SSN College of Engineering
Department of Computer Science and Engineering
UCS1512 – Microprocessors Lab
16 BIT ARITHMETIC OPERATIONS

Name: Prasanna Kumaran D
Registration Number: 185001110
Semester: V

01 September 2020

1 AIM:

To write and execute 8086 programs for arithmetic operations of 16 numbers like addition, subtraction, multiplication and division.

2 PROCEDURE:

- Write the program in a text editor and save it as a ***.asm*** file under the MASM directory.
- Launch DOSBOX application and mount the MASM folder using the command prompt.
- Use the following syntax for mounting : ' **mount [LOCAL DRIVE] FILEPATH** '. Enter into the local drive('LOCAL DRIVE:').
- The code file can be edited using the command **edit FILENAME.asm**. Save the changes and exit.
- Assemble the code using the command '**masm FILENAME.asm**' to generate the object file. The object file is in the format 'FILENAME.obj'
- Add dynamic libraries using the syntax **'link FILENAME.obj'** to generate the executeable(**.exe**) file.
- Enter the debug mode using debug FILENAME.exe to execute and analuse the memory contents. The various commands used in debug mode are as follows:-
  - U :- Displays unassembled code.
  - D :- Refers to the offset from which contents in the memory are displayed.
  - E :- Change the value in memory.
  - G :- execute the code.
  - Q :- Quit debug mode.

3 Algorithm & Program

INITIALIZATION:

- Declare and initialize the operands and the code and data segments.

3.1 16 BIT ADDITION:

- Move the data segment to the AX register and then move it to the DS register.
- Move the first operand to AX register and the second to BX register.
- Initialize CH register to 0000h.(hexadecimal)
- Add the operands using **ADD AX, BX**.
- Check for carry bit using JNC instruction.
  - Increment CX by 1 if there is a carry.
- Move the contents of AX and CX to RESULT and CARRY operands respectively.

3.1.1 16 BIT ADDITION: Program

PROGRAM	COMMENTS
mov ax,opr1	Transfers contents of operand 1to AX register.
mov bx,opr2	Transfers contents of operand 2to BX register.
mov cx,0000h	Initialises CH register with 0000h.
add ah,bh	AX = AX + BX.
jnc here	Jumps to „Here Label if carry bit is not generated.
inc cx	CX = CX + 1(Increment CH by 1).
here: mov result,ax	Transfers contents of AX register to RESULT.
mov carry,cx	Transfers contents of CX register to CARRY.
mov ah, 4ch	Move the hexadecimal value 4c to ah
int 21h	When software interrupt 21 is called with AH=45,process is terminated

```
P:\>debug 16bitadd.exe
-u
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 02E7          ADD     AH,BH
076B:0110 7302          JNB     0114
076B:0112 FEC5          INC     CH
076B:0114 A30400        MOV     [0004],AX
076B:0117 882E0600      MOV     [0006],CH
076B:011B B44C          MOV     AH,4C
076B:011D CD21          INT     21
076B:011F 40            INC     AX
```

Figuur 1: 16 bit ADD Unassembled

```
076B:011F 40            INC     AX
-d 076a:0000
076A:0000  11 11 39 92 00 00 00 00-00 00 00 00 00 00 00 00   ..9.....
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  11 11 39 92 11 A3 00 00-00 00 00 00 00 00 00 00   ..9.....
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   ..........
-
```

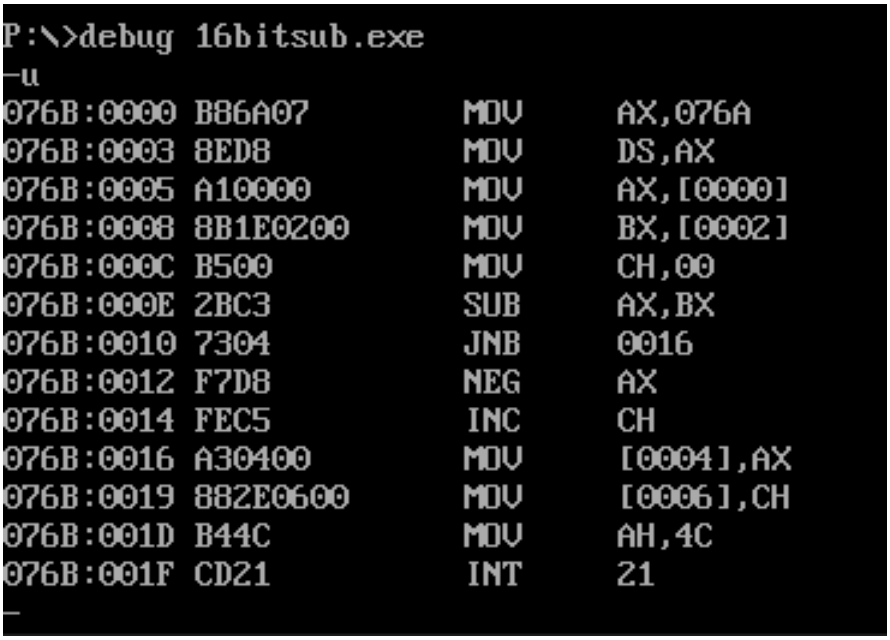
Figuur 2: 16 bit ADD OUTPUT

3.2 16 BIT SUBTRACTION:

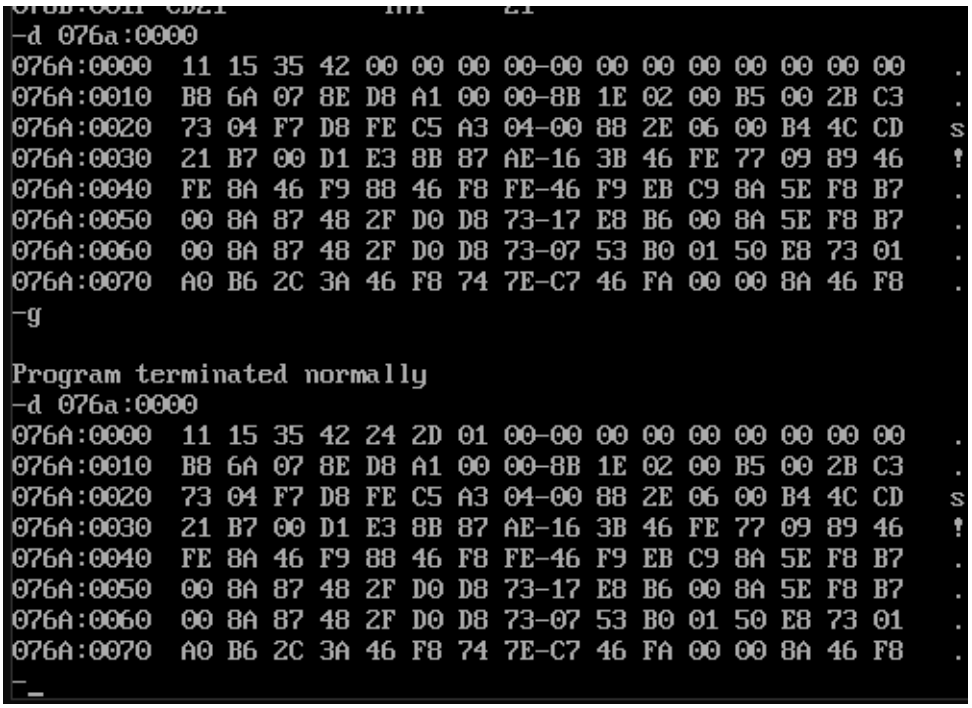
- Move the data segment to AX register and then move it to the DS register.
- Move the first and second operand to AX and BX register respectively.
- Initialize CH register to 0000h.
- Subtract the operands **using SUB AX, BX**.
- Check for carry bit using JNC instruction.
  - If AX ¡ BX carry bit is generated and take 2's complement of AH using **NEG AX** and increment CX by 1.
- Move the contents of AX and CX to RESULT and CARRY operands respectively.

3.2.1 16 BIT SUBTRACTION: Program

Program	**Contents**
mov ax,opr1	Transfers contents of operand 1to AX register.
mov bx,opr2	Transfer contents of operand 2 to BX register.
mov cx,0000h	Intialize CX register with 0000h
sub ax,bx	AX = AX - BX
jnc here	Jumps to "Here Label if carry bit is not generated
neg ax	AX = 2's complement of AH
inc cx	CX = CX + 1
here: mov result,ax	Transfer contents from AX to RESULT
mov carry,cx	Transfer contents from CX to CARRY.
mov ah, 4ch	Move the hexadecimal value 4c to ah
int 21h	When software interrupt 21 is called with AH=45,process is terminated



Figuur 3: 16 bit SUB Unassembled



Figuur 4: 16 bit SUB OUTPUT

3.3 16 BIT MULTIPLICATION:

- Move the data segment to the AX register and then move it to DS register.
- Move the first and second operands to AX and BX registers respectively.
- Multiply the operands using **MUL BX**
  - Note : AX is default operand register for MUL instruction and specifying the second operand register is sufficient.
- The lower and higher order result bits are stored in AX and DX. They are transferred to RESULT1 RESULT2 respectively.

3.3.1 16 BIT MULTIPLICATION: Program

Program	**Contents**
mov al, opr1	Transfers contents of operand 1to AL register
mov bl, opr2	Transfer contents of operand 2 to BL register
mul bx	BX = BX * AX
mov result1,ax	Transfer contents of AX to RESULT1
mov result2,dx	Transfer contents of DX to RESULT2
mov ah, 4ch	Move the hexadecimal value 4c to ah
int 21h	When software interrupt 21 is called with AH=45,process is terminated

```
P:\>debug 16bitmul.exe
-u
076B:0000 B86A07        MOV     AX,076A
076B:0003 8ED8          MOV     DS,AX
076B:0005 BA0000        MOV     DX,0000
076B:0008 A10000        MOV     AX,[0000]
076B:000B 8B1E0200      MOV     BX,[0002]
076B:000F F7E3          MUL     BX
076B:0011 A30400        MOV     [0004],AX
076B:0014 8BC2          MOV     AX,DX
076B:0016 A30600        MOV     [0006],AX
076B:0019 B44C          MOV     AH,4C
076B:001B CD21          INT     21
076B:001D 1E            PUSH    DS
076B:001E 8A5EF9        MOV     BL,[BP-07]
-
```

Figuur 5: 16 bit MUL Unassembled

```
-d 076a:0000
076A:0000  00 15 00 30 00 00 00 00-00 00 00 00 00 00 00 00
076A:0010  B8 6A 07 8E D8 BA 00 00-A1 00 00 8B 1E 02 00 F7
076A:0020  E3 A3 04 00 8B C2 A3 06-00 B4 4C CD 21 1E 8A 5E
076A:0030  F9 B7 00 D1 E3 8B 87 AE-16 3B 46 FE 77 09 89 46
076A:0040  FE 8A 46 F9 88 46 F8 FE-46 F9 EB C9 8A 5E F8 B7
076A:0050  00 8A 87 48 2F D0 D8 73-17 E8 B6 00 8A 5E F8 B7
076A:0060  00 8A 87 48 2F D0 D8 73-07 53 B0 01 50 E8 73 01
076A:0070  A0 B6 2C 3A 46 F8 74 7E-C7 46 FA 00 00 8A 46 F8
-g

Program terminated normally
-d 076a:0000
076A:0000  00 15 00 30 00 00 F0 03-00 00 00 00 00 00 00 00
076A:0010  B8 6A 07 8E D8 BA 00 00-A1 00 00 8B 1E 02 00 F7
076A:0020  E3 A3 04 00 8B C2 A3 06-00 B4 4C CD 21 1E 8A 5E
076A:0030  F9 B7 00 D1 E3 8B 87 AE-16 3B 46 FE 77 09 89 46
076A:0040  FE 8A 46 F9 88 46 F8 FE-46 F9 EB C9 8A 5E F8 B7
076A:0050  00 8A 87 48 2F D0 D8 73-17 E8 B6 00 8A 5E F8 B7
076A:0060  00 8A 87 48 2F D0 D8 73-07 53 B0 01 50 E8 73 01
076A:0070  A0 B6 2C 3A 46 F8 74 7E-C7 46 FA 00 00 8A 46 F8
```

Figuur 6: 16 bit MUL OUTPUT

3.4 16 BIT DIVISION

- Move the data segment to the AX register and then move it to the DS register.
- Now, set DX register to 0000h and move first operand to AX register.(Since we cant directly divide a 16 bit number by 16 bit number in 8086, we now make our dividend 32 bit by storing 0000h in DX register and the 16-bit operand 1 in AX register).
- Move the second operand to the BX register.
- Now divide using DIV BX.(It will perform DXAX / BX. what actually happens is the division of a 32 bit number by a 16 bit number.)
- The quotient and remainder stored in AL and AH should be moved to QUOTIENT and REMAINDER respectively.

3.4.1 16 BIT DIVISION: Program

Program	**Contents**
mov dx, 0000h	Move the value 0000h to DX register.
mov ax, opr1	Transfers contents of operand 1to AX register.
mov bx, opr2	Transfer contents of operand 2 to BX register.
div bx	Performs DXAX/BX.
mov quotient,ax	Transfer contents from AX to QUOTIENT
mov remainder,dx	Transfer contents from DX to REMAINDER
mov ah, 4ch	Move the hexadecimal value 4c to ah
int 21h	When software interrupt 21 is called with AH=45,process is terminated

```
P:\>debug 16bitdiv.exe
-u
076B:0000 B86A07        MOV     AX,076A
076B:0003 8ED8          MOV     DS,AX
076B:0005 BA0000        MOV     DX,0000
076B:0008 A10000        MOV     AX,[0000]
076B:000B 8B1E0200      MOV     BX,[0002]
076B:000F F7F3          DIV     BX
076B:0011 A30400        MOV     [0004],AX
076B:0014 89160600      MOV     [0006],DX
076B:0018 B44C          MOV     AH,4C
076B:001A CD21          INT     21
076B:001C EB1E          JMP     003C
076B:001E 8A5EF9        MOV     BL,[BP-07]
-
```

Figuur 7: 16 bit DIV Unassembled

```
-d 076a:0000
076A:0000  00 25 20 10 00 00 00 00-00 00 00 00 00 00 00 00
076A:0010  B8 6A 07 8E D8 BA 00 00-A1 00 00 8B 1E 02 00 F7
076A:0020  F3 A3 04 00 89 16 06 00-B4 4C CD 21 EB 1E 8A 5E
076A:0030  F9 B7 00 D1 E3 8B 87 AE-16 3B 46 FE 77 09 89 46
076A:0040  FE 8A 46 F9 88 46 F8 FE-46 F9 EB C9 8A 5E F8 B7
076A:0050  00 8A 87 48 2F D0 D8 73-17 E8 B6 00 8A 5E F8 B7
076A:0060  00 8A 87 48 2F D0 D8 73-07 53 B0 01 50 E8 73 01
076A:0070  A0 B6 2C 3A 46 F8 74 7E-C7 46 FA 00 00 8A 46 F8
-g

Program terminated normally
-d 076a:0000
076A:0000  00 25 20 10 02 00 C0 04-00 00 00 00 00 00 00 00
076A:0010  B8 6A 07 8E D8 BA 00 00-A1 00 00 8B 1E 02 00 F7
076A:0020  F3 A3 04 00 89 16 06 00-B4 4C CD 21 EB 1E 8A 5E
076A:0030  F9 B7 00 D1 E3 8B 87 AE-16 3B 46 FE 77 09 89 46
076A:0040  FE 8A 46 F9 88 46 F8 FE-46 F9 EB C9 8A 5E F8 B7
076A:0050  00 8A 87 48 2F D0 D8 73-17 E8 B6 00 8A 5E F8 B7
076A:0060  00 8A 87 48 2F D0 D8 73-07 53 B0 01 50 E8 73 01
076A:0070  A0 B6 2C 3A 46 F8 74 7E-C7 46 FA 00 00 8A 46 F8
-
```

Figuur 8: 16 bit DIV OUTPUT

4 RESULT:

Thus,8086 programs for arithmetic operations of 16 bit numbers like addition, subtraction, multiplication and division have been executed successfully using MS - DOSBox