ECE341 Homework No. 7 Solution

Problem No. 1

The following tables show the predictions made by the two branch predictors for each branch instance:

(a) 1-bit branch predictor, which starts in the LNT state

Branch Instance	Current State	Prediction	Actual Outcome	Next State
1	LNT	NT	T	LT
2	LT	T	T	LT
3	LT	T	T	LT
4	LT	T	NT	LNT
5	LNT	NT	NT	LNT
6	LNT	NT	T	LT
7	LT	T	T	LT
8	LT	T	T	LT
9	LT	T	NT	LNT
10	LNT	NT	NT	LNT

Branch prediction accuracy = 6/10 = **60%**

(b) 2-bit branch predictor, which starts in the SNT state

Branch Instance	Current State	Prediction	Actual Outcome	Next State
1	SNT	NT	T	LNT
2	LNT	NT	T	ST
3	ST	T	T	ST
4	ST	T	NT	LT
5	LT	T	NT	SNT
6	SNT	NT	T	LNT
7	LNT	NT	T	ST
8	ST	T	T	ST
9	ST	T	NT	LT
10	LT	T	NT	SNT

Branch prediction accuracy = 2/10 = **20%**

Problem No. 2

Clock Rate "R" = 2.8 GHz = $2.8 * 10^9$ Hz

The processor incurs a stall of 1-cycle whenever a *Load* instruction is immediately followed by a dependent instruction.

Therefore: Stall frequency = %age of *load* instructions * %age of *load* instruction followed by a dependent instruction = 0.35 * 0.4 = 0.14

Stall penalty = 1 cycle

δ = Stall frequency * Stall penalty = 0.14 * 1 = 0.14

Throughput $P_p = R / (1+\delta) = (2.8 * 10^9) / (1 + 0.14)$ = **$2.456*10^9$ instructions per second**

Problem No. 3

Percentage of branch instructions in the program = 20%
Percentage of branch instructions that are taken = 70%

For processor R_2:
Since the branch predictor always predicts a "not-taken" branch, it will incur a penalty on branches that are "taken". Thus the pipeline will have to stall on every "taken" branch.

Stall frequency = Percentage of branch instructions * Percentage of branches that are taken
= 0.2 * 0.7 = 0.14
Stall penalty = 2 cycles (Branch outcome computed in stage-3)
δ = Stall frequency * Stall penalty = 0.14 * 2 = 0.28

For processor R_3:
The dynamic branch predictor incurs a penalty whenever it predicts a branch incorrectly; causing the pipeline to stall.

Stall frequency = Percentage of branch instructions * Percentage of incorrect branch predictions
= 0.2 * (1 – Branch Prediction Accuracy)
Stall penalty = 2 cycles (Branch outcome computed in stage-3)
δ = Stall frequency * Stall penalty = 0.2 * (1 – Branch Prediction Accuracy) * 2 = 0.4 * (1 – Branch Prediction Accuracy)

(a) Since both R_2 and R_3 have the same clock rates, their throughputs would match, if both of them have the same number of stall cycles per instruction (δ):

0.4 * (1 – Branch Prediction Accuracy) = 0.28
1 – Branch Prediction Accuracy = 0.7
Branch Prediction Accuracy = 0.3 or **30%**

(b) Both the processors have the same clock rate "R".

For processor R_2:
Throughput P_p = R / (1+δ) = R / (1 + 0.28) = 0.781 R

For processor R_3:
δ = 0.4 * (1 – Branch Prediction Accuracy) = 0.4 * (1 – 0.9) = 0.04
Throughput P_p = R / (1+δ) = R / (1 + 0.04) = 0.962 R

Speedup for R_2 relative to R_1 = 0.962R / 0.781R = **1.231**

Problem No. 4

Clock Rate "R" = 3 GHz = 3 * 10^9 Hz
No branch prediction is used.
The compiler is able to fill 80% of the branch delay slots with useful instructions. Therefore 80% of the branches will have no stall penalty.

The remaining 20% delay slots are filled with NOP instructions. Therefore, 20% of the branches will incur a stall penalty.
Stall frequency = %age of branch instructions * %age of delay slots filled with NOP instructions
= 0.2 * 0.2 = 0.04
Since branch outcome is computed in the decode stage (stage-2):
Stall penalty = 1 cycle
δ = Stall frequency * Stall penalty = 0.04 * 1 = 0.04
Throughput $P_p = R / (1+\delta) = (3*10^9) / (1 + 0.04) =$ **$2.885*10^9$ instructions per second**

Problem No. 5

Clock Rate "R" = 2.4 GHz = $2.4*10^9$ Hz
We need to consider stall cycles due to both the cache misses and branch mispredictions.

For cache misses:
Percentage of Load and Store instructions = 20% + 10% = 30%
Percentage of instruction fetches that miss in the cache = 1%
Percentage of data accesses that miss in the cache = 2%
Cache miss penalty = 10 cycles
All the instructions need to access the cache for instruction fetches, whereas only the Load and Store instructions need to access the cache for data accesses.
Therefore:
δ_{miss} = stall frequency * stall penalty = (0.01 + (0.3 * 0.02)) * 10 = 0.16

For branch mispredictions:
Percentage of branch instructions = 20%
Since branch computation is carried out in "Decode" stage (stage-2):
Stall penalty = 1 cycle
Assume that the branch prediction accuracy is represented by "A", then:
Stall frequency = %age of branch instructions * %ge of branches which are mispredicted = (0.2)(1-A)
Therefore:
$\delta_{branch\_penalty}$ = stall frequency * stall penalty = (0.2)(1-A) * 1 = 0.2 - 0.2A

Adding the two different sources of stall cycles:
Total stall cycles per instruction $\delta = \delta_{miss} + \delta_{branch\_penalty} = 0.16 + 0.2 – 0.2A = 0.36 – 0.2A$

Throughput "P_p" is given by:
$P_p = R / (1+\delta) = (2.4*10^9)/(1 + 0.36 - 0.2A) = (2.4*10^9)/(1.36 – 0.2A)$
Since the required throughput is 2 billion instructions per second:
$2*10^9 = (2.4*10^9)/(1.36 – 0.2A)$
$1.36 – 0.2A = 2.4 / 2 = 1.2$
$A = (1.36 - 1.2) / 0.2 = 0.8$
Therefore, the required minimum branch prediction accuracy is **80%**