



Yazılım

Prof. Dr. Eşref ADALI

Doç. Dr. Şule Gündüz Öğüdücü

Sürüm-B

Konular

- Yazılımın Temelleri
- Makine Dili ve Simgesel Dil
- Simgesel Dilde Örnek Programlar
 - Aritmetik İşlemler Üzerine Örnekler
 - Koşullu ve Koşulsuz Dallanma Üzerine Örnekler
 - Sayaç, Çevrim, Geciktirme Örnekleri
 - Ara Açma
 - Bellek Sınama
 - Blok Aktarma
 - Şifreleme
 - Kümeler üzerinde İşlemler
 - Altprogramlı Örnekler
 - Sıralama
 - Giriş-Çıkış ile ilgili Örnekler
 - Kesme Üzerine Örnekler

Yazılım

- Donanım olarak var olan bir bilgisayarın yapacağı işleri, yazılımın yaptıracağı bilinmektedir.
- Mikrobilgisayarlar yüksek düzeyli dillerde programlanabileceği gibi, düşük düzeyli dillerle de programlanırlar. Örneğin, simgesel dilde veya makine dilinde.
- Bir MİB'nin var olduğunda bildiği programlama dili Makine Dilidir. Makine dilinde program yazmak çok zor olduğundan, simgesel dilde program yazmaya geçilmiştir.
- Örneğin ACCA'ı \$1000 sayılı bellek gözü ile yüklemek istediğimizde, makine dilinde yazılacak program (Örnek MİB için) şöyle olabilir:

00 20 10 00

- Aynı program simgesel dilde şöyle yazılabilir:

YÜK A, <1000>

İkinci Derece Denklem Çözümü

İkinci derece denklemi elle çözmek için izlenecek adımlar:

- **1. Aşama** : Ele aldığımız sorun, bilgisayar yardımıyla çözülebilir. Çünkü:
 - a. Aritmetik olarak denklemler bilinmektedir.
 - b. Denklemlerin çözümü vardır
- **2. Aşama** : denkleminin çözümünde ilk olarak, gerçel köklerin bulunup bulunmadığı araştırılır. Bu nedenle önce diskriminant hesaplanır.
- Diskriminantın pozitif olması durumunda iki tane gerçel kök vardır ve bunlar şu iki formülle hesaplanır.
- Eğer diskriminantın değeri eksi çıkarsa, bu durumda iki tane sanal kök vardır ve bu iki kökün hesabında şu iki bağıntı kullanılır.

$$ax^2 + bx + c = 0$$

$$\Delta = b^2 - 4ac$$

$$x_1 = (-b + \sqrt{\Delta}) / 2a$$

$$x_1 = (-b - \sqrt{\Delta}) / 2a$$

$$x_1 = (-b + j\sqrt{-\Delta}) / 2a$$

$$x_1 = (-b - j\sqrt{-\Delta}) / 2a$$

İkinci Derece Denklem Çözümü

İkinci derece denklemi bir bilgisayar programı ile çözmek istersek izlenecek adımlar

- $a = 0$ ise, bir sıfıra bölme ile karşılaşılır. Bilgisayar aritmetiğinde sıfıra bölme tanımlanmış değildir ve yanlışla neden olur.
- Ancak pozitif sayıların karekökleri alındığında gerçel sayılar bulunur. Bu nedenle, karekök alma deyimi sadece pozitif sayılar için geçerlidir. Diskriminantın değeri negatif ise karekök alma işleminde yanlışla düşülecektir.

1. Adım : a, b, c' yi oku

2. Adım : $a = 0$ ise tek kök vardır ve bu kök

$$x = \frac{-c}{b}$$

3. Adım : A sıfır değil ise, $B/(2 \times A)$ değerini ve diskriminantı hesapla

$$gerçell = \frac{b}{2a} \quad \Delta = b^2 - 4ac$$

4. Adım : Diskriminantın değeri eksi ise, bu durumda kökler sanaldır. Sanal köklerin hesabı için, önce diskriminantın salt değerini bul ve bulunan sayının karekökünü al.

$$sanal = \frac{\sqrt{-\Delta}}{2a} \quad \begin{aligned} x_1 &= gerçell + jsanal \\ x_2 &= gerçell - jsanal \end{aligned}$$

5. Adım : Diskriminantın değeri pozitif ise, gerçel iki kök vardır ve bunlar şöyle hesaplanır.

$$\begin{aligned} x_1 &= gerçell + \frac{\sqrt{\Delta}}{2a} \\ x_2 &= gerçell - \frac{\sqrt{\Delta}}{2a} \end{aligned}$$

6. Adım: Sonuçları yazdır.

Kesme

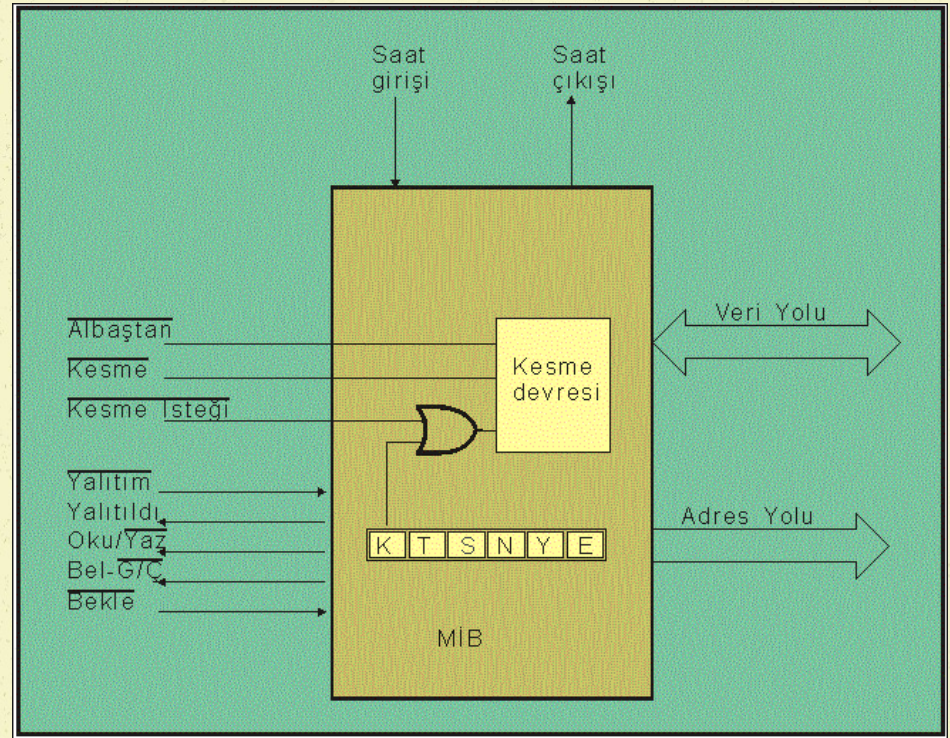
Kesme, bilgisayarın normal çalışmasını, bir süre için kesip, bir başka işe koşturmak ve bu iş bitince de eski işine kaldığı yerden devam etmek olarak tanımlanabilir. Bir bilgisayarda temel olarak üç kesme bulunur:

- Albaştan
- Kesme
- Kesme isteği

Albaştan : Bilgisayarı ilk durumuna getirir.

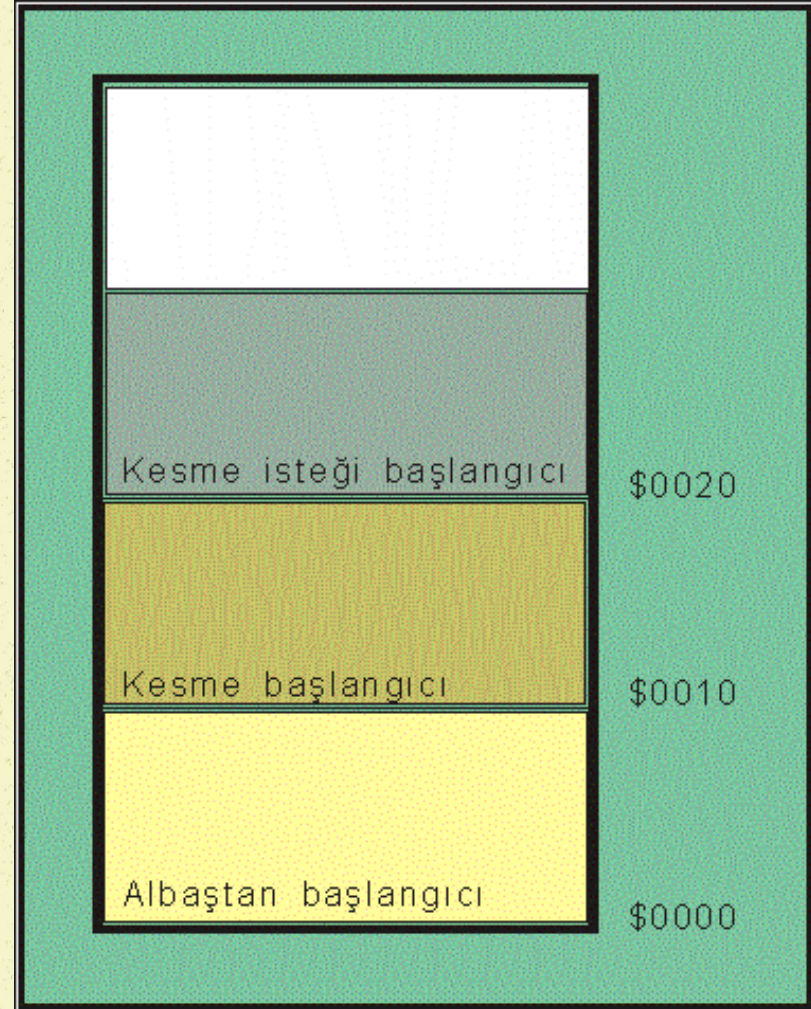
Kesme : O anda yürütülen programın akışını keser; kesme hizmet programına dallanır.

Kesme isteği : Eğer izin verilirse, o anda yürütülen program akışı kesilir; kesme isteği hizmet programına dallanılır.



Doğrudan Adres Belirtme

- Bu yöntemde, kesme hizmet programlarının başlangıç adresleri sabittir. Örneğin, Albaştan için başlangıç adresi \$0000, Kesme için \$0010 ve Kesme İsteği için \$0020 olabilir.
- Buna göre, albaştan edildiğinde, PS \$0000 adresine eşitlenir ve \$0000 adresinden başlayan program çalışmaya başlar.
- Kesme girişi etkin hale geldiğinde ana programın çalışması kesilir ve \$0010'dan başlayan program çalıştırılmaya başlanır.
- Kesme İsteği girişi etkin olduğunda ise, \$0020'den başlayan programa dallanılır.



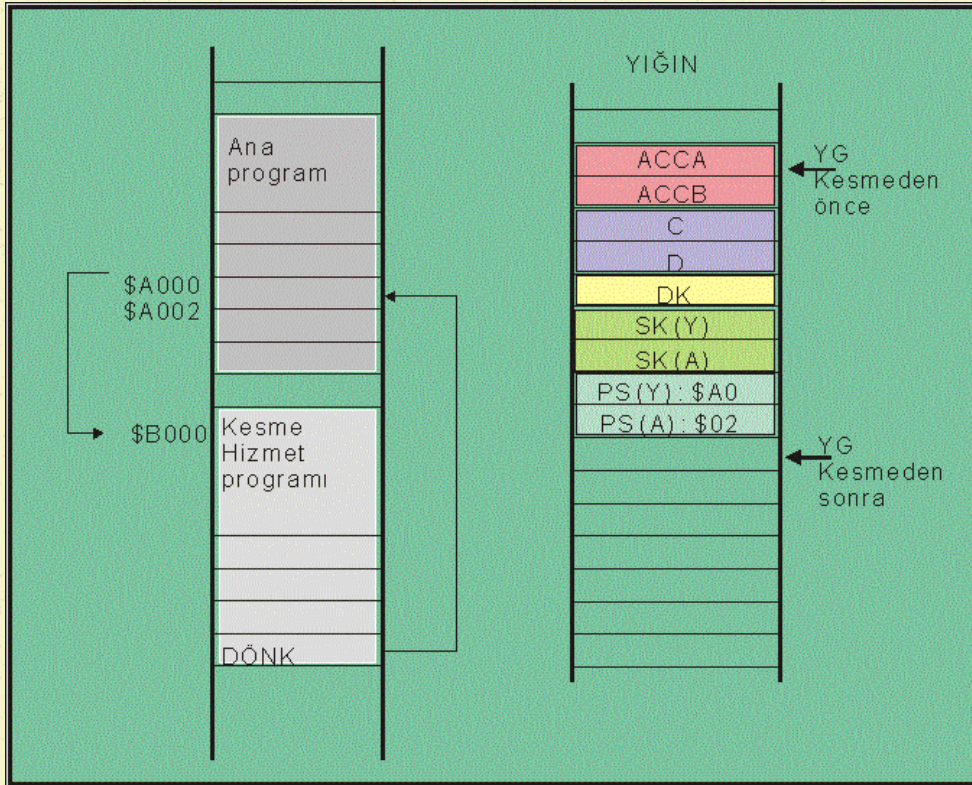
Dolaylı Adres Belirtme

- Kesme hizmet programlarının başlangıç adresleri, belleğin belli gözlerine yazılarak belirtilir. Örneğin;
- Albaştan hizmet programının başlangıç adresi, \$FFFE ve \$FFFF
- Kesme hizmet programının başlangıç adresi \$FFFC ve \$FFFD
- Kesme isteği hizmet programının başlangıç adresi \$FFF8 ve \$FFF9

Albaştan	(Y)	\$FFFF
Albaştan	(A)	\$FFFE
Kesme	(Y)	\$FFFD
Kesme	(A)	\$FFFC
Yazılım kesmesi	(Y)	\$FFFB
Yazılım kesmesi	(A)	\$FFFA
Kesme isteği	(Y)	\$FFF9
Kesme isteği	(A)	\$FFF8
		\$FFF7
		\$FFF6
		\$FFF5

Kesme Hizmet Programından Dönüş

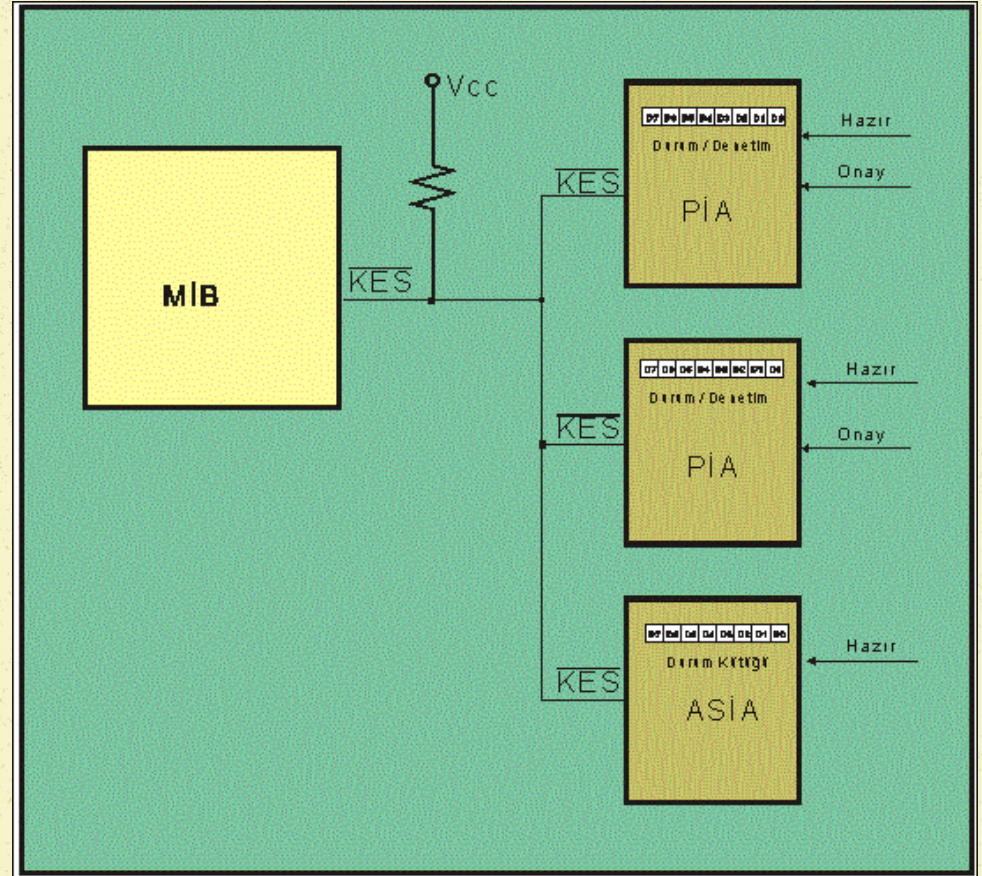
- Kesme geldiğinde ne yapılacağı, kesme hizmet programı ile bellidir. Ancak kesmenin ne zaman geleceği belli değildir. Kesme geldiğinde, kesme hizmet programına gidilir; hizmet verildikten sonra ana programa geri dönülür.
- Kesme hizmet programına dallanılmadan önce, MİB içindeki kütüklerin değerleri saklanmalı ve kesme hizmet programından döndüğünde bu değerler geri yazılmalıdır.



- Bazı bilgisayarlarda, MİB içindeki kütüklerin değerleri, MİB tarafından yığına atılır ve geri dönüşte yığından geri çekilerek yerlerine yazılır. Bu tür bilgisayarlarda kesme hizmet programının son buyruğu **DÖNK** olur.
- Bazı bilgisayarlarda MİB içindeki kütüklerin değerlerinin saklanması, kullanıcıya bırakılır. Bu tür bilgisayarlarda kesme hizmet programının son buyruğu, altprograminkinin aynıdır: **DÖN**

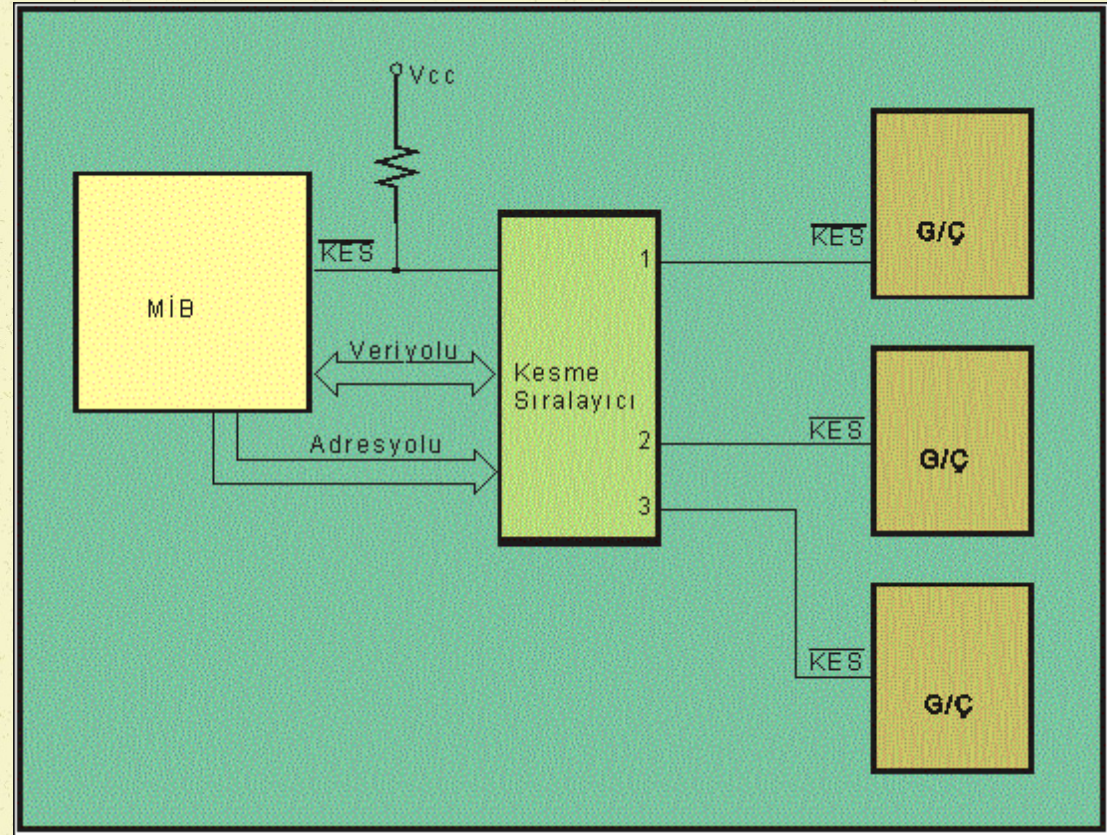
Kesmenin Donanım Yapısı

- Kesme gönderen çok sayıda birimin çıkışları aynı kesme girişine bağlanabilir.
- Bu yöntemde, kesme çıkışları bir bağlamalı VEYA kapısı oluşturacak biçimde MİB kesme girişine bağlanır.
- Bunun sonucu olarak herhangi bir giriş/çıkış arabiriminden gelen kesme, MİB'de kesme olarak yorumlanacak ve kesme hizmet programına gidilecektir.
- Kesme hizmet programına gidildiğinde, yapılacak ilk iş kesmenin nereden geldiğini saptamaktır. Bunun için, sırayla her giriş/çıkış arabiriminin durum kütüğü okunur ve hangisinde kesme ile ilişkili bayrağın çekilmiş olduğu araştırılır.



Kesme Önceliđi

Kesme gönderen birimlerin çıkışlarını toplayan, bunları öncelik sırasına koyan ve kesmenin nereden geldiđini de belirleyen devreler de üretilmektedir. Kesme öncelik devresi veya kesme sıralayıcı diye adlandırılan bu devre, kesme gönderen birim sayısının çok olduđu ve kesmenin önemli olduđu uygulamalar için uygun düşmektedir.



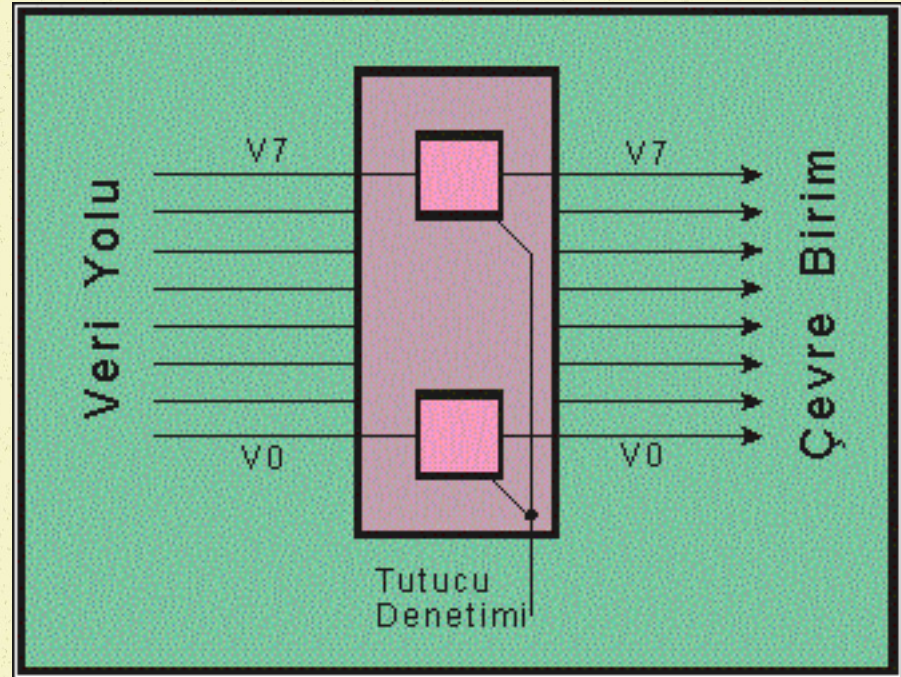
Yazılım Kesmesi

- Donanım kesmesine benzer bir yapı, yazılım için de uygulanmaktadır. Yazılım kesmesi, dışarıdan donanım yoluyla gelen bir kesme olmayıp, bir buyruktur. KES komutu yazılım kesmesi üretir.
- Programın herhangi satırında KES buyruğuna rastlandığında, yazılım kesmesi hizmet programına dallanılır. Kesme hizmet programının yeri, donanım kesmelerininkine benzer biçimde tanımlanır. Yazılım kesmesi hizmet programından ana programa dönüş, donanım kesmesinin aynıdır.
- KES buyruğunun program içindeki yeri, önceden belli olduğu için, yazılım kesmesinin, gerçek kesme anlamından farklı olduğu düşünülebilir. Çünkü gerçek kesmenin, geliş zamanı, programdan bağımsızdır. Bir başka görüşle, yazılım kesmesi sıradan bir altprogram olarak da değerlendirilebilir. Yazılım kesmesini, sıradan bir altprogramdan farklı kılan, yığını kullanma biçimidir. Yazılım kesmesine dallanıldığında MİB içi kütüklerin değerleri yığına atılır; kesme hizmet programından dönüşte geri çekilirler. İkinci fark, yazılım kesmesinin başlangıç adresi program dışında tanımlanmış olmasıdır.

İlkel Verici PIA'nın Tasarımı - I

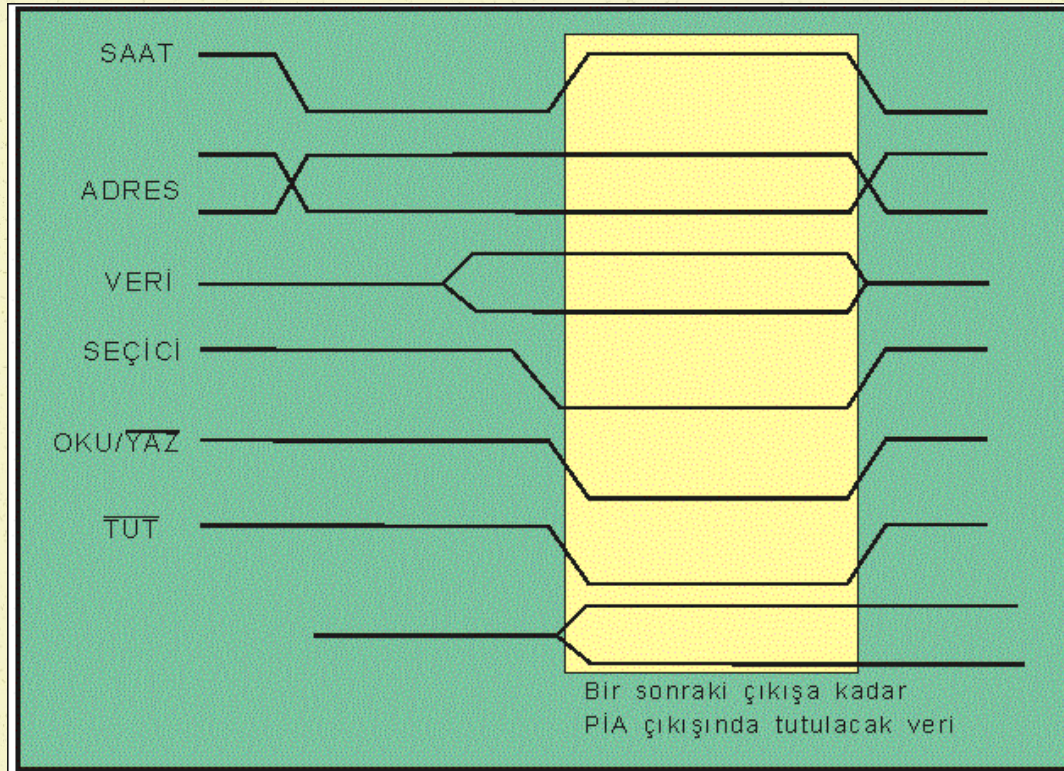
İlkel verici PIA'nın tasarımı, ilkel alıcı PIA'nın tasarımına benzer biçimde yapılabilir. İlkel verici PIA olarak kullanılabilecek en uygun mantık elemanı 74LS373 olarak bilinen tutucu elemandır. Bu elemanın girişine uygulanan veri, tutulabilmekte ve tutulan veri elemanın çıkışına aktarılmaktadır. Tutucu çıkışındaki veri, yeni bir verinin tutulup çıkışa aktarılmasına kadar aynı kalmaktadır.

1. Aşama : PIA'nın girişi, veri yoluna bağlanır. Tutucu devrenin çıkışı ise, bilgisayarın dış dünyasına açıktır.



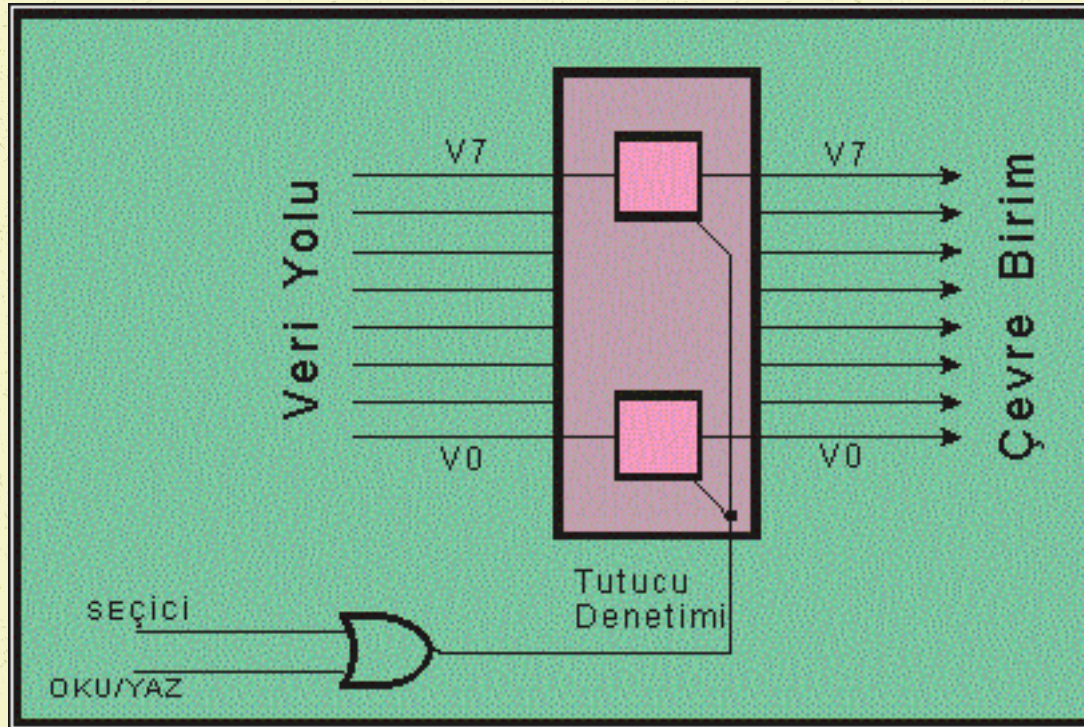
İlkel Verici PIA'nın Tasarımı - II

2. Aşama : Tutma işaretinin oluşturulmasına çalışılır. TUT işareti, YAZ , SAAT ve SEÇİCİ işaretlerinden oluşturulur. TUT işareti, Şekilden görüldüğü gibi, verinin veri yolu üzerinde bulunduğu sürede, saat işaretinin 1 olduğu konumda oluşturulmalıdır. Bu amaçla seçici ile saat işareti VEYA'lanır.



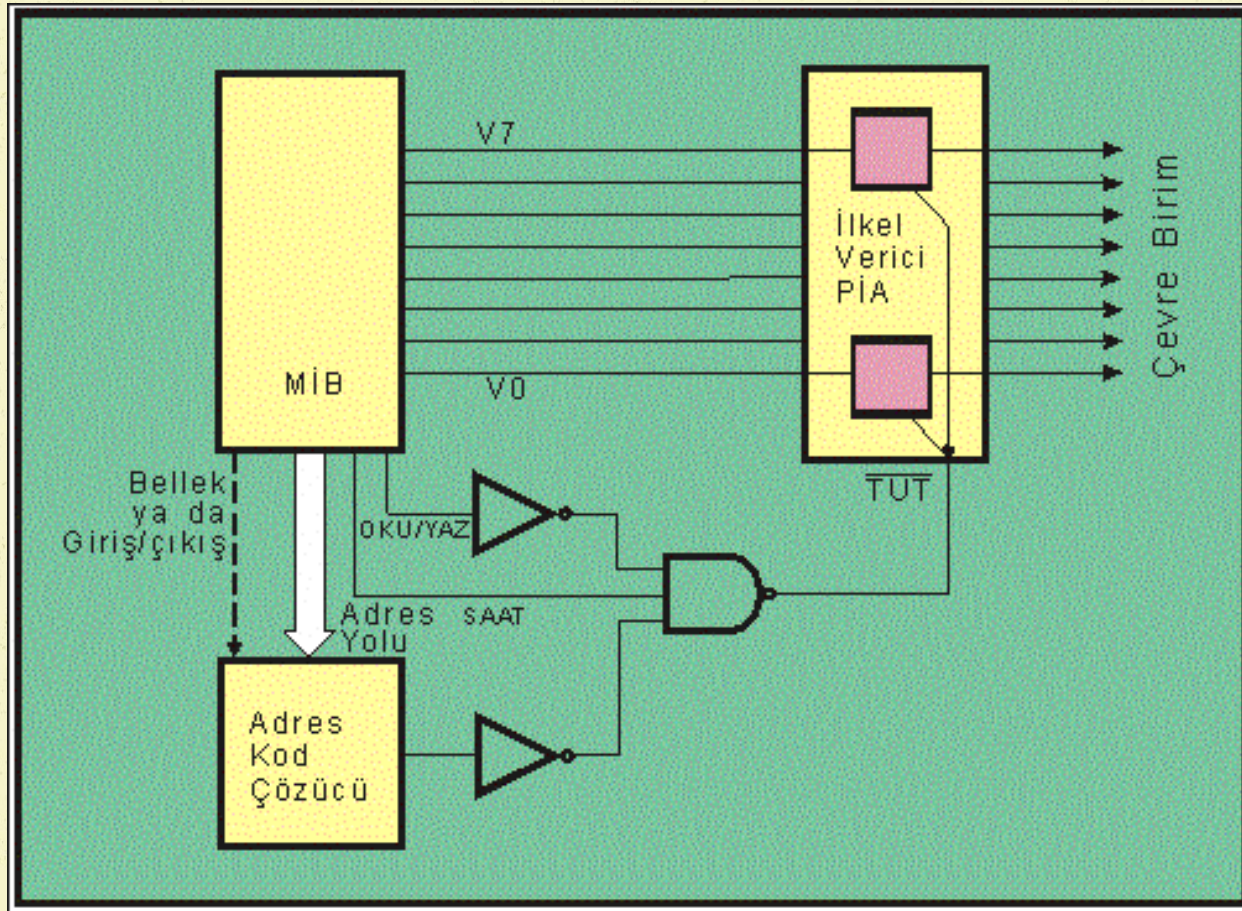
İlkel Verici PIA'nın Tasarımı - III

PIA'nın sadece çıkış olarak kullanılması nedeniyle, YAZ işareti var iken seçilmesi sağlanmalıdır. Zamanlamaya uygun olarak TUT işaretini oluşturacak olan donanım şöyledir:



İlkel Verici PIA'nın Tasarımı - IV

İlkel verici PIA'nın seçimini sağlayacak adres kod çözücü devrenin tasarımı, alıcı PIA için tasarlananın aynı olacaktır.



İlkel PIA'ya İlişkin Programlama

Bellek içinde yer alan ilkel alıcı ve verici PIA'ların adresi aynı olabilir. Örneğin \$8080.
Alıcı ilkel PIA'nın girişindeki veriyi okumak için;

YÜK A,<\$8080>

ACCA'daki veriyi, ilkel verici PIA'nın çıkışına göndermek için;

YAZ A, \$8080

Bellek dışında yer alan ilkel alıcı ve verici PIA'ların adresi aynı olabilir. Örneğin \$80.
Alıcı ilkel PIA'nın girişindeki veriyi okumak için;

GİR A,<\$80>

ACCA'daki veriyi, ilkel verici PIA'nın çıkışına göndermek için;

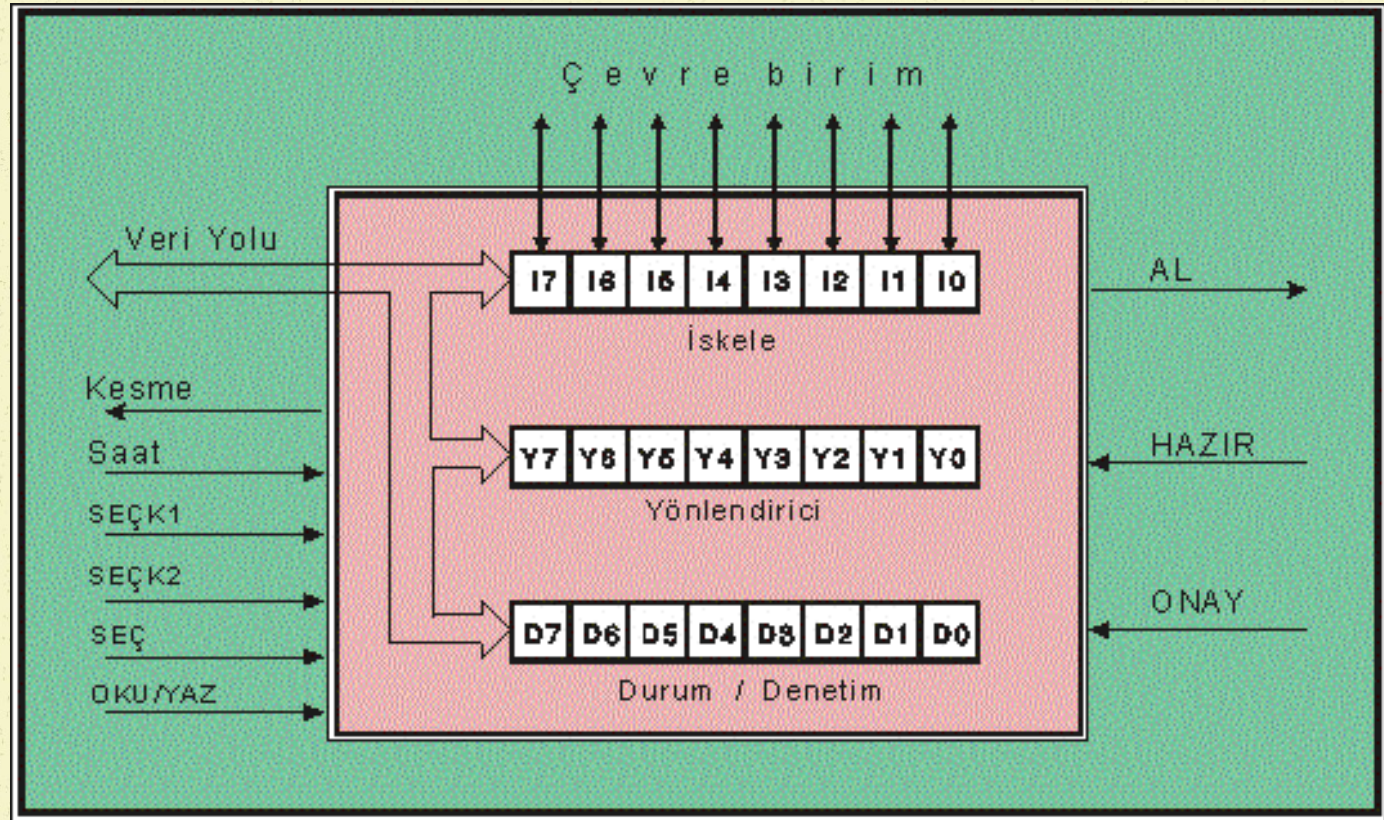
ÇIK A, \$80

buyruğunu yazmamız yeterli olur.

Gelişmiş PİA - I

PİA'nın gelişmişliği, alıcı ve verici olarak iki yönlü kullanılabilmesi ve yön belirleme işleminin yazılım yoluyla sağlanmasından kaynaklanmaktadır. Gelişmiş bir PİA içinde, genelde şu birimler bulunur:

- İskele
- Yönlendirici
- Durum Kütüğü
- Denetim Kütüğü



Gelişmiş PIA - II

İskele

PIA'nın, çevre birime bağlantısını sağlayan birimdir. Bilgisayar ile çevre birim arasındaki veri alışverişi bu birim üzerinden yapılır. Mal yükleme ve boşaltma işlemlerinin yapıldığı iskeleye benzer bir görevi üstlenmiş olması nedeniyle, bu birime İskele diyoruz. 8-bitlik bilgisayarlarda, iskelenin 8 bağlantıya olanak sağlaması uygun düşer. Bu nedenle, 8-bitlik bilgisayarlar için üretilen PIA'lar içindeki iskelelerde 8 kapı bulunmaktadır.

Gelişmiş PIA'nın iskelesindeki kapıların her biri alıcı veya verici olarak konumlandırılabilirler. Bu konumlandırmaya iskeleyi koşullama demekteyiz.

Yönlendirici

İskele kapılarının alıcı ya da verici olarak koşullanması için kullanılan bir kütüktür. Yönlendirici içinde, iskelede bulunan kapı sayısı kadar bit bulunur ve yönlendiricinin her bir biti, kendi sırasındaki kapıyı alıcı ya da verici olarak konumlandırır. Sözgelimi, yönlendiricinin birinci bitine 1 yazılmış ise bu, iskelenin birinci kapısının verici olarak koşullandığını ve bu yönde çalışacağını belirtir. Yönlendirici içinde bir bite 0 yazılması ise bu bitin denetimi altında bulunan kapının alıcı olarak koşullandığını gösterir.

Gelişmiş PİA – III

Durum Kütüğü

El sıkışma bağlantılarının durumunu izlemek üzere ayrılmış bitlerden oluşur. Bağımsız bir kütük olarak düzenlenebileceği gibi, denetim bitleriyle birlikte Durum/Denetim kütüğü içinde de kurulabilir. Örneğimizde D6 ve D7 bitleri Durum Kütüğü bitleri olarak görev yaparlar.

Denetim Kütüğü

El sıkışma bağlantılarını istenen konumlara getirmek üzere gerekli olan bitler kümesidir. Örneğimizde D0, D1, D2, D3, D4, D5 Denetim Kütüğü bitleri olarak görev yaparlar.

- PİA içindeki kütüklerin seçilmesi için iki tane kütük seçici girişi bulunmaktadır (K0, K1).
- PİA'nın kendisini seçmek üzere ayrıca bir tane kırmık seçicisi (SEÇ) bulunmaktadır.
- PİA'nın çevre birime bağlantısı, iki yönde çalışabilen iskele kapıları üzerinden yapılır. Ayrıca el sıkışma bağlantıları bulunmaktadır. El sıkışma bağlantılarından ikisi alıcı (Hazır, Onay) ve diğeri verici (Al) olarak çalışmaktadır.
- PİA'da bir de, el sıkışma uçlarına bağlı olarak kesme üretecek çıkış (Kesme) bulunmaktadır. PİA içindeki işlemlerin senkronizasyonu için bazı PİA'larda saat girişi de bulunur.
- PİA içindeki kütüklere yazma ya da buraları okumak için Oku/Yaz işareti de PİA'ya bağlanır.

Gelişmiş PIA - Hazır Girişi

- Hazır girişi, çevre birimin hazır olup olmadığını anlamak için kullanılır. Durum/Denetim Kütüğünün 7. biti HAZIR el sıkışma bağlantısının durumunu belirtir. Bu bite genellikle durum bayrağı denilir. Bilindiği gibi, bayrağın etkin durumu 1 dir. Yani bayrak çekilmesi durumu, mantıksal 1 ile eş anlamlıdır.
- El sıkışma girişinin hangi konuma geçişinin, bayrağın çekileceği, Denetim Kütüğü içine yazılacak verilerle belirlenir. Bu amaçla, D0 ve D1 HAZIR girişini dolayısıyla, D7 bayrağını etkilemek üzere ayrılmıştır.

D1	D0	Hazır girişi	Kesme çıkışı
0	0	1' den 0' a indiğinde D7 bayrağı çekilir	1' de kalır, kesme üretmez
0	1	0' dan 1' e çıktığında D7 bayrağı çekilir	1' de kalır, kesme üretmez
1	0	1' den 0' a indiğinde D7 bayrağı çekilir	0' a iner, kesme üretir
1	1	0' dan 1' e çıktığında D7 bayrağı çekilir	0' a iner, kesme üretir

Gelişmiş PIA - Onay Girişi

- Çevre birimin, hazır olma durumuna ek olarak, kendisine gönderilen verileri alıp almadığı da sorgulanmak istenebilir. Bu nedenle, ikinci bir el sıkışma bağlantısı gerekebilir. Örnek PIA'da bu girişin adı ONAY girişidir. ONAY girişinin çalışma ilkesi, HAZIR girişine çok benzer. ONAY girişini denetleyen bitler, D2 ve D3 tür. ONAY girişinin durumu D6 bayrağı ile gösterilir.

D3	D2	Onay girişi	Kesme çıkışı
0	0	1' den 0' a indiğinde D6 bayrağı çekilir	1' de kalır, kesme üretmez
0	1	0' dan 1' e çıktığında D6 bayrağı çekilir	1' de kalır, kesme üretmez
1	0	1' den 0' a indiğinde D6 bayrağı çekilir	0' a iner, kesme üretir
1	1	0' dan 1' e çıktığında D6 bayrağı çekilir	0' a iner, kesme üretir

Gelişmiş PIA – Aİ Çıkışı

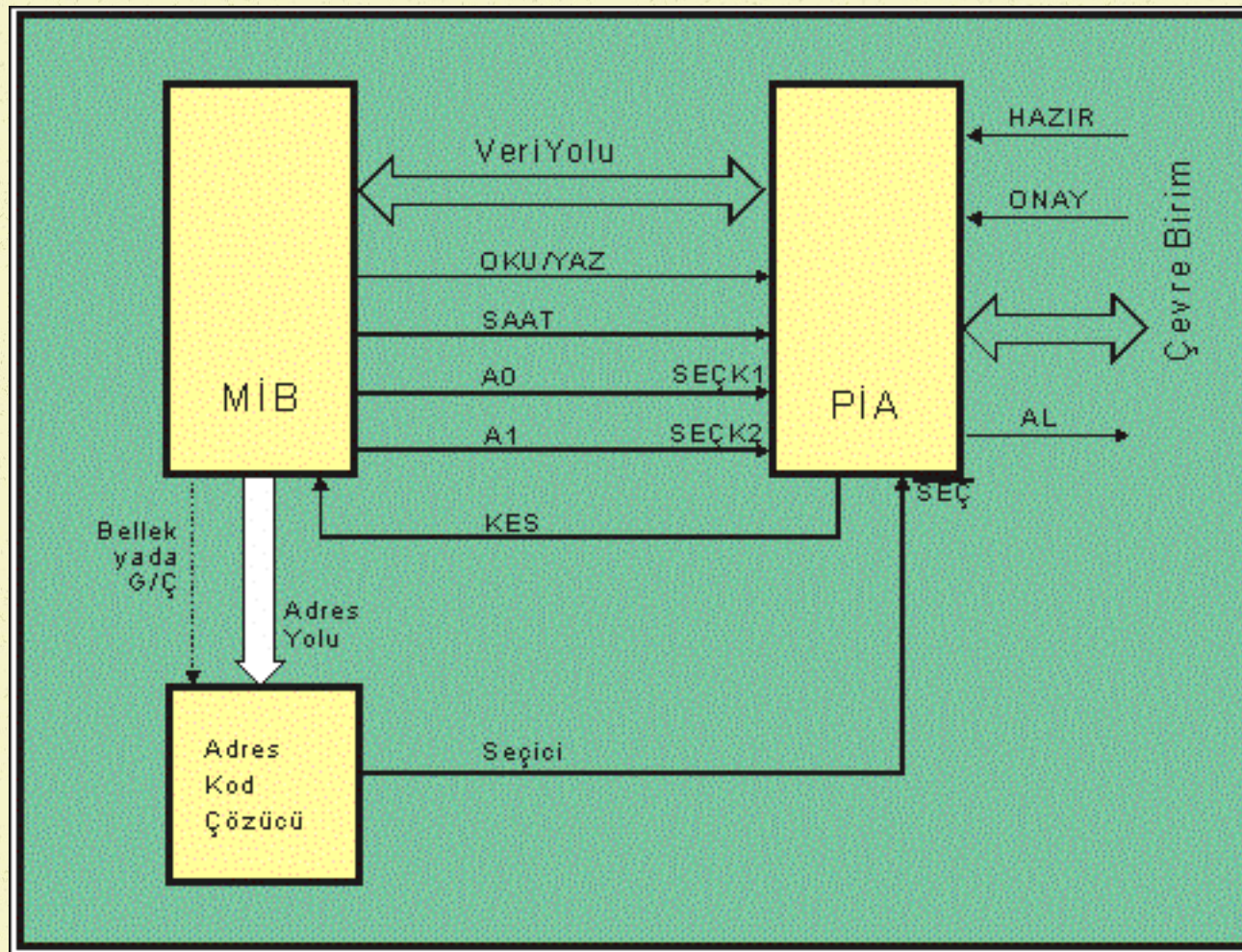
- AL ucu, çıkış biçiminde kullanılmak üzere tasarlanmıştır. Bu çıkış, PIA'nın durumunu, çevre birime iletmek için ya da çevre birime veri gönderildiği uyarısı için kullanılır.
- D5'in 0 olduğu durumlarda, D4'e yazılan değer aynen AL çıkışında görülür. D5'in 1 olduğu durumda ise, AL çıkışı, iskeleye veri yazılmasının hemen ardından, istenen konuma geçer. AL çıkışının kendiliğinden çalışması şöyle anlatılabilir: Eğer, iskeleye her veri yazıldığında, AL çıkışı 1'den 0'a çekilmek isteniyor ise D5=1 ve D4=0 yazılması yeterlidir. Bu uygulamada, iskeleye her veri yazılışının ardından AL çıkışı kendiliğinden sıfıra iner ve bir süre bu konumda kaldıktan sonra eski konumuna geri döner.

D5	D4	AL Çıkışı
0	0	0 konumuna getirilir
0	1	1 konumuna getirilir
1	0	Verinin iskeleye yazılmasının ardından 0 konumuna belli bir süre için iner
1	1	Verinin iskeleye yazılmasının ardından 1 konumuna belli bir süre için çıkar

Gelişmiş PIA – MIB Bağlantısı - I

- 1. Aşama :** PIA'nın veri yolu bilgisayar veri yoluna bağlanır.
 - 2. Aşama :** Saat ve OKU/YAZ denetim bağlantıları yapılır.
 - 3. Aşama :** İlkel PIA için gerçekleştirilen seçici devre, benzer şekilde tasarlanır.
- PIA içindeki kütüklerin seçimi için, bellek içindeki gözlerin seçiminde izlenen yol izlenebilir. Yani A0 ve A1 adres hatları, SEÇK1 ve SEÇK2 girişlerine bağlanabilir.
 - PIA'nın seçilmesi için adres kod çözücünün çıkışı kullanılır. PIA içindeki kütüklerin adresleri, bu temel adrese göre hesaplanır. Örneğin temel adresi \$8080 olan PIA'da, İskelenin adresi \$8080, Yönlendiricinin adresi \$8081 ve Durum/Denetim Kütüğünün adresi \$8082 olacaktır.
 - PIA'yı bellek haritası dışında tutan uygulamalar içinde çözüm aynıdır. Tek farkla, adres seçici devreye, bellek ya da giriş/çıkış arabirimi seçici denetim ucu katılmalıdır.

Gelişmiş PIA – MİB Bağlantısı - II



Gelişmiş PİA – Örnek Programlar

- İ0, İ1, İ2 ve İ3 kapıları alıcı ve diğer kapılar verici olsun
- HAZIR girişi, 1'den 0'a indiğinde, çevre birimin hazır olduğu algılsın ve bu durumda hem bayrak çekilsin hem de kesme üretilsin.
- ONAY girişi, 1'den 0'a indiğinde, çevre birimin onay verdiği algılsın, bu durumda sadece bayrak çekilsin.
- İskeleye veri yazıldığında, AL çıkışı kendiliğinden sıfıra insin.

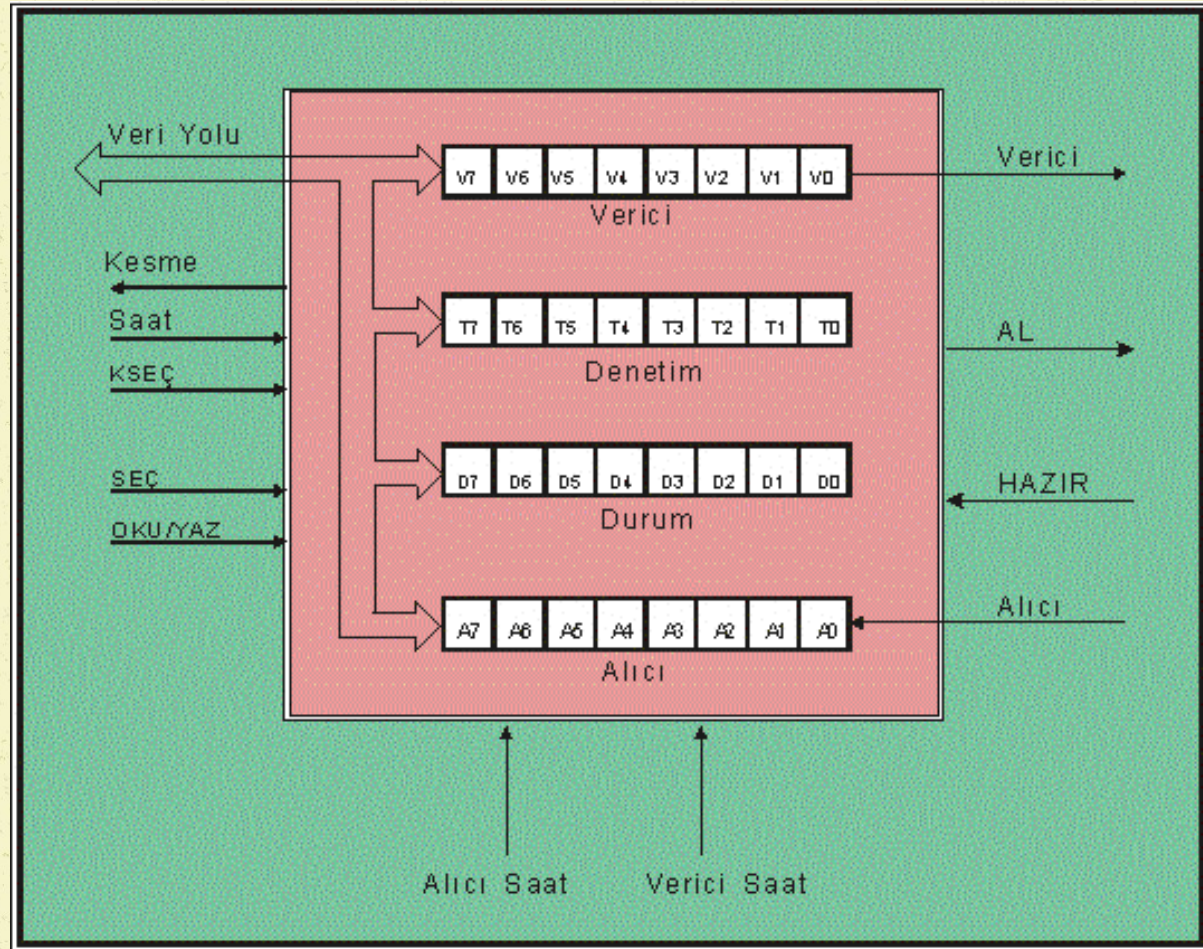
KOŞUL	YÜK A, %0010 0010	: Durum/Denetim kütüğü koşulu
	YÜK B, %1111 0000	: İskelenin üst yarısı verici, alt yarısı alıcı
	YAZ A, \$8082	: Durum denetim kütüğüne istekler yazıldı
	YAZ B, \$8081	: Yönlendiriciye istekler yazıldı.
	DÖN	

PİA koşullandıktan sonra, iskele içeriği her okunduğunda çevre birimden gelen bilgiler MİB'e alınır veya akümülatördeki bilgiler iskeleye yazılarak çevre birime gönderilir. PİA yeniden koşullanana kadar bir önceki koşullama geçerlidir.

Asenkron Seri İletişim Arabirimi

Asenkron Seri İletişim Arabirimi (ASİA) içinde dört temel birim bulunmaktadır

- Verici
- Alıcı
- Durum Kütüğü
- Denetim Kütüğü



ASİA - I

Verici

Verici, aslında paralel giriş ve seri çıkışı olan ötelemeli bir kütüktür. Veri yolu üzerinden paralel olarak gelen veriler, bu kütüğe yüklenir. Bunun ardından, gerekli başla, dur ve eşlik bitleri eklenir ve böylece oluşan kütük içeriği, seri olarak ötelenir ve çevre birime gönderilir. Ötelemenin hızını, yani gönderme hızını, verici saat işareti belirler.

Alıcı

Alıcı, vericinin tersine seri giriş paralel çıkışlı ötelemeli bir kütüktür. Çevre birimden gelen seri bilgiler, alıcıya adım adım yerleşir. Adım adım yerleşmenin hızını alıcı saat işareti belirler. Alıcıya yerleşen verinin, başla, dur ve eşlik bitlerinden arınmış kısmı, daha sonra bilgisayar veri yoluna aktarılır.

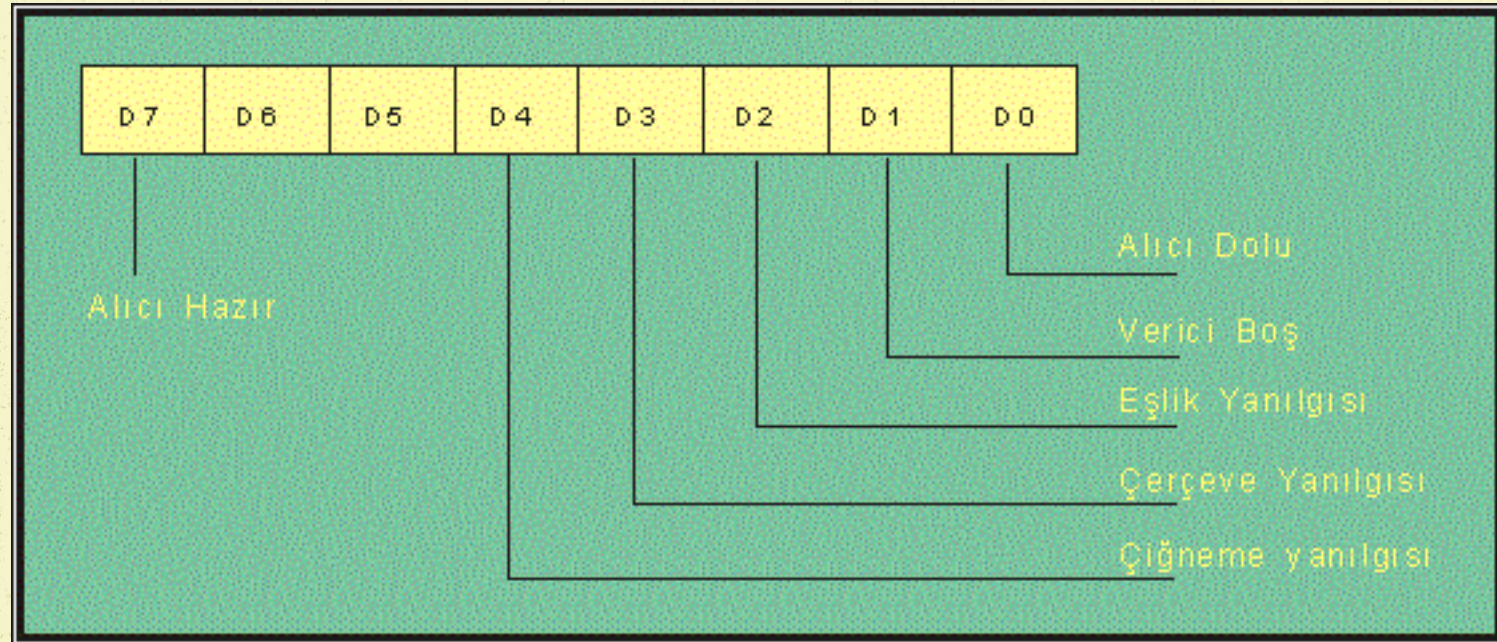
Durum Kütüğü

Durum Kütüğü, alıcıya veri gelip gelmediği, vericideki verinin gönderilip gönderilmediği, gelen verinin doğru alınıp alınmadığı ve çevre birimin hazır olup olmadığı gibi durumları belirtmek için kullanılır.

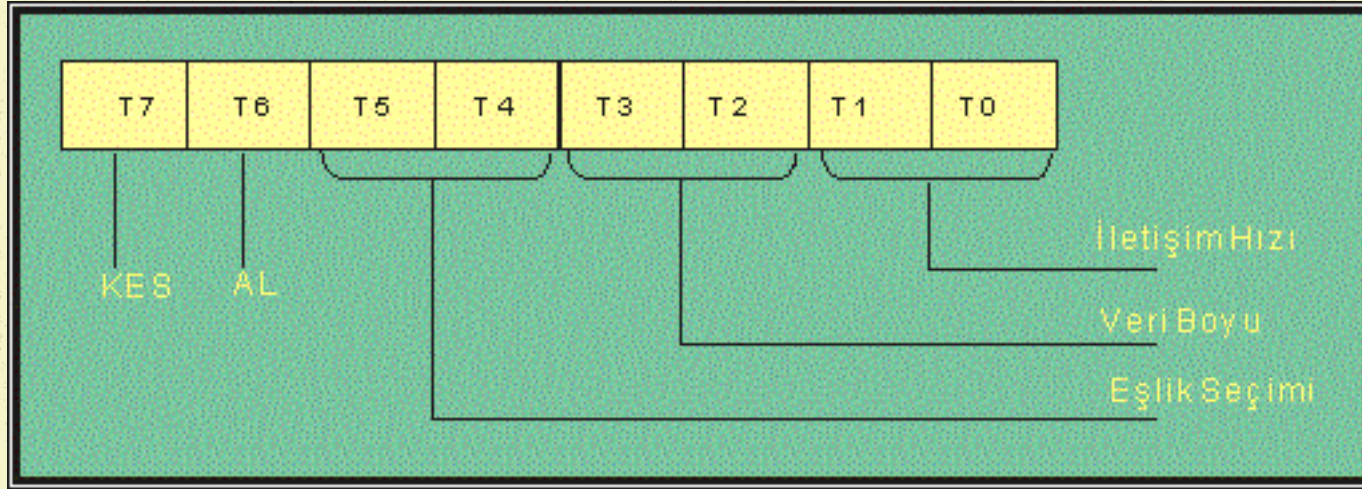
Denetim Kütüğü

Denetim Kütüğü, iletişim protokolunun yerine getirilmesi ve çevre birim ile uyumun sağlanması amacıyla kullanılır.

ASiA Durum Kütüğü



ASiA Denetim Kütüğü



T1	T0	İletişim hızı
0	0	1 / 1
0	1	1 / 2
1	0	1 / 4
1	1	1 / 8

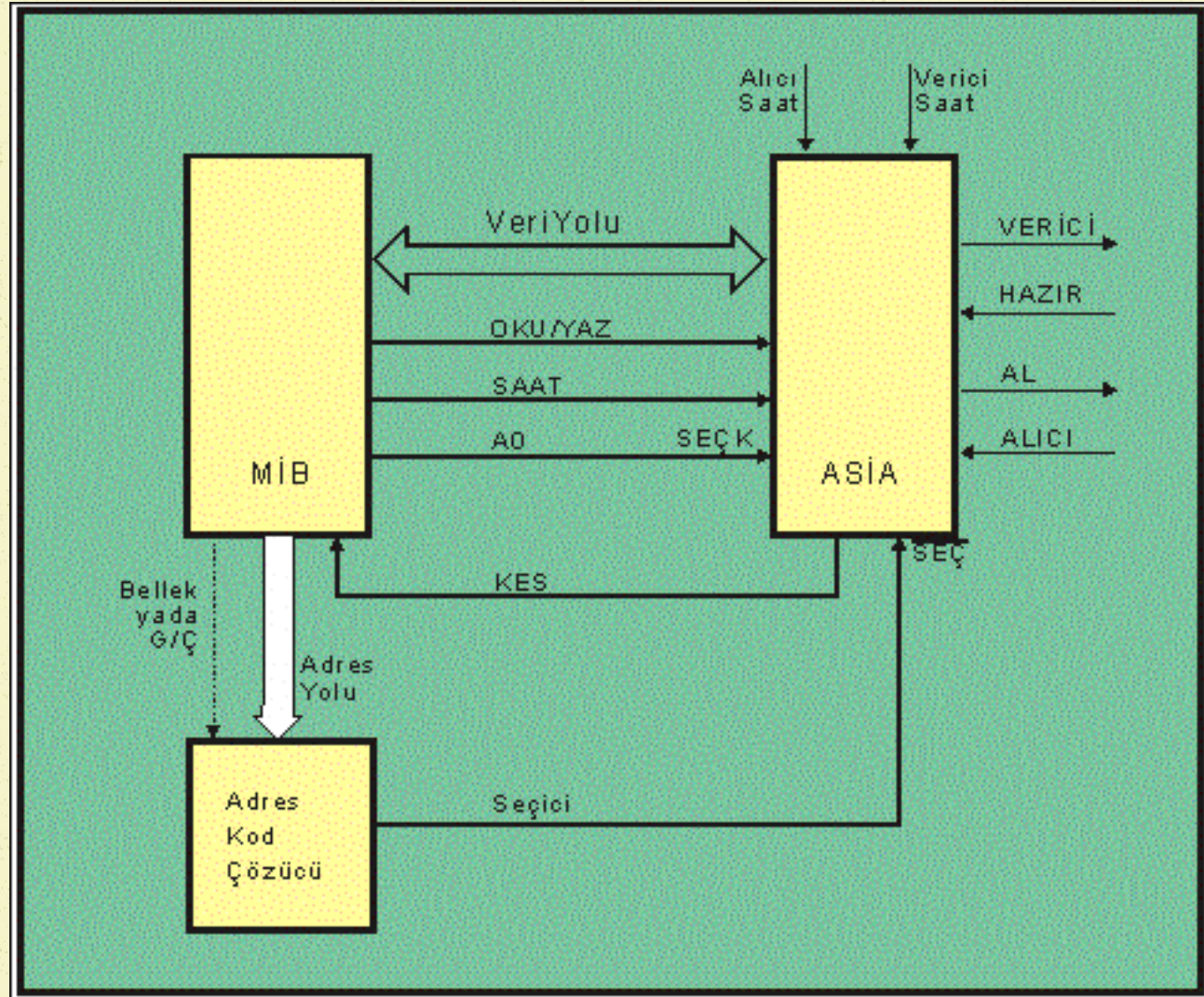
T3	T2	Veri boyu ve dur biti sayısı
0	0	7 veri biti + 1 dur biti
0	1	7 veri biti + 2 dur biti
1	0	8 veri biti + 1 dur biti
1	1	8 veri biti + 2 dur biti

T5	T4	Eşlik biti kullanımı
0	0	Eşlik biti kullanılmıyor
0	1	Tek eşlik biti
1	0	Çift eşlik biti
1	1	Geçersiz durum

ASİA – MİB Bağlantısı

- **1. adım** : ASİA'nın veri yolu bilgisayarın veri yoluna bağlanır.
- **2. adım** : ASİA seçici, temel adres seçicisine bağlanır.
- **3. adım** : ASİA içinde dört kütük olmasına karşın bunlardan Alıcı ve Durum Kütüğü sadece okunabilir ve Verici ile Denetim Kütüğüne sadece yazılabilir. Bu nedenle Alıcı ile Vericinin adresi aynı ve Durum ile Denetim Kütüğünün adresi aynı seçilebilir. Örneğin Alıcı ve Vericinin adresi ASİA temel adresine eşit ve Durum ve Denetim Kütüklerinin adresi temel adresten bir yukarıda olabilir. Bu durumda, Kütük seçici giriş A0 adres hattına bağlanacaktır.
- **4.adım** : Oku/Yaz girişi MİB OKU/YAZ çıkışına bağlanır.
- **5.adım** : İletişim hızını belirleyecek alıcı ve verici saat işaretleri, bu iş için tasarlanmış devreye bağlanır.

ASİA – MİB Bağlantısı



ASIA – Örnek Programlar

İletişim hızı : 1/4
Veri uzunluğu : 7 bit
Dur biti : 2
Eşlik : çift
Kesme : üretilecek

Alıcıyı okumak için, önce alıcının dolu olup olmadığına bakılmalı ve ardından, alıcı okunmalıdır.

Vericiden bir veri göndermeden önce, vericinin boş olup olmadığına bakılmalıdır. Bu amaçla, durum kütüğünün içeriğine bakılmalı, ikinci bitin 1 olup olmadığı sınanmalıdır.

YÜK A,%10100110
YAZ A,\$8081

YOKLA	YÜK	B, <Durum>
	VE	B, \$01
	DEE	YOKLA
	YÜK	A, <ALICI>
	DÖN	

YOKLA	YÜK B, <Durum>
	VE B, \$02
	DEE YOKLA
	YAZ A, VERİCİ
	DÖN