

BM-311 Bilgisayar Mimarisi

Hazırlayan: M.Ali Akcayol
Gazi Üniversitesi
Bilgisayar Mühendisliği Bölümü

Konular

- Giriş
- Mikro işlemler
 - Fetch cycle
 - Indirect cycle
 - Interrupt cycle
 - Execute cycle
 - Instruction cycle
- Mikroişlemcinin denetimi
- Kontrol birimi sinyalleri
- Hardwired kontrol birimi

Giriş

- Mikroişlemciler instruction'ları çalıştırırken **çok sayıda alt işlemi gerçekleştirir.**
- Bir komut kümesi oluşturulduktan sonra her komut için yapılacak **alt işlemlerin donanımsal veya yazılımsal olarak tasarlanması gereklidir.**
- Bir mikroişlemcinin instruction'ları çalıştırmak için mikroişlemci dışındaki bileşenleri de bilmesi gereklidir.
- Aşağıdaki bileşenler **mikroişlemcinin fonksiyonel ihtiyaçları** olarak tanımlanır:
 - İşlemler (opcode)
 - Adresleme modları
 - Register'lar
 - I/O module arayüzleri
 - Memory modül arayüzleri
 - Interrupt'lar

Konular

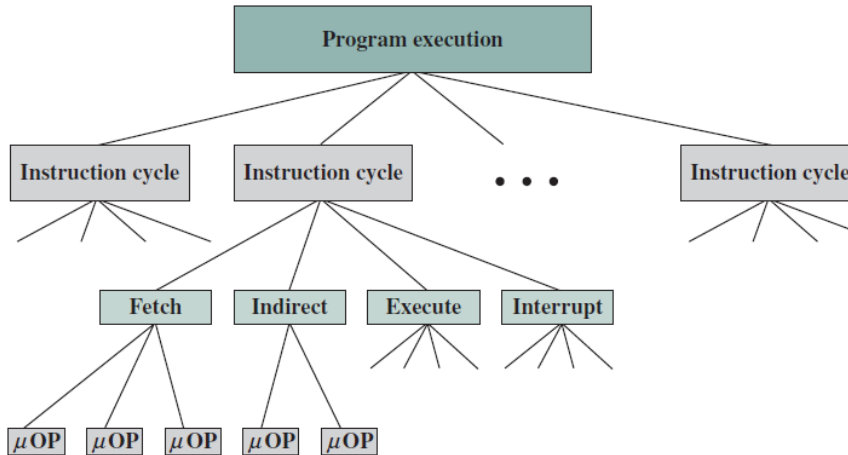
- Giriş
- **Mikro işlemler**
 - Fetch cycle
 - Indirect cycle
 - Interrupt cycle
 - Execute cycle
 - Instruction cycle
- Mikroişlemcinin denetimi
- Kontrol birimi sinyalleri
- Hardwired kontrol birimi

Mikro işlemler

- Bir programı çalıştırırken her komut için instruction cycle tekrarlanır.
- Programın çalışması sırasında **komutlar yazıldığı sırada çalıştırılmayabilir.**
- Run time'da **komutlar zamana göre sıralı** olarak çalıştırılırlar.
- Her **instruction**, belirli sayıda **küçük iş parçaları halinde oluşturulur** (fetch, indirect, execute, interrupt).
- Her iş parçasını **mikro işlemler (micro-operations)** oluşturur.
- **Mikro işlemler** bir mikroişlemcinin **atomik işlemleridir.**

Mikro işlemler

- Bir program çalıştırılırken her instruction için **alt işlemler sırasıyla yerine getirilir.**



Konular

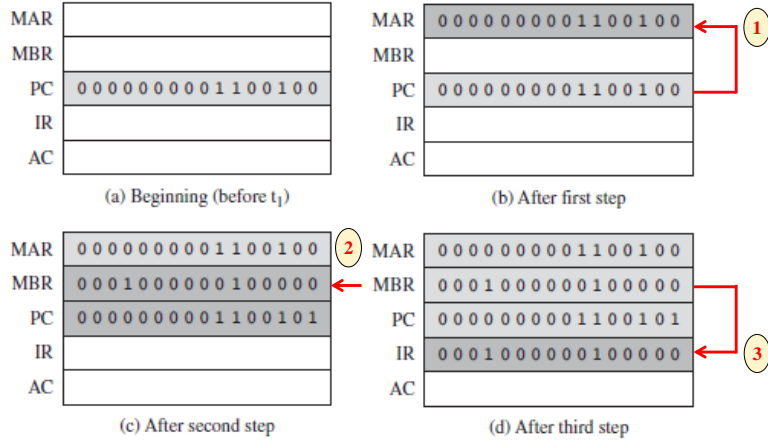
- Giriş
- Mikro işlemler
 - Fetch cycle
 - Indirect cycle
 - Interrupt cycle
 - Execute cycle
 - Instruction cycle
- Mikroişlemcinin denetimi
- Kontrol birimi sinyalleri
- Hardwired kontrol birimi

Fetch cycle

- Fetch cycle her komut döngüsünün başlangıcında gerçekleşir ve komut hafızadan fetch edilir.
- Mikroişlemcide MAR, MBR, PC ve IR bulunmaktadır.
- **MAR (Memory Address Register):** Hafıza üzerinde okuma ve yazma yapılacak adresi tutar.
- **MBR (Memory Buffer Register):** Hafızadan okunan veya hafızaya yazılan veriyi tutar.
- **PC (Program Counter):** Fetch edilecek sonraki komutun adresini tutar.
- **IR (Instruction Register):** Fetch edilen son komutu tutar.

Fetch cycle

- Fetch cycle aşağıdaki alt işlemler ile gerçekleştirilir.

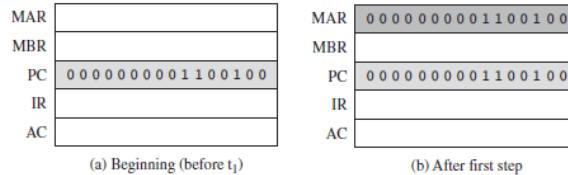


Fetch cycle

- Fetch cycle aşağıdaki sıralı zaman aralıklarında gerçekleştirilir.

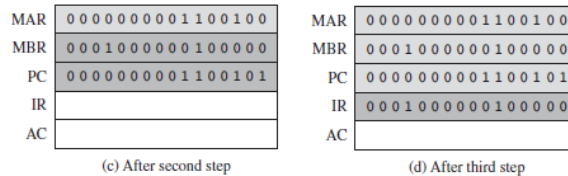
Alternatif 1

$t_1: \text{MAR} \leftarrow (\text{PC})$
 $t_2: \text{MBR} \leftarrow \text{Memory}$
 $\text{PC} \leftarrow (\text{PC}) + I$
 $t_3: \text{IR} \leftarrow (\text{MBR})$



Alternatif 2

$t_1: \text{MAR} \leftarrow (\text{PC})$
 $t_2: \text{MBR} \leftarrow \text{Memory}$
 $t_3: \text{PC} \leftarrow (\text{PC}) + I$
 $\text{IR} \leftarrow (\text{MBR})$



Konular

- Giriş
- Mikro işlemler
 - Fetch cycle
 - **Indirect cycle**
 - Interrupt cycle
 - Execute cycle
 - Instruction cycle
- Mikroişlemcinin denetimi
- Kontrol birimi sinyalleri
- Hardwired kontrol birimi

Indirect cycle

- Indirect cycle'da aşağıdaki işlemler zaman sıralı olarak gerçekleştirilir:
 - Komut içerisindeki adres alanındaki değer ile hafızaya gidilir.
 - **Hafızadan okunan değer MBR'a aktarılır.**
 - Komut içerisindeki adres değeri MBR'daki değer ile güncellenir.

```
t1: MAR ← (IR(Address))  
t2: MBR ← Memory  
t3: IR(Address) ← (MBR(Address))
```

Konular

- Giriş
- Mikro işlemler
 - Fetch cycle
 - Indirect cycle
 - **Interrupt cycle**
 - Execute cycle
 - Instruction cycle
- Mikroişlemcinin denetimi
- Kontrol birimi sinyalleri
- Hardwired kontrol birimi

Interrupt cycle

- Interrupt cycle'da aşağıdaki işlemler sıralı gerçekleştirilir:
 - **PC değeri MBR'a aktarılır.**
 - **Mevcut adres değeri** (MBR), hafızada belirlenen adrese (Save_Address) **saklanır.**
 - Interrupt için **atlanacak adres PC'a aktarılır.**
 - MBR'a aktarılan dönüş adresi hafızada belirlenen adrese (Save_Address) aktarılır.

```
t1: MBR ← (PC)
t2: MAR ← Save_Address
      PC ← Routine_Address
t3: Memory ← (MBR)
```

Konular

- Giriş
- Mikro işlemler
 - Fetch cycle
 - Indirect cycle
 - Interrupt cycle
 - **Execute cycle**
 - Instruction cycle
- Mikroişlemcinin denetimi
- Kontrol birimi sinyalleri
- Hardwired kontrol birimi

Execute cycle

- Fetch cycle, indirect cycle ve interrupt cycle basittir ve her seferinde aynı mikro işlemler tekrarlanır.
- **Execute cycle'da** yapılacak işleme göre hem **zaman aralığı sayısı** hem de **sıralama farklı olmaktadır.**
- **ADD komutu** aşağıdaki gibi gerçekleştirilir:
 - Komut içerisindeki **adres MAR'a aktarılır.**
 - Hafızadan **okunan değer MBR'a aktarılır.**
 - **Toplama komutu execute edilir.**

```
ADD R1, X
```

```
t1: MAR ← (IR(address))
```

```
t2: MBR ← Memory
```

```
t3: R1 ← (R1) + (MBR)
```


Execute cycle

- **ISZ (Increment and Skip If Zero) komutu** aşağıdaki gibi gerçekleştirilir:
 - Komut içerisindeki **adres MAR'a aktarılır.**
 - Hafızadan **okunan değer MBR'a aktarılır.**
 - **MBR değeri 1 artırılır.**
 - MBR değeri hafızaya aktarılır.
 - **MBR=0 ise, PC değeri** Instruction boyutu kadar **artırılır.**

ISZ X

```
t1: MAR ← (IR(address))
t2: MBR ← Memory
t3: MBR ← (MBR) + 1
t4: Memory ← (MBR)
    If ((MBR) = 0) then (PC ← (PC) + I)
```

Execute cycle

- **BSA (Branch and Save PC Address) komutu** aşağıdaki gibi gerçekleştirilir (call/return'de procedure başına adres yazar):
 - Komut içerisindeki **adres MAR'a aktarılır.**
 - **PC değeri MBR'a aktarılır.**
 - Komut içerisindeki **adres PC'a aktarılır.**
 - **MBR değeri hafızaya aktarılır.**
 - PC değeri instruction boyutu kadar artırılır.

BSA X

```
t1: MAR ← (IR(address))
    MBR ← (PC)
t2: PC ← (IR(address))
    Memory ← (MBR)
t3: PC ← (PC) + I
```

Konular

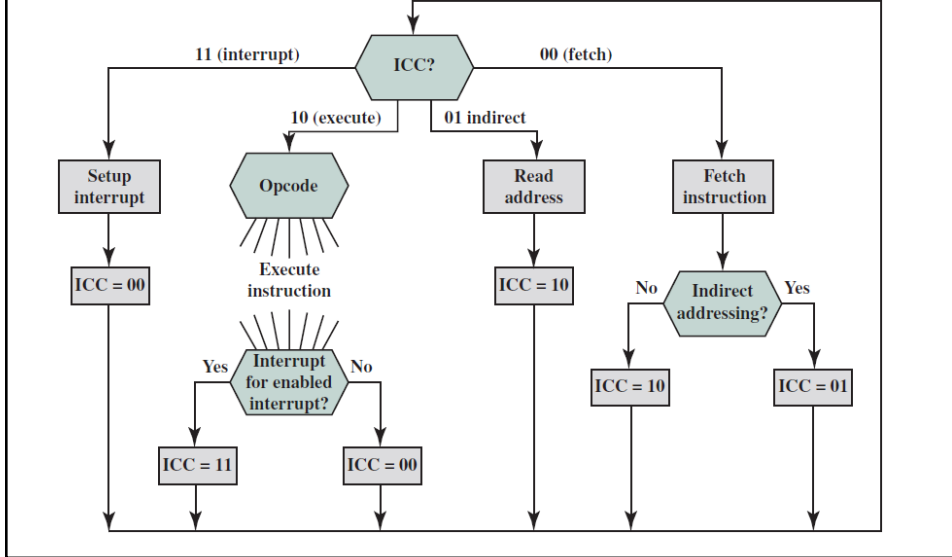
- Giriş
- Mikro işlemler
 - Fetch cycle
 - Indirect cycle
 - Interrupt cycle
 - Execute cycle
 - **Instruction cycle**
- Mikroişlemcinin denetimi
- Kontrol birimi sinyalleri
- Hardwired kontrol birimi

Instruction cycle

- Instruction cycle'in her aşaması zamana göre sıralı küçük işlemler halinde gerçekleştirilir.
- **Her aşama için sadece bir tane sıralı işlem kümesi vardır.**
- Her komut için **opcode içerisinde kendisine ait kod oluşturulur.**
- Her aşama için aşağıdaki komut döngü kodlarının (**Instruction Cycle Code - ICC**) atandığını varsayalım:
 - 00: Fetch
 - 01: Indirect
 - 10: Execute
 - 11: Interrupt

Instruction cycle

- Instruction cycle için akış şeması aşağıdaki gibi tanımlanabilir.



Konular

- Giriş
- Mikro işlemler
 - Fetch cycle
 - Indirect cycle
 - Interrupt cycle
 - Execute cycle
 - Instruction cycle
- Mikroişlemcinin denetimi
- Kontrol birimi
- Hardwired kontrol birimi

Mikroişlemcinin denetimi

Fonksiyonel gereksinimler

- Mikroişlemci tasarımında öncelikle **kontrol biriminin fonksiyonel gereksinimleri belirlenir.**
- Fonksiyonel gereksinimler, **kontrol biriminin yapmak zorunda olduğu işlerdir.**
- Mikroişlemci tasarımında aşağıdakilerin belirlenmesi gereklidir:
 - İşlemcinin **temel bileşenleri tanımlanmalıdır.**
 - İşlemcinin **mikro işlemleri belirlenmelidir.**
 - Mikro işlemlerin gerçekleşmesi için **kontrol biriminin yapması gereken işlevler tanımlanmalıdır.**

Mikroişlemcinin denetimi

İşlemcinin temel bileşenleri

- Aşağıda işlemcinin temel bileşenleri verilmiştir:
 - **ALU:** Temel aritmetik mantık işlem birimi
 - **Register'lar:** İşlemci içerisinde internal veri saklama birimleri
 - **Internal data paths:** Register'lar arasında veya register'lar ile ALU arasında veri aktarımı için kullanılır.
 - **External data paths:** Register'lar ile hafıza ve I/O modülleri arasında veri aktarımı için kullanılır.
 - **Control unit:** İşlemci içerisindeki işlemleri organize eder.

Mikroişlemcinin denetimi

- **Mikro işlemler** bir zaman aralığında **aşağıdakilerden birisini yapar:**
 - Bir register'dan diğerine veri aktarımı
 - Bir register'dan bir external arayüze (örn.: system bus) veri aktarımı
 - Bir external arayüzden bir register'a veri aktarımı
 - Register'lar giriş ve çıkış olmak üzere aritmetik ve mantık işlemin gerçekleştirilmesi
- Kontrol birimi temel olarak iki işlevi gerçekleştirir:
 - **Sıralama (Sequencing):** Mikro işlemlerin doğru sıralanmasını sağlar.
 - **Çalıştırma (Execution):** Kontrol birimi her mikro işlemin gerçekleştirilmesini sağlar.

Mikroişlemcinin denetimi

Kontrol sinyalleri

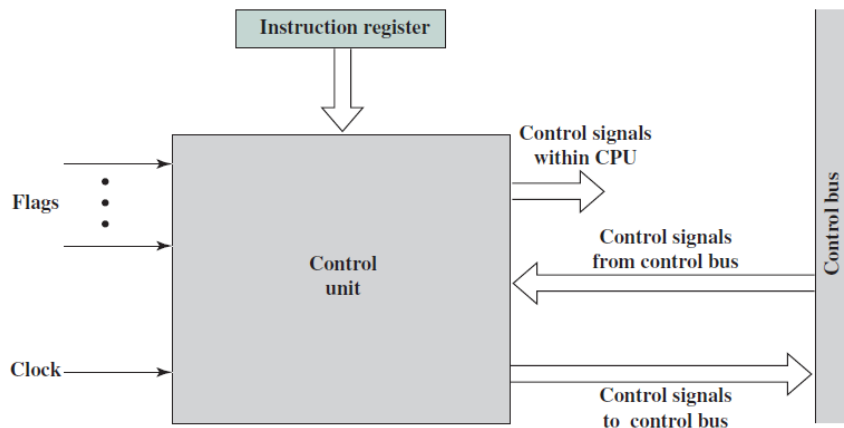
- Kontrol biriminin her bir mikro işlemi gerçekleştirebilmesi için gerekli girişleri alması gereklidir.
- **Kontrol birimi**, aldığı girişlere göre istenen **işlemi gerçekleştirecek birimlerin aktif yapılması** için **kontrol sinyallerini üretir.**
- Kontrol birimi, **CPU içerisindeki birimlere** iletilecek **sinyalleri CPU internal bus aracılığıyla iletir.**
- **Yukarıdaki işlemlerin tümünün gerçekleştirilmesi için gerekli kod bilgisi** instruction register'da bulunan **opcode** içerisinden alınır.

Konular

- Giriş
- Mikro işlemler
 - Fetch cycle
 - Indirect cycle
 - Interrupt cycle
 - Execute cycle
 - Instruction cycle
- Mikroişlemcinin denetimi
- **Kontrol birimi**
- Hardwired kontrol birimi

Kontrol birimi

- Kontrol biriminin temel bileşenleri aşağıdadır.



Kontrol birimi

Kontrol birimine girişler

- **Clock:** Kontrol birimi her clock cycle'da bir veya birkaç tane mikro işlemi yerine getirir.
- **Instruction register:** Opcode ve adresleme modu ile hangi mikro işlemlerin execute aşamasında çalışacağı belirlenir.
- **Flags:** Önceki ALU işlemlerinin sonucunda işlemcinin durumunu belirlemek için kullanılır.
- **Kontrol bus'tan alınan kontrol sinyalleri:** Kontrol bus üzerinden kontrol birimine sinyaller alınır.

Kontrol biriminden çıkışlar

- **İşlemci içerisinde kontrol sinyalleri:** Veriler register'lar arasında aktarılır veya ALU işlemi gerçekleştirilir.
- **Kontrol bus için kontrol sinyalleri:** Hafıza için ve I/O modülleri için kontrol sinyalleri iletilir.

Kontrol birimi

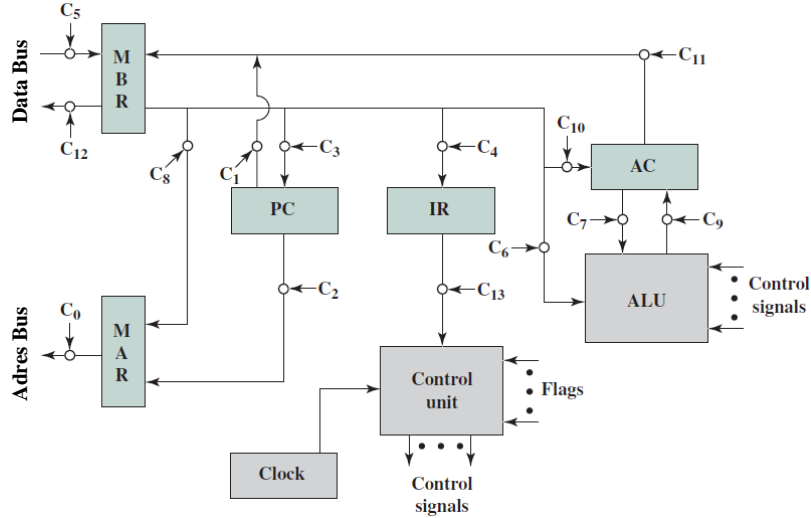
Instruction Fetch Cycle için kontrol biriminin yapacağı işlemler:

- Bir kontrol sinyal ile **MAR içeriğini adres bus'a** aktaracak kapı devreleri açılır.
- Kontrol bus'a hafızadan **okuma işareti** yerleştirilir.
- Bir kontrol sinyal ile **data bus'taki verileri MBR'a** aktaracak devreler açılır.
- Kontrol sinyalleri ile **PC değerine 1 eklenerek, PC değeri güncellenir.**
- Bir kontrol sinyal ile **MBR ile IR arasındaki** kapı devresi açılır.

Kontrol birimi

Kontrol sinyalleri – örnek

- Şekilde veri yolları ve kontrol sinyalleri görülmektedir.



Kontrol birimi

Kontrol sinyalleri – örnek

- Mikroişlemcide **bir tane AC bulunmaktadır.**
- C_i 'ler kontrol sinyalleridir ve veri geçişlerini kontrol ederler.
- Kontrol birimi girişleri: **clock, instruction register ve flags.**
- Her clock cycle'da** kontrol birimi **tüm girişleri okur** ve **bir grup kontrol sinyali üretir.**
- Kontrol sinyalleri 3 ayrı hedefe gönderilir:**
 - Veri yolları:** Kontrol birimi, internal veri akışını kontrol eder.
 - ALU:** Kontrol birimi, ALU'nun yapacağı işlemleri farklı mantık devrelerini aktif yaparak denetler.
 - System bus:** Kontrol birimi, sistem bus üzerinden kontrol komutlarını (örn.: memory READ, memory WRITE) gönderir.

Kontrol birimi

Kontrol sinyalleri – örnek

- Kontrol birimi her clock pulse ile bir mikro işlemi başlatır.
- Her mikro işlem için bir veya daha fazla kontrol sinyali oluşturur.

	Micro-operations	Active Control Signals
Fetch:	$t_1: MAR \leftarrow (PC)$	C_2
	$t_2: MBR \leftarrow \text{Memory}$ $PC \leftarrow (PC) + 1$	C_5, C_R
	$t_3: IR \leftarrow (MBR)$	C_4
Indirect:	$t_1: MAR \leftarrow (IR(\text{Address}))$	C_8
	$t_2: MBR \leftarrow \text{Memory}$	C_5, C_R
	$t_3: IR(\text{Address}) \leftarrow (MBR(\text{Address}))$	C_4
Interrupt:	$t_1: MBR \leftarrow (PC)$	C_1
	$t_2: MAR \leftarrow \text{Save-address}$ $PC \leftarrow \text{Routine-address}$	
	$t_3: \text{Memory} \leftarrow (MBR)$	C_{12}, C_W

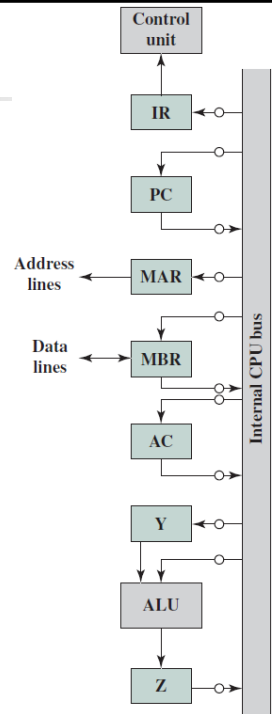
C_R = Read control signal to system bus.

C_W = Write control signal to system bus.

Kontrol birimi

Mikroişlemci iç organizasyonu

- Mikroişlemci içindeki internal bus ile her bileşen arasında bağlantı vardır.
- Bir bus ile **ALU** ve **tüm register'lar** arasında veri aktarımı yapılır.
- Şekildeki mikroişlemciye **Y** ve **Z** olarak **iki adet register eklenmiştir**.
- **ALU işlemi** gerektiğinde **girişlerden birisi** internal bus ile **Y'den alınır**, diğeri ise **AC'den** alınabilir.
- **ALU** kombinatoriyal devredir ve **hafızaya sahip değildir**.
- **ALU çıkışı** doğrudan bus'a verilmez, geçici olarak **Z register'ında tutulur**.

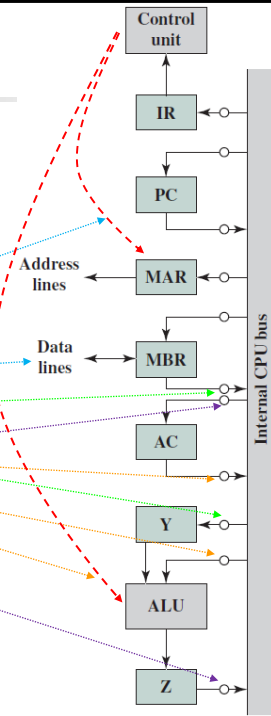


Kontrol birimi

Mikroişlemci iç organizasyonu

- **AC register'ı ile hafızada bir adresteki değerin toplanması ve sonucun AC'ye aktarılması** aşağıdaki gibi gerçekleştirilir:

$t_1: \text{MAR} \leftarrow (\text{IR}(\text{address}))$
 $t_2: \text{MBR} \leftarrow \text{Memory}$
 $t_3: \text{Y} \leftarrow (\text{MBR})$
 $t_4: \text{Z} \leftarrow (\text{AC}) + (\text{Y})$
 $t_5: \text{AC} \leftarrow (\text{Z})$



Konular

- Giriş
- Mikro işlemler
 - Fetch cycle
 - Indirect cycle
 - Interrupt cycle
 - Execute cycle
 - Instruction cycle
- Mikroişlemcinin denetimi
- Kontrol birimi
- **Hardwired kontrol birimi**

Hardwired kontrol birimi

Kontrol birimi girişleri

- Kontrol birimi **donanımsal (hardwired)** veya **mikroprogramlanmış (microprogrammed)** oluşturulabilir.
- Donanımsal kontrol birimi bir **durum makinesi** devresidir.
- Kontrol biriminin **temel girişleri**; instruction register, clock, flags ve control bus sinyalleridir.
- Instruction register'dan alınan **opcode kontrol biriminin girişidir** ve decoder tarafından kullanılır.
- Decoder n bit giriş için 2^n çıkış üretebilir.
- **Clock girişi** kontrol biriminin **sıralı pulse'lar üretmesini sağlar**.
- Clock girişleri t_1, t_2, t_3, \dots için yukarı sayıcı devresine giriş oluşturur ve **her mikro işlemin sonunda sayıcı sıfırlanır**.

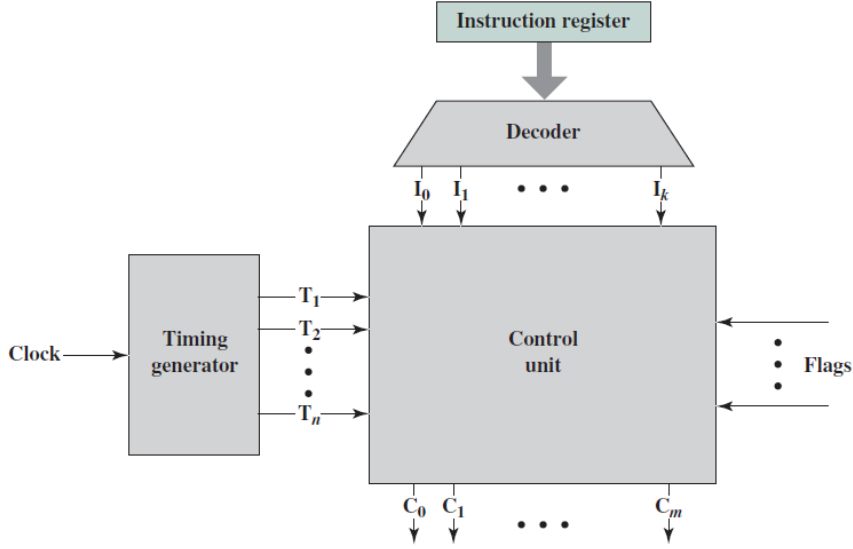
Hardwired kontrol birimi

- 4 giriş ve 16 çıkışa sahip decoder

I1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
I2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
I3	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
I4	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
O1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
O2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
O3	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
O4	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
O5	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
O6	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
O7	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
O8	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
O9	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
O10	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
O11	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
O12	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
O13	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
O14	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
O15	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
O16	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Hardwired kontrol birimi

- Decode edilen girişlerle birlikte kontrol birimi



Hardwired kontrol birimi

Kontrol birim mantık devresi

- Kontrol birimi aldığı girişlere göre **çıkış sinyalleri oluşturur**.
- Herhangi bir kontrol sinyali için Boolean expression yazılabilir.
- PQ instruction cycle'daki aşamaları göstermektedir.

PQ = 00 Fetch Cycle
 PQ = 01 Indirect Cycle
 PQ = 10 Execute Cycle
 PQ = 11 Interrupt Cycle

	Micro-operations	Active Control Signals
Fetch:	t ₁ : MAR ← (PC)	C ₂
	t ₂ : MBR ← Memory	
	PC ← (PC) + 1	
	t ₃ : IR ← (MBR)	C ₄
Indirect:	t ₁ : MAR ← (IR(Address))	C ₅ , C _R
	t ₂ : MBR ← Memory	
	t ₃ : IR(Address) ← (MBR(Address))	
	t ₁ : MBR ← (PC)	C ₁
Interrupt:	t ₂ : MAR ← Save-address	C ₁₂ , C _W
	PC ← Routine-address	
	t ₃ : Memory ← (MBR)	

C_R = Read control signal to system bus.

C_W = Write control signal to system bus.

- C₅ için Boolean expression aşağıdaki gibidir.

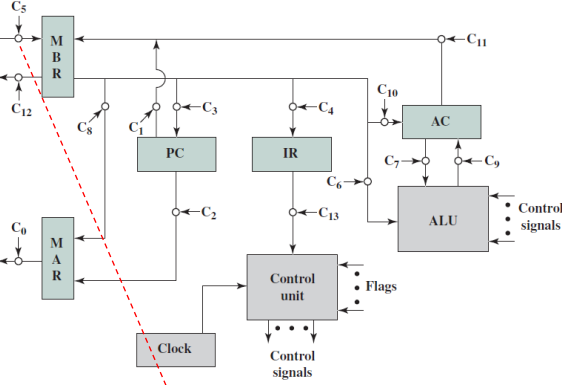
$$C_5 = \bar{P} \cdot \bar{Q} \cdot T_2 + \bar{P} \cdot Q \cdot T_2$$

Hardwired kontrol birimi

Kontrol birim mantık devresi

- C_5 execute aşamasında da kullanılmaktadır.
- **LDA, ADD ve AND** işlemlerinde **hafızadan okuma yapıyor.**

PQ = 00 Fetch Cycle
PQ = 01 Indirect Cycle
PQ = 10 Execute Cycle
PQ = 11 Interrupt Cycle



- C_5 için yeni Boolean expression aşağıdaki gibidir.

$$C_5 = \bar{P} \cdot \bar{Q} \cdot T_2 + \bar{P} \cdot Q \cdot T_2 + P \cdot \bar{Q} \cdot (LDA + ADD + AND) \cdot T_2$$

Ödev

- Mikroprogramlanmış kontrol birimi konusunda detaylı bir ödev hazırlayınız. Hardwired kontrol birimiyle karşılaştırmasını içeren bir araştırma makalesi inceleyip makalenin tam metni ile birlikte inceleme sonuçlarını değerlendirerek ödev raporuna ekleyiniz.