Name: Zack Smaridge

Student ID: 8517 1587

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

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?

Best Case

P1 TP = $\frac{1}{2GHz}$ = 5 ns

P2 TP = $\frac{1}{3GHz}$ = 3.3 ns

4.5

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

P1 Worst case - Stall every other instruction for loads

P1 TP = 1 ns + 50% (5 ns) = 7.25 ns

4.5

P2 worst case - Stall 2 cycles every other instruction for loads

P2 TP = 3.3 ns + 50% (6.6 ns) = 6.6 ns

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).

20

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 MEM1 MEM2 WB

Needed Extra

```
if(regwrite_MEM2=1){
  if(rr0=rw_MEM2)
  if(rr1=rw_MEM2)
}
```

In P1 we must check for RAW hazards up to three cycles later. For P2 we need to extend this an extra cycle. Therefore in the ID phase we will check to see if the register write enable is active in EX, MEM1, MEM2, or WB. If they are we will then check to see if rr0 or rr1 is equal to rw in any of those stages. The forwarding is the same except we will need extra mux space for data coming from the extra mem2 phase.

```
if(dataload_MEM1=1){
  if(rr0=rw_MEM1)
  if(rr1=rw_MEM1)
}
```

Now for loads we usually check one cycle after for hazards that may not be forwarded. we must extend this another stage back such that we check EX and Mem1. The stall will have to be extend to two clock cycles as well. This detection will work by comparing rr1 and rr0 from ID with rw from EX and MEM1 when there is a load in EX or MEM1.

109/120

30

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

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

Compolsory (cold start), Conflict (different data same location), Capacity (not enough space)

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

Write through – shortens time to write to lower level memory

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%.

Avg Access time = $1(.1) + .9(10 + .5(50)) = 31.6$ cycles

↑ L1 hit ↑ L1 miss L2 hit ↑ L1 miss L2 miss

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 C2

C1 – 33%

C2 – 0%

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

22

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?

5

Utilization = (80 IO/sec)(10 ms) = .8 = 80%

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

$T_{sys} = T_{ser} + T_{que}$ $T_{que} = \text{arrival} \cdot \frac{\text{util}}{1-\text{util}} = 80 \cdot \frac{.8}{1-.8} = 320$ sec

7

$T_{sys} = 10\text{ms} + 320\text{sec} = 320.01$ sec

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

10

Utilization = 100% = (x IO/sec)(10 ms)

x = 100 IO/sec

Saturation occurs at arrival rate of 100 IO/sec or greater.

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:

CPI A = .1(5) + .1(3) + .8(1) = 1.6

CPI B = .4(5) + .1(3) + .6(1) = 2.9

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

Performance = CPI · IC · CLK

Impossible to compare w/o knowing clk rates. by how much?

If we assume they're equal then processor A is 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.

Assuming CLKA = CLKB

For B to be faster than A, the CPI must be smaller.

$$CPI\ A \geq CPI\ B$$

$$1.6 \geq .4(x) + .1(3) + .6(1)$$

$$.4x = .7$$

$$x = 1.75$$

Yes

For B to be faster, the load/store operation must occur in 1 clk cycle.

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

15

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

Provides the illusion of a very large memory space for programers to utilize and it is taken care of mostly in software so there is little extra timing expense. Also allows for large applications.

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.

Page faults occur in virtual memory and because the disk is so slow it makes no difference if it is handled in software or hardware, timing wise. So to save more hardware we handle it in software.

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?

Translation lookaside buffer (TLB) - this is a cache that stores address so that we may look them up faster.

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)

Forget name but I believe it occurs in parallel programing when two processors are trying to work on the same piece of data in the cache.

Yes but need name!!

HAVE A NICE SUMMER!