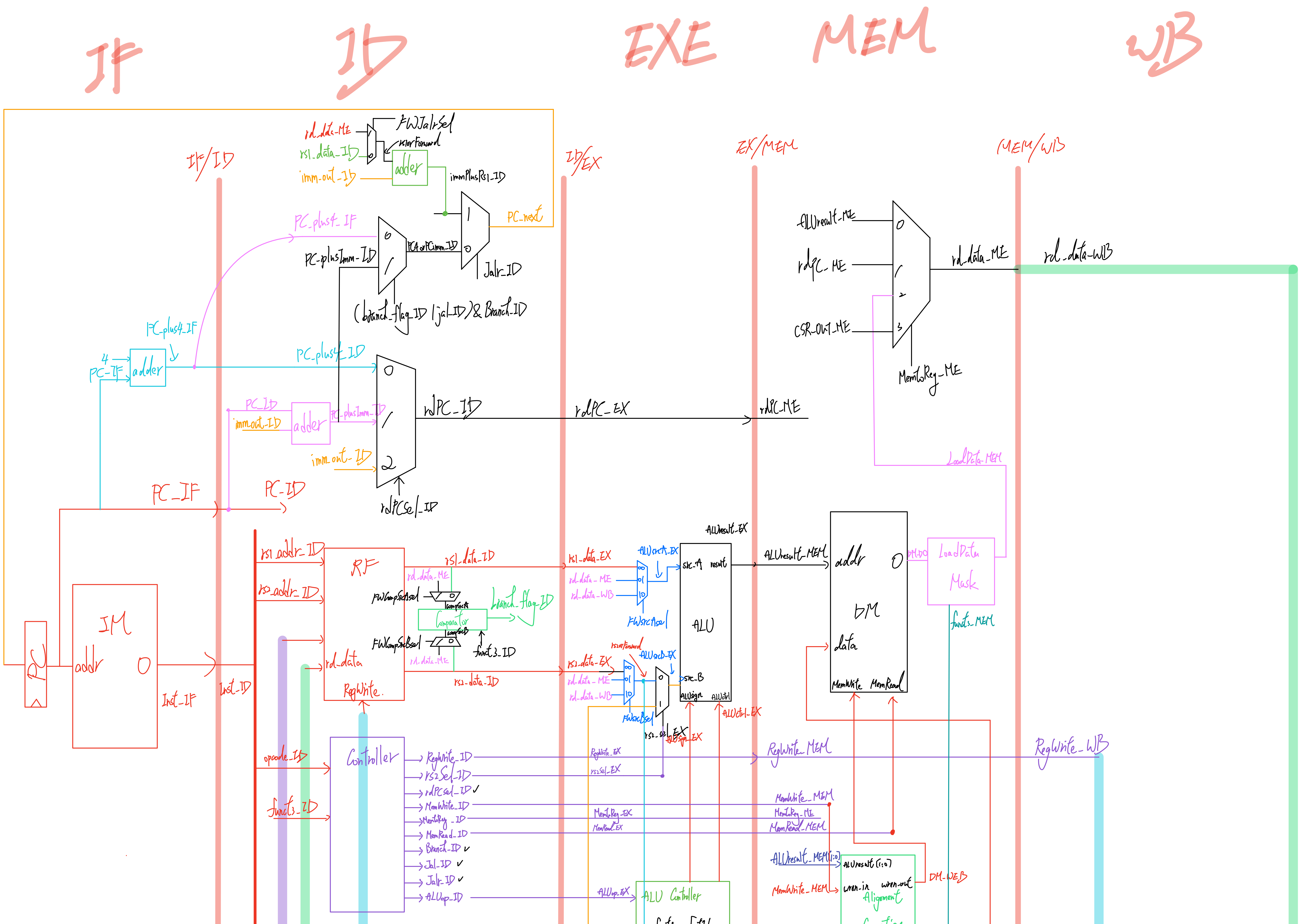
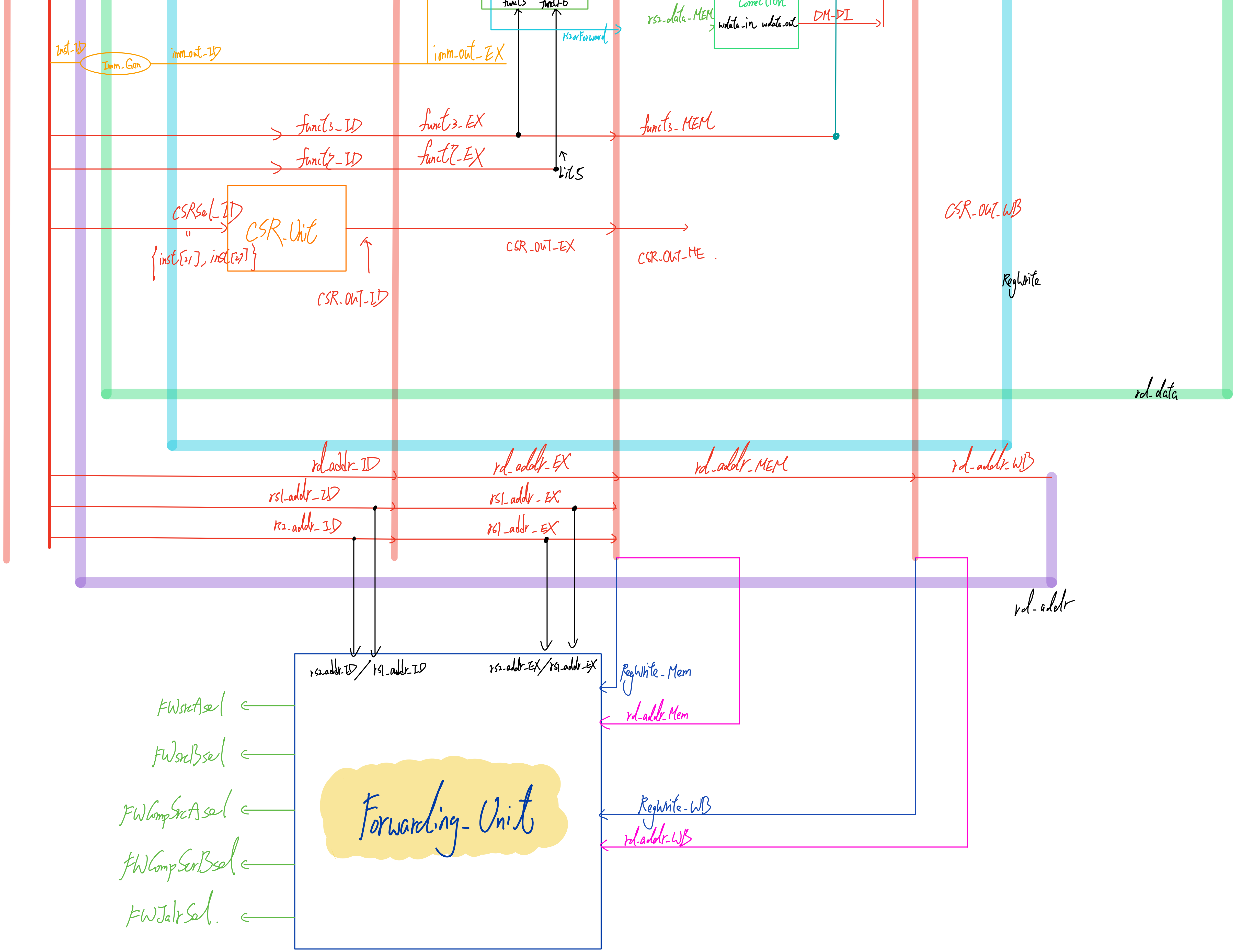

IF
ID
EXE
MEM
WB
IF/ID
ID/EX
EX/MEM
MEM/WB
rd_data_ME
rs1_data_ID
imm_out_ID
FWJalrSel
JalrForward
adder
immPlusRs1_ID
PC_next
PC_plus4_IF
PC_plusImm_ID
PCorPCimm_ID
Jalr_ID
(branch_flag_ID | jal_ID) & Branch_ID
1
0
PC_plus4_IF
4
PC_IF
adder
PC_plus4_ID
PC_ID
imm_out_ID
adder
PC_plusImm_ID
imm_out_ID
0
1
2
rdPC_ID
rdPC_EX
rdPC_ME
rdPCSel_ID
PC_IF
PC_ID
PC
IM
addr
O
Inst_IF
Inst_ID
rs1_addr_ID
rs2_addr_ID
RF
rd_data
RegWrite
rs1_data_ID
rd_data_ME
FWCompSrcASel
compSrcA
Comparator
compSrcB
branch_flag_ID
FWCompSrcBSel
rd_data_ME
funct3_ID
rs2_data_ID
rs1_data_EX
rd_data_ME
rd_data_WB
ALUsrcA_EX
FWsrcASel
rs2_data_EX
rs2orForward
ALUsrcB_EX
rd_data_ME
rd_data_WB
FWsrcBSel
rs2Sel_EX
ALUsign_EX
src_A
result
src_B
ALU
ALUsign
ALUctrl
ALUctrl_EX
ALUresult_EX
ALUresult_MEM
addr
O
DM
data
MemWrite
MemRead
DM_DO
LoadData_MEM
LoadData Mask
funct3_MEM
ALUresult_ME
rdPC_ME
CSR_OUT_ME
0
1
2
3
MemToReg_ME
rd_data_ME
rd_data_WB
opcode_ID
funct3_ID
Controller
RegWrite_ID
rs2Sel_ID
rdPCSel_ID ✓
MemWrite_ID
MemToReg_ID
MemRead_ID
Branch_ID ✓
Jal_ID ✓
Jalr_ID ✓
ALUop_ID
RegWrite_EX
rs2Sel_EX
MemToReg_EX
MemRead_EX
RegWrite_MEM
MemWrite_MEM
MemToReg_ME
MemRead_MEM
RegWrite_WB
ALUop_EX
ALU Controller
ALUresult_MEM[1:0]
ALUresult[1:0]
MemWrite_MEM
wen_in
wen_out
Alignment
DM_WEB

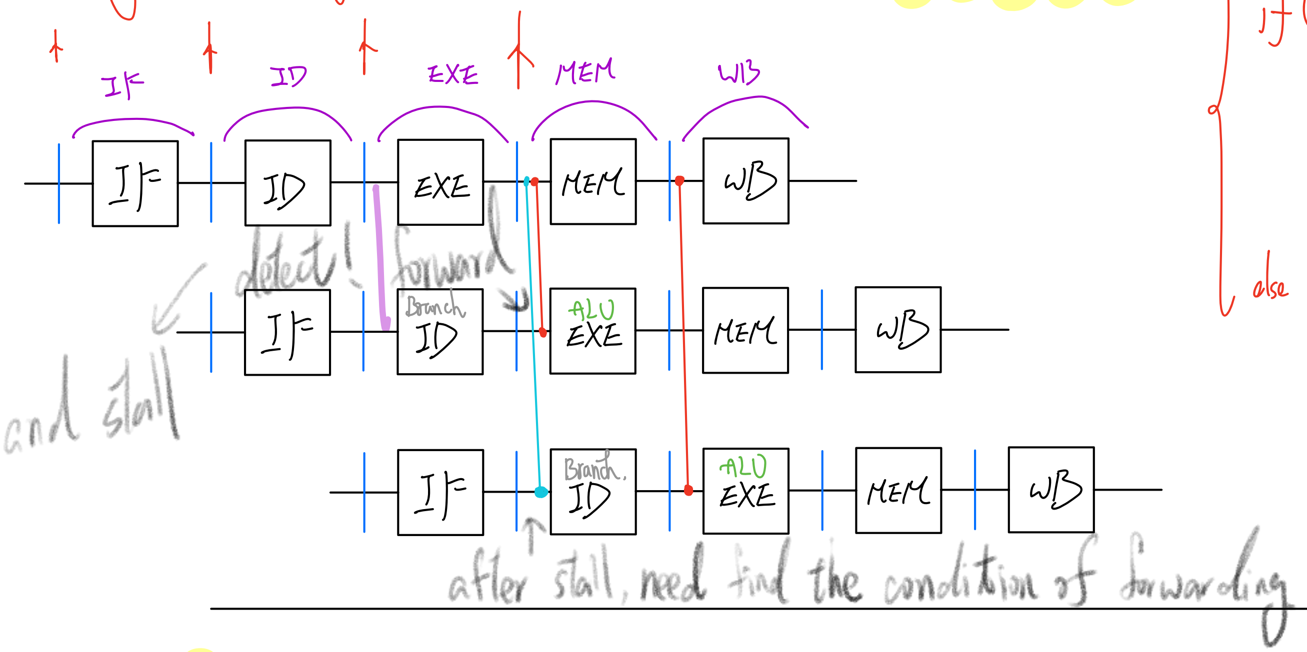


Note：若 WB stage 要寫回 rd 時，剛好有指令 read rd，

則會來不及，可以讓 RF 的輸出=輸入來解決須

等待一 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   FWsrcAsel = 2'b01
    if(rd_addr_MEM == rs2_addr_EX) ALUforwarding_srcB = ALUresult-MEM  FWsrcBsel = 2'b01

else if(RegWrite_WB==1 && rd_addr_WB ≠0)
    if(rd_addr_WB == rs1_addr_EX) ALUforwarding_srcA = ALUresult_WB   FWsrcAsel = 2'b10
    if(rd_addr_WB == rs2_addr_EX) ALUforwarding_srcB = ALUresult_WB   FWsrcBsel = 2'b10
```

R-Branch

need to stall for one cycle.

mind the flush!!

Ⓕ

```
if (RegWrite_MEM ==1 && rd_addr_MEM≠0) {
    if(rd_addr_MEM ==rs1_addr_ID)   FWCompSrcAsel = ALUresult_MEM
    if(rd_addr_MEM == rs2_addr_ID)  FWCompSrcBsel = ALUresult_MEM
}
```

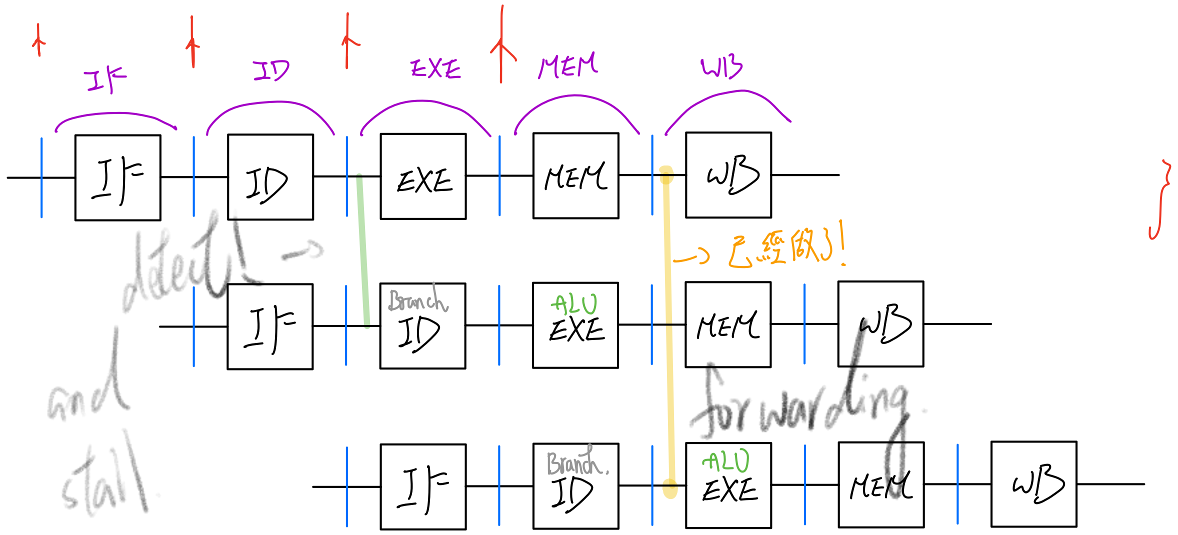
因為ID才decode出來，所以上一條用EX去比.

Ⓢ

```
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 .
}
```

stall 1 cycle & forwarding

Load-R => need stall one cycle.



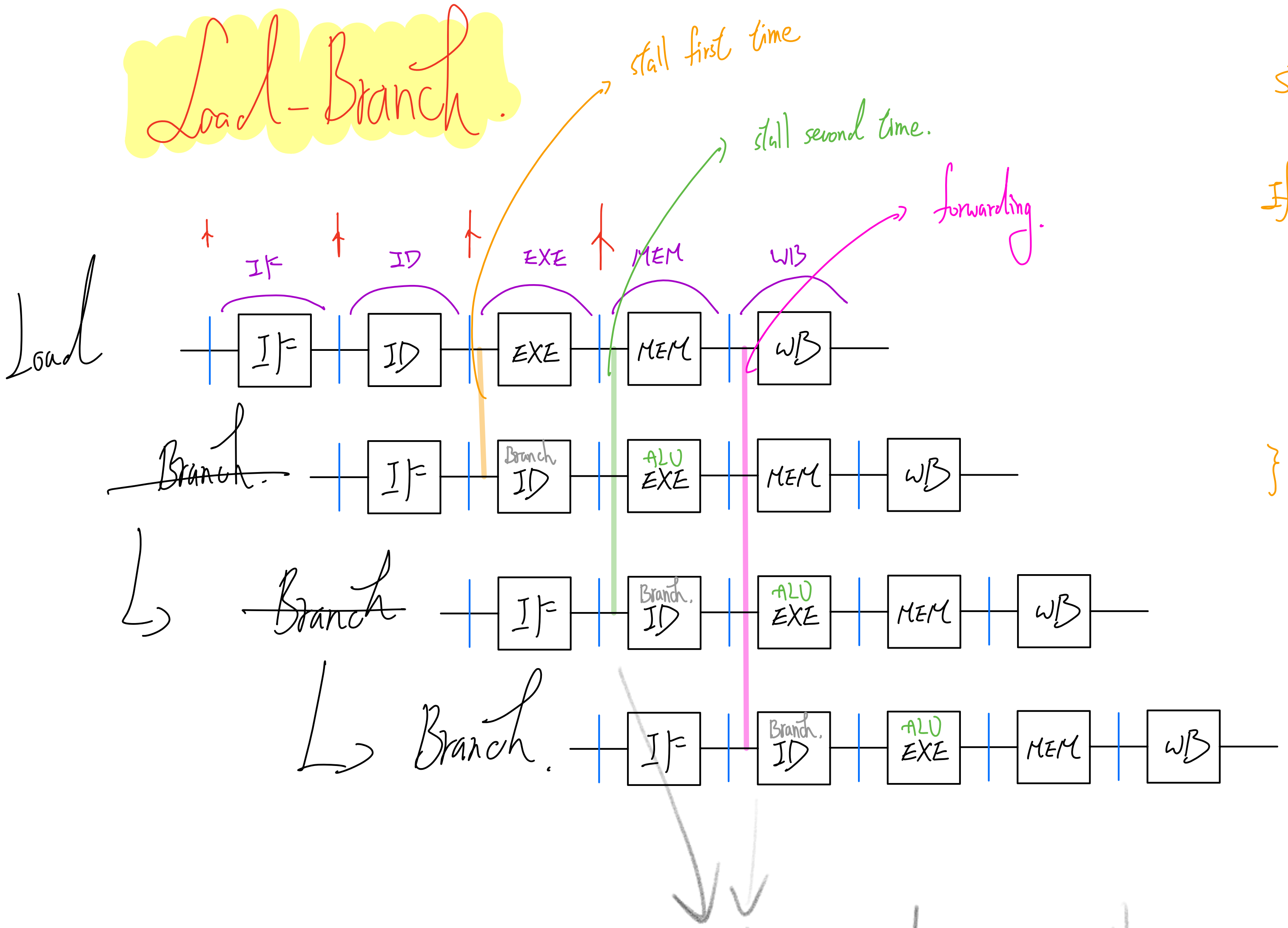
Ⓢ

```
                                    (X)
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.



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

stall first time. Ⓢ1

```
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
}
```

Ⓢ2

stall second time.

```
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;
}
```

stall 2 cycle & forwarding