

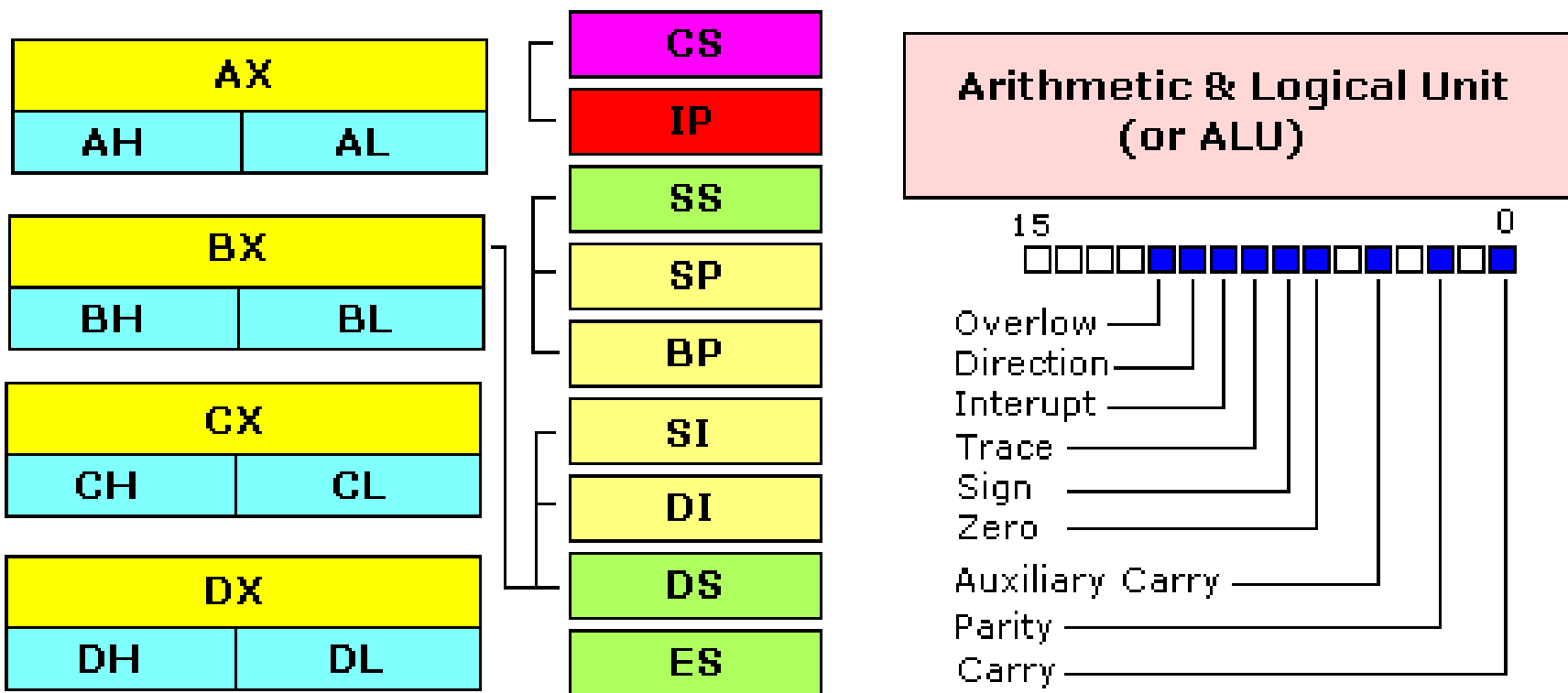
MİKROBİLGİSAYAR SİSTEMLERİ



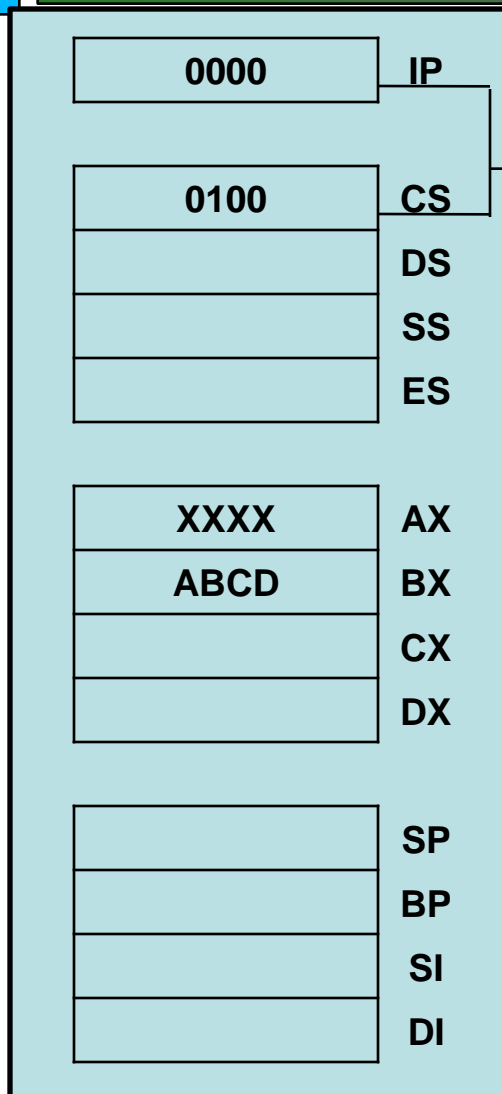
Teknik Bilimler Meslek Yüksekokulu

Register Adresleme Modlari

Central Processing Unit (or CPU)

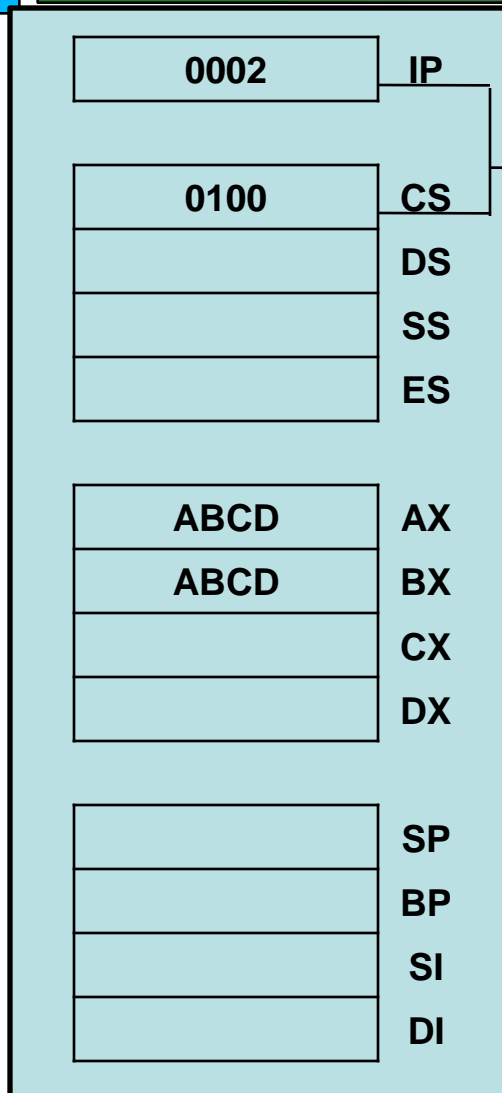


Register Adresleme



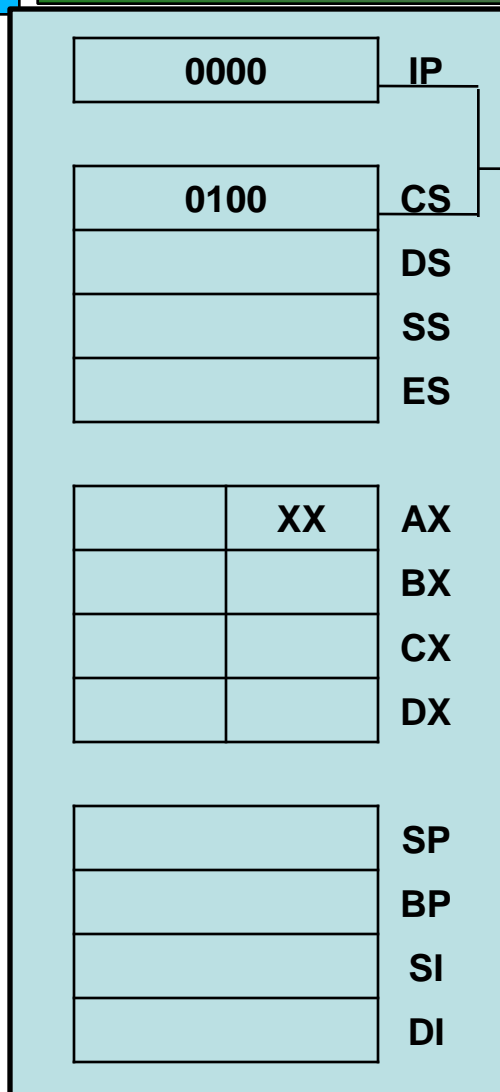
Adres	Bellek	İşlem
→ 01000	8B	MOV AX,BX
01001	C3	
01002	XX	Sonraki İş.

Register Adresleme



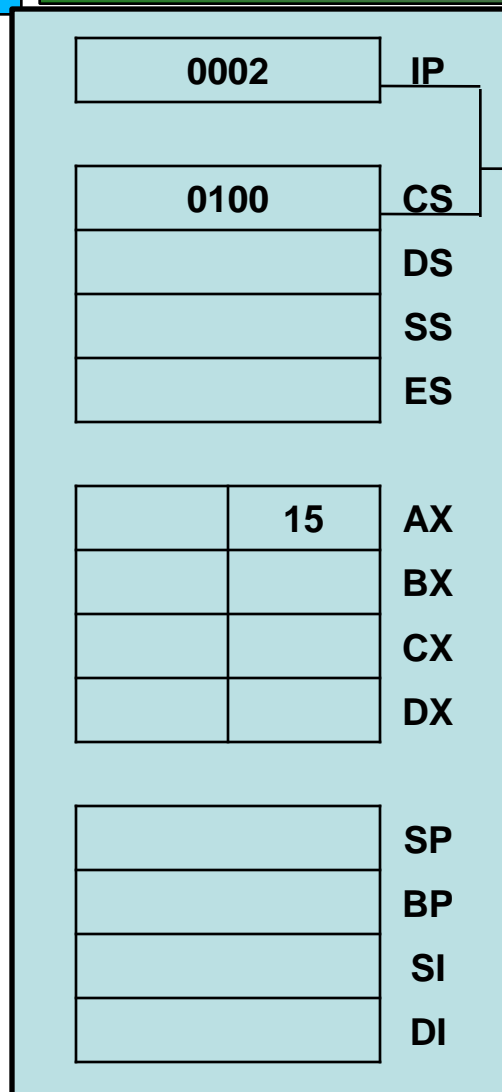
Adres	Bellek	İşlem
→ 01000	8B	MOV AX,BX
01001	C3	
01002	XX	Sonraki İş.

Anlık (Immediate) Adresleme



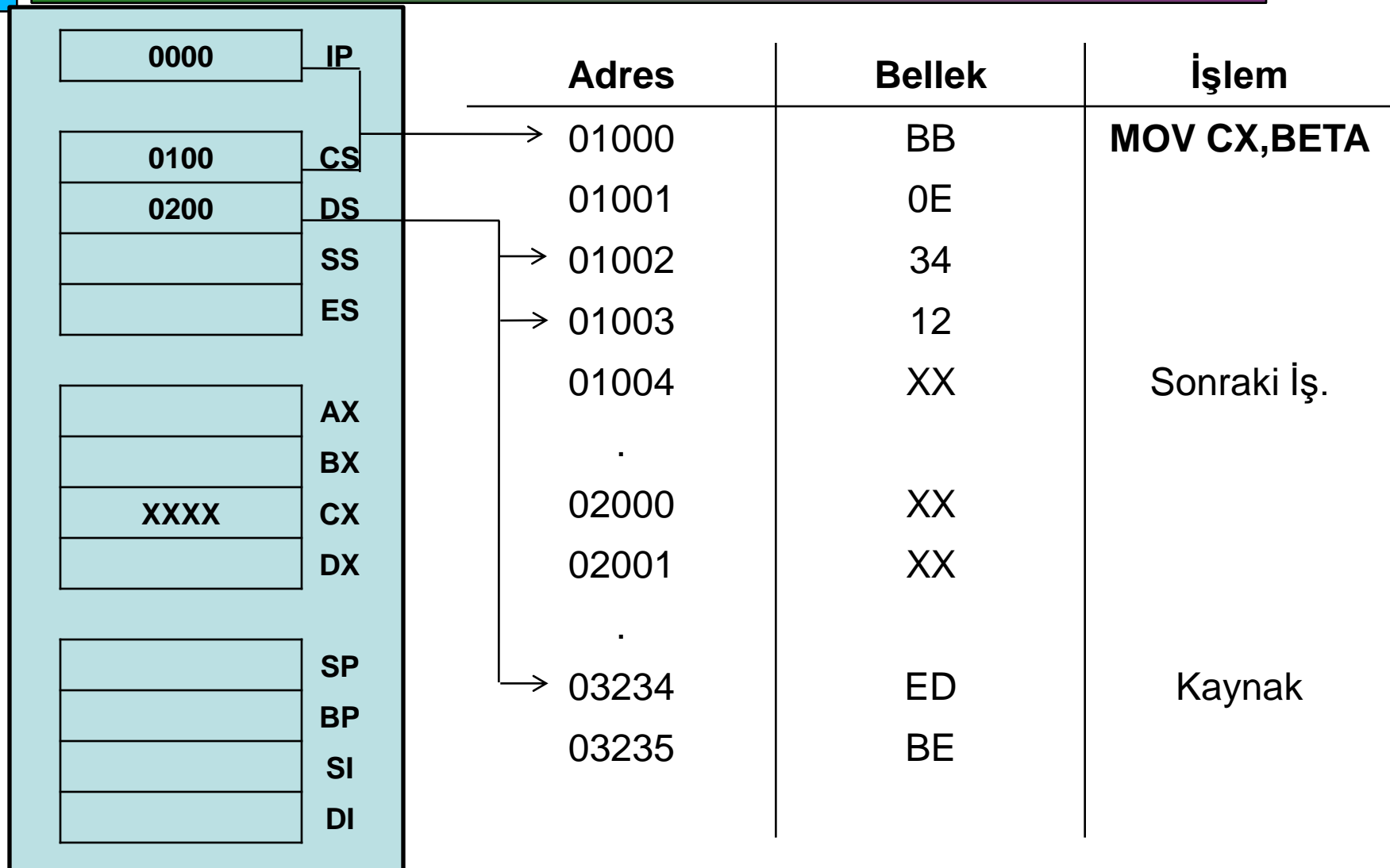
Adres	Bellek	İşlem
→ 01000	B0	MOV AL,15H
01001	15	
01002	XX	Sonraki İş.
01003	XX	

Anlık (Immediate) Adresleme

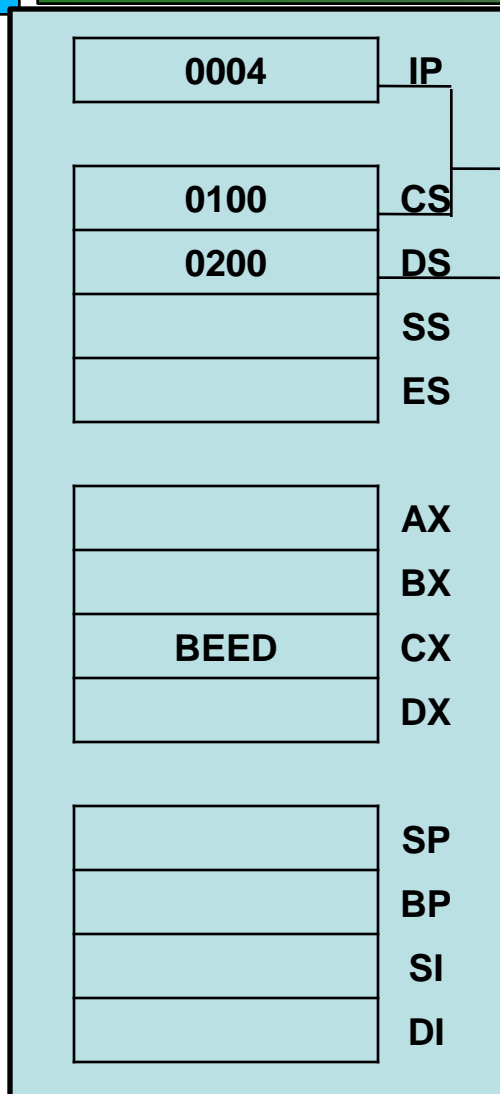


Adres	Bellek	İşlem
→ 01000	B0	MOV AL,15H
01001	15	
01002	XX	Sonraki İş.
01003	XX	

Direkt Adresleme

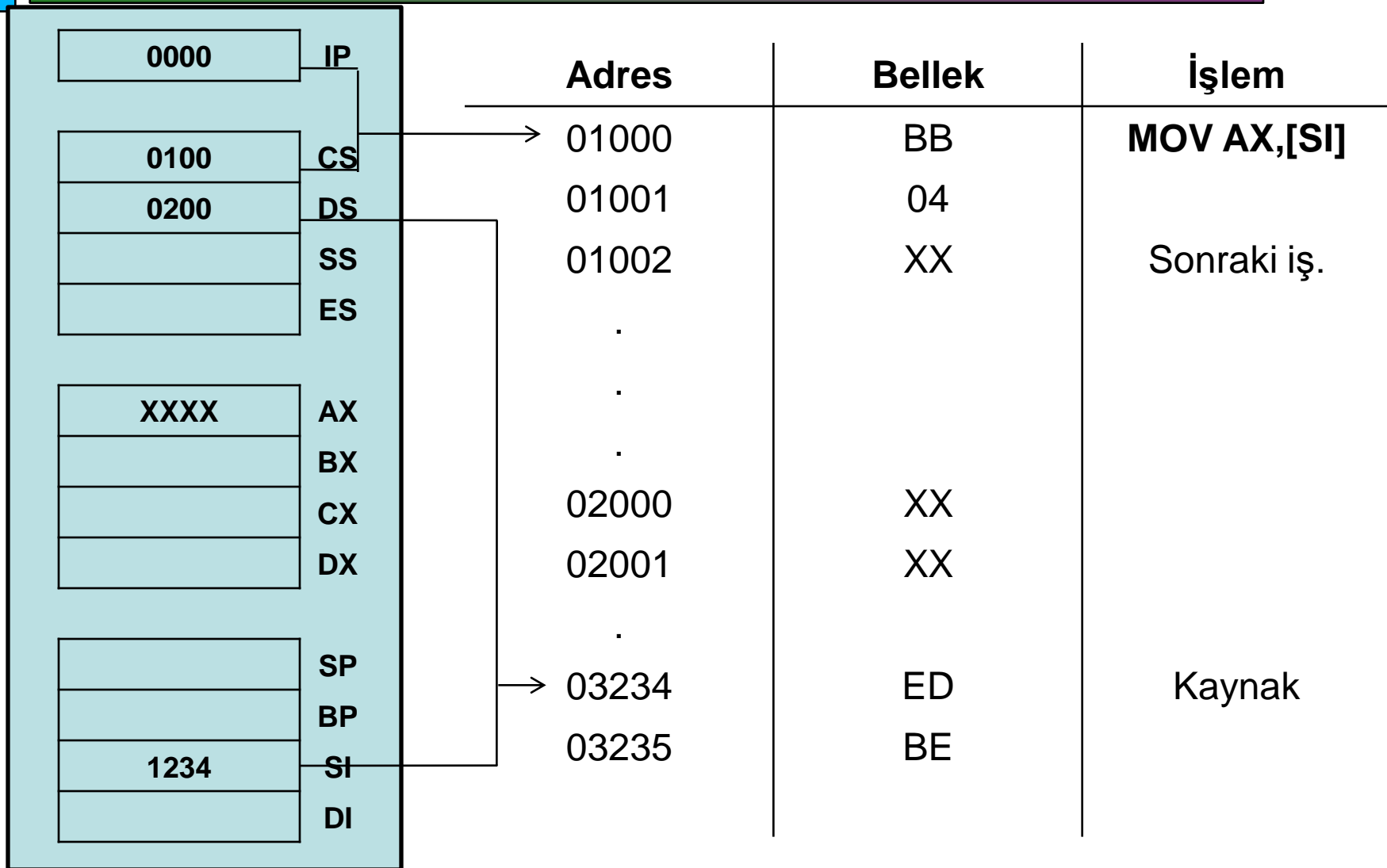


Direkt Adresleme

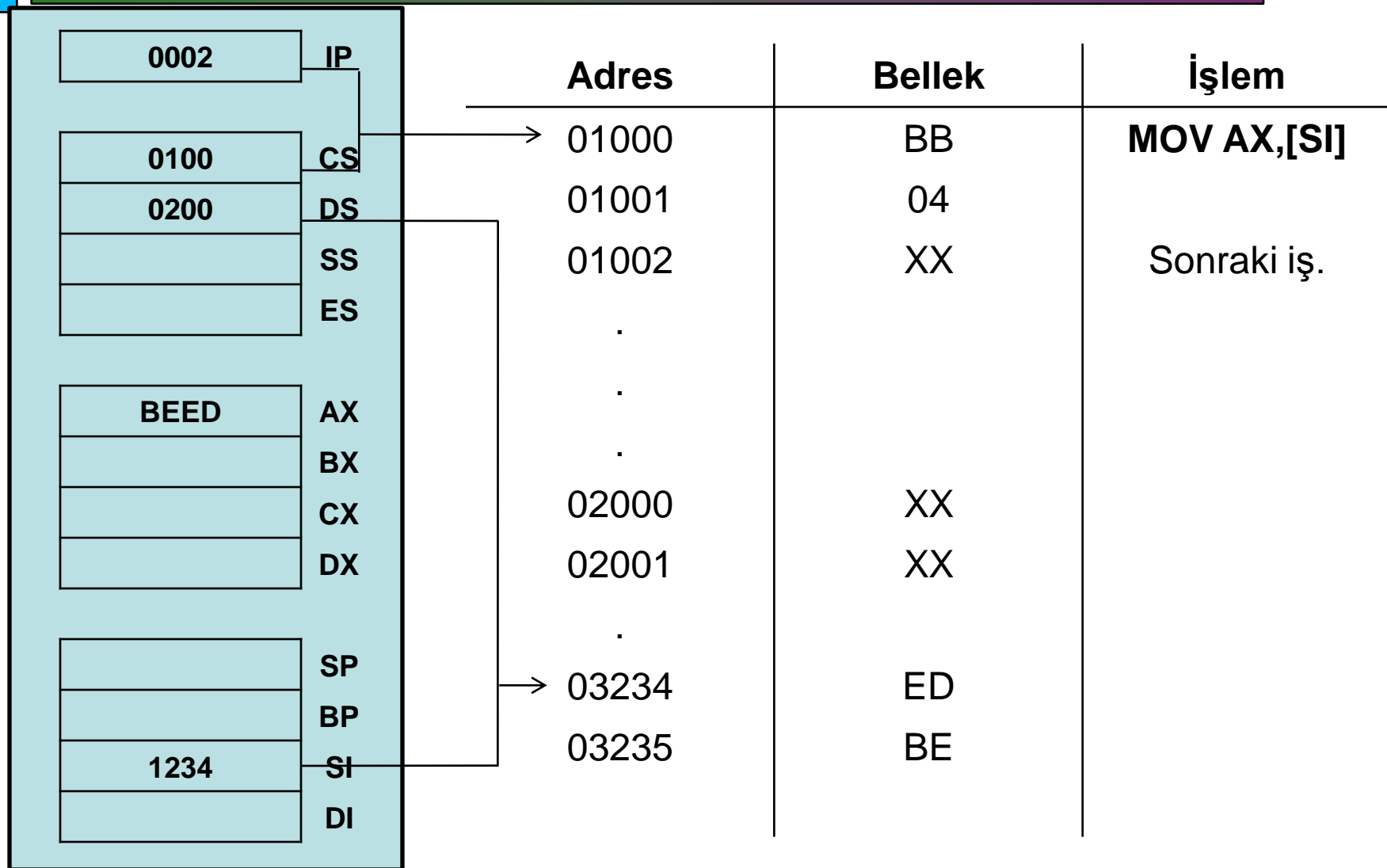


Adres	Bellek	İşlem
→ 01000	BB	MOV CX,BETA
01001	0E	
→ 01002	34	Sonraki İş.
→ 01003	12	
01004	XX	
.		
02000	XX	
02001	XX	
.		
→ 03234	ED	
03235	BE	

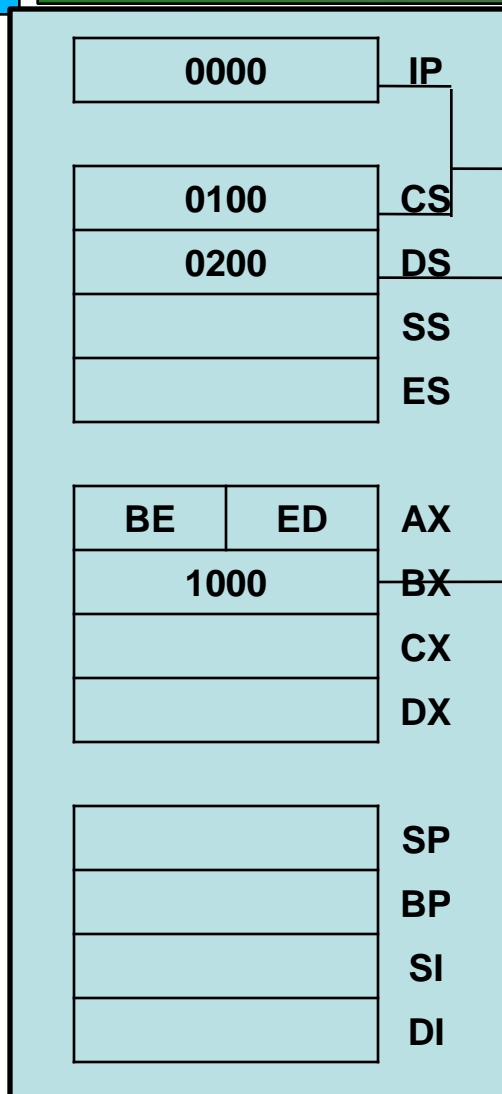
Register İndirekt Adresleme



Register İndirekt Adresleme

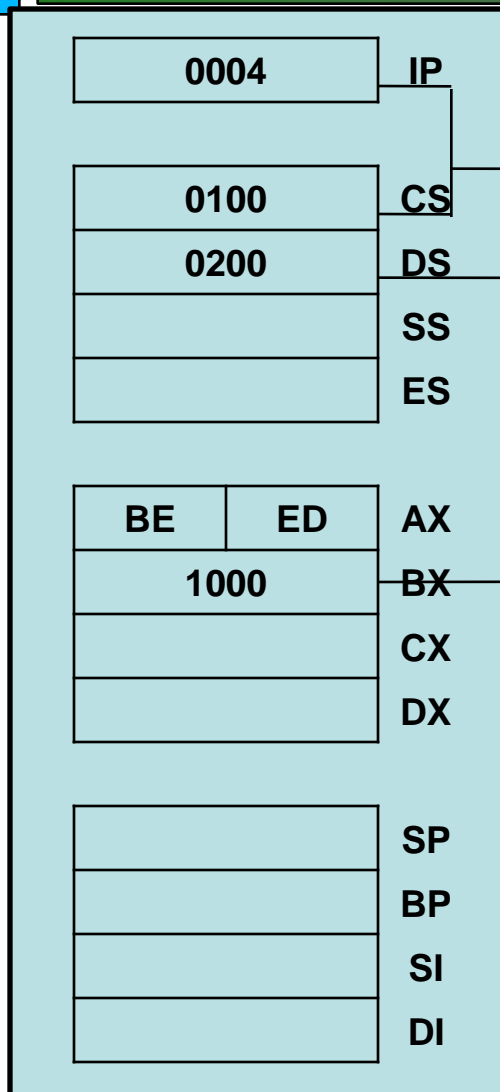


Base (BX) İndex Adresleme



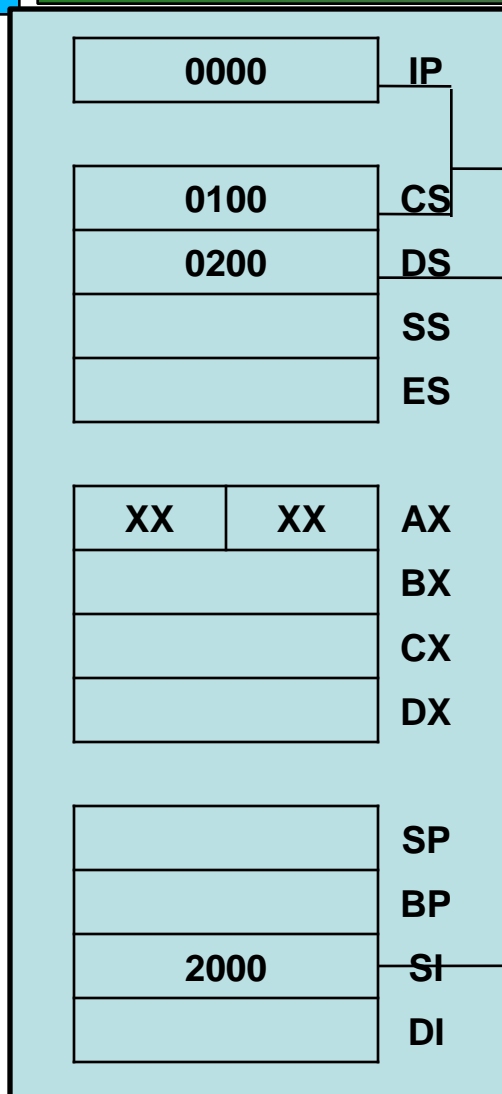
Adres	Bellek	İşlem
→ 01000	BB	MOV [BX]+BETA,AL
01001	07	
→ 01002	34	Sonraki İş.
→ 01003	12	
01004	XX	
.		
02000	XX	
02001	XX	
.		
→ 04234	XX	Hedef
04235	XX	

Base (BX) İndex Adresleme



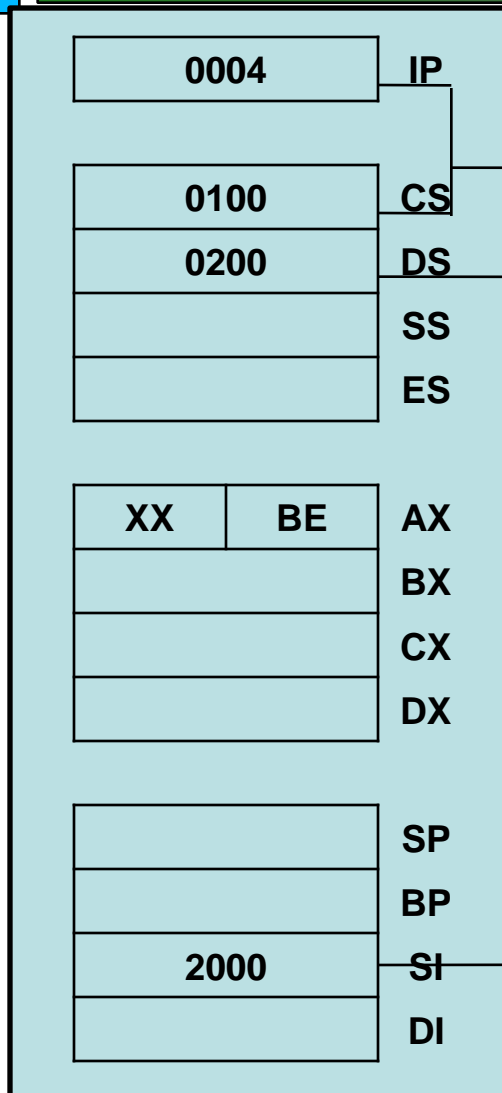
Adres	Bellek	İşlem
→ 01000	BB	MOV [BX]+BETA,AL
01001	07	
→ 01002	34	Sonraki İş.
→ 01003	12	
01004	XX	
.		
→ 02000	XX	
02001	XX	
.		
→ 04234	ED	
04235	XX	

İndexli Adresleme



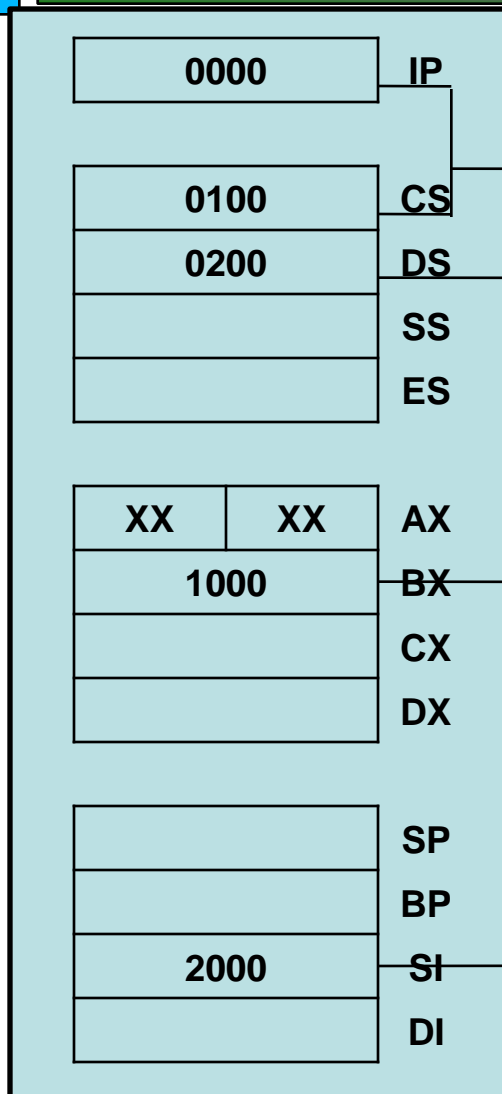
Adres	Bellek	İşlem
→ 01000	BA	MOV AL,[SI]+ARRAY
01001	44	
→ 01002	34	
→ 01003	12	
01004	XX	Sonraki İş.
.		
02000	XX	
02001	XX	
.		
→ 05234	BE	Kaynak

İndexli Adresleme



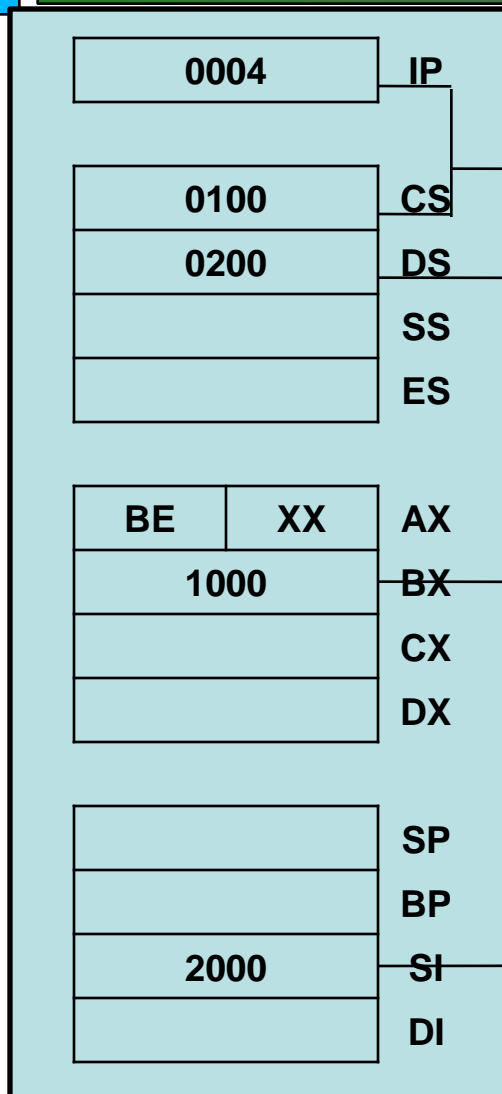
Adres	Bellek	İşlem
→ 01000	BA	MOV AL,[SI]+ARRAY
01001	44	
→ 01002	34	Sonraki İş.
→ 01003	12	
01004	XX	
.		
02000	XX	
02001	XX	
.		
→ 05234	BE	

Base (BX) İndexli Adresleme



Adres	Bellek	İşlem
→01000	BA	MOV AH,[BX]+[SI]+BETA
01001	20	
→01002	34	
→01003	12	
01004	XX	
.		Sonraki İş.
02000	XX	
02001	XX	
.		
→06234	BE	Kaynak

Base (BX) İndexli Adresleme



Adres	Bellek	İşlem
→ 01000	BA	MOV AH,[BX]+[SI]+BETA
01001	20	
→ 01002	34	
→ 01003	12	
01004	XX	Sonraki İş.
.		
02000	XX	
02001	XX	
.		
→ 06234	BE	Kaynak

MS DEBUG KOMUTLARI



```
Komut İstemi - DEBUG
C:\Users\ilker>DEBUG
-R
AX=0000  BX=0000  CX=0000  DX=0000  SP=FFEE  BP=0000  SI=0000  DI=0000
DS=0B3A  ES=0B3A  SS=0B3A  CS=0B3A  IP=0100  NU UP EI PL NZ NA PO NC
0B3A:0100 B80000          MOV     AX,0000
```

- R (Register) : Bu komut o andaki Register değerlerini ve bayrakların durumunu belirlemekte kullanılır.

MS DEBUG KOMUTLARI

cmd Komut İstemi - DEBUG

```
C:\Users\ilker>DEBUG
```

```
-R
```

```
AX=0000 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
```

```
DS=0B3A ES=0B3A SS=0B3A CS=0B3A IP=0100  NU UP EI PL NZ NA PO NC
```

```
0B3A:0100 B80000          MOV     AX,0000
```

```
-R AX
```

```
AX 0000
```

```
:1234
```

```
-R
```

```
AX=1234 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
```

```
DS=0B3A ES=0B3A SS=0B3A CS=0B3A IP=0100  NU UP EI PL NZ NA PO NC
```

```
0B3A:0100 B80000          MOV     AX,0000
```

```
-
```

```
-
```

MS DEBUG KOMUTLARI

Flag İsmi	Set(1)	Clear (0)
Overflow - OF	OV (overflow)	NV (not overflow)
Direction - DF	DN (decrement - down)	UP (increment - up)
Interrupt - IF	EI (enable)	DI (disable)
Sign - SF	NG (negative)	PL (plus)
Zero - ZF	ZR (zero)	NZ (not zero)
Auxiliary Carry- AF	AC (auxiliary carry)	NA (not auxiliary)
Parity - PF	PE (even parity)	PO (odd parity)
Carry - CF	CY (carry yes)	NC (not carry)

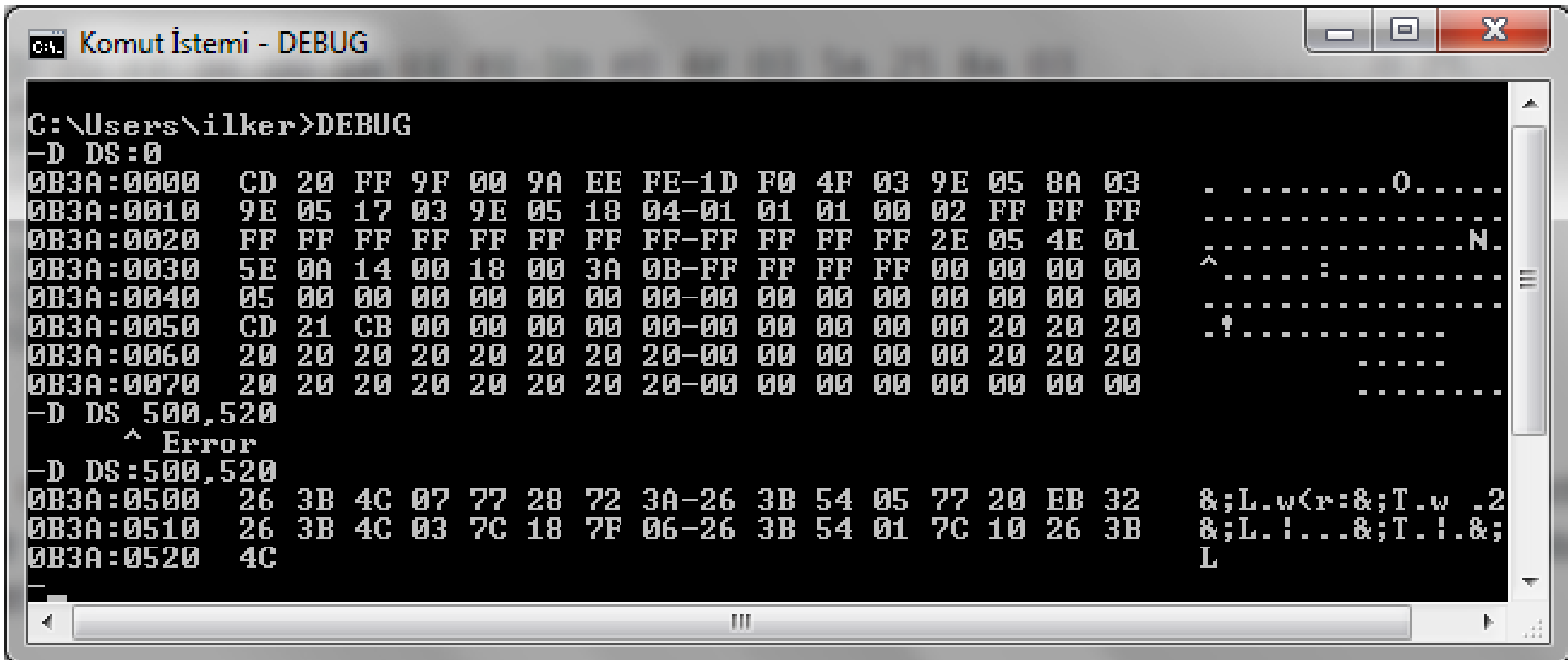
Komut İstemi - DEBUG

```
C:\Users\ilker>DEBUG
```

```
-R F
NU UP EI PL NZ NA PO NC  --
```

MS DEBUG KOMUTLARI

- D (Dump / Display): Bu komut ile belleğin istediğimiz segment' inin istediğimiz offset 'inden itibaren bilgileri görebiliriz.



```
C:\Users\ilker>DEBUG
-D DS:0
0B3A:0000  CD 20 FF 9F 00 9A EE FE-1D F0 4F 03 9E 05 8A 03  . . . . .0. . . .
0B3A:0010  9E 05 17 03 9E 05 18 04-01 01 01 00 02 FF FF FF  . . . . .
0B3A:0020  FF FF FF FF FF FF FF FF-FF FF FF FF 2E 05 4E 01  . . . . .N.
0B3A:0030  5E 0A 14 00 18 00 3A 0B-FF FF FF FF 00 00 00 00  ^ . . . . .
0B3A:0040  05 00 00 00 00 00 00 00-00 00 00 00 00 00 00  . . . . .
0B3A:0050  CD 21 CB 00 00 00 00 00-00 00 00 00 00 20 20 20  - ? . . . . .
0B3A:0060  20 20 20 20 20 20 20 20-00 00 00 00 00 20 20 20  . . . . .
0B3A:0070  20 20 20 20 20 20 20 20-00 00 00 00 00 00 00  . . . . .
-D DS 500,520
^
Error
-D DS:500,520
0B3A:0500  26 3B 4C 07 77 28 72 3A-26 3B 54 05 77 20 EB 32  &;L.w<r=&;T.w .2
0B3A:0510  26 3B 4C 03 7C 18 7F 06-26 3B 54 01 7C 10 26 3B  &;L.!...&;T.!.&;
0B3A:0520  4C                                     L
```

MS DEBUG KOMUTLARI

- A (Assembly) : Bu komut Debug yardımıyla basit programlar yazmayı sağlar. Debug içinde yazılan programlar COM uzantılı programlarda olduğu gibi 100H offset adresinden baslarlar. Bu neden ile program çalıştırılacağı zaman mutlaka IP register'ının değerinin R komutu ile 100H e getirilmesi gerekmektedir.

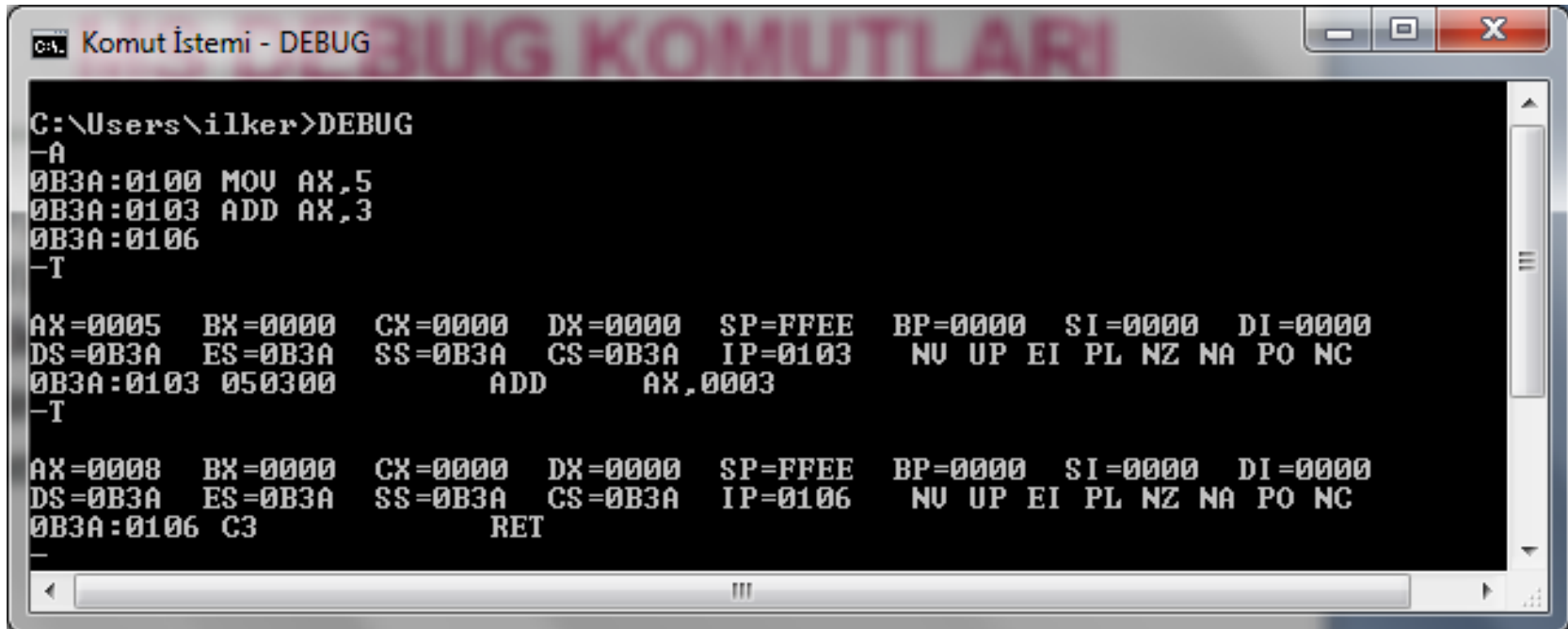
C:\> Komut İstemi - DEBUG

C:\Users\ilker>DEBUG

```
-A
0B3A:0100 MOV AX,5
0B3A:0103 ADD AX,3_
```

MS DEBUG KOMUTLARI

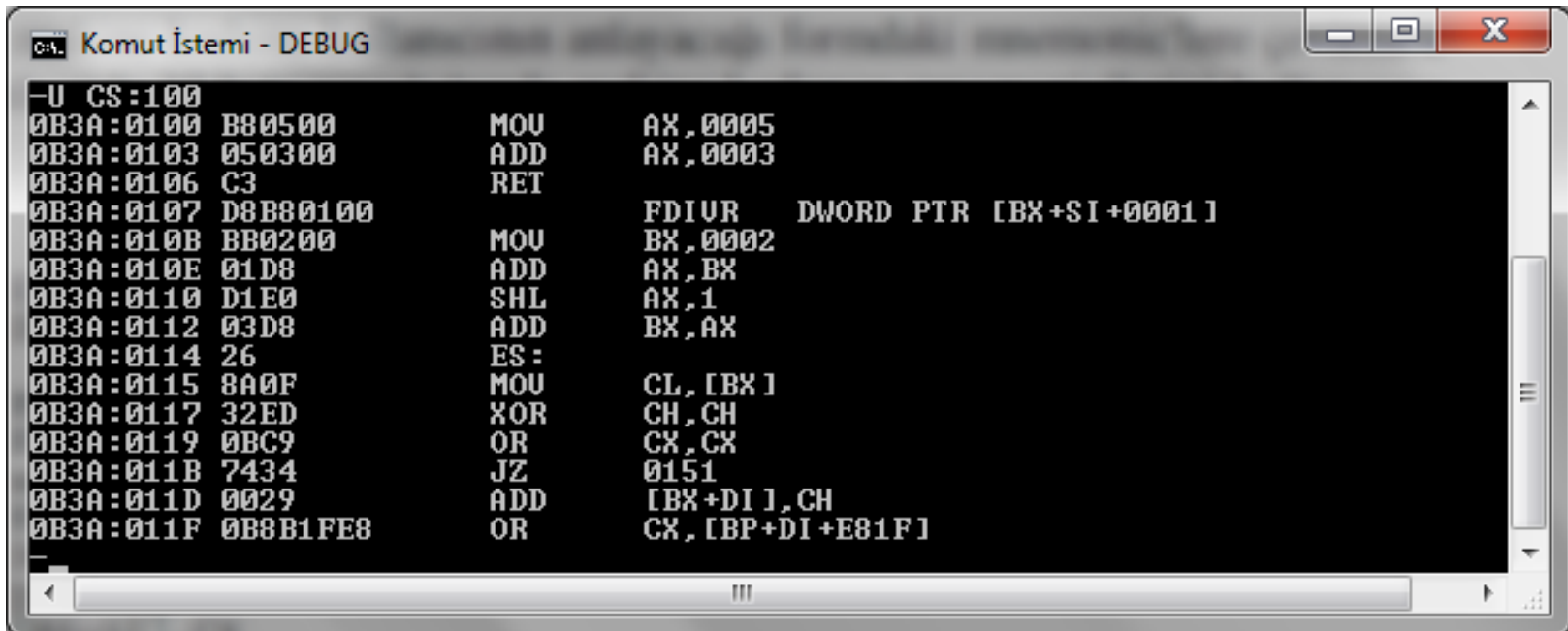
- T (Trace) : Yazılmış olan assembly komutları teker teker çalıştırılarak işlem sonunda Register ve bayrakların aldıkları değerleri ekranda gösterir. T komutu çalışmaya IP'nin gösterdiği noktadan itibaren baslar ve her komut islendikten sonra durur.



```
C:\Users\ilker>DEBUG
-A
0B3A:0100 MOV AX,5
0B3A:0103 ADD AX,3
0B3A:0106
-T
AX=0005  BX=0000  CX=0000  DX=0000  SP=FFEE  BP=0000  SI=0000  DI=0000
DS=0B3A  ES=0B3A  SS=0B3A  CS=0B3A  IP=0103  NU UP EI PL NZ NA PO NC
0B3A:0103 050300      ADD     AX,0003
-T
AX=0008  BX=0000  CX=0000  DX=0000  SP=FFEE  BP=0000  SI=0000  DI=0000
DS=0B3A  ES=0B3A  SS=0B3A  CS=0B3A  IP=0106  NU UP EI PL NZ NA PO NC
0B3A:0106 C3          RET
```

MS DEBUG KOMUTLARI

- U (Unassembly) : CS'de yazılı olan makine kodunu kullanıcının anlayacağı formdaki mnemonic'lere çevirmek için kullanılan bir komuttur. Her çağırılışında 32 byte uzunluğunda makine kodunun mnemonic'lerini kullanıcıya gösterir.



```

C:\> Komut İstemi - DEBUG
-U CS:100
0B3A:0100 B80500      MOV     AX,0005
0B3A:0103 050300      ADD     AX,0003
0B3A:0106 C3          RET
0B3A:0107 D8B80100    FDIUR  DWORD PTR [BX+SI+0001]
0B3A:010B BB0200      MOV     BX,0002
0B3A:010E 01D8      ADD     AX,BX
0B3A:0110 D1E0      SHL     AX,1
0B3A:0112 03D8      ADD     BX,AX
0B3A:0114 26       ES:
0B3A:0115 8A0F      MOV     CL,[BX]
0B3A:0117 32ED      XOR     CH,CH
0B3A:0119 0BC9      OR     CX,CX
0B3A:011B 7434      JZ     0151
0B3A:011D 0029      ADD     [BX+DI],CH
0B3A:011F 0B8B1FE8 OR     CX,[BP+DI+E81F]
```

MS DEBUG KOMUTLARI

E (Edit) : Bellekte herhangi bir yerdeki deęerleri deęiřtirmek amacıyla kullanılan bir komuttur.



```
C:\Users\ilker>DEBUG
-E DS:20
0B3A:0020  FF.CC
-D DS:20,21
0B3A:0020  CC FF
..
```


MS DEBUG KOMUTLARI

- **F (Fill)** : Bellek başlangıç ve bitiş adresleri belirlenen bir alanı belirlenen bir sayı ile doldurmak amacıyla kullanılır.



```
C:\Users\ilker>DEBUG
C:\Users\ilker>DEBUG
-F DS:100,120,AA
-D DS:100,120
0B3A:0100  AA AA AA AA AA AA AA AA-AA AA AA AA AA AA AA
0B3A:0110  AA AA AA AA AA AA AA AA-AA AA AA AA AA AA AA
0B3A:0120  AA
-
```