BIL 362 Mikroişlemciler

M.Ali Akcayol Gazi Üniversitesi Bilgisayar Mühendisliği Bölümü



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 modlari
 - Direct program memory adresleme
 - Relative program memory adresleme
 - Indirect program memory adresleme
- Stack memory adresleme modlari



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.

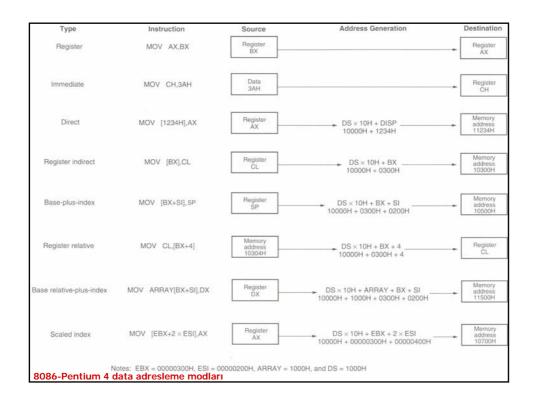


Data Adresleme Modları

• 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)



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.



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.



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 Modlari

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.



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.



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 Modlari

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.

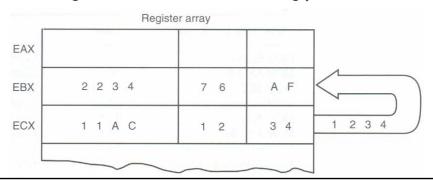


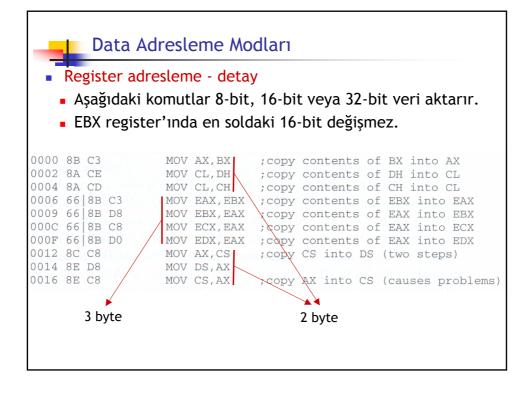
- Register adresleme detay
 - Register adresleme için örnek komutlar

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)

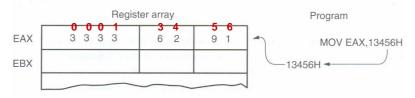


- Register adresleme detay
 - MOV komutu sadece hedef register'ı değiştirir. (CMP ve TEXT komutları hedef register'ı değiştirmez.)
 - MOV BX, CX komutu 1234H değerini CX register'ından BX register'ına kopyalar.
 - EBX register'ının en soldaki 16-bit değişmez.





- 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.



- Immediate 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.



- Immediate 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.

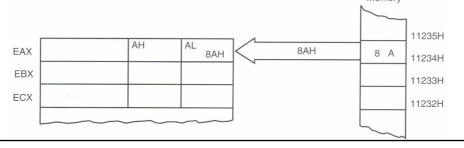
				.MO	DEL TINY	; choose single segment model
	0000			.CO	DE	;start of code segment
				.ST	ARTUP	;start of program
	0100	В8	0000	MOV	AX,0	;place UUUUH into AX
	0103	BB	0000	MOV	BX, 0	;place 0000H into BX
	0106	В9	0000	MOV	CX,0	;place 0000H into CX
	0109	8B	F0	MOV	SI,AX	;copy AX into SI
	010B	8B	F8	MOV	DI,AX	;copy AX into DI
	010D	8B	E8	MOV	BP, AX	;copy AX into BP
MO	V AX. O	kon	nutunun			
	et adre			.EX	IT	;exit to DOS
0113	oct dui c	Jiui		END		end of program;

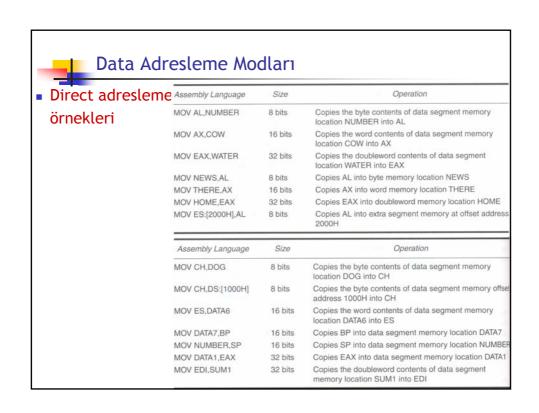


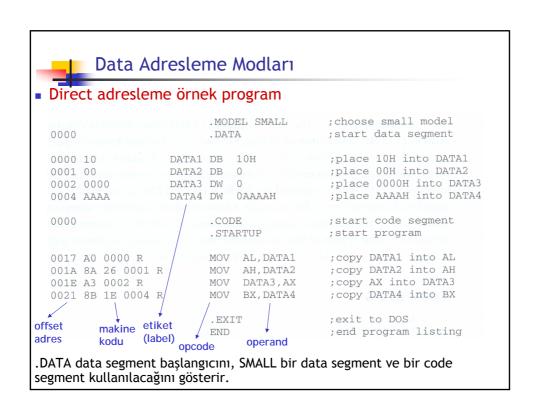
- Immediate adresleme detay
 - Her deyim 4 parçadan oluşur.
 - En solda label bulunur. Bulunduğu hafıza alanına sembolik isim atar.
 - İkinci alan opcode'dur.
 - Üçüncü alan/alanlar opcode'un sağındaki operand'lardır.
 - Son alan en sağdaki comment (açıklama) alanıdır. Açıklama her zaman ; ile başlar.

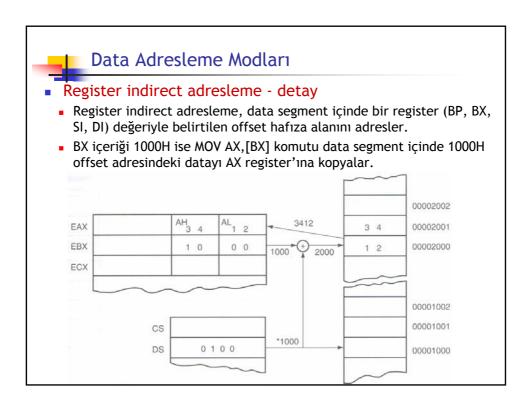
LABEL	OPCODE	OPERAND	COMMENT
DATA1 DATA2	DB DW	23Н 1000Н	;define DATA1 as a byte of 23H ;define DATA2 as a word of 1000H
START:	MOV MOV	AL,BL BH,AL CX,200	copy BL into AL; copy AL into BH; copy 200 into CX

- 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.





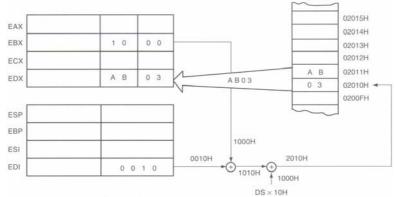




Data Adresleme Modları Register indirect adresleme - detay Assembly Language Size Operation MOV CX,[BX] 16 bits Copies the word contents of the data segment memory location addressed by BX into CX MOV [BP],DL* 8 bits Copies DL into the stack segment memory location addressed by BP 8 bits Copies BH into the data segment memory location addressed by DI Memory-to-memory transfers are not allowed except with MOV [DI],[BX] string instructions Copies the byte contents of the data segment memory MOV AL,[EDX] 8 bits 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.



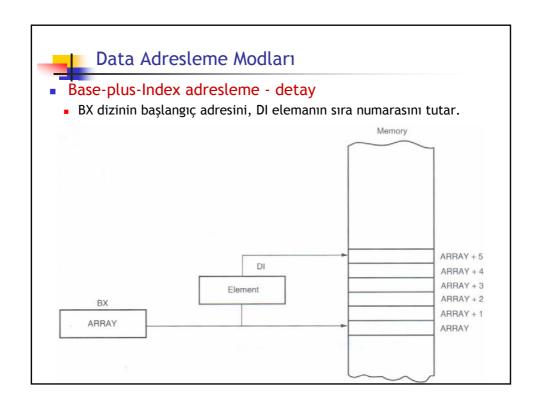
- Base register (BP, BX) ile index register (DI, SI) birlikte kullanılır.
- Base register genellikle bir dizinin başlangıç adresini, index register ise dizideki bir elemanın sırasını belirler.

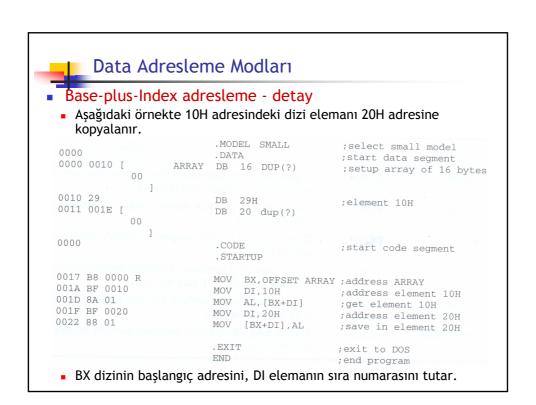


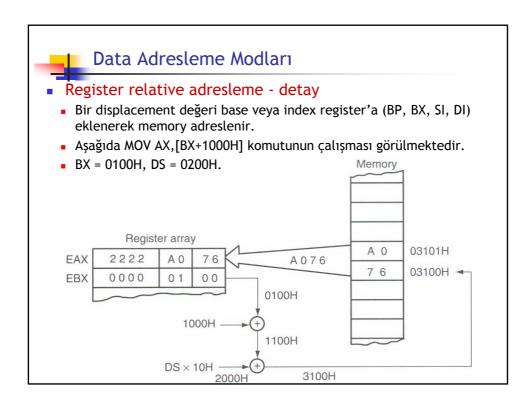
- MOV DX, [BX+DI] komutunun çalışması.
- DS = 0100H, BX = 1000H ve DI = 0010H, MEM = 2010H

- 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 memor 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

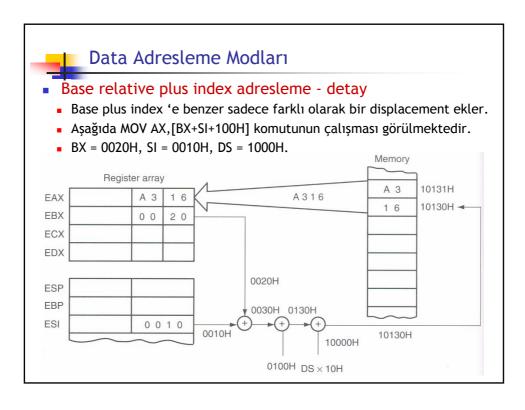






- Register relative adresleme detay
 - Aşağıda register relative adresleme önekleri görülmektedir.

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



- 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



- 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

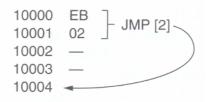
- Direct program memory adresleme yüksek seviyeli dillerdede kullanılır. (GOTO, GOSUB)
- Direct adreslemede bir sonraki komut adresi opcode ile birlikte saklanır.
- Aşağıdaki şekilde program bir sonraki komut için 10000H adresine jump yapmaktadır.
- Komut boyutu toplam 5 byte'tır.

Opcode	Offset (low)	Offset (high)	Segment (low)	Segment (high)
ΕA	0 0	0 0	0 0	1 0



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.

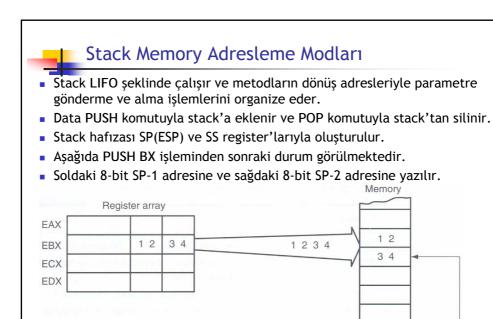




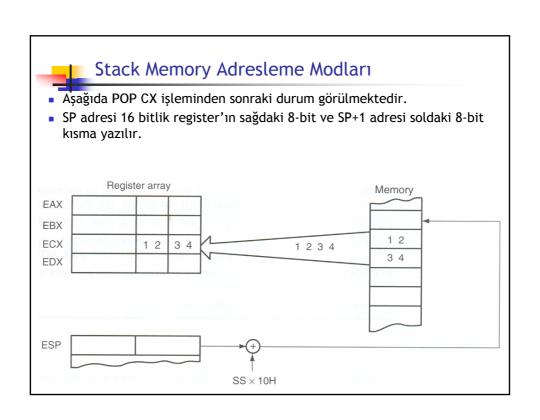
Program Memory Adresleme Modları

- 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



ESP





• 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 stad
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.

0100 B8 1000 MOV AX,1000H ;load test data 0103 BB 2000 MOV BX,2000H	l nt
0103 PP 2000 MOV PV 2000H	
0103 BB 2000 MOV BA, 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	



 Pentium 4 adresleme modlarını araştırınız ve bir rapor hazırlayınız.