8 Bit Arithmetic Operations

Expt No:	1	**Name:**	Shivanirudh S G
Date :	21/08/2020	**Reg No:**	185001146

Aim:

To perform arithmetic operations on two 8 bit numbers.

Procedure:

- Mount masm folder to a drive on DOSBOX.
- Go to mounted drive.
- Save 8086 program with the extension **".asm"** in the same folder using the command **"edit"**.
- Assemble the **.asm** file using the command **"masm filename.asm"**.
- Link the assmebled **.obj** file using the command **"link filename.obj"**.
- Debug the executable file **.exe** with the **"debug filename.exe"** command.
  - The unassembled code can be viewed by the command **"u"**.
  - Use the command **"d segment:offset"** to view the contents of memory locations from the specified segment:offset address.
  - Use command **"e segment:offset"** to change the values in memory.
  - Execute the program using command **"g"**.
  - The command **"q"** exits from the debug session.

---

8 Bit Addition

Algorithm:

- Move the data segment to the AX register and then move it to the DS register.
- Move the first operand to AH register.
- Move the second operand to the BH register.
- Initially set the CH register to 00h.
- Then add using ADD AH,BH.
- Using JNC instruction check for carry and if there is no carry, no need to increment CH.
- Else, increment CH by 1.
- The result and carry stored in AH and CH should be moved to RESULT and CARRY respectively.

Program:

Program	Comments
assume cs:code,ds:data	Declare code and data segments
data segment	Start of data segment
opr1 db 11h	Define byte opr1 with hex value 11
opr2 db 99h	Define byte opr2 with hex value 99
result db 00H	Define byte result with hex value 00
carry db 00H	Define byte carry with hex value 00
data ends	End of data segment
code segment	Start of code segment
org 0100h	Set preferred offset
start: mov ax,data	Move data segment contents to AX register
mov ds,ax	Move data in AX register to DS register
mov ah,opr1	Move contents of opr1 to AH register
mov bh,opr2	Move contents of opr2 to BH register
mov ch,00h	Move hex value 00 to CH register
add ah,bh	AH = AH + BH
jnc here	Jump to the label here, if there is no carry
inc ch	Increment value of CH if there is a carry
here: mov result,ah	Move contents of AH register to result
mov carry,ch	Move contents of CH register to carry
int 21h	Request interrupt routine
code ends	End of code segment
end start	

Input and Output:

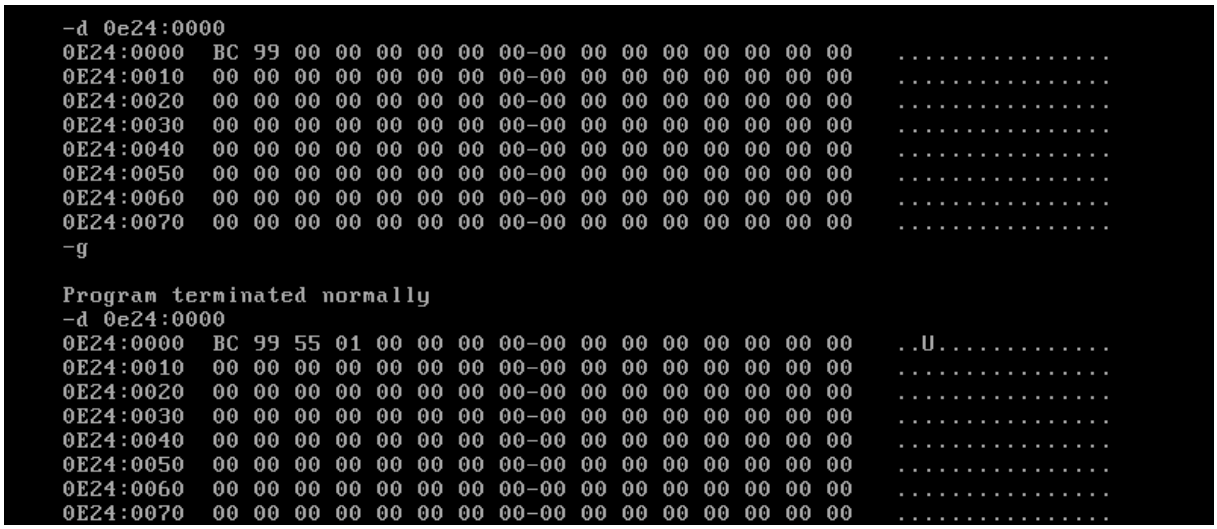

Figure 1: **Input:** *opr1:* BCh, *opr2:* 99h; **Output:** *Result:* 55h, *Carry:* 01h

8 Bit Subtraction

Algorithm:

- Move the data segment to the AX register and then move it to the DS register.
- Move the first operand to AH register.
- Move the second operand to the BH register.
- Initially set the CH register to 00h.
- Then subtract using SUB AH,BH.
- Check for carry using JNC instruction. If no carry then it means AH > BH and hence no need to increment CH and no need to complement AH.
- Else, AH<BH. Hence we have to take 2's complement of AH using NEG AH and also increment CH by 1 using INC CH.
- The result and carry stored in AH and CH should be moved to RESULT and CARRY respectively.

Program:

Program	**Comments**
assume cs:code,ds:data	Declare code and data segments
data segment	Start of data segment
opr1 db 11h	Define byte opr1 with hex value 11
opr2 db 99h	Define byte opr2 with hex value 99
result db 00H	Define byte result with hex value 00
carry db 00H	Define byte carry with hex value 00
data ends	End of data segment
code segment	Start of code segment
org 0100h	Set preferred offset
start: mov ax,data	Move data segment contents to AX register
mov ds,ax	Move data in AX register to DS register
mov ah,opr1	Move contents of opr1 to AH register
mov bh,opr2	Move contents of opr2 to BH register
mov ch,00h	Move hex value 00 to CH register
sub ah,bh	AH = AH - BH
jnc here	Jump to the label here, if there is no carry
inc ch	Increment value of CH if there is a carry
neg ah	Negate the contents of the AH register
here: mov result,ah	Move contents of AH register to result
mov carry,ch	Move contents of CH register to carry
int 21h	Request interrupt routine
code ends	End of code segment
end start	

Input and Output:

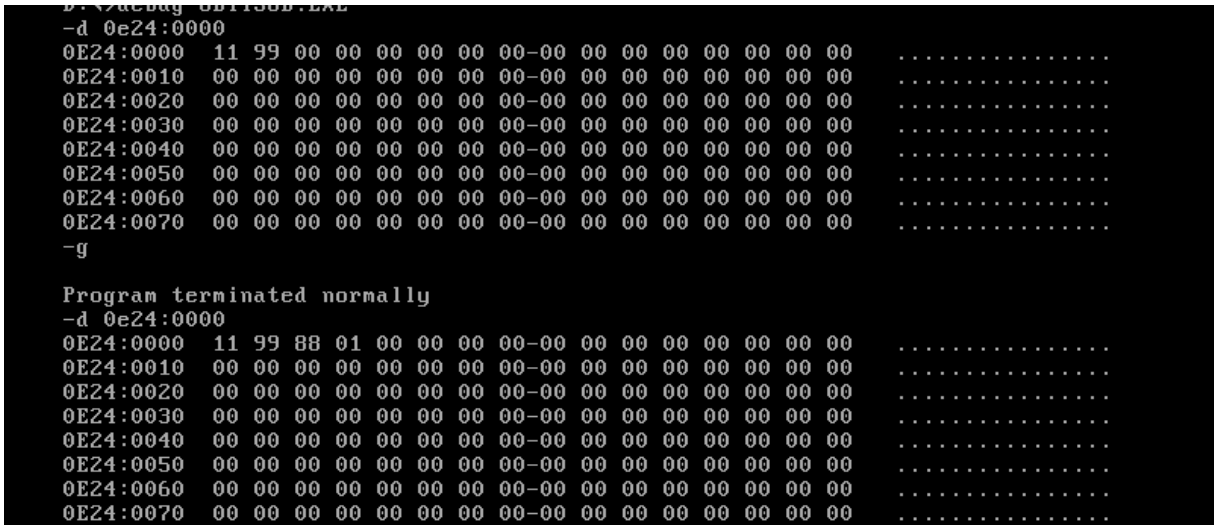

Figure 2: **Input:** *opr1:* 11h, *opr2:* 99h; **Output:** *Result:* 88h, *Sign:* 01h

8 Bit Multiplication

Algorithm:

- Move the data segment to the AX register and then move it to the DS register.
- Move the first operand to AL register.
- Move the second operand to the BL register.
- Then multiply using MUL BL.(Since AL is default operand register for MUL instruction we only need to specify the other operand register.)
- The result stored in AX register(16 bit- because multiplication of two 8 bit numbers yields a 16 bit number) should now be moved to RESULT.

Program:

Program	Comments
assume cs:code,ds:data	Declare code and data segments
data segment	Start of data segment
opr1 db 11h	Define byte opr1 with hex value 11
opr2 db 99h	Define byte opr2 with hex value 99
resulth db 00H	Define byte resulth with hex value 00
resultl db 00H	Define byte resultl with hex value 00
data ends	End of data segment
code segment	Start of code segment
org 0100h	Set preferred offset
start: mov ax,data	Move data segment contents to AX register
mov ds,ax	Move data in AX register to DS register
mov al,opr1	Move contents of opr1 to AL register
mov ah, 00H	Move hex value 00 to AH register
mov bh,opr2	Move contents of opr2 to BH register
mul bh	AX = AL * BH
here: mov resulth,ah	Move contents of AH register to resulth
mov resultl,al	Move contents of AL register to resultl
int 21h	Request interrupt routine
code ends	End of code segment
end start	

Input and Output:

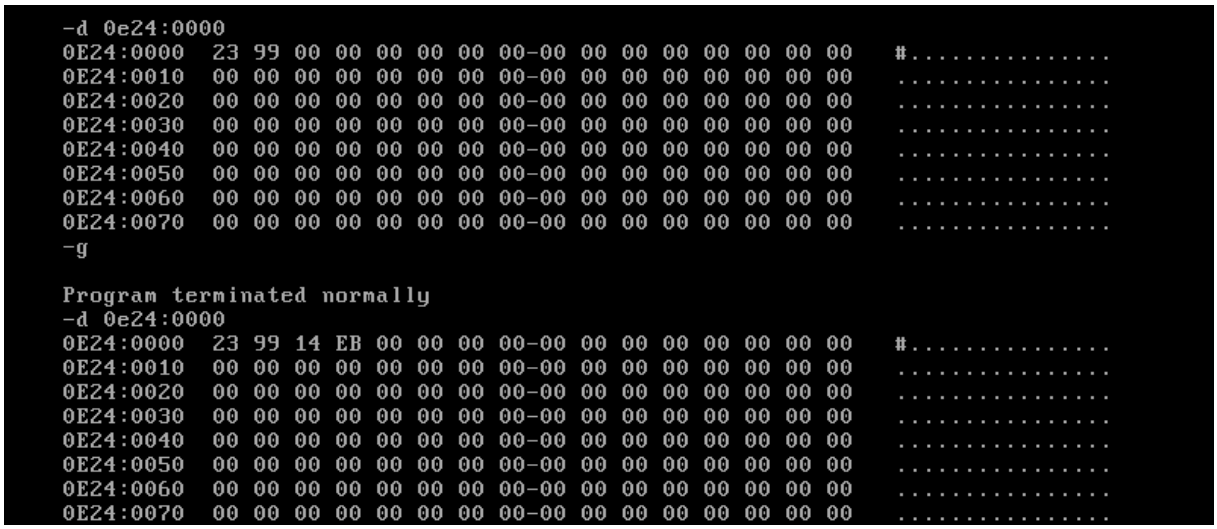

Figure 3: **Input:** *opr1:* 23h, *opr2:* 99h; **Output:** *Result:* EF 14h

8 Bit Division

Algorithm:

- Move the data segment to the AX register and then move it to the DS register.
- Now, set AH register to 00h and move first operand to AL register. (Since we can't directly divide a 8 bit number by 8 bit number in 8086, we now make our dividend 16 bit by storing 00h in AH register and the 8-bit operand 1 in AL register).
- Move the second operand to the BL register.
- Now divide using DIV BL.(It will perform AX / BL. Because AH is 00h, what actually happens is the division of a 16 bit number by a 8 bit number).
- The quotient and remainder stored in AL and AH should be moved to QUOTIENT and REMAINDER respectively

Program:

Program	**Comments**
assume cs:code,ds:data	Declare code and data segments
data segment	Start of data segment
opr1 db 11h	Define byte opr1 with hex value 11
opr2 db 10h	Define byte opr2 with hex value 99
resultq db 00H	Define byte resultq with hex value 00
resultr db 00H	Define byte resultr with hex value 00
data ends	End of data segment
code segment	Start of code segment
org 0100h	Set preferred offset
start: mov ax,data	Move data segment contents to AX register
mov ds,ax	Move data in AX register to DS register
mov al,opr1	Move contents of opr1 to AL register
mov ah, 00H	Move hex value 00 to AH register
mov bh,opr2	Move contents of opr2 to BH register
div bh	AX = AL / BH
here: mov resultq,ah	Move contents of AH register to resultq
mov resultr,al	Move contents of AL register to resultr
int 21h	Request interrupt routine
code ends	End of code segment
end start	

Input and Output:

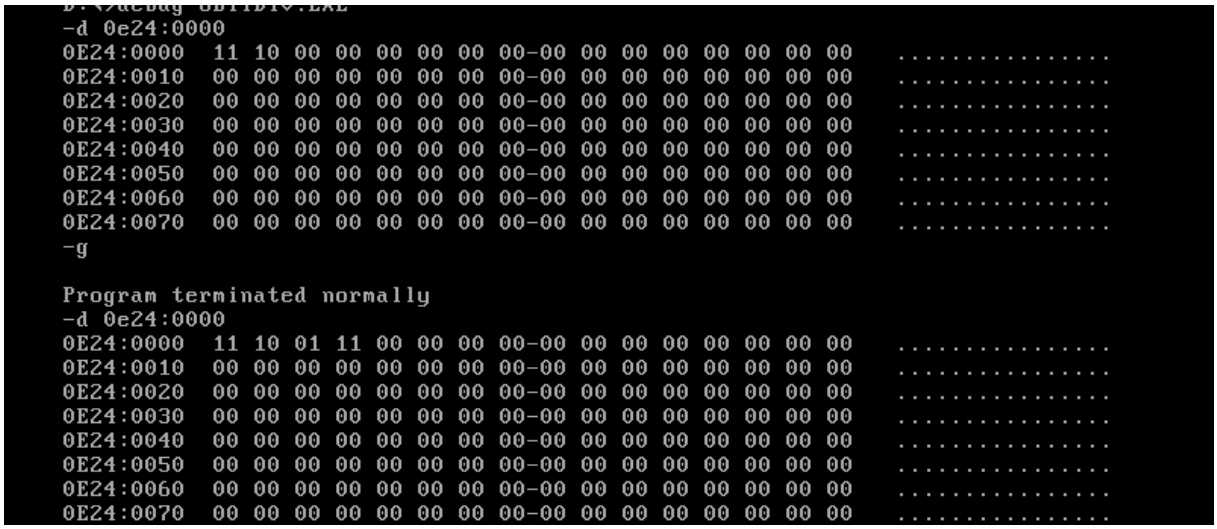

Figure 4: **Input:** *opr1:* 11h, *opr2:* 10h; **Output:** *Quotient:* 01h, *Remainder:* 11h

---

Result:

The 8086 programs were written to perform 8-bit arithmetic operations, and the results observed.