JNIESTRT'S

SMT. INDIRA GANDHI COLLEGE OF ENGINEERING

GHANSOLI, NAVI MUMBAI – 400 709

(Approved by AICTE New Delhi & Govt. of Maharashtra, Affiliated to University of Mumbai)

Computer Engineering Department

ACADEMIC YEAR: - 2023-24 (EVEN SEM)

Experiment No: - 1

Aim: - To perform the addition

Total Marks(10)					Total Marks	DOP	Sign
A	**B**	**C**	**D**	**E**			
2	**3**	**2**	**2**	**1**			

Experiment No: 1.a

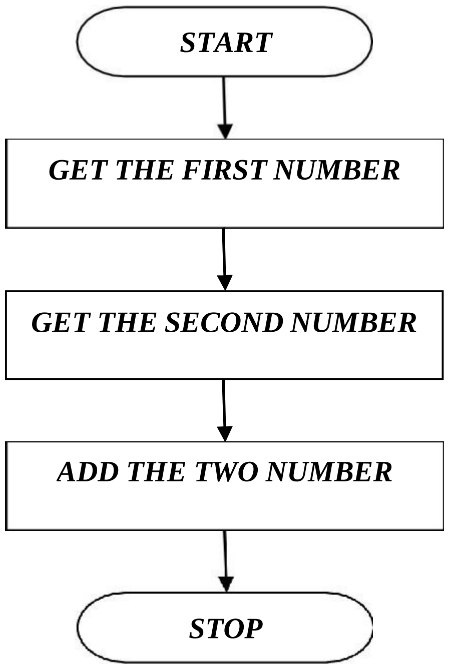
Aim: To perform the 8-bit arithmetic operation

Software: Emulator 8086.

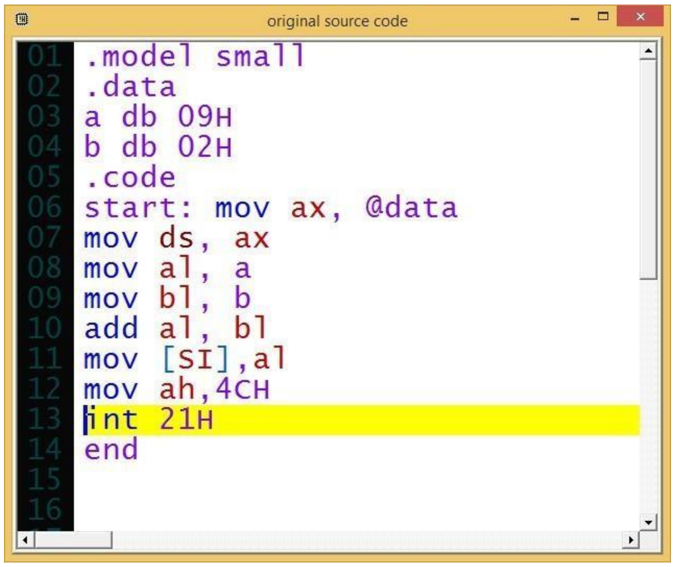
Program:

```
.model small
.data
a db 09H
b db 02H
.code
start: mov ax, @data
mov ds, ax
mov al, a
mov bl, b
add al, bl
mov [SI],al
mov ah,4CH
int 21H
end
```

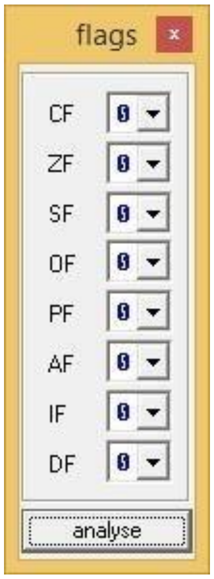
Flowchart:



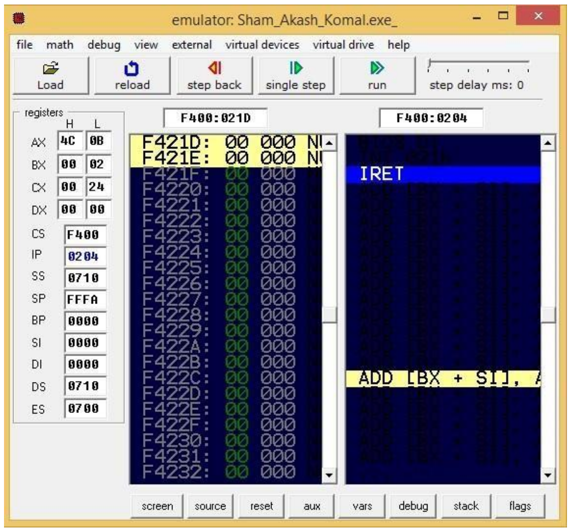
Output:



```
.model small
.data
a db 09H
b db 02H
.code
start: mov ax, @data
mov ds, ax
mov al, a
mov bl, b
add al, bl
mov [SI],al
mov ah,4CH
int 21H
end
```



CF	0
ZF	0
SF	0
OF	0
PF	0
AF	0
IF	0
DF	0



emulator: Sham_Akash_Komal.exe_
file math debug view external virtual devices virtual drive help
Load reload step back single step run step delay ms: 0
screen source reset aux vars debug stack flags

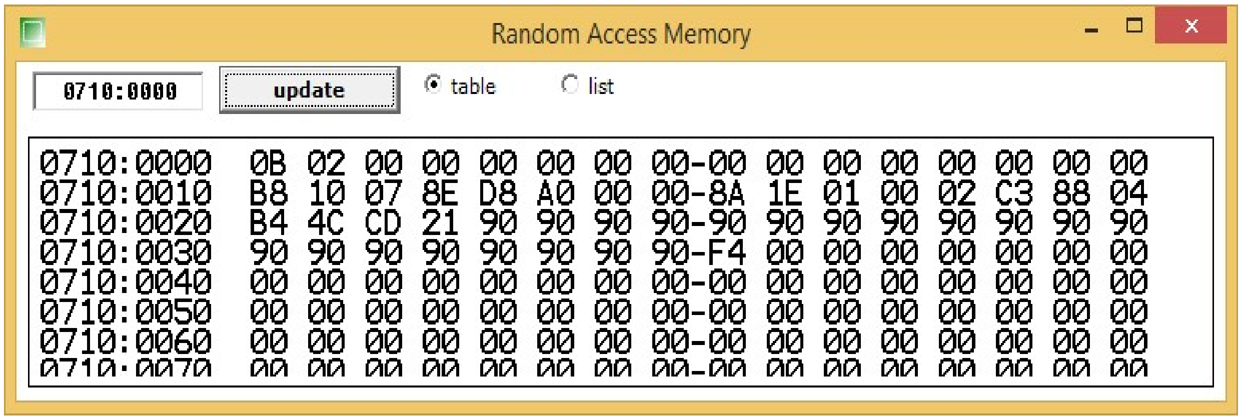
registers

	H	L
AX	4C	0B
BX	00	02
CX	00	24
DX	00	00

CS	F400
IP	0204
SS	0710
SP	FFFA
BP	0000
SI	0000
DI	0000
DS	0710
ES	0700

```
F400:021D
F421D:  00 000 NU
F421E:  00 000 NU
F421F:  00 000
F4220:  00 000
F4221:  00 000
F4222:  00 000
F4223:  00 000
F4224:  00 000
F4225:  00 000
F4226:  00 000
F4227:  00 000
F4228:  00 000
F4229:  00 000
F422A:  00 000
F422B:  00 000
F422C:  00 000
F422D:  00 000
F422E:  00 000
F422F:  00 000
F4230:  00 000
F4231:  00 000
F4232:  00 000
```

```
F400:0204
IRET
ADD [BX + SI],
```



0710:0000

```
0710:0000  0B 02 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0710:0010  B8 10 07 8E D8 A0 00 00-8A 1E 01 00 02 C3 88 04
0710:0020  B4 4C CD 21 90 90 90 90-90 90 90 90 90 90 90 90
0710:0030  90 90 90 90 90 90 90 90-F4 00 00 00 00 00 00 00
0710:0040  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0710:0050  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0710:0060  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0710:0070  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
```

Result:

Thus the assembly language program to perform the 8-bit arithmetic operation has been performed and executed.

Experiment: 1.b

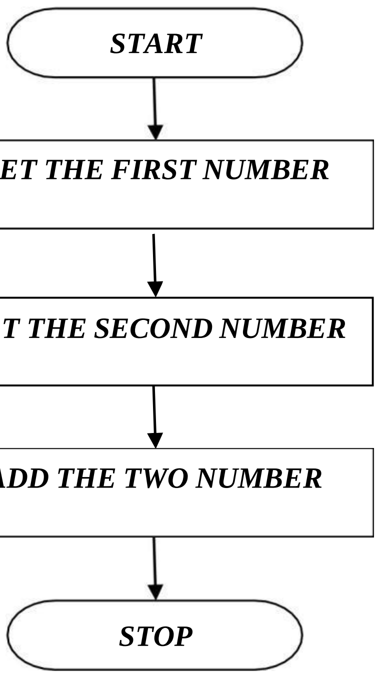
Aim: To perform the addition of 16-bit using assembly language.

Software use: EMU 8086

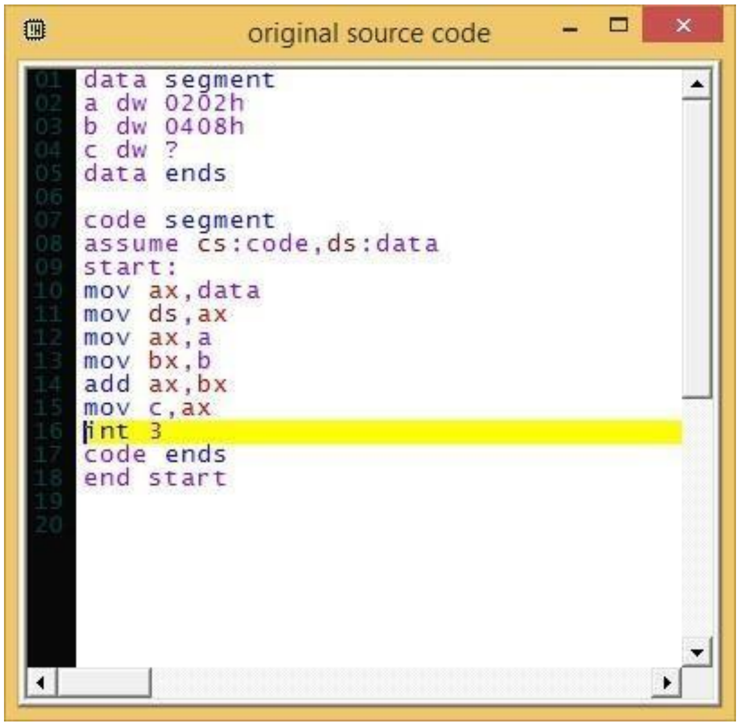
Program:

```
data segment
a dw 0202h
b dw 0408h
c dw ?
data ends

code segment
assume cs:code,ds:data
start:
mov ax,data
mov ds,ax
mov ax,a
mov bx,b
add ax,bx
mov  c,ax
int 3
code ends
end start
```

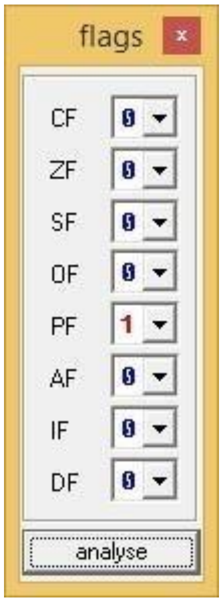
Flowchart:



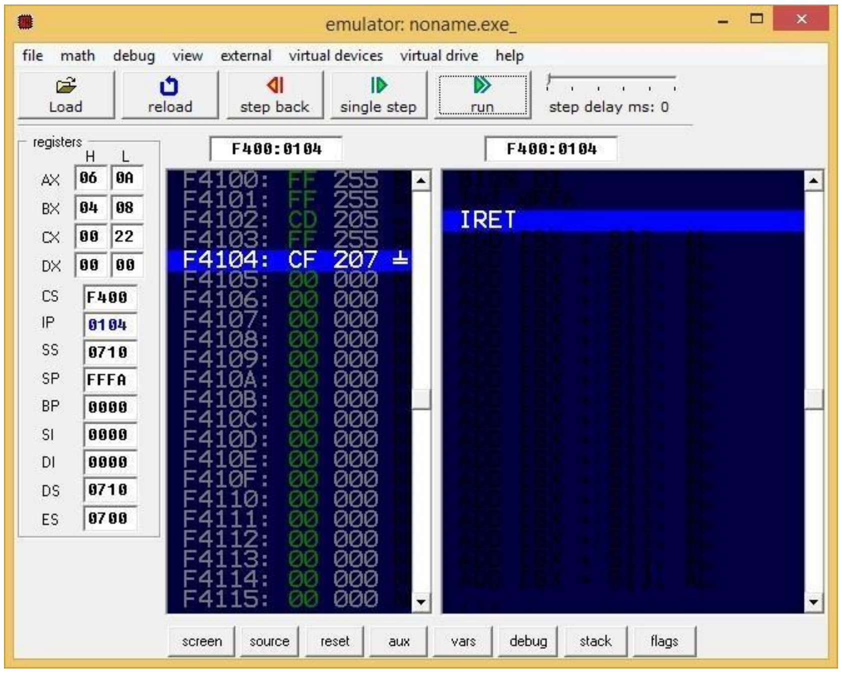
Output:



```
data segment
a dw 0202h
b dw 0408h
c dw ?
data ends

code segment
assume cs:code,ds:data
start:
mov ax,data
mov ds,ax
mov ax,a
mov bx,b
add ax,bx
mov c,ax
int 3
code ends
end start
```



```
CF 0
ZF 0
SF 0
OF 0
PF 1
AF 0
IF 0
DF 0
```



```
registers
     H   L
AX   06  0A
BX   04  08
CX   00  22
DX   00  00
CS   F400
IP   0104
SS   0710
SP   FFFA
BP   0000
SI   0000
DI   0000
DS   0710
ES   0700
```

```
F400:0104

F4100: FF 255
F4101: FF 255
F4102: CD 205
F4103: FF 255
F4104: CF 207 ≐
F4105: 00 000
F4106: 00 000
F4107: 00 000
F4108: 00 000
F4109: 00 000
F410A: 00 000
F410B: 00 000
F410C: 00 000
F410D: 00 000
F410E: 00 000
F410F: 00 000
F4110: 00 000
F4111: 00 000
F4112: 00 000
F4113: 00 000
F4114: 00 000
F4115: 00 000
```

```
F400:0104

IRET
```

Result:

Thus the assembly language program to perform the addition of 16-bit has been performed and executed.

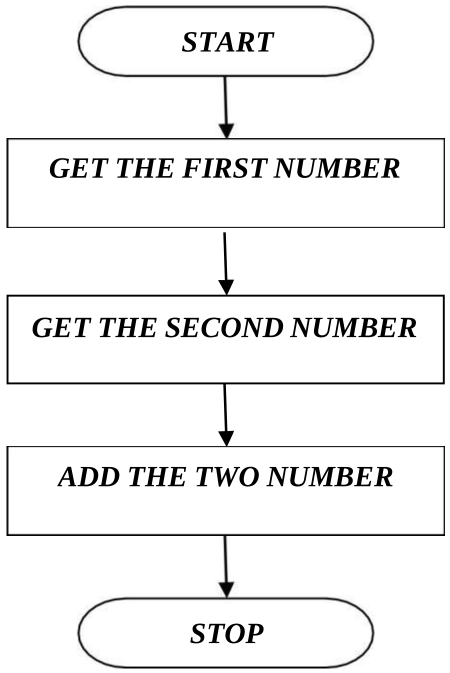
Experiment: 1.c

Aim: To perform the addition of 32-bit using assembly language.

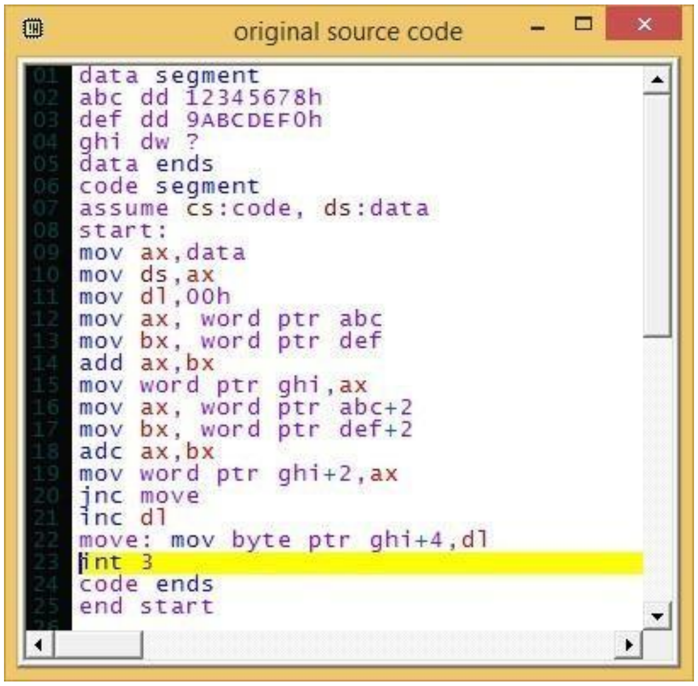
Program:

```
data segment
abc dd 12345678h
def dd 9ABCDEF0h
ghi dw ?
data ends

code segment
assume cs:code, ds:data
start:
mov ax,data
mov ds,ax
mov dl,00h
mov ax, word ptr abc
mov bx, word ptr def
add ax,bx
mov word ptr ghi,ax
mov ax, word ptr abc+2
mov bx, word ptr def+2
adc ax,bx
mov word ptr ghi+2,ax
jnc move
inc dl
move: mov byte ptr ghi+4,dl
int 3
code ends
end start
```

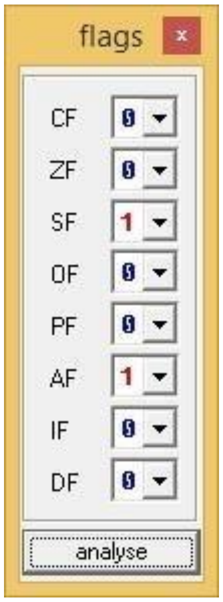
Flowchart:



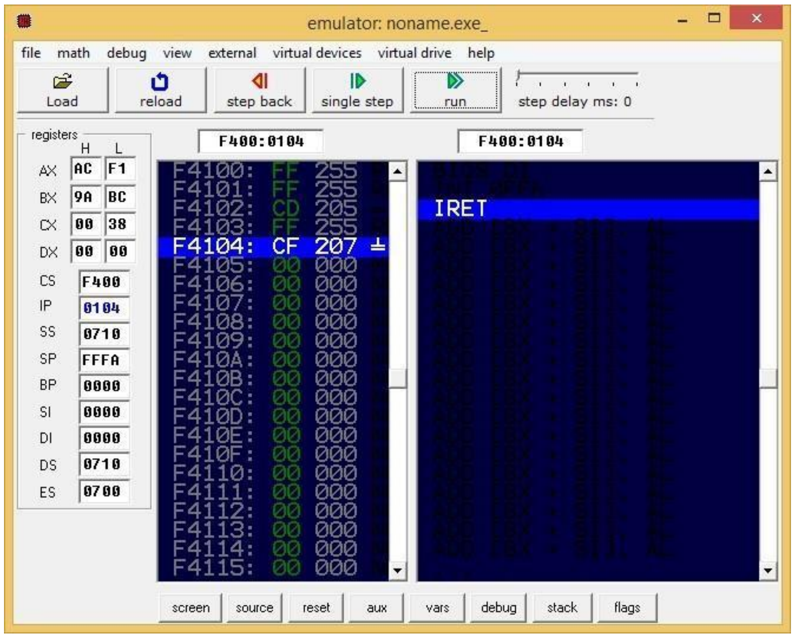
Output:



```
data segment
abc dd 12345678h
def dd 9ABCDEF0h
ghi dw ?
data ends
code segment
assume cs:code, ds:data
start:
mov ax,data
mov ds,ax
mov dl,00h
mov ax, word ptr abc
mov bx, word ptr def
add ax,bx
mov word ptr ghi,ax
mov ax, word ptr abc+2
mov bx, word ptr def+2
adc ax,bx
mov word ptr ghi+2,ax
jnc move
inc dl
move: mov byte ptr ghi+4,dl
int 3
code ends
end start
```



CF	0
ZF	0
SF	1
OF	0
PF	0
AF	1
IF	0
DF	0



registers

	H	L
AX	AC	F1
BX	9A	BC
CX	00	38
DX	00	00

CS	F400
IP	0104
SS	0710
SP	FFFA
BP	0000
SI	0000
DI	0000
DS	0710
ES	0700

```
F400:0104                F400:0104

F4100: FF 255
F4101: FF 255
F4102: CD 205            IRET
F4103: FF 255
F4104: CF 207 ╧
F4105: 00 000
F4106: 00 000
F4107: 00 000
F4108: 00 000
F4109: 00 000
F410A: 00 000
F410B: 00 000
F410C: 00 000
F410D: 00 000
F410E: 00 000
F410F: 00 000
F4110: 00 000
F4111: 00 000
F4112: 00 000
F4113: 00 000
F4114: 00 000
F4115: 00 000
```

Result:

Thus the assembly language program to perform the addition of 32-bit has been performed and executed.

JNIESTRT'S
SMT. INDIRA GANDHI COLLEGE OF ENGINEERING
GHANSOLI, NAVI MUMBAI – 400 709
(Approved by AICTE New Delhi & Govt. of Maharashtra, Affiliated to University of Mumbai)
Computer Engineering Department
ACADEMIC YEAR: - 2023-24 (EVEN SEM)

Experiment No:- 2

Aim:- To perform the subtraction

Total Marks(10)					Total Marks	DOP	Sign
A	**B**	**C**	**D**	**E**			
2	**3**	**2**	**2**	**1**			

Experiment No: 2.a

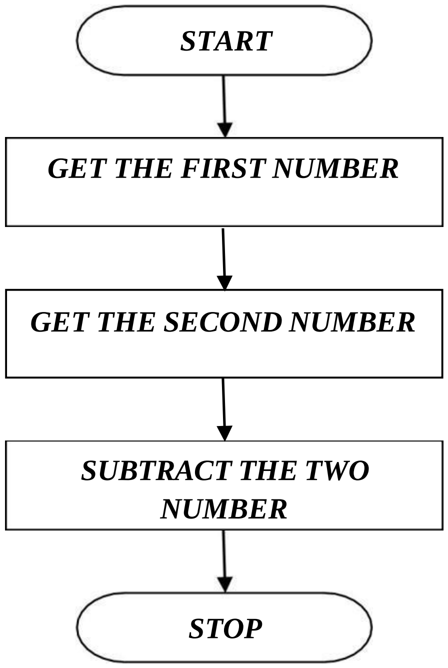
Aim: To perform the subtraction of 8-bit using assembly language.

Software: Emulator 8086.

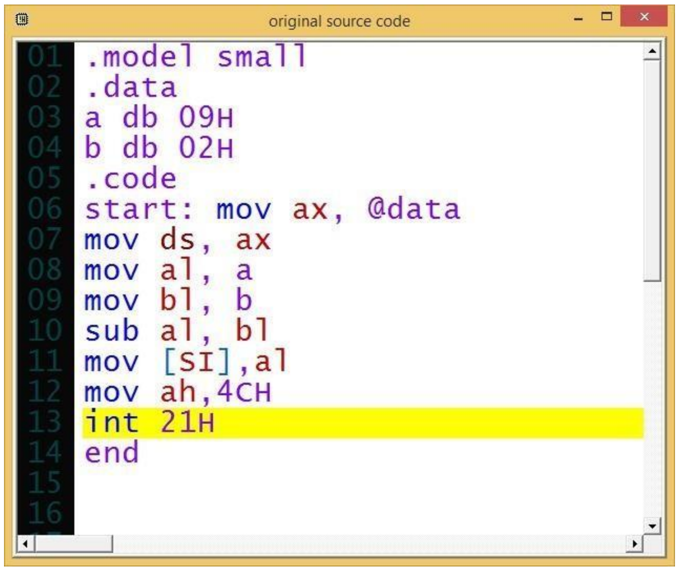
Program:

```
.model small
.data
a db 09H
b db 02H
.code
start: mov ax, @data
mov ds, ax
mov al, a
mov bl, b
sub al, bl
mov [SI],al
mov ah,4CH
int 21H
end
```

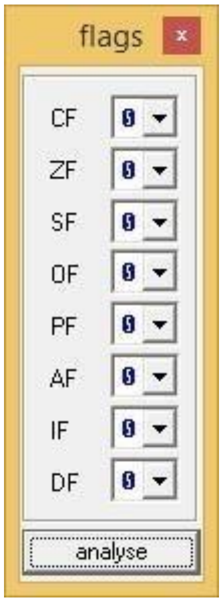
Flowchart:



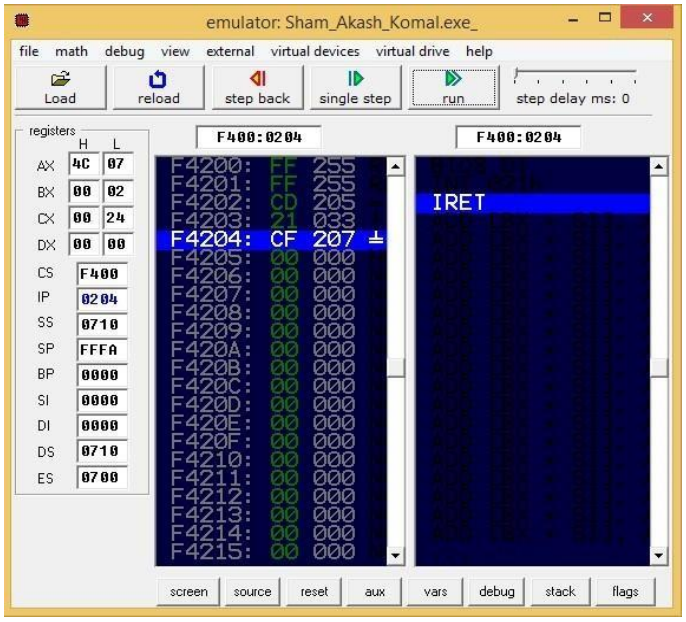
Output:



```
.model small
.data
a db 09H
b db 02H
.code
start: mov ax, @data
mov ds, ax
mov al, a
mov bl, b
sub al, bl
mov [SI],al
mov ah,4CH
int 21H
end
```



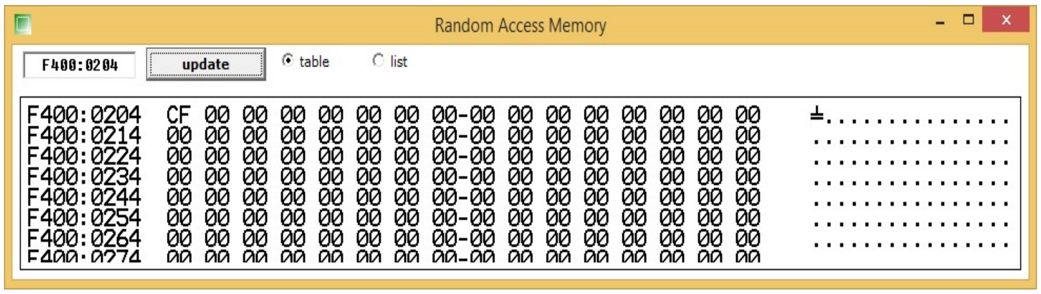
```
CF 0
ZF 0
SF 0
OF 0
PF 0
AF 0
IF 0
DF 0
```


emulator: Sham_Akash_Komal.exe_
file math debug view external virtual devices virtual drive help
Load
reload
step back
single step
run
step delay ms: 0
screen
source
reset
aux
vars
debug
stack
flags

```
registers
     H  L
AX  4C 07
BX  00 02
CX  00 24
DX  00 00
CS  F400
IP  0204
SS  0710
SP  FFFA
BP  0000
SI  0000
DI  0000
DS  0710
ES  0700
```

```
F400:0204
F4200: FF 255
F4201: FF 255
F4202: CD 205
F4203: 21 033
F4204: CF 207 ╧
F4205: 00 000
F4206: 00 000
F4207: 00 000
F4208: 00 000
F4209: 00 000
F420A: 00 000
F420B: 00 000
F420C: 00 000
F420D: 00 000
F420E: 00 000
F420F: 00 000
F4210: 00 000
F4211: 00 000
F4212: 00 000
F4213: 00 000
F4214: 00 000
F4215: 00 000
```

```
F400:0204
IRET
```



```
F400:0204
F400:0204  CF 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ╧...............
F400:0214  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
F400:0224  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
F400:0234  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
F400:0244  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
F400:0254  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
F400:0264  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
F400:0274  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
```

Result:

Thus the assembly language program to perform the subtraction of 8-bit has been performed and executed

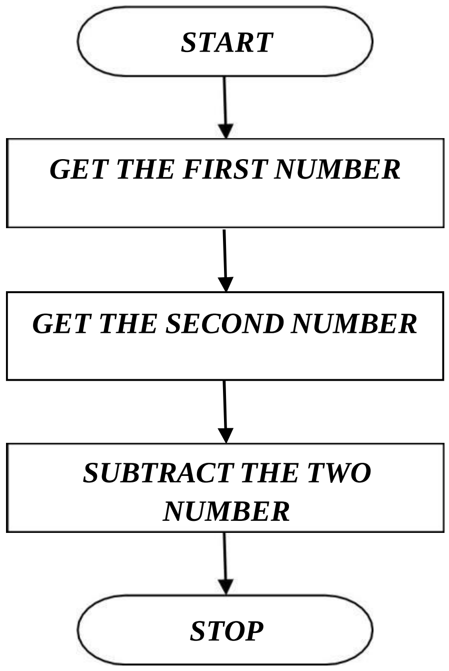
Experiment: 2.b

Aim: To perform the subtraction of 16-bit using assembly language.

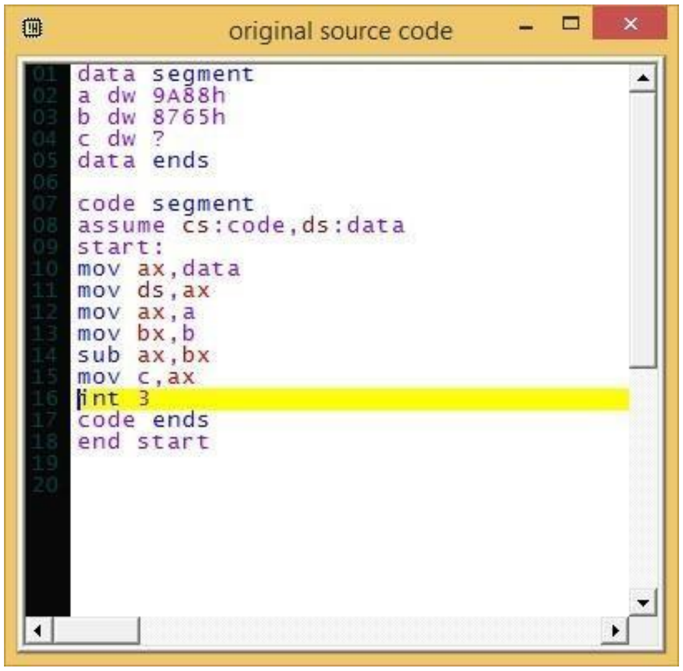
Program:

```
data segment
a dw 9A88h
b dw 8765h
c dw ?
data ends

code segment
assume cs:code,ds:data
start:
mov ax,data
mov ds,ax
mov ax,a
mov bx,b
sub ax,bx
mov  c,ax
int 3
code ends
end start
```

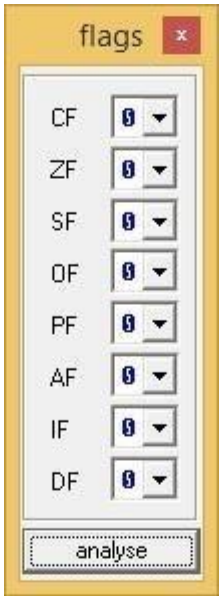
Flowchart:



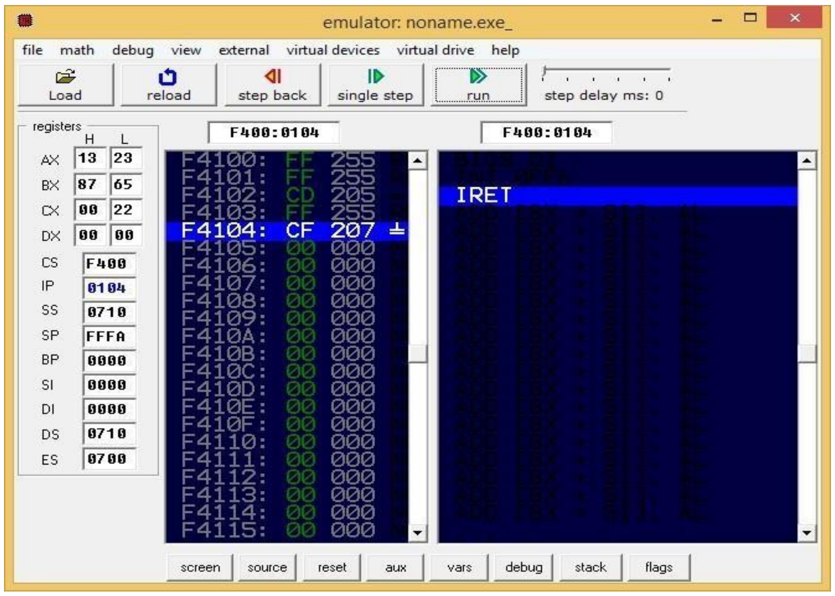
Output:



```
data segment
a dw 9A88h
b dw 8765h
c dw ?
data ends

code segment
assume cs:code,ds:data
start:
mov ax,data
mov ds,ax
mov ax,a
mov bx,b
sub ax,bx
mov c,ax
int 3
code ends
end start
```



```
CF 0
ZF 0
SF 0
OF 0
PF 0
AF 0
IF 0
DF 0
```


emulator: noname.exe_
file math debug view external virtual devices virtual drive help
Load reload step back single step run step delay ms: 0
screen source reset aux vars debug stack flags

```
registers
     H   L
AX   13  23
BX   87  65
CX   00  22
DX   00  00
CS   F400
IP   0104
SS   0710
SP   FFFA
BP   0000
SI   0000
DI   0000
DS   0710
ES   0700

F400:0104                F400:0104

F4100: FF 255
F4101: FF 255
F4102: CD 205
F4103: FF 255
F4104: CF 207 ±          IRET
F4105: 00 000
F4106: 00 000
F4107: 00 000
F4108: 00 000
F4109: 00 000
F410A: 00 000
F410B: 00 000
F410C: 00 000
F410D: 00 000
F410E: 00 000
F410F: 00 000
F4110: 00 000
F4111: 00 000
F4112: 00 000
F4113: 00 000
F4114: 00 000
F4115: 00 000
```

Result:

Thus the assembly language program to perform the subtraction of 16-bit has been performed and executed.

	JNIESTRT'S **SMT. INDIRA GANDHI COLLEGE OF ENGINEERING** **GHANSOLI, NAVI MUMBAI – 400 709** **(Approved by AICTE New Delhi & Govt. of Maharashtra, Affiliated to University of Mumbai)** **Computer Engineering Department** **ACADEMIC YEAR: - 2023-24 (EVEN SEM)**

Experiment No:- 3

Aim:- To perform multiplication

Total Marks(10)					**Total Marks**	**DOP**	**Sign**
A	**B**	**C**	**D**	**E**			
2	**3**	**2**	**2**	**1**			

Experiment No: 3.a

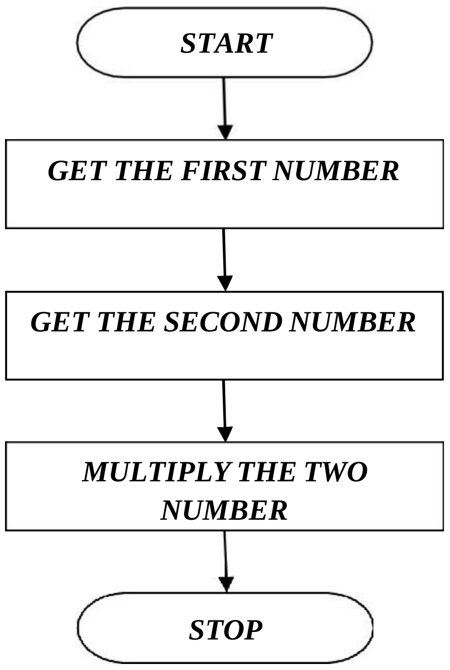
Aim: To perform 8-bit multiplication.

Software use: Emulator 8086.

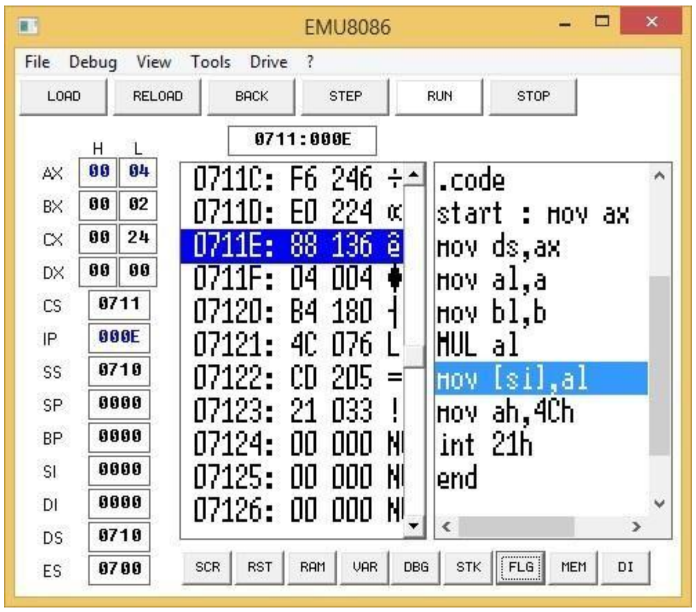
Program:

```
.model small

.data

a db 09

b db 02H       .code

start: mov ax, @data

mov ds, ax

mov al, a

mov bl, b

mul mov [SI],al

mov  ah,4CH

int 21H

end
```

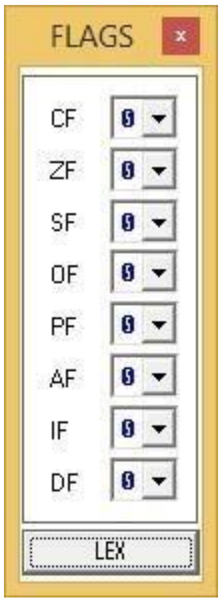
Flowchart:



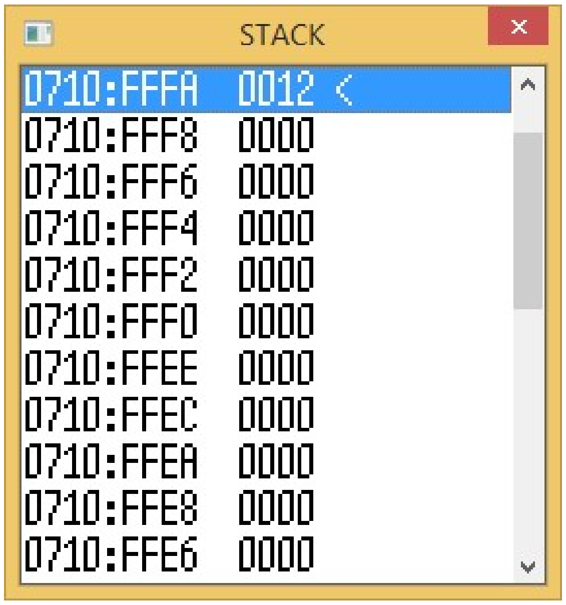
OUTPUT:



```
EMU8086
File  Debug  View  Tools  Drive  ?
LOAD  RELOAD  BACK  STEP  RUN  STOP
0711:000E
     H   L
AX  00  04
BX  00  02
CX  00  24
DX  00  00
CS  0711
IP  000E
SS  0710
SP  0000
BP  0000
SI  0000
DI  0000
DS  0710
ES  0700

0711C: F6 246 ÷     .code
0711D: E0 224 α     start : mov ax
0711E: 88 136 ê     mov ds,ax
0711F: 04 004 ♦     mov al,a
07120: B4 180 ┤     mov bl,b
07121: 4C 076 L     MUL al
07122: CD 205 ═     mov [si],al
07123: 21 033 !     mov ah,4Ch
07124: 00 000 NU    int 21h
07125: 00 000 NU    end
07126: 00 000 NU
```



```
FLAGS
CF  0
ZF  0
SF  0
OF  0
PF  0
AF  0
IF  0
DF  0
```



```
STACK
0710:FFFA   0012 <
0710:FFF8   0000
0710:FFF6   0000
0710:FFF4   0000
0710:FFF2   0000
0710:FFF0   0000
0710:FFEE   0000
0710:FFEC   0000
0710:FFEA   0000
0710:FFE8   0000
0710:FFE6   0000
```

Result:

Thus the assembly language program to perform 8-bit multiplication has been performed and executed

Experiment No: 3.b

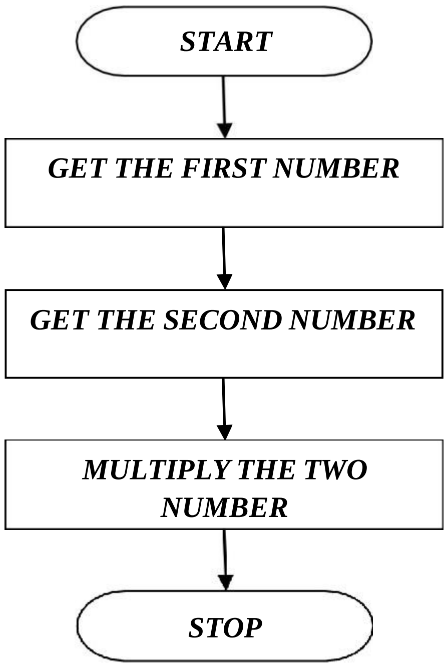
Aim: To perform 16-bit multiplication.

Software Use: Emulator 8086.

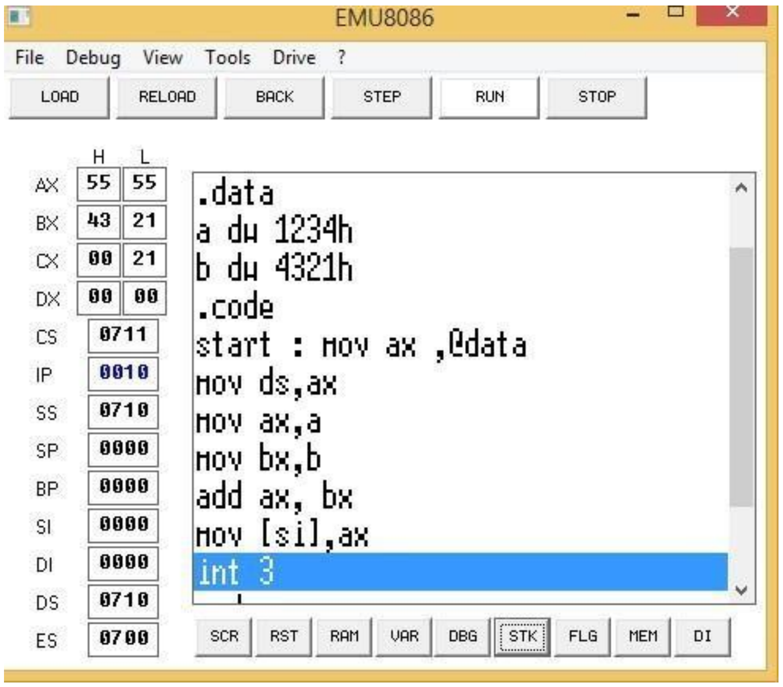
PROGRAM:

```
.model small
.data
a dw 1234h
b dw 4321h
.code
start : mov ax ,@data
mov ds,ax
mov ax,a
mov bx,b
add ax, bx
mov [si],ax
int 3
end
```

Flowchart:

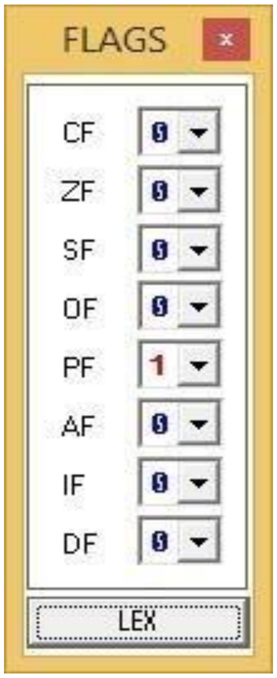


OUTPUT:

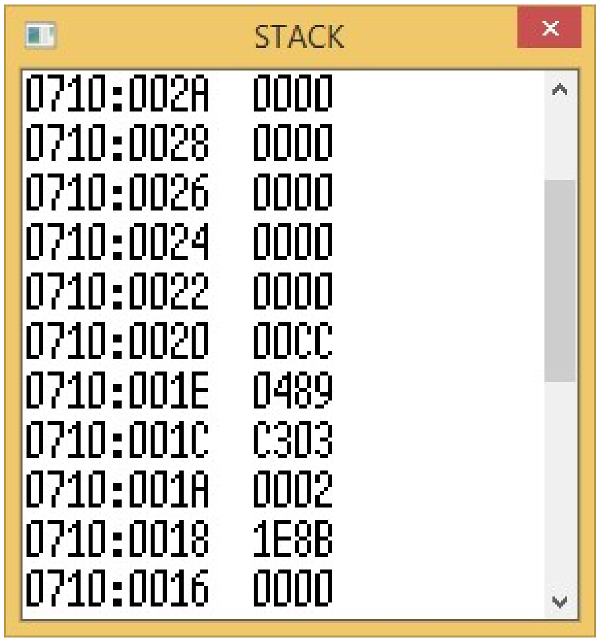

EMU8086
File Debug View Tools Drive ?
LOAD RELOAD BACK STEP RUN STOP
SCR RST RAM VAR DBG STK FLG MEM DI

```
     H  L
AX  55 55
BX  43 21
CX  00 21
DX  00 00
CS  0711
IP  0010
SS  0710
SP  0000
BP  0000
SI  0000
DI  0000
DS  0710
ES  0700
```

```
.data
a dw 1234h
b dw 4321h
.code
start : mov ax ,@data
mov ds,ax
mov ax,a
mov bx,b
add ax, bx
mov [si],ax
int 3
```



```
CF 0
ZF 0
SF 0
OF 0
PF 1
AF 0
IF 0
DF 0
```



```
0710:002A  0000
0710:0028  0000
0710:0026  0000
0710:0024  0000
0710:0022  0000
0710:0020  00CC
0710:001E  0489
0710:001C  C303
0710:001A  0002
0710:0018  1E8B
0710:0016  0000
```

Result:

Thus the assembly language program to perform 16-bit multiplication has been performed and executed

JNIESTRT'S
SMT. INDIRA GANDHI COLLEGE OF ENGINEERING
GHANSOLI, NAVI MUMBAI – 400 709
(Approved by AICTE New Delhi & Govt. of Maharashtra, Affiliated to University of Mumbai)
Computer Engineering Department
ACADEMIC YEAR: - 2023-24 (EVEN SEM)

Experiment No:- 4

Aim:- To perform the division

Total Marks(10)					Total Marks	DOP	Sign
A	B	C	D	E			
2	3	2	2	1			

Experiment: 4.a

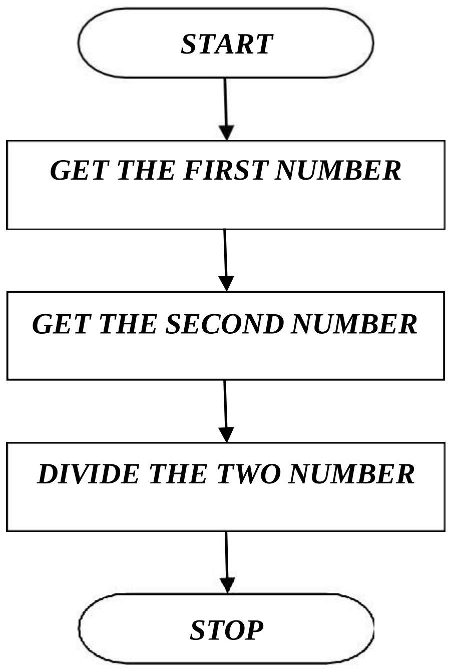
Aim: To perform the division of 8-bit number using assembly language.

Software Use: Emu 8086/MASAM

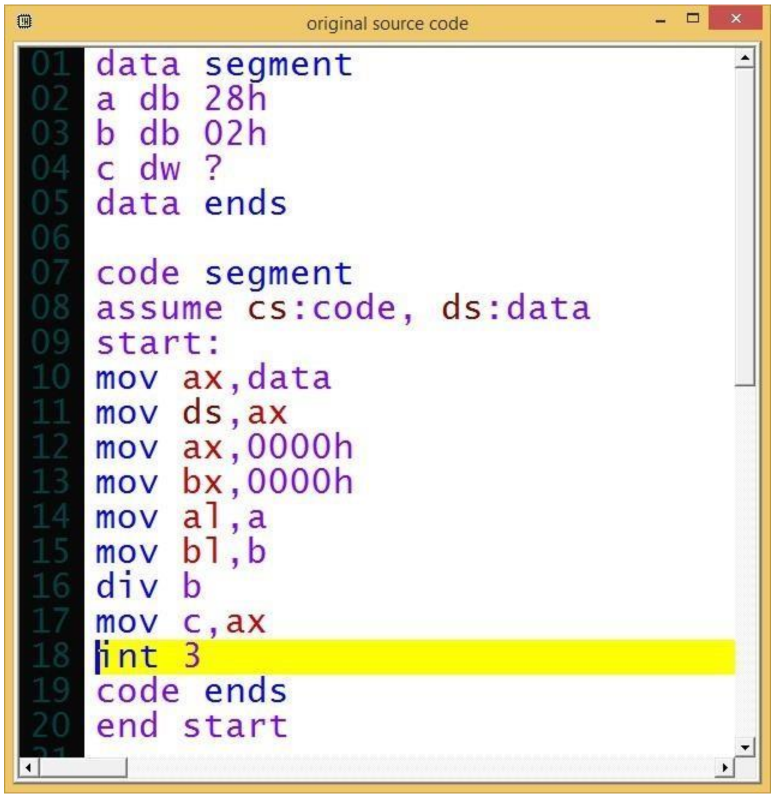
Program:

```
data segment
a db 28h
b db 02h
c dw ?
data ends

code segment
assume cs:code, ds:data
start:
mov ax,data
mov ds,ax
mov ax,0000h
mov bx,0000h
mov al,a
mov bl,b
div b
mov c,ax
int 3
code ends
end start
```

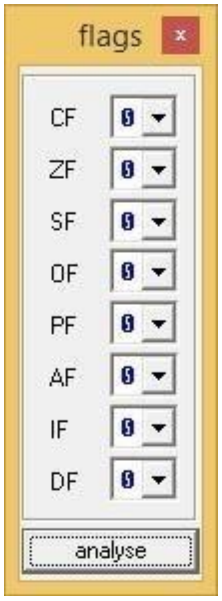
Flowchart:



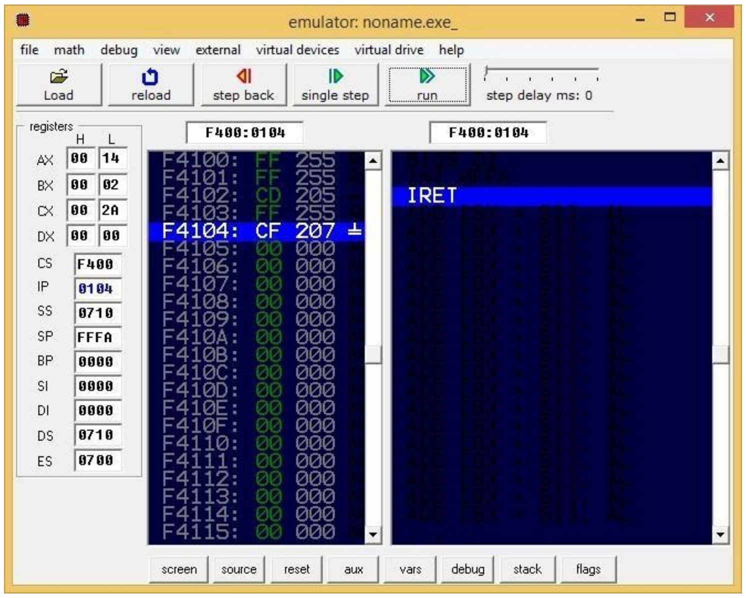
Output:



```
data segment
a db 28h
b db 02h
c dw ?
data ends

code segment
assume cs:code, ds:data
start:
mov ax,data
mov ds,ax
mov ax,0000h
mov bx,0000h
mov al,a
mov bl,b
div b
mov c,ax
int 3
code ends
end start
```



```
CF 0
ZF 0
SF 0
OF 0
PF 0
AF 0
IF 0
DF 0
```


emulator: noname.exe_
file math debug view external virtual devices virtual drive help
Load reload step back single step run
screen source reset aux vars debug stack flags

step delay ms: 0

```
registers
    H  L
AX  00 14
BX  00 02
CX  00 2A
DX  00 00
CS  F400
IP  0104
SS  0710
SP  FFFA
BP  0000
SI  0000
DI  0000
DS  0710
ES  0700
```

```
F400:0104
F4100: FF 255
F4101: FF 255
F4102: CD 205
F4103: FF 255
F4104: CF 207 ±
F4105: 00 000
F4106: 00 000
F4107: 00 000
F4108: 00 000
F4109: 00 000
F410A: 00 000
F410B: 00 000
F410C: 00 000
F410D: 00 000
F410E: 00 000
F410F: 00 000
F4110: 00 000
F4111: 00 000
F4112: 00 000
F4113: 00 000
F4114: 00 000
F4115: 00 000
```

```
F400:0104
IRET
```

Result:

Thus the assembly language program to perform the division of 8-bit number has been performed and executed.

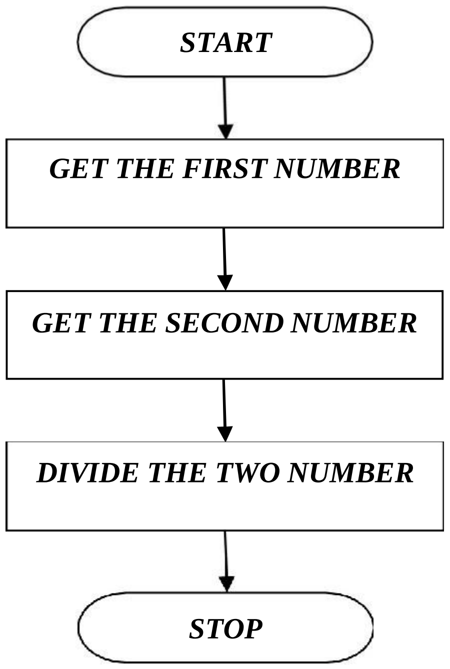
Experiment: 4.b

Aim: To perform the division of 16-bit using assembly language.

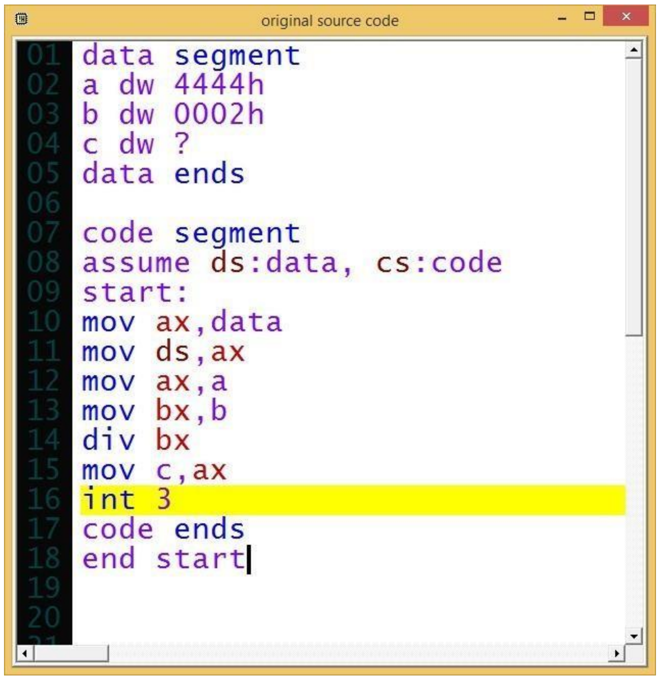
Program:

```
data segment
a dw 4444h
b dw 0002h
c dw ?
data ends

code segment
assume ds:data, cs:code
start:
mov ax,data
mov ds,ax
mov ax,a
mov bx,b
div bx
mov c,ax
int 3
code ends
end start
```

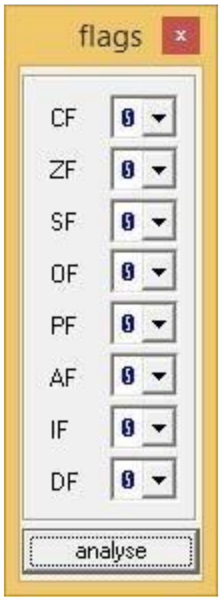
Flowchart:



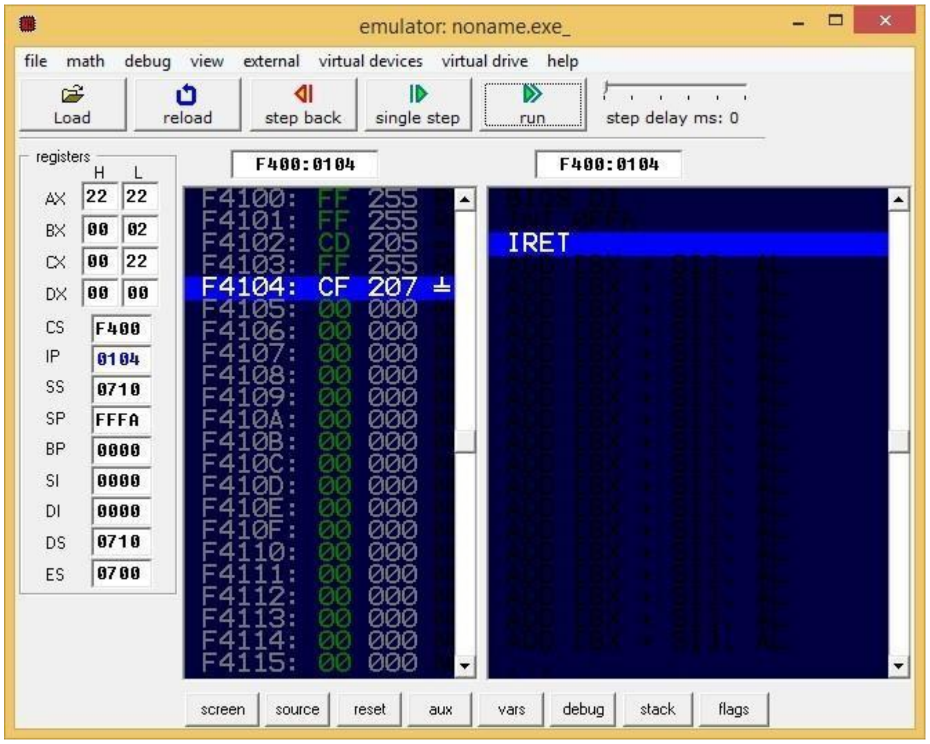
Output:



```
data segment
a dw 4444h
b dw 0002h
c dw ?
data ends

code segment
assume ds:data, cs:code
start:
mov ax,data
mov ds,ax
mov ax,a
mov bx,b
div bx
mov c,ax
int 3
code ends
end start
```



```
flags
CF 0
ZF 0
SF 0
OF 0
PF 0
AF 0
IF 0
DF 0
```



```
emulator: noname.exe_

registers
     H   L
AX   22  22
BX   00  02
CX   00  22
DX   00  00
CS   F400
IP   0104
SS   0710
SP   FFFA
BP   0000
SI   0000
DI   0000
DS   0710
ES   0700

F400:0104                F400:0104

F4100: FF 255            IRET
F4101: FF 255
F4102: CD 205
F4103: FF 255
F4104: CF 207 ╧
F4105: 00 000
F4106: 00 000
F4107: 00 000
F4108: 00 000
F4109: 00 000
F410A: 00 000
F410B: 00 000
F410C: 00 000
F410D: 00 000
F410E: 00 000
F410F: 00 000
F4110: 00 000
F4111: 00 000
F4112: 00 000
F4113: 00 000
F4114: 00 000
F4115: 00 000
```

Result:

Thus the assembly language program to perform the division of 16-bit has been performed and executed.

	JNIESTRT'S **SMT. INDIRA GANDHI COLLEGE OF ENGINEERING** **GHANSOLI, NAVI MUMBAI – 400 709** **(Approved by AICTE New Delhi & Govt. of Maharashtra, Affiliated to University of Mumbai)** **Computer Engineering Department** **ACADEMIC YEAR: - 2023-24 (EVEN SEM)**

Experiment No:- 5

Aim:- To perform the arithmetic equation using assembly language.

Total Marks(10)					**Total Marks**	**DOP**	**Sign**
A	**B**	**C**	**D**	**E**			
2	**3**	**2**	**2**	**1**			

Experiment No: 5

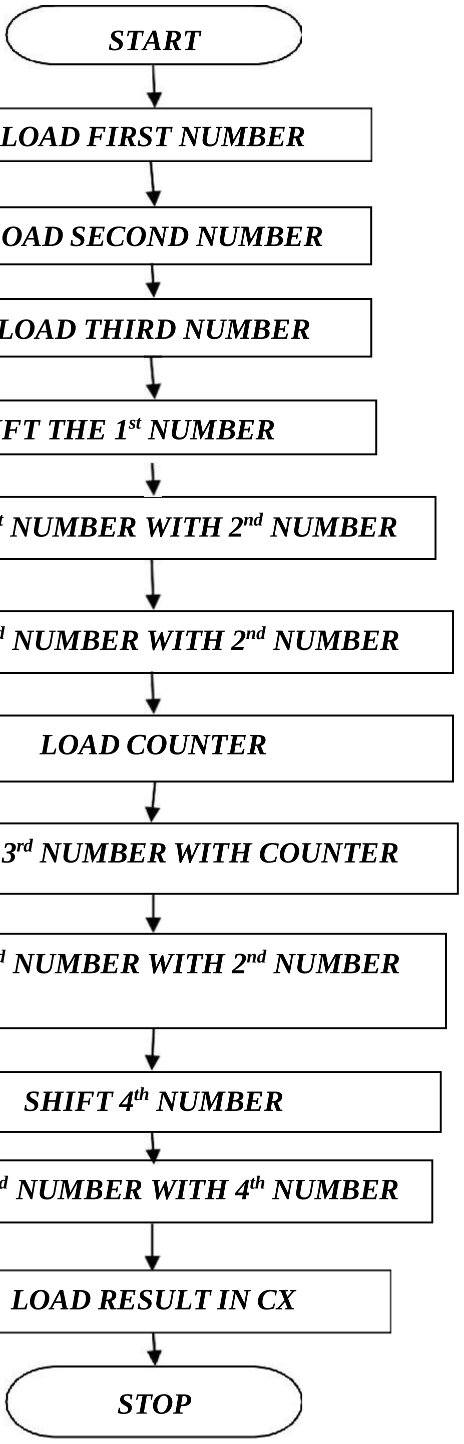
Aim: To perform the arithmetic equation using assembly language.

Software Use: EMU 8086

Program:

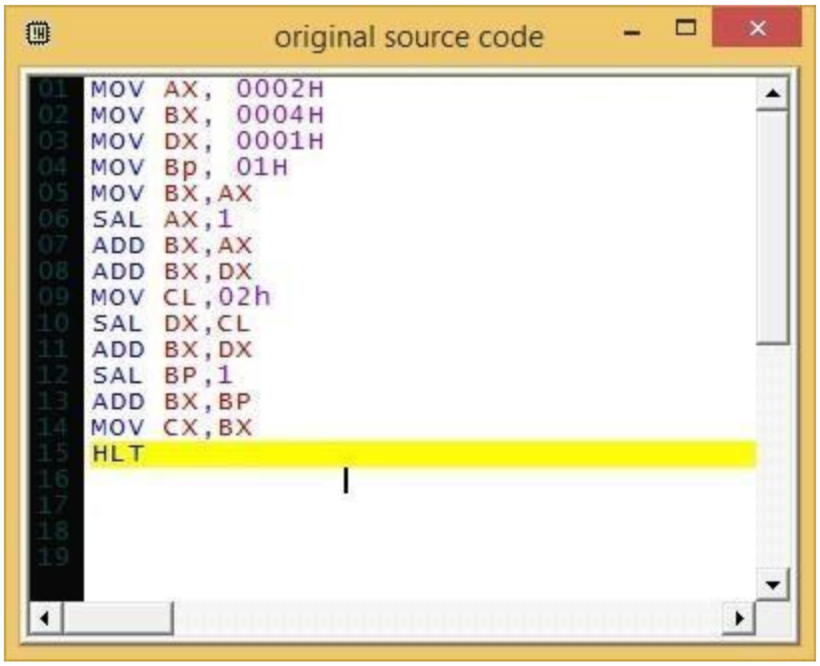
```
MOV AX, 0002H

MOV BX, 0004H

MOV DX, 0001H

MOV BP, 01H

MOV BX,AX

SAL AX,1

ADD BX,AX

ADD BX,DX

MOV CL,02h

SAL DX,CL

ADD BX,DX

SAL BP,1

ADD BX,BP

MOV CX,BX

HLT
```

Flowchart:



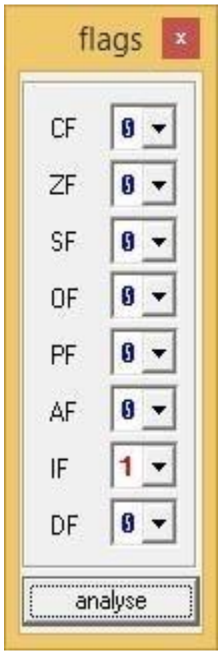
START
LOAD FIRST NUMBER
LOAD SECOND NUMBER
LOAD THIRD NUMBER
SHIFT THE 1st NUMBER
ADD 1st NUMBER WITH 2nd NUMBER
ADD 3rd NUMBER WITH 2nd NUMBER
LOAD COUNTER
SHIFT 3rd NUMBER WITH COUNTER
ADD 3rd NUMBER WITH 2nd NUMBER
SHIFT 4th NUMBER
ADD 2nd NUMBER WITH 4th NUMBER
LOAD RESULT IN CX
STOP

Output:



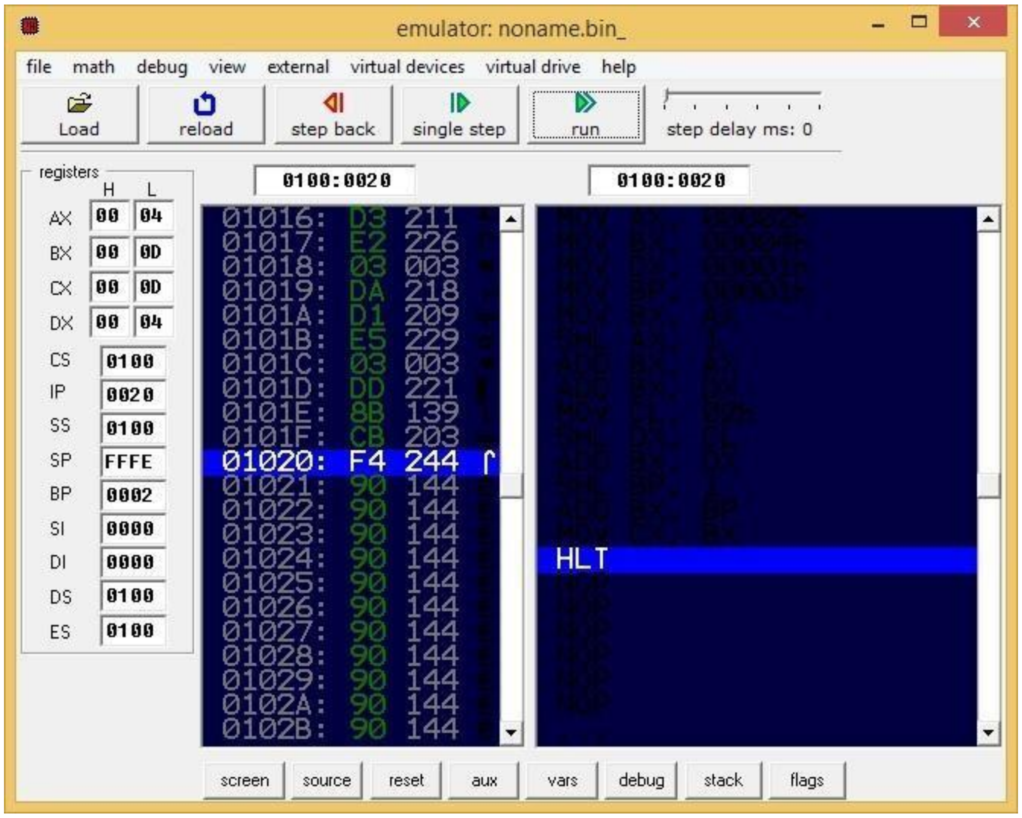
original source code

```
MOV AX, 0002H
MOV BX, 0004H
MOV DX, 0001H
MOV Bp, 01H
MOV BX,AX
SAL AX,1
ADD BX,AX
ADD BX,DX
MOV CL,02h
SAL DX,CL
ADD BX,DX
SAL BP,1
ADD BX,BP
MOV CX,BX
HLT
```



flags

```
CF 0
ZF 0
SF 0
OF 0
PF 0
AF 0
IF 1
DF 0
```



emulator: noname.bin_

```
registers
     H   L
AX   00  04
BX   00  0D
CX   00  0D
DX   00  04
CS   0100
IP   0020
SS   0100
SP   FFFE
BP   0002
SI   0000
DI   0000
DS   0100
ES   0100
```

```
0100:0020

01016: D3 211
01017: E2 226
01018: 03 003
01019: DA 218
0101A: D1 209
0101B: E5 229
0101C: 03 003
0101D: DD 221
0101E: 8B 139
0101F: CB 203
01020: F4 244
01021: 90 144
01022: 90 144
01023: 90 144
01024: 90 144
01025: 90 144
01026: 90 144
01027: 90 144
01028: 90 144
01029: 90 144
0102A: 90 144
0102B: 90 144
```

```
0100:0020

HLT
```

Result:

Thus the assembly language program to perform the arithmetic equation has been performed and executed

	JNIESTRT'S **SMT. INDIRA GANDHI COLLEGE OF ENGINEERING** **GHANSOLI, NAVI MUMBAI – 400 709** **(Approved by AICTE New Delhi & Govt. of Maharashtra, Affiliated to University of Mumbai)** **Computer Engineering Department** **ACADEMIC YEAR: - 2023-24 (EVEN SEM)**

Experiment No: - 6

Aim:- To Interface Stepper motor with 8086.

Total Marks(10)					Total Marks	DOP	Sign
A	**B**	**C**	**D**	**E**			
2	**3**	**2**	**2**	**1**			

Experiment No: 6

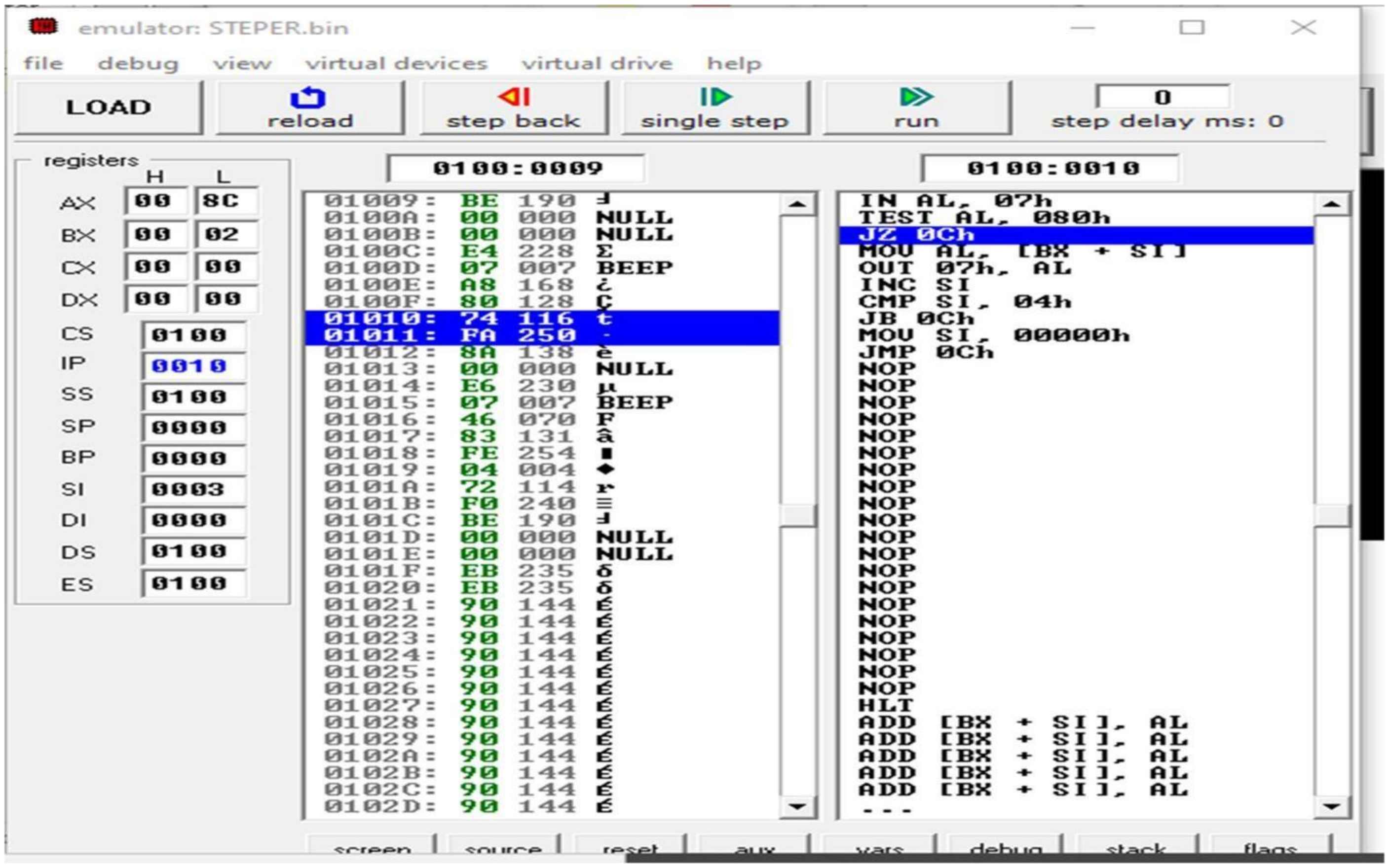
Aim: To Interface Stepper motor with 8086.

Software Use: Emu 8086

Program:

```
#start=stepper_motor.exe#
jmp start
datcw db 0000_0011b
      db  0000_0110b
      db  0000_1100b
      db 0000_1001b


START:
MOV BX,offset datcw
MOV SI ,0
NEXT_STEP:
WAIT: IN AL,07H
TEST AL,10000000b
JZ WAIT
MOV AL,[BX][SI]
OUT 7,AL
INC SI
CMP SI,4
JC NEXT_STEP
MOV SI ,0
JMP NEXT_STEP
```

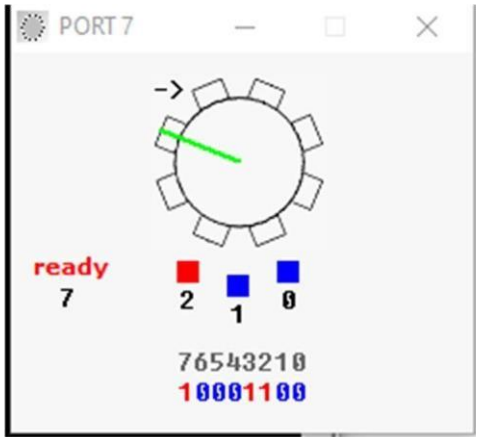
OUTPUT:

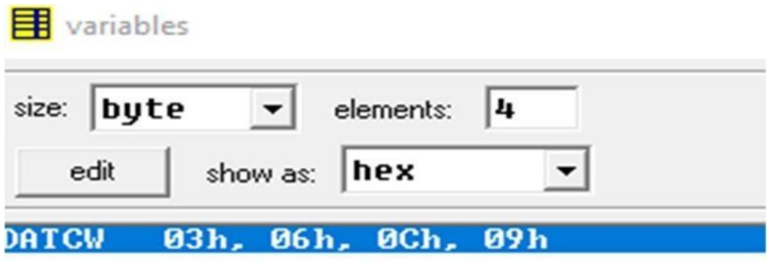

emulator: STEPER.bin
file debug view virtual devices virtual drive help
LOAD
reload
step back
single step
run
step delay ms: 0
screen source reset aux vars debug stack flags

```
registers
      H    L
AX   00   8C
BX   00   02
CX   00   00
DX   00   00
CS   0100
IP   0010
SS   0100
SP   0000
BP   0000
SI   0003
DI   0000
DS   0100
ES   0100
```

```
0100:0009
01009:  BE  190  ╛
0100A:  00  000  NULL
0100B:  00  000  NULL
0100C:  E4  228  Σ
0100D:  07  007  BEEP
0100E:  A8  168  ¿
0100F:  80  128  Ç
01010:  74  116  t
01011:  FA  250  ·
01012:  8A  138  è
01013:  00  000  NULL
01014:  E6  230  µ
01015:  07  007  BEEP
01016:  46  070  F
01017:  83  131  â
01018:  FE  254  ■
01019:  04  004  ♦
0101A:  72  114  r
0101B:  F0  240  ≡
0101C:  BE  190  ╛
0101D:  00  000  NULL
0101E:  00  000  NULL
0101F:  EB  235  δ
01020:  EB  235  δ
01021:  90  144  É
01022:  90  144  É
01023:  90  144  É
01024:  90  144  É
01025:  90  144  É
01026:  90  144  É
01027:  90  144  É
01028:  90  144  É
01029:  90  144  É
0102A:  90  144  É
0102B:  90  144  É
0102C:  90  144  É
0102D:  90  144  É
```

```
0100:0010
IN AL, 07h
TEST AL, 080h
JZ 0Ch
MOV AL, [BX + SI]
OUT 07h, AL
INC SI
CMP SI, 04h
JB 0Ch
MOV SI, 00000h
JMP 0Ch
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
HLT
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
...
```





size: byte elements: 4
show as: hex

```
DATCW    03h, 06h, 0Ch, 09h
```

Result:

Thus the assembly language program to Interface Stepper motor with 8086 has been performed and executed

JNIESTRT'S
SMT. INDIRA GANDHI COLLEGE OF ENGINEERING
GHANSOLI, NAVI MUMBAI – 400 709
(Approved by AICTE New Delhi & Govt. of Maharashtra, Affiliated to University of Mumbai)
Computer Engineering Department
ACADEMIC YEAR: - 2023-24 (EVEN SEM)

Experiment No: - 7

Aim: - To perform the ascending/descending using assembly language.

Total Marks(10)					Total Marks	DOP	Sign
A	**B**	**C**	**D**	**E**			
2	**3**	**2**	**2**	**1**			

Experiment No: 7

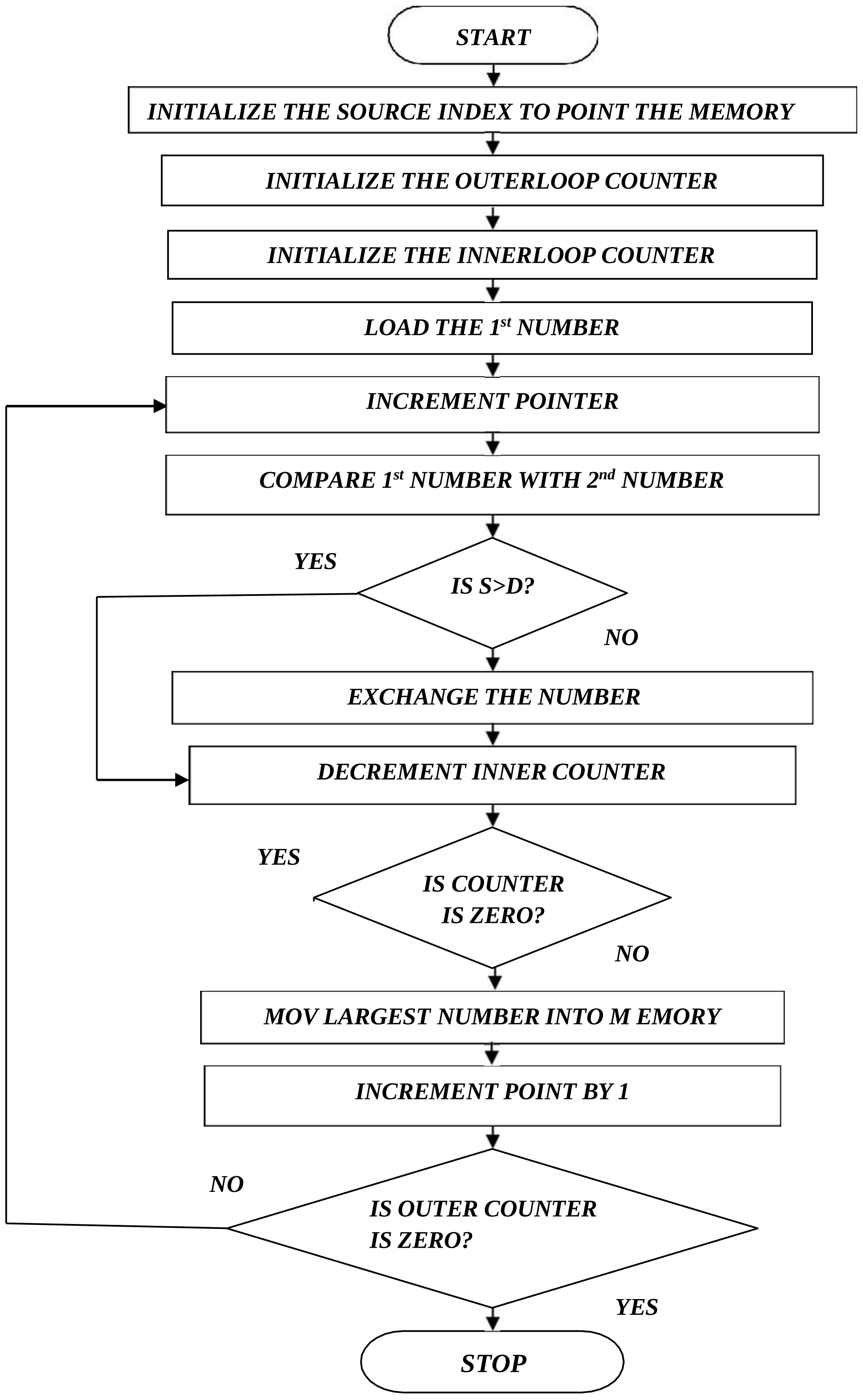
Aim: To perform the ascending/descending using assembly language.

Software Use: EMU 80806/MASAM

Program:

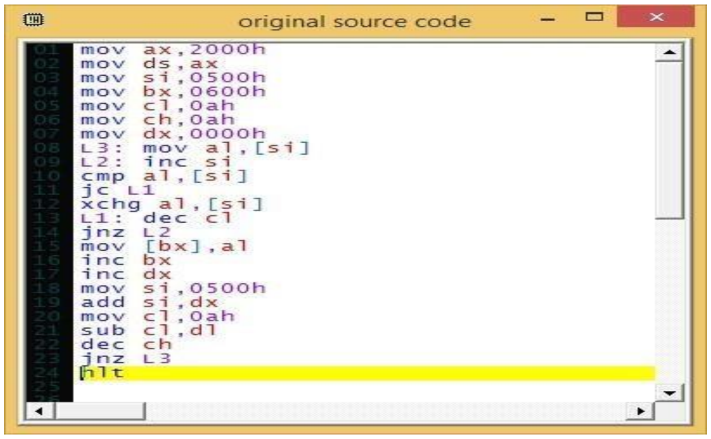
```
 mov ax,2000h
mov  ds,ax
mov si,0500h
mov bx,0600h
mov cl,0ah
mov ch,0ah
mov dx,0000h
L3: mov al,[si]
L2: inc si
 cmp al,[si]
 jc L1
  xchg al,[si]
L1: dec cl
jnz L2
 mov [bx],al
 inc bx
 inc dx
 mov si,0500h
add si,dx
mov cl,0ah
```

Flowchart:

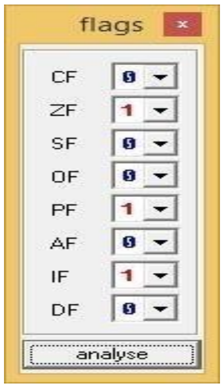


START
INITIALIZE THE SOURCE INDEX TO POINT THE MEMORY
INITIALIZE THE OUTERLOOP COUNTER
INITIALIZE THE INNERLOOP COUNTER
LOAD THE 1st NUMBER
INCREMENT POINTER
COMPARE 1st NUMBER WITH 2nd NUMBER
IS S>D?
YES
NO
EXCHANGE THE NUMBER
DECREMENT INNER COUNTER
IS COUNTER IS ZERO?
YES
NO
MOV LARGEST NUMBER INTO M EMORY
INCREMENT POINT BY 1
IS OUTER COUNTER IS ZERO?
NO
YES
STOP

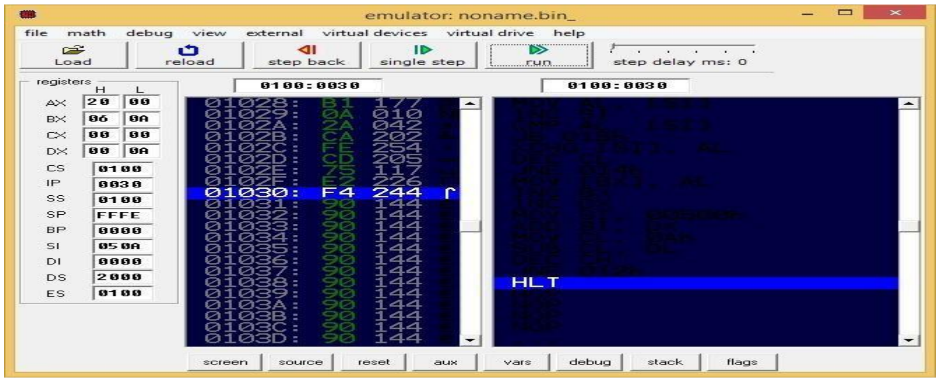
```
 sub cl,dl

dec ch

jnz L3

hlt
```



```
mov ax,2000h
mov ds,ax
mov si,0500h
mov bx,0600h
mov cl,0ah
mov ch,0ah
mov dx,0000h
L3: mov al,[si]
L2: inc si
cmp al,[si]
jc L1
xchg al,[si]
L1: dec cl
jnz L2
mov [bx],al
inc bx
inc dx
mov si,0500h
add si,dx
mov cl,0ah
sub cl,dl
dec ch
jnz L3
hlt
```



```
CF 0
ZF 1
SF 0
OF 0
PF 1
AF 0
IF 1
DF 0
```

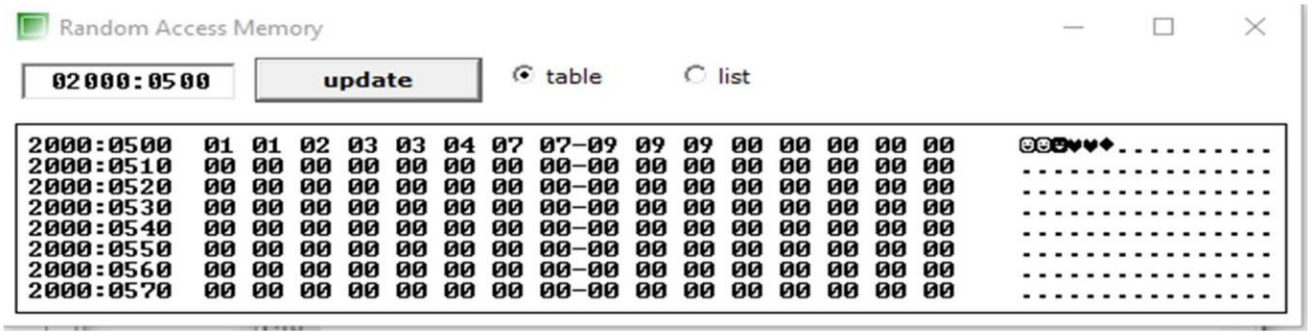


```
registers
     H   L
AX   20  00
BX   06  0A
CX   00  00
DX   00  0A
CS   0100
IP   0030
SS   0100
SP   FFFE
BP   0000
SI   050A
DI   0000
DS   2000
ES   0100
```

```
0100:0030

01028: B1 177
01029: 0A 010
0102A: 2A 042
0102B: CA 202
0102C: FE 254
0102D: CD 205
0102E: 75 117
0102F: E2 226
01030: F4 244
01031: 90 144
01032: 90 144
01033: 90 144
01034: 90 144
01035: 90 144
01036: 90 144
01037: 90 144
01038: 90 144
01039: 90 144
0103A: 90 144
0103B: 90 144
0103C: 90 144
0103D: 90 144
```

```
0100:0030

HLT
```

emulator: noname.bin_
file math debug view external virtual devices virtual drive help
Load reload step back single step run step delay ms: 0
screen source reset aux vars debug stack flags



```
02000:0500

2000:0500  01 01 02 03 03 04 07 07-09 09 09 00 00 00 00 00
2000:0510  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
2000:0520  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
2000:0530  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
2000:0540  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
2000:0550  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
2000:0560  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
2000:0570  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
```

Result:

Thus the assembly language program to perform the ascending / descending has been performed and executed

JNIESTRT'S
SMT. INDIRA GANDHI COLLEGE OF ENGINEERING
GHANSOLI, NAVI MUMBAI – 400 709
(Approved by AICTE New Delhi & Govt. of Maharashtra, Affiliated to University of Mumbai)
Computer Engineering Department
ACADEMIC YEAR: - 2023-24 (EVEN SEM)

Experiment No: - 8

Aim:-. To transfer the block of data using string instruction.

Total Marks(10)					Total Marks	DOP	Sign
A	B	C	D	E			
2	3	2	2	1			

EXPERIMENT NO: 8

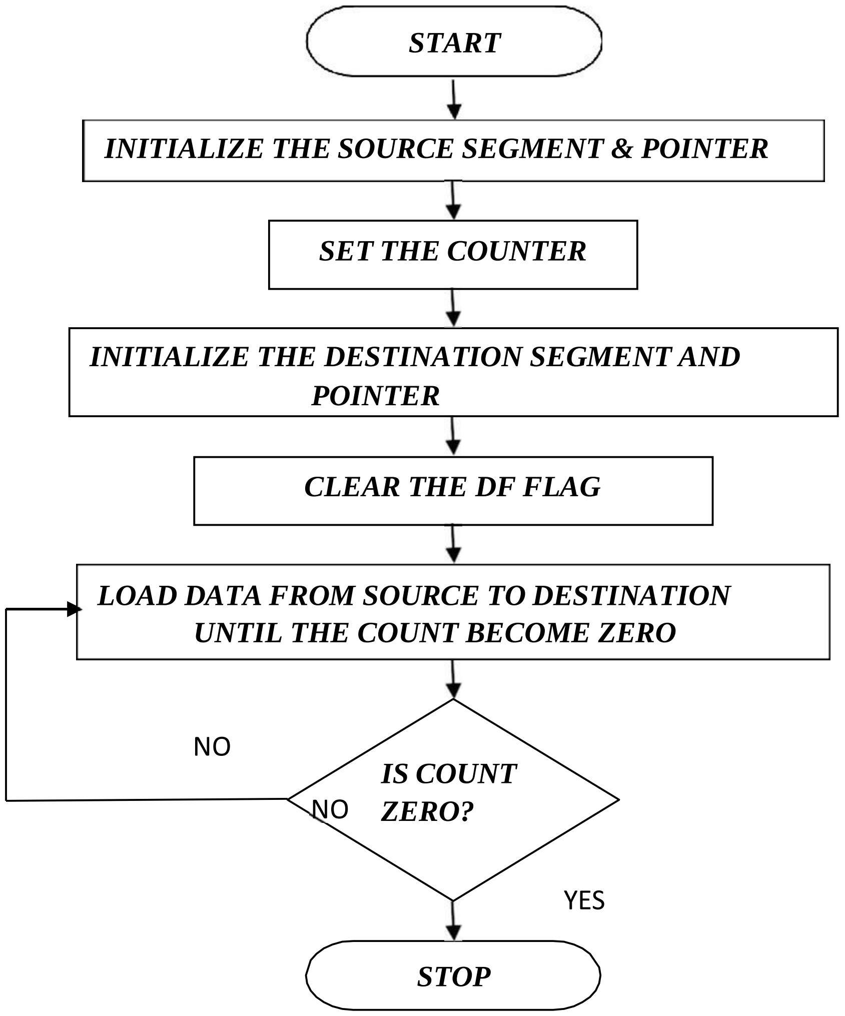
Aim: To transfer the block of data using string instruction.

Software Use: EMU 8086

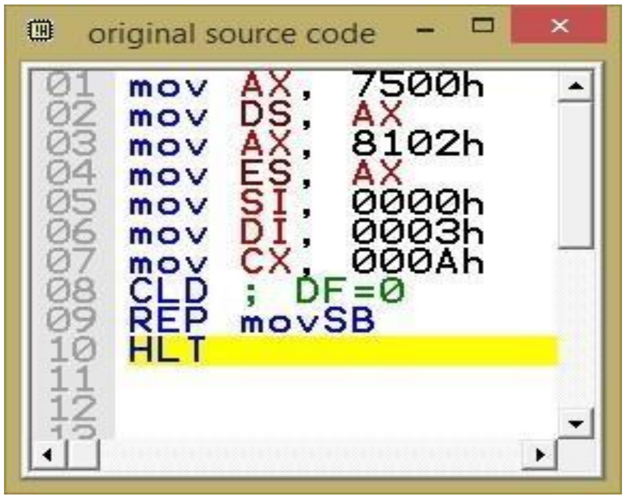
Program:

```
mov AX, 7500h

mov DS, AX

mov AX, 8102h

mov ES, AX

mov SI, 0000h
mov DI, 0003h

mov CX, 000Ah

CLD ; DF=0

REP movSB

HLT
```

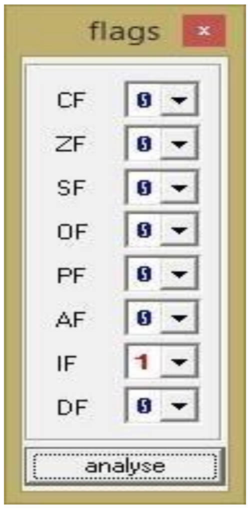
Flowchart:


START
INITIALIZE THE SOURCE SEGMENT & POINTER
SET THE COUNTER
INITIALIZE THE DESTINATION SEGMENT AND POINTER
CLEAR THE DF FLAG
LOAD DATA FROM SOURCE TO DESTINATION UNTIL THE COUNT BECOME ZERO
NO
IS COUNT ZERO?
NO
YES
STOP

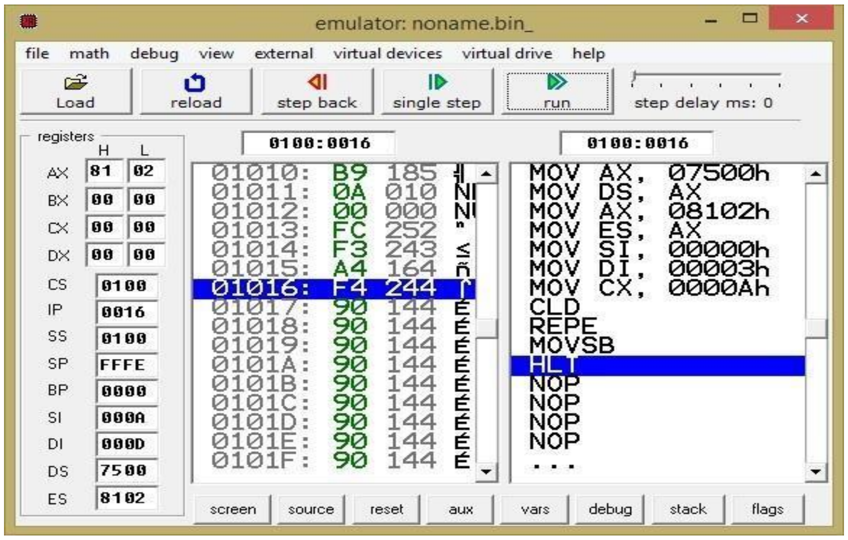
Output:



```
mov AX, 7500h
mov DS, AX
mov AX, 8102h
mov ES, AX
mov SI, 0000h
mov DI, 0003h
mov CX, 000Ah
CLD  ; DF=0
REP movSB
HLT
```



CF	0
ZF	0
SF	0
OF	0
PF	0
AF	0
IF	1
DF	0



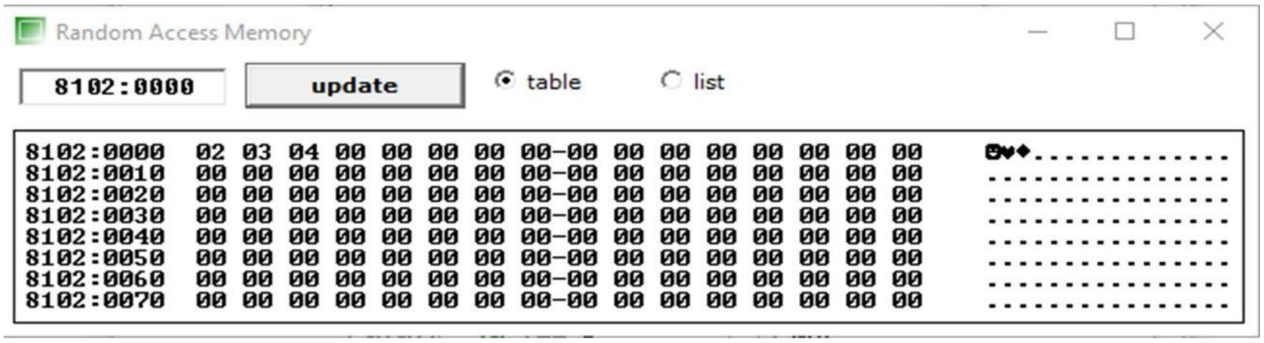
registers

	H	L
AX	81	02
BX	00	00
CX	00	00
DX	00	00

CS	0100
IP	0016
SS	0100
SP	FFFE
BP	0000
SI	000A
DI	000D
DS	7500
ES	8102

```
0100:0016

01010: B9 185 ╣
01011: 0A 010 NL
01012: 00 000 NU
01013: FC 252 ⁿ
01014: F3 243 ≤
01015: A4 164 ñ
01016: F4 244 ⌠
01017: 90 144 É
01018: 90 144 É
01019: 90 144 É
0101A: 90 144 É
0101B: 90 144 É
0101C: 90 144 É
0101D: 90 144 É
0101E: 90 144 É
0101F: 90 144 É
```

```
0100:0016

MOV  AX,  07500h
MOV  DS,  AX
MOV  AX,  08102h
MOV  ES,  AX
MOV  SI,  00000h
MOV  DI,  00003h
MOV  CX,  0000Ah
CLD
REPE
MOVSB
HLT
NOP
NOP
NOP
NOP
...
```



8102:0000 [✓ table] list

```
8102:0000  02 03 04 00 00 00 00 00-00 00 00 00 00 00 00 00   ☻♥♦.............
8102:0010  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
8102:0020  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
8102:0030  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
8102:0040  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
8102:0050  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
8102:0060  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
8102:0070  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
```

Result:

Thus the assembly language program to transfer the block of data using String Operation has been performed and executed

	JNIESTRT'S **SMT. INDIRA GANDHI COLLEGE OF ENGINEERING** **GHANSOLI, NAVI MUMBAI – 400 709** **(Approved by AICTE New Delhi & Govt. of Maharashtra, Affiliated to University of Mumbai)** **Computer Engineering Department** **ACADEMIC YEAR: - 2023-24 (EVEN SEM)**

Experiment No:- 9

Aim:- To find the average of two number using assembly language

Total Marks(10)					**Total Marks**	**DOP**	**Sign**
A	**B**	**C**	**D**	**E**			
2	**3**	**2**	**2**	**1**			

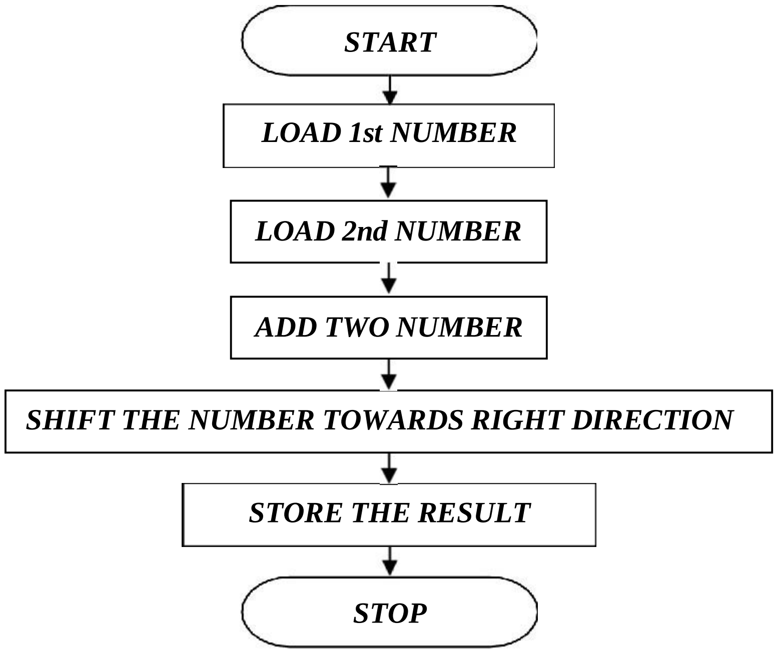
Experiment No: 9

Aim: To find the average of two number using assembly language. **Software Use:** EMU 8086

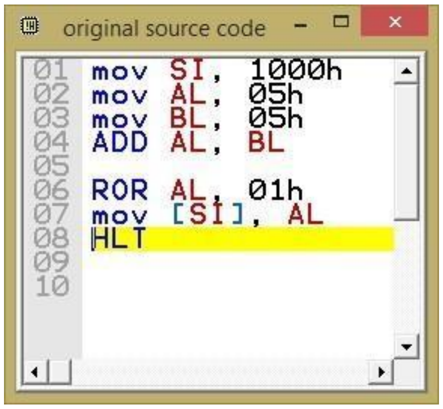
Program:

```
mov SI, 1000H
mov AL, 05H
mov BL, 05H
ADD AL, BL
ROR  AL
mov [SI], AL
HLT
```

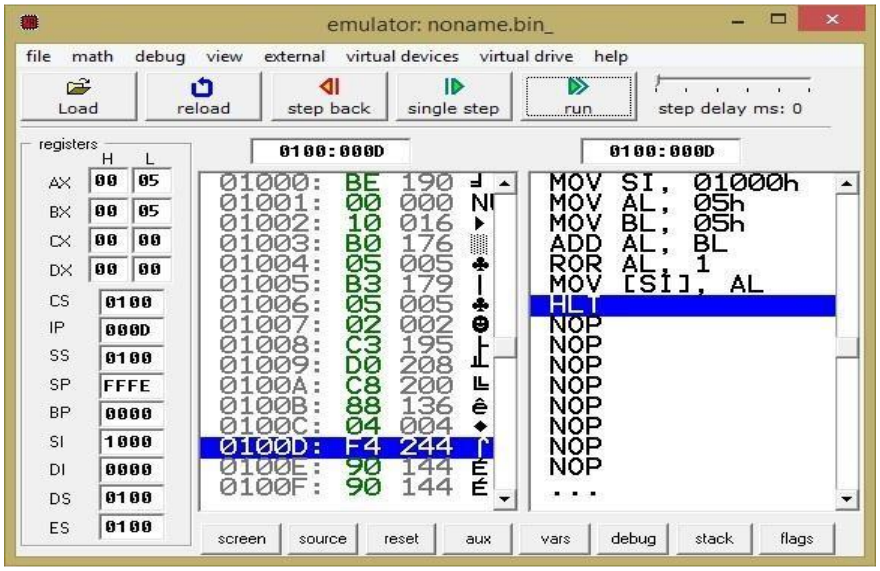
Flowchart:



Output:



```
mov SI, 1000h
mov AL, 05h
mov BL, 05h
ADD AL, BL

ROR AL, 01h
mov [SI], AL
HLT
```


emulator: noname.bin_
file math debug view external virtual devices virtual drive help
Load reload step back single step run
screen source reset aux vars debug stack flags

```
step delay ms: 0

registers
     H   L
AX  00  05
BX  00  05
CX  00  00
DX  00  00
CS  0100
IP  000D
SS  0100
SP  FFFE
BP  0000
SI  1000
DI  0000
DS  0100
ES  0100

0100:000D                      0100:000D
01000:  BE  190  ╛             MOV  SI,  01000h
01001:  00  000  NU            MOV  AL,  05h
01002:  10  016  ►             MOV  BL,  05h
01003:  B0  176  ░             ADD  AL,  BL
01004:  05  005  ♣             ROR  AL,  1
01005:  B3  179  │             MOV  [SI],  AL
01006:  05  005  ♣             HLT
01007:  02  002  ☻             NOP
01008:  C3  195  ├             NOP
01009:  D0  208  ╨             NOP
0100A:  C8  200  ╚             NOP
0100B:  88  136  ê             NOP
0100C:  04  004  ♦             NOP
0100D:  F4  244  ⌠             NOP
0100E:  90  144  É             NOP
0100F:  90  144  É             ...
```

Result:

Thus the average of two number using assembly language program has been performed and executed

Experiment No:- 10

Aim:- To interface LED with 8086

Total Marks(10)					**Total Marks**	**DOP**	**Sign**
A	**B**	**C**	**D**	**E**			
2	**3**	**2**	**2**	**1**			

Experiment No: 10

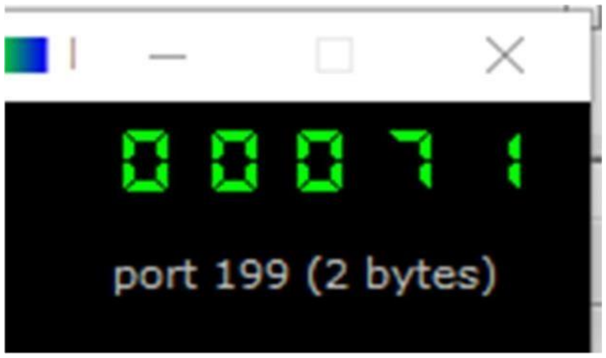
Aim: To find the average of two number using assembly language

Software Use: EMU 8086

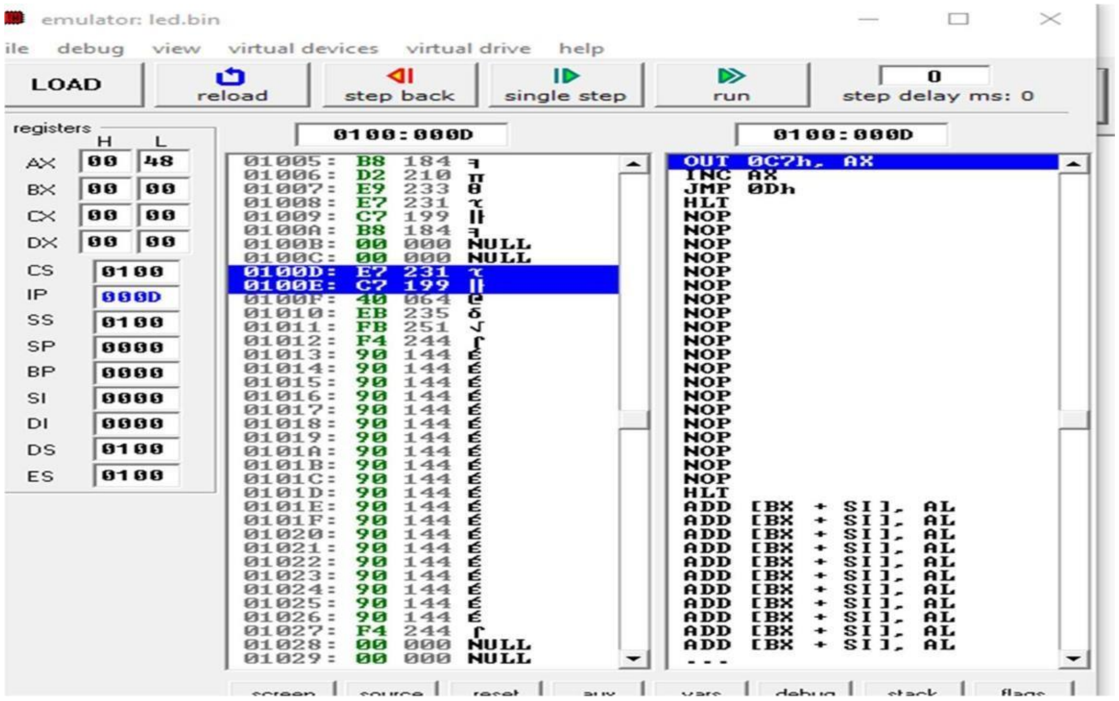
Program:

```
#start=led_display.exe#


name "led"

mov ax, 1234
out 199, ax

mov ax, -5678
out 199, ax

; Eternal loop to write
; values to port:
mov ax, 0
x1:
  out 199, ax
  inc ax
jmp x1

hlt
```

OUTPUT:



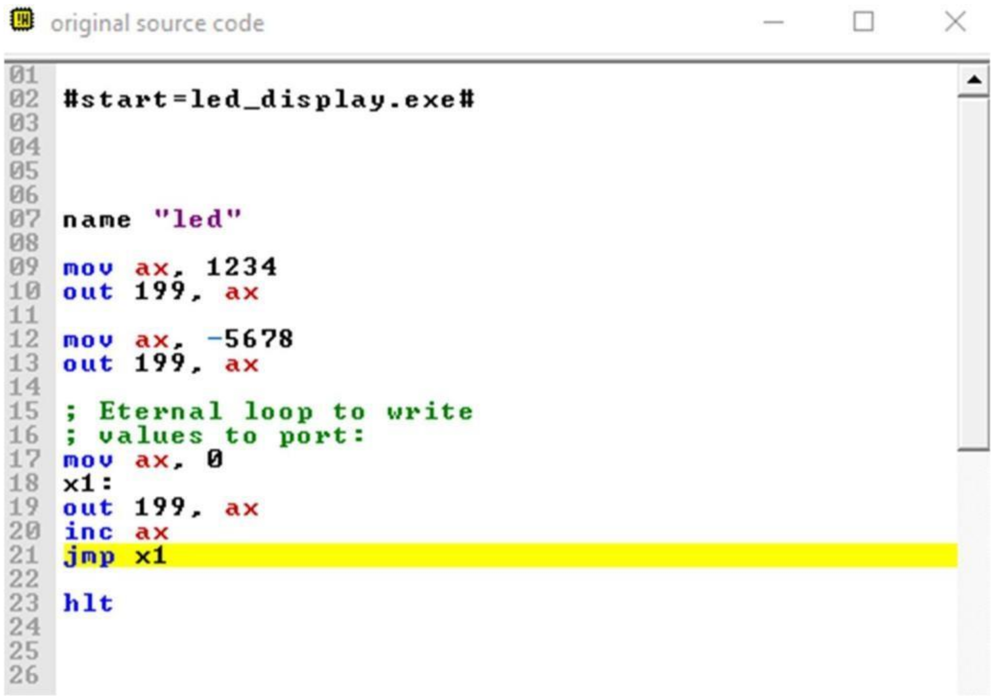
```
00071
port 199 (2 bytes)
```


emulator: led.bin
ile debug view virtual devices virtual drive help
LOAD
reload
step back
single step
run
step delay ms: 0
screen source reset aux vars debug stack flags

```
registers
     H    L
AX  00   48
BX  00   00
CX  00   00
DX  00   00
CS  0100
IP  000D
SS  0100
SP  0000
BP  0000
SI  0000
DI  0000
DS  0100
ES  0100
```

```
0100:000D

01005:  B8  184  ╕
01006:  D2  210  ╥
01007:  E9  233  Θ
01008:  E7  231  τ
01009:  C7  199  ╟
0100A:  B8  184  ╕
0100B:  00  000  NULL
0100C:  00  000  NULL
0100D:  E7  231  τ
0100E:  C7  199  ╟
0100F:  40  064  @
01010:  EB  235  δ
01011:  FB  251  √
01012:  F4  244  ⌠
01013:  90  144  É
01014:  90  144  É
01015:  90  144  É
01016:  90  144  É
01017:  90  144  É
01018:  90  144  É
01019:  90  144  É
0101A:  90  144  É
0101B:  90  144  É
0101C:  90  144  É
0101D:  90  144  É
0101E:  90  144  É
0101F:  90  144  É
01020:  90  144  É
01021:  90  144  É
01022:  90  144  É
01023:  90  144  É
01024:  90  144  É
01025:  90  144  É
01026:  90  144  É
01027:  F4  244  ⌠
01028:  00  000  NULL
01029:  00  000  NULL
```

```
0100:000D

OUT 0C7h, AX
INC AX
JMP 0Dh
HLT
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
HLT
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
...
```



```

#start=led_display.exe#




name "led"

mov ax, 1234
out 199, ax

mov ax, -5678
out 199, ax

; Eternal loop to write
; values to port:
mov ax, 0
x1:
out 199, ax
inc ax
jmp x1

hlt



```

Result:

Thus the average of two number using assembly language program has been performed and executed