German University in Cairo - GUC
Faculty of Media Engineering and Technology- MET
Department of Computer Sciences
Assoc. Prof. Milad Ghantous

CSEN702-Microprocessors
Winter Semester 2021

Midterm Exam - Solution

Bar Code

Instructions: **Read Carefully Before Proceeding.**

1- Non-programmable calculators are allowed

2- Write your solutions in the space provided

3- The exam consists of **(XX) questions**

4- This exam booklet contains **(XX) pages** including this page

5- Total time allowed for this exam is **(120) minutes**

6- When you are told that time is up, stop working on the test

Good Luck!

Question	1	2	3	4	5	∑
Possible Marks	10	20	10	30	20	90
Final Marks						

Question 1 (10 pts)

Mark each of the following statement with T or F. Ambiguous letters will not be considered.

#	Statement	T/F
1	The first access to any array element A[i] will result in a compulsory miss.	F
2	Using multi-level caches helps in reducing the miss penalty.	T
3	Way-prediction lowers the average memory access time by lowering the miss rate.	F
4	The size of the blocks in the blocking optimization, employed by the compiled or the programmer, should be less or equal to the cache block size.	F
5	The only limitation of the loop unrolling is the growing code size.	F
6	An output dependence is a WAW hazard.	T
7	For a program with no loops at all, branch prediction will increase the execution time compared with not using prediction at all.	F
8	Leakage in microprocessor chips increases with lower voltages.	F
9	Thermal Design power (TDP) is the maximum heat generated by the chip.	F
10	The global miss rate of L2 cache should, in theory, be always less than that of the L1 cache.	T

Question 2 (20 pts)

Answer all the following short questions.

A) Consider a pipelined processor with 50% ALU operations, 30% branches, and 20% memory operations. It uses a branch prediction scheme for its branch instructions with a 2 cycles penalty. Given that we have 1 stall cycle every 5 instructions due to data hazards and that no structural hazards are present, what's the minimum **accuracy** of the branch prediction scheme to keep the CPI less than or equal to 1.3? (5 pts)

Answer:

CPI = 1+(1/5) + 0.3 X 2 X mispredictionRate <= 1.3
→ *1.2+ 0.6 X mispredictionRate <=1.3*
→ *mispredictionRate should <=(0.1 / 0.6) = 0.16 or 16%*
→ *This means our branches prediction scheme should be correct with 84% accuracy at least.*

B) Consider the following code:

```
for (i=0; i<9; i++)
  for (j=0; j<9; j++)
    {
      A[j][i]++;
    }
```

Given that A is a double-precision floating point array of size 10x10, and that the cache is fully associative, starts empty, and it has a total size of 192 KB:

What effect does loop interchange on the overall performance compared with the original code performance shown above if the cache block size is 8 bytes? Discuss. 2 pts for effect and 3 for discussion

Solution: no effect because the block (8 bytes) only fits one array cell (also 8 bytes) so we will always miss regardless if we interchange of not.

C) What's the difference between power gating (1 pt) and clock gating (1 pt)? Which one(s) affect the dynamic power and/or the static power? (3 pts)

Solution: power gating is when the power is turned off completely for certain modules and this will help static power or leakage to go down.
Clock gating is turning off the clock of the module which reduces dynamic energy but not static.

D) Two processors A and B exhibit the same average power and the same clock frequency. They have a different instruction set and architecture but manage to get the same CPI. Both have no stalls. Will they have the same energy efficiency (1 pt)? Explain why or why not. (4 pts)

Solution:
E_A = P_A x execution time = P_A x IC_A x CPI_A x CCT_A
E_B = P_B x execution time = P_A x IC_B x CPI_A x CCT_A
Everything is equal except the instruction count IC because they have a different instructions set which might result in a longer code on one than the other, given the same program. Which makes one of them more energy efficient than the other.

Exercise 3 (10 pts)

Consider a cache system with a miss penalty of C cycles for a cache block of size C words. This is due to copying each word into the cache requiring 1 cycle.

Assume the following memory content organized in words.

```
Word 0
Word 1
Word 2
Word 3
Word 4
Word 5
Word 6
Word 7
Word 8
Word 9
Word 10
And so on…
```

i) For a cache that starts empty, with block sizes of 16 bytes, and a critical word first approach employed, compute the miss penalty cycles when word 2 is requested from memory. Also compute the number of cycles reduced compared the conventional approach. (3 pts)

Solution: blocks contain 16 bytes which means 4 words. When word 2 is requested, the block containing it is the first one (word 0,1,2,3) so conventional approach required 4 cycles, while critical requires 1. So the number of cycles reduced is 4-1 = 3 cycles.

ii) For a cache that starts empty, with block sizes of 16 bytes, and a early restart approach employed, compute the miss penalty cycles when word 2 is requested from memory. Also compute the number of cycles reduced compared the conventional approach. (3 pts)
Solution: word 2 is the third one so we need 3 cycles to fetch it in this approach, compared to 4 with conventional so we reduced 1 cycle only.

iii) For which words, do both approaches perform the same 1pt? Discuss. 1 pts
The first word of each block because both will require 1 cycle. So word 0, 4, 8,...

iv) For which words, does early restart perform its worst 1 pt? Discuss 1 pts
The last word in each block, requiring 4 cycles just like the conventional. so words 3,7,...

Exercise 4 (30 pts)

Consider two memory hierarchy systems M1 and M2.

M1 has two-level caches with the following measurements:

- In 100 memory references, 80 hit in L1 and 16 hit in L2.
- Hit time of L1 is 1 cycle and doubles in L2.
- Penalty of L2 is 10 cycles.

M2 has three-level caches with the following measurements:

- In 150 memory references, 120 hit in L1.
- Global miss rate of L2 is 6%, while it's 3% in L3.
- Penalty of L3 is 8 clock cycles.
- Hit time in L1 is 1 cycle, 3 cycles in L2 and 4 cycles in L3.

A) Compute the **local and global** miss rates of all caches in M1 and local miss rates of L1, L2 and L3 in M2. Show your work. 7 pts, 1 for each miss rate.

Solution:

System M1:

Level 1:

20 misses per 100 so 20% local miss rate, and same for global.

Level 2:

receives 20 requests, hit 16 so misses 4, which means local L1 = 4/20 = 20%

Global = 4/100 = 4%.

System M2:

Level 1: local and global miss rates = 30/150 = 20%

Level 2: since global is 6% it means 9 misses out of 150 requests. But L2 only receives 30 of those locally , which makes local miss rate 9/30 = 30%.

Level 3: global is 3% which makes those 4.5 misses out of 150. L3 only receives the misses in L2 which are (9 requests), making the local miss rate 4.5/9 = 50%

B) Compare both systems in terms of average memory access time. Show your work. (7 pts)

For m1: (3)

Avg mem access time = hit time L1 + miss rate L1 x (Hit time L2 + miss rate L2 x miss penalty L2)

1+ 0.2 x (2 + 0.2 x 10) = 1 + 0.2 (4) = 1.8 cycles.

For m2: (3 pts)
Avg mem access time = hit time L1 + miss rate L1 x (Hit time L2 + miss rate L2 x (hit time L3 + miss rate L3 x miss penalty L3)) =
1 + 0.2 x (3 + 0.3 x (4 + 0.5 x8)) = 2.08 cycles.

M1 is better (1 pt)

C) Given that 60% of the instructions are ALU and branch operations and the remaining are loads and stores, Compute the misses per instruction in all caches in both systems M1 and M2. (10 pts)

Since 40% are loads and stores, requiring an extra memory reference, this means each instruction needs 1.4 memory reference on average. (all for instruction fetch and 40% additional for load/store)

We need to compute the misses per instruction for L1 and L2 in M1:

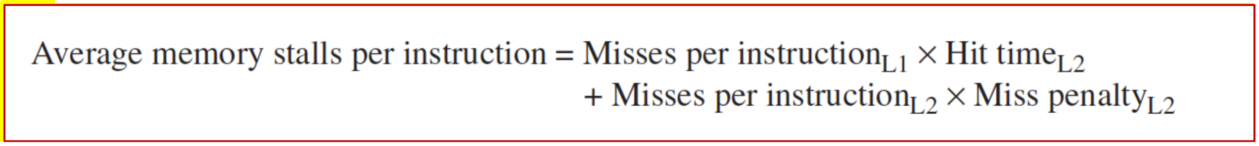
- 100 memory references means 100/1.4 = 71.4 instructions on average.
- 20 misses of those miss in L1 ➔ miss per instruction in L1 = 0.28 misses/instruction.
- 4 misses of those in L2 ➔ miss per instruction in L2 = (4/71.4) = 0.05 misses/instruction

For system M2:

- 150 memory references means 150/1.4 = 107 instructions on average.
- L1: 30 / 107 = 0.28 misses/instruction
- L2: 9/107 = 0.08 misses/instruction
- L3: 4.5/107 = 0.04 misses/instruction

D) compare both systems in terms of average memory stalls per instruction. Show your work. (6 pts)

M1:



$$\text{Average memory stalls per instruction} = \text{Misses per instruction}_{L1} \times \text{Hit time}_{L2} + \text{Misses per instruction}_{L2} \times \text{Miss penalty}_{L2}$$

= 0.28 x 2 + 0.05 x 10 = 1.06 stall cycles per instruction.

M2: miss per inst L1 x hit time L2 + miss per instruction L2 x hit time L3 + miss per inst L3 x miss penalty
= 0.28 x 3 + 0.08 x 4 + 0.04 x 8 = 1.48 stall cycles.

Also m1 is better.

Exercise 5 (20 pts)

Consider the following MIPS code and answer all the parts.

```
LOOP: L.D     F0, 800(R1)
      L.D     F1, 800(R2)
      ADD.D   F0, F0, F1
      MUL.D   F0, F0, F7
      S.D     F0, 800(R1)
      ADDUI    R1, R1, -8
      ADDUI    R2, R2, -8
      BEQ      R2, R3 LOOP
```

Assume the following latencies and that forwarding is applied and also assume branches execute in the decode stage normally and they don't use a delayed slot.

Instruction producing result	*Instruction using result*	*Latency in clock cycles*
FP load	FP ALU op	1
FP add	FP store	2
FP multiply	FP store	4
FP add	FP ALU op	3

3.1) Compute the number of cycles needed for 1 iteration with no scheduling or unrolling.

Show your work. 8 pts (6 for code, and 2 for cycles)

Solution:

```
LOOP: L.D   F0, 800(R1)
 L.D   F1, 800(R2)
 stall
 ADD.D F0, F0, F1
 stall stall stall
 MUL.D F0, F0, F4
 stall stall stall stall
 S.D   F0, 800(R1)
 ADDUI    R1, R1, -8
 ADDUI    R2, R2, -8
 stall
 BEQ      R2, R3 LOOP
```

# of cycles per iteration = 17.

The code is re-given here for your reference.

```
LOOP: L.D     F0, 800(R1)
      L.D     F1, 800(R2)
       ADD.D   F0, F0, F1
       MUL.D   F0, F0, F7
       S.D     F0, 800(R1)
       ADDUI    R1, R1, -8
       ADDUI    R2, R2, -8
       BEQ      R2, R3 LOOP
```

3.2) Unroll the loop 2 times and schedule the loop to reduce the number of stalls to minimum.

Re-compute the number of cycles needed per iteration. (Use the registers F2 and F3, in addition to F0 and F1).

Show your work 12 pts (8 for code, 4 for # of cycles per iteration)

Solution:

```
LOOP: L.D   F0, 800(R1)
      L.D   F1, 800(R2)
      L.D   F2, 792(R1)
      L.D   F3, 792(R2)
      ADD.D F0, F0, F1
      ADD.D F2, F2, F3
      Stall stall
      MUL.D F0, F0, F4
      MUL.D F2, F2, F4
      ADDUI    R1, R1, -16
      ADDUI    R2, R2, -16
      stall
      S.D   F0, 816(R1)
      S.D   F2, 808(R1)
      BEQ      R2, R3 LOOP
```

Total number = 16 cycles per 2 iterations → 8 cycles per iteration.