

# Aben Palayil

Q1

Ex 4.14.1

# Homework #5

Instructions	1	2	3	4	5	6	7	8	9	10	11	12	13	14
lw r2, 0(\$1)	F	D	Ex	Me m	W B									
label1: beg r2,r0, label2		I F	I D	*	Ex	Me m	W B							
lw r3,0(r2)					IF	ID	Ex X	Me m	W B					
beq r3,r0, label1						IF F	I D	*	Ex X	Me m	W B			
label1: beg r2,r0, label2 (branch taken)							I F	*	I D	E X	Me m	W B		
sw r1,0(r2)									IF	ID	Ex	Me m	W B	

Ex 4.14.2

Instructions	1	2	3	4	5	6	7	8	9	10	11	12	13	14
lw r2, 0(\$1)	F	D	Ex	Me m	W B									
label1: beg r2,r0, label2		I F	I D	*	Ex	Me m	W B							
lw r3,0(r2)				IF	*	ID	Ex	Me m	W B					
beq r3,r0, label1						IF	ID	Ex	Me m	W B				
add r1,r3,r1							IF	ID	Ex	Me m	WB			
label1: beg r2,r0, label2 (branch taken)								IF	ID	Ex	Me m	W B		
lw r3,0(r2)								IF	ID	Ex	Me m	W B		
sw r1,D(r2)									IF	ID	Ex	Me m	W B	

## Q2 Ex 4.16.1

In these outcomes the number of T's are 3 and the total numbers of outcomes are 5. Hence from the above the accuracy for Always-Taken predictor is determined by:

$$\text{Accuracy} : \frac{3}{5} = 0.6$$

$\therefore$  It can be said that for Always-Taken predictor the accuracy is 60%,, In these outcomes the number of NT's are 2 and the total number of outcomes are 5. Hence from the above the accuracy for Always-Not-Taken predictor is determined by:

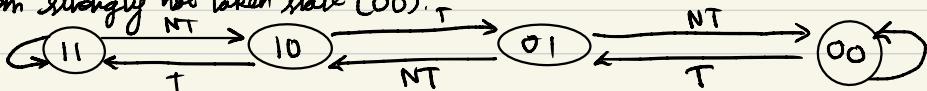
$$\text{Accuracy} : \frac{2}{5} = 0.4$$

$\therefore$  It can be said that for Always-Not-Taken predictor the accuracy is 40%,,

## Ex 4.16.2

Branch Outcomes	Predicted Outcome	Predictor Value	Hit or Miss
Taken	Not Taken	0	Miss
Not Taken	Not Taken	1	Hit
Taken	Not Taken	0	Miss
Taken	Not Taken	0	Miss

The predictor starts off in the bottom left state. It means start the prediction from strongly not taken state (00).



$$\text{Accuracy of predictor} = \frac{\text{No. of Hits}}{\text{Total branches taken}} = \frac{1}{4}$$

$$= 0.25$$

$\therefore$  The accuracy of the two-bit predictor for the first four branches is 25%,

## Ex 4.16.3

- The repeating pattern : T NT T NT NT

- The prediction generated by the 2-bit predictor for this pattern : T T T T T

- The given sequence the accuracy:  $\frac{3}{5} \times 100 = 60\%$ ,

- When the given pattern is repeated forever, then the pattern will be

T NT T T NT TNT TT NT T NT TT NT TT NT ...

- The predictions generated for the given sequence is: TTTTTTTTTTTTTT.
- ∴ Accuracy is:  $\frac{12}{20} \times 100 = 60\%$ .

Thus, using induction, it can be said that the accuracy for the 2-bit predictor will be 60%. when the given pattern repeats forever.

- Q.3 a) Convert above C code to MIPS code register R<sub>1</sub> contains the address of the highest array element, and F<sub>2</sub> contains the S. Register R<sub>2</sub> is precomputed. R<sub>2</sub>(R<sub>2</sub>) contains address of the last element.

LOOP: L.D F0, 0(R1); F0 = array element.

ADD. D F4, F0, F2; add scalar in F<sub>2</sub>.

S. D F4, 0(R1); Store result.

DAD D1 R1, R1, #8; decrement pointer - 8 bytes (per DW)

BNE R1, R2, Loop; branch R1 != R2

- b) Show any hazard on your MIPS code and insert stall if needed (lockcycles needed for unscheduled code).

Issued at (clock cycle)

Loop: L.D F0, 0(R1) [1]

Stall 2 F0 is used in ADD D ⇒ requires stall (RAW hazard)

ADD D F4, F0, F2 [3]

Stall 4 F4 is used in S.D ⇒ requires stall (RAW hazard)

Stalls

S.D F4, 0(R1) [6]

DADDV1 R1, R1, # - 8 [7]

Stall 8 R1 is used in BNE ⇒ requires stall (RAW hazard)

BNE R1, R2, Loop [9]

- c) Issued at (clock cycle)

Loop: L.D F0, 0(R1) [1]

DADDV1 R1, R1, # - 8 [2]

ADD.D F4, F0, F2 [3]

Stall 4 F4 is used in S.D  $\Rightarrow$  requires stall (RAW hazard)

Stall 5

S.D F4, 8(R1) [6]

BNE R1, R2, Loop [7]

d) Stall (Raw hazard)

S.D F4, 0(R1) F4 is used in S.D  $\Rightarrow$  requires stall (Raw hazard)

L.D F6, -8(R1)

ADD.D F8, F6, F2

S.D F8, -8(R1)

DF1D, -16(R1)

ADD.D F12, F10, F2

S.D F12, -16(R1)

L.D F14, -24(R1)

ADD.D F16, F14, F2

SD F16, -24(R1)

DADDV1 R1, R1, # -32

BNE R1, R2, Loop

Rescheduling the code

Loop: L.D F0, 0(R1)

L.D F6, -8(R1)

L.D F10, -16(R1)

L.D F14, -24(R1)

ADD.D F4, F0, F2

ADD.D F8, F6, F2

ADD.D F12, F10, F2

ADD.D F16, F14, F2

SD F4, 0(R1)

SD F8, -8(R1)

DADDV1 R1, R1, # -32

S.D F12, 16(R1)

S.D F16, 8(R1)

BNE R1, R2, Loop

This eliminates the stall cycle & no two dependent instructions are placed consecutively.

(Q4. Ex 5.3.1

The block size can be determined by looking into the offset bits. We have an offset of 4 (0 - 4) bits. Therefore, we have  $2^5$  or 32 words in a block.

Ex 5.3.2

As the index field is of 5 bits (9-5) therefore, no. of entries in the cache =  $2^5$  = 32 entries.

Ex 5.3.3.

Assume 1 valid bit is there for each tag along with the tag bits. ∴ Per block we have:

$$\text{Data bits} = 8 \times 32 = 256 \text{ bits}$$

$$\text{Tag bits} = 22 \text{ bits}$$

$$\text{Valid bits} = 1 \text{ bit}$$

∴ Total bits needed to implement the cache per block =  $256 + 22 + 1 = 279$  bits //

∴ Ratio of total bits over data storage bits = total bits / data =  $279 / 256 = 1.089$ ,

Ex 5.3.4

Address	Index	Offset	Hit/Miss	Replace?	Final Value?
0	00000	00000	M	N	N
4	00000	00100	H	N	N
16	00000	10000	H	N	N
132	00100	00100	M	N	N
232	00111	01000	M	N	Y
160	00101	00000	M	N	Y
1024	00000	00000	M	Y	N
30	00000	11110	M	Y	N
140	00100	01100	H	N	N
3100	00000	11100	M	Y	Y
180	00101	10100	H	N	Y
2180	00100	00100	M	Y	Y

There are 4 entries where there is 'Y' in the replace column.  
So, 4 blocks are replaced.

Ex 5.3.5

Total no. of references = 12

Total no. of hits = 4

$$\therefore \text{Hit ratio} = \frac{4}{12} = 0.33,$$

Ex 5.3.6

$\langle 00100, 0010, \text{mem } [2176] \rangle$

$\langle 00101, 0000, \text{mem } [160] \rangle$

$\langle 00000, 0011, \text{mem } [3072] \rangle$

$\langle 00111, 0000, \text{mem } [224] \rangle$