Document:	T10/00-172r0	Date:	March 8, 2000
To:	T10 Committee Membership		
From:	Edward A. Gardner, Ophidian Designs		
Subject:	SVP Issues		

1. Relationship of VI Architecture and Infiniband

Infiniband provides a superset of VI Architecture functionality. Unfortunately, the two use different terminology for the same functions. Most of Infiniband's extra features are irrelevant to a SCSI protocol. The main Infiniband feature that might affect SVP is enhanced flow control, discussed in the next section. Note that while Infiniband's Reliable Datagram mode is interesting for SVP, allowing its use does not affect the SVP protocol.

Plan for SVP: Write SVP using VI Architecture (version 1.0) terminology. Plan to include an (informational) annex describing the relationship to Infiniband and mapping VI terminology to Infiniband's. Remove that annex if/when published Infiniband specifications include equivalent information.

2. Flow Control

The VI Architecture (version 1.0) requires strict flow control; sending a message when no receive descriptor is available breaks the connection. As a consequence SVP must use a flow control algorithm such as that outlined in document T10/99-316r1 (SVP Overview).

Infiniband allows optimistic flow control. Sending too many messages results in a negative acknowledgement and periodic retransmission until a receive descriptor becomes available. This allows the traditional SCSI approach of returning Busy status and re-sending commands after a delay. Supposedly the VIDF (VI Developer's Forum) plans to publish a new version of the VI Architecture that incorporates this and other Infiniband enhancements, although the schedule for doing so is not definite.

For SVP we have to choose one of three approaches:

1. Write SVP on a presumption of strict flow control as for VI Architecture (version 1.0).
2. Require Infiniband's enhanced flow control for SVP. The resulting SVP would be fundamentally incapable of operating with many existing VI Architecture implementations, including those based on FC-VI (FC-VI does not allow retransmission).
3. Write SVP to work with both strict and enhanced flow control.

Choices 1 and 3 result in very similar SVP specifications. Both include a credit based flow control algorithm in SVP, such as that outlined in document T10/99-316r1 (SVP Overview). I believe that choice 3 actually results in a simpler document, as it omits any rules governing the interaction of flow control and posting receive descriptors.

Plan for SVP: choice 3.

3. Big Endien vs. Little Endien

The VI Architecture is little endien; the least significant byte of addresses and lengths appears first in memory. Unfortunately SCSI has traditionally been big endien. There is no completely clean resolution to this.

Plan for SVP: Use little endien representation for native VI data items, specifically the data buffer descriptor. Use big endien representation for SCSI data items (LUN and CDB). Sense and response data lengths could arguably go either way, use big endien representation for similarity with FCP (this is arguable). Comments?

4. Unnecessary FCP Features

FCP-2 must deal with out of order delivery and lost frames. In contrast VI provides guaranteed in-order delivery (Reliable Delivery and Reliable Reception connections). These are similar to the delivery guarantees provided by parallel SCSI, and as a consequence SVP should (arguably) be more similar to SPI-3 Information Units rather than to FCP-2.

FCP-2 includes the following features which appear unnecessary for SVP (none are present in SPI-3):

1. Command Reference Numbers, used to detect out-of-order commands for tapes and other order sensitive devices.
2. FCP_CONF, an information unit used to confirm (to the target) that a status response has been received by the initiator.
3. Extensive discussion of error recovery, including provision to retry data transfers.

Plan for SVP: omit all of these, they are redundant with VI Architecture mechanisms.

5. Proposed IU Formats

5.1 SVP_CMND

Bit Byte	7	6	5	4	3	2	1	0
0	TYPE = 01h							
1	RESERVED							
2	TAG							
3								
4	LOGICAL UNIT NUMBER							
11								
12	DATA BUFFER VIRTUAL ADDRESS							
19								
20	DATA BUFFER MEMORY HANDLE							
23								
24	DATA BUFFER LENGTH							
27								
28	RESERVED							
29	RESERVED					TASK ATTRIBUTE		
30	TASK MANAGEMENT FLAGS							
31	RESERVED	ADDITIONAL CDB LENGTH = (N-48)/4					RDDATA	WRDATA
32	CDB							
47								
48	ADDITIONAL CDB							
N								

Notes:

1. Bytes 0 through 11 above are identical to bytes 0 through 11 of the SPI L_Q information unit in SPI-3.
2. Bytes 28 through N above are identical to bytes 0 through N of the command information unit in SPI-3.
3. Bytes 28 through N above are substantially identical to bytes 8 through N of the FCP_CMND payload in FCP-2. FCP-2 uses byte 8 (byte 28 above) for a Command Reference Number.

4. Bytes 12 through 27 above are identical to a VI Architecture data segment descriptor.
5. Strictly speaking a data buffer length (bytes 24 to 27 above) may not be necessary. One can argue whether or not it is desirable. FCP includes one, SPI-3 does not, I prefer having it. Comments?
6. The LUN, CDB and additional CDB fields use big endien representation. The lowest numbered byte is the most significant byte of the data item.
7. The Data Buffer Virtual Address, Data Buffer Memory Handle and Data Buffer Length fields use little endien representation. The lowest numbered byte is the least significant byte of the data item.

5.2 SVP_RSP

Bit Byte	7	6	5	4	3	2	1	0
0	TYPE = 08h							
1	TRD_COUNT_INCREMENT							
2	TAG							
3								
4	LOGICAL UNIT NUMBER							
11								
12	RESERVED							
13	RESERVED							
14	RESERVED				RESIDUNDER	RESIDOVER	SNSVALID	RSPVALID
15	STATUS							
16	SVP_RESID							
19								
20	SENSE DATA LIST LENGTH = N							
23								
24	RESPONSE DATA LIST LENGTH = M							
27								
28	RESPONSE DATA (M BYTES LONG)							
27+m								
28+m	SENSE DATA (N BYTES LONG)							
27+m+n								

Notes:

1. Bytes 4 through the end above are substantially identical to the FCP_RSP payload in FCP-2. The above requires the LUN value, FCP-2 shows that as reserved (the Fibre Channel exchange ID identifies the task independent of the LUN). The above omits FCP_CONF_REQ.
2. Byte 1, TRD_COUNT_INCREMENT, is a signed value used to implement credit based flow control (see T10/99-316r1).

3. The LUN, SVP_RESID, Sense Data List Length, Response Data List Length, Response Data and Sense Data fields use big endien representation. The lowest numbered byte is the most significant byte of the data item.

5.3 SVP_TRD_Adjust

Bit Byte	7	6	5	4	3	2	1	0
0	TYPE = 80h							
1	TRD_COUNT_INCREMENT							
2	RESERVED							
3	RESERVED							

5.4 SVP_TRD_Adjust_Response

Bit Byte	7	6	5	4	3	2	1	0
0	TYPE = 81h							
1	RESERVED							
2	RESERVED							
3	RESERVED							