CSE 490/590 - Midterm - March 15 2023 - SOLUTIONS Version B

1. [12 points] Consider three different processors P1, P2, and P3 executing the same instruction set. P1 has a 2.4 GHz clock rate and a CPI of 3.0. P2 has a 1.5 GHz clock rate and a CPI of 2.0. P3 has a 3.0 GHz clock rate and a CPI of 2.4 Which processor has the highest performance expressed in instructions per second? (1 GHz = 10^9 Hz) **Show your steps.**

P1: $\frac{2.4*10^9 cycles}{1 second} * \frac{1 instruction}{3.0 cycles} = 0.8 * 10^9 instructions\ per\ second$

P2: $\frac{1.5*10^9 cycles}{1 second} * \frac{1 instruction}{2.0 cycles} = 0.75 * 10^9 instructions\ per\ second$

P3: $\frac{3.0*10^9 cycles}{1 second} * \frac{1 instruction}{2.4 cycles} = 1.25 * 10^9 instructions\ per\ second$ -- P3 has the highest performance.

2 [12 points] Consider the following MIPS instruction sequence without forwarding:

```
add $s0, $t4, $t4
lw $t3, 0($s1)
sub $t2, $s0, $t5
addi $s1, $s1, -4
sw $t1, $t2, $s2
```

Find the data hazards for the above instruction sequence and explain why it creates such a hazard.

Register $s0 in the 1st and 3rd instructions causes a RAW hazard because in a pipelined processor the 'sub' instruction will read from $s0 before the 'add' instruction can write to $s0, thus 'sub' will get the wrong value from $s0.

Register $t2 in the 3rd and 5th instructions causes a RAW hazard because in a pipelined processor the 'sw' instruction will read from $t2 before the 'sub' instruction can write to $t2, thus 'sw' will get the wrong value from $t2.

Note: If you also listed correct WAR and/or WAW hazards that is fine.

3. [12 points] Given a processor with the following properties:

	Instruction Cache	**Data Cache**
Miss Rate	5%	10%
Miss Penalty	8 cycles	12 cycles

The CPI with ideal cache (no misses) is 3

Given a program with 20% of its instructions being load/store instructions:

a. Compute the actual CPI.

$3 + (0.05 * 8) + 0.20 * (0.10 * 12) = 3.64$

b. If the datapath is improved such that the ideal CPI is reduced from 3 to 2 (all other properties remain the same), compute the actual CPI and compare it with the CPI in part (a).

$2 + (0.05 * 8) + 0.20 * (0.10 * 12) = 2.64$

4. [12 points] Consider three levels of cache backed by DRAM:
   - Access time for L1 cache is 1 cycle and miss rate is 65%
   - Access time for L2 cache is 12 cycles and miss rate is 8%
   - Access time for L3 cache is 75 cycles and miss rate is 1%
   - Access time for DRAM is 90 cycles and miss rate is 0% (no misses)

   Calculate Average Memory Access Time (AMAT). **Show the steps.**

   $AMAT = Average\ Memory\ Acess\ Time = Hit\ Time + Miss\ Rate * Miss\ Penalty$

   $1 + 0.65 * (12 + 0.08 * (75 + 0.01 * (90))) =$ 12.7468

   --- OR ---

   $L1_{AMAT} = 1 + 0.65 * L2_{AMAT} = 1 + 0.65 * 18.072 =$12.7468 = Total AMAT

   $L2_{AMAT} = 12 + 0.08 * L3_{AMAT} = 12 + 0.08 * 75.9 = 18.072$

   $L3_{AMAT} = 75 + 0.01 * DRAM_{AMAT} = 75 + 0.01 * 90 = 75.9$

   $DRAM_{AMAT} = 90$

5. [12 Points] Given a reference (pointer) size of 32 bits and given a cache that is 128kB (2^{17} bytes) in size and a block size of 32 bytes (2^5 bytes), calculate the following and **show your work**:

   a. The number of blocks in the cache.

   $2^{17}/2^5 = 2^{12} = 4096\ blocks$

   b. If using a direct-mapped cache, the number of index bits and tag bits.

   $2^{12} blocks$ —>12 index bits

   $32 - 12 =$ 20 tag bits

   c. If using an 8-way set associative cache, the number of index bits.

   8-way set associative —> 8 blocks per set

   $4096\ blocks$ / $8\ blocks = 2^{12}\ blocks$ / $2^3\ blocks = 2^9\ sets$

   $2^9\ sets$ —> 9 index bits

6. [14 Points]

   a. Write through and write back are approaches used when a cache write hit is encountered. Explain how write through and write back work. Highlight pros and cons of each of these approaches.

   For the write back approach the information is written only to the block in the cache. The modified cache block is written to main memory only when it is replaced.

   For the write through approach the information is written to both the block in the cache and to the block in the lower-level memory.

   Write Back:
   - Pros: Low latency and high throughput for write-intensive applications.
   - Cons: There is data availability risk because the cache could fail (and so suffer from data loss) before the data is persisted to the backing store. This results in the data being lost.

Write Through:

- Pros: Ensures fast retrieval while making sure the data is in the backing store and is not lost on case the cache is disrupted.
- Cons: Writing data will experience latency as you have to write to two places every time.

b. Write allocate and No-write allocate are approaches used when a cache write miss is encountered. Explain how write allocate and no-write allocate work. Highlight pros and cons of each of these approaches.
The write allocate approach fetches the missing block of cache from memory and then updates the appropriate word(s) of the block of cache.

The no-write allocate approach skips the cache write. Instead it simply invalidates the missing cache block and writes the data to the write buffer. The write buffer will eventually write the data back to the next lower memory level.

7. [12 Points] Assume the time for stages is:
- 120ps for register read or write
- 250ps for other stages

Compare the pipelined datapath with the single-cycle datapath for the following instruction sequence:

```
lw $1, 100($0)
lw $2, 200($0)
lw $3, 300($0)
```

The following table provides how much time is spent in each stage by a specific instruction:

Instruction	**Instruction Fetch**	**Register Read**	**ALU Operation**	**Memory Access**	**Register Write**	**Total Time**
lw	250ps	120ps	250ps	250ps	120ps	990ps
sw	250ps	120ps	250ps	250ps		870ps
R-format	250ps	120ps	250ps		120ps	740ps
beq	250ps	120ps	250ps			620ps

The time taken to execute the instructions in the single cycle datapath is the sum of the total time needed for each instruction. This is because each instruction executes one after the other. Assuming each lw instruction takes 990ps, the total time for the given instruction sequence is $990\,ps * 3 = 2970\,ps$.

The time taken to execute the instructions in the pipelined datapath depends on the instruction and stage with the longest time to execute, since every cycle in a pipelined processor must take the same amount of time to execute. In the question above, this is 250 ps. With pipeling the above instruction sequence will take 7 cycles to complete execution. Since each cycles is 250 ps, the total time to execute is $250\,ps * 7 = 1750\,ps$.

Thus, the pipelined processor will execute the given instruction sequence quicker than the single-cycle processor.

8. [14 Points]

   a. Schedule the following instruction sequence for dual-issue MIPS. Consider one ALU/branch instruction and one load/store instruction can be executed in parallel when there is no data dependencies:

```
Loop: lw $t7, 0($t2)       ;$t7=array element
      add $t7, $t7, $t3    ;add scalar in $t3
      sw $t7, 0($s4)       ;store result
      addi $s4, $s4,-4     ;decrement pointer
      add $s3, $t2, $s3    ;update $s3
      bne $s4, $zero, Loop ;branch $s4!=0
```

ALU / Branch	**Load / Store**	**Cycle**
`add $s3, $t2, $s3`	`lw $t7, 0($t2)`	1
`addi $s4, $s4, -4`	`nop`	2
`add $t7, $t7, $t3`	`nop`	3
`bne $s4, $zero, Loop`	`sw $t7, 4($s4)`	4
		5
		6

   b. Compute the IPC in part (a)

   6 / 4 = 1.5 instructions per cycle