Exp No: 7

BCD Addition and Subtraction

Date: 5/10/2020

Name: Gokhulnath T
Register Number: 185001051

Aim: To write assembly program to do following BCD operations:

a) BCD Addition
b) BCD Subtraction

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.

7 a) BCD Addition

Algorithm:

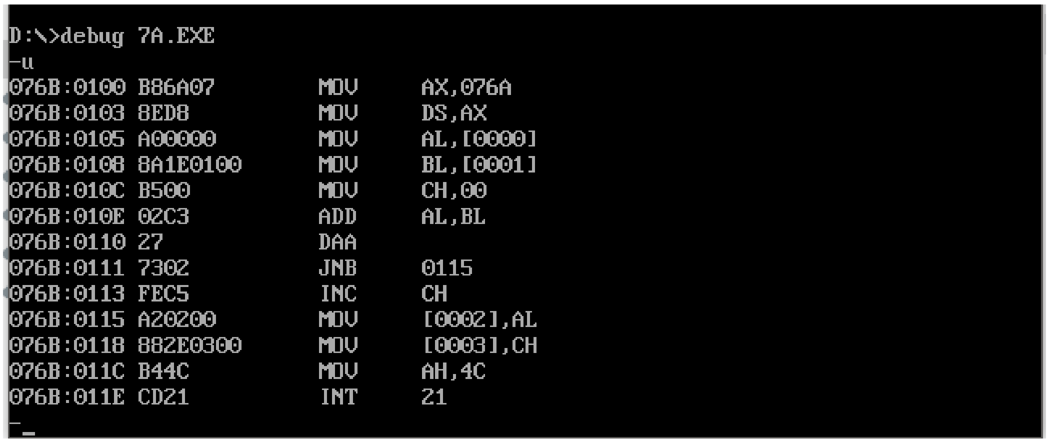
a) Assign data to ax register
b) Load contents of memory location ax in register ds
c) Load contents of memory location opr1 in register al
d) Load contents of memory location opr2 in register bl
e) Load content 00h to register ch
f) Add al and bl store it in al
g) Addition of numbers represented in BCD code
h) Jump to here if no carry else continue
i) Increment ch
j) Here loop starts
k) Load contents of memory location al in result
l) Load contents of memory location ch in register carry
m) Load content 4ch termination code to ah register
n) 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
opr1 db 29h	opr1 is declared and initialized to 29h
opr2 db 95h	Opr2 is declared and initialized to 95h
result db 00H	result is declared and initialized to 00h
carry db 00H	result 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,opr1	Transferring the data from memory location opr1 to al
mov bl,opr2	Transferring the data from memory location opr2 to bl
mov ch,00h	Transferring the data 00h to ch

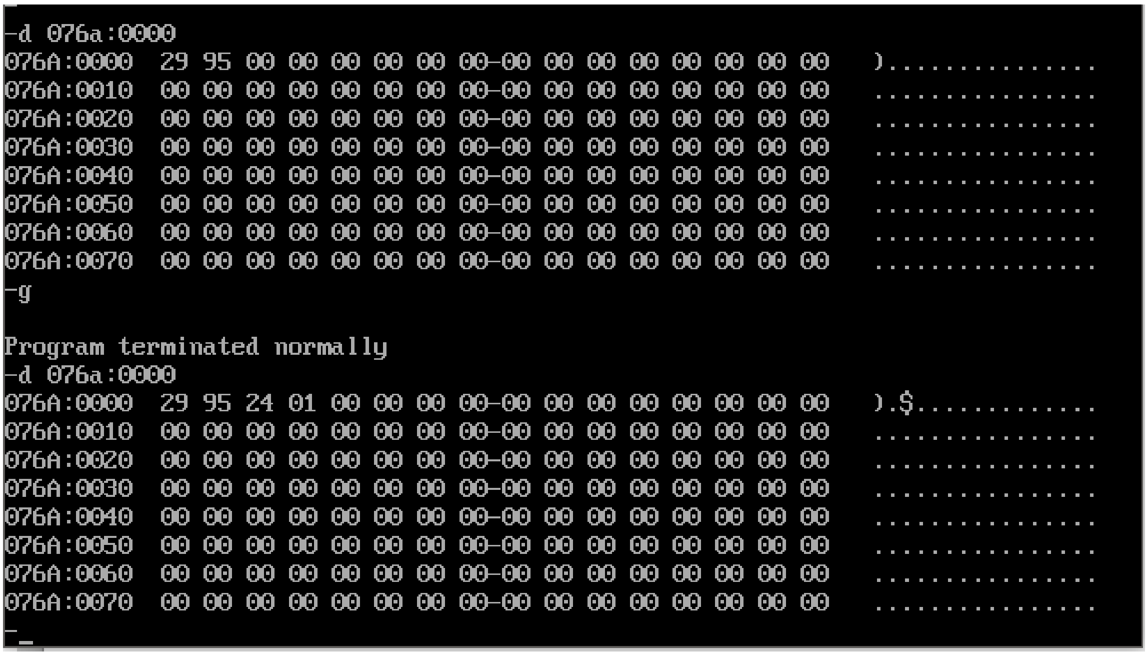
add al,bl	al=>al+bl
daa	Addition of numbers represented in BCD code
jnc here	Jump to here if no carry else continue
inc ch	Increment ch
here: mov result,al	Transferring the data from memory location al to result
mov carry,ch	Transferring the data from memory location ch to carry
mov ah,4ch	Transferring the termination code 4ch to ah
int 21h	Termination
code ends	Code ends
end start	

Unassembled code:



```
D:\>debug 7A.EXE
-u
076B:0100 B86A07        MOV     AX,076A
076B:0103 8ED8          MOV     DS,AX
076B:0105 A00000        MOV     AL,[0000]
076B:0108 8A1E0100      MOV     BL,[0001]
076B:010C B500          MOV     CH,00
076B:010E 02C3          ADD     AL,BL
076B:0110 27            DAA
076B:0111 7302          JNB     0115
076B:0113 FEC5          INC     CH
076B:0115 A20200        MOV     [0002],AL
076B:0118 882E0300      MOV     [0003],CH
076B:011C B44C          MOV     AH,4C
076B:011E CD21          INT     21
-_
```

Sample Input and output:



```
-d 076a:0000
076A:0000  29 95 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  29 95 24 01 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   ................
-_
```

29+95=0124

Result:

Thus, the assembly program for BCD addition is written and executed.

7 b) BCD Subtraction

Algorithm:

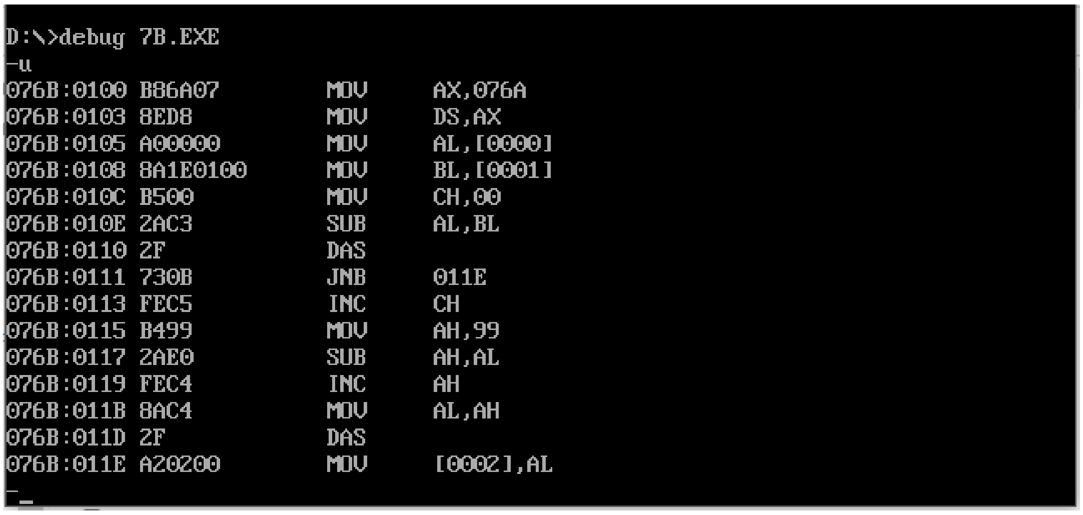
a) Assign data to ax register
b) Load contents of memory location ax in register ds
c) Load contents of memory location opr1 in register al
d) Load contents of memory location opr2 in register bl
e) Load content 00h to register ch
f) Subtract al and bl
g) Adjusts the result of the subtraction of two packed BCD values
h) Jump to here if no carry else continue
i) Increment ch
j) Load content 99h to register ah
k) Subtract ah and al
l) Increment ah
m) Load contents of memory location ah in register al
n) Adjusts the result of the subtraction of two packed BCD values
o) Here loop starts
p) Load contents of memory location al in result
q) Load contents of memory location ch in register sign
r) Load content 4ch termination code to ah register
s) 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
opr1 db 55h	opr1 is declared and initialized to 55h
opr2 db 22h	Opr2 is declared and initialized to22h
result db 00H	result is declared and initialized to 00h
sign db 00H	sign 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

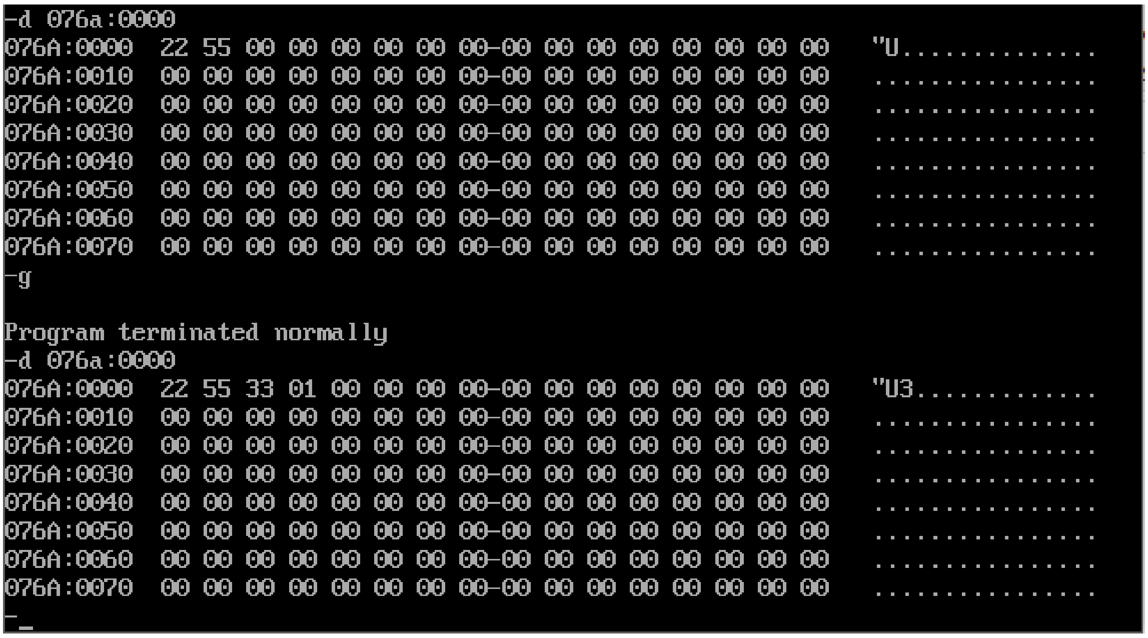
mov ds,ax	Transferring the data from memory location ax to ds
mov al,opr1	Transferring the data from memory location opr1 to al
mov bl,opr2	Transferring the data from memory location opr2 to bl
mov ch,00h	Transferring the data 00h to ch
sub al,bl	al-bl
das	Adjusts the result of the subtraction of two packed BCD values
jnc here	Jump to here if no carry else continue
inc ch	Increment ch
mov ah, 99h	Transferring the data 99h to ah
sub ah, al	ah-al
inc ah	Increment ah
mov al, ah	Transferring the data from memory location ah to al
das	Adjusts the result of the subtraction of two packed BCD values
here: mov result,al	Transferring the data from memory location al to result
mov sign,ch	Transferring the data from memory location ch to sign
mov ah,4ch	Transferring the termination code 4ch to ah
int 21h	Termination
code ends	Code ends
end start	

Unassembled code:



```
D:\>debug 7B.EXE
-u
076B:0100 B86A07        MOV     AX,076A
076B:0103 8ED8          MOV     DS,AX
076B:0105 A00000        MOV     AL,[0000]
076B:0108 8A1E0100      MOV     BL,[0001]
076B:010C B500          MOV     CH,00
076B:010E 2AC3          SUB     AL,BL
076B:0110 2F            DAS
076B:0111 730B          JNB     011E
076B:0113 FEC5          INC     CH
076B:0115 B499          MOV     AH,99
076B:0117 2AE0          SUB     AH,AL
076B:0119 FEC4          INC     AH
076B:011B 8AC4          MOV     AL,AH
076B:011D 2F            DAS
076B:011E A20200        MOV     [0002],AL
-_
```

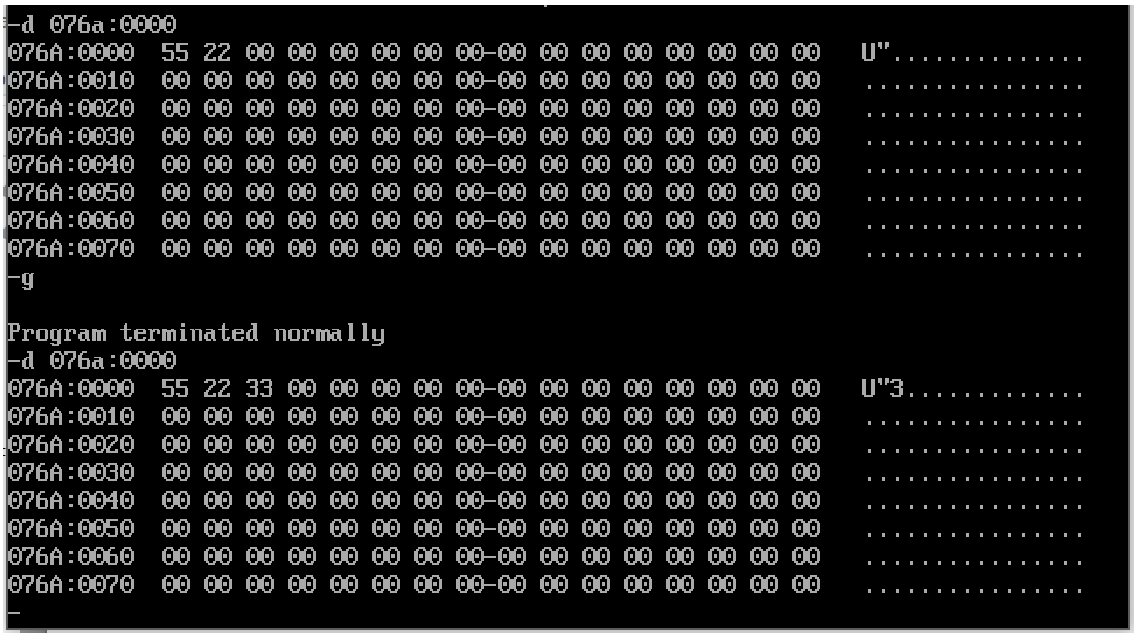
Sample Input and output:



```
-d 076a:0000
076A:0000  22 55 00 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   ................
-g

Program terminated normally
-d 076a:0000
076A:0000  22 55 33 01 00 00 00 00-00 00 00 00 00 00 00 00   "U3.............
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   ................
-_
```

22-55=33 (1 indicates negative)



```
-d 076a:0000
076A:0000  55 22 00 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   ................
-g

Program terminated normally
-d 076a:0000
076A:0000  55 22 33 00 00 00 00 00-00 00 00 00 00 00 00 00   U"3.............
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   ................
-
```

55-22=33 (0 indicates positive)

Result:

Thus, the assembly program for BCD subtraction is written and executed.