



**SAKARYA ÜNİVERSİTESİ**  
**Bilgisayar ve Bilişim Bilimleri Fakültesi**  
**Bilgisayar Mühendisliği Bölümü**

# **Mikroişlemcili Sistemler ve Laboratuvarı**

## **4.HAFTA**

Çok sayıda porta ihtiyaç duyulduğunda basit giriş/çıkış birimleri yetersiz kalır.

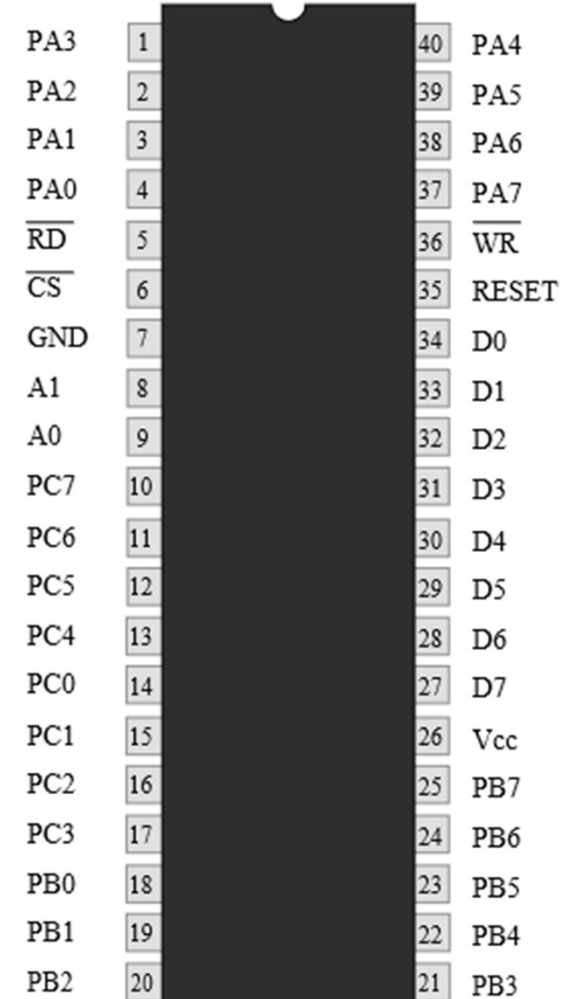
8255; Intel ve birçok diğer mikroişlemciler için kullanılabilen, genel amaçlı programlanabilir bir I/O cihazıdır.

Ayrı ayrı programlanabilen 12'şer pinden oluşan 2 grup halinde bulunan ve 3 ana modda çalışan 24 adet I/O pinine sahiptir.

Mod 0'da, her gruptaki 12 I/O pini, 4'lü ve 8'li kümeler halinde giriş yada çıkış olarak programlanabilir.

Mod 1'de, her gruptaki 8'li hat giriş yada çıkış olarak programlanabilirken, geri kalan 4 pinden 3'ü elşıkışma ve kesme kontrol sinyalleri olarak kullanılır.

Mod 2 çift yönlü hat konfigürasyonu olarak düzenlenmiştir.



## 8255 Pin Detayları

**D0 - D7** pinleri cihaz için veri giriş/çıkış hatlarıdır. Bütün bilgi bu 8 veri hattından 8255'e yazılır yada 8255'ten okunur.

**CS** (*Chip Select Input*). Pin lojik-0 ise 8255 aktif olur ve mikroişlemci 8255'e veri yazar yada 8255 üzerinden veri okur.

**RD** (*Read Input*)

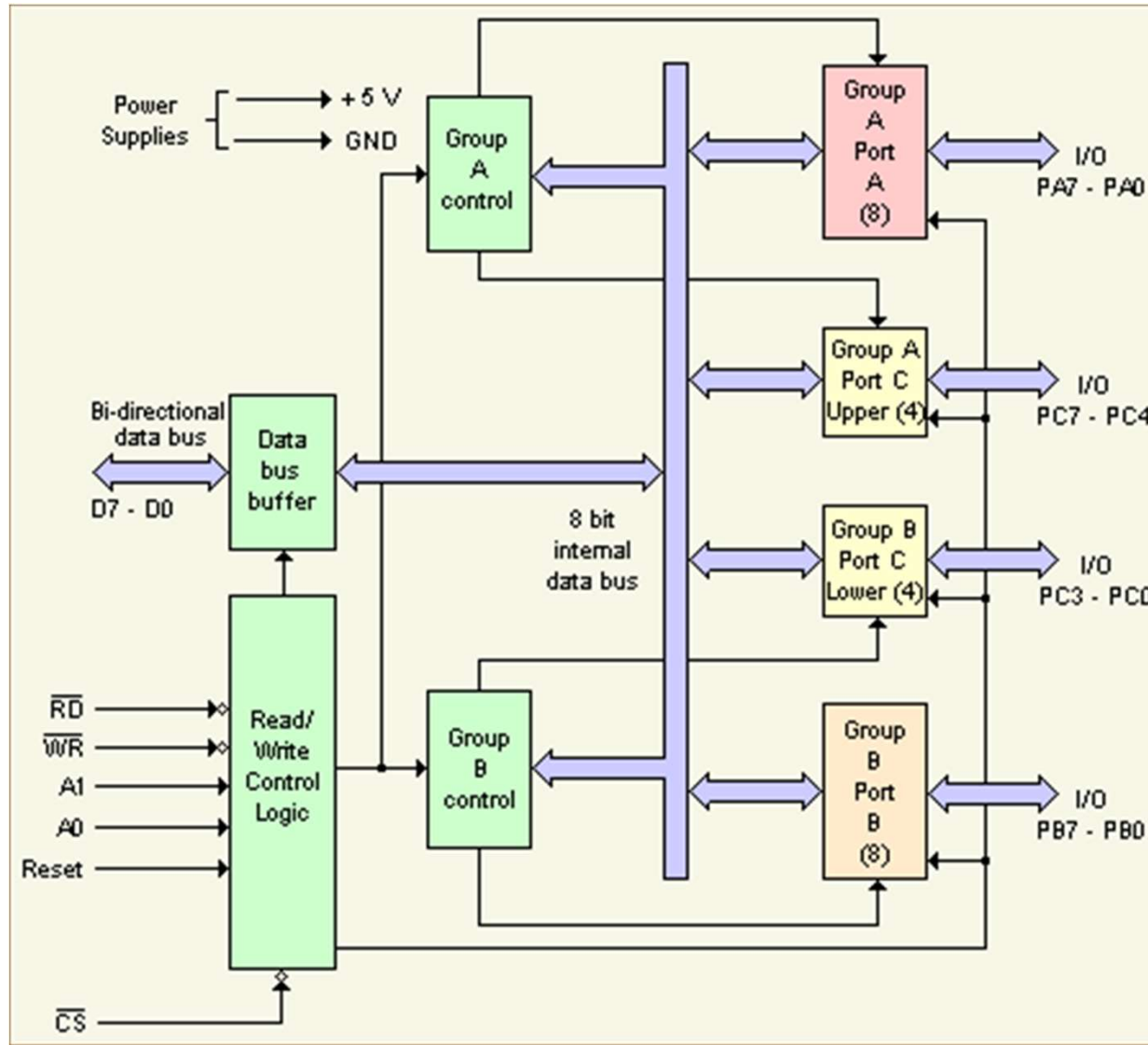
**WR** (*Write Input*)

**A0 - A1** (*Address Inputs*) Bu pinlerin mantıksal kombinasyonuna göre (00, 01, 10 ve 11) hangi dahili kaydedicini kullanılacağı belirlenir.

**RESET**

**PA0 - PA7, PB0 - PB7, PC0 - PC7** Bu sinyaller 8-bit I/O portları olarak kullanılır. Başka çevre birimlerine bağlanabilir. 8255; 3 adet 8 bit I/O porta sahiptir ve herbiri harici cihazın fiziksel hatlarına bağlanabilir. Port A (PA), Port B (PB) ve Port C (PC) olarak adlandırılır.

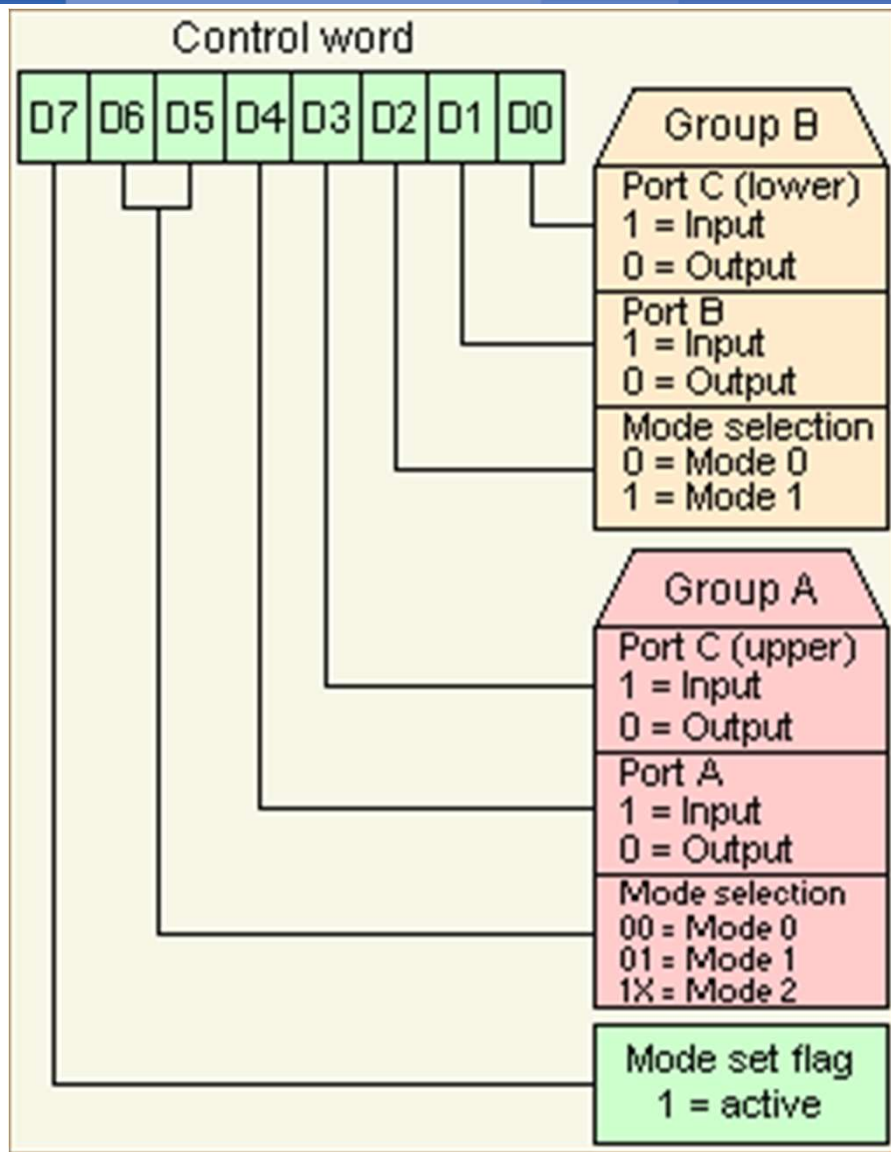
# 8255 Blok Diyagramı



Port A ve Port B 8 bit olarak kullanılırken, Port C ise mod seçeneğine göre Port A ya da Port B ile ilişkilendirilebilir yada tamamen bağımsız olarak kullanılabilir.

<http://www.sharpmz.org/mz-700/8255ovview.htm>

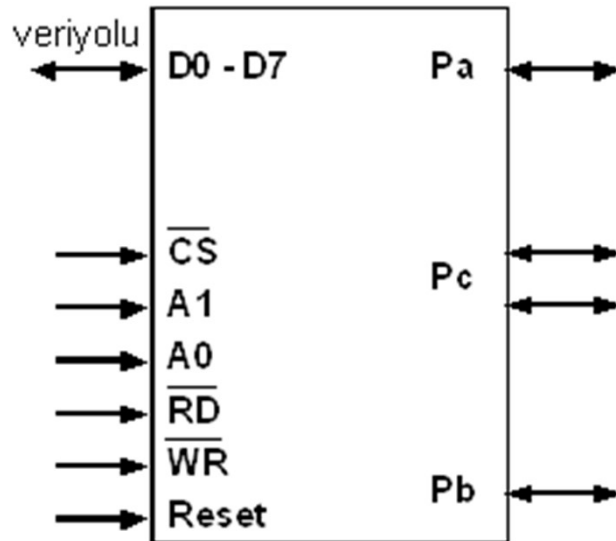
## 8255- Kontrol Kelimesi



8255'i yapılandırmak için detayları şekilde verilen kontrol kelimesindeki (Control Word) D7-D0 pinleri istenilen moda göre yapılandırılarak 8255'e yüklenmelidir.

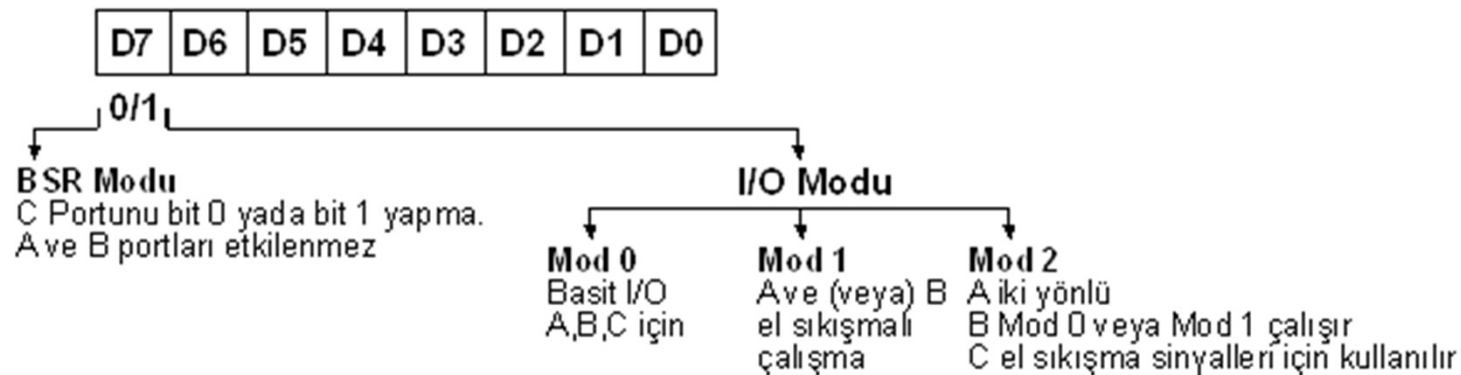
<http://www.sharpmz.org/mz-700/8255ovview.htm>

# 8255 Adresleme A1-A0

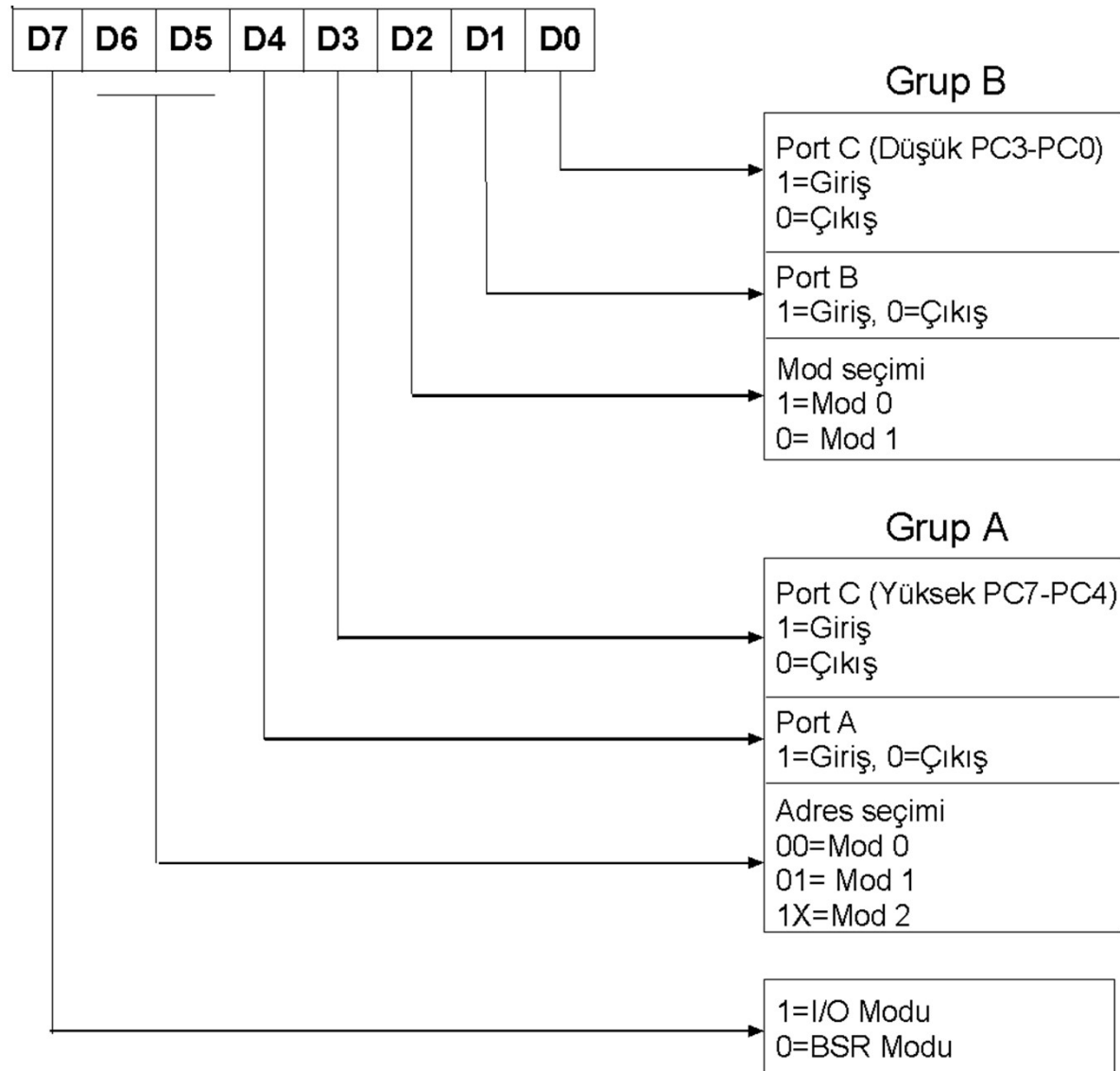


CS	A1	A0	Seçilen	Adres
0	0	0	Port A	C0h
0	0	1	Port B	C1h
0	1	0	Port C	C2h
0	1	1	Kontrol Saklayıcısı	C3h
1	x	x	8255 Seçilemez	

Kontrol Kelimesi



# Kontrol Kelimesi CW-Control Word

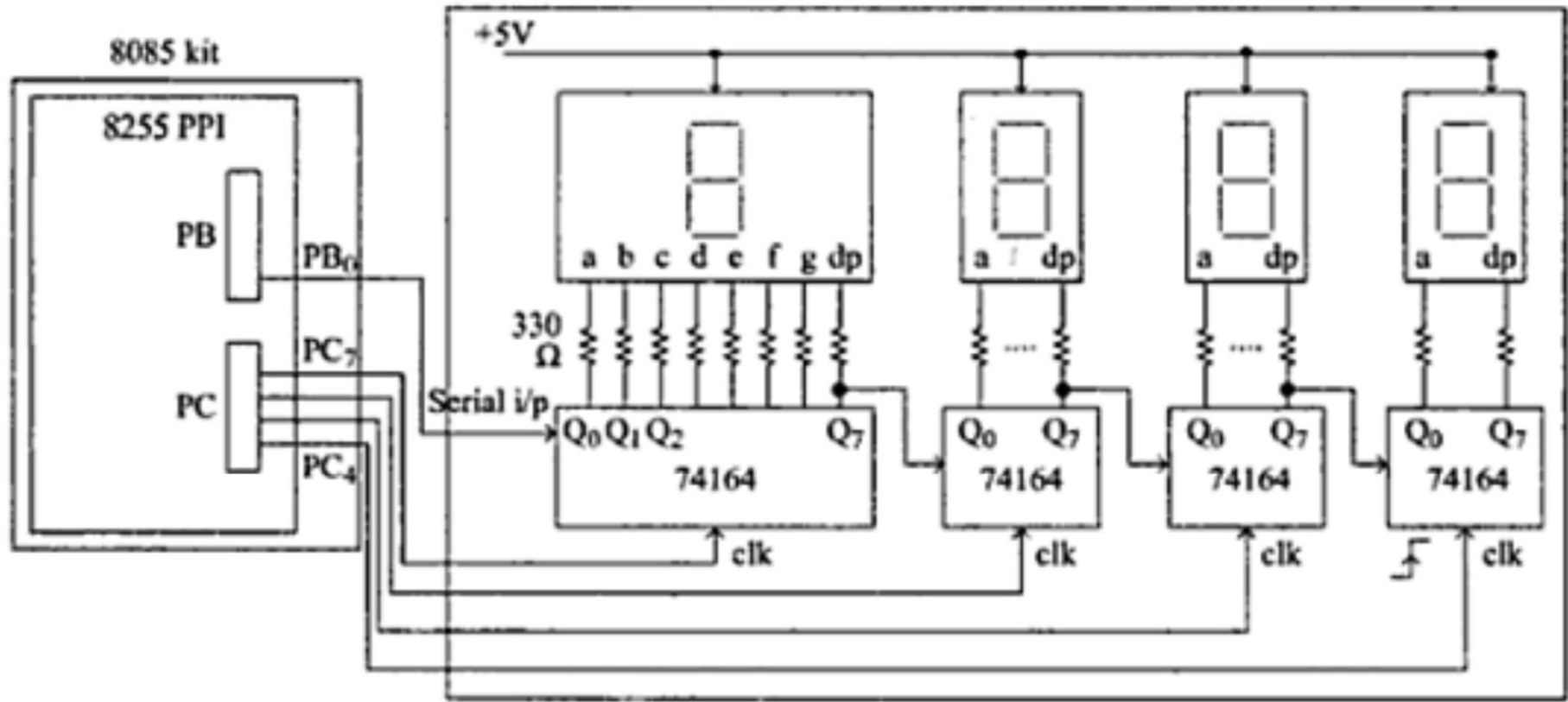


# 8255 Çalışma Modları

- Mod 0 – Basit giriş veya çıkış
  - A ve B portunun tüm kapıları ya alıcı ya da verici konumlanır.
  - C portunun kapıları ise denetim kütüğünün D0-3 bitlerine uygun olarak konumlanır.
- Mod 1 – El sıkışma (handshaking) ile basit giriş veya çıkış
  - A ve B portunun tüm kapıları ya alıcı ya da verici konumlanır.
  - C portunun üst kısmı A ve alt kısmı B portuna el sıkışma (handshaking) işlemleri için destek verir.
  - PC2, B portu için hazır giriş olarak görev yapar. PC1 ise B portu için A1 girişi olarak görev yapar.
  - Hazır bilgisinin alınması ile kesme üretilmek isteniyorsa PC0 bu amaçla kullanılabilir.
- Mod 2 – El sıkışma ile iki yönlü I/O
  - Sadece A portu ve C üst için geçerlidir.
  - Bu modda A portu iki yönlü kullanılabilir.



# Örnek: 8255 üzerinden 7 parçalı gösterge sürme



- Bu örnekte 7 Parçalı Göstergede gösterilmek isteten değer karşılığı bit dizisi PB0 hattından seri olarak 74164 (seri giriş/paralel çıkış kaydedici) üzerinden göstergeye aktarılmaktadır.
- PC4-PC7 hatlarından ise hangi göstergenin aktif olacağı bilgisi clk sinyali ile belirlenmektedir.

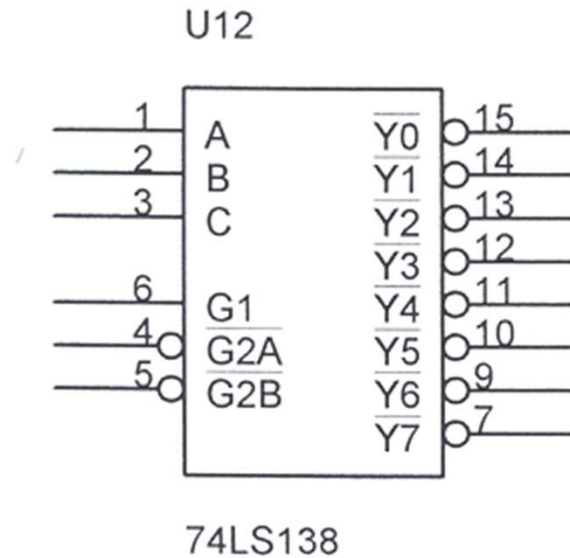
# BELLEK ORGANİZASYONU

- Mikroişlemciye bağlanan tüm çevre birimleri (Bellekler, I/O portları vb.) **adreslenebilir** alanlardır ve bu birimlere veri yazarken yada birimlerden veri okurken adresleri üzerinden işlem yapılır.
- Bu sebeple mikroişlemcili sistem tasarımı yapılırken çevre birimlerinin başlangıçta adreslerinin belirlenmesi ve bu adres değerlerine göre kontrol devresi tasarlanmalıdır.
- Kontrol devreleri; adres ve veri yollarını ortak kullanan çevre birimlerinden hangisinin bu yollara erişim hakkı olacağını girilen adres değerine göre donanımsal olarak belirleyen bir tasarımıdır.

# Bellek Organizasyonu

- Kontrol devresi tasarlanırken Adres yolunun yüksek bitleri seçici eleman olarak tercih edilir.
- Kontrol devresi temel mantık kapılarından oluşabileceği gibi dekodерlerden de oluşabilir.

## 74138 Dekoderi

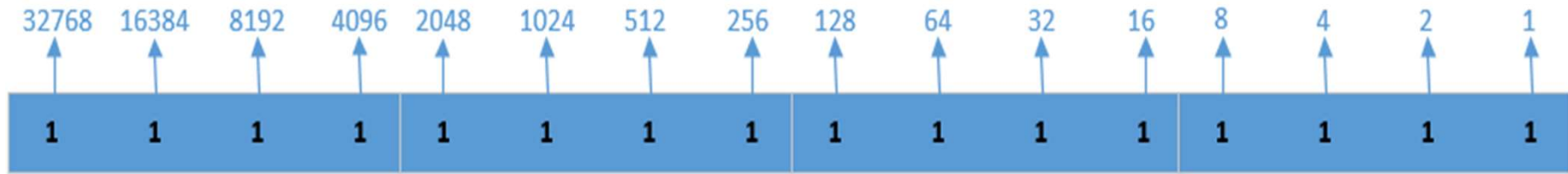


74136, dekoderi çıkışları terslenmiştir. ABC girişlerinden C en yüksek değerliktedir.

3 adet yetki (G1, G2A ve G2B) girişine sahiptir.

Bu girişlerden G1=1, G2A=0 ve G2B=0 olduğu durumda dekoder aktif olur ve ABC girişlerinin durum değerine göre çıkışlardan sadece bir adedini lojik-0 diğerlerini lojik-1 yapar.

# BELLEK ORGANİZASYONU



8	bit	256	
10	bit	1K	0000 ↔ 03FF
11	bit	2K	0000 ↔ 07FF
12	bit	4K	0000 ↔ 0FFF
13	bit	8K	0000 ↔ 1FFF
14	bit	16K	0000 ↔ 3FFF
15	bit	32K	0000 ↔ 7FFF
16	bit	64K	0000 ↔ FFFF

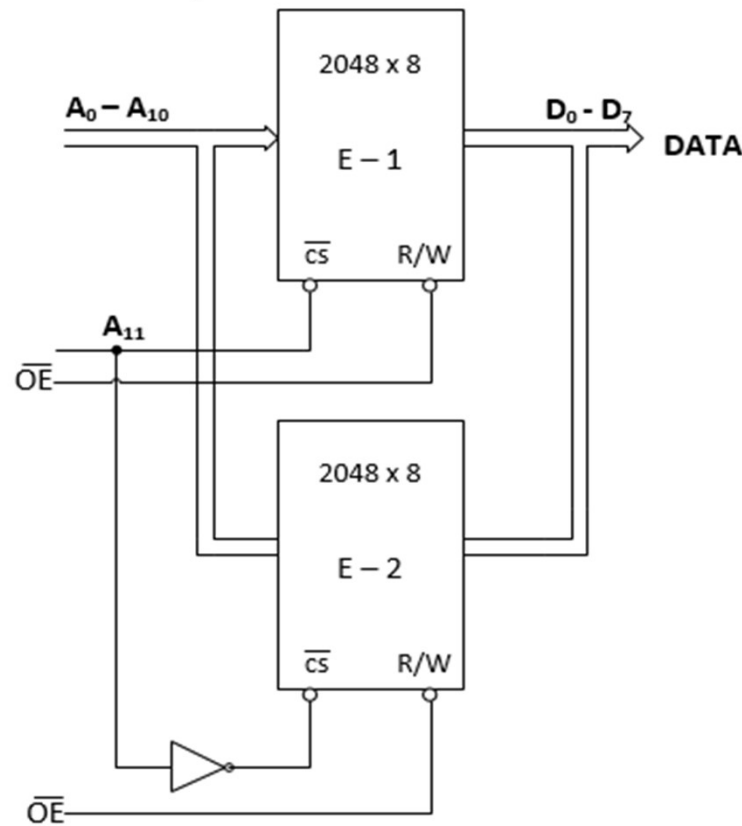
**ÖNEMLİ** : Yüksek değerlikli bitler hafıza birimini seçmek için , düşük değerlikli bitler adres yolu için kullanılırlar.

## ÖRNEK 1

2 adet 2048 x 8 bitlik EPROM ' un 4096 x 8 bitlik bir bellek birimi haline getiriniz

2 adet 2K EPROM bağlantısı

2K -> 0000 - 07FF yani 11 bit



4096 x 8 bit olmuş oldu.

## ÖRNEK 2

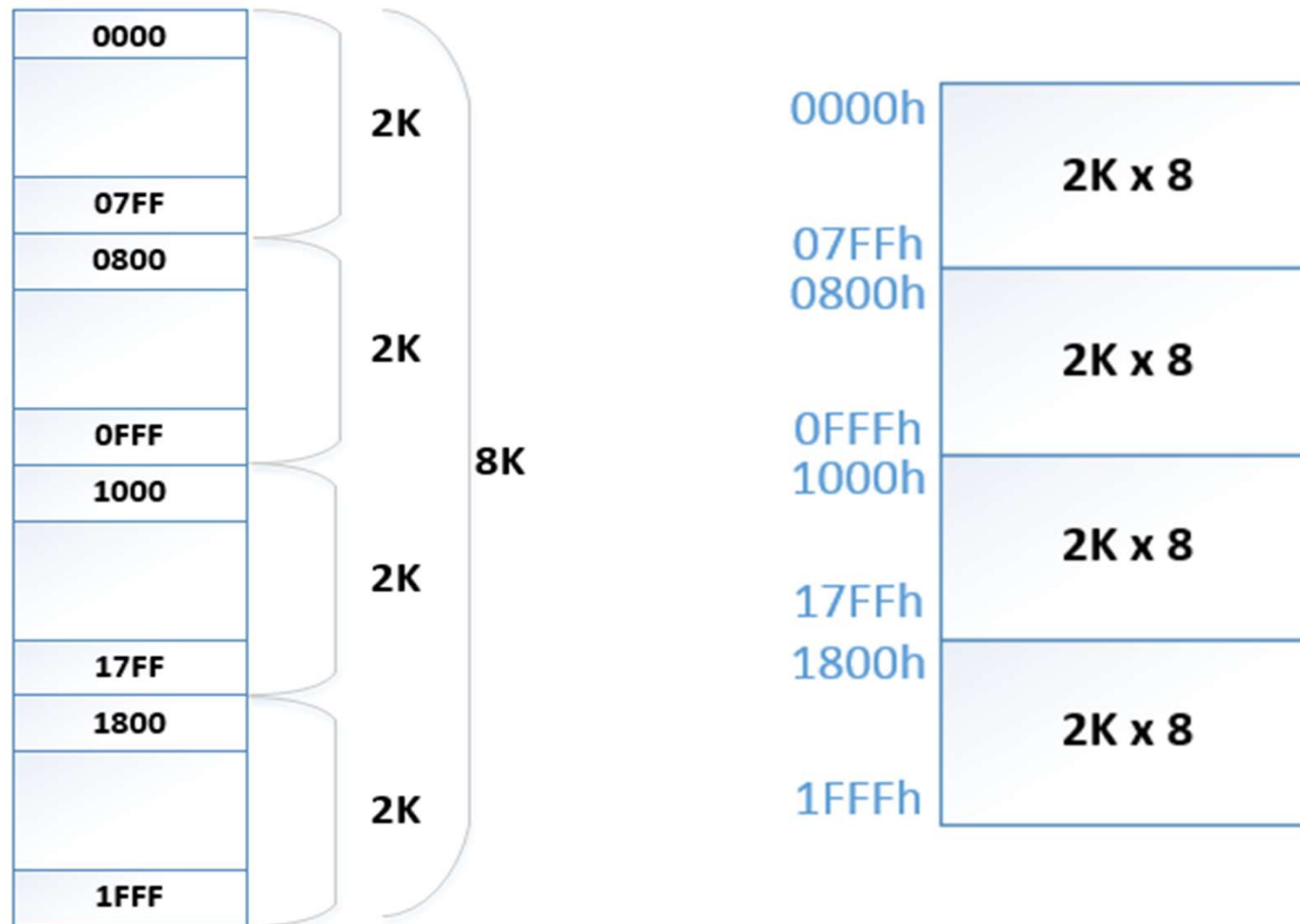
4 adet 2K x 8 lik RAM ' i 8K x 8 bitlik RAM haline getirelim.

8K için 13 bit lazım

2K için 11 bit lazım

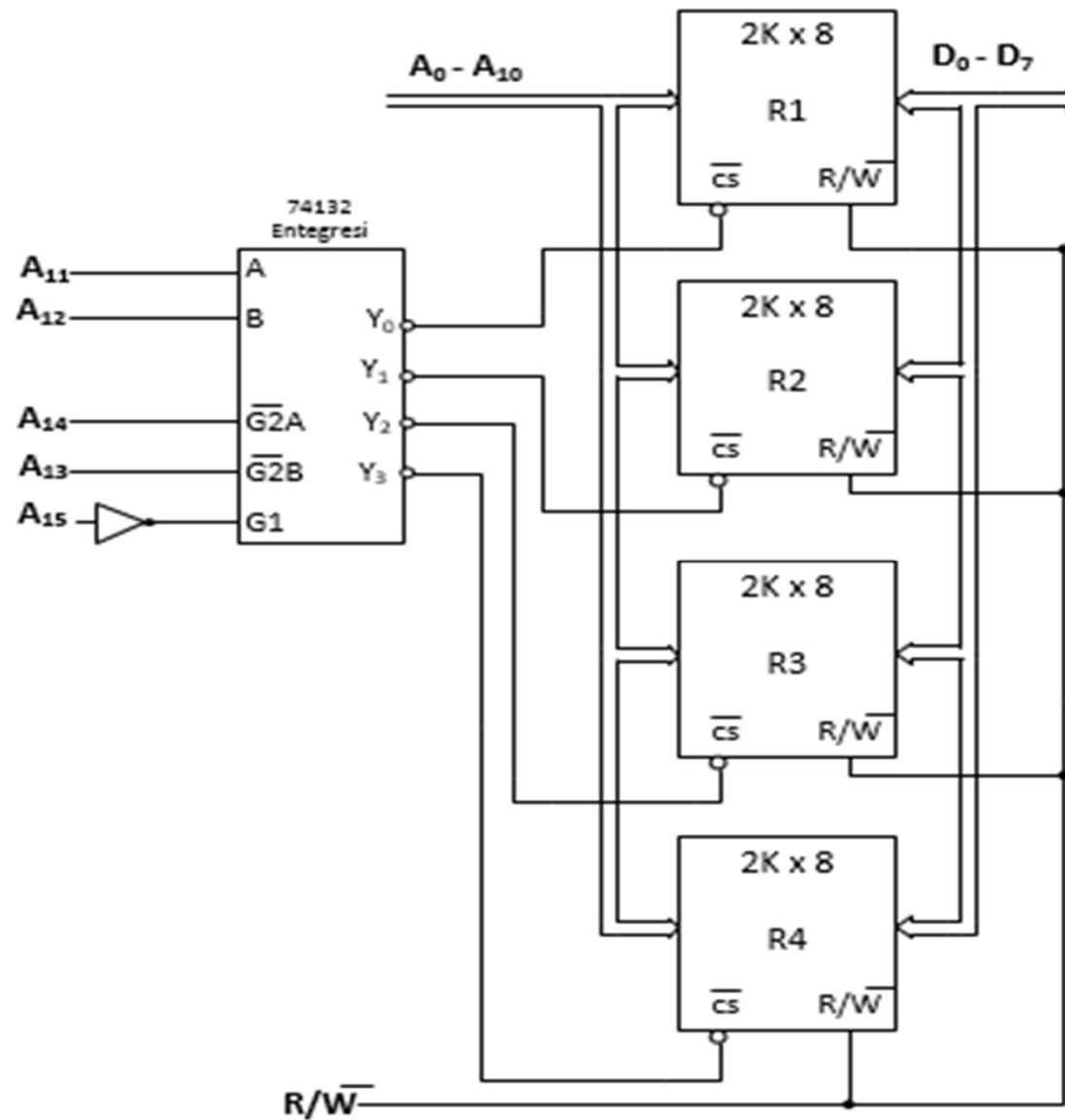
0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	2K R1
0 0 0 0	0 1 1 1	1 1 1 1	1 1 1 1	2K R2
0 0 0 0	1 0 0 0	0 0 0 0	0 0 0 0	
				2K R3
0 0 0 0	1 1 1 1	1 1 1 1	1 1 1 1	
0 0 0 1	0 0 0 0	0 0 0 0	0 0 0 0	2K R4
0 0 0 1	0 1 1 1	1 1 1 1	1 1 1 1	2K R4
0 0 0 1	1 0 0 0	0 0 0 0	0 0 0 0	
				2K R4
0 0 0 1	1 1 1 1	1 1 1 1	1 1 1 1	

## ÖRNEK 2 (DEVAMI)





## ÖRNEK 2 (DEVAMI)



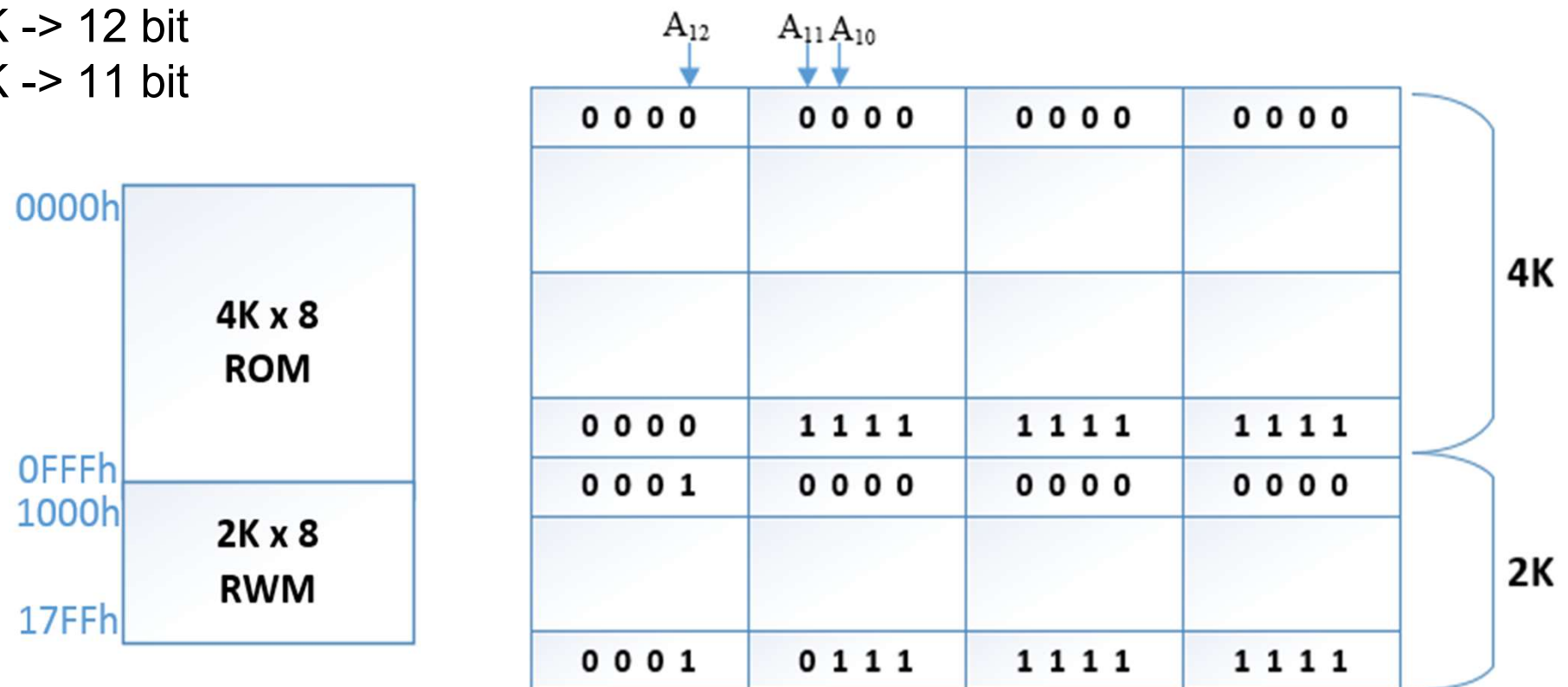
## ÖRNEK 3

4K X 8 EPROM ve 2K X 8 RWM hafızaları adresleyerek tasarımı gerçekleyiniz.

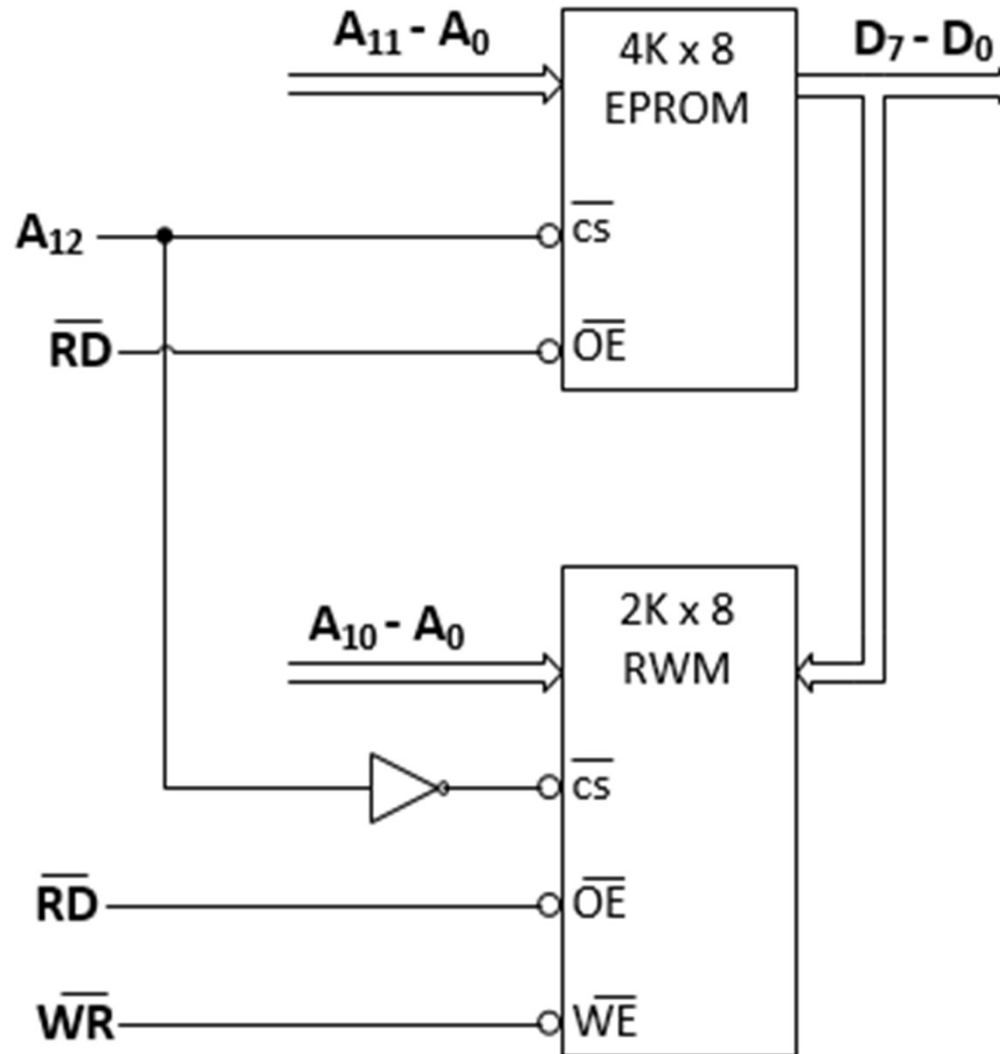
EPROM 0000h adresinden başlayacak ve RWM ardından devam edecek.

4K -> 12 bit

2K -> 11 bit



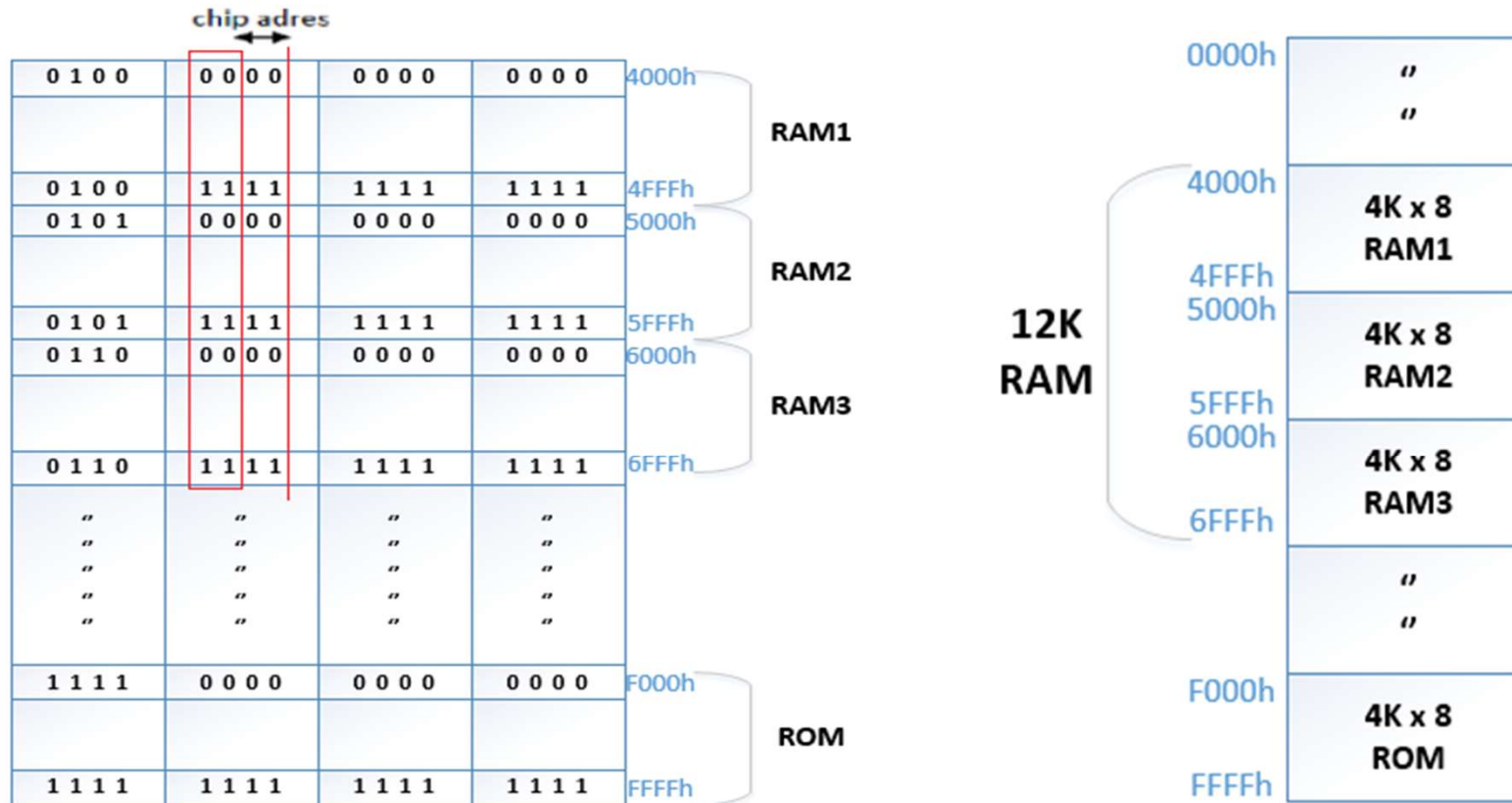
## ÖRNEK 3 (DEVAMI)



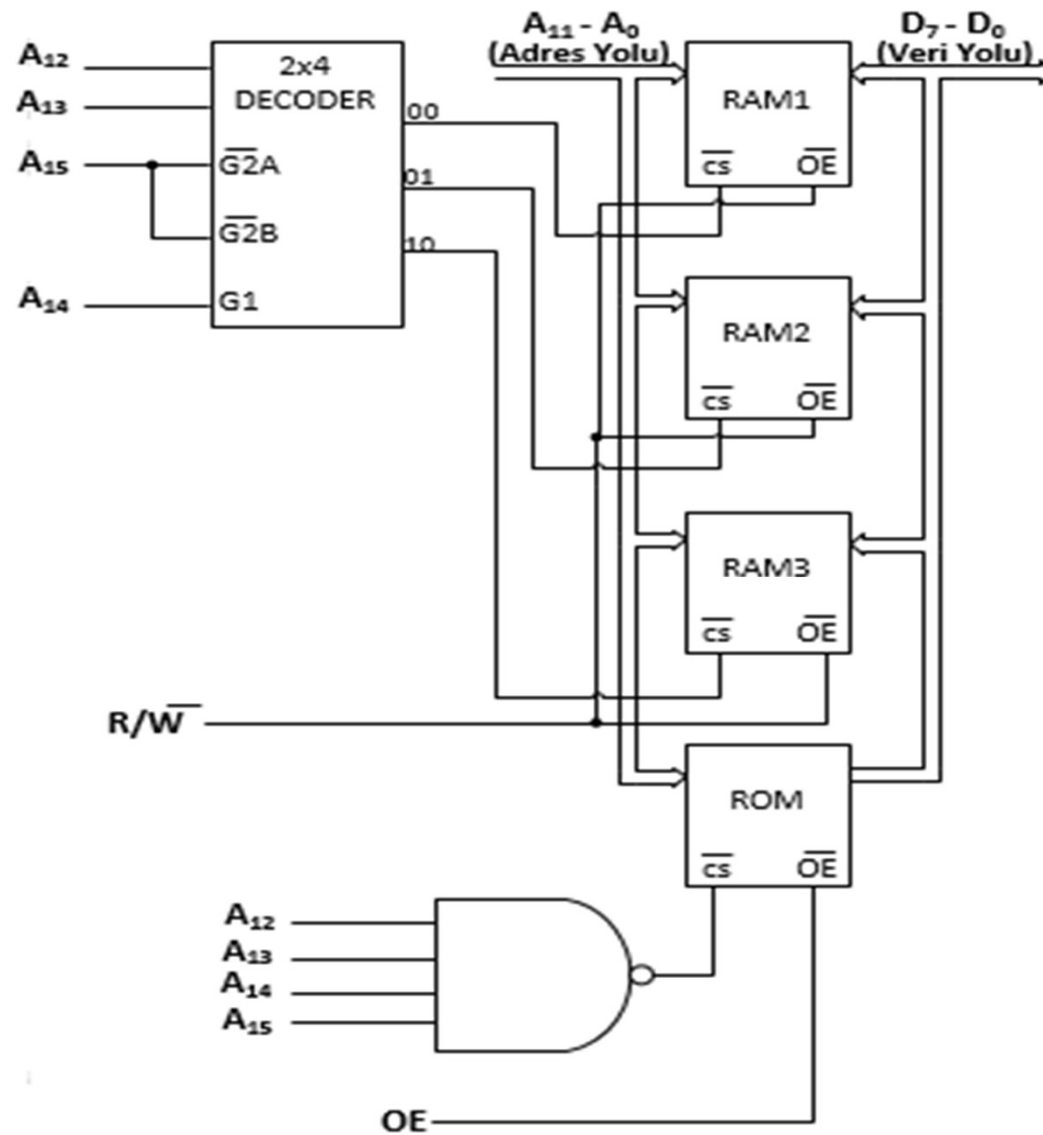
## ÖRNEK 4

4K X 8 RAM ve 4K X 8 ROM kullanarak 12K X 8 RAM ve 4K X 8 ROM belleği oluşturmak isteniliyor. RAM bellek adresi  $(4000h)_{16}$  adresinden ,ROM belleğin başlangıç adresi ise  $(F000h)_{16}$  oluşmaktadır.Bellek mimarisini oluşturunuz.

3 RAM 1 ROM gerekli



## ÖRNEK 4 (DEVAMI)



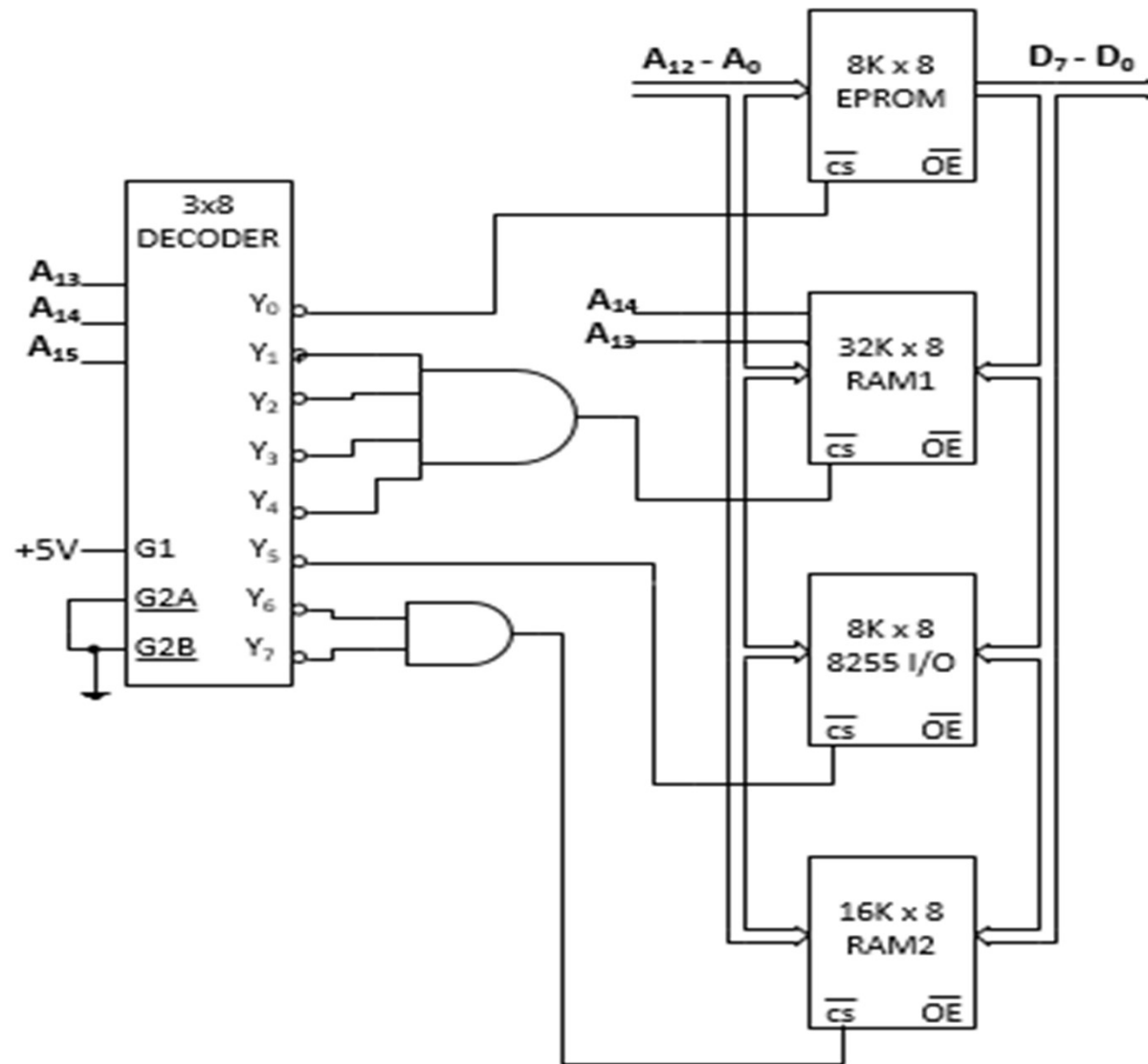
## ÖRNEK 5

8K X 8 EPROM ,32 K X 8 RAM1 , 8255 (8K gibi düşünün) , 16 X 8 RAM2 kullanarak bellek mimarisini oluşturunuz.

0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0000h
0 0 0 1	1 1 1 1	1 1 1 1	1 1 1 1	1FFFh
0 0 1 0	0 0 0 0	0 0 0 0	0 0 0 0	2000h
1 0 0 1	1 1 1 1	1 1 1 1	1 1 1 1	9FFFh
1 0 1 0	0 0 0 0	0 0 0 0	0 0 0 0	A000h
1 0 1 1	1 1 1 1	1 1 1 1	1 1 1 1	BFFFh
1 1 0 0	0 0 0 0	0 0 0 0	0 0 0 0	C000h
1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1	FFFFh



## ÖRNEK 5 (DEVAMI)



## 8085 Hafıza ve IO/M Örneği

FFFF	2Kx8 RAM
E000	
DFFF	BOŞ
C000	
BFFF	BOŞ
A000	
9FFF	BOŞ
8000	
7FFF	BOŞ
6000	
5FFF	BOŞ
4000	
3FFF	8Kx8 RAM
2000	
1FFF	2Kx8 EPROM
0000	

FF	BOŞ
E0	
DF	
C0	BOŞ
BF	
A0	
9F	BOŞ
80	
7F	
60	BOŞ
5F	
40	
3F	BOŞ
20	
1F	
00	4 Byte IO1

