

# BİLGİSAYAR ORGANİZASYONU ve TASARIMI

YRD. DOÇ. DR. FATİH KELEŞ

# **Temel Bilgisayar Yapısı ve Devreleri**

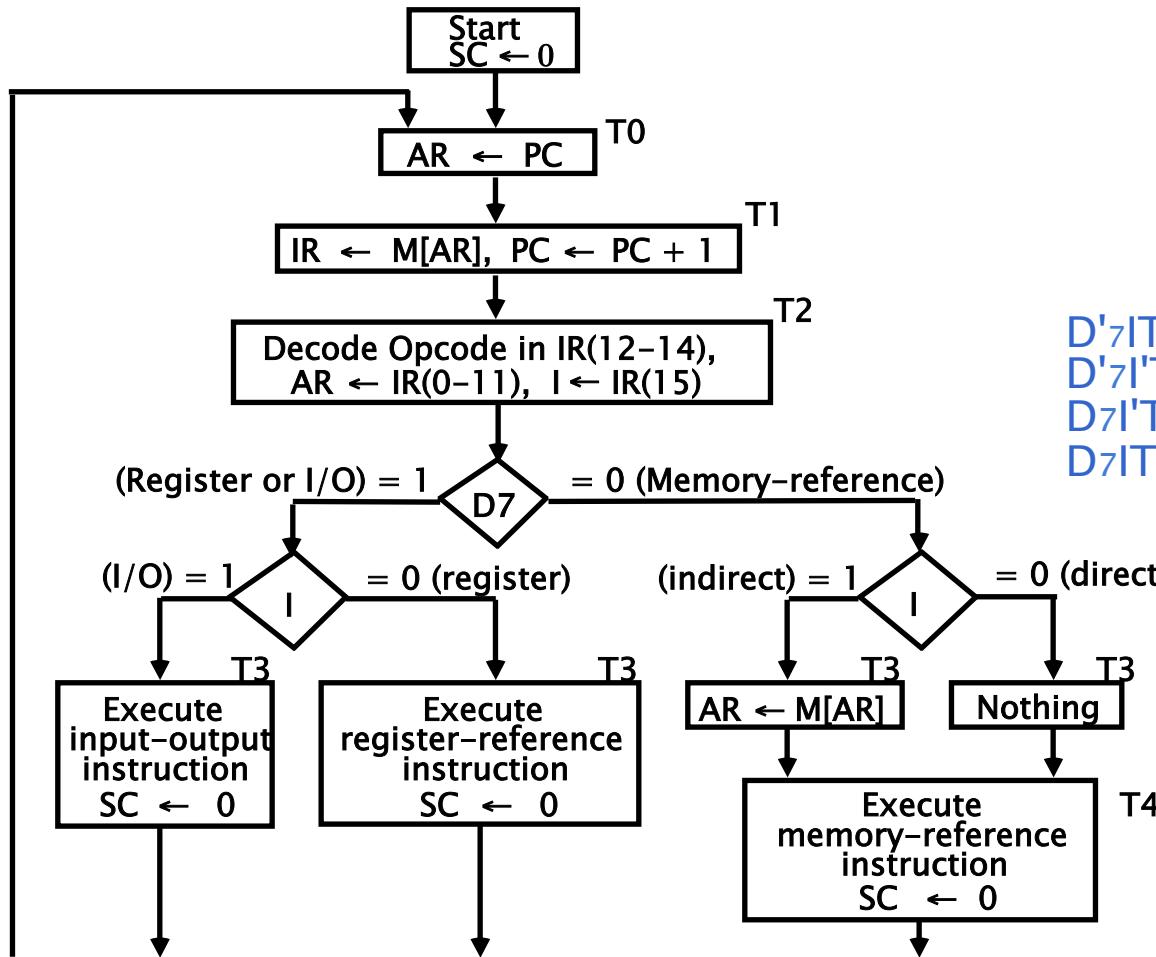
## **Temel Bilgisayar Organizasyonu ve Tasarımı**

- ▶ Komut Türleri
- ▶ Saklayıcı Adreslemeli Komutlar
- ▶ Bellek Adreslemeli Komutlar
- ▶ Giriş/Çıkış Komutları

# 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

# Komut Türünün Belirlenmesi Akışı



D<sub>7</sub>'IT<sub>3</sub>: AR ← M[AR]

D<sub>7</sub>I'T<sub>3</sub>: Hiçbir şey

D<sub>7</sub>I'T<sub>3</sub>: Saklayıcı referanslı komut icrası

D<sub>7</sub>IT<sub>3</sub>: I/O referanslı komut icrası.

IR(12-14)  
= 111

$\left\{ \begin{array}{l} D_7=1 \\ \quad \left\{ \begin{array}{l} \text{Register}(I=0) \rightarrow D_7I'T_3(\text{Execute}) \\ \text{I/O} \quad (I=1) \rightarrow D_7IT_3 (\text{Execute}) \end{array} \right. \\ D_7=0 : \text{Memory Ref.} \end{array} \right.$

Read effective  
Address

$\left\{ \begin{array}{l} \text{Indirect}(I=1) \rightarrow D_7'IT_3(AR \leftarrow M[AR]) \\ \text{Direct } (I=0) \rightarrow \text{nothing in } T_3 \end{array} \right.$

# Saklayıcı Referanslı Komutlar

Saklayıcı referanslı komutu belirleme

- $D_7 = 1, I = 0$
- Komut saklayıcının 0–11 bitleri --- IR(0–11) --- 12 komut
- T3 Zamanlama işaretleri ile icra

$r = D_7 \ I' \ T_3 \Rightarrow$  Saklayıcı referanslı komut ortak kontrol fonksiyonu  
 $B_i = IR(i), i=0,1,2,\dots,11$  komutları ayırtedici kontrol işaretleri

CLA	$r:$	$SC \leftarrow 0$
CLE	$rB_{11}:$	$AC \leftarrow 0$
CMA	$rB_{10}:$	$E \leftarrow 0$
CME	$rB_9:$	$AC \leftarrow AC'$
	$rB_8:$	$E \leftarrow E'$
CIR	$rB_7:$	$AC \leftarrow shr AC, AC(15) \leftarrow E, E \leftarrow AC(0)$
CIL	$rB_6:$	$AC \leftarrow shl AC, AC(0) \leftarrow E, E \leftarrow AC(15)$
INC	$rB_5:$	$AC \leftarrow AC + 1$
SPA	$rB_4:$	if $(AC(15) = 0)$ then $(PC \leftarrow PC+1)$
SNA	$rB_3:$	if $(AC(15) = 1)$ then $(PC \leftarrow PC+1)$
SZA	$rB_2:$	if $(AC = 0)$ then $(PC \leftarrow PC+1)$
SZE	$rB_1:$	if $(E = 0)$ then $(PC \leftarrow PC+1)$
HLT	$rB_0:$	$S \leftarrow 0$ ( $S$ is a start-stop flip-flop)

# Bellek Referanslı Komutlar

Symbol	Operation Decoder	Symbolic Description
AND	D <sub>0</sub>	$AC \leftarrow AC \wedge M[AR]$
ADD	D <sub>1</sub>	$AC \leftarrow AC + M[AR], E \leftarrow C_{out}$
LDA	D <sub>2</sub>	$AC \leftarrow M[AR]$
STA	D <sub>3</sub>	$M[AR] \leftarrow AC$
BUN	D <sub>4</sub>	$PC \leftarrow AR$
BSA	D <sub>5</sub>	$M[AR] \leftarrow PC, PC \leftarrow AR + 1$
ISZ	D <sub>6</sub>	$M[AR] \leftarrow M[AR] + 1, \text{if } M[AR] + 1 = 0 \text{ then } PC \leftarrow PC+1$

- Komut efektif adresi AR saklayıcısındadır ve
- T<sub>2</sub> işaretini boyunca (I = 0 iken), T<sub>3</sub> işaretini boyunca (I = 1 iken)
- Bellek ref. Komut icraları T4 ile başlar.



AND to AC

$$D_0 T_4: DR \leftarrow M[AR]$$

Operand oku  
AC içeriği ile AND le

$$D_0 T_5: AC \leftarrow AC \wedge DR, SC \leftarrow 0$$

ADD to AC

$$D_1 T_4: DR \leftarrow M[AR]$$

Operand oku

$$D_1 T_5: AC \leftarrow AC + DR, E \leftarrow C_{out}, SC \leftarrow 0$$

AC ile topla ve  
eldeyi E f.f. ta tut.

# Bellek Referanslı Komutlar

LDA: AC ye yükle

$D_2 T_4: DR \leftarrow M[AR]$

$D_2 T_5: AC \leftarrow DR, SC \leftarrow 0$

STA: AC yi sakla

$D_3 T_4: M[AR] \leftarrow AC, SC \leftarrow 0$

BUN: Şartsız dallan

$D_4 T_4: PC \leftarrow AR, SC \leftarrow 0$

BSA: Dallan ve Dönüş adresini sakla

$M[AR] \leftarrow PC, PC \leftarrow AR + 1$

# Bellek Referanslı Komutlar

BSA:

$D_5T_4: M[AR] \leftarrow PC, AR \leftarrow AR + 1$

$D_5T_5: PC \leftarrow AR, SC \leftarrow 0$

Bellek, PC, AR ( T4 zamanı )

20	0 BSA	135
PC = 21	Bir sonraki komut	
AR = 135	21(dönüş adr)	
136	Altprogram	

Bellek

İçradan sonraki Bellek, PC

20	0 BSA	135
PC = 136	Bir sonraki komut	
135	21(dönüş adr)	
1	Alt program	

Bellek

$D_5T_4 : M[135] \leftarrow 21(PC), 136(AR) \leftarrow 135 + 1$   
 $D_5T_5 : 136(PC) \leftarrow 136(AR), SC \leftarrow 0$

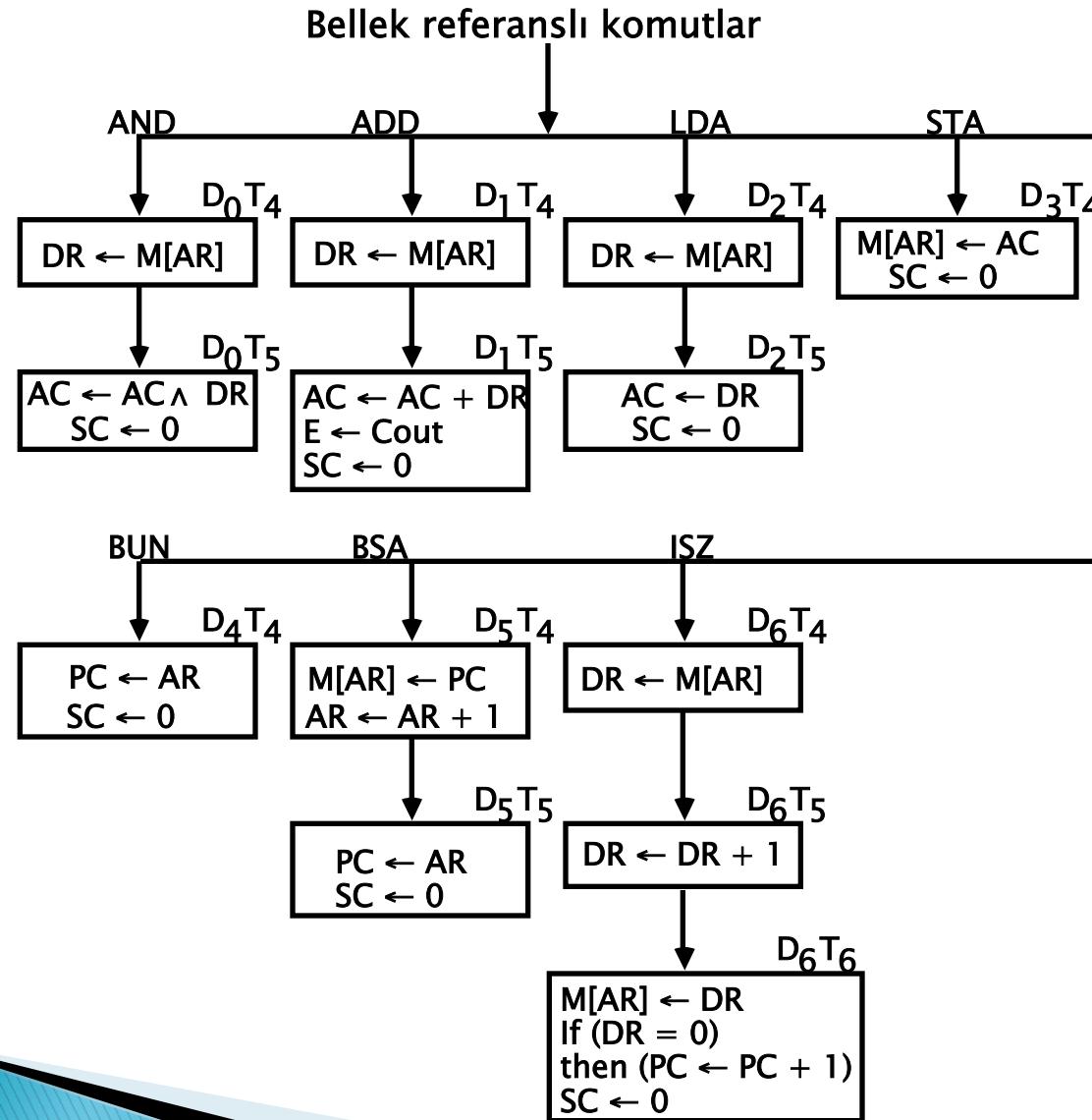
ISZ: Bellek içeriğini artır, içerik=0 ise komut atla

$D_6T_4: DR \leftarrow M[AR]$

$D_6T_5: DR \leftarrow DR + 1$

$D_6T_6: M[AR] \leftarrow DR, \text{ if } (DR = 0) \text{ then } (PC \leftarrow PC + 1), SC \leftarrow 0$

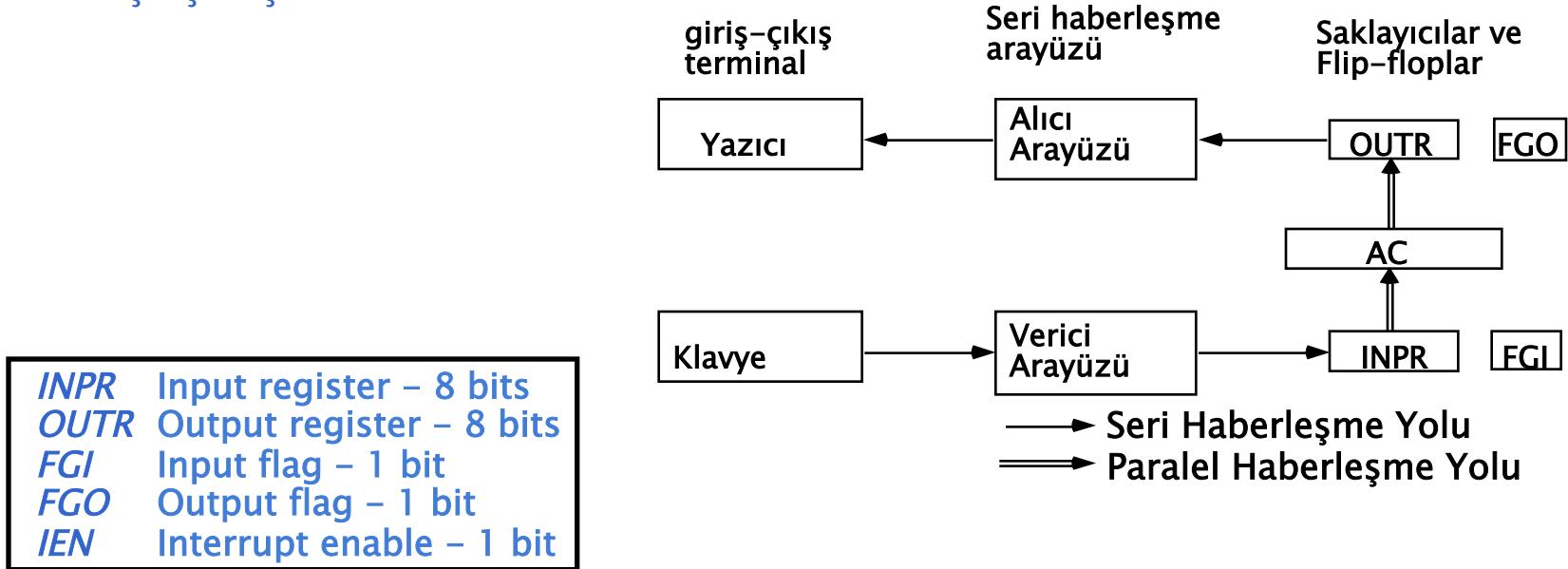
# Bellek Referanslı Komutlar Akışı



# 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.