

第4章 存储器

4.1 概述

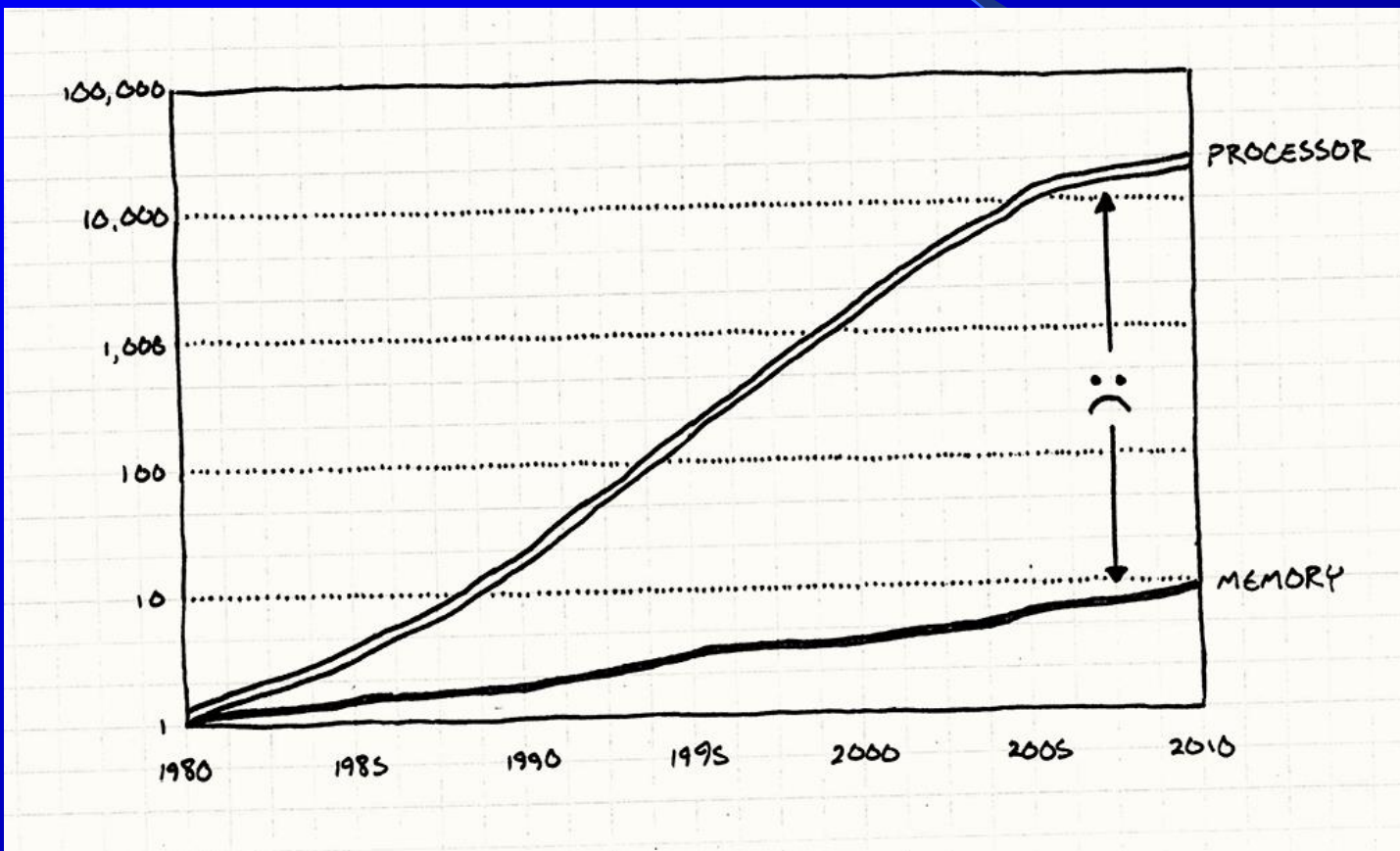
4.2 主存储器

4.3 高速缓冲存储器

4.4 辅助存储器



4.1 概述



一、存储器分类

1. 按存储介质分类

(1) 半导体存储器	TTL、MOS	易失
(2) 磁表面存储器	磁头、载磁体	非易失
(3) 磁芯存储器	硬磁材料、环状元件	
(4) 光盘存储器	激光、磁光材料	

2. 按存取方式分类

(1) 存取时间与物理地址无关（随机访问）

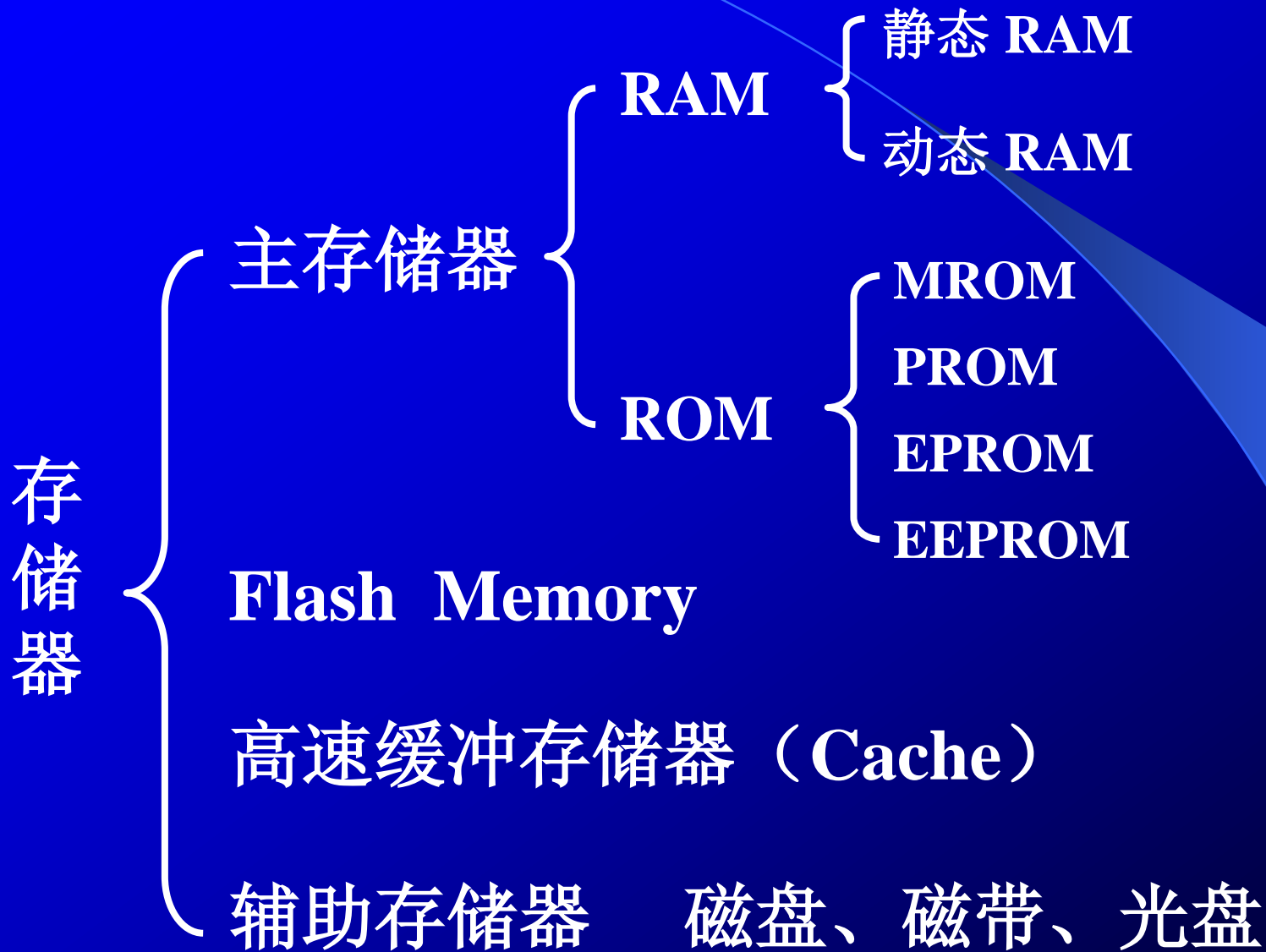
- 随机存储器 在程序的执行过程中 可读 可 写
- 只读存储器 在程序的执行过程中 只 读

(2) 存取时间与物理地址有关（串行访问）

- 顺序存取存储器 磁带
- 直接存取存储器 磁盘

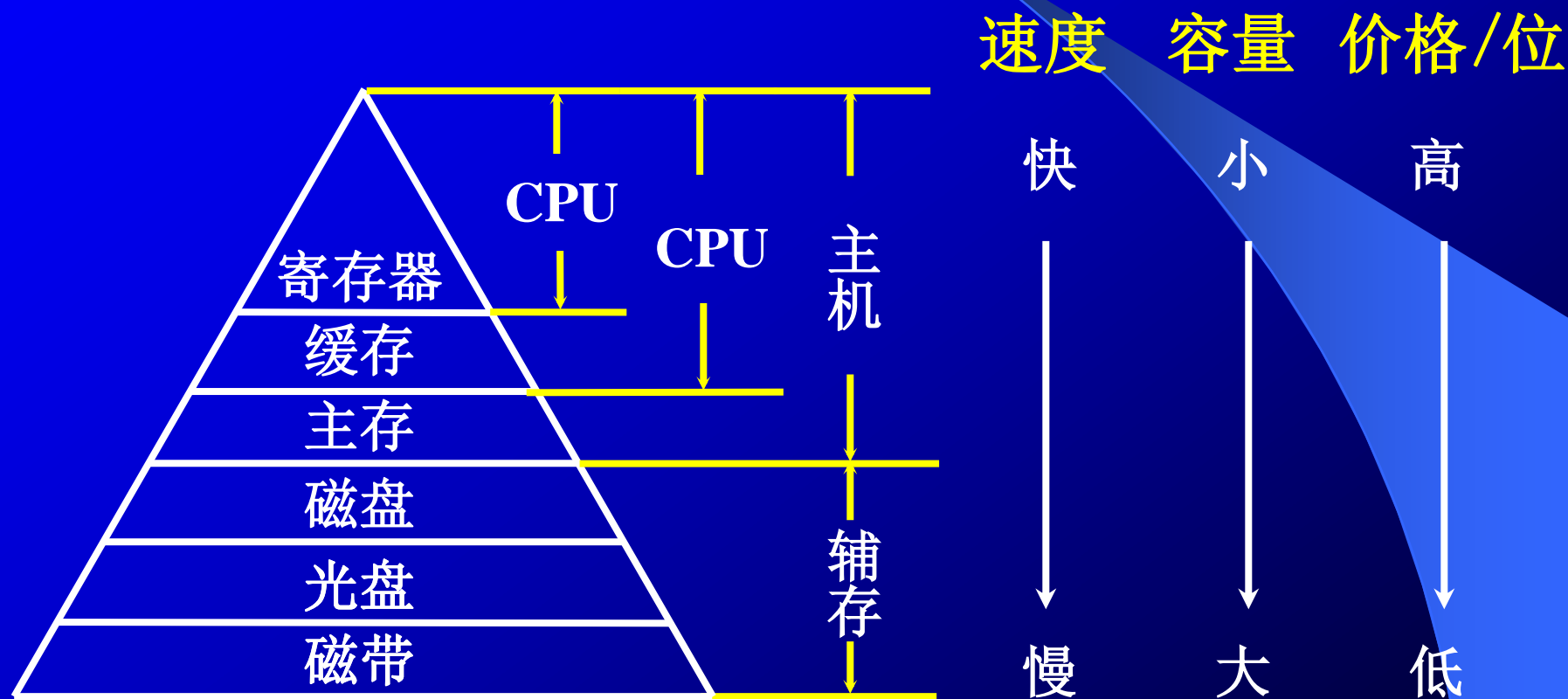
3. 按在计算机中的作用分类

4.1

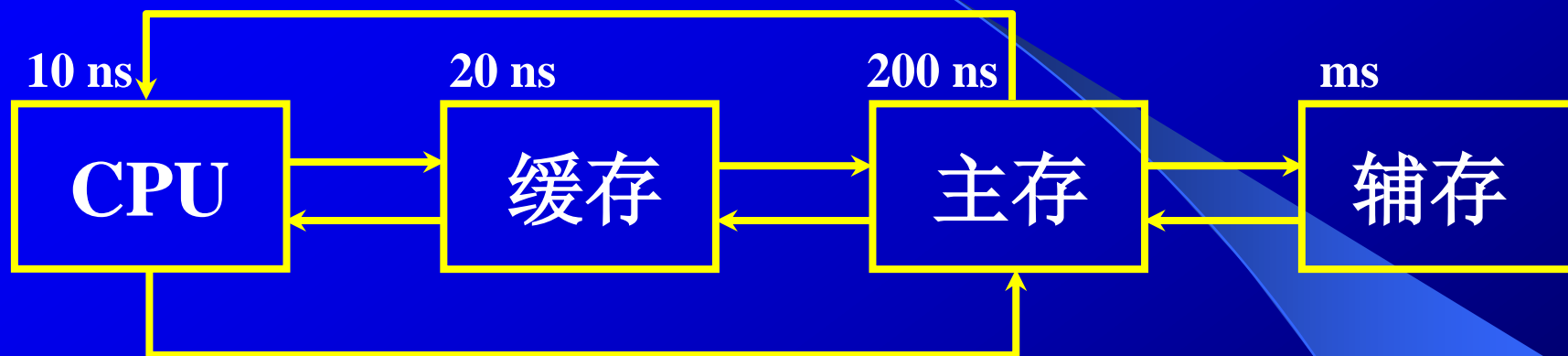


二、存储器的层次结构

1. 存储器三个主要特性的关系



2. 缓存—主存层次和主存—辅存层次 4.1



(速度) (容量)
缓存—主存 主存—辅存

主存储器

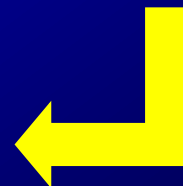
虚拟存储器

实地址

虚地址

物理地址

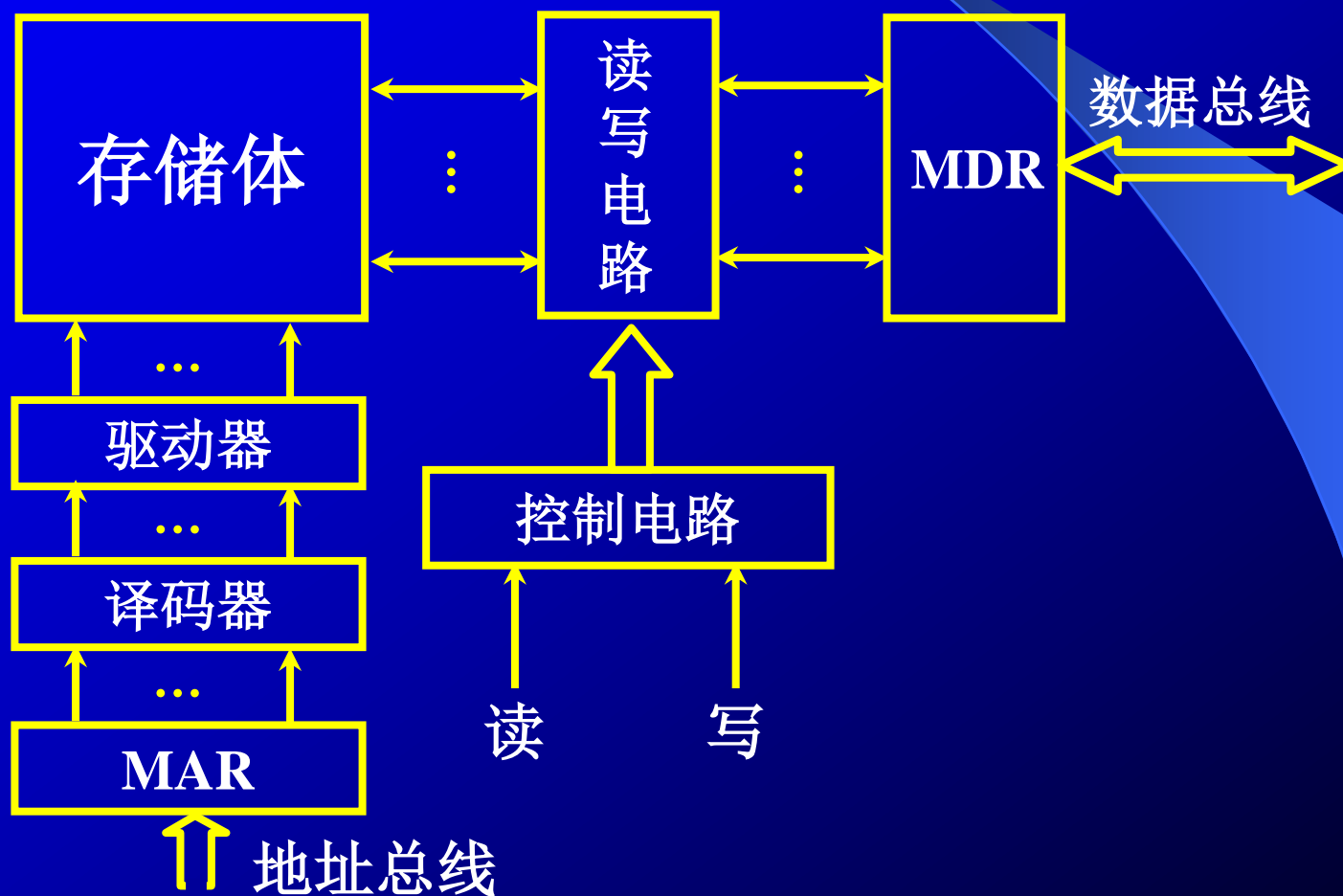
逻辑地址



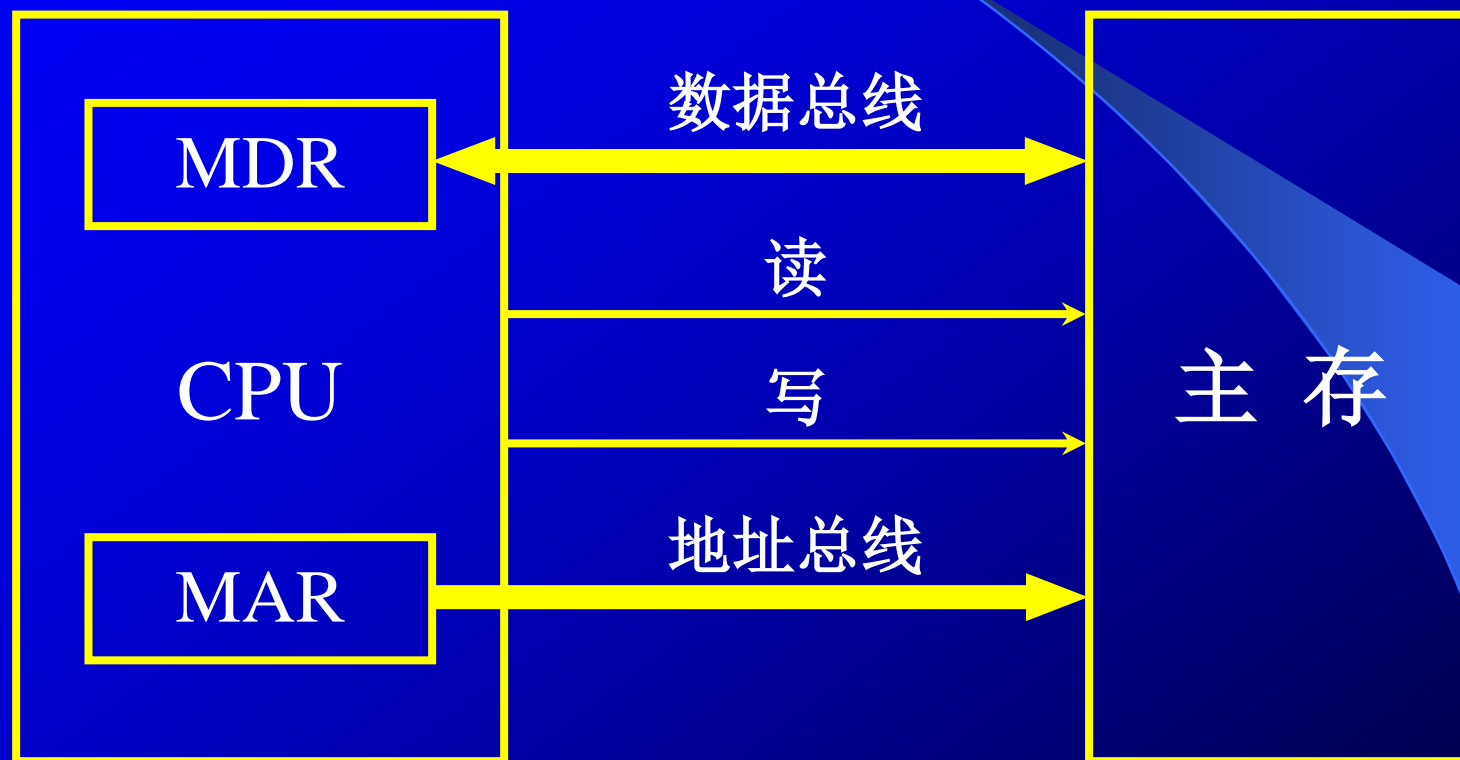
4.2 主存储器

一、概述

1. 主存的基本组成



2. 主存和 CPU 的联系



3. 主存中存储单元地址的分配

高位字节 地址为字地址

低位字节 地址为字地址

字地址	字节地址			
0	0	1	2	3
4	4	5	6	7
8	8	9	10	11

字地址	字节地址	
0	1	0
2	3	2
4	5	4

设地址线 24 根

若字长为 16 位

若字长为 32 位

按 字节 寻址 $2^{24} = 16 \text{ M}$

按 字 寻址 8 M

按 字 寻址 4 M

4. 主存的技术指标

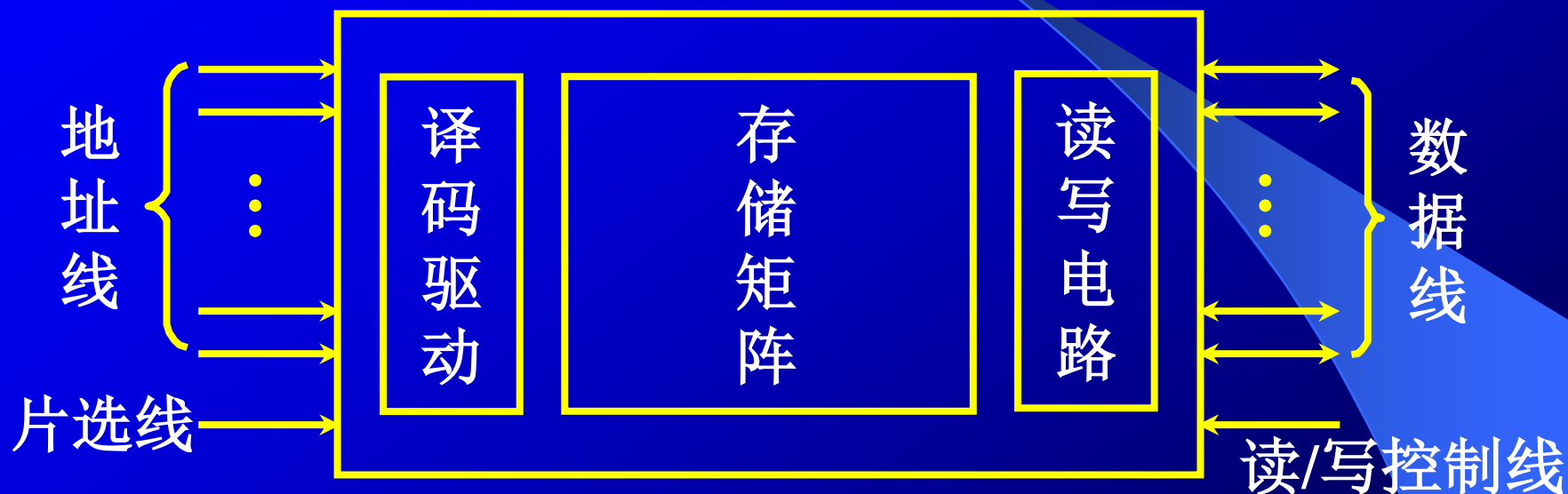
4.2

- (1) 存储容量 主存 存放二进制代码的总位数
- (2) 存储速度
 - 存取时间 存储器的 访问时间
读出时间 写入时间
 - 存取周期 连续两次独立的存储器操作
(读或写) 所需的 最小间隔时间
读周期 写周期
- (3) 存储器的带宽 位/秒



二、半导体存储芯片简介

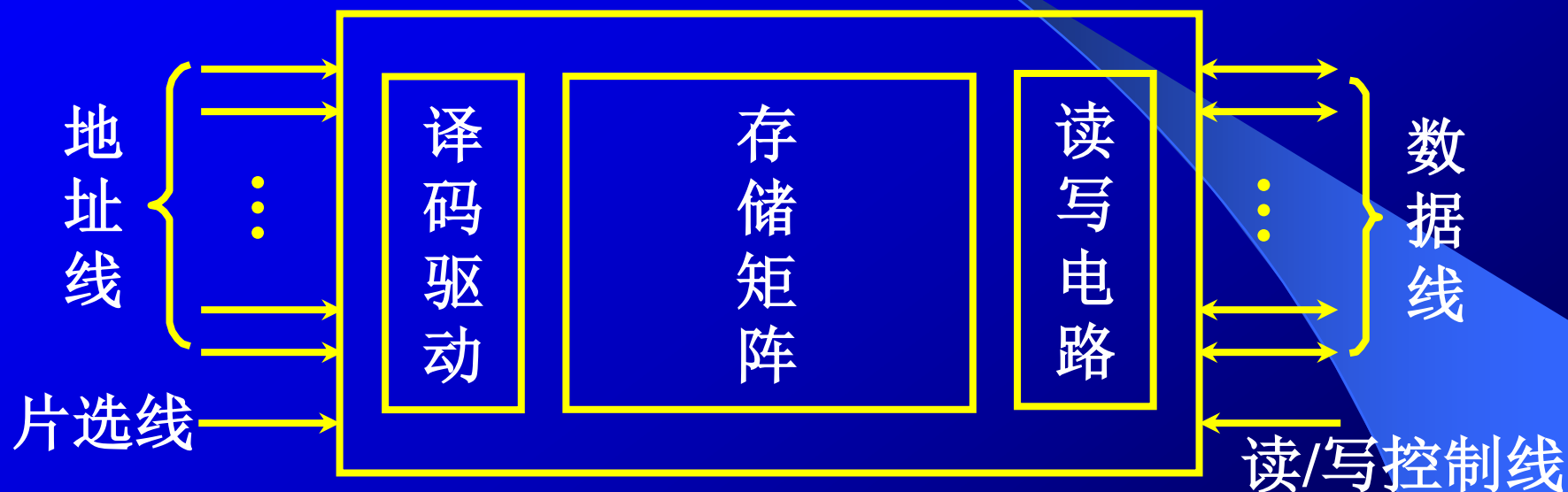
1. 半导体存储芯片的基本结构



地址线（单向）	数据线（双向）	芯片容量
10	4	1K×4位
14	1	16K×1位
13	8	8K×8位

二、半导体存储芯片简介

1. 半导体存储芯片的基本结构

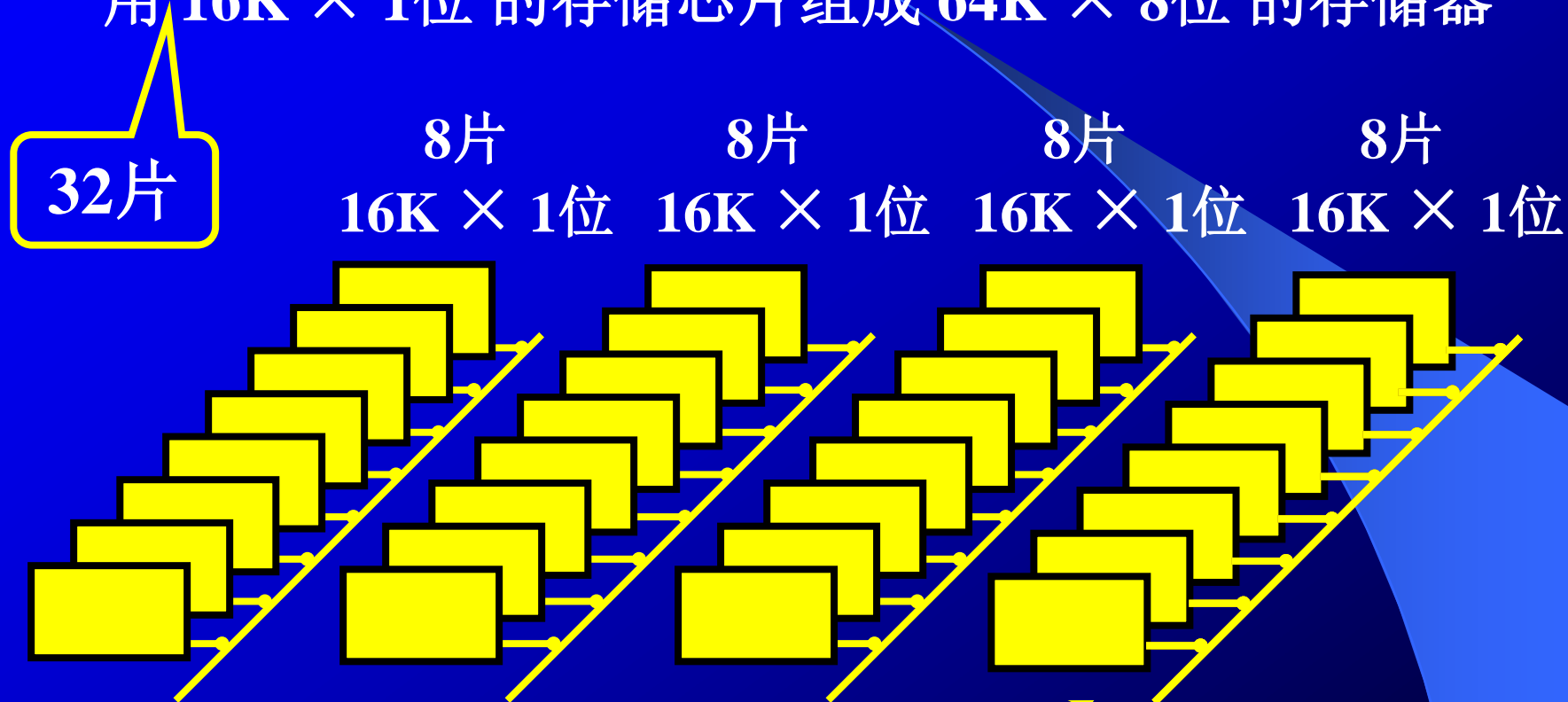


片选线 $\overline{\text{CS}}$ $\overline{\text{CE}}$

读/写控制线 $\overline{\text{WE}}$ (低电平写 高电平读)
 $\overline{\text{OE}}$ (允许读) $\overline{\text{WE}}$ (允许写)

存储芯片片选线的作用

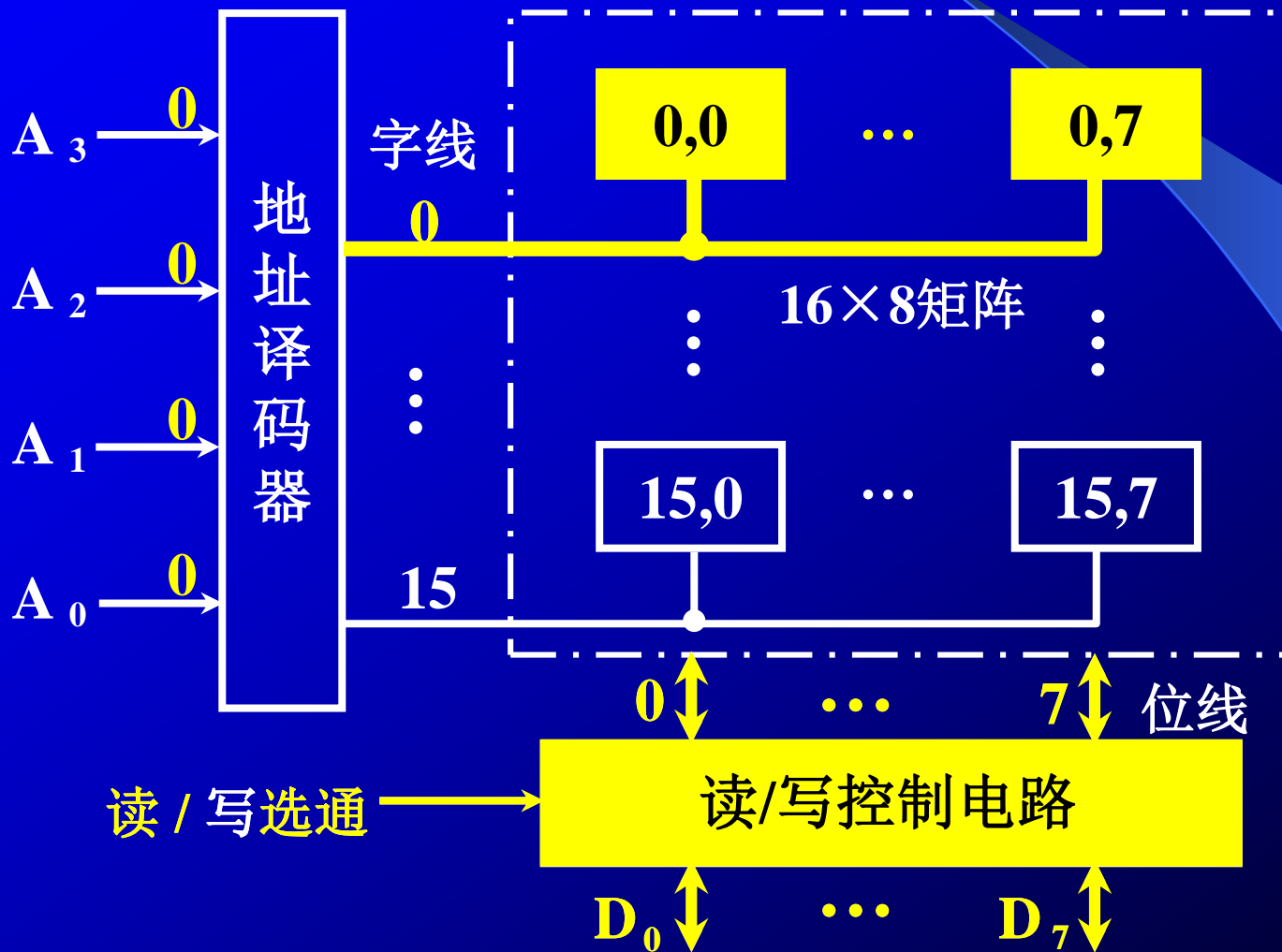
用 $16\text{K} \times 1$ 位的存储芯片组成 $64\text{K} \times 8$ 位的存储器



当地址为 65 535 (**16位**) 时，此 8 片的片选有效

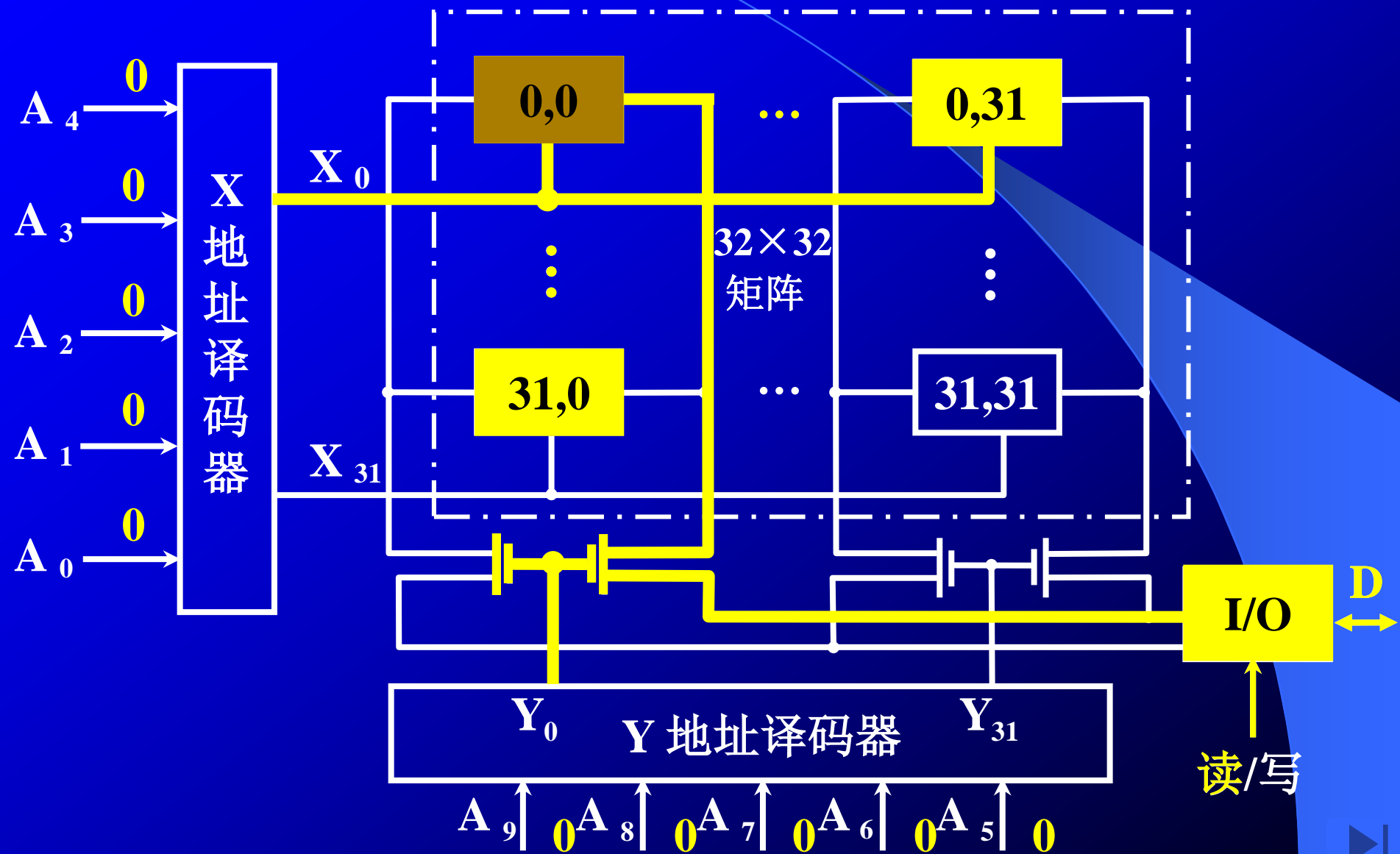
2. 半导体存储芯片的译码驱动方式 4.2

(1) 线选法



(2) 重合法

4.2

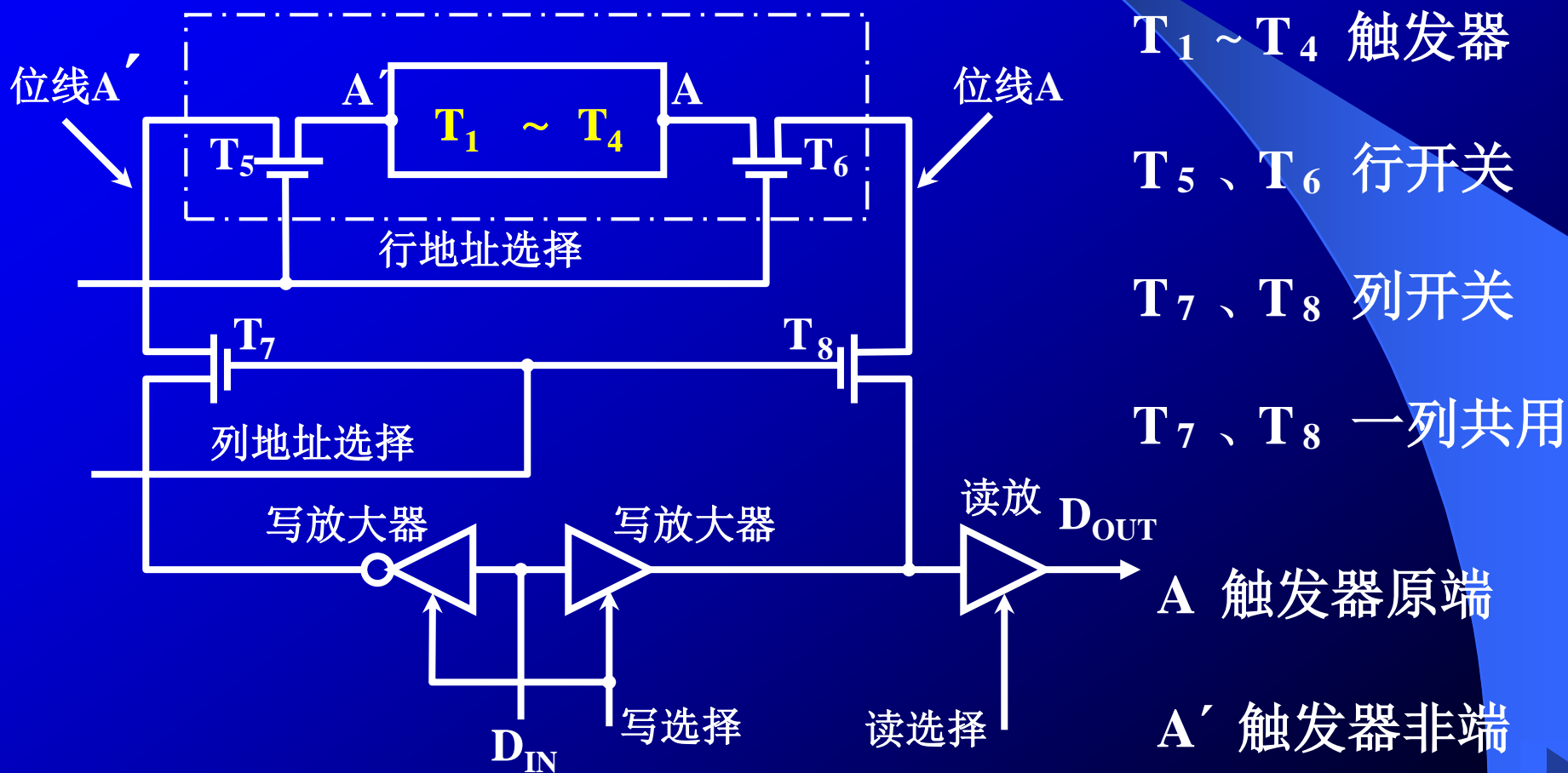


三、随机存取存储器 (RAM)

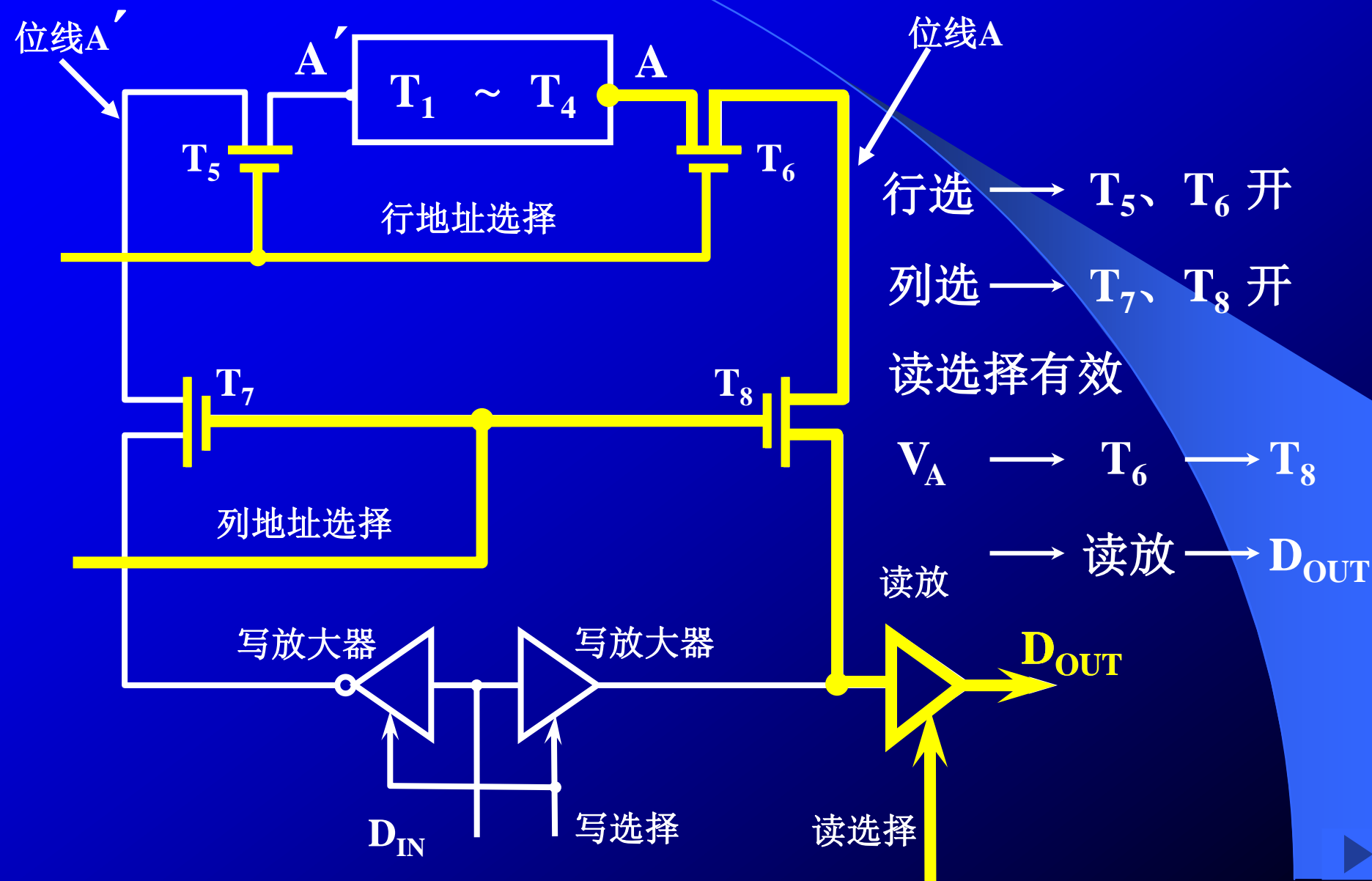
4.2

1. 静态 RAM (SRAM)

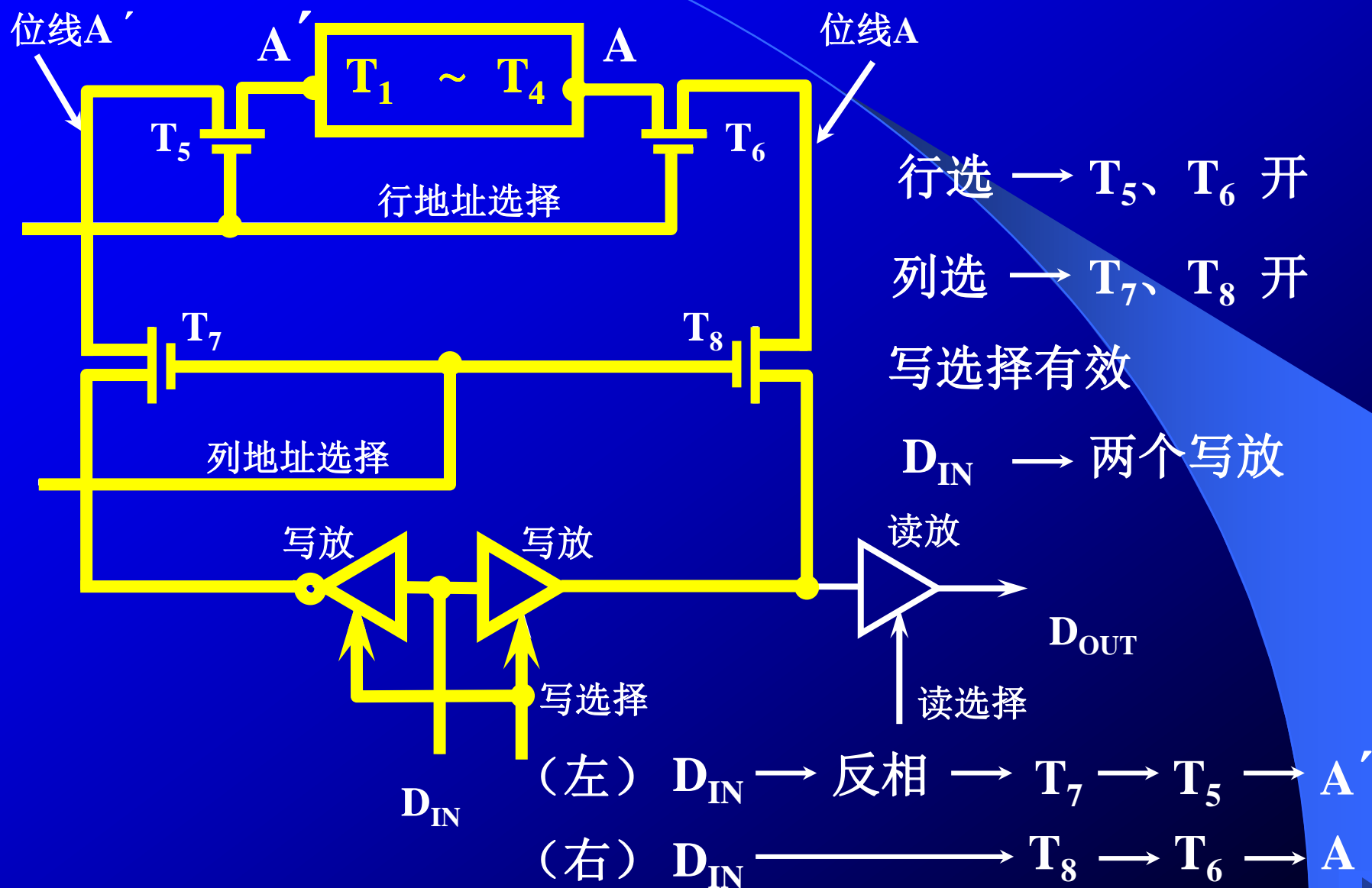
(1) 静态 RAM 基本单元电路



① 静态 RAM 基本电路的 读 操作

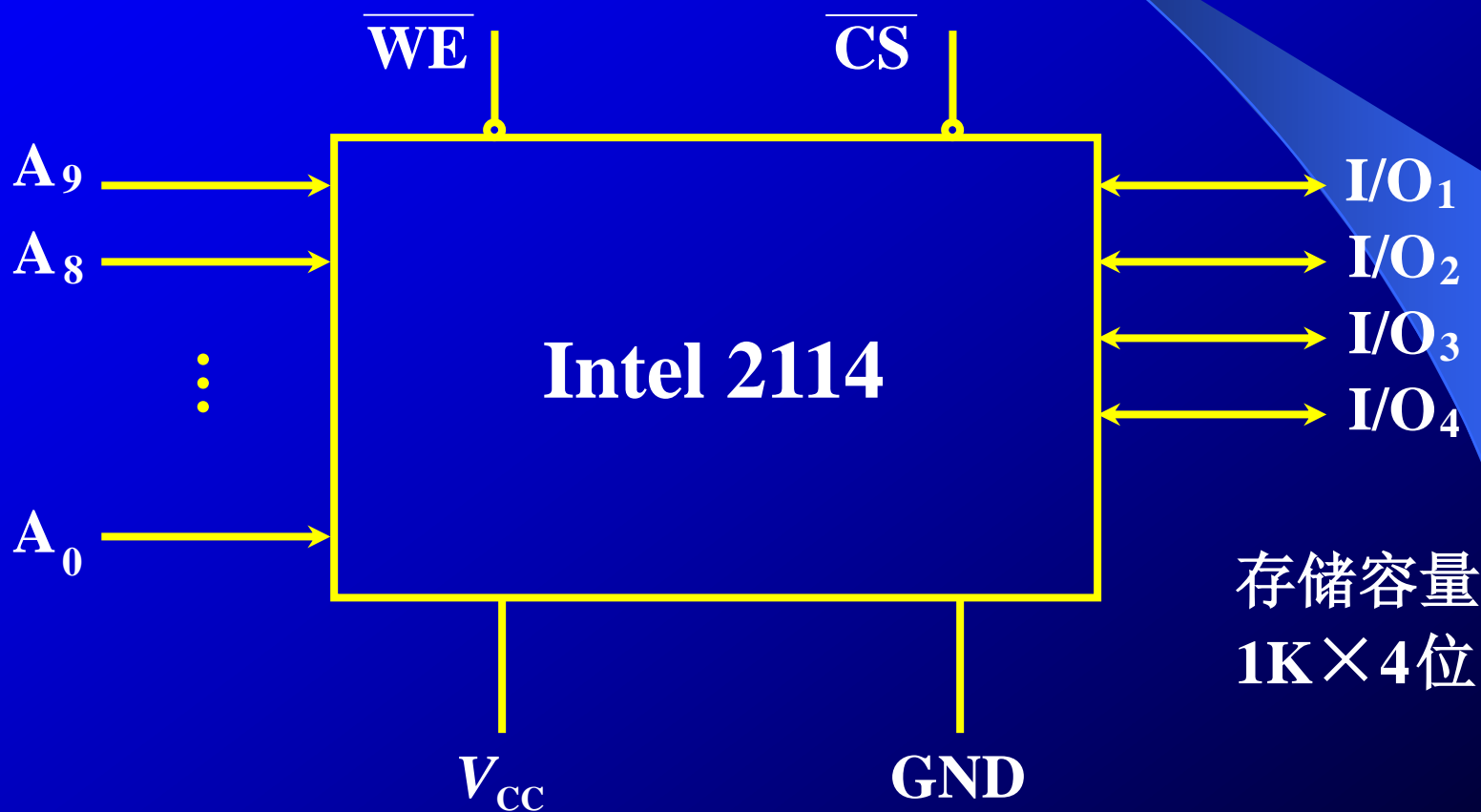


② 静态 RAM 基本电路的 写 操作

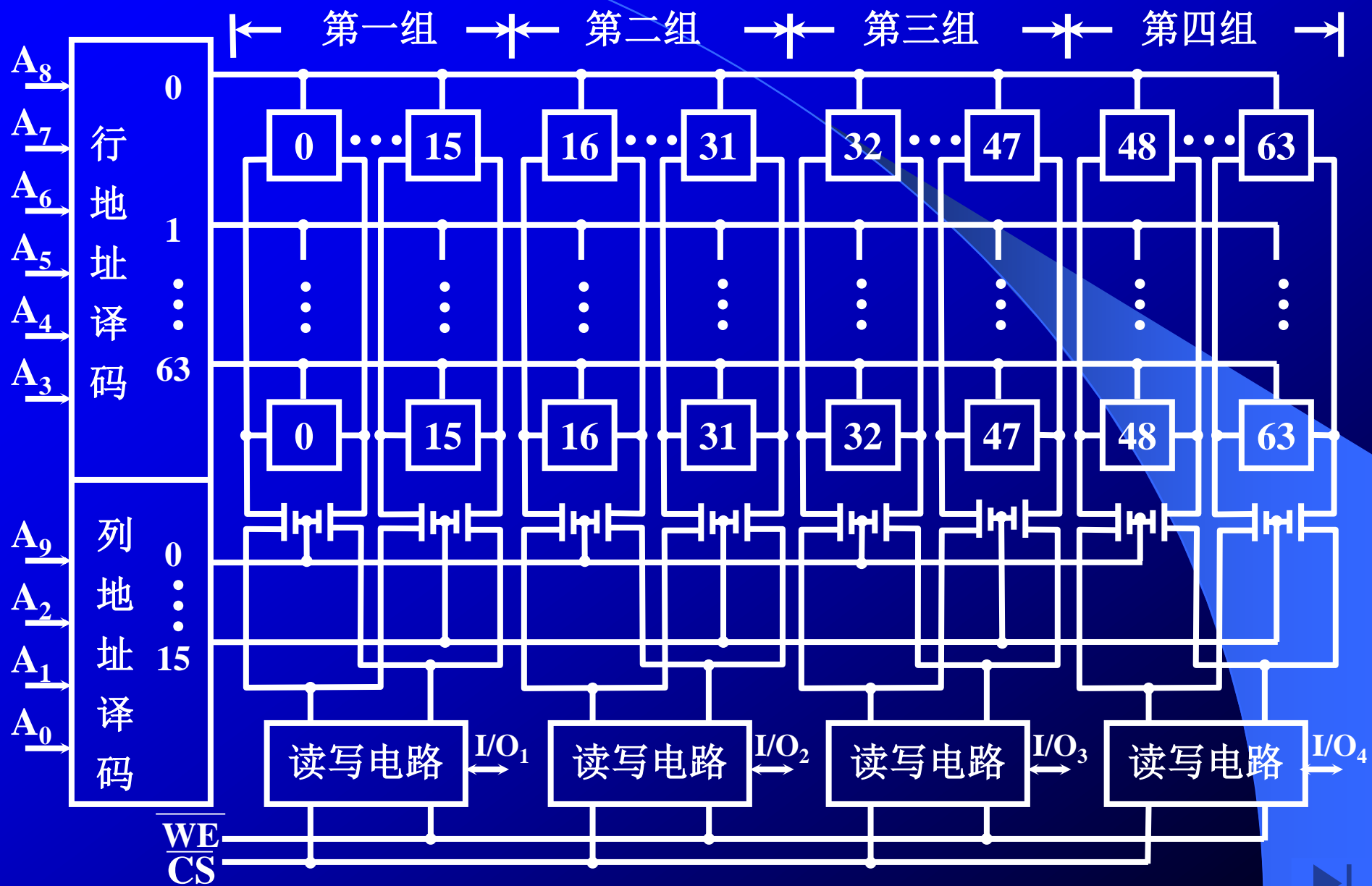


(2) 静态 RAM 芯片举例

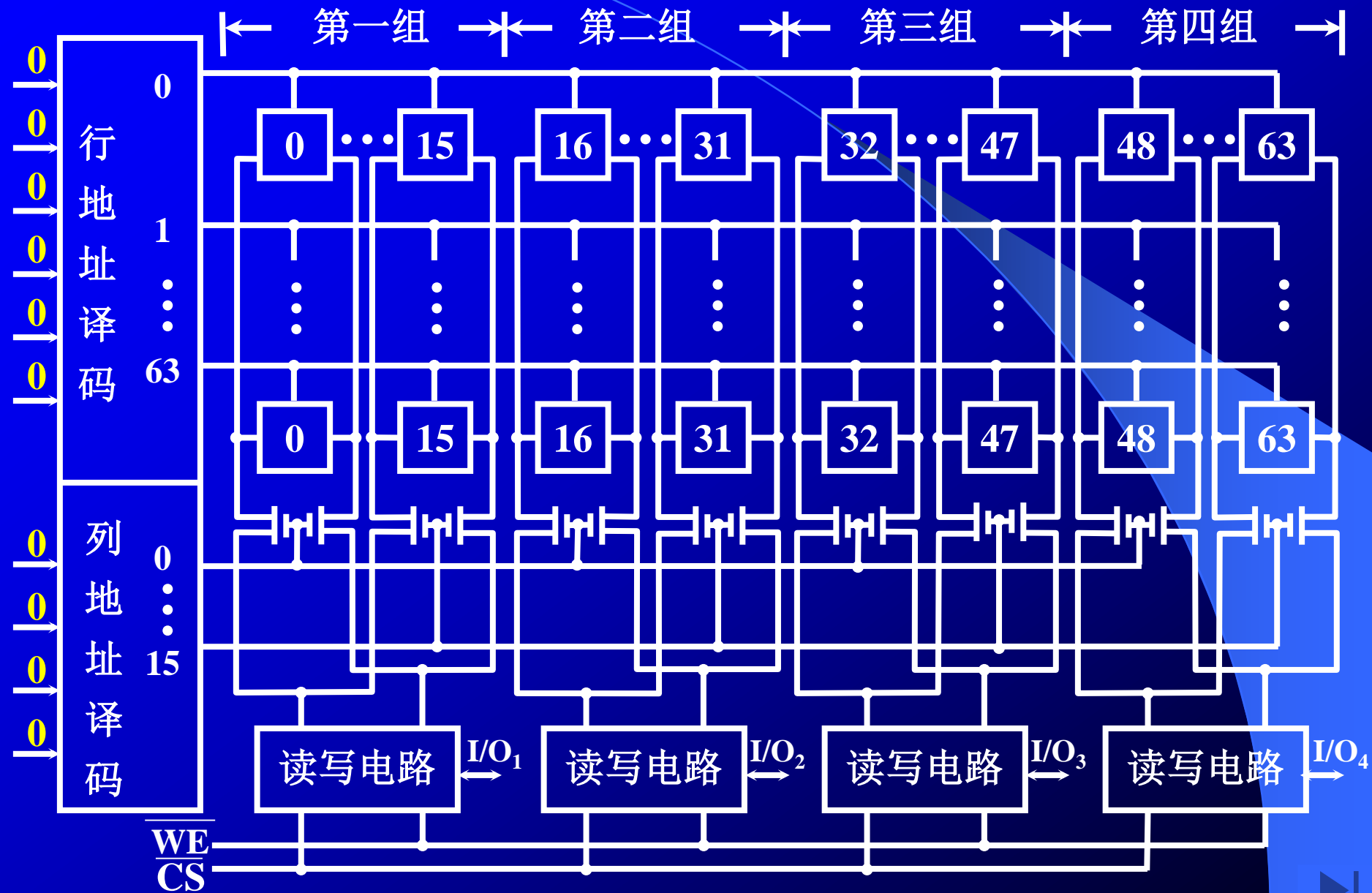
① Intel 2114 外特性



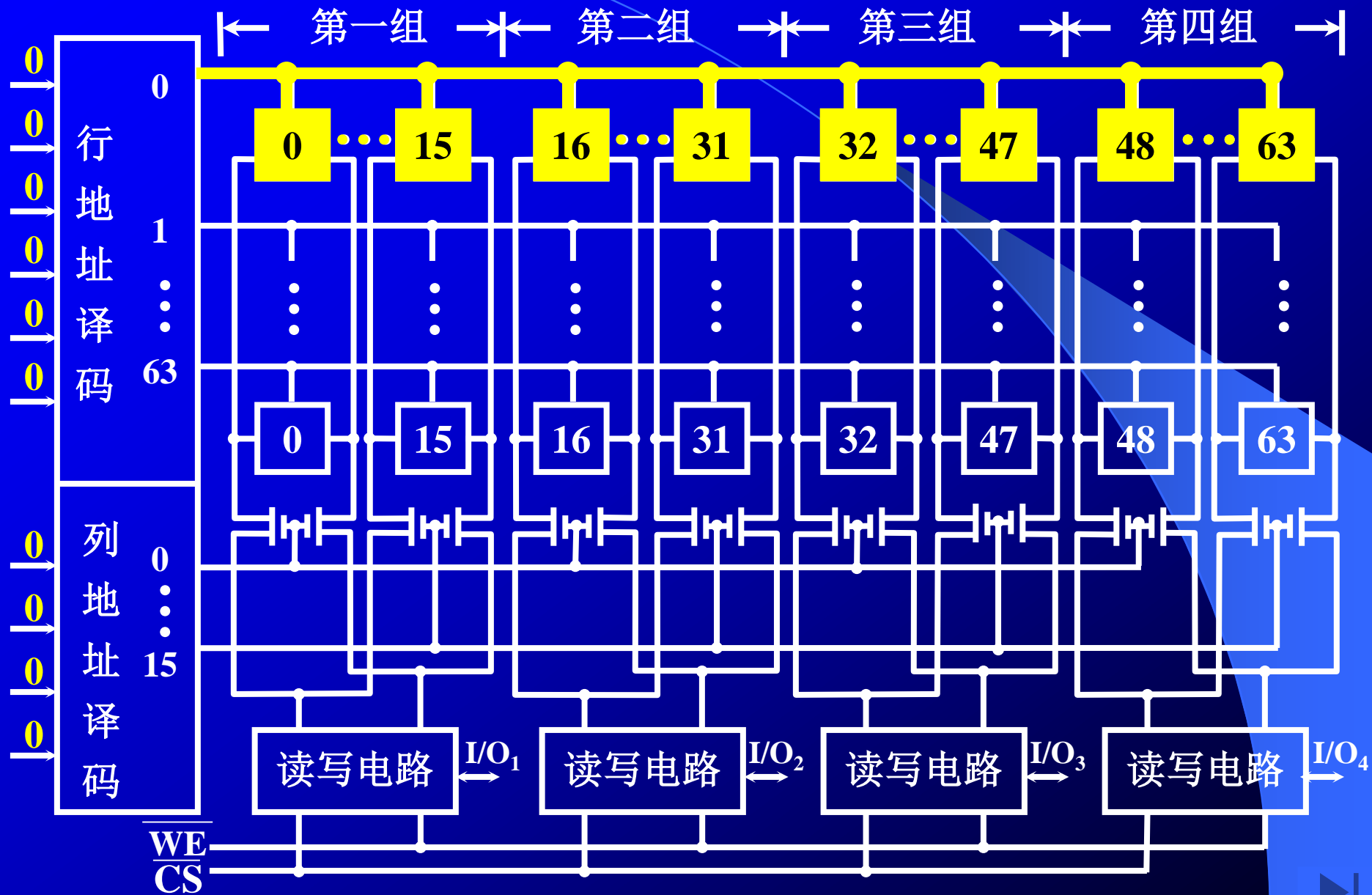
② Intel 2114 RAM 矩阵 (64×64) 读 4.2



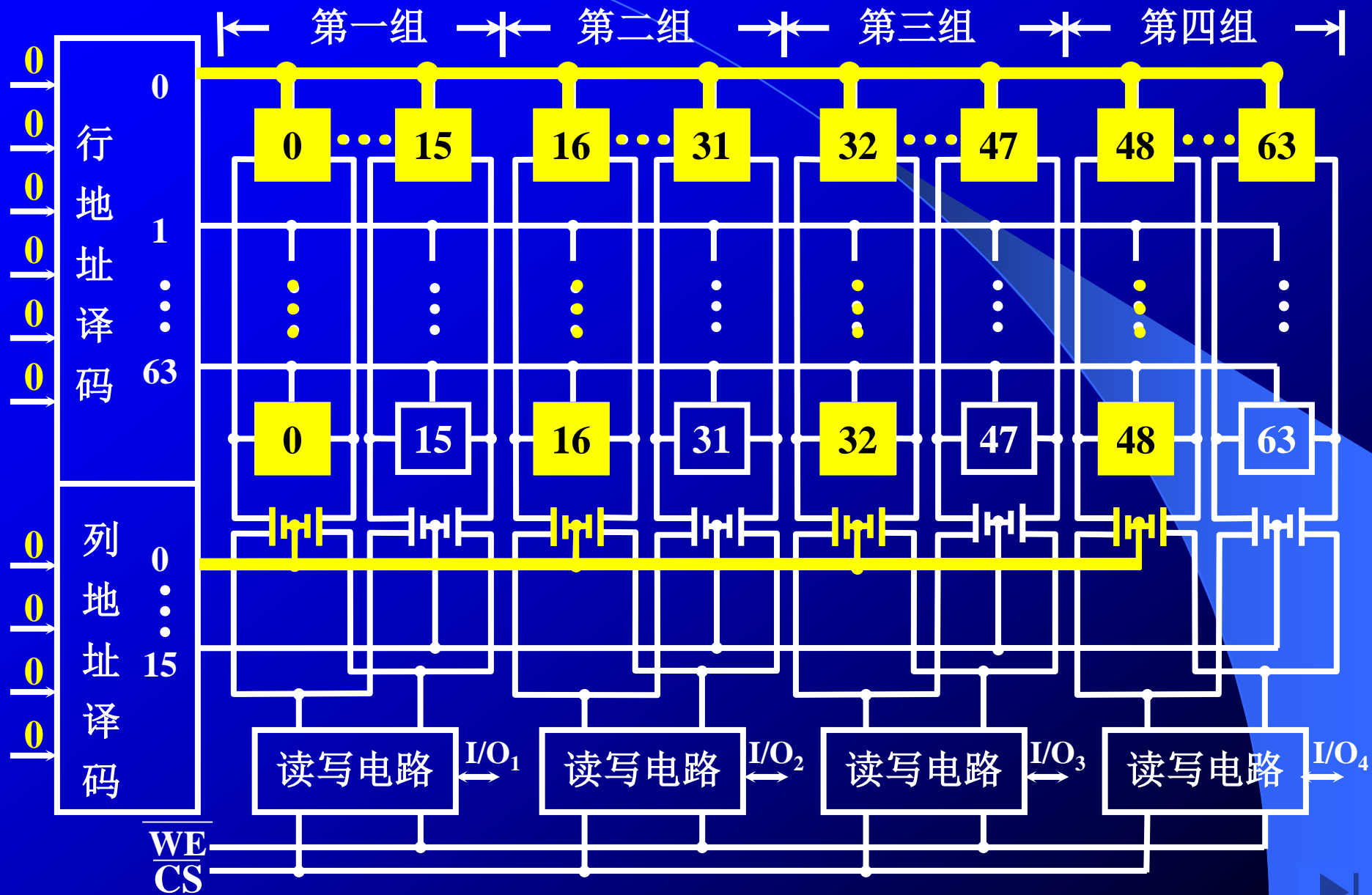
② Intel 2114 RAM 矩阵 (64×64) 读 4.2



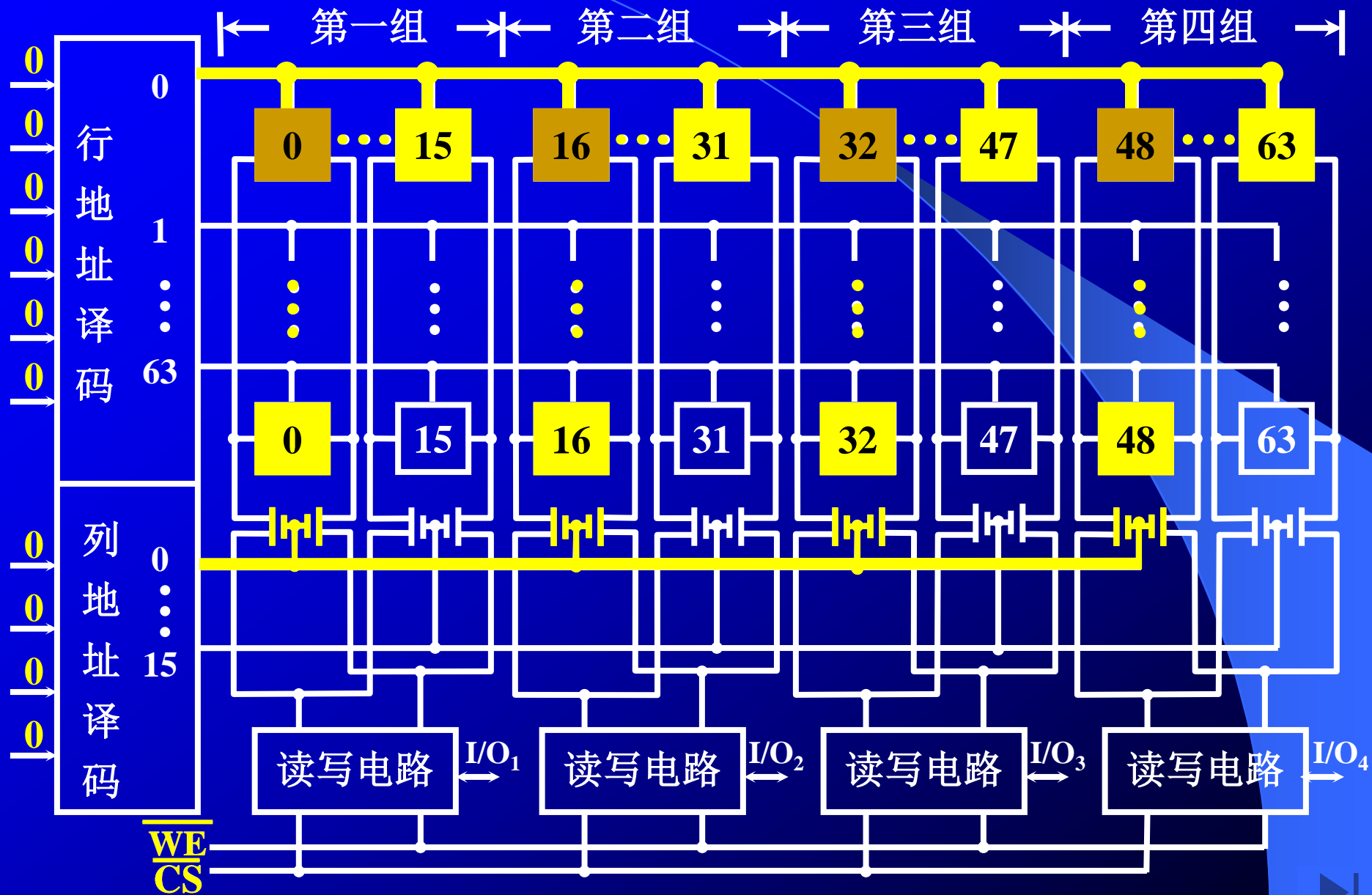
② Intel 2114 RAM 矩阵 (64×64) 读 4.2



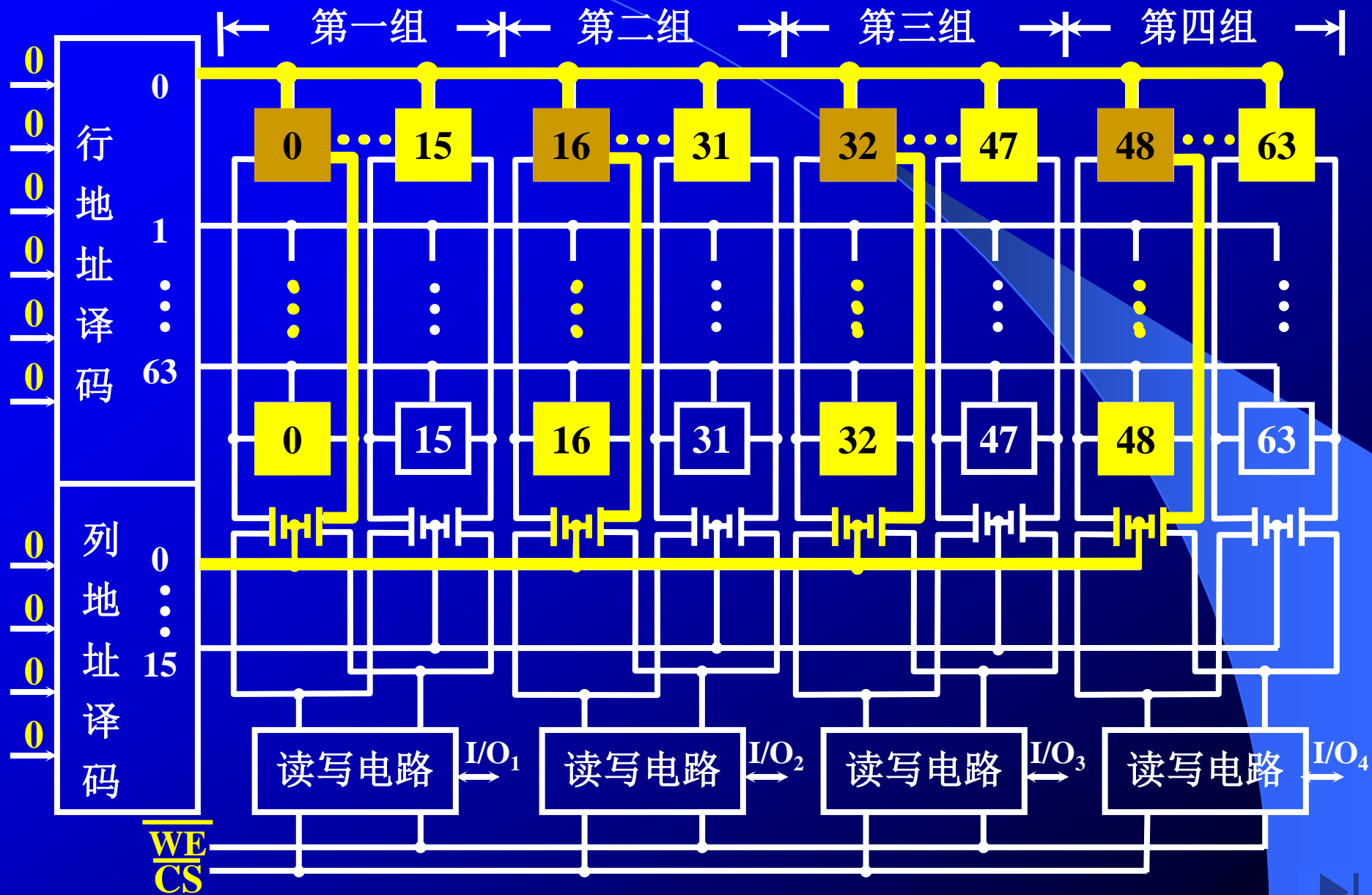
② Intel 2114 RAM 矩阵 (64×64) 读 4.2



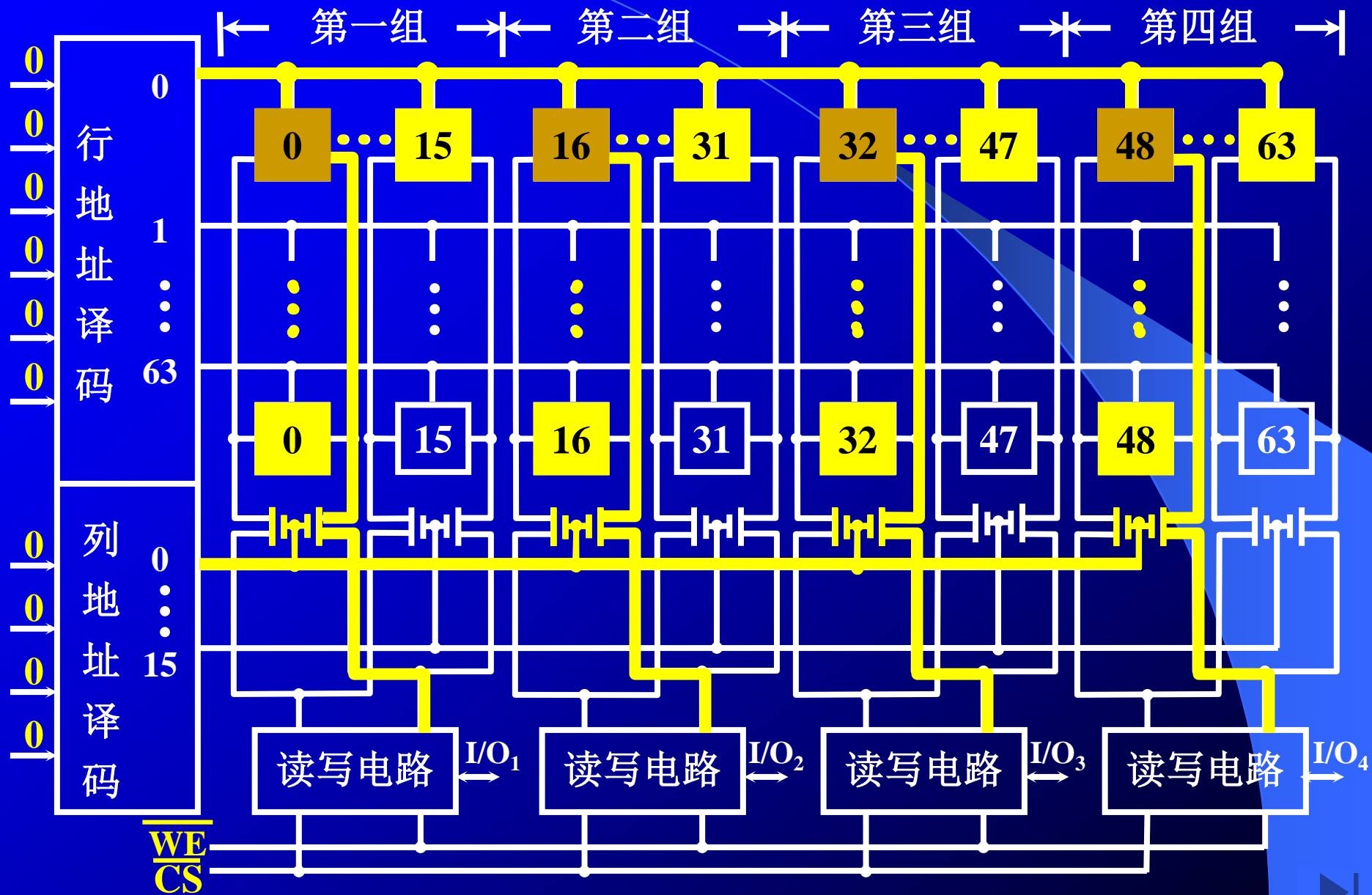
② Intel 2114 RAM 矩阵 (64×64) 读 4.2



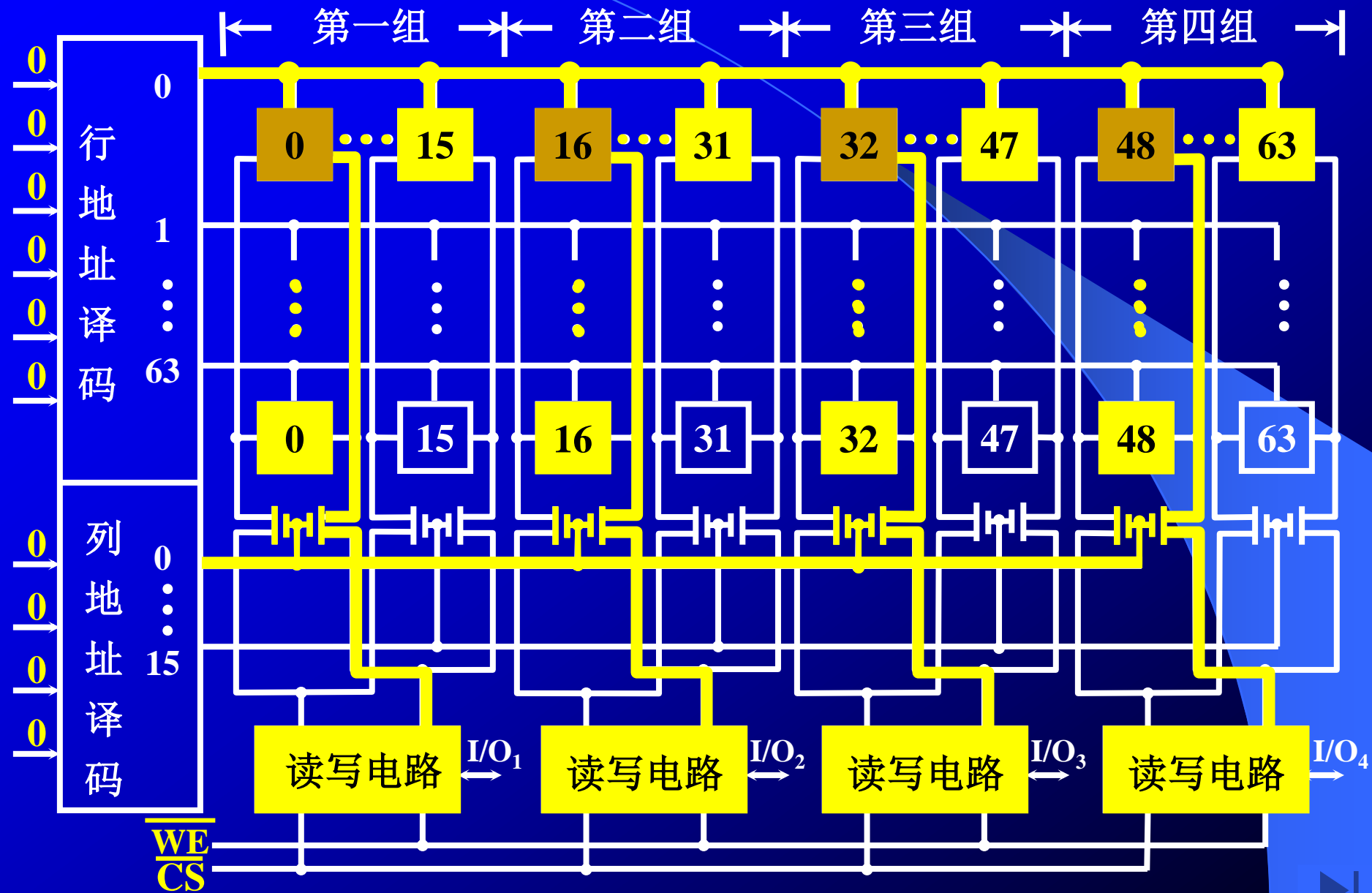
② Intel 2114 RAM 矩阵 (64×64) 读 4.2



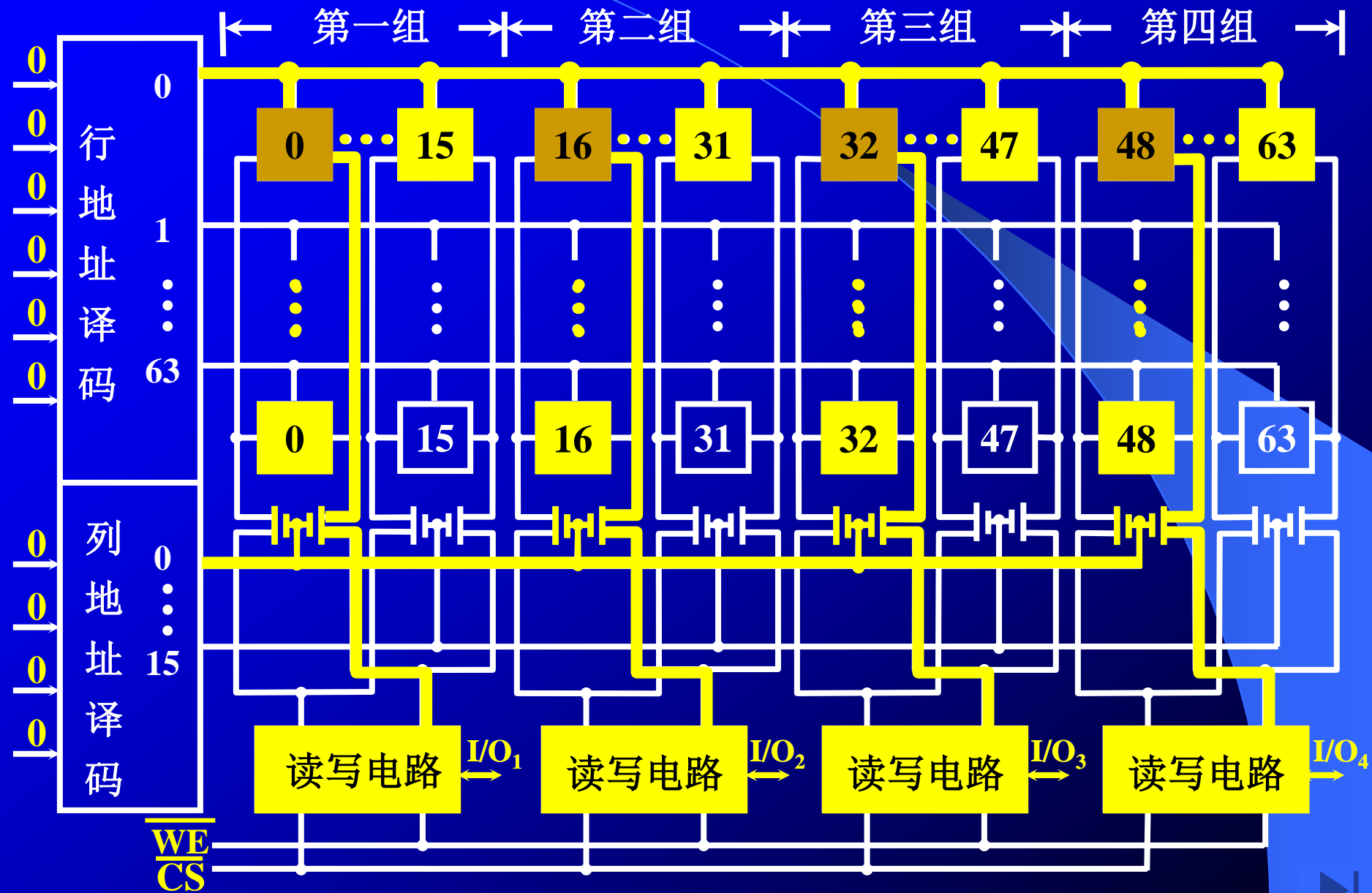
② Intel 2114 RAM 矩阵 (64×64) 读 4.2



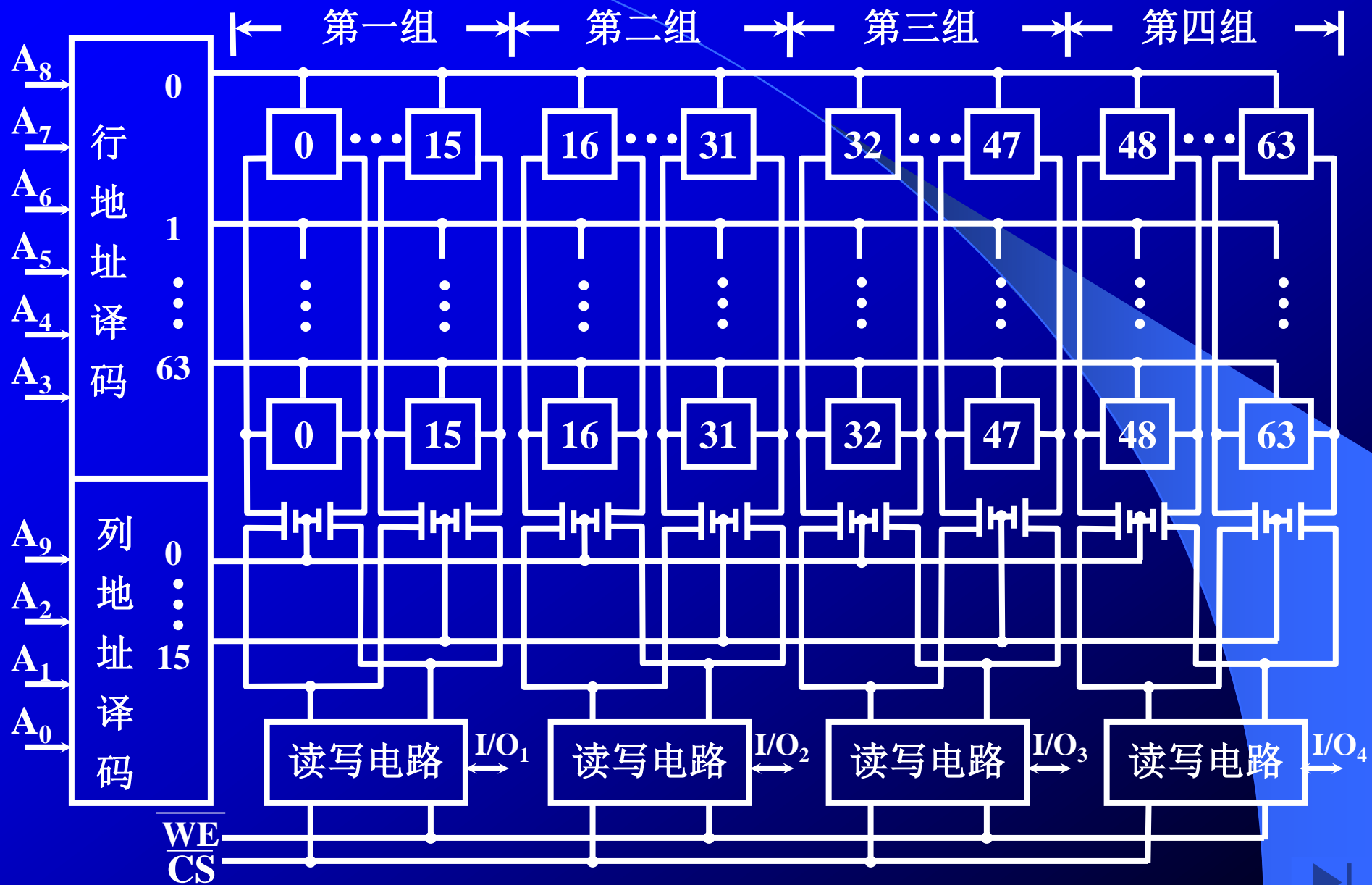
② Intel 2114 RAM 矩阵 (64×64) 读 4.2



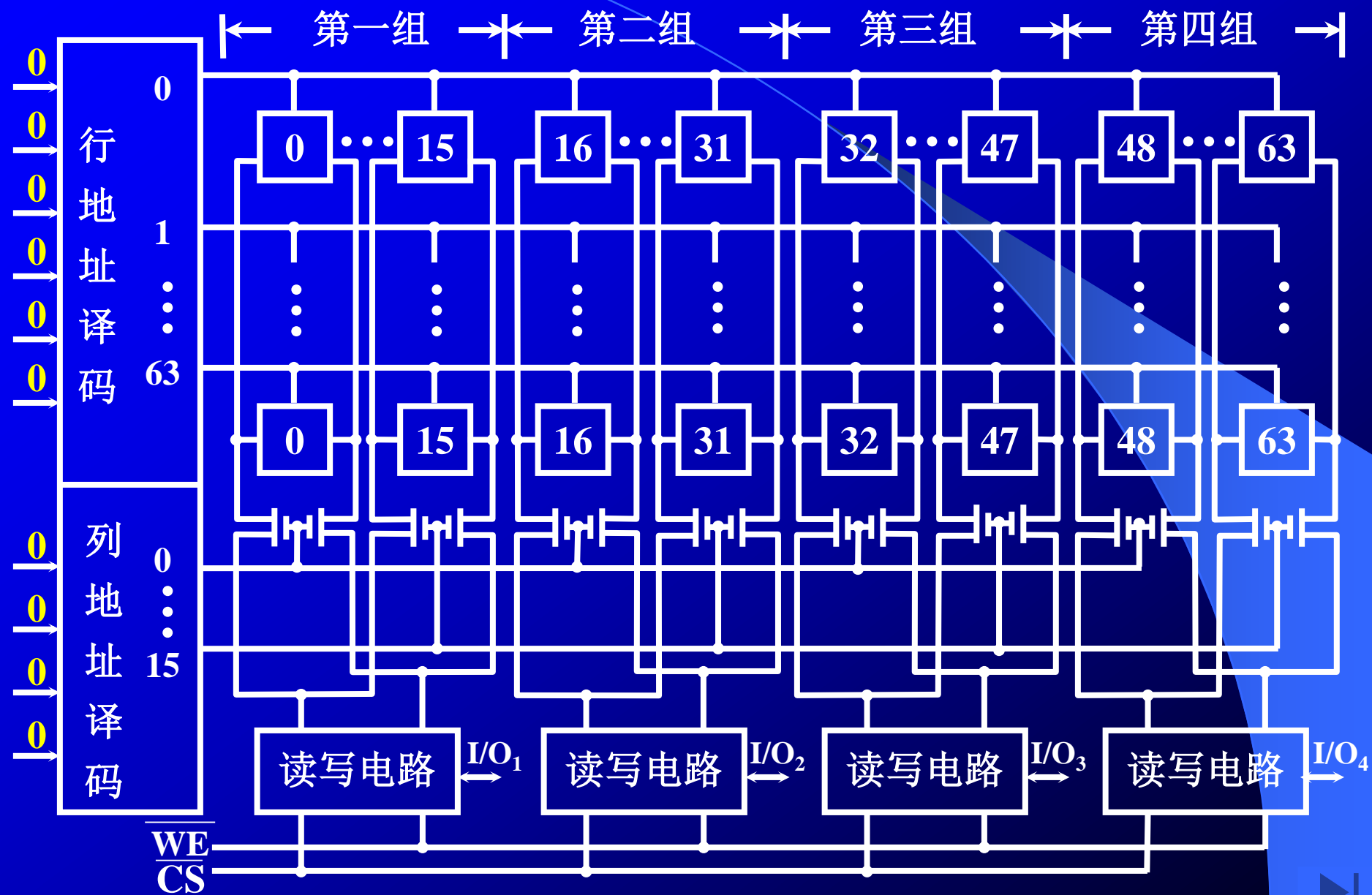
② Intel 2114 RAM 矩阵 (64×64) 读 4.2



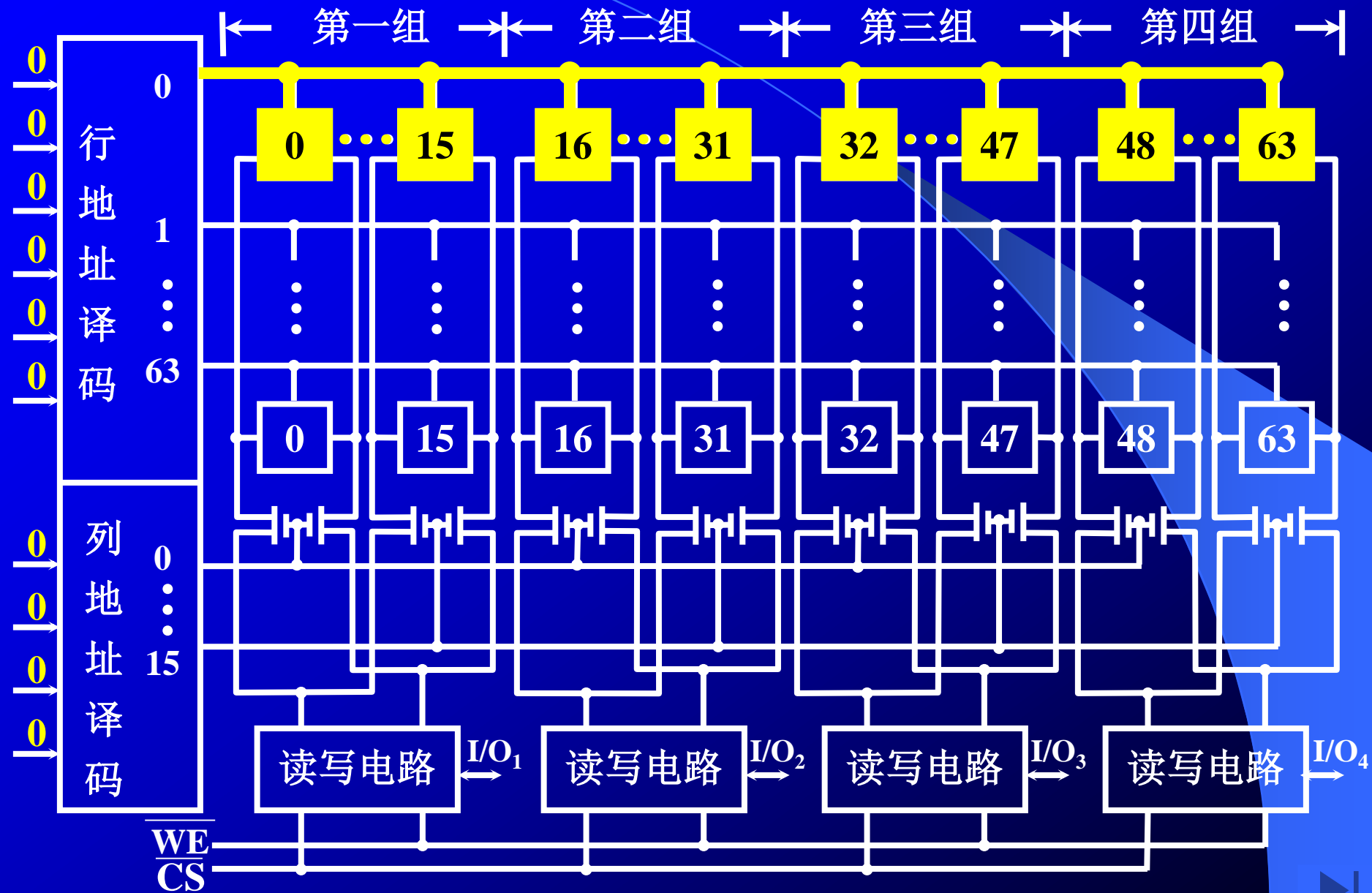
③ Intel 2114 RAM 矩阵 (64×64) 写 4.2



③ Intel 2114 RAM 矩阵 (64×64) 写 4.2



③ Intel 2114 RAM 矩阵 (64×64) 写 4.2



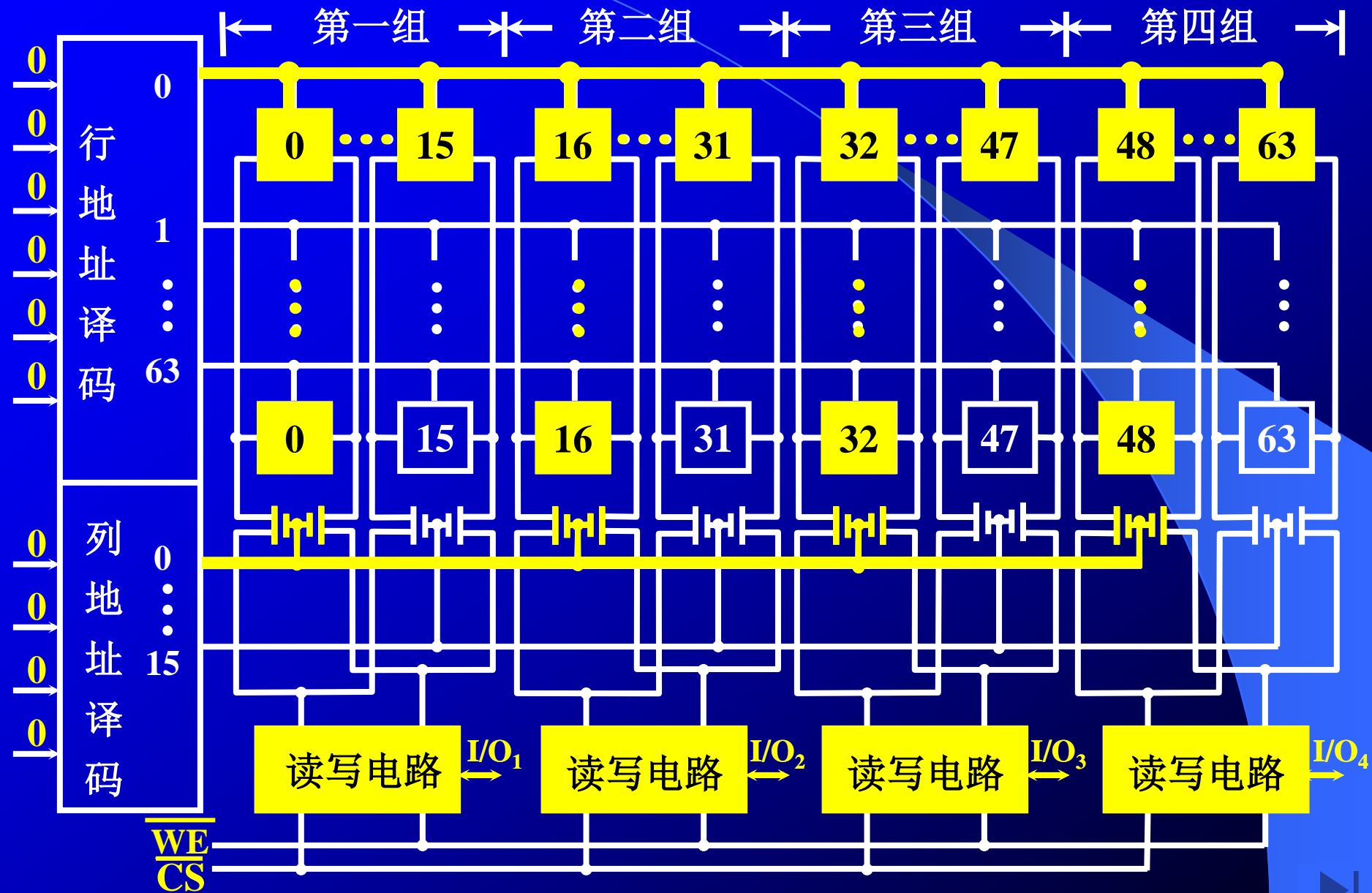
4.2



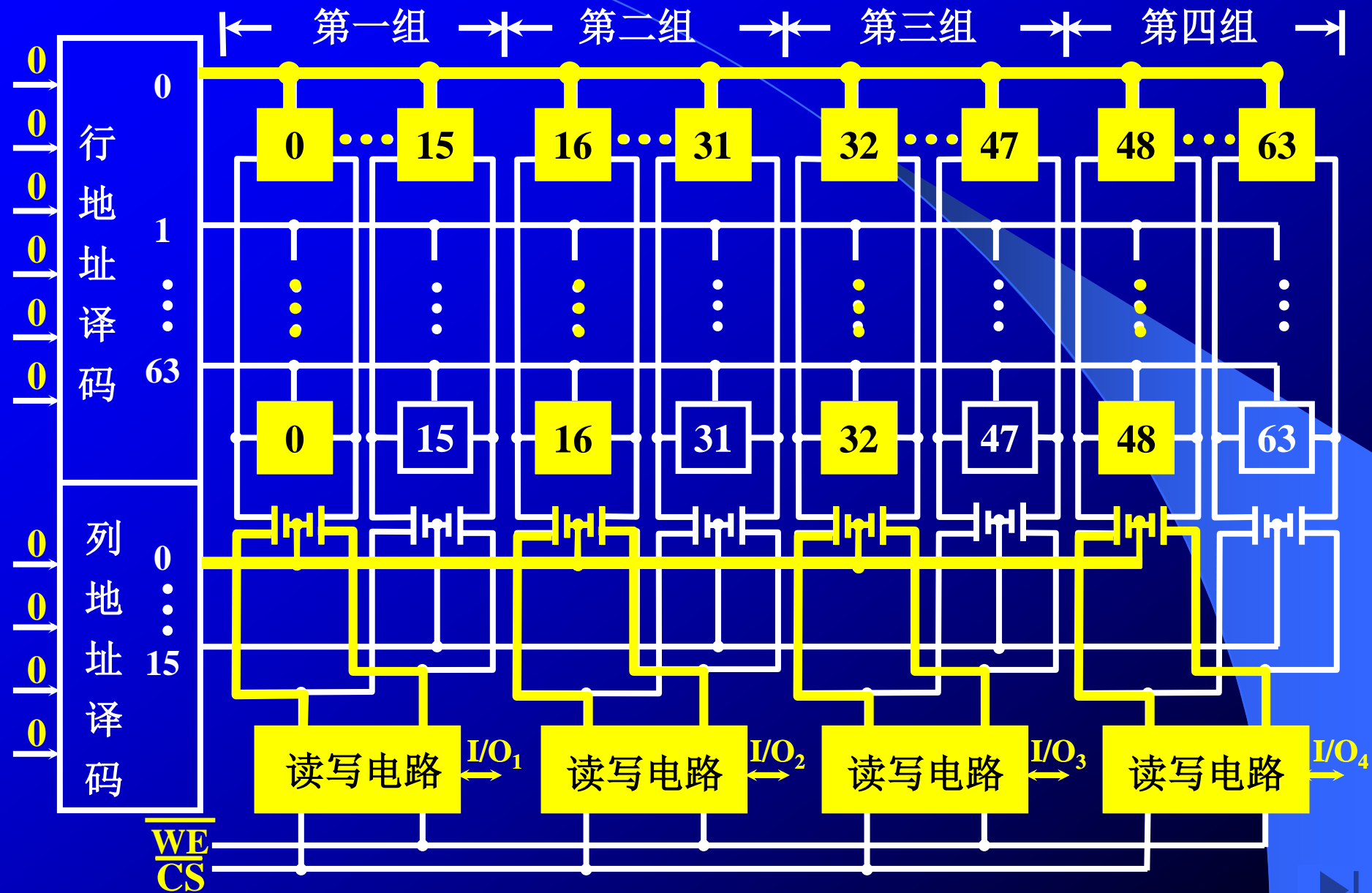
4.2



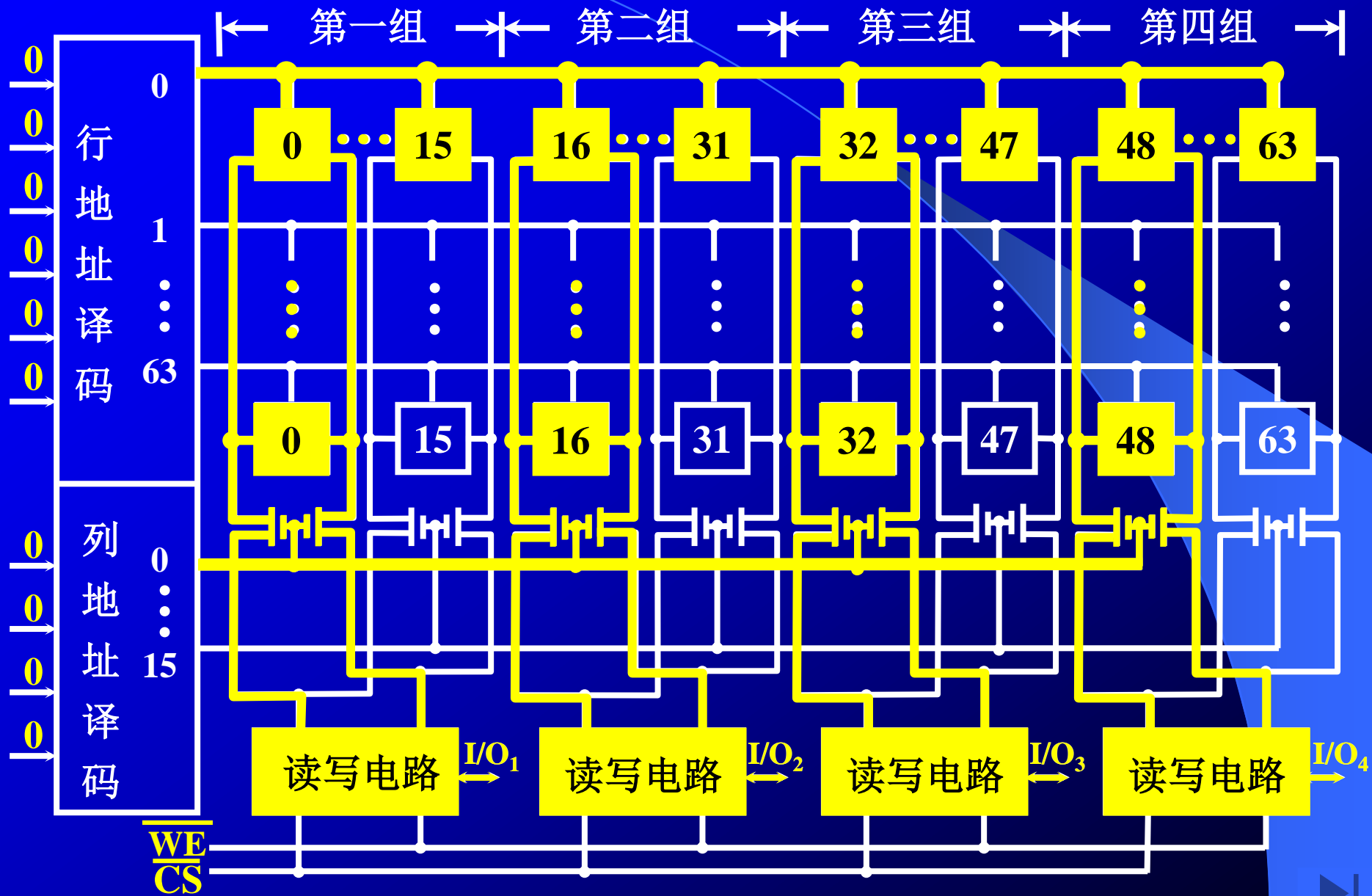
③ Intel 2114 RAM 矩阵 (64×64) 写 4.2



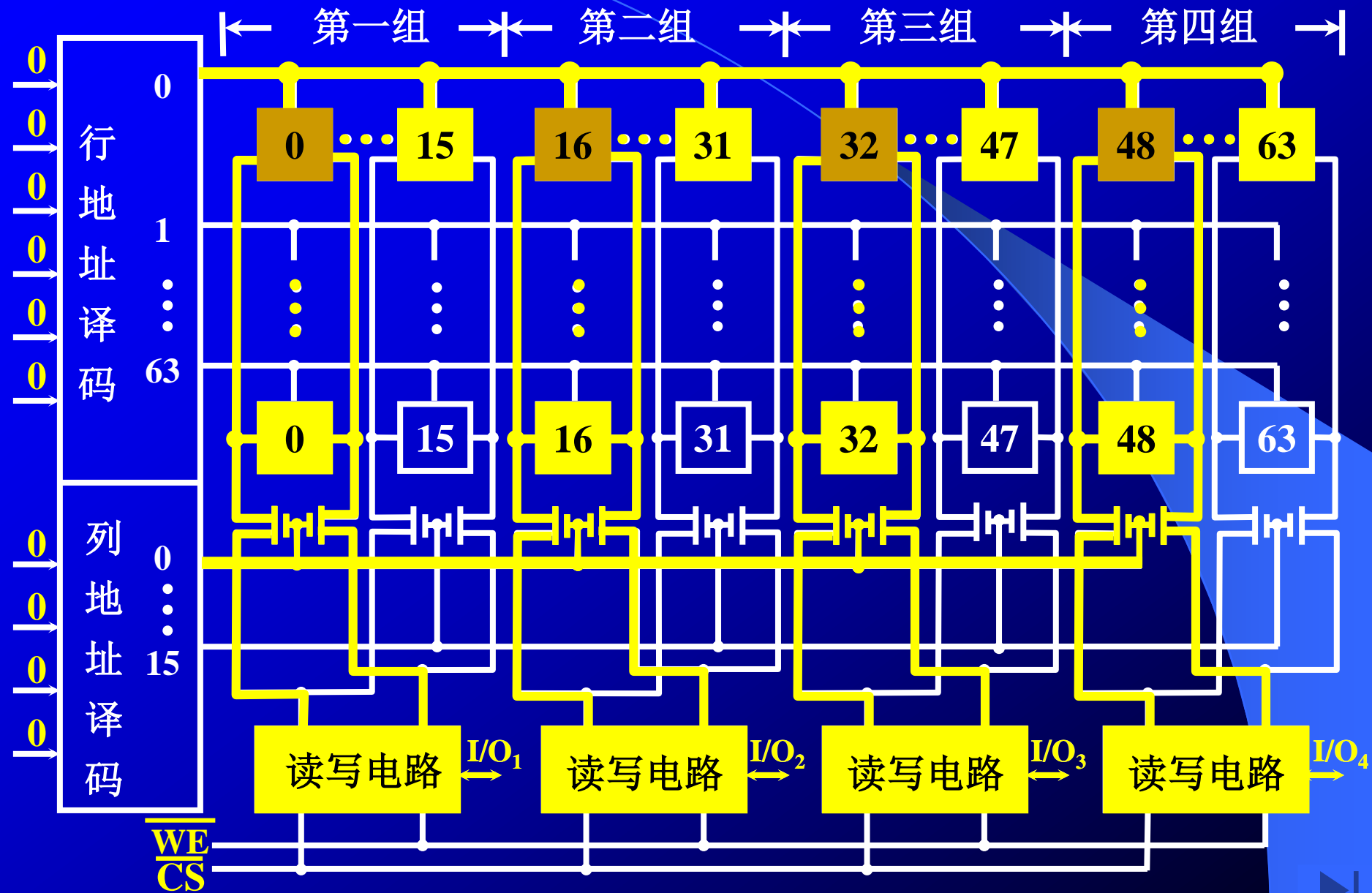
③ Intel 2114 RAM 矩阵 (64×64) 写 4.2



③ Intel 2114 RAM 矩阵 (64×64) 写 4.2

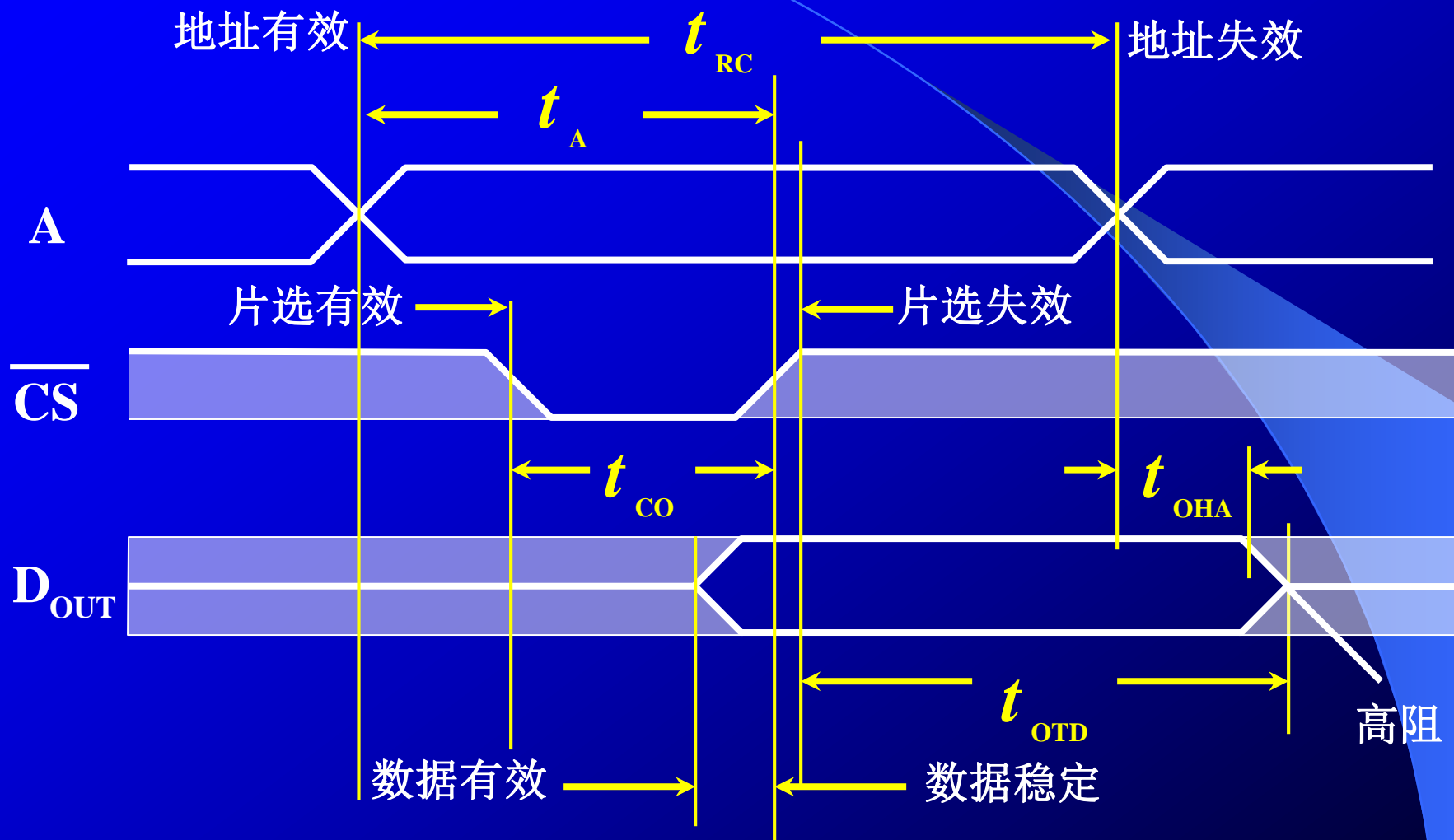


③ Intel 2114 RAM 矩阵 (64×64) 写 4.2



(3) 静态 RAM 读 时序

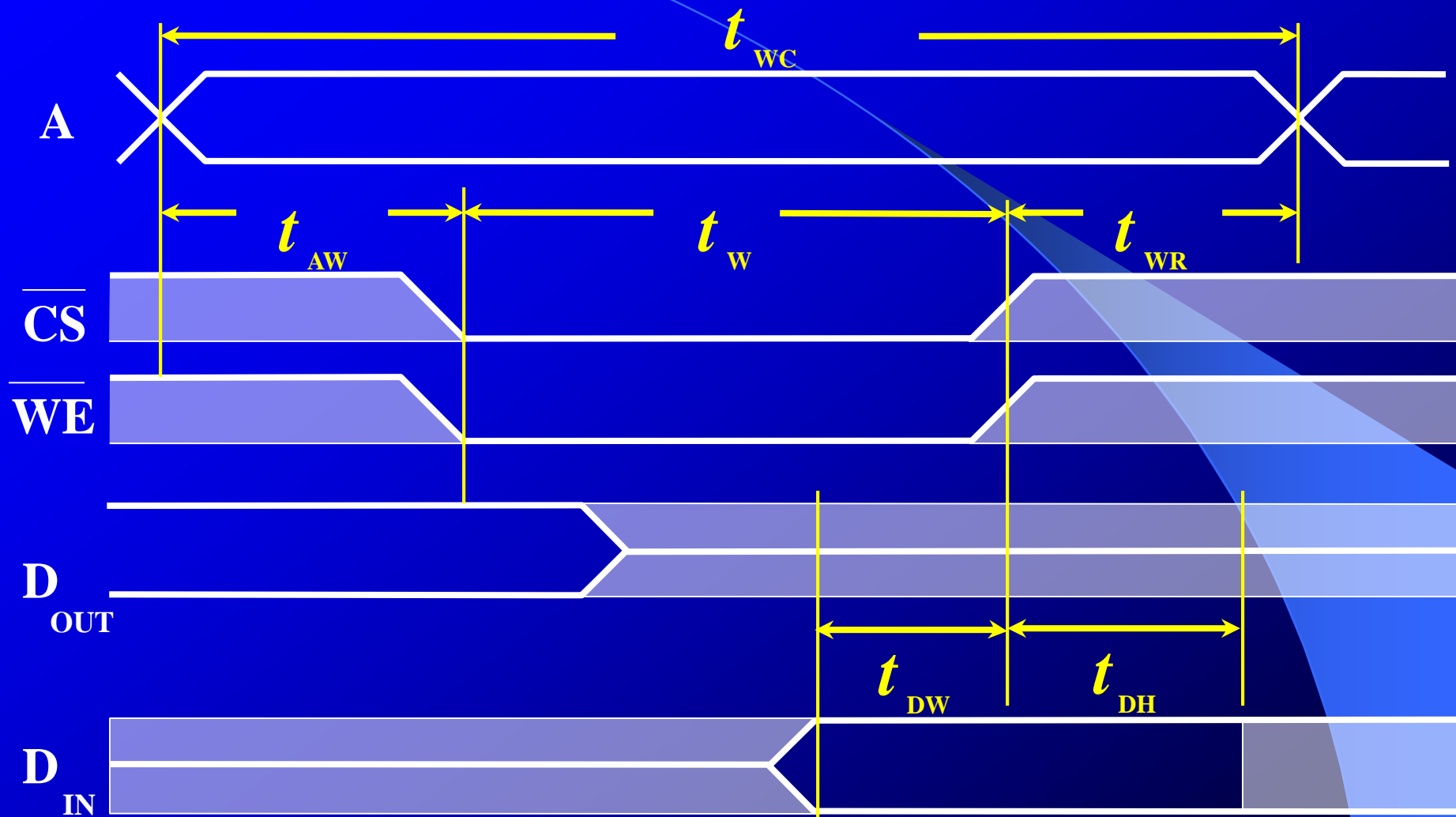
4.2



t_{OHA} 地址失效后的数据维持时间

(4) 静态 RAM (2114) 写时序

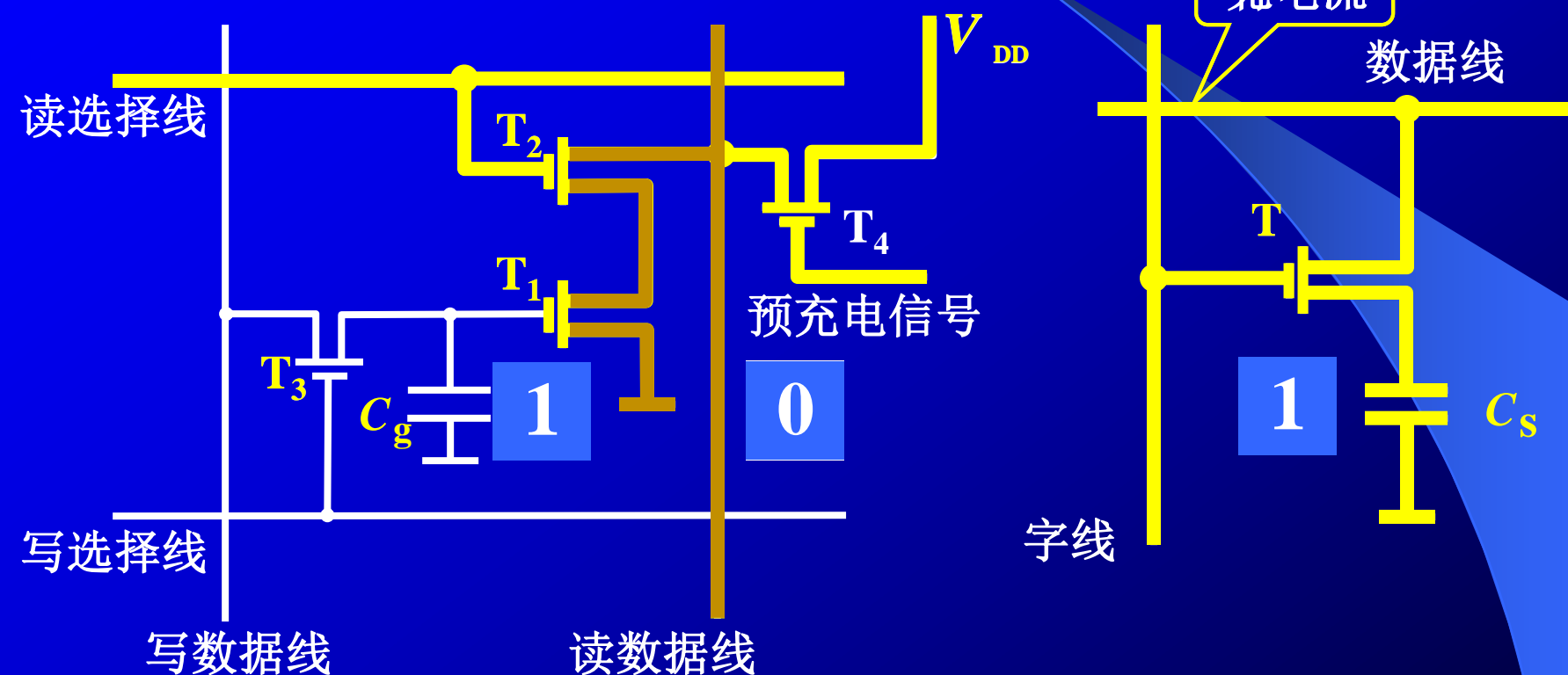
4.2



t_{DH} \overline{WE} 失效后的数据维持时间

2. 动态 RAM (DRAM)

(1) 动态 RAM 基本单元电路



读出与原存信息相反

写入与输入信息相同

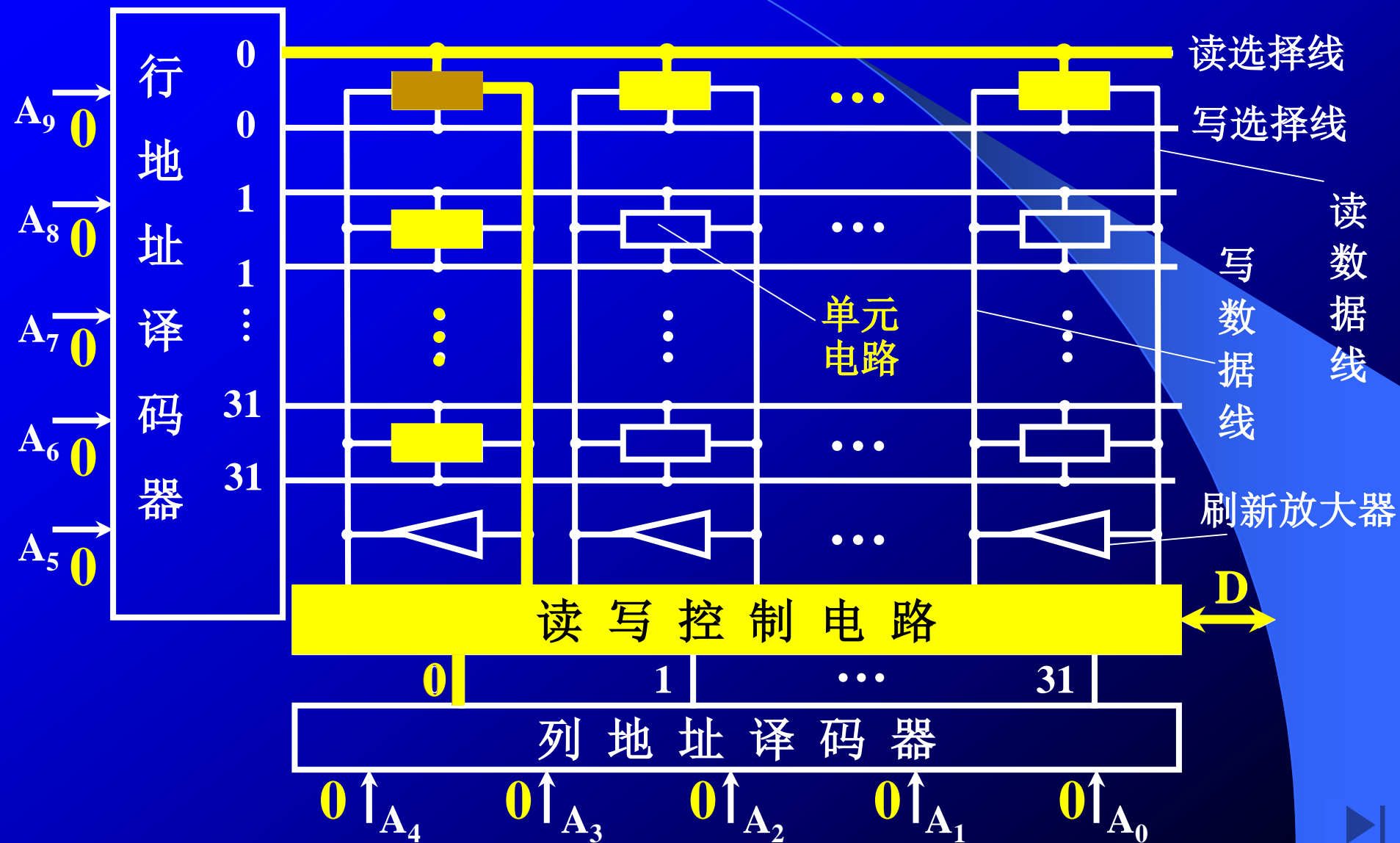
读出时数据线有电流为“1”

写入时 C_s 充电为“1” 放电为“0”

(2) 动态 RAM 芯片举例

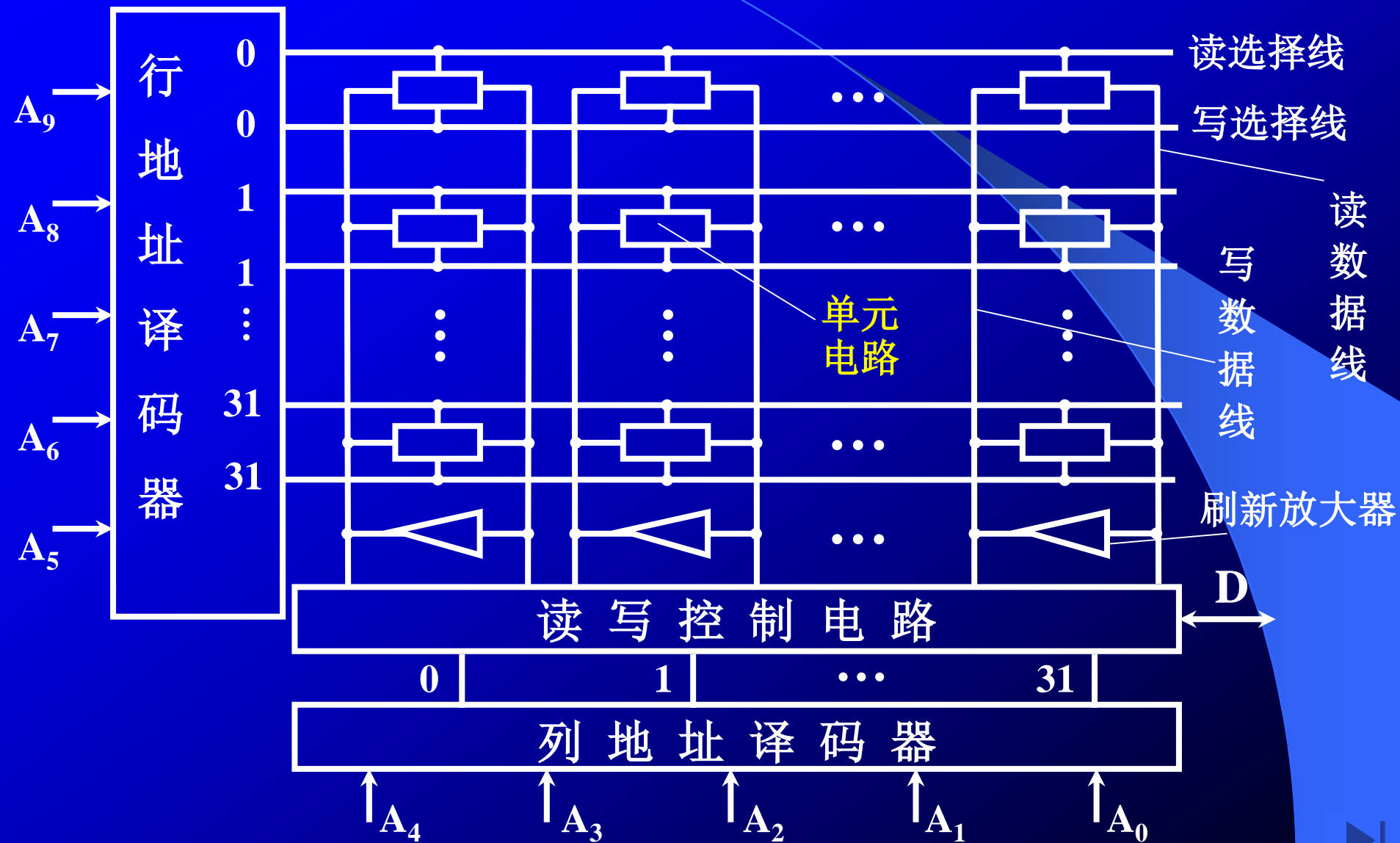
4.2

① 三管动态 RAM 芯片 (Intel 1103) 读



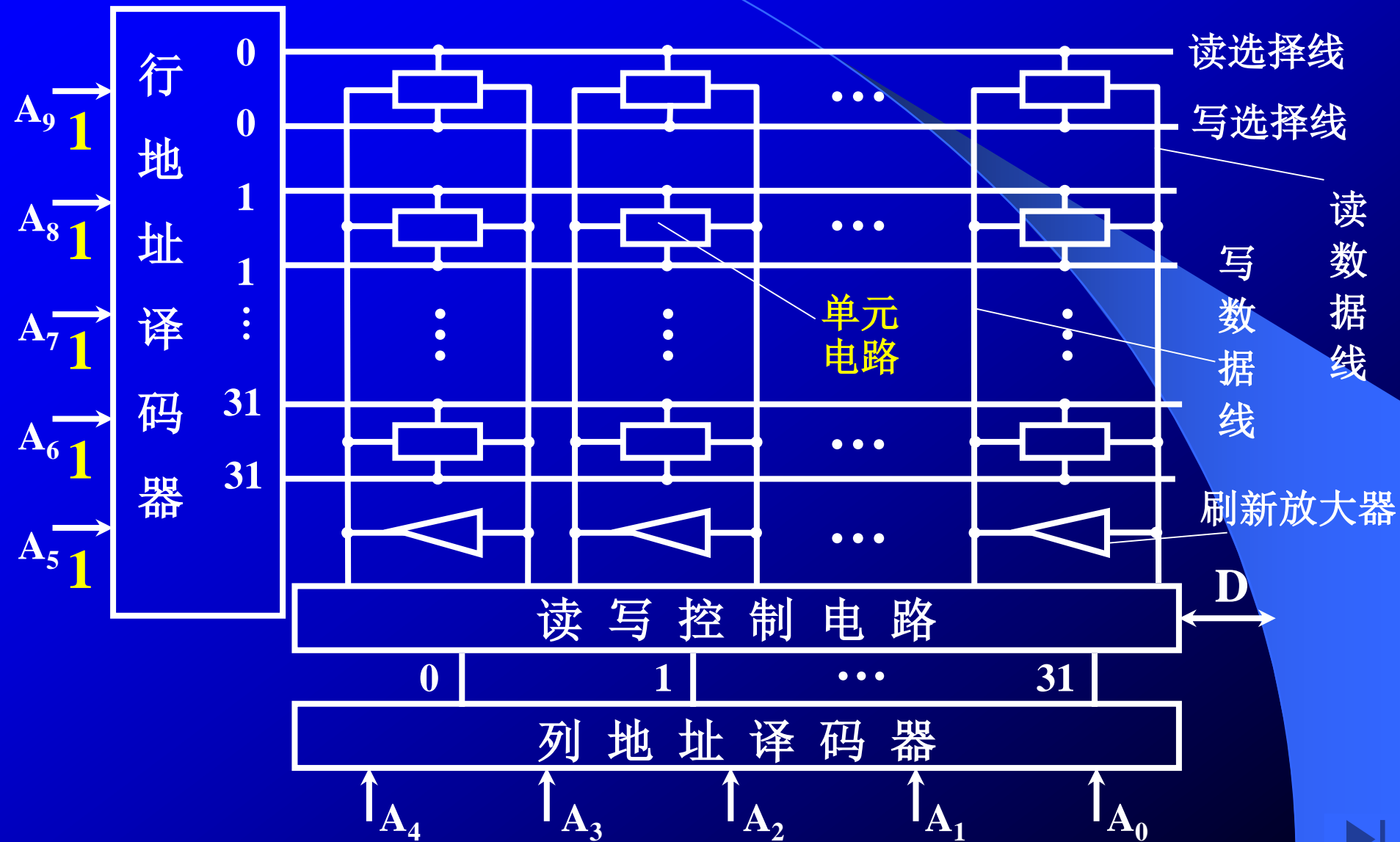
② 三管动态 RAM 芯片 (Intel 1103) 写

4.2



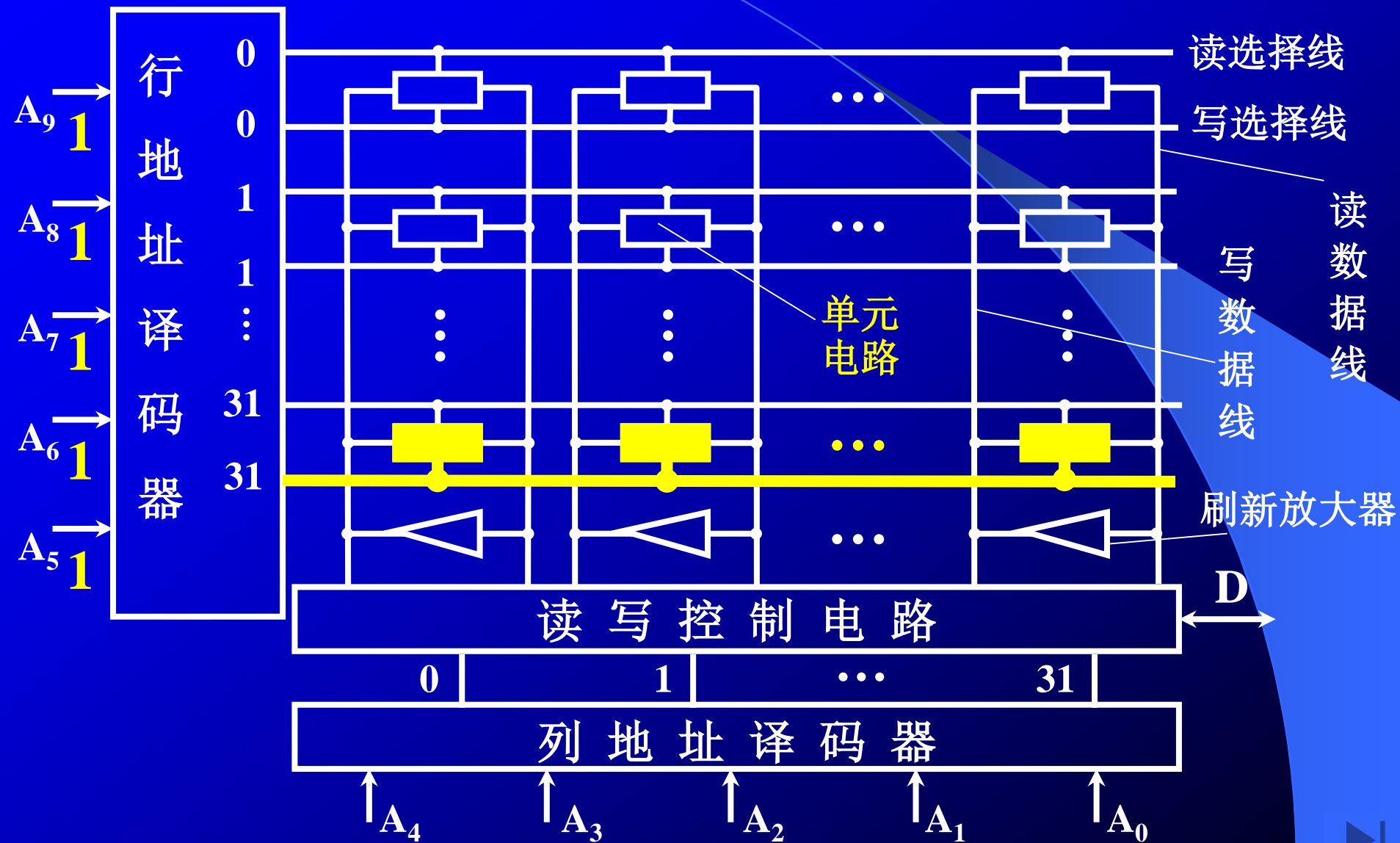
② 三管动态 RAM 芯片 (Intel 1103) 写

4.2



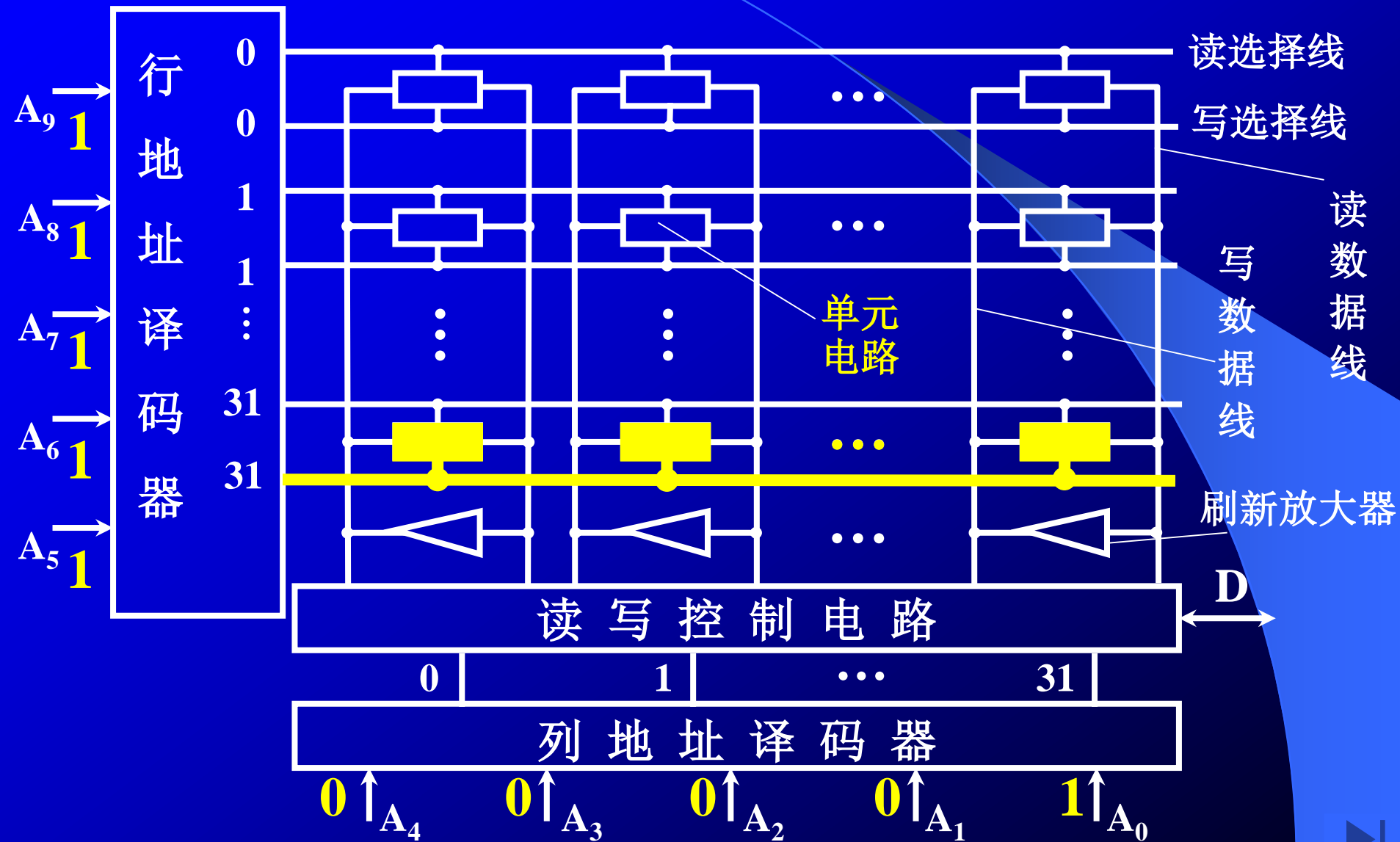
② 三管动态 RAM 芯片 (Intel 1103) 写

4.2



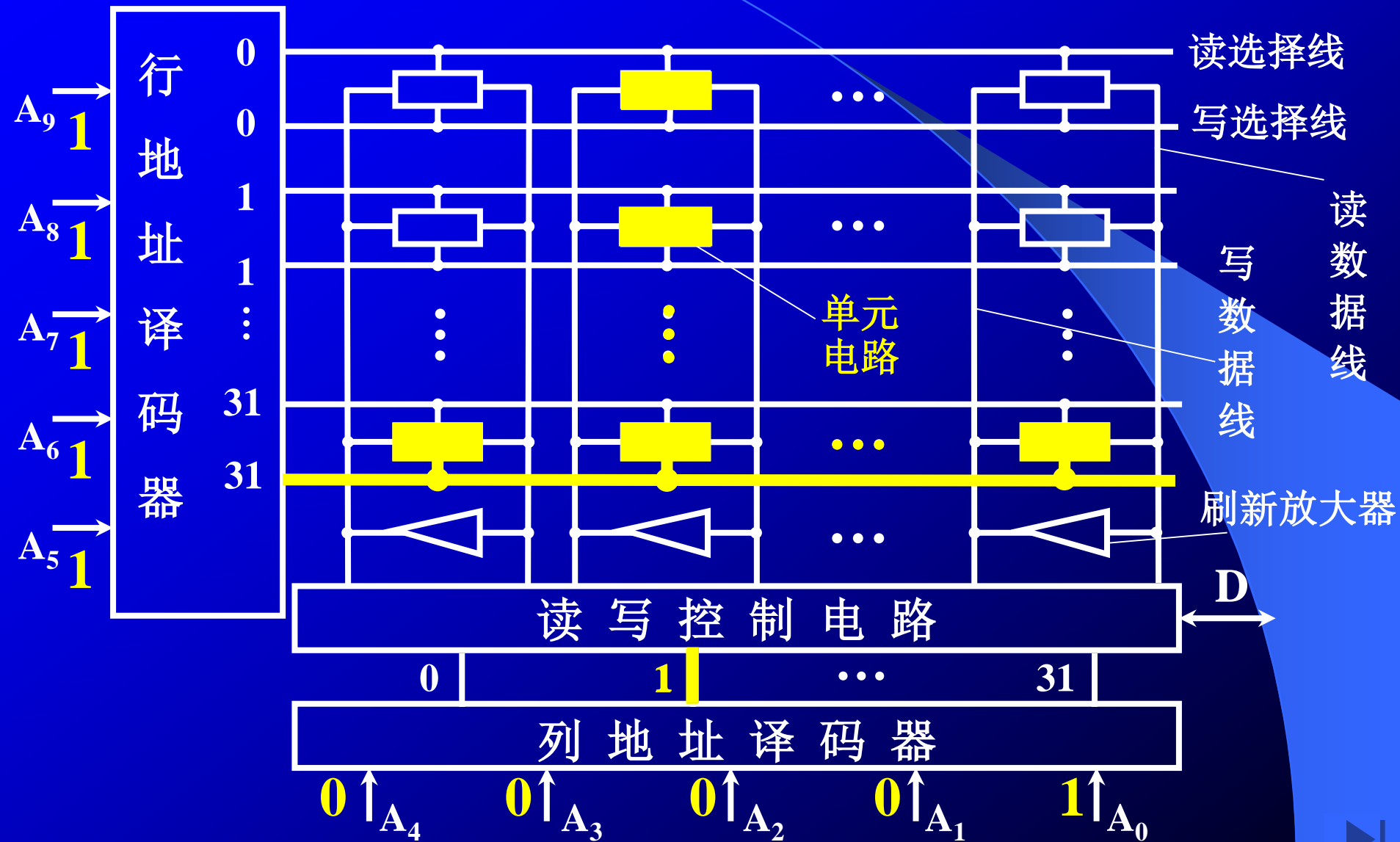
② 三管动态 RAM 芯片 (Intel 1103) 写

4.2



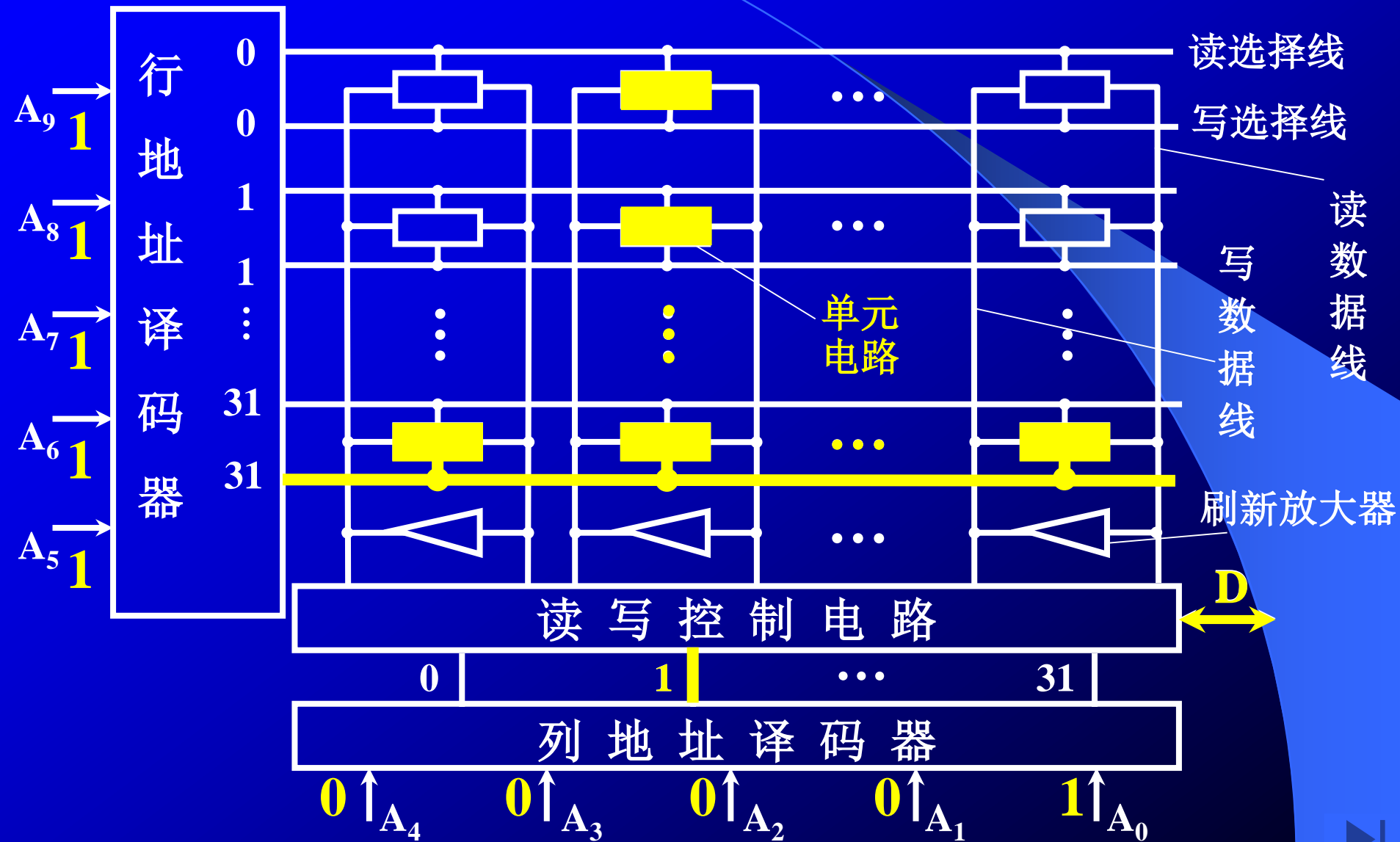
② 三管动态 RAM 芯片 (Intel 1103) 写

4.2



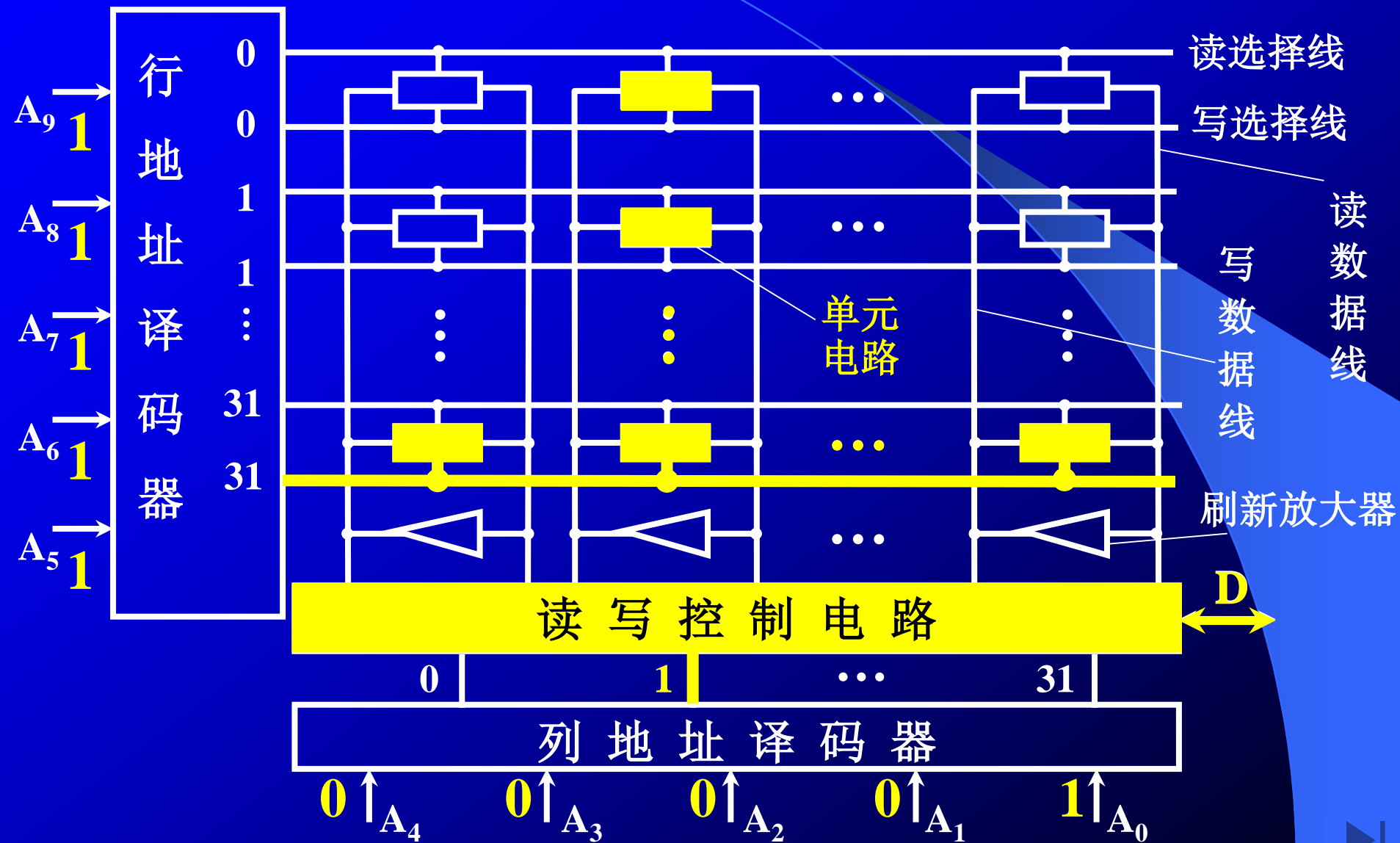
② 三管动态 RAM 芯片 (Intel 1103) 写

4.2



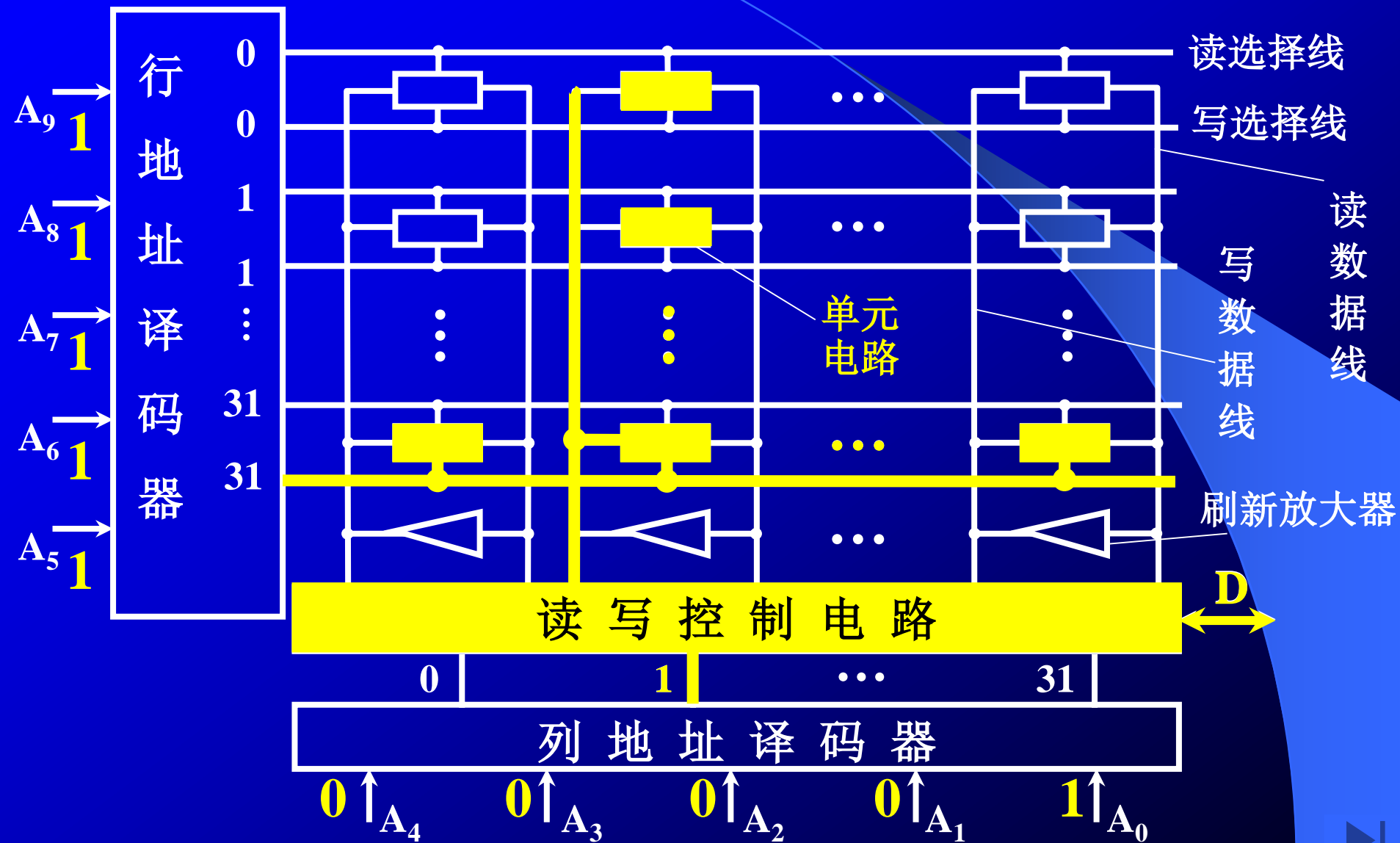
② 三管动态 RAM 芯片 (Intel 1103) 写

4.2



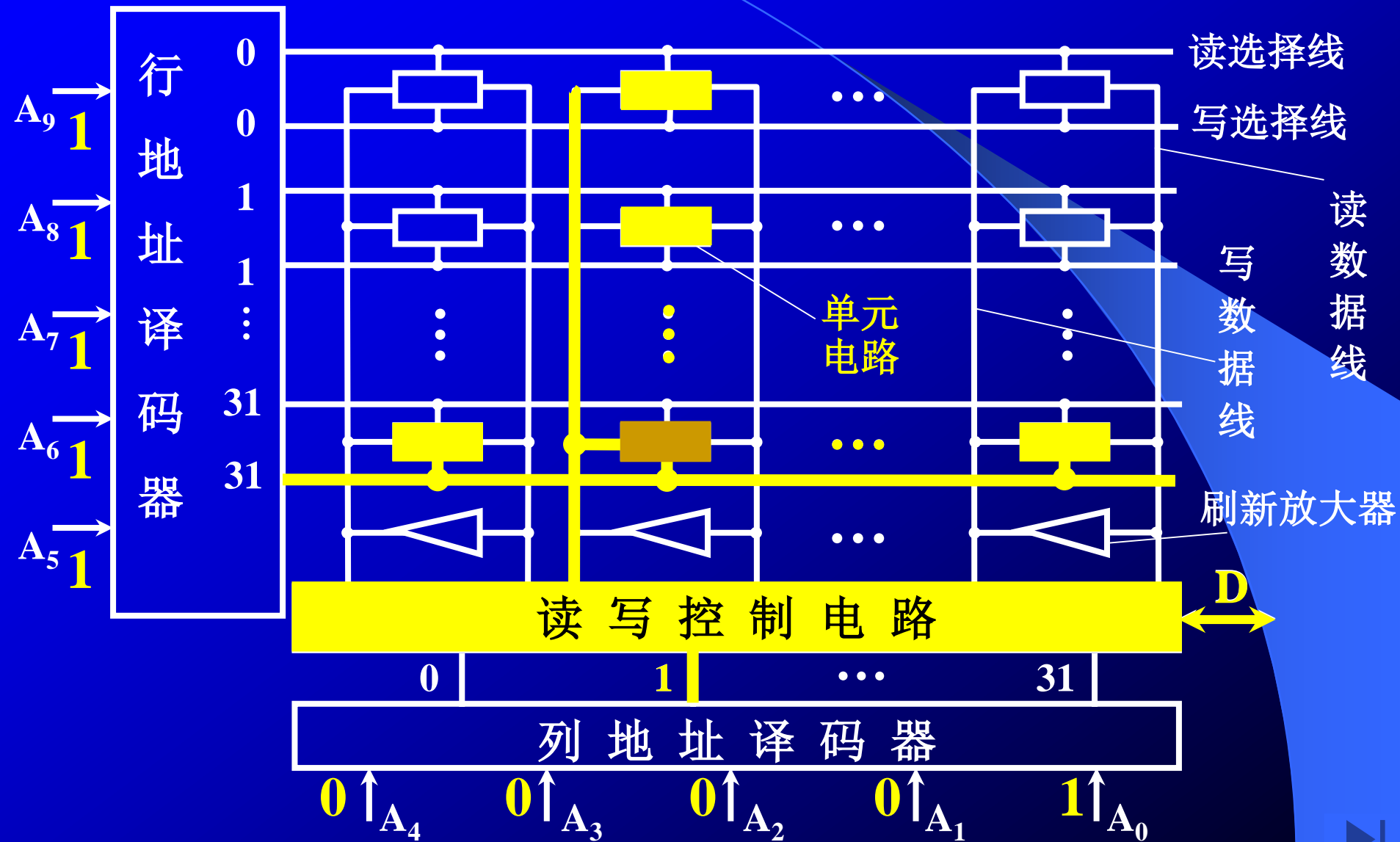
② 三管动态 RAM 芯片 (Intel 1103) 写

4.2

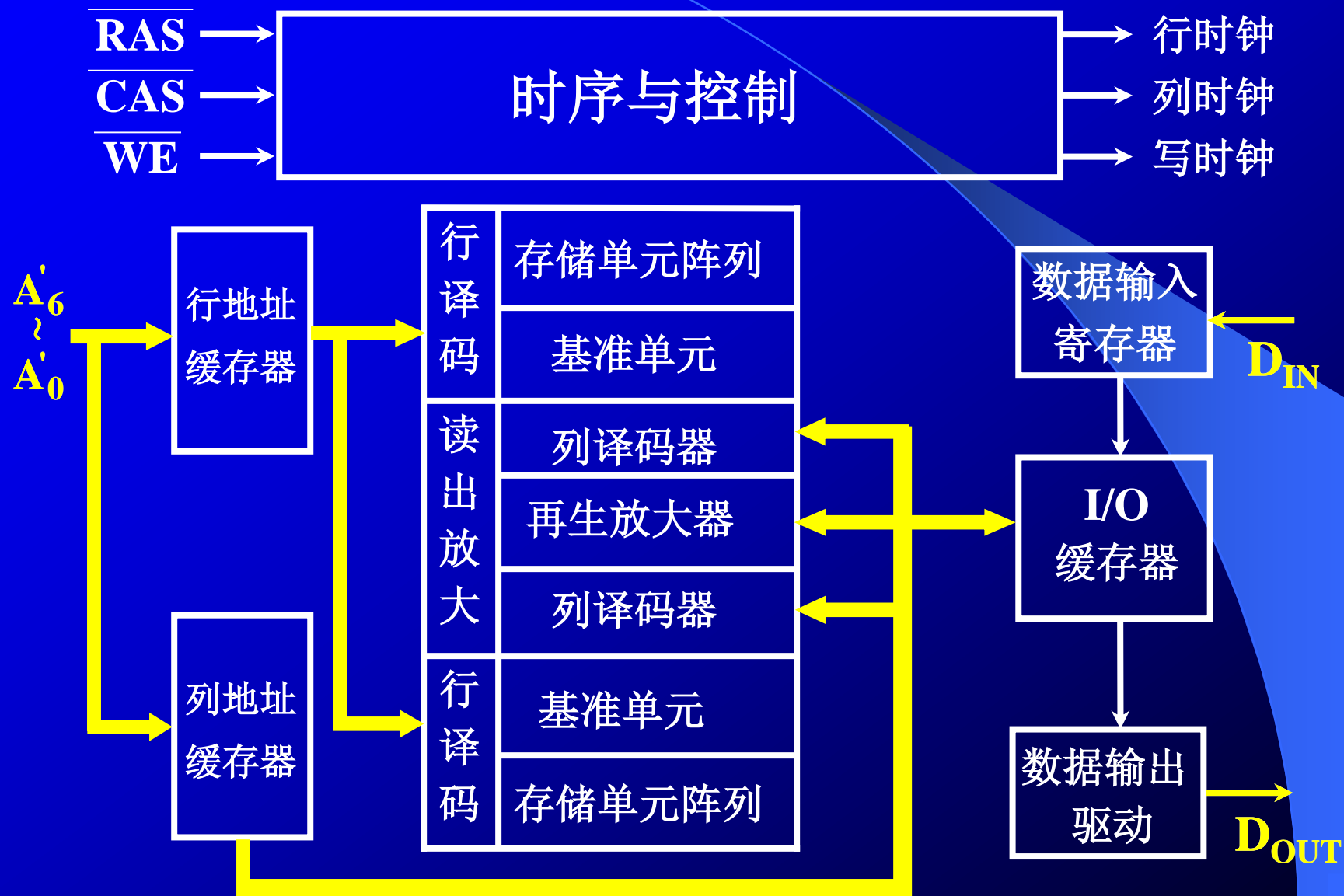


② 三管动态 RAM 芯片 (Intel 1103) 写

4.2

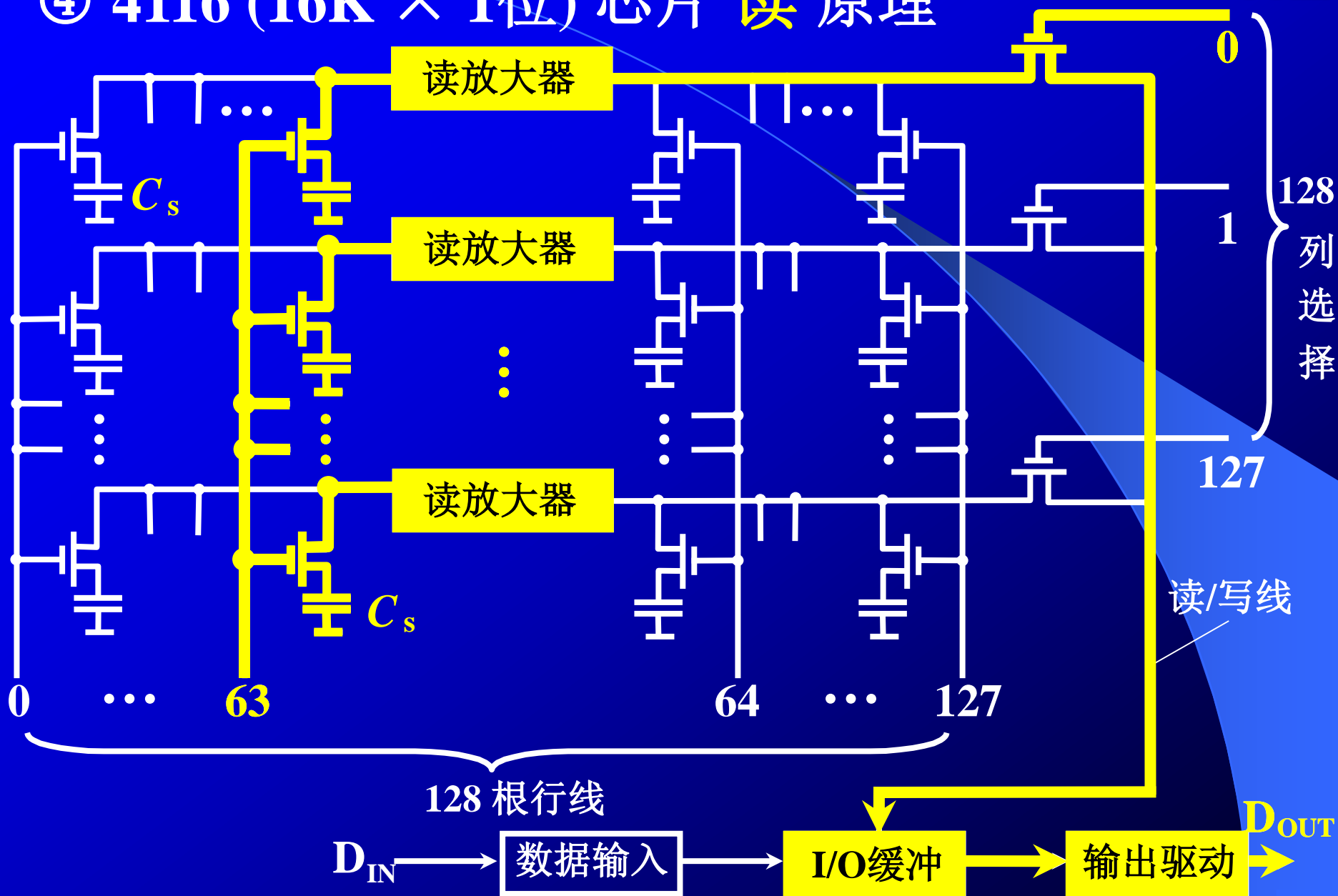


③ 单管动态 RAM 4116 (16K × 1位) 外特性 4.2



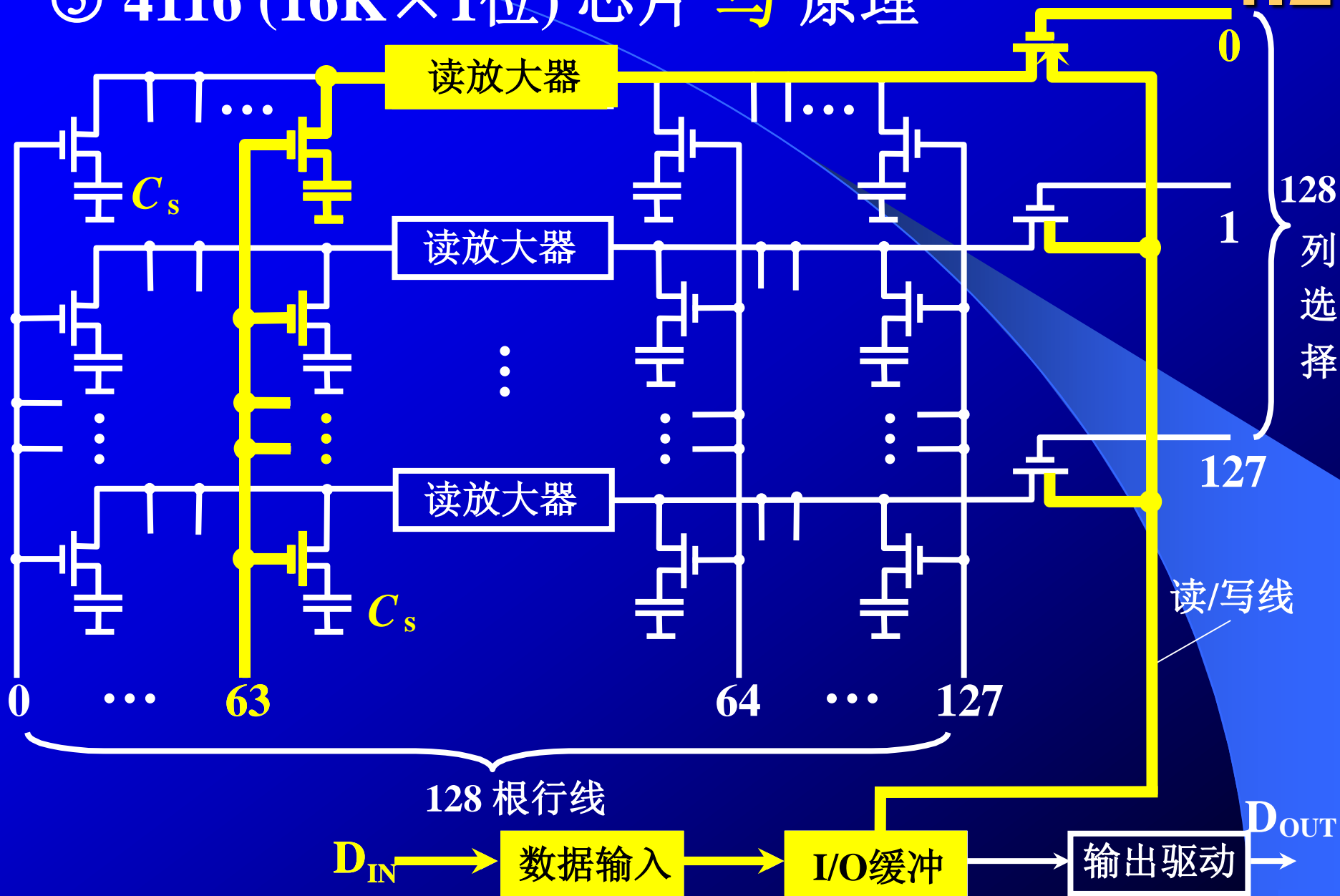
④ 4116 (16K × 1位) 芯片 读 原理

4.2



⑤ 4116 (16K×1位) 芯片 写 原理

4.2



(3) 动态 RAM 时序

行、列地址分开传送

读时序

行地址 $\overline{\text{RAS}}$ 有效
写允许 $\overline{\text{WE}}$ 有效(高)
列地址 $\overline{\text{CAS}}$ 有效
数据 D_{OUT} 有效

写时序

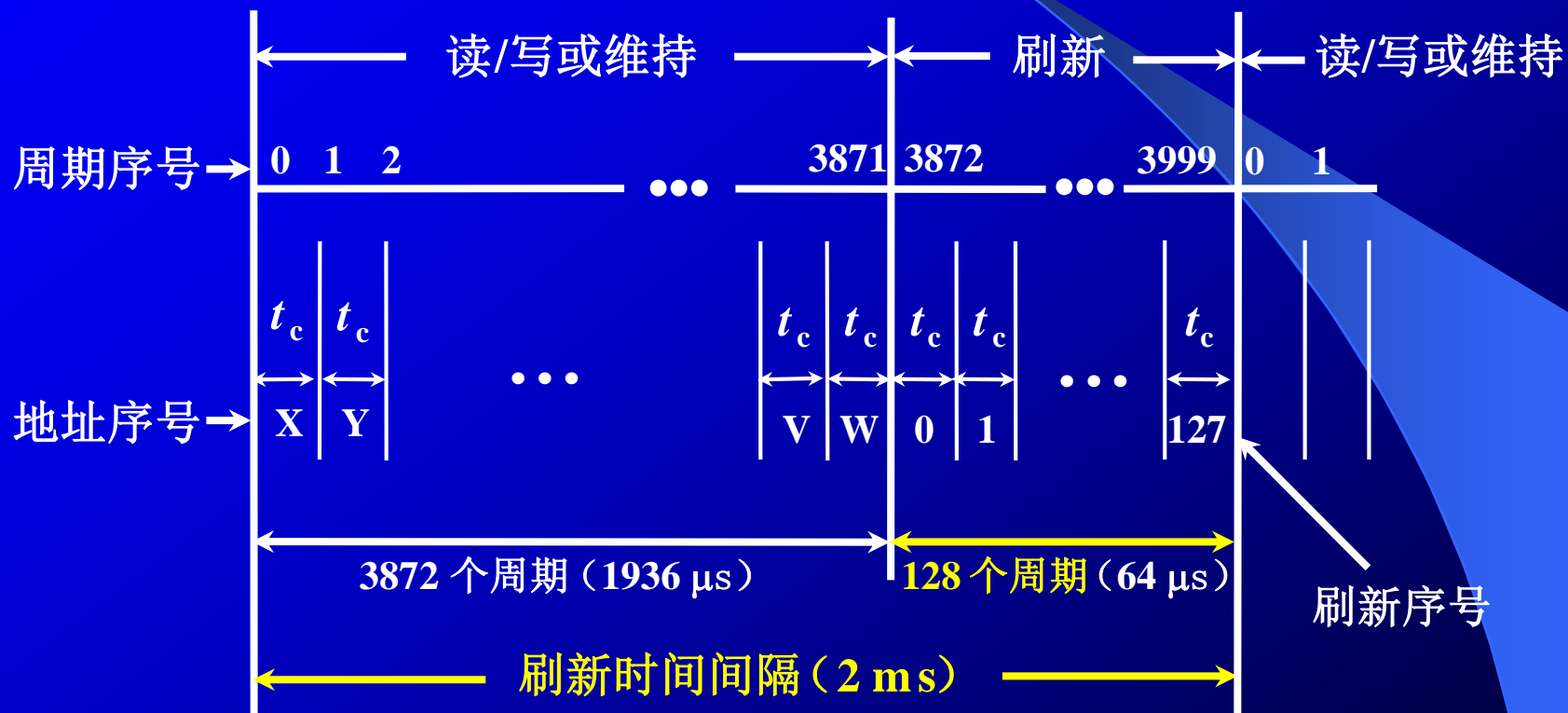
行地址 $\overline{\text{RAS}}$ 有效
写允许 $\overline{\text{WE}}$ 有效(低)
数据 D_{IN} 有效
列地址 $\overline{\text{CAS}}$ 有效

(4) 动态 RAM 刷新

4.2

刷新与行地址有关

① 集中刷新（存取周期为 $0.5\ \mu\text{s}$ ）以 128×128 矩阵为例



“死区” 为 $0.5\ \mu\text{s} \times 128 = 64\ \mu\text{s}$

“死时间率” 为 $128/4\ 000 \times 100\% = 3.2\%$

② 分散刷新（存取周期为 $1\mu\text{s}$ ）

4.2

以 128×128 矩阵为例



$$t_C = t_M + t_R$$

无“死区”

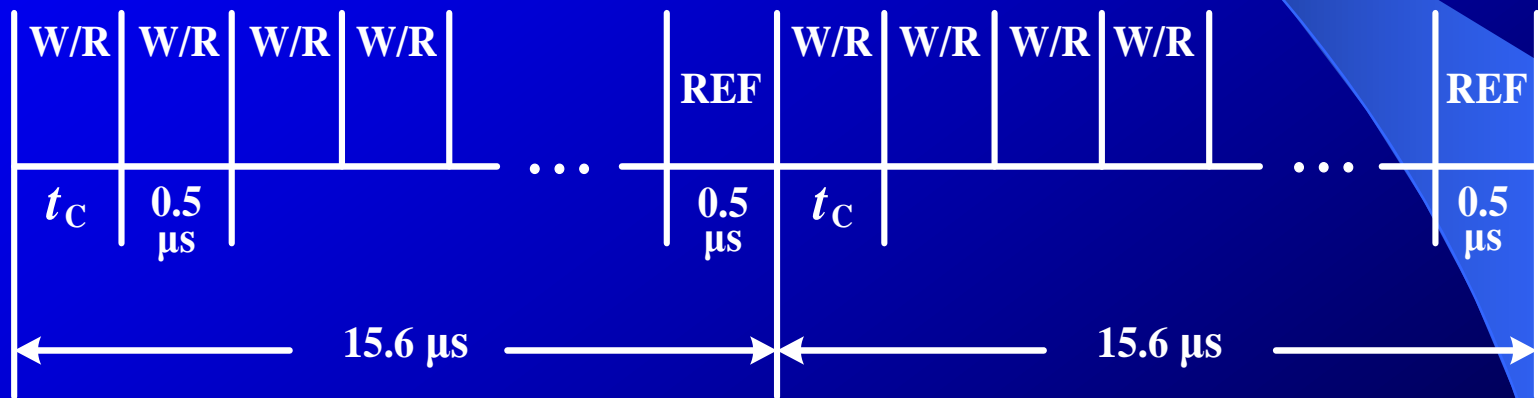
↓ ↓
读写 刷新

(存取周期为 $0.5\mu\text{s} + 0.5\mu\text{s}$)

③ 分散刷新与集中刷新相结合（异步刷新）^{4.2}

对于 128×128 的存储芯片（存取周期为 $0.5 \mu\text{s}$ ）

若每隔 $15.6 \mu\text{s}$ 刷新一次行



每行每隔 2 ms 刷新一次

“死区”为 $0.5 \mu\text{s}$

将刷新安排在指令译码阶段，不会出现“死区”

3. 动态 RAM 和静态 RAM 的比较

	主存 DRAM	SRAM 缓存
存储原理	电容	触发器
集成度	高	低
芯片引脚	少	多
功耗	小	大
价格	低	高
速度	慢	快
刷新	有	无

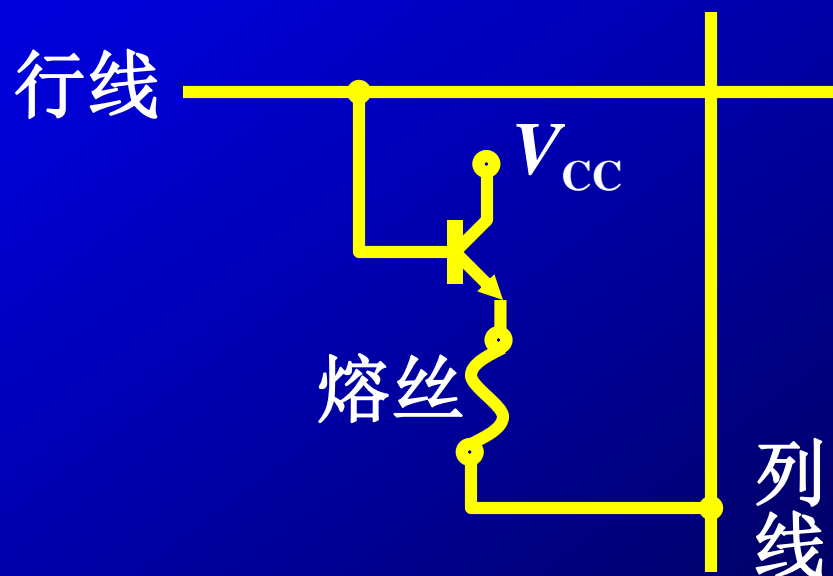
四、只读存储器 (ROM)

1. 掩模 ROM (MROM)

行列选择线交叉处有 MOS 管为 “1”

行列选择线交叉处无 MOS 管为 “0”

2. PROM (一次性编程)

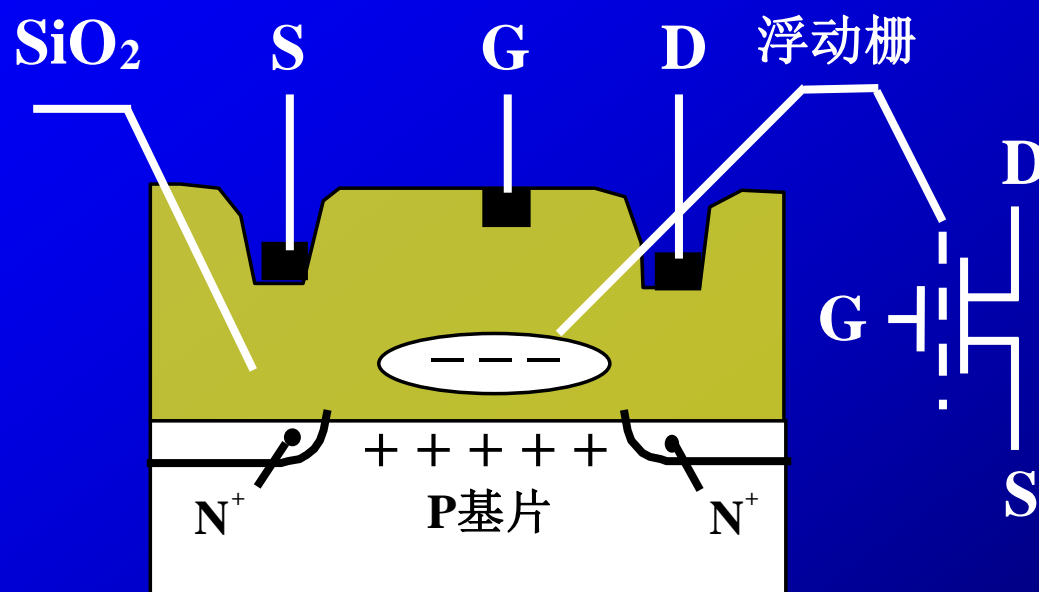


熔丝断 为 “0”

熔丝未断 为 “1”

3. EPROM (多次性编程)

(1) N型沟道浮动栅 MOS 电路



G 栅极

S 源

D 漏

紫外线全部擦洗

D 端加正电压

形成浮动栅

S 与 D 不导通为 “0”

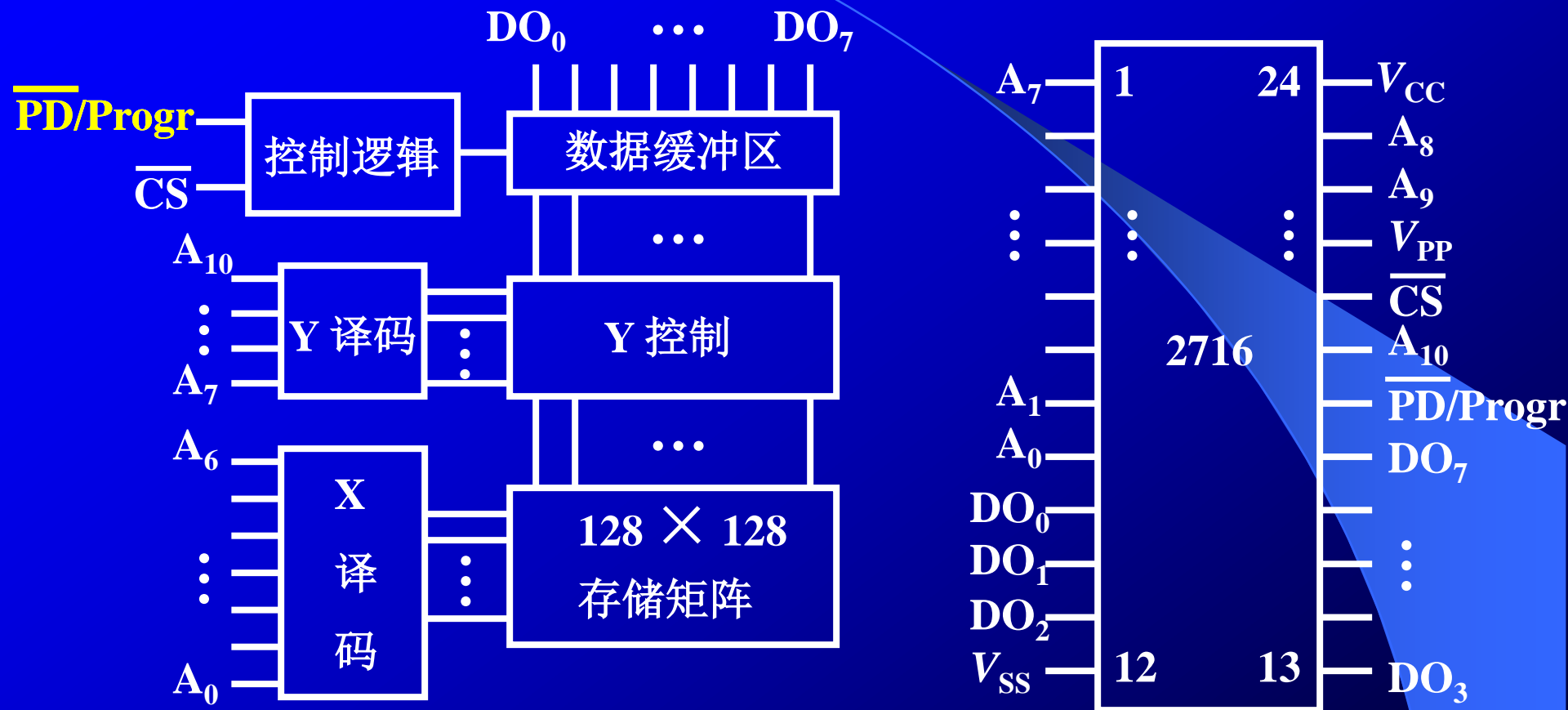
D 端不加正电压

不形成浮动栅

S 与 D 导通为 “1”

(2) 2716 EPROM 的逻辑图和引脚

4.2



$\overline{\text{PD/Progr}}$ 功率下降 / 编程输入端 读出时为低电平



4. EEPROM (多次性编程)

电可擦写

局部擦写

全部擦写

5. Flash Memory (闪速型存储器)

EPROM

价格便宜 集成度高

EEPROM

电可擦写重写

比 EEPROM 快 具备 RAM 功能

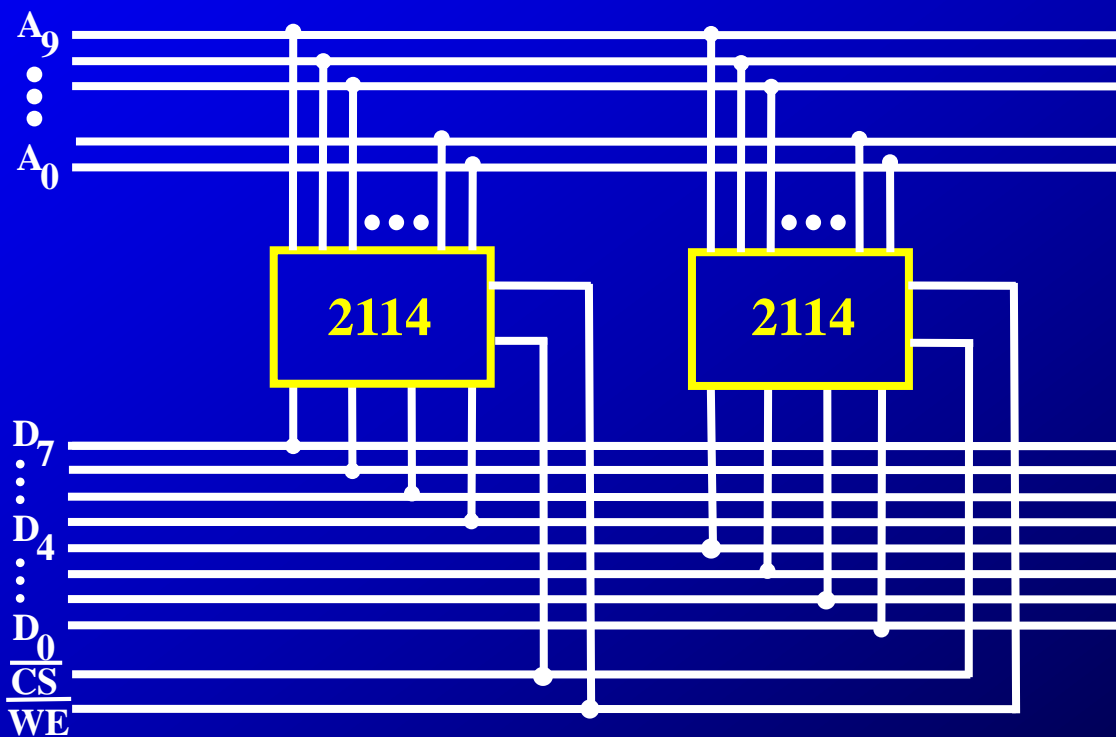
五、存储器与 CPU 的连接

4.2

1. 存储器容量的扩展

(1) 位扩展（增加存储字长）

用 **2片** $1K \times 4$ 位 存储芯片组成 $1K \times 8$ 位的存储器



10根地址线

8根数据线

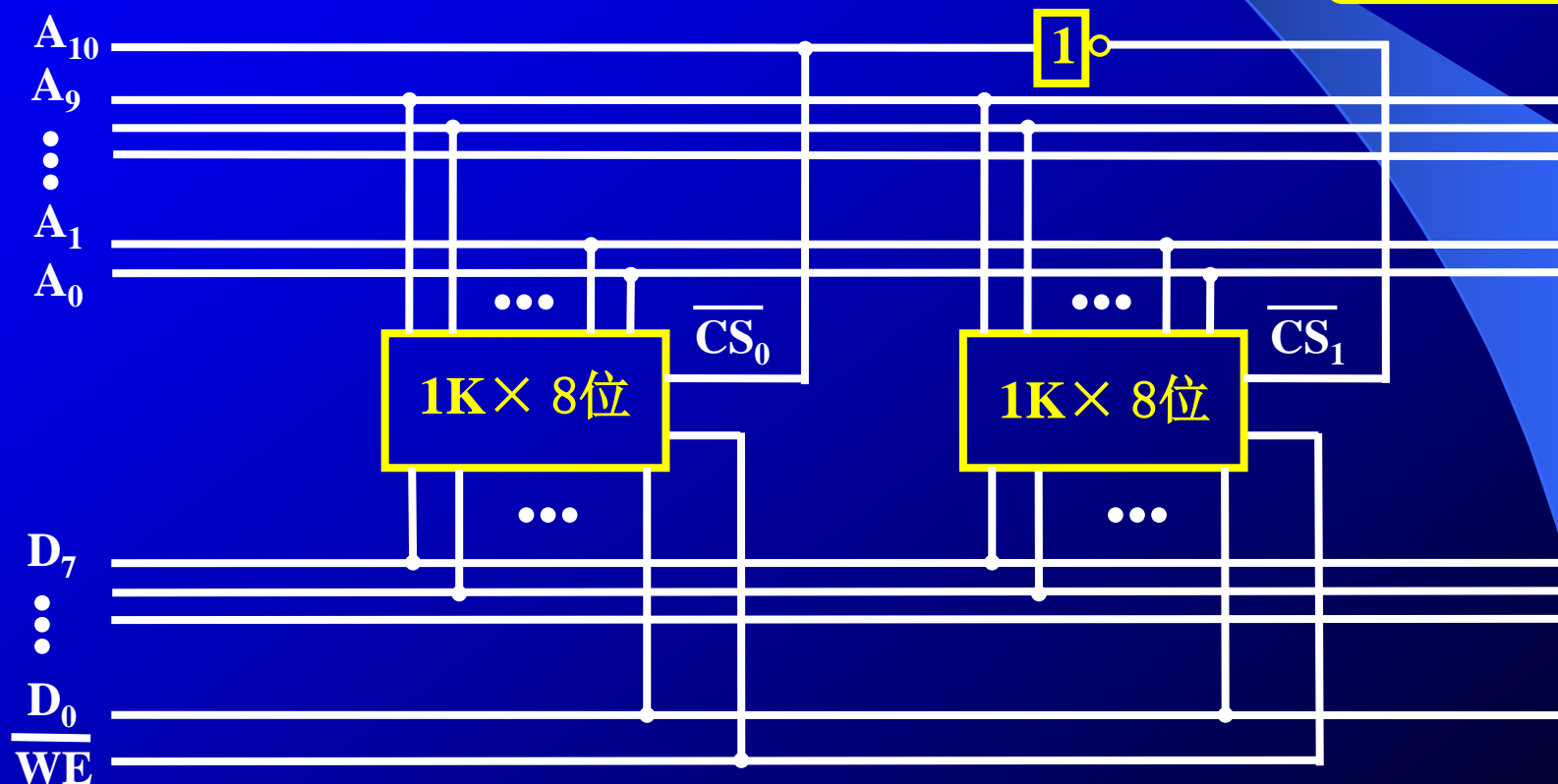
4.2

(2) 字扩展（增加存储字的数量）

11根地址线

用 2片 $1K \times 8$ 位 存储芯片组成 $2K \times 8$ 位的存储器

8根数据线



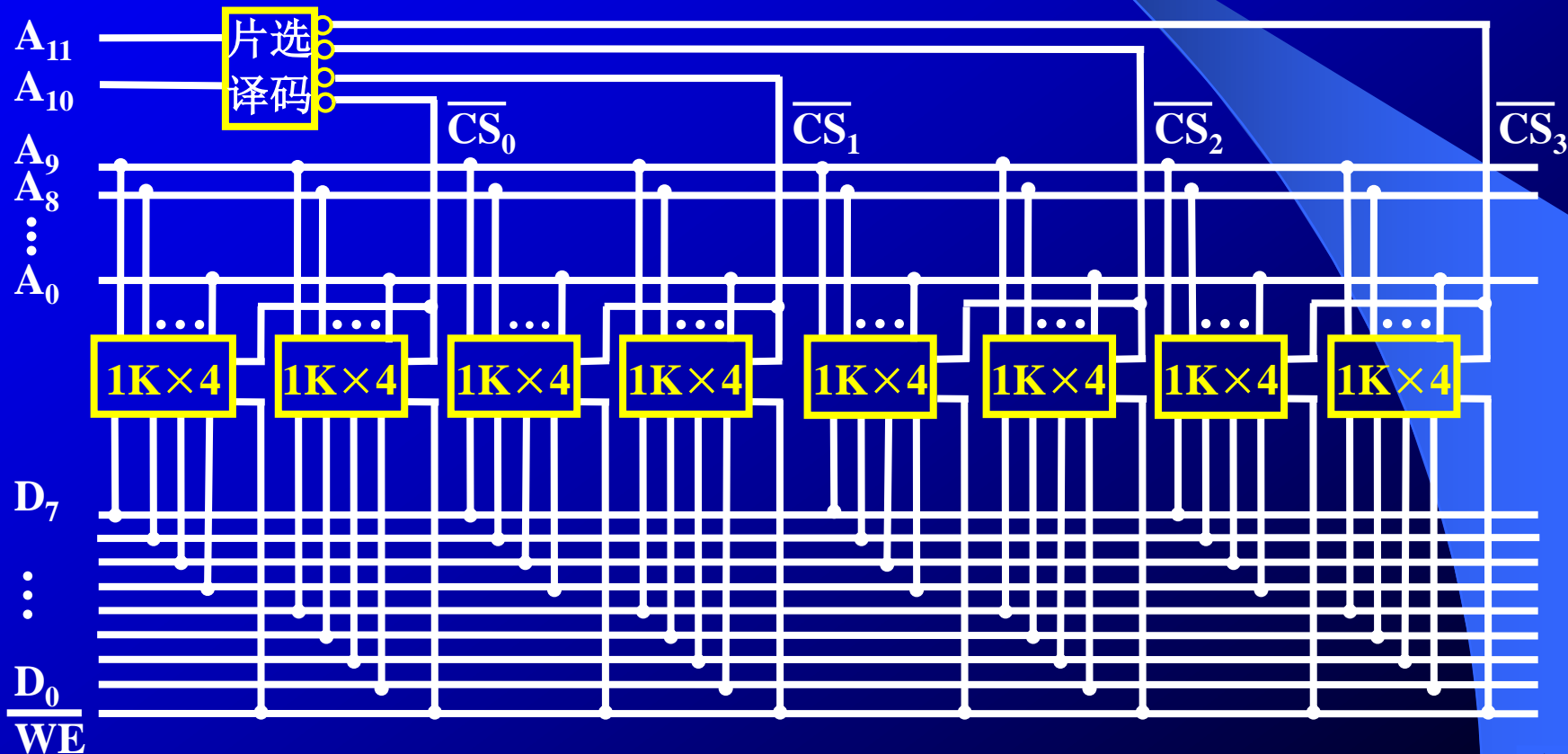
(3) 字、位扩展

4.2

用 8 片 $1\text{K} \times 4$ 位 存储芯片组成 $4\text{K} \times 8$ 位的存储器

12根地址线

8根数据线



2. 存储器与 CPU 的连接

- (1) 地址线的连接
- (2) 数据线的连接
- (3) 读/写命令线的连接
- (4) 片选线的连接
- (5) 合理选择存储芯片
- (6) 其他 时序、负载

例4.1 解:

4.2

(1) 写出对应的二进制地址码

A_{15}	A_{14}	A_{13}	A_{11}	A_{10}	...	A_7	...	A_4	A_3	...	A_0	
0	1	1	0	0	0	0	0	0	0	0	0	} ROM 2K×8位
⋮												
0	1	1	0	0	1	1	1	1	1	1	1	
0	1	1	0	1	0	0	0	0	0	0	0	} 1K×8位 RAM
⋮												
0	1	1	0	1	0	1	1	1	1	1	1	

1片 2K×8位

ROM

2K×8位

1K×8位

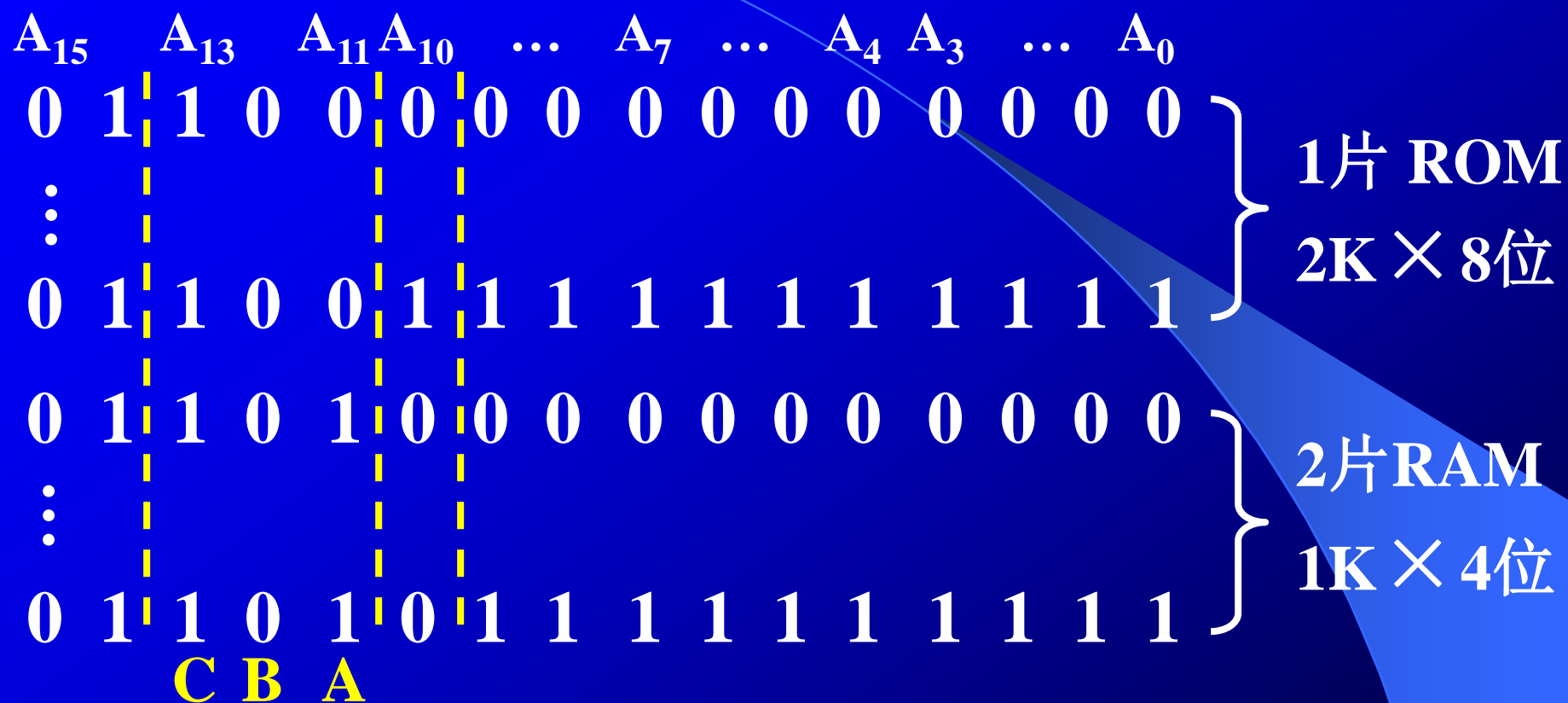
RAM

2片 1K×4位

(2) 确定芯片的数量及类型



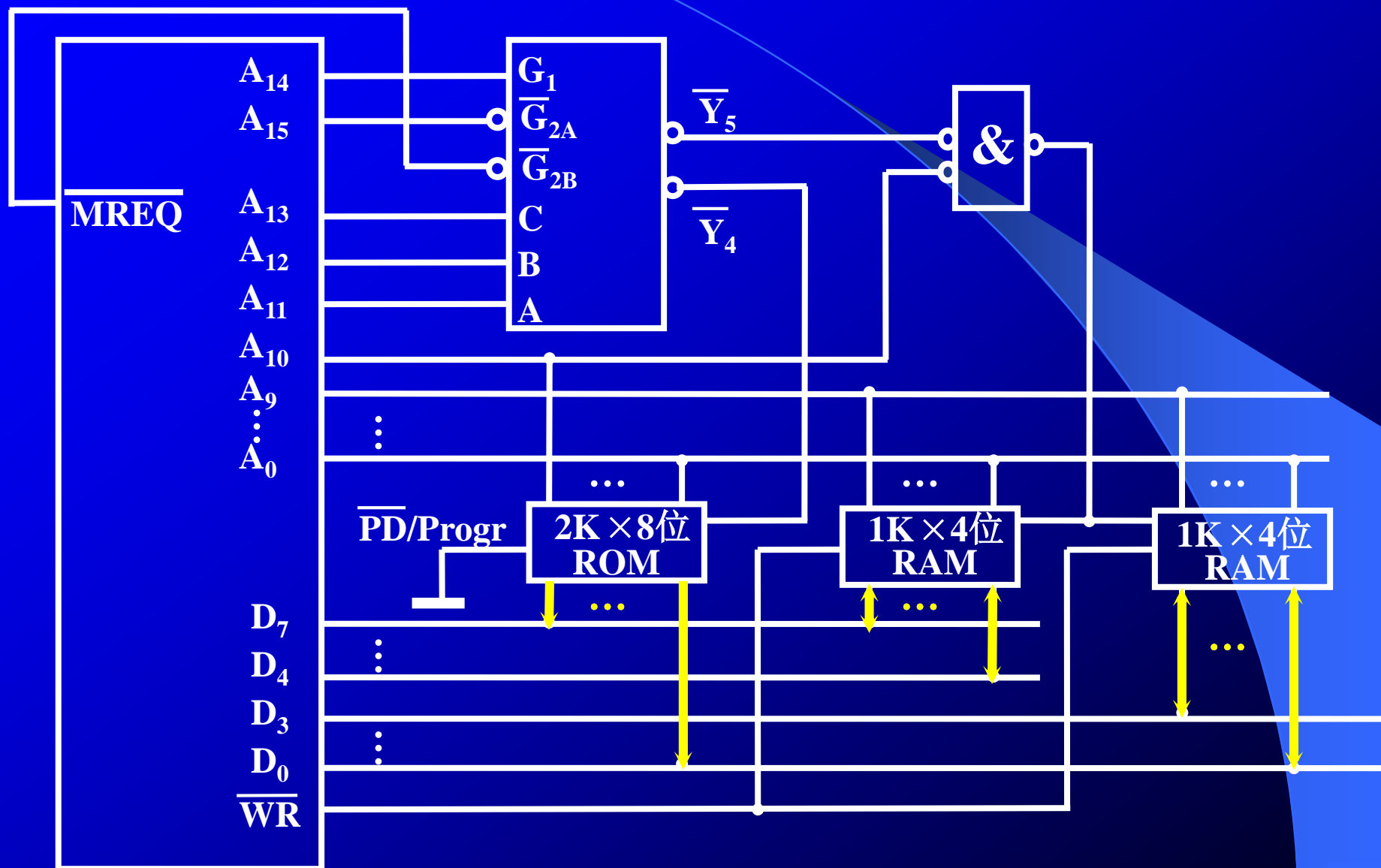
(3) 分配地址线



(4) 确定片选信号

例 4.1 CPU 与存储器的连接图

4.2



例4.2 假设同前，要求最小 4K为系统程序区，相邻 8K为用户程序区。

(1) 写出对应的二进制地址码

(2) 确定芯片的数量及类型

1片 4K×8位 ROM 2片 4K×8位 RAM

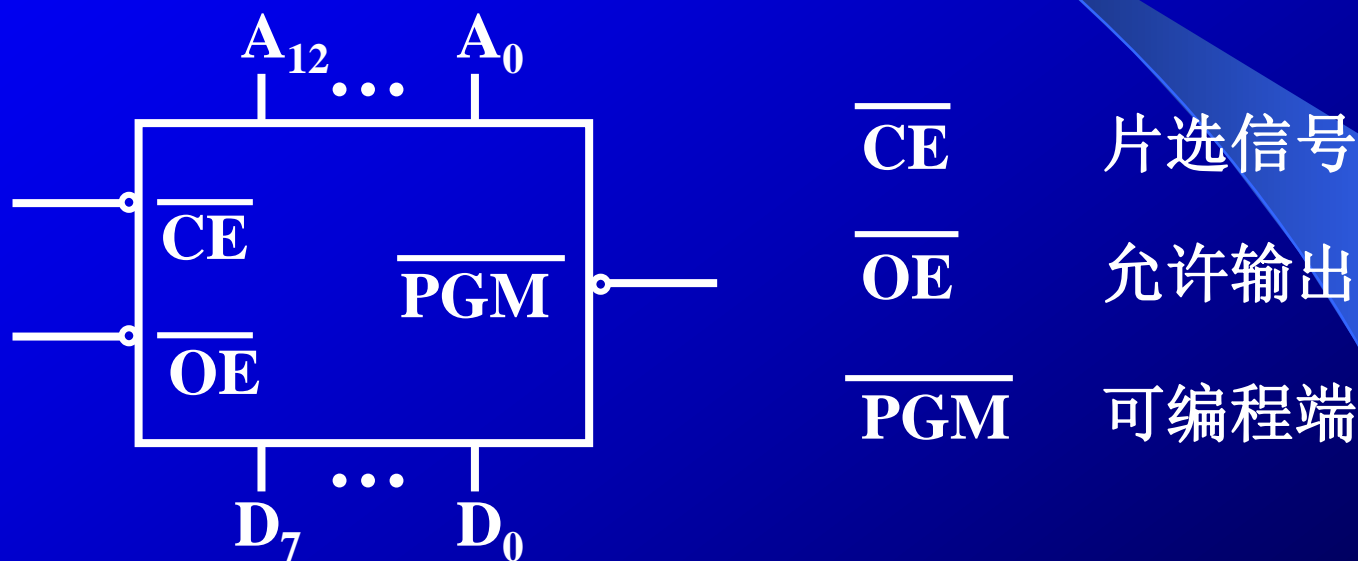
(3) 分配地址线

$A_{11} \sim A_0$ 接 ROM 和 RAM 的地址线

(4) 确定片选信号

例 4.3 设 CPU 有 20 根地址线, 8 根数据线。 **4.2**

并用 $\overline{\text{IO/M}}$ 作访存控制信号。 $\overline{\text{RD}}$ 为读命令,
 $\overline{\text{WR}}$ 为写命令。现有 2764 EPROM (8K × 8位),
外特性如下:



用 138 译码器及其他门电路 (门电路自定) 画出 CPU
和 2764 的连接图。要求地址为 F0000H~FFFFFFH, 并
写出每片 2764 的地址范围。

六、存储器的校验

4.2

0. 检错纠错的概念

在原有数据位之外增加一到几位校验位，使新的代码带有某种特征，之后通过检查该代码是否保持这一特征来判断是否出现了错误，甚至于定位并纠正错误，这就是检错纠错的编码技术。

- 奇偶校验码
- 汉明码
- 循环冗余码



1. 编码的最小距离

任意两组合法代码之间 二进制位数 的 最少差异
编码的纠错、检错能力与编码的最小距离有关

$$L - 1 = D + C \quad (D \geq C)$$

L —— 编码的最小距离 $L = 3$

D —— 检测错误的位数 具有 一位 纠错能力

C —— 纠正错误的位数

汉明码是具有一位纠错能力的编码

2. 汉明码的组成

组成汉明码的三要素

汉明码的组成需增添 ? 位检测位

$$2^k \geq n + k + 1$$

检测位的位置 ?

$$2^i \quad (i = 0, 1, 2, 3, \dots)$$

检测位的取值 ?

检测位的取值与该位所在的检测“小组”中承担的奇偶校验任务有关



各检测位 C_i 所承担的检测小组为

C_1 检测的 g_1 小组包含第 1, 3, 5, 7, 9, 11, ...

C_2 检测的 g_2 小组包含第 2, 3, 6, 7, 10, 11, ...

C_4 检测的 g_3 小组包含第 4, 5, 6, 7, 12, 13, ...

C_8 检测的 g_4 小组包含第 8, 9, 10, 11, 12, 13, 14, 15, 24, ...

g_i 小组独占第 2^{i-1} 位

g_i 和 g_j 小组共同占第 $2^{i-1} + 2^{j-1}$ 位

g_i 、 g_j 和 g_l 小组共同占第 $2^{i-1} + 2^{j-1} + 2^{l-1}$ 位

例4.4 求 0101 按“偶校验”配置的汉明码

解：∵ $n = 4$

根据 $2^k \geq n + k + 1$

得 $k = 3$

汉明码排序如下：

二进制序号	1	2	3	4	5	6	7
名称	C_1	C_2	0	C_4	1	0	1
	0	1		0			

∴ 0101 的汉明码为 **0100101**

练习1 按配偶原则配置 0011 的汉明码 4.2

解: $\because n = 4$ 根据 $2^k \geq n + k + 1$

取 $k = 3$

二进制序号	1	2	3	4	5	6	7
名称	C_1	C_2	0	C_4	0	1	1
	1	0		0			

$$C_1 = 3 \oplus 5 \oplus 7 = 1$$

$$C_2 = 3 \oplus 6 \oplus 7 = 0$$

$$C_4 = 5 \oplus 6 \oplus 7 = 0$$

\therefore 0011 的汉明码为 1000011

3. 汉明码的纠错过程

4.2

形成新的检测位 P_i ，其位数与增添的检测位有关，如增添 3 位 ($k=3$)，新的检测位为 $P_4 P_2 P_1$ 。

以 $k=3$ 为例， P_i 的取值为

$$P_1 = \overset{C_1}{1} \oplus 3 \oplus 5 \oplus 7$$

$$P_2 = \overset{C_2}{2} \oplus 3 \oplus 6 \oplus 7$$

$$P_4 = \overset{C_4}{4} \oplus 5 \oplus 6 \oplus 7$$

对于按“偶校验”配置的汉明码
不出错时 $P_1=0, P_2=0, P_4=0$

例4.5 已知接收到的汉明码为 0100111

(按配偶原则配置) 试问要求传送的信息是什么?

解: 纠错过程如下

$$P_1 = 1 \oplus 3 \oplus 5 \oplus 7 = 0 \quad \text{无错}$$

$$P_2 = 2 \oplus \underset{\checkmark}{3} \oplus \boxed{6} \oplus \underset{\checkmark}{7} = 1 \quad \text{有错}$$

$$P_4 = 4 \oplus \underset{\checkmark}{5} \oplus \boxed{6} \oplus \underset{\checkmark}{7} = 1 \quad \text{有错}$$

$$\therefore P_4 P_2 P_1 = 110$$

第 6 位出错, 可纠正为 0100101,
故要求传送的信息为 0101。

练习2 写出按偶校验配置的汉明码

0101101 的纠错过程

$$P_4 = 4 \oplus 5 \oplus 6 \oplus 7 = 1$$

$$P_2 = 2 \oplus 3 \oplus 6 \oplus 7 = 0$$

$$P_1 = 1 \oplus 3 \oplus 5 \oplus 7 = 0$$

$\therefore P_4 P_2 P_1 = 100$ 第4位错，可不纠

练习3 按配奇原则配置 0011 的汉明码

配奇的汉明码为 0101011

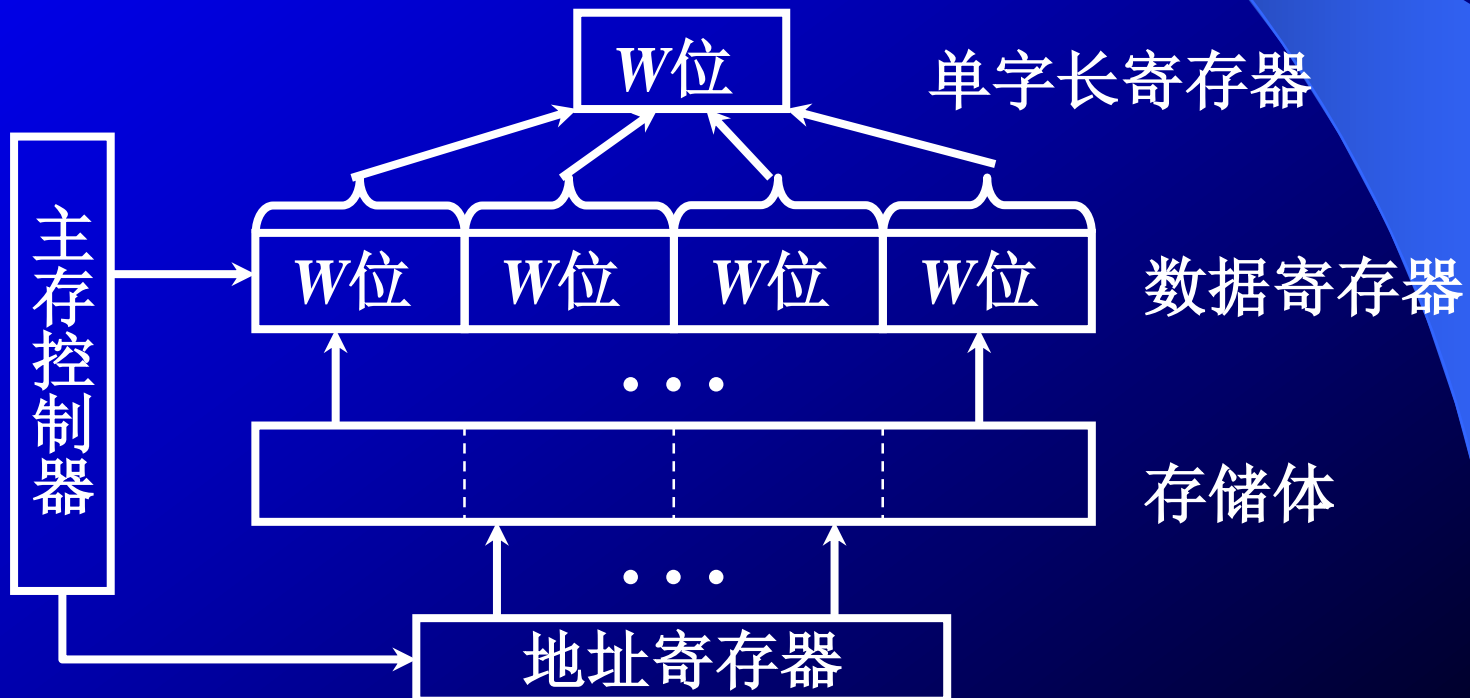
七、提高访存速度的措施

4.2

- 采用高速器件
- 采用层次结构 Cache – 主存
- 调整主存结构

1. 单体多字系统

增加存储器的带宽

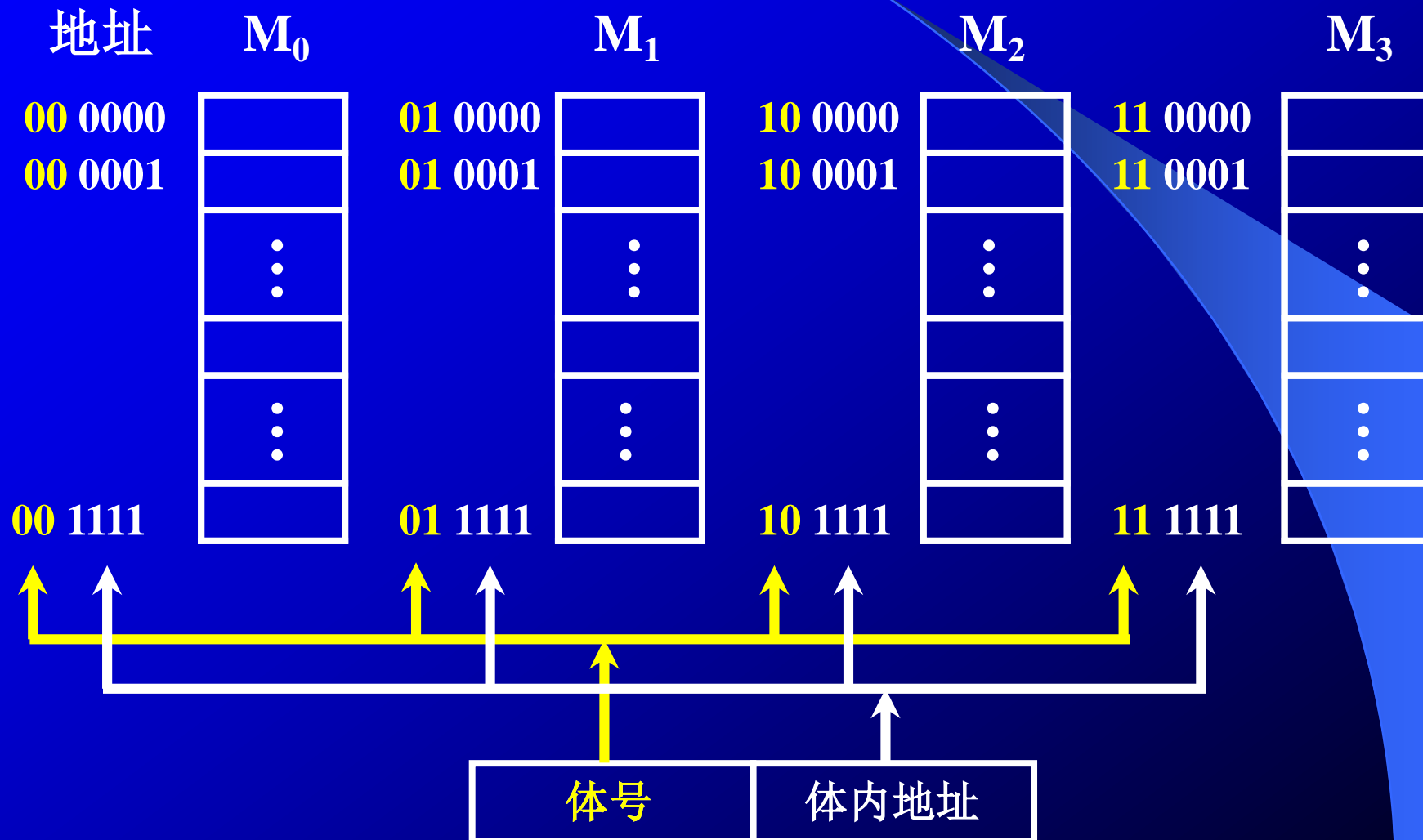


2. 多体并行系统

4.2

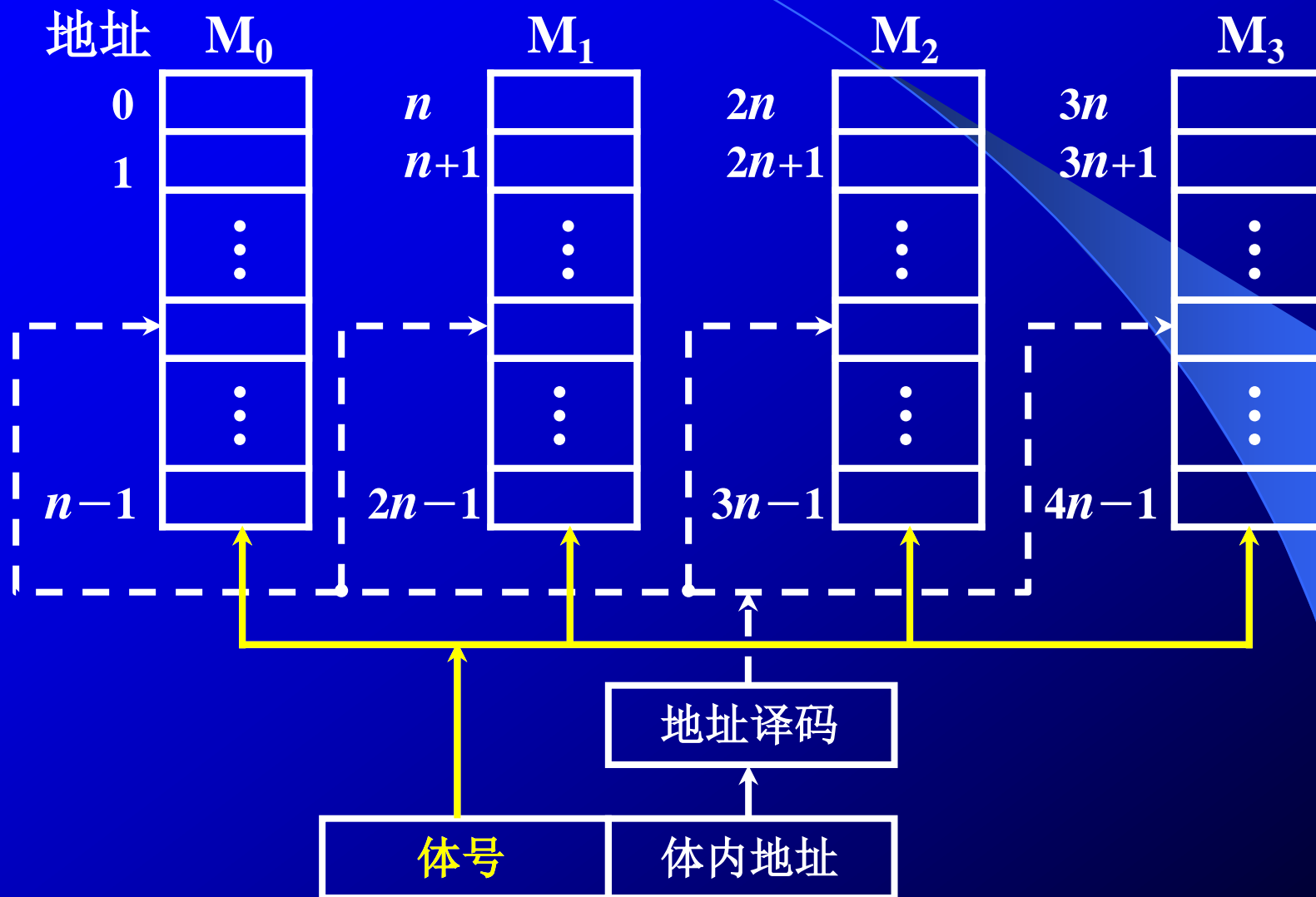
(1) 高位交叉

顺序编址



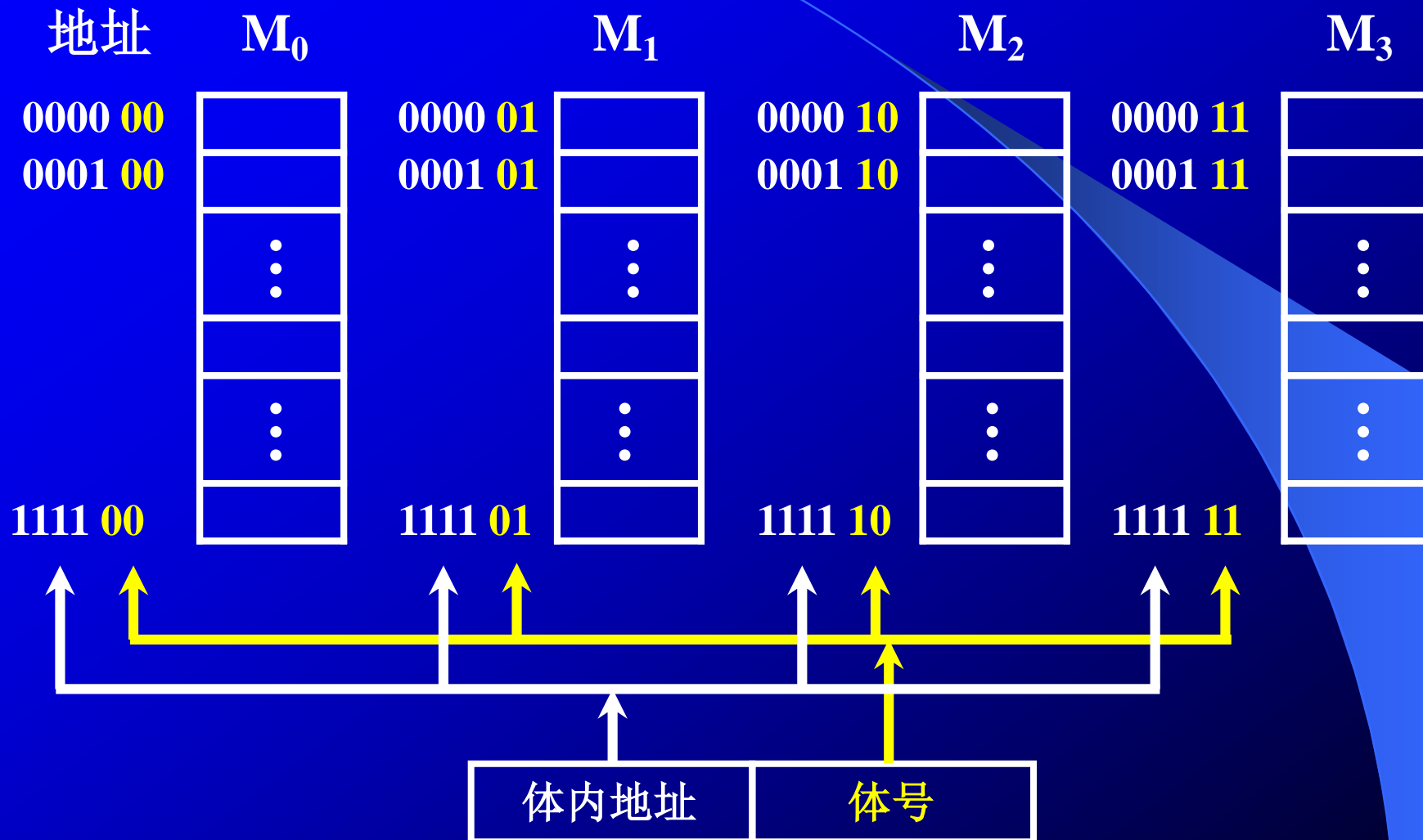
各个体并行工作

4.2



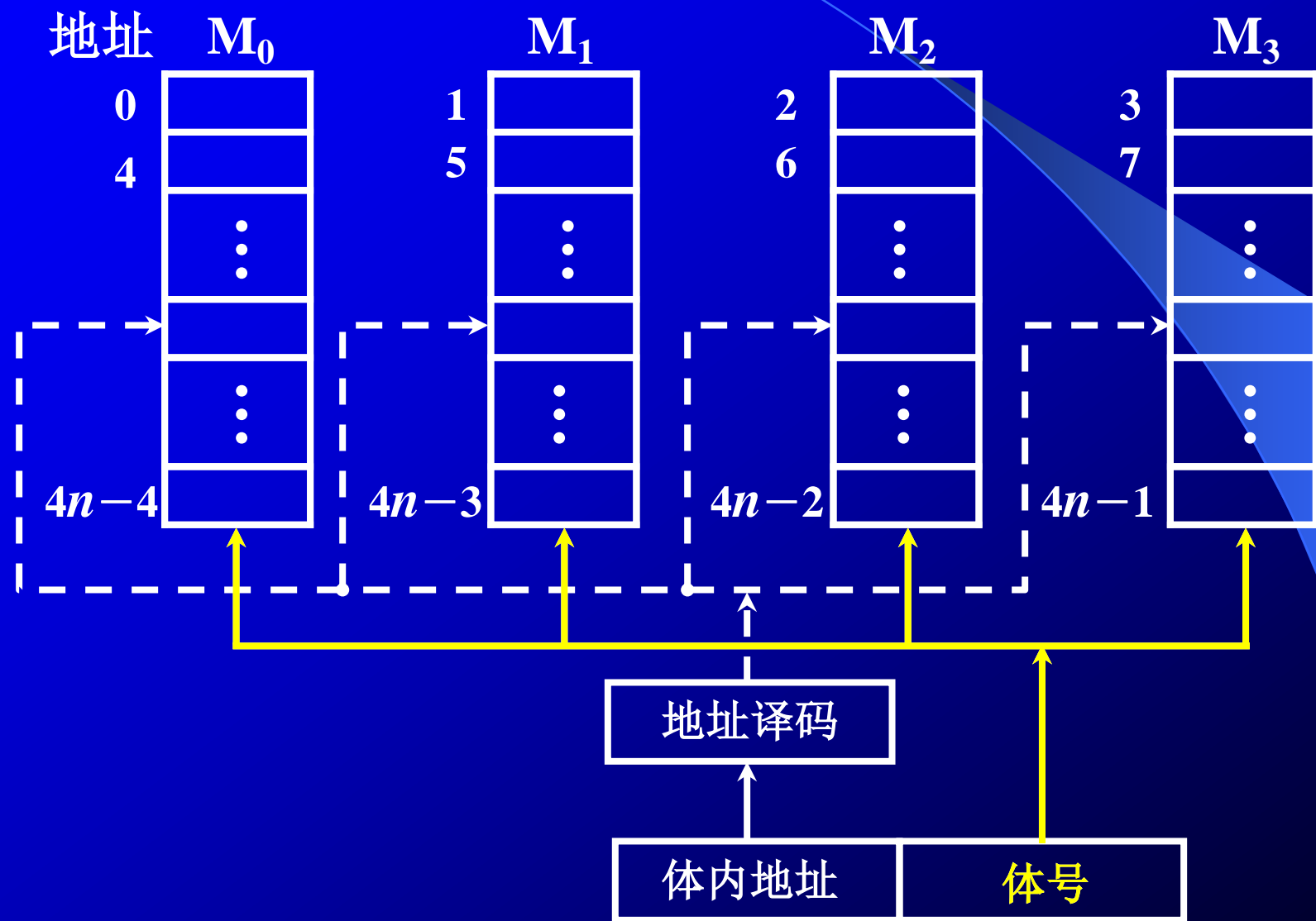
(2) 低位交叉 各个体轮流编址

4.2



(2) 低位交叉 各个体轮流编址

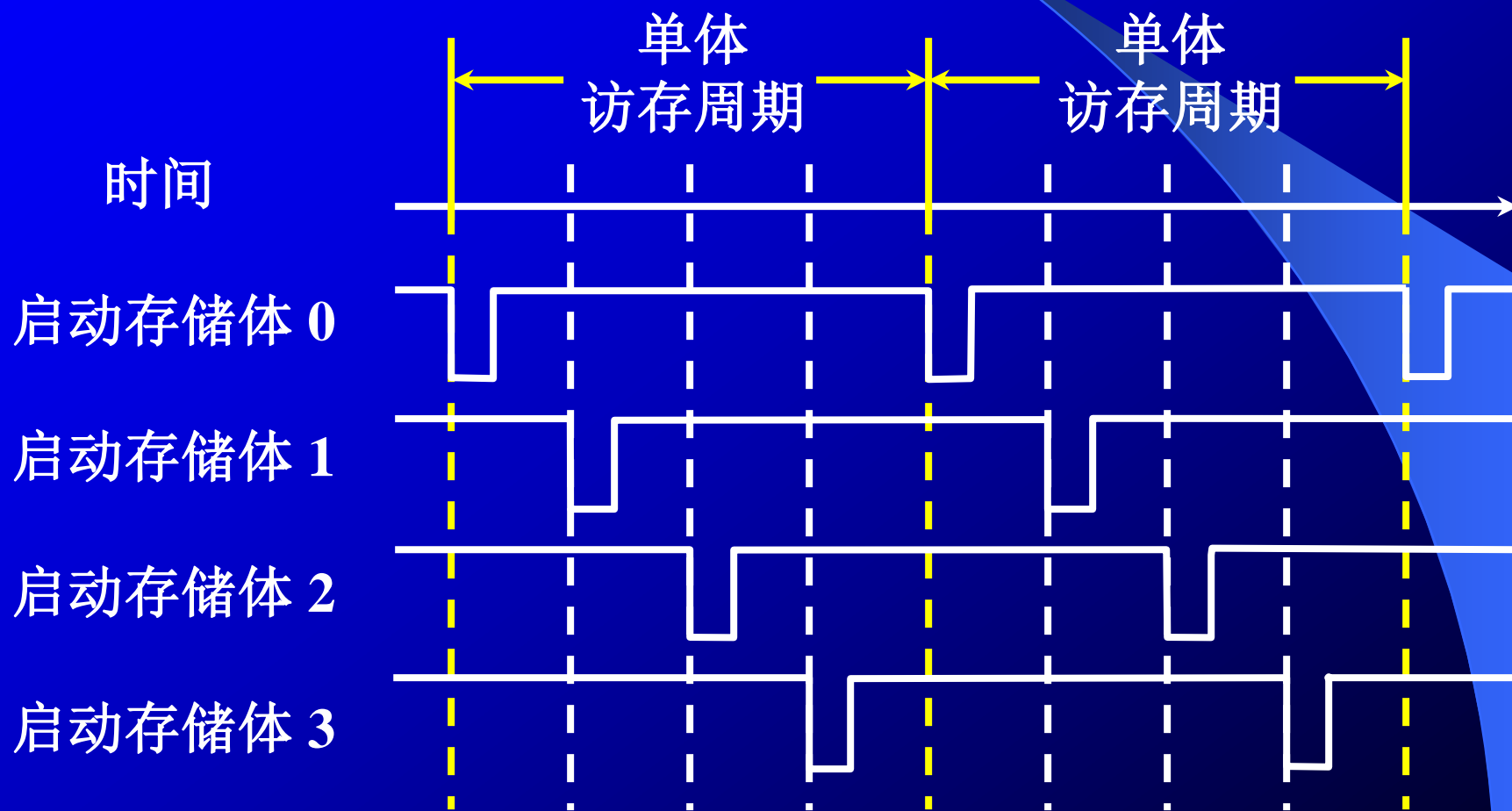
4.2



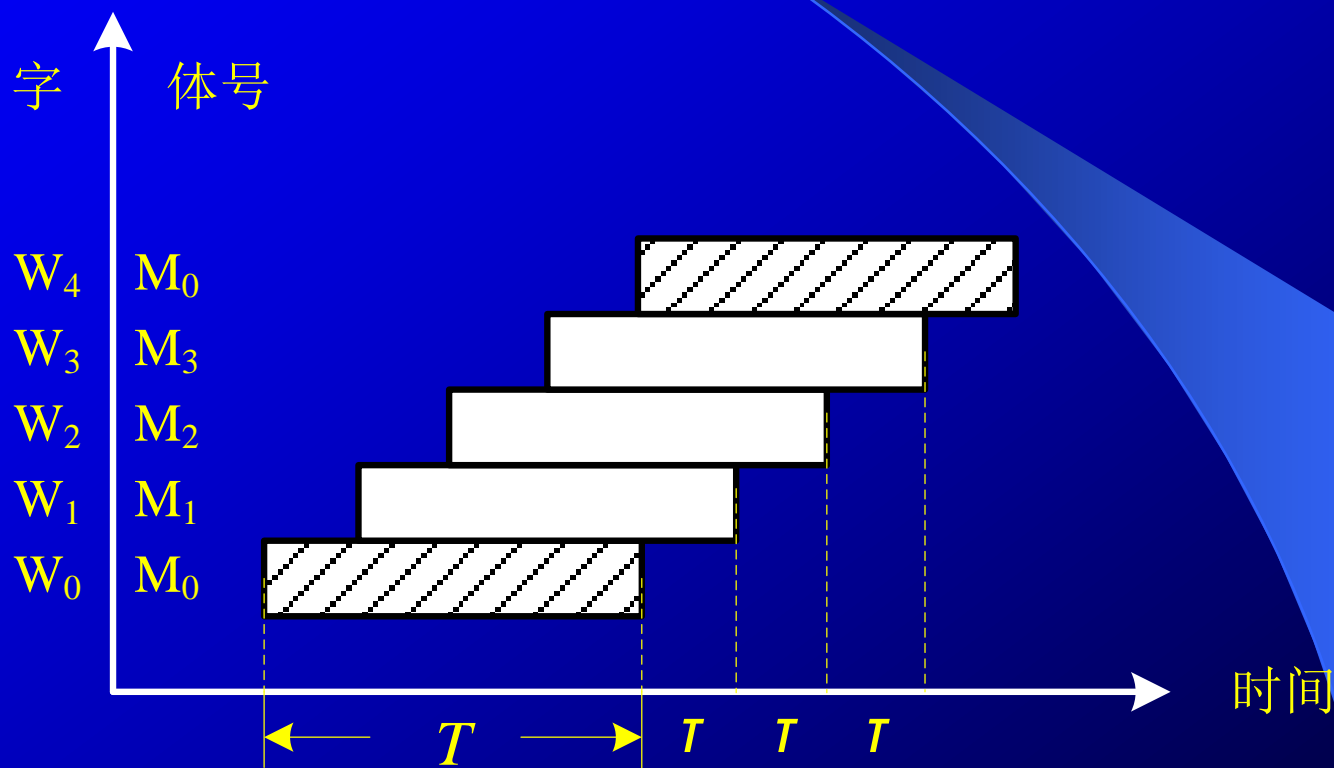
低位交叉的特点

4.2

在不改变存取周期的前提下，增加存储器的带宽

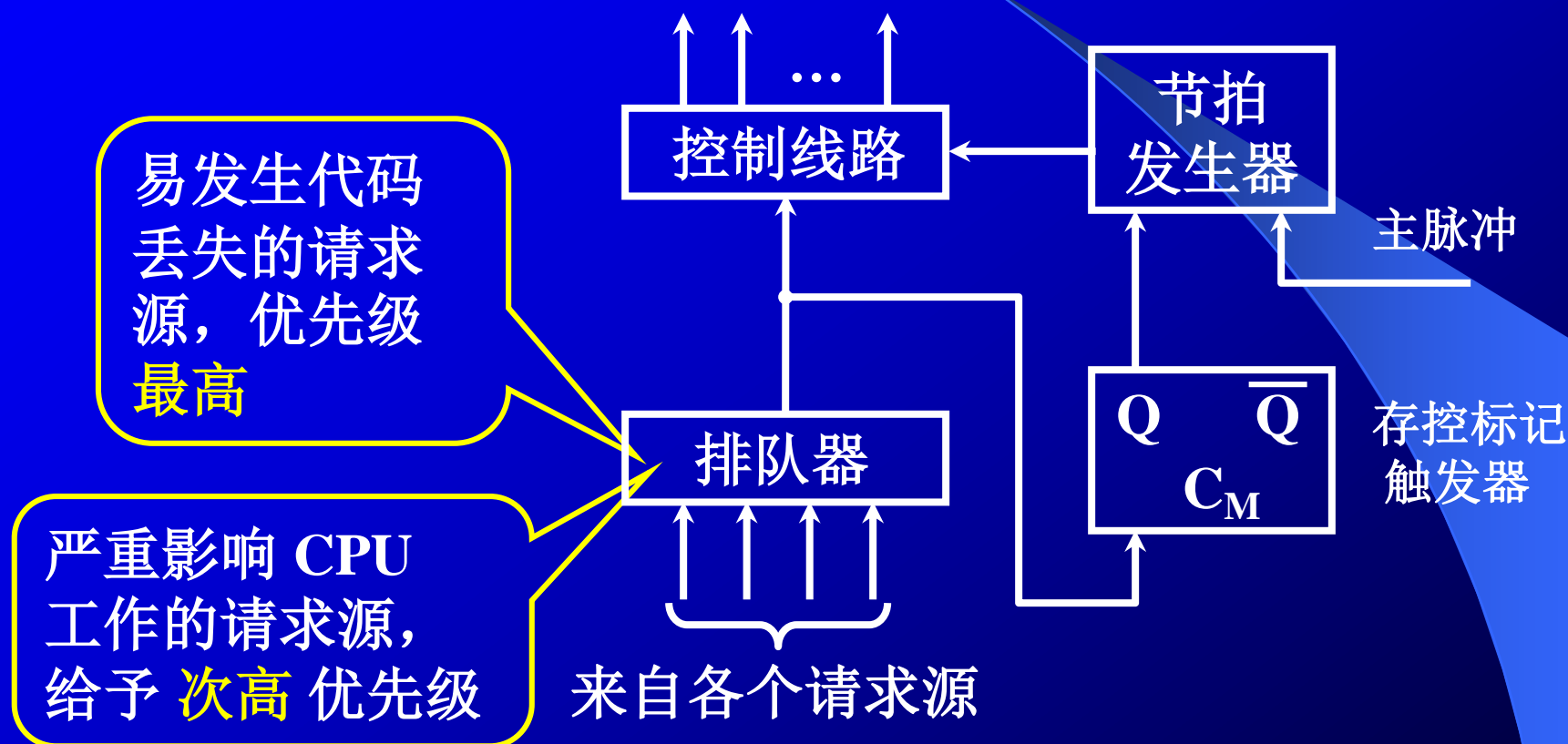


设四体低位交叉存储器，存取周期为 T ，总线传输周期为 τ ，为实现流水线方式存取，应满足 $T = 4\tau$ 。



连续读取 4 个字所需的时间为 $T + (4-1)\tau$

(3) 存储器控制部件（简称存控）



3.高性能存储芯片

4.2

(1) SDRAM (同步 DRAM)

在系统时钟的控制下进行读出和写入

CPU 无须等待

(2) RDRAM

由 **Rambus** 开发，主要解决 存储器带宽 问题

(3) 带 Cache 的 DRAM

在 DRAM 的芯片内 集成 了一个由 **SRAM** 组成的

Cache，有利于 猝发式读取



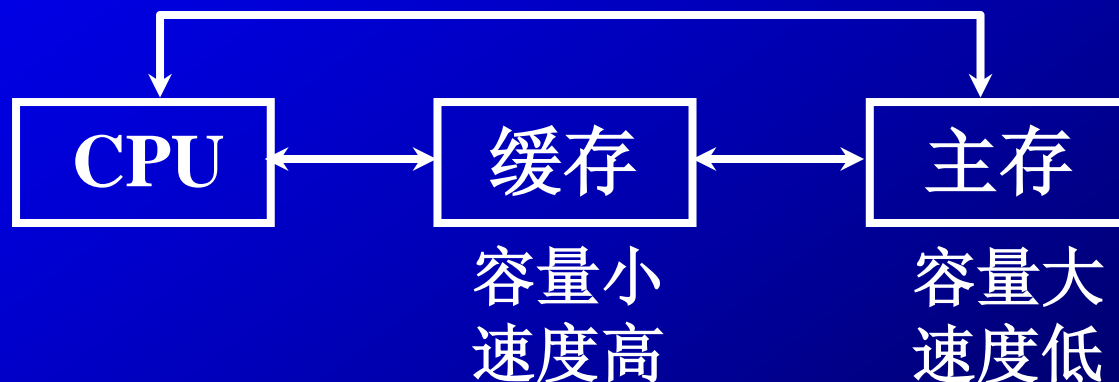
4.3 高速缓冲存储器

一、概述

1. 问题的提出

避免 CPU “空等” 现象

CPU 和主存（DRAM）的速度差异



程序访问的局部性原理

时间局部性：刚被访问过的单元很可能不久又被访问

空间局部性：刚被访问过的单元的邻近单元很可能被访问

程序段A:

```
int sumarrayrows(int A[M][N])
{
    int i, j, sum=0;
    for (i=0; i<M, i++)
        for (j=0; j<N, j++) sum+=A[i][j];
    return sum;
}
```

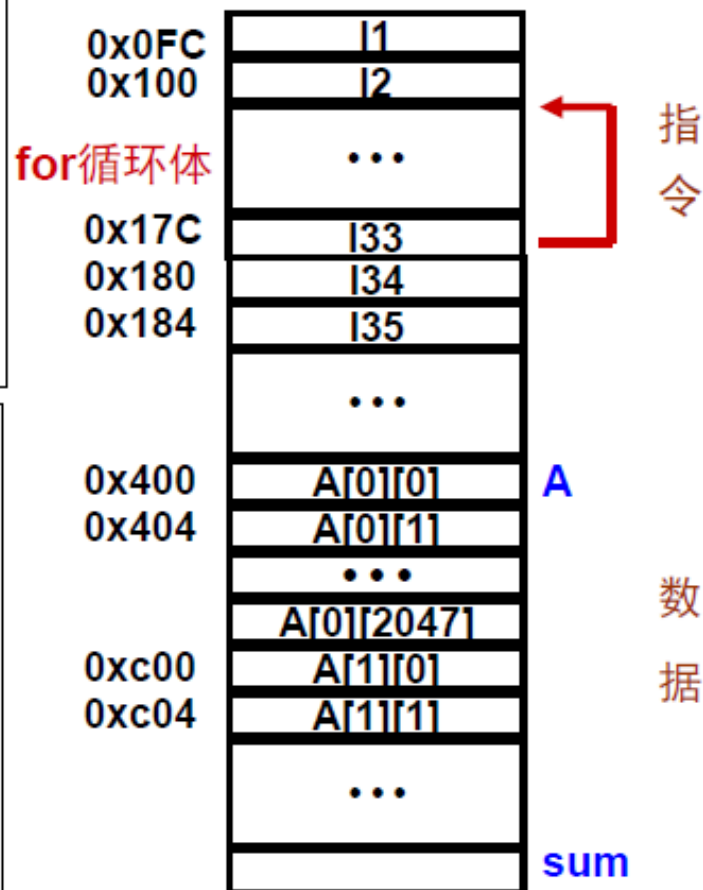
访问顺序与存储
顺序一致

程序段B:

```
int sumarraycols(int A[M][N])
{
    int i, j, sum=0;
    for (j=0; j<N, j++)
        for (i=0; i<M, i++) sum+=A[i][j];
    return sum;
}
```

访问顺序与存储
顺序不一致

M=N=2048时主存的布局:



假定数组在存储器中按行优先顺序存放

实际运行结果(2GHz Intel Pentium 4):

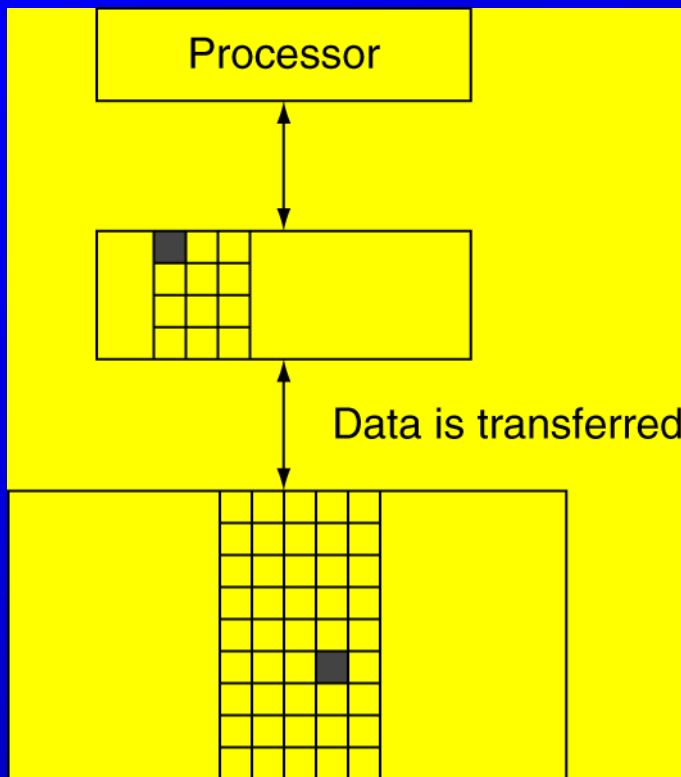
程序A: 59,393,288 时钟周期

程序B: 1,277,877,876 时钟周期

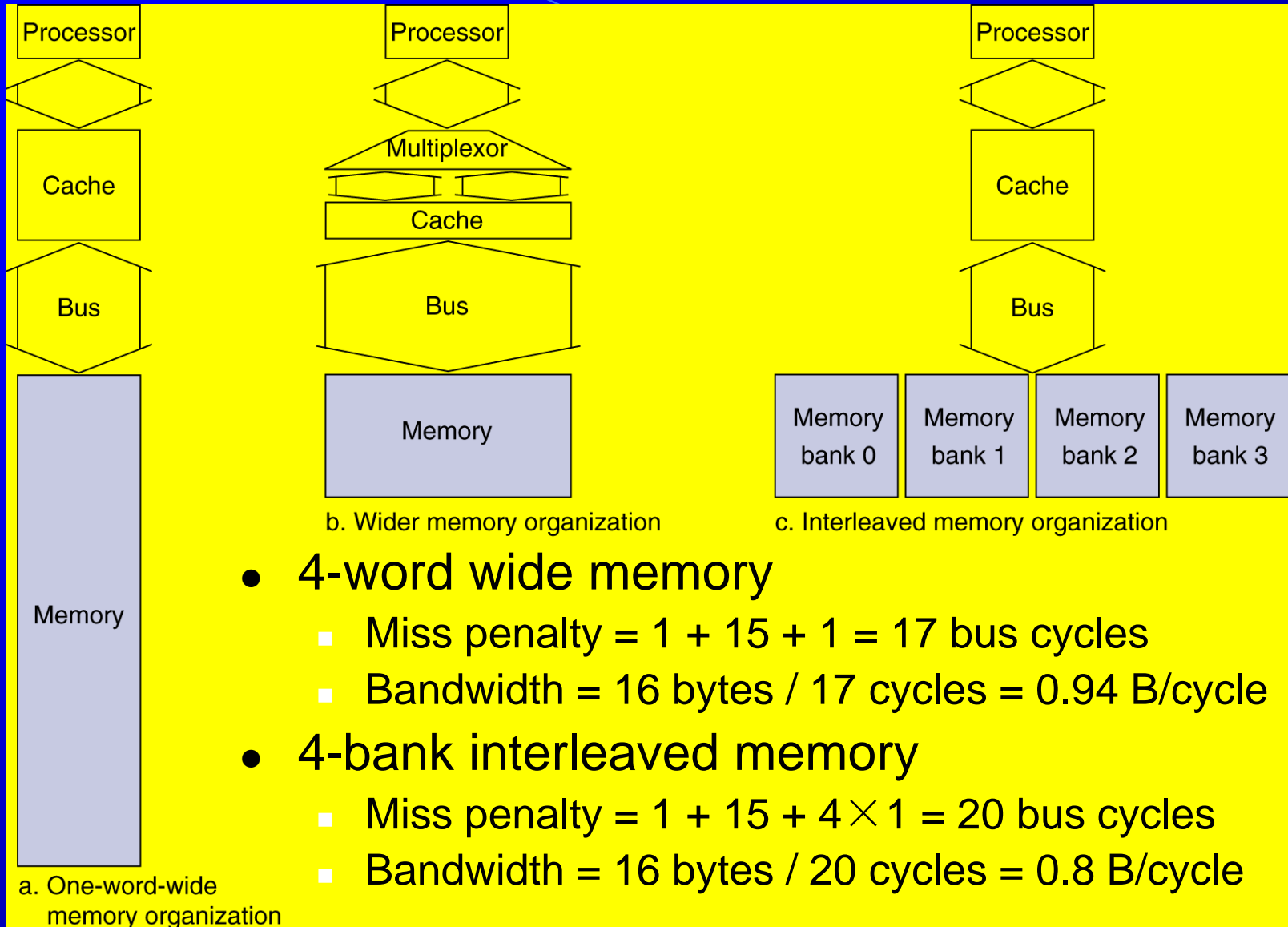
程序A比程序B快
21.5 倍!!

存储层次

- Block (aka line): unit of copying
 - May be multiple words
- If accessed data is present in upper level
 - Hit: access satisfied by upper level
 - Hit ratio: hits/accesses
- If accessed data is absent
 - Miss: block copied from lower level
 - Time taken: miss penalty
 - Miss ratio: misses/accesses
 $= 1 - \text{hit ratio}$
 - Then accessed data supplied from upper level



增加存储器带宽

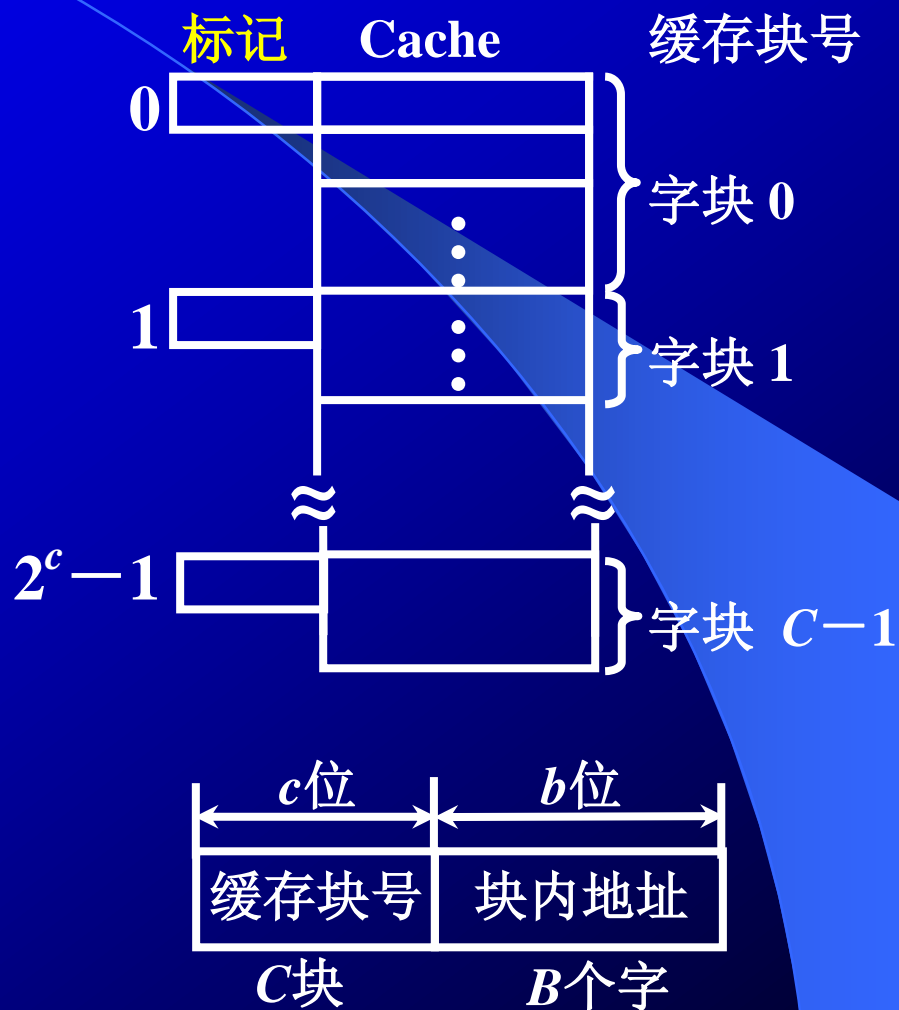
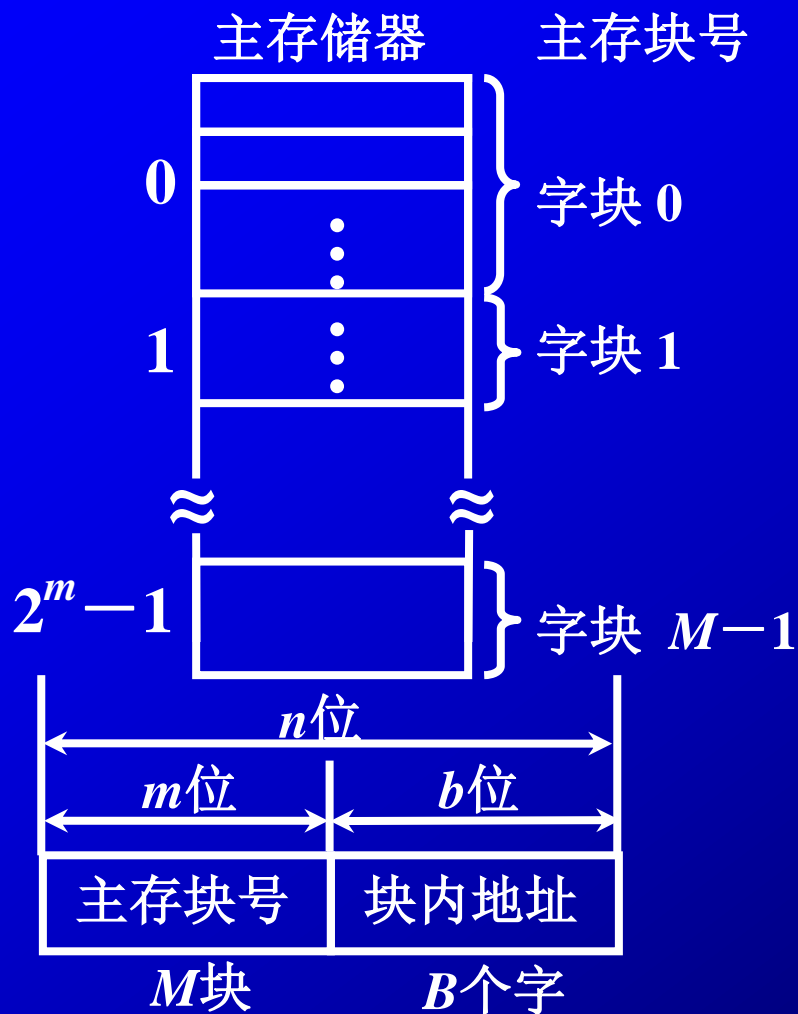


- 4-word wide memory
 - Miss penalty = $1 + 15 + 1 = 17$ bus cycles
 - Bandwidth = $16 \text{ bytes} / 17 \text{ cycles} = 0.94 \text{ B/cycle}$
- 4-bank interleaved memory
 - Miss penalty = $1 + 15 + 4 \times 1 = 20$ bus cycles
 - Bandwidth = $16 \text{ bytes} / 20 \text{ cycles} = 0.8 \text{ B/cycle}$

2. Cache 的工作原理

4.3

(1) 主存和缓存的编址



主存和缓存按块存储

块的大小相同

B 为块长

(2) 命中与未命中

缓存共有 C 块

主存共有 M 块 $M \gg C$

命中 主存块 **调入** 缓存

主存块与缓存块 **建立** 了对应关系

用 **标记记录** 与某缓存块建立了对应关系的 **主存块号**

未命中 主存块 **未调入** 缓存

主存块与缓存块 **未建立** 对应关系

(3) Cache 的命中率

CPU 欲访问的信息在 Cache 中的 **比率**

命中率 与 Cache 的 **容量** 与 **块长** 有关

一般每块可取 4 ~ 8 个字

块长 取一个存取周期内从主存调出的信息长度

CRAY_1	16体交叉	块长取 16 个存储字
IBM 370/168	4体交叉	块长取 4 个存储字 (64位 \times 4=256位)

(3) 块大小的考虑

- Larger blocks should reduce miss rate
 - Due to spatial locality
- But in a fixed-sized cache
 - Larger blocks \Rightarrow fewer of them
 - More competition \Rightarrow increased miss rate
 - Larger blocks \Rightarrow pollution
- Larger miss penalty
 - Can override benefit of reduced miss rate
 - Early restart and critical-word-first can help

(4) Cache –主存系统的效率

效率 e 与 命中率 有关

$$e = \frac{\text{访问 Cache 的时间}}{\text{平均访问时间}} \times 100\%$$

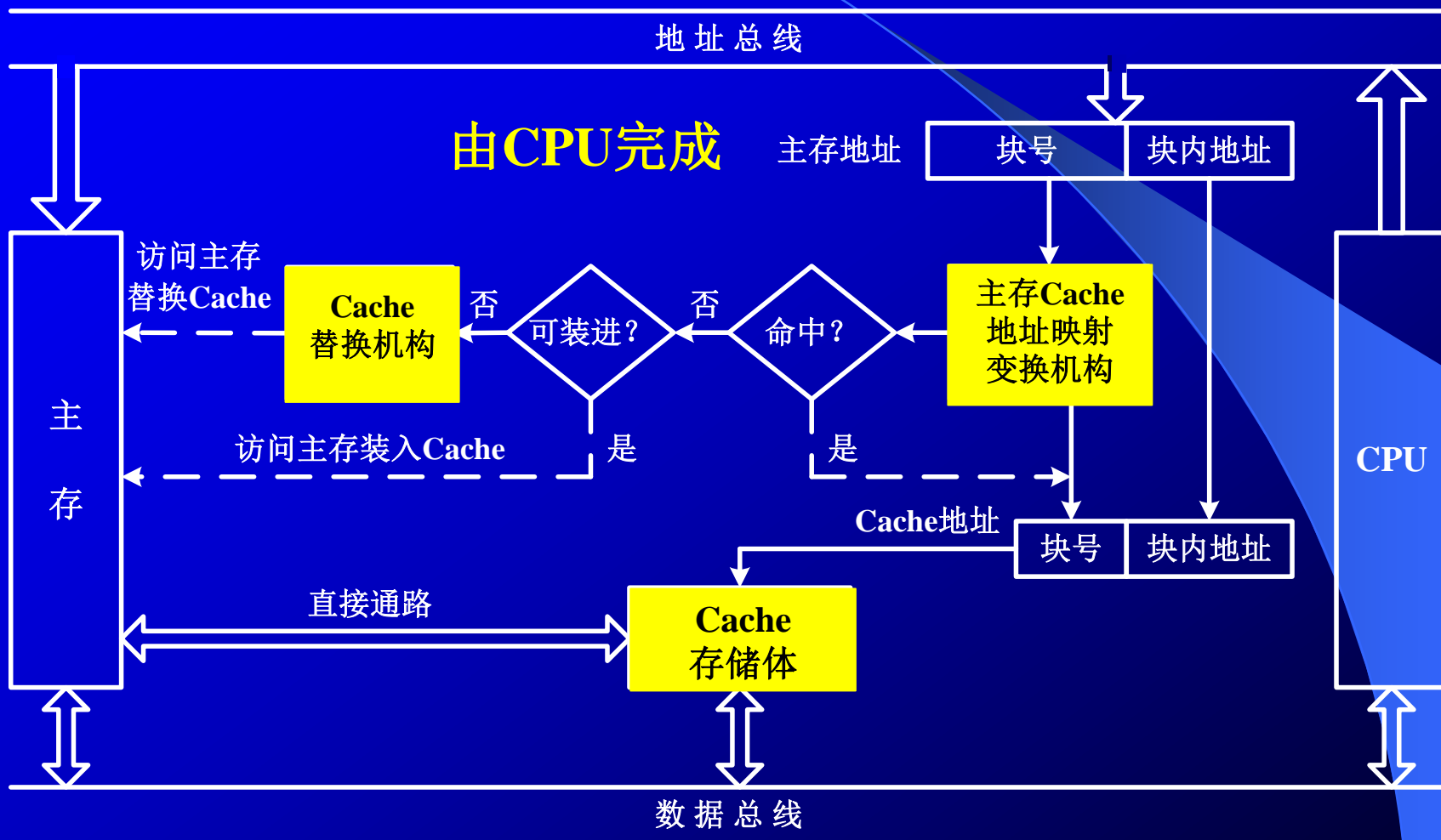
设 Cache 命中率为 h ，访问 Cache 的时间为 t_c ，
访问 主存 的时间为 t_m

$$\text{则 } e = \frac{t_c}{h \times t_c + (1-h) \times t_m} \times 100\%$$



3. Cache 的基本结构

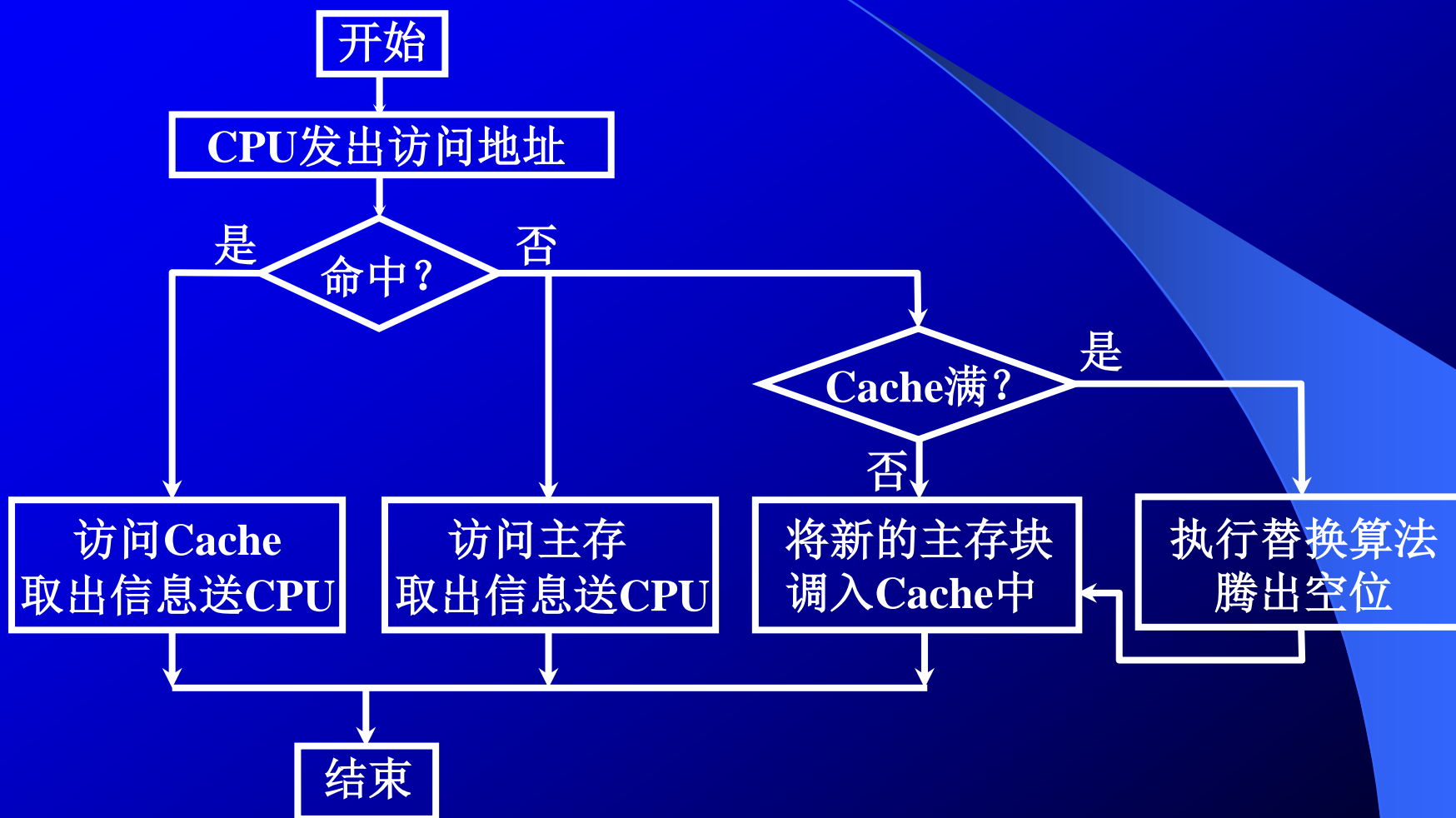
4.3



4. Cache 的读写操作

读

4.3



4. Cache 的 读写 操作

4.3

写 Cache 和主存的一致性

- 写直达法 (Write-through)

写操作时数据既写入Cache又写入主存

写操作时间就是访问主存的时间，读操作时不涉及对主存的写操作，更新策略比较容易实现

- 写回法 (Write-back)

写操作时只把数据写入 Cache 而不写入主存
当 Cache 数据被替换出去时才写回主存

写操作时间就是访问 Cache 的时间，

读操作 Cache 失效发生数据替换时，

被替换的块需写回主存，增加了 Cache 的复杂性



写直达(Write-Through)

- On data-write hit, could just update the block in cache
 - But then cache and memory would be inconsistent
- Write through: also update memory
- But makes writes take longer
 - e.g., if base CPI = 1, 10% of instructions are stores, write to memory takes 100 cycles
 - Effective CPI = $1 + 0.1 \times 100 = 11$
- Solution: write buffer
 - Holds data waiting to be written to memory
 - CPU continues immediately
 - Only stalls on write if write buffer is already full

写回 (Write-Back)

- Alternative: On data-write hit, just update the block in cache
 - Keep track of whether each block is dirty
- When a dirty block is replaced
 - Write it back to memory
 - Can use a write buffer to allow replacing block to be read first

写分配 (Write Allocation)

- What should happen on a write miss?
- Alternatives for write-through
 - Allocate on miss: fetch the block
 - Write around: don't fetch the block
 - Since programs often write a whole block before reading it (e.g., initialization)
- For write-back
 - Usually fetch the block

5. Cache 的改进

4.3

(1) 增加 Cache 的级数

片载（片内）Cache

片外 Cache

(2) 统一缓存和分立缓存

指令 Cache 数据 Cache

与主存结构有关

与指令执行的控制方式有关 是否流水

Pentium	8K 指令 Cache	8K 数据 Cache
---------	-------------	-------------

PowerPC620	32K 指令 Cache	32K 数据 Cache
------------	--------------	--------------

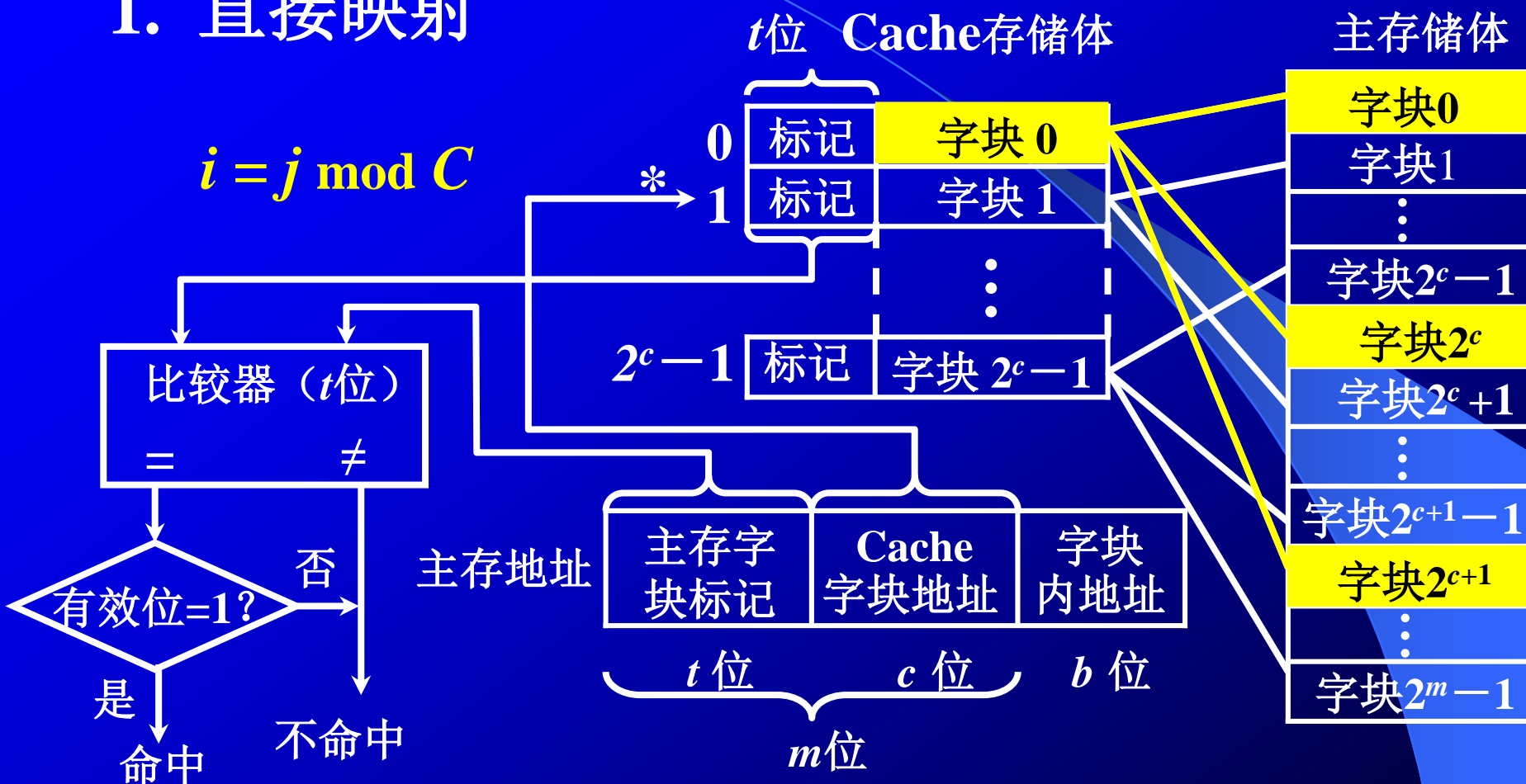


二、Cache – 主存的地址映射

4.3

1. 直接映射

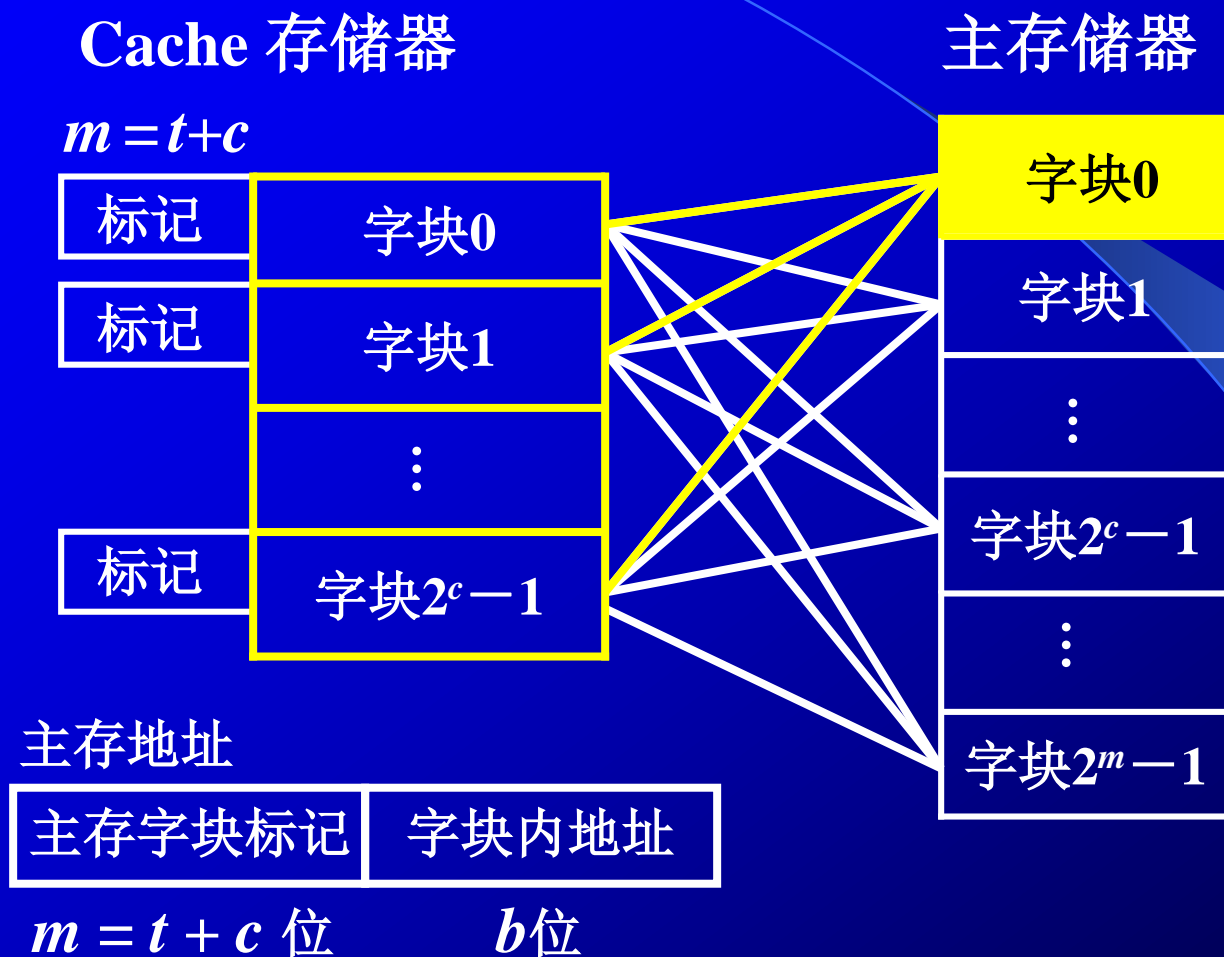
$$i = j \bmod C$$



每个缓存块 i 可以和若干个主存块对应
每个主存块 j 只能和一个缓存块对应

2. 全相联映射

4.3



主存 中的 任一 块 可以映射到 缓存 中的 任一 块

3. 组相联映射

4.3

主存储器

组 Cache 共 Q 组，每组内两块 ($r=1$)

0	标记	字块 0	标记	字块 1
1	标记	字块 2	标记	字块 3
	⋮	⋮	⋮	⋮
$2^{c-r}-1$	标记	字块 2^c-2	标记	字块 2^c-1

主存地址

主存字块标记	组地址	字块内地址
$s = t + r$ 位	$q = c - r$ 位	b 位
m 位		

字块0
字块1
⋮
字块 $2^{c-r}-1$
字块 2^{c-r}
字块 $2^{c-r}+1$
⋮
字块 $2^{c-r}+1$
⋮
字块 2^m-1

$$i = j \bmod Q$$

直接映联映射

某一主存块 j 按模 Q 映射到 缓存 的第 i 组中的 任一块

三、替换算法

1. 先进先出（FIFO）算法

2. 近期最少使用（LRU）算法

小结

成本与活

直接 某一主存块只能固定映射到某一缓存块

全相联 某一主存块能映射到任一缓存块

组相联 某一主存块只能映射到某一缓存组中的任一块

4.4 辅助存储器

一、概述

1. 特点 不直接与 CPU 交换信息（主存-辅存）

2. 磁表面存储器的技术指标

(1) 记录密度 道密度 D_t 位密度 D_b

(2) 存储容量 $C = n \times k \times s$

(3) 平均寻址时间 寻道时间 + 等待时间

辅存的速度 $\left\{ \begin{array}{l} \text{寻址时间} \\ \text{磁头读写时间} \end{array} \right.$

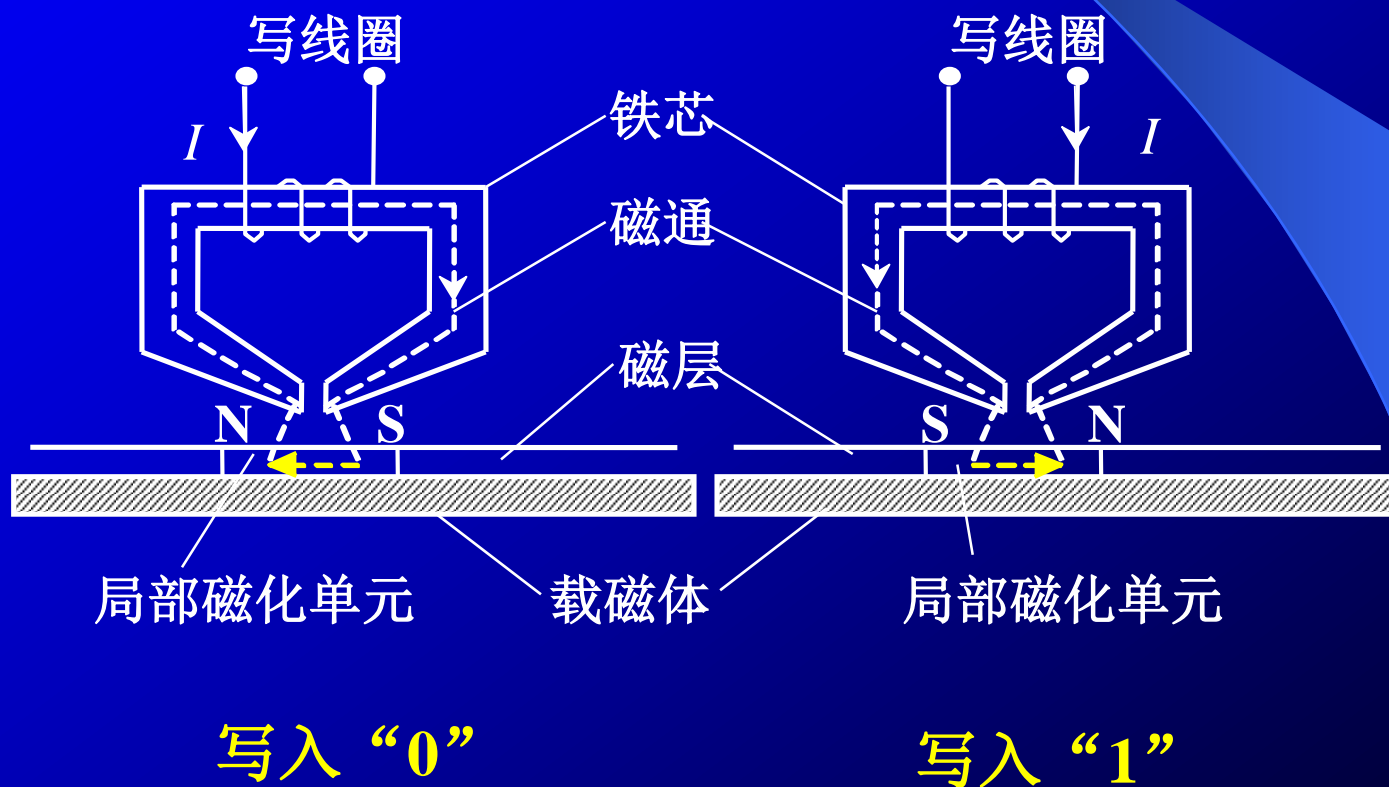
(4) 数据传输率 $D_r = D_b \times V$

(5) 误码率 出错信息位数与读出信息的总位数之比



二、磁记录原理和记录方式

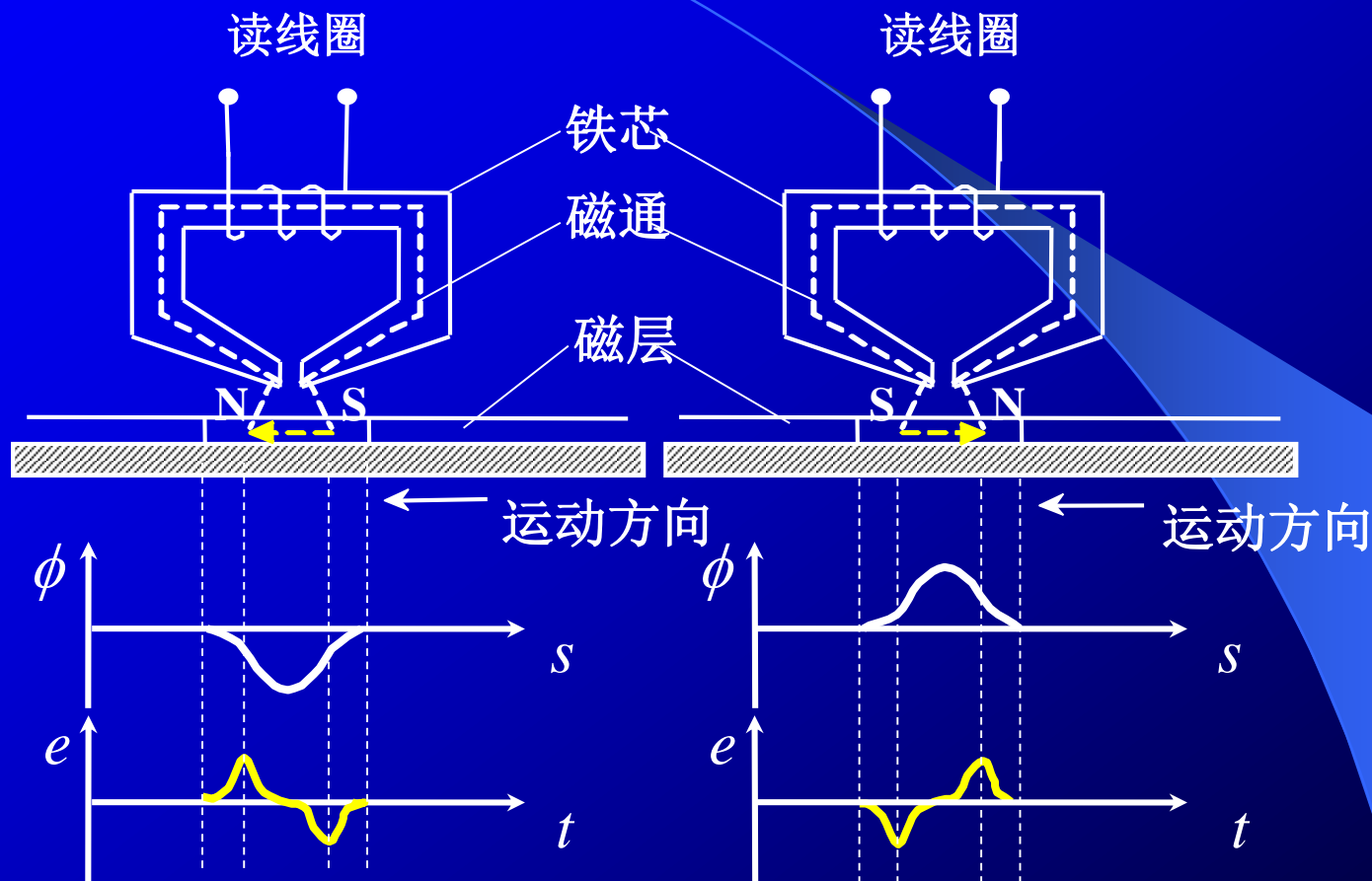
1. 磁记录原理 (磁头和记录介质的相对运动) 写



1. 磁记录原理

4.4

读

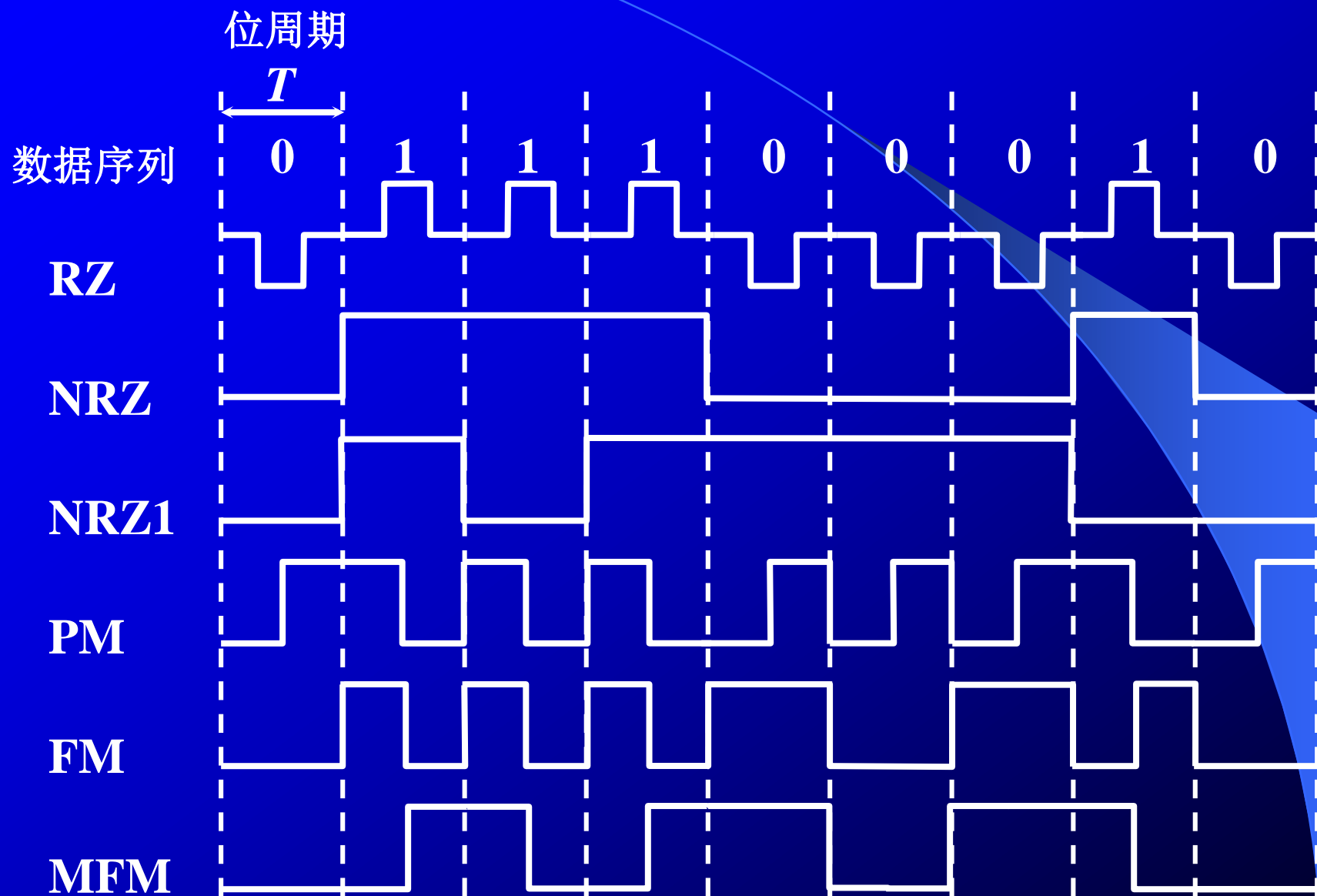


读出“0”

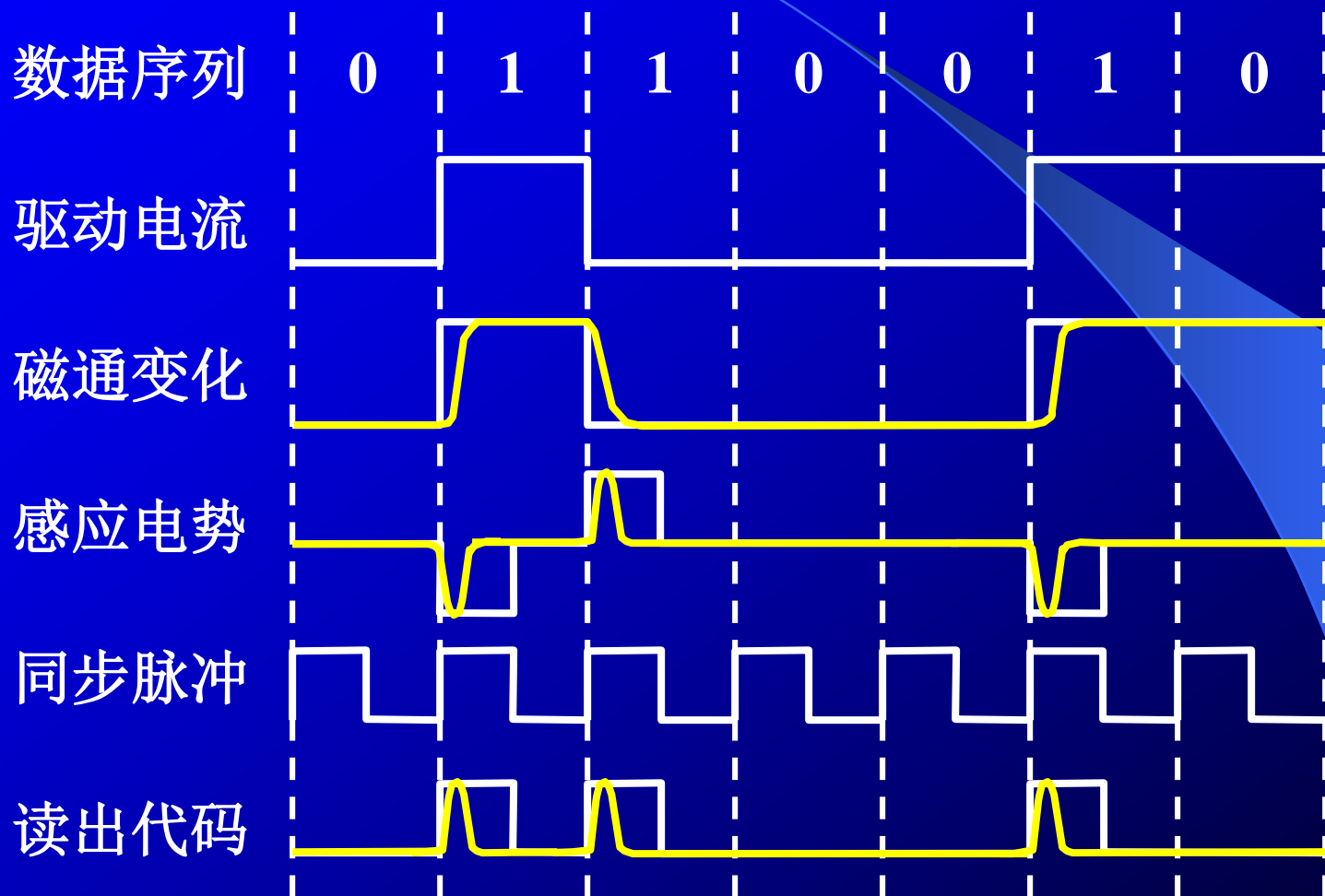
读出“1”



2. 磁表面存储器的记录方式 (编码方式) 4.4



例 NRZ1 的读出代码波形



三、硬磁盘存储器

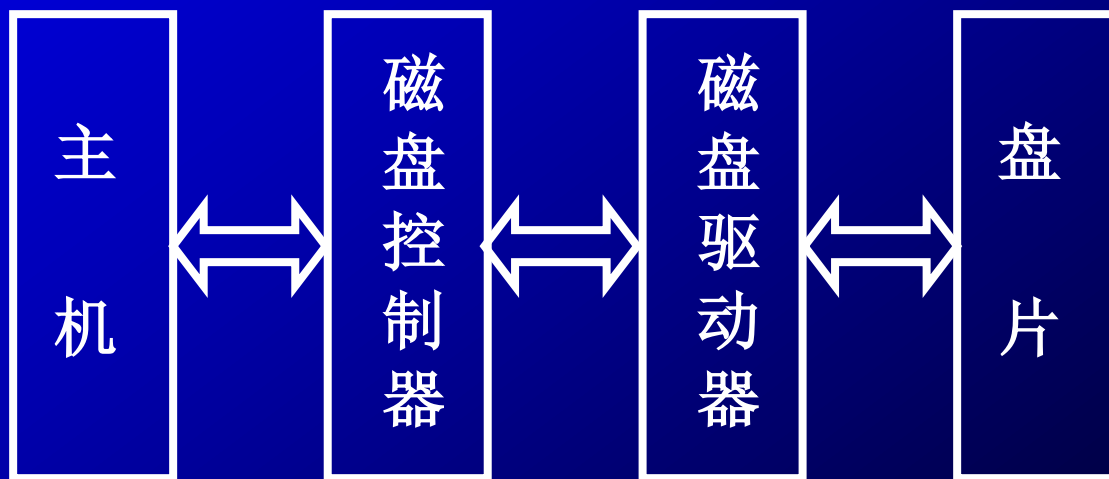
4.4

1. 硬磁盘存储器的类型

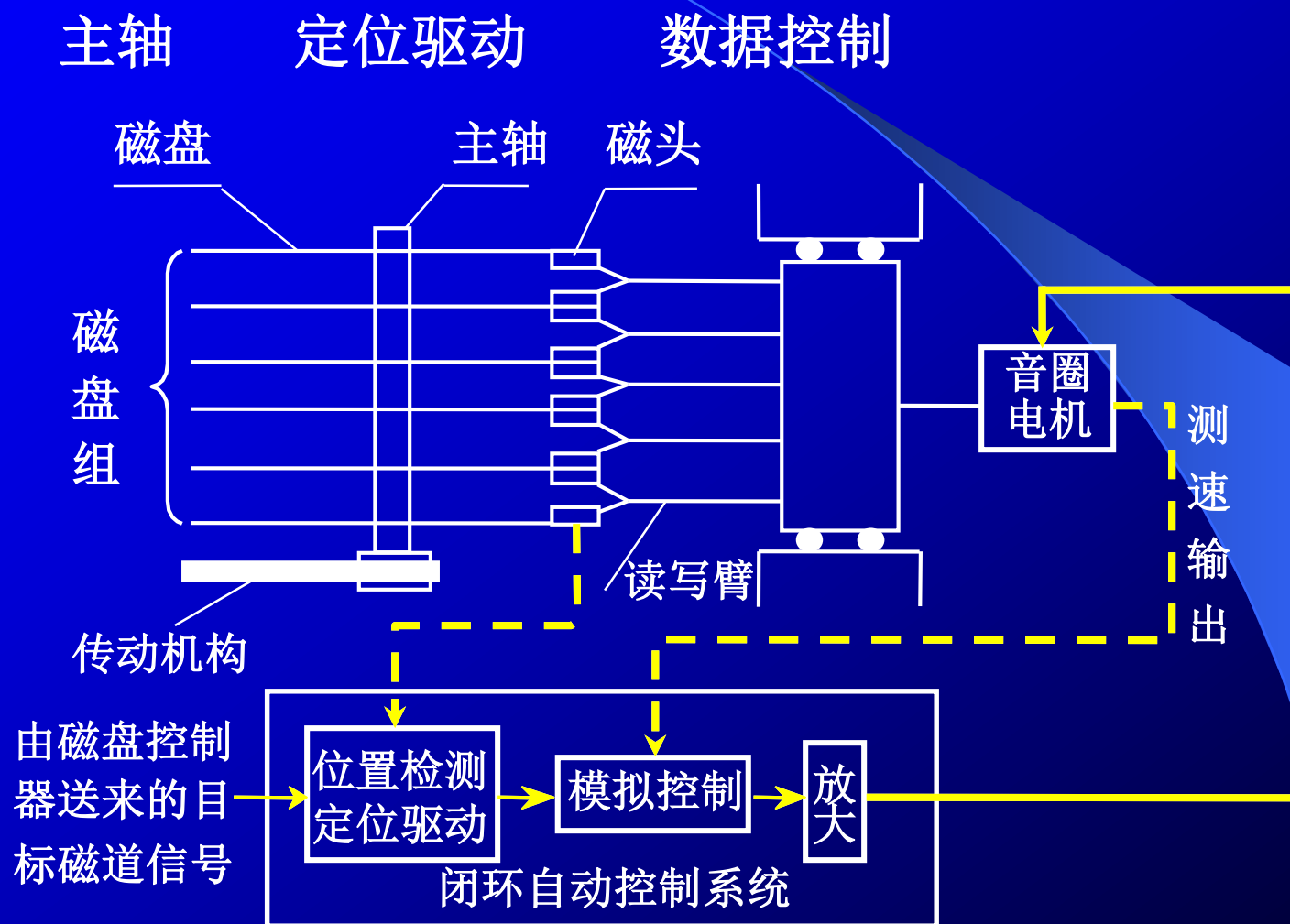
(1) 固定磁头和移动磁头

(2) 可换盘和固定盘

2. 硬磁盘存储器结构



(1) 磁盘驱动器



(2) 磁盘控制器

- 接收主机发来的命令，转换成磁盘驱动器的控制命令
- 实现主机和驱动器之间的数据格式转换
- 控制磁盘驱动器读写

磁盘控制器 是

主机与磁盘驱动器之间的 接口 { 对主机 通过总线
对硬盘 (设备)

(3) 盘片

由硬质铝合金材料制成