20F-COMSCIM151B-1 Homework 5

NEVIN LIANG

TOTAL POINTS

60 / 60

QUESTION 1

Pipeline 10 pts

1.1 Pipeline depth 5 / 5

✓ **+ 5 pts Correct**

+ 1 pts Wrong assumptions

+ 3 pts Wrong ratio

+ 0 pts Incorrect

1.2 Branch execution 5 / 5

✓ **- 0 pts Correct**

- 0 pts Wrong ratio

- 2 pts Wrong assumptions

- 5 pts Incorrect

QUESTION 2

2 Address sequence 10 / 10

✓ **- 0 pts Correct**

- 0 pts Used word-aligned byte addresses

- 7.5 pts C2 has a greater or equal miss rate

- 10 pts Incorrect

QUESTION 3

Caches 20 pts

3.1 cache 1 - cycles 5 / 5

✓ **- 0 pts Correct**

- 5 pts Incorrect

- 4 pts Wrong associativity

- 4 pts Wrong number of instructions fetched

3.2 cache 2 - cycles 5 / 5

✓ **- 0 pts Correct**

- 1 pts Wrong associativity

- 5 pts Incorrect

- 1 pts Wrong number of instructions fetched

3.3 cache 3 - cycles 5 / 5

✓ **- 0 pts Correct**

- 1 pts Wrong associativity

- 5 pts Incorrect

- 1 pts Wrong number of instructions fetched

3.4 cpi 5 / 5

✓ **- 0 pts ok**

QUESTION 4

4 Field widths 5 / 5

✓ **- 0 pts Correct**

- 2.5 pts Mismatched

- 5 pts Incorrect

QUESTION 5

Replacement policy 15 pts

5.1 Sequence 5 / 5

✓ **- 0 pts Correct**

- 3 pts Not minimal

5.2 Miss rate 5 / 5

✓ **- 0 pts Correct**

5.3 Random policy 5 / 5

✓ **- 0 pts YES**

✓ **- 0 pts Good enough justification**

- 1 pts Insufficient justification

- 2 pts Not a justification / Incorrect justification

- 3 pts Blank

- 5 pts NO

QUESTION 6

6 Late submission penalty 0 / 0

✓ **- 0 pts On time**

- 12 pts Late submission

Nevin Liang.

Homework Set – V

Session: Fall 2020 **Instructor: Prof. P. Gupta**

Total Points: 60 **Due Date: 12th Nov (before class starts)**

Instructions

1. **There are two sections in this homework set. The questions under Section – I are for your practice. Solutions to these problems will soon be provided on the course web page. Students are encouraged to solve the problems before looking at the solution provided.**
2. **Problems under Section – II are to be submitted.**

Section – I

1. Consider a cache of 48KB, with 16 byte-blocks. For each question below, describe how or why not. Remember that 1K = 1024.

   (a) Can you make it fully associative?

   (b) How about a set-associative cache?

   (c) And a direct-mapped cache?

2. Locality

   a. Describe the general characteristics of a program that would exhibit very little temporal and spatial locality with regard to instruction fetches. Provide an example program (pseudocode is fine).

   b. Describe the general characteristics of a program that would exhibit very high amounts of temporal locality but very little spatial locality with regard to instruction fetches. Provide an example program (pseudocode is fine).

   c. Describe the general characteristics of a program that would exhibit very little temporal locality but very high amounts of spatial locality with regard to instruction fetches. Provide an example program (pseudocode is fine).

3. Why might a compiler perform the following optimization?

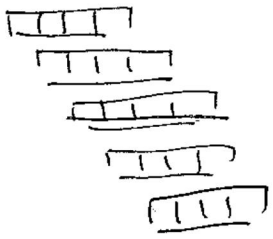
```
/ * Before */

for (j = 0; j < 20; j ++ )
        for ( i = 0; i < 200; i++ )
                x[ i ] [ j ] = x [ i ] [ j ] + 1 ;

/ * After */

for ( i = 0 ; i < 200 ; i ++ )
        for ( j = 0 ; j < 20 ; j ++ )
                x [ i ] [ j ] = x [ i ] [ j ] + 1 ;
```

Section – II

1.

```
        lw $1, 40($6)
        beq $2, $0, Label ;    Assume $2 = $0
        sw $6, 50($2)
Label:  add $2, $3, $4
        sw $3, 50($4)
```

a. For this problem, assume that all branches are perfectly predicted, eliminating all control hazards, and that no delay slots are used. If we change load/store instructions to use a register without an offset as the address, these instructions no longer need to use the ALU. As a result, MEM and EX can be overlapped and the pipeline has only 4 stages. Assuming this change does not affect clock cycle time, what speedup (in percentage) is achieved in this instruction sequence? Show your work in the empty space provided here and write the final answer only in the box. **[5 points]**



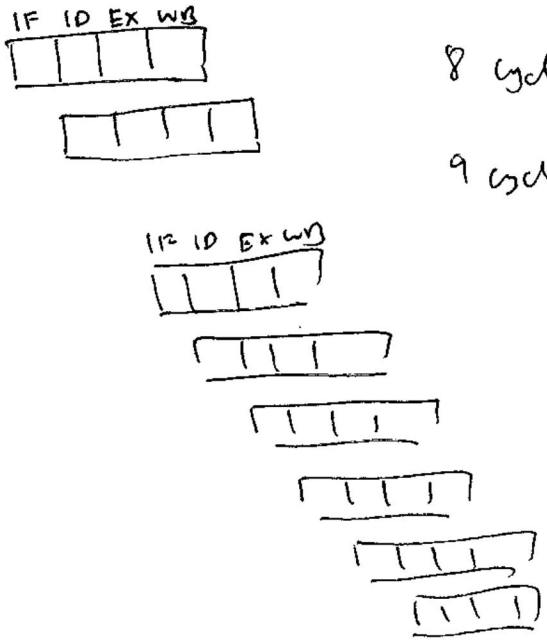
$8 \rightarrow 7$

$\frac{8-7}{8} = 12.5$

8 cycles.

12.5 %

b. Assuming stall on branch and no delay slots, what speedup is achieved on this code if branch outcomes are determined in the ID stage, relative to the execution where branch outcomes are determined in the EX stage? **[5 points]**



8 cycles. advanced scan

9 cycles

11.1%

2. Cache C1 is direct-mapped with 16 one-word blocks. Cache C2 is direct-mapped with 8 two-word blocks. Assume that the miss penalty for C1 is 6 memory bus clock cycles and the miss penalty for C2 is 11 memory bus clock cycles. Assuming that the caches are initially empty, find a series of maximum 5 accesses for which C2 has a lower miss rate but spends equal or more memory bus clock cycles on cache misses than C1. Use word addresses and write down the address for each access. **[10 points]**

C1

1 bit offset

4 bits index

27 bits tag

6 3 miss C1

11 2 miss C2

C2

2 bit offset

3 bit index

27 bit tag

0x0
0x1
0x5

3. Consider three processors with different cache configurations:

   a. Cache 1: Direct-mapped with one-word blocks with instruction miss rate 4%; data miss rate 6%

   b. Cache 2: Direct-mapped with four-word blocks with instruction miss rate 2%; data miss rate 4%

   c. Cache 3: Two-way set associative with four-word blocks with instruction miss rate 2%; data miss rate 3%

For these processors, one-half of the instructions contain a data reference. Assume that the cache miss penalty is `6 + (Block size in words) + 2*Associativity`. Assume that the workload has 1000 instructions. **[5 x 4 = 20 points]**

- Determine the total number of cycles spent by a processor with cache 1. Write the number of cycles only. *If the answer is 10 cycles, write 10 in the box. Follow this instruction for other portions of this question.*

500 instructions have data reference

$0.04 \cdot 1000 = 40$ cycles for instruction

$0.06 \cdot 500 = 30$ cycles for data

Misses = 100

penalty = $6 + 1 + 2 = 9$

$70 \cdot 9 = 630$

$1000 + 630$

1630

- Determine the total number of cycles spent by a processor with cache 2.

$0.02 \cdot 1000 = 20$

$0.04 \cdot 500 = 20$

$40 \cdot (6 + 4 + 2 \cdot 1) = 480$

1480

- Determine the total number of cycles spent by a processor with cache 3.

$0.02 \cdot 1000 = 20$
$0.03 \cdot 500 = 15$
35

$35(6+4+2 \cdot 2) = 35 \cdot 14$
$= 490$

$1000 + 490$

1490

- If the processor with cache 1 has a CPI of 2, what would be the CPI for the processor with cache 3?

$2 - 0.63$

$0.49 + (2 - 0.63) = 1.86$

1.86

4. A 32kB cache has a linesize of 16 bytes and is 4-way set-associative. How many bits of an address will be in the Tag, Index, and Offset? *Only write the number of bits in the boxes provided. E.g., if an answer is **5 bits**, write **5** in the box.* **[5 points]**

cache size = # sets · block size · associativity

$32 \cdot 1024 = S \cdot 16 \cdot 4$

$S = 512$ sets

$\log_2 16 = 4$

$S = 512 \rightarrow i = \log_2 512 = 9$

$64 - 13 = 51$

Tag:	51
Index:	9
Offset:	4

5.

a. In a 2-way set-associative cache, four addresses, A, B, C and D all have the same index but distinct tags. What is a minimum sequence of accesses which, if repeated, will maximize the miss rate in the cache if it uses the LRU replacement policy? **[5 points]**

ABC

b. If the above sequence is repeated for a long period of time, what will the miss rate be if the cache uses an LRU replacement policy? **[5 points]**

A B C Ø A B C Ø A B c Ø A B C Ø ...
M M M M M M

100

c. If the sequence is repeated for a long period of time, will the miss rate be improved (YES or NO)if random is used as the replacement policy i.e., a random block is replaced instead of the least recently used block? Write your justification in the second box. **[5 points]**

Yes.

Justification:

ABC ABC ABC... A B C
it'll always replace A if LRU, but if its random, sometimes it replaces B, then A is still in there and a hit