CONGRUENT SOFTWARE, INC.

3998 Whittle Avenue
Oakland, CA 94602
(510) 531-5472

FROM: Peter Johansson
TO: T10 SBP-2 *ad hoc* Working Group
DATE: February 14, 1997
RE: Transport Status for SBP-2

The proposal below is intended to clarify and enhance the definition of *sbp_status* in SBP-2 Revision 2. For some time there has been general agreement that the status definition was not yet complete. Here is a starting point for discussion. Modifications to the existing SBP-2 Revision 2 clause are shown by change bars.

5.3 Status block

Upon completion of a request, if the *notify* bit in the ORB is one or if there is error status to report, the target shall signal the initiator by storing all or part of the status block shown below. If the *status_FIFO* address is explicitly provided as part of the ORB to which the status pertains, the target shall store the status block at the address specified. Otherwise, the target shall store the status block at the *status_FIFO* address provided by the initiator as part of the login parameters. The target may also store unsolicited status at this address, as defined in 9.4.

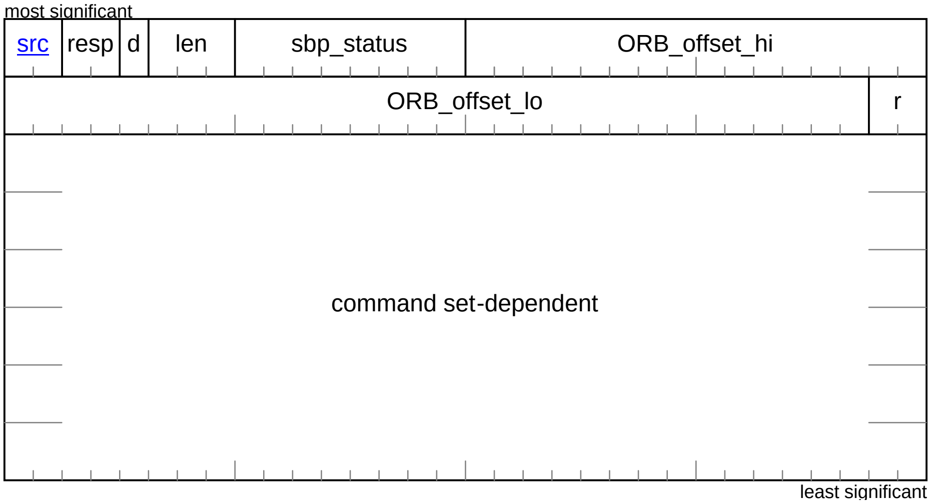

Figure 28 – Status block format

The target shall store a minimum of eight bytes of status information and may store up to the entire 32 bytes defined above so long as the amount of data stored is an integral number of quadlets. A truncated status block shall be interpreted as if the omitted fields had been stored as zeros. The target shall use a single Serial Bus block write transaction to store the status block at the *status_FIFO* address.

The *src* field indicates the origin of the status block, as encoded by the table below.

Value	Description
0	The status block pertains to an ORB identified by *ORB_offset*; at the time the ORB was most recently fetched by the target the *next_ORB* field did not contain a null pointer.
1	The status block pertains to an ORB identified by *ORB_offset*; at the time the ORB was most recently fetched by the target the *next_ORB* field was null.
2	The status block is unsolicited and contains device status information; the contents of the *ORB_offset* field shall be ignored.
3	The status block is unsolicited and contains isochronous error report information as specified by X.

~~The *unsolicited* bit (abbreviated as *u* in the figure above) shall specify the usage of the *ORB_offset* field. If the *unsolicited* bit is zero, the status block pertains to a request identified as described below. When *unsolicited* is one, the status block is not related to any outstanding request and the contents of *ORB_offset* shall be ignored.~~

~~The *end_of_list* bit (abbreviated as *e* in the figure above) shall specify the value of the *next_ORB* field of the ORB to which the status block pertains at the time the ORB was most recently fetched. When *end_of_list* is zero, the *next_ORB* field did not contain a null pointer. Otherwise *next_ORB* was null when last fetched.~~

The *resp* field shall specify the SBP-2 response status for the request identified by *ORB_offset*. Response values are encoded by *resp* as shown by the table below.

Value	Name	Description
0	REQUEST COMPLETE	The request completed without transport protocol error (Either *sbp_status* or command set-dependent status information may indicate the success or failure of the request)
1	TRANSPORT FAILURE	The target detected a nonrecoverable transport failure that prevented the completion of the request
2	ILLEGAL REQUEST	There is an unsupported field or bit value in the ORB; the *sbp_status* field may provide additional information
3	VENDOR DEPENDENT	The meaning of *sbp_status* shall be specified by the vendor

The *dead* bit (abbreviated as *d* in the figure above) shall indicate whether or not the target fetch agent transitioned to the dead state upon storing the status block. When *dead* is zero, the reported status has not affected the state of the fetch agent. If the *dead* bit is set to one, the fetch agent transitioned to the dead state as a consequence of the error condition reported by the status block.

The *len* field shall specify the quantity of valid status block information stored at the *status_FIFO* address. The size of the status block is encoded as *len* + 1 quadlets.

The *sbp_status* field provides additional information that qualifies the response status in *resp*. The meanings assigned to *sbp_status* vary according to the value of *resp* and are described below~~are specified by the table below~~.

When *resp* is equal to zero, REQUEST COMPLETE, the possible values for *sbp_status* are specified by the table below. Any value not enumerated is reserved for future standardization.

Value	Description
0	No additional status to report
1	Request type not supported ~~Invalid request type~~
2	Speed not supported
3	Page size not supported
4	Access denied
5	Logical unit not supported
6	Maximum payload too small
7	Too many channels
8	Resources unavailable
9	Function rejected
10	Login ID not recognized
11	Dummy ORB completed
12	Request aborted
FF_{16}	Unspecified error

If a Serial Bus error occurs in the transport (*resp* is equal to one, TRANSPORT FAILURE), the *sbp_status* field either shall have a value of FF_{16}, unspecified error, or else the field shall be redefined as illustrated below. This format provides for the return of additional information about the transport failure.

most significant   least significant

object	reserved	serial_bus_error

Figure 29 – TRANSPORT FAILURE format for *sbp_status*

The *object* field shall specify which component of an SBP-2 request, the ORB, the data buffer or the page table, was referenced by the target when the error occurred. The value of *object* shall be as defined by the following table.

Value	**Referenced object**
0	Operation request block (ORB)
1	Data buffer
2	Page table
3	Unable to specify

The *serial_bus_error* field shall specify the error response for the failed request, as encoded by the table below.

Value	**Serial Bus error**	**Comment**
0	Missing acknowledge	
1	Reserved; not to be used	
2	Time-out error	An *ack_pending* was received for the request but no response was received within the time-out limit
3	Reserved; not to be used	
4 – 6	Busy retry limit exceeded	The value reflects the last acknowledge, *ack_busy_X*, *ack_busy_A* or *ack_busy_B*.
7 – B$_{16}$	Reserved for future standardization	
C$_{16}$	Conflict error	A resource conflict was detected by the addressed node
D$_{16}$	Data error	The data field failed the CRC check or the observed length of the payload did not match the data_length field
E$_{16}$	Type error	A field in the request was set to an unsupported value or an invalid transaction was attempted (*e.g.*, a write to a read-only address)
F$_{16}$	Address error	The destination_offset field specified an inaccessible address in the addressed node

In the cases of conflict error and data error, these are errors that the target may retry up to an implementation-dependent limit before reporting TRANSPORT FAILURE.

No additional information is provided in *sbp_status* when *resp* equals two, ILLEGAL REQUEST. In this case, *sbp_status* shall be set to FF$_{16}$.

> NOTE – An SBP-2 response code of ILLEGAL REQUEST shall not be used to indicate unsupported fields or bit values in the command set-dependent portion of the ORB. This response code shall be used only to indicate an error in the first 20 bytes of the ORB.

The *ORB_offset_hi* and *ORB_offset_lo* fields together form an *ORB_offset* field that uniquely identifies the ORB to which the status block pertains. If *src* is zero or one~~*unsolicited* is zero~~, the target shall form *ORB_offset* from the least significant 48 bits of the Serial Bus address used to fetch the ORB; the least significant two bits shall be discarded. When the status block contains unsolicited device status (*src* is equal to two), the *ORB_offset* field shall be ignored by the initiator. Otherwise, when *src* equals three, the status block contains an isochronous error report in the format illustrated below.

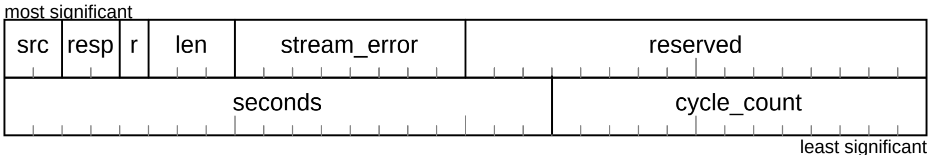

Figure 30 – Unsolicited status format for isochronous errors

The *src* field shall be three to indicate unsolicited status that describes an isochronous error.

The *resp* field shall specify a response status of TRANSPORT FAILURE.

The *len* field shall be equal to one.

The *stream_error* field shall specify the nature of the isochronous error, as encoded by the table below.

Value	**Stream error description**
0	Reserved (not to be used)
1	Missing CYCLE START packet
2	Data CRC error in received isochronous packet
3	Data length error in received isochronous packet
4	Internal underflow with the result that recorded isochronous data was not transmitted on Serial Bus
5	Internal overflow with the result that isochronous data observed on Serial Bus was not recorded on the medium
6 — FE_{16}	Reserved for future standardization
FF_{16}	Unspecified error

The *seconds* field shall contain the least significant 19 bits of the BUS_TIME register at the time of the isochronous stream error.

The *cycle_count* field shall contain the cycle count, between zero and 7999, at the time of the error. The cycle count shall be obtained from the target's free-running cycle timer and shall not be latched from the last observed CYCLE START packet.

For all status block formats, the remainder of the status block after the first two quadlets, up to an overall maximum of 32 bytes, is command set-dependent.