

3.3 DRAM存储器

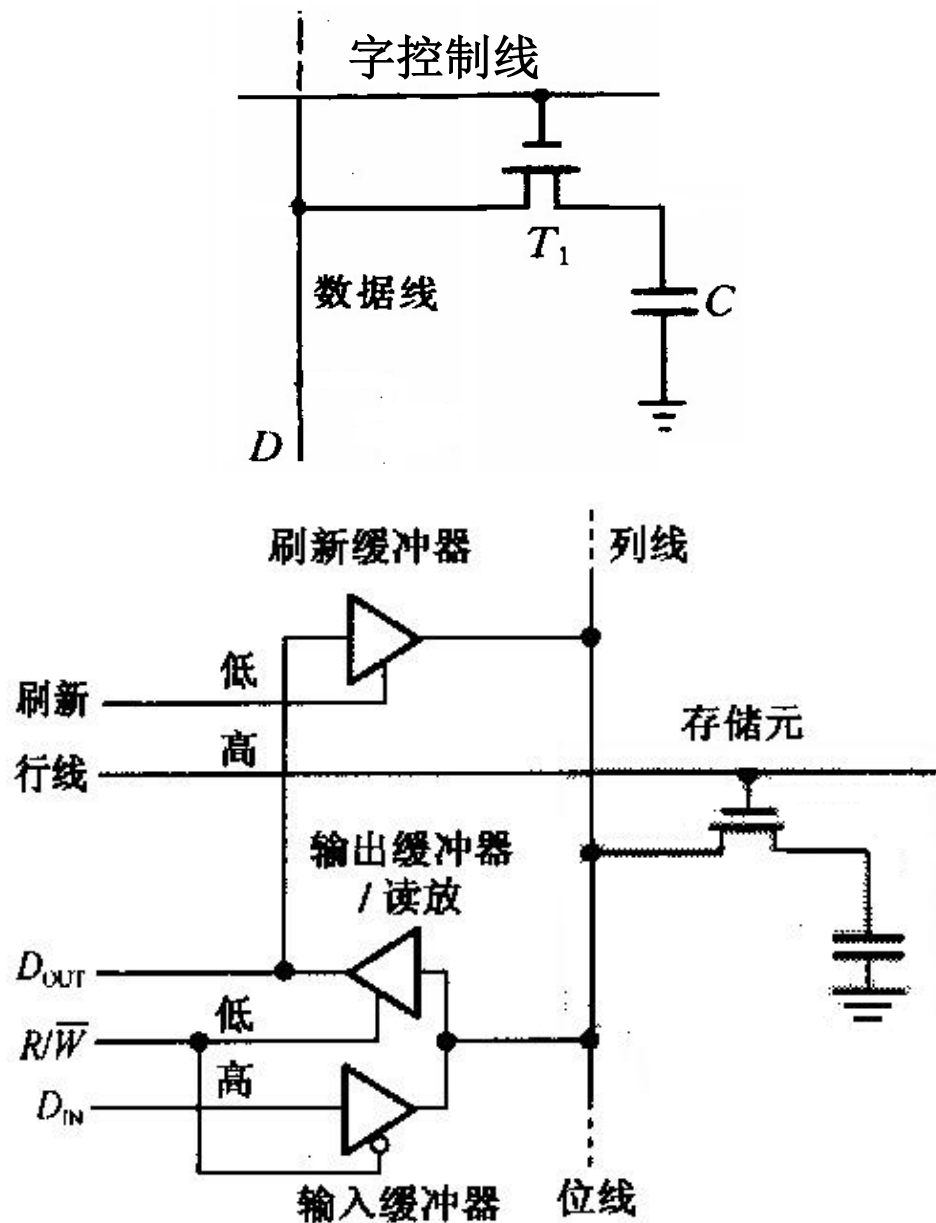
3.3.1 基本存储元

□基本存储元工作原理

- **写入**：字控制线=1，
 - D=0，写0
 - D=1，写1
- **读出**：字控制线=1
 - 数据被破坏；
 - 必须刷新，即**再生**读出。
- **信息暂存**：字控制线=0
 - 必须**定时刷新**。

□DRAM特点：

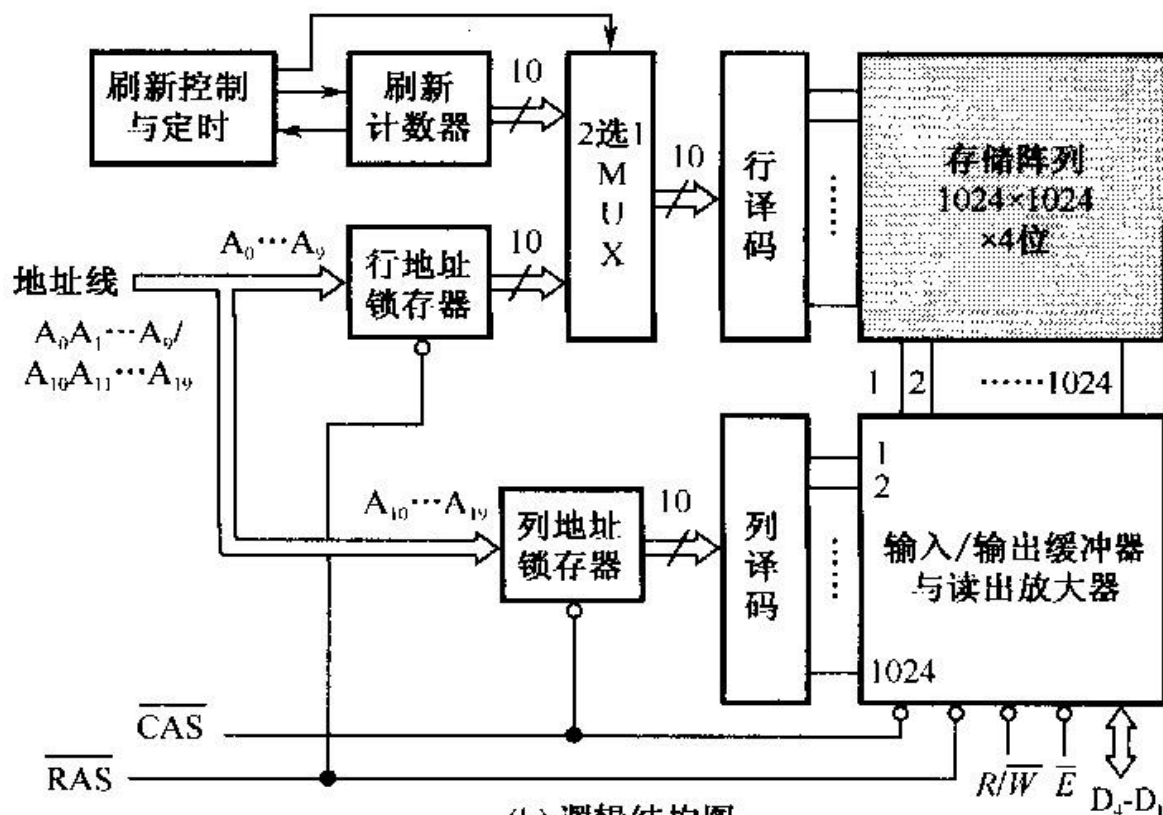
- 利用**记忆电容**来保存信息
- 需要定时刷新；
- 集成度高，功耗低。



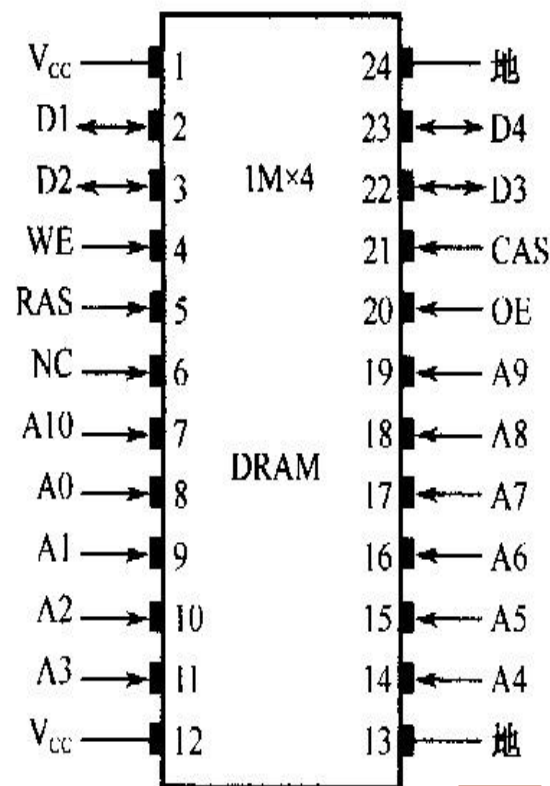
3.3.2 DRAM的逻辑结构

- 内部基本组成：存储阵列、译码电路、输入输出控制电路、[刷新控制电路]。
- 外部引脚：地址线、数据线、控制线。

例：1M X 4位 DRAM结构



(b) 逻辑结构图

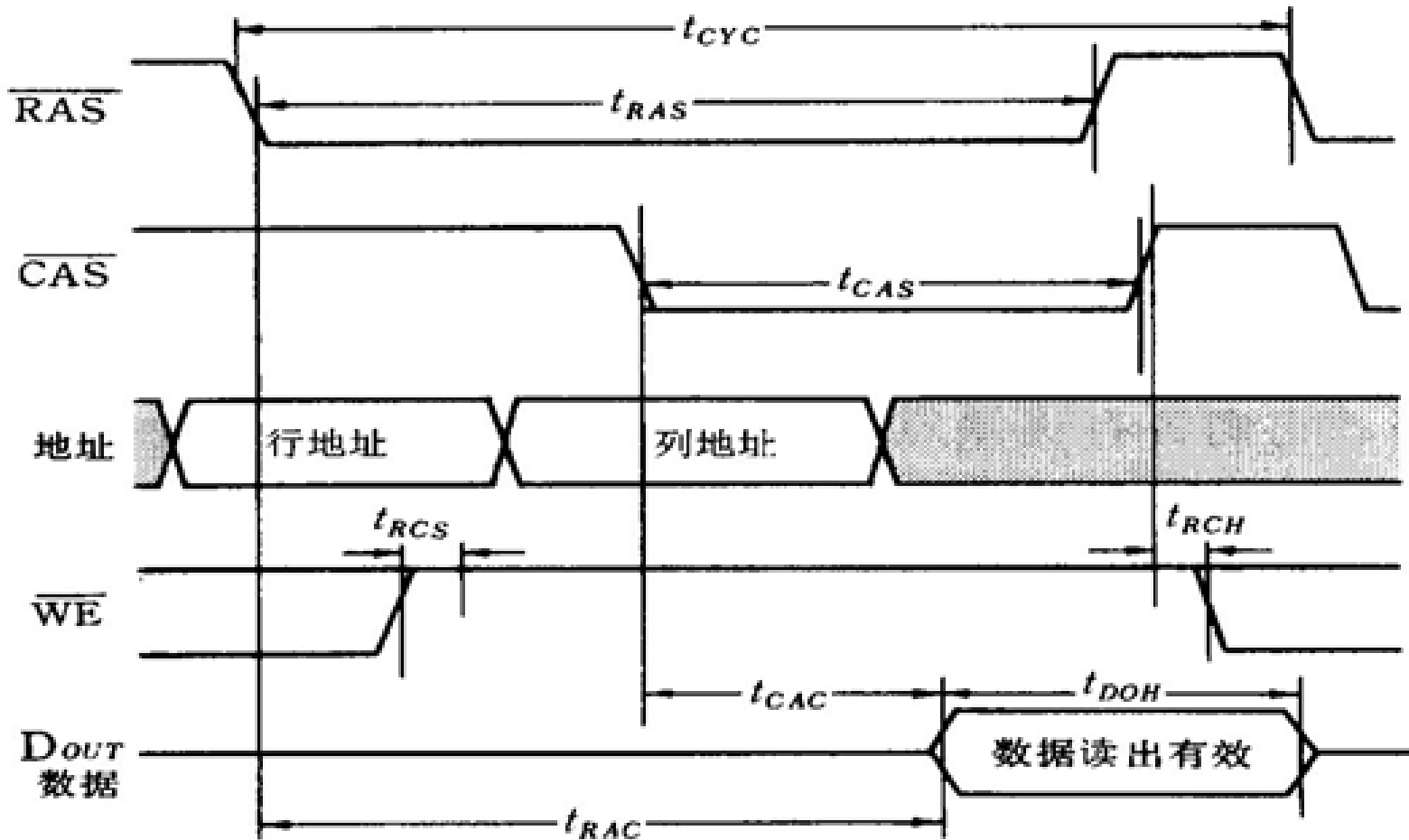


DRAM的地址译码电路

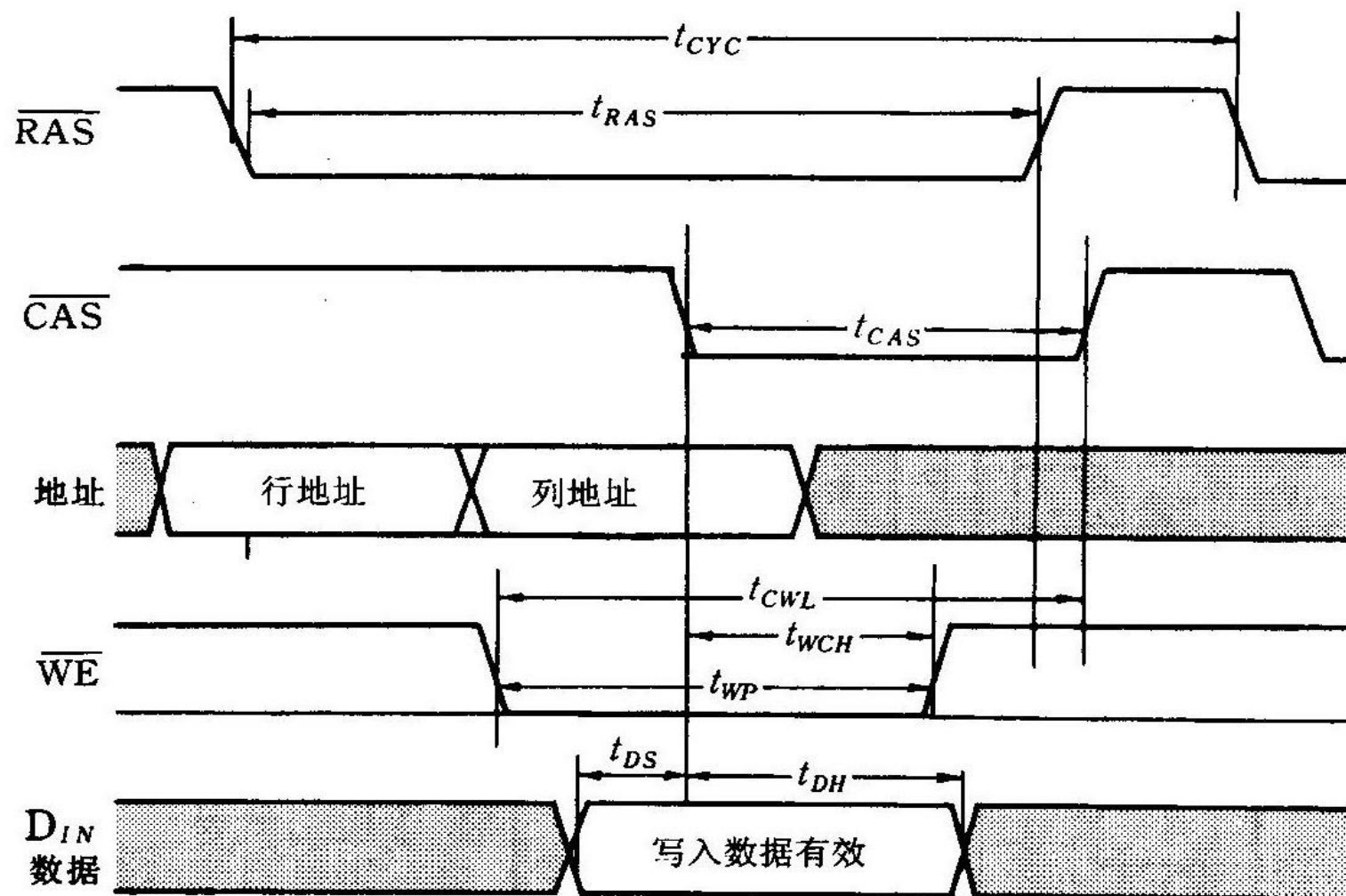
- 采用行、列二维译码结构（行地址、列地址）
- 共用一组外部地址线，分时传送行、列地址
 - 行地址锁存器、RAS
 - 列地址锁存器、CAS
- 需定时进行刷新，与正常读写穿插进行
 - 刷新计数器（刷新地址）与行地址二选一送入行译码

3.3.3 DRAM的读写周期与刷新周期

1、读周期



2、写周期



3、DRAM刷新

□ 存储元基本操作

- 写入：字控制线=1

- 读出：字控制线=1。

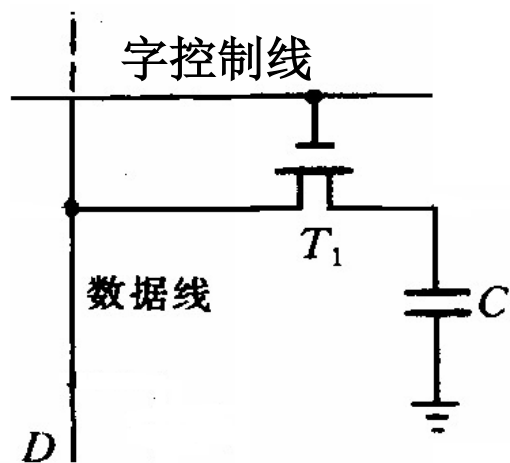


数据被破坏

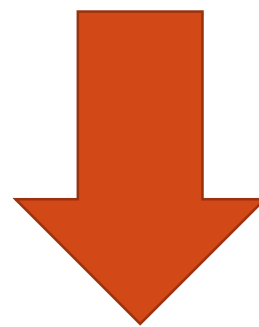
- 信息暂存：字控制线=0



数据随时间丢失



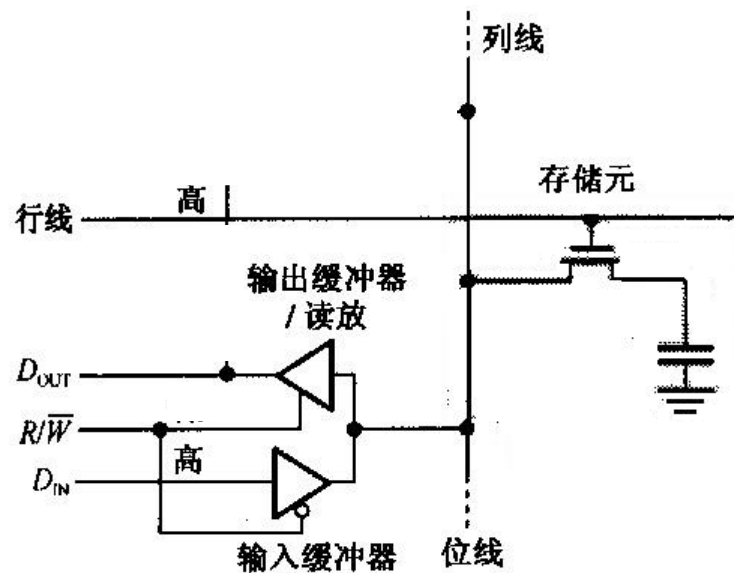
存储元



刷新

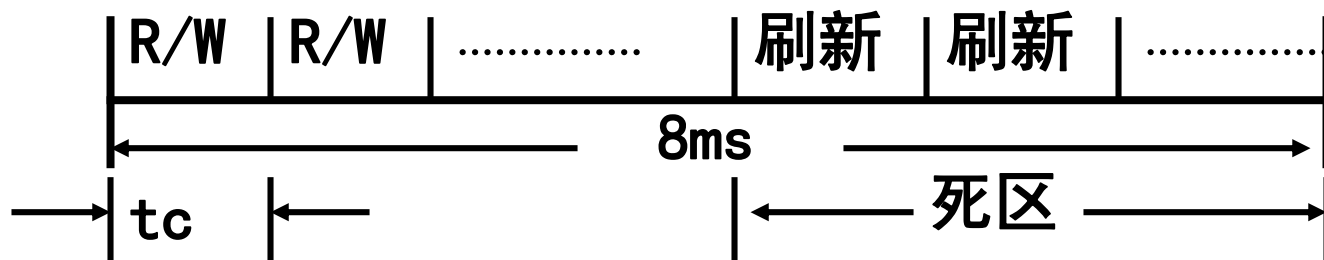
DRAM刷新的基本原理

- 采用“再生读出”方式实现刷新。
- 每个芯片按行进行定时刷新
- 最大刷新周期：在此时间内必须对电容刷新进行，以保持记忆的正确数据。常见：2ms、8ms、16ms等。
- 刷新周期：刷新一行所用时间，与读写周期相同



DRAM刷新方式

- ① **集中刷新**：在最大刷新周期内，**集中**安排DRAM的**所有行**都被刷新**1**次。

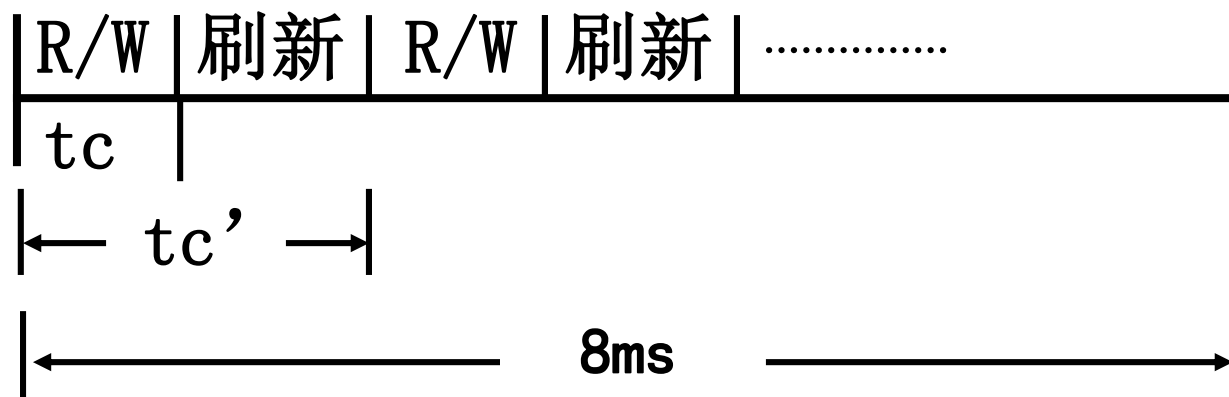


$$\text{死区的时长} = t_c * \text{行数}$$

该方法适用于**实时性要求不高**的场合

DRAM刷新方式

② 分散刷新：在最大刷新周期内，读写周期与刷新周期交替出现。

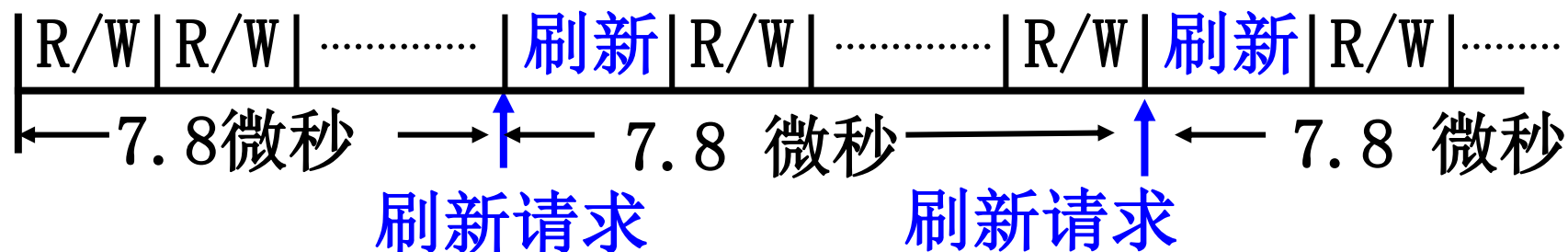


适用于**低速**系统。

DRAM刷新方式

③ 异步刷新：各刷新周期分散安排在最大刷新周期内。

若芯片有1024行，最大刷新周期8ms



最大刷新间隔时间=最大刷新周期/行数

保证在最大刷新周期内，每行至少要被刷新1次。

例题分析

某计算机中由1片DRAM芯片16KB ($128*128$) 构成主存，读写周期为 $0.5\mu\text{s}$ ，最大刷新周期为 2ms 。

- 分析分别采用三种刷新方式的情况
- 若该计算机每 $1\mu\text{s}$ 至少要读写一次，试分析该DRAM芯片用何种刷新方式？

例题分析

	刷新时间	刷新次数
① 集中刷新	$0.5\mu s \times 128\text{行} = 64\mu s$ (死区)	1次/行
② 分散刷新	$\frac{1}{2} \times 2ms = 1ms$	n次/行 (总刷新次数 $\frac{1ms}{0.5\mu s} = 2000\text{次}$)
③ 异步刷新	$0.5\mu s \times 128\text{行} = 64\mu s$ 间隔: $\frac{2ms}{128\text{行}} = 15.625\mu s$ 最终取 $15.5\mu s$ (需为 $0.5\mu s$ 整数倍)	1次/行

DRAM刷新方式小结

□ 集中刷新 vs. 异步刷新

相同：在最大刷新周期内，刷新总次数相同，用于刷新的总时间相同；

不同：前者各行的刷新是连续的，而后者各行的刷新是分散的

□ 分散刷新 vs. 异步刷新

相同：各行的刷新分布在整个最大刷新周期中

不同：分散刷新方式下，各行不止刷新一次。



3.3.4 存储容量的扩充

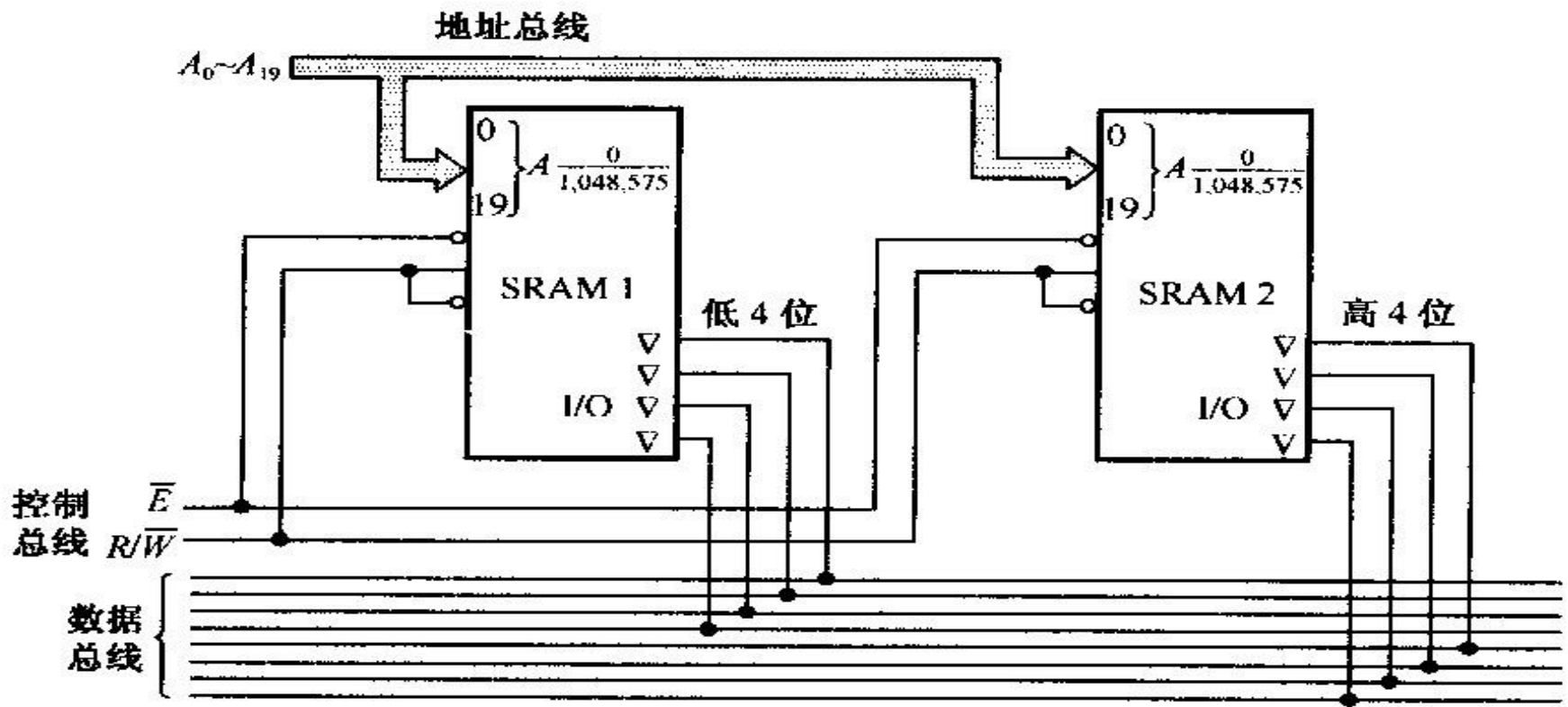
1、位数（字长）扩展方法

- **问题：**给定的芯片字长位数较短，不满足设计要求的存储器字长，则需要多个芯片扩展位数。
- **基本方法：**各芯片的地址线和控制线公用，**数据线单独分开。**

所需芯片数 d =设计要求的存储器容量/已知芯片存储容量

例1: 利用1Mx4位的SRAM芯片，设计一个存储容量为1Mx8位SRAM存储器。

所需芯片数 $d=2$



2、字扩展法（存储容量扩展）

□ **问题：** 给定芯片的容量较小，不满足设计要求的总存储容量，则需要多个芯片扩充字数（容量）。

□ **基本方法：** 芯片地址和数据线公用， R/\overline{W} 公用，片选通过高位地址译码控制。

□ **基本步骤：**

- 计算所需芯片数： $d = \frac{\text{设计要求的存储器容量}}{\text{已知芯片存储容量}}$
- 计算出系统存储容量所需地址数 A_1 ，及芯片的地址数 A_2 。 $A_1 - A_2$ 得出高位地址数。
- 将芯片按高低顺序编号，分配高位地址数值。
- 将高位地址译码后分别控制芯片的片选信号。

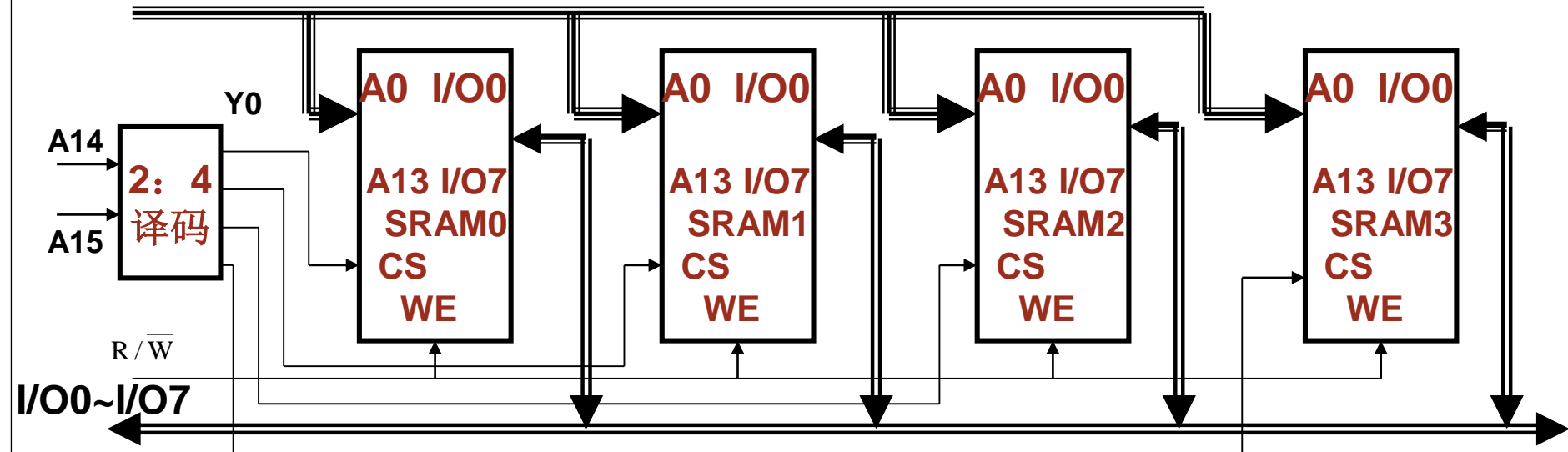
例2: 用16Kx8的SRAM芯片组成65536x8的存储器。

- 芯片个数：4，芯片地址14位
- 高位地址：A15 A14
- 00控制SRAM1~11控制SRAM4
- 片选逻辑确定：

A15A14	A15 ~ A0	
0 0	0000~3FFF	SRAM0
0 1	4000~7FFF	SRAM1
1 0	8000~BFFF	SRAM2
1 1	C000~FFFF	SRAM3

$$CS_0 = \overline{A_{15}} \overline{A_{14}} \quad CS_1 = \overline{A_{15}} A_{14} \quad CS_2 = A_{15} \overline{A_{14}} \quad CS_3 = A_{15} A_{14}$$

A0~A13



3、 字位扩展法

字向和位向都扩充。

例3: 用2114 (1Kx4) 的芯片及译码、门电路组成4Kx8的存储器

A11 A10 A11~A0

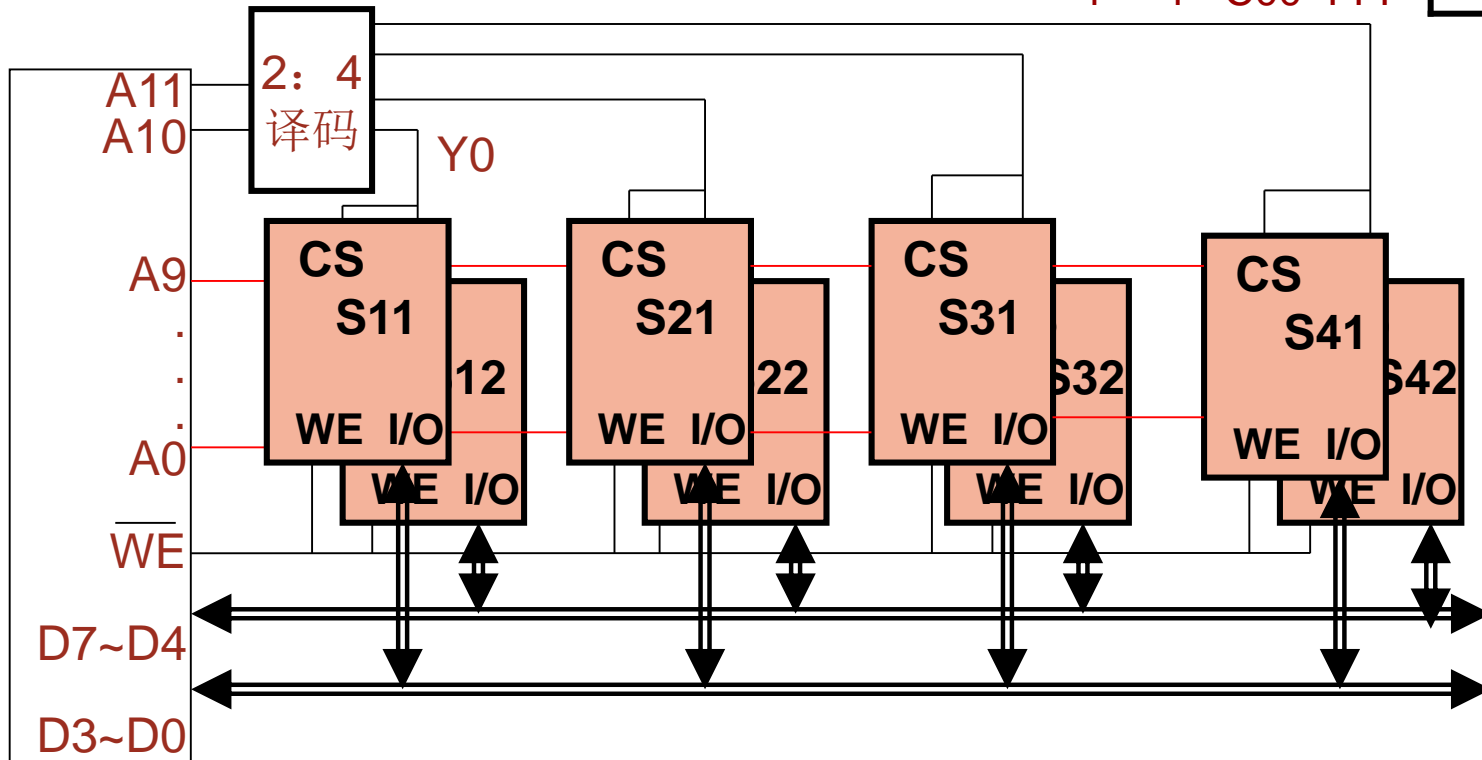
0 0 000~3FF

0 1 400~7FF

1 0 800~BFF

1 1 C00~FFF

S11	S12
S21	S22
S31	S32
S41	S42

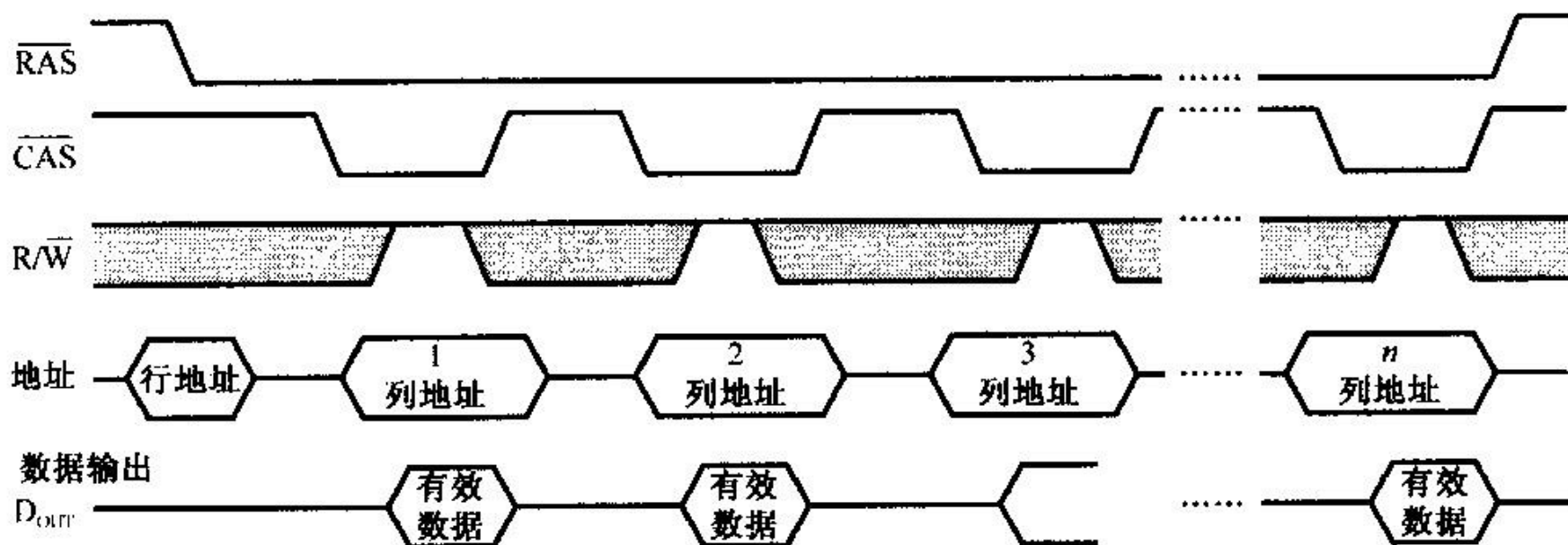


3.3.5 高性能DRAM

目的：增强基本DRAM的功能。

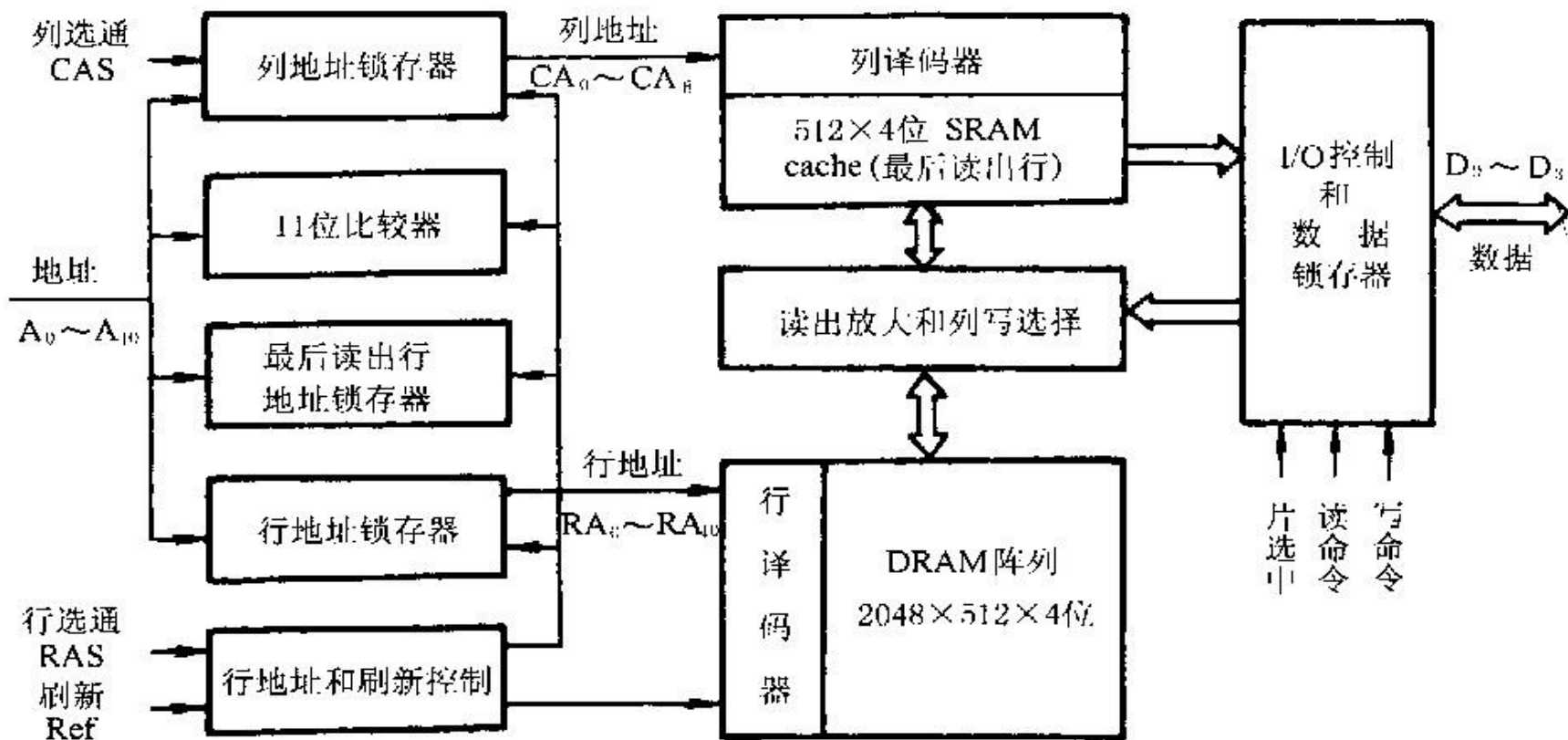
1、FPM-DRAM：快速页模式动态存储器

- 利用程序的局部性原理，1页里连续的数据只改变列地址即可。



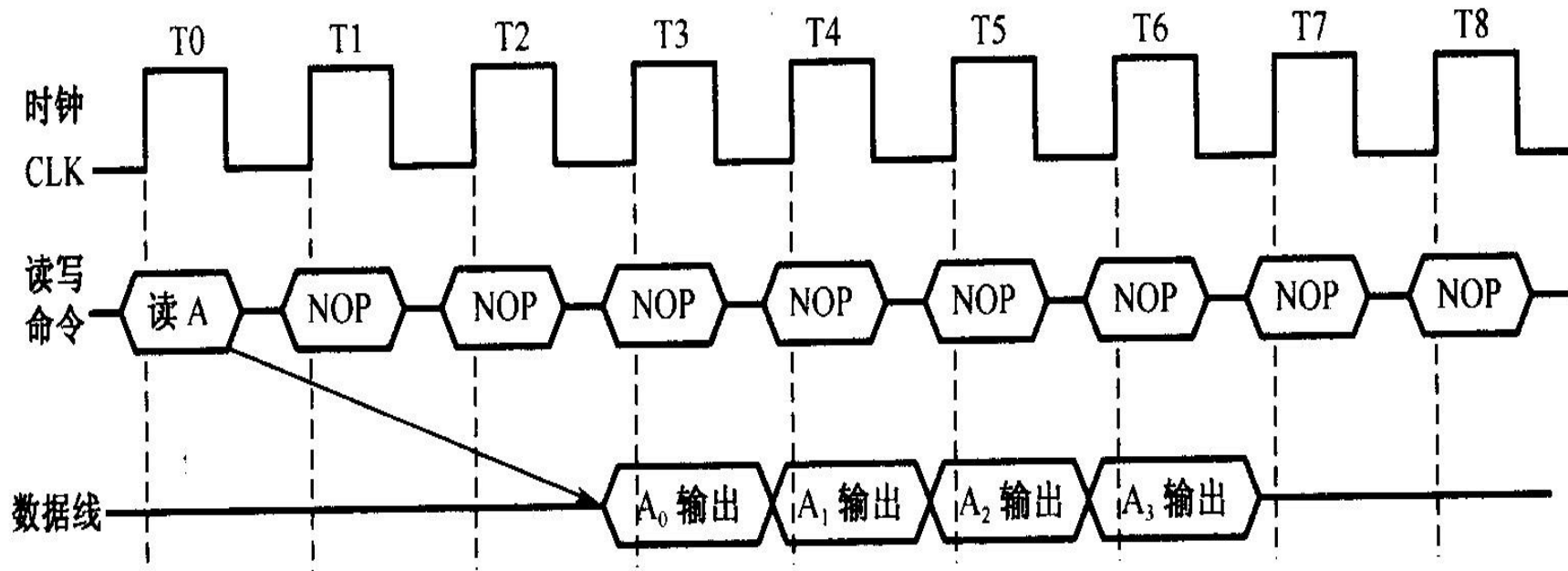
2、CDRAM: 带cache的动态存储器

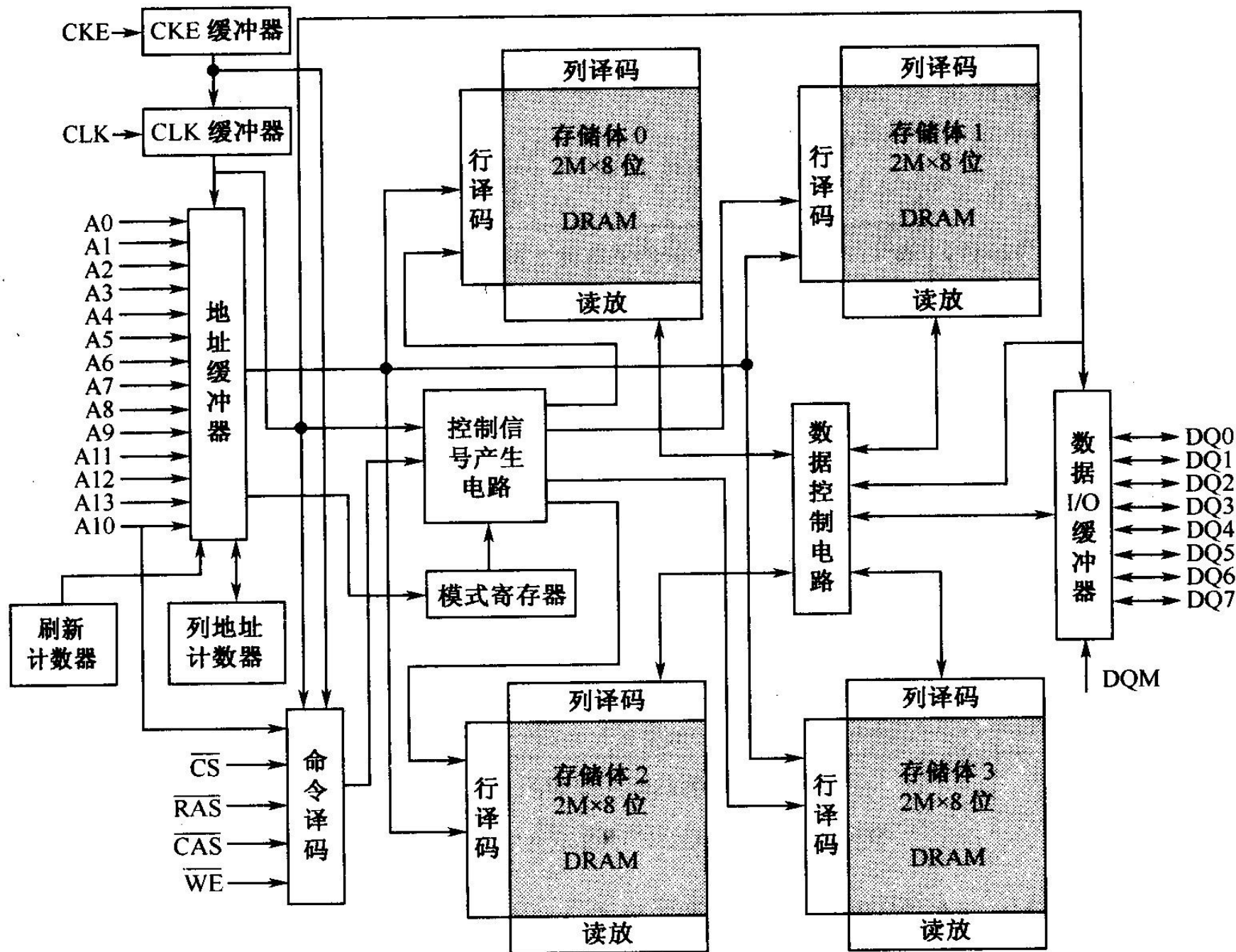
- 在DRAM基础上集成一个**小SRAM**，使得局部连续数据可从SRAM读出（猝发式读取），以提高速度。
- 例1Mx4CDRAM芯片结构



3、SDRAM: 同步型动态存储器

- 在DRAM基础上增加一个时钟信号，使得对数据的读取与系统时钟同步。





例 主存的组成

设某32机，主存容量为 $4\text{M} \times 32$ ，所用DRAM芯片为 $1\text{M} \times 4$ ，DRAM 控制器用W4006AF（CPU与DRAM之间转换控制器）。

- 4个模块
- 每个模块4组
- 每组2个存储器芯片。

