

Hazard Unit

```

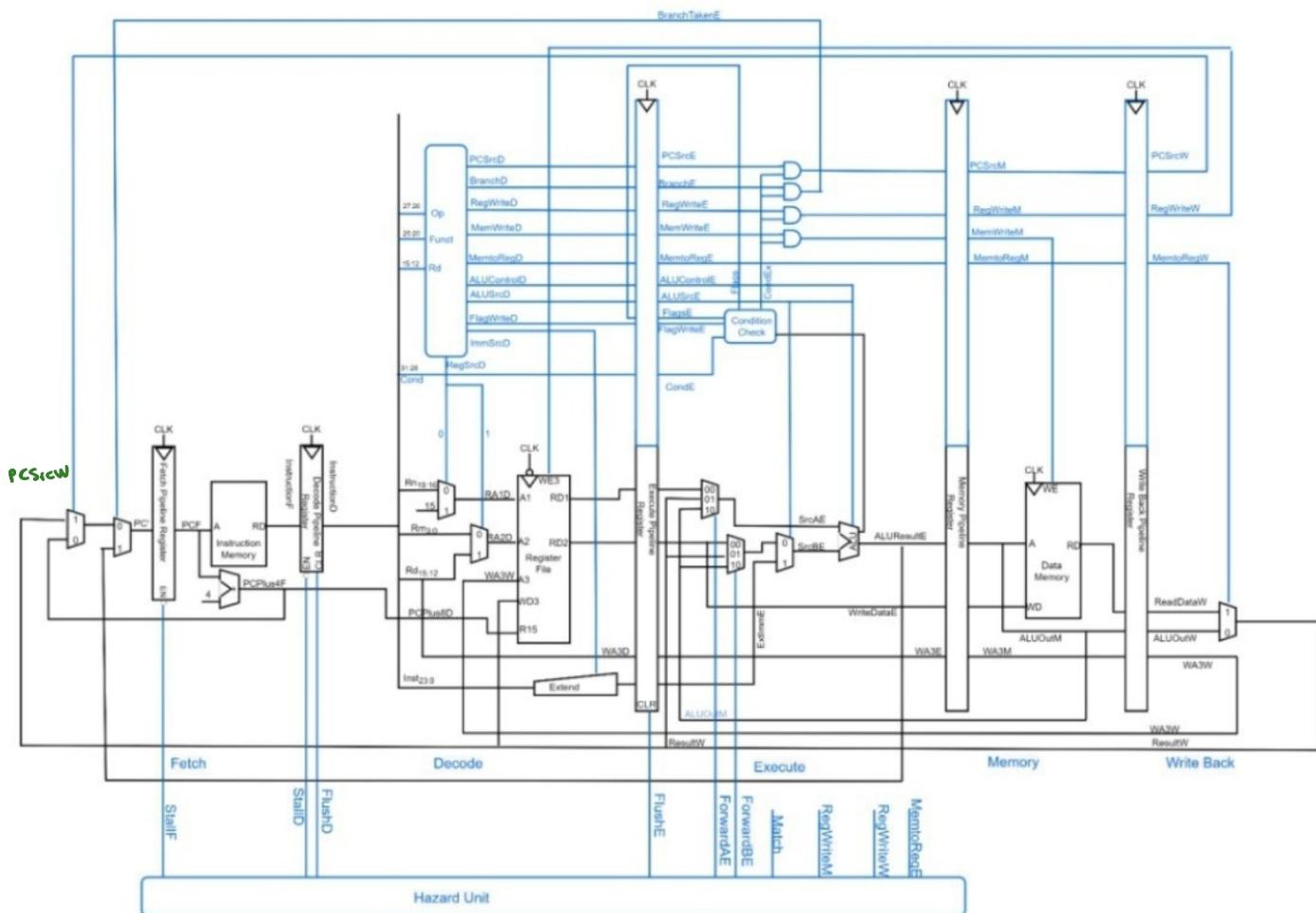
Match_1E_M = (RA1E == WA3M)
Match_1E_W = (RA1E == WA3W)
if (Match_1E_M • RegWriteM)
ForwardAE = 10; // SrcAE = ALUOutM
else if (Match_1E_W • RegWriteW)
ForwardAE = 01; // SrcAE = ResultW
else ForwardAE = 00; // SrcAE from regfile

Match_2E_M = (RA2E == WA3M)
Match_2E_W = (RA2E == WA3W)
if (Match_2E_M • RegWriteM)
ForwardBE = 10; // SrcBE = ALUOutM
else if (Match_2E_W • RegWriteW)
ForwardBE = 01; // SrcBE = ResultW
else ForwardBE = 00; // SrcBE from regfile

```

\rightarrow
Match_12D_E = (RA1D == WA3E) + (RA2D == WA3E)
LDRstall = Match_12D_E • MemtoRegE
BranchTakenE = BranchE • CondEx

PCWrPendingF = PCSrcD + PCSrcE + PCSrcM; Fetch is stalled, Decode is Flushed
StallF = LDRstall + PCWrPendingF; Not asserted during PCSrcW to allow the write
StallID = LDRstall
FlushD = PCWrPendingF + PCSrcW + BranchTakenE; Asserted as long as PC Write is going on or Branch is taken
FlushE = LDRstall + BranchTakenE;



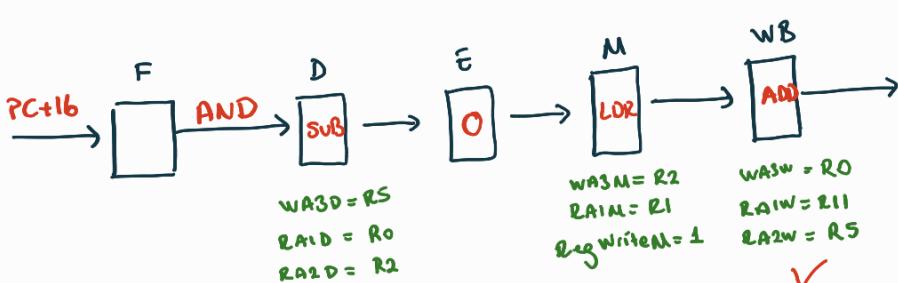
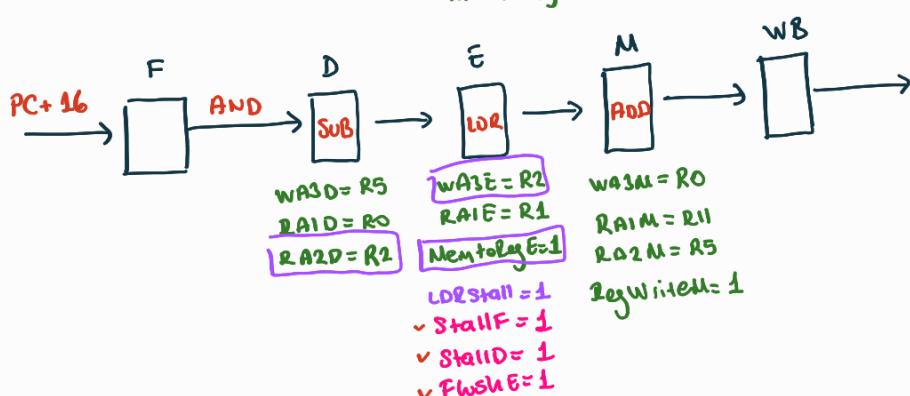
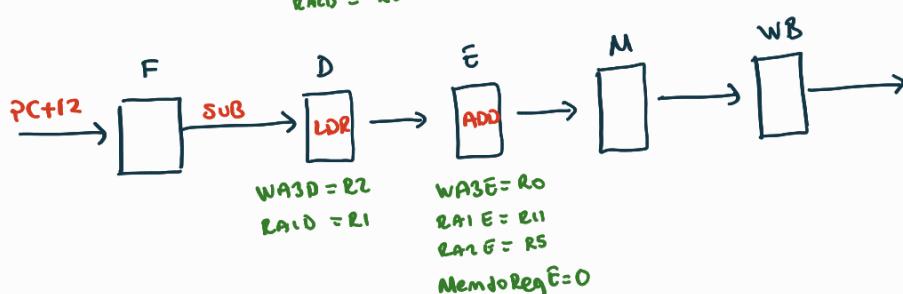
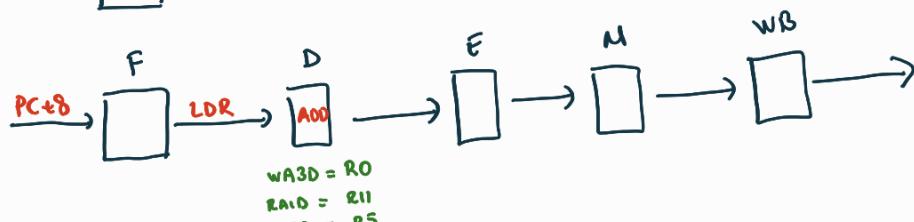
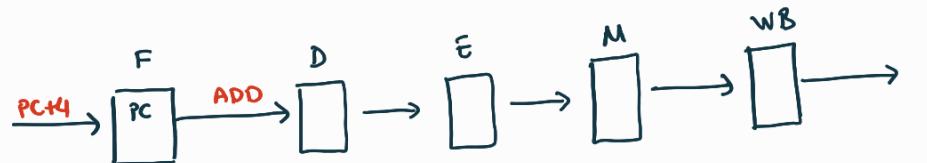
Question 1

ADD R0, R1, R5

LDR R2, [R1, #45] → Data Mem (LDR)

SUB R5, R0, R2 → Data Mem (RAW)

AND R5, R2, R5



$$WA3E == RA2D \rightarrow \text{Match-12-D-E} = 1$$

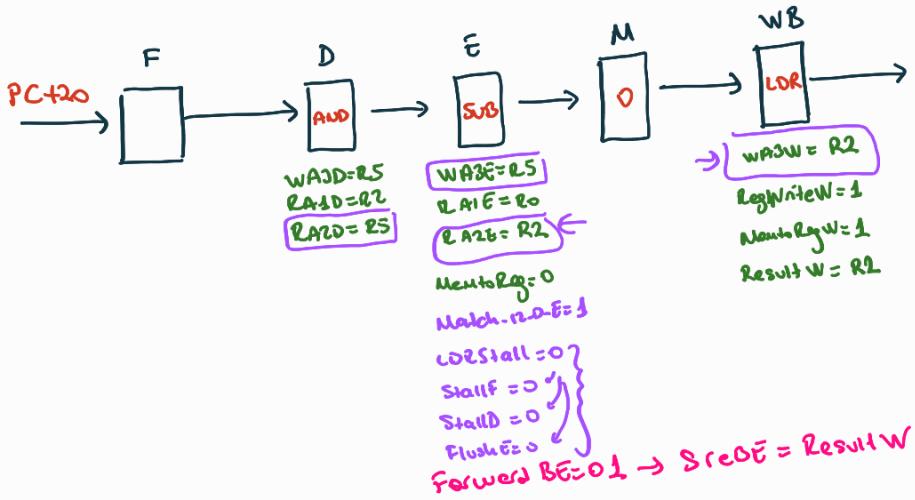
$$\underline{\text{LDR stall}} = (\underline{\text{Match-12-D-E}}) \cdot (\underline{\text{Mem to Reg E}})$$

Stall F = 1

Stall D = 1

Flush E = 1

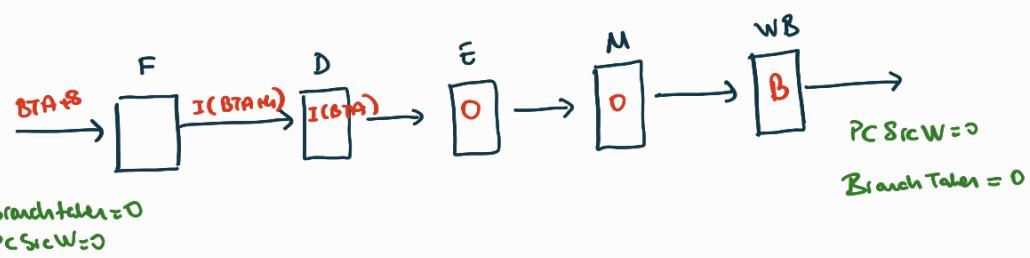
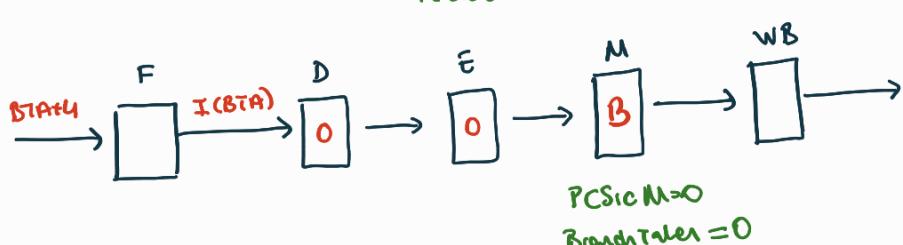
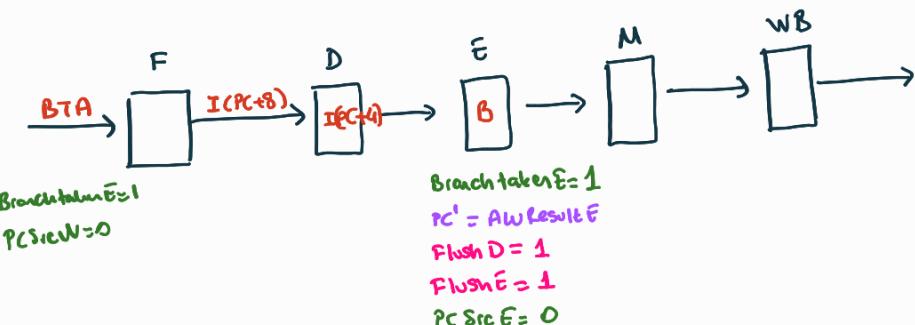
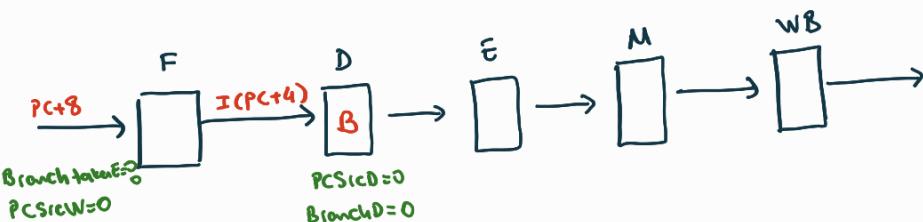
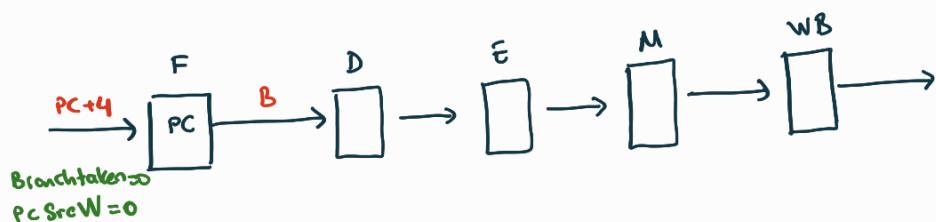
CPI: 1 → 2



Match-12.D.E = (RA.D == WA3.E) = 1
 (RA.E == WA3.W)
 \downarrow
 Match-2E.W = 1 } forward BE = 01
 RegWrittenW = 1 } = 0

Question 2

Show unconditional branch

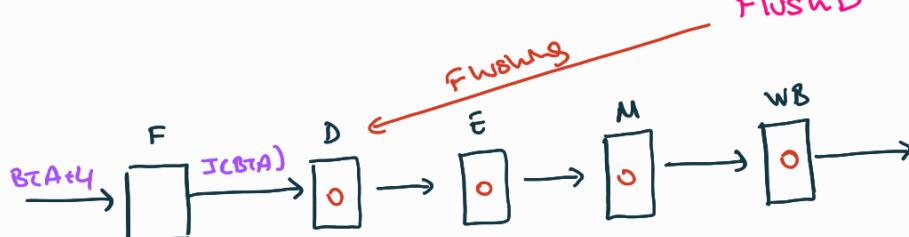
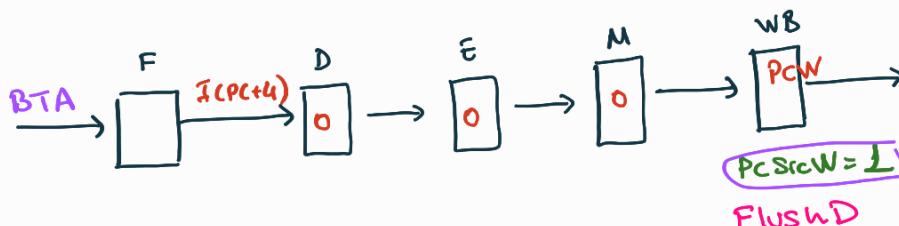
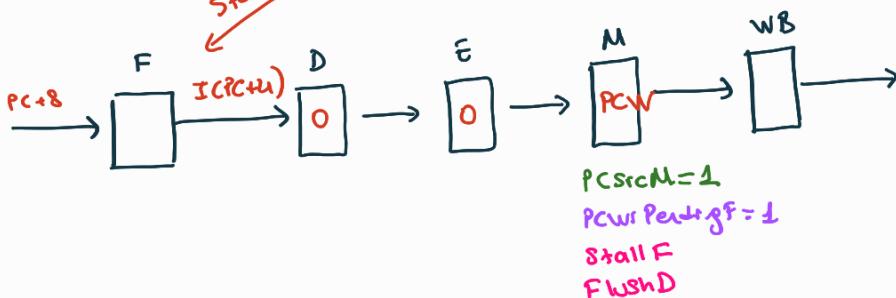
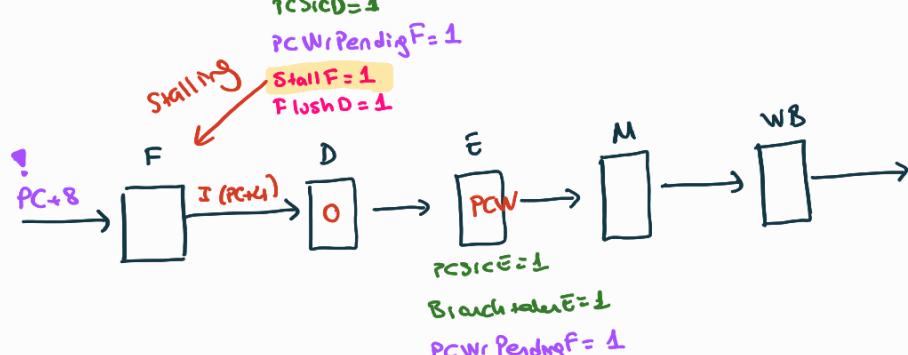
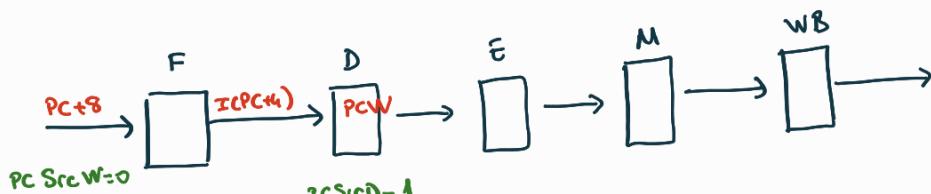
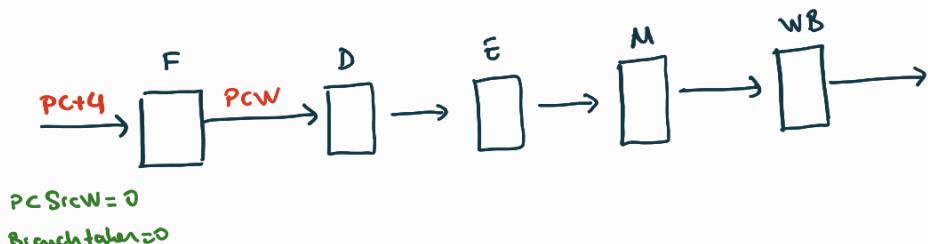


CPI : 1 \rightarrow 3

Question 3

Execute a PCWrite Branch

($PCSrc = (RD == IS) \& RegW$)



CPI 1 $\xrightarrow{+4} 5$