



# **Bölüm 4: Bellek Erişimi**

## **Mikroişlemciler**



# Bellek Erişimi

- Belleğe erişmek için BX, SI, DI, BP yazmaçları kullanılır.
- Bu yazmaçlar [ ] sembolleri içerisinde birleştirilerek,
  - 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 Bađıl Konum

- Kesim yazmacındaki (CS, DS, SS, ES) deđere kesim (*segment*),
- Genel yazmaçlardaki (BX, SI, DI, BP) deđere bađıl konum (*offset*) denir.
- Kesim, belleđin bloklarını temsil ederken,
  - Bađıl konum, o bloktaki belirli bir konumu temsil eder.
- Kesim ve bađıl konum, birleřtirilerek fiziksel bellek adresi oluřturulur.
- DS deđeri 1234h ve SI deđeri 7890h olduđunda;
  - 1234:7890 olarak gosterilebilir.
  - 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 ..*



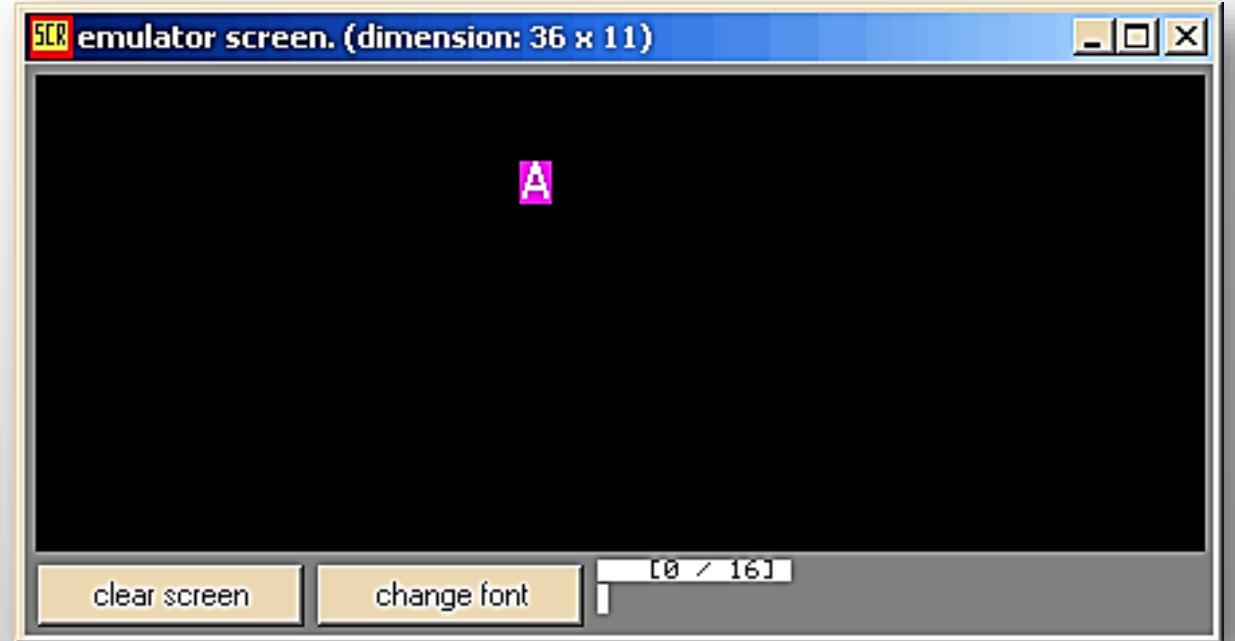
# Program Örneği

```
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 Örneđi

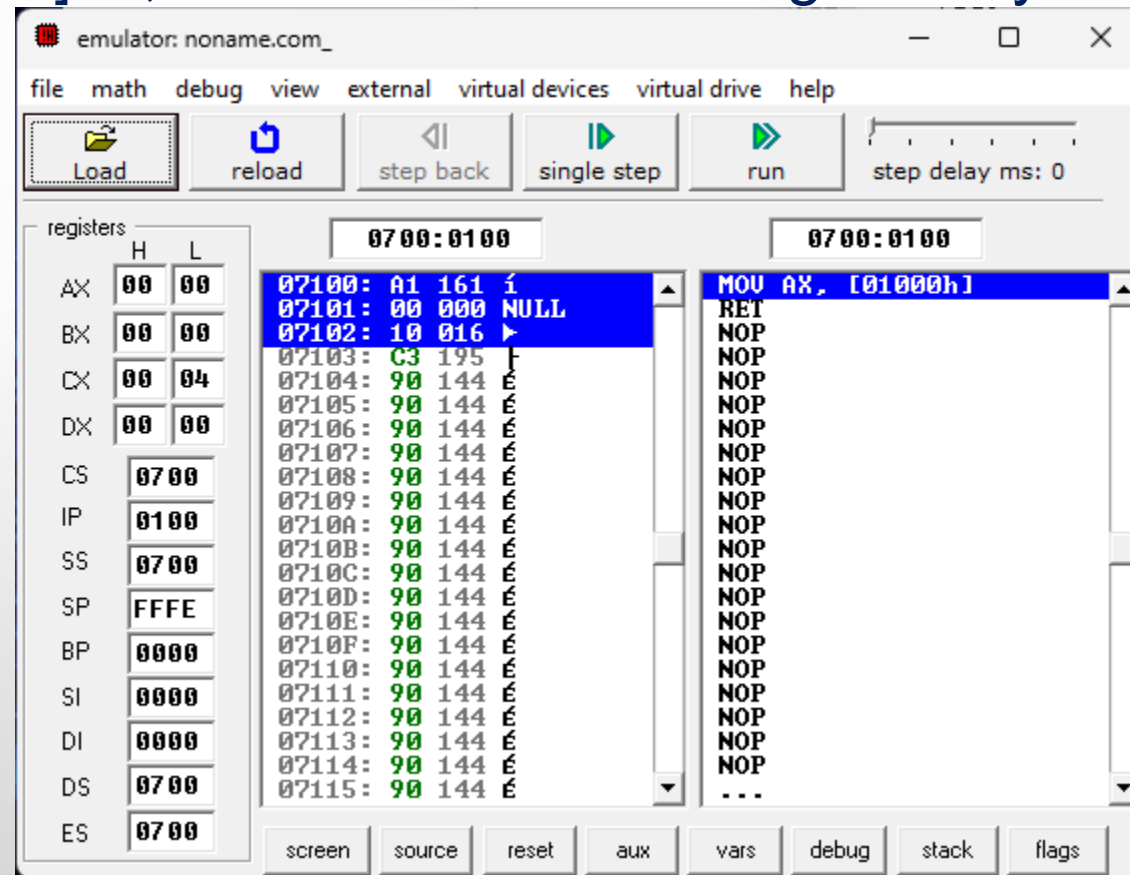
- Program, video belleđi üzerinde 'A' harfini belirli bir konuma kopyalar.
- *MOV*, kaynak ve hedef arasında deđer kopyalamak için kullanılır.
- *ORG*, programın başlangıç adresini belirler.
- ; yorum satırları için kullanılır.





# Bellekten Bir Deęeri Yazmaca Taşıma

- MOV AX, [0x1000] ; Bellek adresindeki deęeri AX yazmacına taşıır





# Bellekten Bir Deęeri Yazmaca Taşıma

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

The screenshot shows an emulator window titled "emulator: noname.com\_". The interface 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, run, and a step delay slider set to 0 ms. Below the toolbar, the registers section shows the following values:

Register	H	L
AX	00	00
BX	00	00
CX	00	04
DX	00	00
CS	07 00	
IP	01 03	
SS	07 00	
SP	FF FE	
BP	00 00	
SI	00 00	
DI	00 00	
DS	07 00	
ES	07 00	

The main window displays the instruction list. The current instruction is highlighted in blue:

```
07100: A1 161 i
07101: 00 000 NULL
07102: 10 016
07103: C3 195 |
07104: 90 144 E
07105: 90 144 E
07106: 90 144 E
07107: 90 144 E
07108: 90 144 E
07109: 90 144 E
0710A: 90 144 E
0710B: 90 144 E
0710C: 90 144 E
0710D: 90 144 E
0710E: 90 144 E
0710F: 90 144 E
07110: 90 144 E
07111: 90 144 E
07112: 90 144 E
07113: 90 144 E
07114: 90 144 E
07115: 90 144 E
```

The instruction list on the right shows the current instruction being executed:

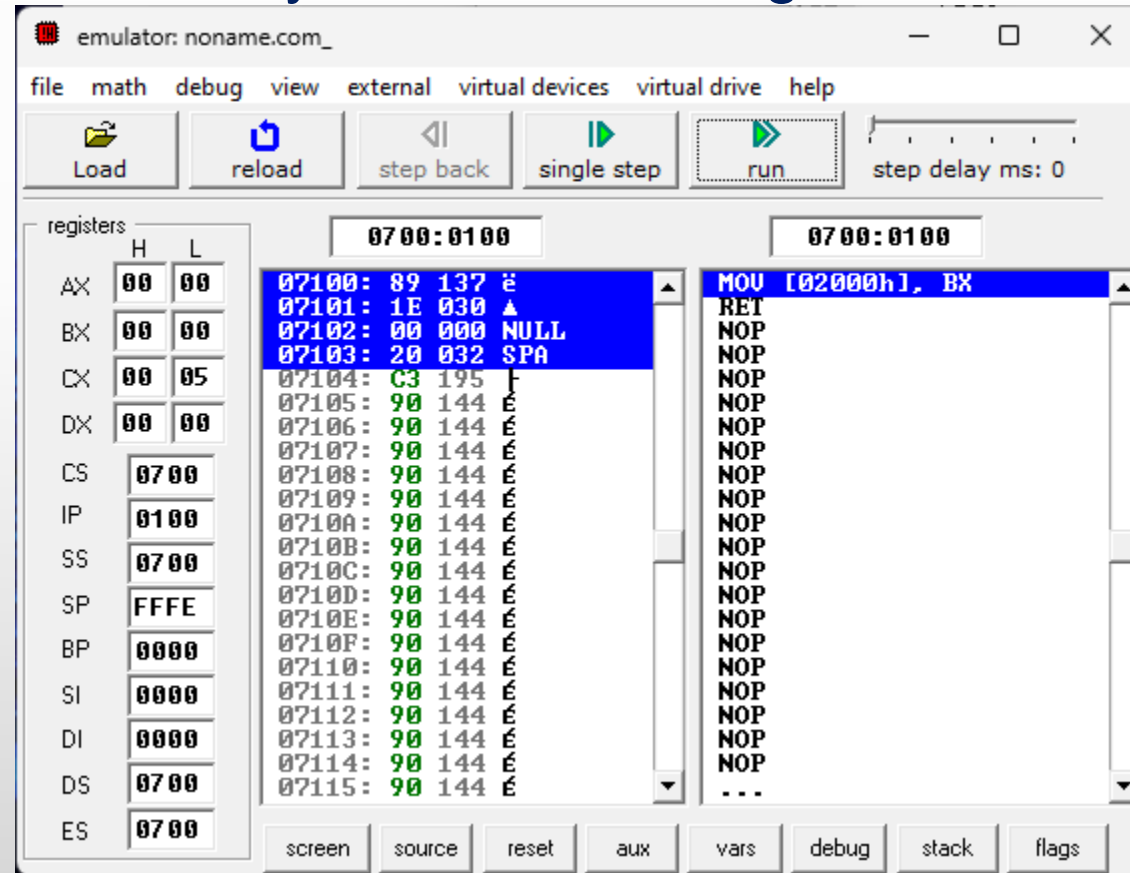
```
MOV AX, [01000h]
RET
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

At the bottom of the window, there are buttons for screen, source, reset, aux, vars, debug, stack, and flags.



# Yazmaçtaki Değeri Belleğe Taşıma

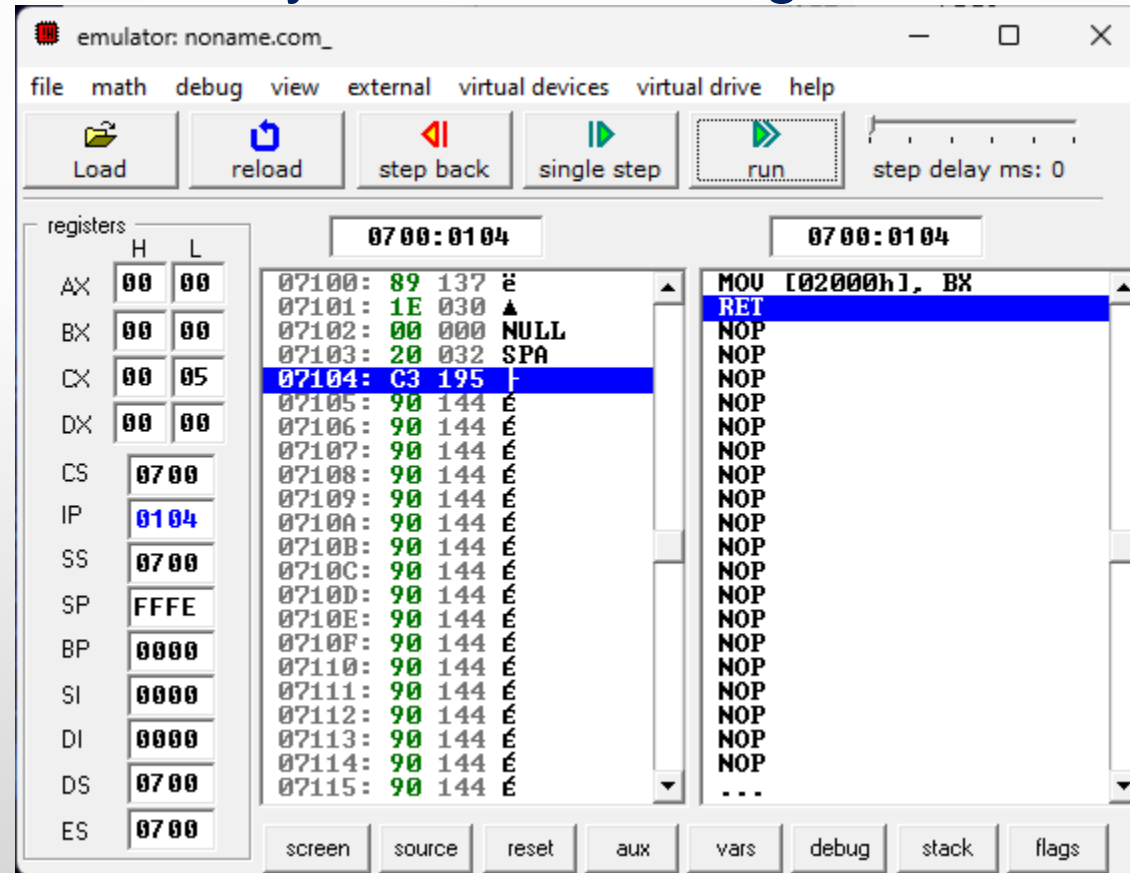
- MOV [0x2000], BX ; BX yazmacındaki değeri bellek adresine taşır





# 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

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

	H	L
AX	00	00
BX	00	00
CX	00	09
DX	00	00
CS	07 00	
IP	01 00	
SS	07 00	
SP	FF FE	
BP	00 00	
SI	00 00	
DI	00 00	
DS	07 00	
ES	07 00	

07 00: 01 00

```
07100: B9 185 |  
07101: 0A 010 NEWL  
07102: 00 000 NULL  
07103: BA 186 |  
07104: 14 020 |  
07105: 00 000 NULL  
07106: 87 135 |  
07107: CA 202 |  
07108: C3 195 |  
07109: 90 144 É  
0710A: 90 144 É  
0710B: 90 144 É  
0710C: 90 144 É  
0710D: 90 144 É  
0710E: 90 144 É  
0710F: 90 144 É  
07110: 90 144 É  
07111: 90 144 É  
07112: 90 144 É  
07113: 90 144 É  
07114: 90 144 É  
07115: 90 144 É
```

07 00: 01 00

```
MOU CX, 0000Ah  
MOU DX, 00014h  
XCHG DX, CX  
RET  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
...
```

screen source reset aux vars debug stack flags



# İki Yazmaç Değerini Takas

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

	H	L
AX	00	00
BX	00	00
CX	00	0A
DX	00	00
CS	07 00	
IP	01 03	
SS	07 00	
SP	FF FE	
BP	00 00	
SI	00 00	
DI	00 00	
DS	07 00	
ES	07 00	

07 00: 01 03

07100:	B9	185	
07101:	0A	010	NEWL
07102:	00	000	NULL
07103:	BA	186	
07104:	14	020	
07105:	00	000	NULL
07106:	87	135	
07107:	CA	202	
07108:	C3	195	
07109:	90	144	É
0710A:	90	144	É
0710B:	90	144	É
0710C:	90	144	É
0710D:	90	144	É
0710E:	90	144	É
0710F:	90	144	É
07110:	90	144	É
07111:	90	144	É
07112:	90	144	É
07113:	90	144	É
07114:	90	144	É
07115:	90	144	É

07 00: 01 03

```
MOU CX, 0000Ah
MOU DX, 00014h
XCHG DX, CX
RET
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags



# İki Yazmaç Değerini Takas

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

	H	L
AX	00	00
BX	00	00
CX	00	0A
DX	00	14
CS	07 00	
IP	01 06	
SS	07 00	
SP	FF FE	
BP	00 00	
SI	00 00	
DI	00 00	
DS	07 00	
ES	07 00	

07 00: 01 06

07100:	B9	185	
07101:	0A	010	NEWL
07102:	00	000	NULL
07103:	BA	186	
07104:	14	020	
07105:	00	000	NULL
07106:	87	135	
07107:	CA	202	
07108:	C3	195	
07109:	90	144	É
0710A:	90	144	É
0710B:	90	144	É
0710C:	90	144	É
0710D:	90	144	É
0710E:	90	144	É
0710F:	90	144	É
07110:	90	144	É
07111:	90	144	É
07112:	90	144	É
07113:	90	144	É
07114:	90	144	É
07115:	90	144	É

07 00: 01 06

```
MOU CX, 0000Ah
MOU DX, 00014h
XCHG DX, CX
RET
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags



# İki Yazmaç Değerini Takas

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

	H	L
AX	00	00
BX	00	00
CX	00	14
DX	00	0A
CS	07 00	
IP	01 08	
SS	07 00	
SP	FF FE	
BP	00 00	
SI	00 00	
DI	00 00	
DS	07 00	
ES	07 00	

07 00: 01 08

```
07100: B9 185 |  
07101: 0A 010 NEWL  
07102: 00 000 NULL  
07103: BA 186 |  
07104: 14 020 |  
07105: 00 000 NULL  
07106: 87 135 |  
07107: CA 202 |  
07108: C3 195 |  
07109: 90 144 É  
0710A: 90 144 É  
0710B: 90 144 É  
0710C: 90 144 É  
0710D: 90 144 É  
0710E: 90 144 É  
0710F: 90 144 É  
07110: 90 144 É  
07111: 90 144 É  
07112: 90 144 É  
07113: 90 144 É  
07114: 90 144 É  
07115: 90 144 É
```

07 00: 01 08

```
MOU CX, 0000Ah  
MOU DX, 00014h  
XCHG DX, CX  
RET  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
...
```

screen source reset aux vars debug stack flags



# 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

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

	H	L
AX	00	00
BX	00	00
CX	00	06
DX	00	00
CS	07 00	
IP	01 00	
SS	07 00	
SP	FF FE	
BP	00 00	
SI	00 00	
DI	00 00	
DS	07 00	
ES	07 00	

0700:0100

```
07100: BE 190 ↓
07101: 00 000 NULL
07102: 30 048 0
07103: 8B 139 i
07104: FE 254 |
07105: C3 195 |
07106: 90 144 é
07107: 90 144 é
07108: 90 144 é
07109: 90 144 é
0710A: 90 144 é
0710B: 90 144 é
0710C: 90 144 é
0710D: 90 144 é
0710E: 90 144 é
0710F: 90 144 é
07110: 90 144 é
07111: 90 144 é
07112: 90 144 é
07113: 90 144 é
07114: 90 144 é
07115: 90 144 é
```

0700:0100

```
MOU SI, 03000h
MOU DI, SI
RET
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags



# Bir Yazmaca Başka Yazmacın İçeriğini Kopyalama

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

	H	L
AX	00	00
BX	00	00
CX	00	06
DX	00	00
CS	07 00	
IP	01 03	
SS	07 00	
SP	FF FE	
BP	00 00	
SI	3 000	
DI	00 00	
DS	07 00	
ES	07 00	

07 00: 01 03

```
07100: BE 190 j
07101: 00 000 NULL
07102: 30 048 0
07103: 8B 139 i
07104: FE 254 i
07105: C3 195 j
07106: 90 144 e
07107: 90 144 e
07108: 90 144 e
07109: 90 144 e
0710A: 90 144 e
0710B: 90 144 e
0710C: 90 144 e
0710D: 90 144 e
0710E: 90 144 e
0710F: 90 144 e
07110: 90 144 e
07111: 90 144 e
07112: 90 144 e
07113: 90 144 e
07114: 90 144 e
07115: 90 144 e
...
```

07 00: 01 03

```
MOU SI, 03000h
MOU DI, SI
RET
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags



# Bir Yazmaca Başka Yazmacın İçeriğini Kopyalama

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

	H	L
AX	00	00
BX	00	00
CX	00	06
DX	00	00
CS	07 00	
IP	01 05	
SS	07 00	
SP	FF FE	
BP	00 00	
SI	30 00	
DI	30 00	
DS	07 00	
ES	07 00	

07 00: 01 05

07100:	BE	190	J
07101:	00	000	NULL
07102:	30	048	0
07103:	8B	139	i
07104:	FE	254	l
07105:	C3	195	
07106:	90	144	E
07107:	90	144	E
07108:	90	144	E
07109:	90	144	E
0710A:	90	144	E
0710B:	90	144	E
0710C:	90	144	E
0710D:	90	144	E
0710E:	90	144	E
0710F:	90	144	E
07110:	90	144	E
07111:	90	144	E
07112:	90	144	E
07113:	90	144	E
07114:	90	144	E
07115:	90	144	E

07 00: 01 05

```
MOU SI, 03000h
MOU DI, SI
RET
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags





# Bir Bellek Adresine Sabit Bir Deęeri Yazma

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

The screenshot shows an emulator window titled "emulator: noname.com\_". The interface 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, run, and a step delay slider set to 0 ms. On the left, a "registers" panel displays the state of various registers:

Register	H	L
AX	00	00
BX	00	00
CX	00	06
DX	00	00
CS	07 00	
IP	01 00	
SS	07 00	
SP	FF FE	
BP	00 00	
SI	00 00	
DI	00 00	
DS	07 00	
ES	07 00	

The main window displays assembly code in two columns. The left column shows memory addresses and hex values, and the right column shows the corresponding assembly instructions. The instruction at address 07104 is highlighted in blue:

Address	Hex	Instruction
07100	C6 198	MOV b.[04000h], 041h
07101	06 006	RET
07102	00 000	NOP
07103	40 064	NOP
07104	41 065	NOP
07105	C3 195	NOP
07106	90 144	NOP
07107	90 144	NOP
07108	90 144	NOP
07109	90 144	NOP
0710A	90 144	NOP
0710B	90 144	NOP
0710C	90 144	NOP
0710D	90 144	NOP
0710E	90 144	NOP
0710F	90 144	NOP
07110	90 144	NOP
07111	90 144	NOP
07112	90 144	NOP
07113	90 144	NOP
07114	90 144	NOP
07115	90 144	NOP

At the bottom of the window, there are buttons for "screen", "source", "reset", "aux", "vars", "debug", "stack", and "flags".



# Bir Bellek Adresine Sabit Bir Deęeri Yazma

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

	H	L
AX	00	00
BX	00	00
CX	00	06
DX	00	00
CS	07 00	
IP	01 05	
SS	07 00	
SP	FF FE	
BP	00 00	
SI	00 00	
DI	00 00	
DS	07 00	
ES	07 00	

07 00: 01 05

07100:	C6	198	⌵
07101:	06	006	⬆
07102:	00	000	NULL
07103:	40	064	⌵
07104:	41	065	A
07105:	C3	195	⌵
07106:	90	144	E
07107:	90	144	E
07108:	90	144	E
07109:	90	144	E
0710A:	90	144	E
0710B:	90	144	E
0710C:	90	144	E
0710D:	90	144	E
0710E:	90	144	E
0710F:	90	144	E
07110:	90	144	E
07111:	90	144	E
07112:	90	144	E
07113:	90	144	E
07114:	90	144	E
07115:	90	144	E

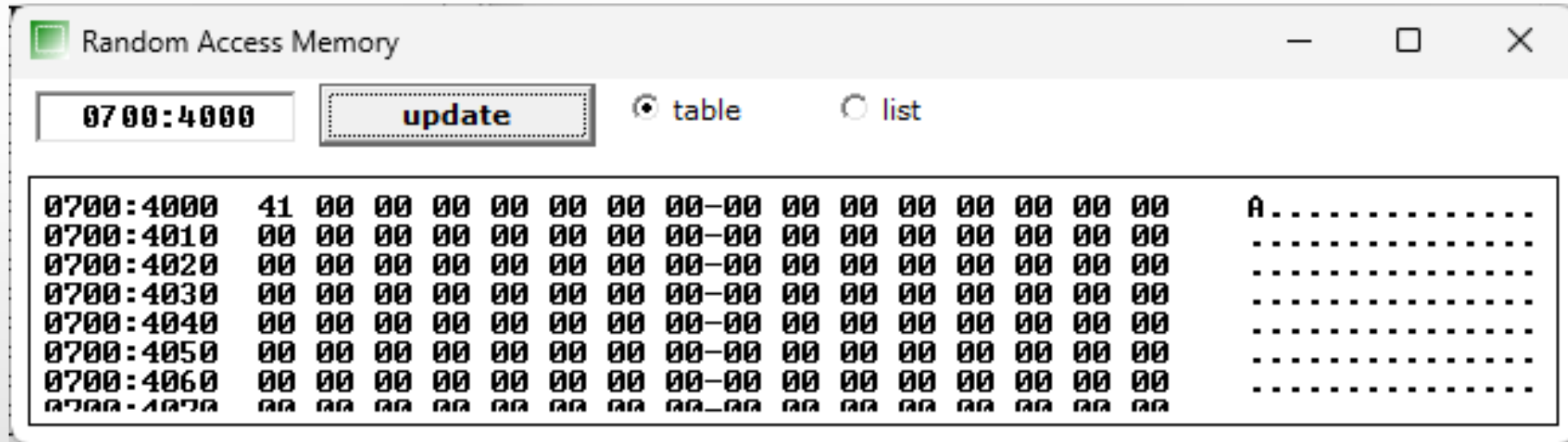
07 00: 01 05

```
MOU b.[04000h], 041h
RET
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags



# 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

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

	H	L
AX	00	00
BX	00	00
CX	00	06
DX	00	00
CS	07 00	
IP	01 00	
SS	07 00	
SP	FF FE	
BP	00 00	
SI	00 00	
DI	00 00	
DS	07 00	
ES	07 00	

0700:0100

07100:	B8	184	7
07101:	34	052	4
07102:	12	018	7
07103:	8A	138	E
07104:	D8	216	1
07105:	C3	195	1
07106:	90	144	E
07107:	90	144	E
07108:	90	144	E
07109:	90	144	E
0710A:	90	144	E
0710B:	90	144	E
0710C:	90	144	E
0710D:	90	144	E
0710E:	90	144	E
0710F:	90	144	E
07110:	90	144	E
07111:	90	144	E
07112:	90	144	E
07113:	90	144	E
07114:	90	144	E
07115:	90	144	E

0700:0100

```
MOU AX, 01234h
MOU BL, AL
RET
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags



# Bir Yazmacın Alt Kısmını Diğerine Kopyalama

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

	H	L
AX	12	34
BX	00	00
CX	00	06
DX	00	00
CS	07 00	
IP	01 03	
SS	07 00	
SP	FF FE	
BP	00 00	
SI	00 00	
DI	00 00	
DS	07 00	
ES	07 00	

0700:0103

07100:	B8	184	7
07101:	34	052	4
07102:	12	018	4
07103:	8A	138	è
07104:	D8	216	÷
07105:	C3	195	†
07106:	90	144	é
07107:	90	144	é
07108:	90	144	é
07109:	90	144	é
0710A:	90	144	é
0710B:	90	144	é
0710C:	90	144	é
0710D:	90	144	é
0710E:	90	144	é
0710F:	90	144	é
07110:	90	144	é
07111:	90	144	é
07112:	90	144	é
07113:	90	144	é
07114:	90	144	é
07115:	90	144	é

0700:0103

```
MOU AX, 01234h
MOU BL, AL
RET
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags



# Bir Yazmacın Alt Kısmını Diğerine Kopyalama

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

	H	L
AX	12	34
BX	00	34
CX	00	06
DX	00	00
CS	07 00	
IP	01 05	
SS	07 00	
SP	FF FE	
BP	00 00	
SI	00 00	
DI	00 00	
DS	07 00	
ES	07 00	

07 00: 01 05

07100:	B8	184	3
07101:	34	052	4
07102:	12	018	5
07103:	8A	138	6
07104:	D8	216	7
07105:	C3	195	8
07106:	90	144	9
07107:	90	144	A
07108:	90	144	B
07109:	90	144	C
0710A:	90	144	D
0710B:	90	144	E
0710C:	90	144	F
0710D:	90	144	10
0710E:	90	144	11
0710F:	90	144	12
07110:	90	144	13
07111:	90	144	14
07112:	90	144	15
07113:	90	144	16
07114:	90	144	17
07115:	90	144	18

MOU AX, 01234h  
MOU BL, AL  
RET  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
...

screen source reset aux vars debug stack flags



# Bir Yazmacın Alt Kısımına Ekleme

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





# Bir Yazmacın Alt Kismına Ekleme

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

	H	L
AX	00	00
BX	00	00
CX	00	07
DX	00	00
CS	07 00	
IP	01 00	
SS	07 00	
SP	FF FE	
BP	00 00	
SI	00 00	
DI	00 00	
DS	07 00	
ES	07 00	

07 00: 01 00

07100:	B8	184	3
07101:	FF	255	RES
07102:	00	000	NULL
07103:	80	128	ç
07104:	C4	196	-
07105:	01	001	0
07106:	C3	195	†
07107:	90	144	É
07108:	90	144	É
07109:	90	144	É
0710A:	90	144	É
0710B:	90	144	É
0710C:	90	144	É
0710D:	90	144	É
0710E:	90	144	É
0710F:	90	144	É
07110:	90	144	É
07111:	90	144	É
07112:	90	144	É
07113:	90	144	É
07114:	90	144	É
07115:	90	144	É

MOU AX, 000FFh  
ADD AH, 01h  
RET  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
...

screen source reset aux vars debug stack flags



# Bir Yazmacın Alt Kısımına Ekleme

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

	H	L
AX	00	FF
BX	00	00
CX	00	07
DX	00	00
CS	07 00	
IP	01 03	
SS	07 00	
SP	FF FE	
BP	00 00	
SI	00 00	
DI	00 00	
DS	07 00	
ES	07 00	

0700:0103

07100:	B8	184	?
07101:	FF	255	RES
07102:	00	000	NULL
07103:	80	128	?
07104:	C4	196	-
07105:	01	001	?
07106:	C3	195	?
07107:	90	144	E
07108:	90	144	E
07109:	90	144	E
0710A:	90	144	E
0710B:	90	144	E
0710C:	90	144	E
0710D:	90	144	E
0710E:	90	144	E
0710F:	90	144	E
07110:	90	144	E
07111:	90	144	E
07112:	90	144	E
07113:	90	144	E
07114:	90	144	E
07115:	90	144	E

0700:0103

```
MOU AX, 000FFh
ADD AH, 01h
RET
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags



# Bir Yazmacın Alt Kismına Ekleme

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

	H	L
AX	01	FF
BX	00	00
CX	00	07
DX	00	00
CS	07 00	
IP	01 06	
SS	07 00	
SP	FFFE	
BP	0000	
SI	0000	
DI	0000	
DS	07 00	
ES	07 00	

0700:0106

Address	Hex	Dec	Symbol
07100:	B8	184	ı
07101:	FF	255	RES
07102:	00	000	NULL
07103:	80	128	Ç
07104:	C4	196	—
07105:	01	001	0
07106:	C3	195	ı
07107:	90	144	É
07108:	90	144	É
07109:	90	144	É
0710A:	90	144	É
0710B:	90	144	É
0710C:	90	144	É
0710D:	90	144	É
0710E:	90	144	É
0710F:	90	144	É
07110:	90	144	É
07111:	90	144	É
07112:	90	144	É
07113:	90	144	É
07114:	90	144	É
07115:	90	144	É

0700:0106

```
MOV AX, 000FFh
ADD AH, 01h
RET
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags



# İkilik, Sekizlik ve Onaltılık Değer Atama

```
; 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

emulator: 5\_sample.com\_

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	00	05
BX	00	0A
CX	00	0A
DX	00	00
CS	07 00	
IP	01 04	
SS	07 00	
SP	FF FE	
BP	00 00	
SI	00 00	
DI	00 00	
DS	07 00	
ES	07 00	

07 00: 01 04

07100:	B0	176	⏏
07101:	05	005	⏏
07102:	B3	179	
07103:	0A	010	NEWL
07104:	B1	177	⏏
07105:	08	008	BACK
07106:	02	002	⏏
07107:	C3	195	
07108:	2A	042	*
07109:	C1	193	
0710A:	90	144	É
0710B:	90	144	É
0710C:	90	144	É
0710D:	90	144	É
0710E:	90	144	É
0710F:	90	144	É
07110:	90	144	É
07111:	90	144	É
07112:	90	144	É
07113:	90	144	É
07114:	90	144	É
07115:	90	144	É

07 00: 01 04

```
MOV AL, 05h
MOV BL, 0Ah
MOV CL, 08h
ADD AL, BL
SUB AL, CL
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags



# İkilik, Sekizlik ve Onaltılık Değer Atama

emulator: 5\_sample.com\_

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	00	05
BX	00	0A
CX	00	08
DX	00	00
CS	07 00	
IP	01 06	
SS	07 00	
SP	FF FE	
BP	00 00	
SI	00 00	
DI	00 00	
DS	07 00	
ES	07 00	

07 00: 01 06

07100:	B0	176	⏏
07101:	05	005	⏏
07102:	B3	179	
07103:	0A	010	NEWL
07104:	B1	177	⏏
07105:	08	008	BACK
07106:	02	002	⏏
07107:	C3	195	
07108:	2A	042	*
07109:	C1	193	⏏
0710A:	90	144	É
0710B:	90	144	É
0710C:	90	144	É
0710D:	90	144	É
0710E:	90	144	É
0710F:	90	144	É
07110:	90	144	É
07111:	90	144	É
07112:	90	144	É
07113:	90	144	É
07114:	90	144	É
07115:	90	144	É

MOV AL, 05h  
MOV BL, 0Ah  
MOV CL, 08h  
ADD AL, BL  
SUB AL, CL  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
...

screen source reset aux vars debug stack flags





# İkilik, Sekizlik ve Onaltılık Değer Atama

emulator: 5\_sample.com\_

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	00	0F
BX	00	0A
CX	00	08
DX	00	00
CS	07 00	
IP	01 08	
SS	07 00	
SP	FF FE	
BP	00 00	
SI	00 00	
DI	00 00	
DS	07 00	
ES	07 00	

07 00: 01 08

07100:	B0	176	⏏
07101:	05	005	⏏
07102:	B3	179	
07103:	0A	010	NEWL
07104:	B1	177	⏏
07105:	08	008	BACK
07106:	02	002	⏏
07107:	C3	195	
07108:	2A	042	*
07109:	C1	193	⏏
0710A:	90	144	É
0710B:	90	144	É
0710C:	90	144	É
0710D:	90	144	É
0710E:	90	144	É
0710F:	90	144	É
07110:	90	144	É
07111:	90	144	É
07112:	90	144	É
07113:	90	144	É
07114:	90	144	É
07115:	90	144	É

07 00: 01 08

```
MOV AL, 05h
MOV BL, 0Ah
MOV CL, 08h
ADD AL, BL
SUB AL, CL
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags



# İkilik, Sekizlik ve Onaltılık Değer Atama

emulator: 5\_sample.com\_

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	00	07
BX	00	0A
CX	00	08
DX	00	00
CS	07 00	
IP	01 0A	
SS	07 00	
SP	FF FE	
BP	00 00	
SI	00 00	
DI	00 00	
DS	07 00	
ES	07 00	

07 00: 01 0A

07100:	B0	176	⏏
07101:	05	005	⏏
07102:	B3	179	
07103:	0A	010	NEWL
07104:	B1	177	⏏
07105:	08	008	BACK
07106:	02	002	⏏
07107:	C3	195	
07108:	2A	042	*
07109:	C1	193	
0710A:	90	144	É
0710B:	90	144	É
0710C:	90	144	É
0710D:	90	144	É
0710E:	90	144	É
0710F:	90	144	É
07110:	90	144	É
07111:	90	144	É
07112:	90	144	É
07113:	90	144	É
07114:	90	144	É
07115:	90	144	É

MOV AL, 05h  
MOV BL, 0Ah  
MOV CL, 08h  
ADD AL, BL  
SUB AL, CL  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
...

screen source reset aux vars debug stack flags



SON