



Computer Organization

“Memory Organization: Addressing and Mapping”

Dr. Cahit Karakuş
Esenyurt Üniversitesi

İçerik

- Allocating memory cells for variables
- CPU
- Memory
- Virtual Memory
- Memory Types
- Memory Mapping
- Memory Addressing
- Assembly Instructions for Memory Access Addressing
- Memory Access Addressing Applications

Veri Tipleri

Veri

- Veri Toplama: Algılayıcılar, detektörler, ölçerler (Analog Sinyal)
- Veri depolama: Sunucular
- Veri Tabanı Yönetimi, Veri Madenciliği
- Veri tipleri, değişkenler, Veri hazırlama, veri içeriği yönetimi

Variable Declaration

- **Each variable has a type and assigned a memory address.**
- **Data-defining pseudo-ops**
 - DB define byte
 - DW define word
 - DD define double word (two consecutive words)
 - DQ define quad word (four consecutive words)
 - DT define ten bytes (five consecutive words)
- **Each pseudo-op can be used to define one or more data items of given type.**

Başlangıç durumunda katsayıların, değişkenlerin, dizilerin boyutları tanımlanmak zorundadır. Çünkü, belleğin ve programdaki değişkenlerin ve verilerin bellekte kaplayacağı alanların boyutlarının yönetilmesi gerekmektedir.

Byte Variables

- **Assembler directive format defining a byte variable**
 - name DB initial value
 - a question mark (“?”) place in initial value leaves variable uninitialized
- **I DB 4** define variable I with initial value 4
- **J DB ?** Define variable J with uninitialized value
- **Name DB “Course”** allocate 6 bytes for Name
- **K DB 5, 3, -1** allocates 3 bytes



Byte olarak sayı sisteminde:

- Ondalık sayı sisteminde: 0 ... 255
- İkili sayı sisteminde (binary): (0000 0000)b, (0000 0001)b, ... , (1111 1111)
- Hex sayı sisteminde: (00)h, (01)h, ..., (FF)h
- İşretli sayı sisteminde (-/+), en anlamlı bit işaret bit olmaktadır.
 - 0,1,2, ..., 127, (00)h ..., (7F)h
 - 1, ..., -128, (FF)h, ...(80)h

isim DB “Cahit Karakuş”

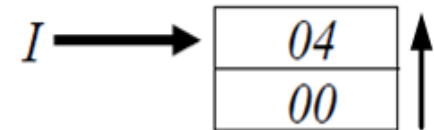
isim, adlı bir dizi tanılanır. Dizinin elemanlar 13 byte’lık bellek alanı işgal eder.

Word Variables

■ Assembler directive format defining a word variable

- name DW initial value

■ I DW 4



■ J DW -2



■ K DW 1ABCh



■ L DW "01"



Değişken tanımında kaç byte? Define Word(DW)

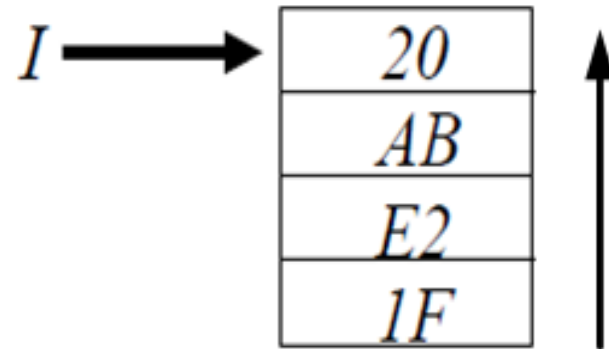
- DW: 16 bit. 2 byte, 4 adet 4 bit.
- I DW 4: (0004)h
- J DW -2: (fffe)h
- KDW 1ABCh: (1ABC)h
- String yapılarında, ASCII karakterleri tanımlanırsa, her bir karakter 16 bit ile temsil edilir.

Double Word Variables

- **Assembler directive format defining a word variable**

- name DD initial value

- **I DD 1FE2AB20H**



- **J DD -4**



Intrinsic data types

- **BYTE, SBYTE**
 - 8-bit unsigned integer; 8-bit signed integer
- **WORD, SWORD**
 - 16-bit unsigned & signed integer
- **DWORD, SDWORD**
 - 32-bit unsigned & signed integer
- **QWORD**
 - 64-bit integer
- **TBYTE**
 - 80-bit integer

Negatif sayılar

- Byte: 8 bit data tanımlar; 8 bitlik bellek gözünü işaret eder.

- Örnek db -4

$$(4)d = (0000\ 0100)b$$

$$(-4)d = \text{Binary } (4) \text{ Tersisi } +1 :$$

$$(-4)d = 1111\ 1011 + 1 = 1111\ 1100 = \text{FC}$$

- Örnek dW -4

$$(4)d = 0000\ 0000\ 0000\ 0100$$

$$(-4)d = \text{Word}(4) \text{ Tersisi } +1$$

$$(-4)d = 1111\ 1111\ 1111\ 1011 + 1 = 1111\ 1111\ 1111\ 1100 = \text{FFFC}$$

Değişkenlerin Kayıt Edildiği Belleklerin Fiziksel Adreslerinin Hesaplanması

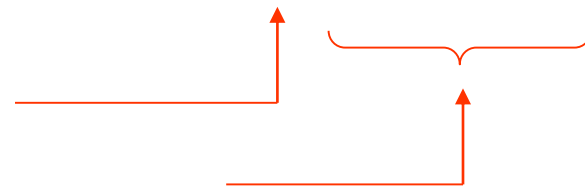
- Örnek: K değişkenin yazılı olduğu belleğin segment adresi (D000)h ve kaynak indis adresi ise (1240)h ve K DB 0CBh ile tanımlı ise
- K değişkenin veri tipi nedir? Byte, işaretsiz
- K değişkenin bellek hücresindeki değerini bit olarak yazınız.
 $0CB)h=(1100\ 1011)b$
- Değişkenler hangi belleğe yazılır. Segment adresi hangi register'ın içeriğindedir. Değeri nedir? Değişkenler RAM belleğe yazılır. Segment adresinin değeri ya Data Segment ya da ekstra segment register içeriğinde kayıtlıdır. Belleğin başlangıç adresini tanımlar. Data segment, 16 bittir. DS:(D000)h
- K değişkenin yazılı olduğu bellek hücresinin fiziksel adresini hesaplayınız.
16 bit DS içeriğinin sağ tarafına 4bit 0 ekleyerek 20 bit lik fiziksel segment adresi hesaplanır.
 $DS: (D000\mathbf{0})h=(1101\ 0000\ 0000\ 0000\ \mathbf{0000})b$
K değişkenin yazılı olduğu bellek gözünün fiziksel adres= (D0000)h + (1240)h=D1240)h

Taşma (Overflow)

- Taşma, toplamdaki bit sayısı, toplanan ve artırılan bitlerin sayısını aştığında meydana gelir.
- Yalnızca her iki sayı da pozitif olduğunda veya her iki sayı da negatif olduğunda oluşur

$$\begin{array}{r} 01111101 \\ + 00111010 \\ \hline 10110111 \end{array} \qquad \begin{array}{r} 126 \\ + 58 \\ \hline 183 \end{array}$$

Sign Incorrect
Magnitude Incorrect



Bit Manipülasyonu

Bit'lerin Aritmetik Toplamı:

$$0+0=0$$

$$0+1=1$$

$$1+0=1$$

$$1+1=10 \text{ (0, elde var 1)}$$

$$1+1+1=11 \text{ (1, elde var 1)}$$

Bit'lerin mantıksal işlemi:

Bool Cebir geçerlidir. Or Kapısı:

$$1+1=1$$

$$0+0=0$$

$$0+1=1$$

$$1+0=1$$

Bit'lerin mantıksal işlemi:

Bool Cebir geçerlidir. And Kapısı:

$$1*1=1$$

$$0*0=0$$

$$0*1=0$$

$$1*0=0$$

Bit'lerin mantıksal işlemi:

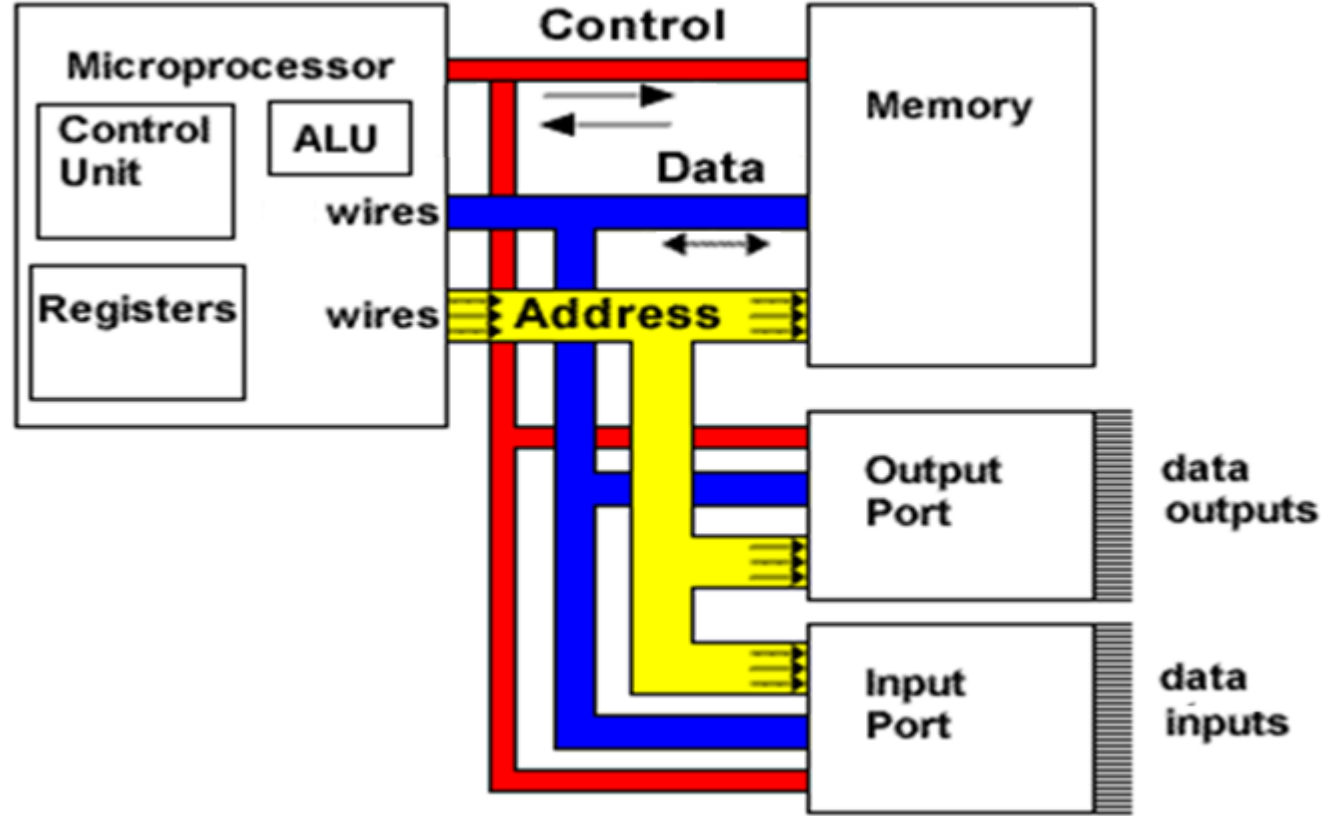
Bool Cebir geçerlidir. Not Kapısı:

$$1'=0$$

$$0'=1$$

CPU SEGMENT ADRESLEME

Bilgisayar Sistemi



CPU'nun Bileşenleri:

- Registers: Genel, Segment, Flags
- Cache (L1)
- Control Unit
- ALU
- Sistem Bus
- Clock and Timing Sinals
- Fiziksel adres üretme, bellek yönetim sistemi

Bir bilgisayar sisteminin temel bileşenleri:

- CPU – Central Processing Unit (Mekezi İşlemci Birimi - Mikroişlemci)
- Ana Bellekler (Ram, Rom)
- Input and output units (I/O)
- System Bus: data bus, address bus and control bus.
- Clock and Timing Signals

Bellek Tanımları

- **Bellek:** Yazılan ya da okunan verilerin belleklerdeki depolandığı ücrelerinin koleksiyonudur.
- **Bellek Mimarisi:** Bir bellekteki veriye nasıl erişileceğinin açıklandığı temel mimari yapıdır.
- **Bellek Organizasyonu:** Bir belleğin gözünde bulunan veriye nasıl erişileceğini tanımlayan temel adreslendirmenin yapılandırılmasıdır .
- **Bellek Haritalama (Mapping):** Belleklerin nasıl yerleştirildiğini gösteren bir veri yapısı gösterimidir.
- **Bellek Adresleme:** Yazılım ve donanım tarafından kullanılan belirli bir bellek konumuna bir referanstır. Her hafıza konumu, bir kod olan fiziksel bir adrese sahiptir.
- **Rasgele Erişimli Bellek (RAM):** Verilerin, seçilen belirli hücreye bağlı olmayan bir zamanda herhangi bir hücreye (veya hücre koleksiyonuna) veya bu hücreden aktarılacağı şekilde düzenlenmiş bir bellek.
- **Bellek Adresi:** Belirli bir bellek ögesini (veya ögeler koleksiyonunu) tanımlayan bir bit vektörü.
- **Çoklu programlama:** CPU'nun bir dizi bağımsız programı aynı anda işlemesini sağlar
- **Bellek Yönetim Sistemi:** yardımcı bellek ile ana bellek arasındaki bilgi akışını denetler

Belleklerde Transistör Sayısı

- Belleklerdeki hücrelerde, veriler, bit (0/1) olarak transistörler üzerinde elektriksel sinyal olarak saklanır.
- Bir Bit SRAM hücresinde 6 transistör bulunur.
- Bir Bit DRAM hücresinde 1 Transistör + 1 kapasitör bulunur.

Örnek: 1TByte DRAM bellekteki transistör sayısı nedir?

- $1\text{TByte} = 2^{40} \text{ Byte} = 2^{40} * 2^3 \text{ bit} = 2^{43} \text{ bit}$ (1 Byte=8 bit= 2^3 bit)
- 1 DRAM hücresinde 1 adet transistör + 1 adet kapasitör var ise
- 1 Tbyte bellekte toplam 2^{43} adet transistör, 2^{43} adet kapasitör bulunur.

Yukarıdaki örnek SRAM olsaydı, $6 * 2^{43}$ adet transistör kullanılırdı.

Memory Hierarchy

– Memory hierarchy in a computer system :

- **Registers in CPU**
- **Cache Memory** : special very-high-speed memory to increase the processing speed (Cache RAM)
- **Main Memory** : memory unit that communicates directly with the CPU (RAM, ROM)
- **Auxiliary Memory** : device that provide backup storage (Disk Drives)

Memory sizes

- We refer to this as a $2^k \times n$ memory.
 - There are k *address lines*, which can specify one of 2^k addresses.
 - Each address contains an n -bit word.

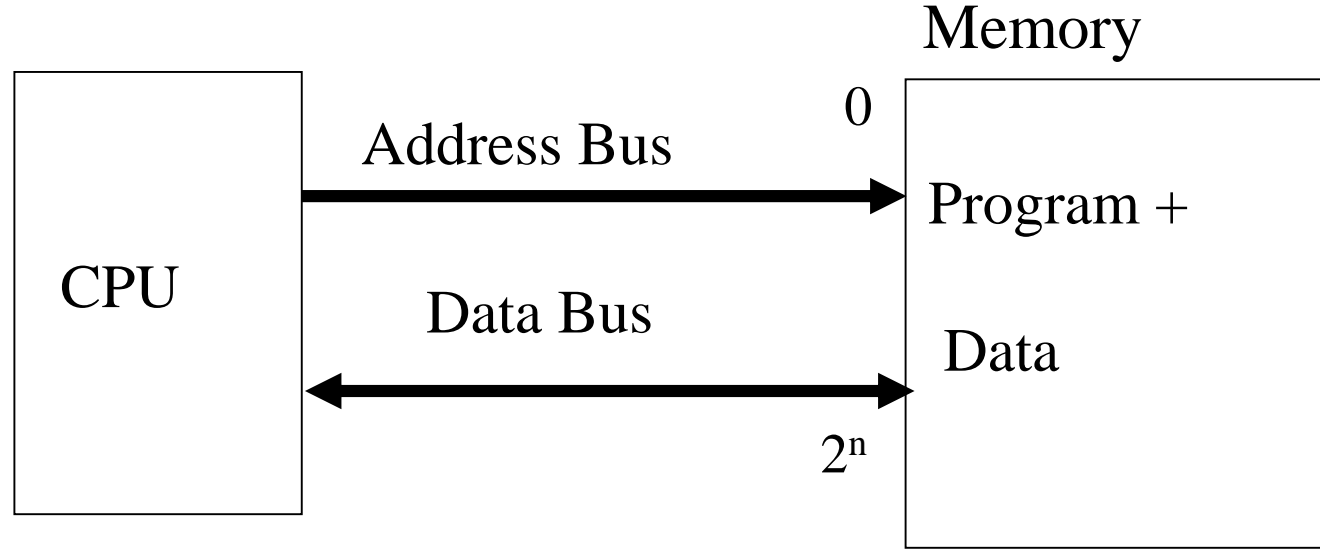
For example: a $2^{26} \times 16$ RAM contains $2^{26} = 64\text{M}$ words, each memory cell is 16 bits long.

- The RAM would need 26 address lines.
- The total bit *storage capacity* is $2^{26} \times 16 \text{ bit} = 2^{27} \times 8 \text{ bit} = 2^{27} \text{ byte} = 2^{30} \text{ bits}$.

- Data bus hat sayısı=16
- $2^{26} \text{ word} = 2^{26} * 16 \text{ bit}$
- $2^{27} \text{ byte} = 2^{27} * 8 \text{ bit}$
- $2^{30} \text{ bit} = 2^{27} * 2^3 \text{ bit}$

- Adres bus hat sayısı=26 adet.
- Hücresel uzunluğu word (16bit) olan belleğin indislenmesi : A25, A23, A22, ..., A1, A0

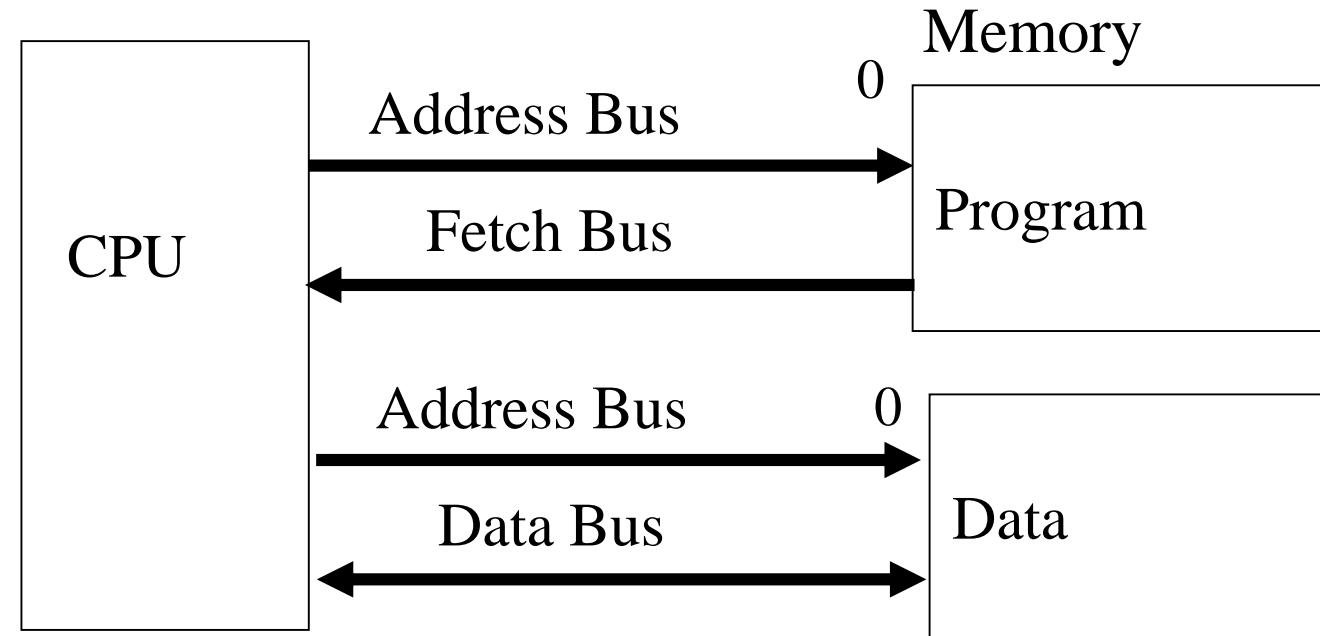
CPU Dış Mimarisi



Von Neumann Architecture

CPU işlev döngüsü:

- Address Decoding Unit
- Fetching: Veriyi alıp getirme ya da götürme
- Execution: Komut Yürütme, Veri uygulaması veya işlenmesi
- Microprocessor clock system



Harvard Architecture

Program Bellek: ROM

Data Bellek: RAM

- Eğer program bellek ve veri belleklere aynı adres ve data bus üzerinden erişiliyorsa Von Neumann mimari yapısıdır.
- Eğer program bellek ve veri belleklere ayrı adres ve data bus üzerinden erişiliyorsa Harvard mimari yapısıdır.

CPU İç Mimarisi

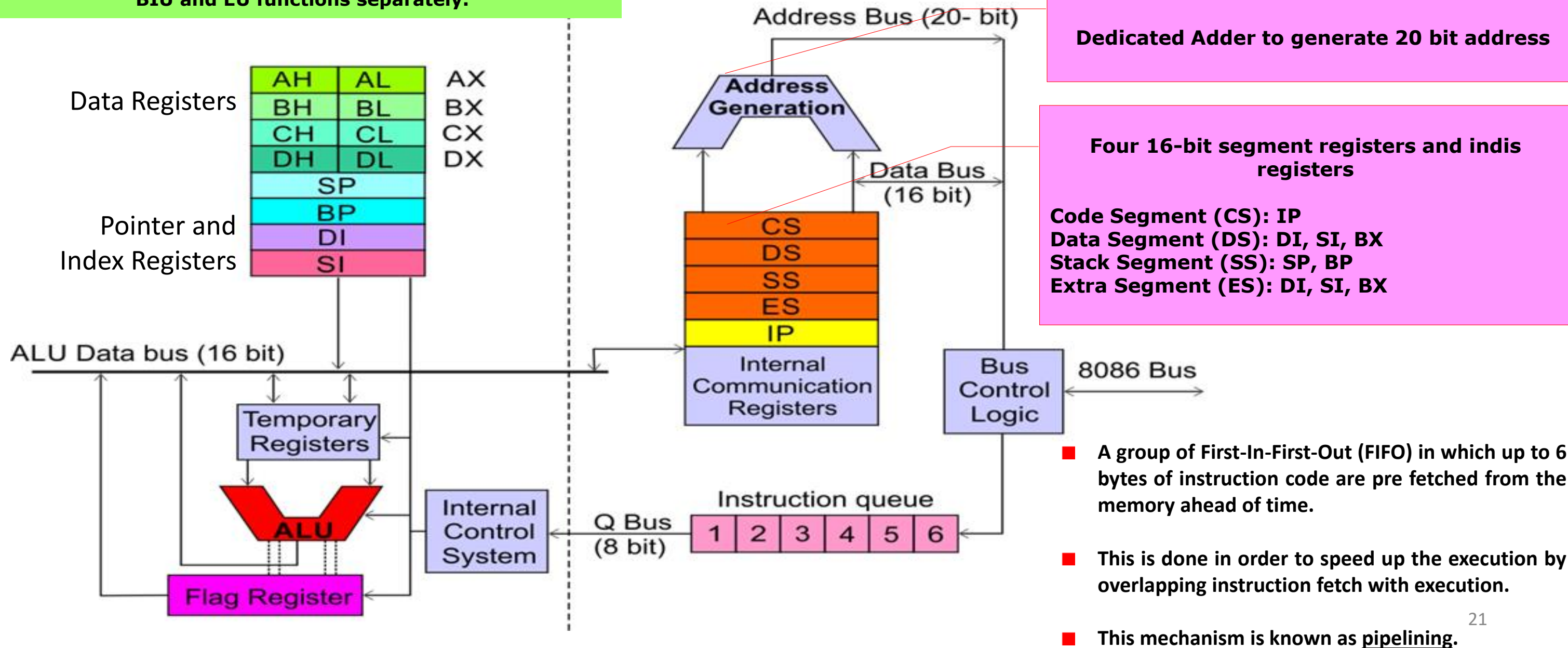
Execution Unit (EU)

EU executes instructions that have already been fetched by the BIU.

BIU and EU functions separately.

Bus Interface Unit (BIU)

BIU fetches instructions, reads data from memory and I/O ports, writes data to memory and I/O ports.



CPU İç Mimarisi

- Registers: Özel amaçlı geçici kayıt ediciler
- Cache Bellek
- ALU: Aritmetik Lojik Birimi
- Control Birimi: Bus Control Logic ve Internal Control System
- Flags
- Instruction Queue: Komut Kuruğu (Pipelining)
- Internal Communication Register (İç haberleşme hatları)
- Temporary Register
- Address Generation (20 bit fiziksel adres üretme)
- Clock and Timing

CPU'nun Temel Döngüsel İşlemlerin Tanımı

- **Veri:** Belleğe ya da giriş – çıkış birimlerine yazılan veya okunan değerlerdir.
- **Adres:** Üzerinde işlem yapılacak belleğin veya giriş – çıkış biriminin konumunu belirtir. Adres hatları belleği ve gözünü seçer. Tipik olarak: n bit hat, 2^n byte'lık konum belirlenmiş olur.
- **Bir döngü operasyon:** Belleğe gönderilen ya da bellekten temin edilen ve gerçekleştirilecek işlemin türünü belirleyen kontrol bilgisi olarak yorumlayan tipik işlemlerdir.
- **Bellek Oku:** Bellekte saklanan bir veri değerini okuyan bir işlem: Adres hatlarına geçerli bir adres girilir. Okunan verilerin kararlı hale gelmesini beklenir.
- **Bellek Yaz:** Belleğe bir veri değeri yazan bir işlem: Adres hatlarına geçerli bir adres ve veri hatları girilir.
- **Okuma veya yazma etkinleştirme hattı:** Hassas zamanlama bilgilerine sahip bir clock periyodu olarak tanımlanır (örn. Okuma Saati, Strobe Yazma).
- Komut Yürütme (Execution)

CPU İç Kayıt ediciler (Registers)

- CPU'nun iç mimarisinin ana bileşenleri:EU, BIU, ALU (Arithmetic Logic Unit), Komut Kuyruğu ve Register'lar.

Kayıt ediciler (Registers): Özel amaçlı saklayıcılardır. Yüksek hızlı veri işler ve transfer eder. Geçici depolama alanıdır. CPU içinde bulunur ve CPU'nun ana bileşenleridir.

- Toplam 14 adet register bulunmaktadır.
 - Data Register: AX, BX, CX, DX; matematiksel ve mantıksal işlemler burada yapılır. Ayrıca, BX register, DS ve ES ile birlikte indis olarak bellek gözünün adresini bulmada kullanılır. CX, register LOOP komutu sayıcı olarak kullanılır. AX ve DX, çarpma, bölme ve I/O erişimlerinde kullanılır.
 - Pointer: Stack segment indisi olarak: SP, BP; DS ve ES indisi olarak: SI, DI, BX; işlenecek bellek gözlerini işaret ederler.
- Register'lar 16bit uzunluğundadır.
- Segment register'lar 4 adettir. Belleklerin başlangıç adreslerini gösterirler.
 - Code Segment, kodların yazılı olduğu programların bulunduğu belleğin başlangıç adresini gösterir.
 - IP: CS ile birlikte indis olarak CPU tarafından işlenecek bir sonraki komutun yerini işaret eder.
 - DS, SS, ES ise veri bellek olan Ram belleklerin başlangıç adreslerini gösterirler.
- Flag: CPU, komutları işlerken değişen durumlar hakkında bilgi verir.

Yürütme Biriminde (Execution Unit - EU) Data Register'lar

Accumulator Register - Saklayıcı (AX)

- 16 bitlik AX saklayıcı olarak kullanılabilen iki adet 8-bitlik AL ve AH birbirinden bağımsız saklayıcılardan oluşur.
- Bu durumda AL, 16 bitlik Word'ün düşük sıralı baytını içerir ve AH, yüksek sıralı baytı içerir.
- G / Ç talimatları, bir G / Ç bağlantı noktasına 16 veya 8 bitlik veri girişi / çıkışı için AX veya AL'yi kullanır.
- Çarpma ve Bölme komutları da AX veya AL kullanır.

Base Register-Saklayıcı (BX)

- 16 bitlik bir BX saklayıcı olarak kullanılabilen iki adet 8 bitlik saklayıcı bloğu BL ve BH'den oluşur.
- Bu durumda BL, Word'ün düşük sıralı baytını içerir ve BH, yüksek sıralı baytı içerir.
- Bu saklayıcının içeriği hafızayı adreslemek için kullanılabilen tek genel amaçlı kayıttır.
- Bu kayıt içeriğini adresleme için kullanan tüm bellek referansları, varsayılan segment saklayıcı olarak DS'yi kullanır.

Counter Register – Saklayıcı (CX)

- 16 bitlik bir CX saklayıcı olarak kullanılabilen iki adet 8 bitlik saklayıcılar CL ve CH'den oluşur.
- Birleştirildiğinde, CL saklayıcı Word'ün düşük sıralı baytını içerir ve CH yüksek sıralı baytı içerir.
- SHIFT, ROTATE ve LOOP gibi döngüsel talimatlar, CX'in içeriğini sayaç olarak kullanır.

Data Register – Saklayıcı (DX)

- 16 bitlik bir DX olarak kullanılabilen iki adet 8 bitlik saklayıcı DL ve DH'den oluşur.
- Birleştirildiğinde, DL saklayıcı Word'ün düşük sıralı baytını içerir ve DH yüksek sıralı baytı içerir.
- 16 X 16 çarpmada yüksek 16 bitlik sonucu (veri) veya $32 \div 16$ bölmeden önce yüksek 16 bitlik payı (veri) ve bölmeden sonra 16 bitlik saklayıcıda tutmak için kullanılır.
- Bu kayıt AX ile birlikte, 16 bitlik bir çarpmanın 32 bitlik bir sonucunun üst yarısını depolamak veya bir tamsayı bölmesinden sonra kalanı tutmak gibi şeylere izin veren özel aritmetik işlevler için kullanılır.
- Giriş / Çıkış işlemlerinde kullanılır.

Pointer – Index Registers

Stack Pointer (SP) and Base Pointer (BP)

- SP ve BP, yığın segmentindeki (SS) verilere erişmek için kullanılır.
- SP, harici bellekteki yığın segmentini içeren talimatların yürütülmesi sırasında mevcut SS'den bir ofset olarak kullanılır.
- POP veya PUSH komutunun yürütülmesi sırasında, SP içerikleri otomatik olarak güncellenir (artırılır / azaltılır).
- BP, mevcut SS'de, temelli adresleme modunu kullanan talimatlar tarafından kullanılan bir ofset adresi içerir.

Source Index (SI) and Destination Index (DI)

- Endekslenmiş adreslemede kullanılır.
- Veri dizilerini işleyen komutlar, kaynak ve hedef adresleri ayırt etmek için sırasıyla DS ve ES ile birlikte SI ve DI kayıtlarını kullanır.

Instruction Pointer (IP)

- Her zaman o anda yürütülen kod segmenti içinde yürütülecek bir sonraki talimatı işaret eder.
- Kod segment alanının bellek içindeki bir sonraki talimat kodunu işaret eden 16 bitlik ofset adresini içerir.
- Bir sonraki talimatın yürütülmesi sırasında içeriği otomatik olarak artırılır.

Segment Registers

Segment Register 16bittir. Segment register'lerden 20bitlik fiziksel adres üretilir.

Code Segment Register- 16-bit

- CS, geçerli kod segmentinin tabanını veya başlangıcını içerir; IP, bu adresten alınacak bir sonraki talimat baytına kadar olan mesafeyi veya ofseti içerir.
- Address Generation CS'in içeriğinin sağ tarafına 4bit sıfır ekleyerek yani mantıksal olarak 4 adet sıfır bit sola kaydırarak ve ardından 16 bit IP içeriğini ekleyerek 20 bitlik fiziksel adresi hesaplar.
- Bir programın tüm talimatları CS yazmacının içeriğine göre 16 ile çarpılır ve ardından IP tarafından sağlanan ofset eklenir.

Data Segment Register - 16-bit

- Mevcut veri segmentine işaret eder; Çoğu talimat için işlenenler veriler bu segmentten alınır.
- Kaynak Dizininin (SI) veya Hedef Dizininin (DI) 16 bitlik içeriği veya 16 bitlik bir yer değiştirme, 20 bitlik fiziksel adresi hesaplamak için ofset olarak kullanılır. DS'in içeriği olan bit dizinine sağdan 4 bit 0 eklenerek öteleme (ya da 16 ile çarpılır).

Stack Segment Register - 16-bit

- Mevcut yığına işaret eder.
- 20-bit fiziksel yığın adresi, PUSH ve POP gibi yığın talimatları için Yığın Segmentinden (SS) ve Yığın İşaretçisinden (SP) hesaplanır.
- Esaslı adresleme modunda, 20 bitlik fiziksel yığın adresi Yığın segmentinden (SS) ve Temel İşaretçiden (BP) hesaplanır.

Extra Segment Register - 16-bit

- Verilerin (DS tarafından işaret edilen 64K'dan fazla) depolandığı ekstra segmenti işaret eder.
- Dize talimatları, hedefin 20 bitlik fiziksel adresini belirlemek için ES ve DI'yi kullanır.

Segment Register'lar

Segment register'lar 16bitliktir. Belleklerin başlangıç adreslerini saklarlar. Gerçek fiziksel adres 20 bitliktir. Segment register'ların sonuna 4 bit 0 eklenir. Bit dizisinin sağ tarafına bir bit 0 eklenirse, 2 ile çarpılmış olur. 4 bit sıfır eklenirse 16 ile çarpılmış olur.

- CS: Code segment 16 bit ROM belleğin başlama adresini saklar.
- DS: Data segment 16 bit RAM belleğin başlama adresini saklar.
- SS: Stack segment 16 bit RAM, Yığın veri saklama belleğin başlama adresini saklar.
- ES: Extra Segment 16 bit RAM, Ekstra yığın veri saklama belleğin başlama adresini saklar.

Bellek Organizasyonunda Segment ve İlgili Register'lar

Bellek Organizasyonunda aktif olarak kullanılan Register'lar:

- CS, DS, SS, ES belleklerin başlangıç adreslerini gösterirler (fiziksel 20bitlik).
- Başlangıç adresinden komutun ya da verinin olduğu bellek gözüne erişimde kullanılan register'lar:
 - IP: CS ile belirlenen fiziksel 20 bitlik ROM belleğin başlangıç adresinden bir sonraki adımda işlenecek komut kodunun olduğu göze erişimde kullanılır.
 - SI, DI, BX: DS ve ES ile belirlenen fiziksel 20 bitlik RAM belleğin başlangıç adresinden işlenecek verilerin olduğu göze erişimde kullanılır.
 - BP, SP: SS ile belirlenen fiziksel 20 bitlik RAM belleğin başlangıç adresinden işlenecek verilerin olduğu göze erişimde kullanılır.

Fiziksel Segment Adresleme

- BIU blok diyagramının merkezinde, CS, DS, SS ve ES olarak etiketlenmiş bir dizi segment özel amaçlı saklayıcı (register) bulunur. Bu dört adet 16 bitlik saklayıcı, öğeleri bellek alanından saklamak ve geri almak için işaretçi (Pointer register) ve dizin (Index Register) saklayıcılar ile birlikte kullanılır.
- Önce segment saklayıcısındaki değeri alır ve bit olarak sıfır ekleyerek dört basamak sağdan sola kaydırır. Örneğimizde, segment saklayıcısındaki değer $(3241)_{16} = (0011\ 0010\ 0100\ 0001)_2$ 'dir. Segmentin sağ tarafına dört bit sıfır eklendiğinde, $(32410)_{16} = (0011\ 0010\ 0100\ 0001\ 0000)_2$ elde edilir.
- Daha sonra işaretçi veya dizin saklayıcıların içeriği 20 bitlik fiziksel adrese eklenir. Böylece belleğin okunacak ya da yazılacak gözü belirlenmiş olur.
- İkili sayı sisteminin sağ tarafına bir bit sıfır konduğunda 2 ile, iki bit sıfır konduğunda 4 ile üç bit sıfır konduğunda 8 ile 4 bit sıfır konduğunda 16 ile çarpılmış olmaktadır.

$$\begin{array}{r} 0011\ 0010\ 0100\ 0001\ 0000 \\ + \quad \quad 1010\ 0011\ 0100\ 1110 \\ \hline 0011\ 1100\ 0111\ 0101\ 1110 \end{array} \quad \text{or} \quad \begin{array}{r} 32410_{16} \\ + \quad \underline{A34E}_{16} \\ \hline 3C75E_{16} \end{array}$$

Adres Bus Üzerinden Bellek Gözü Seçmek

Data Bus hat sayısı ile Adres Bus arasındaki ilişki

Örnek: Data Bus hat sayısı= 8 bit ise adres bus hatları tanımı byte olarak verilir. Adres hat sayısı=16 bit ise bellek kapasitesi bit olarak nedir?

Bellek kapasitesi byte olarak= $2^{16}=2^6 * 2^{10}$ byte (Data bus hat sayısı=8 bit=1Byte olarak verilmiş)
=64KByte

Bellek kapasitesi bit olarak= $2^{16} * 2^3$ bit= 2^{19} bit

Örnek: Data Bus hat sayısı= 16 bit ise adres bus hatları tanımı byte olarak verilir. Adres hat sayısı=24 bit ise bellek kapasitesi nedir?

Bellek kapasitesi word olarak (2 Byte)= $2^{24}=2^4 * 2^{20}$ word (Data bus hat sayısı=16 bit=2Byte olarak verilmiş) =16MWord

Bellek Kapasitesi Byte olarak= $2^{24} * 2$ Byte= 2^{15} Byte=32 Mbyte

Bellek Kapasitesi Bit olarak= $2^{24} * 2^3$ Bite= 2^{27} Bit= 128 Mega bit

Adres hatları üzerindeki bitler verinin yazılacağı ya da okunacağı bellek gözünü işaret eder

Adres bus hatları üzerindeki elektriksel sinyal olarak bulunan bit (0/1) değerleri seçilecek olan belleğin gözünü işaret eder.

Örneğin Adres hattı üzerinde (110)_b değerleri var ise belleğin hangi gözünü seçer? Belleğin indis olarak 6 ıncı gözünü seçer. Seçilen bellek gözünün içeriği nedir? (0011 0011)_b=(33)_h

Example memory contents:

- A memory with 3 address bits & 8 data bits has:
- $k = 3$ and $n = 8$ so $2^3 = 8$ addresses labeled 0 to 7.
- $2^3 = 8$ words of 8-bit data

Memory Address		Memory Content
Binary	Decimal	
0 0 0	0	1 0 0 0 1 1 1 1
0 0 1	1	1 1 1 1 1 1 1 1
0 1 0	2	1 0 1 1 0 0 0 1
0 1 1	3	0 0 0 0 0 0 0 0
1 0 0	4	1 0 1 1 1 0 0 1
1 0 1	5	1 0 0 0 0 1 1 0
1 1 0	6	0 0 1 1 0 0 1 1
1 1 1	7	1 1 0 0 1 1 0 0

Main Memory

- Because computers operate by storing numbers as bit pattern, the address is also represented as bit pattern
- Address starts from 0 to last addressable word in address space.
- To address 1 kB (2^{10}) of memory, you need touse 16 bit for addressing

Örnek: Yandaki bellek haritalamada (000)h ile (3FFF)h aralığında bellek kapasitesi nedir?

- Bitiş adresinden başlangıç adres çıkarılır 1 eklenerek bellek kapasitesi byte olarak bulunur (data Bus hat sayısı 8 olarak verilmiş olsun)
- $3F)h - 00)h = 3FFh$
- Bellek hat sayısı = $3FFh + 1 = 400h$ (0100 0000 0000), 11 adet
- Bellek Kapasitesi = $2^{10} \text{Byte} = 1 \text{Kbyte}$

<u>Memory Address</u>		<u>Memory Contents</u>
<u>Binary</u>	<u>Hex</u>	
00-0000-0000	000	10011001
00-0000-0001	001	00111000
00-0000-0010	002	11001001
00-0000-0011	003	00111011
⋮	⋮	⋮
11-1111-1100	3FC	01101000
11-1111-1101	3FD	10111001
11-1111-1110	3FE	00110100
11-1111-1111	3FF	00011000

1024 X 8 (or 1KX8) Memory

Örnek: Bellek gözü seçmek

- 8 bit (byte) veri gözlerine, 4 bit adres hattına sahip bir belleğin
- Seçilecek göz sayısı= $2^n=2^4=16$ byte
- Burada, n: bellek adres hattı sayısıdır.
- Seçilecek gözlerin indislenmesi: 0, ..., 2^n-1
- 0,1,2, ... , 15
- Örneğin Adres hattı üzerinde (1100)b değerleri var ise belleğin hangi gözünü seçer? Belleğin indis olarak 12inci gözünü seçer. Seçilen bellek gözünün içeriği nedir? (1001 1100)b=(9C)h

Bellek Gözü Adresleri					Bellek İçeriği								
Binary				Ondalık	Binary								Hex
0	0	0	0	0	1	0	1	1	0	0	1	0	B2h
0	0	0	1	1	0	1	0	1	0	1	0	0	54h
0	0	1	0	2	1	0	1	0	0	1	0	1	A5h
0	0	1	1	3	1	1	0	0	1	1	0	0	CCh
0	1	0	0	4	0	0	1	0	0	1	0	0	24h
0	1	0	1	5	1	1	0	0	0	1	1	0	C6h
0	1	1	0	6	0	1	1	1	1	0	1	1	7Bh
0	1	1	1	7	0	0	1	1	0	0	1	1	33h
1	0	0	0	8	1	1	0	1	1	1	1	0	DEh
1	0	0	1	9	1	0	1	0	1	0	1	0	AAh
1	0	1	0	10	0	1	0	1	1	1	0	1	5Dh
1	0	1	1	11	1	1	1	0	1	0	0	0	E8h
1	1	0	0	12	1	0	0	1	1	1	0	0	9Ch
1	1	0	1	13	1	0	1	0	1	0	1	1	ABh
1	1	1	0	14	1	0	0	1	1	0	0	1	99H
1	1	1	1	15	1	1	0	0	1	0	0	1	C9h

Bellek Kapasitesi Gösterim Formatı

- Bellekler, AByte x N (**16Kx8**) formatında gösterilir. A belleğin kapasitesini byte olarak gösterir. (K:Kilobyte= 2^{10} byte, M:Megabyte= 2^{20} byte, G:Gigabyte= 2^{30} byte, T: Terabyte= 2^{40} byte, P:Petabyte= 2^{50} byte).
- N ise data bus hat sayısını bit olarak verir. N=8 bit(byte), 16bit(word), 32bit(Longword), 64 bit(Quadword) olur.

Soru: Bir belleğin gösterim formatı 16Kx8 ise belleğin veri saklama kapasitesini byte olarak, adres hattı sayısı adet olarak ve data bus hat sayısını bit olarak bulunuz.

- Belleğin veri saklama kapasitesi=16Kbyte (**16Kx8**)
- Belleğin adres hattı sayısının bulunması: Belleğin veri saklama kapasitesi=16KByte= 2^{14} Byte'dır ($2^4 * 2^{10}$). O halde 2 üzeri üs ifadesi belleğin adres hattı sayısını verir ki, adres hattı sayısı=14 adet bulunur.
- Data bus hattı sayısı=8 bit'dir. (16Kx**8**)

Örnek - Bellek Kapasitesi Gösterim Formatı

Soru: 128Mx16 formatında verilen belleğin özelliklerini yorumlayınız.

Bu belleğin veri saklama kapasitesi 128Megabyte, veri yazıp okuması 16bittir.

Bir belleğin veri saklama kapasitesi 2^{27} byte, veri yazıp okuması 16bit

Adres Bus hat sayısı=27dir.

Her bir hat üzerinde bit olarak 0 ya da 1 elektriksel sinyal vardır.

Soru: 32Kx8 formatında verilen belleğin özelliklerini yorumlayınız.

Bu belleğin veri saklama kapasitesi 32Kbyte, veri yazıp okuması 8bittir.

Bir belleğin veri saklama kapasitesi 2^{15} byte, veri yazıp okuması 8bit

Adres Bus hat sayısı=15dir.

Her bir hat üzerinde bit olarak 0 ya da 1 elektriksel sinyal vardır.

Örnek: Data Bus hat uzunluğu 16 bit olduğunda

Soru: 32 bit (adet) adres yolu hattı ve 16 bit (adet) veri yolu hattı, word (2 byte) hattı varsa hafızanın boyutunu nasıl hesaplayabiliriz.

- Belleklerde bellek boyutunu, adres hattı sayısını kullanarak buluyoruz.
- $m=32$ ise Bellek boyutu= 2^m byte'dır. Bellek boyutu= 2^{32} byte= $2^2 \times 2^{30}$ =4Gbyte
- Adres yolu: Belleğin veri yazılacak ya da okunacak her bir gözünü seçer
- Data yolu yazılacak ya da okunacak veriyi temsil eder.
- 16bit dendiğinde iki durum söz konusu:
 - Byte, 8 bit yazıp okur. Byte olarak toplam bellek kaç göz var? 2^{32} adet bellek gözü var. Her bir göze bir byte (8bit) yazılıp okunur.
 - Word, 16 bit yazıp okur. Word olarak bellekte kaç göz vardır. Word=2 byte=16bit. $2^{32}/2=2^{31}$ word. Yani 2Gigaword.

Örnek: Data bus hat uzunluğu 64 bit olduğunda

Soru: 32 bit (adet) adres yolu hattı ve 64 bit (adet) veri yolu hattı, word (8 byte) hattı varsa hafızanın boyutunu nasıl hesaplayabiliriz.

- Belleklerde bellek boyutunu, adres hattı sayısını kullanarak buluyoruz. Adres yolu, belleğin veri yazılacak ya da okunacak her bir gözünü seçer. Burada $m=32$ ise Bellek boyutu = 2^m byte'dır. Bellek boyutu = 2^{32} byte = $2^2 \times 2^{30}$ = 4Gbyte
- Data yolu yazılacak ya da okunacak veriyi temsil eder. 64bit dendiğinde iki durum söz konusudur:
 - Byte, 8 bit yazıp okur. Byte olarak toplam bellek kaç göz var? Adres yolu hat sayısından 2^{32} adet (Byte) bellek gözü olduğu biliyor. Her bir göze bir byte (8bit) yazılıp okunur.
 - Word, 16 bit yazıp okur. Word olarak bellekte kaç göz vardır. 1 Word = 2 byte = 16bit. $2^{32}/2 = 2^{31}$ word. Yani 2Gigaword.
 - Double word, 32 bit yazıp okur. Double word olarak bellekte kaç göz vardır. Double Word = 4 byte = 32bit. $2^{32}/2^2 = 2^{30}$ double word. Yani 1Gigadoubleword.
 - Qdouble word, 64 bit yazıp okur. Qdouble word olarak bellekte kaç göz vardır. Qdouble Word = 8 byte = 64bit. $2^{32}/2^3 = 2^{29}$ word. Yani 512Qdoubleword .

Memory Mapping

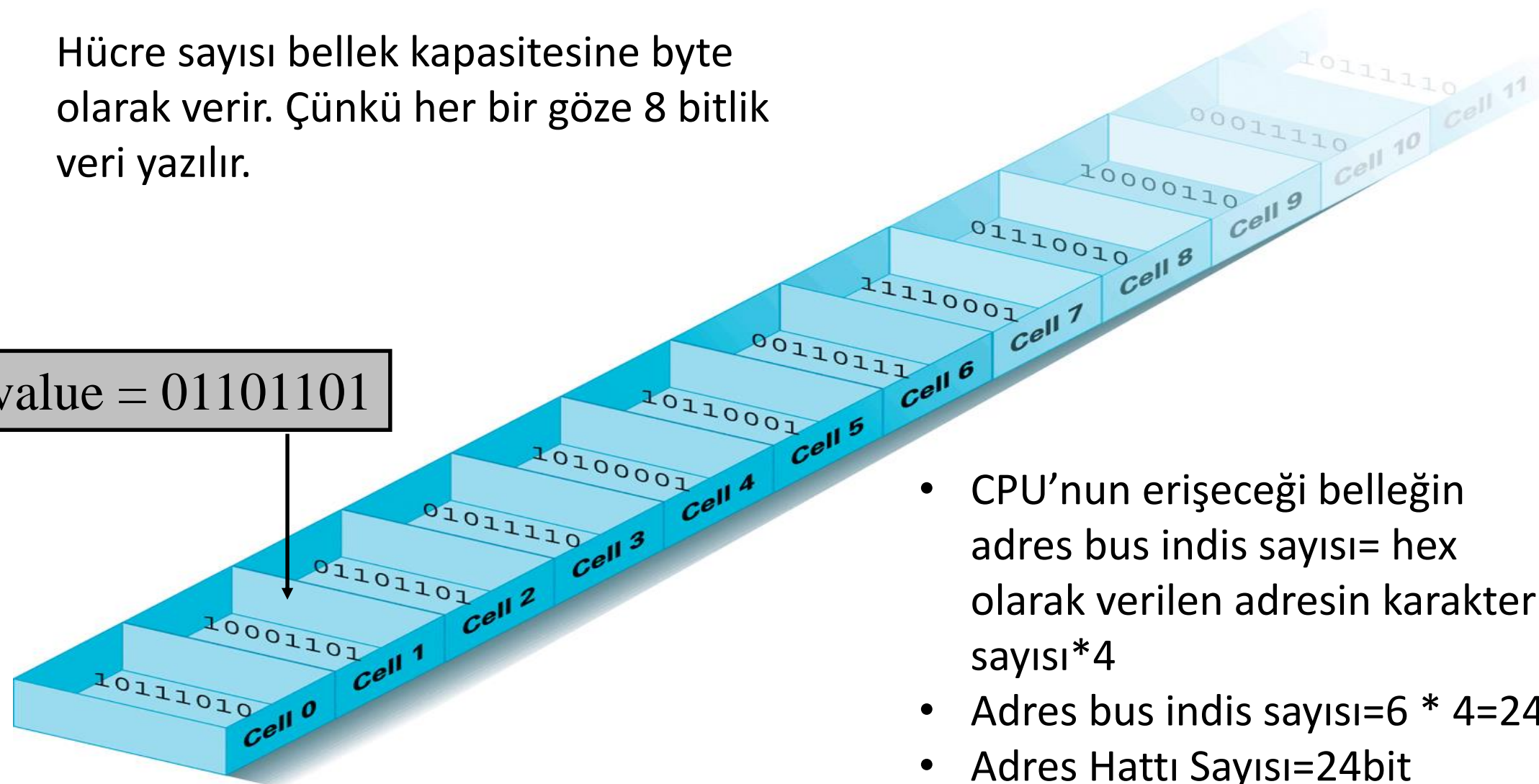
Bellek Denildiğinde

- Belleğin kapasitesi byte (8 bit) olarak verilmiş ise kapasite yte olarak belleğin adres bus hat sayısıdır.
- Belleğe, yazma ve okuma döngüsünde data bus'daki hat sayısı: Byte, Word, DW, QW,...
- Range, aralık: bellek haritalamada belleklerin başlangıç ve bitiş adresleri, ikili sayı sisteminde (binary) ya da hex olarak ifade edilir. Belleğin kapasitesini bulmak için bitiş adresinden başlangıç adresi çıkarılır. +1 eklenir. Çünkü bellek haritalaması 0 dan başlar: 0. bellek gözü, 1. bellek gözü,
- Adres decoding devresi, CPU tarafından hangi belleğin seçileceğini tanımlar.

Arrangement of Memory Cells

Hücre sayısı bellek kapasitesine byte olarak verir. Çünkü her bir göze 8 bitlik veri yazılır.

value = 01101101



- CPU'nun erişeceği belleğin adres bus indis sayısı= hex olarak verilen adresin karakter sayısı*4
- Adres bus indis sayısı=6 * 4=24
- Adres Hattı Sayısı=24bit
- Kapasite=2^24=16Megabyte
- İndisleyin: A23, A22, ..., A1, A0

Address	Data
000000	8 Bit
000001	
000002	
.	
.	
.	
.	
.	
.	
.	
.	
.	
FFFFFFD	
FFFFFFE	
FFFFFFF	8 Bit

Örnek:

- Bir belleğin 32 MB (Megabyte) saklama kapasitesi vardır. Bellekteki herhangi bir **byte (Data Bus)** veriyi adreslemek için kaç bit ya da hat gereklidir?

Bellek adres alanı 32 MB veya 2^{25} ($2^5 \times 2^{20}$) Byte. Bu, her Byte'ı adreslemek için $\log_2 2^{25}$ veya 25 bit'e ya da 25 adet adres hattına ihtiyacınız olduğu anlamına gelir. Grup olarak paralel çalışırlar. Herbiri üzerinde bir bit: 0 ya da 1 elektriksel sinyal değeri mevcuttur. Tek yönlüdür.

Adres hattı indisleri: A24, A23, A22, ..., A1, A0

Example:

Bir belleğin 128 MB saklama kapasitesi vardır. Veriyolu (Data Bus) = 8 bit ise Bellekteki herhangi bir veriye erişmek için kaç bit adres hattı gereklidir?

Bellek adres alanı 128 MB, yani 2^{27} 'dir.

27 adet adres hattına ihtiyaç olduğu anlamına gelir.

Adres hattı indisleme: A26, A25, A24, ... , A5, A4, A3, A2, A1, A0

Range Calculate – Memory Capacity

Example: Bellek kapasite aralığı (4000)h – (5FFFh) ise bellek kapasitesi kaç byte'dır?

Range=(5fff)h-(4000)h

Çıkarma işlemi: (4000)H değerine ne eklersem, (5fff)h elde ederim.

(5FFF)h=(0101 1111 1111 1111)b

(4000)h=(0100 0000 0000 0000)b

(5FFF)h – (4000)h=(1FFF)h;

(1FFF)h=(0001 1111 1111 1111)b

Bellek Kapasitesi=(0001 1111 1111 1111)b +1 = (0010 0000 0000 0000)b = 2^{13} byte = $2^3 2^{10}$ byte =8Kbyte

Not: indis 0 dan başlar. Sağdan sola sayılır. 1'in olduğu indis:13 dür.

Range Calculate – Memory Capacity

Example: Range: Bir belleğin erişim aralığıdır. Başlangıç ve bitiş adres değerleri ile verilir.

Bir belleğin erişim aralığı: (4000)h – (6FFF)h

Range= Bitiş adresi – başlangıç adresi=(6fff)h-(4000)h

(6FFF)h=(0110 1111 1111 1111)b

(4000)h=(0100 0000 0000 0000)b

(6FFF)h – (4000)h=(2FFF)h;

Toplam kapasite=2FFFh + 1=3000h

Bellek Kapasitesi=(0010 1111 1111 1111)b +1 = (0011 0000 0000 0000)b = $(2^{13} + 2^{12})$ byte = $2^{12} (2+1)$ byte = $3 \cdot$

2^{12} byte = $3 \cdot 2^2 \cdot 2^{10}$ byte=12Kbyte

Range Calculate – Memory Capacity

Example: Belleğin fiziksel başlangıç adres=(C000)h ve bellek kapasitesi 4Kbyte ise belleğin adres hattı sayısı kaç adettir? İndisleyin ve belleğin fiziksel bitiş adresini bulunuz.

$$4\text{Kbyte}=2^2*2^{10}\text{ byte}=2^{12}\text{ byte}$$

Belleğin adres hattı sayısı=12 adettir.

Adres indisi: A11, A10, A9, ..., A2, A1, A0

$$(C000)h=(1100\ 0000\ 0000\ 0000)b$$

$$(0FFF)h=(0000\ 1111\ 1111\ 1111)b$$

Belleğin fiziksel bitiş adresi= Belleğin başlangıç adresi + kapasite

$$\text{Belleğin fiziksel bitiş adresi}=(C000)h+(0FFF)=(CFFF)h$$

Range Calculate – Memory Capacity

Example: 64Kbyte 8 bit belleğin

- a) Bu belleğe CPU'dan gelen adres hattı sayısı kaç adettir? Bu belleğin adres hatlarını indisleyin**
- b) Data hattı sayısı kaç adettir?**
- c) Belleğin fiziksel başlangıç ve bitiş adresini belirleyiniz.**

Yanıt:

a) $64\text{Kbyte} = 2^6 * 2^{10} \text{ byte} = 2^{16} \text{ byte}$, belleğin adres hattı sayısı=16 adettir.

Adres hatları indisi: A15, A14, A13, ..., A2, A1, A0

b) 8 adettir. Data hatları indisi: D7, D6, D5, ..., D1, D0

c) Belleğin başlangıç adresi = (0000 0000 0000 0000)b=(0000)h

Belleğin bitiş adresi = (1111 1111 1111 1111)b=(FFFF)h

Hex sayı sistemine +1 ekleme

- $0h+1 = (0000)_b+1 = (0001)_b=1h$
- $1h+1 = (0001)_b+1 = (0010)_b=2h$
- $2h+1 = (0010)_b+1 = (0011)_b=3h$
- $3h+1 = (0011)_b+1 = (0100)_b=4h$
- $4h+1 = (0100)_b+1 = (0101)_b=5h$
- $5h+1 = (0101)_b+1 = (0110)_b=6h$
- $6h+1 = (0110)_b+1 = (0111)_b=7h$
- $7h+1 = (0111)_b+1 = (1000)_b=8h$
- $8h+1 = (1000)_b+1 = (1001)_b=9h$
- $9h+1 = (1001)_b+1 = (1010)_b=Ah$
- $Ah+1 = (1010)_b+1 = (1011)_b=Bh$
- $Bh+1 = (1011)_b+1 = (1100)_b=Ch$
- $Ch+1 = (1100)_b+1 = (1101)_b=Dh$
- $Dh+1 = (1101)_b+1 = (1110)_b=Eh$
- $Eh+1 = (1110)_b+1 = (1111)_b=Fh$
- $Fh+1 = (1111)_b+1 = (1\ 0000)_b=10h$
- $Ah+5 = 0Fh$
- $Bh+5 = 10h$
- $Ch+5 = 11h$
- $Dh+5 = 12h$
- $Eh+5 = 13h$
- $Fh+5 = 14h$
- $Ah+Ah = (10)_d+(10)_d=(20)_d=14h$

Adres Bus Üzerinden Bellek Gözü Adresinin Belirlenmesi

- Örnek: Bir belleğin Kapasitesi 256MByte ise
 - a) Bellek kapasitesini byte olarak bulunuz. $256\text{MByte}=2^8 \cdot 2^{20}=2^{28}$ Byte
 - b) Adres hattı sayısını bulunuz. 28 adet
 - c) Adres hattını indisleyiniz. A27, A26, ... , A1, A0
 - d) Adres hattı kaç adet hex sayı sisteminden oluşur. $28/4=7$ adet.
 - e) Belleğin başlangıç ve bitiş adreslerini hex cinsinden yazınız. (0000000)h ... (FFFFFFF)h
 - f) 7500 üncü gözün adresini hex cinsinden yazınız. Bellek gözü 0, 1, 2, ... 7500, ...
 $(7500)_d=2^{12}+2^{11}+2^{10}+2^8+2^6+2^3+2^2$
 $(7500)_d=(0001\ 1101\ 0100\ 1100)_b=(1D4C)_h$
 - h) Bu belleğin segment adresi (A000)h ise ilgili bellek gözünün 20 bitlik fiziksel adresini bulunuz.
 $(\text{Segment adres})_h \times (10)_h = (A000)_h \times (10)_h = (A0000)_h$
Bellek gözünün fiziksel adresi= $(A0000)_h + (1D4C)_h = (A1D4C)_h$

Adres Bus Üzerinden Bellek Gözü Adresinin Belirlenmesi

- Örnek: İki belleğin her birinin kapasitesi 64KByte, Segment adresleri ise (0000)h ve (A000)h ise
 - a) Bellek kapasitelerini byte olarak bulunuz. $64\text{KByte} = 2^6 * 2^{10} = 2^{16}$ Byte
 - b) Adres hattı sayısını bulunuz. 16 adet
 - c) Adres hattını indisleyiniz. A15, A14, ... , A1, A0
 - d) Adres hattı kaç adet hex sayı sisteminden oluşur. $16/4 = 4$.
 - e) Belleğin başlangıç ve bitiş adreslerini hex cinsinden yazınız. (0000)h ... (FFFF)h
 - f) Birinci belleğin kapasite aralığı: (0000)h
 - h) Birinci belleğin segment adresi (0000)h ise ilgili bellek gözünün 20 bitlik fiziksel adresini bulunuz.
 $(\text{Segment adres})h \times (10)h = (0000)h \times (10)h = (00000)h$
Başlangıç fiziksel adresi: (00000)h, Bitiş fiziksel adresi: (0FFFF)h
 - i) İkinci belleğin segment adresi (A000)h ise ilgili bellek gözünün 20 bitlik fiziksel adresini bulunuz.
 $(\text{Segment adres})h \times (10)h = (A000)h \times (10)h = (A0000)h$
Başlangıç fiziksel adresi: (A0000)h, Bitiş fiziksel adresi: (AFFFF)h

Başlangıç ve Bitiş Adreslerinden Bellek Kapasitesinin Belirlenmesi

- Örnek: Bir belleğin başlangıç adresi (C0000)h ve bitiş adresi (DFFFF)h ise belleğin kapasitesi nedir?
- Bitiş adresinden başlangıç adresi çıkarılır. $(DFFFF)h - (C0000)h = (1FFFF)h$
- $(1FFFF)h = (0001\ 1111\ 1111\ 1111\ 1111)b$
- $(0001\ 1111\ 1111\ 1111\ 1111)b + 1 = (0010\ 0000\ 0000\ 0000\ 0000)b$
- Sağdan indis-0 ile başlanarak indislenir. 1 olan indislerin 2 üzeri değerleri toplanarak byte cinsinden kapasite bulunmuş olur.
- İndis, $n=17$
- Belleğin kapasitesi $= 2^n = 2^{17} = 128\text{Kbyte}$

Başlangıç ve Bitiş Adreslerinden Bellek Kapasitesinin Belirlenmesi

- Örnek: Bir belleğin başlangıç adresi (A000)h ve bitiş adresi (BCFF)h ise belleğin kapasitesi nedir?
- Bitiş adresinden başlangıç adresi çıkarılır. $(BCFF)h - (A000)h = (1CFF)h$
- $(1CFF)h = (0001\ 1100\ 1111\ 1111)b$
- $(0001\ 1100\ 1111\ 1111)b + 1 = (0001\ 1101\ 0000\ 0000)b = (1D00)h$
- Sağdan indis-0 ile başlanarak indislenir. 1 olan indislerin 2 üzeri değerleri toplanarak byte cinsinden kapasite bulunmuş olur.
- İndisler: 11, 10, 8
- Belleğin kapasitesi = $2^{12} + 2^{11} + 2^{10} + 2^8 = 2^8(2^4 + 2^3 + 2^2 + 1) = 2^8(16 + 8 + 4 + 1) = 29 * 2^8 \text{ byte} = 29 * 256 = 7424 \text{ Byte}$

Belleğin Başlangıç Adresi ve Kapasitesinden Bitiş Adresinin Belirlenmesi

- Örnek: Bir belleğin başlangıç adresi (A000)h ve belleğin kapasitesi 3328 byte ise bitiş adresi nedir?
- Belleğin kapasitesi 2 üzeri olarak indislenir toplamı biçiminde yazılır.
- $3328=2048+1024+256= 2^{11}+2^{10}+2^8$
- Belirlenen indislerden ikili sayı sistemi oluşturulur. (0000 1101 0000 0000)b
- İkili sayı sisteminden -1 çıkarılır. $(0000\ 1101\ 0000\ 0000)_b - 1 = (0000\ 1100\ 1111\ 1111)_b = (0CFF)_h$
- Bulunan değer başlangıç adresi ile toplanarak belleğin bitiş adresi belirlenir.
- $(A000)_h + (0CFF)_h = (ACFF)_h$

Belleğin Başlangıç Adresi ve Kapasitesinden Bitiş Adresinin Belirlenmesi

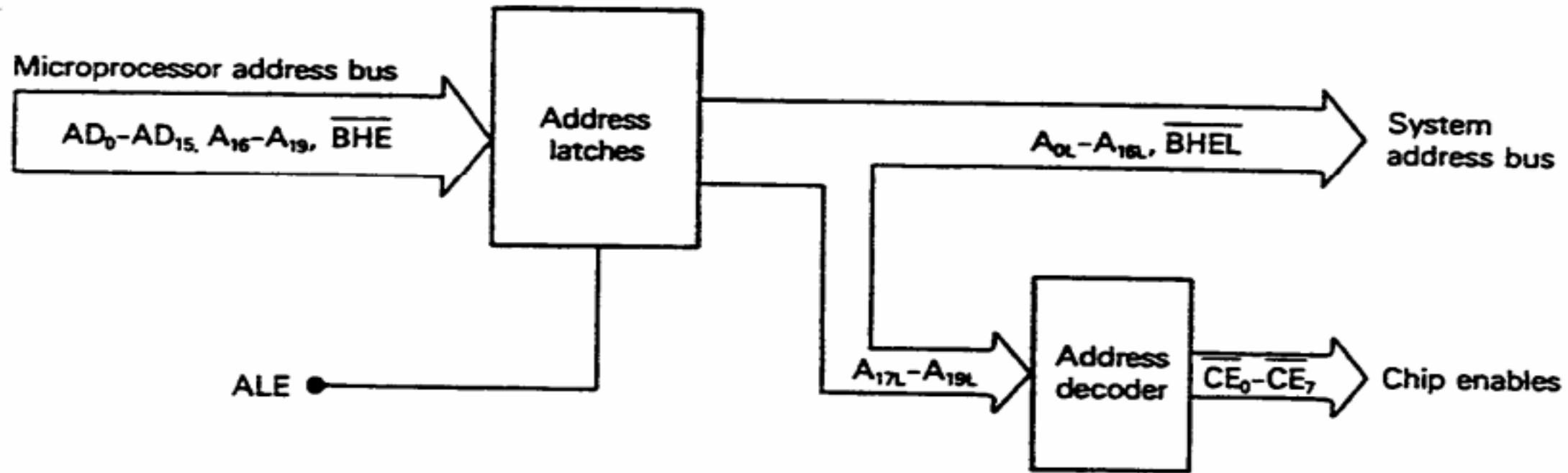
- Örnek: Bir belleğin başlangıç adresi (900)h ve belleğin kapasitesi 475byte ise bitiş adresi nedir?
- Belleğin kapasitesi 2 üzeri olarak indislenir toplamı biçiminde yazılır.
- $480=256+128+64+32= 2^8+2^7+2^6+2^5$
- Belirlenen indislerden ikili sayı sistemi oluşturulur. (0001 1110 0000)b
- İkili sayı sisteminden -1 çıkarılır. (0001 1110 0000)b - 1= (0001 1101 1111)b= (1DF)h
- Bulunan değer başlangıç adresi ile toplanarak belleğin bitiş adresi belirlenir.
- (1001 0000 0000)b + (0001 1101 1111)b=(1010 1101 1111)b
- (900)h + (1DF)h=(ADF)h

Belleğin Bitiş Adresi ve Kapasitesinden Başlangıç Adresinin Belirlenmesi

- Örnek: Bir belleğin bitiş adresi (BCFF)h ve kapasitesi 3328byte ise belleğin başlangıç adresi nedir?
- Belleğin kapasitesi 2 üzeri olarak indislenir toplamı biçiminde yazılarak ikili sayı sistemine dönüştürülür.
- $3328=2048+1024+256=2^{11}+2^{10}+2^8$
- Belirlenen indislerden ikili sayı sistemi oluşturulur. (1101 0000 0000)b
- Belirlenen ikili sayı sisteminden -1 çıkarılır. (1101 0000 0000)b -1=(1100 1111 1111)b=(CFF)h
- Bitiş adresinden kapasitesi çıkarılarak başlangıç adresi bulunur.
- (BCFF)h-(CFF)h=(B000)h

Address Decoding Circuit

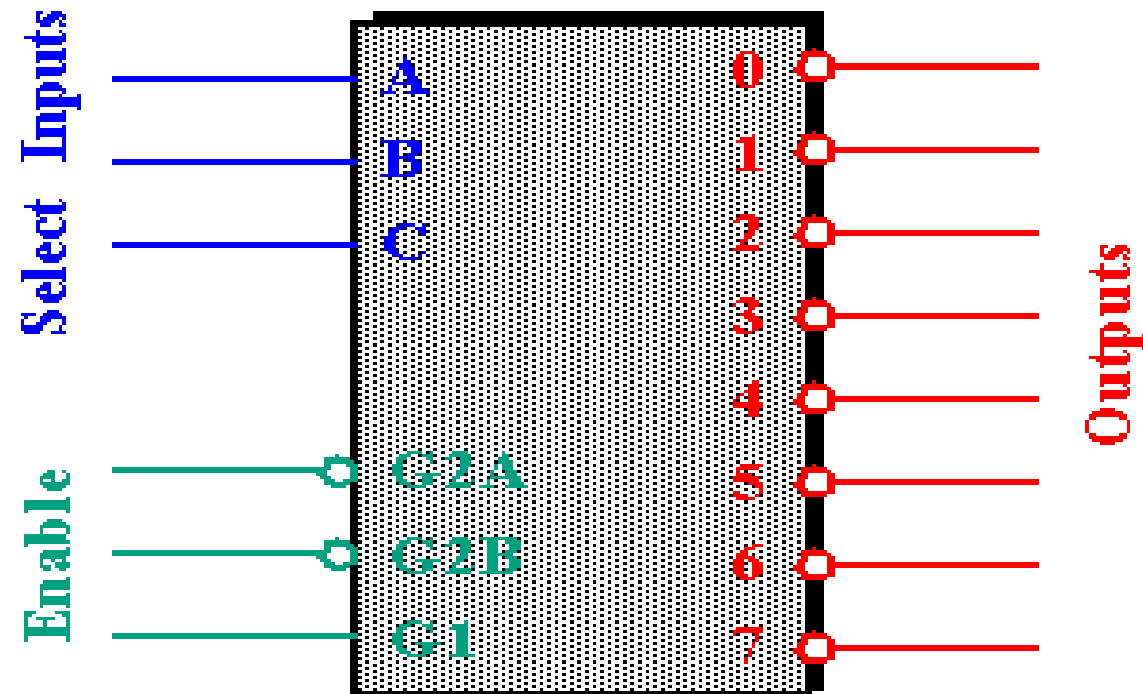
Address Bus Configuration with Address Decoding



Bu örnekte data bus hat sayısı, 8 bit alınacaktır.

- Adres dekoding devresi bellek seçmede kullanılır.
- CPU'dan çıkan toplam adres hattı sayısı=20 adettir.
- CPU'dan çıkan adres hatları indisler= $A_{19}, A_{18}, A_{17}, \dots, A_2, A_1, A_0$
- CPU'nun toplam adresleme kapasitesi= 2^{20} byte=1 Mbyte
- Adres decoding devresine giren adres hattı sayısı=3 adet (A_{19}, A_{18}, A_{17})
- Bellek sayısı= $2^3=8$ adettir. Bir anda bir bellek seçilir.
- Geriye kalan $A_{16}, A_{15}, A_{14}, \dots, A_2, A_1, A_0$ bellek gözü seçmek için belleklere paralel olarak gider.

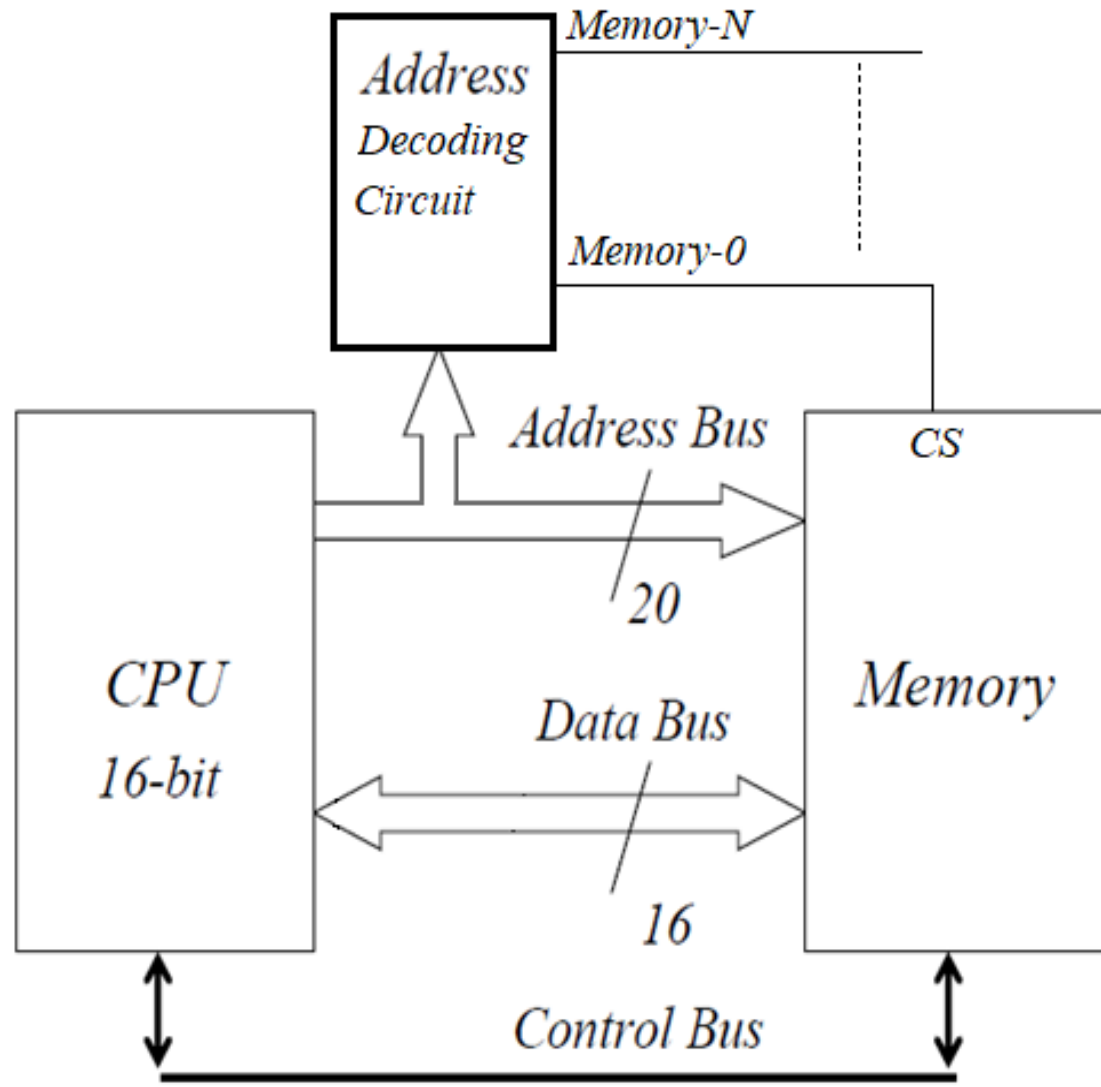
The 3-to-8 Line Decoder (74LS138)



Inputs						Output							
Enable			Select			0	1	2	3	4	5	6	7
G2A	G2B	G1	C	B	A	0	1	2	3	4	5	6	7
1	X	X	X	X	X	1	1	1	1	1	1	1	1
X	1	X	X	X	X	1	1	1	1	1	1	1	1
X	X	0	X	X	X	1	1	1	1	1	1	1	1
0	0	1	0	0	0	0	1	1	1	1	1	1	1
0	0	1	0	0	1	1	0	1	1	1	1	1	1
0	0	1	0	1	0	1	1	0	1	1	1	1	1
0	0	1	0	1	1	1	1	1	0	1	1	1	1
0	0	1	1	0	0	1	1	1	1	0	1	1	1
0	0	1	1	0	1	1	1	1	1	1	0	1	1
0	0	1	1	1	0	1	1	1	1	1	1	0	1
0	0	1	1	1	1	1	1	1	1	1	1	1	0

Figure: The 74LS138 3-to-8 line decoder and function table.

İşlemcinin Organizasyonu: CPU – Bellek Arayüzü



- Adres Bus CPU'dan belleklere ve Address Decoding Devresine giden adres bus'a ait hatlardan oluşur.
- Adres Bus Hatları paraleldir, grup halinde çalışır; her bir hat üzerinde elektriksel sinyal olarak ikili durum (bit) 0 ya da 1 değeri mevcuttur.
- CPU'dan çıkan Adres Bus hat sayısı CPU'nun erişebileceği maksimum bellek kapasitesini verir. Buradaki 20 adet adres bus hat sayısı, $n=20$ ise adreslenecek maksimum bellek kapasitesi $=2^{20} = 1\text{MWord}$ dir. (1Word=16 bit=2Byte)
- Address Decoding Devresi erişilecek bellekleri seçer.
- CPU'dan çıkıp belleklere gelen Adres Bus hat sayısı, belleklerin kapasitesini belirler ve verinin kayıt edildiği bellek gözüne erişimi sağlar.
- Data bus, bellek gözüne yazılıp ya da okunacak veridir. Hatlar paraleldir ve grup olarak çalışır. Her bir hat üzerinde 0 ya da 1 ikili durum mevcuttur.
- Data Bus hat sayısı CPU'nun veri transfer özelliğini verir. CPU, 16 bit dendiğinde Data Bus hat sayısı 16 olan bir CPU'dan bahseder.
- Word olarak 16 bit yazılıp okunur. 2 Byte'a denk gelir.
- Control bus hatları da paraleldir. Grup olarak çalışmaz. Ayrık çalışır. Her bir hat üzerinde 0 ya da 1 ikili durum mevcuttur.

Örnek: CPU ve Sistem Bus

Address Bus: 34 bit , Data Bus: 16bit (Word)

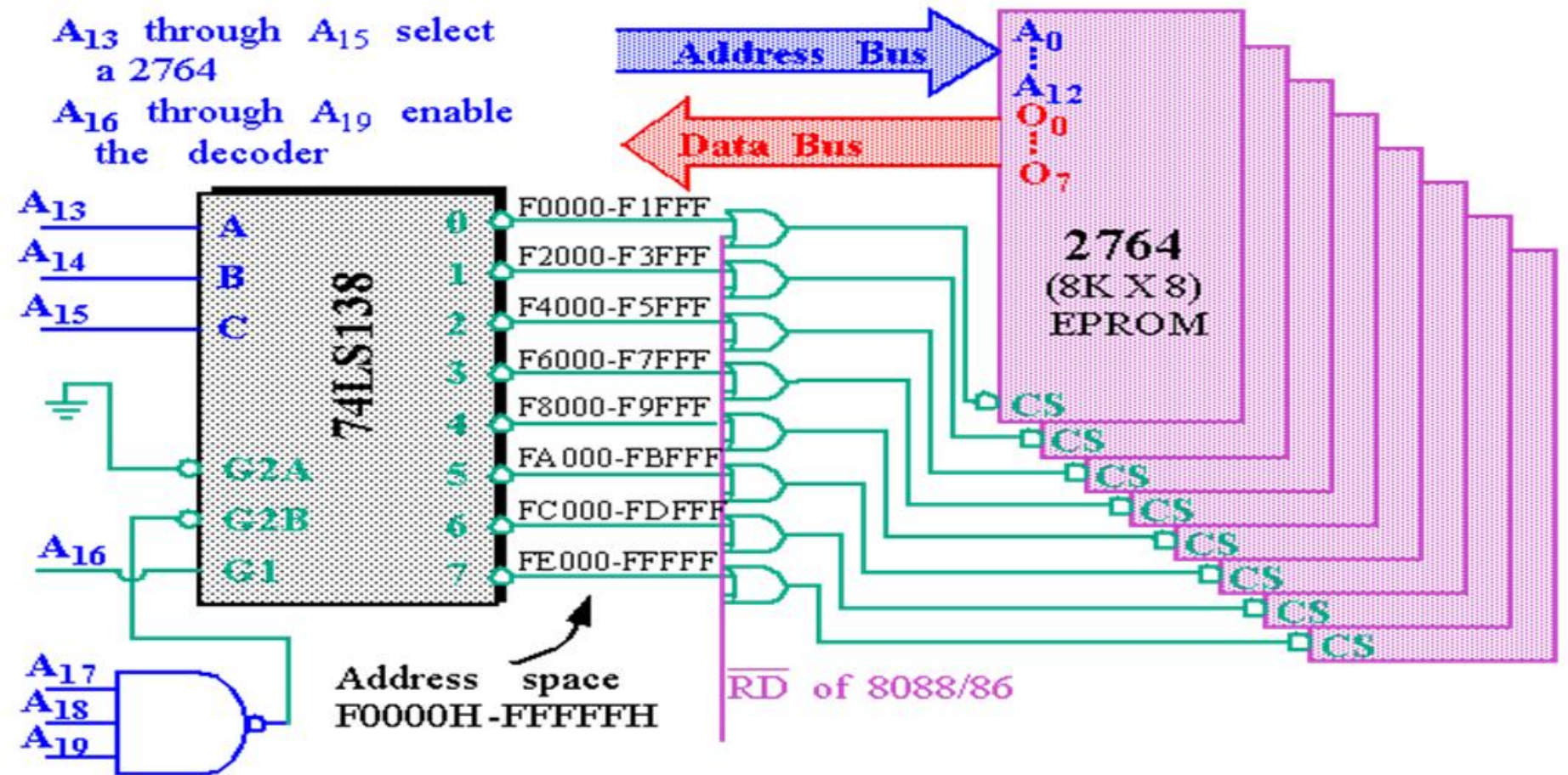
- Number of wires of Adress Bus: 34; (Every on wire is bit:0 or 1, Hatlar, paralel ve grup olarak çalışırlar)
- Number of wires of Data Bus: 16; (Every on wire is bit: 0 or 1 Hatlar, paralel ve grup olarak çalışırlar)
- Total Accessable Memory Capacity: 2^{34} Word
- Memory Capacity: $2^{34} = 2^4 \cdot 2^{30} = 16$ Word
- Index of Address bus: A33 A0
- Index of Data Bus: D15 D0

A circuit that uses eight 2764 EPROMs for a 64K × 8 section of memory in a microprocessor-based system.

Toplam kaç adet bellek var? 8 adet.

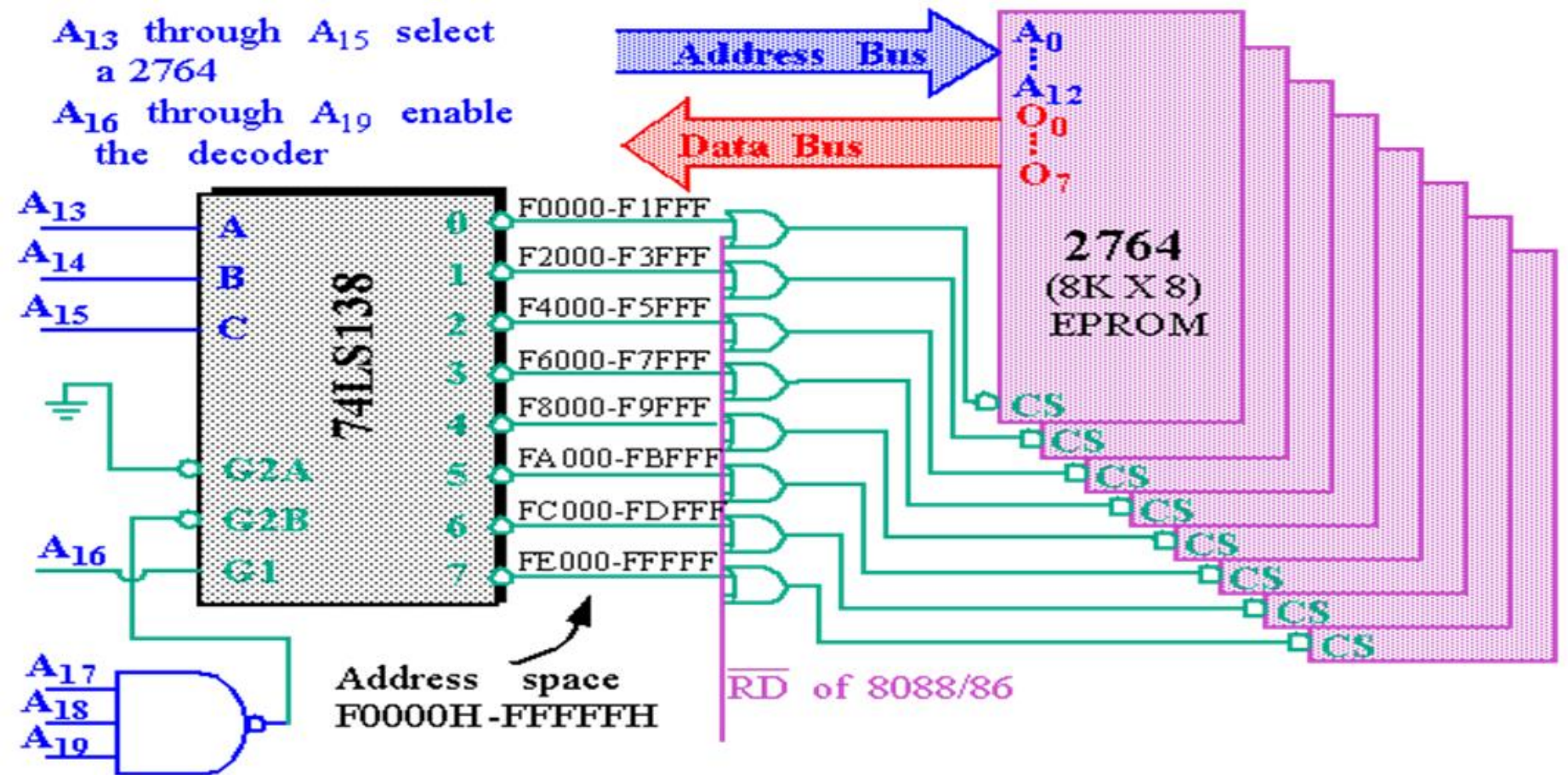
Adres decoding devresinin çıkış 8 adet ise girişine CPU dan kaç adet adres bus hattı gelir? $2^3=8$, 3 adet. İndisleyin: A15, A14, A13.

Her bir belleğe CPU'dan A13, A12, ..., A1, A0 gelir.



A circuit that uses eight 2764 EPROMs for a 64K × 8 section of memory in a microprocessor-based system.

- The addresses selected in this circuit are F0000H–FFFFFFH.
- ROM bellek, sadece okunan bellek olduğundan Data Bus CPU'ya doğru tek yönlüdür.
- 64K x 8 ifadesinde 8'in anlamı nedir? 8: data bus hat sayısını gösterir. Hat üzerindeki değer bit (0 ya da 1). Elektriksel sayısal sinyal vardır.
- 64K x 8 ifadesinde 64K'nın anlamı nedir? 64Kbyte, bellek toplam kapasitesidir. Kaç byte eder? $2^6 \times 2^{10} = 2^{16}$ byte. Adres hattının sayısı =16 dır. İndisleyin: A15, a14, A13,, A1, A0
- 8K x 8 ifadesinde 8K'nın anlamı nedir? 8Kbyte, bellek toplam kapasitesidir. Kaç byte eder? $2^3 \times 2^{10} = 2^{13}$ byte. Adres hattının sayısı =13 dır. İndisleyin: A12, a11, A10,, A1, A0



Address Decoding Circuit

- CPU'nun okuyup ya da yazacağı belleği seçer
- Prevent overlap
- Input of decoding: CPU- address bus: m wires
- Output of decoding: 2^m
- cs, en, Address decoding enable.

Example: 10 adet belleğim var. CPU dan Address decoding devresinin girişi için kaç adet adres hattına ihtiyaç var?

$m=?$, $m=4$, $10 \rightarrow 16=2^4$

Addressing of the Memory

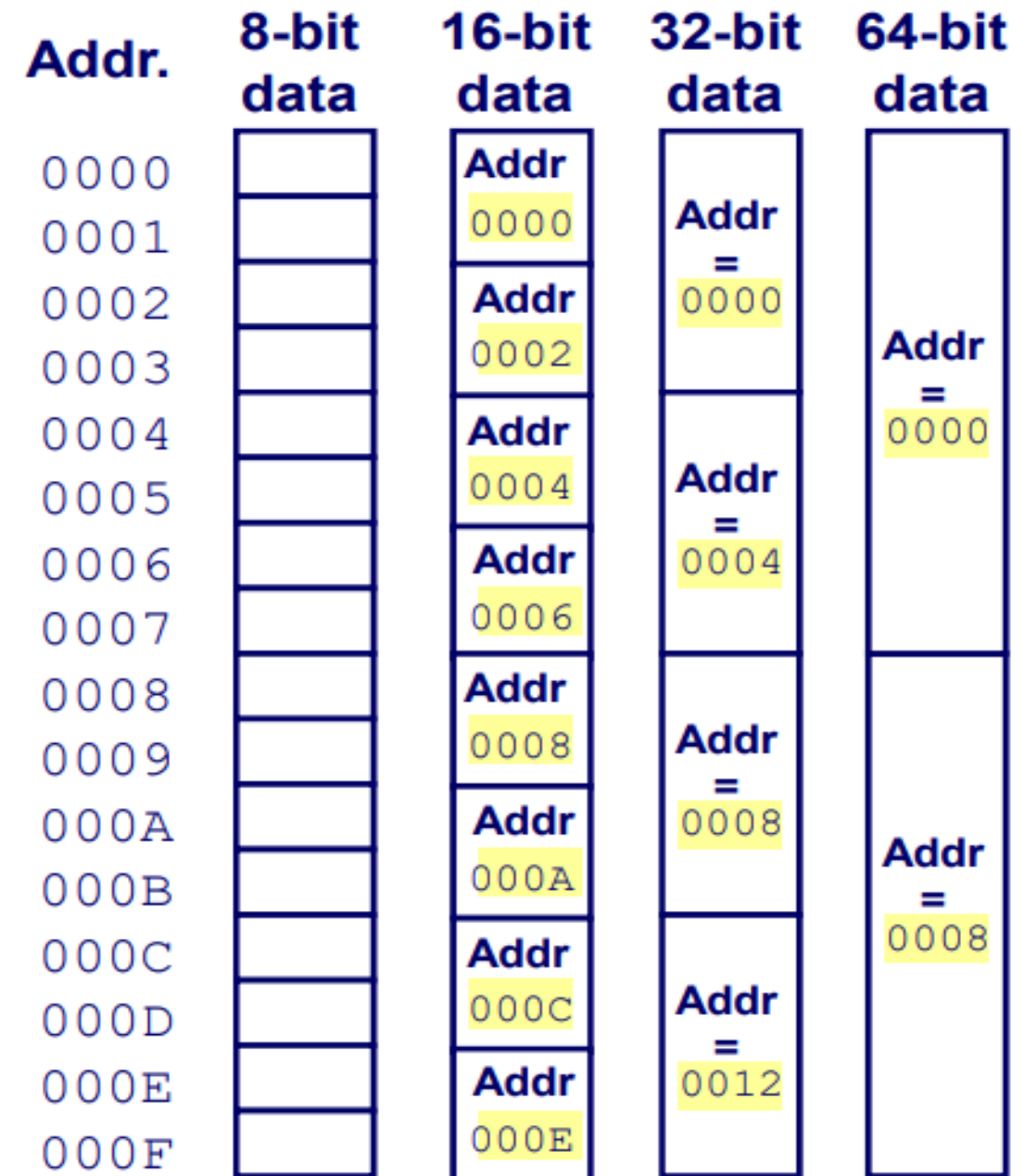
- The total capacity of a memory is calculated by using the number of wires on address bus: 2^n
- The number of wires on address bus is n
- Addressing index of the memory: $A_{n-1} A_{n-2} \dots A_2 A_1 A_0$
- The number of wires of chip select input for decoding: m
- The number of wires of chip select output for decoding: 2^m , also,
 $2^m \geq$ the number of the rams

CPU'dan çıkacak toplam adres hattı sayısı=n+m

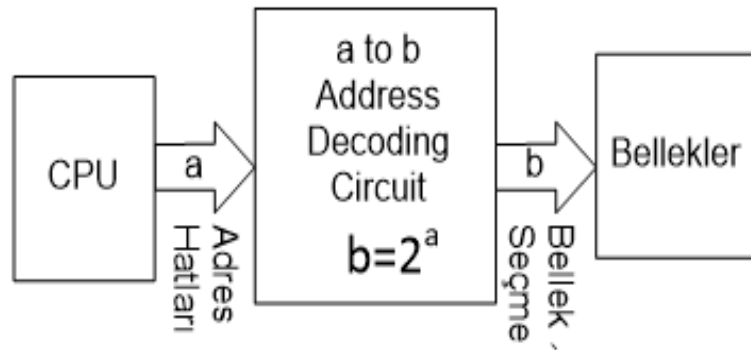
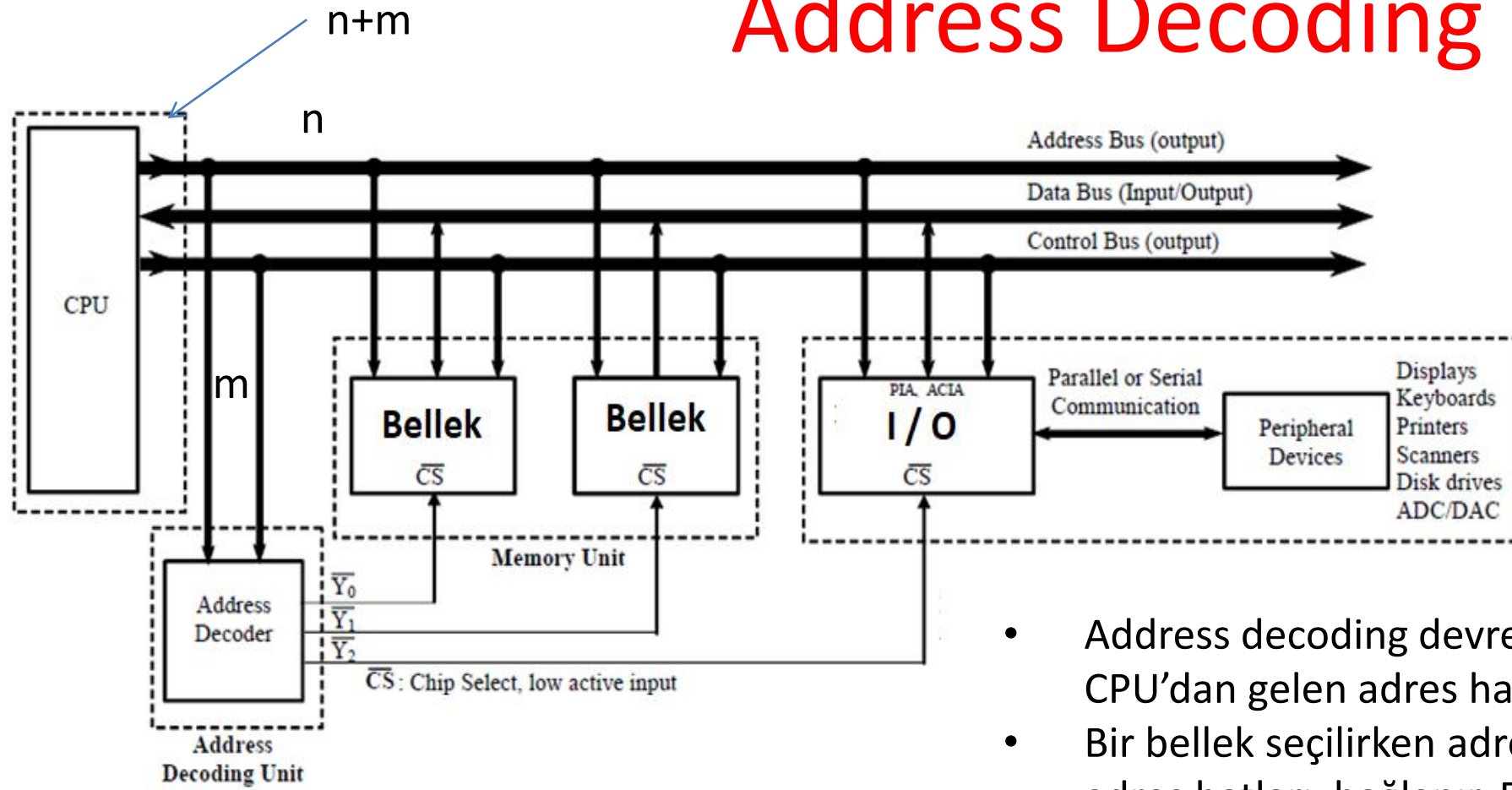
- Overlap: It must be written only selected memory cell. The decoding prevents from overlaps.

Address of Multi-byte Data

- Every byte has a unique address
- *So, if data spans multiple bytes, what is address?*
- Data always addressed by its lowest address
 - address of first byte in memory
- **Alignment**
 - Data elements are aligned by size
 - for a primitive (single datum) with K bits, address must be multiple of K
 - **chars, booleans** at any address
 - **shorts** at even addresses
 - **ints, floats, pointers** every 4th addr
 - **doubles** every 8th address
 - etc.
 - Arrays, structures, and classes
 - alignment determined by size of largest primitive (single datum)



Address Decoding Unit



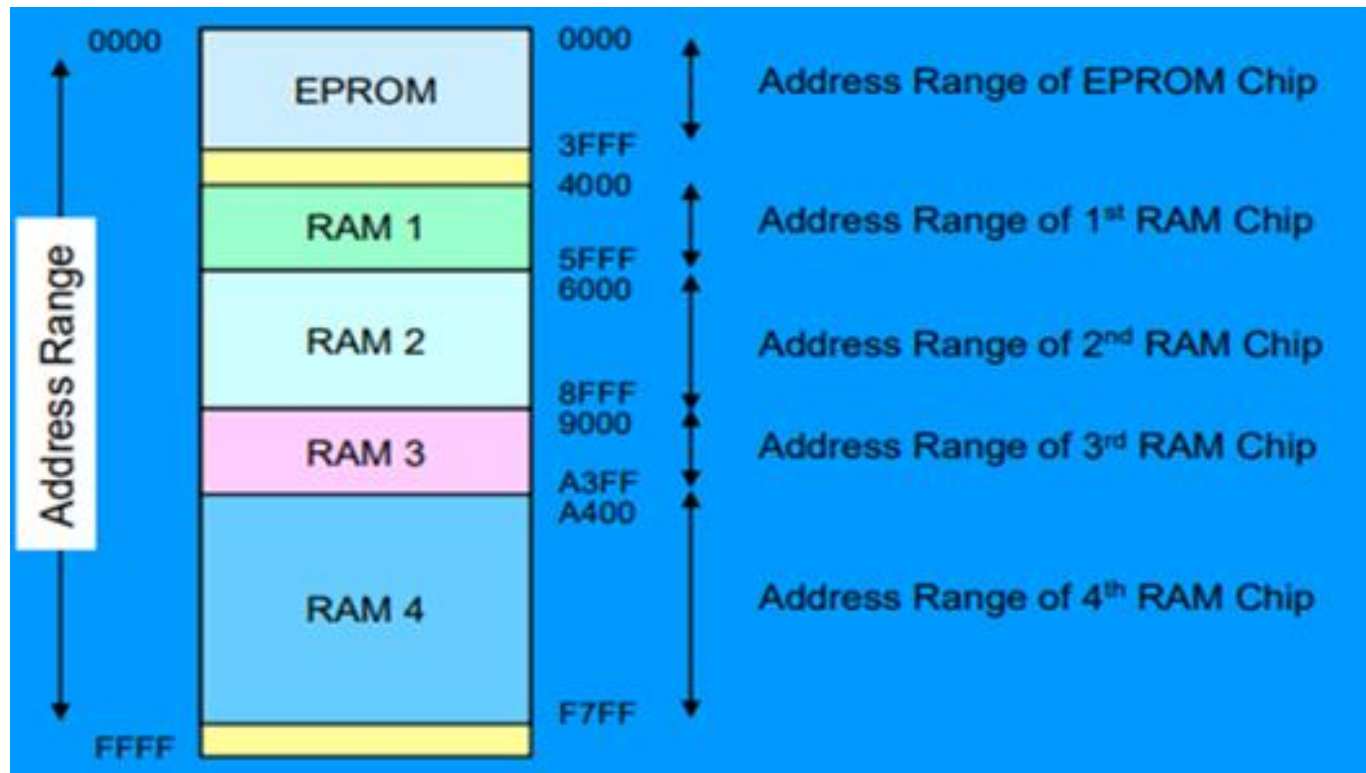
- Address decoding devresinin çıkışları bellek seçer. Seçilen belleğin gözüne CPU'dan gelen adres hatları yardımıyla erişilir.
- Bir bellek seçilirken adres decoding devresinin girişine CPU'dan gelen adres hatları bağlanır. Böylece adres decoding devresinin çıkışları bellekleri seçmede kullanılır.
- Seçilecek bellek sayısı=Address decoding devresi çıkış sayısı= 2^m dir. Burada m CPU'dan address decoding devresinin girişine gelen adres hattı sayısıdır.
- Herbir belleğin kapasitesini belirleyen (Kapasite= 2^n) n adet hat CPU'dan gelir.
- Amaç aynı anda bir belleğin ilgili veri gözününün seçilmesidir.
- Adres decoding devresinin çıkış sayısı bellek sayısına eşit ya da büyük olmak zorundadır. (2'nin üssüne eşit olmayan bellek sayılarında)

Örnek: Address Decoding Devresi

- Seçilecek bellek sayısı=Address decoding devresi çıkış sayısı= 2^m dir.
- Örnek: 172 adet bellek var ise CPU'dan gelecek adres hattı sayısı kaçtır? $2^m \geq 172$ ise $m=8$ alınır. Çünkü $2^7 = 128$ dir. Bellek sayısına eşit ya da büyük olan seçilir.
- O halde CPU'dan Address Decoding Devresinin girişine gelecek adres hattı hat sayısı=8 dir. Bu durumda seçilecek bellek sayısı 256 olur. ($256 - 172=84$ adet yedek)
- Bellekler, Address Decoding Devresi yardımıyla seçilir. Yazılıp ya da okunacak belleğin ilgili gözü ise CPU dan belleklere paralel gelecek n adres hatları ile belirlenir. Bu durumda seçilen belleğin kapasitesi= 2^n byte olur.

Memory Map and Addresses

- The memory map is a picture representation of the address range and shows where the different memory chips are located within the address range.



Soru: Toplam bellek haritalam kapasitesi nedir: (0000)h – (FFFF)

Adres hat sayısı, $n=4*4=16$

Kapasite= $2^{16}=64*2^{10}$ byte=64kbyte

Eprom kapasite=Bitiş adresi -Başlangıç adresi +1

Eprom kapasite=(3fff)h-(0000)h+1=(3fff)h+1=(4000)h=
(0100 0000 0000 0000)b= 2^{14} byte=16Kbyte

Ram1 kapasite=Bitiş adresi -Başlangıç adresi +1

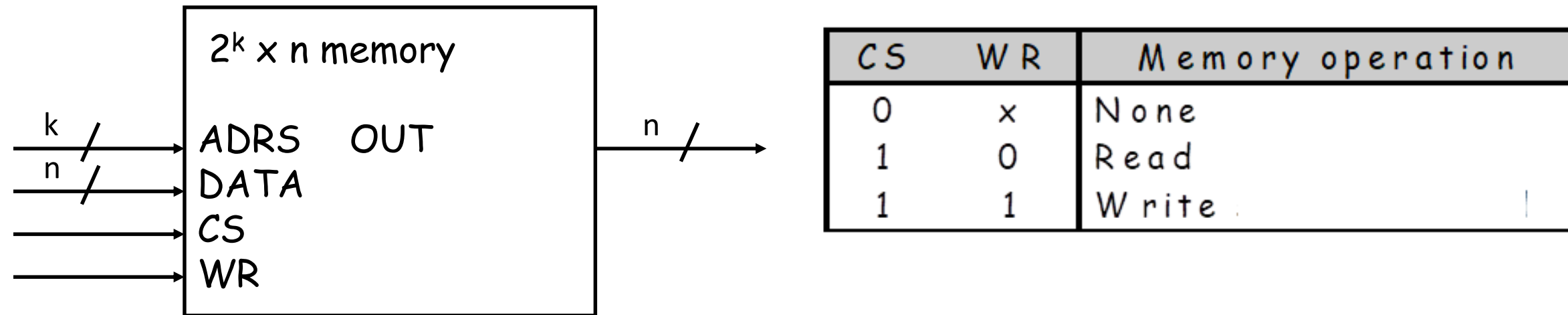
Ram1 kapasite=(5fff)h - (4000)h+1 =(1fff)h+1=(2000)h=
(0010 0000 0000 0000)b= 2^{13} byte=8Kbyte

Ram2 kapasite=Bitiş adresi -Başlangıç adresi +1

Ram2 kapasite=(8fff)h - (6000)h+1 =(2fff)h+1=(3000)h=
(0011 0000 0000 0000)b= $2^{13} + 2^{12}=2^{12}(2+1)=12$ Kbyte

Memory Organization

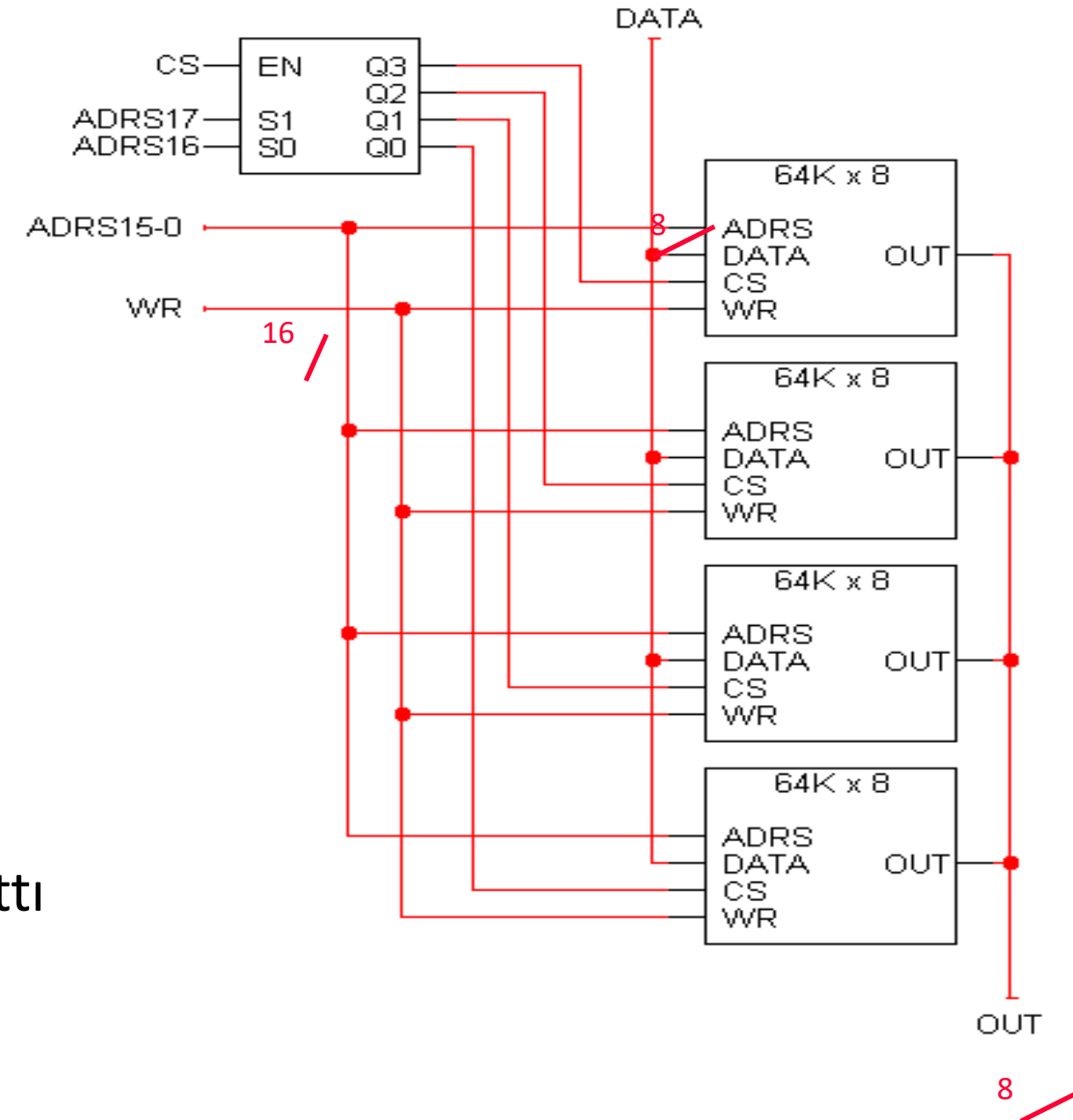
Block diagram of RAM



- This block diagram introduces the main interface to RAM.
 - A Chip Select, **CS**, enables or disables the RAM. (CS, Adres Dekoding devresinden gelir)
 - **ADRS** specifies the address or location to read from or write to. Adres Bus CPU'dan gelir.
 - **WR** selects between reading from or writing to the memory. WR (1:Write,0:Read)
 - ▶ To read from memory, WR should be set to 0.
OUT will be the n-bit value stored at ADRS. Çıkış data bus'dır.
 - ▶ To write to memory, we set WR = 1.
DATA is the n-bit value to save in memory. (Data Bus CPU'dan gelir.)
- This interface makes it easy to combine RAMs together, as we'll see.

Analyzing the 256K x 8 RAM

- There are 256K words of memory, spread out among the four smaller 64K x 8 RAM chips.
- 4 adet bellek var. Toplam kapasite= $4 * 64K = 256K$
- Data bus hat sayısı nedir? 8 adet
- Herbir bellek 64K ise adres hattı sayısı nedir?
 $64K = 2^6 * 2^{10} = 2^{16}$ byte ise adres hattı sayısı=16 adettir.
- Adres hatlarını indisleyin: A15, A14, ..., A1, A0
- Adres decoding devresini çıkış kaç adet olmalıdır.
 - Adres decoding devresini çıkış sayısı \geq bellek sayısı
 - Adres decoding devresini çıkış sayısı, $m=4$ adet
- Adres decoding devresi giriş sayısı= k ; $m=2^k$
 - Adres decoding devresi giriş sayısı=2
- Adres decoding devresinin girişleri CPU'dan gelir. Adres hattı indisleri: A17, A16 dır. (belleklerin maksimum adres indisinden devam eder.)



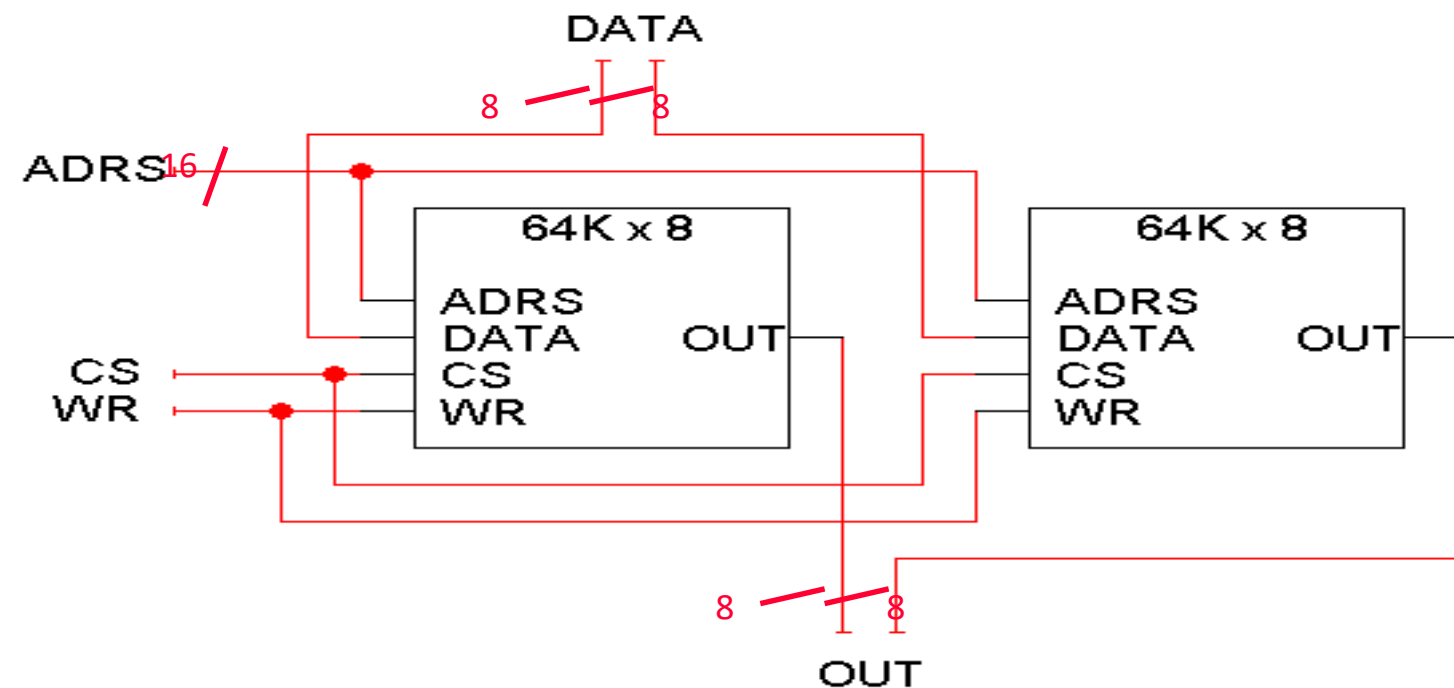
Making a wider memory

- You can also combine smaller chips to make wider memories, with the same number of addresses but more bits per word.

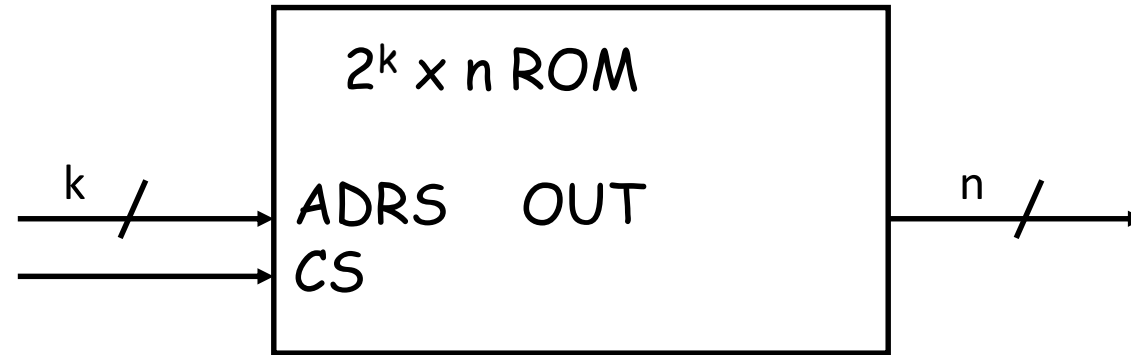
- **Example**

Design a 64K x 16 RAM, using two 64K x 8 chips.

- The left chip contains the most significant 8 bits of the data.
- The right chip contains the lower 8 bits of the data.

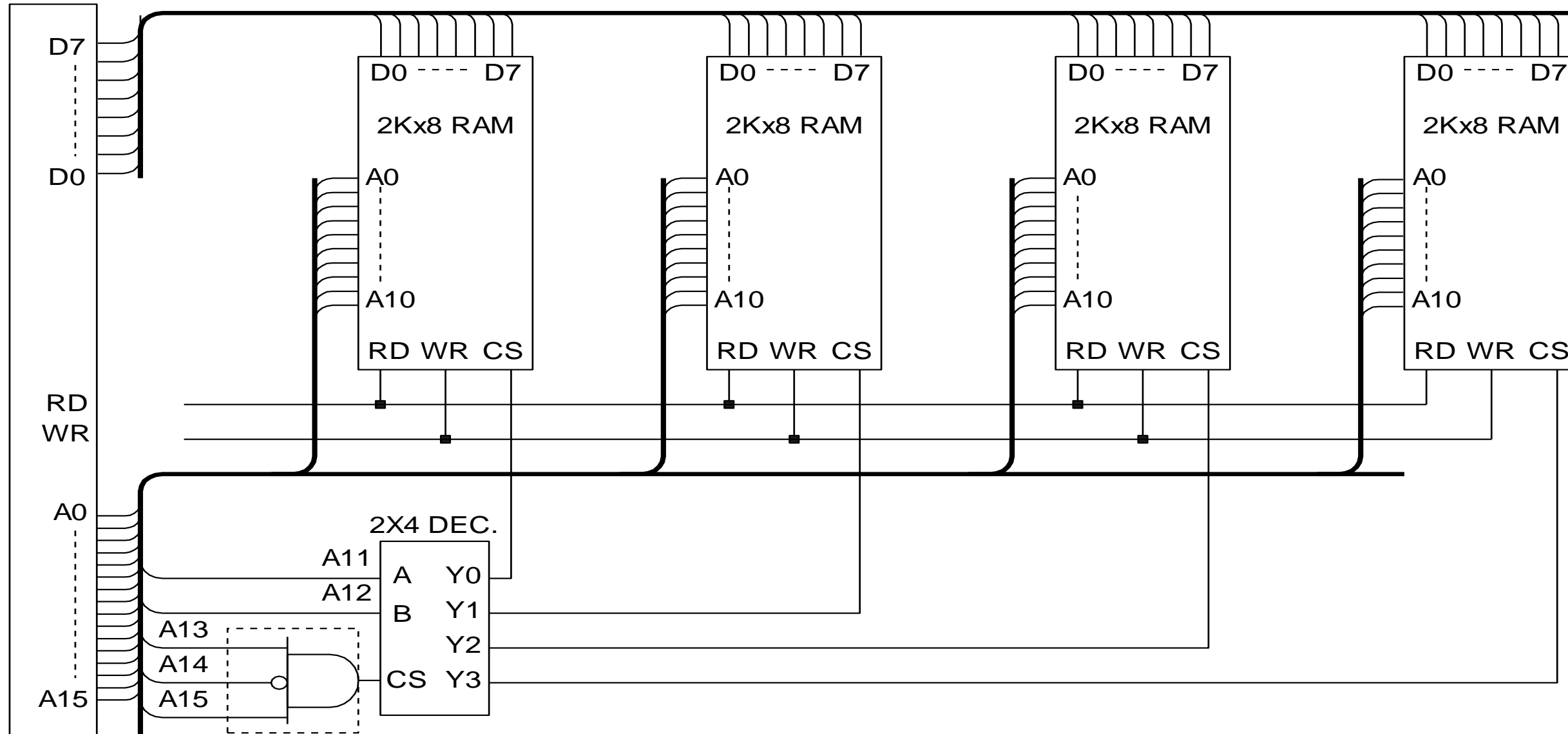


Read-only memory

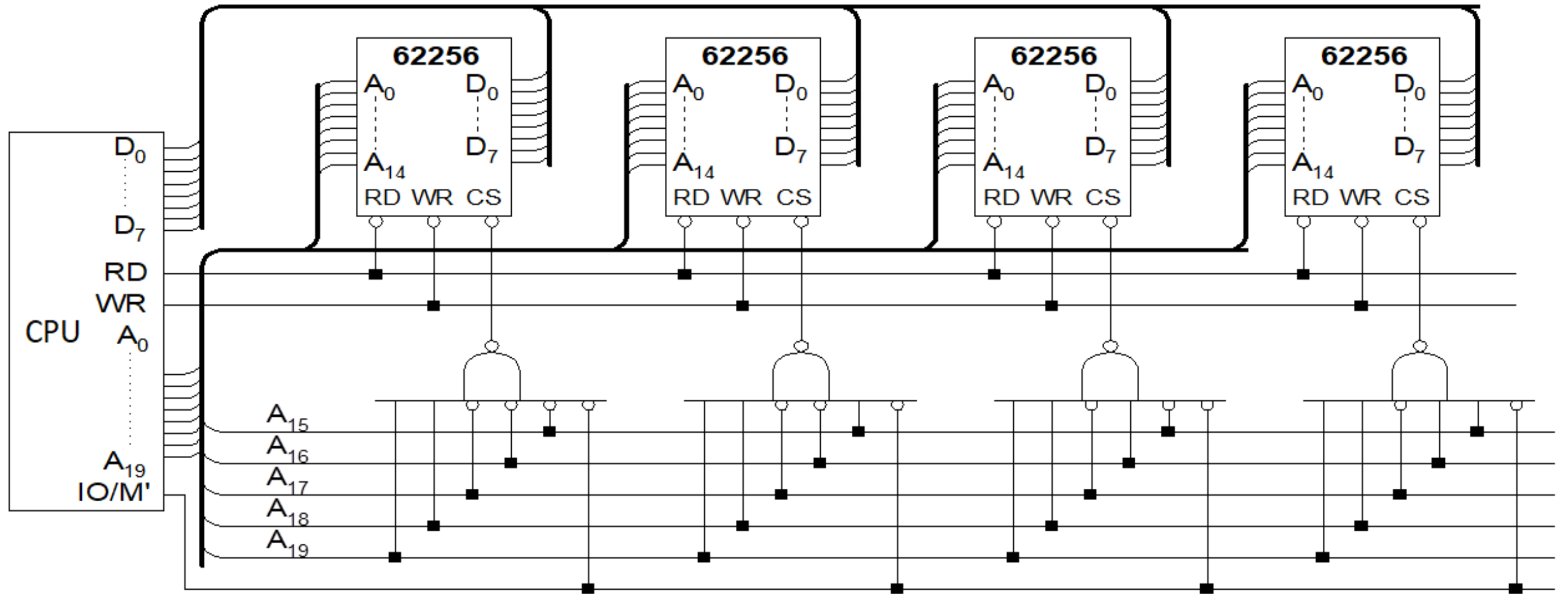


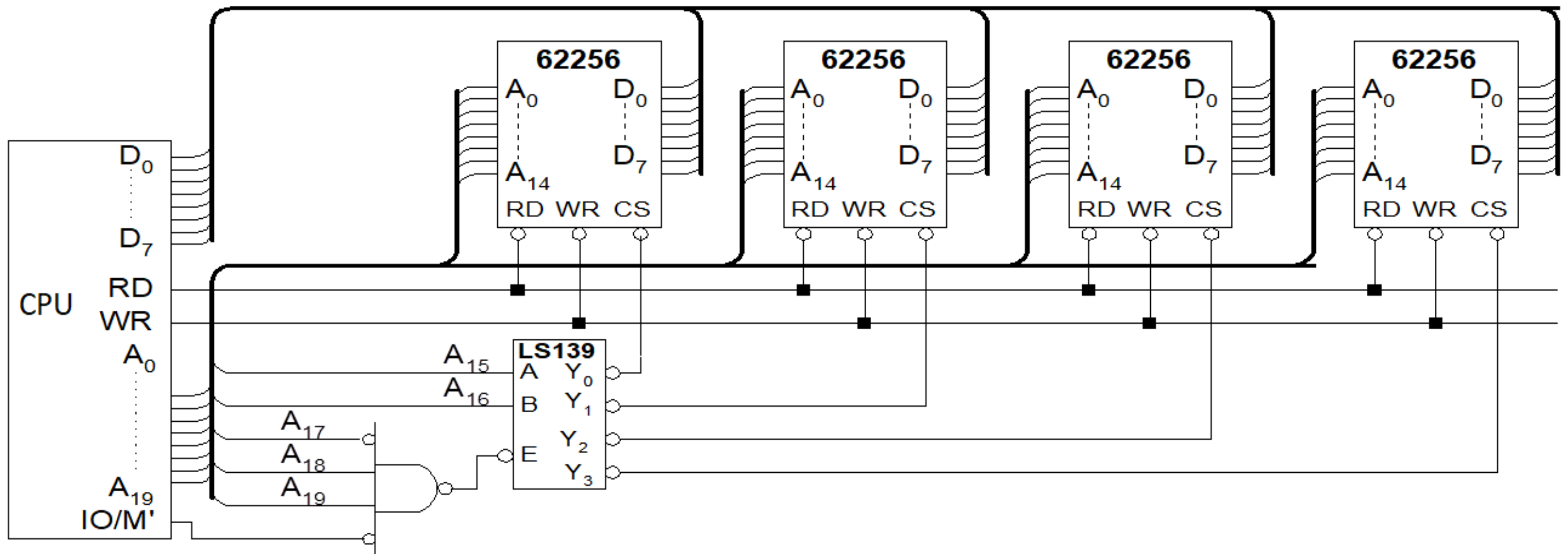
- A **ROM**, is a special kind of memory whose contents cannot be easily modified.
 - The **WR** and **DATA** inputs that we saw in RAMs are not needed.
 - Data is stored onto a ROM chip using special hardware tools.
- ROMs are useful for holding data that never changes.
 - Arithmetic circuits might use tables to speed up computations of logarithms or divisions.
 - Many computers use a ROM to store important programs that should not be modified, such as the system BIOS.
 - Game machines, cell phones, vending machines and other electronic devices may also contain non-modifiable programs.

Circuit Diagram



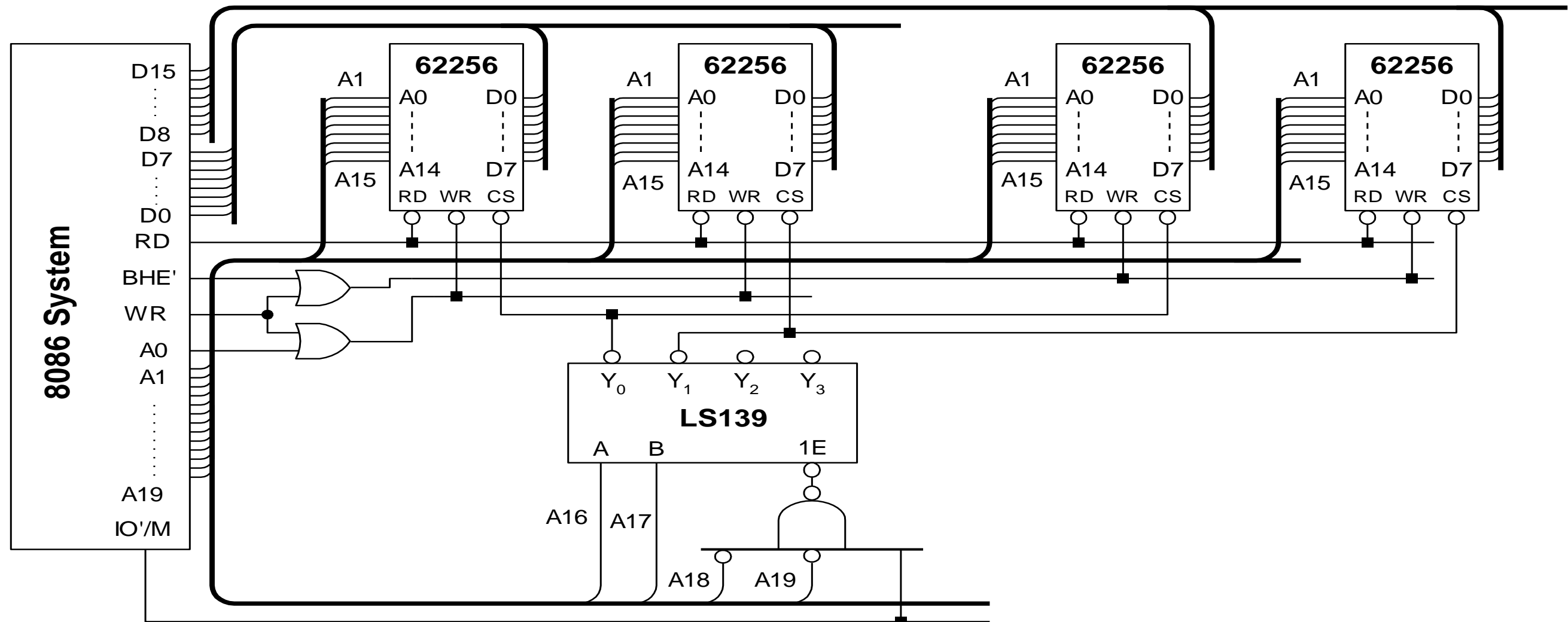
- A15, A14, A13: Adres Decoding Birimini seçer.
- A12, A11: Adres Decoding devresinin girişidir 4 bellekten birini seçer.
- A10, ..., A0: Seçilen belleğin bellek gözünü seçer
- D7, ..., D0: Data bus hattır. Byte olarak bellekере veri yazar ya da okur.
- RD, WR kontrol sinyalleri veriyi yazmada ya da okumada kullanılır.





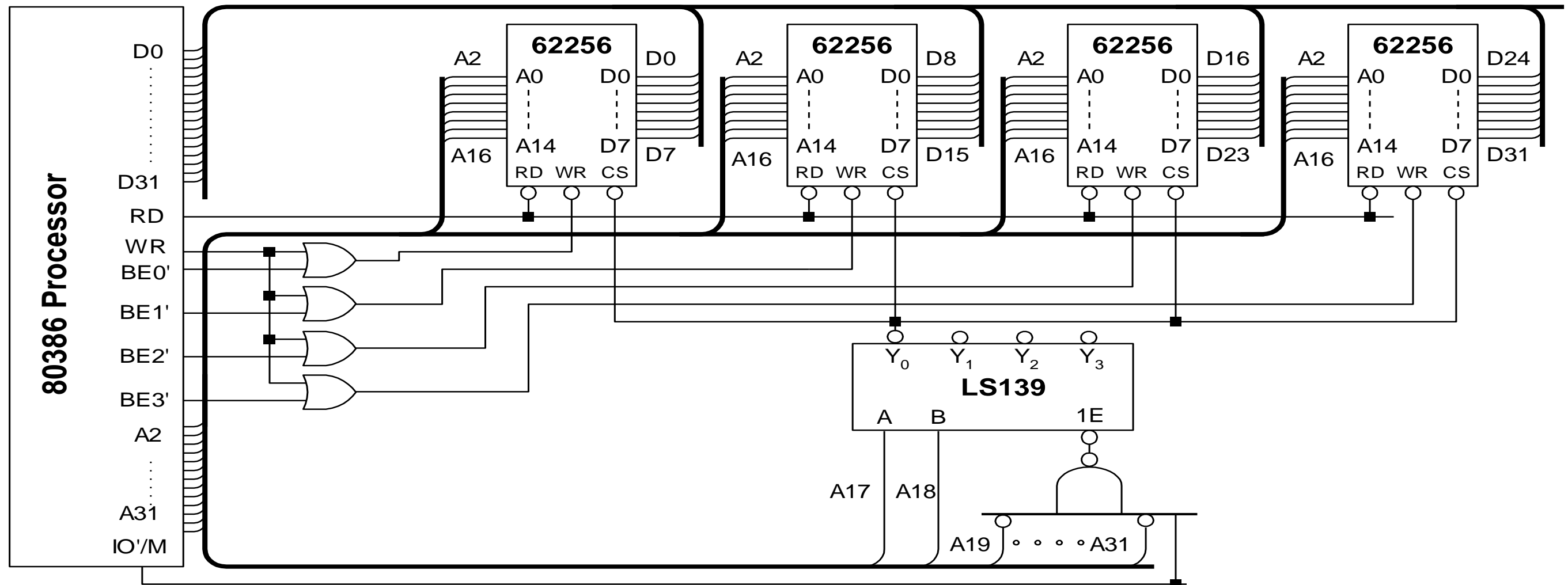
16-bit Memory Interfacing using separate bank write signals

- With this method the decoder always enables both banks.
- On a memory read operation, the data from both banks is loaded on the data bus. The microprocessor selects internally the appropriate bank, according to the instruction being executed.
- On a memory write operation, only the WR signal of the appropriate bank is enabled, thus data is copied only in the appropriate memory chip.



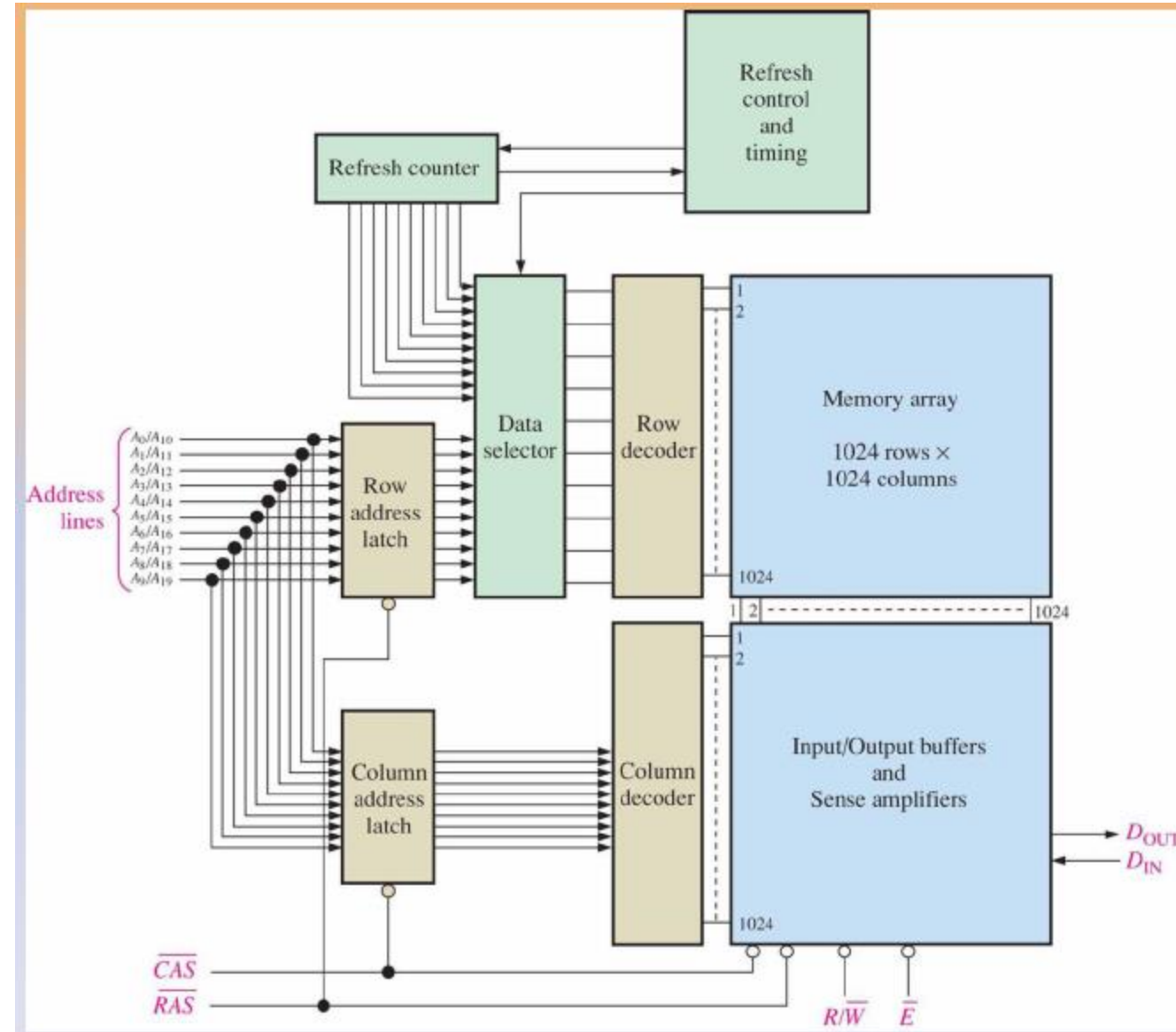
32-bit Memory Interfacing using separate bank write signals

- The 80386 microprocessor has four bank enable signals to select one out of 4 memory banks. The address lines A0 and A1 are not available.
- On a memory read operation, the data from all banks is loaded on the data bus. The microprocessor selects internally the appropriate bank, according to the instruction being executed.
- On a memory write operation, only the WR signal of the appropriate bank is enabled, thus data is copied only in the appropriate memory chip.

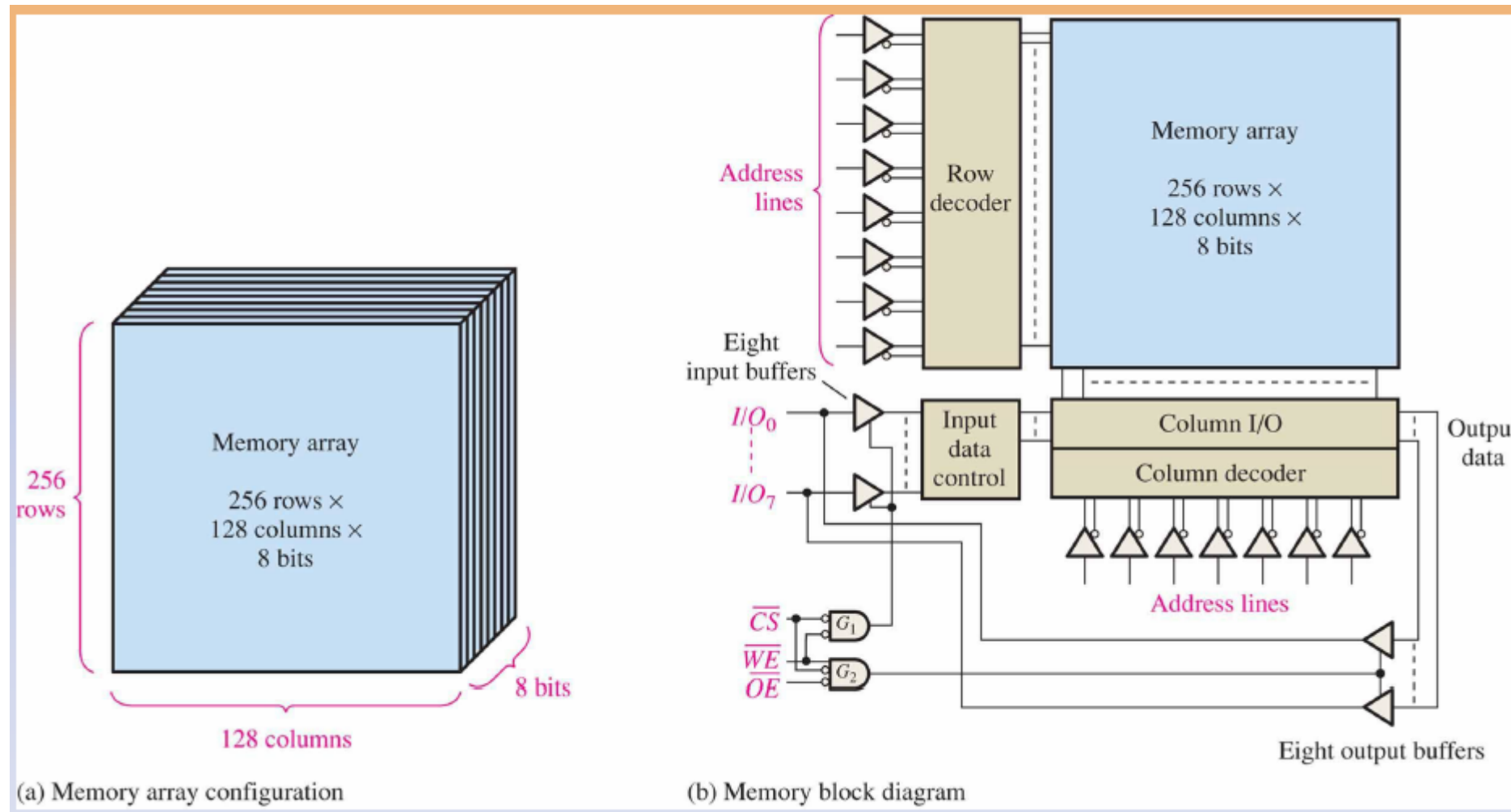


RAM Memory Organization

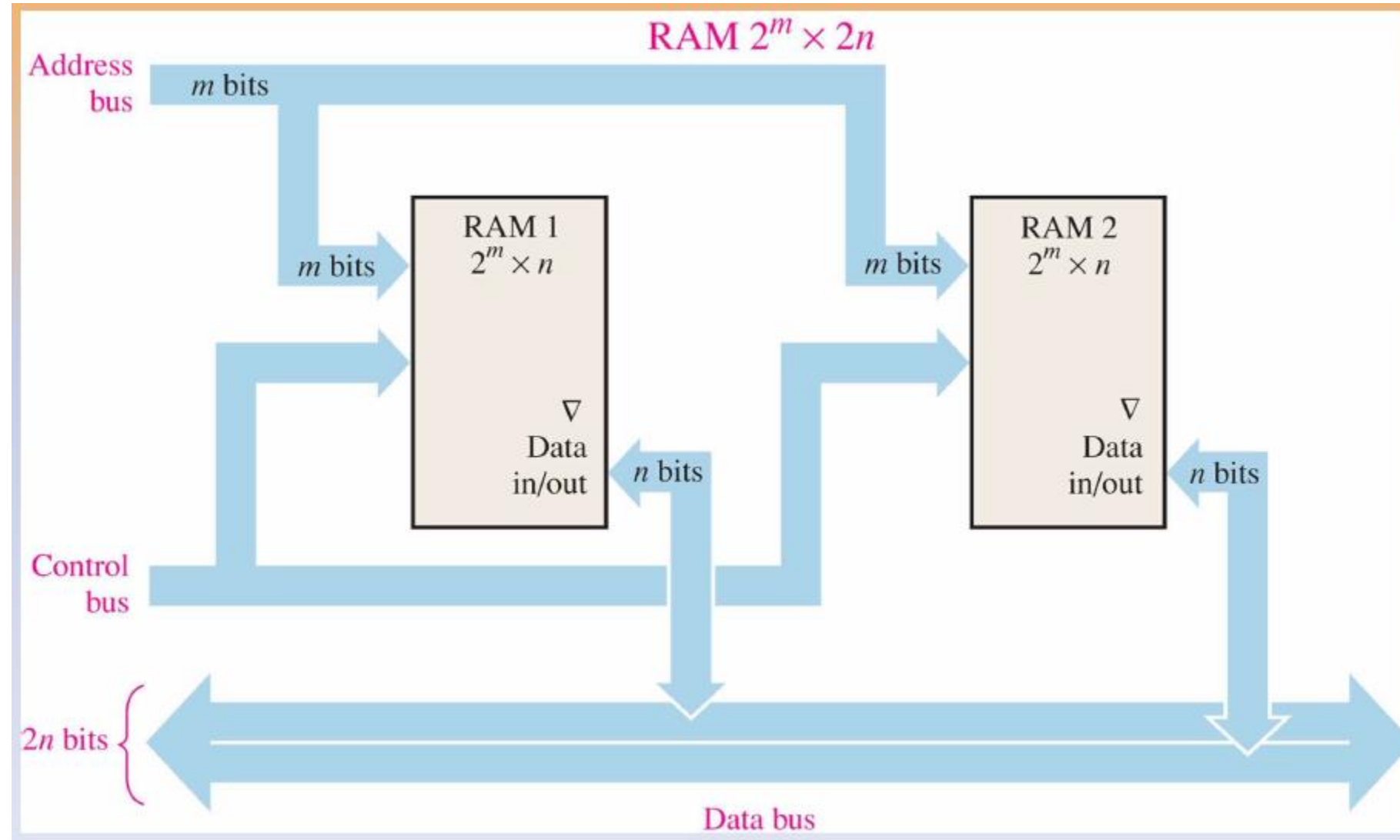
1M x 1 DRAM'in Basitleştirilmiş iç yapısı



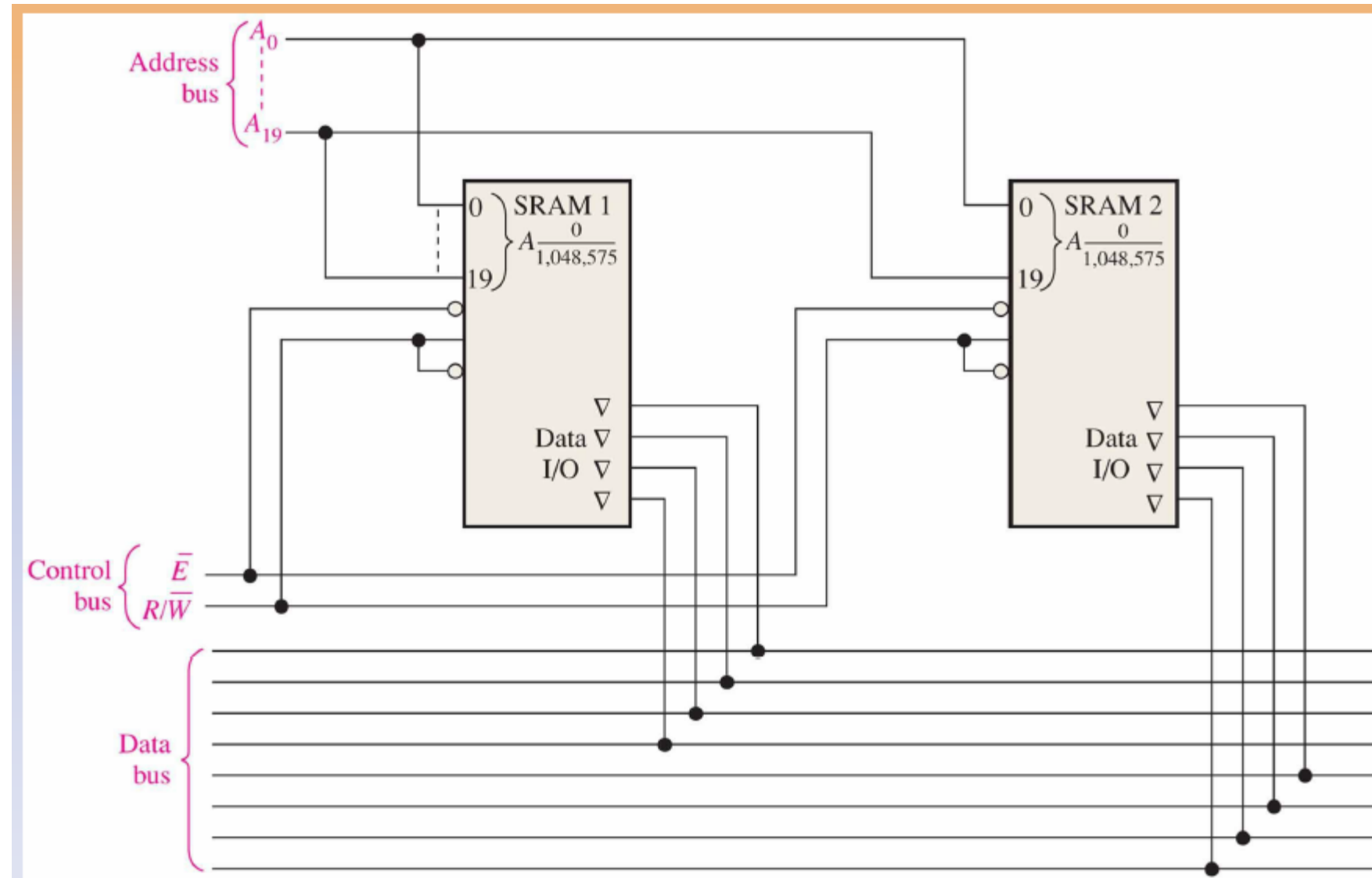
32k x 8 SRAM belleğin Organizasyonu



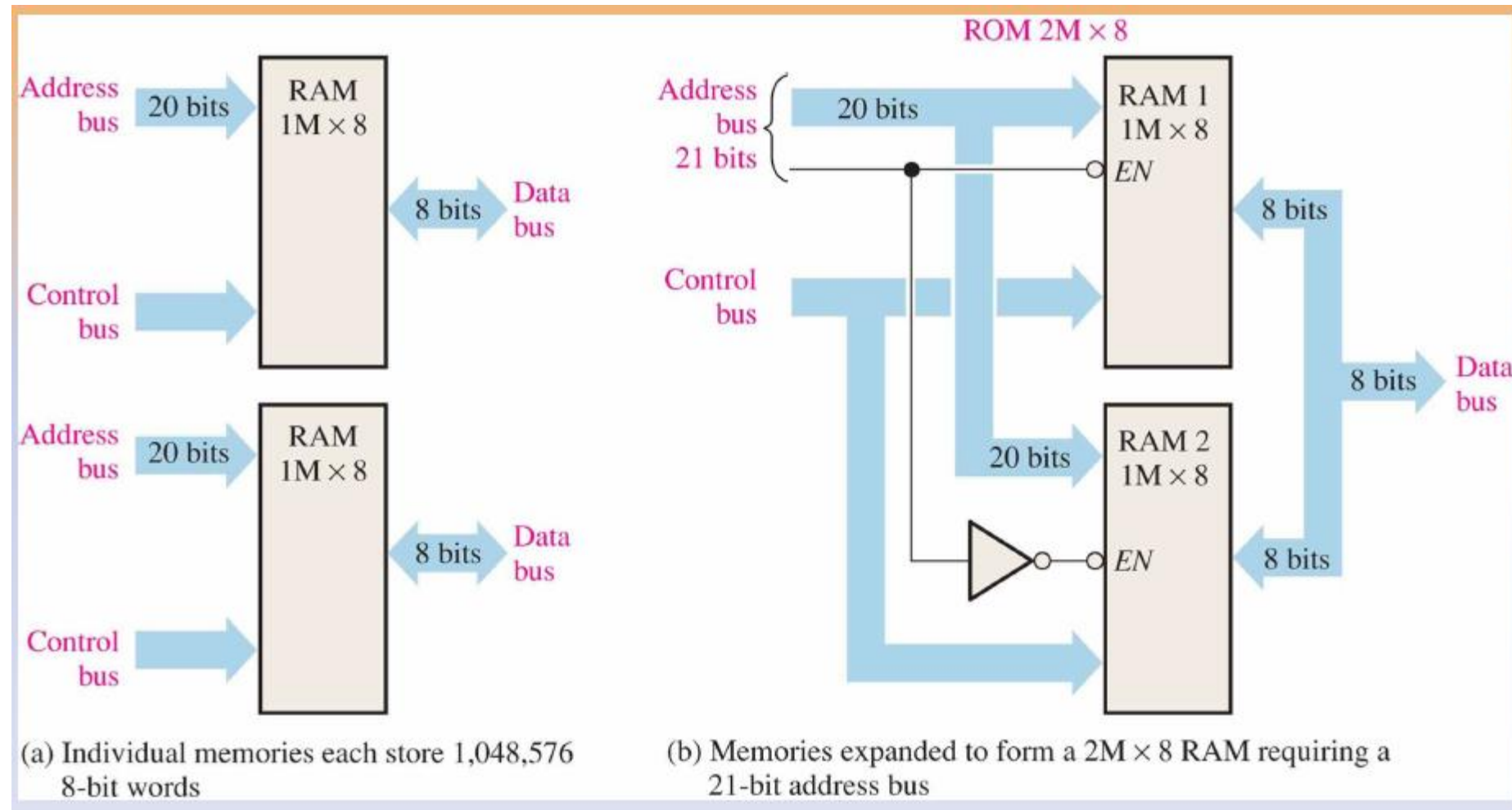
RAM Belleğin Kelimesesinin Genişletilmesi



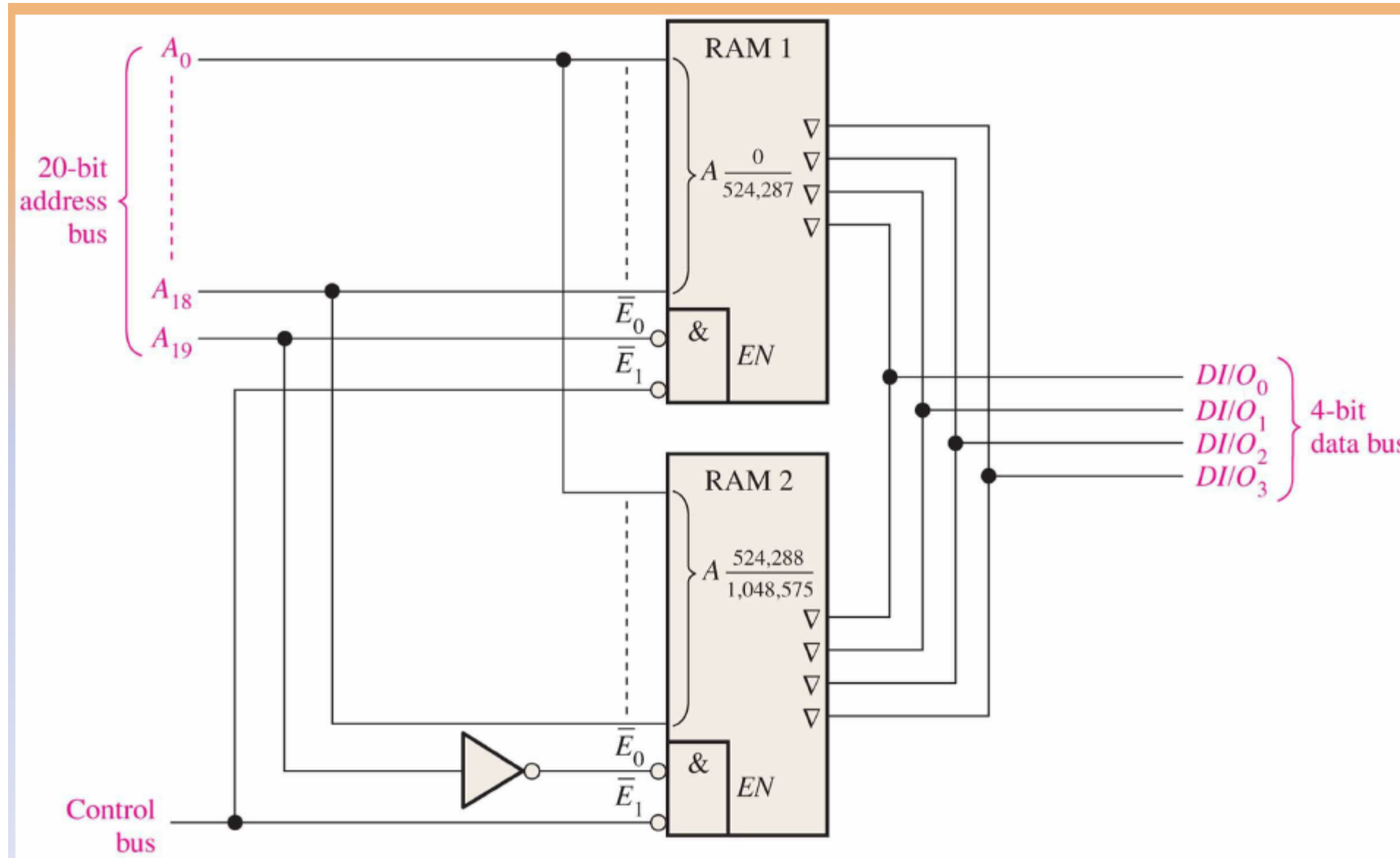
1MX4 RAM bellekten 1MX8 bellek modülünün elde edilmesi.



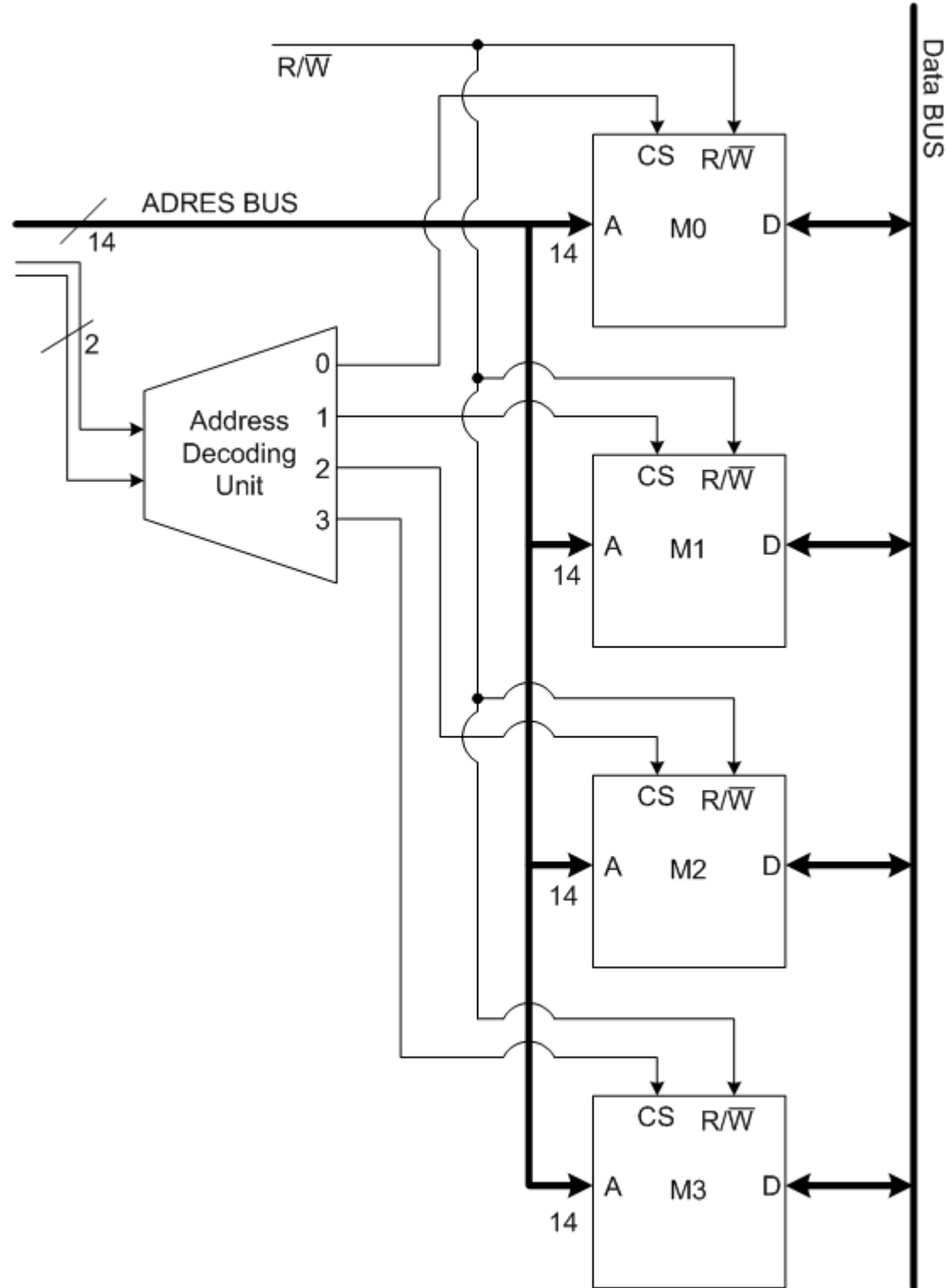
Kelime kapasitesini arttırma



Kelime kapasitesini arttırma



Örnek:



- Herbir belleğin kapasitesini bulunuz.
- Belleklere gelen adres hatlarını indisleyiniz.
- Toplam Bellek kapasitesini bulunuz.
- CPU'dan çıkacak adres hatlarını indisleyiniz.
- Address decoding birimine giriş yapan adres hatlarını indisleyiniz.
- 54A3Hex adresi seçilirse hangi bellek aktif olur?

Yanıt

- 2^{14} byte=16Kbyte
- A13, A12, A11, ..., A1, A0
- 2^{16} byte=64Kbyte
- A15, A14, A13, ..., A1, A0
- A15, A14
- 54A3Hex, ifadesinde 16 adet bit bulunmaktadır.

(A15, A14, A13, ..., A1, A0)_b = 54A3Hex = (0101 0100 1010 0011)_b

Bu durumda A15, A14 giriş değerlerine göre Address Decoding Biriminin çıkış değerleri:

(00)_b = 0

(01)_b = 1

(10)_b = 2

(11)_b = 3

(A15, A14)_b = (01)_b = 1 olduğunu M1 belleği aktif olur.

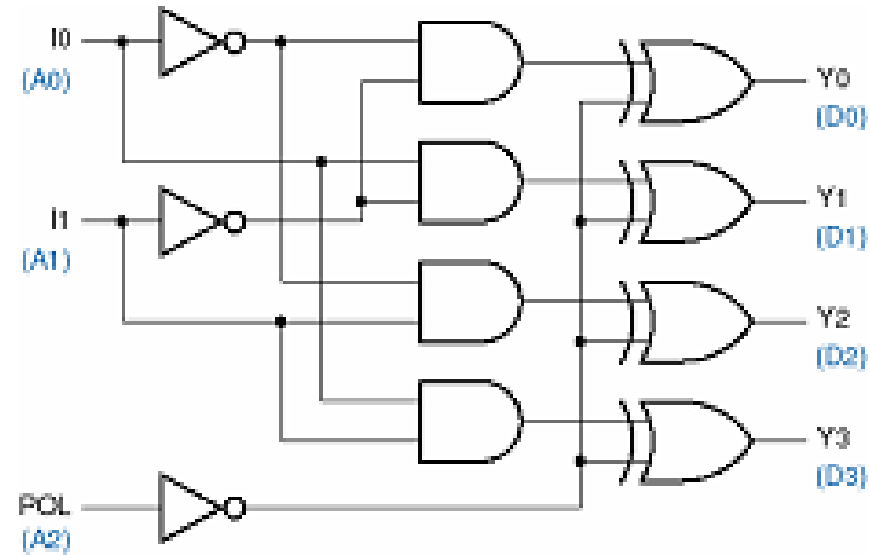
ROM Memory Organization

Memories and functions

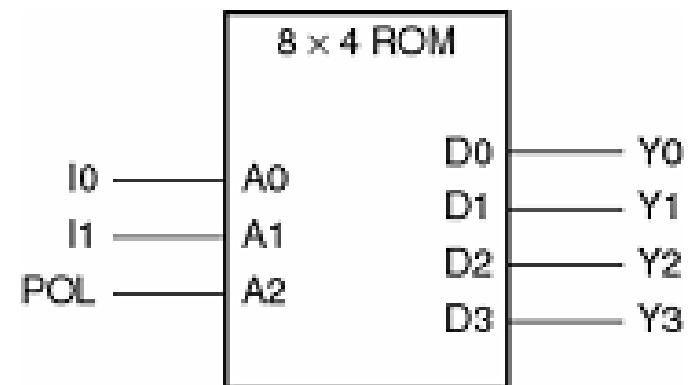
- ROMs are actually combinational devices, not sequential ones!
 - You can store arbitrary data into a ROM, so the same address will always contain the same data.
 - You can think of a ROM as a combinational circuit that takes an address as input, and produces some data as the output.
- A **ROM table** is basically just a truth table.
 - The table shows what data is stored at each ROM address.
 - You can generate that data combinatorially, using the address as the input.

Address $A_2A_1A_0$	Data $V_2V_1V_0$
000	000
001	100
010	110
011	100
100	101
101	000
110	011
111	011

Logic-in-ROM Example



Inputs			Outputs			
A2	A1	A0	D3	D2	D1	D0
0	0	0	1	1	1	0
0	0	1	1	1	0	1
0	1	0	1	0	1	1
0	1	1	0	1	1	1
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0



PROM/Register Sequential Circuit

Need to construct FSM with:

15 states

6 inputs

4 outputs

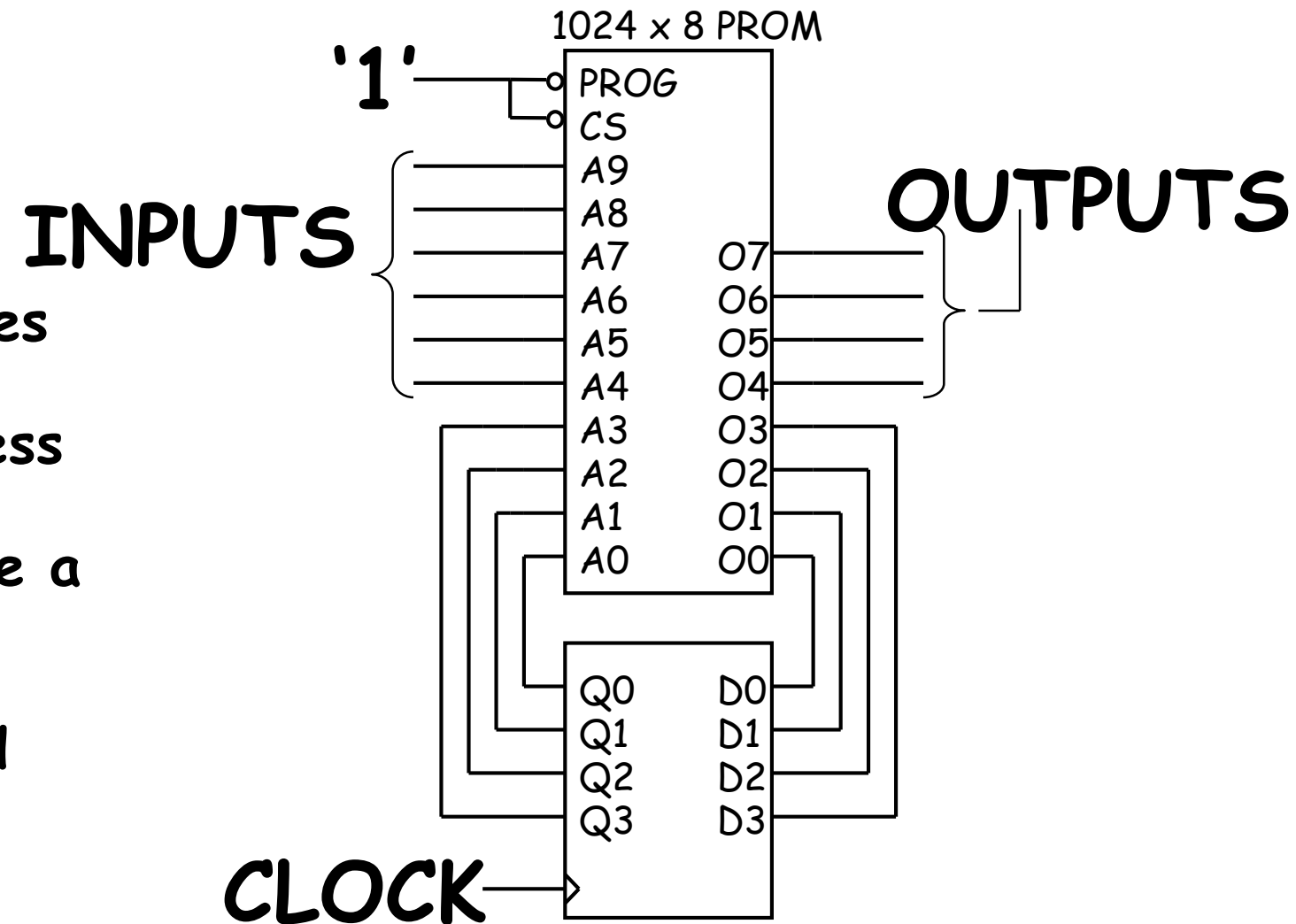
We need at least 4 FFs to store the 15 states

combinational logic inputs (i.e. PROM address signals)

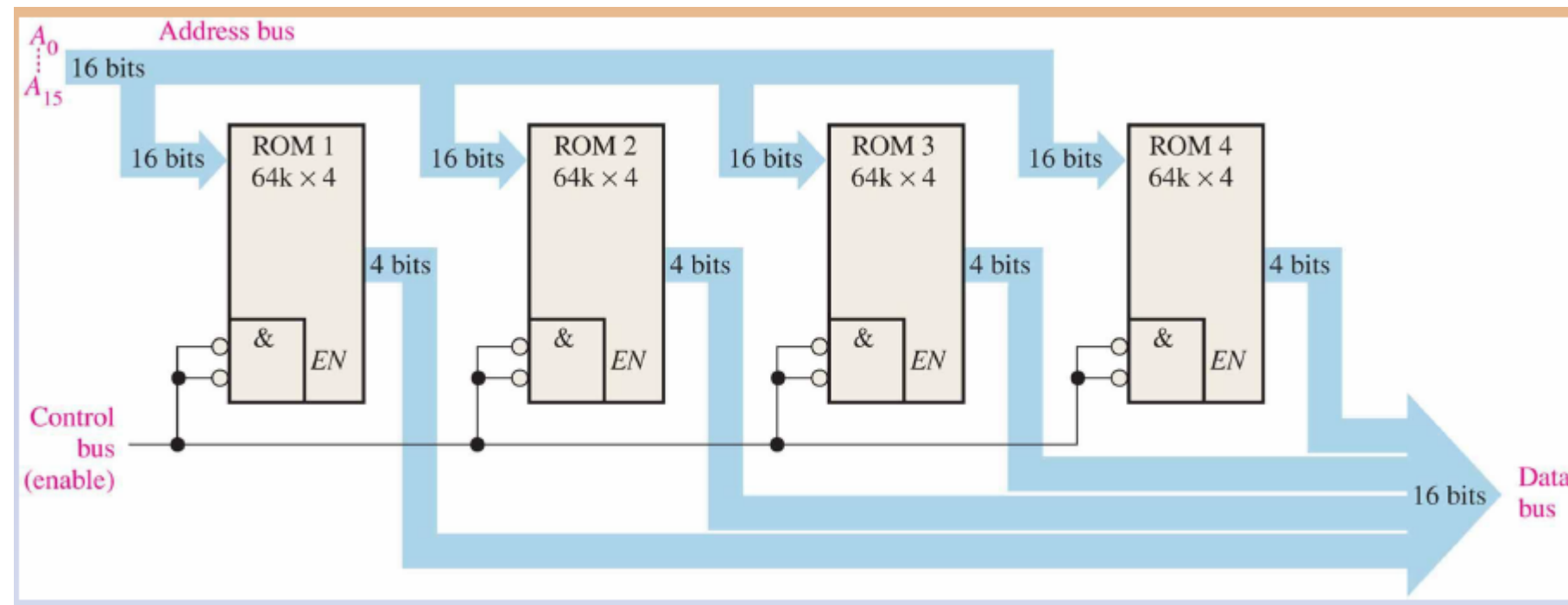
is $6 + 4 = 10$; therefore the PROM must have a minimum of 2^{10} or 1024 locations

Each location has to store the next state and output values; therefore we need at least $4 + 4 = 8$ bits / location

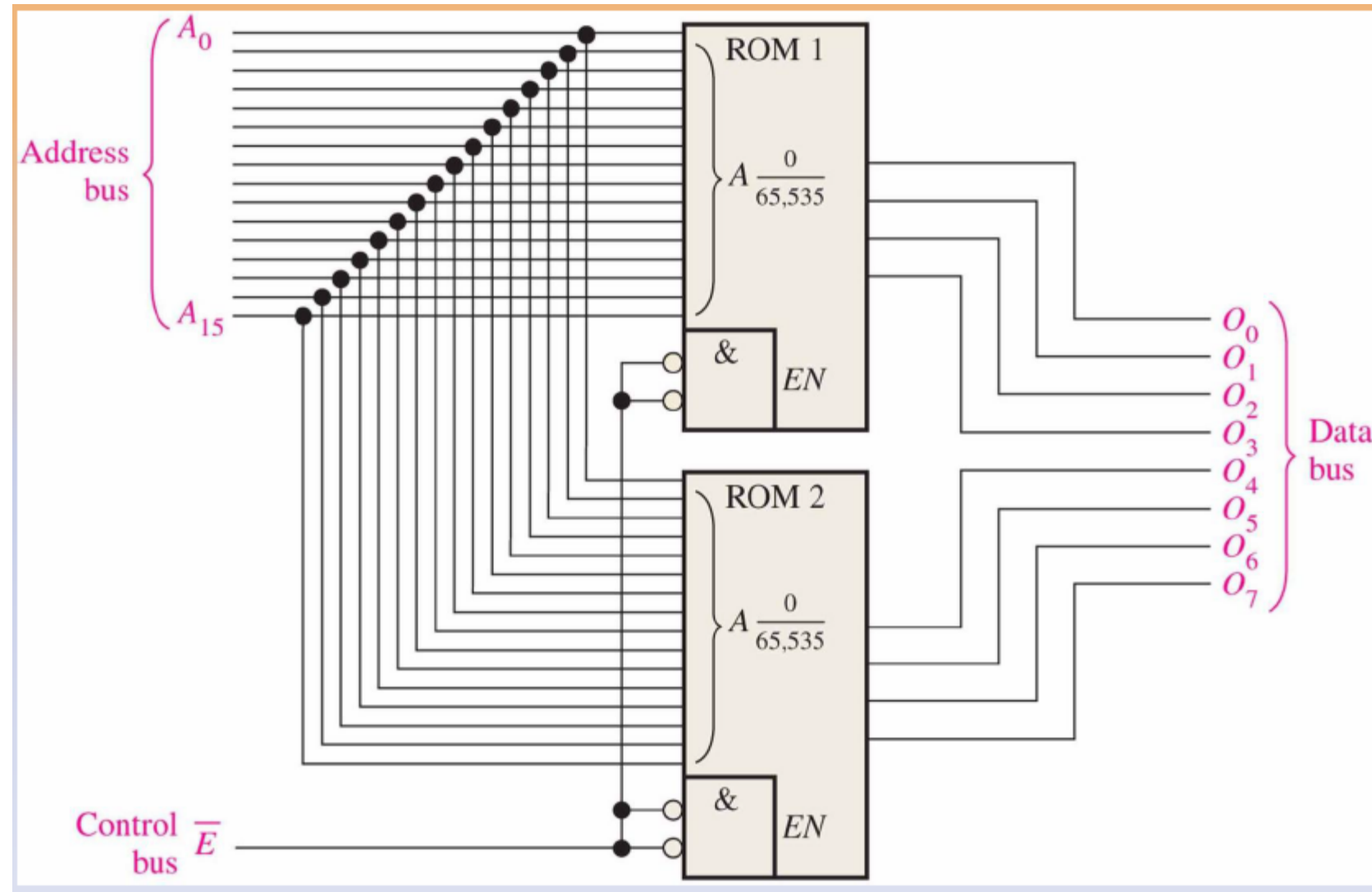
PROM size $\geq 1024 \times 8$



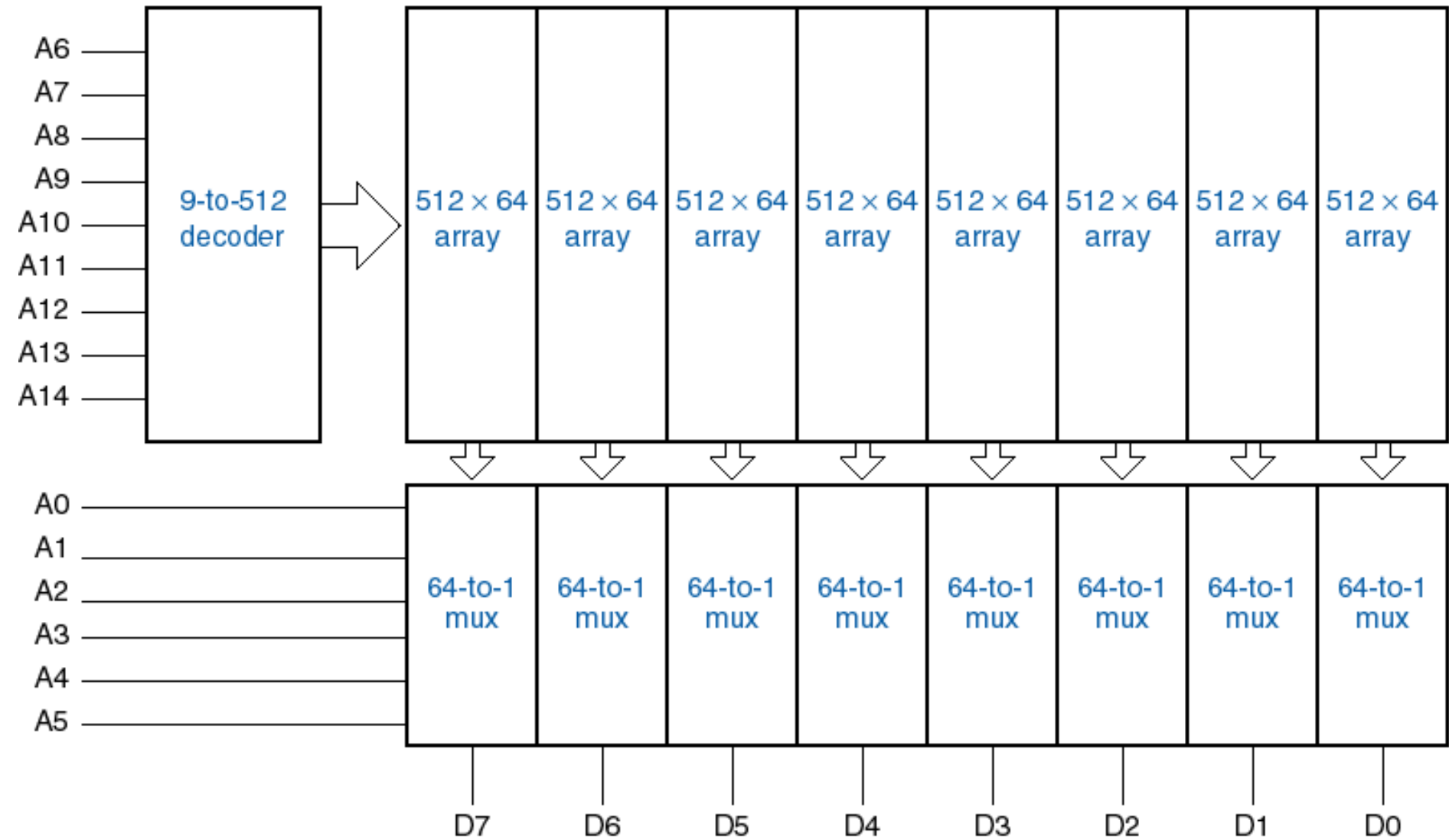
Kelimenin Geniřletilmesi



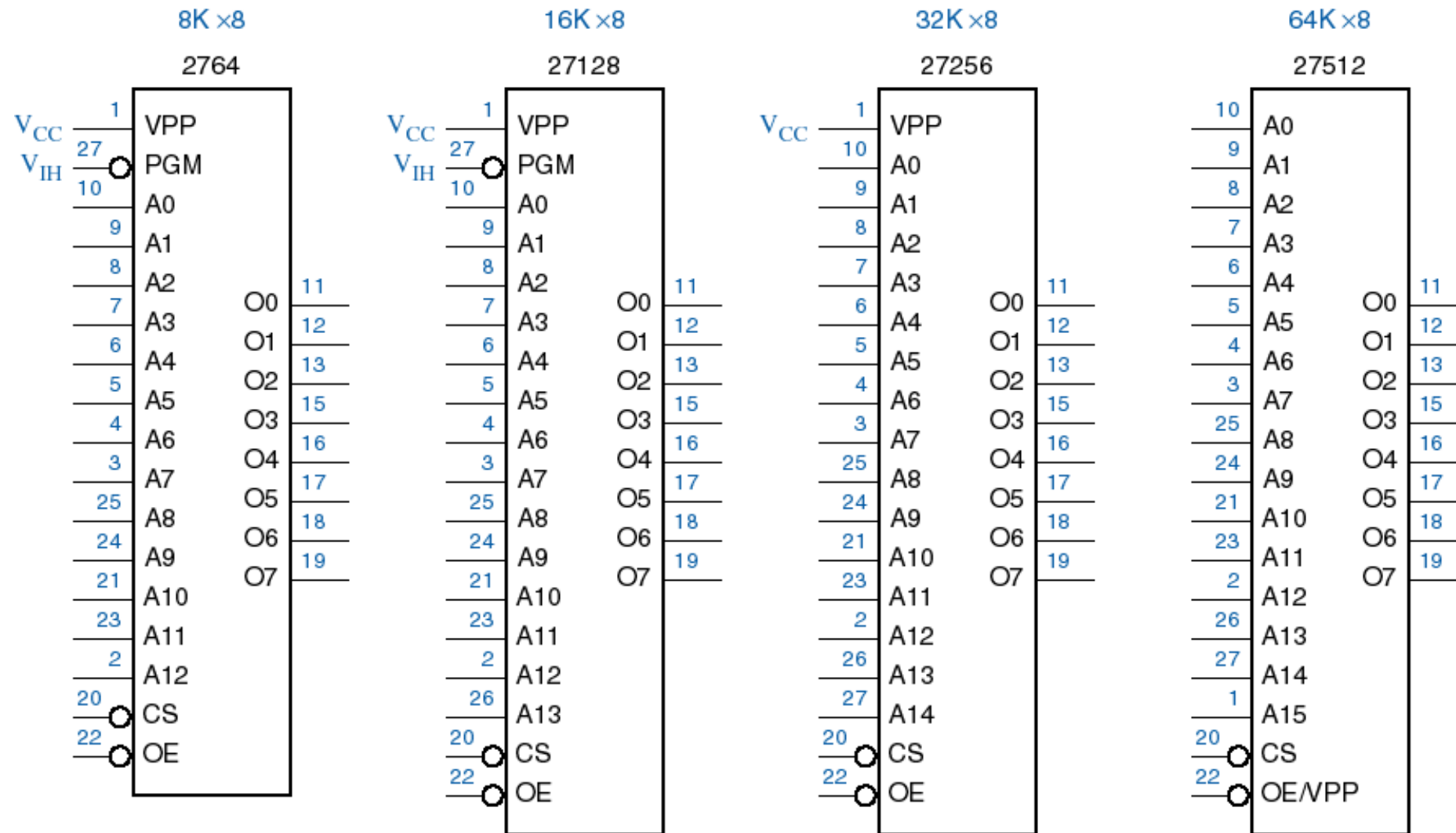
Kelime Genişleme 64Kx8, 64KBayt



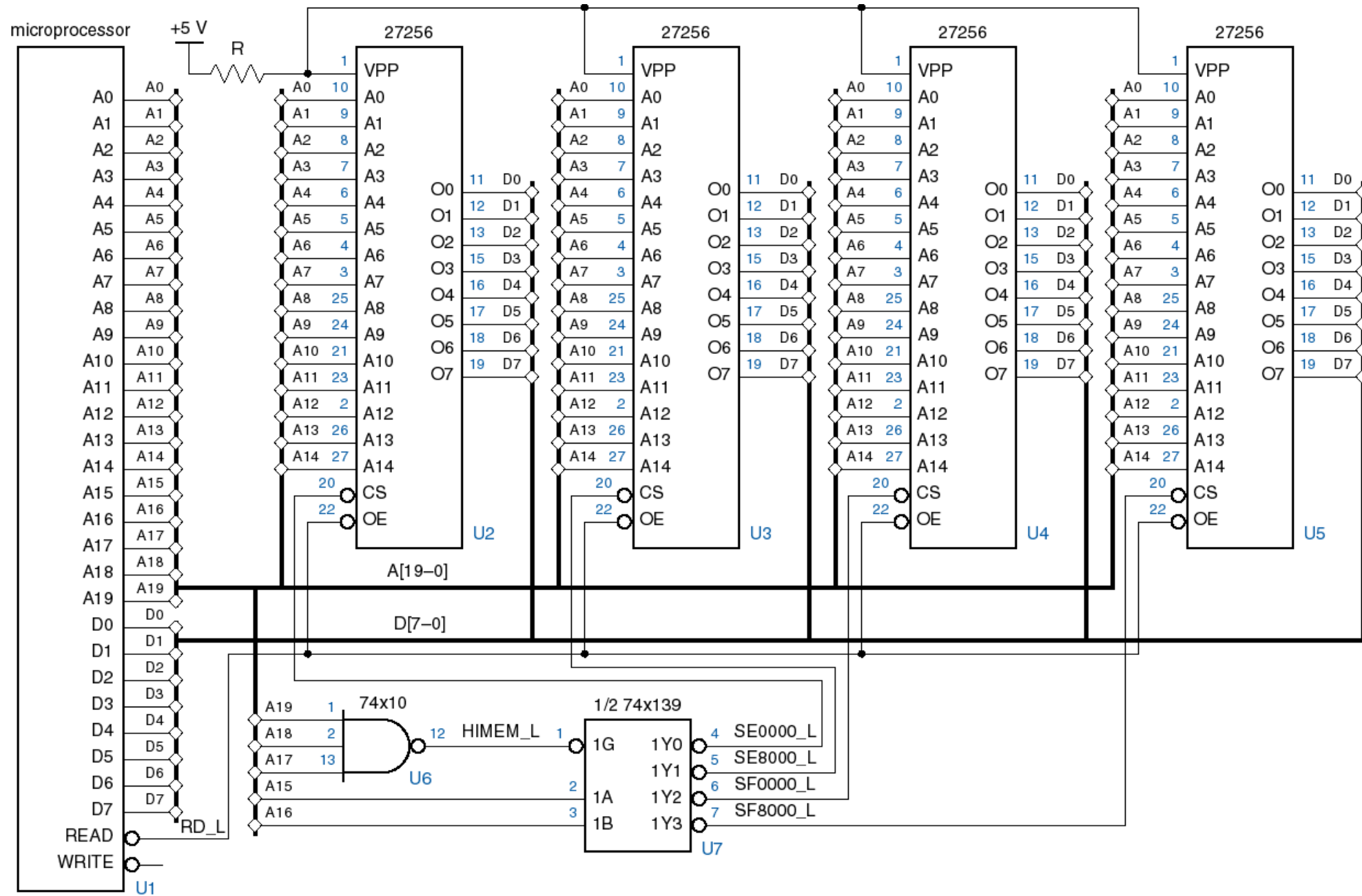
32Kx8 ROM



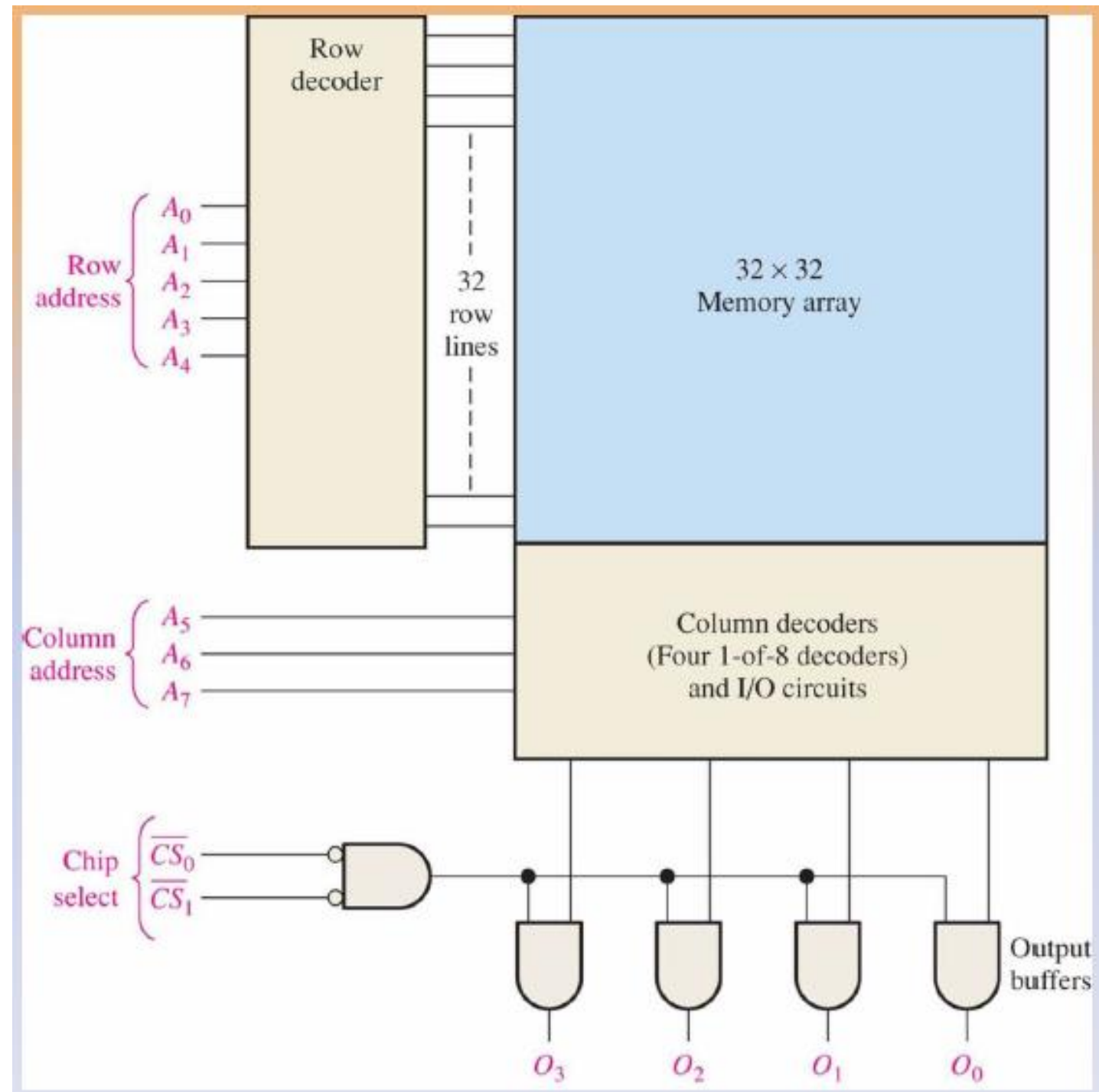
Typical commercial EEPROMs



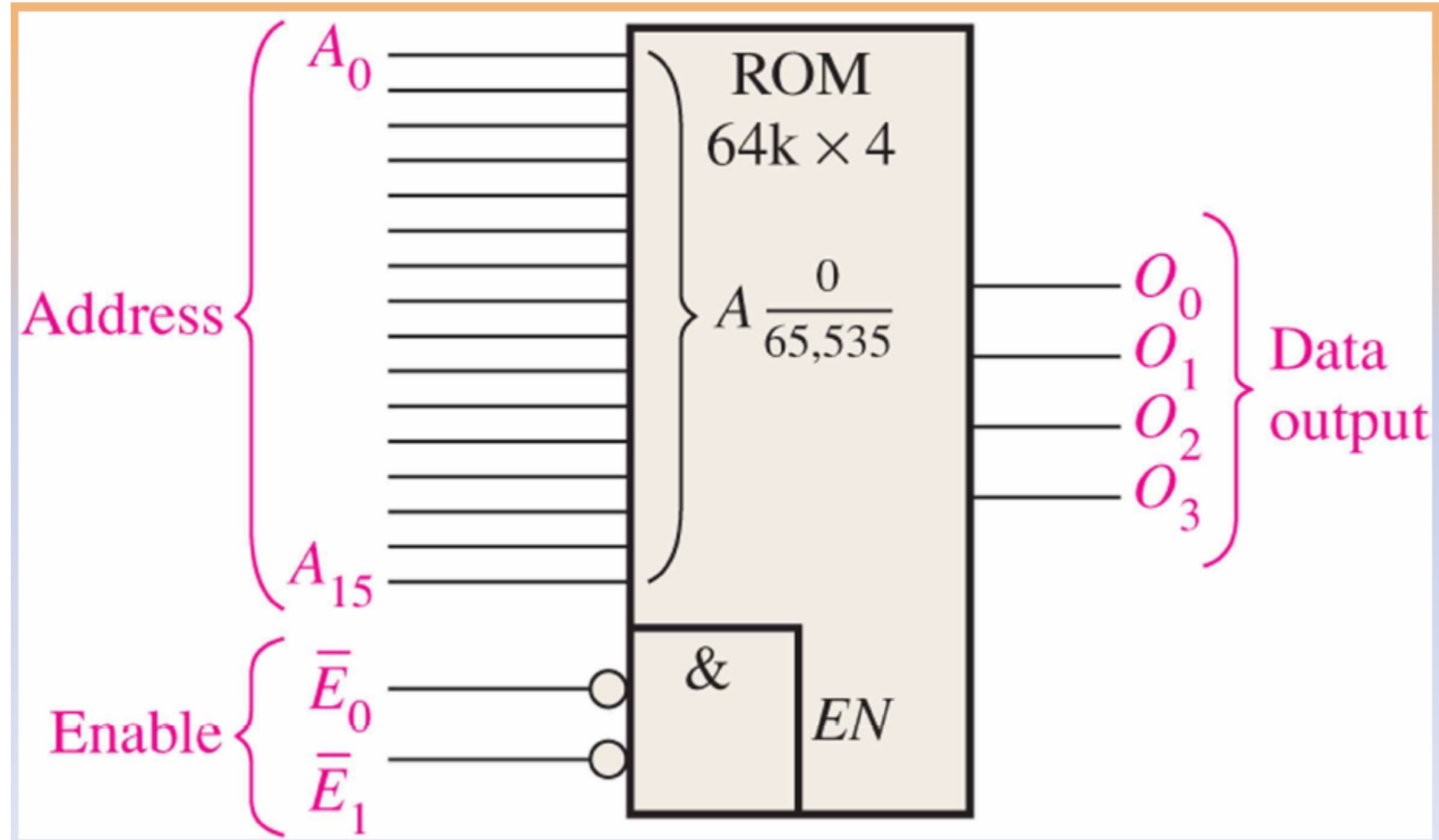
Microprocessor EPROM application



1024-bit ROM, 256 x 4



64k x 4 ROM



Usage Notes

- A lot of slides are adopted from the presentations and documents published on internet by experts who know the subject very well.
- I would like to thank who prepared slides and documents.
- Also, these slides are made publicly available on the web for anyone to use
- If you choose to use them, I ask that you alert me of any mistakes which were made and allow me the option of incorporating such changes (with an acknowledgment) in my set of slides.

Sincerely,

Dr. Cahit Karakuş

cahitkarakus@esenyurt.edu.tr