The University of Alabama in Huntsville
ECE Department
CPE 431 01
Test 2
November 17, 2015

Name: _____

1. (1 point) A _____ is the smallest amount of information that is read or written on a disk.
2. (1 point) _____ is the time required to fetch a block into a level of the memory hierarchy from the lower level.
3. (1 point) A _____ cache is a cache structure in which a block can be placed in any location in the cache.-
4. (1 point) A _____ is a queue that holds data while the data is waiting to be written to memory.
5. (1 point) A _____ cache is a cache structure in which each memory location is mapped to exactly one location in the cache.

6. (10 points) Consider the code snippet shown below. Suppose that it is executed on a system with a 2-way set associative 16 KB data cache with 8 word blocks, 32-bit words, and an LRU replacement policy. Assume that int is word sized. Also assume that the word address of a is 0x0, that `i` and `x` are kept in registers, and that the cache is initially empty. How many data cache misses are there? How many hits are there? Assume that there is no byte offset.

```
int i;
int a[1024*1024];
int x = 0;

for (i = 0; i < 1024; i++)
  x += a[i] + a[1024*i]
```

7. (20 points) Here is a series of address references given as byte addresses: 118, 483, 2069, 321, 368, 1077, 1505, 812, 2832, 373, 1411, 511, 1463, 690, 4820, 1714, 1508, 1080. Assuming a two-way set associative-mapped cache with four-word blocks and a total size of 32 words that is initially empty and uses LRU, (a) label each reference in the list as a hit or a miss and (b) show the entire history of the cache, including tag and data.

Byte Address (Decimal)	Byte Addres (Hexadecimal)	
118	76	
483	1E3	
2069	815	
321	141	
368	170	
1505	5E1	
812	32C	
2832	B10	
373	175	
1411	583	
511	1FF	
1463	5B7	
690	2B2	
4820	12D4	
1714	6B2	
1508	5E4	
1080	438	

8. (15 points) a) Schedule the following code on a 2-issue pipeline in which one slot takes lw/sw instructions and the other slot takes R-type and branch instructions. You have forwarding from the MEM stage only and the branch completes in the ID stage.

```
Loop: lw    $s2, 0($s0)
      sub   $s4, $s2, $s3
      sw    $s4, 0($s0)
      lw    $s2, 4($s0)
      sub   $s4, $s2, $s3
      sw    $s4, 4($s0)
      lw    $s2, 8($s0)
      sub   $s4, $s2, $s3
      sw    $s4, 8($s0)
      addi  $s0, $s0, 12
      bne   $s0, $t3, Loop
```

Cycle	Issue Slot R type/Branch	Issue Slot lw/sw
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		

9. (15 points) Multilevel caching is an important technique to overcome the limited amount of space that a first level cache can provide while still maintaining its speed. Consider a processor with the following parameters.

Base CPI, no memory stalls	Processor speed	Main memory access time	First-level cache miss rate per instruction	Second-level cache, direct-mapped speed	Global miss rate with second-level cache, direct-mapped	Second-level cache, eight-way s et associative speed	Global miss rate with second-level cache, eight-way set associative
2.0	**2.5 GHz**	**125 ns**	**4.8 %**	**15 cycles**	**2.2 %**	**25 cycles**	**1.4 %**

Calculate the CPI for the processor given that the Memory accesses/instruction = 1.36 and that hits in the L1 cache incur no miss stalls for a) only a first-level cache, b) a second-level direct-mapped cache, and c) a second-level eight-way set-associative cache.

10. (15 points) Virtual memory uses a page table to track the mapping of virtual addresses to physical addresses. The following table is a stream of virtual addresses as seen on a system. Assume 8 KB pages, a four-entry fully associative TLB, and true LRU replacement. If pages must be brought in from disk, increment the next largest page number. Given the address stream, and the shown initial state of the TLB and page table, show the final state of the system. Also list for each reference if it is a hit in the page table, or a page fault.

12948, 49419, 46814. 13975, 40004, 12707, 52236

TLB

Valid	Tag	Physical Page Number
1	11	12
1	7	4
1	3	6
0	4	9

Page table

VPN	Valid	Physical page or in disk
0	1	5
1	0	Disk
2	0	Disk
3	1	6
4	1	9
5	1	11
6	0	Disk
7	1	4
8	0	Disk
9	0	Disk
10	1	3
11	1	12
12	0	Disk

11. (20 points) Using the code below, unroll the loop sot that two iterations are executed. Arrange the unrolled code to maximize performance. You may assume that the loop executes a multiple of two times. If the original loop executes 50 times, calculate the number of cycles for the original and for the unrolled, rearranged code. Assume full forwarding and one cycle delay for taken branches.

```
Loop:  lw    $s0, 0($t1)
       mul   $s0, $s0, $s2
       lw    $s4, 0($t2)
       add   $s0, $s0, $s4
       sw    $s0, 0($t2)
       addi  $t1, $t1, 4
       addi  $t2, $t2, 4
       bne   $t1, $zero, Loop
```