Table Ordering Conventions

Bob Nixon

bob.nixon@emulex.com

1 Issue

The subclause that describes the convention for bit and byte ordering has led some to believe it constrains storage of data in memory or storage of data in devices or transmission of data on communication media. It is my belief that the only valid purpose for this subclause is to define the conventions for establishing an abstract ordering (e.g., MSB and LSB, first byte and last byte) of the data presented in a data structure table, since ordering of data storage and transmission may vary from standard to standard or even for different applications within a standard. If needed, other parts of the normative text of a standard should map this abstract ordering to transmission ordering or storage ordering.

2 Proposal

A broad restatement of the subclause that describes the convention for bit and byte ordering is proposed below.

3 Instructions to editor

3.1 Conventions

This proposal references 05-085r4 for numbers of clauses, subclauses, tables, figures, etc. Deletions are denoted by ~~red strikeout text~~. Additions are denoted by blue text.

3.2 Changes to 05-085r4

Modify body of subclause 3.7:

In this standard, data structures may be defined by a table. A tabular definition of a data structure defines a complete abstract ordering of elements (i.e., bits, bytes, fields, and dwords) within the structure. The abstract ordering of elements within a tabular definition does not in itself constrain the order of storage or transmission of the data structure, but in combination with other normative text in this standard, may constrain the order of storage or transmission of the structure.

In a tabular definition of a data structure, any structure element that is presented in a row above another structure element in a lower row is more significant than the lower structure element, and any data structure element presented to the left of another structure element in the same row is more significant than the structure element to the right.

If the bits within a table are numbered (see table 3), the least significant bit (LSB) is numbered 0 and each more significant bit has the next greater number than the immediately less significant bit. If the bytes or characters within a table are numbered (see table 4), the most significant byte or character is numbered 0 and each less significant byte or character has the next greater number than the immediately more significant byte.

In a field in a table consisting of more than one bit that contains a single value (e.g., a number), the least significant bit (LSB) is shown on the right and the most significant bit (MSB) is shown on the left (e.g., in a byte, bit 7 is the MSB and is shown on the left, bit 0 is the LSB and is shown on the right). The MSB and LSB are not labeled if the field consists of 8 or fewer bits. The MSB and LSB are labeled if the field consists of more than 8 bits and has no internal structure defined.

Editors Note 1 - xxx: Do the arabic numbers in the paragraph just above need fixing for compliance with 6.4.2.1 (e.g., change “8” to “eight”?

~~In a field in a table consisting of more than one byte that contains a single value (e.g., a number), the byte containing the MSB is stored at the lowest address and the byte containing the LSB is stored at the highest address (i.e., big-endian byte ordering). The MSB and LSB are labeled.~~

In a field in a table consisting of more than one byte that contains multiple fields each with their own values (e.g., a descriptor), there is no MSB and LSB of the field itself and thus there are no MSB and LSB labels. Each individual field has an MSB and LSB, but they are not labeled.

In a field containing a text string (e.g., ASCII or UTF-8), only ~~the MSB label is~~ the MSB of the first character and ~~the LSB label is~~ the LSB of the last character are labeled.

Multiple byte fields are represented with only two rows, with the non-sequentially increasing byte number ~~indicating~~ denoting the presence of additional bytes.

A data dword consists of 32 bits. Table 3 shows a data dword containing a single value, where the MSB is on the left in bit 31 and the LSB is on the right in bit 0. In this example, the MSB and LSB are labeled.

Table 3 — Data dword containing a value

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSB															Value																LSB

Table 4 shows a data dword containing four one-byte fields, where byte 0 (the first byte) is on the left and byte 3 (the fourth byte) is on the right. Each byte has an MSB on the left and an LSB on the right. In this example, the MSB and LSB of each byte are not labeled because each field (i.e., byte) consists of 8 or fewer bits.

Table 4 — Data dword containing four one-byte fields

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
~~MSB~~			Byte 0 (First byte)				~~LSB~~	~~MSB~~			Byte 1 (Second byte)				~~LSB~~	~~MSB~~			Byte 2 (Third byte)				~~LSB~~	~~MSB~~			Byte 3 (Fourth byte)				~~LSB~~