

## 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

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

高阻

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

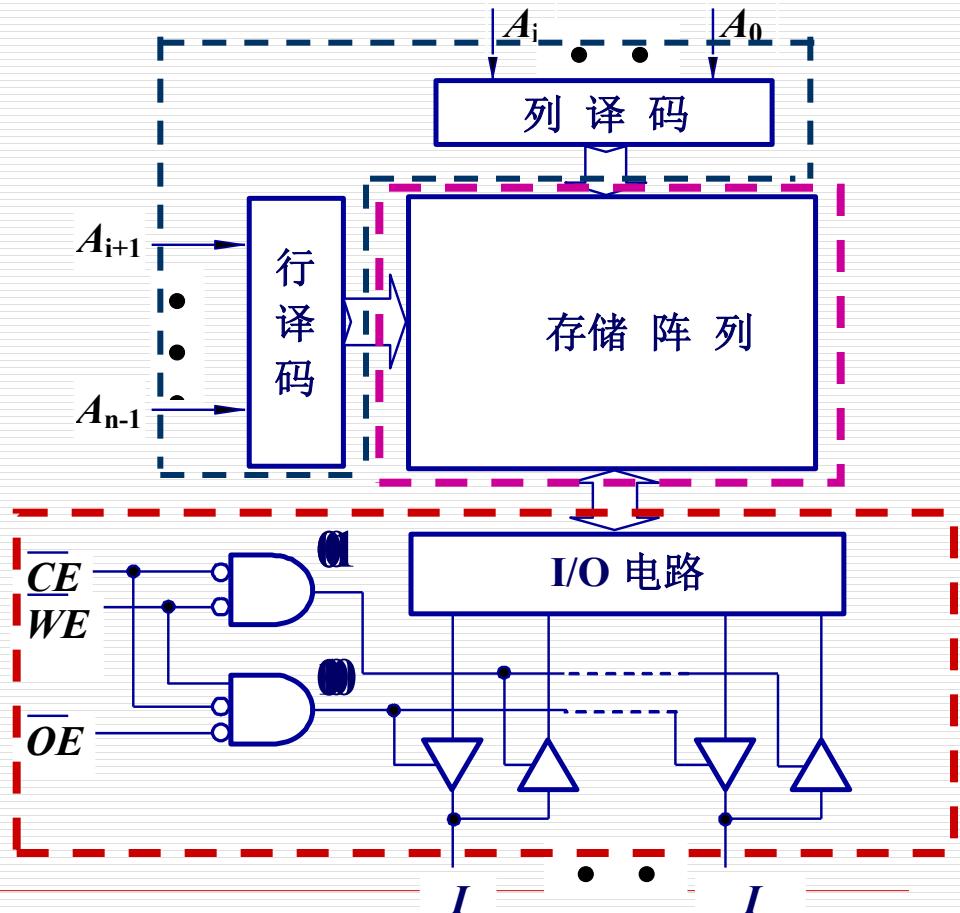
输出

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

输入

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

高阻

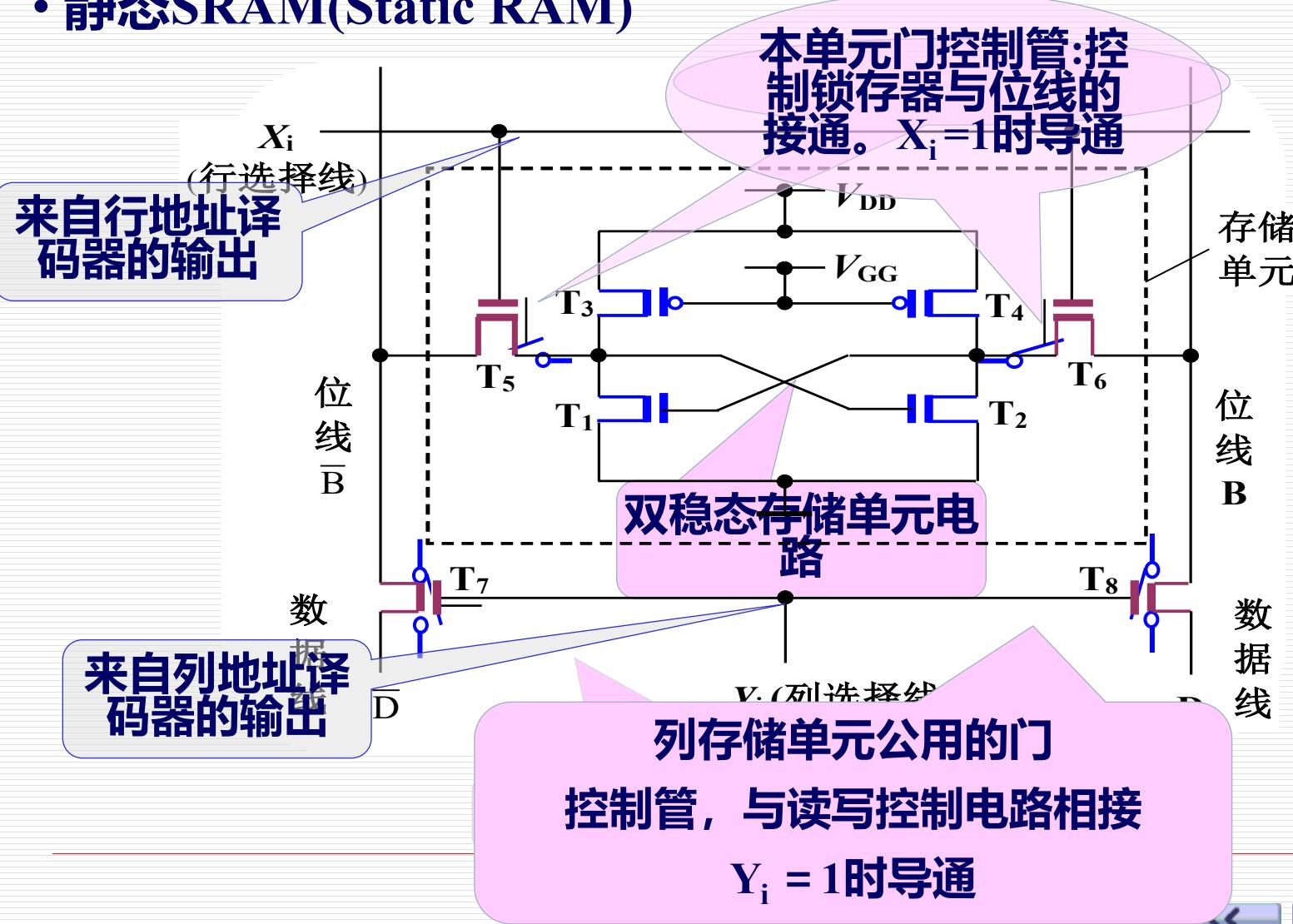


## SRAM 的工作模式

| 工作模式     | $\overline{CE}$ | $\overline{WE}$ | $\overline{OE}$ | $I/O_0 \sim I/O_{m1}$ |
|----------|-----------------|-----------------|-----------------|-----------------------|
| 保持 (微功耗) | 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                    |

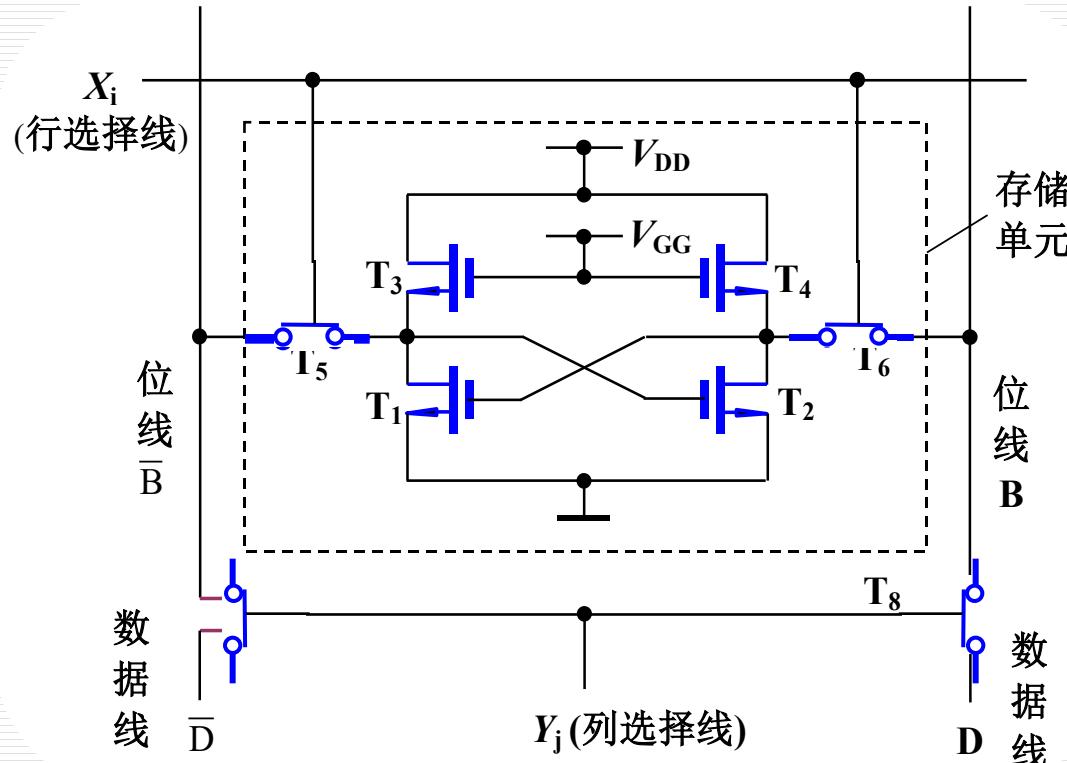
## 2. RAM存储单元

### • 静态SRAM(Static RAM)



## 2. RAM存储单元

- 静态SRAM(Static RAM)



$$X_i = 1$$

- $T_5$ 、 $T_6$ 导通

锁存器与位线接通

$$Y_j = 1$$

- $T_7$ 、 $T_8$ 均导通

• 锁存器的输出与数据线接通，该单元通过数据线传送数据。

因此，存储单元进行读/写操作的条件是：与之相连的行、列选择线均为高电平 ( $X_i = 1$  ,  $Y_j = 1$ )

读操作时，

### 3. SRAM的读写操作及定时图

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

#### 读操作定时图

输出保  
持时间

地址

读周期

数据被传到数据线上

输出数据

读出单元的地址有效

$t_{AA}$

$t_{OHA}$

地址存  
取时间

读周期

数据输出有效

(a) 地址控制的读操作

$\overline{CE}$

$\overline{OE}$

输出使能  
高阻维持  
时间

输出使  
能时间

片选保  
持时间

数据输出

片选高阻  
维持时间

数据输出有效

(b) 片选控制的读操作

写操作时，

## 写操作定时图

写周期

$CE$

$WE$

$OE$

= 00X

数据线上数据被写入SRAM

地址建立时间

地址

地址有效

$t_{WC}$

$CE$

$\overline{WE}$

$t_{SA}$

$t_{AW}$

$t_{SCE}$

$t_{HA}$

写结束前地址保持时间

数据

输入数据有效

(a) 片选控制的写操作  
写结束前数据建立时间

$t_{WC}$

地址有效

地址

$CE$

$\overline{WE}$

数据

$t_{AW}$

$t_{HA}$

$t_{SA}$

$t_{SD}$

$t_{HD}$

输入数据有效

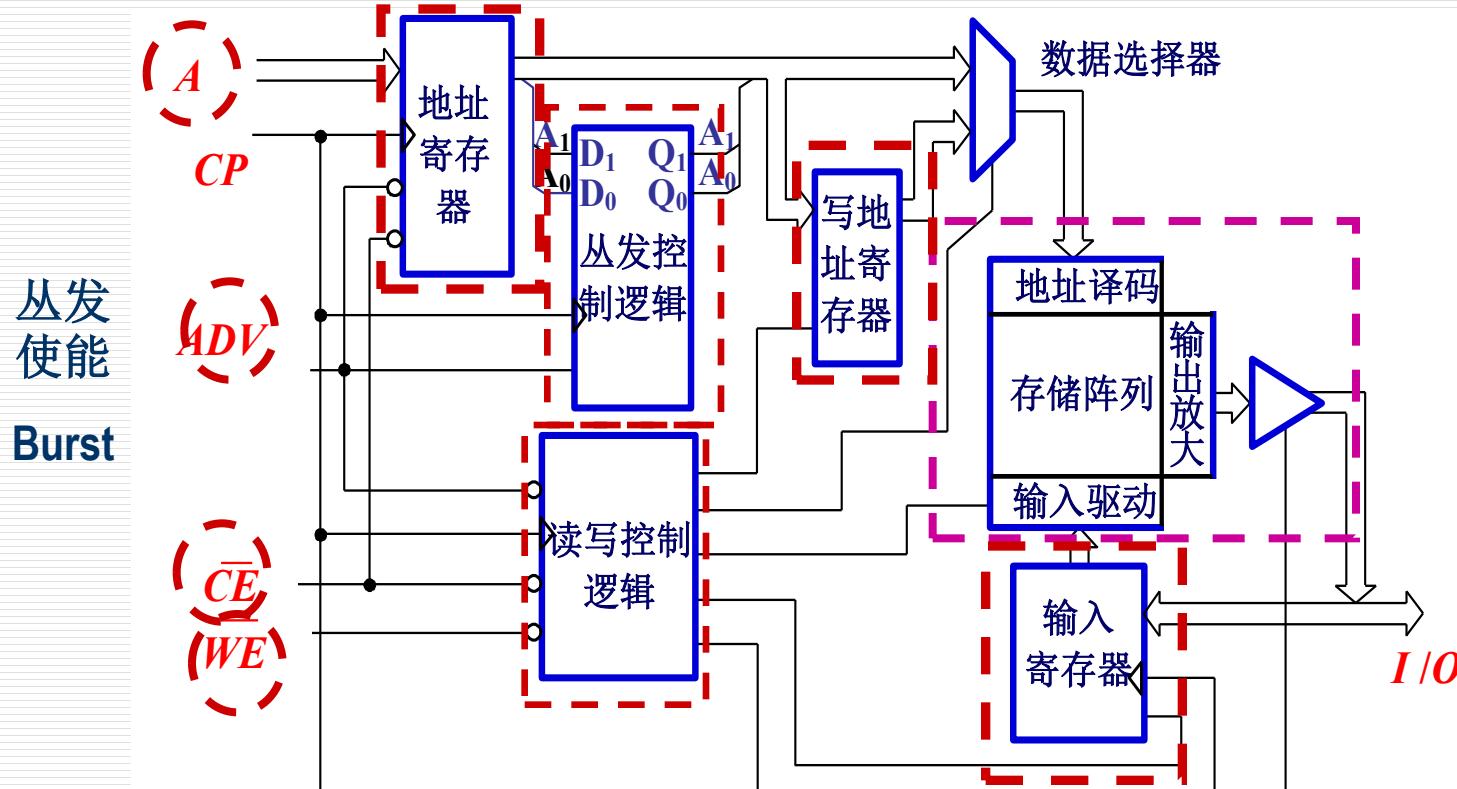
写结束后地址维持时间

写结束后数据维持时间

(b) 写信号控制的写操

## 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。区别于异步SRAM。



SSRAM: 在时钟有效沿到达时, 地址、数据、控制等信号被锁存到SSRAM内部的寄存器中, 读写过程的延时等待被控制在时钟作用间隔内。好处: 应用简单、工作速度快

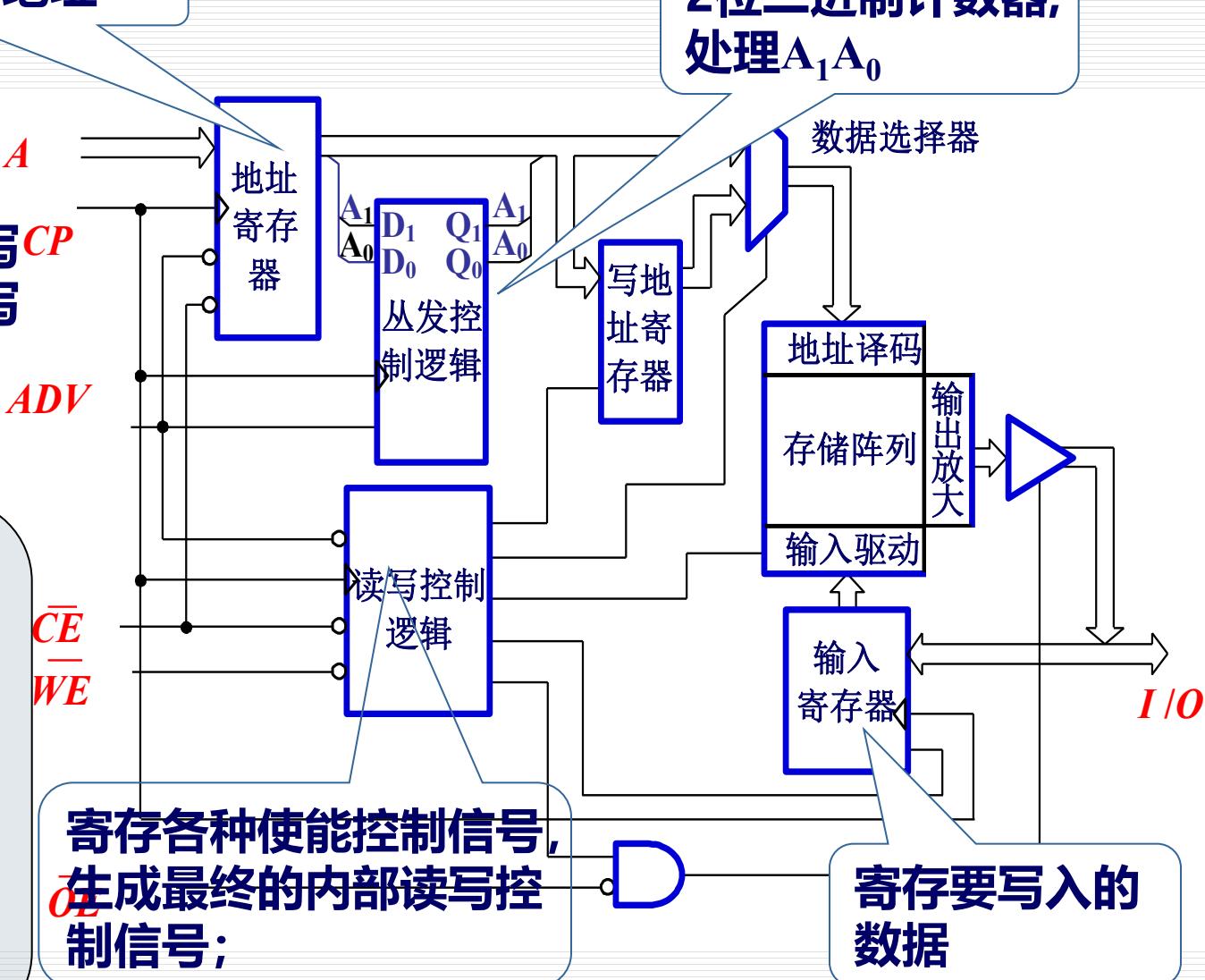
寄存地址线上的地址

2位二进制计数器，  
处理  $A_1 A_0$

$ADV=0$ : 普通模式读写 CP  
 $ADV=1$ : 从发模式读写

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

在该工作模式下，只要外部给定读写存储单元的首地址，在CP作用下，内部地址计数器会提供首地址之后的一组连续地址，此时不需要外部输入地址。因此可以减少外部地址总线占用时间，提高读写效率

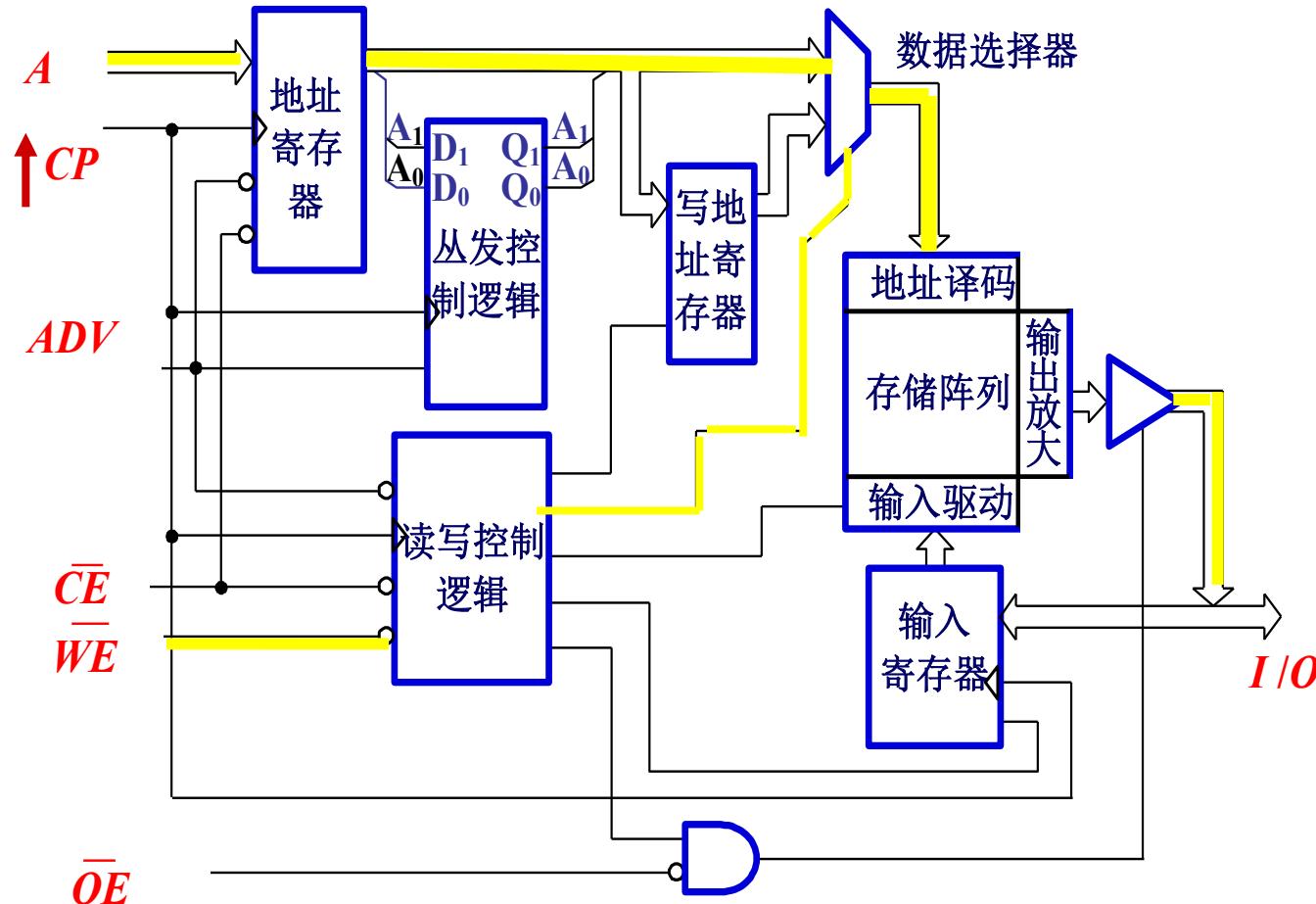


寄存各种使能控制信号，  
 $D$  生成最终的内部读写控  
制信号；

寄存要写入的  
数据

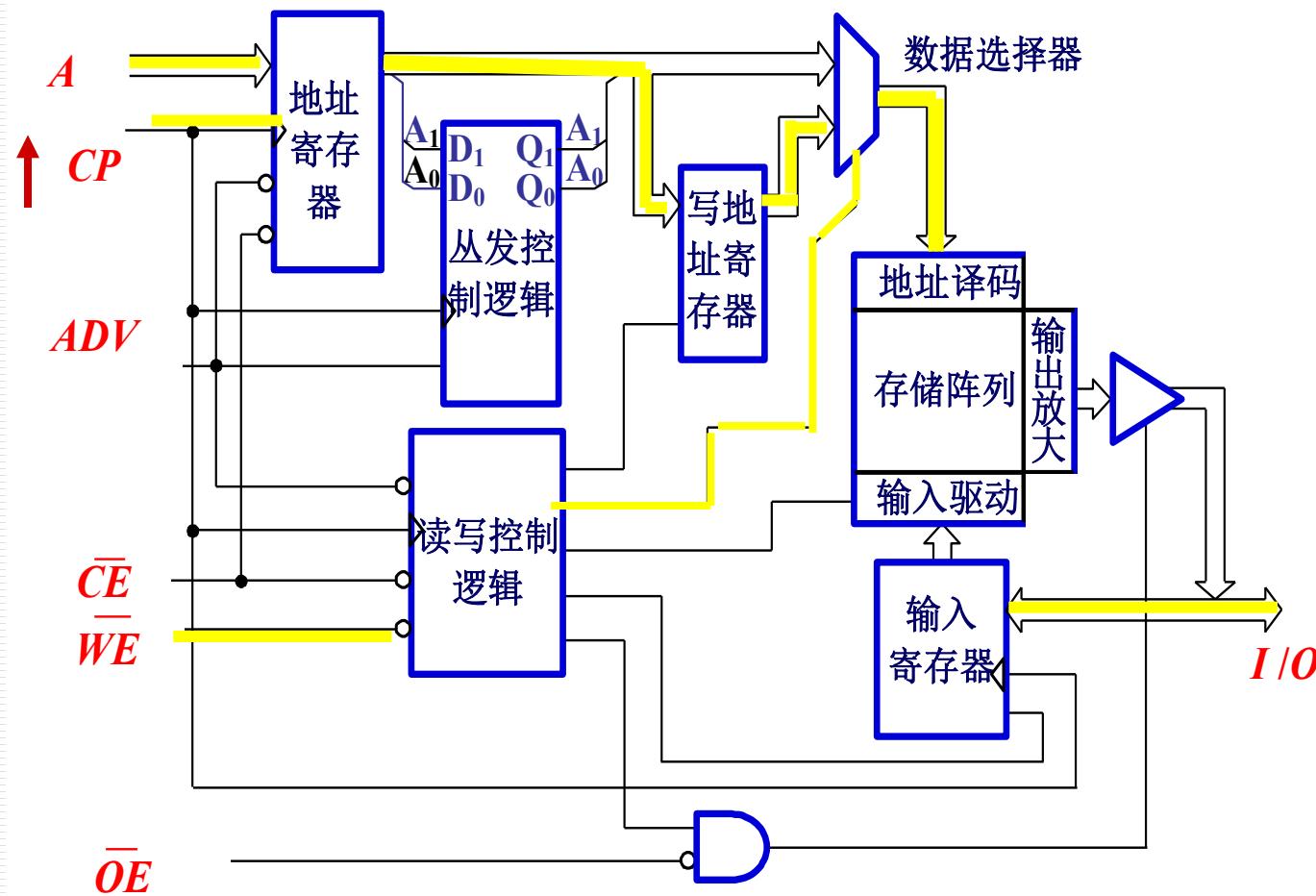
$ADV=0$ :普通模式读写

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



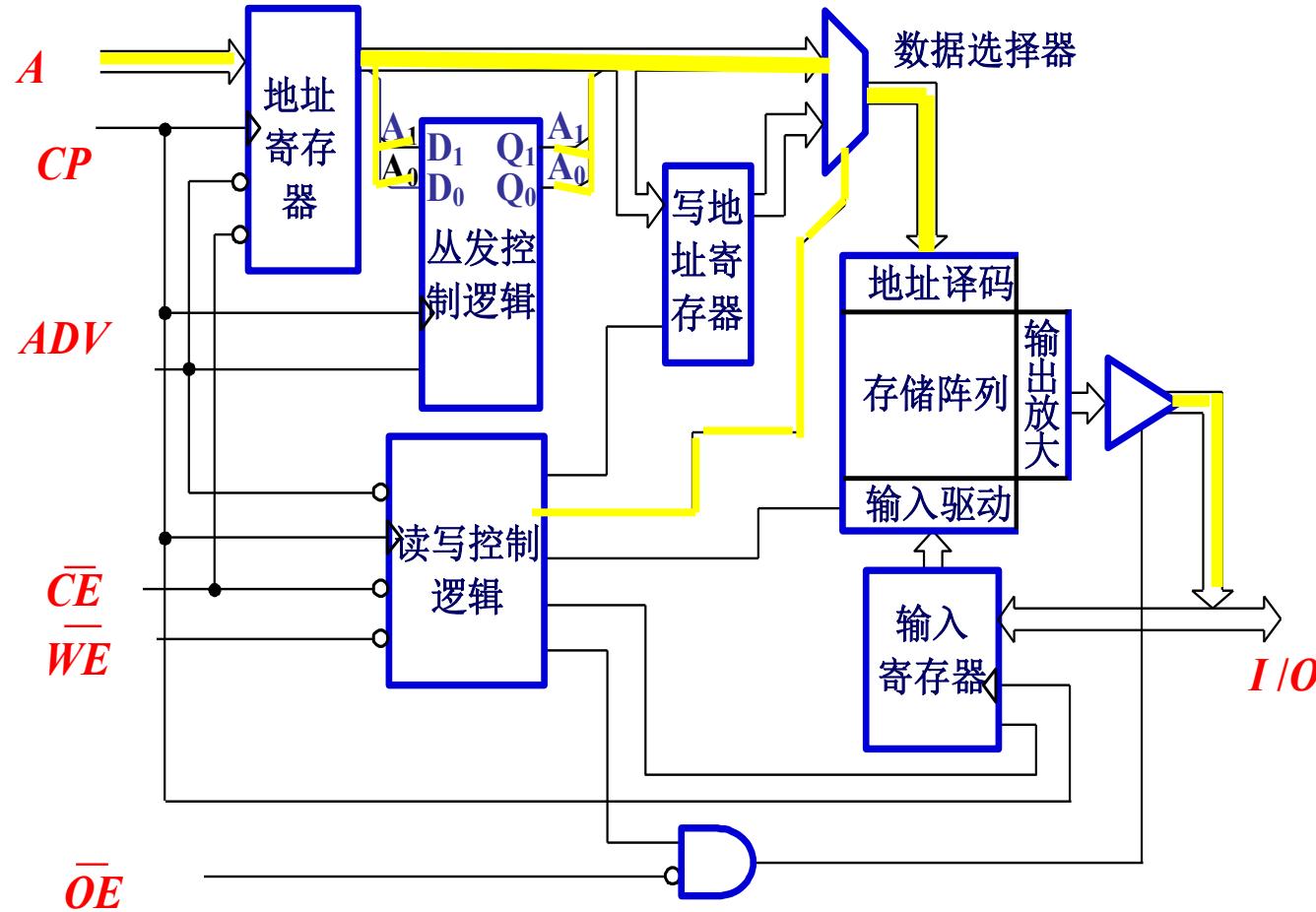
$ADV=0$ :普通模式读写

$\overline{CE} = 0$   $\overline{WE} = 0$ :写操作



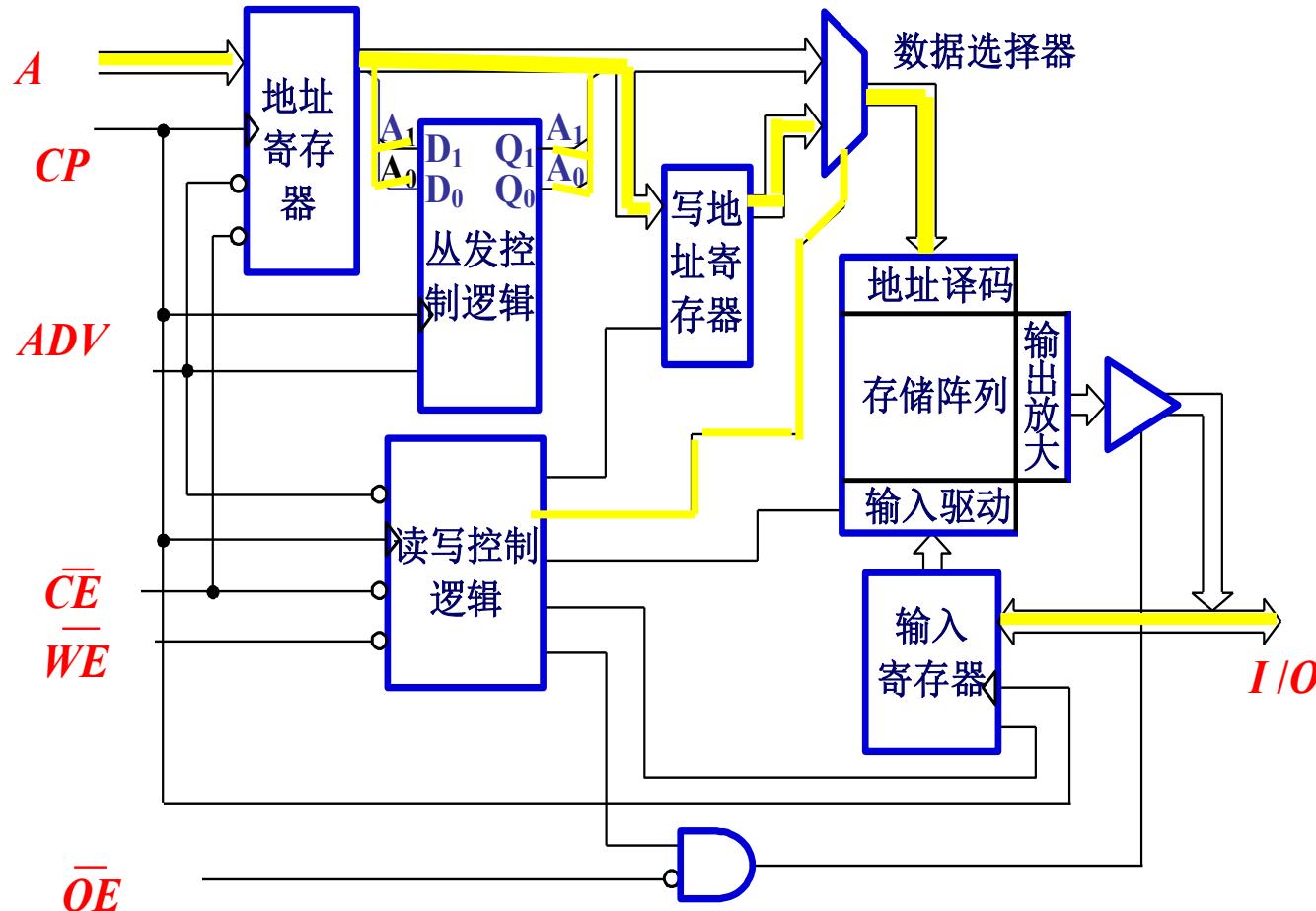
$ADV=1$ : 从发模式读写

$\overline{WE}=1$ : 读操作



$ADV=1$ : 从发模式读写

$\overline{WE}=0$ : 写操作



## ***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。SSRAM的丛发模式减少了外部地址总线的占用时间，提高了读写效率

## 其它同步SSRAM：

DDR（双倍数据传输率，上升沿和下降沿各传输一次数据）

QDR（四倍数据传输率，上升沿和下降沿各传输一次数据外，为读写分别提供独立的接口，使得读写可以同时进行）

## 7.2.3 动态随机存取存储器(DRAM)

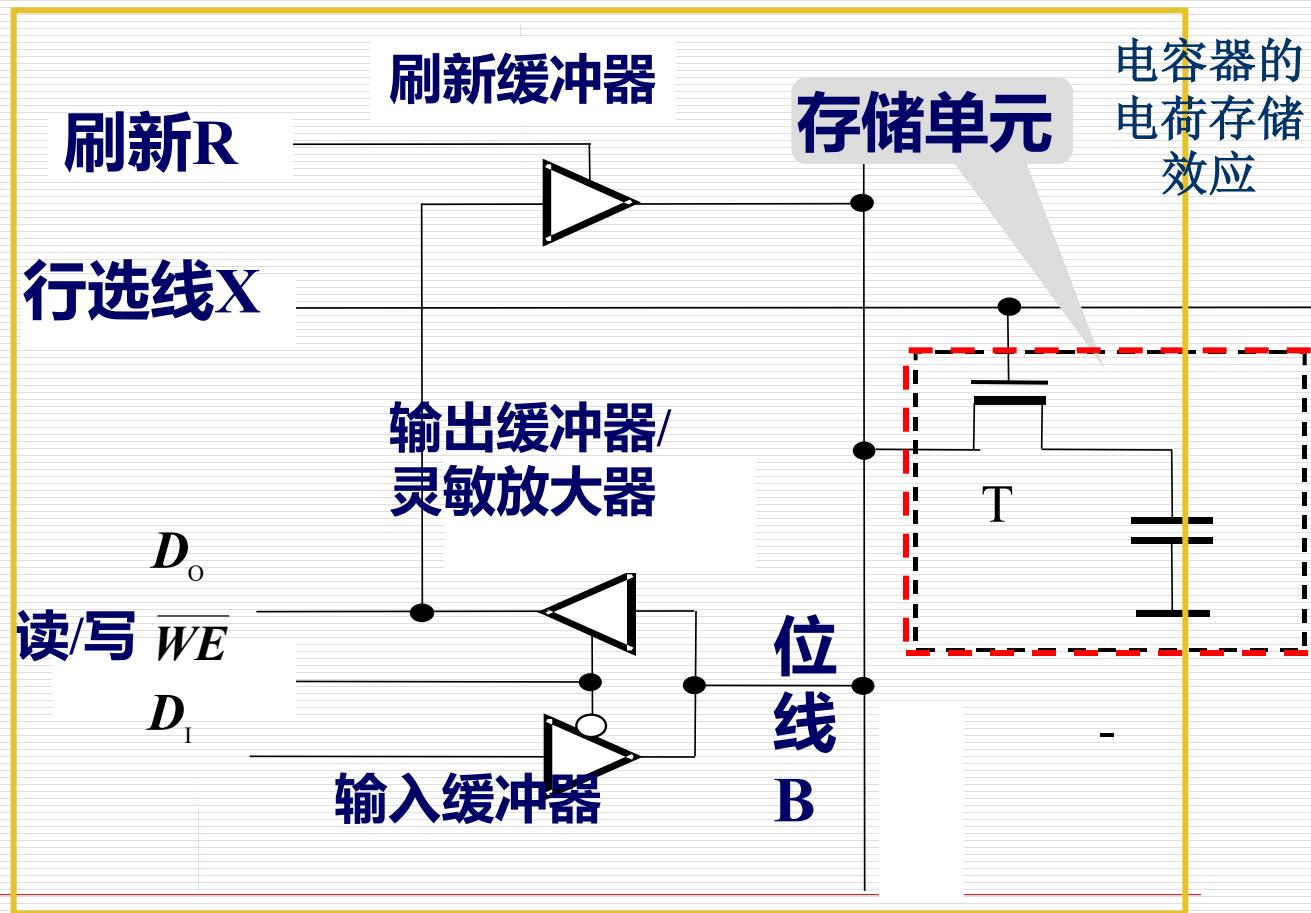
### 1、DRAM存储单元

写操作: $X=1 \quad \overline{WE}=0$

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。

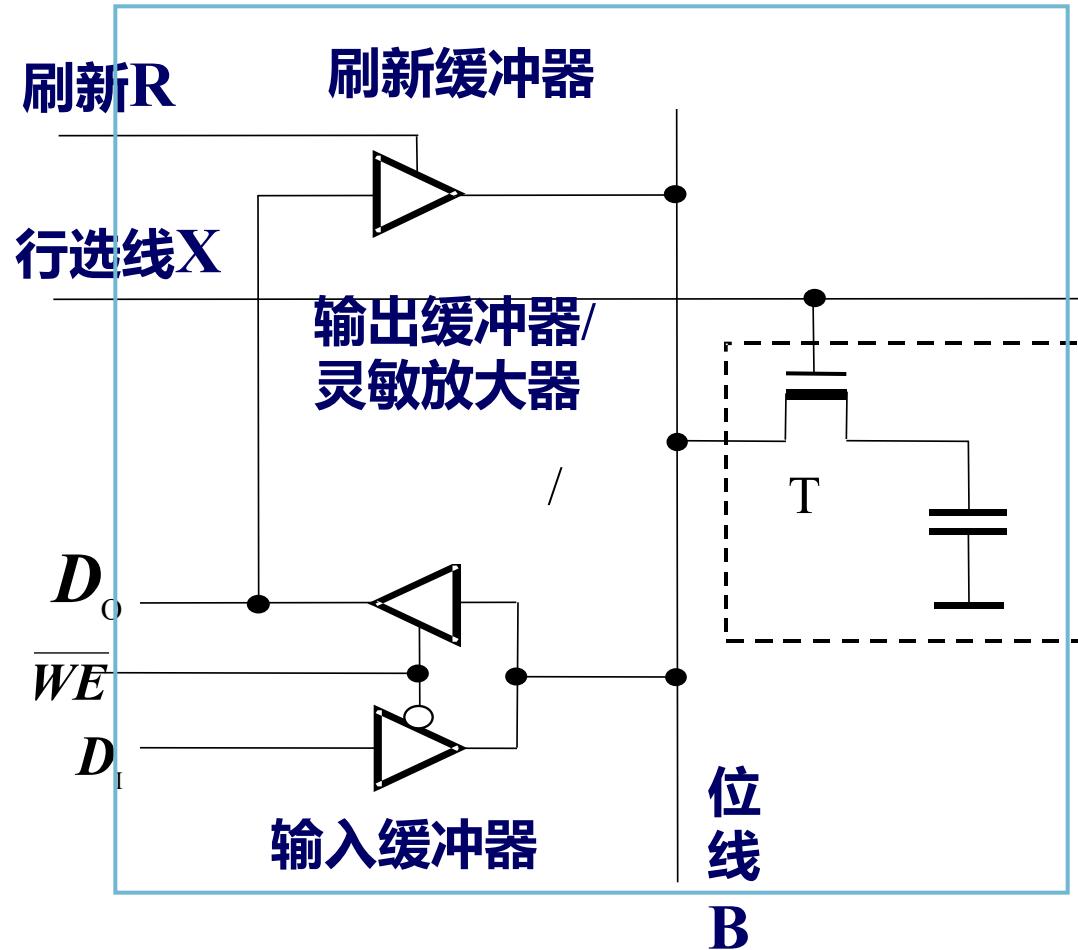


读操作: $X=1$   $\overline{WE}=1$

## T导通，电容器C与位线B连通

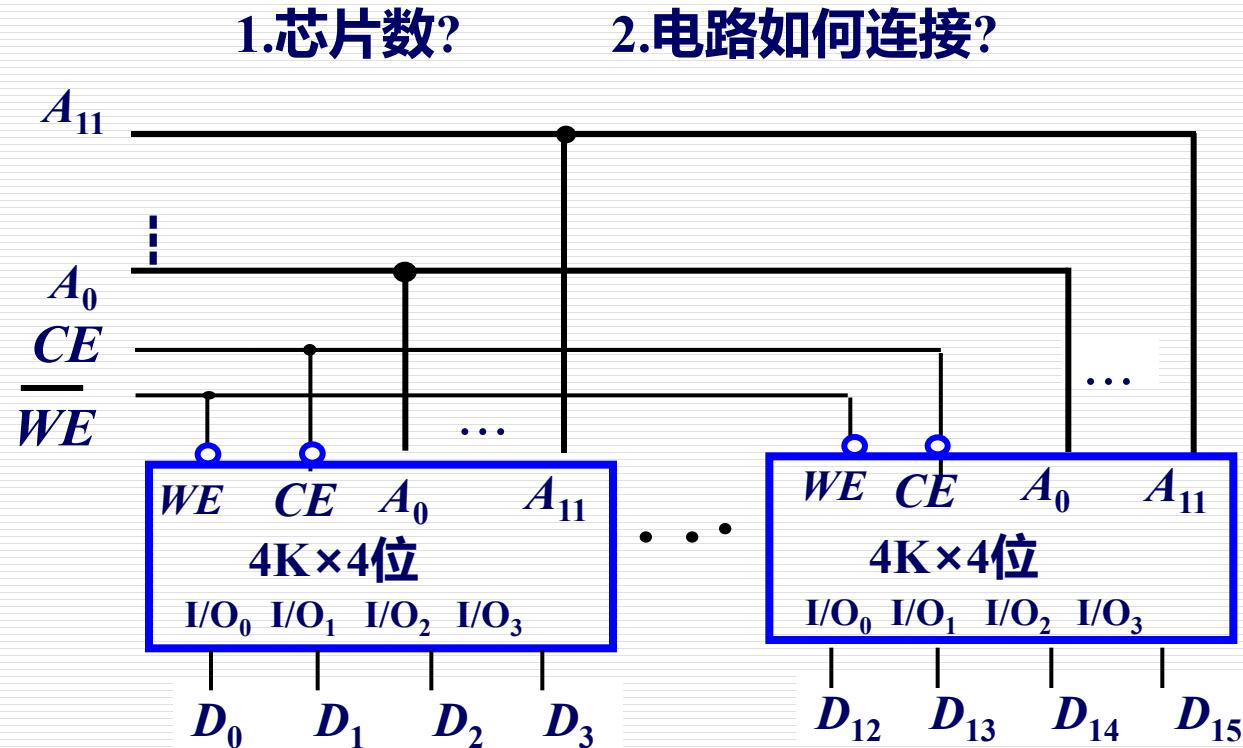
输出缓冲器/灵敏放大器被选通， $C$ 中存储的数据通过位线和缓冲器输出

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 $4K \times 4$ 位的芯片组成 $4K \times 16$ 位的存储系统。



位扩展可以利用芯片的并联方式实现。

用 $4K \times 4$ 位的芯片组成 $4K \times 16$ 位的存储系统。

## 7.2.4 RAM存储容量的扩展

### 2. 字数的扩展—用 $8K \times 8$ 位的芯片组成 $32K \times 8$ 位的存储系统。

1. 芯片数?

芯片:  $A_0 \sim A_{12}$

芯片数 = 4

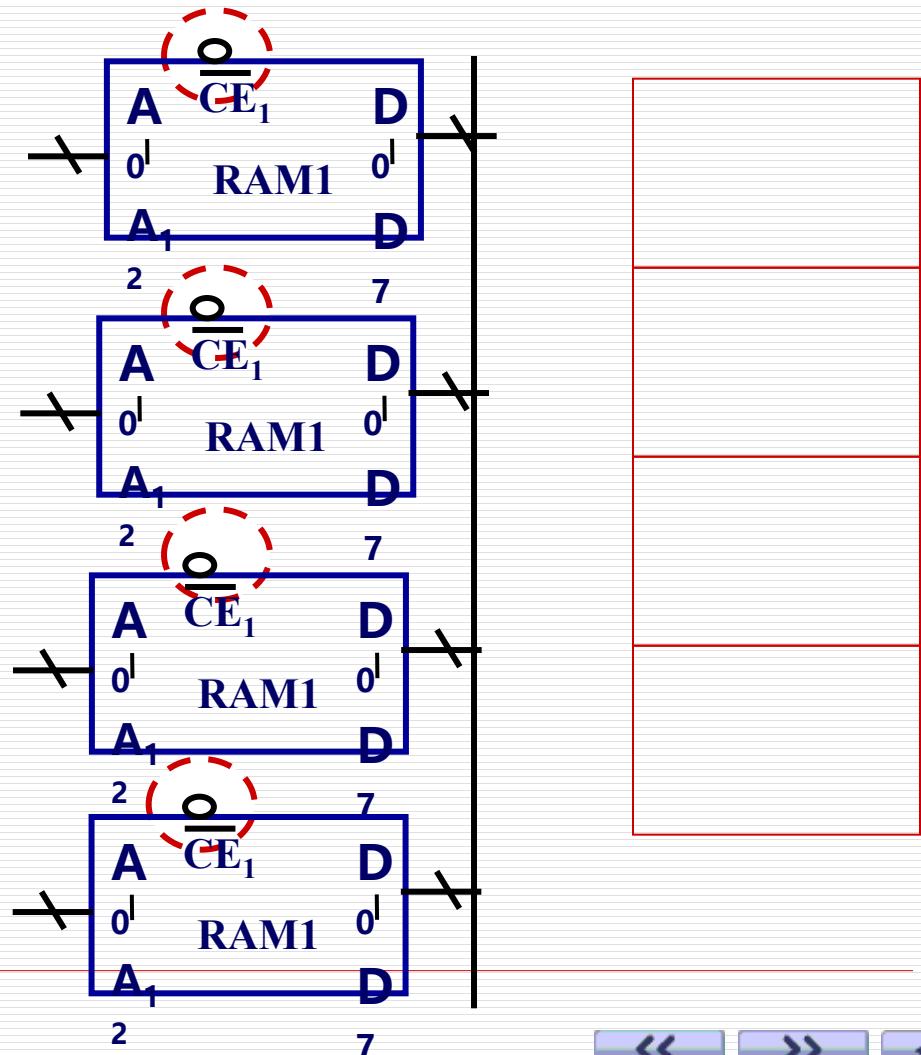
系统地址线数 = 15

系统:  $A_0 \sim A_{14}$

每块芯片的地址范围?

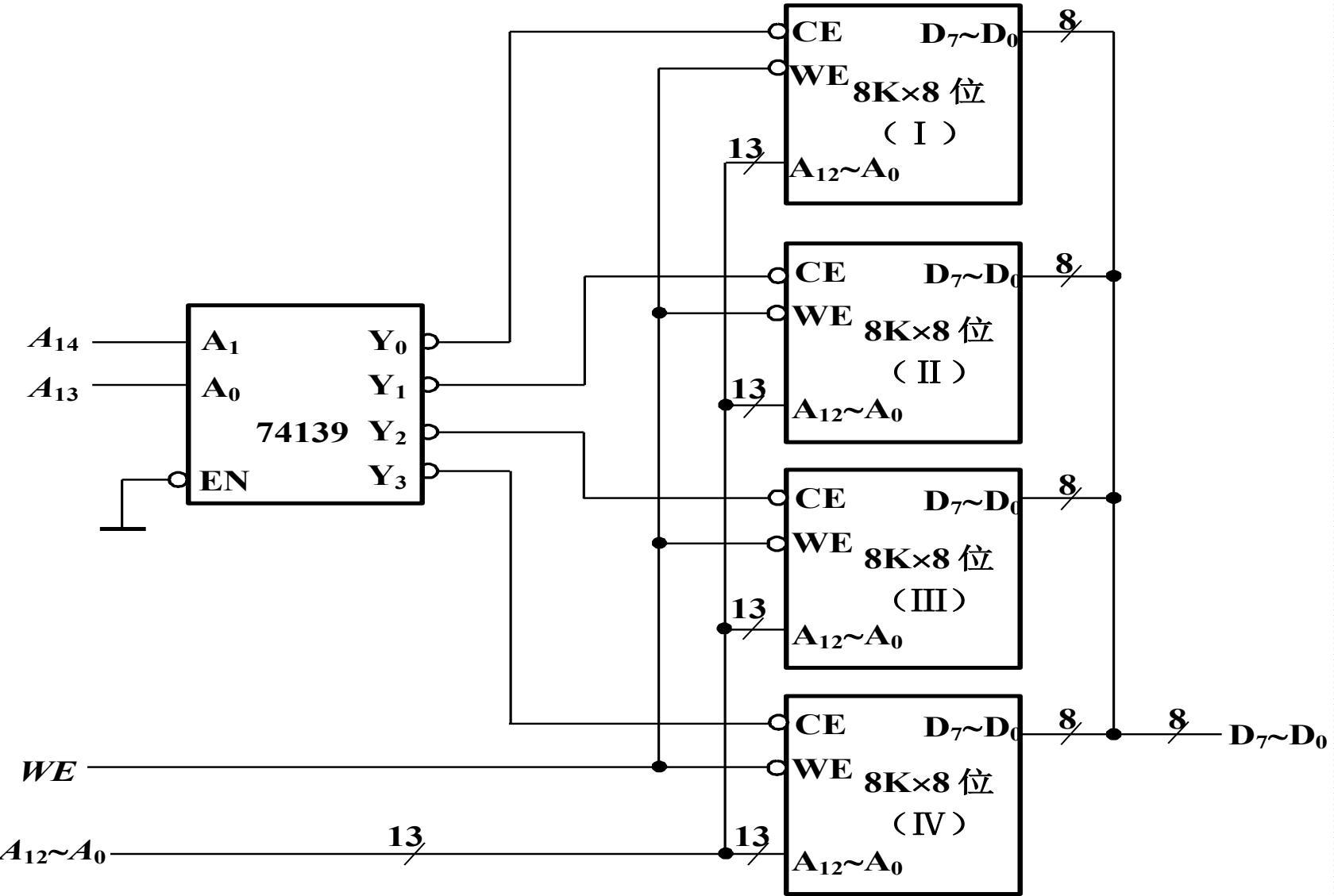
$A_{13} \sim A_{14}$  如何连接?

2. 电路如何连接?



# 32K×8位存储器系统的地址分配表

| 各<br>RAM<br>芯片 | 译码器<br>有效输出端   | 扩展的地<br>址输入端<br>A <sub>14</sub> A <sub>13</sub> | 8K×8位RAM芯片地址输入端 |                 |                 |                |                |                |                |                |                |                |                |                | 对应的十<br>六进制地<br>址码 |       |
|----------------|----------------|---|-----------------|-----------------|-----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|--------------------|-------|
|                |                |   | A <sub>12</sub> | A <sub>11</sub> | A <sub>10</sub> | A <sub>9</sub> | A <sub>8</sub> | A <sub>7</sub> | A <sub>6</sub> | A <sub>5</sub> | A <sub>4</sub> | A <sub>3</sub> | A <sub>2</sub> | A <sub>1</sub> | A <sub>0</sub>     |       |
| I              | Y <sub>0</sub> | 0 0   | 0               | 0               | 0               | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0                  | 0000H |
|                |                |   | 0               | 0               | 0               | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0                  | 0001H |
|                |                |   | 0               | 0               | 0               | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 1              | 0                  | 0002H |
|                |                |   | 1               | 1               | 1               | 1              | 1              | 1              | 1              | 1              | 1              | 1              | 1              | 1              | 1                  | 1FFFH |
| II             | Y <sub>1</sub> | 0 1   | 0               | 0               | 0               | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0                  | 2000H |
|                |                |   | 0               | 0               | 0               | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 1                  | 2001H |
|                |                |   | 0               | 0               | 0               | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 1              | 0                  | 2002H |
|                |                |   | 1               | 1               | 1               | 1              | 1              | 1              | 1              | 1              | 1              | 1              | 1              | 1              | 1                  | 3FFFH |
| III            | Y <sub>2</sub> | 1 0   | 0               | 0               | 0               | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0                  | 4000H |
|                |                |   | 0               | 0               | 0               | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 1                  | 400H  |
|                |                |   | 0               | 0               | 0               | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 1              | 0                  | 4002H |
|                |                |   | 1               | 1               | 1               | 1              | 1              | 1              | 1              | 1              | 1              | 1              | 1              | 1              | 1                  | 5FFFH |
| IV             | Y <sub>3</sub> | 1 1   | 0               | 0               | 0               | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0                  | 6000H |
|                |                |   | 0               | 0               | 0               | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 1                  | 6001H |
|                |                |   | 0               | 0               | 0               | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 1              | 0                  | 6002H |
|                |                |   | 1               | 1               | 1               | 1              | 1              | 1              | 1              | 1              | 1              | 1              | 1              | 1              | 1                  | 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

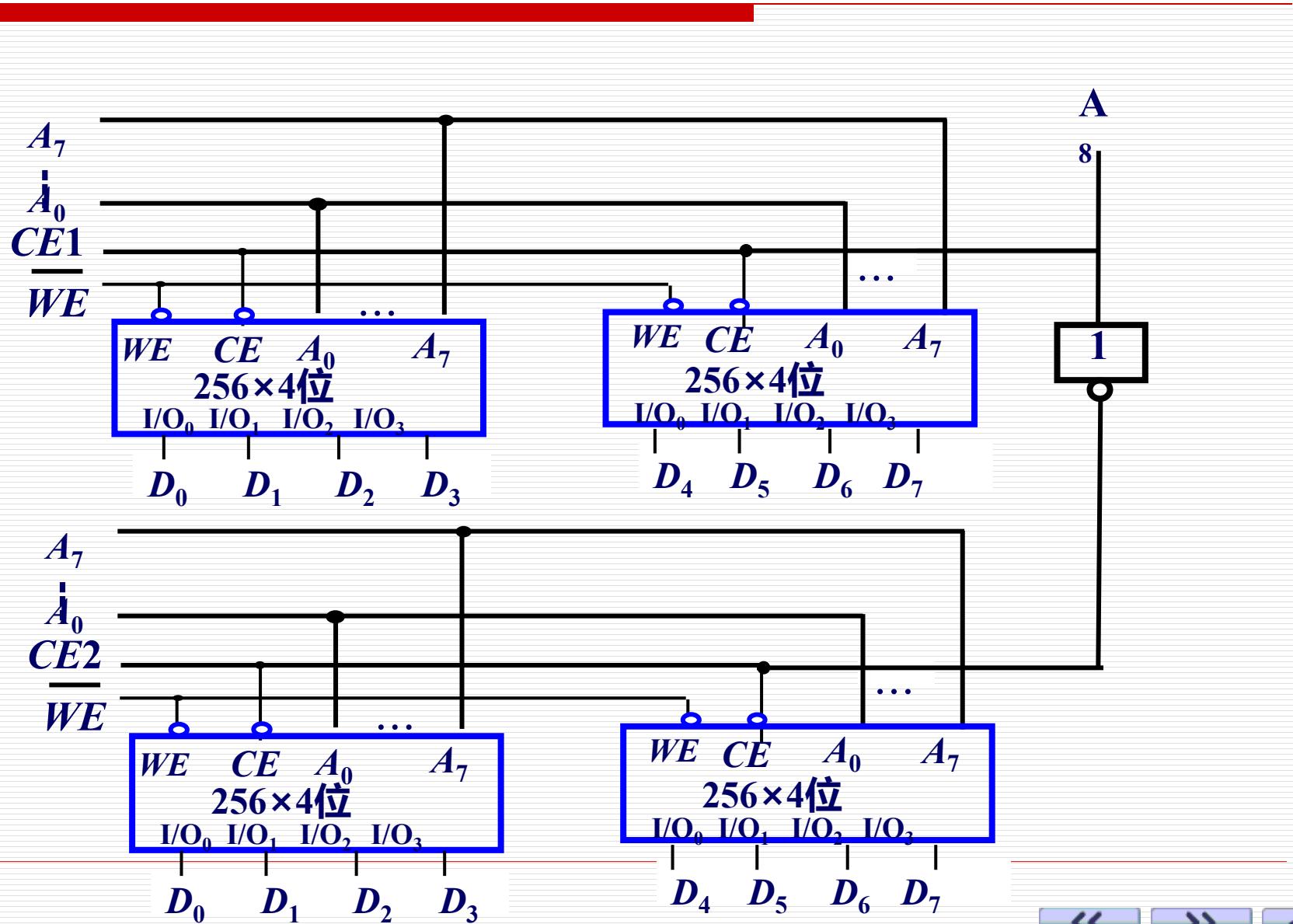
---

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。



□ 思考：如何用一个具有片选使能、输出使能、读写控制、容量为 $4K \times 4$ 位的SRAM芯片和必要的逻辑门，扩展成为 $8K \times 8$ 位的存储器系统？试画出逻辑图。

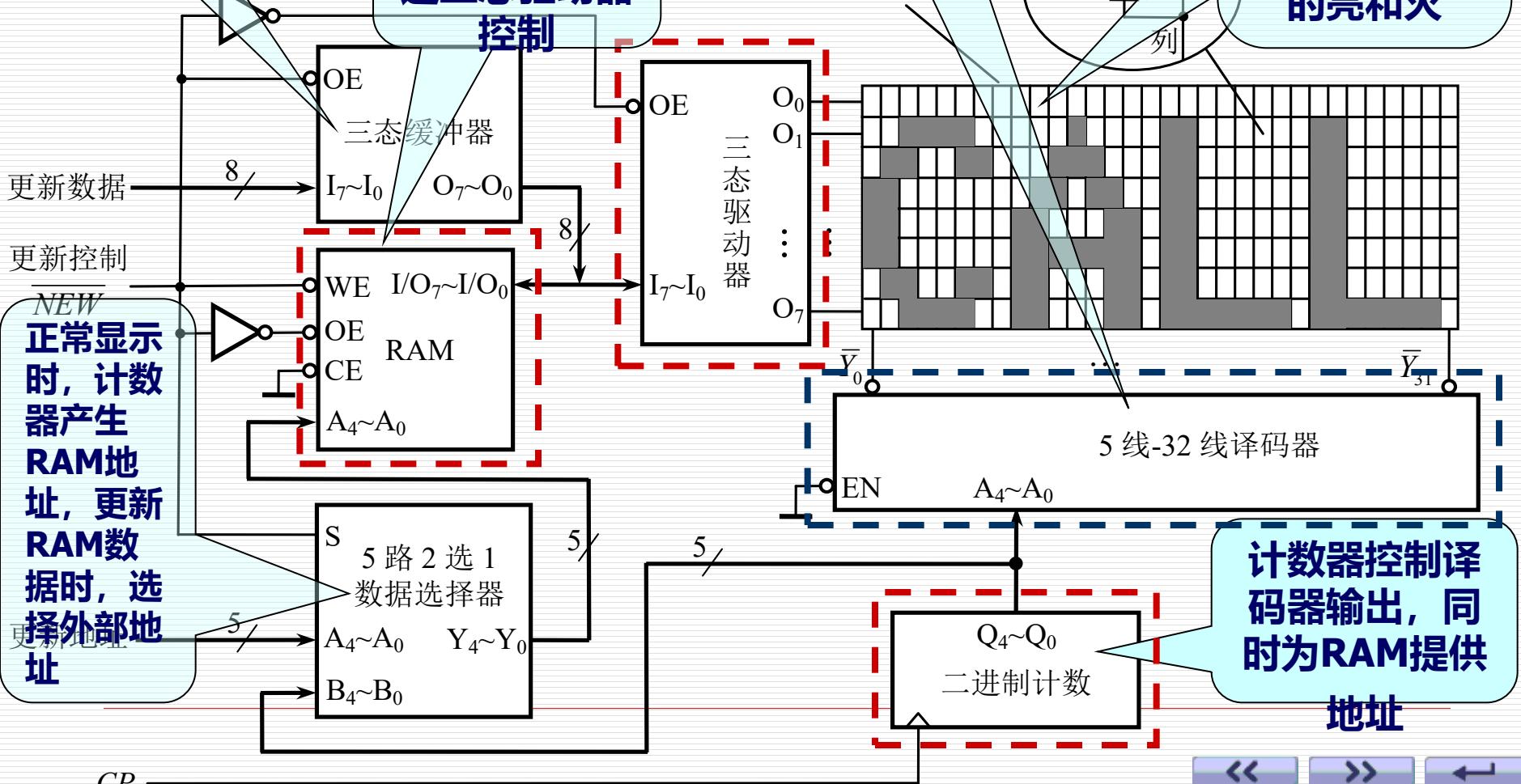
## 7.2.5 RAM应用举例—LED点阵显示屏

## 内容更新时 的数据输入

行线的电平由  
RAM读出经  
过三态驱动器  
控制

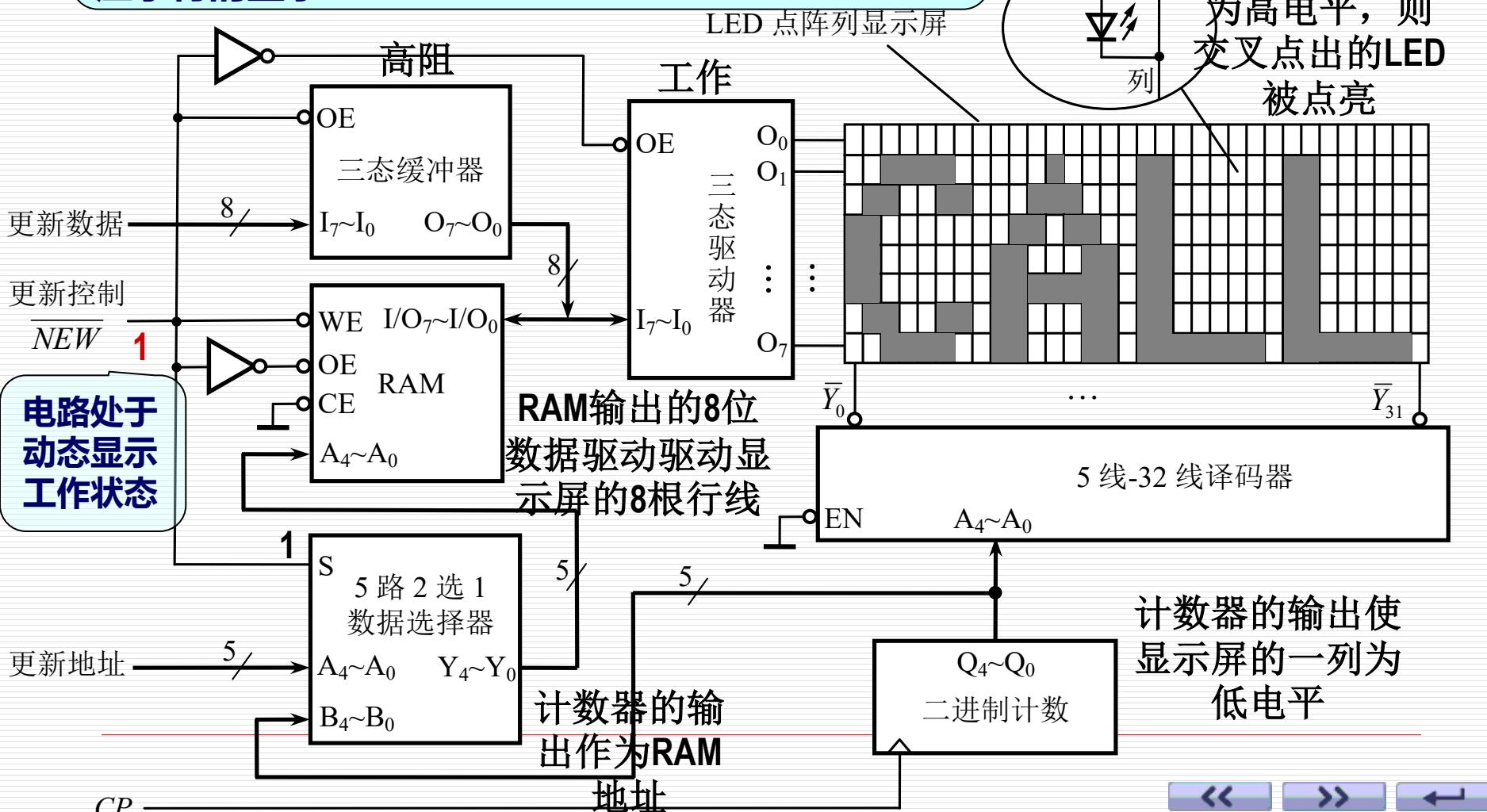
列线的电平  
由5-32线译  
码器控制

**F** 显示屏  
由8\*32个  
LED构成，行  
线和列线的电  
平控制交叉点  
的亮和灭



## 7.2.5 RAM应用举例—LED点阵显示屏

因此，通过将要显示的字符以点阵形式存储在RAM中（需要发光的点存1，否则存0），则可以实现相应字符的显示



## 7.2.5 RAM应用举例—LED点阵显示屏

