



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

The screenshot displays the EMU 8086 Microprocessor Emulator interface, which is divided into several panels:

- Registers Panel:** Shows the current state of the 8086 registers. The **BP** register is highlighted with a red box, showing its value as **0109**. Other registers like **AX**, **BX**, **DX**, **SI**, **DI**, **DS**, and **ES** are also visible.
- Source Code Panel:** Displays the assembly code being executed. The code includes comments in Turkish and instructions like **LEA BX, sayi**, **MOV AX, [BX]**, **MOV BP, offset sayi**, and **ret**. The **ret** instruction is highlighted in yellow.
- Variables Panel:** Shows the current values of variables. The variable **SAYI** is listed with a value of **0BCDEh**.
- Disassembly Panel:** Shows the disassembled instructions, including **MOV BX, 00109h**, **MOV AX, [BX]**, **MOV BP, 00109h**, **RET**, and **PID10R w.[SI] + 09090h**.

The interface also includes a menu bar (file, edit, bookmarks, file, math, debug, view, external, virtual devices, virtual drive, help) and a toolbar with buttons for Load, reload, step back, single step, run, and step delay ms: 0.

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

The screenshot displays the EMU 8086 Microprocessor Emulator interface, which is divided into several windows:

- Assembly Code Window:** Shows the assembly code being executed. The code includes comments in Turkish and assembly instructions. The current instruction being executed is `MOV BX, 00106h` at address `07106h`.
- Registers Window:** Displays the state of the 8086 registers. The `AX` register contains `00 05`, `BX` contains `01 06`, `CX` contains `00 0A`, and `DX` contains `00 00`. The `IP` (Instruction Pointer) register is highlighted, showing `0105`.
- Source Code Window:** Shows the original source code in assembly language. The code includes comments and instructions, with the current instruction being `MOV AL, [BX]` at address `07106h`.
- Variables Window:** Displays the state of variables. The variable `SAYILAR` is shown with values `05h, 06h, 07h, 08h`.

The interface also includes a menu bar (file, math, debug, view, external, virtual devices, virtual drive, help) and a toolbar with buttons for Load, reload, step back, single step, and run. A status bar at the bottom shows the current step delay in milliseconds (0 ms).

8086 16-Bit Mikroişlemci

EMU 8086-MICROPROCESSOR EMULATOR

;dizide bir sonraki adrese ulaşma:

MOV AL, [SI+1] ; SI nin gösterdiği yeri 1 arttır
ordaki değeri AL ye atadık (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 kullanımı
```

```
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 kullanımı
```

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

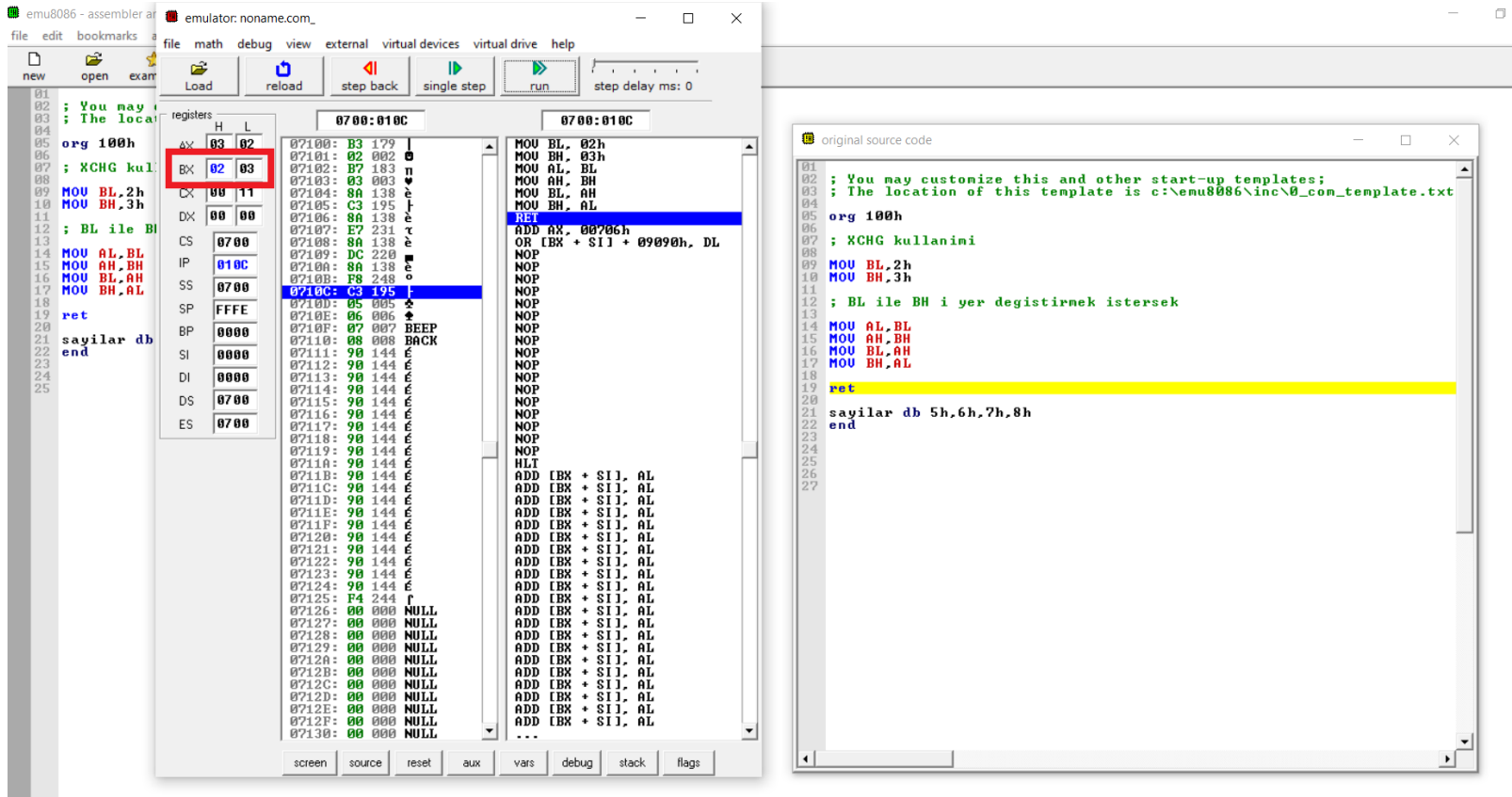


```
XCHG BL,BH  
ret
```

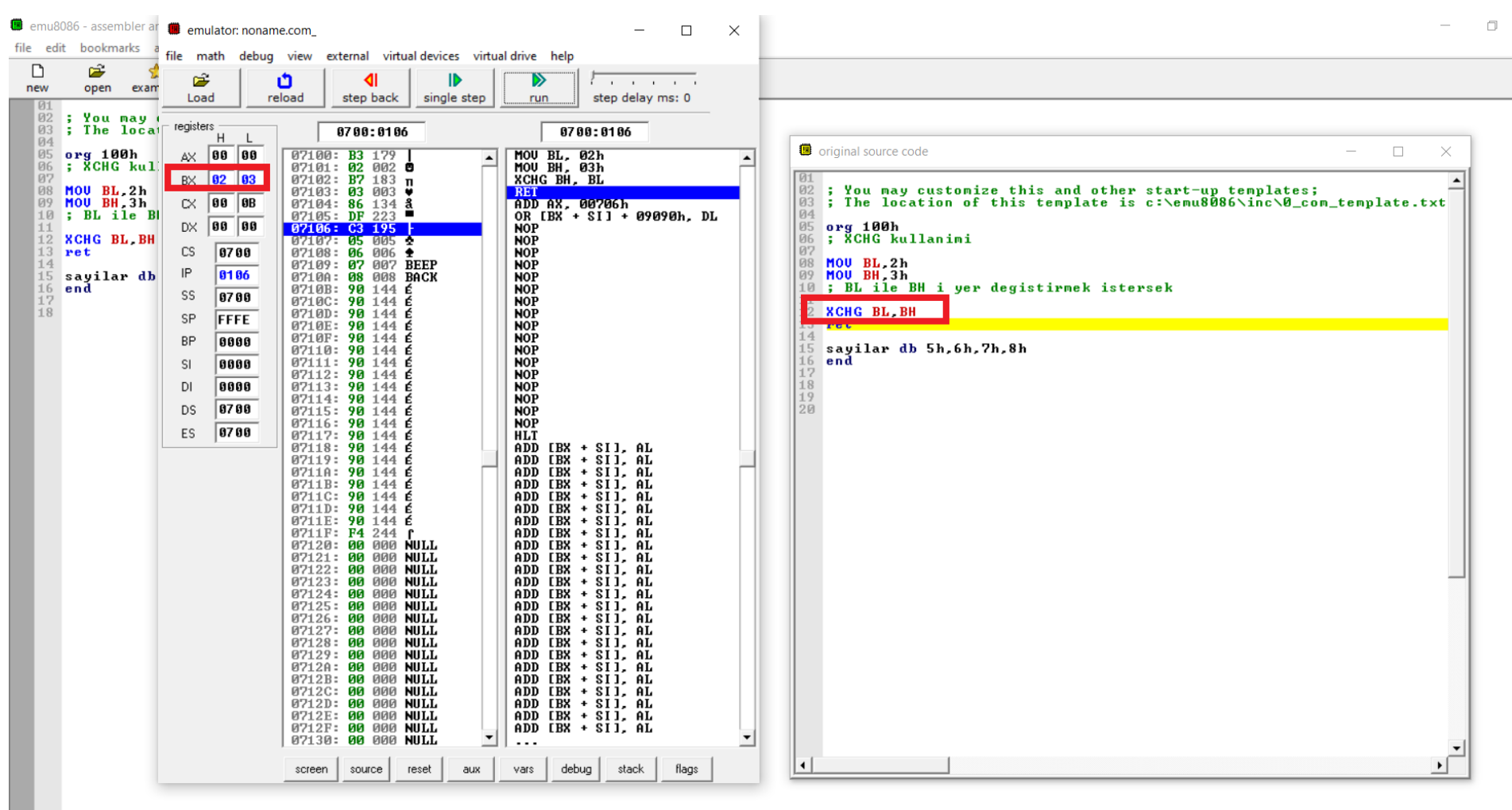
```
sayilar db 5h,6h,7h,8h  
end
```

8086 16-Bit Mikroişlemci

EMU 8086-MICROPROCESSOR EMULATOR



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

The screenshot displays the EMU 8086 Microprocessor Emulator interface, which is divided into several windows:

- Registers Window:** Shows the state of the 8086 registers. The AX register is highlighted with a red box, showing a value of 0007. The IP (Instruction Pointer) register is also highlighted, showing a value of 0106. The CS (Code Segment) register is 0700, and the DS (Data Segment) register is 0700.
- Assembly Code Window:** Displays the assembly code being executed. The code is as follows:

```
01 ; You may customize this and other start-up templates;  
02 ; The location of this template is c:\emu8086\inc\0_com_template.txt  
03  
04  
05 org 100h  
06  
07 LEA BX, sayilar ; sayilar dizisinin baslangic adresini BX registerin  
08 MOV AL, 2 ; kacinci indise ulasmak istiyorsak onu AL register icine  
09 XLATB ; AL = 7h  
10  
11 ret  
12 sayilar db 5h,6h,7h,8h  
13  
14 end  
15  
16  
17  
18
```

The line `XLATB ; AL = 7h` is highlighted with a yellow box, and the value `7h` is also highlighted with a red box.
- Variables Window:** Shows the memory location of the variable `SAYILAR` at address 05h, 06h, 07h, and 08h. The variable is defined as `SAYILAR db 05h, 06h, 07h, 08h`.
- Stack Window:** Shows the stack memory, which is currently empty.
- Flags Window:** Shows the status of the 8086 flags, which are currently all zero.

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
```

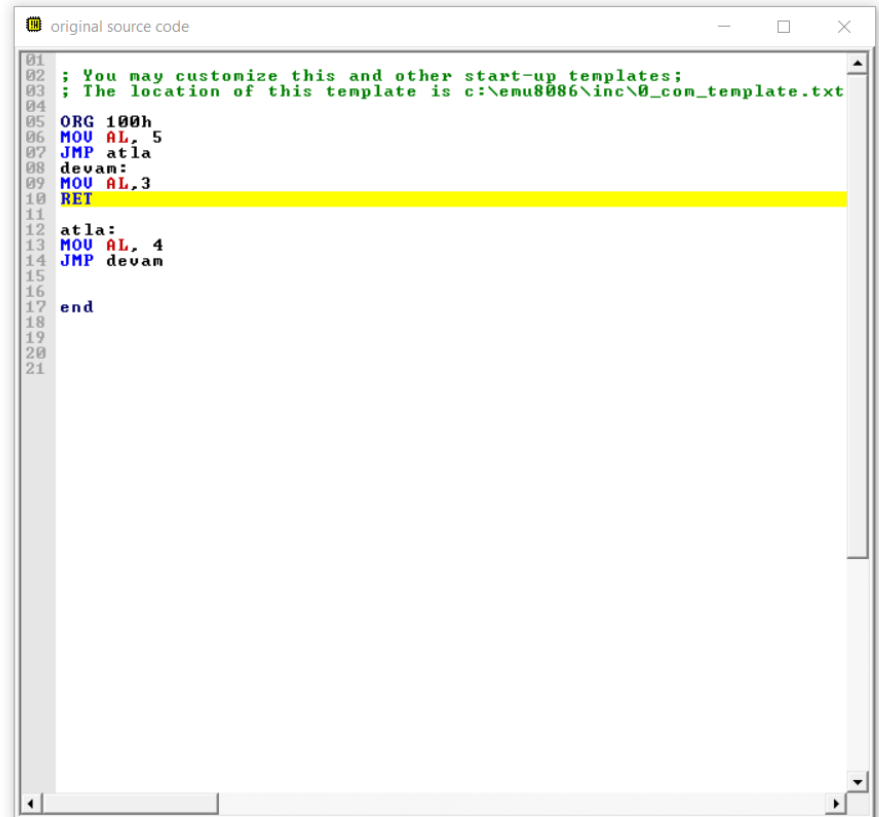
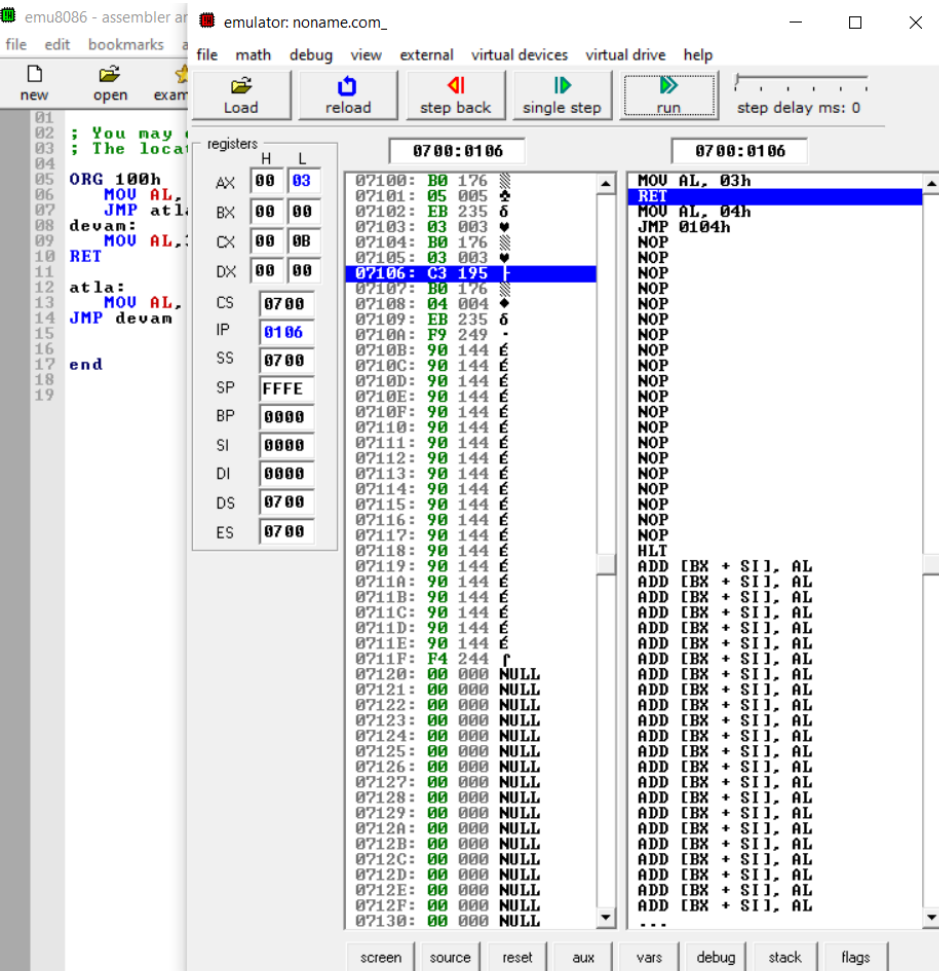
```
RET
```

```
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 edeceğini
belirtiyoruz

dongu1:

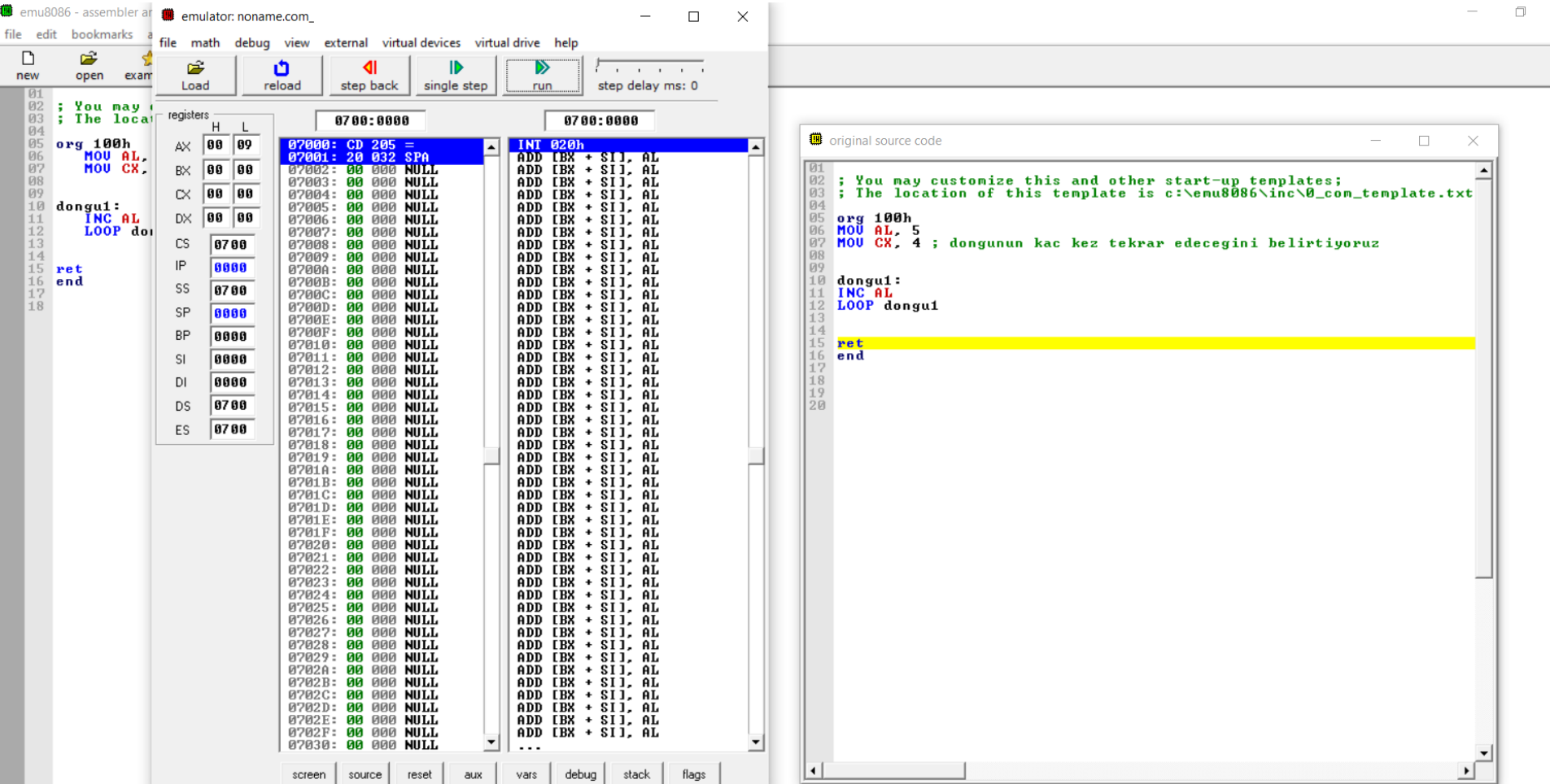
INC AL

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 arttirmek 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

The screenshot displays the EMU 8086 Microprocessor Emulator interface, which is divided into several windows:

- Registers Window:** Shows the state of the 8086 registers. The registers are organized into two columns: H (High) and L (Low). The registers are: AX, BX, CX, DX, SI, DI, BP, SP, IP, CS, DS, ES. The values are: AX=0010, BX=0114, CX=0000, DX=0000, SI=0004, DI=0000, BP=0118, SP=FFFE, IP=0113, CS=0700, DS=0700, ES=0700.
- Source Code Window:** Displays the assembly code being executed. The code is as follows:

```
01 ; You may customize this and other start-up templates;  
02 ; The location of this template is c:\emu8086\inc\0_com_template.txt  
03  
04 org 100h  
05  
06 ;dizi1=10,12,14,16  
07 ;dizi2 dizisine bu elemanlari aktaran kod  
08  
09 LEA BX, dizi1 ; BX=dizi1 in baslangic adresini tuttum  
10 LEA BP, dizi2 ; BP= dizi2 nin baslangic adresini tuttum  
11  
12 ; MOV [BP], [BX] ; BX daki adresin gosterdigi degeri BP  
13 ; ancak MOV bana memory, memory aktarmaya izin vermiyor  
14 ; MOV [BP+SI], [BX] ; BP+SI daki adresin gosterdigi degeri BP  
15 ; ancak MOV bana memory, memory aktarmaya izin vermiyor  
16 MOV CX, 4 ; LOOP komutu CX te tutuyordu kac kez dongu tekrarlanacak  
17 MOV SI, 0 ; bellekteki degeri arttirmak icin kullaniyoruz  
18  
19 DONGU1:  
20 MOV AL, [BX+SI]  
21 MOV [BP+SI], AL  
22 INC SI  
23 LOOP DONGU1  
24  
25 ret  
26  
27 dizi1 db 10,12,14,16  
28 dizi2 db 4 dup(?) ; RAM de 4 byte alan ayiriyoruz  
29  
30 sayac db 0  
31 end  
32  
33
```
- Variables Window:** Shows the variables defined in the code. The variables are: DIZI1 (10, 12, 14, 16), DIZI2 (10, 12, 14, 16), and SAYAC (0).

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 arttirmek 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

The screenshot displays the EMU 8086 Microprocessor Emulator interface. The main window is titled "emu8086 - assembler and emulator: noname.com". It features a menu bar (file, edit, bookmarks, math, debug, view, external, virtual devices, virtual drive, help) and a toolbar with buttons for Load, reload, step back, single step, run, and step delay ms: 0.

The interface is divided into several panes:

- Registers:** A table showing the state of 16-bit registers. The AX register is highlighted with a value of 000A. Other registers like BX, CX, DX, SI, DI, BP, SP, and DS are also visible.
- Assembly Code:** A list of assembly instructions being executed. The instruction "DEC DI" is highlighted in blue. The code includes comments in Turkish, such as "You may customize this and other start-up templates;" and "The location of this template is c:\emu8086\inc\0_com_template.txt".
- Original Source Code:** A window showing the original source code in assembly language. It includes comments and instructions like "org 100h", "LEA BX, dizil1", "LEA BP, dizil2", "MOV AL, [BX+DI]", "MOV [BP+SI], AL", "INC SI", "DEC DI", and "LOOP DONGU1".
- Variables:** A window showing the values of variables. The variable "DIZI1" is set to 10, 12, 14, 16, and "DIZI2" is set to 16, 14, 12, 10. The variable "SAYAC" is set to 0.

The assembly code in the main pane includes comments in Turkish, such as "You may customize this and other start-up templates;" and "The location of this template is c:\emu8086\inc\0_com_template.txt". The code also includes instructions like "org 100h", "LEA BX, dizil1", "LEA BP, dizil2", "MOV AL, [BX+DI]", "MOV [BP+SI], AL", "INC SI", "DEC DI", and "LOOP DONGU1".

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

The screenshot displays the EMU 8086 Microprocessor Emulator interface, which is divided into several windows:

- Assembly Code Window:** Shows the assembly code being executed. The code includes comments in Turkish, such as "You may customize this and other start-up templates;" and "The location of this template is c:\emu8086\inc\0_com_template.txt". The code starts with `org 100h` and includes instructions like `LEA BX, metin`, `LEA BP, tersmetin`, `MOV CX, 23`, `MOV SI, 0`, `MOV DI, 22`, `DONGU1:`, `MOV AL, [BX+SI]`, `MOV [BP+SI], AL`, `INC SI`, `DEC DI`, `LOOP DONGU1`, `ret`, `dizi1 db 10,12,14,16`, `dizi2 db 4 dup{?}`, `metin db "bucunlesterstenyazilacak"`, `tersmetin db 23 dup{0}`, and `end`.
- Registers Window:** Displays the state of the 8086 registers. The `AX` register is highlighted, showing a value of `00 62`. Other registers like `BX`, `CX`, `DX`, `CS`, `IP`, `SS`, `SP`, `BP`, `SI`, `DI`, `DS`, and `ES` are also visible.
- Memory Window:** Shows the memory address `0713:0007` and the corresponding memory contents. The memory is organized into a table with columns for address, hex value, and ASCII value. The memory contains the string `"bucunlesterstenyazilacak"` followed by 23 zeros.
- Variables Window:** Displays the variables defined in the code. The variables are `DIZI1` (10, 12, 14, 16), `DIZI2` (0, 0, 0, 0), `METIN` (a string of 23 characters), and `TERSMETIN` (a string of 23 zeros).

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
```

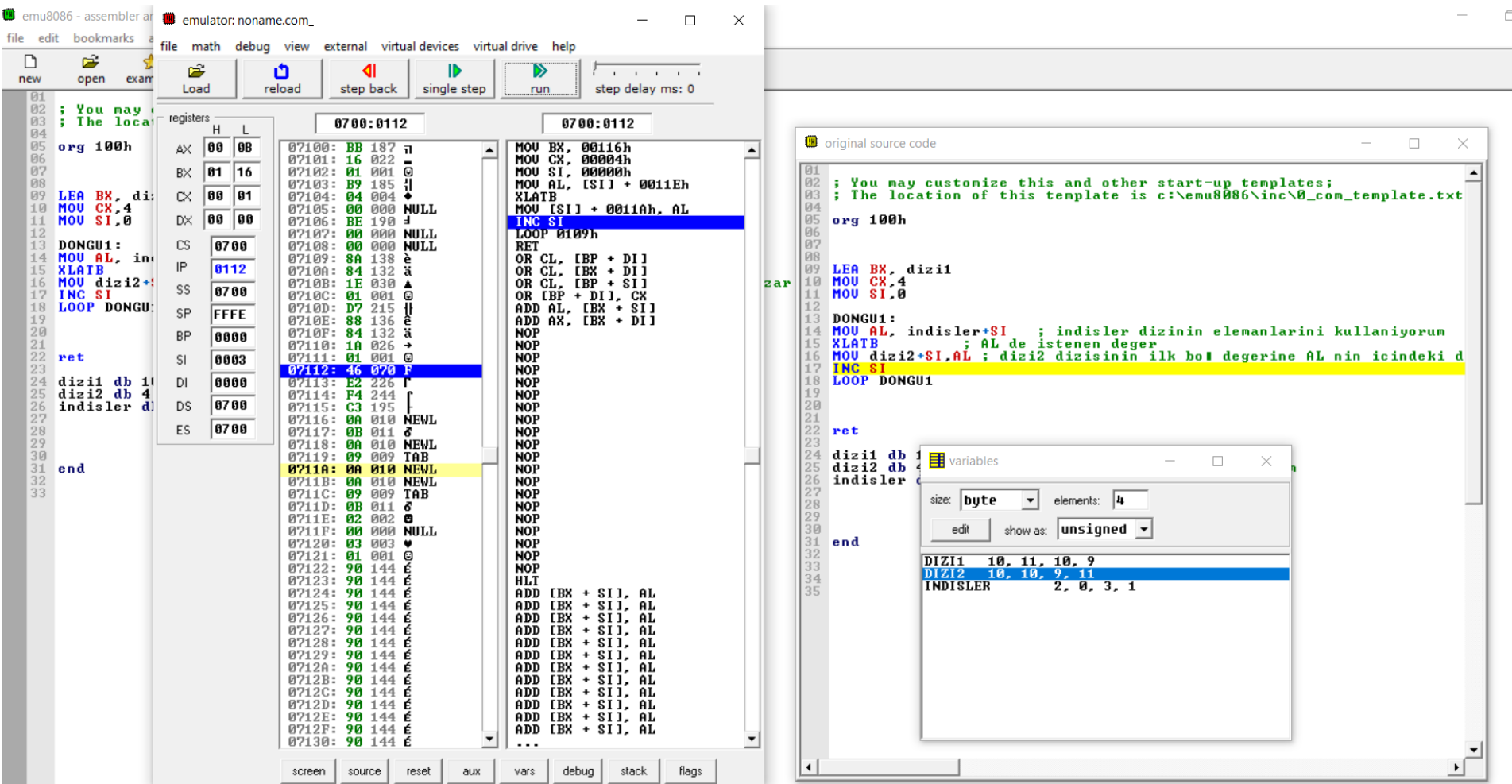
DONGU1:

```
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
```

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

The screenshot displays the EMU 8086 Microprocessor Emulator interface, which is divided into several windows:

- Assembly Code Window:** Shows the assembly code being executed. The code includes comments like "; You may customize this and other start-up templates;" and "; The location of this template is c:\emu8086\inc\0_com_template.txt". The code starts with `org 100h` and includes a loop labeled `DONGU1` that increments `SI` and `DI` registers. The code ends with `ret`.
- Registers Window:** Displays the current values of the 8086 registers. The `AX` register is `00 0A`, `BX` is `01 16`, `CX` is `00 00`, and `DX` is `00 00`. The `CS` register is `F400`, `IP` is `0154`, `SS` is `0700`, `SP` is `FFFA`, `BP` is `011A`, `SI` is `0004`, `DI` is `0000`, `DS` is `0700`, and `ES` is `0700`.
- Memory Window:** Shows the memory contents at address `F400:0154`. The memory contains the BIOS `INT 020h` routine, which includes instructions like `IRET`, `ADD`, `DEC`, `SBB`, `INC`, `MOV`, `LOOP`, and `ret`.
- Variables Window:** Displays the current values of the variables `DIZI1` and `DIZI2`. `DIZI1` is `10, 11, 10, 9` and `DIZI2` is `0Bh, 0Ch, 0Bh, 0Ah`.