

BİLGİSAYAR ORGANİZASYONU ve TASARIMI

DR. FATİH KELEŞ

Mikroprogramlanmış Kontrol (Yazılımsal Kontrol)

- ▶ Kontrol Belleği
- ▶ Mikrokomut Düzenleme
- ▶ Mikroprogram Örneği
- ▶ Kontrol Biriminin Tasarımı
- ▶ Mikrokomut Formatı

Giriş

▶ **Donanımsal Kontrol**

Kontrol işaretlerinin klasik lojik tasarım teknikleri ile donanımsal olarak üretilmesi

Örnek: Temel Bilgisayar Kontrol Ünitesi

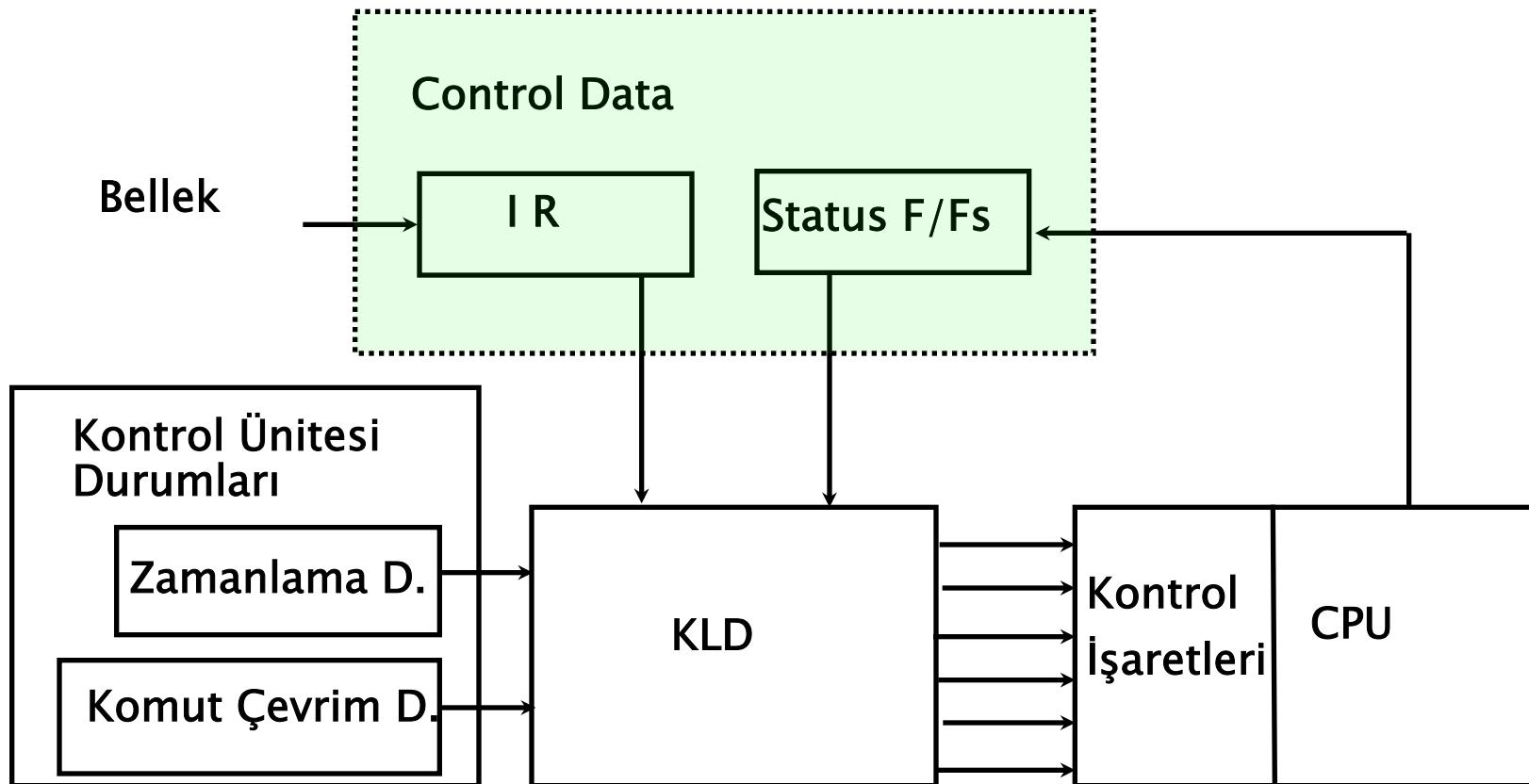
Yazılımsal Kontrol

Kontrol işaretlerinin bellekte saklanarak yazılımsal olarak üretilmesi

Örnek: Mikroprogramlanmış Kontrol Ünitesi
(Bu Bölümdeki Kontrol Ünitesi)

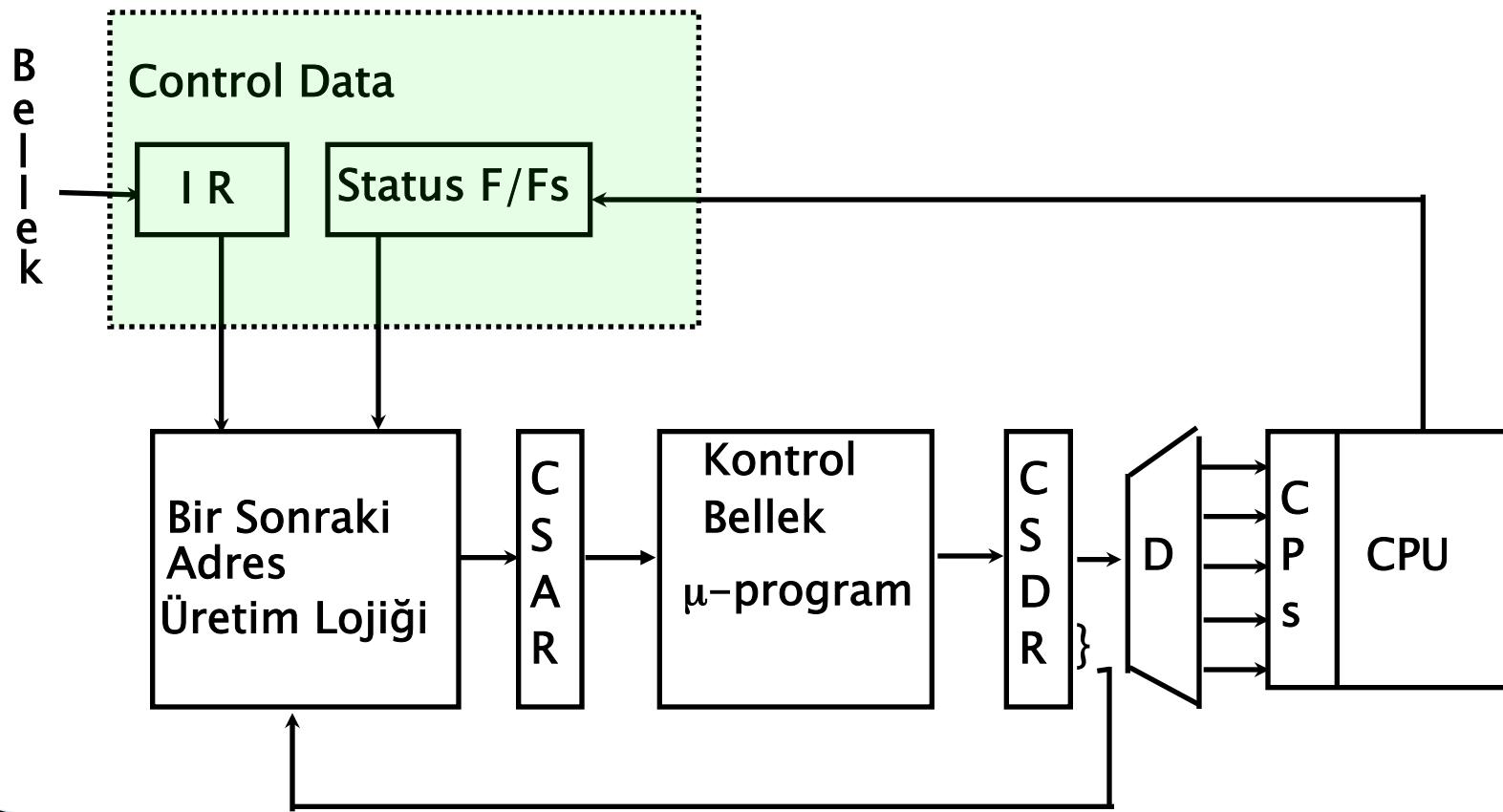
Donanımsal Kontrol Gerçekleme

Kombinezonsal Lojik Devreler



Yazılımsal Kontrol Birimi

Mikroprogram



Kontrol Belleği

- ▶ Kontrol biriminin en önemli görevi, **mikro işlemlerin sırasını başlatmaktadır.**
- ▶ Kontrol sinyalleri donanım tarafından kararlaştırılmış lojik tasarım teknikleri kullanılarak oluşturulduğunda, kontrol **donanımsal** olur.

Kontrol Belleği

- ▶ Bir kontrol biriminin ikili kontrol değişkenleri bellekte saklanır, mikro programlanmış kontrol birimi olarak adlandırılır.
- ▶ Kontrol belleğindeki her kelime bir mikro komut içerir.
- ▶ Kontrol belleği ROM olabilir.

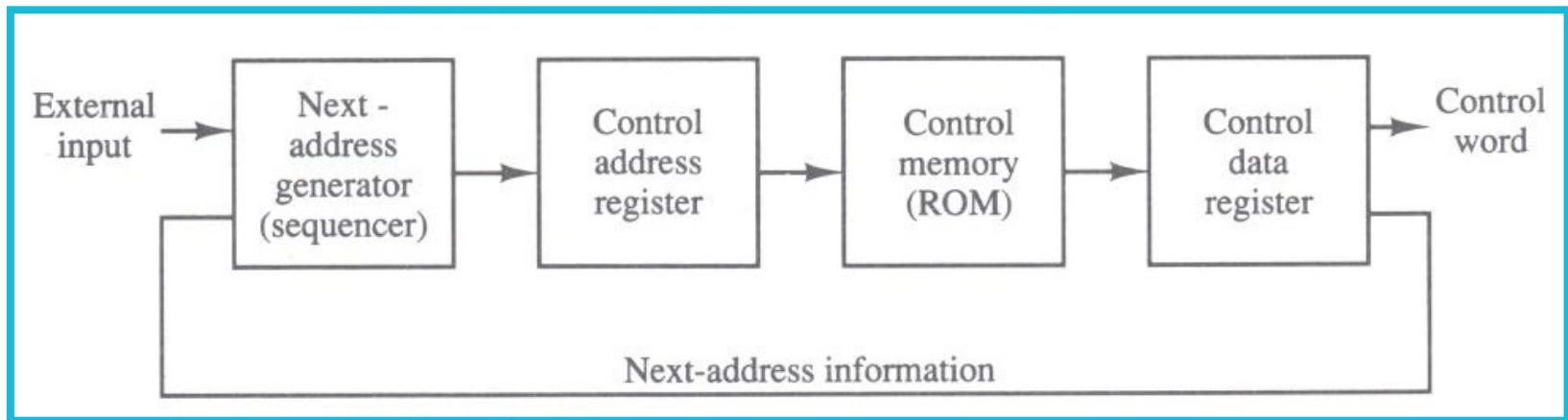
Kontrol Belleği

- ▶ Mikro programlanmış kontrol birimi içeren bilgisayarın iki ayrı belleği vardır; ana bellek ve kontrol belleği.
- ▶ Kontrol belleği, **kullanıcının istediği zaman değiştirmediği** sabit bir mikro programı elinde tutar.
- ▶ Her bir makine komutu, kontrol belleğinde bir dizi mikro komutları başlatır.

Kontrol Belleği

- ▶ Mikroişlemler bir araya gelerek bir komutu oluşturmaktadır. BUN, STA gibi komutlar kontrol belleğinde saklanmaktadır.

Mikroprogramlı Kontrol Organizasyonu



Tanımlar

- **Mikroprogram**

– Komut Kümesini doğru olarak yürütmek için gerekli tüm kontrol işaretlerini üreten bellekte saklı program.

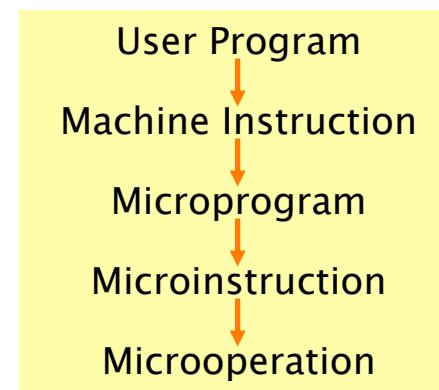
– Mikrokomutlardan oluşur.

- **Mikrokomutlar**

Kontrol kelimesi ve adres düzenleme bilgisi içerir.

Kontrol Kelimesi – Bir saat çevrimi için gerekli tüm kontrol bilgisidir.

Adres Düzenleme Kelimesi – Bir sonraki mikrokomut adresine karar vermek için gereken bilgidir.

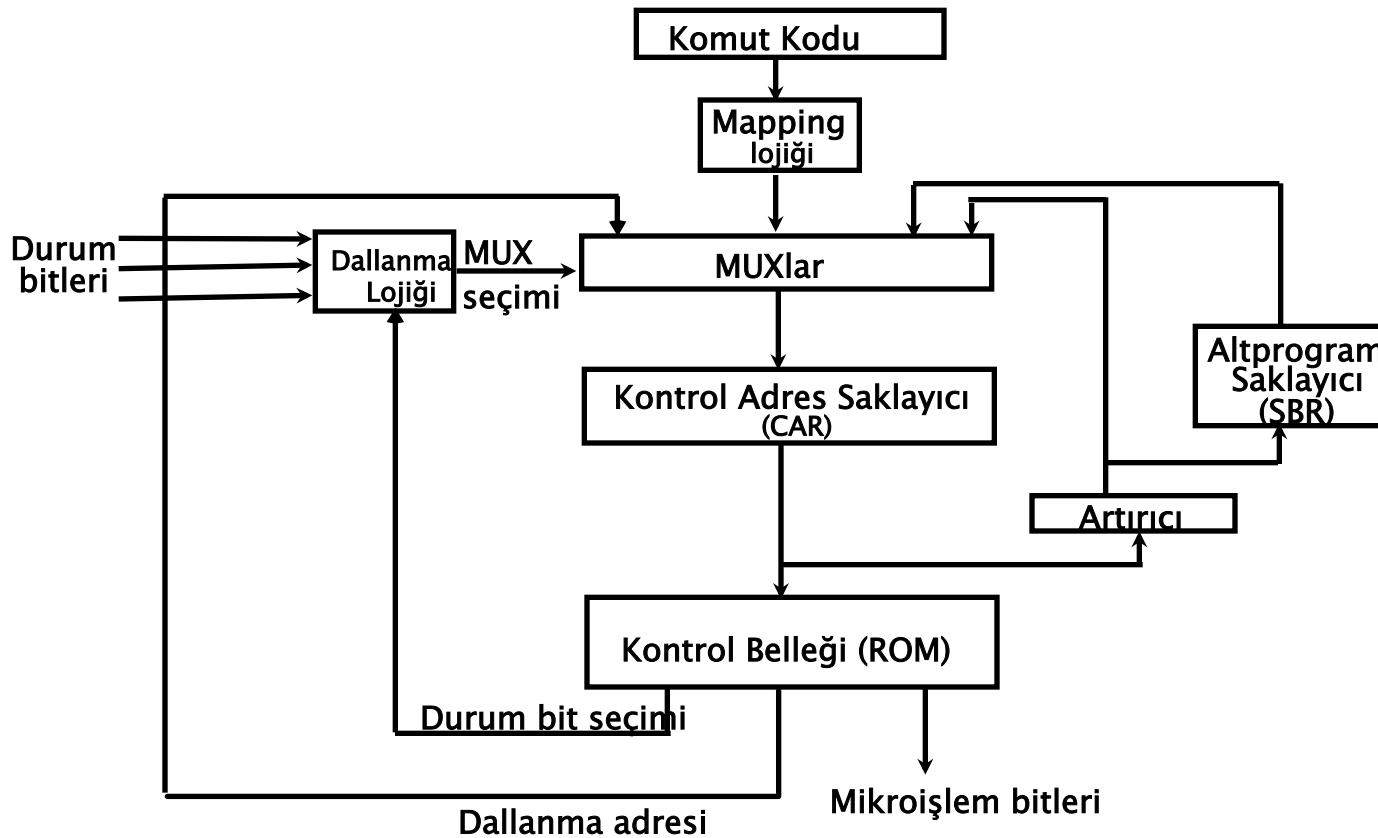


Mikroprogram Düzenleyici

Bir sonraki saat çevriminde yürütülecek Mikrokomut adresini belirleyen Mikroprogram Kontrol Birimi

- Sıradaki komut
- Dallanma
- Şartlı Dallanma
- Altprogram
- Döngü
- Komut OP Kodunu adrese dönüşümü (mapping)

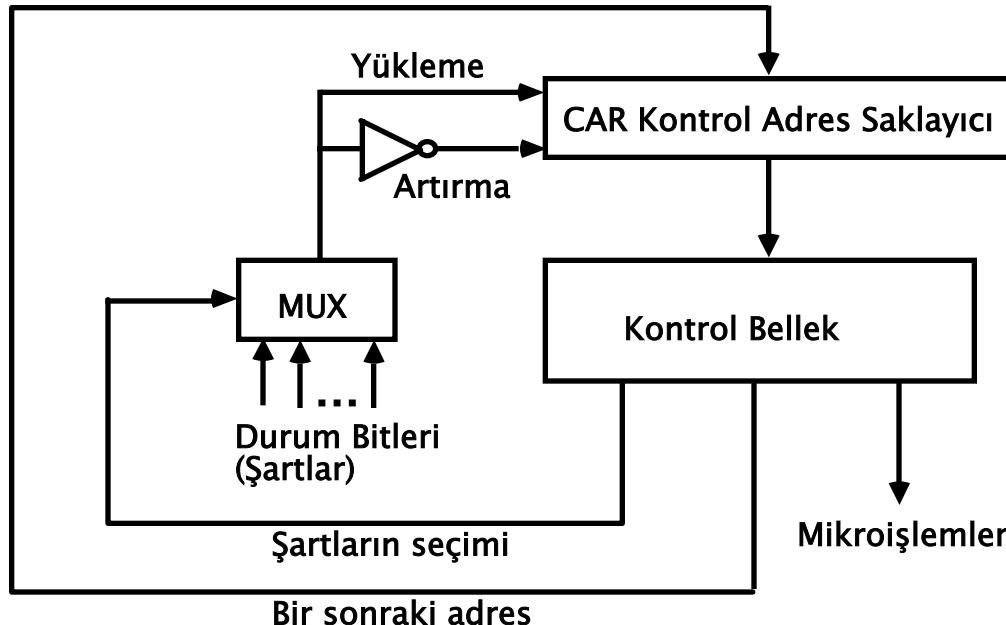
Mikrokomut Düzenleme



Adres Düzenleme Yeteneği

- CAR nin 1 artırılması ($CAR+1$) : bir sonraki adres
- Şartlı ve Şartsız Dallanma
- Komut opkod bilgisinin bir kontrol bellek adresine dönüşümü
- Altprogram çağrıma ve dönüş imkanı

Şartlı Dallanma



Şartlı Dallanma

Şart gerçekleşmişse Dallan (mikrokomutun içerdiği adrese)
aksi halde Bir sonraki adrese dallan

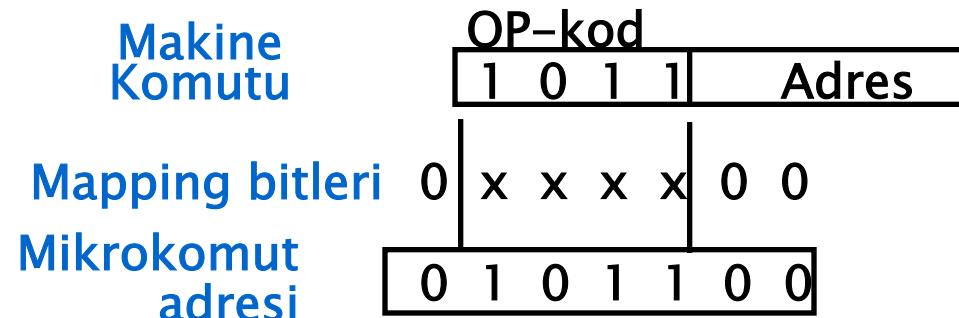
Test Şartları: O(Taşma), N(negatif),
Z(sıfır), C(elde),

Şartsız Dallanma

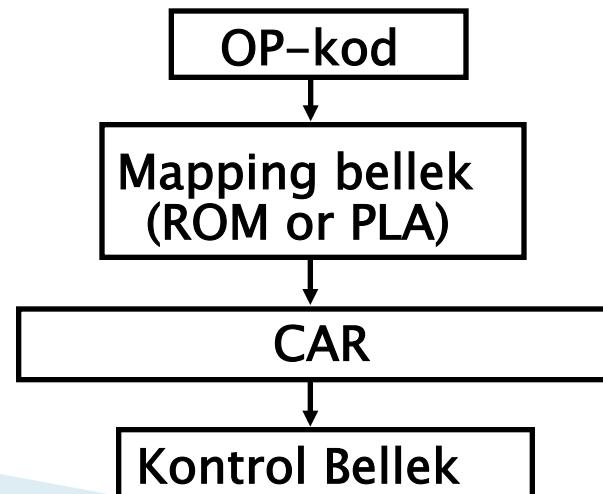
Mux girişindeki durum bitini 1 e sabitle

Yazılımsal Kontrol Mapping (Eşleme) İşlemi

Bir komutun işlem kodundan yazılımsal kontrol belleğindeki mikroprogramın başlangıç adresini bulmak

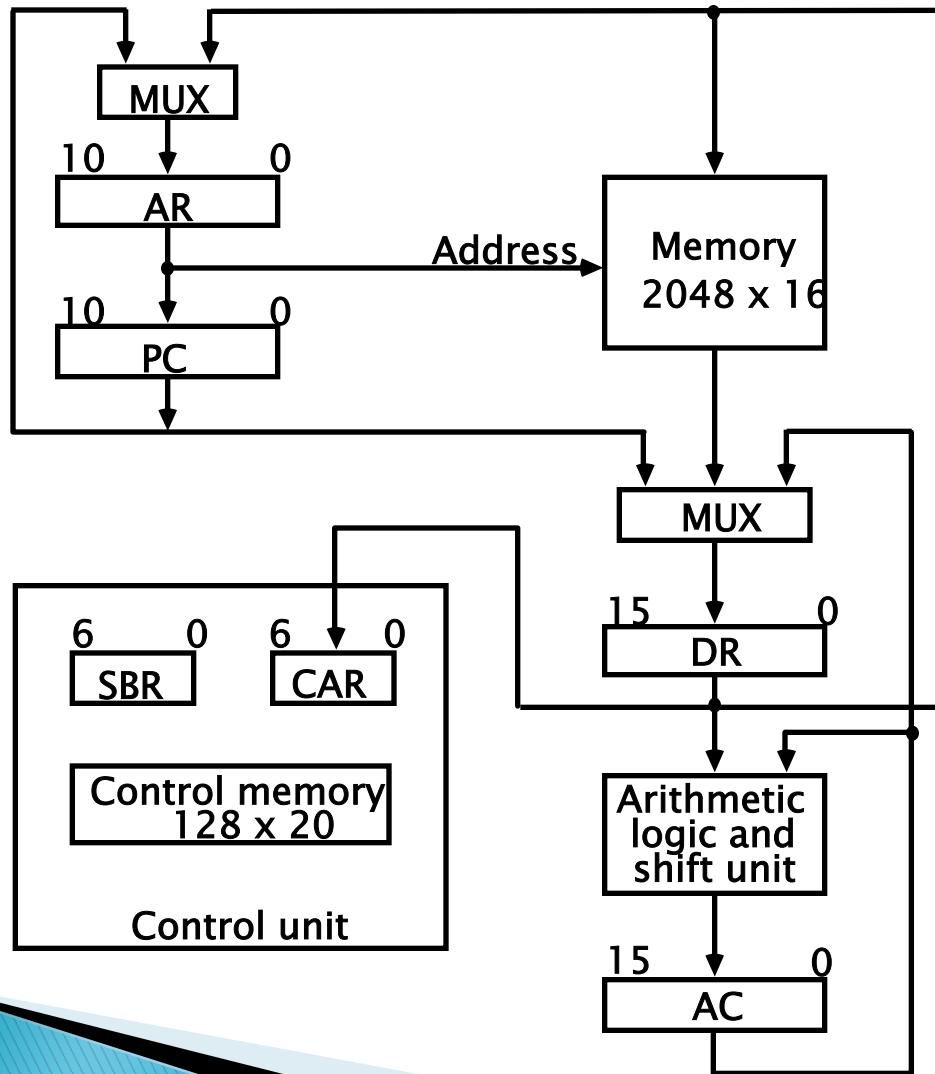


ROM veya PLA kullanarak MAPPING Lojiğinin gerçekleşmesi



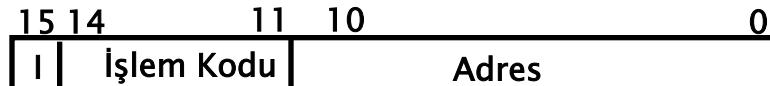
Mikroprogram Örneği

Bilgisayar Konfigürasyonu



Makine Komut Formatı

Makine komutu Formatı

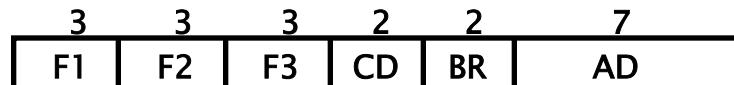


Örnek Makine Komutları

Sembol	İşlem Kodu	Tanımlama
ADD	0000	$AC \leftarrow AC + M[EA]$
BRANCH	0001	if ($AC < 0$) then ($PC \leftarrow EA$)
STORE	0010	$M[EA] \leftarrow AC$
EXCHANGE	0011	$AC \leftarrow M[EA], M[EA] \leftarrow AC$

EA efektif adres

Mikrokomut Formatı



F1, F2, F3: Mikroişlem alanları

CD: Dallanma için Şartlar

BR: Dallanma Alanı

AD: Adres Alanı

Mikrokomut Alan Tanımlamaları F1, F2, F3

F1	Mikroişlem	Sembol
000	None	NOP
001	$AC \leftarrow AC + DR$	ADD
010	$AC \leftarrow 0$	CLRAC
011	$AC \leftarrow AC + 1$	INCAC
100	$AC \leftarrow DR$	DRTAC
101	$AR \leftarrow DR(0-10)$	DRTAR
110	$AR \leftarrow PC$	PCTAR
111	$M[AR] \leftarrow DR$	WRITE

F2	Mikroişlem	Sembol
000	None	NOP
001	$AC \leftarrow AC - DR$	SUB
010	$AC \leftarrow AC \vee DR$	OR
011	$AC \leftarrow AC \wedge DR$	AND
100	$DR \leftarrow M[AR]$	READ
101	$DR \leftarrow AC$	ACTDR
110	$DR \leftarrow DR + 1$	INCDR
111	$DR(0-10) \leftarrow PC$	PCTDR

F3	Mikroişlem	Sembol
000	None	NOP
001	$AC \leftarrow AC \oplus DR$	XOR
010	$AC \leftarrow AC'$	COM
011	$AC \leftarrow shl AC$	SHL
100	$AC \leftarrow shr AC$	SHR
101	$PC \leftarrow PC + 1$	INCPC
110	$PC \leftarrow AR$	ARTPC
111	Reserve	

Mikrokomut Alan Tanımlamaları - CD, BR

CD	Şart	Sembol	Açıklama
00	Daima = 1	U	Şartsız dallanma
01	DR(15)	I	Dolaylı adres biti
10	AC(15)	S	AC nin işaretti
11	AC = 0	Z	AC içeriğinin sıfır kontrolü

BR	Sembol	Fonksiyon
00	JMP	CAR \leftarrow AD if şart = 1 CAR \leftarrow CAR + 1 if şart = 0
01	CALL	CAR \leftarrow AD, SBR \leftarrow CAR + 1 if şart= 1 CAR \leftarrow CAR + 1 if şart = 0
10	RET	CAR \leftarrow SBR (Altprogramdan dönüş)
11	MAP	CAR(2-5) \leftarrow DR(11-14), CAR(0,1,6) \leftarrow 0

Sembolik Mikrokomutlar

- Semboller asembler dilindeki gibi mikrokomutlarda kullanılırlar.
- Bir sembolik mikroprogram, bir mikroprogram asembler yardımcıyla ikili eşdeğerine dönüştürülür.

Örnek Format

5 Alan: Etiket; mikroişlemler; CD; BR; AD

Etiket: Boş olabilir veya bir sembolik adres olabilir (: ile sonlanır)

Mikroişlemler: Birbirinden , ile ayrılır. 1, 2 veya 3 sembol içerir.

CD: {U, I, S, Z} sembollerinden biri ,

U: Şartsız Dallanma

I: Dolaylı adres biti

S: AC nin işaretçi

Z: AC nin sıfır içerik kontrolü

BR: {JMP, CALL, RET, MAP} sembollerinden biri

AD: {Sembolik adres, NEXT, boş}

Sembolik Mikroprogram

Fetch Rutini

FETCH işlemi esnasında, bellekten bir komut okunur,
Komut kodu çözülür ve PC ye bir sonraki adres yüklenir

Fetch çevriminde icra edilen mikroişlem dizinleri

```
AR ← PC
DR ← M[AR], PC ← PC + 1
AR ← DR(0-10), CAR(2-5) ← DR(11-14), CAR(0,1,6) ← 0
```

Sembolik mikroprogramı :

	ORG 64
FETCH:	PCTAR U JMP NEXT
	READ, INCPC U JMP NEXT
	DRTAR U MAP

Assembler tarafından dönüştürülen ikili kod eşdeğerleri

Binary address	F1	F2	F3	CD	BR	AD
1000000	110	000	000	00	00	1000001
1000001	000	100	101	00	00	1000010
1000010	101	000	000	00	11	0000000

Sembolik Mikroprogram

- Kontrol Bellek Kapasitesi : 128 adet 20-bit kelime 128x20
- İlk 64 kelime : 16 makine komutu için rutin programları
- Son 64 kelime: Çeşitli amaçlar için kullanılan rutinler (örneğin , fetch rutini, dolaylı adres hesabı rutini,)
- Mapping: OP-kod XXXX kullanılarak 0XXXX00 şeklinde bir adrese dönüşüm,
- 16 rutin için başlangıç adresler : 0(0 0000 00), 4(0 0001 00), 8, 12, 16, 20, ..., 60

Örnek komutlar için Sembolik Mikroprogramlar

Etiket	Mikroişlemler	CD	BR	AD
ADD:	ORG 0 NOP READ ADD	I U U	CALL JMP JMP	INDRCT NEXT FETCH
BRANCH:	ORG 4 NOP NOP	S U	JMP JMP	OVER
OVER:	NOP ARTPC	I	CALL JMP	FETCH INDRCT FETCH
STORE:	ORG 8 NOP ACTDR WRITE	I U U	CALL JMP JMP	INDRCT NEXT FETCH
EXCHANGE:	ORG 12 NOP READ ACTDR, DRTAC WRITE	I U U U	CALL JMP JMP JMP	INDRCT NEXT NEXT FETCH
FETCH:	ORG 64 PCTAR READ, INCPC	U U	JMP JMP	NEXT NEXT
INDRCT:	DRTAR READ DRTAR	U U U	MAP JMP RET	NEXT

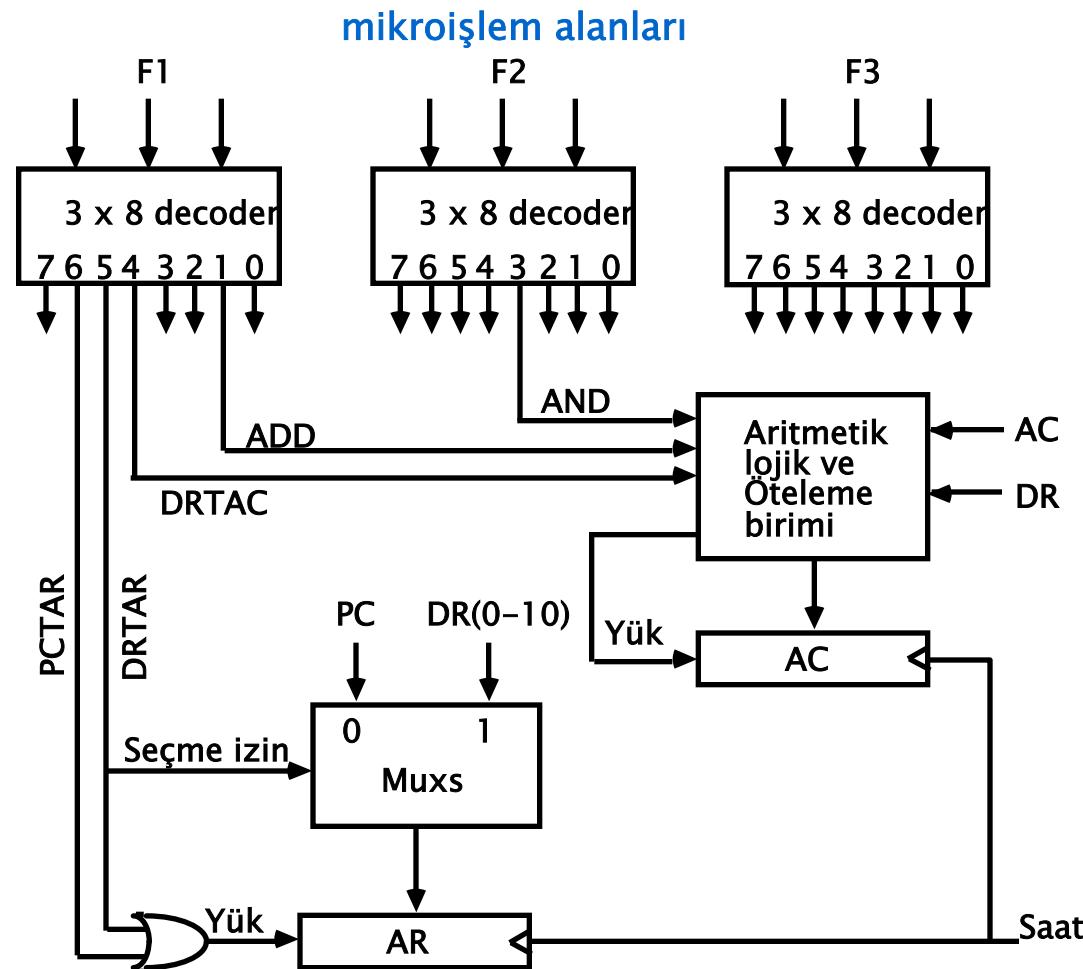
İkili Mikroprogram

Mikro Rutin	Adres		İkili Mikrokomut					
	Onlu	İkili	F1	F2	F3	CD	BR	AD
ADD	0	0000000	000	000	000	01	01	1000011
	1	0000001	000	100	000	00	00	0000010
	2	0000010	001	000	000	00	00	1000000
	3	0000011	000	000	000	00	00	1000000
BRANCH	4	0000100	000	000	000	10	00	0000110
	5	0000101	000	000	000	00	00	1000000
	6	0000110	000	000	000	01	01	1000011
STORE	7	0000111	000	000	110	00	00	1000000
	8	0001000	000	000	000	01	01	1000011
	9	0001001	000	101	000	00	00	0001010
	10	0001010	111	000	000	00	00	1000000
EXCHANGE	11	0001011	000	000	000	00	00	1000000
	12	0001100	000	000	000	01	01	1000011
	13	0001101	001	000	000	00	00	0001110
	14	0001110	100	101	000	00	00	0001111
	15	0001111	111	000	000	00	00	1000000
FETCH	64	1000000	110	000	000	00	00	1000001
	65	1000001	000	100	101	00	00	1000010
INDRCT	66	1000010	101	000	000	00	11	0000000
	67	1000011	000	100	000	00	00	1000100
	68	1000100	101	000	000	00	10	0000000

Bu mikroprogramlar bir ROM kullanılarak gerçekleştirilebilir.

Kontrol Ünitesi Tasarımı

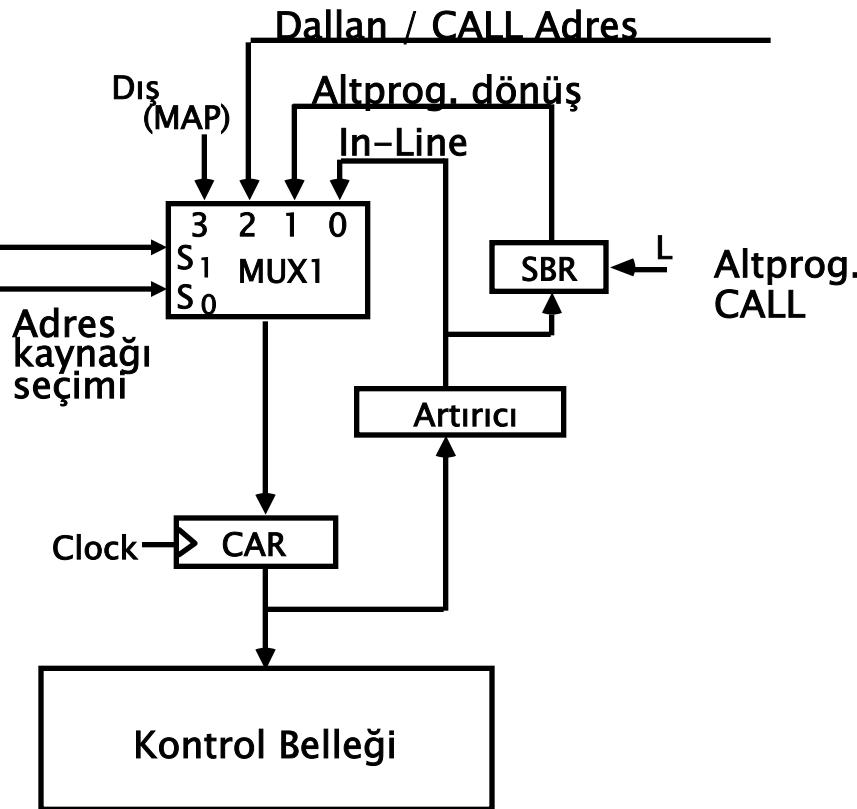
Kodçözümü ve ALU birimi



Mikroprogram Düzenleyici

Bir Sonraki Mikrokomut Adres Seçimi

$S_1 S_0$	Adres Kaynakları
00	CAR + 1, Bir sonraki adr
01	SBR RETURN
10	CW(AD), Dallan veya CALL
11	MAP

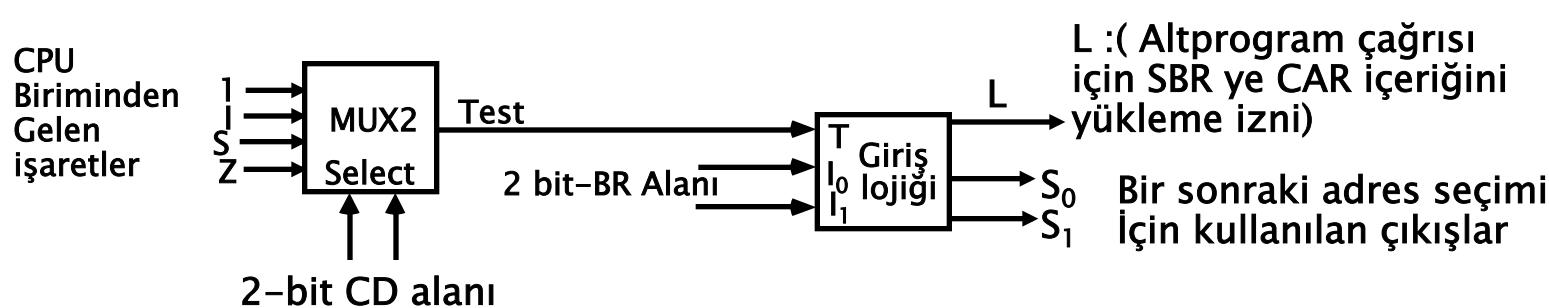


MUX-1 4 kaynak adresinden birini seçer ve CAR ye yönlendirir:

- Bir sonraki adres \rightarrow CAR + 1
- Dallan/Branch), Altprog.Çağrı(Subroutine Call) \rightarrow Kontrol kelimesinin AD alanı
- Altprog. dönüşü \rightarrow SBR dan alınan adres
- Yeni Makine Komutu \rightarrow MAP

Mikroprogram Düzenleyici

ŞART ve DALLANMA KONTROLÜ



Giriş Lojiği

$I_0 I_1 T$	Anlamı	Adres Kaynakları	$S_1 S_0$	L
000	Bir sonraki	CAR+1	00	0
001	JMP	CW(AD)	10	0
010	Bir sonraki	CAR+1	00	0
011	CALL	CW(AD) and SBR <- CAR+1	10	1
10x	RET	SBR	01	0
11x	MAP	DR(11-14)	11	0

$$\begin{aligned}S_0 &= I_0 \\S_1 &= I_0 I_1 + I_0 ' T \\L &= I_0 ' I_1 T\end{aligned}$$

Mikroprogram Düzenleyici

