# 计算机组成原理

## 第十三讲

刘松波

哈工大计算学部

模式识别与智能系统研究中心

# 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

(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

非易失

## 2. 按存取方式分类

**(1) 存取时间与物理地址无关（随机访问）**

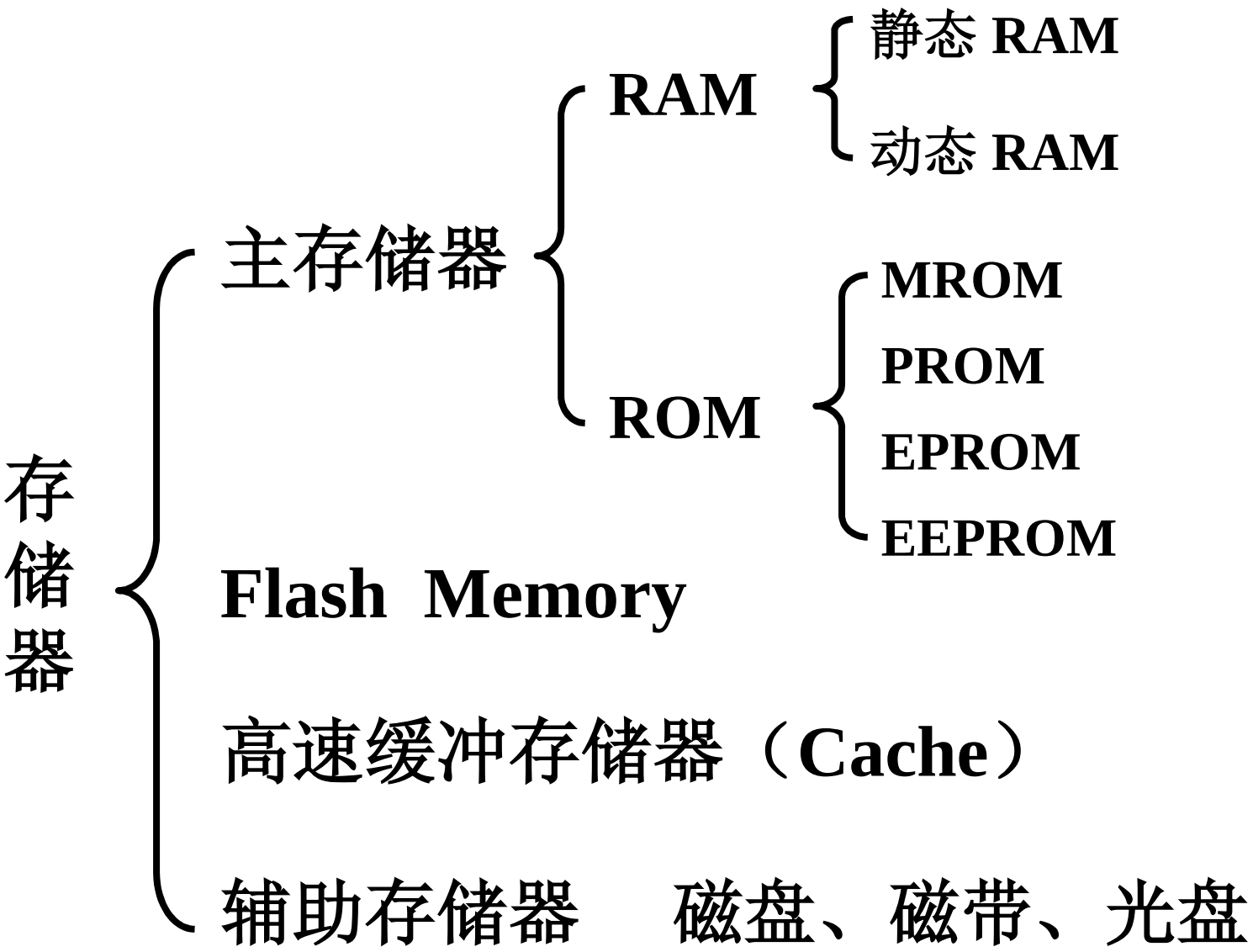
- 随机存储器　　在程序的执行过程中 可 读 可 写
- 只读存储器　　在程序的执行过程中 只 读

**(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

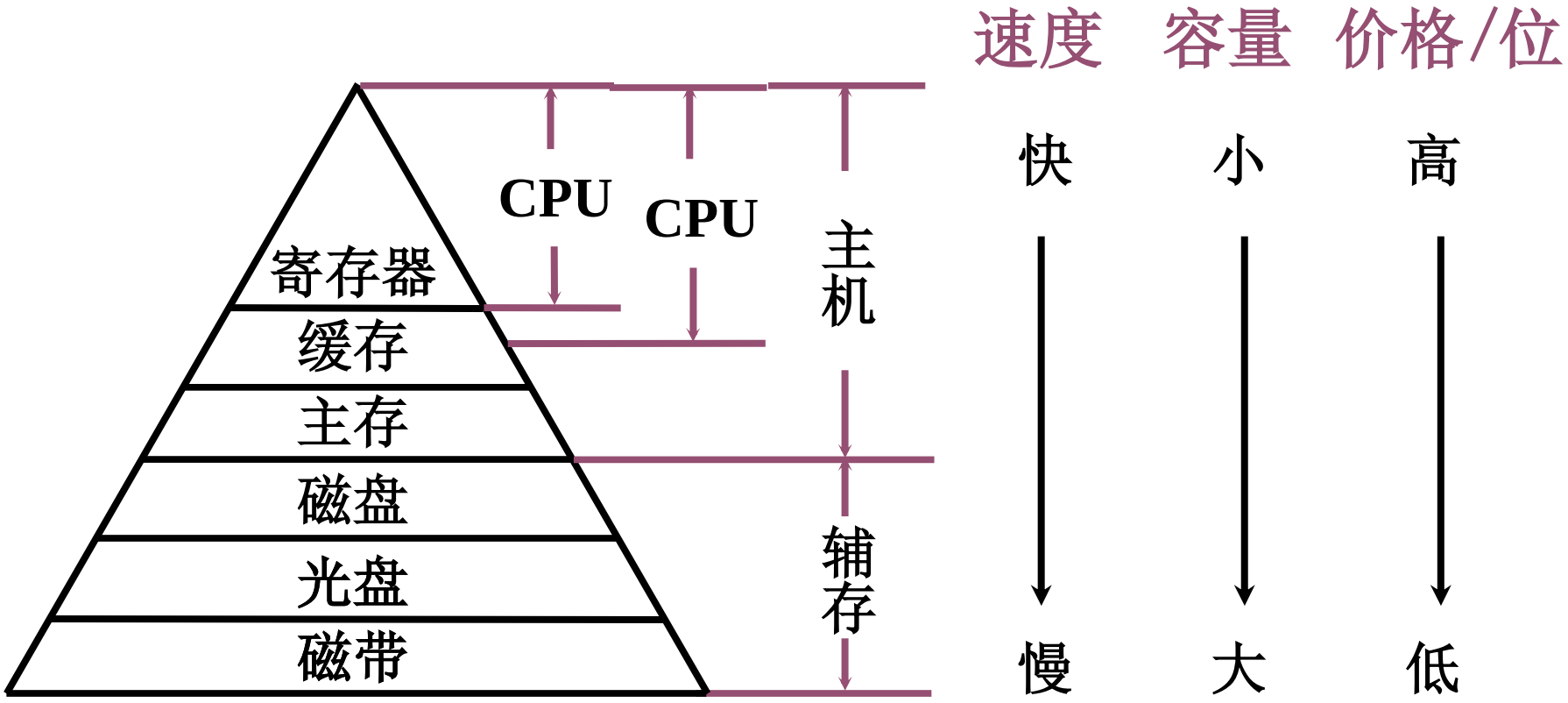




# 二、存储器的层次结构

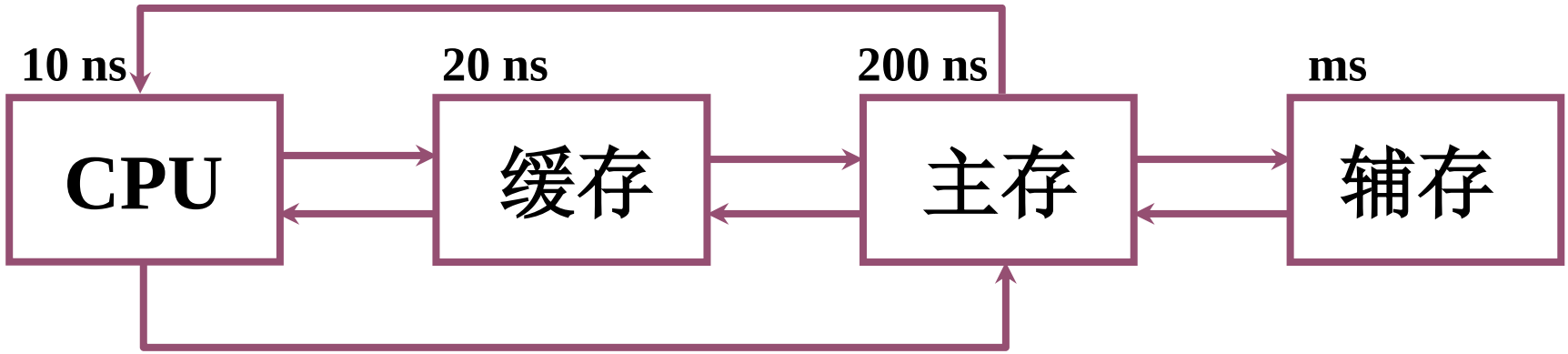
4.1

## 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






（速度）　　（容量）

缓存—主存　　主存—辅存

主存储器　　虚拟存储器

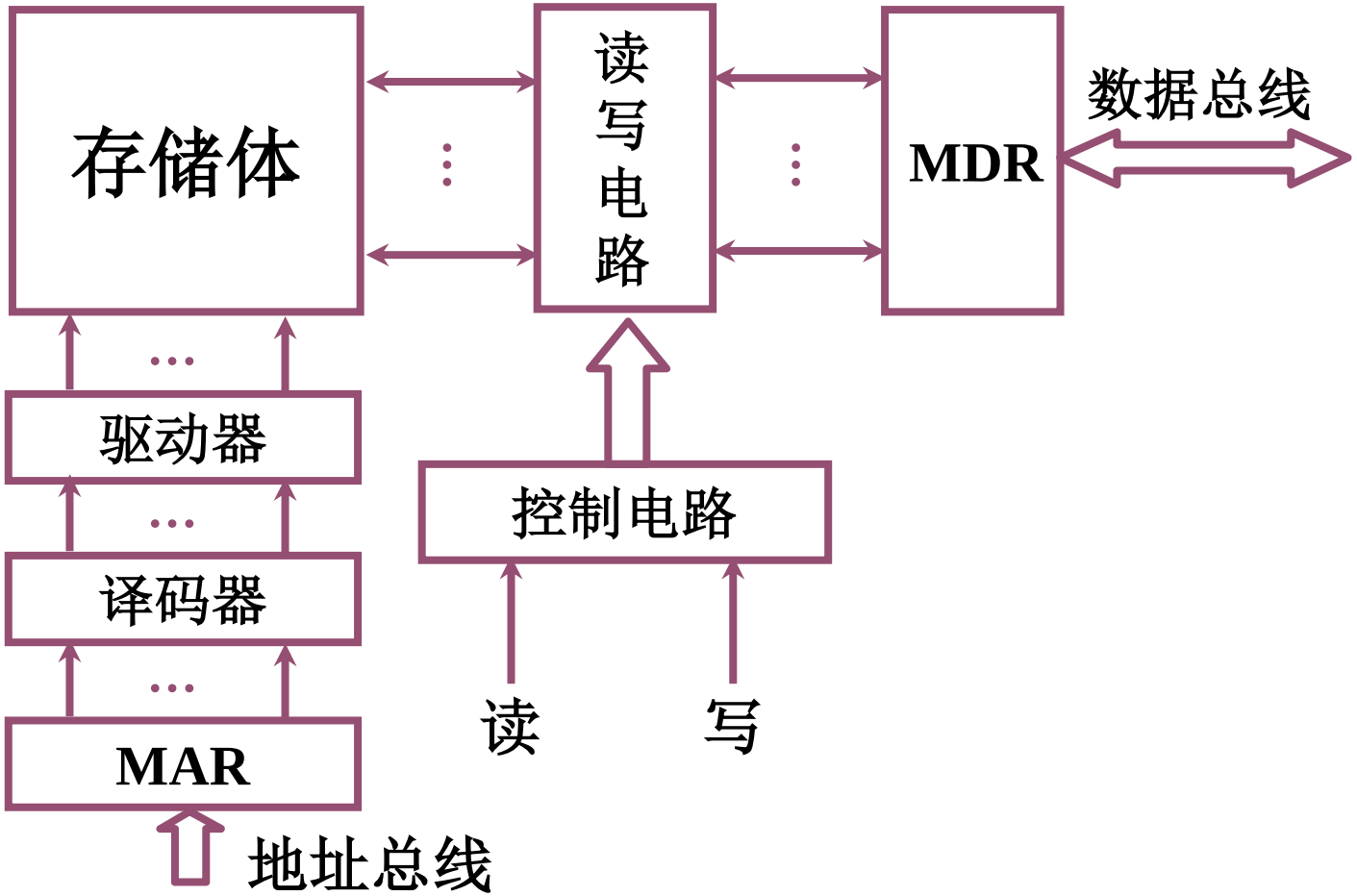
实地址　　虚地址

物理地址　　逻辑地址

# 4.2 主存储器

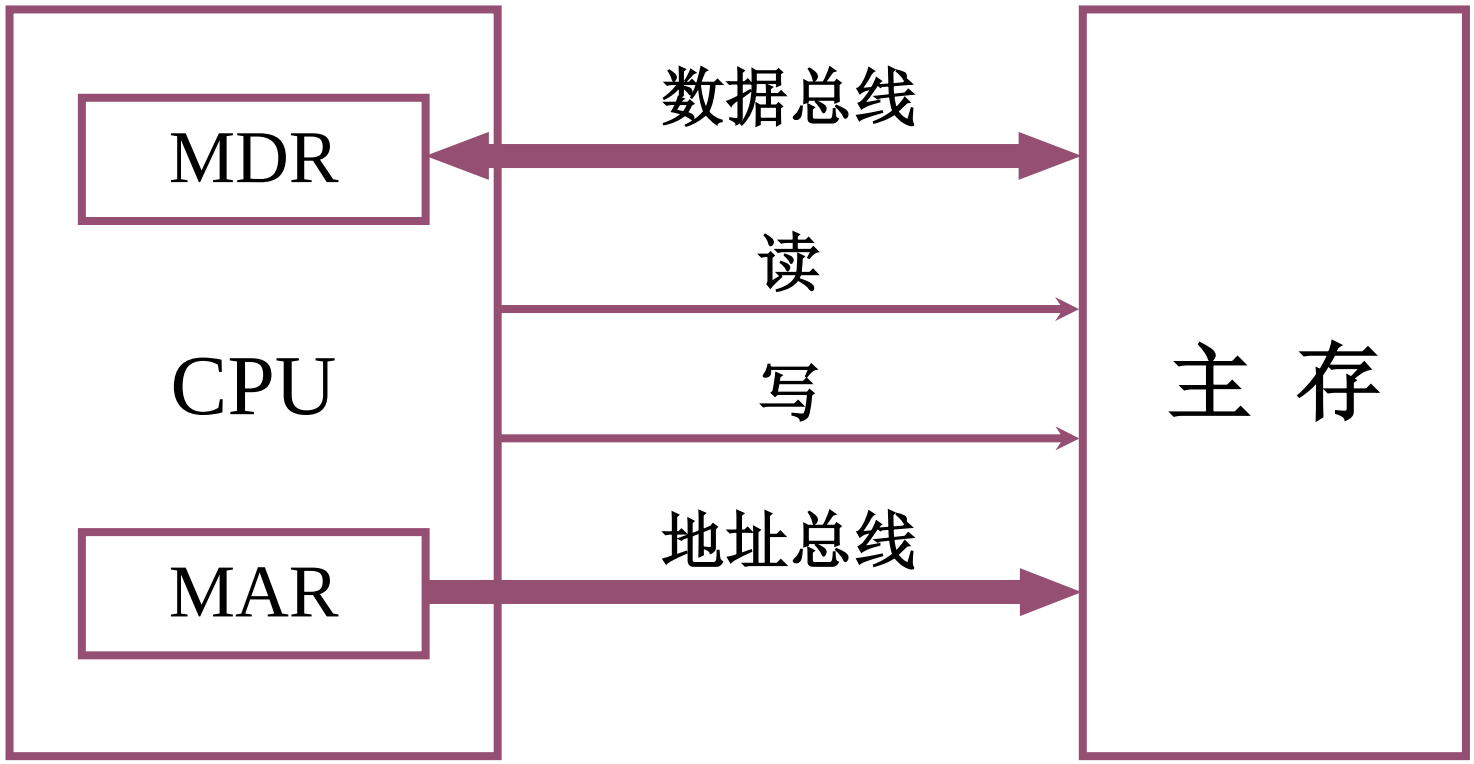
## 一、概述

### 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

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

低位字节 地址为字地址

| 字地址 | 字节地址 | |
|---|---|---|
| 0 | 1 | 0 |
| 2 | 3 | 2 |
| 4 | 5 | 4 |

设地址线 24 根　　按 字节 寻址　$2^{24}$ = 16 MB

若字长为 16 位　　按 字 寻址　8 MW

若字长为 32 位　　按 字 寻址　4 MW

# 4. 主存的技术指标



(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

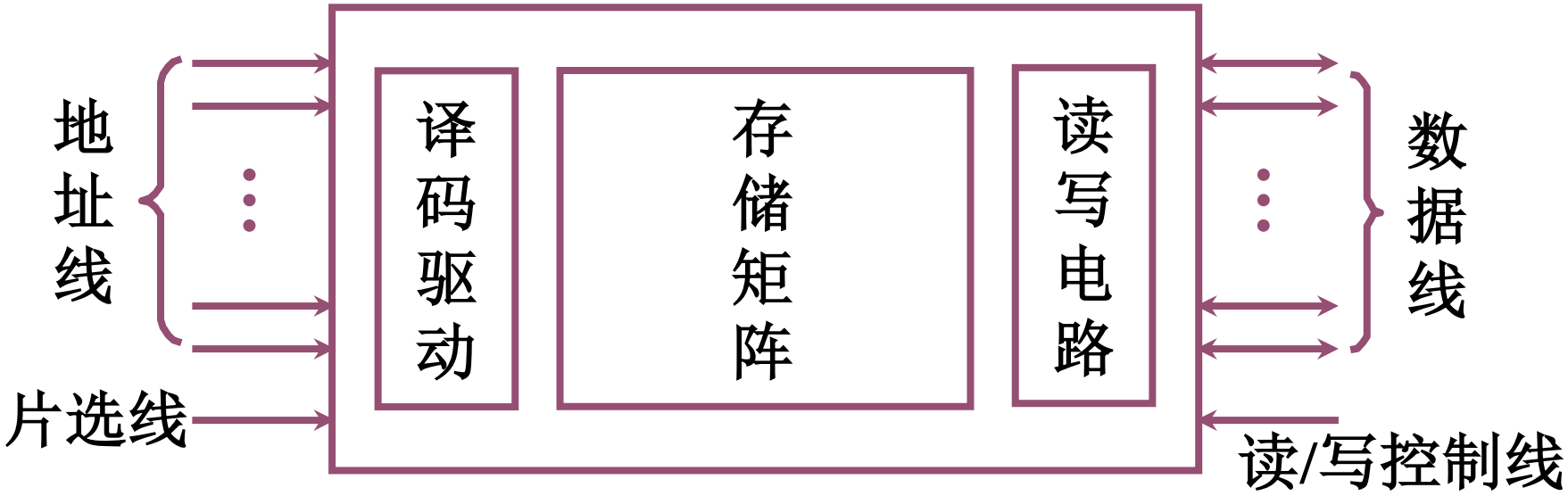
  读周期　写周期

(3) 存储器的带宽　位/秒

# 二、半导体存储芯片简介

4.2

## 1. 半导体存储芯片的基本结构

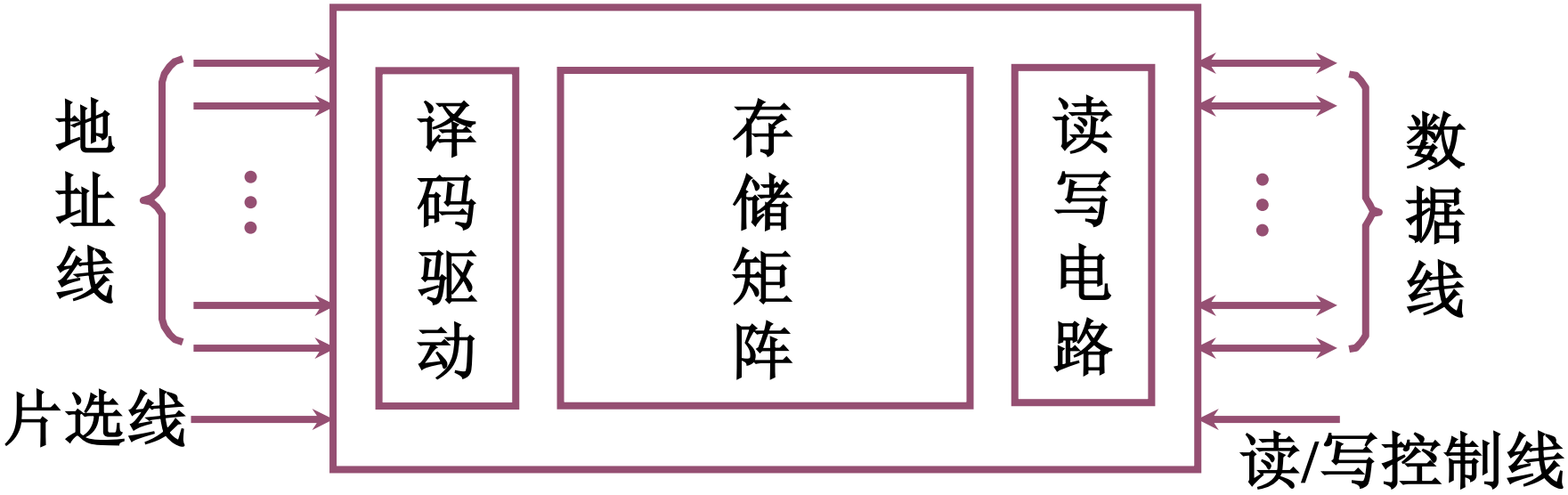



| 地址线（单向） | 数据线（双向） | 芯片容量 |
|---|---|---|
| 10 | 4 | 1K×4位 |
| 14 | 1 | 16K×1位 |
| 13 | 8 | 8K×8位 |

# 二、半导体存储芯片简介

4.2

## 1. 半导体存储芯片的基本结构




片选线 $\overline{CS}$ $\overline{CE}$

读/写控制线 $\overline{WE}$（低电平写 高电平读）

$\overline{OE}$（允许读）$\overline{WE}$（允许写）

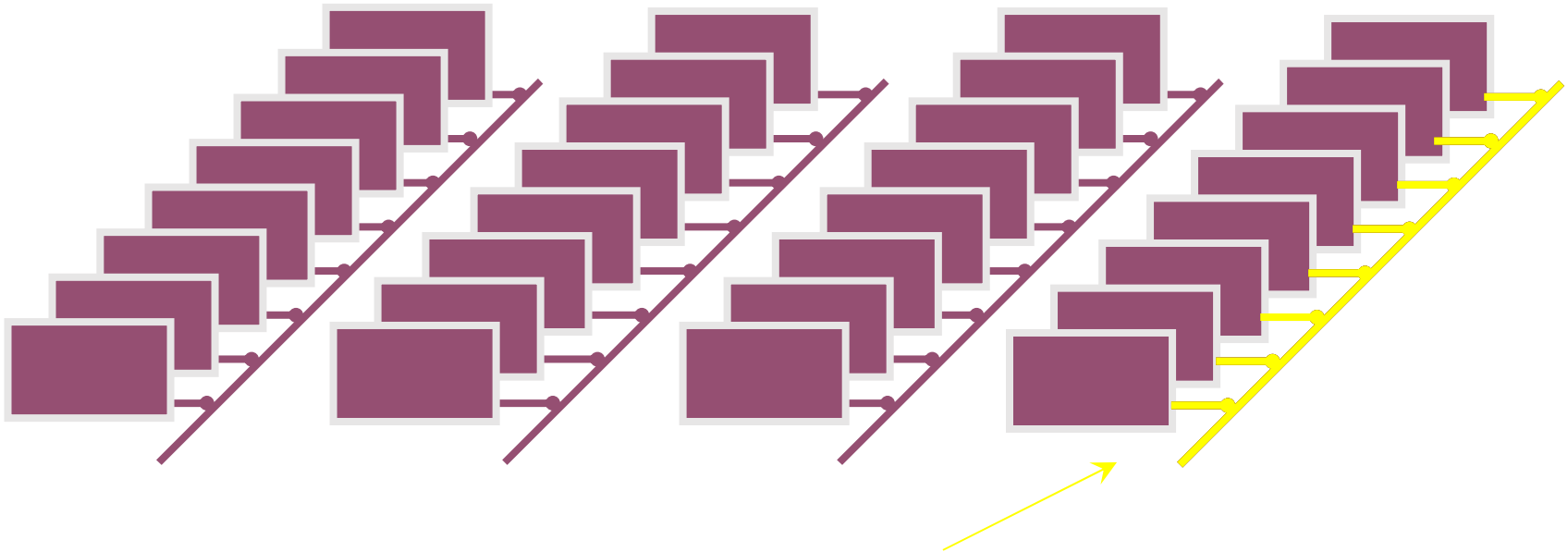
# 存储芯片片选线的作用



用 16K × 1位 的存储芯片组成 64K × 8位 的存储器

32片

8片 16K × 1位　8片 16K × 1位　8片 16K × 1位　8片 16K × 1位

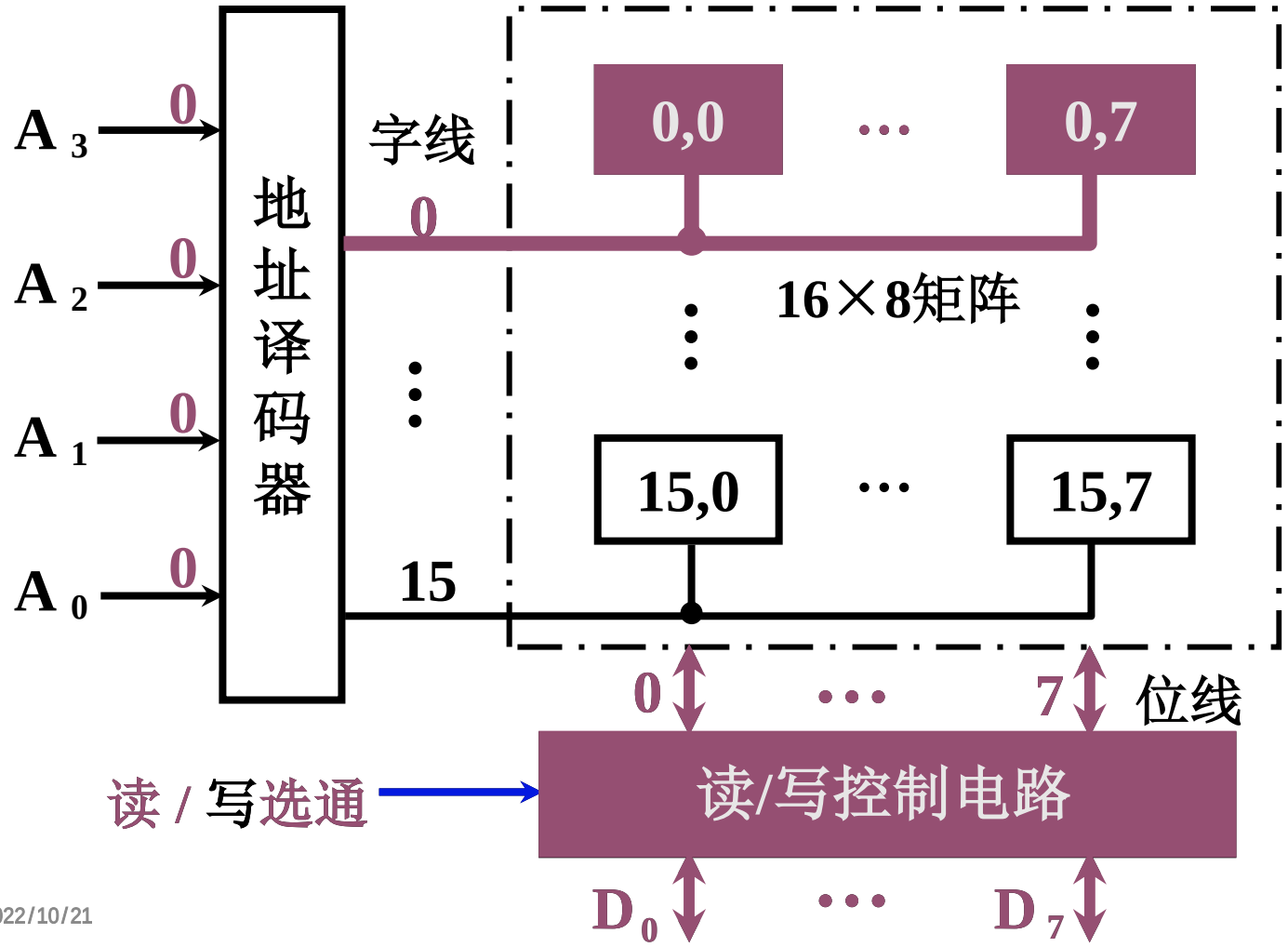


当地址为 65 535 时，此 8 片的片选有效

# 2. 半导体存储芯片的译码驱动方式

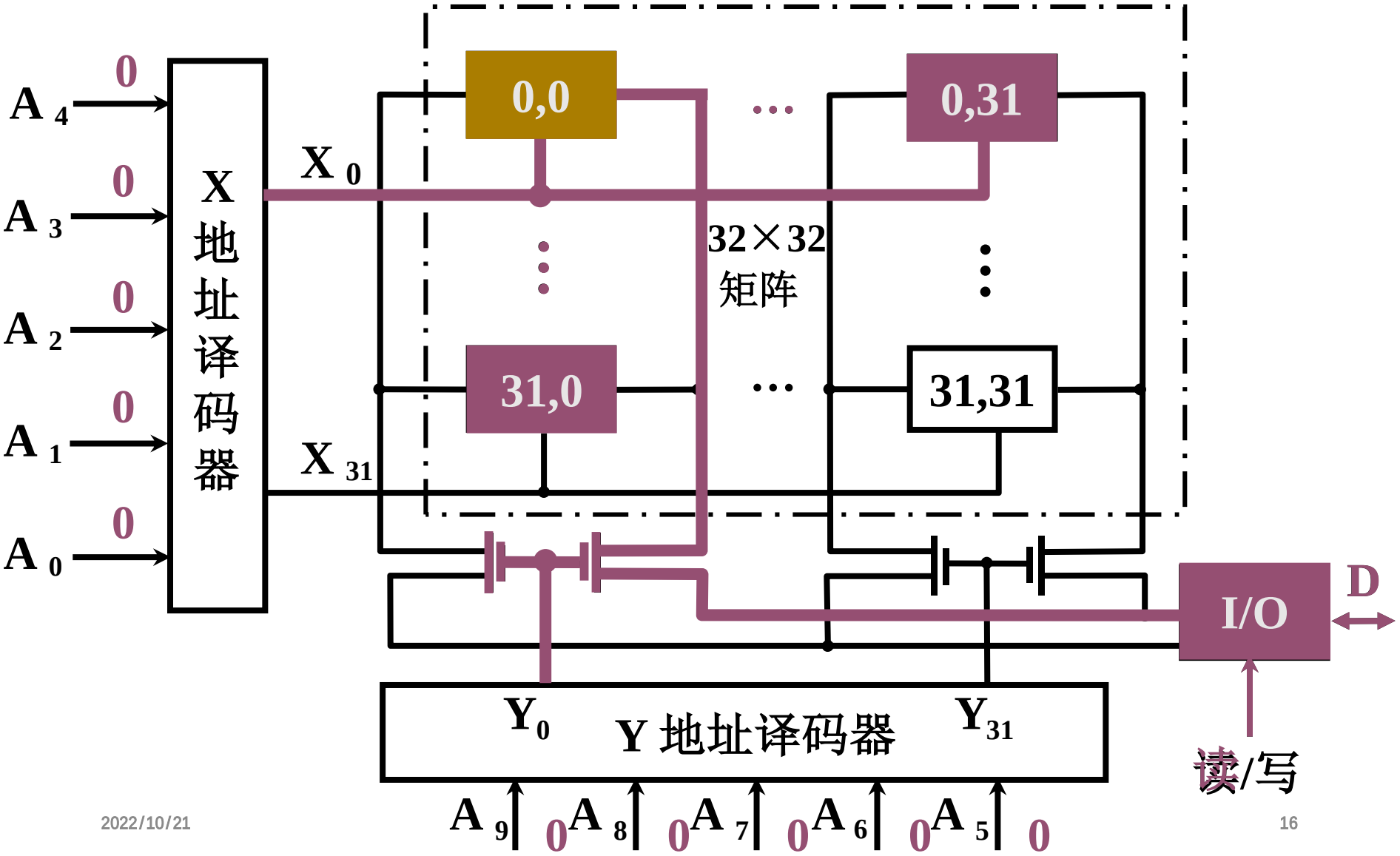


## (1) 线选法



# (2) 重合法



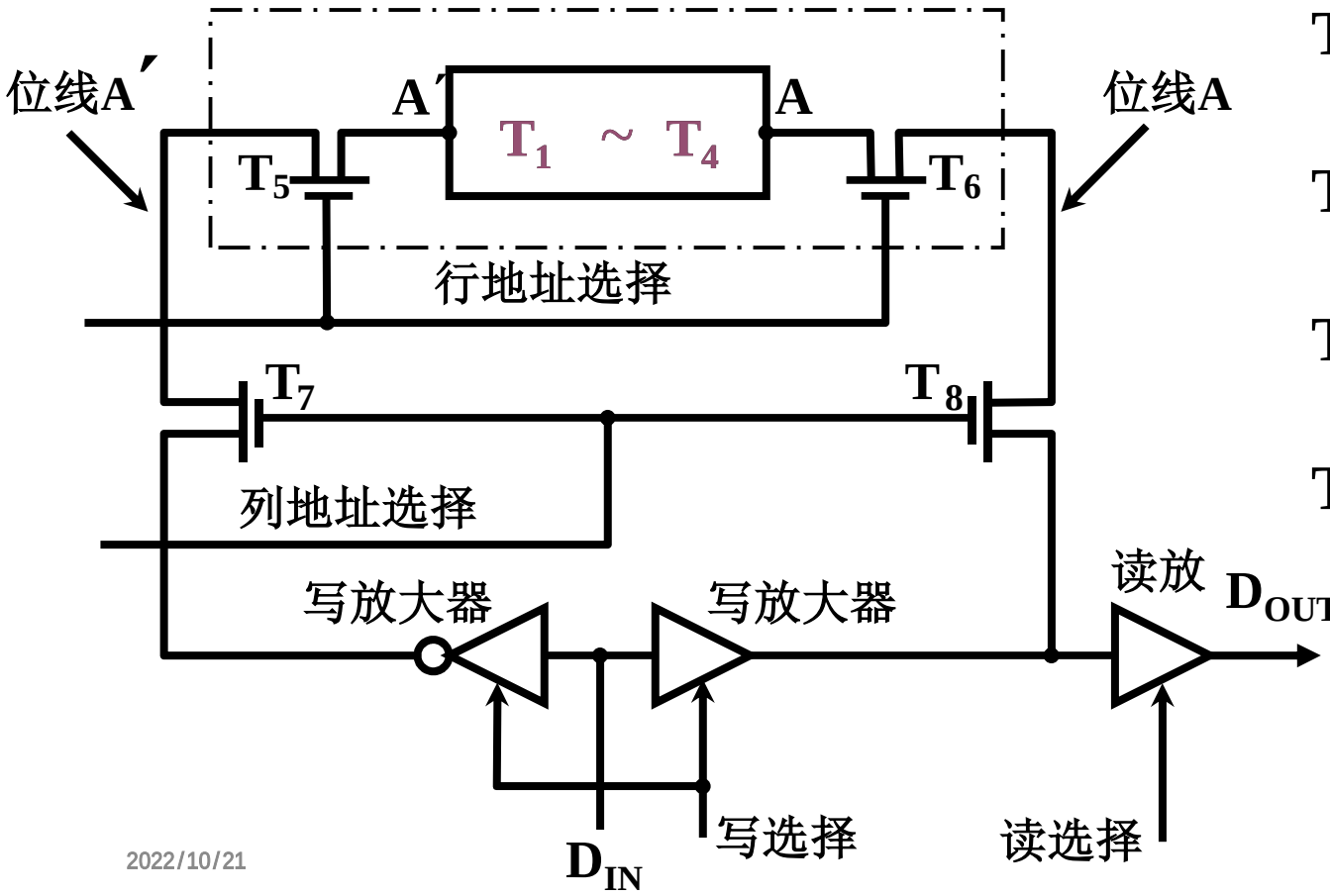

A 4
A 3
A 2
A 1
A 0
0
0
0
0
0
X 地址译码器
X 0
X 31
0,0
0,31
31,0
31,31
32×32 矩阵
I/O
D
Y 0
Y 地址译码器
Y 31
读/写
A 9
A 8
A 7
A 6
A 5
0
0
0
0
0

# 三、随机存取存储器 ( RAM )

4.2

## 1. 静态 RAM (SRAM)

### (1) 静态 RAM 基本电路




$T_1 \sim T_4$ 触发器

$T_5$ 、$T_6$ 行开关

$T_7$ 、$T_8$ 列开关

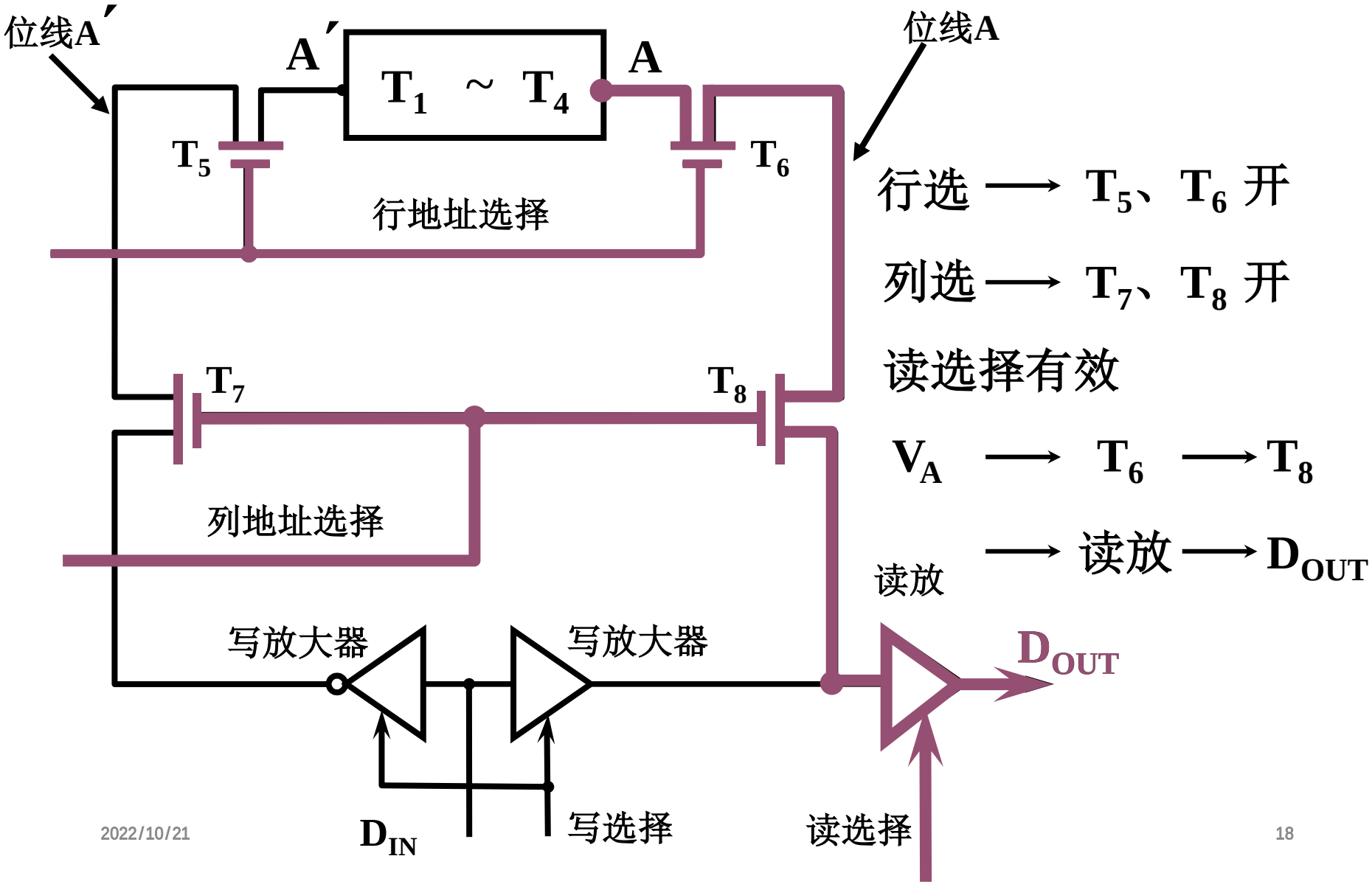
$T_7$ 、$T_8$ 一列共用

A 触发器原端

A′ 触发器非端

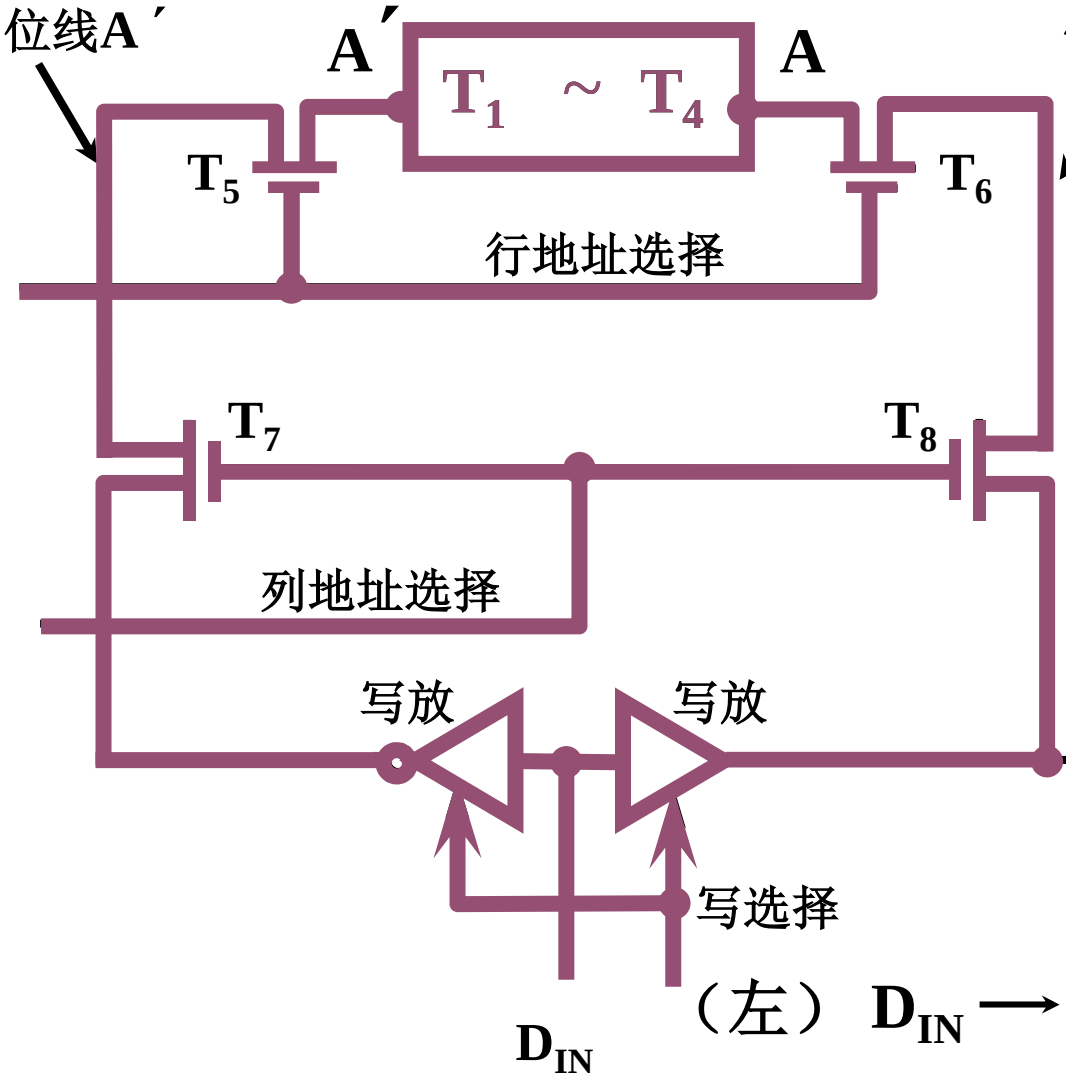
# ① 静态 RAM 基本电路的 读 操作





# ② 静态 RAM 基本电路的 写 操作






行选 → $T_5$、$T_6$ 开

列选 → $T_7$、$T_8$ 开

写选择有效

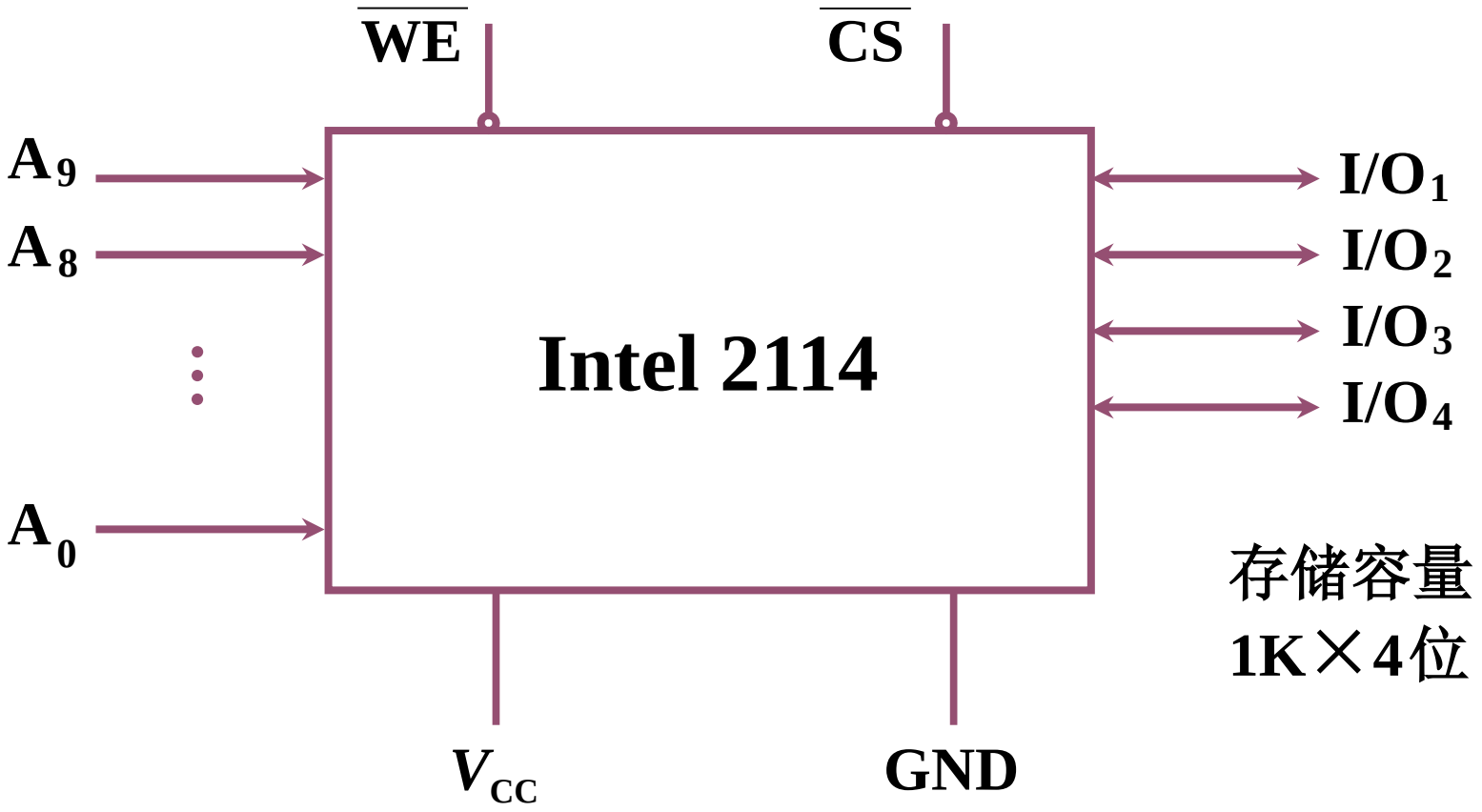
$D_{IN}$ → 两个写放

（左） $D_{IN}$ → 反相 → $T_7$ → $T_5$ → A′

（右） $D_{IN}$ → $T_8$ → $T_6$ → A

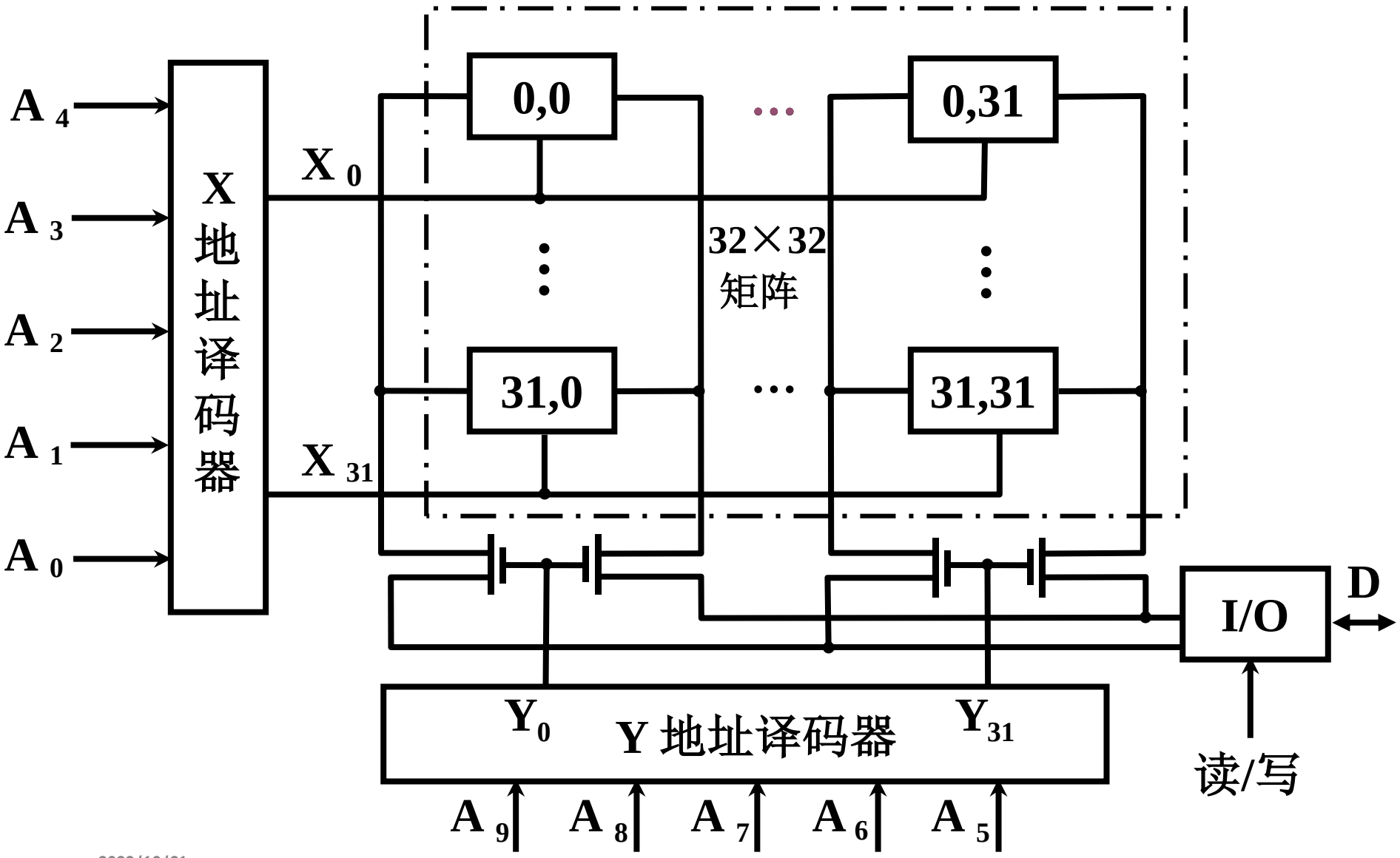
## (2) 静态 RAM 芯片举例

### ① Intel 2114 外特性



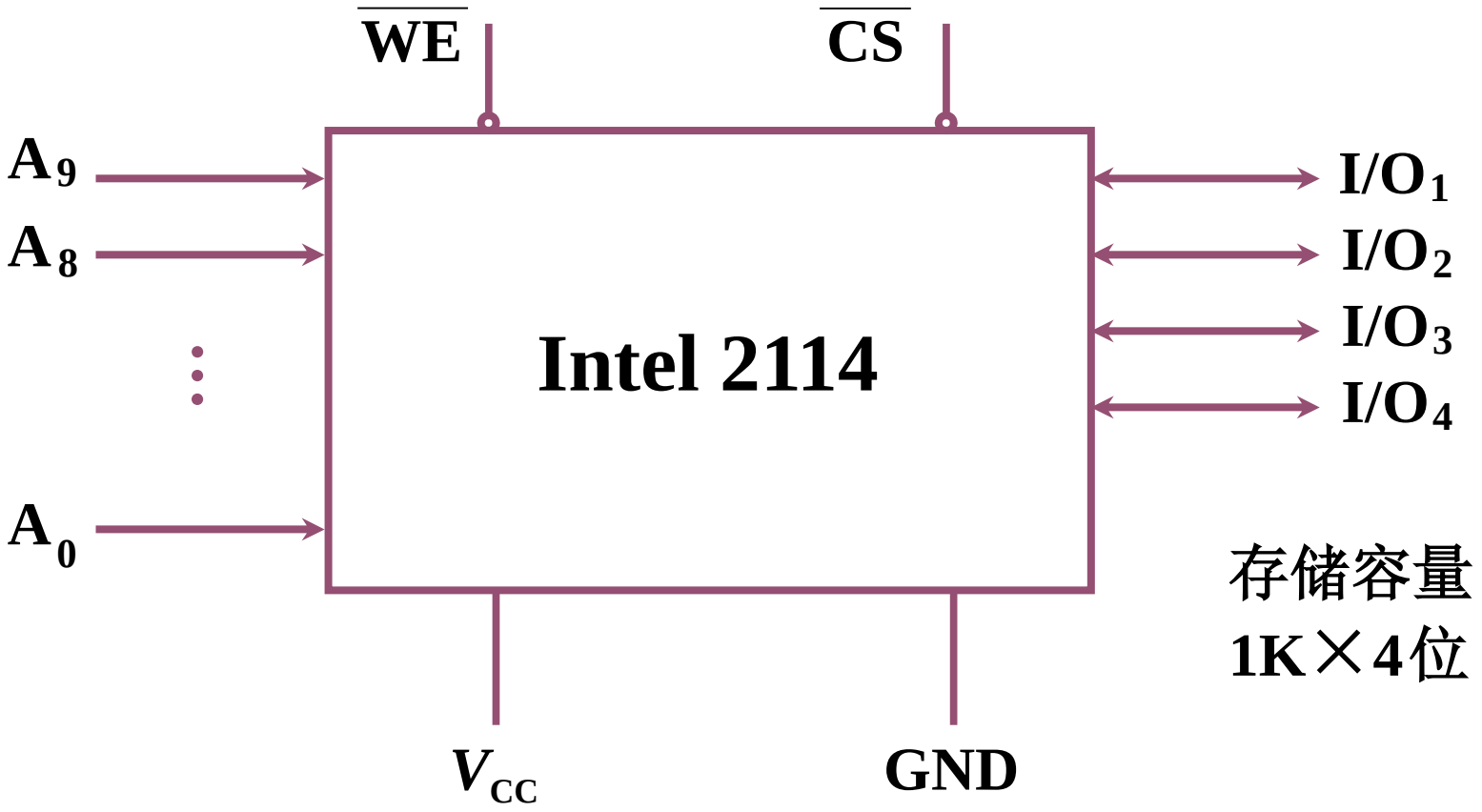

存储容量
1K×4位

# 曾经讲到过的重合法，怎么实现选一次四列？



## (2) 静态 RAM 芯片举例

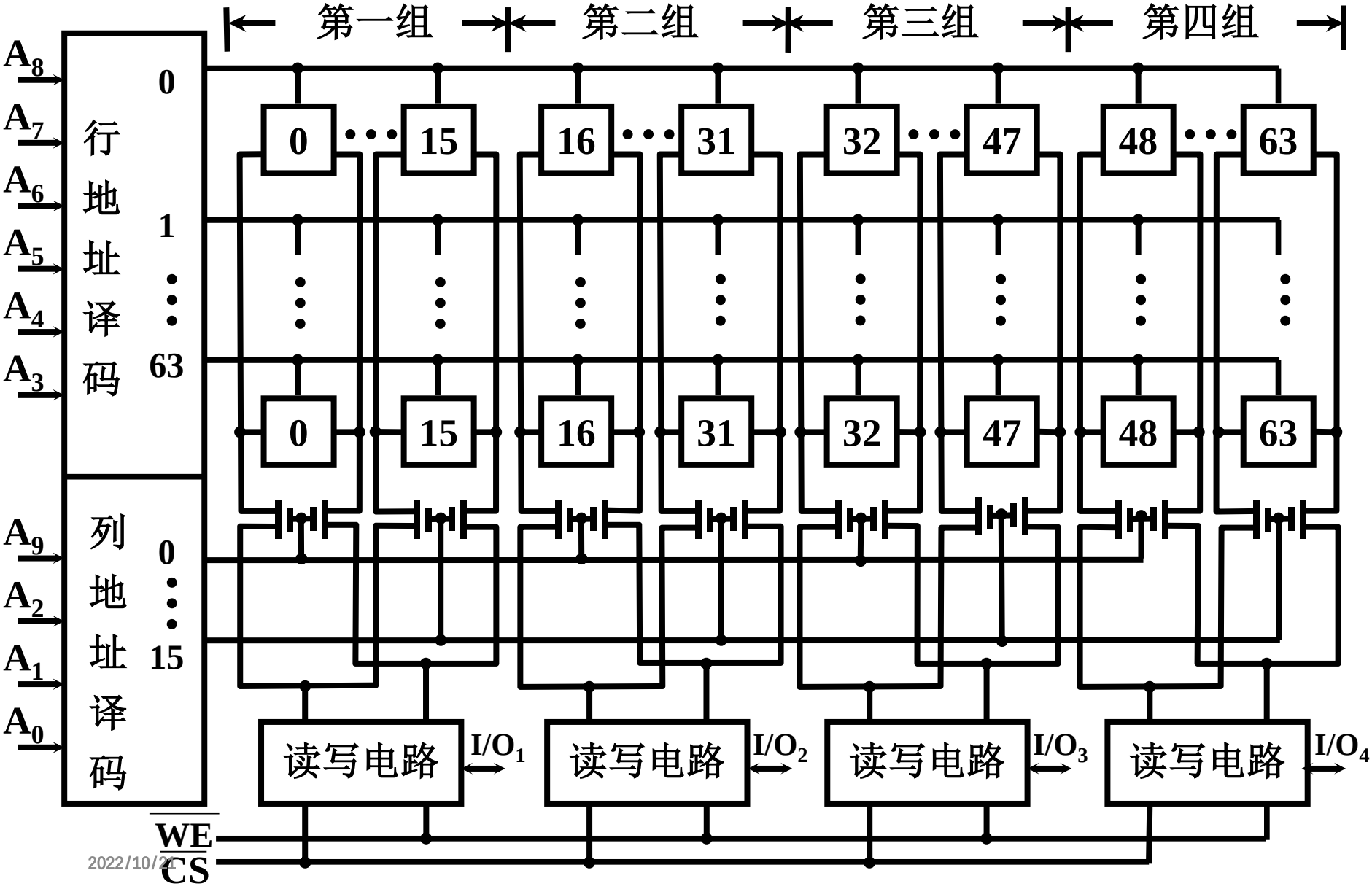
### ① Intel 2114 外特性




存储容量
1K×4位

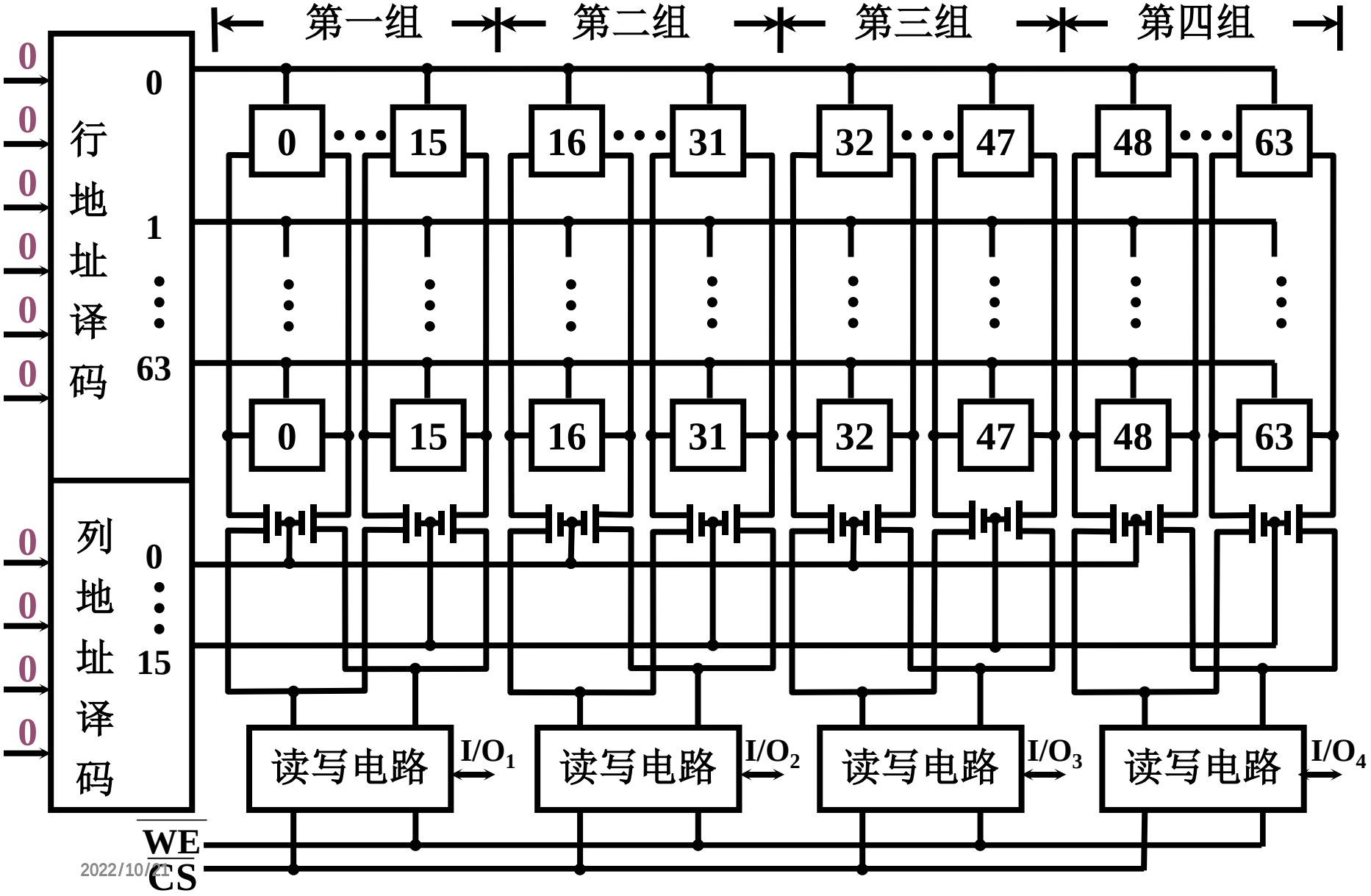
# ② Intel 2114 RAM 矩阵 (64 × 64) 读

4.2



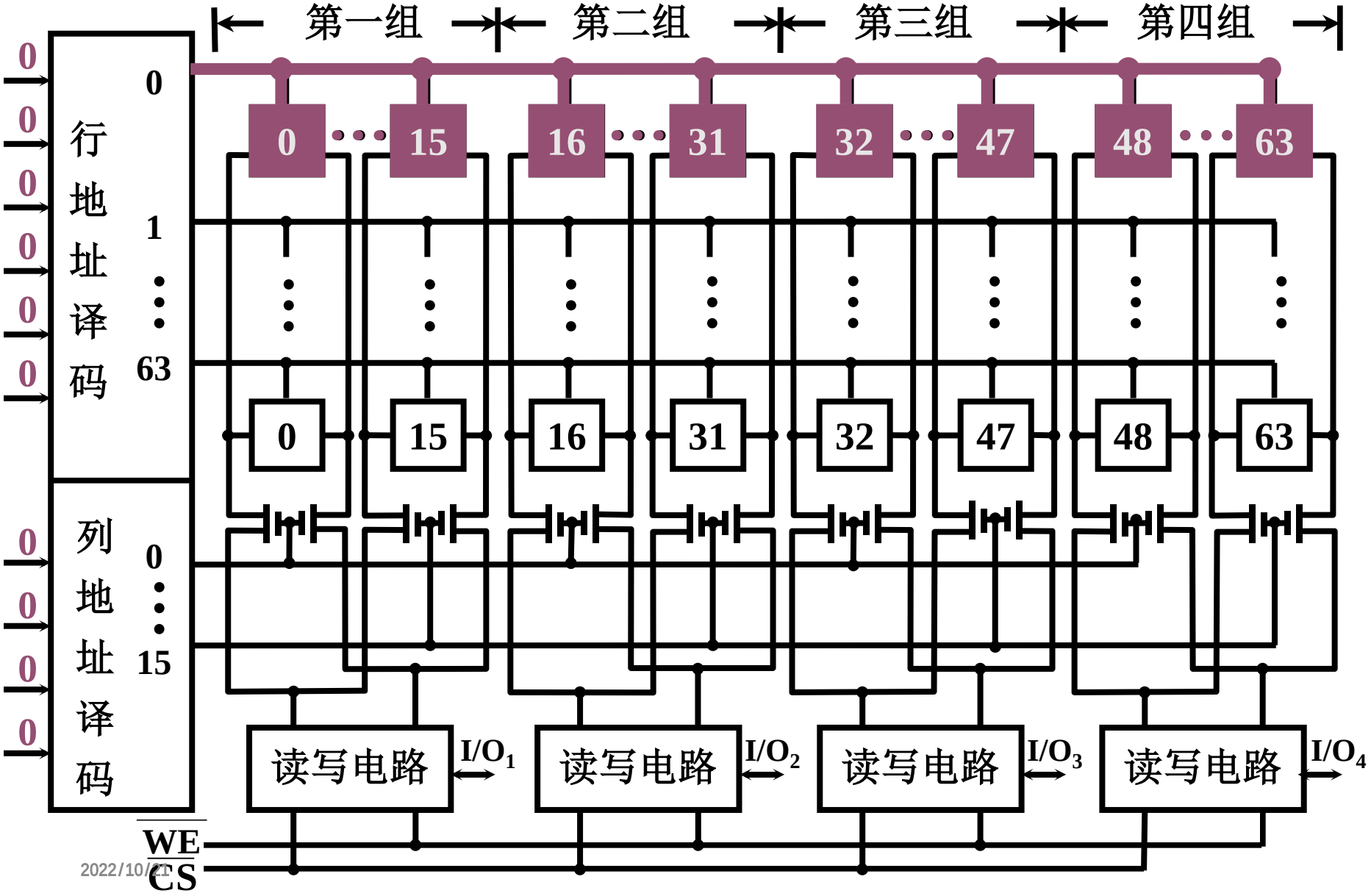
# ② Intel 2114 RAM 矩阵 (64 × 64) 读

4.2



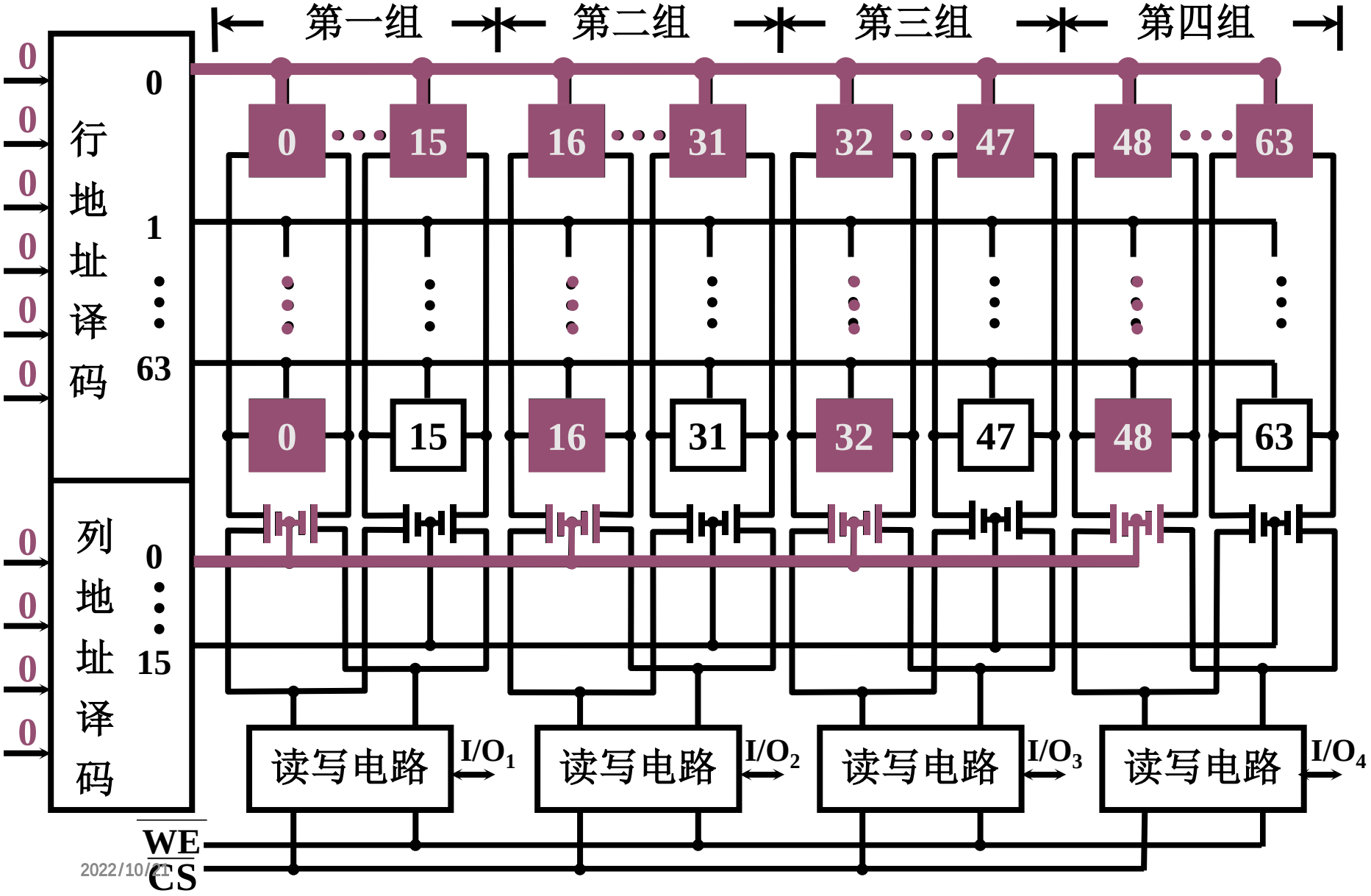
# ② Intel 2114 RAM 矩阵 (64 × 64) 读





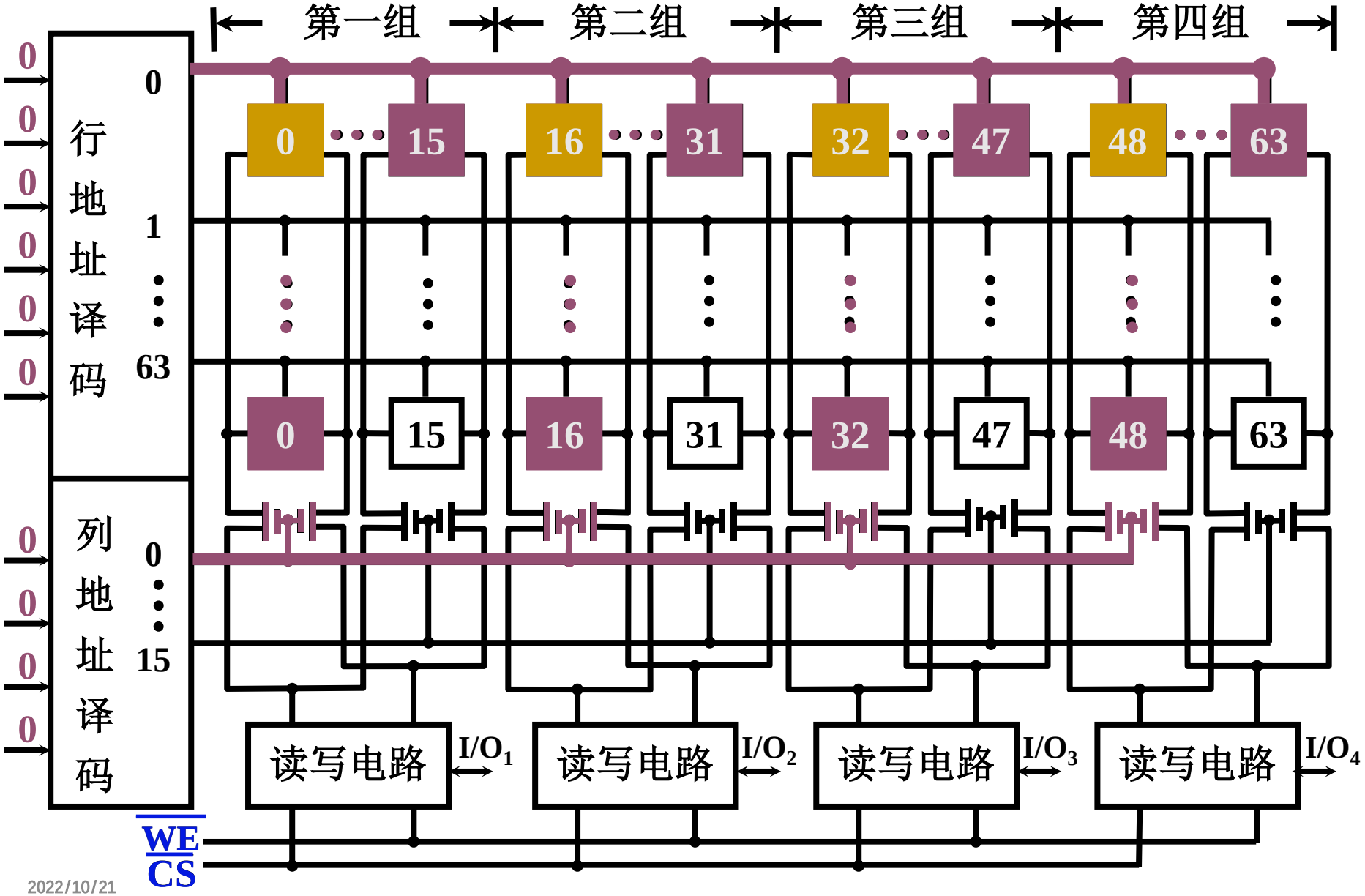
# ② Intel 2114 RAM 矩阵 (64 × 64) 读





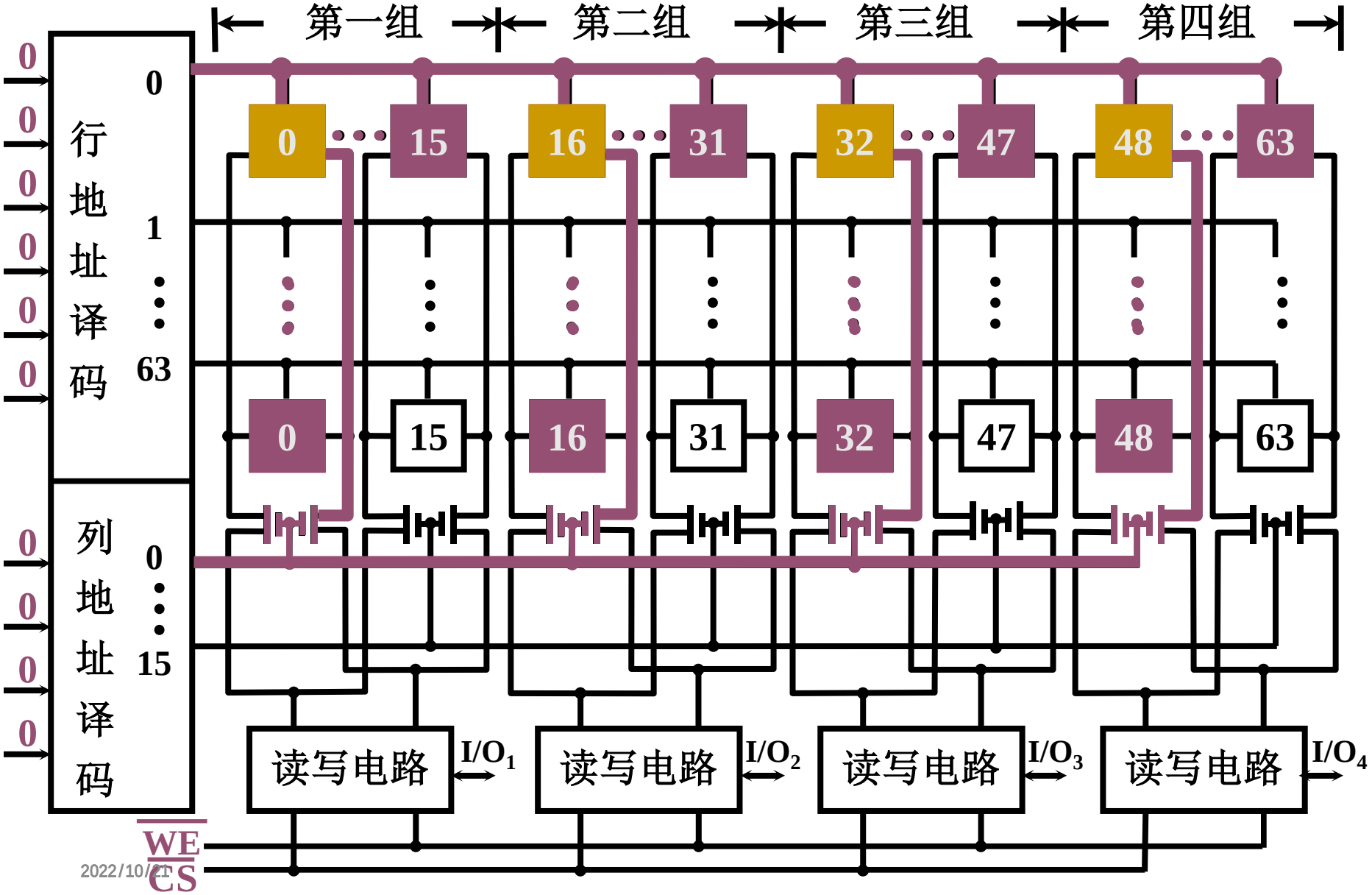
# ② Intel 2114 RAM 矩阵 (64 × 64) 读

4.2



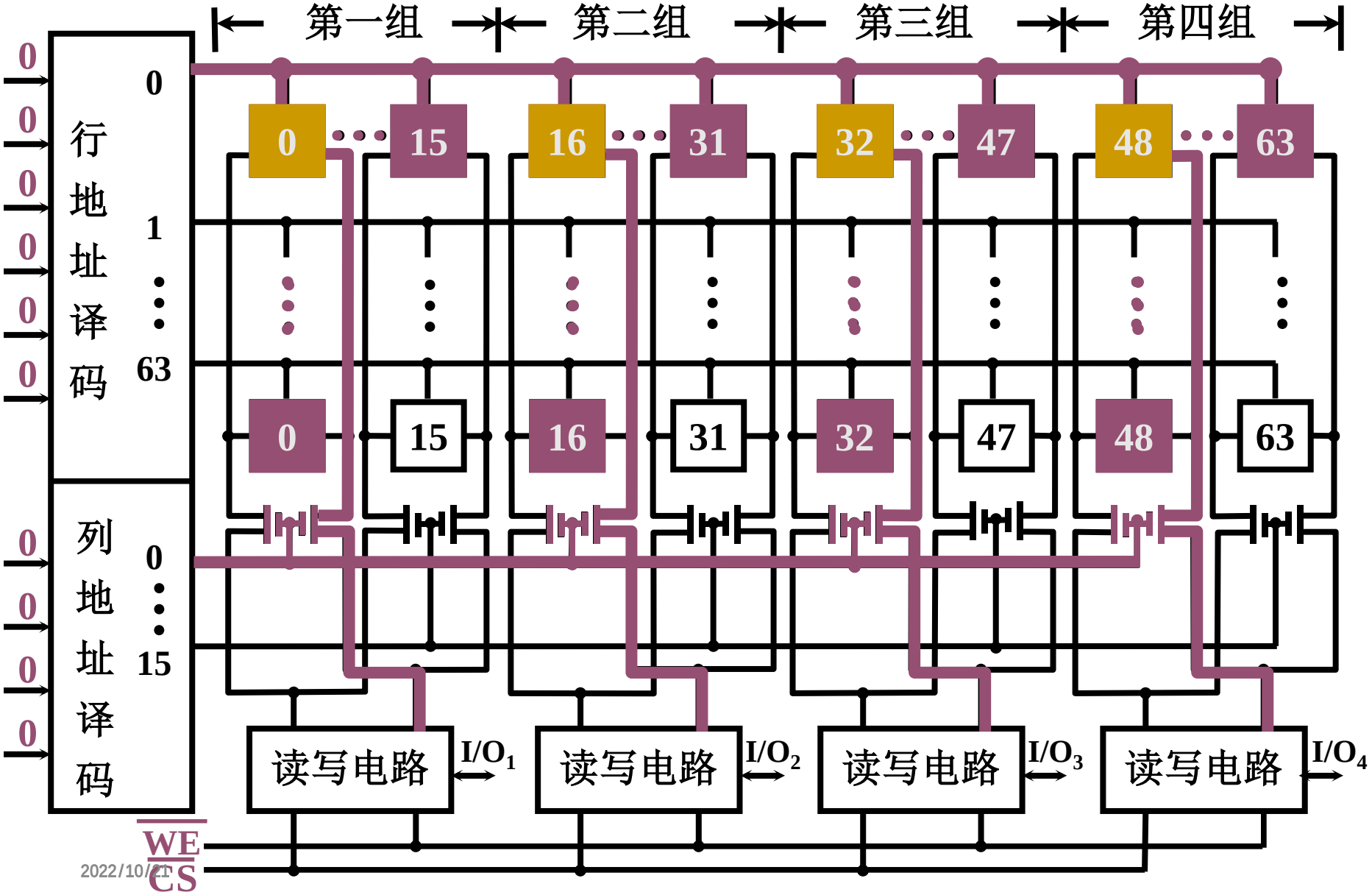
# ② Intel 2114 RAM 矩阵 (64 × 64) 读

4.2



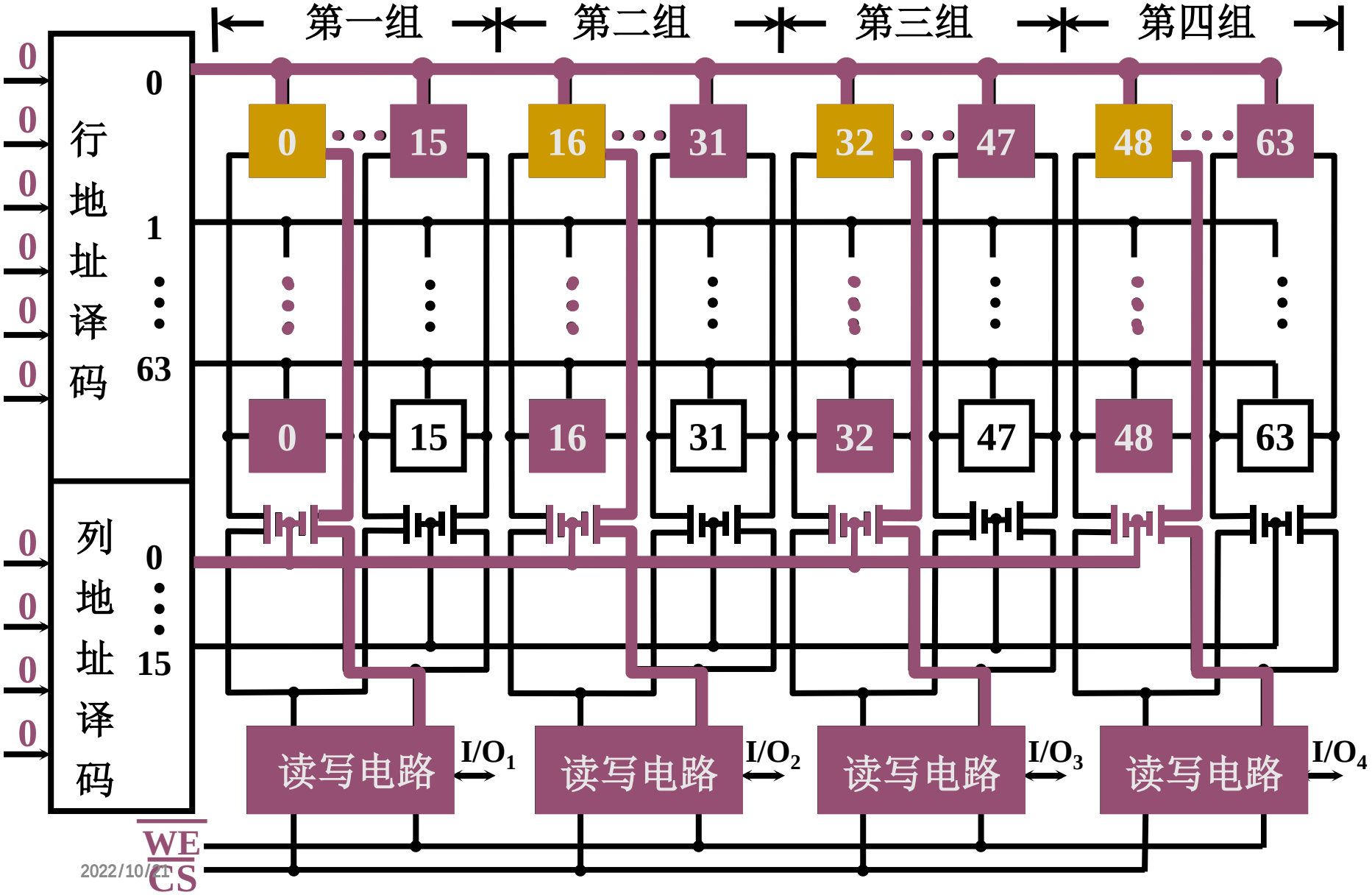
# ② Intel 2114 RAM 矩阵 (64 × 64) 读

4.2



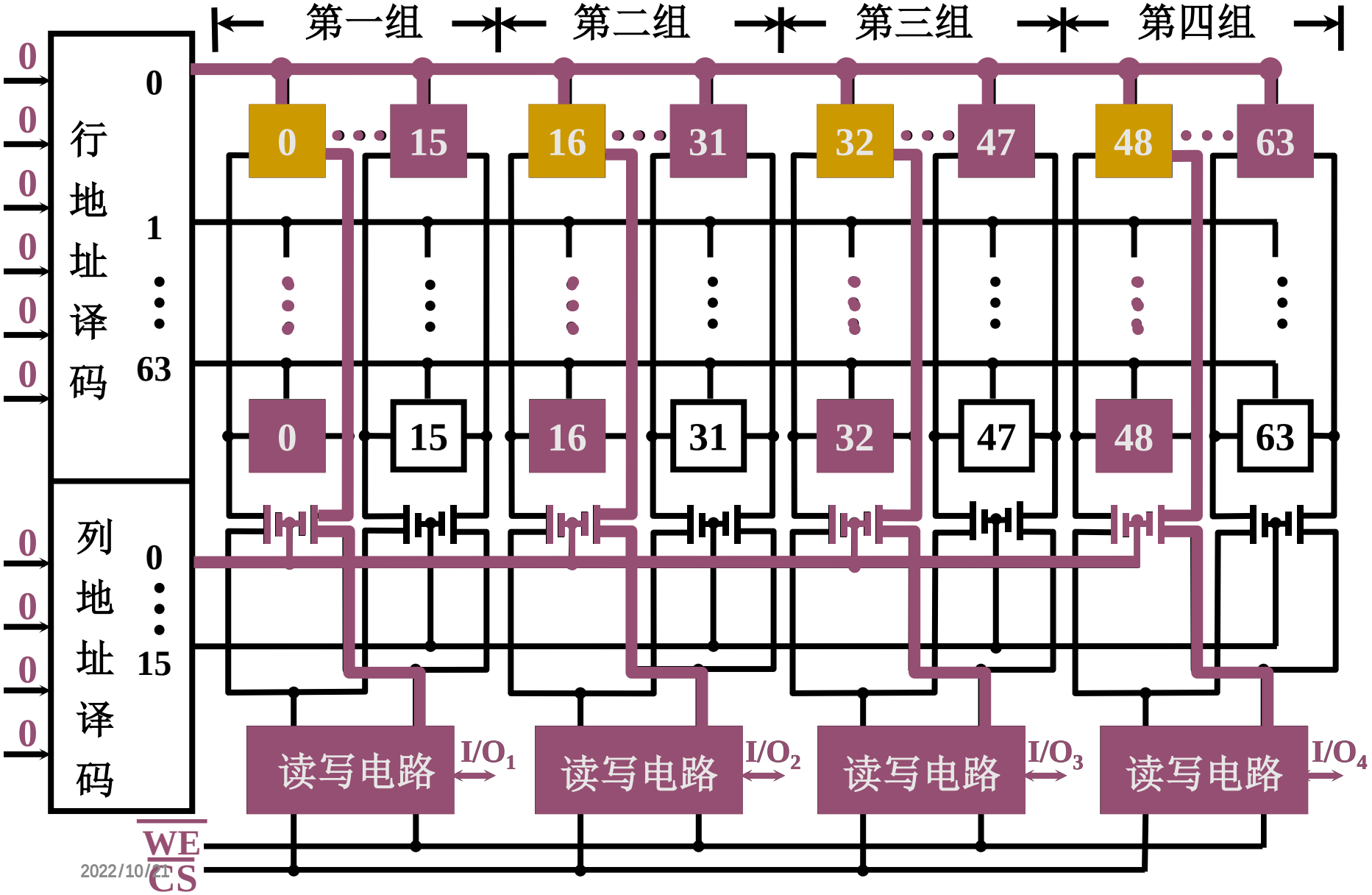
# ② Intel 2114 RAM 矩阵 (64 × 64) 读

4.2



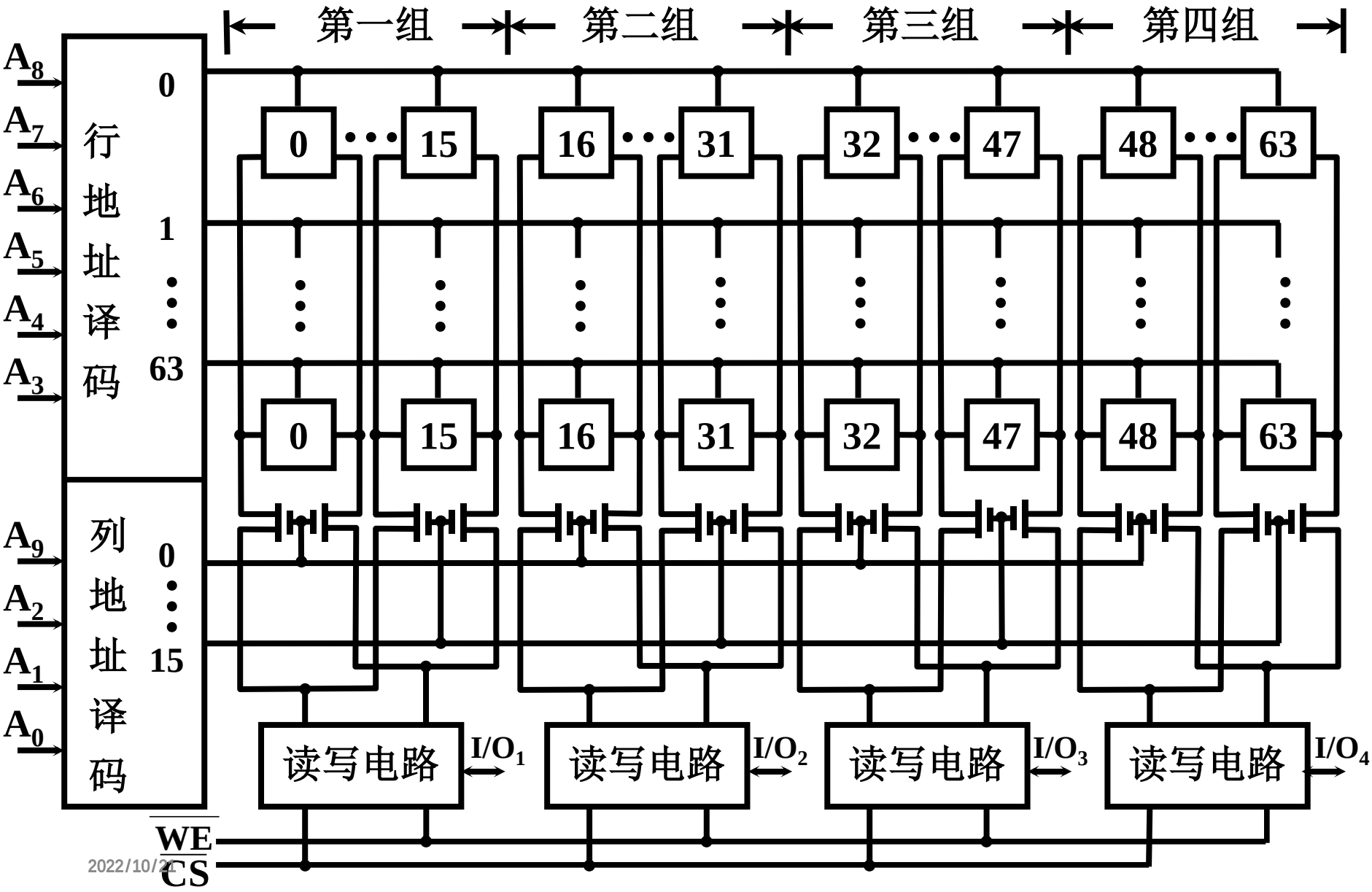
# ② Intel 2114 RAM 矩阵 (64 × 64) 读

4.2



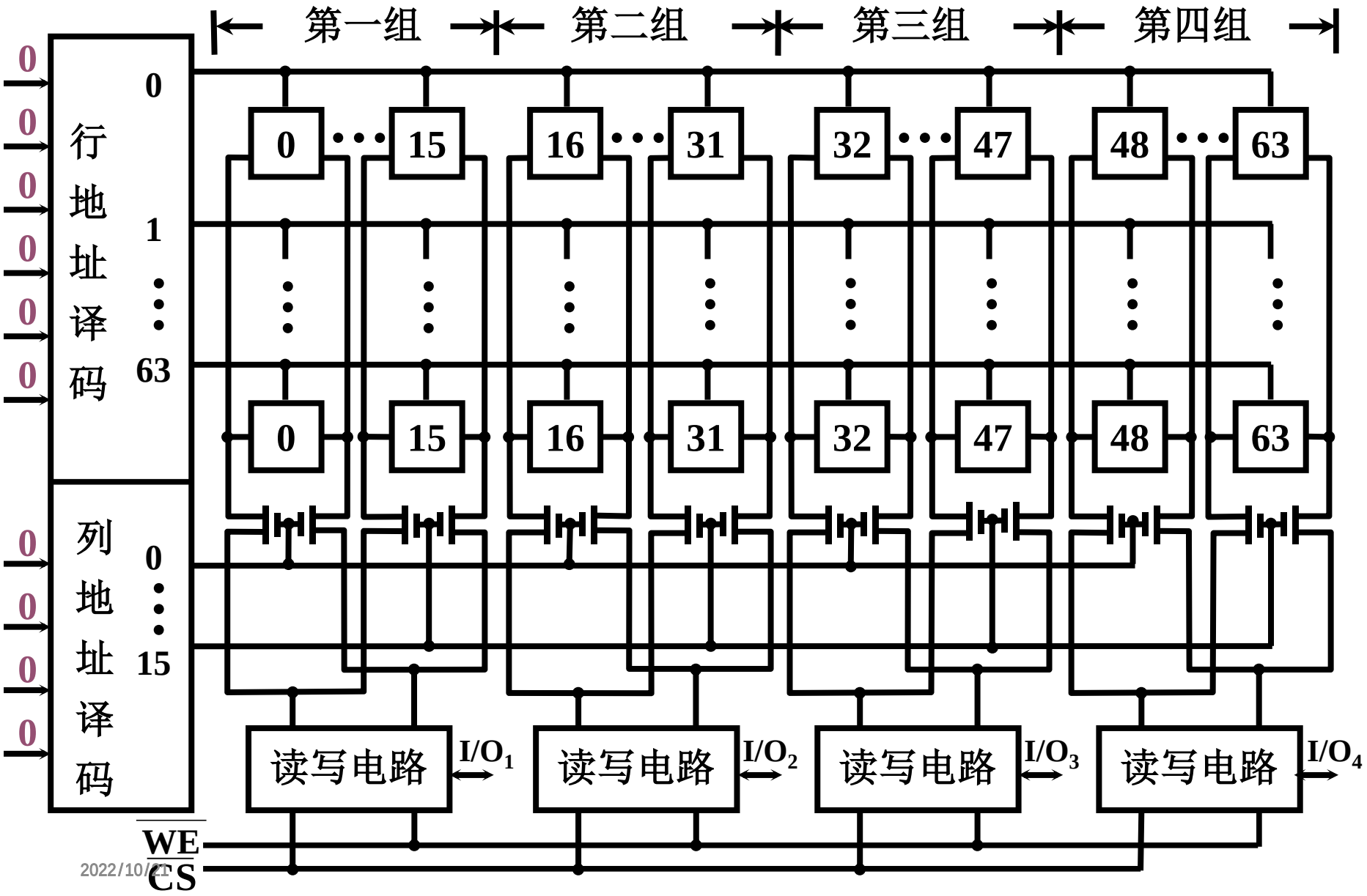
# ③ Intel 2114 RAM 矩阵 (64 × 64) 写

4.2



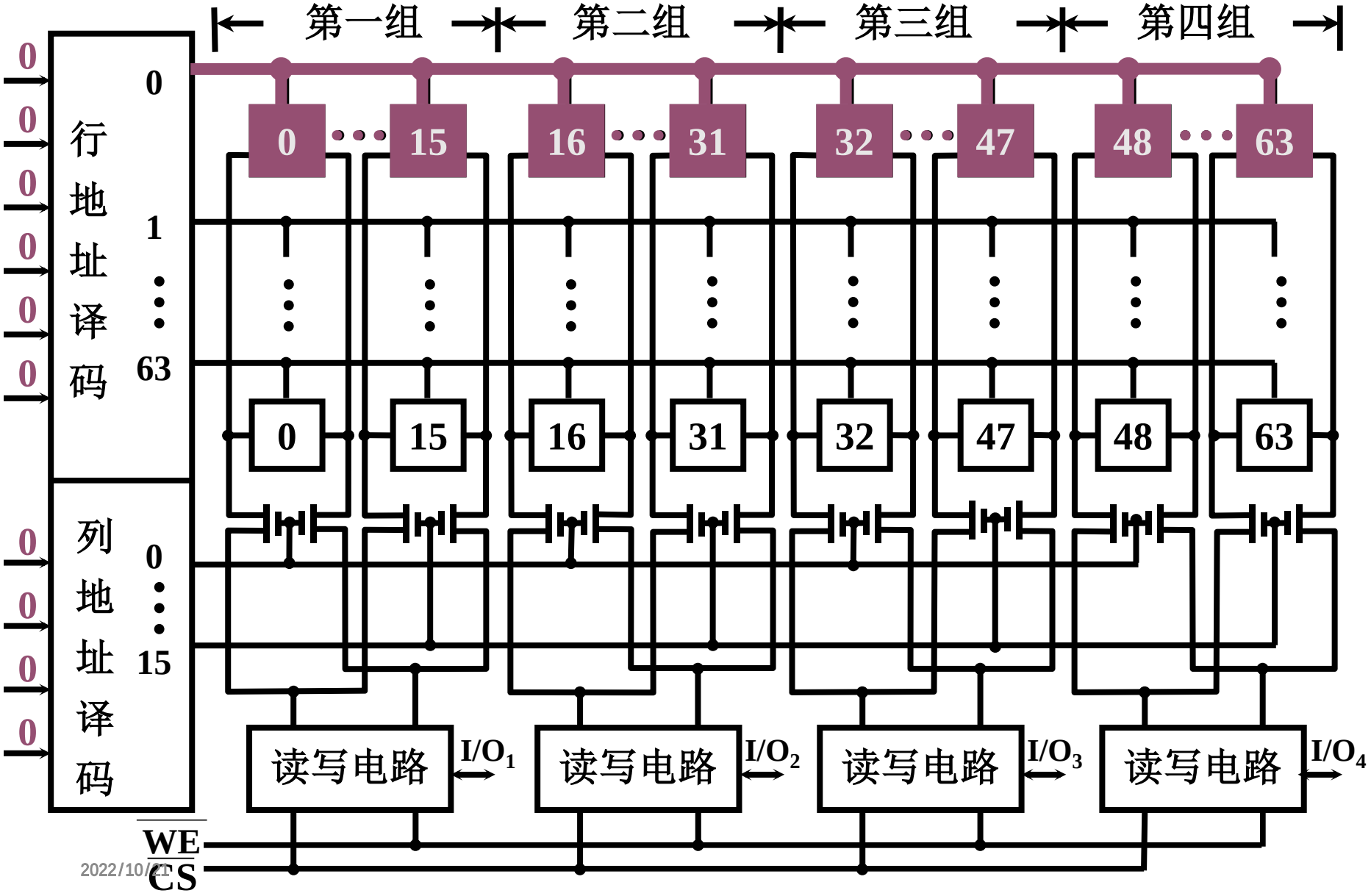
# ③ Intel 2114 RAM 矩阵 (64 × 64) 写

4.2



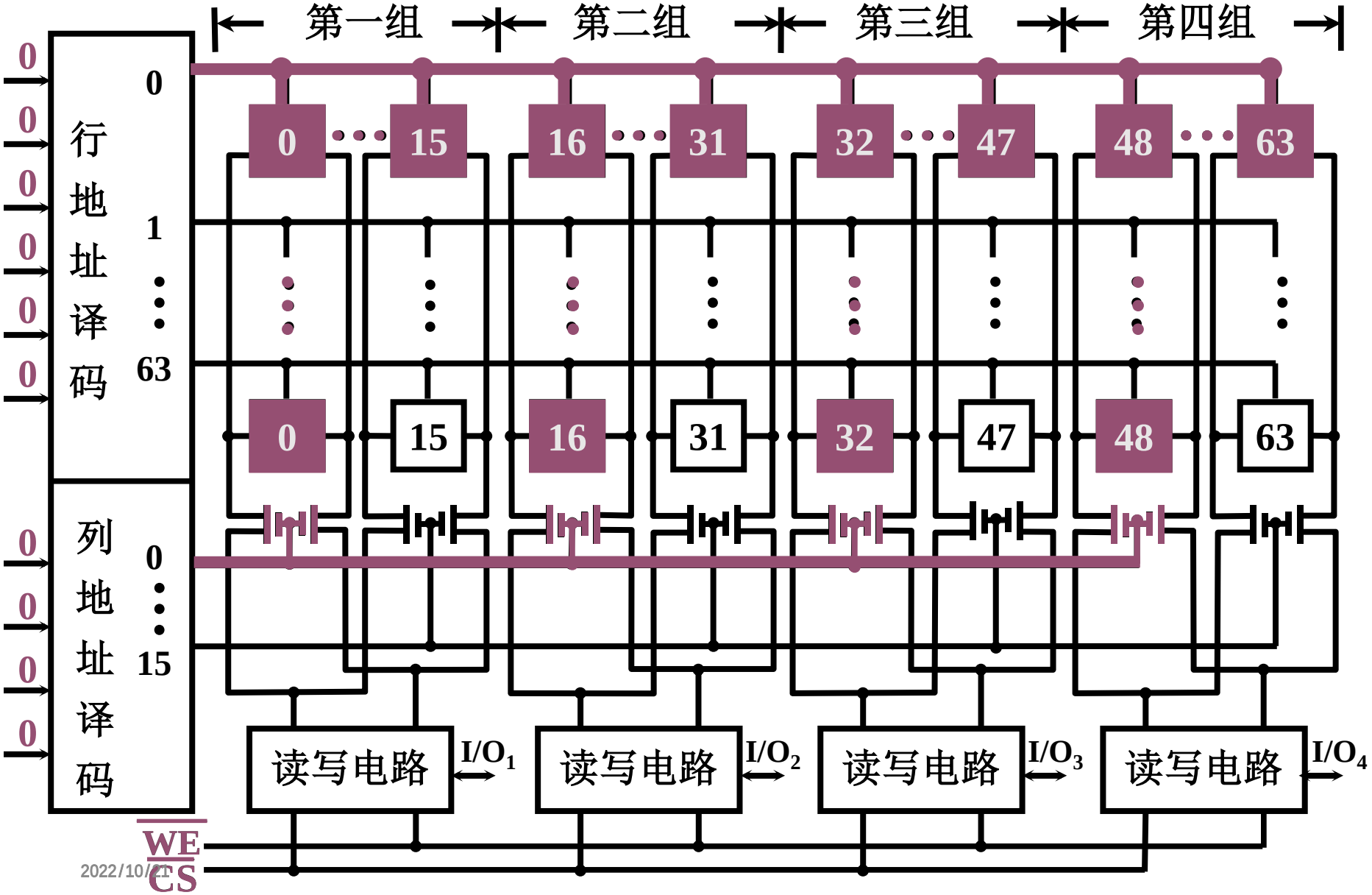
# ③ Intel 2114 RAM 矩阵 (64 × 64) 写





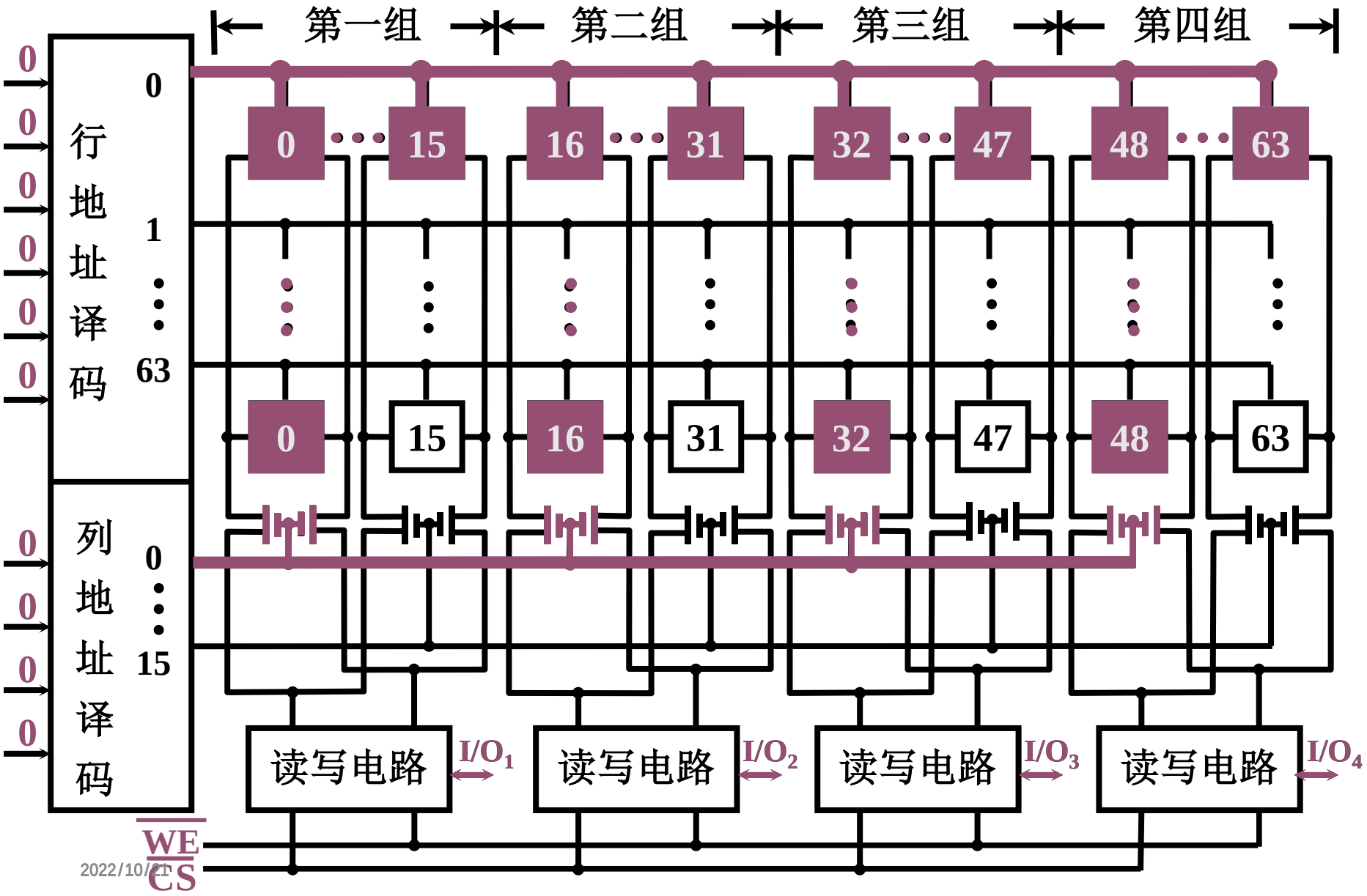
# ③ Intel 2114 RAM 矩阵 (64 × 64) 写





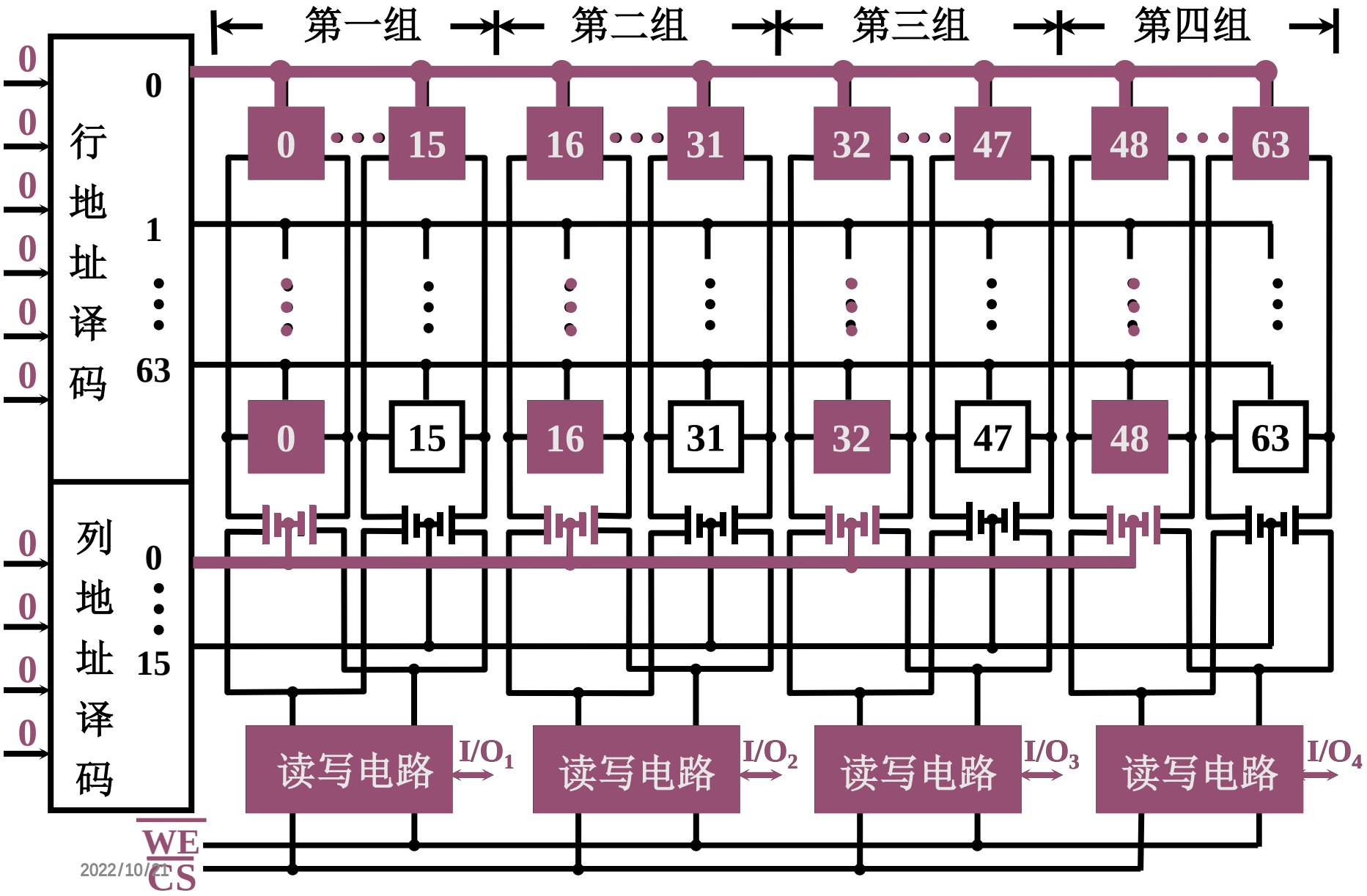
# ③ Intel 2114 RAM 矩阵 (64 × 64) 写





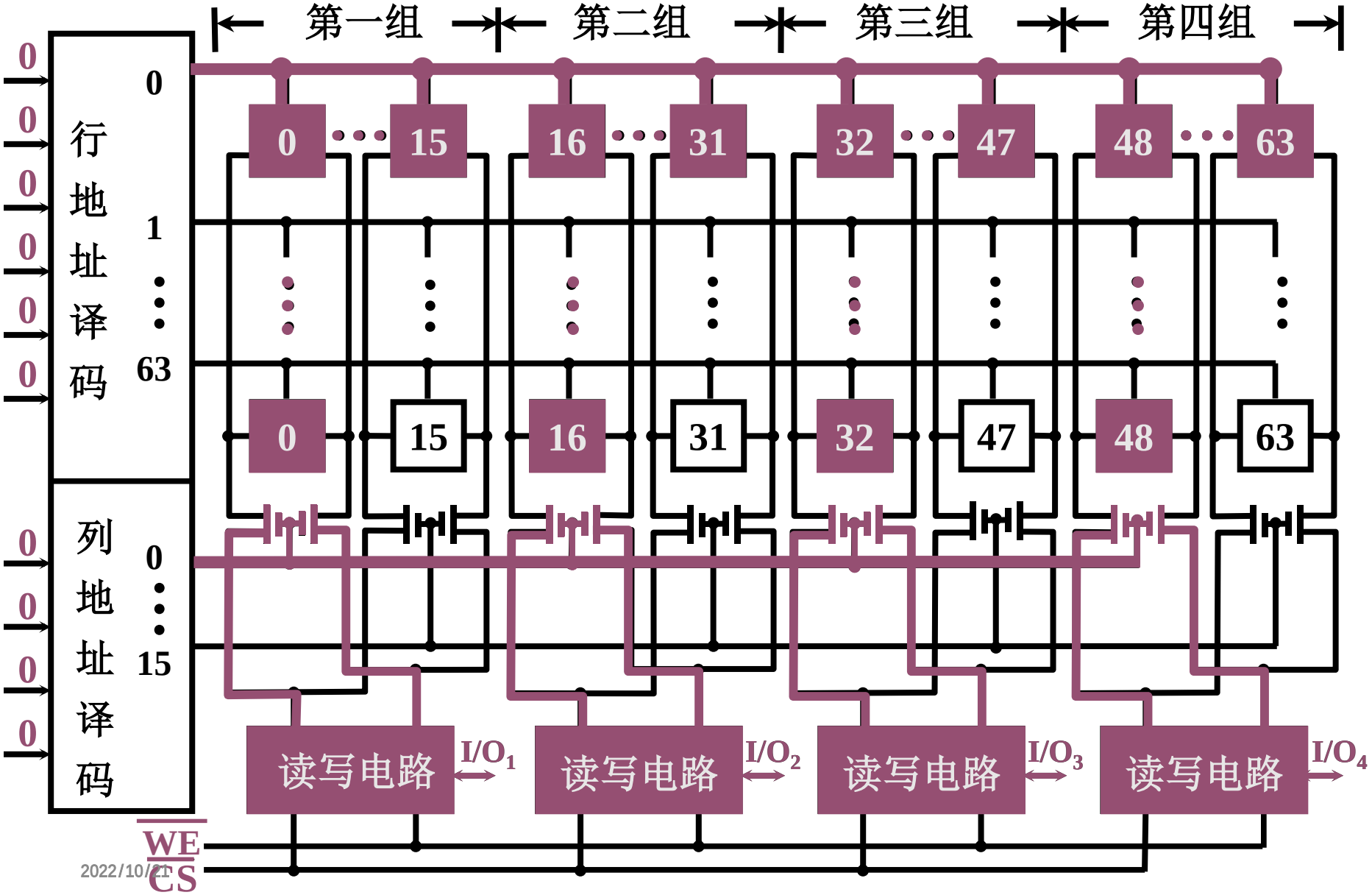
# ③ Intel 2114 RAM 矩阵 (64 × 64) 写





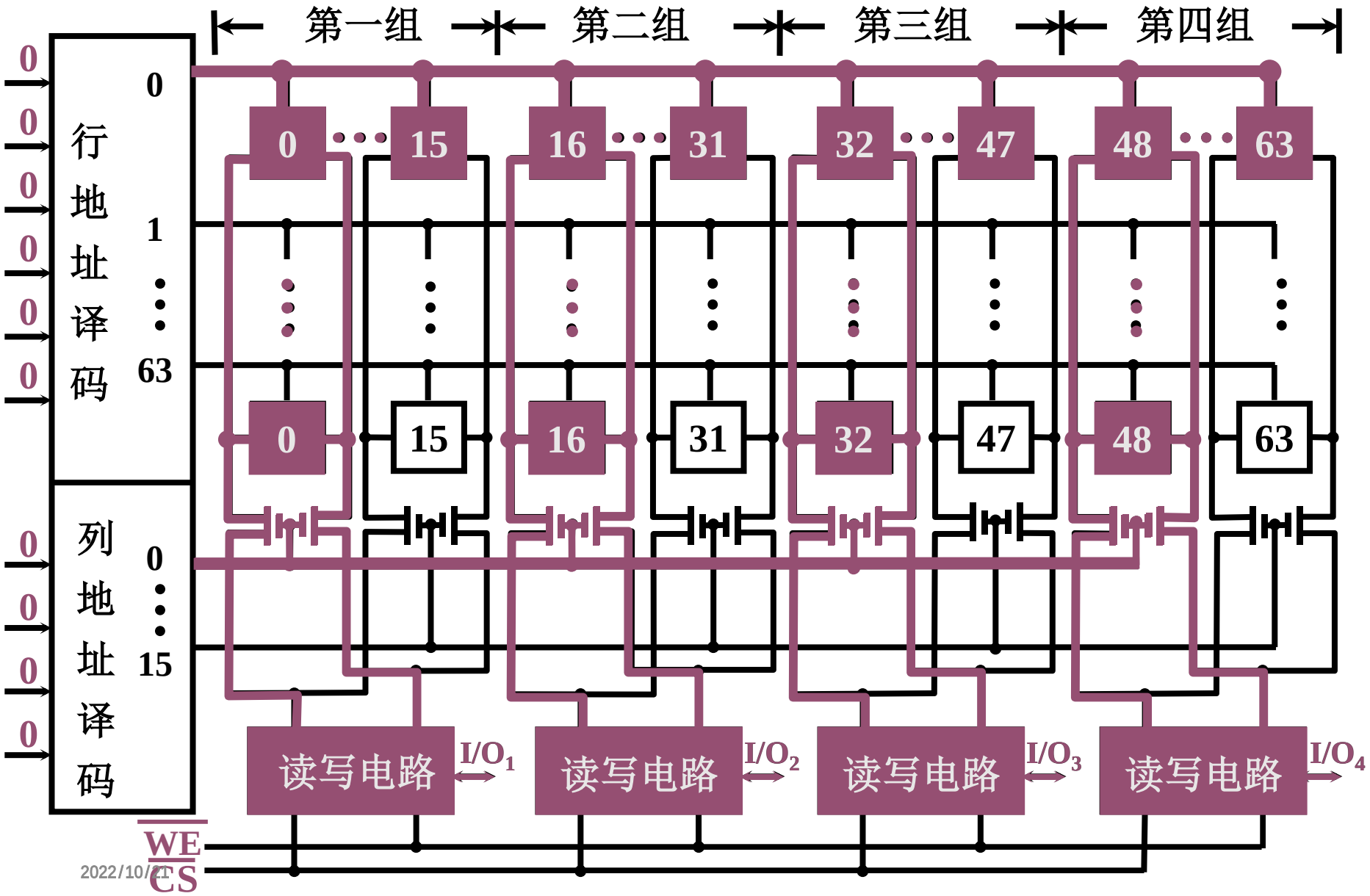
# ③ Intel 2114 RAM 矩阵 (64 × 64) 写

4.2



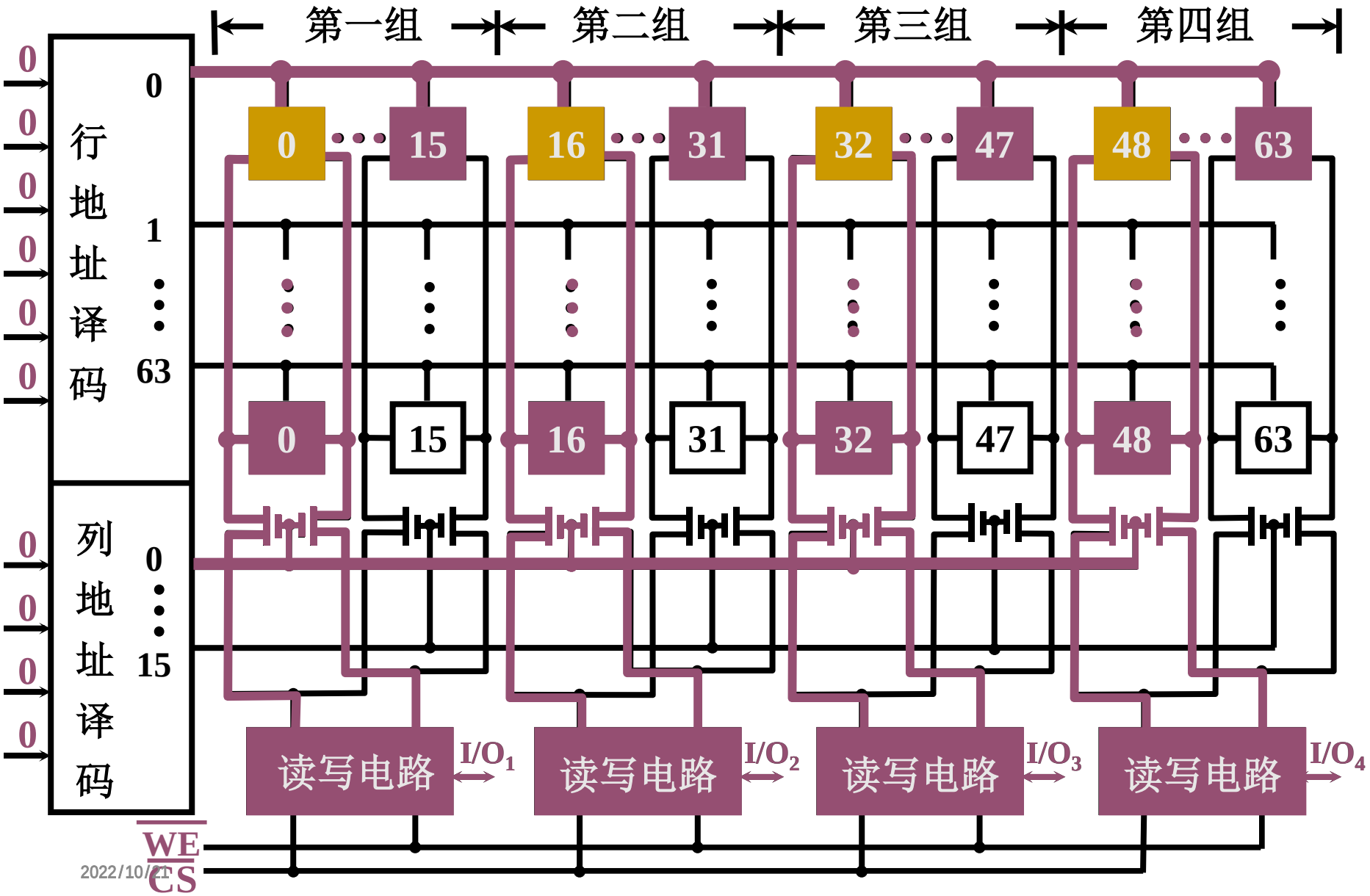
# ③ Intel 2114 RAM 矩阵 (64 × 64) 写





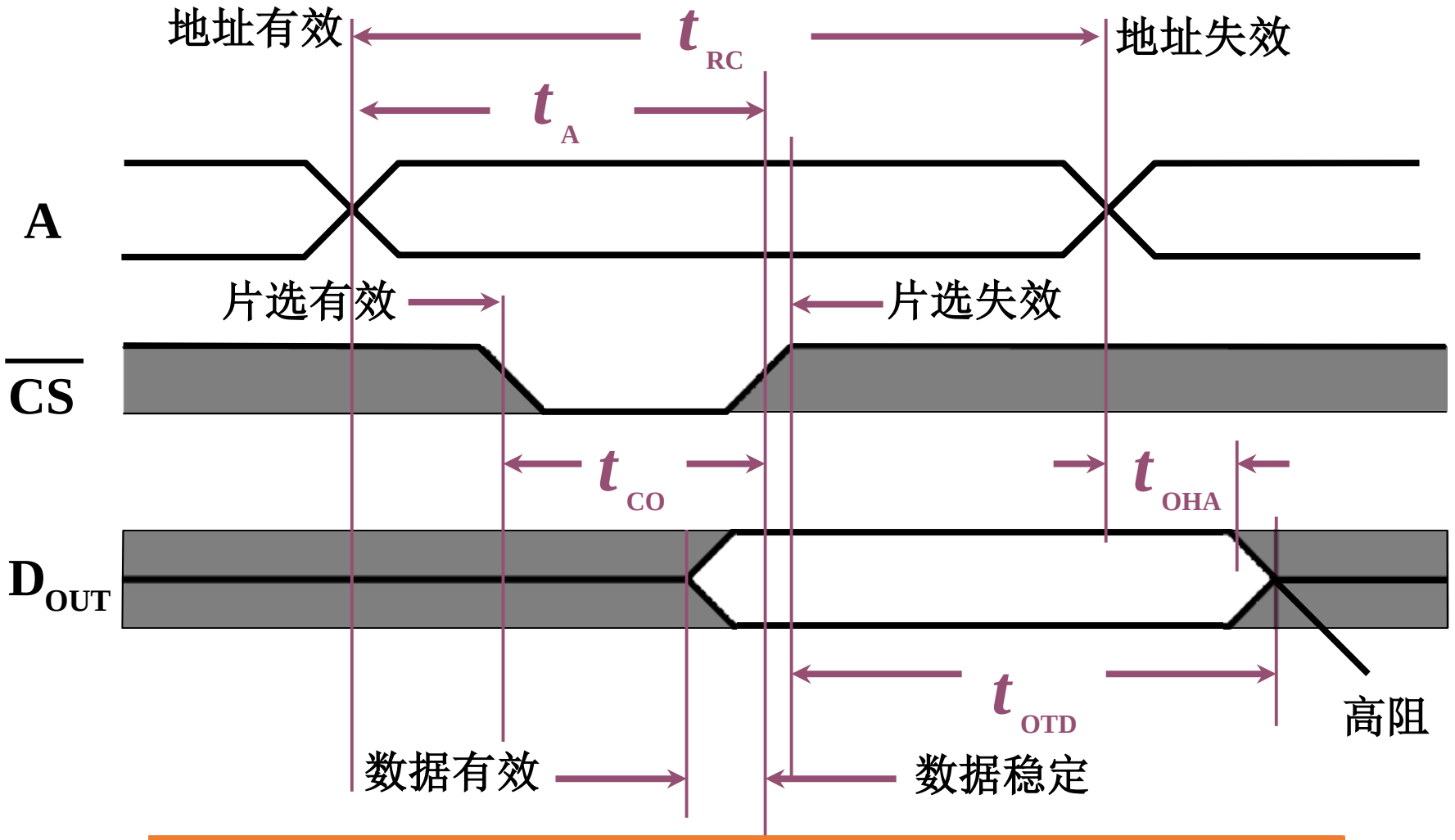
# ③ Intel 2114 RAM 矩阵 (64 × 64) 写

4.2



# (3) 静态 RAM 读 时序

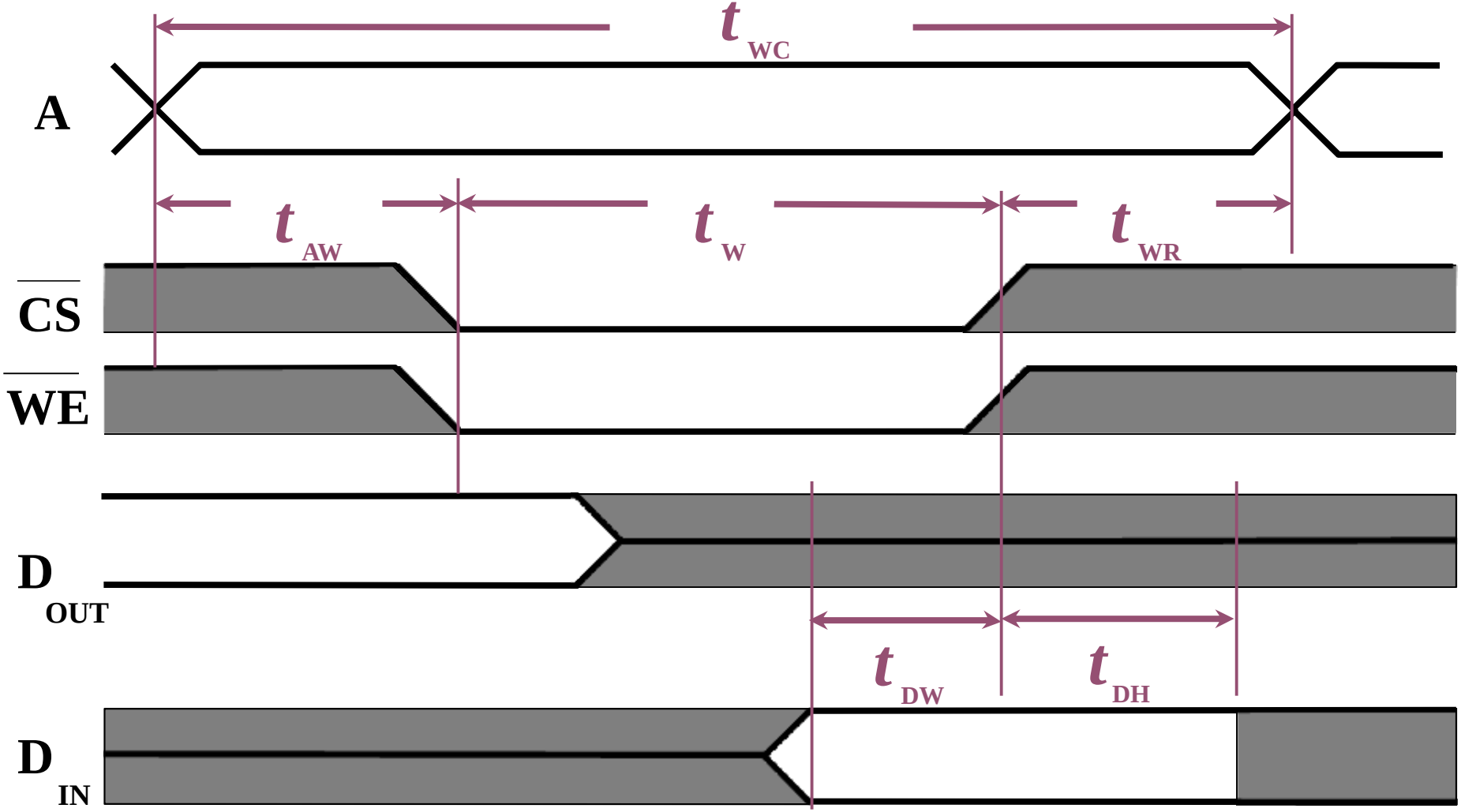





$t_{OHA}$ 地址失效后的 数据维持时间

# (4) 静态 RAM (2114) 写 时序

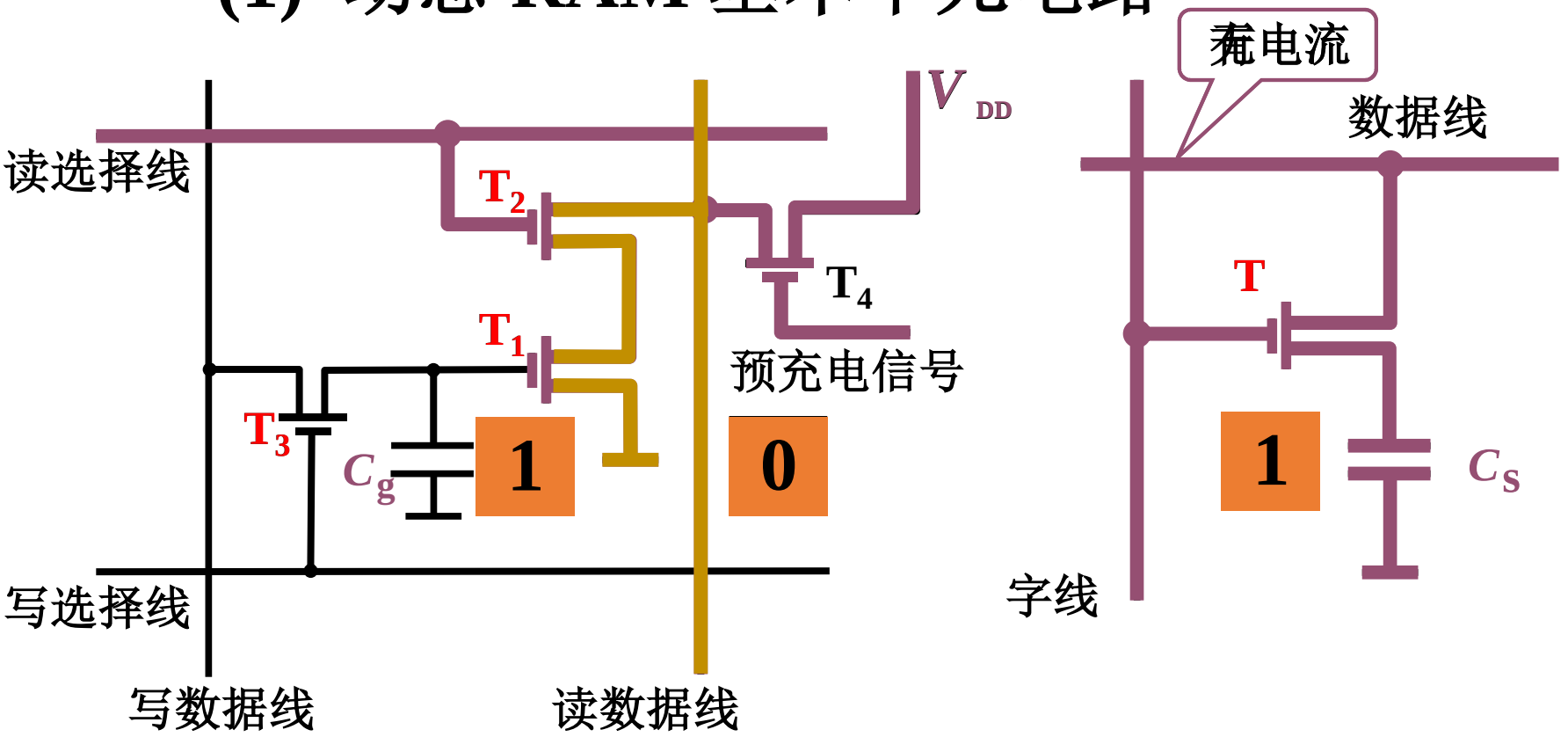





$t_{DH}$ $\overline{WE}$ 失效后的数据维持时间

# 2. 动态 RAM ( DRAM )

## (1) 动态 RAM 基本单元电路



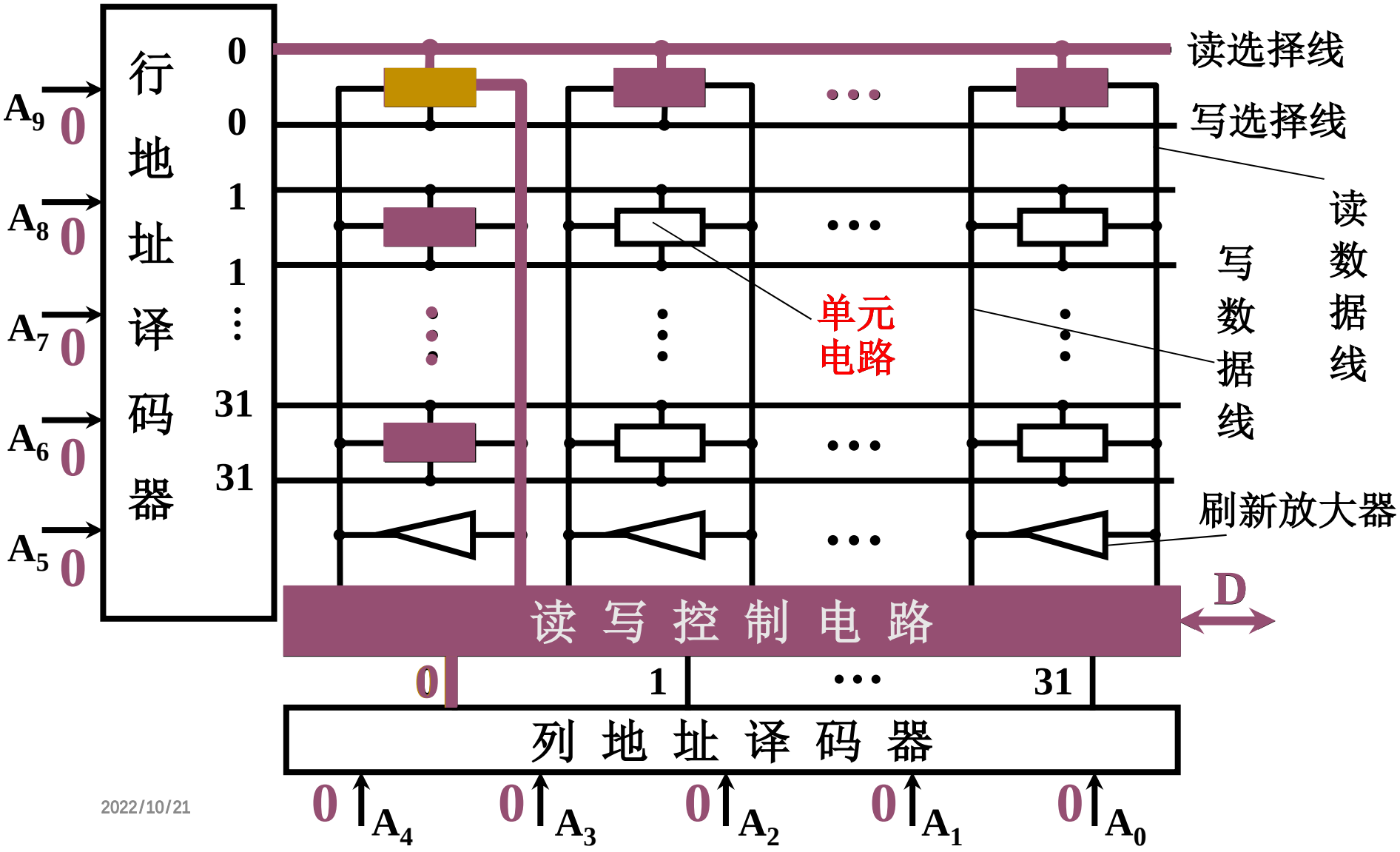

读出与原存信息相反　　读出时数据线有电流 为 “1”

写入与输入信息相同　　写入时 $C_S$ 充电为 “1” 放电为 “0”

# (2) 动态 RAM 芯片举例

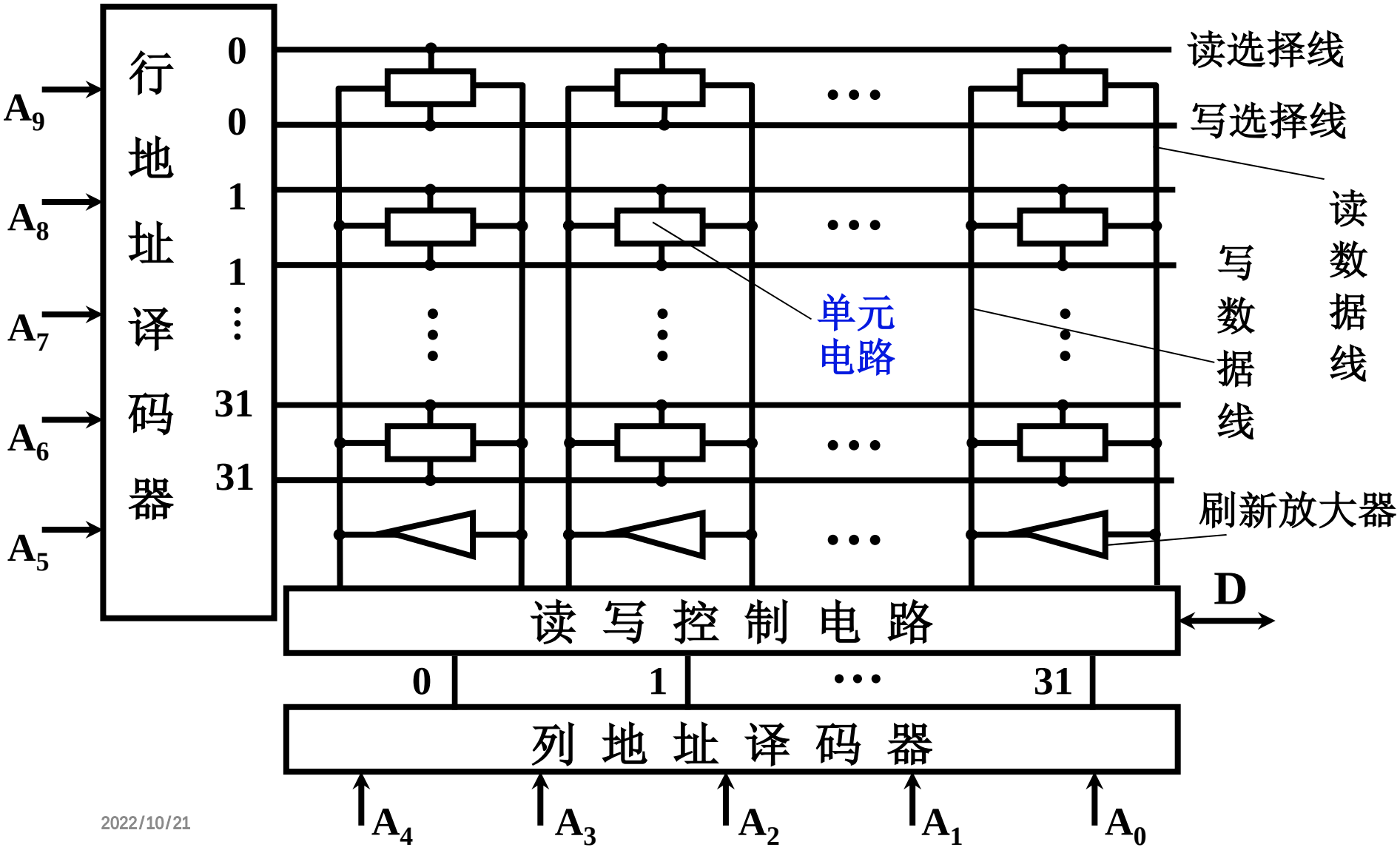
4.2

## ① 三管动态 RAM 芯片 (Intel 1103) 读



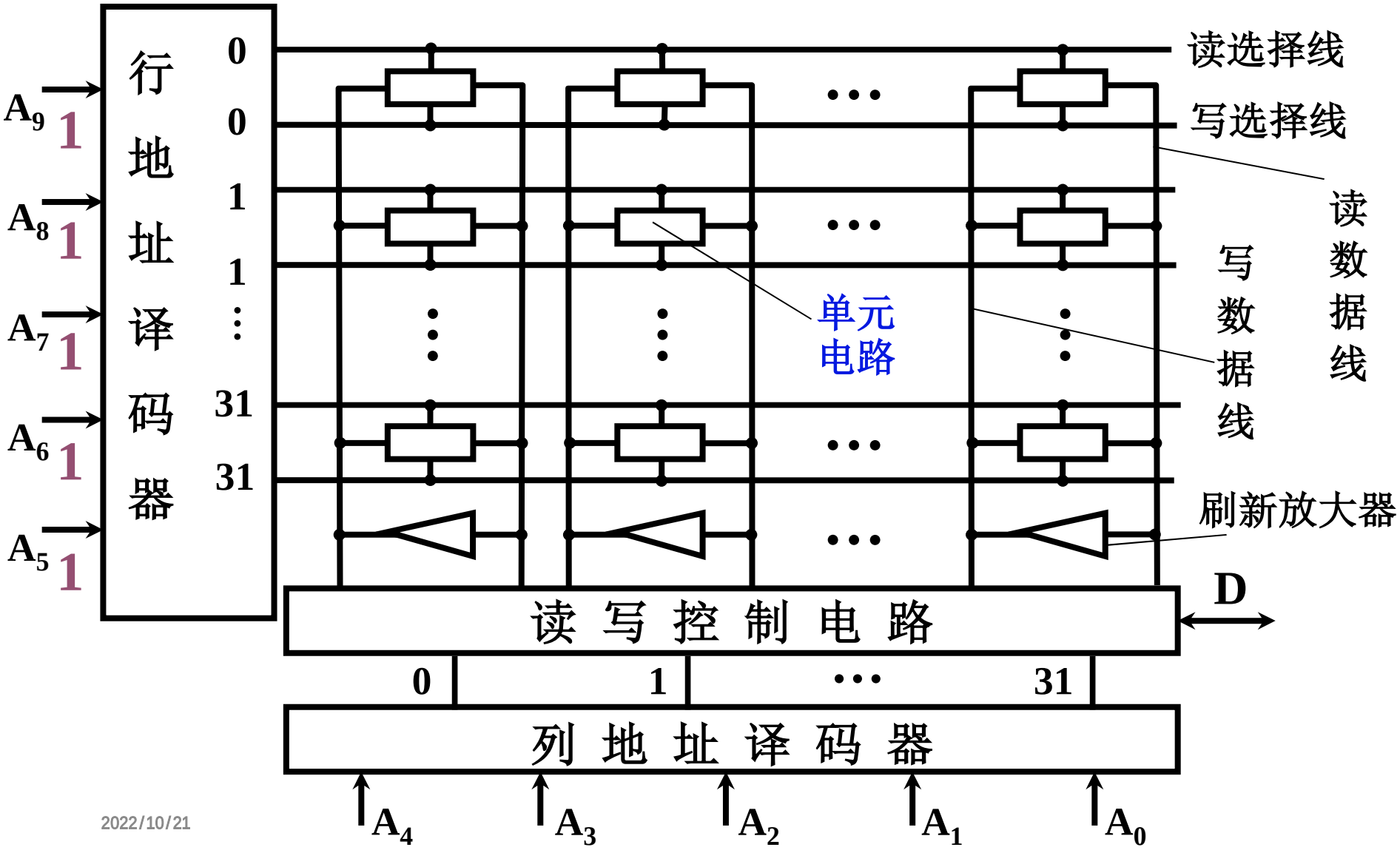
# ② 三管动态 RAM 芯片 (Intel 1103) 写





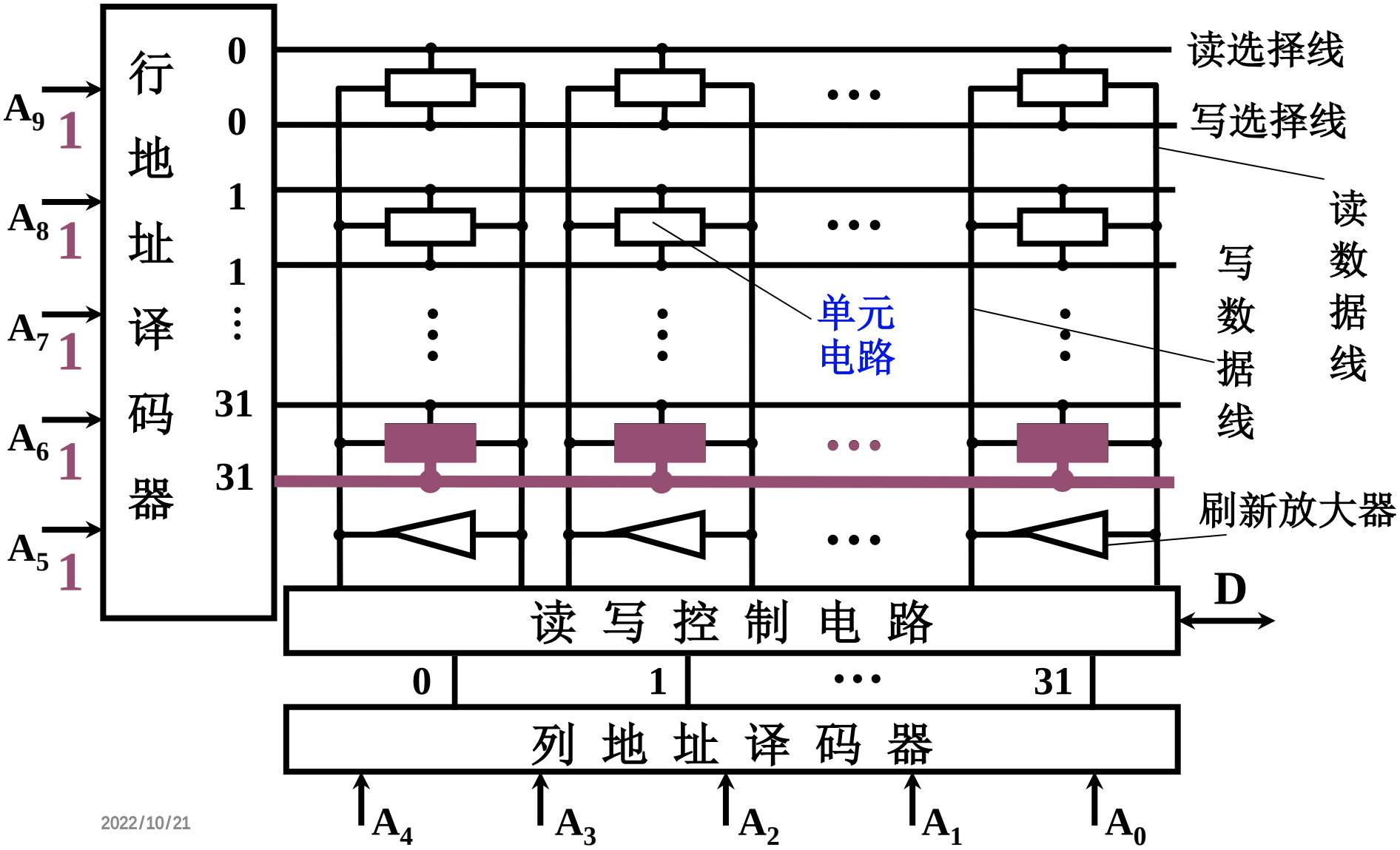
# ② 三管动态 RAM 芯片 (Intel 1103) 写





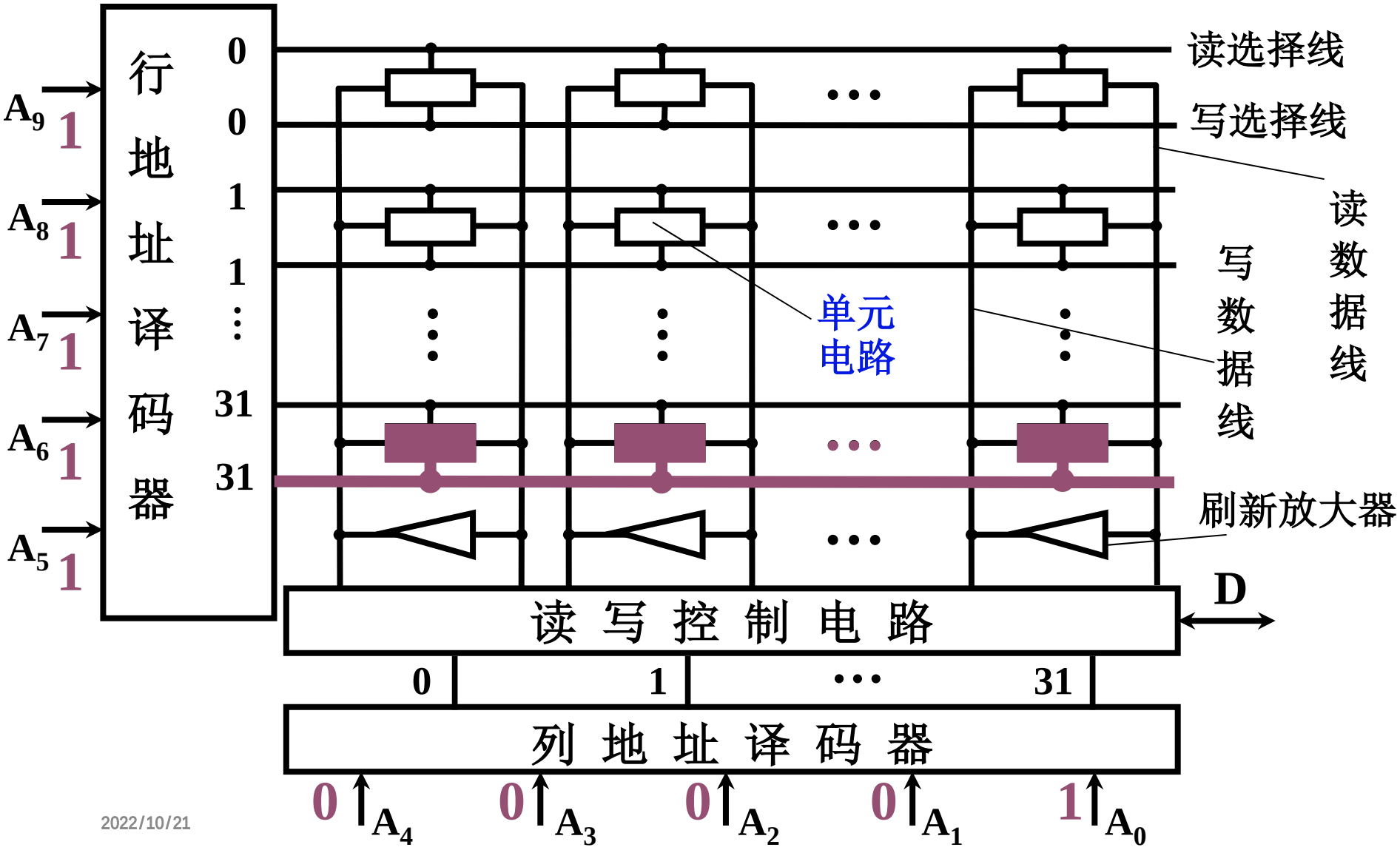
# ② 三管动态 RAM 芯片 (Intel 1103) 写





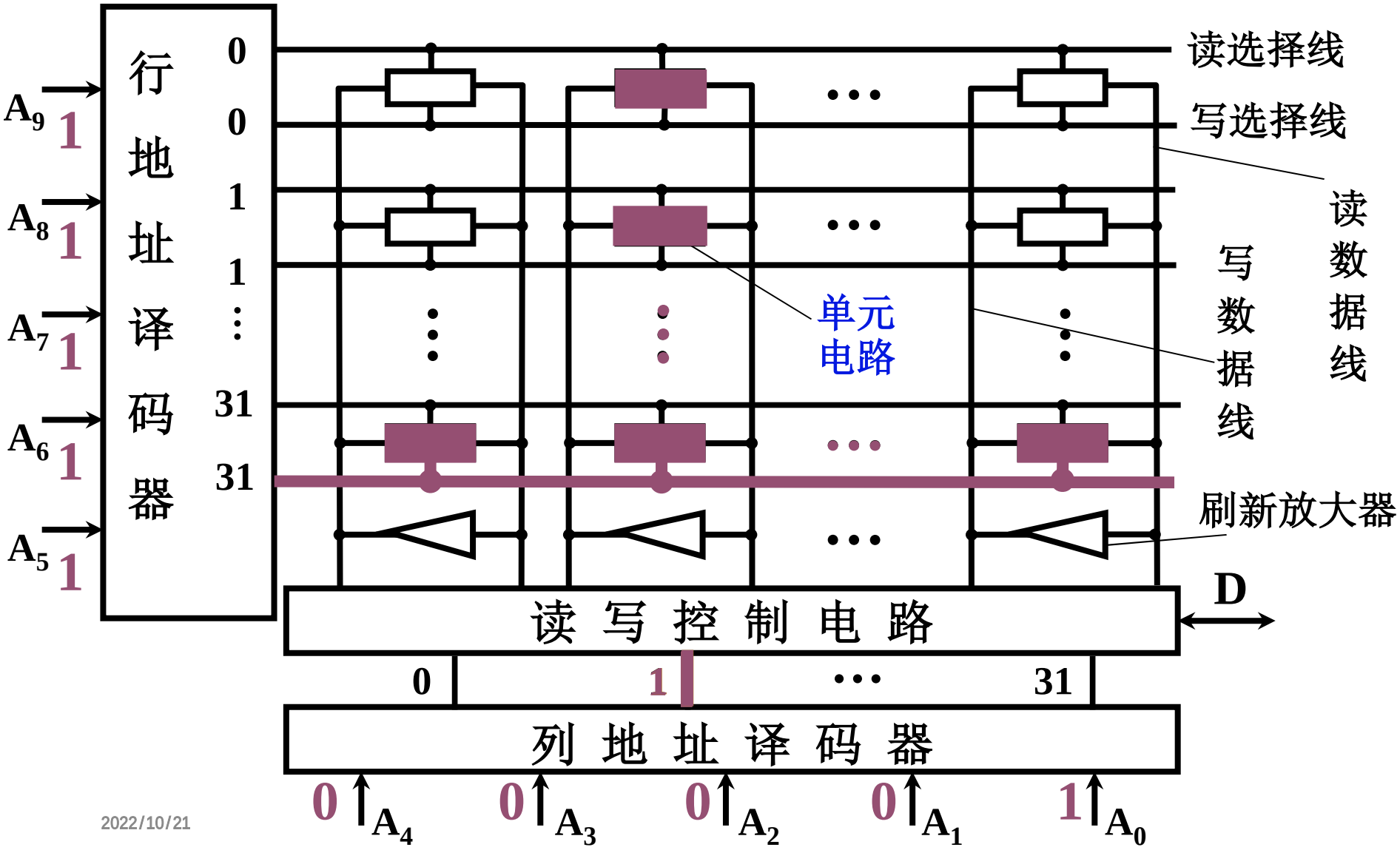
# ② 三管动态 RAM 芯片 (Intel 1103) 写





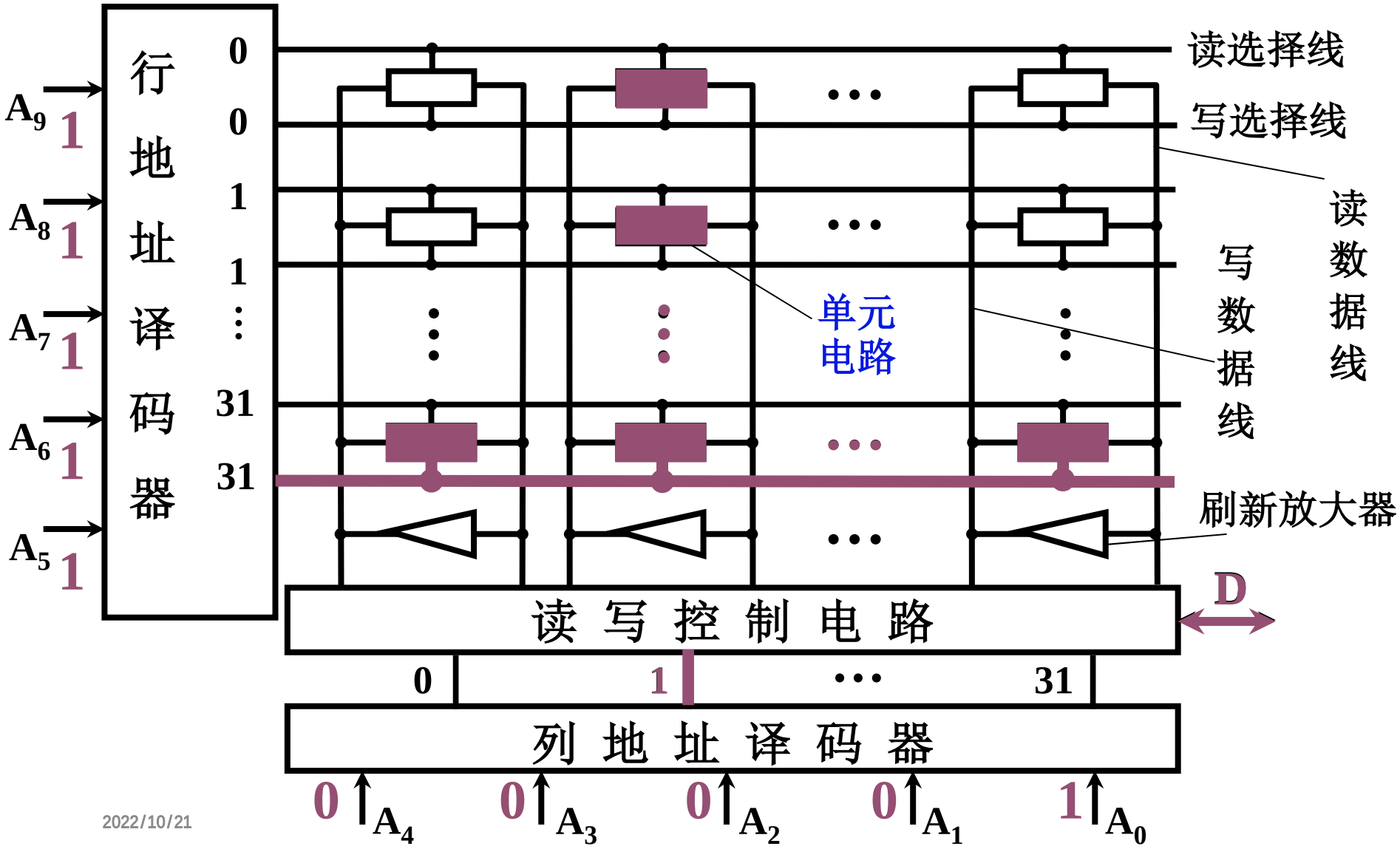
# ② 三管动态 RAM 芯片 (Intel 1103) 写





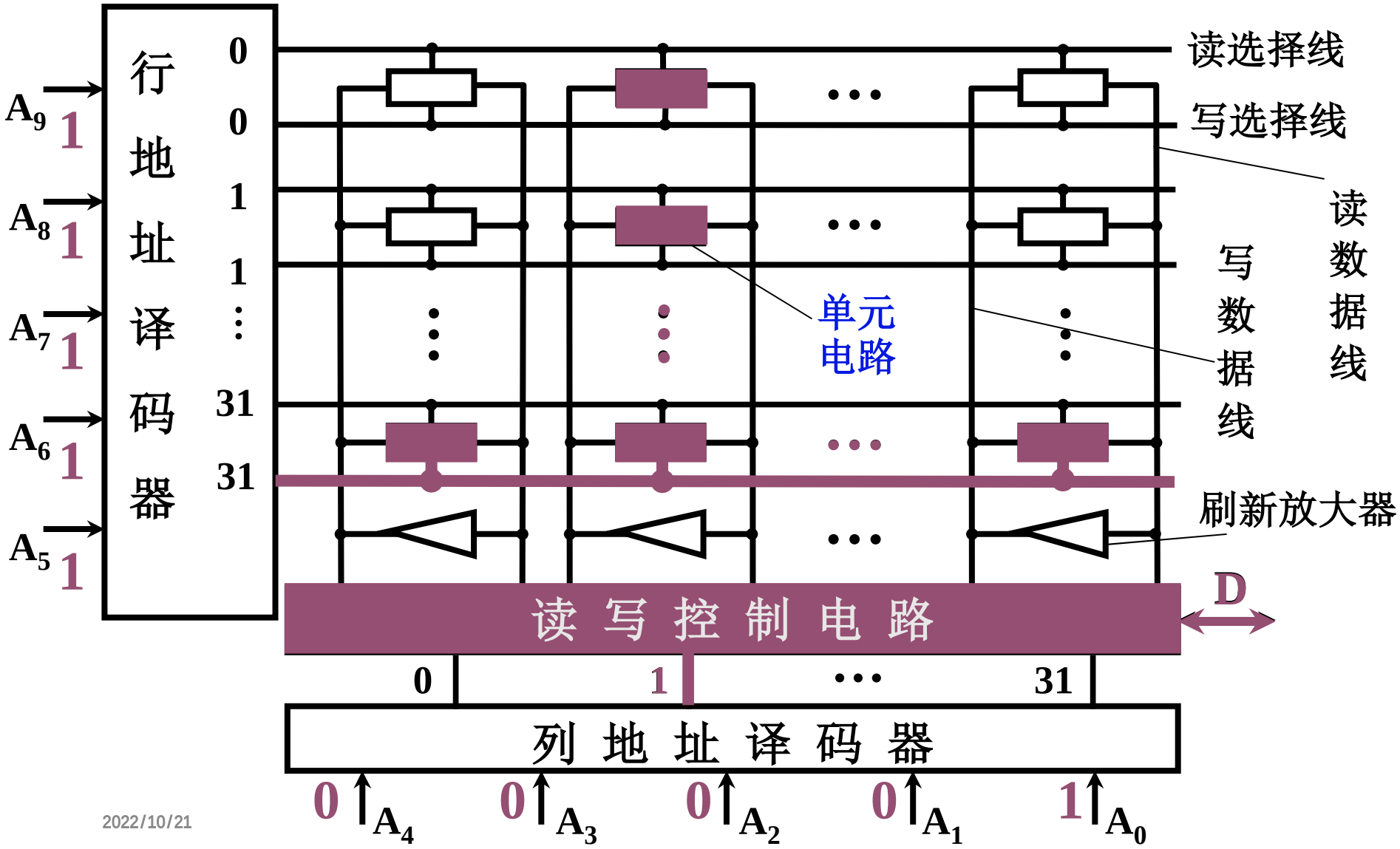
# ② 三管动态 RAM 芯片 (Intel 1103) 写





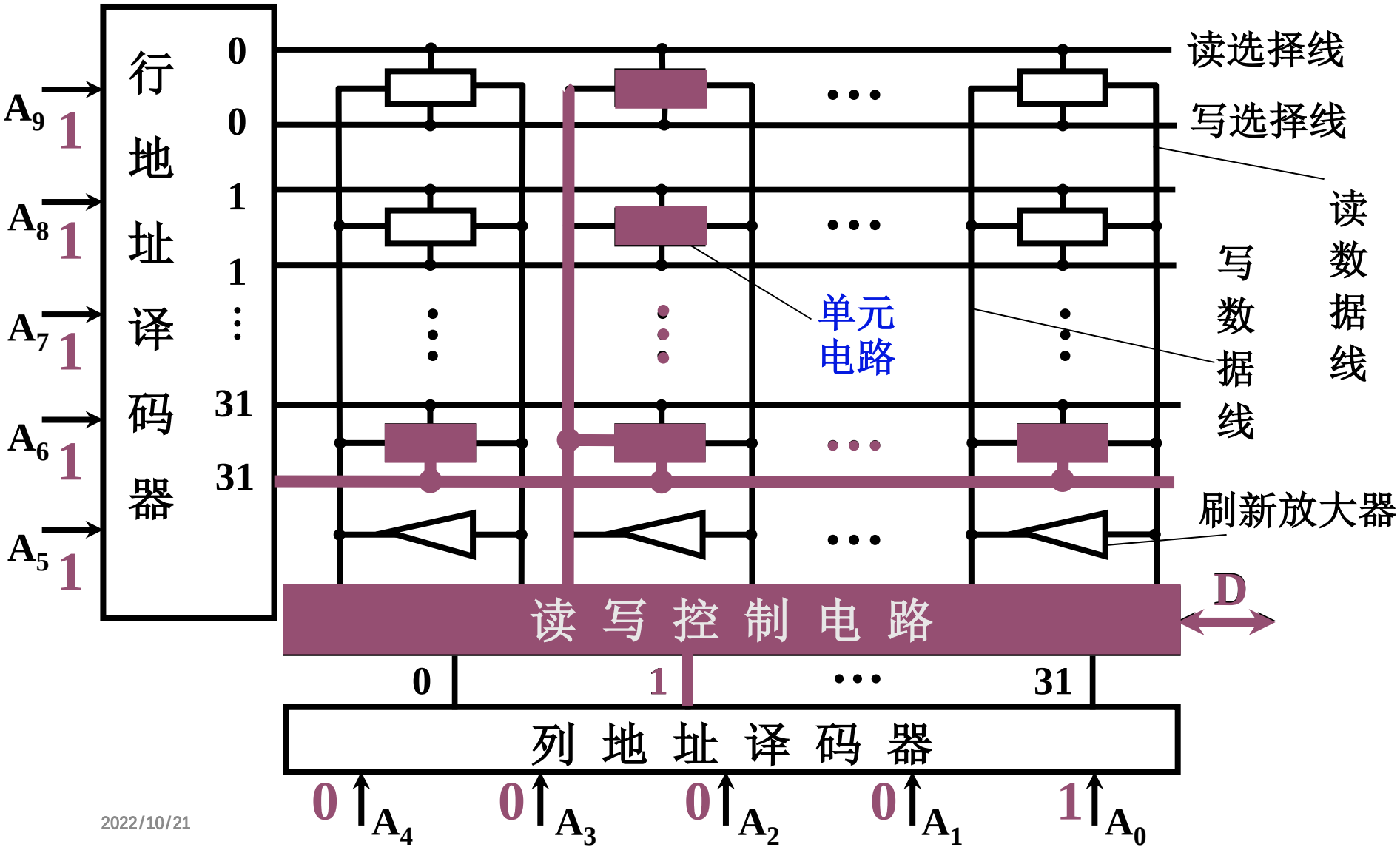
# ② 三管动态 RAM 芯片 (Intel 1103) 写





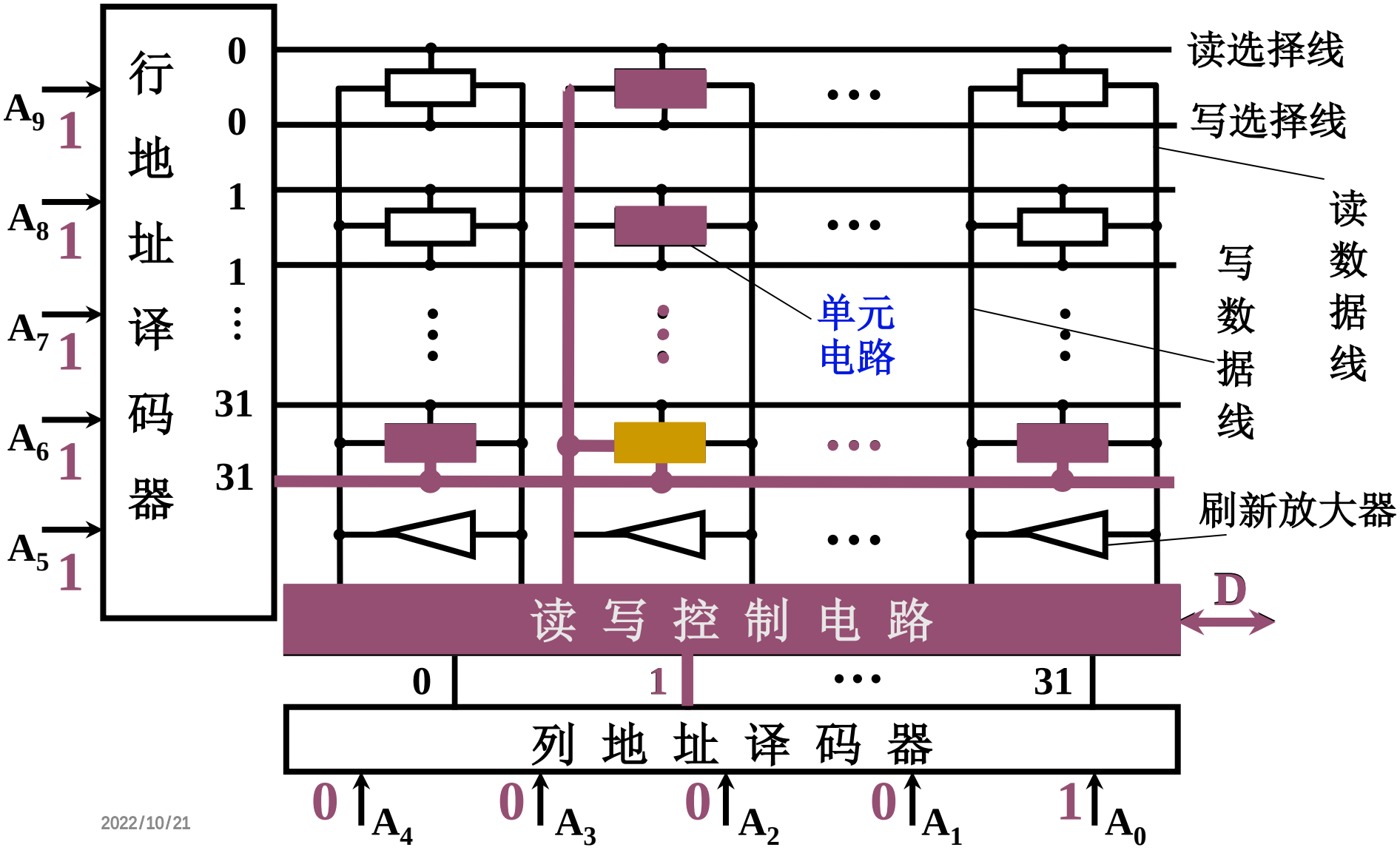
# ② 三管动态 RAM 芯片 (Intel 1103) 写



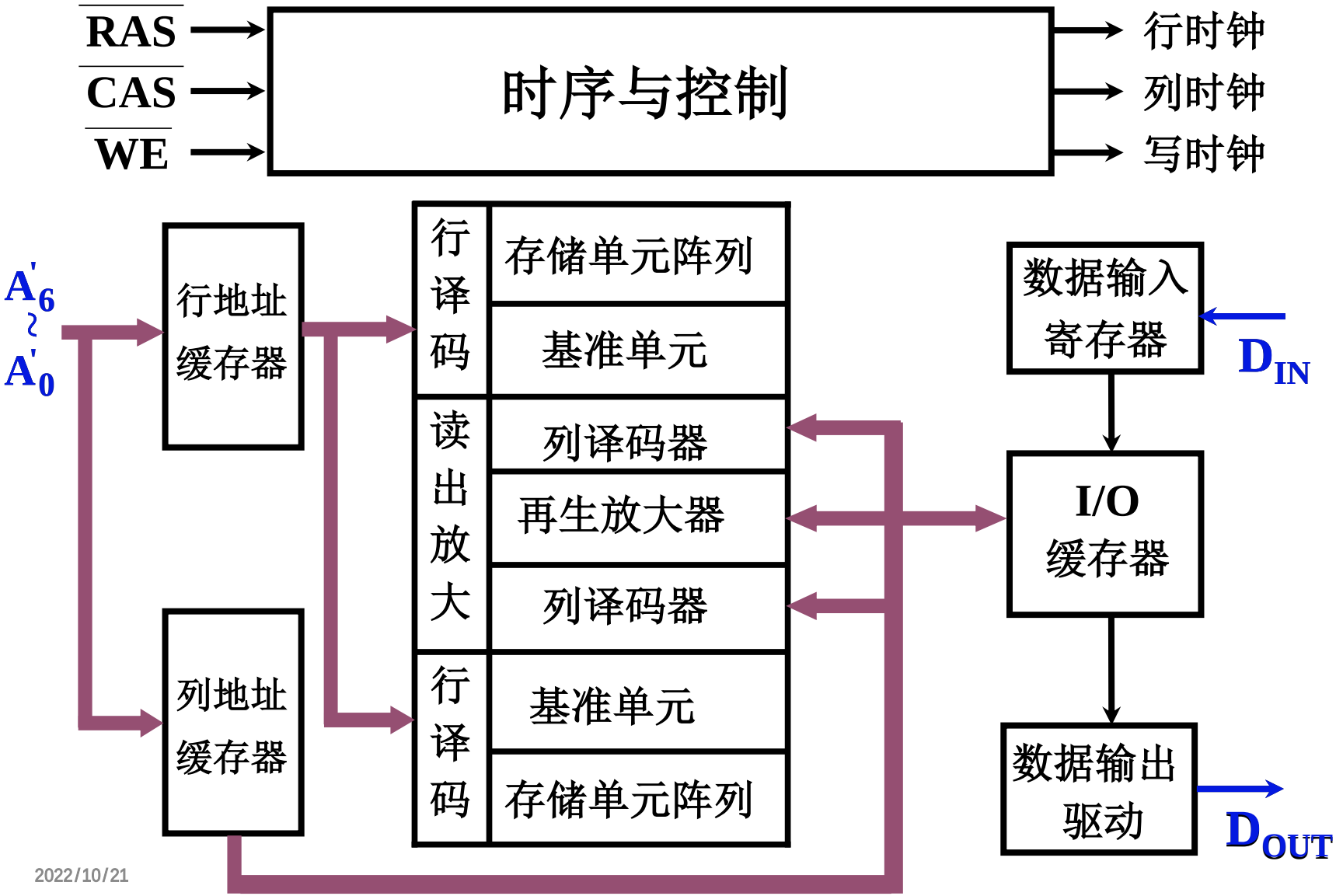


# ② 三管动态 RAM 芯片 (Intel 1103) 写

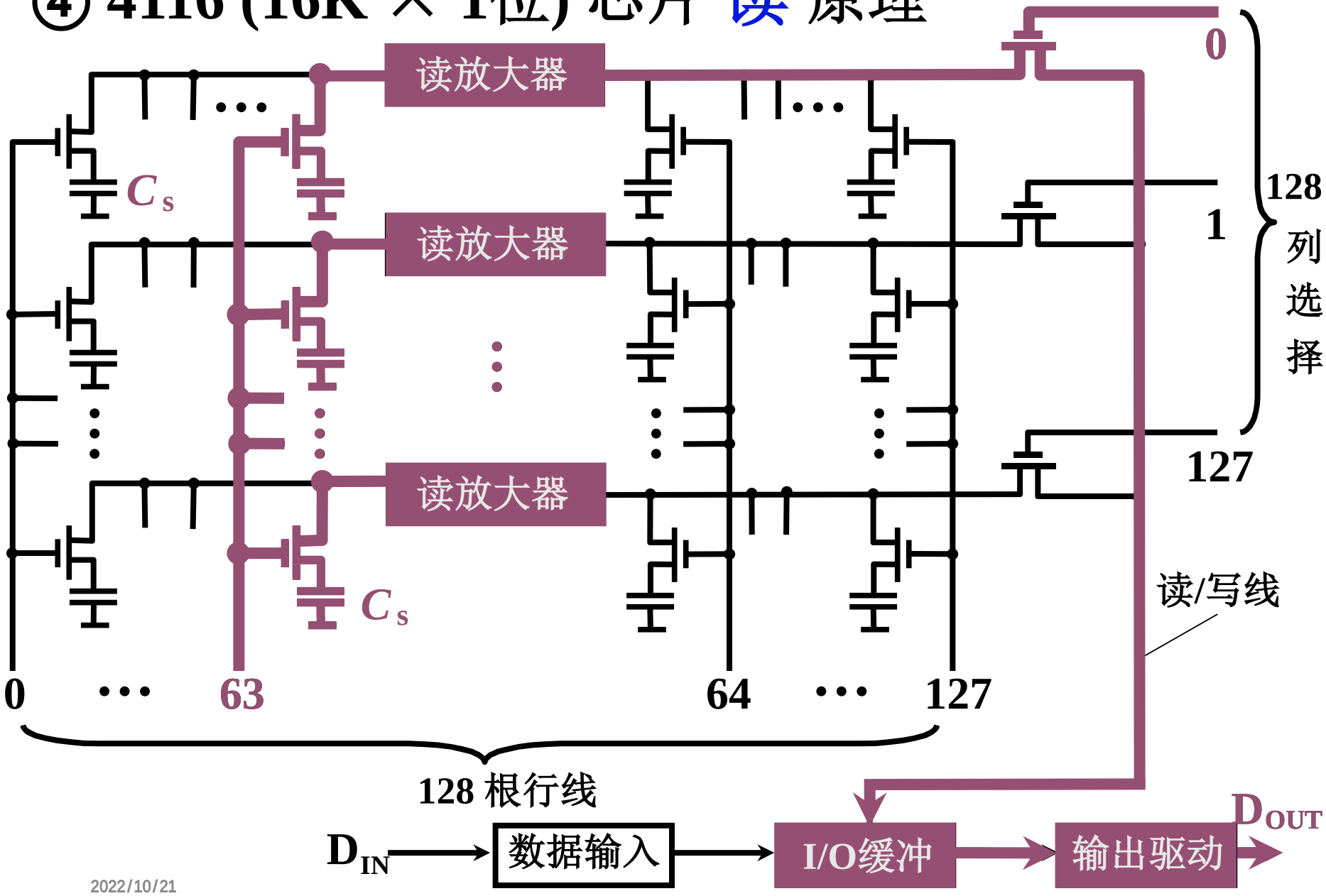




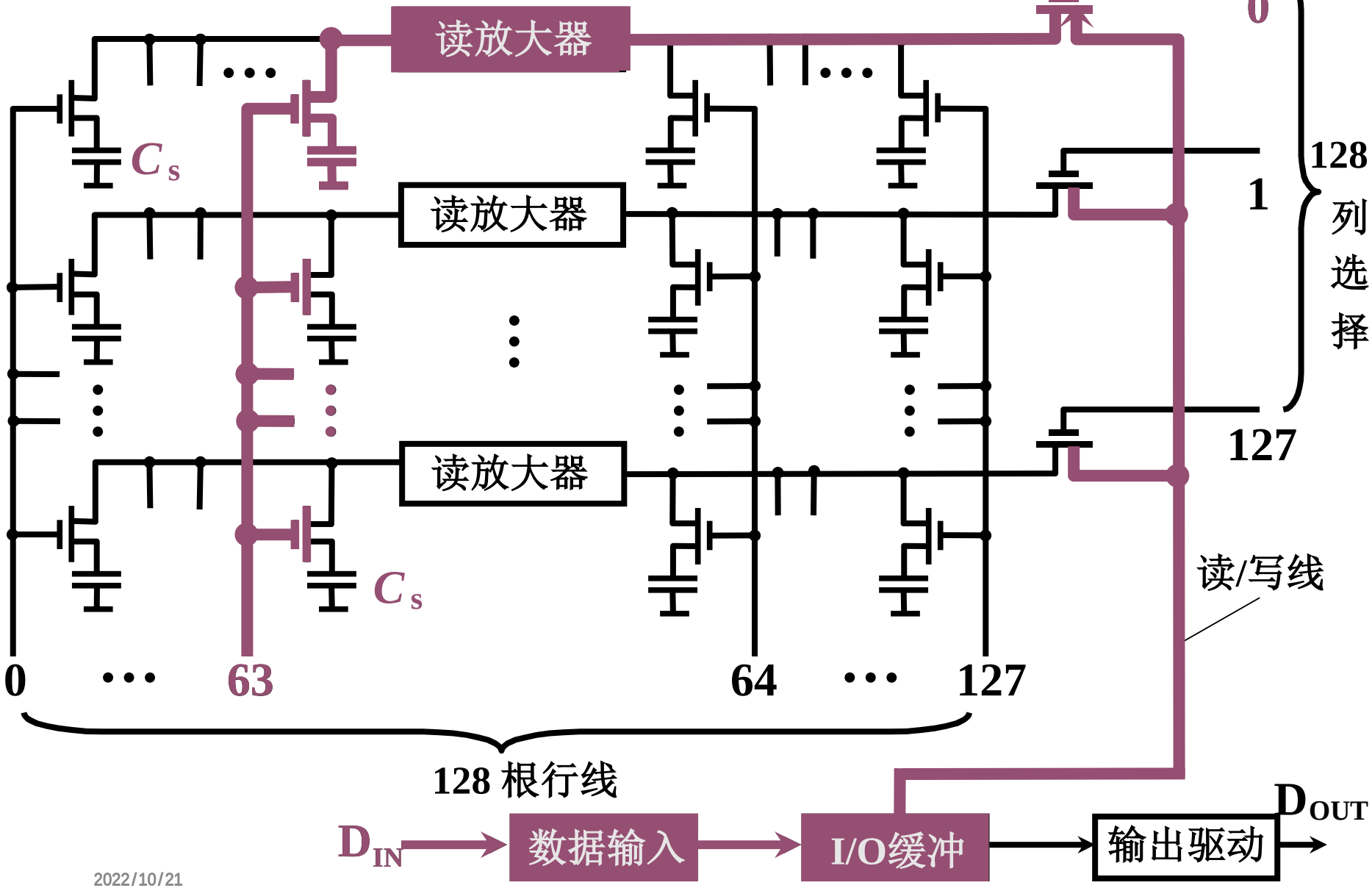
# ③ 单管动态 RAM 4116 (16K × 1位) 外特性 4.2



# ④ 4116 (16K × 1位) 芯片 读 原理



# ⑤ 4116 (16K×1位) 芯片 写 原理



# (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

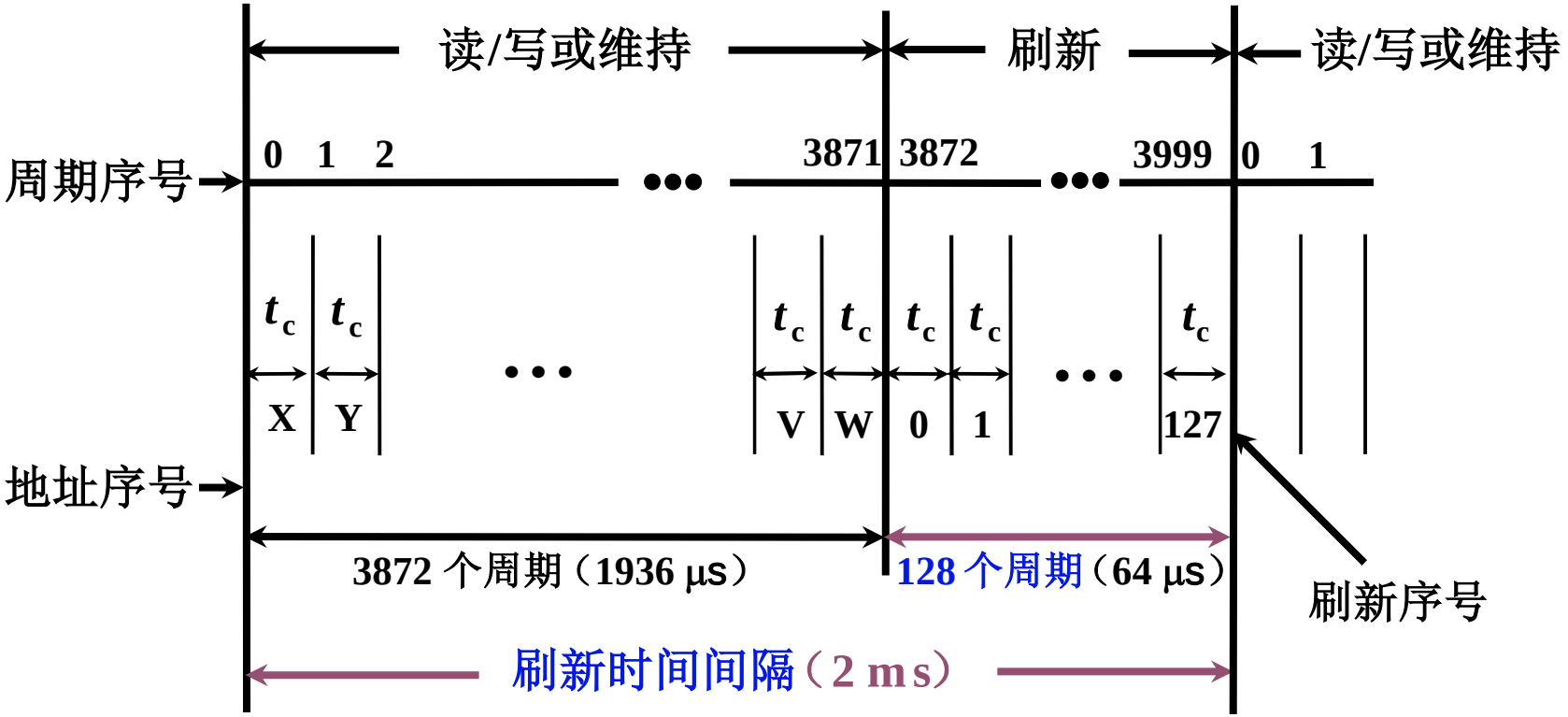
列地址 $\overline{\text{CAS}}$ 有效

# (4) 动态 RAM 刷新



## 刷新与行地址有关

① 集中刷新　(存取周期为0.5 μs )以128 × 128 矩阵为例



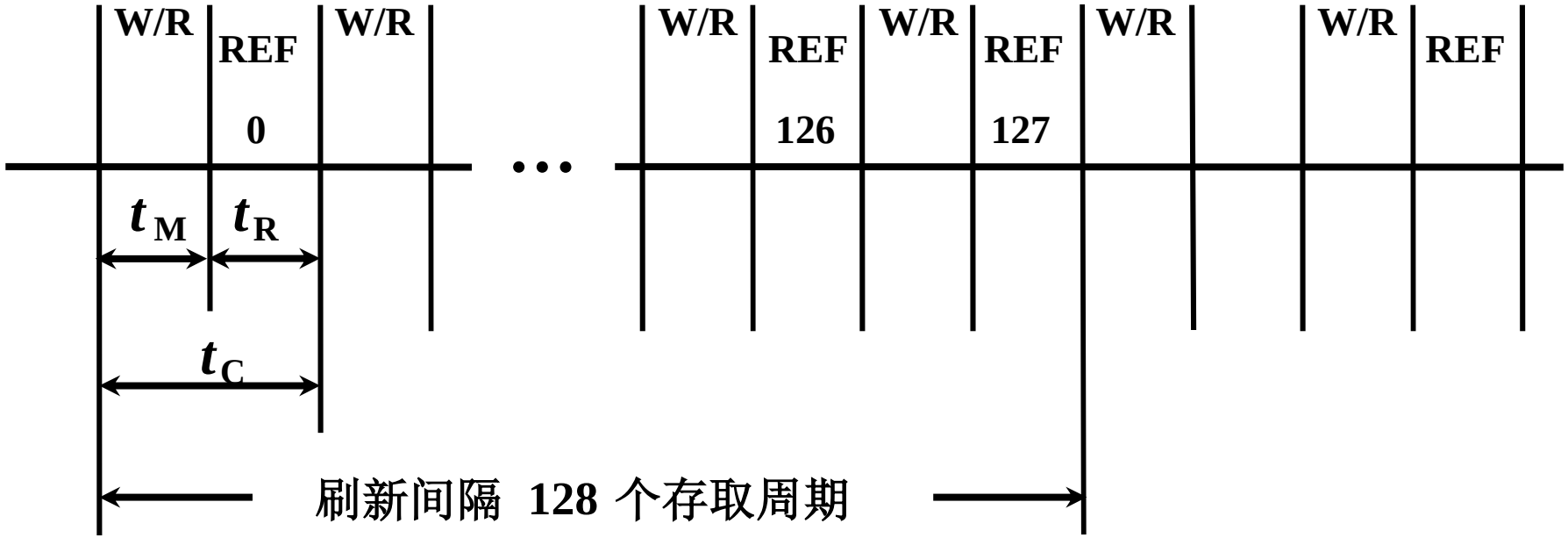

“死区”为　0.5 μs ×128 = 64 μs

“死时间率”为　128/4 000 ×100% = 3.2%

# ② 分散刷新（存取周期为1μs）



以 128 ×128 矩阵为例



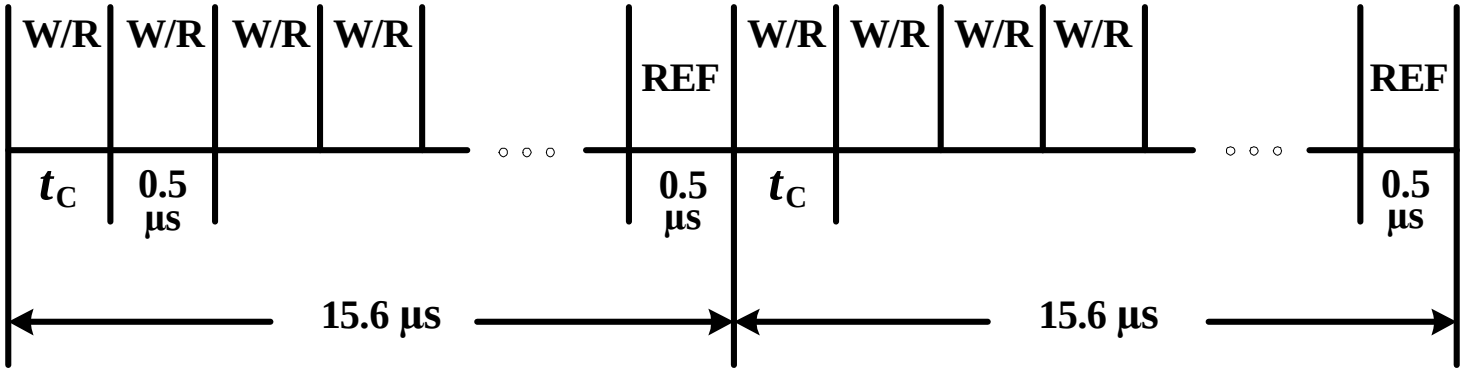

$t_C = t_M + t_R$　　无“死区”

读写　刷新　　(存取周期为 0.5 μs + 0.5 μs)

## ③ 分散刷新与集中刷新相结合（异步刷新）

对于 128 ×128 的存储芯片（存取周期为 0.5 μs）

若每隔 15.6 μs 刷新一行




每行每隔 2 ms 刷新一次　　“死区”为 0.5 μs

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

缓存

| | DRAM | SRAM |
|---|---|---|
| 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 |

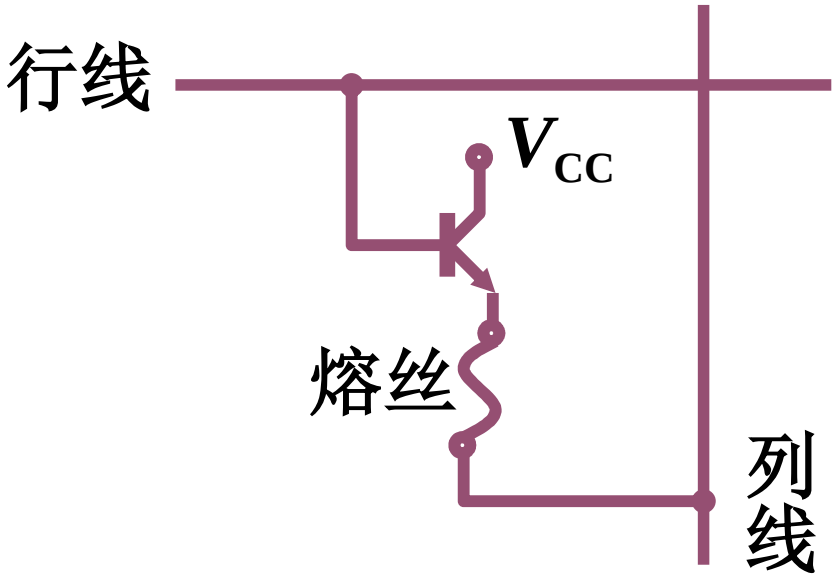
# 四、只读存储器（ROM）

## 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)

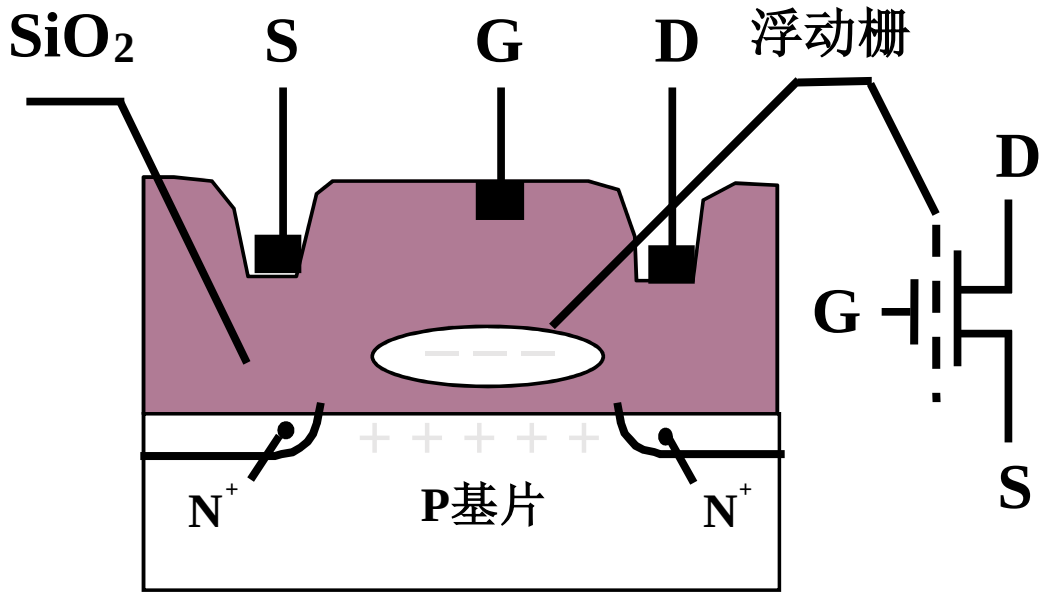



熔丝断　　为“0”

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

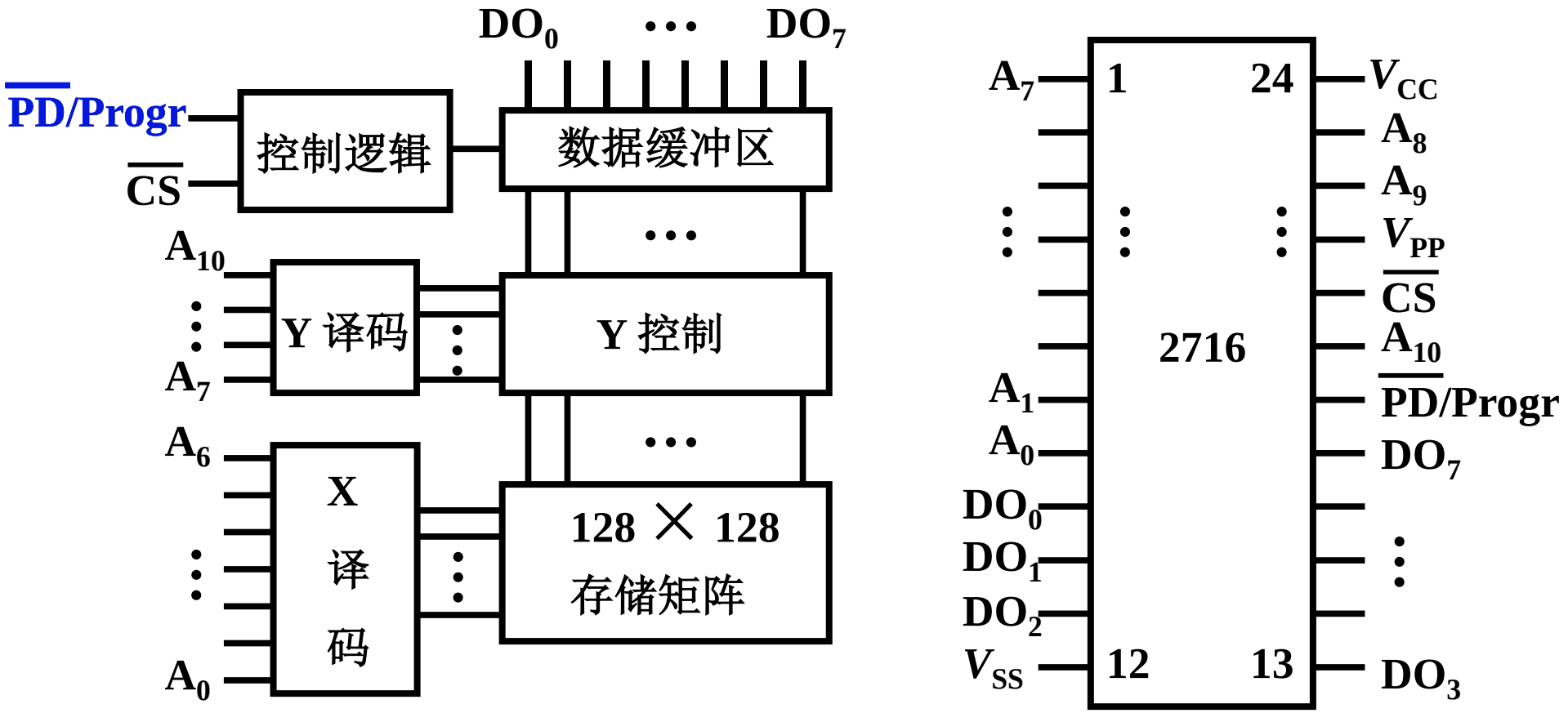
紫外线全部擦洗

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






$\overline{PD}$/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** 电可擦洗重写

比 **EEPROM**快 具备 **RAM** 功能

# 五、存储器与 CPU 的连接



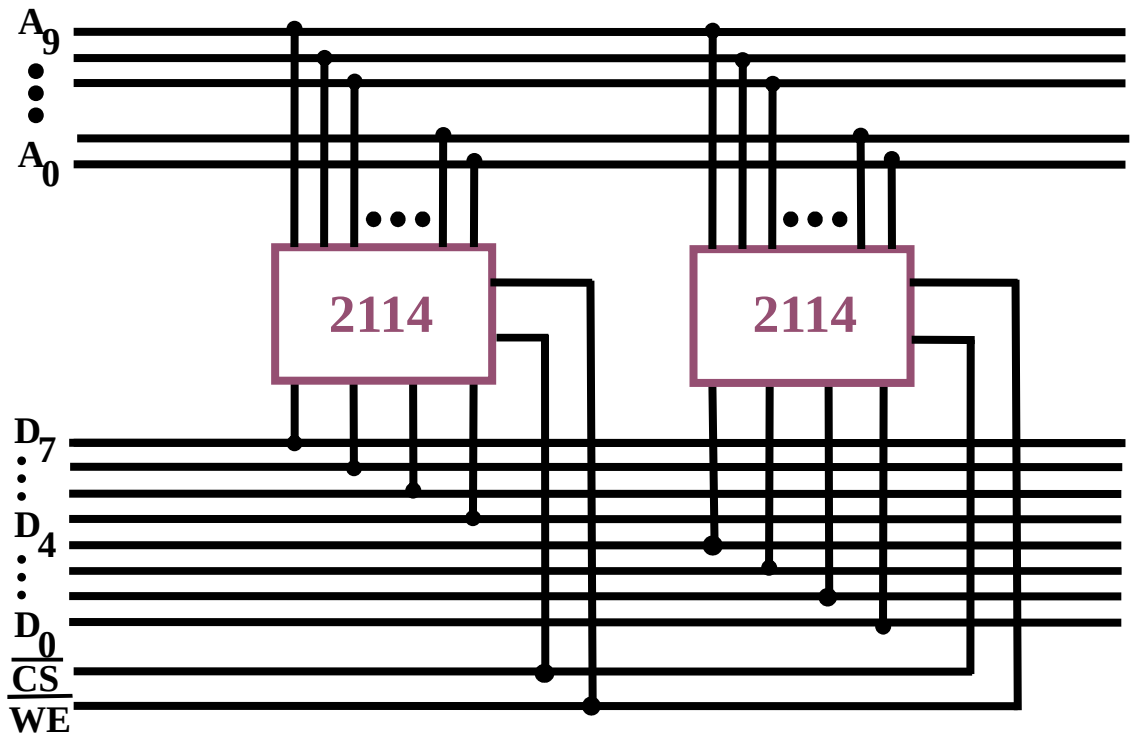
## 1. 存储器容量的扩展

### (1) 位扩展（增加存储字长）

用 ?2片 1K×4位 存储芯片组成 1K×8位 的存储器

10根地址线

8根数据线

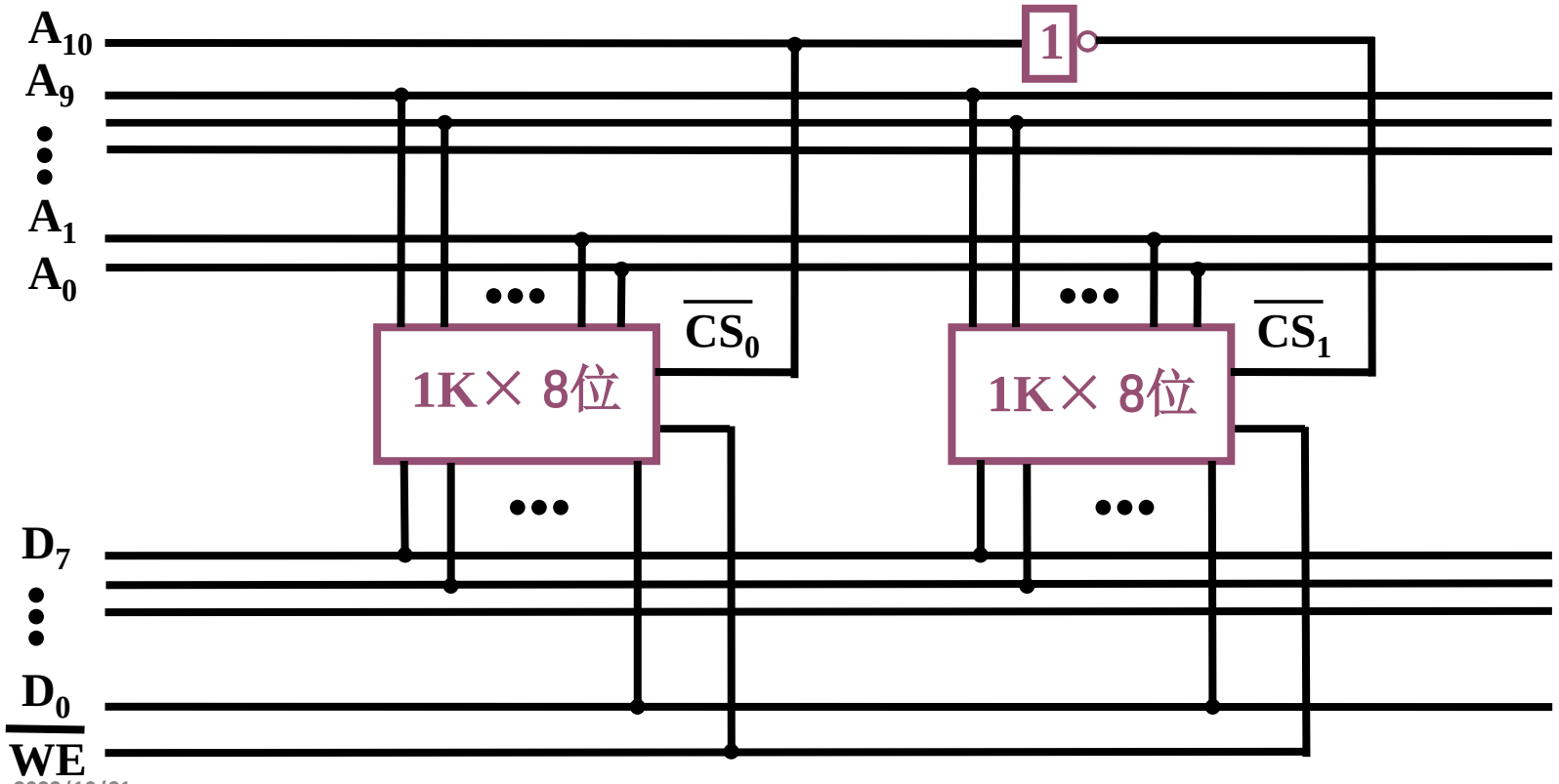


# (2) 字扩展（增加存储字的数量）

11根地址线

用 2片 1K×8位 存储芯片组成 2K×8位 的存储器

8根数据线



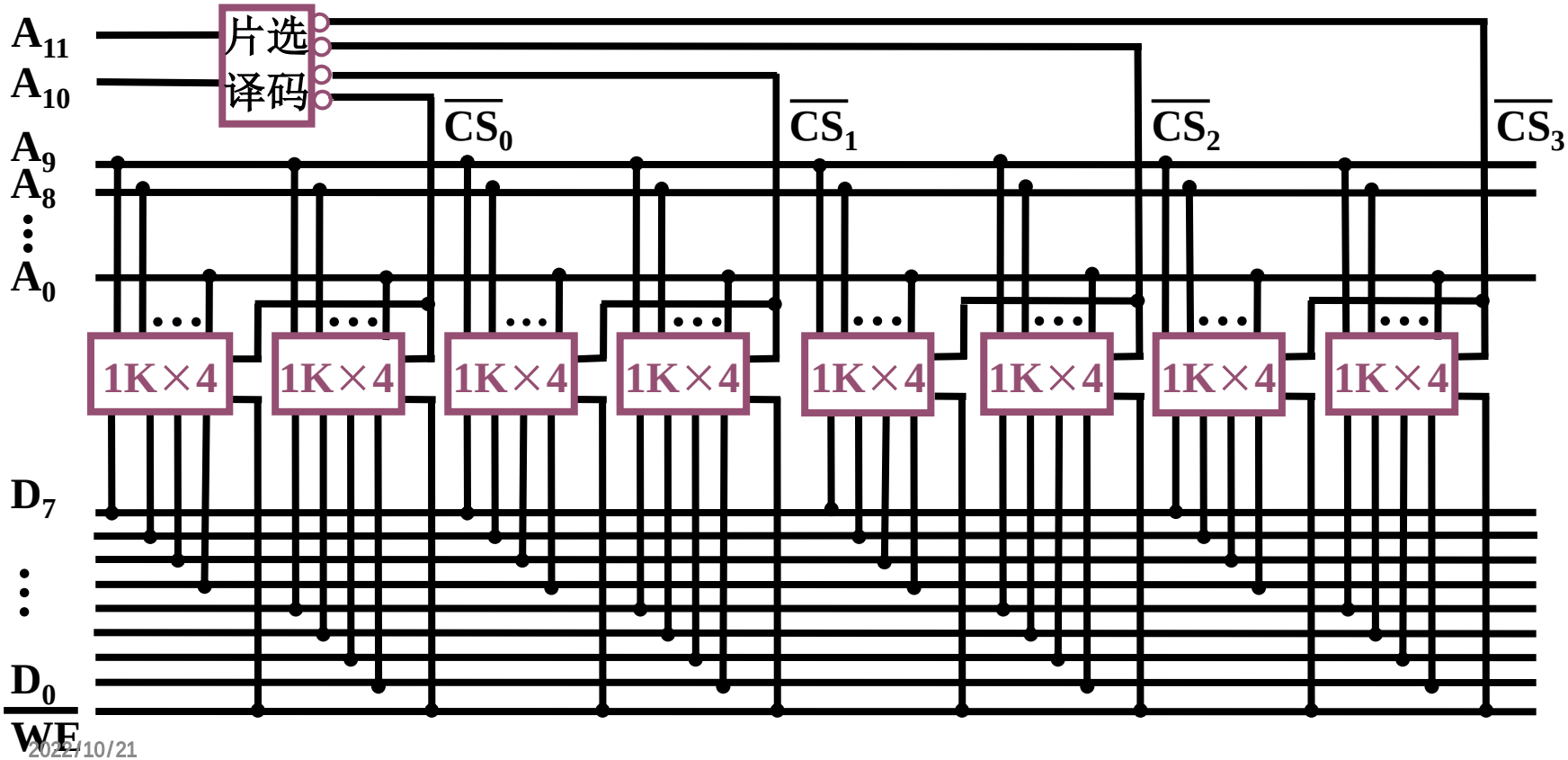
# (3) 字、位扩展



用 ?8片 1K × 4位 存储芯片组成 4K × 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

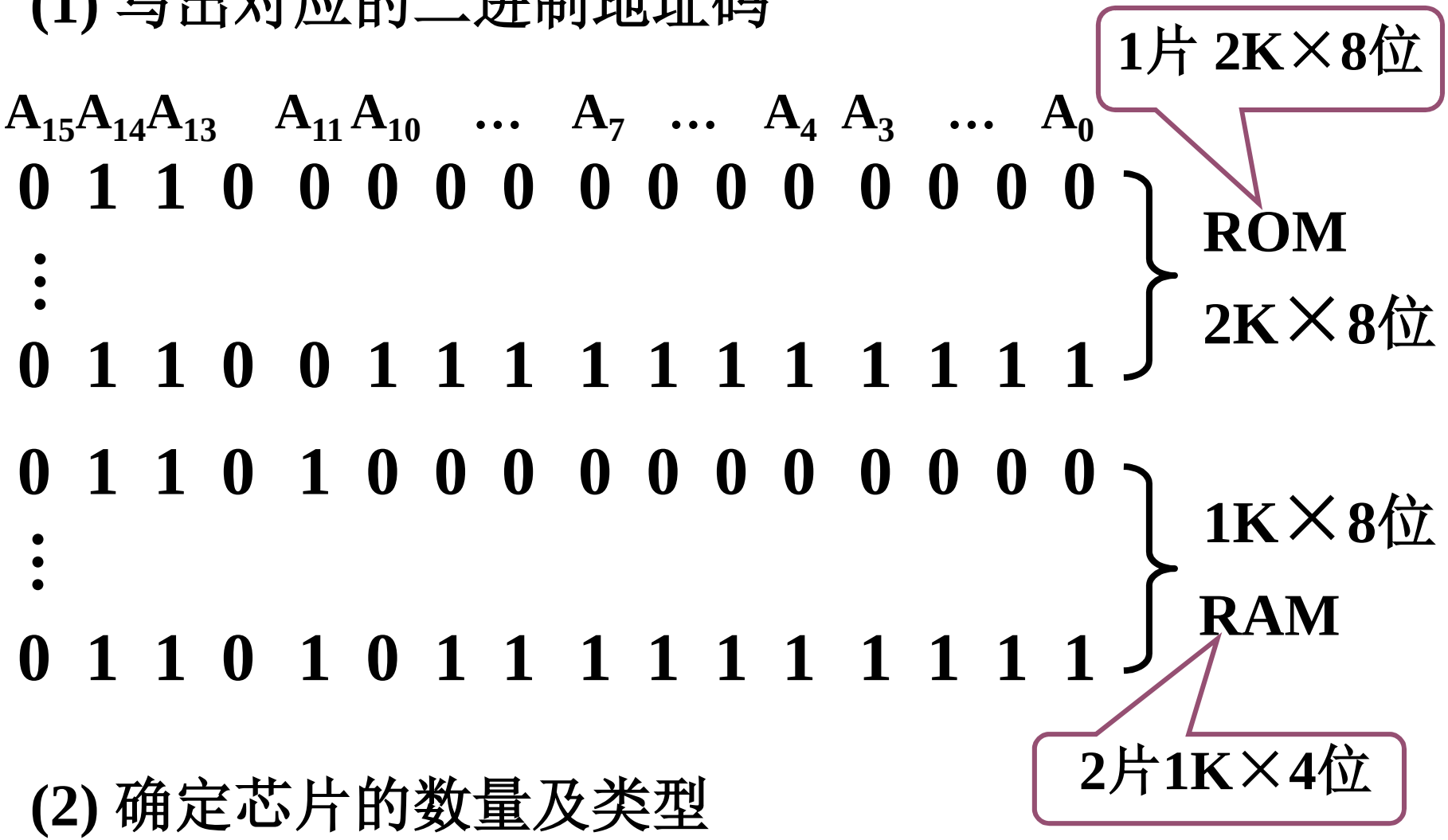
(5) 合理选择存储芯片

(6) 其他　　时序、负载

# 例4.1 解：



(1) 写出对应的二进制地址码




(2) 确定芯片的数量及类型

## (3) 分配地址线

| $A_{15}$ | $A_{13}$ | $A_{11}$ | $A_{10}$ | … | $A_7$ | … | $A_4$ | $A_3$ | … | $A_0$ | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 1 | 1 | 0 | 0 | 0 | 0 0 | 0 | 0 | 0 | 0 | 0 | 1片 ROM 2K×8位 |
| ⋮ | | | | | | | | | | | |
| 0 1 | 1 | 0 | 0 | 1 | 1 1 | 1 | 1 | 1 | 1 | 1 | |
| 0 1 | 1 | 0 | 1 | 0 | 0 0 | 0 | 0 | 0 | 0 | 0 | 2片RAM 1K×4位 |
| ⋮ | | | | | | | | | | | |
| 0 1 | 1 | 0 | 1 | 0 | 1 1 | 1 | 1 | 1 | 1 | 1 | |

C B A

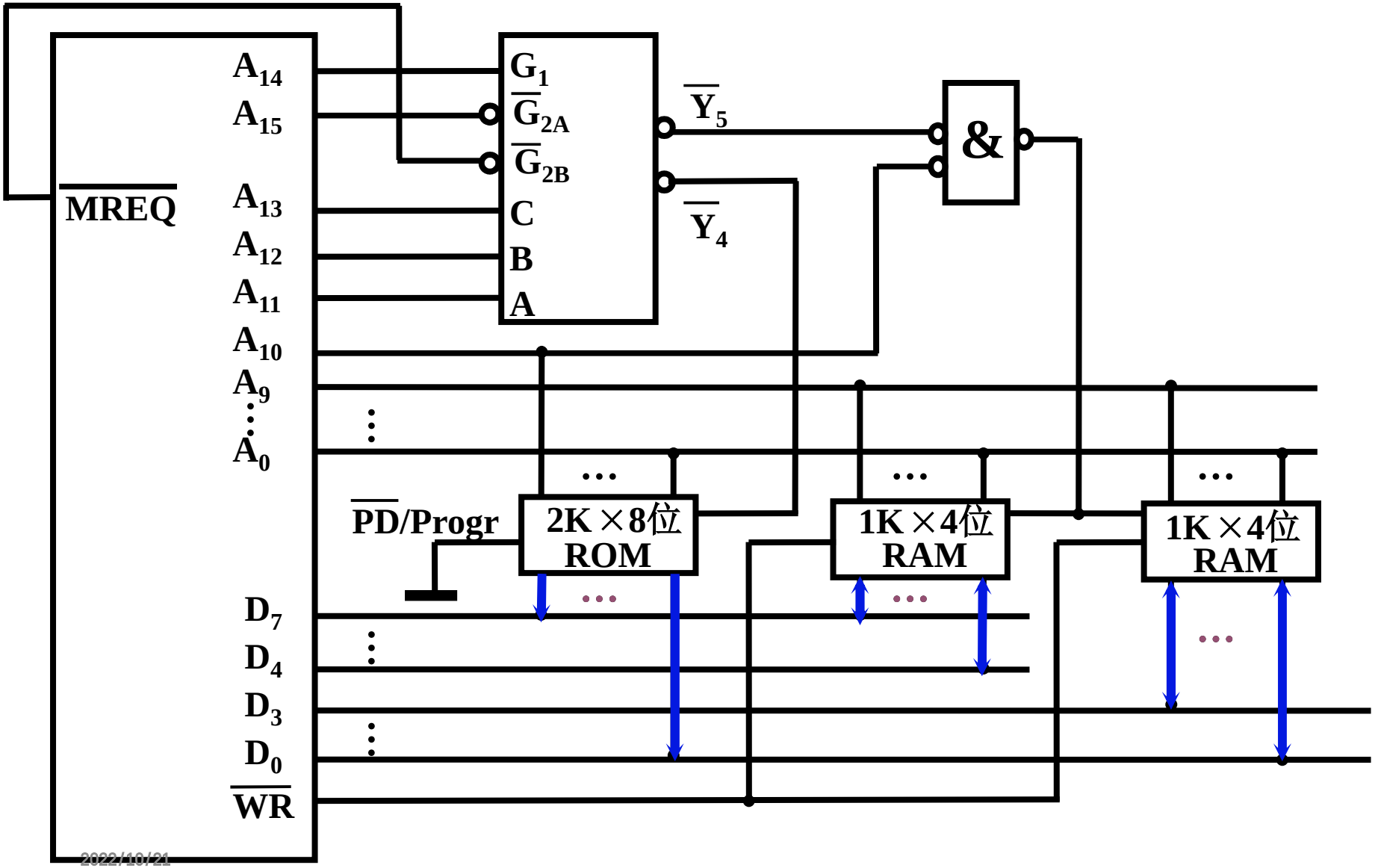
$A_{10}$~$A_0$ 接 2K×8位 ROM 的地址线

$A_9$ ~ $A_0$ 接 1K×4位 RAM 的地址线

## (4) 确定片选信号

# 例 4.1 CPU 与存储器的连接图





**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

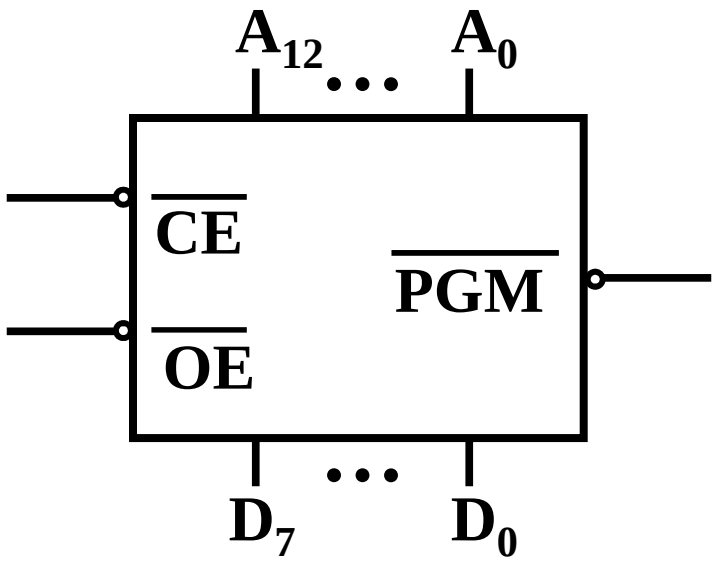
1片 4K×8位 ROM　2片 4K×8位 RAM

(3) 分配地址线

$A_{11}$~$A_0$ 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。并用 IO/$\overline{M}$ 作访存控制信号。$\overline{RD}$ 为读命令，$\overline{WR}$ 为写命令。现有 2764 EPROM ( 8K × 8位 ), 外特性如下：




$\overline{CE}$ 片选信号

$\overline{OE}$ 允许输出

$\overline{PGM}$ 可编程端

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~FFFFFH , 并写出每片 2764 的地址范围。

# 六、存储器的校验

4.2

## 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

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

$L$ —— 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

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

检测位的位置？

$$2^i \ (i = 0，\ 1，\ 2，\ 3，\cdots)$$

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：　∵ $n = 4$　　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 | | | |

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

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

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

∴ 0011 的汉明码为 **1000011**

# 3. 汉明码的纠错过程



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

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

$P_2 = 2 \oplus 3 \oplus 6 \oplus 7 = 1$　有错

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

$\therefore \quad P_4P_2P_1 = 110$

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

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

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

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

$\therefore P_4P_2P_1 = 100$ 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

# 4.2 主存储器

- 一、概述
- 二、半导体存储芯片简介
- 三、随机存取存储器 ( RAM )
- 四、只读存储器（ROM）
- 五、存储器与 CPU 的连接
- 六、存储器的校验
- 七、提高访存速度的措施

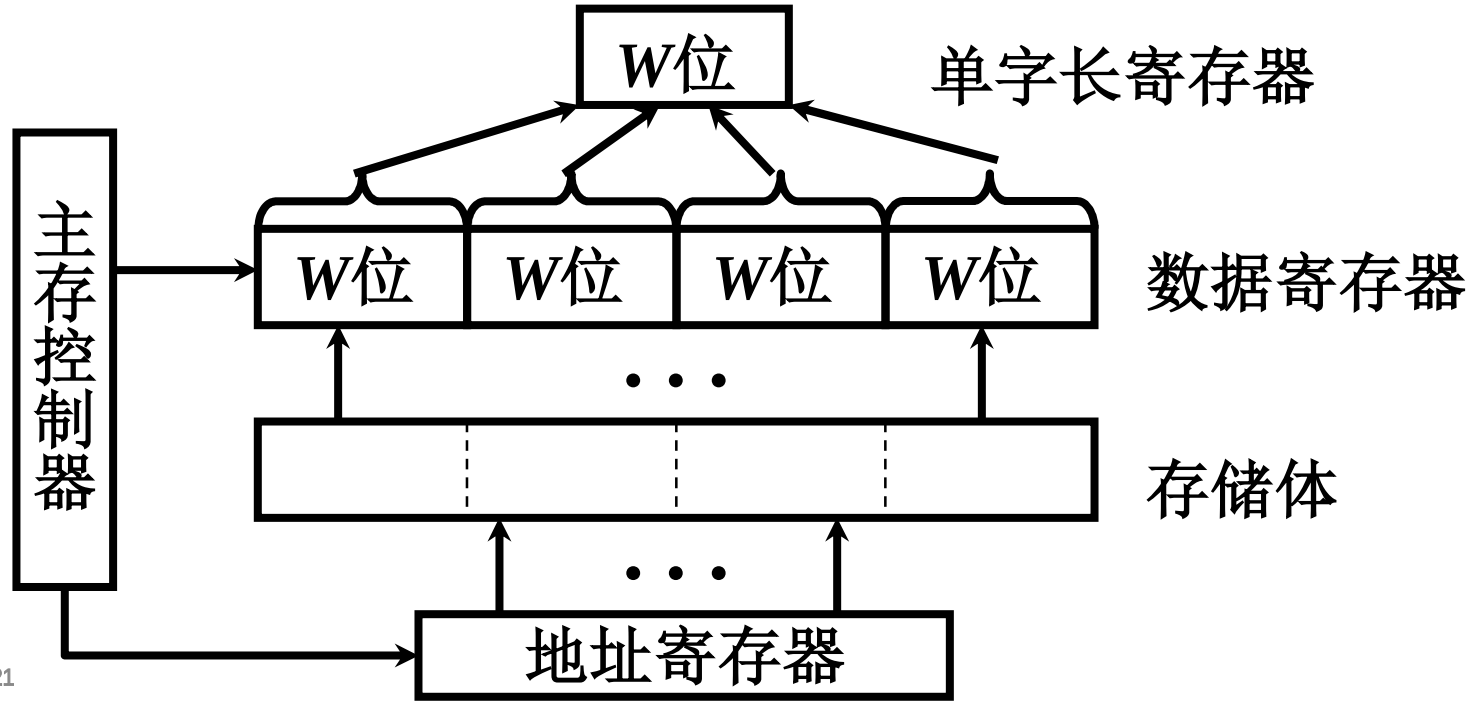
# 七、提高访存速度的措施



- 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

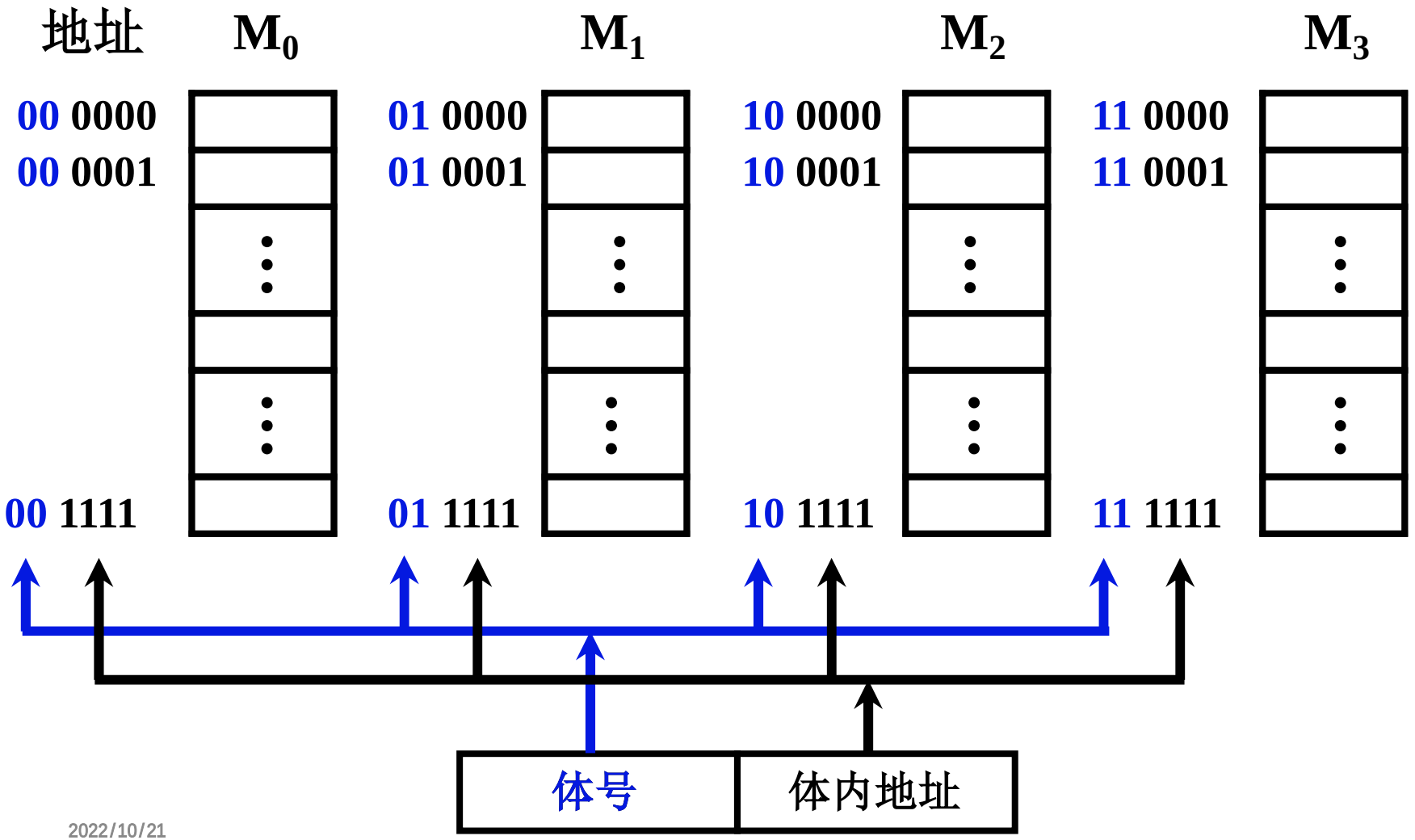
增加存储器的带宽



# 2. 多体并行系统

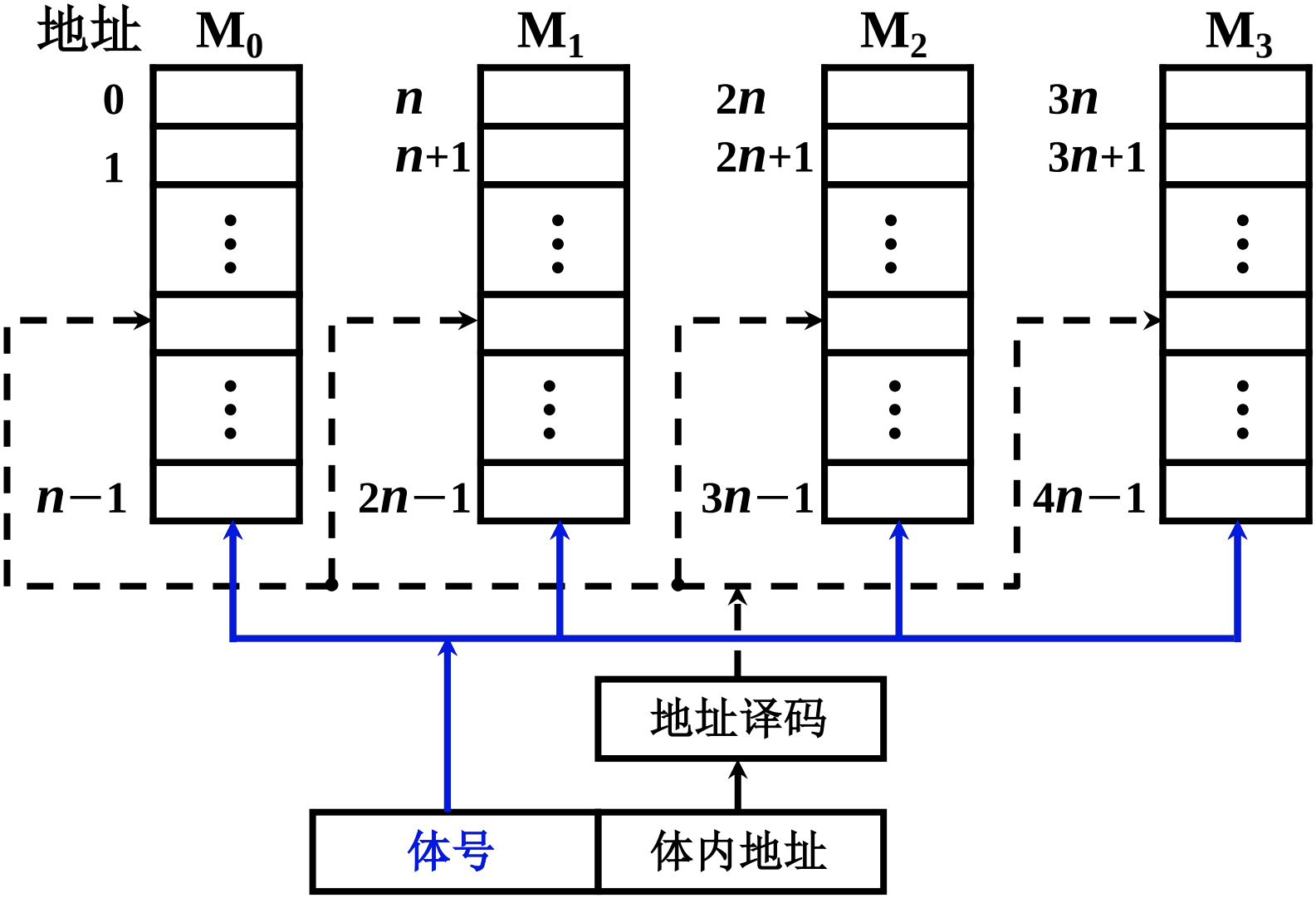


## (1) 高位交叉　　顺序编址



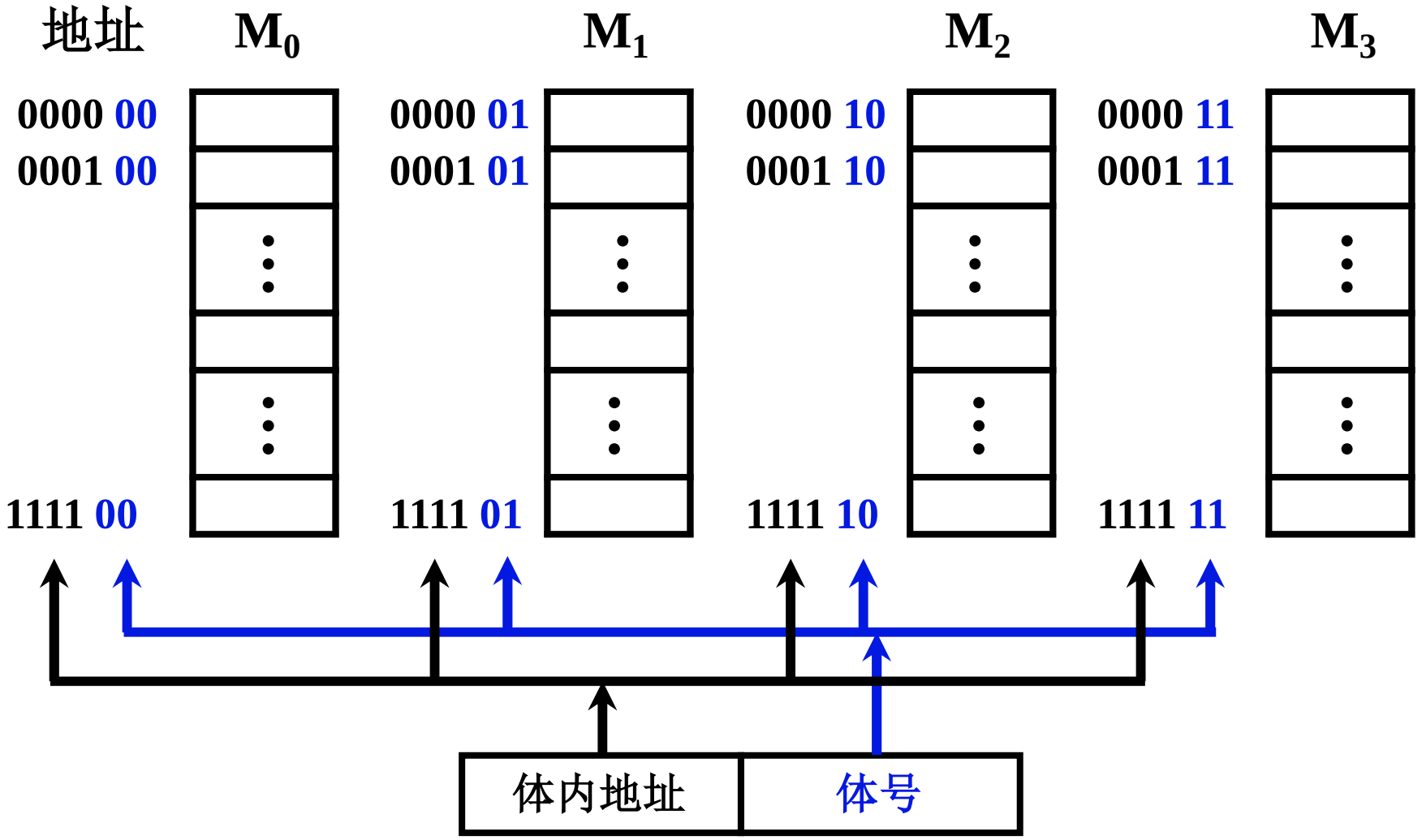
# (1) 高位交叉　各个体并行工作





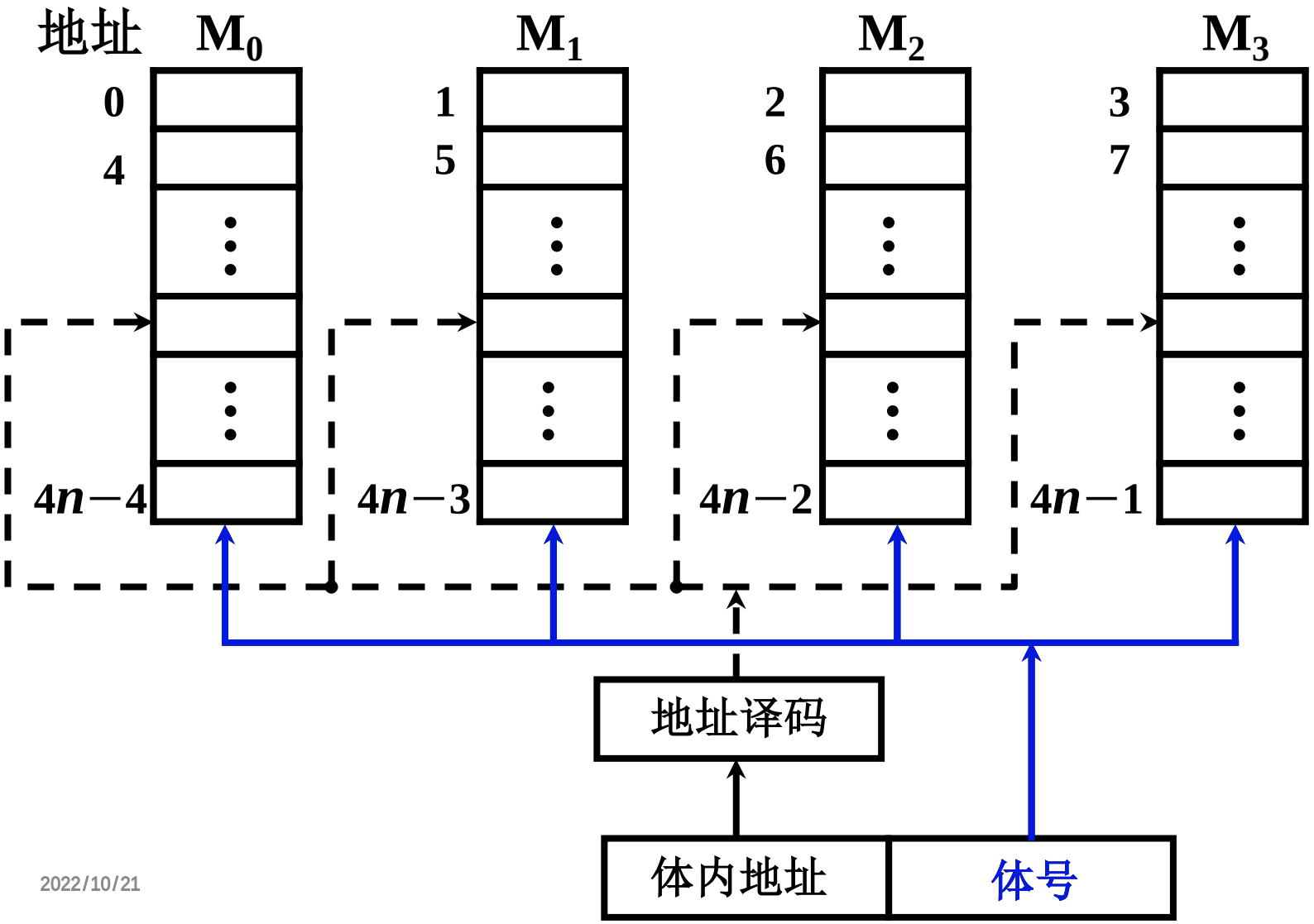
# (2) 低位交叉　各个体轮流编址





# (2) 低位交叉　各个体轮流编址

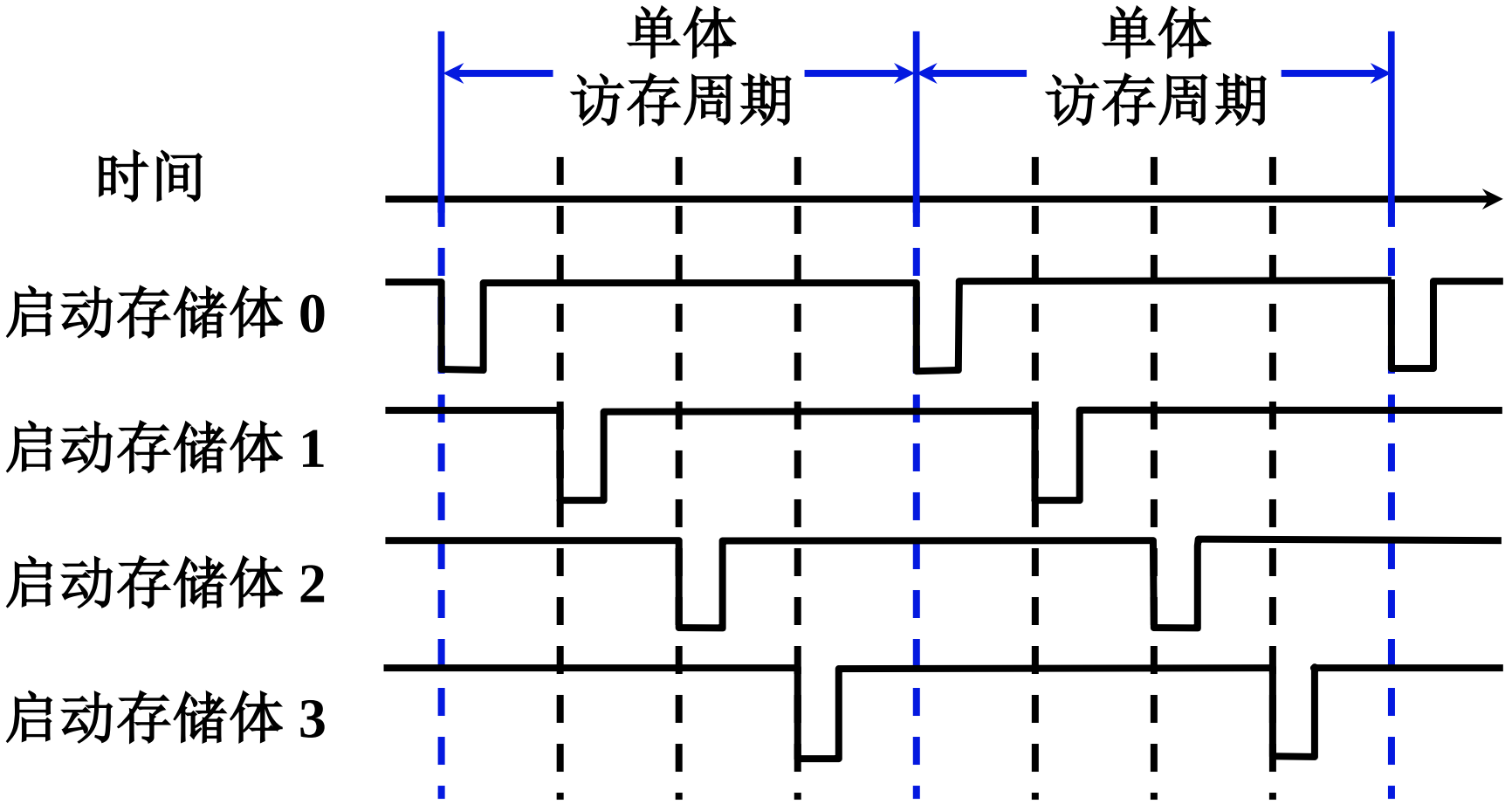




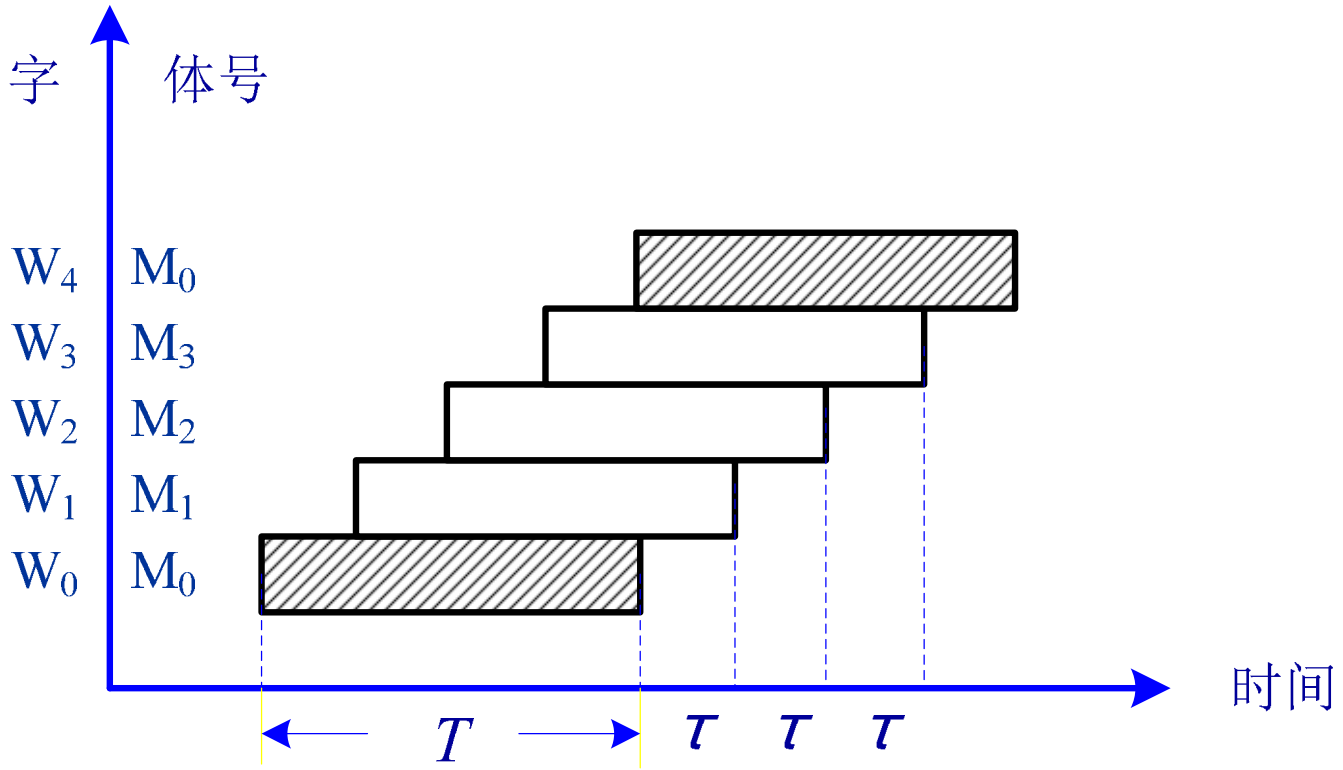
# 低位交叉的特点



在不改变存取周期的前提下，增加存储器的带宽

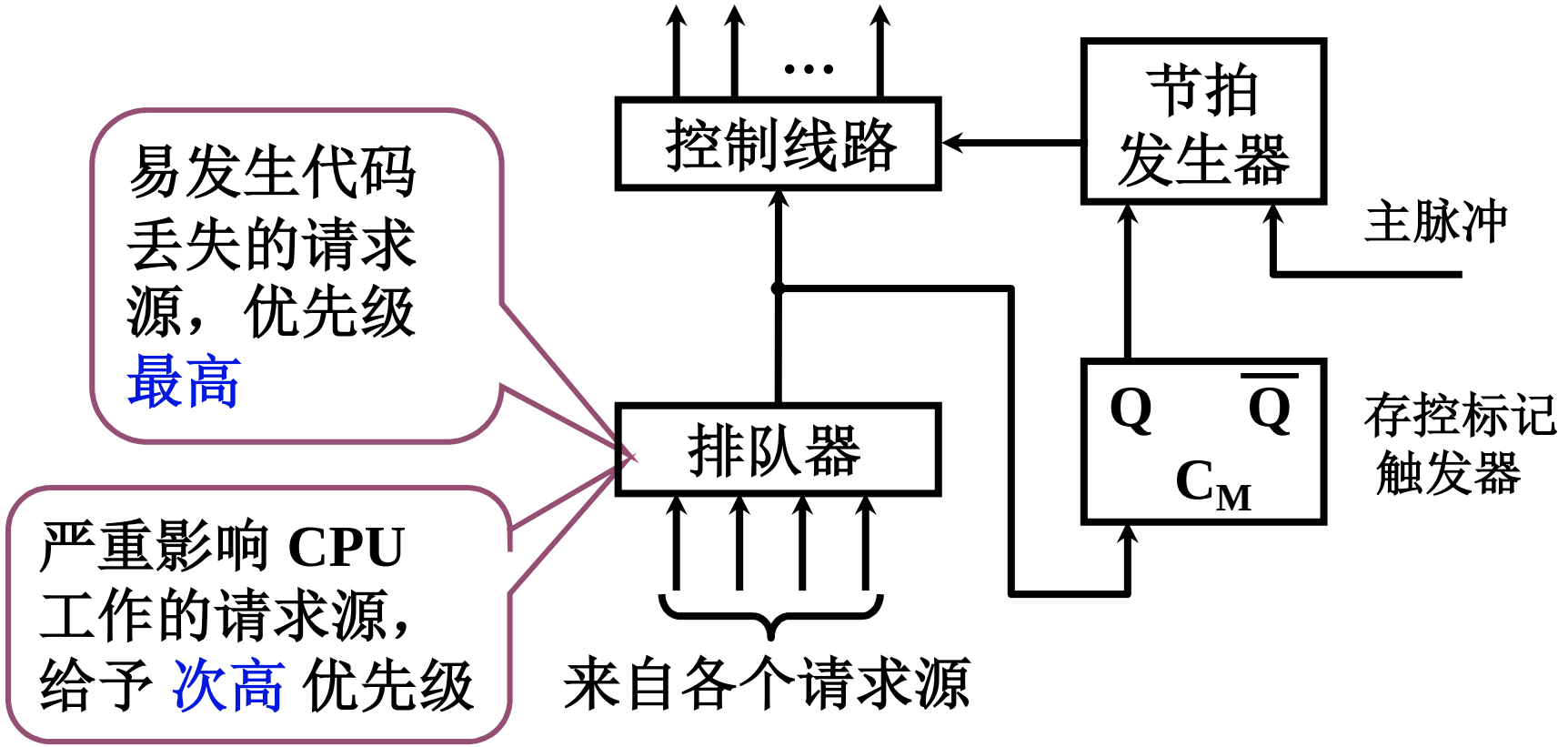


设四体低位交叉存储器，存取周期为$T$，总线传输周期为$\tau$，为实现流水线方式存取，应满足 $T = 4\tau$。




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



## (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

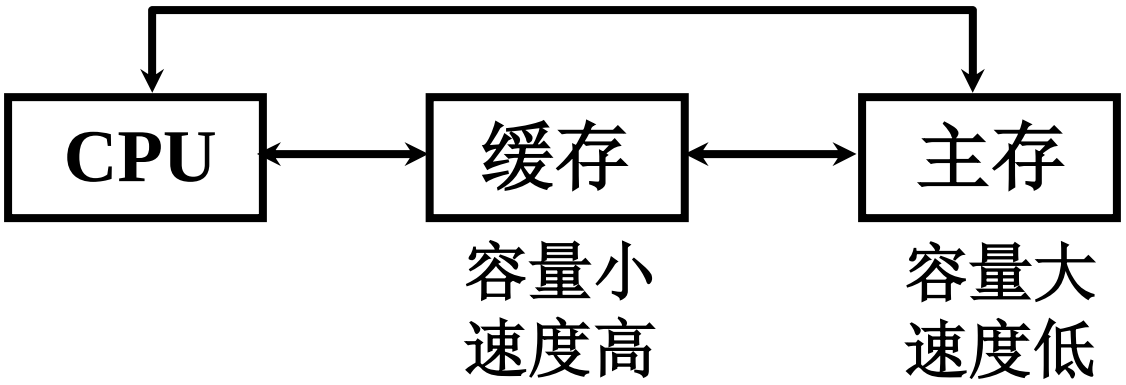
# 4.3 高速缓冲存储器

## 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

# (1) 程序访问的局部性原理



对于绝大多数程序来说，程序所访问的指令和数据在地址上不是均匀分布的，而是相对簇聚的。

程序访问的局部性包含两个方面：

- 时间局部性：程序马上将要用到的信息很可能就是现在正在使用的信息。
- 空间局部性：程序马上将要用到的信息很可能与现在正在使用的信息在存储空间上是相邻的。

# (2) 局部性举例



```
sum = 0;
for (i = 0; i < n; i++)
    sum += a[i];
return sum;
```

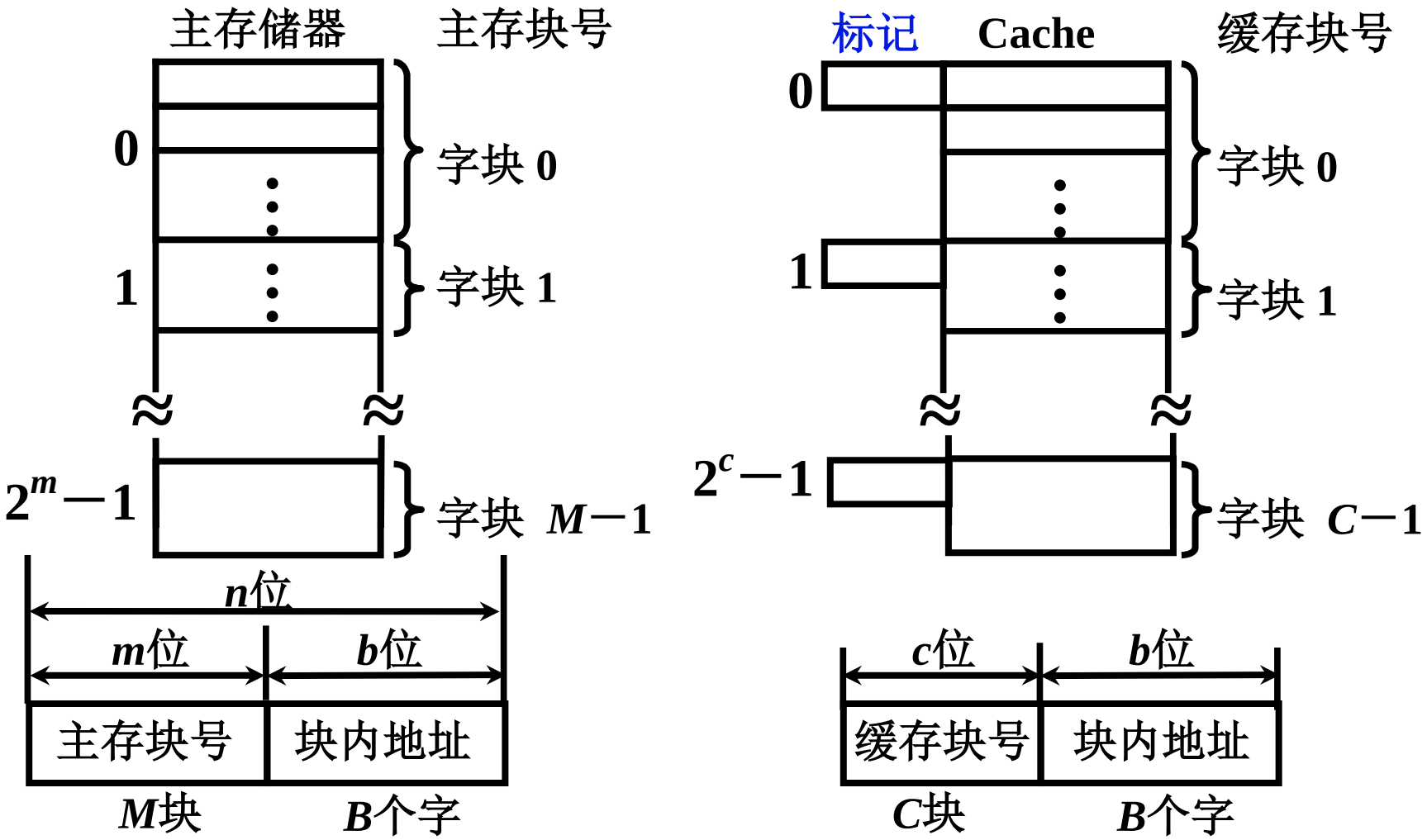
- 对数据的引用
  - 顺序访问数组元素
    (步长为1的引用模式)　　空间局部性
  - 变量sum在每次循环迭代中被引用一次　　时间局部性

- 对指令的引用
  - 顺序读取指令　　空间局部性
  - 重复循环执行for循环体　　时间局部性

# 2. Cache 的工作原理



## (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　$M >> C$

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×4 = 256位）

# (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 $t_m$

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

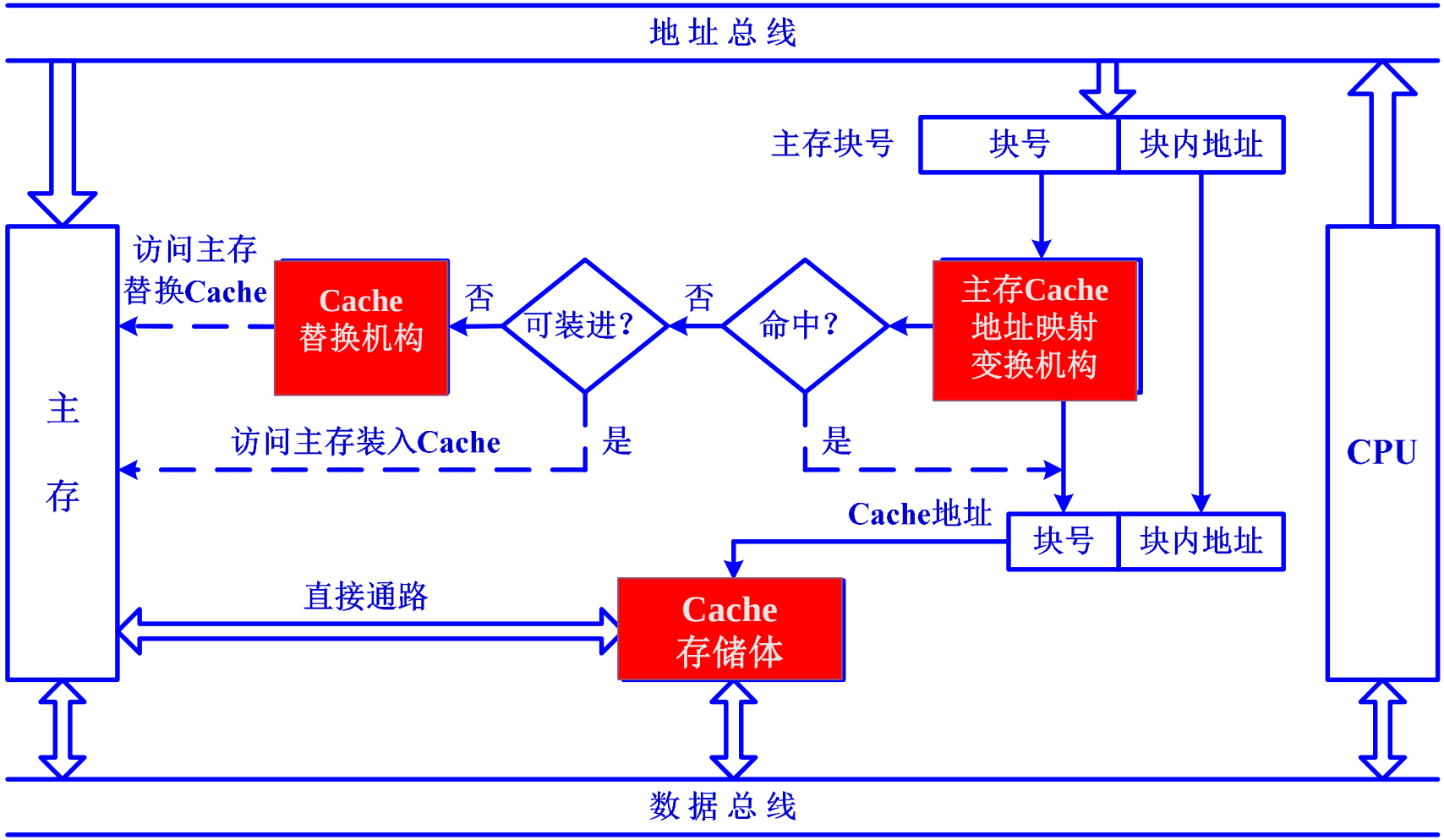
# 存储层次的四个问题



1. 当把一个块调入高一层(靠近CPU)存储器时，可以放在哪些位置上?
   (映象规则　调入块可以放在哪些位置)
2. 当所要访问的块在高一层存储器中时，如何找到该块?
   (查找算法　如何在映象规则 规定的候选位置查找)
3. 当发生失效时，应替换哪一块?
   (替换算法 规定的候选位置均被别的块占用)
4. 当进行写访问时，应进行哪些操作?
   (写策略　如何处理写操作)

# 3. Cache 的基本结构

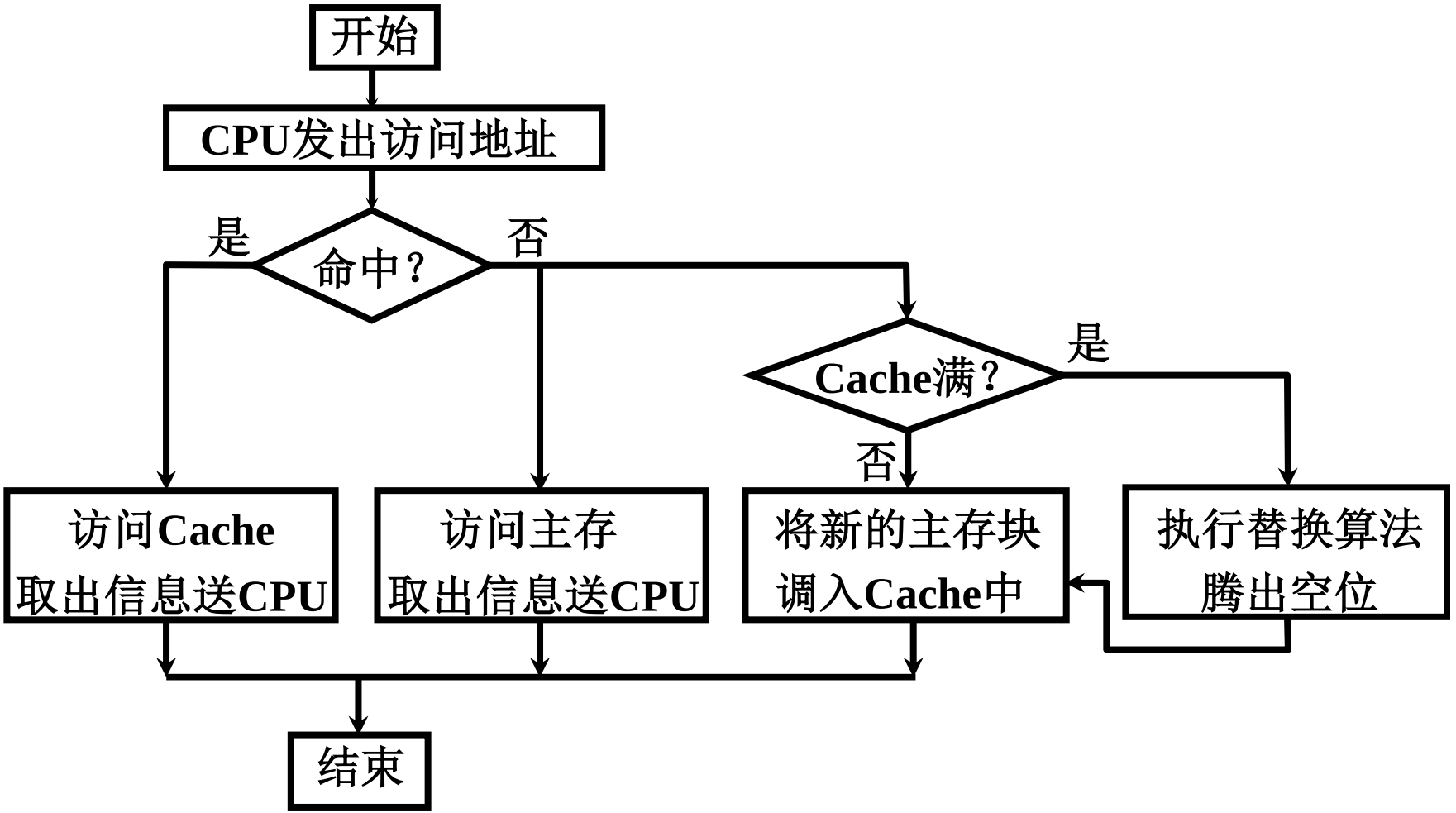




# 4. Cache 的 读写 操作



## 读



# 4. Cache 的 读写 操作



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

# 5. Cache 的改进



## (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

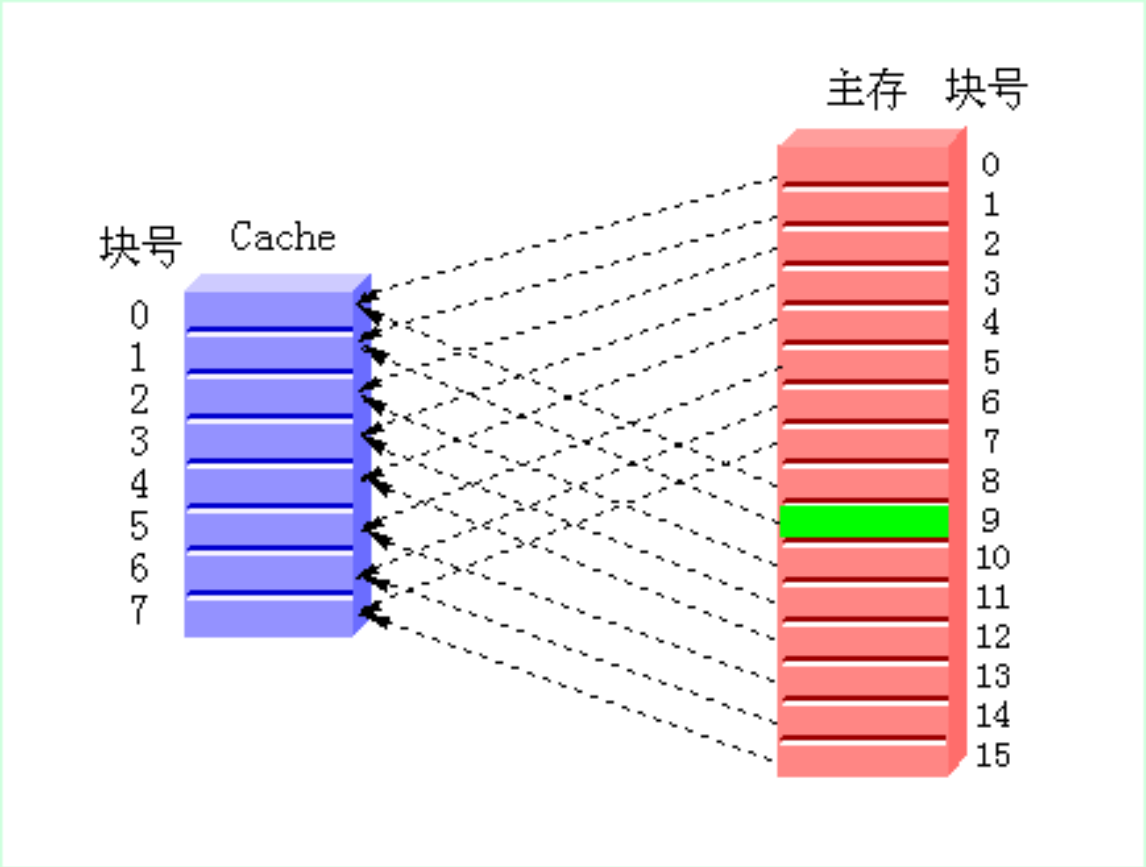
直接映射：主存中的每一块只能被放置到Cache中唯一的一个位置。 (循环分配)

对比：阅览室位置 --只有一个位置可以坐

特点：空间利用率最低，冲突概率最高，实现最简单。

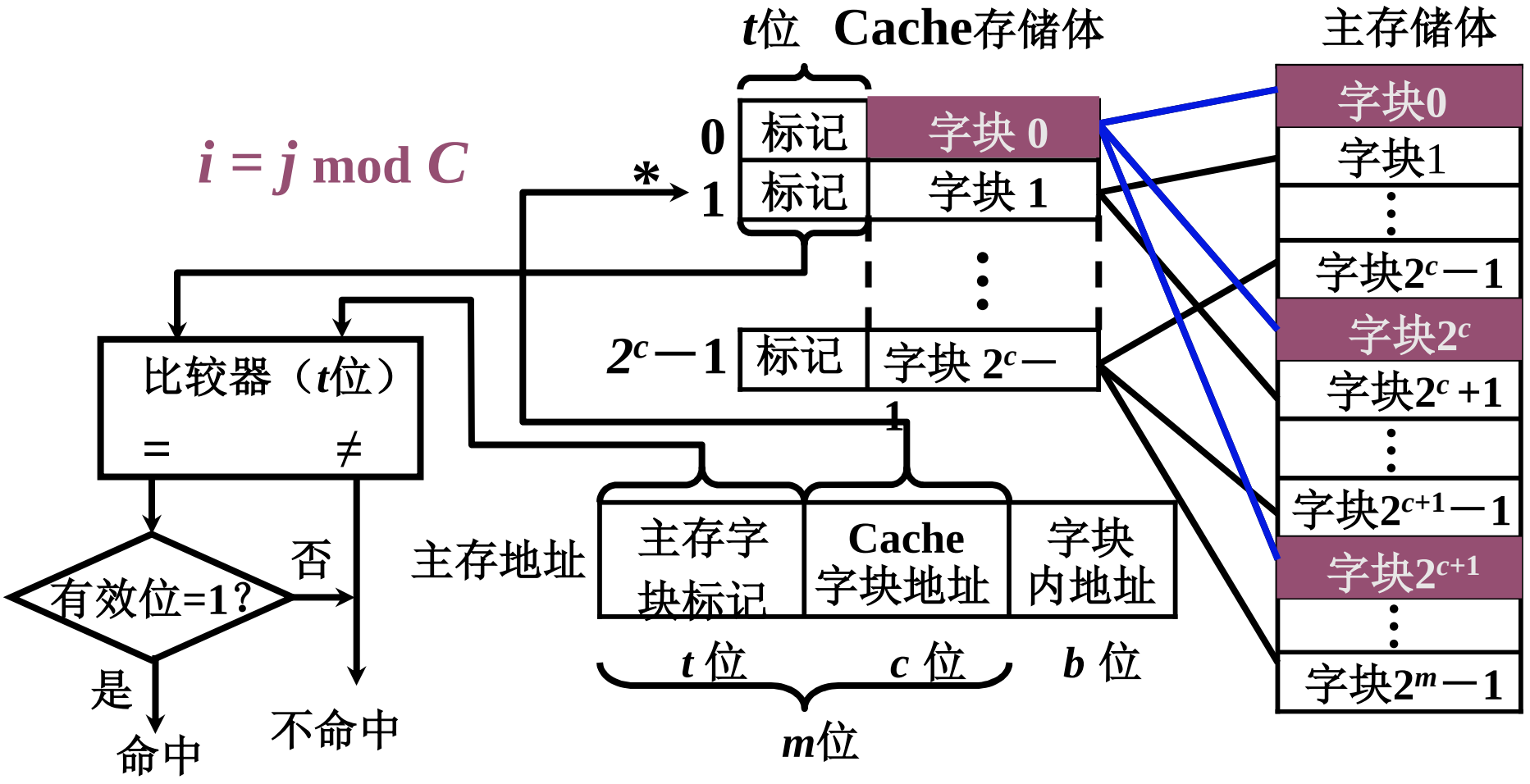
# 直 接 映 射

## （举 例）



# 1. 直接映射






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



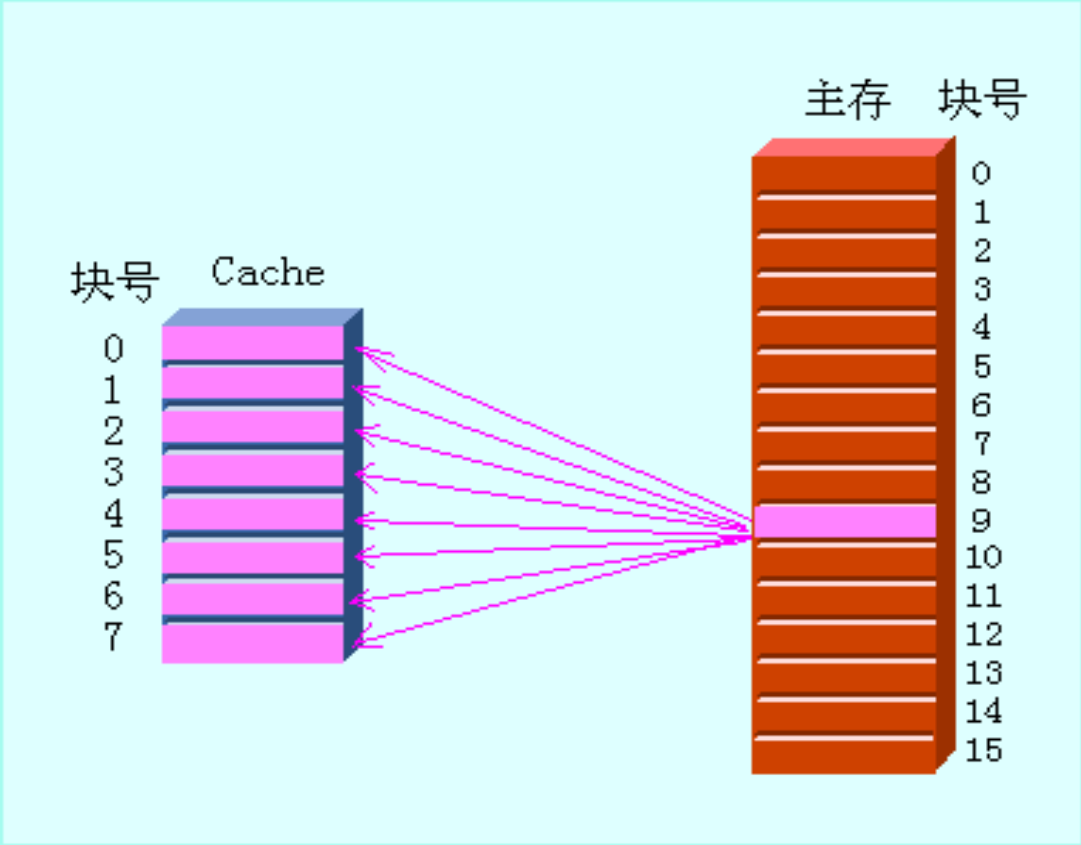
全相联：主存中的任一块可以被放置到Cache中的任意一个位置。

对比：　阅览室位置--随便坐

特点：　空间利用率最高，冲突概率最低，实现最复杂。

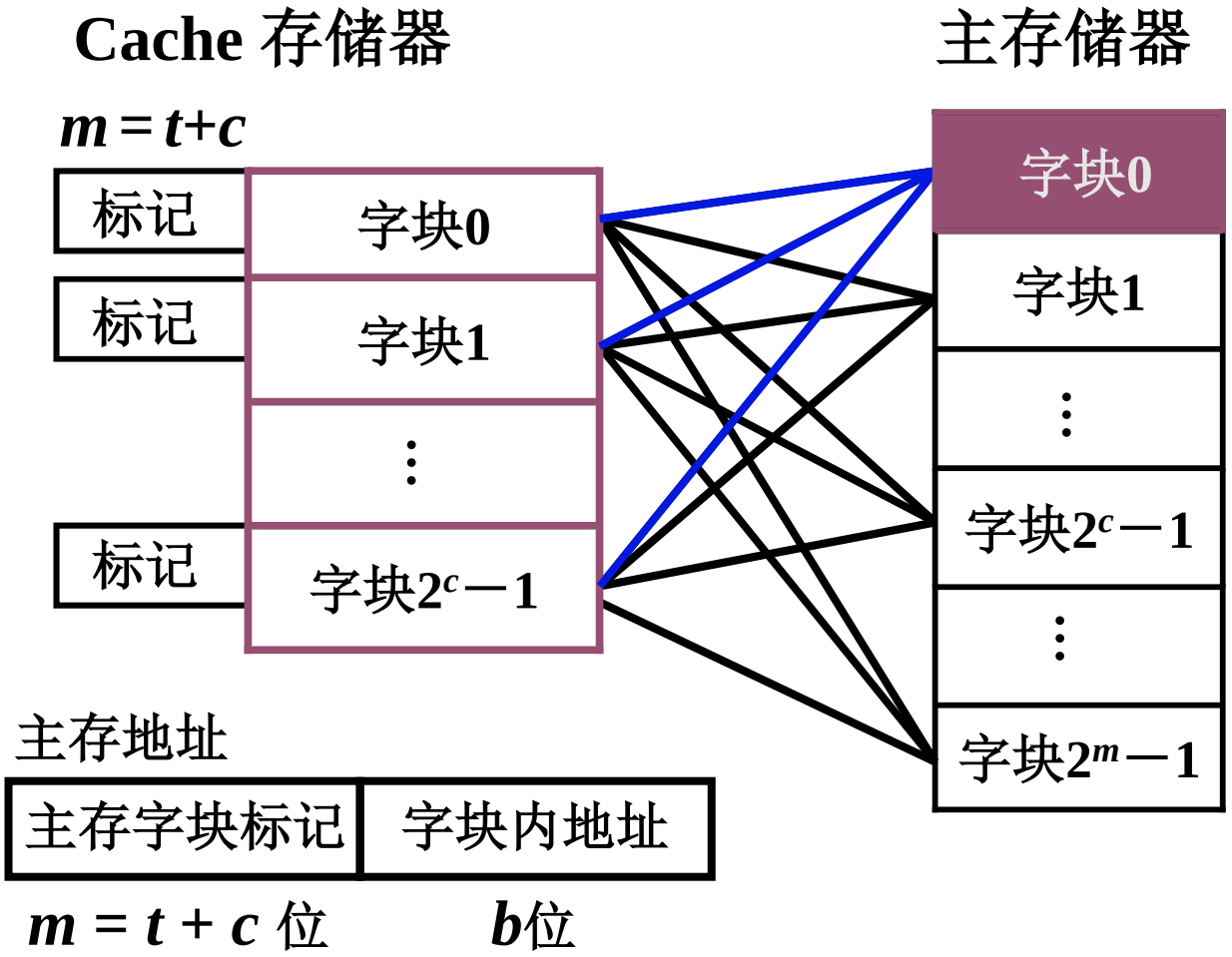
# 全相联映射

## （举例）



# 2. 全相联映射





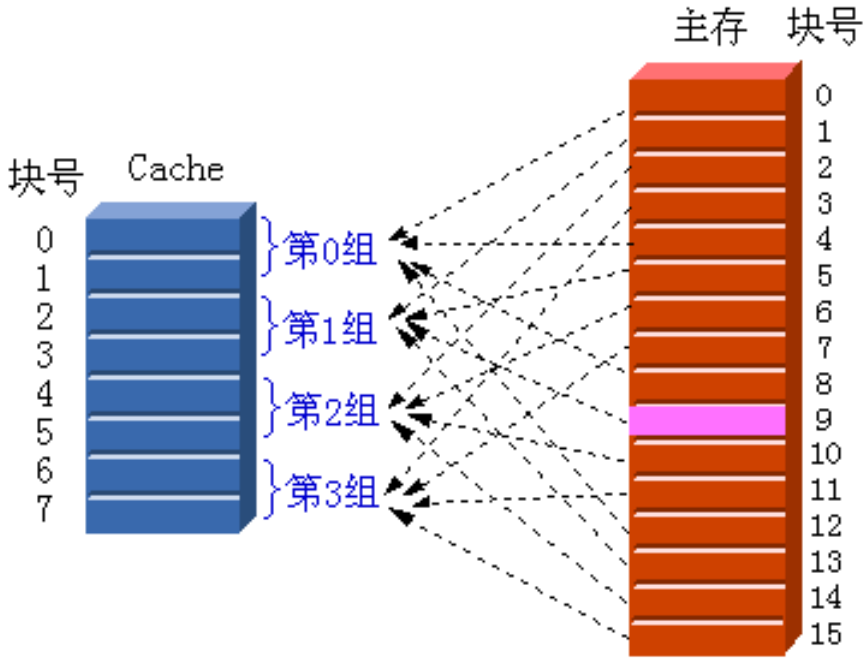

主存 中的 任一块 可以映射到 缓存 中的 任一块

# 3. 组相联映射



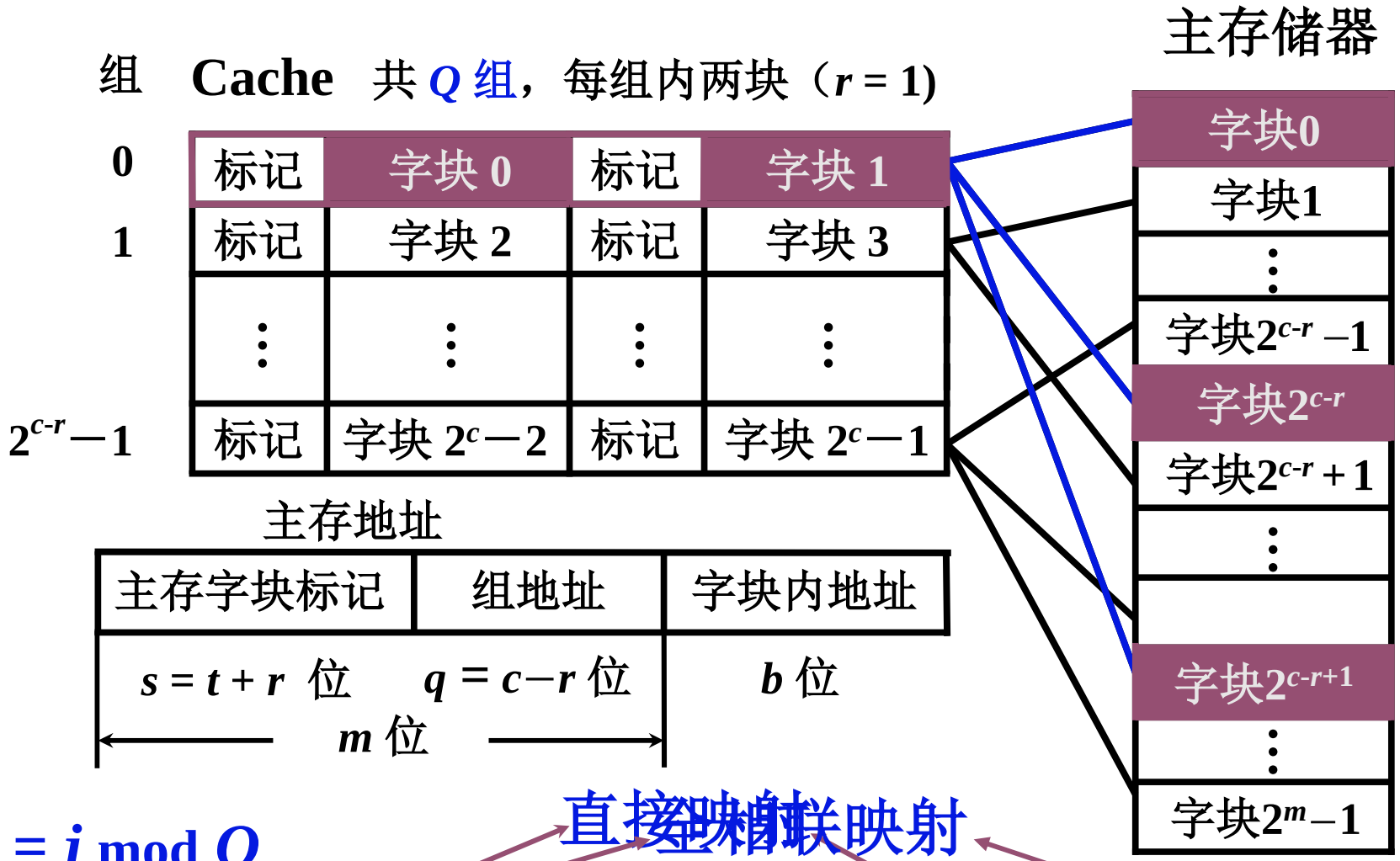
组相联：主存中的每一块可以被放置到Cache中唯一　的一个组中的任何一个位置。

组相联是直接映象和全相联的一种折中



# 3. 组相联映射






$i = j \bmod Q$

直接映射 全相联映射

某一主存块 $j$ 按模 $Q$ 映射到 缓存 的第 $i$ 组中的 任一块

- ***n*** 路组相联：每组中有***n*** 个块(***n*** = ***M/G*** )，***n*** 称为相联度

相联度越高，Cache空间的利用率就越高，块冲突概率就越低，失效率也就越低。

| | ***n*** (路数) | ***G*** (组数) |
|---|---|---|
| 全相联 | ***M*** | 1 |
| 直接映象 | 1 | ***M*** |
| 组相联 | $1<n<M$ | $1<G<M$ |

- 大多数计算机的Cache: $n \leq 4$

想一想：相联度是否越大越好？

# 三、替换算法



- **最近最少使用法LRU**

- 选择近期最少被访问的块作为被替换的块。
  （实现比较困难）
- 实际上：选择最久没有被访问过的块作为被替换的块。
- 优点：命中率较高

- **LRU和随机法分别因其不命中率低和实现简单而被广泛采用。**

- 模拟数据表明，对于容量很大的Cache，LRU和随机法的命中率差别不大。

# 四、写策略 4.3

1. “写”操作所占的比例

Load指令：26%　　Store指令：9%

“写”在所有访存操作中所占的比例：

9%/(100%＋26%＋9%)≈7%

“写”在访问数据Cache操作中所占的比例：

9%/(26%＋9%)≈25%

2. “写”操作必须在确认是否命中后才可进行

3. “写”访问有可能导致Cache和主存内容的不一致

4．两种写策略

- 写直达法：执行“写”操作时，不仅写入Cache，而且也写入下一级存储器。
- 写回法：执行“写”操作时，只写入Cache。仅当Cache中相应的块被替换时，才写回主存。（设置“脏位”）

5．两种写策略的比较

- 写回法的优点：速度快，占用存储器频带低
- 写直达法的优点：易于实现，一致性好

## 6.写缓冲器

## 7. “写”操作时的调块

- 按写分配(写时取)：写失效时，先把所写单元所在的块调入Cache，再行写入。
- 不按写分配(绕写法)：写失效时，直接写入下一级存储器而不调块。

## 8. 写策略与调块

写回法 —— 按写分配

写直达法 —— 不按写分配

# 五、Cache结构举例 4.3

## 例子：DEC的Alpha AXP21064中的内部数据Cache

1．简介

容量：8KB

块大小：32B

块数：256

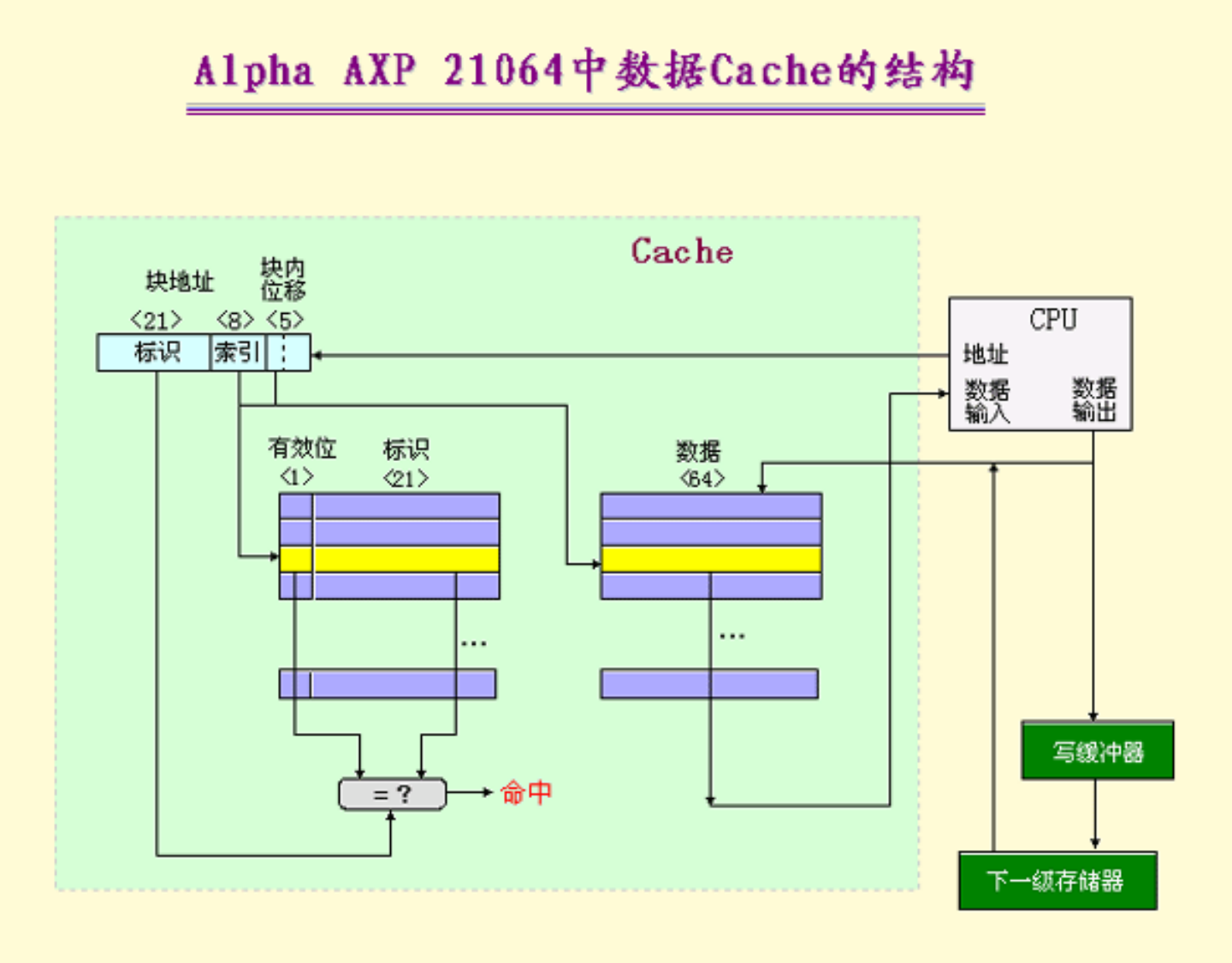
映射方法：直接映射

“写”策略：写直达—不按写分配

写缓冲器大小：4个块

内存地址：34位（块地址29位，块内地址5位）

## 结构图


Alpha AXP 21064中数据Cache的结构
Cache
块地址
块内位移
〈21〉
〈8〉
〈5〉
标识
索引
有效位
〈1〉
标识
〈21〉
数据
〈64〉
CPU
地址
数据输入
数据输出
写缓冲器
下一级存储器
= ?
命中

# 3. 工作过程

① 处理器传送给Cache物理地址

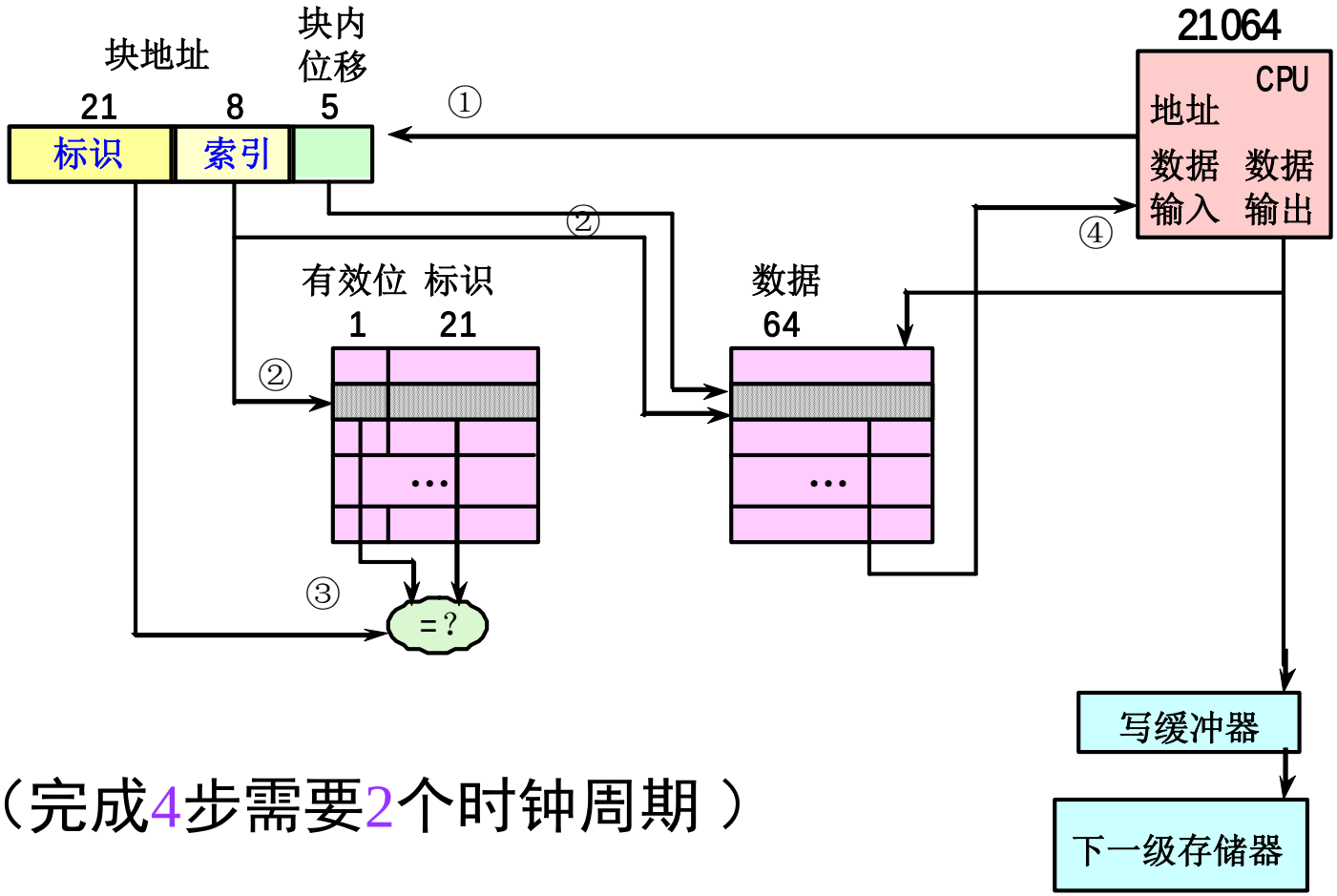
- Cache的容量与索引index、相联度、块大小之间的关系

  Cache的容量=$2^{index}$×相联度×块大小

把容量为8192、相联度为1、块大小为32（字节）代入：

索引index：8位　　标识：29－8＝21位

## 3. 工作过程




（完成4步需要2个时钟周期）

3. 工作过程

① 处理器传送给Cache物理地址

② 由索引选择标识的过程

- 根据索引从目录项中读出相应的标识和有效位

③ 从Cache中读出标识之后，用来同从CPU发来的块地址中标志域部分进行比较

- 为了保证包含有效的信息，必须要设置有效位

④ 如果有一个标识匹配，且标志位有效，则此次命中

- 通知CPU取走数据

- “写”访问命中
  - 前三步一样，只有在确认标识匹配后才把数据写入
- 设置了一个写缓冲器

  （提高“写”访问的速度）
  - 按字寻址的，它含有4个块，每块大小为4个字。
  - 当要进行写入操作时，如果写缓冲器不满，那么就把数据和完整的地址写入缓冲器。对CPU而言，本次“写”访问已完成，CPU可以继续往下执行。由写缓冲器负责把该数据写入主存。
  - 在写入缓冲器时，要进行写合并检查。即检查本次写入数据的地址是否与缓冲器内某个有效块的地址匹配。如果匹配，就把新数据与该块合并 。

# 发生读不命中与写不命中时的操作 4.3

- **读不命中：** 向CPU发出一个暂停信号，通知它等待，并从下一级存储器中新调入一个数据块（32字节）
  - Cache与下一级存储的数据通路宽度为16B，传送一次需5个周期，因此，一次传送需要10个周期
- **写不命中：** 将使数据“绕过”Cache，直接写入主存。
  - 写直达 –不按写分配
- 因为是写直达，所以替换时不需要写回

# 六、改进Cache性能 4.3

平均访存时间＝命中时间＋失效率×失效开销

可以从三个方面改进Cache的性能：

(1) 降低失效率

例如：增加块大小、提高相联度

(2) 减少失效开销

例如：多级Cache、写缓冲、请求字

(3) 减少Cache命中时间

例如：容量小且结构简单的Cache

# 4.4 辅助存储器

## 一、概述

### 1. 特点　不直接与 CPU 交换信息

### 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 { 寻址时间；磁头读写时间

(4) 数据传输率　　$D_r = D_b \times V$

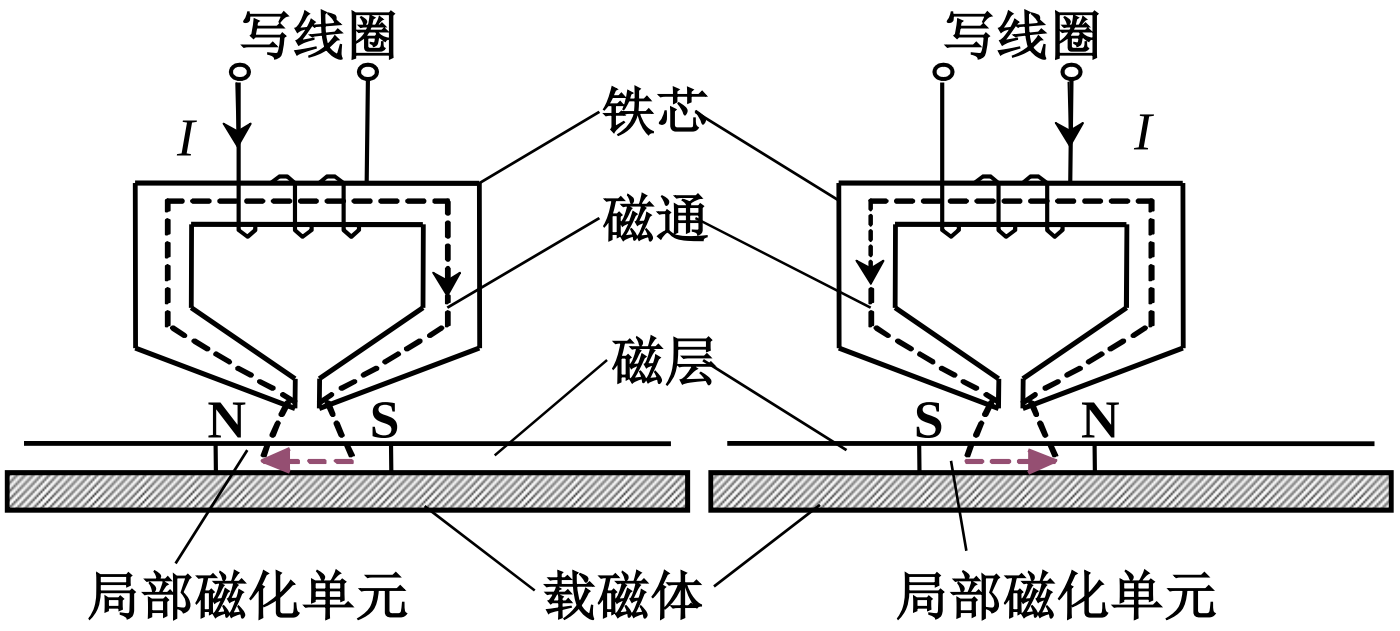
(5) 误码率　　出错信息位数与读出信息的总位数之比

# 二、磁记录原理和记录方式

4.4

## 1. 磁记录原理

### 写



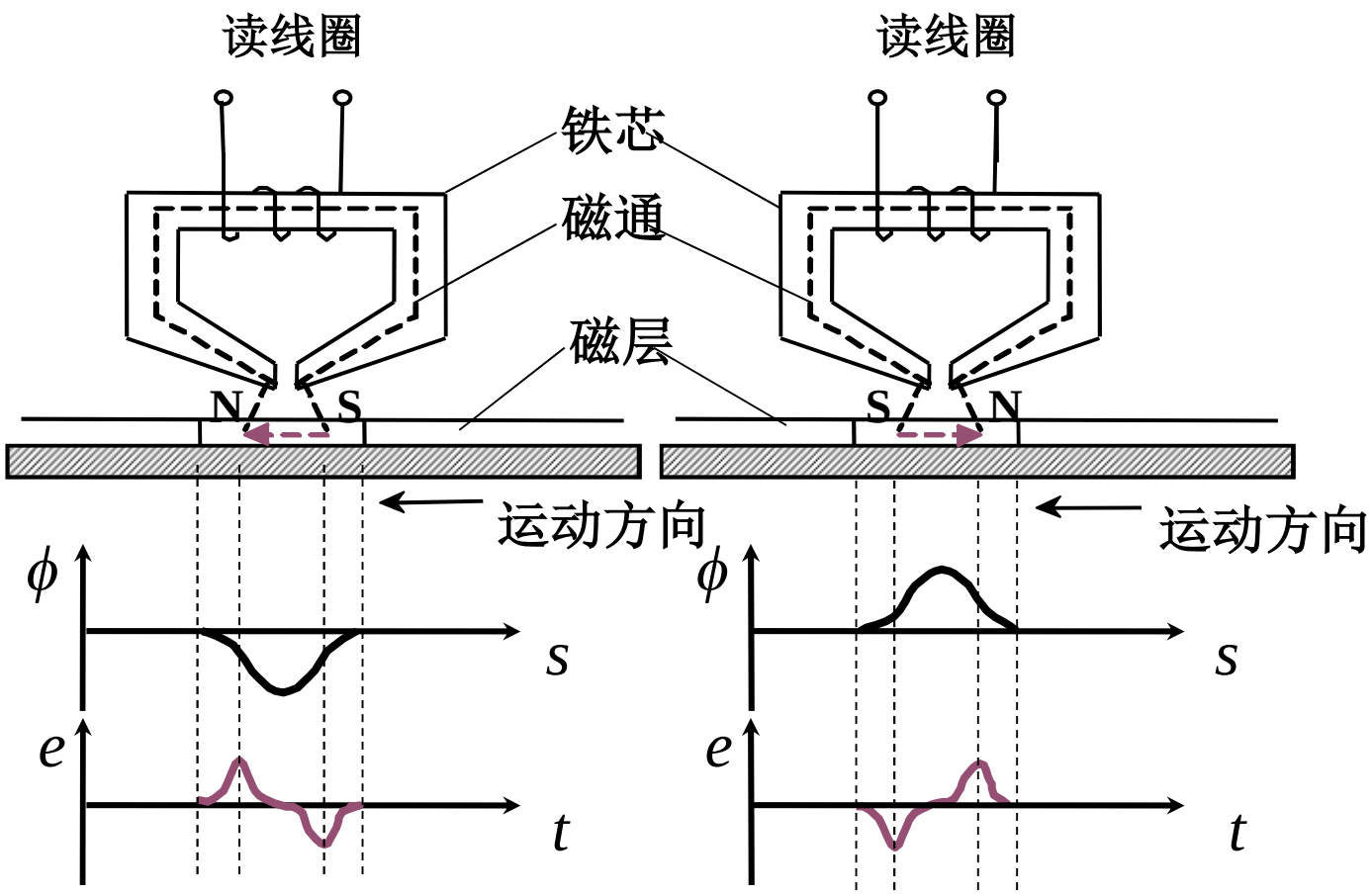

写入“0”

写入“1”

# 1. 磁记录原理



读

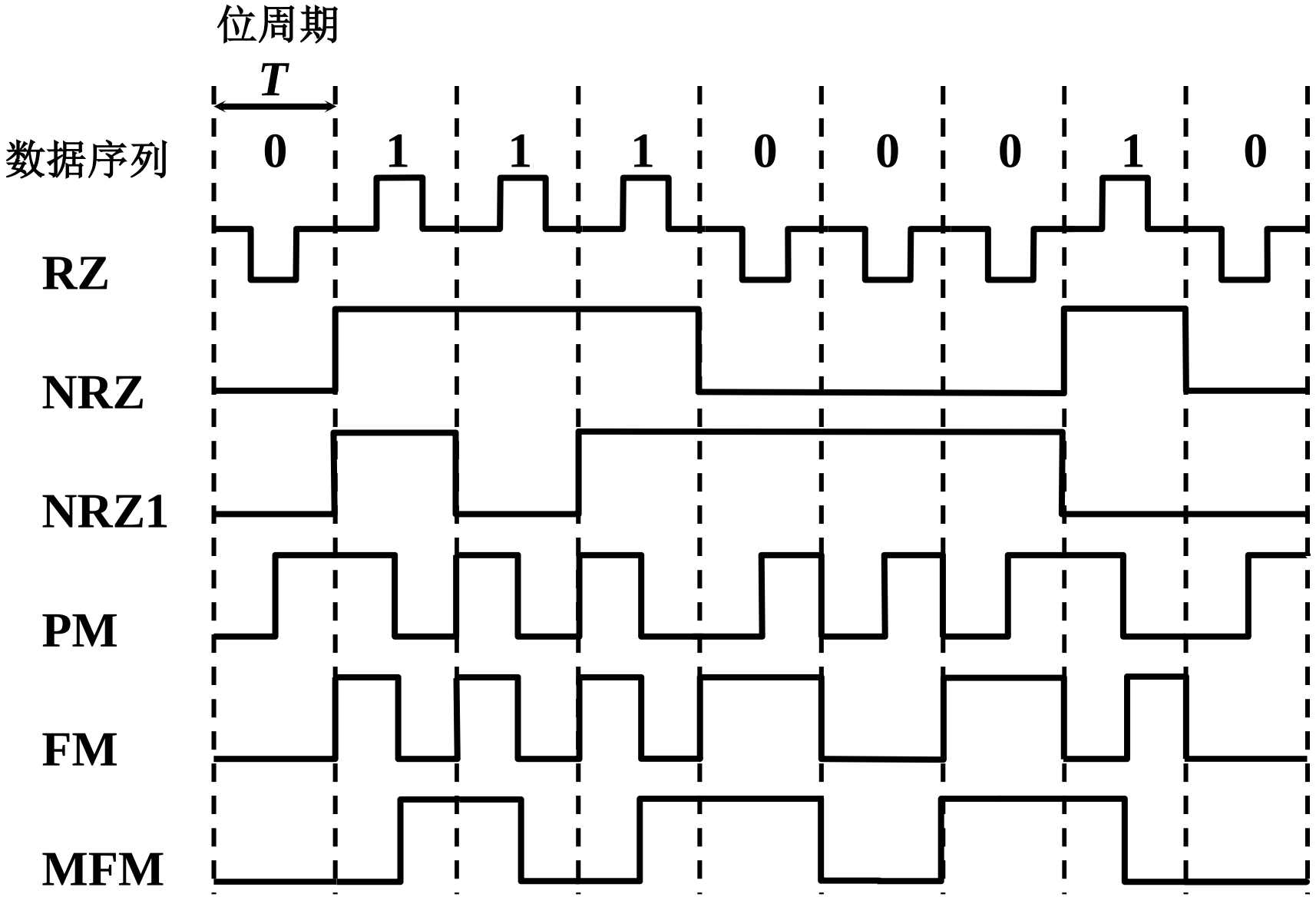



读出 “0”

读出 “1”

# 2. 磁表面存储器的记录方式





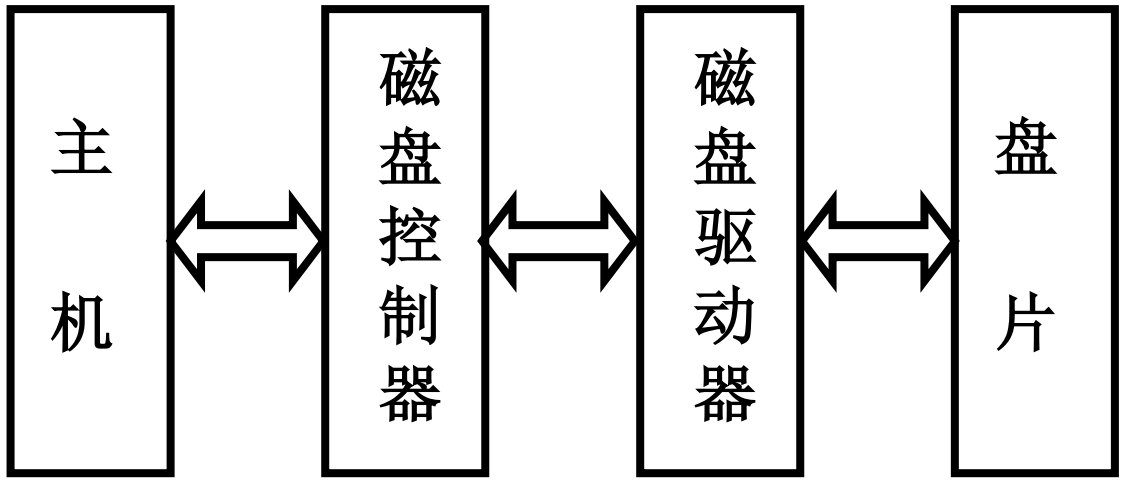
# 三、硬磁盘存储器 4.4

## 1. 硬磁盘存储器的类型

(1) 固定磁头和移动磁头

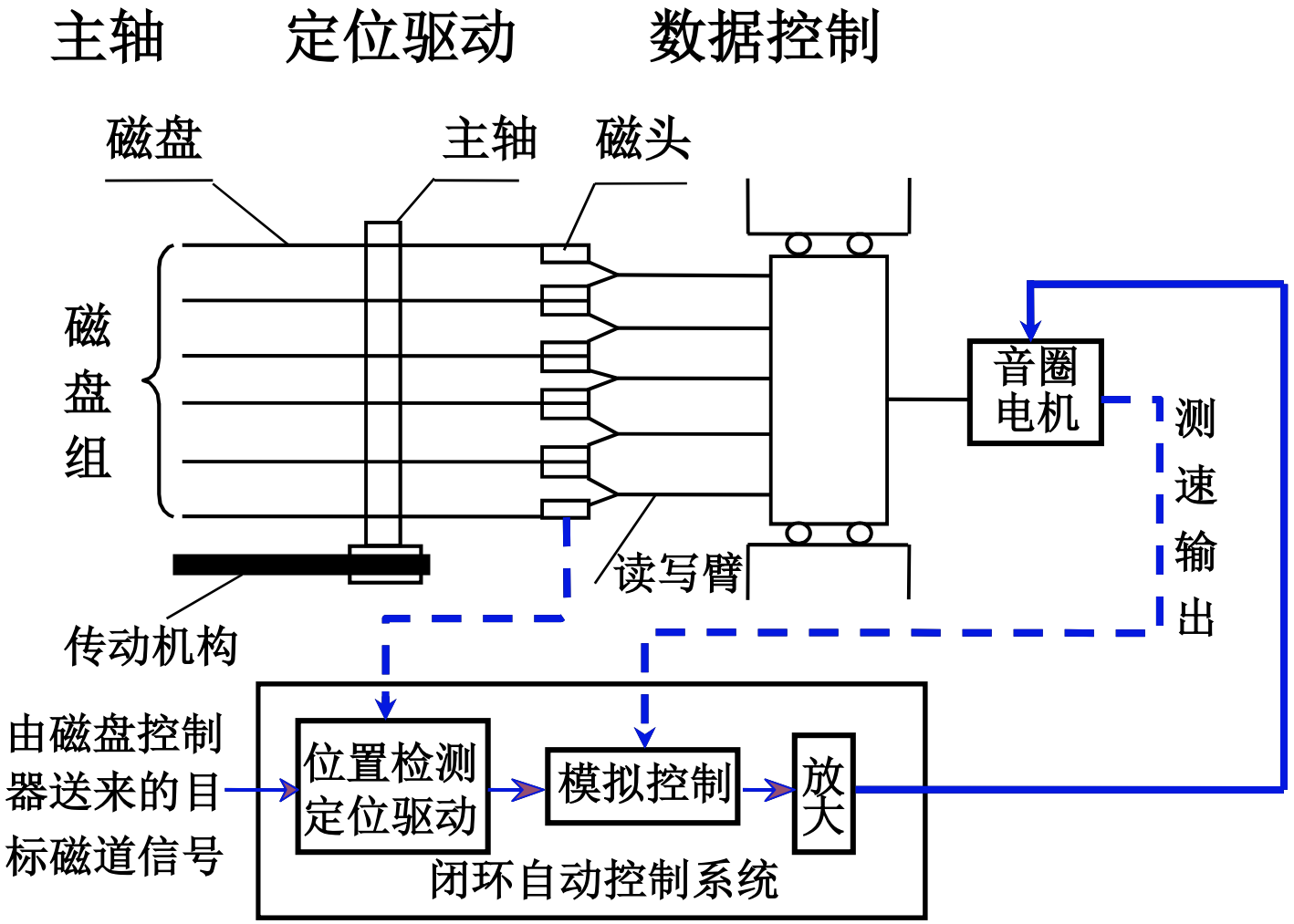
(2) 可换盘和固定盘

## 2. 硬磁盘存储器结构



# (1) 磁盘驱动器




主轴
定位驱动
数据控制
磁盘
主轴
磁头
磁盘组
音圈电机
测速输出
读写臂
传动机构
由磁盘控制器送来的目标磁道信号
位置检测定位驱动
模拟控制
放大
闭环自动控制系统

## (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

# 四、软磁盘存储器

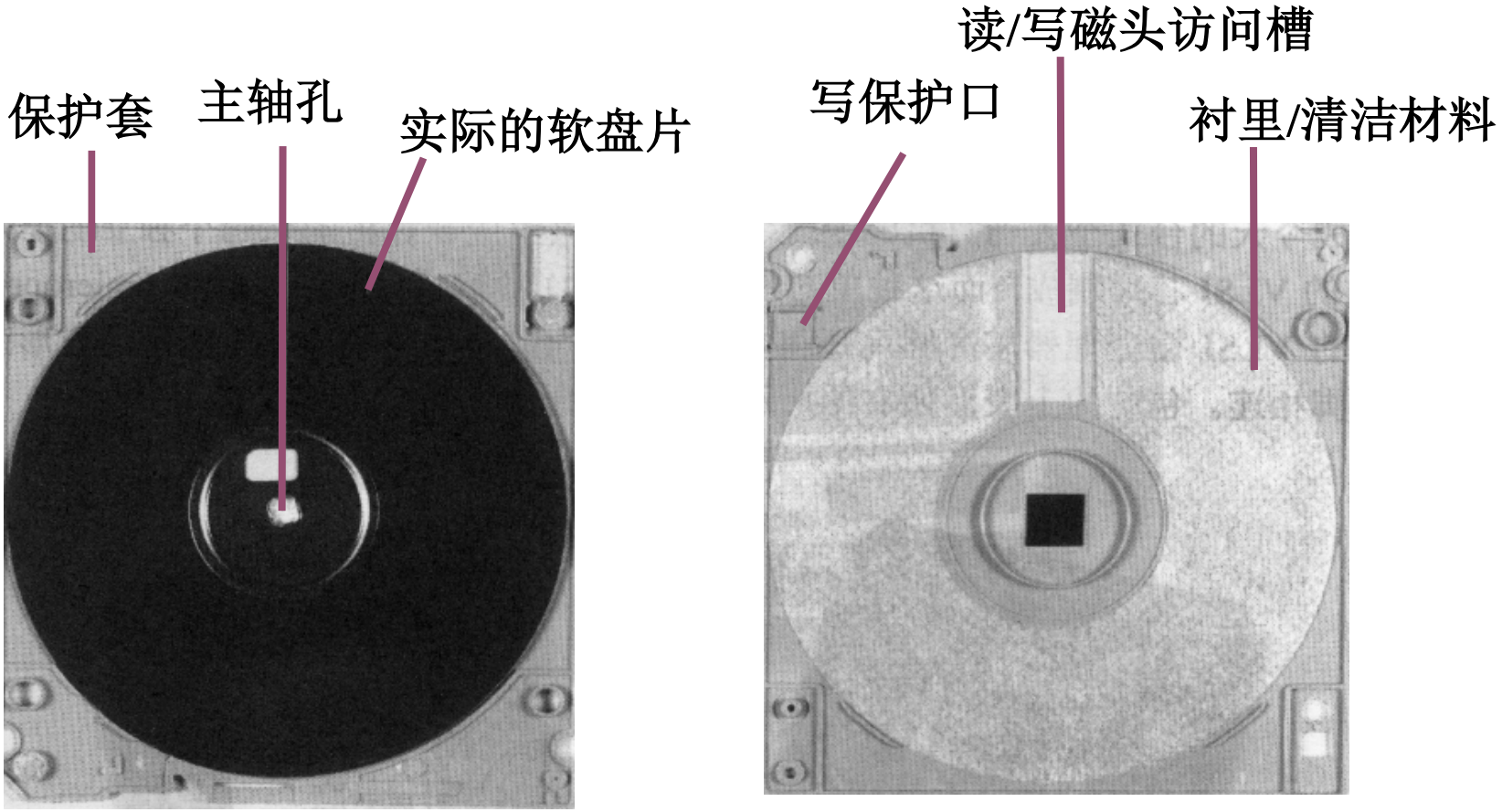


## 1. 概述

| | 硬盘 | 软盘 |
|---|---|---|
| 速度 | 高 | 低 |
| 磁头 | 固定、活动 | 活动 |
| | 浮动 | 接触盘片 |
| 盘片 | 固定盘、盘组<br>大部分不可换 | 可换盘片 |
| 价格 | 高 | 低 |
| 环境 | 苛刻 | |

# 2. 软盘片

## 由聚酯薄膜制成



# 五、光盘存储器



## 1. 概述

采用光存储技术　　利用激光写入和读出

- 第一代光存储技术　　采用非磁性介质　　不可擦写
- 第二代光存储技术　　采用磁性介质　　可擦写

## 2. 光盘的存储原理

只读型和只写一次型　热作用（物理或化学变化）

可擦写光盘　　热磁效应

# 第 5 章　输入输出系统

## 5.1 概述

## 5.2 外部设备

## 5.3 I/O接口

## 5.4 程序查询方式

## 5.5 程序中断方式

## 5.6 DMA方式

# 5.1 概 述

## 一、输入输出系统的发展概况

### 1. 早期

分散连接

CPU 和 I/O设备 串行 工作　程序查询方式

### 2. 接口模块和 DMA 阶段

总线连接

CPU 和 I/O设备 并行 工作 { 中断方式 / DMA 方式

### 3. 具有通道结构的阶段

### 4. 具有 I/O 处理机的阶段

# 二、输入输出系统的组成 5.1

## 1. I/O 软件

### (1) I/O 指令　　CPU 指令的一部分


### (2) 通道指令　　通道自身的指令

指出数组的首地址、传送字数、操作命令

如 IBM/370 通道指令为 64 位

## 2. I/O 硬件

设备　　I/O 接口

设备　　设备控制器　　通道

# 三、I/O 设备与主机的联系方式 5.1

## 1. I/O 设备编址方式

(1) 统一编址　　用取数、存数指令

(2) 不统一编址　　有专门的 I/O 指令

## 2. 设备选址

用设备选择电路识别是否被选中

## 3. 传送方式

(1) 串行

(2) 并行

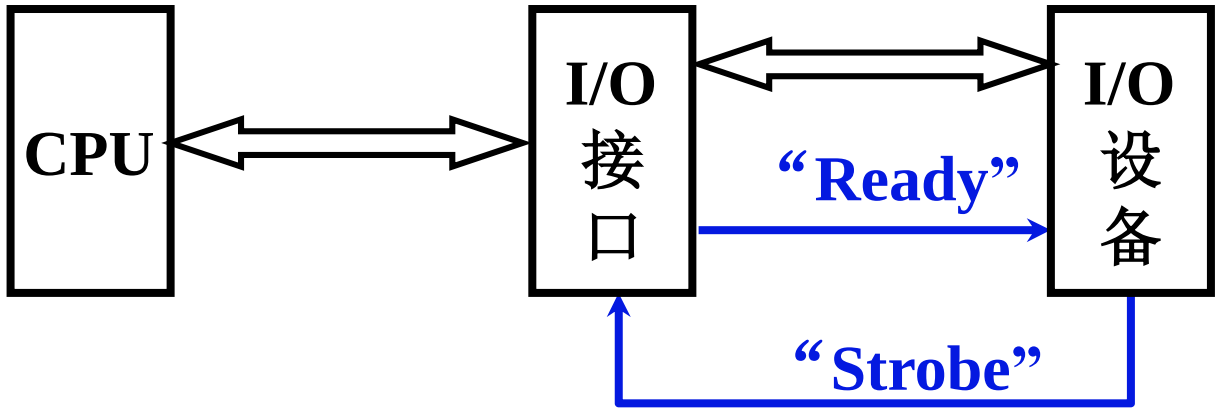
# 4. 联络方式



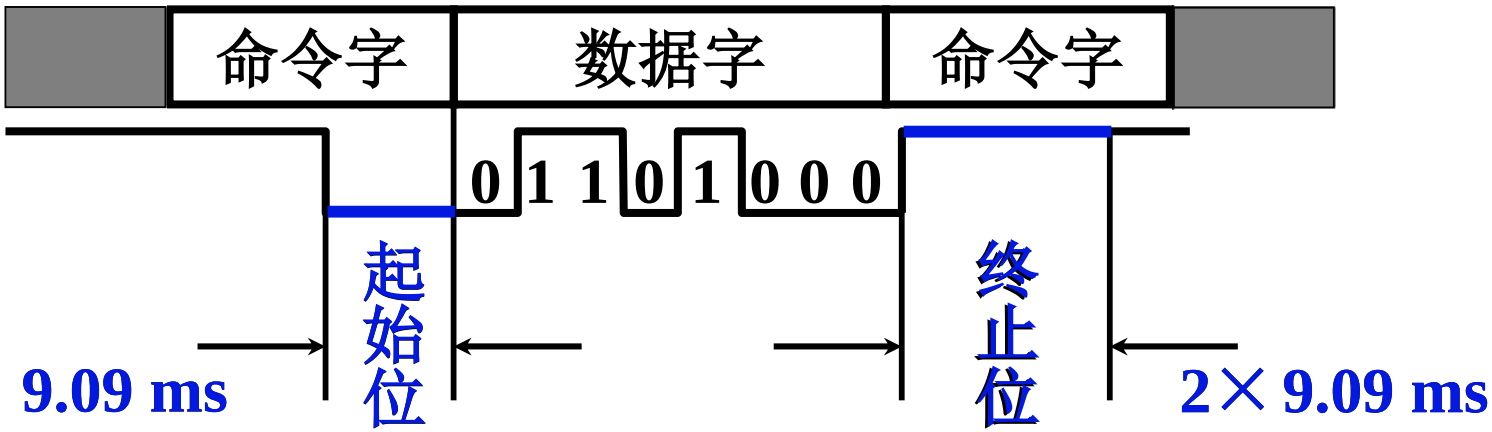
(1) 立即响应

(2) 异步工作采用应答信号

并行




串行

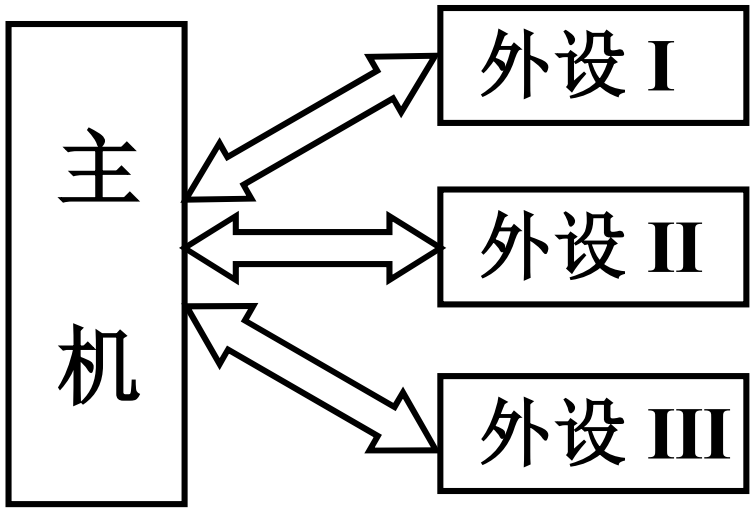



(3) 同步工作采用同步时标

# 5. I/O 设备与主机的连接方式



## (1) 辐射式连接




每台设备都配有一套

控制线路和一组信号线

不便于增删设备

## (2) 总线连接

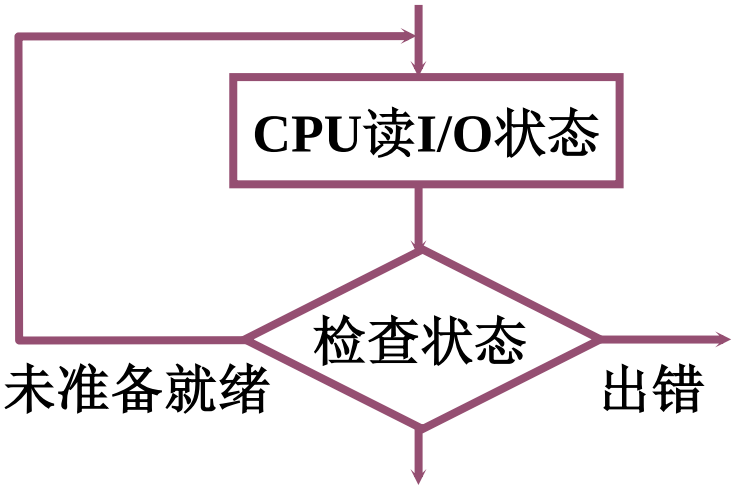
便于增删设备

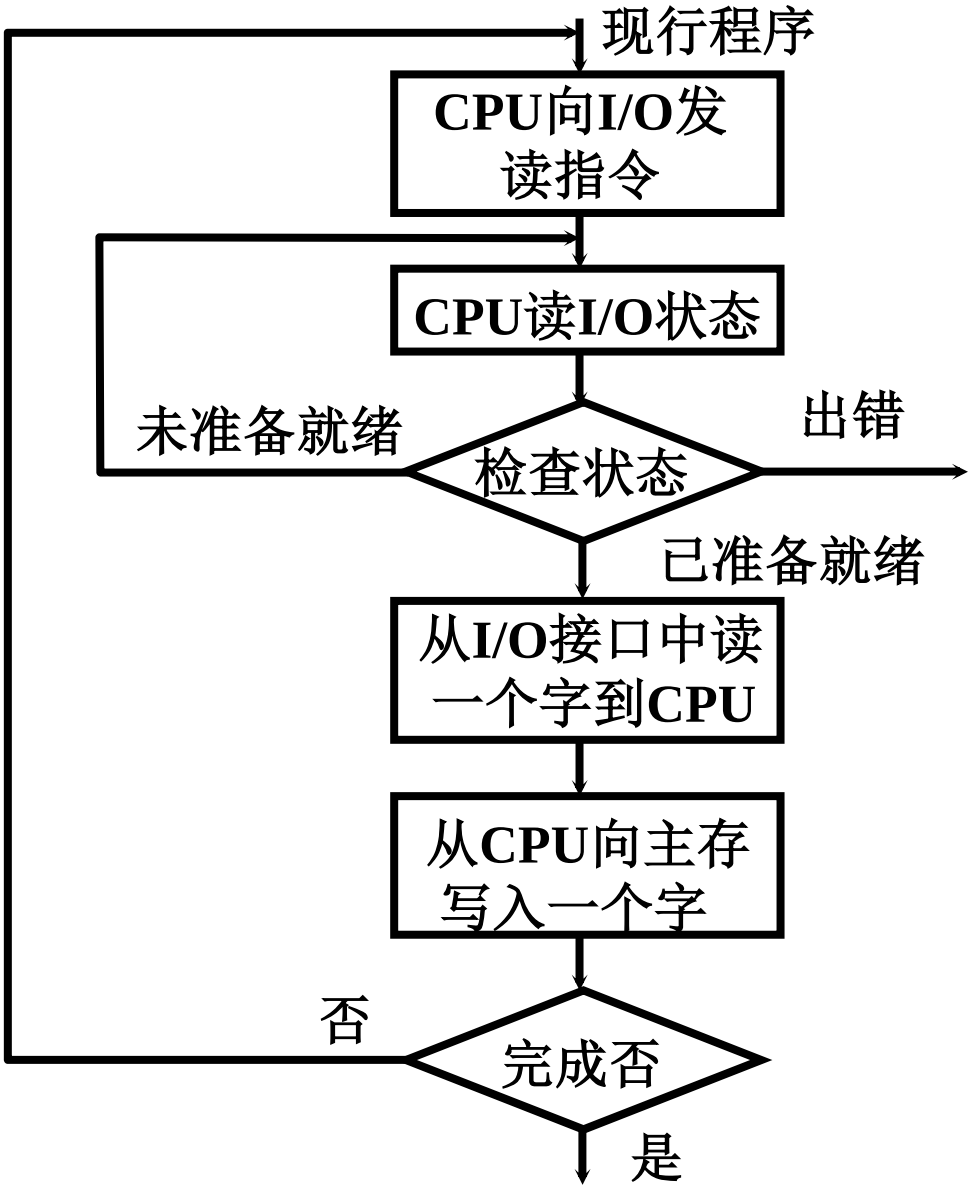
# 四、I/O设备与主机信息传送的控制方式 5.1

## 1. 程序查询方式

**CPU 和 I/O 串行工作**

**踏步等待**

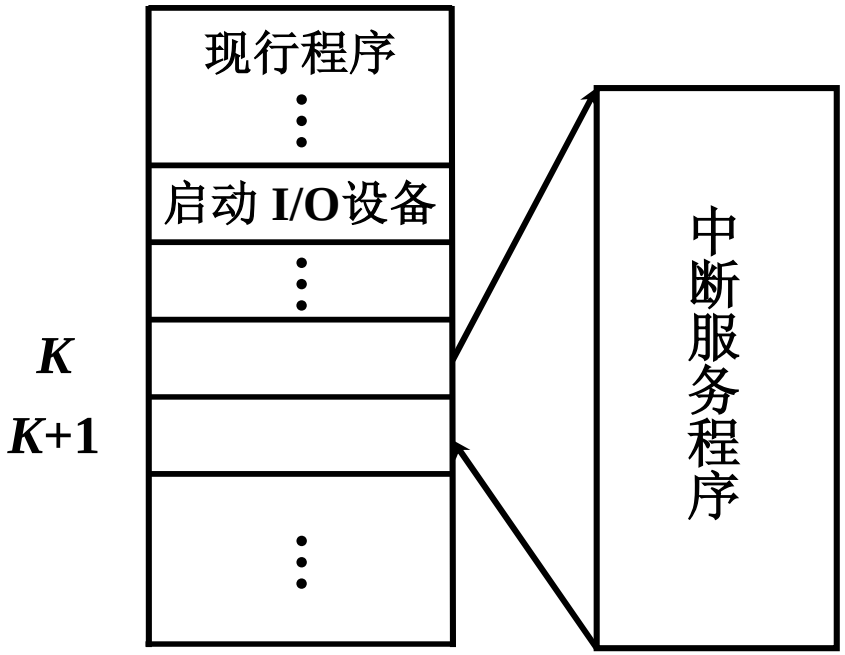


# 2. 程序中断方式

I/O 工作 {
- 自身准备　　CPU 不查询
- 与主机交换信息　CPU 暂停现行程序

**CPU 和 I/O 并行工作**

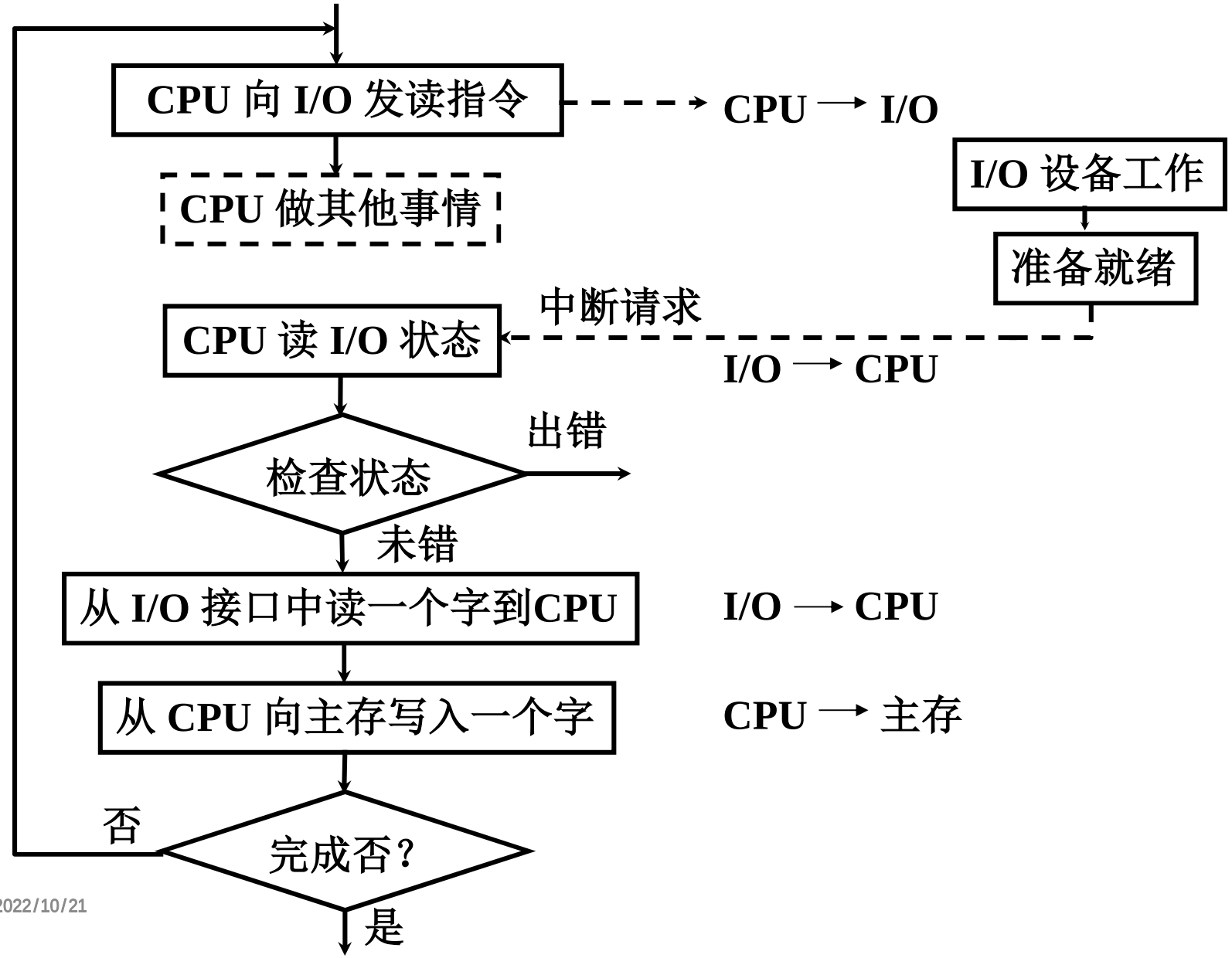



没有踏步等待现象

中断现行程序

# 程序中断方式流程




CPU 向 I/O 发读指令
CPU → I/O
I/O 设备工作
CPU 做其他事情
准备就绪
中断请求
CPU 读 I/O 状态
I/O → CPU
出错
检查状态
未错
从 I/O 接口中读一个字到CPU
I/O → CPU
从 CPU 向主存写入一个字
CPU → 主存
否
完成否？
是

# 3. DMA 方式

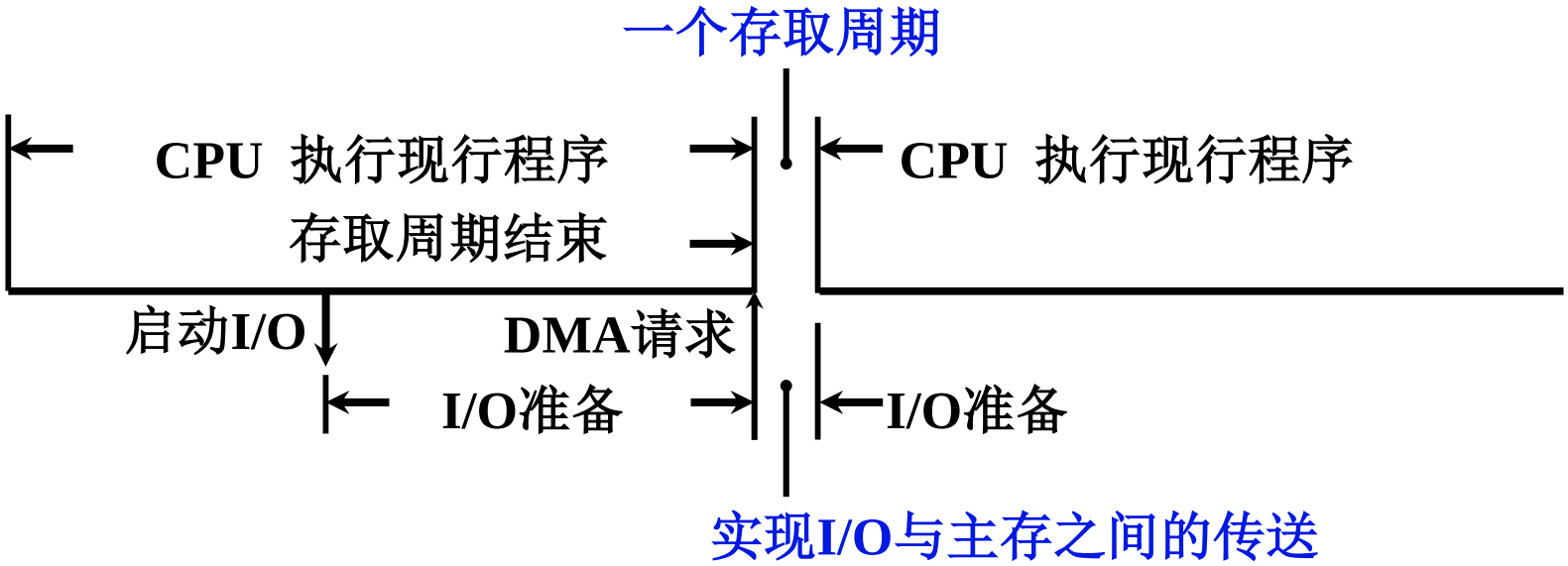


主存和 I/O 之间有一条直接数据通道

不中断现行程序

周期挪用（周期窃取）

**CPU 和 I/O 并行工作**



# 三种方式的 CPU 工作效率比较