

UCS1512 – Microprocessors Lab

Code Conversion

Exp no : 4

Name: Sreedhar V

Date : 18-09-2020

Reg no: 185001161

AIM:

To program and execute the code conversion of BCD to Hexadecimal and vice versa in 8086 using an emulator.

BCD to Hexadecimal:

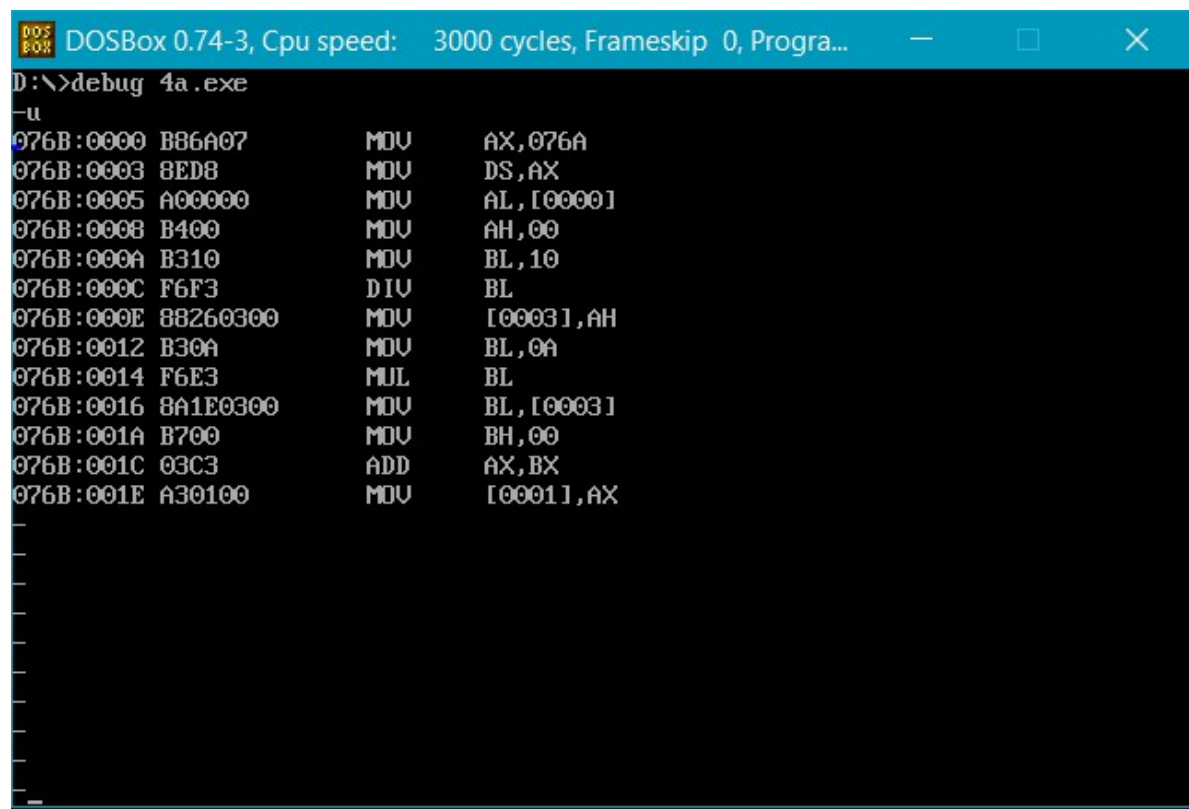
Algorithm:

- Program is set to run from any specified memory position.
- Move the address of data segment to register DS .
- Move the BCD value to AX register.
- Move 10h to BL register and divide BL to separate the each value and store the results.
- Move 0Ah to BL register and multiply BL for conversion(as quotient is already in AL).
- Move the remainder to BX register .
- Add AX register and BX register to get final Hexadecimal value.
- Terminate the program.

Program:

CODE	COMMENT
<p>Program for BCD to Hexadecimal:</p> <p>assume code: cs,ds:data data segment decimal db 12h hexa dw 0000h rem db 00h data ends code segment org 0100h start : mov ax,data mov ds,ax mov al,decimal mov ah,00h mov bl,10h div bl mov rem,ah mov bl,0ah mul bl mov bl,rem mov bh,00h add ax,bx mov hexa,ax mov ah,4ch int 21h code ends ends start</p>	<p>Data segment is initialized decimal is initialized and set to 12h.</p> <p>Code segment begins Originating address is set to 0100h Address of the data is transferred to AX , from AX transferred to DS.</p> <p>Move the decimal to AL Extend AX by moving 00h to AH Move 10h to BL register</p> <p>Divide by BL</p> <p>Move the AH to rem to store remainder Move 0ah to BL register</p> <p>Multiply BL</p> <p>Move rem to BL Extend BX by moving 00h to BH</p> <p>Add AX and BX registers.</p> <p>Store the result Program terminates</p>

Unassembled code:



```
DOS BOX DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
D:\>debug 4a.exe
-u
076B:0000 B86A07      MOV     AX,076A
076B:0003 8ED8      MOV     DS,AX
076B:0005 A00000     MOV     AL,[0000]
076B:0008 B400      MOV     AH,00
076B:000A B310      MOV     BL,10
076B:000C F6F3      DIV     BL
076B:000E 88260300  MOV     [0003],AH
076B:0012 B30A      MOV     BL,0A
076B:0014 F6E3      MUL     BL
076B:0016 8A1E0300  MOV     BL,[0003]
076B:001A B700      MOV     BH,00
076B:001C 03C3      ADD     AX,BX
076B:001E A30100     MOV     [0001],AX
```

Execution:

```

DOS
BOX
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
-d 076a:0000
076A:0000 12 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
076A:0010 B8 6A 07 8E D8 A0 00 00-B4 00 B3 10 F6 F3 88 26 .j.....&
076A:0020 03 00 B3 0A F6 E3 8A 1E-03 00 B7 00 03 C3 A3 01 .....
076A:0030 00 B4 4C CD 21 8B 87 AE-16 3B 46 FE 77 09 89 46 ..L.!....;F.w..F
076A:0040 FE 8A 46 F9 88 46 F8 FE-46 F9 EB C9 8A 5E F8 B7 ..F..F..F....^..
076A:0050 00 8A 87 48 2F D0 D8 73-17 E8 B6 00 8A 5E F8 B7 ...H/..s.....^..
076A:0060 00 8A 87 48 2F D0 D8 73-07 53 B0 01 50 E8 73 01 ...H/..s.S..P.s.
076A:0070 A0 B6 2C 3A 46 F8 74 7E-C7 46 FA 00 00 8A 46 F8 ...,;F.t~.F....F.
-g

Program terminated normally
-d 076a:0000
076A:0000 12 0C 00 02 00 00 00 00-00 00 00 00 00 00 00 00 .....
076A:0010 B8 6A 07 8E D8 A0 00 00-B4 00 B3 10 F6 F3 88 26 .j.....&
076A:0020 03 00 B3 0A F6 E3 8A 1E-03 00 B7 00 03 C3 A3 01 .....
076A:0030 00 B4 4C CD 21 8B 87 AE-16 3B 46 FE 77 09 89 46 ..L.!....;F.w..F
076A:0040 FE 8A 46 F9 88 46 F8 FE-46 F9 EB C9 8A 5E F8 B7 ..F..F..F....^..
076A:0050 00 8A 87 48 2F D0 D8 73-17 E8 B6 00 8A 5E F8 B7 ...H/..s.....^..
076A:0060 00 8A 87 48 2F D0 D8 73-07 53 B0 01 50 E8 73 01 ...H/..s.S..P.s.
076A:0070 A0 B6 2C 3A 46 F8 74 7E-C7 46 FA 00 00 8A 46 F8 ...,;F.t~.F....F.

```

Result:

Conversion of BCD to Hexadecimal is executed and verified using an emulator.

Hexadecimal to BCD:

Algorithm:

- Program is set to run from any specified memory position.
- Move the address of data segment to register DS .
- Move the hexadecimal value to AX register.
- Move 64h to BL register and divide BL and store the results.
- Move 0Ah to BL register and divide BL with remainder of the previous operation.
- Shift left the quotient of the last operation by 4 bits and add remainder to pack into one byte.
- Store the result.
- Terminate the program.

Program:

CODE	COMMENT
Program for Hexadecimal to BCD: assume code: cs,ds:data data segment hexa db 0FFh decimal dw ? data ends code segment org 0100h start : mov ax,data mov ds,ax mov ah,00h mov al,hexa mov bl,64h div bl mov dl,al mov al,ah mov ah,00h mov bl,0ah div bl mov cl,004h rol al,cl mov dh,ah add dh,al mov decimal,dx mov ah,4ch int 21h code ends ends start	 Data segment is initialized hexa is initialized and set to FFh decimal is initialized . Code segment begins Originating address is set to 0100h Address of the data is transferred to AX , from AX transferred to DS. Move hexa to AX with extension of bits. Move 64h to BL register. Divide by BL. Store the quotient in DL. Move remainder from AH to AL. Move 00h to AH for extension. Move 0Ah to BL Divide by BL Shift left by 4 bits in AL register(quotient) Move the Remainder from AH to DH Add DH and AL registers Store the results. Program terminates

Unassembled code:

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
D:\>debug 4b.exe
-u
076B:0000 B86A07      MOV     AX,076A
076B:0003 8ED8             MOV     DS,AX
076B:0005 B400             MOV     AH,00
076B:0007 A00000        MOV     AL,[0000]
076B:000A B364             MOV     BL,64
076B:000C F6F3             DIV     BL
076B:000E 8AD0             MOV     DL,AL
076B:0010 8AC4             MOV     AL,AH
076B:0012 B400             MOV     AH,00
076B:0014 B30A             MOV     BL,0A
076B:0016 F6F3             DIV     BL
076B:0018 B104             MOV     CL,04
076B:001A D2C0             ROL     AL,CL
076B:001C 8AF4             MOV     DH,AH
076B:001E 02F0             ADD     DH,AL
```

Execution:

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
-d 076a:0000
076A:0000 FF 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
076A:0010 B8 6A 07 8E D8 B4 00 A0-00 00 B3 64 F6 F3 8A D0 .j.....d....
076A:0020 8A C4 B4 00 B3 0A F6 F3-B1 04 D2 C0 8A F4 02 F0 .....
076A:0030 89 16 01 00 B4 4C CD 21-16 3B 46 FE 77 09 89 46 ....L.!.;F.w..F
076A:0040 FE 8A 46 F9 88 46 F8 FE-46 F9 EB C9 8A 5E F8 B7 ..F..F..F....^..
076A:0050 00 8A 87 48 2F D0 D8 73-17 E8 B6 00 8A 5E F8 B7 ...H/.s.....^..
076A:0060 00 8A 87 48 2F D0 D8 73-07 53 B0 01 50 E8 73 01 ...H/.s.S..P.s.
076A:0070 A0 B6 2C 3A 46 F8 74 7E-C7 46 FA 00 00 8A 46 F8 ...:F.t~.F....F.
-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 B8 6A 07 8E D8 B4 00 A0-00 00 B3 64 F6 F3 8A D0 .j.....d....
076A:0020 8A C4 B4 00 B3 0A F6 F3-B1 04 D2 C0 8A F4 02 F0 .....
076A:0030 89 16 01 00 B4 4C CD 21-16 3B 46 FE 77 09 89 46 ....L.!.;F.w..F
076A:0040 FE 8A 46 F9 88 46 F8 FE-46 F9 EB C9 8A 5E F8 B7 ..F..F..F....^..
076A:0050 00 8A 87 48 2F D0 D8 73-17 E8 B6 00 8A 5E F8 B7 ...H/.s.....^..
076A:0060 00 8A 87 48 2F D0 D8 73-07 53 B0 01 50 E8 73 01 ...H/.s.S..P.s.
076A:0070 A0 B6 2C 3A 46 F8 74 7E-C7 46 FA 00 00 8A 46 F8 ...:F.t~.F....F.
```

Result:

Conversion of hexadecimal to BCD is executed and verified using an emulator.