BANGLADESH UNIVERSITY OF PROFESSIONALS

FACULTY OF SCIENCE & TECHNOLOGY

DEPT. OF COMPUTER SCIENCE & ENGINEERING (CSE)

Lab Report
NO - 01

COURSE NAME: Microprocessors, Microcontrollers, and Assembly Language Laboratory

COURSE CODE: CSE-3106

TITLE : Introduction to 8086 Microprocessor Architecture & Assembly Language

SUBMITTED BY:

NAME : Md Spondon Sarwar

ROLL : 2252421017

SECTION : A

LEVEL/TERM : 3rd

SEMESTER : 1st

DATE OF SUBMISSION: November 3, 2024

Experiment No : 01

Name of the Experiment : Introduction to 8086 Microprocessor Architecture & Assembly Language

Lab Questions :

1. **Write an assembly program that initializes the AX register with the value 1234H and the BX register with 0F12H. Then, add BX to AX and subtract 567H from the result. Store the final result in the CX register.**

2. Write an assembly program that initializes the AX register with 1234H. Then, triple the value in AX by using addition instructions. Finally, store the result in the BX register.

3. Create a program that initializes AX with 1000H, BX with 2000H, and CX with 3000H. Calculate the sum of these three registers and store the result in the DX register.

Code :

1. **Solution :**

```
CODE SEGMENT
   ASSUME CS:CODE, DS:CODE
   MOV AX, 1234H
   MOV BX, 0F12H
   ADD AX, BX
   SUB AX, 567H
   MOV CX, AX
HLT
CODE ENDS
END
```

2. **Solution :**

```
CODE SEGMENT
   ASSUME CS:CODE, DS:CODE
   MOV AX, 1234H
   ADD AX, AX
   ADD AX, AX
   MOV BX, AX
```

```
HLT
CODE ENDS
END
```

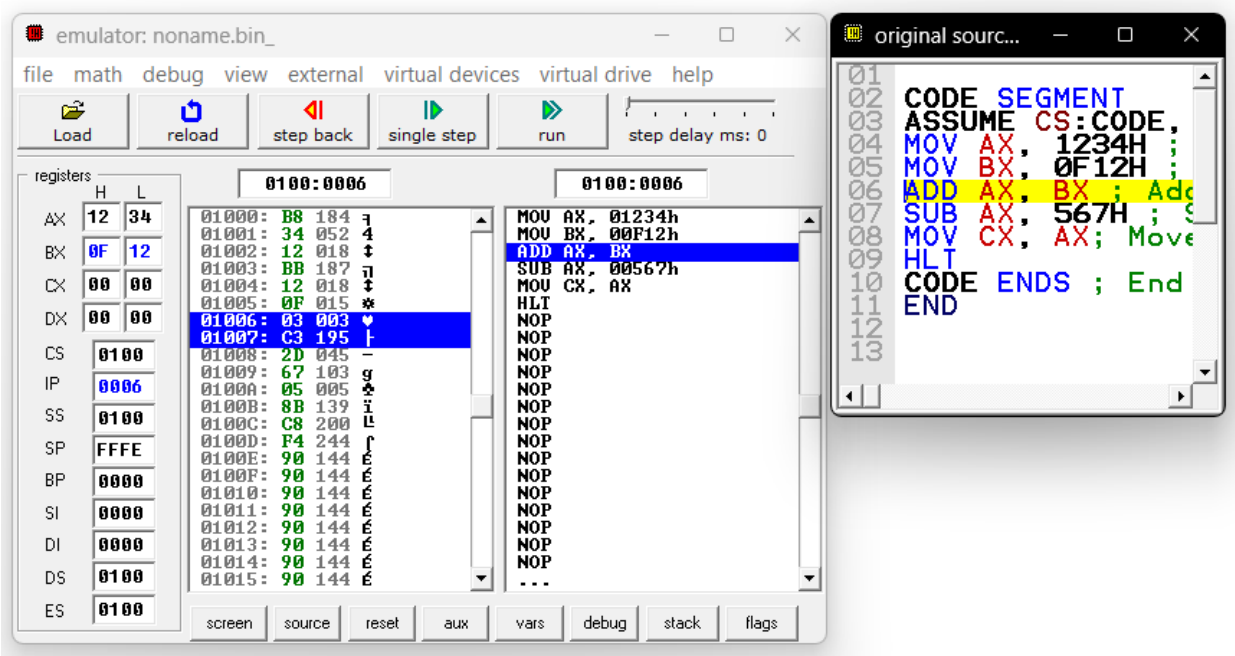
3. Solution :

```
CODE SEGMENT
    ASSUME CS:CODE, DS:CODE
    MOV AX, 1000H
    MOV BX, 2000H
    MOV CX, 3000H
    ADD BX, AX ; BX = AX + BX
    ADD CX, BX ; CX = CX + BX
    MOV DX, CX
HLT
CODE ENDS
END
```

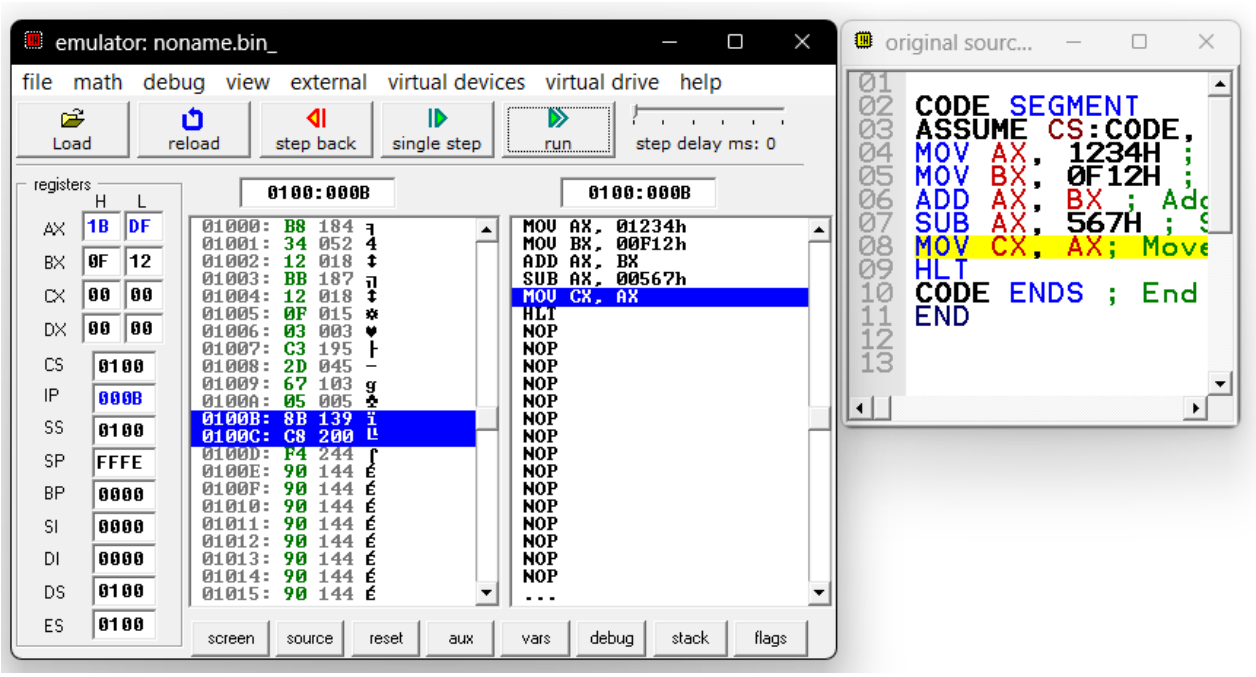
Output : The attached images are conveyed in the following manner : Value Initialization , Halfway of Operations and finally Final Output.

Solution 1 :

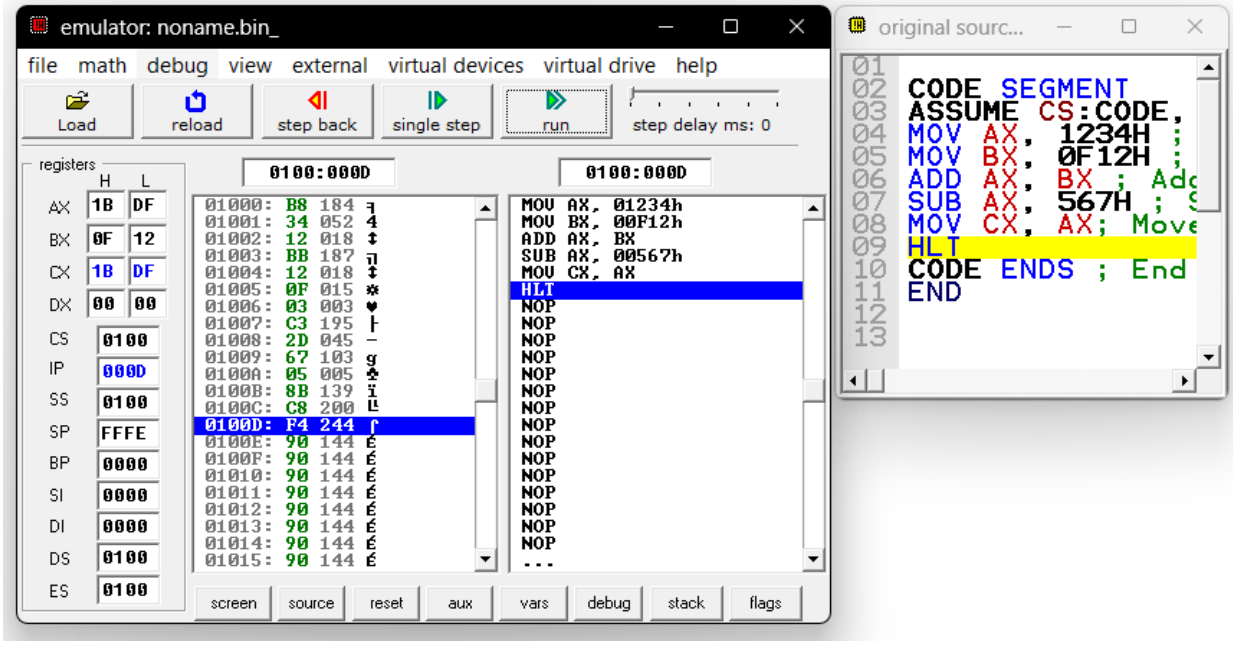
AX = 1234H, BX = 0F12H



AX = AX + BX - 567H = 1BDFH

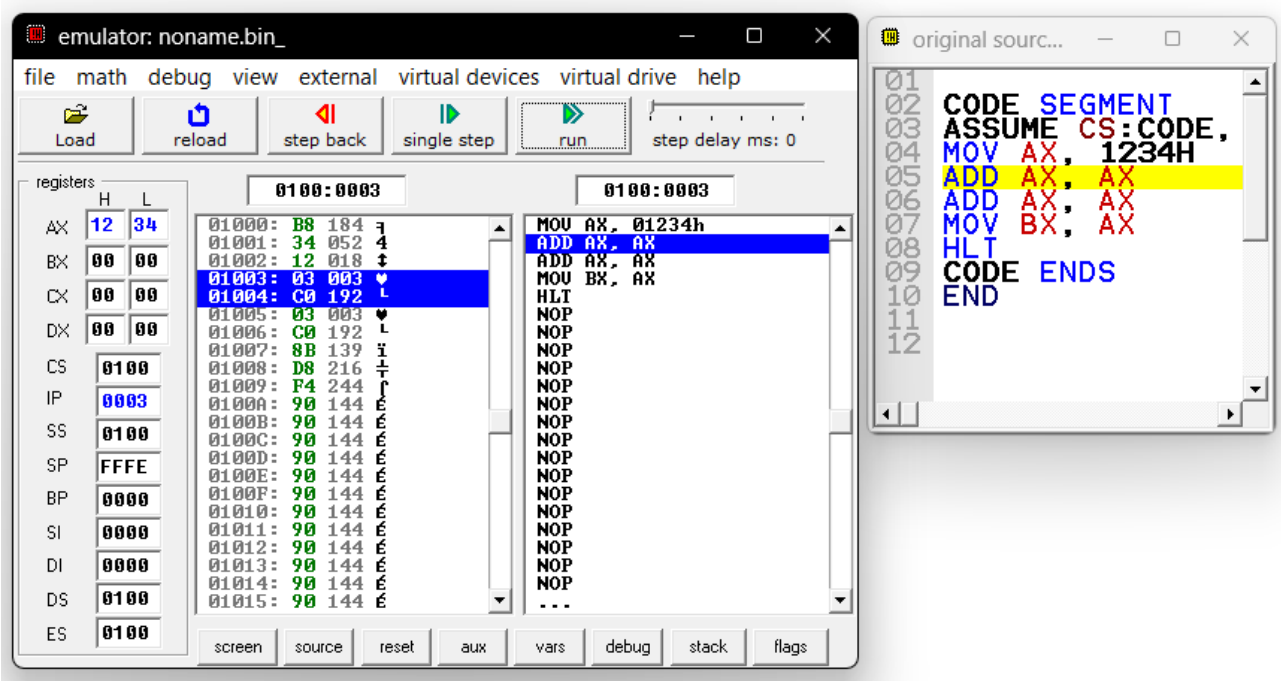


CX = AX = 1BDFH



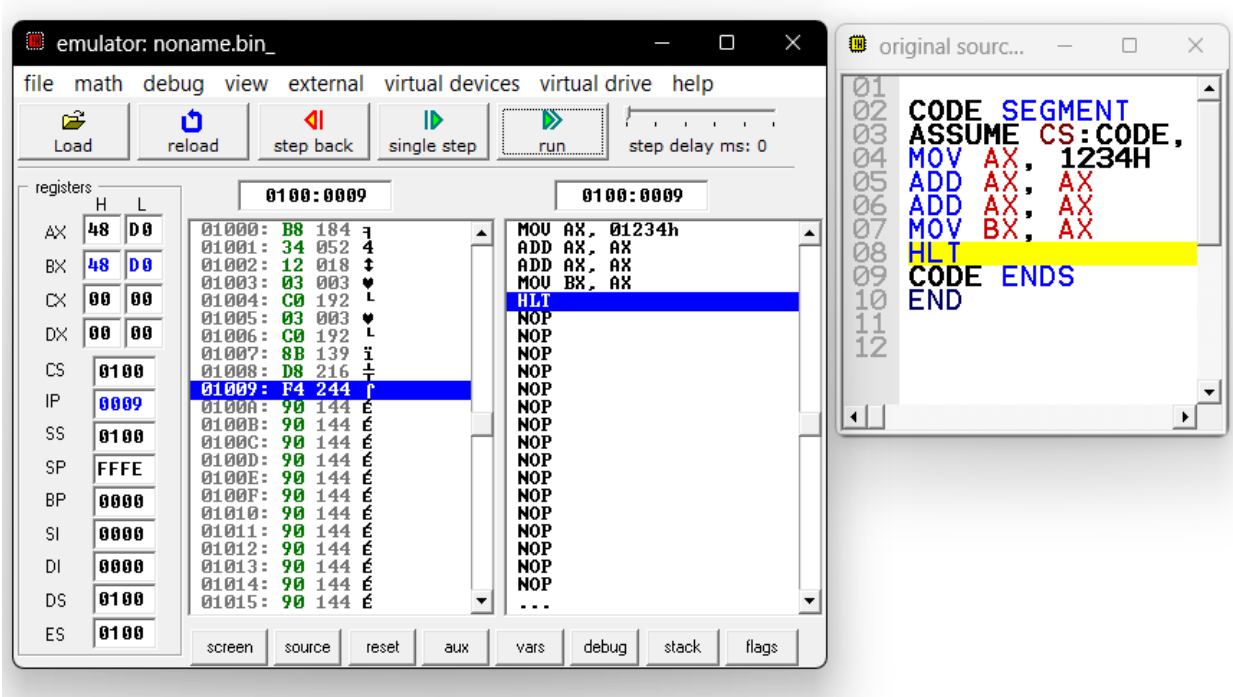
Solution 2 :

AX = 1234H



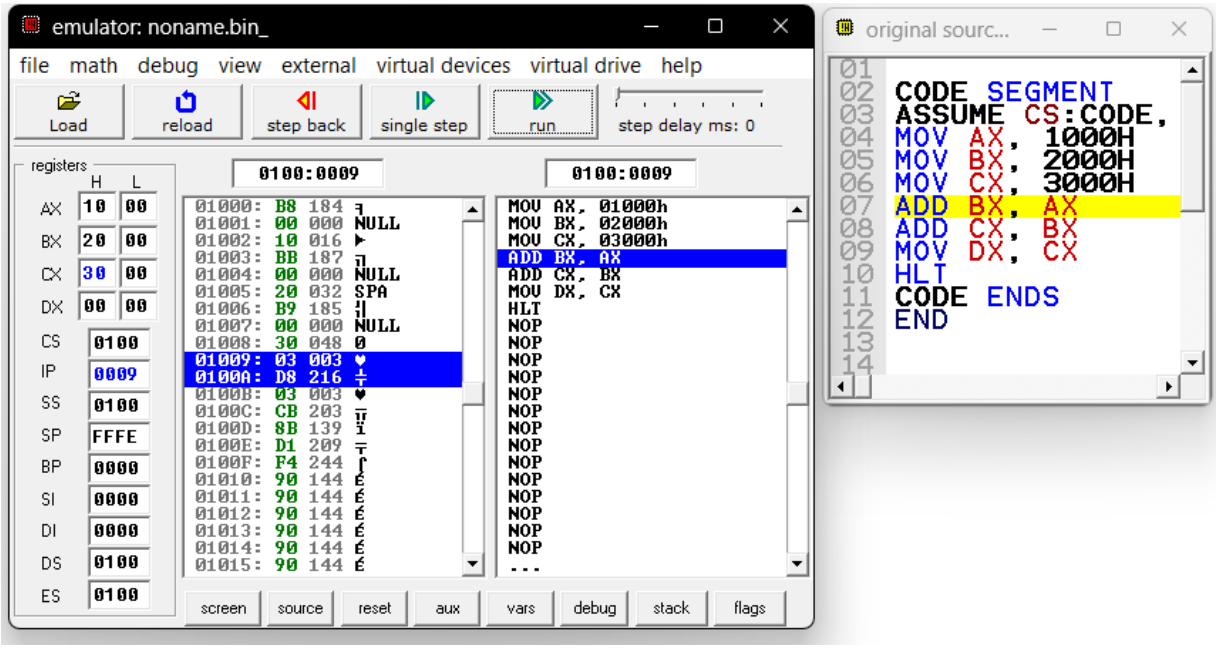
AX = AX + AX + AX = 3AX = 48D0H

BX = AX = 48D0H



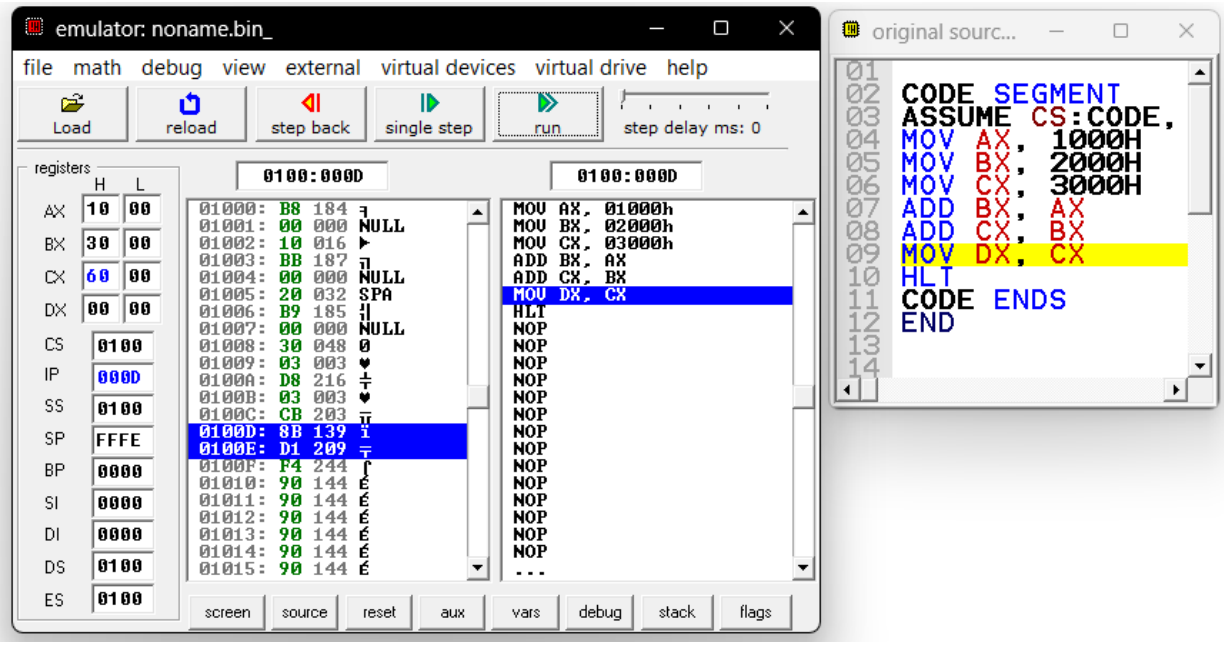
Solution 3 :

AX = 1000H , BX = 2000H, CX = 3000H

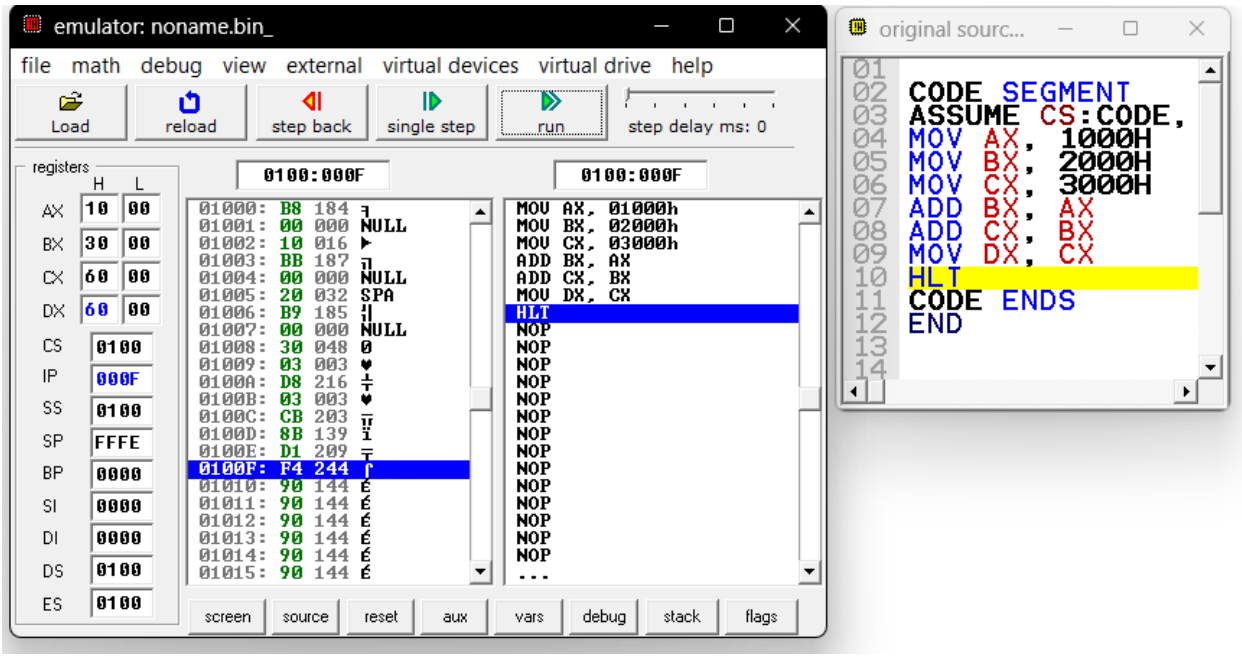


BX = BX + AX = 3000H

CX = CX + BX = 6000H



DX = CX


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 10 00
BX 30 00
CX 60 00
DX 60 00
CS 0100
IP 000F
SS 0100
SP FFFE
BP 0000
SI 0000
DI 0000
DS 0100
ES 0100
0100:000F
01000: B8 184 ╕
01001: 00 000 NULL
01002: 10 016 ►
01003: BB 187 ╗
01004: 00 000 NULL
01005: 20 032 SPA
01006: B9 185 ╣
01007: 00 000 NULL
01008: 30 048 0
01009: 03 003 ♥
0100A: D8 216 ╪
0100B: 03 003 ♥
0100C: CB 203 ╦
0100D: 8B 139 ï
0100E: D1 209 ╤
0100F: F4 244 ⌠
01010: 90 144 É
01011: 90 144 É
01012: 90 144 É
01013: 90 144 É
01014: 90 144 É
01015: 90 144 É
0100:000F
MOV AX, 01000h
MOV BX, 02000h
MOV CX, 03000h
ADD BX, AX
ADD CX, BX
MOV DX, CX
HLT
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
screen source reset aux vars debug stack flags
original sourc...
01
02 CODE SEGMENT
03 ASSUME CS:CODE,
04 MOV AX, 1000H
05 MOV BX, 2000H
06 MOV CX, 3000H
07 ADD BX, AX
08 ADD CX, BX
09 MOV DX, CX
10 HLT
11 CODE ENDS
12 END
13
14