

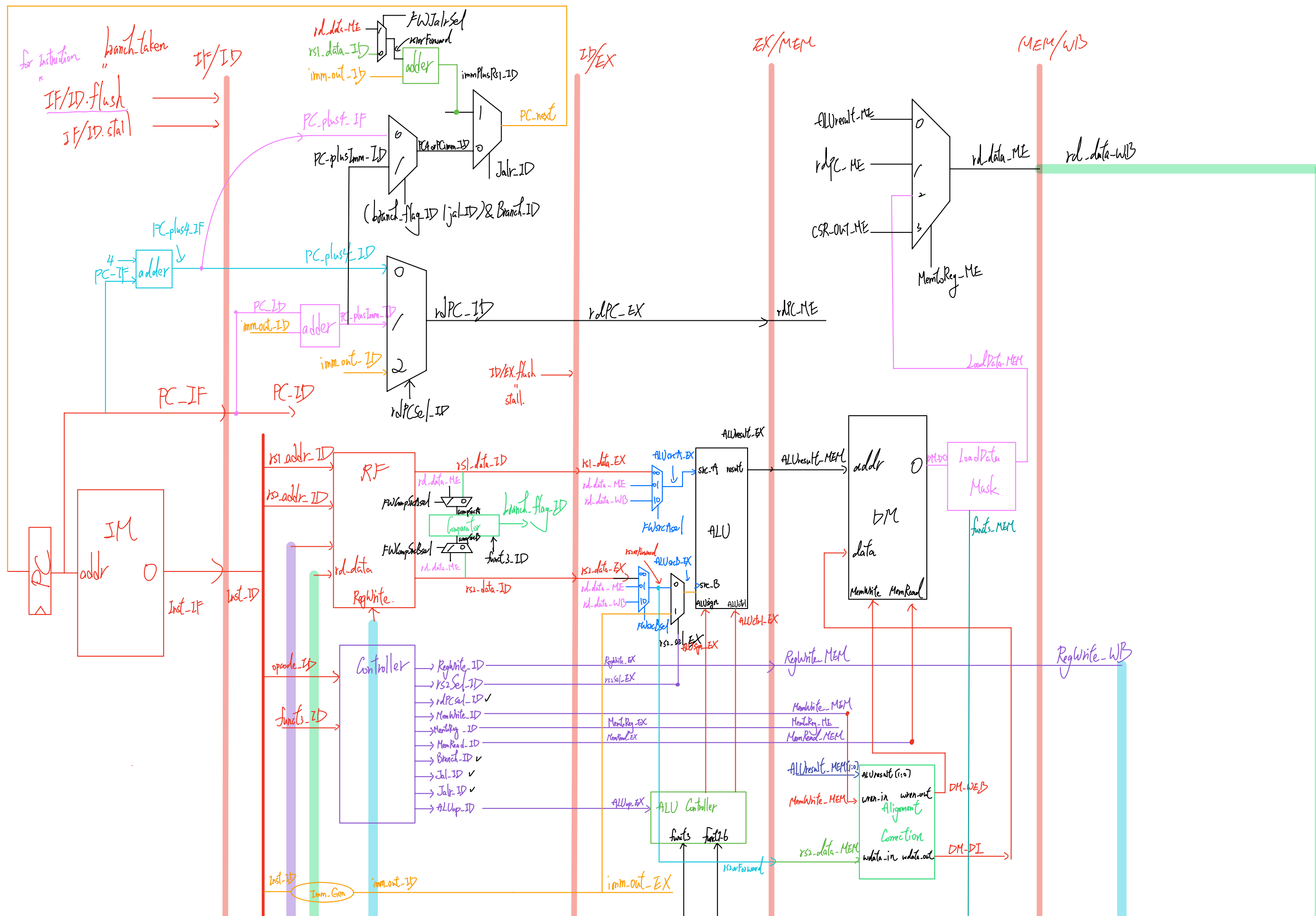
IF

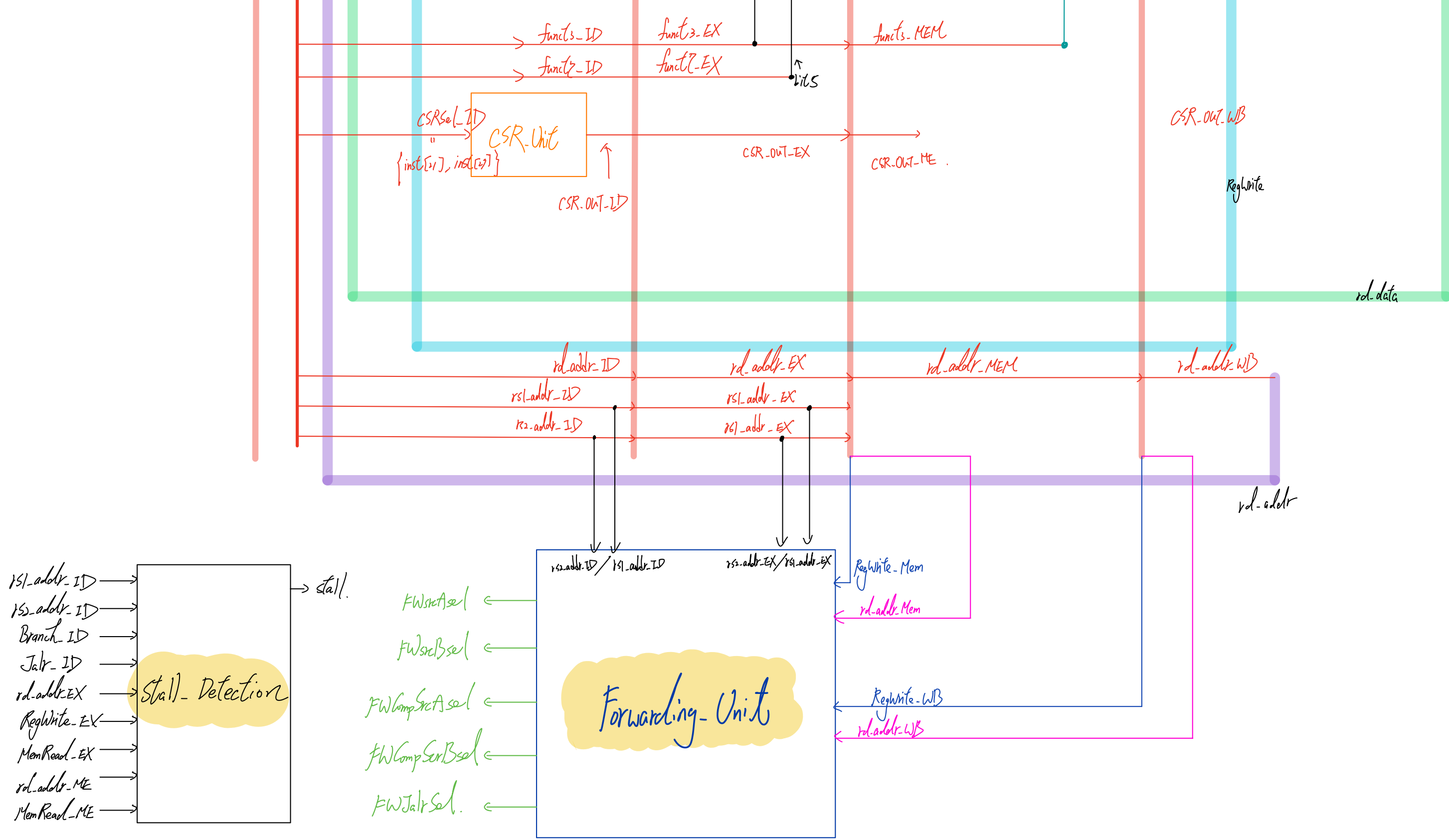
ID

EXE

MEM

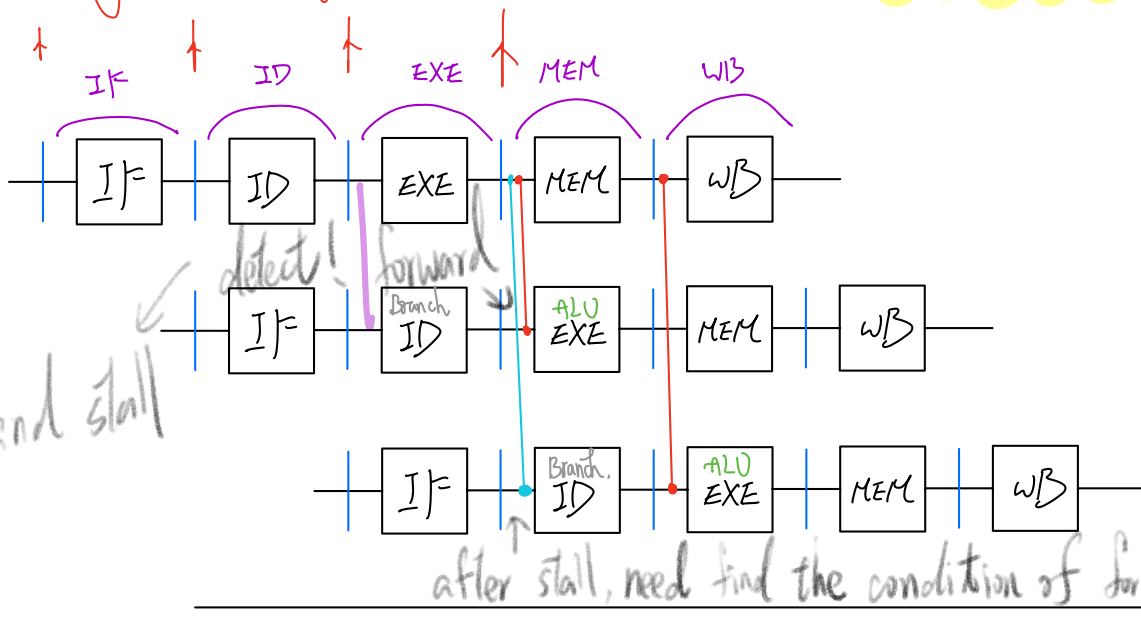
WB





Note: 若 WB stage 要寫回 rd 時, 剛好有指令 read rd,
則會來不及, 可以讓 RF 的輸出-輸入來解決須
等待 - cycle 的問題!

Only forwarding



R-R or R-X-R

if (RegWrite-MEM == 1 && rd_addr-MEM != 0)
 if (rd_addr-MEM == rs1_addr-EX) ALUforwarding-srcA = ALUresult-MEM **FWsrcAse1 = 1'b01**
 if (rd_addr-MEM == rs2_addr-EX) ALUforwarding-srcB = ALUresult-MEM **FWsrcBse1 = 1'b01**
 else if (RegWrite-WB == 1 && rd_addr-WB != 0)
 if (rd_addr-WB == rs1_addr-EX) ALUforwarding-srcA = ALUresult-WB **FWsrcAse1 = 1'b10**
 if (rd_addr-WB == rs2_addr-EX) ALUforwarding-srcB = ALUresult-WB **FWsrcBse1 = 1'b10**

因為ID才decode出來，所以上一條用EX去寫。

R-Branch

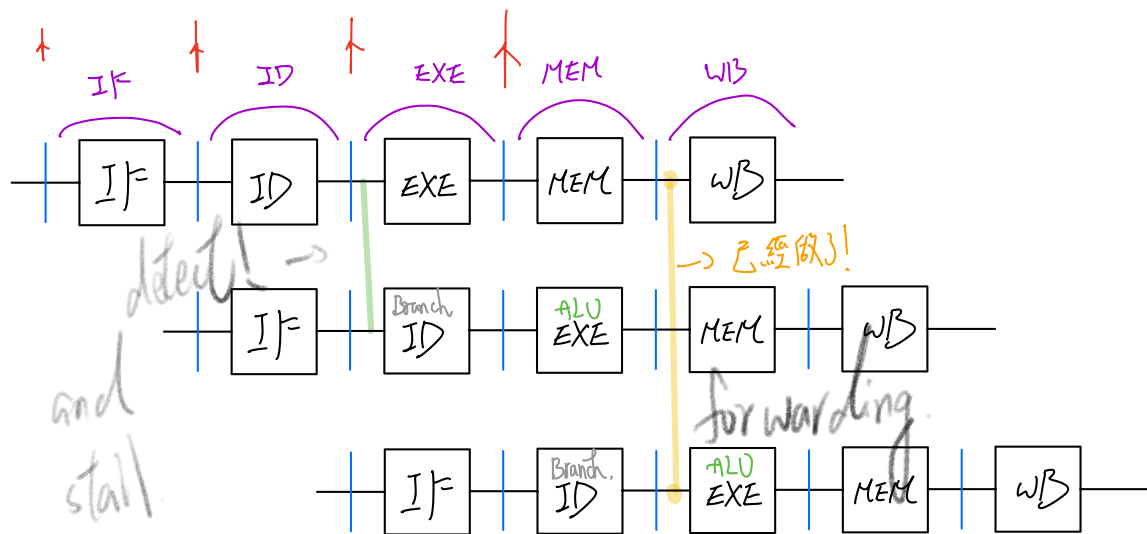
need to stall for one cycle, mind the flush!!

stall 1 cycle & forwarding

if (RegWrite-MEM == 1 && rd_addr-MEM != 0) {
 if (rd_addr-MEM == rs1_addr-ID) **FWCompSrcAse1 = ALUresult-MEM**
 if (rd_addr-MEM == rs2_addr-ID) **FWCompSrcBse1 = ALUresult-MEM**
 }

if (RegWrite-EX == 1 && (Branch-ID | Jalr-ID)) {
 if (rd_addr-EX == rs1_addr-ID || rd_addr-EX == rs2_addr-ID)
 stall = 1
 else
 stall = 0.
 }

Load-R => need stall one cycle.



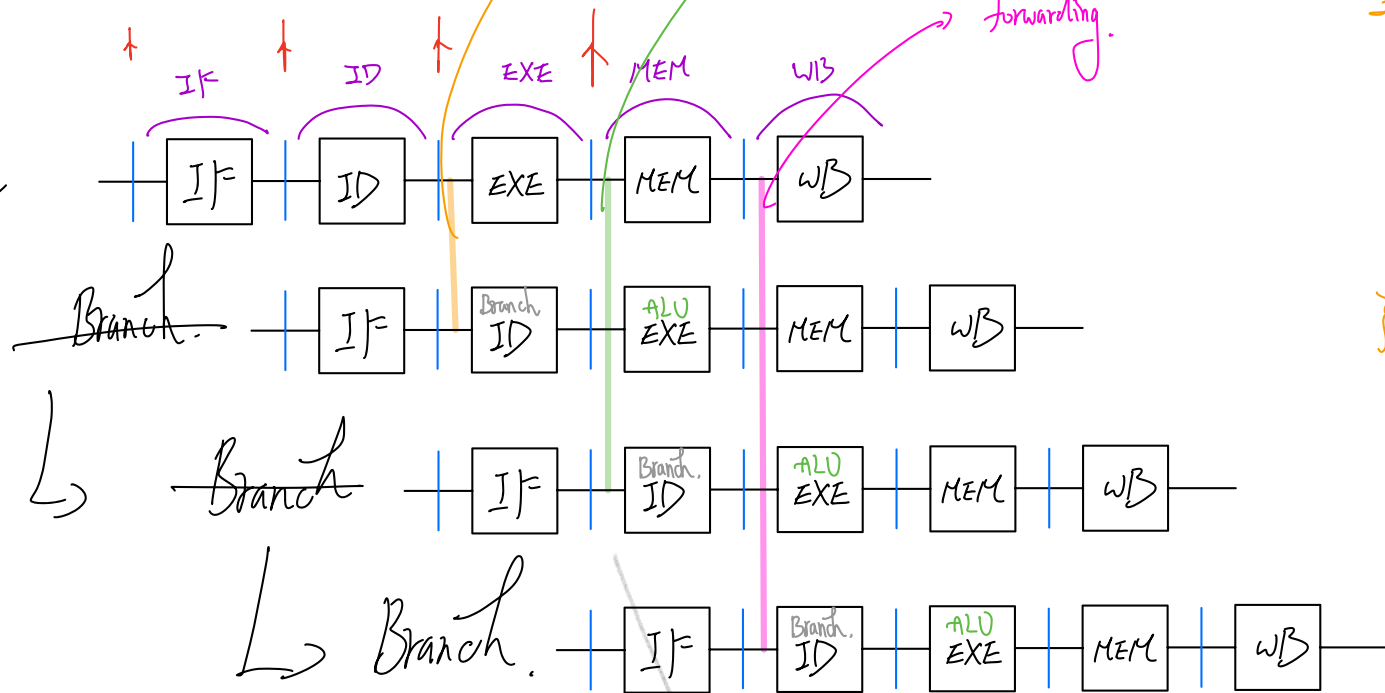
if (MemRead-EX == 1 && RegWrite-EX == 1) {
 if (rd_addr-EX == rs1_addr-ID || rd_addr-EX == rs2_addr-ID)
 stall = 1
 else
 stall = 0
 }

Same.

stall 1 cycle & forwarding.

Load-Branch.

Load



already forwarding in other condition (R-R, R-B)

stall 2 cycle & forwarding

stall first time. (S)

```

if (memRead-EX == 1 && RegWrite-EX == 1) {
    if (rd-addr-EX == rs1-addr-ID || rd-addr-EX == rs2-addr-ID)
        stall = 1;
    else
        stall = 0;
}
    
```

stall second time. (S)

```

if (memRead-MEM == 1 && (Branch-ID || Jalr-ID)) {
    if (rd-addr-MEM == rs1-addr-ID || rd-addr-MEM == rs2-addr-ID)
        stall = 1;
    else
        stall = 0;
}
    
```

X