

微机原理与接口技术

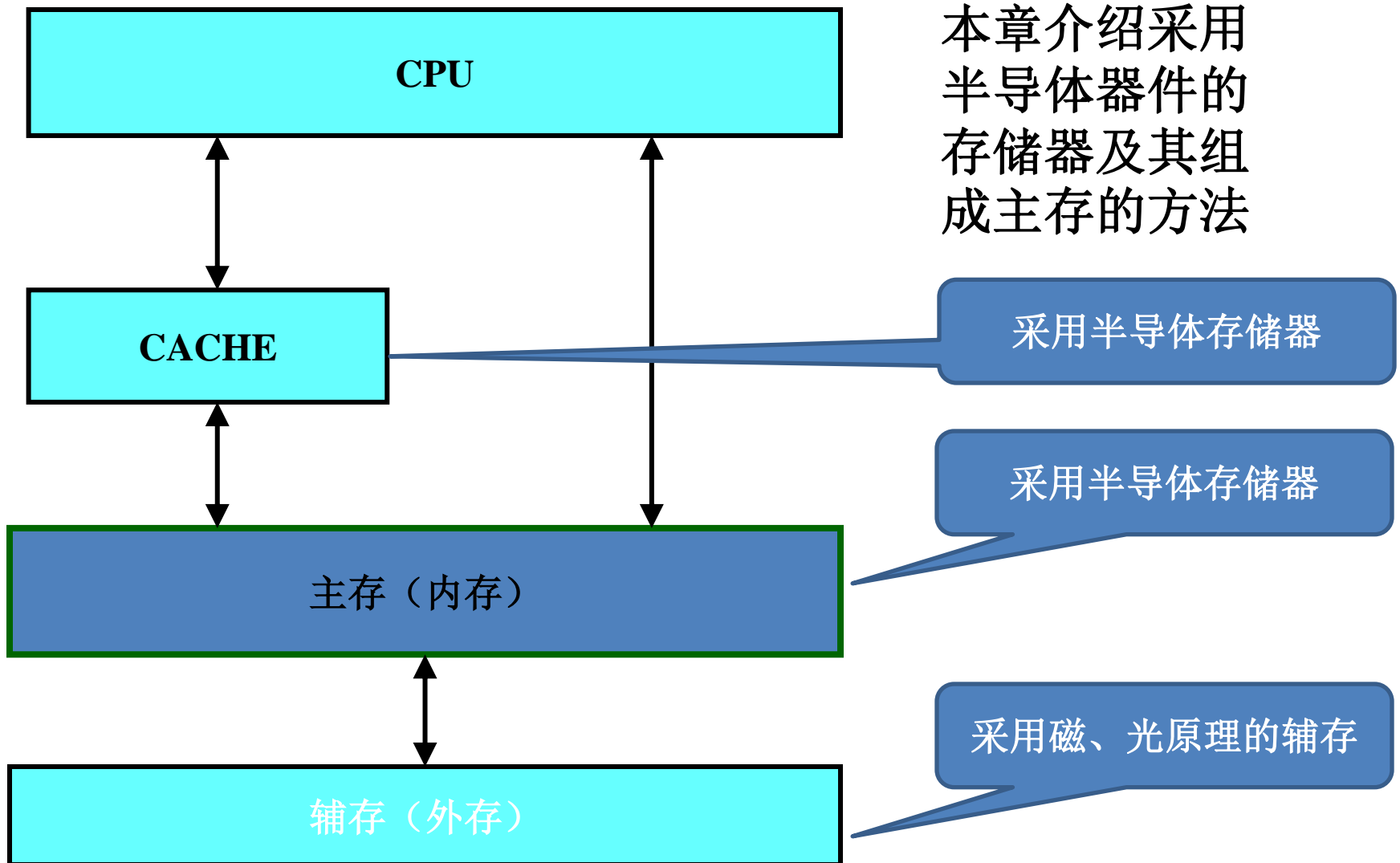
第五章 半导体存储器及接口

第五章 半导体存储器及接口

- 教学重点

- 芯片SRAM 2114和DRAM 4116
- 芯片EPROM 2764和EEPROM 2817A
- SRAM、EPROM与CPU的连接

(1) 存储系统结构



(2) 半导体存储器的分类

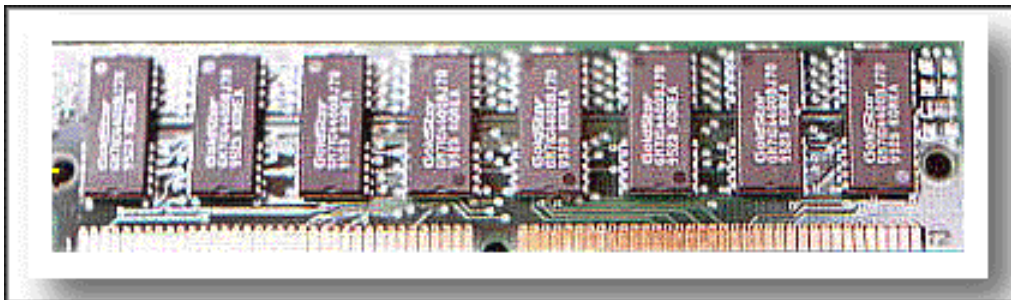
- 按制造工艺
 - 双极型：速度快、集成度低、功耗大
 - MOS型：速度慢、集成度高、功耗低
- 按使用属性
 - 随机存取存储器RAM：可读可写、断电丢失
 - 只读存储器ROM：正常只读、断电不丢失

(2) 半导体存储器的分类



1) 读写存储器RAM

	组成单元	速度	集成度	应用
SRAM	触发器	快	低	小容量系统
DRAM	极间电容	慢	高	大容量系统
NVRAM	带微型电池	慢	低	小容量非易失



2) 只读存储器ROM

- **掩膜ROM**: 信息制作在芯片中, 不可更改
- **PROM**: 允许一次编程, 此后不可更改
- **EPROM**: 用紫外光擦除, 擦除后可编程; 并允许用户多次擦除和编程
- **EEPROM (E²PROM)**: 采用加电方法在线进行擦除和编程, 也可多次擦写
- **Flash Memory (闪存)**: 能够快速擦写的EEPROM, 但只能按块 (Block) 擦除

(3) 半导体存储器芯片的内部结构

① 存储体

- 存储器芯片的主要部分，用来存储信息

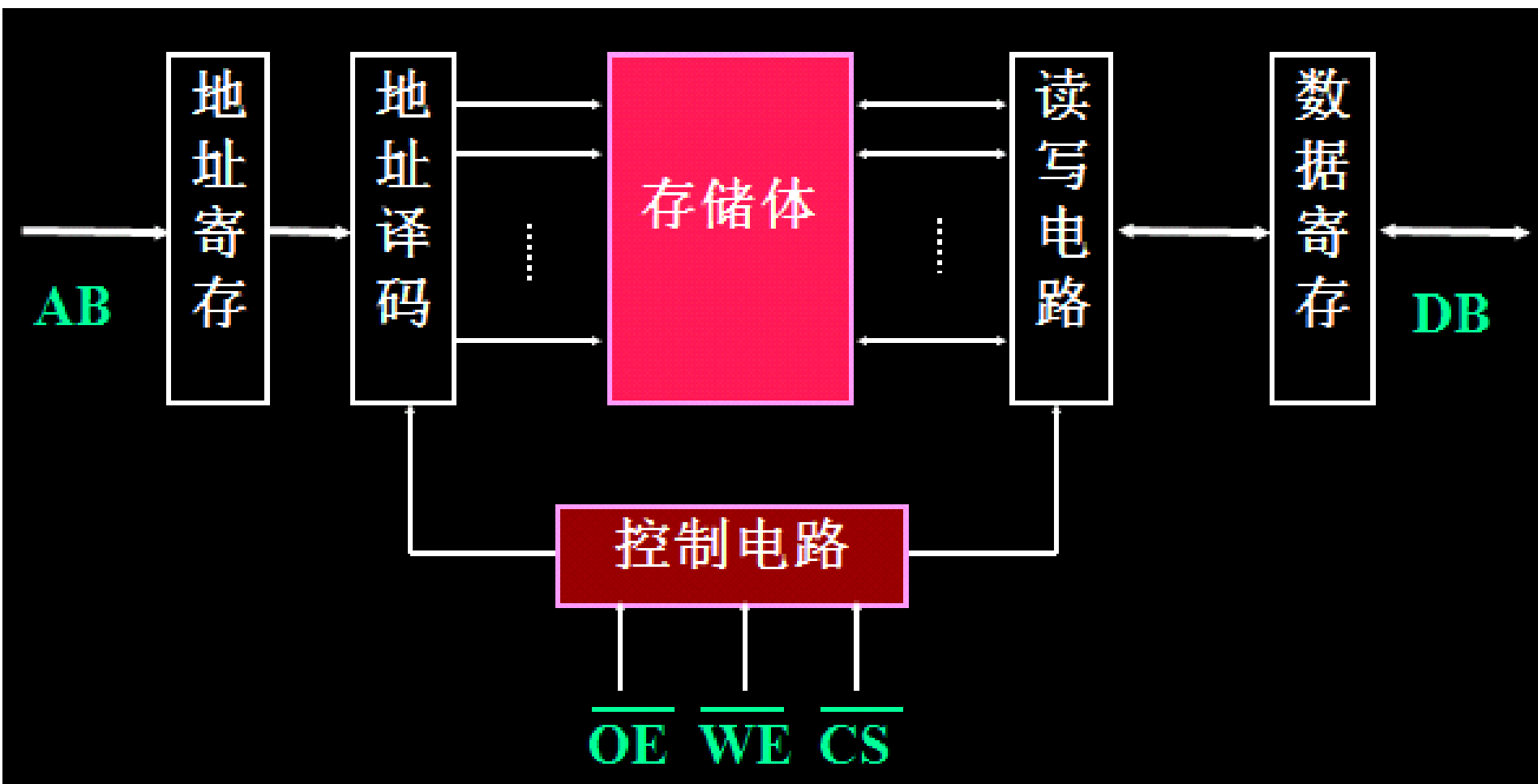
② 地址译码电路

- 根据输入的地址编码来选中芯片内某个特定的存储单元

③ 片选和读写控制逻辑

- 选中存储芯片，控制读写操作

(3) 半导体存储器芯片的内部结构



1) 存储体内部结构概述

- 每个存储单元具有一个唯一的地址，可存储1位（位片结构）或多位（字片结构）二进制数据
- 存储容量（bits）与地址、数据线个数有关：
芯片的存储容量 = $2^M \times N$
= 存储单元数 × 存储单元的位数

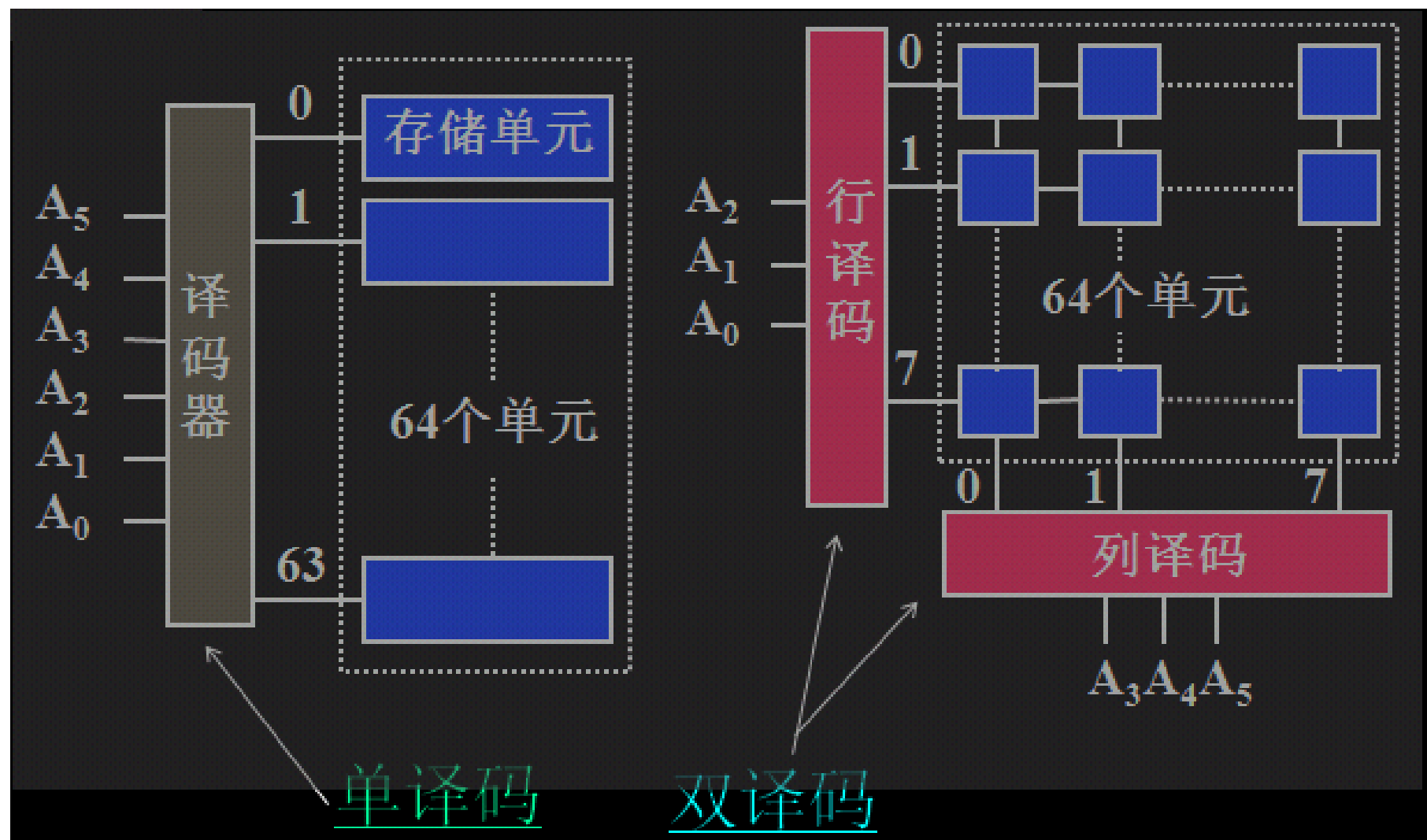
M: 芯片的地址线根数

N: 芯片的数据线根数

2) 地址译码电路结构

- 单译码结构
- 双译码结构
 - 双译码可简化芯片设计
 - 主要采用的译码结构

2) 地址译码电路结构



3) 片选和读写控制逻辑

- 片选端**CS***或**CE***
 - 有效时，可以对该芯片进行读写操作
- 输出**OE***
 - 控制读操作。有效时，芯片内数据输出
 - 该控制端对应系统的读控制线
- 写**WE***
 - 控制写操作。有效时，数据进入芯片中
 - 该控制端对应系统的写控制线

(4) 随机存取存储器

静态RAM

SRAM 2114

SRAM 6264

动态RAM

DRAM 4116

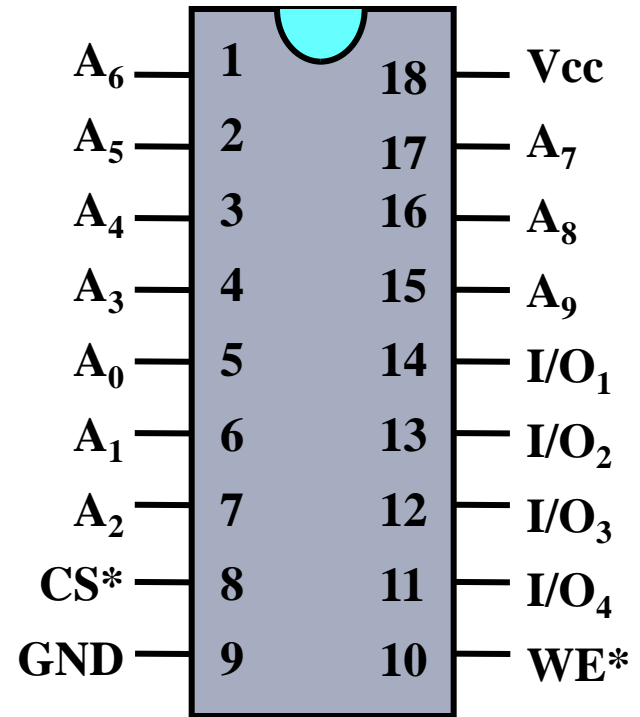
DRAM 2164

1) 静态RAM (SRAM)

- SRAM的基本存储单元是**触发器电路**
- 每个基本存储单元存储二进制数一位
- 许多个基本存储单元形成行列存储矩阵
- SRAM一般采用“**字结构**”存储矩阵：
 - 每个存储单元存放多位（4、8、16等）
 - 每个存储单元具有一个地址

SRAM芯片2114外部特性

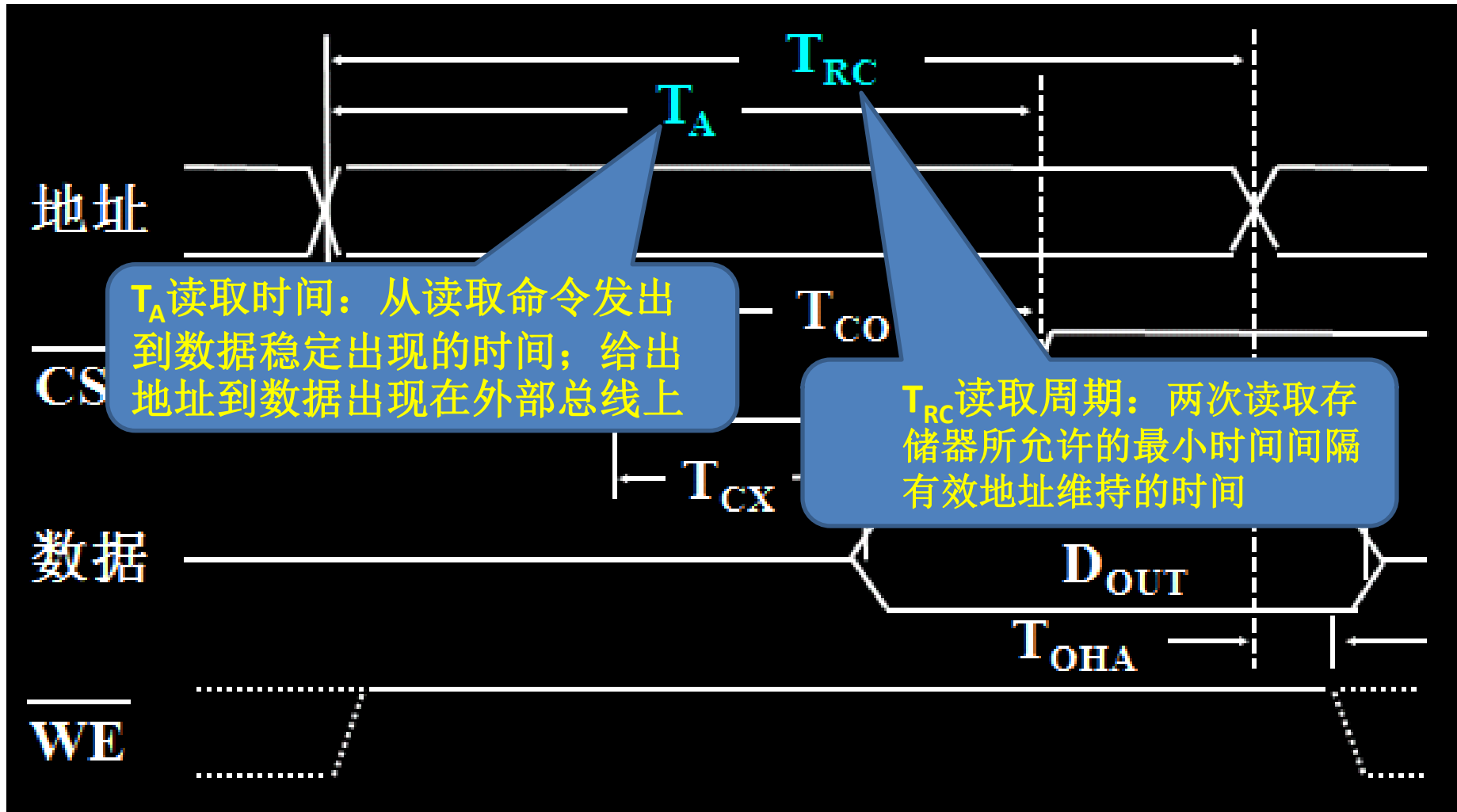
- 存储容量为 1024×4
- 18个引脚：
 - 10根地址线 $A_9 \sim A_0$
 - 4根数据线 $I/O_4 \sim I/O_1$
 - 片选 CS^*
 - 读写 WE^*



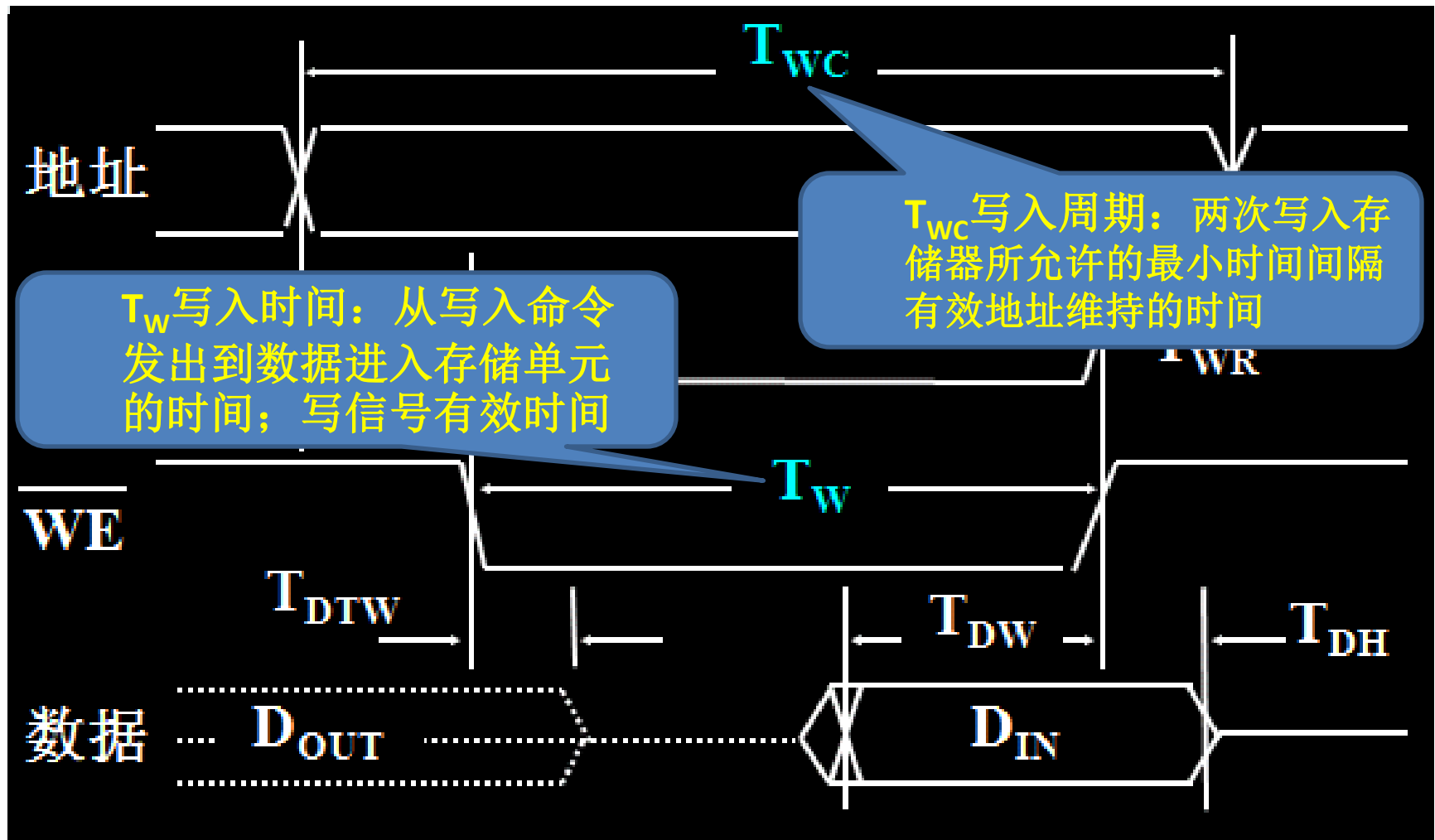
SRAM 2114的外部特性

工作方式	CS*	WE*	I/O4~I/O1
未选中	1	×	高阻
读操作	0	1	输出
写操作	0	0	输入

SRAM芯片2114读周期

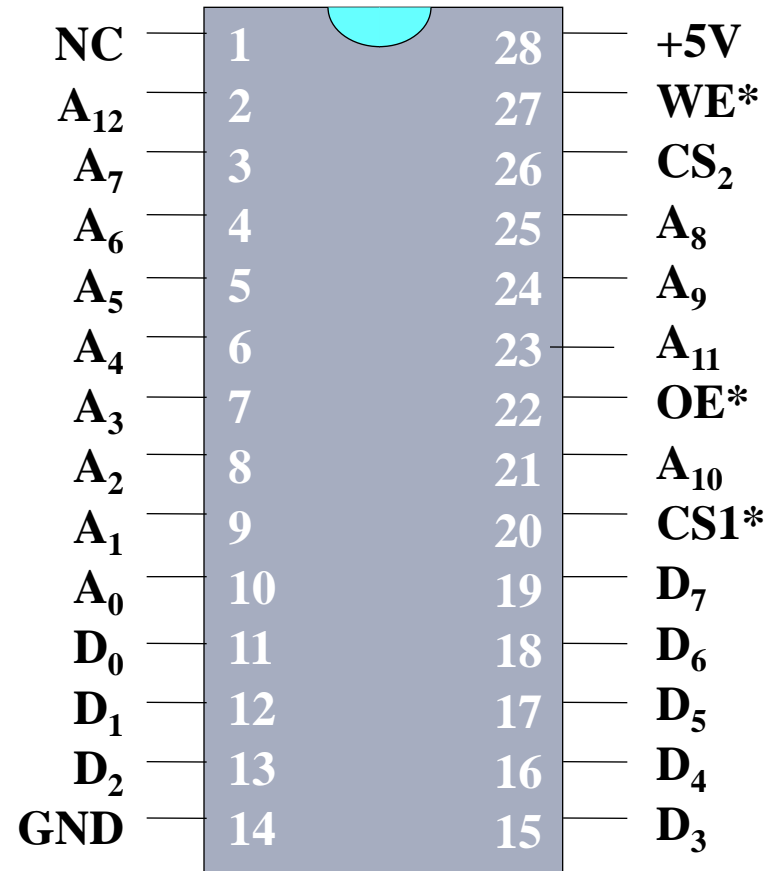


SRAM 2114的写周期



SRAM芯片6264外部特性

- 存储容量为 $8K \times 8$
- 28个引脚：
 - 13根地址线 $A_{12} \sim A_0$
 - 8根数据线 $D_7 \sim D_0$
 - 片选 $CS1^*$ 、 $CS2$
 - 读写 WE^* 、 OE^*



SRAM 6264的外部特性

工作方式	CS1*	CS2	WE*	OE*	D7~D0
未选中	1	×	×	×	高阻
未选中	×	0	×	×	高阻
读操作	0	1	1	0	输出
写操作	0	1	0	1	输入

2) 动态RAM

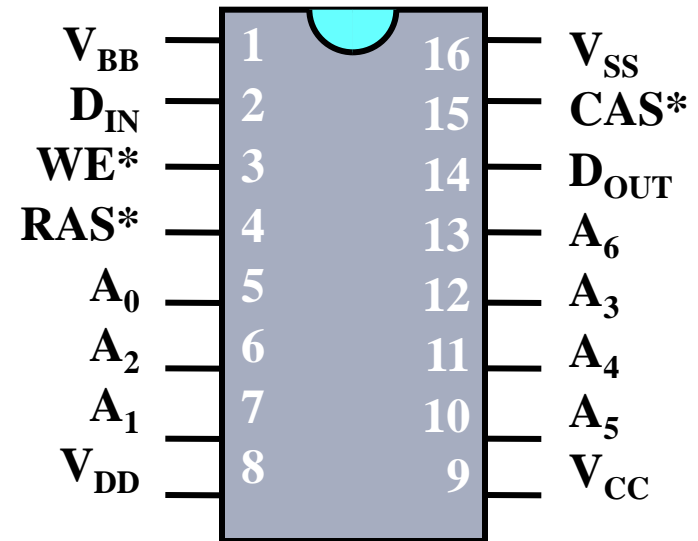
- DRAM的基本存储单元是单个场效应管及其极间电容
- 必须配备“读出再生放大电路”进行刷新
- 每次同时对一行的存储单元进行刷新
- 每个基本存储单元存储二进制数一位
- 许多个基本存储单元形成行列存储矩阵

2) 动态RAM

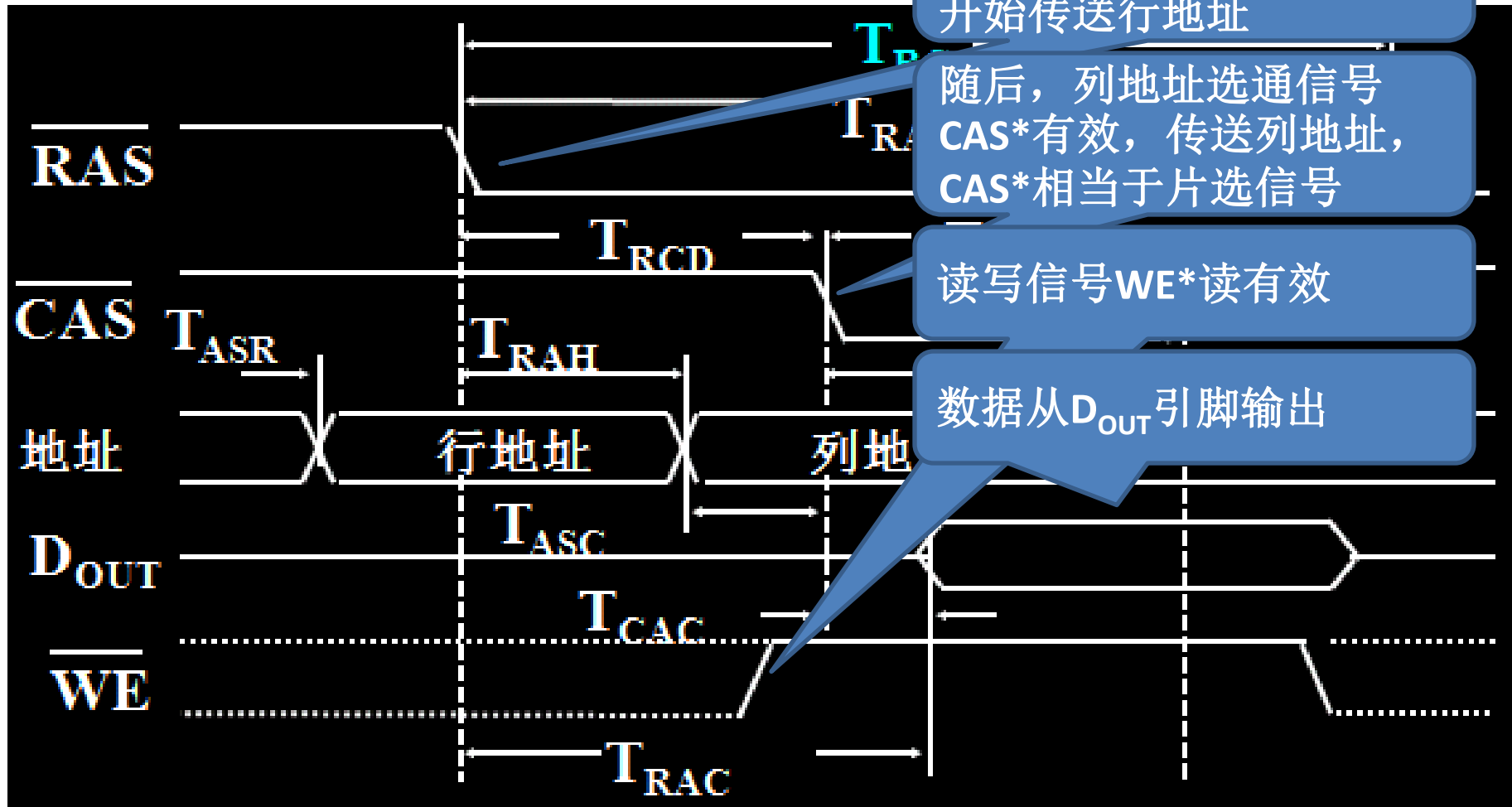
- DRAM一般采用“位结构”存储体：
 - 每个存储单元存放一位，芯片内每个位单元具有独立地址
 - 需要8个存储芯片构成一个字节单元，每个字节存储单元具有一个地址

DRAM芯片4116外部特性

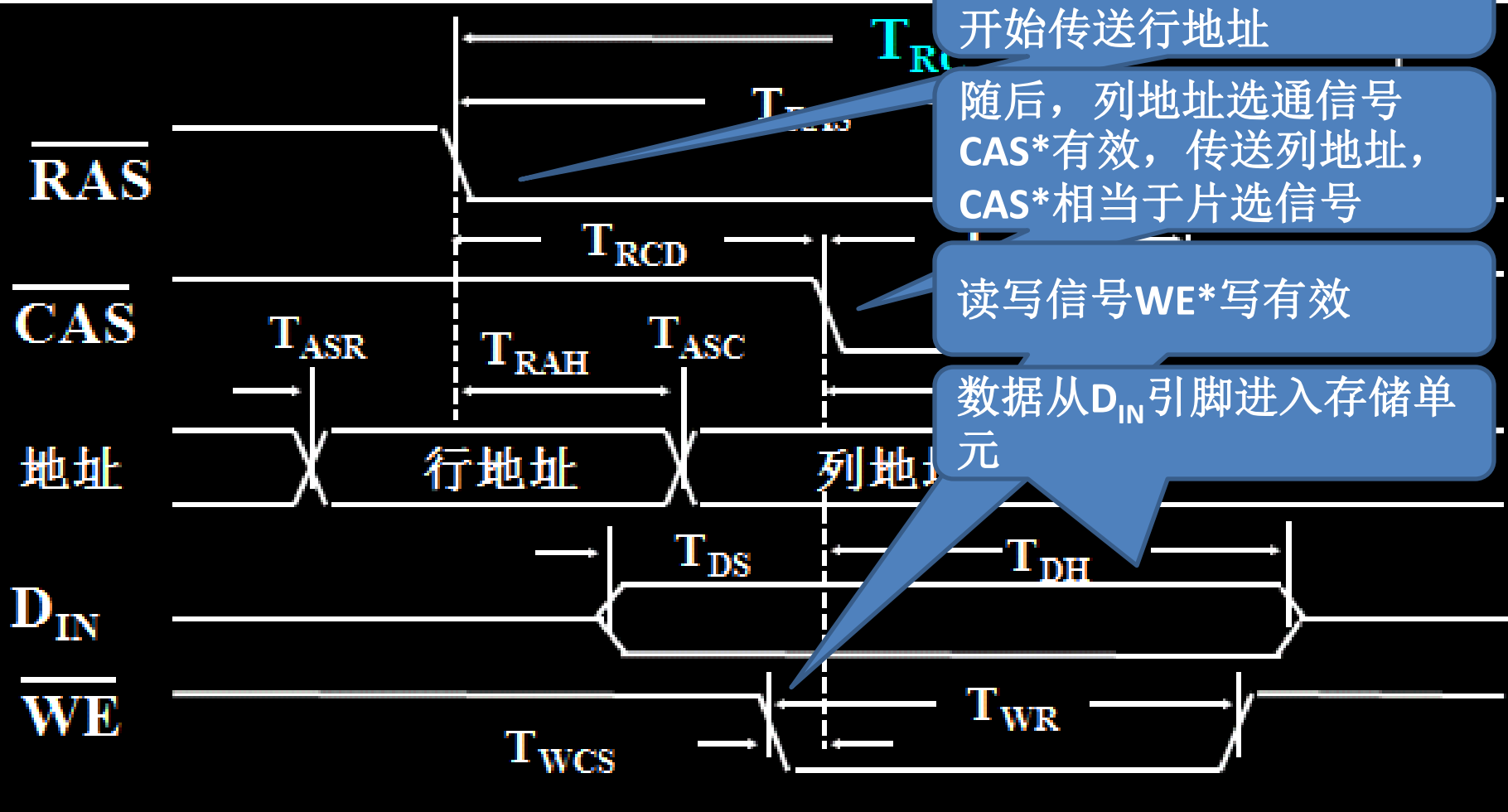
- 存储容量为 $16K \times 1$
- 16个引脚：
 - 7根地址线 $A_6 \sim A_0$
 - 1根数据输入线 D_{IN}
 - 1根数据输出线 D_{OUT}
 - 行地址选通 RAS^*
 - 列地址选通 CAS^*
 - 读写控制 WE^*



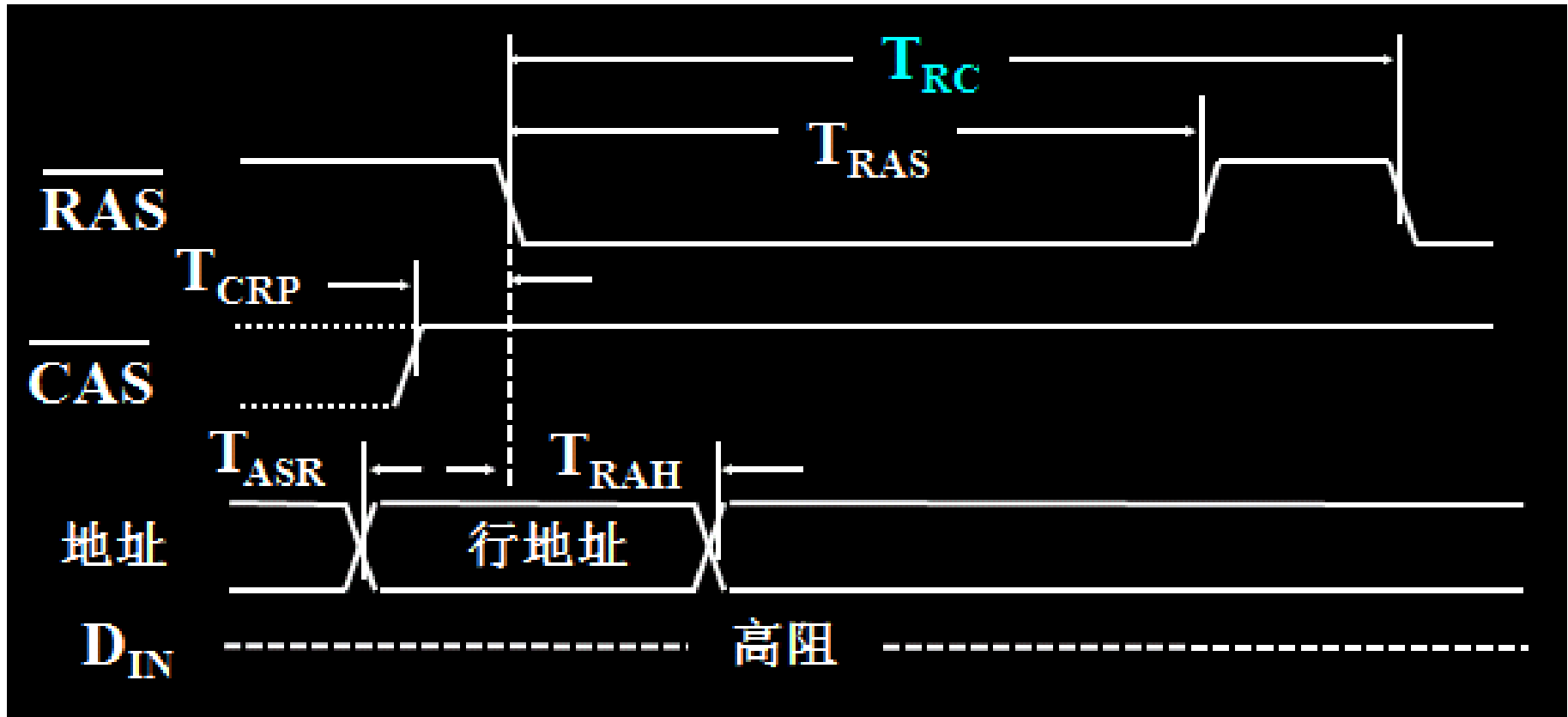
DRAM 4116的读写时序



DRAM 4116的引脚图



DRAM 4116的刷新



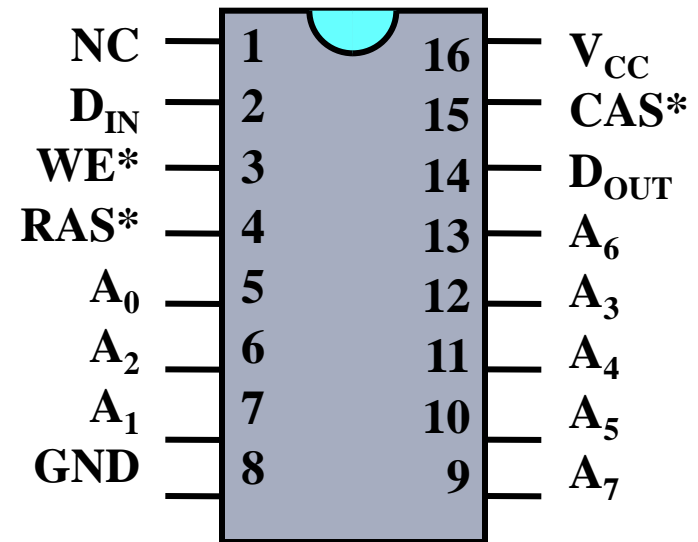
DRAM 4116的刷新

采用“仅行地址有效”方法刷新

- 行地址选通RAS*有效，传送行地址
- 列地址选通CAS*无效，没有列地址
- 芯片内部实现一行存储单元的刷新
- 没有数据输入输出
- 存储系统中所有芯片同时进行刷新
- DRAM必须每隔固定时间就刷新

DRAM芯片2164外部特性

- 存储容量为 $64K \times 1$
- 16个引脚：
 - 8根地址线 $A_7 \sim A_0$
 - 1根数据输入线 D_{IN}
 - 1根数据输出线 D_{OUT}
 - 行地址选通 RAS^*
 - 列地址选通 CAS^*
 - 读写控制 WE^*



(5) 只读存储器

EPROM

EPROM 2716

EPROM 2764

EEPROM

EEPROM 2717A

EEPROM 2864A

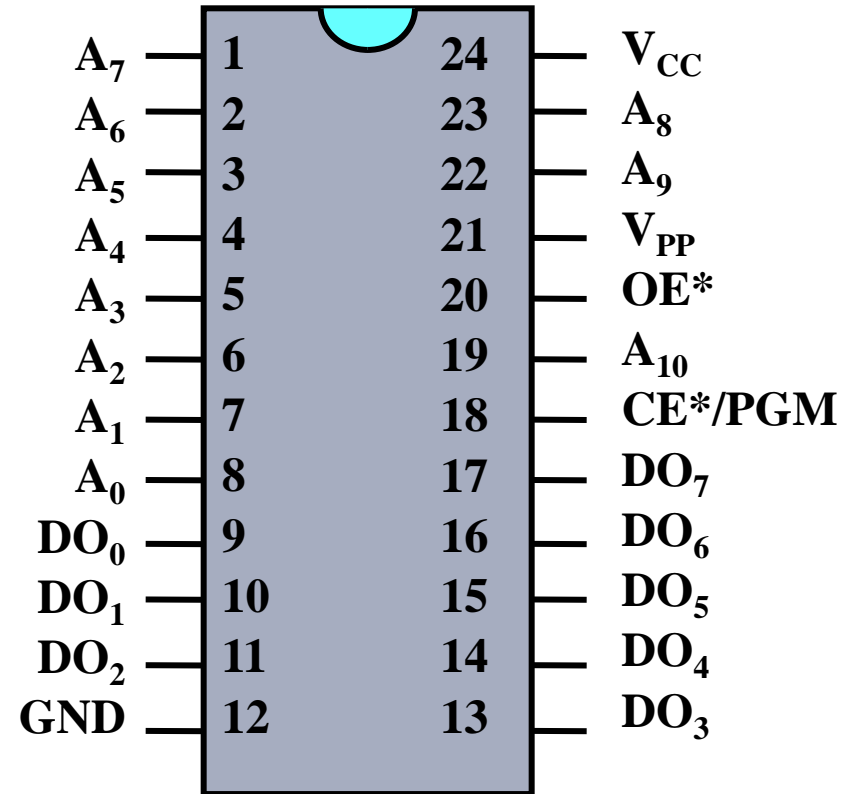
1) EPROM



- 顶部开有一个圆形的石英窗口，用于紫外线透过擦除原有信息
- 一般使用专门的编程器（烧写器）进行编程
- 编程后，应该贴上不透光封条
- 出厂未编程前，每个基本存储单元都是信息1
- 编程就是将某些单元写入信息0

EPROM芯片2716

- 存储容量为 $2K \times 8$
- 24个引脚：
 - 11根地址线 $A_{10} \sim A_0$
 - 8根数据线 $DO_7 \sim DO_0$
 - 片选/编程 CE^*/PGM
 - 读写 OE^*
 - 编程电压 V_{PP}

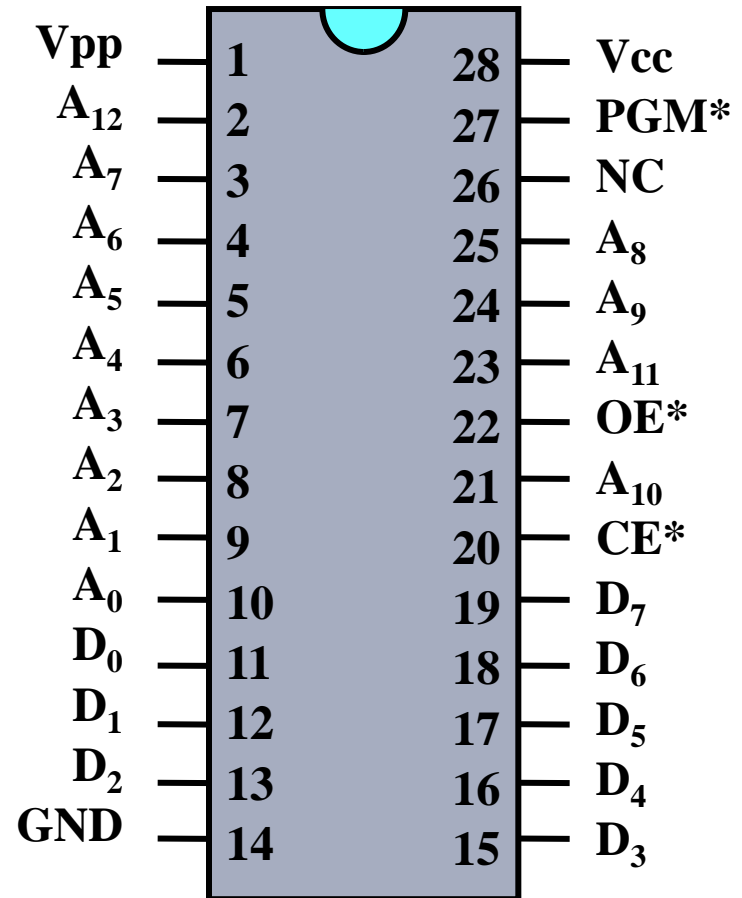


EPROM芯片2716

工作方式	CE*/PGM	OE*	V _{CC}	V _{PP}	DO ₇ ~DO ₀
待用	1	×	+5V	+5V	高阻
读出	0	0	+5V	+5V	输出
读出禁止	0	1	+5V	+5V	高阻
编程写入	正脉冲	1	+5V	+25V	输入
编程校验	0	0	+5V	+25V	输出
编程禁止	0	1	+5V	+25V	高阻

EPROM芯片2764

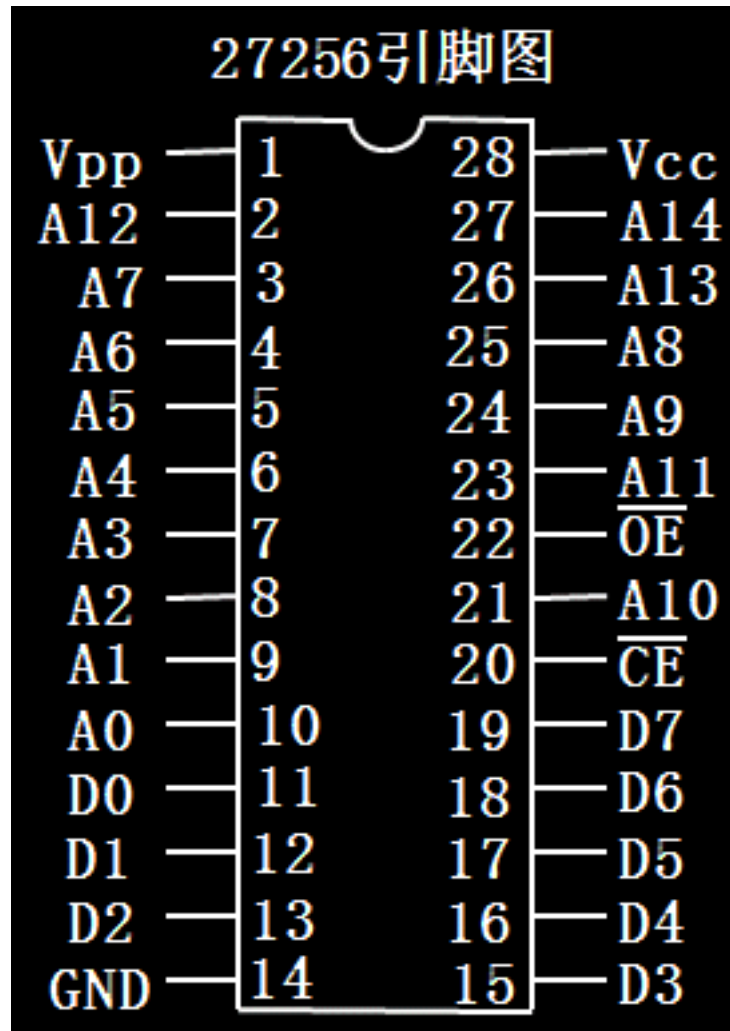
- 存储容量为 $8K \times 8$
- 28个引脚：
 - 13根地址线 $A_{12} \sim A_0$
 - 8根数据线 $D_7 \sim D_0$
 - 片选 CE^*
 - 编程 PGM^*
 - 读写 OE^*
 - 编程电压 V_{PP}



EPROM芯片2764

工作方式	CE*	OE*	PGM*	A ₉	V _{PP}	DO ₇ ~DO ₀
读出	0	0	1	×	+5V	输出
读出禁止	0	1	1	×	+5V	高阻
待用	1	×	×	×	+5V	高阻
Intel标识	0	0	+12V	1	+5V	输出编码
标准编程	0	1	负脉冲	×	+25V	输入
Intel编程	0	1	负脉冲	×	+25V	输入
编程校验	0	0	1	×	+25V	输出
编程禁止	1	×	×	×	+25V	高阻

EPROM芯片27256

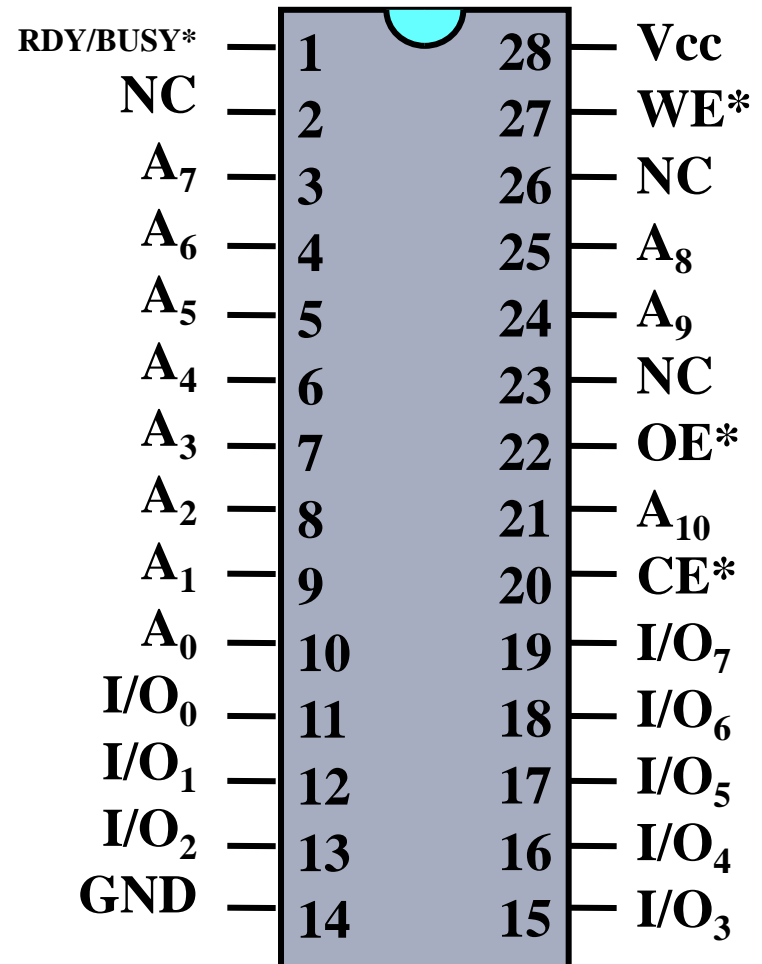


2) EEPROM

- 用加电方法，进行在线（无需拔下，直接在电路中）擦写（擦除和编程一次完成）
- 有字节擦写、块擦写和整片擦写方法
- 并行EEPROM：多位同时进行
- 串行EEPROM：只有一位数据线

EEPROM芯片2817A

- 存储容量为 $2K \times 8$
- 28个引脚：
 - 11根地址线 $A_{10} \sim A_0$
 - 8根数据线 $I/O_7 \sim I/O_0$
 - 片选 CE^*
 - 读写 OE^* 、 WE^*
 - 状态输出 $RDY/BUSY^*$

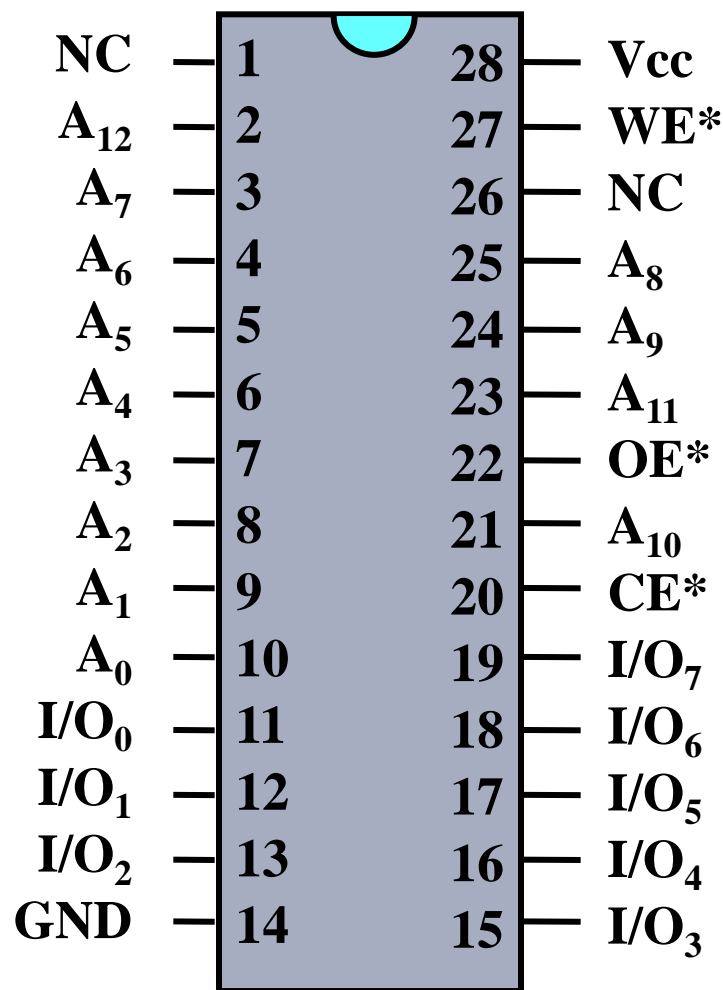


EEPROM芯片2817A

工作方式	CE*	OE*	WE*	RDY/BUSY*	I/O ₇ ~I/O ₀
读出	0	0	1	高阻	输出
维持	1	×	×	高阻	高阻
字节写入	0	1	0	0	输入

EEPROM芯片2864A

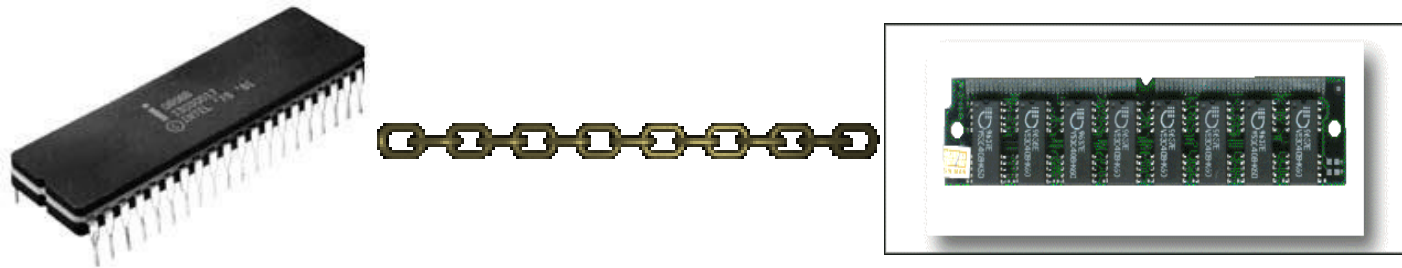
- 存储容量为 $8K \times 8$
- 28个引脚：
 - 13根地址线 $A_{12} \sim A_0$
 - 8根数据线 $I/O_7 \sim I/O_0$
 - 片选 CE^*
 - 读写 OE^* 、 WE^*



EEPROM芯片2864A

工作方式	CE*	OE*	WE*	I/O ₇ ~I/O ₀
读出	0	0	1	输出
维持	1	×	×	高阻
写入	0	1	负脉冲	输入
数据查询	0	0	1	输出

(6) 半导体存储器与CPU的连接



- SRAM、EPROM与CPU的连接是本章的重点内容
- 译码方法同样适合I/O端口

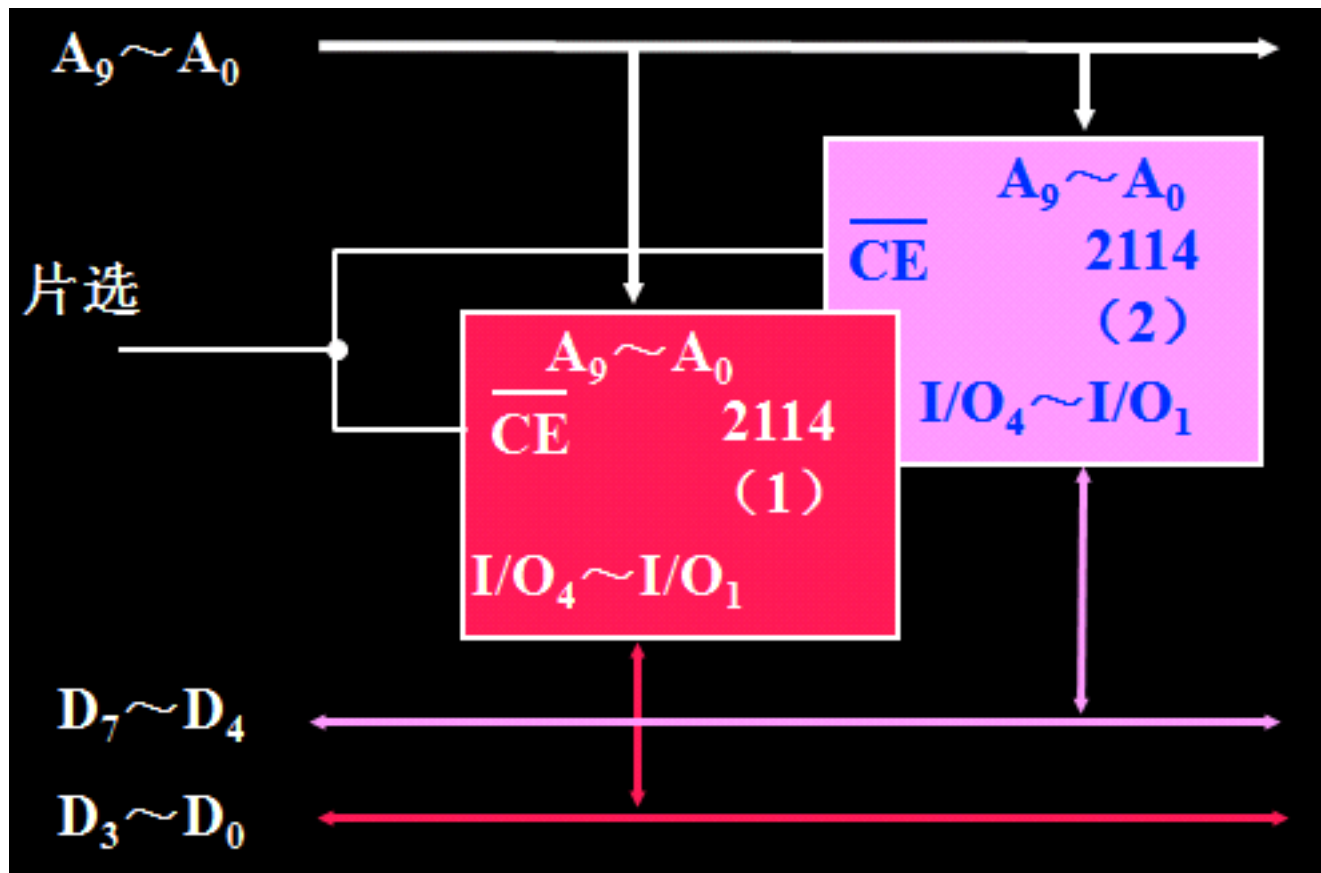
（6）半导体存储器与CPU的连接

- 与连接相关的内容：
- 存储芯片的数据线
- 存储芯片的地址线
- 存储芯片的片选端
- 存储芯片的读写控制线

1) 存储芯片数据线的处理

- 若芯片的数据线正好8根：
 - 一次可从芯片中访问到8位数据
 - 全部数据线与系统的8位数据总线相连
- 若芯片的数据线不足8根：
 - 一次不能从一个芯片中访问到8位数据
 - 利用多个芯片扩充数据位
 - 这个扩充方式简称“位扩充”

位扩充




位扩充

- 多个位扩充的存储芯片的数据线连接于系统数据总线的不同位数
- 其它连接都一样
- 这些芯片应被看作是一个整体，常被称为“芯片组”

2) 存储芯片地址线的连接

- 芯片的地址线通常应全部与系统的低位地址总线相连
- 寻址时，这部分地址的译码是在存储芯片内完成的，我们称为“片内译码”

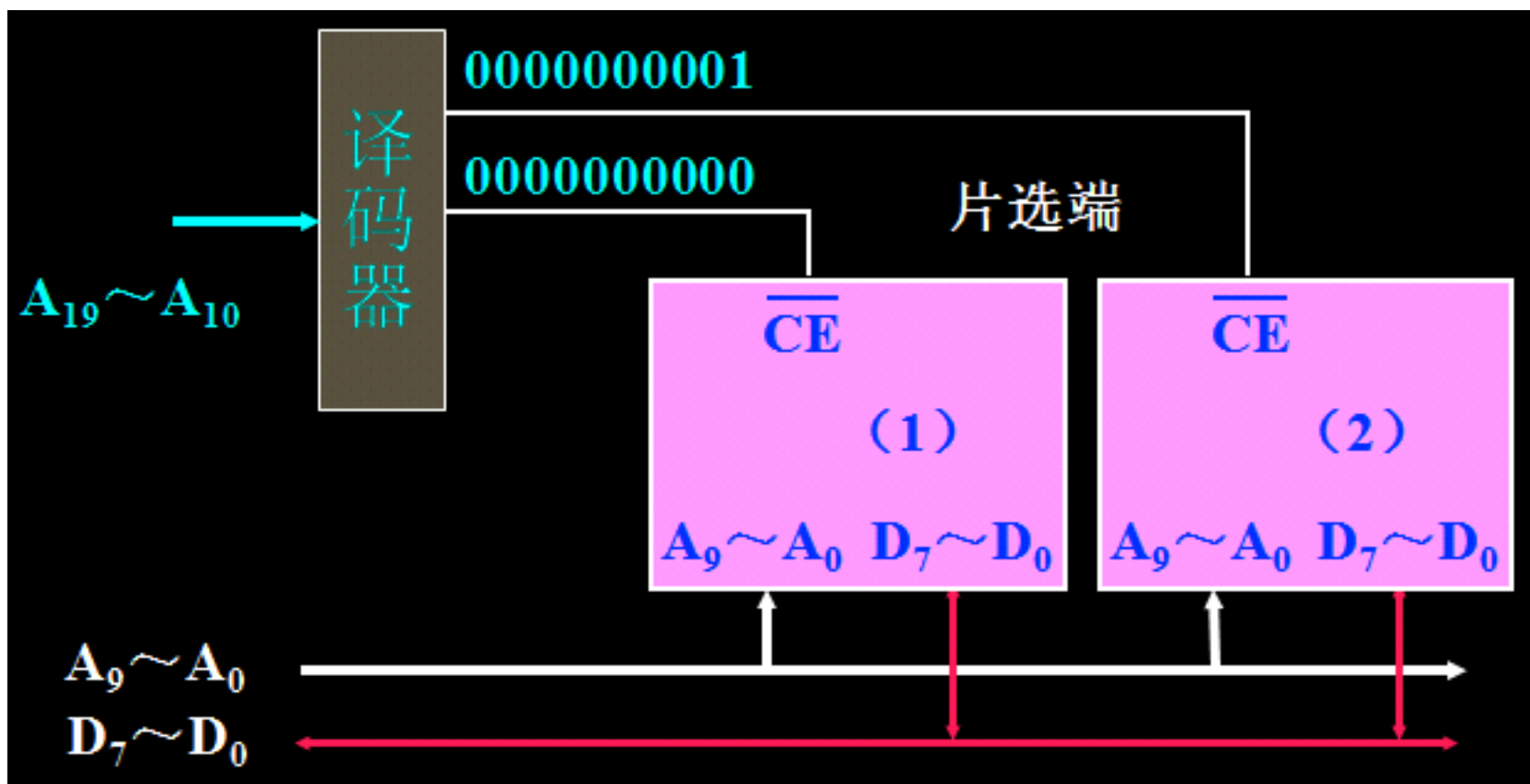
片内译码

$A_9 \sim A_0$		范围（16进制）
00...00	全0	000H
00...01		001H
00...10		002H
...		...
11...01		3FDH
11...10		3FEH
11...11	全1	3FFH

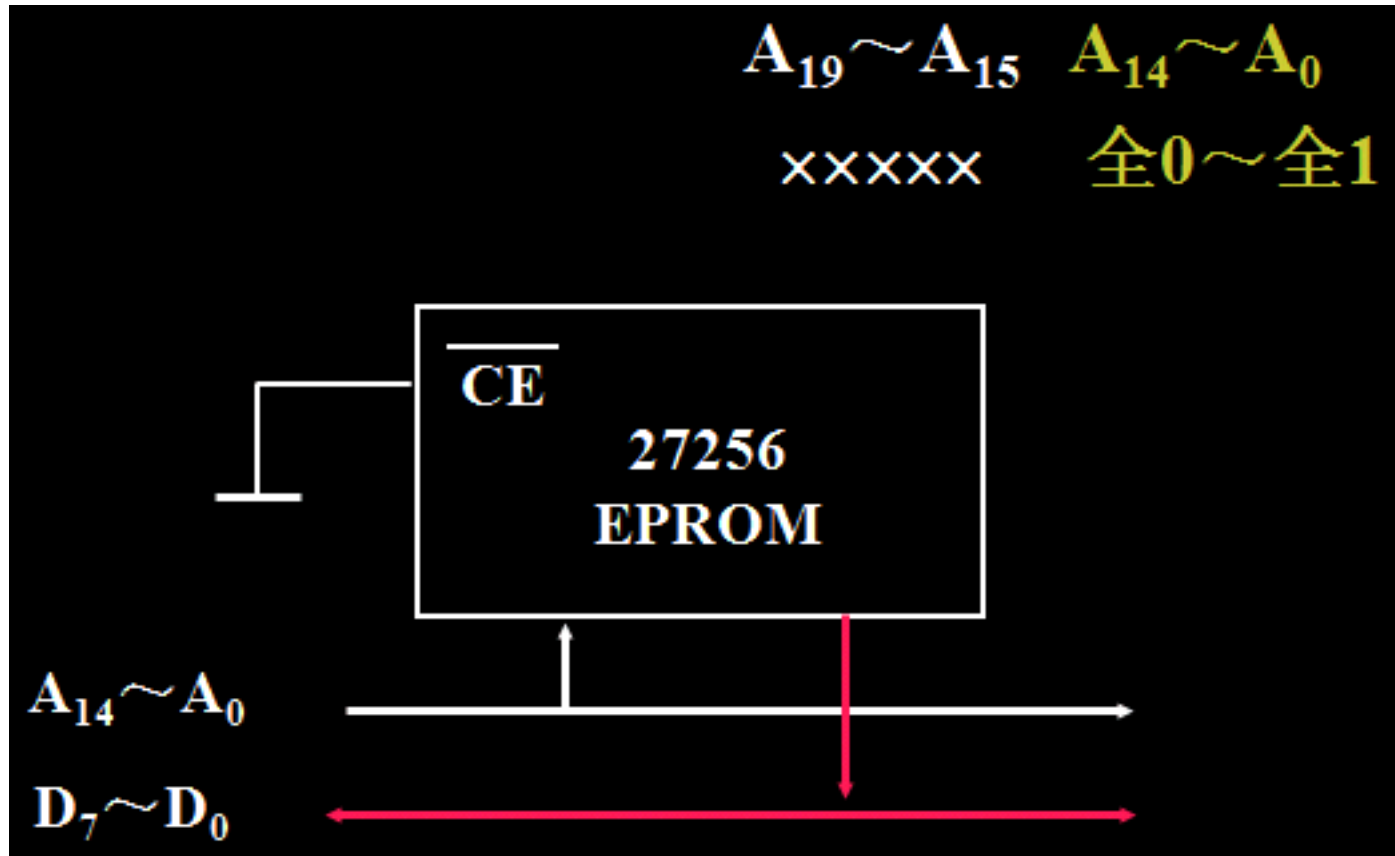
3) 存储芯片片选端的译码

- 存储系统常需利用多个存储芯片扩充容量，也就是扩充了存储器地址范围
- 进行“地址扩充”，需要利用存储芯片的片选端对多个存储芯片（组）进行寻址
- 这个寻址方法，主要通过将存储芯片的片选端与系统的高位地址线相关联来实现
- 这种扩充简称为“地址扩充”或“字扩充”

片选端译码：地址扩充（字扩充）



片选端常有效



片选端常有效

- 令芯片（组）的片选端常有效
- 不与系统的高位地址线发生联系
- 芯片（组）总处在被选中的状态
- 虽简单易行、但无法再进行地址扩充，会出现“地址重复”

地址重复

- 一个存储单元具有多个存储地址的现象
- 原因：有些高位地址线没有用、可任意
- 使用地址：出现地址重复时，常选取其中既好用、又不冲突的一个“可用地址”
- 例如：00000H~07FFFH
- 选取的原则：高位地址全为0的地址

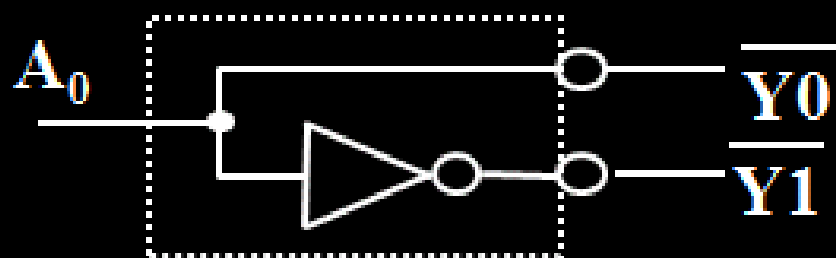
存储芯片片选端的功能

- 存储芯片的片选控制端可以被看作是一根最高位地址线
- 在系统中，主要与地址发生联系：包括**地址空间的选择**（接系统的IO/M*信号）和**高位地址的译码选择**（与系统的高位地址线相关联）
- 对一些存储芯片通过片选无效可关闭内部的输出驱动机制，起到降低功耗的作用

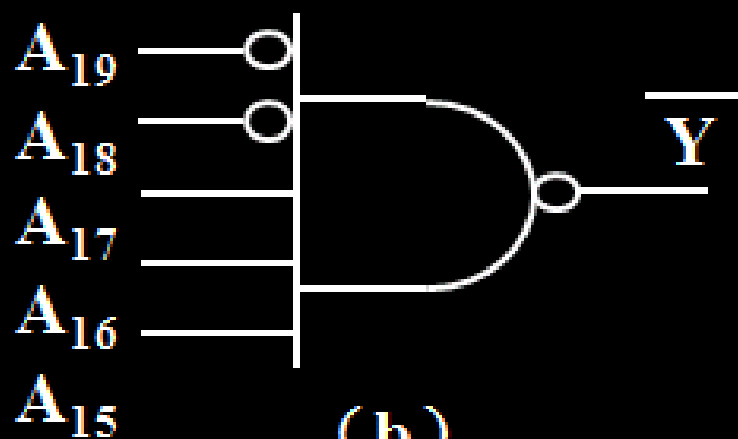
1.译码和译码器

- **地址译码：**将某个特定的“编码输入”翻译为唯一“有效输出”的过程
- 译码电路可以使用**门电路组合逻辑**
- 译码电路更多的是采用**集成译码器**
 - 常用的2:4译码器：74LS139
 - 常用的3:8译码器：74LS138
 - 常用的4:16译码器：74LS154

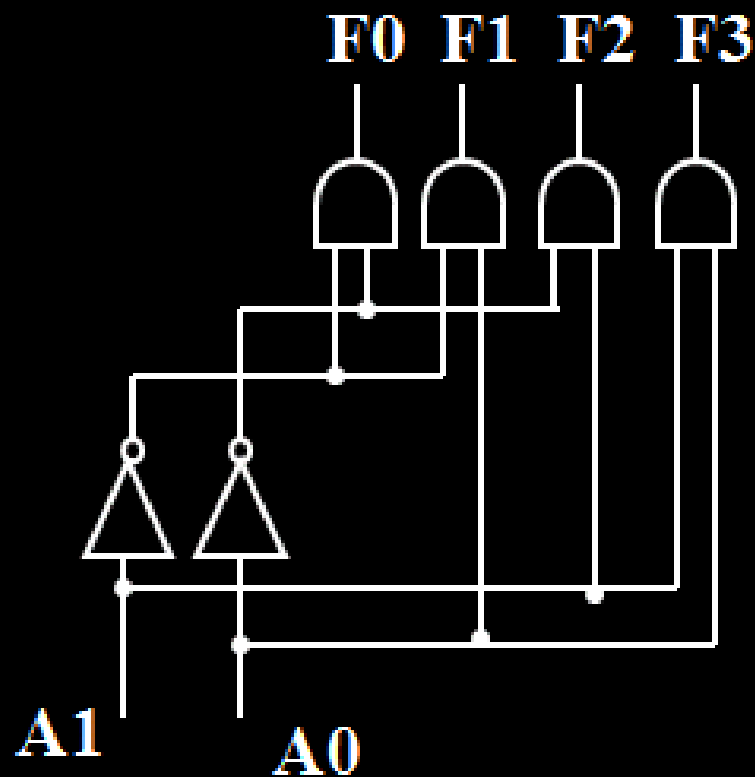
门电路译码示例



(a)

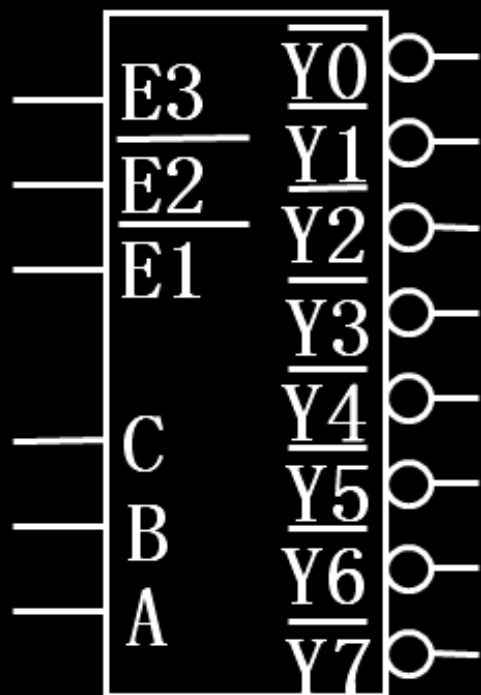


(b)

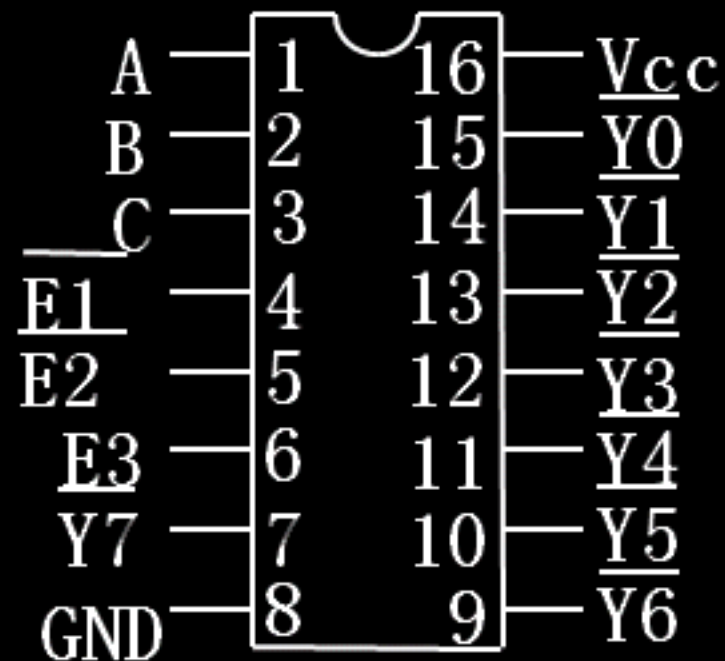


译码器74LS138

74LS138原理图



74LS138引脚图



译码器74LS138真值表

片选输入	编码输入	输出
E3 E2* E1*	C B A	Y7* ~ Y0*
1 0 0	0 0 0	11111110 (仅 Y0* 有效)
	0 0 1	11111101 (仅 Y1* 有效)
	0 1 0	11111011 (仅 Y2* 有效)
	0 1 1	11110111 (仅 Y3* 有效)
	1 0 0	11101111 (仅 Y4* 有效)
	1 0 1	11011111 (仅 Y5* 有效)
	1 1 0	10111111 (仅 Y6* 有效)
	1 1 1	01111111 (仅 Y7* 有效)
非上述情况	×××	11111111 (全无效)

3~8译码器应用实例

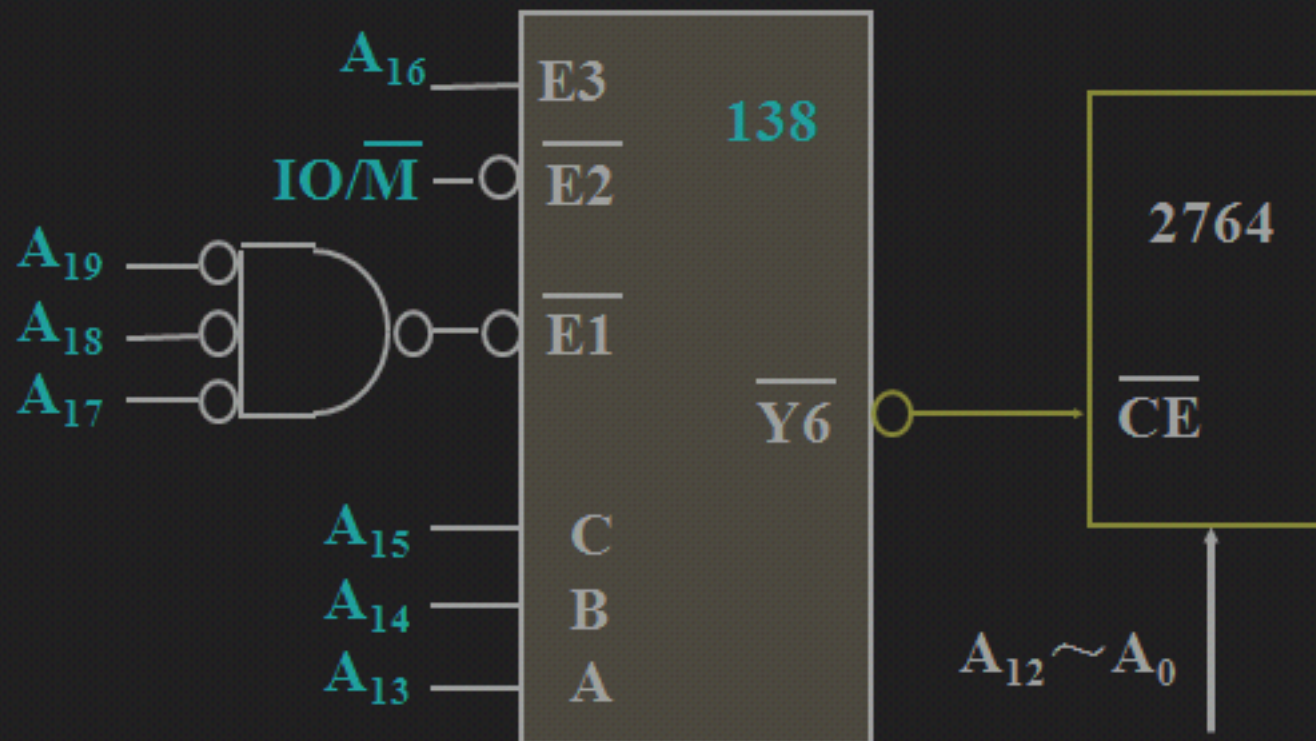
例如,以 8088 为 CPU 的微型计算机系统的 RAM 系统,由 8 片容量为 8KB 的 6264 芯片构成。此 RAM 系统的地址区域为 40000H~4FFFFH。利用 74LS138 作地址译码器,采用全译码方式。则地址译码器的连接线如图



2.全译码

- 所有的系统地址线均参与对存储单元的译码寻址
- 包括低位地址线对芯片内各存储单元的译码寻址（片内译码），高位地址线对存储芯片的译码寻址（片选译码）
- 采用全译码，**每个存储单元的地址都是唯一的**，不存在地址重复
- 译码电路可能比较复杂、连线也较多

全译码示例



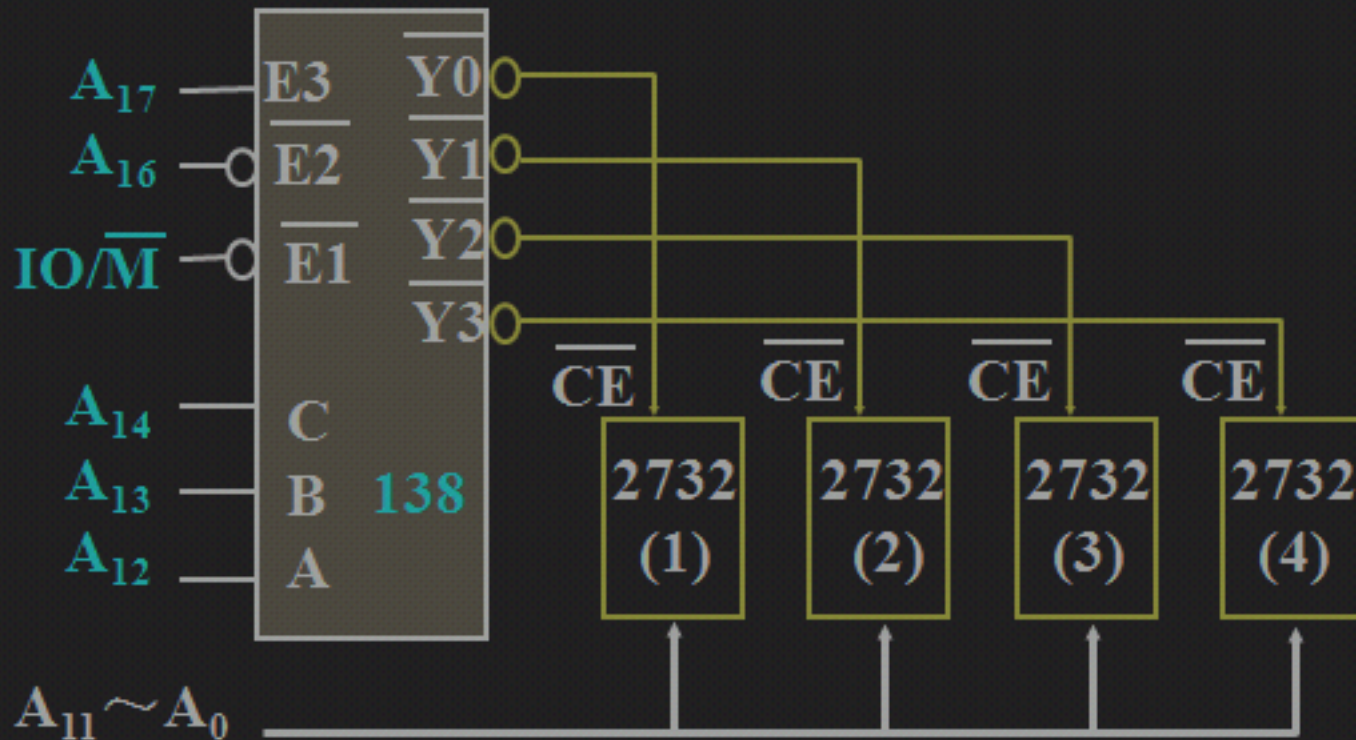
全译码示例

$A_{19}A_{18}A_{17}A_{16}A_{15}A_{14}A_{13}$	$A_{12} \sim A_0$	地址范围
<div>0 0 0 1 1 1 0</div> <div>⋮</div> <div>0 0 0 1 1 1 0</div>	<div>全0</div> <div>⋮</div> <div>全1</div>	<div>1C000H</div> <div>⋮</div> <div>1DFFFH</div>

3.部分译码

- 只有部分（高位）地址线参与对存储芯片的译码
- 每个存储单元将对应多个地址（地址重复），需要选取一个可用地址
- 可简化译码电路的设计
- 但系统的部分地址空间将被浪费

部分译码示例



部分译码示例

	$A_{19} \sim A_{15}$	$A_{14} \sim A_{12}$	$A_{11} \sim A_0$	一个可用地址
1	$\times \times \mathbf{10} \times$	000	全0~全1	20000H~20FFFH
2	$\times \times \mathbf{10} \times$	001	全0~全1	21000H~21FFFH
3	$\times \times \mathbf{10} \times$	010	全0~全1	22000H~22FFFH
4	$\times \times \mathbf{10} \times$	011	全0~全1	23000H~23FFFH

部分译码示例

例如,某一 16 位微机系统中,存储器的映象图如图

		A ₁₉	A ₁₈	A ₁₇	A ₁₆	A ₁₅	A ₁₄	A ₁₃	A ₁₂ ~A ₀
EPROM2	0E000H~0FFFFH	⊗	0	0	0	1	1	1	X~X
EPROM1	0C000H~0DFFFH	⊗	0	0	0	1	1	0	X~X
⋮	⋮								X~X
RAM2	02000H~03FFFH	⊗	0	0	0	0	0	1	X~X
RAM1	00000H~01FFFH	⊗	0	0	0	0	0	0	X~X

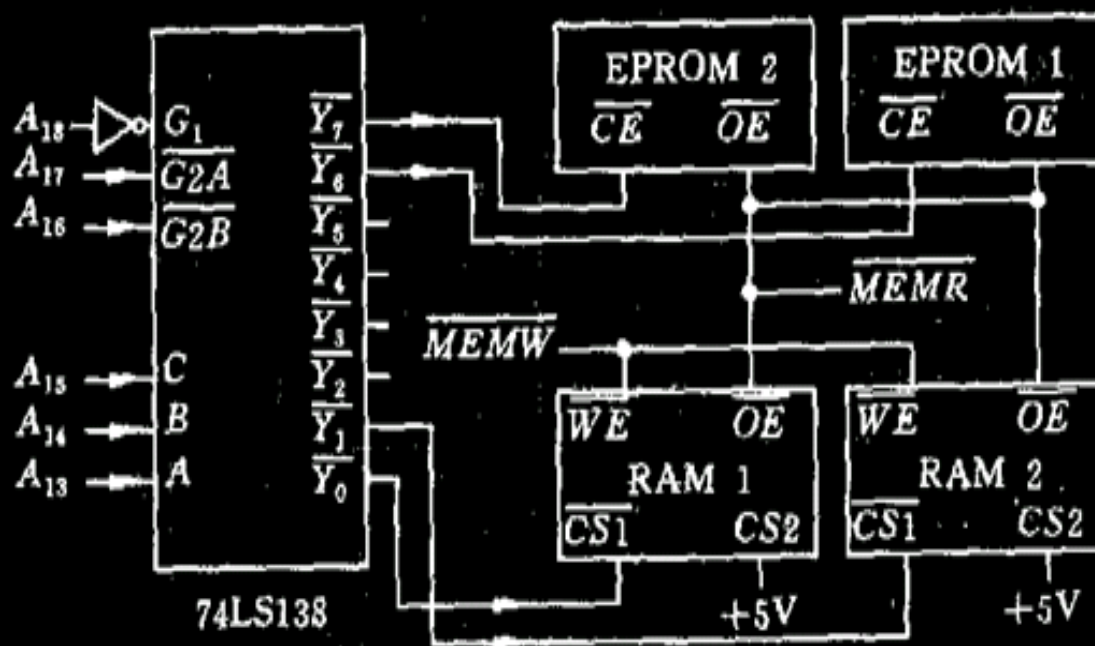
高位地址

低位地址

存储器系统配置 16KB RAM 和 16KB ROM, RAM 由 2 片 8KB 容量的 6264 芯片组成, RAM1 占地址区域为 00000H~01FFFH, RAM2 占地址区域为 02000H~03FFFH; ROM 由 2 片 8KB 容量的 2764 芯片组成, EPROM1 占地址区域为 0C000H~0DFFFH, EPROM2 占地址区域为 0E000H~0FFFFH。

部分译码示例

设计地址译码器时,高位地址 A_{18} 未参加译码,只有 $A_{18} \sim A_{13}$ 6 根高地址线参加译码。
存储器地址选择连接图如图



局部译码同线选方式一样,存在存储器单元地址有重叠现象。

部分译码示例

最后，图5-28示例了一个综合性存储芯片的连接电路。存储器芯片由SRAM和EPROM组成，与最大组态的8088连接，采用部分译码。请读者分析每个芯片的地址范围。

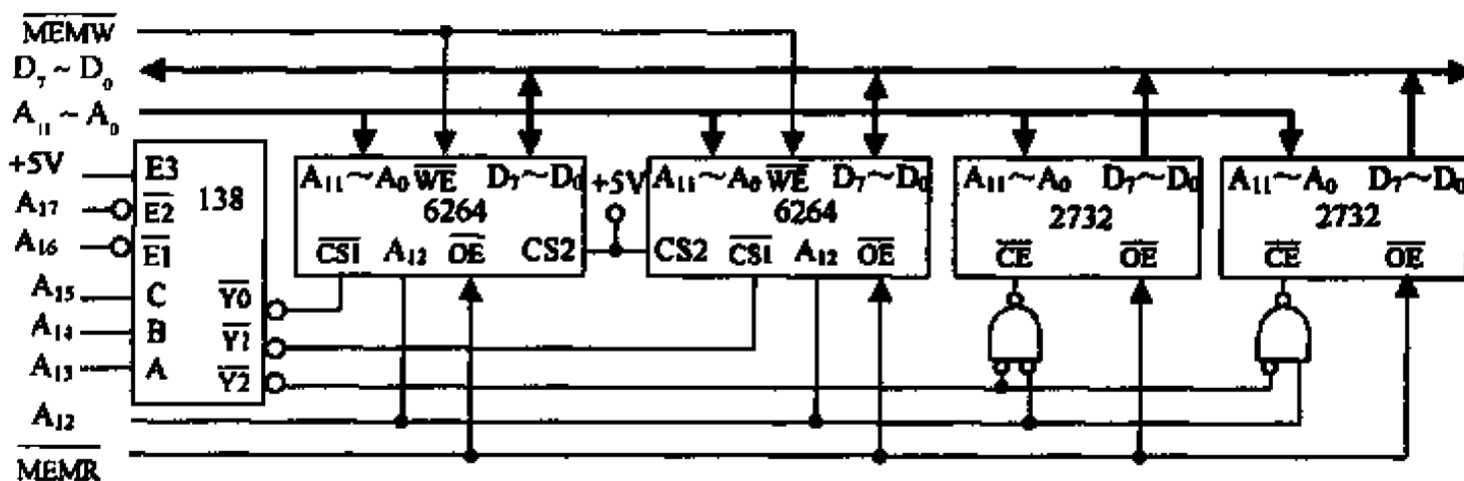
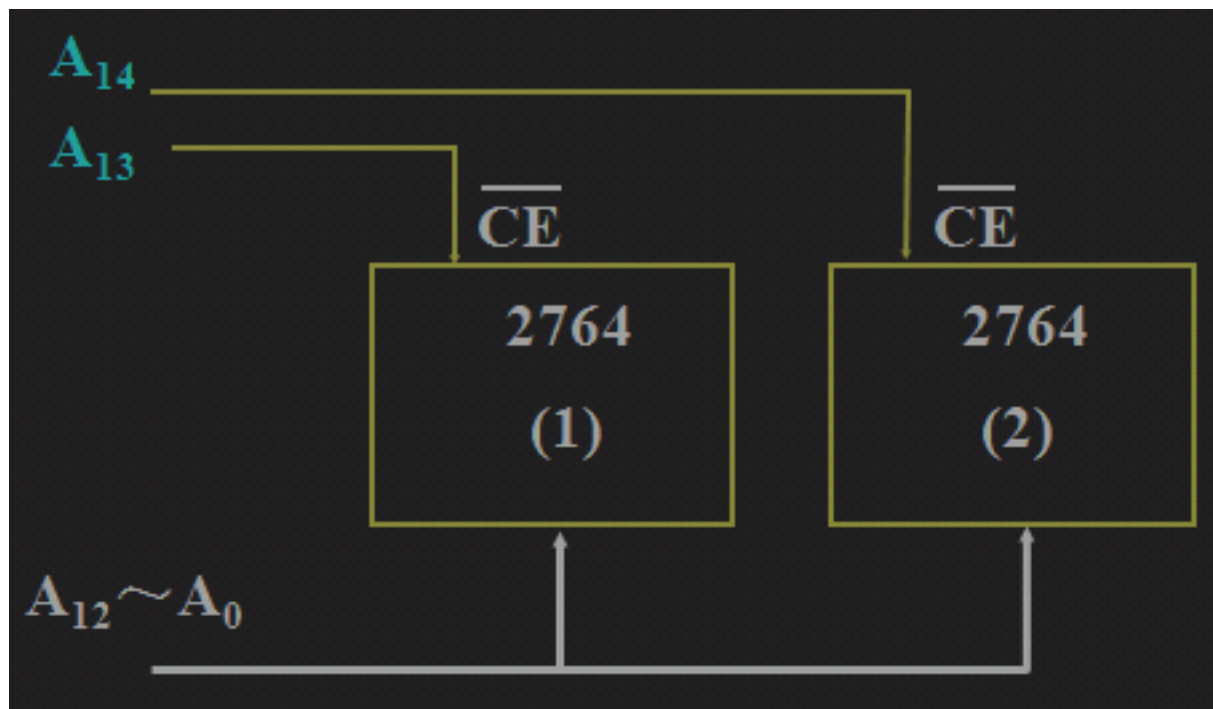


图5-28 综合性示例（8088最大组态）

4.线选译码

- 只用少数几根高位地址线进行芯片的译码，且每根负责选中一个芯片（组）
- 虽构成简单，但地址空间严重浪费，必然会出现地址重复
- 一个存储地址会对应多个存储单元
- 多个存储单元共用的存储地址不应使用

线选译码示例



线选译码示例

	$A_{19} \sim A_{15}$	$A_{14} A_{13}$	$A_{12} \sim A_0$	一个可用地址
1	× × × × ×	1 0	全0 ~ 全1	04000H ~ 05FFFH
2	× × × × ×	0 1	全0 ~ 全1	02000H ~ 03FFFH

$A_{14} A_{13} = 00$ 的情况不能出现

00000H ~ 01FFFH 的地址不可使用

4) 存储芯片的读写控制

- 芯片**OE***与系统的读命令线相连
 - 当芯片被选中、且读命令有效时，存储芯片将开放并驱动数据到总线
- 芯片**WE***与系统的写命令线相连
 - 当芯片被选中、且写命令有效时，允许总线数据写入存储芯片

(7) 存储芯片与CPU的配合

- 存储芯片与CPU总线的连接，还有两个很重要的问题：
- **CPU的总线负载能力**
 - CPU能否带动总线上包括存储器在内的连接器件
- **存储芯片与CPU总线时序的配合**
 - CPU能否与存储器的存取速度相配合

1) 总线驱动

- CPU的总线驱动能力有限
- 单向传送的地址和控制总线，可采用三态锁存器和三态单向驱动器等来加以锁存和驱动
- 双向传送的数据总线，可以采用三态双向驱动器来加以驱动

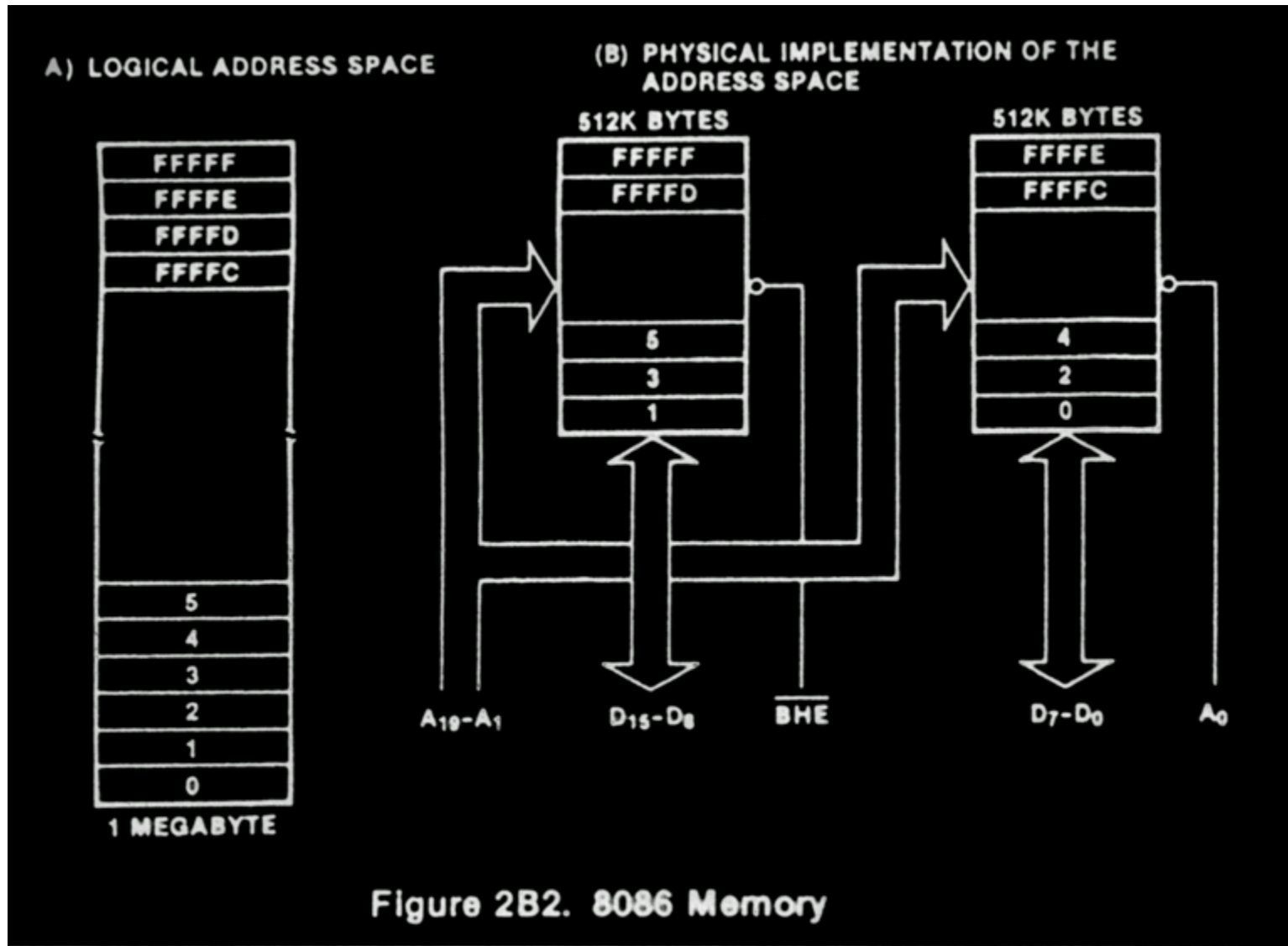
2) 时序配合

- **时序配合是连接中的难点**
- 分析存储器的存取速度是否满足CPU总线时序的要求
- 如果不能满足：
 - 考虑更换芯片
 - 总线周期中插入等待状态 T_w

(8) 8086存储器组织

- **1) 寻址空间(20位地址线)**
 $2^{20} = 1\text{M bytes}$ 的存储器寻址空间
分段的概念
- **2) 分段组织**
- 段寄存器的16位值左移4位，得到的20位值加上16位的偏移量。
- **3) 字与字节访问**
- AD_0 信号和BHE信号组合，选择奇偶字节或字。

8086 存储器组织



奇偶寻址 字节访问

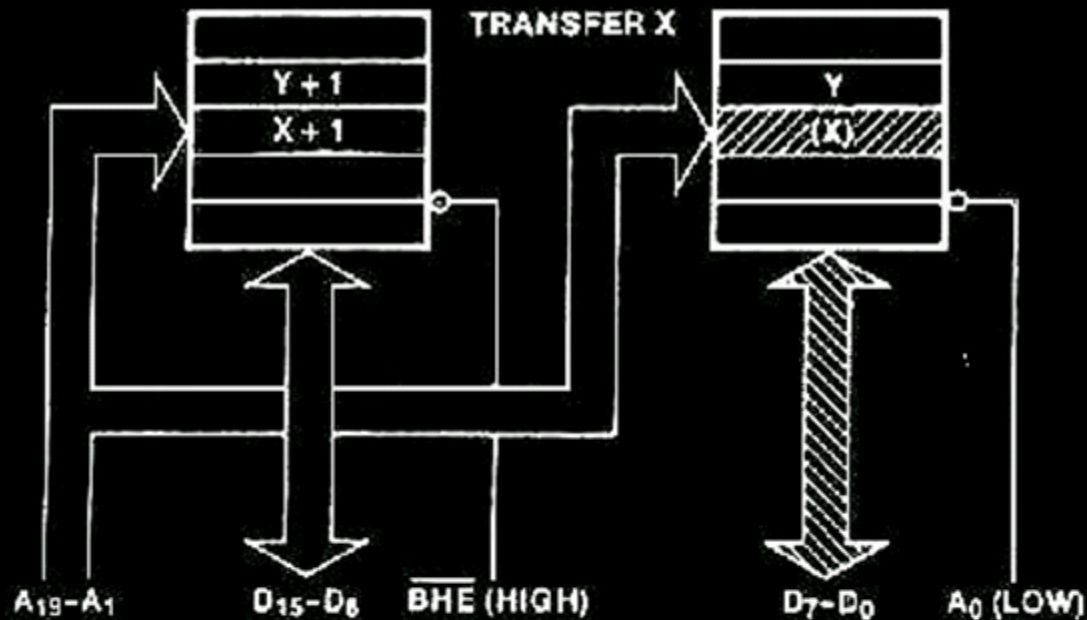


Figure 2B3a. Even Addressed Byte Transfer

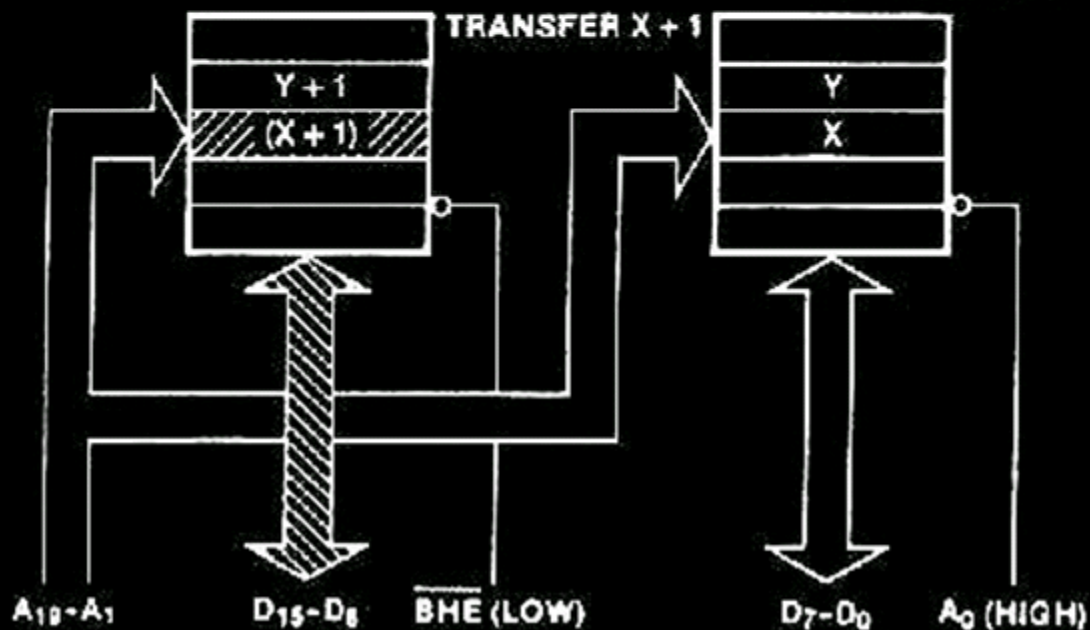
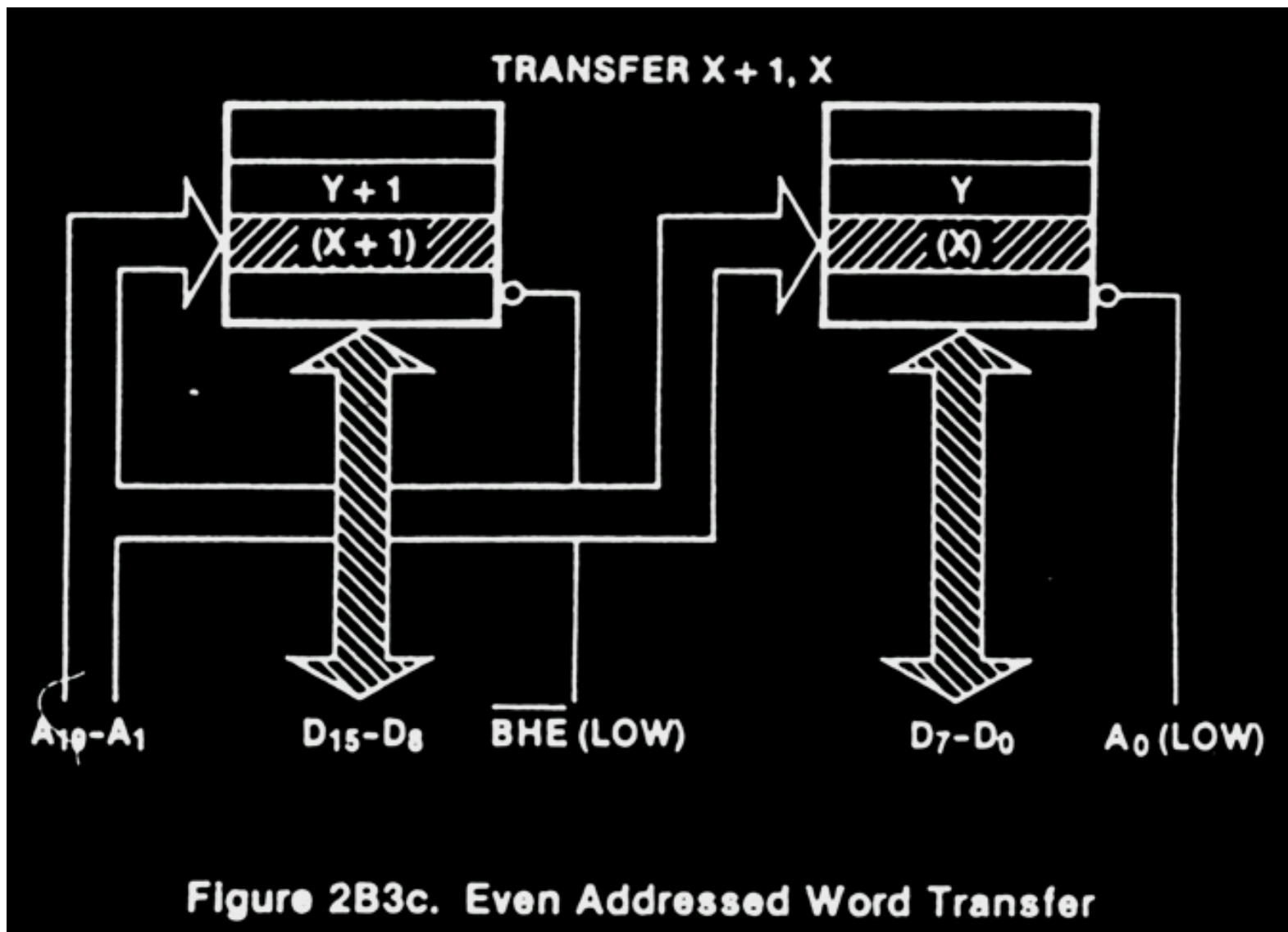
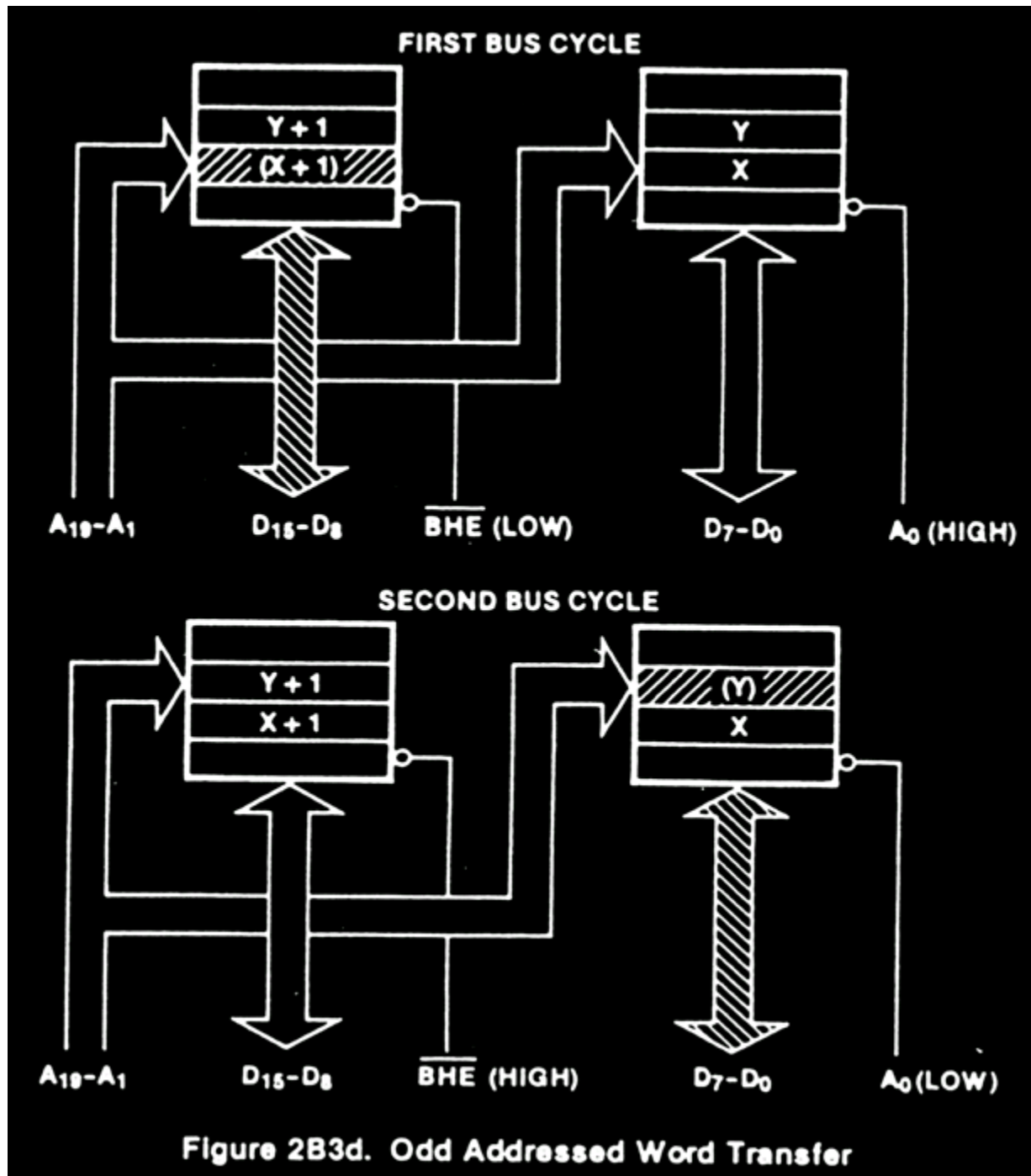


Figure 2B3b. Odd Addressed Byte Transfer

偶地址字传送



奇地址字
传送



(8) 8086 存储器组织

BHE *	AD0	对应的操作	所用的管脚
0	0	从偶地址开始读/写一个字	AD ₁₅ ~AD ₀
0	1	从奇地址单元读/写一个字节	AD ₁₅ ~AD ₈
1	0	从偶地址单元读/写一个字节	AD ₇ ~AD ₀
0 1	1 0	从奇地址开始读/写一个字（在第一个总线周期，传送低8位数据到AD ₁₅ ~AD ₈ ；第二个总线周期传送高8位数据到AD ₇ ~AD ₀ ）	AD ₁₅ ~AD ₈ AD ₇ ~AD ₀

第5章教学要求

- 1. 了解各类半导体存储器的应用特点;
- 2. 熟悉半导体存储器芯片的结构;
- 3. 掌握SRAM 2114、DRAM 4116、EPROM 2764、EEPROM 2817A的引脚功能;
- 4. 理解SRAM读写原理、DRAM读写和刷新原理、EPROM和EEPROM工作方式

第5章教学要求

- 5. 掌握存储芯片与CPU连接的方法，特别是片选端的处理；
- 6. 了解存储芯片与CPU连接的总线驱动和时序配合问题。