

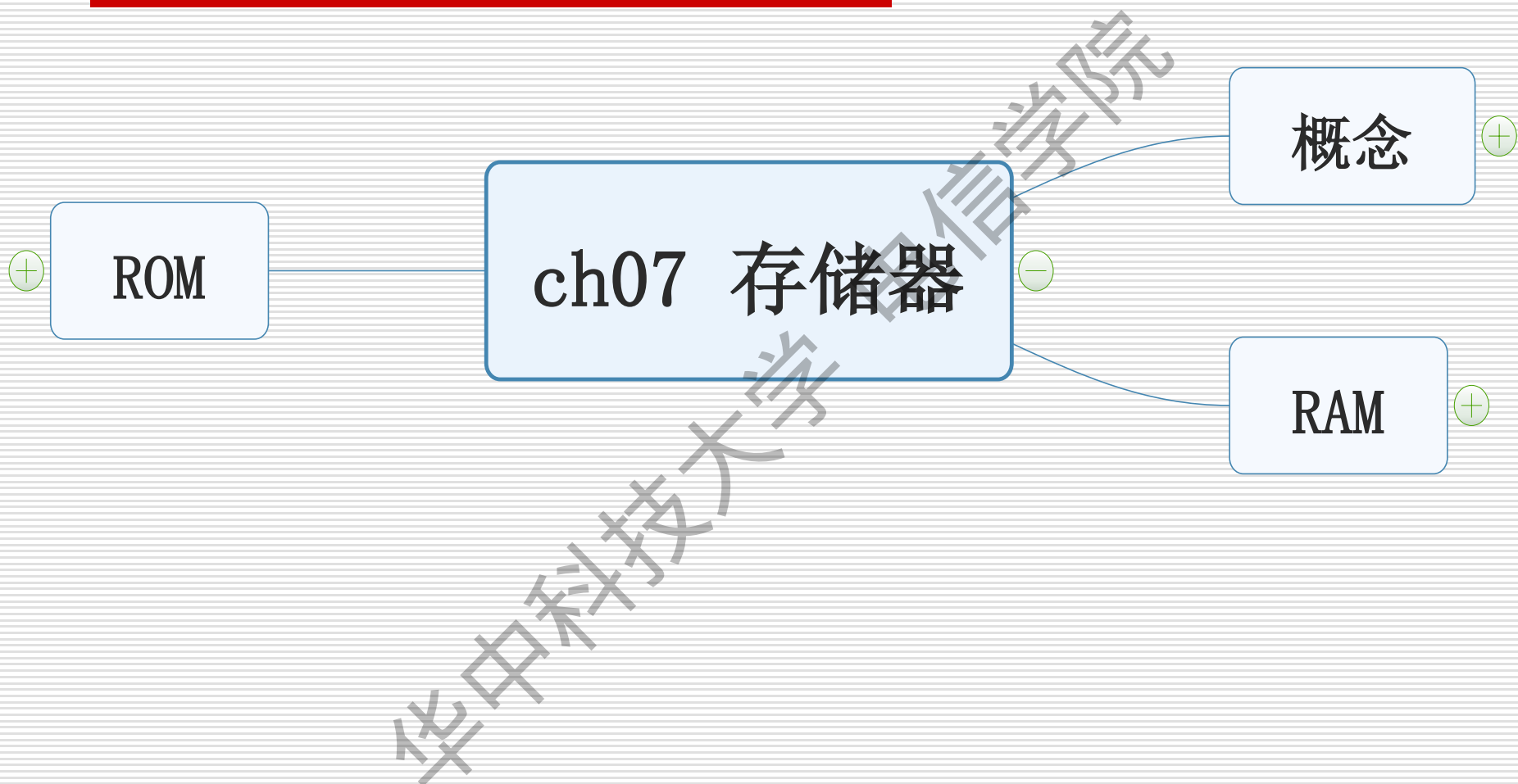
# 7. 半导体存储器

---

## 7.1 只读存储器

## 7.2 随机存取存储器

# 7. 半导体存储器



## 教学基本要求:

- 掌握半导体存储器字、位、存储容量、地址、等基本概念。
- 掌握RAM、ROM的工作原理及典型应用。
- 了解存储器的存储单元的组成及工作原理。

# 概 述

---

半导体存贮器能存放大量二值信息的半导体器件。

存储器的主要性能指标

存储数据量大——存储容量大

取快速度——存储时间短

# 7.1 只读存储器

---

## 7.1.1 ROM的基本结构

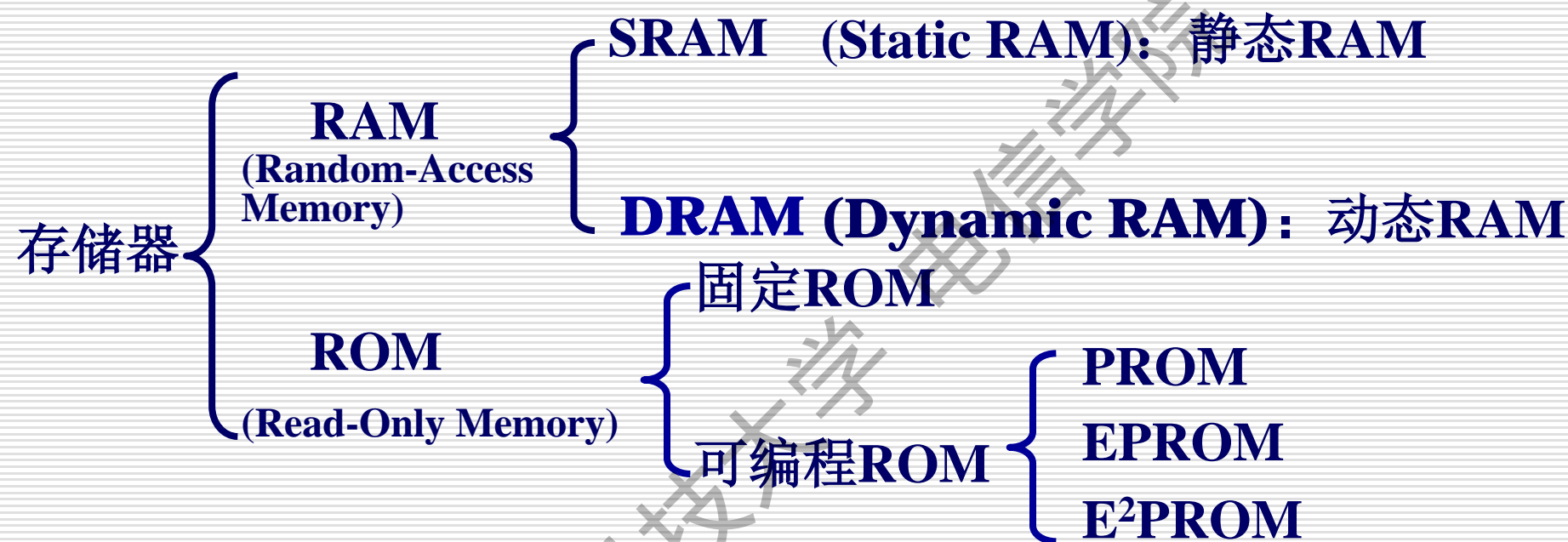
## 7.1.2 二维译码与存储阵列

## 7.1.3 可编程ROM

## 7.1.4 ROM的读操作实例

## 7.1.5 ROM的应用举例

## 7.1 只读存储器



**RAM(随机存取存储器):** 在运行状态可以随时进行读或写操作。  
存储的数据必须有电源供应才能保存, 一旦掉电, 数据全部丢失。

**ROM(只读存储器):** 在正常工作状态只能读出信息。

断电后信息不会丢失, 常用于存放固定信息 (如程序、常数等)。

## 几个基本概念：

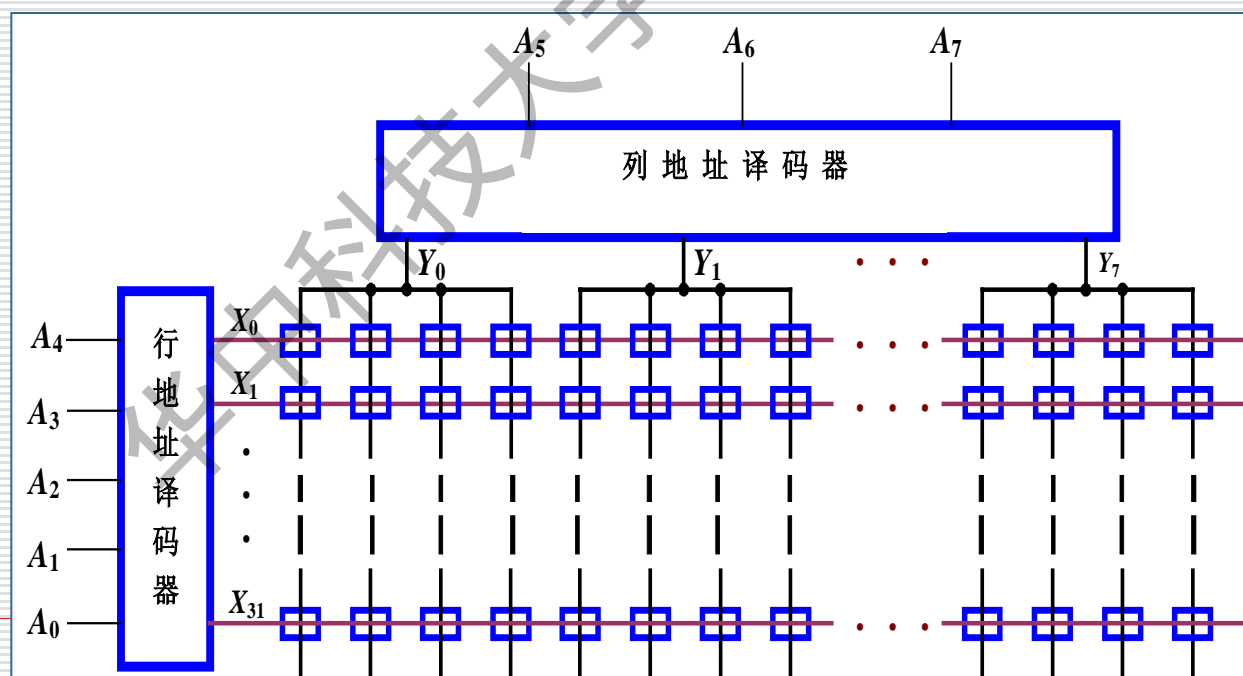
**字：**计算机中作为一个整体被存取传送处理的一组数据。

**字长：**一个字所含的位数称为字长。4位

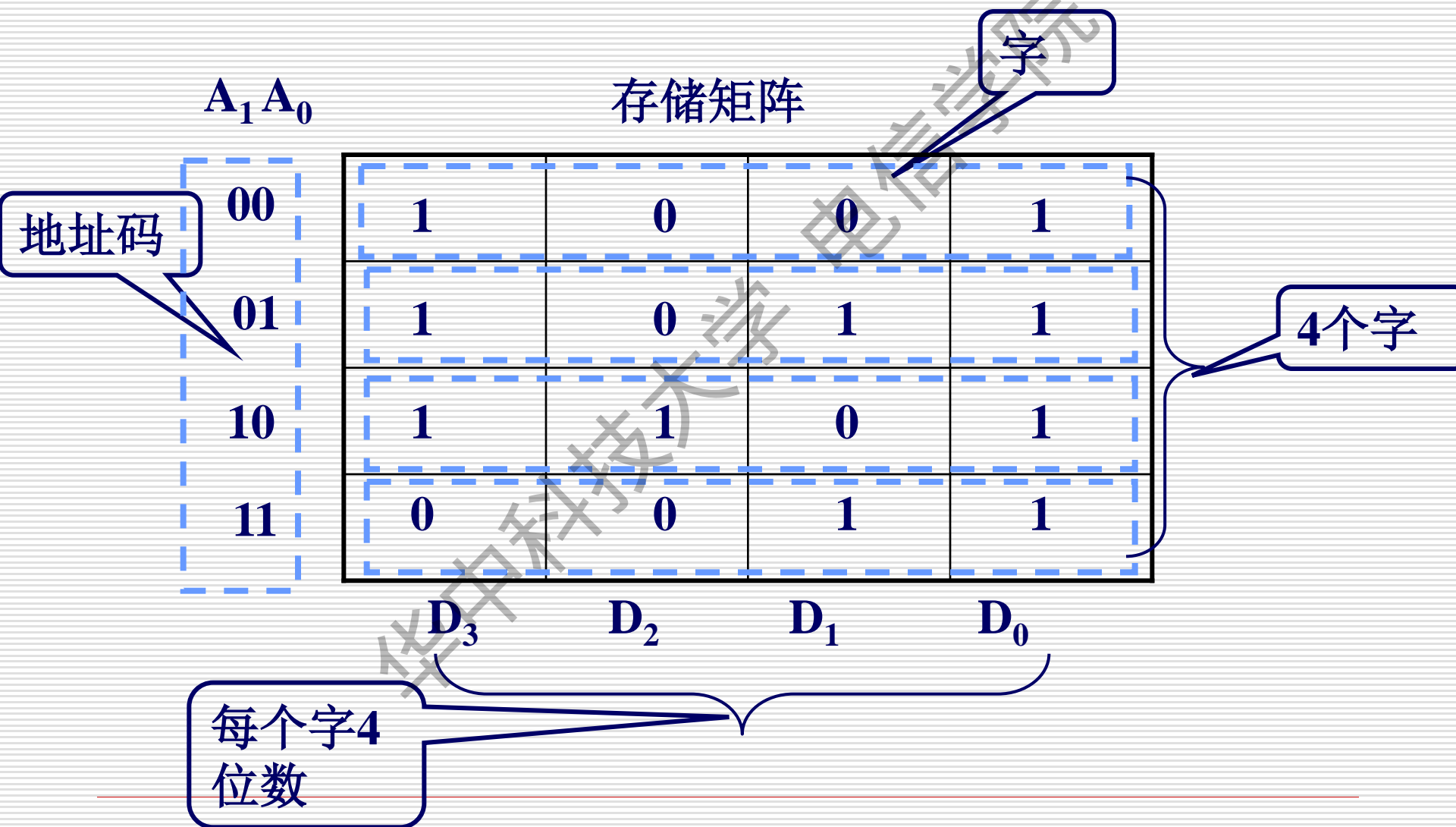
**字数：**字的总量。  $2^5 \times 2^3 = 256$  字数 =  $2^n$  (n为地址线的总数)

**地址：**每个字的编号。

**存储容量 (M)：**存储单元的数目。 存储容量 (M) = 字数 × 位数



## 几个基本概念:





## 7.1.1 ROM的基本结构

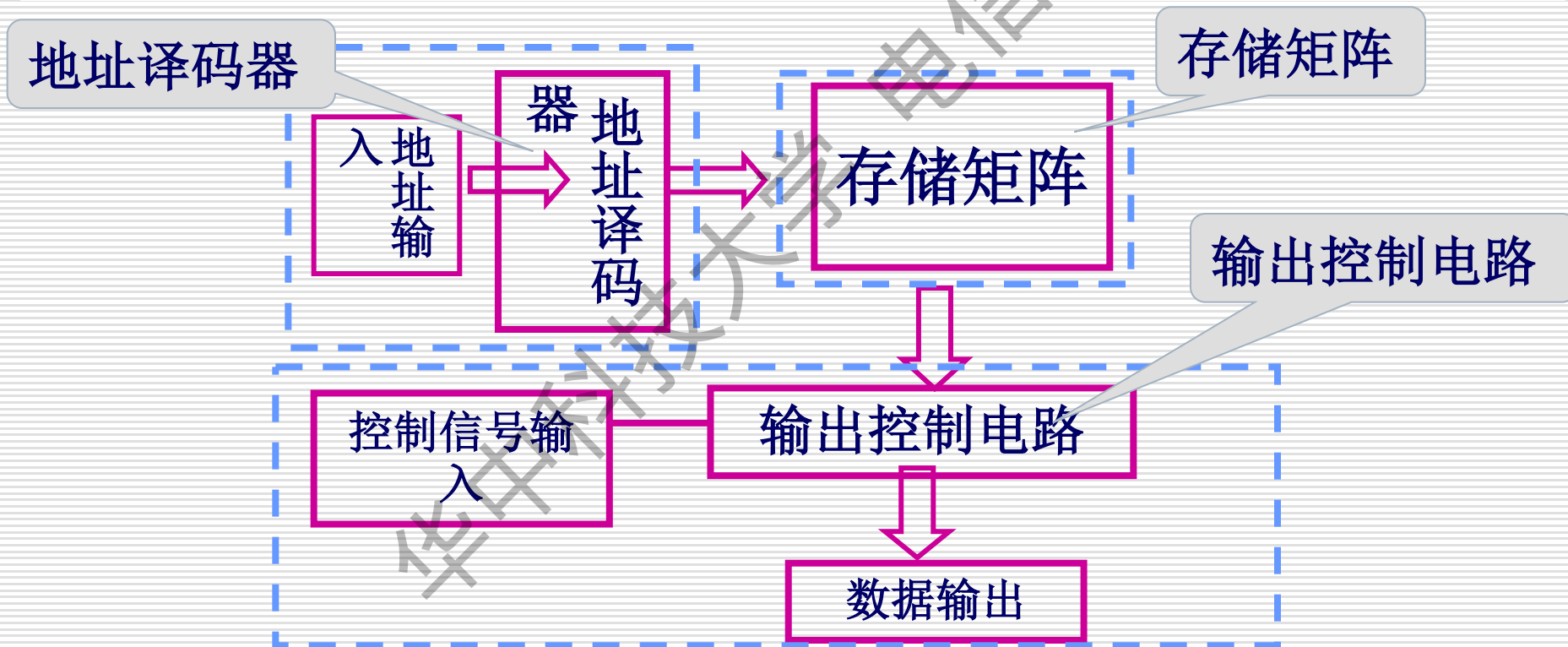
只读存储器，工作时内容只能读出，不能随时写入，所以称为只读存储器。(Read-Only Memory)

### ROM的分类



## 7.1.1 ROM的基本结构

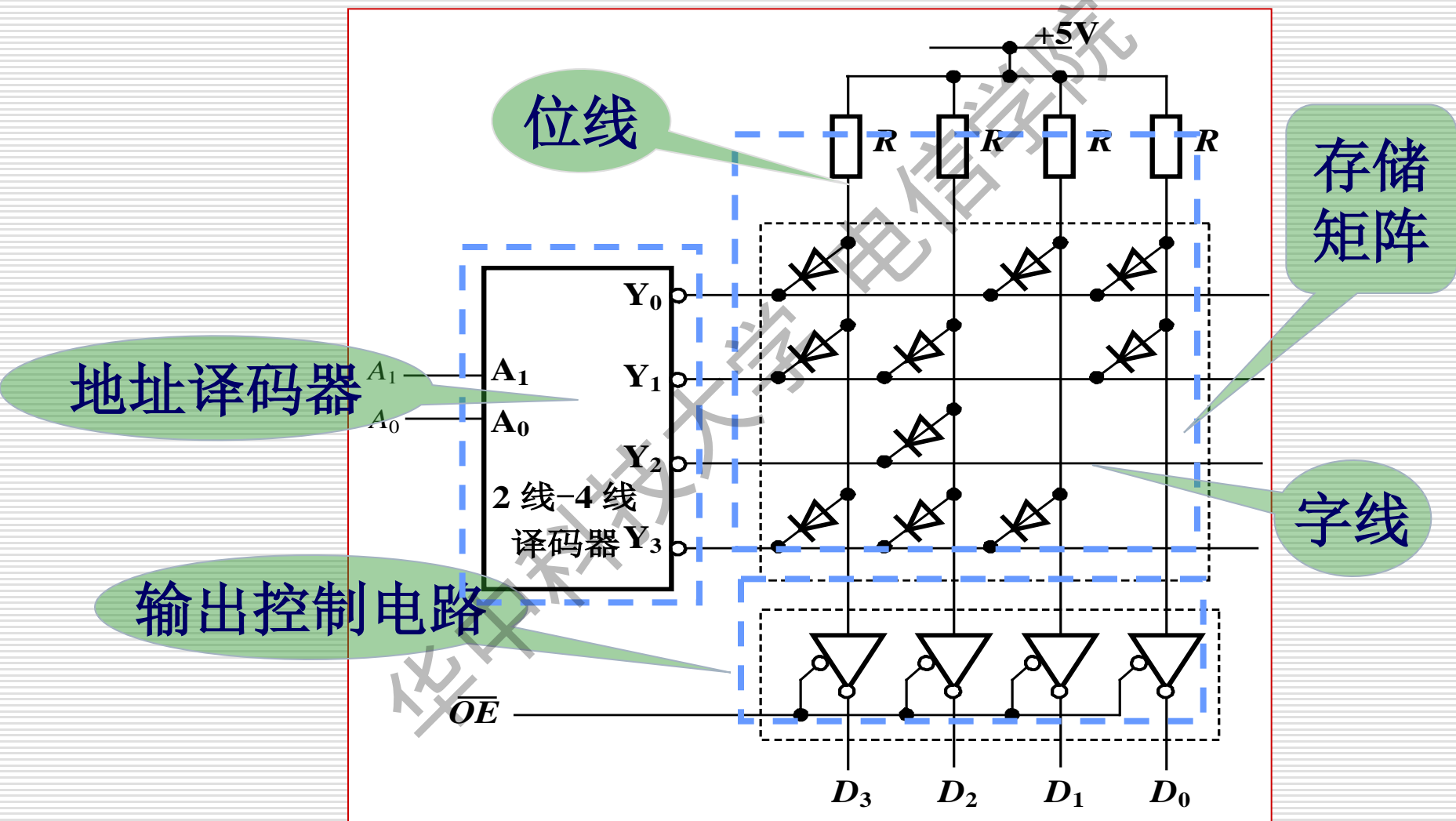
**ROM**是一种永久性数据存储器，其中的数据一般由专用的装置写入，数据一旦写入，不能随意改写，在切断电源之后，数据也不会消失。



**ROM**主要由地址译码器、存储矩阵和输出控制电路三部分组成。

# 1) ROM (二极管PROM) 结构示意图

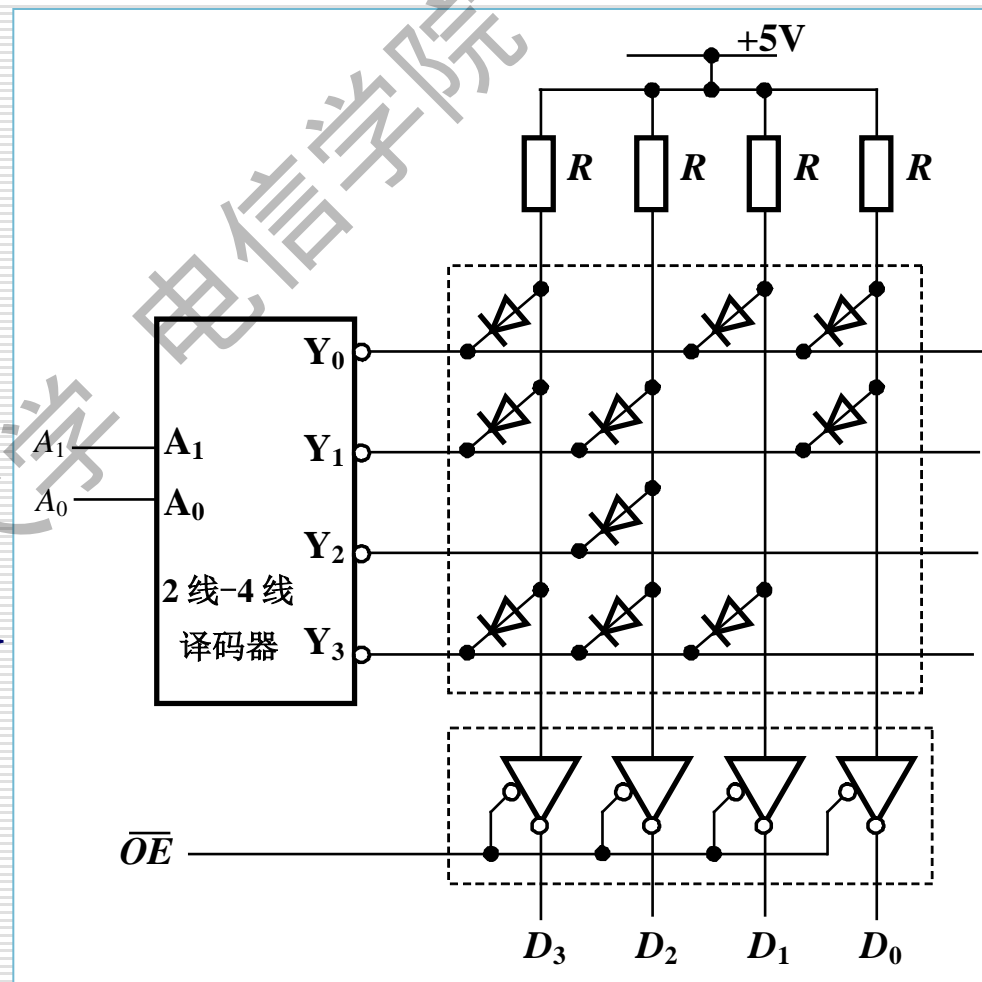
$M=4 \times 4$



当 $\overline{OE}=0$ 时

地 址		内 容			
$A_1$	$A_0$	$D_3$	$D_2$	$D_1$	$D_0$
0	0	1	0	1	1
0	1	1	1	0	1
1	0	0	1	0	0
1	1	1	1	1	0

• 字线与位线的交点都是一个存储单元。交点处有二极管相当存1，无二极管相当存0  
当 $\overline{OE}=1$ 时输出为高阻状态



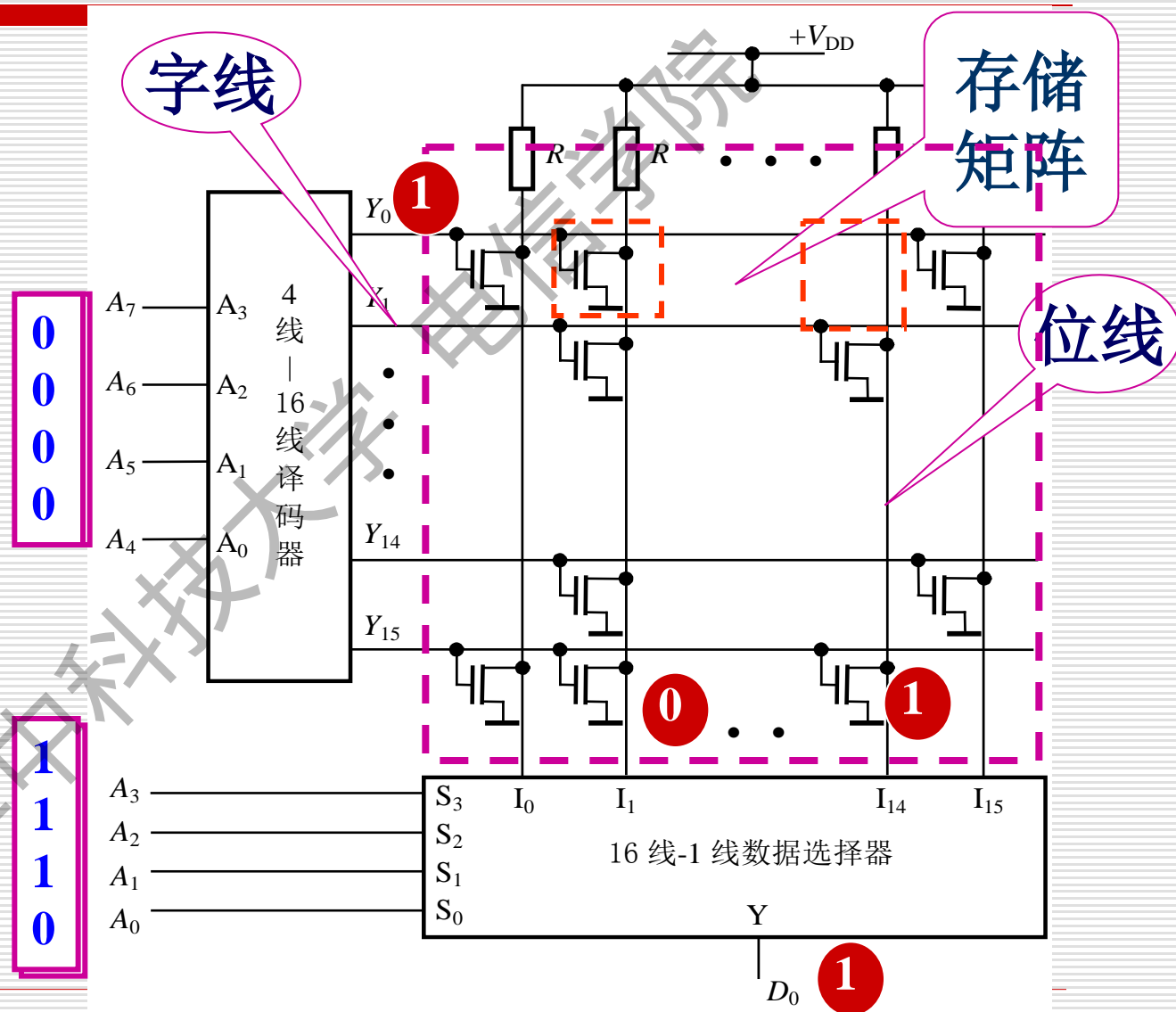
## 7.1.2 二维译码与存储阵列

• 字线与位线的交点都是一个存储单元。

• 交点处有MOS管相当存0，无MOS管相当存1。

该存储器的容量=?

**256 × 1**



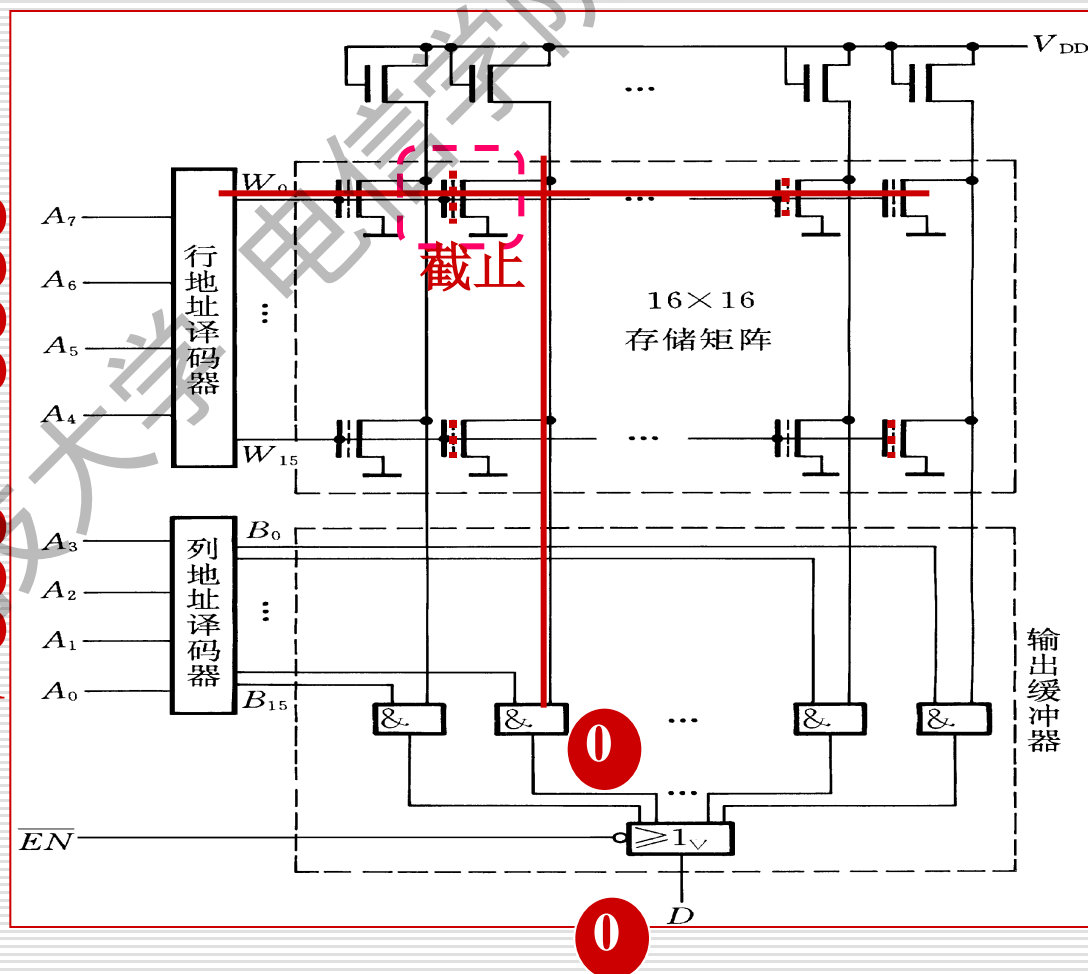
## 7.1.3 可编程ROM (256X1位EPROM)

256个存储单元排成16×16的矩阵

行译码器从16行中选出要读的一行

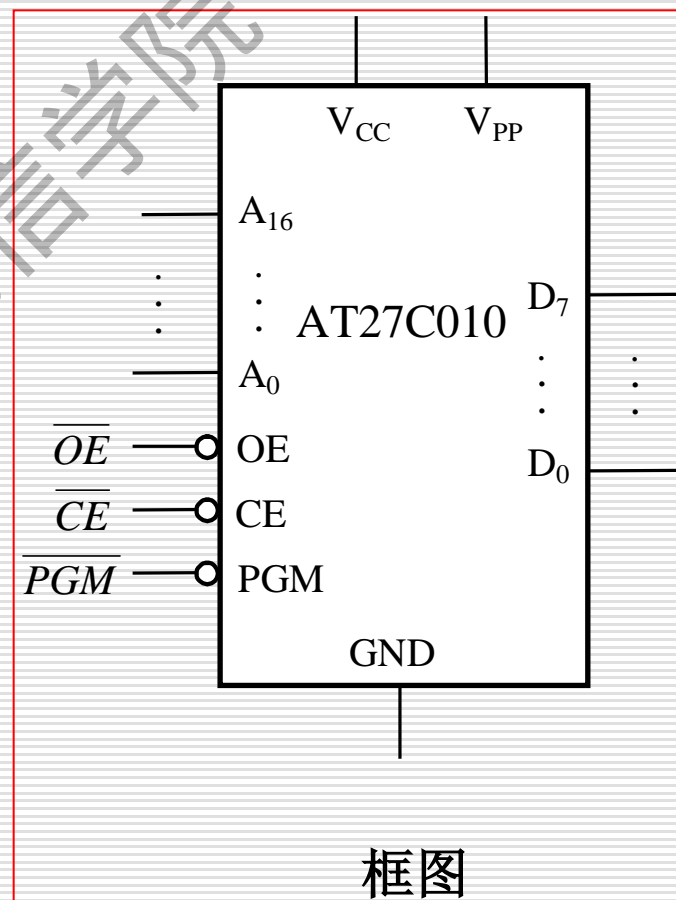
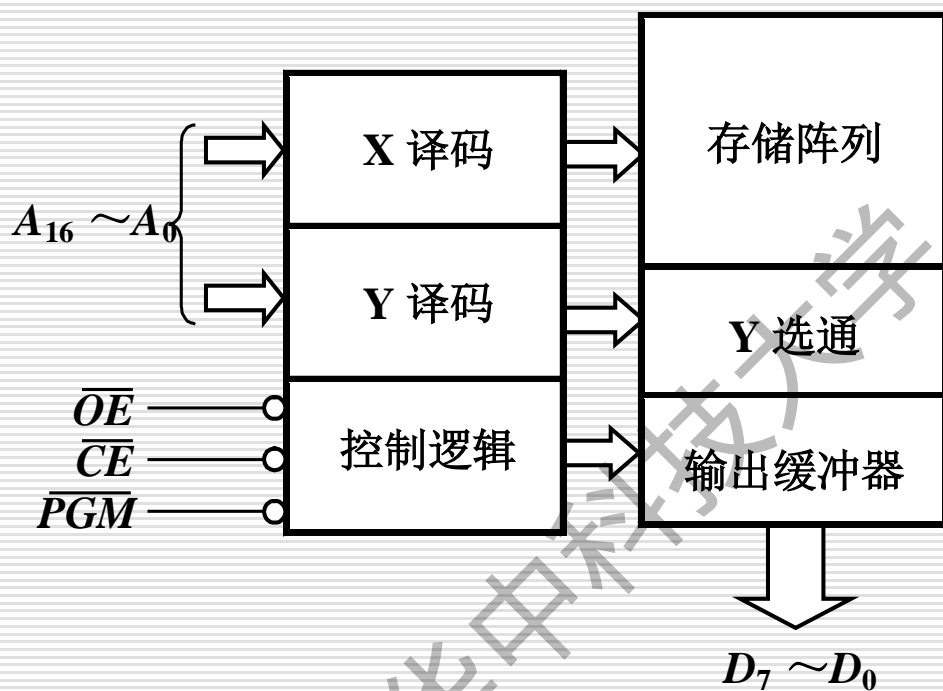
列译码器再从选中的一行存储单元中选出要读的一列的一个存储单元。

如选中的存储单元的MOS管的浮栅注入了电荷，该管截止，读得1；相反读得0



## 7.1.4 ROM读操作实例

### 1. PROM芯片AT27C010简介



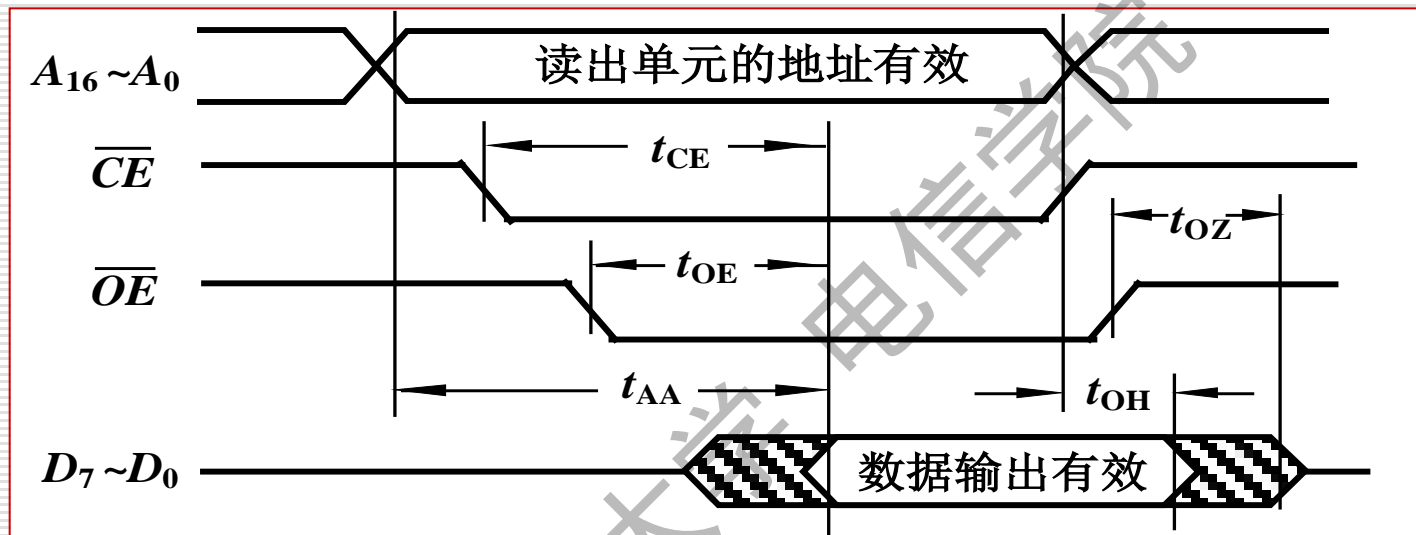
128K×8位ROM

## 工作模式

工作模式	$\overline{CE}$	$\overline{OE}$	$\overline{PGM}$	$A_{16} \sim A_0$	$V_{PP}$	$D_7 \sim D_0$
读	0	0	X	$A_i$	X	数据输出
输出无效	X	1	X	X	X	高阻
等待	1	X	X	$A_i$	X	高阻
快速编程	0	1	0	$A_i$	$V_{PP}$	数据输入
编程校验	0	0	1	$A_i$	$V_{PP}$	数据输出



## 2. 读操作与定时图



- (1) 欲读取单元的地址加到存储器的地址输入端;
- (2) 加入有效的片选信号  $\overline{CE}$
- (3) 使输出使能信号  $\overline{OE}$  有效, 经过一定延时后, 有效数据出现在数据线上;
- (4) 让片选信号  $\overline{CE}$  或输出使能信号  $\overline{OE}$  无效, 经过一定延时后数据线呈高阻态, 本次读出结束。

## 7.1.5 ROM应用举例

- (1) 用于存储固定的专用程序
- (2) 利用ROM可实现查表或码制变换等功能

查表功能 —— 查某个角度的三角函数

把变量值（角度）作为地址码，其对应的函数值作为存放在该地址内的数据，这称为“造表”。使用时，根据输入的地址(角度)，就可在输出端得到所需的函数值，这就称为“查表”。

码制变换 —— 把欲变换的编码作为地址，把最终的目的编码作为相应存储单元中的内容即可。

# 用ROM实现二进制码与格雷码相互转换的电路

$C$	$I_3 I_2 I_1 I_0$ 二进制码	$O_3 O_2 O_1 O_0$ 格雷码	$C$	$I_3 I_2 I_1 I_0$ 格雷码	$O_3 O_2 O_1 O_0$ 二进制码
0	0000	0000	1	0000	0000
0	0001	0001	1	0001	0001
0	0010	0011	1	0010	0011
0	0011	0010	1	0011	0010
0	0100	0110	1	0100	0111
0	0101	0111	1	0101	0110
0	0110	0101	1	0110	0100
0	0111	0100	1	0111	0101
0	1000	1100	1	1000	1111
0	1001	1101	1	1001	1110
0	1010	1111	1	1010	1100
0	1011	1110	1	1011	1101
0	1100	1010	1	1100	1000
0	1101	1011	1	1101	1001
0	1110	1001	1	1110	1011
0	1111	1000	1	1111	1010

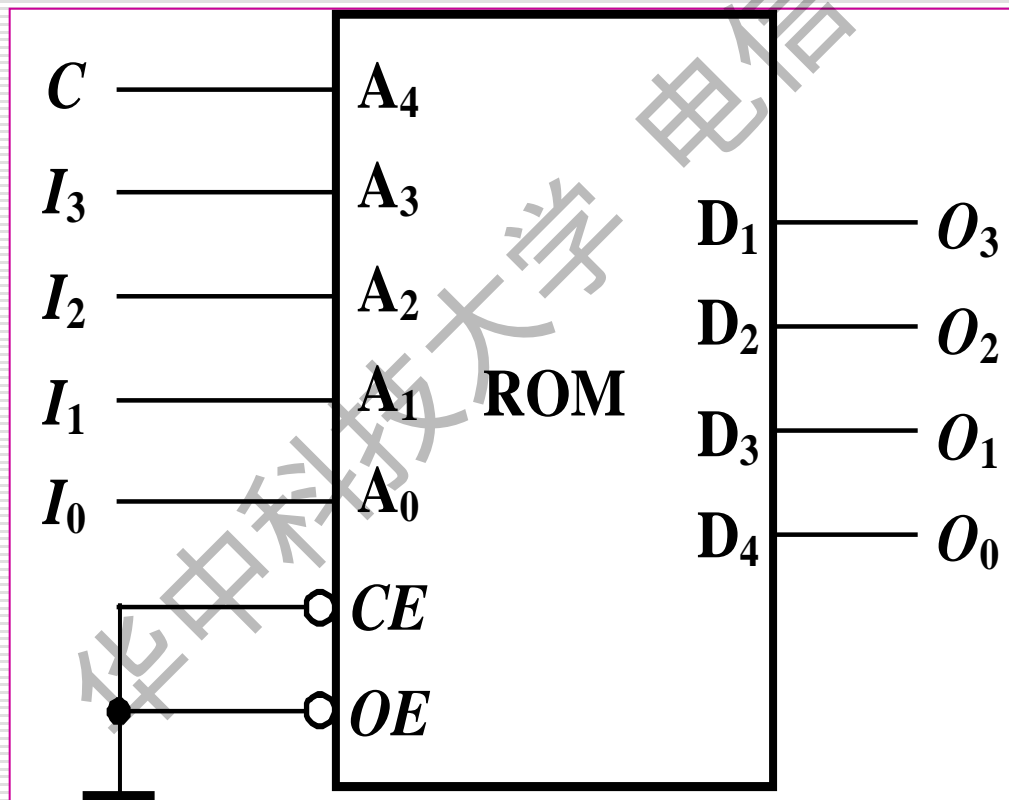
$$C=A_4$$

$$I_3 I_2 I_1 I_0 = A_3 A_2 A_1 A_0$$

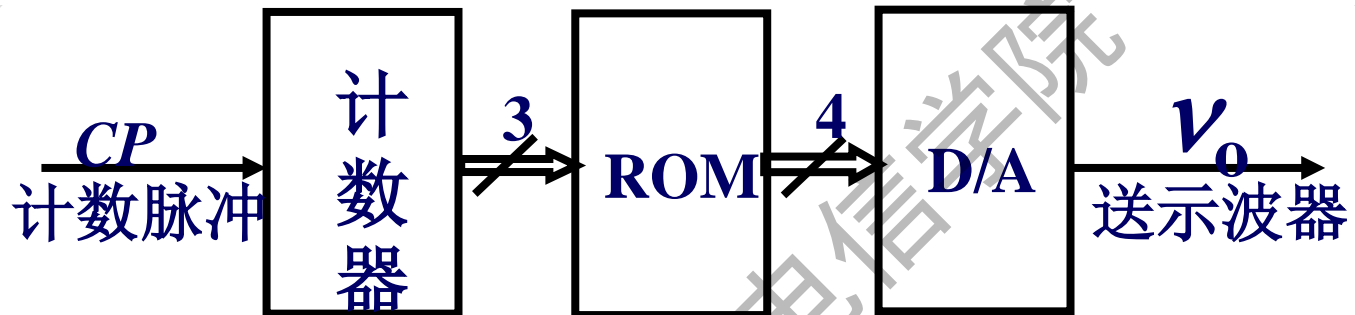
$$O_3 O_2 O_1 O_0 = D_3 D_2 D_1 D_0$$

$C$ ( $A_4$ )	$I_3 I_2 I_1 I_0$ ( $A_3 A_2 A_1 A_0$ ) 二进制码	$O_3 O_2 O_1 O_0$ ( $D_3 D_2 D_1 D_0$ ) 格雷码	$C$ ( $A_4$ )	$I_3 I_2 I_1 I_0$ ( $A_3 A_2 A_1 A_0$ ) 格雷码	$O_3 O_2 O_1 O_0$ ( $D_3 D_2 D_1 D_0$ ) 二进制码
0	0000	0000	1	0000	0000
0	0001	0001	1	0001	0001
0	0010	0011	1	0010	0011
0	0011	0010	1	0011	0010
0	0100	0110	1	0100	0111
0	0101	0111	1	0101	0110
0	0110	0101	1	0110	0100
0	0111	0100	1	0111	0101
0	1000	1100	1	1000	1111
0	1001	1101	1	1001	1110
0	1010	1111	1	1010	1100
0	1011	1110	1	1011	1101
0	1100	1010	1	1100	1000
0	1101	1011	1	1101	1001
0	1110	1001	1	1110	1011
0	1111	1000	1	1111	1010

## 用ROM实现二进制码与格雷码相互转换的电路

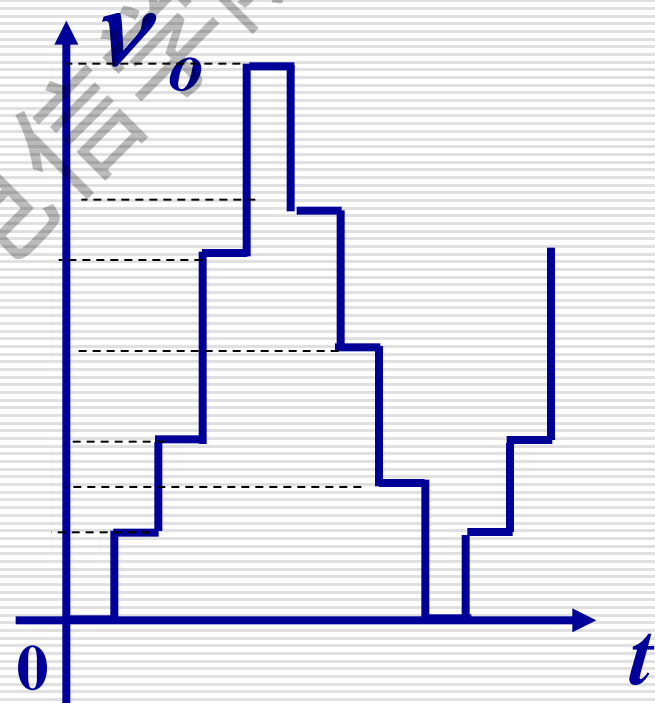


## (2) ROM 在波形发生器中的应用



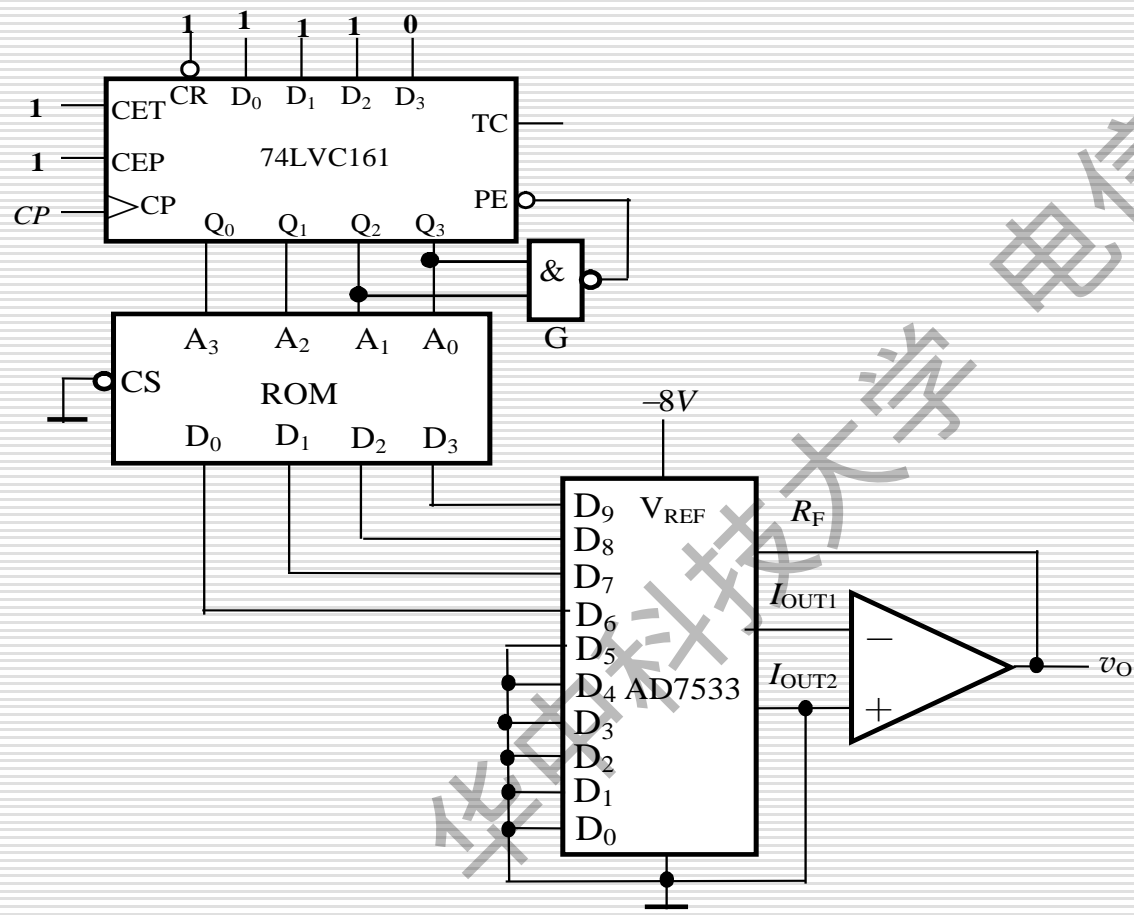
$A_2$	$A_1$	$A_0$	$D_3$	$D_2$	$D_1$	$D_0$	D/A
0	0	0	0	0	0	0	0
0	0	1	0	0	1	0	2
0	1	0	0	1	0	0	4
0	1	1	1	0	0	0	8
1	0	0	1	1	0	0	12
1	0	1	1	0	0	1	9
1	1	0	0	1	1	0	6
1	1	1	0	0	1	1	3

$A_2$	$A_1$	$A_0$	$D_3$	$D_2$	$D_1$	$D_0$	D/A
0	0	0	0	0	0	0	0
0	0	1	0	0	1	0	2
0	1	0	0	1	0	0	4
0	1	1	1	0	0	0	8
1	0	0	1	1	0	0	12
1	0	1	1	0	0	1	9
1	1	0	0	1	1	0	6
1	1	1	0	0	1	1	3



# 例（补充）

ROM的数据表



$A_3$	$A_2$	$A_1$	$A_0$	$D_9$	$D_8$	$D_7$	$D_6$
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	0
0	0	1	1	0	0	1	1
0	1	0	0	0	1	0	0
0	1	0	1	0	1	1	1
0	1	1	0	1	0	0	0
0	1	1	1	0	1	1	1
1	0	0	0	0	0	0	1
1	0	0	1	0	0	1	0
1	0	1	0	0	0	1	1
1	0	1	1	0	1	0	0
1	1	0	0	0	0	1	1
1	1	0	1	0	1	1	0
1	1	1	0	0	1	1	1
1	1	1	1	1	0	0	0

$D_9 D_8 D_7 D_6 = 0 0 0 1$  时  $v_O = 0.5V$

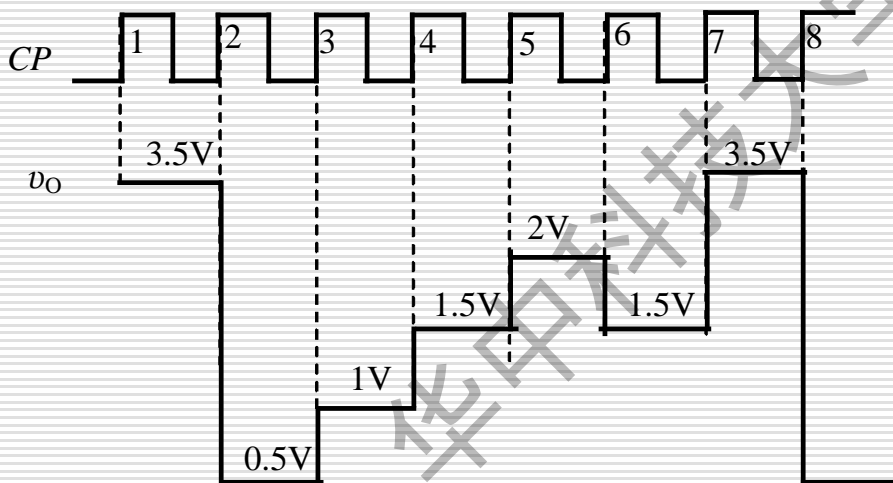


# 计数器状态表

$Q_3$	$Q_2$	$Q_1$	$Q_0$
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0

# ROM的数据表

$A_3$	$A_2$	$A_1$	$A_0$	$D_9$	$D_8$	$D_7$	$D_6$
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	0
0	0	1	1	0	0	1	1
0	1	0	0	0	1	0	0
0	1	0	1	0	1	1	1
0	1	1	0	1	0	0	0
0	1	1	1	0	1	1	1
1	0	0	0	0	0	0	1
1	0	0	1	0	0	1	0
1	0	1	0	0	0	1	1
1	0	1	1	0	1	0	0
1	1	0	0	0	0	1	1
1	1	0	1	0	1	1	0
1	1	1	0	0	1	1	1
1	1	1	1	1	0	0	0



如要求输出波形为矩形波,幅值=1V?

如要求输出波形为锯齿波,幅值=3.5V  $D_9 D_8 D_7 D_6 = 0 0 0 1$  时  $v_O = 0.5V$

## 7.2 随机存取存储器（RAM）

---

### 7.2.1 静态随机存取存储器(SRAM)

### 7.2.2 同步静态随机存取存储器（SSRAM）

### 7.2.3 动态随机存取存储器(DRAM)

### 7.2.4 存储器容量的扩展

### 7.2.5 RAM应用举例

## 7.2 随机存取存储器 (RAM)

### 7.2.1 静态随机存取存储器(SRAM)

#### 1. SRAM 的基本结构及输出

$$\overline{CE} \quad \overline{WE} \quad \overline{OE} = 1XX$$

高阻

$$\overline{CE} \quad \overline{WE} \quad \overline{OE} = 010$$

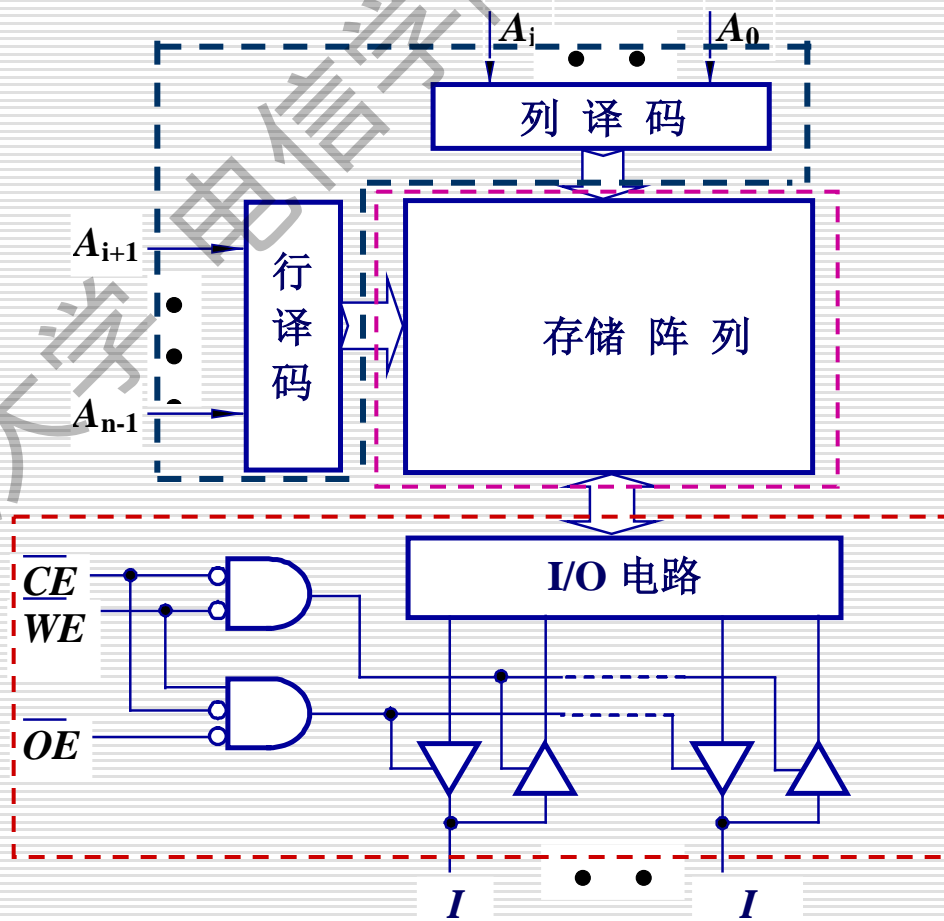
输出

$$\overline{CE} \quad \overline{WE} \quad \overline{OE} = 00X$$

输入

$$\overline{CE} \quad \overline{WE} \quad \overline{OE} = 011$$

高阻



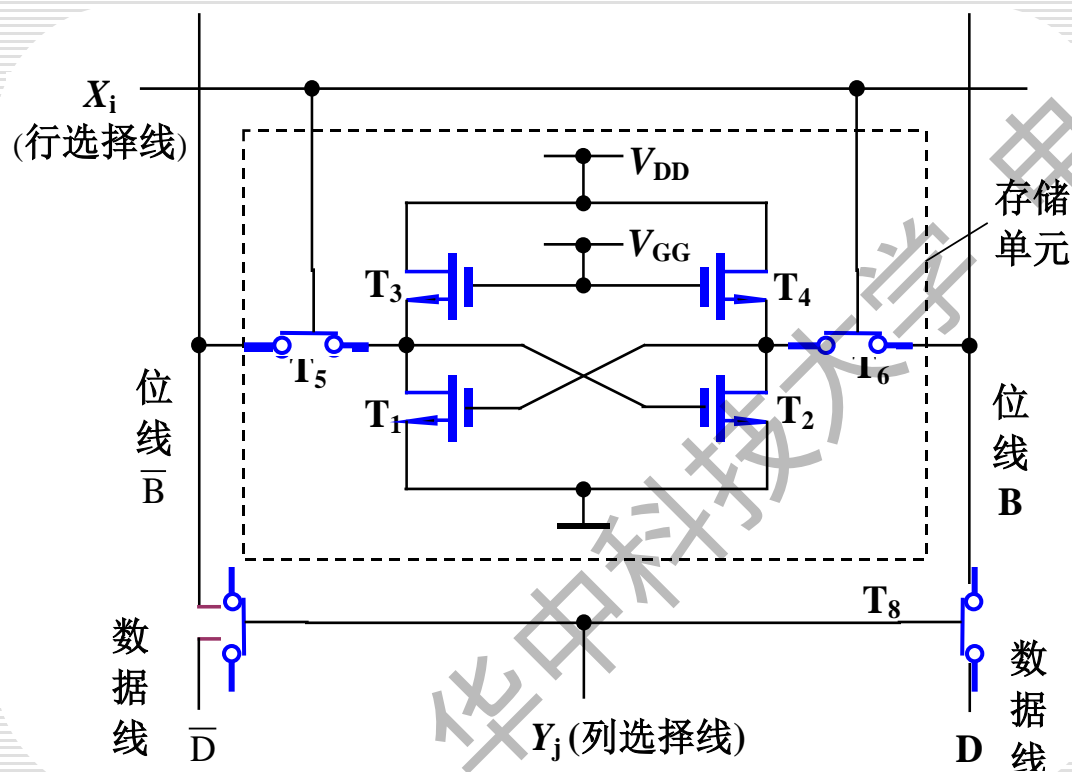
## SRAM 的工作模式

工作模式	$\overline{CE}$	$\overline{WE}$	$\overline{OE}$	$I/O_0 \sim I/O_{m-1}$
保持 (低功耗)	1	X	X	高阻
读	0	1	0	数据输出
写	0	0	X	数据输入
输出无效	0	1	1	高阻



## 2. SRAM存储单元

### • 静态SRAM(Static RAM)



$$X_i = 1$$

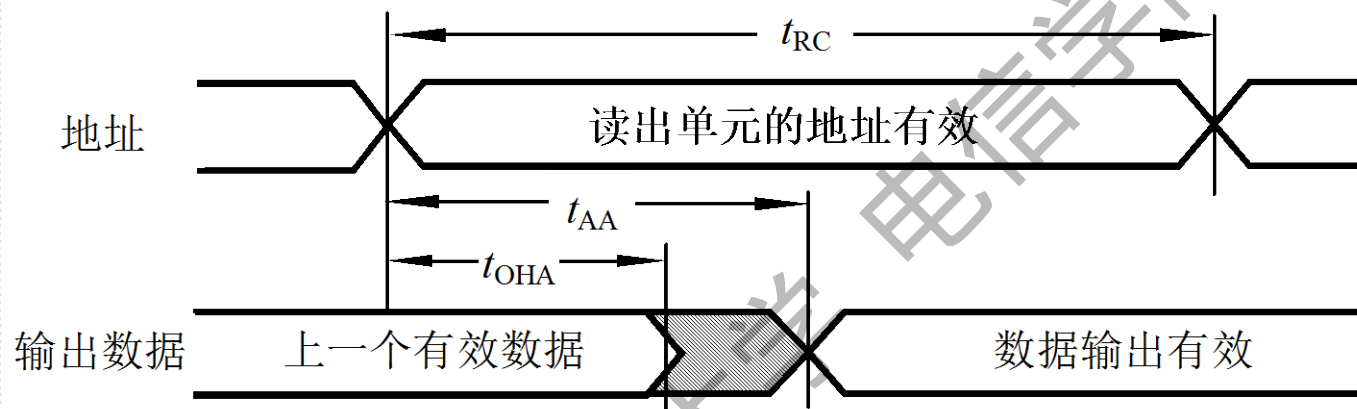
- $T_5$ 、 $T_6$  导通  
触发器与位线接通

$$Y_j = 1$$

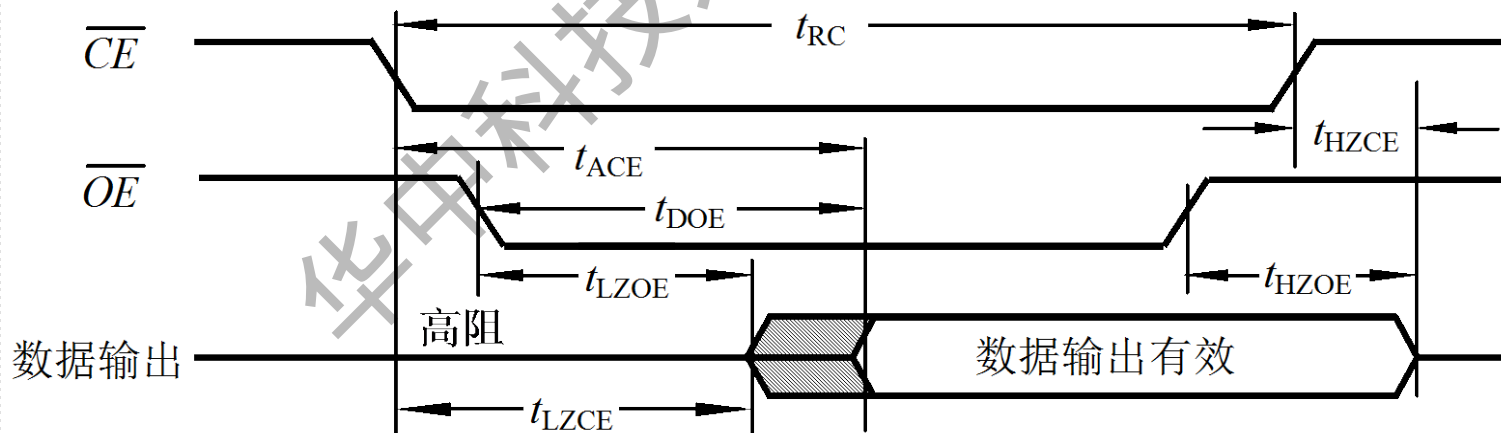
- $T_7$ 、 $T_8$  均导通
- 触发器的输出与数据  
线接通，该单元通过  
数据线读取数据。

### 3. SRAM的读写操作及定时图

#### 读操作定时图

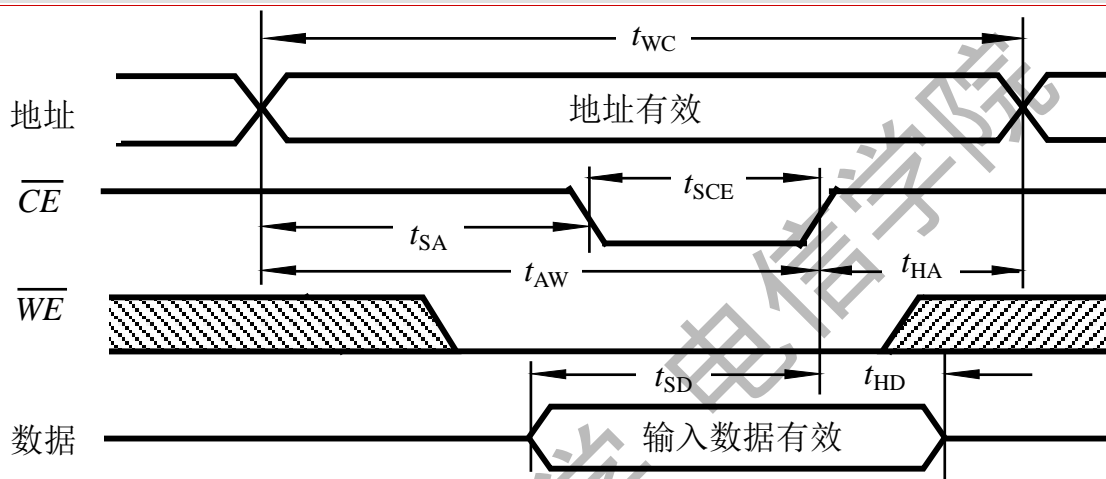


(a) 地址控制的读操作

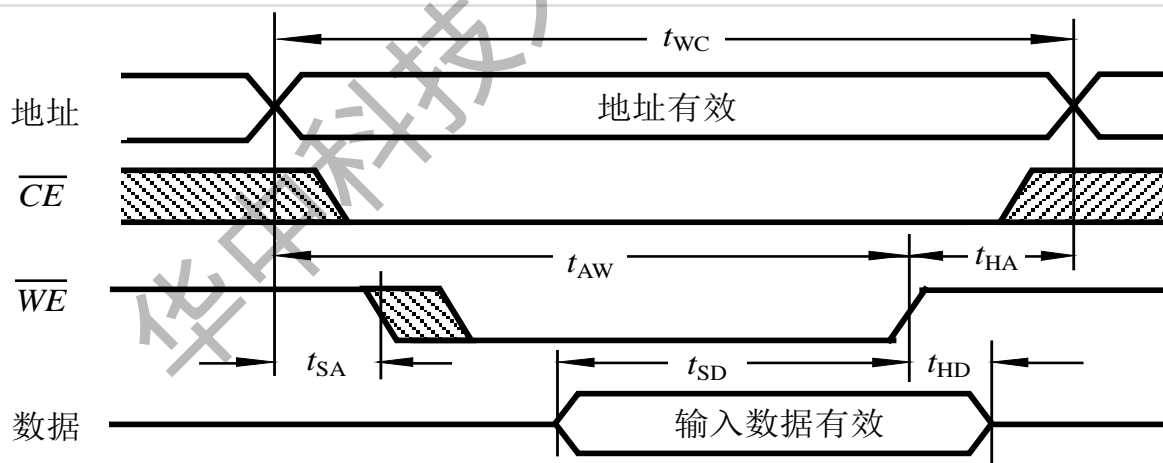


(b) 片选控制的读操作

# 写操作定时图



(a) 片选控制的写操作

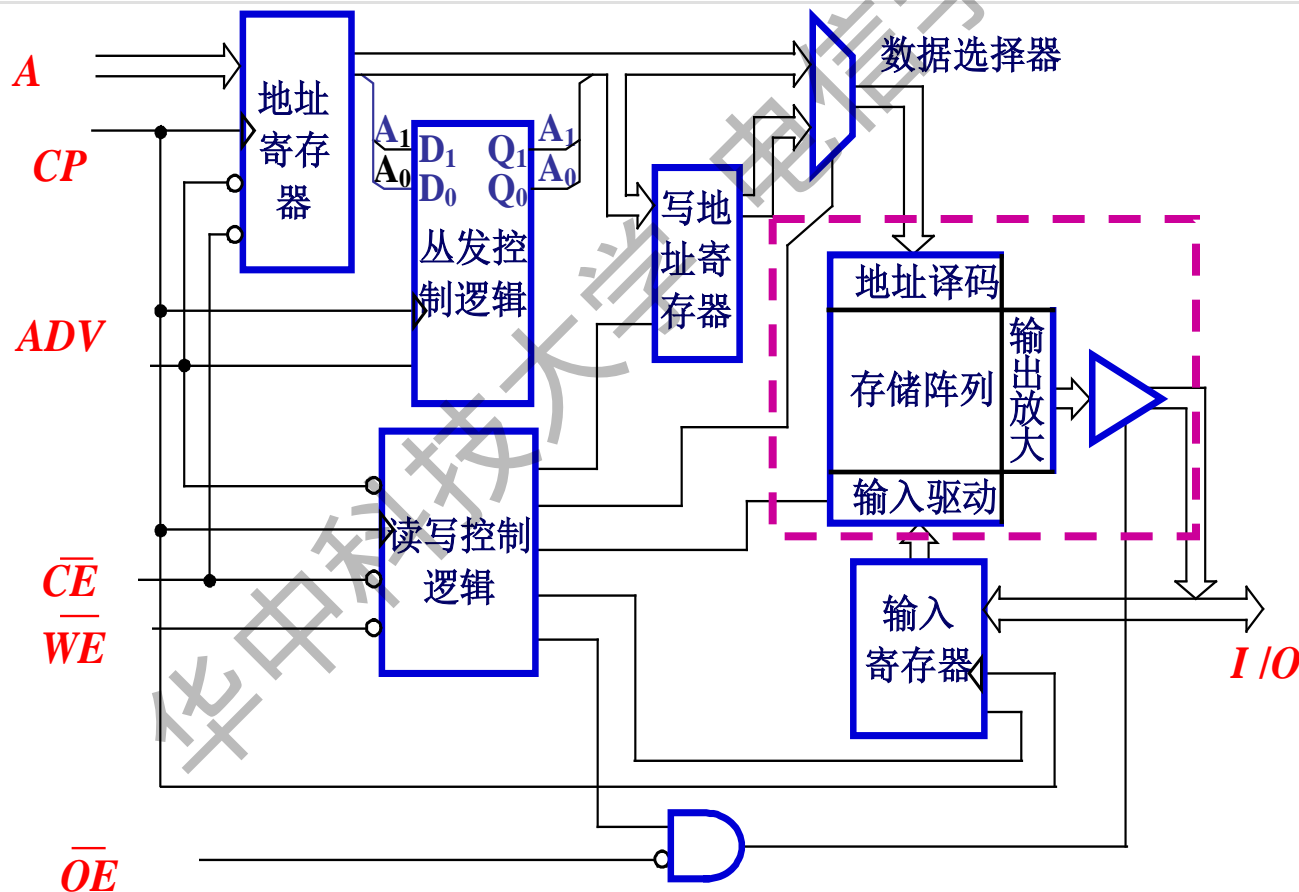


(b) 写信号控制的写操作



## 7.2.2 同步静态随机存取存储器 (SSRAM)

SSRAM是一种高速RAM。与SRAM不同,SSRAM的读写操作是在时钟脉冲节拍控制下完成的。



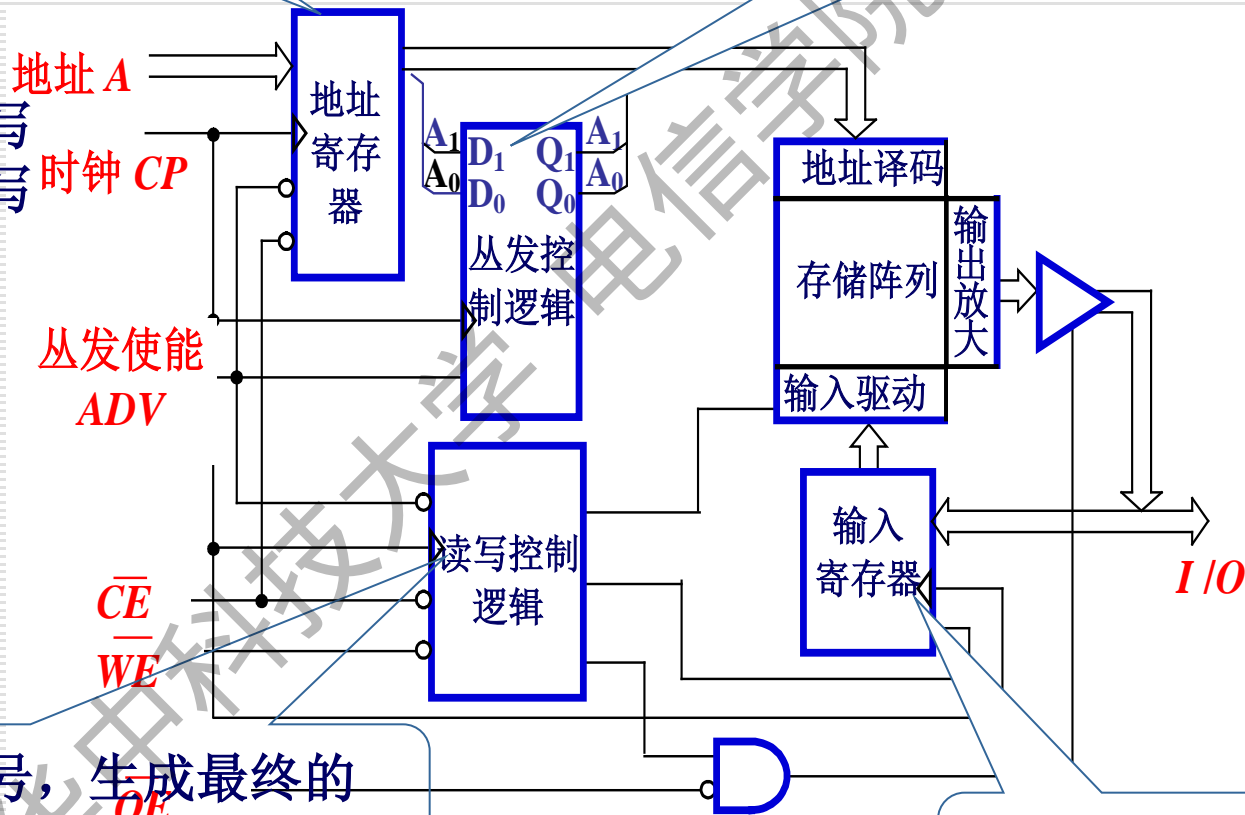
寄存地址线上的地址

2位二进制计数器，  
处理 $A_1A_0$

$ADV=0$ :普通模式读写  
 $ADV=1$ :从发模式读写

$\overline{WE}=0$ :写操作  
 $\overline{WE}=1$ :读操作

寄存各种使能控制信号，生成最终的  
内部读写控制信号；



寄存要写入的  
数据

## **$ADV=0$ :普通模式读写**

$\overline{WE}=0$ :写操作      $\overline{WE}=1$ :读操作

普通模式读写:地址直接送到地址译码器, 不受从发控制逻辑电路影响, 按外部给定地址进行读(写)操作。

## **$ADV=1$ :从发模式读写**

从发模式读写: 地址寄存器不接受外部新地址, 在其原有的地址上, 由从发计数器加1产生新地址。因此, 可产生4个不同的地址。若超过4个CP后,  $ADV$ 仍为1, 则从发计数器循环计数。地址总线让出。

## SSRAM的特点:

---

在由SSRAM构成的计算机系统中，由于在时钟有效沿到来时，地址、数据、控制等信号被锁存到SSRAM内部的寄存器中，因此读写过程的延时等待均在时钟作用下，由SSRAM内部控制完成。此时，系统中的微处理器在读写SSRAM的同时，可以处理其他任务，从而提高了整个系统的工作速度。

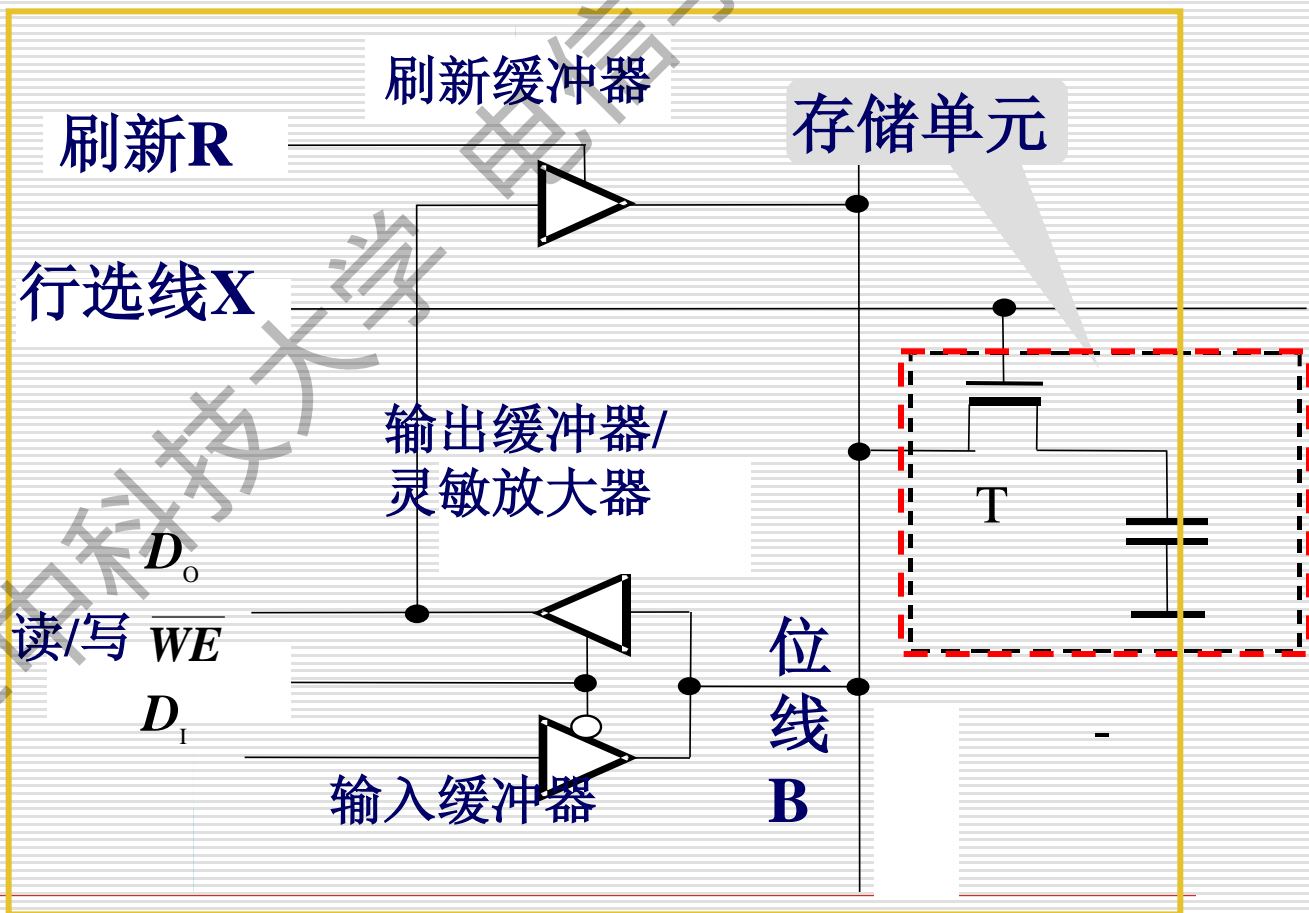
## 7.2.3 动态随机存取存储器(DRAM)

### 1、DRAM存储单元

写操作: $X=1$   $\overline{WE}=0$  T导通, 电容器C与位线B连通

输入缓冲器被选通, 数据 $D_I$ 经缓冲器和位线写入存储单元

如果 $D_I$ 为1, 则向电容器充电, C存1;反之电容器放电, C存0。

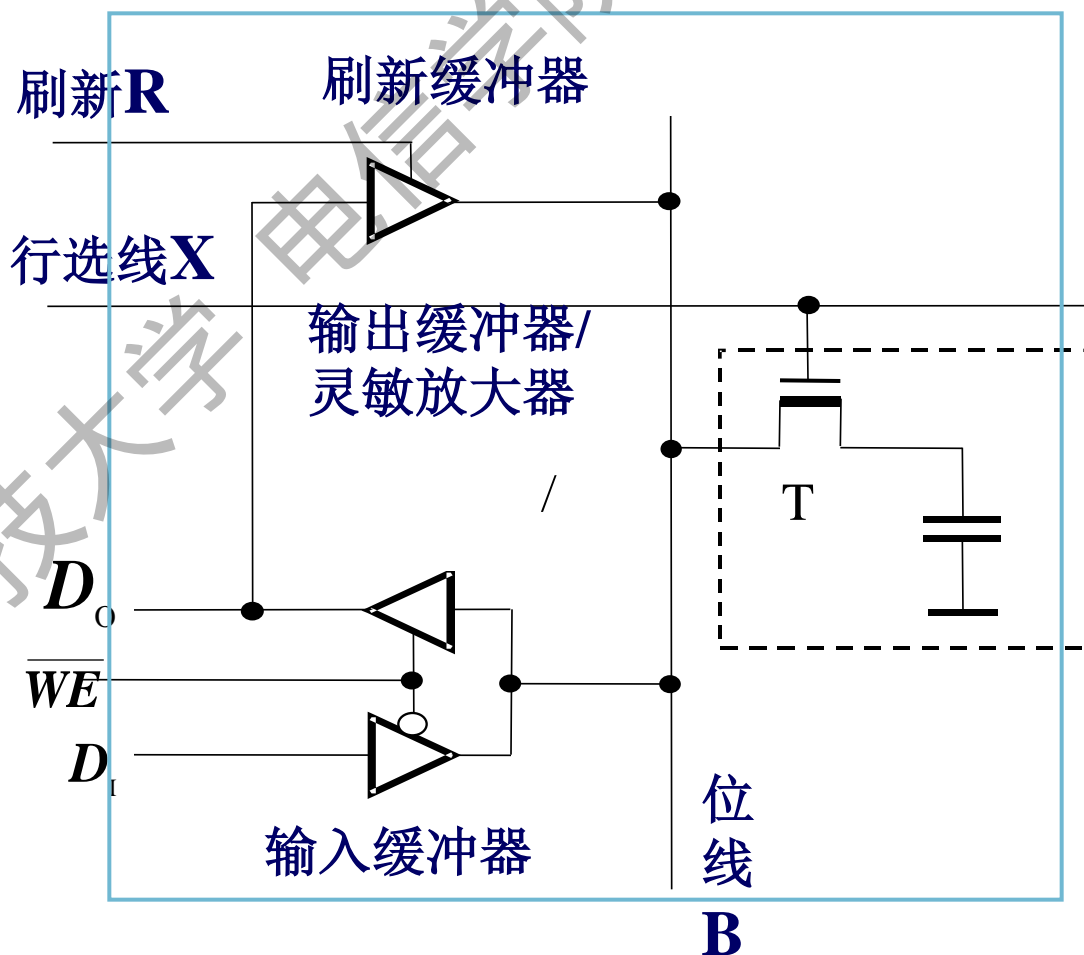


读操作: $X=1$   $\overline{WE}=1$

T导通, 电容器C与位线B连通

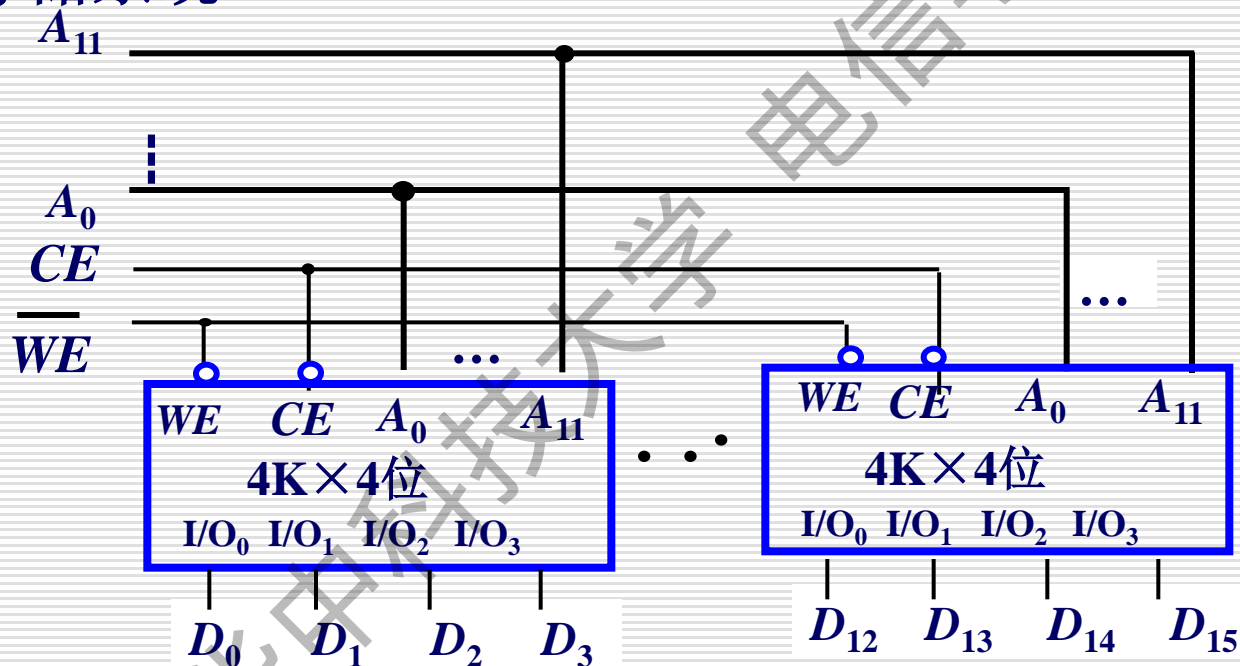
输出缓冲器/灵敏放大器  
被选通, C中存储的数据  
通过位线和缓冲器输出

每次读出后, 必须及时  
对读出单元刷新, 即此  
时刷新控制R也为高电平,  
则读出的数据又经刷新  
缓冲器和位线对电容器C  
进行刷新。



## 7.2.4 存储器容量的扩展

1. 字长（位数）的扩展---用4KX4位的芯片组成4KX16位的存储系统。



位扩展可以利用芯片的并联方式实现。

## 7.2.4 RAM存储容量的扩展

2. 字数的扩展—用8K×8位的芯片组成32K×8位的存储系统。

芯片:  $A_0 \sim A_{12}$

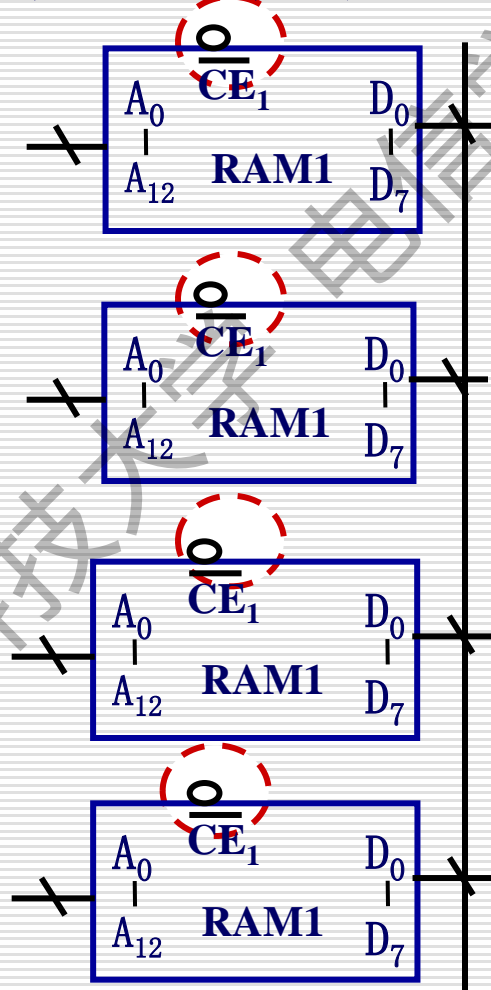
芯片数=4

系统地址线数=15

系统:  $A_0 \sim A_{14}$

$A_{13} \sim A_{14}$ ?

每块芯片的地址范围?



0000H  
0001H  
0002H  
⋮  
1FFFH

2000H  
2001H  
2002H  
⋮  
3FFFH

4000H  
4001H  
4002H  
⋮  
5FFFH

6000H  
6001H  
6002H  
⋮  
7FFFH



芯片RAM1的地址范围?

$A_{14}$	$A_{13}$	$A_{12}$	$A_{11}$	$A_{10}$	$A_9$	$A_8$	$A_7$	$A_6$	$A_5$	$A_4$	$A_3$	$A_2$	$A_1$	$A_0$	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000H
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1FFFH

RAM1

0000H  
0001H  
0002H  
⋮  
1FFFH

RAM2

2000H  
2001H  
2002H  
⋮  
3FFFH

芯片RAM2的地址范围?

$A_{14}$	$A_{13}$	$A_{12}$	$A_{11}$	$A_{10}$	$A_9$	$A_8$	$A_7$	$A_6$	$A_5$	$A_4$	$A_3$	$A_2$	$A_1$	$A_0$	
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	2000H
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	3FFFH

RAM3

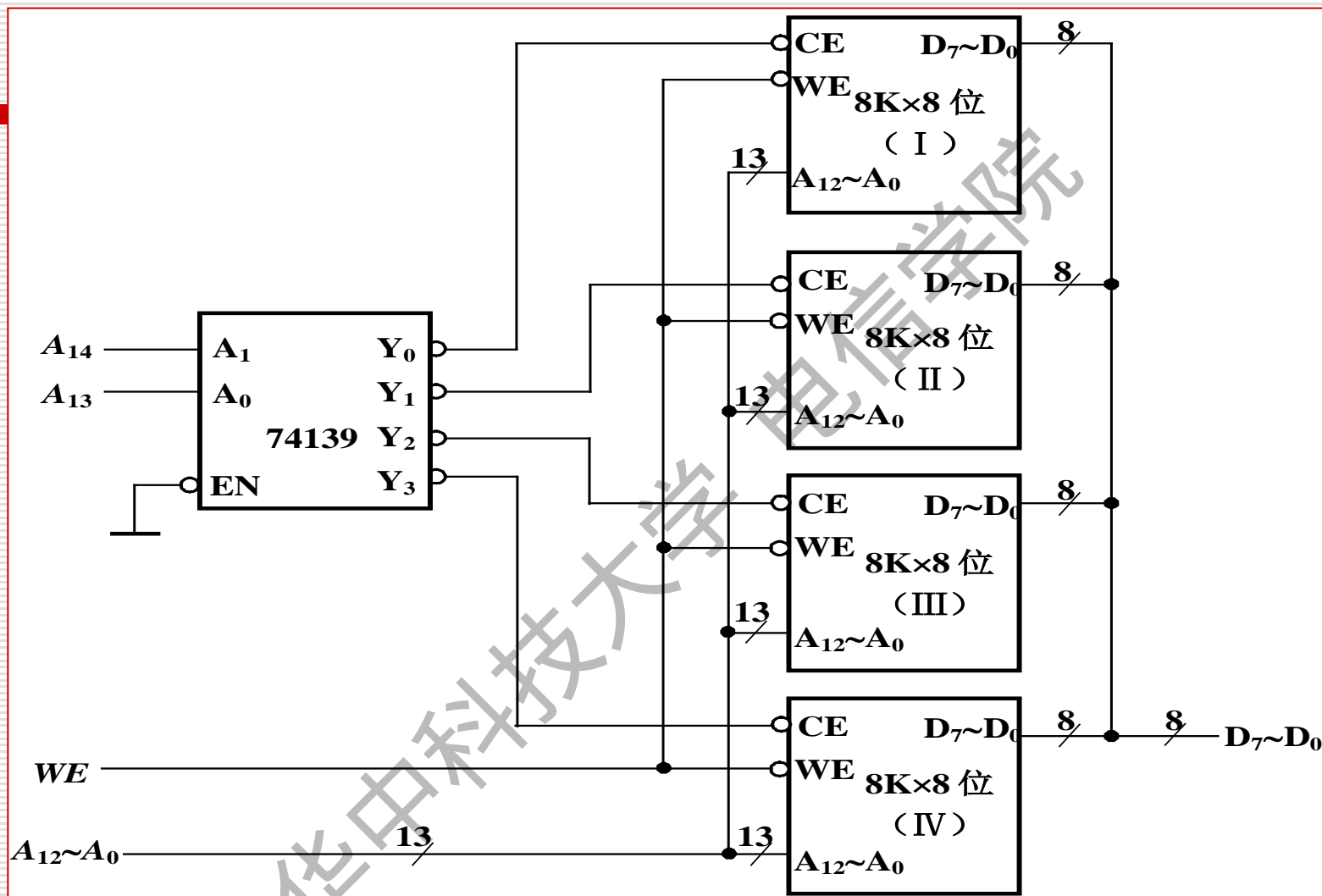
4000H  
4001H  
4002H  
⋮  
5FFFH

RAM4

6000H  
6001H  
6002H  
⋮  
7FFFH

## 32K×8位存储器系统的地址分配表

各RAM芯片	译码器有效输出端	扩展的地址输入端 $A_{14} A_{13}$	8K×8位RAM芯片地址输入端 $A_{12} A_{11} A_{10} A_9 A_8 A_7 A_6 A_5 A_4 A_3 A_2 A_1 A_0$	对应的十六进制地址码
I	$Y_0$	0 0	<div>0 0 0 0 0 0 0 0 0 0 0 0 0</div> <div>0 0 0 0 0 0 0 0 0 0 0 0 1</div> <div>0 0 0 0 0 0 0 0 0 0 0 1 0</div> <div>⋮</div> <div>1 1 1 1 1 1 1 1 1 1 1 1 1</div>	0000H 0001H 0002H ⋮ 1FFFH
II	$Y_1$	0 1	<div>0 0 0 0 0 0 0 0 0 0 0 0 0</div> <div>0 0 0 0 0 0 0 0 0 0 0 0 1</div> <div>0 0 0 0 0 0 0 0 0 0 0 1 0</div> <div>⋮</div> <div>1 1 1 1 1 1 1 1 1 1 1 1 1</div>	2000H 2001H 2002H ⋮ 3FFFH
III	$Y_2$	1 0	<div>0 0 0 0 0 0 0 0 0 0 0 0 0</div> <div>0 0 0 0 0 0 0 0 0 0 0 0 1</div> <div>0 0 0 0 0 0 0 0 0 0 0 1 0</div> <div>⋮</div> <div>1 1 1 1 1 1 1 1 1 1 1 1 1</div>	4000H 400H 4002H ⋮ 5FFFH
IV	$Y_3$	1 1	<div>0 0 0 0 0 0 0 0 0 0 0 0 0</div> <div>0 0 0 0 0 0 0 0 0 0 0 0 1</div> <div>0 0 0 0 0 0 0 0 0 0 0 1 0</div> <div>⋮</div> <div>1 1 1 1 1 1 1 1 1 1 1 1 1</div>	6000H 6001H 6002H ⋮ 7FFFH



字数的扩展可以利用外加译码器控制存储器芯片的片选输入端来实现。

### 3.同时实现字数、位数的扩展

---

使用 $256 \times 4$ 位芯片组成 $512 \times 8$ 位存储器，问需要多少芯片？电路应如何连接？

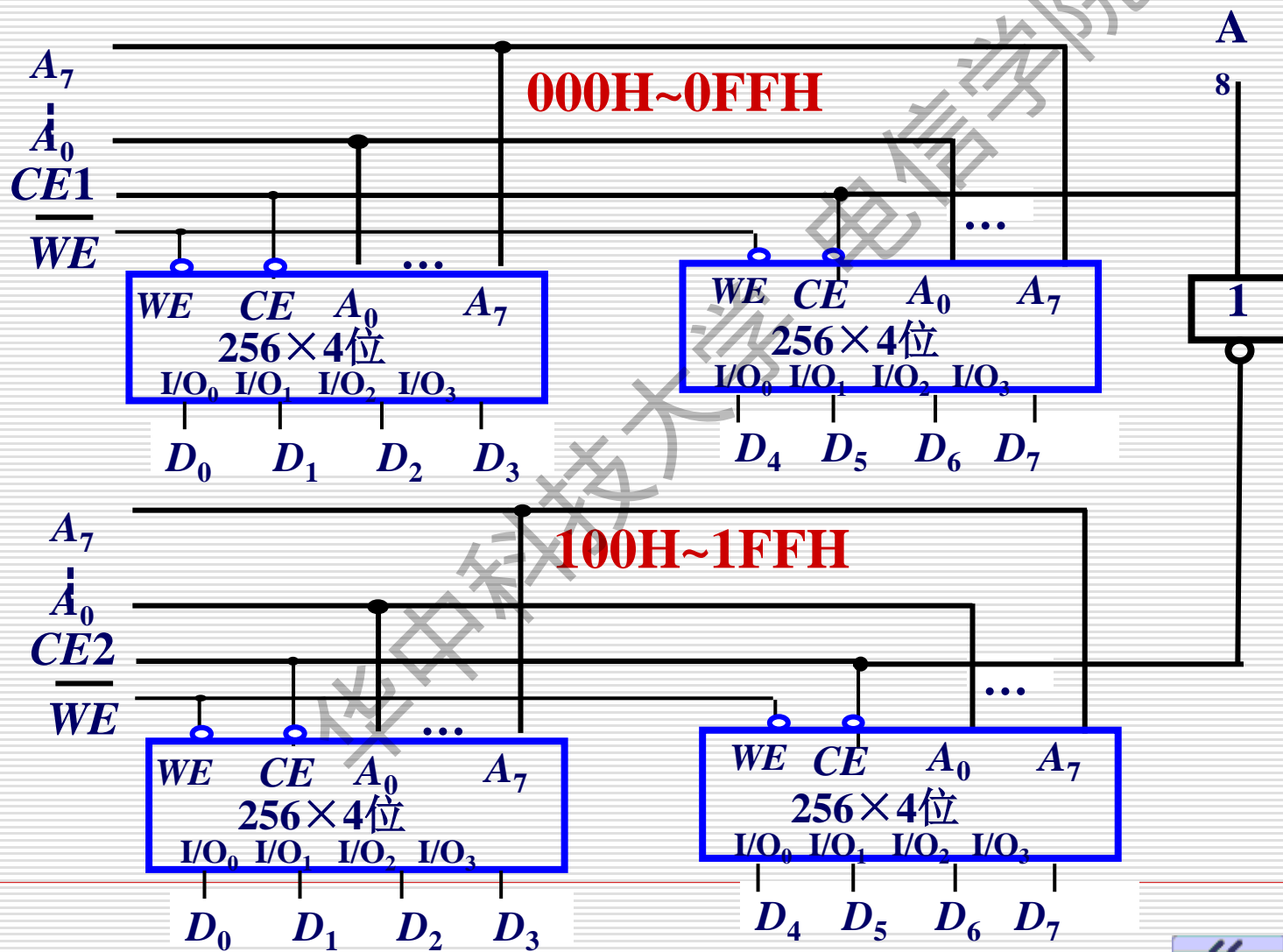
电路设计：

1、位扩展组成 $256 \times 8$ ：采用两块芯片的并联方式实现。

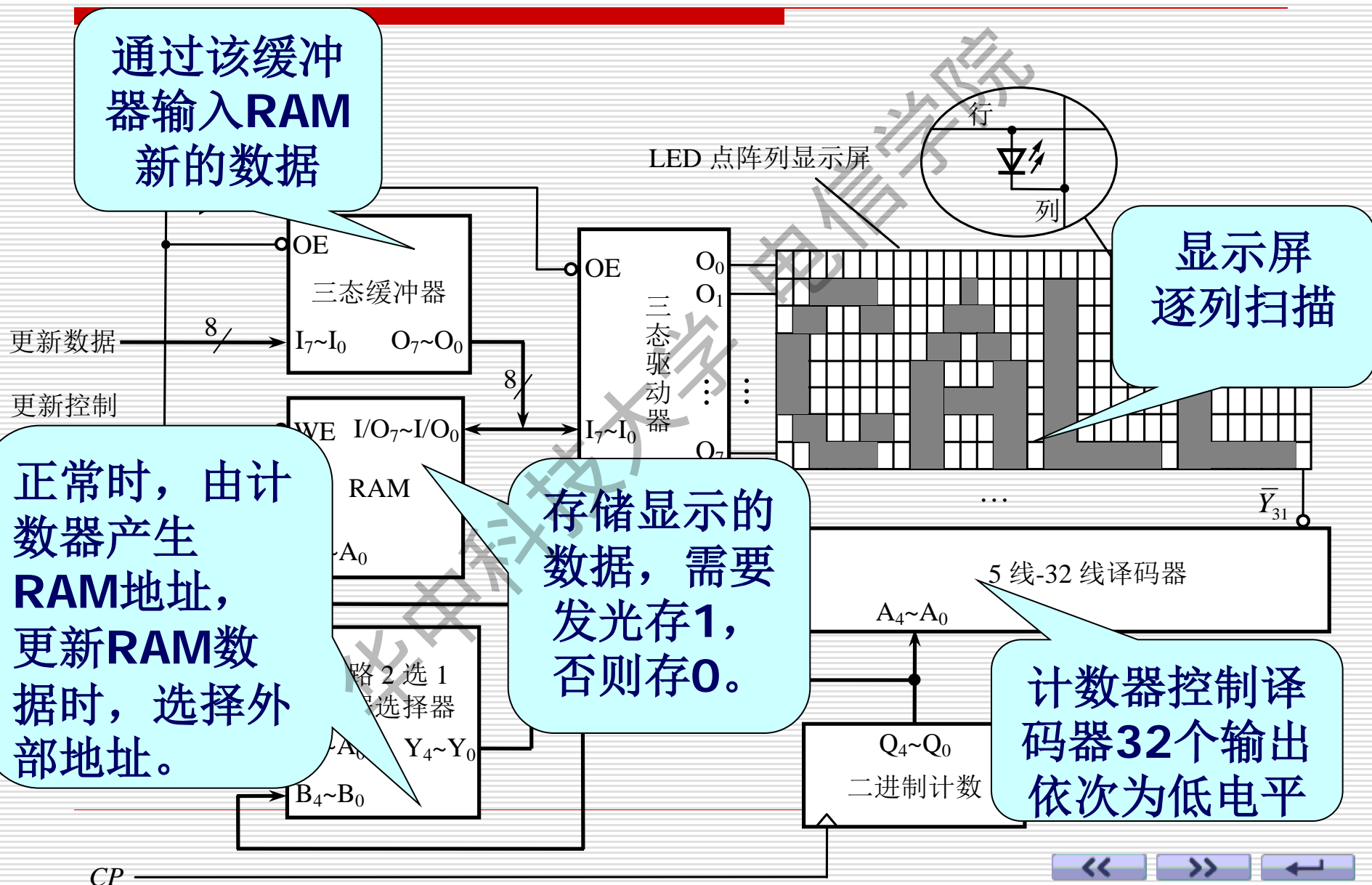
2、字扩展组成 $512 \times 8$

采用两组 $256 \times 8$ 采用字扩展方式实现。

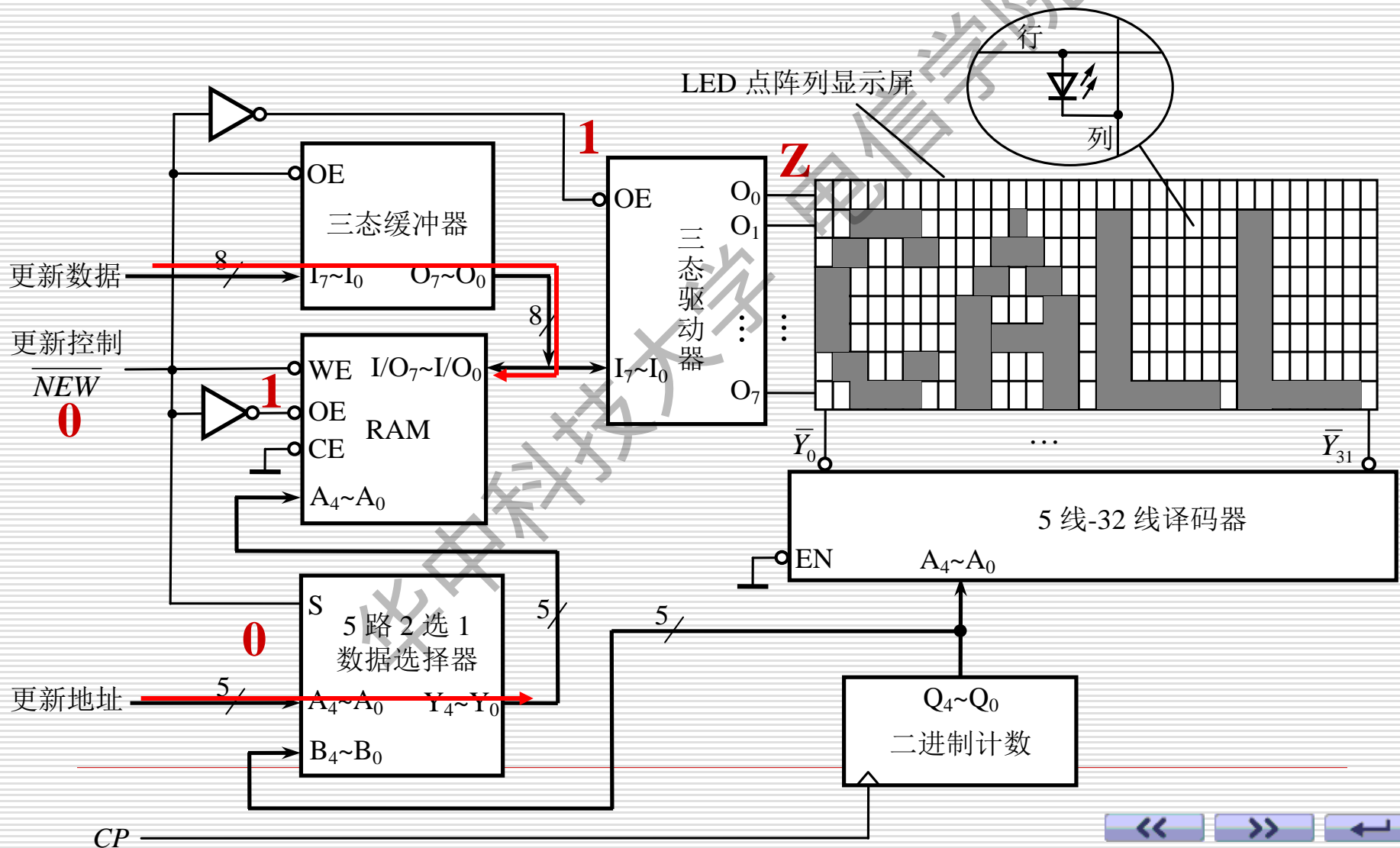
## 每块芯片的地址范围？



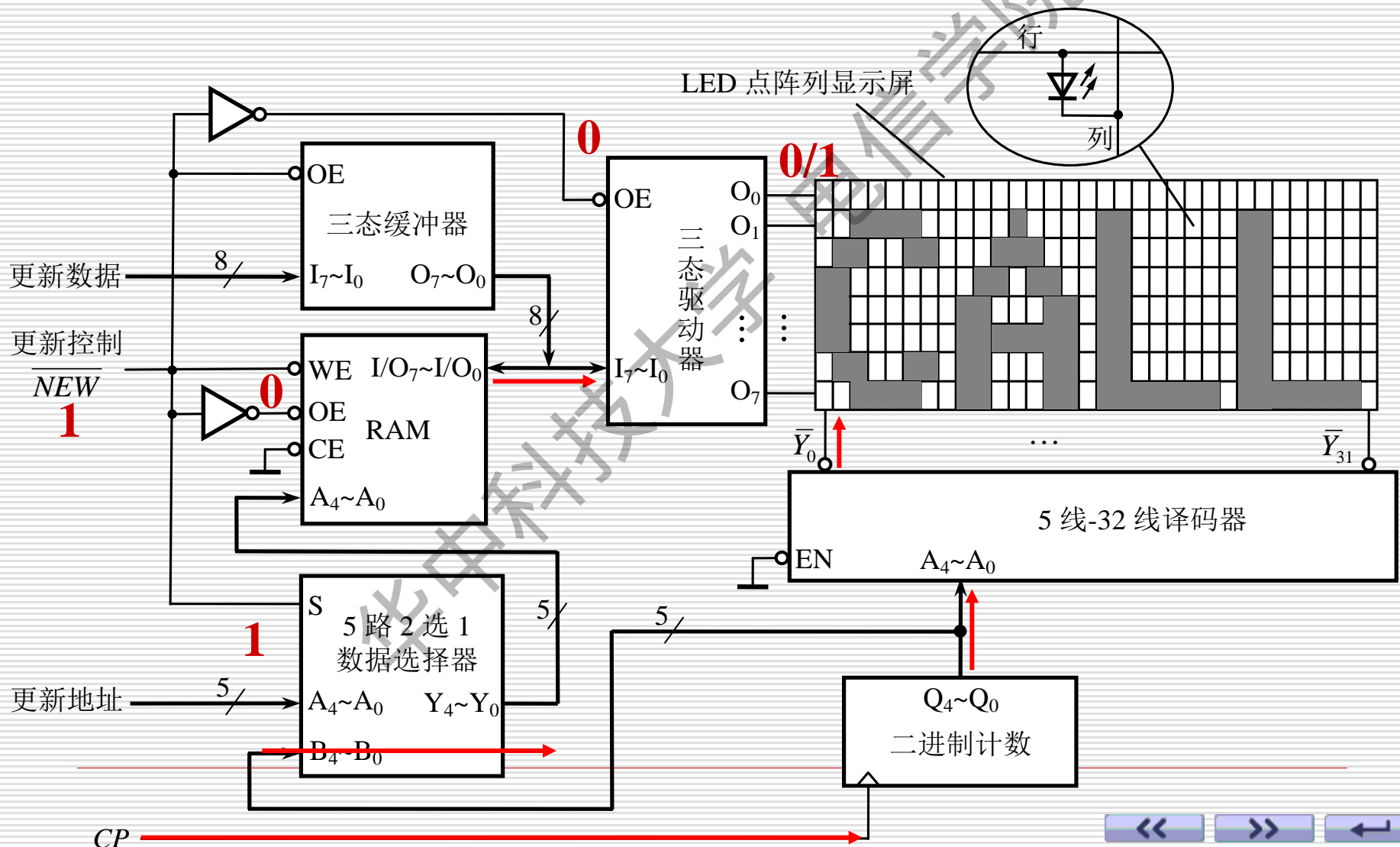
## 7.2.5 RAM应用举例—LED点阵显示屏



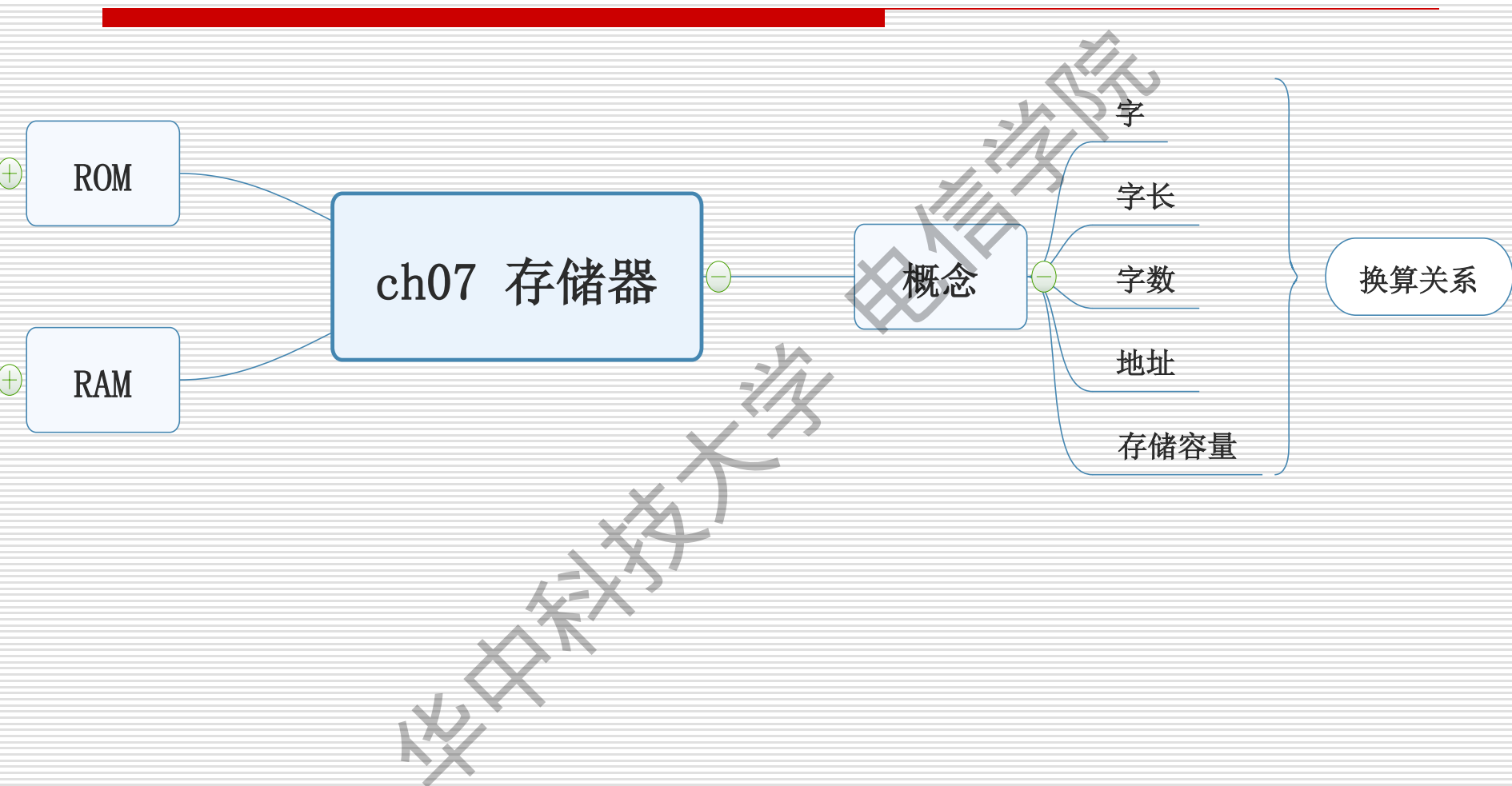
# 更新RAM数据

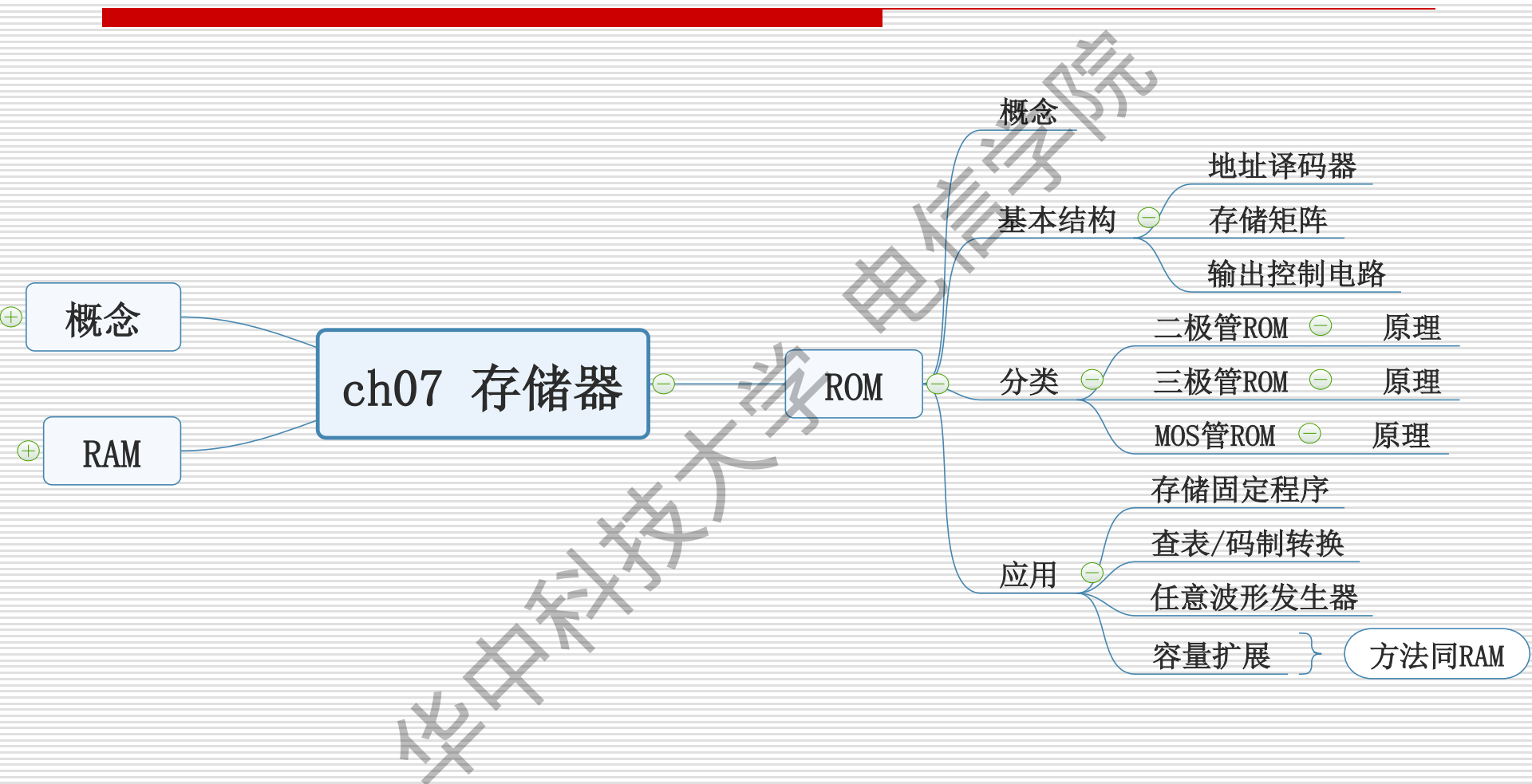


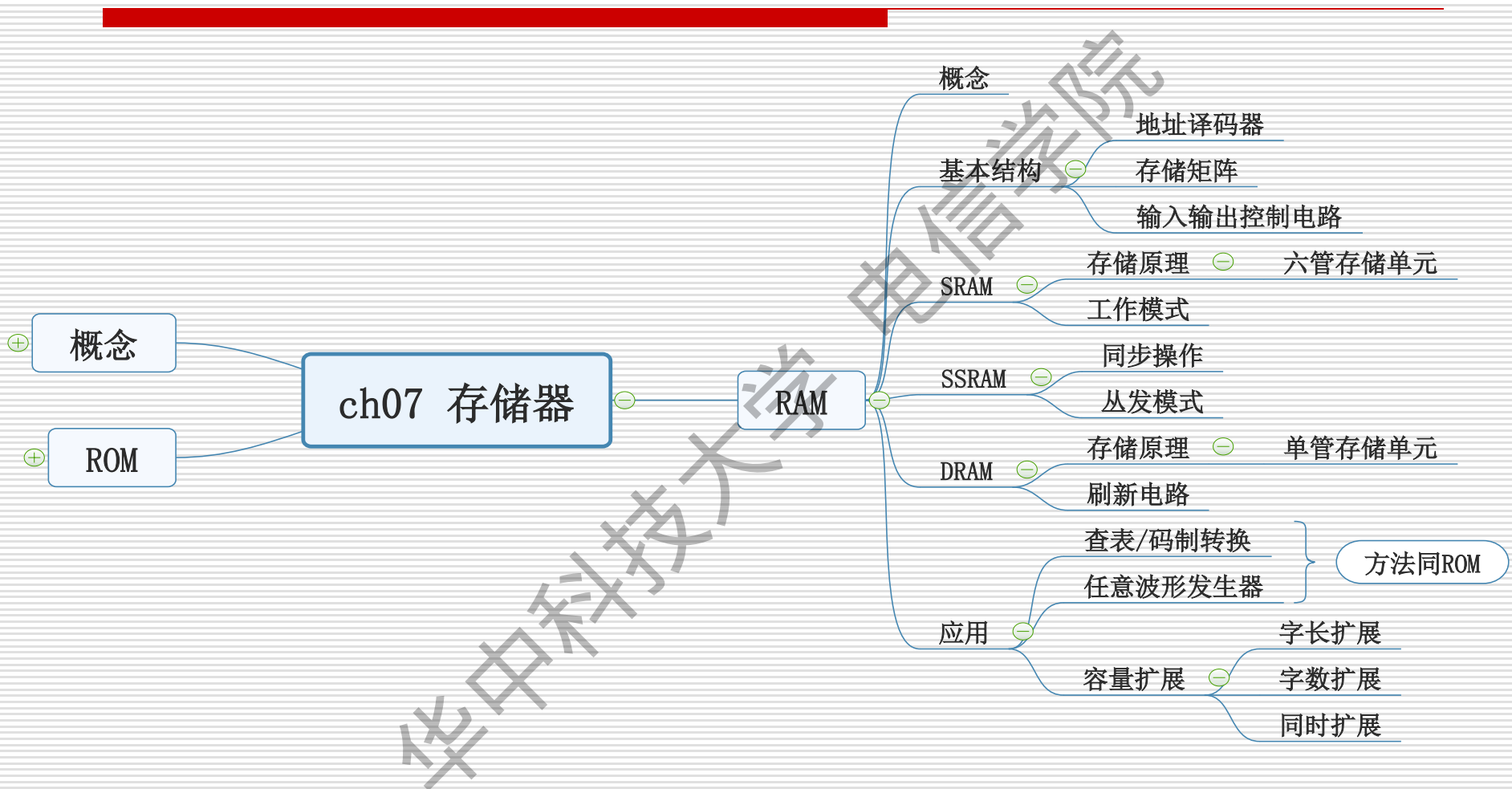
# 正常显示











# ch07 存储器

## 概念

换算关系

字  
字长  
字数  
地址  
存储容量

## ROM

### 概念

### 基本结构

地址译码器

存储矩阵

输出控制电路

### 分类

原理 ⊖ 二极管ROM

原理 ⊖ 三极管ROM

原理 ⊖ MOS管ROM

### 应用

存储固定程序

查表/码制转换

任意波形发生器

容量扩展

方法同RAM

## RAM

### 概念

地址译码器

存储矩阵

输入输出控制电路

### 基本结构

### SRAM

存储原理 ⊖ 六管存储单元

工作模式

### SSRAM

同步操作

丛发模式

### DRAM

存储原理 ⊖ 单管存储单元

刷新电路

### 应用

查表/码制转换

任意波形发生器

方法同ROM

容量扩展

字长扩展

字数扩展

同时扩展

# 作业

---

➤ 7.1.1

➤ 7.1.4

➤ 7.1.3

➤ 7.1.5

华中科技大学 电信学院