(TEST) BUT NO solutions)

Computer Science Comprehensive Examination
Computer Architecture
[60 points]

Problem 1: Short Answer [2 points each, 12 points total] (Keep your answer to one line or less)

A. Increasing the block size of a cache while holding total size and associativity constant will improve performance for programs that exhibit what type of locality? # of blocks fewer

spatial

B. List one advantage and one disadvantage of variable-length instruction sets.

C. Dynamic scheduling using register renaming allows instructions to run out-of-order without concern for which type or types of hazards? WAW - output
WAR - anti-dependencies

D. Suppose you run a program that has 50% vector operations and 50% scalar operations on a machine with 10MFLOPS scalar performance and 1GFLOPS vector performance. To first approximation what will the performance of your program be?

E. Does a pipelined processor with a single-cycle ALU and bypassing need interlocks to stall the pipeline on read-before-write hazards on arithmetic operations? Why?

F. Which will have better performance on conditional code (code with many 'IF' statements), a SIMD machine or a MIMD machine?

Problem 2: Cache Architecture [3 points each, 12 points total]

A. Suppose you have a 4Kbyte 2-way associative cache with a block size of 32 bytes. What size tag array (directory) is needed for this cache?

B. Suppose you take the cache from part (A) and divide each block into two 16-byte half-blocks, each with their own valid bit. (This is called a sectored cache). With the new organization, only a half-block is fetched on each cache miss but replacements are done a whole block at a time. What affect does this change have on miss rate? on miss penalty?

C. For what types of programs would the sectored cache of (B) outperform the cache of (A)?

D. Suppose you have two caches with a capacity of 4 4-byte blocks. One is direct mapped and the other is fully associative. Give an address sequence (byte addresses) for which the fully associative cache outperforms the direct-mapped cache. Give a second sequence for which the direct-mapped cache outperforms the fully associative cache.

Problem 3: Pipeline Microarchitecture [12 points]

A. [5 points] Consider the five-stage pipeline shown on the last page. Suppose it is intended to execute a fixed-format 32-bit instruction set with a 5-bit opcode, 3 5-bit register specifiers, and a 12-bit signed immediate. All data and address paths are 32-bits. Draw the missing lines connecting the boxes and label each with its width in bits. Insert any multiplexers that are needed for correct operation without bypassing. Do not add bypass multiplexers.

B. [5 points] Consider the instruction sequence

```
Load  r2 <- [r1 + 4]
Add   r4 <- r2 + r3
Store [r1 + 8] <- r4
```

How many cycles does this take to execute from the time the IP points to the load instruction until the store has committed? Explain your answer. You may want to draw a timeline.

C. [2 points] Suppose the `Store` instruction in the sequence above raises an exception in the fetch stage (perhaps a protection violation in accessing the I-cache). Explain briefly (3 lines or less) how your pipeline must handle this exception to ensure 'precise exceptions'.

Problem 4: Virtual Memory [12 points]

A. [2 points] Consider two machines. X supports just segmented memory, and Y supports just paged memory. List one advantage in favor of each machine.

B. [5 points] Virtual addresses are byte addresses and are 32-bits in length. Y's pages are 4K-bytes in length and are referenced through a direct-mapped translation-lookaside buffer (TLB) with 8 entries. Sketch a block diagram showing the translation process and label and dimension the fields of the virtual address and physical address.

C. [5 points] X supports arbitrary sized segments from 4-bytes up to 2^{32} bytes in length (in increments of 4-bytes) and can support up to 2^{30} 4-byte segments. X uses a fully associative TLB with address masking to perform the translation. Sketch a possible format for an entry from this TLB labeling and dimensioning each field.

Problem 5: Performance [12 points]

Suppose you have a machine with the following instruction mix:

ALU	60%
LOAD	20%
STORE	10%
BRANCH	10%

All ALU instructions complete in one cycle. Branch instructions complete in one cycle if correctly predicted but take 5 cycles if mispredicted. The machine has blocking LOAD and STORE instructions. That is, subsequent instructions are stalled until each access completes.

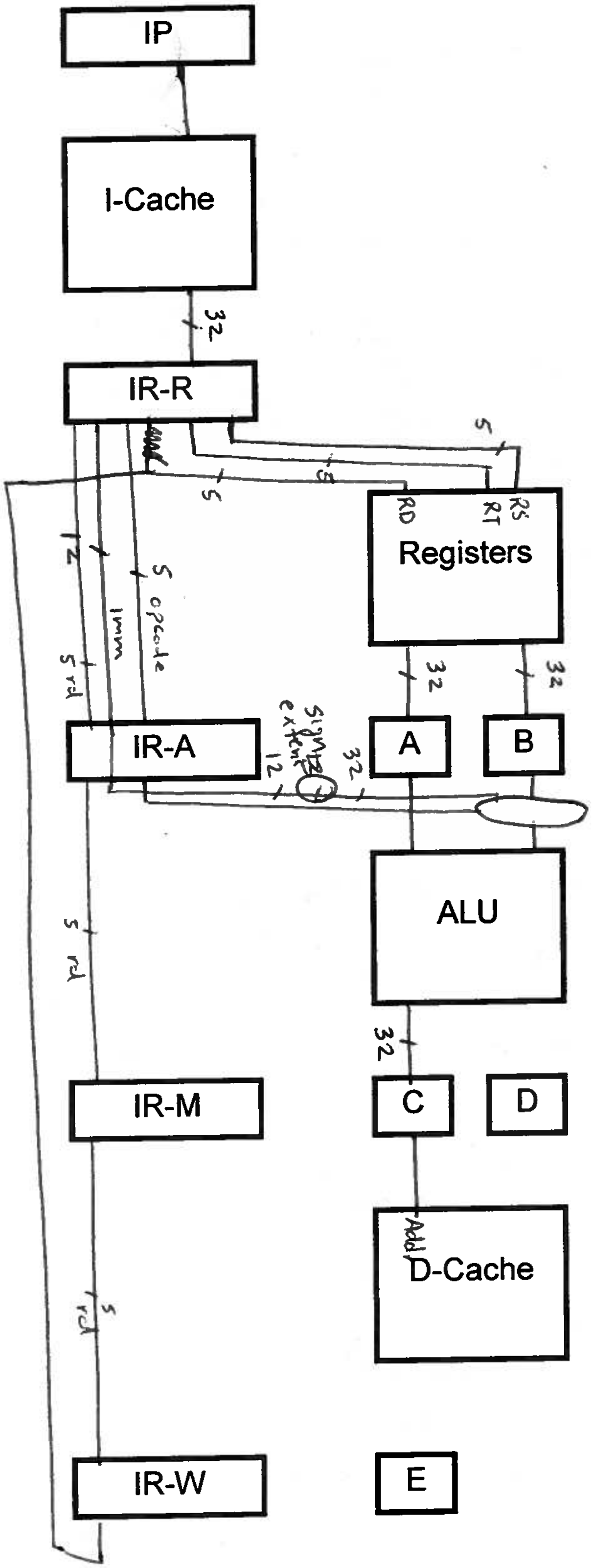
A. [2 points] If you had an ideal memory system (all accesses complete in one cycle) and the branch predictor is 80% accurate, what will the CPI of this machine be?

B. [10 points] Unfortunately ideal memory systems cannot be built. You are considering three alternatives for a memory system with parameters as shown below.

	L1		Main Mem	
Option	Miss	Access	Access	
A	10%	1	20	
B	5%	3	20	
C	10%	2	20	Non blocking

The first option (A) is to use a small cache that can be accessed in one cycle but has a high miss rate (10%). The second option is to use a larger cache that has a smaller miss rate but has a 3-cycle access time on a hit. The third option is to use the small cache but to make memory instructions non-blocking. That is, the pipeline is never stopped by a store (we'll assume an infinite write buffer for now) and is stopped by a load only when the result is needed. The complexity of the non-blocking memory hardware adds a cycle to the hit latency. Assume that on average execution can continue for 5 instructions before the load value is needed.

What is the average memory access time for each approach? What is the CPI for each approach? Which would you choose?


IP
I-Cache
32
IR-R
5
5
5
RD
RT
RS
Registers
12
Imm
5 opcode
5 rd
32
32
IR-A
A
B
sign extent
12
32
ALU
5 rd
32
IR-M
C
D
Add
D-Cache
5 rd
IR-W
E