SSN College of Engineering
Department of Computer Science and Engineering
UCS1512 – Microprocessors Lab
CODE CONVERSION

Name: Prasanna Kumaran D
Registration Number: 185001110 15 September 2020
Semester: V

1 AIM:

To write and execute 8086 programs for converting Binary coded decimal to and from its Hexadecimal equivalent.

2 PROCEDURE:

- Write the program in a text editor and save it as a ***.asm*** file under the MASM directory.
- Launch DOSBOX application and mount the MASM folder using the command prompt.
- Use the following syntax for mounting : ' **mount [LOCAL DRIVE] FILEPATH** '. Enter into the local drive('LOCAL DRIVE:').
- The code file can be edited using the command **edit FILENAME.asm**. Save the changes and exit.
- Assemble the code using the command '**masm FILENAME.asm**' to generate the object file. The object file is in the format 'FILENAME.obj'
- Add dynamic libraries using the syntax **'link FILENAME.obj'** to generate the executeable(**.exe**) file.
- Enter the debug mode using debug FILENAME.exe to execute and analuse the memory contents. The various commands used in debug mode are as follows:-
  - U :- Displays unassembled code.
  - D :- Refers to the offset from which contents in the memory are displayed.
  - E :- Change the value in memory.
  - G :- execute the code.
  - Q :- Quit debug mode.

3 Algorithm & Program

INITIALIZATION:

- Declare and initialize the operands and the code and data segments.

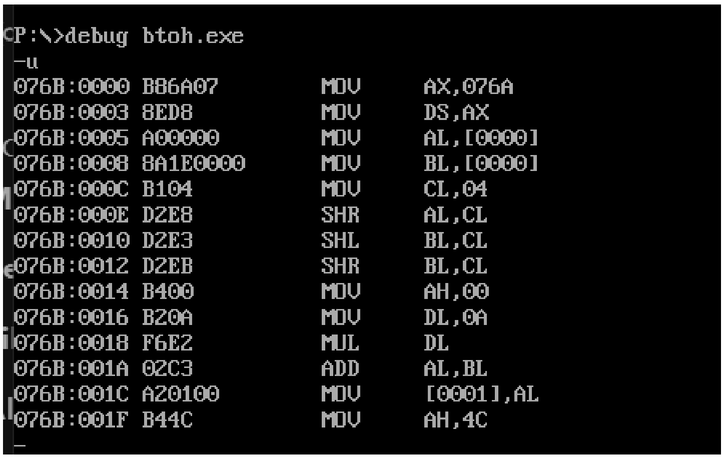
3.1 BCD to Hexadecimal:

To convert Binary coded decimal into its hexadecimal equivalent.

- Initialize data and code segment and variables
- Move the starting address of data segment to DS
- Transfer the contents of BCD to AL and BL, Initialize CL to 4
- Right rotate AL by CL times
- Left rotate BL by CL times followed by right rotating CL times
- Multiply DL
- Add AL and BL
- Transfer the contents of AL to output variable
- Terminate the program

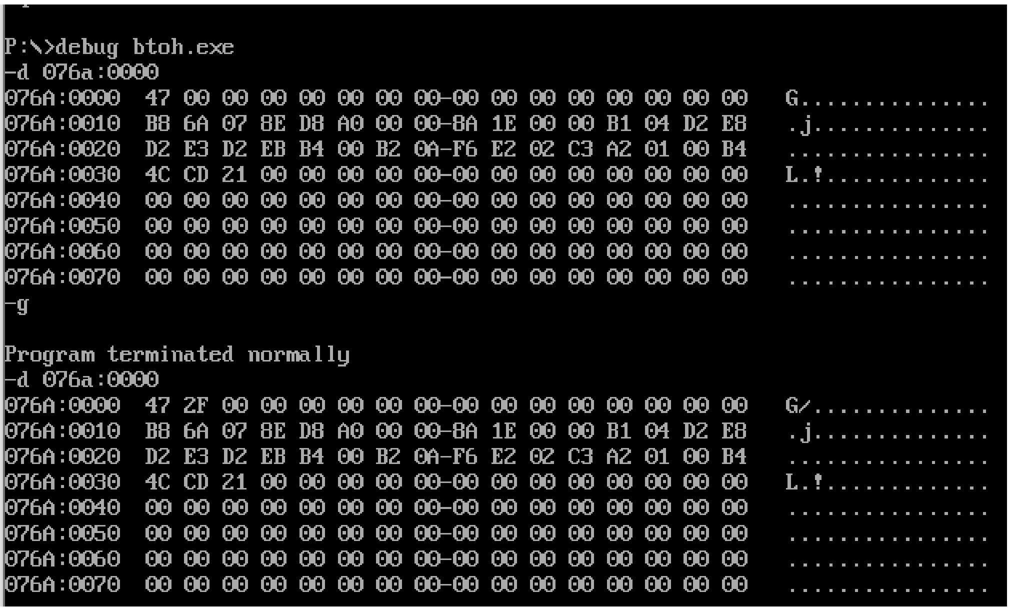
3.1.1 BCD to Hexadecimal: Program

Program	**Comments**
assume cs:code, ds:data	
data segment bcd db 47h hex db 00h data ends	Initialize data segment and variables End data segment
code segment org 0000h start: mov ax, data mov ds, ax mov al, bcd mov bl, bcd mov cl, 04h shr al, cl shl bl, cl shr bl, cl mov ah, 00h mov dl, 0Ah mul dl add al, bl mov hex, al mov ah, 4ch int 21h code ends end start	Initialize code segment Transfer address of data segment to DS Transfer contents of BCD to AL and BL Intialize CL to 4 Right rotate AL by CL times Left rotate BL by CL times Right rotate BL by CL times Intialize AH to 0 and DL to A Multiply DL AL = AL + BL Transfer contents of AL to hex Termination of execution End of code segment Terminate program



```
P:\>debug btoh.exe
-u
076B:0000 B86A07        MOV     AX,076A
076B:0003 8ED8          MOV     DS,AX
076B:0005 A00000        MOV     AL,[0000]
076B:0008 8A1E0000      MOV     BL,[0000]
076B:000C B104          MOV     CL,04
076B:000E D2E8          SHR     AL,CL
076B:0010 D2E3          SHL     BL,CL
076B:0012 D2EB          SHR     BL,CL
076B:0014 B400          MOV     AH,00
076B:0016 B20A          MOV     DL,0A
076B:0018 F6E2          MUL     DL
076B:001A 02C3          ADD     AL,BL
076B:001C A20100        MOV     [0001],AL
076B:001F B44C          MOV     AH,4C
-
```

Figure 1: BCD to hexadecimal - unassembled



```
P:\>debug btoh.exe
-d 076a:0000
076A:0000  47 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   G...............
076A:0010  B8 6A 07 8E D8 A0 00 00-8A 1E 00 00 B1 04 D2 E8   .j..............
076A:0020  D2 E3 D2 EB B4 00 B2 0A-F6 E2 02 C3 A2 01 00 B4   ................
076A:0030  4C CD 21 00 00 00 00 00-00 00 00 00 00 00 00 00   L.!.............
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  47 2F 00 00 00 00 00 00-00 00 00 00 00 00 00 00   G/..............
076A:0010  B8 6A 07 8E D8 A0 00 00-8A 1E 00 00 B1 04 D2 E8   .j..............
076A:0020  D2 E3 D2 EB B4 00 B2 0A-F6 E2 02 C3 A2 01 00 B4   ................
076A:0030  4C CD 21 00 00 00 00 00-00 00 00 00 00 00 00 00   L.!.............
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   ................
```

Figure 2: BCD to Hexadimal - Output

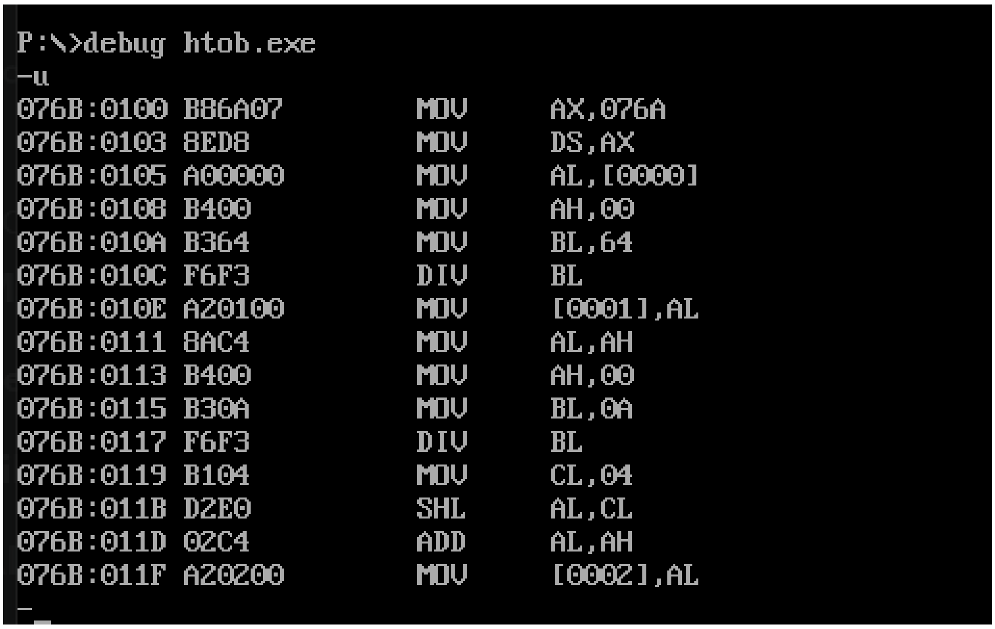
3.2 Hexadecimal to BCD:

To convert Hexadecimal 2 digit number into its BCD equivalent.

- Initialize data and code segment and variables
- Move starting address of data segment to DS and hexadecimal value to AL
- Initialize AH to 0 and BL to 64
- Divide BL and the remainder is stored in AL and store quotient in BCD 1
- reset AH to 0, initialize BL to A and CL to 4
- Shift left AL by CL times
- ADD AL and AH and transfer the result to BCD 2
- Terminate the program

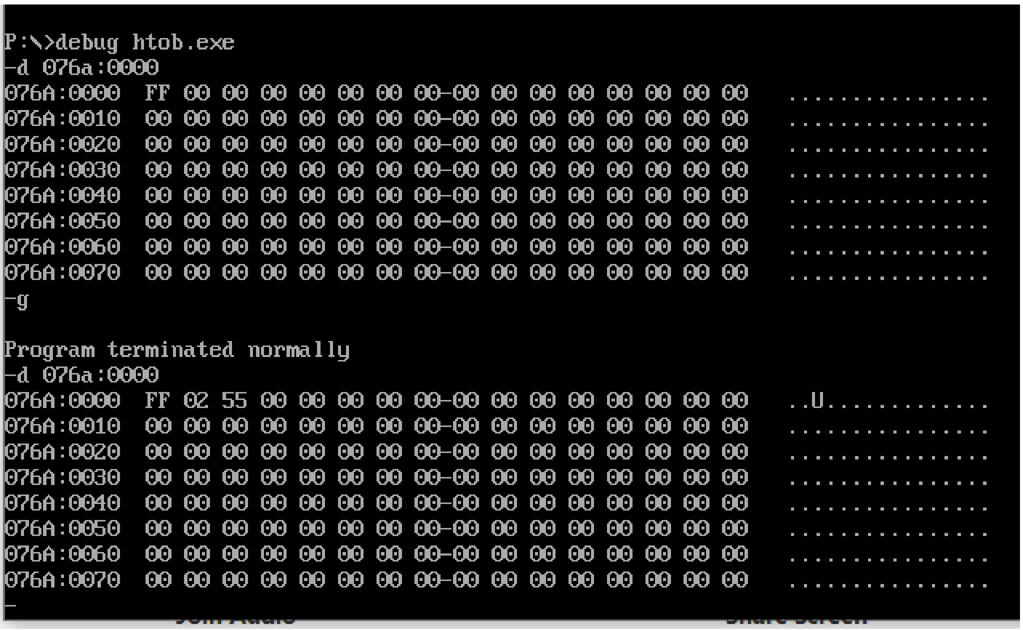
3.2.1 Hexadecimal to BCD: Program

Program	**Comments**
assume cs:code, ds:data	
data segment	Initialize data segment and variables
hex db 0fbh	
bcd1 db 00h	
bcd2 db 00h	
data ends	End data segment
code segment	Initialize code segment
org 0100h	
start:	
mov ax, data	Transfer address of data segment to DS
mov ds, ax	
mov al, hex	Transfer contents of hex to AL
mov ah, 00h	Initialize AH to 0 and BL to 64
mov bl, 64h	Divide BL (divide AL by BL)
div bl	
mov bcd1, al	Transfer contents of AL to bcd1
mov al, ah	Transfer contents of AH to AL
mov ah, 00	Reset AH to 0 and Initialize BL to A
mov bl, 0Ah	Divide BL
div bl	Initialize CL to 4
mov cl, 04h	Left rotate AL by CL times
shl al, cl	Al = AL + AH
add al, ah	Transfer AL to bcd2
mov bcd2, al	
mov ah, 4ch	Termination of execution
int 21h	End of code segment
code ends	
end start	Terminate program



```
P:\>debug htob.exe
-u
076B:0100 B86A07        MOV     AX,076A
076B:0103 8ED8          MOV     DS,AX
076B:0105 A00000        MOV     AL,[0000]
076B:0108 B400          MOV     AH,00
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 B30A          MOV     BL,0A
076B:0117 F6F3          DIV     BL
076B:0119 B104          MOV     CL,04
076B:011B D2E0          SHL     AL,CL
076B:011D 02C4          ADD     AL,AH
076B:011F A20200        MOV     [0002],AL
-_
```

Figure 3: Hexadecimal to BCD - unassembled



```
P:\>debug htob.exe
-d 076a:0000
076A:0000  FF 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  FF 02 55 00 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 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   ................
-
```

Figure 4: Hexadecimal to BCD - Output

4 RESULT:

Thus, 8086 programs for converting BCD to and from its hexadecimal equivalent have been executed successfully using MS - DOSBox.