

# Bölüm 7

## Mikro programlanmış Denetim

# 7.1 Denetim Belleği

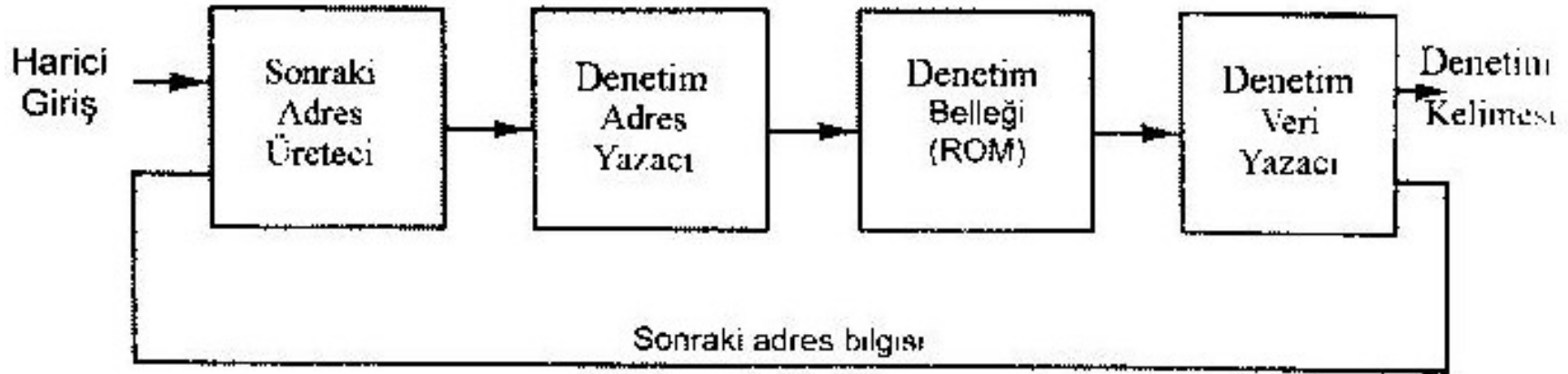
- Bir bilgisayardaki denetim biriminin görevi mikroişlemlerin sırasını başlatmaktır. Sistemdeki mikro işlemlerin sayısı sınırlıdır.
- Donanım karmaşıklığı, çalıştırılacak sıralı mikroişlemlerin sayısı ile doğru orantılıdır.
- Denetim biriminin donanımsal olarak yaptığı mikroişlemlerin işleyiş sıralaması için 2. bir yol mikroprogramlamadır.
- Denetim fonksiyonu, binary bir değişkendir ve mikroişlemleri belirler.
- Herhangi bir zamanda verilen denetim değişkeni 1 ve 0'lerden oluşan bir dizidir. Denetim kelimesi olarak adlandırılır.
- Denetim kelimesi, sistemin parçaları üzerinde çeşitli işlemleri yerine getirebilecek şekilde programlanabilir.

## 7.1 Denetim Belleği-2

- Bir denetim biriminin binary değişkenleri bellekte saklanır. Mikro programlanmış denetim olarak adlandırılır.
- Denetim belleğindeki her kelime bir mikro buyruk içerir.
- Mikro buyruk bir veya daha fazla mikroişlem içerir.
- Mikro buyrukların sırası ise bir mikro programı oluşturur.
- Denetim birim aktifken, mikro buyrukların içeriği değişmez.
- Denetim belleği ROM olabilir.
- Daha ileri bir uygulamada ise denetim mikroprogramı bir HD .çerisinde barındırılır. Buna Dinamik mikro program denir.

## 7.1 Denetim Belleği-3

- Mikro programlanmış denetim birimi içeren bilgisayarlarda iki ayrı bellek vardır.
- Ana bellek: **kullanıcı programlarının ve verilerinin bulunduğu bellektir.**
- Denetim belleği: **Değiştirilemez, sabit bir mikroprogramın tutulduğu bellektir. Her bir makine buyruğu, denetim belleğindeki bir dizi mikro buyrukları başlatır.**
- **Mikroprogramlanmış denetim biriminin genel görünümü Ş.7.1dedir.**



**Şekil 7.1** Mikro programlanmış denetim organizasyonu

# Adres Sıralama

- Mikrobuyruklar, her gurubun bir yordam (routine) belirttiği guruplardaki denetim belleklerinde saklanır.
- Her bilgisayarın buyruğu kendi denetim belleğinde kendi mikro programına sahiptir.
- Denetim hafızasının adres sıralamasını kontrol eden donanım, bir yordam ile mikro buyrukların sıralamasını ve bir yordamdan diğer yordama dallanabilmesini sağlamak zorundadır.
- Bir bilgisayar açıldığında ilk adres denetim yazacına yüklenir. Bu adres genelde ilk mikrobuyruğun adresidir ve buyru al-getir (fetch) yordamını içerir.
- Al-getir yordamının sonundaki buyruk, bilgisayarın buyruk yazacının içindekidir.
- Daha sonra denetim belleği yordamı inceler ve işlenenin etkin adresi elde edilir.

# Haritalama

- Efektif adres hesaplama yordamı bittiğinde işlenenin adresi bellek adres yazacında bulunmaktadır.
- Bir sonraki adım bellekten getirilen buyruğu, işlem kodu kısmına bakılarak işlemci yazacında oluşturmaktır.
- Herbir buyruğun kendi mikroprogram yordamı vardır.ve bu program denetim belleğinde verilen bir adrese yerleştirilmiştir.
- Denetim belleğine yerleştirilmiş olan rutin(yordam) buyruk kod bitlerinin bir adres dönüşümü ile bulunması **haritalama** olarak isimlendirilir.
- Haritalama işlemi, bir buyruk kodunu denetim bellek adresine dönüştüren bir kuraldır.
- Buyrukları icra eden mikrobuyruklar denetim adres yazacının arttırılması ile sıralanır.
- Buyruğun icrası tamamlandığında,denetim al-getir yordamına dönmelidir. Buda şartsız dallanma ile olmalıdır.

# Haritalama -1

- Özetle, denetim belleğinde adres sıralama işlemleri;

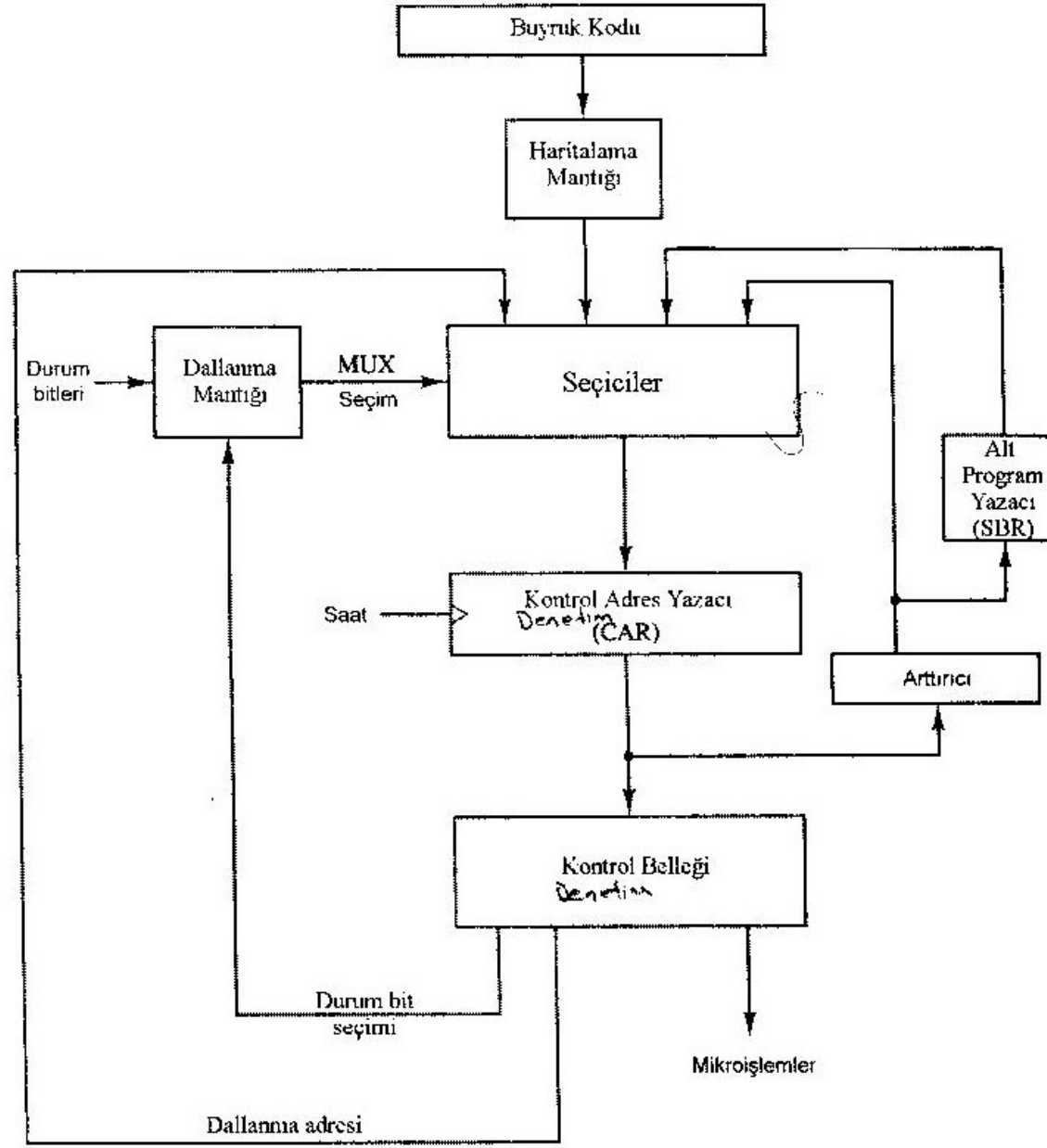
1- Denetim adres yazacında artma

2-Durum bitleinin kuralına bağlı olarak şartlı veya şartsız dallanma

3-Buyruk bitlerinin bir adrese denetim belleği için haritalama işlemi

4-Alt programa giriş ve çıkış için kolaylıklar.

Şekil 7.2’de mikrobuyuğun adres seçimi için ihtiyaç duyulan denetim belleği ve ilişkili donanım şeması verilmektedir.



Şekil 7.2 Denetim belleği için adresin seçimi

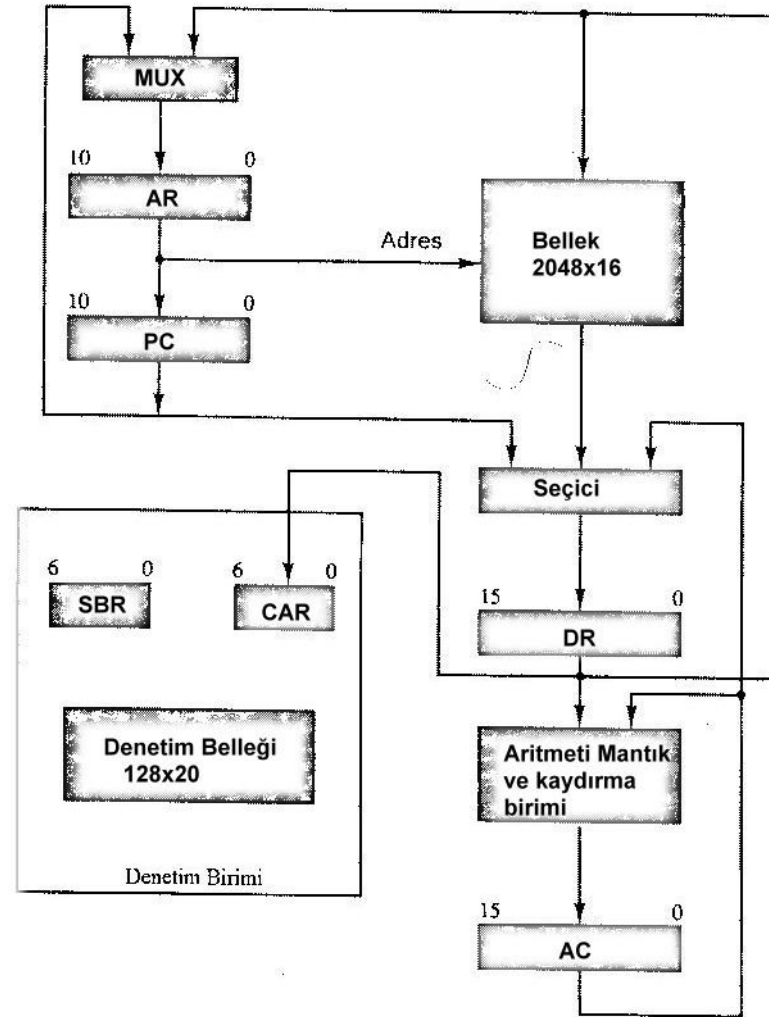


# Mikroprogram Örneği

- Bir bilgisayarın konfigirasyonu ve mikroprogramlanmış denetim birimi kurulduktan sonra,tasarımcının görevi denetim belleği için mikrokodlar yaratmaktır.
- Bu kod yaratmaya mikro programlama denir. Bu işlem çevirici programlamanın benzeridir.
- Bunu bir örnekle inceleyelim.

# Bilgisayar Konfigirasyonu

- Bilgisayarın blok şeması 7.4'de verilmiştir. Bunun 2 bellek birimi vardır.
- Denetim belleği mikro buyrukları saklamak içindir.
- Denetim birimine bağlı iki yazaç vardır( CAR(Denetim adresi yazacı) ve SBR(Alt program yazacı)). İşelmcı yazaçları PC,AR,DR,AC yazaçları.
- Denetim belleği ve bunun yazaçları, miro programlanmış denetim birimi olarak organize edilmişlerdir (Ş.7.2'ye bakın)
- Yazaçlar arası bilgi aktarımı, ortak veriyolu ile değil MUX'larla yapılır.
- DR,AC den, PC'den veya bellekten bilgi alabilir.



Şekil 7.4 Bilgisayar donanım düzeni

# Buyruk biçimi -2

- Mikroişlemlerin herbiri (F1,F2,F3) 3 bitlik alanlara yazılır. Çizelge 7.1 toplam 21 mikroişlem eder. CD alanı durum şarlarını gösteren 2 bitlik alandır. BR alanı kullanılacak dallanmanın tipini belirtir. Adres alanı 7 bittir. (Denetim belleği 128 kelimedir).
- Bir mikrobuyruk 3'ten fazla mikroişlem içermez.
- İki eşzamanlı mikroişlem F2 ve F3 ve boş alan F1 olsun.

$DR \leftarrow M[AR]$       F2 = 100

$PC \leftarrow PC+1$       F3 = 101

Dolayısıyla mikroişlem alanları,  
000 100 101 olur.

Aynı anda birbirine çakışan mikroişlem gurubu olamaz.

Ç.7.1de mikroişlemlerin herbiri yazaç aktarım diliyle yazılmış ve sembollerle gösterilmiş şekli vardır.

# Buyruk Biçimi



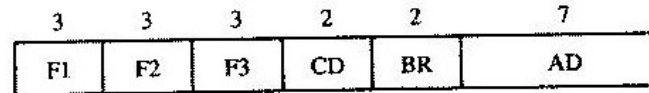
(a) Buyruk biçimi

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

EA Efektif Adrestir

(b) Dört bilgisayar buyruğu

**Şekil 7.5** Bilgisayar buyruk biçimi



F1,F2,F3 : Mikroişlem alanları  
 CD : Dallanma için şart  
 BR : Dallanma alanı  
 AD : Adres Alanı

**Şekil 7.6** Mikro buyruk kod biçimi (20 bit).

**Çizelge 7.1** Mikro buyruk olanları için sembol ve ikili kodları

F1	Mikro işlem	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	Mikro işlem	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	Mikro işlem	Sembol
000	None	NOP
001	$AC \leftarrow AC \oplus DR$	XOR
010	$AC \leftarrow \overline{AC}$	COM
011	$AC \leftarrow \text{Shl } AC$	SHL
100	$AC \leftarrow \text{Shr } AC$	SHR
101	$PC \leftarrow PC + 1$	INCP
110	$PC \leftarrow AR$	ARTPC
111	Ayrılmış	

CD	Şart	Sembol	Yorumu
00	her zaman = 1	U	şartsız dallanma
01	$DR(15)$	I	dolaylı adres bit
10	$AC(15)$	S	AC nin işaret biti
11	$AC = 0$	Z	AC nin değeri sıfır

BR	Sembol	Fonksiyon
00	JMP	$CAR \leftarrow AD$ eğer şart = 1 ise $CAR \leftarrow CAR + 1$ eğer şart = 0 ise
01	CALL	$CAR \leftarrow AD, SBR \leftarrow CAR + 1$ eğer şart = 1 ise $CAR \leftarrow CAR + 1$ eğer şart = 0 ise
10	RET	$CAR \leftarrow SBR$ (All push and on push down)
11	MAP	$CAR(15) \leftarrow DR(15), I(15) \leftarrow AR(0-10), Z(15) \leftarrow 0$

# Sembolik Mikro buyruk

Bir sembolik mikroprogram birleştirici aracılığı ile binary karşılığınadönüştürülür.

Bu birleştirici dil, mikro buyruğun her alanı için semboller tanımlamalı ve kullanıcıların kendi adreslerini sembollerle belirleme imkanı vermelidir.

- Birleştirici dilindeki bir mikro programın her satırı sembolik bir mikro buyruktur. Bu mikrobuyrukların herbiri 5 alana ayrılır. Başlık, mikro işlemler, CD, BR, ve AD.

1-Başlık alanı boş olabilir veya sembolik adres verilebilir. Başlık alanı : ile biter.

2-Mikroişlem alanında , ile ayrılmış 1,2 veya 3 sembol bulunabilir. Semboller çiz.7.1'de verilmiştir. NOP, mikroişlem olmadığında kullanılır. Çevirici tarafından 9 tane 0 olarak yorumlanır

3- CD alanında U,I,S veya Z 'den biri bulunur.4

4- BR alanında Çiz.7.1'deki 4 sembolden biri bulunur.

5-Ad alanı adres için ayrılmış olup, 3 farklı biçim olabilir.

a)Sembolik adres: Aynı zamanda başlıkta da yer almalıdır.

b)NEXT sembolu sıradaki bir sonraki adresi gösterir.

c) BR alanında RET veya MAP sembolü varsa, AD alanı boştur. Çevirici tarafından 7 tane 0 olarak aktarılır.

# AL GETİR YORDAMI

- Denetim belleği 128 kelime ve her kelime 20 bittir. Denetim belleğini mikroprogramlamak için 128 kelimenin bitlerinin tek tek belirlenmesi gerekir. 0-63 adresleri arasındaki 64 kelime 16 buyruğa ayrılmıştır. Al-getir yordamı için başlangıç adresi 64 olabilir.

- Al-getir (FETCH) yordamı için gerekli mikrobuyruklar;

$AR \leftarrow PC$

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

$AR \leftarrow DR(0-10), CAR(2-5) \leftarrow DR(11-14), CAR(0,1,6) \leftarrow 0$

Buyruğun adresi PC'den AR'ye aktarılır. Ve buyruk belekten DR'ye okunur. Buyruk yazacı olmadığından buyruk kodu DR'de kalır. Adres parçası AR'ye aktarılır. Ve denetim 16 yordamdan birine geçer. Bunun için işlem kodu DR'den alınır ve haritalanarak CAR'a gönderilir.



# AL Getir, kodunu ÇÖZ

- AL GETİR yordamında 3 mikro buyruk olacaktır. Bunlar 64,65,66 adreslerinde bulunacaktır. tanımlanan çevirici dile göre, al-getir evresi için mikro program;

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

*Burada MAP mikrobuyruğu bir dallanma adresi tanımlar. 0xxxx00. Buradaki xxxx işlem kısmının 4 bitidir. Örneğin bu bir toplama buyruğu ise bunun işlem kodu 0000 Dur. Bu durumda haritalama mikrobuyruğu (MAP), CAR'a 0000000 adresini yazar. Bu da ADD rutininin denetim belleğindeki başlangıç adresidir.*

- Bu sembolik programın binary karşılığı;

Binary adres	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

Bit değerleri ç.7.1'den alınmıştır. Binary şekil, denetim belleğine konan gerçek biçimdir.

# Sembolik Mikroprogram

Çizelge 7.2 Sembolik mikro program

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

# Sembolik mikro programlama-2

- Çizelge 7.2'de değişik rutinler için sembolik mikroprogramlar bulunur. Burada 4 yordam vardır.
- Her rutinde, etkin adresi hesaplayacak mikro buyruk bulunmalıdır. Dolaylı adres kipi bütün bellek adreslemeli buyruklarda vardır. Dolaylı adreslerle ilgili ilgili buyruklar bir alt programda saklanır. Bu alt program, INDRCT dir. FECT rutininin ardından gelir.

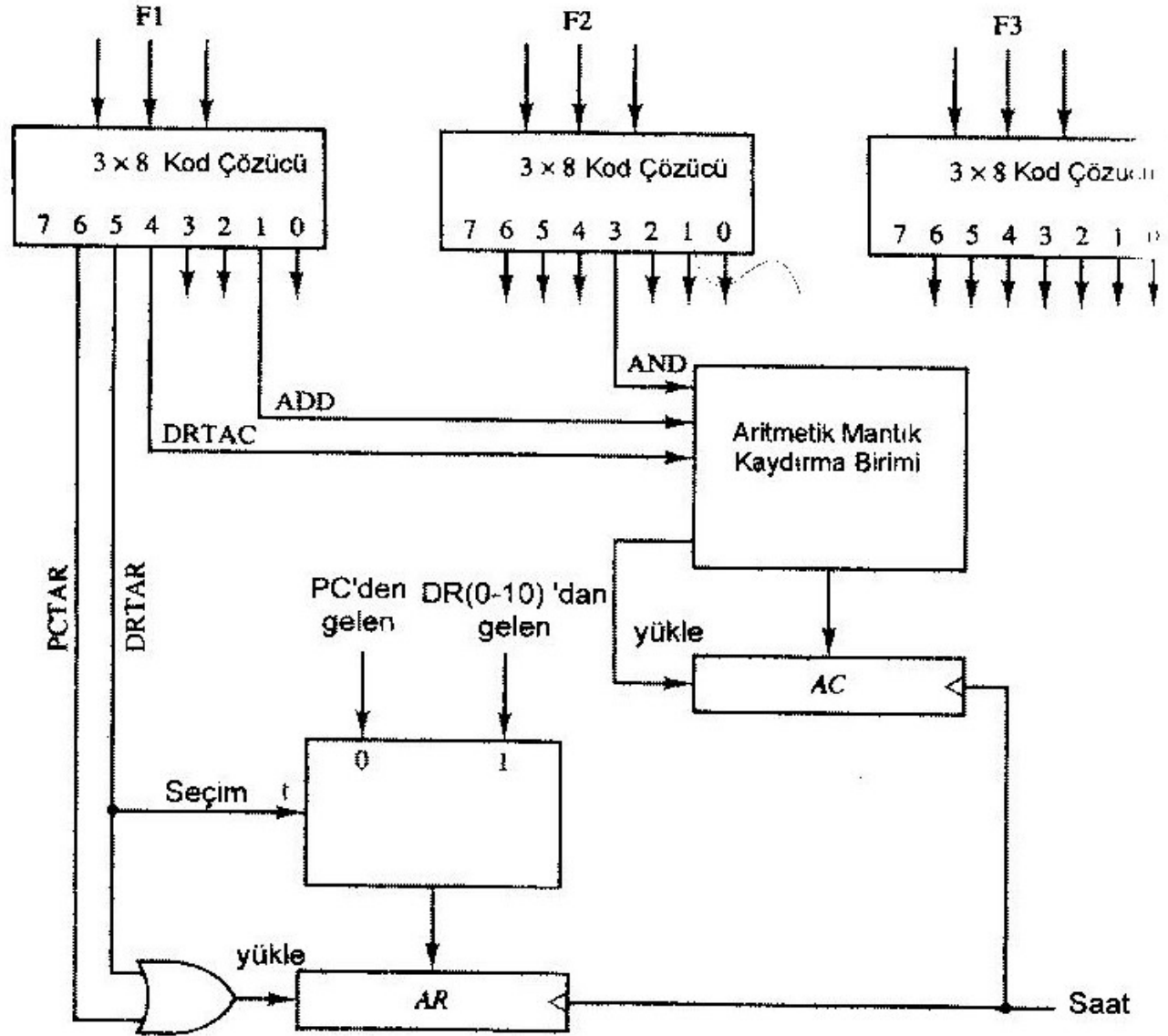
# Binary Mikro program

**Çizelge 7.3** Denetim belleği için ikili mikro program parçası

Mikro Yordam	Adres		İkili mikro buyruk					
	Ondalık	İkilik	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
	7	0000111	000	000	110	00	00	1000000
STORE	8	0001000	000	000	000	01	01	1000011
	9	0001001	000	101	000	00	00	0001010
	10	0001010	111	000	000	00	00	1000000
	11	0001011	000	000	000	00	00	1000000
EXCHANGE	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
	66	1000010	101	000	000	00	11	0000000
INDIRECT	67	1000011	000	100	000	00	00	1000100
	68	1000100	101	000	000	00	10	0000000

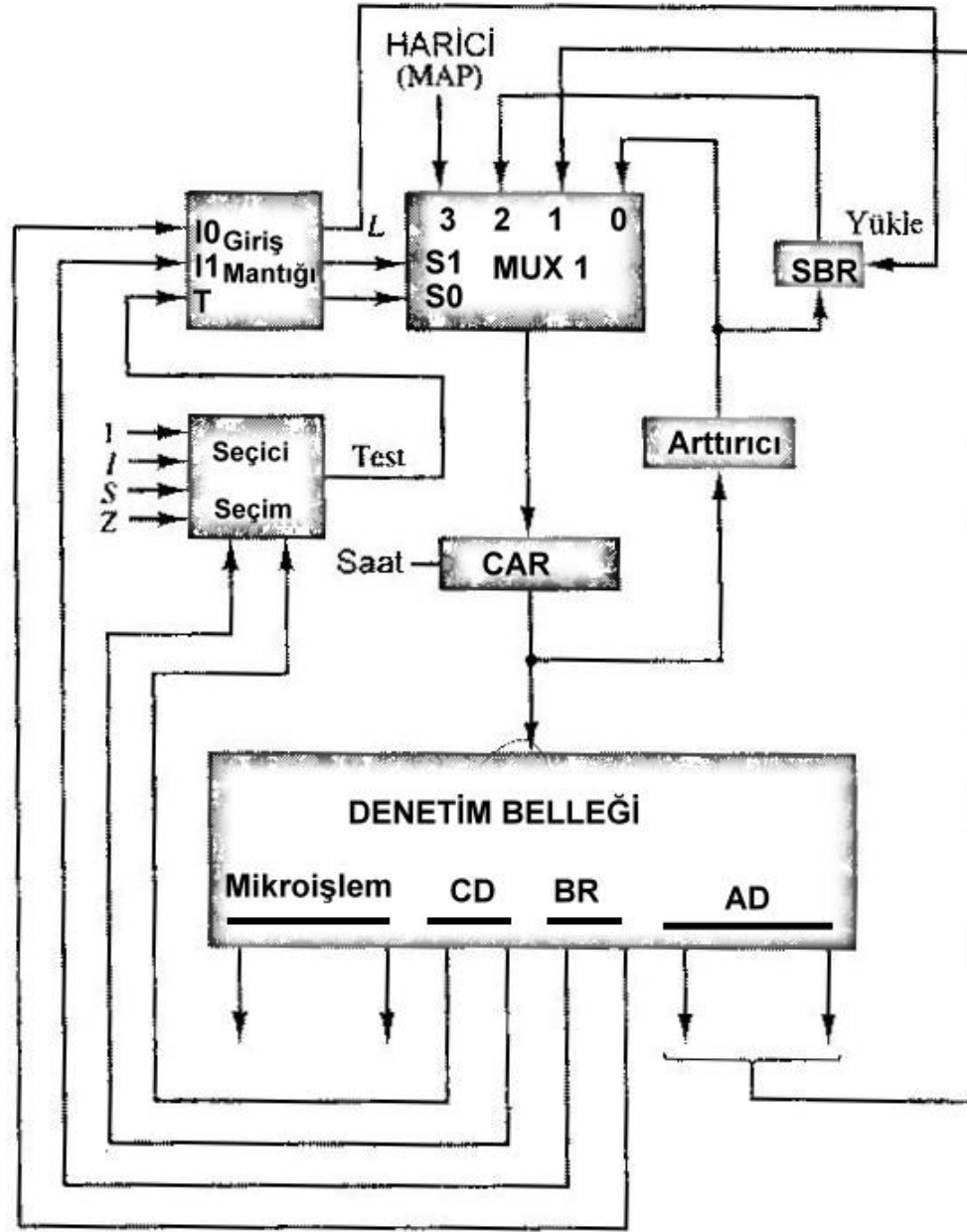
# Denetim Birimi

- Mikrobuyruğun bitleri, çoğunlukla bitakım alanlar şeklinde guruplanır. Her bir alan farklı ve belirgin bir fonksiyonu tanımlar. Bu fonksiyon ve alanlar sistemdeki mikro işlemleri başlatır.
- Özel bitlerle, adresin nasıl elde edileceği belli olur. Dallnma için adres verilir.
- Mikroişlemlerin sayısı guruplama ile azaltılır. Bu guruplama alanları, gereken denetim sinyallerini oluşturmak için bir kod çözücüyeye gerek duyar.
- Denetim buyruğundaki 9 bitlik mikro işlem alanı 3 alana ayrılmıştı.Dolayısıyla, denetim belleği çıkışı ayrı mikroişlemler tanımlamalıdır.Kod çözücülerin çıkışları işlemcinin uygun girişlerine bağlanmalıdır.



Şekil 7.7 Mikro işlem alanlarının kodunun çözülmesi

- Ş.7.7 de 3 kod çözücü ve bunların bazı bağlantıları görülmektedir. Kod özüklerin çıkışları, ilgili mikro işlemleri başlatacak yerlere bağlanmıştır.
- Örneğin  $F1=5$  (101) olduğunda, DR(0-10) içeriği AR'ye clock vuruşu ile aktarılacaktır. (DRTAR)
- Şekilden görüldüğü gibi, F1 kod çözücüsünün 5 ve 6 çıkışları AR'nin yükle girişine bağlıdır. Dolayısıyla bu iki çıkıştan biri aktif olursa **MUX'lardaki** bilgi AR'ye aktarılır.



**Şekil 7.8** Bir denetim belleği için mikro program sıralayıcı