CS 3410 Homework 5

1. a) As we move down the memory hierarchy (to larger and larger caches), does the associativity of the caches typically increase or decrease? Why?

   b) Suppose we have run a program in the past and recorded its every memory access, and suppose we know it will behave in exactly the same way the next time we run it. Is LRU still the best page replacement policy we could use during page faults? If not, what is a policy that would likely be more effective?

   c) You have a direct-mapped cache with 32-bit addresses and four cache lines, each 256 bytes long. The engineers at NoobTech Inc. have managed to incorrectly connect the address bus: bits 0-7 (the least-significant bits) are wired correctly, but bits 8-31 are wired backwards so that bit 31 becomes bit 8, bit 30 becomes bit 7, etc. This cache is used in a system without virtual addressing.

   Although addressing still works correctly, using this system on real programs will be very slow. Why is this?

2. a) Suppose we have a 4-line cache with 16-byte lines, and consider the following sequence of memory addresses accesses. For each row of each column, indicate with an H or M whether that memory access results in a cache hit or a cache miss given the caching scheme for that column.

Address of Access	Fully-associative with LRU	2-way set-associative with LRU	Direct-mapped
0x100			
0x110			
0x120			
0x130			
0x140			
0x100			
0x110			
0x120			
0x130			

   b) The hit rates for the above example are a bit surprising in that we usually expect fully-associative caches to have the best hit rate. Suppose we have a cache that is 8 bytes wide, and an array that is 1.5 times as large as our cache, and consider the following code:

```
int[] A; //array
int n; //length of array
for(int k = 0; k < n; k++){
   print(A[k]);
}
for(int k = 0; k < n; k++){
   print(A[k]);
}
```

   Suppose the base address of A is such that it maps to the beginning of the first cache line in a direct-mapped scheme. How many hits and misses will a fully-associative cache with LRU have, as a function of n? How many hits and misses will a direct-mapped cache have, as a function of n?

3. a) Problem 5.9.1 on page 556 of Patterson and Hennessy *revised* 4th edition.

   b) Problem 5.9.2 on page 556 of Patterson and Hennessy revised 4th edition.

c) Problem 5.9.3 on page 556 of Patterson and Hennessy revised 4th edition.

4. Suppose we have an 8-bit address bus, and a 4-line direct-mapped cache with two-word cache lines. Consider the following code sequence:

```
LW $1, 0($zero)
LW $2, 8($zero)
LW $3, 4($zero)
SW $2, 4($zero)
SW $3, 20($zero)
SW $3, 16($zero)
```

At the beginning of execution, memory contains the following values:

Address	Value
0x00	1
0x04	1
0x08	2
0x0c	3
0x10	5
0x14	8

a) Assume that the cache uses write-through and no-write-allocate policies. Fill in the following tables describing the contents of the cache and memory just after the last line of the above code has executed.

V	Tag	Word 0	Word 1

Address	Value
0x00	
0x04	
0x08	
0x0c	
0x10	
0x14	

b) Now assume that the cache uses write-back and write-allocate policies. Fill in the following tables describing the contents of the cache and memory just after the last line of the above code has executed.

V	D	Tag	Word 0	Word 1

Address	Value
0x00	
0x04	
0x08	
0x0c	
0x10	
0x14	