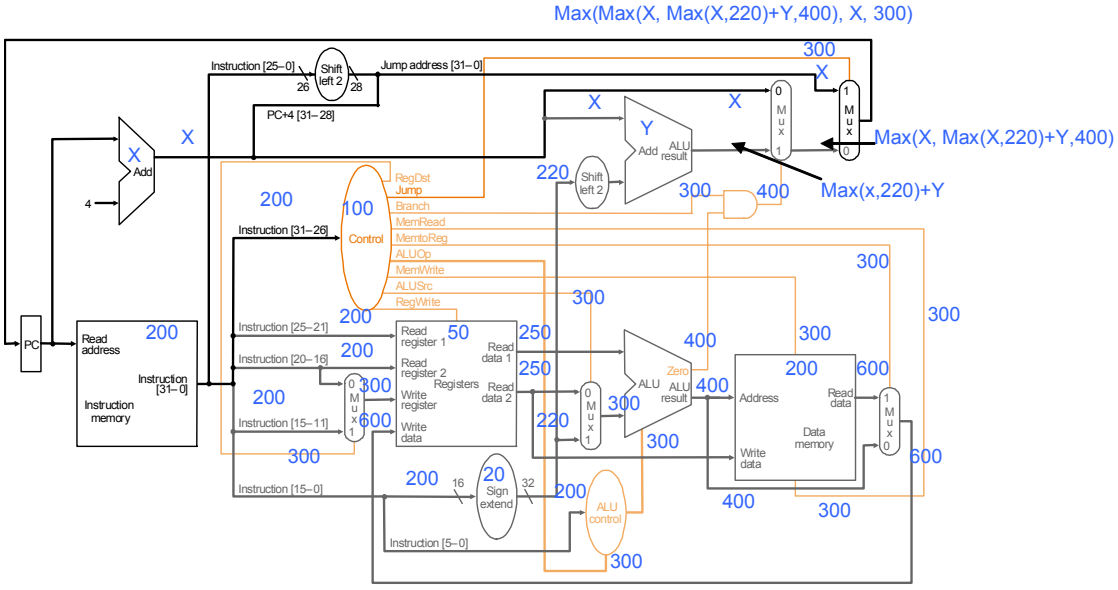
The University of Alabama in Huntsville
Electrical & Computer Engineering Department
CPE 431 01
Test 2 Solution
Fall 2005

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?

   Paths to consider are: (1) lw completion – 650 ps
   (2) PC determination - Max(Max(X, Max(x,220)+Y,400), X, 300)
   Taking the maximum of all paths yields
   Max(Max(Max(x, Max(x,220)+Y,400), X, 300), 650)

   a. X = 50, Y =75, CC = Max(Max(Max(50, Max(50,220)+75,400), 50, 300), 650)
   = Max(Max(Max(50, 295, 400), 50, 300), 650)
   = Max(Max(400, 50, 300), 650)
   = Max(400, 650) = 650
   b. X = 30, Y = 60, CC = Max(Max(Max(30, Max(30,220)+60,400), 30, 300), 650
   = Max(Max(Max(30, 280, 400), 30, 300, 650)
   = Max(Max(400, 30, 300), 650)
   = Max(400, 650) = 650



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

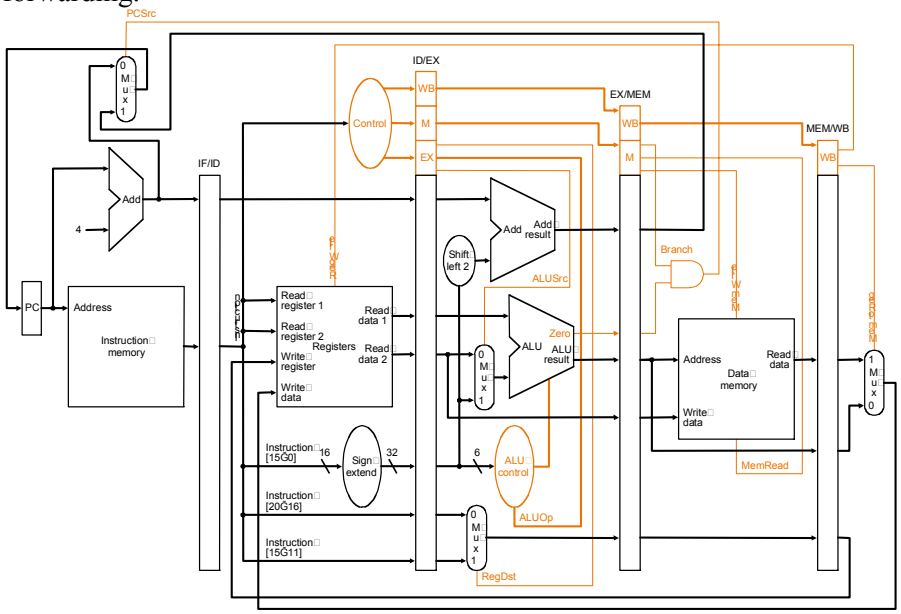
3. (1 point) The __program counter_ register contains the address of the instruction in the program being executed.

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

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

6. (1 point) A __fully associative_ 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.




```
144  sort:    addi  $sp, $sp, -20
148           sw    $ra, 16($sp)
152           sw    $s3, 12($sp)
156           sw    $s2, 8($sp)
160           sw    $s1, 4($sp)
164           sw    $s0, 0($sp)
168           add   $s2,$a0, $zero
172           add   $s3, $a1, $zero
176           add $s0, $zero, $zero
180  for1tst: slt   $t0, $s0, $s3
184           beq $t0, $zero, exit1
188           addi $s1, $s0, -1
192  for2tst: slt   $t0, $s1, $zero
196           bne $t0, $zero, exit2
200           add  $t1, $s1, $s1
204           add  $t1, $t1, $t1
208           add  $t2, $s2, $t1
212           lw   $t3, 0($t2)
216           lw    $t4, 4($t2)
220           slt   $t0, $t4, $t3
224           beq $t0, $zero, exit2
228           add   $a0, $s2, $zero
232           add   $a1, $s1, $zero
236           jal   swap
240           addi  $s1, $s1, -1
244           j     for2tst
248  exit2:   addi  $s0, $s0, 1
252           j     for1tst
256  exit1:   lw    $s0, 0($sp)
260           lw    $s1, 4($sp)
264           lw    $s2, 8($sp)
268           lw    $s3, 12($sp)
272           lw    $ra, 16($sp)
276           addi  $sp, $sp, 20
280           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 = 256_{10}
IF/ID.Instruction = op = 000010, address = 0101101 = $0800002D_{16}$

Cycle	IF	ID	EX	MEM	WB
1	add $t1				
2	add $t1	add $t1			
3	add $t2	add $t1	add $t1		
4	lw $t3	add $t2	add $t1	add $t1	
5	lw $t4	lw $t3	add $t2	add $t1	add $t1
6	slt $t0	lw $t4	lw $t3	add $t2	add $t1
7	beq $t0	slt $t0	lw $t4	lw $t3	add $t2
8	beq $t0	slt $t0	bubble	lw $t4	lw $t3
9	add $a0	beq $t0	slt $t0	bubble	lw $t4
10	add $a1	add $a0	beq $t0	slt $t0	bubble
11	jal swap	add $a1	add $a0	beq $t0	slt $t0
12	addi $s0	bubble	bubble	bubble	beq $t0
13	j for1tst	addi $s0	bubble	bubble	bubble
14	lw $s0	j for1tst	addi $s0	bubble	bubble

The instruction of interest is `j for1tst`.

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.)

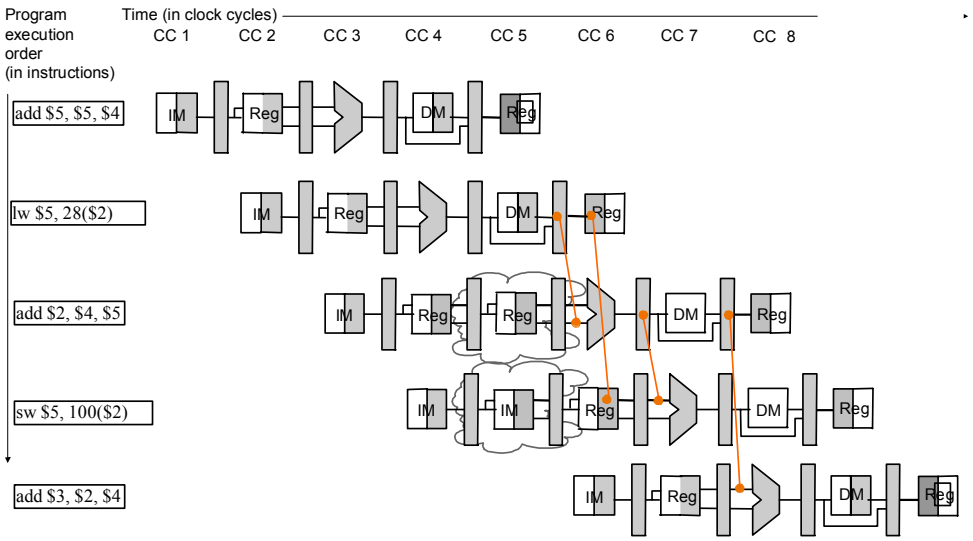
   The page offset is $\log_2(32MB) = 25$, therefore there are $2^{64}/2^{25}$ pages, or 2^{39} entries in the page table.

   Each entry consists of a physical page number of (48-25) 23 bits plus the valid, protection, dirty, and use bits (6) for 29 bits

   So, the total size is 2^{39}*29 bits

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.**

			Dependency	How Handled
a	add	$5, $5, $4	b-c	stall, then forward from MEM/WB
b	lw	$5, 28($2)	b-d	after stall, data is available in register file
c	add	$2, $4, $5	c-d	forward from EX/MEM
d	sw	$5, 100($2)	c-e	forward from MEM/WB
e	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.

$$16words * \frac{1block}{1word} * \frac{1set}{2blocks} = 8sets$$

Block Offset - 0 bits, Index - 3 bits

	index	
2	0000 010	m
3	0000 011	m
11	0001 011	m
16	0010 000	m
21	0010 101	m
13	0001 101	m
64	1000 000	m
48	0110 000	m
19	0010 011	m
11	0001 011	h
3	0000 011	m
22	0010 110	m
4	0000 100	m
27	0011 011	m
6	0000 110	m
11	0001 011	m

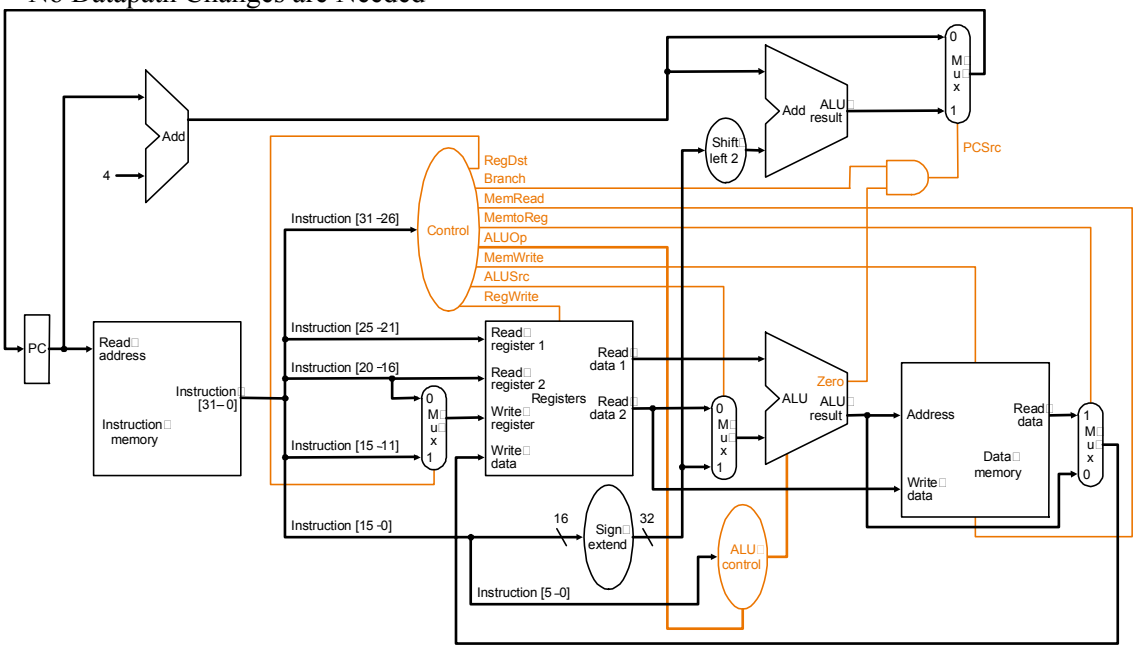
0	~~16~~, 48	64
1		
2	2	
3	~~3~~, ~~19~~, ~~3~~, 11	~~11~~, 27
4	4	
5	21	13
6	22	6
7		

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

No Datapath Changes are Needed



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
`lw` (variant)	1	0	0	1	0	0	0	0	0