CPSC 3300 – Spring 2017 – Final Exam Name: _____

1. Find the execution time for a program that executes 5 billion instructions on a processor with an average CPI of 2 and a clock frequency of 2.5 GHz. (2 pts.)

2. For the following instruction set workload and cycle values, find the average CPI. (1 pt.)

type	freq	cycles
alu	0.4	1
branch	0.2	2
ld/st	0.4	3

3. What is the overall speedup if an enhancement with speedup of 10 can be used 2/3rds of the time? Use Amdahl's Law to determine the answer, and show the answer as a fraction. (2 pts.)

4. Consider the following benchmark results. Would you be able to replicate the results? If not, explain why not. (4 pts.)

> ***Sieve of Eratosthenes Benchmark in Java*** [from https://imagej.nih.gov/nih-image/java/benchmarks/sieve.html]
>
> This is a simple integer benchmark that generates a list of prime numbers. Note that moving the mouse while the benchmark is running may result in lower scores. […]
>
> This benchmark only measures integer "CPU" performance. To measure graphics perfomanace [sic], run the Plasma benchmark or the "Plugins/Utilities/Benchmark" command in the ImageJ.

Score	System	Java Virtual Machine
19,328	Intel i7/2.8x4, Win7	IE 8, Java 1.6 (64-bits)
17,064	Intel i7/2.8x4, Win7	IE 8, Java 1.6 (32-bits)
13,501	Intel-Mac/3.0x4	Safari, Java 1.5
…	…	…

Arithmetic/Harmonic/Geometric. Circle ***only one*** of A, H, or G. (1 pt. each)

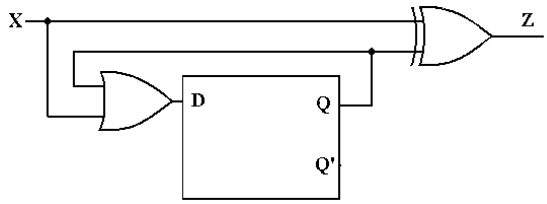
5. A / H / G Used for averaging execution rates.
6. A / H / G Used for averaging execution times.
7. A / H / G Used for averaging by SPEC when reporting scores for benchmark suites.

8. Is a 32-bit-wide ALU an instance of combinational or sequential logic? Explain. (1 pt.)

9. Is a 32-bit register an instance of combinational or sequential logic? Explain. (1 pt.)

10. Design a two-bit counter with the following repeated sequence: 0,2,3,1. (2 pts. each)
    (a) Give the state diagram.
    (b) Give the state transition table with current states QA(t) and QB(t), and next states QA(t+1) and QB(t+1).
    (c) Give the simplified logic expressions for QA(t+1) and QB(t+1).

11. Analyze the following synchronous sequential logic circuit and give the state diagram. The clock input to the D flip-flop is implicit. (5 pts.)



12. Identify and briefly explain at least two differences between a hardwired control unit and a microprogrammed control unit. (2 pts.)

13. What are the five stages in the standard pipeline we studied, and what action does each perform for the add instruction add $1,$2,$3? (5 pts.)

14. Explain why the standard five-stage pipeline has a load-use data hazard even with data forwarding. (2 pts.)

15. For the MIPS instruction sequence below, identify the dependencies in a data dependency diagram. (4 pts.)

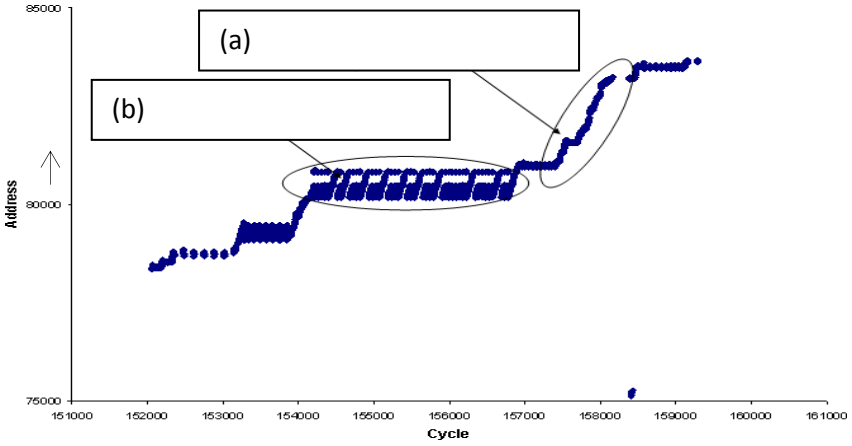
```
i1: add $1, $2, $3  // Reg[1] <- Reg[2] + Reg[3]
i2: ld  $4, 0($1)   // Reg[4] <- Memory[ 0 + Reg[1] ]
i3: sub $1, $4, $5  // Reg[1] <- Reg[4] - Reg[5]
i4: sw  $7, 12($8)  // Memory[ 12 + Reg[8] ] <- Reg[7]
```

16. Circle one answer. (2 pts.) A reorder buffer provides for:
    a. branch misprediction recovery
    b. precise exceptions even in the face of out-of-order completion of instruction execution
    c. both a and b
    d. neither a nor b

17. Fill in the empty cells (a)-(d) in this table. (2 pts. each)

Memory hierarchy level	Access time	Type of traffic between levels	Unit of transfer	Agent of transfer
Registers	(a)			
		load, store, and instruction fetch	word	compiler
Cache	1-5 nanoseconds			
		(b)	(c)	(d)
Main memory	20-100 nanoseconds			
		page read on a page fault and write back of a modified page on page replacement	page	OS
Disk	1-10 milliseconds			

18. In the memory reference diagram shown below, fill in boxes (a) and (b) with the name of the type of memory referencing behavior that best describes the associated circled regions of memory references. (3 pts. each)



19. What is a burst-mode transfer from a memory controller? (2 pts.)

20. Draw a bus cycle timing diagram showing activity on the address, data, and control lines of the memory bus for a 4-word burst-mode memory read transfer with two wait states. (4 pts.)

21. Consider a computer system that contains a cache with 5-nsec access time and a memory with 50-nsec access time. If the hit rate is 80% and a cache miss requires both a single cache access and a single memory access, what is the average memory access time (AMAT)? (4 pts.)

22. Consider an 8 GiB byte-addressable main memory and a 1 MiB 4-way set-associative cache with 16 bytes/line.

(a) How many total lines are there in cache? (not just per bank) (2 pts.)

(b) Show how the main memory address is partitioned into fields for the cache access and give the bit lengths of those fields. (6 pts.)

(c) How many total tag comparators are there in the cache? (1 pt.)

23. Explain what each of the four states represent in the MESI cache coherency protocol by filling in the empty cells in the table below. (1/2 pt. each)

State	Letter stands for?	In this cache?	Possibly in other caches?	Are the contents of the line in this cache identical to the contents in memory?
M		Yes		
E				
S		Yes		
I	Invalid	No	Yes	Not Applicable

24. Explain why write invalidate is typically preferred over write update. A superficial answer such as “better performance” or “efficiency” with no explanation will be assigned 0 points. (2 pts.)

25. Assume a 256-byte main memory and a four-line cache with **two bytes per line**. The cache is initially empty. For the byte address reference stream (reads) given below circle which of the references are hits for an 8-byte direct-mapped cache. Also, show the final contents of the cache. (The byte addresses are in decimal.) (8 pts.)

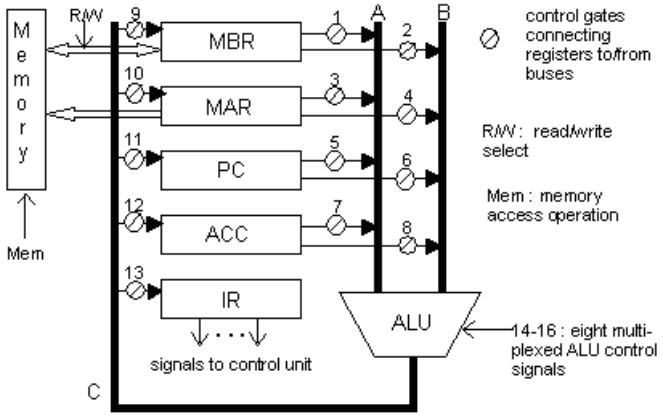
    0, 12, 1, 2, 13, 3, 4, 12, 5, 6, 13, 7

26. Assume a 256-byte main memory and a four-line cache with **two bytes per line**. The cache is initially empty. For the byte address reference stream (reads) given below circle which of the references are hits for an 8-byte two-way set-associative cache with LRU replacement. Also, show the final contents of the cache. (The byte addresses are in decimal.) (8 pts.)

    0, 12, 1, 2, 13, 3, 4, 12, 5, 6, 13, 7

27. What is false sharing, and what programming step do you take to avoid false sharing? (4 pts.)

XC. Consider the following datapath. Assume all registers are edge-triggered and thus immune from races. Numeric control signal identifiers 1-13 are given for the "in" and "out" control points of the registers. Additional control signals include memory signals "Mem" (to enable a memory access), "R" (read select, causing MBR <- memory[MAR] when used in conjunction with Mem), "W" (write select, causing memory[MAR] <- MBR when used in conjunction with Mem), and a 3-bit ALU function field F (causing an action as shown in the table). A, B, and C are internal busses.




```
ALU functions (three-bit F field)
---------------------------------
000: C = A + B     100: C = A - B
001: C = A         101: C = not A
010: C = A + 1     110: C = A - 1
011: C = A << 1    111: C = A >> 1
```

Complete the step-by-step RTL and the control signal sequence to fetch and execute a "add X" instruction. Assume that the memory is word-addressable, that initially the address of the instruction is in the PC, and that the instruction is composed of two memory words: a one-word opcode followed by a one-word address. The action of the instruction is ACC <- ACC + memory[X], where X is the memory address given in the second word of the instruction. (up to 5 pts.)

// fetch opcode and place in IR	// control signals		
MAR <- PC	5 (A=PC),	F=001 (C=A),	10 (MAR=C)
PC <- PC + 1	5 (A=PC),	F=010 (C=A+1),	11 (PC=C)
MBR <- memory[MAR]	Mem/R		
IR <- MBR	1 (A=MBR),	F=001 (C=A),	13 (IR=C)