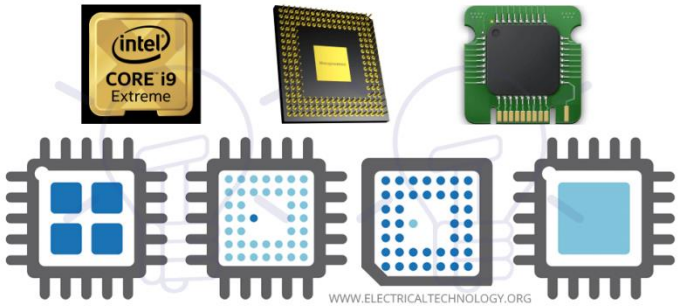



# MİKROİŞLEMCİLER

Dr. Meltem KURT PEHLİVANOĞLU

W-10

# MİKROİŞLEMCİLER

**Digital Logic** +

**Digital Design** +

**Computer Architecture** +

**Microprocessors** +

**Microcontrollers** +

**Assembly Language Programming**

# 8086 16-Bit Mikroişlemci

- Segment ve adres register çiftleri:
- CS:IP
- SS:SP SS:BP
- DS:BX DS:SI
- DS:DI
- ES:DI

# 8086 16-Bit Mikroişlemci

## EMU 8086-MICROPROCESSOR EMULATOR

**LEA (Load Effective Address)KOMUTU**: Register,Memory

**LEA** operand1,operand2: **operand2** de gösterilen **bellek hücresinin adresini operand1** e aktarır.

- **Başlangıç adres** tutucuları **(BX,BP)** ve **indeks (SI,DI)** registerları kullanılmalıdır.
- **LEA: MOV** operand1, **offset** operand2 ifadesi ile aynı işlemi yapar (buradaki offset anahtar kelimesi ile direk operand2 nin efektif adresini alabilirsiniz)

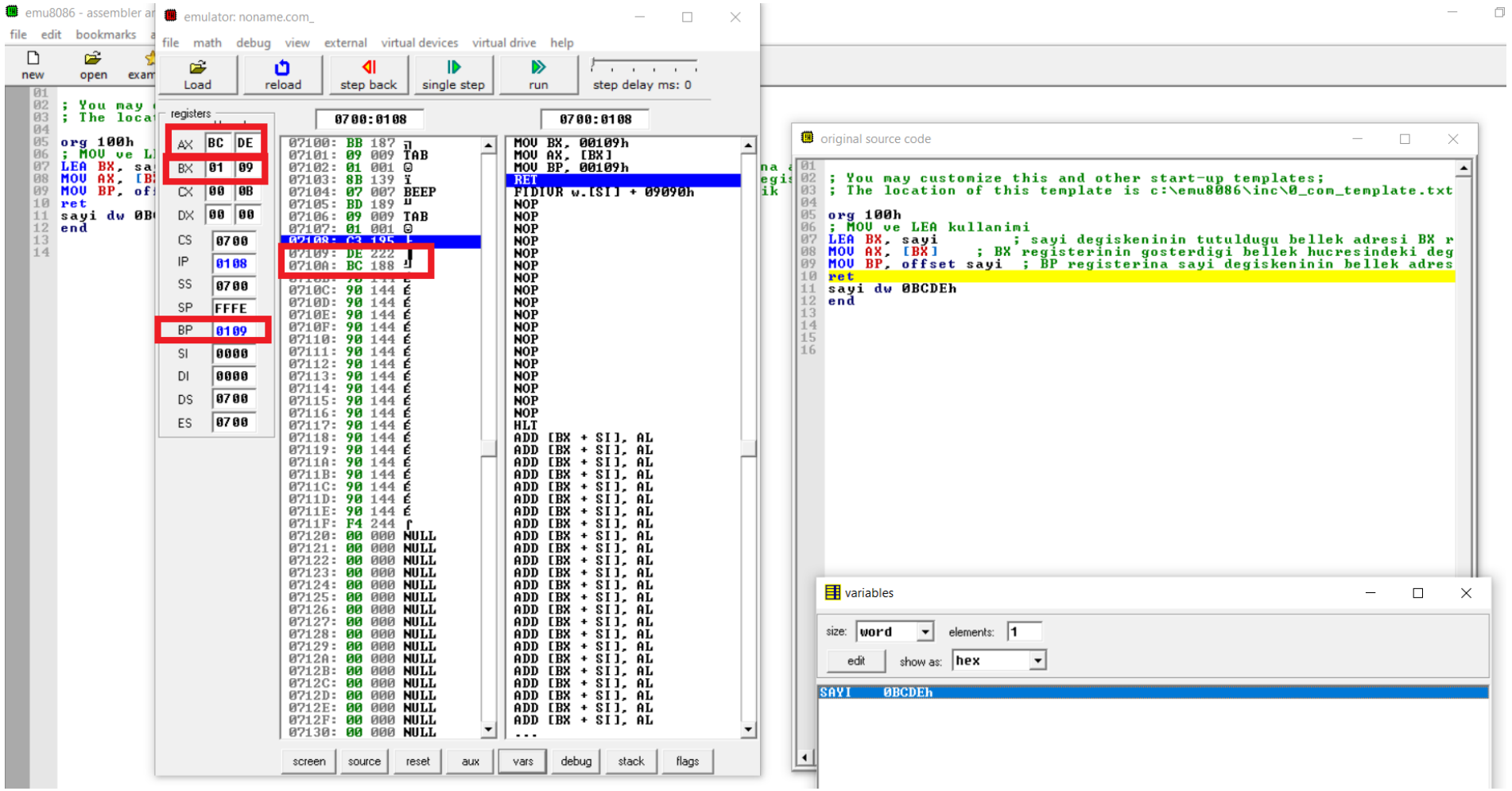
# 8086 16-Bit Mikroişlemci

## EMU 8086-MICROPROCESSOR EMULATOR

```
org 100h
; MOV ve LEA kullanimi
LEA BX, sayi        ; sayi degiskeninin tutuldugu bellek adresi BX registerina atandi
MOV AX, [BX]    ; BX registerinin gosterdigi bellek hucresindeki degeri AX registerina atadik
MOV BP, offset sayi  ; BP registerina sayi degiskeninin bellek adresini atadik
ret
sayi dw 0BCDEh
end
```

# 8086 16-Bit Mikroişlemci

## EMU 8086-MICROPROCESSOR EMULATOR



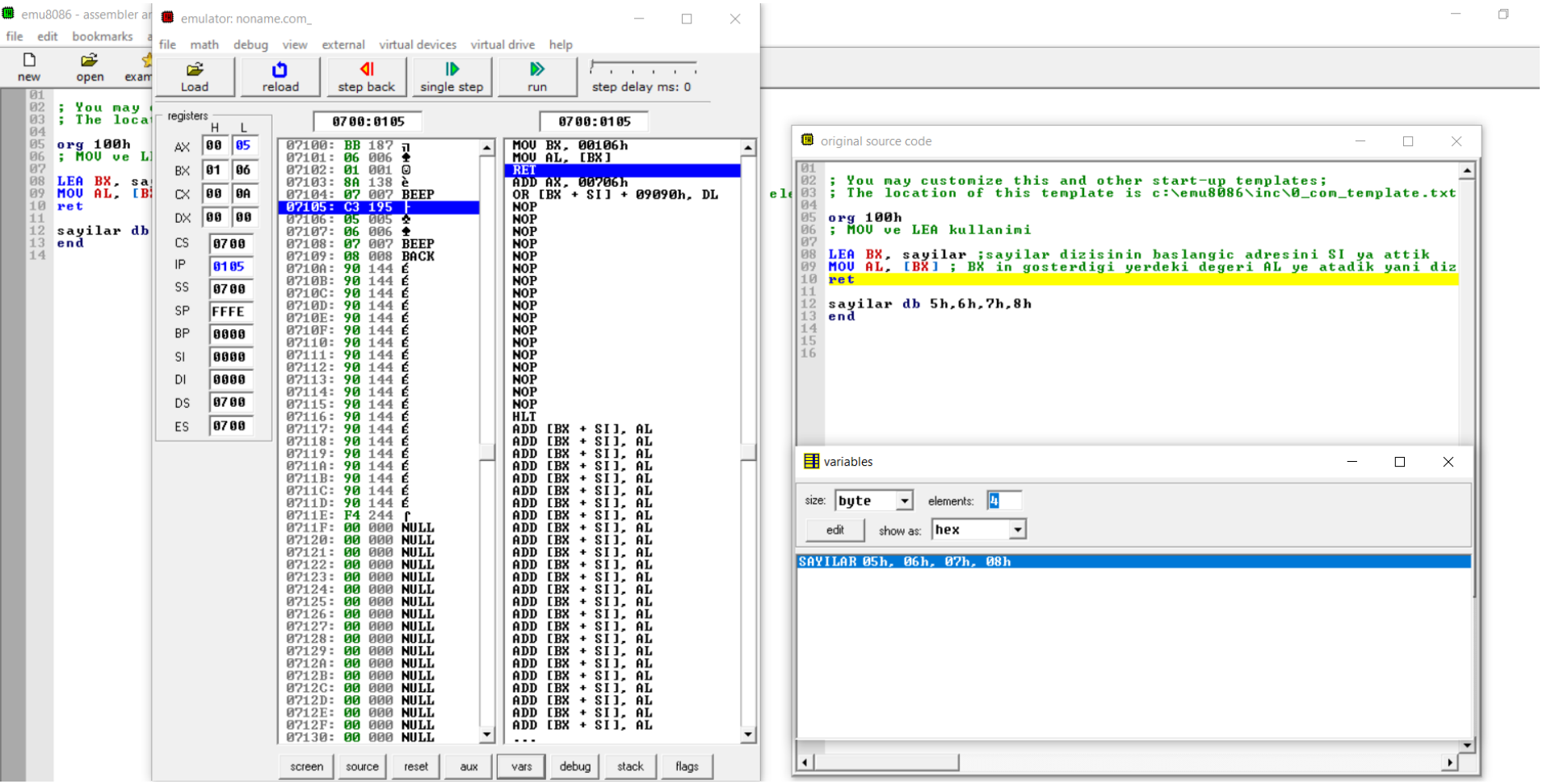
# 8086 16-Bit Mikroişlemci

## EMU 8086-MICROPROCESSOR EMULATOR

```
org 100h
; MOV ve LEA kullanimi

LEA BX, sayilar ;sayilar dizisinin baslangic adresini SI ya attik
MOV AL, [BX] ; BX in gosterdigi yerdeki degeri AL ye atadik yani dizinin ilk elemani
ret

sayilar db 5h,6h,7h,8h
end
```

# 8086 16-Bit Mikroişlemci

## EMU 8086-MICROPROCESSOR EMULATOR



# 8086 16-Bit Mikroişlemci

## EMU 8086-MICROPROCESSOR EMULATOR

;dizide bir sonraki adrese ulaşma:

MOV AL, [SI+1] ; SI nin gosterdigi yeri 1 arttır ordaki degeri AL ye atadik (byte tipinde olduğu için +1)

MOV AL, [SI+2] ; word tipinde olduğu için +2 yapmalısın çünkü word tipindeki her değişken 2 byte yer kaplıyor bellekte

**NOT: DİZİNİN İNDİS DEĞERİ 0 DAN BAŞLAR!!**

# 8086 16-Bit Mikroişlemci

## EMU 8086-MICROPROCESSOR EMULATOR

**XCHG (exchange) KOMUTU:**

operand1, operand2

**XCHG** operand1,operand2: **operand1 ve operand2 değerleri birbirleriyle yer değiştirir**

- Tek seferde iki değeri yer değiştirmek için kullanılır

REG, memory

Memory, REG

REG,REG

# 8086 16-Bit Mikroişlemci

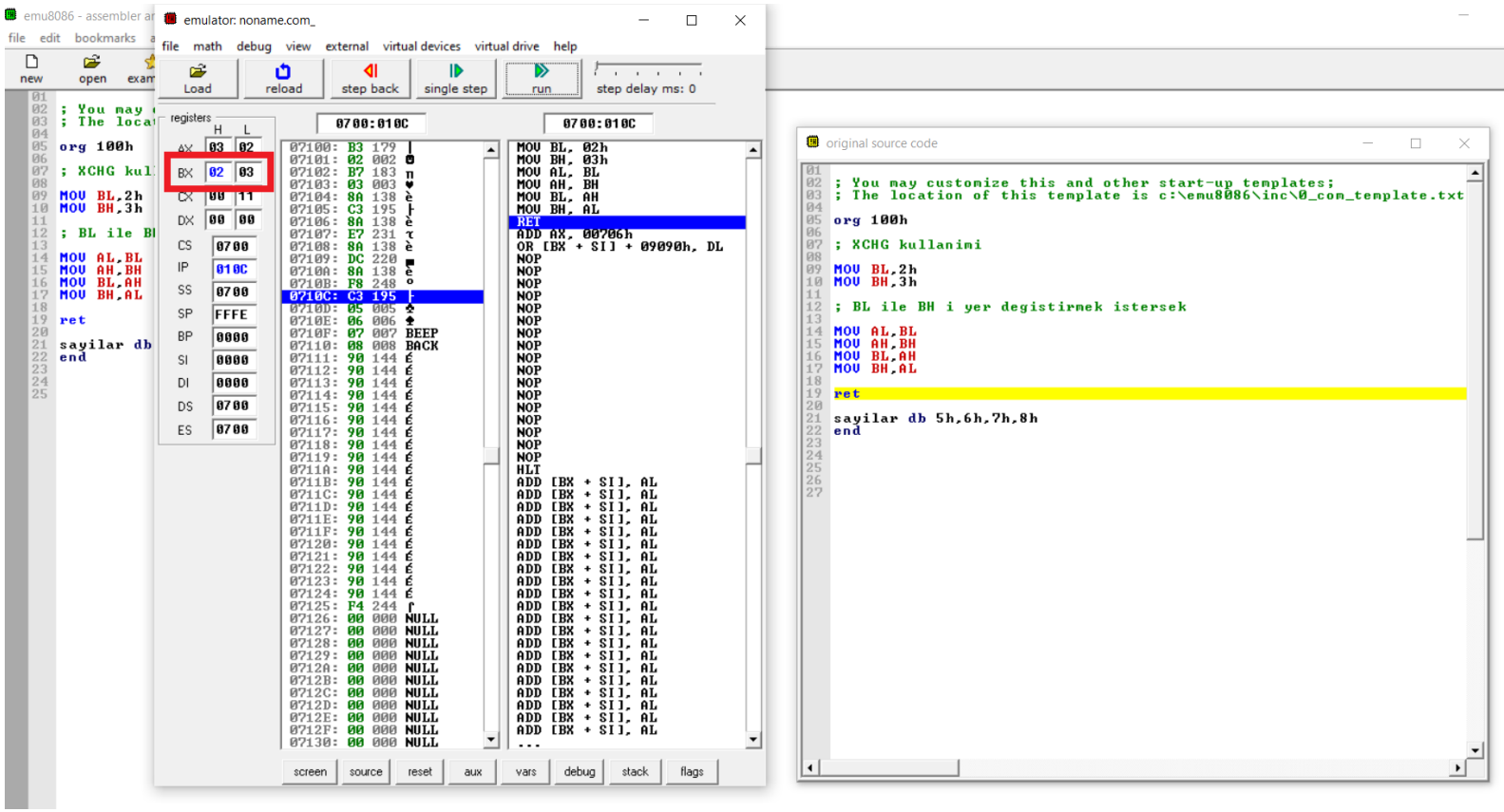
## EMU 8086-MICROPROCESSOR EMULATOR

```
org 100h
; XCHG kullanimi

MOV BL,2h
MOV BH,3h
; BL ile BH i yer degistirmek istersek

MOV AL,BL
MOV AH,BH
MOV BL,AH
MOV BH,AL

ret

sayilar db 5h,6h,7h,8h
end
```

```
org 100h
; XCHG kullanimi

MOV BL,2h
MOV BH,3h
; BL ile BH i yer degistirmek istersek

XCHG BL,BH
ret

sayilar db 5h,6h,7h,8h
end
```

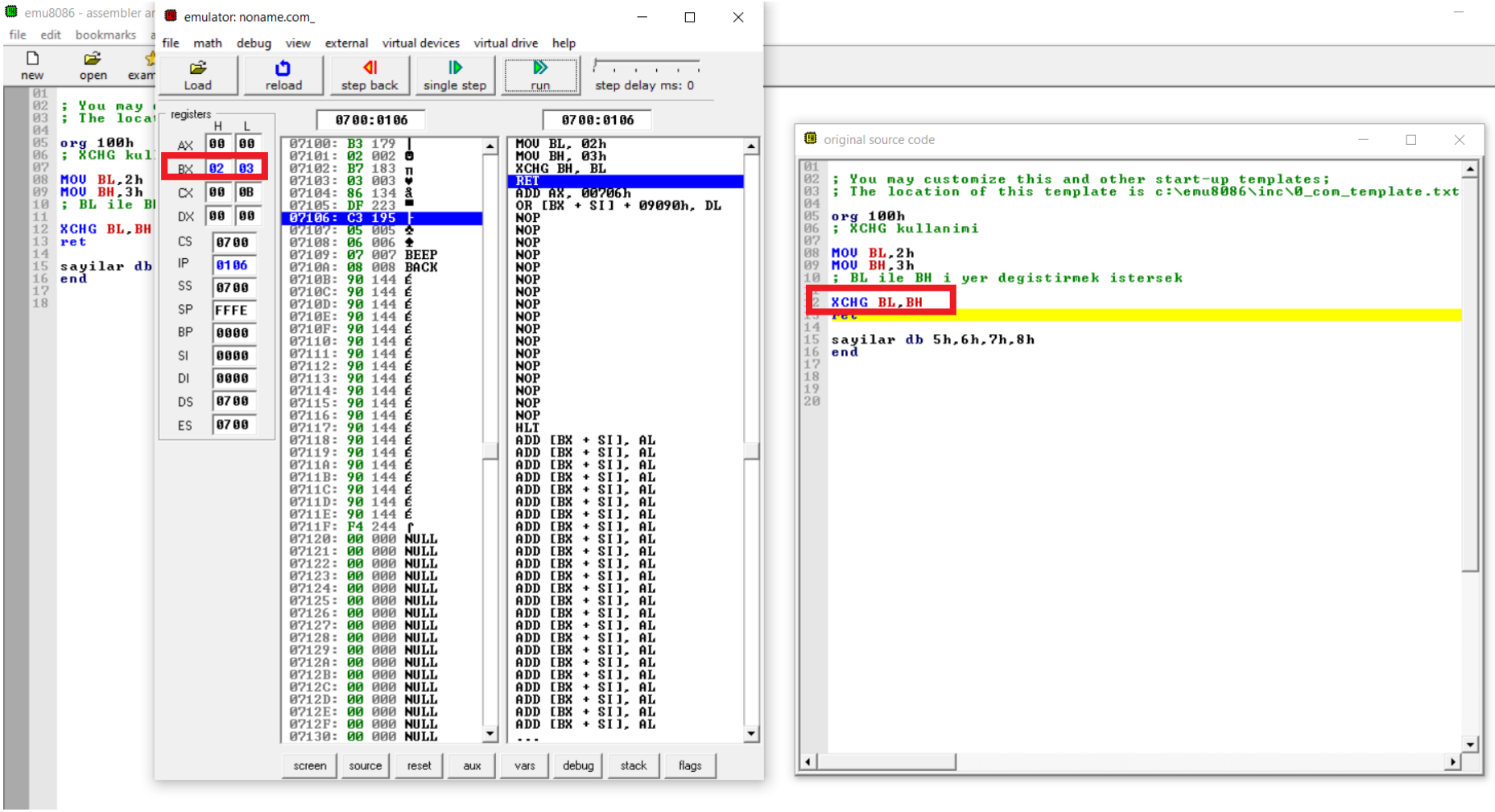
# 8086 16-Bit Mikroişlemci

## EMU 8086-MICROPROCESSOR EMULATOR



# 8086 16-Bit Mikroişlemci

## EMU 8086-MICROPROCESSOR EMULATOR



# 8086 16-Bit Mikroişlemci

## EMU 8086-MICROPROCESSOR EMULATOR

- **INC KOMUTU:**

REG

MEMORY

; degeri 1 arttırma komutudur
MOV AH,5
INC AH ; AH=6 olur

- **DEC KOMUTU:**

REG
MEMORY
; degeri 1 azaltma komutudur
MOV AH,5
DEC AH ; AH=4 olur

# 8086 16-Bit Mikroişlemci

## EMU 8086-MICROPROCESSOR EMULATOR

- **XLATB KOMUTU:**

**operand almaz**

- **Dizinin başlangıç adresini BX registerı içine atamak zorunludur.**
- **Erişmek istediğimiz dizinin elemanının konumu AL registerına yüklemeniz zorunludur**
- **XLATB komutu yazılır sonra istenen eleman AL registerı içine kaydedilmiş olur.**

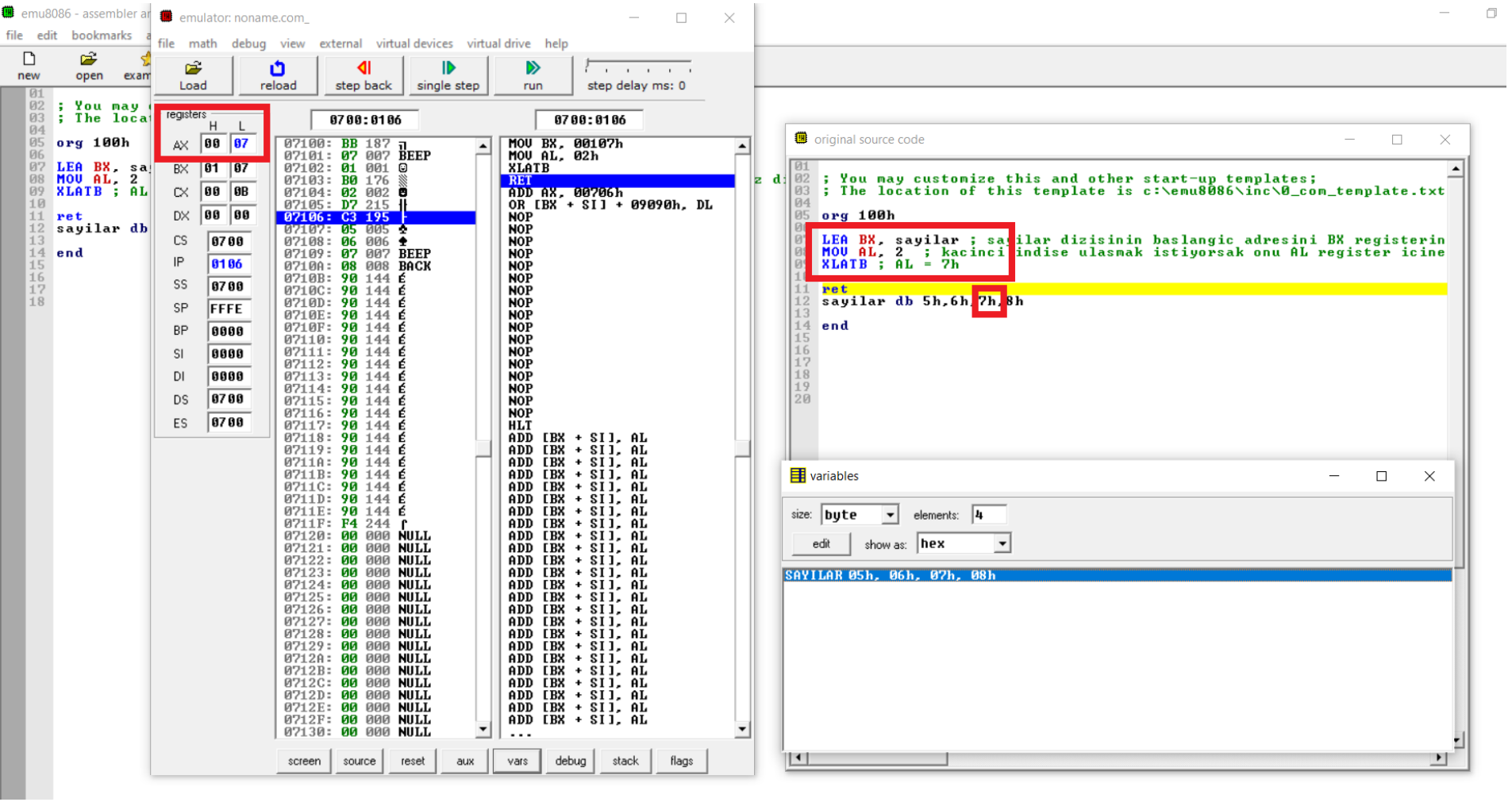
# 8086 16-Bit Mikroişlemci

## EMU 8086-MICROPROCESSOR EMULATOR

```
ORG 100h

LEA BX, sayilar ; sayilar dizisinin baslangic adresini
BX registerina atadik

MOV AL, 2  ; kacinci indise ulasmak istiyorsak onu
AL register icine atiyoruz dizi indisi 0 dan baslar

XLATB ; AL = 7h

RET

sayilar db 5h,6h,7h,8h
```

# 8086 16-Bit Mikroişlemci

## EMU 8086-MICROPROCESSOR EMULATOR



# 8086 16-Bit Mikroişlemci

## EMU 8086-MICROPROCESSOR EMULATOR

- **JMP KOMUTU:**

**Koşulsuz dallanma programda istenilen yere atlanır.**

**JMP operand1**

**Opearand1 burada etiket olur. Bu etiket aslında bellek adresidir ve siz etiketin gösterdiği bellek adresine atlamış olursunuz.**

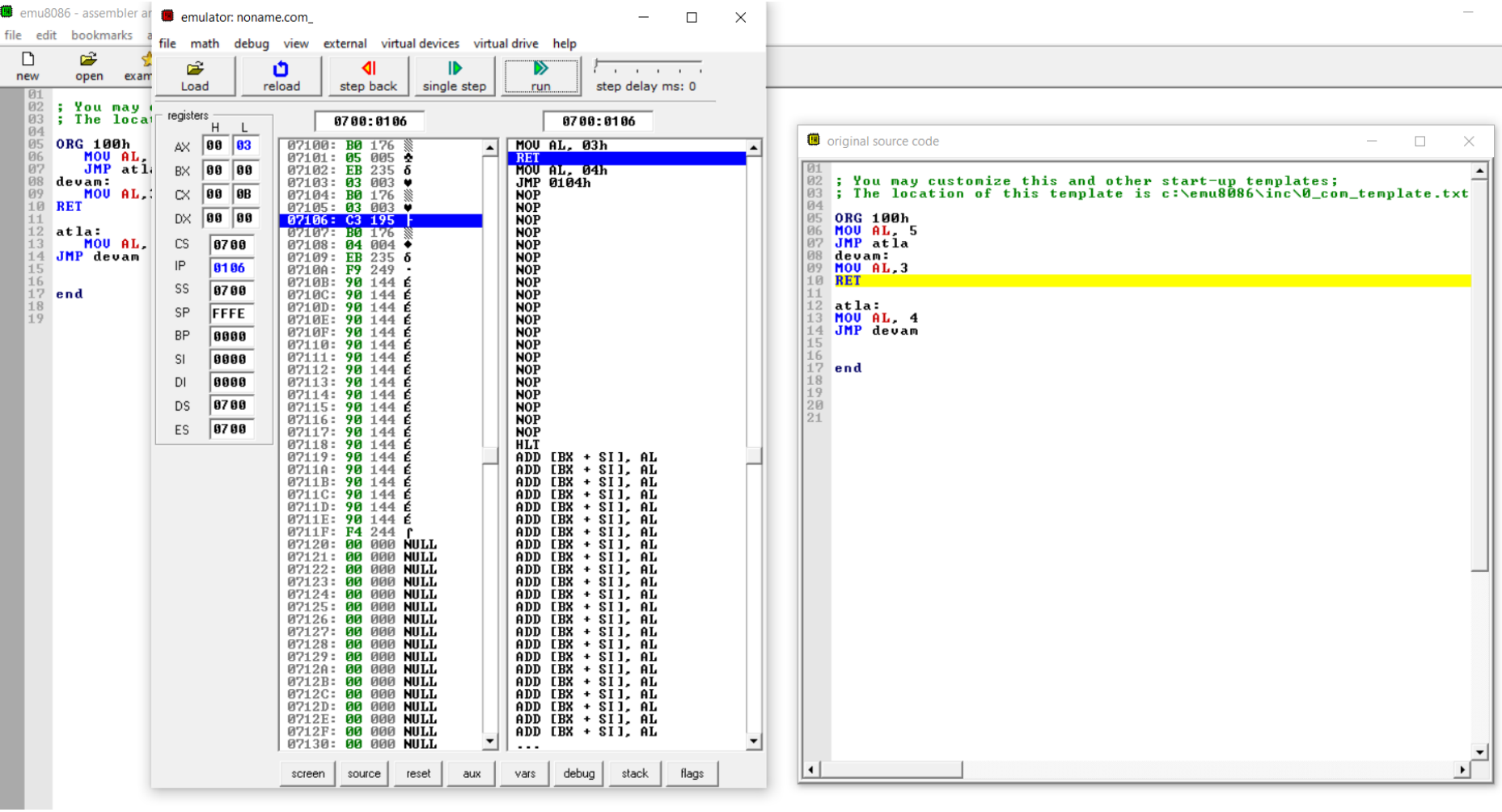
# 8086 16-Bit Mikroişlemci

## EMU 8086-MICROPROCESSOR EMULATOR

```
ORG 100h
MOV AL, 5

JMP atla
devam:
MOV AL,3

RET

atla:
MOV AL, 4
JMP devam

end
```

# 8086 16-Bit Mikroişlemci

## EMU 8086-MICROPROCESSOR EMULATOR



# 8086 16-Bit Mikroişlemci

## EMU 8086-MICROPROCESSOR EMULATOR

- **LOOP KOMUTU:**

**Operand1**

**Etiket:**

- **CX registerına döngünün kaç kez döneceğini atamak zorundasınız.**
- **Komut CX=0 olana kadar devam eder**

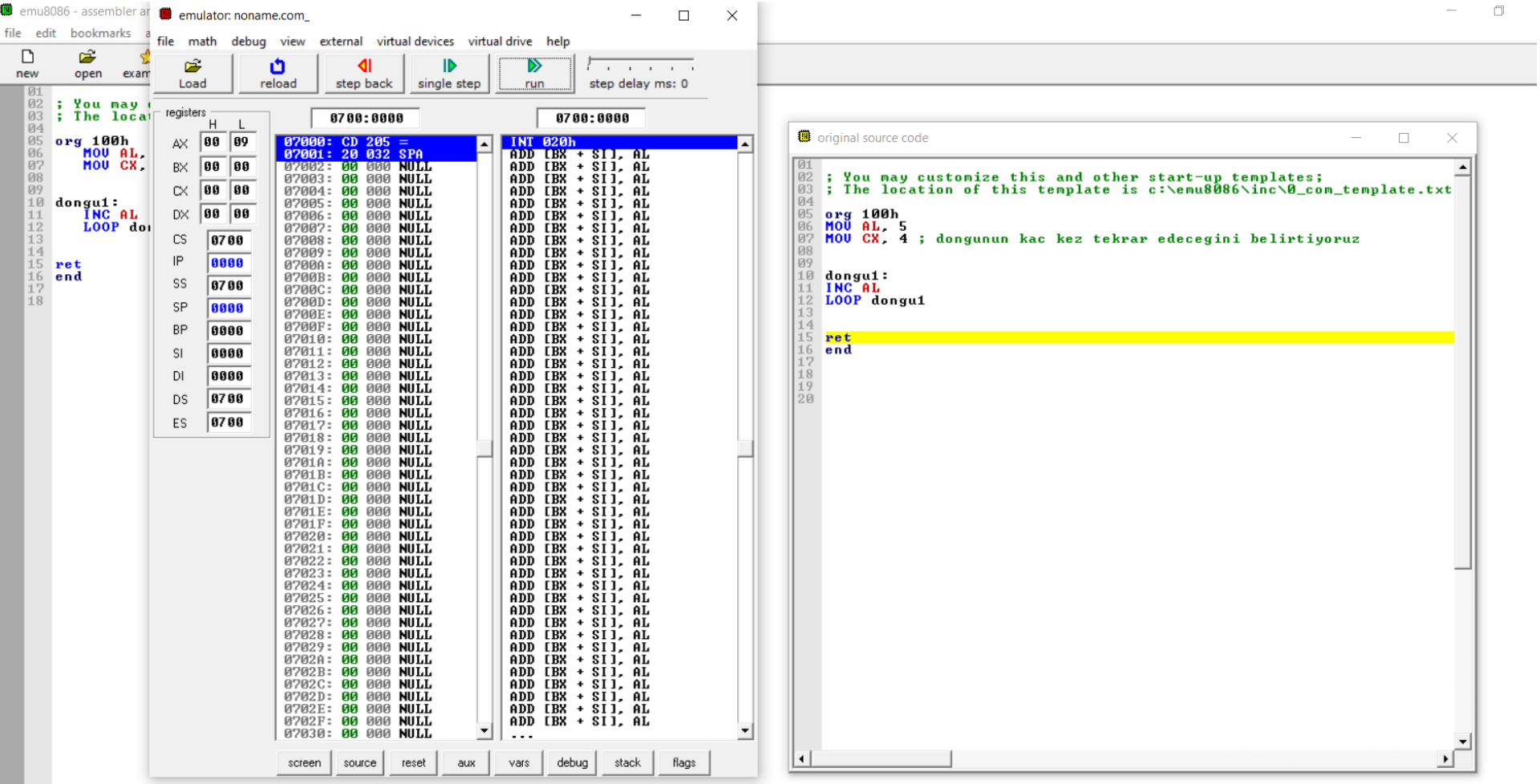
# 8086 16-Bit Mikroişlemci

## EMU 8086-MICROPROCESSOR EMULATOR

```
org 100h
   MOV AL, 5
   MOV CX, 4 ; dongunun kac kez tekrar edecegini belirtiyoruz

dongu1:
   INC AL
   LOOP dongu1
ret
```

# 8086 16-Bit Mikroişlemci

## EMU 8086-MICROPROCESSOR EMULATOR



# 8086 16-Bit Mikroişlemci

## EMU 8086-MICROPROCESSOR EMULATOR

Soru:dizi1 dizisindeki elemanları dizi2 dizisine aktaran 8086 assembly kodunu yazınız?

dizi1=10,12,14,16

**dizi2= 10,12,14,16** olacak

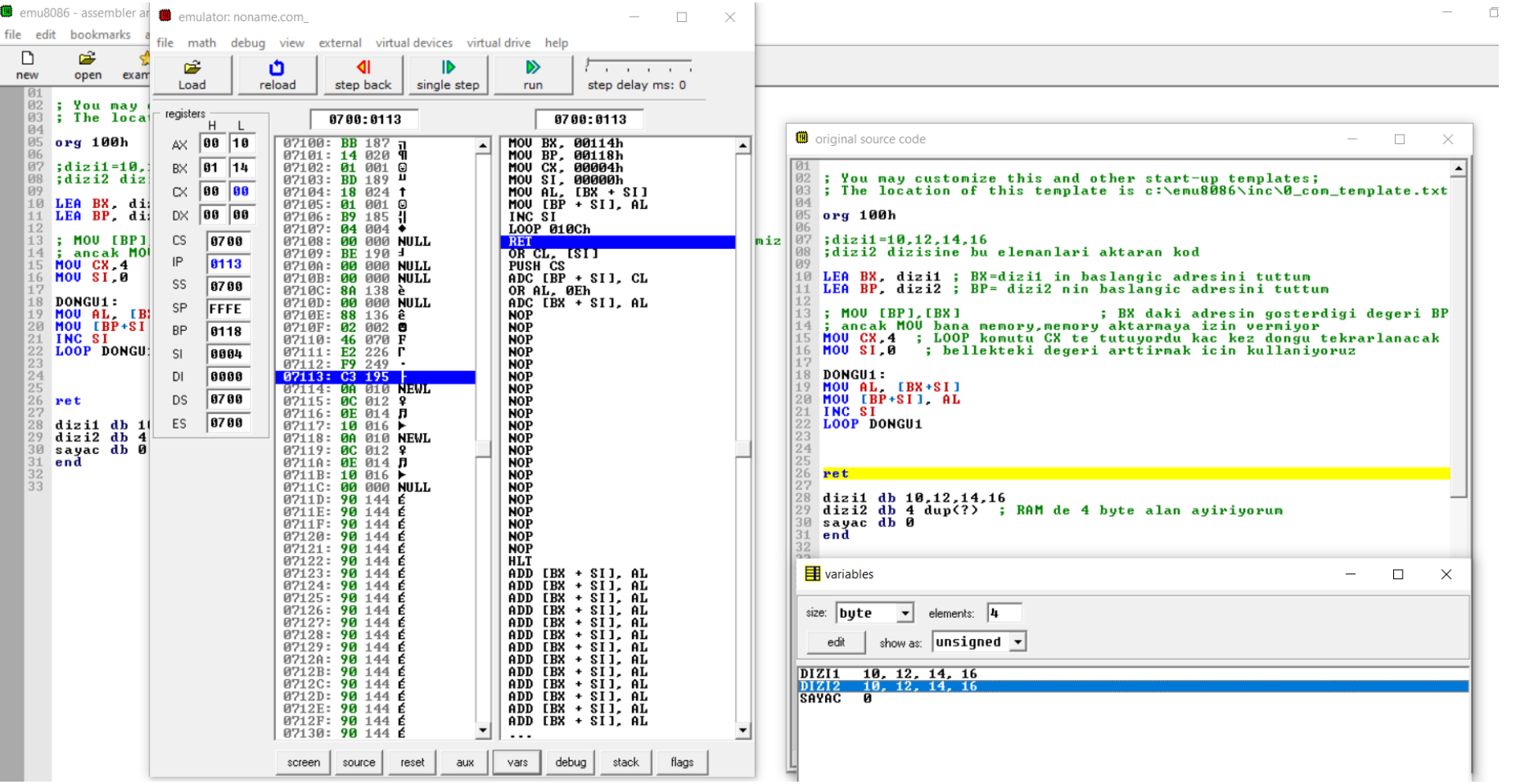
# 8086 16-Bit Mikroişlemci

## EMU 8086-MICROPROCESSOR EMULATOR

```
org 100h

;dizi1=10,12,14,16
;dizi2 dizisine bu elemanlari aktaran kod

LEA BX, dizi1 ; BX=dizi1 in baslangic adresini tuttum
LEA BP, dizi2 ; BP= dizi2 nin baslangic adresini tuttum

; MOV [BP],[BX]            ; BX daki adresin gosterdigi degeri BP ya atmamiz gerekiyor
; ancak MOV bana memory,memory aktarmaya izin vermiyor
MOV CX,4  ; LOOP komutu CX te tutuyordu kac kez dongu tekrarlanacak
MOV SI,0   ; bellekteki degeri arttirmak icin kullaniyoruz

DONGU1:
MOV AL, [BX+SI]
MOV [BP+SI], AL
INC SI
LOOP DONGU1



ret

dizi1 db 10,12,14,16
dizi2 db 4 dup(?)  ; RAM de 4 byte alan ayiriyoruz
end
```

# 8086 16-Bit Mikroişlemci

## EMU 8086-MICROPROCESSOR EMULATOR



# 8086 16-Bit Mikroişlemci

## EMU 8086-MICROPROCESSOR EMULATOR

Soru:dizi1 dizisindeki elemanları dizi2 dizisine tersten aktaran 8086 assembly kodunu yazınız?

dizi1=10,12,14,16

**dizi2= 16,14,12,10** olacak

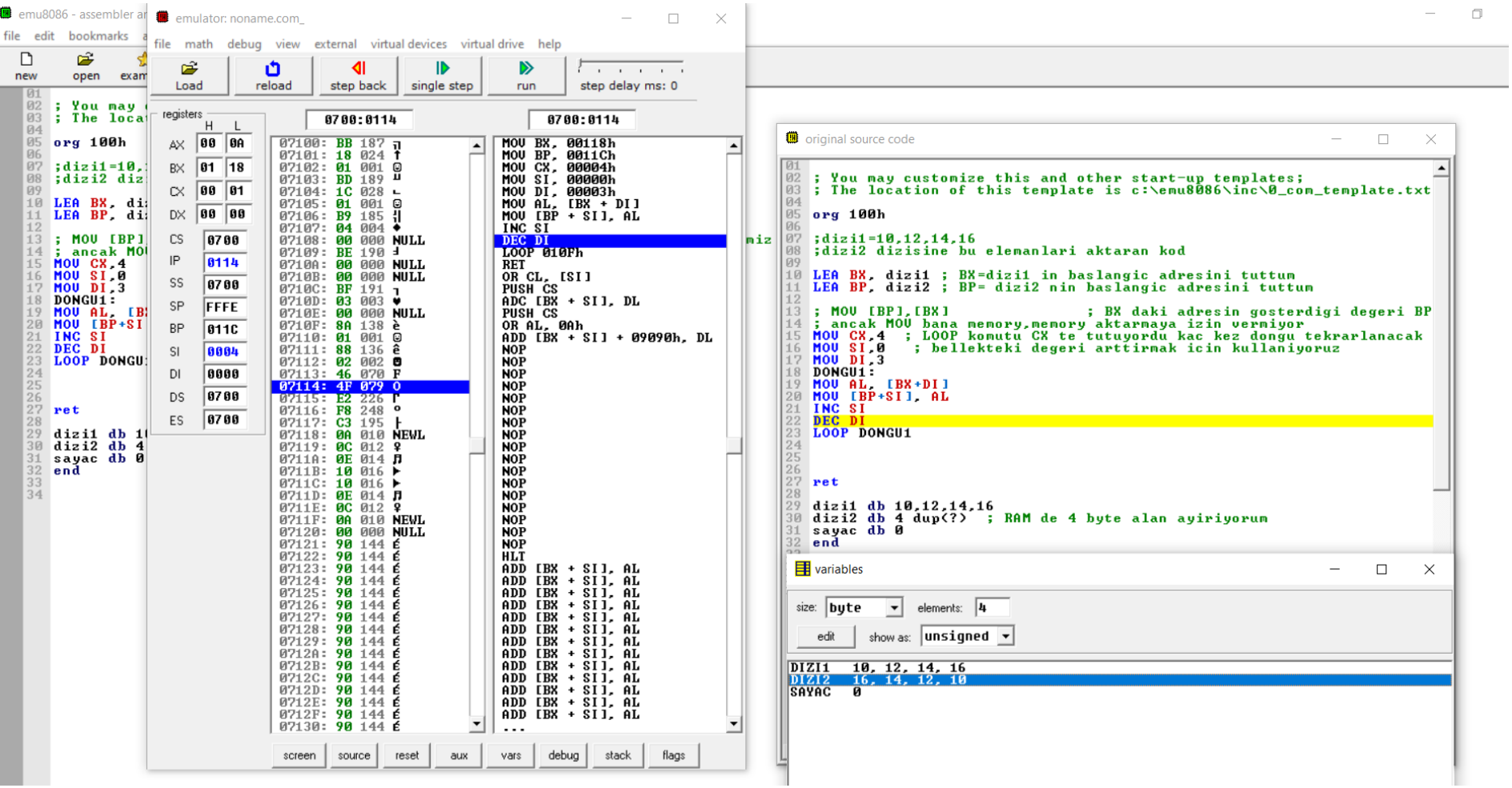
# 8086 16-Bit Mikroişlemci

## EMU 8086-MICROPROCESSOR EMULATOR

```
org 100h

;dizi1=10,12,14,16
;dizi2 dizisine bu elemanlari aktaran kod

LEA BX, dizi1 ; BX=dizi1 in baslangic adresini tuttum
LEA BP, dizi2 ; BP= dizi2 nin baslangic adresini tuttum

; MOV [BP],[BX]             ; BX daki adresin gosterdigi degeri BP ya atmamiz gerekiyor
; ancak MOV bana memory,memory aktarmaya izin vermiyor
MOV CX,4  ; LOOP komutu CX te tutuyordu kac kez dongu tekrarlanacak
MOV SI,0   ; bellekteki degeri arttirmak icin kullaniyoruz
MOV DI,3
DONGU1:
MOV AL, [BX+DI]
MOV [BP+SI], AL
INC SI
DEC DI
LOOP DONGU1



ret

dizi1 db 10,12,14,16
dizi2 db 4 dup(?)  ; RAM de 4 byte alan ayiriyorum
sayac db 0
end
```

# 8086 16-Bit Mikroişlemci

## EMU 8086-MICROPROCESSOR EMULATOR



# 8086 16-Bit Mikroişlemci

## EMU 8086-MICROPROCESSOR EMULATOR

Soru:metin karakter dizisindeki elemanları tersmetin dizisine tersten aktaran 8086 assembly kodunu yazınız?

metin ="bucumleterstenyazilacak"

tersmetin =" kacalizaynetsretelmucub"

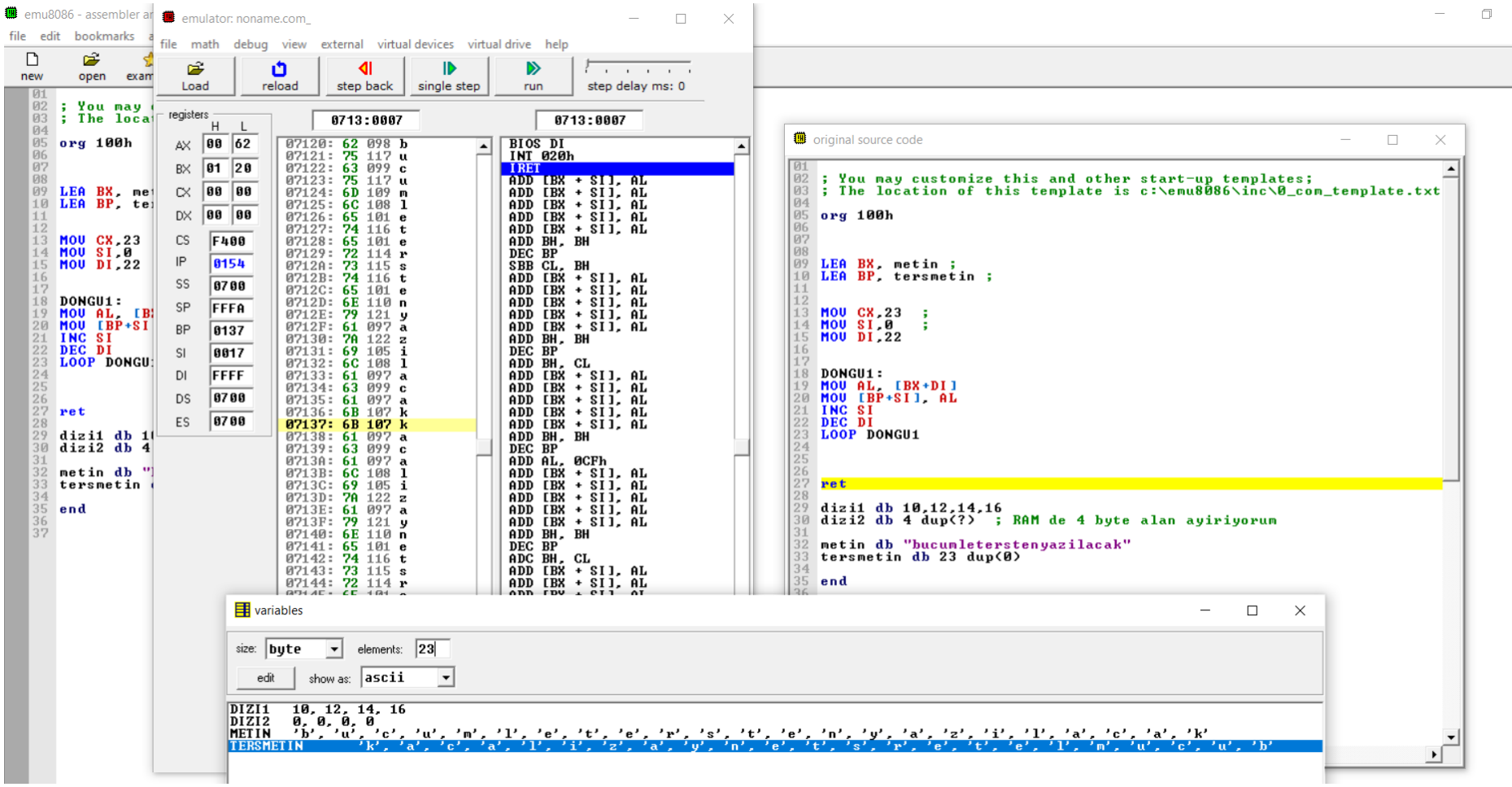
# 8086 16-Bit Mikroişlemci

## EMU 8086-MICROPROCESSOR EMULATOR

```
org 100h



LEA BX, metin ;
LEA BP, tersmetin ;


MOV CX,23  ;
MOV SI,0   ;
MOV DI,22


DONGU1:
MOV AL, [BX+DI]
MOV [BP+SI], AL
INC SI
DEC DI
LOOP DONGU1



ret

dizi1 db 10,12,14,16
dizi2 db 4 dup(?)  ; RAM de 4 byte alan ayiriyorum

metin db "bucumleterstenyazilacak"
tersmetin db 23 dup(0)

end
```

# 8086 16-Bit Mikroişlemci

## EMU 8086-MICROPROCESSOR EMULATOR



# 8086 16-Bit Mikroişlemci

## EMU 8086-MICROPROCESSOR EMULATOR

Soru: dizi1 dizisindeki elemanları indisler dizisindeki indis değerleri dikkate alınarak hedef dizisine aktaran 8086 assembly kodunu yazınız?

dizi1 =10,11,10,9

indisler =2,0,3,1

**hedef= 10,10,9,11**

# 8086 16-Bit Mikroişlemci

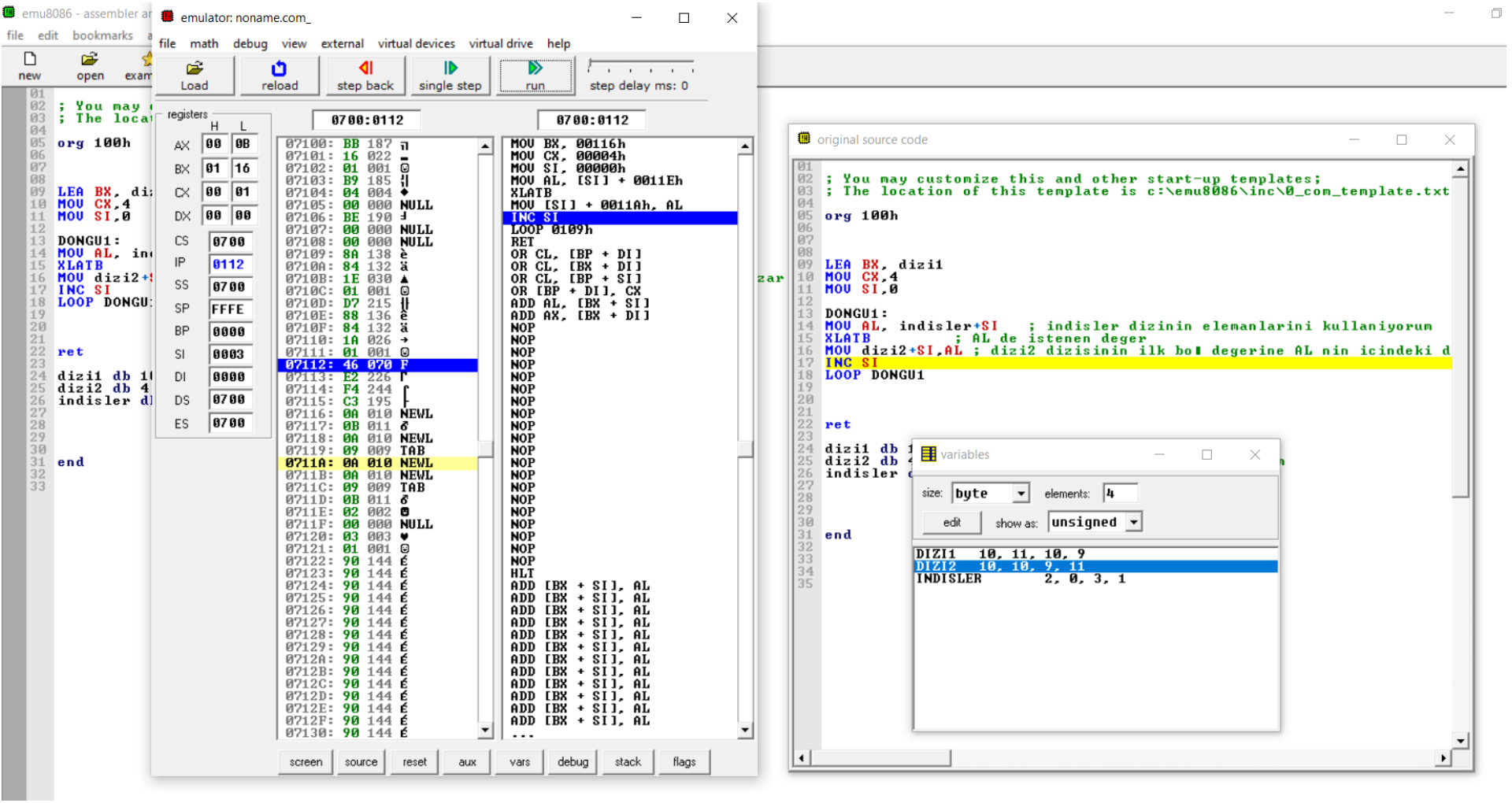
## EMU 8086-MICROPROCESSOR EMULATOR

```
org 100h


LEA BX, dizi1
MOV CX,4
MOV SI,0

DONGU1:
MOV AL, indisler+SI   ; indisler dizinin elemanlarini kullaniyorum
XLATB        ; AL de istenen deger
MOV dizi2+SI,AL ; dizi2 ye ilgili indise AL nin icindeki degeri yazar
INC SI
LOOP DONGU1


ret

dizi1 db 10,11,10,9
dizi2 db 4 dup(?)  ; RAM de 4 byte alan ayiriyorum
indisler db 2,0,3,1


end
```

```
org 100h


LEA BX, dizi1
LEA BP,indisler
MOV CX,4
MOV SI,0

DONGU1:
MOV AL, [BP+SI]   ; indisler dizinin elemanlarini kullaniyorum
XLATB        ; AL de istenen deger
MOV [dizi2+SI],AL ; dizi2 dizisinin ilk boş degerine AL nin icindeki degeri yazar
INC SI
LOOP DONGU1


ret

dizi1 db 10,11,10,9
dizi2 db 4 dup(?)  ; RAM de 4 byte alan ayiriyorum
indisler db 2,0,3,1
```

# 8086 16-Bit Mikroişlemci

## EMU 8086-MICROPROCESSOR EMULATOR



# 8086 16-Bit Mikroişlemci

## EMU 8086-MICROPROCESSOR EMULATOR

Soru: dizi1 dizisindeki elemanların değerlerini 1 arttırarak dizi2 dizisine aktaran 8086 assembly kodunu yazınız?

# 8086 16-Bit Mikroişlemci

## EMU 8086-MICROPROCESSOR EMULATOR

```
org 100h



LEA BX, dizi1
LEA BP,dizi2
MOV CX,4
MOV SI,0

DONGU1:
MOV AL, [BX+SI]
INC AL
MOV [BP+SI],AL ;
INC SI
LOOP DONGU1



ret

dizi1 db 10,11,10,9
dizi2 db 4 dup(?)  ; RAM de 4 byte alan ayiriyorum




end
```

# 8086 16-Bit Mikroişlemci

## EMU 8086-MICROPROCESSOR EMULATOR