Computer Science Comprehensive Examination
Computer Architecture
[60 points]

This examination is open book. Please do all of your work on these sheets. Do not do your work in a blue book.

By placing your number below you indicate that you agree in the spirit of the Stanford Honor Code to neither give nor receive unpermitted aid on this exam.

Number: _____

Problem	Max Score	Your Score
1	48	
2	25	
3	27	
TOTAL	100	

Problem 1: Short Answer [4 points each, 48 points total]

A. Cache A is an 8K-byte direct-mapped cache. Cache B is a 16K-byte four-way set associative cache. Cache C is a 32K-byte four-way set associative cache. All caches have sixteen-byte lines, start in the same initial state of all lines invalid, use an LRU replacement policy, and see the same sequence of memory references. After this sequence of references, which of the following statements are true. Circle all that apply.

(a) Cache B will always contain every line in cache A
(b) Cache C will always contain every line in cache B
(c) Cache C will always contain every line in cache A

B. Compared to an 8K-byte direct-mapped cache, what type of misses will a 16K-byte direct-mapped cache have fewer of? ? Circle all that apply.

(a) compulsory
(b) conflict
(c) capacity

C. Adding a cache memory to a system changes which of the following memory parameters? Circle all that apply and denote the direction of change with an up arrow or a down arrow.

(a) Memory latency
(b) Memory bandwidth
(c) Memory address space
(d) Memory reliability

D. In a 32K-byte four-way set-associative cache with 32-byte blocks, how large is the index field used to address the cache array? (write down the number of bits)

_____

E. If the cache of question D is physically tagged and physical addresses are 40-bits long, what is the minimum possible length the tag may be for correct operation? (write down the number of bits)

_____

F. Which instruction set will give the best code density?

(a) An accumulator instruction set
(b) A RISC instruction set with fixed 32-bit instruction formats
(c) A CISC instruction set with variable-length instruction formats
(c) A stack instruction set

G. Which mean should be used to combine execution times of programs? Circle all that apply.

(a) Arithmetic mean
(b) Geometric mean
(c) Harmonic mean

H. In the steady state, what will be the prediction accuracy of a two-bit branch predictor on the sequence TTTTNTTTTN (T=taken, N=not taken)?

I. Which will have better throughput, a machine that uses scoreboarding (with no bypassing) to avoid read-after-write hazards, or a machine that uses bypassing to avoid these hazards?

(a) Scoreboarding with no bypassing
(b) Bypassing

J. A processor has a six-stage pipeline with stages Fetch, Decode, Register Read, Execute, Memory, Register Write and is able to execute one instruction per cycle in the absence of branches. All branches are predicted as not taken. When a branch is taken, the condition and address are both generated during the Execute pipeline stage. If conditional branches account for 10% of all instructions and 30% of conditional branches are taken, what will be the execution rate of this processor? Express your answer in clocks per instruction (CPI).

_____

K. A machine with register renaming is able to reorder instructions without regard to what type of dependencies? Circle all that apply.

(a) Data dependencies
(b) Output dependencies
(c) Anti-dependencies
(d) Control dependencies

L. An instruction for protected subsystem entry (sometimes called system call) must change which subset of the following five things atomically? (circle all that apply)

(a) The contents of a data register
(b) The program counter (sometimes called instruction pointer)
(c) The contents of the page table
(d) The contents of the cache
(e) The privilege level

Problem 2: Pipeline Architecture [25 points total]

Suppose you have a CPU with a 6 stage pipeline containing the following stages:

F: instruction fetch
D: instruction decode
R: register read (and branch address calculation)
A: execute ALU operations (including determination of branch condition)
M: memory load and store
W: write back to register file

The register file cannot read a value that is being written in the same cycle. The machine has bypass paths from the output of the A, M, and W stages to both inputs of the ALU.

A. [5 points] What is the latency of an unconditional branch on this machine? Assume no prediction or speculation is employed. (Note: Latency is defined as the number of cycles from when this instruction is executed until the next instruction is executed. E.g., the latency of a simple instruction with no dependencies is one.)

B. [5 points] What is the latency of a conditional branch? Again assume no speculation about branch direction or distance.

C. [5 points] The instruction register at each pipeline stage is denoted by IR_{stage}; for example IR_A is the instruction register at the ALU stage. The source and destination fields of the IR are denoted IR.A (source 1), IR.B (source 2), and IR.C (the destination). For example, the destination field of the IR at the M stage is IR_M.C. Each IR also has a valid field IR.V that indicates when it contains a valid instruction. Using this notation, write a logical expression that indicates when the bypass path from the output of the M stage to the first (A) input of the A stage should be activated.

D. [10 points] Consider the following instruction sequence:

```
Brz     r5, EXIT
Add     r1 <- r5 + r6
Load    r2 <- [r1 + 4]
Add     r4 <- r2 + r3
```

Assuming that the branch is not taken and that no instructions are fetched speculatively, how many cycles does this take to execute from the time the IP points to the branch instruction until the result of the add is written to the register file? Explain your answer. You may want to draw a timeline.

Problem 3. Instruction Issue [27 points total]

Consider the following instruction sequence:

```
1       LD      X,R1
2       ADD     R1,#5,R2        ; R2 <- R1 + 5
3       LD      Y,R3
4       ADD     R2,R3,R4        ; R4 <- R2 + R3
5       LD      Z,R5
6       ADD     R1,R5,R6        ; R6 <- R1 + R5
7       MUL     R6,R4,R7        ; R7 <- R6 * R4
8       ADD     R7,#3,R8        ; R8 <- R7 + 3
```

You may assume that all arithmetic operations have two-cycle latency and all loads have four-cycle latency. That is, the result of an arithmetic (memory) operation is available two (four) cycles after that operation enters the first execution stage of the machine. Also assume that there is full bypassing, that all loads hit in the cache, and that an arbitrary number of loads (and arithmetic operations) can be in flight at a single time. Hint: you need only consider the execution and memory stages of the pipeline to answer this question.

A. [7 points] How many cycles does this sequence take to execute on a single-issue in-order machine? Measure time from the cycle that the first instruction issues (enters the execution stage) to the cycle in which last instruction completes. For example, the load of instruction 1 issues in cycle 1 and completes in cycle 4. This enables the ADD of instruction 2 to issue in cycle 5 and complete in cycle 6. Show your work.

B. [7 points] If issue order and resources are not limited, what is the shortest time this sequence of instructions could take? Show your work.

C. [7 points] How many cycles does this sequence take to execute on an in-order machine with multiple issue (assume an issue width as wide as you need)? Show your work

D. [6 points] Can you statically reorder the code to give the wide in-order machine of part C the performance bound of part B? If so, show the new ordering.