

Exp No: 4  
Date: 19/9/2020

## Code Conversion

Name: Gokhulnath T  
Register Number: 185001051

**Aim:** To write assembly program to do following Code Conversion:

- a) BCD to Hexadecimal
- b) Hexadecimal to BCD

**Procedure:**

1. Install all the required file for executing MASM programs.(Masm, edit, link, debug etc..).
2. Write the assembly program in any editor before mounting the folder to the MASM.
3. Mount the folder that contains the assembly program with any name such as "d".
  - mount d e:\masm
4. Create the object file of the assembly program using masm.
  - masm 16BITADD.asm
5. Use the link to create the executable file of the object file created from the above step.
  - Link 16BITADD.obj
6. Run the executable file using debug.
  - debug 16BITADD.exe
7. By un-assembling the program you can check the code segment of the program
  - u 076b:0100
8. To check the data memory segment, you can use the memory option to view the data stored.
  - d 076a:0000
9. To enter your own values, you can use the enter option which will prompt for new values.
  - e 076a:0000
10. To execute the program, you can use go option
  - G
11. After successful execution and termination of the program, you can check the result by checking the data memory segment
  - d 076a:0000
12. The result can be viewed in the respective address mentioned in the program.

#### 4 a) BCD to Hexadecimal

##### Algorithm:

- a) Assign data to ax register
- b) Load contents of memory location ax in register ds
- c) Load contents of memory location bcd in register al
- d) Load contents of memory location bcd in register bl
- e) Load contents 04h in register cl
- f) Shift bits to right of contents in cl and store it in al
- g) Load contents of memory location al in one
- h) Shift bits to left of contents in cl and store it in bl
- i) Shift bits to right of contents in cl and store it in bl
- j) Load contents of memory location bl in two
- k) Load contents 00h in register ah
- l) Load contents 0Ah in register dl
- m) Multiply dl => ax/bl
- n) Add al => al+bl
- o) Load contents of memory location al in register hex
- p) Load content 4ch termination code to ah register
- q) Stops execution of the program

##### Program:

Program	Comments
assume cs:code,ds:data,es:extra	Initializing the code, data and extra segments to assembler
data segment	Data segment
bcd db 59h	bcd is declared and initialized to 59h
one db 00h	one is declared and initialized to 00h
two db 00h	two is declared and initialized to 00h
hex db 00h	hex is declared and initialized to 00h
data ends	
code segment	Code segment
org 0100h	assemble the code starting from address range 0100h
start: mov ax,data	Transferring the data from memory location data to ax
mov ds,ax	Transferring the data from memory location ax to ds

mov al,bcd	Transferring the data from memory location bcd to al
mov bl,bcd	Transferring the data from memory location bcd to bl
mov cl,04h	Transferring the 04h to cl
shr al,cl	Shift bits to right of contents in cl and store it in al
mov one,al	Transferring the data from memory location al to one
shl bl,cl	Shift bits to left of contents in cl and store it in bl
shr bl,cl	Shift bits to right of contents in cl and store it in bl
mov two,bl	Transferring the data from memory location bl to two
mov ah,00h	Transferring the 00h to ah
mov dl,0Ah	Transferring the 0Ah to dl
mul dl	Multiply dl => ax/bl
add al,bl	Add al => al+bl
mov hex,al	Transferring the data from memory location al to hex
mov ah,4ch	Transferring the termination code 4ch to ah
int 21h	Termination
code ends	Code ends
end start	

### Unassembled code:

```
DOS BOX DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG
D:\>debug BCDTOHEX.EXE
-u
076B:0064 B86A07      MOV     AX,076A
076B:0067 8ED8      MOV     DS,AX
076B:0069 A00000     MOV     AL,[0000]
076B:006C 8A1E0000   MOV     BL,[0000]
076B:0070 B104      MOV     CL,04
076B:0072 D2E8      SHR     AL,CL
076B:0074 A20100     MOV     [0001],AL
076B:0077 D2E3      SHL     BL,CL
076B:0079 D2EB      SHR     BL,CL
076B:007B 8B1E0200   MOV     [0002],BL
076B:007F B400      MOV     AH,00
076B:0081 B20A      MOV     DL,0A
076B:0083 F6E2      MUL     DL
-
```

### Sample Input and output:

```
-d 076a:0000
076A:0000  59 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  Y.....
076A:0010  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  .....
076A:0020  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  .....
076A:0030  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  .....
076A:0040  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  .....
076A:0050  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  .....
076A:0060  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  .....
076A:0070  00 00 00 00 B8 6A 07 8E-D8 A0 00 00 8A 1E 00 00  .....j.....
-g
Program terminated normally
-d 076a:0000
076A:0000  59 05 09 3B 00 00 00 00-00 00 00 00 00 00 00 00  Y...;.....
076A:0010  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  .....
076A:0020  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  .....
076A:0030  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  .....
076A:0040  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  .....
076A:0050  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  .....
076A:0060  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  .....
076A:0070  00 00 00 00 B8 6A 07 8E-D8 A0 00 00 8A 1E 00 00  .....j.....
-
```

### Result:

Thus, the assembly program for BCD to Hexadecimal conversion is written and executed.

#### 4 b) Hexadecimal to BCD

##### Algorithm:

- a) Assign data to ax register
- b) Load contents of memory location ax in register ds
- c) Load contents 00h in register ah
- d) Load contents of memory location input in register al
- e) Load contents 64h in register bl
- f) Div bl => ax/bl
- g) Load contents of memory location al in out1
- h) Load contents of memory location ah in al
- i) Load contents 00h in register ah
- j) Load contents 0Ah in register bh
- k) Div bl => ax/bl
- l) Load contents 04h in register cl
- m) Shift bits to left of contents in cl and store it in bl
- n) Add al => al+ah
- o) Load contents of memory location al in register out2
- p) Load content 4ch termination code to ah register
- q) Stops execution of the program

##### Program:

Program	Comments
assume cs:code,ds:data,es:extra	Initializing the code, data and extra segments to assembler
data segment	Data segment
input db 0AAh	input is declared and initialized to 0AAh
out1 db 00h	Out1 is declared and initialized to 00h
Out2 db 00h	Out2 is declared and initialized to 00h
data ends	
code segment	Code segment
org 0100h	assemble the code starting from address range 0100h
start: mov ax,data	Transferring the data from memory location data to ax
mov ds,ax	Transferring the data from memory location ax to ds
mov al,input	Transferring the data from memory location input to al
mov bl,64h	Transferring the data 64h to bl

div bl	Div bl => ax/bl
mov out1,al	Transferring the data from memory location al to out1
mov al,ah	Transferring the data from memory location ah to al
mov ah,00h	Transferring the data 00h to ah
mov bh,0Ah	Div bl => ax/bl
div bh	Div bl => ax/bl
mov cl,04h	Transferring the data 04h to cl
shl al,cl	Shift bits to left of contents in cl and store it in al
add al,ah	Add al => al+ah
mov out2,al	Transferring the data from memory location al to out2
mov ah,4ch	Transferring the termination code 4ch to ah
int 21h	Termination
code ends	Code ends
end start	

## Unassembled code:

```
DOS
BOX DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG
E:\>debug HEXTOBCD.EXE
-u
076B:0100 B86A07      MOV     AX,076A
076B:0103 8ED8        MOV     DS,AX
076B:0105 B400        MOV     AH,00
076B:0107 A00000      MOV     AL,[0000]
076B:010A B364        MOV     BL,64
076B:010C F6F3        DIV     BL
076B:010E A20100      MOV     [0001],AL
076B:0111 8AC4        MOV     AL,AH
076B:0113 B400        MOV     AH,00
076B:0115 B70A        MOV     BH,0A
076B:0117 F6F7        DIV     BH
076B:0119 B104        MOV     CL,04
076B:011B D2E0      SHL     AL,CL
076B:011D 02C4      ADD     AL,AH
076B:011F A20200      MOV     [0002],AL
-
```

## Sample Input and output:

```
-d 076a:0000
076A:0000  AA 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  .....
076A:0010  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  .....
076A:0020  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  .....
076A:0030  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  .....
076A:0040  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  .....
076A:0050  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  .....
076A:0060  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  .....
076A:0070  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  .....
-g
Program terminated normally
-d 076a:0000
076A:0000  AA 01 70 00 00 00 00 00-00 00 00 00 00 00 00 00  ..p.....
076A:0010  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  .....
076A:0020  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  .....
076A:0030  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  .....
076A:0040  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  .....
076A:0050  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  .....
076A:0060  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  .....
076A:0070  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  .....
```

## Result:

Thus, the assembly program for Hexadecimal to BCD conversion is written and executed.