SSN College of Engineering
Department of Computer Science and Engineering
UCS1512 – Microprocessors Lab
Matrix Operations

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

22 September 2020

1 AIM:

To write and execute 8086 programs for performing matrix addition and subtraction.

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.

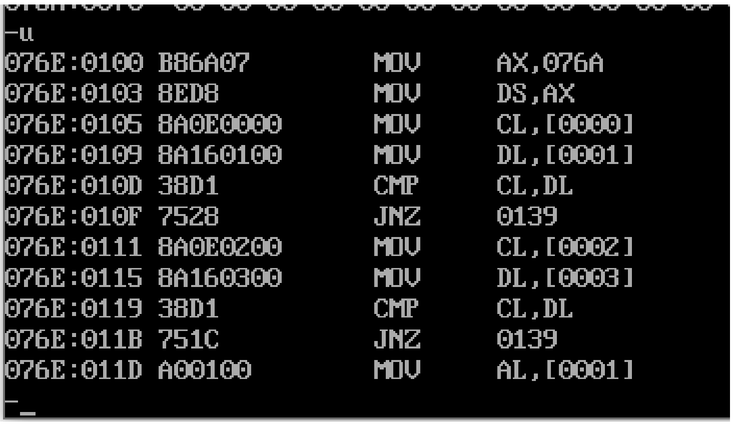
3.1 Matrix addition:

To add corresponding elements of 2 matrices.

1. Initialize data and code segment and variables
2. Move the starting address of data segment to DS
3. Move the values of the number of rows in the two matrices to CL and DL registers
4. Compare if DL and CL are equal
    - if not equal terminate the program else continue
5. Perform step(3) with the number of columns
6. Calculate the number of addition operations to perform by multiplying number of rows and columns of a matrix
7. Move the starting addresses of matrices and result to SI, DI and BX registers
8. Loop while there are operations to be performed
    (a) Move the content which the SI register points to AL register
    (b) ADD AL and [DI] where DI points to the current address in the second matrix
    (c) Move the result of AL to the current address pointed by the BX register
    (d) Increment SI, DI and BX (Move the pointer to the next memory address)
9. Terminate the program

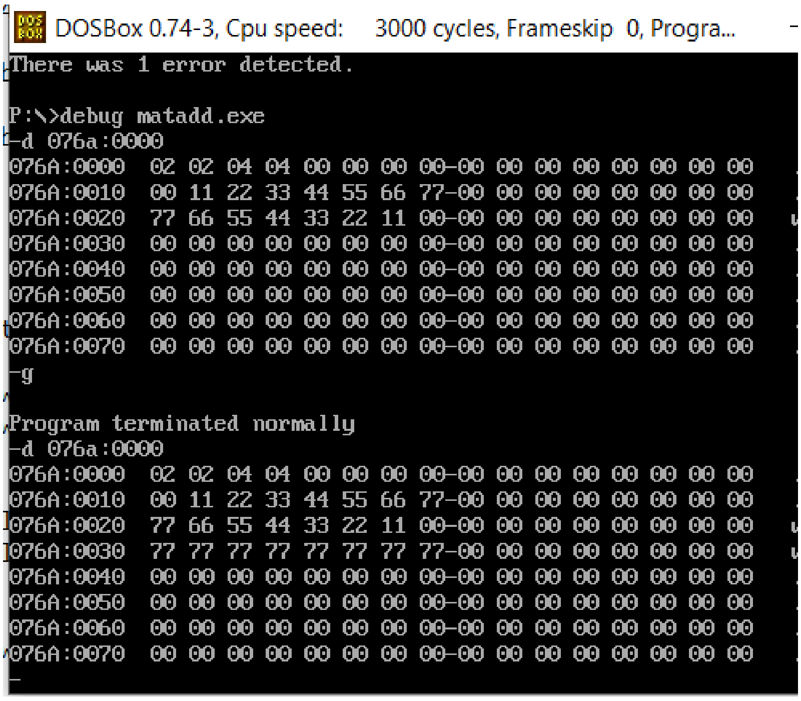
3.1.1 Matrix addition: Program

Program	Comments
assume cs:code, ds:data	
data segment	
row1 db 02h	Initialize data segment and variables
row2 db 02h	
col1 db 04h	
col2 db 04h	
org 0010h	
matrix1 db 00h,11h,22h,33h,	
44h,55h,66h,77h	
org 0020h	
matrix2 db 77h,66h,55h,44h,	
33h,22h,11h,00h	
org 0030h	
result db 8 DUP(0)	End data segment
data ends	
code segment	Initialize code segment
org 0100h	
start: mov ax,data	
mov ds,ax	Transfer address of data segment to DS
mov cl,row1	Transfer contents of row1 and row2 to CL and DL registers
mov dl,row2	Compare CL and DL, if not equal terminate
cmp cl,dl	
jne last	Transfer contents of column1 and column2 to CL and DL registers
mov cl,col1	Compare CL and DL, if not equal terminate
mov dl,col2	
cmp cl,dl	
jne last	Multiply CL with AL and store result in AX
mov al,row2	
mul cl	
mov cx,ax	Move the starting addresses of matrix1, matrix2
mov si, offset matrix1	and result to SI, DI, and BX registers
mov di, offset matrix2	
mov bx, offset result	
here:	
mov al, [si]	Move the content of the address currently pointed by the SI register
add al, [di]	ADD AL with the content of the address currently pointed by DI register
mov [bx], al	Move the result to the address currently pointed by the BX register
inc si	Increment SI, DI, and BL register to the next memory location
inc di	
inc bl	
loop here	
last:	
mov ah,4ch	Termination of execution
int 21h	
code ends	End of the code segment
end start	Terminate program



```
-u
076E:0100 B86A07        MOV     AX,076A
076E:0103 8ED8          MOV     DS,AX
076E:0105 8A0E0000      MOV     CL,[0000]
076E:0109 8A160100      MOV     DL,[0001]
076E:010D 38D1          CMP     CL,DL
076E:010F 7528          JNZ     0139
076E:0111 8A0E0200      MOV     CL,[0002]
076E:0115 8A160300      MOV     DL,[0003]
076E:0119 38D1          CMP     CL,DL
076E:011B 751C          JNZ     0139
076E:011D A00100        MOV     AL,[0001]
-_
```

Figure 1: Matrix addition - unassembled



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

```
There was 1 error detected.

P:\>debug matadd.exe
-d 076a:0000
076A:0000  02 02 04 04 00 00 00 00-00 00 00 00 00 00 00 00
076A:0010  00 11 22 33 44 55 66 77-00 00 00 00 00 00 00 00
076A:0020  77 66 55 44 33 22 11 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  02 02 04 04 00 00 00 00-00 00 00 00 00 00 00 00
076A:0010  00 11 22 33 44 55 66 77-00 00 00 00 00 00 00 00
076A:0020  77 66 55 44 33 22 11 00-00 00 00 00 00 00 00 00
076A:0030  77 77 77 77 77 77 77 77-00 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
-
```

Figure 2: Matrix addition - Output

3.2 Matrix subtraction:

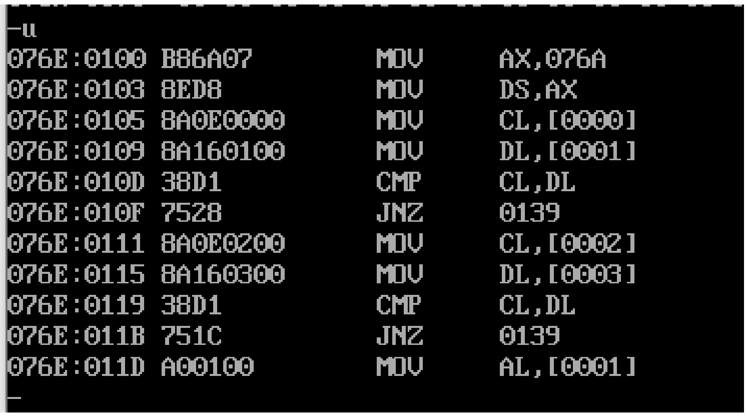
To subtract corresponding elements of 2 matrices.

1. Initialize data and code segment and variables
2. Move the starting address of data segment to DS
3. Move the values of the number of rows in the two matrices to CL and DL registers
4. Compare if DL and CL are equal
   - if not equal terminate the program else continue
5. Perform step(3) with the number of columns
6. Calculate the number of addition operations to perform by multiplying number of rows and columns of a matrix

7. Move the starting addresses of matrices and result to SI, DI and BX registers
8. Loop while there are operations to be performed
   (a) Move the content which the SI register points to AL register
   (b) Subtract AL and [DI] where DI points to the current address in the second matrix
   (c) Move the result of AL to the current address pointed by the BX register
   (d) Increment SI, DI and BX (Move the pointer to the next memory address)
9. Terminate the program

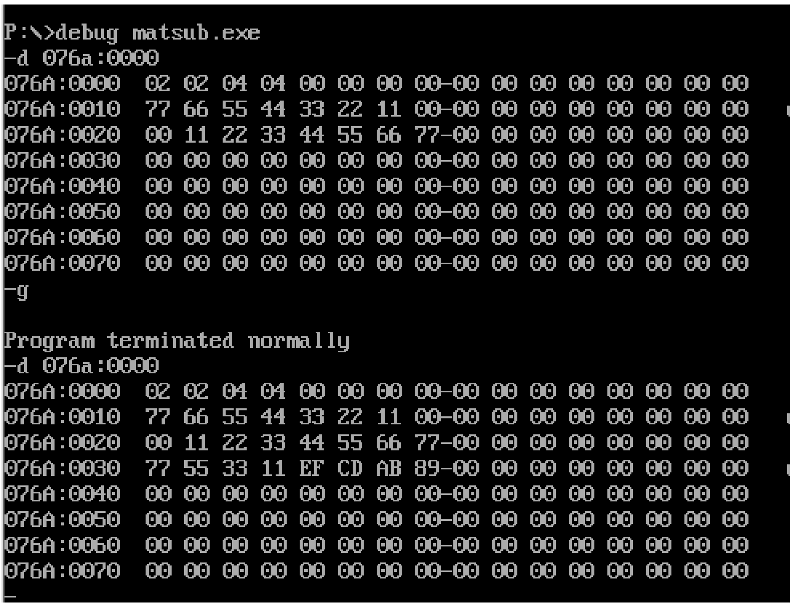
3.2.1 Matrix subtraction: Program

Program	**Comments**
assume cs:code, ds:data	
data segment	
row1 db 02h	Initialize data segment and variables
row2 db 02h	
col1 db 04h	
col2 db 04h	
org 0010h	
matrix1 db 77h,66h,55h, 44h, 33h,22h,11h,00h	
org 0020h	
matrix2 db 00h,11h,22h,33h, 44h,55h,66h,77h	
org 0030h	
result db 8 DUP(0)	End data segment
data ends	
code segment	Initialize code segment
org 0100h	
start: mov ax,data	
mov ds,ax	Transfer address of data segment to DS
mov cl,row1	Transfer contents of row1 and row2 to CL and DL registers
mov dl,row2	Compare CL and DL, if not equal terminate
cmp cl,dl	
jne last	Transfer contents of column1 and column2 to CL and DL registers
mov cl,col1	Compare CL and DL, if not equal terminate
mov dl,col2	
cmp cl,dl	
jne last	Multiply CL with AL and store result in AX
mov al,row2	
mul cl	
mov cx,ax	Move the starting addresses of matrix1, matrix2 and result to SI, DI, and BX registers
mov si, offset matrix1	
mov di, offset matrix2	
mov bx, offset result	
here:	
mov al, [si]	Move the content of the address currently pointed by the SI register
sub al, [di]	Sub AL with the content of the address currently pointed by DI register
mov [bx], al	Move the result to the address currently pointed by the BX register
inc si	Increment SI, DI, and BL register to the next memory location
inc di	
inc bl	
loop here	
last:	
mov ah,4ch	Termination of execution
int 21h	
code ends	End of the code segment
end start	Terminate program



```
-u
076E:0100 B86A07        MOV     AX,076A
076E:0103 8ED8          MOV     DS,AX
076E:0105 8A0E0000      MOV     CL,[0000]
076E:0109 8A160100      MOV     DL,[0001]
076E:010D 38D1          CMP     CL,DL
076E:010F 7528          JNZ     0139
076E:0111 8A0E0200      MOV     CL,[0002]
076E:0115 8A160300      MOV     DL,[0003]
076E:0119 38D1          CMP     CL,DL
076E:011B 751C          JNZ     0139
076E:011D A00100        MOV     AL,[0001]
-
```

Figure 3: Matrix subtraction - unassembled



```
P:\>debug matsub.exe
-d 076a:0000
076A:0000  02 02 04 04 00 00 00 00-00 00 00 00 00 00 00 00
076A:0010  77 66 55 44 33 22 11 00-00 00 00 00 00 00 00 00
076A:0020  00 11 22 33 44 55 66 77-00 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  02 02 04 04 00 00 00 00-00 00 00 00 00 00 00 00
076A:0010  77 66 55 44 33 22 11 00-00 00 00 00 00 00 00 00
076A:0020  00 11 22 33 44 55 66 77-00 00 00 00 00 00 00 00
076A:0030  77 55 33 11 EF CD AB 89-00 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
-
```

Figure 4: Matrix subtraction - Output

4 RESULT:

Thus, 8086 programs for performing matrix addition and subtraction have been executed successfully using MS - DOSBox.