Computer Science Department
Stanford University
Comprehensive Examination in Computer Architecture
Autumn 1994

October 19, 1994

READ THIS FIRST!

1. You should write your answers for this part of the Comprehensive Examination in a BLUE BOOK. Be sure to write your MAGIC NUMBER on the cover of every blue book that you use.

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.

SOLUTIONS:
Comp. Architecture

Oct 19, 1994

Question 1.0 Short Answer (10 points)

closed book
partial credit

1.a (3 points)
If you were comparing the performance of two computers, how would you summarize the performance each computer on a set of floating point benchmarks if the performance is reported in MFLOPS? Justify your answer.

You should use harmonic mean because MFLOPS is a rate. A harmonic mean of rates will show the same relative performance of the two computers as total execution time.

1.b (3 points)
What advantage does a branch target buffer (BTB) have over a branch history table (BHT)? What disadvantage does it have?

Advantage: The BTB has a zero branch delay when the prediction is correct. The BHT has at least a one cycle branch delay.
Disadvantage: The BTB requires far more silicon area for the same number of branch instruction slots because it must store the target address along with the prediction bits. Furthermore, the area required by a BHT could be drastically decreased if the tag area were eliminated. This is not possible for a BTB since two branches might map to the same slot, yet they may not share the target instruction address.

1.c (4 points)
Some designers have advocated using a small fully associative on-chip "victim cache" to hold the lines that are replaced from a direct-mapped on-chip primary cache. The victim cache is checked at the same time as the primary cache. If there is a miss in the primary cache and the line is found in the victim cache, the primary cache line and victim cache line are swapped. If the line is not found in the victim cache, it must be fetched form the next level of the memory hierarchy. Using what you know about cache access times and the 3 C's (compulsory misses, capacity misses, conflict misses) model, explain why the victim cache is a good idea.

For a given cache size a direct mapped cache has the lowest access time but the highest number of conflict misses. In certain pathological situations were consecutive accesses are to the same set, conflict misses will result in very high miss rates. The victim cache makes it possible to satisfy these conflict misses without going off chip. A victim cache has the effect of adding associativity to a direct mapped cache with out increasing the access time.

Question 2.0 Pipelining (30 points)

A dual issue superscalar processor implementation of the DLX ISA can execute up to two instructions per cycle as long as there are no dependencies between the instructions. Here is the pipeline for a dual issue processor.

Pipeline

Stage	Function
IF	instruction fetch
ID	decode, register fetch, PC-target, dependency check
ADDR	memory address generate, branch condition
EX/MEM	ALU operation, memory access
WB	writeback

You will use the following information to evaluate this processor. The frequencies in these tables are presented as percentages of all instructions executed.

Branch statistics

Branch type	Frequency of occurrence	Frequency of correct prediction
Taken	10%	9%
Not taken	7%	5%

Some DLX instruction pair frequencies

Type	Instruction sequence	Frequency
1	`ALU op Rx, -, -` `ALU op -, -, Rx or -, Rx, -`	10%
2	`ALU op Rx, -, -` `Store Rx, -(Rb)`	5%
3	`ALU op Rx, -, -` `Load/Store -, -(Rx)`	5%
4	`ALU op Rx, -, -` `JumpRegister Rx`	1%
5	`ALU op Rx, -, -` `Branch Rx`	2%
6	`Load Rx, -(-)` `ALU op -, -, Rx or -, Rx, -`	15%
7	`Load Rx, -(Rb)` `Load/Store -, -(Rx)`	3%
8	`Load Rx, -(Rb)` `Branch Rx, -`	2%
9	`Load Rx, -, -` `JumpRegister Rx`	1%

2.a (5 points)
How many adders, register file ports, and data memory ports does this processor need to minimize structural hazards?

6 Adders

IF	1	sequential instruction address
ID	1	branch target address
ADDR	2	memory addresses
EX	2	ALUs

6 Register File Ports

4 read ports
2 write ports

2 Data memory Ports

2 loads or two stores in the same instruction pair

2.b (8 points)
What is the minimum number of register specifier comparators required to implement dependency checks and forwarding in this processor? Show your reasoning.

All instructions that compute results have at most one destination register. Instructions can have up to two source registers.

Dependency Check

Need to make sure that second instruction does not depend on the first. We must compare two source registers of the second instruction against the destination of the first instruction. This requires **2** comparators.

Forwarding

Source -> destination

EX/MEM -> EX/MEM

Need 2 comparators for each register destination of the source instruction. We have two pipes so we need 4 comparators per source instruction. We have two sources so we need a total of **8** comparators.

EX/MEM -> ADDR

Each destination instruction uses one source register so we need **4** comparators

EX/MEM -> ID

Jump register only uses one source register so we need **4** comparators

Total # of comparators = 2 + 8 + 4 + 4 = 18

2.c (1 point).
What is the ideal CPI of this processor with an ideal memory system?

If we can execute two instructions every cycle, the ideal CPI of this processor is 0.5

2.d (6 points)
Assume the processor uses squashing branches with static branch prediction which is encoded in the opcode of the branch. What is the CPI due to control hazards? Make all reasonable assumptions to minimize CPI. Assume that the branch is the second instruction in the pair.

Here are the cases to consider:

Branch Prediction	Branch outcome	penalty (cycles)	frequency
taken	taken	1	9%
taken	not taken	2	1%
not taken	taken	2	2%
not taken	not taken	0	5%

CPI due to control hazards = 0.09*1 + 0.01*2 + 0.02*2 = 0.15

2.e (4 points)
Using the instruction sequence table, what is the CPI due to instructions that cannot be issued in pairs because of register dependencies? Assume the worst case.

In the worst case none of the instruction pairs in the table can execute together. Thus all pairs will have a CPI of 1. This is 0.5 more than the ideal CPI, thus
Extra CPI = 0.5(0.10 + 0.05 + 0.05 + 0.01 + 0.02 + 0.15 + 0.03 + 0.02 + 0.01) = 0.22

2.f (5 points)
Using the instruction sequence table, what is the CPI due to instruction sequences that cause stalls on this processor?

Stalls are caused when the ADDR or ID stages use a result generated in the EX/MEM stage. Here are the cases
Types 3, 5, 7, and 8 cause a 1 stall cycle
Types 4 and 9 cause a 2 stall cycles

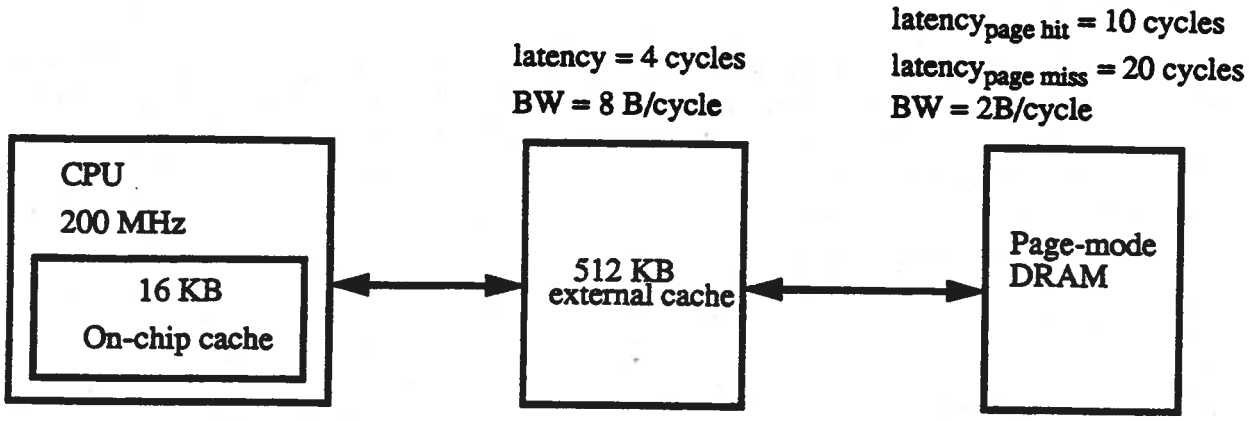
Stall CPI = 0.05*1 + 0.02*1 + 0.03*1 + 0.02*1 + 0.01*2 + 0.01*2 = 0.16

2.g (1 point)
What is the real CPI of this processor with an ideal memory system?

Add up CPI from parts c, d, e, and f.
real CPI = 0.5 + 0.15 + 0.22 + 0.16 = 1.03

Question 3.0 Memory System Design (20 points)

You are considering external cache option for a new high-performance workstation that will use a 200 MHz CPU with an on-chip cache (1 cycle access, direct-mapped). The system is shown below:



After running a set of real programs, you have come up with the following **global** miss ratios:

line size (bytes)	16 KB direct-mapped	512 KB direct-mapped	512 KB 2-way set associative
8	0.17	0.025	0.021
16	0.13	0.018	0.015
32	0.11	0.015	0.012
64	0.08	0.01	0.008

Other necessary information:

1. The probability of a page hit in the page-mode DRAM is 0.3.
2. Making the 512 KB external cache 2-way set associative adds one cycle to its latency but does not affect its bandwidth.

3.a (14 points)
Find the line size and set-associativity that minimizes AMAT measured in CPU cycles for the 512 KB cache. To guarantee multilevel inclusion, the line sizes of the 16 KB cache and 512KB cache must be equal.

Here is the AMAT expressio for the direct-mapped case

$$\text{AMAT} = 1 + \text{MR}_{16K} \times \left(4 + \frac{L}{8}\right) + \text{MR}_{512K} \times \left(10\,(0.3) + 20\,(0.7) + \frac{L}{2}\right)$$

line size	AMAT 256 KB 1-way	AMAT 256 KB 2-way
8	2.38	2.46
16	2.23	2.29
32	2.37	-
64	-	-

A direct mapped cache with a line size of 16 bytes has the lowest AMAT increasing the line size or making external cache set-associative will not improve performance. The - means no need to calculate.

3.b (7 points)
Assume that excluding the memory system the CPI of the processor is 1.4 and that there are 0.4 data references per instruction. How long will it take to execute one million instructions on this processor?

First calculate CPI

$$\text{CPI} = \text{CPI}_{\text{CPU}} + \text{CPI}_{\text{memory}}$$

$$= 1.4 + \left(\frac{\text{Instruction refs} + \text{Data refs}}{\text{\# of instructions}}\right) \times (\text{AMAT} - 1)$$

$$\text{CPI} = 1.4 + 1.4\,(2.23 - 1) = 3.12$$

Now calculate CPU time

$$\text{CPU time} = \text{Instruction count} \times \text{CPI} \times 1/\text{clock freq.}$$

$$= 1\text{M} \times 3.12 \times 1/200\text{MHz}$$

$$= 0.0156 \text{ seconds}$$