UCS1512 Microprocessors Lab

Experiment 2: 16-bit Arithmetic Operations

Date: 27-08-2020

Name: Nivedhitha D

Register Number: 185001104

Aim

To write programs for performing 16-bit arithmetic operations in an 8086 microprocessor using MASM and DOSBox.

Algorithm

- 1. Define the values in the data segment and assign the initial values if required
- 2. Initialize the data segment register with a data segment address
- 3. Load the data into the appropriate registers and perform addition/ subtraction/ multiplication/ division and store the sum/ difference/product/quotient-remainder to the result address for display
- 4. Terminate the program

Program for adding two 16-bit numbers

Program Name: 16BITADD.ASM

Program	Comments						
;Program for adding two 16-bit numbers	Naming the CS and DS for the program						
ASSUME CS:CODE,DS:DATA							
DATA SEGMENT OPR1 DW 1111H OPR2 DW 2219H	Declaring and initializing the values for the two operands in the data segment						
RESULT DW 0000H CARRY DB 00H DATA ENDS	The carry and result must be set to zero to obtain the correct result and get rid off garbage values						
CODE SEGMENT ORG 0100H	Providing an offset value						
START: MOV AX, DATA MOV DS, AX	Initializing the data segment register with the data segment address						
MOV AX, OPR1 MOV BX, OPR2 MOV CH, 00H	Loading the data into the appropriate registers						
ADD AX, BX	Performing addition: $AX = AX + BX$						
JNC HERE	If the carry is zero, go-to label, HERE						
INC CH	If carry has occurred, increment register, CH						
HERE: MOV RESULT, AX MOV CARRY, CH	Loading the sum and carry into the appropriate locations for display						
MOV AH, 4CH INT 21H CODE ENDS END START	Calling the DOS Function to enter the display screen using interrupt 21H and to terminate the program						

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
                                                                                    X
D:\>link 16bitadd.obj;
Microsoft Object Linker V2.01 (Large)
(C) Copyright 1982, 1983 by Microsoft Inc.
Warning: No STACK segment
There was 1 error detected.
D:\>debug 16bitadd.exe
076B:0100 B86A07
                          MOV
                                   AX,076A
076B:0103 8ED8
                          MOV
                                   DS, AX
076B:0105 A10000
                          MOV
                                   AX,[0000]
076B:0108 8B1E0200
                          MOV
                                   BX,[0002]
076B:010C B500
                          MOV
                                   CH,00
076B:010E 03C3
                          ADD
                                   AX, BX
076B:0110 7302
                          JNB
                                   0114
076B:0112 FEC5
                          INC
                                   CH
076B:0114 A30400
                          MOV
                                   [0004],AX
076B:0117 882E0600
                          MOV
                                   [00061,CH
076B:011B B44C
                          MOV
                                   AH,4C
076B:011D CD21
                          INT
                                   21
076B:011F 40
                          INC
                                   AX
```

Snapshot of Input and Output

Input

Two 16-bit values

Output

Sum in two memory locations and the carry in another location

Without carry

B DOSBox	0.7	4-3,	Ср	u sp	eed	l:	300	0 cycle:	s, Fr	ame	eskij	0,	Pro	gra		000		×
076B:011B	B440)			M)Ų		AH,4C										
076B:011D	CD2:	L			I	T		21										
076B:011F	40				I	1C		AX										
-d 076a:00	00																	
076A:0000	11	11	19	22	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			
076A:0030	00	00	∞	00	00	00	∞	00-00	∞	00	00	∞	00	∞	00			
076A:0040	00	00	∞					00-00				∞	∞	∞	00			
076A:0050	00	00	00		00			00-00						00				
076A:0060	00	00						00-00			00				00			
076A:0070	00	00	00	00	00	00	00	00-00	00	00	00	00	00	$\Theta\Theta$	00			
-g																		
Program te	rmi	nate	ed 1	norn	na l l	ly												
-d 076a:00	00																	
076A:0000	11	11	19	22	2A	33	00	00-00	∞	00	00	00	00	00	00	"*	3	
076A:0010	00	00	∞					00-00							00			
076A:0020	00	00	∞	00	∞	∞	00	00-00	∞	00	00	∞	00	∞	00			
076A:0030	00	∞	∞	00	∞	∞	∞	00-00	∞	00	00	∞	00	∞	00			
076A:0040	00	00	∞	∞	00			00-00			∞	∞	∞	∞	00			
076A:0050	00	00	00	00	00			00-00							00			
076A:0060	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			

With carry

₩ DOSBox	0.74	4-3,	Ср	u sp	eed	l:	300	0 су	cles	s, Fra	ame	eskip	o 0,	Pro	gra		·	×
·q																		
	7.837	245	202															
: \> debug		tac	ld . e	exe														
d 076a:00		363	12.027	2727	200	2727	2020	2020	2020	2020	2020	20120	2720	202	27.20	2020		
76A:0000								00-									" .	
76A:0010								00-										
76A:0020				00				00-						00				
76A:0030					00			00-						00				
76A:0040		00	00		00	00		00-						00				
6A:0050					00			00-						00				
76A:0060				00				00-						00				
76A:0070		∞	00	00	00	00	00	00-	-00	00	00	00	00	00	00	00		
976a:00																		
76A:0000	11.	.FF	1	11.I	E	19	9.96	ì	22.	FF								
1 076a:00	00																	
76A:0000	FF	FE	9A	FF	00	00	00	00-	-00	00	00	00	00	00	00	00		
6A:0010	00	00	00	00	00	00	00	00-	-00	00	00	00	00	00	00	00		
6A:0020	00	$\Theta\Theta$	00	00	00	$\Theta\Theta$	$\Theta\Theta$	00-	-00	00	$\Theta\Theta$	$\Theta\Theta$	$\Theta\Theta$	00	00	00		
6A:0030	00	00	00	00	00	00	00	00-	-00	00	00	00	00	00	00	00		
6A:0040	00	$\Theta\Theta$	00	$\Theta\Theta$	00	$\Theta\Theta$	$\Theta\Theta$	00-	-00	$\Theta\Theta$	$\Theta\Theta$	$\Theta\Theta$	$\Theta\Theta$	00	00	00		
6A:0050	00	00	00	00	00	00	00	00-	-00	00	00	00	00	00	00	00		
6A:0060	00	00	00	00	00	00	00	00-	-00	00	00	00	00	00	00	00		
76A:0070	00	00	00	00	00	00	00	00-	-00	00	00	00	00	00	00	00		
g																		
rogram te d 076a:00		nat	ed	nor	ma 1	ly												
76A:0000		FE	9A	FF	99	FE	01	00-	-00	00	00	00	00	00	00	00		
76A:0010	00			00				00-							00	00		
76A:0020	00			00											00	00		
76A:0030								00-								00		
00000	~	\sim	\sim	\sim	\sim	\sim	\sim	~	\sim	~	\sim	~	~	~	~	~		

·q

D:\>

Program for subtracting two 16-bit numbers

Program Name: 16BITSUB.ASM

Program	Comments
;Program for subtracting two 16-bit numbers	Naming the CS and DS for the program
ASSUME CS:CODE,DS:DATA	
DATA SEGMENT OPR1 DW 1211H OPR2 DW 3599H	Declaring and initializing the values for the two operands in the data segment
DIFF DW 0000H SIGN DB 00H DATA ENDS	The difference and sign must be set to zero to obtain the correct result and get rid off garbage values
CODE SEGMENT ORG 0100H	Providing an offset value
START: MOV AX, DATA MOV DS, AX	Initializing the data segment register with the data segment address
MOV AX, OPR1 MOV BX, OPR2 MOV CH, 00H	Loading the data into the appropriate registers
SUB AX, BX	Performing subtraction: $AX = AX - BX$
JNC HERE	If the carry is zero, go-to label, HERE
NEG AX INC CH	If carry has occurred, increment register, CH to denote sign & 2's complement of difference
HERE: MOV DIFF, AX MOV SIGN, CH	Loading the difference and sign into the appropriate locations for display
MOV AH, 4CH INT 21H CODE ENDS END START	Calling the DOS Function to enter the display screen using interrupt 21H and to terminate the program

```
D:\>debug 16bitsub.exe
076B:0100 B86A07
                         MOV
                                 AX,076A
076B:0103 8ED8
                         MOV
                                 DS,AX
076B:0105 A10000
                                 AX,[0000]
                         MOV
076B:0108 8B1E0200
                         MOV
                                 BX,[0002]
076B:010C B500
                         MOV
                                 CH,00
076B:010E 2BC3
                                 AX, BX
                         SUB
076B:0110 7304
                         JNB
                                 0116
076B:0112 F7D8
                         NEG
                                 AX
076B:0114 FEC5
                         INC
                                 CH
076B:0116 A30400
                         MOV
                                 [0004],AX
076B:0119 882E0600
                                 [00061,CH
                         MOV
076B:011D B44C
                         MOV
                                 AH,4C
076B:011F CD21
                         INT
                                 21
```

Snapshot of Input and Output

Input

Two 16-bit values

Output

Difference in two memory locations and the indication of the sign in one

(eg: FF-FE = 01, indication of sign is 00 i.e. positive

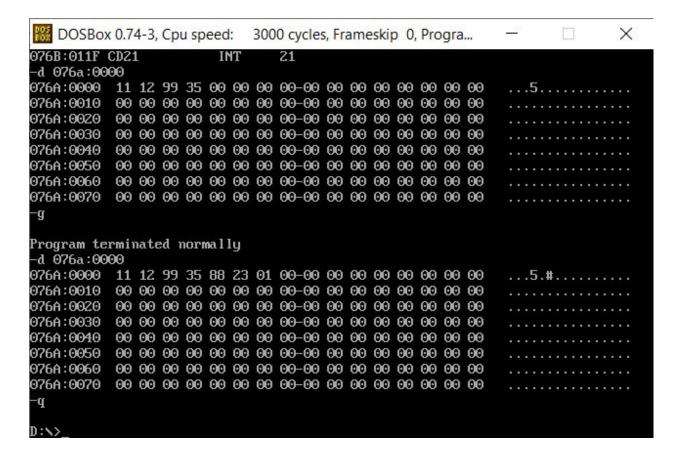
FE - FF = 01, indication of sign is 01 i.e. negative)

Positive Sign

DOSBox	0.7	4-3,	Ср	u sp	eed	:	300	0 cycles	s, Fr	ame	eskij	o 0,	Pro	gra		·—	\times
-q																	
D:\>debug	16ki	tei	ıka	200													
-d 076a:00		LUSE		-^-													
076A:0000		12	99	35	00	99	00	00-00	00	00	00	00	00	00	00	5	
076A:0010								00-00							$\widetilde{00}$		
076A:0020				00				00-00			00				00		
076A:0030		00		00				00-00		00	00	00	00	00	00		
076A:0040		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		
-е 076a:00	00																
076A:0000	11.	FF	1	12.1	FF	99	9.11	1 35	.12								
-d 076a:00	00																
076A:0000		FF	11	12	00	00	00	00-00	00	00	00	00	00	00	00		
076A:0010	7.5	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		

```
Program terminated normally
-d 076a:0000
976A:0000 FF FF 11 12 EE ED 00 00-00 00 00 00 00 00 00 00
076A:0010
   076A:0020
   076A:0030
   076A:0040
   076A:0050
   076A:0060
   076A:0070
```

Negative Sign



Program for multiplying two 16-bit numbers

Program Name: 16BITMUL.ASM

Program	Comments
;Program for multiplying two 16-bit numbers	Naming the CS and DS for the program
ASSUME CS:CODE,DS:DATA	
DATA SEGMENT OPR1 DW 1111H OPR2 DW 4444H	Declaring and initializing the values for the two operands in the data segment
PRODH DW 0000H PRODL DW 0000H DATA ENDS	The higher and lower order bits of the product must be set to zero to obtain the correct result and get rid off garbage values
CODE SEGMENT ORG 0100H	Providing an offset value
START: MOV AX, DATA MOV DS, AX	Initializing the data segment register with the data segment address
MOV AX, OPR1 MOV BX, OPR2	Loading the data into the appropriate registers
MUL BX	Performing multiplication: DXAX = AX * BX
MOV PRODH, DX MOV PRODL, AX	Since the product is stored in the DX & AX register, the higher and lower order bits are extracted
MOV AH, 4CH INT 21H CODE ENDS END START	Calling the DOS Function to enter the display screen using interrupt 21H and to terminate the program

```
BB DOSBox 0.74-3, Cpu speed:
                             3000 cycles, Frameskip 0, Progra...
                                                                               X
D: Nolink 16bitmul.obj;
  Microsoft Object Linker V2.01 (Large)
(C) Copyright 1982, 1983 by Microsoft Inc.
Warning: No STACK segment
There was 1 error detected.
D:N>debug 16bitmul.exe
076B:0100 B86A07
                         MOV
                                 AX,076A
076B:0103 8ED8
                         MOV
                                 DS,AX
976B:0105 A10000
                         MOV
                                 AX,[0000]
                                 BX,[0002]
076B:0108 8B1E0200
                         MOV
076B:010C F7E3
                         MUL
                                 BX
                                 [0004],DX
076B:010E 89160400
                         MOV
076B:0112 A30600
                         MOV
                                 [00061,AX
076B:0115 B44C
                         MOV
                                 AH,4C
076B:0117 CD21
                         INT
                                 21
076B:0119 FD
                         STD
076B:011A 00B0FF77
                         ADD
                                 [BX+SI+77FF1,DH
076B:011E 01408B
                         ADD
                                 [BX+SI-75],AX
```

Snapshot of Input and Output

Input

Two 16-bit values

Output

Product in 32 bits

: \> debug d 076a:00		LTML	11.6	exe													
76A:0000		11	44	44	00	00	00	00-00	00	00	00	00	00	00	00	DD.	
76A:0010	00	00	00	00	00	00	00	00-00	00	00	00	00	00	00	00		
76A:0020	00	00	00	00	00	00	00	00-00	00	00	00	$\Theta\Theta$	$\Theta\Theta$	00	00		
76A:0030	00	00	00	00	00	00	00	00-00	00	00	00	$\Theta\Theta$	00	00	00		
76A:0040	00	00	00	00	00	00	00	00-00	00	00	00	$\Theta\Theta$	$\Theta\Theta$	00	00		
76A:0050	00	00	00	00	00	00	00	00-00	00	00	00	$\Theta\Theta$	00	00	00		
76A:0060	00	00	00	00	00	00	00	00-00	00	00	00	00	00	00	$\Theta\Theta$		
UH - 0000	~																
76A:0070					00			00-00				00	00		00		
76A:0070 (f rogram te 1 076a:00	00 rmir	00	00	00		00						00	00		00		
76A:0070 ; rogram te	00 rmir 00	00 nate	00 ed 1	00 norr	na l l	00 ly	00		00	00	00			00	7979		
76A:0070 (rogram te l 076a:00	00 rmir 00 11	00 nate	00 ed 1	00 norr 44	na 1 1 8D	00 ly 04	00 84	00-00	00 00	00 00	00 00	90	90	00 00	7979		
76A:0070 	00 rmir 00 11 00	00 nate 11 00	00 ed 1 44 00	00 norr 44 00	na 11 8D 00	00 ly 04 00	84 00	90-90 90-90	00 00 00	90 90 90	00 00 00	90 90	90 90	90 90 90	00	DD.	
76A:0070 f rogram te l 076a:00 76A:0000 76A:0010 76A:0020	00 rmir 00 11 00 00	00 nate 11 00 00	60 ed 1 ed	00 norr 44 00 00	na 11 8D 00 00	00 ly 04 00 00	84 90 90	00-00 0C-00 00-00	90 90 90 90	99 99 99 99	99 99 99 99	00 00 00	90 90 90	99 99 99 99	00 00	DD.	
76A:0070 f rogram te l 076a:00 76A:0000 76A:0010 76A:0020 76A:0030	00 rmir 00 11 00 00	00 nate 11 00 00	60 ed 1 44 60 60 60 60	00 norr 44 00 00	na 11 8D 00 00	00 ly 04 00 00	84 60 60 60	00-00 0C-00 00-00 00-00	90 90 90 90 90	90 90 90 90 90	90 90 90 90 90	00 00 00	00 00 00 00	90 90 90 90 90	00 00 00	DD.	
76A:0070 Pogram te 1 076a:00 76A:0000 76A:0010 76A:0020 76A:0030 76A:0040 76A:0050	00 rmir 00 11 00 00 00	00 nate 11 00 00 00 00	60 ed 1 44 00 00 00 00 00 00	00 norr 44 00 00 00 00	8D 00 00 00 00	00 ly 04 00 00 00 00	84 90 90 90 90	00-00 00-00 00-00 00-00 00-00 00-00	90 90 90 90 90 90	90 90 90 90 90	90 90 90 90 90	99 99 99 99	90 90 90 90 90	90 90 90 90 90	90 90 90 90	DD .	
76A:0070 f rogram te l 076a:00 76A:0000 76A:0010 76A:0020 76A:0030	00 rmir 00 11 00 00 00 00	00 nate 11 00 00 00 00	60 ed 1 44 00 00 00 00 00 00	00 norr 44 00 00 00 00	8D 00 00 00 00	00 ly 04 00 00 00 00	84 90 90 90 90	00-00 00-00 00-00 00-00 00-00	90 90 90 90 90 90	90 90 90 90 90	90 90 90 90 90	99 99 99 99	90 90 90 90 90	90 90 90 90 90	00 00 00 00 00	DD.	

Program for dividing two 16-bit numbers

Program Name: 16BITDIV.ASM

Program	Comments
;Program for dividing two 16-bit numbers	Naming the CS and DS for the program
ASSUME CS:CODE,DS:DATA	
DATA SEGMENT OPR1 DW 1111H OPR2 DW 9999H	Declaring and initializing the values for the two operands in the data segment
QUOTIENT DW 0000H REMINDER DW 0000H DATA ENDS	The quotient and remainder must be set to zero to obtain the correct result and get rid off garbage values
CODE SEGMENT ORG 0100H	Providing an offset value
START: MOV AX, DATA MOV DS, AX	Initializing the data segment register with the data segment address
MOV BX, OPR1 MOV AX, OPR2	Loading the data into the appropriate registers
MOV DX, 0000H	Setting the DX bits to zero
DIV BX	Performing division: AX = DXAX / BX
MOV REMINDER, DX MOV QUOTIENT, AX	Since the quotient and remainder are stored in the AX register and DX register respectively, they need to be extracted
MOV AH, 4CH INT 21H CODE ENDS END START	Calling the DOS Function to enter the display screen using interrupt 21H

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
                                                                              X
D:N>link 16bitdiv.obj;
  Microsoft Object Linker V2.01 (Large)
(C) Copyright 1982, 1983 by Microsoft Inc.
Warning: No STACK segment
There was 1 error detected.
D:\>debug 16bitdiv.exe
-u
976B:0100 B86A07
                        MOV
                                 AX,076A
976B:0103 8ED8
                        MOV
                                 DS, AX
976B:0105 8B1E0000
                        MOV
                                 BX,[0000]
                                 AX,[0002]
976B:0109 A10200
                        MOV
976B:010C BA0000
                                 DX,0000
                        MOV
976B:010F F7F3
                        DIU
                                 BX
976B:0111 89160600
                        MOV
                                 [00061,DX
976B:0115 A30400
                                 [0004],AX
                        MOV
                                 AH,4C
976B:0118 B44C
                        MOV
976B:011A CD21
                        INT
                                 21
                                 [BX+01]
976B:011C FF7701
                        PUSH
976B:011F 40
                         INC
                                 AX
```

Snapshot of Input and Output

Input

Two 16-bit values(**Note:** No dedicated instruction available in 8086 to perform 16 bit / 16 bit)

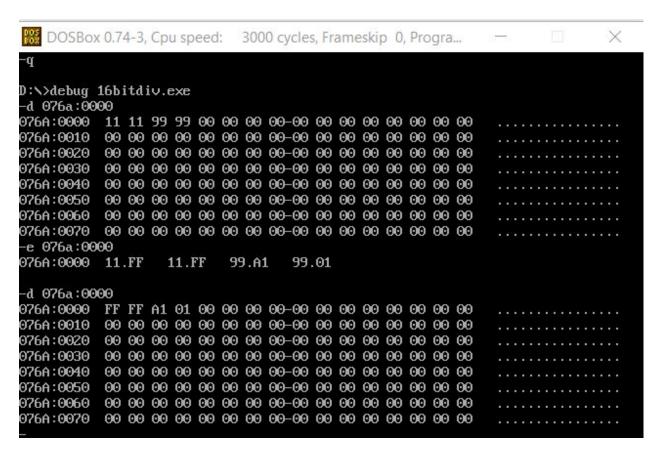
Output

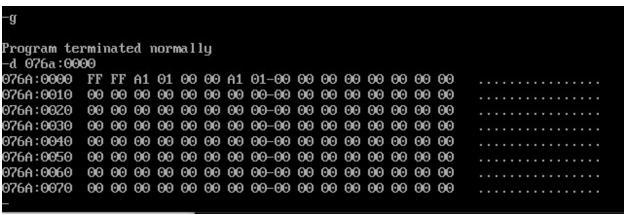
Quotient in two memory locations and the remainder in another two

Without reminder

```
D:\>debug 16bitdiv.exe
-d 076a:0000
076A:0000
   11 11 99 99 00 00 00 00-00 00 00 00 00 00 00 00
076A:0010
   076A:0020
976A:0030
   976A:0040
   076A:0050
   976A:0060
   976A:0070
   -g
Program terminated normally
-d 076a:0000
976A:0000
   11 11 99 99 09 00 00 00-00 00 00 00 00 00 00 00
976A:0010
   076A:0020
   076A:0030
   976A:0040
   076A:0050
076A:0060
   076A:0070
```

With reminder





Result

Programs for performing 16-bit arithmetic operations in an 8086 microprocessor using MASM and DOSBox were implemented and the outputs were verified.