Computer Architecture Theory + Lab (CS 305/341)

Assignment 6: Pipelining Due Date: 27/10/20

(Theory Assignment 3)

Name: Niraj Mahajan

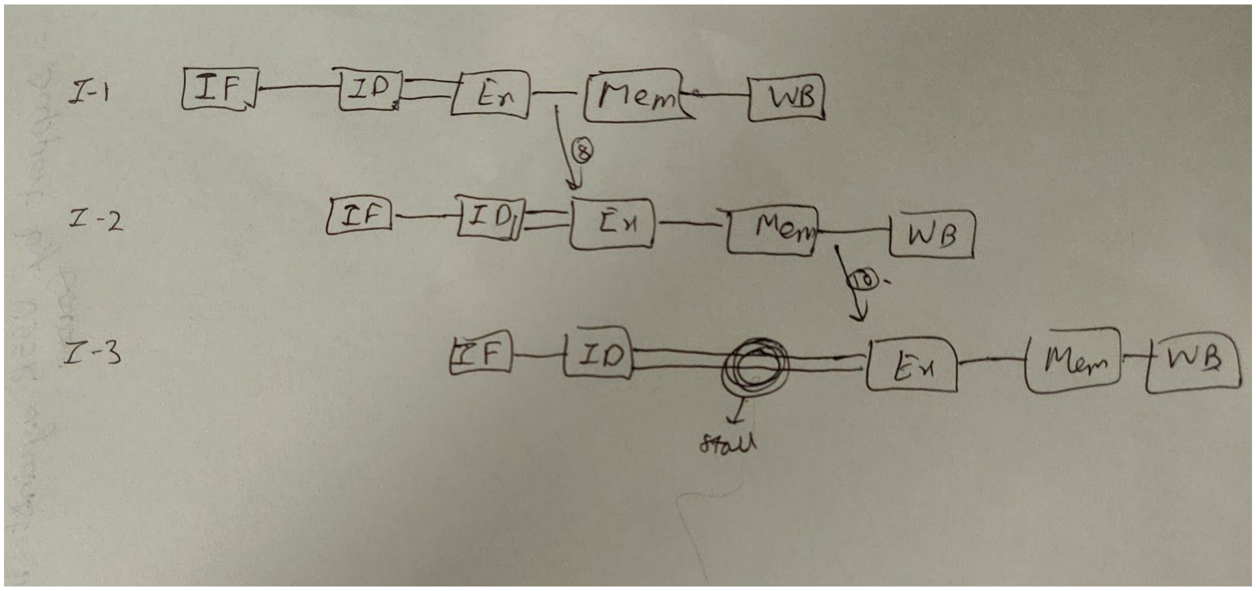
Roll Number: 180050069

[QUESTION 1]

1. Consider the execution of the following sequence of three instructions on the 5-stage MIPS pipeline.

I1	add	8 9 10
I2	lw	10 400(8)
I3	sub	9 10 8

2. List all dependences and their types
   a. RAW - True Dependencies
      a. Register 8 - Between I1, I2
      b. Register 8 - Between I1, I3
      c. Register 10 - Between I2, I3
   b. WAR - Anti Dependencies
      a. Register 10 - Between I1, I2
      b. Register 9 - Between I1, I3
3. Show all forwarding links between each instruction pair assuming full forwarding



There is a Stall in the last part of the diagram. This is due to the RAW dependency for register 10 between I2, I3.

4. Assume three cases separately – (i) no forwarding (ii) ALU-ALU forwarding only

and (iii) full forwarding. Also, assume that the pipeline clock cycle times are 250, 270 and 280 picosecs. for the three cases respectively. Answer the following questions for each of the three cases.

(i) No forwarding

(c) Draw a multicycle pipeline diagram showing clearly the stage each instruction is in during each cycle

Inst\ Cycle	1	2	3	4	5	6	7	8	9	10	11
add 8 9 10	IF	ID	EX	MEM	WB						
lw 10 400(8)		IF	ID	[RAW]	[RAW]	EX	MEM	WB			
sub 9 10 8			IF	[STRUCT]	[STRUCT]	ID	[RAW]	[RAW]	EX	MEM	WB

(d) What is the total execution time?

Number of cycle * Time per cycle = 11 * 250 = 2750 ps

(e) What is the average utilization of the pipeline?

Average Utilisation = Instruction/ Num. cycles = 3/11 = 0.2727

(d) Can the instructions be re-ordered to decrease the total execution time? If so, how?

No, we cannot reorder instructions since each instruction has a dependency on the previous one by RAW.

(ii) ALU-ALU forwarding only

(c) Draw a multicycle pipeline diagram showing clearly the stage each instruction is in during each cycle

Inst\ Cycle	1	2	3	4	5	6	7	8	9
add 8 9 10	IF	ID	EX	MEM	WB				
lw 10 400(8)		IF	ID	EX	MEM	WB			
sub 9 10 8			IF	ID	[RAW]	[RAW]	EX	MEM	WB

(d) What is the total execution time?

Number of cycle * Time per cycle = 9 * 270 = 2430 ps

(e) What is the average utilization of the pipeline?

Average Utilisation = Instruction/ Num. cycles = 3/9 = 0.333

(d) Can the instructions be re-ordered to decrease the total execution time? If so, how?

No, we cannot reorder instructions since each instruction has a dependency on the previous one by RAW.

(iii) Full Forwarding

(c) Draw a multicycle pipeline diagram showing clearly the stage each instruction is in during each cycle

Inst\ Cycle	1	2	3	4	5	6	7	8
add 8 9 10	IF	ID	EX	MEM	WB			
lw 10 400(8)		IF	ID	EX	MEM	WB		
sub 9 10 8			IF	ID	[RAW]	EX	MEM	WB

(d) What is the total execution time?

Number of cycle * Time per cycle = 8 * 280 = 2240 ps

(e) What is the average utilization of the pipeline?

Average Utilisation = Instruction/ Num. cycles = 3/8 = 0.375

(d) Can the instructions be re-ordered to decrease the total execution time? If so, how?

No, we cannot reorder instructions since each instruction has a dependency on the previous one by RAW.

[QUESTION 2]

The goal of this problem is to compare the performance (accuracy) of different predictors for the snippet below.

```
main        move  t3    $0
            addi  t0    $0    100
chalo:      andi  t1    t0    7
branch:     bgt   t1    2     here
            sll   t2    t0    2
            add   t3    t3    t2
            b     join
```

```
here:       sll     t2      t0      1
            sub     t3      t3      t2
join:       addi    t0      t0      -1
            bnez    t0      chalo
```

In each part below assume that the branch predictor is initially in the "Strong NOT Taken"state.

ASSUMPTIONS : For (m,k) case: all the m bits are initialised to NT at start (ie, not taken). Mispredictions are counted only for the bgt instruction (ie bgt t1 2 here). In the asymptotic case, assuming n to be very large, the constant term is neglected, unless the coefficient of n is 0. Also, the least significant k bits are treated to be different for both branches.

a. The number of mispredictions with a (0, 2) branch predictor is ___51___ out of 100 and is asymptotically ____n/2____

Value of t1	7	6	5	4	3	2	1	0
Branch taken?	T	T	T	T	T	NT	NT	NT
state (before exec)	4	3	2	1	1	1	2	3

b. The number of mispredictions with a (0, 2) branch predictor is ___42___ out of 100 and is asymptotically ____3n/8____

Value of t1	7	6	5	4	3	2	1	0
Branch taken?	T	T	T	T	T	NT	NT	NT
state (before exec)	4	3	2	1	1	1	2	3

c. The number of mispredictions with a (0, 2) branch predictor is ___26___ out of 100 and is asymptotically ____n/4____ and occurs for k = 1.

Value of t1	7	6	5	4	3	2	1	0
Branch taken?	T	T	T	T	T	NT	NT	NT
state (before exec)	2	1	1	1	1	1	2	2

d. The number of mispredictions with a (2, 2) correlating branch predictor is ___28___ out of 100 and is asymptotically _____n/4______. (An (*m*, *k*) correlating branch predictor keeps track of the outcomes of the most recent branches – both local and global.

Value of t1	7	6	5	4	3	2	1	0
Branch taken?	T	T	T	T	T	NT	NT	NT
state (before exec)	4	2	1	1	1	1	3	4
history	1	3	3	3	3	3	1	1

e. As the number of iterations, *n*, tends to infinity, the minimum number of mispredictions with an (*m*, *k*) correlating branch predictor is achieved for *m* = ___10___ and *k* = ___1___. For this predictor, the number of mispredictions is ___7___ out of 100 and is asymptotically ___0*n + 7___.

Answer Explanation:

We need some number of m history bits such that given these m bits we can accurately predict what the decision for the next branch will be. Note that bnez branch will always be accepted. Also, note that there is a pattern for the branch outcomes TTTTTNNN. So we just need to make a prediction for the bgt branching. So given any 5 consecutive set of history bits, we can easily predict the next outcome. The idea here is to consider the minimal consecutive set of bits, such that they are unique in the pattern (since if m was 4, then TTTT is a pattern which is non-unique, and occurs twice in TTTTTNNN). So we need a set of 5 history bits to predict the next outcome of bgt. But since there are 5 T bits of bnez as well, overall, we need 10 bits to accurately predict the outcome of branching.
The value of k does not affect the mispredictions, as after a certain buffer/adjustment period at the start everything is either strongly taken or strongly not taken (1 or 2^k).
However, for k = 1, we observe that the number of mispredictions is the minimum, hence I have reported this answer.