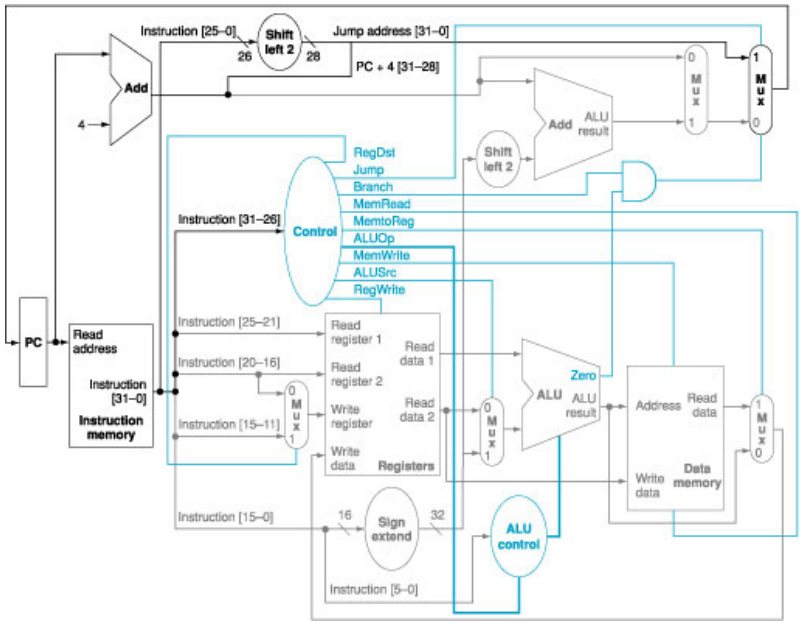
The University of Alabama in Huntsville
Electrical & Computer Engineering Department
CPE 431 01
Test 2
November 17, 2005

Name: _____

1. (15 points) In estimating the performance of the single-cycle implementation, we assumed that only the major functional units had any delay (i.e., the delay of the multiplexors, control unit, PC access, sign extension unit, and wires was considered to be negligible). Assume that we change the delays specified such that we use a different type of adder for simple addition:
   - ALU: 100 ps
   - adder for PC + 4: X ps
   - adder for branch address computation: Y ps

   Also, continue to assume that multiplexors, wires, and PC access have 0 delay but that the delay of the control unit is 100 ps and sign extension is 20 ps. The delay for memory units is 200 ps and the delay for a register file read or write is 50 ps.

   a. What would the cycle time be if X = 50 and Y = 75?
   b. What would the cycle time be if X = 30 and Y = 60?



2. (1 point) For _____ elements, their outputs depend only on the current inputs.

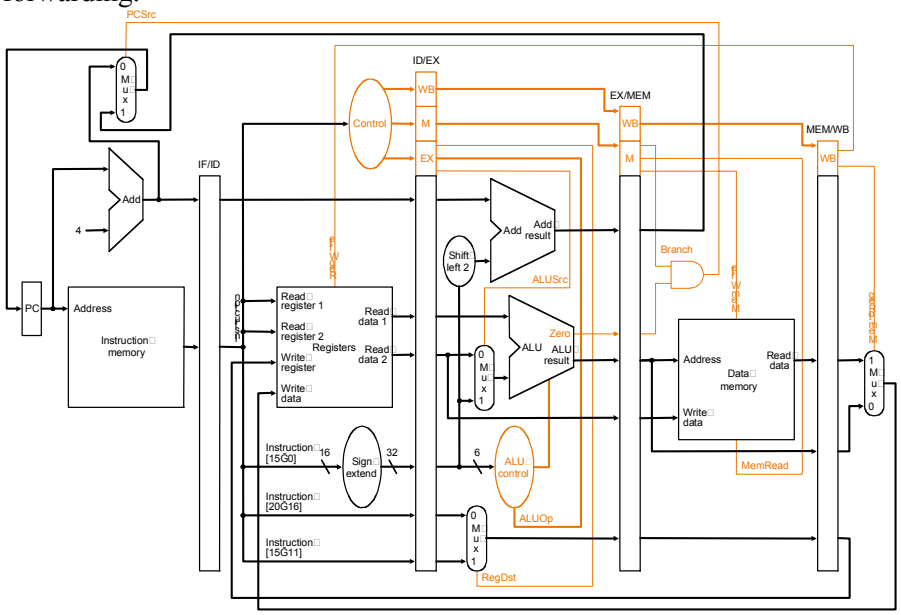
3. (1 point) The _____ register contains the address of the instruction in the program being executed.

4. (1 point) A _____ element is a functional unit used to operate on or hold data within a processor.

5. (1 point) _____ is a method of resolving a branch hazard that assumes a given outcome.

6. (1 point) A _____ cache structure is one in which a block can be placed in any location in the cache.

7. (20 points) Consider executing the following code on a pipelined datapath like the one shown except that it has forwarding.




```
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    $s0, 0($sp)
         lw    $s1, 4($sp)
         lw    $s2, 8($sp)
         lw    $s3, 12($sp)
         lw    $ra, 16($sp)
         addi  $sp, $sp, 20
         jr    $ra
```

If the `add $t1` instruction two instructions after the `for2tst` label begins executing in cycle 1 and the `beq $t0, $zero, exit2` is taken, what are the values stored in the following fields of the IF/ID pipeline register in the 14th cycle? Assume that before the instructions are executed, the state of the machine was as follows:

The PC has the value 200_{10}, the address of the `add $t1` instruction
Every register has the initial value 20_{10} plus the register number.
Every memory word accessed as data has the initial value 10000_{10} plus the byte address of the word.

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

IF/ID.PCInc =
IF/ID.Instruction =

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

8. (10 points) Consider a virtual memory system with the following properties:
   64-bit virtual byte address
   32 MB pages
   48-bit physical address
   What is the total size of the page table for each process on this processor, assuming that the valid, protection, dirty, and use bits take a total of 6 bits and that all the virtual pages are in use? (Assume that disk addresses are not stored in the page table.)

9. (15 points) (a) Identify all of the data dependencies in the following code. (b) How is each data dependency either handled or not handled by forwarding? **Draw a multiple clock cycle style diagram to support your answer.**

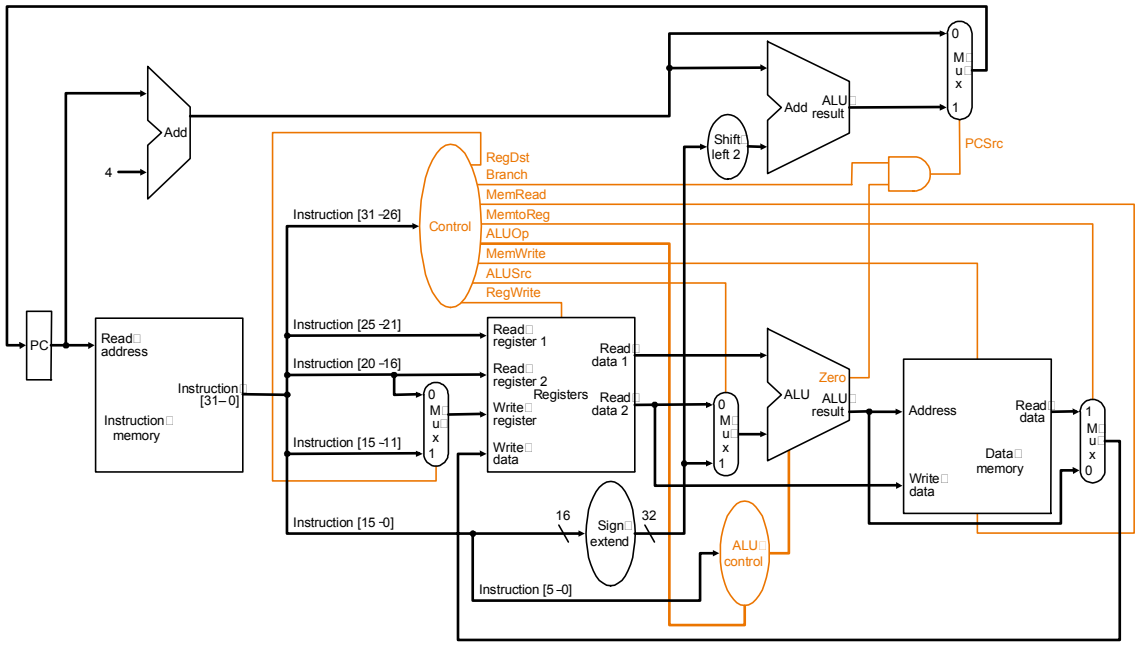
```
add   $5, $5, $4
lw    $5, 28($2)
add   $2, $4, $5
sw    $5, 100($2)
add   $3, $2, $4
```

10. (20 points) Here is a series of address references given as word addresses: 2, 3, 11, 16, 21, 13, 64, 48, 19, 11, 3, 22, 4, 27, 6, and 11. Assuming a two-way set associative cache with one-word blocks and a total size of 16 words that is initially empty, label each reference in the list as a hit or a miss and show the final contents of the cache.

11. (15 points) Add a variant of the `lw` instruction which sums the contents of two registers to obtain the address of the data and which uses the R format to the single-cycle datapath shown in the figure below. Add any necessary datapaths and control signals and show the necessary additions to the table of control signals given.

```
lw   $9, $6, $22
```
$9 ← MEM[$6+$22]

36	6	22	9	0	32



Instruction	RegDst	ALUSrc	Memto-Reg	Reg Write	Mem Read	Mem Write	Branch	ALU Op1	ALU Op0	
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	