

# Bölüm 7

## Mikro Programlanmış Denetim

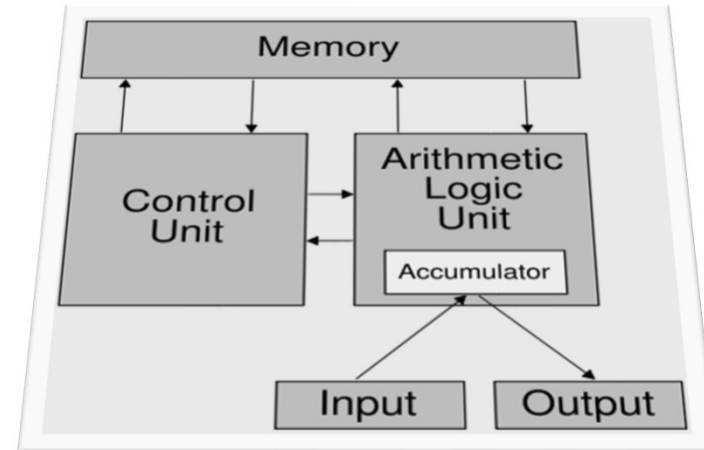
170260101 – Mert İNCİDELEN

16260406 – Fahrettin VARLIK

16260061 – Kazım BAŞLAK

- Bir mimaride tanımlı bilgisayar buyruklarının hangi işi gerçekleştireceği denetim birimi yardımı ile belirlenir.
- Denetim biriminde bilgisayar buyruklarını gerçekleştiren mikro buyruklar tanımlıdır.
- Mikro buyruklar, mikro programda bulunurlar ve mikro işlemler barındırırlar.
- Mikro program, denetim birimindeki denetim belleğinde bulunur.
- Bellekte bulunan mikro buyruklar mikro işlemleri icra ederler.

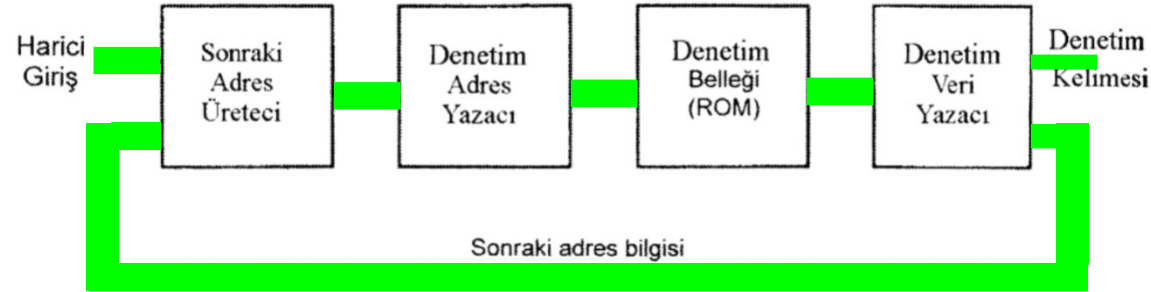
**Denetim birimi organizasyonunu anlamaya çalışalım.**



## 7.1. Denetim Belleği

---

- Mikro program, denetim birimindeki denetim belleğinde bulunur.
- Her denetim kelimesi bir mikro buyruktur.
- Mikro buyruklar bir veya daha fazla mikro işlem barındırırlar.
- Bu mimarimizde denetim belleğindeki **denetim kelimeleri 20 bit uzunluğundadır.**
- Bellek adresini göstermek için denetim birimimizde **7 bit uzunluğunda bir adres yazacı** bulunur.
- Dolayısıyla belleğimiz **128 adet 20 bit uzunluğunda adrese** sahiptir.



**Yukarıda, kabaca bir mikro program denetim organizasyonu gösterilmiştir.**

- İlk olarak denetim biriminin dışından haritalama işlemi ile oluşturulmuş harici bir adres geldiğini düşünelim. (Adres, seçici yardımıyla 4 farklı yol ile belirlenebilir)
- Bu adres denetim adres yazacına (CAR) yüklenir.
- Adres yazacı denetim belleğindeki mikro buyruğun adresini gösterir.
- Bellekten okunan mikro buyruk denetim veri yazacında tutulur.
- Denetim kelimesi bir ya da daha fazla mikro işlemi tarifler.
- Sonraki adres bilgisi; şartlı ve şartsız dallanmaya göre ya da adresi bir arttırılarak seçici yardımıyla 4 durumdan biri seçilip belirlenir.

## 7.2. Adres Sıralama

---

**Denetim biriminde adres seçimi 4 farklı şekilde gerçekleştirilebilir:**

1. Bilgisayar buyruğundan denetim adresi çıkarımı (Haritalama)
2. Denetim belleğindeki adres yazacını bir arttırma.
3. Alt programdan dönüşte alt program yazacındaki adrese dönme.
4. Mikro programdaki durum bitlerine bağlı olarak şartlı veya şartsız dallanma.

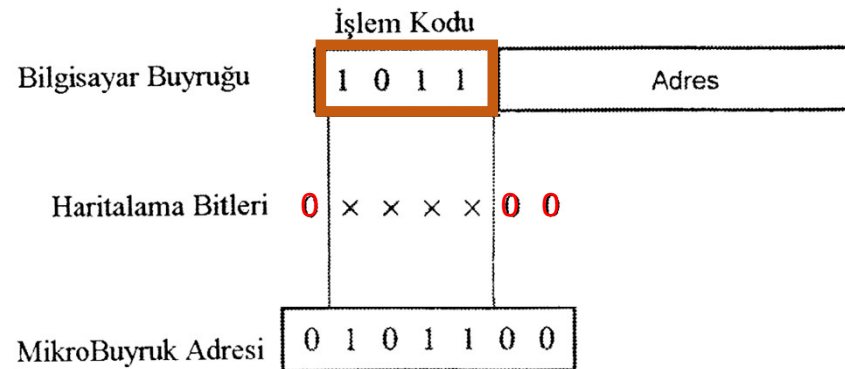
## 7.2.1. Haritalama İşlemi

(Bilgisayar Buyruğundan Denetim Adresinin Çıkarımı)

---

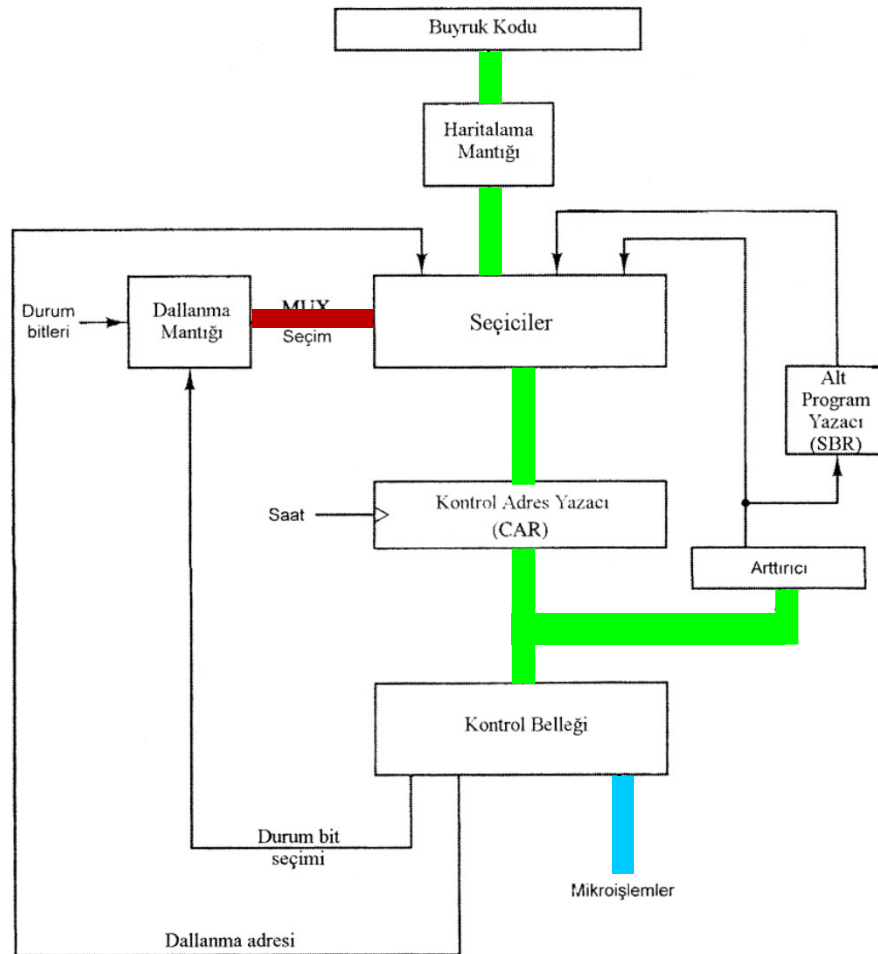
- Her bir bilgisayar buyruğu bir dizi mikro buyruğa sahiptir.
- Mikro buyruklar, mikro programın adreslerinde yer alırlar ve bir dizi mikro işlemi gerçekleştirirler.
- Bilgisayar buyruğundan denetim belleğindeki mikro buyruklar için adres saptaması yapabiliriz.
- Bu saptama işlemi haritalama ile gerçekleştirilir.

## Bir bilgisayar buyruğunu ele alalım:



- Buyruğun 4 bitlik işlem kodu alınır.
- Denetim belleğimizin adreslemesi 7 bit olduğundan haritalama yöntemiyle işlem kodunun en ağırlıklı bitinin önüne bir adet 0 ve en ağırlıksız bitinin ardına iki adet 0 biti eklenerek haritalama işlemi gerçekleştirilir.
- Oluşan 7 bitlik sözcük mikro buyruğu gösteren bir adres olur.

# Harici bir adres belirleyerek mikro işlemler üretelim:



- Buyruk kodumuz haritalama işlemi ile 7 bitlik adres üretir.
- Seçici yardımıyla bu adres seçilerek adres yazacına aktarılır.
- Adres yazacı bellekte ilgili mikro buyruğun adresini tarifler. Aynı zamanda bu yazaçtaki adres arttırıcıya gönderilir.
- Adresin gösterdiği yerde bulunan mikro buyruk, mikro işlemler üretir.

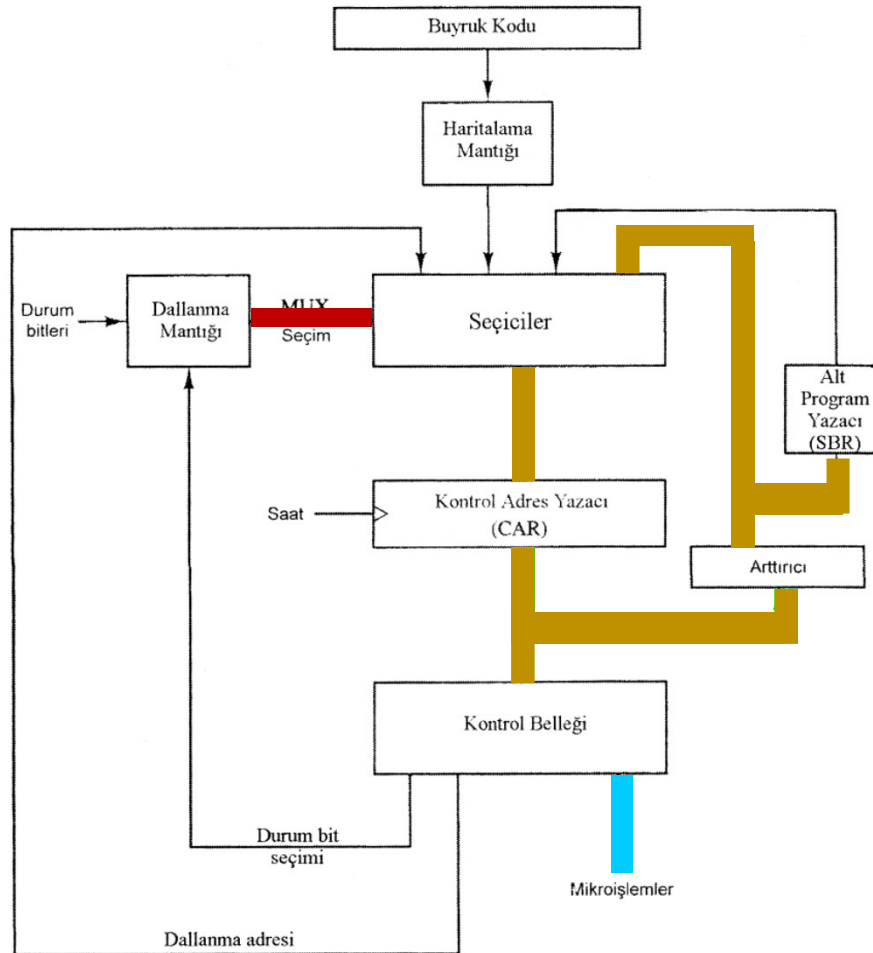


## 7.2.2. Arttırıcı İle Bir Sonraki Adresin Belirlenmesi

---

- Adres yazacındaki adres denetim belleğindeki buyruğun adresini tariflerken, bir sonraki bellek adresinin tespiti için arttırıcı birimine gönderilir.
- Burada arttırılan adres bir sonraki mikro buyruğun adresini tarifler.
- Seçici yardımıyla arttırıcıdaki adres seçilerek program sonraki adresten icra edilmiş olur.

# Arttırıcı ile bir sonraki adrese gitme:



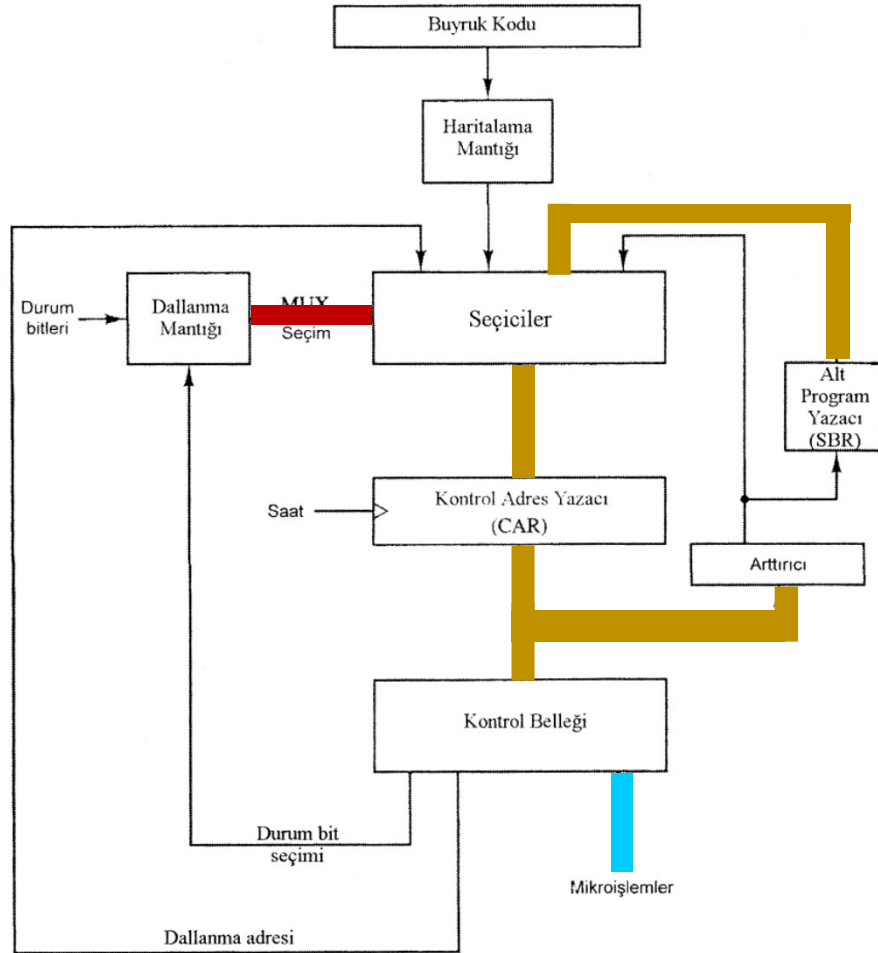
- Önceki adres arttırıcıya gelir.
- Arttırıcı, adresi 1 arttırır ve sonraki adres belirlenir.
- (Ayrıca alt programa gidildiğinde dönülecek adresin bir sonraki adres olması nedeniyle alt program yazacına da yazılabilir.)
- Seçici yardımı ile arttırıcı seçilerek bir sonraki adres kontrol adres yazacına yazılmış olur.
- Yeni adresin gösterdiği yerde bulunan mikro buyruk, mikro işlemler üretir. Ve bu adres de arttırıcıya gönderilmiş olur

## 7.2.3. Alt Programdan Kalınan Adrese Dönme

---

- Alt programa gidileceği zaman, adres yazacındaki adres bir arttırılarak alt program yazacına yazılır.
- Böylece programdan dönülürken, gidilen adresin bir sonraki adresine dönülme imkanı sağlanır.
- Alt programdan dönerken seçici tarafından alt program yazacı seçilerek dönülecek adres, adres yazacına yüklenir.

# Alt programdan kalınan adrese dönme:



- Alt programa gidilirken kalınan adres bir arttırılarak alt program yazacına yazılmıştı.
- Bu adrese dönmek için seçici yardımıyla alt program adresi seçilerek kontrol adres yazacına aktarılır.
- Kalınan adresin gösterdiği yerde bulunan mikro buyruk, mikro işlemler üretir. Ve bu adres de arttırıcıya gönderilmiş olur.

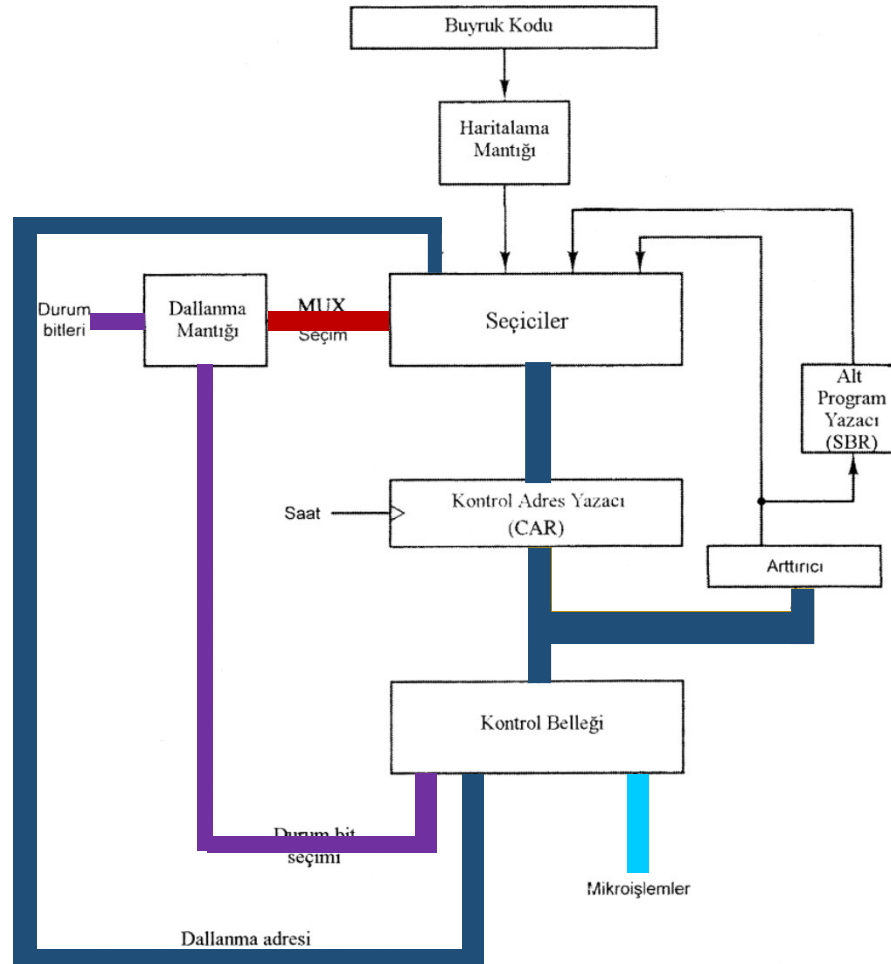
## 7.2.4. Şartlı ve Şartsız Dallanma

---

- Mikro programda icra edilen mikro buyruk, bir şarta bağlı olarak ya da şartsız başka bir adres üretebilir.
- Program bu adrese dallanarak devam eder.
- Şartsız dallanmada üretilen adres şart aranmaksızın seçici yardımıyla adres yazacına yüklenerek dallanma işlemi gerçekleştirilmiş olur.
- Şartlı dallanma ise durum bitlerine bağlı olarak gerçekleştirilir, üretilen adres seçici ile seçilerek adres yazacına yüklenir.

**Not:** Eğer bir alt programa gidiliyor ise, kalınan adres alt program yazacına yazılarak dallanma işlemi gerçekleştirilir.

## Şartlı ve şartsız dallanma ile:



- İcra edilen mikro buyruk bir şartsız dallanma adresi üretebilir.
- (Durum bitlerinin kuralına bağlı olarak şartlı ya da şartsız dallanma gerçekleştirilebilir.)
- Seçici ile bu adres, kontrol adres yazacına aktarılır.
- Dallanılan adresin gösterdiği yerde bulunan mikro buyruk, mikro işlemler üretir. Ve bu adres de arttırıcıya gönderilmiş olur.

## 7.3. Mikro Program Örneği

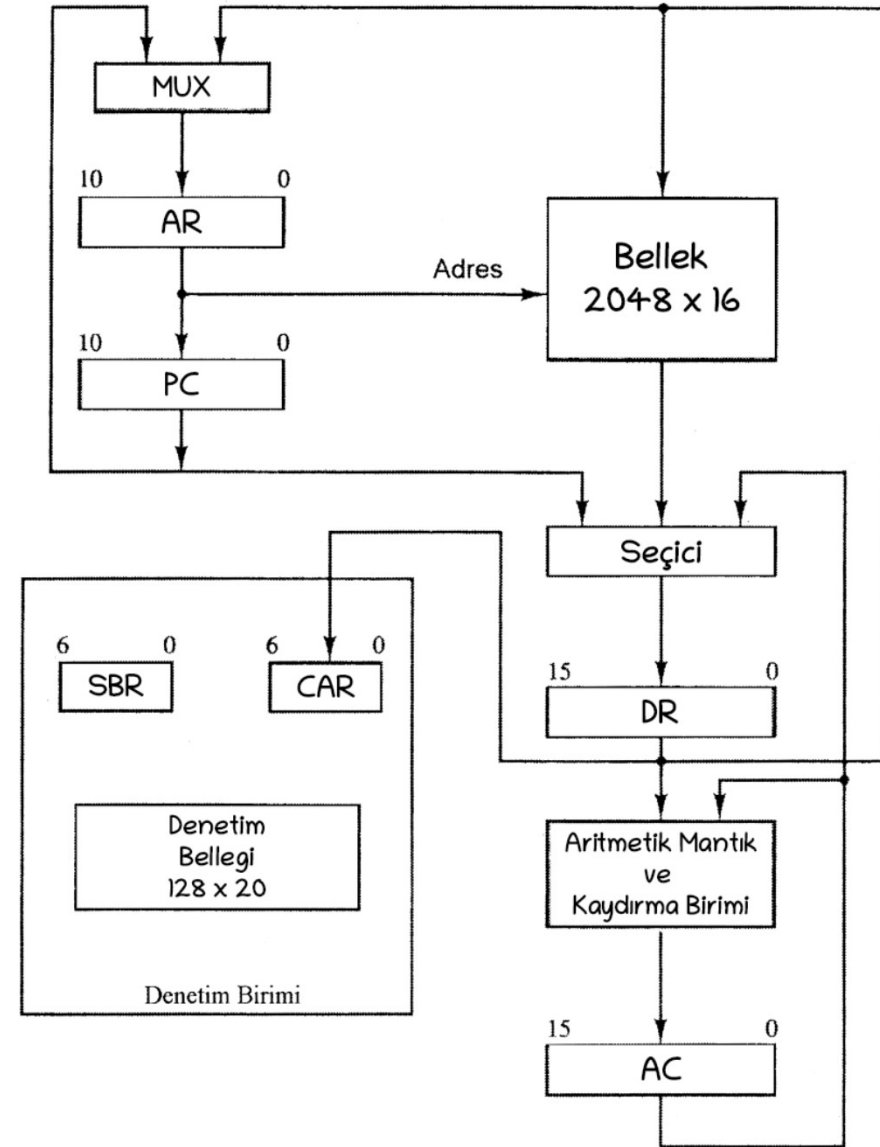
---

- Denetim belleğine, denetim mikro buyrukları için tasarımcı tarafından mikro kodlar yazılır.
- Bu kodlama işlemine mikro programlama denir.
- Mikro programlar, mikro buyrukları gerçekleştirirler.
- Program, 128 adrese sahip denetim belleğinde bulunur.

**Not:** Burada anlatılacak yapı, 5. bölümde anlatılan bilgisayara benzer olmakla beraber birebir özdeşi değildir.

**Yandaki şekilde bir bilgisayarın yapısı gösterilmiştir.**

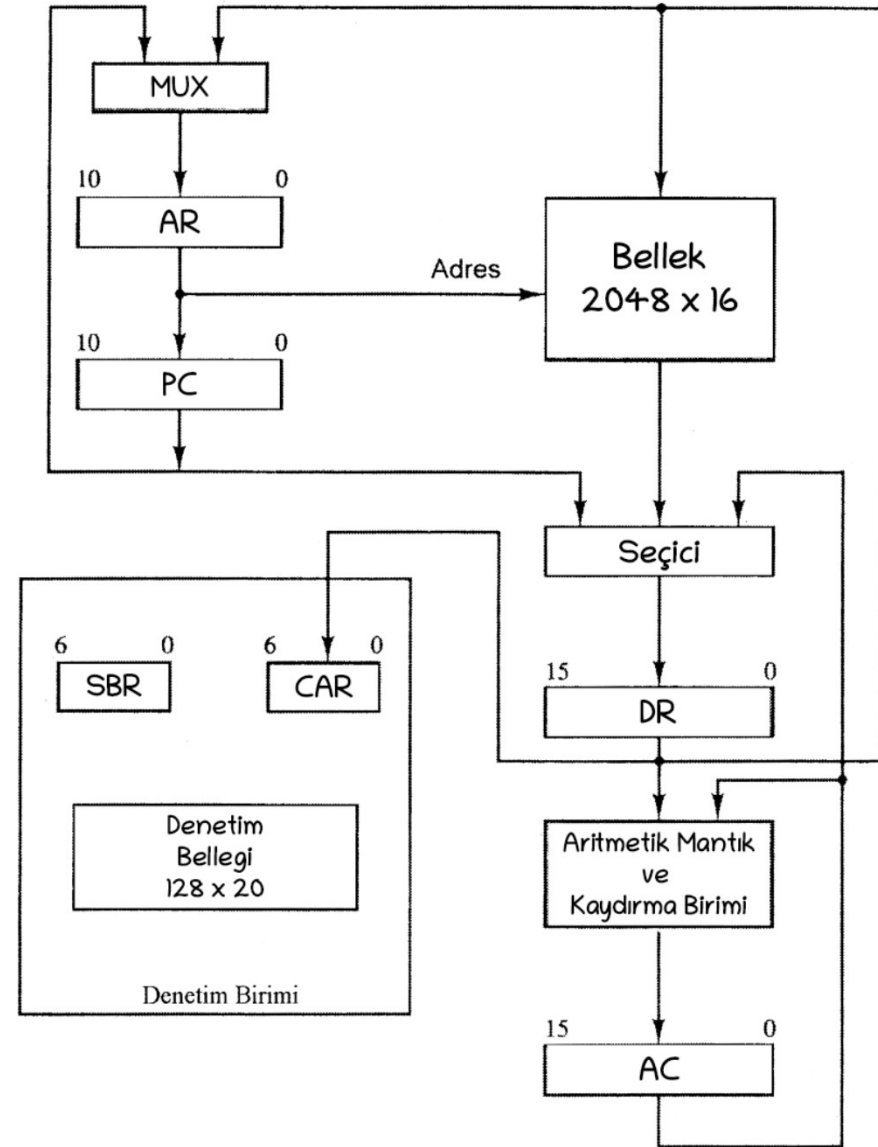
- DR, AR ve PC işlemci yazaçlarıdır. SBR ve CAR ise denetim biriminde bulunan yazaçlardır.
- Bu mimaride belleğimiz 16 bitlik 2048 adresten oluşmaktadır. Denetim Belleğimiz ise 20 bitlik 128 adrese sahiptir.
- DR yazacımız bellek, Akümülatör ya da PC yazacındaki veriyi barındırabilir.
- Kontrol adres yazacımız, dışarıdan gelecek adres için DR yazacından veri alabilir. (Haritalama)



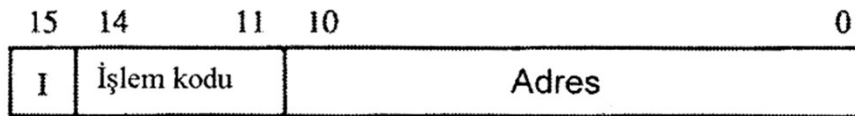


**Yandaki şekilde bir bilgisayarın yapısı gösterilmiştir.**

- Bellekte bilgisayar buyrukları ve veriler bulunur.
- Bu yapıdaki denetim belleğimiz 128 adresten oluşmaktadır.
- 128 adres için, adres bilgisi tutan SBR ve CAR yazacılarımız 7 bitlidir.
- SBR yazacı alt programa gidildiğinde dönülmesi gereken adresi bilgisini tutar.
- CAR yazacı denetim belleğindeki adresi gösterir.



## Bilgisayar buyruk biçimini hatırlayalım:



(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

- Yanda 16 bitlik bir bilgisayar buyruğu biçimi ile birlikte 4 bilgisayar buyruğu örneği ve işlem kodları gösterilmiştir.
- Bir bilgisayar buyruğu işlem kodundan ve adres bilgisinden oluşmaktaydı.

## 20 bitlik mikro buyruk biçimi:



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

- F1, F2 ve F3 alanlarında, icra edilecek 3 bitlik mikro işlem kodları bulunur.
- CD alanında dallanma şartı ve BR alanında dallanma alanı bulunur.
- AD, 7 bitlik adres alanıdır.

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	$PC \leftarrow PC + 1$	INCPC
110	$PC \leftarrow AR$	ARTPC
111	Ayrılmış	

**F1, F2 ve F3 alanlarına yazılabilecek 3 bitlik mikro işlem kodları yandaki çizelgede gösterilmiştir.**

- Bir mikro buyruk aynı anda F1, F2 ve F3 mikro işlemlerinden birer tane bulundurarak eş zamanlı işlem gerçekleştirebilir. Fakat tek mikro buyrukta F1, F2 veya F3 mikro işlemlerinden birden fazla bulunamaz.
- Örneğin aynı mikro buyrukta F3'teki 000 ve 011 işlemleri aynı anda bulunamaz, çünkü 3 bitlik F3 işlem alanına yalnızca bir adet F3 mikro işlemi yazılabilir. Bir başka F3 mikro işlemi sonraki mikro buyrukta belirtilebilir.

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	$PC \leftarrow PC + 1$	INCPC
110	$PC \leftarrow AR$	ARTPC
111	Ayrılmış	

Aşağıdaki örnek için gerekli F1, F2, F3 işlem kodlarını yazalım:

$DR \leftarrow M[AR]$

$PC \leftarrow PC + 1$

İlk işlem adresteki veriyi veri yazacına aktarma mikro işlemi yürütür ve bu işlem F2 işlem kodları arasında tanımlıdır.

İkinci işlem PC yazacını 1 arttırma işlemidir ve F3 işlem kodları arasında bulunur.

F1 kodlarından herhangi bir işlem bulunmadığından None mikro işlemi seçilir.

Üçer bitlik işlem kodu alanları şöyle olur:

F1   F2   F3  
000   100   101

## İki bitlik CD alanı işlem kodları şöyledir:

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

**CD alanı dallanma işleminin gerçekleşip gerçekleşmeyeceğine karar verir.**

- **CD = 00 ise** dallanma şarta gerek duyulmaksızın gerçekleşecektir.
- **CD = 01 ise** DR yazacının 15. bitine (dolaylı adres biti) bakılması gerektiğini belirtir ve BR'deki işlem bu bitin 0 ve 1 olmasına göre gerçekleşir.
- **CD = 10 ise** AC yazacının 15. bitine (işaret biti) bakılması gerektiğini belirtir ve BR'deki işlem bu bitin 0 ve 1 olmasına göre gerçekleşir.
- **CD = 11 ise** AC yazacının içeriğine bakılır. İçeriği 0 ise şart 1'dir, değilse şart 0'dır ve BR'deki işlem bu şartın 0 ve 1 olmasına göre gerçekleşir.

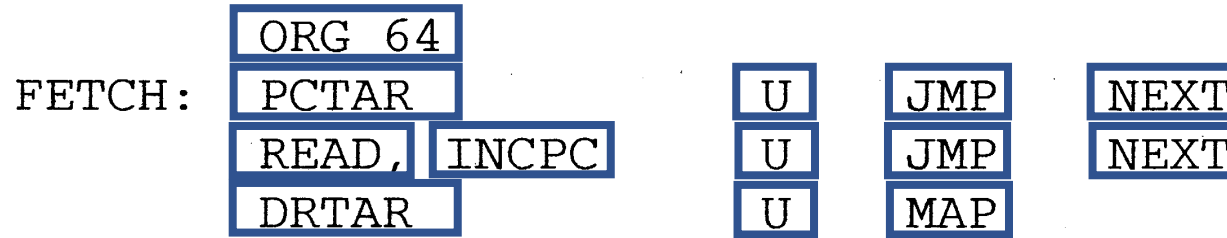
## İki bitlik BR alanı işlem kodları şöyledir:

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	$CAR \leftarrow SBR$ (Alt programdan geri dönüş)
11	MAP	$CAR(2-5) \leftarrow DR(11-14), CAR(0,1,6) \leftarrow 0$

**BR alanı, CD şartına bağlı olarak hangi tip dallanma işleminin gerçekleşeceğini belirler.**

- **BR = 00 ise** CD'nin belirlediği şart biti 1 olması durumunda AD alanının içeriğini CAR'a atacak böylece adres yazacındaki adrese gidilecektir. (**şartlı ve şartsız dallanma**)
- **BR = 01 ise** CD'nin belirlediği şart biti 1 olması durumunda o anki CAR değerinin bir fazlasını (kaldığı yeri) alt program yazacına (SBR) yazacak, AD alanının içeriğini CAR'a atarak adres yazacındaki adrese gitmiş olacaktır. (**alt programa gitme**)
- **BR = 10 ise** CD'nin belirlediği şart biti 1 olması durumunda alt program yazacı (SBR) içeriğini CAR'a atarak kalınan adrese gidilmesini sağlayacaktır. (**alt programdan dönme**)
- **BR = 11 ise** dallanılacak adres haritalama yöntemine göre CAR yazacına yazılacak ve ilgili adrese gidilmesi sağlanacaktır. (**haritalama**)

## Al-getir yordamıyla anlamaya çalışalım:



Bu yordamımız denetim belleğinde 64. adresten başlıyor ve üç mikro buyruktan oluşuyor (64., 65., ve 66. adreslerde)

Mikro buyruklardaki ilgili mikro işlemler için ikilik kelime karşılığını bulmak için F1, F2, F3, CD ve BR tablolarındaki ikilik değerleri ile kelimeleri oluşturalım:

İkili 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



**Sembolik  
gösterim:**

```

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

```

**İkilik  
gösterim:**

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

- Bunun gibi bir çok yordam bilgisayar buyruklarını gerçekleştirmek üzere bir araya gelerek denetim belleğindeki denetim programını oluştururlar.

**Denetim programlarının yordamların bir araya gelmiş haliyle sembolik ve ikilik gösterimlerini inceleyelim.**

Aşağıda, mikro buyrukların gerçekleştirdiği yordamlardan oluşan sembollerle ifade edilmiş bir mikro program örneği verilmiştir:

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
	ACTOR	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	

**Bu sembolik programın her adreste 20 bitlik ikilik karşılıklarıyla denetim belleğindeki hali ise şöyledir:**

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

## 7.4. Denetim Birimi Tasarımı

---

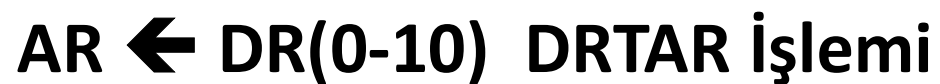
- Denetim birimi tasarlarken mikro işlem sayısı denetim kelimelerinin fonksiyonlarının bit uzunluğuna bağlıdır. Örneğin F1 alanı 3 bit olduğundan  $2^3$  adet mikro işlem tanımlanabilir.
- Mikro işlem belirleyen alanların aldığı değerlere göre mikro işlemler icra edilmelidir. Bu nedenle her alana gelen değerlere göre bir takım işlemleri gerçekleştirecek yapılar oluşturulmalıdır.

## 7.4.1. F1, F2 ve F3 Alanlarının Kodunun Çözülmesi

---

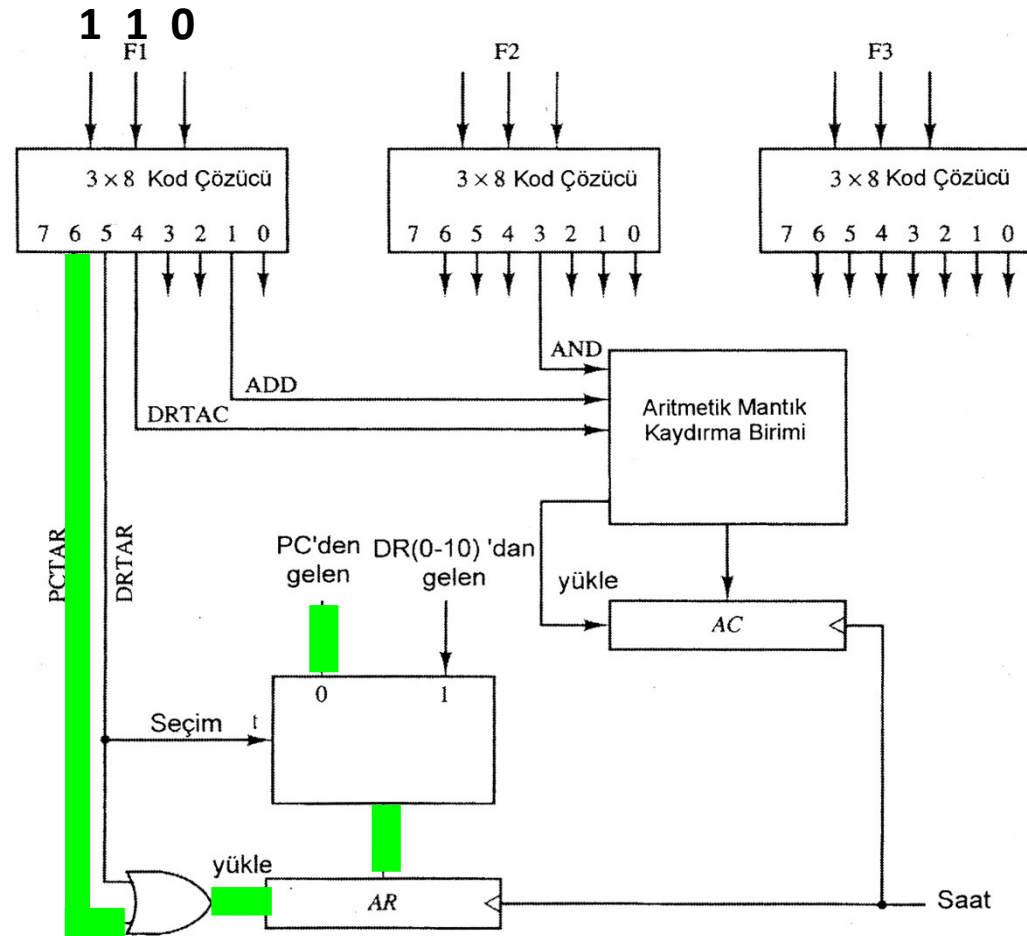
- Denetim kelimemizde F1, F2, F3 şeklinde üç alan vardı. Bu alanlar farklı mikro işlemler gerçekleştirebilen değerler alarak buyrukta en az bir, en çok üç mikro işlem gerçekleştirebiliyorlardı.
- Bunu sağlamak için her alan için 3x8 kod çözücü kullanılır. Bu çözücüler, alanlardaki tanımlamalara göre işlemin gerçekleştirilmesini başlatacak yere bağlıdır.

- Örneğin F1 seçme ucunun 101 olduğunu düşünelim
- Seçme bitleri kod çözücünün 5. çıkışını seçerek 1 gönderir.
- Çıkıştan gelen 1, veya kapısından geçerek AR yazacına yazmayı aktifler.
- Aynı zamanda seçiciye gelen 1, DR yazacının ilgili bitlerini çıkışa aktarır.
- Böylece F1'de 101 koduyla tanımlı DRTAR işlemi gerçekleştirilerek DR yazacının ilgili bitleri AR yazacına aktarılmış olur.



**Yandaki şekilde F1, F2 ve F3 alanlarının 3x8 kod çözücüler ile ilgili işlemleri gerçekleştirmek üzere çıkışlarının bağlantıları kabaca gösterilmiştir.**

- F1 seçme ucuna 110 geldiğini düşünelim
- Seçme bitleri kod çözücünün 6. çıkışı seçerek 1 gönderir.
- Çıkıştan gelen 1, veya kapısından geçerek AR yazacına yazmayı aktifler.
- Seçicinin seçme ucu 0 olduğundan PC yazacının içeriğini AR yazacına aktarmış olur.
- Böylece F1'de 110 koduyla tanımlı PCTAR işlemi gerçekleştirilerek PC yazacının içeriği AR yazacına aktarılmış olur.



## AR ← PC PCTAR işlemi

## 7.4.2. Mikro Program Sıralayıcı

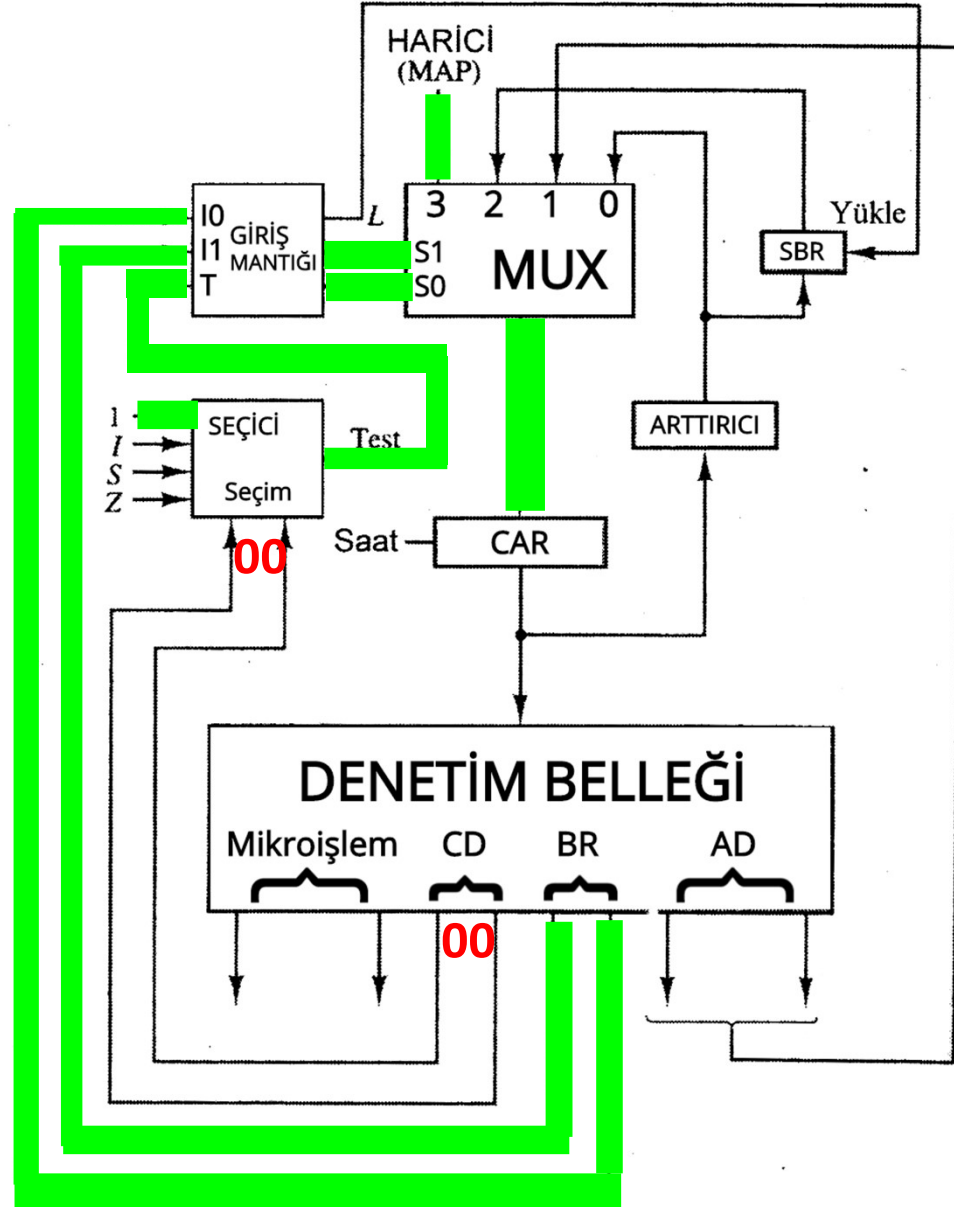
---

- Mikro programlanmış denetim biriminin temel yapısını oluşturan unsurlardan biri de adres seçimini gerçekleştiren devrelerdir.
- Sıralayıcı özel işlevleri yerine getirmek üzere tasarlanır ve adresleri buna göre belirler.
- Buna göre bir mikro program sıralayıcı ve adres belirleme işlemlerini devre üzerinde inceleyelim.



**Yandaki şekilde denetim belleğindeki denetim kelimesine göre adres belirleme işlemi kabaca gösterilmiştir.**

- Örneğin CD alanının 00 olduğunu düşünelim.
- Seçicinin seçme uçlarına 00 geleceğinden, çıkışına şart olmadan direkt 1 aktarır. (BR=10 ve BR=11 için çıkışındaki değer önemsizdir.)
- BR alanından da 11 geldiğini varsayalım.
- Buna göre BR kod tablosundan da hatırlanacağı üzere 11 gelme durumuyla birlikte giriş mantığı 11 üreterek haritalama yöntemiyle gelen bitlerin bağlı olduğu muxların 3. girişinin bitlerini seçer.
- Böylece haritalama işlemiyle belirlenen adres seçilerek adres yazacına (CAR) yüklenmiş olur.



**Bu devre için doğruluk değerleri aşağıdaki çizelgede gösterilmiştir.**

BR Alanı		Giriş			MUX 1		SBR yükle
		$I_1$	$I_0$	$T$	$S_1$	$S_0$	$L$
0	0	0	0	0	0	0	0
0	0	0	0	1	0	1	0
0	1	0	1	0	0	0	0
0	1	0	1	1	0	1	1
1	0	1	0	x	1	0	0
1	1	1	1	x	1	1	0

**Not:** BR=10 (alt programdan dönüş) ve BR=11 (haritalama) durumları için CD alanındaki değerler ve bağlı olduğu seçicinin çıktısındaki 1 ile 0 değeri önemsizdir. Bu nedenle çizelgede ilgili bitler x ile gösterilmiştir.