American International University- Bangladesh

Department of Computer Science

Lab Report Cover Sheet

Course Name	MICROPROCESSOR AND EMBEDDED SYSTEMS
Lab Report No.	05
Lecturer Name	**MD. ALI NOOR**
Semester	SUMMER 2021-22
Submission Date	23/06/2022
Section	O
Group No.	03

Student Name	**Student ID**	**Contribution (out of 100%)**
1. EMON, MEDEDI HASAN	19-41118-2	
2. ASIF AHAMMED	19-40020-1	
3. EASINUL ABEDIN SAYEM	19-40291-1	
4. MD REDOY SHEIKH	19-40101-1	
5. MD. NAZIM HASAN	19-40118-1	
6. MOHI UDDIN ANANDO	20-43585-1	

Title: Familiarization of assembly language programs

Objective:

1. To get familiarized with assembly language and work with advanced 8086 instructions.
2. Understand the process of branching and looping instructions in assembly language
3. Write, compile and execute flow control structures in assembly language programs using EMU 8086

Introduction:

The microprocessor 8086 serves as the foundation for the Intel X-86 family of processors. With this 16-bit processor under one's belt, one can investigate the 80386, 80406, and Pentium processors.

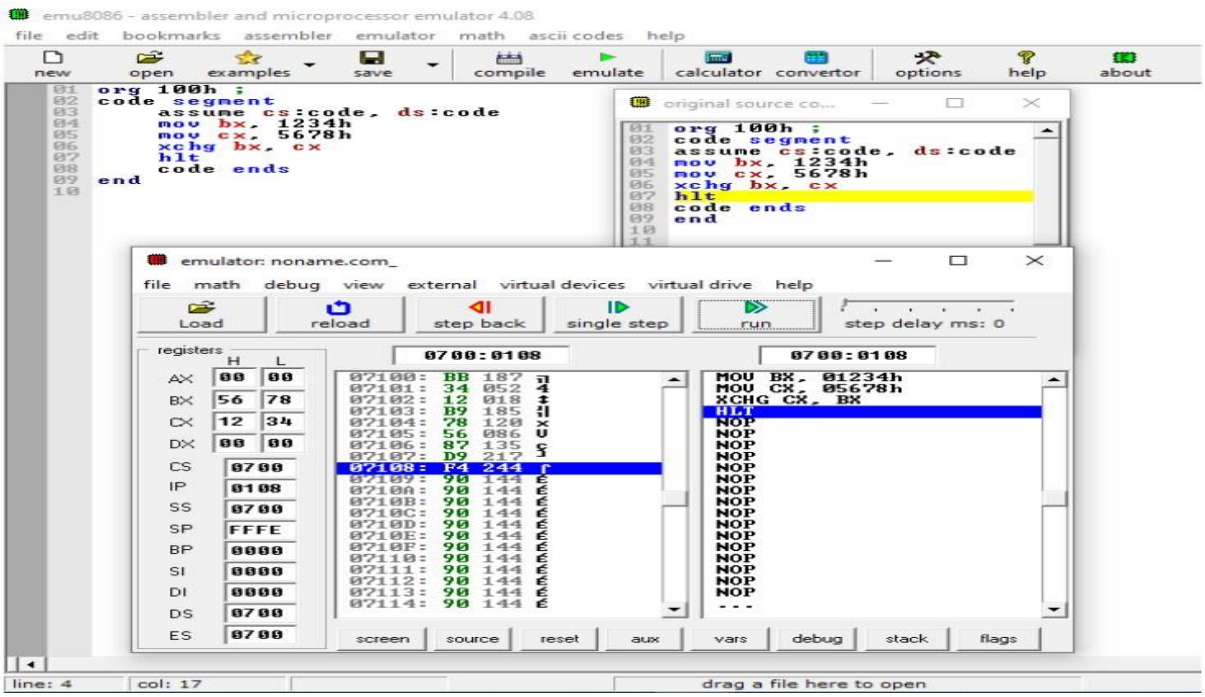
"MTS-86c" and "MDA 8086" are the micro-kits we're employing. There are numerous instructions in the 8086. In this experiment, the main objective is to learn how to write assembly programs using 8086 instructions and arrays.

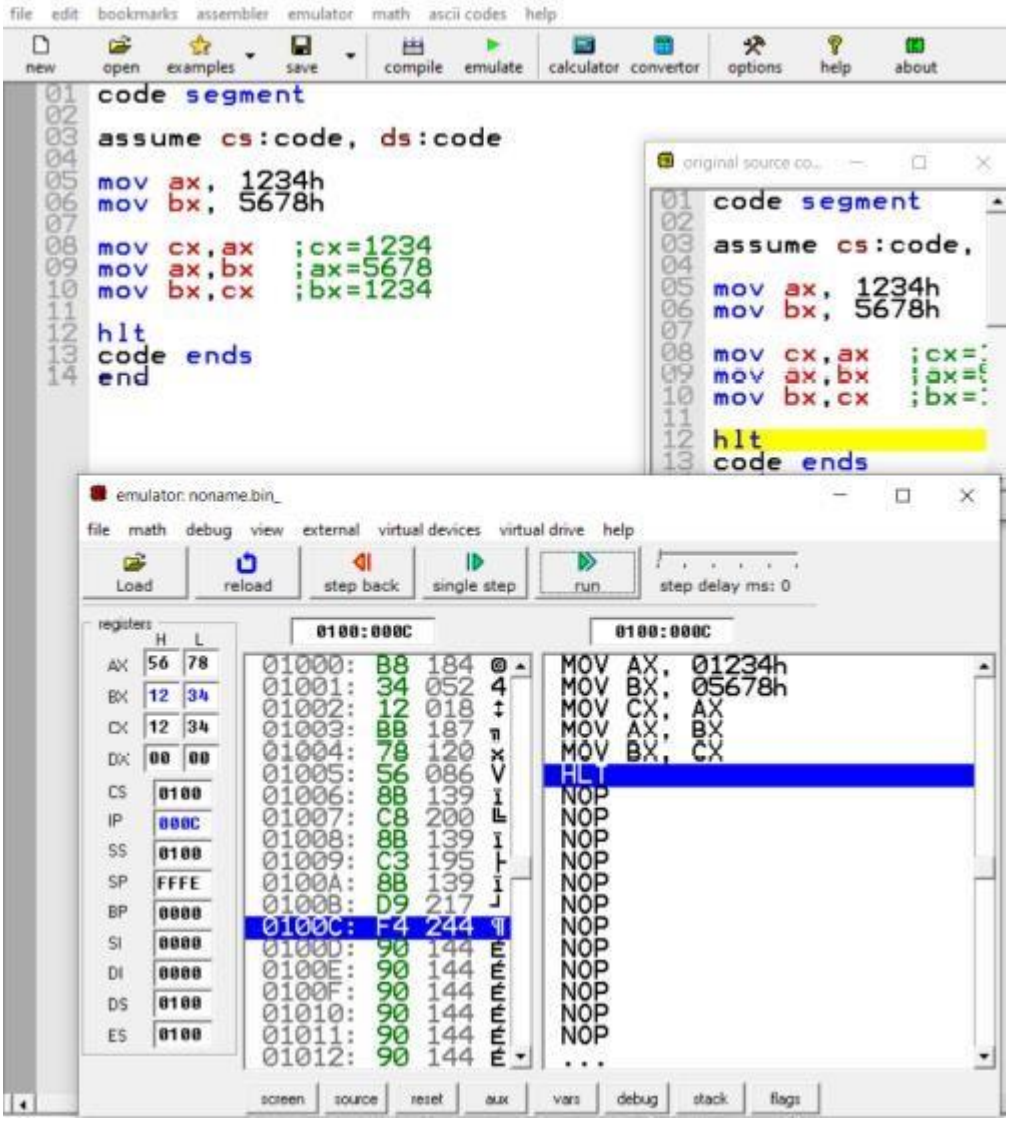
Apparatus:

Serial No.	Components	Rating	Quantity
1	EMU8086	[ver.408 (32 bit WINOS compatible)]	1
2	PC having Intel Microprocessor		1

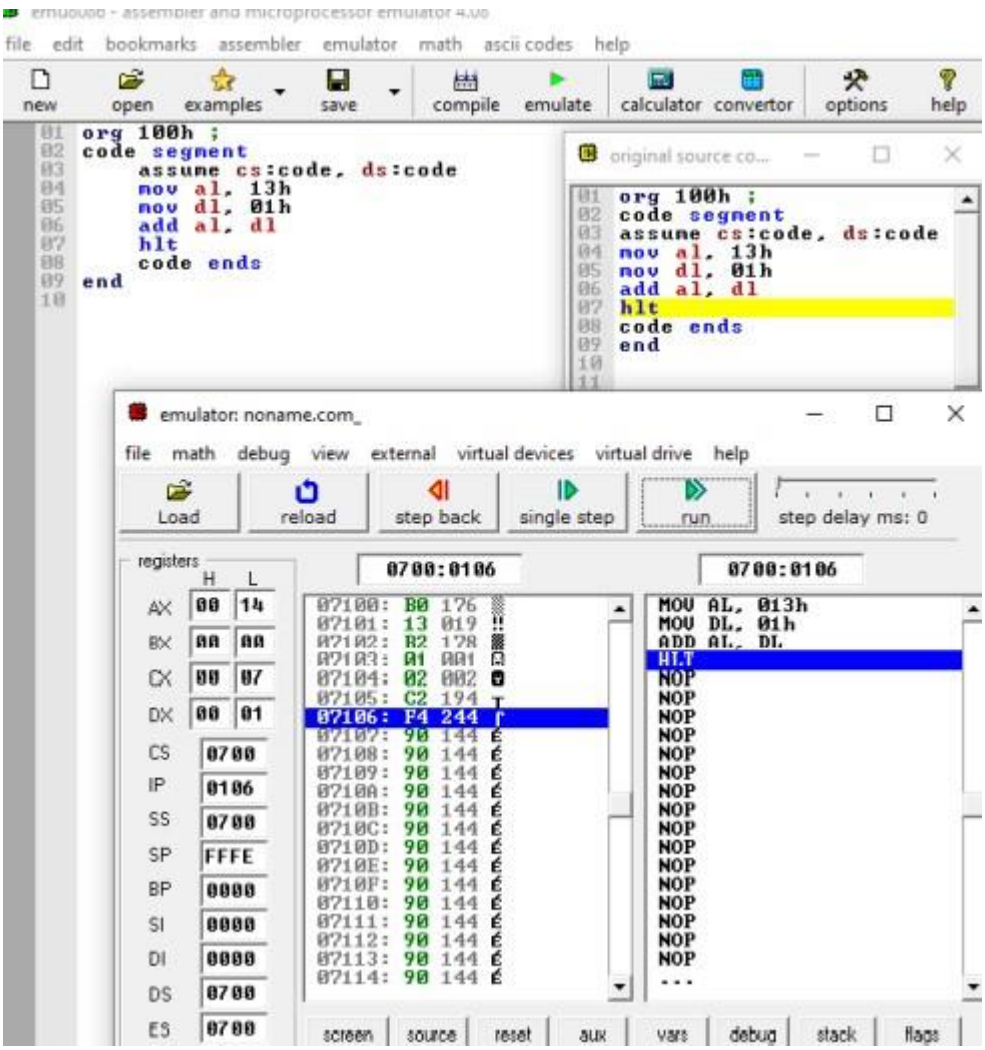
Lab Task(Code/Simulation):

Write a program to exchange the contents of two registers.

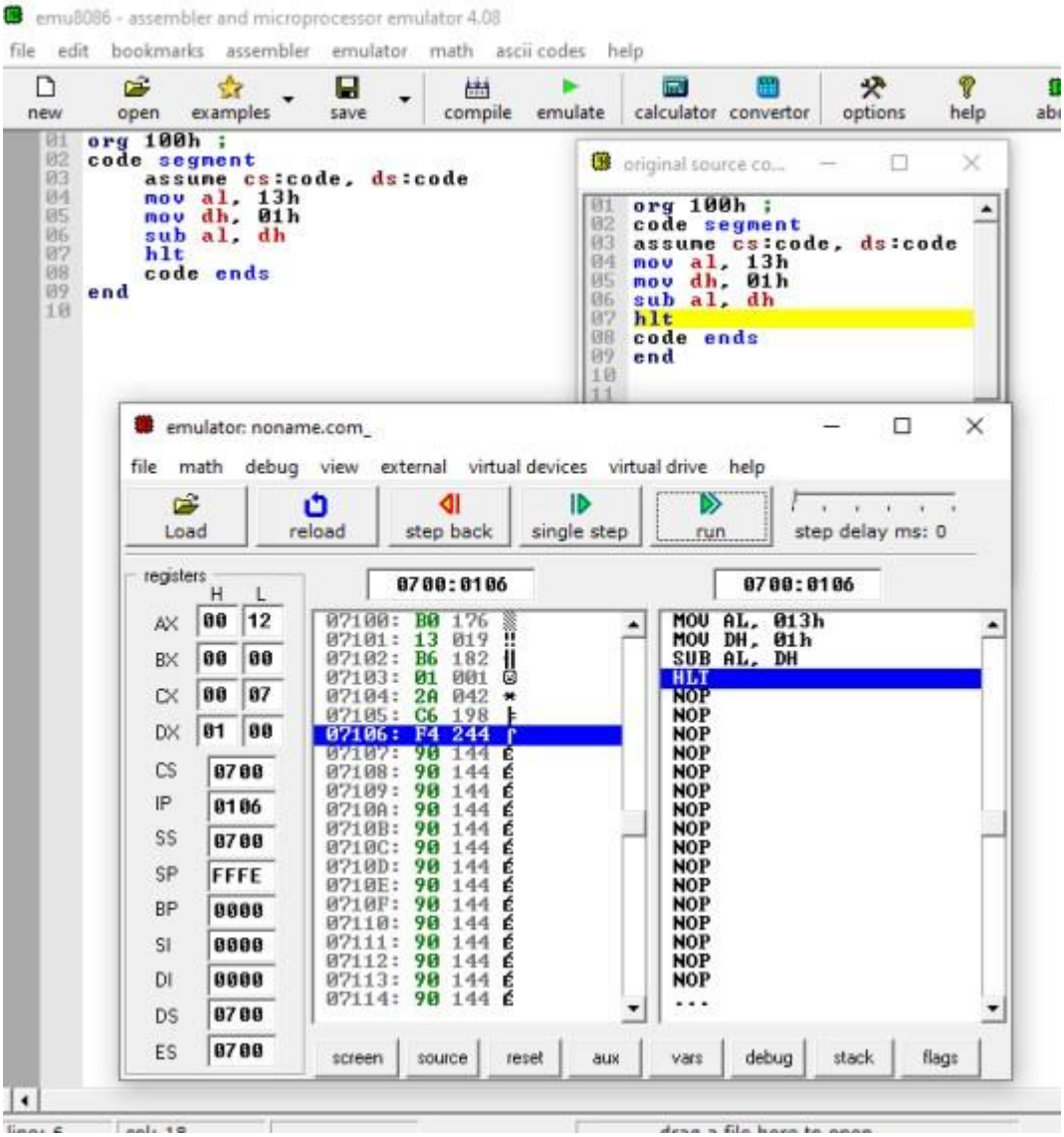



file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
code segment
assume cs:code, ds:code
mov ax, 1234h
mov bx, 5678h
mov cx,ax ;cx=1234
mov ax,bx ;ax=5678
mov bx,cx ;bx=1234
hlt
code ends
end
original source co...
emulator: noname.bin_
file math debug view external virtual devices virtual drive help
Load reload step back single step run step delay ms: 0
registers
H L
AX 56 78
BX 12 34
CX 12 34
DX 00 00
CS 0100
IP 000C
SS 0100
SP FFFE
BP 0000
SI 0000
DI 0000
DS 0100
ES 0100
0100:000C
0100:000C
01000: B8 184
01001: 34 052 4
01002: 12 018
01003: BB 187
01004: 78 120 x
01005: 56 086 V
01006: 8B 139
01007: C8 200
01008: 8B 139
01009: C3 195
0100A: 8B 139
0100B: D9 217
0100C: F4 244
0100D: 90 144
0100E: 90 144
0100F: 90 144
01010: 90 144
01011: 90 144
01012: 90 144
MOV AX, 01234h
MOV BX, 05678h
MOV CX, AX
MOV AX, BX
MOV BX, CX
HLT
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
screen source reset aux vars debug stack flags

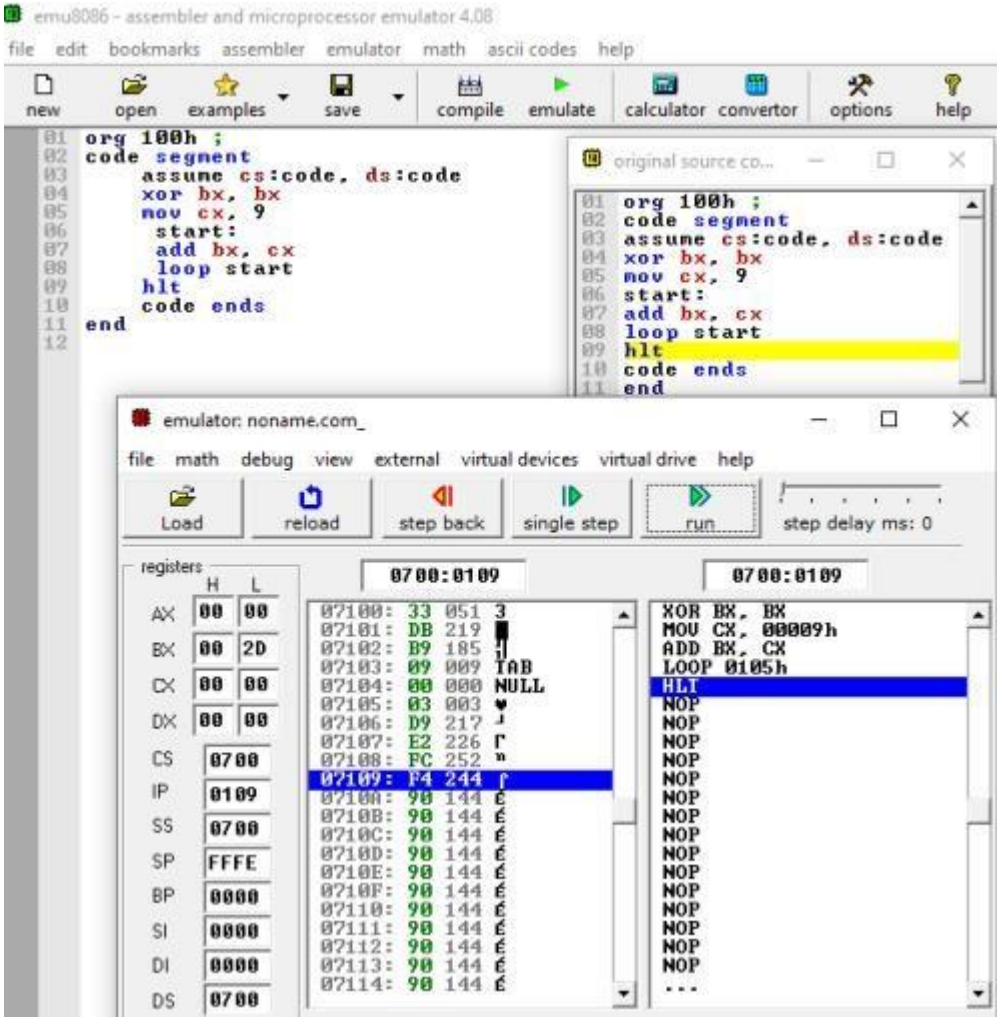
Write a program for adding two numbers.



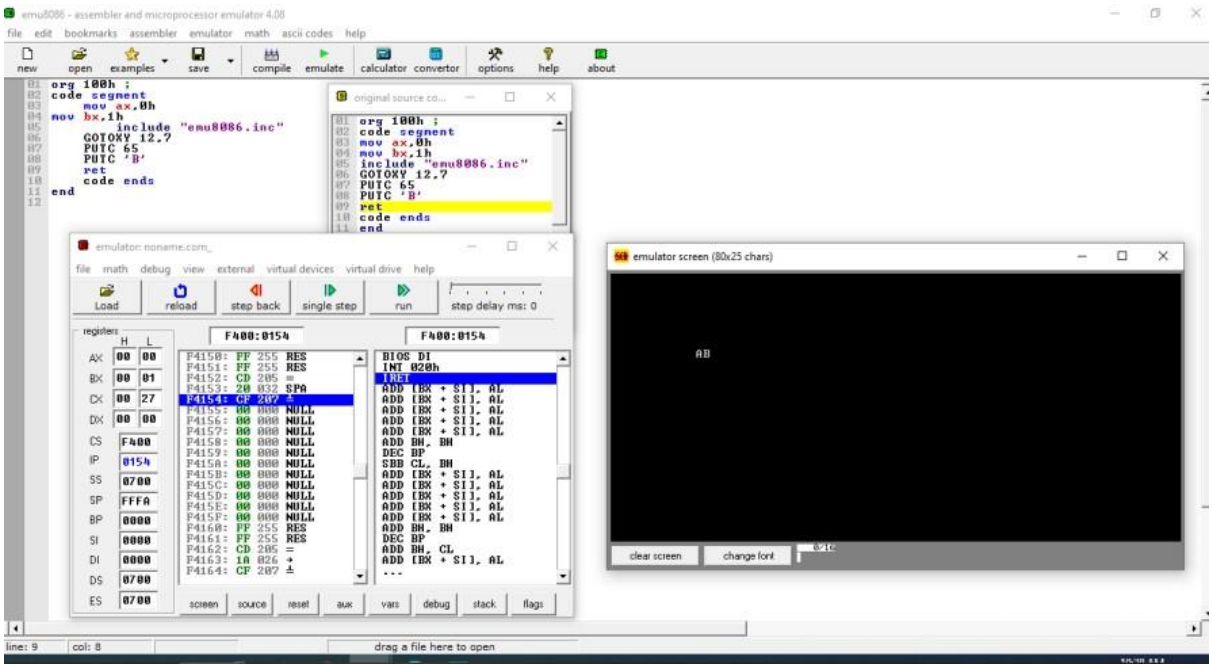
Write a program for subtraction between two numbers



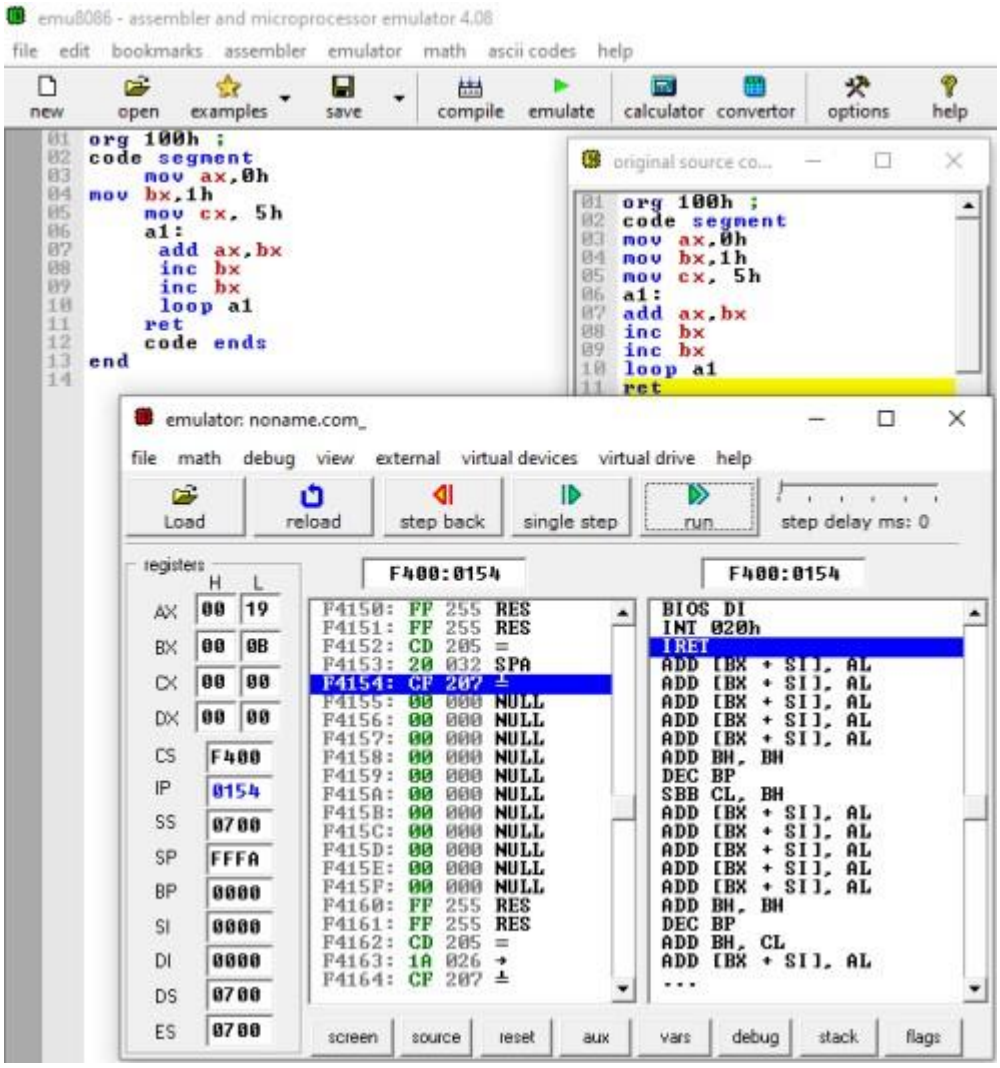
Summation of a series: [1+2+3+4+……+N] = BX . The value of N is stored in CX.



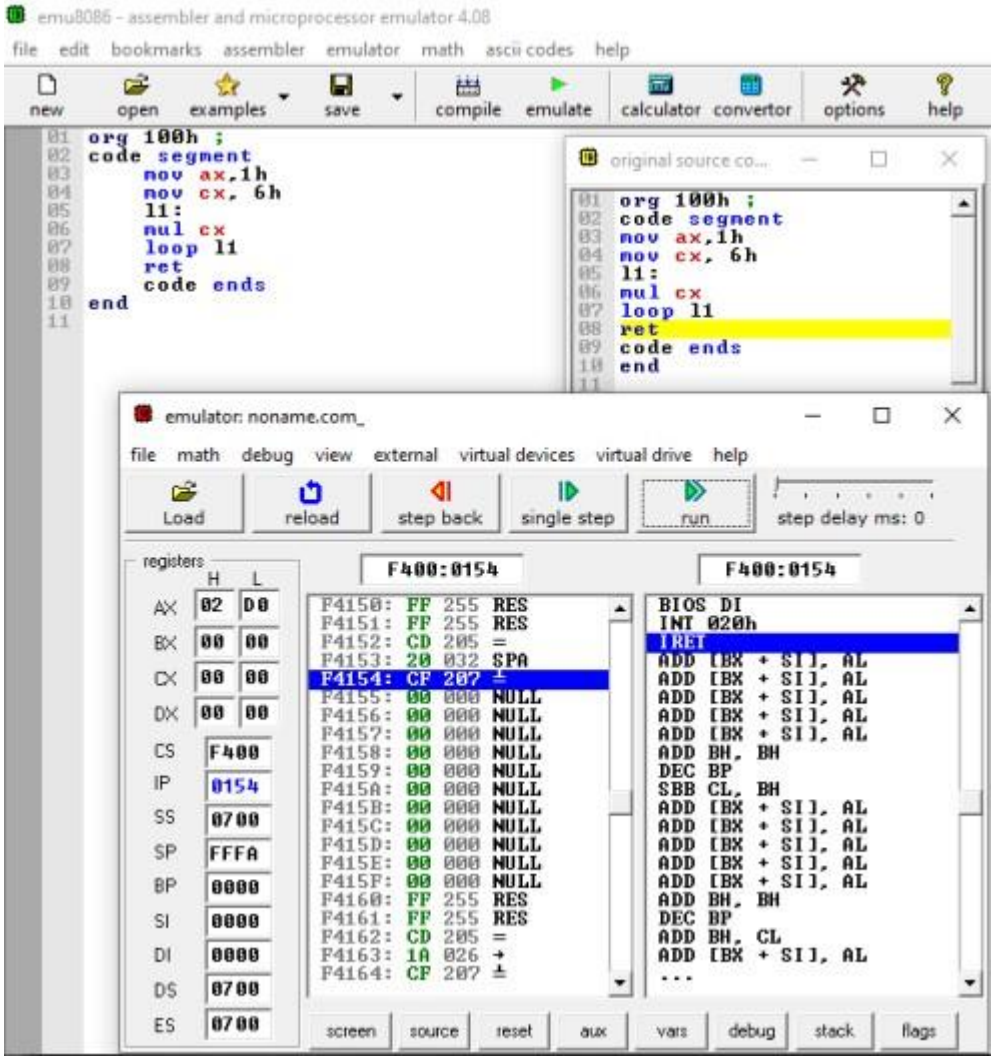
Write a program which display two characters at column#12 and row#7 at emulator screen.



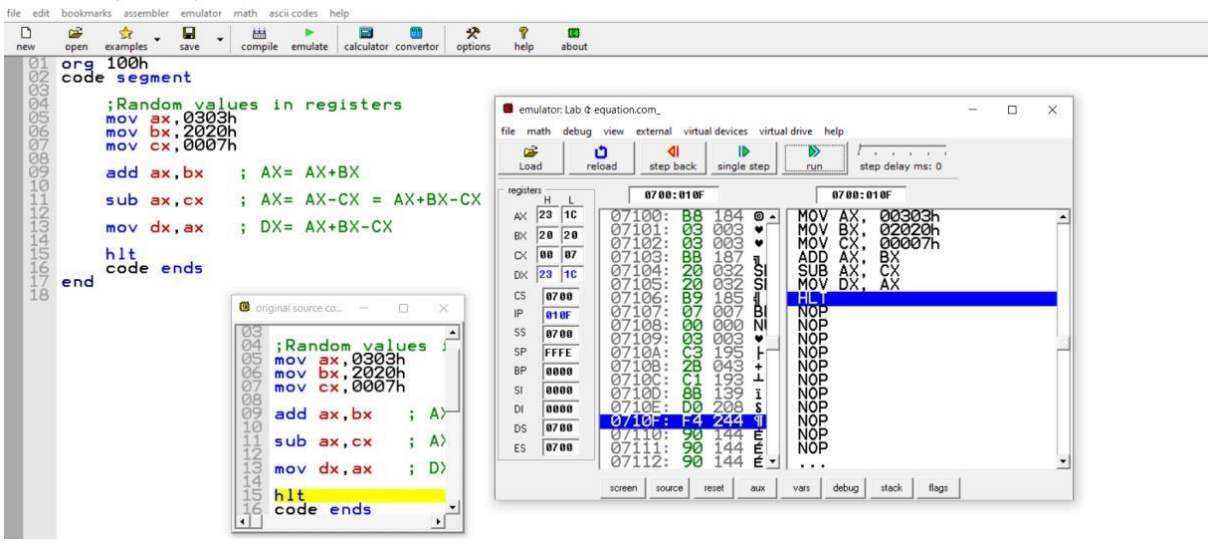
Write the assembly code for the following sequence 1+3+5+7…..+N. Where N = 5 using a loop.



Write a code for finding the value of 6!


emu8086 - assembler and microprocessor emulator 4.08
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help
org 100h ;
code segment
mov ax,1h
mov cx, 6h
l1:
mul cx
loop l1
ret
code ends
end
original source co...
emulator: noname.com_
file math debug view external virtual devices virtual drive help
Load reload step back single step run step delay ms: 0
registers
AX 02 D0
BX 00 00
CX 00 00
DX 00 00
CS F400
IP 0154
SS 0700
SP FFFA
BP 0000
SI 0000
DI 0000
DS 0700
ES 0700
F400:0154
F4150: FF 255 RES
F4151: FF 255 RES
F4152: CD 205 =
F4153: 20 032 $PA
F4154: CF 207 ±
F4155: 00 000 NULL
F4156: 00 000 NULL
F4157: 00 000 NULL
F4158: 00 000 NULL
F4159: 00 000 NULL
F415A: 00 000 NULL
F415B: 00 000 NULL
F415C: 00 000 NULL
F415D: 00 000 NULL
F415E: 00 000 NULL
F415F: 00 000 NULL
F4160: FF 255 RES
F4161: FF 255 RES
F4162: CD 205 =
F4163: 1A 026 →
F4164: CF 207 ±
F400:0154
BIOS DI
INT 020h
IRET
ADD [BX + SI], AL
ADD BH, BH
DEC BP
SBB CL, BH
ADD BH, CL
screen source reset aux vars debug stack flags

Homework:


org 100h
code segment
;Random values in registers
mov ax,0303h
mov bx,2020h
mov cx,0007h
add ax,bx ; AX= AX+BX
sub ax,cx ; AX= AX-CX = AX+BX-CX
mov dx,ax ; DX= AX+BX-CX
hlt
code ends
end
emulator: Lab 6 equation.com_
MOV AX, 0303h
MOV BX, 2020h
MOV CX, 0007h
ADD AX, BX
SUB AX, CX
MOV DX, AX
HLT
NOP

Discussion:

While writing each program, we learnt the reason for using each function and the associated values. We were able to understand each line of code, which allowed us to solve a specific problem or task using the knowledge we've gained from this experiment.