Code Conversion

Expt No:	4	**Name:**	Shivanirudh S G
Date :	18/09/2020	**Reg No:**	185001146

Aim:

To perform code conversion in 8086.

BCD to Hexadecimal

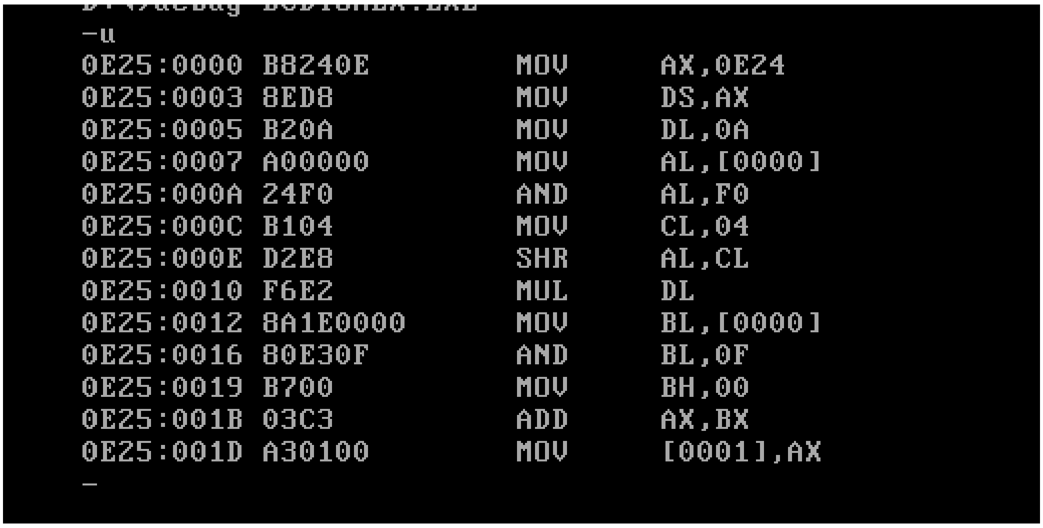
Algorithm:

- Move the data segment to the AX register and then move it to the DS register.
- Move hexadecimal value 0A to DL register, 04 to CL register, and 00 to BH register.
- Move BCD value to AL register.
- Perform AND operation with AND AL, F0H to mask the lower byte.
- Shift AL register 4 bits to right by SHR AL, CL.
- Multiply AL with DL register with MUL DL.
- Move BCD value to BL register.
- Perform AND operation with AND BL, 0FH to mask the upper byte.
- Add value in BX register to AX register using ADD AX, BX.
- Store result in HEX from AX register.

Program:

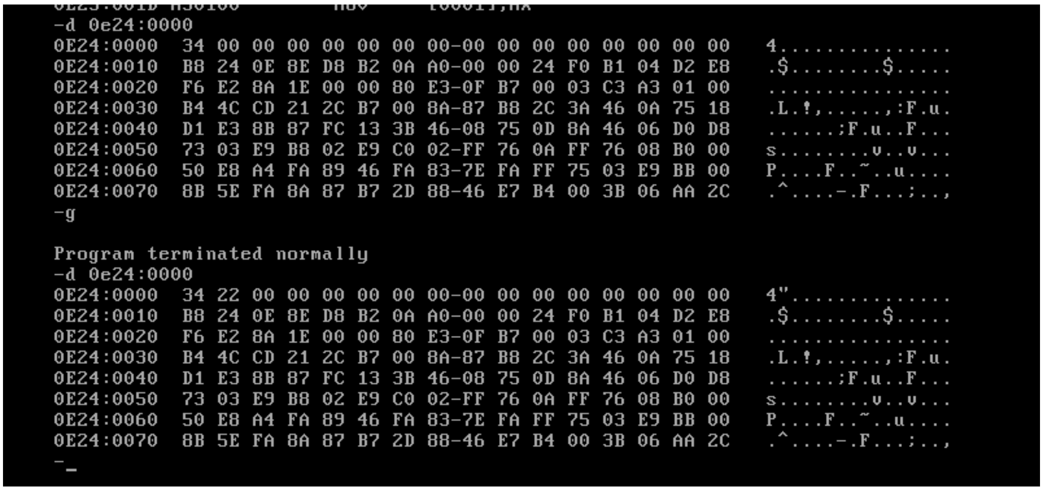
Program	**Comments**
assume cs:code, ds:data	Declare code and data segments
data segment	Start of data segment
bcd db 34H	Define byte bcd with value 34
hex dw 0000H	Define word hex with hex value 0000
data ends	End of data segment
code segment	Start of code segment
start: mov ax, data	Move data to AX register
mov ds, ax	Move contents of AX register to DS register
mov dl, 0AH	Move hex value 0A to DL register
mov al, bcd	Move contents of bcd to AL register
and al, 0F0H	Perform AL = Al & F0
mov cl, 04H	Move hex value 04 to CL register
shr al, cl	Shift AL register right by 04 bits
mul dl	Perform AX = AL x DL
mov bl, bcd	Move contents of bcd to BL register
and bl, 0F0H	Perform BL = Bl & 0F
mov bh, 00H	Move hex value 00 to CL register
add ax, bx	Perform AX = AX + BX
mov hex, ax	Move contents of AX register to HEX
int 21h	Request interrupt routine
code ends	End of code segment
end start	

Unassembled code:



```
D:\>debug BCDTOHEX.EXE
-u
0E25:0000 B8240E        MOV     AX,0E24
0E25:0003 8ED8          MOV     DS,AX
0E25:0005 B20A          MOV     DL,0A
0E25:0007 A00000        MOV     AL,[0000]
0E25:000A 24F0          AND     AL,F0
0E25:000C B104          MOV     CL,04
0E25:000E D2E8          SHR     AL,CL
0E25:0010 F6E2          MUL     DL
0E25:0012 8A1E0000      MOV     BL,[0000]
0E25:0016 80E30F        AND     BL,0F
0E25:0019 B700          MOV     BH,00
0E25:001B 03C3          ADD     AX,BX
0E25:001D A30100        MOV     [0001],AX
-
```

Input and Output:



```
0E25:001D A30100        MOV     [0001],AX
-d 0e24:0000
0E24:0000  34 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   4...............
0E24:0010  B8 24 0E 8E D8 B2 0A A0-00 00 24 F0 B1 04 D2 E8   .$........$.....
0E24:0020  F6 E2 8A 1E 00 00 80 E3-0F B7 00 03 C3 A3 01 00   ................
0E24:0030  B4 4C CD 21 2C B7 00 8A-87 B8 2C 3A 46 0A 75 18   .L.!,.....,:F.u.
0E24:0040  D1 E3 8B 87 FC 13 3B 46-08 75 0D 8A 46 06 D0 D8   ......;F.u..F...
0E24:0050  73 03 E9 B8 02 E9 C0 02-FF 76 0A FF 76 08 B0 00   s........v..v...
0E24:0060  50 E8 A4 FA 89 46 FA 83-7E FA FF 75 03 E9 BB 00   P....F..~..u....
0E24:0070  8B 5E FA 8A 87 B7 2D 88-46 E7 B4 00 3B 06 AA 2C   .^....-.F...;..,
-g

Program terminated normally
-d 0e24:0000
0E24:0000  34 22 00 00 00 00 00 00-00 00 00 00 00 00 00 00   4"..............
0E24:0010  B8 24 0E 8E D8 B2 0A A0-00 00 24 F0 B1 04 D2 E8   .$........$.....
0E24:0020  F6 E2 8A 1E 00 00 80 E3-0F B7 00 03 C3 A3 01 00   ................
0E24:0030  B4 4C CD 21 2C B7 00 8A-87 B8 2C 3A 46 0A 75 18   .L.!,.....,:F.u.
0E24:0040  D1 E3 8B 87 FC 13 3B 46-08 75 0D 8A 46 06 D0 D8   ......;F.u..F...
0E24:0050  73 03 E9 B8 02 E9 C0 02-FF 76 0A FF 76 08 B0 00   s........v..v...
0E24:0060  50 E8 A4 FA 89 46 FA 83-7E FA FF 75 03 E9 BB 00   P....F..~..u....
0E24:0070  8B 5E FA 8A 87 B7 2D 88-46 E7 B4 00 3B 06 AA 2C   .^....-.F...;..,
-_
```

Figure 1: **Input:** *BCD:* 34 ; **Output:** *Hexadecimal:* 22H

Hexadecimal to BCD

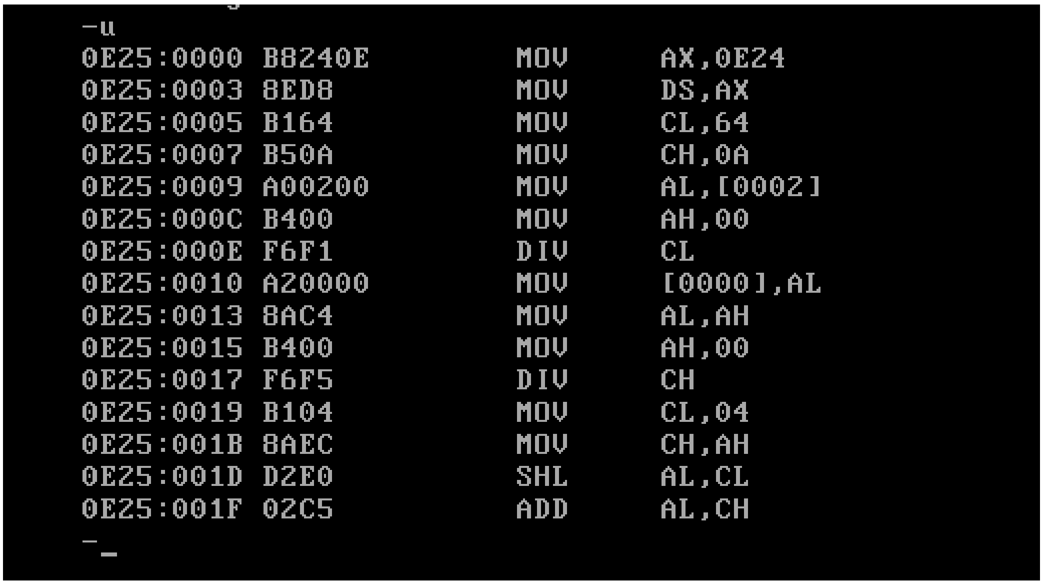
Algorithm:

- Move the data segment to the AX register and then move it to the DS register.
- Move hexadecimal value 0A to CH register, and 64 to CL register.
- Move HEX value to AL register, 00H to AH register.
- Divide AX by Cl using DIV CL.
- Store quotient from AL in BCDU.
- Move value of AH to AL register, and 00H to AH register.
- Divide AX by CH using DIV CH.
- Move 04H to the CL register.
- Move value of AH register to CH register.
- Shift AL register 4 bits to left by SHL AL, CL.
- Add CH to AL register using ADD AL, CH.
- Move contents of AL register to BCDL.

Program:

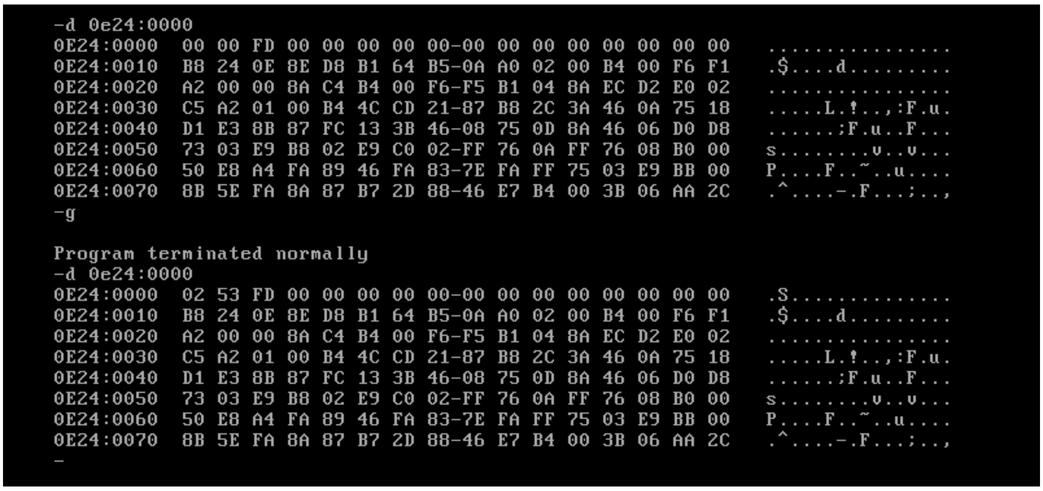
Program	**Comments**
assume cs:code, ds:data	Declare code and data segments
data segment	Start of data segment
bcd$_u db 00H$	Define byte bcd$_u with value 00$
bcd$_l db 00H$	Define byte bcd$_l with value 00$
hex dw 0FDH	Define byte hex with hex value FD
data ends	End of data segment
code segment	Start of code segment
start: mov ax, data	Move data to AX register
mov ds, ax	Move contents of AX register to DS register
mov cl, 64H	Move hex value 64 to CL register
mov ch, 0AH	Move hex value 0A to CH register
mov al, hex	Move contents of bcd to AL register
mov ah, 00H	Move hex value 00 to AH register
div cl	Perform AL = AX / CL
mov bcdu, al	Move contents of AL register to bcdu
mov al, ah	Move contents of AH register to AL
mov ah, 00H	Move hex value 00 to AH register
div ch	Perform AL = AX / CH
mov cl, 04H	Move hex value 04 to CL register
mov ch, ah	Move contents of AH to CH register
shl al, cl	Shift AL register left by 04 bits
add al, ch	Perform AL = AL + CH
mov bcdl, al	Move contents of AL to bcdl register
int 21h	Request interrupt routine
code ends	End of code segment
end start	

Unassembled code:



```
-u
0E25:0000 B8240E        MOV     AX,0E24
0E25:0003 8ED8          MOV     DS,AX
0E25:0005 B164          MOV     CL,64
0E25:0007 B50A          MOV     CH,0A
0E25:0009 A00200        MOV     AL,[0002]
0E25:000C B400          MOV     AH,00
0E25:000E F6F1          DIV     CL
0E25:0010 A20000        MOV     [0000],AL
0E25:0013 8AC4          MOV     AL,AH
0E25:0015 B400          MOV     AH,00
0E25:0017 F6F5          DIV     CH
0E25:0019 B104          MOV     CL,04
0E25:001B 8AEC          MOV     CH,AH
0E25:001D D2E0          SHL     AL,CL
0E25:001F 02C5          ADD     AL,CH
-
```

Input and Output:



```
-d 0e24:0000
0E24:0000  00 00 FD 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0E24:0010  B8 24 0E 8E D8 B1 64 B5-0A A0 02 00 B4 00 F6 F1   .$....d.........
0E24:0020  A2 00 00 8A C4 B4 00 F6-F5 B1 04 8A EC D2 E0 02   ................
0E24:0030  C5 A2 01 00 B4 4C CD 21-87 B8 2C 3A 46 0A 75 18   .....L.!..,:F.u.
0E24:0040  D1 E3 8B 87 FC 13 3B 46-08 75 0D 8A 46 06 D0 D8   ......;F.u..F...
0E24:0050  73 03 E9 B8 02 E9 C0 02-FF 76 0A FF 76 08 B0 00   s........v..v...
0E24:0060  50 E8 A4 FA 89 46 FA 83-7E FA FF 75 03 E9 BB 00   P....F..~..u....
0E24:0070  8B 5E FA 8A 87 B7 2D 88-46 E7 B4 00 3B 06 AA 2C   .^....-.F...;..,
-g

Program terminated normally
-d 0e24:0000
0E24:0000  02 53 FD 00 00 00 00 00-00 00 00 00 00 00 00 00   .S..............
0E24:0010  B8 24 0E 8E D8 B1 64 B5-0A A0 02 00 B4 00 F6 F1   .$....d.........
0E24:0020  A2 00 00 8A C4 B4 00 F6-F5 B1 04 8A EC D2 E0 02   ................
0E24:0030  C5 A2 01 00 B4 4C CD 21-87 B8 2C 3A 46 0A 75 18   .....L.!..,:F.u.
0E24:0040  D1 E3 8B 87 FC 13 3B 46-08 75 0D 8A 46 06 D0 D8   ......;F.u..F...
0E24:0050  73 03 E9 B8 02 E9 C0 02-FF 76 0A FF 76 08 B0 00   s........v..v...
0E24:0060  50 E8 A4 FA 89 46 FA 83-7E FA FF 75 03 E9 BB 00   P....F..~..u....
0E24:0070  8B 5E FA 8A 87 B7 2D 88-46 E7 B4 00 3B 06 AA 2C   .^....-.F...;..,
-
```

Figure 2: **Input:** *Hexadecimal:* FDH ; **Output:** *BCD:* 253

Result:

The 8086 programs were written to perform code conversion from BCD to Hexadecimal and vice versa, and the results observed.