

第九讲

第三部分：主存储器

- 一. 存储系统概述
- 二. 存储单元电路
- 三. 存储器芯片结构
- 四. 存储器扩展
- 五. DRAM的刷新

1.1 存储系统概述

❖ 存储器分类

➤ 按介质分类:

- 半导体存储器（易失性）
- 磁介质存储器（非易失性）
- 光盘存储器（非易失性）

➤ 按访问方式分类:

- 随机访问存储器（**Random Access Memory—RAM**）
- 顺序访问存储器（**Tape**）
- 直接访问存储器（**Disk**）
- 只读存储器（**Read Only Memory—ROM**）

➤ 按功能分类:

- 高速缓冲存储器（**Cache**）
- 主存储器
- 辅助存储器
- 控制存储器



1.1 存储系统概述

❖ 存储器的性能指标

➤ 访问时间 (Access Time) : T_A

- 随机访问存储器：访问时间指读或写操作所用时间，即从给定地址到存储器完成读或写操作所需时间。
- 其他类型：指将读写机构定位到目标位置所需的时间。

➤ 存储周期 (Cycle Time) : T_C

- 仅对RAM而言，指两次访问存储单元间的最小时间间隔。
- $T_C > T_A$

➤ 带宽 (Bandwidth) / 数据传输率 (Transfer Rate)

- 一般的随机访问存储器：1 / Cycle Time;
- 其他类型： $T_N = T_A + N/R$ ， 带宽： N / T_N

T_N : 读写N Bits所需的平均时间

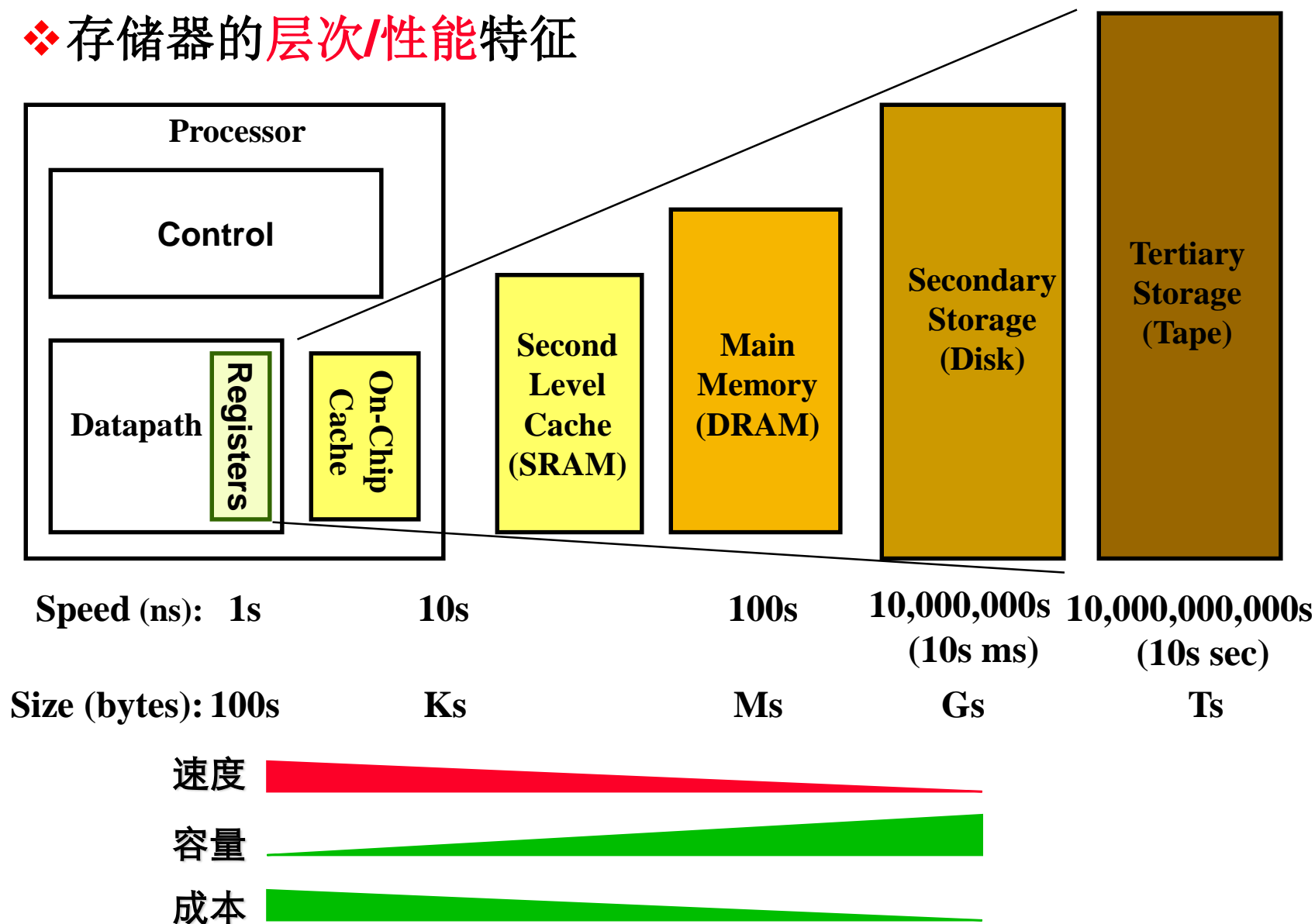
T_A : 访问时间

N: N Bits

R: 存储部件的数据传输率 (bits /s)

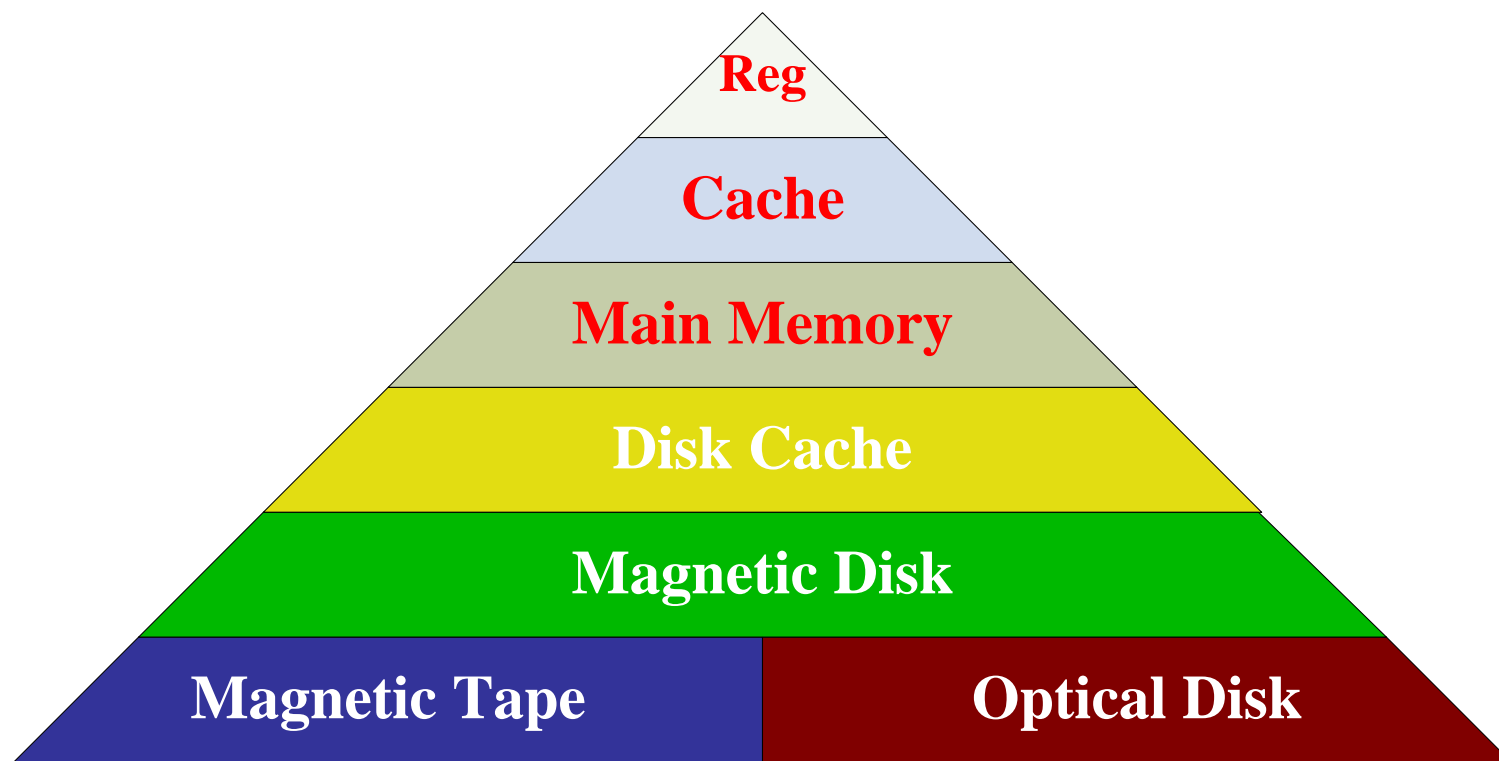
1.1 存储系统概述

❖ 存储器的层次/性能特征



1.1 存储系统概述

❖ 存储器的层次结构



二级存储系统：高速缓冲存储器（**Cache**）+主存储器

1.2 半导体存储器

❖ 半导体存储器从访问方式上可分：

➤ 随机访问存储器RAM、只读存储器ROM

❖ RAM从实现原理上，又可分为：

➤ 静态随机访问存储器SRAM（Static RAM）

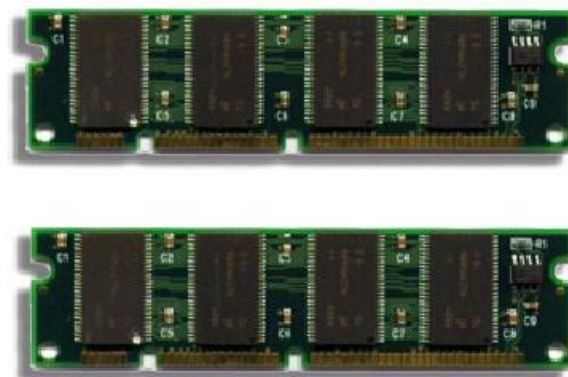
■ 静态存储器，相对动态而言，集成度低，不必刷新。用作Cache。

➤ 动态随机访问存储器DRAM（Dynamic RAM）

■ 动态存储器，需要刷新，相对而言，集成度高。用做主存。



SRAM



DRAM

1.2 半导体存储器

❖ 主流DRAM

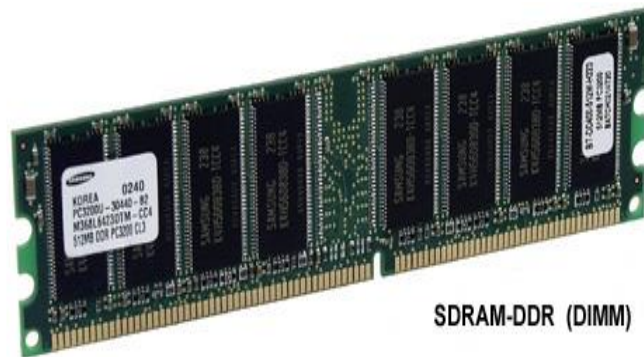
➤ SDRAM (Synchronous DRAM) :

同步DRAM，与CPU采用相同时钟，避免了不必要的等待周期，减少数据存储时间，数据可在脉冲上升期便开始传输。SDRAM内存又分PC66、PC100、PC133等不同规格，相应带宽分别为528MB/S、800MB/S和1.06GB/S。



SDRAM (DIMM)

➤ **DDR (Double Data Rate) SDRAM:** 双倍速率SDRAM。SDRAM只在一个时钟的上升期传输一次数据；而DDR内存则在一个时钟的上升期和下降期各传输一次数据，因此称为双倍速率SDRAM。DDR SDRAM可以在与SDRAM相同的总线频率下达到更高的数据传输率。



SDRAM-DDR (DIMM)

1.2 半导体存储器

❖ 主流DRAM

► **DDR2 (Double Data Rate 2) SDRAM:**

DDR2内存拥有两倍于DDR内存预读取能力，即：DDR2内存 每个时钟能够以4倍外部总线的速度读/写数据，例如，在同样100MHz的工作频率下，DDR的实际频率为200MHz，而DDR2则可达到400MHz。DDR2内存采用1.8V电压，相对于DDR标准的2.5V，降低了不少。



SDRAM-DDR2 (DIMM)

► **DDR3 (Double Data Rate 3) SDRAM:**

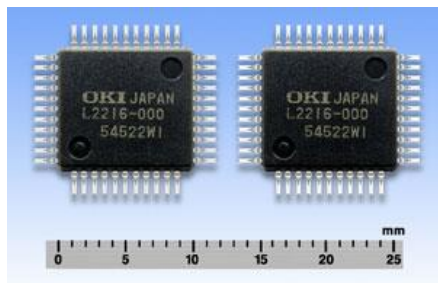
最初主要用于显卡内存，频率在800M以上。DDR3是在DDR2基础上采用的新型设计，与DDR2 SDRAM相比具有功耗和发热量较小、工作频率更高、降低显卡整体成本、通用性好的优势。DDR3内存工作电压1.5V，DDR3内存预读取能力为DDR2的二倍。



1.2 半导体存储器

❖ 只读存储器（ROM）—— 非易失性

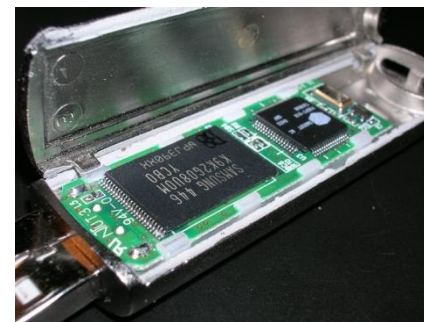
- 固定掩膜（Masks）ROM
- PROM（Programmable ROM）：一次性可编程
- EPROM（Erasable PROM）：可擦除可编程（紫外线擦除）
- EEPROM（Electrically Erasable PROM）：电擦除
- Flash Memory（闪存）：本质上属于电擦除可编程ROM，如SM（Smart Media）卡、CF（Compact Flash）卡，MMC（Multi Media Card）卡、SD（Secure Digital）卡和记忆棒（Memory Stick）等。



Mask ROM

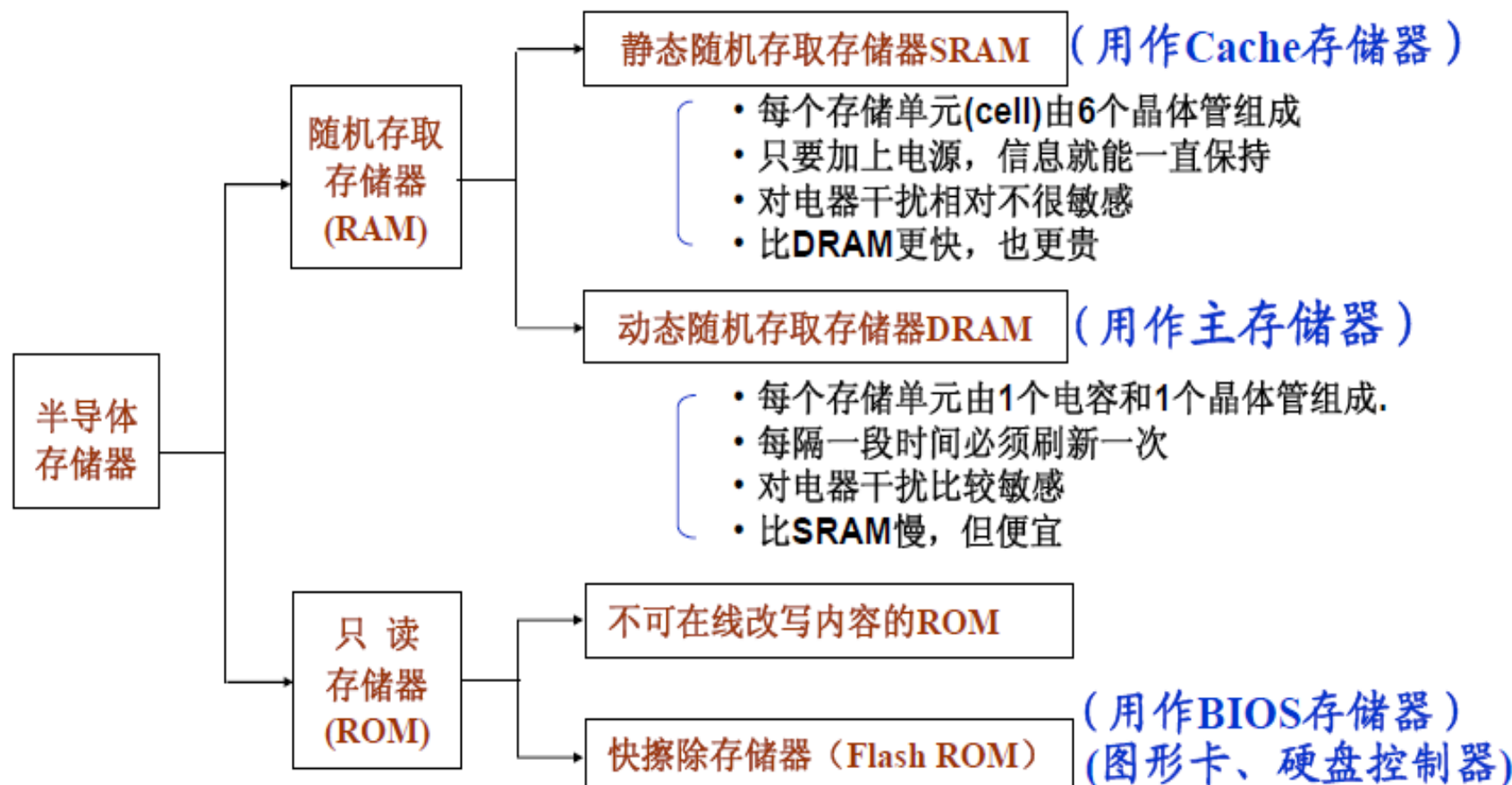


EPROM



U盘上的Flash memory

1.2 半导体存储器



第三部分：主存储器

- 一. 存储系统概述
- 二. 存储单元电路
- 三. 存储器芯片结构
- 四. 存储器扩展
- 五. DRAM的刷新

2.1 存储单元电路

❖ 存储单元电路

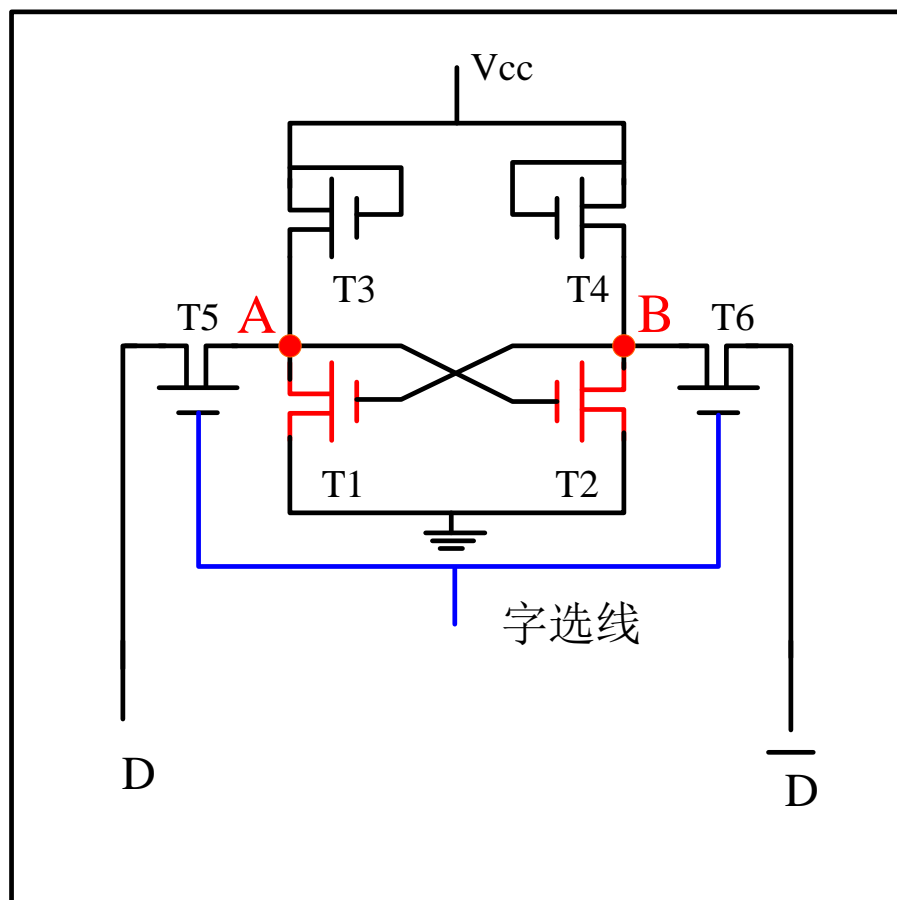
- 存储器中用来存储一位二进制信息（0或1）的电路
- 是组成存储器的基础和核心
- 也称存储元件、存储基元、存储位元、存储元

❖ 基本要求

- 具有两种稳定（或半稳定）状态，用来表示二进制的 0和1
- 可以实现状态写入（或设置）
- 可以实现状态读出（或感知）

2.2 SRAM存储单元电路

❖ SRAM存储单元电路（六管单元电路）



MOS管功能:

T1, T2: 工作管;

T3, T4: 负载管;

T5, T6: 门控管;

稳定状态:

“1”: T1 截止, T2 导通

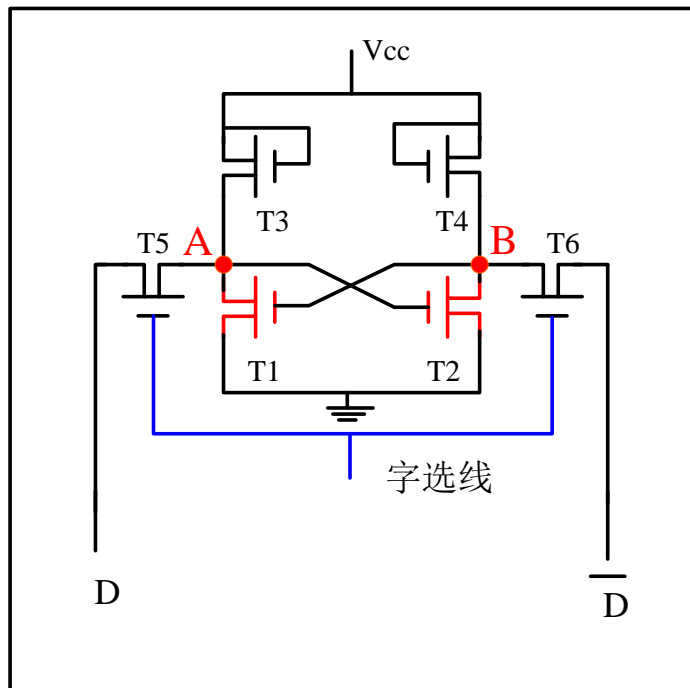
“0”: T2 截止, T1 导通

保持状态:

字选线低电平, T5 和 T6 截止, 内部保持稳定。

2.2 SRAM存储单元电路

❖ SRAM存储单元电路工作原理（写入）



稳定状态:

“1”: T1 截止, T2 导通

“0”: T2 截止, T1 导通

保持状态:

字选线低电平, T5 和 T6 截止, 内部保持稳定。

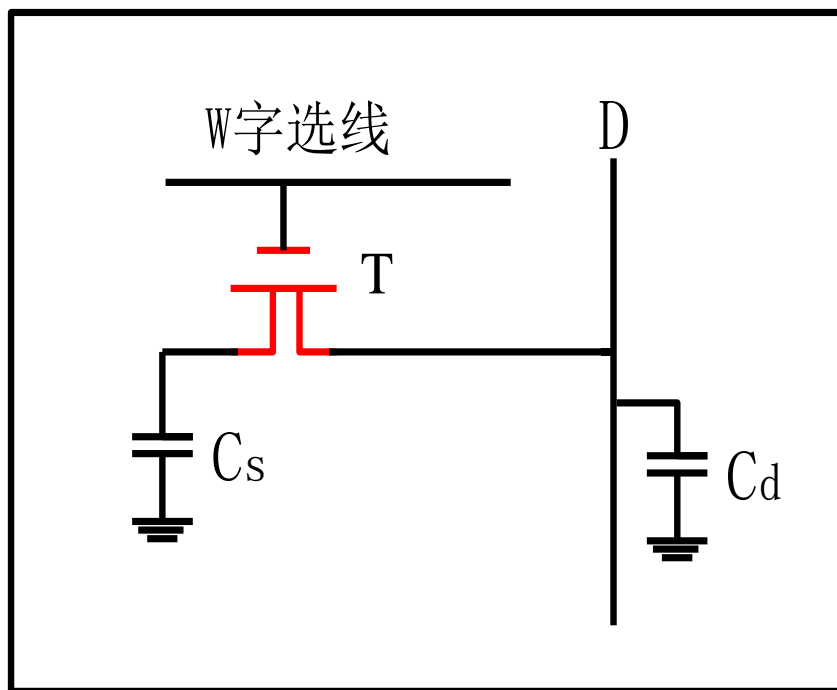
写入操作:

写 1: D 线高电平, \overline{D} 线低电平, 字选线高电平, T5 和 T6 导通, T1 截止, T2 导通, 写入 1。

写 0: D 线低电平, \overline{D} 线高电平, 字选线高电平, T5 和 T6 导通, T2 截止, T1 导通, 写入 0。

2.3 DRAM存储单元电路

❖ DRAM存储单元电路（单管单元电路）

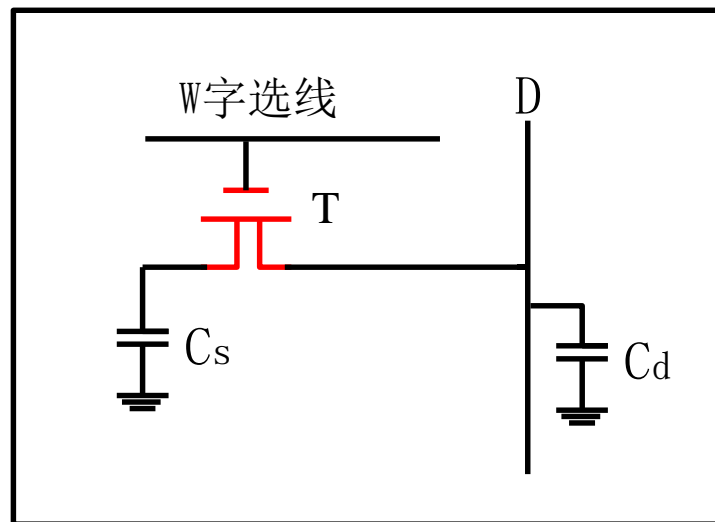


- C_s 电容 $\ll C_d$ 电容
- C_s 上有电荷表示 ‘1’
- C_s 上无电荷表示 ‘0’
- 保持状态：字选线低电平，T截止，理论上内部保持稳定状态。

注意：在保存二进制信息 “1” 的状态下， C_s 有电荷，但 C_s 存在漏电流， C_s 上的电荷会逐渐消失，状态不能长久保持，在电荷泄漏威胁到所保存的数据性质之前，需要补充所泄漏的电荷，以保持数据性质不变。这种电荷的补充称之为刷新（或再生）。

2.3 DRAM存储单元电路

❖ DRAM存储单元电路工作原理（读出）

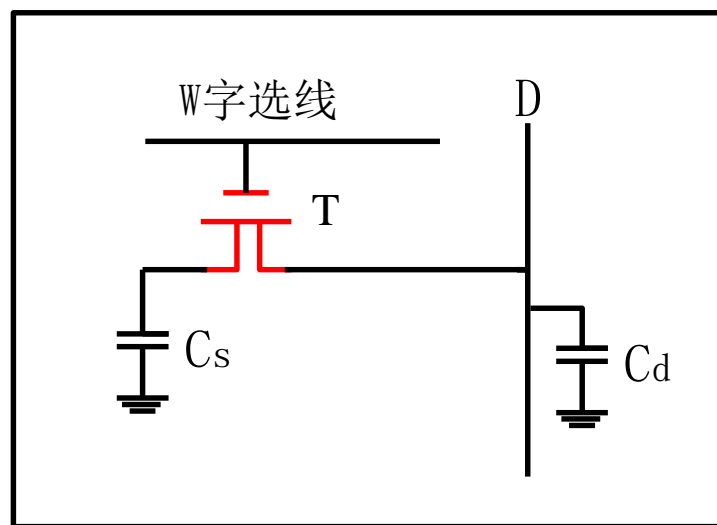


读出时：**D** 线先预充电到 $V_{pre}=2.5V$ ，然后字选线高电平，**T** 导通

- 若电路保存 信息**1**（ $V_{cs}=3.5V$ ），电流方向从单元电路内部向外；
- 若电路保存信息 **0**（ $V_{cs}=0.0V$ ），电流方向从外向单元电路内部；
- 因此，根据数据线上电流的方向，可判断单元电路保存的是 **1** 还是 **0**。
- 读出过程实际上是**Cs**与**Cd**上的电荷重新分配的过程，也是**Cs**与**Cd**上的电压重新调整的过程。**Cd**上的电压，即是**D**线上的电压。

2.3 DRAM存储单元电路

❖ DRAM存储单元电路工作原理（写入）

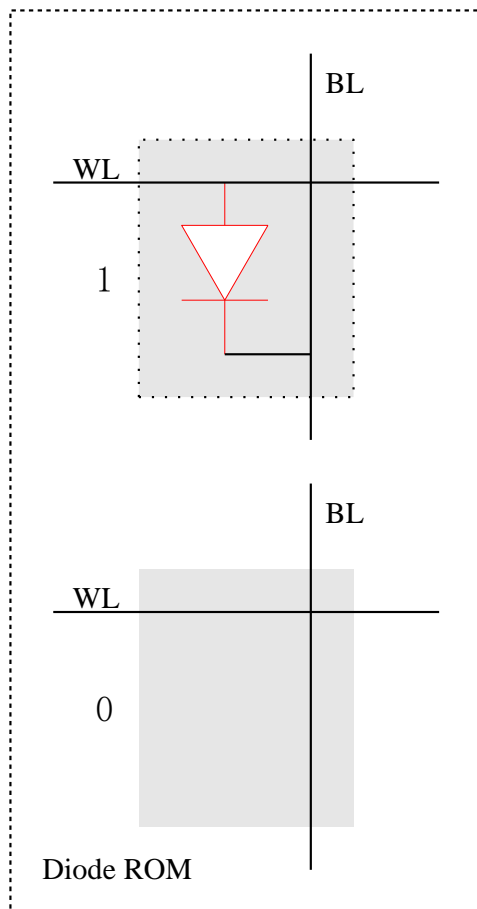


写入操作：**D** 线加高电平（1）或低电平（0），字选择线置高电平，**T**导通；

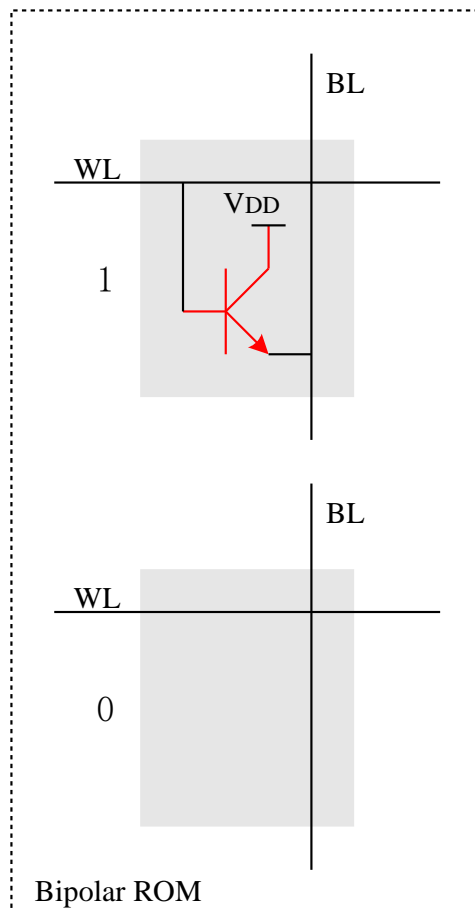
- 写1时，**D**线高电平，对**Cs**充电；
- 写0时，**D**线低电平，**Cs**放电；

2.4 ROM存储单元电路

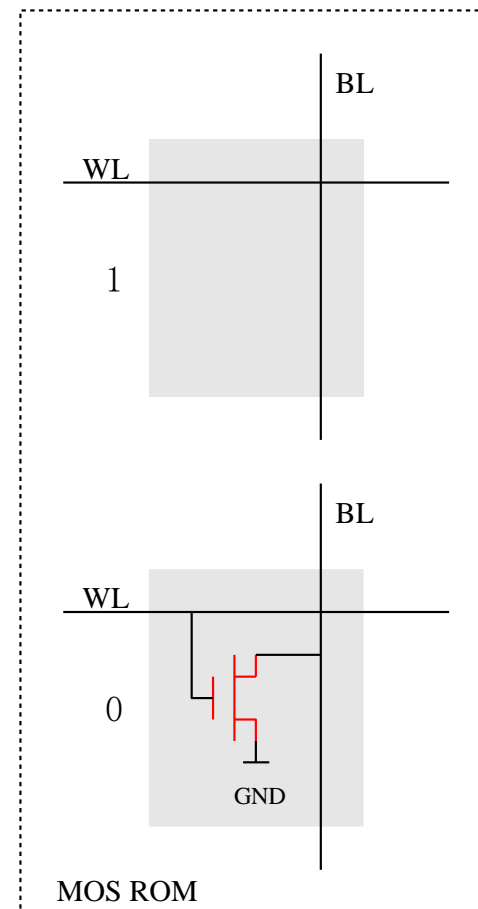
❖ 固定掩膜ROM单元电路（不能重写，非易失性）



含二级管的电路
表示1，不含电
路表示0



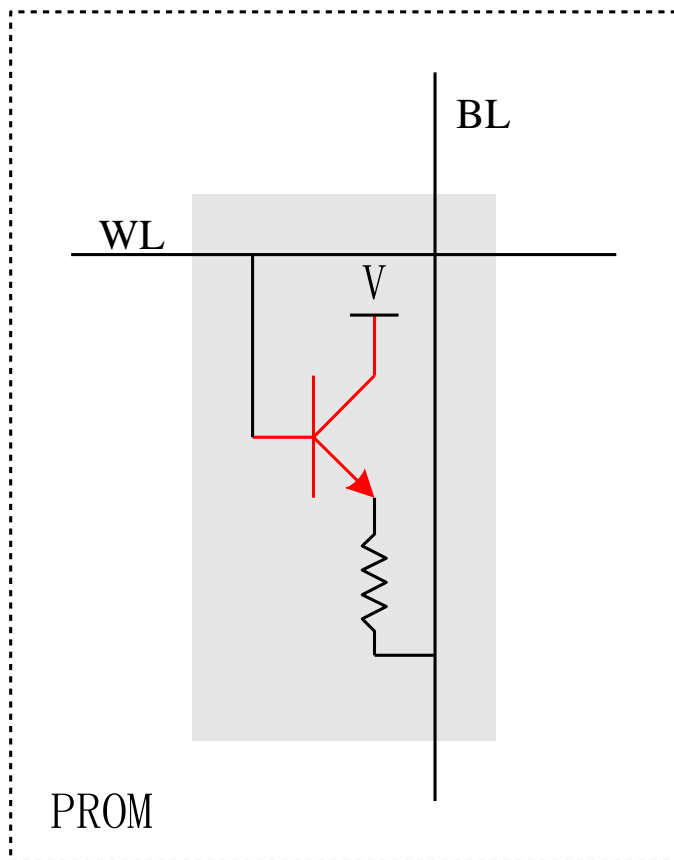
含三极管的电路
表示1，不含电
路表示0



含MOS管的电路
表示0，不含电
路表示1

2.4 ROM存储单元电路

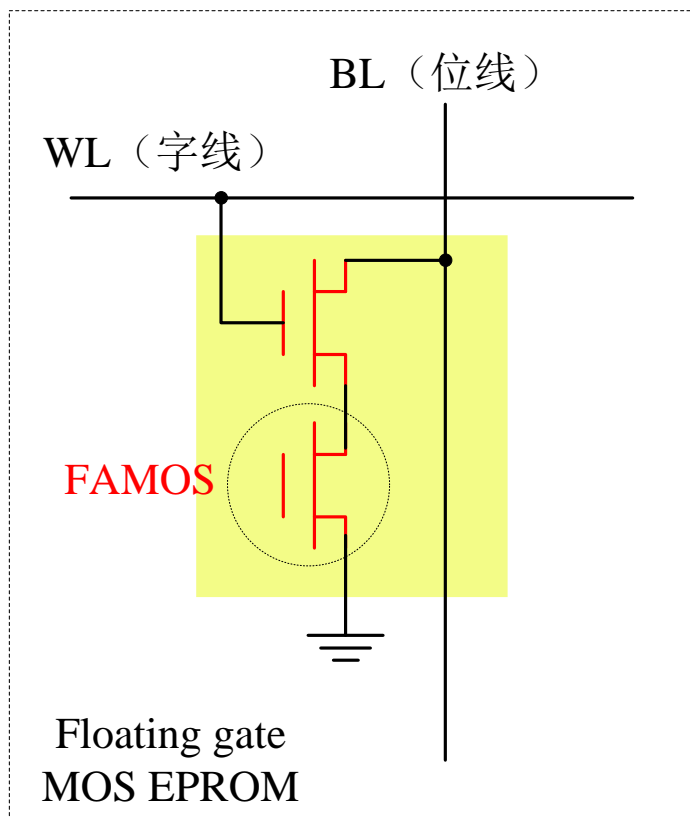
❖ 可编程的PROM单元电路 (一次性改写)



- 出厂时所有位均为1。
- 编程时（写入数据），对写0的单元加入特定的大电流，熔丝被烧断，变为另一种表示0的状态，且不可恢复。
- 工作时，加入正常电流。

2.4 ROM存储单元电路

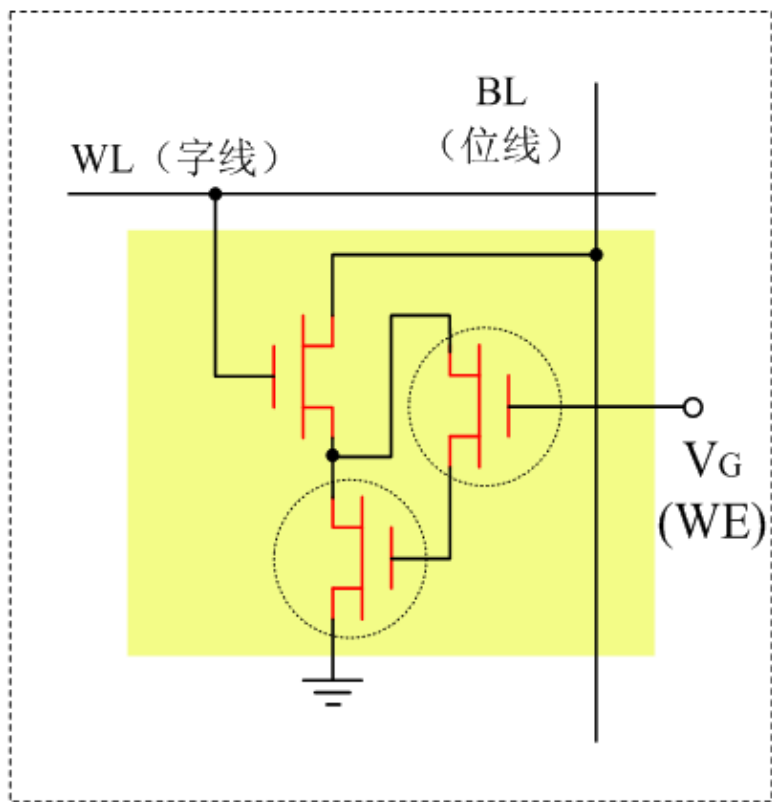
❖ 紫外线擦除可编程的EPROM单元电路（可多次改写）



- 出厂时所有位均为 **1**，FAMOS（浮空栅极MOS）G极无电荷，处于截止状态。
- 编程时（写入数据），对写**0**的单元加入特定的电压，FAMOS上的G极与D极被瞬时击穿，大量电子聚集到G极上，撤销编程电压后，G极上的聚集的电子不能越过隔离层，FAMOS导通，表示0。
- 工作时，加入正常电压，FAMOS的状态维持不变。
- 擦除时，用紫外线照射，FAMOS聚集在G极上的电子获得能量，越过隔离层泄漏，FAMOS恢复截止状态。

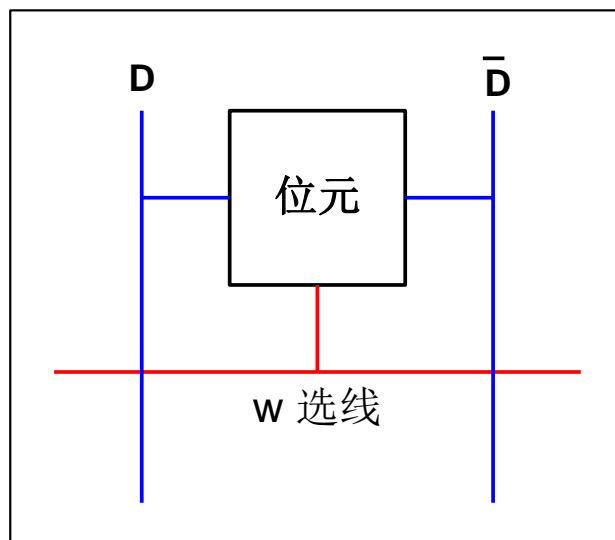
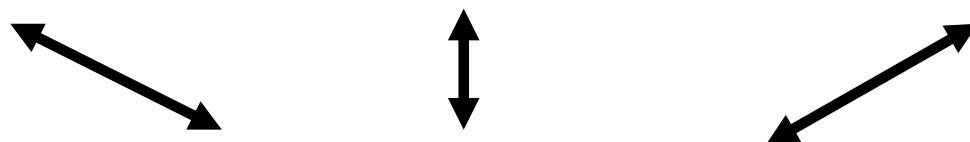
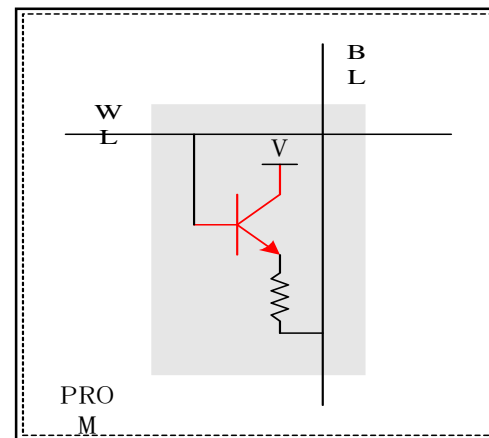
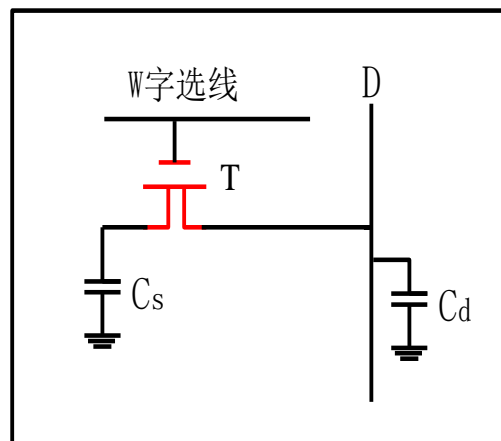
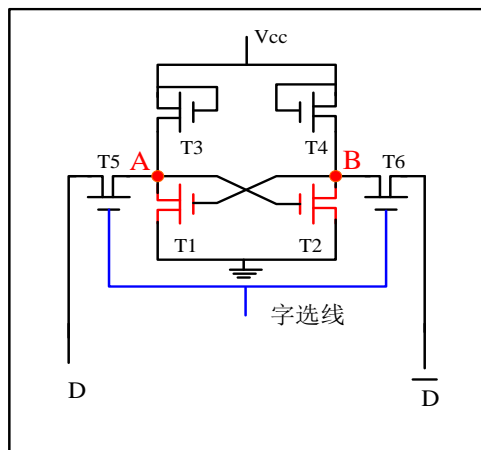
2.4 ROM存储单元电路

❖ EEPROM单元电路（可多次改写）

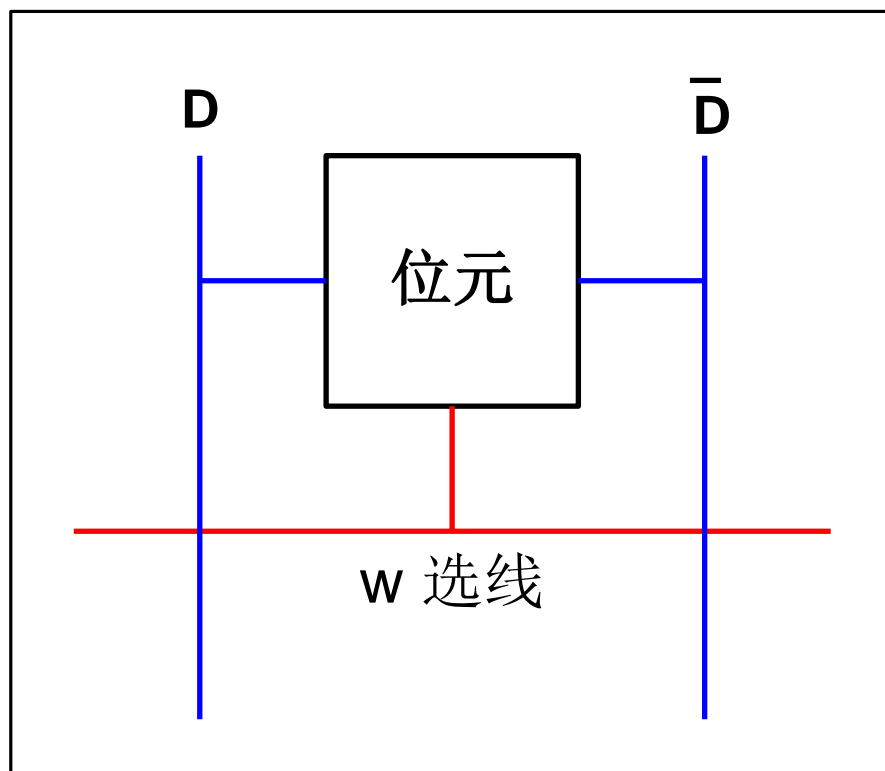


- 与EPROM相似，它是在EPROM基本单元电路的浮空栅的上面再生成一个浮空栅，前者称为第一级浮空栅，后者称为第二级浮空栅。第二级浮空栅引出一个电极，接某一电压 V_G 。
- 若 V_G 为正电压，第二浮空栅极与漏极之间产生隧道效应，使电子注入第一浮空栅极，即编程写入。
- 若使 V_G 为负电压，强使第一级浮空栅极的电子散失，即擦除。擦除后可重新写入。

2.5 存储单元电路的符号表示



存储单元电路：可存储1位(1bit)二进制代码



存储单元电路（存储位元） → 存储芯片 → 存储器

第三部分：主存储器

- 一. 存储系统概述
- 二. 存储单元电路
- 三. 存储器芯片结构
- 四. 存储器扩展
- 五. DRAM的刷新

2.2 存储芯片内部结构

❖ 存储芯片容量的基本描述（字单元数 × 每个字单元的位数）

➤ $1K \times 2$ ：1024 个字单元，每个字单元 2 位（二进制位）

意味着任一时刻可以（也只能）访问1024个独立字单元中的任意一个，每次读写的数据位数是一个字单元的容量（2位）

对于 $1K \times 2$ 的存储芯片：

有多少个存储位元？共1K个（1024个）字单元，每个字单元2位 2048

需多少条地址线？按字单元寻址，1K个（1024个 或 2^{10} 个）字单元 10

需要多少条数据线？一次访问一个字单元，每个字单元是2位 2

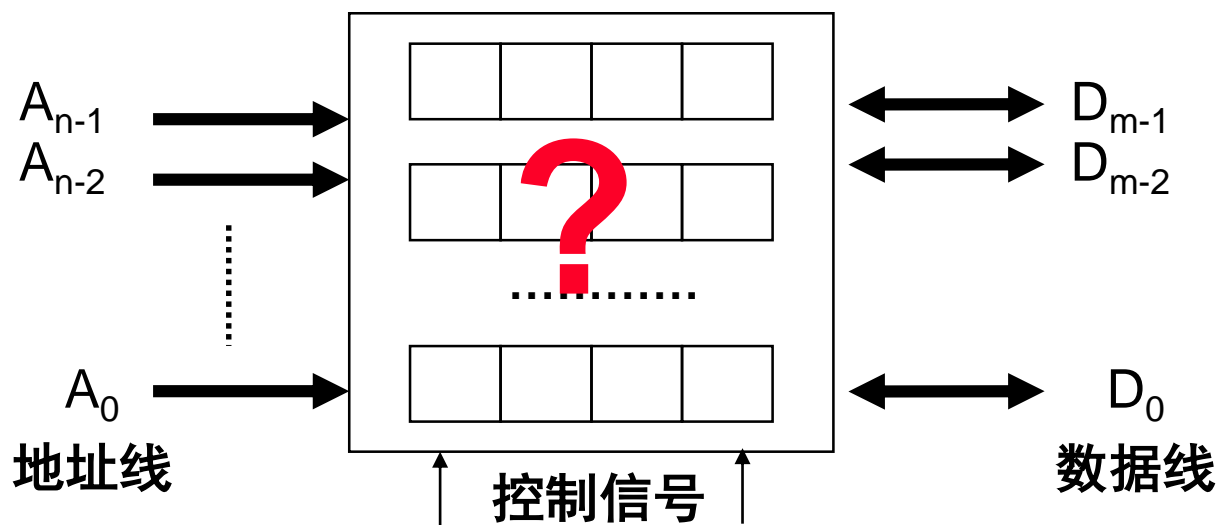
➤ $64K \times 8$ ：65536（64K）个字单元，每个字单元8位

有多少个存储位元？需要多少条地址线？多少条数据线？

2.2 存储芯片内部结构

存储芯片容量的描述: $2^n \times m$ (字单元数 \times 每个字单元的位数)

- ❖ 存储位元: $2^n \times m$ 个
- ❖ 地址线: n 位 $\rightarrow 2^n$ 个字单元, $A_{n-1}..A_0$ 单向
- ❖ 数据线: m 位 $\rightarrow m$ 位/字单元, $D_{m-1}..D_0$ 双向

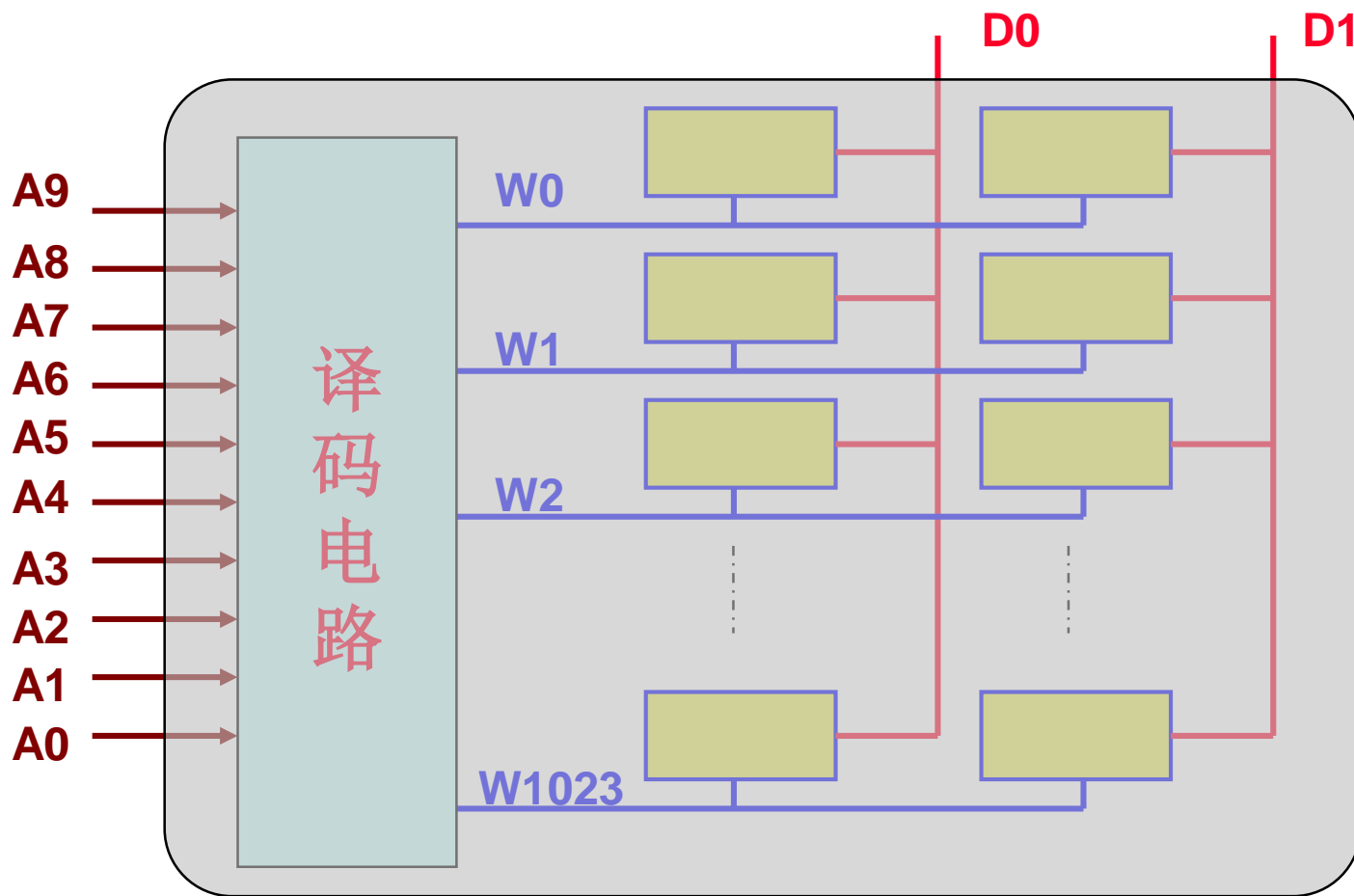


一维地址结构(线选法) / 二维地址结构(重合法)

2.2 存储芯片内部结构

❖ 存储芯片结构（一维地址结构）

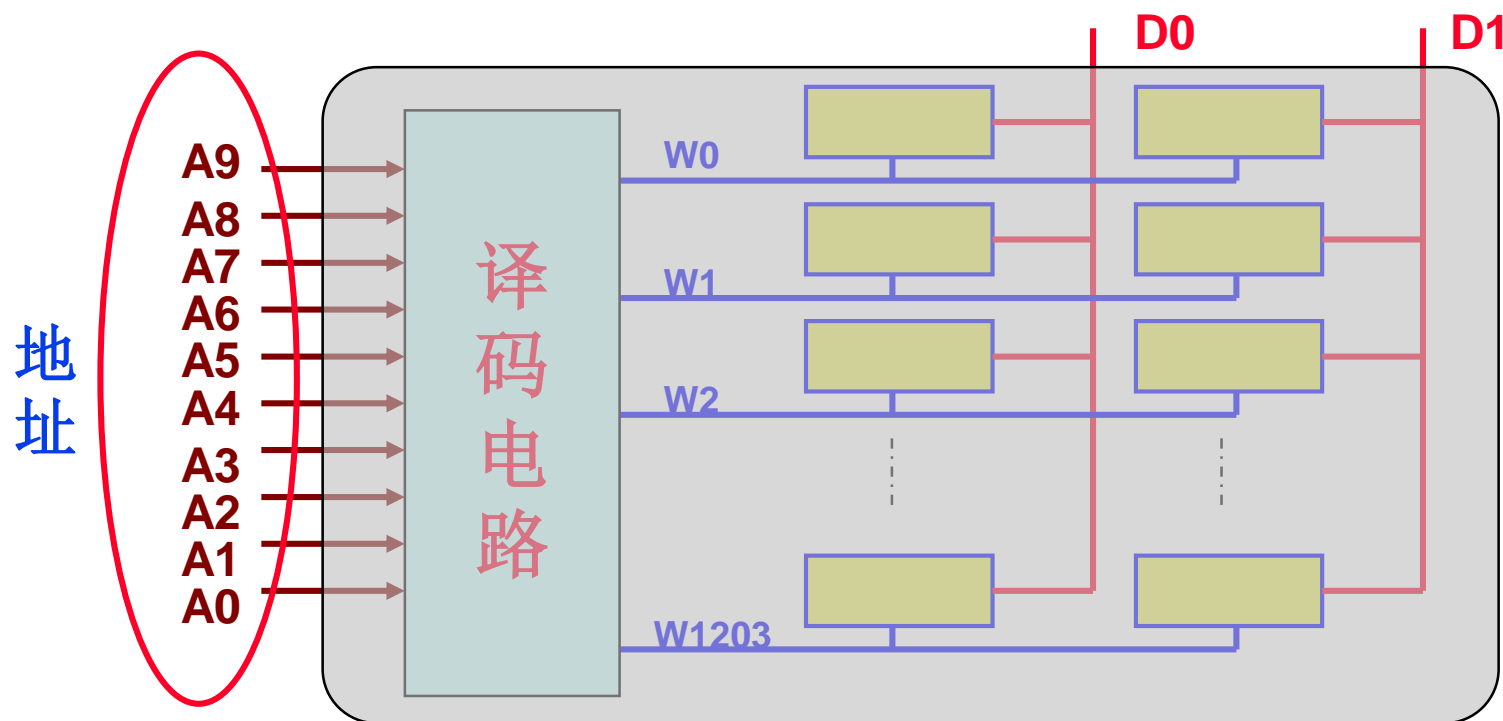
1024×2 ：1024 个字单元，每个字单元 2 个二进制位，共2048个存储位元



2.2 存储芯片内部结构

❖ 问题：如何识别这些字单元？

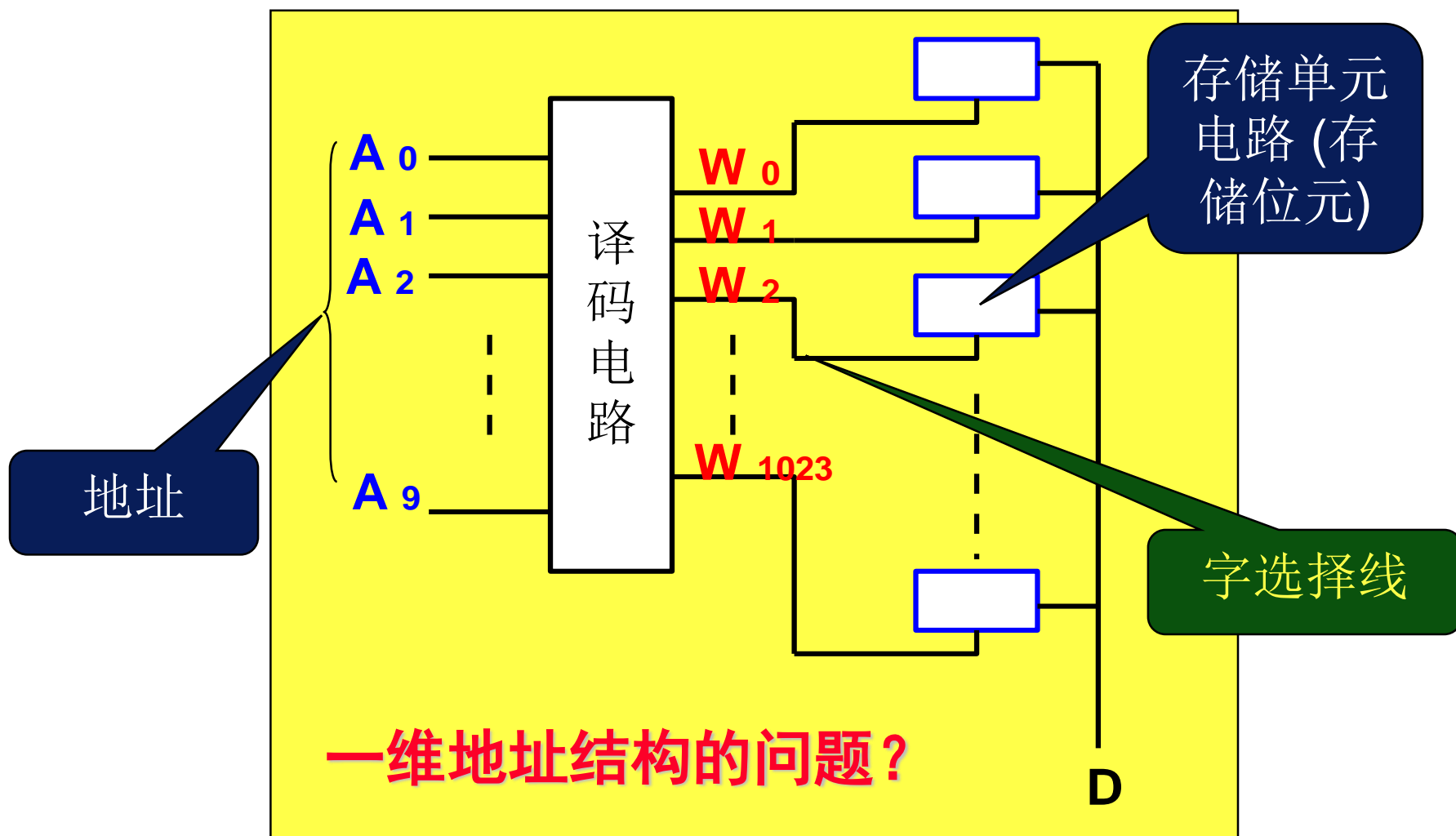
- 1024×2 ：1024 个字单元，需要1024个不同的标示
- 地址编码：译码电路使得字选择线 W_i 处于工作状态的输入信号（2进制信号），称为 W_i 所选中字单元的地址编码（简称地址）
- 对于每一个字单元，地址是唯一的



2.2 存储芯片内部结构

❖ 存储芯片结构（一维地址结构）

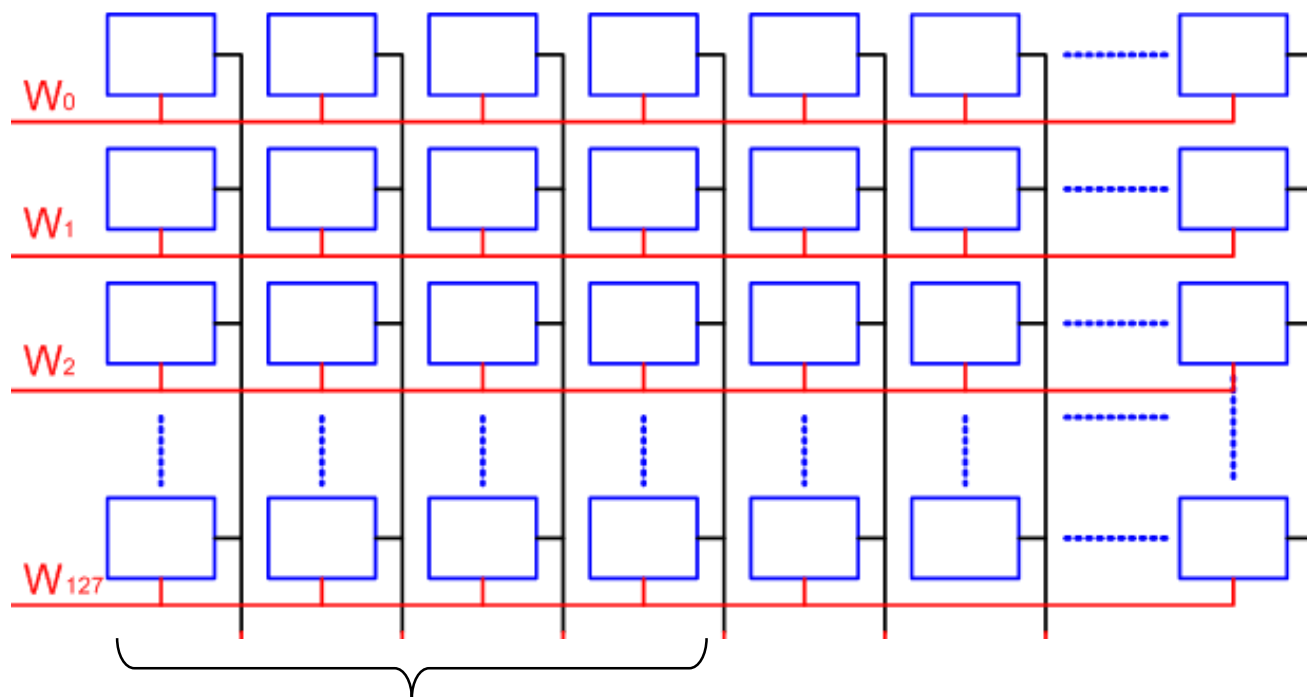
1024*1：1024 个字单元，每个字单元 1 个二进制位。



2.2 存储芯片内部结构

❖ 二维地址结构（SRAM）： 容量 4096 X 4

- 4096 个字单元，每个字单元 4 位
- $4096 * 4 = 2^{14}$ 个存储位元
- 存储矩阵： $2^7 * 2^7$ (128行*128列)



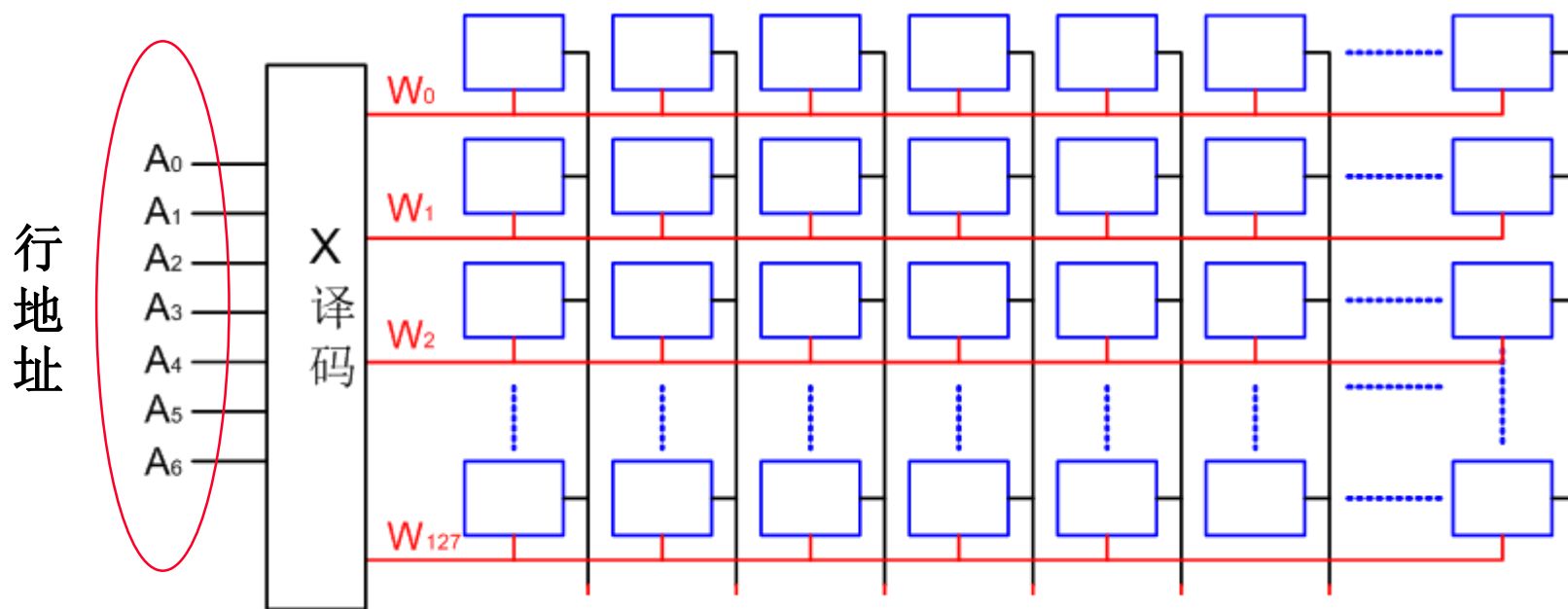
一行共有128个位元，每4个位元一组，组成1个字单元，一行共32个字单元

2.2 存储芯片内部结构

❖ 二维地址结构（SRAM）：容量4096 X 4

➤ 存储矩阵： $2^7 * 2^7$ （128行*128列）

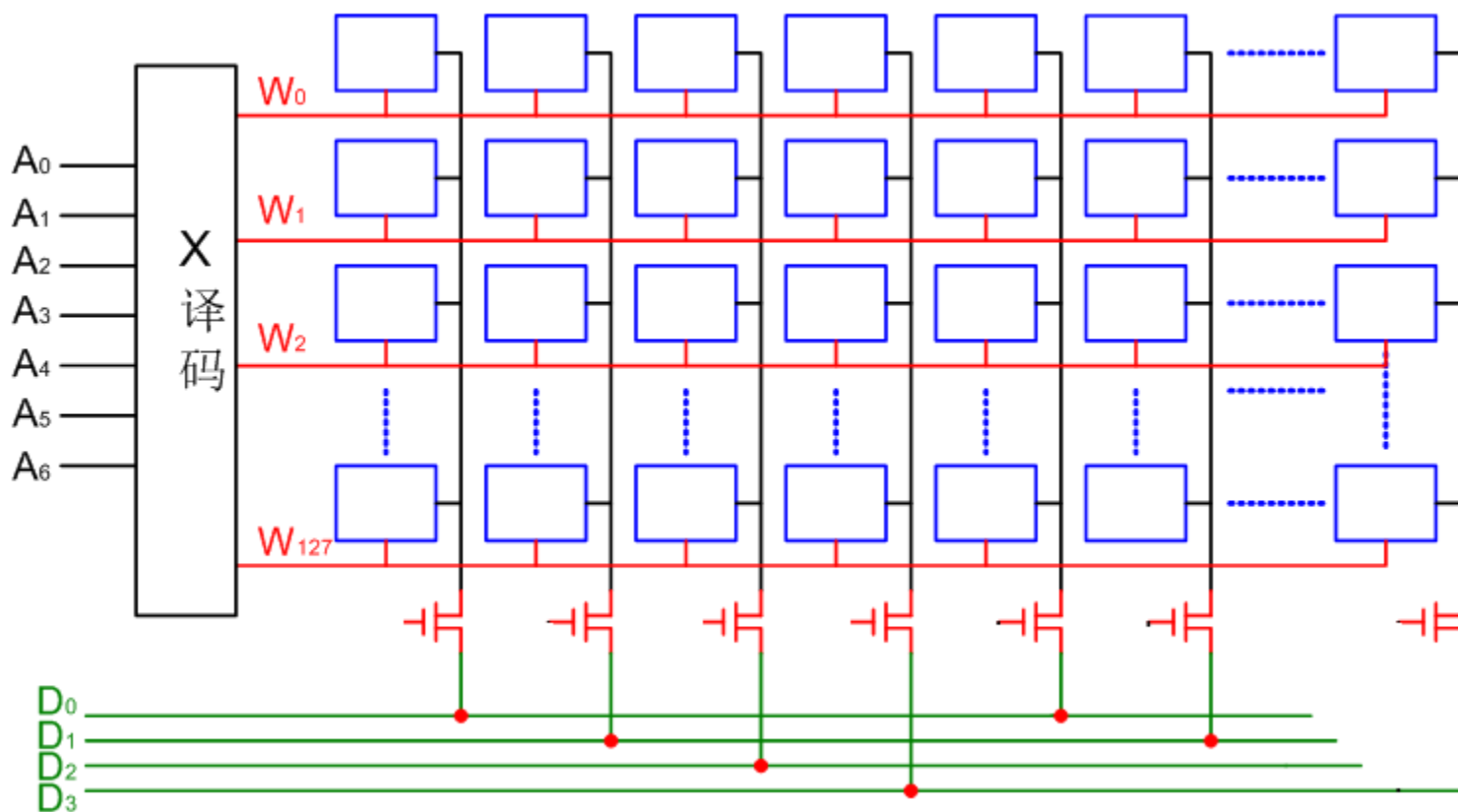
➤ 行译码：x译码，行地址需要 7位 $A_0A_1A_2...A_6$



2.2 存储芯片内部结构

❖ 二维地址结构（SRAM）：容量4096 X 4

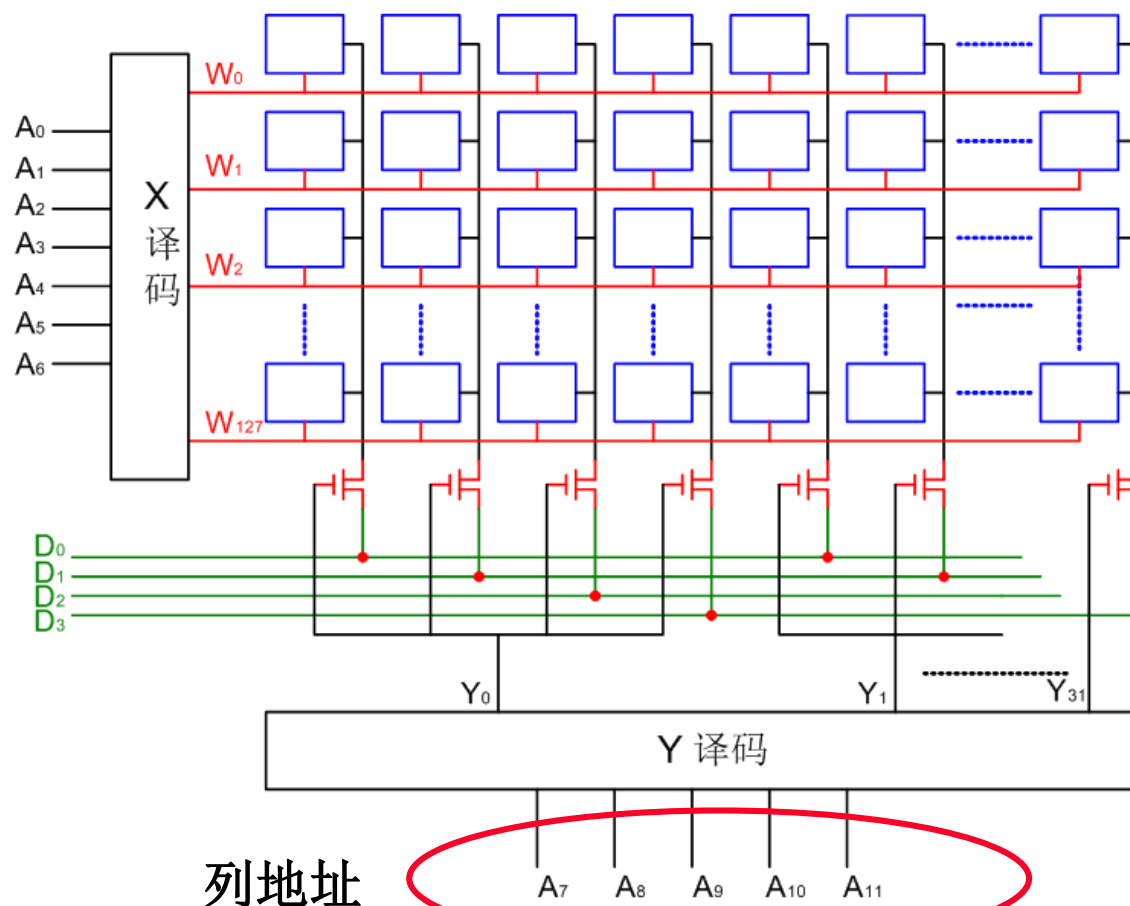
- 存储矩阵： $2^7 * 2^7$ （128行*128列）
- 一行包括32个字单元共128位，任何时刻只能其中1个字单元被选中，所以每个字单元的位线分别接到数据线D₀D₁D₂D₃



2.2 存储芯片内部结构

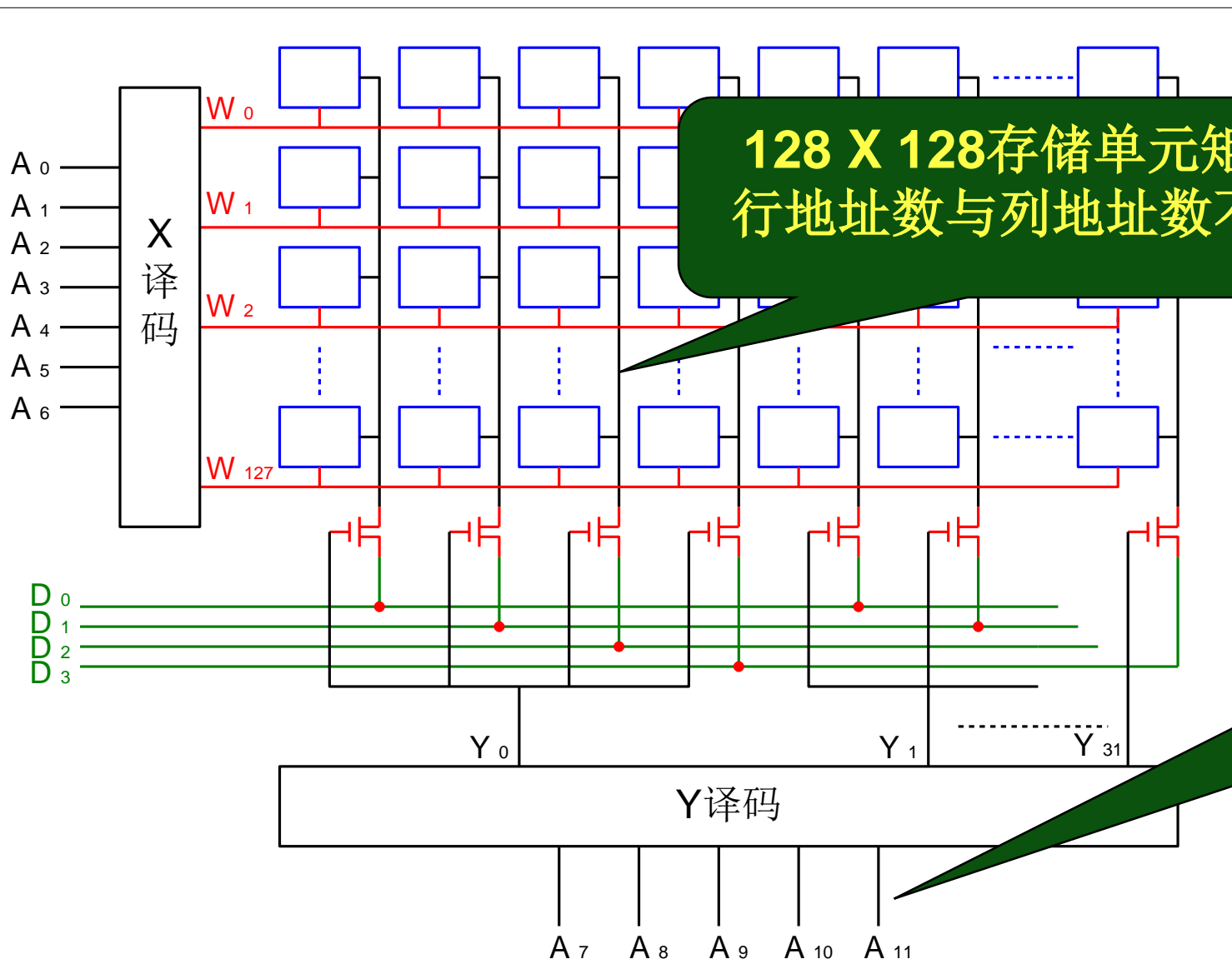
❖ 二维地址结构（SRAM）：容量 4096 X 4

- 存储矩阵： $2^7 * 2^7$ （128行*128列）
- 一行包括32个字单元，要进行32选1的译码（Y译码），列地址5位，为 $A_7A_8A_9A_{10}A_{11}$



2.2 存储芯片内部结构

❖ 二维地址结构（SRAM）：4096*4：4096 个字，每个字 4 位。

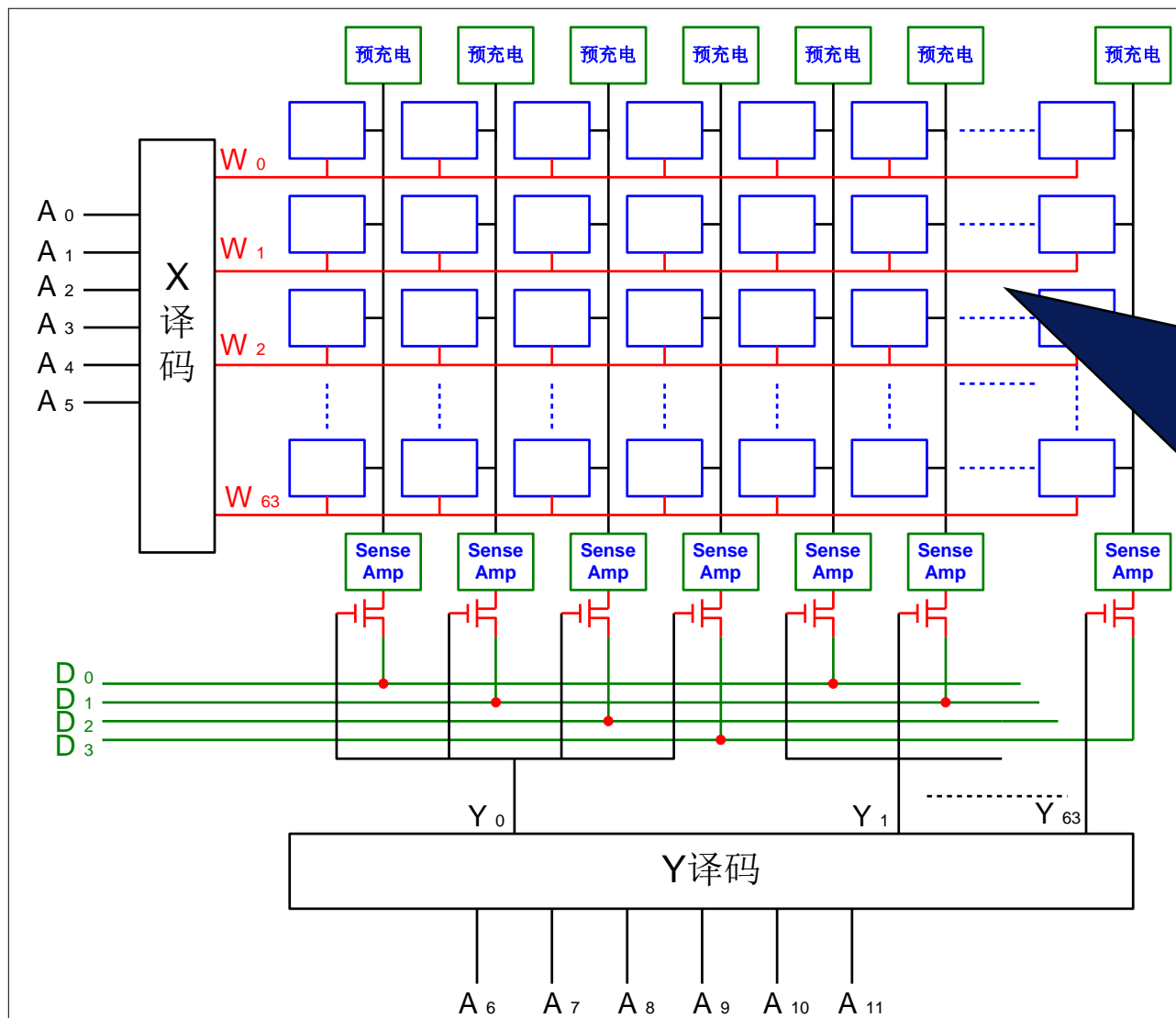


行地址

列地址

2.2 存储芯片内部结构

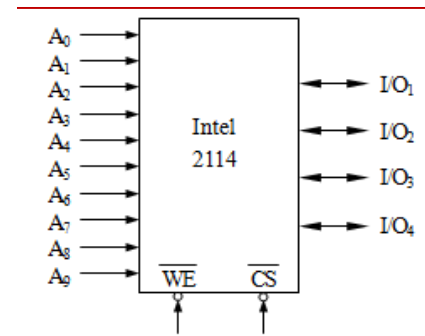
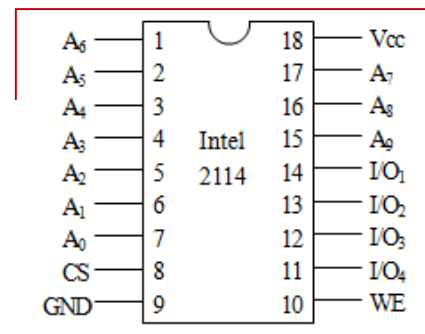
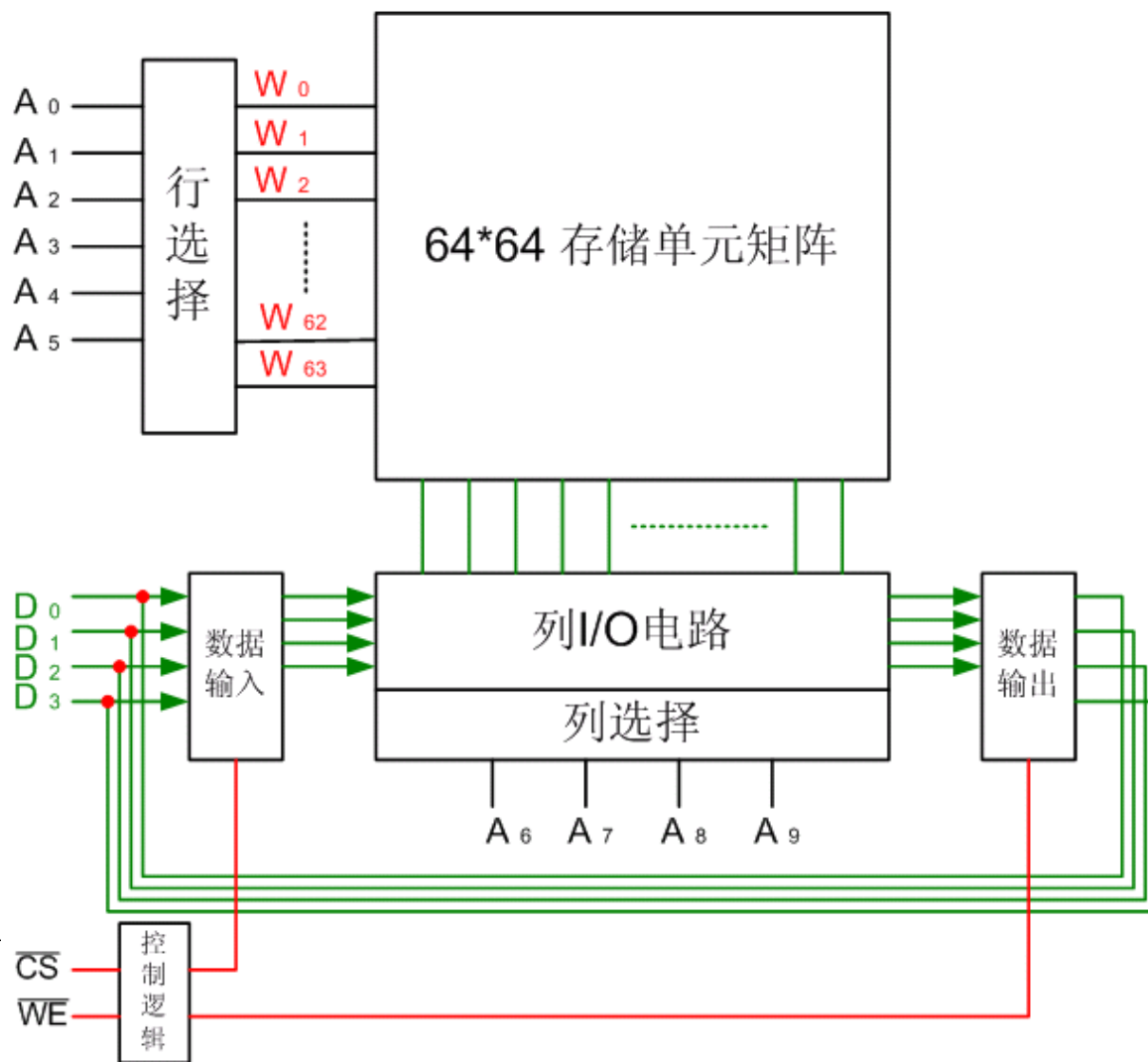
❖ 二维地址结构（DRAM）：4096*4：4096 个字，每个字 4 位。



64 X
256 存
储矩阵
→
行地址
数与列
地址数
相等

存储芯片结构示例

❖ SRAM 2114(1024*4)芯片结构

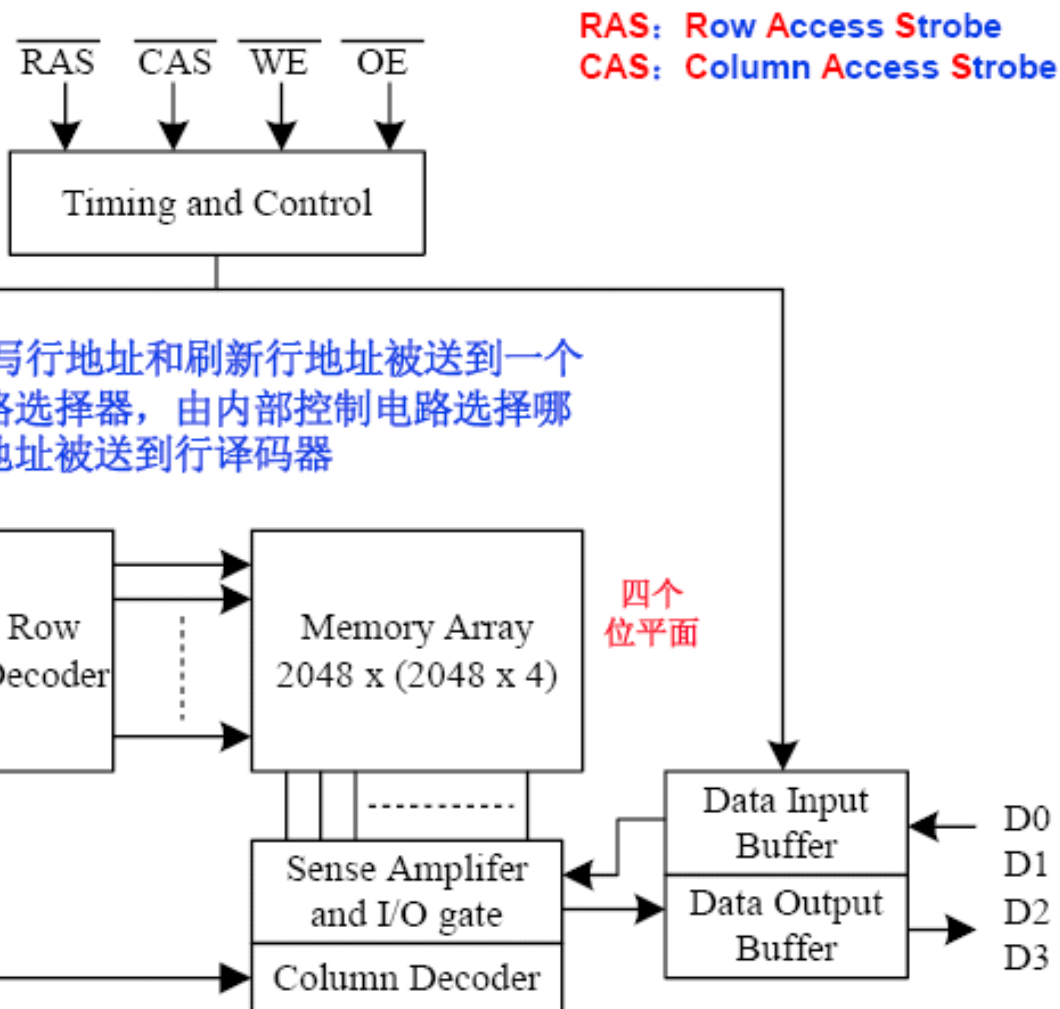


片选信号

存储芯片结构示例

❖ DRAM 4M*4 DRAM芯片结构 (内部包含刷新电路)

所有**DRAM**芯片同时刷新，由刷新计数器自动计数，按行刷新(只需行地址)，对**CPU**透明。



第十讲

上一讲简要回顾

❖ 存储系统概述

- 存储器的分类：介质、访问方式、功能
- 存储器的性能指标：访问时间、存储周期、带宽
- 存储器的层次结构：Reg-Cache-MM-2ndS-3rdS

❖ 主存储器 —— 存储单元电路

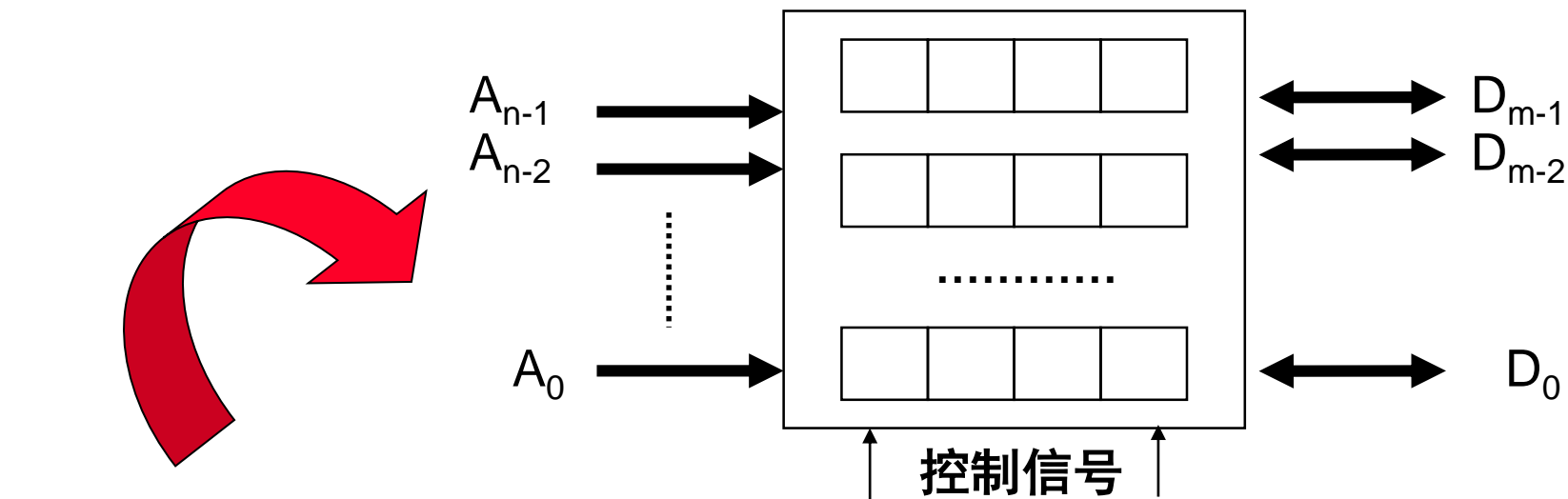
- 存储单元电路的概念及其基本条件（0/1状态、读出、写入）
- SRAM存储单元电路的工作原理（六管、触发器）
- DRAM存储单元电路的工作原理（单管、电容充放电、需要刷新）
- ROM存储单元电路的工作原理（ROM、PROM、EPROM、EEPROM，晶体管、MOS管开关特性（非易失），熔丝、浮空栅极来编程写入）

上一讲简要回顾(续)

❖ 主存储器 —— 存储芯片的内部结构

- 芯片容量的基本描述 (字单元数 \times 每个字单元的位数, $2^n \times m$)
- 一维地址结构 (矩阵 $2^n \times m$, 2^n 选择线, m 数据线)
- 二维地址结构 (矩阵 $2^{n/2} \times (2^{n/2} \times m)$, $2^{n/2} + 2^{n/2}$ 选择线, m 数据线)
- 存储芯片的片选信号/ \overline{CS} (DRAM \overline{RAS}) 和读写控制信号/ \overline{WE}
- DRAM存储芯片的地址管脚复用 (\overline{RAS} 和 \overline{CAS} 控制)
- DRAM的刷新计数器 (Refresher Counter, 生成行地址, 按行刷新)

存储单元电路 → 存储芯片



可存储 $2^n \times m$ 位的存储芯片

可存储1位(1bit)的存储单元电路

存储单元电路 → 存储芯片 → 存储器?

第三部分：主存储器

- 一. 存储系统概述
- 二. 存储单元电路
- 三. 存储器芯片结构
- 四. 存储器扩展
- 五. DRAM的刷新

存储器芯片的扩展

❖ 单个存储器芯片不能满足存储系统的容量需求

- 位空间不够
- 字空间（地址空间）不够
- 位空间和字空间都不够

❖ 存储扩展

- 位扩展
- 字扩展
- 混合扩展

4.1 存储器芯片的扩展 —— 位扩展

位扩展：（ $2^n \times m$ ）

- 存储器芯片提供的字空间，满足整个存储空间的字空间（地址空间）要求
- 但存储器芯片的位空间不能满足要求

❖ 原因

- 芯片中存储字单元的数量够，即 2^n 够
- 存储字单元的位数不够，即 m 不够

❖ 方法

- 多个存储器芯片的数据位空间拼在一起

4.1 存储器芯片的扩展 —— 位扩展

例：1Kx4 SRAM芯片构成1Kx8的存储器

➤ 1K×4 芯片管脚：

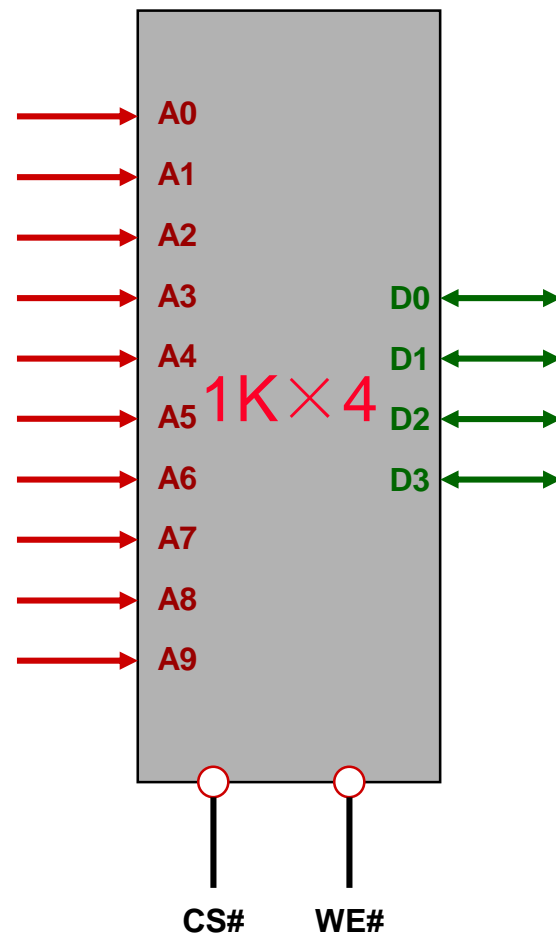
- 10个地址管脚 **A9~A0**
- 4个数据管脚 **D3~D0**
- 1个片选输入管脚 **CS#**
- 1个读写控制管脚 **WE#**
- 芯片地址空间：**000 H~3FF H**

➤ CPU访问1Kx8的存储器需提供：

- 地址总线10根：**AB9~AB0**
- 数据总线8根：**DB7~DB0**
- 读写控制信号：**MemW**
- 存储器地址空间：**000 H~3FF H**

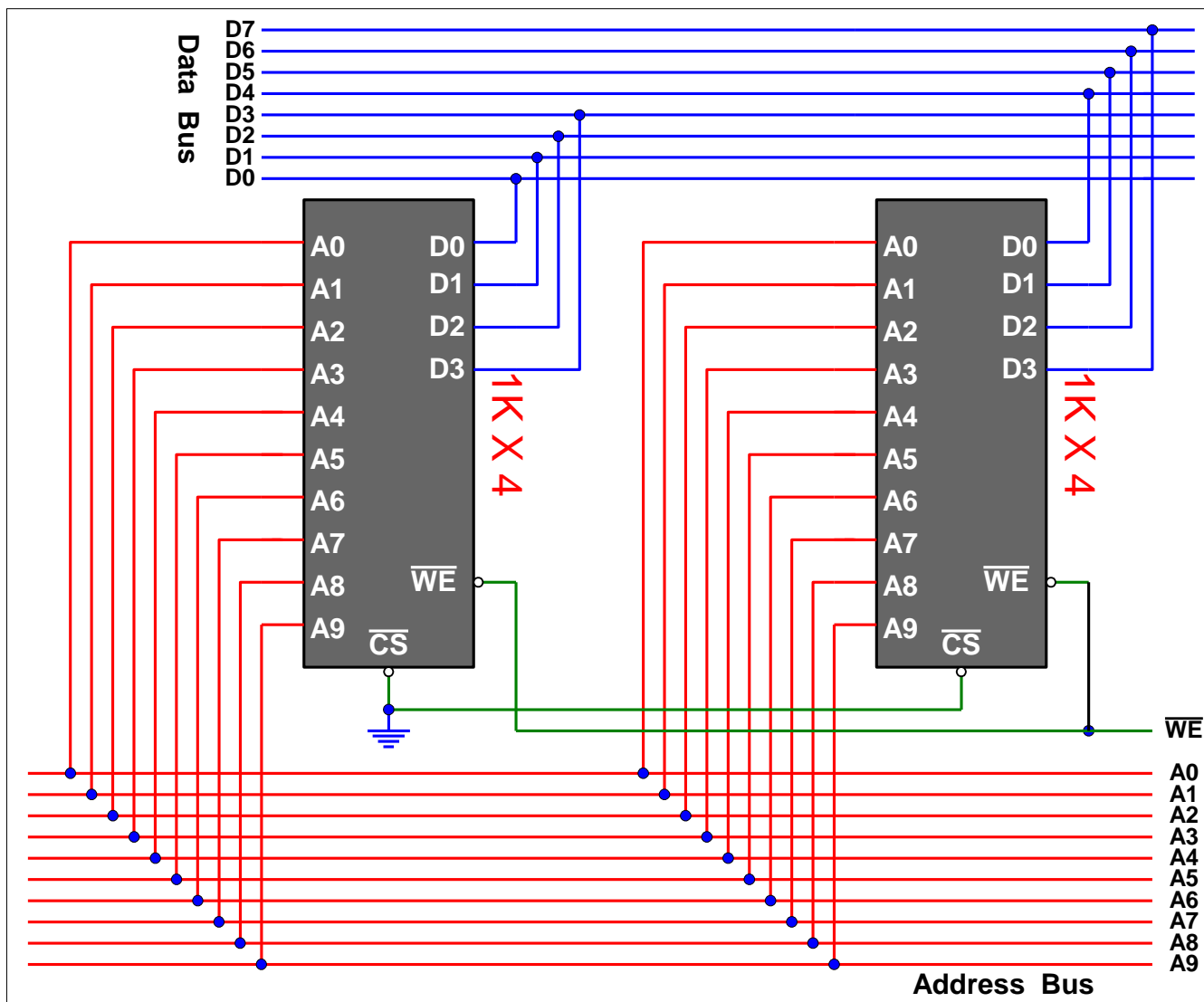
➤ 需要芯片： $(1K \times 8) / (1K \times 4) = 2$ 片

- 芯片地址管脚：都连接到**AB9~AB0**
- 芯片数据管脚：2个芯片的数据管脚，分别连到总线**DB7~DB4**、**DB3~DB0**
- 芯片读写控制管脚：连接**MemW**



4.1 存储器芯片的扩展 —— 位扩展

例：1K × 4 的SRAM存储芯片构建 1K × 8 的存储器



4.2 存储器芯片的扩展 —— 字扩展

字扩展： ($2^n \times m$)

- 存储器芯片提供的字空间，不能满足整个存储空间的字空间要求
- 但存储器芯片的位空间满足要求

❖ 原因

- 存储器芯片存储字数量不够，即 2^n 不够
- 存储字单元的数据位数够，即 m 够

❖ 方法

- 多个存储器芯片的字空间（地址空间）拼在一起

4.2 存储器芯片的扩展 —— 字扩展

例：1Kx8 SRAM芯片构成 4Kx8 的存储器

➤ 1K×8 芯片管脚：

- 10个地址管脚 **A9~A0**
- 8个数据管脚 **D7~D0**
- 1个片选输入管脚 **CS#**
- 1个读写控制管脚 **WE#**
- 芯片地址空间：**000 H ~ 3FF H**

➤ CPU访问 4Kx8的存储器需提供：

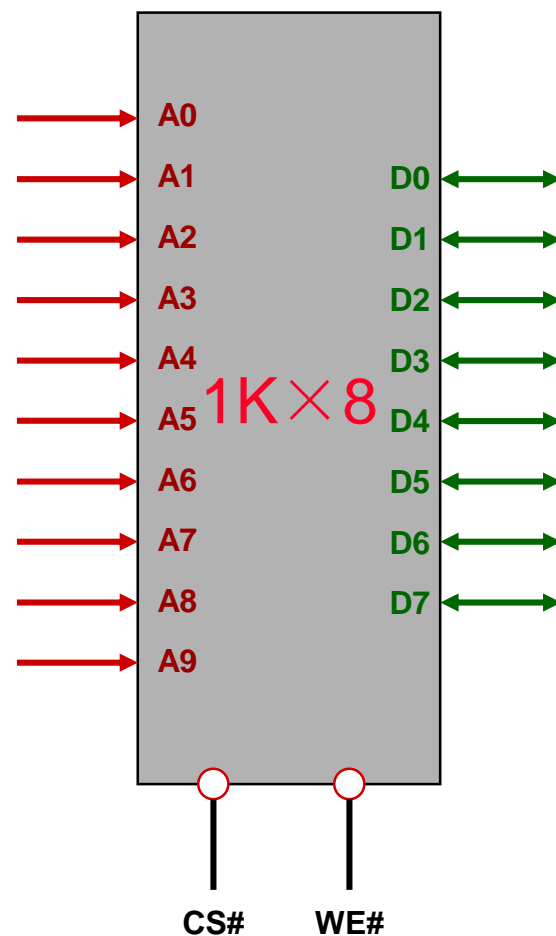
- 地址总线12根：**AB11~AB0**
- 数据总线8根：**DB7~DB0**
- 读写控制信号：**MemW**
- 存储器地址空间：**000 H ~ FFF H**

➤ 需要芯片数： $(4K \times 8) / (1K \times 8) = 4$ 片

- 芯片地址管脚：都连接到地址总线 **AB9~AB0**
- 芯片数据管脚：都连接到数据总线 **DB7~DB0**
- 芯片读写控制管脚：连接**MemW**

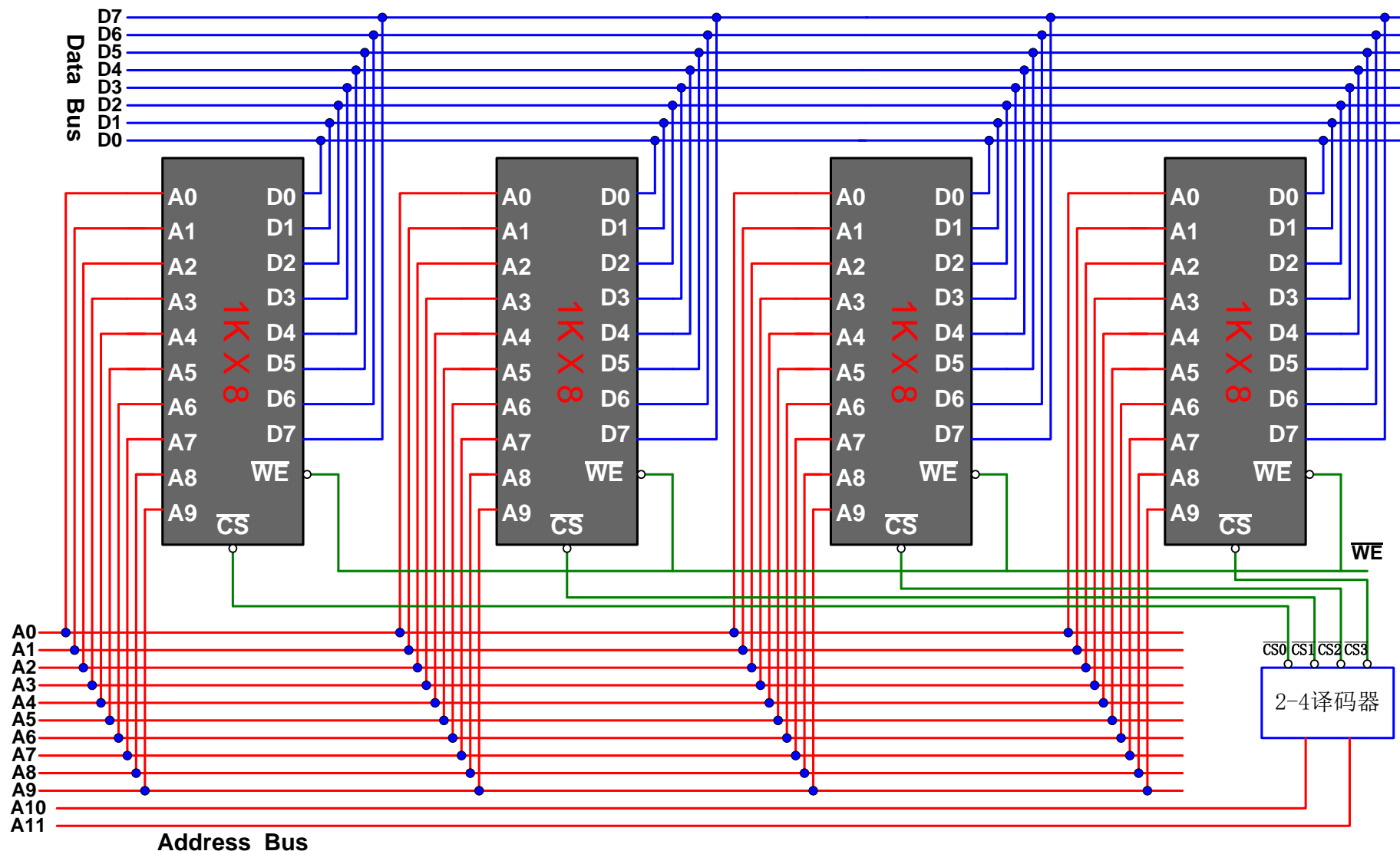
➤ 一个2-4译码器产生4个片选信号

- 译码器输入：**AB11~AB10**
- 译码器输出：分别接4个芯片片选管脚



4.2 存储器芯片的扩展 —— 字扩展

例：1Kx8 SRAM存储芯片构成4Kx8的存储器



4.3 存储器芯片的扩展 ——混合扩展

混合扩展： ($2^n \times m$)

- 存储器芯片提供的字空间不能满足整个存储空间的字空间要求
- 位空间也不能满足要求

❖ 原因

- 存储器芯片存储字数量不够，即 2^n 不够
- 存储字单元的数据位数不够，即 m 不够

❖ 方法

- 综合运用字扩展和位扩展

4.3 存储器芯片的扩展 —— 混合扩展

例：4Kx4 SRAM存储芯片构成16Kx8的存储器

➤ 4K×4芯片：

- 12个地址管脚 **A11~A0**
- 4个数据管脚 **D3~D0**
- 1个片选输入管脚 **CS#**
- 1个读写控制管脚 **WE#**
- 芯片地址空间：**000 H ~ FFF H**

➤ CPU访问16Kx8的存储器需提供：

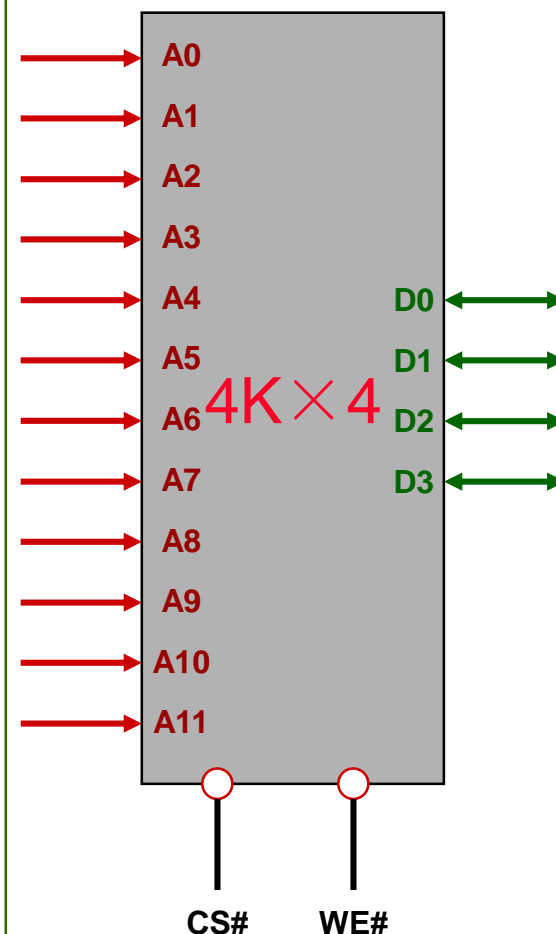
- 地址总线14根：**AB13~AB0**
- 数据总线 8 根：**DB7~DB0**
- 读写控制信号：**MemW**
- 存储器地址空间：**0000 H ~ 3FFF H**

➤ 需要芯片数： $(16K \times 8) / (4K \times 4) = 4 \times 2 = 8$ 片

- 分4组（字扩展），每组2个芯片（位扩展）

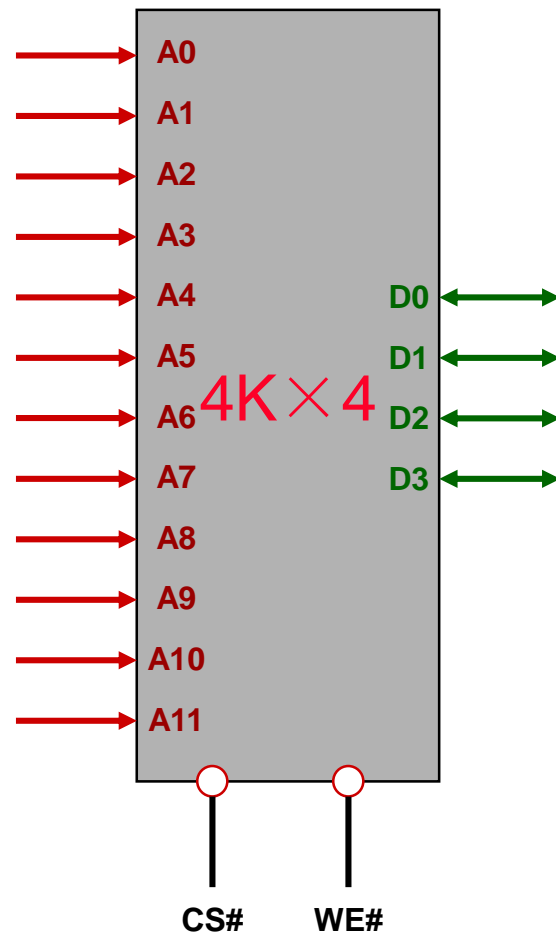
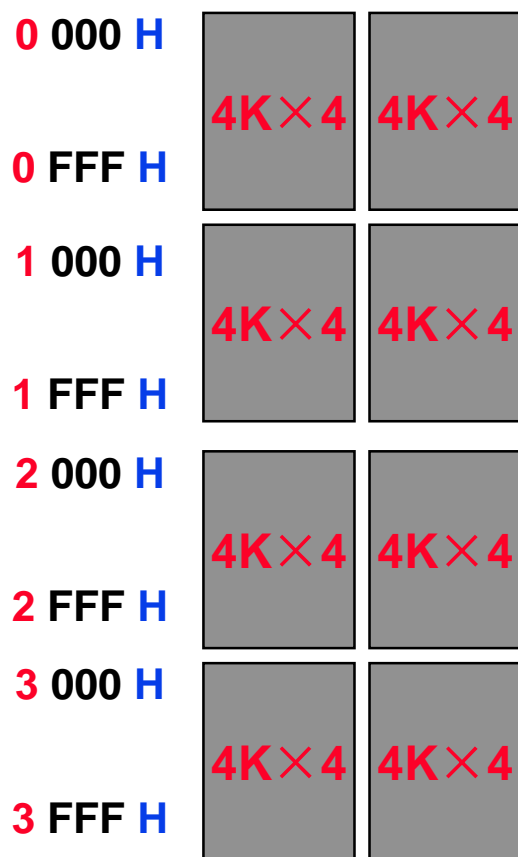
➤ 一个2-4译码器产生4个片选信号

- 译码器输入：**AB13~AB12**
- 译码器输出：分别接4组芯片片选管脚



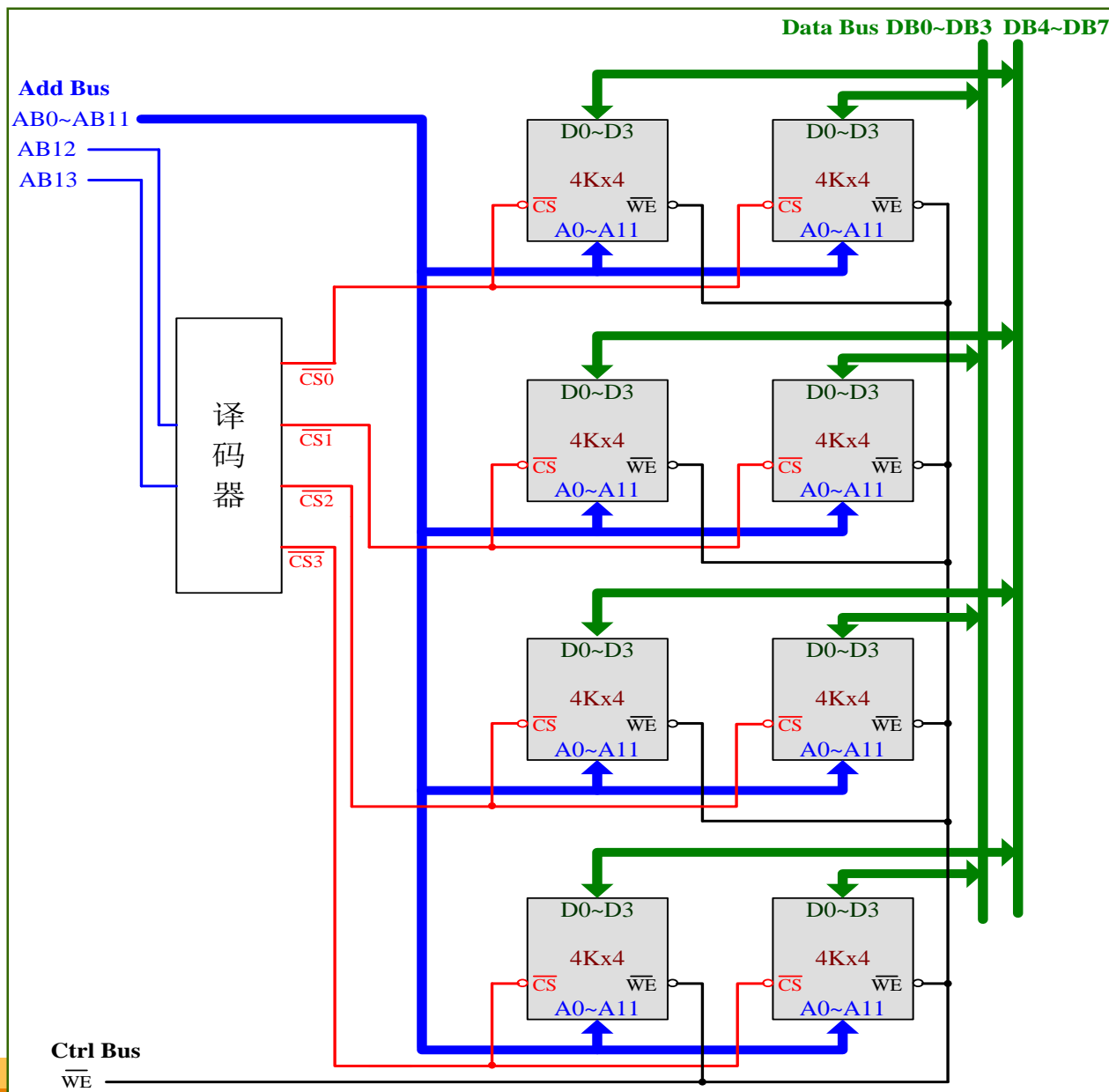
4.3 存储器芯片的扩展 —— 混合扩展

4Kx4 SRAM存储芯片构成16Kx8的存储器地址空间划分

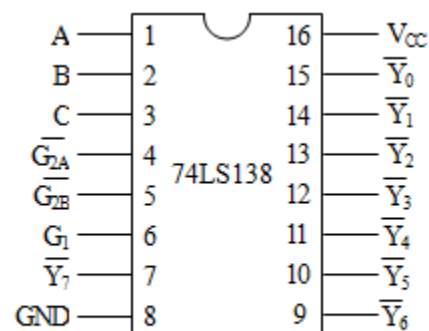
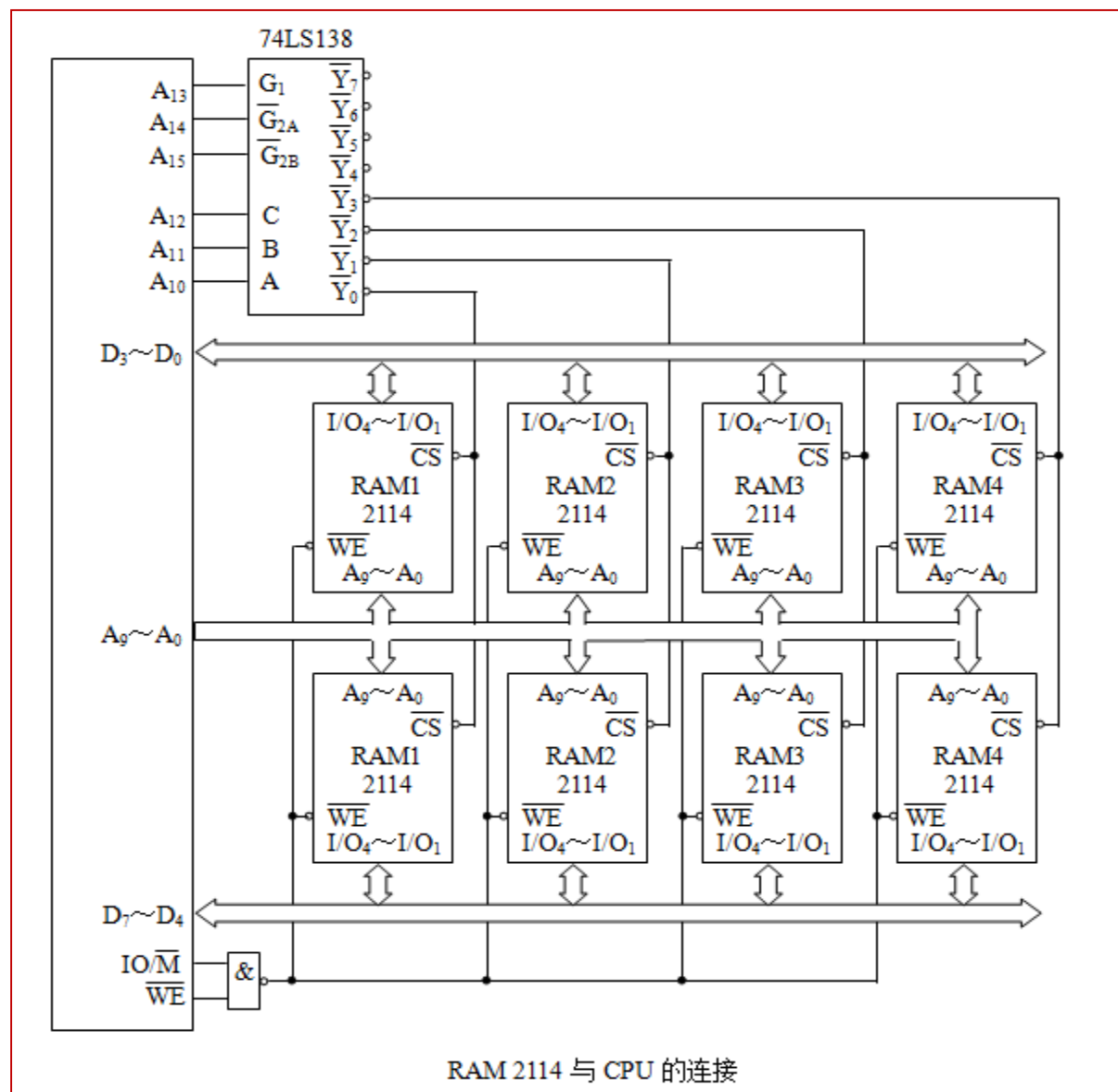


4.3 存储器芯片的扩展 —— 混合扩展

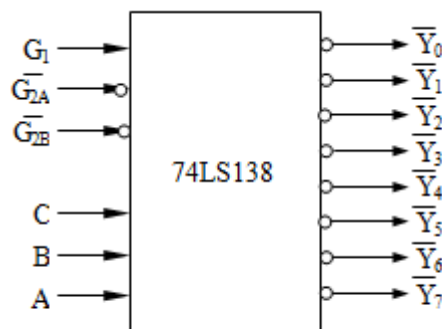
4Kx4 SRAM存储芯片构成
16Kx8的存储器连接图



4.4 存储器芯片的扩展示例 —— CPU与主存的连接



74LS138 引脚图



74LS138 逻辑符号

4.4 存储器芯片的扩展示例 —— 扩展方法小结

➤ 基本思路

1. 确定每个芯片的地址管脚数、数据管脚数。
2. 确定整个存储空间所需的地址总线和数据总线的数量。
3. 计算所需芯片的数量，确定每个芯片在整个存储空间中的地址空间范围、位空间范围。
4. 所有芯片的地址管脚全部连接到地址总线对应的地址线上。
5. 同一字空间的芯片CS信号连在一起。
6. 同一位空间的芯片数据线连在一起，并连接到对应的数据总线上。
7. 根据每个芯片的地址空间范围，设计芯片所需片选信号CS的逻辑电路，电路的输入一定是地址总线中，没有连接到芯片地址管脚上的那部分地址线。
8. 统一读写控制。

4.4 存储器芯片的扩展示例 —— 异种芯片

❖ 用3片16K×4的SRAM芯片和若干8K×4的SRAM芯片，组成一个64K×8的按字节编址的存储器。

1. 确定每个芯片的地址管脚数、数据管脚数

➤ 16K×4芯片： 14位地址，4位数据，芯片地址空间：0000 H~3FFF H

➤ 8K×4芯片： 13位地址，4位数据，芯片地址空间：0000 H~1FFF H

2. 确定整个存储空间所需的地址总线和数据总线的数量

➤ 64K×8存储器： 16位地址，8位数据，地址空间：0000 H~FFFF H

3a. 计算所需存储器芯片的数量

➤ 16K×4芯片： 已有3片

➤ 需要8K×4芯片数： $[(64K \times 8) - 3 \times (16K \times 4)] / (8K \times 4) = 10$ 片

4.4 存储器芯片的扩展示例 —— 异种芯片

3b. 确定每个存储器芯片在整个存储空间中的地址空间范围、位空间范围

A _{15~0}	A _{15~12}	A _{11~8}	A _{7~4}	A _{3~0}	D _{7~4}	D _{3~0}
0000H	0000	0000	0000	0000	16K x 4	16K x 4
3FFFH	0011	1111	1111	1111	16K x 4	8K x 4
4000H	0100	0000	0000	0000	16K x 4	8K x 4
5FFFH	0101	1111	1111	1111		
6000H	0110	0000	0000	0000		
7FFFH	0111	1111	1111	1111		
8000H	1000	0000	0000	0000	8K x 4	8K x 4
9FFFH	1001	1111	1111	1111		
A000H	1010	0000	0000	0000	8K x 4	8K x 4
BFFFH	1011	1111	1111	1111		
C000H	1100	0000	0000	0000	8K x 4	8K x 4
DFFFH	1101	1111	1111	1111		
E000H	1110	0000	0000	0000	8K x 4	8K x 4
FFFFH	1111	1111	1111	1111		

➤ 共有:

3片16Kx4芯片

10片8Kx4芯片

➤ 如将10片8Kx4

看作5片16Kx4

则相当于共有:

8片16Kx4芯片

➤ 存储器为64Kx8

可组成4X2方阵

4.4 存储器芯片的扩展示例 —— 异种芯片

4. 所有芯片的地址管脚全部连接到地址总线对应的地址线上

$A_{15\sim0}$	$A_{15\sim12}$	$A_{11\sim8}$	$A_{7\sim4}$	$A_{3\sim0}$		
0000H	0000	0000	0000	0000	16K x 4	16K x 4
3FFFH	0011	1111	1111	1111	16K x 4	8K x 4
4000H	0100	0000	0000	0000	16K x 4	8K x 4
5FFFH	0101	1111	1111	1111		
6000H	0110	0000	0000	0000		
7FFFH	0111	1111	1111	1111		
8000H	1000	0000	0000	0000	8K x 4	8K x 4
9FFFH	1001	1111	1111	1111		
A000H	1010	0000	0000	0000	8K x 4	8K x 4
BFFFH	1011	1111	1111	1111		
C000H	1100	0000	0000	0000	8K x 4	8K x 4
DFFFH	1101	1111	1111	1111		
E000H	1110	0000	0000	0000	8K x 4	8K x 4
FFFFH	1111	1111	1111	1111		

➤ 对16Kx4芯片，
连接A13~A0

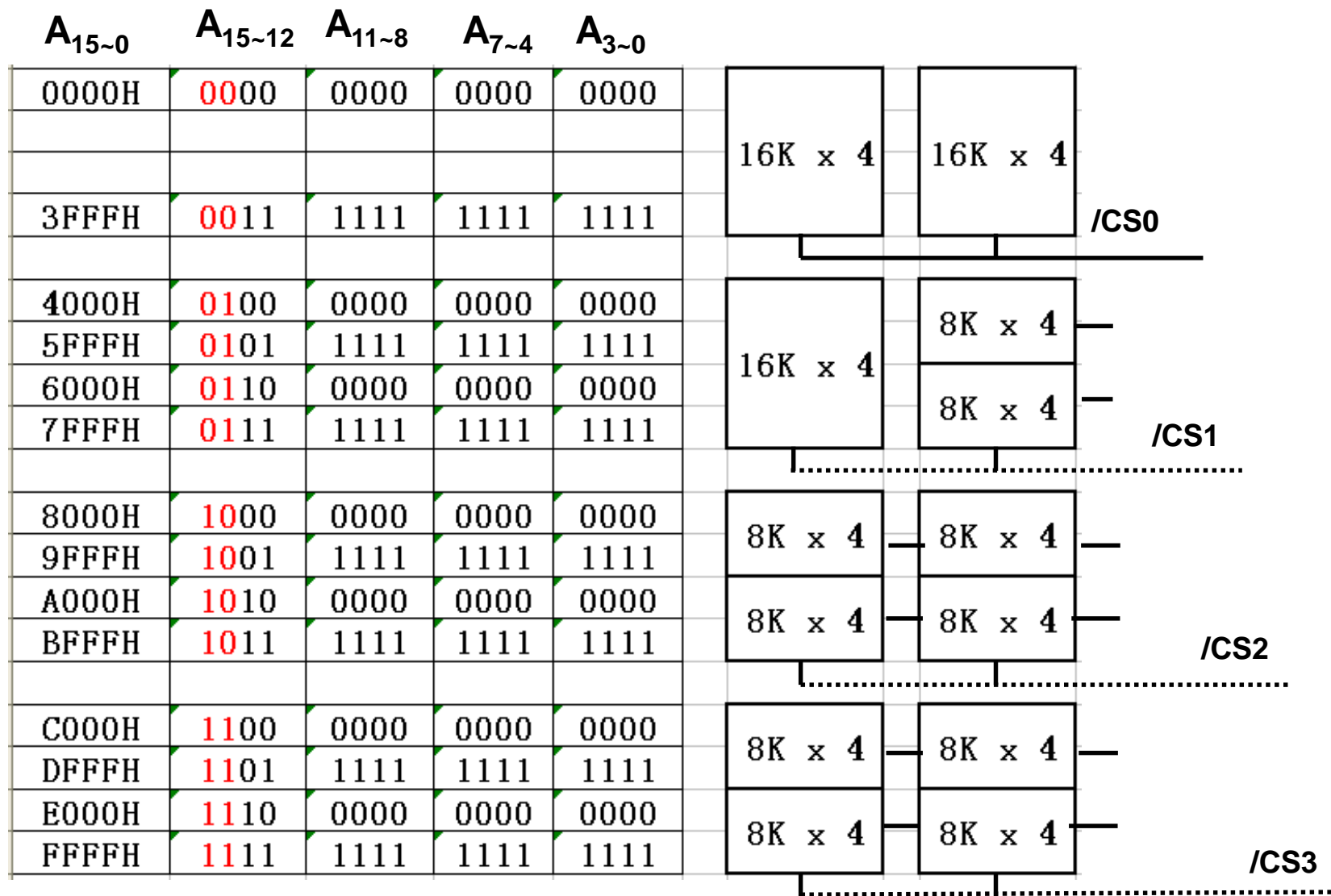
➤ 剩余A15, 14
用于CS的生成

➤ 对8Kx4芯片，
连接A12~A0

➤ 剩余A15,14,13
用于CS的生成

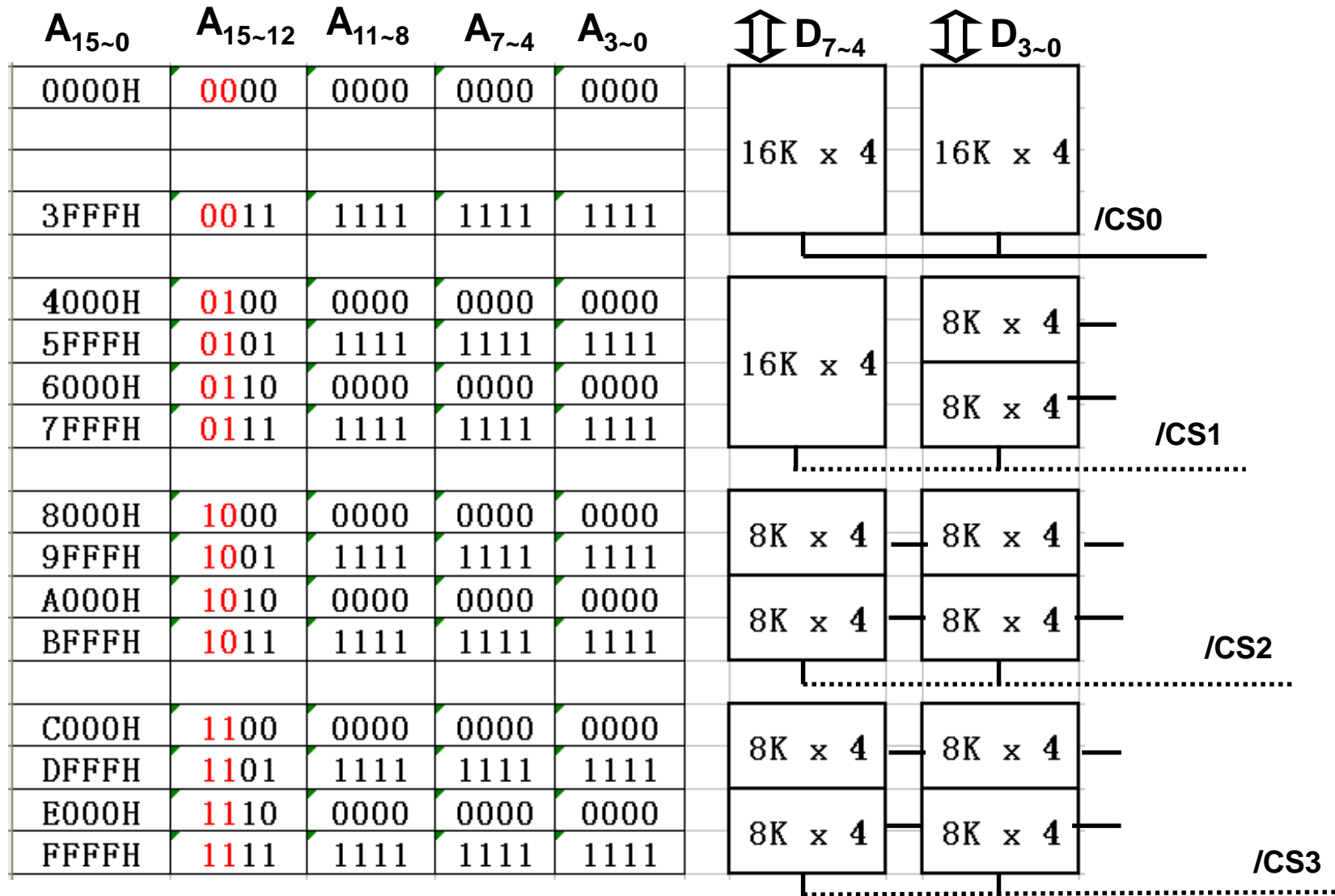
4.4 存储器芯片的扩展示例 —— 异种芯片

5. 同一字空间的存储芯片CS信号连在一起



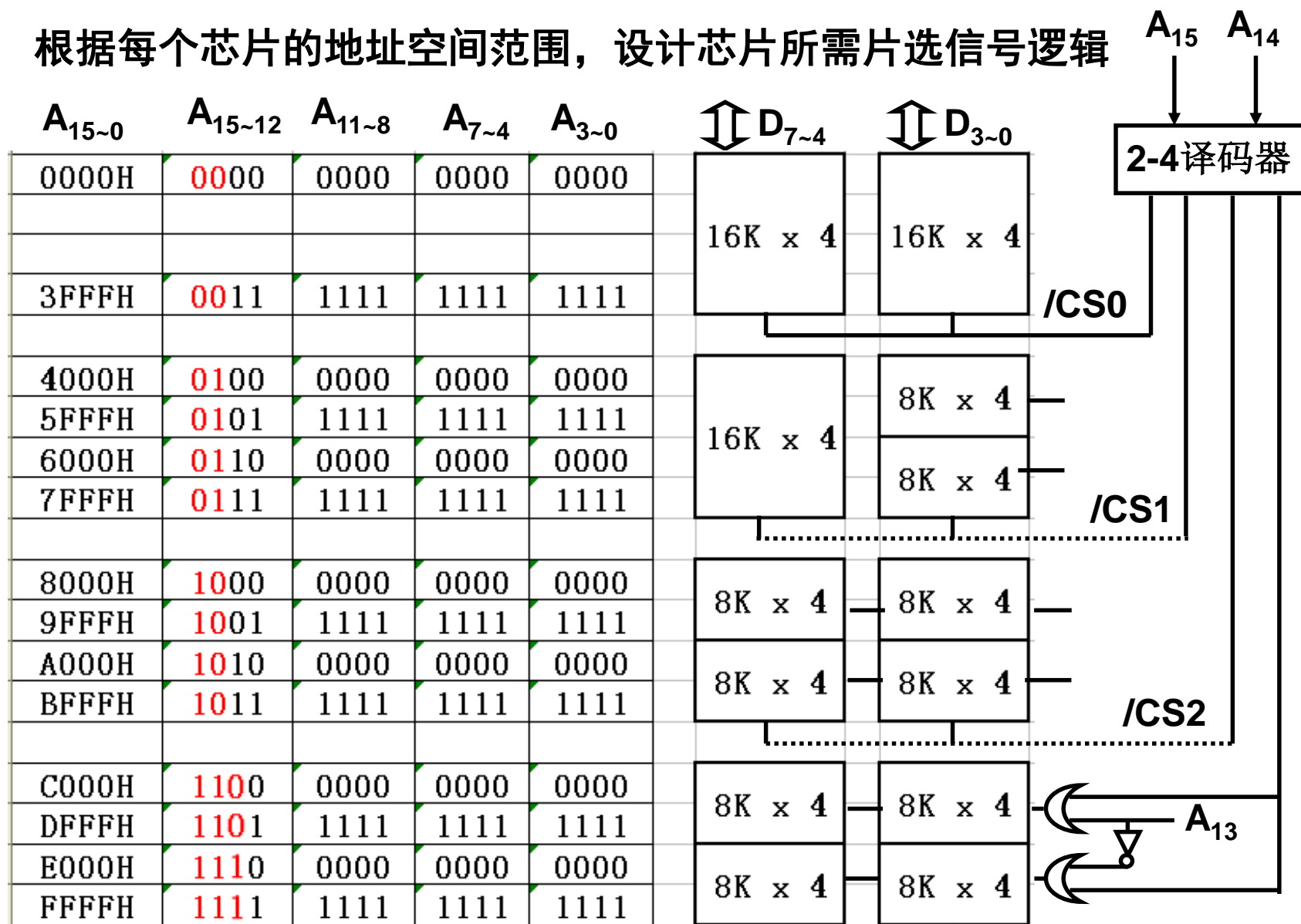
4.4 存储器芯片的扩展示例 —— 异种芯片

6. 同一位空间的数据线连在一起，并连接到对应的数据总线上



4.4 存储器芯片的扩展示例 —— 异种芯片

7. 根据每个芯片的地址空间范围，设计芯片所需片选信号逻辑



4.4 存储器芯片的扩展示例 —— 异种芯片

8. 统一读写控制

$A_{15\sim0}$	$A_{15\sim12}$	$A_{11\sim8}$	$A_{7\sim4}$	$A_{3\sim0}$		
0000H	0000	0000	0000	0000	16K x 4	16K x 4
3FFFH	0011	1111	1111	1111	16K x 4	8K x 4
4000H	0100	0000	0000	0000	16K x 4	8K x 4
5FFFH	0101	1111	1111	1111		
6000H	0110	0000	0000	0000		
7FFFH	0111	1111	1111	1111		
8000H	1000	0000	0000	0000	8K x 4	8K x 4
9FFFH	1001	1111	1111	1111		
A000H	1010	0000	0000	0000	8K x 4	8K x 4
BFFFH	1011	1111	1111	1111		
C000H	1100	0000	0000	0000	8K x 4	8K x 4
DFFFH	1101	1111	1111	1111		
E000H	1110	0000	0000	0000	8K x 4	8K x 4
FFFFH	1111	1111	1111	1111		

➤ 所有芯片
的读写控
制信号连
接到总线
读写控制
信号上

4.4 存储器芯片的扩展示例 —— 异种芯片

CPU地址线A15~A0，数据线D7~D0， \overline{WR} 为读/写信号， \overline{MREQ} 为访存请求信号。0000H~3FFFH为系统程序区，4000H~FFFFH为用户程序区。用8K×4位ROM芯片和16K×8位RAM芯片构成该存储器，要求说明地址译码方案，并将ROM芯片、RAM芯片与CPU连接。

解：因为0000H~3FFFH为系统程序区，ROM区高两位总是00，低14位为全译码。

ROM区大小为： $2^{14} \times 8\text{位} = 16\text{K} \times 8\text{位} = 16\text{KB}$

ROM芯片数为： $16\text{K} \times 8\text{位} / 8\text{K} \times 4\text{位} = 2 \times 2 = 8$ ，字方向扩展2倍，位方向扩展2倍

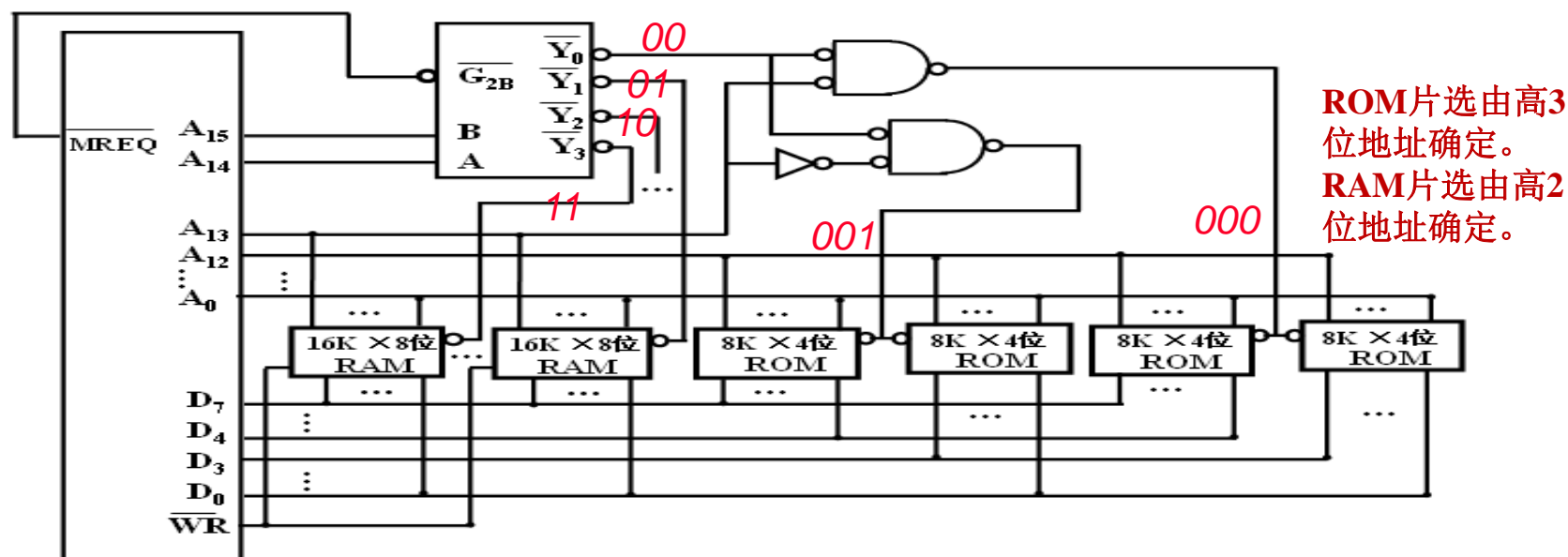
ROM芯片内地址位数为13位，连到CPU低13位地址线A12~A0

因为4000H~FFFFH为用户程序区，RAM区高两位是01、10、11，低14位为全译码。

RAM区大小为： $3 \times 2^{14} \times 8\text{位} = 3 \times 16\text{K} \times 8\text{位} = 48\text{KB}$

RAM芯片数为： $48\text{K} \times 8\text{位} / 16\text{K} \times 8\text{位} = 3 \times 1 = 3$ ，字方向上扩展3倍，位方向上不扩展。

RAM芯片内地址位数为14位，连到CPU低14位地址线A13~A0。





4.5 存储器的符号表示

❖ 读操作

➤ 输入

- 读单元地址: **Address**
- 读控制信号: **MemRead**

➤ 输出

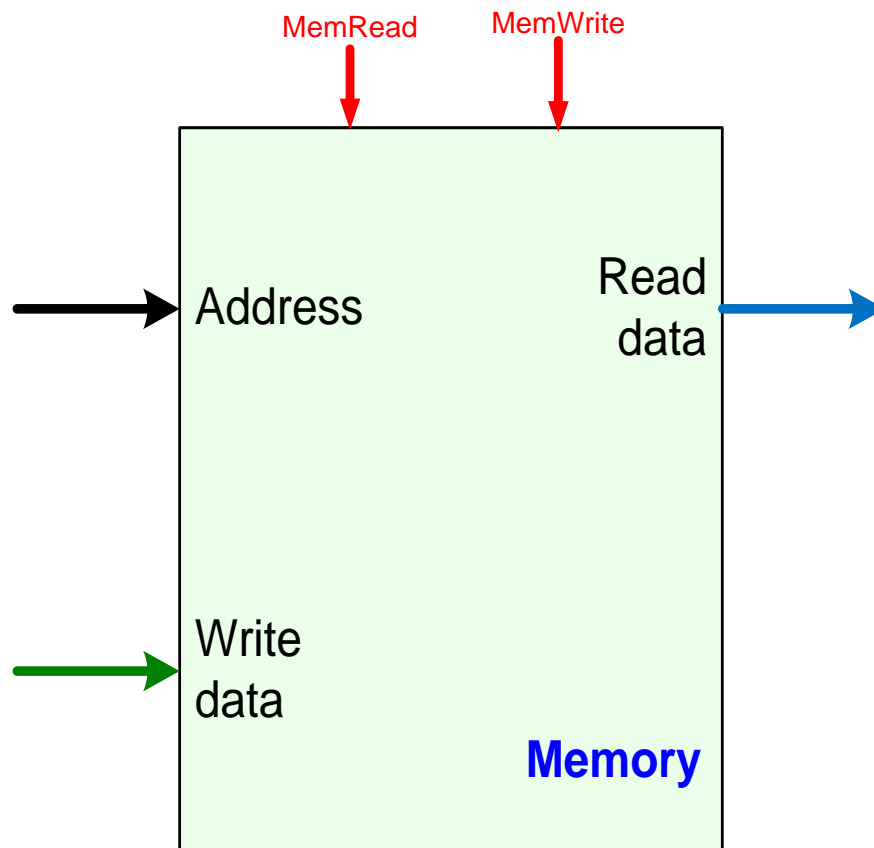
- 读出数据: **Readdata**

❖ 写操作

➤ 输入

- 写单元地址: **Address**
- 待写入数据: **Writedata**
- 写控制信号: **MemWrite**

➤ 输出: 无

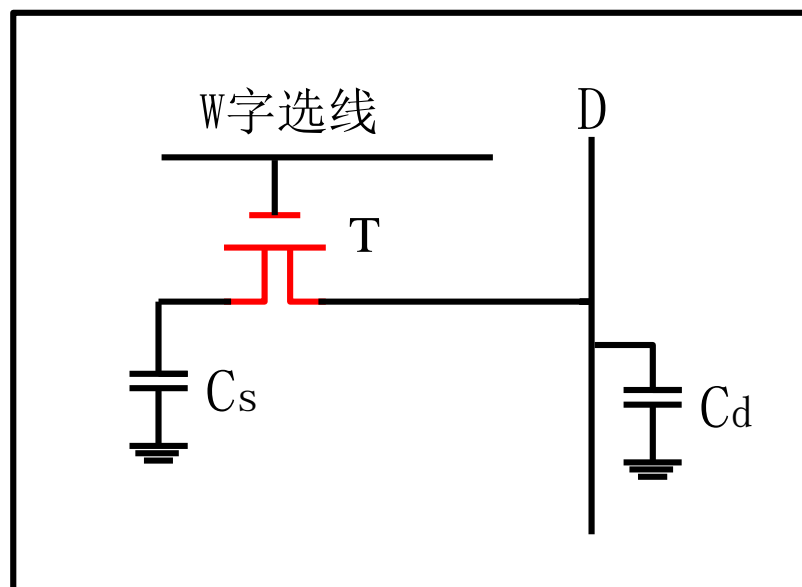


第三部分：主存储器

- 一. 存储系统概述
- 二. 存储单元电路
- 三. 存储器芯片结构
- 四. 存储器扩展
- 五. DRAM的刷新

5.1 DRAM存储单元电路的刷新

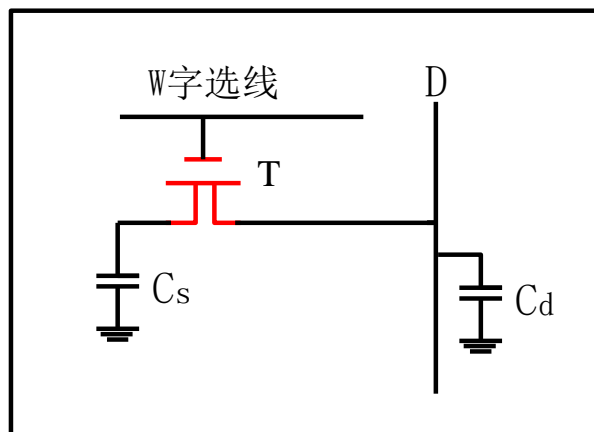
❖ DRAM存储单元电路的刷新问题



1. 在保持状态下，T管截止， C_s 与外部隔开，但 C_s 两极间存在漏电流，所以， C_s 上的电荷也会出现变化，必须在一个时间内重写数据，这个时间称为单元电路的刷新周期，一般为4ms、8ms。
2. 读出操作是一种破坏性操作，读1时， C_s 放电；读0时， C_s 充电；所以读出操作后，原保存在 C_s 上的数据（电荷）被破坏，应该立即进行恢复（重写或刷新）。

5.1 DRAM存储单元电路的刷新

❖ DRAM单管单元电路的工作特征



V_d' : D线在读出调整后的电压

V_{cs} : C_s 原来的电压

ΔV : D线上读出过程前后的变化量

$$\Delta V = V_d' - V_{pre} = (V_{cs} - V_{pre}) \times C_s / (C_s + C_d)$$

由于 C_d 要比 C_s 大一两个数量级, 所以

ΔV 不会太大(1%到10%), 一般为100mV左右。

D线上的电压在读出过程中的变化量实例计算:

假定 $C_s = 1\text{pf}$, $C_d = 50\text{pf}$, $V_{pre} = 2.5\text{V}$

存储1时, $V_{cs} = 3.5\text{V}$, 存储0时, $V_{cs} = 0\text{V}$

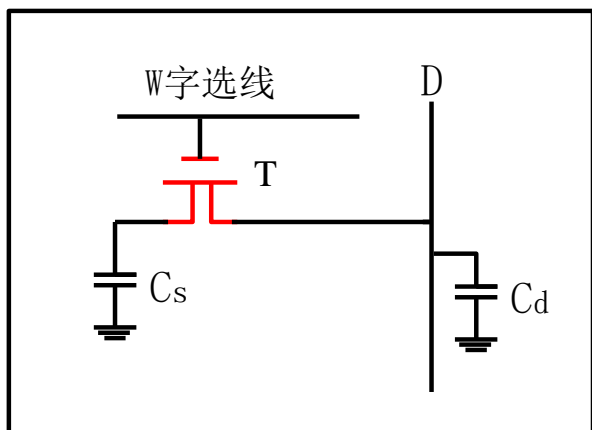
则:

$$\Delta V(1) = (3.5\text{v} - 2.5\text{v}) \times 1\text{pf} / (1\text{pf} + 50\text{pf}) = 19.6\text{mv}$$

$$\Delta V(0) = (0\text{v} - 2.5\text{v}) \times 1\text{pf} / (1\text{pf} + 50\text{pf}) = -49\text{mv}$$

5.1 DRAM存储单元电路的刷新

❖ DRAM存储单元电路的信号刷新问题



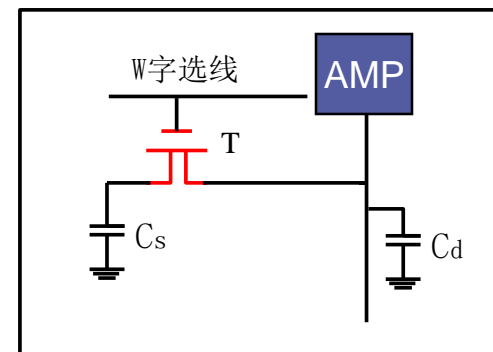
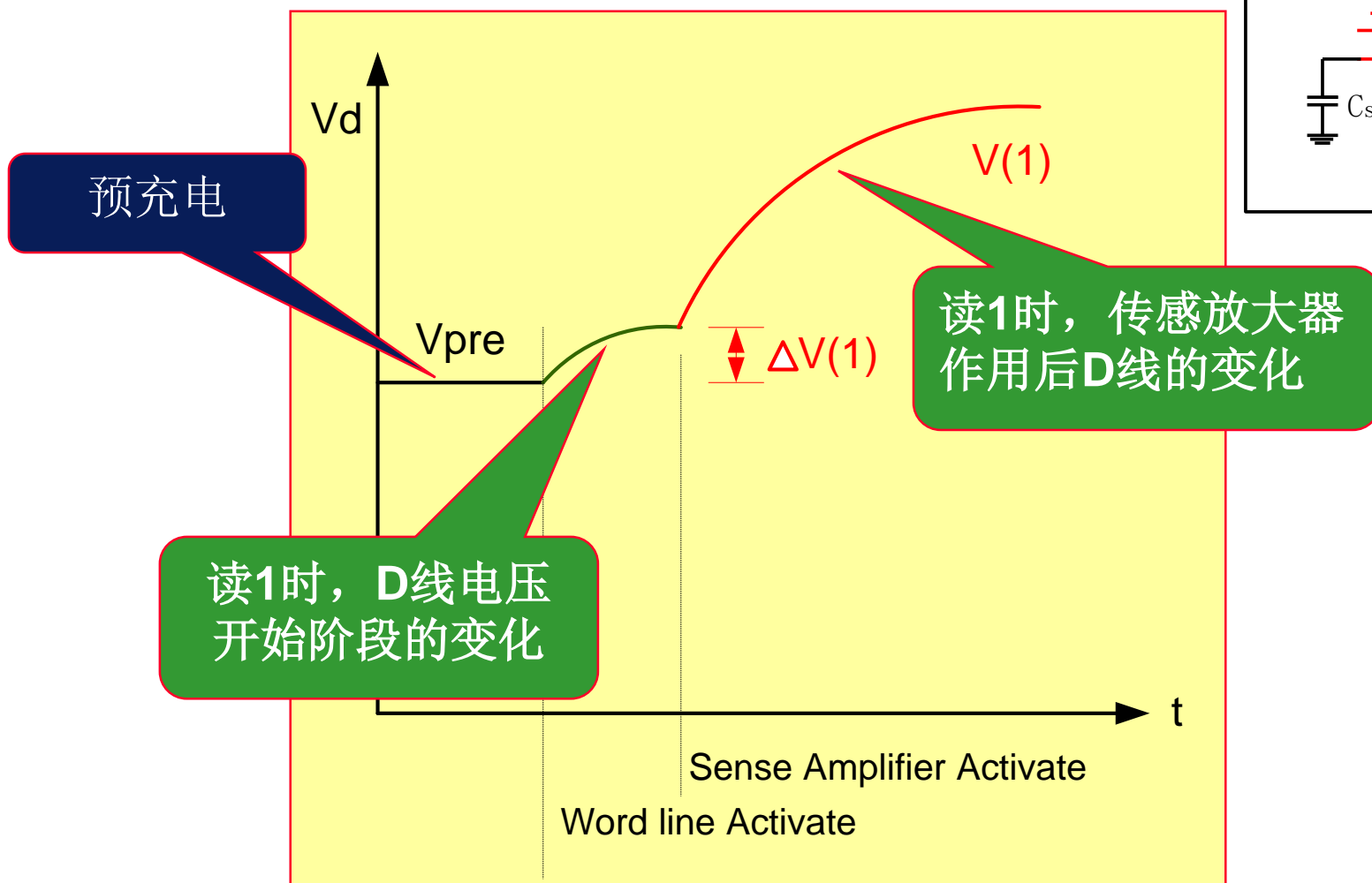
由于读出过程D线电压变化量较小，需要对变化量进行放大，才能得到有效的数据，因此，单管存储单元电路中，D线上必须增加传感放大器(Sense Amplifier)。

- 在D线上增加了传感放大器后，读过程实际上就是一次刷新过程。
- 刷新由传感放大器在读出过程中同时完成。
- 事实上，DRAM的刷新，就是通过这样的读操作来实现的。

5.1 DRAM存储单元电路的刷新

❖ DRAM单管单元电路

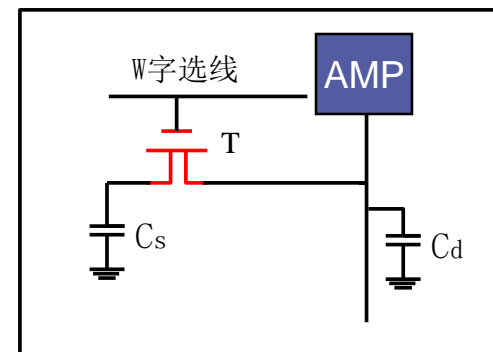
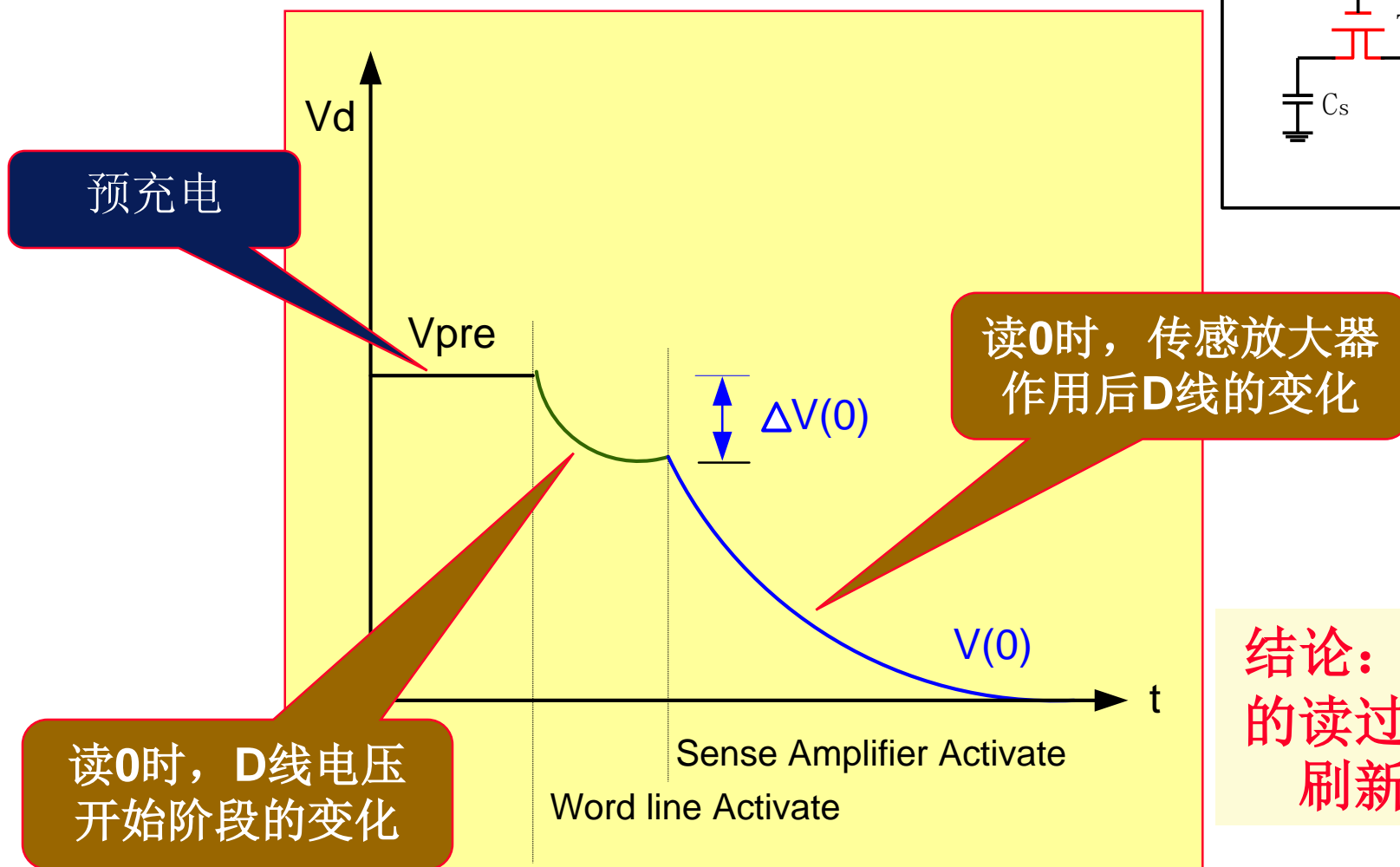
➤ D线上的电压在读“1”过程中的变化情况



5.1 DRAM存储单元电路的刷新

❖ DRAM单管单元电路

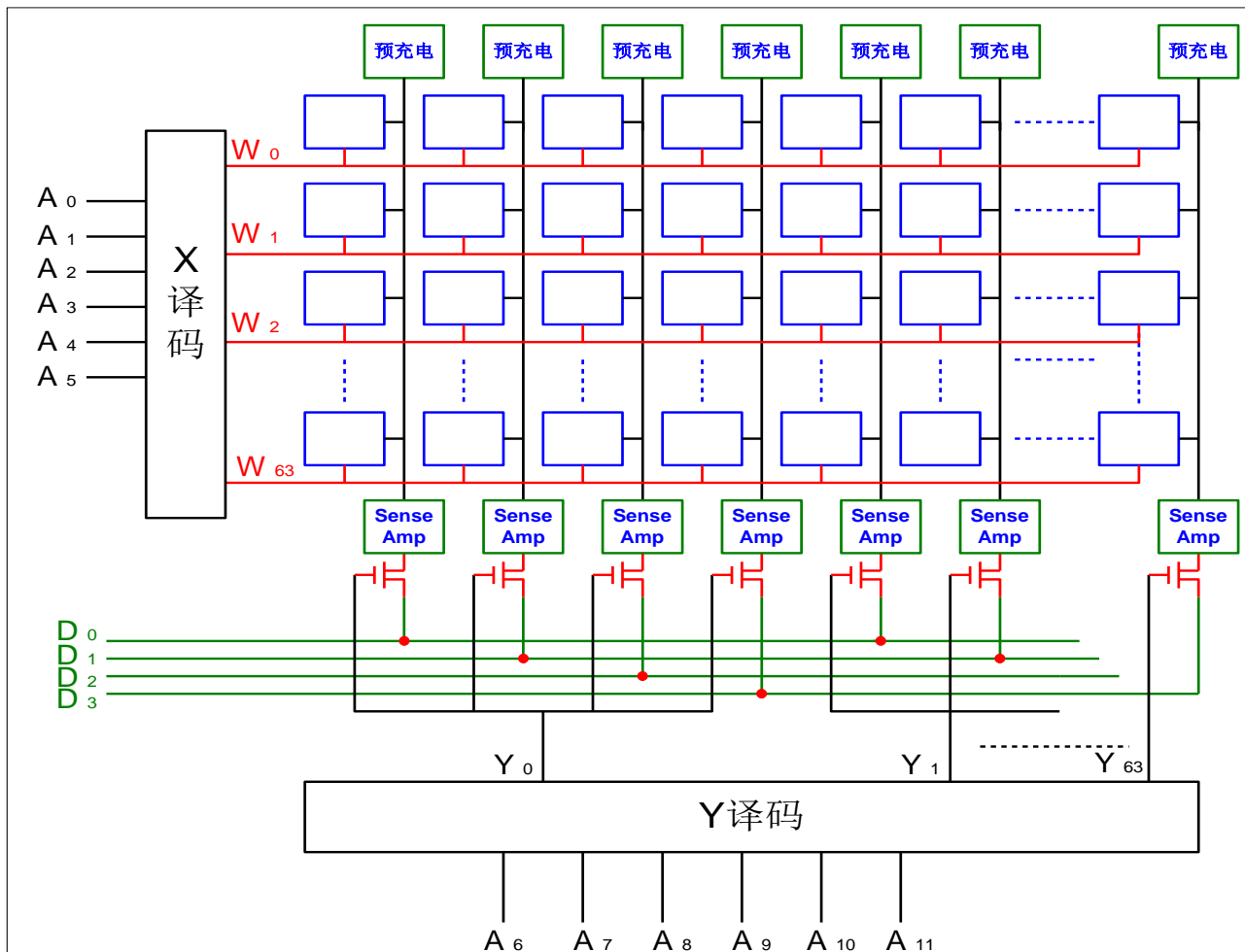
➤ D线上的电压在读“0”过程中的变化情况



**结论: DRAM
的读过程就是
刷新过程**

5.2 DRAM存储芯片的刷新

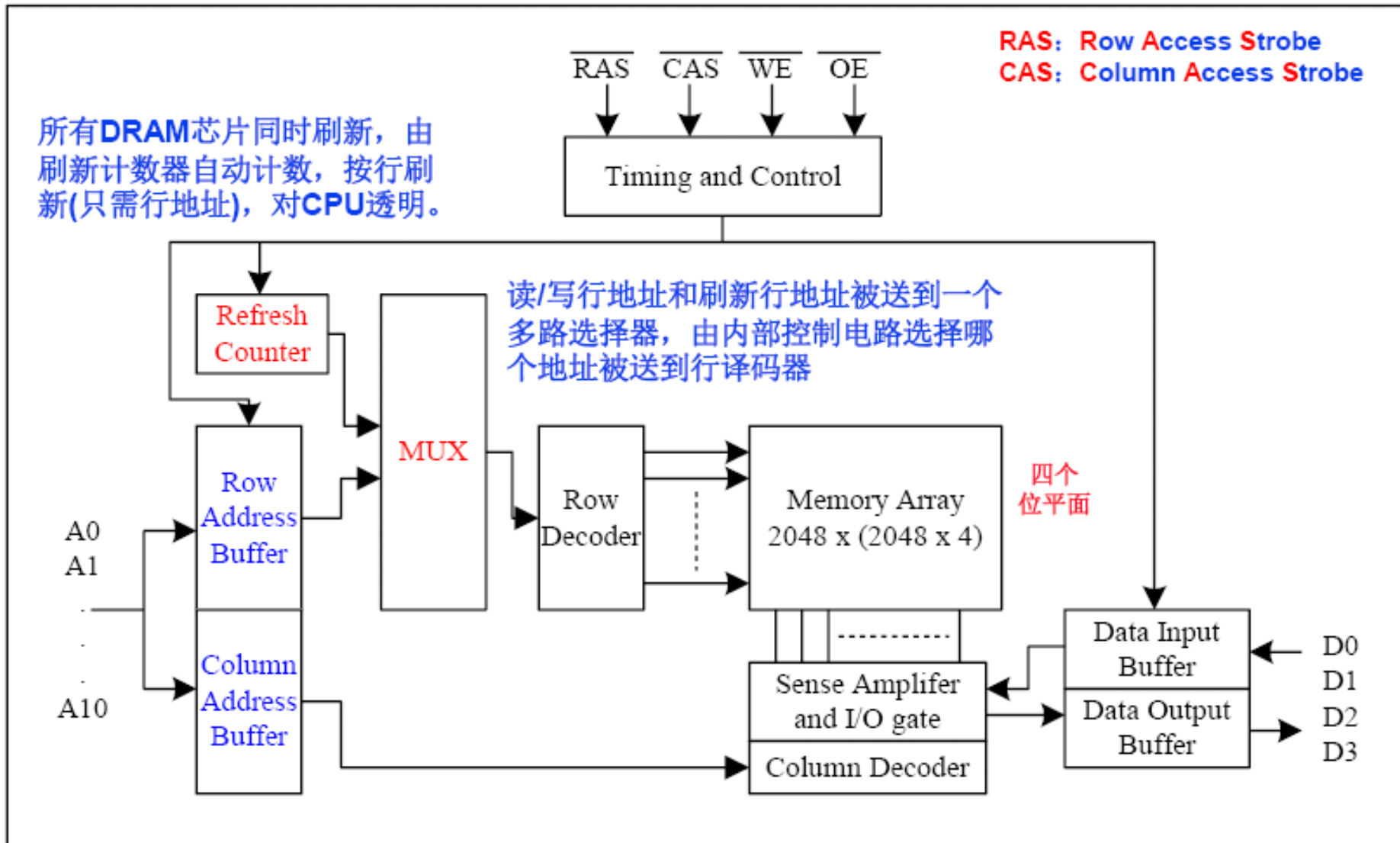
❖ 二维地址结构（4096*4 DRAM）



按行刷新，每次刷新1行！刷新地址？

存储芯片结构示例

❖ DRAM 4M*4 DRAM芯片结构 (内部包含刷新电路)



5.3 DRAM的刷新方式

❖ DRAM刷新的特点

- 刷新操作：读操作
- 按行刷新、所有芯片同时进行
- 刷新操作与CPU访问内存分开进行
- 刷新周期：2ms, 4ms, 8ms
- 刷新地址及刷新地址计数器

❖ DRAM常用的刷新方式

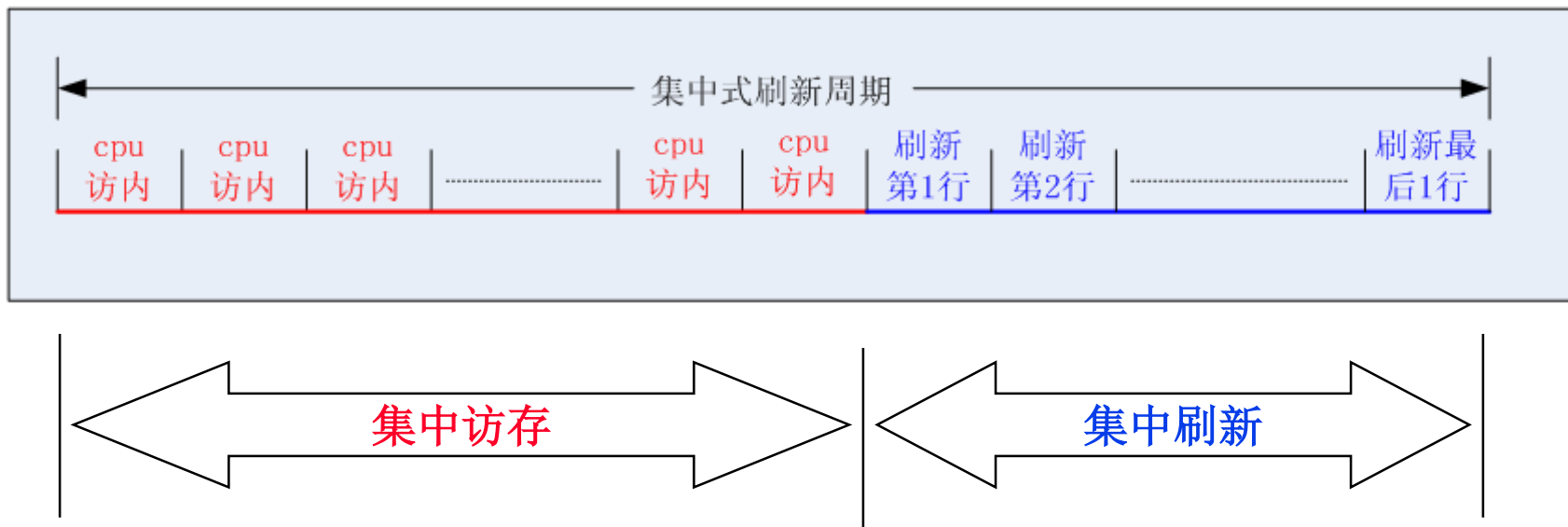
- 集中式
- 分散式
- 分布式

5.3 DRAM的刷新方式

❖ 集中刷新方式

将刷新周期分成两部分：在一个时间段内，刷新存储器所有行，此时CPU停止访问内存；另一个时间段内，CPU可以访问内存，刷新电路不工作。

集中刷新间隔 = 刷新周期



5.3 DRAM的刷新方式

例：假设刷新周期为2ms，读/写周期为 $0.5\mu\text{s}$ ，采用集中刷新方式，对 128×128 矩阵存储器刷新

集中刷新时间相当于128个读周期，用于读128行；

1个刷新周期中有4000个读/写周期，其中：

128个周期（ $64\mu\text{s}$ ）用来集中刷新操作，**不能访存**；

3872个周期（ $1936\mu\text{s}$ ）用于读/写或维持信息；

当3872个周期结束，便开始进行**128个周期（ $64\mu\text{s}$ ）**的刷新操作

存在不能进行读写操作的死区时间（例： $64\mu\text{s}$ ），很少使用

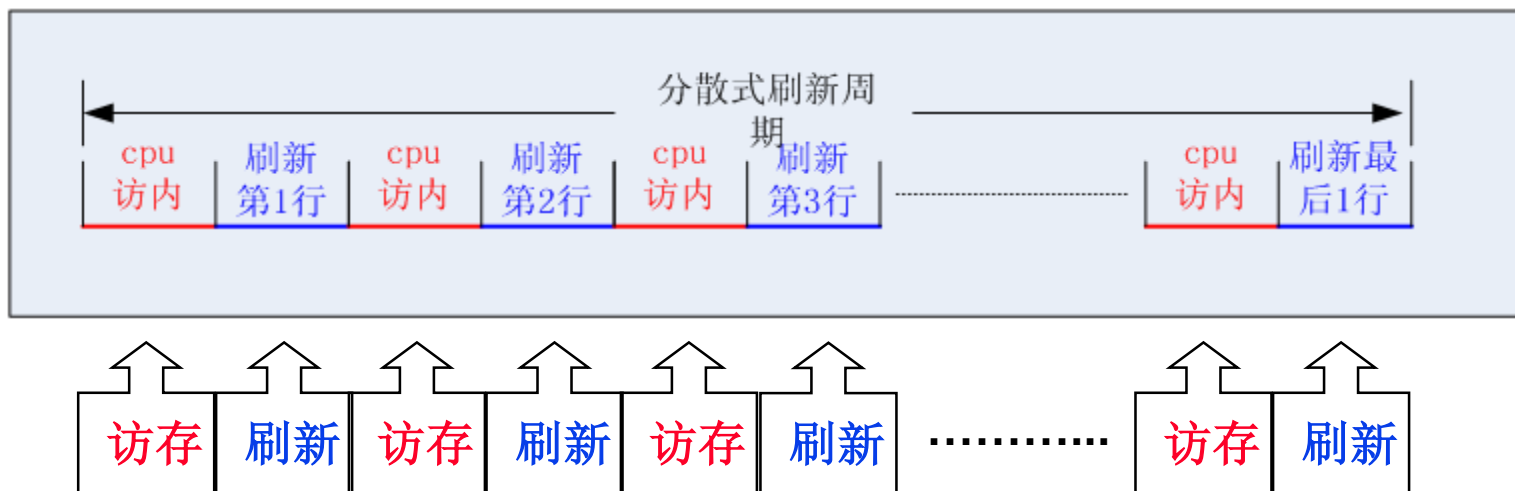
5.3 DRAM的刷新方式

❖ 分散刷新方式

CPU与刷新电路交替访问内存，一个存储周期刷新1行，下一个存储周期刷新另一行，直至最后1行后，又开始刷新第1行。

同1行两次被刷新的时间间隔，可能小于刷新周期。

分散刷新间隔 = 刷新行数 \times 存储周期 \leq 刷新周期



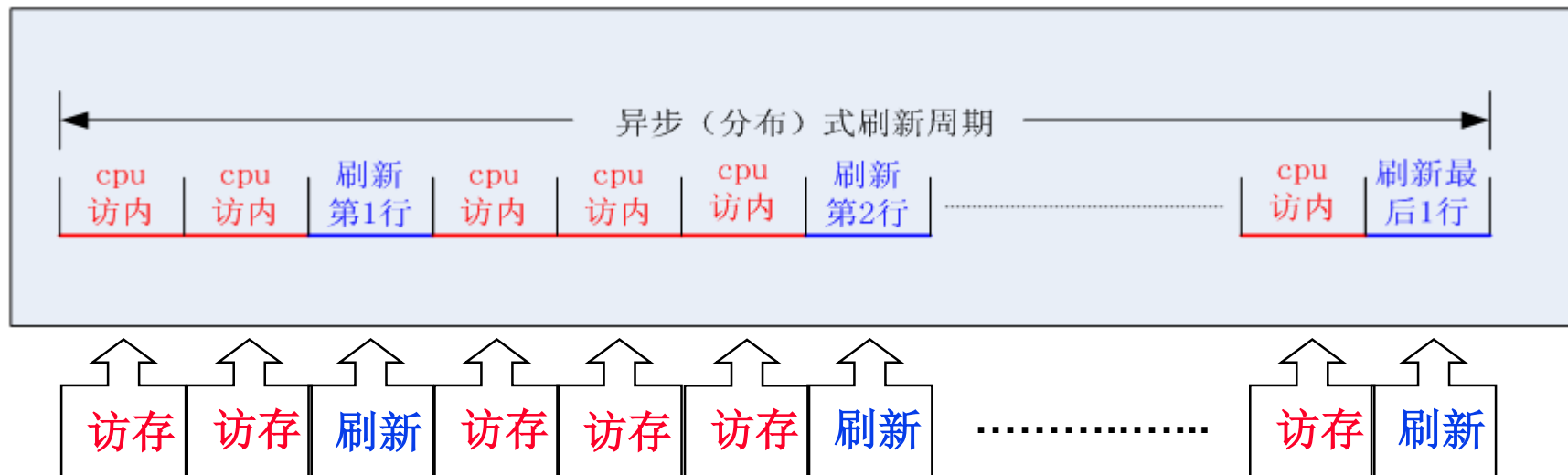
分散式刷新使系统速度降低，但不存在停止读写操作的死时间

5.3 DRAM的刷新方式

❖ 分布式（异步）刷新方式

保证在一个刷新周期内，将存储芯片内的所有行刷新一遍，两次刷新操作之间，可能等时间间距，也可能不等。

异步刷新间隔 = 刷新周期



是前两种方式的结合，可减少死时间，同时保证性能

5.3 DRAM的刷新方式

❖ 分布式（异步）刷新方式

异步刷新间隔 = 刷新周期

以128行为例，在2ms时间内，必须轮流对每一行刷新一次，即每隔 $2\text{ms}/128=15.5\mu\text{s}$ 刷新一行。

这时假定读/写、刷新操作时间都为 $0.5\mu\text{s}$ ，则可用前 $15\mu\text{s}$ 进行正常读/写操作，最后 $0.5\mu\text{s}$ 完成刷新操作。



5.3 DRAM的刷新方式

