

MIT Practicals
Assignment 7
Krunal Rank (U18CO081)

Question 1: Write a program to convert a given number of binary data bytes into their BCD equivalents, and store them as unpacked BCDs in the output buffer. The number of data bytes is specified in register D in the main program. The converted numbers should be stored in groups of three consecutive memory locations. If the number is not large enough to occupy all three locations, Zeros should be loaded in those locations.

```
1
2 ;<Question 1>
3
4 jmp start
5
6 ;data
7
8
9 ;code
10 start: LXI D,5000h
11        PUSH D
12        LXI H,2000h
13 LOOP:  MVI D,00H
14        XRA A
15        MOV C, M
16 loop:  ADI 01H
17        DAA
18        JNC skip
19        INR D
20 skip:  DCR C
21        JNZ loop
22        MOV B,A
23        MVI C,04
24        ANI 00F0h
25 L1:    RRC
26        DCR C
27        JNZ L1
28        POP D
29        XCHG
```

```

30      MOV M,A
31      MOV A,B
32      ANI 0Fh
33      INX H
34      MOV M,A
35      INX H
36      PUSH H
37      XCHG
38      HLT
39

```

dle

| Address (Hex) | Address | Data |
|---------------|---------|------|
| 2000 | 8192 | 68 |

| Address (Hex) | Address | Data |
|---------------|---------|------|
| 5000 | 20480 | 6 |
| 5001 | 20481 | 8 |

Question 2: A set of ten BCD readings is stored in the Input Buffer. Convert the numbers into binary and add the numbers. Store the sum in the Output Buffer, the sum can be larger than FFH.

Registers

| | |
|--------|-------|
| A | 38 |
| BC | 00 FE |
| DE | 20 0A |
| HL | 00 38 |
| PSW | 00 00 |
| PC | 42 2C |
| SP | FF FF |
| nt-Reg | 00 |

Flags

| | |
|----|---|
| S | 0 |
| Z | 1 |
| AC | 0 |
| P | 1 |
| C | 0 |

Assembly Code Editor

Load memory at address:

```

9 ;code
10 start: lxi h,2000h
11         mvi b,0ah
12         mvi d,00h
13         mvi e,00h
14 repeat: push d
15         inx h
16         mov a,m|
17         mov d,a
18         ani 0fh
19         mov e,a
20         mov a,d
21         ani 0f0h
22         rrc
23         mov d,a
24         rrc
25         rrc
26         add d
27         add e
28         pop d
29         add e
30         mov e,a
31         jnc done
32         inr d
33 done:   dcr b
34         jnz repeat
35         xchg
36         shld 1fffh
37         hlt
          
```

Decimal - Hex Conversion

| | |
|----------------------|----------------------|
| Decimal | Hex |
| <input type="text"/> | <input type="text"/> |
| > To Hex | < To Dec |

I/O Ports

Port Value: Operation:

Memory

Address: Value: Operation:

Execution Results

| Start Address | End Address | Data |
|---------------|-------------|------|
| 1FFF | 8191 | 56 |
| 2000 | 8192 | 0 |
| 2001 | 8193 | 20 |
| 2002 | 8194 | 20 |
| 2003 | 8195 | 20 |
| 2004 | 8196 | 20 |
| 2005 | 8197 | 0 |
| 2006 | 8198 | 0 |
| 2007 | 8199 | 0 |
| 2008 | 8200 | 0 |
| 2009 | 8201 | 0 |

Assembler Message Log

| | |
|---------|--------------------------------|
| Line No | Message |
| 0 | Program assembled successfully |

Question 3: A set of ASCII Hex digits is stored in the Input Buffer memory. Write a program to convert these numbers into binary. Add these numbers in binary, and store the result in the Output-Buffer memory.

Registers

| | |
|---------|-------|
| A | 1C |
| BC | 00 1C |
| DE | 00 00 |
| HL | 20 05 |
| PSW | 00 00 |
| PC | 42 24 |
| SP | FF FF |
| Int-Reg | 00 |

Flag

| | |
|----|---|
| S | 0 |
| Z | 1 |
| AC | 0 |
| P | 1 |
| C | 0 |

Decimal - Hex Conversion

Decimal

Hex

0

0

> To Hex

< To Dec

I/O Ports

0

-

+

05

Update Port Value

Memory

0

-

+

39

Update Memory

Load me at

```

1
2 ;<Question 3>
3
4 jmp start
5
6 ;data
7
8
9 ;code
10 start: lxi h,2000h
11         mov b,m
12         mvi c,00h
13 repeat: inx h
14         mov a,m
15         cpi 39h
16         jc digit
17         sbi 41h
18         adi 0ah
19         jmp sum
20 digit:  sui 30h
21 sum:    add c
22         mov c,a
23         dcr b
24         jnz repeat
25         mov a,c
26         sta 1fffh
27         hlt

```

Data Stack KeyPad Memor

Start 1ffff

| Address (Hex) | Address | Data |
|---------------|---------|------|
| 1FFF | 8191 | 28 |
| 2000 | 8192 | 5 |
| 2001 | 8193 | 49 |
| 2002 | 8194 | 48 |
| 2003 | 8195 | 65 |
| 2004 | 8196 | 67 |
| 2005 | 8197 | 53 |
| 2006 | 8198 | 0 |
| 2007 | 8199 | 0 |
| 2008 | 8200 | 0 |
| 2009 | 8201 | 0 |

Line No

Assembler Message

0

Program assembled successfully