To: SBP-2 Working Group

From: John Nels Fuller

Date: 04/09/97

Re: Enhanced Page Tables

The page table mechanism in SBP-2 was designed and optimized for disk transfers. However, it is overly restrictive for other classes of devices that may want to use SBP-2. Additionally, it is not clear in the current draft the meaning of the page table when the *page_table_present* bit is one and *page_size* is zero. I would like to use this combination to indicate that the page table is an unrestricted scatter/gather list.

I propose that support of the unrestricted scatter/gather list be optional and indicated by a new bit in the Logical_Unit_Characteristics entry of the unit directory or logical unit directory in the configuration ROM of the device.

I propose the following changes to the document:

In section 5.1.2.1 add the following paragraph just before the paragraph beginning "If *page_table_present* is zero, the *data_size* field…"

If *page_table_present* is one and *page_size* is zero then the page table represents an unrestricted scatter/gather list with no inherent page size.

In section 5.2:

The data buffer associated with an ORB is specified 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. In this case, *data_descriptor* contains the 64-bit address of the data buffer and *data_size* specifies its length, in bytes.

In the other case, when *page_table_present* is equal to one, the data buffer is composed of segments that are discontiguous within Serial Bus address space and it is necessary to use a page table to describe the segments that form the data buffer. The page table is a variable-length array of elements whose format is ~~shown below~~determined by *page_size*. When *page_size* is non-zero the page table uses the normalized format. When *page_size* is zero the page table uses the unrestricted scatter/gather list format. Each element describes one segment that is contiguous within Serial Bus address space. Page table elements 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 nonzero, 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.

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.

5.2.1   Normalized page tables

The 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 2^{page_size+8} bytes so long as a block read transaction does not cross Serial Bus address boundaries that occur every 2^{page_size+8} bytes.

most significant

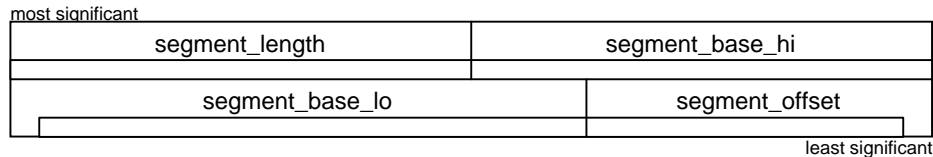| segment_length	segment_base_hi
segment_base_lo	segment_offset

least significant

Figure 1 – 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 specify the length, in bytes, of the portion of the data buffer described by the page table element. The value of *segment_length* shall be less than or equal to 2^{page_size+8}.

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 specify 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 *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.

	Total page table elements		
Element	**1**	**2**	**n (where n >= 3)**
0	No additional restrictions	*segment_length* = 2^{page_size+8} - *segment_offset*	
1 – *n*-2	—	*segment_offset* = 0	*segment_offset* = 0 *segment_length* = 2^{page_size+8}
n-1	—	—	*segment_offset* = 0

The presence of a page table is indicated by the value of *page_table_present* in the ORB. When *page_table_present* is nonzero, 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.

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. The 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 2^{page_size+8} bytes so long as a block read transaction does not cross Serial Bus address boundaries that occur every 2^{page_size+8} bytes.

5.2.2 Unrestricted scatter/gather lists

The 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 *data_size**8 bytes so long as a block read transaction does not cross Serial Bus address boundaries that occur every 512 bytes.
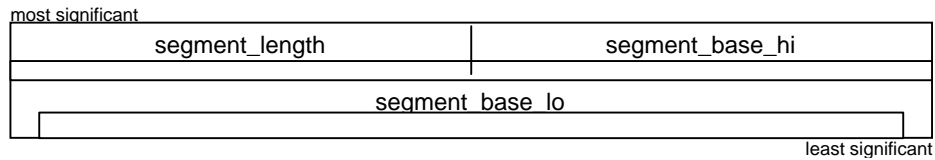
most significant

segment_length	segment_base_hi
segment_base_lo	

least significant

Figure 32 – Page table element (when *page_size* equals zero)

The *segment_length* field shall specify the length, in bytes, of the portion of the data buffer described by the page table element. When the *segment_length* field contains a value of zero 65536 bytes are specified.

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 *data_descriptor* field in the ORB, *segment_base_hi* and *segment_base_lo*.

In section 7.3.7:

The Logical_Unit_Characteristics entry is an immediate entry that, when present in the unit directory, specifies characteristics of the target implementation. Figure 2 shows the format of this entry.
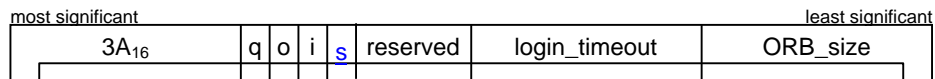
most significant | | | | | | | least significant

$3A_{16}$	q	o	i	s	reserved	login_timeout	ORB_size

Figure 2 – Logical_Unit_Characteristics entry format

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

● Page 3

The *q* bit shall specify the task management (queuing) model implemented by the target. If *q* is zero, the target implements the basic task management model defined by this standard in **Error! Reference source not found.**. When *q* is one, the task management model is dependent upon the command set specified by the Command_Set_Spec_ID and Command_Set_Version entries.

The *s* bit shall specify that the logical unit's support of unrestricted scatter/gather lists. When the *s* bit is one the logical unit supports unrestricted scatter/gather lists. When the *s* bit is zero the initiator shall not issue ORBs with both *page_table_present* set to one and *page_size* set to zero.

The *ordered* bit (abbreviated as *o* in the figure above) specifies the manner in which the target executes tasks signaled to the normal command block agent. If the target executes and reports completion status without any ordering constraints, the *ordered* bit shall be zero. Otherwise, if the target both executes all tasks in order and reports their completion status in the same order, the *ordered* bit shall be one.