CMSC411 Spring 2012 Midterm 2

Name: _____

Instructions

- You have 75 minutes to take this exam.
- There are 100 points in this exam, so spend about 45 seconds per point.
- You do not need to provide a number if you can show the appropriate fraction. E.g., 1/13 is acceptable in place of .0769.
- This is a closed book exam. No notes or other aids are allowed.
- If you have a question, please raise your hand and wait for the instructor.
- Answer essay questions concisely using 2-3 sentences. Longer answers are not necessary and a penalty may be applied.
- In order to be eligible for partial credit, show all of your work and clearly indicate your answers.
- Write neatly. Credit cannot be given for illegible answers.

	Problem	Score
1	Memory Hierarchy	/15
2	Instruction Scheduling	/25
3	Dynamic Scheduling	/30
4	Branch Prediction	/30
	Total	/100

Power of 2	Value
2^4	16
2^5	32
2^6	64
2^7	128
2^8	256
2^9	512
2^{10}	1K
2^{20}	1M
2^{30}	1G
2^{40}	1T

1. (15 pts) Memory hierarchy

   a. (6 pts) Suppose we have a virtual memory of size 1 Terabyte (1024GB, or 2^{40} bytes), where pages are 8KB (2^{13} bytes) each, and the machine has 4GB (2^{32} bytes) of physical memory. Compute the number of page table entries needed if all the pages are being used.

   b. (3 pts) Suppose the memory hierarchy consists of a 256K L1 cache, 1 MB L2 cache, 4 GB DRAM memory, and 2 TB disk. If memory accesses were randomly distributed through the entire 1 TB virtual memory (2^{40} bytes), what are the chances that a particular reference would hit the 256K (2^{18} bytes) L1 cache?

   c. (3 pts) Given the answer to (b), explain why programs typically have hit rates of 80% or more in L1 cache?

   d. (3 pts) If the miss rates were 10% for L1 cache, 2% for L2 cache, and 0.1% for memory, what percent of references require accessing the disk (paged virtual memory)?

Instruction	Effect
LD F1, 0(Rx)	F1 ← Mem(Rx)
ADD.D F1, F2, F3	F1 ← F2 + F3
MULT.D F1, F2, F3	F1 ← F2 * F3

I1: LD F1, 0(Rx)
I2: LD F2, 8(Rx)
I3: MULT.D F3, F1, F2
I4: MULT.D F4, F1, F1
I5: ADD.D F4, F2, F1
I6: ADD.D F1, F2, F3

2. (25 pts) Instruction scheduling

   Consider the following sequence of instructions:

   a. (8 pts) List all RAW, WAR, and WAW hazards found in the code.

   b. (3 pts) Compilers may reorder instructions at compile time to reduce stalls. Which registers may be renamed to permit more instructions to be reordered? List both the instruction and register (e.g., I2, F1 refers to the register F1 in instruction I2).

```
I1: LD F1, 0(Rx)
I2: LD F2, 8(Rx)
I3: MULT.D F3, F1, F2
I4: MULT.D F4, F1, F1
I5: ADD.D F4, F2, F1
I6: ADD.D F1, F2, F3
```

Instruction	Latency
Memory LD	+2
ADD.D	+0
MULT.D	+1

c. (8 pts) Given the following instruction latencies, show how instructions would be scheduled (with stalls) if instructions stalled only for true/flow/RAW dependences.

d. (6 pts) Consider a multiple-issue design processor design. Show how instructions would be scheduled (with stalls) if the processor can issue and execute two instructions per cycle. Note instructions must still be issued in order (i.e., an instruction cannot be issued until all previous instructions have been issued). Assume instructions stall only for true/flow/RAW dependences.

Schedule for 2c			Schedule for 2d		
Cycle	Instruction		Cycle	Instruction	Instruction
1			1		
2			2		
3			3		
4			4		
5			5		
6			6		
7			7		
8			8		
9			9		
10			10		
11			11		
12			12		

2. (30 pts) Dynamic scheduling

```
I1: LD F1, 0(Rx)
I2: LD F2, 8(Rx)
I3: MULT.D F3, F1, F2
I4: MULT.D F4, F1, F1
I5: ADD.D F4, F2, F1
I6: ADD.D F1, F2, F3
```

Instruction	Latency
Memory LD	+2
ADD.D	+0
MULT.D	+1

Consider the execution of a single-issue Tomasulo-style CPU. Assume the following:

- The CPU has 1 of each: load buffer, FP adder, FP multiplier functional unit.
- An unlimited number of reservation stations for each functional unit & load buffer.
- Functional units are not pipelined.
- No forwarding between functional units; results can only come from the CDB.
- **If multiple instructions attempt to use the CDB in the same cycle, the instruction issued earliest goes first**.

The table on the right represents the latency incurred by each instruction. Instruction execution times (latencies) are provided as +n cycles, where an instruction executes in 1+n cycles (i.e., spends 1+n cycles in the EX stage). For example: an instruction with latency +2 executed at cycle 4 would finish in cycle 6 (4+2) and put its result on the CDB in cycle 7 (if CDB is not busy).

For each problem, show what clock cycle each instruction is issued and when it begins execution (i.e., enters its first EX cycle). Also show when each instruction writes the CDB. If an instruction execution or completion stalls, **list the length of stall and provide a reason for the stall(s)**.

Instruction	Cycle #			Stalls
	Issue	Exec	Write CDB	
I1: LD F1, 0(Rx)	1	2	5	
I2: LD F2, 8(Rx)				
I3: MULT.D F3, F1, F2				
I4: MULT.D F4, F1, F1				
I5: ADD.D F4, F2, F1				
I6: ADD.D F1, F2, F3				

3. (30 pts) Branch prediction

For a loop containing two branches B1 & B2 (branch actions provided) for each loop iterations, show on each loop iterations the state of the branch predictors (and branch history tables, if needed), the predictions. Assume that all predictors are initialized to not taken, and that the correlation bits are initially set to not taken. When multiple predictors may be used, circle (or underline) the predictors used to make a prediction

a. (8 pts) (2,1) global predictor without branch address

Loop Iteration	Branch B1			Branch B2		
	predictor	prediction	action	predictor	prediction	action
1			NT			T
2			T			NT
3			T			NT
Exit						

b. (12 pts) (2,2) predictor w/ global history + branch address (standard 2-bit counter)

Loop Iteration	Branch B1			Branch B2		
	predictor	prediction	action	predictor	prediction	action
1			NT			T
2			NT			NT
3			T			T
4			NT			T
5			T			T
Exit						

c. (10 pts) tournament predictor (saturating 2-bit counter)

Loop Iteration	Branch B1				
	predictor X	predictor Y	tournament predictor	prediction	action
1	T	NT			NT
2	NT	T			T
3	NT	T			NT
4	NT	T			T
5	T	T			T
6	T	NT			T
Exit					