

```
squash_D = val_X && (op_X == bne) && !eq_X
squash_F = squash_D || (val_D && ((op_D == jal) || (op_D == jr)))
```

```

stall_load_use_X_rs1_D =
    val_D && rs1_en_D && val_X && rf_wen_X
        && (inst_rs1_D == rf_waddr_X) && (rf_waddr_X != 0)
        && (op_X == lw)

stall_load_use_X_rs2_D =
    val_D && rs2_en_D && val_X && rf_wen_X
        && (inst_rs2_D == rf_waddr_X) && (rf_waddr_X != 0)
        && (op_X == lw)

```

stall\_D =  
 val\_D && ( stall\_load\_use\_X\_rs1\_D || stall\_load\_use\_X\_rs2\_D )  
  
 bypass\_waddr\_X\_rs1\_D =  
 val\_D && rs1\_en\_D && val\_X && rf\_wen\_X  
 && (inst\_rs1\_D == rf\_waddr\_X) && (rf\_waddr\_X != 0)  
 && (op\_X != lw)  
  
 bypass\_waddr\_X\_rs2\_D =  
 val\_D && rs2\_en\_D && val\_X && rf\_wen\_X  
 && (inst\_rs2\_D == rf\_waddr\_X) && (rf\_waddr\_X != 0)  
 && (op\_X != lw)

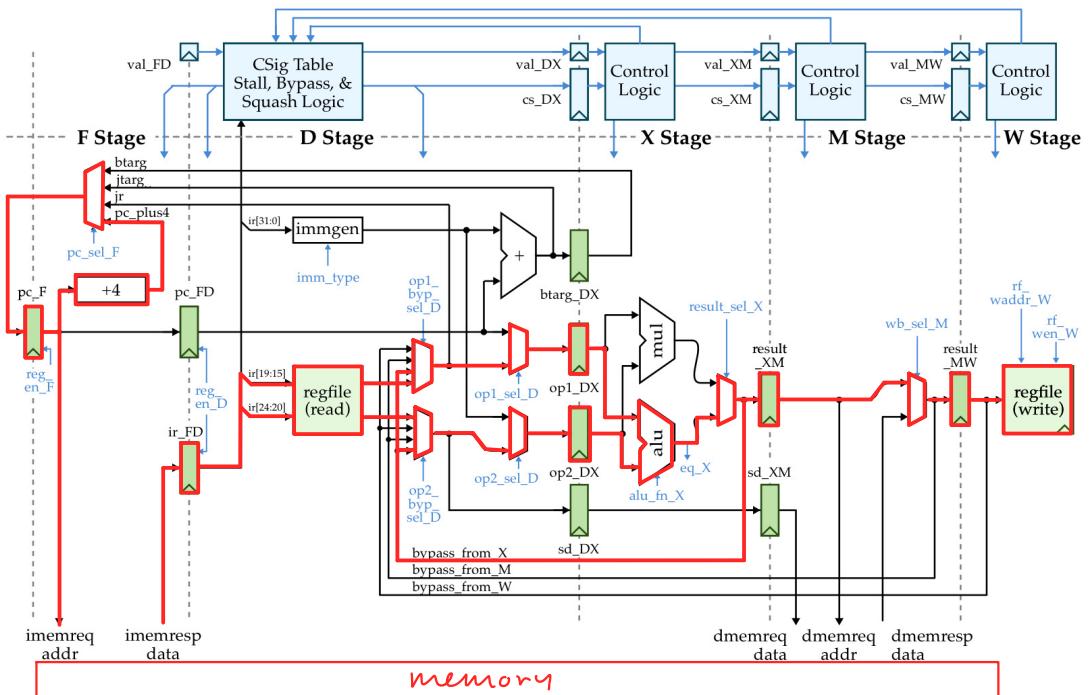
add  $(x_1)$   $x_1$   $x_0$  F D X M W  
 add  $(x_0)$   $(x_1)$   $x_0$  F D X M W  
 add  $x_1$   $(x_1)$   $x_0$  F D X M W

↙ Similar for M, W

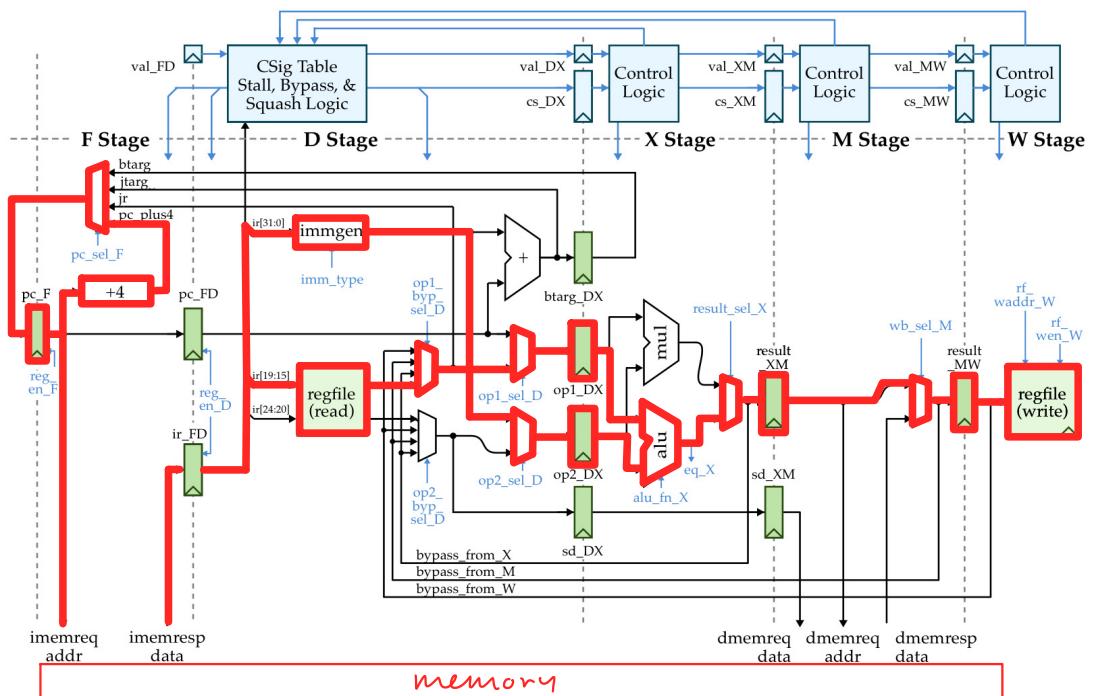
x M w  
c'malid?

prioritize closer bypass path!

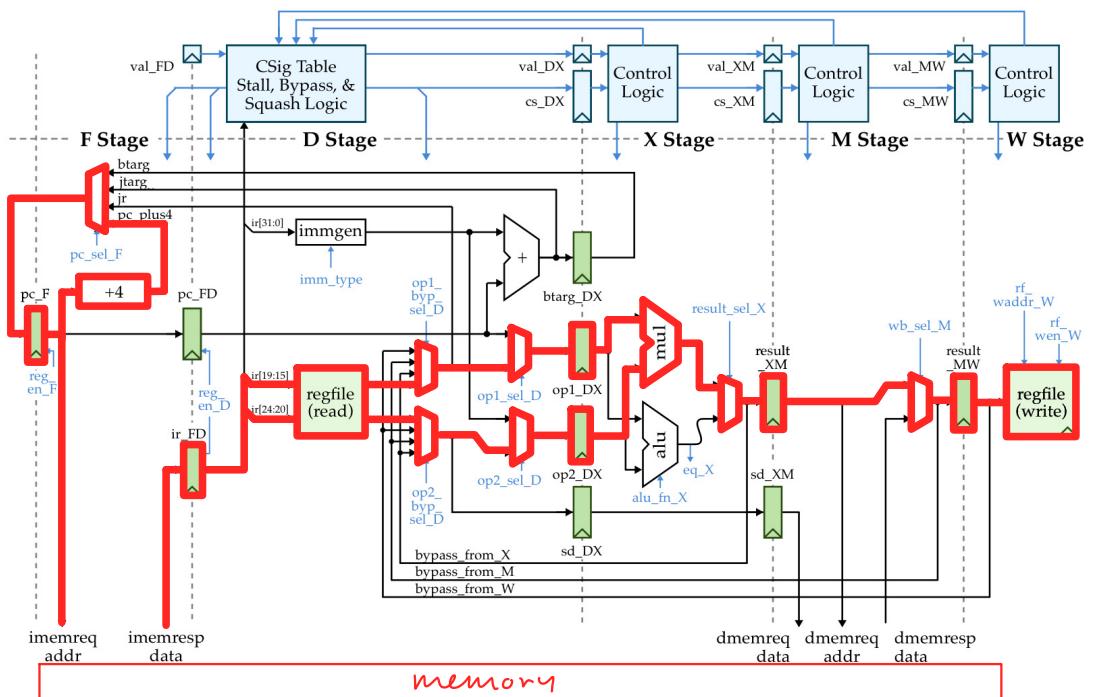
# ADD



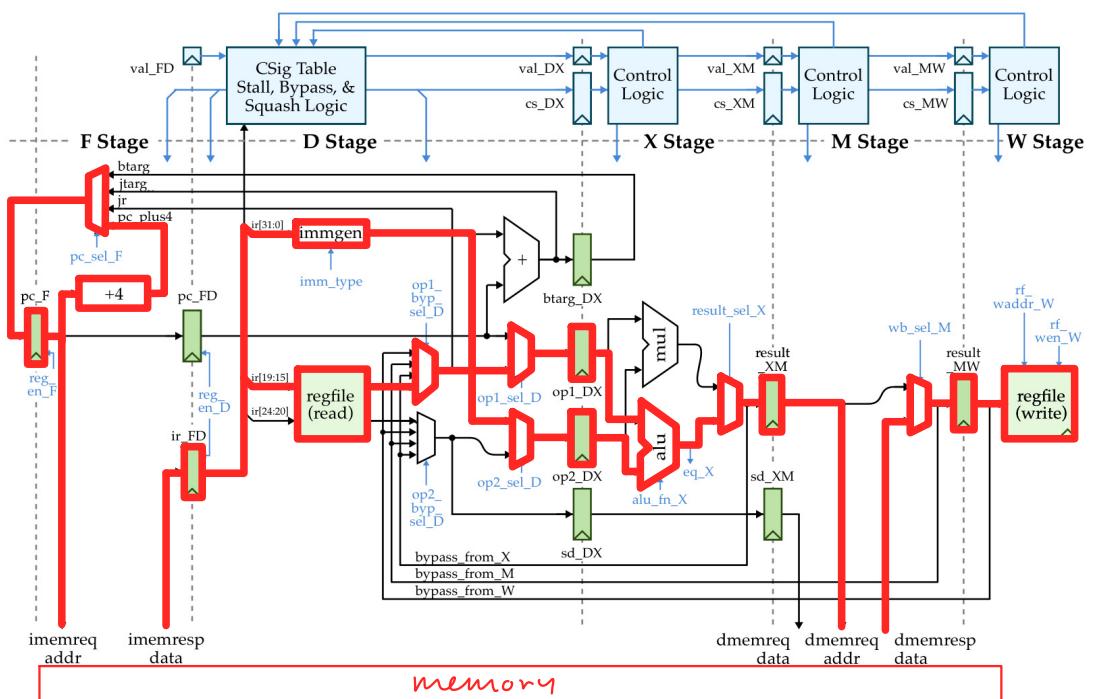
# ADDI



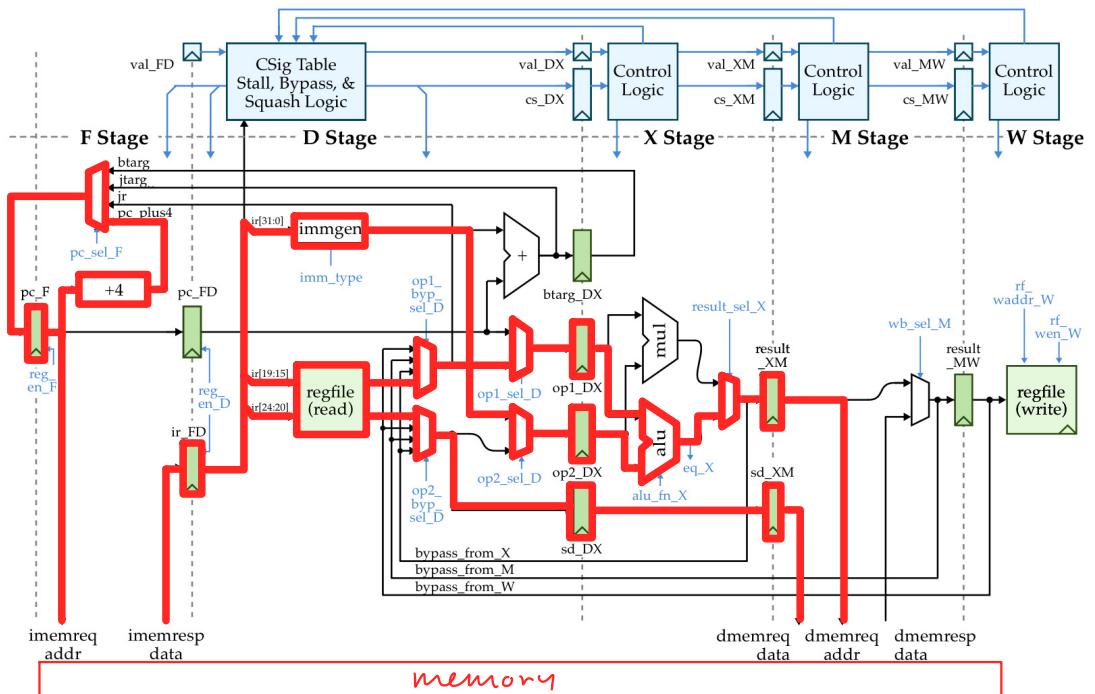
# MVL



LW

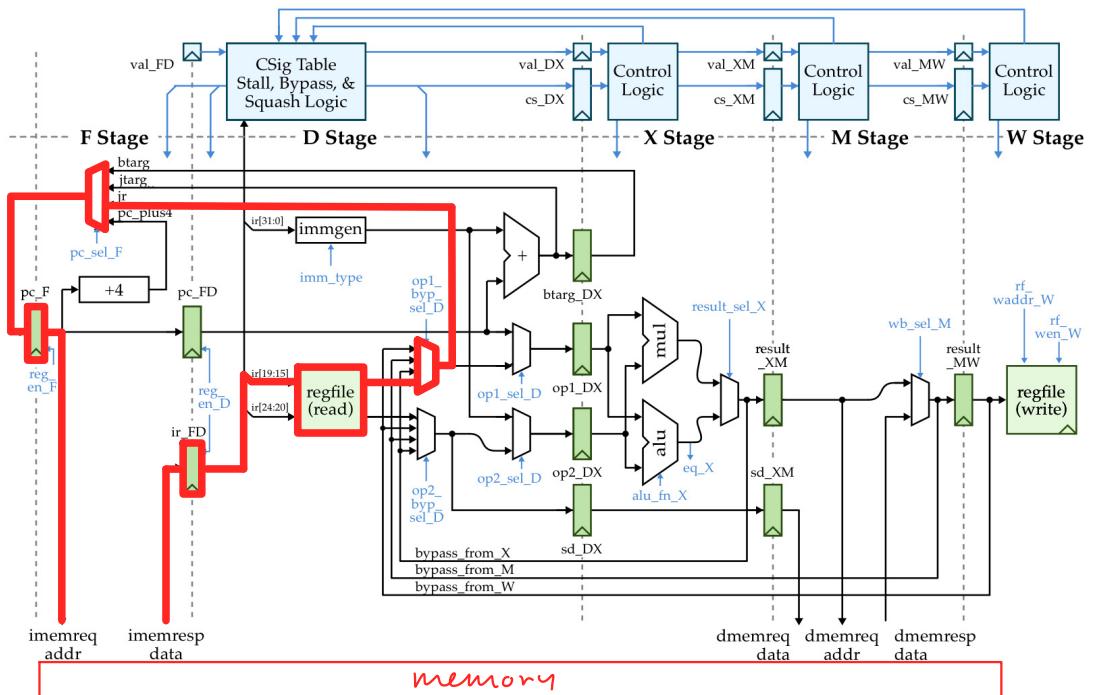


SW



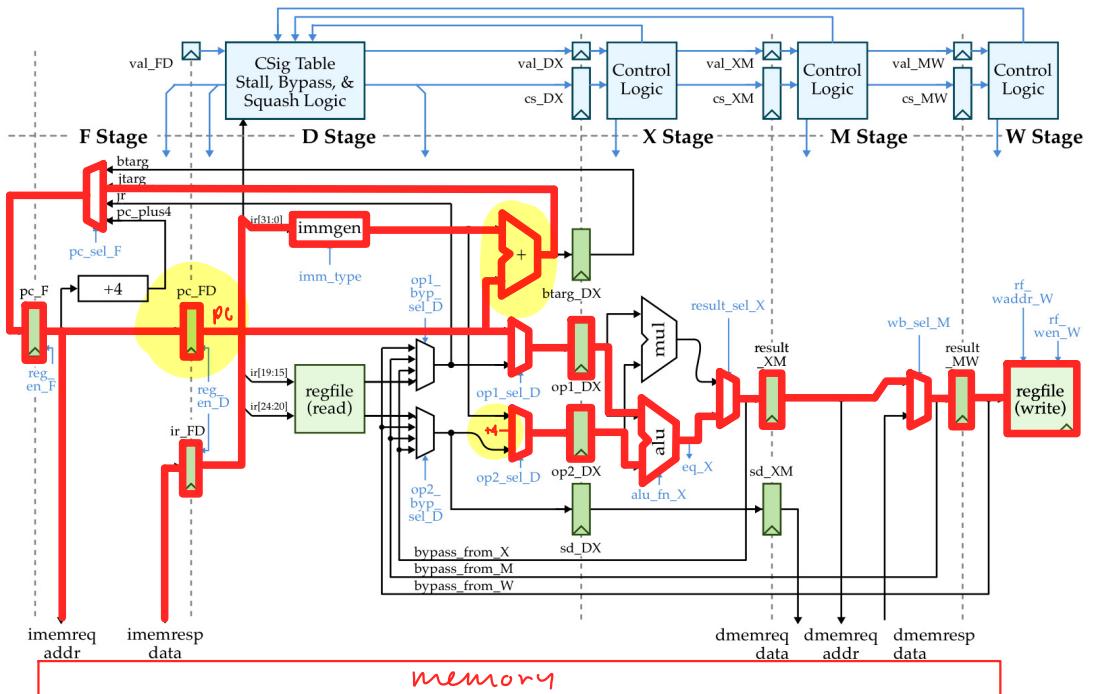
sw rs2 imm(rs1)

JR



jr rs1

# JAL



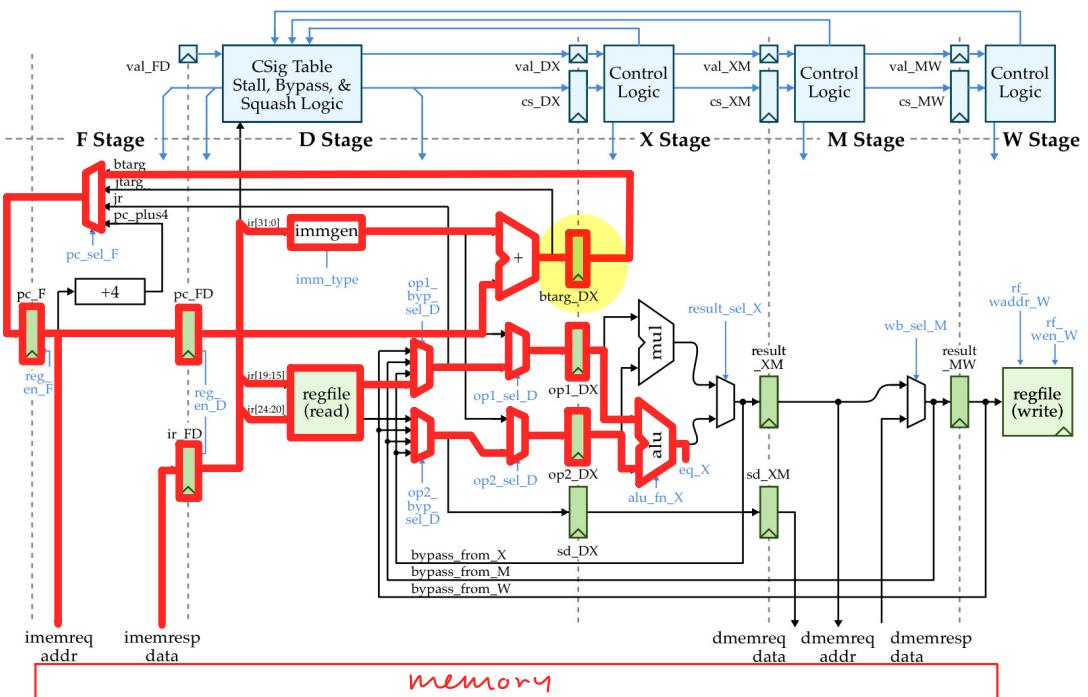
jal rd addr

$$R[rd] \leftarrow PC + 4$$

$$PC \leftarrow \underbrace{addr}_{}$$

$$PC + sext(cimm)$$

BNE



bne rs1 rs2 addr

if RCrs1] == RCrs2]

$PC \leftarrow PC + 4$

if  $R[rs1] \neq R[rs2]$

$\text{PC} \leftarrow \text{addr}$

PC+ Sext (imm)

# CSR R/W

