8085 Programming

4.1. PROGRAMMING STEPS

In programming technique five different points have been given :

(1) Define the problem to be solved.

The problem for which you are preparing the program, the different terms must be clearly mentioned such as :

(a) What are the input to your program ?

In the case of 8085, 3 possible conditions arises :

(i) The direct data is available.

(ii) The data is stored in general purposes register.

(iii) The data is stored in memory location.

(2) What is the operation you are expecting.

(3) What you want the output?

Solution Plan : The plan to solve the problem should be prepared it includes the following points :

(a) How are you taking input data.

(b) Which method you are using to solve the problem ? What are the steps in this method ?

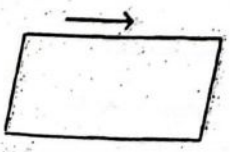
(c) How will you output the result ?

(4) **Flow chart :** Prepare the flow chart of the plan you have decided which will give the exact idea of how the program should flow to get collect result.

(5) **Program :** Go on putting the instructions instead of flow chart blocks.

(6) **Check the result :** Now find the codes for instructions feed in 8085 system and execute the program. It will give you the result.

Symbols commonly used in flow charting :

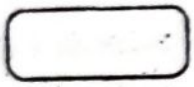


Arrow : To indicate the direction of flow.

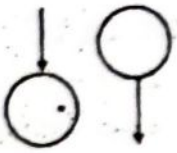
Parallelogram : Input/output unit.

Rectangle : Indicates process operation.

Diamond unit : Decision unit.



Oval : Start and step unit.



Circle with arrow : Represents continuation.

Drill Questions

Q. 1. Write suitable instructions to perform the following tasks :

(a) To examine the contents of stack position.

```
LXI H, 0000 H
DAD SP.
```

(b) To examine the contents of program counter (PC)

Main program	Subroutine
Location	C300H XTHL
2000 H call C300H	C301H PCHL

(c) To examine the contents of flag register.

```
LXISP, FFFFH
PUSH PSW
POP D
```

(d) To clear all flags.

```
LXI, SP, COFFH
MOV D, A
MVI E, 00H
PUSH D
POP PSW.
```

Q. 2. List out the instructions that clears the accumulator :

(a) MVI A, 00H

(b) XRA A

(c) SUB A

(d) ANI, 00H

Q. 3. The accumulator of an 8085 contains C5H and carry flag is set. What will be the accumulator and carry contain followng each of the instruction given below ?

(1) XRA A (2) ANA A (3) ADI 94H

(i) XRA A $\Rightarrow$ A = 0, Cy = 0

(ii) ANA A $\Rightarrow$ A = C5 Cy = 0

(iii) ADI 94H A = 59 Cy = 1

Q. 4. Write set of instructions to load program counter with 2nd and 3rd byte of instruction.

LXIH, YY XX
PCHL
where XX : 2nd byte of instruction.
YY : 3rd byte of insruction.

Q. 5. Write a set of instructions to have no change in normal execution except incrementing the program counter.

⇒ Using NOP instruction.

4.2. SIMPLE PROBLEMS

Example 1. Store 8-bit data in memory.

Statement : Stored the data byte 52H into memory location 2000H.

Program 1.

MVI A, 52H → Store 52H in the accumuloator
STA 2000H → Copy accumulator contents at address 2000H
HLT → Terminate program execution.

Program 2.

LXI H, 2000H → Load HL with 2000H
MVI M, 52H → Store 52H in memory location pointed by HL register pair (2000H)
HLT → Terminate program execution

The result of both programs will be the same.

Note : In program 1 direct addressing instruction is used, whereas in program 2 indirect addressing instruction is used.

Example 2. Exchange the contents of memory locations.

Statement : Exchange the contents of memory locations 1000H and 2000H.

Program 1. LDA 1000H → Get the contents of memory location 1000H into accumulator
MOV B, A → Save the contents in B register
LDA 2000H → Get the contents of memory location 2000H ionto accumulator.
STA 1000H → Store the contents of accumulator at address 1000H.
MOV A, B → Get the saved contents back into A register
STA 2000H → Store the contents of accumulator at address 2000H
HLT → Terminate program execution.

Program 2.

LXI H, 1000H → Initialize HL register pair as a pointer to memory location 1000H
LXI D, 2000H →Initialize DE register pair as a pointer to memory location 2000H

MOV B, M → Get the contents of memory location 1000H into B register
LDAX D → Get the contents of memory location 2000H into A register
MOV, M, A → Store the contents of A register into memory location 1000H
MOV A, B → Copy the contents of B register into accumulator
STAX D → Store the contents of A register into memory location 2000H
HLD → Terminate program execution.

Note : In program 1 direct addressing instructions are used, whereas in program 2 indirect addressing instructions are used.

Example 3. Add two 8 bit numbers.

Statement : Add the contents of memory locations 2000H and 2001H and place the result in memory location 2002H.

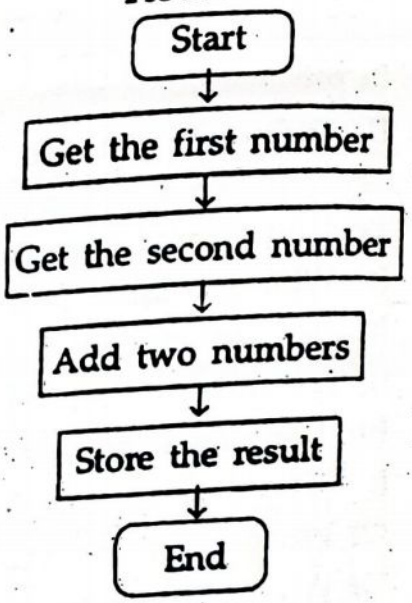
Sample Problem

(2000H) = 14H
(2001H) = 89H
Result = 14H + 89 = 9DH

Program

LXI H, 2000H → HL points 2000H
MOV A, M → Get first operand
INX H → HL points 2001H
ADD M → ADD second operand
INX H → HL points 2002H
MOV, M, A → Store result at 2002H
HLT → Terminate program execution.

Flow/chart



Example 4. Subtract two 8 bit numbers.

Statement : Subtract the contents of memory location 2001H from the memory location 2000H and place the result in memory location 2002H.

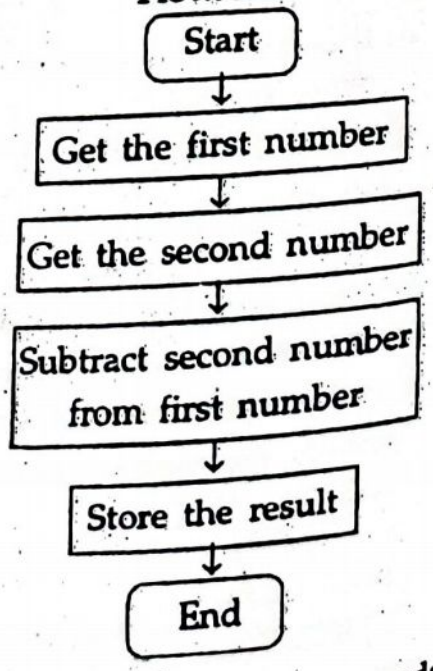
Sample Problem

(2000H) = 51H
(2001H) = 19H
Result = 51H – 19H = 38H

Program

LXI H, 2000H → HL points 2000H
MOV A, M → Get first operand
INX H → HL points 2001H
SUB M → Subtract second operand
INX H → HL points 2002H
MOV M, A→ Store result at 2002H
HLT → Terminate program execution.

Flow/chart



Note. **Always write program according to the format given in Example 13 onwards.**

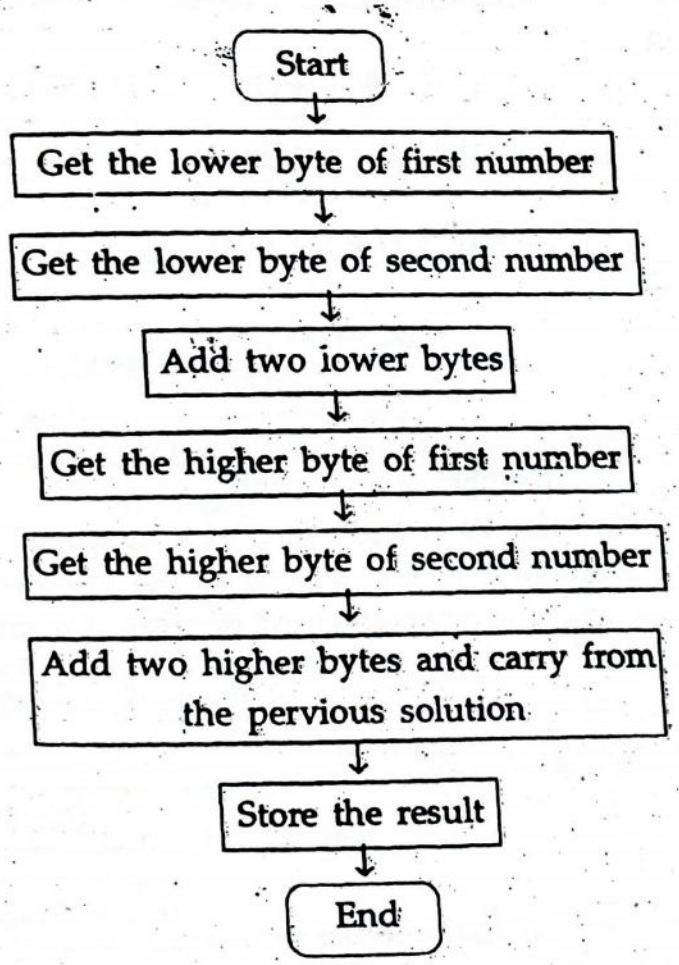
Example 5. Add two 16 bit numbers.

Statement : Add the 16 bit number in memory locations 2000H and 2001H to the 16-bit number in memory locations 2002H and 2003H. The most significant eight bits of the two numbers to be added are in memory locations 2001H and 2003H. Store the result in memory locations 2004H and 2005H with the most significant byte in memory location 2005H.

Sample Problem

(2000H) = 15H
(2001H) = 1CH
(2002H) = B7H
(2003H) = 5AH
Result = 1C15 + 5AB7H = 76CCH
(2004H) = CCH
(2005H) = 76H

Flow/chart



Program 1.

LHLD 2000H → Get first 16 bit number in HL
XCHG → Save first 16 bit number in DE
LHLD 2002H → Get second 16 bit number in HL
MOV A, E → Get over byte of the first number
ADD L → Add lower byte of the second number

MOV L, A → Store result in L register
MOV A, D → Get higher byte of the first number
ADC H → Add higher byute of the second number with carry
MOV H, A → Store result in H register
SHLD 2004H → Store 16 bit result in memory locations 2004H and 2005H.
HLT → Terminate program execution.

Source Program 2

LHLD 2000H → Get first 16 bit number
XCHG → Save first 16 bit number in DE
LHLD 2002H → Gate second 16 bit number in HL
DAD D → Add DE and HL
SHLD 2004H → Store 16 bit result in memory locations 2004H and 2005H
HLT → Terminate program execution.

In program 1, eight bit addition instructions are used (ADD and ADC) and addition is performed in two steps. First lower byte addition using ADD instruction and then higher byte addition using ADC instruction. In program 2, 16 bit addition instruction (DAD) is used.

Example 6. Subtract two 16 bit numbers.

Statement : Subtract the 16 bit number in memory locations 2002H and 2003H from the 16-bit number in memory locations 2000H and 2001H. The most significant eight bits of the two numbers are in memory locations 2001H and 2003H. Store the result in memory location 2004H and 2005H with the most significant byte in memory location 2005H.

Sample Problem

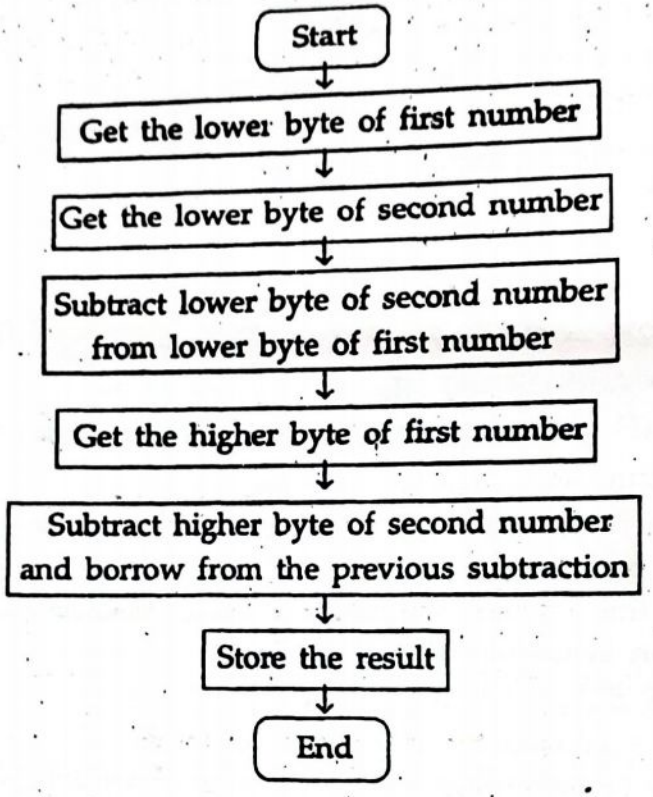
(2000H) = 19H
(2001H) = 6AH
(2002H) = 15H
(2003H) = 5CH
Result = 6A 19H – 5C 15H = OE04H
(2004H) = 04H
(2005H) = OEH

Program

LHLD 2000H → Get first 16 bit number in HL
XCHG → Save first 16 bit number in DE
LHLD 2002H → Get second 16 bit number in HL
MOV A, E → Get lower byte of the first number
SUB L → Subtract lower byte of the second number
MOV L, A → Store the result in L register
MOV A, D → Get higher byte of first number
SBB H → Subtract higher byte of second number with borrow
MOV H, A → Store 16 bit result

SHLD 2004 H → Store 16 bit result in memory locations 2004H and 2005H
HLT → Terminate program execution.

Flow-chart



Example 7. Check result after execution of INR B, INR C and INX B instructions.

Statement : If the contents of B = FFH and C = FFH then after execution of following instructions give the contents of register B and register C.

Instructions :

1. INR B
2. INR C
3. INX B

1. INR B

$$\begin{array}{r} B \rightarrow FFH \\ +\ 01H \\ \hline 00H \rightarrow B \end{array}$$

$\therefore$ B = 00H and C = FFH

2. INR C

$$\begin{array}{r} C \rightarrow FFH \\ +\ 01H \\ \hline 00H \rightarrow C \end{array}$$

$\therefore$ B = FFH and C = 00H

3. INX B

$$\begin{array}{rcccl} BC \rightarrow & FF & FF & H & \\ + & 00 & 01 & H & \\ \hline & 00 & 00 & H & \rightarrow BC \end{array}$$

$\therefore$ B = 00H and C = 00H.

Example 8. Check results after execution of DCR C, DCR B and DCX B instructions.

Statement : If the contents of B = 00H and C = 00H then after execution of following instructions give the contents of register B and register C.

Instructions :

1. DCR C
2. DCR B
3. DCX B

1. DCR C

$$\begin{array}{rl} C \rightarrow & 00H \\ - & 01H \\ \hline & FFH \rightarrow C \end{array}$$

$\therefore$ B = 00H and C = FFH

2. DCR B

$$\begin{array}{rl} B \rightarrow & 00H \\ - & 01H \\ \hline & FFH \rightarrow B \end{array}$$

$\therefore$ B = FFH and C = 00H

3. DCX B

$$\begin{array}{rl} BC \rightarrow & 0000H \\ - & 0001H \\ \hline & FFFFH \rightarrow BC \end{array}$$

$\therefore$ B = FFH and C = FFH

Example 9. Find the 1's complement of a number.

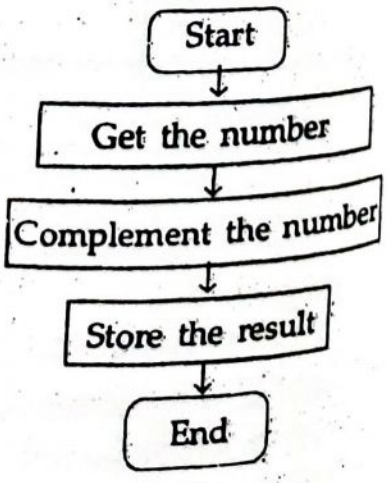
Statement : Find the 1's complement of the number stored at memory location 2200H and store the complemented number at memory location 2300H.

Sample Problem

(2200H) = 55H

Result = (2300H) = A AH

Flow-chart



Program

LDA 2200H → Get the number
CMA → Complement number
STA 2300H → Store the result
HLT → Terminate program execution.

Example 10. Find the 2's complement of a number.

Statement : Find the 2's complement of the number stored at memory location 2200H and store the complemented number at memory location 2300H.

Sample Problem

(2200H) = 55H

(Result) = (2300H) = A AH + 1 = ABH

Program

LDA 2200 H → Get the number

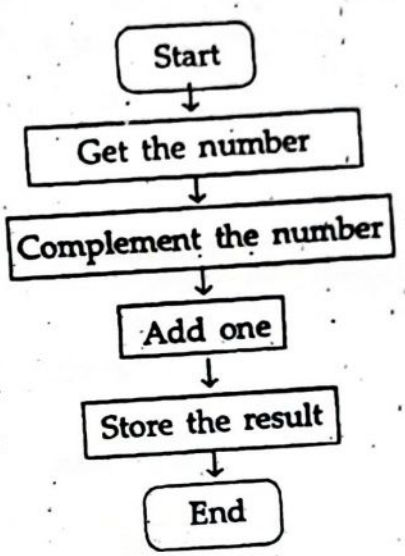
CMA → Complement the number

ADI, 01H → Add one in the number

STA 2300A → Stored the result

HLT → Terminate program execution.

Flow-chart



Example 11. Pack the two unpacked BCD numbers.

Statement : Pack the two unpacked BCD numbers stored in memory locations 2200H and 2201H and store result in memory location 2300H. Assume the least significant digit is stored at 2200H.

Sample Problem

(2200H) = 04

(2201H) = 09

Result = (2300H) = 94

Source Program

LDA 2201 H → Get the most sifnificant BCD digit

RLC

RLC

RLC

RLC → Adjust the position

ANI F0H → Make least significant BCD digit zero

MOV C, A → Store the partial result

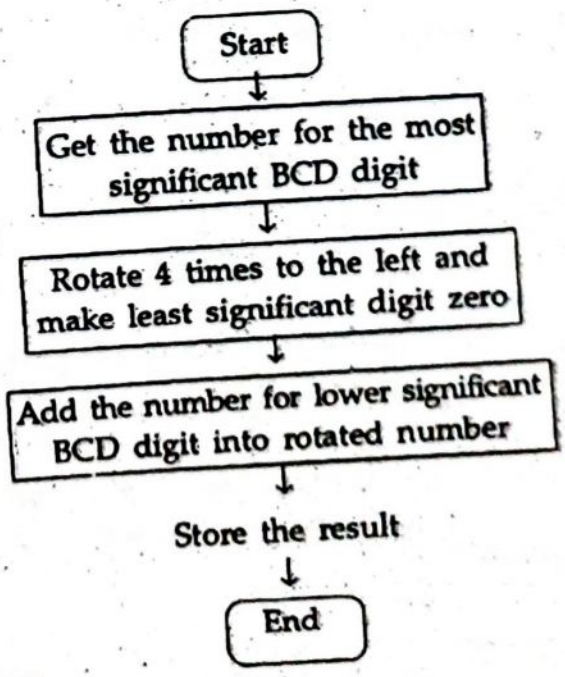
LDA 2200H → Get the lower BCD digit

ADD C → Add lower BCD digit

STA 2300H → Store the result

HLT → Terminate program execution.

Flow-chart



Example 12. Unpack the BCD number.

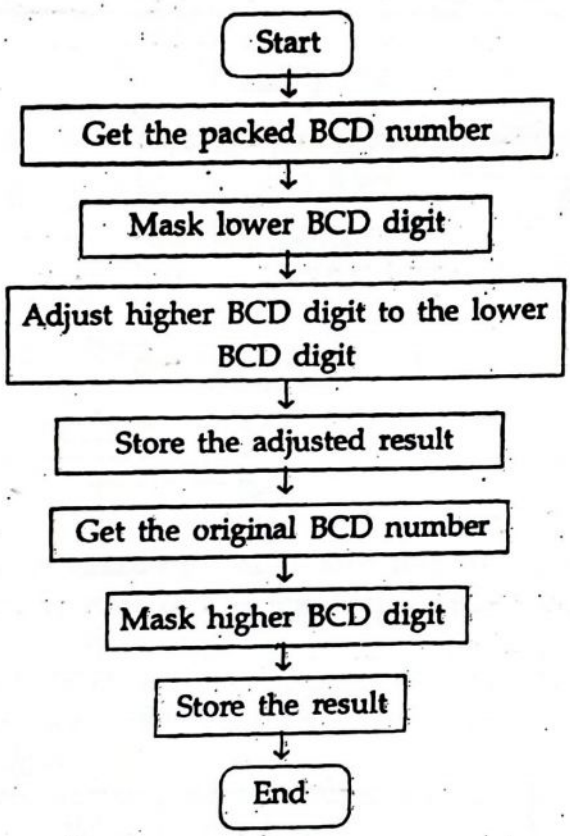
Statement : Two digit BCD number is stored in memory location 2200H. Unpack the BCD number and store the two digit in memory locations 2300H and 2301 such that memory location 2300H will have lower BCD digit.

Sample Problem

(2200H) = 58

Result = (2300H) = 08 and (2301H) = 05

Flow-chart



Source Program

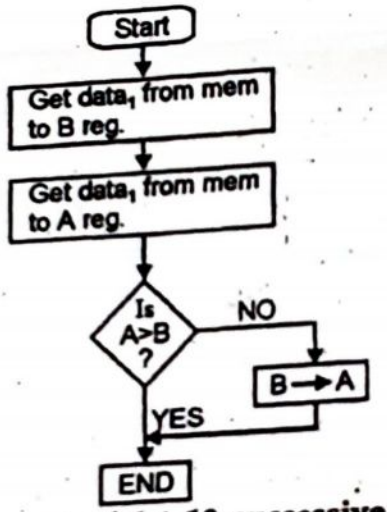
LDA 2200H → Get the packed BCD number
ANI FOH → Mask lower nibble
RRC
RRC
RRC
RRC → Adjust higher BCD digit as a lower digit
STA 2301H → Store the partial result
LDA 2200H → Get the original BCD number
ANI 0F H → Mask higher nibble
STA 2201H → Store the result
HLT → Terminate program execution.

Example 13. Find a greater number out of given 2 numbers.

Statement : Write a program to find a greater number out of given 2 numbers. The two numbers are stored at location 2000H and 2001H. Store the result in accumulator.

Memory location	Label	Mnemonics	Remarks
C000H		LDA 2000H	(2000H) → A
C003H		MOV B, A	A → B
C004H		LDA 2001H	(2001H) → A
C007H		CMP B	A compare B if A < BCy = 1 A > B Cy = 0
C008H		JNC : MUSK	If Cy = 0, go to MUSK
C00BH		MOV A, B	If Cy = 1, B → A
C00CH	MUSK	HLT	Stop

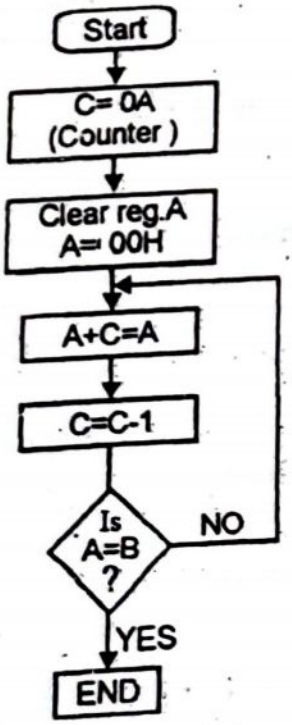
Flow-chart



Example 14. To find sum of 1st 10 successive numbers from 1 to 10.

Statement : Write a program to find sum of 1st 10 successive numbers from 1 to 10.

Flow-chart



Program

Memory loction	Label	Memonics	Remarks
2000H		MVIC, 0AH	0A → C reg.
2002H	/	MVI A, 00H	00 → A reg.
2004H	XYZ	ADD C	A + AC → A
2005H		DCR C	C – 1 → C
2006H		JNZ XYZ	is C = 0
2009H		HLT	Stop

Example 15(a). Calculate the sum of series of numbers.

Statement : Calculate the sum of series of numbers. The length series is in memory location C200H and the series itself begins from m location C201H.

(a) Assume the sum to be 8 bit number so you can ignore carries. the sum of memory location C300H.

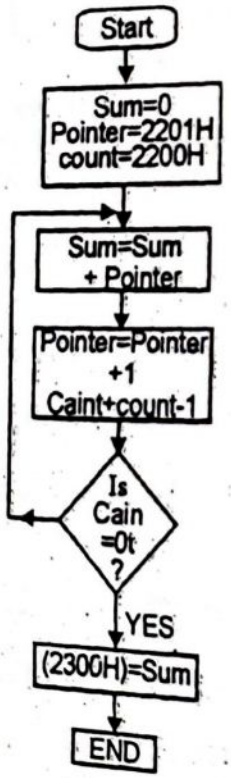
(b) Assume the sum to be 16 bit number. Store the sum at m locations C300H and C301H.

Let,

C200H = 04H
C201H = 20H
C202H = 15H
C203H = 13H
C204H = 22H

$\therefore$ Result = 20 + 15 + 13 + 22 = 6AH
C300H = 6AH

Flow-chart



Program

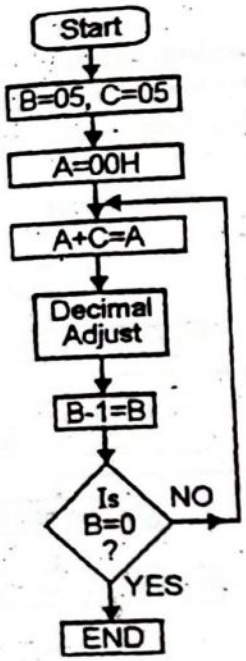
Memory	Label	Mnemonics	Remark
2000H		LDA C200A	C200 ⇒ A
2003H		MOV C, A	A ⇒ C
2004H		SUB A	A – A = A
2005H		LXIH, C201H	HL = C201H
2008H	XYZ	ADD M	A + M = A
2009H		INXH	M = M + 1
200AH		DCRC	C = C – 1
200BH		JNZ : XYZ	If C ≠ 0 repe
2000EH		STA C300H	A ⇒ C300A
2011H		HLT	Stop

Note : For every program draw table like above this is standard w for writing program (Now onwards programs are written without table.)

Example 15(b). To find sqaure of a given 8 bit number.

Statement : Write a program to find the square of a number. The numb is 05 the result should be in BCD format.

Flow-chart



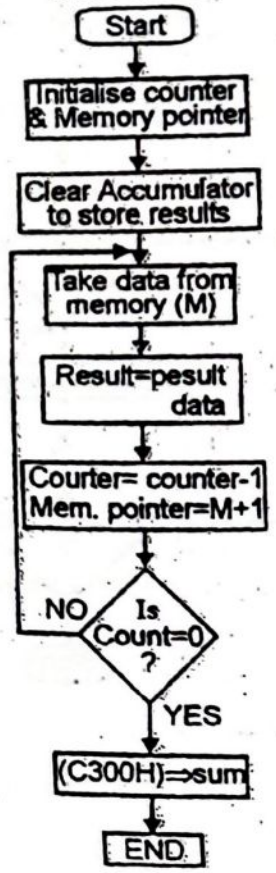
Program : $(5)^2 = 5 \times 5$
$= 5+5+5+5+5$

Label	Mnemonics	Remarks
	MVI A, 00	
	MVI B, 05	
	MVI C, 05	
XYZ	ADD C	
	DAA	
	DCR B	
	JNZ XYZ	
	HLT	

Example 16. Addition of 10 8 bit numbers.

Statement : Write a program to add ten data bytes. Data is stored in memory location stating from 5200H. The result is 8 bit only. Store the result at 5300H location.

Flow-chart



	LXI H, 5200H	
	MVI C, 0AH	
	MVI A, 00H	
XYZ	MOV B, M	
	ADD B	
	INX H	
	DCR C	
	JNZ XYZ	
	STA 5300H	
	HLT.	

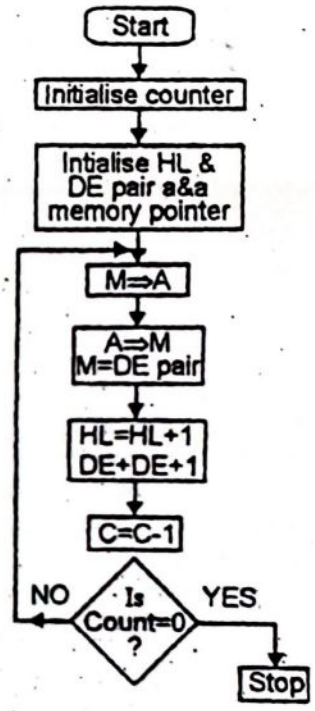
Example 17. Data transfer from block of memory M_1 to memory block M_2.

Statement : Write a program to transfor ten bytes of data from one memory to another memory block source memory block starts fro memory location C200H where as destination memory block starts from memory location C300H.

Program :

```
      MVI C, 0AH
      LXI H, C200H
      LXI D, C300H
XYZ : MOV A, M
      STAX D
      INX H
      INX D
      DCR C
      JNZ XYZ
      HLT.
```

Flow-chart



Example 18. Addition of 1st 10 even hex. numbers.

Statement : Write a program to add Ist 10 even hex. numbers store the result in D register.

Program : Ist 10 even hex. numbers are 0, 2, 4, 6, 8, A, C, E, 10, 12.

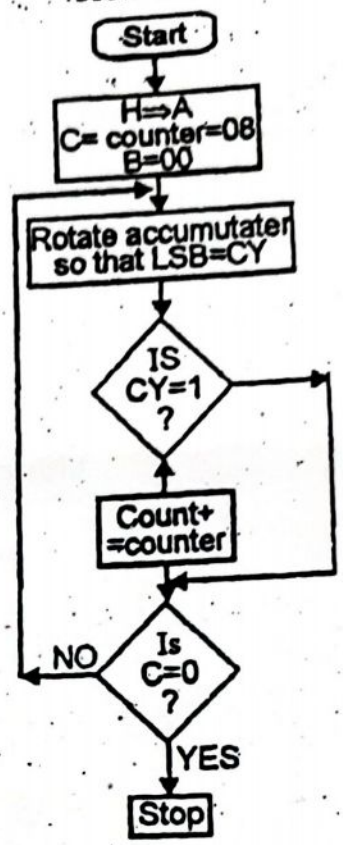
```
      MVI B 00H
      MVI A, 00H
      MVI C, 0A
XYZ   ADD B
      INR B
      INR B
      DCR C
      JNZ XYZ
      MOV D, A
      HLT.
```

Example 19. Count number of one's in a number.

Statement. Write a program to count number of 1's in the contents of H register and store the count in the B register

Label	Program
	MVI B, 00H
	MOV A, H
	MVI C, 08H
YAMUNA :	RAR
	JNC GANGA
	INR B
GANGA :	DCR C
	JNZ YAMUNA
	HLT.

Flow-chart



4.3. COMPLICATED PROGRAMS

Example 20. Write a program to count number of '1' and '0' bits in a register. Assume data is in C register and store number of '1' in D register, '0' in E register.

Program

```
        MOV A, C
        MVI D, 00H
        MVI B, 08H
UP :    RAL
        JNC BELOW
        INR D
BELOW : DCR B
        JNZ UP
        MVI A, 08
        SUB D
        MOV E, A.
        HLT.
```

Example 21. To find +ve number from an array of 10 elements.

Statement : Write a program to find +ve numbers in any array of 10 elements. Assume array starts from 6700H onwards. Store result at 7700H.

Program :

```
        LXI H, 6700H
        MVI C, 00H
XYZ :   MOV A, M
        RAL
        JC Below
```