



**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.

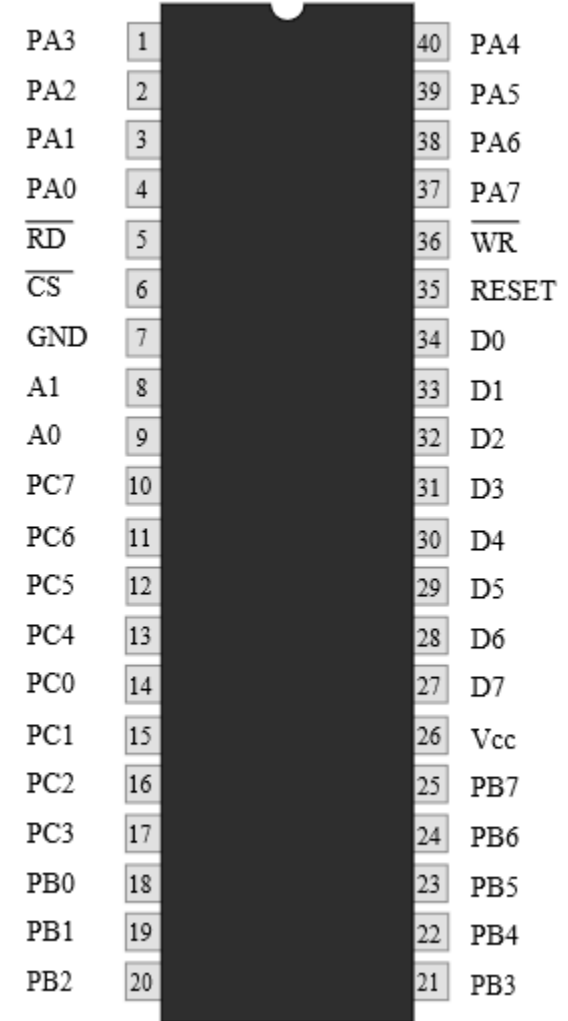
8255; Intel ve birçok diğer mikroişlemciler için kullanılabilen, genel amaçlı programlanabilir bir I/O cihazıdır.

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*)

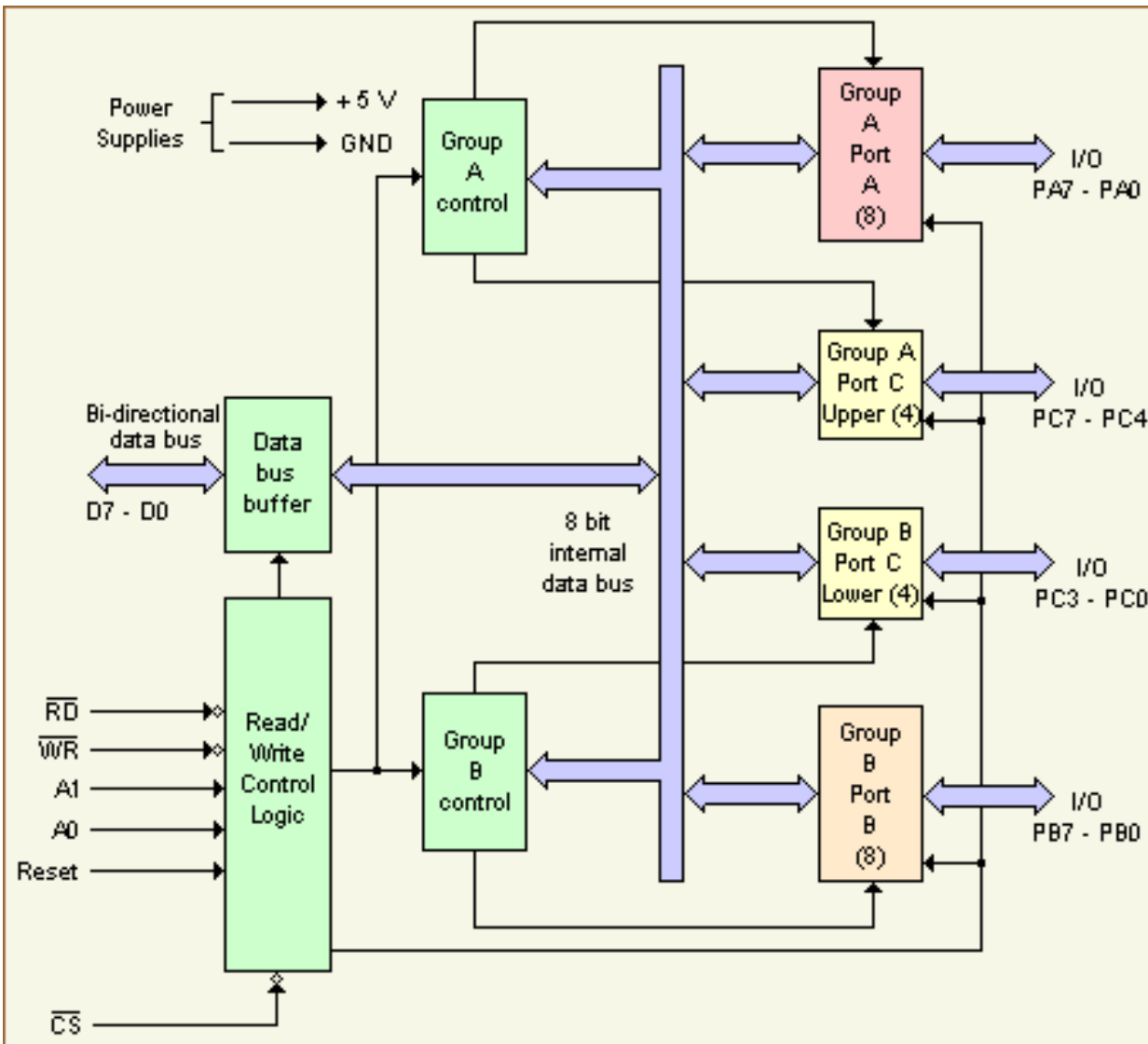
**WR** (*Write Input*)

**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.

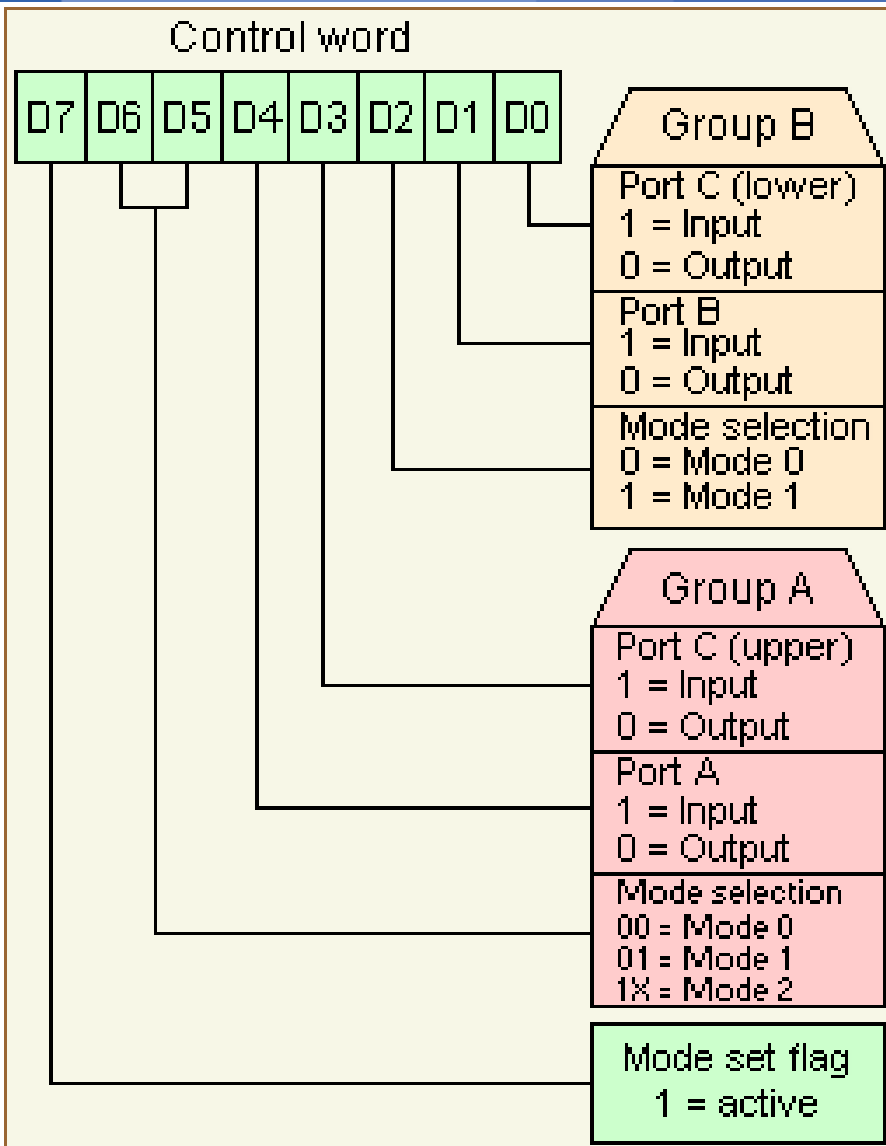
# 8255 Blok Diyagramı



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>

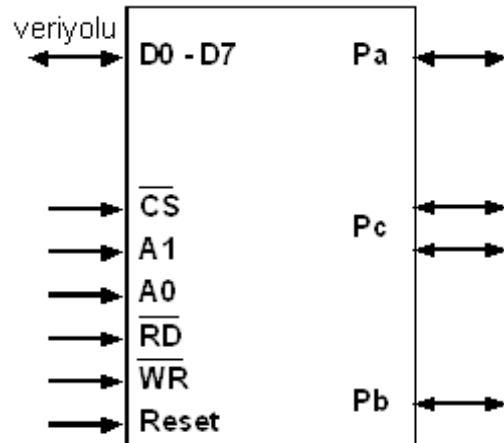
# 8255- Kontrol Kelimesi



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

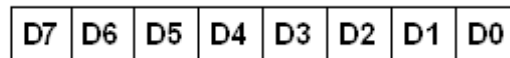
<http://www.sharpmz.org/mz-700/8255ovview.htm>

# 8255 Adresleme A1-A0



CS	A1	A0	Seilen	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 Seilemez	

Kontrol Kelimesi



**BSR Modu**  
C Portunu bit 0 yada bit 1 yapma.  
A ve B portları etkilenmez

**I/O Modu**

**Mod 0**  
Basit I/O  
A,B,C için

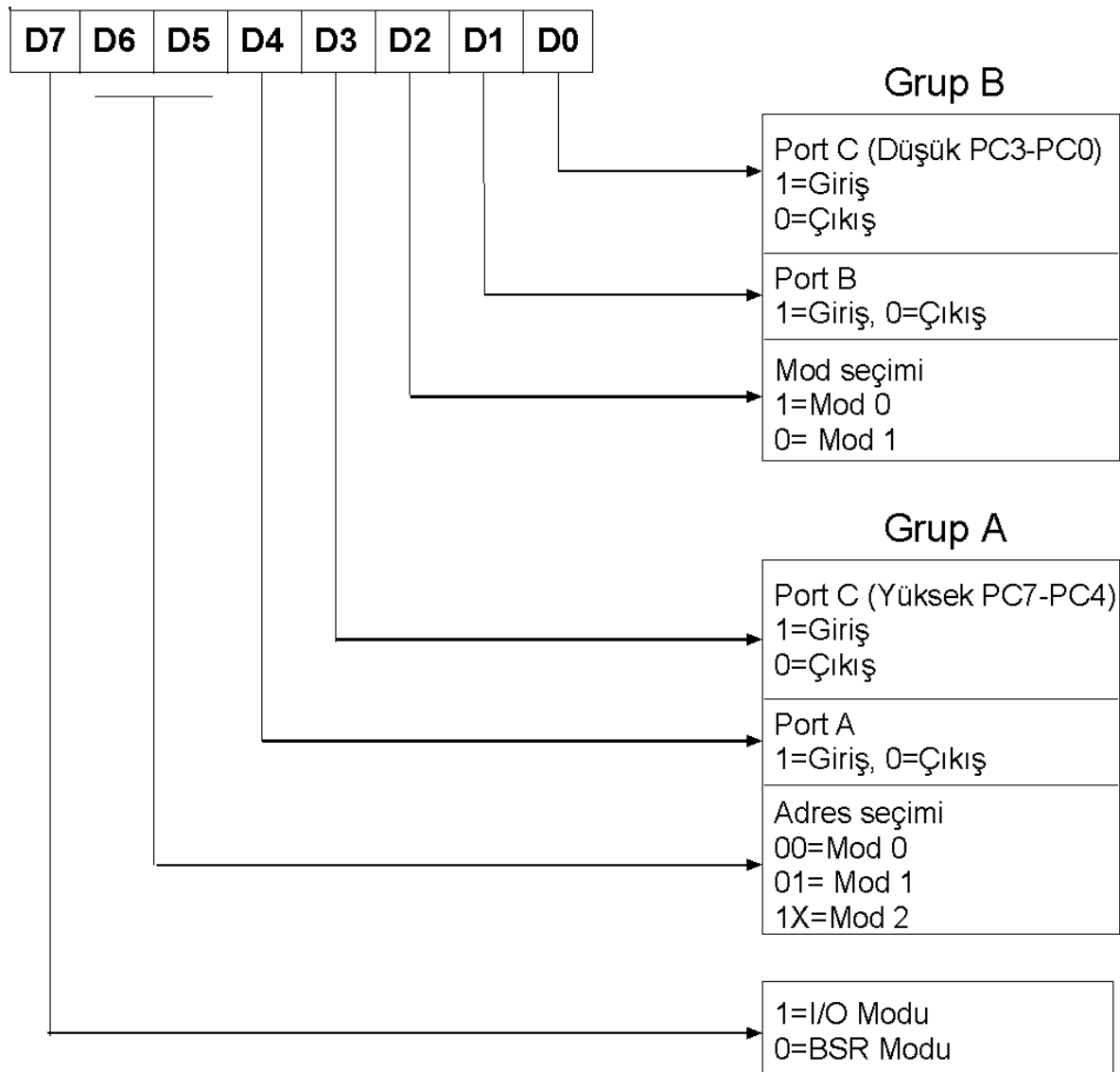
**Mod 1**  
A ve (veya) B  
el sıkışmalı  
alışma

**Mod 2**  
A iki yönlü  
B Mod 0 veya Mod 1 alışır  
C el sıkışma sinyalleri için kullanılır

0 -BSR (Bit Set Reset) Modu: Bu mod yalnızca C portunun bitlerini ayarlamak veya sıfırlamak için kullanılır

1 -I/O Modu

# Kontrol Kelimesi CW-Control Word

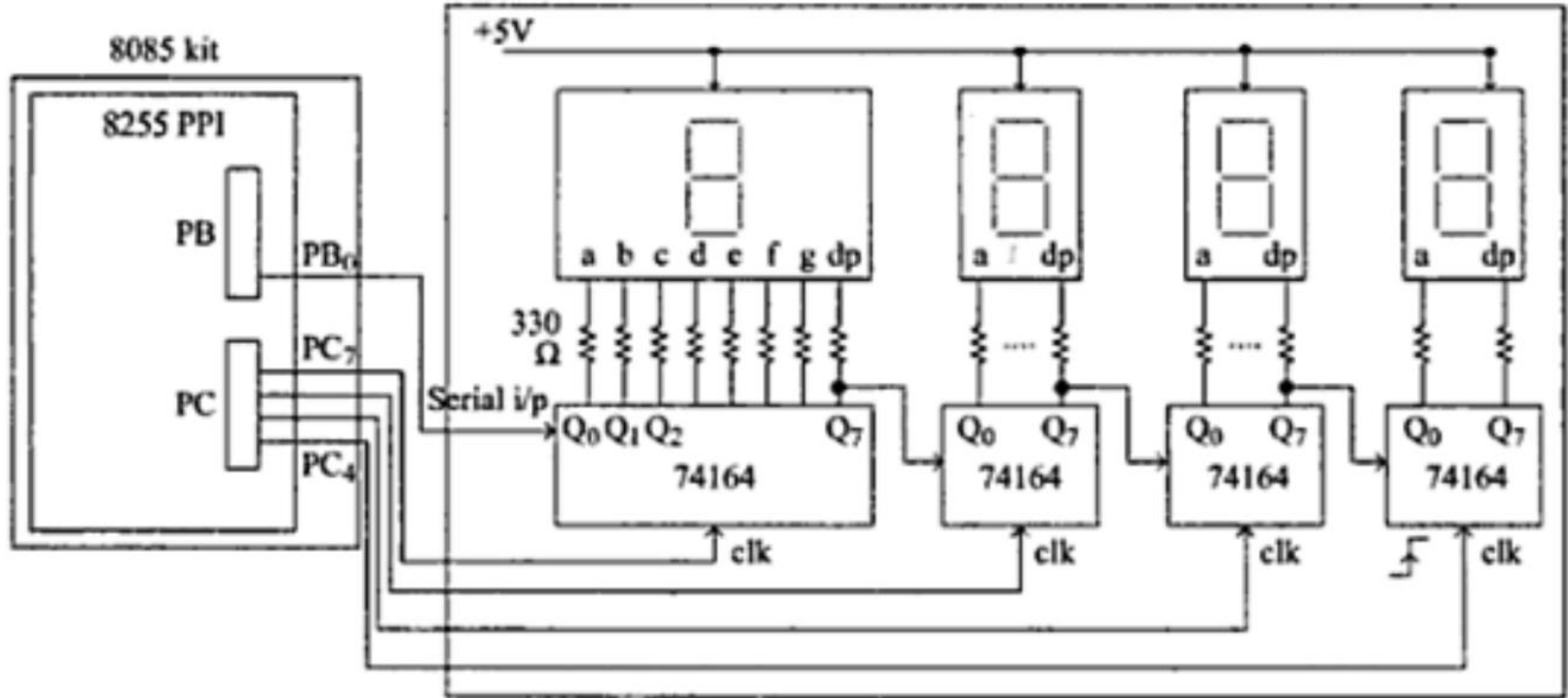


# 8255 Çalışma Modları

- 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



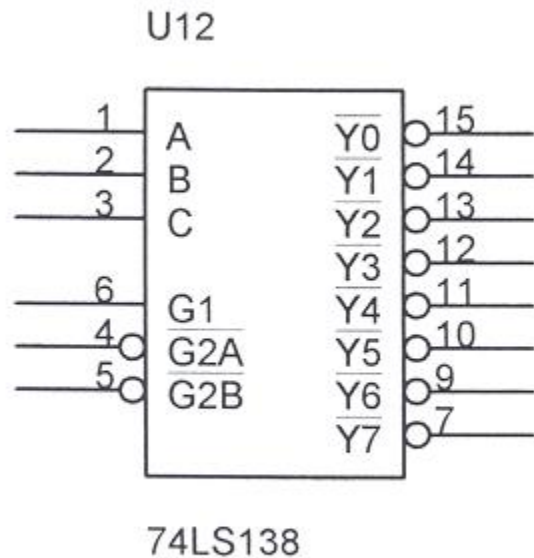
- Bu örnekte 7 Parçalı Göstergede gösterilmek isteten değer karşılığı bit dizisi PB0 hattından seri olarak 74164 (seri giriş/paralel çıkış kaydedici) üzerinden göstergeye aktarılmaktadır.
- PC4-PC7 hatlarından ise hangi göstergenin aktif olacağı bilgisi clk sinyali ile belirlenmektedir.

# BELLEK ORGANİZASYONU/Memory Interfacing

- 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 mikroşlemcili 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.

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

# 74138 Dekoderi

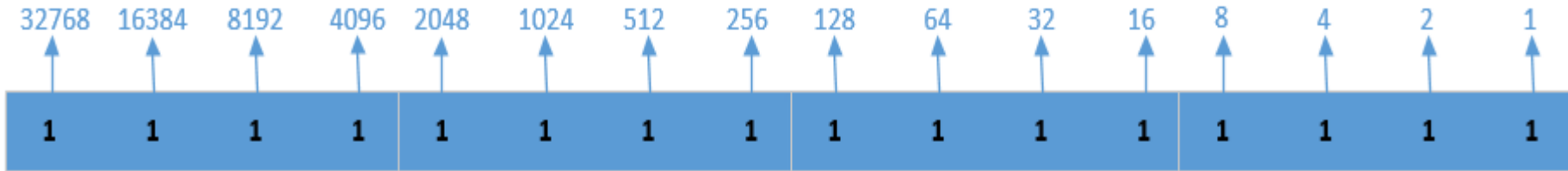


74136, dekoderi çıkışları terslenmiştir. ABC girişlerinden C en yüksek değerliktedir.

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

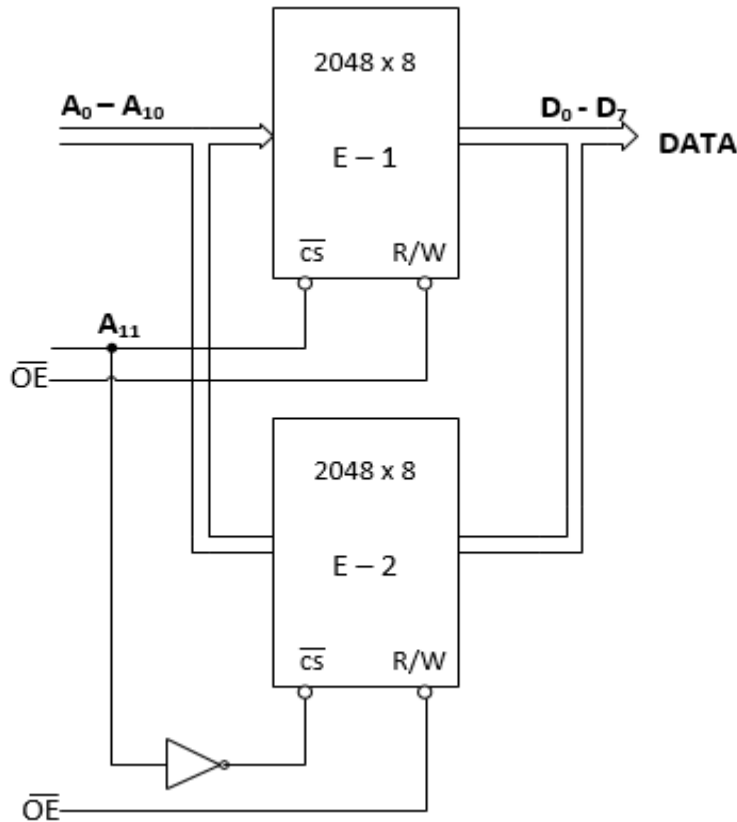
**ÖNEMLİ** : Yüksek değerlikli bitler hafıza birimini seçmek için , düşük değerlikli bitler adres yolu için kullanılırlar.

# 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

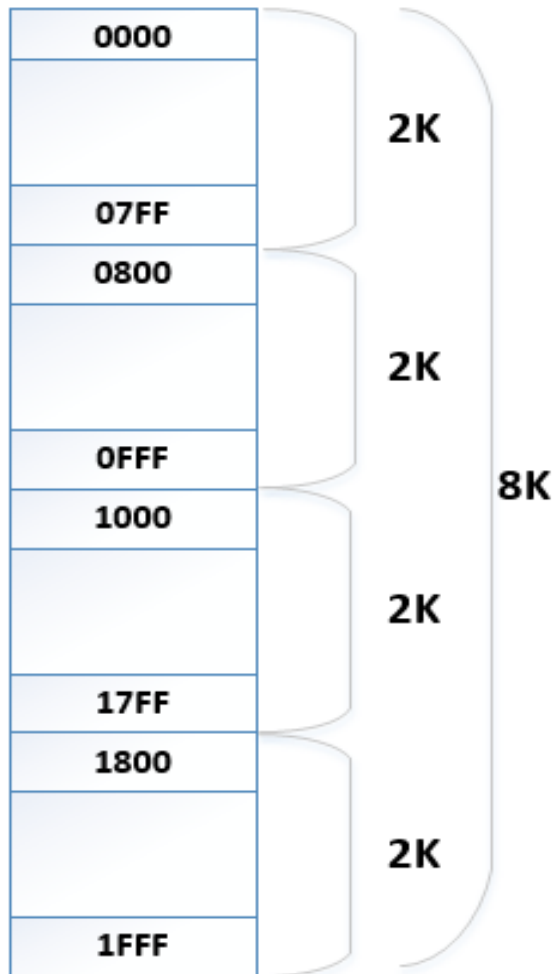
4 adet 2K x 8 lik RAM ' i 8K x 8 bitlik RAM haline getirelim.

8K için 13 bit lazım

2K için 11 bit lazım

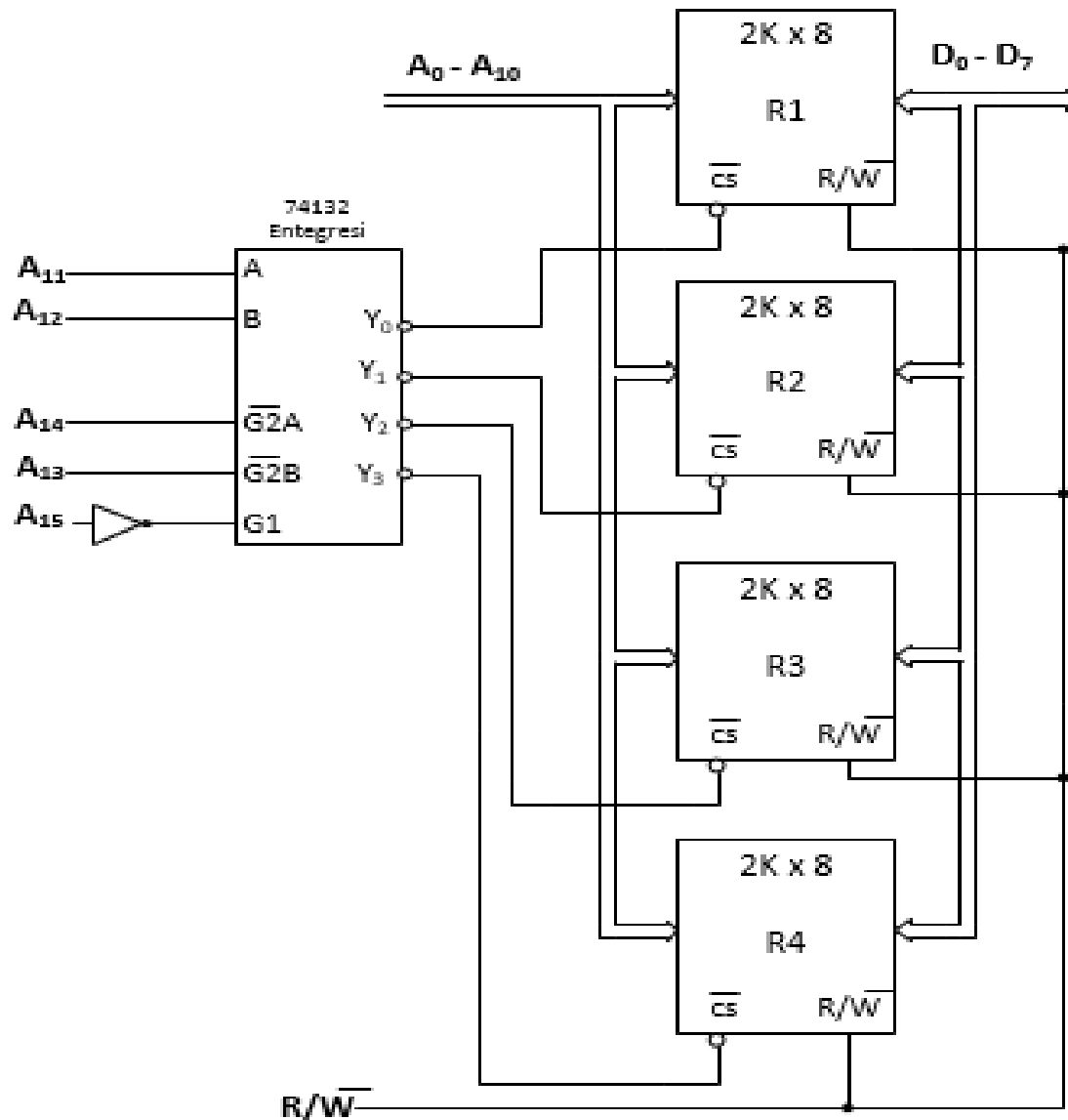
0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	2K R1
0 0 0 0	0 1 1 1	1 1 1 1	1 1 1 1	2K R2
0 0 0 0	1 0 0 0	0 0 0 0	0 0 0 0	
				2K R3
0 0 0 0	1 1 1 1	1 1 1 1	1 1 1 1	
0 0 0 1	0 0 0 0	0 0 0 0	0 0 0 0	2K R4
0 0 0 1	0 1 1 1	1 1 1 1	1 1 1 1	2K R4
0 0 0 1	1 0 0 0	0 0 0 0	0 0 0 0	
				2K R4
0 0 0 1	1 1 1 1	1 1 1 1	1 1 1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

# ÖRNEK 2 (DEVAMI)





## ÖRNEK 2 (DEVAMI)



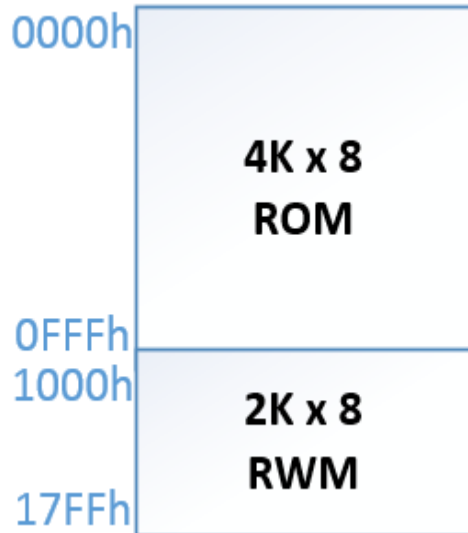
# ÖRNEK 3

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

EPROM 0000h adresinden başlayacak ve RWM ardından devam edecek.

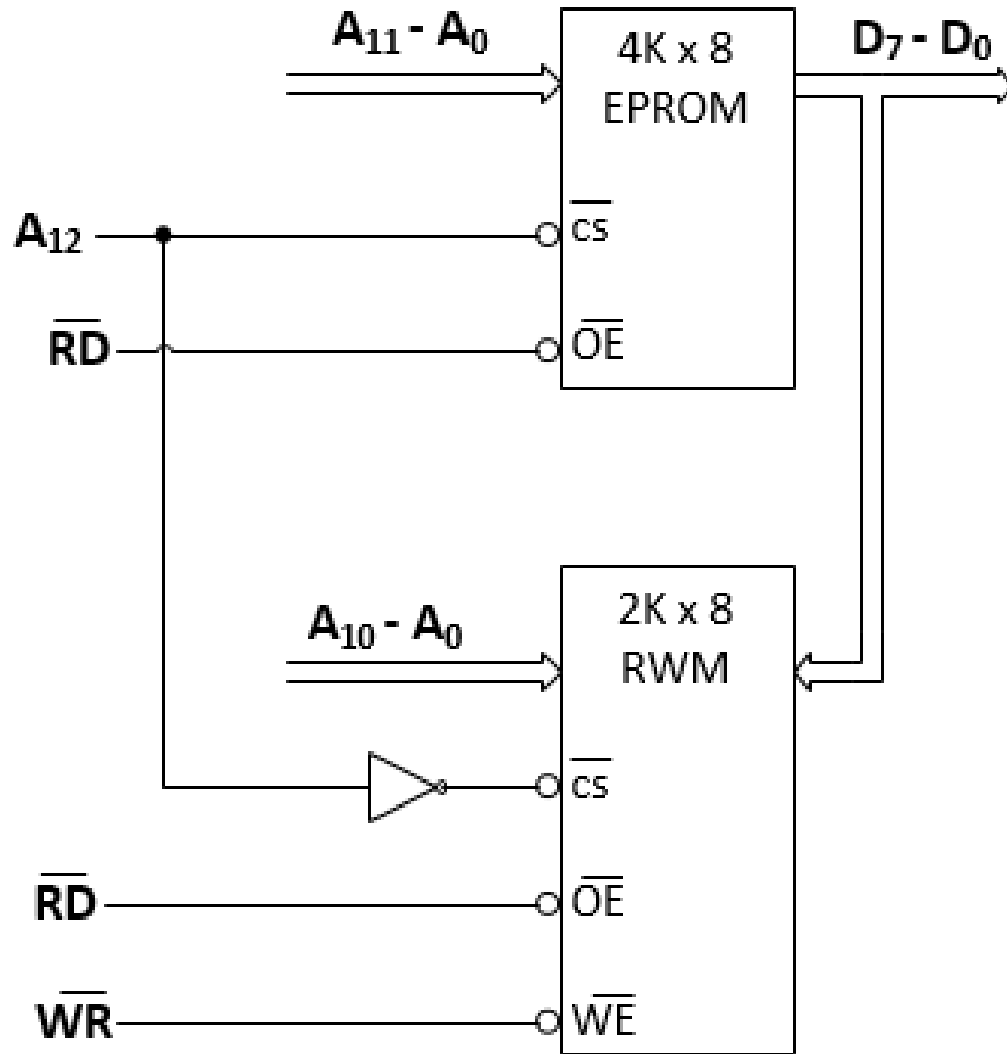
4K -> 12 bit

2K -> 11 bit



				$A_{12}$	$A_{11}$	$A_{10}$	
				↓	↓	↓	
0 0 0 0				0	0	0	0
0 0 0 0				0	0	0	0
0 0 0 0				0	0	0	0
0 0 0 0				0	0	0	0
0 0 0 0				1	1	1	1
0 0 0 1				0	0	0	0
0 0 0 1				0	0	0	0
0 0 0 1				0	1	1	1
0 0 0 1				1	1	1	1
0 0 0 1				1	1	1	1
0 0 0 1				1	1	1	1

## ÖRNEK 3 (DEVAMI)



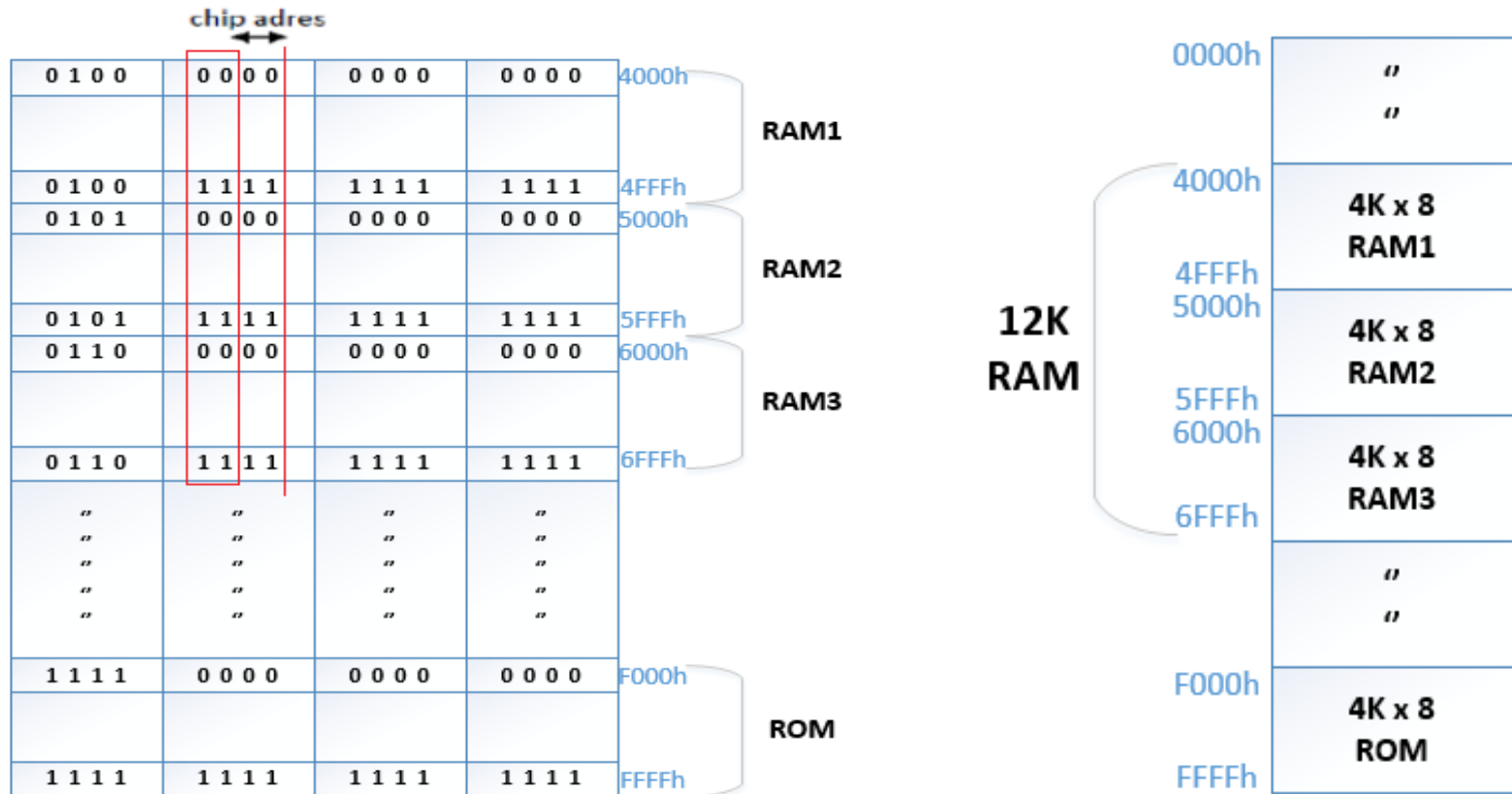
Yansımaz: (Exhaustive/Absolute Decoding) En küçük birime göre

Yansımali (Partial Decoding): Mümkün olan en büyük elemana göre yapılabilir

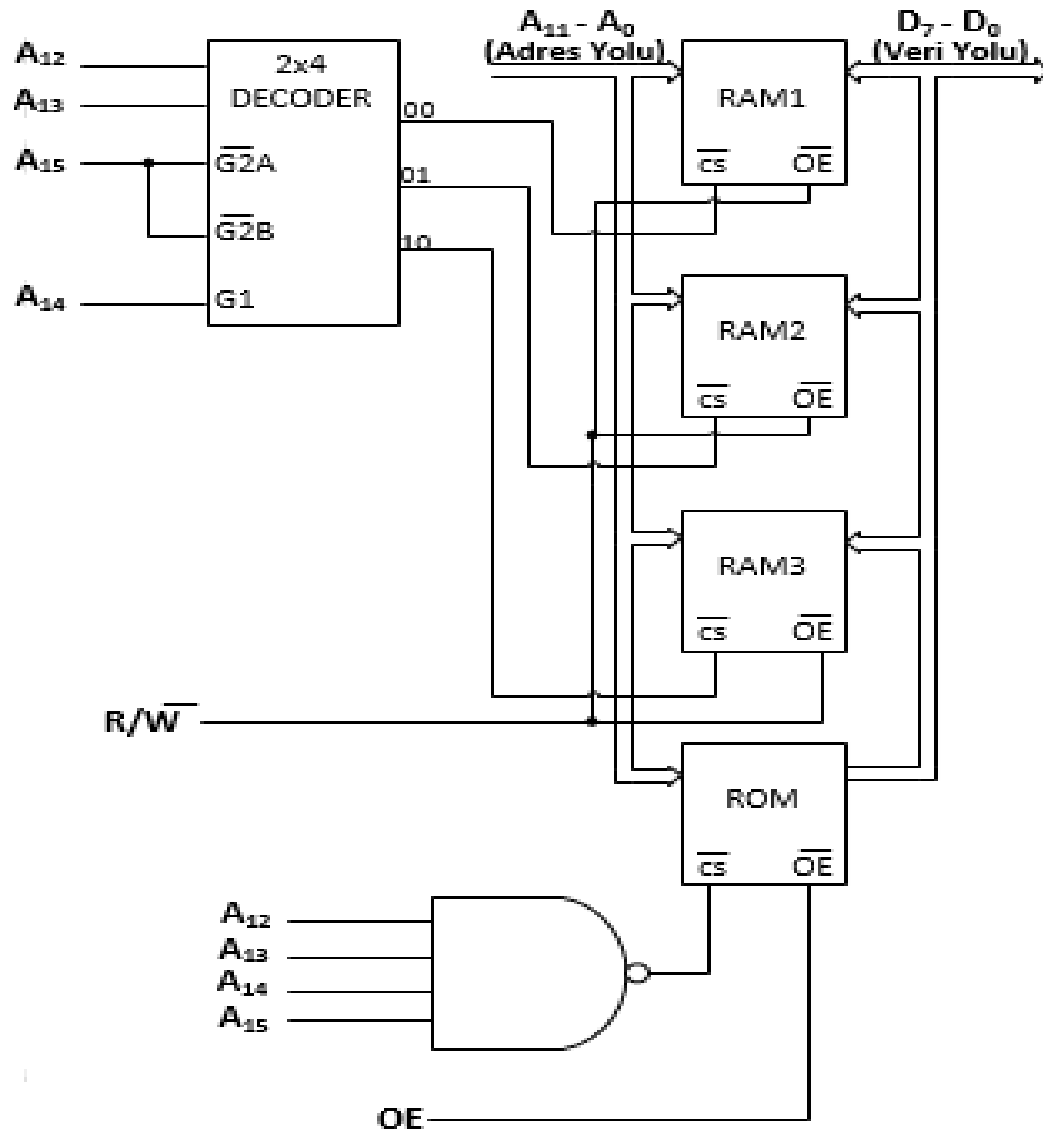
# ÖRNEK 4

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



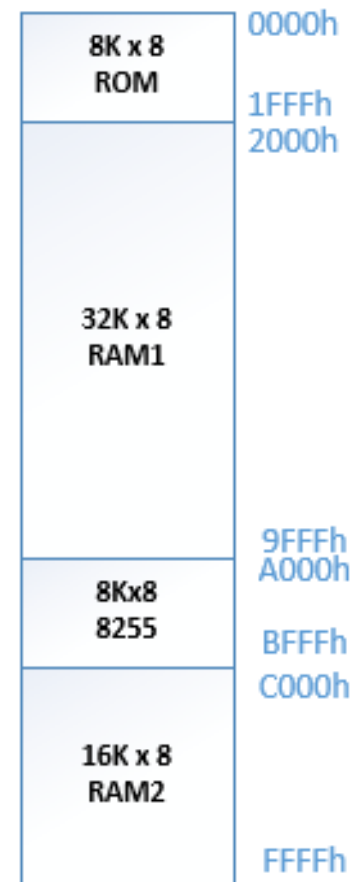
# ÖRNEK 4 (DEVAMI)



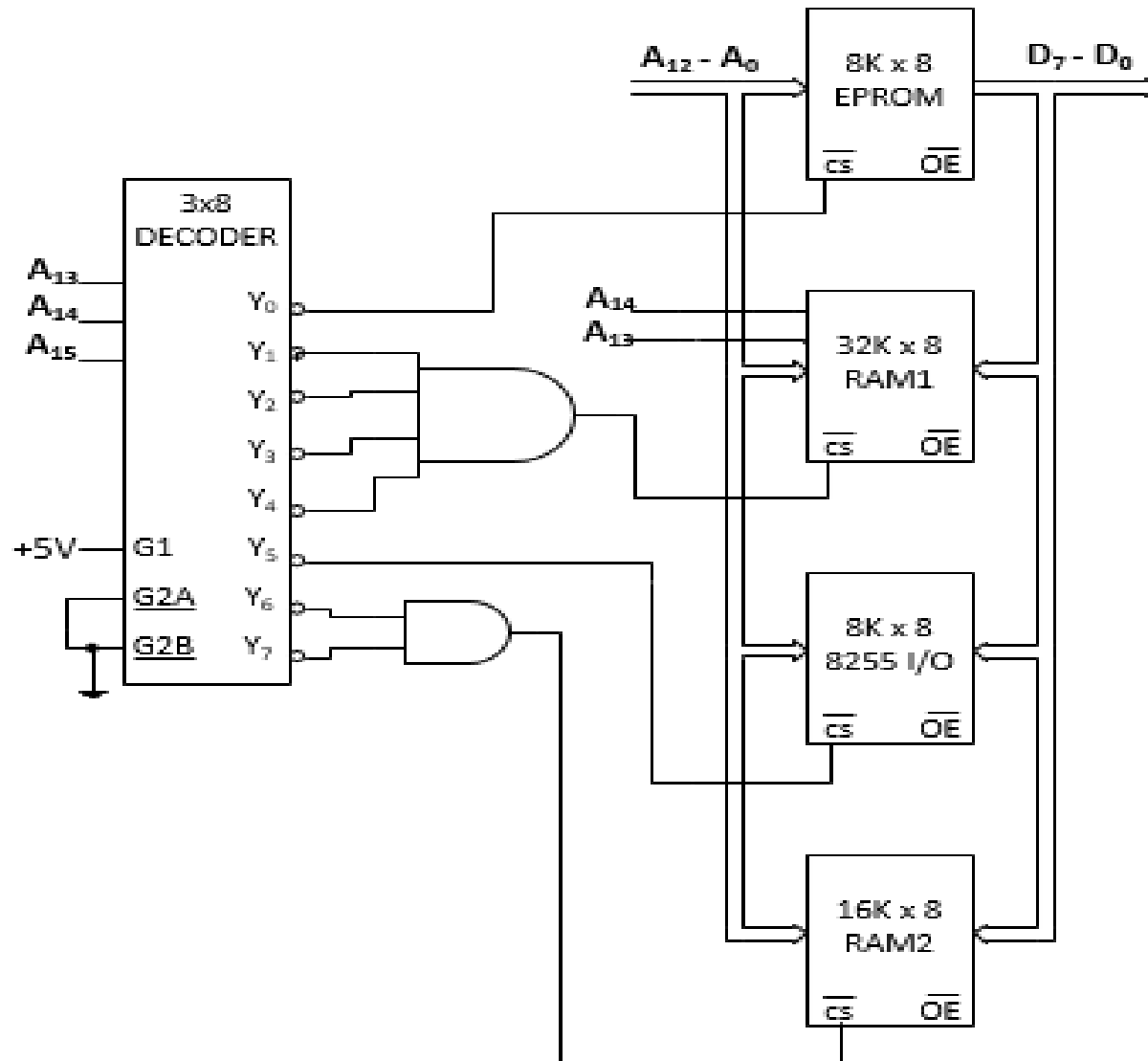
# ÖRNEK 5

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

Elimizde 3x8'lik dekoderler mevcut

Memory Map

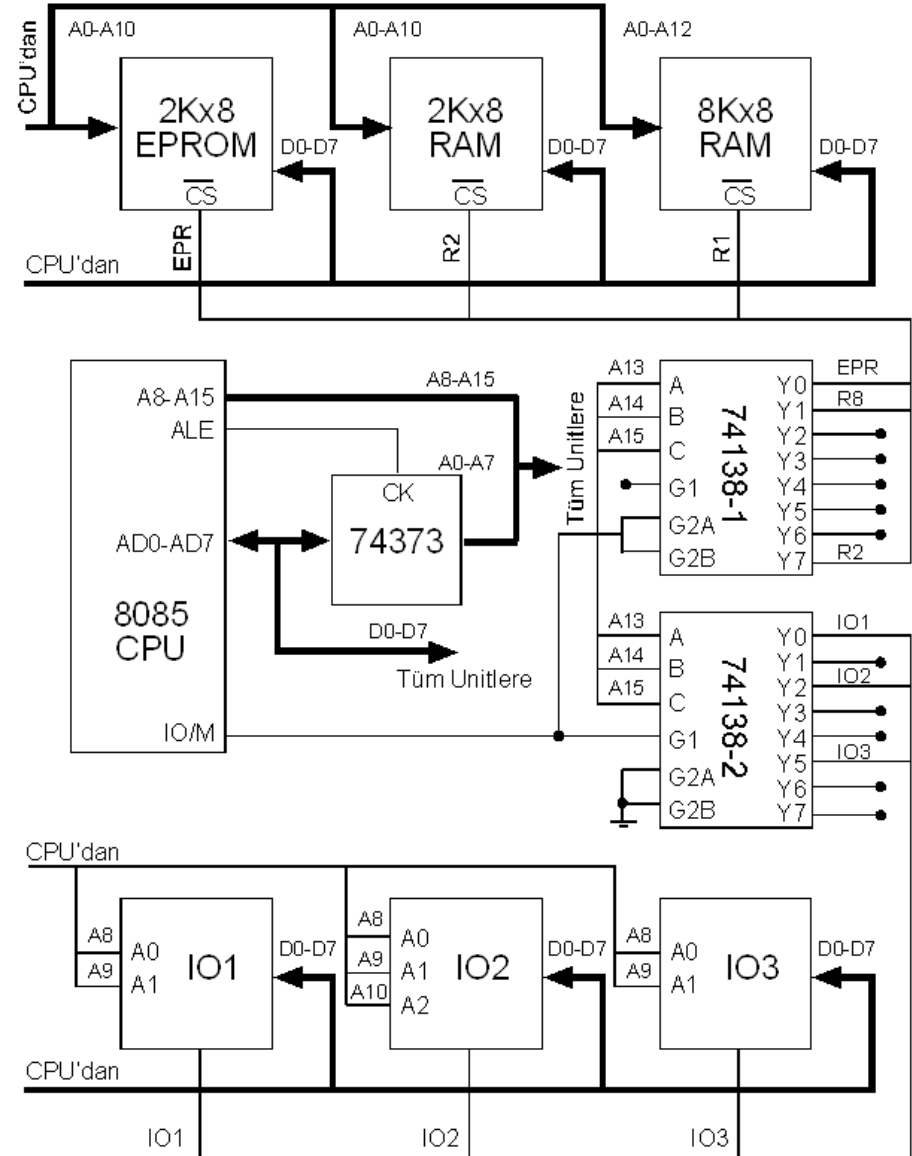
FFFF	2Kx8 RAM
E000	
DFFF	BOŞ
C000	
BFFF	BOŞ
A000	
9FFF	BOŞ
8000	
7FFF	BOŞ
6000	
5FFF	BOŞ
4000	
3FFF	8Kx8 RAM
2000	
1FFF	2Kx8 EPROM
0000	

2K EPROM 0000-07FFh  
8K RAM1 2000-3FFFh  
2K RAM2 E000-E7FFh

IO Map

FF	BOŞ
E0	
DF	BOŞ
C0	
BF	4 Byte IO3
A0	
9F	BOŞ
80	
7F	BOŞ
60	
5F	8 Byte IO2
40	
3F	BOŞ
20	
1F	4 Byte IO1
00	

4B IO1 00h (0000h)  
8B IO2 40h (4000h)  
4B IO3 A0h (A000h)





## Örnek 6

2Kx8 EPROM 0000h adresinde

8Kx8 RWM 2000h adresinde

2Kx8 RAM E000h adresinde buluna hafızaları  
adresleyerek tasarımı gerçekleyiniz. (yansımali ve  
yansimasiz)

## Örnek 7

4Kx8 (8255) 1000h

8Kx8 RAM 2000h

4Kx8 ROM 4000h

16Kx8 ROM2 5000h

8Kx8 RAM2 E000h

4Bx8 IO1 20h

4Bx8 IO2 70h

8Bx8 IO3 A0h

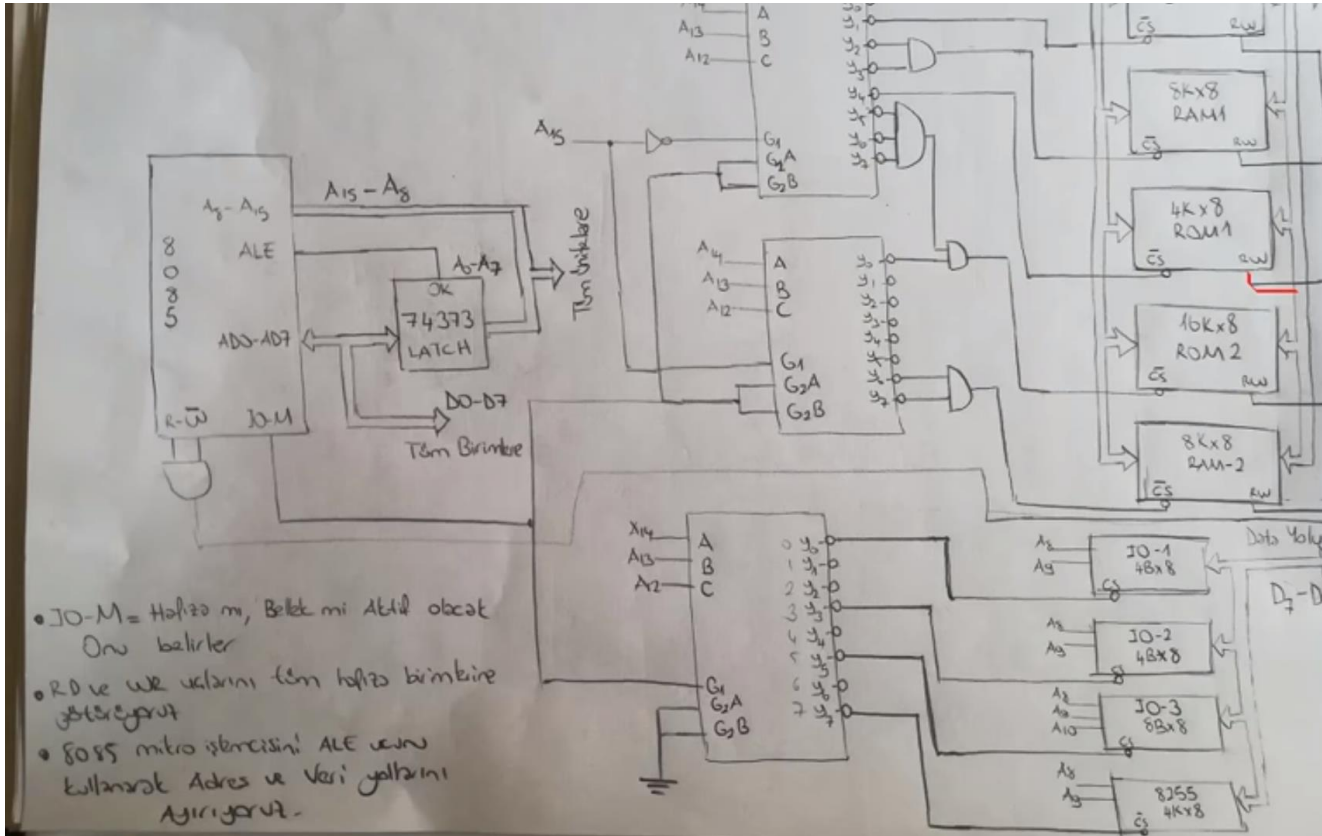
4Bx8 8255 F0h

Bellek birimleri yansımaz, I/O birimleri yansımalı

Elimizde 3x8'lik kod çözücüler (dekoder) var

# Örnek 7

- I/O birimleri için A15-A14-A13 seçim ucu
- Bellek birimleri için A14-A13-A12 dekode giriş uçları, A15 dekode giriş ucu



# Örnek 7

Bellek

8255 4K	0000h	4Kx8	1000h	0001	0000	0000	0000
8255 4Kx8	0FFFh	8255	1FFFh	0001	1111	1111	1111
RAM 8Kx8	2000h	RAM	2FFFh	0010	0000	0000	0000
RAM 8Kx8	3FFFh	RAM	3FFFh	0011	0000	0000	0000
ROM 4Kx8	4000h	ROM	4FFFh	0100	0000	0000	0000
ROM 4Kx8	5000h	ROM	5FFFh	0101	0000	0000	0000
ROM 4Kx8	6000h	ROM	6FFFh	0110	0000	0000	0000
ROM 4Kx8	7FFFh	ROM	7FFFh	0111	0000	0000	0000
ROM 4Kx8	8000h	ROM	8FFFh	1000	1111	1111	1111

4Kx8	1FFFh	1FFFh	0001	1111	1111	1111	1111
RAM	2000h	2000h	0010	0000	0000	0000	0000
RAM	2FFFh	2FFFh	0011	0000	0000	0000	0000
RAM	3000h	3000h	0100	0000	0000	0000	0000
RAM	3FFFh	3FFFh	0101	0000	0000	0000	0000
RAM	4000h	4000h	0110	0000	0000	0000	0000
RAM	4FFFh	4FFFh	0111	0000	0000	0000	0000
RAM	5000h	5000h	1000	1111	1111	1111	1111
RAM	5FFFh	5FFFh	1001	1111	1111	1111	1111
RAM	6000h	6000h	1010	1111	1111	1111	1111
RAM	6FFFh	6FFFh	1011	1111	1111	1111	1111
RAM	7000h	7000h	1100	1111	1111	1111	1111
RAM	7FFFh	7FFFh	1101	1111	1111	1111	1111
RAM	8000h	8000h	1110	1111	1111	1111	1111
RAM	8FFFh	8FFFh	1111	1111	1111	1111	1111
RAM	9000h	9000h	1110	1111	1111	1111	1111
RAM	9FFFh	9FFFh	1111	1111	1111	1111	1111
RAM	A000h	A000h	1110	1111	1111	1111	1111
RAM	AFFFh	AFFFh	1111	1111	1111	1111	1111

## Örnek 7

			A <sub>15</sub> A <sub>14</sub> A <sub>13</sub> A <sub>12</sub>	A <sub>11</sub> A <sub>10</sub> A <sub>9</sub> A <sub>8</sub>	
IO-1 4Bx8	0x20h	J01 4Bx8	0x20h	1 0 0 1 0	0 0 0 0
BOS	0x23h		0x23h	1 0 0 1 0	0 0 1 1
IO-2 4Bx8	0x70h	J02 4Bx8	0x70h	0 1 1 1	0 0 0 0
	0x73h		0x73h	0 1 1 1	0 0 1 1
IO-3 8Bx8	0xA0h	J03 8Bx8	0xA0h	1 0 1 0	0 0 0 0
	0xA7h		0xA7h	1 0 1 0	0 1 1 1
BOS	0xA7h	8255	0xF0h	1 1 1 1	0 0 0 0
8255	0xF0h		0xF3h	1 1 1 1	0 0 1 1
	0xF3h				
BOS	0xFFh				

A<sub>10</sub> → 1010 0000  
A<sub>3</sub> → 1010 0011  
A<sub>4</sub> → 1010 0100  
A<sub>7</sub> → 1010 0111

A<sub>14</sub> A<sub>13</sub> A<sub>12</sub> → IO biriminde programlı  
Seçim ucu olarak kabul edildi. J01 ile J03 seçildi.  
A<sub>15</sub> A<sub>14</sub> A<sub>12</sub>  
Seçici Ucu olarak kabul edildi.  
Geliştirme durumu ortada kaldı.

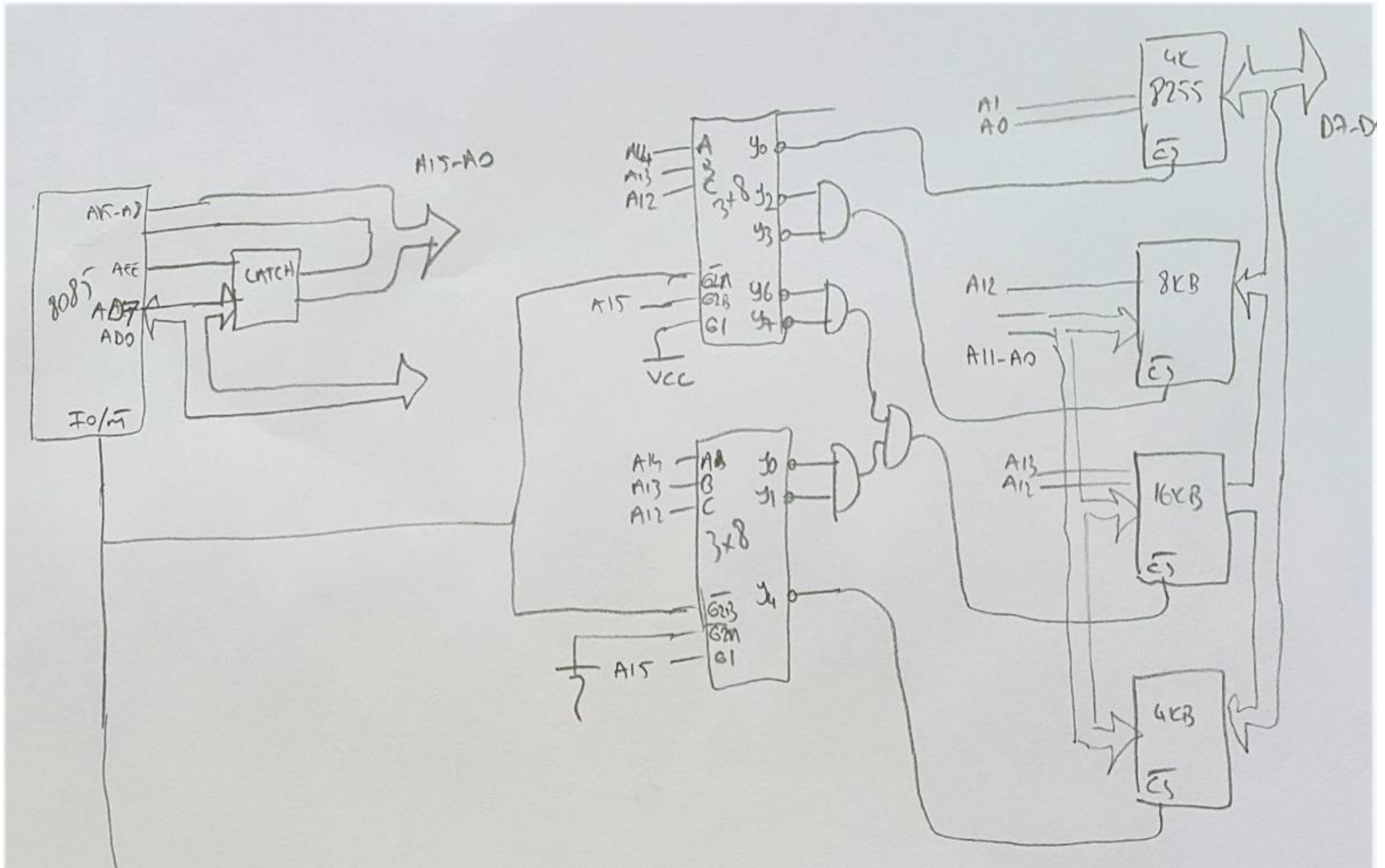
# SORU (Vize)

- 8085 mikroişlemcisine, bellek ve I/O birimlerine sahip bir mikroişlemcili sistemde;
- • Bellek birimleri: 8255 arabirimi (4Kx8 gibi düşünün) 0000h adresinden, 8Kx8'lik RAM'in 2000h adresinden, 16Kx8'lik ROM'un 6000h adresinden, 4Kx8'lik RAM'in C000h adresinden itibaren,
- • I/O birimleri: 4BX8'lik I/O-1 birimi 20h adresinden, 4BX8'lik I/O-2 birimi 70h adresinden, 8BX8'lik I/O-3 birimi A0h adresinden, 8255 arabirimi F0h adresinden itibaren yerleştirilmiştir.
- Bu verilere göre tüm sistem için kontrol devresini tasarlayarak işlemci, adres tutucu, bellek ve I/O'ların bağlantılarını çiziniz.
- (Elimizde kod çözücü olarak 3x8'likler ve mantık kapıları (AND ve OR) mevcuttur. Bellek birimleri için yansımasız, I/O birimleri için yansımali çözüm yapılacaktır.)

# SORU (Vize)

4 B	20	0010	0000	4 KB	8255	0000	0000	0000	0000
	23	0010	0011			0FFF	0000	1111	1111
4 B	70	0111	0000	8 KB		2000	0010	0000	0000
	73	0111	0011			2FFF	0010	1111	1111
						3000	0011	0000	0000
8 B	A0	1010	0000	16 KB		3FFF	0011	1111	1111
	A7	1010	0111						
8255	F0	1111	0000			6FFF	0110	0000	0000
	F3	1111	0011			7FFF	0110	1111	1111
						8000	0111	0000	0000
						8FFF	0111	1111	1111
						9000	1000	0000	0000
						9FFF	1000	1111	1111
							1001	0000	0000
							1001	1111	1111
				4 K		C000	1100	0000	0000
						CFFF	1100	1111	1111

# SORU (Vize)





# SORU (Vize)

