

BCD to BINARY CONVERSION

Problem Statement :-

A BCD number between 0 & 99 is stored in an R/W memory location. Called the input Buffer (INBUF). Write a main program and Conversion Subroutine (BCDBIN) to convert the BCD number into its equivalent binary number. Store the result in a memory location defined as the Output Buffer (OUTBUF)

Program

```
START: LXI SP, STACK ; Initialize stack pointer
      LXI H, INBUF ; Point HL index to the
                  ; input Buffer memory location
                  ; where BCD number is stored.
      LXI B, OUTBUF ; Point BC index to the Output
                  ; Buffer memory where Binary
                  ; numbers will be stored.
      MOV A, M ; Get BCD number
      CALL BCDBIN ; Call BCD to binary conversion
                  ; routine.
      STAX B ; Store Binary number in the
            ; output Buffer.
      HLT ; End of program
```

```
BCDBIN : PUSH B ; Save BC registers
        PUSH D ; Save DE registers.
        MOV B, A ; Save BCD number
        ANI 0FH ; Mask most significant
                ; four bits.
```

Example: Assume BCD no is 72:

A 0111 0010 → 72₁₀

B 0111 0010 → 72₁₀

A 0000 0010 → 02₁₀

MOV C, A ; Save unpacked BCD₁ in C C 0000 0010 → 02₁₆

MOV A, B ; Get BCD again A 0111 0010 → 72₁₆

ANI F0H ; Mask least significant four bits A 0111 0000 → 70₁₆

JZ BCD₁ ; if BCD₂ = 0, the result is only BCD₂

RR C ; Convert most significant four

RR C ; bits into unpacked BCD₂

RRC

RRC

MOV D, A ; Save BCD₂ in D

XRA A ; Clear accumulator

MVI E, 0AH ; Set E as multiplier of 10

SUM: ADD E ; Add 10 until (D) = 0

DCR D ; Reduce BCD₂ by one.

JNZ SUM ; Is multiplication complete?
; if not, go back & add again

BCD1: ADD C ; ADD BCD₁

POP D ; Retrieve previous contents

POP B

RET

A 0000 0111 → 07₁₆

D 0000 0111 → 07₁₆

E 0000 1010 → 0AH
Add E as many times as (D)

After adding E seven times A contains:

0100 0110

C +0000 0010

A 0000 1000 → 48H