SSN COLLEGE OF ENGINEERING
Department of Computer Science and Engineering
UCS 1512- MICROPROCESSOR LAB
SEMESTER PRACTICALS

Name: Anirudh. H
Registration Number: 185001019 19th November 2020

QUESTIONS:

Write an ALP using 8086 to perform matrix addition.
Write an ALP using 8051 to find sum of all odd numbers in a list.

ALGORITHM:

MATRIX ADDITION:

1. Initialize data and code segments, 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 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 the 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, BX
9. Terminate the program

CODE:

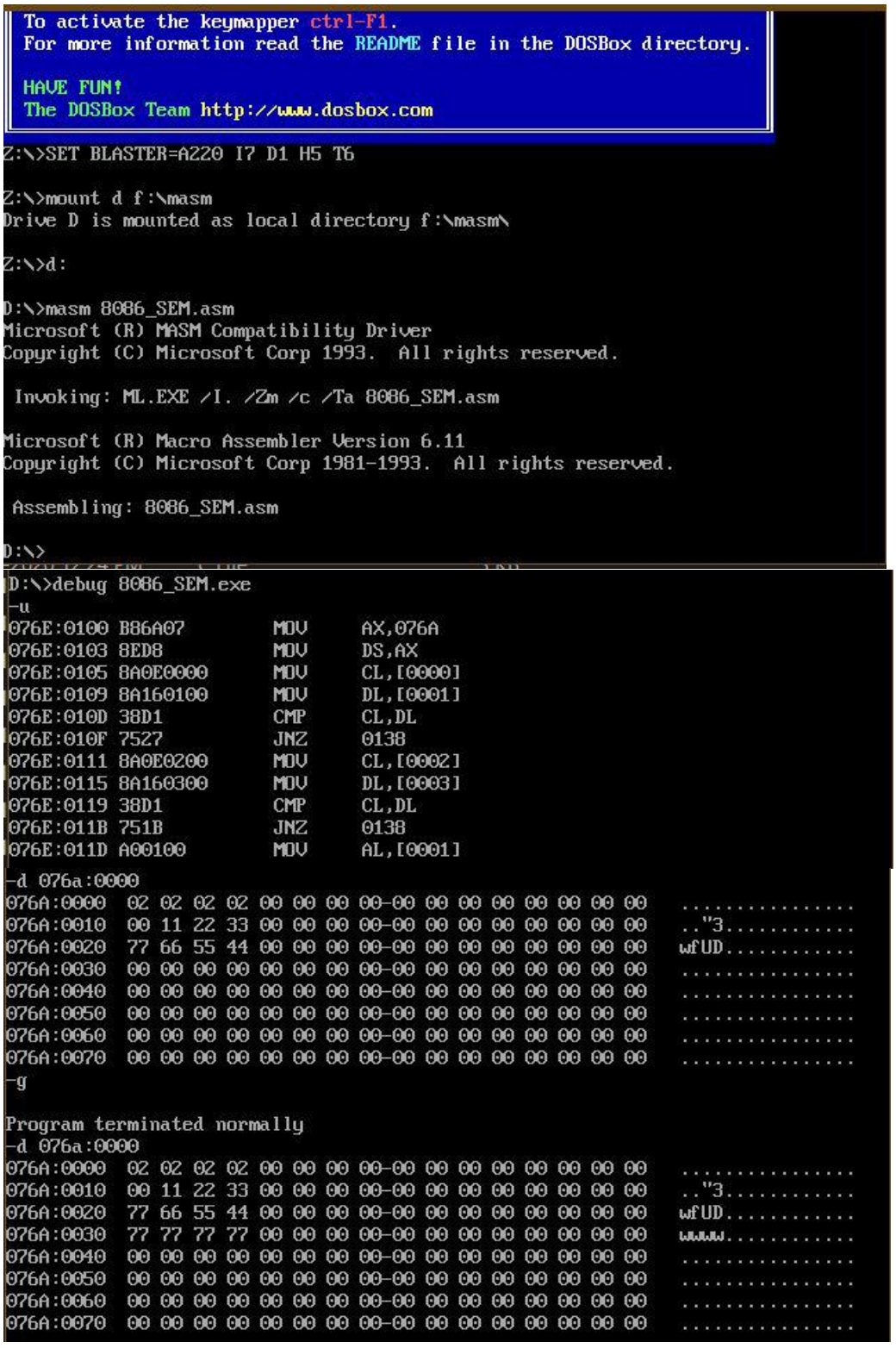
```
assume cs:code, ds:data

data segment
    row1 db 02h
    row2 db 02h
    col1 db 02h
    col2 db 02h
    org 0010h
    matrix1 db 00h, 11h, 22h, 33h
    org 0020h
    matrix2 db 77h, 66h, 55h, 44h
```

```
        org 0030h
        result db 4 DUP(0)
data ends

code segment
        org 0100h
start: mov ax,data
               mov ds, ax
               mov cl, row1
               mov dl, row2
               cmp cl, dl
               jne last
               mov cl, col1
               mov dl, col2
               cmp cl, dl
               jne last
               mov al,row2
               mul cl
               mov cx,ax
               mov si, offset matrix1
               mov di, offset matrix2
               mov bx, offset result
here:     mov al, [si]
               add al, [di]
               mov [bx], al
               inc si
               inc di
               inc bx
               loop here
last:     mov ah, 4ch
               int 21h
code ends
end start
```

OUTPUT:


To activate the keymapper ctrl-F1.
For more information read the README file in the DOSBox directory.
HAVE FUN!
The DOSBox Team http://www.dosbox.com
Z:\>SET BLASTER=A220 I7 D1 H5 T6
Z:\>mount d f:\masm
Drive D is mounted as local directory f:\masm\
Z:\>d:
D:\>masm 8086_SEM.asm
Microsoft (R) MASM Compatibility Driver
Copyright (C) Microsoft Corp 1993. All rights reserved.
Invoking: ML.EXE /I. /Zm /c /Ta 8086_SEM.asm
Microsoft (R) Macro Assembler Version 6.11
Copyright (C) Microsoft Corp 1981-1993. All rights reserved.
Assembling: 8086_SEM.asm
D:\>
D:\>debug 8086_SEM.exe
-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 7527 JNZ 0138
076E:0111 8A0E0200 MOV CL,[0002]
076E:0115 8A160300 MOV DL,[0003]
076E:0119 38D1 CMP CL,DL
076E:011B 751B JNZ 0138
076E:011D A00100 MOV AL,[0001]
-d 076a:0000
076A:0000 02 02 02 02 00 00 00 00-00 00 00 00 00 00 00 00 ....
076A:0010 00 11 22 33 00 00 00 00-00 00 00 00 00 00 00 00 ..."3............
076A:0020 77 66 55 44 00 00 00 00-00 00 00 00 00 00 00 00 wfUD............
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 02 02 00 00 00 00-00 00 00 00 00 00 00 00 ....
076A:0010 00 11 22 33 00 00 00 00-00 00 00 00 00 00 00 00 ..."3............
076A:0020 77 66 55 44 00 00 00 00-00 00 00 00 00 00 00 00 wfUD............
076A:0030 77 77 77 77 00 00 00 00-00 00 00 00 00 00 00 00 wwww............
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 ....

8051 ODD SUM:

CODE AND OUTPUT:

```
mov r0, #010h ;
mov r1, #04h ;
mov r2, #00h ;
mov r5, #00h ;

loop:   mov a,@r0
        mov b,#02
        div ab
        mov r4,b
        cjne r4,#01,here1
        mov a,@r0
        add a,r2
        jnc here
        inc r5
here:   mov r2,a
here1:  inc r0
        djnz r1,loop
        mov r0, #00
halt:   sjmp halt
```