BİL 362 Mikroişlemciler

Hazırlayan: M.Ali Akcayol
Gazi Üniversitesi
Bilgisayar Mühendisliği Bölümü

Konular

Adresleme Modları

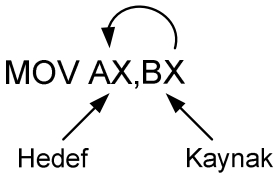
- Data adresleme modları
  - Register adresleme
  - Immediate adresleme
  - Direct adresleme
  - Register indirect adresleme
  - Base-plus-index adresleme
  - Register relative adresleme
  - Base relative-plus-index adresleme
  - Scaled-index adresleme
- Program memory adresleme modları
  - Direct program memory adresleme
  - Relative program memory adresleme
  - Indirect program memory adresleme
- Stack memory adresleme modları

Adresleme Modları

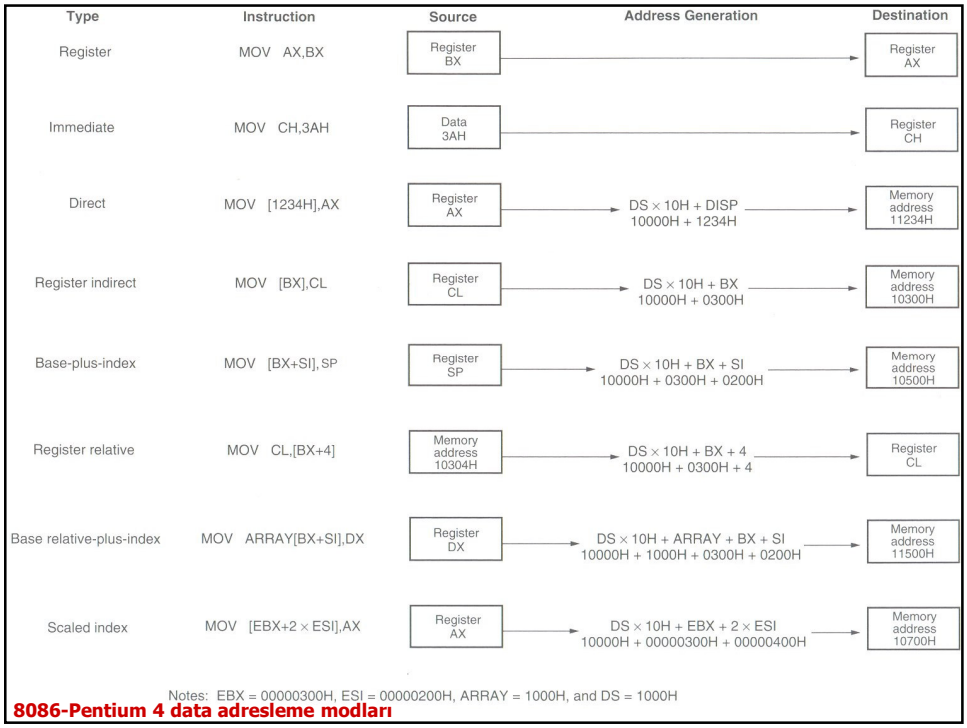
- Etkin yazılım geliştirmek için mikroişlemcideki herbir instruction'da kullanılabilecek adresleme modları bilinmelidir.
- Bu kısımdaki tüm adresleme modları için esnek olmasından dolayı MOV komutu örnek olarak kullanılacaktır.
- MOV komutu register-register veya register-memory arasında 8086-80286 işlemciler için byte veya word 80386 ve üstü işlemciler için byte, word veya doubleword data transfer eder.
- Program memory adresleme için örneklerde CALL ve JUMP komutları kullanılarak programın akışının nasıl değiştirildiği gösterilmiştir.
- Stack memory adresleme için PUSH ve POP komutları kullanılmıştır.

Data Adresleme Modları

- MOV komutunun genel yapısı aşağıdaki gibidir.



- Tüm komutlarda en sağdaki kaynak, solundaki hedef ve en soldaki ise opcode'dur.
- Memory-memory bilgi aktarımına izin verilmemektedir.(MOVS hariç)
- MOV komutu yukarıda bir word bilgi aktarımı yapar. BX register'inin içeriğini AX registerine aktarır.
- BX registerinin içeriği değişmez sadece kopyalama yapılır.
- Hedef registerin (AX) içeriği değişir. (CMP ve TEST gibi komutlarda hedef değişmez)



Type	Instruction	Source	Address Generation	Destination
Register	MOV AX,BX	Register BX		Register AX
Immediate	MOV CH,3AH	Data 3AH		Register CH
Direct	MOV [1234H],AX	Register AX	DS × 10H + DISP 10000H + 1234H	Memory address 11234H
Register indirect	MOV [BX],CL	Register CL	DS × 10H + BX 10000H + 0300H	Memory address 10300H
Base-plus-index	MOV [BX+SI], SP	Register SP	DS × 10H + BX + SI 10000H + 0300H + 0200H	Memory address 10500H
Register relative	MOV CL,[BX+4]	Memory address 10304H	DS × 10H + BX + 4 10000H + 0300H + 4	Register CL
Base relative-plus-index	MOV ARRAY[BX+SI],DX	Register DX	DS × 10H + ARRAY + BX + SI 10000H + 1000H + 0300H + 0200H	Memory address 11500H
Scaled index	MOV [EBX+2 × ESI],AX	Register AX	DS × 10H + EBX + 2 × ESI 10000H + 00000300H + 00000400H	Memory address 10700H

Notes: EBX = 00000300H, ESI = 00000200H, ARRAY = 1000H, and DS = 1000H

8086-Pentium 4 data adresleme modları

Data Adresleme Modları

- Register adresleme
  - Bir byte veya word datayı kaynak register veya memory'den hedef register veya memory'ye aktarır.
  - **MOV CX,DX** komutu word (16-bit) boyutundaki DX içeriğini CX register'ına aktarır.
  - 80386 ve üstü işlemcilerde doubleword (32-bit) boyutundaki veri aktarılabilir.
  - **MOV ECX,EDX** komutu doubleword EDX içeriğini ECX register'ına aktarır.

Data Adresleme Modları

- Immediate adresleme
  - Bir byte veya word datayı hedef register veya memory'ye aktarır.
  - **MOV AL,22H** komutu byte boyutundaki 22H değerini AL register'ına aktarır.
  - 80386 ve üstü işlemcilerde doubleword (32-bit) boyutundaki data aktarılabilir.
  - **MOV EBX,12345678H** komutu doubleword 12345678H değerini EBX register'ına kopyalar.

Data Adresleme Modları

- Direct adresleme
  - Bir byte veya word datayı register ile memory arasında aktarır.
  - Komut kümesi memory-to-memory aktarımına izin vermemektedir. (MOVS hariç)
  - **MOV CX,LIST** komutu hafızada LIST adresindeki word boyutundaki içeriği CX register'ına aktarır.
  - 80386 ve üstü işlemcilerde doubleword (32-bit) boyutundaki data aktarılabilir.
  - **MOV ESI,LIST** komutu hafızada LIST adresindeki doubleword boyutundaki içeriği ESI register'ına aktarır.

Data Adresleme Modları

- Register Indirect adresleme
  - Bir byte veya word datayı register ile index register veya base register tarafından adreslenen memory alanı arasında aktarır.
  - Index veya base register'lar SI, DI, BP veya BX olabilir.
  - **MOV AX,[BX]** komutu data segment içerisinde BX offset adresindeki bir word datayı AX register'ına aktarır.
  - 80386 ve üstü işlemcilerde doubleword (32-bit) boyutundaki data aktarılabilir. Adresleme için EAX, EBX, ECX, EDX, EBP, EDI veya ESI kullanılabilir.
  - **MOV AL,[ECX]** komutu data segment içerisinde ECX offset adresindeki bir byte datayı AL register'ına aktarır.

Data Adresleme Modları

- Base-plus-index adresleme
  - Bir byte veya word datayı register ile base register(BP,BX)+ index register (DI,SI) tarafından adreslenen memory alanı arasında aktarır.
  - **MOV [BX+DI],CL** komutu bir byte CL içeriğini data segment içerisinde BX+DI offset adresine aktarır.
  - 80386 ve üstü işlemcilerde EAX, EBX, ECX, EDX, EBP, EDI veya ESI register'ları birlikte kullanılarak adres belirlenebilir.
  - **MOV [EAX+EBX],CL** komutu bir byte CL içeriğini data segment içerisinde EAX+EBX offset adresine aktarır.

Data Adresleme Modları

- Register relative adresleme
  - Bir byte veya word datayı register ile base register(BP,BX) veya index register(DI,SI)+displacement(kayma) tarafından adreslenen memory alanı arasında aktarır.
  - **MOV AX,[BX+4]** komutu AX register'ına data segment içerisinde BX+4 offset adresinin içeriğini aktarır.
  - **MOV AX,ARRAY[BX]** komutu AX register'ına data segment içerisinde ARRAY+BX offset adresinin içeriğini aktarır.
  - 80386 ve üstü işlemcilerde herhangi bir register adresleme için kullanılabilir.
  - **MOV AX,[ECX+4]** veya **MOV AX,ARRAY[EBX]** şeklinde kullanılabilir.

Data Adresleme Modları

- Base relative-plus-index adresleme
  - Bir byte veya word datayı register ile base register(BP,BX)+index register(DI,SI)+displacement(kayma) tarafından adreslenen memory alanı arasında aktarır.
  - **MOV AX,ARRAY[BX+DI]** komutu AX register'ına data segment içerisinde ARRAY+BX+DI offset adresinin içeriğini aktarır.
  - **MOV AX,[BX+DI+4]** komutu AX register'ına data segment içerisinde BX+DI+4 offset adresinin içeriğini aktarır.
  - 80386 ve üstü işlemcilerde herhangi bir register adresleme için kullanılabilir.
  - **MOV EAX,ARRAY[EBX+ECX]** komutu EAX register'ına data segment içerisinde ARRAY+EBX+ECX offset adresinin içeriğini aktarır.

Data Adresleme Modları

- Scaled-index adresleme
  - 80386 ve üstü işlemcilerde vardır.
  - Offset adres için kullanılan ikinci register bir ölçek faktörüyle (scale factor) 2x, 4x, 8x değiştirilir.
  - **MOV EDX,[EAX+4*EBX]** komutu EDX register'ına data segment içerisinde EAX+4*EBX offset adresinin içeriğini aktarır.
  - Ölçeklendirme word (2x), doubleword (4x) veya quadword (8x) içeriğe ulaşmayı sağlar.
  - **MOV EAX,ARRAY[EBX+ECX]** komutu EAX register'ına data segment içerisinde ARRAY+EBX+ECX offset adresinin içeriğini aktarır.

Data Adresleme Modları

- Register adresleme - detay
  - En yaygın kullanılan data adresleme şeklidir. Register adlarının bilinmesi yeterlidir.
  - 8-bit register adresleme için AH, AL, BH, BL, CH, CL, DH ve DL registerleri kullanılır.
  - 16-bit register adresleme için AX, BX, CX, DX, SP, BP, SI ve DI registerleri kullanılır.
  - 32-bit register adresleme için EAX, EBX, ECX, EDX, ESP, EBP, ESI ve EDI registerleri kullanılır.
  - Bazı MOV komutlarıyla, PUSH ve POP komutları CS, ES, DS, SS, FS ve GS register'larını kullanır.
  - Farklı boyutlardaki register'lar birlikte kullanılamaz.(MOV AX, AL veya MOV EAX, AL). SHL gibi bazı komutlarda farklı boyutlarda register kullanılabilir.
  - MOV bayrak (FLAG) bitlerini etkilemez.

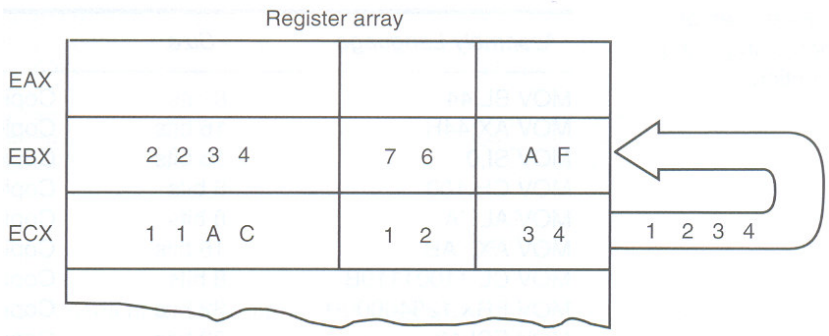
Data Adresleme Modları

- Register adresleme - detay
  - Register adresleme için örnek komutlar

Assembly Language	*Size*	*Operation*
MOV AL,BL	8 bits	Copies BL into AL
MOV CH,CL	8 bits	Copies CL into CH
MOV AX,CX	16 bits	Copies CX into AX
MOV SP,BP	16 bits	Copies BP into SP
MOV DS,AX	16 bits	Copies AX into DS
MOV SI,DI	16 bits	Copies DI into SI
MOV BX,ES	16 bits	Copies ES into BX
MOV ECX,EBX	32 bits	Copies EBX into ECX
MOV ESP,EDX	32 bits	Copies EDX into ESP
MOV DS,CX	16 bits	Copies CX into DS
MOV ES,DS	—	Not allowed (segment to segment)
MOV BL,DX	—	Not allowed (mixed sizes)
MOV CS,AX	—	Not allowed (the code segment register may not be the destination register)

Data Adresleme Modları

- Register adresleme - detay
  - MOV komutu sadece hedef register'ı değiştirir. (CMP ve TEXT komutları hedef register'ı değiştirmez.)
  - MOV BX, CX komutu 1234H değerini CX register'ından BX register'ına kopyalar.
  - EBX register'ının en soldaki 16-bit değişmez.



Data Adresleme Modları

- Register adresleme - detay
  - Aşağıdaki komutlar 8-bit, 16-bit veya 32-bit veri aktarır.
  - EBX register'ında en soldaki 16-bit değişmez.

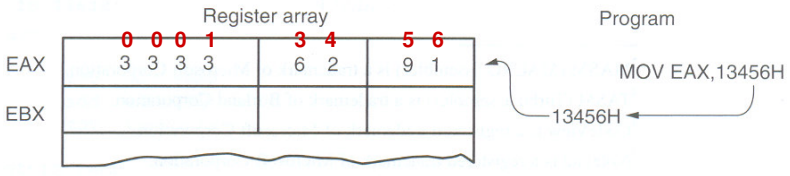
```
0000 8B C3          MOV AX,BX      ;copy contents of BX into AX
0002 8A CE          MOV CL,DH      ;copy contents of DH into CL
0004 8A CD          MOV CL,CH      ;copy contents of CH into CL
0006 66|8B C3       MOV EAX,EBX    ;copy contents of EBX into EAX
0009 66|8B D8       MOV EBX,EAX    ;copy contents of EAX into EBX
000C 66|8B C8       MOV ECX,EAX    ;copy contents of EAX into ECX
000F 66|8B D0       MOV EDX,EAX    ;copy contents of EAX into EDX
0012 8C C8          MOV AX,CS      ;copy CS into DS (two steps)
0014 8E D8          MOV DS,AX
0016 8E C8          MOV CS,AX      ;copy AX into CS (causes problems)
```

3 byte

2 byte

Data Adresleme Modları

- Immediate adresleme - detay
  - Data, opcode'un hemen ardından gelir ve sabittir.
  - Immediate adresleme 8- veya 16-bit data ile işlem yapar.
  - 80386 – Pentium 4 işlemcilerde doubleword (32-bit) data kullanılabilir.
  - Aşağıdaki komut 13456H değerini (opcode'dan hemen sonra hafızada yer alır) EAX register'ına kopyalar.



  - Bazı assembler'lar MOV AX, #13456H şeklinde immediate adresleme yapar. (# işareti kullanılır)
  - Intel ASM, Microsoft **MASM** ve Borland TASM # kullanmaz.

Data Adresleme Modları

- İmmediate adresleme - detay
  - Immediate adresleme için örnek komutlar

Assembly Language	*Size*	*Operation*
MOV BL,44	8 bits	Copies 44 decimal (2CH) into BL
MOV AX,44H	16 bits	Copies 0044H into AX
MOV SI,0	16 bits	Copies 0000H into SI
MOV CH,100	8 bits	Copies 100 decimal (64H) into CH
MOV AL, 'A'	8 bits	Copies ASCII A into AL
MOV AX, 'AB'	16 bits	Copies ASCII BA* into AX
MOV CL,11001110B	8 bits	Copies 11001110 binary into CL
MOV EBX,12340000H	32 bits	Copies 12340000H into EBX
MOV ESI,12	32 bits	Copies 12 decimal into ESI
MOV EAX,100B	32 bits	Copies 100 binary into EAX

*This is not an error. The ASCII characters are stored as BA, so exercise care when using word-sized pairs of ASCII characters.

Data Adresleme Modları

- İmmediate adresleme - detay
  - .MODEL TINY deyimi assembler'ın programı tek code segment içerisine yerleştirmesini sağlar.
  - .CODE deyimi code segment başlangıcını gösterir.
  - .STARTUP deyimi programın başlangıç komutunu gösterir.
  - .EXIT deyimi programın sonlanıp DOS'a çıkmasını sağlar.
  - END deyimi program dosyasının sonunu gösterir.

```
                        .MODEL TINY             ;choose single segment model
0000                    .CODE                   ;start of code segment
                        .STARTUP                ;start of program
0100  B8 0000           MOV AX,0                ;place 0000H into AX
0103  BB 0000           MOV BX,0                ;place 0000H into BX
0106  B9 0000           MOV CX,0                ;place 0000H into CX

0109  8B F0             MOV SI,AX               ;copy AX into SI
010B  8B F8             MOV DI,AX               ;copy AX into DI
010D  8B E8             MOV BP,AX               ;copy AX into BP

                        .EXIT                   ;exit to DOS
                        END                     ;end of program
```

MOV AX, 0 komutunun offset adresidir.

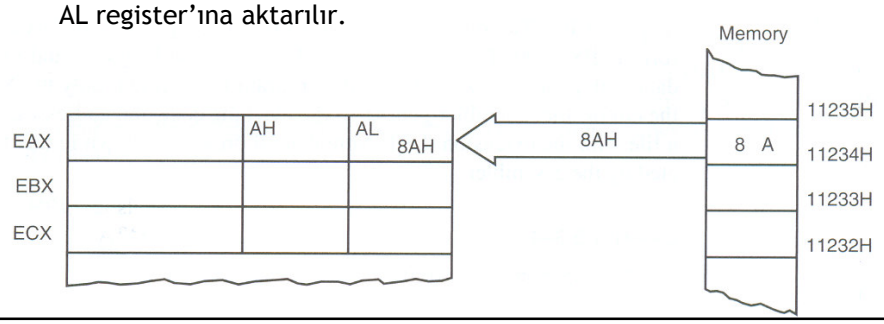
Data Adresleme Modları

- Immediate adresleme - detay
  - Her deyim 4 parçadan oluşur.
  - En solda label bulunur. Bulunduğu hafıza alanına sembolik isim atar.
  - İkinci alan opcode'dur.
  - Üçüncü alan/alanlar opcode'un sağındaki operand'lardır.
  - Son alan en sağdaki comment (açıklama) alanıdır. Açıklama her zaman ; ile başlar.

```
LABEL     OPCODE   OPERAND    COMMENT

DATA1     DB       23H        ;define DATA1 as a byte of 23H
DATA2     DW       1000H      ;define DATA2 as a word of 1000H

START:    MOV      AL,BL      ;copy BL into AL
          MOV      BH,AL      ;copy AL into BH
          MOV      CX,200     ;copy 200 into CX
```

Data Adresleme Modları

- Direct adresleme - detay
  - MOV komutu direct adreslemede data segment içindeki bir hafıza alanı ile bir register (AL, AX, EAX, …) arasında veri transfer eder.
  - MOV AL,DATA komutu data segment içinde DATA(örn.: 1234H) ile gösterilen hafıza alanından 8-bit'i AL ye aktarır.
  - Bazı assembler'larda MOV AL,DS:[1234H] veya MOV AL,DS:[DATA] şeklinde gösterilir.
  - DS = 1000H ise data segment içindeki 11234H fiziksel adresindeki data AL register'ına aktarılır.



Data Adresleme Modları

- Direct adresleme örnekleri

Assembly Language	Size	Operation
MOV AL,NUMBER	8 bits	Copies the byte contents of data segment memory location NUMBER into AL
MOV AX,COW	16 bits	Copies the word contents of data segment memory location COW into AX
MOV EAX,WATER	32 bits	Copies the doubleword contents of data segment location WATER into EAX
MOV NEWS,AL	8 bits	Copies AL into byte memory location NEWS
MOV THERE,AX	16 bits	Copies AX into word memory location THERE
MOV HOME,EAX	32 bits	Copies EAX into doubleword memory location HOME
MOV ES:[2000H],AL	8 bits	Copies AL into extra segment memory at offset address 2000H

Assembly Language	Size	Operation
MOV CH,DOG	8 bits	Copies the byte contents of data segment memory location DOG into CH
MOV CH,DS:[1000H]	8 bits	Copies the byte contents of data segment memory offset address 1000H into CH
MOV ES,DATA6	16 bits	Copies the word contents of data segment memory location DATA6 into ES
MOV DATA7,BP	16 bits	Copies BP into data segment memory location DATA7
MOV NUMBER,SP	16 bits	Copies SP into data segment memory location NUMBER
MOV DATA1,EAX	32 bits	Copies EAX into data segment memory location DATA1
MOV EDI,SUM1	32 bits	Copies the doubleword contents of data segment memory location SUM1 into EDI

Data Adresleme Modları

- Direct adresleme örnek program

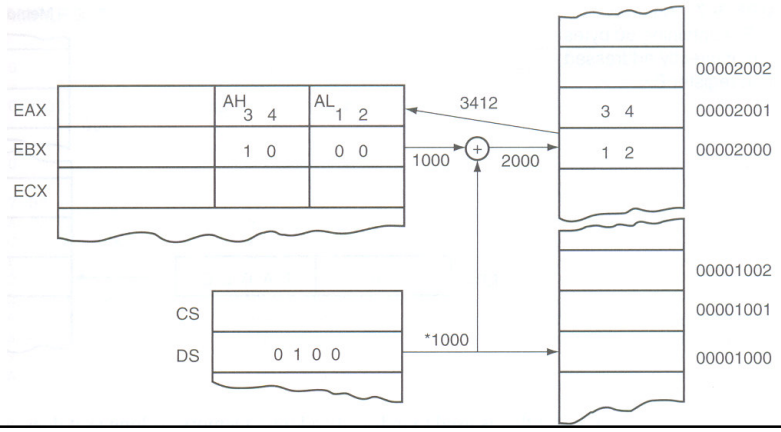
```
                        .MODEL SMALL        ;choose small model
0000                    .DATA               ;start data segment

0000 10         DATA1 DB  10H               ;place 10H into DATA1
0001 00         DATA2 DB  0                 ;place 00H into DATA2
0002 0000       DATA3 DW  0                 ;place 0000H into DATA3
0004 AAAA       DATA4 DW  0AAAAH            ;place AAAAH into DATA4

0000                    .CODE               ;start code segment
                        .STARTUP            ;start program

0017 A0 0000 R          MOV  AL,DATA1       ;copy DATA1 into AL
001A 8A 26 0001 R       MOV  AH,DATA2       ;copy DATA2 into AH
001E A3 0002 R          MOV  DATA3,AX       ;copy AX into DATA3
0021 8B 1E 0004 R       MOV  BX,DATA4       ;copy DATA4 into BX

                        .EXIT               ;exit to DOS
                        END                 ;end program listing
```

offset adres — makine kodu — etiket (label) — opcode — operand

.DATA data segment başlangıcını, SMALL bir data segment ve bir code segment kullanılacağını gösterir.

Data Adresleme Modları

- Register indirect adresleme - detay
  - Register indirect adresleme, data segment içinde bir register (BP, BX, SI, DI) değeriyle belirtilen offset hafıza alanını adresler.
  - BX içeriği 1000H ise MOV AX,[BX] komutu data segment içinde 1000H offset adresindeki datayı AX register'ına kopyalar.



Data Adresleme Modları

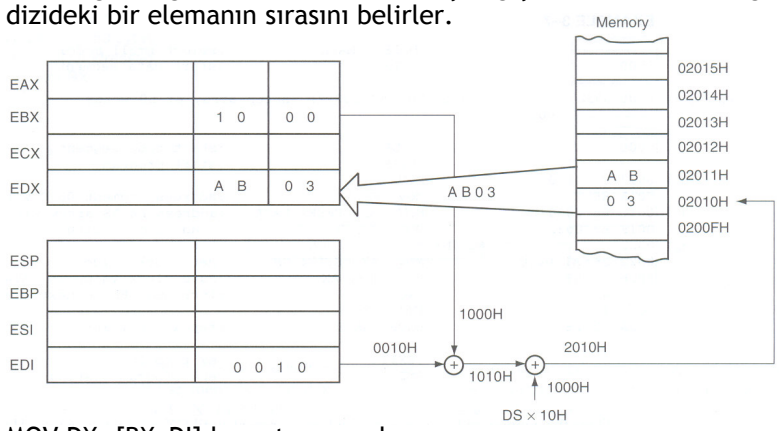
- Register indirect adresleme - detay

Assembly Language	*Size*	*Operation*
MOV CX,[BX]	16 bits	Copies the word contents of the data segment memory location addressed by BX into CX
MOV [BP],DL*	8 bits	Copies DL into the stack segment memory location addressed by BP
MOV [DI],BH	8 bits	Copies BH into the data segment memory location addressed by DI
MOV [DI],[BX]	—	Memory-to-memory transfers are not allowed except with string instructions
MOV AL,[EDX]	8 bits	Copies the byte contents of the data segment memory location addressed by EDX into AL
MOV ECX,[EBX]	32 bits	Copies the doubleword contents of the data segment memory location addressed by EBX into ECX

* BP veya EBP ile data adreslemede default (varsayılan) segment stack segment'tir. Diğerlerinde data segment'tir.

Data Adresleme Modları

- Base-plus-Index adresleme - detay
  - Base register (BP, BX) ile index register (DI, SI) birlikte kullanılır.
  - Base register genellikle bir dizinin başlangıç adresini, index register ise dizideki bir elemanın sırasını belirler.



  - MOV DX, [BX+DI] komutunun çalışması.
  - DS = 0100H, BX = 1000H ve DI = 0010H, MEM = 2010H

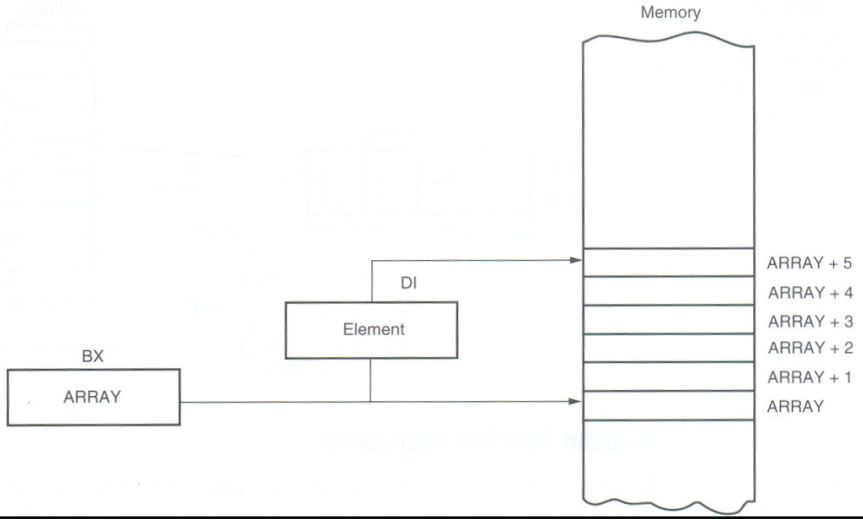
Data Adresleme Modları

- Base-plus-Index adresleme - detay
  - Örnek komutlar.

Assembly Language	*Size*	*Operation*
MOV CX,[BX+DI]	16 bits	Copies the word contents of the data segment memory location addressed by BX plus DI into CX
MOV CH,[BP+SI]	8 bits	Copies the byte contents of the stack segment memory location addressed by BP plus SI into CH
MOV [BX+SI],SP	16 bits	Copies SP into the data segment memory location addressed by BX plus SI
MOV [BP+DI],AH	8 bits	Copies AH into the stack segment memory location addressed by BP plus DI
MOV CL,[EDX+EDI]	8 bits	Copies the byte contents of the data segment memory location addressed by EDX plus EDI into CL
MOV [EAX+EBX],ECX	32 bits	Copies ECX into the data segment memory location addressed by EAX plus EBX

Data Adresleme Modları

- Base-plus-Index adresleme - detay
  - BX dizinin başlangıç adresini, DI elemanın sıra numarasını tutar.



Data Adresleme Modları

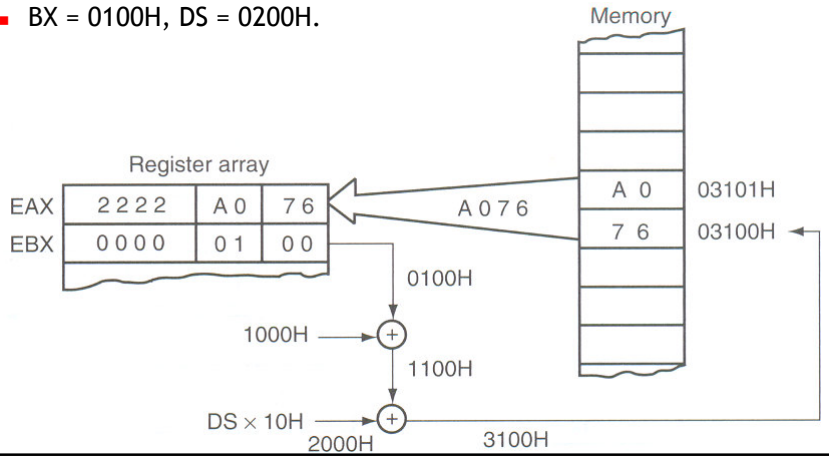
- Base-plus-Index adresleme - detay
  - Aşağıdaki örnekte 10H adresindeki dizi elemanı 20H adresine kopyalanır.

```
                        .MODEL  SMALL           ;select small model
0000                    .DATA                   ;start data segment
0000 0010 [     ARRAY   DB  16  DUP(?)          ;setup array of 16 bytes
             00
           ]
0010 29                 DB  29H                 ;element 10H
0011 001E [             DB  20  dup(?)
             00
           ]
0000                    .CODE                   ;start code segment
                        .STARTUP

0017 B8 0000 R          MOV  BX,OFFSET ARRAY    ;address ARRAY
001A BF 0010            MOV  DI,10H             ;address element 10H
001D 8A 01              MOV  AL,[BX+DI]         ;get element 10H
001F BF 0020            MOV  DI,20H             ;address element 20H
0022 88 01              MOV  [BX+DI],AL         ;save in element 20H

                        .EXIT                   ;exit to DOS
                        END                     ;end program
```

  - BX dizinin başlangıç adresini, DI elemanın sıra numarasını tutar.

Data Adresleme Modları

- Register relative adresleme - detay
  - Bir displacement değeri base veya index register'a (BP, BX, SI, DI) eklenerek memory adreslenir.
  - Aşağıda MOV AX,[BX+1000H] komutunun çalışması görülmektedir.
  - BX = 0100H, DS = 0200H.



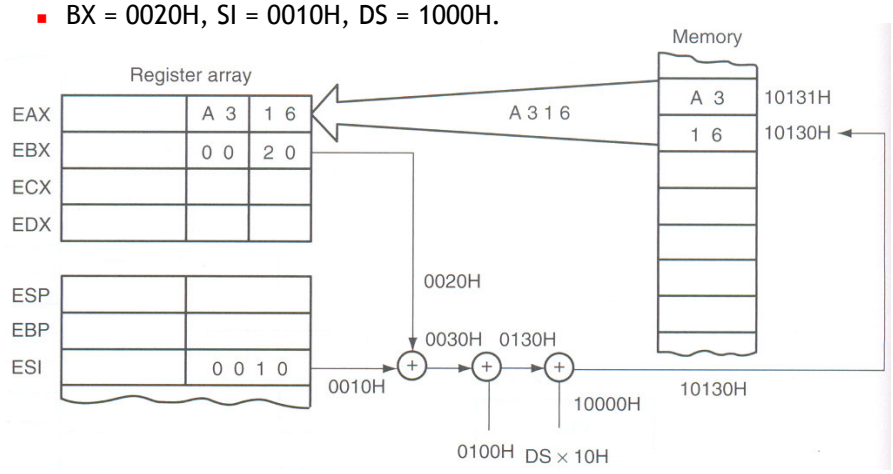
Data Adresleme Modları

- Register relative adresleme - detay
  - Aşağıda register relative adresleme önekleri görülmektedir.

Assembly Language	*Size*	*Operation*
MOV AX,[DI+100H]	16 bits	Copies the word contents of the data segment memory location addressed by DI plus 100H into AX
MOV ARRAY[SI],BL	8 bits	Copies BL into the data segment memory location addressed by ARRAY plus SI
MOV LIST[SI+2],CL	8 bits	Copies CL into the data segment memory location addressed by the sum of LIST, SI, and 2
MOV DI,SET_IT[BX]	16 bits	Copies the word contents of the data segment memory location addressed by SET_IT plus BX into DI
MOV DI,[EAX+10H]	16 bits	Copies the word contents of the data segment location addressed by EAX plus 10H into DI
MOV ARRAY[EBX],EAX	32 bits	Copies EAX into the data segment memory location addressed by ARRAY plus EBX

Data Adresleme Modları

- Base relative plus index adresleme - detay
  - Base plus index ‘e benzer sadece farklı olarak bir displacement ekler.
  - Aşağıda MOV AX,[BX+SI+100H] komutunun çalışması görülmektedir.
  - BX = 0020H, SI = 0010H, DS = 1000H.



Data Adresleme Modları

- Base relative plus index adresleme - detay
  - Aşağıda base relative plus index adresleme önekleri görülmektedir.

Assembly Language	*Size*	*Operation*
MOV DH,[BX+DI+20H]	8 bits	Copies the byte contents of the data segment memory location addressed by the sum of BX, DI and 20H into DH
MOV AX,FILE[BX+DI]	16 bits	Copies the word contents of the data segment memory location addressed by the sum of FILE, BX and DI into AX
MOV LIST[BP+DI],CL	8 bits	Copies CL into the stack segment memory location addressed by the sum of LIST, BP, and DI
MOV LIST[BP+SI+4],DH	8 bits	Copies DH into the stack segment memory location addressed by the sum of LIST, BP, SI, and 4
MOV EAX,FILE[EBX+ECX+2]	32 bits	Copies the doubleword contents of the memory location addressed by the sum of FILE, EBX, ECX, and 2 into EAX

Data Adresleme Modları

- Scaled index adresleme - detay
  - 80386 ve Pentium 4 mikroişlemcilerde bulunmaktadır.
  - 32-bit bir base ve bir index register ile adresleme yapılır.
  - Index register ölçeklendirme faktörüyle (1X, 2X, 4X veya 8X) çarpılır.
  - 2X her elemanı bir word olan diziye, 4X doubleword olan diziye ve 8X ise quadword olan diziye erişmek için kullanılır.
  - MOV AX,[EDI+2*ECX] için faktör 2, MOV EAX,[4*EDI] sadece bir register kullanır ve faktör 4 olarak belirlenmiştir.
  - Aşağıda örnek komutlar görülmektedir.

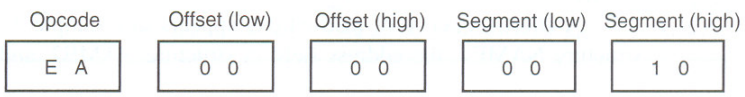
Assembly Language	*Size*	*Operation*
MOV EAX,[EBX+4*ECX]	32 bits	Copies the doubleword contents of the data segment memory location addressed by the sum of 4 times ECX plus EBX into EAX
MOV [EAX+2*EDI+100H],CX	16 bits	Copies CX into the data segment memory location addressed by the sum of EAX, 100H, and 2 times EDI
MOV AL,[EBP+2*EDI+2]	8 bits	Copies the byte contents of the stack segment memory location addressed by the sum of EBP, 2, and 2 times EDI into AL
MOV EAX,ARRAY[4*ECX]	32 bits	Copies the doubleword contents of the data segment memory location addressed by the sum of ARRAY and 4 times ECX into EAX

Program Memory Adresleme Modları

- Program memory adresleme modları direct, relative ve indirect olarak üç tanedir.
- JMP ve CALL komutlarıyla kullanılır.
- **Intersegment jump** hafızada herhangi bir alana atlamayı sağlar.
- **Intrasegment jump** aktif code segment içinde bir alana atlamayı sağlar.

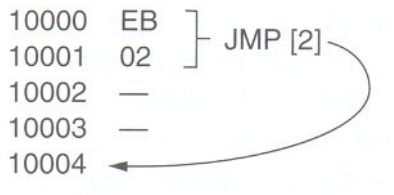
Direct program memory adresleme

- Direct program memory adresleme yüksek seviyeli dillerdede kullanılır. (GOTO, GOSUB)
- Direct adreslemede bir sonraki komut adresi opcode ile birlikte saklanır.
- Aşağıdaki şekilde program bir sonraki komut için 10000H adresine jump yapmaktadır.
- Komut boyutu toplam 5 byte'tır.



Program Memory Adresleme Modları

- Relative program memory adresleme
  - IP değerine göre göreceli atlamayı sağlar.
  - Eğer JMP 2 adres ileri atlayacaksa IP değerine 2 eklenir.
  - JMP ve CALL komutları 8-bit veya 16-bit işaretli displacement değeri kullanır. (Böylece ileri veya geri atlama gerçekleştirilir.)
  - Assembler atlama yapılacak adrese göre displacement değerini hesaplar. Eğer çok uzak adrese atlanacaksa direct adresleme seçilir.
  - 8-bit displacement ile +127 ile -128, 16-bit displacement ile ±32K ve 32-bit displacement ile ±2G (sadece protected mode) atlanabilir.



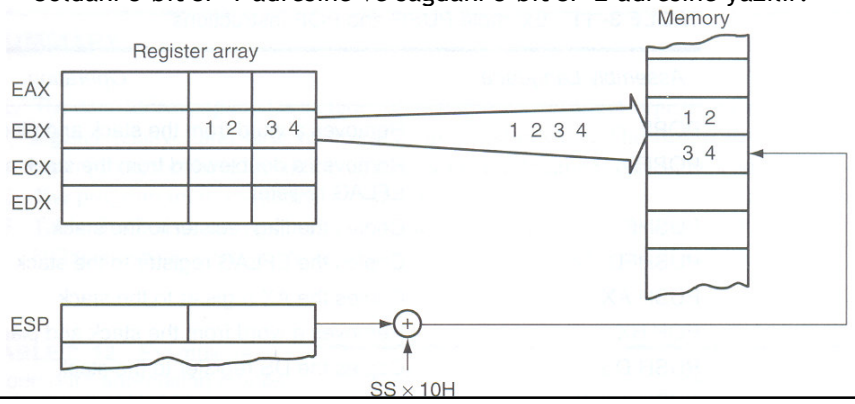
Program Memory Adresleme Modları

- Indirect program memory adresleme
  - JMP ve CALL komutları çok farklı sayıda indirect adresleme yapabilir.
  - Eğer BX = 1000H ise JMP BX komutu aktif code segment içinde 1000H offset adresine atlamayı gerçekleştirir.
  - JMP [BX] komutu, data segment içinde BX offset adresine sahip alan içindeki değeri atlama değeri olarak kullanır. (double indirect veya indirect-indirect olarak adlandırılır.)

Assembly Language	*Operation*
JMP AX	Jumps to the current code segment location addressed by the contents of AX
JMP CX	Jumps to the current code segment location addressed by the contents of CX
JMP NEAR PTR[BX]	Jumps to the current code segment location addressed by the contents of the data segment location addressed by BX
JMP NEAR PTR[DI+2]	Jumps to the current code segment location addressed by the contents of the data segment memory location addressed by DI plus 2
JMP TABLE[BX]	Jumps to the current code segment location addressed by the contents of the data segment memory location address by TABLE plus BX
JMP ECX	Jumps to the current code segment location addressed by the contents of ECX

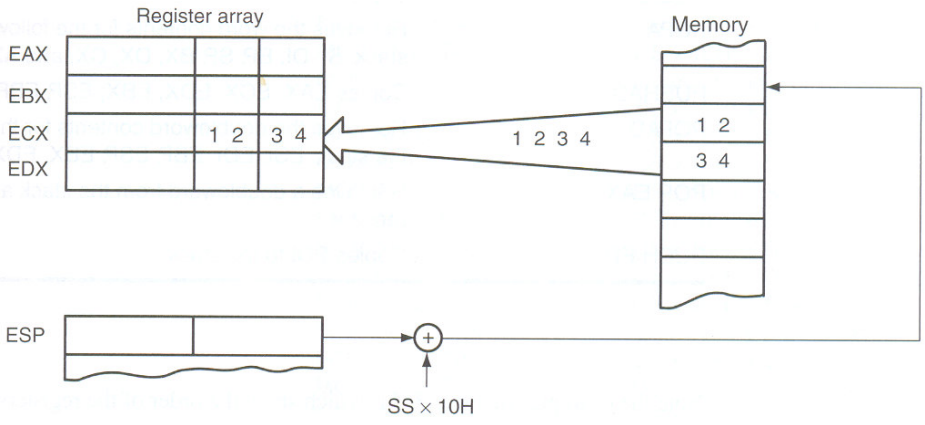
Stack Memory Adresleme Modları

- Stack **LIFO** şeklinde çalışır ve metodların dönüş adresleriyle parametre gönderme ve alma işlemlerini organize eder.
- Data **PUSH** komutuyla stack'a eklenir ve **POP** komutuyla stack'tan silinir.
- Stack hafızası SP(ESP) ve SS register'larıyla oluşturulur.
- Aşağıda **PUSH BX** işleminden sonraki durum görülmektedir.
- Soldaki 8-bit SP-1 adresine ve sağdaki 8-bit SP-2 adresine yazılır.



Stack Memory Adresleme Modları

- Aşağıda **POP CX** işleminden sonraki durum görülmektedir.
- SP adresi 16 bitlik register'ın sağdaki 8-bit ve SP+1 adresi soldaki 8-bit kısma yazılır.



Stack Memory Adresleme Modları

- Aşağıda **PUSH** ve **POP** komutları görülmektedir.

Assembly Language	*Operation*
POPF	Removes a word from the stack and places it into the flag register
POPFD	Removes a doubleword from the stack and places it into the EFLAG register
PUSHF	Copies the flag register to the stack
PUSHFD	Copies the EFLAG register to the stack
PUSH AX	Copies the AX register to the stack
POP BX	Removes a word from the stack and places it into the BX register
PUSH DS	Copies the DS register to the stack
PUSH 1234H	Copies a word-sized 1234H to the stack
POP CS	This instruction is illegal
PUSH WORD PTR[BX]	Copies the word contents of the data segment memory location addressed by BX onto the stack
PUSHA	Copies AX, CX, DX, BX, SP, BP, DI and SI to the stack
POPA	Removes the word contents for the following registers from the stack: SI, DI, BP, SP, BX, DX, CX, and AX
PUSHAD	Copies EAX, ECX, EDX, EBX, ESP, EBP, EDI, and ESI to the stack
POPAD	Removes the doubleword contents for the following registers from the stack: ESI, EDI, EBP, ESP, EBX, EDX, ECX, and EAX
POP EAX	Removes a doubleword from the stack and places it into the EAX register
PUSH EDI	Copies EDI to the stack

Stack Memory Adresleme Modları

- **PUSH** ve **POP** komutları örnek program.
- AX, BX ve CX register'larının değerleri yer değiştirilmektedir.

```
                        .MODEL TINY          ;select tiny model
0000                    .CODE                ;start code segment
                        .STARTUP             ;start program
0100 B8 1000            MOV   AX,1000H       ;load test data
0103 BB 2000            MOV   BX,2000H
0106 B9 3000            MOV   CX,3000H

0109 50                 PUSH  AX             ;1000H to stack
010A 53                 PUSH  BX             ;2000H to stack
010B 51                 PUSH  CX             ;3000H to stack

010C 58                 POP   AX             ;3000H to AX
010D 59                 POP   CX             ;2000H to CBX
010E 5B                 POP   BX             ;1000H to BX
                        .exit                ;exit to DOS
                        end                  ;end program
```

Ödev

- Pentium 4 adresleme modlarını araştırınız ve bir rapor hazırlayınız.