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: June 26, 2001

RE: Distributed data buffers for SBP-3

SBP-2 requires the data buffer and a page table that describes it to be co-located within the same node. This proposal was developed by the working group in Chicago in response to a request by Eric Anderson; it permits the data buffer and page table to be in different nodes and it also permits the data buffer to be distributed among multiple nodes.

The proposed facility is optional and would be identified by a new *distributed_data* bit in the Unit_Characteristics entry in configuration ROM.

5.2 Page tables

The data buffer specified by a normal command block ORB is described by the *data_descriptor*, *page_table_present*, *page_size* and *data_size* fields. The data buffer is a logically contiguous area in system memory. As previously described, when *page_table_present* is zero, the data buffer is also contiguous within Serial Bus address space and no more than 65,535 bytes in length. In this case, *data_descriptor* contains the 64-bit address of the data buffer and *data_size* specifies its length, in bytes.

When the data buffer cannot be directly addressed (either because it is discontiguous or too large), it is necessary to describe it *via* a page table. A page table is a variable-length array of elements, each of which describes a segment that is contiguous within Serial Bus address space. Page table elements are eight bytes long and shall be octlet aligned.

The presence of a page table is indicated by the value of *page_table_present* in the ORB. When *page_table_present* is one, the *data_descriptor* field in the ORB shall contain the address of the page table and the *data_size* field shall contain the number of elements in the page table.

Page tables may have one of two formats: an unrestricted page table or a normalized page table. The page table format is determined by *page_size*. When *page_size* is zero there are no underlying page boundaries to restrict the size or alignment of data buffer segments; this is the unrestricted format. Otherwise the size and alignment of data buffer segments is determined by the nonzero *page_size*; this is the normalized format.

~~When a page table is used it shall be located in the same node as the data buffer it describes.~~ The *spd* and *max_payload* fields of the ORB shall describe data transfer capabilities for ~~both the data buffer and~~ the page table; if the data buffer is co-located within the same node, these fields also pertain to the data buffer. Otherwise, when the data buffer is distributed among one or more nodes (at least one of which is not the node that contains the page table), the data transfer capabilities for each node shall be described by a node selector entry (see 5.2.3) embedded within the page table. Whether the data buffer is distributed or contained within a single node, system memory addressed by a target request subaction that accesses the data buffer shall be entirely contained within a data buffer segment described by a single page table element.

5.2.1 Unrestricted page tables

An unrestricted page table shall be contiguous within Serial Bus address space and shall be accessible to block read requests with a *data_length* less than or equal to *data_size* * 8 bytes. The format of elements in an unrestricted page table is shown by Figure .

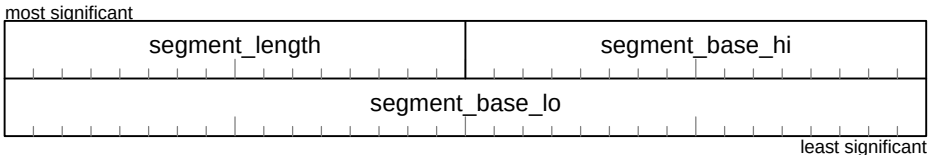

Figure 34 – Page table element (unrestricted page table)

The *segment_length* field shall contain the length, in bytes, of the portion of the data buffer (segment) described by the page table element. The value of *segment_length* shall be nonzero.

NOTE – A zero value in the same position as the *segment_length* field differentiates a node selector from a page table entry (see 5.2.3).

The *segment_base_hi* and *segment_base_lo* fields together shall specify the base address of the segment within the node's 48-bit system memory address range.

The 64-bit system memory address used to address the data is formed by the concatenation of the 16-bit *node_ID* field from the previous node selector or, if there is no previous node selector in the page table, the *node_ID* field from the *data_descriptor* field in the ORB, *segment_base_hi* and *segment_base_lo*.

5.2.2 Normalized page tables

A normalized page table shall be contiguous within Serial Bus address space and shall be accessible to Serial Bus block read transactions with a *data_length* less than or equal to the smaller of *data_size* * 8 bytes or $2^{page\_size + 8}$ bytes so long as they do not cross Serial Bus address boundaries that occur every $2^{page\_size + 8}$ bytes.

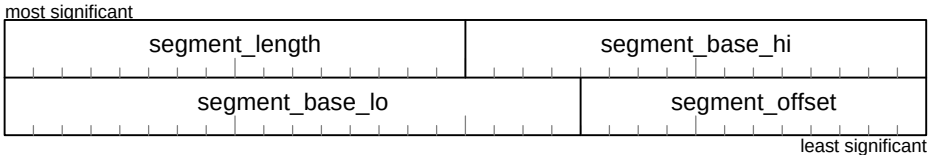

Figure 35 – Page table element (when *page_size* equals four)

NOTE – In the figure above, the field widths of *segment_base_lo* and *segment_offset*, 20 and 12 bits, respectively, are chosen only for the purposes of illustration. The size of *segment_base_lo* and *segment_offset* vary according to *page_size*. The field width, in bits, of *segment_offset* shall be *page_size* + *8*. In the example shown above, the page size is assumed to be 4096 bytes.

The *segment_length* field shall contain the length, in bytes, of the portion of the data buffer (segment) described by the page table element. The value of *segment_length* shall be nonzero and less than or equal to $2^{page\_size + 8}$.

NOTE – A zero value in the same position as the *segment_length* field differentiates a node selector from a page table entry (see 5.2.3).

The *segment_base_hi* and *segment_base_lo* fields together shall specify the base address of the segment within the node's 48-bit system memory address range.

The *segment_offset* field shall contain the starting address for data transfer within the segment.

The 64-bit system memory address used to address the data is formed by the concatenation of the 16-bit *node_ID* field from the previous node selector or, if there is no previous node selector in the page table, the *node_ID* field from the *data_descriptor* field in the ORB, *segment_base_hi*, *segment_base_lo* and *segment_offset*.

In all page table elements, the sum of *segment_length* and *segment_offset* shall be less than or equal to $2^{page\_size + 8}$.

In addition to the preceding requirements, the values of *segment_length* and *segment_offset* are constrained by their position within the page table. These additional restrictions are summarized below.

Element Position	Total page table elements		
	1	**2**	***n* (where *n* >= 3)**
First	No additional restrictions	*segment_length* = $2^{page\_size + 8}$ - *segment_offset*	
Middle	—		segment_offset = 0 *segment_length* = $2^{page\_size + 8}$
Last	—	*segment_offset* = 0	

5.2.3 Node selectors

A node selector is an 8-byte entry in a page table that identifies the node referenced by subsequent page table entries. A node selector applies to all subsequent page table entries until another node selector or the end of the page table is encountered. Node selectors permit a data buffer to be located in a different node than the page table; they also permit a data buffer to be distributed among more than one node. The format of a node selector is shown by Figure 35a.

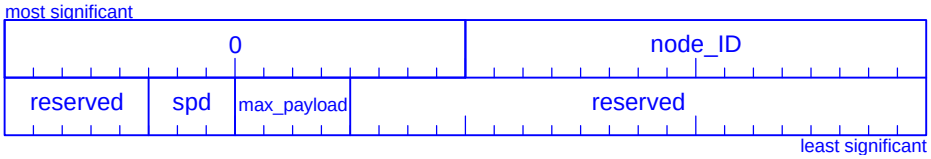

Figure 35a – Node selector

The most significant 16 bits of a node selector shall be zero.

The *node_ID* field shall identify the Serial Bus node to which to which subsequent page table entries pertain; it shall contain either a local node ID, as specified by IEEE 1394, or a node handle supplied by a target, as specified by this standard.

The *spd* and *max_payload* fields specify the speed and maximum data payload that shall be used by the target in request subactions addressed o the node identified by *node_ID*. The encoding of these fields is the same as the identically named fields in the normal command block ORB (see 5.1.2.1).

Target support for node selectors is optional and is indicated by the Unit_Characteristics entry in configuration ROM (see 7.6.8).

7.6.8 Unit_Characteristics entry

The Unit_Characteristics entry is an immediate entry in the unit directory which specifies characteristics of the target implementation. Figure 59 shows the format of this entry.

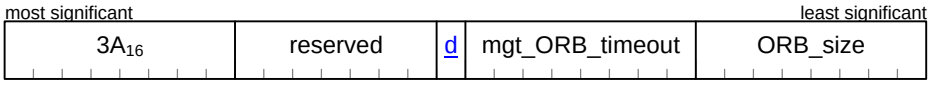

Figure 59 – Unit_Characteristics entry format

$3A_{16}$ is the concatenation of *key_type* and *key_value* for the Unit_Characteristics entry.

When the *distributed_data* bit (abbreviated as *d* in the figure above) is one, the target supports node selectors within page tables (see 5.2.3). This permits the initiator to distribute the data buffer among one

or more nodes independent of each other or the node that contains the page table. Otherwise, when *distributed_data* is zero, there is no target support for node selectors and the page table and data buffer shall be located within the same node.

The *mgt_ORB_timeout* field shall specify, in units of 500 milliseconds, the maximum time an initiator shall allow for a target to store a status block in response to a management ORB. The time-out commences when the initiator receives either *ack_complete* or *resp_complete* from the target in response to the block write of the management ORB address to the MANAGEMENT_AGENT register.

The *ORB_size* field shall specify, in quadlets, the fetch size used by the target to obtain ORBs from initiator memory. The initiator shall allocate, on a quadlet aligned boundary, at least this much memory for each ORB signaled to the target.