

Assignment 08 | MFP

CE-092

Assignment submission for Microprocessor Fundamentals and Programming subject week 8.

nevilparmar24@gmail.com

Task 1:

Write an assembly language to convert the code of a 4 digit BCD number to hexadecimal and vice versa.

Code:

```
; BCD to HEX
data segment
    bcd db '1234'
    hex_num dw 0
    mult_factor dw 1000
    digit_count dw 4
data ends

code segment
    assume cs:code,ds:data
start:
    mov ax,data
    mov ds,ax
    mov bx , 10
    mov cx , digit_count
    lea si , bcd
up:
    mov al , [si]
```

```

    and ax , 000fh
    mul mult_factor
    add hex_num , ax
    mov ax , mult_factor
    mov dx , 00
    div bx
    mov mult_factor , ax
    inc si
    loop up

    mov cx , hex_num

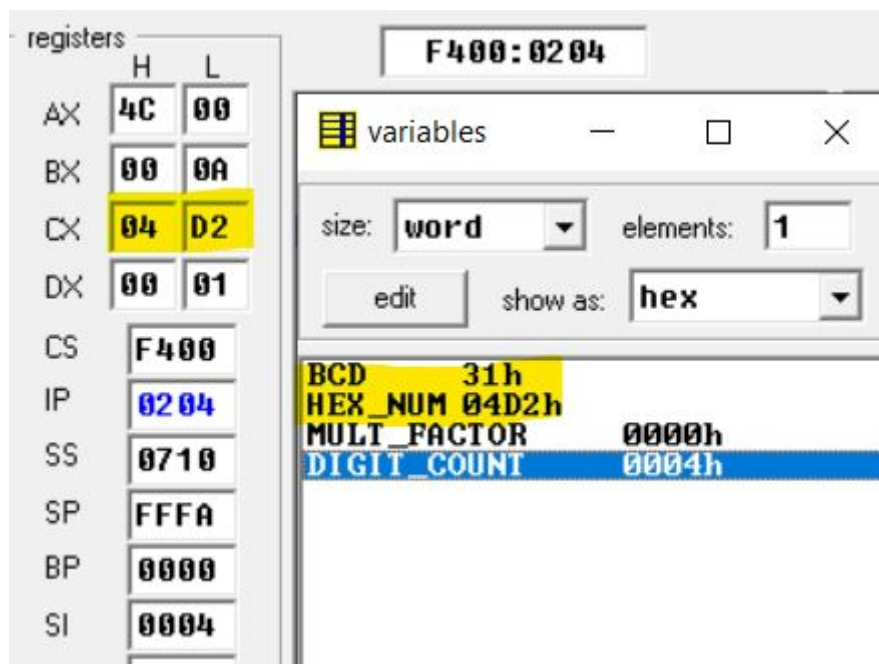
    mov ah , 4ch
    int 21h

```

code *ends*

end start

Output:



Task 1-2:

HEX to BCD

Code:

```
; HEX to BCD
data segment
    hex db 0afh
    bcd dw 0
    cnt db 0
data ends

code segment
    assume cs:code,ds:data
start:
    mov ax,data
    mov ds,ax

    mov al,hex
    cmp al,00
    jz exit

loop1:
    mov ah,00
    mov bl,0ah
    div bl
    mov dh,00
    mov dl,ah
    mov bl,al
    mov al,04
    mul cnt
    mov cl,al
    rol dx,cl
```

```

        or bcd,dx
        mov al,bl
        inc cnt
        cmp al,0
        jnz loop1
exit:
        mov bx,bcd
        mov ax,4c00h
        int 21h

code ends
end start

```

Output:

The screenshot shows a debugger interface with the following components:

- Registers Panel:**

	H	L
AX	4C	00
BX	01	75
CX	00	08
DX	01	00
CS	F400	
IP	0204	
SS	0710	
SP	FFFA	
BP	0000	
SI	0000	
DI	0000	
DS	0710	
ES	0700	
- Variables Panel:**
 - size: **byte** elements: **1**
 - edit show as: **hex**
 - Variables:

Variable	Value
HEX	00AFh
BCD	0175h
CNT	03h
- Memory Panel:**
 - Address: **F400:0204**
 - Content:

Address	Value	Comment
F4213:	00 000	NULL
F4214:	00 000	NULL
F4215:	00 000	NULL
- Disassembly Panel:**
 - Address: **F400:0204**
 - Instruction: **RET**
 - Comment: **NT 021h**
- Buttons:** screen, source, reset, aux, vars, debug, stack, flags

Task 2:

Write an assembly language to convert ASCII to 7 segment display.

Code:

```
data segment
    ascii db '1'
    lookup db
3fh,06h,5bh,4fh,66h,6dh,7dh,07h,7fh,6fh
    result db ?
data ends

code segment
    assume cs:code,ds:data
    mov ax,data
    mov ds,ax
    mov bx, offset lookup
    mov al,ascii
    and al,0fh
    xlat
    mov result,al
    mov ax,4c00h
    int 21h
code ends
end
```

Output:

```
-d
072A:0180  00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
072A:0190  00 00 00 00 00 00 E9 8C-55 FC 8C C8 8E D8 8E D0 .....U.....
072A:01A0  31 3F 06 5B 4F 66 6D 7D-07 7F 6F 00 00 00 00 00 1?.[Ofm}..o.....
072A:01B0  B8 44 07 8E D8 BB 01 00-A0 00 00 24 0F D7 A2 0B .D.....$. ....
072A:01C0  00 B8 00 4C CD 21 01 C3-E8 E0 05 26 F6 06 04 00 ...L.!.....&....
072A:01D0  04 74 0C 26 80 26 01 00-F8 26 80 0E 01 00 04 26 .t.&&...&....&
072A:01E0  A1 16 00 99 26 A3 0A 00-26 89 16 0C 00 26 88 16 ...&...&...&..
072A:01F0  24 00 33 C0 26 A3 14 00-26 A3 16 00 26 80 26 05 $.3.&...&...&.&.
```

Task 3:

Write an assembly program to convert an 8 bit BCD number to corresponding octal number.

Code:

```
prnstr macro msg
    mov ah, 09h
    mov dx, offset msg
    int 21h
endm

data segment
    buf1 db "Enter a BCD number : $"
    buf2 db 0ah, "Invalid BCD Number...$"
    buf3 db 0ah, "Equivalent octal number is : $"
    buf4 db 6
        db 0
        db 6 dup(0)
    multiplier db 0ah
data ends

code segment
    assume cs:code, ds:data
start :
```

```

    mov ax, data
    mov ds, ax
    mov es, ax

    prnstr buf1

    mov ah, 0ah
    lea dx, buf4
    int 21h

    mov si, offset buf4 + 2
    mov cl, byte ptr [si-1]
    mov ch, 00h
subtract :
    mov al, byte ptr [si]
    cmp al, 30h
    jnb cont1
    prnstr buf2
    jmp stop
cont1 :
    cmp al, 3ah
    jb cont2
    prnstr buf2
    jmp stop
cont2 :
    sub al, 30h
    mov byte ptr [si], al

    inc si
    loop subtract

    mov si, offset buf4 + 2

```

```

        mov cl, byte ptr [si-1]
        mov ch, 00h
        mov ax, 0000h

calc :
        mul multiplier
        mov bl, byte ptr [si]
        mov bh, 00h
        add ax, bx
        inc si
        loop calc

        mov si, offset buf4 + 2
        mov bx, ax
        mov dx, 0000h
        mov ax, 8000h

convert :
        mov cx, 0000h

conv :
        cmp bx, ax
        jb cont3
        sub bx, ax
        inc cx
        jmp conv

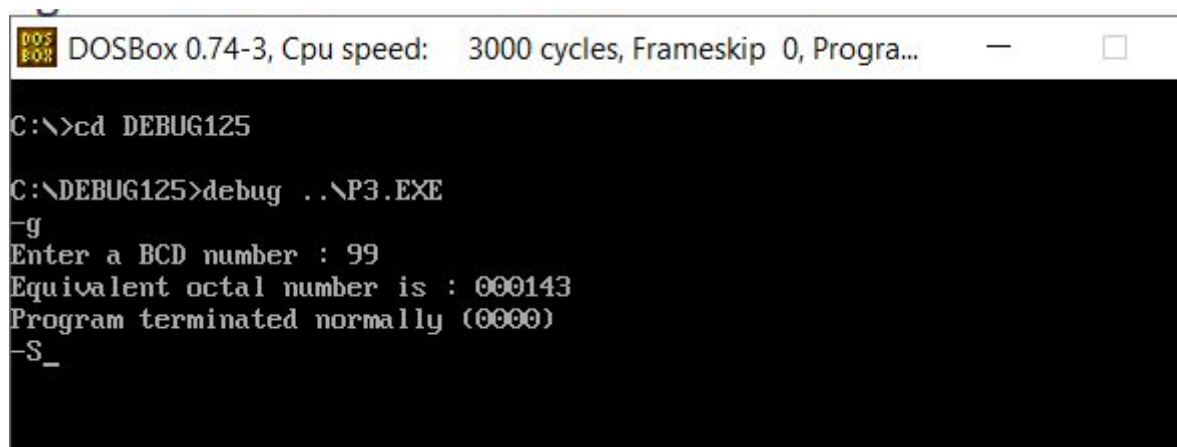
cont3 :
        add cl, 30h
        mov byte ptr [si], cl
        inc si
        mov cx, 0008h
        div cx
        cmp ax, 0000h
        jnz convert

```



```
        mov byte ptr [si], '$'
        prnstr buf3
        prnstr buf4+2
stop :
        mov ax, 4c00h
        int 21h
code ends
end start
```

Output:



```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
C:\>cd DEBUG125
C:\DEBUG125>debug ..\P3.EXE
-g
Enter a BCD number : 99
Equivalent octal number is : 000143
Program terminated normally (0000)
-S_
```

Nevil Parmar
CE-092
<https://nevilparmar.me>