EEL 6764 Principles of Computer Architecture
Final Exam Sol

Name: _____ **UID:** _____

Problem	Points	Your Points
1	20	
2	24	
3	12	
4	12	
5	10	
6	10	
7	12	
8 (Extra Credits)	6	
Total	100	

- The **final exam** is comprehensive, and takes 120 minute long.
- **Closed textbook, and closed notes.**
- **Other than a scientific calculator, all electronics must be turned off and put away.**
- Make sure that your writing is legible; otherwise you grade may be negatively affected.
- Show work when necessary to get partial credits.
- You are allowed to bring **two** 8.5" × 11" sheets with notes you deem to be necessary.
- Use the back of the exam paper as necessary, but indicate clearly which problems that the answers on the back correspond to.

Problem 1 ($10 \times 2 = 20$ **points)** Multiple choice questions. **More than one choice may be correct. You will get credits only if you identify all correct choices.**

1. Failure rate is inversely proportional to ____a____
   (a) MTTF
   (b) CPI
   (c) CMOS
   (d) MIPS

2. The block number 12 from main memory will go to set number _____ in a 2-way set associative cache that can hold 8 blocks:
   (a) anywhere
   (b) 4
   (c) 8
   (d) 0

3. With increased associativity in caches, the following increase(s):_____
   (a) power
   (b) miss rate
   (c) hit time
   (d) conflict misses

4. Which of the follow techniques improve the miss rate?
   (a) Second level cache
   (b) Write buffer
   (c) Early Restart & Critical Word First
   (d) none of the above

5. ISA affects the following performance related metrics:_____
   (a) instruction count
   (b) clock frequency
   (c) CPI
   (d) all of the above

6. Which of the following are true dependencies that must be met by processors?
   (a) output dependency
   (b) anti-dependency
   (c) control dependency
   (d) data dependency.

7. Reorder buffer is a technique to
   (a) improve memory hit time
   (b) improve prediction accuracy
   (c) enable hardware speculation
   (d) enable write back

8. Which of the follow features describe a hardware speculative superscalar processor?
   (a) in-order issue
   (b) in-order commit
   (c) out-of-order execution
   (d) none of the above

9. In a shared memory multiprocessor,
   (a) memory is physically centralized
   (b) memory can be accessed by all processors
   (c) memory can be distributed
   (d) memory access latency is uniform

10. Snooping protocol solves following problem(s):
    (a) hit time
    (b) miss latency
    (c) miss rate
    (d) cache coherency

Problem 2 ($8 \times 3 = 24$ **points) Answer the following questions concisely and clearly.**

1. Explain *write-allocate* policy.

2. What cache parameter(s) can large cache blocks improve? What cache parameter(s) can become worse with larger cache block size?

3. Explain how a blocking cache works.

4. Suppose that the length of instructions for a processor is 12 bits, and there are 32 general purpose registers. What is the max number of possible encodings for 2-address instructions?

4

5. Consider the following code on the 5-stage MIPS pipeline implementation. Show the logic condition to detect the hazard between those two instructions.

```
LD.   Rx, 45(R2)
DADD  R5, Ry, Rz
```

Make sure you use proper pipeline registers and their corresponding components for the answer.

ID/EX. Rx == IF/ID. Ry or EX/MEM.Rx == IF/ID.Ry or EX/MEM.Rx == IF/ID.Rz

ID/EX. Rx == IF/ID. Rz

6. Describe how simultaneous multithreading works.

7. Consider the code below. Use register renaming to eliminate all dependences while maintaining the original program semantics. Assume that additional registers $F_1, F_2, \ldots$ are at your disposal.

```
ADD.D  F6, F0, F8   <- F1
SUB.D  F8, F10, F14
MUL.D  F8, F10, F8
S.D    F8, 0(R1)
```

F2

8. Describe the key architecture differences between **vector** processors and **superscalar** processors, and concisely explain why vector processors are more energy more efficient.

Problem 3 ($3 \times 4 = 12$ **points**) **Amdahl's Law and Cache Design**

1. Let us say we want to enhance the processor used for Web serving. The new processor is 10 times faster on computation in web serving application than the original processor. Assuming that the original processor is busy with computation 40% of the time and is waiting for I/O 60% of the time, what is the overall speedup gained by this enhancement?

$f = 0.4$, $S = 10$

$$\text{speed up} = \frac{1}{\frac{f}{S} + (1-f)} = \frac{1}{0.04 + 0.6} = \frac{1}{0.64}$$

2. Consider two level-1 data caches; both are 2 bytes in size, 1 byte per block. Cache C1 is direct-mapped, cache C2 is 2-way set associative with LRU replacement. Suppose a program issues load instructions which access bytes of memory in the following sequence: 0, 1, 2, 0, 1, 2, 0, 1, 2, 0, ... for a large number of iterations. What is the approximate miss rate for C1 and C2?

$C_1 = \frac{2}{3}$, $C_2 = 1$

3. Locality of memory accesses is critical for cache performance. Rewrite the following code in order to improve its memory access locality. Assume that 2-dimensional arrays are stored in memory using row-major order.

```
for (j = 0; j < 100; j = j+1)
   for (i = 0; i < 100; i = i+1)
      x[i][j] = x[i][j] * 2
```

```
for (i=0, i<100, i=i+1)
for (j=0, j<100, j=j+1)
x[i][j] = x[i][j] * 2
```

Problem 4 ($2 \times 6 = 12$ points) The Tomasulo's algorithm Assume the instruction latencies as shown below. Time starts at cycle 1. Also assume all registers are ready for use.

ADD	1
SUB	1
MUL	5

Note: The instruction latency is the number of pipeline cycles that an instruction spends in the execute stage.

---

1. Fill the tables when all instructions are issued. Fill the instruction status table (the 1st table) with corresponding cycle numbers.

	Issue	Execute	Write Result
SUB.D F8, F10, F14	1	2	3
MUL.D F6, F10, F8	2	3	
ADD.D F8, F0, F12	3		

RS	Busy	Op	Vj	Vk	Qj	Qk	A
Add1		SUB	Reg[F10]	Reg[F14]			
Add2	*	ADD	Reg[F0]	Reg[F12]			
Mult1	*	MUL	Reg[F10]	Add1			
Mult2							

	F0	F6	F8	F10	F12	F14
Qi		Mult1	Add2			

2. Based on the results from previous question, fill the tables when `ADD.D` finishes *Write Result* step. Fill the instruction status table (the 1st table) with corresponding cycle numbers.

	Issue	Execute	Write Result
SUB.D F8, F10, F14			
MUL.D F6, F10, F8			
ADD.D F8, F0, F12			

RS	Busy	Op	Vj	Vk	Qj	Qk	A
Add1							
Add2							
Mult1							
Mult2							

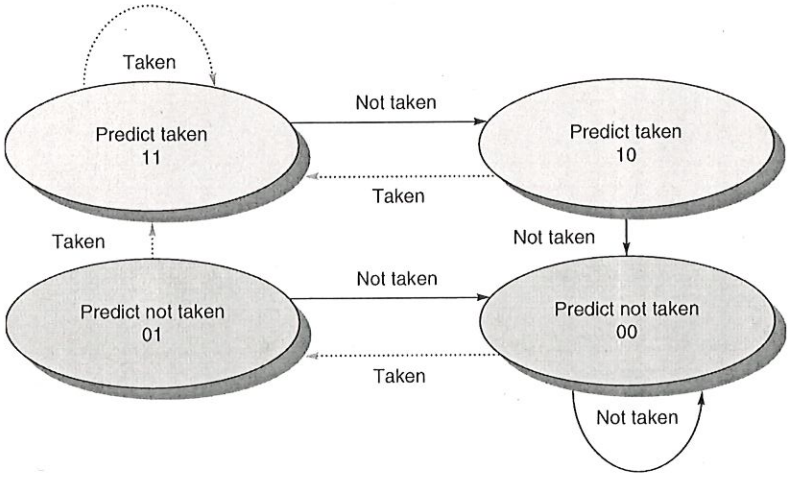
	F0	F6	F8	F10	F12	F14
Qi						

Problem 5 $(5 + 5 = 10$ **points) Branch prediction**

In this problem, ✓ indicates good prediction, while ✗ indicates wrong prodiction. `T` means that a branch is taken, while `N` means that a branch is not taken.

1. Use a 2-bit predictor to predict the branch outcomes for the following sequence by filling the empty entries of the table below. Refer to the the 2-bit prediction FSM under the table.

Cycle	1	2	3	4	5	6	7	8	9	10	11	12	13
Branch Outcome	T,	T,	T,	N,	N,	N,	N,	N,	T,	N,	T,	T,	N
Predictor state	10	11	11	11	10	00	00	00	00	01	00	01	11
Prediction Outcome	✓	✓	✓	✗	✗	✓	✓	✓	✗	✗	✗	✗	✗



2. Assume that branches comprise 20% of all instructions. Also assume that the branch prediction is 90% accurate and incurs a 3 cycle stall on each misprediction. What is the impact of control hazards on the CPI of the pipelined processor? Ignore all other sources of pipeline hazards.

Ideal CPI = 1

Effective CPI = 1 + branch stalls

$= 1 + 0.2 \times (1 - 0.9) \times 3$

$= 1 + 0.06$

$= 1.06$

Problem 6 (10 points) Multiple Issue Consider a multiple-issue design and the code shown below. Suppose you have two execution pipelines, each capable of beginning execution of one instruction per cycle, and enough fetch/decode bandwidth in the front end so that it will not stall your execution. Assume results can be immediately forwarded from one execution unit to another, or to itself. Further assume that the only reason an execution pipeline would stall is to observe a true data dependency. Schedule the code on this two-issue processor for optimal performance. Now how many cycles does each loop iteration require? Show your work.

			Latencies beyond single cycle	
Loop:	LD	F2,0(RX)	Memory LD	+4
I0:	DIVD	F8,F2,F0	Memory SD	+1
I1:	MULTD	F2,F6,F2	Integer ADD, SUB	+0
I2:	LD	F4,0(Ry)	Branches	+1
I3:	ADDD	F4,F0,F4	ADDD	+1
I4:	ADDD	F10,F8,F2	MULTD	+5
I5:	ADDI	Rx,Rx,#8	DIVD	+12
I6:	ADDI	Ry,Ry,#8		
I7:	SD	F4,0(Ry)		
I8:	SUB	R20,R4,Rx		
I9:	BNZ	R20,Loop		

see. 3.3

Sample solution on the next page

	Execution pipe 0		Execution pipe 1
Loop:	**LD F2,0(Rx)**	;	<nop>
	<stall for LD latency>	;	<nop>
	<stall for LD latency>	;	<nop>
	<stall for LD latency>	;	<nop>
	<stall for LD latency>	;	<nop>
	DIVD F8,F2,F0	;	**MULTD F2,F6,F2**
	LD F4,0(Ry)	;	<nop>
	<stall for LD latency>	;	<nop>
	<stall for LD latency>	;	<nop>
	<stall for LD latency>	;	<nop>
	<stall for LD latency>	;	<nop>
	ADD F4,F0,F4	;	<nop>
	<stall due to DIVD latency>	;	<nop>
	<stall due to DIVD latency>	;	<nop>
	<stall due to DIVD latency>	;	<nop>
	<stall due to DIVD latency>	;	<nop>
	<stall due to DIVD latency>	;	<nop>
	<stall due to DIVD latency>	;	<nop>
	ADDD F10,F8,F2	;	**ADDI Rx,Rx,#8**
	ADDI Ry,Ry,#8	;	**SD F4,0(Ry)**
	SUB R20,R4,Rx	;	**BNZ R20,Loop**
	<nop>	;	**<stall due to BNZ>**
	cycles per loop iter 22		

Problem 7 ($3 \times 4 = 12$ **points)** Consider a symmetric multiprocessor with the initial state of the cache and memory as shown below. A processor can execute operations of the form:

```
P#: <op> <address> [<value>]
```

where `P#` designates the CPU (e.g., P0), `<op>` is the CPU operation (e.g., read or write), `<address>` denotes the memory address, and `<value>` indicates the new word to be assigned on a write operation. **Assume the caches are direct mapped, and coherence protocol is snoop write-invalidate.**

P0

	Coherency state	Address tag	Data	
B0	I	100	00	10
B1	S	108	00	08
B2	M	110	00	30
B3	I	118	00	10

P1

	Coherency state	Address tag	Data	
B0	I	100	00	10
B1	M	128	00	68
B2	I	110	00	10
B3	S	118	00	18

.....

P3

	Coherency state	Address tag	Data	
B0	S	120	00	20
B1	S	108	00	08
B2	I	110	00	10
B3	I	118	00	10

Memory

Address	Data	
....	....	....
100	00	10
108	00	08
110	00	10
118	00	18
120	00	20
128	00	28
130	00	30
...	...	...

1. Show **the states of the blocks that are changed** and **the value returned** by operation `P0: read 128`.

P0 : B1 : S : 128 : 00 68, P1 : B1 : S : 128 : 00 68

return 00 68

2. In the state after the above operation, show **the states of all the blocks that are changed** by operation `P0: write 108 <0040>`.

P0 : B1 : M : 108 : 0040

~~P1 : B1 : I : 128 .....~~

P3 : B1 : I : 108 .....

to be continued

3. In directory based coherence protocol, what issue of snoopy protocol does it overcome? And what information is included in directory to support such benefit? Concisely give your answer.

Issue: large # of messages broadcast on bus.

directory: includes nodes that share a particular block.

Problem 8 (Extra credits, 6 points) **Vector Processor** Consider the vector operation $\mathbf{Y} = a \times \mathbf{X} + \mathbf{Y}$ where $\mathbf{X}$ and $\mathbf{Y}$ are both vectors of 64 elements. Show MIPS assembly code for the MIPS superscalar processor, and vector assembly code for VMIPS processor, and compare their performance in terms of the total number of instructions executed on these two processors.

- If you do not remember the ISAs, use simple sentences like "`load X[i]`" for a scalar load, or "`vector-scalar multiply X and a`" for $a \times \mathbf{X}$, etc.
- Assume the vector length of VMIPS is 64.

Skip this one

This page is left blank intentionally.

This page is left blank intentionally.