Homework 9

Pipeline

1. please write down the HCL code for the following signals in PIPE implementation. NOTE: You should refer to Chapter 4.5 of CSAPP book.

   a. d_valB

   ```
   word d_valB = [
           d_srcB == e_dstE : e_valE;
           d_srcB == M_dstM : m_valM;
           d_srcB == M_dstE : M_valE;
           d_srcB == W_dstM : W_valM;
           d_srcB == W_dstE: W_ValE;
           1 : d_rvalB;
   ]
   ```

   b. D_stall

   ```
   bool D_stall =
   E_icode in { IMRMOVQ, IPOPQ } &&
   E_dstM in { dsrcA, dsrcB };
   ```

   c. E_bubble

   ```
   bool E_bubble =
       (E_icode == IJXX && !e_Cnd) ||
       E_icode in { IMRMOVQ, IPOPQ } && E_dstM in { d_srcA, d_srcB };
   ```

2. What's the difference between signal e_dstE and E_dstE? When are they updated?

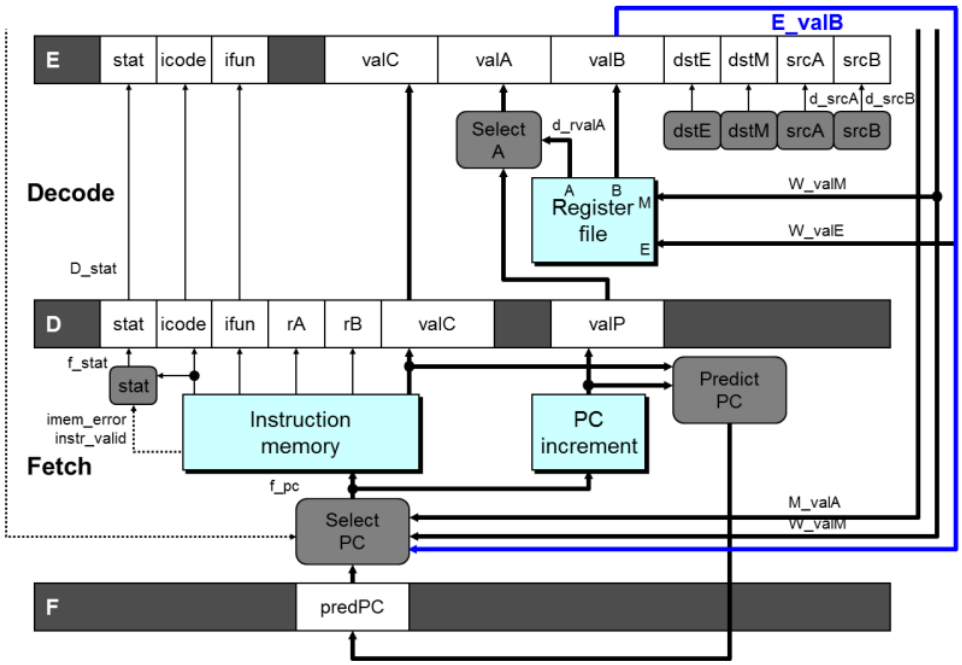
e_dstE is the signal generated and updated in Execute stage, while E_dstE is in the Pipeline Register updated during a rising clock.

3. Suppose we add a new instruction rjmp rB to Y86 instruction sets. It will jmp to the address stored in rB.

a. Fill in the function of each stage for rjmp rB instruction in Y86 sequential implementation. (NOTE: use valB to update PC)

Stage	rjmp rB
F	icode:ifun ← M1[PC] rA:rB ← M1[PC+1] valP ← PC + 2
D	valB ← R[rB]
E	--
M	--
W	--
P	PC ← valB

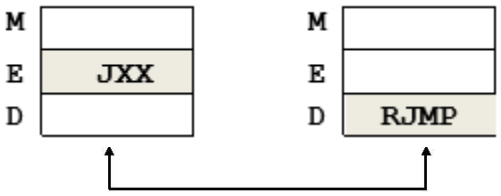
b. As shown in the new PIPE logic figure, we add a forwarding logic from E_valB to f_pc to support rjmp instruction, since the target address require read from register file. Please describe all possible hazards due to new instruction rjmp. You need provide detail explanation and list detection conditions like Figure 4.64 and control action like Figure 4.66. (Do not consider the hazard combinations here.)



Condition	Trigger	F	D	E	M	W
Target-addr Hazard	IRJMP in {D_icode}	S	B	N	N	N

c. Please list all hazard combinations (arise simultaneously) including rjmp instruction. You need draw pipeline states figures like Figure 4.67 and list pipeline control action like tables after Figure 4.67 for each combination.

Condition	F	D	E	M	W
Load	Stall	Stall	Bubble	Normal	Normal
RJMP	Stall	Bubble	Normal	Normal	Normal
Combination	Stall	B+S	Bubble	Normal	Normal
Desired	Stall	Stall	Bubble	Normal	Normal



Condition	F	D	E	M	W
JXX	Normal	Bubble	Bubble	Normal	Normal
RJMP	Stall	Bubble	Normal	Normal	Normal
Combination	SorBorN	Bubble	Bubble	Normal	Normal

d. The original PIPE implementation of Y86 should be modified to support the rjmp instruction. Please describe the modification and provide HCL of f_pc and D_bubble logic. (NOTE: Only need to write the code about rjmp instruction)

```
word f_pc = [

        E_icode == IRJMP : E_valB

    ];

bool D_bubble = [

    D_icode == IRJMP && !{E_icode in { IMRMOVQ, IPOPQ } && E_dstM in { d_srcA, d_srcB}}

]
```