

## Chapter 6

### I Program statement

1. Write and verify the op.
1. Load the number 8BH in register D.
2. Load the number 6FH in register C
3. Increment the contents of register C by one
4. Add the contents of register C and D and display the sum at the output port 1.

```

XX00    MVI    D, 8BH
01
02    MVI    C, 6FH
03
04    INR    C
05    MOV    A, C
06    ADD    D
07    OUT    PORT 1
08.    PORT 1
09    HLT.

```

$$\begin{array}{r}
 70H. \quad 0111 \quad 0000 \\
 8BH. \quad 1000 \quad 1011 \\
 \hline
 \underline{\text{FBH.}} \quad \boxed{0} \quad \underline{\underline{1111 \quad 1011}}
 \end{array}$$

CY

Flag status S=1, Z=0, CY=0

(Since addition of signed numbers not taking place the S=1 can be ignored. It is bit of amnesia to S, and so it is 1)

2. WAP to do the following

1. Load the number 30H in reg B and 39H in reg C
2. Subtract 39H from 30H
3. Display the answer at PORT 1

Steps:

1. 39H      0011 1001

2 1st comp.    1100 0110

3 Add 1    
$$\begin{array}{r} 0000 \quad 0001 \\ + 1100 \quad 0111 \\ \hline 1100 \quad 0111 \end{array}$$

4 Add 30H.    
$$\begin{array}{r} 0011 \quad 0000 \\ + 1111 \quad 0111 \\ \hline \boxed{0} \quad 1111 \quad 0111 \end{array}$$
  
C<sub>y</sub>.

5 Complement carry  $\boxed{1} \quad 1111 \quad 0111 \quad F7H.$

S=1, Z=0, C=1.

00 X00      MVI B, 30H

01

02      MVI C, 39H

03

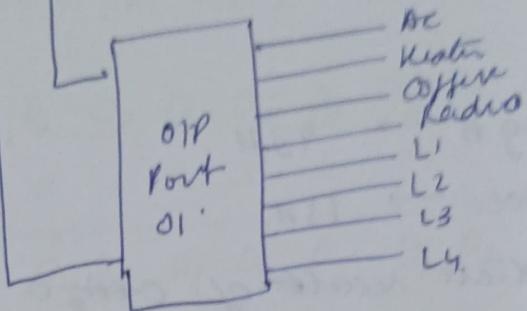
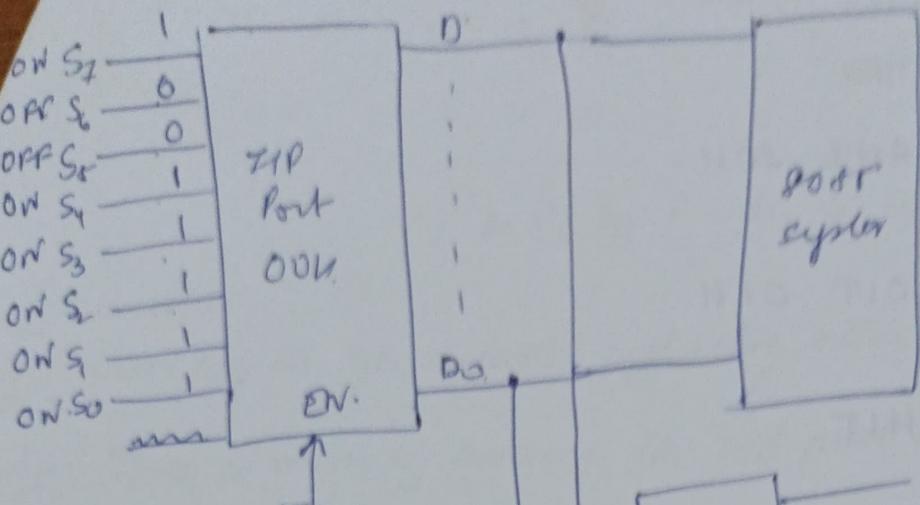
04.      MOV A, B

05      S0B C

06      OUT PORT1

07      PORT 1

08      HLT.



3. To conserve energy in a house
1. Turn on the air conditioner if switch S7 is off or the ZIP port OOU is on
  2. Ignore all other switches of the ZIP port even if someone attempts to turn on any other appliance below.
- ① Assuming some other switches are on. we get.  
QFU on input port.

Accumulator will have data  $\rightarrow$  QFU

- ② To know if switch S7 is on or not, mask the bit D6 through D0 by & ANDing with a byte with 0 in D6 to D0 position and 1 in D7

$$\begin{array}{ccccccccc}
 D_7 & D_6 & D_5 & D_4 & D_3 & D_2 & D_1 & D_0 \\
 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 = 80H
 \end{array}$$

- ③ After masking send the remaining byte to OUP port 01

XXOO MVI A, data

- 01 a 9H

02 ANI .80H.

03.

04. OUT 01H.

05

06 INT

eg Reg B  $\rightarrow$  93H

Acc  $\rightarrow$  15H.

Illustrate results of ORAB, XRAB and CMA

1. ORAB  $\rightarrow$  1001 0011

$$\begin{array}{r} 0001 \text{ } 0101 \\ \hline 10010111 \end{array} \text{ 93H.}$$

② XRAB  $\rightarrow$  1001 0011

$$\begin{array}{r} 0001 \text{ } 0101 \\ \hline 10000110 \end{array} \text{ 86H.}$$

③ CMA  $\rightarrow$  1110 1010

eg In the fig, keep radio on ( $D_4$ ) continuously without affecting the function of other appliances even if someone turns off switch  $S_4$ .

Sehr To keep radio on without affecting other appliances bit  $D_4$  should be set by using instruction byte 10H.

$$\begin{array}{l}
 \text{IN00N (A1} \quad D_7 \quad D_6 \quad D_5 \quad D_4 \quad D_3 \quad D_2 \quad D_1 \quad D_0 \\
 \text{OUT 10H.} \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \\
 \hline
 D_7 \quad D_6 \quad D_5 \quad 1 \quad D_3 \quad D_2 \quad D_1 \quad D_0
 \end{array}$$

In words,

(eg) Turn off the AC without affecting other appliances  
Soln This is done by ANDing the ZP port with TRH.

$$\begin{array}{l}
 * \text{ IN00N (A1: } \quad D_7 \quad D_6 \quad D_5 \quad D_4 \quad D_3 \quad D_2 \quad D_1 \quad D_0 \\
 \text{ANI TRH.} \quad 0 \quad 1 \\
 \hline
 0 \quad D_6 \quad D_5 \quad D_4 \quad D_3 \quad D_2 \quad D_1 \quad D_0
 \end{array}$$

### Unconditional Pump

2000 IN 00 H

2001

2002 OUT01H

2003

2004 JMP start

2005

2006

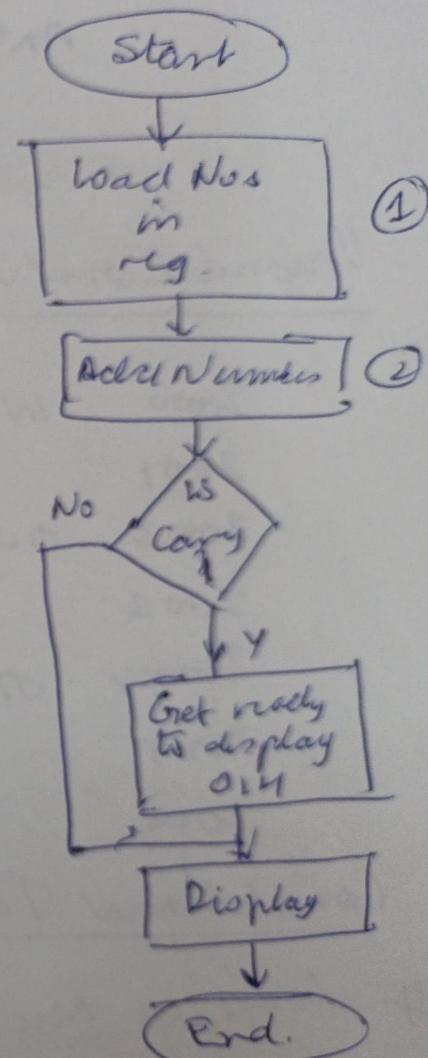
### Conditional Pump

Eg Load the two numbers 9BH and A7H in registers D and E respectively. Add the numbers.

① If sum > RRH display 01H at OPP Port 1

② Else display sum.

- 2000 MVI D, 9B4  
 2001  
 2002 MVI E, A7H  
 2003  
 2004 MOV A, 0  
 2005 ADD C  
 2006 INC DISPLAY  
 2007  
 2008  
 2009 MVI A, 01H  
 2010  
 2011  
 2012 MVR 00H (DISPLAY)  
 2013  
 2014 HLT



Smallest bytes of data stored in a memory location XX80H to XX8FH. Transfer the entire block of data to new memory location XX70H.

OpCode      Operand  
 XX00      SMPLR      LXZ      H, XX80H

01

02

03.      LXZ      D, XX70H

04

05

06.      MVZ      B, 10H      (Set up B to count  
16 bytes)

07

08      NEXT      MOV      A, M

09      SMAX      D

0A      INX      H

0B      INX      D

0C      DCR      B

0D      JNZ      NEXT

0E

0F

10

MLT

If counter not zero  
→ go back to transfer  
→ next byte

Ex. A set of 10 current readings is stored in a memory location starting at XX60H. The readings are expected to be positive ( $< 128_{10}$ ). What to

① check each reading to determine whether +ve or -ve

② reject all -ve readings

③ add all +ve readings

④ Our FFH is Port 1 at any time when the sum exceeds eight bits to indicate overload, otherwise

display the sum.

- ⑤ If no op. port is available in system go to step 6.
- ⑥ Store PFL in memory location XX10H when the sum exceeds 8 bits otherwise store sum.

Data

2B, DF, C2, 21, 24, 30, 2F, 19, F2, 9R

Code

Memory	Label	Op code	Op and	
XX00		MVZ	B,00H	(Clear B to sum sum)
01				
02		MVZ	C,0AH	Set up RC as counter
03				
04		LXZ	H,XX60H	(HL as memory pointer)
05				
06				
07	*NEXT	MOV	A,M	Get byte
08		RAL		Shift D7 in CY
09		J C	REJECT	If D7=1 reject byte and go to increment pointer
0A				
0B				
0C			RAR	If byte + M, restore
0D		ADD	B.	Add previous note A
0E		J C	OVRLD	If sum > PFL it is overload.
0F				
10.				
11		MOV	B,A	Save sum
12.	REJECT	INX	H.	Point to next ready
13		DCL	C	Decrement Counter

14

JNZ NEUT.

15

16

17

MOV A,B.

18.

OUR PORT1

19

20

1A

HLT.

1B

NOP

To match Jump location

1C

OVERLOAD

MNZ A, PSH

OVERLOAD in memory storage

1D

1E

OUR PORT1

1F

20

HLT.

RA RRM

RRC, RCL

### Some stack Instructions

PUSH - Data bytes or in a ny pair of the GP can be stored on the stack (two at a time) in reverse order (decreasing memory address) by using the instruction PUSH

POP - Data bytes can be transferred from the stack to the respective registers by using POP instruction

③ Because 2 bytes are being stored at a time, 16 bit address in SP register is decremented by 2 when data bytes are retrieved, the dest address is incremented

④ Address in SP indicates next 2 memory locations

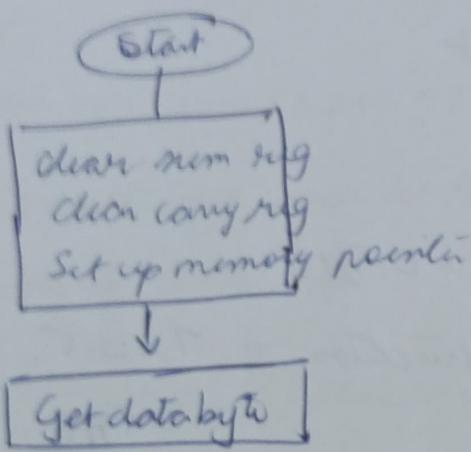
eg A set of readings is stored in a memory location at XX80H, the end of the data string is indicated by the byte 00H. Add the sets of readings. The answer may be larger than PCH. Then

Display the sum at PORT1 and PORT2 or store the answer at the memory location XX70 and XX71 H.

Ans 00 <sup>label</sup> Start LXI H, XX80H

01  
02  
03 MVI C, 00H Clear(C) to save sum  
04  
05 MOV B, C Clear B to save carry  
06 NXTBYTE MOV A, M  
07 CPI 00H  
  
08 00  
09 JZ DISPLAY If yes go to display  
0A  
0B  
0C ADD C  
0D JNC SAVE  
0E  
0F  
10 INRB \_\_\_\_\_ Update carry register  
11 SAVE MOV C, A  
12 INX H  
13 JMP NXTBYT Go back to next reading  
14 06  
15 XX

16	Display	MOV A, <sub>1C</sub>	
17		OUT Port 1	Display lower order bytes of sum
18			
19		MOV A, <sub>1B</sub>	
1A		OUT Port 2	Display higher order bytes of sum
1B			
1C		HLT	



### Assignment

- ① A set of 3 readings is stored in memory location starting at XX80H. Sort the readings in ascending order  
87, 56, 42
- ② The following block of data is stored in the memory location XX85H to XX8AH. Transfer the data to the location XX80H to XX85H in the reverse order (eg data byte 22H should be stored in XX85H)  
Data (H) : 22, A5, B2, 99, 7E, 37

Execution and flow control  
Delays and time delays

Time delay using 1 register

$T_{\text{state}}$

MVI C, FFH 7

loop DCR C 4

JNZ loop 10/7

① given clock freq = 2MHz

$$\text{Clock period } T = \frac{1}{f} = \frac{1}{2} = 0.5 \mu\text{s}$$

Time to execute MVI instruction =  $7 \times 0.5$

② C is loaded with FFH  
 $= 2^{15}_{10}$

③ MVI → executed once  $\rightarrow 7 T_{\text{state}}$

④ DCR and JNZ → form a loop  $\rightarrow 4 \times 10 = 14 T_{\text{state}}$

⑤ Loop is repeated 285 times

$\therefore$  Time delay in the loop

$$T_L = (T \times \text{Loop } T_{\text{state}} \times N_{10})$$

$T$  = system clock period

$N_{10}$  = Duinal equivalent of hex count

$$T_L = (0.5 \times 10^{-6} \times 14 \times 285)$$

$$= 1785 \mu\text{s}$$

107 -

107 states - required to execute a conditional jump when it jumps to change the sequence.

7T states - when it goes to the instruction following TNZ

∴ Last cycle will have 7T states.

∴ Loop delay would be calculated by subtracting 3 states time i.e

$$T_{L2} = T_L - (3T_{\text{states}} \times \text{clock period})$$

$$= 1785 \mu\text{s} - (3 \times 0.5)$$

$$= 1785 - 1.5 = 1783.5 \mu\text{s}$$

Now total delay

= time to execute instruction + loop outside loop

$$= (7 \times 0.5 \mu\text{s}) + 1783.5$$

$$= 1787 \mu\text{s} = 1.8 \text{ ms}$$

## Time delay using a register pair

		Total 10 BC $\rightarrow$ 16 bit count
Loop	DCX B	6
	MOV A,C	4
	ORA B	4
	JNZ Loop	10/7

Sdn

$$2384 = 2 \times (16)^3 + 3 \times (16)^2 + 8 \times (16)^1 + 4 \times (16)^0 \\ = 9092_{10}$$

Time delay = no. of clock periods required for  
 DCX, MOV, ORA, JNZ = 24T steps

$$T_L = (0.5 \times 24 \times 9092)$$

$$= 109 \text{ ms.}$$

3.745

$$\text{Total delay} = 109 \text{ ms} + (0.5 \times 7 \mu\text{s}) \\ \underline{\approx} 109 \text{ ms.}$$

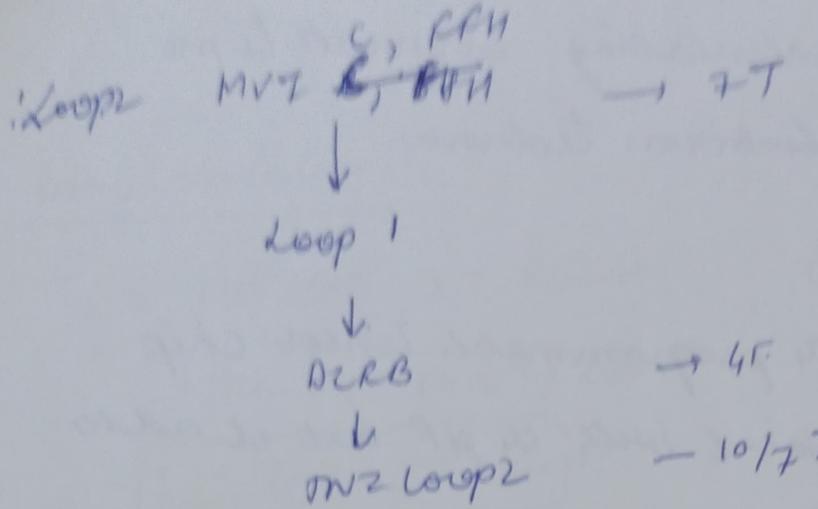
## Time delay using a loop within a loop

MVI B, 38H	7T
loop2 MVI C, RRH	7T
loop1 DCR C	4T
JNZ loop1	10/7 T
DCR B	4T
JNZ loop2	10/7 T

## Delay calculation

Loop 1 delay same as earlier

$$TL1 = 1783.5 \mu s$$



Sohm  $38_H = 3 \times 16^1 + 8 \times 16^0 = 58$

Combining loop 2 as 1 loop

$$\begin{aligned} TL2 &= 58 (TL1 + 21 \cdot T_{\text{status}} \times 0.545) \\ &= 58 (1783.5 + 10.545) \\ &= 100.46 \text{ ms} \end{aligned}$$

- ① Time delay outside the loop is negligible and hence can be ignored.
- ② Additional time delay can be added
  - ① by using register pair
  - ② by adding NOP instruction

## Disadvantage of using software delay

- ① The accuracy of the time delay depends on the accuracy of system clock

② The port is occupied simply in a waiting loop, otherwise it could be employed to perform other functions

③ The task of calculating accurate time always is tedious, tedious

### Solution

The Intel 8284 programmable timer chip can be interfaced with the µP but it adds to the cost

### Hexadecimal Counter

Q WAP to count continuously a hexadecimal from FFF to 001 in a system with a 0.5 µs clock period. Use register C to set up a 1 millisecond delay between each count and display the number at one of the output ports

Label

XX00.

MVI B, 001 → B as counter

01

Next. DCR B (FFF) → Increment counts

02

MVI C, count.

Load C with delay count

03

DCR C

Decrement delay count

04

JNZ NextDelay Delay

05

MOV A, B

06

07

08

09

OA

OUT PORT 1

OB

OC

JMP Next

OD

OE

Delay calculation

$$T_L = 14 \text{ Tstates} \times T \times \text{Count}$$

$$= 14 \times (0.5 \times 10^{-6}) \times \text{Count}$$

$$\boxed{T_L = 7.0 \times 10^{-6} \times \text{Count}}$$

( Delay loop  
DCRB & JNZ )Delay outside loop

DCRB	—	4T
MUL C, Count	—	7T
MOV A,B	—	4T
Out Port 1	—	10T
JMP	—	10T
		<u><u>35 Tstates</u></u>

TO = outside loop

$$= 35 \times 0.5 \times 10^{-6}$$

$$\boxed{TO = 17.5 \mu\text{s.}}$$

$$\text{Total delay} = TO + TL = \cancel{TO_{outside}} + TL$$

$$= T_{outside} + T_{loop}$$

$$\text{Lms.} = 17.5 \times 10^{-6} + (7.0 \times 10^{-6}) \times \text{Count}$$

$$\text{Lms.} = \frac{1 \times 10^{-6} - 17.5 \times 10^{-6}}{7.0 \times 10^{-6}}$$

$$= 140_{10} \rightarrow 8CH$$

$$\boxed{140_{10} \rightarrow 8CH} \rightarrow \text{Count} > 8CH$$

## Module -10- counter

Eg. Write a count from 0 to 9 with a 1 second delay between each count. At the count of 9 the routine should reset itself to 0 and repeat the sequence continuously. Use register pair HL to set up the delay and display each count at one of the output ports. Assume the clock frequency of the 411 is 1 MHz.

01	Start	MVZB,00H	—	counter
02				
03	Display	DUPORT #		
04				
05		LXI H, 16 bit $\frac{10}{10}$	To	
06	LD			delay
07	H1			
08	Loop	DEXH	6	
09		MOV A, L	4	
0A		ORA H	4	$T_2 \rightarrow 24T$ states stack
0B		JNZ Loop	10/7	
0C				
0D				
0E		INR B	4	
0F		MOV A,B	4	
10		CPZ OA H	7	
11		JNZ Display	10/7	O OPP Content = then 3rd flag set.
12				
13				
14				
15		JZ Start		

① ORAH → used to check the contents of H and L. (to check zero flag)  
cannot be done directly so the contents of L have to be moved to A  
ORed → to check if result is 0 or not  
HL has reached 0 or not

zero flag set only when both H and L are 0

② CPIA, OAH → compares the contents of counter (reg B) in every cycle.  
When reg B reaches the number OAH, the PC program sequence is reset/rewritten to reset the counter to zero without displaying OAH.

### Triple delay calculation

③ Time delay required to reset counter to zero is slightly different from the time delay between each count.

### Time delay calculation

Major delay provided by loop

$$T_L = 24 \times T_{\text{count}}$$

$$1 \text{ second} = 24 \times 1.0 \times 10^{-6} \times \text{count}$$

$$\text{Count} = \frac{1}{24 \times 10^{-6}} = 41666 \approx 42000$$

∴ Delay A2C2H in HL reg pair will provide a 1 second delay (approx)

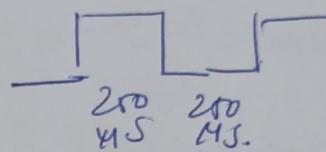
$$\text{Total delay} = T_0 + T_L$$

$$1 \mu\text{sec} = (45 \times 1.0 \times 10^{-6}) + (24.8 \times 10^{-6} \times \text{Count})$$

$$\text{Count} = 41665$$

Difference very little so insignificant

By WAP to generate a continuous square wave with the period of  $\frac{500}{250}$  MS, Assume the system clock is 325 ns and use bit D0 to O/R the square wave.



Solution

Step	Label	Instruction	Notes
00		MVZ D, AA.	Load AAH
01		MOV A, D.	
02	Rotabs	RLC.	
03		MOV D, A	
04		ANL 01H.	
05			
06			
07		OUT Port 1	
08			
09		MVZ B, Count.	(load delay of 250 MS)
0A			
0B	Delay	DCR B.	
0C		JNZ Delay.	
0D			
0E			
0F		SMP Rotabs	