Foothill College
CS 10 Computer Architecture and Organization

Topic Lecture Notes – Byte Ordering

This is an auxiliary discussion to our text on Big Endian vs Little Endian. **Endianness** refers to the order of the bytes, comprising a digital word, in computer memory.

Byte Ordering

The ordering of bytes within a **multi-byte** data item defines the *endian-ness* of the architecture.

In BIG-ENDIAN systems the most significant byte of a multi-byte data item always has the lowest address, while the least significant byte has the highest address.

In LITTLE-ENDIAN systems, the least significant byte of a multi-byte data item always has the lowest address, while the most significant byte has the highest address.

In the following example, table cells represent bytes, and the cell numbers indicate the address of that byte in main memory. Note: by convention we draw the bytes within a memory word left-to-right for big-endian systems, and right-to-left for little-endian systems.

Big-Endian

Word Address				
0	0	1	2	3
4	4	5	6	7
8	8	9	10	11
12	12	13	14	15

MSB ——————-> LSB

LSB

Little-Endian

Word Address				
0	3	2	1	0
4	7	6	5	4
8	11	10	9	8
12	15	14	13	12

MSB ——————>

Note: an N-character ASCII string value is not treated as one large multi-byte value, but rather as N byte values, i.e. the first character of the string always has the lowest address, the last character has the highest address. This is true for both big-endian and little-endian. An N-character Unicode string would be treated as N two-byte value and each two-byte value would require suitable byte-ordering.

Example: Show the contents of memory at word address 24 if that word holds the number given by 122E 5F01H in both the big-endian and the little-endian schemes.

Big Endian

	MSB ----------> LSB			
	24	25	26	27
Word 24	12	2E	5F	01

Little Endian

	MSB ----------> LSB			
	27	26	25	24
Word 24	12	2E	5F	01

Example: Show the contents of main memory from word address 24 if those words hold the text JIM SMITH.

Big Endian

	+0	+1	+2	+3
Word 24	J	I	M	
Word 28	S	M	I	T
Word 32	H	?	?	?

Little Endian

	+3	+2	+1	+0
Word 24		M	I	J
Word 28	T	I	M	S
Word 32	?	?	?	H

The bytes labelled with ? are unknown. They could hold important data, or they could be don't care bytes – the interpretation is left up to the programmer.

If interested to see more endianness examples, refer to lecture notes for week 2 topic: Big Endian vs. Little Endian.