The University of Alabama in Huntsville
Electrical & Computer Engineering Department
CPE 431 01
Final Exam
December 6, 2007

Name: _____

1. (10 points) Here is a series of address references given as word addresses: 1, 4, 8, 5, 20, 17, 19, 56, 9, 11, 4, 43, 5, 6, 61. Assuming a two-way set-associative cache with four-word blocks and a total size of 32 words that is initially empty, (a) label each reference in the list as a hit or a miss and (b) show the final contents of the cache. Use an LRU replacement policy.

2. (10 points) What size messages would result in ATM outperforming Ethernet by a factor of ten, assuming the following latencies and bandwidths?

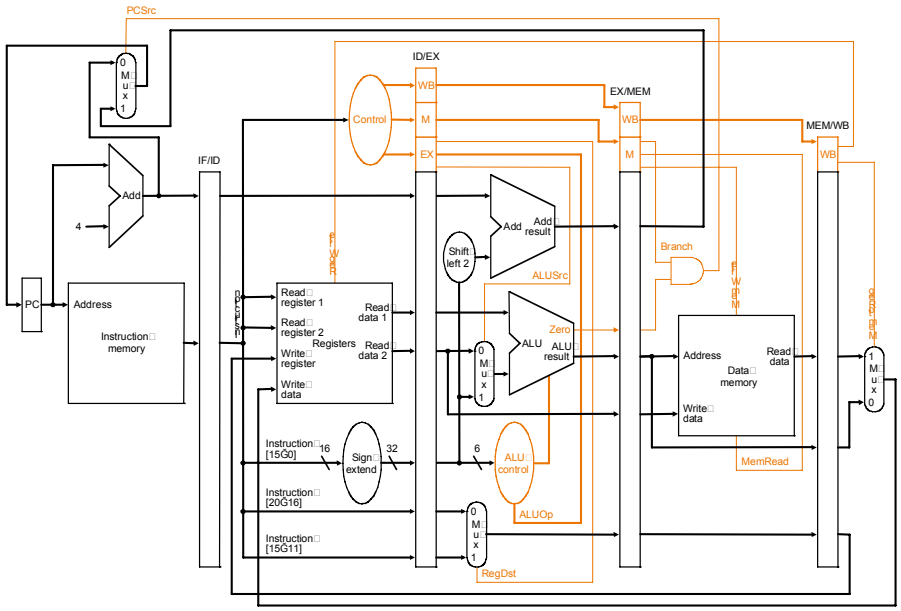
Characteristic	Ethernet	ATM
Bandwidth from node to network	1.125 MB/sec	10 MB/sec
Interconnect latency	15 μs	50 μs
HW latency to/from network	6 μs	6 μs
SW overhead sending to network	200 μs	207 μs
SW overhead receiving from network	241 μs	360 μs

3. (10 points) Suppose we have a system with the following characteristics:
    1. A memory and bus system supporting block access of 4 and 16 32-bit words.
    2. A 64-bit synchronous bus clocked at 200 MHz, with each 64-bit transfer taking 1 clock cycle, and 1 clock cycle required to send an address to memory.
    3. Two clock cycles needed between each bus operation. (Assume the bus is idle before an access.)
    4. A memory access time for the first four words of 150 ns; each additional set of four words can be read in 15 ns.

    Assume that the bus and memory systems described above are used to handle disks that transfer data at 30 MB/sec. If the I/O is allowed to consume 100% of the bus and memory bandwidth, what is the maximum number of simultaneous disk transfers that can be sustained for the two block sizes?

4. (10 points) Consider program P, which runs on a 2 GHz machine M in 10 seconds. An optimization is made to P, replacing all instances of multiplying a value by 4 (mult X, X, 4) with two instructions that set X to X + X twice (add X, X; add X, X) Call this new optimized program P'. The CPI of a multiply instruction is 5, and the CPI of an add is 1. After recompiling, the program now runs in 8.5 seconds on machine M. How many multiplies were replaced by the new compiler?

5. (15 points) Consider executing the following code on a pipelined datapath like the one shown, except that it has forwarding and in which branches complete in the MEM stage:




```
sort:    addi$sp, $sp, -20
         sw  $ra, 16($sp)
         sw  $s3, 12($sp)
         sw  $s2, 8($sp)
         sw  $s1, 4($sp)
         sw  $s0, 0($sp)
         add $s2, $a0, $zero
         add $s3, $a1, $zero
         add $s0, $zero, $zero
for1tst: slt $t0, $s0, $s3
         beq $t0, $zero, exit1
         addi$s1, $s0, -1
for2tst: slt $t0, $s1, $zero
         bne $t0, $zero, exit2
         add $t1, $s1, $s1
         add $t1, $t1, $t1
         add $t2, $s2, $t1
         lw  $t3, 0($t2)
         lw    $t4, 4($t2)
         slt   $t0, $t4, $t3
         beq   $t0, $zero, exit2
         add   $a0, $s2, $zero
         add   $a1, $s1, $zero
         jal   swap
         addi  $s1, $s1, -1
         j     for2tst
exit2:   addi  $s0, $s0, 1
         j     for1tst
exit1:   lw    $ra, 16($sp)
         lw    $s0, 0($sp)
         lw    $s1, 4($sp)
         lw    $s2, 8($sp)
         lw    $s3, 12($sp)
         addi  $sp, $sp, 20
         jr    $ra
```

If the `slt $t0` instruction at the `for1tst` label begins executing in cycle 1 and the `beq $t0, $zero, exit1` is not taken and the `bne $t0, $zero, exit2` is not taken and the `beq $t0, $zero, exit2` is taken, what are the values stored in the following fields of the ID/EX pipeline register in the 16th cycle? Assume that before the instructions are executed, the state of the machine was as follows:

The PC has the value 500_{10}, the address of `slt $t0` (at the label `for1tst`).

Every register has the initial value 20_{10} plus the register number.

Every memory word accessed as data has the initial value 2000_{10} plus the byte address of the word.

Fill in all of the fields, even if the current instruction in that state is not using them.

ID/EX.WB =
ID/EX.MEM =
ID/EX.EX =
ID/EX.PCInc =
ID/EX.ReadData1 =
ID/EX.ReadData2 =
ID/EX.SignExtend =
ID/EX.WriteRt =
ID/EX.WriteRd =

Cycle	IF	ID	EX	MEM	WB
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					

6. (5 points) A binary word with odd parity and no errors will have an odd number of 1s in it. Compute the parity bit for each of the following 8-bit words so that the resulting 9-bit word has odd parity.

    a. 01100111
    b. 01010101

7. (1 point) The alternative model to message passing for parallel processing is

_____

8. (1 point) The term _____ refers to a single program that runs on several processors simultaneously.

9. (1 point) A _____ is a synchronization scheme in which processors wait and do not proceed until every process is ready.

10. (1 point) _____ are networks of off-the-shelf, whole computers.

11. (1 point) A _____ protocol maintains consistency in the value of data between several processors.

12. (10 points) Represent 52419.5625 as a single precision floating point number.

13. (5 points) Consider a virtual memory system with the following properties:
    - 64-bit virtual byte address
    - 128-KB pages
    - 36-bit physical byte address

    What is the total size of the page table for each process on this machine, assuming that the valid, protection, dirty, and use bits take a total of 4 bits and that all the virtual pages are in use? (Assume that disk addresses are not stored in the page table.)

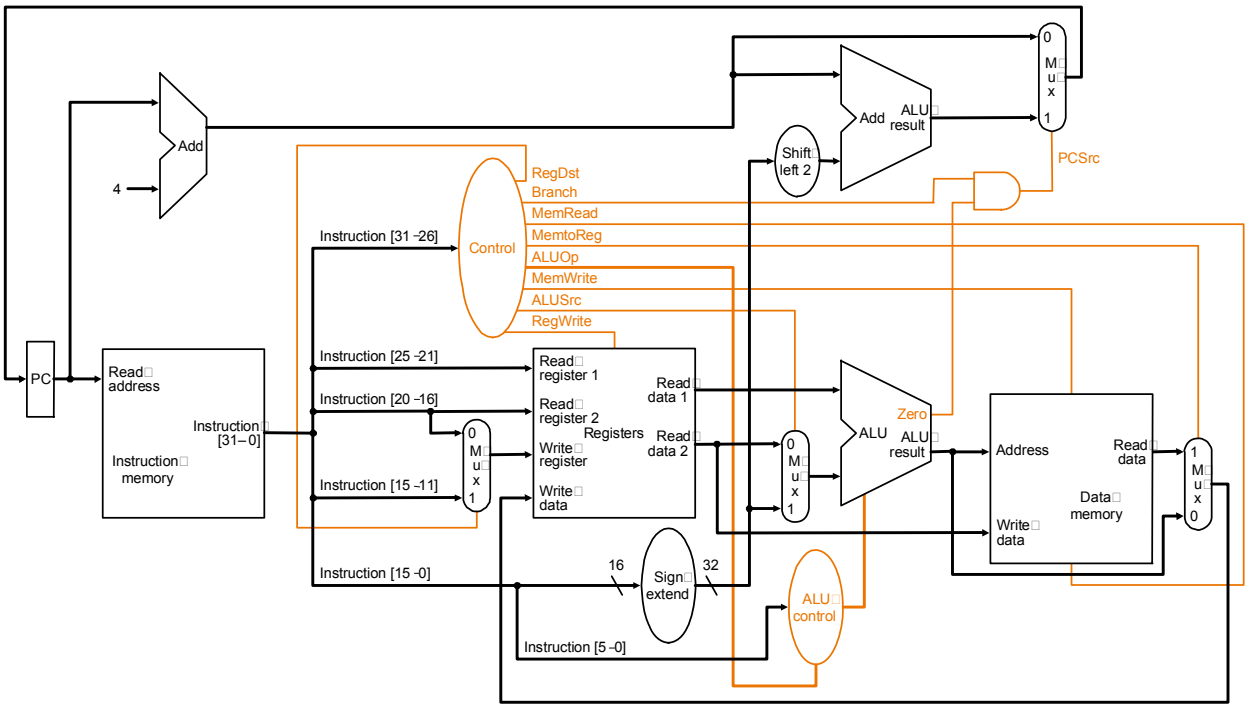
14. (10 points) Show the single MIPS instruction or minimal sequence of instructions for this C statement:

```
z[10] = x[10] + y[10];
```

    Assume that array `x` (an array of 32-bit integers) has a base address of $4{,}000{,}000_{10}$ which is stored in register `$t0,` that y. (an array of 32-bit integers) has a base address of $5{,}000{,}000_{10}$ which is stored in register `$t1,` and that z (an array of 32-bit integers) has a base address of $6{,}000{,}000_{10}$ which is stored in register `$t2.`

15. (10 points) Consider each of the following stuck-at-1 faults separately: RegDst = 1, ALUSrc = 1, MemtoReg = 1, RegWrite = 1. Which instructions, if any would still work?

	RegDst = 1	ALUSrc = 1	MemtoReg = 1	RegWrite= 1
R-type				
lw				
sw				
beq				



Instruction	RegDst	ALUSrc	Memto Reg	Reg Write	Mem Read	Mem Write	Branch	ALUOp1	ALUOp0
R-format	1	0	0	1	0	0	0	1	0
lw	0	1	1	1	1	0	0	0	0
sw	d	1	d	0	0	1	0	0	0
beq	d	0	d	0	0	0	1	0	1