The University of Alabama in Huntsville
Electrical & Computer Engineering Department
CPE 431 01
Test 2
November 5, 2009

Name: _____

Show all work. You will not receive full credit for a problem if you do not show your work!

1. (1 point) _____ is the space on the disk reserved for the full virtual memory of a process.
2. (1 point) MIPS exceptions are collected in the _____ register.
3. (1 point) The _____ contain the address information required to identify whether a word in the cache corresponds to the requested word.
4. (1 point) Getting a missing data item early from the internal resources of the pipeline is called _____.
5. (1 point)) In the case where memory and cache have different values for the same memory location, the cache and memory are said to be _____.
6. (15 points) Virtual memory uses a page table to track the mapping of virtual addresses to physical addresses. Consider this stream of virtual addresses as seen on a system: 9452, 30964, 19136, 46502, 38110, 16653, 48480. Assume 4KB pages, a four-entry fully associative TLB, and true LRU replacement. If pages must be brought in from disk, increment the next largest page number.

TLB

Valid	Tag	Physical Page Number
1	11	12
1	7	4
1	3	6
0	4	9

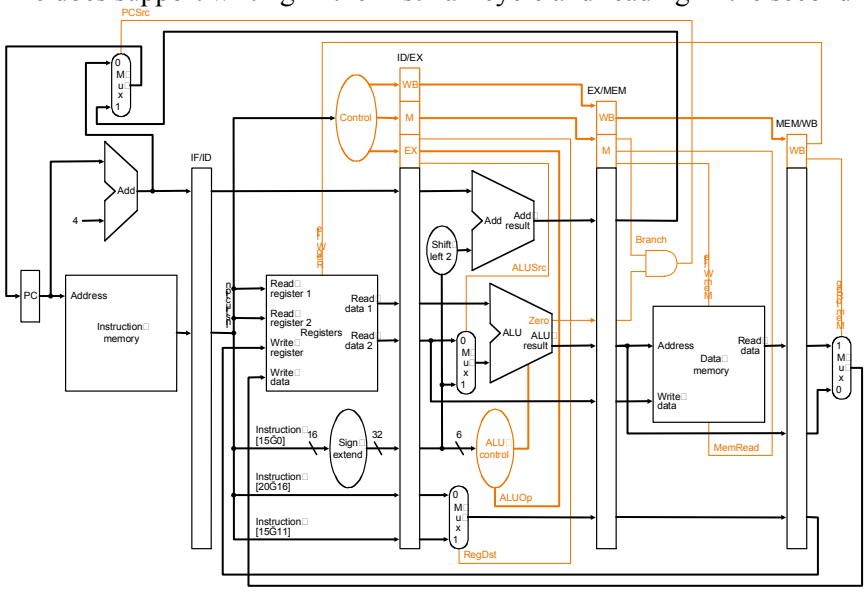
Page table

Valid	Physical page or in disk
1	5
0	Disk
0	Disk
1	6
1	9
1	11
0	Disk
1	4
0	Disk
0	Disk
1	3
1	12

7. (15 points) Unroll the following loop once and schedule it for maximum performance on a MIPS pipeline. Assume that the loop always executes an even number of iterations. You can use registers **$10** through **$20** when changing the code to eliminate dependences. The MIPS pipeline has full forwarding, no delay slots, branch prediction of not taken and branches completing in the ID stage. If the original loop executes 20 times, what is the speedup for the unrolled, scheduled loop as compared to the original loop?

```
Loop: lw    $1, 40($6)
      add   $5, $5, $1
      sw    $1, 20($5)
      addi  $6, $6, 4
      addi  $5, $5, -4
      beq   $5, $0, Loop
```

8. (15 points) Consider executing the following code on a pipelined datapath like the one shown except that it supports j instructions that cause no control hazards and has full forwarding. The register file does support writing in the first half cycle and reading in the second half cycle.




```
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 `lw $t4` instruction six 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 12th cycle? Show the instructions being executed in each stage of the pipeline during each 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 `lw $t4` 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.PC+4 =
IF/ID.Instruction =

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

9. (15 points) Here is a series of address references given as byte addresses: 0, 4, 16, 132, 232, 160, 1024, 30, 140, 3100, 180, 2180. Assuming a four-way set associative cache with one-word blocks and a total size of 8 words that is initially empty and uses LRU, (a) label each reference in the list as a hit or a miss and (b) show the entire history of the cache.

10. (10 points) The following code is written in C, where elements within the same row are stored contiguously. References to which variables exhibit spatial locality?

```
for (J=0; J<8; J++)
   for (I=0; I<8000; I++)
      A[I][J]=B[J][0]+A[J][I];
```

11. (15 points) It is possible to have an even greater cache hierarchy than two levels. Consider a processor with the following parameters.

Base CPI, no memory stalls	Processor speed	Main memory access time	First-level cache miss rate	Second-level cache, direct-mapped speed	Local miss rate with second-level cache, direct-mapped	Third-level cache, eight-way set associative speed	Local miss rate with third-level cache, eight-way set associative	Fourth-level cache, fully associative speed	Local miss rate with fourth-level cache, fully associative
2.0	3 GHz	125 ns	5 %	15 cycles	60 %	50 cycles	43 %	100 cycles	62 %

Calculate the CPI for the processor given that the Memory accesses/instruction = 1.36 and that hits in the L1 cache incur no miss stalls.

12. (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:

```
add   $1, $1, 0($15)
add   $1, $1, 4($15)
add   $2 ,$2, 0($1)
add   $2, $2, 4($1)
store $2, 8($1)
```

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