**Homework 5**

**Due: Thursday Nov 10th at 11:59 pm**

**Name: Jeremiah Webb**

**Review Questions**

1. (5 points) For a direct-mapped cache, a main memory address is viewed as consisting of three fields. List and define the three fields.

Tag – Portion of the cache address used as a unique identifier for a group of data.

Line Index – Indicates which portion of cache that will hold one block.

Block Offfset – Indicate which word to access within the block.

2. (5 points) For a set-associative cache, a main memory address is viewed as consisting of three fields. List and define the three fields.

Tag – Portion of cache set to indicate tag position in the set.

Set Index – Indicates which set to center to access its blocks.

Block Offset – Indicates which word to access within the block.

**Problems**

3. (5 points) Consider the following code:

**for** (i = 0; i < 20; i+ +)

**for** (j = 0; j < 10; j+ +)

a[i] = a[i]\* j

**a.** Give one example of the spatial locality in the code.

Accessing array a[], since the consequent array accesses will be accessing memory locations that are next to each other.

**b.** Give one example of the temporal locality in the code.

For the inner for loop, the array accesses element a[i], as the j is incremented with each iteration since a[i] is accessed multiple times within a short time frame.

4. (5 points) A **set-associative** cache consists of 64 lines, divided into four-line sets. Main memory contains 4K blocks of 128 words each. Calculate word offset bits, set index bits and tag bits for the given set-associative cache.

Main memory = 4KB = 22 x 210 x 27= 219 (19 address bits total)

# of sets = 64 cache lines (26) / 4 lines per set (22) = 16 sets = 24 (4 set index bits)

Block size = 128 words = 27 bytes (7 offset bits)

19-4-7 =8 tag bits

5. (10 points) Consider a machine with a byte addressable main memory of 216 bytes and block size of 8 bytes. Assume that a direct mapped cache consisting of 32 lines is used with this machine.

a.How is a 16-bit memory address divided into tag, line number, and byte number?

Main memory = 216 bytes (16 address bits total)

# of lines = 32 = 25 (5 line index bits)

Block size = 8 bytes = 23 bytes (3 offset bits)

16 - 5 - 3 = 8 tag bits

b.Into what line would bytes with each of the following addresses be stored?

0001 0001 0001 1011 = line 00011

1100 0011 0011 0100 = line 00110

1101 0000 0001 1101 = line 00011

1010 1010 1010 1010 = line 10101

6. (5 points each = 20 points) Consider a computer with the following characteristics: total of 1Mbyte of main memory; word size of 1 byte; block size of 16 bytes; and cache size of 64 Kbytes.

**a.** For the main memory addresses of F0010, 01234, and CABBE, give the corresponding tag, cache line address, and word offsets for a direct-mapped cache.

Main memory = 20 x 220 = 220 bytes (20 address bits total)

# of lines = 64KB cache (216)/16B block size (24) = 212 (12 line index bits)

Block size = 16 bytes = 24 (4 offset bits)

20 – 12 - 4 = 4 tag bits

F0010: 1111 0000 0000 0001 0000

Tag- 1111

Line- 0000 0000 0001

Offset- 0000

01234: 0000 0001 0010 0011 0100

Tag- 0000

Line- 0001 0010 0011

Offset- 0100

CABBE: 1100 1010 1011 1011 1110

Tag- 1100

Line- 1010 1011 1011

Offset- 1110

**b.** Give any two main memory addresses with different tags that map to the same cache slot for a direct-mapped cache.

C0003

* 1. Offset- 0011
  2. Line- 0000 0000 0000
  3. Tag- 1100

5000A

* 1. Offset- 1010
  2. Line- 0000 0000 0000
  3. Tag- 0101

**c.** For the main memory addresses of F0010 and CABBE, give the corresponding tag and offset values for a fully associative cache.

F0010: Tag- F001

Offset- 0

CABBE: Tag- CABB

Offset- E

**d.** For the main memory addresses of F0010 and CABBE, give the corresponding tag, cache set, and offset values for a two-way set-associative cache.

F0010: 1111 0000 0000 0001 0000

Tag- 1111 0

Set- 000 0000 0001

Offset- 0000

CABBE: 1100 1010 1011 1011 1110

Tag- 1100 1

Set- 010 1011 1011

Offset- 1110