CSE240A Midterm

Xinhao Luo

TOTAL POINTS

43.5 / 45

QUESTION 1

8 pts

1.1 2 / 2

✓ **- 0 pts correct**

- 1 pts incorrect result, correct calculation

- 2 pts incorrect approach

1.2 2 / 2

✓ **- 0 pts correct**

- 1 pts incorrect answer, correct approach

- 2 pts incorrect approach

1.3 3 / 4

- 0 pts Correct

✓ **- 1 pts Incorrect or missing result**

- 3 pts Incorrect or missing major details (CPI_new, CPI_branch, CPI_other, execution time, etc)

- 0.5 pts Incorrect total cycles

- 0.5 pts Incorrect $$CPI_{new}$$

- 0.5 pts Incorrect $$CPI_{other}$$

- 0.5 pts Incorrect ET

- 0.5 pts Incorrect $$Cycle_{others}$$

- 1 pts Missing partial details

+ 2 pts Partial Credits

QUESTION 2

2 6 / 6

✓ **- 0 pts correct**

- 2 pts incorrect result

- 1 pts incorrect/missing CPI equation

- 1 pts incorrect/missing instruction cache miss equation

- 1 pts incorrect/missing data cache miss equation

- 0.5 pts incorrect/missing instruction cache penalty

- 0.5 pts incorrect/missing data cache penalty

QUESTION 3

3 8 / 8

✓ **- 0 pts Correct**

- 2 pts Incorrect A

- 2 pts Incorrect B

- 1 pts Incorrect C.1

- 1 pts Incorrect C.2

- 1 pts Incorrect C.3

- 1 pts Incorrect C.4

QUESTION 4

4 8 / 8

✓ **+ 1.5 pts No of Blocks = 1024**

✓ **+ 1.5 pts No of bits in index = 8**

✓ **+ 1.5 pts No of bits in offset = 6**

✓ **+ 1.5 pts No of bits in tag = 34**

✓ **+ 2 pts Final overhead = 6.8%**

QUESTION 5

5 14.5 / 15

- 0 pts Correct

- 1 pts Single mistake for total cycles

- 1.5 pts Multiple mistakes for total cycles

- 1 pts For each instruction with misplaced pipeline stage

- 1 pts For each instruction with misplaced pipeline stage

- 1 pts For each instruction with misplaced pipeline stage

- 1 pts For each missing or incomplete or extra instruction

✓ **- 0.5 pts For each wrong bypassing arrow (missing or extra**

- 0.5 pts For each wrong bypassing arrow (missing or extra

- 0.5 pts For each wrong bypassing arrow (missing

or extra

- 0.5 pts For each wrong bypassing arrow (missing or extra

- 1 pts For each missing or incomplete or extra instruction

- 1 pts For each missing or incomplete or extra instruction

- 1 pts For each missing or incomplete or extra instruction

CSE 240A: Principles of Computer Architecture

Midterm Examination

Xinhao Luo

Fall 2022

1	8	
2	6	
3	8	
4	8	
5	15	
Total	**45**	

Instructions This exam is open book and open notes. Personal calculators *are* allowed. Show your work and insert your answer in the space(s) provided. **Please provide details on how you reach a result unless directed by the question as not to.**

The exam totals 45 points. This exam counts for 45% of your course grade. Please submit typed answers to the following questions as a PDF via Gradescope by **Monday, November 14, 2022 at 11:59 PM**. It would be great if you can finish the midterm by typing rather than handwriting.

1 Performance Evaluation (8 points)

1A) A program P has an instruction count of 10 billion, an average CPI of 3, and runs on a processor with a clock rate of 2 GHZ. What is the execution time for program P?

$\frac{1\times10^{10}\times3}{2\times10^{9}} = 15seconds$

1B) We have a program with 30 billion instructions that takes 45 seconds to run on a 2GHz machine. It is given that the program consists of 25% branch instructions, and the CPI of branch instructions is 4. What is the average CPI of the program?

Total Cycles

$$45 \times 2 \times 10^9 = 9 \times 10^{10} \tag{1}$$

Average CPI

$$\frac{9 \times 10^{10}}{3 \times 10^{10}} = 3 \tag{2}$$

1C) We use a newly developed compiler to recompile the original program given in **1B**. The recompiled program now uses 20 billion instructions. It is still composed of 25% branch instructions, but the CPI of the branch instructions has been reduced by a factor of 2 (CPI of the other types of instructions remains the same). What is the expected execution time speedup of the new program over the original program (on the same machine)?

1.1 2 / 2

✓ **- 0 pts correct**

- 1 pts incorrect result, correct calculation

- 2 pts incorrect approach

CSE 240A: Principles of Computer Architecture

Midterm Examination

Xinhao Luo

Fall 2022

1	8	
2	6	
3	8	
4	8	
5	15	
Total	**45**	

Instructions This exam is open book and open notes. Personal calculators *are* allowed. Show your work and insert your answer in the space(s) provided. **Please provide details on how you reach a result unless directed by the question as not to.**

The exam totals 45 points. This exam counts for 45% of your course grade. Please submit typed answers to the following questions as a PDF via Gradescope by **Monday, November 14, 2022 at 11:59 PM**. It would be great if you can finish the midterm by typing rather than handwriting.

1 Performance Evaluation (8 points)

1A) A program P has an instruction count of 10 billion, an average CPI of 3, and runs on a processor with a clock rate of 2 GHZ. What is the execution time for program P?

$\frac{1\times10^{10}\times3}{2\times10^{9}} = 15seconds$

1B) We have a program with 30 billion instructions that takes 45 seconds to run on a 2GHz machine. It is given that the program consists of 25% branch instructions, and the CPI of branch instructions is 4. What is the average CPI of the program?

Total Cycles

$$45 \times 2 \times 10^9 = 9 \times 10^{10} \tag{1}$$

Average CPI

$$\frac{9 \times 10^{10}}{3 \times 10^{10}} = 3 \tag{2}$$

1C) We use a newly developed compiler to recompile the original program given in **1B**. The recompiled program now uses 20 billion instructions. It is still composed of 25% branch instructions, but the CPI of the branch instructions has been reduced by a factor of 2 (CPI of the other types of instructions remains the same). What is the expected execution time speedup of the new program over the original program (on the same machine)?

1.2 **2 / 2**

✓ **- 0 pts** **correct**

- 1 pts incorrect answer, correct approach

- 2 pts incorrect approach

Total number of branch instruction cycles:

$$2 \times 10^{10} \times 25\% \times 2 = 1 \times 10^{10} \tag{3}$$

Average CPI of other instructions:

$$\frac{3 - (25\% \times 4)}{1 - 25\%} = \frac{8}{3} \tag{4}$$

Total cycles:

$$\frac{8}{3} \times (1 - 25\%) \times 2 \times 10^{10} + 1 \times 10^{10} = 5 \times 10^{10} \tag{5}$$

Speedup:

$$\frac{9 - 5}{9} * 100 = 44.4\% \tag{6}$$

2 Cache Performance (6 points)

Assuming the base CPI_{base} (no stall) of a pipeline is 1. A program has 25% of load/store instructions. The processor only has one level of cache, i.e., an L1 instruction cache and an L1 data cache. Cache miss rate and penalty are as following:

- L1 instruction cache: $\%_{miss} = 2\%, t_{miss} = 100$ cycles
- L1 data cache: $\%_{miss} = 30\%, t_{miss} = 100$ cycles

What is the CPI?

CPI:

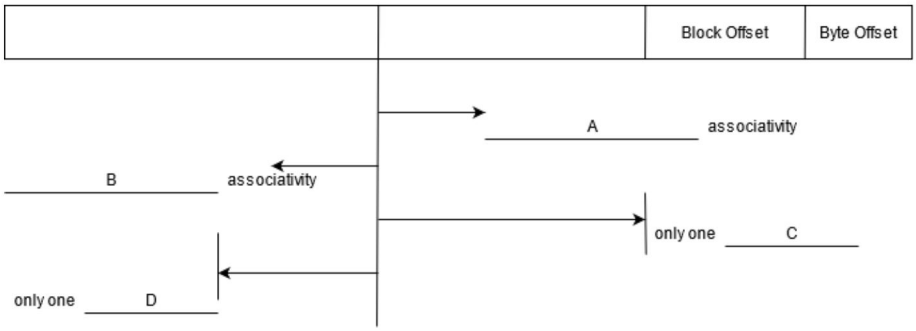
$$1 + (100\% \times 2\% \times 100) + (25\% \times 30\% \times 100) = 1 + 2 + 7.5 = 10.5 \tag{7}$$

3 Caches and Memory Hierarchy

A Allow cache and memory to be inconsistent, i.e., write the data **only** into the cache block. Is this write-back or write through? *write-back* (No further explanation needed)

B Require cache and memory to be consistent, i.e., always write the data into both the cache block and the next level in the memory hierarchy. Is this write-back or write through? *write through* (No further explanation needed)

C Consider the diagram below that shows the dividing line between the bits used for tag compare and those used to select the cache set. Fill in the lines indicating whether the associativity **increases** or **decreases** and whether the resulting cache has only one **way** or only one **set**. (No further explanation needed)



1.3 **3 / 4**

- **0 pts** Correct

✓ **- 1 pts Incorrect or missing result**

- **3 pts** Incorrect or missing major details (CPI_new, CPI_branch, CPI_other, execution time, etc)
- **0.5 pts** Incorrect total cycles
- **0.5 pts** Incorrect $$CPI_{new}$$
- **0.5 pts** Incorrect $$CPI_{other}$$
- **0.5 pts** Incorrect ET
- **0.5 pts** Incorrect $$Cycle_{others}$$
- **1 pts** Missing partial details
- **2 pts** Partial Credits

Total number of branch instruction cycles:

$$2 \times 10^{10} \times 25\% \times 2 = 1 \times 10^{10} \tag{3}$$

Average CPI of other instructions:

$$\frac{3 - (25\% \times 4)}{1 - 25\%} = \frac{8}{3} \tag{4}$$

Total cycles:

$$\frac{8}{3} \times (1 - 25\%) \times 2 \times 10^{10} + 1 \times 10^{10} = 5 \times 10^{10} \tag{5}$$

Speedup:

$$\frac{9 - 5}{9} * 100 = 44.4\% \tag{6}$$

2 Cache Performance (6 points)

Assuming the base CPI_{base} (no stall) of a pipeline is 1. A program has 25% of load/store instructions. The processor only has one level of cache, i.e., an L1 instruction cache and an L1 data cache. Cache miss rate and penalty are as following:

- L1 instruction cache: $\%_{miss} = 2\%, t_{miss} = 100$ cycles
- L1 data cache: $\%_{miss} = 30\%, t_{miss} = 100$ cycles

What is the CPI?

CPI:

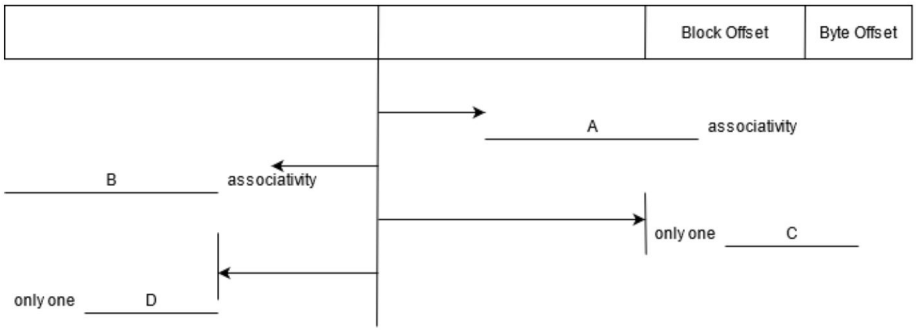
$$1 + (100\% \times 2\% \times 100) + (25\% \times 30\% \times 100) = 1 + 2 + 7.5 = 10.5 \tag{7}$$

3 Caches and Memory Hierarchy

A Allow cache and memory to be inconsistent, i.e., write the data **only** into the cache block. Is this write-back or write through? *write-back* (No further explanation needed)

B Require cache and memory to be consistent, i.e., always write the data into both the cache block and the next level in the memory hierarchy. Is this write-back or write through? *write through* (No further explanation needed)

C Consider the diagram below that shows the dividing line between the bits used for tag compare and those used to select the cache set. Fill in the lines indicating whether the associativity **increases** or **decreases** and whether the resulting cache has only one **way** or only one **set**. (No further explanation needed)



2 6 / 6

✓ **- 0 pts correct**

- 2 pts incorrect result

- 1 pts incorrect/missing CPI equation

- 1 pts incorrect/missing instruction cache miss equation

- 1 pts incorrect/missing data cache miss equation

- 0.5 pts incorrect/missing instruction cache penalty

- 0.5 pts incorrect/missing data cache penalty

Please fill your answers to the corresponding blanks below:
A: *increase*
B: *decrease*
C: *set*
D: *way*

4 Cache tag overhead (8 points)

Assume:

- A processor has a 64KB 4-way set associative cache
- The cache access uses physical addresses only
- A physical address is 48 bits long
- Each block holds 64 bytes of data
- Tag overhead includes the valid bit and tag bits

How much is the tag overhead in percent?

Cache Block size: $log_2(64) = 6$
Block number: $64 \times 1024/64 = 1024$
Set number: $\frac{1024}{4} = 256$
Index bit: $\log_2(256) = 8$
Tag bit = Address Length - offset - index $= 48 - 6 - 8 = 34$ bit
Overhead = $\frac{tagbits+validbit}{cachesize} = \frac{34+1}{64\times8} = 6.836\%$

5 Pipelining (15 points)

Assume the following program is running on the 5-stage in-order pipeline processor shown in class. All registers are initialized to 0. Assuming only WX and WD (register file internal forwarding) forwarding, branches are resolved in **Decode** stage, and branches are always predicted **not-taken**. How many cycles will it take to execute the program, if the branch outcome is **actually taken**? Draw a pipeline diagram (table) to show the details of your work. Use arrows to indicate forwarding.

```
      lw $r6 0($r10)
      lw $r7 0($r11)
      add $r2 $r6 $r7
      beq $r2 $r3 label
      sub $r6 $r8 $r4
      sw $r6 0($r10)
label:lw $r1 0($r2)
      or $r4 $r2 $r1
```

The instruction would take 15 cycles to execute.

3 **8 / 8**

✓ **- 0 pts Correct**

- 2 pts Incorrect A

- 2 pts Incorrect B

- 1 pts Incorrect C.1

- 1 pts Incorrect C.2

- 1 pts Incorrect C.3

- 1 pts Incorrect C.4

Please fill your answers to the corresponding blanks below:
A: *increase*
B: *decrease*
C: *set*
D: *way*

4 Cache tag overhead (8 points)

Assume:

- A processor has a 64KB 4-way set associative cache
- The cache access uses physical addresses only
- A physical address is 48 bits long
- Each block holds 64 bytes of data
- Tag overhead includes the valid bit and tag bits

How much is the tag overhead in percent?

Cache Block size: $log_2(64) = 6$
Block number: $64 \times 1024/64 = 1024$
Set number: $\frac{1024}{4} = 256$
Index bit: $\log_2(256) = 8$
Tag bit = Address Length - offset - index $= 48 - 6 - 8 = 34$ bit
Overhead = $\frac{tagbits+validbit}{cachesize} = \frac{34+1}{64\times8} = 6.836\%$

5 Pipelining (15 points)

Assume the following program is running on the 5-stage in-order pipeline processor shown in class. All registers are initialized to 0. Assuming only WX and WD (register file internal forwarding) forwarding, branches are resolved in **Decode** stage, and branches are always predicted **not-taken**. How many cycles will it take to execute the program, if the branch outcome is **actually taken**? Draw a pipeline diagram (table) to show the details of your work. Use arrows to indicate forwarding.

```
      lw $r6 0($r10)
      lw $r7 0($r11)
      add $r2 $r6 $r7
      beq $r2 $r3 label
      sub $r6 $r8 $r4
      sw $r6 0($r10)
label:lw $r1 0($r2)
      or $r4 $r2 $r1
```

The instruction would take 15 cycles to execute.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
lw $r6 0($r10)	F	D	X	M	W															
lw $r7 0($r11)		F	D	X	M	W														
add $r2 $r6 $r7			F	D	D	X	M	W												
beq $r2 $r3 label					F	D	D	D	X	M	W									
sub $r6 $r8 $r4								F												
sw $r6 0($r10)																				
label: lw $r1 0($r2)									F	D	X	M	W							
or $r4 $r2 $r1										F	D	D	X	M	W					

4 8 / 8

- ✓ **+ 1.5 pts** No of Blocks = 1024
- ✓ **+ 1.5 pts** No of bits in index = 8
- ✓ **+ 1.5 pts** No of bits in offset = 6
- ✓ **+ 1.5 pts** No of bits in tag = 34
- ✓ **+ 2 pts** Final overhead = 6.8%

5 **14.5 / 15**

- **0 pts** Correct
- **1 pts** Single mistake for total cycles
- **1.5 pts** Multiple mistakes for total cycles
- **1 pts** For each instruction with misplaced pipeline stage
- **1 pts** For each instruction with misplaced pipeline stage
- **1 pts** For each instruction with misplaced pipeline stage
- **1 pts** For each missing or incomplete or extra instruction

✓ **- 0.5 pts For each wrong bypassing arrow (missing or extra**

- **0.5 pts** For each wrong bypassing arrow (missing or extra
- **0.5 pts** For each wrong bypassing arrow (missing or extra
- **0.5 pts** For each wrong bypassing arrow (missing or extra
- **1 pts** For each missing or incomplete or extra instruction
- **1 pts** For each missing or incomplete or extra instruction
- **1 pts** For each missing or incomplete or extra instruction