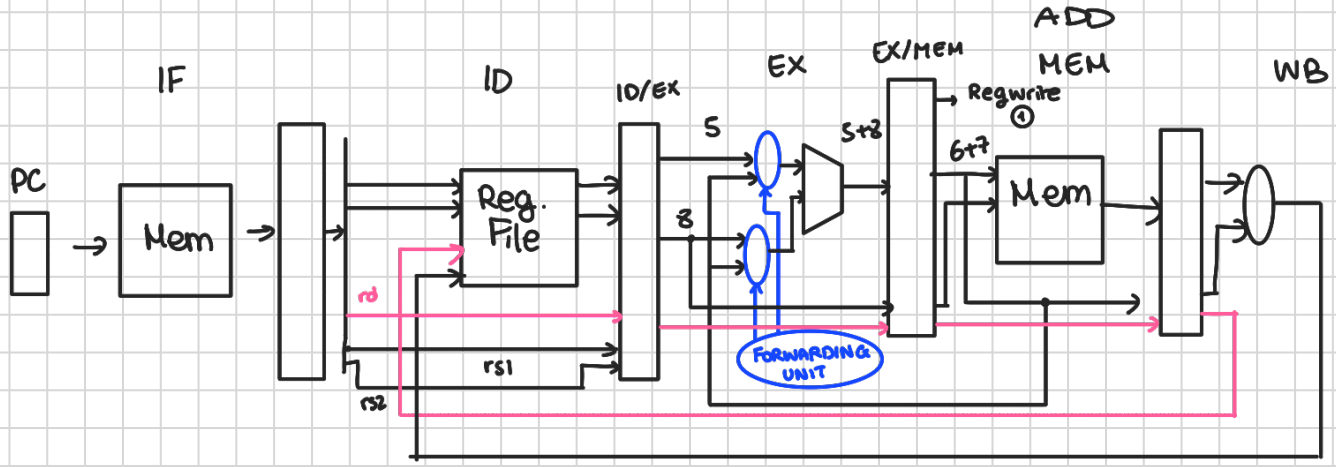
Comp. Arch - Unit 2



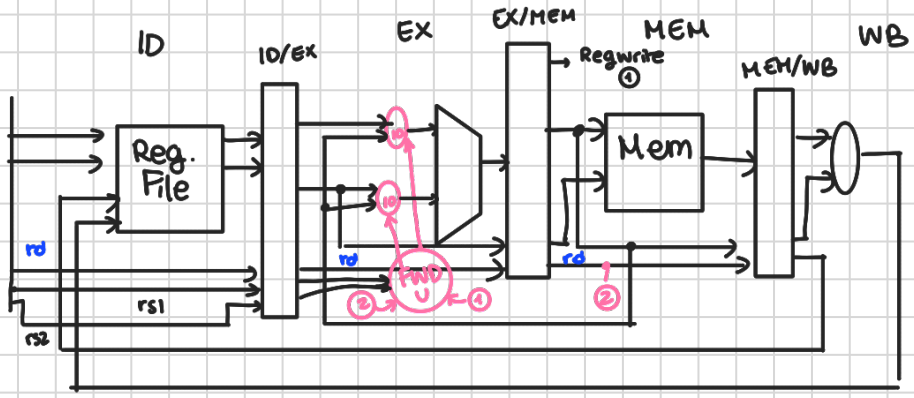
FORWARDING UNIT → controls both of the MUX's in such a way that

add 5, 6, 7
sub 9, 5, 8
} HAZARD

We need to make sure the values don't overlap

if its going to write and where

we need the n. of register that has been read



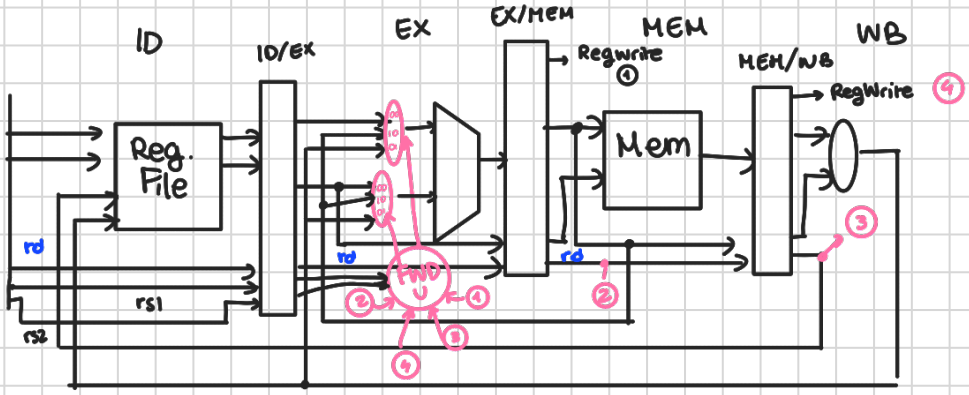
if EX/MEM. Regwrite and EX/MEM. rd ≠ 0
and EX/MEM. rd = ID/EX. rs1

forwardingA = 10

if EX/MEM. regWrite and EX/MEM. rd ≠ 0
and EX/MEM. rd = ID/EX. rs2

forwardingA = 10

```
add  5, 6, 7
sub  8, 9, 10
and  11, 5, 12
```

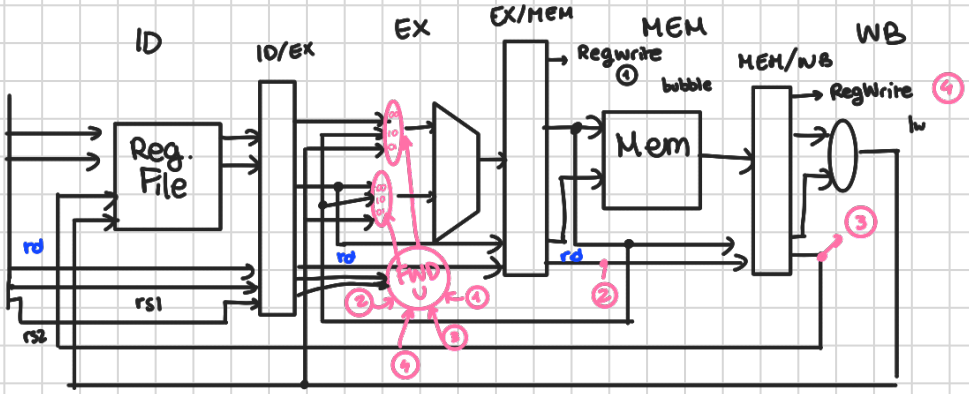


if MEM/WB.Regwrite and MEM/WB.rd ≠ 0 and MEM/WB.rd = rs1

forwardingA = 01

```
add  5, 6, 7
sub  5, 9, 10
and  11, 5, 12
```

```
lw  x5, 12(x8)
add x6, x5, x4
```

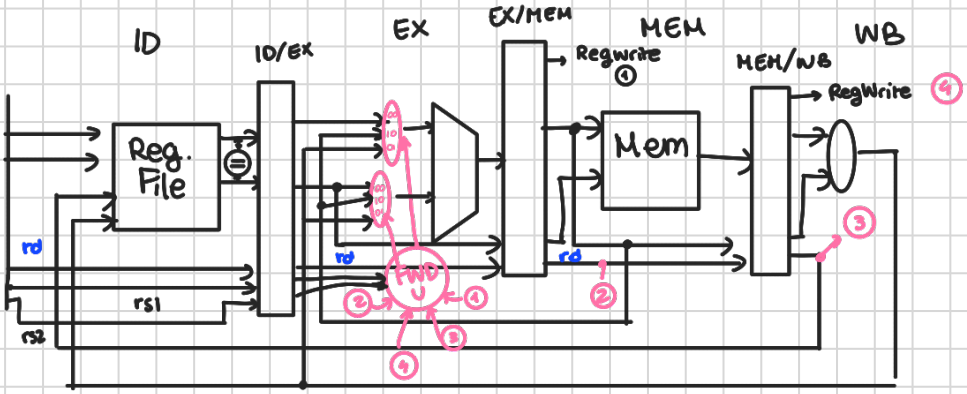



```
.data

    x: .word  5,6,7,-2,5
    n: .word  5

.text

    la  t0, x
    lw  t1, n
    lw  t2, 0
```

```
ciclo: lw  t3, 0(t0)   ← bubble
       add t2, t2, t3
       addi t0, t0, 4
         = t1, t1, -1          WB
       bne t1, zero, ciclo     MEM
                               EX
```

→

```
ciclo: lw  t3, 0(t0)
       add t2, t2, t3
       lw  t4, 4(t0)
       add t2, t2, t4
       addi t0, t0, 8
       addi t1, t1, -2
       bne t1, zero, ciclo
```



ciclo: lw t3, 0(t0) (A)
add t2, t2, t3 (B)
addi t0, t0, 4 (C)
addi t1, t1, -1 (D)
bne t1, zero, ciclo (E)

(A, B) (B, E) HAZARDS

↓ BUBBLE ↓ BUBBLE