NWEN 242 Assignment3

Yongbo Yu ——ID:300390526

October 3, 2017

Q1

a

Pipelined:Cycle time determined by the slowest stage : 350ps
Single-cycle: Cycle time determined by sum of all stages : 1250ps

b

LW instruction uses all the five stages
Pipelined processor takes 5 cycles at 350ps per cycle for total latency of 1750ps.
Single-cycle processor takes (250+350+150+300+200)=1250ps.

c

Data memory is utilized only by LW and SW instructions. So the utilization is 35% of the clock cycles.

d

The utilization for this part is determined by R-type instructions and LW instructions.The utilization is 65%.

e

CPI for Muti-cycle machine =45% $\times$ 4 + 20% $\times$5+ 15% $\times$ 4 = 4

	Single-cycle	Multi-cycle	Pipeline
Cycle time	1250ps	350ps	350ps
CPI	1	4	1
Excution time	3.6	4	1

Q2

a

firstly we assume write first then read.

add r5,r2,r1
nop
nop
lw r3,4(r5)
lw r2,0(r2)
nop
or r3,r5,r3
nop

nop
sw r3.0(r5)

b

We can't have any performance gain by reordering these instructions. Because no matter how we rearrange the code there always has at least five "nop".

c

Yes, we still need to insert nops to ensure correct execution. With forwarding we only reduce the cycle bubbles. But we still need use 1 cycle bubble between lw and R-type to make the code work correctly.

Q3

a

I assume write first then read.
lw $t1, 4($t2)
nop
lw $t2 ,4($t1)
nop
add $t3,$t1, $t2
sw $t3, 8($t1)
lw $t4, 12($t0)
nop
add $t5, $t1,$t4
sw $t5, 16($t2)

b

lw $t1, 4($t2)
lw $t4, 12($t0)
lw $t2 ,4($t1)
add $t5, $t1,$t4
add $t3,$t1, $t2
sw $t5, 16($t2)
sw $t3, 8($t1)

After reordering the code, we don't need to insert nop to make the code works correctly, which will improve the performance.

c

lw $t4, 12($t0)
add $t5, $t1, $t4
sw $t5, 16($t2)
MEM/WB.RegisterRd=ID/EX.RegisterRs
EX/MEM.RegisterRd=ID/EX.RegisterRd

Q4

a

16: lw \$t2, 4(\$t1)
20: beq \$t2, \$t1, 10
...
xx: sw \$t2, 16(\$t2)
xx should be 64. Because as the instruction branch equal be executed then index become 24. Then as it shown above the offset is 10. Then xx= 10*4+24=64.

b

16: lw \$t2, 4(\$t1)
20: beq \$t2, \$t1, 10
24:nop
28:nop
Because we want the result from the beq instruction. But we can only have the result after the code fully executed which means we need to put ten nops between beq instruction and the next instruction.

c

lw \$t2, 4(\$t1)
beq \$t2, \$t1, 10
nop
sw \$t2, 16(\$t2)
With extra memory and fully forwarding we only need to insert one nop between beq and sw.

Q5

a

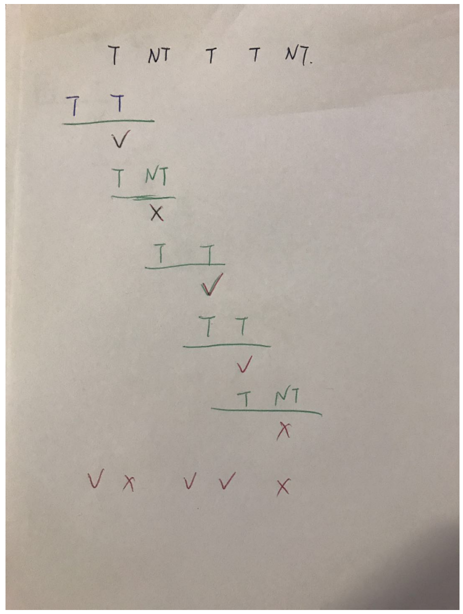
	T	NT	T	T	NT
	✓	✗	✗	✓	✗
T	T	NT	T	T	NT

From the table we can find the accuracy for always-taken is $\frac{2}{5}$=40%

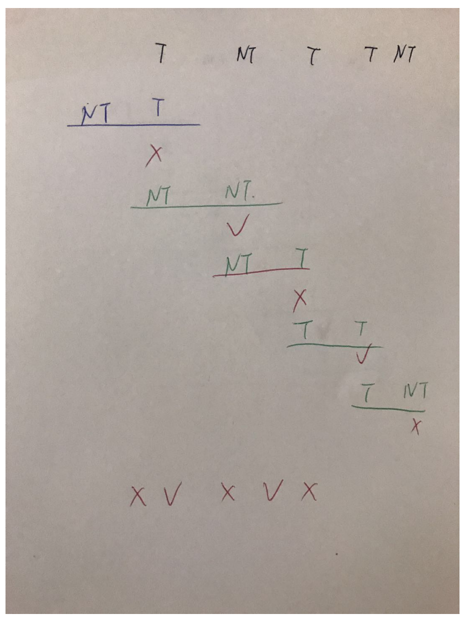
	T	NT	T	T	NT
	✗	✗	✗	✓	✗
NT	T	NT	T	T	NT

The accuracy for always-not-taken is $\frac{3}{5}$=20%

b



The accuracy of always-taken is 60%



The accuracy of always-not-taken is 40%

c