

```
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)
```

*Handwritten notes:*  
 jr x1 F D X M W  
 opA  
 op x1

*Handwritten notes:*  
 bne x4, x2, L1 F D X M W  
 opA  
 opB

*Handwritten notes:*  
 add x1, x0 F D X M W  
 add x1, x0 F D X M W  
 add x1, x0 F D X M W

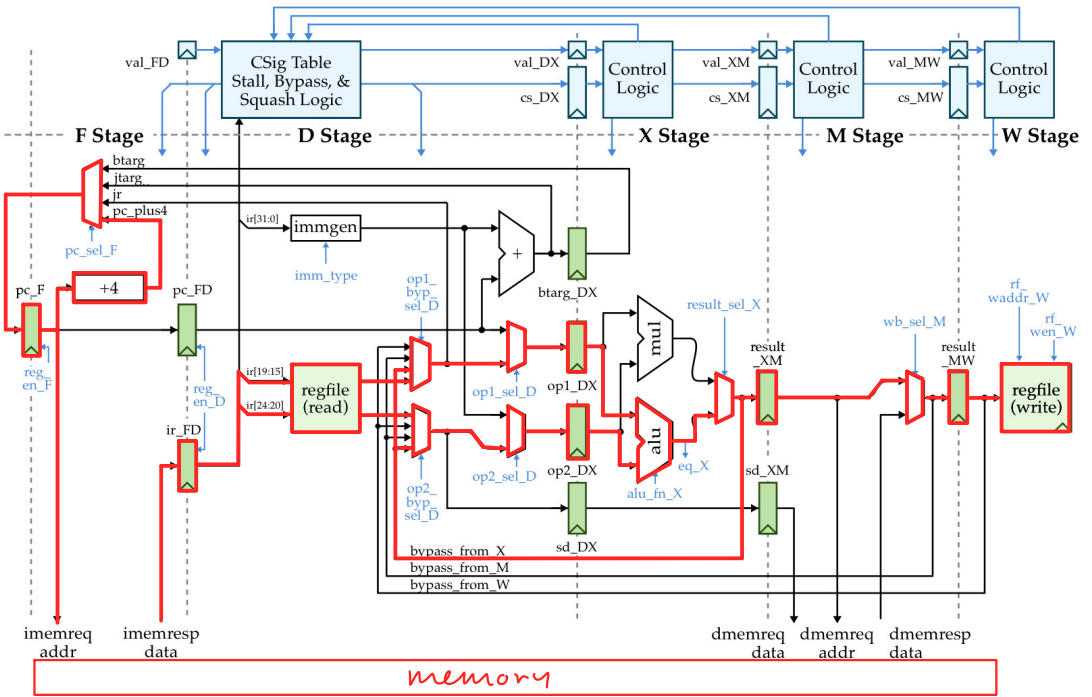
*Handwritten notes:*  
 lw x1, 0(x0) F D X M W  
 add x1, x1, x0 F D D X M W

*Handwritten note:*  
 similar for M, W

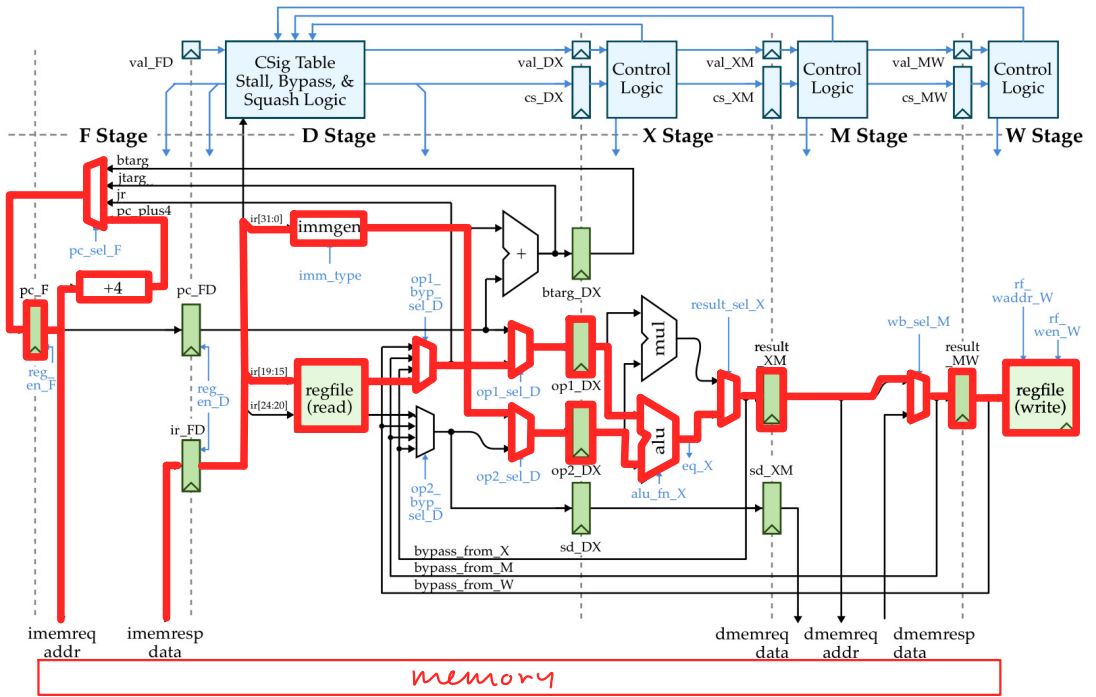
*Handwritten note:*  
 X M W (invalid?)

*Handwritten note:*  
 prioritize closer bypass path!

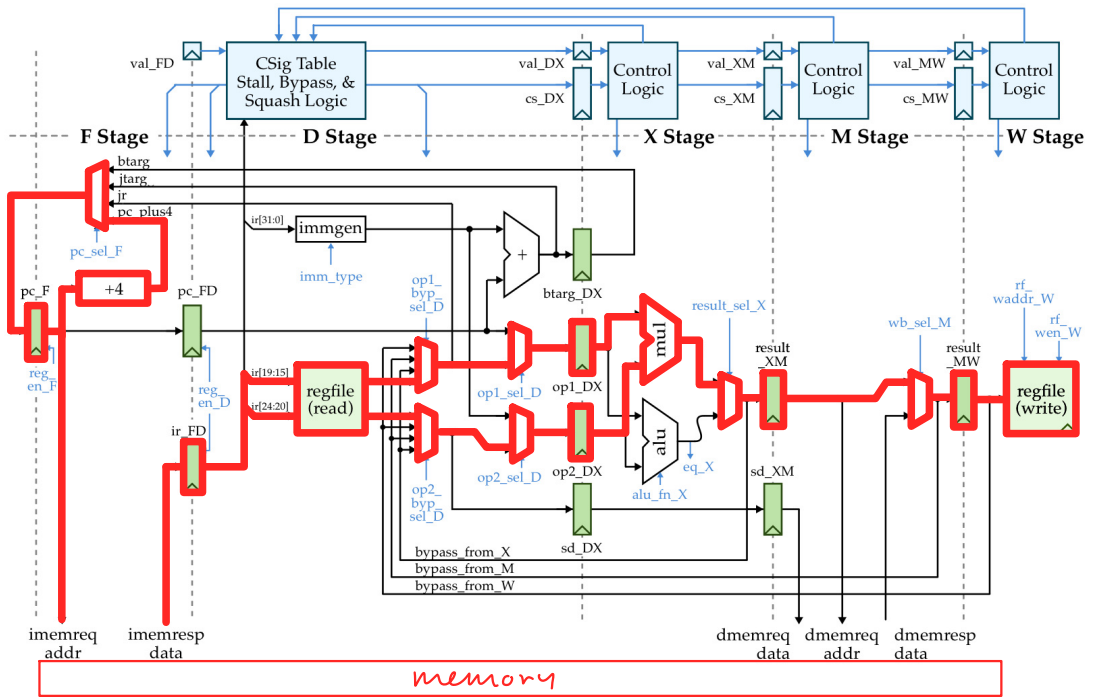
# ADD



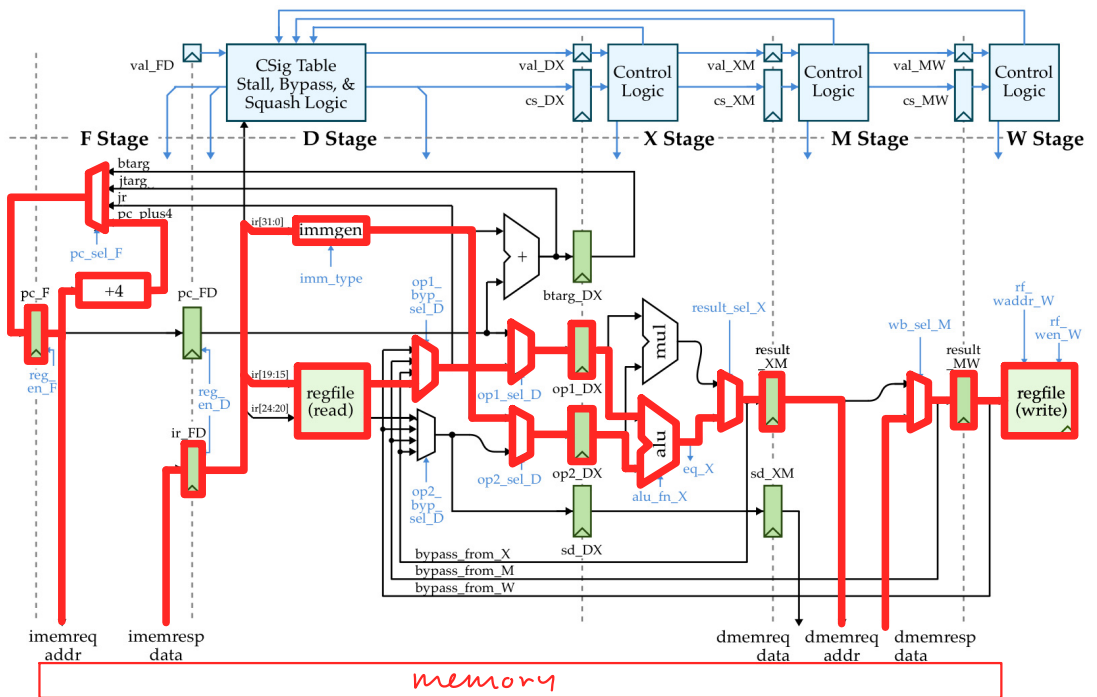
ADDI



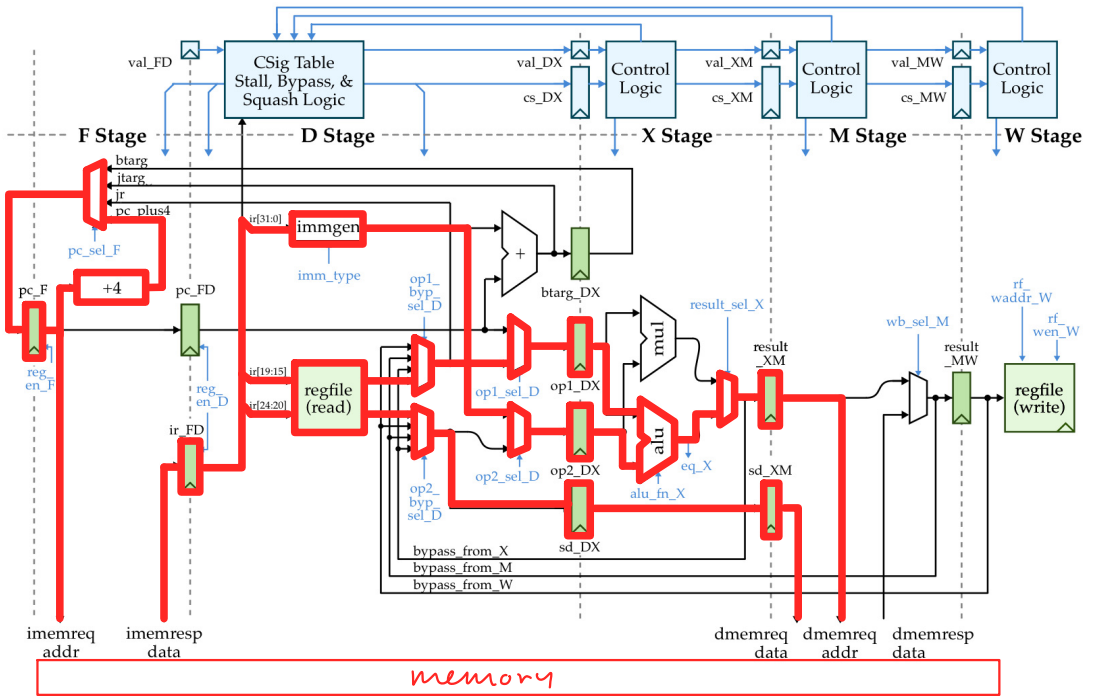
# MVL



# LW

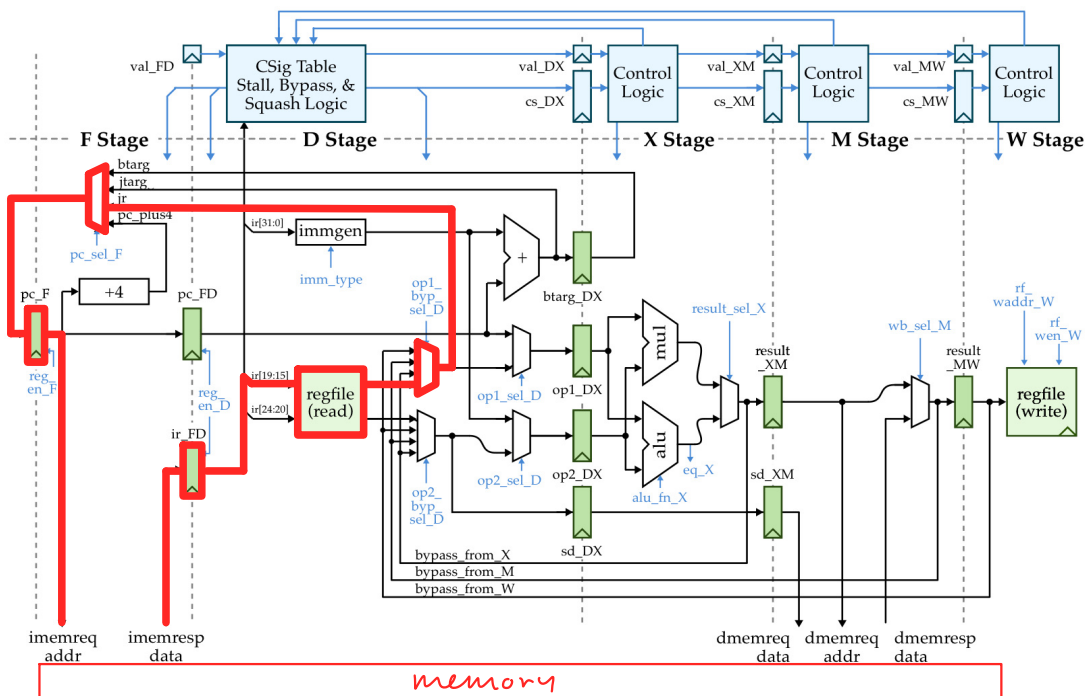


# SW



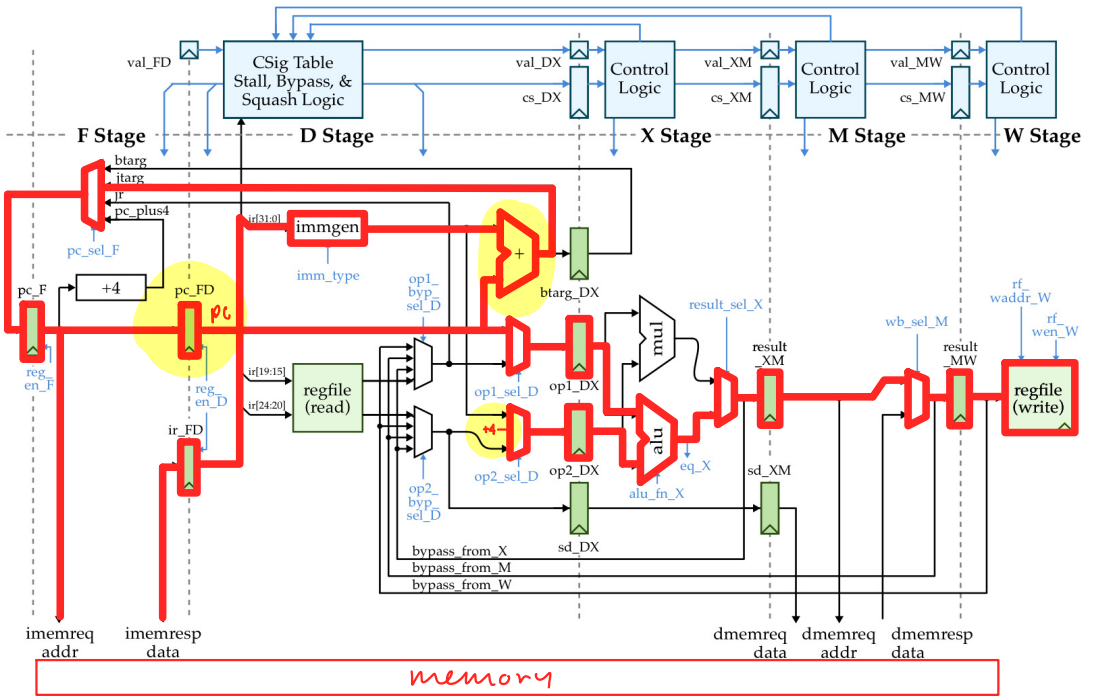
## SW r52 imm(r41)

JR



jr rs1

# JAL



jal rd addr

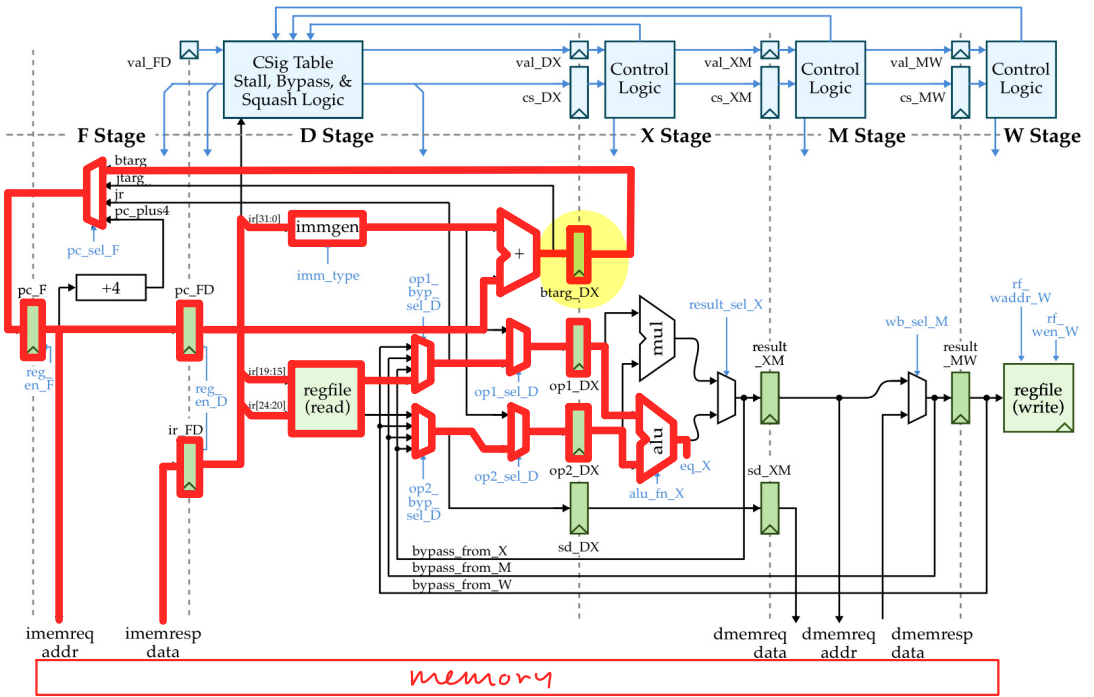
$R[rd] \leftarrow PC + 4$

$PC \leftarrow \text{addr}$

$PC + \text{sext}(imm)$



# BNE



bne rs1 rs2 addr

if  $R[rs1] == R[rs2]$

$PC \leftarrow PC + 4$

if  $R[rs1] != R[rs2]$

$PC \leftarrow \text{addr}$

$PC + \text{sext}(\text{imm})$

# CSRR/W

