

**Computer Science Department
Stanford University
Comprehensive Examination in Computer Architecture
Autumn 1998**

October 28 1998

READ THIS FIRST!

1. All work to be done on the test itself! Fill in the blanks, **NO BLUE BOOKS**. There are 5 pages.
2. This exam is **CLOSED BOOK**. You may not use notes, articles, or books.
3. Show your work, since **PARTIAL CREDIT** will be given for incomplete answers. For example, you can get credit for making a reasonable start on a problem even if the idea doesn't work out; you can also get credit for realizing that certain approaches are incorrect. On a true/false question, you might get partial credit for explaining why you think something is true when it is actually false. But no partial credit can be given if you write nothing.

Computer Science Comprehensive Examination

Computer Architecture

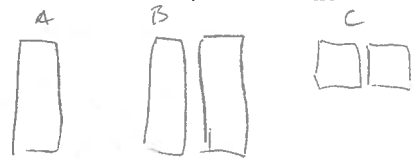
[60 points]

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

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

- A. Suppose machine A has a 8K-byte direct mapped cache, machine B has a 16K-byte two-way set associative cache, and machine C has an 8K-byte two-way set associative cache. All caches have 16-byte lines. If the three machines process identical sequences of memory references, which of the following statements are true? Circle all that apply.

- (a) B's cache will always contain every line in A's cache
 (b) C's cache will always contain every line in A's cache
 (c) B's cache will always contain every line in C's cache
 (d) A's cache will always contain every line in C's cache



- B. Which instruction set makes it easier to express instruction-level parallelism: a stack instruction set or a three-address general-register instruction set?

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

- (a) Data dependencies RAW
 (b) Output dependencies WAW
 (c) Anti-dependencies WAR
 (d) Control dependencies br

- D. Adding an interleaved 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

- E. A RISC machine with a 5-stage pipeline uses the adder in the execution stage to perform branch address calculations. If we add a special branch address adder to the register read stage, what instructions will benefit? By how much?

unconditional branches, jumps
 end of RR → end of X
 one cycle

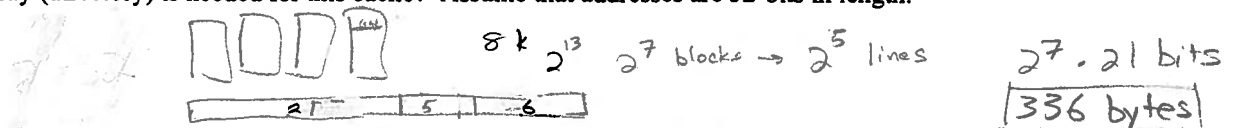


- F. Which will have better throughput, a machine that uses scoreboarding (with no bypassing) to avoid read-after-write hazards, or a machine that uses bypassing to avoid these hazards?

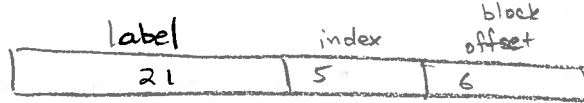
bypassing

Problem 2: Cache Architecture [3 points each, 15 points total]

A. Suppose you have an 8Kbyte 4-way associative cache with a block size of 64 bytes. What size tag array (directory) is needed for this cache? Assume that addresses are 32-bits in length.



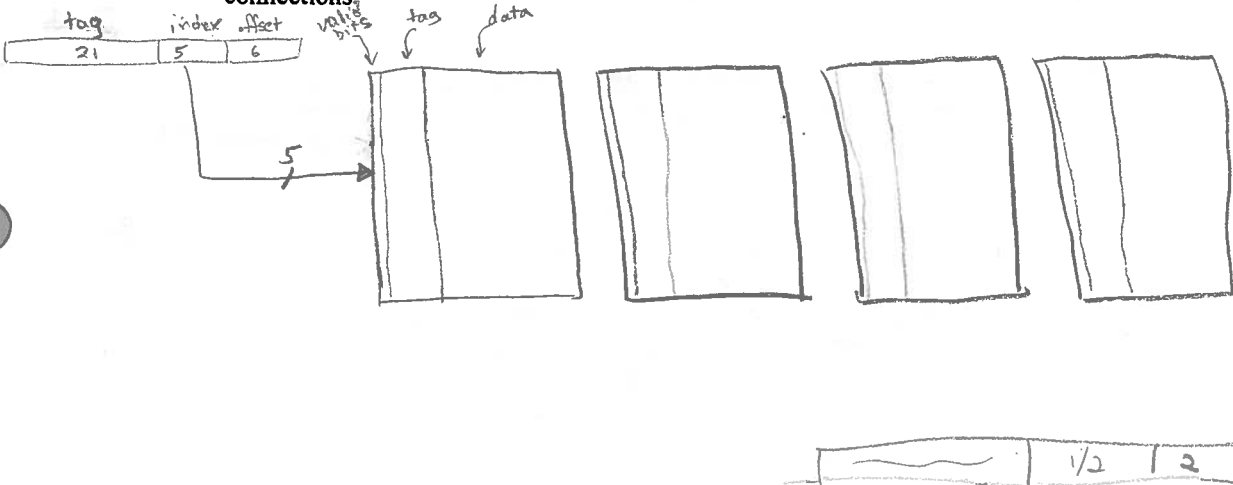
B. Show an address word divided into the fields used to access the cache of part A. Label and dimension each field.



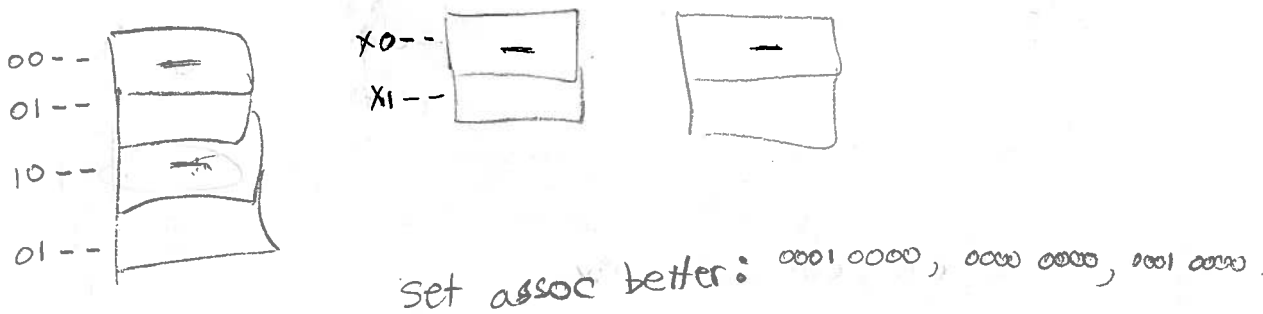
C. Show a directory entry for the cache of part A. Label and dimension each field.



D. Sketch a block diagram of the cache of part A. Show the tag array, data array, all multiplexers or tri-state drivers, and all comparators. Show, label, and dimension (with bit widths) all address and data connections.



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



dm better: 00, 10, A00, B00, 10, A00, B00, 10

Read A0
Read B0
write B0
write B0

Problem 3: Pipeline Microarchitecture [10 points]

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.

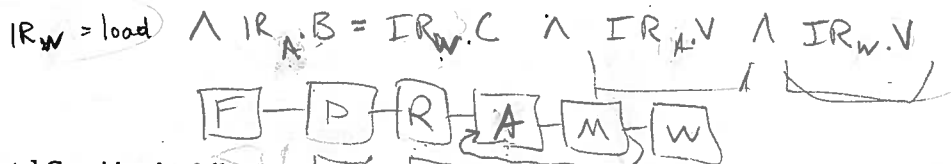
- A. [2 points] What is the latency of an unconditional branch on this machine? Assume no prediction or speculation is employed. (Note: Latency is defined as the number of cycles from when this instruction is executed until the next instruction is executed. E.g., the latency of a simple instruction with no dependencies is one.)



- B. [2 points] What is the latency of a conditional branch? Again assume no speculation about branch direction or distance.

4 cycles

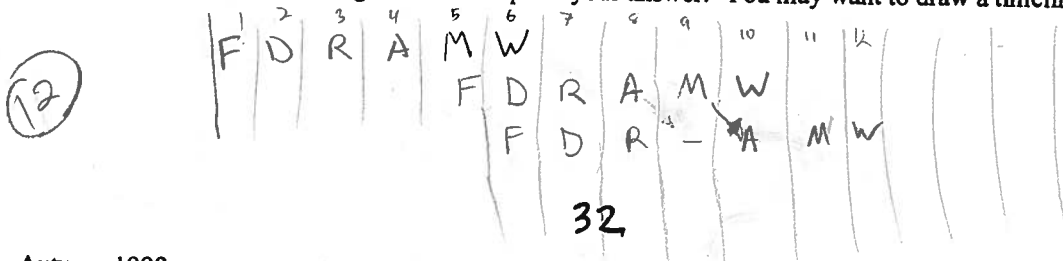
- C. [3 points] The instruction register at each pipeline stage is denoted by IR_{stage} ; for example IR_A is the instruction register at the ALU stage. The source and destination fields of the IR are denoted $IR.A$ (source 1), $IR.B$ (source 2), and $IR.C$ (the destination). For example, the destination 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 second (B) input of the A stage should be activated.



- D. [3 points] Consider the following instruction sequence:

Brz	r5, FUBAR
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 4: Virtual Memory [12 points]

Consider a hypothetical machine with one-byte virtual addresses consisting of a 4-bit page field and a 4-bit offset field. The page field of a virtual address references a page table stored in page 0 of physical memory. Each entry of the page table is either the index of the page frame in physical memory that contains the page in question or the constant FF if the page is not in physical memory. At a given point in time the page 0 of physical memory has the following entries (all numbers are in hexadecimal):

page 0:	0: 07	8: 06
	1: 01	9: 02
	2: 03	A: 04
	3: FF	B: 05
	4: FF	C: FF
	5: FF	D: FF
	6: FF	E: FF
	7: FF	F: 07

- A. [3 points] What physical address, if any, corresponds to virtual address 84 (hex)?

64

- B. [3 points] What virtual address, if any, corresponds to physical address 47 (hex)?

A7

- C. [3 points] Is virtual address 47 (hex) in main memory?

no

- D. [3 points] Is the mapping from virtual addresses to physical addresses one-to-one? Explain your answer?

no virtual addresses

00 and F0 refer to
the same physical address

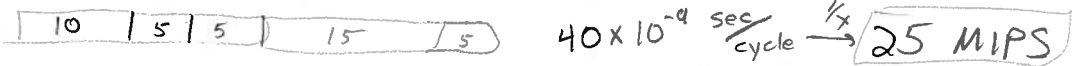
Problem 5: Performance [11 points]

Consider an unpipelined microprocessor that executes all instructions in a single 40ns clock cycle. The cycle is broken down into the following steps

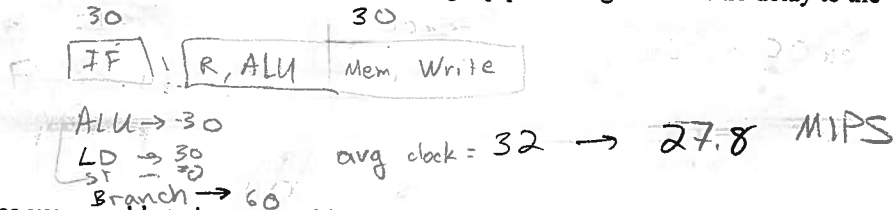
- 10ns Instruction fetch (read instruction referenced by IP)
- 5ns Register read (fetch source operands of instruction)
- 5ns ALU operations (execute ALU instructions, resolve branches)
- 15ns Memory (read or write data memory)
- 5ns Register write (write result to register file)

Suppose this machine executes an instruction mix that is 50% ALU operations, 20% loads, 10% stores, and 20% branches. Further assume that all instructions depend on the immediately preceding instruction and that the register file cannot be read and written simultaneously.

A. [3 points] What is the performance of this baseline machine in MIPS?



B. [4 points] Suppose you are able to insert a single pipeline stage into this microprocessor without adding bypass paths or branch prediction. Where would you insert the pipeline stage (between which two of the above steps) to get the maximum performance benefit? What is the performance of the machine with this two-stage pipeline? Assume that adding a pipeline register adds no delay to the numbers above.



C. [4 points] Suppose you are able to insert an arbitrary number of pipeline stages into this microprocessor. How many stages would you insert? and where would you insert these stages (between which steps above) to get the maximum performance? What is the performance of this configuration?

