Danilo Barros Mendes

CSE 2120 – Spring 2016 Assignment # 3 Due: by start of class on Thursday, March 31
(submit in Canvas or on paper)

1. Assume you have a byte-addressable machine that uses 32-bit integers and you are storing the hex value **`0x7F3D4B2A`** starting at address 0000.

a. Show how this is stored on a Big Endian machine.

b. Show how this is stored on a Little Endian machine.

Address (byte)	0000	0001	0002	0003
a. Big Endian				
b. Little Endian				

2. Assume that the same computer is storing the following ASCII characters "**`EAST`**" starting at address 0000. Hint: You will have to look up the ASCII characters to get their hex values.

a. Show how this is stored on a Big Endian machine.

b. Show how this is stored on a Little Endian machine.

Address (byte)	0000	0001	0002	0003
a. Big Endian				
b. Little Endian				

3. Suppose a computer using direct mapped cache has $\mathbf{2^{24}}$ bytes of byte-addressable main memory, and a cache of **64** blocks, where each block of memory (and each block of cache) contains **16** bytes.

a. How many blocks of main memory are there?

b. What is the format of a memory address as seen by the cache, i.e., what are the sizes of the **tag**, **block**, and **offset** fields?

c. Which cache block will the memory address **`0x0C4A36`** map into?

4. Assume we have an empty cache that can hold no more than 10 variables and that the following 10 variables are stored into and accessed from the cache in the following order:

x9, x4, x2, x7, x3, x1, x6, x8, x2, x5, x0, x1, x8, x6, x4, x5, x7, x2, x9, x3

If a new variable must be stored in the cache, one of these 10 must be removed.

a. Which variable will be the one to remove if the cache uses a **First-In, First-out** replacement policy?

b. Which variable will be the one to remove if the cache uses a **Least Recently Used** replacement policy?

5. Given a system with a hit rate of **80%**, a cache access time of **20ns** and a memory access time of **100ns**.

Calculate the Effective Access Time (EAT) for this cache.

Danilo Barros Mendes

1)

_0000	_0001	_0002	_0003
7F	3D	4B	2A
2A	4B	3D	7F

2)

_0000	_0001	_0002	_0003
45	41	53	54
54	53	41	45

Dont change for little, because they are individual variables as ASCII
45 41 53 54

3)

a) 2^24 / 16 = 2^20 blocks

b) Offset : 4 bits
Blocks : 6 bits
Tag : 14 bits 2^24 -> tag = 24 - offset - blocks

c) 0x0C4A36 -> 805430 mod 16 = 6_{10} or 0x6

0C4A36 -> [0000 1100 0100 10][10 0011] [0110] -> separate counting bits of offset/block/tag -> 10 0011 = 35 dec / 23 hex

4)

a) x9
b) x0

5) EAT = (0.8*20)+(0.2*100) = 36ns