CEG 4136 Tutorial 1

QUESTION #1

A direct mapped cache with WTWA (i.e. Write Through Write Allocate) is used to reduce the average memory access time on a computer. This cache holds 4K entries, 16-bit data, and ties into a 32-bit address bus with no byte or word select bits. Determine the contents of the cache and the memory below after the memory requests shown are sent to the cache. Immediately after powering up, the cache is initially empty and all valid bits are reset to indicate entries are invalid. Keep track of hits and misses so you can compute the cache miss rate when you are done. All values are in hex, "X" means undefined.

Cache Before

Block	Valid	Tag	Data
1	0	X	X
2	0	X	X
3	0	X	X
4	0	X	X
5	0	X	X

Cache After

Block	Valid	Tag	Data
1			
2			
3			
4			
5			

Memory Before

Address	Data
00100003	0ACE
00040003	1234
00040005	789A
00050004	0ABE

Memory After

Address	Data
00100003	
00040003	
00040005	
00050004	

Memory Requests

Address	Type	Data	Hit (y/n)
00040003	Read		
00050004	Write	0ECE	
00040005	Read		
00100003	Write	00EE	
00040005	Read		
00040005	Read		
00050004	Read		
00040003	Read		
00040005	Read		
00100003	Read		

Cache miss rate (include reads only) = _____%

QUESTION #2

Consider a machine with 2 processors that share the same memory. Multiply and Accumulate operation is performed:

$$global\_MAC = X[0]*Y[0]+ X[1]*Y[1]+\ldots X[N-1]*Y[N-1]$$

The MAC subroutine is implemented on both processors and it is shown below. Modify the program to make it suitable for execution in a four-processor machine.

If processor P1 starts executing MAC subroutine before the processor P0, will the final result be different. Why?

```
id = mypid ();  // Assign identification number:  id=0 for processor 0, and id=1 for processor 1
read_array(X, Y, N); //read arrays X and Y that have size N
if (id == 0) //initialize the MAC
{
        LOCK(global_MAC);
        global_MAC = 0;
        UNLOCK(global_MAC);
}
BARRIER(2);      //waits for all processors to get to this point in the program
local_MAC = 0;
for (i =id*N/2; i < (id+1)*N/2; i++)
        local_MAC += X[i]*Y[i];
LOCK(global_MAC);
global_MAC += local_MAC;
UNLOCK(global_MAC);
BARRIER(2);      //waits for all processors to get to this point in the program
END;
```

QUESTION #3

A four-way set associative cache is given the memory operations shown below. It contains four lines and has one word per block A LRU (Lease Recently Used) replacement policy is used on this cache. The cache is initially empty except for the two entries shown. The first column of the cache gets the first entry after the valid bits have been reset and then it fills left to right. In the blanks provided in the cache below, list the original address values only (i.e. 12 in blank, means tag address and data for memory location 12 is in cache).

Memory Read Operations

Address	Hit (y/n)
0	
1	
5	
19	
25	
9	
1	
13	
22	
4	
21	
22	
23	
5	
999	
401	

Final Cache Contents

Block	Element 1	Element2	Element 3	Element 4
0	4			
1	1			
2				
3				

Hit rate for all memory operations is _____%

QUESTION #4

Consider a machine with a byte addressable main memory of 2^{16} bytes and a 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?

(b) Into what line would bytes with each of the following addresses be stored?
0001 0001 0001 1011
1100 0011 0011 0100
1101 0000 0001 1101
1010 1010 1010 1010

(c) Suppose the byte address 0001 1010 0001 1010 is stored in the cache. What are the addresses of the other bytes stored along with it?

(d) How many total bytes of memory can be stored in the cache?

(e) Why the tag is also stored in the cache?