

24. BELLEK ARAYÜZÜ

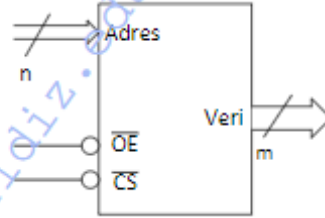
Bu bölümde bellek cihazları tanıtılarak, 8086 ile bellek arayüzlerinin gerçekleştirilmesi detaylı olarak anlatılacaktır.

24.1 Bellek Cihazları

Mikroişlemcili sistemlerde programların saklanması için kalıcı bellek yapısında olan Read Only Memory (ROM) ve türevleri kullanılmaktadır. Çalışma sırasında geçici bellek gereksinimlerinde ise Random Access Memory (RAM) ve türevleri kullanılmaktadır.

24.1.1 Read Only Memory (ROM)

Kalıcı bellek (**non-volatile memory**) elemanları olup enerjileri kesildiğinde de sakladıkları değerler kaybolmaz. ROM genel blok yapısı Şekil 24-1 ile verilmektedir. n adet adres ucuna ve m adet veri ucuna sahip bir ROM $2^n \times m$ kapasitesindedir. Bunun anlamı ROM içerisinde 2^n adet bellek gözü olup bunların her biri m bit saklama kapasitesindedir. ROM için iki ayrı seçim ucu mevcuttur. Bunlardan ilki \overline{CS} ucu olup ROM'un çalışma adres aralığı ile ilgili olarak seçilmesini sağlar. Dolayısıyla \overline{CS} ucu adres çözümleme devresinin bir çıkışına bağlanır. İkinci seçim ucu olan \overline{OE} ise okuma işlemleri ile ilgili seçim ucu olup mikroişlemcinin \overline{RD} ucuna bağlanır.



Şekil 24-1: $2^n \times m$ kapasiteli ROM blok diyagramı

Masked-ROM, PROM (Programmable ROM-programlanabilir ROM), EPROM (Erasable ROM-silinebilir ROM), EEPROM (Electrically Erasable Programmable ROM-elektriksel olarak yazılıp silinebilir ROM) gibi çeşitleri mevcuttur.

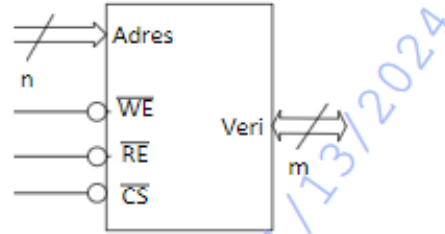
ROM, adres uçlarının seçim ucu olduğu bir kodçözücü yapısı ve kodçözücü ile seçilen bellek gözlerinden oluşmaktadır.

24.1.2 Random Access Memory (RAM)

Geçici bellek (**volatile memory**) elemanları olup enerjileri kesildiğinde sakladıkları değerleri yitirirler. Temel yapıtaşı flip-flop olan SRAM ve temel yapıtaşı transistör ile kapasite olan DRAM şeklinde iki farklı çeşidi mevcuttur.

24.1.2.1 Statik RAM (SRAM)

SRAM genel blok yapısı Şekil 24-2 ile verilmektedir. n adet adres ucuna ve m adet veri ucuna sahip bir SRAM $2^n \times m$ kapasitesindedir. SRAM için üç ayrı seçim ucu mevcuttur. Bunlardan ilki \overline{CS} ucu olup SRAM'ın çalışma adres aralığı ile ilgili olarak seçilmesini sağlar. Dolayısıyla \overline{CS} ucu adres çözümleme devresinin bir çıkışına bağlanır. İkinci seçim ucu olan \overline{RE} ise okuma işlemleri ile ilgili seçim ucu olup mikro-işlemcinin \overline{RD} ucuna bağlanır. Üçüncü seçim ucu olan \overline{WE} de yazma işlemleri ile ilgili seçim ucu olup mikro-işlemcinin \overline{WR} ucuna bağlanır.

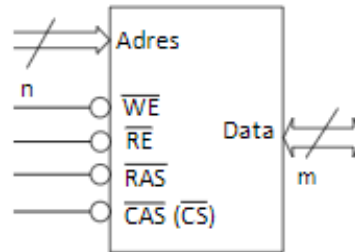


Şekil 24-2: $2^n \times m$ kapasiteli SRAM blok diyagramı

SRAM, adres uçlarının seçim ucu olduğu bir kodçözücü yapısı ve kodçözücü ile seçilen bellek gözlerinden oluşmaktadır.

24.1.2.2 Dinamik RAM (DRAM)

DRAM genel blok yapısı Şekil 24-3 ile verilmektedir. n adet adres ucuna ve m adet veri ucuna sahip bir DRAM $2^n \times m$ kapasitesindedir. DRAM için dört ayrı seçim ucu mevcuttur. Bunlardan ilki olan \overline{RE} ucu okuma işlemleri ile ilgili seçim ucu olup mikro-işlemcinin \overline{RD} ucuna bağlanır. İkinci seçim ucu olan \overline{WE} yazma işlemleri ile ilgili seçim ucu olup mikro-işlemcinin \overline{WR} ucuna bağlanır. Bunlar dışındaki iki seçim ucu ise \overline{RAS} (row address strobe – satır adres seçimi) ve \overline{CAS} (column address strobe – sütun adres seçimi) uçlarıdır. \overline{CAS} seçim ucu aynı zamanda \overline{CS} seçim görevini de yürütür. DRAM, satır ve sütun seçimi için iki ayrı kodçözücü yapısı barındırmaktadır. \overline{RAS} ile satır kodçözücü aktif edilirken, \overline{CAS} ile sütun kodçözücü aktif edilir.



Şekil 24-3: $2^n \times m$ kapasiteli DRAM blok diyagramı

DRAM'de bir bellek gözüne erişilmek istendiğinde öncelikle $\overline{\text{RAS}}$ ucu aktif edilerek adres uçlarındaki değer satır adresi olarak tutulur. Tutulan bu satır adresi kod-çözücü yardımıyla çözümlenerek bir satırın seçilmesini sağlar. Daha sonra öncelikle $\overline{\text{CAS}}$ ucu aktif edilerek adres uçlarındaki değer sütun adresi olarak tutulur. Tutulan bu sütun adresi sütun kodçözücü yardımıyla çözümlenerek bir bir sütunun seçilmesini sağlar. Böylelikle DRAM içindeki satır ve sütun olarak çözümlenmiş bellek gözüne erişilmiş olur. Bu süreçte önce n adet adres ucuyla 2^n satır çözümlenirken, hemen sonra yine n adet adres ucuyla 2^n sütun çözümlenmiş olur. Dolayısıyla DRAM'de n adet ucuyla $2^n \times 2^n = 2^{2n}$ bellek gözü adreslenebilir.

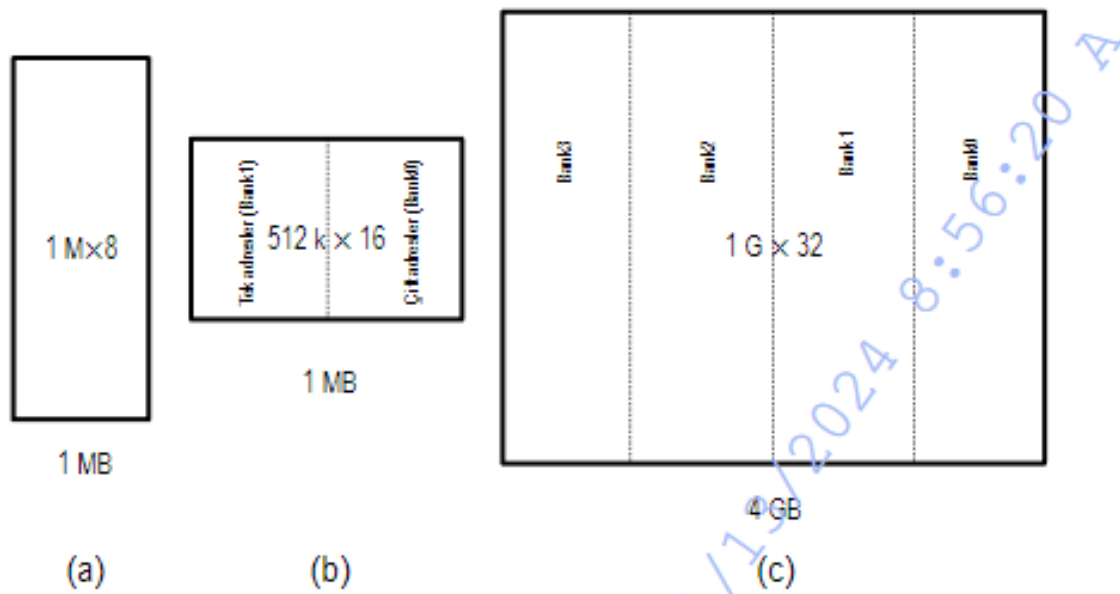
24.2 8086 Adres Uzaı ve Bellek Arayüzü

8086, 20 adet adres ucu ve 16 adet veri ucuna sahiptir. Tasarımı gereği 8086 çift adresten sekiz bit'lik, tek adresten sekiz bit'lik ve çift adresten 16 bit'lik okuma yazmayı tek çevrimde yapabilir. Bellek elemanlarının 8 bit'lik birimlerden oluşturulduğu durum da göz önüne alınarak toplamda $2^{20} = 1\text{M}$ bellek birimi adreslenebilir ve bunların her biri 8 bit veri tutabildiğinden dolayı, 8086 adres uzayının 1MB kapasitesinde olduğu söylenebilir. Benzer bir durum (toplam adres uzaı kapasitesi 1MB) 20 adres ucuna ve 8 veri ucuna sahip 8088 mikroişlemcisi için de geçerlidir. Fakat 8088 sadece; çift adresten sekiz bit ve tek adresten sekiz bit okuma yazma işlemlerini donanım gereği tek çevrimde destekleyebilmektedir.

Mikroişlemcinin tek seferde desteklediği okuma/yazma bit genişliği ile uyumlu olacak şekilde bellek ve G/Ç birimlerinin de uyumlu şekilde yerleştirilmesi gerekmektedir. Bellek ve G/Ç birimlerinin mikroişlemci ile uyumlu formatta yerleştirilmesi uygun bellek arayüzü olarak isimlendirilmektedir. Bu durumda 8088 ve 8086 mikroişlemciler için toplam adres uzaı kapasiteleri aynı olmasına rağmen uygun bellek arayüzleri sırasıyla Şekil 24-4a ve Şekil 24-4b ile verildiği gibidir. 8088 için uygun bellek arayüzü 1 Mx8, 8086 için ise uygun bellek arayüzü 512 kx16'dır.

32 bit adres ve veri yolu genişliğine sahip 80386 için de uygun bellek arayüzü Şekil 24-4c ile verilmiştir. 80386 için uygun bellek arayüzü 1 Gx32 şeklinde oluşmaktadır.

8088 tüm adres uzayını bir blok olarak kullanabilmektedir. 8086 da ise adres uzaı çift ve tek adresli bloklar (bank) halinde görülmektedir. 8086'da çift adreslerin (Bank0) seçimi A0 ($\overline{\text{BLE}}$ – bus low enable), tek adreslerin (Bank1) seçimi ise $\overline{\text{BHE}}$ – bus high enable - ile sağlanır. 80386 için benzer şekilde bank seçimi $\overline{\text{BE0}}$, $\overline{\text{BE1}}$, $\overline{\text{BE2}}$ ve $\overline{\text{BE3}}$ - byte enable 0...3 – uçları ile sağlanır.



Şekil 24-4: a) 8088 için uygun bellek arayüzü, b) 8068 için uygun bellek arayüzü, c) 80386 için uygun bellek arayüzü

8086, çift adresten 16 bit'lik komut yürüttüğünde, $\overline{BHE} = 0$ ve $\overline{BLE} = 0$ değerlerini oluşturarak her iki bankın da aktif olmasını sağlar. 8086 çift adresten sekiz bit'lik komut yürüttüğünde ise, $\overline{BHE} = 1$ ve $\overline{BLE} = 0$ değerlerini oluşturarak sadece Bank0'ın aktif olmasını sağlar. Son olarak tek adresten sekiz bit'lik komut yürütüldüğünde ise, 8086, $\overline{BHE} = 0$ ve $\overline{BLE} = 1$ değerlerini oluşturarak sadece Bank1'in aktif olmasını sağlar. 8086 ile tek adresten 16 bit'lik işlemler ise önce tek adresten sekiz bit'lik ve bir sonraki çift adresten tekrar sekiz bit'lik işlem olarak iki çevrimde tamamlanabilir. Bu durum Tablo 24-1 ile özetlenmiştir.

Tablo 24-1: 8086 Bank seçimi

	Örnek Komut	\overline{BHE}	\overline{BLE}	Seçilen Bank	Çevrim Sayısı
Çift Adresten 8 Bit'lik Komut	MOV AL, [100H]	1	0	Bank0	1
Tek Adresten 8 Bit'lik Komut	MOV AL, [101H]	0	1	Bank1	1
Çift Adresten 16 Bit'lik Komut	MOV AX, [100H]	0	0	Bank1 ve Bank0	1
Tek Adresten 16 Bit'lik Komut	MOV AX, [101H]	0	1	Önce Bank1	2
		1	0	Sonra Bank0	

Benzer durum 80386'da da söz konusudur. 80386, son iki bit'i 00 olan adreslerden 32 bit, son bit'i 0 olan adreslerden 16 bit ve tüm adreslerden sekiz bit okuma/yazma işlemini tek çevrimde gerçekleştirebilir. Bunun dışında hizalı olmayan adreslerden yapılan 16 ve 32 bit'lik işlemler ise iki veya üç çevrimde tamamlanır.

Tablo 24-2: 80386 Bank seçimi

Adres Hizası	Komut	A1	A0	Örnek Komut	$\overline{\text{BE}}3$	$\overline{\text{BE}}2$	$\overline{\text{BE}}1$	$\overline{\text{BE}}0$	Seçilen Bank
Hızlı Adresler (Çevrim Sayısı = 1)	8 Bit'lik Komut	0	0	MOV BL, [100H]	1	1	1	0	Bank0
		0	1	MOV BL, [101H]	1	1	0	1	Bank1
		1	0	MOV BL, [102H]	1	0	1	1	Bank2
		1	1	MOV BL, [103H]	0	1	1	1	Bank3
	16 Bit'lik Komut	0	0	MOV BX, [100H]	1	1	0	0	Bank1 ve Bank0
		1	0	MOV BX, [102H]	0	0	1	1	Bank3 ve Bank2
	32 Bit'lik Komut	0	0	MOV EBX, [100H]	0	0	0	0	Bank3, Bank2, Bank1 ve Bank0
Hızlı Olmayan Adresler	16 Bit'lik Komut	1	1	MOV BX, [103H]	0	1	1	1	Önce Bank3
					1	1	1	0	Sonra Bank0
	32 Bit'lik Komut	1	0	MOV EBX, [102H]	0	0	1	1	Önce Bank3 ve Bank2
					1	1	0	0	Sonra Bank1 ve Bank0

24.3 Bellek Birimleri için Adres Çözümleme

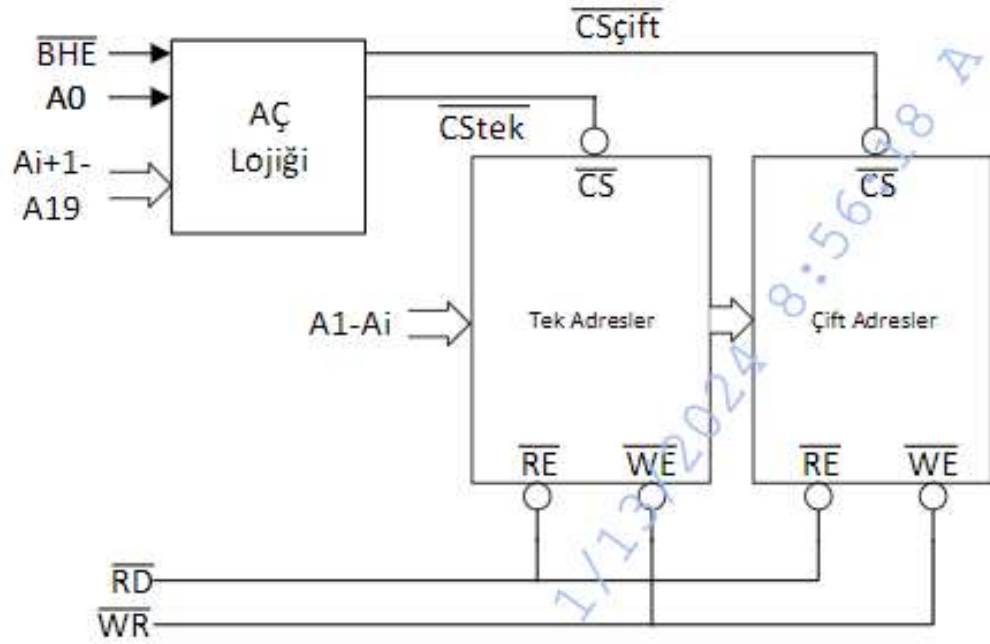
Bellek ve G/Ç birimleri ortak veri ve adres yollarını kullanır. Bir seferde mikroişlemci ile veri iletişimde olan tek bir çevre birimi olmasını sağlamak amacıyla **adres çözümleme** kullanılır. Adres çözümleme ile bellek ve G/Ç birimleri sadece belirli adres aralıklarına yerleştirilmiş olur.

24.3.1 Çift ve Tek Banklar için Ayrı Yetkilendirme ile Adres Çözümleme

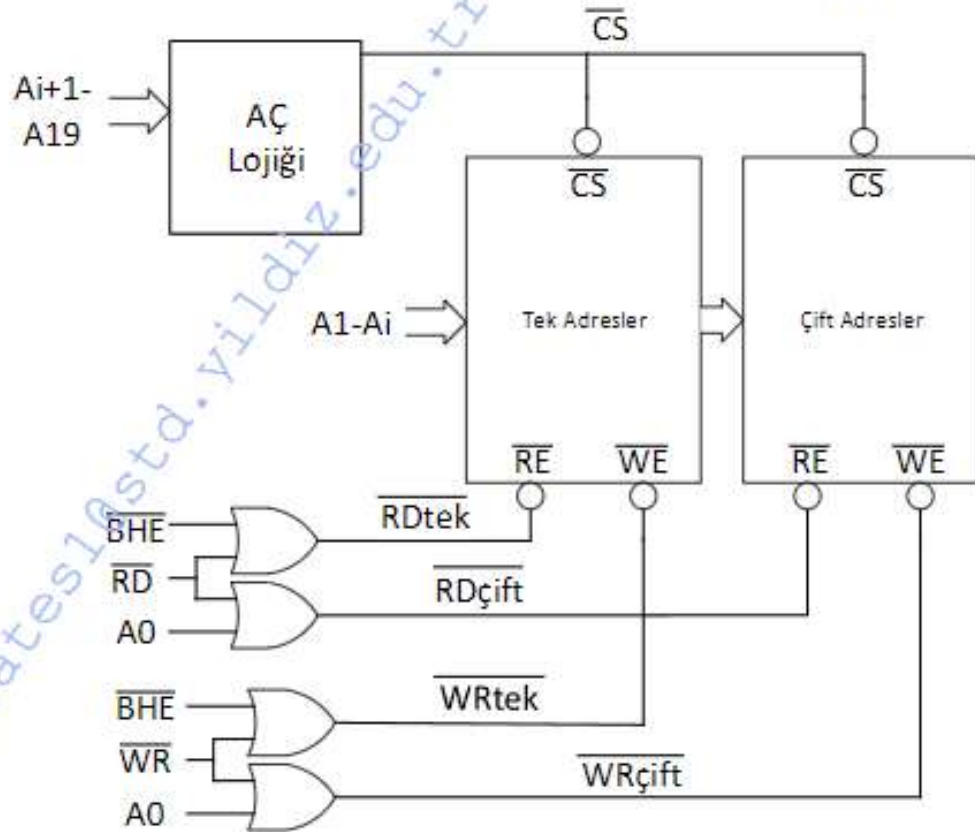
Verilen adres aralığına giren çift ve tek adres değerleri için ayrı ayrı $\overline{\text{CS}}$ üretilir.

24.3.2 Çift ve Tek Banklar için Ayrı Okuma/Yazma Sinyali ile Adres Çözümleme

Verilen adres aralığı için tek $\overline{\text{CS}}$ üretilir, çift ve tek adres değerleri için ayrı okuma/yazma işaretleri üretilir.



Şekil 24-5: Çift ve tek banklar için ayrı yetkilendirme ile adres çözümleme



Şekil 24-6: Çift ve tek banklar için ayrı okuma/yazma sinyali ile adres çözümleme

24.4 Bellek Elemanlarında Birlikte Kullanım ile Kapasite Genişletme

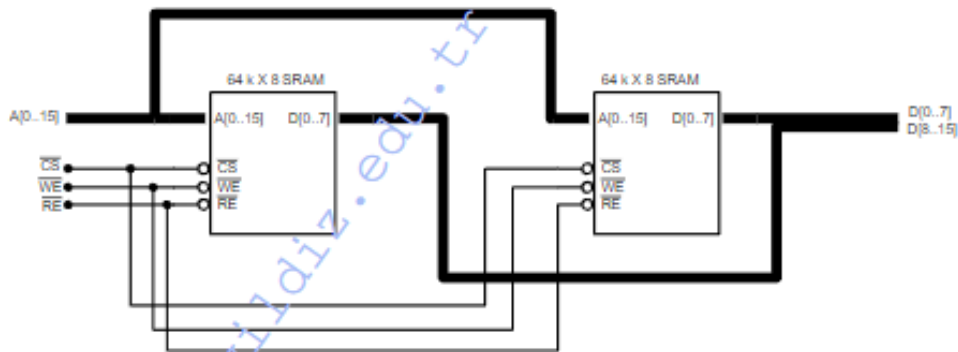
Düşük kapasiteli bellek birimleri bir arada kullanılarak kapasite genişlemesi sağlanabilir. Kapasite genişlemesi, aynı anda birden fazla bellek elemanının seçilmesi ile veri yolları bir arada kullanılarak yapılıyorsa bu durumda veri yolu genişletme olarak isimlendirilir. Benzer şekilde, farklı adres aralıkları için farklı bellek elemanlarının ortak veri yolu kullanmasıyla kapasite genişlemesi yapılıyorsa bu durum adres genişletme olarak isimlendirilir. Son olarak, bellek elemanlarının bir arada kullanılmasıyla hem veri yolu hem de adres birlikte genişletilebilir.

24.4.1 Veri yolu Genişletme

Örnek 24-1

64 k×8 SRAM modülleri kullanarak 64 k×16 RAM bloğu oluşturunuz.

64 k×8 SRAM modülünde veri yolu genişliği sekiz bit, adres yolu genişliği ise $\log_2 64 k = 16$ bit olarak bulunabilir. Elde edilmesi istenen RAM bloğu ise 16 bit veri yolu genişliğindedir.



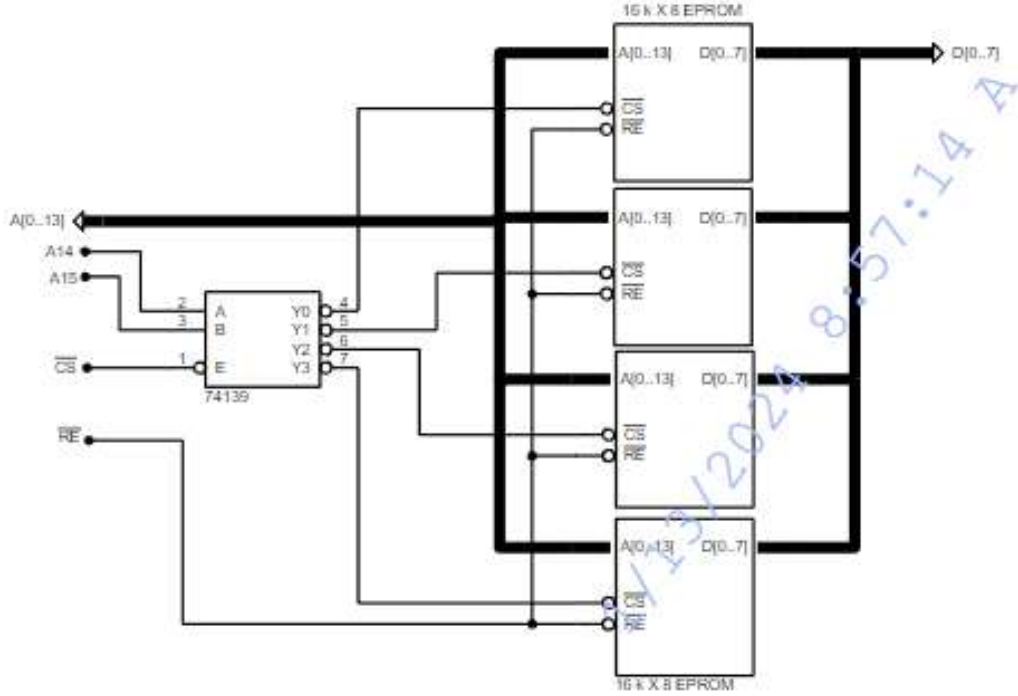
Şekil 24-7: Örnek 24-1 için gerekli bağlantı şekli

24.4.2 Adres Genişletme

Örnek 24-2

16 k×8 EPROM modülleri ve kodçözücü kullanarak 64 k×8 ROM bloğu oluşturunuz.

16 k×8 EPROM modülünde veri yolu genişliği sekiz bit, adres yolu genişliği ise $\log_2 16 k = 14$ bit olarak bulunabilir. Elde edilmesi istenen ROM bloğu ise $\log_2 64 k = 16$ bit adres yolu genişliğindedir. Toplamda kullanılması gereken EPROM modül sayısı $64 k / 16 k = 4$ olarak bulunur.



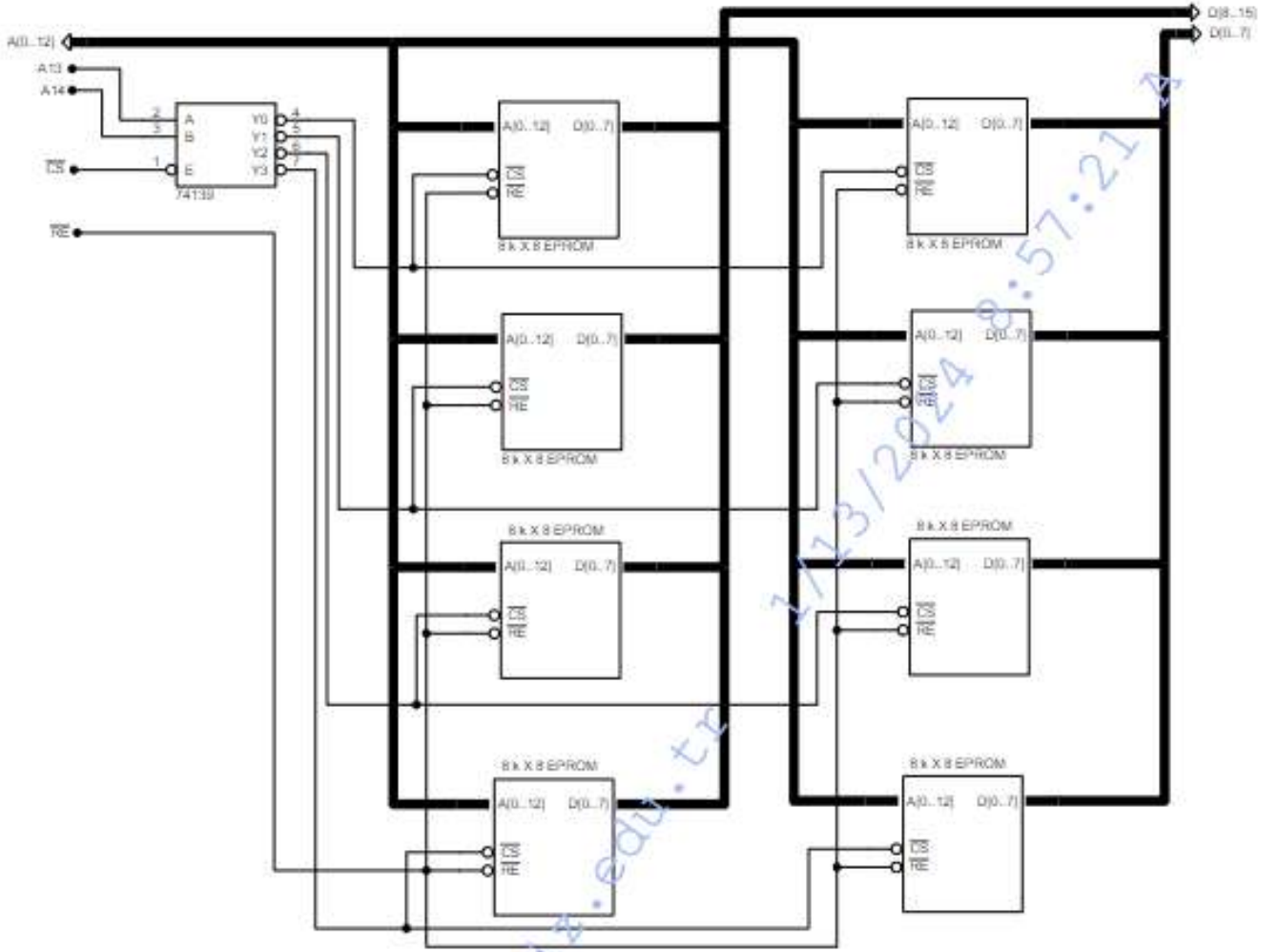
Şekil 24-8: Örnek 24-2 için gerekli bağlantı şekli

24.4.3 Veri yolu ve Adres Genişletme

Örnek 24-3

8 kx8 EPROM modülleri ve kodçözücü kullanarak 32 kx16 ROM bloğu oluşturunuz.

8 kx8 EPROM modülünde veri yolu genişliği sekiz bit, adres yolu genişliği ise $\log_2 8$ k = 13 bit olarak bulunabilir. Elde edilmesi istenen ROM bloğu ise $\log_2 32$ k = 15 bit adres yolu genişliğindedir. Toplamda kullanılması gereken EPROM modül sayısı $32 \text{ k} \times 16 / 8 \text{ k} \times 8 = 8$ olarak bulunur.



Şekil 24-9: Örnek 24-3 için gerekli bağlantı şekli