Name: Long Nguyen

Student ID: 1957 8983

EEL 4713: Computer Architecture - Spring 2012
Midterm 2 - 120 points possible

10

1. **(30 points total)** Architectures:

Consider two pipelined implementations of the MIPS: P1 and P2. Assume there is a split level-one cache, with perfect (100%) hit rates.

- In P1, there are five stages (IF, ID, EX, MEM, WB) as discussed in class and in your lab. The clock rate is 2GHz, and the hit latency for both the instruction and the data caches is 1 cycle.
- In P2, the clock rate is increased to 3.0GHz but the number of pipeline stages is increased to 6: IF, ID, EX, MEM1, MEM2, WB. The MEM stage is divided into two separate stages: in the first stage (MEM1) the address for the memory access is used to index the cache and compare tags; in the second stage (MEM2), the data is retrieved from the cache (for a load) or stored in the cache (for a store).

A) (5 points) What is the best-case throughput (in instructions per second) that each pipeline can achieve?

$P_1 = 2 \times 10^9$

$P_2 = 3 \times 10^9$

5

B) (5 points) What is the worst-case throughput (in instructions per second) for each pipeline? Assume there are no stalls due to branches.

$P_1 = 2 \times 10^9$

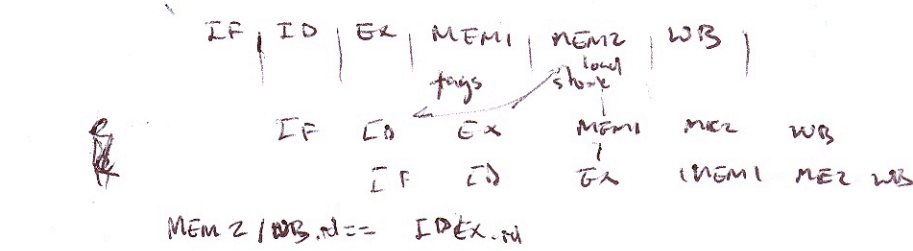
$P_2 = 3 \times 10^9$

0

C) **(20 points)** Describe what additional hazards need to be detected in the P2 pipeline (with respect to P1). Describe which ones can be dealt with by forwarding, and which ones cause the pipeline to stall. (Consider only register-register operations, loads, and stores – don't worry about branches).

For each hazard you specify, write down the comparison that is needed to detect the hazard condition (as done in the book (e.g., IF/ID.rs == ID/EX.rd) and explain which pipeline registers need to be stalled under this condition, or what is the source and destination in the datapath for forwarding.

IF ID EX MEM WB



MEM 2 / WB.rd == ID/EX.rd

MEM2/WB.rd = ID/EX.rd forward MEM 2 To EX/ID, rs

MEM 2 /WB.rd = = IF/ID.rd Stall IF

MEM 2/WB .rd == ~~ID/EX~~. forward rd to rs
EX/MEM. rs

R L S

R

L

S

2. **(30 points total)** Caches:

A) **(6 points)** List the 3 Cs for cache miss classification.

content, coherence, conflict, clock

B) **(4 points)** A write buffer improves the performance of which cache write policy?

write misses?

C) **(10 points)** Given a two level cache hierarchy where the level one cache has a 1 cycle hit latency and 10 cycle miss penalty and the level two cache has a 50 cycle miss penalty, calculate the average memory access time if the level one cache hit rate is 10% and the level two cache hit rate is 50%.

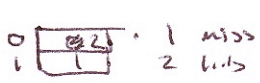
35 cycles

how? I can't give you partial credit

D) **(10 points)** Consider two level-one data caches; both are 2 bytes in size, 1 byte per block. Cache C1 is direct-mapped, cache C2 is 2-way set associative with LRU replacement. Suppose a program issues load instructions that access bytes of memory in the following sequence: 0, 1, 2, 0, 1, 2, 0, 1, 2, 0, ... for a large number of iterations. What is the approximate hit rate for C1 and C2?

C1 33%

C2 0%



3. **(25 points total)** Queuing Theory:

A) **(5 points)** What is the disk (server) utilization if a disk gets 80 I/O requests per second and the average disk service time is 10 ms?

server util = (0.08 req/s)(10 ms)

= 0.8 ms

80%

B) **(10 points)** What is the average system response time for the disk in part A)

$$\frac{(0.8)(10\text{ms})}{(1-0.5)} = 40\text{ms} + 10$$

C) **(10 points)** What arrival rate in I/O requests per second would cause the system to saturate (server utilization > 100%)?

above 100 I/O requests would make server util > 100%

80 req/s

0.08 ms

server response = server util * / (1 − server util)

server util = disk time * requests

4. **(20 points total)** Processor Evaluation:

Assume processors A and B have the following instruction mixes and cycles per instruction type:

Instruction type	Processor A		Processor B	
	Mix	Cycles	Mix	Cycles
Load/store	10%	5	40%	5
Branch	10%	3	10%	3
ALU	80%	1	60%	1

a) **(6 points)** Calculate the CPI for each processor:

A = (5(10%) + 3(10%) + 1(80%)) = 1.6

B = (5(40%) + 3(10%) + 1(60%)) = 2.9

b) **(6 points)** How much faster is processor A as compared to processor B?

almost twice

difference is 1.3

2.9/1.6 = 1.8 · A is 1.8 times faster

c) **(8 points)** Suppose you want to make processor B faster than processor A and you choose to optimize the load/store operation. The optimization would reduce the number of cycles required by load/store operations by a whole number amount (i.e., 1, 2, 3, or 4 cycles instead of 5). If you only want to optimize load/store until the point that processor B is faster than processor A, is it possible to make processor B faster than processor A? If so, how many cycles will the optimized load/store require? Show your work to support your answer.

B = (1(40%) + 3(10%) + 1(60%)) = 1.3

B = (2(40%) + 3(10%) + 1(60%)) = 1.7

Load store cycle would have to be 1 in order to be faster than A. only if processor A ~~changes~~ does not change load/store cycles.

~~values~~

load time	A	B
1	1.2	1.3
2	1.3	1.7
3	1.4	2.1
4	1.5	2.5
5	1.6	2.9

3

5. **(15 points total)** Virtual memory:

A) **(5 points)** List two advantages that virtual memory provides.

decreases load/store times
no need to add hardware

0

B) **(5 points)** Page faults are typically handled by software routines instead of using dedicated hardware. however software is orders of magnitude slower than hardware. Explain why a software routine is still appropriate to handle page faults.

amount of hardware needed ~~[illegible]~~ offsets the speed. software is still cheaper.

3

C) **(5 points)** Since virtual addresses must be translated into physical addresses before indexing into a physically-indexed physically-tagged cache, this translation is on the critical path. What fundamental structure reduces the average translation time?

decoder

0

BONUS QUESTION (5 EXTRA CREDIT POINTS):

What is the 4th cache miss C? (I said I wouldn't "test" you on this, but if you were paying attention in the review session you should get this)

5

coherance

HAVE A NICE SUMMER!