The University of Alabama in Huntsville
Electrical & Computer Engineering Department
CPE 431 01
Test 2 Solution
Fall 2004

1. (5 points) Given the pipelined processor of Chapter 6 with forwarding, determine which instruction is being executed in each stage of the pipeline in cycle 8 of the following instruction sequence if it begins executing in cycle 1

```
1.    addi  $23, $29, 12
2.    sw    $2, 0($29)
3.    sw    $15, 4($29)
4.    sw    $16, 8($29)
5.    muli  $8, $5, 4
6.    add   $7, $4, $2
7.    lw    $15, 0($2)
8.    lw    $16, 4($2)
9.    sw    $17, 0($2)
10.   sw    $18, 4($2)
11.   lw    $2, 0($29)
12.   lw    $15, 4($29)
13.   lw    $16, 8($29)
14.   addi  $29, $29, 12
```

IF _____8_____ ID _____7_____ EX _____6_____ MEM _____5_____ WB _____4_____

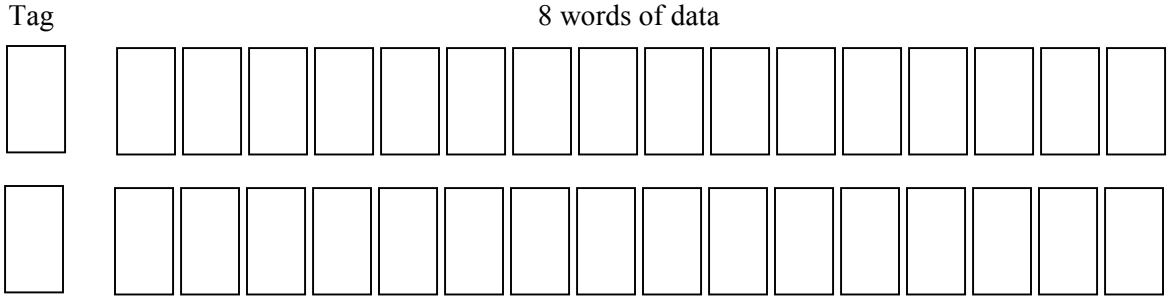
2. (15 points) You have been given 50 16K x 16-bit SRAMS to build an instruction cache for a processor with a 32-bit address. You do not have a byte offset. What is the largest size (i.e., the largest size of the data storage area in bytes) direct-mapped instruction cache that you can build with eight-word blocks? Show the breakdown on the address into its cache access components and describe how the various SRAM chips will be used.

Some chips will store tags, some will store data.

Byte offset is 0 bits, Block offset is 3 bits, Index n bits, Tag 32 – n – 3

It takes 16 chips to get eight words, and we have increments of 128 K words(16 K sets * 1 block/set * 8 words/block). For 16 K sets, 14 bits of index are required, leaving 15 bits of tag. The 15 bits of tag take up one more chip. So, 128 K words requires 16+1 or 17 chips. We can double this, but not quadruple it with the number of chips we have. In total, we can use 34 of the 50 chips and build a cache of 256 K words or 1024 K bytes, or 1 MB. The address is broken down as follows

Byte offset – 0 bits, Block offset – 3 bits, Index – 15 bits, Tag – 14 bits



3. (10 points) Consider a pipeline for a register-memory architecture. The architecture has two instruction formats: a register-register format and a register-memory format. There is a single memory addressing mode (offset + base register). There is a set of ALU operations as follows:

ALUop← Rdest,Rsrc1,Rsrc2,offset

Rdest ← Rsrc1 ALUop Rsrc2
or Rdest ← Rsrc1 ALUop MEM[Rsrc2 + offset]
or
Rdest ← MEM[Rsrc2 + offset]
or
MEM[Rsrc2 + offset] ← Rsrc1
where the ALUop is one of the following:
Add, Subtract, And, Or(with or without offset)
Load(Rsrc1 omitted)
Store(Rdest omitted)
Rdest, Rsrc1 and Rsrc2 are registers.

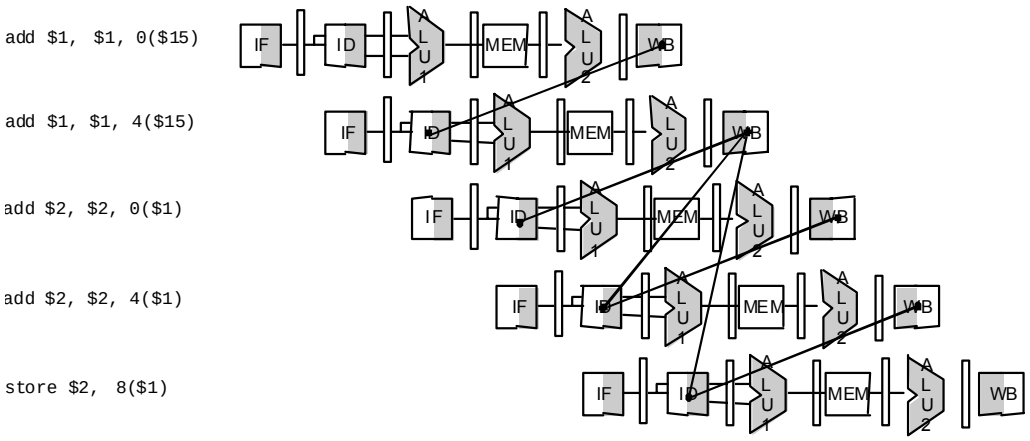
Branches use a full compare of two registers and are PC-relative. Assume that this machine is pipelined so that a new instruction is started every clock cycle. The pipeline structure is

IF	RF	ALU1	MEM	ALU2	WB						
	IF	RF	ALU1	MEM	ALU2	WB					
		IF	RF	ALU1	MEM	ALU2	WB				
			IF	RF	ALU1	MEM	ALU2	WB			
				IF	RF	ALU1	MEM	ALU2	WB		
					IF	RF	ALU1	MEM	ALU2	WB	

The first ALU stage is used for effective address calculation for memory references and branches. The second ALU stage is used for operations and branch comparisons. RF is both a decode and register-fetch stage. Assume that when a register read and a register write of the same register occur in the same clock cycle, the write data is forwarded. For the following code fragment:

```
a  add   $1, $1, 0($15)        Dependencies
b  add   $1, $1, 4($15)        a & b,  c & d
c  add   $2 ,$2, 0($1)         b & c,  d & e
d  add   $2, $2, 4($1)         b & d
d  store $2, 8($1)             b & e
```

identify the data dependencies and draw a figure (using the multiple clock style) showing them as lines.

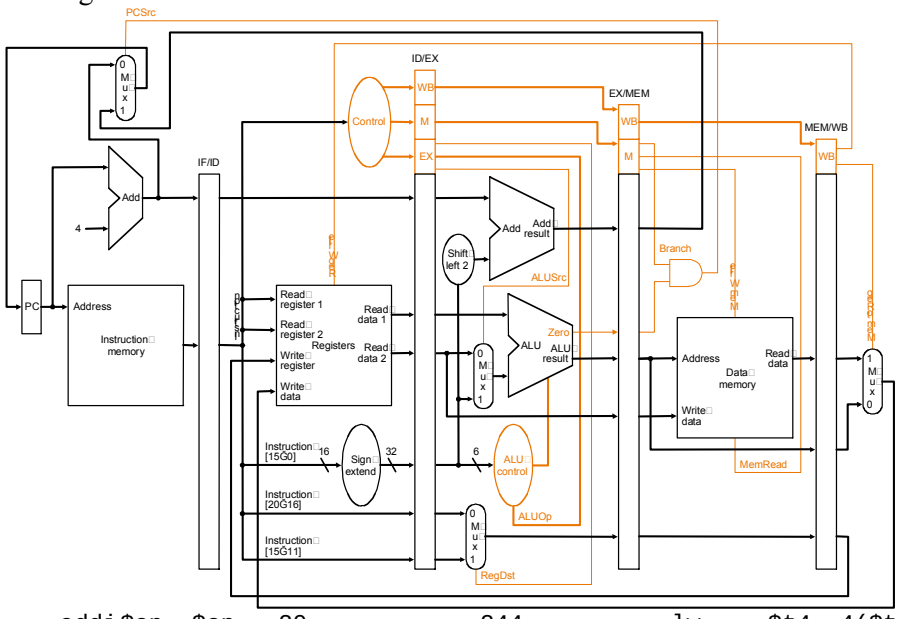


3. (1 point) Assuming a branch is not taken is a crude form of ___branch prediction___.
4. (1 point) MIPS exceptions are collected in the ___cause__ register.
5. (1 point) The ___miss penalty__ is the time to replace a block in the upper level with the corresponding block from the lower level in a hierarchical cache memory system.

6. (1 point) The __tags__ contain the address information required to identify whether a word in the cache corresponds to the requested word.

8. (1 point) Getting a missing data item early from the internal resources of the pipeline is called _____.

9. (20 points) Consider executing the following code on a pipelined datapath like the one shown which has no forwarding.




```
     sort:      addi$sp, $sp, -20          244            lw     $t4, 4($t2)
                sw  $ra, 16($sp)           248            slt    $t0, $t4, $t3
                sw  $s3, 12($sp)           252            beq    $t0, $zero, exit2
                sw  $s2, 8($sp)            256            add    $a0, $s2, $zero
                sw  $s1, 4($sp)            260            add    $a1, $s1, $zero
                sw  $s0, 0($sp)            264            jal    swap
                add $s2, $a0, $zero        268            addi   $s1, $s1, -1
                add $s3, $a1, $zero        272            j      for2tst
200             add $s0, $zero, $zero      276  exit2:     addi   $s0, $s0, 1
304  for1tst:   slt $t0, $s0, $s3          280            j      for1tst
208             beq $t0, $zero, exit1      284  exit1:     lw     $s0, 0($sp)
212             addi$s1, $s0, -1           288            lw     $s1, 4($sp)
216  for2tst:   slt $t0, $s1, 220$zero     292            lw     $s2, 8($sp)
224             bne $t0, $zero, exit2      296            lw     $s3, 12($sp)
228             add $t1, $s1, $s1                         lw     $ra, 16($sp)
232             add $t1, $t1, $t1                         addi   $sp, $sp, 20
236             add $t2, $s2, $t1                         jr     $ra
240             lw  $t3, 0($t2)
```

If the `add $s0` instruction seven instructions after the `sort` label begins executing in cycle 1 and the `beq $t0, $zero, exit1` is taken, what are the values stored in the following fields of the EX/MEM 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 $s0` 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 state is not using them.

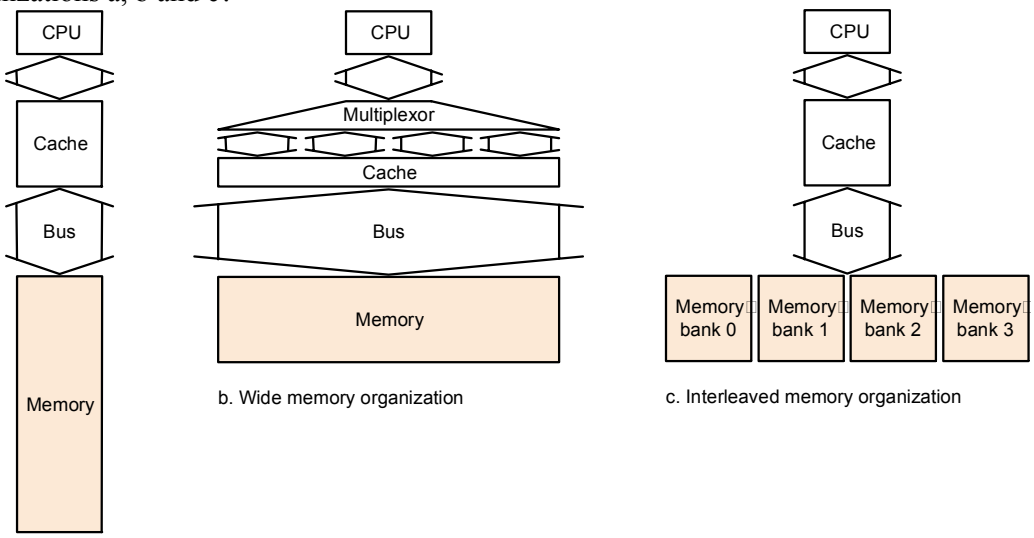
EX/MEM.WB = 11
EX/MEM.MEM = 010
EX/MEM.Target = PC + 4 + 4*SE(offset) = 288
EX/MEM.Zero = 0
EX/MEM.ALUOut = 0 + $sp = 49
EX/MEM.Readdata2 = $s0 = 36
EX/MEM.Writerd = 16

Cycle		ID	EX	MEM	WB
1	`add $s0`				
2	`slt $t0`	`add $s0`			
3	`beq $t0`	`slt $t0`	`add $s0`		
4	`beq $t0`	`slt $t0`	`bubble`	`add $s0`	
5	`beq $t0`	`slt $t0`	`bubble`	`bubble`	`add $s0`
6	`addi $s1`	`beq $t0`	`slt $t0`	`bubble`	`bubble`
7	`addi $s1`	`beq $t0`	`bubble`	`slt $t0`	`bubble`
8	`addi $s1`	`beq $t0`	`bubble`	`bubble`	`slt $t0`
9	`slt $t0`	`addi $s1`	`beq $t0`	`bubble`	`bubble`
10	`bne $t0`	`slt $t0`	`addi $s1`	`beq $t0`	`bubble`
11	`lw $s0`	`bubble`	`bubble`	`bubble`	`beq $t0`
12	`lw $s1`	`lw $s0`	`bubble`	`bubble`	`bubble`
13	`lw $s2`	`lw $s1`	`lw $s0`	`bubble`	`bubble`
14	`lw $s3`	`lw $s2`	`lw $s1`	`lw $s0`	`bubble`

So, the instruction of interest is lw $s0, 0($sp), op = 35, rs = 29, rt = 16, offset = 0

10. (5 points) For an m-stage pipeline, how many cycles does it take to execute n instructions if the pipeline is empty when these instructions begin to execute? _____m+n-1_

11. (15 points) Consider a memory hierarchy using one of the three organizations for main memory shown below. Assume that the cache block size is 32 words, that the width of organization b of the figure is four words, and that the number of banks in organization c is eight. If the main memory latency for a new access is 50 cycles and the transfer time is 3 cycles, what are the miss penalties for organizations a, b and c?



a. One-word-wide memory organization

b. Wide memory organization

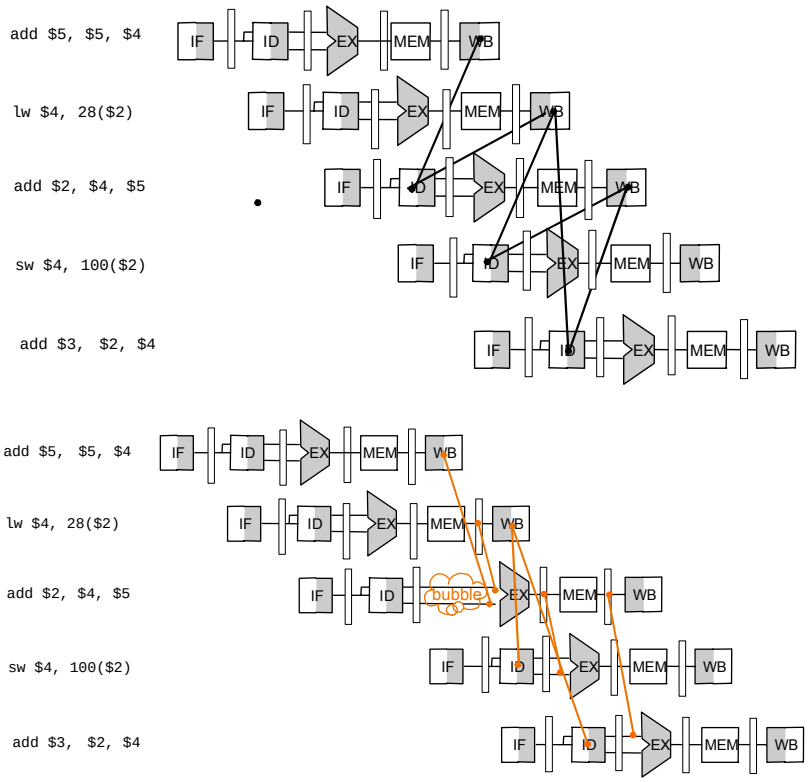
c. Interleaved memory organization

a. send address – 3, read 32*50, transfer 32*3, total 1699 cycles
b. send address – 3, read 8*50, transfer 8*3, total 427 cycles
c. send address – 3, read 4*50, transfer 32*3, total 299 cycles

12. (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.

```
a       add   $5, $5, $4
b       lw    $4, 28($2)
c       add   $2, $4, $5
d       sw    $4, 100($2)
e       add   $3, $2, $4
```

Dependency	Hazard?	Handled by Forwarding?
a & c	yes	after stall is inserted, no forwarding needed
b & c	yes	stall is required, then forward from MEM/WB
b & d	yes	after stall is inserted, no forwarding needed
b & e	no	
c & d	yes	forward from EX/MEM
c & e	yes	forward from MEM/WB



13. (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. Assuming a direct-mapped cache with four-word blocks and a total size of 64 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.

$$64 words * \frac{1 block}{4 words} * \frac{1 set}{1 block} = 16 sets$$

Block Offset - 2 bits, Index - 4 bits

	index	
1	0 0000 01	m
4	0 0001 00	m
8	0 0010 00	m
5	0 0001 01	h
20	0 0101 00	m
17	0 0100 01	m
19	0 0100 11	h
56	0 1110 00	m
9	0 0010 01	h
11	0 0010 11	h
4	0 0001 00	h
43	0 1010 11	m
5	0 0001 01	h
6	0 0001 10	h
9	0 0010 01	h
17	0 0100 01	h

0	0	1	2	3
1	4	5	6	7
2	8	9	10	11
3				
4	16	17	18	19
5	20	21	22	23
6				
7				
8				
9				
10	40	41	42	43
11				
12				
13				
14	56	57	58	59
15				
16				