

Experiment No. 7: BCD Addition and Subtraction

Date: 05-10-2020

NAME: Harshini S

REG.NO: 185001058

A. AIM:

Program for performing addition of two 8-bit BCD numbers.

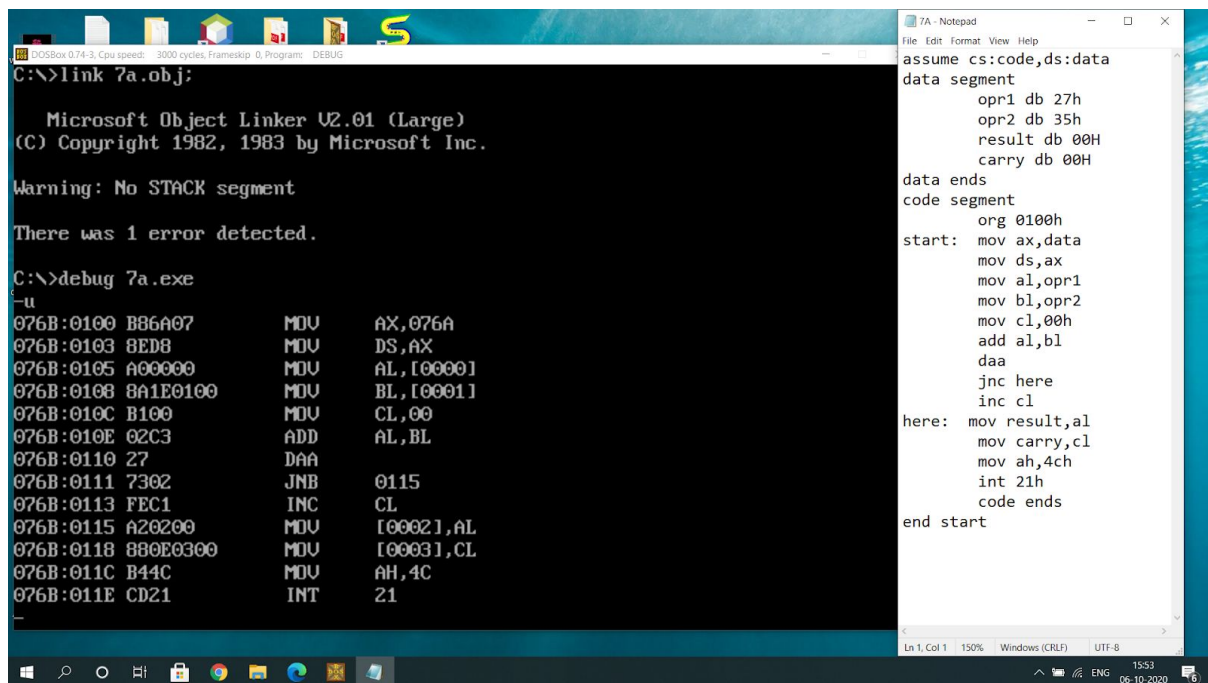
ALGORITHM:

- Initialize the data segment
- Move data segment address to ds
- Load opr1 to al and opr2 to bl
- Load 00h to cl register for carry
- Add al and bl
- Execute daa instruction to adjust the result of the addition of two packed BCD values to create a packed BCD result
- If there is no carry being generated, goto here segment else, increment cl by 1
- In here segment,
 - Load al to result
 - Load cl to carry
 - Terminate the program

PROGRAM:

PROGRAM	COMMENTS
Start: mov ax,data mov ds,ax mov al,opr1 mov bl,opr2 mov cl,00h add al,bl daa jnc here inc cl	Transferring address of data segment to ds Value of opr1 is loaded to al Value of opr2 is loaded to bl Initializing the value of cl with 00h al=al+bl Add numbers represented in 8-bit packed BCD code Jump to "here" segment if no carry is generated Increments cl by 1
Here: mov result,al mov carry,cl mov ah,4ch int 21h	Load register value of al to result Load cl value to carry Terminate the program

UNASSEMBLED CODE:



The screenshot shows a DOSBox window on the left and a Notepad window on the right. The DOSBox window displays the output of the linker (link 7a.obj) and the debugger (debug 7a.exe). The linker output shows a warning about no stack segment and one error detected. The debugger output shows the unassembly of the 7a.exe file, listing instructions and their addresses. The Notepad window shows the assembly code for 7a.asm, which includes data and code segments.

```
C:\>link 7a.obj:

Microsoft Object Linker V2.01 (Large)
(C) Copyright 1982, 1983 by Microsoft Inc.

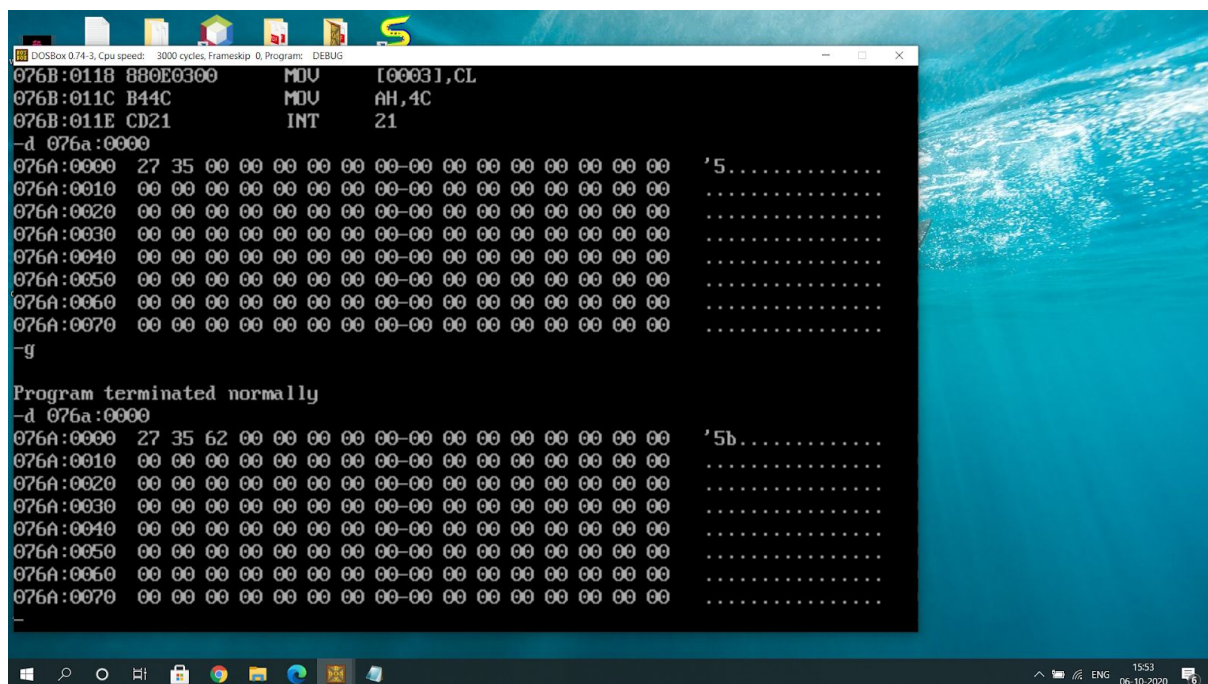
Warning: No STACK segment

There was 1 error detected.

C:\>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 B100        MOV     CL,00
076B:010E 02C3      ADD     AL,BL
076B:0110 27         DAA
076B:0111 7302      JNB     0115
076B:0113 FEC1      INC     CL
076B:0115 A20200      MOV     [0002],AL
076B:0118 880E0300     MOV     [0003],CL
076B:011C B44C      MOV     AH,4C
076B:011E CD21      INT     21

7A - Notepad
File Edit Format View Help
assume cs:code,ds:data
data segment
    opr1 db 27h
    opr2 db 35h
    result db 00H
    carry db 00H
data ends
code segment
    org 0100h
start: mov ax,data
       mov ds,ax
       mov al,opr1
       mov bl,opr2
       mov cl,00h
       add al,bl
       daa
       jnc here
       inc cl
here:  mov result,al
       mov carry,cl
       mov ah,4ch
       int 21h
       code ends
end start
```

SAMPLE INPUT/OUTPUT:



The screenshot shows the DOSBox window displaying the memory dump and program termination. The memory dump shows the contents of memory addresses 076A:0000 to 076A:0070. The program terminated normally, and the memory dump shows the contents of memory addresses 076A:0000 to 076A:0070.

```
076B:0118 880E0300     MOV     [0003],CL
076B:011C B44C      MOV     AH,4C
076B:011E CD21      INT     21
-d 076a:0000
076A:0000 27 35 00 00 00 00 00 00-00 00 00 00 00 00 00 00 00 '5.....
076A:0010 00 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 00 .....
076A:0030 00 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 00 .....
076A:0050 00 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 00 .....
076A:0070 00 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 27 35 62 00 00 00 00 00-00 00 00 00 00 00 00 00 00 '5b.....
076A:0010 00 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 00 .....
076A:0030 00 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 00 .....
076A:0050 00 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 00 .....
076A:0070 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 00 .....
```

RESULT:

Thus addition of two BCD numbers has been performed.

B. AIM:

Program for performing subtraction of two 8-bit BCD numbers.

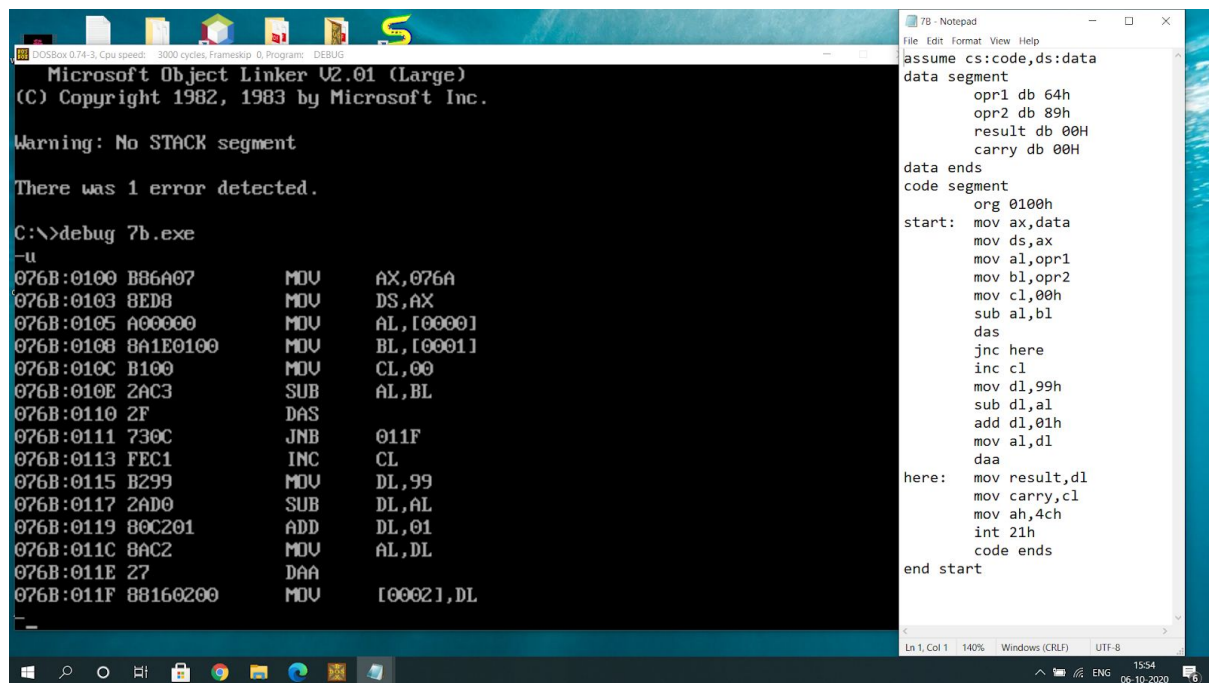
ALGORITHM:

- Initialize the data segment
- Move data segment address to ds
- Load opr1 to al and opr2 to bl
- Load 00h to cl register
- Subtract al and bl
- Execute das instruction to adjust the result of the subtraction of two packed BCD values to create a packed BCD result
- If al is greater than bl, goto here segment else, increment cl by 1 and find the 10's complement of result and decimal adjust it.
- In here segment,
 - Load dl to result
 - Load cl to carry
 - Terminate the program

PROGRAM:

PROGRAM	COMMENTS
mov ax,data mov ds,ax mov al,opr1 mov bl,opr2 mov cl,00h sub al,bl das jnc here inc cl mov dl,99h sub dl,al add dl,01h mov al,dl daa	Load data segment to ds Value of opr1 is loaded to al Value of opr2 is loaded to bl Initializing the value of cl with 00h al=al-bl Subtract numbers represented in 8-bit packed BCD code Jump to "here" segment if al>bl Increment value of cl Load dl with 99h dl=dl-al dl=dl+01h Load al with value of dl Add numbers represented in 8-bit packed BCD code
here: mov result,dl mov carry,cl mov ah,4ch int 21h	Load register value of dl to result Load cl value to carry Terminate the program

UNASSEMBLED CODE:



```
Microsoft Object Linker V2.01 (Large)
(C) Copyright 1982, 1983 by Microsoft Inc.

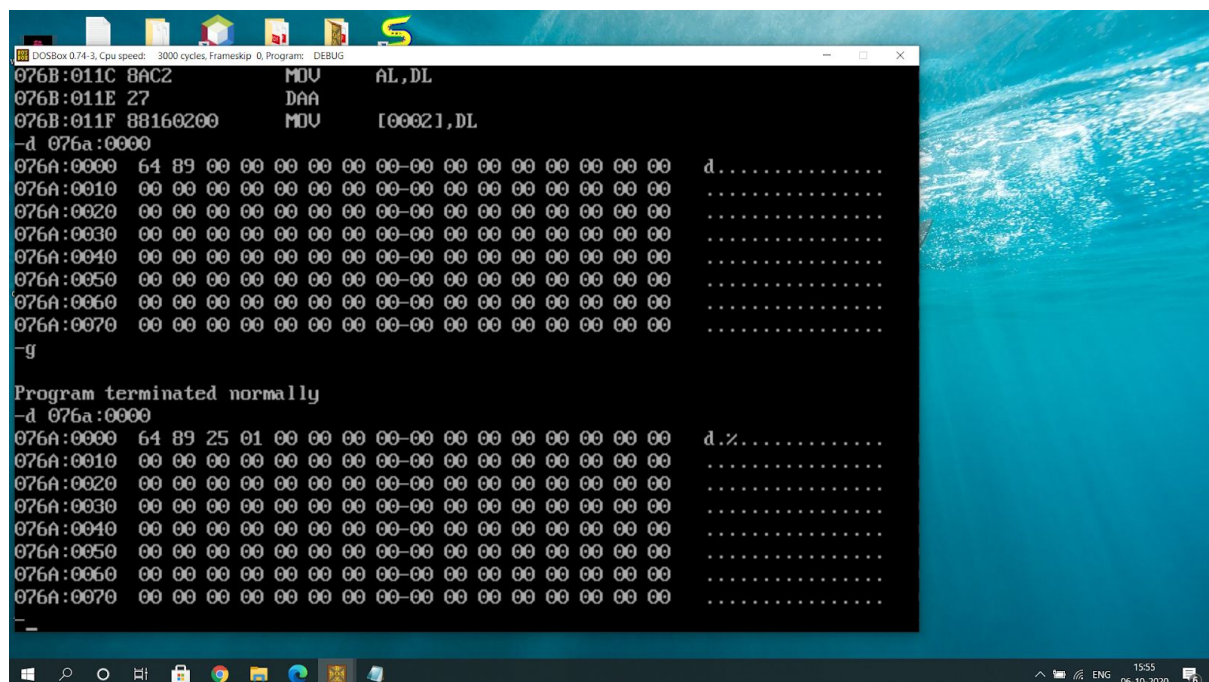
Warning: No STACK segment

There was 1 error detected.

C:\>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 B100        MOV     CL,00
076B:010E 2AC3        SUB     AL,BL
076B:0110 2F          DAS
076B:0111 730C        JNB     011F
076B:0113 FEC1        INC     CL
076B:0115 B299        MOV     DL,99
076B:0117 2AD0        SUB     DL,AL
076B:0119 80C201     ADD     DL,01
076B:011C 8AC2        MOV     AL,DL
076B:011E 27          DAA
076B:011F 88160200     MOV     [0002],DL

7B - Notepad
File Edit Format View Help
assume cs:code,ds:data
data segment
    opr1 db 64h
    opr2 db 89h
    result db 00h
    carry db 00h
data ends
code segment
    org 0100h
start: mov ax,data
       mov ds,ax
       mov al,opr1
       mov bl,opr2
       mov cl,00h
       sub al,bl
       das
       jnc here
       inc cl
       mov dl,99h
       sub dl,al
       add dl,01h
       mov al,dl
       daa
here:  mov result,dl
       mov carry,cl
       mov ah,4ch
       int 21h
       code ends
end start
```

SAMPLE INPUT/OUTPUT



```
076B:011C 8AC2        MOV     AL,DL
076B:011E 27          DAA
076B:011F 88160200     MOV     [0002],DL
-d 076a:0000
076A:0000 64 89 00 00 00 00 00 00 00 00 00 00 00 00 00 00 d.....
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 64 89 25 01 00 00 00 00 00 00 00 00 00 00 00 00 d.%.....
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 .....
```

RESULT:

Thus subtraction of two BCD numbers has been performed.