

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
<pre>;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</pre>	<p>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</p> <p>Code segment begins Originating address is set at 0100</p> <p>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 place Junction created</p> <ul style="list-style-type: none">• Jump if no carry• Else increment ch <p>data transferred from ah to result data transferred from ch to carry</p> <p>Program terminates</p>

Snapchat of the program:

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
D:\>masm 8bitadd.asm
Microsoft (R) MASM Compatibility Driver
Copyright (C) Microsoft Corp 1993. All rights reserved.

Invoking: ML.EXE /I. /Zm /c /Ta 8bitadd.asm

Microsoft (R) Macro Assembler Version 6.11
Copyright (C) Microsoft Corp 1981-1993. All rights reserved.

Assembling: 8bitadd.asm

D:\>link 8bitadd.obj

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

Run File [8BITADD.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:
Warning: No STACK segment

There was 1 error detected.

D:\>
```

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
D:\>debug 8bitadd.exe
-u
076B:0100 B86A07      MOV     AX,076A
076B:0103 8ED8        MOV     DS,AX
076B:0105 8A260000      MOV     AH,[0000]
076B:0109 8A3E0100      MOV     BH,[0001]
076B:010D B500        MOV     CH,00
076B:010F 02E7        ADD     AH,BH
076B:0111 7302        JNB     0115
076B:0113 FEC5        INC     CH
076B:0115 88260200      MOV     [0002],AH
076B:0119 882E0300      MOV     [0003],CH
076B:011D B44C        MOV     AH,4C
076B:011F CD21      INT     21
-d 076a:0000
076A:0000 11 99 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 .....
-
```


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:

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
C:\>

C:\>masm 8bitsub.asm
Microsoft (R) MASM Compatibility Driver
Copyright (C) Microsoft Corp 1993. All rights reserved.

    Invoking: ML.EXE /I. /Zm /c /Ta 8bitsub.asm

Microsoft (R) Macro Assembler Version 6.11
Copyright (C) Microsoft Corp 1981-1993. All rights reserved.

    Assembling: 8bitsub.asm

C:\>link 8bitsub.obj;

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

Warning: No STACK segment

There was 1 error detected.

C:\>

C:\>=
```

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
C:\>debug 8bitsub.exe
-u
076B:0100 B86A07      MOV     AX,076A
076B:0103 8ED8        MOV     DS,AX
076B:0105 8A260000      MOV     AH,[0000]
076B:0109 8A3E0100      MOV     BH,[0001]
076B:010D B500          MOV     CH,00
076B:010F 2AE7          SUB     AH,BH
076B:0111 7304          JNB     0117
076B:0113 F6DC          NEG     AH
076B:0115 FEC5          INC     CH
076B:0117 88260200      MOV     [0002],AH
076B:011B 882E0300      MOV     [0003],CH
076B:011F B44C          MOV     AH,4C
-d 076a:0000
076A:0000 11 99 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 .....
```

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
-g
Program terminated normally
-d 076a:0000
076A:0000  11 99 88 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:

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 opr1 db 11h opr2 db 99h result db 00H res db 00H data ends code segment org 0100h start: mov ax,data mov ds,ax mov al,opr1 mov bl,opr2 mul bl mov result,ah mov res,al 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 res 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 al Value of opr2 transferred to bl Multiply al and bl data transferred from ah to result data transferred from al to res Program terminates

Snapchat of the program:

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
-
C:\>masm 8bitmul.asm
Microsoft (R) MASM Compatibility Driver
Copyright (C) Microsoft Corp 1993. All rights reserved.

Invoking: ML.EXE /I. /Zm /c /Ta 8bitmul.asm

Microsoft (R) Macro Assembler Version 6.11
Copyright (C) Microsoft Corp 1981-1993. All rights reserved.

Assembling: 8bitmul.asm

C:\>link 8bitmul.obj;

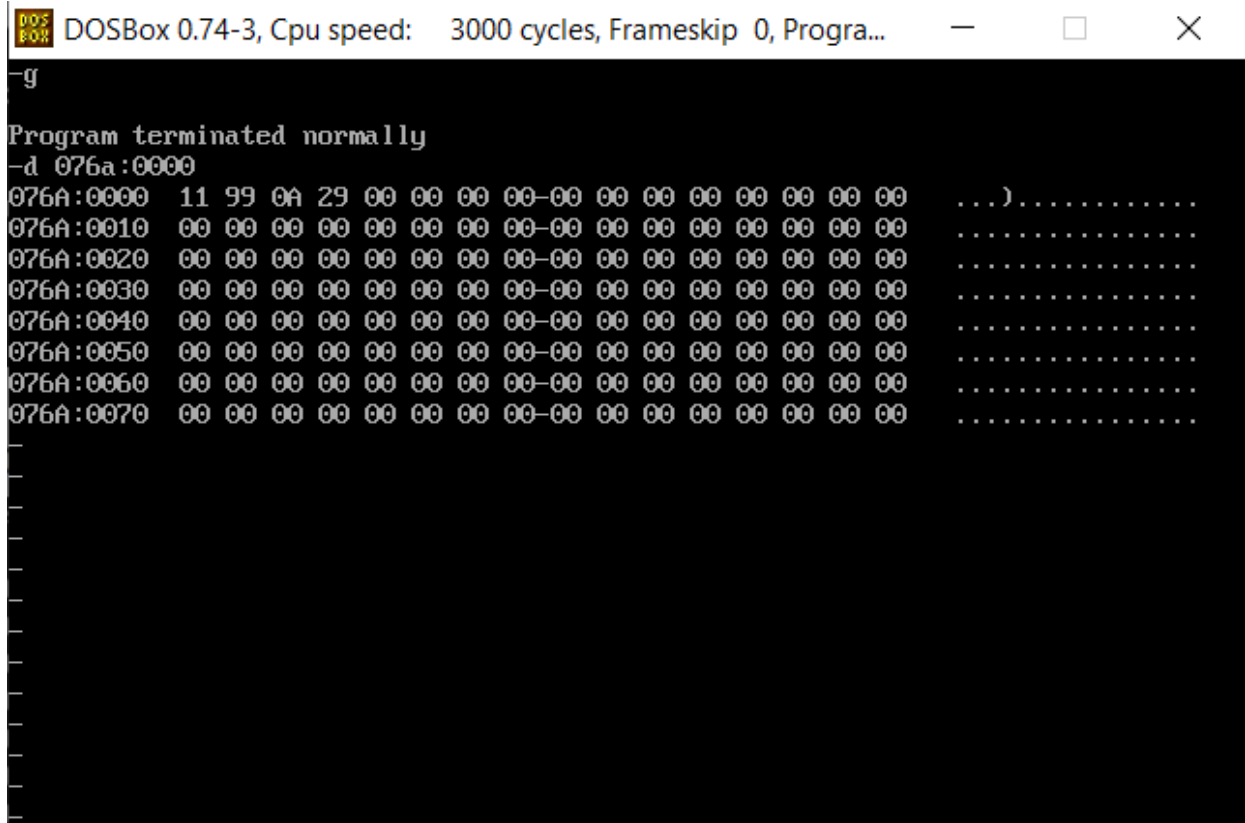
Microsoft Object Linker V2.01 (Large)
(C) Copyright 1982, 1983 by Microsoft Inc.

Warning: No STACK segment

There was 1 error detected.

C:\>
```

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
-
C:\>debug 8bitmul.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 F6E3        MUL     BL
076B:0110 88260200    MOV     [0002],AH
076B:0114 A20300      MOV     [0003],AL
076B:0117 B44C        MOV     AH,4C
076B:0119 CD21      INT     21
076B:011B B0FF      MOV     AL,FF
076B:011D 7701      JA      0120
076B:011F 40        INC     AX
-d 076a:0000
076A:0000 11 99 00 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 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:

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
<pre>;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</pre>	<p>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</p> <p>Code segment begins Originating address is set at 0100</p> <p>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</p> <p>Program terminates</p>

Snapchat of the program:

```
DOS BOX DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
C:\>masm 8bitdiv.asm
Microsoft (R) MASM Compatibility Driver
Copyright (C) Microsoft Corp 1993. All rights reserved.

Invoking: ML.EXE /I. /Zm /c /Ta 8bitdiv.asm

Microsoft (R) Macro Assembler Version 6.11
Copyright (C) Microsoft Corp 1981-1993. All rights reserved.

Assembling: 8bitdiv.asm

C:\>link 8bitdiv.obj;

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

Warning: No STACK segment

There was 1 error detected.

C:\>

C:\>

C:\>
```

```
DOS BOX DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
C:\>debug 8bitdiv.exe
-u
076B:0100 B86A07      MOV     AX,076A
076B:0103 8ED8        MOV     DS,AX
076B:0105 B400        MOV     AH,00
076B:0107 A00000      MOV     AL,[0000]
076B:010A 8A1E0100     MOV     BL,[0001]
076B:010E F6F3        DIV     BL
076B:0110 A20200      MOV     [0002],AL
076B:0113 88260300     MOV     [0003],AH
076B:0117 B44C        MOV     AH,4C
076B:0119 CD21        INT     21
076B:011B B0FF        MOV     AL,FF
076B:011D 7701        JA      0120
076B:011F 40        INC     AX
-d 076a:0000
076A:0000 99 11 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 .....
```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra... — □ ×

```
-g
Program terminated normally
-d 076a:0000
076A:0000  99 11 09 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 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:

8 bit division is executed and verified using an emulator.