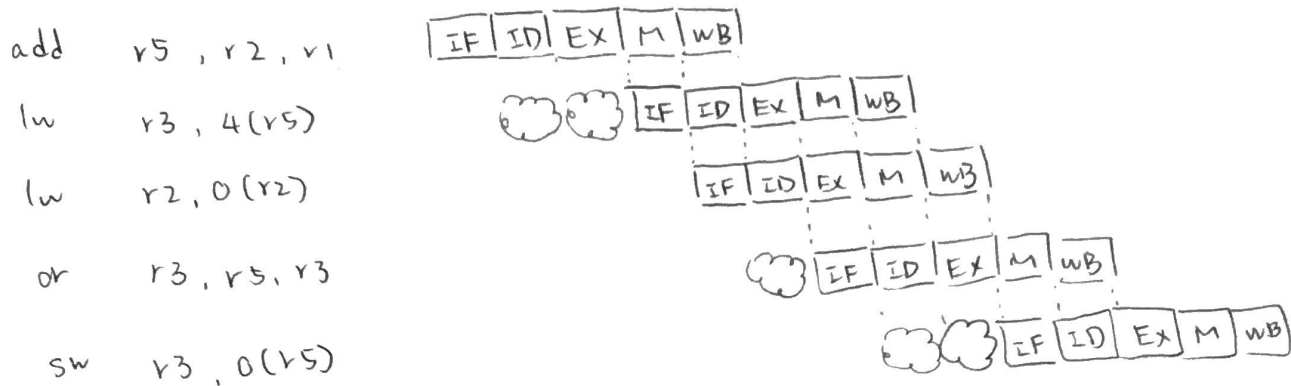


4.13.1



So correct execution with nops is :

```

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)
    
```

4.14.1

# Resolve in EX stage

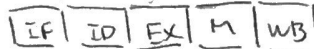
lw r2, 0(r1)



beq stalled (data hazard)



beq r2, r0, label2 → NT



sw becomes nop (control hazard)



whatever instruction after sw becomes nop (control hazard)



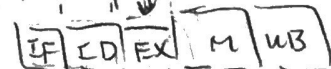
lw r3, 0(r2)



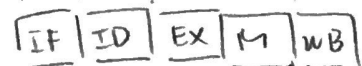
beq stalled (data hazard)



beq r3, r0, label1 → T



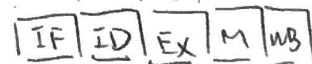
lw r2, 0(r1)



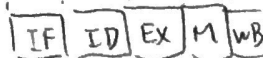
beq stalled (data hazard)



beq r2, r0, label2 → T



sw r1, 0(r2)



4.16.1

C = correct ; M = miss

always - taken

T	NT	T	T	NT
C	M	C	C	M

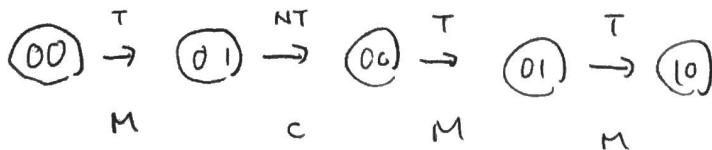
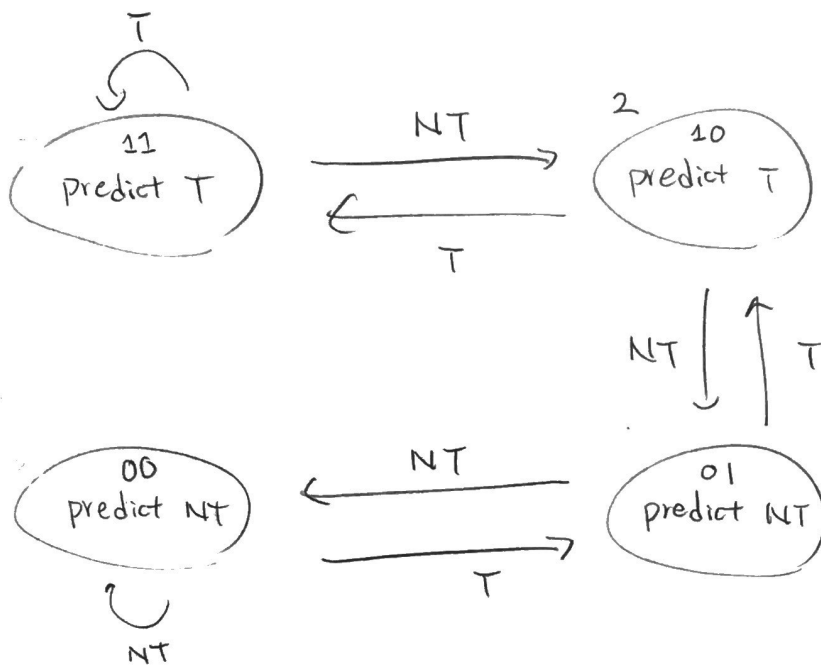
$$\frac{3}{5} \times 100 = 60 [\%]$$

always - not - taken

T	NT	T	T	NT
M	C	M	M	C

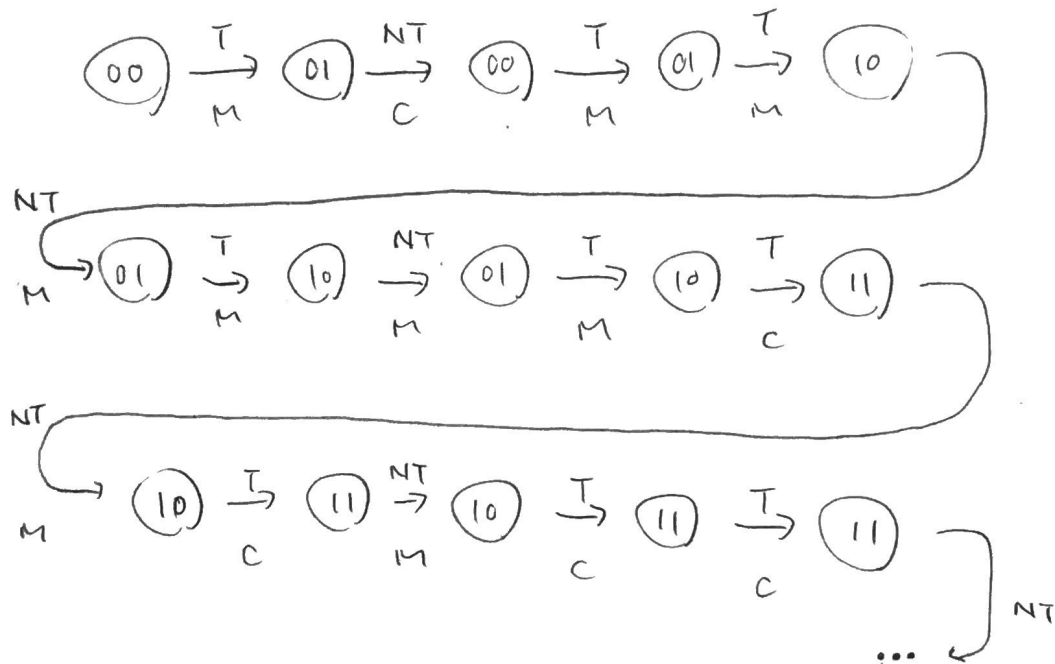
$$\frac{2}{5} \times 100 = 40 [\%]$$

4.16.2



$$\frac{1}{4} \times 100 = 25 \%$$

4.16.3



Eventually, we have C, M, C, C, M.

So the accuracy is  $\frac{3}{5} \times 100 = 60\%$

4.13. extra

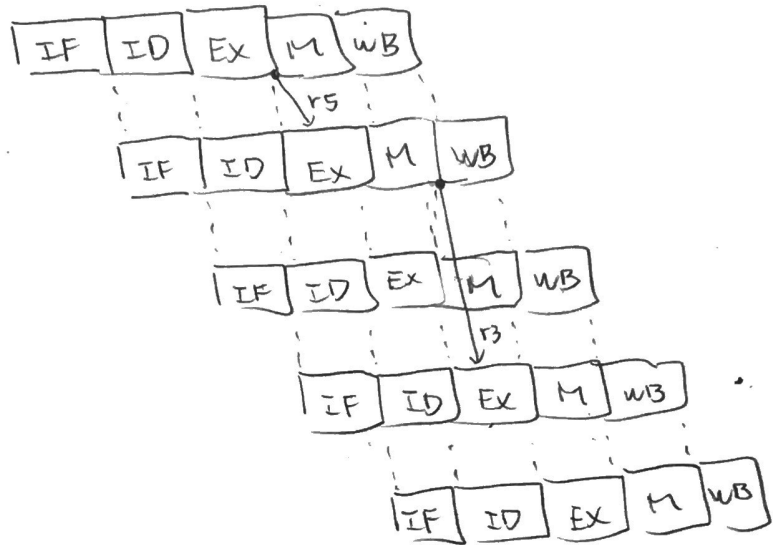
add r5, r2, r1

lw r3, 4(r5)

lw r2, 0(r2)

or r3, r5, r3

sw r3, 0(r5)



9 cycles

#### 4.14. extra

Resolve in ID stage

lw r2, 0(r1)

beq stalled (data hazard)

beq stalled (data hazard)

beq r2, r0, label2 → NT

sw becomes nop (control hazard)

lw r3, 0(r2)

beq stalled (data hazard)

beq stalled (data hazard)

beq r3, r0, label1 → T

lw r2, 0(r1)

beq stalled (data hazard)

beq stalled (data hazard)

beq r2, r0, label2 → T

sw r1, 0(r2)

