REPORT – Assignment 10

Implementing Cache hits/misses:

We are given two inputs x: hit probability and N: cycles to stall after a cache miss.
In the MEM stage whenever there is a need of a memory access (load or store) then we call a function that returns 1 with a probability of x. If there is a hit then we continue the processing normally, if there is a miss then we stall for n – 1 cycles.

Stalling:

When there is a cache miss the MEM stage sends a signal to all the previous stages to wait of N – 1 cycles and passes null information to the MEM-WB latch i.e. all the previous stages wait for N – 1 cycles and the WB stage is idle.

Testing:

For checking probability: Testcase with 1000 load operations

lw $t0 1000 $zero * 1000

Results:

N = 5, x = 0.2
Toal Cycles: 4184
No of instruction executed: 1000
Average instruction per cycle: 0.239006
Hit Count: 205
Miss Count: 795

N = 10, x = 0.2
Toal Cycles: 8231
No of instruction executed: 1000
Average instruction per cycle: 0.121492
Hit Count: 197
Miss Count: 803

N = 5, x = 0.4
Toal Cycles: 3460
No of instruction executed: 1000
Average instruction per cycle: 0.289017
Hit Count: 386
Miss Count: 614

N = 10, x = 0.4
Toal Cycles: 6647
No of instruction executed: 1000
Average instruction per cycle: 0.150444
Hit Count: 373
Miss Count: 627

N = 5, x = 0.8
Toal Cycles: 1756
No of instruction executed: 1000
Average instruction per cycle: 0.569476
Hit Count: 812
Miss Count: 188

N = 10, x = 0.8
Toal Cycles: 3038
No of instruction executed: 1000
Average instruction per cycle: 0.329164
Hit Count: 774
Miss Count: 226

N = 5, x = 1
Toal Cycles: 1004
No of instruction executed: 1000
Average instruction per cycle: 0.996016
Hit Count: 1000
Miss Count: 0

N = 10, x = 1
Toal Cycles: 1004
No of instruction executed: 1000
Average instruction per cycle: 0.996016
Hit Count: 1000
Miss Count: 0

****

//Storing first $v1 fibonacci numbers in the momory from (4095..4095 - $v0 + 1)
$a1 <= 1, $at <= 1

```
label1:
sub $sp $sp $at
sw $a0 0 $sp
sub $sp $sp $at
sw $a1 0 $sp
add $v0 $v0 $at
label2:
add $a2 $a1 $a0
sub $sp $sp $at
sw $a2 0 $sp
lw $a0 1 $sp
lw $a1 0 $sp
add $v0 $v0 $at
beq $v0 $v1 label3
j label2
label3:
```

N = 5, x = 0.2
Toal Cycles: 181
No of instruction executed: 76
Average instruction per cycle: 0.41989
Hit Count: 6
Miss Count: 23

N = 10, x = 0.2
Toal Cycles: 305
No of instruction executed: 76
Average instruction per cycle: 0.24918
Hit Count: 5
Miss Count: 24

N = 5, x = 0.4
Toal Cycles: 153
No of instruction executed: 76
Average instruction per cycle: 0.496732
Hit Count: 13
Miss Count: 16

N = 10, x = 0.4
Toal Cycles: 251
No of instruction executed: 76
Average instruction per cycle: 0.302789
Hit Count: 11
Miss Count: 18

N = 5, x = 0.8
Toal Cycles: 125
No of instruction executed: 76
Average instruction per cycle: 0.608
Hit Count: 20
Miss Count: 9

N = 10, x = 0.8
Toal Cycles: 134
No of instruction executed: 76
Average instruction per cycle: 0.567164
Hit Count: 24
Miss Count: 5

N = 5, x = 1
Toal Cycles: 89
No of instruction executed: 76
Average instruction per cycle: 0.853933
Hit Count: 29
Miss Count: 0

N = 10, x = 1
Toal Cycles: 89
No of instruction executed: 76
Average instruction per cycle: 0.853933
Hit Count: 29
Miss Count: 0