2102-COL216 Quiz 2

Viraj Agashe

TOTAL POINTS

13.5 / 20

QUESTION 1

1 Processor Performance 10 / 10

✓ **+ 2 pts** **Correct percentage of instructions executed in the benchmark**

Part (a)

✓ **+ 2 pts** **Correctly identifying penalty reduction**

✓ **+ 1 pts** **Correct Effective CPI**

✓ **+ 1 pts** **Correct Speedup**

+ 0 pts Incorrect/Unattempted

Part (b)

✓ **+ 2 pts** **Correctly calculating overall average branch penalty**

✓ **+ 1 pts** **Correct Effective CPI**

✓ **+ 1 pts** **Correct speedup**

+ 1 pts Correct Idea, but wrong/incomplete calculation

+ 0 pts Incorrect/Unattempted

+ 0 pts Incorrect/Unattempted

QUESTION 2

2 Miss Penalty 3.5 / 10

+ 0 pts Unattempted

+ 0 pts Wrong

✓ **+ 2 pts** **Finding memory transfer times**

+ 1 pts Case-A: WB policy (read miss time)

+ 1 pts Case-A: WB policy (write miss time)

+ 1 pts Case-A: WTWA policy (read miss time)

✓ **+ 1 pts** **Case-A: WTWA policy (write miss time)**

+ 1 pts Case-A: WTNWA policy (read miss time)

✓ **+ 1 pts** **Case-A: WTNWA policy (write miss time)**

+ 2 pts Case-B (WB, WTWA, WTNWA)

- 0.5 **Point adjustment**

Partially incorrect answers (-1)

Name	Entry No.
VIRAJ AGASHE	2020CS10567

COL216 Computer Architecture **Quiz 2** **17.03.2022**

Q 1. A processor is implemented as a 5 stage pipeline. On encountering a branch instruction, any inline instructions following it in the pipeline are flushed. The decision whether branch is to be taken and computation of the target address, both happen in the third stage. The next instruction is fetched after that. The processor shows an effective CPI of 1.4 for a specific benchmark. Find the percentage of instructions executed in the benchmark that are branch instructions. Assume that in 70% of branch instructions branch is taken. The implementation team is considering one of the following two enhancements. Find the performance speed up in each case.

a) Delayed branch with 1 delay slot is introduced for all branch instructions. Obviously, the instructions in the delay slots don't get flushed. Assume that for the benchmark under consideration, the compiler is able to fill up 80% of delay slots with useful instructions and remaining ones with no-ops.

b) Branch Target Buffer (BTB) is introduced which is looked up in the fetch cycle itself. If there is a BTB hit (assume hit rate of 80%), the target instruction is fetched in the next cycle from the address given by BTB. If there is a BTB miss, the processor continues with inline instructions (no flushing). When the branch instruction reaches the third stage, it is revealed whether the next instruction fetched is correct or not (assume that it is correct 90% of the time when there is a BTB hit and 80% when there is a BTB miss). Correct next instruction is fetched in the following cycle if required.

Q1. Let the fraction of branch instructions is f.

Due to the branch instructions having inline instructions flushed, If branch instruction starts at C, next instruction will start at cycle C+3.

So, $(1-f)I$ => Delay of 1 cycles

$(f)I$ => Delay of 3 cycles.

ROUGH
b (1)
add 4 (+3)
C+3
b → 3

$$\therefore CPI = \frac{(1-f)I \times 1 + f \times I \times 3}{I} = 1.4$$

$$\Rightarrow 1 + 2f = 1.4 \Rightarrow f = 0.2$$

$\therefore$ 20% instructions are branch instructions.

(a) Speed-up with delayed branch.

Delay: $fI \Rightarrow (0.20)(3) + (0.80)(2) = 1.6 + 0.6 = 2.2$

$$\therefore CPI = \frac{(1-f)I \times 1 + f \times I \times 2.2}{I} = 0.8 \times 1 + 0.2 \times 2.2$$

$$= 0.80 + 0.44$$

$CPI_{new} = 1.24$

So the speedup achieved is

$$\frac{CPI_{old}}{CPI_{new}} = \frac{1.4}{1.24} = \text{Ans.}$$

(b) Cycles for branch ⇒

$$fI \Rightarrow \underbrace{0.80}_{\text{hit in BTB}} \times \underbrace{0.90}_{\text{correct}} \times \underbrace{1}_{\text{1 cycle delay}} + 0.80 \times 0.10 \times 3$$

$$+ \underbrace{0.20}_{\text{miss in BTB}} * \underbrace{0.80 \times 1}_{\text{correct (inline)}} + 0.20 \times \underbrace{0.20}_{\text{incorrect}} \times \underbrace{3}_{\text{3 cycle delay}}$$

$$= 0.72 + 0.24 + 0.16 + 0.12$$

$$= 0.40 + 0.84 = 1.24.$$

$$\therefore CPI_{new} = \frac{(1-f)\cancel{I} \times 1 + f \times \cancel{I} \times 1.24}{\cancel{I}}$$

2.48

$$= 0.8 + 0.2 \times 1.24$$

$$= 0.8 + 0.248$$

$$= 1.048$$

$$\text{Speedup} = \frac{CPI_{old}}{CPI_{new}} = \frac{1.4}{1.048} = \text{Ans.}$$

Name	Entry No.
VIRAJ AGASHE	2020CS10567

COL216 Computer Architecture **Quiz 2** **17.03.2022**

Q 2. Compare read miss penalty and write miss penalty for 3 different write policies for cache - WB, WTWA and WTNWA. Assume that the memory interface requires n_1 cycles to send address, n_2 cycles to read/write the first word after sending address and n_3 cycles to read/write each of the subsequent words. Wherever applicable, assume that the probability of a block being dirty at the time of its eviction is p. Do the comparison for the following two read/load policies.

a) Sequential read, no data forwarding, no wrap-around load
b) Concurrent read, data forwarding, wrap-around load

Q2. (a) Sequential read, no data forwarding, no wrap-around load.

Read miss penalty	Write Miss Penalty
WB ⇒	

WB ⇒ The entire block is read from memory & written into the cache.

Let a block have B+1 words, then.

Write Miss penalty ⇒ $n_1 + n_2 + B \times n_3$.

WTWA ⇒ Block is read from memory into the cache, also written into the memory.

Write miss penalty = $n_1 + n_2 \times B \times n_3 + \underbrace{n_1 + n_2}_{\text{for writing into mem}}$

WTNWA ⇒ Only written into memory, block not loaded into the cache.

W.M. ⇒ $\underbrace{n_1 + n_2}_{\text{for writing into memory only}}$ one word.

VIRAJ AGASHE
2020CS10567

Read Miss Penalty.

Read into memory in each case.

WA ⇒

Delay in WA, WTWA & WTNWA

⇒ $n_1 + n_2 + B \times n_3$