

BM-311 Bilgisayar Mimarisi

Hazırlayan: M.Ali Akcayol
Gazi Üniversitesi
Bilgisayar Mühendisliği Bölümü

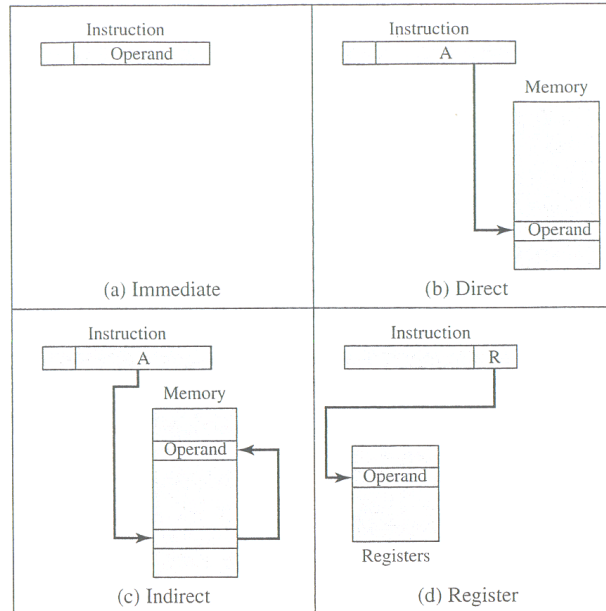
Konular

- Adresleme modları
- Pentium ve PowerPC adresleme modları
- Komut formatları

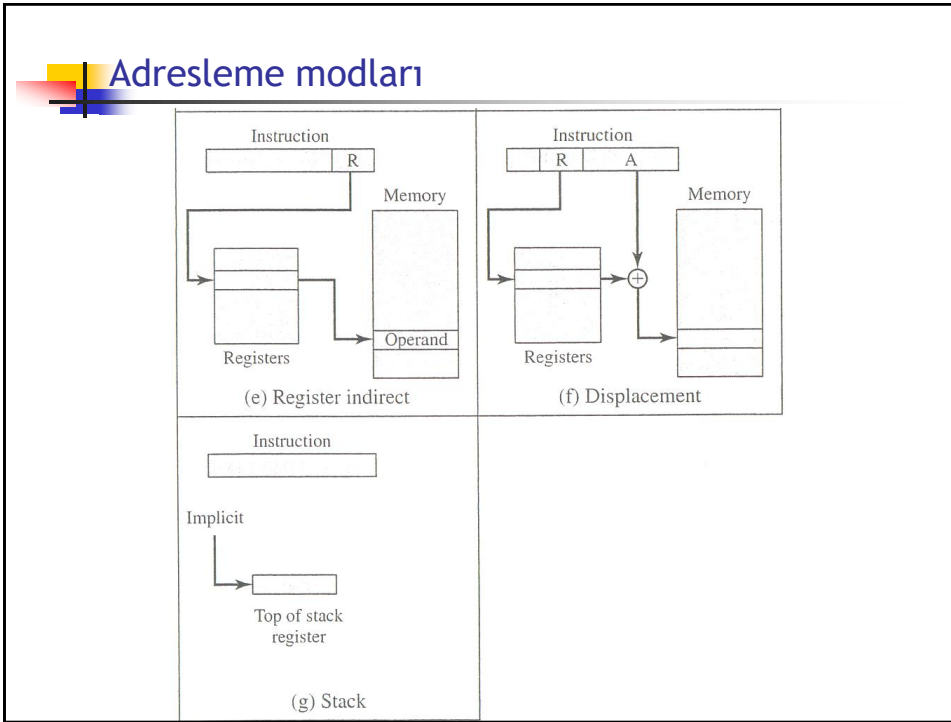
Adresleme modları

- Komutlarda, işlem yapmak için gerekli **operandlar** farklı **adresleme modlarıyla belirlenir.**
- Adresleme modlarını belirlerken **adres aralığı, adresleme esnekliği, memory adres sayısı** ve **adres hesaplama karmaşıklığı** arasında seçim yapılır.
- Temel adresleme modları:
 - **Immediate**
 - **Direct**
 - **Indirect**
 - **Register**
 - **Register indirect**
 - **Displacement**
 - **Stack**

Adresleme modları



Adresleme modları



Adresleme modları

- Kullanılan notasyonlar aşağıda verilmiştir.
 - A = komuttaki adres alanının içeriği
 - R = komuttaki register'ı gösteren alanın içeriği
 - EA = komuttaki operand'la referans gösterilen efektif adres
 - (X) = X adresinin veya X register'ının gösterdiği hafıza adresinin içeriği
- Temel adresleme modları:

Mode	Algorithm	Principal Advantage	Principal Disadvantage
Immediate	Operand = A	No memory reference	Limited operand magnitude
Direct	EA = A	Simple	Limited address space
Indirect	EA = (A)	Large address space	Multiple memory references
Register	EA = R	No memory reference	Limited address space
Register indirect	EA = (R)	Large address space	Extra memory reference
Displacement	EA = A + (R)	Flexibility	Complexity
Stack	EA = top of stack	No memory reference	Limited applicability

Adresleme modları

Immediate (hemen) adresleme

- En basit adresleme modudur. **Operand doğrudan komut içinde verilir.**

$$\text{Operand} = A$$

- Genellikle sabitlerde ve değişkenlerin başlangıç değerlerinin verilmesinde kullanılır.
- Sayılar, 2 tümleyen aritmetiğinde ifade edilir.

Avantaj / dezavantaj

- Avantaj: **Hafıza erişimi gerektirmez.**
- Dezavantaj: **Sayının boyutu ayrılan bit sayısına bağlıdır.** Çoğu komutta az sayıda bit kullanılır.

Adresleme modları

Direct (doğrudan) adresleme

- Komut içindeki adres alanı operandın efektif adresine sahiptir.

$$EA = A$$

- İlk jenerasyon bilgisayarlarda yaygın kullanılmıştır.
- Günümüzdeki modern mikroişlemcilerde **aritmetik işlemlerde yaygın kullanılmamaktadır.**
- **Bir hafıza erişimi gerektirir** ve fazladan adres hesaplaması gerektirmez.

Avantaj / dezavantaj

- Avantaj: **Fazladan adres hesaplaması gerektirmez.**
- Dezavantaj: **Sınırlı adresleme kapasitesine sahiptir.**

Adresleme modları

Indirect (dolaylı) adresleme

- Komut içindeki **adres alanı hafızada bir adresi gösterir.** Operandın efektif adresi bu adresten alınır.

$$EA = (A)$$

- Doğrudan **komutla adreslenebilir** alan 2^K olur (K = komuttaki bit sayısıdır.).
- Her alan için adreslenebilir aralık 2^N olur (N = word length).
- Toplam adreslenebilir alan 2^{K+N} olur.
- Bazı işlemciler kaskad çok seviyeli dolaylı adresleme kullanır.

$$EA = (...(A)...)$$

Avantaj / dezavantaj

- Avantaj: **Daha fazla adreslenebilir alan** oluşturur.
- Dezavantaj: **İki kez hafıza erişimi** gerektirir.

Adresleme modları

Register adresleme

- Direct adreslemeye benzer. Komut içerisinde **hafıza yerine register adreslenir.**

$$EA = R$$

- Hafıza erişimi ve adres hesaplaması yapılmadığı için **çok hızlıdır.**

Avantaj / dezavantaj

- Avantaj: Register sayısı az olduğu için, adresleme için **komuttaki bit sayısı da azdır.**
- **Operand için hafıza erişimi gerektirmez.**
- Dezavantaj: **Adres aralığı azdır.**

Adresleme modları

Register indirect adresleme

- Indirect adreslemeye benzer.
 $EA = (R)$
- **Hafıza erişimi gerektirir.**
- Adres aralığı register adreslemeden fazladır.

Avantaj / dezavantaj

- Avantaj: Register sayısı az olduğu için **komutta adresleme için gerekli bit sayısı azdır.**
- Dezavantaj: **Hafıza erişimi gerektirir.**

Adresleme modları

Displacement adresleme

- Direct adresleme ve register indirect adreslemenin güçlü yönlerini birleştirir.
 $EA = A + (R)$
- Displacement adresleme **en az iki operand gerektirir.** Birisi dolaylı olabilir.
- Örnekte, A direct adresleme ile gösterilmiştir. (R) register indirect adreslemeyle gösterilmiştir.
- Adreslerden birisi opcode ile dolaylı gösterilebilir.
- Aşağıdaki **3 displacement adresleme modu** yaygın kullanılır:
 - Relative addressing
 - Base-register addressing
 - Indexing

Adresleme modları

Displacement adresleme – Relative addressing

- **PC** değerine göre **göreceli adresleme yapar.**
- Hafıza adreslemeler genellikle yakın yerlere yapılıyorsa kullanılır.
- **Adres alanı ikinin tümleyeni** şeklinde gösterilir.

Displacement adresleme – Base-register addressing

- Adreslenen **register hafızada bir adresi gösterir, komut içindeki adres oraya göre yer değiştirmeyi belirler.**
- Register doğrudan veya dolaylı gösterilebilir.
- Adreslenebilecek **register sayısı N** ise ve **komut içindeki adres bit sayısı K** ise, **N farklı bölgede 2^K boyutunda alan adreslenebilir.**

Adresleme modları

Displacement adresleme – Indexing

- Base-register adreslemenin tersi çalışmaya sahiptir.
- Komut içindeki **adres hafızada bir adresi gösterir, register oraya göre yer değiştirmeyi belirler.**
- **Adres alanı ikinin tümleyeni** şeklinde gösterilir.
- Register boyutu, komut içindeki adres alanından daha büyüktür.
- **Tekrarlı işlemlerde performansı artırır.**
- Sadece register değeri değiştirilerek adresleme yapılabilir.

$$EA = A + (R)$$

$$(R) \leftarrow (R) + 1$$

Adresleme modları

Stack adresleme

- Stack önceden ayrılmış belirli bir bloktur.
- Stack üzerinde **en üstteki eleman** bir **işaretçi (pointer) ile gösterilir.**
- **Stack üzerinde adresleme register indirect adreslemedir.**
- Genellikle **procedure call/return** işlemlerinde ve **konfigürasyonu geçici saklamak** için kullanılır.

Konular

- Adresleme modları
- **Pentium ve PowerPC adresleme modları**
- Komut formatları

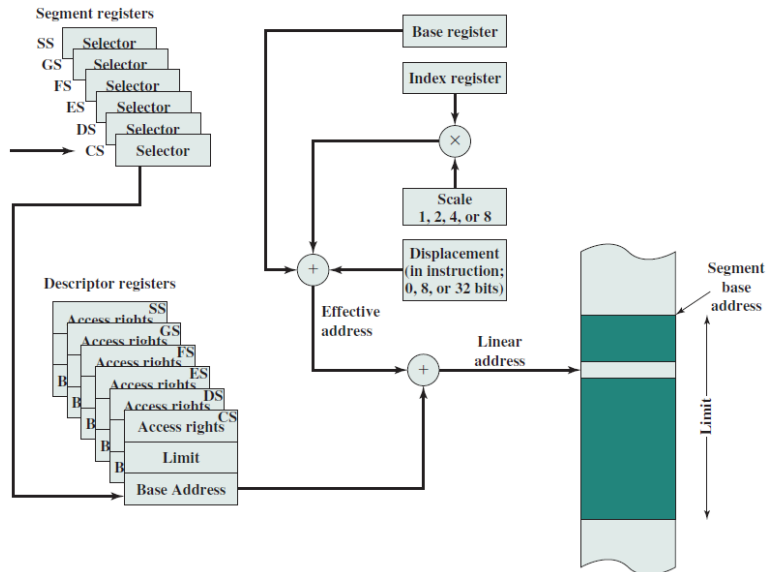
Pentium ve PowerPC adresleme modları

Pentium adresleme modları

- Pentium işlemcilerde adresleme birimi tarafından **effective** veya **virtual** adres denilen **segmentin offset adresi** elde edilir.
- Segmentin **başlangıç adresi** (base adres) ve **efektif adres** toplanarak **linear adres** elde edilir.
- Segment register'ları (DS, ES, FS, GS, CS, SS) **descriptor register**'lardan birisini seçer.
- Descriptor register'lar **segment base adresini**, **limitini** ve **erişim haklarını** belirler.

Pentium ve PowerPC adresleme modları

Pentium adresleme modları



Pentium ve PowerPC adresleme modları

Pentium adresleme modları - devam

- Pentium işlemcilerde çok sayıda adresleme modu vardır.

Mode	Algorithm
Immediate	Operand = A
Register Operand	LA = R
Displacement	LA = (SR) + A
Base	LA = (SR) + (B)
Base with Displacement	LA = (SR) + (B) + A
Scaled Index with Displacement	LA = (SR) + (I) × S + A
Base with Index and Displacement	LA = (SR) + (B) + (I) + A
Base with Scaled Index and Displacement	LA = (SR) + (I) × S + (B) + A
Relative	LA = (PC) + A

LA = linear address

(X) = contents of X

SR = segment register

PC = program counter

A = contents of an address field in the instruction

R = register

B = base register

I = index register

S = scaling factor

Pentium ve PowerPC adresleme modları

Pentium adresleme modları - devam

- **Immediate mode:** Operand komut içinde verilir (byte, word, doubleword).
- **Register operand mode:** Operand register ile verilir (EAX, EBX, ...).
- **Displacement mode:** Operand'ın offset adresi komut içinde verilir (8, 16 veya 32 bit olabilir).
- **Base mode:** Offset adres bir base register ile verilir (8, 16 veya 32 bit olabilir).
- **Base with displacement mode:** Komut içinde base register'a eklenecek bir displacement değeri verilir.

Pentium ve PowerPC adresleme modları

PowerPC adresleme modları

- PowerPC işlemciler Pentium işlemcilere göre **basit ve az sayıda adresleme moduna sahiptir.**

Mode	Algorithm
	Load/Store Addressing
Indirect	$EA = (BR) + D$
Indirect Indexed	$EA = (BR) + (IR)$
	Branch Addressing
Absolute	$EA = I$
Relative	$EA = (PC) + I$
Indirect	$EA = (L/CR)$
	Fixed-Point Computation
Register	$EA = GPR$
Immediate	Operand = I
	Floating-Point Computation
Register	$EA = FPR$

EA = effective address
(X) = contents of X
BR = base register
IR = index register
L/CR = link or count register
GPR = general-purpose register
FPR = floating-point register
D = displacement
I = immediate value
PC = program counter

Pentium ve PowerPC adresleme modları

PowerPC adresleme modları - load/store mimarisi

Indirect adresleme

- Komut içinde **16 bit displacement değeri** vardır ve **base register'a** eklenir.
- Dizi, döngü işlemleri gibi tekrarlı işlemler için önceki adres update edilebilir.

Indirect indexed adresleme

- Bir **index register** ile bir **base register toplanır.**
- Dizi, döngü işlemleri gibi tekrarlı işlemler için önceki adres update edilebilmektedir.

Pentium ve PowerPC adresleme modları

PowerPC adresleme modları – devam

Branch adresleme

- Üç branch adresleme modu: **absolute**, **relative** ve **indirect**.
- **Absolute:**
 - Şartsız atlama komutlarında kullanılır. **Sonraki komutun efektif adresi komut içinde verilen 24-bit değerle bulunur.**
 - **24-bit değer, 32-bit'e sign extend** edilir.
- **Relative:**
 - **Şartsız** atlamalar için **24-bit değer**, **şartlı** atlamalar için **14-bit değer** önce **extent edilir**, sonra PC değerine eklenir.
- **Indirect:**
 - **Sonraki komutun efektif adresi** bir **count veya link register** ile belirlenir.
 - **Count register döngü sayısını da saklayabilir.**

Pentium ve PowerPC adresleme modları

PowerPC adresleme modları – devam

Aritmetik komutlar

- Integer aritmetik işlemlerde **tüm operand'lar register'larda** veya komutun içinde **immediate** olarak tutulur.
- Floating-point aritmetik işlemlerde **tüm operandlar floating-point register'larda** saklanır.
- Bu durumda sadece **register adresleme yapılır.**

Konular

- Adresleme modları
- Pentium ve PowerPC adresleme modları
- **Komut formatları**

Komut formatları

- Komut formatları, komutların bit dizilimini ve her bileşenin (alanın) işlevini gösterir.
- Bir komut formatı, **opcode** bulundurmak zorundadır, doğrudan veya dolaylı **birden fazla operand** bulundurabilir.
- **Komut formatı**, operandlar için doğrudan veya dolaylı **adresleme modları belirler.**

Komut uzunluğu

- **Komut uzunluğu**; hafıza boyutu, hafıza organizasyonu, bus yapısı, işlemci karmaşıklığı ve işlemci hızını etkiler.
- Komut uzunluğu arttıkça, programcı için **daha esnek** hale gelir.
- Komut uzunluğu arttıkça, **operand sayısı** ve **adresleme modu sayısı artar.**

Komut formatları

Bitlerin atanması

- Komut içindeki **alanların işlevlerinin belirlenmesidir.**
- **Sabit genişlikteki komutlar için opcode boyutu arttıkça operandlar için ayrılan yer azalacaktır.**
- Aşağıdaki faktörler bitlerin atanması ve komut uzunluğunu belirlerken göz önüne alınır:
 - **Adresleme modu sayısı:** Adresleme modu dolaylı veya doğrudan belirlenebilir. Doğrudan belirlemede bir veya birkaç bit ayrılması gerekir.
 - **Operand sayısı:** Günümüzde iki operandlı komutlar yaygın kullanılır.
 - **Register/hafıza seçimi:** Bir tane ve dolaylı adreslenen register kullanılabilir (**AC**). Birden fazla register kullanılırsa birkaç bitin ayrılması gerekir. Günümüzde 8-32 adet **user visible register** kullanılmaktadır.
 - **Adres aralığı:** Adres aralığı komut içinde ayrılan bitlerin sayısına bağlıdır.
- **Orthogonality:** Komutun bileşenleri opcode'dan bağımsızdır.

Komut formatları

Farklı uzunlukta (variable-length) komutlar

- Komut boyutlarının farklı olması **işlemci karmaşıklığını artırır.**
- RISC ve superscalar işlemciler **fixed-length komutları kullanır.**
- Komut uzunluklarının aynı olması **decode işlemini kolaylaştırır.**
- Komut uzunluklarının ve operand'ların **aynı uzunlukta olması operand fetch ile decode işleminin aynı anda yapılmasına olanak sağlar.**