

# 半导体存储器

# 主要内容：

- 半导体存储器基本概念
- 微机中的存储器系统
- 半导体存储芯片的外部特性及其与系统的连接
- 存储器接口设计（存储器扩展技术）
- 高速缓存

# 半导体存储器概述

# 1. 半导体存储器

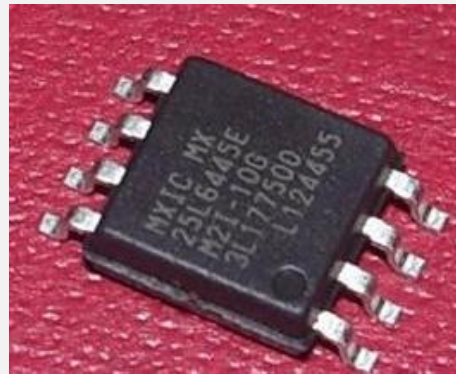
- 半导体存储器由能够表示二进制数“0”和“1”的、具有记忆功能的半导体器件组成。
- 能存放一位二进制数的半导体器件称为一个存储元
- 若干存储元构成一个存储单元。

每个存储单  
元由8个存  
储元构成



## 2. 半导体存储器的分类

内存存储器 { 随机存取存储器 (RAM)  
只读存储器 (ROM)



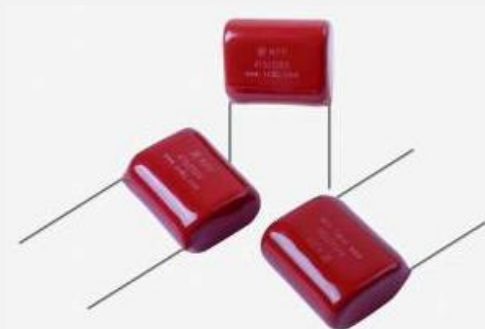
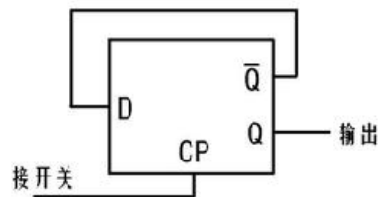
# 随机存取存储器 ( RAM )

RAM

静态存储器 (SRAM)

动态存储器 (DRAM)

存储元为双  
稳态电路



# 只读存储器 ( ROM )



# 3. 半导体存储器的主要技术指标

## ■ 存储容量

- 存储单元个数 $\times$ 每单元的二进制数位数

$$1\text{M} \times 8\text{bit} = 1\text{MB}$$

## ■ 存取时间

- 实现一次读/写所需要的时间

## ■ 存取周期

- 连续启动两次独立的存储器操作所需间隔的最短时间

## ■ 可靠性，功耗



# 微型机中的存储器系统

# 1. 微机中的存储器

- 微型机中的存储器总体上包括：
  - 内存和外存



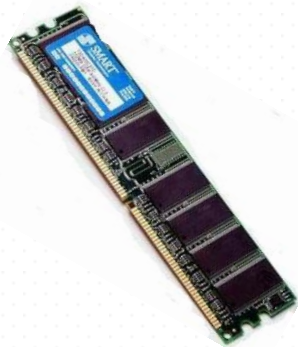
# 微机中的存储器

## ■ 内存储器

- 主内存
- 高速缓冲存储器

## ■ 外存储器

- 联机外存
  - 脱机外存
- ## ■ 虚拟存储器



# 微机中的存储器

- 内存和外存在工作速度、容量、价格、制造材料等各方面都不相同。

	内存	外存
速度	快	慢
容量	小	大
单位容量价格	高	低
制造材料	半导体	磁性材料

## 2. 微机中的存储器系统

### ■ 存储器系统：

- 将两个或两个以上速度、容量和价格各不相同的存储器用硬件、软件或软硬件相结合的方法连接起来，使整个系统的存储速度接近最快的存储器，容量接近最大的存储器，价格接近于最便宜的存储器。

### ■ 微型计算机中的存储系统主要有：

- Cache存储器系统
- 虚拟存储器系统

# Cache存储系统

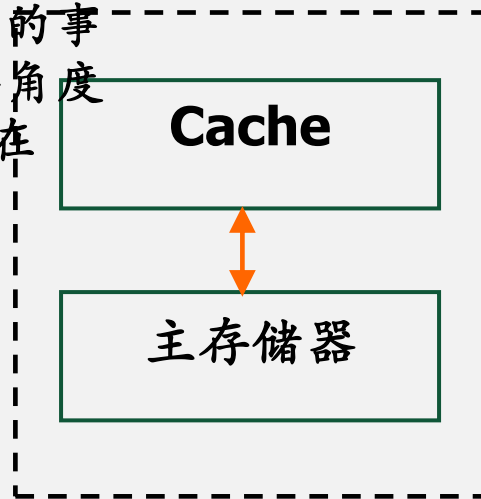
- Cache存储系统由高速缓冲存储器（Cache）和主内存构成，由硬件系统负责管理。

- 对程序员透明

对原本存在的事物，从某个角度看上去不存在

- 主要设计目标：

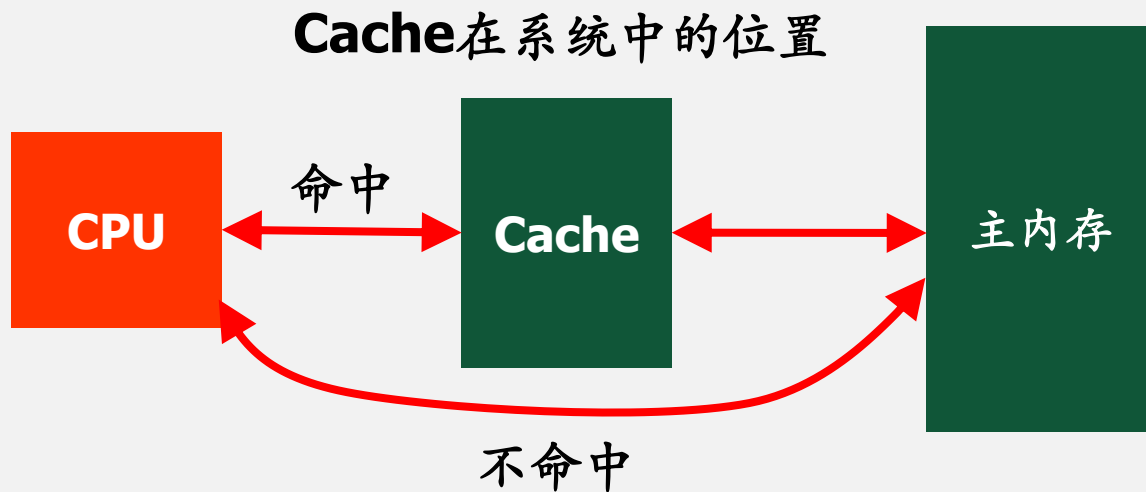
- 提高CPU访问内存的存取速度。



# Cache

- Cache

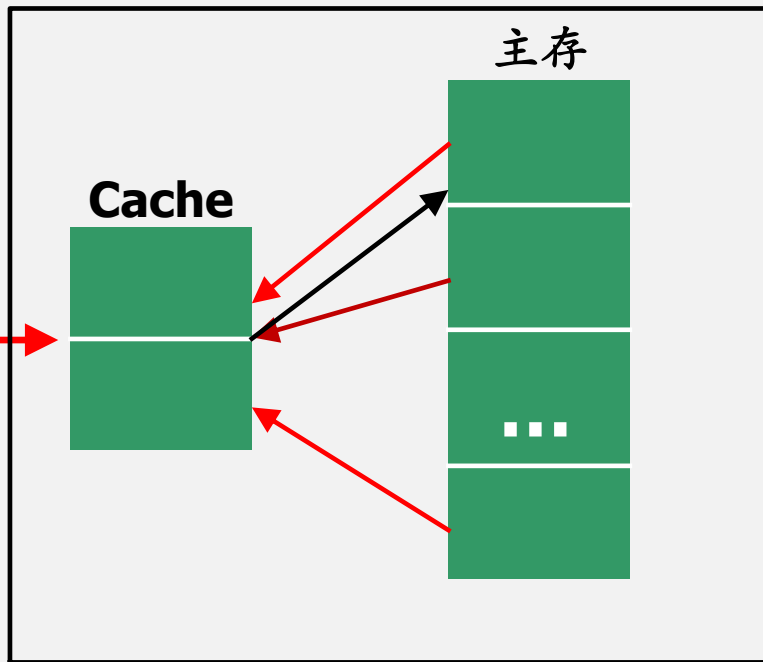
- 高速缓存存储器



当命中率足够高时，整个Cache存储系统的：

- 1) 访问速度接近与Cache的存取速度；
- 2) 存储容量接近与主存的容量；
- 3) 价格接近与主存的价格。

**CPU**



系统存取时间=

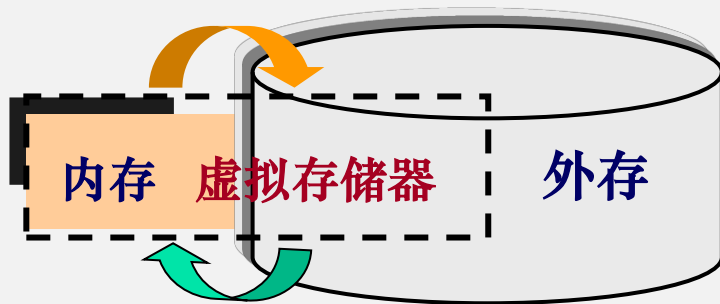
命中率\*Cache存取时间+不命中率\*主存存取时间

$$T = H * T_1 + (1 - H) * T_2$$

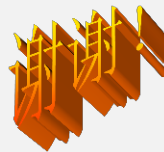
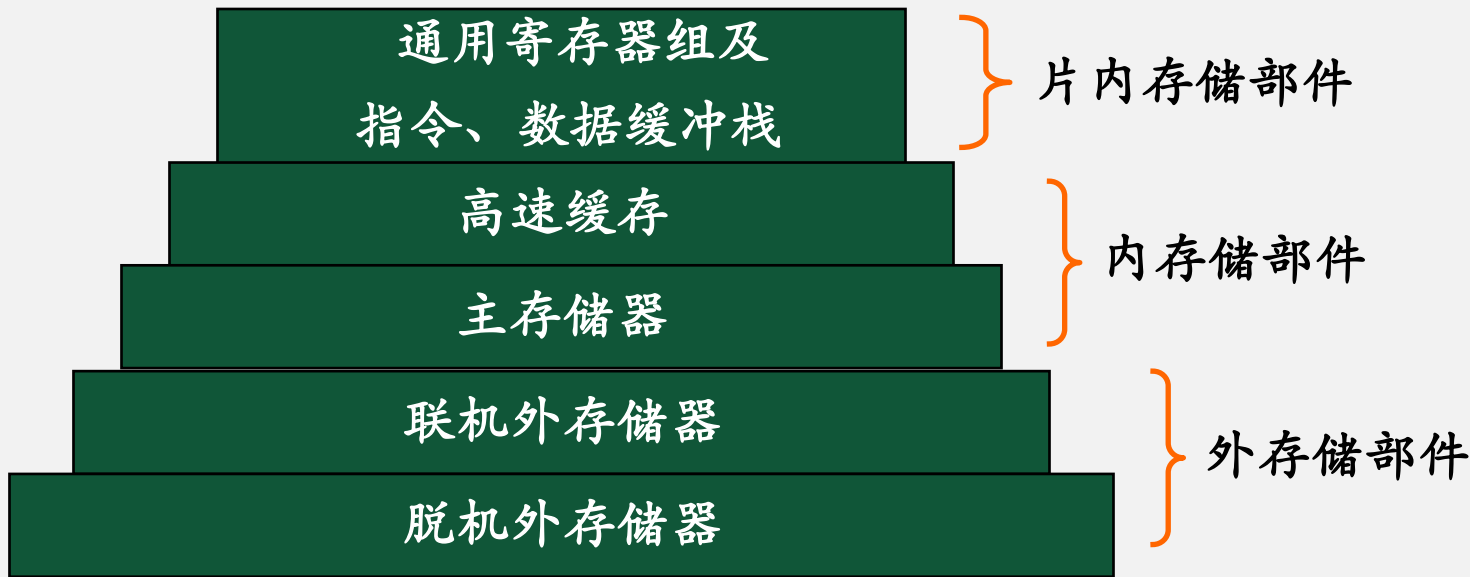


# 虚拟存储器系统

- 虚拟存储器系统由主内存和部分硬磁盘构成，主要由操作系统管理。
  - 对应用程序员是透明的
- 主要设计目标：
  - 扩大存储容量



### 3. 微机中的存储器



# 随机存取存储器

# 随机存取存储器

## ■ 特点：

- 可以随机读或写操作
- 掉电后存储内容即丢失

## ■ 类型：

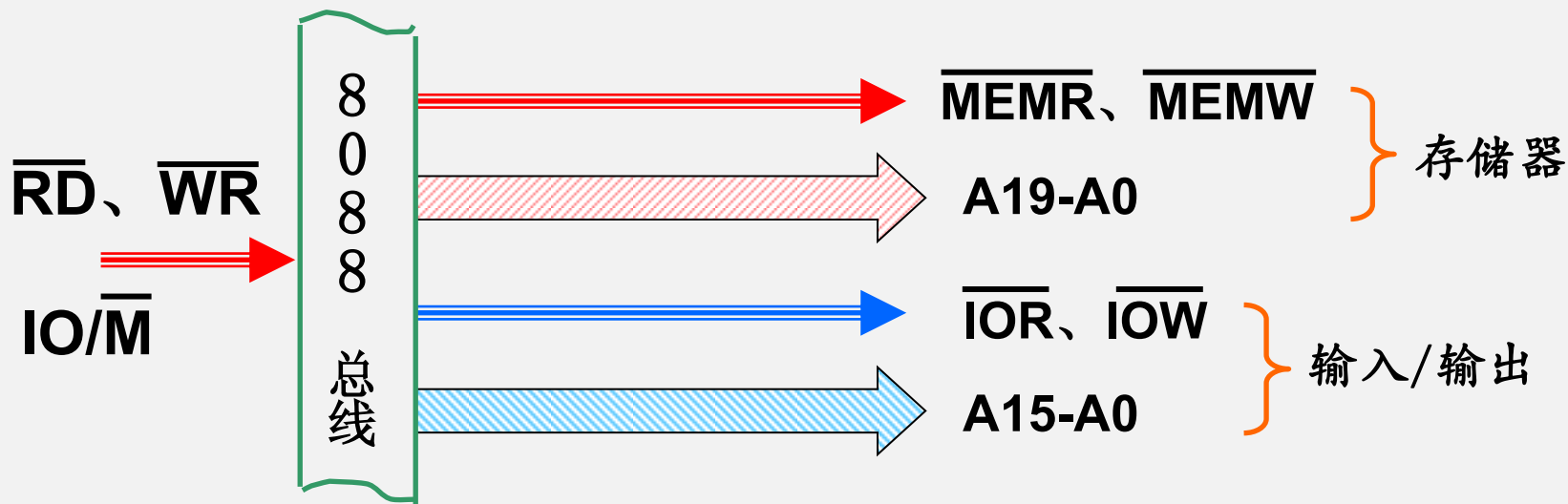
- 静态随机存取存储器 (SRAM)
- 动态随机存取存储器 (DRAM)

## 要求：

利用已有存储器芯片，设计需要的半导体存储器

# 存储单元的编址

# 8088总线信号

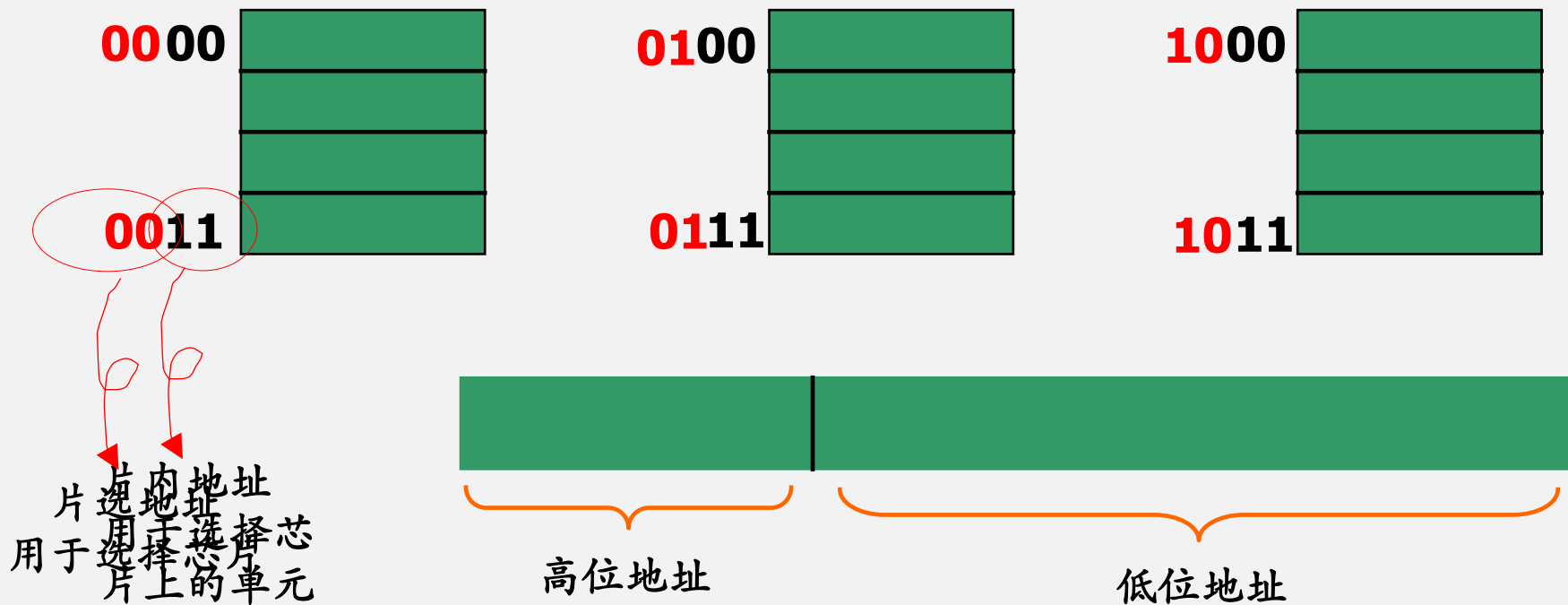


- **IO/#M=0**

- **#RD=0 → #MEMR=0**

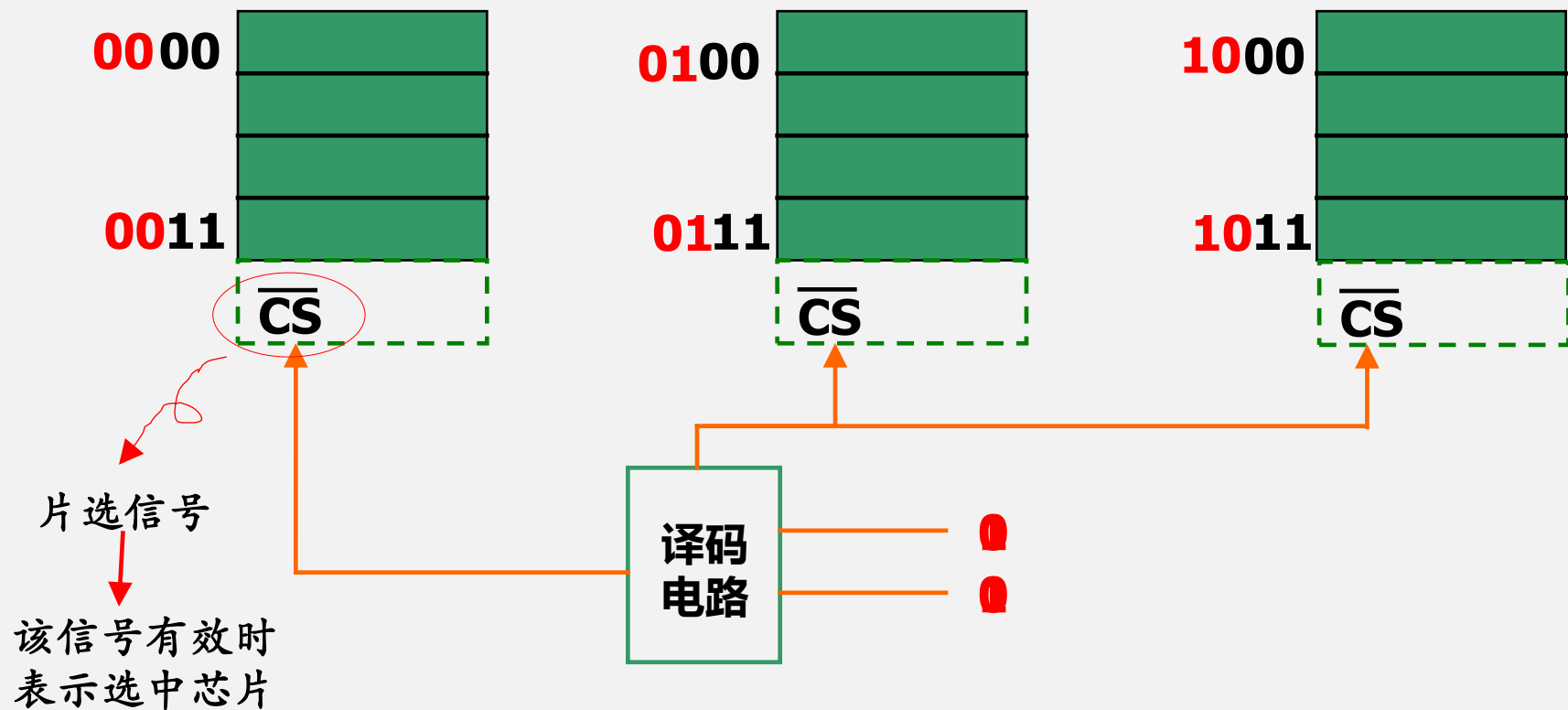
- **#WR=0 → #MEMW=0**

# 存储器编址



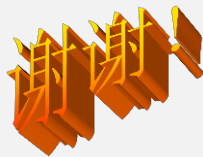


# 存储器编址



# 译码电路

- 将输入的一组高位地址信号通过变换，产生一个有效的输出信号，用于选中某一个存储器芯片，从而确定了该存储器芯片在内存中的地址范围。
- 将输入的一组二进制编码变换为一个特定的输出信号。





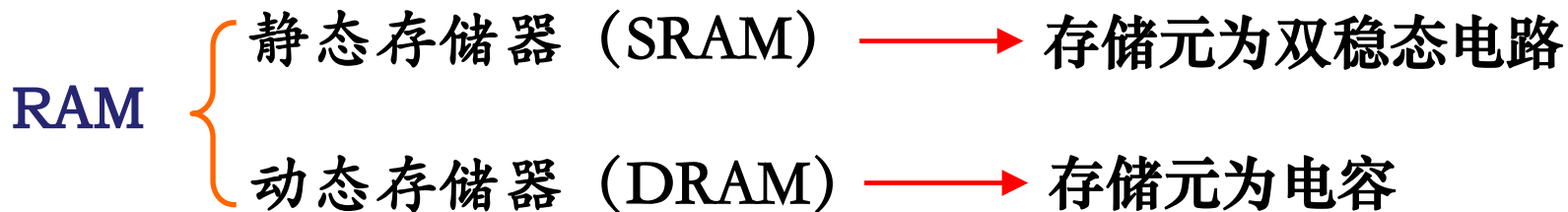
西安交通大学  
XI'AN JIAOTONG UNIVERSITY



# 随机存取存储器RAM



# 随机存取存储器 ( RAM )

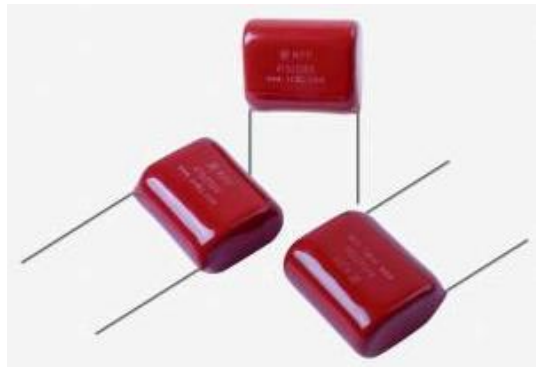


# DRAM的特点

- 存储元主要由电容构成；
- 主要特点：
  - ◆ 需要**定时刷新**



定时为存储元进行读或写  
操作（电容充电、放电）



# 静态随机存取存储器SRAM

# 静态随机存取存储器

存储元由双稳电路构成，存储信息稳定。

**示例芯片: SRAM 6262**

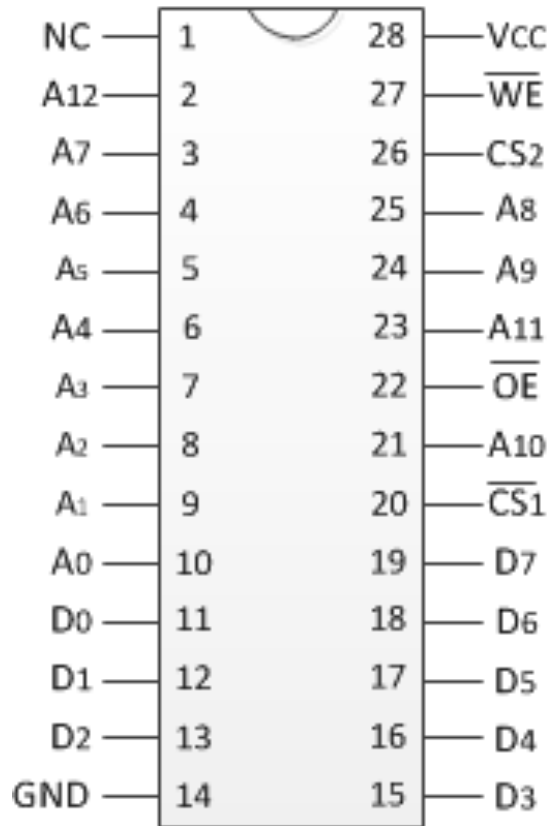
# 1. SRAM 6264

## ■ 容量:

- ◆  $8K \times 8b$

## ■ 主要引线:

- ◆ 地址线:  $A_0$ ----- $A_{12}$ ;
- ◆ 数据线:  $D_0$ ----- $D_7$ ;
- ◆ 输出允许信号:  $\#OE$ ;
- ◆ 写允许信号:  $\#WE$ ;
- ◆ 选片信号:  $\#CS_1$ ,  $CS_2$ 。

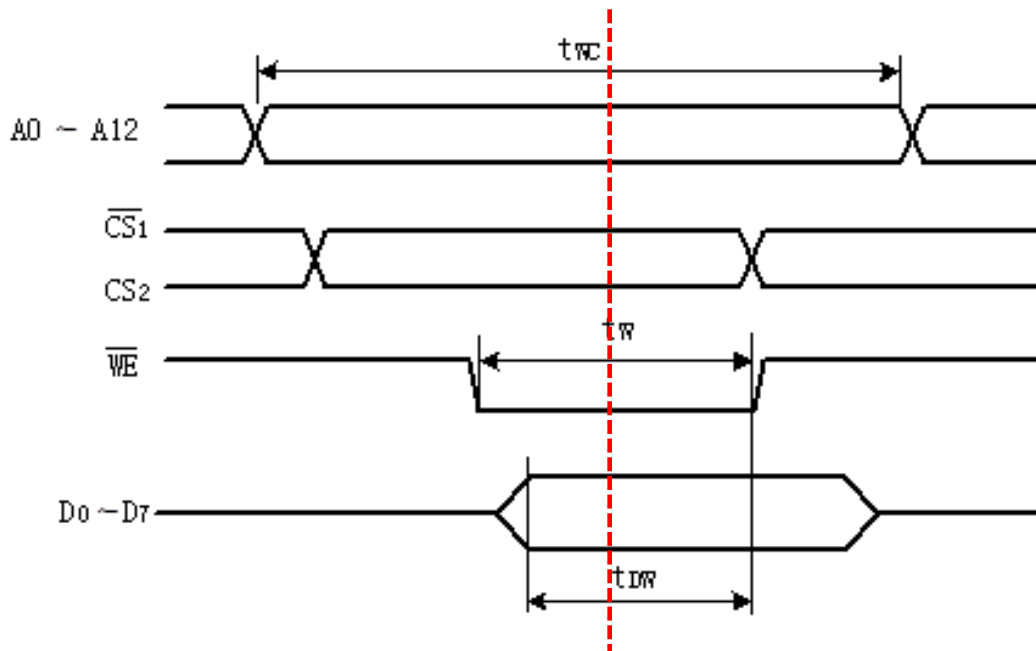




# 6264的工作过程

- 读操作
- 写操作

写操作时序



## 2. 6264芯片与系统的连接

### ■ 存储器芯片与系统的连接分为两部分：

#### ◆ 确定要访问的存储芯片

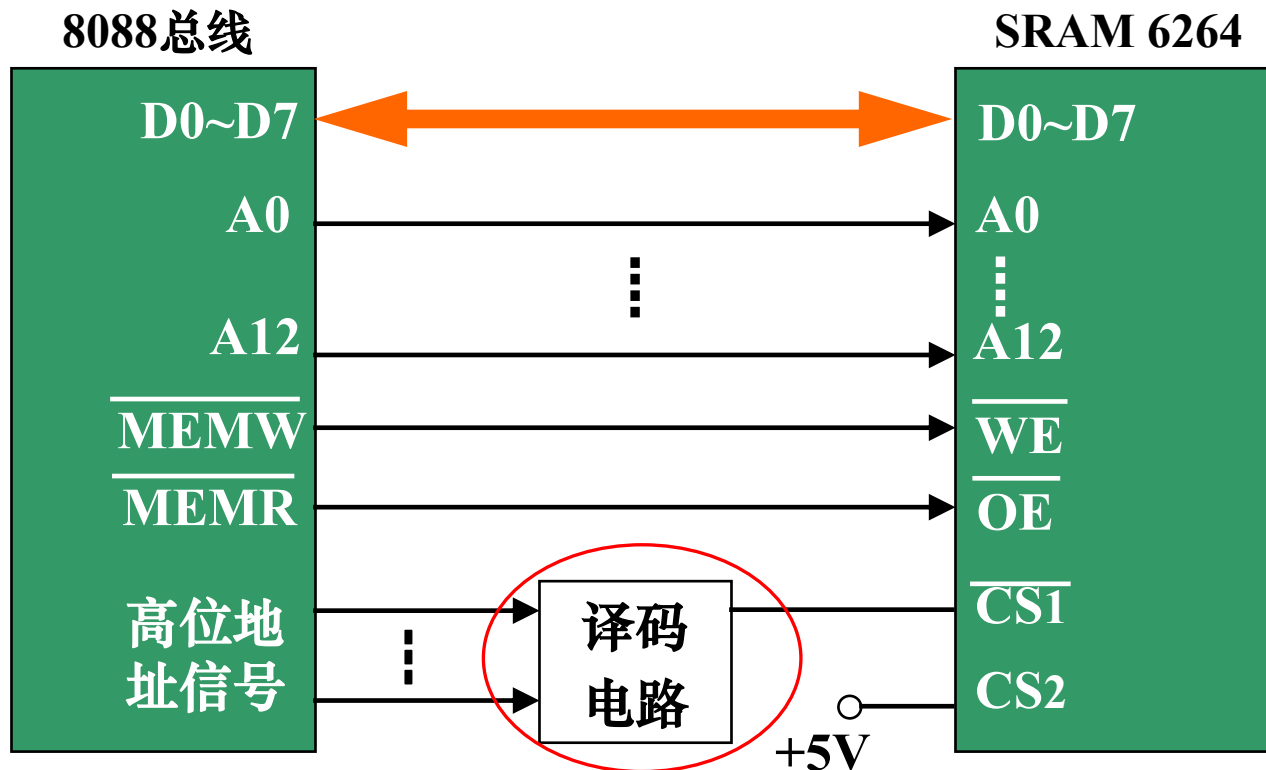
- 系统中可能存在多片存储器芯片，要访问的单元只能存在于某一片芯片上。

#### ◆ 找到芯片后，寻找该芯片上要访问的单元。

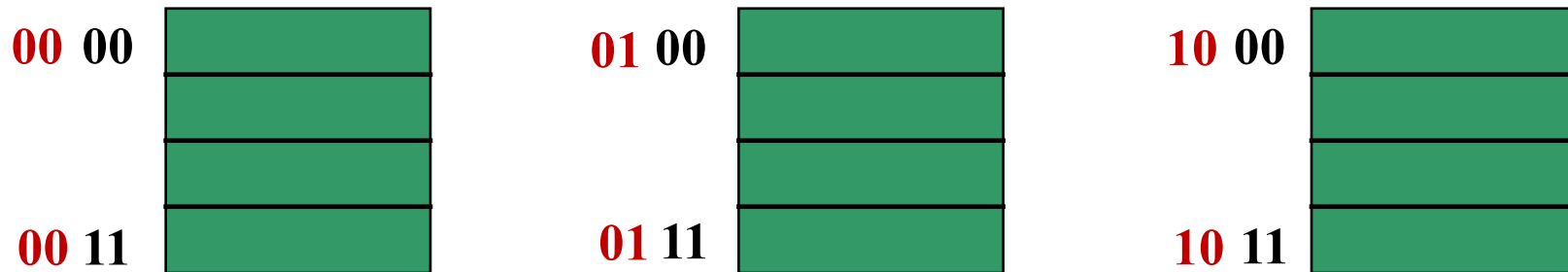
- 6264芯片上有8K个单元，每个单元在该芯片上有惟一的13位地址码
- 每片6264芯片上第一个单元在该芯片上的地址：0
- 每片6264芯片上最后一个单元在该芯片上的地址：8191

用芯片的13位地址码A0-A12寻址片内的每个单元

# 6264与系统的连接框架图



# 存储器编址



# 6264芯片的编址



芯片上所有的单元具有相同的高位地址（片选地址）

从片首地址到片尾地址，构成芯片在内存空间中占有的地址范围

# 3. 译码电路设计

## ■ 译码：

- ◆ 将输入的一组高位地址信号通过变换，产生一个有效的输出信号，用于选中某一个存储器芯片，从而确定了该存储器芯片在内存中的地址范围。

**将输入的一组二进制编码变换为一个特定的输出信号**

# 译码方式

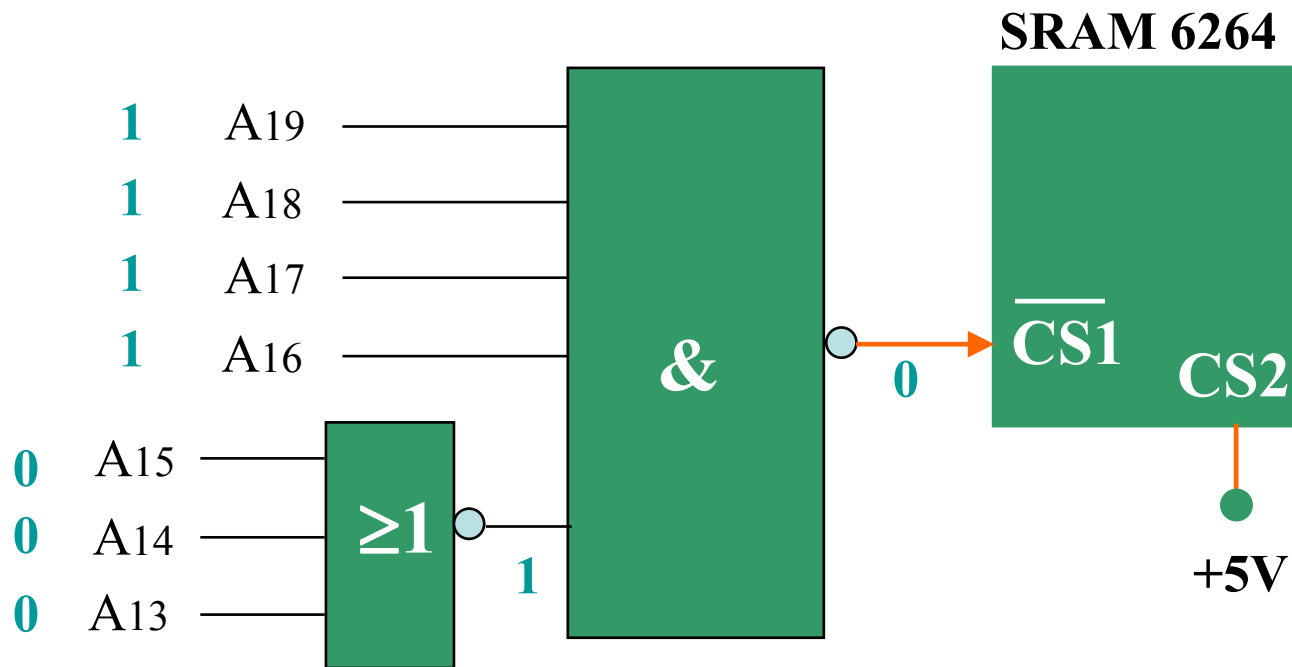
- 全地址译码
- 部分地址译码

# 全地址译码

用全部的高位地址信号作为译码信号，使得存储器芯片的每一个单元都占据一个唯一的内存地址。



# 全地址译码例



## 6264芯片全地址译码例



**该6264芯片的地址范围 = F0000H~F1FFFH**

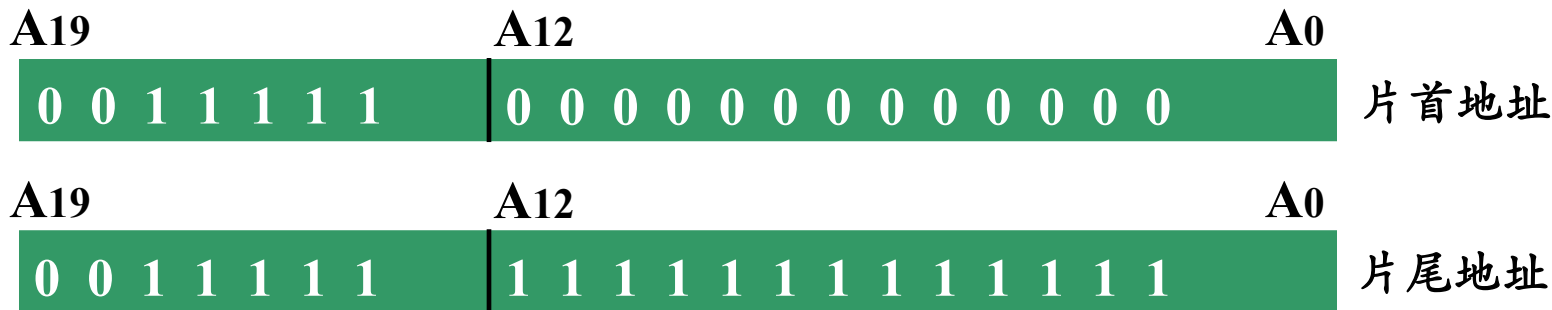
## 全地址译码例

■ 例：

- ◆ 已知某SRAM 6264芯片在内存中的地址为：3E000H~3FFFFH。试画出将该芯片连接到系统的译码电路。

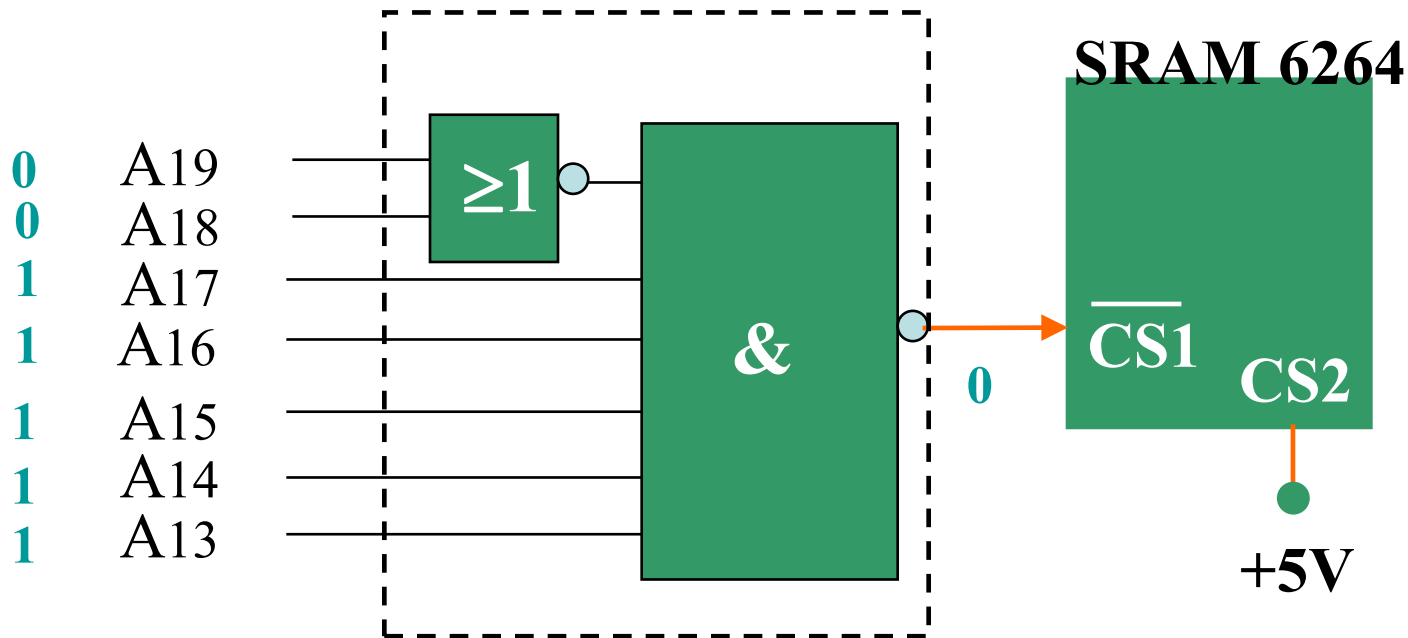
### ■ 设计步骤:

- ◆ 写出地址范围的二进制表示;
- ◆ 确定各高位地址状态;
- ◆ 设计译码器。



# 全地址译码例

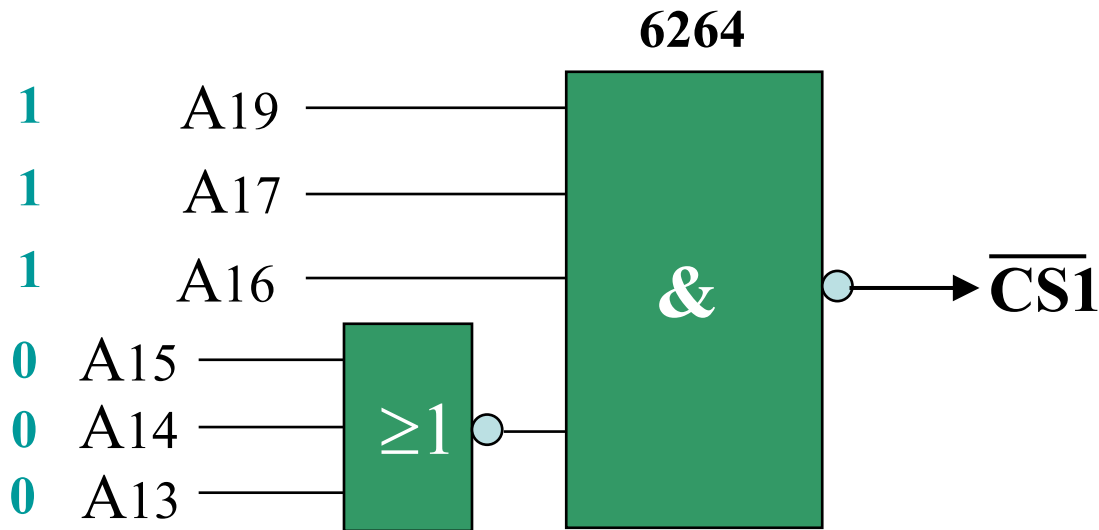
高位地址：0011111



# 部分地址译码

用部分高位地址信号（而不是全部）作为译码信号，使得被选中存储器芯片占有几组不同的地址范围。

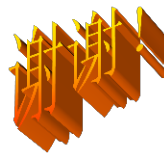
# 部分地址译码例



高位地址:  $1 \times 11000$  ———  $1011000$ ,  $1111000$

两组地址: **F0000H ——— F1FFFH**

**B0000H ——— B1FFFH**



# 只读存储器 ROM

# 只读存储器 ( Read only Memory )

1 EPROM

2 EEPROM

3 Flash



➤ **EPROM**

# 1. 特点

- 可多次编程写入；
- 掉电后内容不丢失；
- 内容的擦除需用紫外线擦除器。

EPROM芯片因其较高的稳定性，使用时常用作程序存储器，存放相应的控制程序。

RAM芯片则因其便利性，常用作数据存储器，存放操作的数据。

## 2. EPROM 2764

- 8K×8bit 芯片
- 地址信号：A0 —— A12
- 数据信号：D0 —— D7
- 输出信号： $\overline{OE}$
- 片选信号： $\overline{CE}$
- 编程脉冲输入： $\overline{PGM}$
- 其引脚与SRAM 6264完全兼容。

# 2764的工作方式

## ■ 数据读出

- 可在线随机读取

## ■ 编程写入

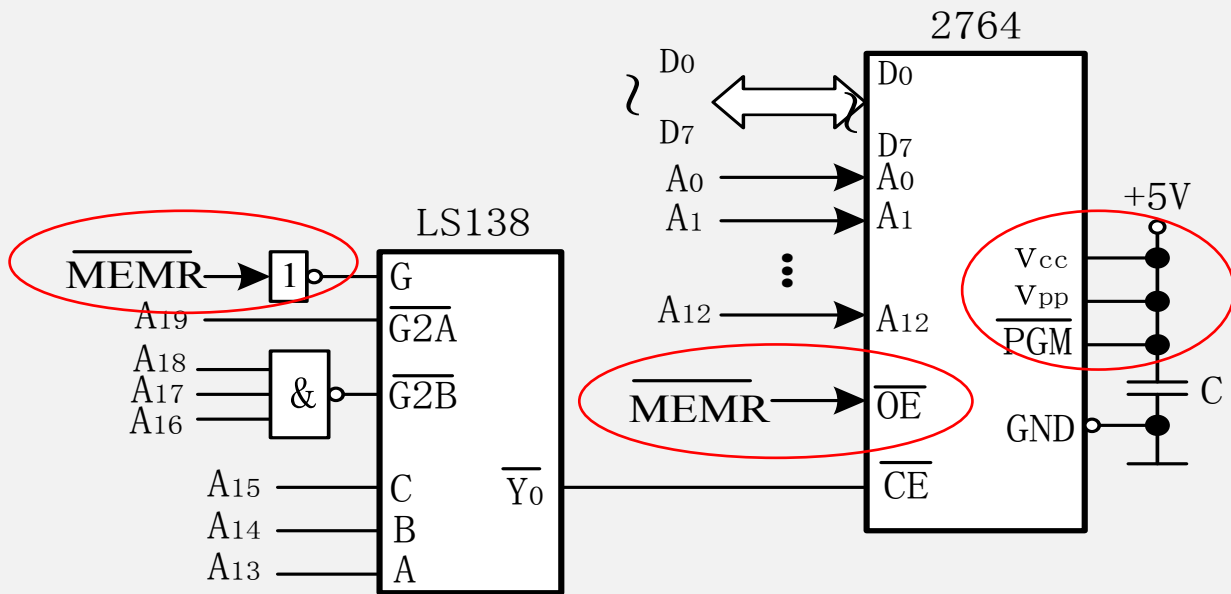
- 不可在线写操作；
- 需专用编程写操作环境。在编程写脉冲控制下完成写操作（每一个写脉冲写入1字节数据）

## ■ 擦除

- 紫外光擦除

# EPROM 2764的应用

## 2764与系统的连接



## 二、EEPROM

# 1. 特点

- 可在线编程写入；
- 掉电后内容不丢失；
- 电可擦除。

## 2. 工作方式

- 数据读出

- 编程写入 { 字节写入：每次写入一个字节  
自动页写入：每次写入一页（1~ 32字节）

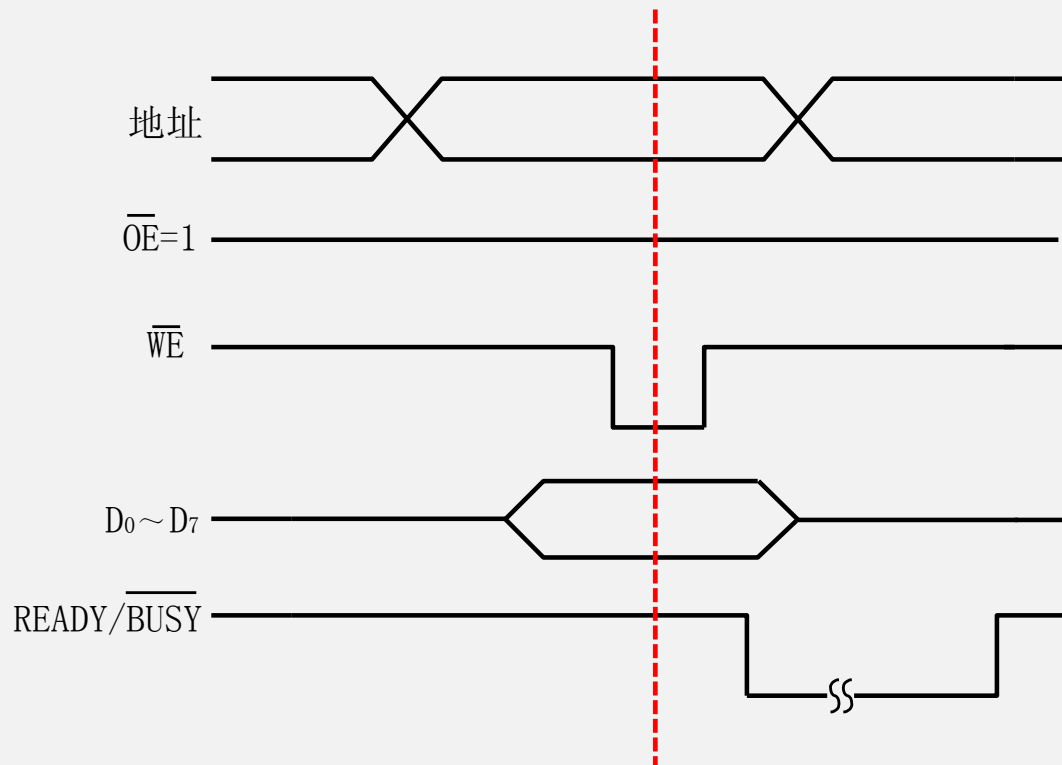
- 擦除 { 字节擦除：一次擦除一个字节  
片擦除：一次擦除整片



### 3. 典型EEPROM芯片98C64A

- 8K×8bit 芯片；
- 13根地址线 (A0 —— A12) ；
- 8位数据线 (D0 —— D7) ；
- 输出允许信号 ( $\overline{\text{OE}}$ ) ；
- 写允许信号 ( $\overline{\text{WE}}$ ) ；
- 选片信号 ( $\overline{\text{CE}}$ ) ；
- 状态输出端 (READY /  $\overline{\text{BUSY}}$ ) 。

# 98C64A工作时序



## 4. EEPROM的应用

- 可通过程序实现对芯片的读写；
- 仅当 $\text{READY} / \overline{\text{BUSY}}=1$ 时才能进行“写”操作
- “写”操作的方法：
  - ① 根据参数定时写入
  - ② 通过判断 $\text{READY} / \# \text{BUSY}$ 端的状态进行写入
    - 仅当该端为高电平时才可写入下一个字节。
  - ③ 中断控制方式
    - 当 $\text{READY} / \# \text{BUSY}$ 端为高电平时，该高电平可作为中断请求信号

# 例：

- 将一片98C64A接到系统总线上，使其地址范围在3E000H~3FFFFH之间。并编程序将芯片的所有存储单元写入66H。
- 题目分析：
  - 硬件设计：实现存储器芯片与系统的连接
  - 软件设计：通过定时方式循环向各单元写入66H



根据芯片参数：

每100us写入1字节数

# 硬件设计

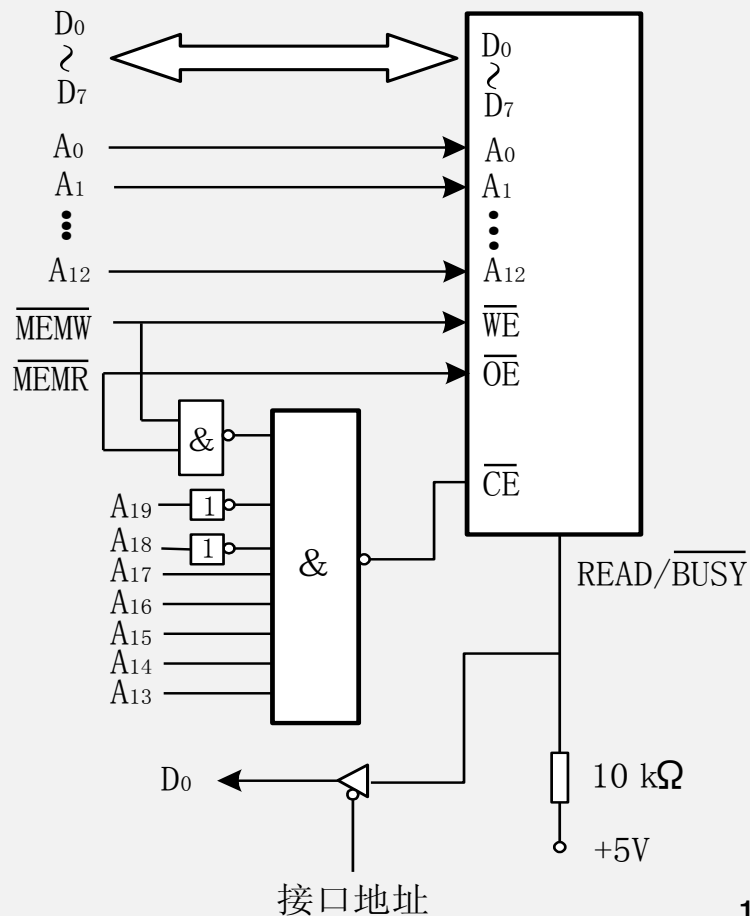
地址范围:

**0011 1110 0000 0000 0000**

|

**0011 1111 1111 1111 1111**

系统总线信号



# 软件设计

通过定时方式进行写操作：

```
START: MOV AX, 3E00H
        MOV DS, AX
        MOV SI, 0000H
        MOV CX, 8192
AGAIN:  MOV AL, 66H
        MOV [SI], AL
        CALL TDELAY120μs
        INC SI
        LOOP AGAIN
        HLT
```

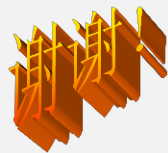
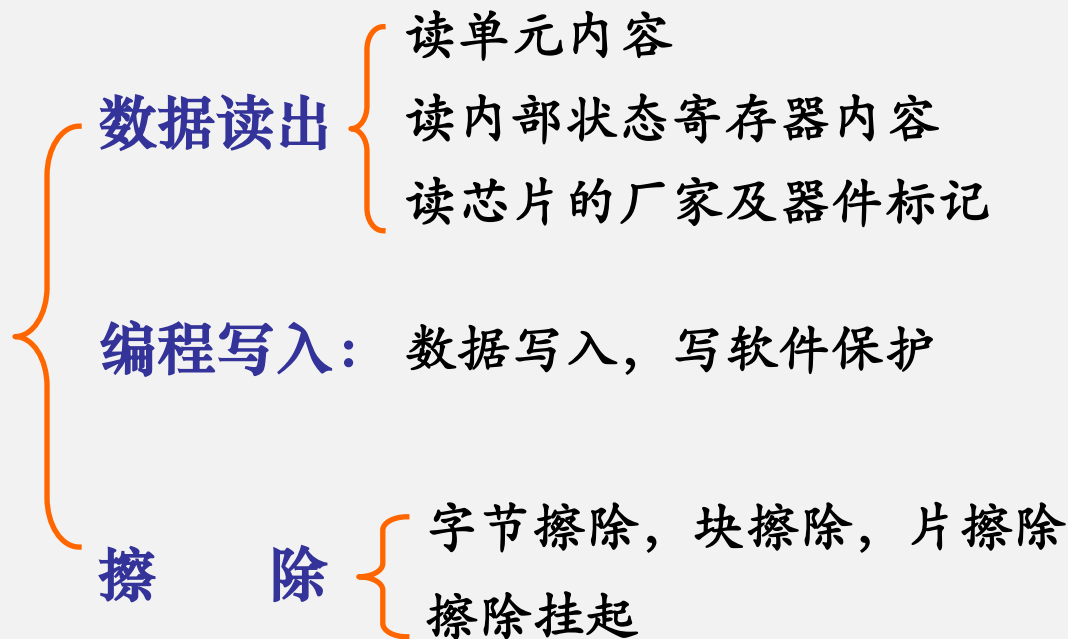
## 三、 闪存存储器 Flash

# 1. Flash的特点

- 通过向内部控制寄存器写入命令的方法来控制芯片的工作方式。
- 通过读状态寄存器的值，获取芯片当前工作状态
- 与SRAM的区别：
  - 在进行写入和擦除操作时需要12V编程电压
- 与普通EEPROM的区别：
  - 通过读状态寄存器的内容确定是否可继续写入
  - 提高写命令字的方式控制其处于何种工作方式。



## 2. 工作方式



# 存储器扩展技术

用已有的存储器芯片构造一个需要的存储空间

# 存储器扩展

存储器芯片的存储容量等于：

单元数 × 每单元的位数

- 用多片存储芯片构成一个需要的内存空间；  
字节数 存储器芯片 字长 在整个内存中占据不同的地址范围；
- 任一时刻仅有一片（或一组）被选中。

# 存储器扩展方法

位扩展       扩展字长

字扩展       扩展单元数

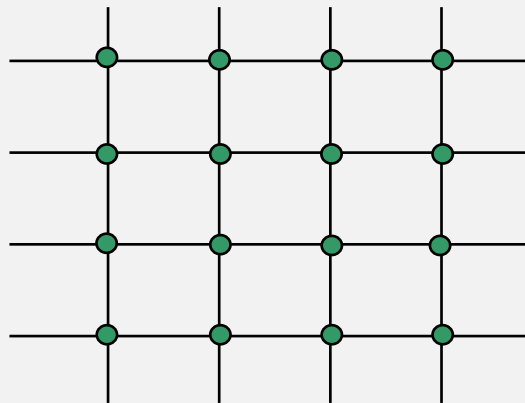
字位扩展       既扩展字长也扩展单元数

# 1. 位扩展

- 构成内存的存储器芯片的字长小于内存单元要求的字长——需进行位扩展。
- **位扩展**：每单元字长的扩展。

# 位扩展例

- 实例芯片：DRAM 2164A：
  - $64K \times 1\text{bit}$
- 采用行地址和列地址来确定一个单元；
- 行列地址分时传送，共用一组地址信号线；
  - 地址信号线的数量仅为同等容量SRAM芯片的一半。



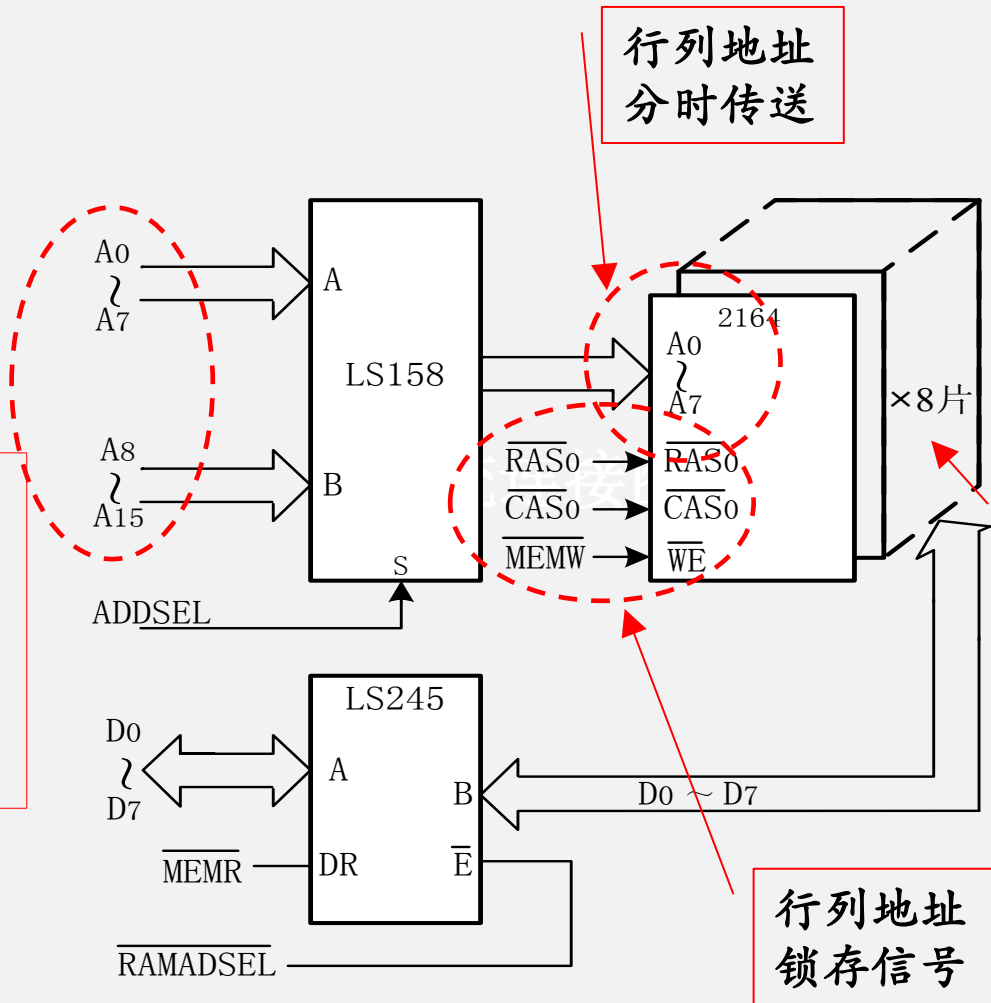
2164A芯片每单元仅1位二进制码，故1片芯片不能构成独立存储器。

# 位扩展例

用8片2164A构成64KB  
存储体

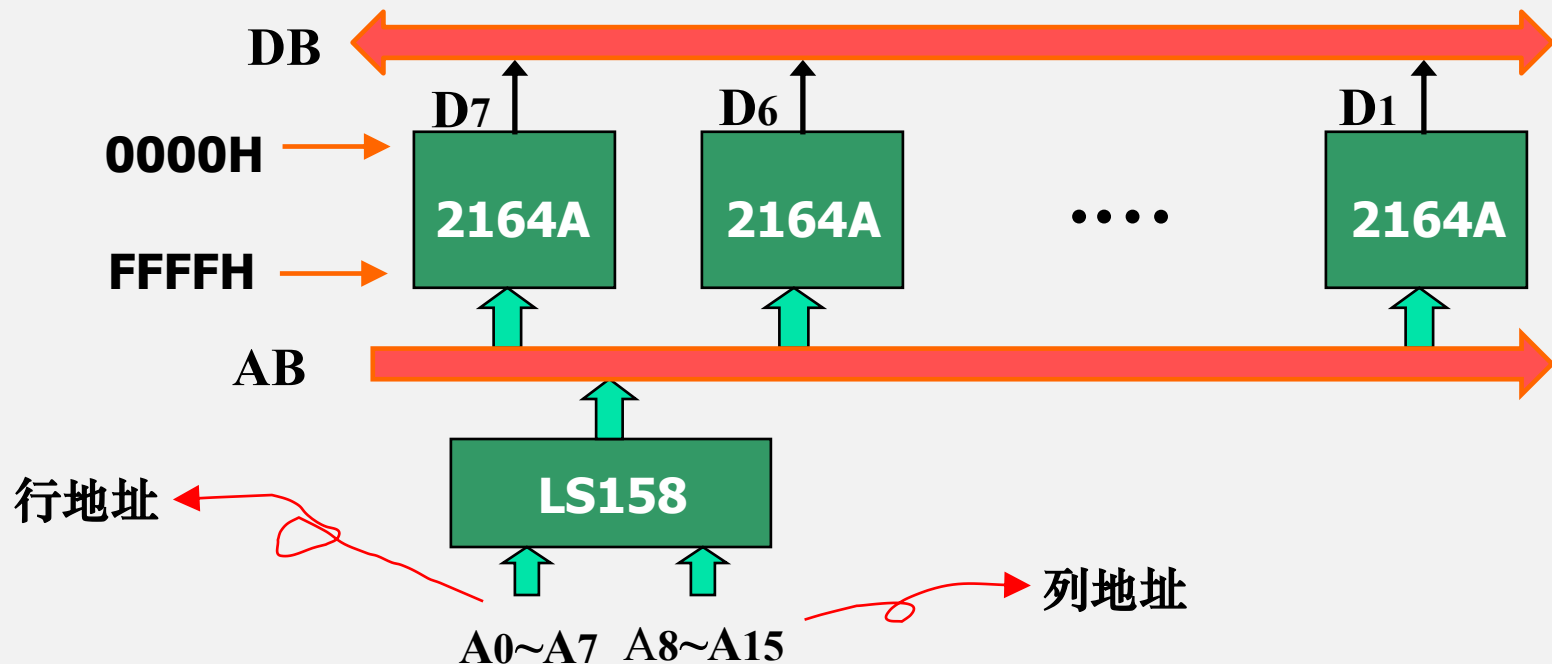
8片2164A必须具有  
完全相同的地址，必  
须同时被选中或同时  
不被选中

通过高、低  
位地址分时传送  
行和列地址



# 位扩展例

用8片2164A芯片构成64KB存储器。





# 位扩展方法：

- 位扩展的连线特点：

- 将每片的地址线、控制线并联，数据线分别引出。

- 效果：

- 存储器的单元数不变，位数增加。

**位扩展：**确保所有芯片具有**完全相同**的地址范围

**对需要位扩展的存储芯片，单独1片没有意义**

## 2. 字扩展

### ■ 地址空间的扩展

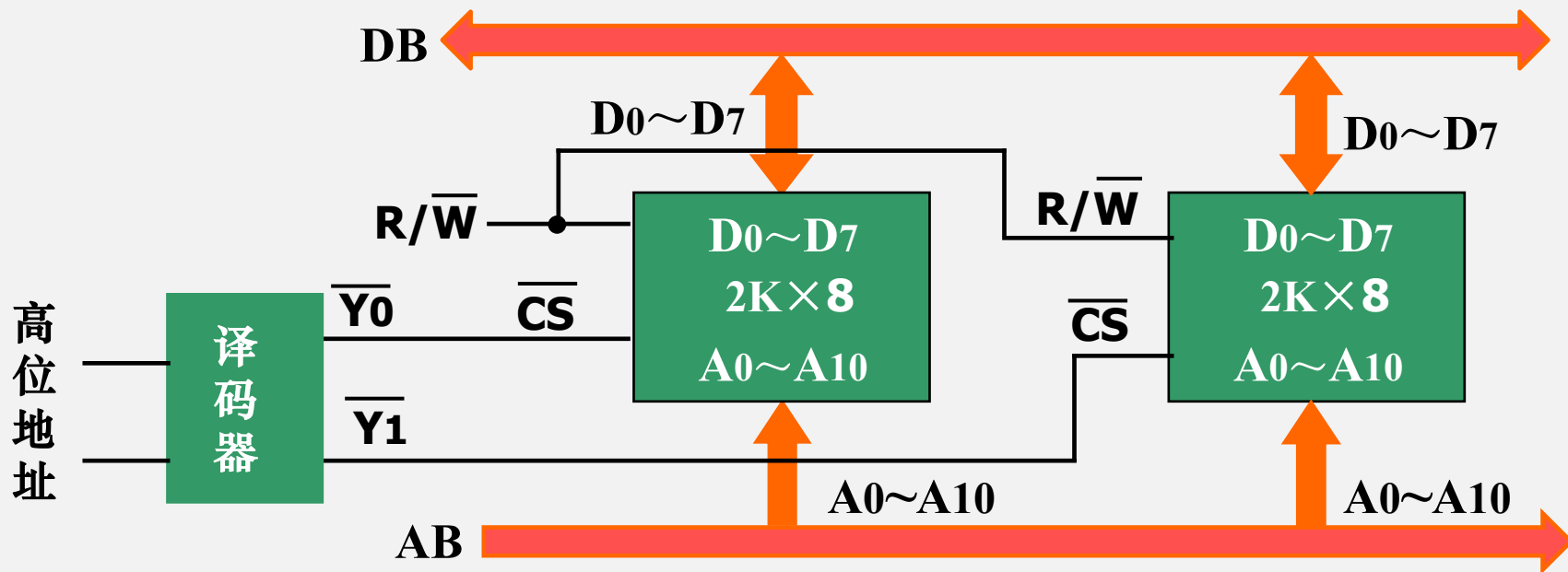
- 地址空间的扩展
- 芯片每个单元中的字长满足，但单元数不满足。

**字扩展：确保所有芯片具有完全不同的地址范围**

### ■ 扩展原则：

- 每个芯片的地址线、数据线、控制线并联。
- 片选端分别引出，以使每个芯片有不同的地址范围。

# 字扩展示意图



# 字扩展例

## ■ 例：

- 用SRAM 6264芯片构成容量为32KB的存储器
- 存储器的地址范围为：
  - 20000H~27FFFH

## ■ 设计：

- 由地址范围得：
  - 需4片6264芯片
- 存储器地址范围：

■ 001000000000000000000000 ~ 0010011111111111111111



高位地址

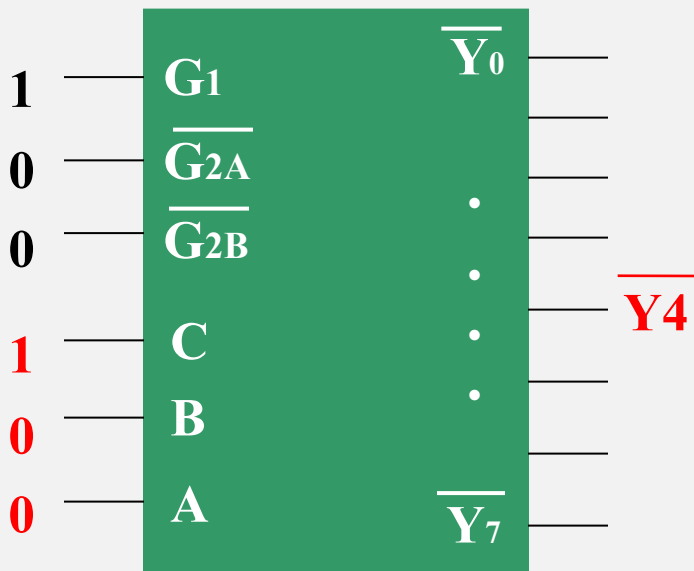
对多片芯片构成存储器，电路设计时应考虑使用专用译码器设计

# 74LS138译码器

- 73LS138:
  - 3输入8输出的专用译码器
- 可以同时控制8片芯片；
- 在任一时刻，其所连接的8片芯片只有1片被选中。

# 74LS138译码器

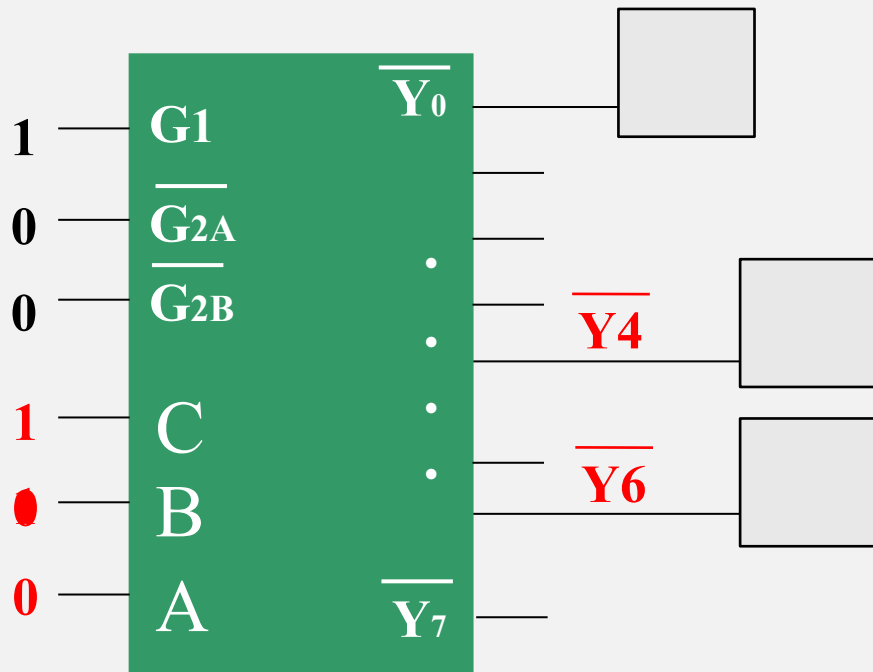
## 主要引脚及功能



使能端			输入端			输 出 端							
$G_1$	$\#G_{2A}$	$\#G_{2B}$	C	B	A	$\#Y_0$	$\#Y_1$	$\#Y_2$	$\#Y_3$	$\#Y_4$	$\#Y_5$	$\#Y_6$	$\#Y_7$
$\times$	1	1	$\times$	$\times$	$\times$	1	1	1	1	1	1	1	1
0	$\times$	$\times$	$\times$	$\times$	$\times$	1	1	1	1	1	1	1	1
1	0	0	0	0	0	0	1	1	1	1	1	1	1
1	0	0	0	0	1	1	0	1	1	1	1	1	1
1	0	0	0	1	0	1	1	0	1	1	1	1	1
1	0	0	0	1	1	1	1	1	0	1	1	1	1
1	0	0	1	0	0	1	1	1	1	0	1	1	1
1	0	0	1	0	1	1	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	1	1	1	1	1	1	1	0

# 74LS138译码器主要功能

根据输入的不同编码组合，确保其控制的各电路（芯片）在任一时刻只有一路（1个芯片）处于工作状态。



# 字扩展例

高位地址：

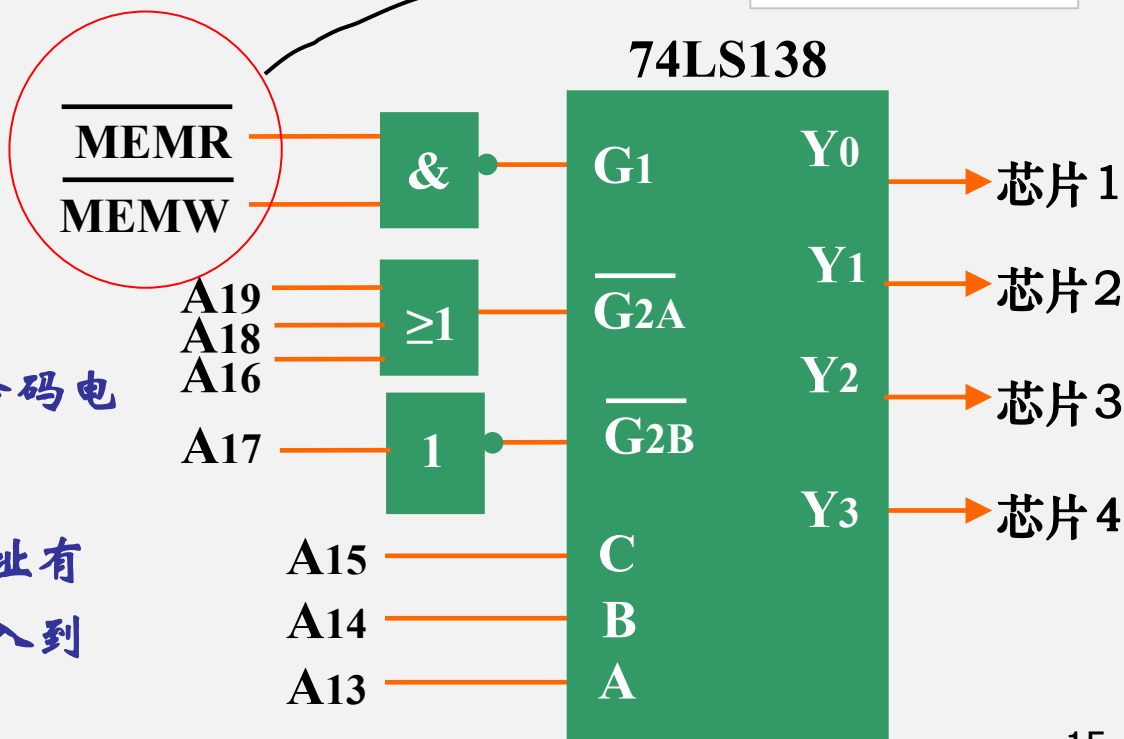
首：0010 000

尾：0010 011

A19

A13

- 利用74LS138译码器设计译码电路。
- 首地址和尾地址的高位地址有两位状态不同，须将其接入到138的输入端。





## 3. 字位扩展

- 单元数及每单元字长均不满足要求
- 设计过程：
  - 根据内存容量及芯片容量确定所需存储芯片数；
  - 进行位扩展以满足字长要求；
  - 进行字扩展以满足容量要求。

# 字位扩展例

## ■ 例：

- 用32Kb芯片构成256KB的内存。

## ■ 解：

- 首先进行位扩展：

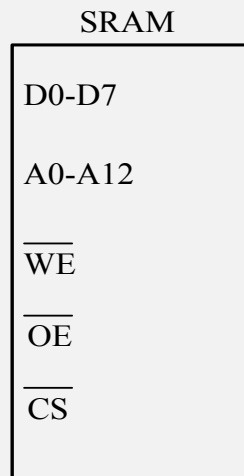
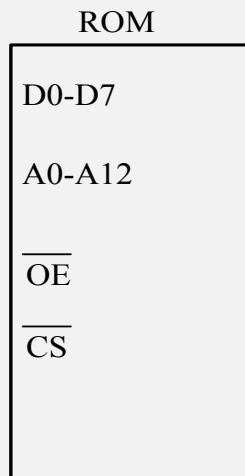
- 用8片芯片构成32KB存储体
- 利用A0~A14寻址存储体内32K个单元
- 所有控制信号线和地址信号线并联引出

- 再进行字扩展：

- 用8个32KB存储体构成256KB存储器
- 寻址8个存储体，至少需要3位高位地址信号

# 半导体存储器系统设计练习

利用如图所示的SRAM和ROM芯片构成32KB的程序存储器和16KB的数据存储器，要求程序存储器的地址范围为F0000H~F7FFFH，数据存储器的地址范围为E0000H~E3FFFH。请设计完成该存储器系统。



# 题目分析

译码地址为  
高7位地址

## ■ 由图得：

- 两种存储器芯片容量均为8KB
- 需4片ROM，2片RAM

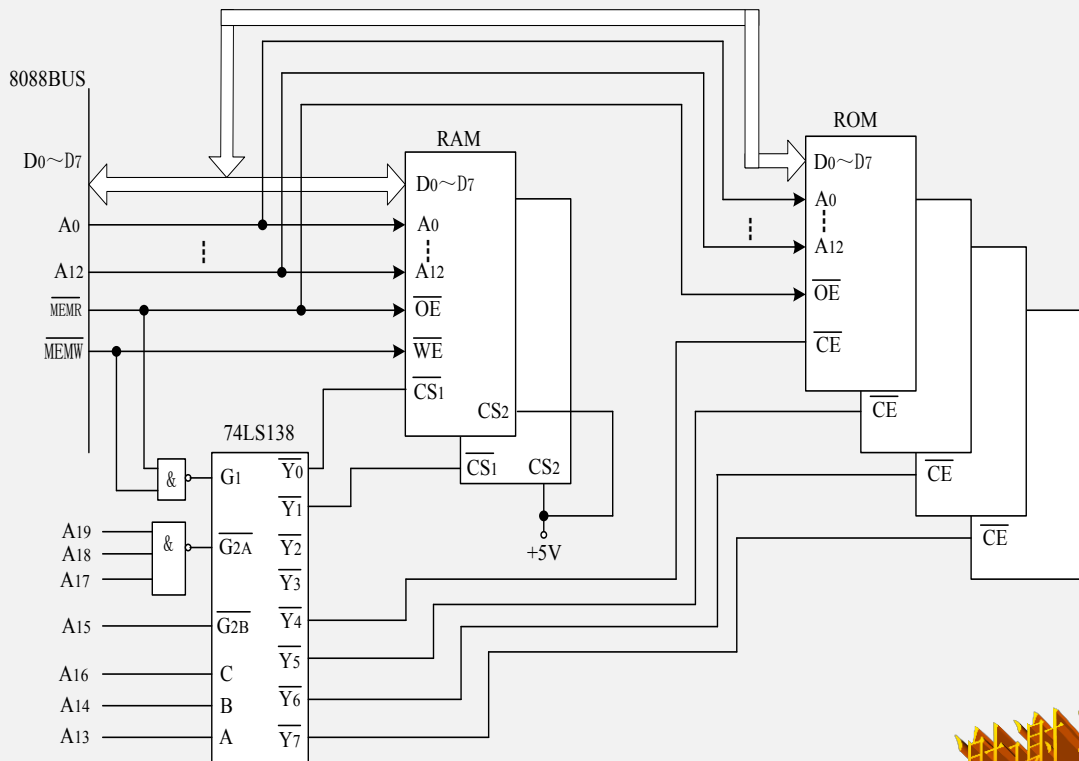
## ■ 由题目得：

- 程序存储器高位地址：

■ 1111000 ---- 1111011

- 数据存储器高位地址：

■ 1110000 ---- 1110001



谢谢



西安交通大学  
XI'AN JIAOTONG UNIVERSITY



# 半导体存储器 内容回顾



# 本章主要应掌握的知识点

## ■ 基本概念：

- ◆ 不同半导体存储器的特点及应用场合
- ◆ 微机中的存储器系统
  - Cache存储器系统
  - 虚拟存储器系统

## ■ 半导体存储器系统设计

- ◆ 存储器芯片与系统的连接
- ◆ 译码电路及其他控制信号
- ◆ 存储器扩展技术。

**能够设计出所需要的内存存储器**

# 存储器接口设计注意点：

- 片内地址用于寻址芯片上的单元，高位地址用于选择芯片（片选）
- #MEMW和#MEMR用于确保只有在对存储器芯片进行读或写操作时，译码电路才可工作。所以，它们须作为译码器输入信号。
- 74LS138译码器的使能端及输入端均不能悬空。
- 对全地址译码，要求全部高位地址都需作为译码器输入。

# SRAM存储器接口设计例


- 将 SRAM 6264 芯片与系统连接，使其地址范围为：38000H~39FFFH。
- 使用74LS138译码器构成译码电路。



# SRAM存储器接口设计例

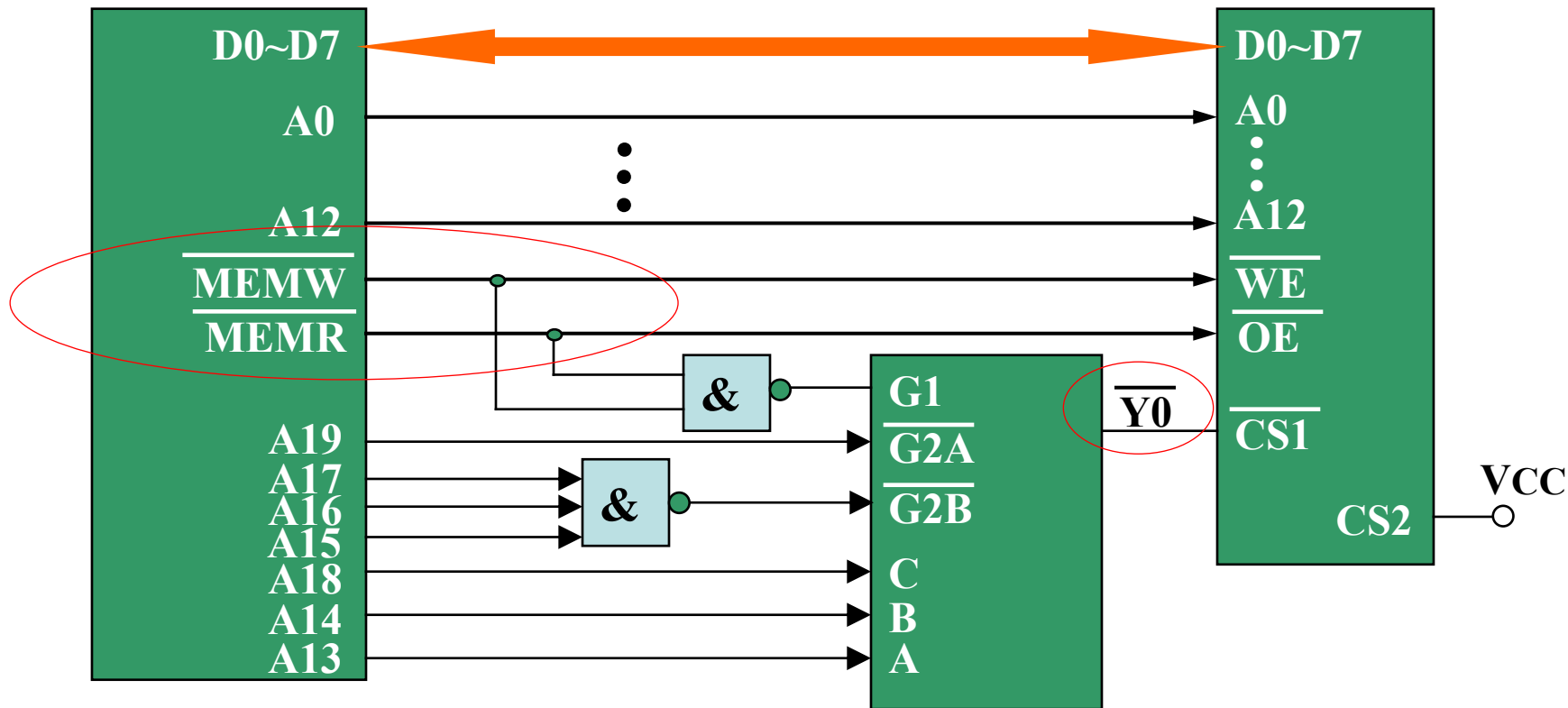
由题知地址范围：

A19							A12				A0
0	0	1	1	1	0	0	0	...	...	...	0
0	0	1	1	1	0	0	1	...	...	...	1



高位地址

# SRAM存储器接口设计例



# 半导体存储器接口设计例

- 用容量为32Kb的存储器芯片构造256KB的存储器。其地址范围：

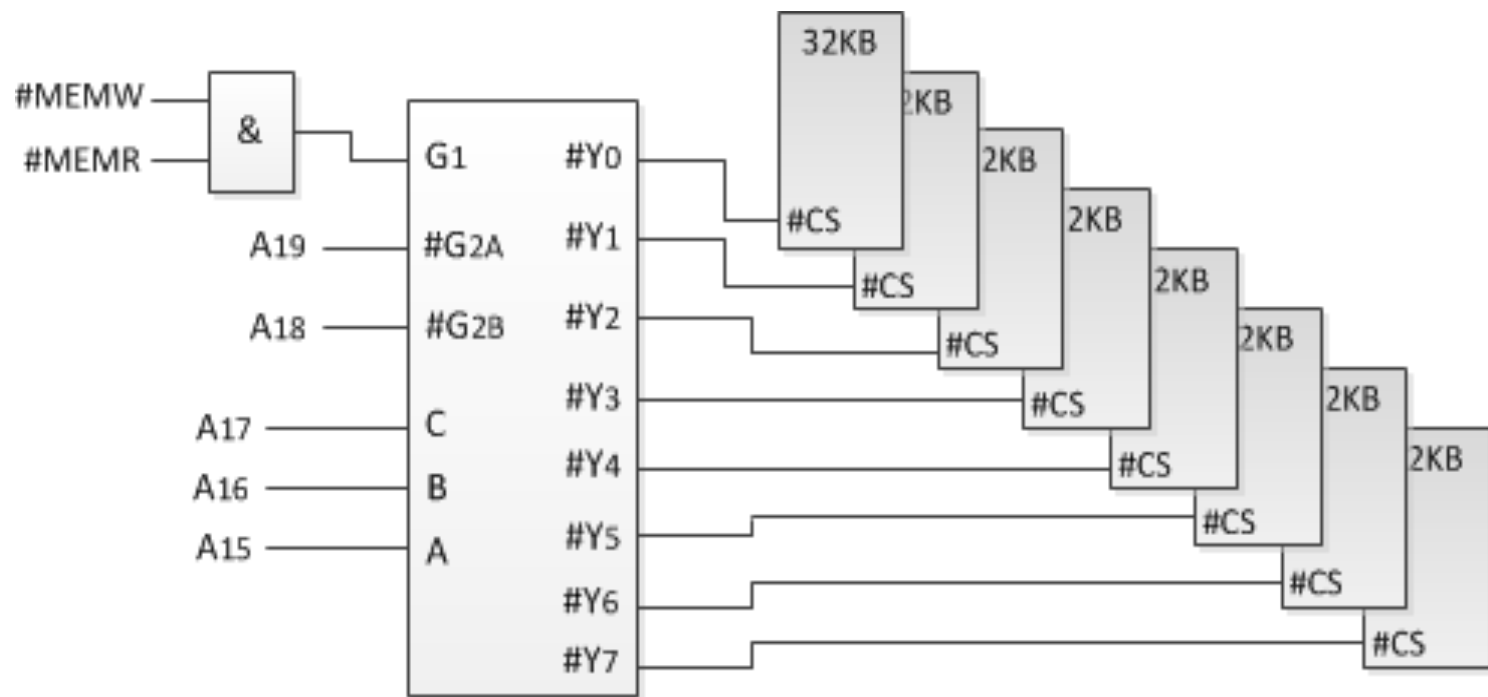
- ◆ 00000~07FFFH
- ◆ 08000~0FFFFH
- ◆ 10000~17FFFH
- ◆ 18000~1FFFFH
- ◆ 20000~27FFFH
- ◆ 28000~2FFFFH
- ◆ 30000~37FFFH
- ◆ 38000~3FFFFH

- ① 用8片32Kb芯片构成32KB存储体
- ② 8个存储体构成256KB存储器
- ③ 寻址32KB存储体需要15位片内地址

片选地址为高5位地址：A19—A15

*00000~00111*

# 存储器接口



谢谢