CHALMERS
EXAMINATION / TENTAMEN

Course code/kurskod	Course name/kursnamn			
DAT105	computer Architecture			
Anonymous code Anonym kod		Examination date Tentamensdatum	Number of pages Antal blad	Grade Betyg
DAT105-0002-WPA		2023-8-15	8	4

* I confirm that I've no mobile or other similar electronic equipment available during the examination.
Jag intygar att jag inte har mobiltelefon eller annan liknande elektronisk utrustning tillgänglig under eximinationen.

Solved task Behandlade uppgifter No/nr		Points per task Poäng på uppgiften	Observe: Areas with bold contour are to completed by the teacher. Anmärkning: Rutor inom bred kontur ifylles av lärare.
1	✓	12	
2	✓	7	
3	✓	8	
4	✓	4	
5	✓	10	
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
Bonus poäng			
Total examination points Summa poäng på tentamen			

5

1A) for computer A:

weighted arithmetic mean $= \frac{1\times 50\% + 2\times 25\% + 4\times 25\%}{1}$

$= 2.5$

for computer B:

weighted arithmetic mean $= \frac{2\times 50\% + 2\times 25\% + 2\times 25\%}{1}$

$= 2$

for computer Reference: $= \frac{6\times 0.5 + 2\times 0.25 + 2\times 0.25}{1}$

$= 4$

Computer A and computer have the same performance on P_1, P_2, P_3.

Weighted arithmetic means is sensetive to outliers. If a compute outpefformaned than another one for 9 out of 10 programe, but only one underperformaned, it could be evaluted as lower performance them the other. 2

ii) $SP_{A,R} = \frac{T_{R,P_1}}{T_{A,P_1}} = 6$, $SP_{A,P_2} = \frac{2}{2} = 1$, $SP_{A,B} = \frac{2}{4} = 0.5$.

Gemotric mean $= \sqrt[3]{6\times 1\times 0.5} = 1.443$

$SP_{B,P_1} = \frac{T_{R,P_1}}{T_{B,P_1}} = \frac{6}{2} = 3$, $SP_{B,P_2} = \frac{2}{2} = 1$, $SP_{B,P_3} = \frac{2}{2} = 1$

Gemotric mean $= \sqrt[3]{3\times 1\times 1} = 1.443$. 4

A B have the same performance.

Geometric mean is no sensetive to outliers.

CHALMERS | Anonymous code / Anonym kod: DAT-0002-WPA | Points for question (to be filled in by teacher) / Poäng på uppgiften (ifylles av lärare) | Consecutive page no. / Löpande sid nr 2 | Question no. / Uppgift nr 1

1B)

$$10 = \frac{1}{P + \frac{(1-P)}{100}}$$

$$10 = \frac{1}{(1-P) + \frac{P}{F}}$$

$$10 = \frac{1}{(1-P) + \frac{P}{100}} \Rightarrow 1 - P + 0.01P = 0.1$$

$$0.9 = 0.99P.$$

$$P \approx 91\%.$$

3

1C). ~~Assume function F~~

$$(20\% \cdot I_C \times 2 + 80\% \cdot I_C) \times T_C = 1.25.$$

$$(0.4 I_C + 0.8 I_C) \cdot T_{ns} = 1.25.$$

$$1.2 I_C \times T_{ns} = 1.25.$$

$$I_C = 10^9.$$

3

12

2A) If floating point unit is fully pipelined, then I_2 could be trigger before I_1, because I_2 don't use the ~~[illegible]~~ floating point unit.

A branch predictor can be added between IF and ID.

2B)

ii) We can schedul the instructions like below to eliminate cycle lost.

```
I1  Loop  L.S   F0, 0(R1)   (1)
I2        L.S   F1, 0(R2)   (1)
I3        ADD.S F2, F1, F0  (2)
I5        ADDI  R1, R1, #4  (1)
I6        ADDI  R2, R2, #4  (1)
I7        SUBI  R3, R3, #1  (1)
I4        S.S   F2, -4(R1)  (1)
I8        BNEZ  R3, Loop.   (2).
```

That's not what is asked for

i) $I_2 \rightarrow I_3$: is RAW hazard, I_3 must to waiting for I_2 finish, then can get the operand.

$I_3 \rightarrow I_4$: is also RAW hazard. I_4 must waiting for I_3 to finish.

$I_7 \rightarrow I_8$ is also RAW hazard. I_8 must waiting for I_7 to finish.

3

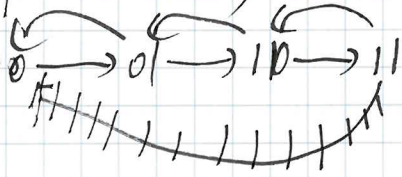
2C

There four states for 2-bit predictor.

00: untaken 01: weakly untaken in this two states, the branch will be predicted as untake.

10: weakly taken, 11: taken. in this two states, the branch will be predicted as taken.

When ~~brar~~ predictor predict correctly, the states will not change, otherwise it will change from 00 → 01 → 10 → 11



.

if a branch is always taken ~~fdbral~~ untaken.

like ~~"untaken"~~ "untaken", "taken", "untaken", "taken"...

for 1-bits predictors. if it initial the ~~untake~~ untake. first one is correct, it's state 0, predict next one is also "untaken" but it's "taken", it will change to state "1", so for the subsequential prediction are all incorrect.

for 2-bits predictor, if it inintial with the untaken. first on is correct. it's "00". next one is "taken", state ~~chg~~ change to "01" still predice untaken, so the next one is also correct. So the ~~pe of~~ performance is better than one-bit predictor.

3/7

3A). when I_2 issuing and access to the RAT, F4 will be associated with a tag from I_2. And when I_3 access to the RAT, F4 will also be associated with tag from I_3. There are different tags, and only I_4 can change the value of register file.

4

3B). ROB is the main mechnasm to track the vaule of speculative execution. ROB buffers all the instructions in order as they appear in the programe. when an instruction reach the top of ROB, it will be retired/committed in the next cycle and the vaule will be written back to register file.

A processor will continues all the instruction before branch and speculathe execution, when the branch reach the top of ROB, if the prediction is incorrect, the all speculatively execution win be flushed, and re-execution the right branch with correct adderess

what about exceptions

-4

4

8

5

4A).

for block cache:

~~4X7=28~~ $3X8+7+100=131$.

for non-block cache:

$4X8=32$.

how faster: $\frac{131-32}{32}\approx 3$.

✓

4B).

4C). inclusive two-level cache: The block in the ~~lower~~ upper must ~~in~~ exist in lower cache.

exclusive two-level cache: The upper and lower cache have different content.

4

4

5

5A). The value return back R_2 is 0. Because P_2 read the value from its own cache.

We can simplely invalidate the block of P_2 during the instruction "W1=8". After that, when p_2 send a redd request will cause read miss. Then a BusRed message will be posted on the bus. and P_1 will response the request to copy the block to P_1.

4

5B).

S→M. ~~W~~ Write to its own cache will change the block from ~~to~~ Shared to modified. And send an BusUpgrad messge on the bus to ~~invila~~ invidate other processor's cache with the copy of block.

~~M→I. A read miss transcation will post a BusRead on the bus, if some processor has the copy of this block.~~

M→I. A cache block in M mems it's up-to-~~date~~ update. If now another processor ~~with~~ send a write request on its own cache with the same block. The the BusUpgrad message will inforced the M block to I block.

4

5

5c). Fine-grain: thred switch happens in each cycle. thread swith should be added between IF and ID. Each ~~thred~~ thread should be compared with thread ID.

Coarse-grain: thread switch happens in a long latency waiting. thredd switch should be added between IF and ID. But detecting should be implemented as memory stage.

not explained how pipeline is modified

2

10