

DATEPAGE NO.EXPT. NO.

## DAY-1

PROBLEM 1: Load  $23H$  to D Register and  $5643H$  in BC Register pair. Copy the content of D Register to A Register.

### Solution - 1

| ADDRESS | MNEMONICS      | HEX CODE |       |       |
|---------|----------------|----------|-------|-------|
| $9000H$ | MVI D, $23H$   | $16H$    | $23H$ | -     |
| $9002H$ | LXI B, $5643H$ | $01H$    | $43H$ | $56H$ |
| $9005H$ | MOV A, D       | $7AH$    | -     | -     |
| $9006H$ | RST 1          | $CFH$    | -     | -     |

Input: Reg B :  $56H$   
 Reg C :  $43H$   
 Reg D :  $23H$

Output : Reg A  $\leftarrow 23H$   
 Reg B  $\leftarrow 56H$   
 Reg C  $\leftarrow 43H$   
 Reg D  $\leftarrow 23H$

DATE

PAGE NO.

EXPT. NO.

Problem - 2 : Load  $44_H$  in Reg. A and  $08_H$  in Reg. B.

Add the content of Reg. B to Reg. A. Check result.

| Solution 2 | ADDRESS | MNEMONICS     | HEX    | CODE   |
|------------|---------|---------------|--------|--------|
|            | $8000H$ | MVI A, $44_H$ | $3FH$  | $44_H$ |
|            | $8002H$ | MVI B, $08_H$ | $06H$  | $08_H$ |
|            | $8004H$ | ADD B         | $80H$  | -      |
|            | $8005H$ | RST1          | $CFFH$ | -      |

Input: Reg A =  $44_H$   
Reg B =  $08_H$

Output: Reg A  $\leftarrow 4CH$   
Reg B  $\leftarrow 08H$   
Reg F  $\leftarrow 00H$

Problem - 3 : Load Register A with  $23H$  and Register B with  $08H$ , AND the content of register A and B and check result.

Solution - 3

| ADDRESS          | MNEMONICS    | HEX    | CODE  |
|------------------|--------------|--------|-------|
| <del>A000H</del> | MVI A, $23H$ | $3EH$  | $23H$ |
| <del>A002H</del> | MVI B, $08H$ | $06H$  | $08H$ |
| <del>A004H</del> | ANA B        | $A0H$  | -     |
| <del>A005H</del> | RST1         | $CFFH$ | -     |

Input: Reg A:  $23H$   
Reg B:  $08H$

Output: Reg A  $\leftarrow 00H$   
Reg F  $\leftarrow 54H$

MERIT®

DAY - 2PROBLEM - 1

Two 8 bit numbers are stored in memory locations  $C100H$  &  $C101H$ . Add the numbers and store the result in memory locations  $C102H$  &  $C103H$ .

| (LOC)<br>Address | (SOURCE CODE)<br>Mnemonics | OBJ<br>HEX CODE | COMMENTS  |
|------------------|----------------------------|-----------------|---|
| 8000             | MVI D, 00H                 | 16 00           | Load 00H into D register  |
| 8002             | LXT H, C100H               | 21 00 C1        | H points to C100 memory location  |
| 8005             | MOV A,M                    | 7E              | Move contents of memory to accumulator  |
| 8006             | LXT H, C101H               | 21 01 C1        | Loads an address in H   |
| 8009             | ADD M                      | 86              | Adds contents of M and accumulator<br><small>If no carry is generated</small> |
| 800A             | JNC 800F                   | D2 0E 80        | Jump to location 800F if<br>no carry is generated                             |
| 800D             | INR D                      | 14              | Increase address in D   |
| 800E             | INX H                      | 23              | Increase address in H   |
| 800F             | MOV M,A                    | 77              | Move contents of acc. to M  |
| 8010             | INX H                      | 23              | Increase address in H   |
| 8011             | MOV M,D                    | 72              | Move contents of D to M   |
| 8012             | RST1                       | CF              | Stop execution  |

Input  $\rightarrow$   $C100H : 23H$

$C101H : FAH$

Output  $\rightarrow$  ~~C100 23 FA 1D 01 \$~~ (i.e.  $C102H \rightarrow 1DH$   
 $C103H \rightarrow 01H$ )

DATE

PAGE NO.

EXPT. NO.

## PROBLEM - 2

Subtract the content of one register from the content of accumulator.

DATA : REGISTER CONTENT : FA<sub>H</sub>

ACCUMULATOR CONTENT : 79<sub>H</sub>

| ADDRESS (LOC) | MNEMONICS (SOURCE CODE) | HEX CODE (OBJ) |    | COMMENTS                              |
|---------------|-------------------------|----------------|----|---------------------------------------|
| 9000          | MVI D, 00 <sub>H</sub>  | 10             | 00 | Load 00 <sub>H</sub> to D register    |
| 9002          | MVI A, 79 <sub>H</sub>  | 3E             | 79 | Load 79 <sub>H</sub> to A register    |
| 9004          | MVI B, FA <sub>H</sub>  | 06             | FA | Load FA <sub>H</sub> to B register    |
| 9006          | SUB B                   | 90             |    | Subtract B from accumulator           |
| 9007          | JNC 9005                | D2             | 05 | Jump to 9005 if no carry is generated |
| 900A          | INR D                   | 14             |    | Increase address in D                 |
| 900B          | RST 1                   | CF             |    | Stop execution                        |

Output:

A = 7F<sub>H</sub>, B = FA<sub>H</sub> D = ?



Correction

JNC 900B

D = 01<sub>H</sub>

## DAY - 3

### PROBLEM - 1

Two 16-bit numbers are stored into the memory location starting from C200H to C203H. Add these data and results will be stored into next memory location.

**DATA:**

|             |              |                   |
|-------------|--------------|-------------------|
| C200H : 23H | ← Lower Byte | } 1st 2 byte data |
| C201H : EBH | Higher byte  |                   |

I/P  $\Rightarrow$

|             |          |                   |
|-------------|----------|-------------------|
| C202H : 79H | Lower "  | } 2nd 2 byte data |
| C203H : 72H | Higher " |                   |

O/P  $\Rightarrow$

|                     |          |
|---------------------|----------|
| C204H : Lower Byte  | } Result |
| C205H : Higher Byte |          |
| C206H : Carry Byte  |          |

| ADDRESS (LOC) | MNEMONICS  | SOURCE CODE | HEX CODE (OBJ) |    | COMMENTS   |
|---------------|------------|-------------|----------------|----|--|
| 8000          | MVI C, 00H |             | 0E             | 00 | Load 00H into C register                                       |
| 8002          | LHLD C200  |             | 2A             | 00 | C2 Directly load content of C200 and C201 in H-L register pair |
| 8005          | XCHG       | EB          |                |    | Exchange the data of D-E and H-L register pair                 |
| 8006          | LHLD C202  |             | 2A             | 02 | C2 Directly load content of C202 and C203 in H-L register pair |

DATE      PAGE NO.

EXPT. NO.

| <del>ANALY</del><br>LOC | SOURCE CODE | HEX CODE | COMMENT   |
|-------------------------|-------------|----------|---|
| 8009                    | DAD D       | 19       | (Double) Addition of D-E register pair with H-L register pair |
| 800A                    | JNC 800E    | D2 0E 80 | Jump to 800E if carry is not generated                        |
| 800D                    | INRC        | 0C       | Increase address in C   |
| 800E                    | SHLD C204   | 22 04 C2 | The contents of H-L register are stored in specified location |
| 8011                    | MOV A,C     | 79       | Move contents of A to C reg.                                  |
| 8012                    | STA C206    | 32 06 C2 | Store accumulator content to specified address                |
| 8015                    | RST 1       | CF       | Stop execution.   |

Output : C204 : 9CH

C205 : 5DH

C206 : 01H

## PROBLEM-2

Four 8-bit numbers are stored in memory location starting from C200H. Perform the BCD addition on these numbers and store the results in the next memory location.

|       |             |
|-------|-------------|
| DATA: | C200H : 23H |
| I/P   | C201H : 44H |
|       | C202H : 79H |
|       | C203H : 72H |

|     |                |
|-----|----------------|
| O/P | C204H : Result |
|     | C205H : Carry  |

| ADDRESS<br>(LOC) | MNEMONICS<br>(SOURCE CODE) | HEX CODE (OBJ) | COMMENTS                                     |
|------------------|----------------------------|----------------|--|
| 8000             | MVI B, 00H                 | 06 00          | Load 00H to B register                       |
| 8002             | MVI C, 04H                 | 0E 04          | Load 04H to C register                       |
| 8004             | MVI A, 00H                 | 3E 00          | Load 00H to A register                       |
| 8006             | LXI H, C200                | 21 00          | H-L points to C200 memory location           |
| 8009             | ADD M                      | 86             | Add contents of M and acc                    |
| 800A             | DAA                        | 27             | Allows addition of numbers in 8 bit BCD code |
| 800B             | JNC 800F                   | D2 0F 80       | Jumps to 800F if no carry is generated.      |
| 800E             | INR B                      | 04             | Increases B register                         |
| 800F             | INX H                      | 23             | Increases address in H-L                     |
| 8010             | DCR C                      | 0D             | Decrement of C by 1                          |

DATE

PAGE NO.

EXPT. NO.

|      |          |    |    |    |   |
|------|----------|----|----|----|---|
| S011 | JNZ 8009 | C2 | 09 | 80 | Jumps to 8009 if C is not zero                  |
| S014 | STA C204 | 32 | 04 | C2 | Stores accumulator content to specified address |
| S017 | MOV A,B  | 78 |    |    | Move contents of B to acc.                      |
| S018 | STA C205 | 32 | 05 | C2 | Stores accumulator content to specified address |
| S01B | RST 1    |    | CF |    | Stop execution                                  |

Output: C204 : 18 <sup>H</sup> (Result)

C205 : 02 <sup>H</sup>. (Carry)

~~Always 00/00/00~~

# DAY - 4

## PROBLEM - 1

Write a program for packing and unpacking of BCD numbers.

| ADDRESS<br>(LOC) | MNEMONICS<br>(SOURCE CODE) | HEX CODE (OBJ) | COMMENTS                                       |
|------------------|----------------------------|----------------|--|
| 8000             | MVI A, 57H                 | 3E 57          | Load 57H to A register                         |
| 8002             | MOV D,A                    | 57             | Move contents of A to D register               |
| 8003             | ANI OF                     | E6 0F          | We AND the contents of A with 0F               |
| 8005             | MOV B,A                    | 47             | Move contents of A to B register               |
| 8006             | MOV A,D                    | 7A             | Move contents of D to A register               |
| 8007             | ANT FO                     | E6 F0          | We AND the contents of A with F0               |
| 8009             | RRRC                       | OF             | rotate<br>Shift all bits in acc. by 1 to right |
| 800A             | RRRC                       | OF             | rotate<br>Shift all bits in acc. by 1 to right |
| 800B             | RRRC                       | OF             | rotate<br>Shift all bits in acc. by 1 to right |
| 800C             | RRRC                       | OF             | rotate<br>Shift all bits in acc. by 1 to right |
| 800D             | MOV C,A                    | 4F             | Move contents of A to C register               |
| 800E             | RRRC                       | OF             | rotate<br>Shift all bits in acc. by 1 to right |
| 800F             | RRRC                       | OF             | rotate<br>Shift all bits in acc. by 1 to right |
| 8010             | RRRC                       | OF             | rotate<br>Shift all bits in acc. by 1 to right |
| 8011             | RRRC                       | OF             | rotate<br>Shift all bits in acc. by 1 to right |
| 8012             | ORA B                      | BO             | OR the contents of A with B                    |
| 8013             | MOV E,A                    | 5F             | Move contents of A to E register               |
| 8014             | RST 1                      | CF             | Stop execution                                 |

Output :- B = 07<sub>H</sub> (lower nibble)      Input :- 57<sub>H</sub>  
 C = 05<sub>H</sub> (upper nibble)  
 E = 57<sub>H</sub> (packed BCD)

## PROBLEM-2

Load 9E<sub>H</sub> into memory location C100<sub>H</sub>. Convert the number into ASCII code and store the result in next memory location.

| ADDRESS<br>(Loc) | MNEMONICS<br>(SOURCE CODE) | HEX CODE (OBJ) | COMMENTS |  |   |
|------------------|----------------------------|----------------|----------|--|---|
| 8000             | MVI C, 00 <sub>H</sub>     | 0E 00          |          |  | Load 00 <sub>H</sub> to C register              |
| 8002             | LXI H, C100                | 21 00 C1       |          |  | H-L points to C100 memory location              |
| 8005             | MOV A, M                   | 7E             |          |  | Move contents of M to A                         |
| 8006             | MOV D, A                   | 57             |          |  | Move contents of A to D                         |
| 8007             | ANI 0F <sub>H</sub>        | E6 0F          |          |  | AND the contents of A to 0F                     |
| 8009             | MOV B, A                   | 47             |          |  | Move contents of A to B                         |
| 800A             | MOV A, D                   | 7A             |          |  | Move contents of D to A                         |
| 800B             | ANI F0 <sub>H</sub>        | E6 F0          |          |  | AND the contents of A to F0                     |
| 800D             | RRC                        | OF             |          |  | rotate<br>Shift all bits in acc. by 1 to right  |
| 800E             | RRC                        | OF             |          |  | rotate<br>Shift all bits in acc. by 1 to right  |
| 800F             | RRC                        | OF             |          |  | rotate<br>Shift all bits in acc. by 1 to right  |
| 8010             | RRC                        | OF             |          |  | rotate<br>Shift all bits in acc. by 1 to right  |
| 8011             | MOV E, A                   | 5F             |          |  | Move contents of A to E register                |
| 8012             | CPI OA                     | FE OA          |          |  | Compares <sup>contents</sup> value of A with OA |
| 8014             | JNC 801C                   | D2 1C 80       |          |  | Jumps to 801C if no carry is generated          |
| 8017             | ADI 30 <sub>H</sub>        | C6 30          |          |  | Adds 30 to acc.                                 |
| 8019             | JMP 801E                   | C3 1E 80       |          |  | Jumps to 801E memory location                   |

| ADDRESS (Loc)                                  | MNEMONICS<br>(SOURCE CODE) | HEXCODE (OBJ)       | COMMENTS                                     |
|--|----------------------------|---------------------|--|
| 801C   | ADT 37H                    | C6 37               | Adds 37 to acc.                              |
| 801E   | STA C101                   | 32 01               | C1 stores contents of acc to C101 location   |
| 8021   | MOV A,B                    | 78                  | move contents of B to A                      |
| 8022   | CPI 0AH                    | FE 0A               | Compares content of acc to 0A<br>to generate |
| 8024   | JNC 802C                   | D2 2C               | jumps to 802C location if no carry           |
| 8027   | ADT 30H                    | C6 30               | adds 30 to acc.                              |
| 8029   | JMP 802E                   | C3 2E               | jumps to 802E location                       |
| 802C   | ADI 37H                    | C6 37               | Adds 37 to acc content.                      |
| 802E   | STA C102                   | 32 02               | C1 stores contents of acc to C102 location   |
| 8031   | RST 1                      | CF                  | stop execution                               |
| <del>Output :- C101 : 39H<br/>C102 : 45H</del> |                            | Input :- C100 = 9EH |  |

## DAY-5

### PROBLEM-1

Shift a block of data starting from memory location  $C050_H$  to another block starting from  $C000_H$ .

| ADDRESS<br>(LOC) | MNEMONICS<br>(SOURCE CODE) | HEX CODE (OBJ) |    |    | COMMENTS   |
|------------------|----------------------------|----------------|----|----|--|
| 8000             | MVI C, 0AH                 | 0E             | 0A |    | Load 0AH to C register   |
| 8002             | LXI H, C050                | 21             | 50 | C0 | H-L reg. pair points to C050 <sub>H</sub> memory location          |
| 8005             | LXI D, C000                | 11             | 00 | C0 | D-E points to C000 memory location                                 |
| 8008             | MOV A,M                    | 7E             |    |    | Move contents of M to A  |
| 8009             | STAX D                     | 12             |    |    | Stores content of accumulator in location pointed by D-E reg. pair |
| 800A             | INX D                      | 13             |    |    | Increases address in D-E   |
| 800B             | MVI M, 00H                 | 36             | 00 |    | Load 0D <sub>H</sub> to memory location pointed by H-L reg. pair   |
| 800D             | INX H                      | 23             |    |    | Increase address in H-L  |
| 800E             | DCR C                      | 0D             |    |    | Decrease content of C by 1   |
| 800F             | JNZ 8008                   | C2             | 08 | 80 | Jump to 8008 if content of C register is not zero                  |
| 8012             | RST 1                      | CF             |    |    | stop execution   |

Output:-  
 C000: 01<sub>H</sub>  
 C001: 02<sub>H</sub>  
 C002: 03<sub>H</sub>  
 C003: 04<sub>H</sub>

Input:-  
 C050: 01<sub>H</sub>  
 C051: 02<sub>H</sub>  
 C052: 03<sub>H</sub>  
 C053: 04<sub>H</sub>

OUTPUT (contd)

C004: 05H

C005: 06H

C006: 07H

C007: 08H

C008: 09H

C009: 0AH

INPUT (contd)

C054: 05H

C055: 06H

C056: 07H

C057: 08H

C058: 09H

C059: 0AH

C050: 00H

C051: 00H

C052: 00H

? : :

C059: 00H

PROBLEM-2

Write a program to find minimum of 5 numbers.

| ADDRESS<br>(LOC) | MNEMONICS (SOURCE<br>CODE) | HEX CODE (OBJ) |    |    | COMMENTS                              |
|------------------|----------------------------|----------------|----|----|---------------------------------------|
| 8000             | MVI C, 05H                 | OE             | 05 |    | Load 05H to C register                |
| 8002             | LXI H, C200                | 21             | 00 | C2 | H-L points to C200 memory location    |
| 8005             | MOV A, M                   | 7E             |    |    | Move contents of M to A               |
| 8006             | MOV B, A                   | 47             |    |    | Move contents of A to B               |
| 8007             | MOV A, M                   | 7E             |    |    | Move contents of M to A               |
| 8008             | CMP B                      | B8             |    |    | Compares content of B with that of A  |
| 8009             | JNC 800D                   | D2             | 0D | 80 | Jump to 800D if no carry is generated |
| 800C             | MOV B, A                   | 47             |    |    | Move contents of BA to B              |
| 800D             | INX H                      | 23             |    |    | Increase address in H-L.              |

DATE

PAGE NO.

EXPT. NO.

| ADDRESS<br>(LOC) | MNEMONICS<br>(SOURCE CODE) | HEX CODE (OBJ) | COMMENTS                                 |
|------------------|----------------------------|----------------|--|
| 800E             | DCR C                      | 0D             | Decrease content of C by 1               |
| 800F             | JNZ 8007                   | C2 07 80       | Jump to 8007 if content of C is not zero |
| 8012             | RST 1                      | CF             | Stop execution                           |

Output:- B = 01H

Input: C200 : 07H

C201 : 03H

C202 : 05H

C203 : 01H

C204 : 09H

~~Amanesh  
20/05/19~~

## DAY-6

### PROBLEM-1

A string of readings is stored in memory locations starting at  $C070_H$ , and the end of the string is indicated by the byte  $OD_H$ . Write a program to check each byte in the string, and save the bytes in the range of  $30_H$  to  $39_H$  (both inclusive) in memory locations starting from  $C090_H$ .

DATA :  $35_H, 2F_H, 30_H, 39_H, 3A_H, 37_H, 7F_H, 31_H, OD_H, 32_H$

| ADDRESS (LOC) | (SOURCE MNEMONICS CODE) | HEX CODE(OBJ) | COMMENTS  |
|---------------|-------------------------|---------------|---|
| 8000          | LXI H, C070             | 21 70 C0      | H-L points to C070 memory location                              |
| 8003          | LXI D, C090             | 11 90 C0      | D-E points to C090 memory location                              |
| 8006          | MOV A,M                 | 7E            | Move contents of M to A   |
| 8007          | CPI OD <sub>H</sub>     | FE OD         | Compares content of acc. to OD <sub>H</sub>                     |
| 8009          | JZ 801C                 | CA 1C 80      | Jumps to 801C location if 0 is in acc.                          |
| 800C          | CPI 30 <sub>H</sub>     | FE 30         | Compares content of acc. to 30 <sub>H</sub>                     |
| 800E          | JC 8018                 | DA 18 80      | Jump to 8018 location if carry is 1 generated                   |
| 8011          | CPI 3AH                 | FE 3A         | Compares content of acc to 3AH                                  |
| 8013          | JNC 8018                | D2 18 80      | Jump to 8018 location if no carry is generated                  |
| 8016          | STAX D                  | 12            | Stores acc. content in memory location pointed by D-E reg. pair |
| 8017          | INX D                   | 13            | Increases address in D-E by 1                                   |
| 8018          | INX H                   | 23            | Increases address in H-L  |
| 8019          | JMP 8006                | C3 06 80      | Jump to location 8006   |

| ADDRESS (LOC) | MNEMONICS (SOURCE CODE) | HEX CODE (OBJ) |  |  | COMMENTS       |
|---------------|-------------------------|----------------|--|--|----------------|
| 801C          | RST 1                   | CF             |  |  | Stop execution |

Output :-

C090: 35H

C091: 30H

C092: 39H

C093: 37H

C094: 31H

Input:-

C070: 35H

C071: 2FH

C072: 30H

C073: 39H

C074: 3AH

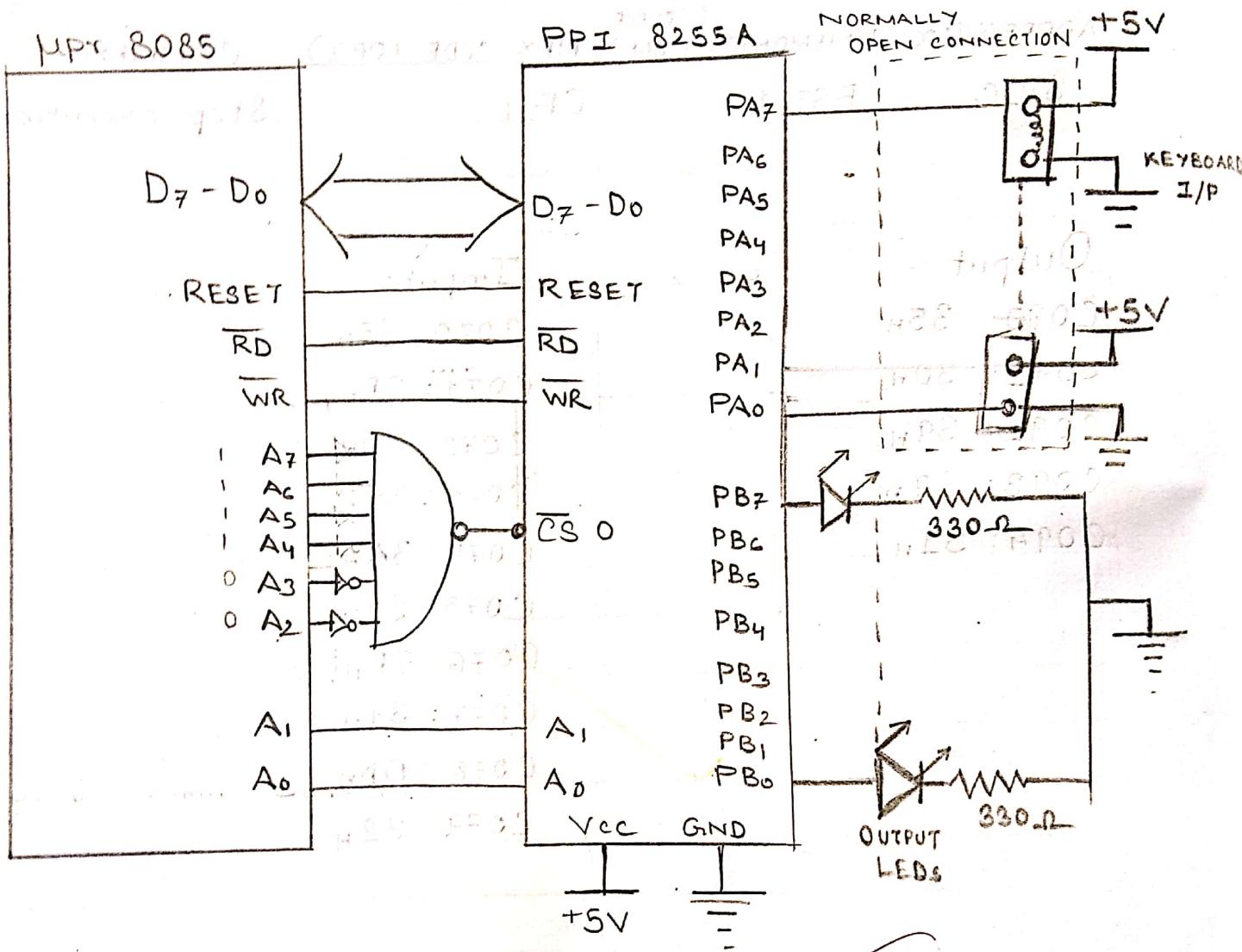
C075: 37H

C076: 7FH

C077: 31H

C078: 0DH

C079: 32H



~~FIGURE SHOWING INTERFACING OF OUTPUT LEDS AND KEYBOARD  
I/P WITH MPR 8085A via PPI 8255A~~

DAY - 7 :

PROBLEM - 1

Interface O/P and I/P with ppr 8085 such a way, that LEDs will glow according to the Keyboard I/P.

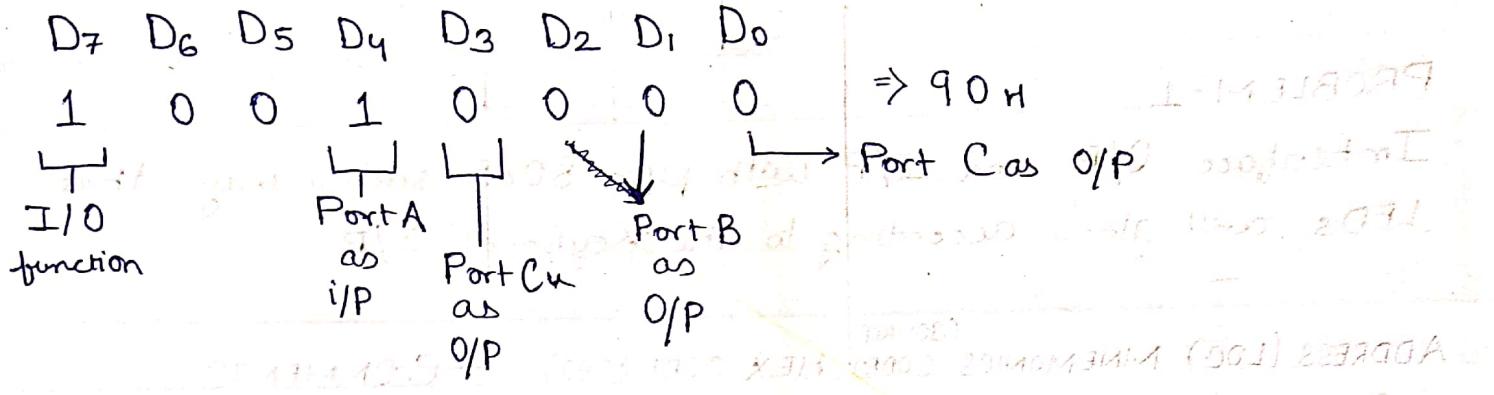
| ADDRESS (LOC) | MNEMONICS (CODE) | (SOURCE)<br>HEX CODE (OBJ) | COMMENTS                                     |
|---------------|------------------|----------------------------|--|
| 8000          | MVI A, 90H       | 3E 90                      | Load 90H to A register                       |
| 8002          | OUT F3           | D3 F3                      | 90H is loaded to Control Word Register (CWR) |
| 8004          | IN F0            | DB F0                      | Port A is taken as input                     |
| 8006          | OUT F1           | D3 F1                      | Port B is taken as output                    |
| 8008          | JMP 8004         | C3 04 80                   | Jump to 8004 location                        |

PROBLFM - 2

Interfacing with switches and LEDs and glowing LEDs according to read switch status and blinking using delay Subroutines through PPI 8255A with 8085A trainer kit.

| ADDRESS (LOC) | MNEMONICS (CODE) | (SOURCE)<br>HEX CODE (OBJ) | COMMENTS                         |
|---------------|------------------|----------------------------|----------------------------------|
| 9000          | MVI A, 80H       | 3E 80                      | Load 80H to A register           |
| 9002          | OUT F3           | D3 F3                      | 80H is loaded to CWR             |
| 9004          | MVI A, 05H       | 3E 05                      | Load 05H to A register           |
| 9006          | OUT F1           | D3 F1                      | Port B is taken as output        |
| 9008          | CALL C300        | CD 00 C3                   | Delay code at C300 loc is called |

## CONTROL WORD REGISTER CONFIGURATION (CWR)



OUTPUT - A of port A = 1 → 90H → 0001 0000 1001 0000 1000 0000 0000 0000 → 0008

Keyboard Input :-  
 i) 1111 0000  
 ii) 1010 0111

LED Output :-  
 i) 1111 0000 → 0008  
 ii) 1010 0111 → 0008

LED Output (for Blinking)

0000 0000

1111 1111

0000 0000

1111 1111

0000 0000

1111 1111

0000 0000

1111 1111

0000 0000

1111 1111

0000 0000

| ADDRESS (LOC) | MNEMONICS (CODE) | (SOURCE) | HEX CODE (OBJ) | COMMENTS                          |
|---------------|------------------|----------|----------------|-----------------------------------|
| 900B          | MVI A, 00H       |          | 3E 00          | 00H is loaded to A register       |
| 900D          | OUT F1           |          | D3 F1          | Port B taken as output            |
| 900F          | CALL C300        |          | CD 00 C3       | Delay code at C300 loc. is called |
| 9012          | JMP 9004         |          | C3 04 90       | Jump to 9004 location             |

### DELAY (FUNCTION) :

| ADDRESS (LOC) | MNEMONICS (CODE) | (SOURCE) | HEX CODE (OBJ) | COMMENTS  |
|---------------|------------------|----------|----------------|---|
| C300          | MVI B, FFH       |          | 06 FF          | FFH is loaded to B register                             |
| C302          | MVI C, FFH       |          | 0E FF          | FFH is loaded to C register                             |
| C304          | DCR C            |          | 0D             | Decrease contents of C register by 1                    |
| C305          | JNZ C304         |          | C2 04 C3       | If contents of C is not zero then jump to C304 location |
| C308          | DCR B            |          | 05             | Decrease contents of B register by 1                    |
| C309          | JNZ C302         |          | C2 02 C3       | If contents of B is not zero then jump to C302 location |
| C30C          | RET              |          | C9             | Return control to main program                          |

With self explanatory words

MERIT®

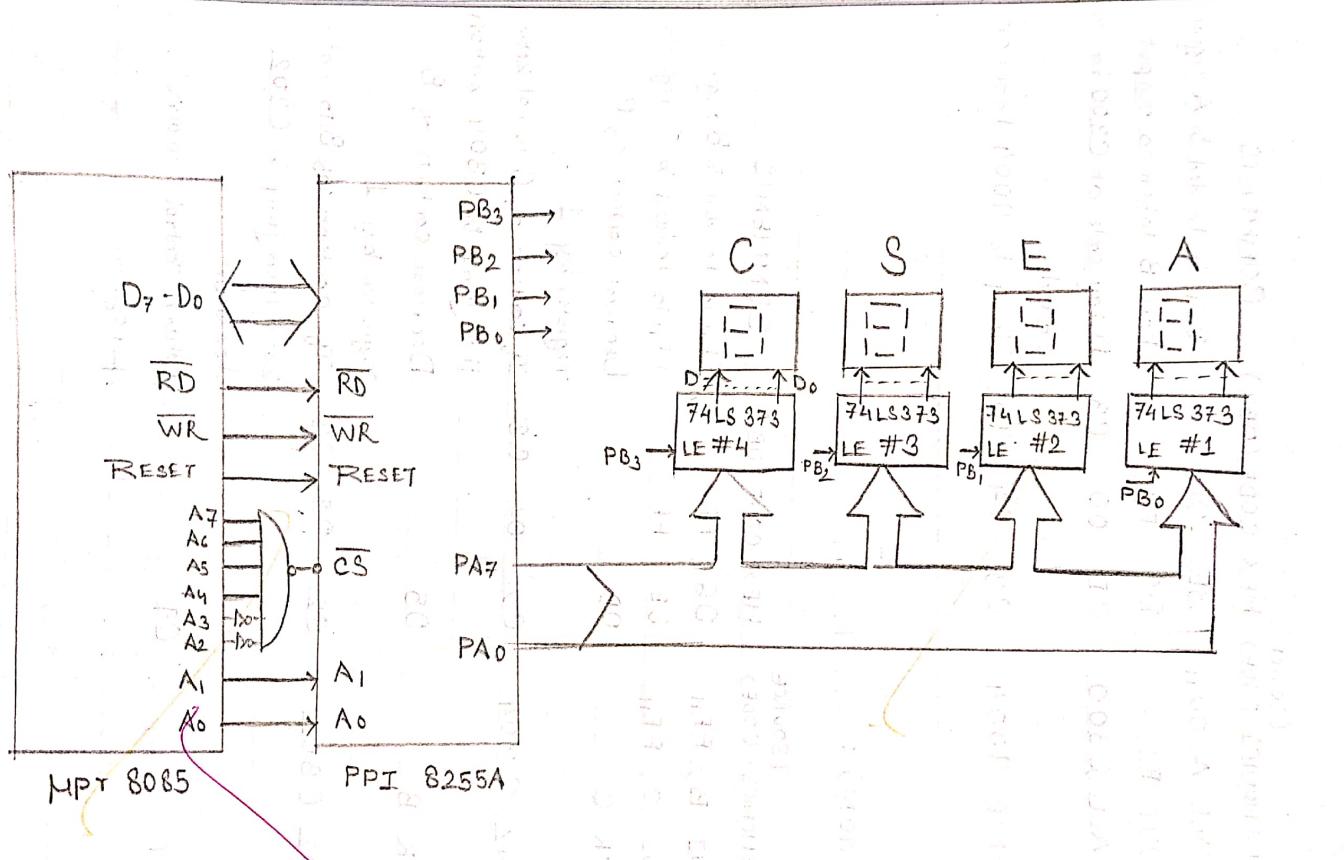


FIG : INTERFACING CC SEVEN SEGMENT DISPLAY WITH  $\mu$ pr 8085 VIA PPI 8255A  
(COMMON CATHODE)

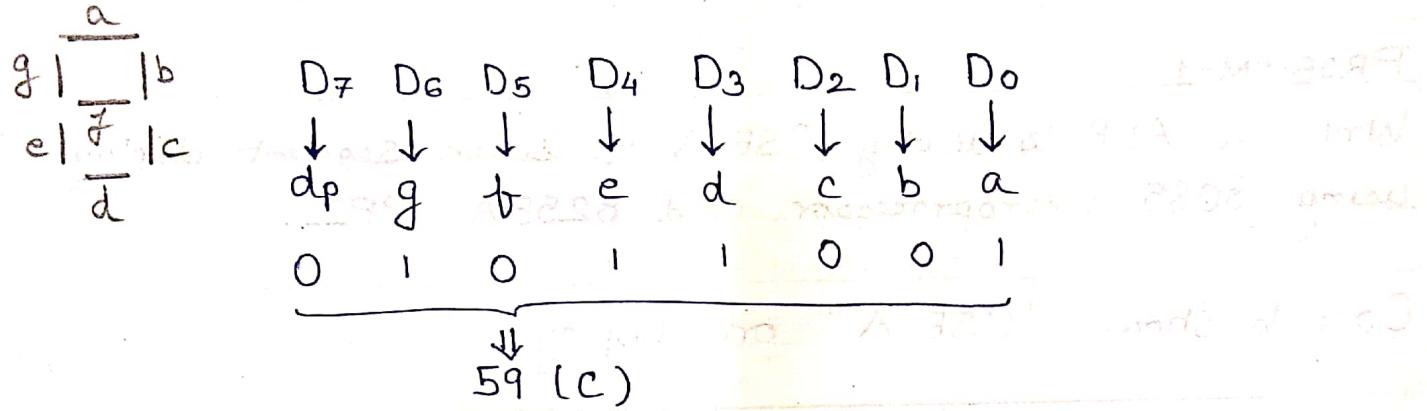
DAY - 8PROBLEM-1

Write an ALP to display CSE A in seven segment display using 8085 microprocessor and 8255A PPI.

Code to show "CSE A" on display

| ADDRESS(LOC) | MNEMONICS<br>(SOURCE CODE) | HEXCODE (OBJ) |
|--------------|----------------------------|---------------|
| 8000         | MVI A, 80H                 | 3E 80         |
| 8002         | OUT F3                     | D3 F3         |
| 8004         | MVI A, 59H                 | 3E 59         |
| 8006         | OUT F0                     | D3 F0         |
| 8008         | MVI A, 08H                 | 3E 08         |
| 800A         | OUT F1                     | D3 F1         |
| 800C         | MVI A, 00H                 | 3E 00         |
| 800E         | OUT F1                     | D3 F1         |
| 8010         | MVI A, GDH                 | 3E 6D         |
| 8012         | OUT F0                     | D3 F0         |
| 8014         | MVI A, 04H                 | 3E 04         |
| 8016         | OUT F1                     | D3 F1         |
| 8018         | MVI A, 00H                 | 3E 00         |
| 801A         | OUT F1                     | D3 F1         |
| 801C         | MVI A, 79H                 | 3E 79         |
| 801E         | OUT F0                     | D3 F0         |
| 8020         | MVI A, 02H                 | 3E 02         |

## Seven Segment Display



Similarly : S → 6D<sub>H</sub>

E → 79<sub>H</sub>

A → 77<sub>H</sub>

CWR

| D <sub>7</sub> | D <sub>6</sub> | D <sub>5</sub> | D <sub>4</sub> | D <sub>3</sub> | D <sub>2</sub> | D <sub>1</sub> | D <sub>0</sub> |
|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| 1              | 0              | 0              | 0              | 0              | 0              | 0              | 0              |

All ports are set to O/P mode

DATE

PAGE NO.

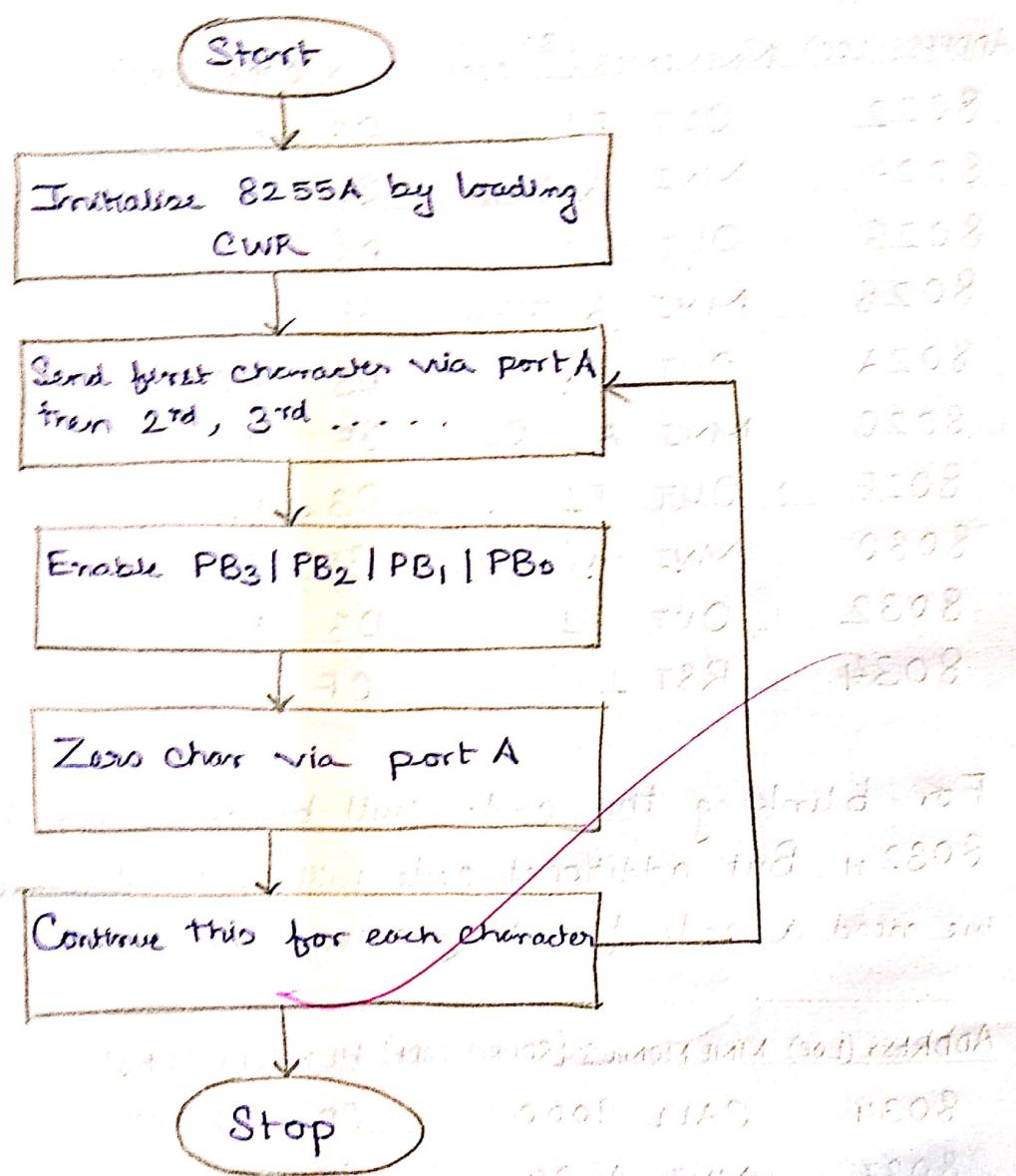
EXPT. NO.

| ADDRESS (LOC) | MNEMONICS (SOURCE CODE) | HEX CODE (OBJ) |    |
|---------------|-------------------------|----------------|----|
| 8022          | OUT F1                  | D3             | F1 |
| 8024          | MVI A, 00H              | 3E             | 00 |
| 8026          | OUT F1                  | D3             | F1 |
| 8028          | MVI A, 77H              | 3E             | 77 |
| 802A          | OUT FD                  | D3             | F0 |
| 802C          | MVI A, 01H              | 3E             | 01 |
| 802E          | OUT F1                  | D3             | F1 |
| 8030          | MVI A, 00H              | 3E             | 00 |
| 8032          | OUT F1                  | D3             | F1 |
| 8034          | RST 1                   | CF             |    |

For blinking the code will be same from 8000H to 8032H. But additional code will be written from 8034H provided we need a code for delay which will be there at 9000H

| ADDRESS (LOC) | MNEMONICS (SOURCE CODE) | HEX CODE (OBJ) |       |
|---------------|-------------------------|----------------|-------|
| 8034          | CALL 9000               | CD             | 00 90 |
| 8037          | MVI A, 00H              | 3E             | 00    |
| 8039          | OUT FD                  | D3             | F0    |
| 803B          | MVI A, 0F <sub>H</sub>  | 3E             | 0F    |
| 803D          | OUT F1                  | D3             | F1    |
| 803F          | MVI A, 00H              | 3E             | 00    |
| 8041          | OUT F1                  | D3             | F1    |
| 8043          | CALL 9000               | CD             | 00 90 |
| 8046          | JMP 8004                | C3             | 04 80 |

## LOW CHART :



## DELAY CODE:

| ADDRESS (LOC) | MNEMONICS (SOURCE CODE) | HEXCODE (OBJ) |          |  |  |
|---------------|-------------------------|---------------|----------|--|--|
| 9000          | MVI D, FF <sub>H</sub>  | 16 FF         |          |  |  |
| 9002          | MVI E, FF <sub>H</sub>  | 1E FF         |          |  |  |
| 9004          | DCR E                   | 1D            |          |  |  |
| 9005          | JNZ 9004                | C2 04 90      |          |  |  |
| 9008          | DCR D                   | 15            |          |  |  |
| 9009          | JNZ 9002                | C2 02 90      | 18 05 36 |  |  |
| 900C          | RET                     | C9            |          |  |  |

i/p for PROBLEM -1

C200 - 05 80 81

o/p for PROBLEM -1

C200 - 05 80 81 07 19 05 25 102

outputs

garbage value  
as no value was  
stored in this location

## DAY - 9

### PROBLEM-1

Five numbers are stored in memory starting from C200H location. Add all positive numbers and store the result into memory location C206H. Find the total number of negative numbers and store the same into memory location C207H.

| ADDRESS (LOC) | MNEMONICS   | (SOURCE CODE) | HEXCODE (OBJ) | Comments |
|---------------|-------------|---------------|---------------|----------|
| 8FFE          | IXT H, C200 | 21 00         | C2            |          |
| 9001          | MVI C, 00H  | 0E 00         |               |          |
| 9003          | MVI D, 05H  | 16 05         |               |          |
| 9005          | MVI E, 00H  | 1E 00         |               |          |
| 9007          | MVI B, 00H  | 06 00         |               |          |
| 9009          | MOV A, M    | 7E            |               |          |
| 900A          | RLC         | 07            |               |          |
| 900B          | JC 9016     | DA            | 16 90         |          |
| 900E          | MOV A, C    | 79            |               |          |
| 900F          | NOP         | 00            |               |          |
| 9010          | MOV B, M    | 46            |               |          |
| 9011          | ADD B       | 80            |               |          |
| 9012          | MOV C, A    | 4F            |               |          |
| 9013          | JMP 9017    | C3            | 17 90         |          |
| 9016          | TMR F       | 1C            |               |          |
| 9017          | INX H       | 23            |               |          |
| 9018          | DCR D       | 15            |               |          |

Output after passing a value of 100 to the  
function and selecting choice 2 and then  
the output of the function is 300, which  
is the required output.

I/P for PROBLEM-2 : NIL

O/P for PROBLEM-2 :

C 200 00 -

01 -

01 -

02 -

03 -

05 -

08 -

0D -

15 -

22 -

| ADDRESS (LOC) | MNEMONICS | (SOURCE CODE) | HEXCODE (OBJ) |    |    |
|---------------|-----------|---------------|---------------|----|----|
| 9019          | JNZ 9009  |               | C2            | 09 | 90 |
| 901C          | MOV A,C   |               | 79            |    |    |
| 901D          | STA C206  |               | 32            | 06 | C2 |
| 9020          | MOV A,E   |               | 7B            |    |    |
| 9021          | STA C207  |               | 32            | 07 | C2 |
| 9024          | RST 1     |               | CF            |    |    |

## PROBLEM-2

Write a program to generate a fibonacci series.

| ADDRESS (LOC) | MNEMONICS   | (SOURCE CODE) | HEX CODE (OBJ) |    |    |
|---------------|-------------|---------------|----------------|----|----|
| 8000          | MVI D, 0AH  |               | 16             | 0A |    |
| 8002          | MVI A, 00H  |               | 3E             | 00 |    |
| 8004          | MVI B, 01H  |               | 06             | 01 |    |
| 8006          | LXI H, C200 |               | 21             | 00 | C2 |
| 8009          | MOV M,A     |               | 77             |    |    |
| 800A          | ADD B       |               | 80             |    |    |
| 800B          | MOV C,A     |               | 4F             |    |    |
| 800C          | MOV A,B     |               | 78             |    |    |
| 800D          | MOV B,C     |               | 41             |    |    |
| 800E          | JNX H       |               | 23             |    |    |
| 800F          | MOV M,A     |               | 77             |    |    |
| 8010          | DCR D       |               | 15             |    |    |
| 8011          | JNZ 800A    |               | C2             | 0A | 80 |
| 8014          | RST 1       |               | CF             |    |    |

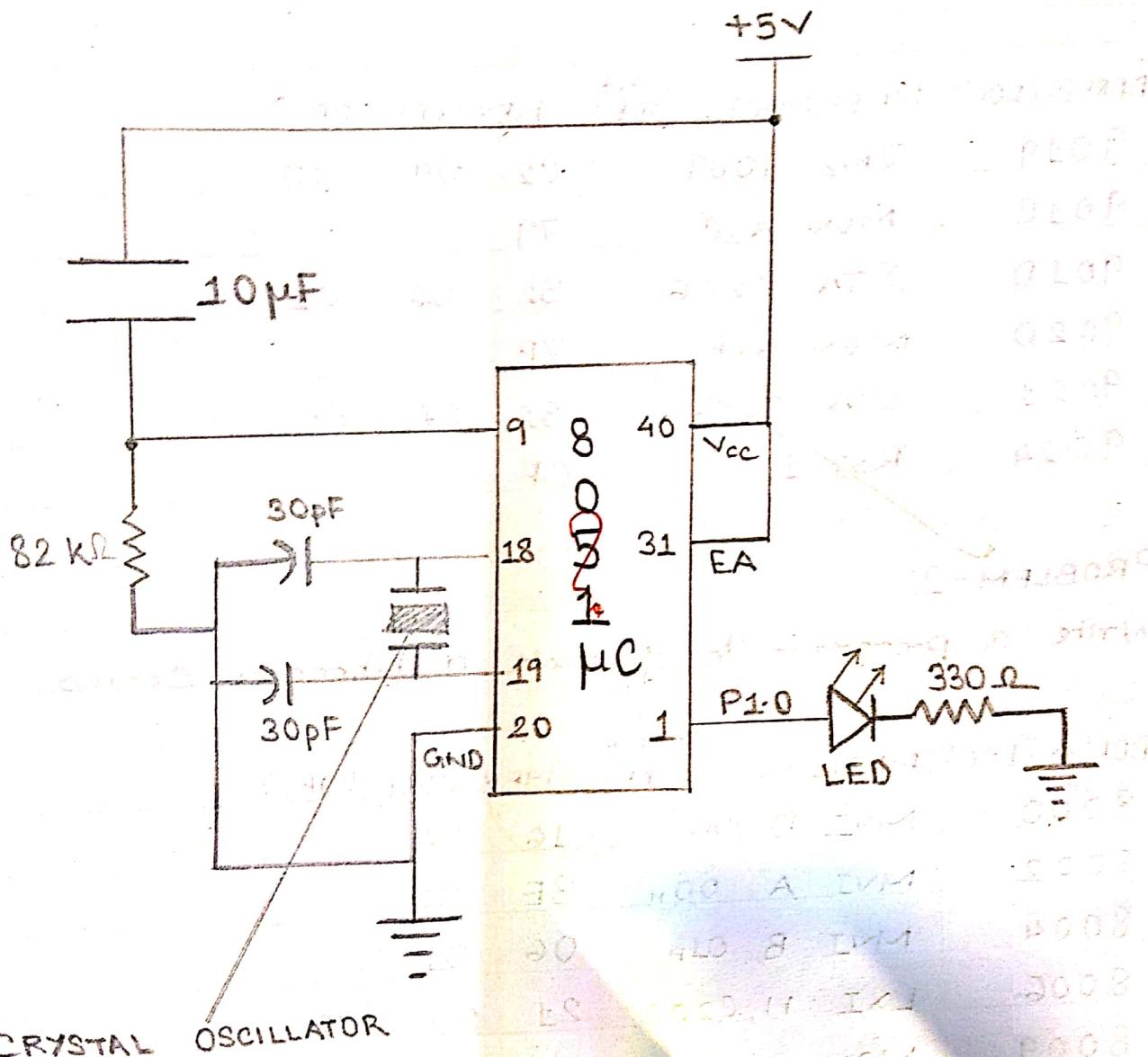


FIG : POWER ON RESET CIRCUIT WITH LED ON P1.0

OUTPUT (REMARKS)

LED is blinking.

DAY - 10

PROBLEM - 1

Grow and blink LED connected on P1.0 of microcontroller 8051.

MICROCGR1.asm

```
ORG 0000H
SJMP MAIN
ORG 0030H
MAIN:    MOV P1, #OFFH
L1:      MOV P1, #OFFH
          ACALL DELAY
          MOV P1, #00H
          ACALL DELAY
          SJMP L1
DELAY:   MOV R1, #OFFH
L2:      MOV R0, #OFFH
L3:      DJNZ R0, L3
          DJNZ R1, L2
          RET
END
```

i/p:-

2000 - 00  
2001 - 01  
2002 - 02  
2003 - 03  
2004 - 04  
2005 - 05  
2006 - 06  
2007 - 07  
2008 - 08  
2009 - 09

O/p:-

3000 - 00

3001 - 01

3002 - 02

3003 - 03

3004 - 04

3005 - 05

3006 - 06

3007 - 07

3008 - 08

3009 - 09

DETAILED

DETAILED

AGENCY REPORT

## DAY - 11

Familiarization of 8086A microprocessor kit / simulator and assembly language programming using 8086A microprocessor / simulator for :

### PROBLEM - 1

Shifting a block of data from one memory location to another.

| SEGMENT ADDRESS | OFFSET ADDRESS | (SOURCE MNEMONICS CODE)   |
|-----------------|----------------|---------------------------|
| 0000            | 1000           | MOV SI, 2000 <sub>H</sub> |
| 0000            | 1003           | MOV DI, 3000 <sub>H</sub> |
| 0000            | 1006           | MOV CX, 000A <sub>H</sub> |
| 0000            | 1009           | REPNZ MOVS B              |
| 0000            | 100B           | INT 3                     |

i/p :-

|           |           |
|-----------|-----------|
| 2000 - 00 | 3000 - 12 |
| 2001 - 01 | 3001 - 11 |
| 2002 - 02 | 3002 - 10 |
| 2003 - 03 | 3003 - 09 |
| 2004 - 04 | 3004 - 08 |
| 2005 - 05 | 3005 - 07 |
| 2006 - 06 | 3006 - 06 |
| 2007 - 07 | 3007 - 05 |
| 2008 - 08 | 3008 - 04 |
| 2009 - 09 | 3009 - 03 |

o/p :-

Bx = 00FF (ie- String didn't match)

## PROBLEM-2

## String matching