

Exp No: 2
Date: 28/8/2020

16-bit Arithmetic Operations

Name: Gokhulnath T
Register Number: 185001051

Aim: To write assembly program to do following 16-bit arithmetic operations:

- a) 16-bit Addition
- b) 16-bit subtraction
- c) 16-bit multiplication
- d) 16-bit division

Procedure:

1. Install all the required file for executing MASM programs.(Masm, edit, link, debug etc..).
2. Write the assembly program in any editor before mounting the folder to the MASM.
3. Mount the folder that contains the assembly program with any name such as "d".
 - mount d e:\masm
4. Create the object file of the assembly program using masm.
 - masm 16BITADD.asm
5. Use the link to create the executable file of the object file created from the above step.
 - Link 16BITADD.obj
6. Run the executable file using debug.
 - debug 16BITADD.exe
7. By un-assembling the program you can check the code segment of the program
 - u 076b:0100
8. To check the data memory segment, you can use the memory option to view the data stored.
 - d 076a:0000
9. To enter your own values, you can use the enter option which will prompt for new values.
 - e 076a:0000
10. To execute the program, you can use go option
 - G
11. After successful execution and termination of the program, you can check the result by checking the data memory segment
 - d 076a:0000
12. The result can be viewed in the respective address mentioned in the program.

2 a) 16-bit addition

Algorithm:

- a) Assign 0000h to ax register
- b) Load contents of memory location opr1 in register ax
- c) Load contents of memory location opr2 in register bx
- d) Assign 0000h to cx register
- e) Add ax and bx content, store the result in ax
- f) If carry is generated continue else jump to step g
- g) Increment cx register
- h) Load content of ax register to memory location result
- i) Load content of cx register to memory location carry
- j) Load content 4ch termination code to ah register
- k) Stops execution of the program

Program:

Program	Comments
start: mov ax,data	Transferring the data from memory location data to ax
mov ds,ax	Transferring the data from memory location ax to da
mov ax,0000h	Load the data 0000h in to memory location of ax
mov ax,opr1	Transferring the data from memory location opr1 to ax
mov bx,opr2	Transferring the data from memory location opr2 to bx
mov cx,0000h	Transferring the data 0000h to cx
add ax,bx	ax = ax + bx
jnc here	Jump if no carry to here
inc cx	Increment cx
here: mov result,ax	Transferring the data from memory location ax to result
mov carry,cx	Transferring the data from memory location cx to carry
mov ah,4ch	Transferring the termination code 4ch to ah
int 21h	Termination
code ends	Code ends

Unassembled code:

```
D:\>debug 16BITADD.EXE
-u
076B:0100 B86A07      MOV     AX,076A
076B:0103 8ED8        MOV     DS,AX
076B:0105 B80000      MOV     AX,0000
076B:0108 A10000      MOV     AX,[0000]
076B:010B 8B1E0200     MOV     BX,[0002]
076B:010F B90000      MOV     CX,0000
076B:0112 03C3        ADD     AX,BX
076B:0114 7301        JNB     0117
076B:0116 41          INC     CX
076B:0117 A30400      MOV     [0004],AX
076B:011A 890E0600     MOV     [0006],CX
076B:011E B44C        MOV     AH,4C
```

Sample Input and output:

```
-e 076a:0000
076A:0000 11.ab 00.08 99.78 00.0b

-d 076a:0000
076A:0000 AB 08 78 0B 00 00 00 00-00 00 00 00 00 00 00 00 00  ..X.....
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 AB 08 78 0B 23 14 00 00-00 00 00 00 00 00 00 00 00  ..X.#.....
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  .....
```

Output explanation: 08AB + 0B78 = 1423 with carry 0

Result:

Thus, the assembly program for 16-bit adder is written and executed.

2 b) 16-bit Subtraction

Algorithm:

- a) Assign 0000h to ax register
- b) Load contents of memory location opr1 in register ax
- c) Load contents of memory location opr2 in register bx
- d) Assign 0000h to cx register
- e) sub ax and bx content, store the result in ax
- f) If carry is generated continue else jump to step g
- g) Increment cx register
- h) 2's complement(ax).
- i) Load content of ax register to memory location result
- j) Load content of cx register to memory location carry
- k) Load content 4ch termination code to ah register
- l) Stops execution of the program

Program:

Program	Comments
start: mov ax,data	Transferring the data from memory location data to ax
mov ds,ax	Transferring the data from memory location ax to da
mov ax,0000h	Load the data 0000h in to memory location of ax
mov ax,opr1	Transferring the data from memory location opr1 to ax
mov bx,opr2	Transferring the data from memory location opr2 to bx
mov cx,0000h	Transferring the data 0000hto cx
sub ax,bx	ax = ax - bx
jnc here	Jump if no carry to here
inc cx	Increment cx
neg ax	2's complement(ax).
here: mov result,ax	Transferring the data from memory location ax to result
mov carry,cx	Transferring the data from memory location cx to carry
mov ah,4ch	Transferring the termination code 4ch to ah
int 21h	Termination
code ends	Code ends

Unassembled code:

```
D:\>debug 16BITSUB.EXE
-u
076B:0100 B86A07      MOV     AX,076A
076B:0103 8ED8        MOV     DS,AX
076B:0105 B80000      MOV     AX,0000
076B:0108 A10000      MOV     AX,[0000]
076B:010B 8B1E0200    MOV     BX,[0002]
076B:010F B90000      MOV     CX,0000
076B:0112 2BC3        SUB     AX,BX
076B:0114 7303        JNB     0119
076B:0116 41          INC     CX
076B:0117 F7D8        NEG     AX
076B:0119 A30400      MOV     [0004],AX
076B:011C 890E0600    MOV     [0006],CX
-
```

Snapshot of sample input and output:

```
-e 076a:0000
076A:0000 11.2d  00.fe  99.ad  00.bc

-d 076a:0000
076A:0000 2D FE AD BC 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 2D FE AD BC 80 41 00 00-00 00 00 00 00 00 00 00  -....A.....
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  -.....
-
```

Output explanation: FE2D + BCAD = 4180 (00 carry indicates positive)

Snapshot of sample input and output:

```
-e 076a:0000
076A:0000  11.ab  00.08  99.78  00.0b

-d 076a:0000
076A:0000  AB 08 78 0B 00 00 00 00-00 00 00 00 00 00 00 00 00  ..x.....
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  AB 08 78 0B CD 02 01 00-00 00 00 00 00 00 00 00 00  ..x.....
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  .....
-
```

Output explanation: 08AB + 0B78 = -2CD (01 carry indicates negative)

Result:

Thus, the assembly program for 16-bit subtractor is written and executed.

2 c) 16-bit Multiplication

Algorithm:

- a) Assign 0000h to ax register
- b) Load contents of memory location opr1 in register ax
- c) Load contents of memory location opr2 in register bx
- d) mul ax and bx content, store the result in dxax
- e) Load contents of memory location dx in result1
- f) Load contents of memory location ax in result2
- g) Load content 4ch termination code to ah register
- h) Stops execution of the program

Program:

Program	Comments
start: mov ax,data	Transferring the data from memory location data to ax
mov ds,ax	Transferring the data from memory location ax to da
mov ax,0000h	Load the data 0000h in to memory location of ax
mov ax,opr1	Transferring the data from memory location opr1 to ax
mov bx,opr2	Transferring the data from memory location opr2 to bx
mul bx	Multiplication DXAX = AX x BX,
Mov result1, dx	dx->result1; first 16bit of the result
Mov result2, ax	ax->result2; second 16bit of the result
mov ah,4ch	Transferring the termination code 4ch to ah
int 21h	Termination
code ends	Code ends

Unassembled code:

```
D:\>debug 16BITMUL.EXE
-u
076B:0100 BB6A07      MOV     AX,076A
076B:0103 8ED8        MOV     DS,AX
076B:0105 B80000        MOV     AX,0000
076B:0108 A10000        MOV     AX,[0000]
076B:010B 8B1E0200      MOV     BX,[0002]
076B:010F F7E3        MUL     BX
076B:0111 89160400      MOV     [0004],DX
076B:0115 A30600        MOV     [0006],AX
076B:0118 B44C        MOV     AH,4C
076B:011A CD21        INT     21
076B:011C FF7701      PUSH    [BX+01]
076B:011F 40          INC     AX
-
```

Snapshot of sample input and output:

- Input and output:

```
-e 076a:0000
076A:0000  11.ab  00.08  99.78  00.0b

-d 076a:0000
076A:0000  AB 08 78 0B 00 00 00 00-00 00 00 00 00 00 00 00 00  ..x.....
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  AB 08 78 0B 63 00 28 69-00 00 00 00 00 00 00 00 00  ..x.c.(i.....
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  .....
-
```

Output explanation: 08AB * 0B78 = 0063 6928

Result:

Thus, the assembly program for 16-bit multiplier is written and executed.

2 d) 16-bit Division

Algorithm:

- a) Assign 0000h to ax register
- b) Assign 0000h to dx register
- c) Load contents of memory location opr1 in register ax
- d) Load contents of memory location opr2 in register bx
- e) div dxax and bx content, store the result in bx
- f) Load contents of memory location ax in quotient
- g) Load contents of memory location dx in remainder
- h) Load content 4ch termination code to ah register
- i) Stops execution of the program

Program:

Program	Comments
start: mov ax,data	Transferring the data from memory location data to ax
mov ds,ax	Transferring the data from memory location ax to da
mov ax,0000h	Load the data 0000h in to memory location of ax
mov dx,0000h	Load the data 0000h in to memory location of dx
mov ax,opr1	Transferring the data from memory location opr1 to ax
mov bx,opr2	Transferring the data from memory location opr2 to bx
div bx	Division DXAX / BX
mov quotient,ax	Transferring the data from memory location ax to quotient
mov remainder,dx	Transferring the data from memory location dx to remainder
mov ah,4ch	Transferring the termination code 4ch to ah
int 21h	Termination
code ends	Code ends

Unassembled code:

```
D:\>debug 16BITDIV.EXE
-u
076B:0100 B86A07      MOV     AX,076A
076B:0103 8ED8        MOV     DS,AX
076B:0105 B80000      MOV     AX,0000
076B:0108 BA0000      MOV     DX,0000
076B:010B A10000      MOV     AX,[0000]
076B:010E 8B1E0200     MOV     BX,[0002]
076B:0112 F7F3        DIV     BX
076B:0114 A30400      MOV     [0004],AX
076B:0117 89160600     MOV     [0006],DX
076B:011B B44C        MOV     AH,4C
076B:011D CD21        INT     21
076B:011F 40          INC     AX
-
```

Snapshot of sample input and output:

- Input and output:

```
-e 076a:0000
076A:0000  11.ab  00.08  99.78  00.0b

-d 076a:0000
076A:0000  AB 08 78 0B 00 00 00 00-00 00 00 00 00 00 00 00 00  ..X.....
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  AB 08 78 0B 00 00 AB 08-00 00 00 00 00 00 00 00 00  ..X.....
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  .....
```

Output explanation: 08AB / 0B78 => Quotient = 0000, Remainder = 08AB

Result:

Thus, the assembly program for 16-bit divider is written and executed.