The University of Alabama in Huntsville
Electrical & Computer Engineering Department
CPE 431 01
Test II Solution
November 15, 2001

Name: _____

1. (5 points) For a twelve-stage pipeline, how many cycles does it take to execute n instructions if the pipeline is empty when these instructions begin to execute? _12 + (n-1) * 1 = n + 11__

2. (1 point) A _____data hazard_____ occurs when an instruction depends on the results of a previous instruction still in the pipeline.

3. (15 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, 9, 17. Assuming a direct mapped 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.

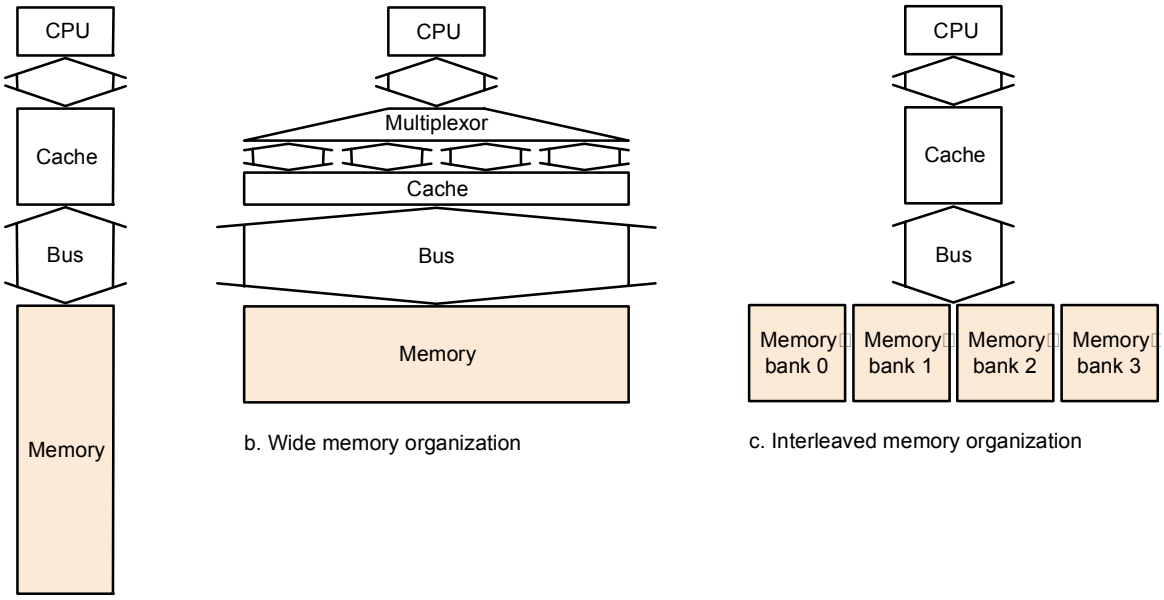
$$32words * \frac{1block}{4words} * \frac{1set}{1block} = 8sets$$

Block offset – 2 bits, Index – 3 bits

	index
1	0 000 01 m
4	0 001 00 m
8	0 010 00 m
5	0 001 01 h
20	0 101 00 m
17	0 100 01 m
19	0 100 11 h
56	1 110 00 m
9	0 010 01 h
11	0 010 11 h
4	0 001 00 h
43	1 010 11 m
5	0 001 01 h
6	0 001 10 h
9	0 010 01 m
17	0 100 01 h

0	0	1	2	3
1	4	5	6	7
2	~~8~~, ~~40~~, 8	~~9~~, ~~41~~, 9	~~10~~, ~~42~~, 10	~~11~~, ~~43~~, 11
3				
4	16	17	18	19
5	20	21	22	23
6	56	57	58	59
7				

4. (15 points) Consider a memory hierarchy using one of the three organizations for main memory shown below. Assume that the cache block size is 8 words, that the width of organization b of the figure is four words, and that the number of banks in organization c is four. If the main memory latency for a new access is 30 cycles and the transfer time is 3 cycles, what are the miss penalties for these organizations?



a. One-word-wide memory organization

b. Wide memory organization

c. Interleaved memory organization

a. 3 send address + 8 * (30 read + 3 transfer) = 3 + 8*(33) = 3 + 264 = 267 cycles
b. 3 send address + 2 * (30 read + 3 transfer) = 3 + 2*33 = 3 + 66 = 69 cycles
c. 3 send address + 2*30 read + 8*3 transfer = 3 + 60 + 24 = 87 cycles

5. (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? If forwarding occurs, state the source of the data.

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

Data dependencies:	Hazard?	Forward?
1. add $2, $5, $4 – lw $2, 28($2)	Yes	Yes, from EX/MEM
2. lw $2, 28($2) – add $4, $2, $5	Yes	stall, then from MEM/WB
3. lw $2, 28($2) – sw $4, 100($2)	Yes	not needed after stall
4. lw $2, 28($2) – add $3, $2, $4	No	
5. add $4, $2, $5 – sw $4. 100($2)	Yes	Yes, from EX/MEM
6. add $4, $2, $5 – add $3, $2, $4	Yes	Yes, from MEM/WB

6. (1 point) Pipelining improves performance by increasing _____ _____, as opposed to decreasing the execution time of an individual instruction.

7. (3 points).The three C's of cache memories are: _____conflict_____. _____compulsory_____, and _____capacity_____.

8. (10 points) Consider a virtual memory system with the following properties:
   34-bit virtual byte address
   8 Kbyte pages
   18-bit physical 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 6 bits and that all the virtual pages are in use? (Assume that disk addresses are not stored in the page table.)

   For 8 Kbyte pages, that means a page offset of 13 bits. 34 –13 is 21, so there are 2^{21} virtual pages. Each entry in the page table is 18 – 13 for a page number + 6 bits of bookkeeping.

   So, the size is

   $2^{21} * (5 + 6) = 2^{21} * 11$ bits

9. (20 points) Consider executing the following code on a pipelined datapath like the one in Chapter 6, which has no forwarding, which has hazard detection, in which branches complete in the MEM stage, and in which there is no branch prediction:

```
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 $s1, $s0, $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 addi instruction with the label sort begins executing in cycle 1 and the beq $t0, $zero, exit1 is taken, what are the values stored in the following fields of the ID/EX pipeline register in the 19^{th} cycle? Assume that before the instructions are executed, the state of the machine was as follows:

The PC has the value 300_{10}, the address of the addi instruction.
Every register has the initial value 20_{10} plus the register number.
Every memory word accessed as data has the initial value 4000_{10} plus the byte address of the word.

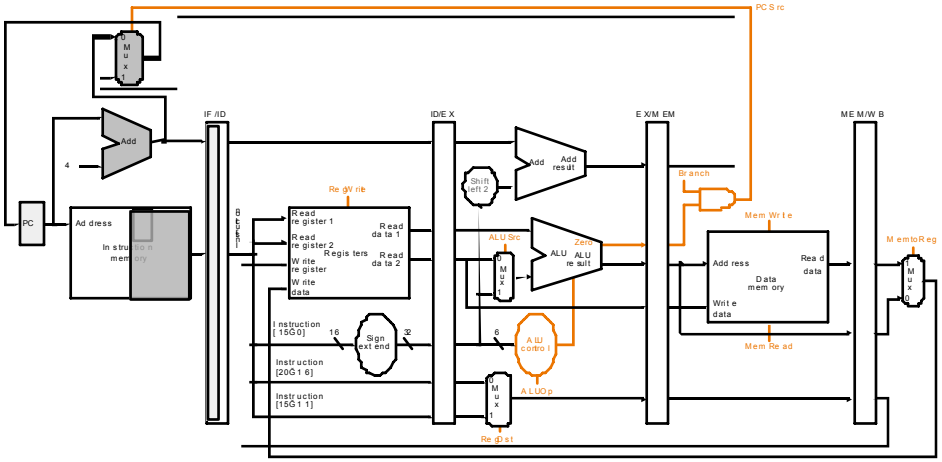
ID/EX.EX = d010
ID/EX.PCInc = 344
ID/EX.ReadRegister1 = 28
ID/EX.ReadRegister2 = 0
ID/EX.WriteRt = 0
ID/EX.WriteRd = 0

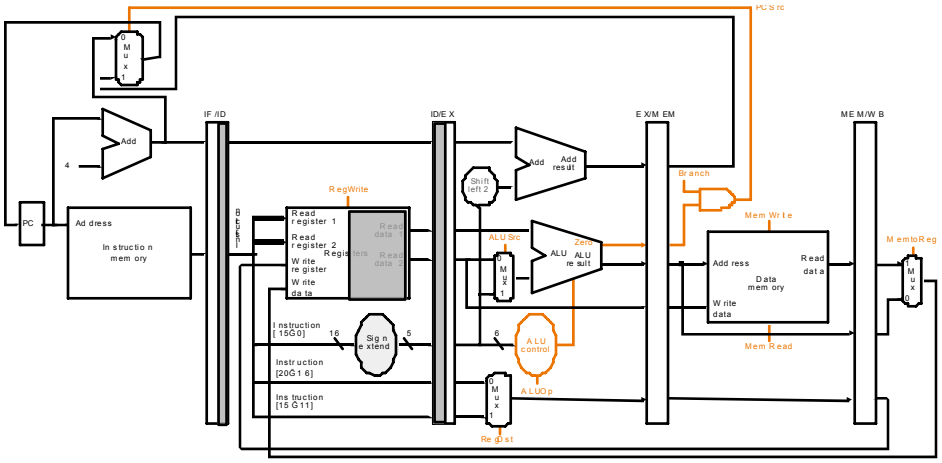
The instruction is beq $t0 which has the following fields:

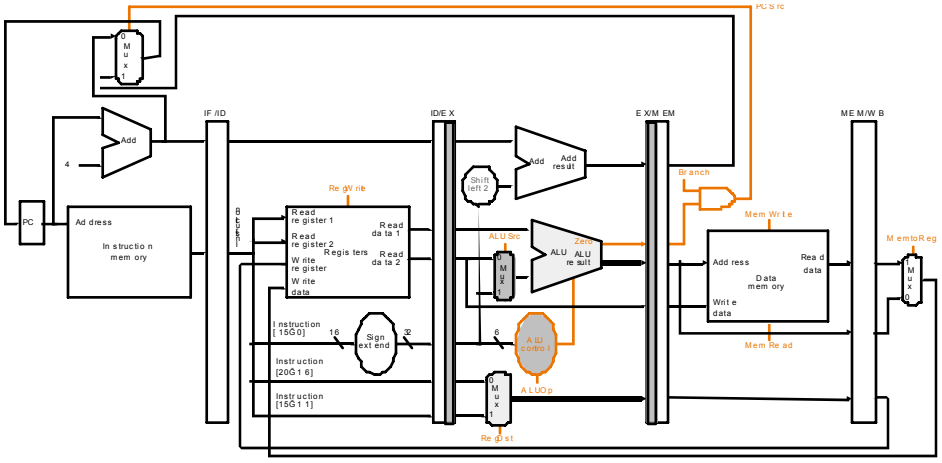
opcode	rs	rt	offset
4	8	0	17

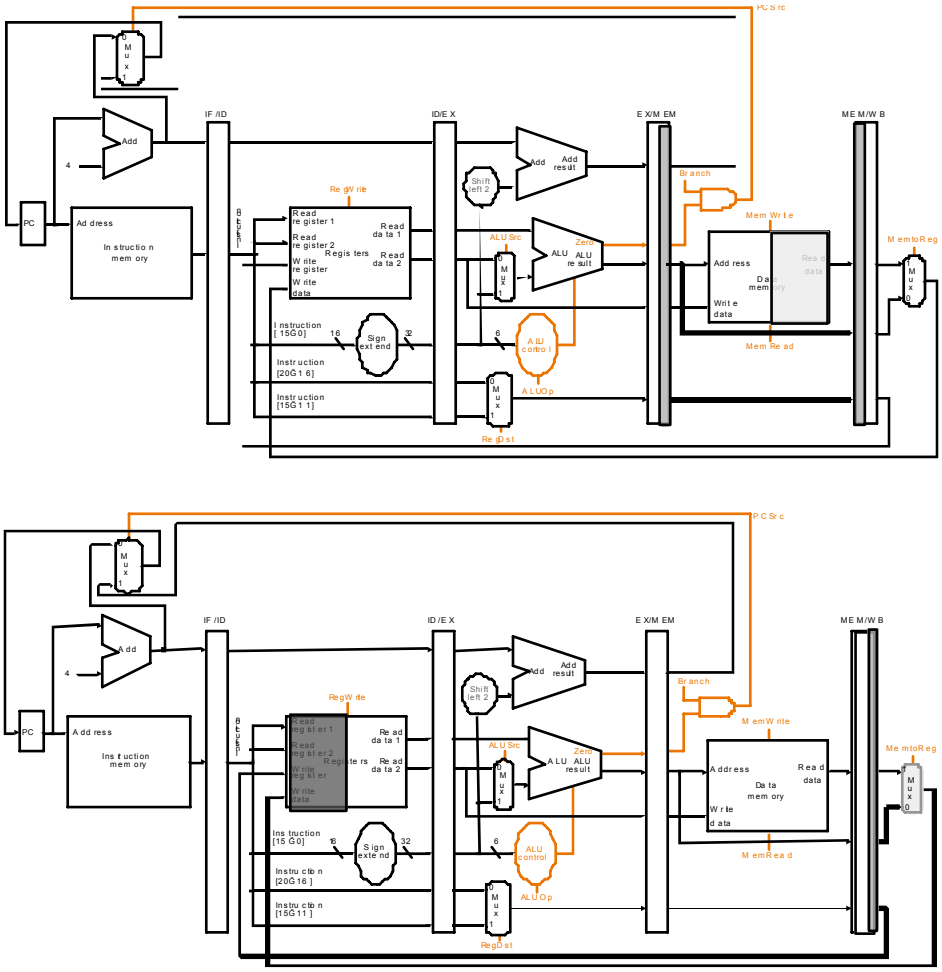
Cycle	IF	ID	EX	MEM	WB
1	addi $sp				
2	sw $ra,	addi $sp			
3	sw $s3	sw $ra	addi $sp		
4	sw $s3	sw $ra	bubble	addi $sp	
5	sw $s3	sw $ra	bubble	bubble	addi $sp
6	sw $s2	sw $s3	sw $ra,	bubble	bubble
7	sw $s1	sw $s2	sw $s3	sw $ra,	bubble
8	sw $s0	sw $s1	sw $s2	sw $s3	sw $ra,
9	add $s2	sw $s0	sw $s1	sw $s2	sw $s3
10	add $s3	add $s2	sw $s0	sw $s1	sw $s2
11	add $s0	add $s3	add $s2	sw $s0	sw $s1
12	slt $t0	add $s0	add $s3	add $s2	sw $s0
13	beq $t0	slt $t0	add $s0	add $s3	add $s2
14	beq $t0	slt $t0	bubble	add $s0	add $s3
15	beq $t0	slt $t0	bubble	bubble	add $s0
16	beq $t0	beq $t0	slt $t0	bubble	bubble
17	beq $t0	beq $t0	bubble	slt $t0	bubble
18	addi $s1	beq $t0	bubble	bubble	slt $t0
19	slt $s1	addi $s1	beq $t0	bubble	bubble

10. (10 points) Using the figures provided below, indicate which portions of the datapath are active (performing an operation used later by the instruction's execution) and which are inactive (performing an operation discarded during the instruction's execution) in each of the five stages of a `lw` instruction.









11. (5 points) Use the following code fragment:

```
loop: lw    $1, 0($2)
      addi  $1, $1, 4
      sw    $1, 0($2)
      addi  $2, $2, 4
      bne   $2, $3, loop
```

Consider the pipelined datapath of Chapter 6 with forwarding. Insert any necessary stalls, or reorder, if possible, to prevent stalling.

```
loop: lw    $1, 0($2)
      addi  $2, $2, 4
      addi  $1, $1, 4
      sw    $1, -4($2)
      bne   $2, $3. loop
```