UCS1512 - MICROPROCESSORS LAB

BCD Addition and Subtraction

Exp. No: 7 Name: Anirudh H

Date: 14/10/2020 Reg. No: 185001019

Aim: To write and execute MASM code to do BCD Addition and BCD Subtraction.

7a. BCD Addition

Algorithm:

- Declare the segments- cs and ds as code and data respectively.
- Initialise the data segment with two operands (opr1 and opr2), result (initialised to 0) and carry (initialised to 0)
- Assemble the code segment and set the origin(starting address) as 0100(in this case).
- Manually transfer the data segment to the register ds through ax.
- Load the value in opr1 and opr2 to al and bl respectively.
- Assign the value 00 to ch, the carry register.
- Add the contents of al and bl and store the result in al.
- Convert the sum in al to BCD using daa command.
- If there is a carry, increment ch register.
- If there is no carry, jump to the here label, and transfer the answer in al to result, and the carry value in ch to carry register.
- Interrupt the program and terminate.

Program	Comments
assume cs:code,ds:data	Declare segments cs and ds
data segment	
opr1 db 80h	Initialise opr1 to 80h
opr2 db 80h	Initialise opr2 to 80h
result db 00H	Initialise result to 00h
carry db 00H	Initialise carry to 00h
data ends	
code segment	
org 0100h	Assign origin to be 0100h
start: mov ax,data	Moving data segment to ax
mov ds,ax	Moving ax to ds- data segment is in ds now
mov al,opr1	Moving contents of opr1 to al
mov bl,opr2	Moving contents of opr2 to bl
mov ch,00h	Assign ch as 00h
add al,bl	Add al and bl(al+bl), store the result in al
daa	Converting sum in al to valid BCD.
jnc here	If no carry, jump to label here

inc ch
here: mov result,al
mov carry,ch
mov ah,4ch
int 21h
code ends
end start
	Increment ch if there is a carry
	Moving contents of al to result
	Moving contents of ch to carry
	
	Invoking interrupt to terminate program.

Unassembled code:

```
-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
-
```

Snapshot of I/P and O/P for Addition:

```
-d 076a:0000
076A:0000  80 80 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  80 80 60 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   ................
-
```

7b. BCD Subtraction

Algorithm:

- Declare the segments- cs and ds as code and data respectively.
- Initialise the data segment with two operands (opr1 and opr2), result (initialised to 0) and carry (initialised to 0)
- Assemble the code segment and set the origin(starting address) as 0100(in this case).
- Manually transfer the data segment to the register ds through ax.
- Load the value in opr1 and opr2 to al and bh respectively.
- Assign the value 00 to ch, the carry register.
- Subtract the contents of al and bh(al-bh) and store the result in al.
- Convert the difference of al and bh in al to BCD using das command.
- If there is a carry, increment ch register, and find the 10's complement of the answer in al register by assigning 99h to ah, subtracting ah and ah, and adding 1 to the difference.
- Convert the final 10's complement value of the answer to BCD using daa command.
- If there is no carry, jump to the here label, and transfer the answer in al to result, and the carry value in ch to carry register.
- Interrupt the program and terminate

Program	Comments
assume cs:code,ds:data	Declare segments cs and ds
data segment	
opr1 db 10h	Initialise opr1 to 10h
opr2 db 20h	Initialise opr2 to 20h
result db 00H	Initialise result to 00h
carry db 00H	Initialise carry to 00h
data ends	
code segment	
org 0100h	Assign origin to be 0100h
start: mov ax,data	Moving data segment to ax
mov ds,ax	Moving ax to ds- data segment is in ds now
mov al,opr1	Moving contents of opr1 to al
mov bh,opr2	Moving contents of opr2 to bh
mov ch,00h	Assign ch as 00h
sub al,bh	Subtract al and bh (al-bh), store the result in al
das	Converting difference in al to valid BCD
jnc here	If no carry, jump to label here
inc ch	Increment ch if there is a carry
mov ah,99h	Assign ah as 99h
sub ah,al	Subtract ah and al(ah-al), store the result in ah
mov al,ah	Moving contents of ah to al
add al,01h	Add al and 01h (finding 10's complement)
daa	Converting sum in al to valid BCD.
here: mov result,al	Moving contents of ah to result
mov carry,ch	Moving contents of ch to carry
mov ah,4ch	
int 21h	Invoking interrupt to terminate program.
code ends	
end start	

Unassembled code:

```
-u
076B:0100 B86A07        MOV     AX,076A
076B:0103 8ED8          MOV     DS,AX
076B:0105 A00000        MOV     AL,[0000]
076B:0108 8A3E0100      MOV     BH,[0001]
076B:010C B500          MOV     CH,00
076B:010E 2AC7          SUB     AL,BH
076B:0110 2F            DAS
076B:0111 730D          JNB     0120
076B:0113 FEC5          INC     CH
076B:0115 B499          MOV     AH,99
076B:0117 2AE0          SUB     AH,AL
076B:0119 8AC4          MOV     AL,AH
076B:011B B401          MOV     AH,01
076B:011D 02C4          ADD     AL,AH
076B:011F 27            DAA
-
```

```
-u
076B:0120 A20200        MOV     [0002],AL
076B:0123 882E0300      MOV     [0003],CH
076B:0127 B44C          MOV     AH,4C
076B:0129 CD21          INT     21
```

Snapshot of I/P and O/P for Subtraction:

```
-d 076a:0000
076A:0000  10 20 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  10 20 10 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   ................
-
```

Result: The MASM code for BCD Addition and BCD Subtraction was written and executed.