

# Code Conversions

**Expt No:** 4

**Date :** 17/09/2020

**Name:** Mahesh Bharadwaj K

**Reg No:** 185001089

---

## **Aim:**

To write and execute 8086 programs for code conversion from BCD to Hexadecimal and vice-versa.

## **Procedure:**

- Mount masm folder to a drive on DOSBOX.
- Navigate to mounted drive using 'dir' .
- Save 8086 program with the extension '**.asm**' in the same folder using the command '**edit**'.
- Assemble the **.asm** file using the command '**masm filename.asm**'.
- Link the assembled **.obj** file using the command '**link filename.obj**'.
- Debug the executable file **.exe** with the '**debug filename.exe**' command.
  - i. **U**: To view the un-assembled code.
  - ii. **D**: Used as 'D segment:offset' to see the content of memory locations starting from segment:offset address.
  - iii. **E**: To change the values in memory.
  - iv. **G**: Execute the program using command.
  - v. **Q** exits from the debug session.

## **Algorithm:**

### **1. BCD to Hexadecimal Conversion**

- \* Move the data segment address to the AX register and then move it to the DS register.
- \* Move the bcd value into BL register.
- \* Perform bitwise and operation with 0Fh to get lower order bits and with F0h to get higher order bits
- \* Shift higher order bits 4 time to the right to obtain the digit
- \* Multiply Higher order bits by 0Ah(hex value for 10) and add Lower order bits
- \* Store the Result.

## 2. Hexadecimal to BCD Conversion

- \* Move the data segment address to the AX register and then MOVE it to the DS register.
  - \* Move the hex value into AL register & Set AH to be 00h
  - \* Move 64 into BL register (decimal 100)
  - \* Divide AX by BL
  - \* Stored the quotient (hundreds digit) at the higher order byte of result.
  - \* Move the remainder from AH register to AL register
  - \* Move 0Ah into BL register
  - \* Divide AX by BL
  - \* Move remainder obtained(units digit) into DL register.
  - \* The quotient represents the 10s digit.
  - \* Left shift AL four times and add remainder in DL register.
  - \* Store as lower order byte of result.
- 

## 1. BCD to Hexadecimal Conversion

**Program:**

Program	Comments
<b>start:</b> MOV AX,data	Move data segment address contents to AX register
MOV ds,AX	Move data in AX register to DS register
MOV BL, bcd	Move the BCD value into BL register
AND BL, 0Fh	bitwise AND with 0Fh to obtain units digit
MOV AL, bcd	Move the BCD value into AL register
AND AL, 0F0h	Bitwise AND with F0h to obtain tens digit
MOV CL, 04h	Amount of bits to shift
SHR AL, CL	Shift AL register by 4 bits
MOV DL, 0Ah	
MUL DL	Multiply AL with 0Ah
ADD AL, BL	Add units digit to Al
MOV result, AL	Storing final result in AL register
MOV ah,4ch	
int 21h	Request interrupt routine

Unassembled Code:

```
D:\>debug 4-A.EXE
-U
076B:0100 B86A07      MOV     AX,076A
076B:0103 8ED8        MOV     DS,AX
076B:0105 8A1E0000      MOV     BL,[0000]
076B:0109 80E30F      AND     BL,0F
076B:010C A00000      MOV     AL,[0000]
076B:010F 24F0        AND     AL,F0
076B:0111 B104        MOV     CL,04
076B:0113 D2E8        SHR     AL,CL
076B:0115 B20A        MOV     DL,0A
076B:0117 F6E2        MUL     DL
076B:0119 02C3      ADD     AL,BL
076B:011B A20100      MOV     [0001],AL
076B:011E B44C        MOV     AH,4C
```

Input and Output:

Figure 1: **Input:** bcd = 12 & **Output:** result(hexadecimal) = 0C

```
-d 076A:0000
076A:0000 12 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 12 0C 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 .....
```

## Hexadecimal to BCD Conversion

Program:

Program	Comments
start: MOV AX,data	Move data segment address contents to AX register
MOV ds,AX	Move data in AX register to DS register
MOV AH, 00h	Load 00h onto AH register
MOV AL, hex	Load Hex value into AL register
MOV BL, 64h	
DIV BL	Divide AX by BL(decimal value 100)
MOV result.h, AL	Store quotient in AL register in higher order byte of result word
MOV AL, AH	Load Remainder of division into AL register
MOV AH, 00h	Clear AH register for division
MOV BL, 0Ah	
DIV BL	Divide remainder of previous division by 0Ah(decimal 10) To get tens digit and units digit
MOV DL, AH	Load remainder into DL register
MOV CL, 04h	
SHL AL, CL	Left shift quotient by 4 bits
ADD AL, DL	Add shifted quotient and remainder
MOV result.l, AL	Store the result obtained in lower order byte of result word
MOV ah,4ch	
int 21h	Request interrupt routine

Unassembled Code:

```
D:\>debug 4-B.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 A20200          MOV     [0002],AL
076B:0111 8AC4             MOV     AL,AH
076B:0113 B400             MOV     AH,00
076B:0115 B30A             MOV     BL,0A
076B:0117 F6F3             DIV     BL
076B:0119 8AD4             MOV     DL,AH
076B:011B B104             MOV     CL,04
076B:011D D2E0             SHL     AL,CL
076B:011F 02C2             ADD     AL,DL
```

## Input and Output:

Figure 2: **Input:** hex = FFh & **Output:** bcd(result) = 02 55

```
-d 076A:0000
076A:0000  FF 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 .....
076A:0020  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 .....
076A:0040  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 .....
076A:0060  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 .....
-g

Program terminated normally
-d 076A:0000
076A:0000  FF 55 02 00 00 00 00 00-00 00 00 00 00 00 00 .U.....
076A:0010  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 .....
076A:0030  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 .....
076A:0050  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 .....
076A:0070  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
```

## Result:

8086 ASL programs for Code conversion from hexadecimal to BCD and Vice-Versa have been executed successfully using MS - DOSBox.