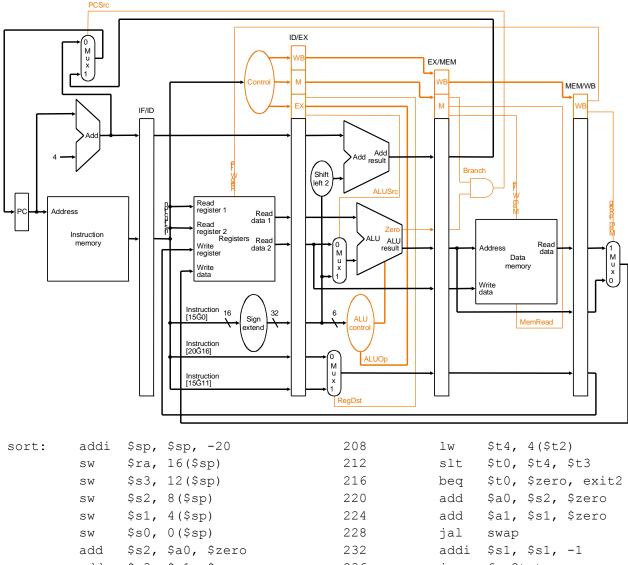
The University of Alabama in Huntsville ECE Department CPE 431 01 Test 2 November 8, 2018

	Name:						
1.	(3 points) The three C's of cache misses are,						
	and						
2.	(1 point) is a technique that uses main memory as a "cache"	for					
	secondary storage.						
3.	(1 point) A is a changing of the internal state of the pro	cessor to					
	allow a different process to use the processor that includes saving the state needed to return to						
	the currently executing process.						
4.	(10 points) Calculate the total number of bits required for a cache with the parameters below, assuming a 32-bit address and byte addressability. Include valid and dirty bits.	listed					
	Cache Data Size: 32 KiB						
	Cache Block Size: 2 64-bit words						
	Cache Set Associativity: 4-way						
	Cache Access Time: 1 cycle						

5. (20 points) Here is a series of address references given as byte addresses: 118, 483, 2069, 321, 368, 1505, 812, 2832, 373, 1411, 511, 122, 690, 4820, 1714, 1508, 2070, 1080. Assuming a two-way set associative-mapped cache with 4-word blocks and a total size of 16 64-bit 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, including tag and data.

Byte			
Address	Byte Address	Byte Address	
(Decimal)	(Hexadecimal)	(Binary)	Hit/Miss
118	76	0000 0000 0111 0110	
483	1E3	0000 0001 1110 0011	
2069	815	0000 1000 0001 0101	
321	141	0000 0001 0100 0001	
368	170	0000 0001 0111 0000	
1505	5E1	0000 0101 1110 0001	
812	32C	0000 0011 0010 1100	
2832	B10	0000 1011 0001 0000	
373	175	0000 0001 0111 0101	
1411	583	0000 0101 1000 0011	
511	1FF	0000 0001 1111 1111	
122	7A	0000 0000 0111 1010	
690	2B2	0000 0010 1011 0010	
4820	12D4	0001 0010 1101 0100	
1714	6B2	0000 0110 1011 0010	
1508	5E4	0000 0101 1110 0100	
2070	816	0000 1000 0001 0110	
1080	438	0000 0100 0011 1000	

6. (15 points) Consider executing the following code on a pipelined datapath like the one shown except that 1) it supports j instructions that complete in the ID stage, and 2) it has full forwarding. The register file supports writing in the first half cycle and reading in the second half cycle.



SOIL:	addi	asp,	\$SP, -20	200	⊥ W	ρί4 , 4(ρίΔ)
	SW	\$ra,	16(\$sp)	212	slt	\$t0, \$t4, \$t3
	SW	\$s3,	12(\$sp)	216	beq	\$t0, \$zero, exit2
	SW	\$s2,	8(\$sp)	220	add	\$a0, \$s2, \$zero
	SW	\$s1,	4(\$sp)	224	add	\$a1, \$s1, \$zero
	SW	\$s0,	0(\$sp)	228	jal	swap
	add	\$s2,	\$a0, \$zero	232	addi	\$s1, \$s1, -1
	add	\$s3,	\$a1, \$zero	236	j	for2tst
	add	\$s0,	\$zero, \$zero	240 exit2:	addi	\$s0, \$s0, 1
for1ts	st: slt	\$t0,	\$s0, \$s3	244	j	for1tst
	beq	\$t0,	\$zero, exit1	248 exit1:	lw	\$s0, 0(\$sp)
	addi	\$s1,	\$s0, -1	252	lw	\$s1, 4(\$sp)
for2ts	st: slt	\$t0,	\$s1, \$zero	256	lw	\$s2, 8(\$sp)
	bne	\$t0,	\$zero, exit2	260	lw	\$s3, 12(\$sp)
	add	\$t1,	\$s1, \$s1	264	lw	\$ra, 16(\$sp)
	add	\$t1,	\$t1, \$t1	268	addi	\$sp, \$sp, 20
200	add	\$t2,	\$s2, \$t1	272	jr	\$ra
204	lw	\$t3,	0(\$t2)			

If the add \$t2 instruction four instructions after the for2tst label begins executing in cycle 1 and the beq \$t0, \$zero, exit2 is taken, what instructions are found in each of the five stages of the pipeline in the 11^{th} cycle? Show the instructions being executed in each stage of the pipeline during each cycle. What value is stored in the ReadData1 field of the ID/EX pipeline register in the 11^{th} cycle? Assume that before the instructions are executed, the state of the machine was as follows:

The PC has the value 200₁₀, the address of the add \$t2 instruction

Every register has the initial value 20₁₀ plus the register number.

Every memory word accessed as data has the initial value 10000₁₀ plus the byte address of the word.

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

IL)/E	ΞX.	Re	adl	Da	ta1	. =						

7. (20 points) Consider the execution of the following loop in a statically scheduled superscalar processor that has full forwarding. Additionally, any branches are resolved in the EX stage.

```
Loop: lw $1, 40($6)
add $1, $7, $1
sw $1, 20($5)
addi $6, $6, 4
addi $5, $5, -4
bne $5, $0, Loop
```

- a. (15 points) Unroll this loop so that four iterations of it are done at once and schedule it for maximum performance on a 2-issue static superscalar processor that has one slot for R-type/branch instructions and one slot for lw/sw instructions. Assume that the loop always executes a number of iterations that is a multiple of 4. You can use registers \$10 through \$20 when changing the code to eliminate dependences.
- b. (5 points) Calculate the number of cycles for the original and for the unrolled, scheduled code and the speedup of unrolled, scheduled compared to the original.

Cycle	R-type/branch	lw/sw
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		

8. (15 points) Virtual memory uses a page table to track the mapping of virtual addresses to physical addresses. The following table is a stream of virtual addresses as seen on a system. Assume 32 KiB pages, byte addressing, a four-entry fully associative TLB, and true LRU replacement. If pages must be brought in from disk, increment the next largest page number. Given the address stream, and the shown initial state of the TLB and page table, show the final state of the system. Also list for each reference if it is a hit in the page table, or a page fault.

TLB

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

Page table

VPN	Valid	Physical page or in disk
0	1	5
1	0	Disk
2	0	Disk
3	1	6
4	1	9
5	1	11
6	0	Disk
7	1	4
8	0	Disk
9	0	Disk
10	1	3
11	1	12
12	0	Disk

	TLB	Page	Page
Address	Hit/Miss	Table	Fault
		Hit/Miss	Y/N
184,760			
110,824			
236,100			
119,200			
56,312			
151,692			
126,856			
40,000			

9. (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

```
ΙF
               ALU1
       RF
                      MEM
                              ALU2
                                     WB
       IF
               RF
                      ALU1
                              MEM
                                     ALU2
                                             WB
               IF
                      RF
                                             ALU2
                              ALU1
                                     MEM
                                                     WB
                      IF
                              RF
                                                     ALU2
                                                            WB
                                     ALU1
                                             MEM
                              IF
                                     RF
                                             ALU1
                                                     MEM
                                                            ALU2
                                                                    WB
                                     IF
                                             RF
                                                     ALU1
                                                            MEM
                                                                    ALU2
```

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)
load $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. You have space on the next page for this figure if you choose to use it.

10. (5 points) The following code is written in C, where elements within the same row are stored contiguously. Does B (j, 0) exhibit spatial locality? temporal locality? Explain your answers. A, B, and C are all arrays of integers 8000 by 8000.

```
for (i = 0; i< 8000; i++)
for (j = 0; j < 8000; j++)
A(i,j) = B(j, 0) + A(j, i) + C(0, i);
```