Your name:

EXAM #2

CS 2410 Graduate Computer Architecture

Spring 2016, MW 11:00 AM – 12:15 PM

Directions: This exam is closed book. Put all materials under your desk, including cell phones, smart phones, smart watches, headphones, laptops, tablets, e-readers, etc. All questions are marked with their point value. There should be plenty of workspace provided in the exam booklet, but if you need extra pages, you may use blank pieces of paper.

Show work: Be sure to show all work and turn in any extra pages that you use. If you do not show your work, you may not receive full or partial credit for a correct or wrong answer. Write legibly. If your handwriting cannot be read, then you will not receive credit for an answer.

Assumptions: If something is confusing or you think there is an error, state your assumptions and explain the issue.

Your name:

1. *2 points* each. Write "T" or "F" to indicate whether the statement is strictly true or false.

___ Scoreboarding renames registers for WAW hazards but not WAR hazards.

____ Scoreboarding cannot forward results, while Tomasulo's algorithm can effectively forward results to implement data flow execution.

___ Tomasulo's algorithm avoids both WAR and WAW hazards.

___ In Scoreboarding, an instruction is stalled in the Read Operand stage until all source operands are available.

___ In Tomasulo's algorithm, an instruction reads all operands from the register file.

____ The only information that needs to be kept in a reservation station is the opcode (operation type), availability (busy) of the functional unit, and the tags of producer functional units of source operands.

____ Speculation requires branch prediction, dynamic scheduling, and buffering results of uncommitted instructions.

___ Increasing set associativity never increases the conflict miss rate.

____ Using a larger block size for a cache of fixed capacity will not decrease the cache miss rate.

___ A typical modern high-performance processor has at most two levels of cache.

_____ Data cache prefetching can be very effective when programs have regular access patterns to memory, regardless of the data structure type (e.g., array vs. heap object).

___ Compulsory misses (a.k.a., cold misses) are not affected by changing cache capacity.

___ Capacity misses are not affected by changing cache associativity.

___ Write through caches use write allocate to avoid writing dirty blocks to lower cache levels.

___ In a processor with speculative execution, a store can always be committed as soon as its stored value is available in the store buffer.

_____ In a processor with speculative execution, a load buffer allows the processor to reorder the execution of loads without worrying about store instructions.

2. *12 points*. Consider the pseudo-code below. Assume that each array reference (A[i], A[i-1], A[i+65]) corresponds to one memory instruction (load or store). Assume there are no other loads or stores in the loop other than the array references. Consider only the data cache. Suppose A is an array of integers, A[0] is located at address 0, an integer is 4 bytes, and the addition expression is evaluated left to right (i.e., A[i-1] is loaded before A[i+65]). For each cache, indicate in the table whether the given array access is a cache hit or miss. Use "M" for miss and "H" for hit.

```
for (i = 2; i < 18; i += 2)
   A[i] = A[i-1] + A[i+65];
```

- Direct-mapped, 16-byte block size, 16 blocks, writeback, write-allocate.

i	**A[i]**	**A[i-1]**	**A[i+65]**
2			
4			
6			
8			
10			
12			
14			
16			

- 4-way set-associative, 16-byte block size, 16 blocks, writeback, write-allocate.

i	**A[i]**	**A[i-1]**	**A[i+65]**
2			
4			
6			
8			
10			
12			
14			
16			

3. Consider a memory system with two levels of cache L1 and L2. Ignore the instruction cache. Assume the L1 data cache is 2-way set associative, physically tagged/indexed, has 128 blocks, the block size is 16 bytes, the hit time is $1ns$ and the hit rate is 60%. The L2 cache is 16-way set-associative with 1024 blocks and a block size of 64 bytes. It has a hit time of $4ns$, and a hit rate of 90%. When main memory is accessed on a L2 cache miss, the L2 cache takes $40ns$ to get the first word and 1ns for each subsequent word in the cache block. Both caches are write-back, write-allocate and operate on physical addresses. The physical address size is 32 bits and a word is 4 bytes. Answer the following questions.

   - *3 points* How many bits are needed for the set index for the L2 cache?

   - *3 points* How many bits are needed for the tag of the L2 cache?

   - *3 points* How much total storage (in bits) does the L2 cache require to store the tags?

   - *4 points* What is the miss penalty for the L2 cache?

   - *5 points* What is the miss penalty for the L1 cache?

(Continues on next page.)

- *4 points* What is the average memory access time (AMAT) for the memory?

- You will notice that the L1 and L2 caches have different block sizes. Think about why this choice might be made and the implications of it.
  - *3 points* Why might this choice be made (what are the advantages)?

  - *3 points* What are the implications of this choice (what are the disadvantages)?

4. *16 points*. Let's consider Tomasulo's algorithm. Assume add takes 1 cycle in execute and a multiply takes 2 cycles. Complete the state/reg tables for each clock cycle, given the code:

```
S1: MULTD F0,F1,F2
S2: ADDD F3,F0,F1
S3: ADDD F1,F2,F1
```

For each cycle, the first table is the reservation station state. Op is one of S1, S2, or S3; V is a value; and Q is a res. station name. The second table is Reg Map. Here, write the producer for a renamed register. Write either "N/A" or leave blank to indicate there is "no" state associated with that field in the Reservation Station or Reg Map tables.

Assume the registers initially have the values: F0=10, F1=20, F2=30, F3=40.

Cycle 0: Issue S1

Res. Station	Op	Vj	Vk	Qj	Qk
mult					
add1					
add2					

Reg	F0	F1	F2	F3
Name				

Cycle 1: Issue S2

Res. Station	Op	Vj	Vk	Qj	Qk
mult					
add1					
add2					

Reg	F0	F1	F2	F3
Name				

Cycle 2: Issue S3

Res. Station	Op	Vj	Vk	Qj	Qk
mult					
add1					
add2					

Reg	F0	F1	F2	F3
Name				

Cycle 3: Continue (nothing to issue)

Res. Station	Op	Vj	Vk	Qj	Qk
mult					
add1					
add2					

Reg	F0	F1	F2	F3
Name				

5. Consider Tomasulo's algorithm extended with a reorder buffer for speculative execution.

   - *4 points* Explain the purpose of the Commit stage.

   - *4 points* Explain how mispredicted branches can be most efficiently handled.

   - *4 points* Explain when exceptions can be handled.