CONGRUENT 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: February 7, 2003

RE: SBP-3 isochronous operations

This document proposes informative and normative text for SBP-3 isochronous operations and completes an idea first put forward at the working group meeting in Huntington Beach, namely that the isochronous transport facilities of SBP-3 could be used with simple devices, *e.g.*, DVD players, printers or scanners, without modification of their command sets if there were a mechanism to associate an isochronous channel (and, in the case of an output stream, speed and maximum data payload) with the data to be transferred.

Central to the proposal is the use of an input or output plug control register (iPCR or oPCR) associated with a logical unit to provide channel number, speed and maximum data payload per isochronous subaction to a command set otherwise unaware of these essential features of Serial Bus isochronous data transfer. A method for associating an iPCR or oPCR with a logical unit, the configuration ROM Plug_Control_Register entry, has already been added to the draft (see T10/03-009r1). This document describes how an initiator allocates Serial Bus isochronous resources, programs a plug control register to establish a point-to-point connection with a logical unit and then signals command block ORBs to the logical unit to effect the isochronous data transfer.

4.8 Streams

Streams are objects that are based upon the isochronous capabilities of Serial Bus. A stream consists of all of the target and logical unit functions and resources that are necessary to transfer isochronous data from one or more Serial Bus channels to the device's medium (the target is a listener) or to transfer data isochronously from the device's medium to one or more Serial Bus channels (the target is a talker). The direction, listener or talker, of any stream is independent of any other stream. Within each stream all of the data flows in the same direction.

Streams require Serial Bus resources as well as target resources. These include the aggregate bandwidth necessary for the stream, the channel numbers utilized by the stream and the isochronous connections that characterize the stream. An application (usually, but not necessarily, co-located with the initiator) allocates all necessary resources before activating a logical unit isochronous stream.

A stream of isochronous data appears on Serial Bus as subactions with a transaction code (*tcode*) of A_{16} during an isochronous period. This in turn is represented by an ordered byte stream of data on the device medium. The presentation of this data is controlled by command block ORBs that request data transfer to or from the medium.

Figure 8a illustrates the relationship between the different stream components during playback (the logical unit is assumed to have direct-access capabilities).

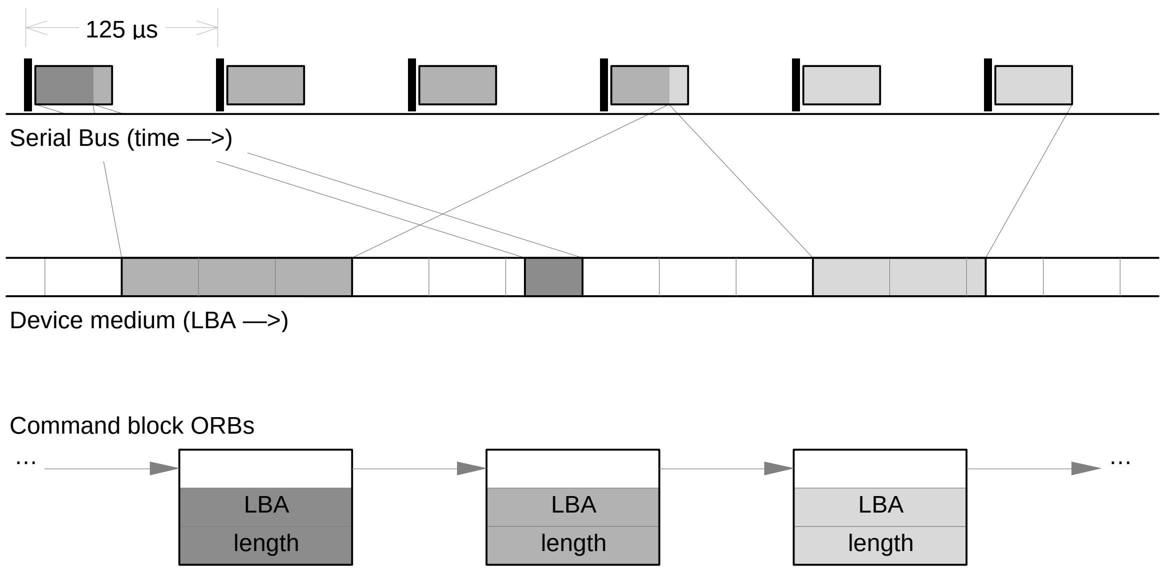

Figure 8a – Components of an isochronous stream (direct-access logical unit)

The figure shows that the order of presentation of bytes in a stream is determined by the order of command block ORBs—but that this order is independent of the location of the data on device medium. In this example, the size of the isochronous packet transmitted each cycle is determined by information previously recorded on the medium. The example shows a stream with only one channel (one packet per isochronous period) and a fixed packet size, but streams may consist of more than one channel and the packet size may be different each isochronous period.

Streams differ fundamentally from asynchronous data transfers. First, streams do not require any address context for the transfer of data to or from system memory: a channel number and the time-ordered location of the data within the stream identify the data. Second, the stream's flow of isochronous data may be controlled and synchronized to time or other time-dependent events.

In order to fully exploit these differences, a logical unit's command set should be cognizant of Serial Bus isochronous behavior. Such a command set may enable a logical unit to perform frame-precise synchronization or to manage multi-channel streams, to give just two examples. However, it is possible to adapt existing command sets (ones designed without awareness of Serial Bus isochronous facilities) to use isochronous data transfer methods.

In order to leverage existing command sets, simplifying assumptions are required. The devices are limited to single-channel streams, either input or output but not both the same time. The essential information that characterizes his data flow on a single isochronous channel is a) bandwidth, b) channel number and c) transmission speed. IEC 61883-1 defines plug control registers (PCRs) that provide this information. Output plug control registers (oPCRs) specify speed, channel number and maximum data payload per isochronous subaction (bandwidth) while input control registers (iPCRs) specify only channel number. The plug control registers also enable connection management and a simple on or off scheme to control the flow of isochronous data.

The information available in a plug control register is combined with data transfer length information in a command block ORB when the *isochronous* bit in the ORB is set to one; this causes the logical unit to use isochronous subactions to transfer data to or from the device medium in accordance with the parameters in the plug control register.

One example of a simple, single data stream device that might benefit from isochronous data transfer would be an isochronous DVD player. Contemporary DVD players use a multi-media command set (MMC) that assumes a transport protocol that provides confirmed, asynchronous data transfer. This is workable when the rendering device (typically a computer) and the DVD player are the only devices that share the transport medium: ample bandwidth is available. But if the DVD player is an SBP device and is in competition with many other devices for Serial Bus bandwidth, the best-effort nature of asynchronous data transfer on Serial Bus might result in late delivery of data to the rendering device. A practical solution is to leverage existing command set, unchanged, and use the isochronous facilities of SBP-3 to implement an isochronous DVD player. If isochronous bandwidth is reserved in advance, the rendering device should receive all of the data transmitted by the DVD player in time for display.

The steps an initiator (which is also co-located with the rendering application) would take to control an isochronous DVD player are as follows:

a) Read the DVD player's plug control registers to determine the maximum speed and maximum data payload supported by the DVD player and attempt to allocate the required bandwidth;

b) If the bandwidth allocation is successful, attempt to allocate a channel number. If this fails, release the previously allocated bandwidth and try again later;

c) Otherwise, program the DVD player's output plug control register (oPCR) with the speed, channel number and maximum data payload information and at the same time establish a point-to-point connection in accordance with the procedures specified by IEC 61883-1;

d) Program Serial Bus adapter hardware (co-located with the initiator) to receive isochronous data on the specified channel;

e) Signal command block ORBs to the DVD player to request that it transmit data from its device medium as a sequence of isochronous subactions;

f) When complete, program the DVD player's oPCR to relinquish the point-to-point connection in accordance with the procedures specified by IEC 61883-1 and then release the previously allocated bandwidth and channel number.

An initiator would utilize a similar procedure to control isochronous data transfer for logical unit that is a listener, for example and isochronous scanner.

9.3a Isochronous data transfer

The isochronous transfer of data associated with a command block ORB is only partly the responsibility of the logical unit. Depending upon the logical unit's role, either talker listener, successful data transfer also depends upon the other endpoint; either the listeners' correct receipt or the talker's correct transmission of isochronous subactions are essential. When an ORB specifies a data stream (*i.e.*, *data_size* is nonzero) and the *isochronous* bit is one, the target shall transmit or receive Serial Bus isochronous subactions according to whether the *direction* bit in the ORB is one or zero, respectively.

If the logical unit is a talker, it shall transmit the data specified by the command in a sequence of isochronous subactions, zero or one per isochronous period; the data shall be transmitted on Serial Bus in the identical order it was obtained from the device medium or other source. The total transfer length may be larger than the maximum data payload that can be accommodated in a single isochronous subaction. The logical unit shall limit the size of each isochronous subaction to less than or equal to the maximum permitted; this information may be obtained either from the logical unit's oPCR (see 11.1) or by command set-dependent means. The channel number and the speed at which the isochronous subaction is to be transmitted shall be obtained from the same source as the maximum data payload.

If the logical unit is a listener, it shall receive the data specified by the command from a sequence of isochronous subactions, zero or one per isochronous period; the data shall be stored on the device medium or other destination in the identical order it was received from Serial Bus. The channel number from which to receive the information may be identified either by the logical unit's iPCR (see 11.2) or by command set-dependent means. The logical unit shall allocate buffers each one of which shall be sufficiently large to receive the maximum data payload that may arrive in any isochronous period. Unless the maximum data payload is determined by command set-dependent means, the logical unit shall allocate buffers whose size is equal to the maximum isochronous subaction size permitted by IEEE 1394 for the data rate capability specified in the target's iMPR.

Because isochronous subactions are unacknowledged, some kinds of data transfer errors are undetectable by the logical unit. Absent a higher-level protocol, a talker cannot determine if a transmitted isochronous subaction is correctly received by the intended listeners nor can a listener determine, in all circumstances, if it has failed to receive isochronous subactions transmitted by the talker. Certain data transfer errors can be detected; these include the following:

– Failure to observe a cycle start subaction. Either the talker or the listener may detect this condition. Recovery strategies are implementation-dependent but might include a) ignoring the missed cycle start or b) terminating the task and aborting the task set.

– Data underrun at the talker. A malformed isochronous subaction results, because the talker is unable to obtain data from the device medium or other source in time for transmission on Serial Bus. Recovery strategies are implementation-dependent but might include a) terminating the task and aborting the task set, b) padding the remainder of the isochronous subaction with constant fill data and generating a valid CRC or c) truncating the isochronous subaction or generating an invalid CRC or both. A choice between b) and c) is strongly dependent on the data format and the effect of its receipt on the listeners.

– Data overrun at the listener. A loss of data occurs, because the listener resources are busy and the listener is unable to correctly receive all of the data from the isochronous subaction. Recovery strategies are implementation-dependent but might include a) terminating the task and aborting the task set, b) if the isochronous subaction's header was correctly receive and the *data_length* of the payload is known, filling some or all of the isochronous subaction with data or c) discarding the entire isochronous subaction. A choice between b) and c) is strongly dependent on the data format.

- Data length or CRC error detected by the listener. As in the case of data overrun, a loss of data occurs because the listener is unable to correctly receive all of the data from the isochronous subaction. The same recovery strategies are applicable and the same considerations about data format apply.

Logical units that implement command sets designed with knowledge of Serial Bus isochronous behavior should mandate specific recovery strategies for each of the error cases above. Logical units that implement command sets designed without such knowledge to consider the recommendations made in 11.3.

11 Isochronous operations

~~For each active channel on Serial Bus, isochronous data consists of zero or one isochronous packet transmitted by a talker in an isochronous period and received by zero or more listeners in the same isochronous period. This section describes how an initiator may control isochronous data transfers when a target is either the talker or a listener.~~

This section describes procedures that may be used by an initiator to request a logical unit to transfer data associated with a command by isochronous instead of asynchronous methods. Two fundamentally different methods are available to control logical unit isochronous operations:

- Command set-dependent. This provides the most flexible and fully featured control of isochronous operations, since the command set is designed with intimate knowledge of Serial Bus isochronous behavior. Command set-dependent methods are beyond the scope of this standard—although an example of the transport of one such command set is given in Annex D.
- Transport protocol-dependent. This method leverages existing command sets designed without knowledge of Serial Bus isochronous behavior. Because such command sets do not specify all the information necessary to transfer data isochronously, supplemental information available from the transport protocol or other external source is merged with the data transfer length obtained from each command. This approach makes simplifying assumptions and is not as flexible as command set-dependent methods.

The remainder of this section specifies how to use SBP-3 facilities in conjunction with the connection management methods described in IEC 61883-1 to control isochronous operations of devices whose command sets are unaware of the isochronous facilities of Serial Bus. The methods are applicable only to devices that transmit no more than one isochronous output stream and receive no more than one isochronous input stream at a time. Example devices might include DVD players or printers (talkers) and scanners (listeners).

~~From the perspective of Serial Bus,~~ Control of isochronous ~~streams~~ operations involves the following elements:

- ~~the allocation of target resources;~~
- the allocation of Serial Bus resources, such as channel numbers and bandwidth (isochronous resource management);
- the establishment or breaking of connections between the ~~target~~ logical unit and the talker ~~and~~ or the listener(s) (connection management);
- the transfer of isochronous data to or from the ~~target's~~ logical unit's medium (command block ORBs);
- ~~the starting, stopping and synchronization of isochronous data reception or transmission by the target from or to Serial Bus; and~~
- ~~the allocation of Serial Bus resources, such as channel numbers or bandwidth.~~

Since there are significant differences between isochronous talkers and listeners, the operational details are described separately in the clauses that follow.

11.1 Talker operations

An isochronous talker is permitted to transmit zero or one isochronous subactions per isochronous period on a specified channel; the maximum data payload of the subaction is constrained both by the transmission speed and by the bandwidth previously allocated for the talker. The maximum data payload permitted at different transmission speeds is specified by IEEE 1394 and, for convenience of reference, is summarized by the table below.

Table 3 – Maximum payload for isochronous subactions

Speed	Maximum data payload (bytes)
S100	1024
S200	2048
S400	4096
S800	8192
S1600	16384
S3200	32768

Since the logical unit's command set has no means to specify channel, speed or maximum data payload, the logical unit shall obtain this information from the output plug control register (oPCR) identified by the configuration ROM Plug_Control_Register entry (whose *direction* bit is one) associated with the logical unit. If there is more than one such entry with a *direction* bit equal to one, unpredictable behavior beyond the scope of this standard may result. The initiator programs the channel and speed information in the oPCR but the maximum data payload is implementation-dependent and provided by the logical unit. Whenever the logical unit is ready to accept data transfer commands, the *online* bit in the oPCR shall be one and the *payload* field shall report the logical unit's maximum data payload per isochronous subaction, in quadlets. The size reported by the *payload* field shall not exceed the maximum data payload permitted for the speed reported by the *data_rate_capability* field in the target's output master plug register (oMPR).

NOTE – A *payload* value of zero encodes a size of 1024 quadlets, per IEC 61883-1.

Before an initiator signals data transfer commands to a logical unit within ORBs whose *isochronous* bit is one, it shall allocate the necessary isochronous resources and program the logical unit's oPCR as specified below:

a) The initiator shall read the target's oMPR to determine the fastest transmission speed supported and shall read the logical unit's oPCR to determine the maximum data payload;

b) The initiator shall select a transmission speed[1] less than or equal to the fastest speed supported by the target and shall adjust the maximum data payload to the smaller of the value obtained from the logical unit's oPCR and the maximum permitted at the selected transmission speed. The initiator shall attempt to obtain the necessary bandwidth for the data payload from the isochronous resource manager's BANDWIDTH_AVAILABLE register. If the bandwidth is unavailable, the initiator shall not signal any command block ORBs to the logical unit with an *isochronous* bit equal to one;

c) Otherwise, the initiator shall attempt to allocate a channel from the isochronous resource manager's CHANNELS_AVAILABLE register. If no channel is available, the initiator shall release the bandwidth previously obtained and shall not signal any command block ORBs to the logical unit with an *isochronous* bit equal to one;

d) Once both bandwidth and channel have been allocated, the initiator shall program the logical unit's oPCR with channel number and speed and shall increment the point-to-point connection count in accordance with the procedures specified by IEC 61883-1. If the speed selected by the

[1] The choice of transmission speed is influenced both by Serial Bus topology between the target (talker) and one or more listeners and by the speed capabilities of the listeners.

initiator limits the maximum data payload to a value smaller than that reported by the logical unit in the oPCR *payload* field, the logical unit shall update the *payload* field with the maximum data payload permitted for the selected speed.

After the oPCR has been programmed with channel number and speed and the point-to-point connection count is nonzero, the logical unit is ready to accept command block ORBs whose *isochronous* bit is one. Because the data shall be transmitted isochronously, the ORB shall not specify the address of a buffer but shall specify a total *data_size* for the data to be transmitted by the command; the *direction* bit in the ORB shall be one (see 5.1.2). The logical unit shall transmit the requested data on Serial Bus as specified by 9.3a. When all of the requested data, up to the limit of *data_size*, has been transmitted, the logical unit shall store completion status for the ORB at the initiator's *status_FIFO*.

When the initiator has concluded isochronous operations with the logical unit (either because logout is imminent or in anticipation of substantial logical unit idle time), the initiator shall follow the procedures specified by IEC 61883-1 to program the logical unit's oPCR and decrement the point-to-point connection count. The initiator shall also release the isochronous resources, bandwidth and channel number, previously allocated.

11.2 Listener operations

An isochronous listener expects to receive zero or one isochronous subactions per isochronous period on a specified channel; the maximum data payload of the subaction is constrained by the reception speed as specified by IEEE 1394 and summarized in Table 3.

Since the logical unit’s command set has no means to specify channel number, the logical unit shall obtain this information from the input plug control register (iPCR) identified by the configuration ROM Plug_Control_Register entry (whose *direction* bit is zero) associated with the logical unit. If there is more than one such entry with a *direction* bit equal to zero, unpredictable behavior beyond the scope of this standard may result. The initiator programs the channel number information in the iPCR. Whenever the logical unit is ready to accept data transfer commands, the *online* bit in the iPCR shall be one.

The size reported by the *payload* field shall not exceed the maximum data payload permitted for the speed reported by the *data_rate_capability* field in the target's output master plug register (oMPR).

Before an initiator signals data transfer commands to a logical unit within ORBs whose *isochronous* bit is one, it shall either allocate the necessary isochronous resources or determine that they have been allocated and program the logical unit's iPCR as specified below:

a) The initiator shall read the target's oMPR to determine the fastest reception speed supported and shall examine Serial Bus topology between the talker and the target (listener) to determine the fastest speed isochronous subactions may be transmitted from the talker to the target;

b) If the initiator controls the talker, it shall configure it to transmit isochronous subactions no faster than the speed determined in the preceding step. Otherwise, the initiator shall determine the speed at which the talker is transmitting or will transmit isochronous subactions; if it is greater than speed determined in the preceding step, the initiator shall not signal any command block ORBs to the logical unit with an isochronous bit equal to one;

c) If the initiator does not control the talker it shall skip this step. Otherwise, it shall determine the maximum data payload for the talker's isochronous subactions and shall attempt to obtain the necessary bandwidth for the data payload from the isochronous resource manager's BANDWIDTH_AVAILABLE register. If the bandwidth is unavailable, the initiator shall not signal any command block ORBs to the logical unit with an isochronous bit equal to one;

d) If the initiator does not control the talker it shall skip this step. Otherwise, it shall attempt to allocate a channel from the isochronous resource manager's CHANNELS_AVAILABLE register. If no channel is available, the initiator shall release the bandwidth previously obtained and shall not signal any command block ORBs to the logical unit with an isochronous bit equal to one;

e) Otherwise, the initiator shall attempt to allocate a channel from the isochronous resource manager's CHANNELS_AVAILABLE register. If no channel is available, the initiator shall release previously obtained bandwidth, if any, and shall not signal any command block ORBs to the logical unit with an isochronous bit equal to one;

f) Once both bandwidth and channel have been allocated (whether by the initiator or another device), the initiator shall program the logical unit's iPCR with channel number and shall increment the point-to-point connection count in accordance with the procedures specified by IEC 61883-1.

After the iPCR has been programmed with channel number and the point-to-point connection count is nonzero, the logical unit is ready to accept command block ORBs whose *isochronous* bit is one. Because the data shall be received isochronously, the ORB shall not specify the address of a buffer but shall specify a total *data_size* for the data to be received by the command; the *direction* bit in the ORB shall be zero (see 5.1.2). The logical unit shall receive the requested data from Serial Bus as specified by 9.3a. When all of the requested data, up to the limit of *data_size*, has been received, the logical unit shall store completion status for the ORB at the initiator's *status_FIFO*.

When the initiator has concluded isochronous operations with the logical unit (either because logout is imminent or in anticipation of substantial logical unit idle time), the initiator shall follow the procedures specified by IEC 61883-1 to program the logical unit's iPCR and decrement the point-to-point connection count. If the initiator previously allocated isochronous resources, bandwidth and channel number, it shall release them.

11.3 Implementation recommendations (informative)

The following suggestions are intended as useful guidance for implementers. They may not be the most appropriate choices for all command sets. For example, a device whose command set is cognizant of data streaming requirements may find it preferable to ignore certain isochronous errors rather than abort tasks and task sets.

- The *online* bits in the logical unit's plug control registers should be zero when no initiator is logged in;
- An initiator should perform an exclusive login before programming a logical unit's plug control registers;
- Logical units should ignore missed cycle start indications. Talkers may transmit the data during the next isochronous period; listeners that implement "loose isochronous" reception (as permitted by IEEE Std 1394a-2000) are likely to receive any isochronous subaction intended for them even if they failed to observe the cycle start subaction;
- A talker that detects data underrun during transmission or a listener that detects data overrun during reception of an isochronous subaction should abort the task and the task set to which it belongs;
- An initiator receiving data from a talking logical unit should, once completion status has been stored at the initiator's *status_FIFO*, verify that the quantity of data received is equal to the quantity expected. If there is a mismatch, the initiator should abort the task set and reissue the command;
- An initiator transmitting data to a listening logical unit should time the completion of the command in order to detect isochronous data transfer errors. This is because the listening logical unit cannot determine when all data has been transferred except when the ORB *data_size* field is reached or exceeded. An exception to this recommendation exists when the data format used by the logical unit's command set is self-descriptive and the logical unit is capable of parsing the received data to autonomously determine when data transfer is complete.

The preceding recommendations have been made from the point of view that when the logical unit is the talker the initiator is the listener and vice versa, but this is not a requirement. The initiator may be neither talker nor listener and, if the logical unit is a listener, may not necessarily be in control of talker.