

18. 8255 PPI (PROGRAMMABLE PERIPHERAL INTERFACE)

8255 programlanabilir çevre arayüzü (Programmable Peripheral Interface - PPI) olarak isimlendirilen, üç adet genel amaçlı iskele barındıran, 8 bit veri genişliğine sahip, TTL uyumlu G/Ç birimlerini mikroişlemci ile bağlamak için kullanılan bir programlanabilir arayüz tüm devresidir.

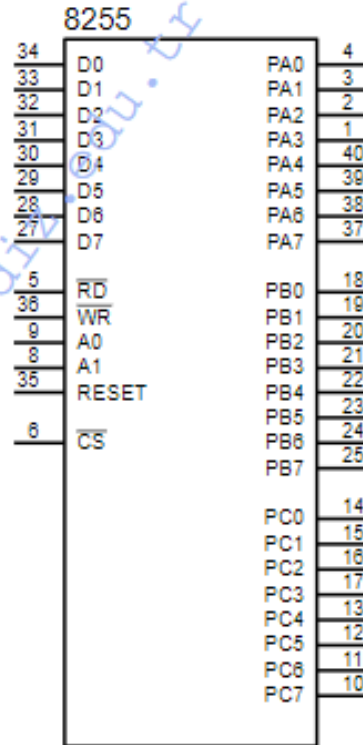
Genel kullanım amacı:

- temel paralel G/Ç (ör. yedi parçalı gösterge sürme)
- tetikleme temelli paralel G/Ç (ör. Kip 1)

olarak verilebilir.

18.1 8255 Uç Tanımları

8255 fonksiyonel uç tanımları Şekil 18-1 ile verilmektedir. 8255 sekiz adet veri (D7-D0), iki adet adres (A1, A0), okuma kontrol (\overline{RD}), yazma kontrol (\overline{WR}), yetkilendirme (\overline{CS}), reset uçlarına ve her biri sekiz uçlu üç adet iskeleye (PortA, PortB, PortC) sahiptir.



Şekil 18-1: 8255 fonksiyonel uç tanımları

Adres, okuma, yazma ve yetkilendirme uçlarının durumuna göre 8255'te gerçekleştirilen işlemler ve erişilen yazmaçlar Tablo 18-1 ile verilmiştir.

Tablo 18-1: 8255 adresleme ve yazmaç erişimi

A1	A0	$\overline{\text{RD}}$	$\overline{\text{WR}}$	$\overline{\text{CS}}$	İşlem (Okuma)
0	0	0	1	0	PortA → Veri yolu
0	1	0	1	0	PortB → Veri yolu
1	0	0	1	0	PortC → Veri yolu
1	1	0	1	0	Kontrol Kelimesi → Veri yolu
A1	A0	$\overline{\text{RD}}$	$\overline{\text{WR}}$	$\overline{\text{CS}}$	İşlem (Yazma)
0	0	1	0	0	Veri yolu → PortA
0	1	1	0	0	Veri yolu → PortB
1	0	1	0	0	Veri yolu → PortC
1	1	1	0	0	Veri yolu → Kontrol Kelimesi

18.2 8255 İç Yapısı

8255 iç yapısı incelendiğinde iskelelerin GrupA ve GrupB olarak fonksiyonel iki bölüme ayrıldığı görülmektedir. GrupA içerisinde PortA ve PortCH (PortC'nin yüksek anlamlı dört bit'i), GrupB içerisinde ise PortB ve PortCL (PortC'nin düşük anlamlı dört bit'i) yer almaktadır. 8255 iç yapısı; iç yollar, kontrol birimleri, ilgili veri ve tampon (buffer) yazmaçlarından oluşmaktadır.

PortA; sekiz bit çıkış yönlü tampon/mandal ve sekiz bit giriş yönlü mandal yazmaçları barındırır. Dolayısıyla PortA sekiz bit giriş veya sekiz bit çıkış birimi olarak kullanılabilir. PortB; sekiz bit çıkış yönlü tampon/mandal ve sekiz bit giriş yönlü mandal yazmaçları barındırır. Dolayısıyla PortB sekiz bit giriş veya sekiz bit çıkış birimi olarak kullanılabilir. PortC iki adet dört bit çıkış yönlü tampon/mandal ve iki adet dört bit giriş yönlü tampon barındırır. Dolayısıyla PortCL ve PortCH 4'er bit G/Ç birimleri olarak kullanılabilirler. PortC Kip 1 ve Kip 2'de kontrol ve durum yazmaçları olarak çalışır. Buna göre 8255'te iki adet sekiz bit'lik ve iki adet dört bit'lik iskeleler olduğu söylenebilir.

18.3 8255 Kipleri

8255 için dört farklı kip tanımlanabilir. Bunlar Bit Kipi, Kip 0, Kip 1 ve Kip 2'dir.

Bit Kipi PortC uçlarında lojik 0 veya lojik 1 değerleri oluşturmak için kullanılabilecek bir kiptir. Aynı zamanda Kip 1 kesme aktifleştirme bit'leri INTEA, INTEB ile Kip 2 kesme aktifleştirme bit'leri INTE1 ve INTE2 ayarlamada kullanılır.

Kip 0 basit giriş çıkış kipi'dir. LED, yedi parçalı gösterge, buton ve tuş tarama takımı sürme gibi tokalaşma (handshaking) gerektirmeyen giriş çıkış işlemlerinde kullanılır.

Kip 1 ve Kip 2 tokalaşma temelli kipleridir. Kip 1 tek yönlü olarak sadece giriş veya sadece çıkış olarak çalışırken, Kip 2 yürütülen komuta göre giriş veya çıkış yönlü çalışabilir.

18.3.1 Bit Kipi (Bit Set Reset Mode)

Bit Kipi (Bit Set Reset - BSR) 8255 PortC uçlarından herhangi bir bit'ine lojik 0 veya 1 yazmak için tanımlanmıştır. BSR kipi bir bit'lik temel çıkış için kullanılabilir. Kip 1 ve Kip 2'de bit bazında ayarlamalar yapmak için de kullanılır.

BSR kipini kullanabilmek için kontrol yazmacına Tablo 18-2 ile verilen uygun formattaki kontrol kelimesinin yazılması gerekmektedir.

Tablo 18-2: 8255 BSR kip kontrol yazmacı

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
0	X	X	X	B ₂	B ₁	B ₀	S/R
PortC'nin (B ₂ B ₁ B ₀) ₂ numaralı pin'inde Set (1) veya Reset (0) oluşturma							

Örnek 18-1

80H adresinden itibaren ardışık çift adreslere uygun adres çözümleme devresi ile yerleştirilmiş ve mikroişlemci ile bağlantıları uygun şekilde yapılmış 8255'te

- PortC2 ucunu lojik 1 olacak şekilde BSR kipi ile ayarlayınız.
- PortC6 ucunda görev oranı %66 olan kare dalgayı BSR kipi ile üretiniz.

```

MOV AL, 00000101B ; BSR kip, PortC2, set
OUT 86H, AL ; kontrol kelimesini yaz
TEKRAR: MOV AL, 0xxx1101B ; BSR kip, PortC6, set
OUT 86H, AL ; kontrol kelimesini yaz
CALL Delay ; Delay 207yordamı dönüş süresi kadar bekle
CALL Delay ; Delay yordamı dönüş süresi kadar bekle
MOV AL, 0xxx1100B ; BSR kip, PortC6, reset
OUT 86H, AL ; kontrol kelimesini yaz
CALL Delay ; Delay yordamı dönüş süresi kadar bekle
JMP TEKRAR ; sonsuz döngüde tekrarla

```

18.3.2 Kip 0

Temel G/Ç işlemlerinde örneğin doğrudan çıkış birimi (LED, yedi parçalı gösterge) sürme, doğrudan giriş birimi (buton, tuş tarama takımı) okuma işlemlerinde Kip 0 kullanılır. Kip 0 ile sekiz bit'lik PortA, sekiz bit'lik PortB, dört bit'lik PortC ve

²⁰⁷ Delay yerleşik/hazır bir yordam olmayıp, belirli bir süre gecikme sağlayacak şekilde yazılmış bir yordam olduğu varsayılmıştır

dört bit'lik PortCH'den her biri ayrı ayrı giriş veya çıkış yönlü olarak ayarlanabilir. 8255 Kip 0 kullanımı için Tablo 18-3 ile verilen uygun kontrol kelimesinin Kontrol Yazmacına yazılması gerekmektedir.

Tablo 18-3: 8255 Kip 0 (Temel G/Ç) kontrol yazmacı

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
1	0	0	PA	PC _H	0	PB	PC _L

PA	PB	PC _H	PC _L	Port Yönü
0	0	0	0	Çıkış
1	1	1	1	Giriş

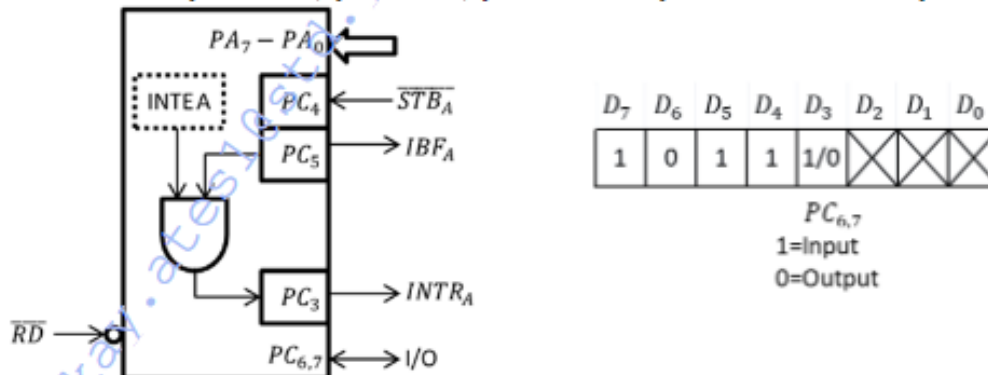
18.3.3 Kip 1

Tek yönlü tetikleme temelli paralel veri aktarımı için Kip 1 kullanılır. GrupA veya GrupB Kip 1'de kullanılabilir.

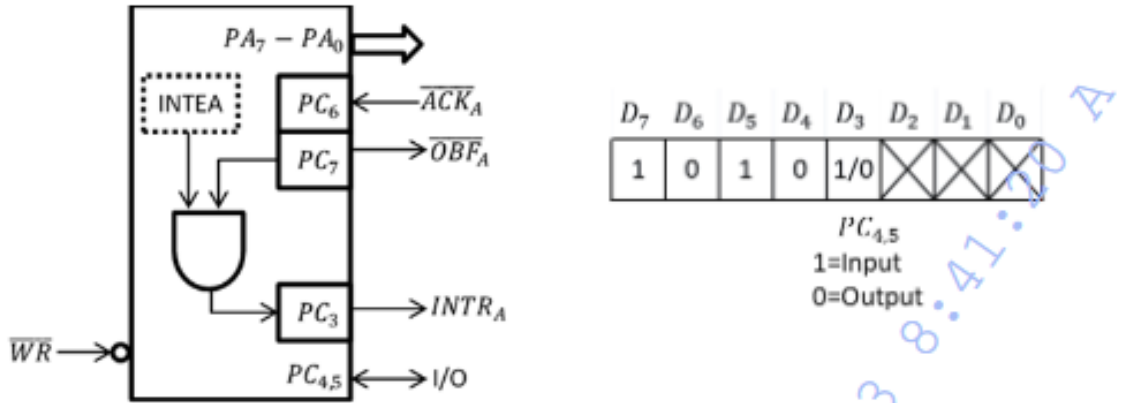
GrupA giriş yönlü olarak kullanılıyorsa veri PortA'dan okunurken, PortC3, PortC4, PortC5 kontrol uçları olarak çalışmaktadır. GrupA giriş olarak kullanıldığında boşta kalan PortC6 ve PortC7 ise G/Ç için kullanılabilir. GrupA çıkış yönlü olarak kullanılıyorsa veri PortA'dan gönderilirken, PortC3, PortC6, PortC7 kontrol uçları olarak çalışmaktadır. GrupA çıkış olarak kullanıldığında boşta kalan PortC4 ve PortC5 ise G/Ç için kullanılabilir.

GrupB giriş yönlü olarak kullanılıyorsa veri PortB'den okunurken, PortC0, PortC1, PortC2 kontrol uçları olarak çalışmaktadır. GrupB çıkış yönlü olarak kullanılıyorsa veri PortB'den gönderilirken, PortC0, PortC1, PortC2 kontrol uçları olarak çalışmaktadır.

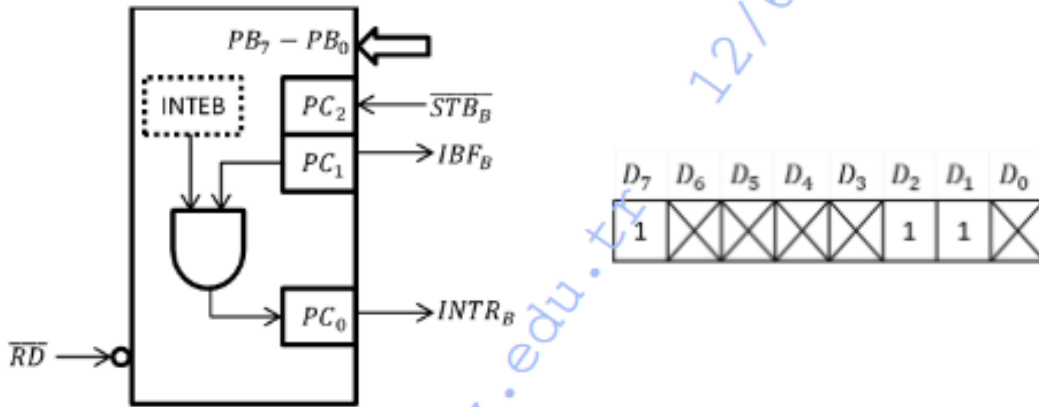
Kip 1 GrupA, GrupB için giriş, çıkış yönlü, 8255 uç tanımları ve uygun kontrol kelimeleri Şekil 18-2, Şekil 18-3, Şekil 18-4 ve Şekil 18-5 ile verilmiştir.



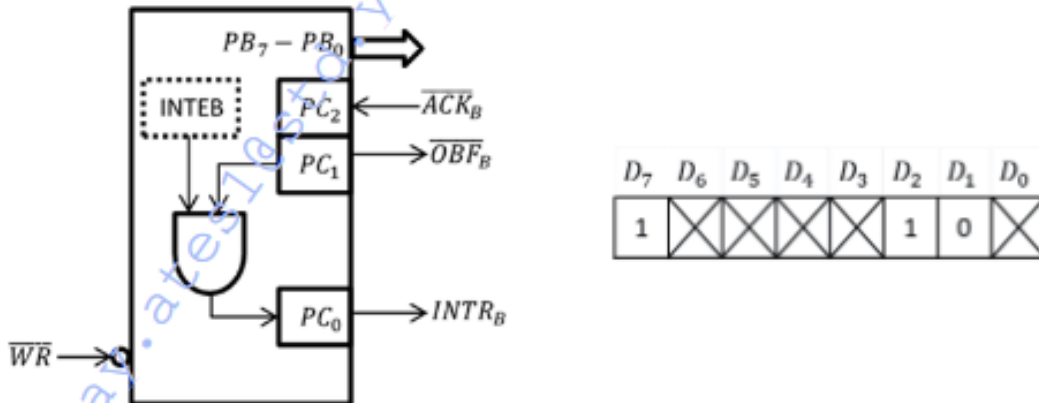
Şekil 18-2: 8255 kip 1 grupA giriş yönlü uç tanımları ve kontrol kelimesi formatı



Şekil 18-3: 8255 kip 1 grupA çıkış yönlü uç tanımları ve kontrol kelimesi formatı



Şekil 18-4: 8255 kip 1 grupB çıkış yönlü uç tanımları ve kontrol kelimesi formatı



Şekil 18-5: 8255 kip 1 grupB giriş yönlü uç tanımları ve kontrol kelimesi formatı

Kip 1 giriş yönlü kullanıldığında tetikleme temelli kontrol uçları ve görevleri şu şekilde oluşur;

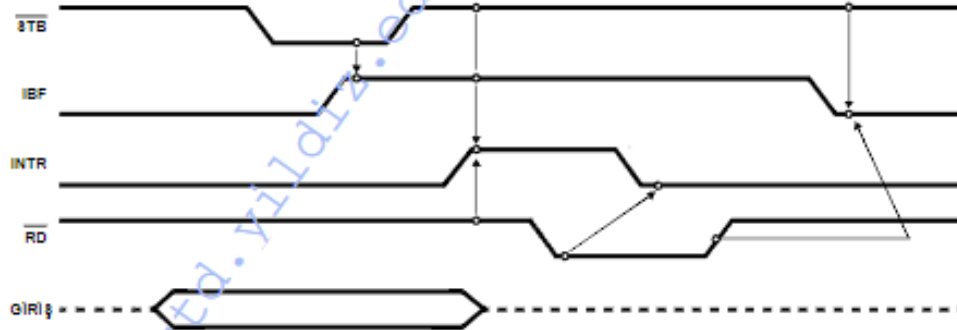
STB (Strobe): 8255 açısından bir giriş ucudur. Bu uçtaki lojik 0 seviyesi veri uçlarındaki değerin giriş yazmaçlarına kopyalanmasını başlatır.

IBF (Input buffer full): 8255 açısından bir çıkış ucudur. Bu uçtaki lojik 1 seviyesi verinin giriş yazmacına kopyalanmış olduğunu bildirir. Bu uç bir bakıma verinin alınmış olduğunu bildiren onay ucudur. IBF ucu, STB girişinin lojik 0 olmasından belirli bir süre sonra lojik 1 seviyesine geçer. IBF ucu, RD ucunun düşen kenarıyla ise lojik 0 seviyesine geçer.

INTR (Interrupt request): 8255 açısından bir çıkış ucudur. 8255'te verinin hazır olduğu 8086'ya bir kesme isteği ile bildirilmek istenirse bunun için INTR ucu kullanılabilir. INTR ucu, STB lojik 1, IBF lojik 1 ve INTE lojik 1 olduğunda lojik 1 seviyesine geçer. INTR ucu, RD ucunun yükselen kenarıyla ise lojik 0 seviyesine geçer.

INTE (Interrupt request enable): Kesme temelli veri okuma yapabilmek için INTE bit'lerinin BSR kipinde lojik 1 seviyesinde ayarlanmaları gerekmektedir. INTEA PortC4, INTEB ise PortC2 ile kontrol edilir.

Giriş yönlü Kip 1 için sinyalleşme yapısı Şekil 18-6 ile verilmektedir.



Şekil 18-6: Giriş yönlü kip 1 için sinyalleşme yapısı

Kip 1 çıkış yönlü kullanıldığında tetikleme temelli kontrol uçları ve görevleri şu şekilde oluşur:

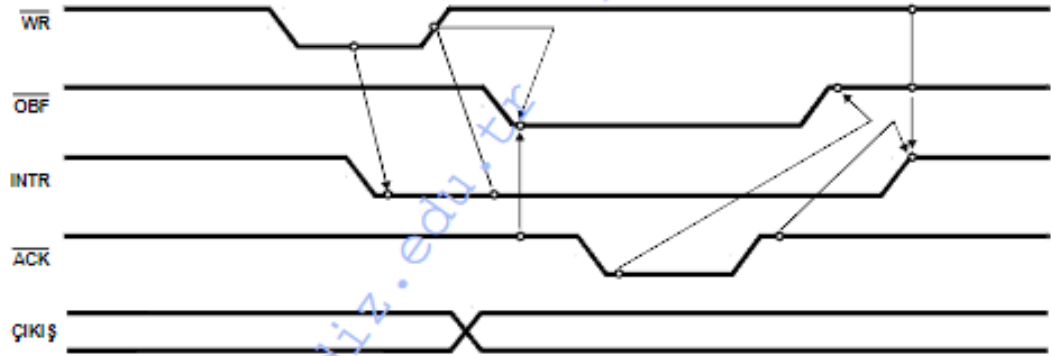
OBF (Output buffer full): 8255 açısından bir çıkış ucudur. OBF ucu, 8086'nın dışarı aktarılmak üzere 8255'e veri yazmasıyla yani WR ucunun yükselen kenarıyla lojik 1 seviyesine geçer. OBF ucu, ACK ucunun lojik 0 olmasıyla lojik 0 seviyesine geçer.

ACK (Acknowledge): 8255 açısından bir giriş ucudur. Bu uçta görülen lojik 0 seviyesi veri uçlarından gönderilen değerin dış birim tarafından kabul edilerek alındığı anlamına gelir.

INTR (Interrupt request): 8255 açısından bir çıkış ucudur. 8255'in yeni veri göndermeye hazır olduğu 8086'ya bir kesme isteği ile bildirilmek istenirse bunun için INTR ucu kullanılabilir. INTR ucu, ACK lojik 1, OBF lojik 1 ve INTE lojik 1 olduğunda lojik 1 seviyesine geçer. INTR ucu, WR ucunun düşen kenarıyla ise lojik 0 seviyesine geçer.

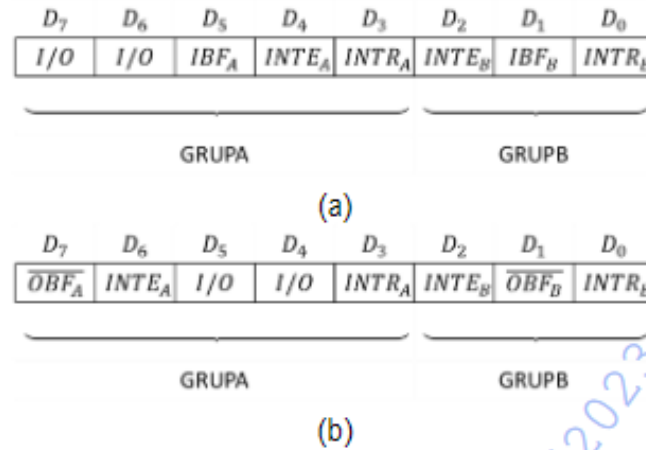
INTE (Interrupt enable): Kesme temelli veri gönderme yapabilmek için INTE bit'lerinin BSR kipinde lojik 1 seviyesinde ayarlanmaları gerekmektedir. INTEA PortC6, INTEB ise PortC2 ile kontrol edilir.

Çıkış yönlü Kip 1 için sinyalleşme yapısı Şekil 18-7 ile verilmektedir.



Şekil 18-7: Çıkış yönlü kip 1 için sinyalleşme yapısı

8255 kip 1 için ayarlanmışsa PORTC'den yapılan okumalarda durum yazmacına erişilmiş olur. Okunan değer ise durum kelimesi (status word) olarak isimlendirilir. 8255 kip 1 durum kelimesi ve bit bazındaki fonksiyonları Şekil 18-8 ile verilmiştir. Durum kelimesi ile OBF, IBF, INTR değerleri ile I/O için ayrılan PORTC uçları (giriş yönlü Kip 1'de PortC6, PortC7, çıkış yönlü Kip 1'de PortC4, PortC5) okunabilir. 8255 kullanılarak kip 1 ile veri gönderme veya alma işlemi kontrol temelli veri aktarımı (polling) ile gerçekleştirilecekse, durum yazmacı bir döngü içerisinde okunarak ilgili OBF ve IBF bit'leri test edilir. Bu işlem ilgili bit'ler beklenen değere ulaşana kadar tekrarlanır. Benzer bir kontrol INTR bit'leri için de yapılabilir.



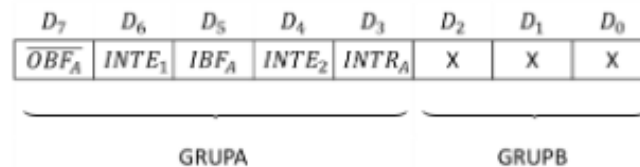
Şekil 18-8: 8255 grupA ve grupB kip 1, a) giriş, b) çıkış yönlü durum kelimesi

18.3.4 Kip 2

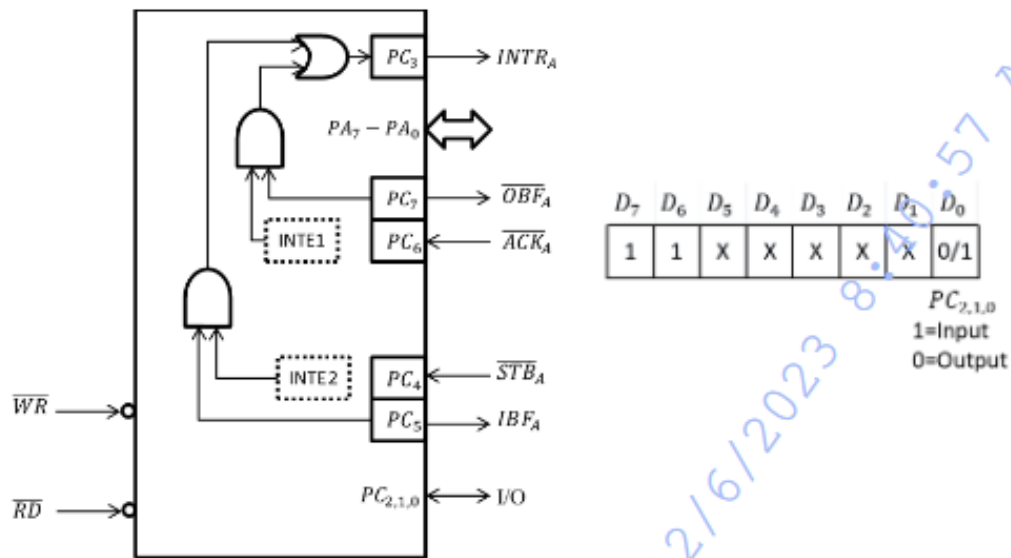
Çift yönlü tetikleme temelli paralel veri aktarımı için Kip 2 kullanılabilir. 8255 için sadece GrupA kip 2 ile veri aktarımını desteklemektedir. GrupA kip 2 ile kullanılırken veri PortA ile yazılıp okunur, PortC3, PortC4, PortC5, PortC6 ve PortC7 ise kontrol uçları olarak çalışmaktadır. Kip 2 için PortA yönü o an için yürütülen komuta göre belirlenmektedir. Kip 2 ile veri aktarımı için 8255 uç tanımları ve uygun kontrol kelimesi Şekil 18-10 ile verilmiştir.

$INTR_A$, \overline{OBF}_A , \overline{ACK}_A , \overline{STB}_A ve IBF_A uçları Kip 1'de verilen tanımları ile fonksiyon göstermektedir. Veri göndermeyi kesme temelli gerçekleştirebilmek için $INTE_1$, veri okumayı kesme temelli gerçekleştirebilmek için $INTE_2$ lojik 1 seviyesinde ayarlanmalıdır. $INTE_1$ için PortC6, $INTE_2$ için ise PortC4 BSR kipinde lojik 1 seviyesinde ayarlanmalıdır. 8255 kip 2 için tek bir fiziki kesme istek ucu mevcuttur. Dolayısıyla kesme temelli veri aktarımında; kesme servis programında durum yazmacı okunarak kesmenin kaynağı test edilmelidir. Kontrol temelli veri aktarımı yapılacaksa kontrol yazmacındaki \overline{OBF}_A (8255'in kip 2'de veri gönderme için hazır olduğunu belirtir) ve IBF_A (8255'in kip 2'de veri okuma için hazır olduğunu belirtir) bit'leri bir döngü içerisinde beklenen değere ulaşana kadar sürekli kontrol edilebilir.

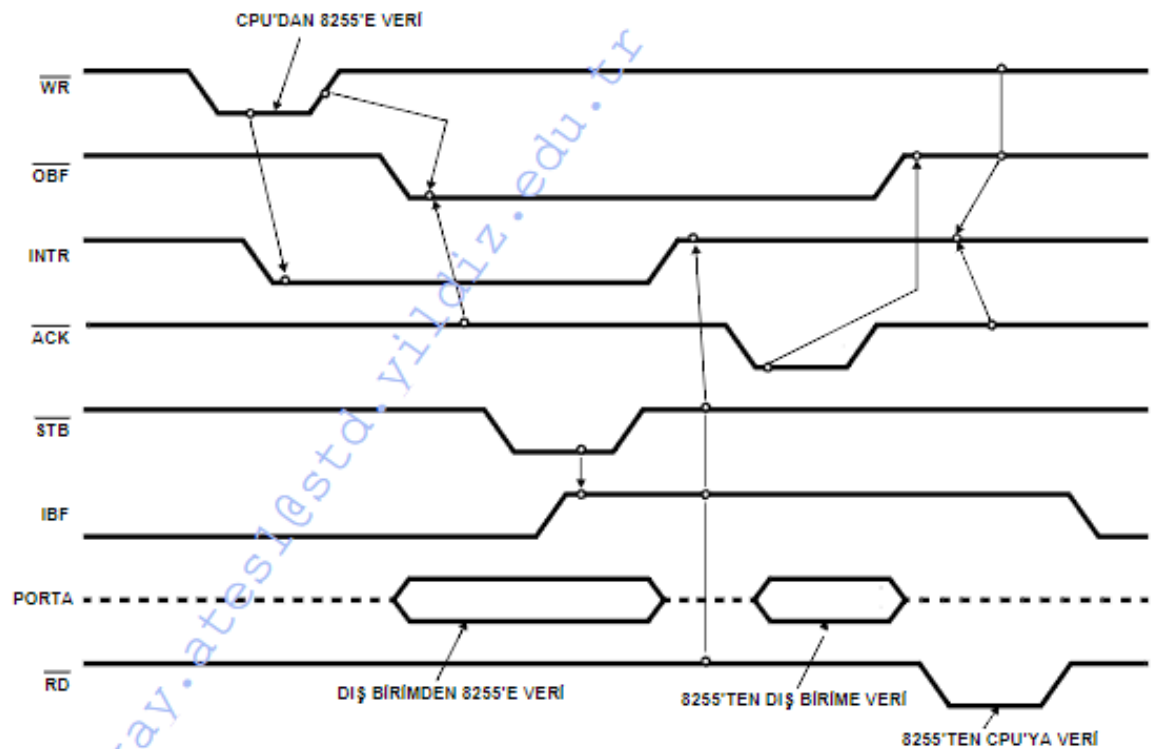
8255 Kip 2 durum yazmacı içeriği ve ilgili bit'lerin fonksiyonları Şekil 18-9 ile verilmiştir. 8255 Kip 2 için sinyalleşme yapısı Şekil 18-11 ile verilmektedir.



Şekil 18-9: 8255 kip 2 durum kelimesi

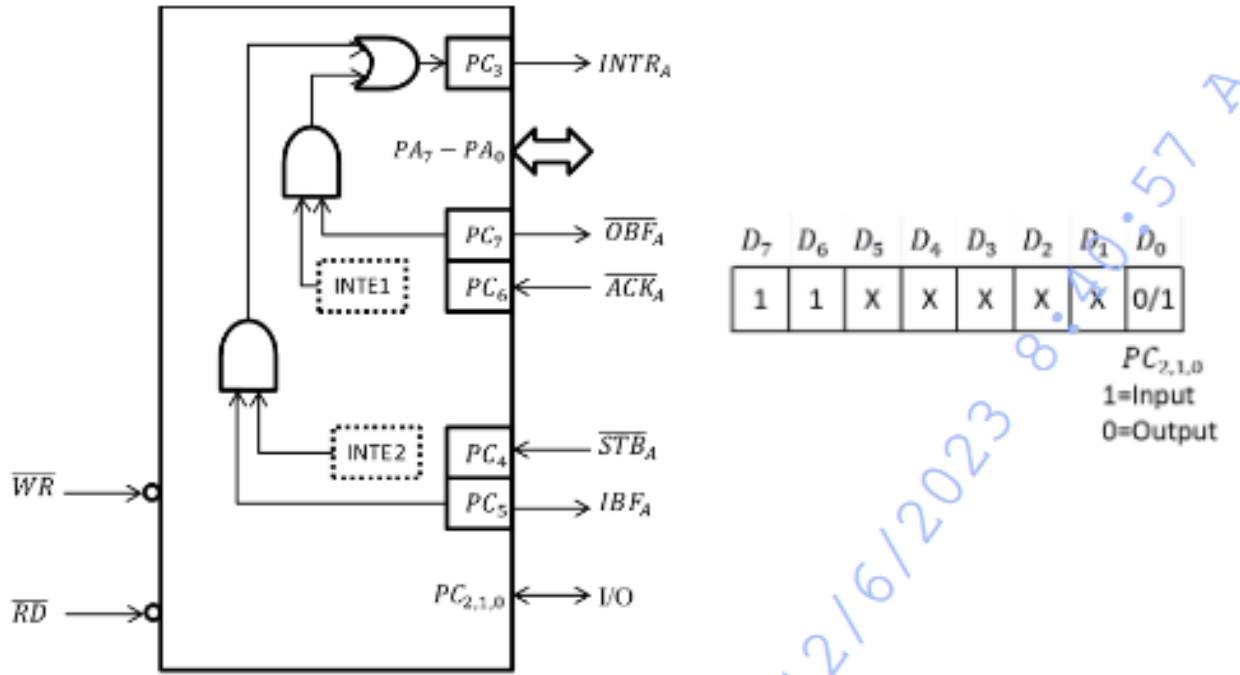


Şekil 18-10: 8255 kip 2 uç tanımları ve kontrol kelimesi

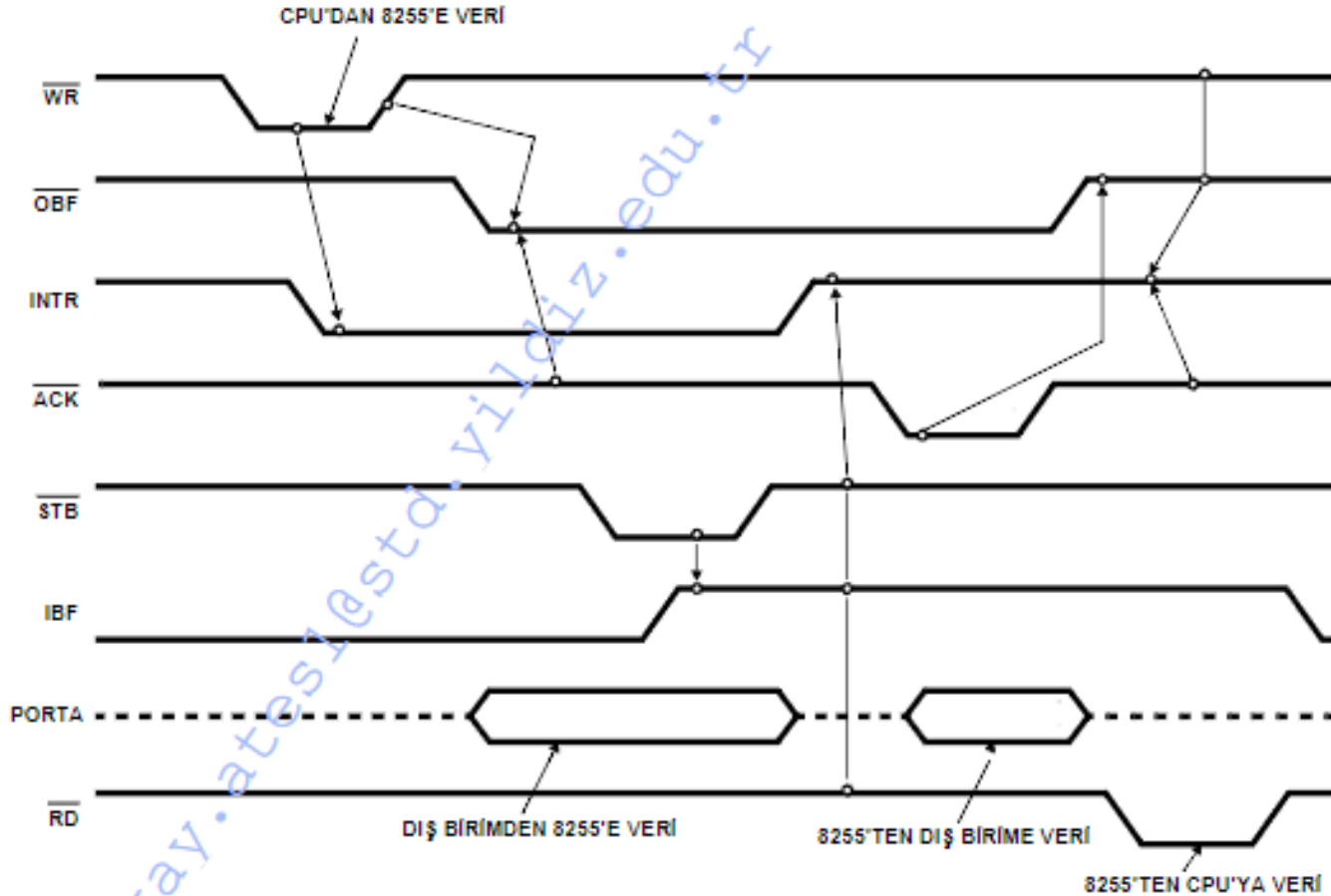


Şekil 18-11: 8255 kip 2 için sinyalleşme yapısı

18. 8255 PPI (Programmable Peripheral Interface)



Şekil 18-10: 8255 kip 2 uç tanımları ve kontrol kelimesi



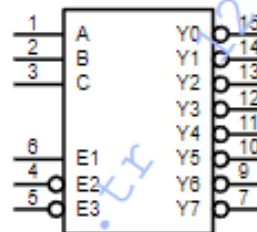
Şekil 18-11: 8255 kip 2 için sinyalleşme yapısı

18.4 8255 Örnek Uygulamaları

8255 kullanımıyla ilgili olarak adres çözümleme, kip 0 çıkış, kip 0 giriş, kip 1 ve kip 2 uygulamalarına ilerleyen bölümlerde yer verilmiştir.

18.4.1 Kodçözücü ile Adres Çözümleme

8 bit'lik G/Ç adres değerleri ile işlem yapılacak ise bunun için G/Ç assembly komutları sabit adres ile kullanılabilir. Bu durumda G/Ç biriminin adres aralığı belirleyen 8 adet adres ucundan (A7-A0) gerekli olanlar ve çeşitli kontrol uçları (\overline{M}/IO , \overline{WR} , \overline{RD} , BHE) adres çözümleme için 3x8 kodçözücü ile birlikte kullanılabilir. Bellek ve çevre birimlerinin seçim uçları aktif 0 mantıkta olduğundan aktif 0 çıkışlı 74138 tümdevresi adres çözümlemede kullanılabilir. Şekil 18-12 ve Tablo 18-4 ile sırasıyla, 74138 tümdevresinin fonksiyonel uç tanımları ve fonksiyon tablosuna yer verilmiştir.



Şekil 18-12: 3x8 kodçözücü uç tanımları

74138 üç adet yetkilendirme ucuna (E1, E2, E3) sahiptir. Yetkilendirme uçlarından bir tanesi (E1) aktif 1, iki tanesi (E2, E3) ise aktif 0 mantıkta çalışmaktadır. Üç adet seçim ucu bulunmaktadır. C ucu en anlamlı seçim ucudur. 74138'in sekiz adet aktif 0 mantıkta çalışan çıkışı mevcuttur.

Tablo 18-4: 74138 3x8 kodçözücü için fonksiyon tablosu

GİRİŞLER						ÇIKIŞLAR								SEÇİLEN ÇIKIŞ
YETKİ			SEÇİM											
E1	E2	E3	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7	
0	X	X	X	X	X	1	1	1	1	1	1	1	1	-
X		X	X	X	X	1	1	1	1	1	1	1	1	-
X	X	1	X	X	X	1	1	1	1	1	1	1	1	-
1	0	0	0	0	0	0	1	1	1	1	1	1	1	Y0
1	0	0	0	0	1	1	0	1	1	1	1	1	1	Y1
1	0	0	0	1	0	1	1	0	1	1	1	1	1	Y2
1	0	0	0	1	1	1	1	1	0	1	1	1	1	Y3
1	0	0	1	0	0	1	1	1	1	0	1	1	1	Y4
1	0	0	1	0	1	1	1	1	1	1	0	1	1	Y5
1	0	0	1	1	0	1	1	1	1	1	1	0	1	Y6
1	0	0	1	1	1	1	1	1	1	1	1	1	0	Y7

Örnek 18-2

8086 G/Ç uzayında 60H adresinden itibaren ardışık çift adreslere bir adet 8255 yerleştirilmek isteniyor. Bağlantı Kapılı G/Ç haritalama yöntemi ile kullanarak.

- Adres çözümleme devresini 3×8 kodçözücü ve gerekli basit lojik kapılar kullanarak tasarlayınız.
- 8255 ile mikroişlemci arasındaki bağlantıları çizerek gösteriniz.

60H adresinden itibaren ardışık çift adresler ele alındığında dört farklı adres değerinin (PortA, PortB, PortC, Kontrol yazmaç adresleme için) A2 ve A1 uçlarındaki değişimden kaynaklandığı görülmektedir (Tablo 18-5). Dolayısıyla A2 ve A1 adres uçlarının 8255 adres uçlarına bağlı olması gerekmektedir.

Tablo 18-5: Örnek 18-2 ile verilen 8255'in adres aralığı

Adres	A7	A6	A5	A4	A3	A2	A1	A0
60H	0	1	1	0	0	0	0	0
62H	0	1	1	0	0	0	1	0
64H	0	1	1	0	0	1	0	0
66H	0	1	1	0	0	1	1	0

G/Ç birimi ile ilgili işlem yapılacağı ve adres aralığı dikkate alınacak olursa adres çözümleme devresi için gerekli uçlar ve bunların alması gereken değerler Tablo 18-6 ile verilmiştir.

Tablo 18-6: Örnek 18-2 ile verilen 8255 adres çözümlemesi için sabit değerler

M/\bar{O}	A7	A6	A5	A4	A3	A0
0	0	1	1	0	0	0

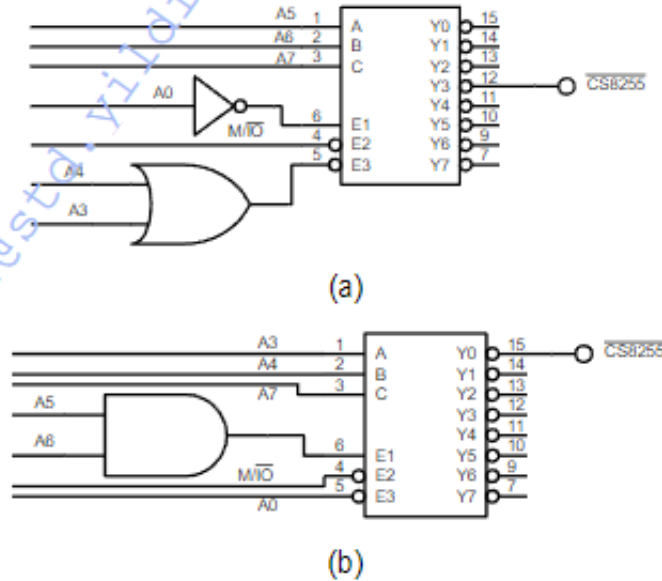
Adres çözümleme devresi Tablo 18-6 ile verilen yedi adet uça belirlenen değerlerin aynı anda bulunması durumunda seçim yapmalıdır. Tablo 18-6 ile belirlenen değerlerden biri dahi farklı olsa seçim yapmamalıdır. Örnek 18-2 için elde edilebilecek pek çok adres çözümleme seçeneklerinden iki tanesi Şekil 18-13 ile verilmektedir.

Yedi farklı uç için değerler belirlenmişken, 3×8 kodçözücü için altı giriş ucu bulunmaktadır. Bu durumda aynı lojik seviyeye sahip olacak uçların uygun şekilde birleştirilmesi gerekecektir.

Şekil 18-13 (a) ile verilen adres çözümleme devresinde, verilen adres aralığı için lojik 0 değerleri alması gereken A4 ve A3 uçları OR kapısı ile birleştirilerek aktif 0

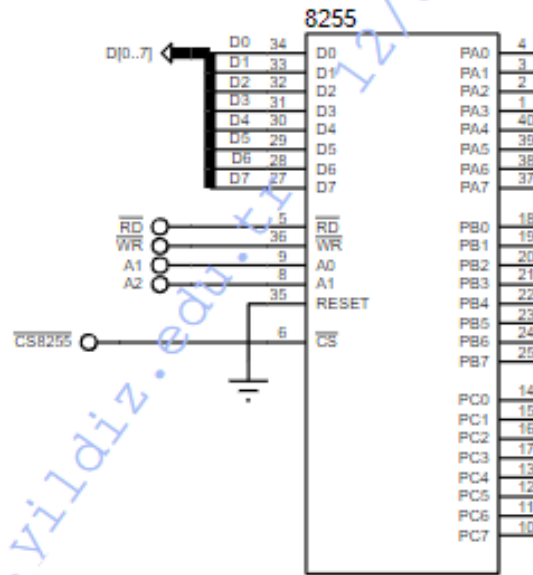
yetki uçlarından birine $\bar{E}3$ bağlanmıştır. Burada $A4 \neq 0$ veya $A3 \neq 0$ durumu oluşturan herhangi bir başka adres için $\bar{E}3$ ucunda lojik 1 değeri oluşacak ve kodçözücü çıkış üretmeyecektir. Bağlantı kapılı G/Ç haritalama yönteminde \bar{M}/\bar{IO} ucu adres çözümlemeye kullanılır. G/Ç birimi adresleneceği için uçta lojik 0 değeri ile seçim yapılmalıdır. Bunun için \bar{M}/\bar{IO} ucu doğrudan aktif 0 bir yetki ucuna ($\bar{E}2$) bağlanmıştır. A0 ucu verilen adres aralığı için lojik 0 değerini alır, bundan dolayı A0 ucu NOT kapısından geçirilerek aktif 1 yetki ucuna ($E1$) bağlanmıştır. Geri kalan adres uçları ($A7, A6, A5$) anlam sırasına göre kodçözücü seçim uçlarına bağlanmıştır. Kodçözücü seçim uçlarında anlam sırası büyükten küçüğe doğru C, B, A sırasındadır. Dolayısıyla verilen adres aralığı için kodçözücü tarafından yapılan $(A_7A_6A_5)_2 = (011)_2 = 3$ çözümlemesiyle $\bar{Y}3 = 0$ olacak şekilde çıkış üretilir (bkz. Tablo 18-4).

Şekil 18-13 (b) ile verilen adres çözümleme devresinde, verilen adres aralığı için lojik 1 değerleri alması gereken $A6$ ve $A5$ uçları AND kapısı ile birleştirilerek aktif 1 yetki ucuna ($E1$) bağlanmıştır. Burada $A6 \neq 1$ veya $A5 \neq 1$ durumu oluşturan herhangi bir başka adres için $E1$ ucunda lojik 0 değeri oluşacak ve kodçözücü çıkış üretmeyecektir. Bağlantı kapılı G/Ç haritalama yönteminde \bar{M}/\bar{IO} ucu adres çözümlemeye kullanılır. G/Ç birimi adresleneceği için uçta lojik 0 değeri ile seçim yapılmalıdır. Bunun için \bar{M}/\bar{IO} ucu doğrudan aktif 0 bir yetki ucuna ($E2$) bağlanmıştır. A0 ucu verilen adres aralığı için lojik 0 değerini alır, bundan dolayı A0 ucu başka bir aktif 0 yetki ucuna ($E3$) bağlanmıştır. Geri kalan adres uçları ($A7, A4, A3$) anlam sırasına göre kodçözücü seçim uçlarına bağlanmıştır. Verilen adres aralığı için kodçözücü tarafından yapılan $(A_7A_4A_3)_2 = (000)_2 = 0$ çözümlemesiyle $\bar{Y}0 = 0$ olacak şekilde çıkış üretilir (bkz. Tablo 18-4).



Şekil 18-13: Örnek 18-2 için 3x8 kodçözücü kullanarak tasarlanan adres çözümleme devresi, a) seçenek 1, b) seçenek 2

8255 ile mikroişlemci arasında yapılması gereken bağlantılar verilen adres değerlerine göre Şekil 18-14 ile gösterilmiştir. Verilen adres değerleri ardışık çift adresler olduğundan dolayı 8255 veri yolu mikroişlemci veri yolunun düşük anlamlı byte'ı ile uygun anlam sırasında bağlanmıştır. 8255'e ilişkin okuma ve yazma işlemlerini yerine getirebilmek için mikroişlemcinin \overline{WR} , \overline{RD} uçları 8255 \overline{WR} , \overline{RD} uçlarına bağlanmıştır. Adres çözümleme devresinden elde edilen seçim ucu 8255 seçim ucuna bağlanmıştır. Verilen adres değerlerinde dört farklı adres A2 ve A1 adres uçlarındaki değişimler ile sağlanmaktadır. Mikroişlemcinin A2 ve A1 uçları sırasıyla 8255 A1 ve A0 adres uçlarına bağlanmıştır. Buna göre PortA 60H, PortB 62H, PortC 64H ve Kontrol Yamacı 66H adresine yerleştirilmiş olur. Eğer mikroişlemci A2, A1 adres uçları 8255'te sırasıyla A0, A1 adres uçlarına bağlanmış olsaydı PortB 64H, PortC 62H adresine yerleşmiş olurdu, bunun dışında PortA ve kontrol yazmaç adresleri değişmezdi.



Şekil 18-14: Örnek 18-2 için 8255 ile 8086 arasında yapılması gereken bağlantılar

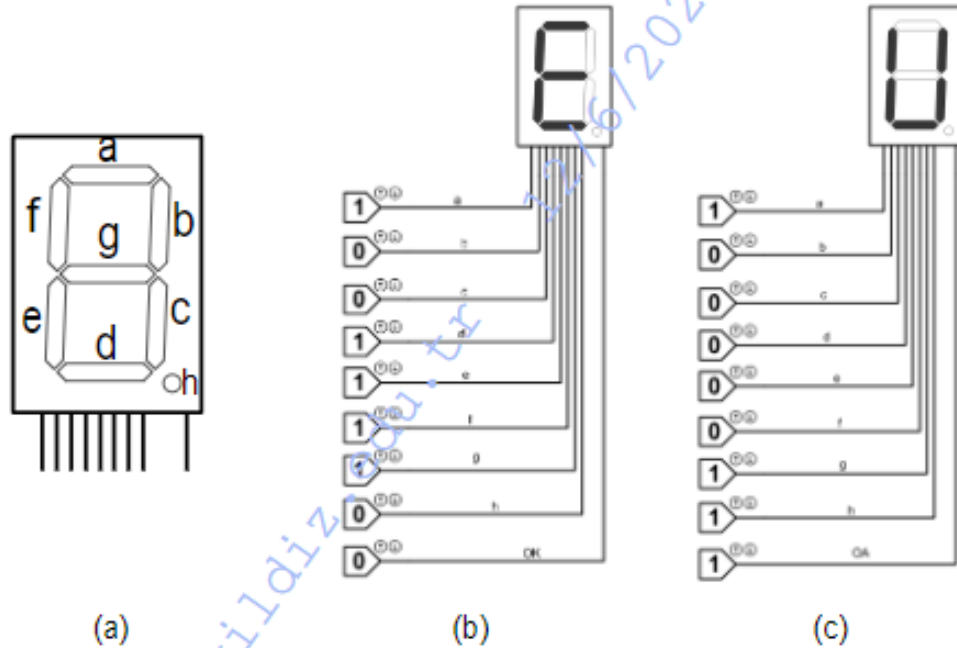
18.4.2 Yedi Parçalı Gösterge

Yedi (veya ondalık nokta ile sekiz) parçalı gösterge LED'lerin uygun şekilde yerleştirilmesi ile ondalık, onaltılık basamaklar veya çeşitli alfasayısal değerler oluşturabilen göstergelerdir. Yedi parçalı göstergeler tek başlarına bir basamak gösterecek şekilde veya birden fazla kullanılarak çok basamaklı bir sayı gösterecek şekilde kullanılabilir.

Yedi parçalı göstergelerde sekiz tane veri ucu ve bir adet de ortak uç bulunur. Ortak uç kullanılan LED'lerin katot uçlarını birleştiriyorsa bu ortak uç ortak katot ucu ola-

arak isimlendirilir ve gösterge **ortak katotlu yedi parçalı gösterge** olarak isimlendirilir. Benzer şekilde ortak uç kullanılan LED'lerin anot uçlarını birleştiriyorsa bu ortak uç ortak anot ucu olarak isimlendirilir ve gösterge **ortak anotlu yedi parçalı gösterge** olarak isimlendirilir.

Ortak katotlu yedi parçalı göstergelerde değer oluşturabilmek için ortak katot ucuna mutlaka lojik 0 uygulamak gereklidir. Ortak anotlu yedi parçalı göstergelerde değer oluşturmak için ise ortak anot ucuna mutlaka lojik 1 değer uygulamak gerekir. Bu halî ile göstergeleri açıp kapatmak için ortak uçların kullanılabileceği söylenebilir.



Şekil 18-15: Yedi parçalı gösterge için, a) segment'lerin isimlendirilmesi, b) ortak katotlu, c) ortak anotlu yedi parçalı gösterge kullanımı

Şekil 18-15 (a) ile yedi parçalı göstergede parça (segment) isimlendirmelerine yer verilmiştir. Şekil 18-15 (b) ile ortak katotlu yedi parçalı göstergede E yakılabilmesi için uçlarda oluşturulması gereken değerlere yer verilmiştir. Ortak katotlu yedi parçalı göstergede değer oluşturabilmek için ortak ucun mutlaka lojik 0'a bağlanması gerekmektedir. Ortak katot durumunda herhangi bir parçanın yakılabilmesi için parça kontrol ucuna lojik 1 verilmelidir. Buna göre a, d, e, f ve g parçalarını yakmak için ilgili kontrol uçlarına lojik 1, b, c ve h parçalarını söndürmek için ilgili kontrol uçlarına lojik 0 uygulanmıştır. Şekil 18-15 (c) ile ortak anotlu yedi parçalı göstergede U yakılabilmesi için uçlarda oluşturulması gereken değerlere yer verilmiştir. Ortak anotlu yedi parçalı göstergede değer oluşturabilmek için ortak ucun mutlaka lojik

1'e bağlanması gerekmektedir. Ortak anot durumunda herhangi bir parçanın yakılabilmesi için parça kontrol ucuna lojik 0 verilmelidir. Buna göre b, c, d, e ve f parçalarını yakmak için ilgili kontrol uçlarına lojik 0, a, g ve h parçalarını söndürmek için ilgili kontrol uçlarına lojik 1 uygulanmıştır.

Örnek 18-3

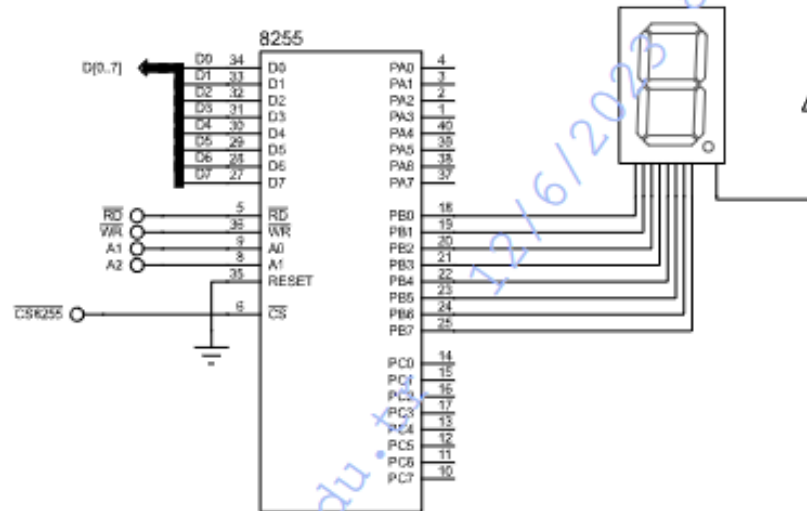
Örnek 18-2 ile verilen adres aralıklarına yerleştirilmiş ve gerekli mikroişlemci bağlantıları yapılmış 8255 kullanılarak, PortB yardımıyla ortak anotlu yedi parçalı göstergede sırasıyla 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 şeklinde ondalık basamakları oluşturan assembly kodunu yazınız (Şekil 18-16).

Ortak anotlu yedi parçalı göstergede ondalık basamakları yakabilmek için gerekli parça lojik değerleri Tablo 18-7 (tablonun ilk 10 satırı) ile verilmiştir.

Tablo 18-7: Ortak anotlu yedi parçalı göstergede onaltılık basamak değerlerine karşılık aktif edilecek parçalar

Basamak Değeri	Parça Değerleri								Onaltılık Değeri
	h	g	f	e	d	c	b	a	
0	1	1	0	0	0	0	0	0	C0H
1	1	1	1	1	1	0	0	1	F9H
2	1	0	1	0	0	1	0	0	A4H
3	1	0	1	1	0	0	0	0	B0H
4	1	0	0	1	1	0	0	1	99H
5	1	0	0	1	0	0	1	0	92H
6	1	0	0	0	0	0	1	0	82H
7	1	1	1	1	1	0	0	0	F8H
8	1	0	0	0	0	0	0	0	80H
9	1	0	0	1	0	0	0	0	90H
A	1	0	0	0	1	0	0	0	88H
B	1	1	0	0	0	0	1	1	B3H
C	1	0	0	0	0	1	1	0	86H
D	1	1	1	0	0	0	0	1	E1H
E	1	0	0	0	0	1	1	0	86H
F	1	0	0	0	1	1	1	0	8EH

Ortak anotlu yedi parçalı gösterge için veri uçları (hgfdedba) sırasıyla PortB'de (B₇B₆B₅B₄B₃B₂B₁B₀) uçlarına sırasıyla bağlanmıştır. Yedi parçalı göstergenin ortak ucu lojik 1'e bağlanmıştır. Dolayısıyla veri uçlarından lojik 0 uygulandığında ilgili parça yanacaktır. Gerçeklenecek sistemde sadece PortB ile yedi parçalı gösterge veri uçlarında değer oluşturabilmek istendiği için 8255'in Kip 0'da, PortB'nin de çıkış yönlü olarak ayarlanması gerekmektedir. 8255 ile mikroişlemci arasındaki bağlantılar Örnek 18-2'te verildiği şekilde yapılmıştır. Örneğe ilişkin bağlantı yapısı Şekil 18-16 ile verilmektedir.



Şekil 18-16: 8255 ile bir tane ortak anotlu yedi parçalı gösterge sürmek için yapılan bağlantılar

```

;-----
; PROGRAM : Örnek 18-3.ASM
; 8255 kullanılarak, PortB yardımıyla ortak anotlu yedi parçalı göstergede
; sırasıyla 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 şeklinde ondalık başmakları
; oluşturan assembly program
;-----
PAGE 60,80
TITLE 8255 PortB ortak anotlu yedi parçalı göstergede uygulaması
STAK SEGMENT PARA STACK 'STACK'
DW 20 DUP(?)
STAK ENDS
DATA SEGMENT PARA 'DATA'
DIGITS DB 0C0H,0F9H,0A4H,0B0H,99H,92H,82H,0F8H,80H,90H,88H
DATA ENDS
CODE SEGMENT PARA 'CODE'
ASSUME CS:CODE, DS:DATA, SS:STAK
START PROC FAR
;-----
; DATASG ismiyle tanımlı kesim alanına erişebilmek için gerekli tanımlar
;-----
MOV AX, DATA
MOV DS, AX
;-----
; 8255 kip 0, tüm portlar çıkış yönlü ayarlanır
;-----
MOV AL, 80H

```

↳ Control word atandı

```

OUT 66H, AL  → 66H'teki Kontrol worde yollandı,
CALL CLEAR_DISP

ENDLESS:
XOR BX, BX   ; DIGITS dizisinin ilk elemanından itibaren
MOV CX, 10   ; tüm basamaklar için

TEKRAR:
MOV AL, DIGITS[BX]
OUT 62H, AL  ; PortB'ye basamak gösterge değeri gönderilir
INC BX       ; sonraki basamağa geçilir
CALL DELAY   ; yakılan değerin görülmesi için bekle
CALL CLEAR_DISP
LOOP TEKRAR
JMP ENDLESS  ; sonsuz döngü

START
ENDP

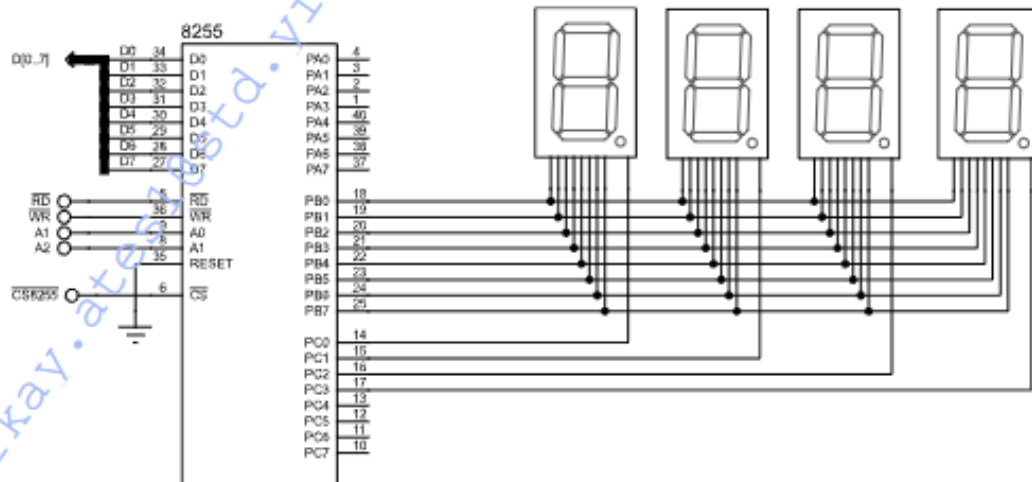
;-----
; PortB'ye bağlı göstergeyi temizleyen altıyordam
;-----
CLEAR_DISP PROC NEAR
PUSH AX
MOV AL, 0FFH
OUT 62H, AL
POP AX
RET
CLEAR_DISP ENDP

;-----
; Bekleme sağlayan altıyordam
;-----
DELAY PROC NEAR
PUSH CX
MOV CX, 0FFFFH
LOOP_DELAY: LOOP LOOP_DELAY
POP CX
RET
DELAY ENDP
CODE
ENDS

END START

```

Örnek 18-4



Şekil 18-17: 8255 ile dört adet ortak anotlu yedi parçalı gösterge sürmek için yapılan bağlantılar

Örnek 18-2 ile verilen adres aralıklarına yerleştirilmiş ve gerekli mikroişlemci bağlantıları yapılmış 8255 kullanılarak, PortB ve PortC yardımıyla dört adet ortak anotlu yedi parçalı gösterge sürülmek isteniyor. Tüm göstergelerin veri uçları (hgfedcba) sırasıyla PortB'de (B₇B₆B₅B₄B₃B₂B₁B₀) sırasıyla bağlanmıştır. PortC uçları (C₃C₂C₁C₀) ise sırasıyla göstergelerin ortak anot uçlarına (OA₃OA₂OA₁OA₀) bağlıdır. Verilenler ile dört basamaklı 1234 sayısını göstergede yakabilmek için gerekli assembly kodunu yazınız.

```

;-----
; PROGRAM : Örnek 18-4.ASM
; 8255 kullanarak PortB ve PortC yardımıyla dört adet ortak anotlu yedi
; parçalı göstergede 1234 sayısını süren assembly program
;-----

                PAGE 60,80
                TITLE dört adet OA 7 parçalı göstergede 1234 sürme
STAK            SEGMENT PARA STACK 'STACK'
                DW 20 DUP(?)
STAK            ENDS
DATA            SEGMENT PARA 'DATA'
DIGITS          DB 0C0H,0F9H,0A4H,0B0H,99H,92H,82H,0F8H,80H,90H
DISPLAY         DW 1234H
DATA            ENDS
CODE            SEGMENT PARA 'CODE'
                ASSUME CS:CODE, DS:DATA, SS:STAK
START           PROC FAR
;-----
; DATA ismiyle tanımlı kesim alanına erişebilmek için gerekli tanımlar
;-----
                MOV AX, DATA
                MOV DS, AX
;-----
; 8255 kip 0, PortA, PortB, PortC çıkış
;-----
                MOV AL, 80H
                OUT 66H, AL
                CALL CLEAR_DISP
ENDLESS:        MOV AL, 08H ; OA3'ü için
                MOV DI, DISPLAY
                MOV CX, 4    ; 4 basamak için döngü
DIGIT_COUNT:    OUT 64H, AL ; sıradaki OA ucu aktif
                MOV SI, DI
                AND SI, 0FH ; en düşük anlamlı 4 bit'ini maskele
                PUSH CX
                MOV CX, 4
                SHR DI, CL   ; sıradaki basamak en düşük anlamlı 4 bit'e
                POP CX
                CALL DISPLAY_DIGIT
                CALL CLEAR_DISP
                SHR AL, 1    ; sıradaki OA seçimi
                LOOP DIGIT_COUNT
                JMP ENDLESS ; sonsuz döngü
START           ENDP
;-----
; PortB'den gösterge değeri sağlayan altıyordam
;-----
DISPLAY_DIGIT  PROC NEAR
                PUSH AX
                MOV AL, DIGITS[SI]
                OUT 62H, AL

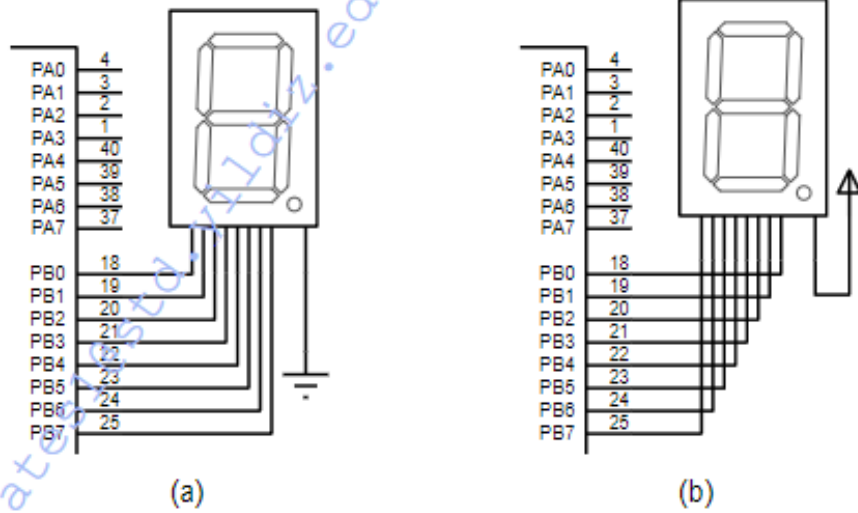
```

```

CALL DELAY
POP AX
DISPLAY_DIGIT ENDP
; Bekleme sağlayan altıyordam
;
DELAY PROC NEAR
PUSH CX
MOV CX, 0FH
LOOP_DELAY: LOOP LOOP_DELAY
POP CX
RET
DELAY ENDP
; PortB'ye bağlı göstergeleri temizleyen altıyordam
;
CLEAR_DISP PROC NEAR
PUSH AX
MOV AL, 0FFH
OUT 62H, AL
POP AX
RET
CLEAR_DISP ENDP
CODE
ENDS
END START

```

Örnek 18-5



Şekil 18-18: (a) ortak katotlu yedi parçalı gösterge bağlantısı, (b) ortak anotlu yedi parçalı göstergede veri uçlarının anlam sırasının değiştirilmesi

- a) Örnek 18-3 ile verilen göstergede ortak anotlu yedi parçalı gösterge yerine ortak katotlu yedi parçalı gösterge kullanılıyorsa istenenleri yerine getirebilmek için yapılması gereken değişiklikler nelerdir?

- b) Örnek 18-3 ile verilen göstergede ortak anotlu yedi parçalı göstergenin veri uçları (hgfedcba) sırasıyla PortB'de ($B_7B_6B_5B_4B_3B_2B_1B_0$) sırası yerine ($B_0B_1B_2B_3B_4B_5B_6B_7$) sırasıyla bağlansaydı istenenleri yerine getirebilmek için yapılması gereken değişiklikler nelerdir?

Ortak anotlu yedi parçalı gösterge yerine ortak katotlu yedi parçalı gösterge kullanılmış olsaydı, biri donanımda biri de yazılımda olmak üzere iki değişiklik yeterli olacaktır. Ortak katotlu yedi parçalı gösterge kullanıldığında donanımsal olarak gösterge ortak ucunun lojik 0'a bağlanması gerekir (Şekil 18-14 (a)). İstenen basamak değerlerinin ortak katotlu yedi parçalı göstergede yakılabilmesi için gösterge veri uçlarına ortak anotlu durumun lojik olarak değerlerinin uygulanması gerekir. Bu ise yazılımsal olarak veri kesiminde tanımlanmış olan ortak anotlu durum için tanımlanmış gösterge değerlerinin değilinin alınması ile sağlanabilir.

```
MOV AL, DIGITS[BX]
```

```
NOT AL
```

```
OUT 62H, AL ; PortB'ye basamak gösterge değeri gönderilir
```

Gösterge veri uçlarında (hgfedcba) sırası PortB uçlarında ($B_0B_1B_2B_3B_4B_5B_6B_7$) sırasıyla bağlanırsa (Şekil 18-14 (b)), veri kesiminde tanımlanmış ortak anotlu durum için tanımlanmış olan gösterge değerlerinin bit anlam sırasının değiştirilmesi ile istenenlerin yerine getirilmesi sağlanabilir.

```
MOV AL, DIGITS[BX]
```

```
CALL REVERSE_BYTE
```

```
OUT 62H, AL ; PortB'ye basamak gösterge değeri gönderilir
```

```
...
```

```
; byte için bit anlam sırasını tersine çeviren alt yordam
```

```
REVERSE_BYTE PROC NEAR
```

```
PUSH CX
```

```
PUSH BX
```

```
MOV CL, 8
```

```
REV_LOOP: RCR AL, 1
```

```
RCL BL, 1
```

```
LOOP REV_LOOP
```

```
MOV AL, BL
```

```
POP BX
```

```
POP CX
```

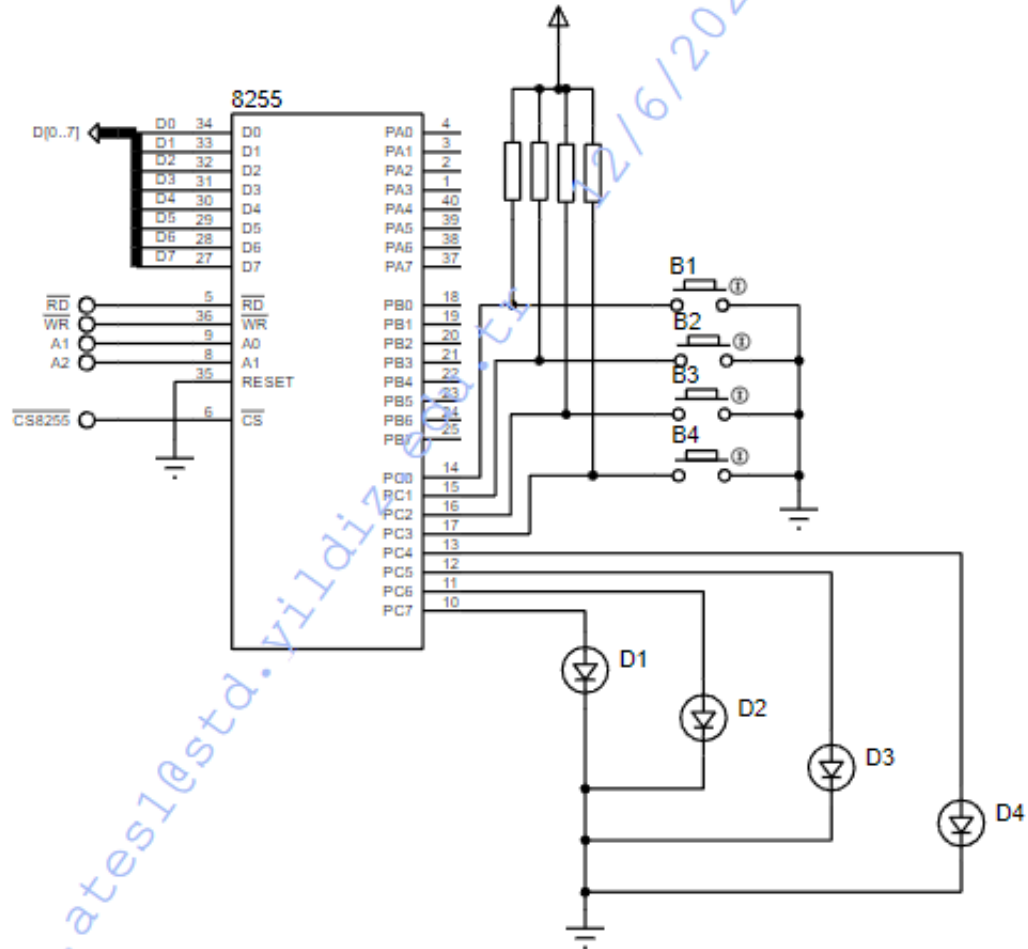
```
RET
```

```
REVERSE_BYTE ENDP
```

18.4.3 Buton ve LED Uygulaması

Örnek 18-6

Örnek 18-2 ile verilen adres aralıklarına yerleştirilmiş ve gerekli mikroişlemci bağlantıları yapılmış 8255 kullanılarak, PortC'nin düşük anlamlı dört ucuna birer buton kaldırma dirençleri (pull-up resistor) kullanılarak bağlanmıştır. Benzer şekilde PortC'nin yüksek anlamlı dört ucuna, dört adet LED'in anot uçları bağlanmıştır. Varsayılan olarak tüm LED'ler yanarken herhangi bir butona basıldığında aynı sıradaki LED'i söndüren assembly kodunu yazınız.



Şekil 18-19: Örnek 18-6 donanımsal bağlantılar

```

;-----
; PROGRAM : Örnek 18-6.ASM
; 8255 kullanarak basılan butona göre LED söndüren
; assembly program
;-----

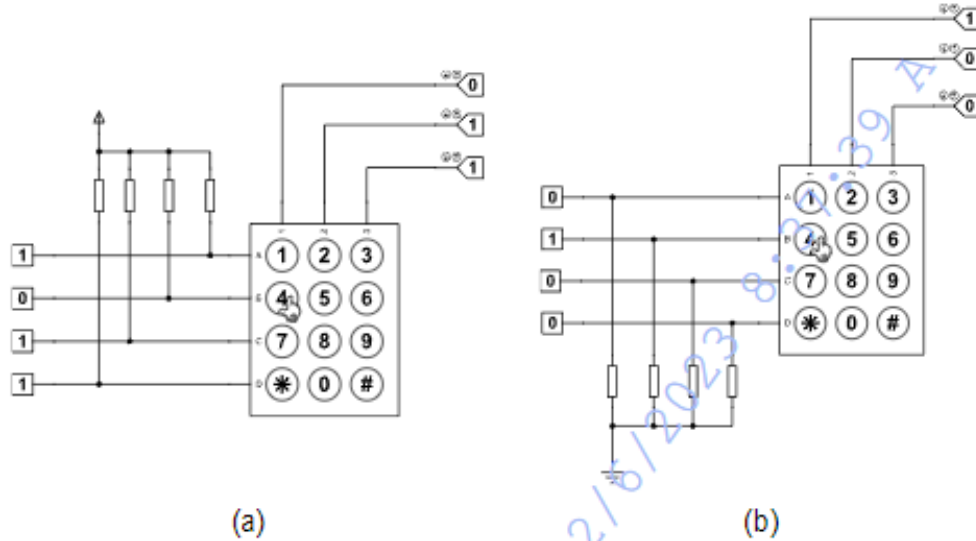
                PAGE 60,80
                TITLE buton ve LED uygulaması
STAK            SEGMENT PARA STACK 'STACK'
                DW 20 DUP(?)
STAK            ENDS
CODE            SEGMENT PARA 'CODE'
                ASSUME CS:CODE, SS:STAK
START           PROC FAR
                ; ana yordam
                MOV AL, 81H      ; PCL giriş, PCH çıkış
                OUT 66H, AL      ; olarak ayarlanır
                MOV AL, 0FFH     ; LED'ler varsayılan
                OUT 64H, AL      ; olarak yanıyor
ENDLESS:        IN AL, 64H      ; PC okunur
                OR AL, 0F0H      ; PCH maskelenerek 1 yapılır
                CMP AL, 0FFH     ; butonlar basılı değilken lojik 1 verir
                JZ ENDLESS      ; hiçbir butona basılmamış
                MOV CL, 4        ; basılan buton karşılığı
                SHL AL, CL       ; 4 birim sola öteleme ile PCH'ye taşınır
                OUT 64H, AL      ; LED'lerde basılı buton karşılığı söndürülür
                JMP ENDLESS      ; sonsuz döngü
START           ENDP
CODE            ENDS
                END START

```

18.4.4 Tuş Tarama Takımı

Buton kullanımında bir port ucu bir butona ayrılmaktadır. Bu durumda sayısal bir tuş takımı uygulamasında, örneğin 0-9 arası basamaklar ve iki özel karakter için toplam 12 uç gerekmektedir. Seçenek olarak daha az uç gerektiren fakat karmaşıklığı buton taramaya göre daha fazla olan tuş tarama takımı kullanılabilir. Tuş takımında basılan butonun bağlı olduğu sütun ve satır uçları kısa devre yapılır. Tuş tarama takımında sekiz adet uçla (4 tanesi giriş, 4 tanesi çıkış yönlü ayarlanmış olmak kaydıyla) toplam 16 adet buton sürülebilir. Şekil 18-20 ile tuş tarama takımı için iki farklı donanımsal gerçeklemeye yer verilmiştir²⁰⁸. Şekil 18-20 (a) ile verilen bağlantılarda butonlar basılı değilken satırlardan lojik 1 okunur, dolayısıyla herhangi bir sütun sürülmek isteniyorsa lojik 0 ile sürülmesi gerekir. Sürülen sütunda bir butona basılmışsa ilgili satırdan lojik 0 okunur. Şekil 18-20 (b) ile verilen bağlantılarda butonlar basılı değilken satırlardan lojik 0 okunur, dolayısıyla herhangi bir sütun sürülmek isteniyorsa lojik 1 ile sürülmesi gerekir. Sürülen sütunda bir butona basılmışsa ilgili satırdan lojik 1 okunur.

²⁰⁸ Aynı anda iki butona basılmayacağı varsayılmıştır. Aynı anda birden fazla butona basılma durumu dikkate alınmak isteniyorsa uygun diyotlar ile kaynak ve 8255 uçlarının korunması gerekmektedir.



Şekil 18-20: Tuş tarama takımı (a) kaldırma dirençleriyle, (b) indirme dirençleriyle

Tuş tarama takımının sürülmesinde, sütunlar çıkış yönlü olarak ayarlanmış, satırlar ise giriş yönlü olarak ayarlanmış 8255 uçlarına bağlanır. Her seferinde sadece bir sütun aktif edilerek²⁰⁹, sırasıyla tüm sütunlar sürülür, son sütuna gelindiğinde tekrar ilk sütuna dönülerek işlem tekrarlanır. Herhangi bir butona basılmamışsa satırlardan varsayılan değerler²¹⁰ okunacaktır. Herhangi bir butona basıldığında ise sürülen sütun, butonun sütununa denk geldiği anda butonun bulunduğu satırda varsayılandan farklı değer okunur. Sütunlar sıra ile sürüldüğünden dolayı aktif sütun yazılım tarafında bilinmektedir, benzer şekilde hangi satırdan varsayılan harici değer okunduğu bilgisi de dikkate alınarak basılan butonun sütun-satır değerleri dolayısıyla buton değeri tespit edilebilir.

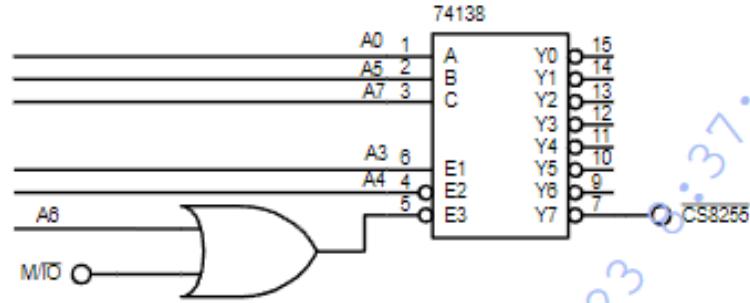
Örnek 18-7

8086 G/Ç uzayında 0A9H adresinden itibaren ardışık tek adreslere 8255 yerleştirilerek, bir adet ortak katotlu yedi parçalı gösterge ve satırları indirme dirençleri ile sürülen 4x3 tuş tarama takımı sürülmek istenmektedir. 8255 PortA uçlarına (PA7-PA0) sırasıyla ortak katotlu yedi parçalı gösterge veri uçları (hgfdcbba) sırasıyla bağlanmıştır. 8255 PortB uçlarıyla (PB3-PB0) 4x3 tuş tarama takımı sütunları sürülmektedir. 8255 PortCL uçları ile 4x3 tuş tarama takımı satırları okunmaktadır. Buna göre, 4x3 tuş tarama takımında basılan buton değerinin buton bırakıldığı anda yedi

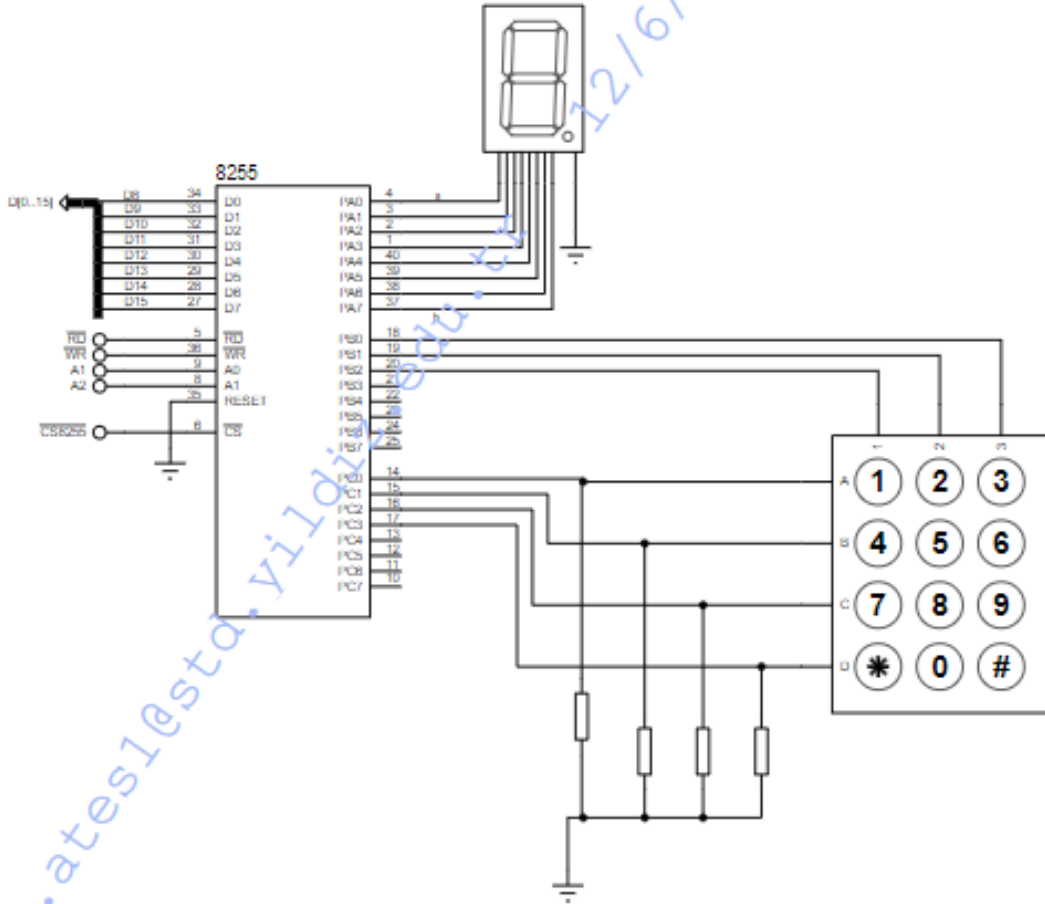
²⁰⁹ Kaldırma dirençleri kullanılıyorsa sütuna lojik 0, indirme dirençleri kullanılıyorsa sütuna lojik 1 verilmelidir.

²¹⁰ Kaldırma dirençleri kullanılıyorsa tüm satırlardan lojik 1, indirme dirençleri kullanılıyorsa tüm satırlardan lojik 0 verilmelidir.

parçalı göstergede yakılabilmesi için, donanım bağlantılarını gerçekleştirerek gerekli assembly kodunu yazınız.



Şekil 18-21: Örnek 18-7 için adres çözümleme devresi



Şekil 18-22: Örnek 18-7 için donanımsal bağlantılar

```

; -----
; PROGRAM : Örnek 18-7.ASM
; 8255 kullanarak, PortA ile 7 parçalı gösterge, PortB ile tuş tarama
; takımı sürerek, basılan tuş karşılığını 7 parçalı göstergede
; yakan assembly program
; -----
                PAGE 60,80
                TITLE tuş tarama takımı ve 7 parçalı gösterge
STAK            SEGMENT PARA STACK 'STACK'
                DW 20 DUP(?)
                ENDS
STAK            SEGMENT PARA 'DATA'
DATA            DB 4FH, 7DH, 67H, 40H, 5BH, 6DH, 7FH, 3FH, 06H, 66H, 07H, 49H
DATA            ENDS
CODE            SEGMENT PARA 'CODE'
                ASSUME CS:CODE, DS:DATA, SS:STAK
START           PROC FAR ; ana yordam
; -----
; Dönüş için gerekli olan değerler yığında saklanıyor
; -----
                PUSH DS
                XOR AX, AX
                PUSH AX
; -----
; DATA ismiyle tanımlı kesim alanına erişebilmek için gerekli tanımlar
; -----
                MOV AX, DATA
                MOV DS, AX
                MOV AL, 81H ; PortA, PortB, PortCH output
                OUT 0AFH, AL ; PortCL input yönlü
                MOV AL, 00H
                OUT 0ABH, AL ; tuş tarama takımında hiçbir sütun aktif değil
                OUT 0A9H, AL ; 7 parçalı göstergede tüm parçalar sönmük
ENDLESS:        MOV DX, 00H
                MOV CL, 3 ; tuş tarama takımında üç sütun için tekrarlar
                MOV BL, 08H ; BL'nin 3. Bit'i 1, diğer bit'leri 0
SUTUN:          SHR BL, 1 ; ilk geçişte BL'nin 2. Bit'i, ikinci geçişte
                        ; 1. bit'i, üçüncü geçişte 0. Bit'i 1,
                        ; diğer bit'leri 0
                MOV AL, BL
                OUT 0ABH, AL ; ilk geçişte 1 no'lu sütun, ikinci geçişte
                        ; 2 no'lu sütun, üçüncü geçişte 3 no'lu sütun
                        ; aktif edilir
                IN AL, 0ADH ; satırları oku
                AND AL, 0FH ; PortCL bit'lerini koru,
                        ; PortCH bit'lerini maske ile 0'la
                CMP AL, 00H ; aktif sütundan herhangi bir tuşa
                JE DEVAM1 ; basılmamışsa sonuç 0, bir sonraki sütuna geç
                PUSH AX ; basılmışsa satır değerlerini yığına sakla
TUSBASILI:      IN AL, 0ADH
                AND AL, 0FH
                CMP AL, 00H
                JNE TUSBASILI ; basılı tuş bırakılana kadar bekle
                POP AX
                MOV DH, AL ; okunan satır değeri DH'ta
                MOV CH, CL
                DEC CH
                MOV AL, 4
                MUL CH ; AX<-aktif olan sütunun ilk tuşu baştan
                        ; kaçınıcı sırada
                MOV DL, 0
                SHR DH, 1 ; DH'taki 1 değeri kaçınıcı bitte
                CMP DH, 00H
                JE DEVAM2 ; DH<-0, DL<-basılı tuşun satır numarası

```

Örnek 18-8

8086 G/Ç uzayında 200H adresinden itibaren ardışık çift adreslere 8255 yerleştirilmek isteniyor. 8255 GrupA kip 1'de giriş yönlü kullanılarak 4×4 tuş tarama tümdevresi (74922) ile GrupB ise kip 1'de çıkış yönlü kullanılarak yedi parçalı göstergesürme tümdevresi (4511) ile haberleşecek şekilde kullanılacaktır. Donanımsal bağlantılar Şekil 18-23 ile verilmiştir.

74922 4×4 tuş tarama takımında herhangi bir tuşa basıldığında önce tuş değerine karşılık ikili değeri iç yazmacında oluşturduktan hemen sonra, DA ucunda lojik 1 seviyesi oluşturur. 74922; OE ucunda lojik 0 seviyesi oluşturulduğunda ise iç yazmaçta saklanan değer veri uçlarına çıkartır. 4511; LE/STB ucunda lojik 0 seviyesi oluşturulduğunda veri uçlarındaki BCD değere karşılık ortak katotlu yedi parçalı göstergede verilen değeri oluşturur. 4511 tümdevresinin bir onay bildirim ucu bulunmamaktadır, bu sebeple LE/STB ucuna verilen değer bir süre geciktirilecek (iki NOT kapısından geçirilerek) sanal bir onay bildirimi sağlanmıştır.

Verilenlere göre 8255 GrupA, giriş yönlü kip 1'de 74922 ile ve GrupB, çıkış yönlü kip 1'de 4511 ile haberleşecek şekilde ayarlayan ve basılan tuş değerine karşılık yedi parçalı göstergede bu değeri yakan assembly kodunu yazınız.

```

;-----
; PROGRAM : Örnek 18-8.ASM
; 8255 kullanarak kip 1 giriş ile tuş tarama takımını okuyup
; kip 1 çıkış ile 7 parçalı göstere süren assembly program
;-----
                PAGE 60,80
                TITLE 8255 kip 1 giriş kip 1 çıkış
CODE            SEGMENT PARA 'CODE'
                ASSUME CS:CODE
START          PROC FAR ; ana yordam
                MOV DX, 206H
                MOV AL, 09H
                OUT DX, AL ; BSR kip bit set
                MOV AL, 05H
                OUT DX, AL ; BSR kip bit set
                MOV AL, 0BCH
                OUT DX, AL ; Kip 1 haberleşme
ENDLESS:        MOV DX, 204H ; 74922 hazır olana kadar bekle
                IN AL, DX
                TEST AL, 08H
                JE ENDLESS
                MOV DX, 200H ; 74922'den veriyi oku
                IN AL, DX
                MOV BL, AL
L2:             MOV DX, 204H ; 4511 hazır olana kadar bekle
                IN AL, DX
                TEST AL, 01H
                JE L2
                MOV DX, 202H ; 4511'e veriyi yaz
                MOV AL, BL
                OUT DX, AL
                JMP ENDLESS ; sonsuz döngüde tekrarla
START          ENDP
CODE           ENDS
                END START

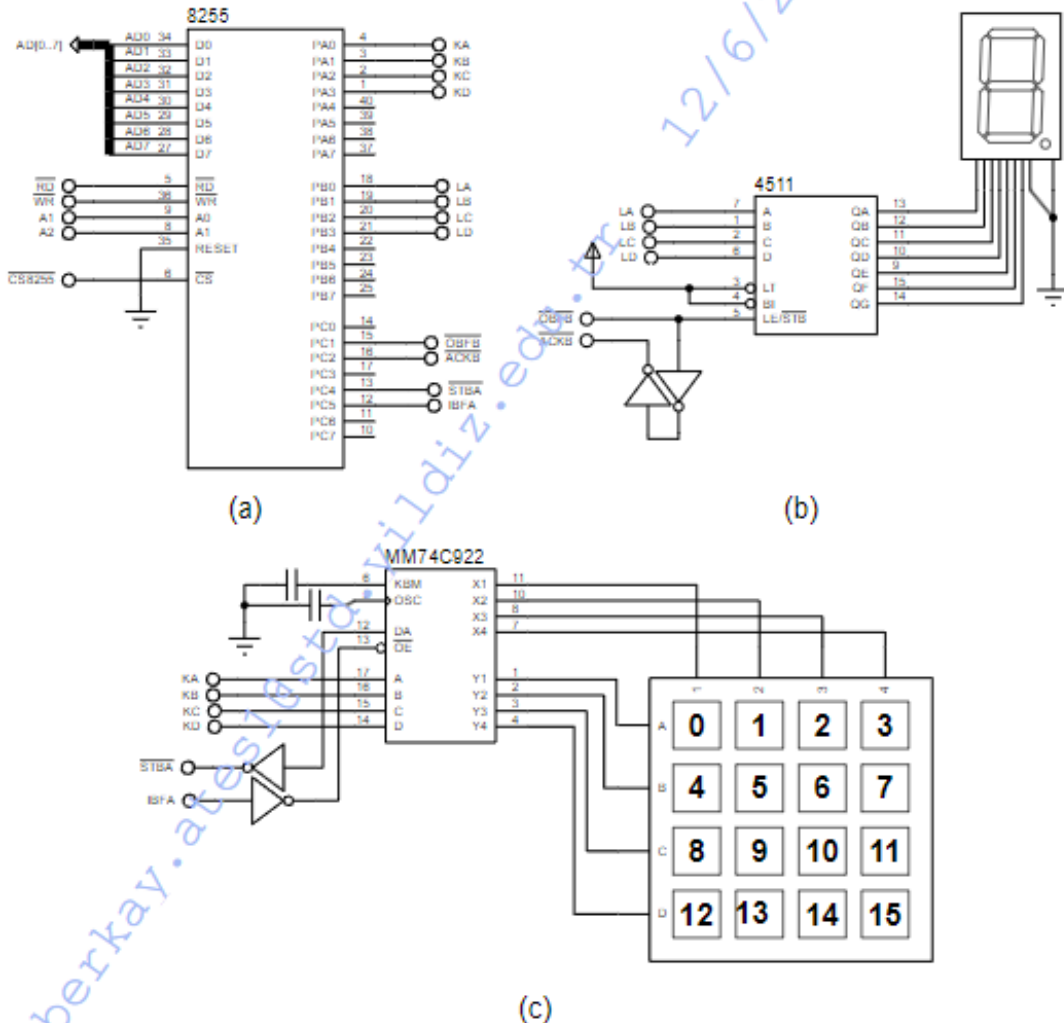
```

```

        INC DL
        JMP DEVAM3
DEVAM2:  ADD AX, DX      ; AX <- basılan tuş indisi
        MOV SI, AX
        MOV AL, DIGITS[SI]
                                ; AL<-basılan tuşa karşılık 7 parçalı gösterge verisi
                                ; tuş değeri 7 parçalı göstergede yakılır
DEVAM1:  OUT 0A9H, AL
        LOOP SUTUN      ; bir sonraki sütun
        JMP ENDLESS     ; sonsuz döngüde tekrarla
        RETF
START    ENDP
CODE     ENDS
        END START

```

18.4.5 Kip 1 Uygulaması



Şekil 18-23: Örnek 18-8 için donanımsal bağlantılar