Bölüm 4: Bellek Erişimi

Mikroişlemciler

Bellek Erişimi

- Belleğe erişmek için BX, SI, DI, BP yazmaçları kullanılır.
- Bu yazmaçlar [ ] sembolleri içerisinde birleştirerek,
  - farklı bellek konumlarına erişilebilir.
  - Bu birleştirmeler, adresleme modları olarak adlandırılır.

[BX + SI] [BX + DI] [BP + SI] [BP + DI]	[SI] [DI] d16 [BX]	[BX + SI + d8] [BX + DI + d8] [BP + SI + d8] [BP + DI + d8]
[SI + d8] [DI + d8] [BP + d8] [BX + d8]	[BX + SI + d16] [BX + DI + d16] [BP + SI + d16] [BP + DI + d16]	[SI + d16] [DI + d16] [BP + d16] [BX + d16]

İleri Kaydırma (Displacement)

- **d8**: 8 bit işaretli dolaysız ileri kaydırma (*immediate displacement*)
  - Örneğin: 22, 55h, -1 ..
- **d16**: 16 bit işaretli dolaysız ileri kaydırma (*immediate displacement*)
  - Örneğin: 300, 5517h, -259 ..
- Herhangi bir değer veya değişkenin bağıl konumu (*offset*) olabilir.
- Birden fazla değer varsa, derleyici tek bir değer hesaplar.
- İleri kaydırma [ ] sembolleri içinde veya dışında olabilir,
  - *assembler* her iki durum için de aynı makine kodunu üretir.
- İşaretli bir değerdir, bu nedenle hem pozitif hem negatif olabilir.

Adresleme Modları

- Bellek erişimi için yazmaçların kombinasyonlarına dayanır.
- Geniş bir esneklik sağlar.
- SS kesim yazmacı, BP yazmacı ile beraber kullanılır.
- DS kesim yazmacı, BP içermeyen modlarda kullanılır.
- BX ve BP, aynı modda bir araya gelmez.
- SI ve DI, aynı modda bir araya gelmez.
- Tablo kullanılarak tüm geçerli kombinasyonlar oluşturulabilir.
- Örnekler: [BX+5], [BX+SI], [DI+BX-4]

Kesim Ve Bağıl Konum

- Kesim yazmacındaki (CS, DS, SS, ES) değere kesim (*segment*),
- Genel yazmaçlardaki (BX, SI, DI, BP) değere bağıl konum (*offset*) denir.
- Kesim, belleğin bloklarını temsil ederken,
  - Bağıl konum, o bloktaki belirli bir konumu temsil eder.
- Kesim ve bağıl konum, birleştirilerek fiziksel bellek adresi oluşturulur.
- DS değeri 1234h ve SI değeri 7890h olduğunda;
  - 1234:7890 olarak gösterilebilir.
  - Fiziksel adres: 1234h * 10h + 7890h = 19BD0h.

Veri Türü Belirtme

- Derleyiciye veri türü hakkında bilgi vermek için ön ekler kullanılır.
  - byte ptr [BX]: Bir byte erişimi için.
  - word ptr [BX]: İki byte erişimi için.
- Assembler, daha kısa ön ekleri de destekler.
  - b. [SI]: SI kaydındaki değeri bir byte olarak anlamlandırır.
  - w. [DI]: DI kaydındaki değeri iki byte olarak anlamlandırır.
- Bazı durumlarda assembler, veri türünü otomatik olarak hesaplayabilir.

MOV Komutu

- MOV komutu değerleri kopyalamak veya taşımak için kullanılır.
- *Kaynak*, anlık değer, genel amaçlı yazmaç, bellek konumu olabilir.
- *Hedef*, genel amaçlı yazmaç, bellek konumu olabilir.
- CS ve IP yazmaçlarına değer atamak için *kullanılmaz*.
- İki işlenenin boyutları aynı olmalıdır,
  - Bir *byte* veya bir *word* olabilir.
- MOV AX, BX:
  - BX yazmacındaki değeri AX yazmacına kopyalar.
- MOV [SI], 10h:
  - Bellekte SI yazmacındaki adrese 10h değerini kopyalar.

Desteklenen İşlenenler (Operands)

- MOV REG, memory
- MOV memory, REG
- MOV REG, REG
- MOV memory, immediate
- MOV REG, immediate

- **REG**: *AX, BX, CX, DX, AH, AL, BL, BH, CH, CL, DH, DL, DI, SI, BP, SP.*
- **memory**: *[BX], [BX+SI+7], değişken.*
- **immediate**: 5, -24, 3Fh, 10001101b ..

Desteklenen İşlenenler (Segment Registers)

- MOV SREG, memory
- MOV memory, SREG
- MOV REG, SREG
- MOV SREG, REG

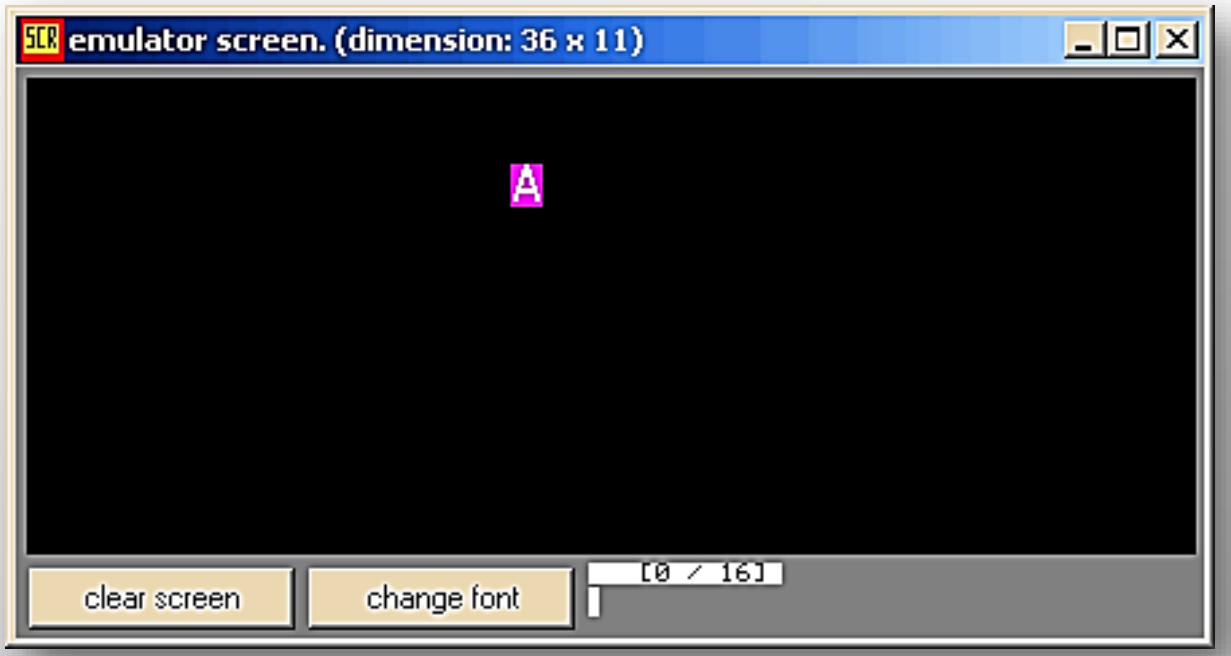
- **SREG**: *DS, ES, SS,* ve *CS* (sadece ikinci işlenen olarak).
- **REG**: *AX, BX, CX, DX, AH, AL, BL, BH, CH, CL, DH, DL, DI, SI, BP, SP.*
- **memory**: *[BX], [BX+SI+7], değişken* ..

Program Örneği

```
ORG 100h          ; Bir segment .com programı için gereklidir.
MOV AX, 0B800h    ; AX'e B800h değeri atar.
MOV DS, AX        ; AX'in değerini DS'ye kopyalar.
MOV CL, 'A'       ; CL'ye A harfinin ASCII kodunu atar(41h).
MOV CH, 11011111b ; CH'e 11011111b değeri atar.
MOV BX, 15Eh      ; BX'e 15Eh değeri atar.
MOV [BX], CX      ; CX'in içeriğini B800:015E adresine kopyalar
RET               ; İşletim sistemine geri döner.
```

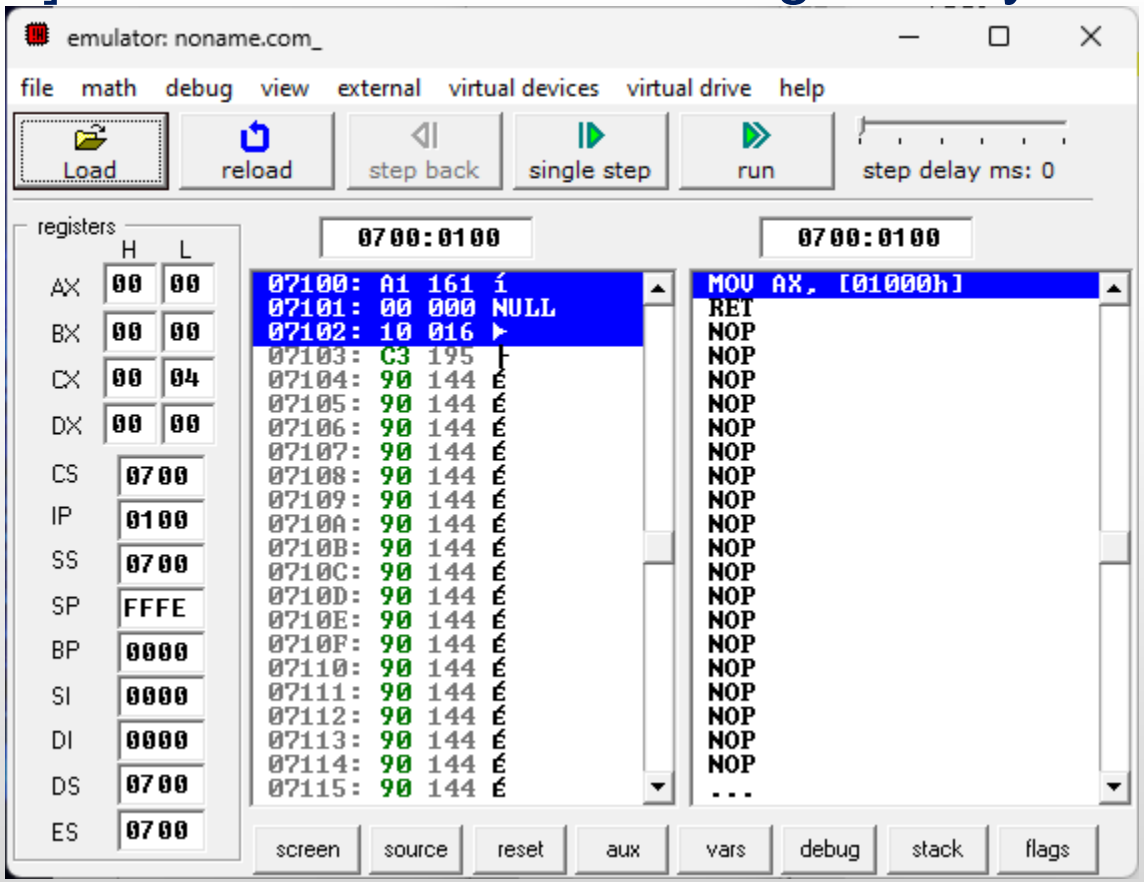
Program Örneği

- Program, video belleği üzerinde 'A' harfini belirli bir konuma kopyalar.
- *MOV*, kaynak ve hedef arasında değer kopyalamak için kullanılır.
- *ORG*, programın başlangıç adresini belirler.
- ***;*** yorum satırları için kullanılır.



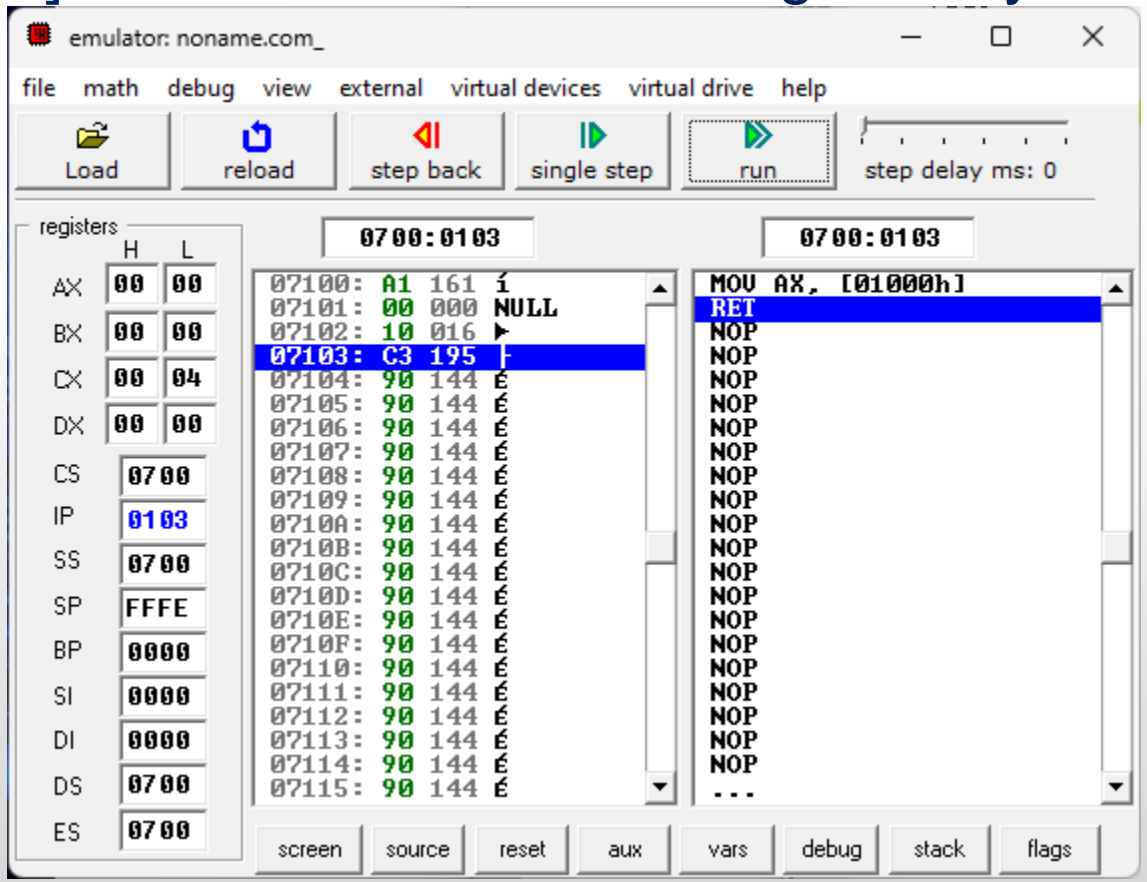
Bellekten Bir Değeri Yazmaca Taşıma

- MOV AX, [0x1000] ; Bellek adresindeki değeri AX yazmacına taşır



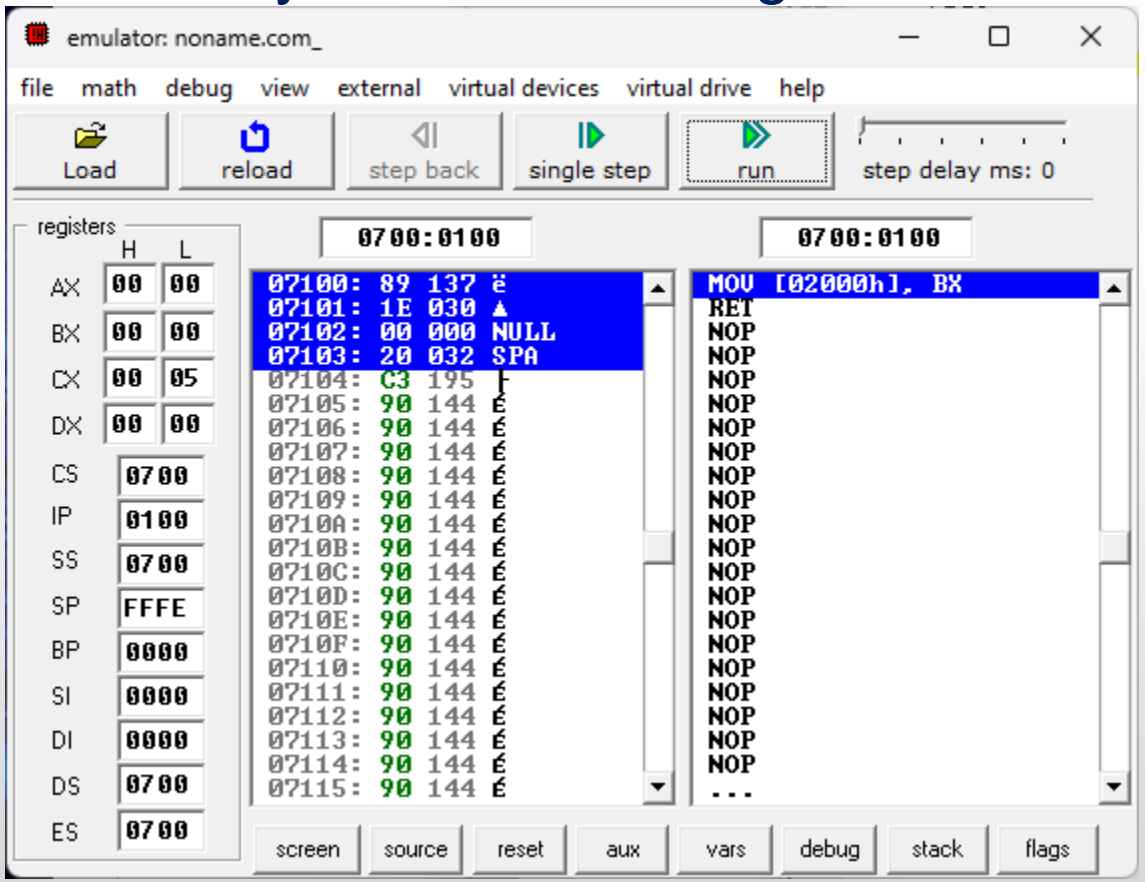
Bellekten Bir Değeri Yazmaca Taşıma

- MOV AX, [0x1000] ; Bellek adresindeki değeri AX yazmacına taşıma



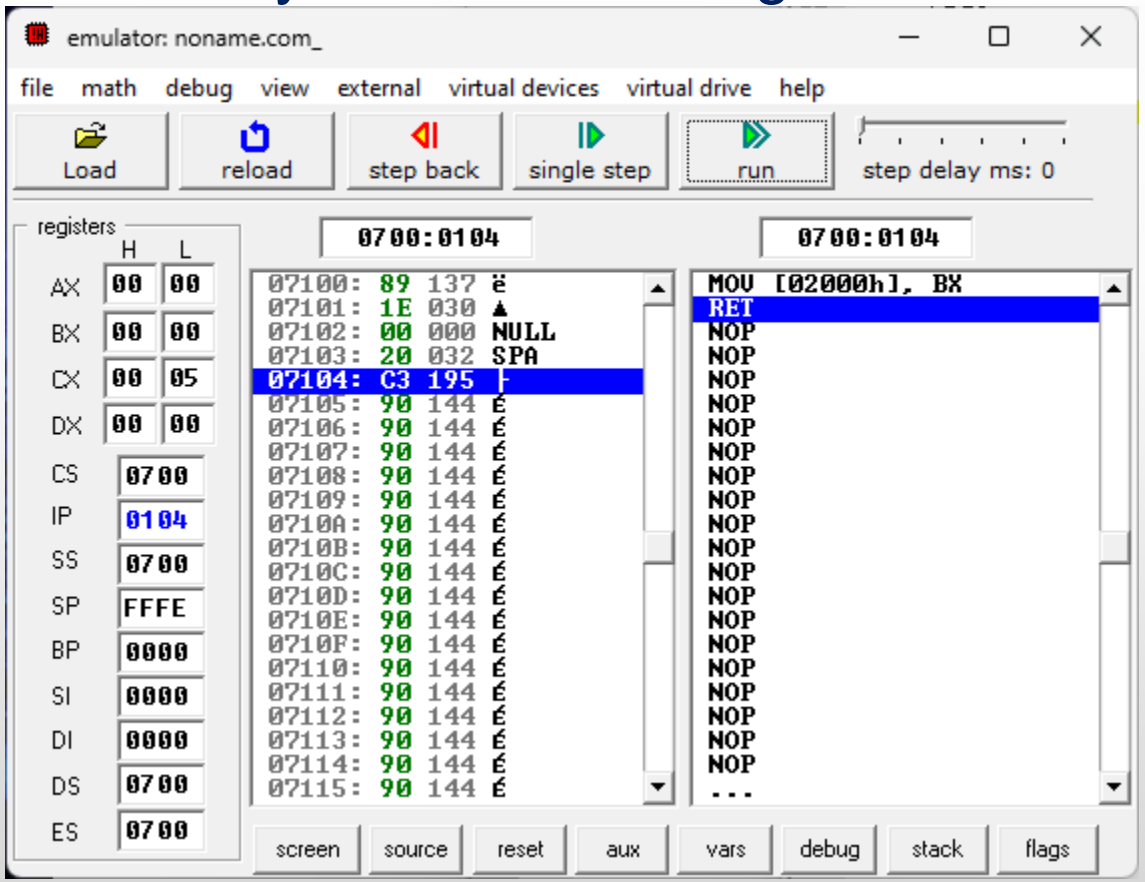
Yazmaçtaki Değeri Belleğe Taşıma

- MOV [0x2000], BX ; BX yazmacındaki değeri bellek adresine taşır



Yazmaçtaki Değeri Belleğe Taşıma

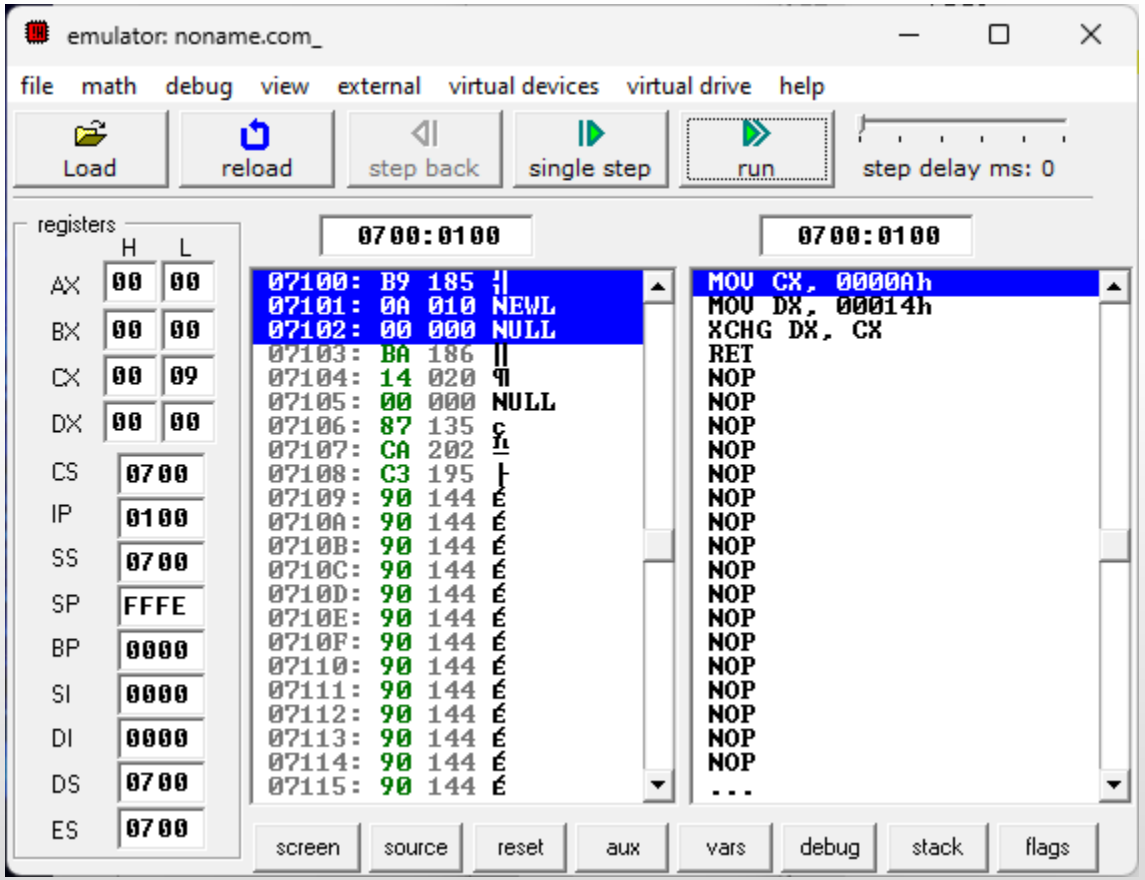
- MOV [0x2000], BX ; BX yazmacındaki değeri bellek adresine taşır



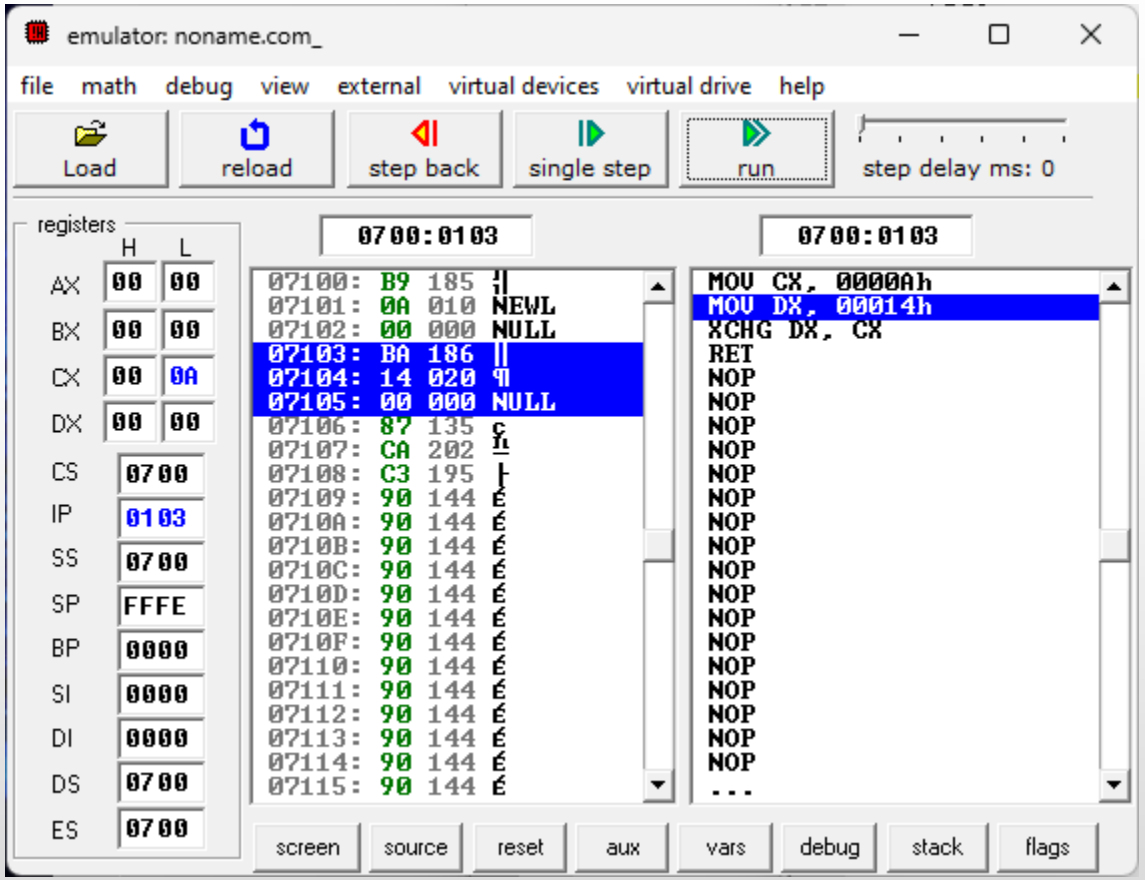
İki Yazmaç Değerini Takas

- MOV CX, 10 ; CX yazmacına 10 değerini ata
- MOV DX, 20 ; DX yazmacına 20 değerini ata
- XCHG CX, DX ; CX ve DX yazmaçlarının değerlerini değiştirme

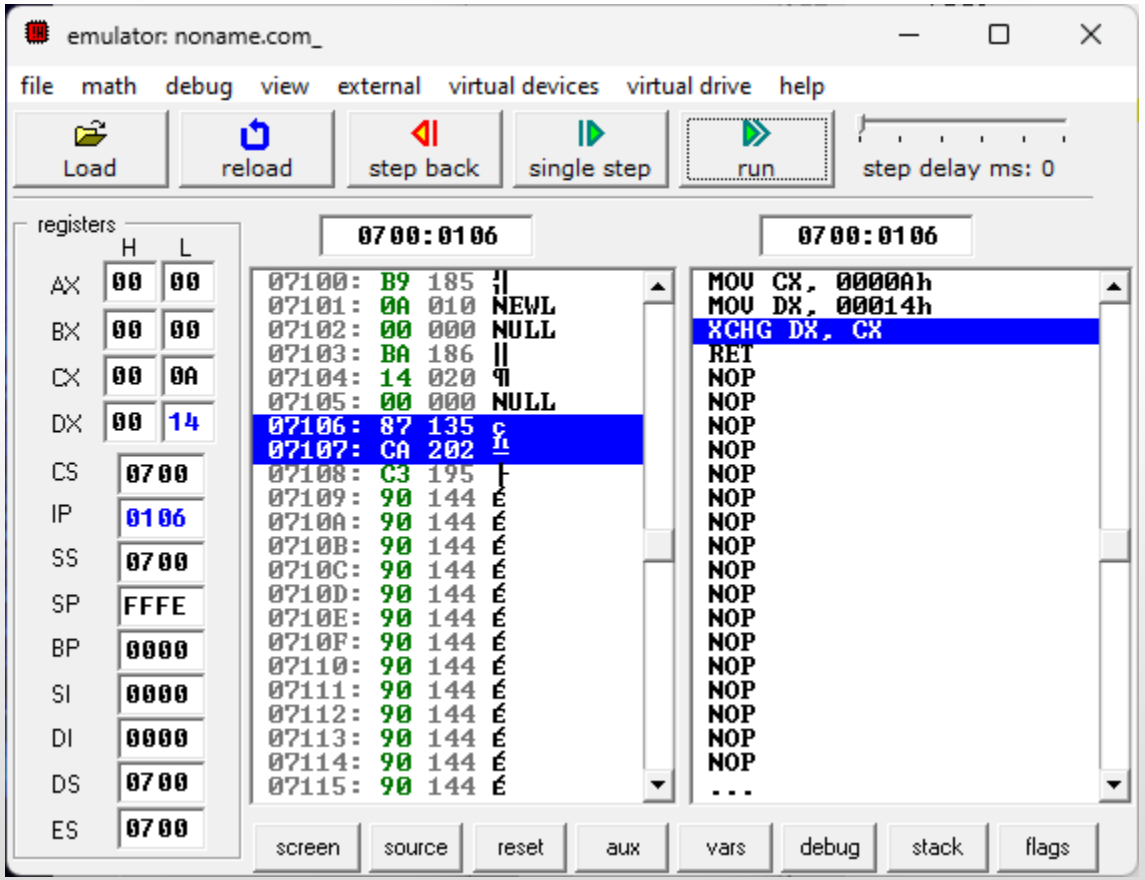
İki Yazmaç Değerini Takas



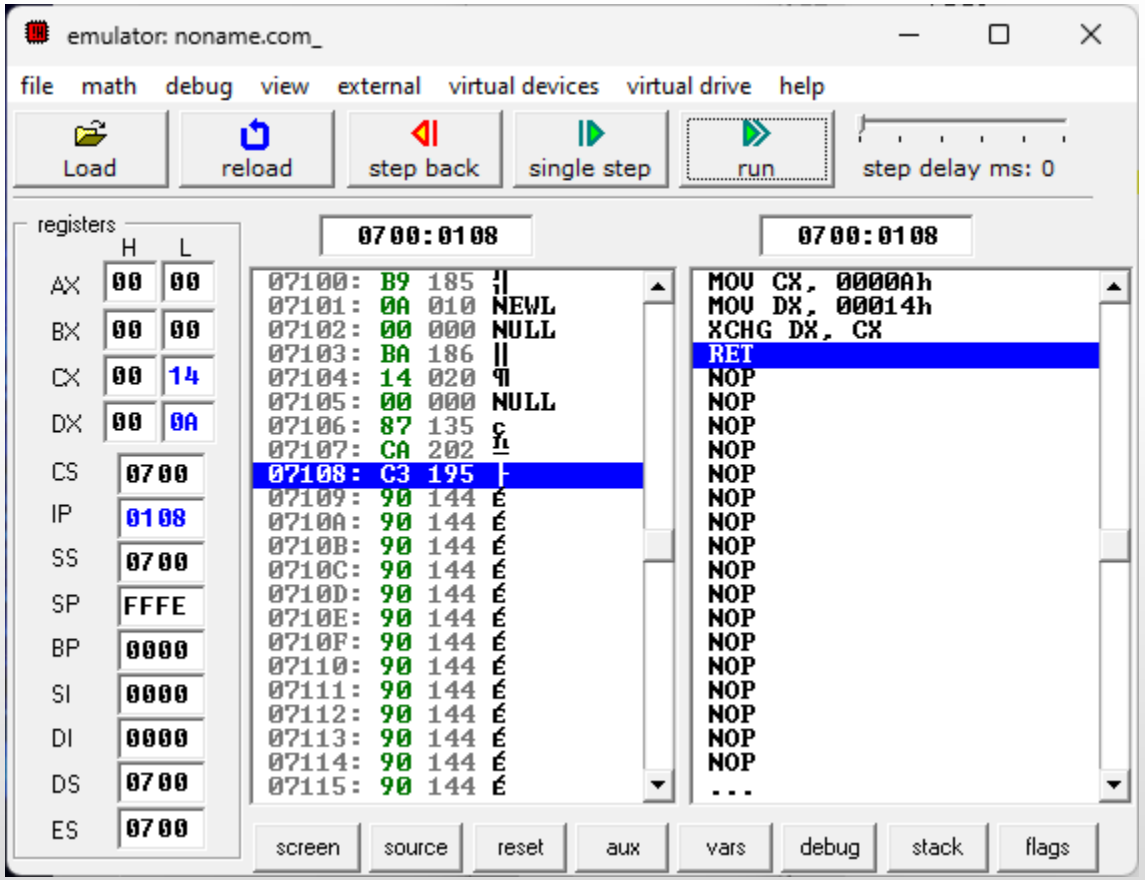
İki Yazmaç Değerini Takas



İki Yazmaç Değerini Takas



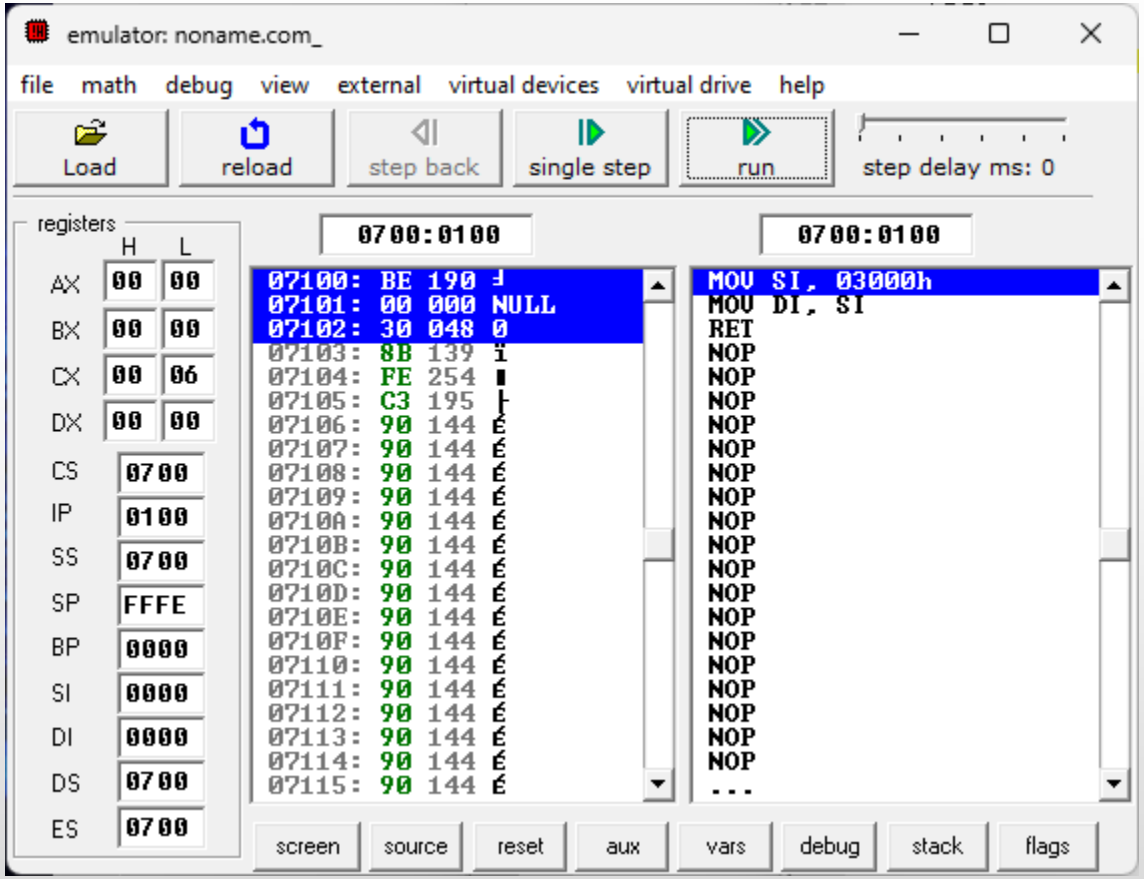
İki Yazmaç Değerini Takas



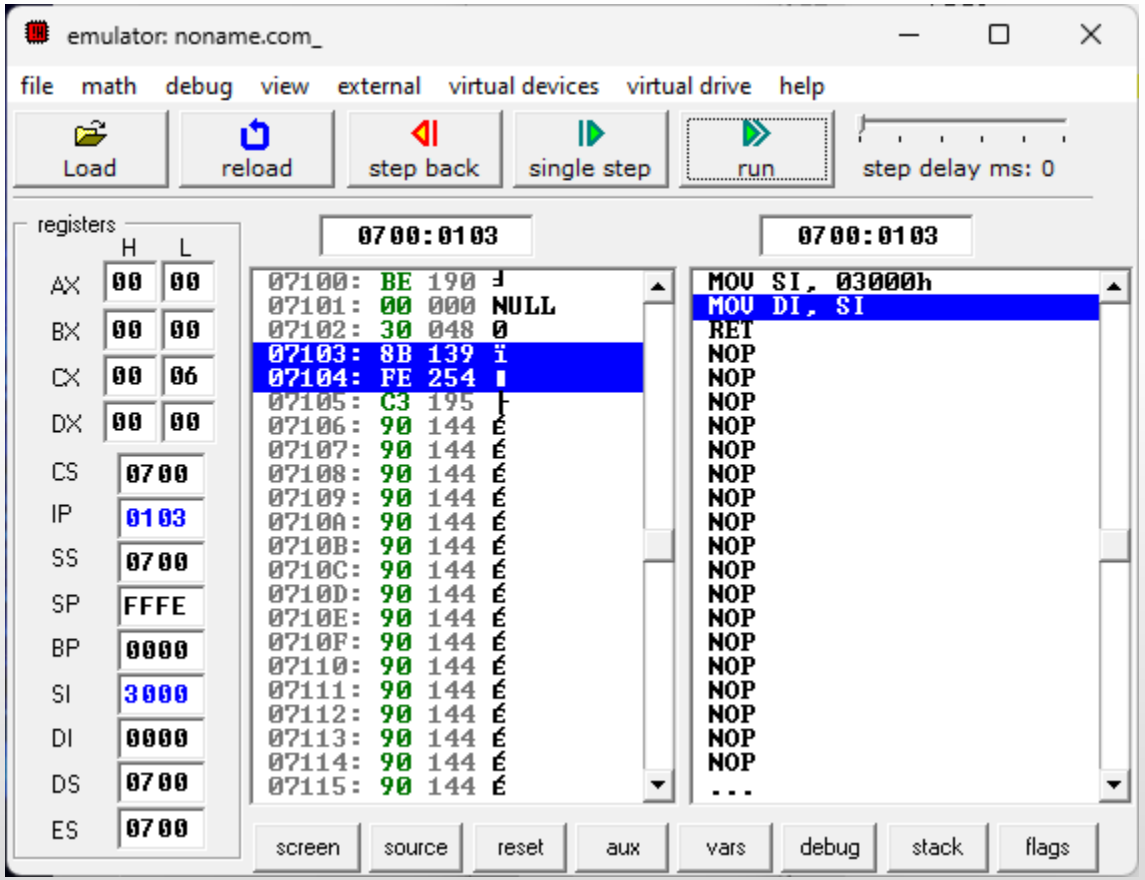
Bir Yazmaca Başka Yazmacın İçeriğini Kopyalama

- MOV SI, 0x3000 ; SI yazmacına bellek adresi atama
- MOV DI, SI ; DI yazmacına SI yazmacının içeriğini kopyalama

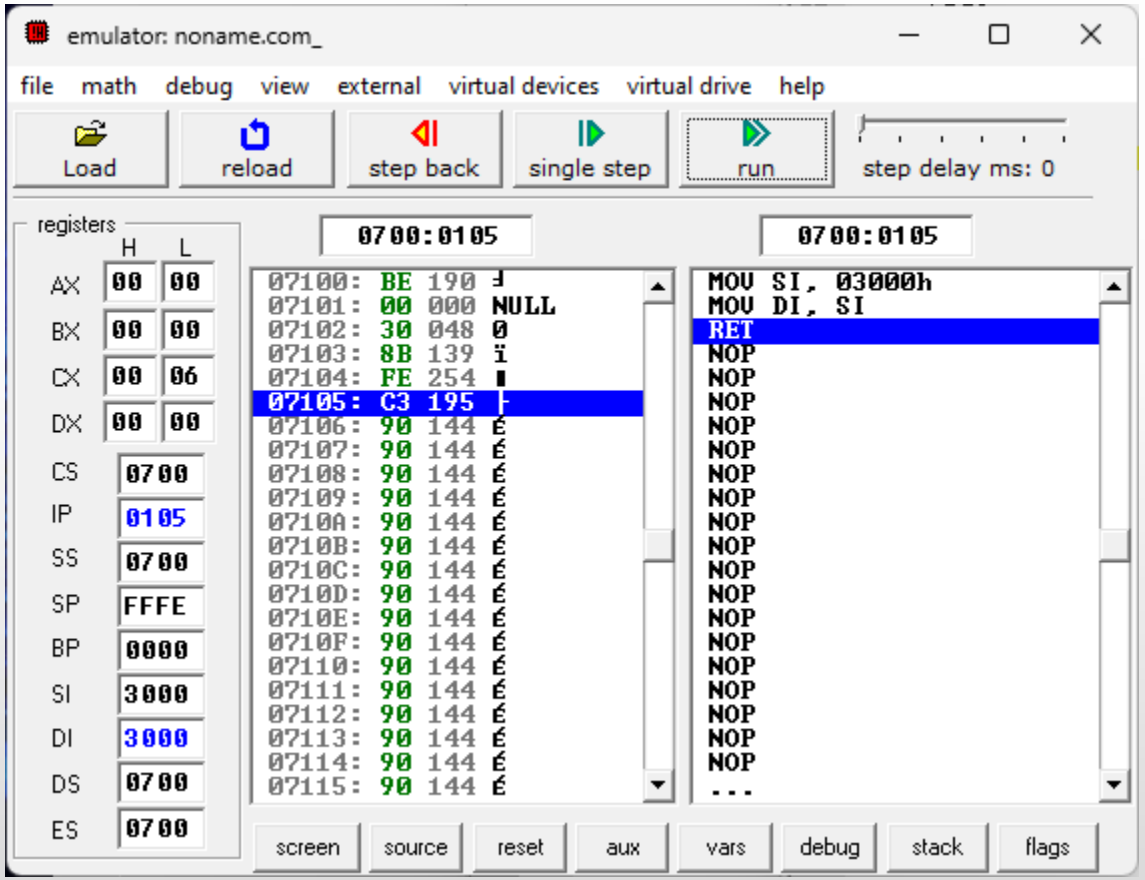
Bir Yazmaca Başka Yazmacın İçeriğini Kopyalama



Bir Yazmaca Başka Yazmacın İçeriğini Kopyalama

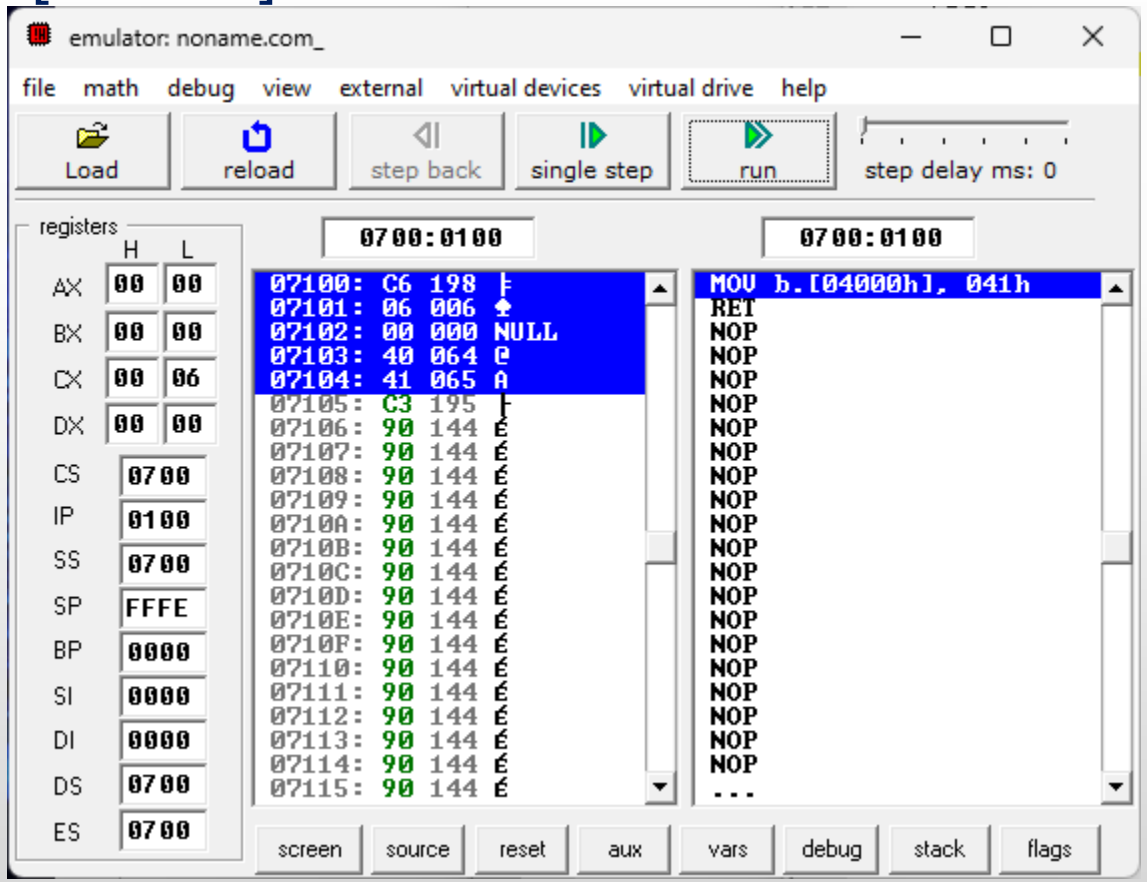


Bir Yazmaca Başka Yazmacın İçeriğini Kopyalama

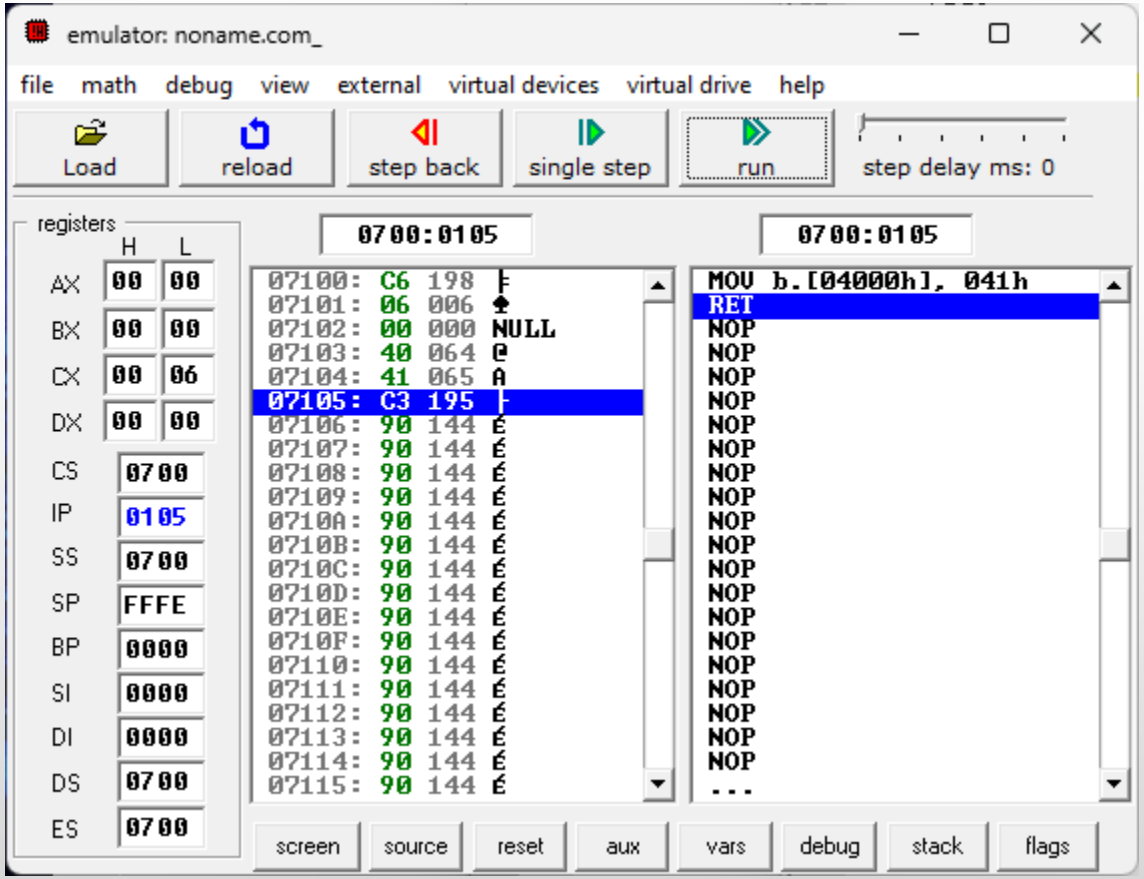


Bir Bellek Adresine Sabit Bir Değeri Yazma

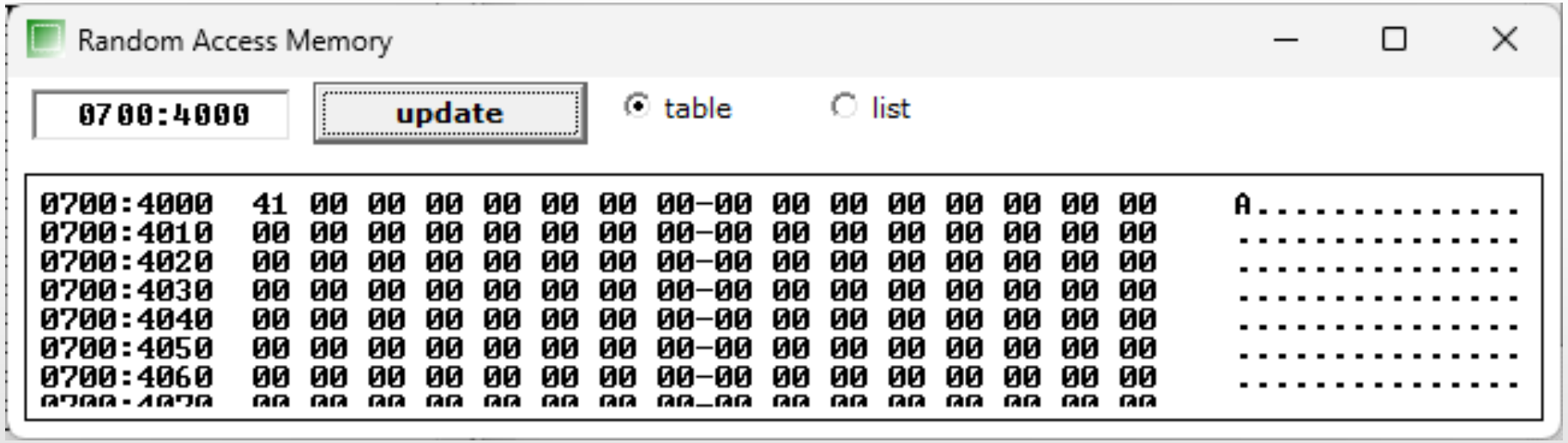
- MOV BYTE PTR [0x4000], 65 ; Bellek adresine 'A' karakterini yazma



Bir Bellek Adresine Sabit Bir Değeri Yazma



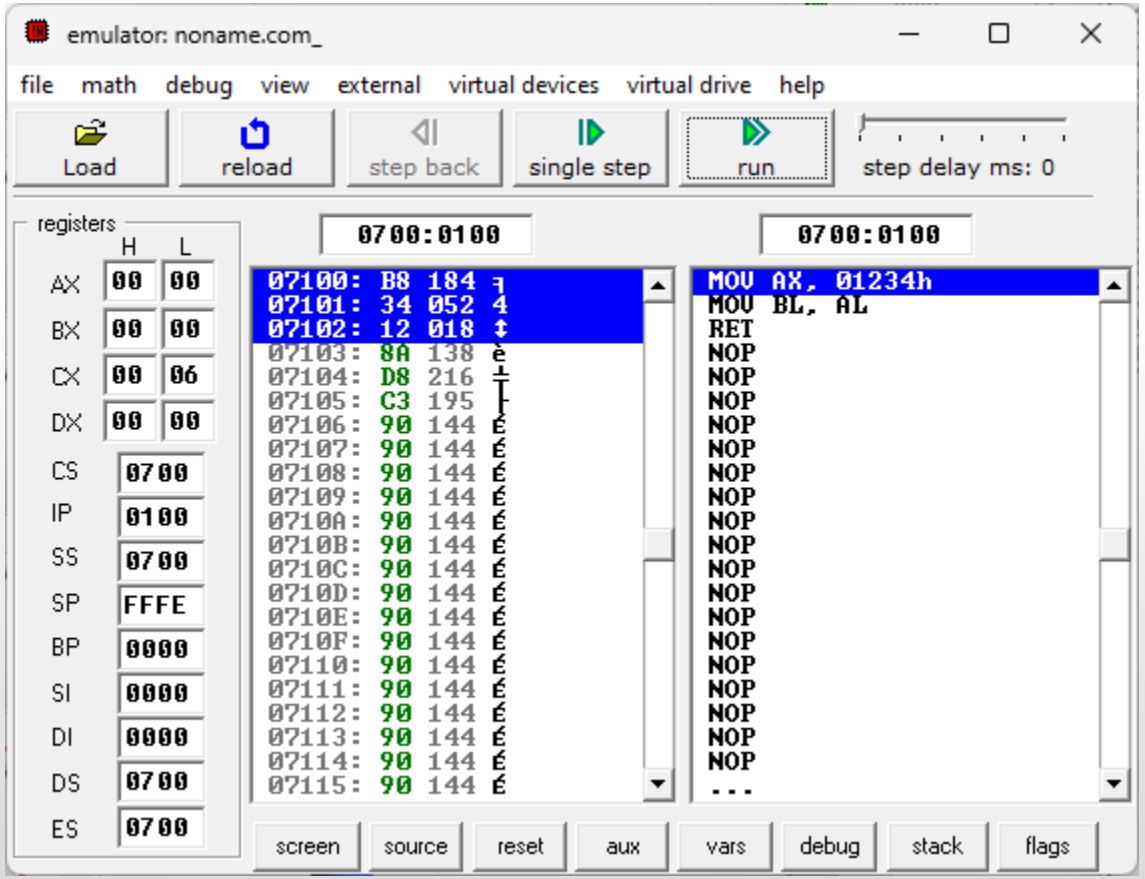
Bir Bellek Adresine Sabit Bir Değeri Yazma



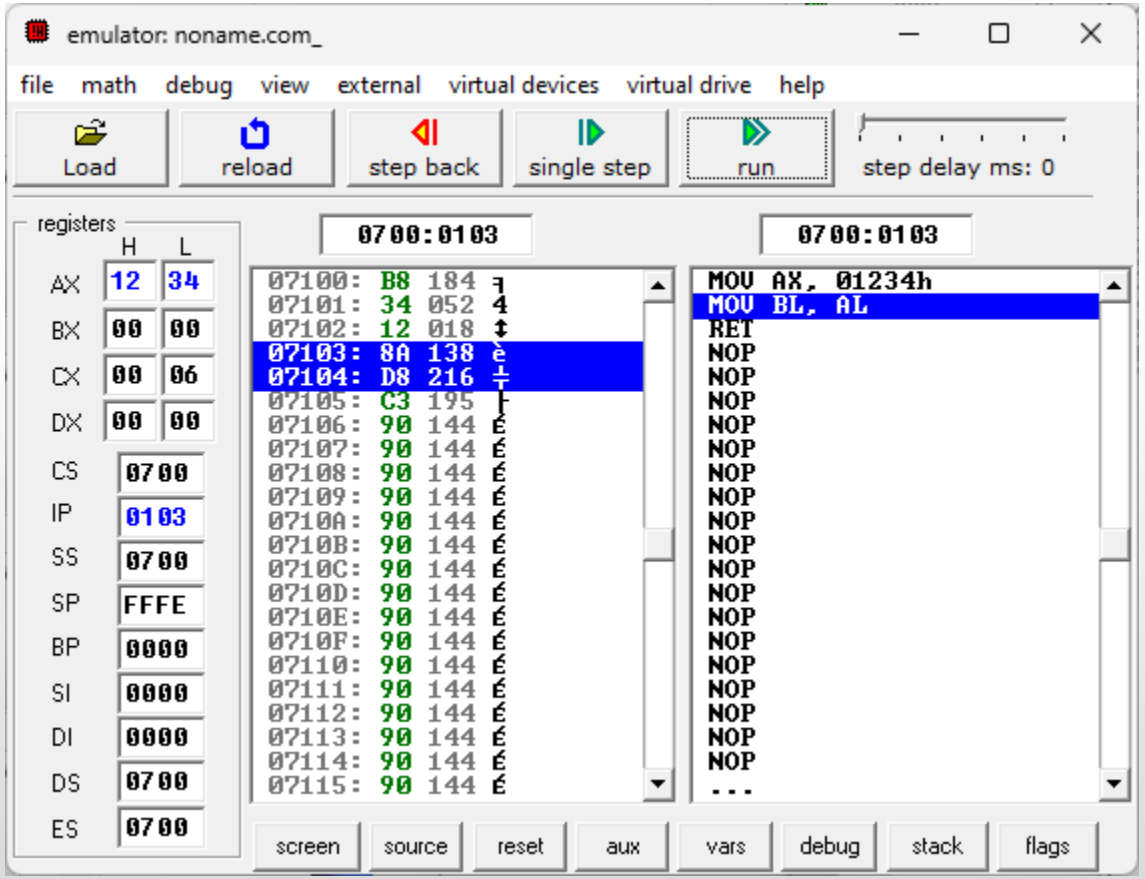
Bir Yazmacın Alt Kısmını Diğerine Kopyalama

- MOV AX, 0x1234 ; AX yazmacına 0x1234 değeri ata
- MOV BL, AL ; AL yazmacının alt kısmını BL yazmacına kopyala

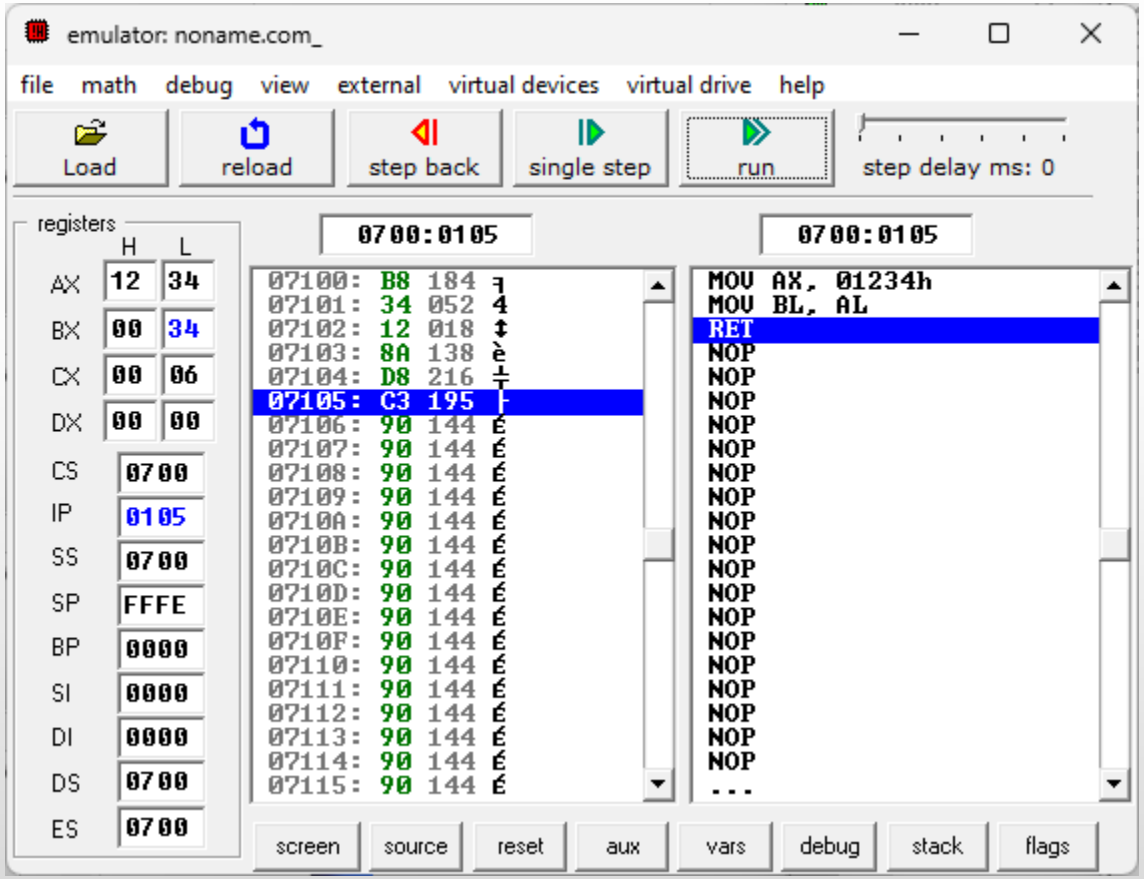
Bir Yazmacın Alt Kısmını Diğerine Kopyalama



Bir Yazmacın Alt Kısmını Diğerine Kopyalama



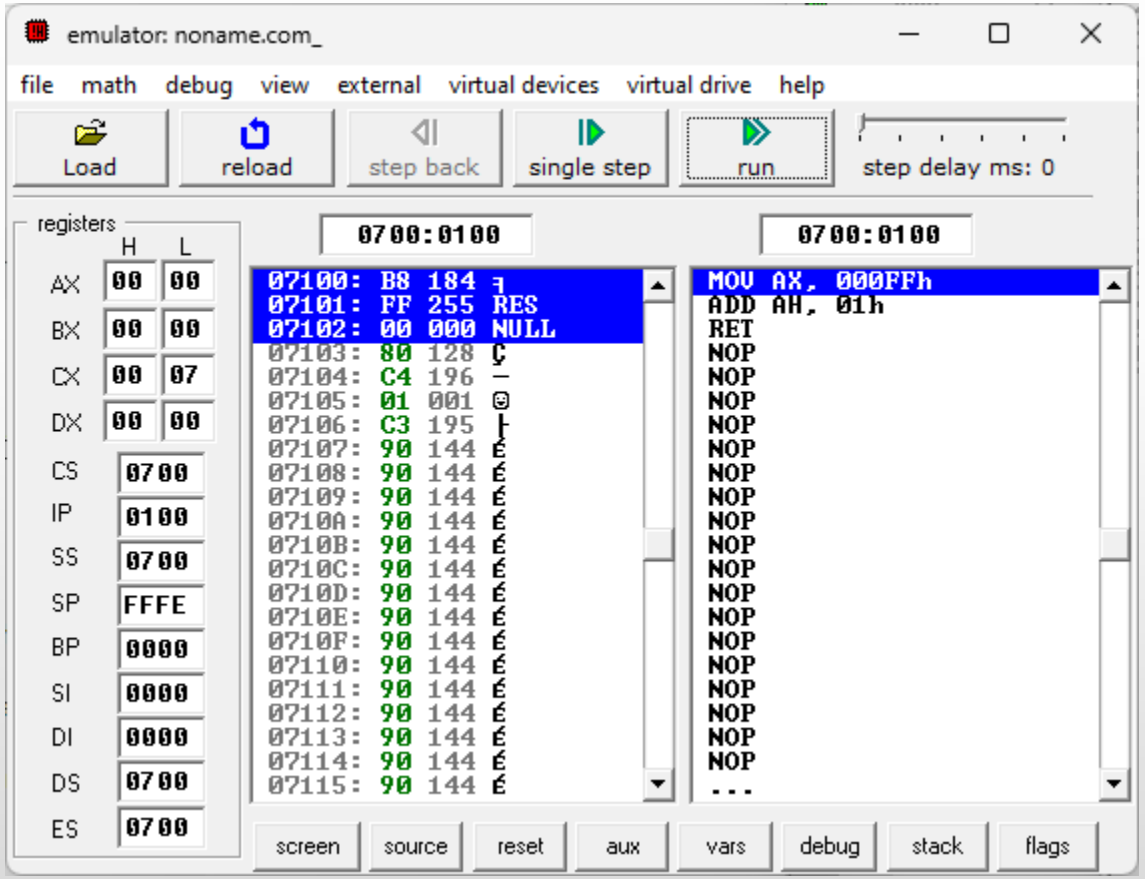
Bir Yazmacın Alt Kısmını Diğerine Kopyalama



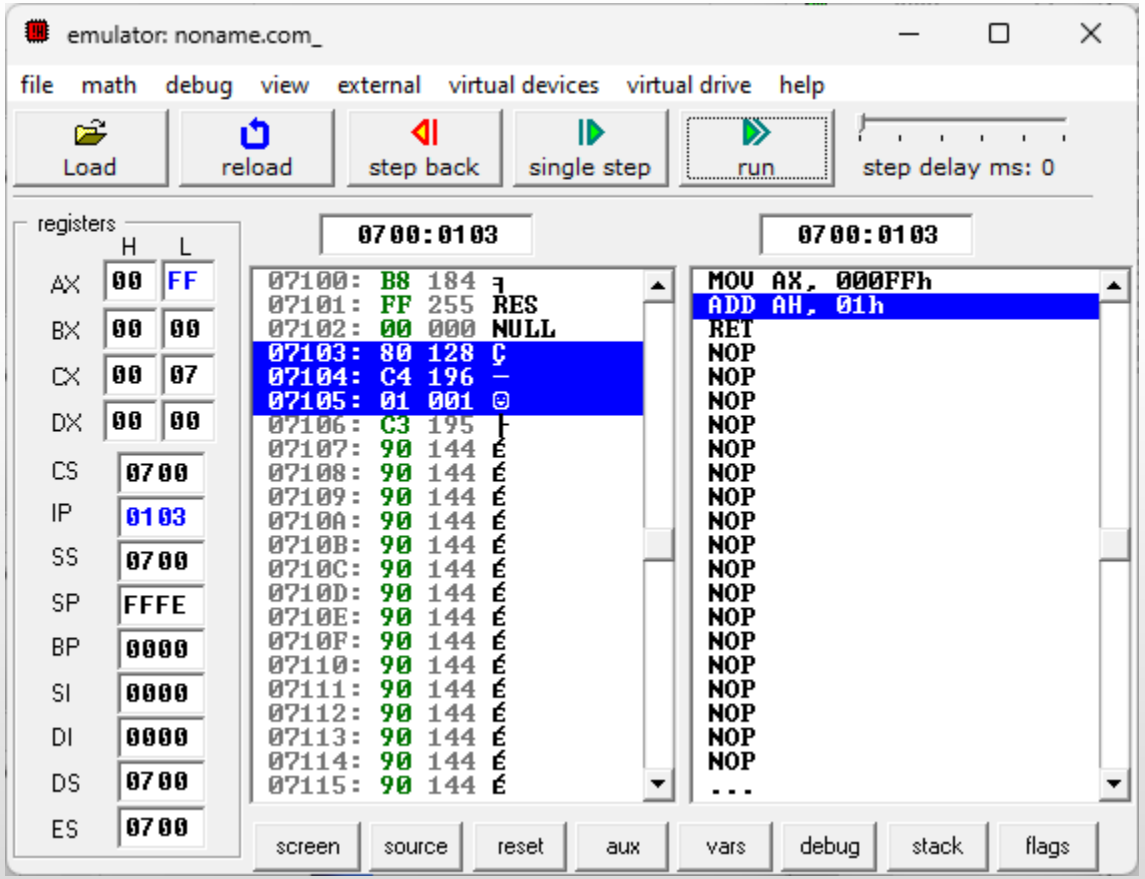
Bir Yazmacın Alt Kısmına Ekleme

- MOV AX, 0x00FF ; AX yazmacına 0x00FF değeri taşı
- ADD AH, 0x01 ; AH yazmacına 0x01 ekle

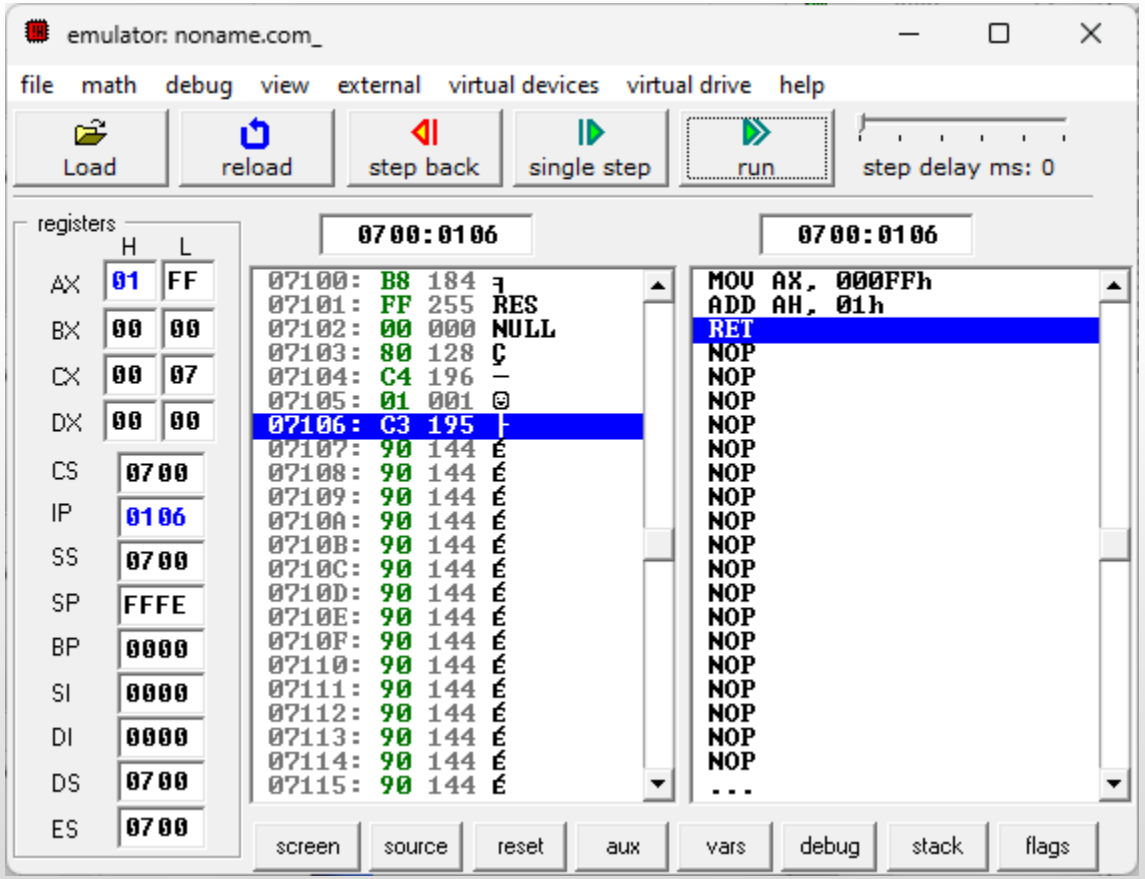
Bir Yazmacın Alt Kısmına Ekleme



Bir Yazmacın Alt Kısmına Ekleme



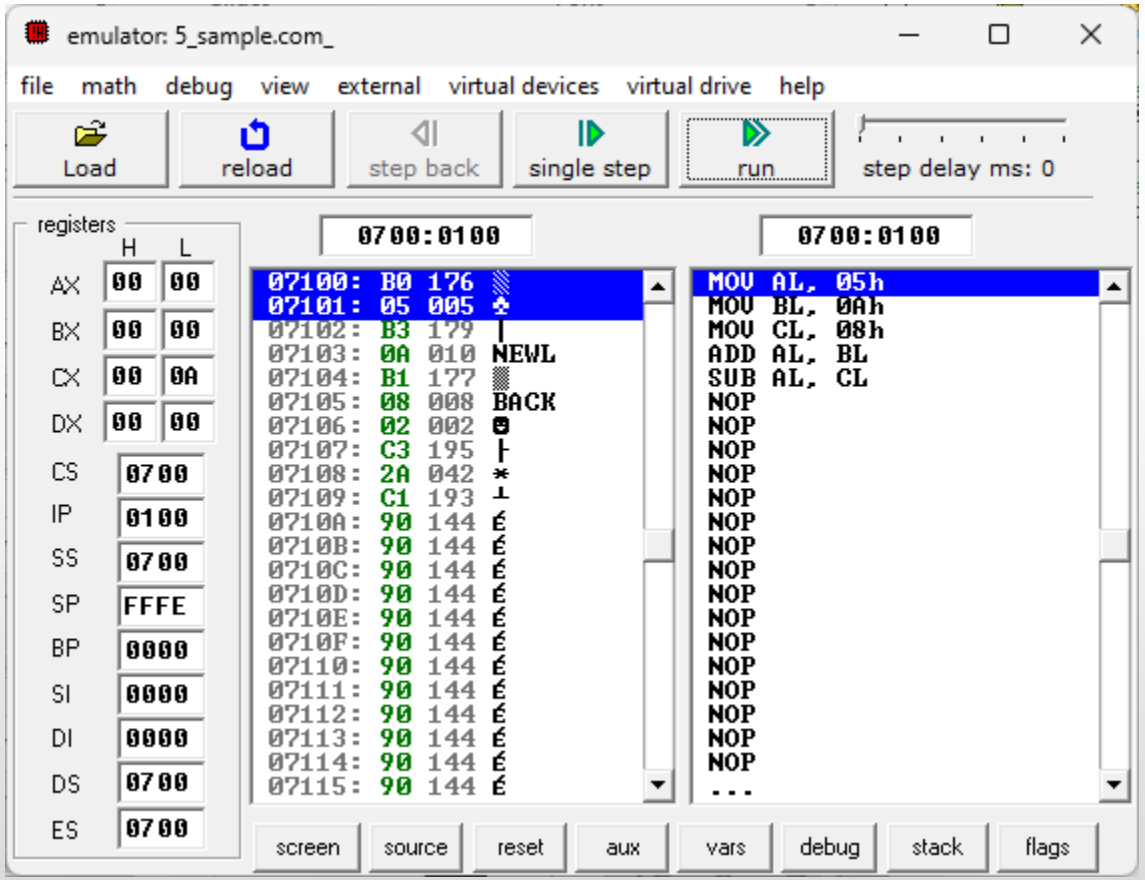
Bir Yazmacın Alt Kısmına Ekleme



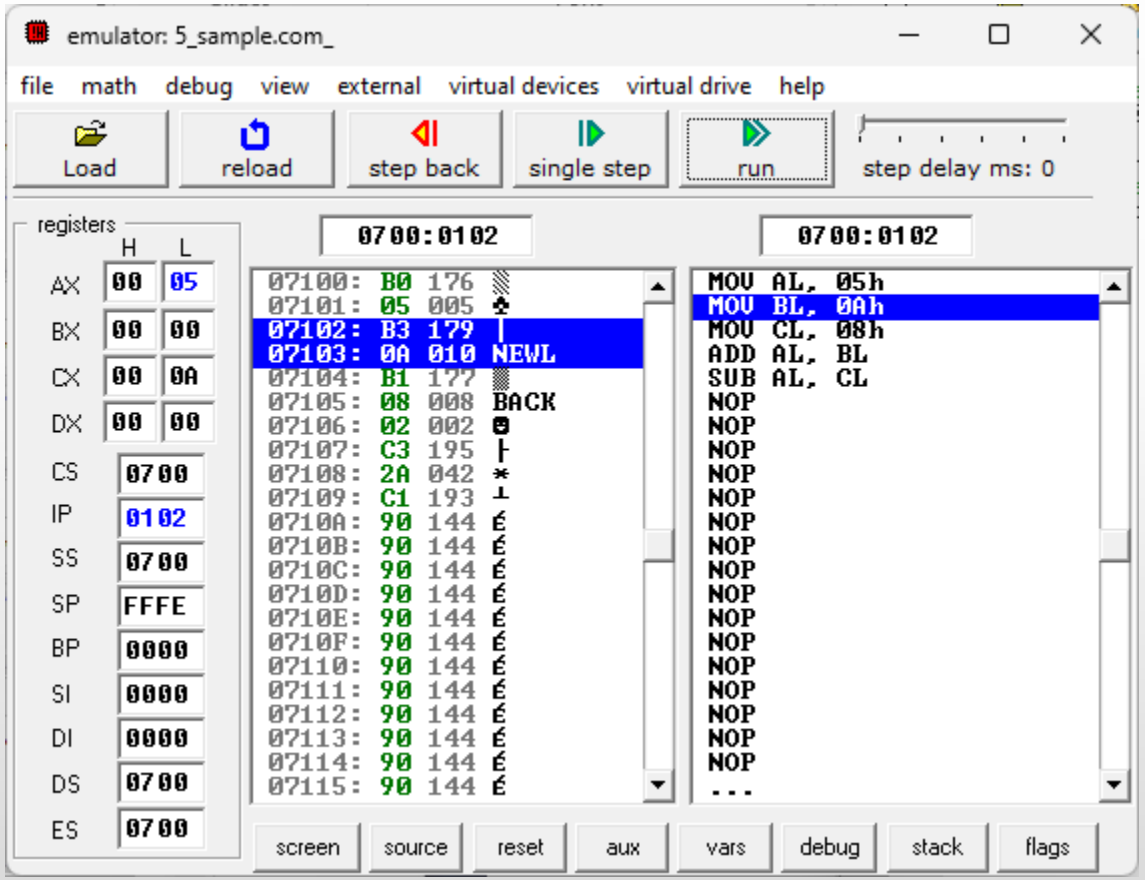
İkilik, Sekizlik ve Onaltılık Değer Atama

- org 100h
- ; load binary value:
- mov al, 00000101b
- ; load hex value:
- mov bl, 0ah
- ; load octal value:
- mov cl, 10o
- ; 5 + 10 = 15 (0fh)
- add al, bl
- ; 15 - 8 = 7
- sub al, cl

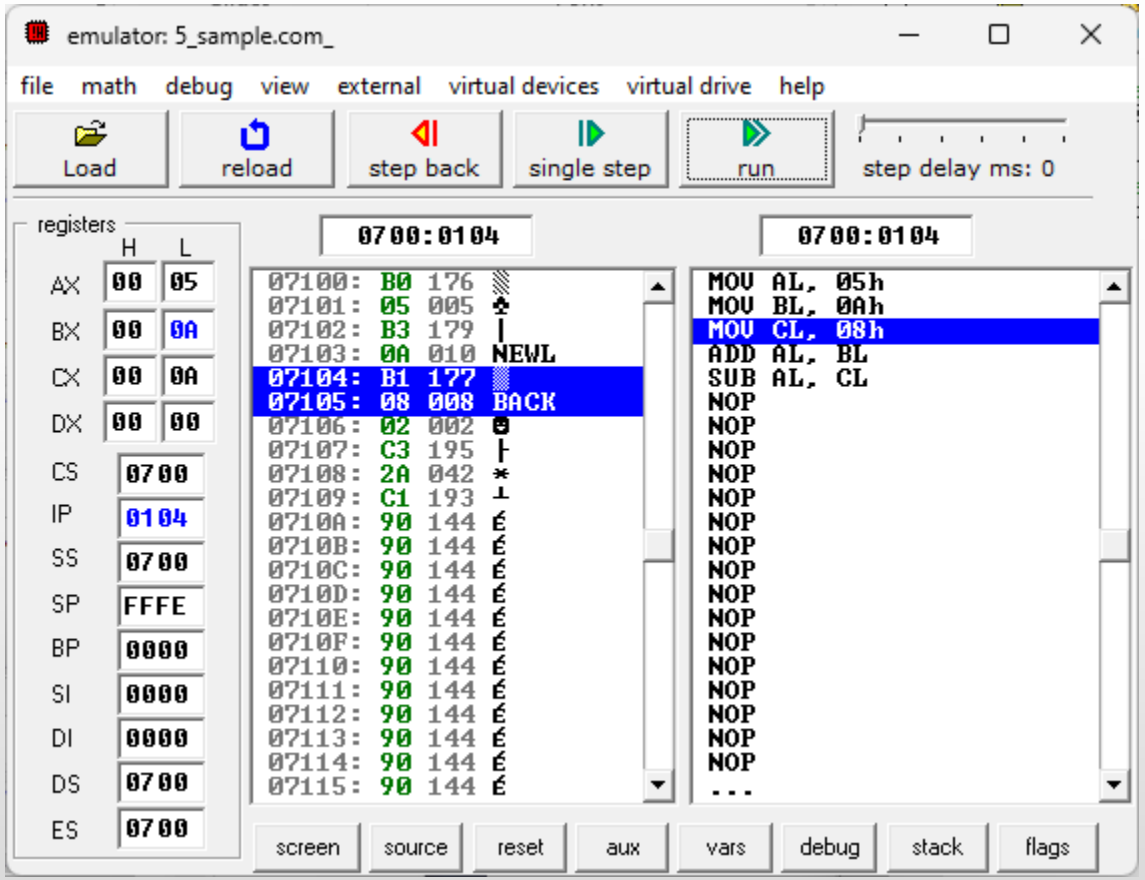
İkilik, Sekizlik ve Onaltılık Değer Atama



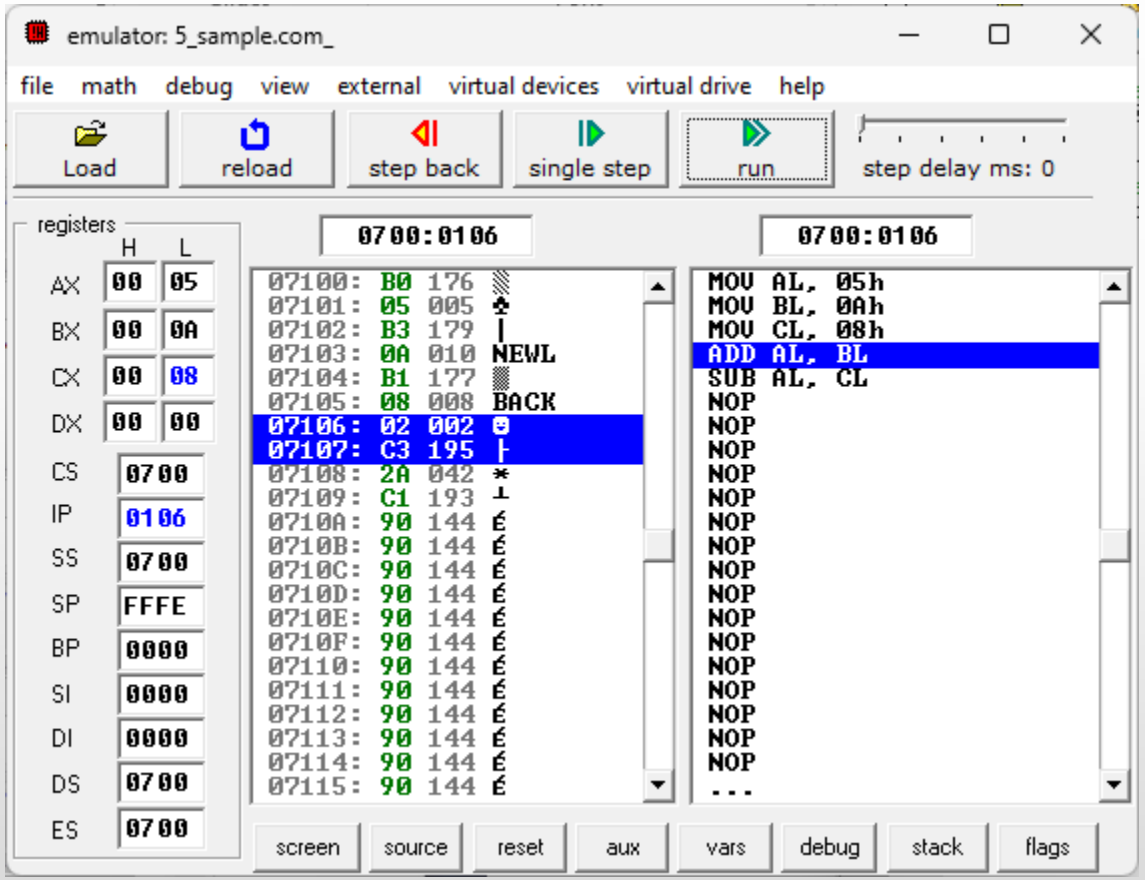
İkilik, Sekizlik ve Onaltılık Değer Atama



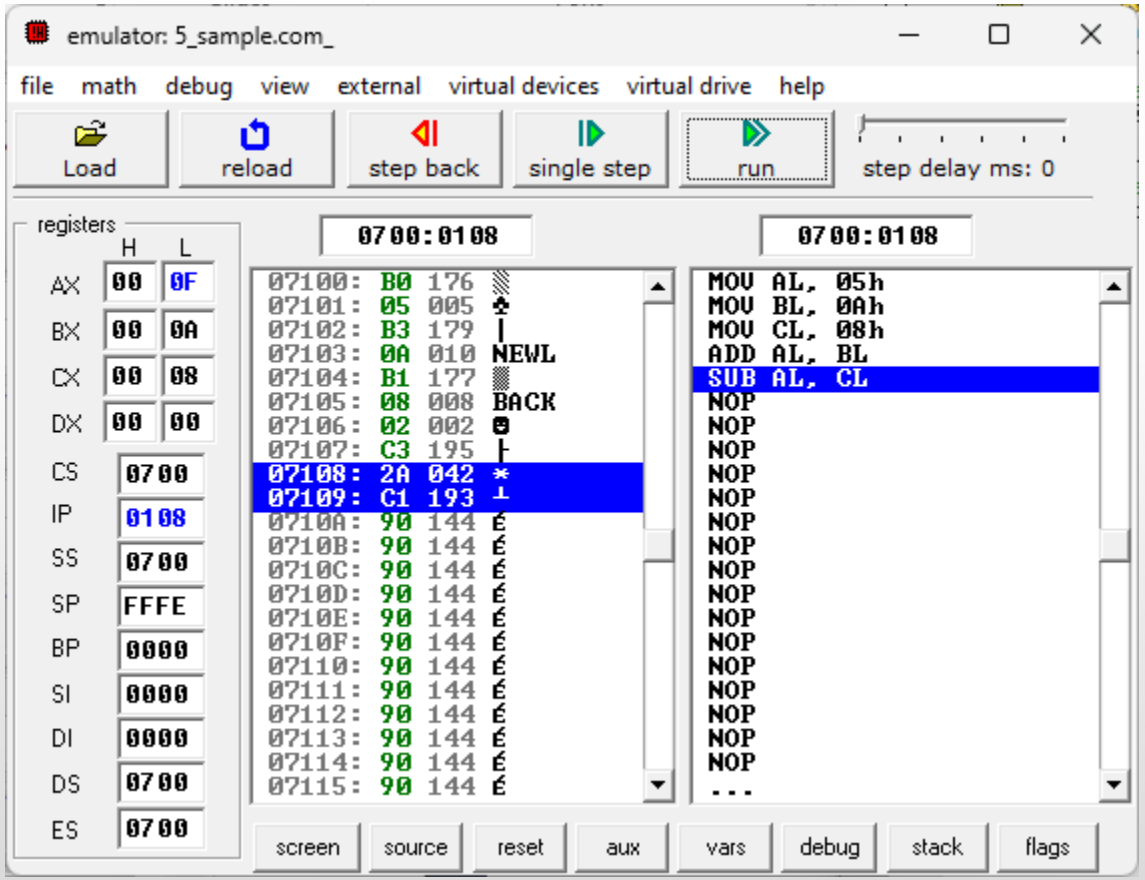
İkilik, Sekizlik ve Onaltılık Değer Atama



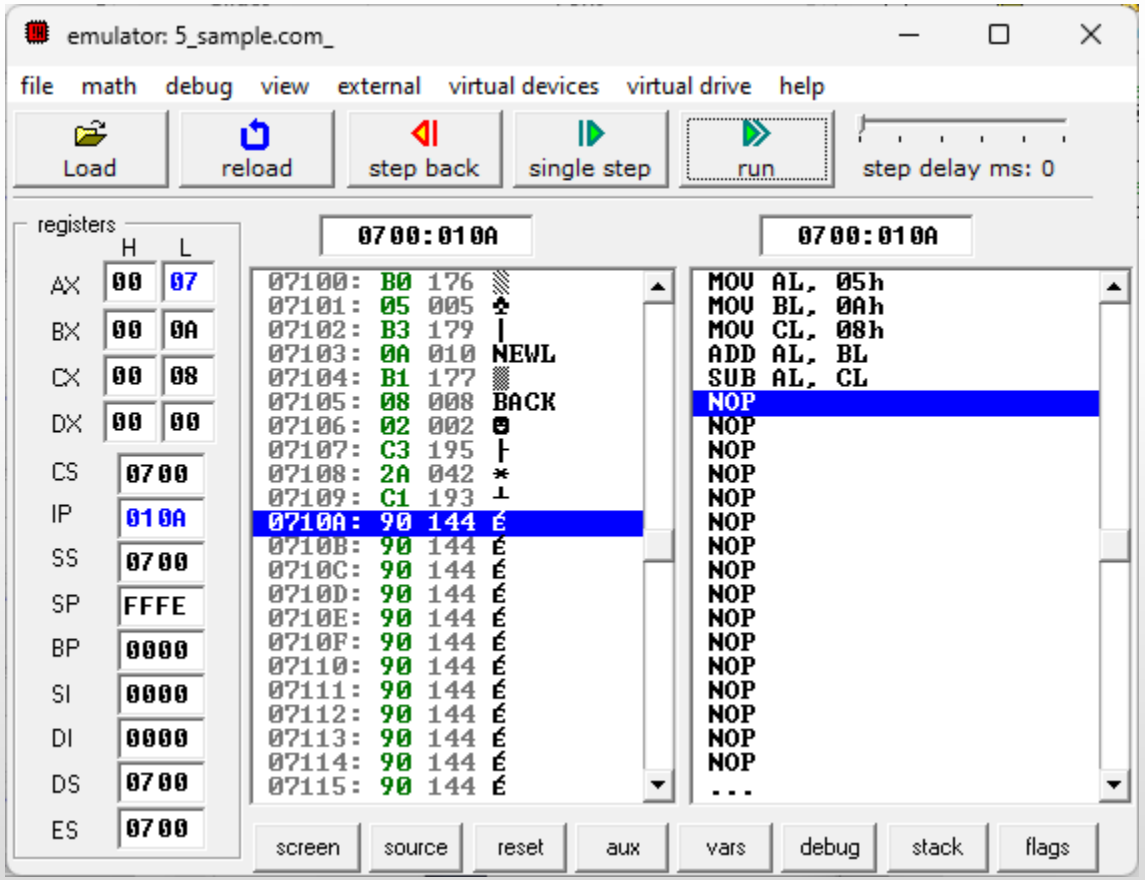
İkilik, Sekizlik ve Onaltılık Değer Atama



İkilik, Sekizlik ve Onaltılık Değer Atama



İkilik, Sekizlik ve Onaltılık Değer Atama



SON