8.3 An Acc-ISA CPU executes the following instructions using 3-bit op-codes and 5-bit address or 2's complement data. Do the following:

LD address	//Acc ← Memory [address], read from LM2
LD data	//Acc ← data (a 2's complement number, sign extended)
ADD data	//Acc ← Acc + data (data is a 2's complement number, sign extended)
SUB data	//Acc ← Acc - data (data is a 2's complement number, sign extended)
ADD (address)	//Acc ← Acc + Memory[address]
STM (address)	//M[address] ← Acc
SUB (address)	//Acc ← Acc - Memory[address]
JMP address	//PP ← address
JZ address	

a) Draw a data path for the CPU assuming the DM has separate input and output bus as in the data path shown in Fig. 8.7. Do not include additional data paths not used by the instructions. **(15 pts)**

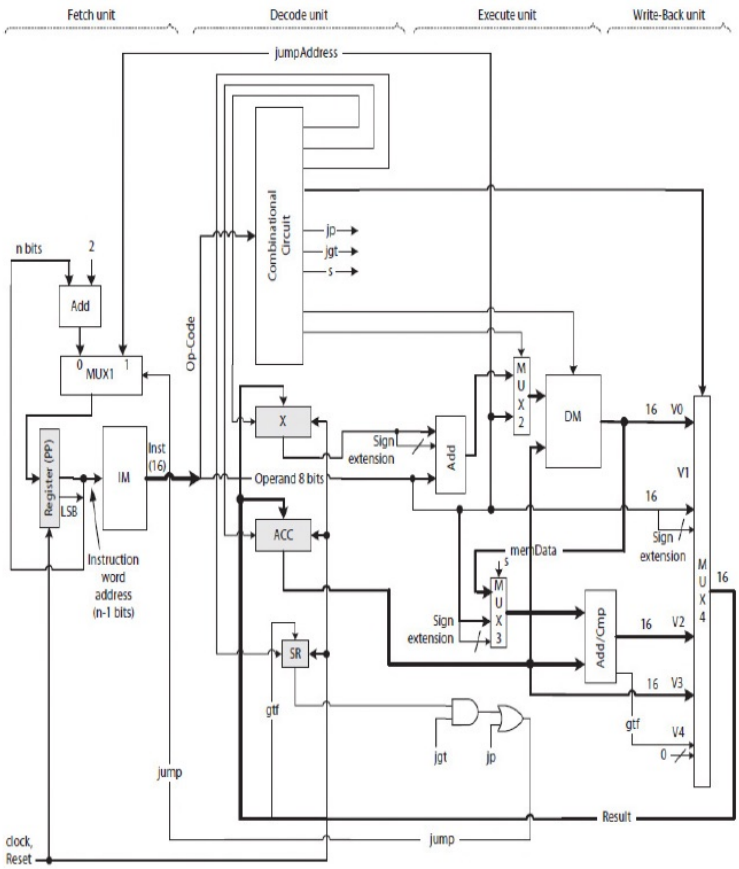
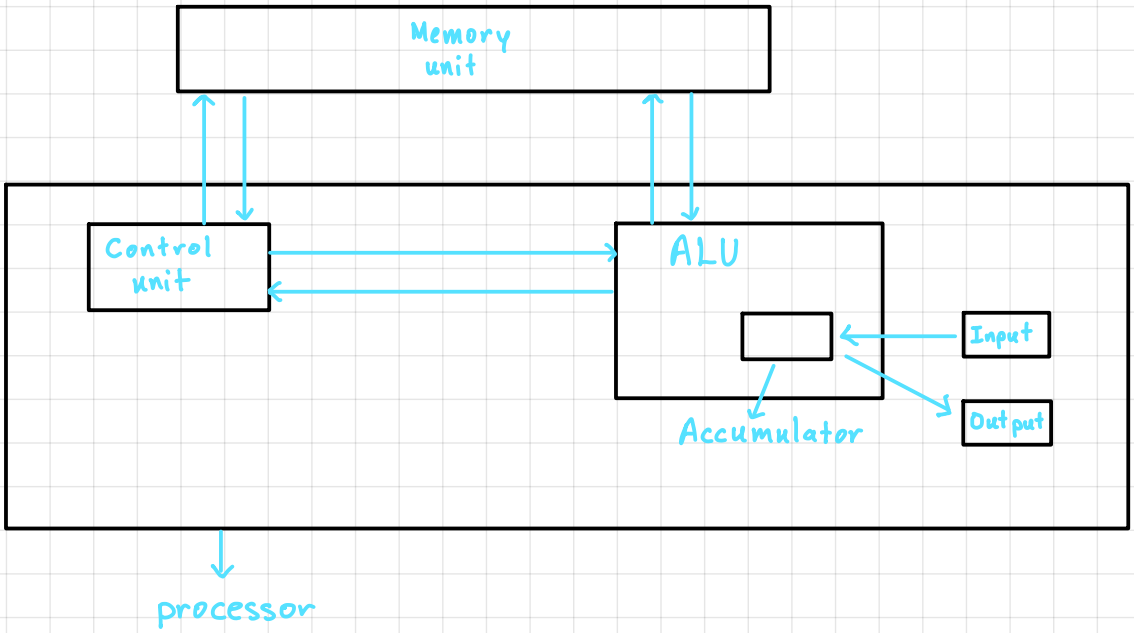
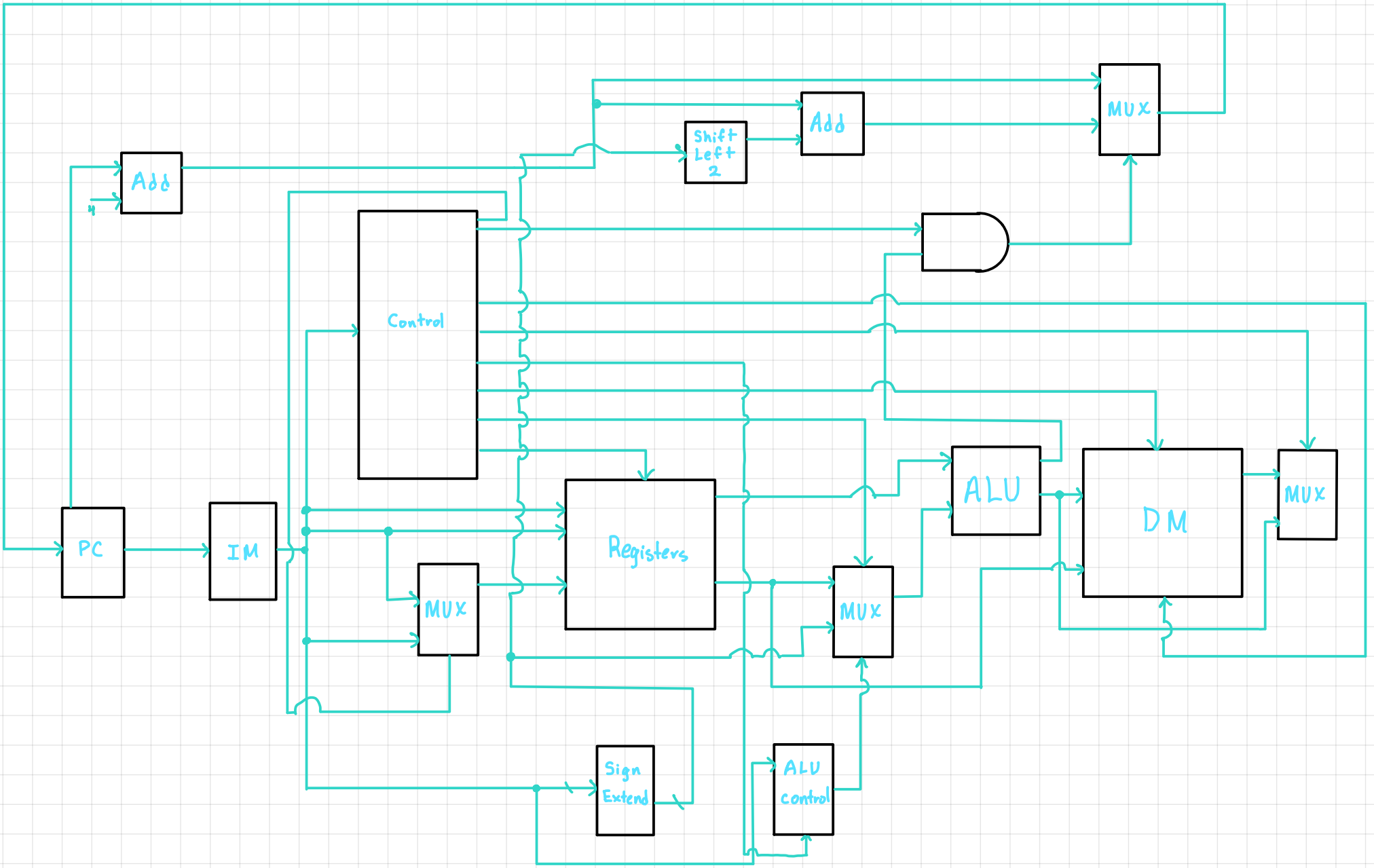


FIGURE 8.7 The Acc-ISA single-cycle data path to execute the program in Example 8.2.




Add
4
Shift Left 2
Add
MUX
Control
PC
IM
MUX
Registers
MUX
ALU
DM
MUX
Sign Extend
ALU control

Problem I. Computation is performed by a RISC ISA. A = B * (C + D). What is the value in R4 after the execution of code line # 6: (B = 5; C = 10; D = 15) ie: Code line # 6 has been completed. (5 pts)

R4 = _____

RISC-ISA: Example of assembly program

```
1. LD    R1,   (C)
2. LD    R2,   (D)
3. ADD   R3,   R1, R2
4. LD    R4,   (B)
5. MUL   R5,   R3, R4
6. ST    (A),  R5
```

1. LD R1, (C) // load c=10 into R_1
2. LD R2, (D) // load d=15 into R_2
3. ADD R3, R1, R2 // $R_1 + R_2 = 10 + 15 = 25$; store result in R_3
4. LD R4, (B) // load b=5 into R_4
5. MUL R5, R3, R4 // R3 x R4 = 25 x 5 = 125; store result in R5
6. ST (A), R5 // Store R5 = 125 into A

∴ The value of R4 after the execution of code line #6 is R4 = 5