

BİLGİSAYAR ORGANİZASYONU ve TASARIMI

YRD. DOÇ. DR. FATİH KELEŞ

Temel Bilgisayar Yapısı ve Devreleri

Temel Bilgisayar Organizasyonu ve Tasarımı

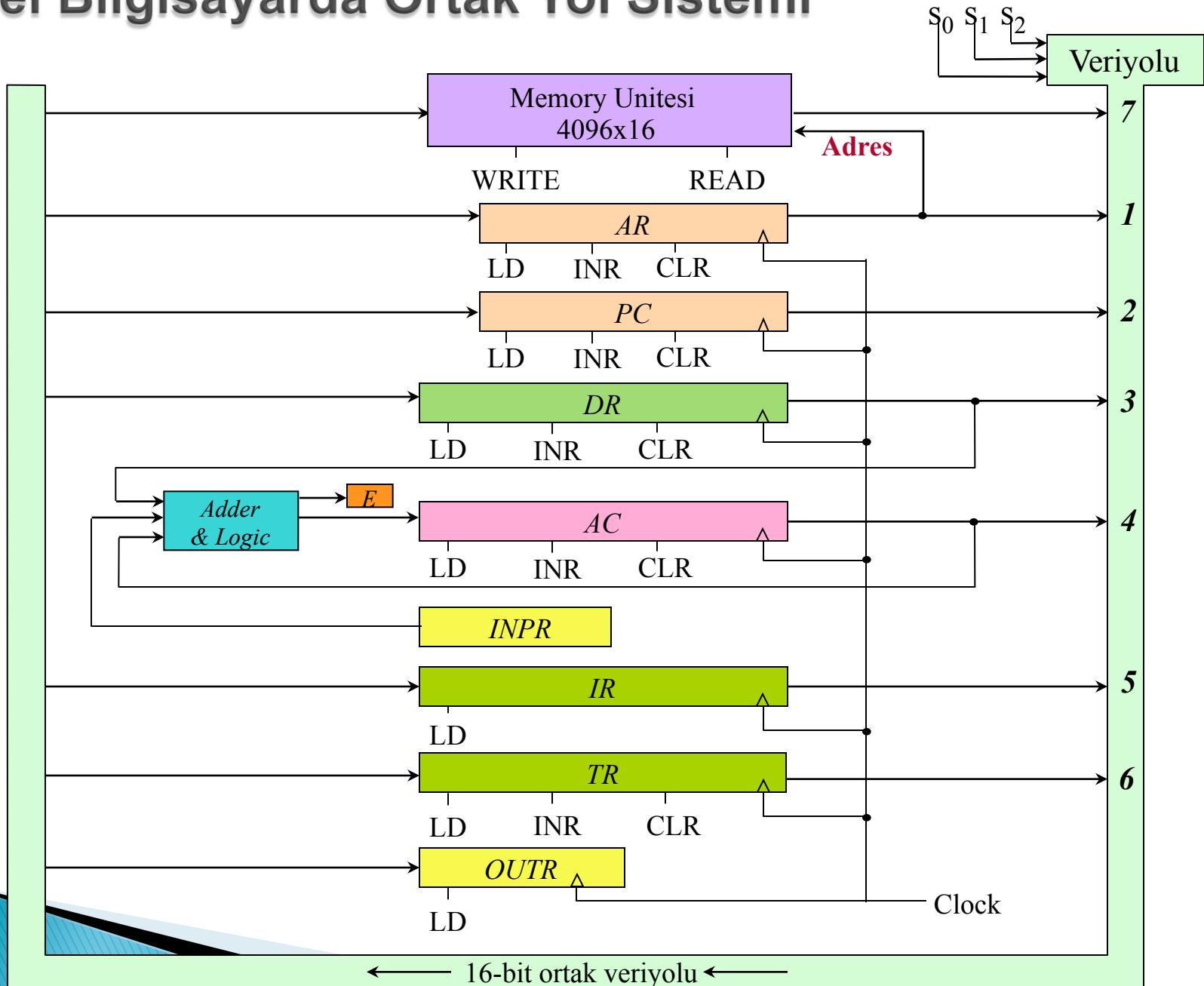
- ▶ Giriş/Çıkış Komutları ve Kesmeler
- ▶ Bilgisayarın Tamamlanmış Tanımı
- ▶ Temel Bilgisayarın Tasarımı
- ▶ Kontrol Devreleri
- ▶ ALU Birimi Tasarımı

Temel Bilgisayar Yapısı ve Devreleri

Temel Bilgisayar Organizasyonu ve Tasarımı

- ▶ Temel Bilgisayarın Tasarımı
- ▶ Kontrol Devreleri
 - Saklayıcı / Bellek kontrol devreleri/ifadeleri
 - Bayrak – FF kontrol devreleri
 - Ortak yol kontrol devreleri
 - AC devresi ve kontrol devresi
- ▶ ALU Birimi Tasarımı
 - ALU devresi

Temel Bilgisayarda Ortak Yol Sistemi

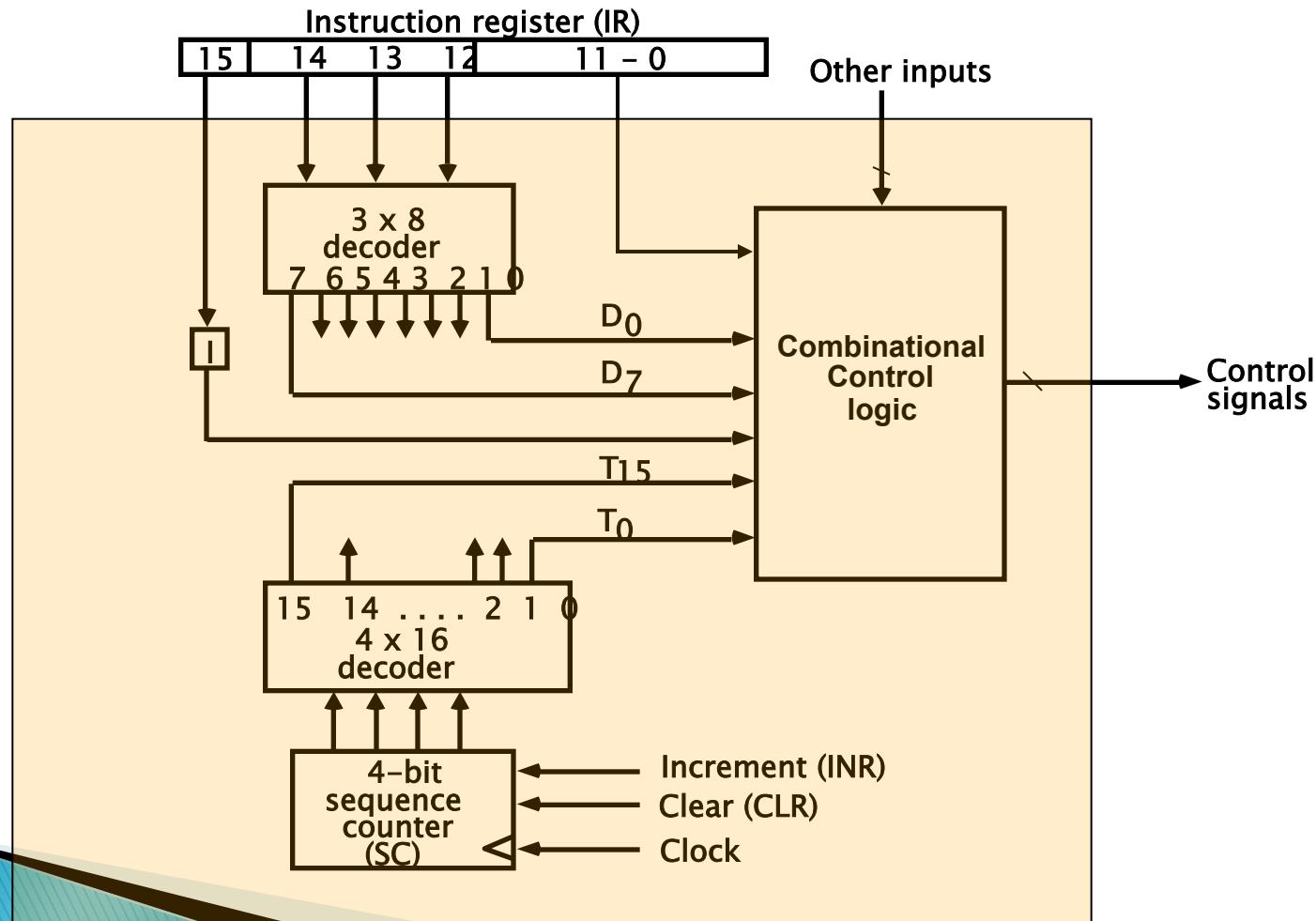


Temel Bilgisayar Komutları

Symbol	Hexadecimal code		Description
	$I = 0$	$I = 1$	
AND	0xxx	8xxx	AND memory word to <i>AC</i>
ADD	1xxx	9xxx	Add memory word to <i>AC</i>
LDA	2xxx	Axxx	Load memory word to <i>AC</i>
STA	3xxx	Bxxx	Store content of <i>AC</i> in memory
BUN	4xxx	Cxxx	Branch unconditionally
BSA	5xxx	Dxxx	Branch and save return address
ISZ	6xxx	Exxx	Increment and skip if zero
CLA	7800		Clear <i>AC</i>
CLE	7400		Clear <i>E</i>
CMA	7200		Complement <i>AC</i>
CME	7100		Complement <i>E</i>
CIR	7080		Circulate right <i>AC</i> and <i>E</i>
CIL	7040		Circulate left <i>AC</i> and <i>E</i>
INC	7020		Increment <i>AC</i>
SPA	7010		Skip next instruction if <i>AC</i> positive
SNA	7008		Skip next instruction if <i>AC</i> negative
SZA	7004		Skip next instruction if <i>AC</i> zero
SZE	7002		Skip next instruction if <i>E</i> is 0
HLT	7001		Halt computer
INP	F800		Input character to <i>AC</i>
OUT	F400		Output character from <i>AC</i>
SKI	F200		Skip on input flag
SKO	F100		Skip on output flag
ION	F080		Interrupt on
IOF	F040		Interrupt off

Zamanlama ve Kontrol

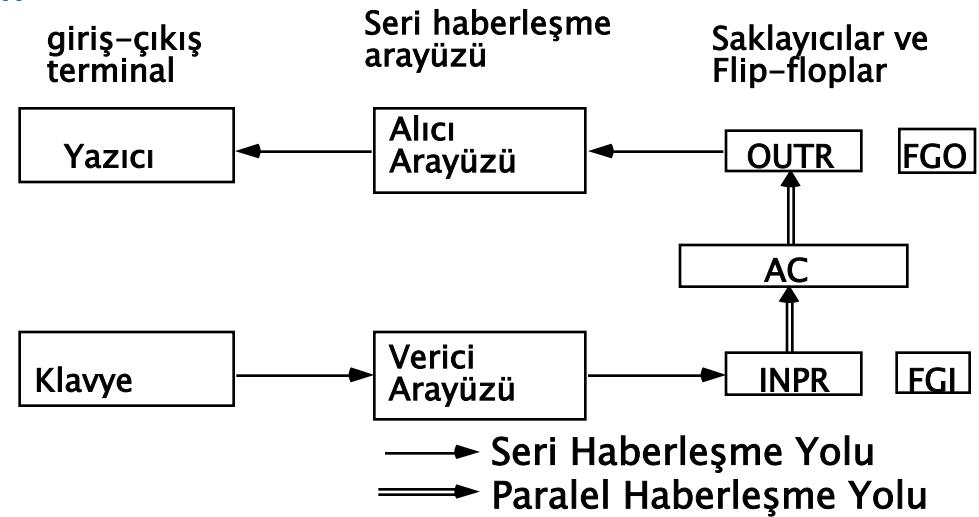
Temel Bilgisayarın Kontrol Ünitesi



Giriş/Çıkış ve Kesme

Bir Klavye ve bir Yazıcıdan oluşan Terminal

- Giriş-Çıkış Terminal Sistemi



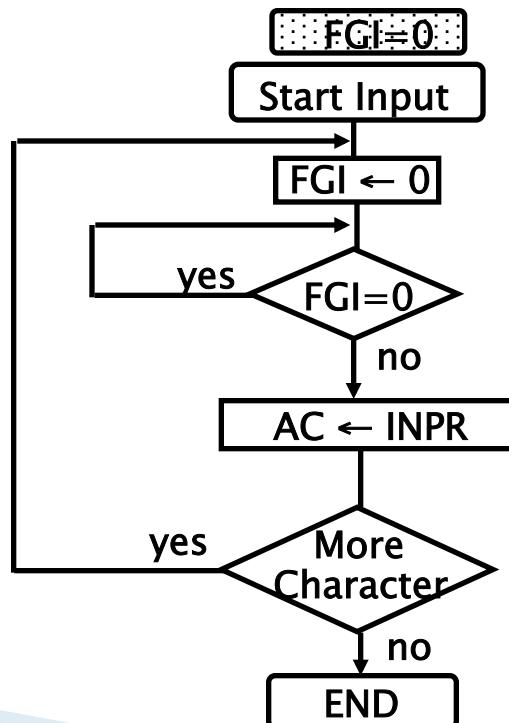
- Terminal seri bilgi gönderir ve alır.
- Klavyeden alınan seri bilgi INPR saklayıcısına ötelenir.
- Yazıcıya gönderilecek seri bilgi OUTR saklayıcısında tutulur.
- INPR ve OUTR terminallerle seri, AC ile paralel haberleşir.
- Bayraklar (flags) I/O elemanları ve bilgisayar arasındaki zamanlama farklarını senkronize etmek için kullanılırlar.

Yazılımsal Kontrollü Data Transferi

```
/* Data Girişi  
/* FGI = 0 (başl. değer)
```

loop: If FGI = 0 goto loop

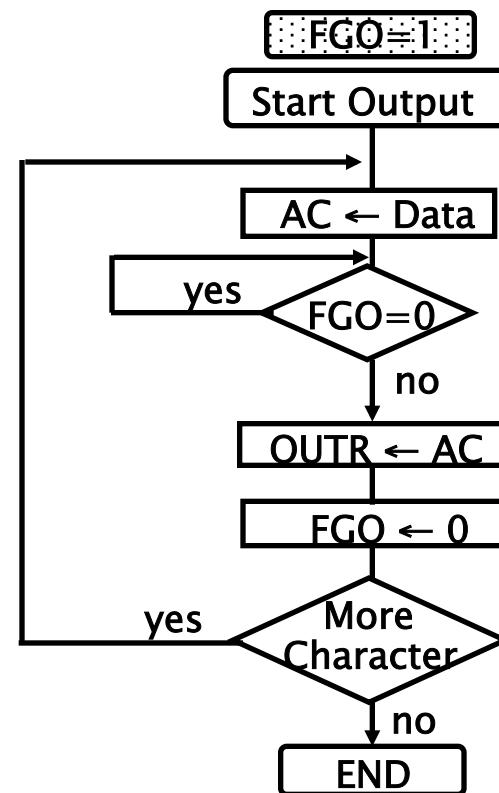
$AC \leftarrow INPR$, $FGI \leftarrow 0$



```
/* Data Çıkışı  
/* FGO = 1 (başl. değer)
```

loop: If FGO = 0 goto loop

$OUTR \leftarrow AC$, $FGO \leftarrow 0$



Giriş-Çıkış Komutları

$D_7IT_3 = p$

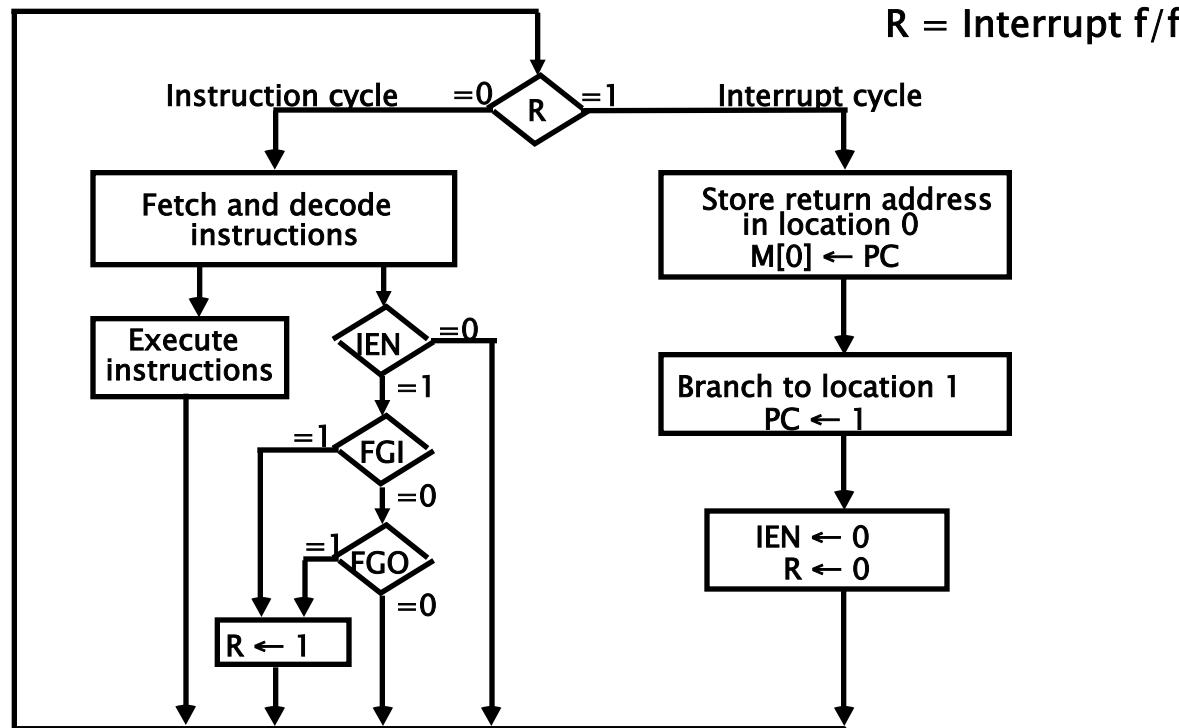
$IR(i) = B_i, i = 6, \dots, 11$

INP	$pB_{11}: AC(0-7) \leftarrow INPR, FGI \leftarrow 0$	AC ye karakter gir
OUT	$pB_{10}: OUTR \leftarrow AC(0-7), FGO \leftarrow 0$	AC den karakter çıkar
SKI	$pB_9: if(FGI = 1) \text{ then } (PC \leftarrow PC + 1)$	Giriş Bayrağına göre Atla
SKO	$pB_8: if(FGO = 1) \text{ then } (PC \leftarrow PC + 1)$	Çıkış Bayrağına göre Atla
ION	$pB_7: IEN \leftarrow 1$	Kesme açık
IOF	$pB_6: IEN \leftarrow 0$	Kesme kapalı

Kesme Kontrollü Giriş/Çıkış

- Kesme: Bazı dataların bilgisayara açık haberleşme ile girilmesi
 - I/O arayüzü, CPU yerine I/O elemanını görüntüler.
 - Arayüz I/O elemanını data transfer için hazır bulunduğuanda, CPU ya kesme isteğini bildirir.
 - Bir kesme isteği alındığında, işlemci geçici olarak yapmakta olduğu görevi durdurur, data transferi gerçeklemek üzere kesme servis rutine gider ve daha sonra yapmakta olduğu eski görevi geri döner.
-
- IEN (kesme izin flip-flop)
 - Komutlarla sıfırlanabilir veya birlenebilir.
 - Sıfırlanırsa, bilgisayar kesmeye kapalıdır.

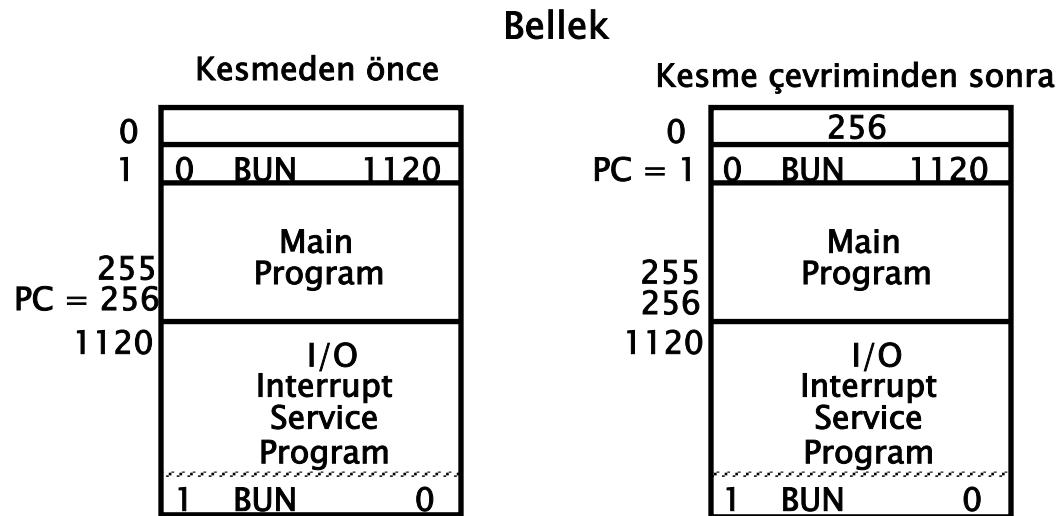
Kesme Çevrimi Akışı



Kesme Çevrimi dallan ve dönüş adresini sakla işleminin donanımsal gerçeklemesidir. Bir sonraki komut çevriminin başlangıcında, bellekten okunan komut 1 nolu adreste edir.

- 1 nolu bellek adresinde, programcı kesme servis rutin programına kontrolü yönlendiren bir dallanma komutu saklamalıdır.
- Orijinal programa kontrolü yönlendiren komut “(dolaylı) I BUN 0“ komutudur.

Kesme Çevrimi Servis Programı



- Programcı önceden kesmeye karşılık bir hizmet programını 1120 bellek adresinden başlayarak yerleştirmiştir.
 - R=1 durumunda kontrol kesme sürecine girer.
 - PC'daki 256 adresi belleğin 0 adresine yazılır. PC'a 1 adresi yazılır ve R=0 yapılır.
- Bir sonraki komut sürecinin başında 1 bellek adresindeki komut, kontrolü 1120 adresindeki I/O kesme servis programına taşır.
 - Bu program bayrakları kontrol edip hangi bayrağın set edildiğini bulur ve bayrağın gösterdiği I/O işlemini yapar.
- ION komutu ile IEN=1 yapılarak sonraki kesmeler izinlendirilir.
- Program kesmenin geldiği adrese döner.

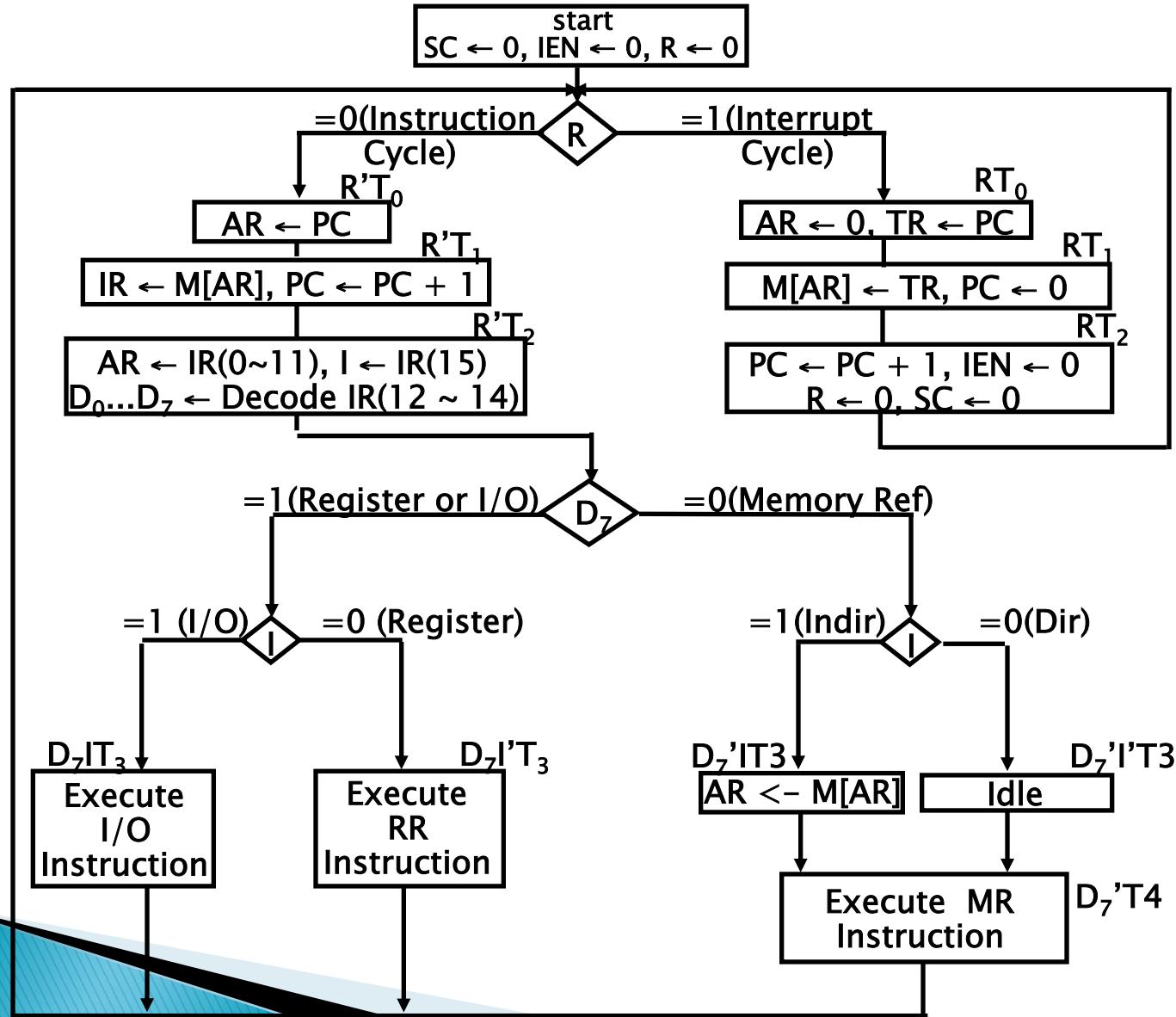
Kesme Çevriminde Saklayıcı Transfer İşlemleri

RTL ifadeleri (Kesme Çevrimi)

- $R \ F/F \leftarrow 1 \quad \text{if } IEN \ (FGI + FGO)T_0'T_1'T_2'$
 $\Leftrightarrow T_0'T_1'T_2' \ (IEN)(FGI + FGO): \ R \leftarrow 1$
- Komut çevriminin fetch ve decode aşamaları değiştirilmelidir:
 T_0, T_1, T_2 yerine $R'T_0, R'T_1, R'T_2$
- Kesme Çevrimi
 - $RT_0: \ AR \leftarrow 0, \ TR \leftarrow PC$
 - $RT_1: \ M[AR] \leftarrow TR, \ PC \leftarrow 0$
 - $RT_2: \ PC \leftarrow PC + 1, \ IEN \leftarrow 0, \ R \leftarrow 0, \ SC \leftarrow 0$

Bilgisayarın Tamamlanmış Tanımı

İşlemlerin Akış Şeması



Tüm Bilgisayar Tanımlama

Kontrol fonksiyonu ve Mikroişlemler

Fetch	R'T0:	$AR \leftarrow PC$
	R'T1:	$IR \leftarrow M[AR], PC \leftarrow PC + 1$
Decode	R'T2:	$D0, \dots, D7 \leftarrow \text{Decode } IR(12 \sim 14), AR \leftarrow IR(0 \sim 11), I \leftarrow IR(15)$
Indirect Interrupt	D7'IT3:	$AR \leftarrow M[AR]$
	T0'T1'T2'(IEN)(FGI + FGO):	$R \leftarrow 1$
	RT0:	$AR \leftarrow 0, TR \leftarrow PC$
	RT1:	$M[AR] \leftarrow TR, PC \leftarrow 0$
	RT2:	$PC \leftarrow PC + 1, IEN \leftarrow 0, R \leftarrow 0, SC \leftarrow 0$
Memory-Reference		
AND	D0T4:	$DR \leftarrow M[AR]$
	D0T5:	$AC \leftarrow AC \wedge DR, SC \leftarrow 0$
ADD	D1T4:	$DR \leftarrow M[AR]$
	D1T5:	$AC \leftarrow AC + DR, E \leftarrow Cout, SC \leftarrow 0$
LDA	D2T4:	$DR \leftarrow M[AR]$
	D2T5:	$AC \leftarrow DR, SC \leftarrow 0$
STA	D3T4:	$M[AR] \leftarrow AC, SC \leftarrow 0$
BUN	D4T4:	$PC \leftarrow AR, SC \leftarrow 0$
BSA	D5T4:	$M[AR] \leftarrow PC, AR \leftarrow AR + 1$
	D5T5:	$PC \leftarrow AR, SC \leftarrow 0$
ISZ	D6T4:	$DR \leftarrow M[AR]$
	D6T5:	$DR \leftarrow DR + 1$
	D6T6:	$M[AR] \leftarrow DR, \text{ if}(DR=0) \text{ then } (PC \leftarrow PC + 1), SC \leftarrow 0$

Tüm Bilgisayar Tanımlama Kontrol fonksiyonu ve Mikroişlemler

Register-Reference

	D7I'T3 = r	(tüm saklayıcı referanslı komutlar için ortak)
	IR(i) = Bi	(i = 0,1,2, ..., 11)
	r:	SC \leftarrow 0
CLA	rB11:	AC \leftarrow 0
CLE	rB10:	E \leftarrow 0
CMA	rB9:	AC \leftarrow AC'
CME	rB8:	E \leftarrow E'
CIR	rB7:	AC \leftarrow shr AC, AC(15) \leftarrow E, E \leftarrow AC(0)
CIL	rB6:	AC \leftarrow shl AC, AC(0) \leftarrow E, E \leftarrow AC(15)
INC	rB5:	AC \leftarrow AC + 1
SPA	rB4:	If(AC(15) = 0) then (PC \leftarrow PC + 1)
SNA	rB3:	If(AC(15) = 1) then (PC \leftarrow PC + 1)
SZA	rB2:	If(AC = 0) then (PC \leftarrow PC + 1)
SZE	rB1:	If(E=0) then (PC \leftarrow PC + 1)
HLT	rB0:	S \leftarrow 0

Input-Output

	D7IT3 = p	(tüm I/O referanslı komutlar için ortak)
	IR(i) = Bi	(i = 6,7,8,9,10,11)
	p:	SC \leftarrow 0
INP	pB11:	AC(0-7) \leftarrow INPR, FGI \leftarrow 0
OUT	pB10:	OUTR \leftarrow AC(0-7), FGO \leftarrow 0
SKI	pB9:	If(FGI=1) then (PC \leftarrow PC + 1)
SKO	pB8:	If(FGO=1) then (PC \leftarrow PC + 1)
ION	pB7:	IEN \leftarrow 1
IOF	pB6:	IEN \leftarrow 0

Tüm Bilgisayar Tanımlama

Mikroişlemler

- Bu RTL'ler bilgisayarın iç yapısını özetlemektedir.
- Lojik devrelerinin tasarıımı için gerekli tüm bilgileri de vermektedir.
- Kontrol ve şartlı kontrol deyimleri de kapılar için gerekli Boole fonksiyonlarını bulmak için yararlıdır.
- Bu kapılar kontrol biriminin bir kısmını oluşturur.
- Mikroişlemlerin listesi, bellek ve saklayıcılar için gerekli kontrol girişlerinin tiplerini belirler.
- RTL, sayısal bir sistemin iç tasarımının tanımlanmasının yanında bunun tasarımı için gerekli devrelerin belirlenmesinde de yararlıdır.

Temel Bilgisayarın Tasarımı

TB'nin DONANIMSAL BİLEŞENLERİ

Bir bellek ünitesi: 4096 x 16.

Saklayıcılar:

AR, PC, DR, AC, IR, TR, OUTR, INPR, ve SC

Flip-Floplar (Durum bilgilerini tutarlar):

I, S, E, R, IEN, FGI ve FGO

Kod-çözüçüler: 3x8 Opkod kod-çözücü
4x16 Zamanlama kod-çözücü

Ortak Yol: 16-bit

Kontrol Lojik Kapıları

ALU : AC ye bağlı lojik devre

Kontrol Lojik Kapıları

- 9 adet saklayıcının giriş kontrolleri
- Bellek Oku/Yaz Kontrolleri
- Flip-Flop, Set, Clear veya Complement Kontrolleri
- Ortak Yola saklayıcı seçenek kontroller S_2, S_1, S_0
- AC ve ALU

Saklayıcı ve Bellek Kontrolleri

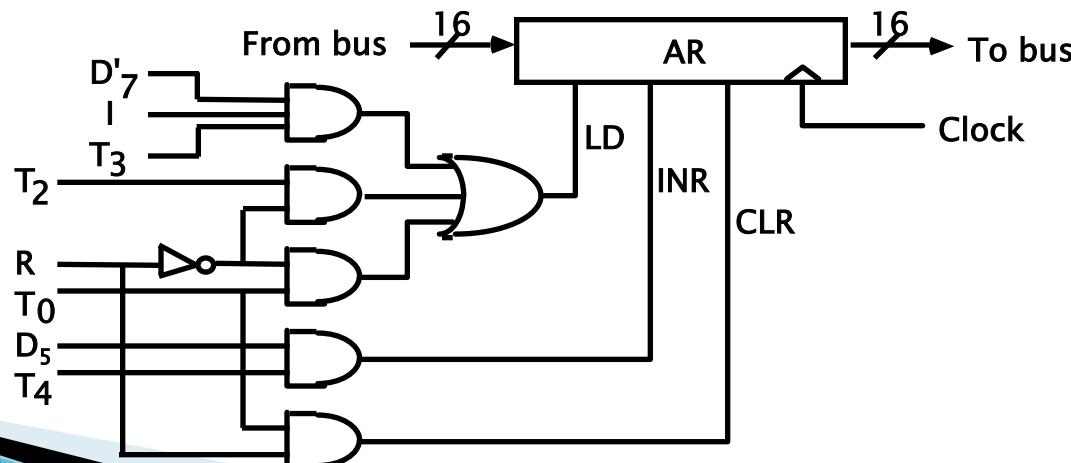
Adres Saklayıcı; AR

AR nin içeriğini etkiyen tüm RTL ifadeleri:

$R'T_0$:	$AR \leftarrow PC$	$LD(AR)$
$R'T_2$:	$AR \leftarrow IR(0-11)$	$LD(AR)$
D'_7IT_3 :	$AR \leftarrow M[AR]$	$LD(AR)$
RT_0 :	$AR \leftarrow 0$	$CLR(AR)$
D_5T_4 :	$AR \leftarrow AR + 1$	$INR(AR)$



$$LD(AR) = R'T_0 + R'T_2 + D'_7IT_3$$
$$CLR(AR) = RT_0$$
$$INR(AR) = D_5T_4$$



Saklayıcı ve Bellek Kontrolleri

Örneğin, bellek oku işlemi $\leftarrow M[AR]$ sembolüyle gösterildiğinden;

Bellek oku girişi:

$$\text{READ} = R'T_1 + D_7'IT_3 + (D_0 + D_1 + D_2 + D_6)T_4$$

WRITE = ?

Bayrakların Kontrolü

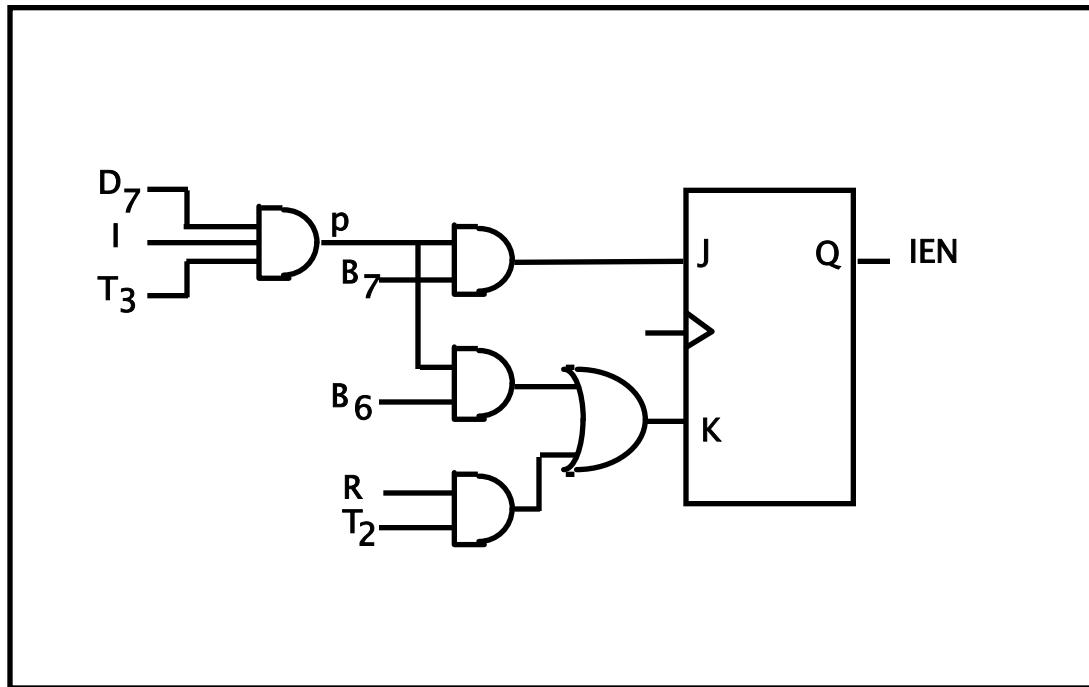
IEN: Kesme İzin Bayrağı

pB7: IEN \leftarrow 1 (I/O Komutu)

pB6: IEN \leftarrow 0 (I/O Komutu)

RT₂: IEN \leftarrow 0 (Kesme)

p = D₇IT₃ (Giriş/Çıkış Komutu)



IF ifadeli komut

Temel bilgisayara ait olan PC saklayıcısına ilişkin kontrol işaretleri ve *RTL* ifadeleri

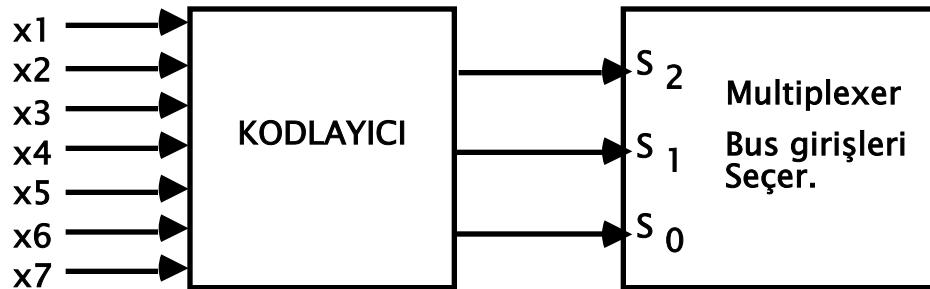
$$Z_{DR} = 1 \text{ if } DR=0; Z_{AC} = 1 \text{ if } AC=0$$

$$\begin{aligned} INC(PC) = & R'T_1 + RT_2 + D_6T_6Z_{DR} + pB_9(FGI) + pB_8(FCI) \\ & + rB_4(AC_{15})' + rB_3(AC_{15}) + rB_2Z_{AC} + rB_1E' \end{aligned}$$

$$LD(PC) = D_4T_4 + D_5T_5$$

$$CLR(PC) = RT_1$$

Ortak Yol Kontrolü



x1	x2	x3	x4	x5	x6	x7	S2	S1	S0	seçilen saklayıcı
0	0	0	0	0	0	0	0	0	0	none
1	0	0	0	0	0	0	0	0	1	AR
0	1	0	0	0	0	0	0	1	0	PC
0	0	1	0	0	0	0	0	1	1	DR
0	0	0	1	0	0	0	1	0	0	AC
0	0	0	0	1	0	0	1	0	1	IR
0	0	0	0	0	1	0	1	1	0	TR
0	0	0	0	0	0	1	1	1	1	Memory

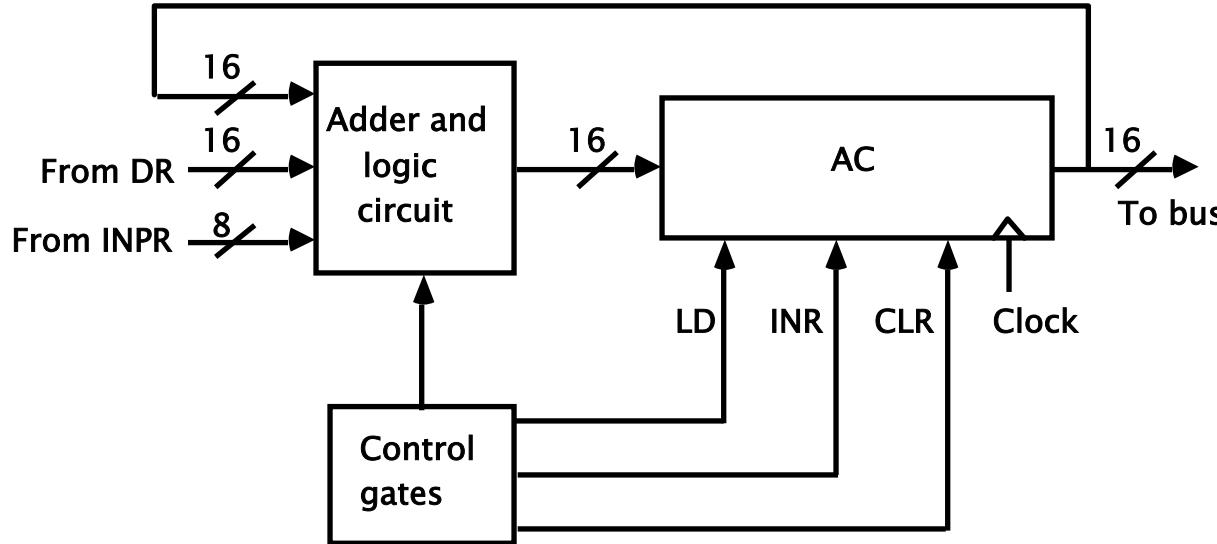
AR saklayıcısı için

D₄T₄: PC ← AR
D₅T₅: PC ← AR



$$x1 = D_4T_4 + D_5T_5$$

Akumulatör Lojik Devresi

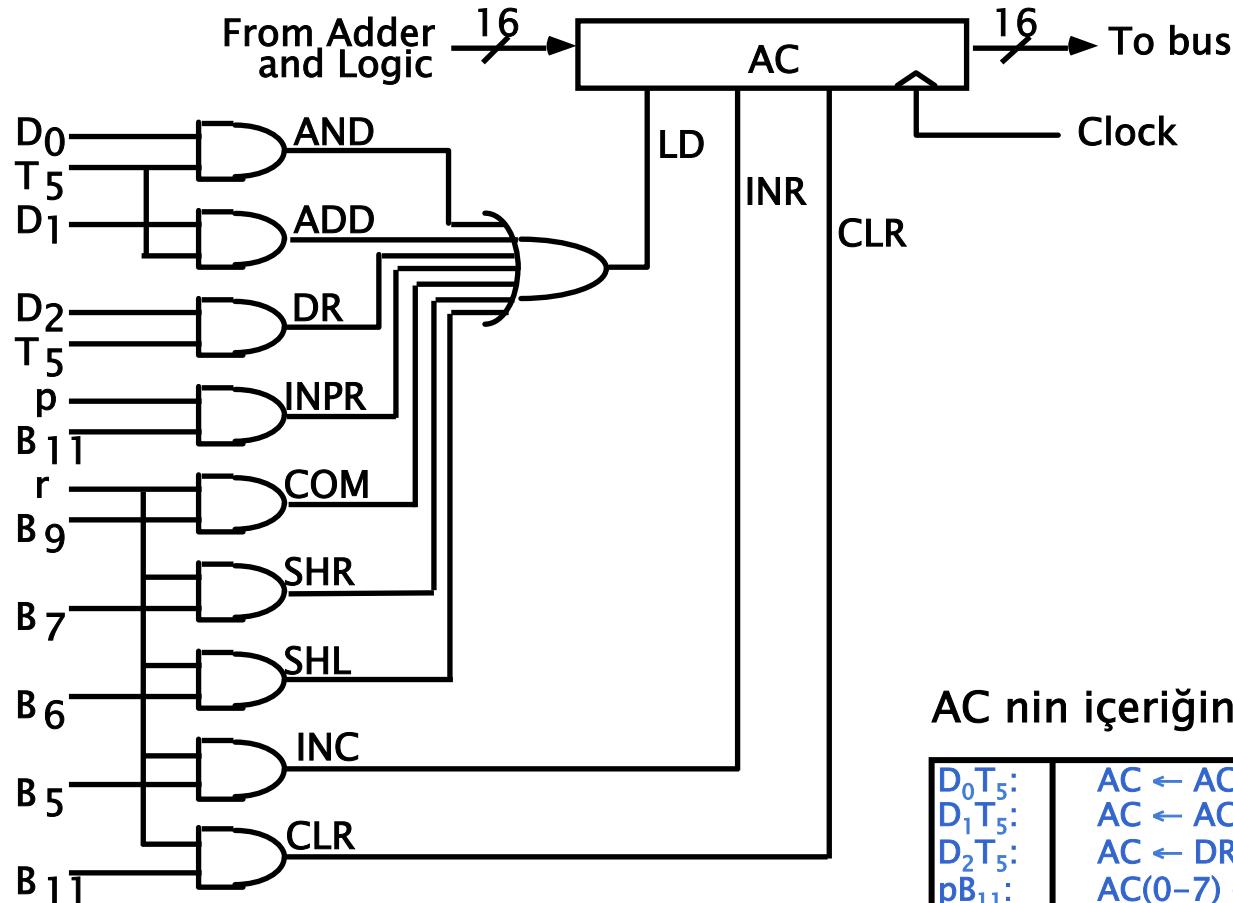


AC nin içeriğini etkileyen tüm RTL ifadeleri:

$D_0 T_5 :$	$AC \leftarrow AC \wedge DR$	AND with DR
$D_1 T_5 :$	$AC \leftarrow AC + DR$	Add with DR
$D_2 T_5 :$	$AC \leftarrow DR$	Transfer from DR
$pB_{11} :$	$AC(0-7) \leftarrow INPR$	Transfer from INPR
$rB_9 :$	$AC \leftarrow AC'$	Complement
$rB_7 :$	$AC \leftarrow shr AC, AC(15) \leftarrow E$	Shift right
$rB_6 :$	$AC \leftarrow shl AC, AC(0) \leftarrow E$	Shift left
$rB_{11} :$	$AC \leftarrow 0$	Clear
$rB_5 :$	$AC \leftarrow AC + 1$	Increment

AC Saklayıcısı Kontrolü

LD, INR ve CLR kontrol kapıları

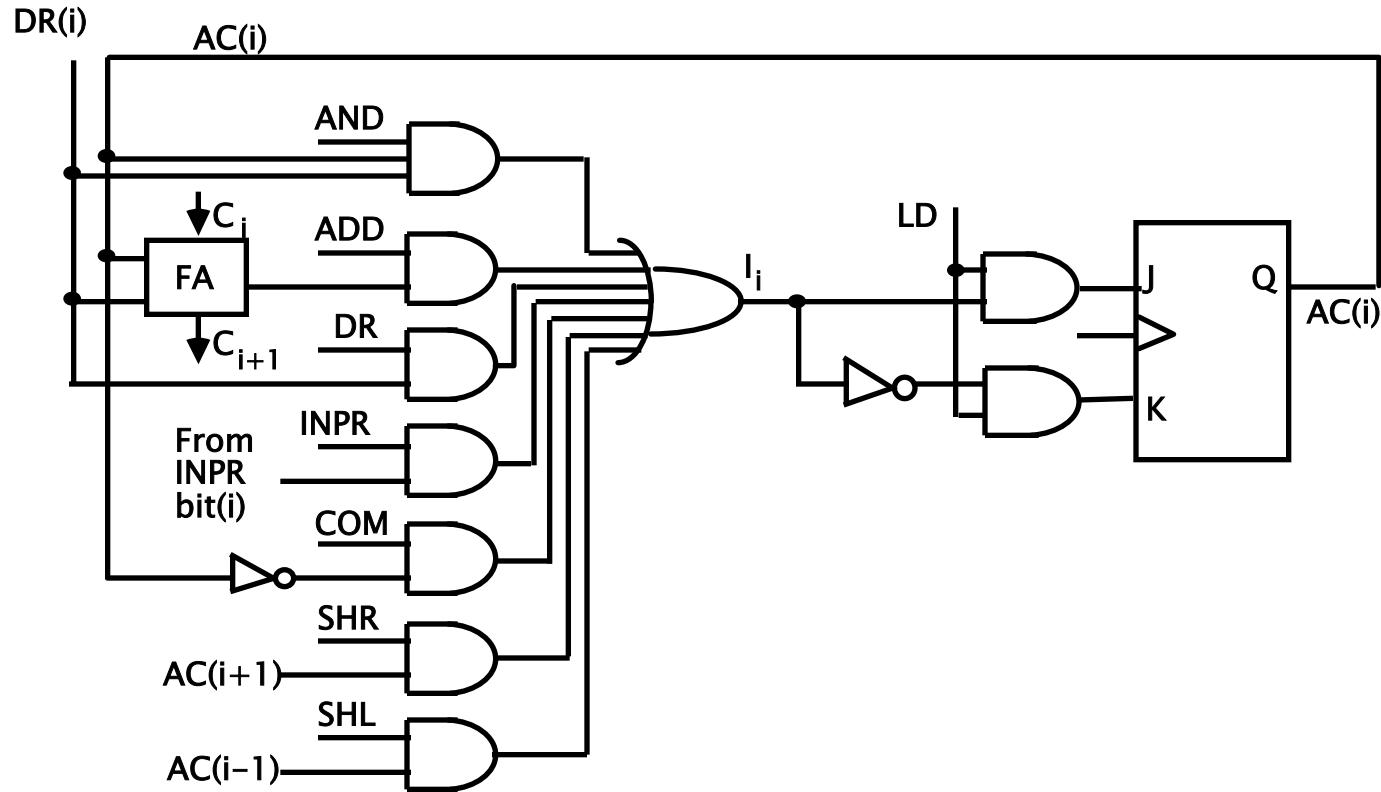


AC nin içeriğini etkileyen tüm RTL ifadeleri

$D_0 T_5:$	$AC \leftarrow AC \wedge DR$	AND with DR
$D_1 T_5:$	$AC \leftarrow AC + DR$	Add with DR
$D_2 T_5:$	$AC \leftarrow DR$	Transfer from DR
$p B_{11}:$	$AC(0-7) \leftarrow INPR$	Transfer from INPR
$r B_9:$	$AC \leftarrow AC'$	Complement
$r B_7:$	$AC \leftarrow shr AC, AC(15) \leftarrow E$	Shift right
$r B_6:$	$AC \leftarrow shl AC, AC(0) \leftarrow E$	Shift left
$r B_{11}:$	$AC \leftarrow 0$	Clear
$r B_5:$	$AC \leftarrow AC + 1$	Increment

ALU (Toplayıcı ve Lojik Devre)

ALU' nun bir katı (i. bit)



Paralel Yüklemeli Saatli Silmeli 4 Bit İkili Sayıcı

