UCS1512 – Microprocessors Lab

8 BIT ARITHMETIC OPERATIONS

Exp no : 1 Name: Sreedhar V

Date : 21-08-2020 Reg no: 185001161

AIM:

To program and execute the 8 bit arithmetic operations like addition, subtraction, multiplication and division in 8086 using an emulator.

Procedure for executing MASM:

- Install and run DOSBox and mount the masm folder to a drive in DOSBox.
- Go to the mounted drive.[usually d is used]
- Save the 8086 program with extension .asm in the same folder using command "edit".
- After creating the file, assemble it using the command "masm filename.asm".
- Link the file using the command "link filename.obj;".
- Use debug command with filename.exe to execute and analyse the memory contents – "debug filename.exe".
- In debug, the command "u" will display the unassembled code.
- Use command "d segment:offset" to see the content of memory locations starting from segment:offset address.
- To change the value in memory, use the command "e segment:offset". To stop editing, press enter.
- Verify the memory contents to ensure the updates using command "d".
- Execute using the command "g" and check the outputs.
- Use command "q" to exit from debug and command "exit" from command prompt to close DOSBox.

8- Bit Addition:

Algorithm:

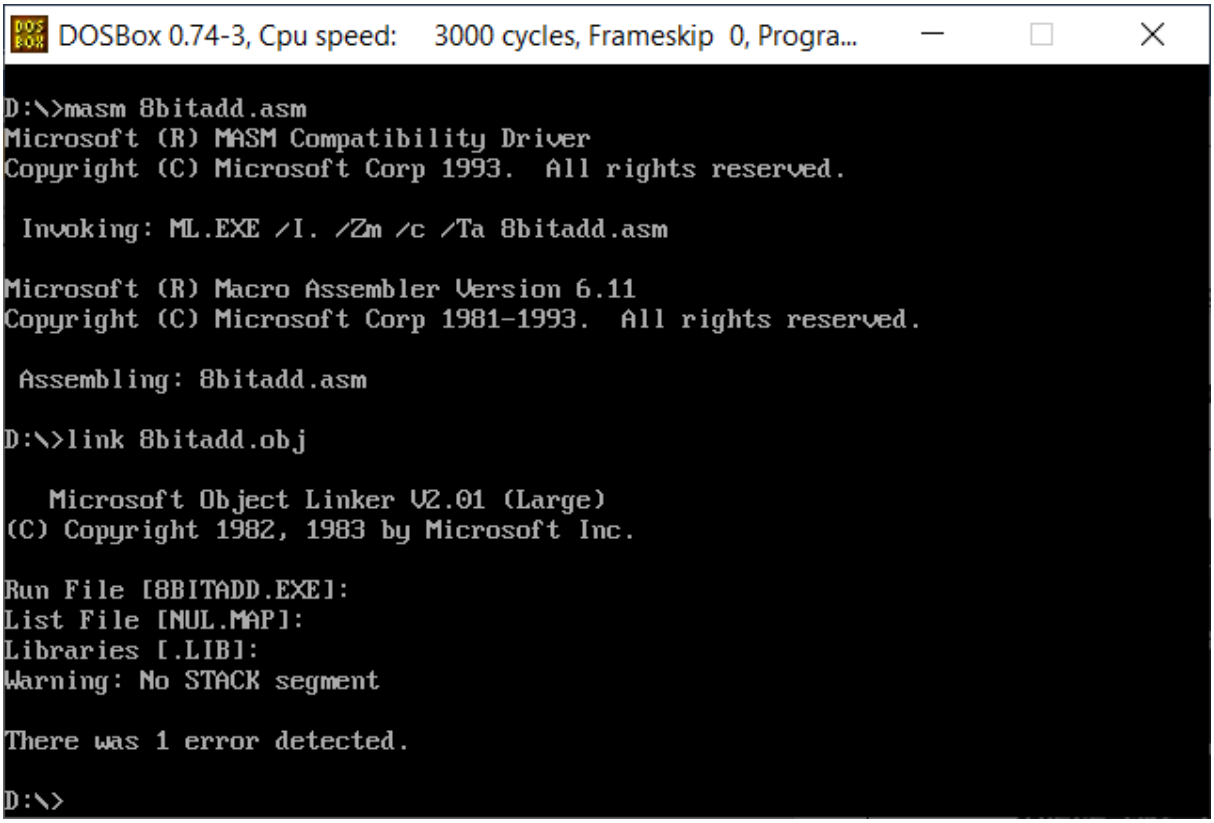
- Program is set to run from any specified memory position.
- Load data from opr1 to register AL (first number)
- Load data from opr2 to register BL (second number)
- Add these two numbers (contents of register AL and register BL)
- Initialise carry to 0.
- Jump to final steps if there is no carry.
- Increment carry.
- Store additional values to result.
- Terminate the program.

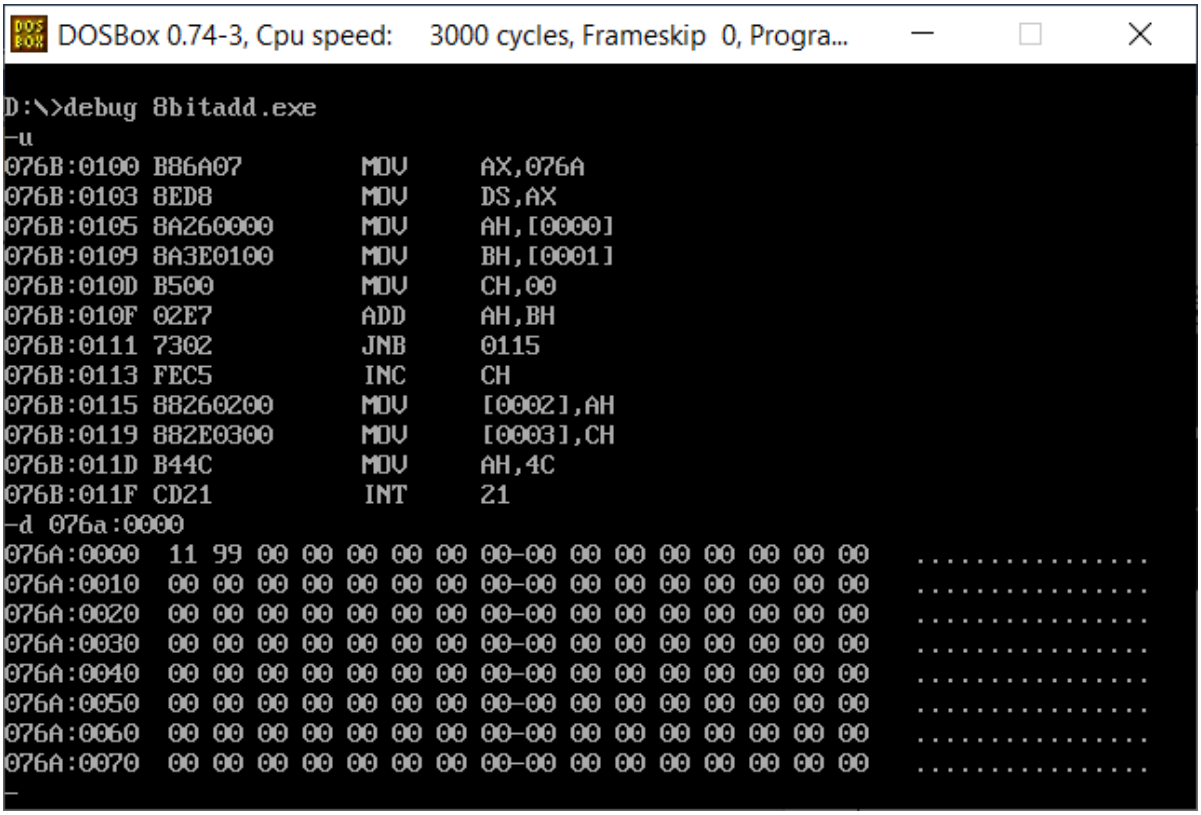
Program:

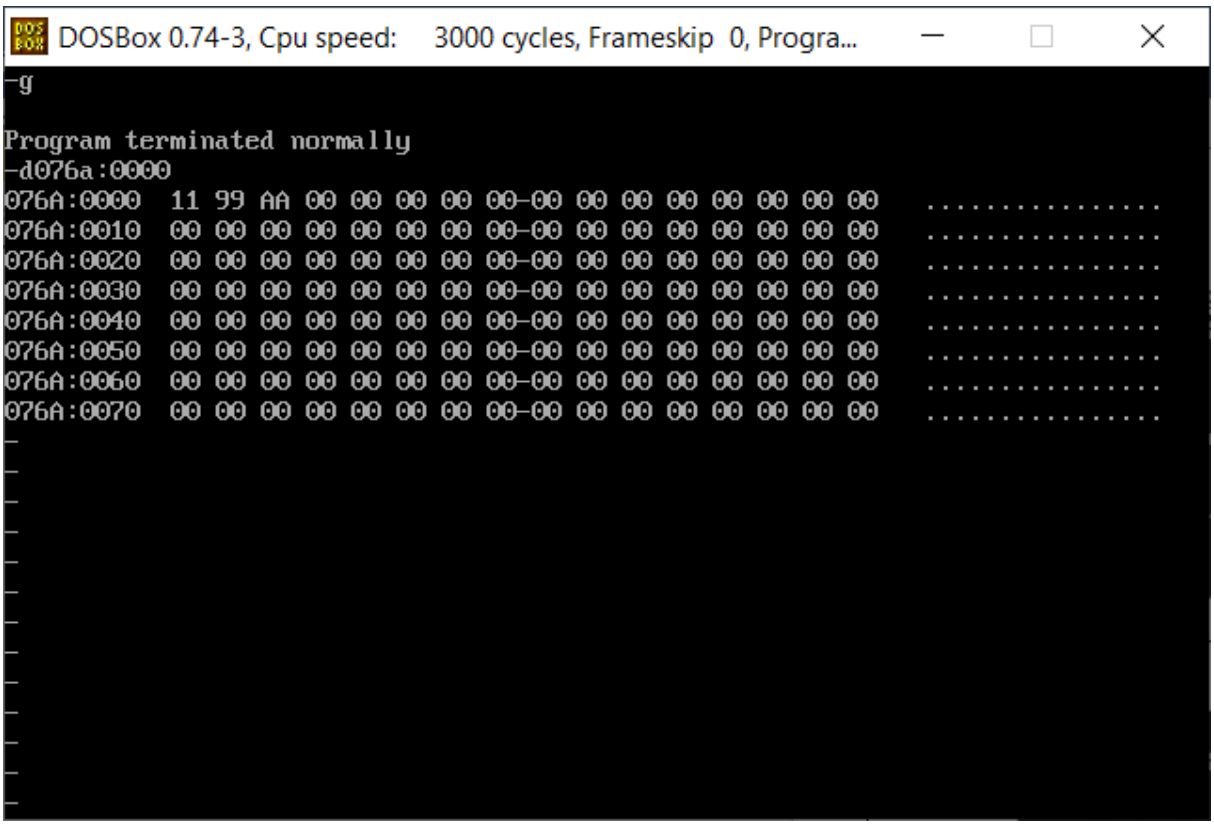
CODE	COMMENT

;Program for adding 2, 8 bit numbers
 	
assume cs:code,ds:data
data segment
opr1 db 11h
opr2 db 99h
result db 00H
carry db 00H
data ends
code segment
org 0100h
start:
mov ax,data
mov ds,ax
mov ah,opr1
mov bh,opr2
mov ch,00h
add ah,bh
jnc here
inc ch
here:
mov result,ah
mov carry,ch
mov ah,4ch
int 21h
code ends
end start
	Data segment initialized
	opr1 initialised and set to 11
	opr2 initialised and set to 99
	result initialised and set to 00
	carry initialised and set to 00
	
	Code segment begins
	Originating address is set at 0100
	
	Address of data segment moved to ax
	From ax, transferred to ds
	Value of opr1 transferred to ah
	Value of opr2 transferred to bh
	ch is initialised and set to 0
	Addition takes palce
	Junction created
	• Jump if no carry
	• Else increment ch
	
	data transferred from ah to result
	data transferred from ch to carry
	
	Program terminates

Snapchat of the program:







Result:

8 bit addition is executed and verified using an emulator.

8- Bit Subtraction:

Algorithm:

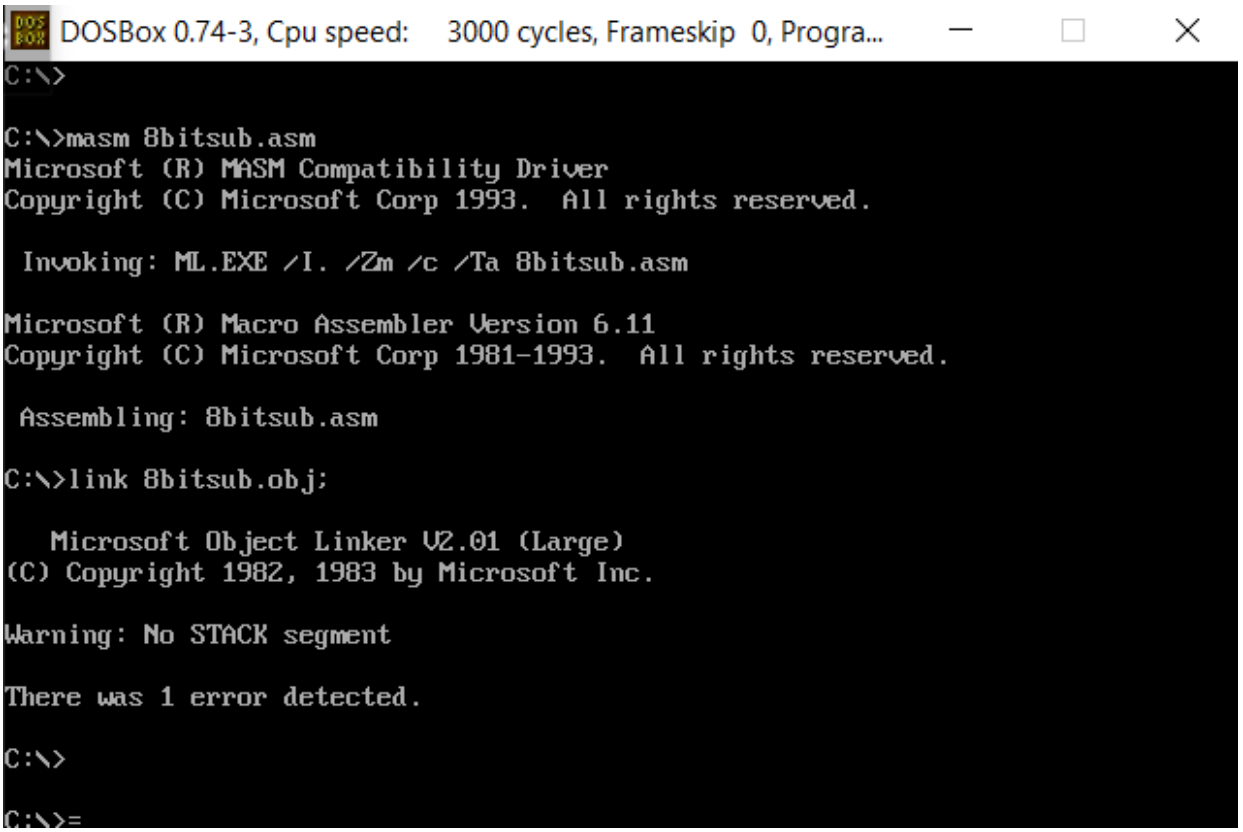
- Program is set to run from any specified memory position.
- Load data from opr1 to register AL (first number)
- Load data from opr2 to register BL (second number)
- Subtract these two numbers (contents of register AL and register BL)
- Initialise carry to 0.
- Jump to final steps if there is no carry.
- Increment carry.
- And the result is negated.
- Store answer to result.
- Terminate the program.

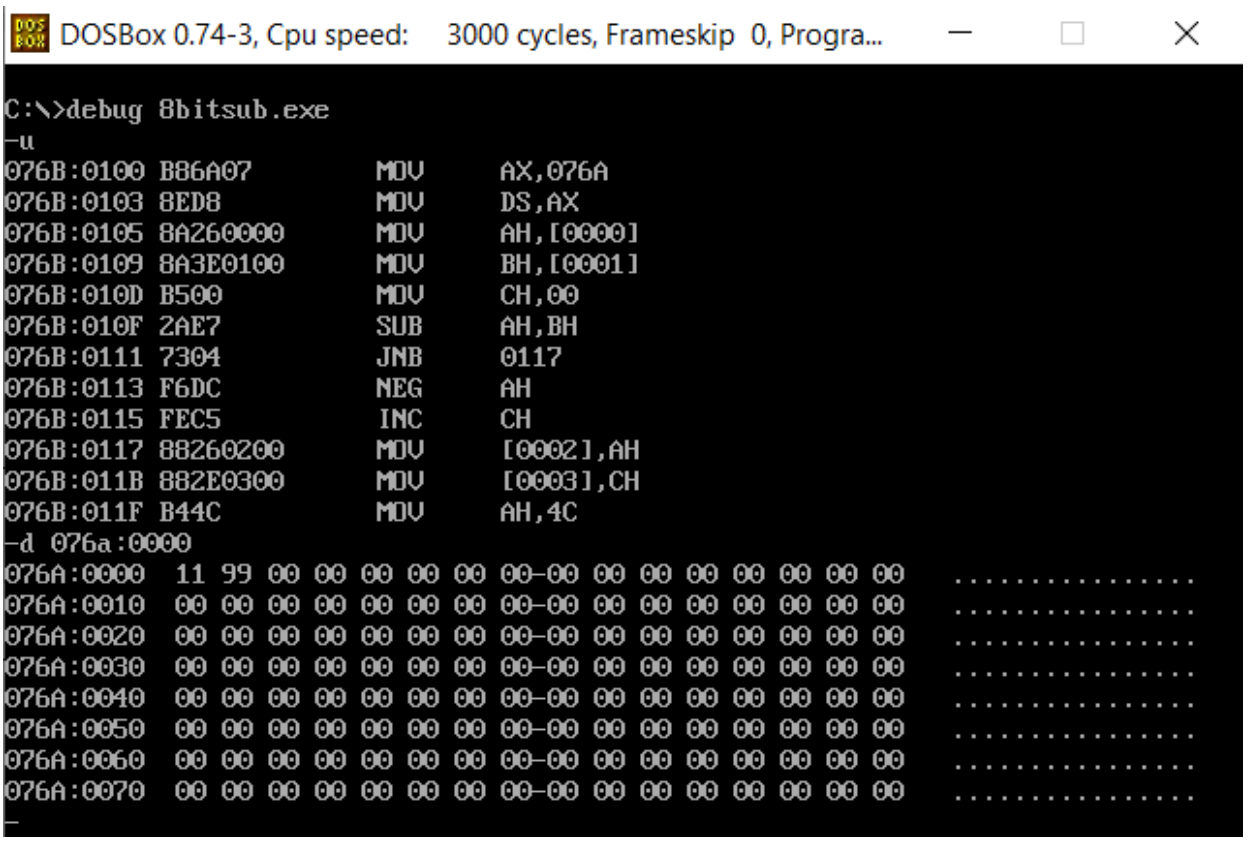
Program:

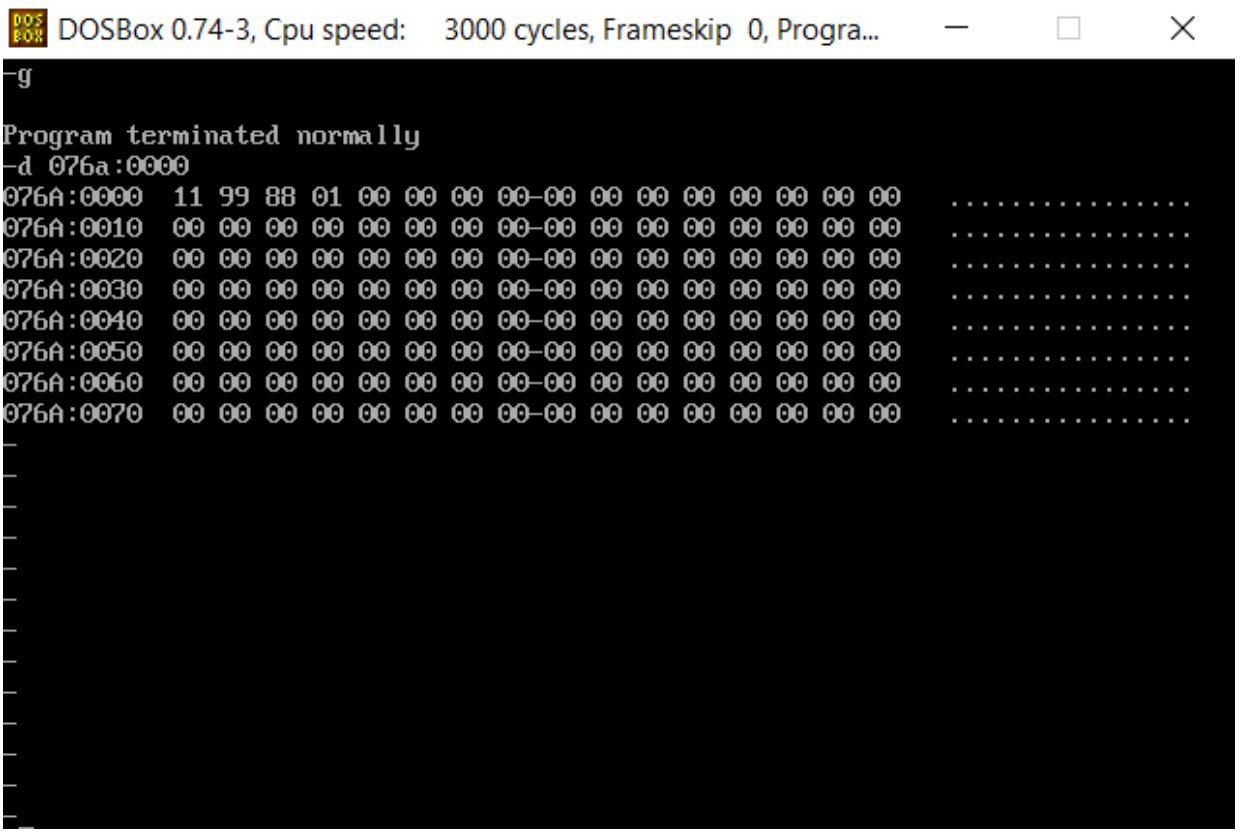
CODE	COMMENT

Program for Subtracting 2, 8-bit numbers
 	
assume cs:code,ds:data data segment
opr1 db 11h
opr2 db 99h
result db 00H
carry db 00H
data ends
code segment
org 0100h
start:
mov ax,data
mov ds,ax
mov ah,opr1
mov bh,opr2
mov ch,00h
sub ah,bh
jnc here
neg ah
inc ch
here:
mov result,ah
mov carry,ch
mov ah,4ch int
21h
code ends
end start
	Data segment initialized opr1
	initialised and set to 11 opr2
	initialised and set to 99
	result initialised and set to 00
	carry initialised and set to 00
	
	Code segment begins
	Originating address is set at 0100
	
	Address of data segment moved to ax
	From ax, transferred to ds
	Value of opr1 transferred to ah
	Value of opr2 transferred to bh
	
	ch is initialise and set to 0
	Substarction takes palce
	Junction created
	• Jump if no carry
	• Else: negate ah and increment ch
	
	data transferred from ah to result
	data transferred from ch to carry
	
	Program terminates

Snapchat of the program:







Result:

8 bit subtraction is executed and verified using an emulator.

8-Bit Multiplication:

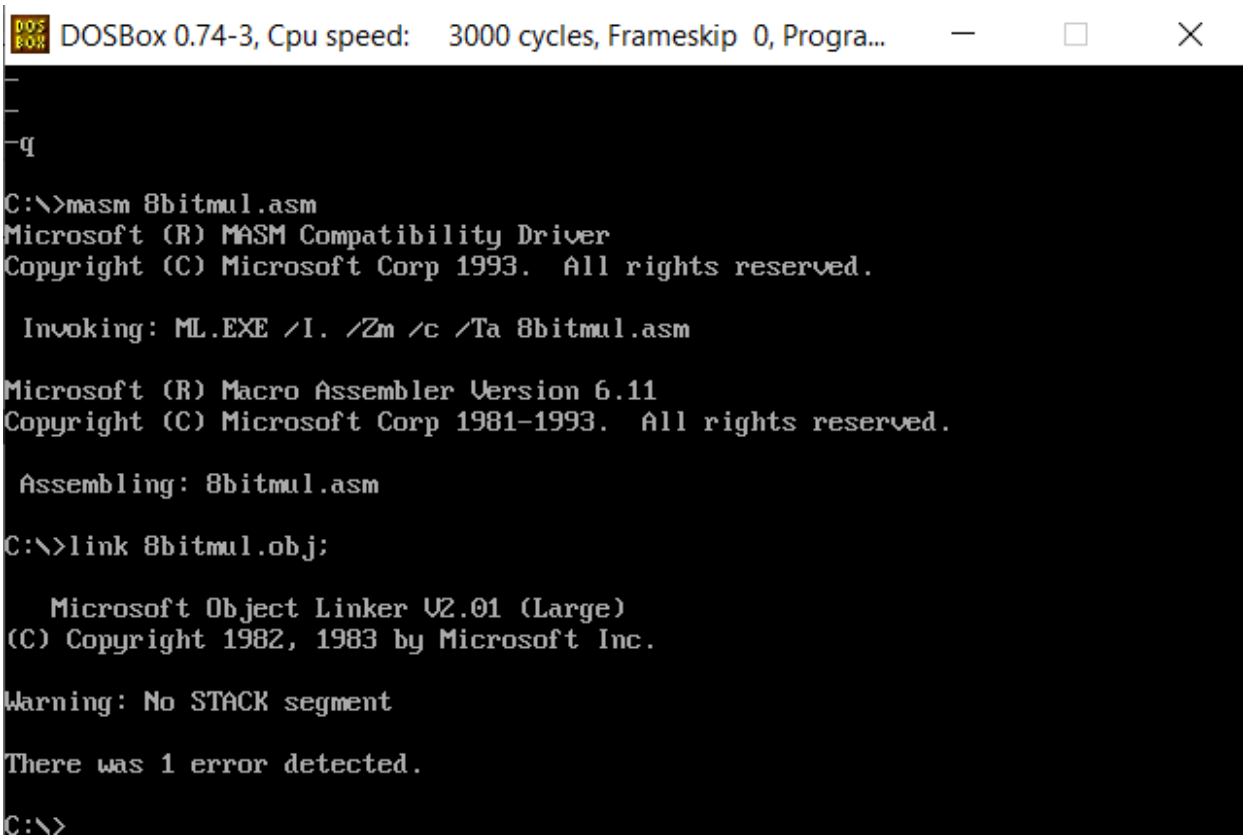
Algorithm:

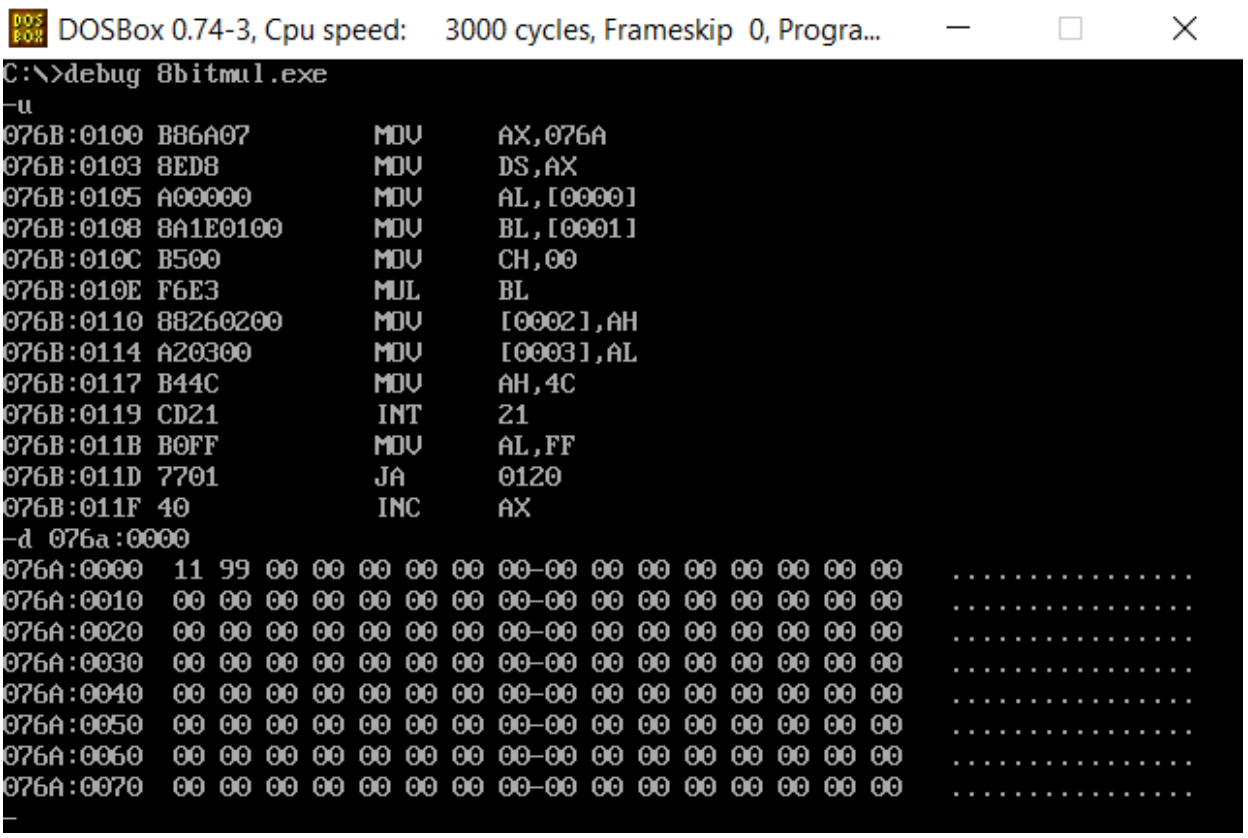
- Program is set to run from any specified memory position.
- Load data from opr1 to register AL (first number)
- Load data from opr2 to register BL (second number)
- Multiply these two numbers (contents of register AL and register BL)
- Initialise carry to 0.
- Multiplied values is stored in ah and al
- These two values are stored in different locations for better representation.
- Terminate the program.

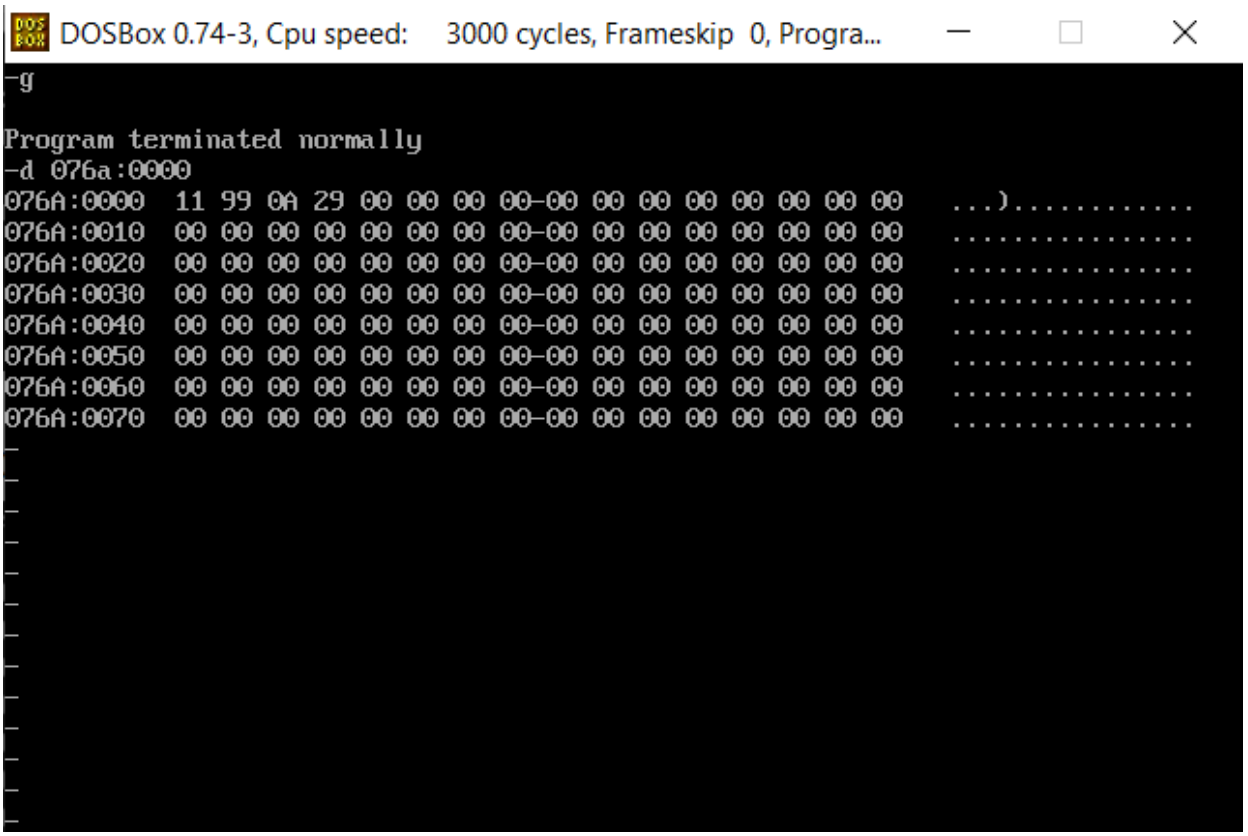
Program:

CODE	COMMENT
Program for Multiplying 2, 8 bit numbers	
assume cs:code,ds:data	
data segment	Data segment initialized
opr1 db 11h	opr1 initialised and set to 11
opr2 db 99h	opr2 initialised and set to 99
result db 00H	result initialised and set to 00
res db 00H	res initialised and set to 00
data ends	
code segment	Code segment begins
org 0100h	Originating address is set at 0100
start:	
mov ax,data	Address of data segment moved to ax
mov ds,ax	From ax, transferred to ds
mov al,opr1	Value of opr1 transferred to al
mov bl,opr2	Value of opr2 transferred to bl
mul bl	Multiply al and bl
mov result,ah	data transferred from ah to result
mov res,al	data transferred from al to res
mov ah,4ch	
int 21h	Program terminates
code ends	
end start	

Snapchat of the program:







Result:

8 bit multiplication is executed and verified using an emulator.

8- Bit Division:

Algorithm:

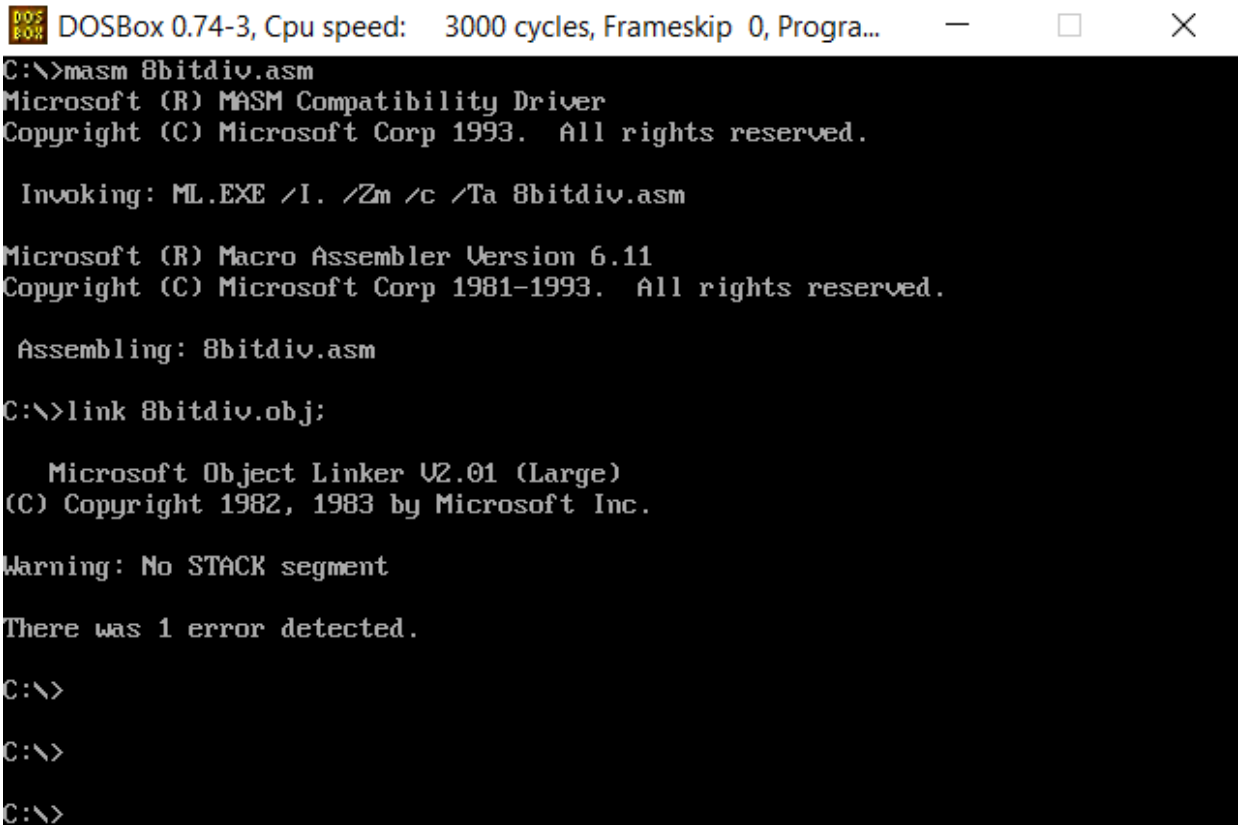
- Load data from opr1 to register AL (first number)
- Load data from opr2 to register BL (second number)
- Initialise a variable for remainder to 0.
- Divide these two numbers (contents of register AL and register BL)
- Move al value to quotient variable.
- Move ah value to remainder variable.
- Terminate the program.

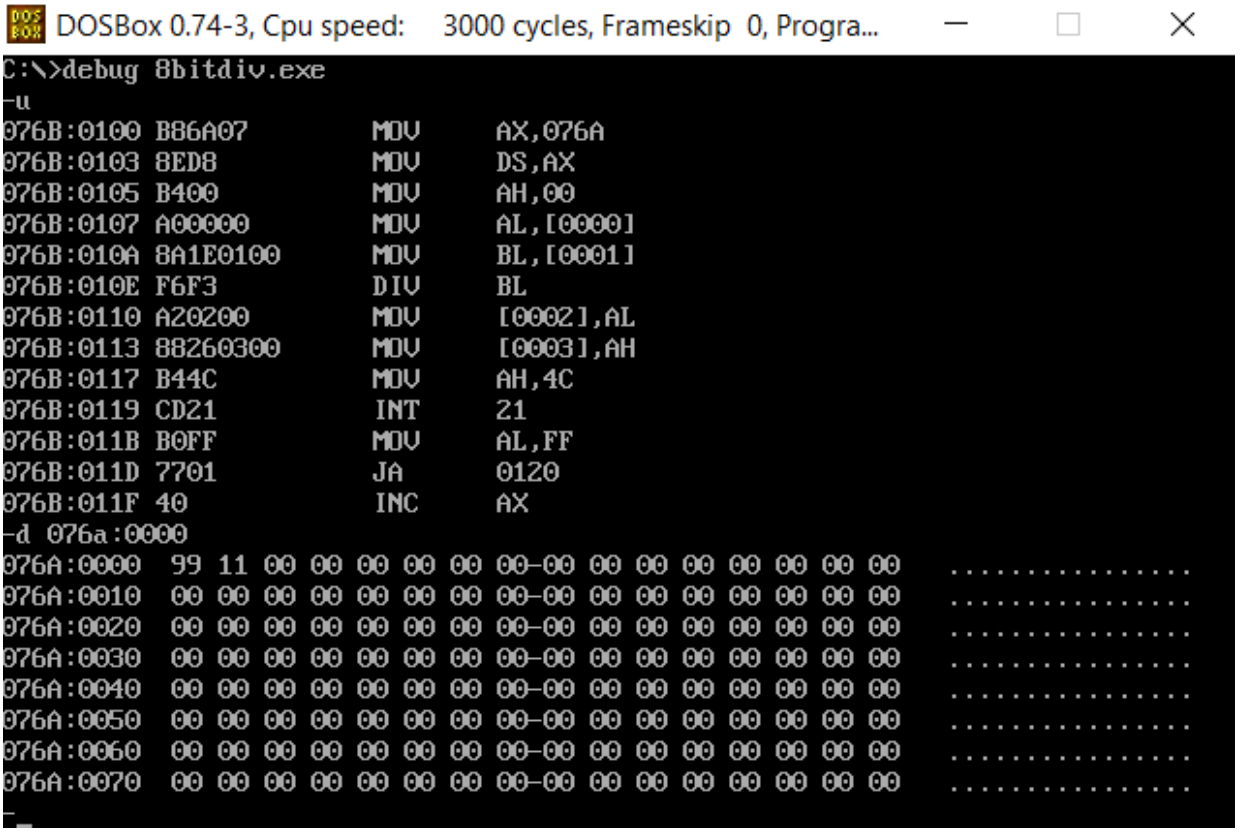
Program:

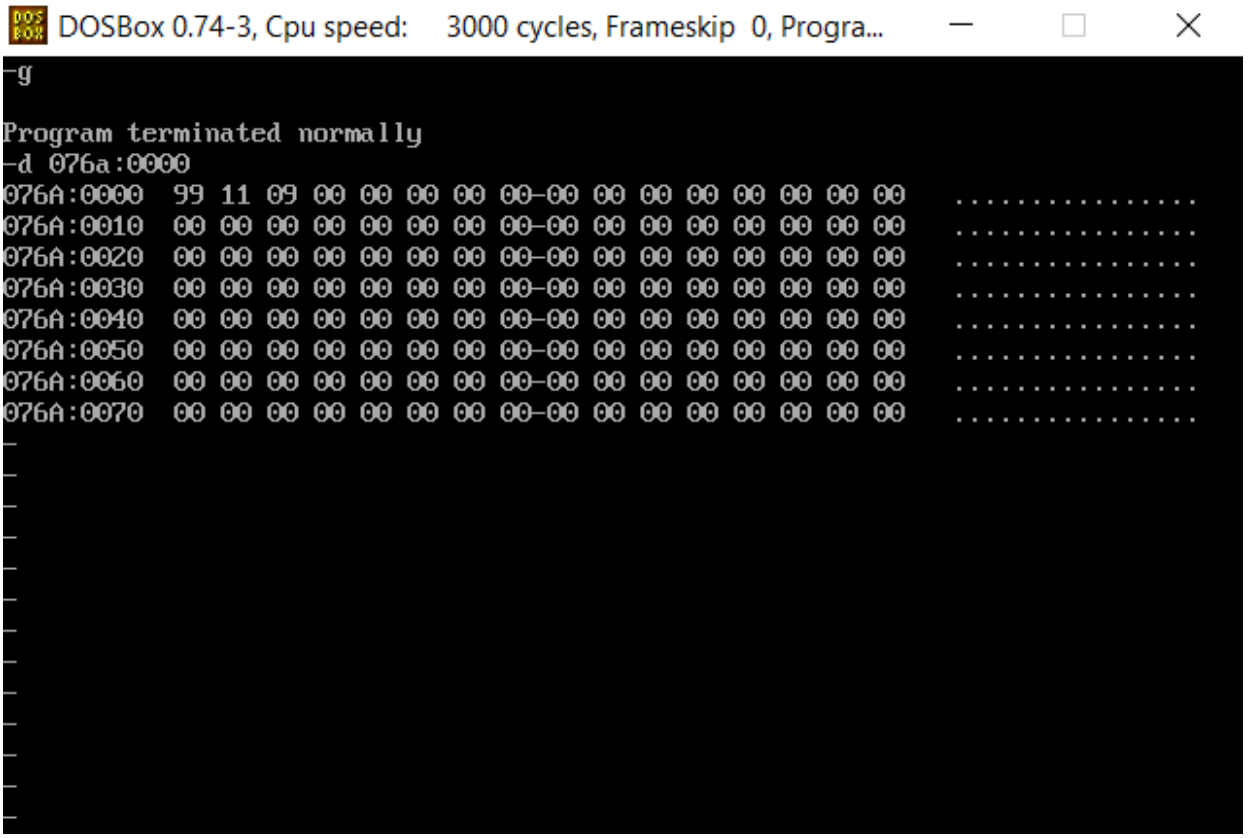
CODE	COMMENT

;Program for Dividing 2, 8 bit numbers
 	
assume cs:code,ds:data
data segment
opr1 db 99h
opr2 db 11h
quotient db 00H
rem db 00H
data ends
code segment
org 0100h
start:
mov ax,data
mov ds,ax
mov ah,00h
mov al,opr1
mov bl,opr2
div bl
mov quotient,al
mov rem,ah
mov ah,4ch
int 21h
code ends
end start
	Data segment initialized
	opr1 initialised and set to 99
	opr2 initialised and set to 11
	quotient initialised and set to
	00 rem initialised and set to 00
	
	Code segment begins
	Originating address is set at 0100
	
	Address of data segment moved to ax
	From ax, transferred to ds
	ah is initialise and set to 0
	Value of opr1 transferred to al
	Value of opr2 transferred to bl
	Division takes place
	data transferred from al to result
	data transferred from ah to rem
	
	Program terminates

Snapchat of the program:







Result:

8 bit division is executed and verified using an emulator.