CONGREUENT SOFTWARE, INC.
98 Colorado Avenue
Berkeley, CA 94707
(510) 527-3926
(510) 527-3856 FAX

FROM:	Peter Johansson
TO:	T10 SBP-3 working group
DATE:	July 15, 2002
RE:	Bare-bones isochronous

At the working group meeting in Cupertino in August, 01-222r0 was presented for discussion. That proposal suggested that SBP isochronous operations could be greatly simplified by collapsing the two isochronous fetch agents described in the SBP-3 working draft into a single one responsible for stream operations.

As much as the proposal was received as a step in the right direction, the working group expressed an interest in even more radical simplification. This proposal is an exploration of those ideas. The goals of 01-222r0 remain the same, in that support is envisioned for the following classes of device:

– Simple devices, such as printers or scanners, which neither record (for subsequent playback) nor playback (from prior recording) streams but have a need to use isochronous, instead of asynchronous, mechanisms for data transfer;

– Simple record and playback devices that operate on isochronous streams that contain a single channel; and

– More sophisticated devices that may filter or mix multiple channels during recording or playback.

I think that the addition of the *isochronous* bit to the normal command block ORB and the *plug_index* field to the CREATE STREAM ORB meets the requirements of the first device class. The second device class may be sufficiently supported by these SBP extensions or may require command set-dependent facilities and methods, while the third, most complex, type of device assuredly requires a command set tailored to its needs.

More discussion is needed on CREATE STREAM ORB. Is it really necessary to create a second fetch agent if the device is isochronous, only? The task set created by LOGIN could be used if there were means to communicate *talker* and *plug_index* to the device.

The following normative text changes are proposed for a forthcoming SBP-3 working draft. Sections and clauses to be deleted in their entirety are listed in the table below (it references SBP-3 Revision 2a); the new text, as well as more piecemeal deletions, follows.

Clause	Proposed change
4.8	Replace entire clause with new informative text (yet to be written)
5.1.3	Delete
9.2a	Stream data transfer (yet to be written)
9.3a	Interim request status (yet to be written)
11	Convert to normative (but optional) annex
12	Replace entire clause with new text (yet to be written)
H	Delete

5.1.2.1 Normal command block ORBs

The format of a normal command block ORB with a single data descriptor is illustrated by the figure below.

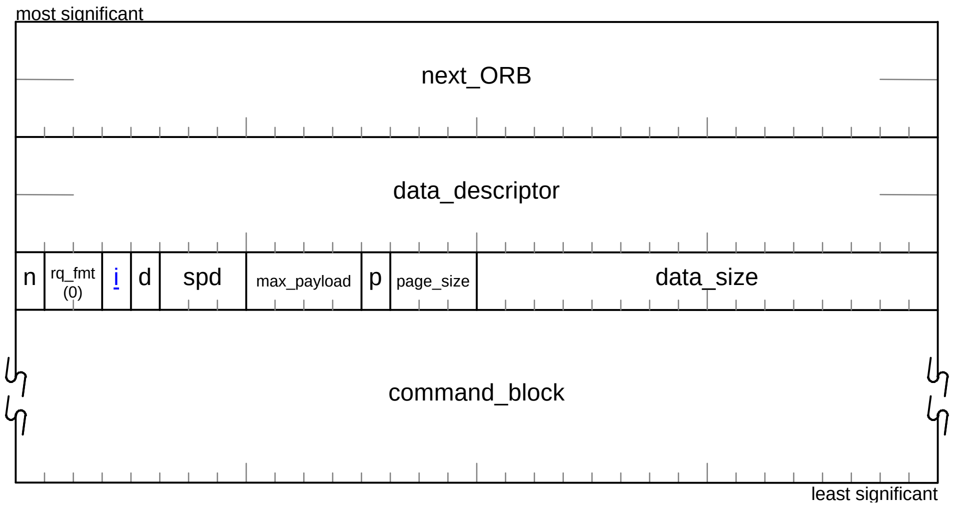

Figure 16 – Normal command block ORB (single data descriptor)

The *next_ORB* field shall contain a null pointer or the address of a dummy ORB or a normal command block ORB and shall conform to the address pointer format illustrated by Figure 12.

The value of the *data_descriptor* field is valid only when *data_size* is nonzero, in which case this field shall contain either the address of the data buffer or the address of a page table that describes the memory segments that make up the data buffer, dependent upon the value of *page_table_present* bit. The format of the *data_descriptor* field, when it directly addresses a data buffer, shall be a 64-bit Serial Bus address or, when it addresses a page table, shall be as specified by Figure 11. When *data_descriptor* specifies the address of a page table, the format of the page table shall conform to that described in 5.2.

The *notify* bit and *rq_fm*t field are as previously defined for all ORB formats. The *rq_fmt* field shall be zero for an ORB which contains a single buffer descriptor.

The *isochronous* bit (abbreviated as *i* in the figure above) specifies the transfer method used for data associated with the ORB. When this bit is zero, Serial Bus read or write transactions shall be used to move data to or from the buffer. Otherwise, when *isochronous* is one, data transfer shall be effected by Serial Bus streams, *i.e.*, packets with a transaction code of A_{16}.

NOTE – The use of Serial Bus streams for the data transfer associated with a command does not preclude the use of *data_descriptor* and its associated fields to describe supplementary or parametric data pertinent to the command. When *isochronous* is one and *data_size* is nonzero, command set-dependent information correlates each data set with its transfer method. For commands that operate on a single data set, the data descriptor fields are ignored when *isochronous* is one.

The *direction* bit (abbreviated as *d* in the figure above) specifies direction of data transfer for the buffer. If the *direction* bit is zero, the target shall use Serial Bus read transactions to fetch data destined for the device medium. Otherwise, when the *direction* bit is one, the target shall use Serial Bus write transactions to store data obtained from the device medium.

The *spd* field specifies the speed that the target shall use for data transfer transactions addressed to the data buffer or page table, as encoded by Table 1.

Table 1 – Data transfer speeds

Value	Speed
0	S100
1	S200
2	S400
3	S800
4	S1600
5	S3200
6 – 7	Reserved for future standardization

The maximum data transfer length is specified as $2^{max\_payload + 2}$ bytes, which is the largest data transfer length that may be requested by the target in a single Serial Bus read or write transaction addressed to the data buffer. The *max_payload* field shall specify a maximum data transfer length less than or equal to the length permissible at the data transfer rate specified by *spd*.

The *page_table_present* bit (abbreviated as *p* in the figure above) shall be zero if *data_descriptor* directly addresses the data buffer. When *data_descriptor* addresses a page table, this bit shall be one.

The *page_size* field shall specify the underlying page size of the data buffer memory. A *page_size* value of zero indicates that the underlying page size is not specified. Otherwise the page size is $2^{page\_size + 8}$ bytes.

When *page_table_present* is one, the *page_size* field also specifies the format of the data structure that describes the data buffer. A *page_size* value of zero implies the unrestricted page table format (also known as a scatter/gather list). Otherwise, a nonzero *page_size* indicates a normalized page table.

If *page_table_present* is zero, the *data_size* field shall contain the size, in bytes, of the system memory addressed by the *data_descriptor* field. Otherwise *data_size* shall contain the number of elements in the page table addressed by *data_descriptor*.

The *command_block* field contains information not specified by this standard.

NOTE – The normal command block ORB with dual data descriptors is omitted from this proposal because it is unaltered from SBP-3 Revision 2a.

5.1.4.3 Create stream ORB

Before any stream requests are made of a target, the initiator shall first complete a create stream procedure that uses the ORB format shown below.

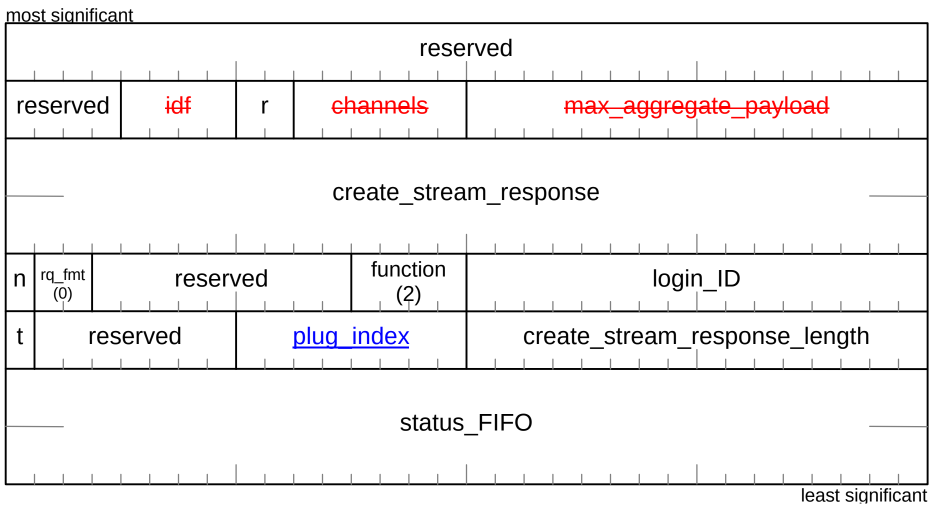

Figure 27 – Create stream ORB

~~The *idf* field specifies the format of recorded isochronous data. Valid values for *idf* are shown in the table below.~~

~~**Value**~~	~~**Data format**~~	~~**Description**~~	~~**Reference**~~
~~0~~	~~Unspecified~~	~~The data format is determined by the command set or device type and is beyond the scope of this standard~~	
~~1~~	~~—~~	~~Reserved for future standardization~~	
~~2~~	~~Isochronous data interchange (without cycle mark indices)~~	~~The data is structured by cycle marks, which are recorded on the medium.~~	~~Section 11 (also 12.2.3)~~
~~3~~	~~Isochronous data interchange (with cycle mark indices)~~	~~The data is structured by cycle marks which are recorded on the medium; a cycle mark index is present every 512 bytes.~~	~~Section 11 (also 11.3 and 12.2.3)~~
~~4 – F_{16}~~	~~—~~	~~Reserved for future standardization~~	

~~The *channels* field specifies the maximum number of isochronous channels that are to be simultaneously transmitted or received.~~

~~The *max_aggregate_payload* field is the aggregate maximum isochronous payload that the target is requested to support for the stream. That is, the sum of all the *data_length* fields of Serial Bus isochronous packets transmitted or received for all of the stream's isochronous channels shall not exceed *max_aggregate_payload* in a single isochronous period. The target is not required to enforce this limit.~~

The *create_stream_response* and *create_stream_response_length* fields specify the address and size of a buffer allocated for the return of the create stream response. The *create_stream_response* field shall conform to the format for address pointers specified by Figure 11. The buffer shall be in the same node as the initiator and shall be accessible to a Serial Bus block write transaction with a data transfer length less than or equal to *create_stream_response_length*. The initiator shall set *create_stream_response_length* to a value of at least ~~24~~12; the target may ignore this field.

The *notify* bit and the *rq_fmt* field are as previously defined for management ORB formats.

The *login_ID* field shall contain a login ID value obtained as the result of a successful login.

The *talker* bit (abbreviated as *t* in the figure above) shall specify the type of isochronous stream requested. If the target resources are to be configured for listening, *talker* shall be zero.

The *plug_index* field may uniquely identify a single source or sink or stream data within the context of the target. If *plug_index* is zero, command set-dependent methods specify the parameters of all stream data. When *plug_index* is in the range one to $1F_{16}$, inclusive, it identifies a plug control register that mediates reception or transmission of stream data. The type of plug control register referenced, either an input or output plug control register, is determined by the value of the *talker* bit. Consult IEC 61883-1 for the specification of plug control registers. Other values of *plug_index* are reserved for future standardization.

The *status_FIFO* field is as previously defined for management ORB formats and shall contain an address allocated for the return of status for the CREATE STREAM request, status for all subsequent requests signaled to either the *stream_command_block_agent* or *stream_control_agent* allocated for this login and any unsolicited isochronous error report(s) generated by the logical unit for this stream.

If the create stream request fails the contents of the response buffer are unspecified. Otherwise, upon successful completion of a create stream request, the response is returned in the format illustrated below.

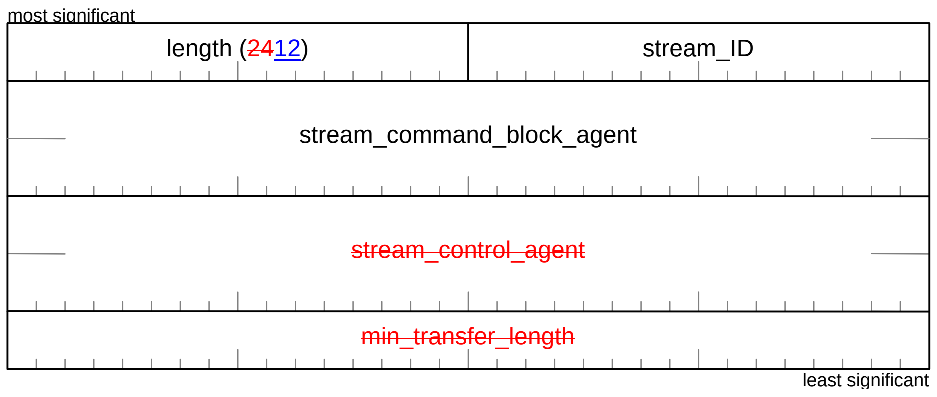

Figure 28 – Create stream response

The *length* field shall contain the length, in bytes, of the create stream response data and shall be equal to ~~24~~12.

The *stream_ID* identifies an isochronous stream for which target resources have been allocated. The initiator shall use this value to identify all subsequent requests directed to the target's management agent that pertain to this stream.

~~The *stream_command_block_agent* and *stream_control_agent* fields specify the base address of the agent's CSRs, which are defined in 6.4. Both fields shall conform to the format for address pointers specified by Figure 11. The *node_ID* portion of either field shall have a value equal to the most significant~~

~~16 bits of the target's NODE_IDS register. If the target does not implement a stream control agent, the contents of *stream_control_agent* are unspecified and shall be ignored by the initiator.~~

~~The *min_transfer_length* field advises the initiator of the minimum *stream_length* value desired by the target in stream command block ORBs in order to sustain the isochronous data transfer rate requested by the login. If the initiator presents any stream command block ORBs whose *stream_length* value is less than this minimum, the target may experience underflow or overflow in isochronous data while talking or listening at the requested rate. Even if this minimum is respected it is still possible for underflow or overflow to occur.~~