



# 计算机组成原理

## 第5章 存储器

-SRAM/DRAM原理、主存系统构建

唐本第四章\$4.1, \$4.2

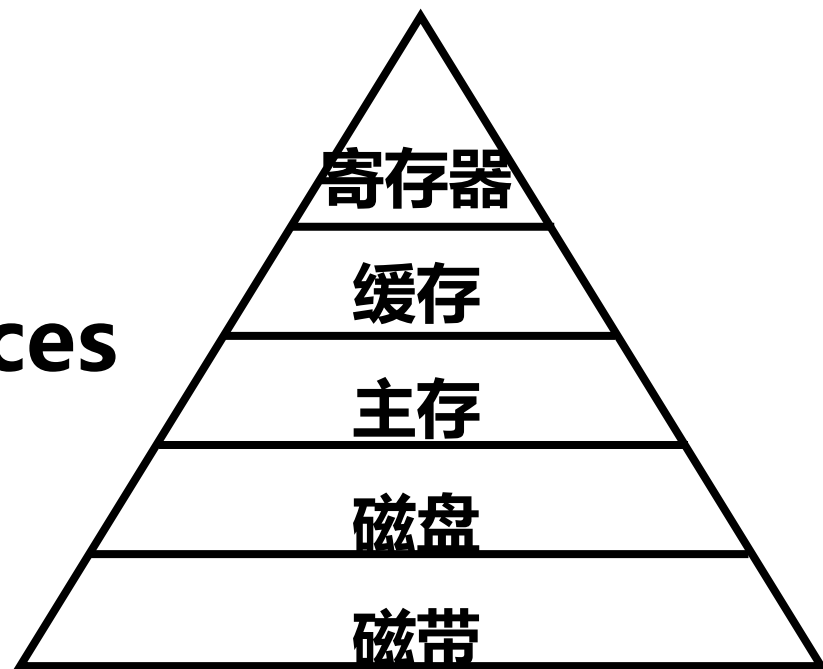
COD5 \$5.2, A.9

llxx@ustc.edu.cn



# 本章内容 (唐本)

- 4.1 概述
  - 分类, 指标
- 4.2 主存储器
  - SRAM/DRAM devices
  - 主存系统组成
  - 提高访存性能
- 4.3 高速缓冲存储器
- 4.4 辅助存储器



# 分类



- 按存储介质
  - 半导体存储器, 磁表面存储器, 磁芯存储器, 光盘存储器
- 按存取方式 (read/write、load/store、access)
  - 内存 (memory, main/primary/internal memory)
    - 随机存储器RAM (Random Access Memory)
    - 只读存储器ROM (Read-Only Memory)
  - 外存 (storage, Auxiliary/External/secondary memory)
    - 串行访问存储器, 直接访问存储器
- 按工作机制: 存储技术, 同步方式
  - SRAM, DRAM, SDRAM, DDR SDRAM, Flash (EEPROM)
- 按系统地位
  - 主存 (内存)、辅存 (外存)、缓存 (Cache)
  - 在线 (online)、离线 (offline)





# 存储器的分类—按存储介质

## • 半导体存储器

- 半导体器件组成，超大规模集成电路芯片
- 优点：体积小，功耗低，存取时间短
- 缺点：电源消失，所存信息也随即丢失，属于一种**易失性 (Volatile)**
- 两类：**双极型 (TTL)** 半导体存储器，速度快；**MOS** 半导体存储器，集成度高且制造简单、成本低廉、功耗小，**广泛应用**。

## • 磁表面存储器

- 在金属或塑料基体的表面涂上一层磁性材料作为记录介质。
- 按载磁体形状的不同，分为磁盘、磁带和磁鼓。

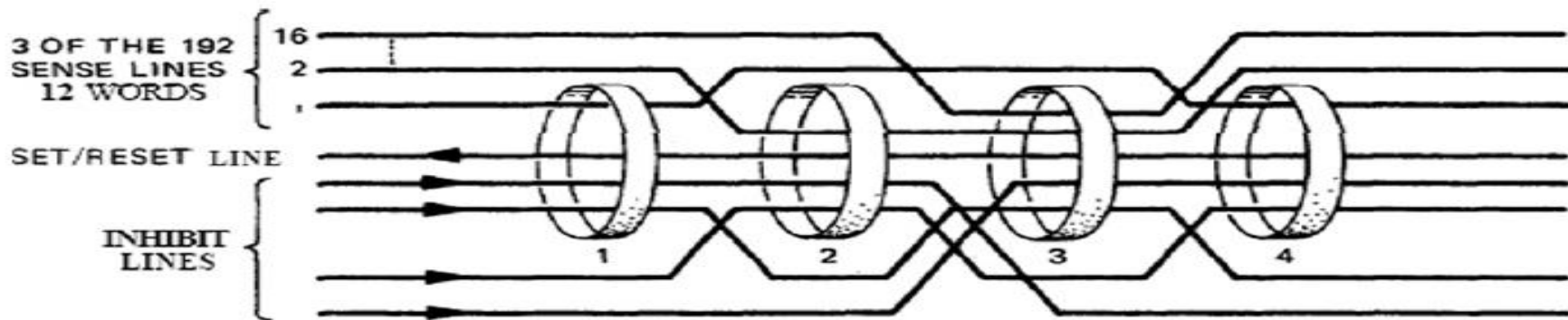
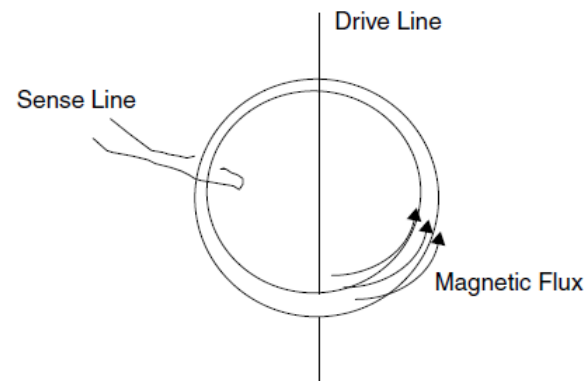
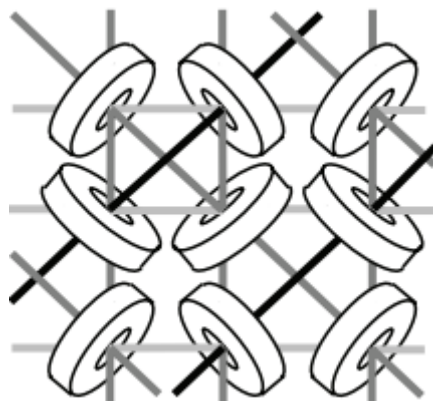
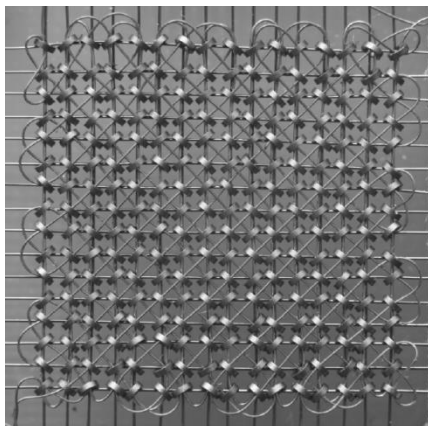
## • 光盘存储器

- 光盘存储器是应用激光在记录介质（如磁光材料等）上进行读写的存储器，具有非易失性的特点。
- 光盘记录密度高、耐用性好、可靠性高和可互换性强等优良特点。

# 存储器的分类—按存储介质

- **磁芯存储器 (Core memory)**

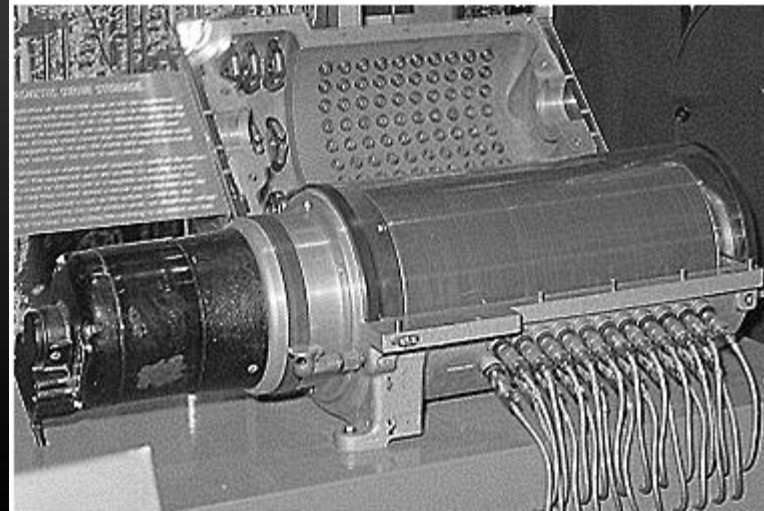
- 磁芯是使用硬磁材料做成的环状元件，在磁心中穿有驱动线（通电流）和读出线，这样就可以进行读写操作（访问时间：10ms）
- 磁芯属于磁性材料，**非易失性 (Non-Volatile)**
- 阿波罗飞船AGC使用：体积庞大、工艺复杂（by hand）且功耗大，**已弃用**



# Magnetic drum storage (1932发明)



- 磁头与鼓筒表面保持微小而恒定的间隙(0.02~以下)并沿鼓筒轴线均匀排列
- IBM 650系列计算机中的主存储器，1953~1962
  - 磁鼓长12英寸，12500转/S，每支可以保存1万个字符(不到10K)。





# 存储器的分类—按存取方式

- **随机存储器RAM (Random Access Memory)**
  - 存储器中任何存储单元的内容都能随机存取
    - **存取时间**与存储单元的物理位置无关。
    - 如主存储器
  - 由于存取原理的不同，又分为静态RAM和动态RAM。
    - SRAM以**触发器**原理寄存信息
    - DRAM以**电容**充放电原理寄存信息
    - SDRAM?
    - SSRAM?
  - DRAM was patented in 1968 by Robert Dennard at IBM

# 存储器的分类—按存取方式



- **只读存储器ROM (Read Access Memory)**
  - 只能对其存储的内容读出，不能对其重新写入的存储器。
  - 掩模型只读存储器MROM (Masked ROM)
    - 采用掩模工艺，把原始信息记录在芯片中，一旦制成就无法更改。
- **可编程只读存储器PROM (Programmable ROM)**
  - 可擦除可编程只读存储器EPROM (Erasable Programmable ROM)
  - 电可擦除可编程的只读存储器EEPROM (Electrically Erasable Programmable ROM)
  - 快擦型存储器Flash Memory
    - 在线编程，速度比EEPROM要快得多





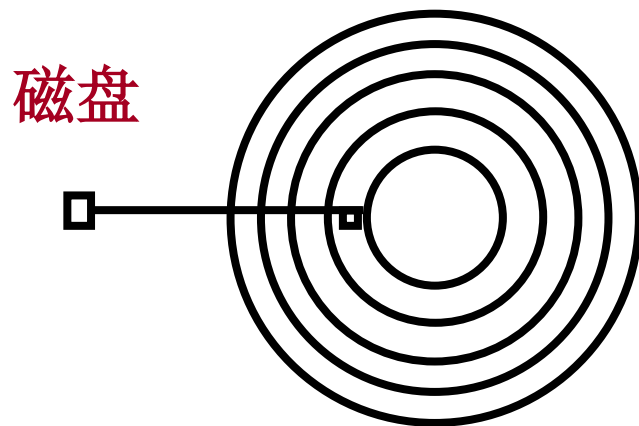
# 存储器的分类—按存取方式

- **串行访问存储器**

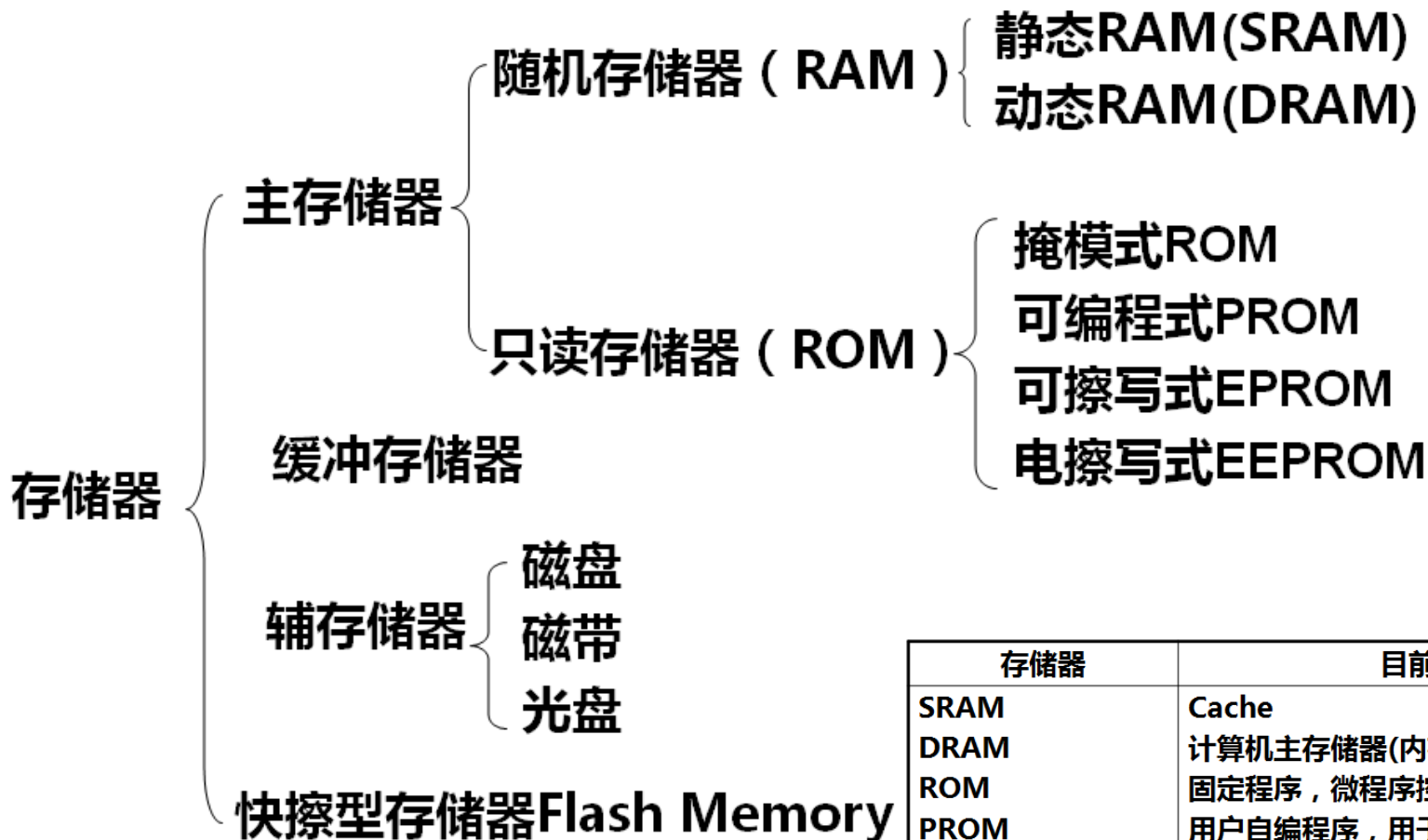
- 对存储单元进行读写操作时，需按其物理位置的先后顺序寻找地址，则这种存储器叫做**串行访问存储器**。如**磁带**。也叫**顺序存取存储器**。

- **直接存取存储器。**

- 如**磁盘**



# 存储器分类



存储器	目前主要应用
SRAM	Cache
DRAM	计算机主存储器(内存)
ROM	固定程序, 微程序控制器存储器
PROM	用户自编程序, 用于工业控制机或电器中
EPROM	用户编写并修改程序或产品试制阶段编程序
E <sup>2</sup> PROM	IC卡存储器
Flash Memory	固态盘(优盘)、IC卡



## 4.2 主存储器

1. 概述
2. 半导体存储芯片
3. RAM
4. ROM
5. 存储器与CPU的连接
6. *存储器的校验—海明码*
7. 提高访存速度的措施



# 本节内容

## 1. 主存储器的基本组成和技术指标



## 3. 存储器与CPU的连接

## 4. 存储器的校验—海明码

## 5. 提高访存速度的措施

# 主存的基本组成：存储芯片

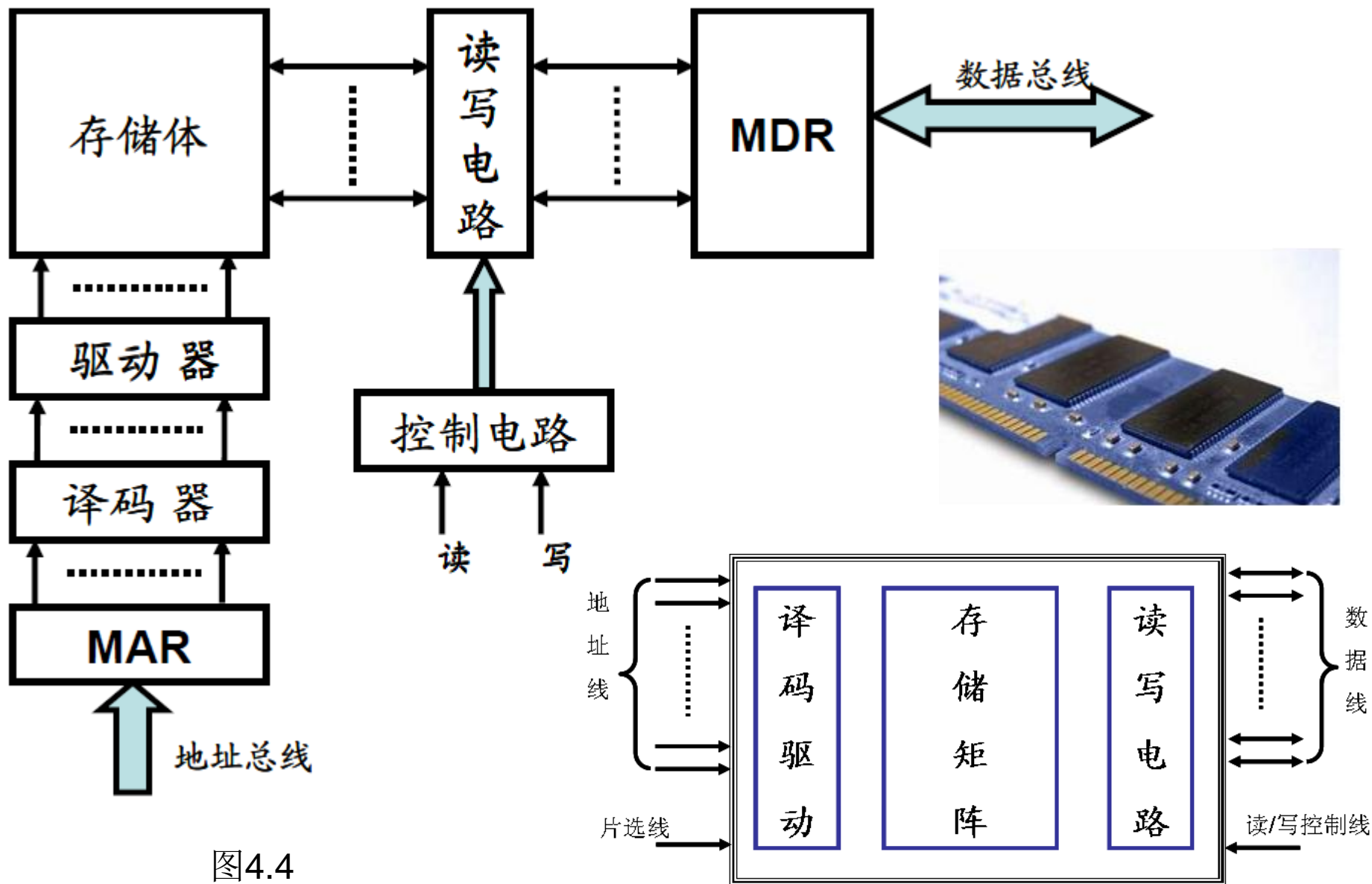


图4.4

# 主存和CPU (Cache) 的接口

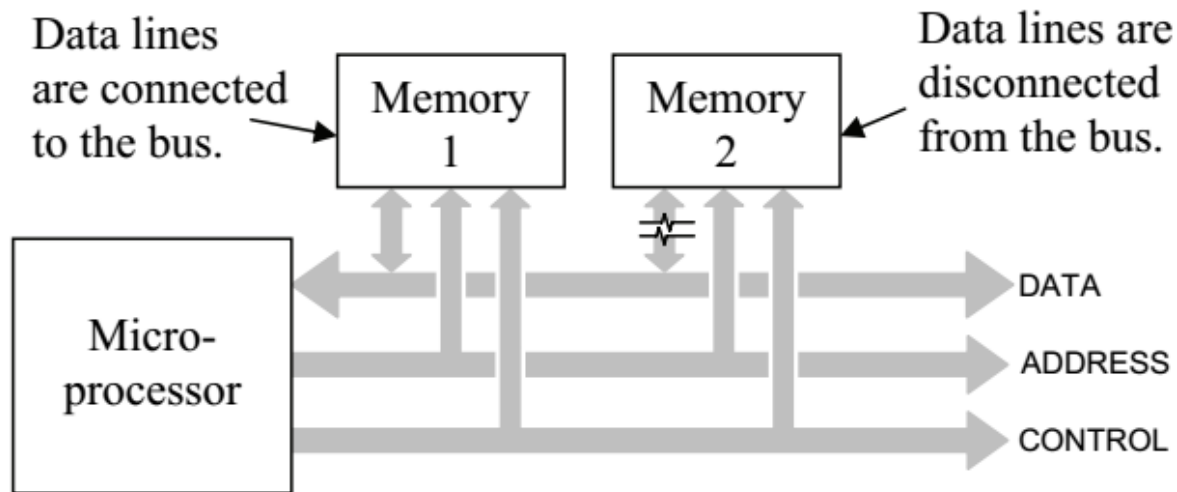
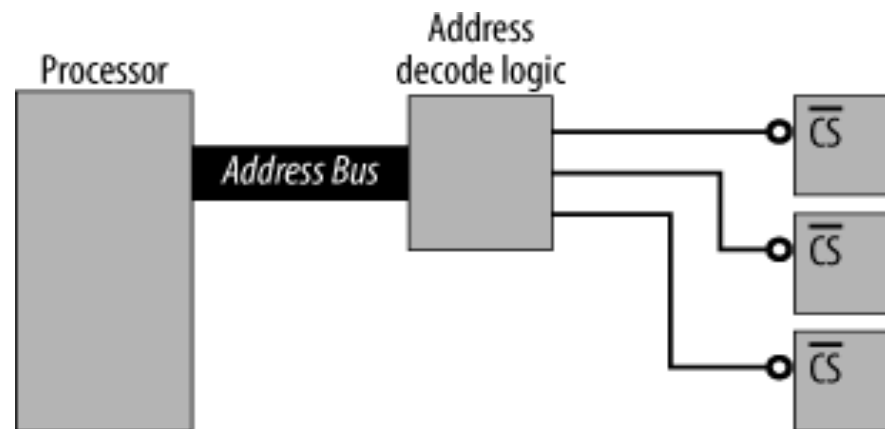
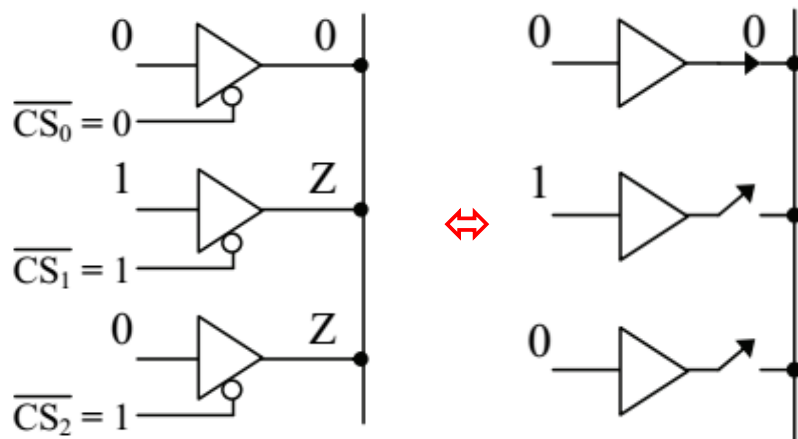
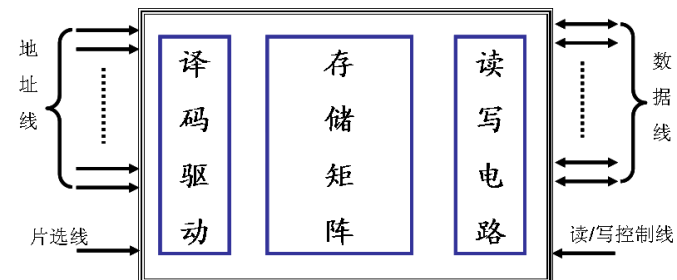


图4.7



# 主存的主要技术指标



## 存储容量:

指存储器所能存储的二进制代码总数。

有2种表示方法:

1) 存储二进制信息的总**位数**: bits

存储容量 = 存储单元个数 × 存储字长

2) 存储**字节**的总数: Bytes

存储容量 = 存储单元个数 × 存储字长 / 8

## 几个单位:

字节: Byte    位: bit            1B=8b

1KB=2<sup>10</sup>B,    1MB=2<sup>20</sup>B,    1GB=2<sup>30</sup>B,    1TB=2<sup>40</sup>B



# 主要技术指标

**速度指标：**存取时间和存储周期、存储器的带宽。

- **存取时间 (MEM Access Time)：**指从启动一次存储器操作到完成该操作所需的全部时间。

**读出时间：**从存储器接收到有效地址开始，到产生有效输出所需的全部时间。

**写入时间：**从存储器接收到有效地址开始，到数据写入被选中单元所需的全部时间。

- **存储周期 (Mem Cycle Time)：***指连续启动两次独立的操作 (如两次读操作) 所需的最小间隔时间。*

通常大于存取时间。

- **存储器的带宽：**单位时间里存储器所存取的最大信息量。  
单位：以位/秒、字节/秒。

例：存取周期为500ns，每个存取周期可访问16位，则它的带宽为

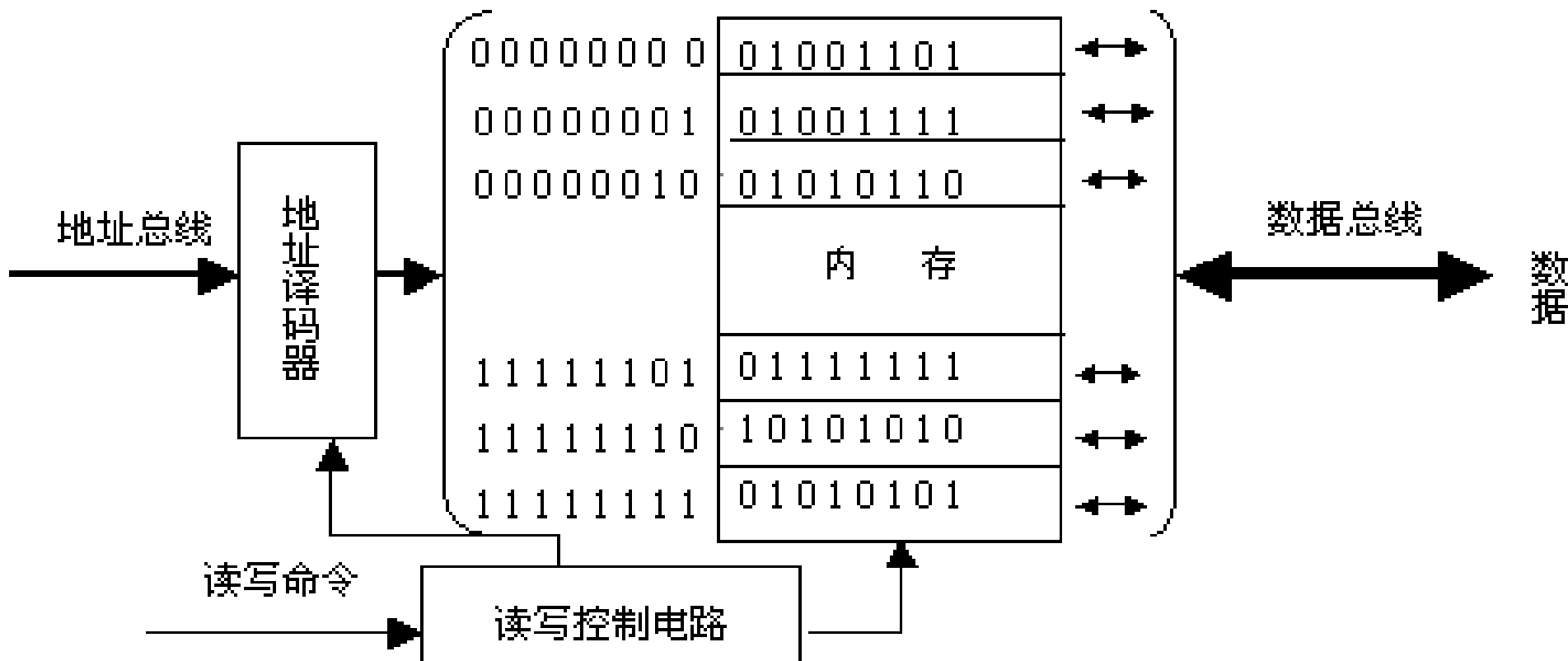
$$16\text{位} \times \frac{1}{500 \times 10^{-9}\text{秒}} = 32 \times 10^6 \text{位/秒} = 32\text{M位/秒}$$





# 主存中存储单元的地址分配

- 按字节寻址，按字寻址
- 地址单元译码方式：线选法，重合法



# 存储芯片的译码驱动方式—线选法

- 特点：用一根字选择线（**字线**）直接选中一个存储单元的各位（如一个字节）。
- 这种方式结构简单，但适于容量不大的存储芯片。
- **位线**：数据线

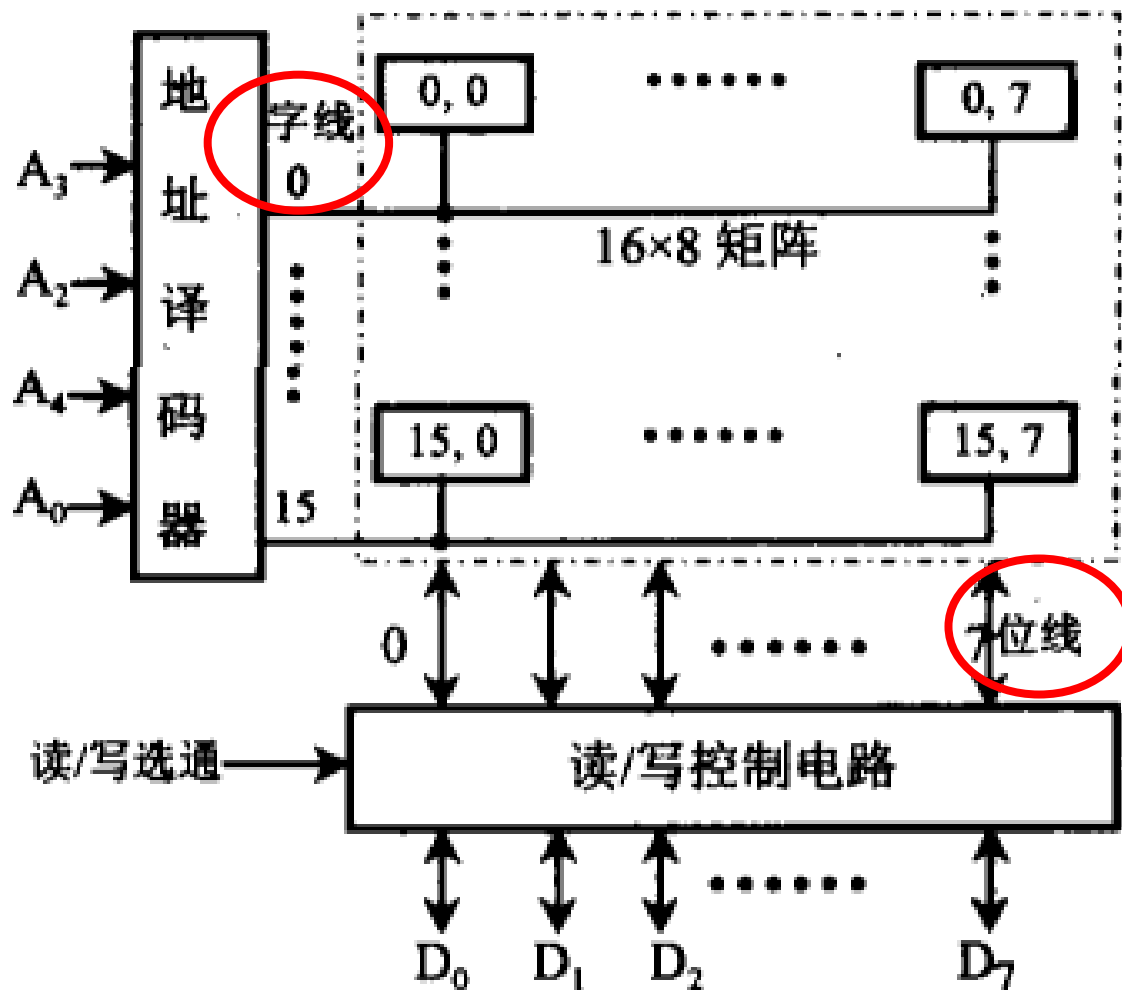


图4.9

# 存储芯片的译码驱动方式—重合法



- 特点：被选单元由X、Y两个方向的地址决定，因此叫重合法。

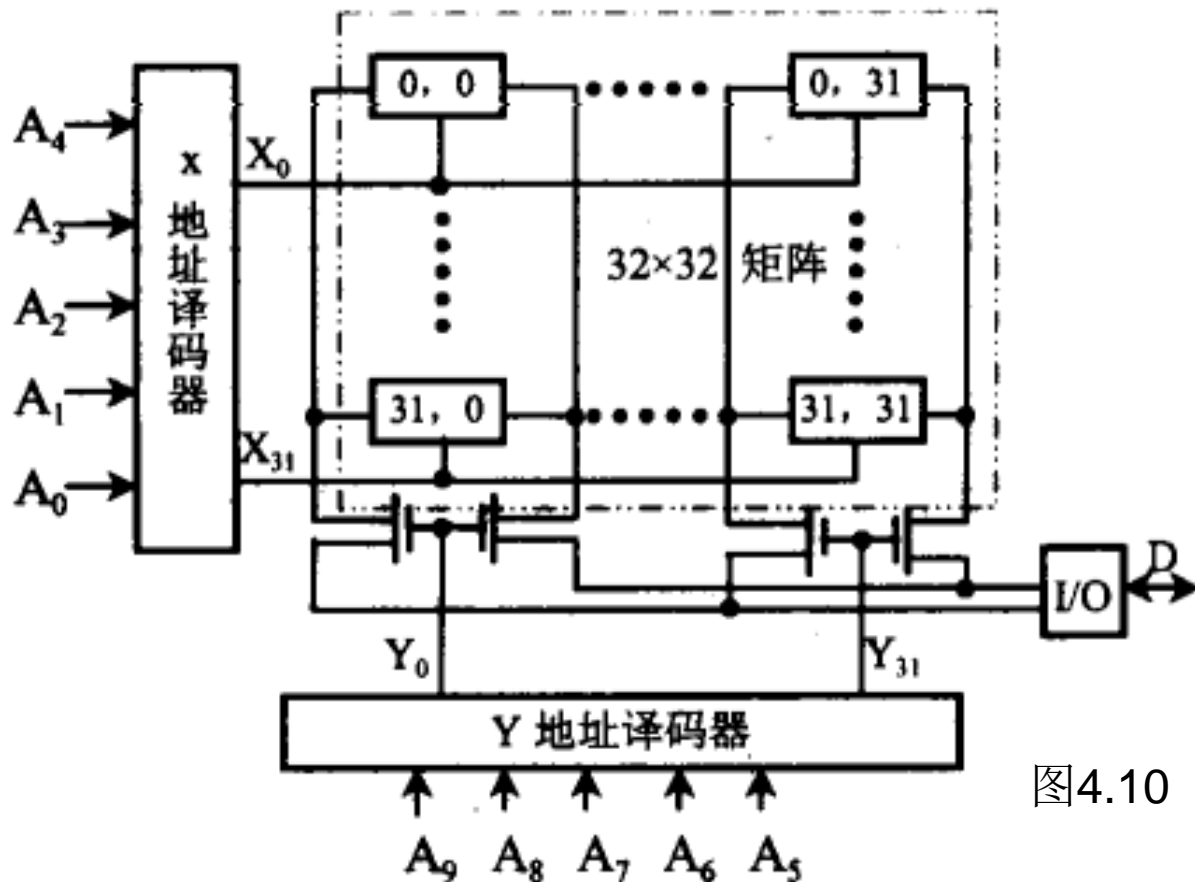


图4.10

- 一位片：当构成1KX1字节的存储器时，需用8片

# 静态RAM (SRAM)



基本存储电路由6个MOS管构成:

$T_1 \sim T_4$ 组成双稳态触发器

$T_1, T_2$ 为放大管,

$T_3, T_4$ 为负载管,

$T_5, T_6$ 为开关控制管

读出: 选通 $T_5, T_6$

A点与I/O连通

B点与 $\overline{I/O}$ 连通

写入: 选通 $T_5, T_6$

A点与I/O连通

B点与 $\overline{I/O}$ 连通

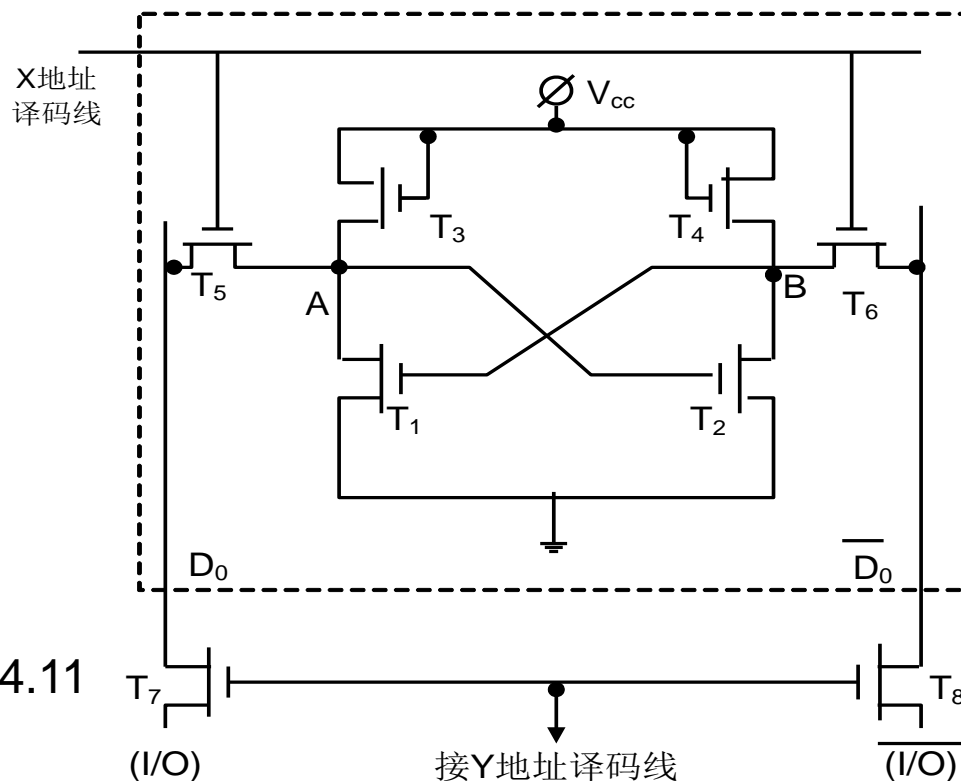
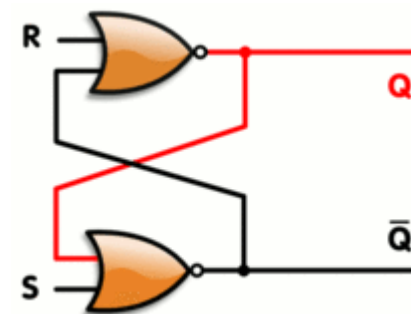


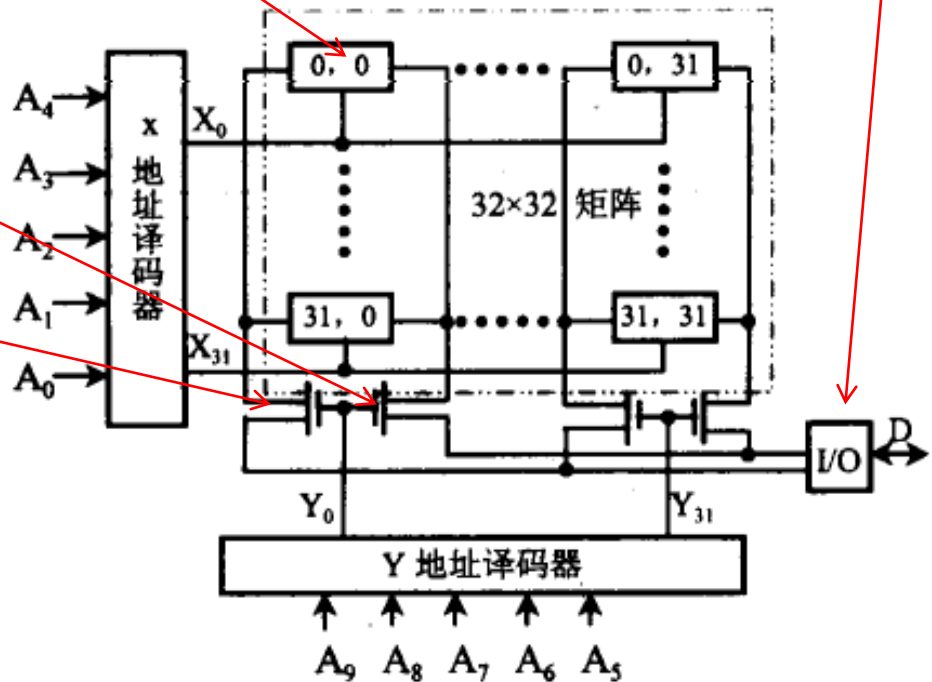
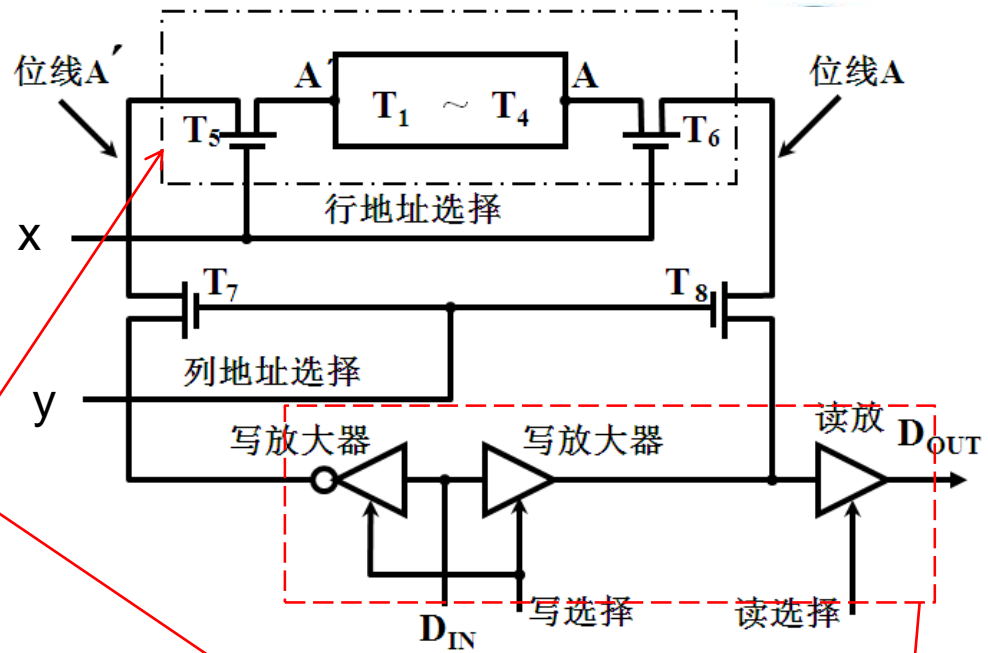
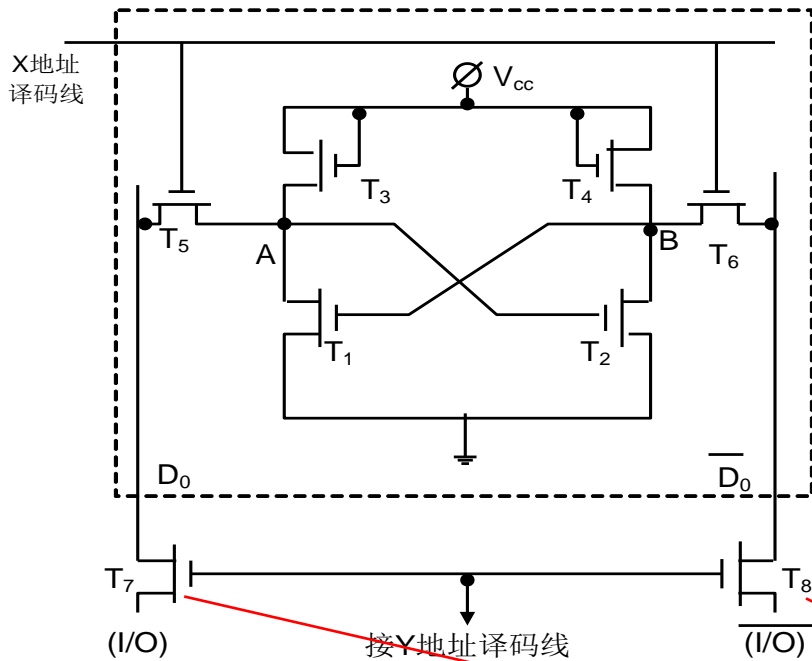
图4.11



# 基本工作原理

- A,B点的电平代表该六管单元的二进制信息状态;
  - 写信息: 当地址选择线为高电平,  $T_5, T_6$  导通, 则I/O线与A点相连,  $\overline{I/O}$ 线与B点相连;  
若写入 '1', 则A='1', B='0', 使 $T_1$ 截止,  $T_2$ 导通  
写入 '0', A='0', B='1'—— $T_1$ 导通,  $T_2$ 截止  
此时写入信号与地址译码信号消失, 该六管单元状态仍然保持稳定;
  - 当要读出信息:  
地址选择线为高电平, 使 $T_5, T_6$ 导通,  
则A点, B点状态分别被送至I/O,  $\overline{I/O}$ 线,  
且六管单元的状态不变
- SRAM电路工作稳定, 不需要刷新电路, 但MOS管数多, 集成度不高, 且功耗较大。

# I/O读写控制?



地址：字线

数据：位线

控制：读选择，写选择

# 芯片结构

- 由许多基本的六管单元电路组成；
- 容量为单元数与数据线位数的乘积，1K\*4位。

## Intel 2114外特性示意图

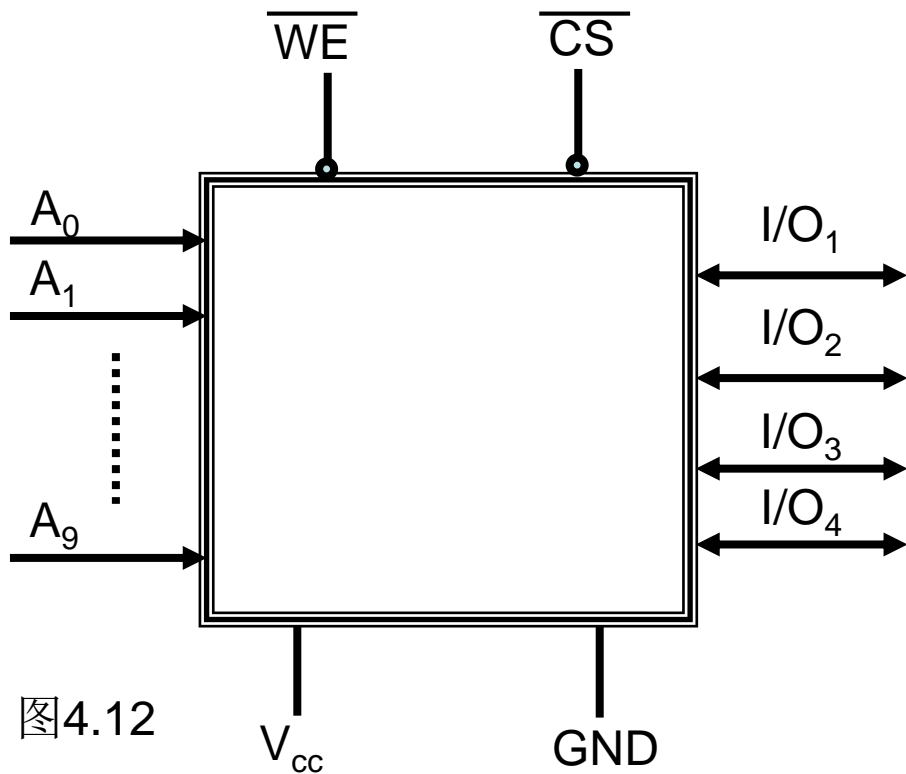
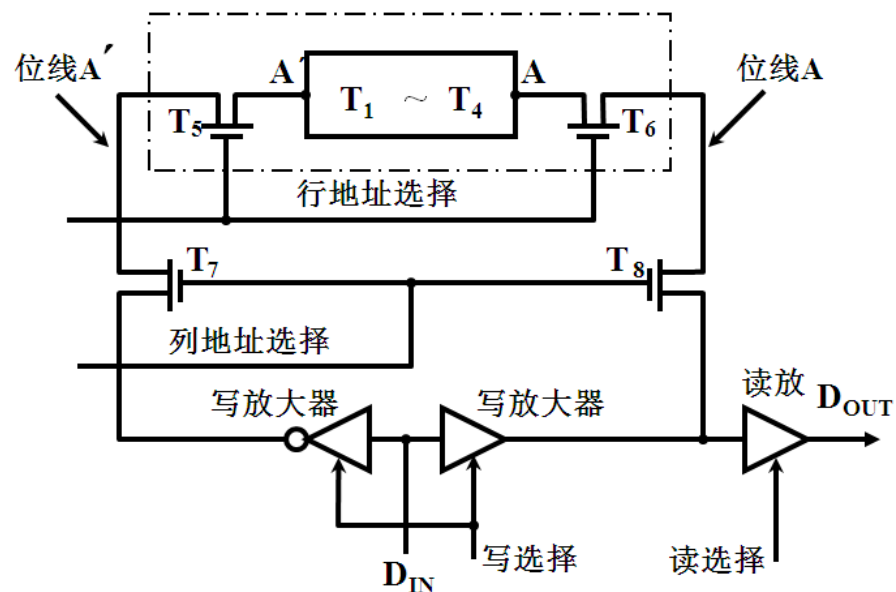


图4.12



# 2114RAM矩阵结构示意图

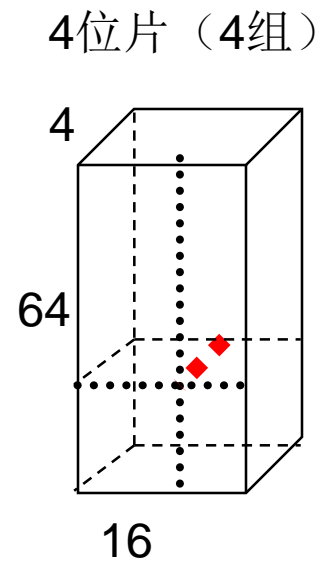
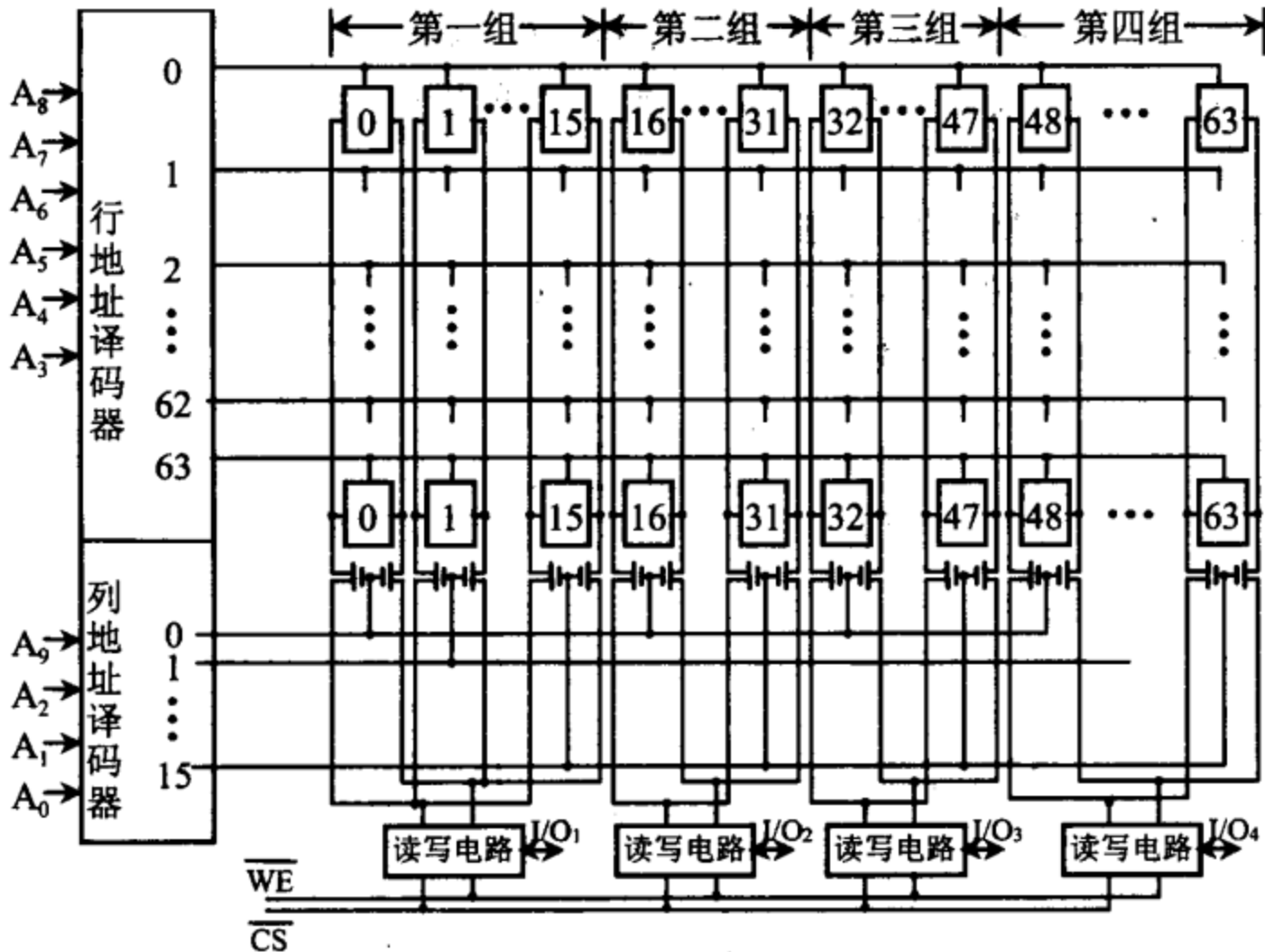


图4.14



# 2114 (SRAM) 时序



## • 读周期:

- 地址有效→CS有效→数据输出→CS复位→地址撤销
- 只有当地址有效经 $t_A$ 后, 且当片选有效经 $t_{CO}$ 后, 数据才能稳定输出。

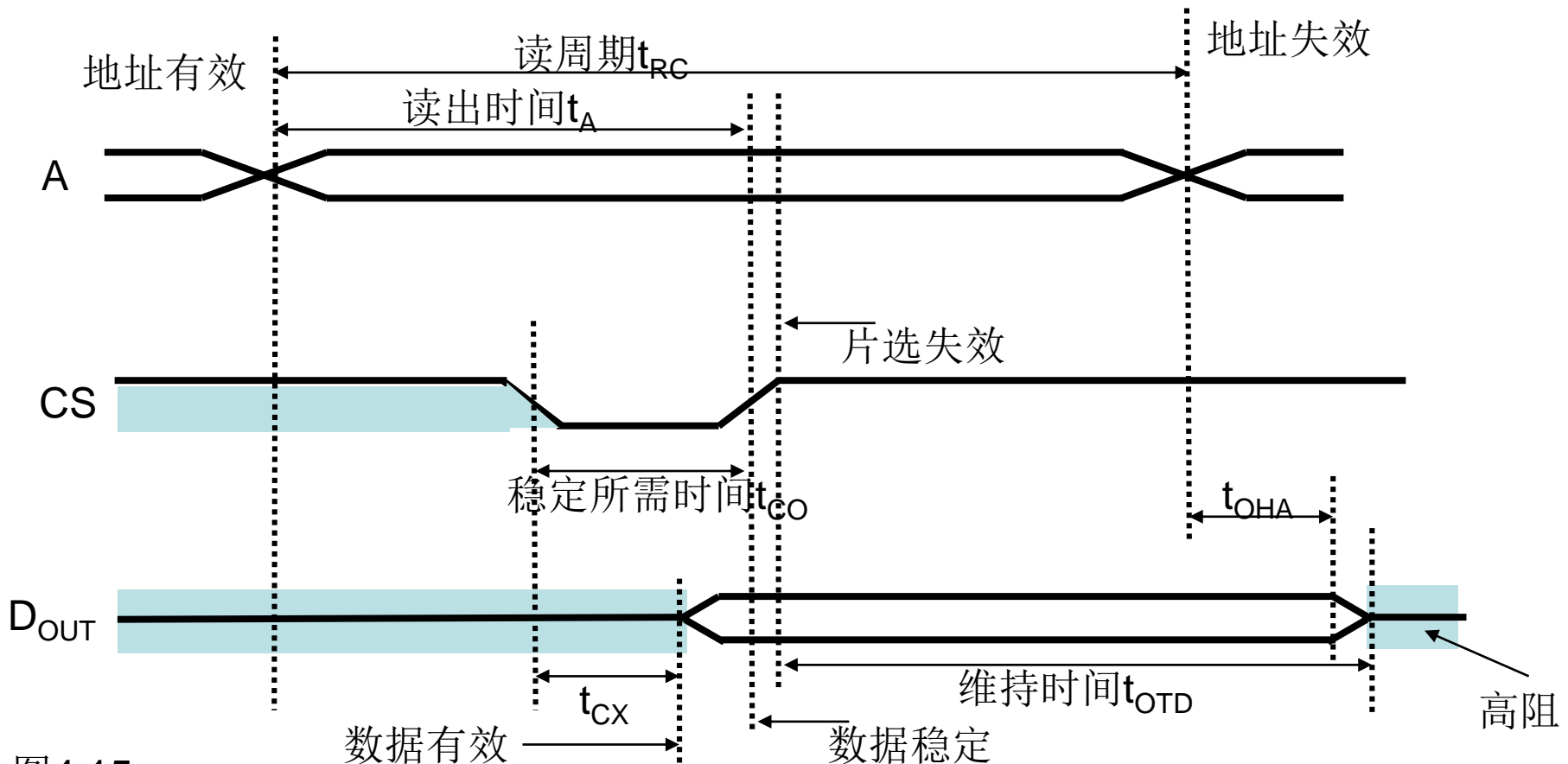


图4.15



# 2114 (SRAM) 时序

## 写周期:

- 地址有效 → CS有效 → 数据有效 → CS复位 (数据输入) → 地址撤销
- $t_{AW}$  滞后时间;  $t_w$  写入时间;  $t_{WR}$  写恢复时间。

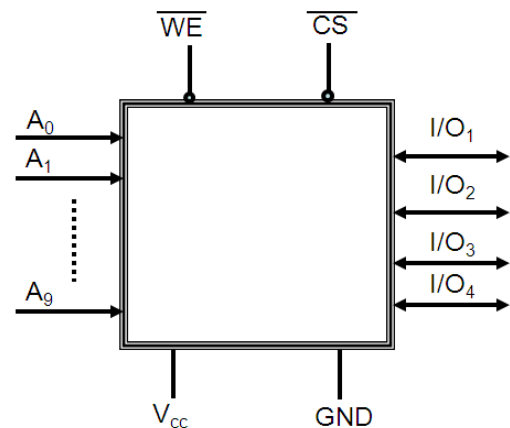
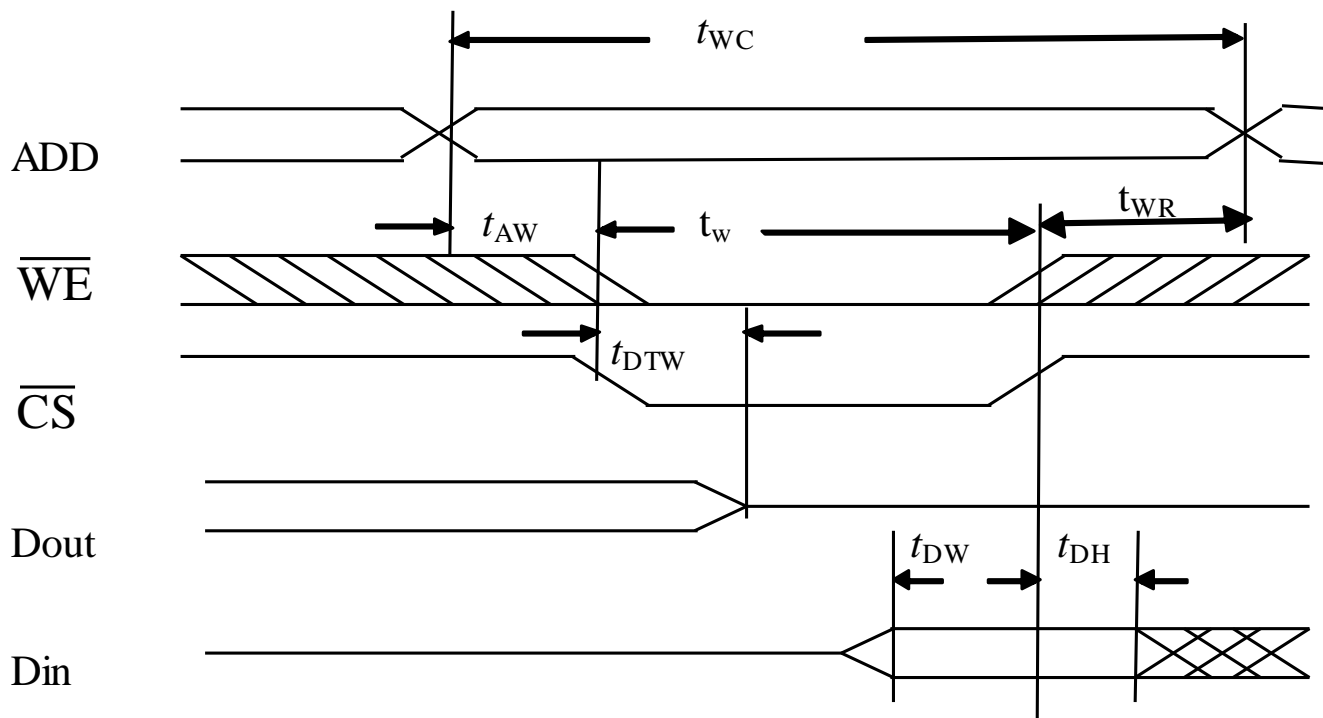


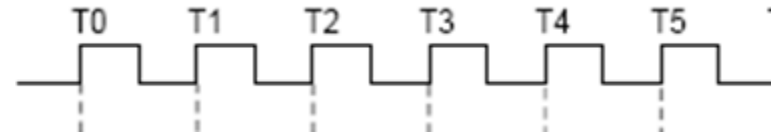
图4.16 异步: 需要精心设计信号时序



# SSRAM: 同步SRAM

- SDR SSRAM: Single Data Rate

- 应用简便: 信号被锁存到寄存器中, 读写过程的延时等待, 被限制在时钟作用间隔内
- 支持突发读写模式
- 在时钟的上升沿传输数据
- 读写操作共用数据总线



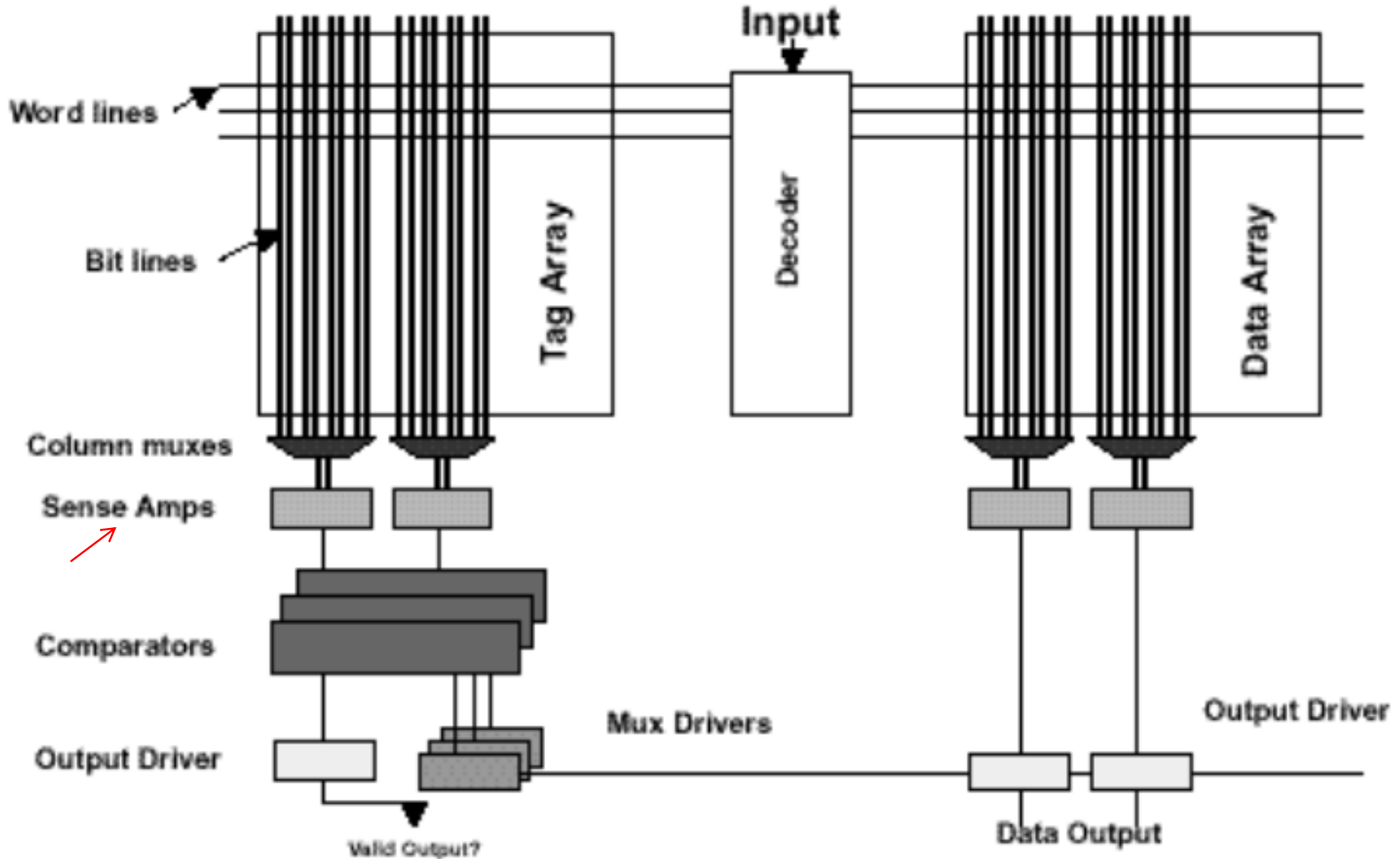
- DDR SRAM

- 在时钟的上升沿和下降沿各传输一次数据

- QDR SRAM

- 读写总线独立: 可同时进行读写

# Cache Memory organization



# 动态RAM(DRAM)



- DRAM靠电容存储电荷的原理寄存信息。
  - 电容上有足够多的电荷表示存“1”，电容上无电荷表示存“0”。
  - 电容上的电荷一般只能维持1~2ms。
  - 再生或刷新：必须在2ms内对其所有存储单元恢复一次原状态。
  - 与SRAM相比，集成度更高、功耗更低。
- 常见的DRAM基本单元电路有**三管式**和**单管式**两种。

# 三管MOS动态RAM



读出时:

T4预充电, 使读数据线为  $V_{DD}$  ( ? )。

读选择打开T2。

若  $C_g$  有电荷, 则T1导通, 数据线降为零电平; 若无电荷, T1截止, 数据线保持高电平。

读出信息与原存信息相反。

写入时:

由写选择线打开T3,  $C_g$  随写入信息充电 (写“1”) 或放电 (写“0”)。

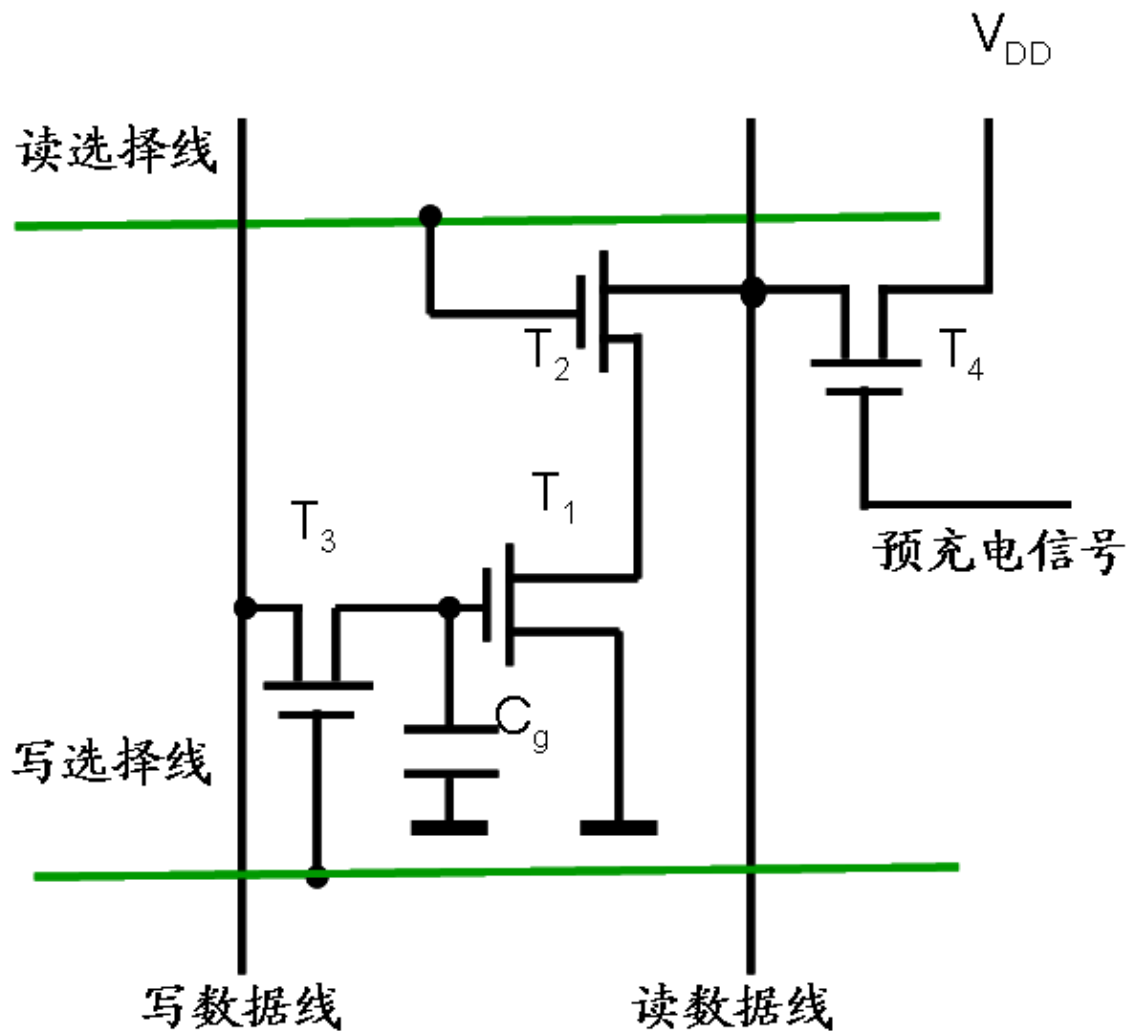


图4.17

# 1KX1位三管MOS动态RAM结构

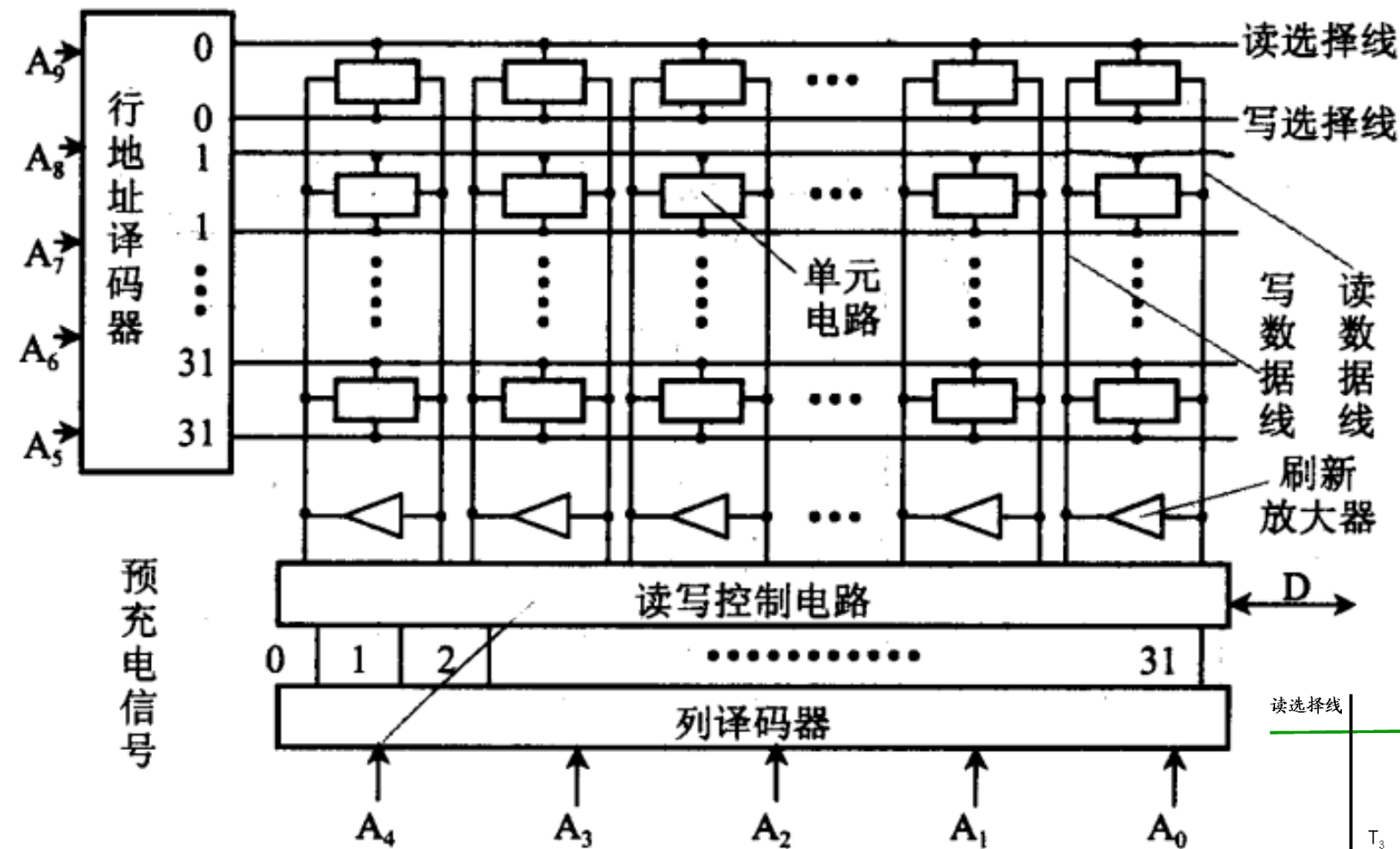
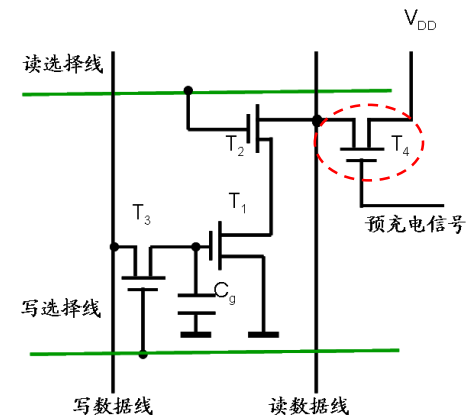


图4.19

$T_4$  每列一个



刷新放大器：行选后，数据输出到读数据线，经刷新放大器，从写数据线重新写入。

# 单管MOS动态RAM



- 读：字线 (wordline) 为高使T导通，若电容 $C_s$ 上有电荷，经T管在数据线 (bitline) 上产生电流，可视为读出“1”。若 $C_s$ 无电荷，则数据线上无电流，可视为读出“0”。

- 读结束时， $C_s$ 的电荷已泄放完毕，故是破坏性读出，必须再生。

- 读出-刷新放大器

- 写：字线为高电平使T导通，若数据线上为高电平，则经T管对 $C_s$ 充电，使其存“1”；若数据线为低电平，则 $C_s$ 经T放电，使其无电荷而存“0”。

- 字线 (wordline) = 行选

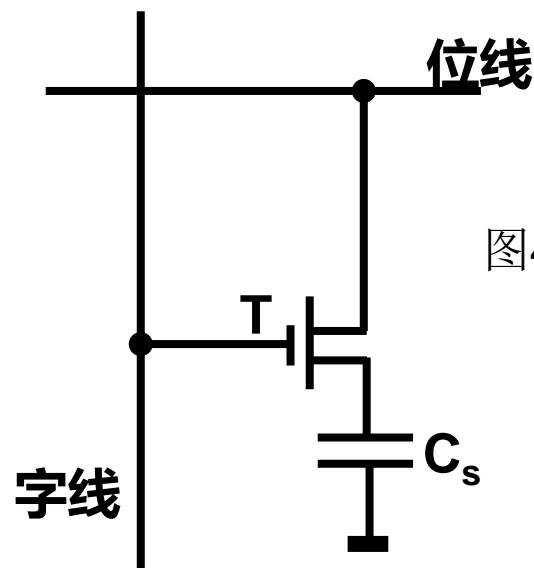
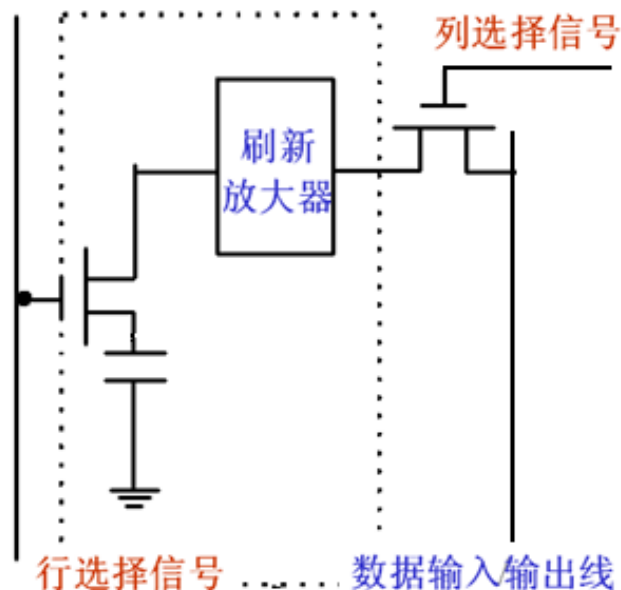


图4.18





# 单管动态RAM芯片—4116芯片存储矩阵

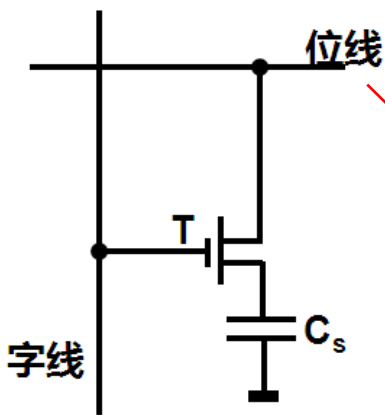


图4.18

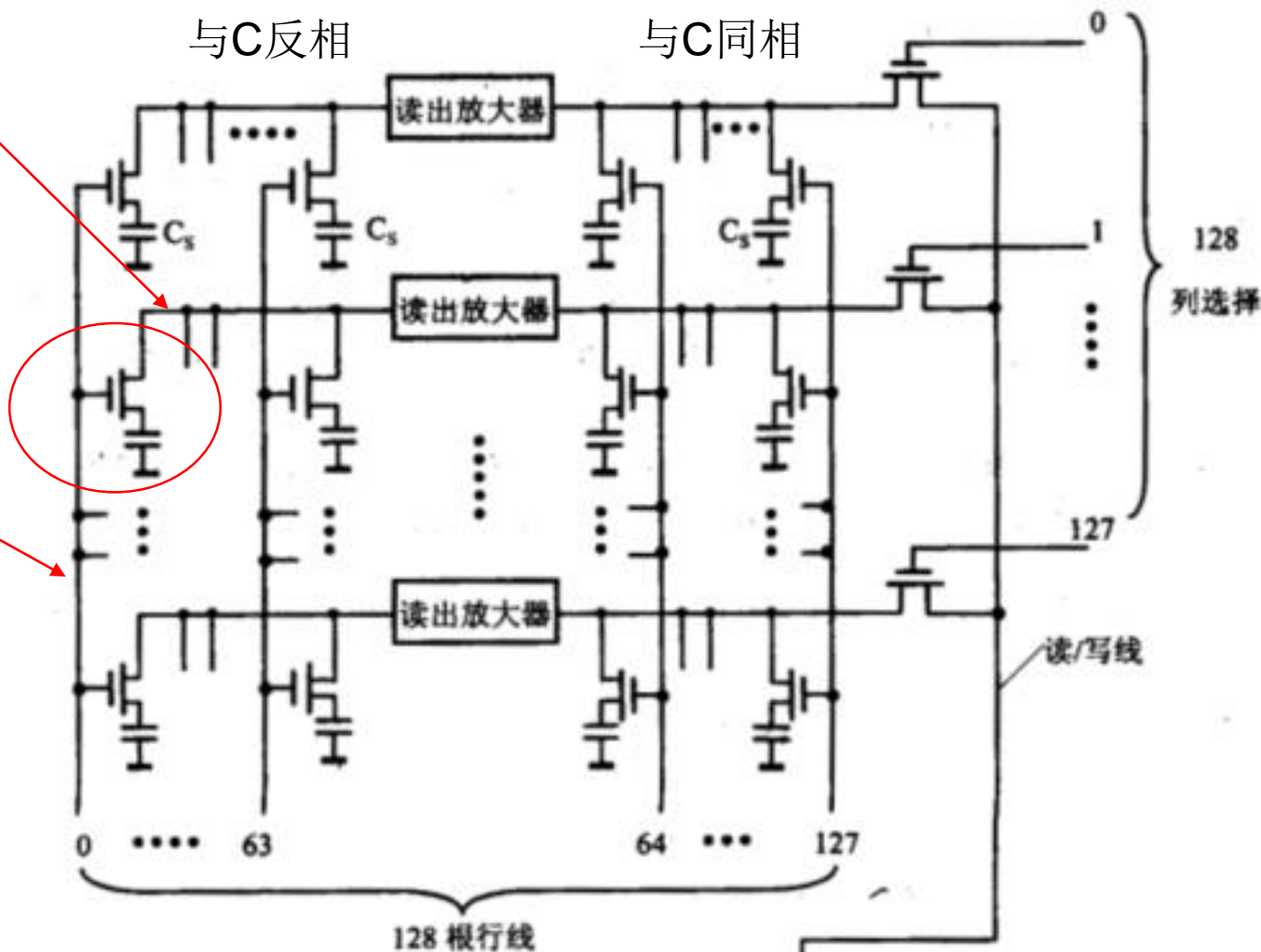


图4.21



# 单管动态RAM芯片—4116芯片结构



- 16KX1位存储芯片；行列地址分开送；128X128存储矩阵。

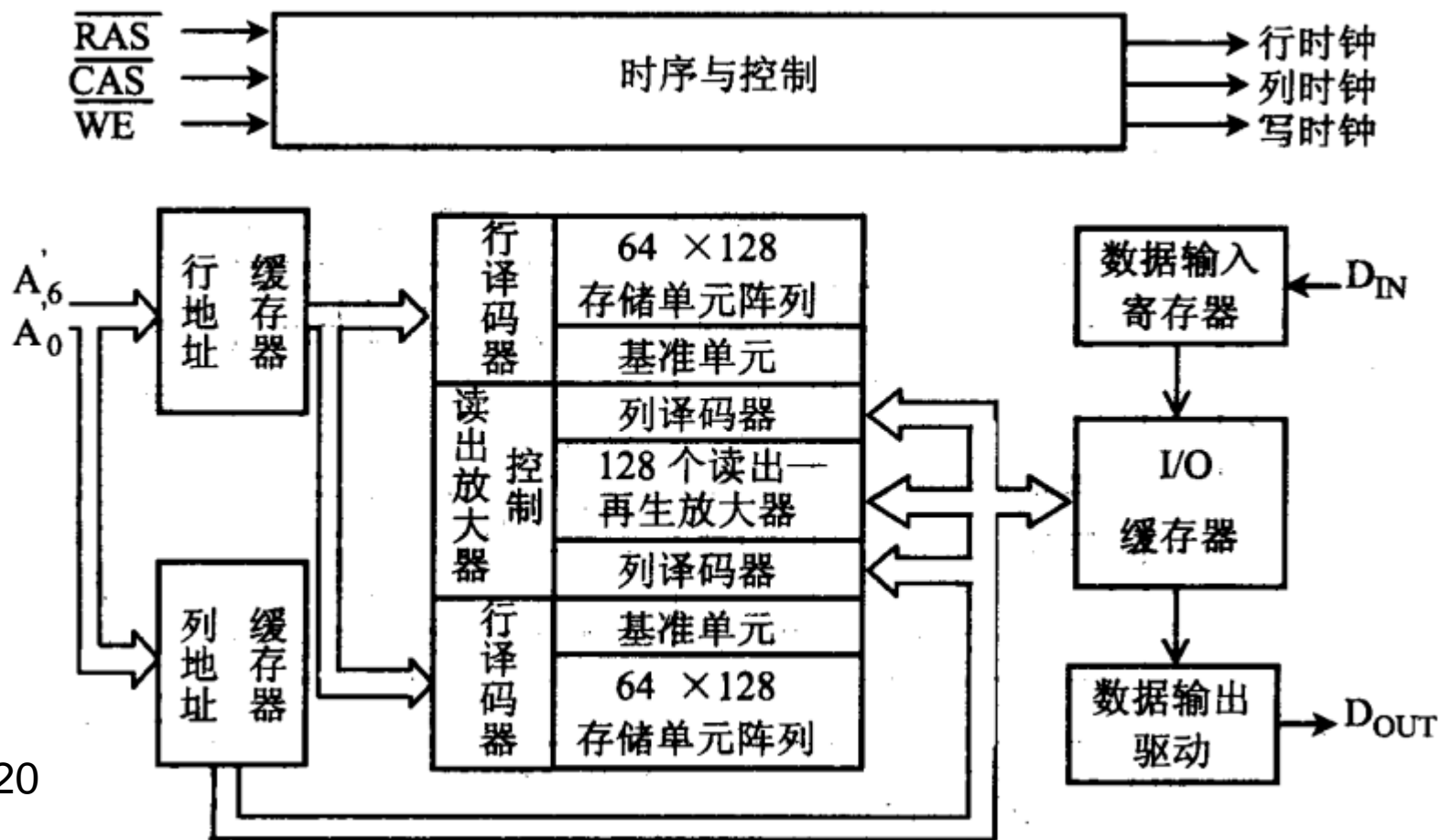
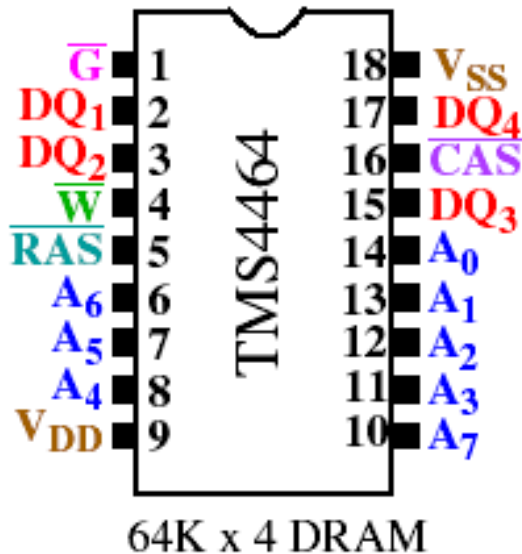


图4.20

# DRAM芯片4464



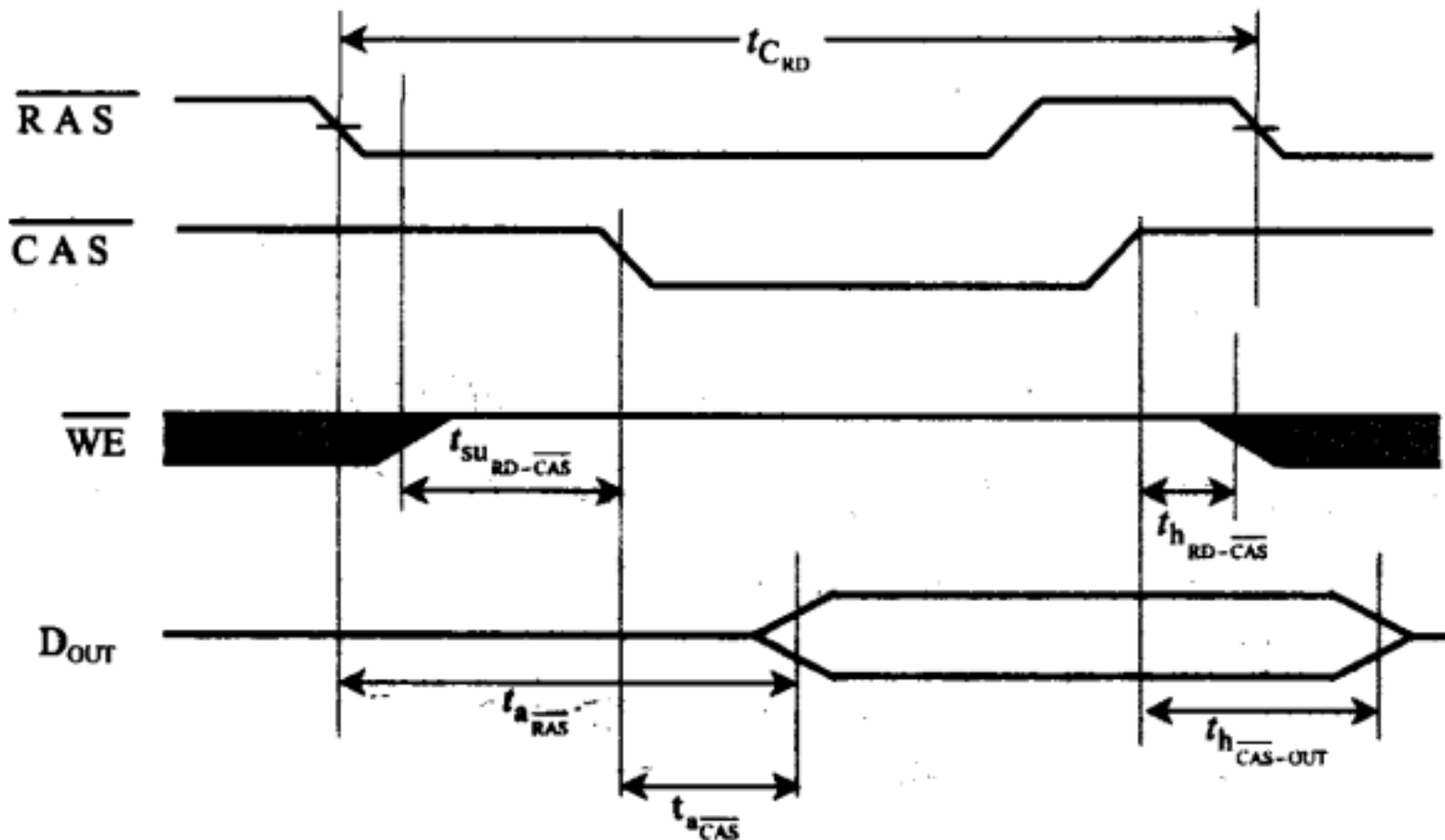
TI TMS4464 DRAM (64K X 4):



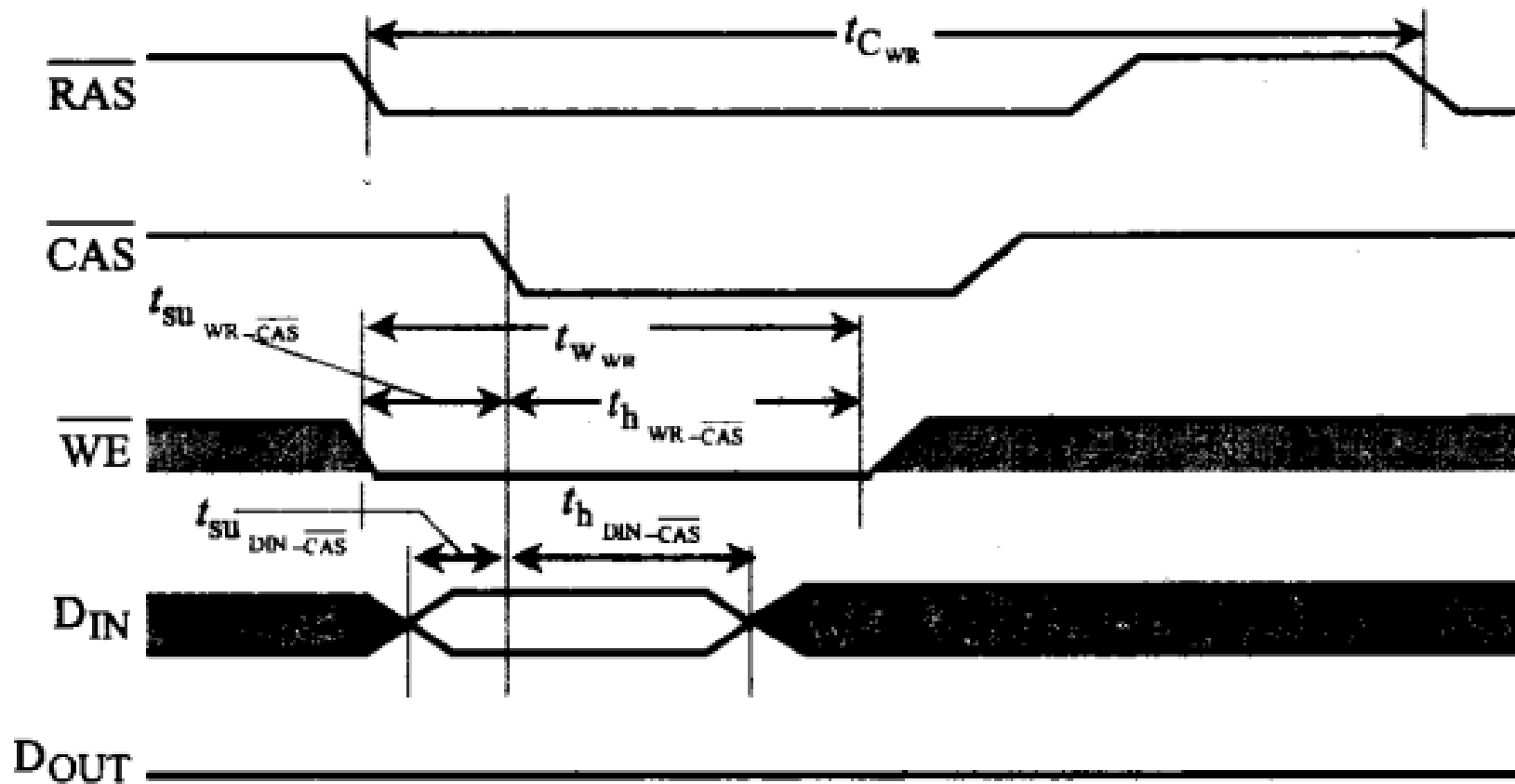
Pin(s)	Function
$A_0-A_7$	Address
$DQ_0-DQ_4$	Data In/Data Out
$\overline{RAS}$	Row Address Strobe
$\overline{CAS}$	Column Address Strobe
$\overline{G}$	Output Enable
$\overline{W}$	Write Enable

- It has **64K** addressable locations which means it needs **16** address inputs, but it has only **8**.
  - The row address ( $A_0$  through  $A_7$ ) are placed on the address pins and strobed into a set of internal latches.
  - The column address ( $A_8$  through  $A_{15}$ ) is then strobed in using  $\overline{CAS}$ .

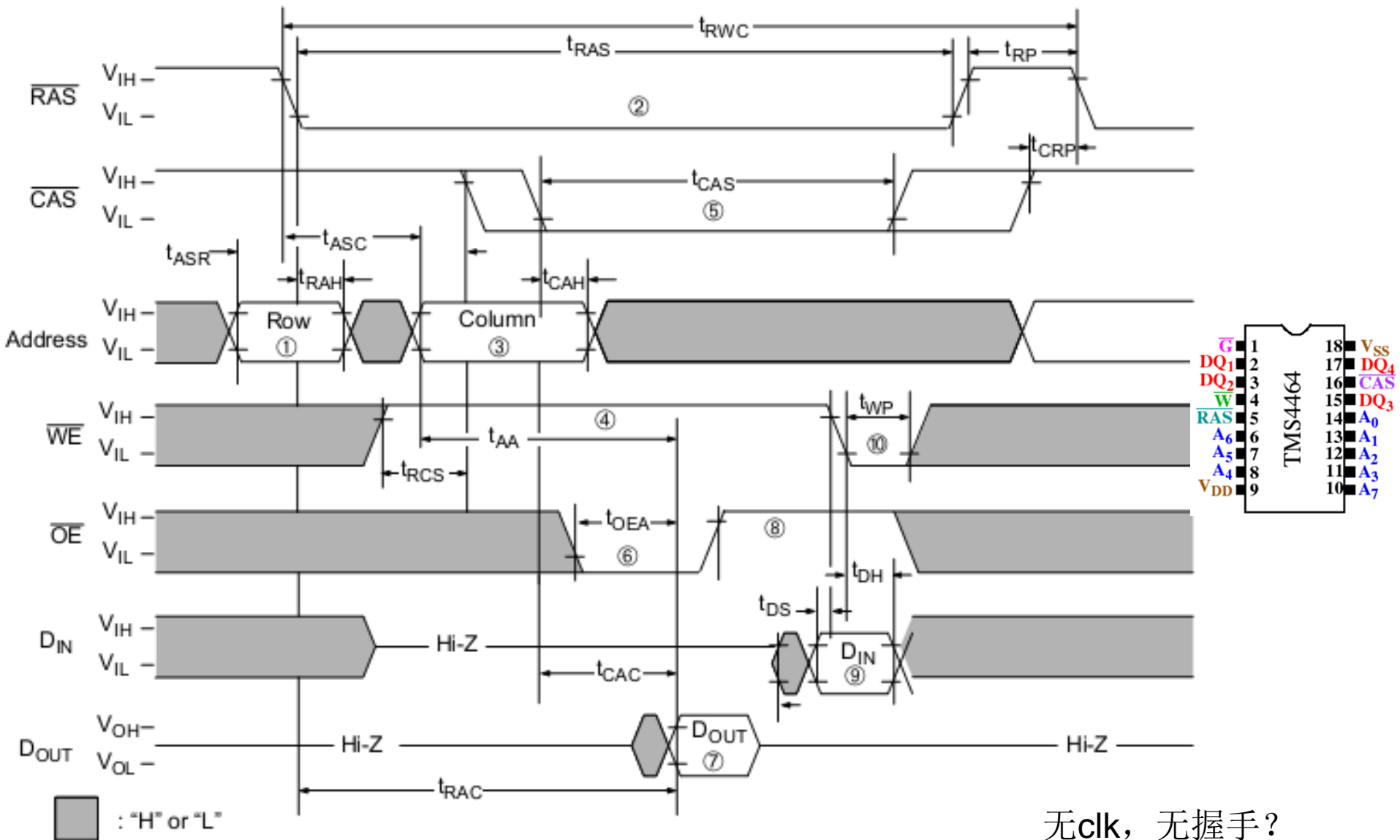
# 动态RAM读时序 (图4.22)



# 动态RAM写时序

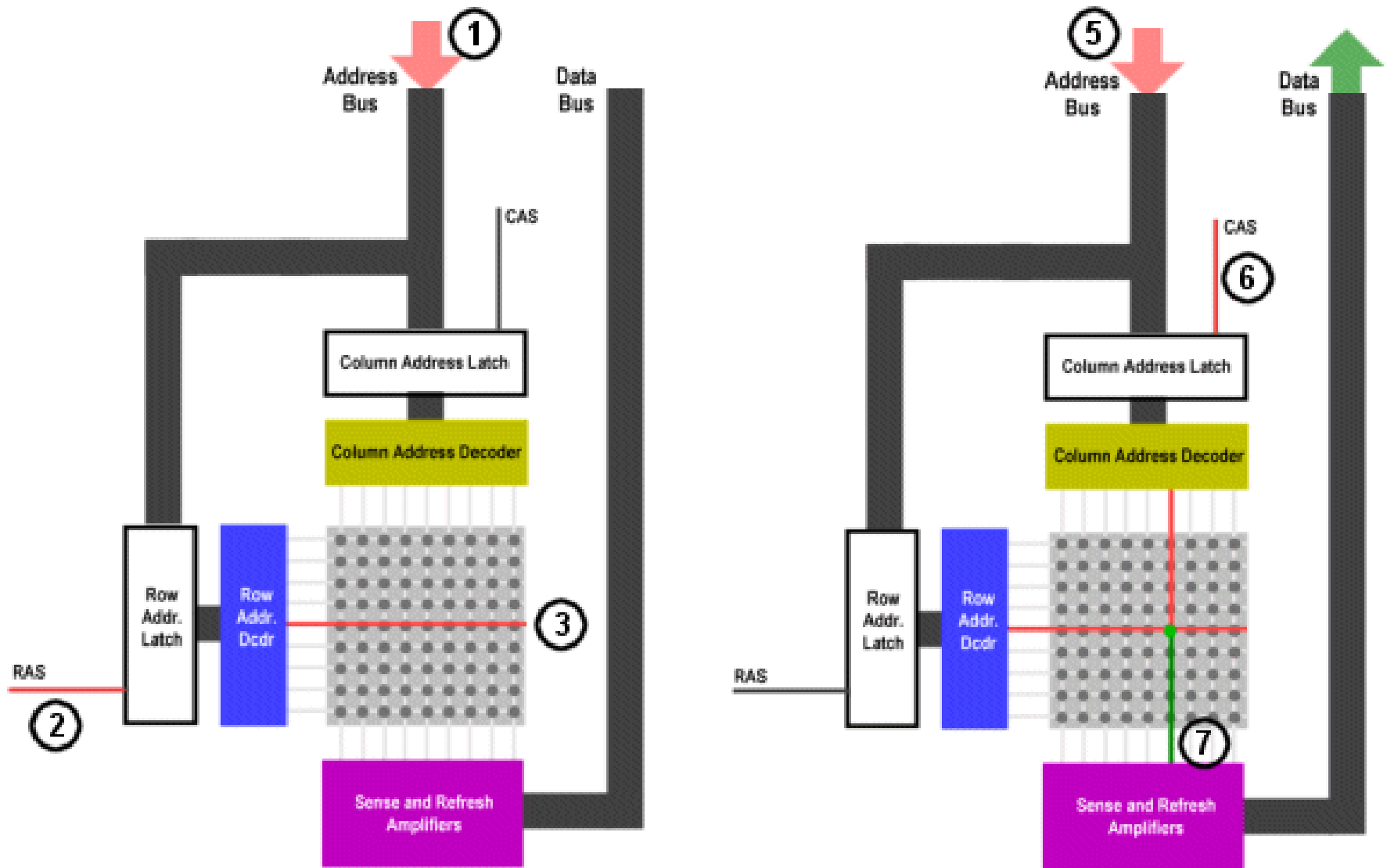


# Simplified Read-Modify-Write Cycle



无clk, 无握手?

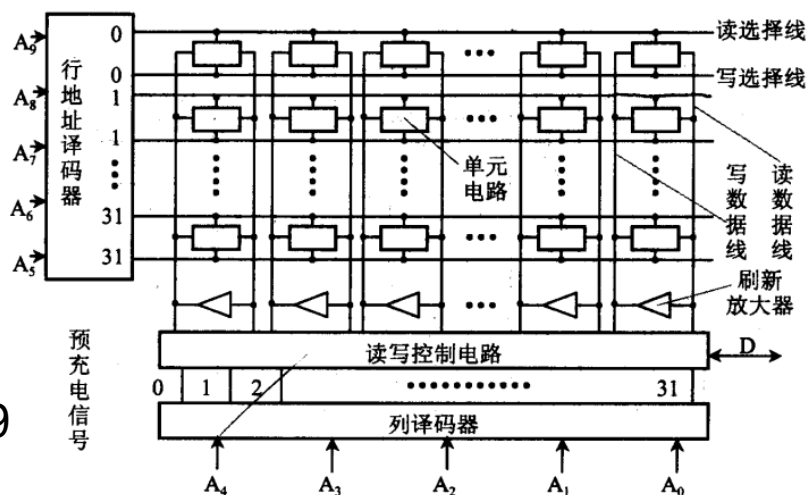
# How DRAM Works (读1位, 重合译码)



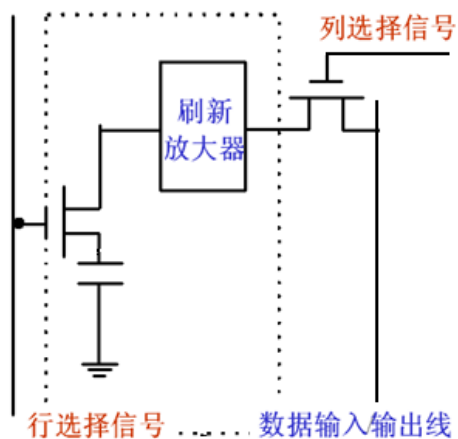
按上页“RMW周期”时序，第4步为WE。Sense & Refresh amplifier=RowBuf

# 动态RAM的刷新, 唐\$4.2.3

- 刷新过程
  - “读出 - 写回”：由刷新放大器（读出放大器）协助
- 刷新时机：“按位则没有时间进行正常访存！” ——见COD
  - 读写操作时：选通一行，即进行该行的刷新！
  - 刷新周期（**再生周期**）：停止正常读写，**按行刷新**。
- 刷新周期刷新方式：
  - 集中刷新：在一个刷新周期（**2ms**）内，对所有行刷新。——死区
  - 分散刷新：每个**存取周期**刷新一行。——过密
  - 异步刷新：每隔一定间隔刷新一行。——将死区分散



唐图4.19



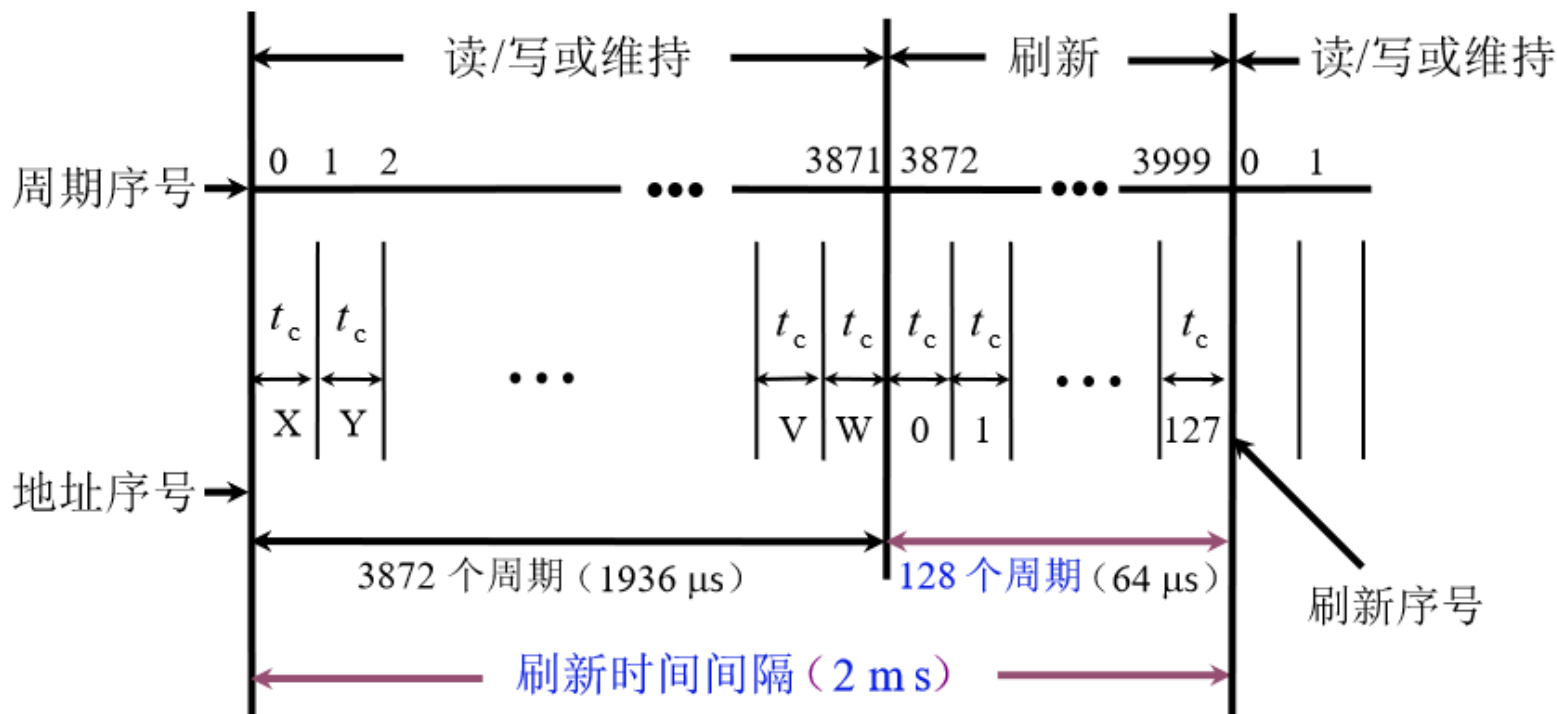
唐图4.18





# 动态RAM的刷新：集中式刷新时间分配图

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



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

唐图4.24

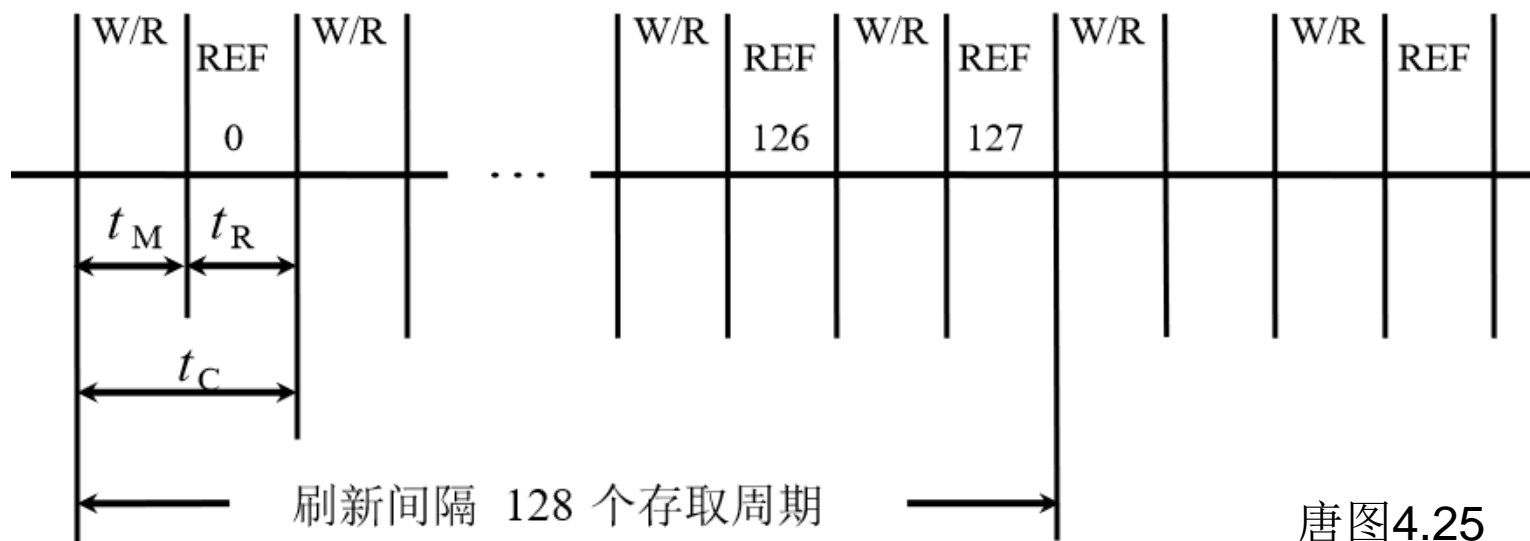
“死时间率”为  $128/4\ 000 \times 100\% = 3.2\%$

PC133 标准：每次刷新用时= 9 个时钟周期



# 动态RAM的刷新：分散刷新

② 分散刷新（存取周期为 $1\ \mu\text{s}$ ）刷新被分散到每个存取周期  
以  $128 \times 128$  矩阵为例



唐图4.25

$$t_C = t_M + t_R$$



读写 刷新

无“死区” 但存取周期延长1倍

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

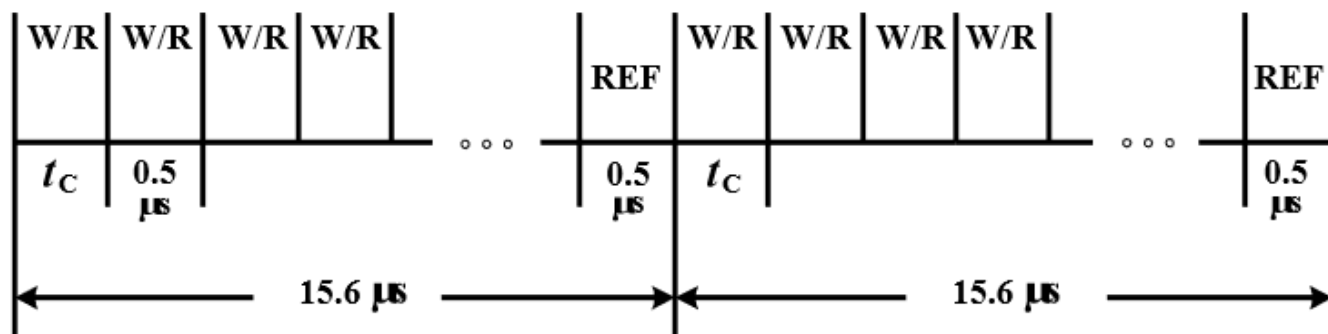


# 动态RAM的刷新：异步刷新

③ 分散刷新与集中刷新相结合（异步刷新）

对于  $128 \times 128$  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



唐图4.26

每行每隔  $2 \text{ ms}$  刷新一次

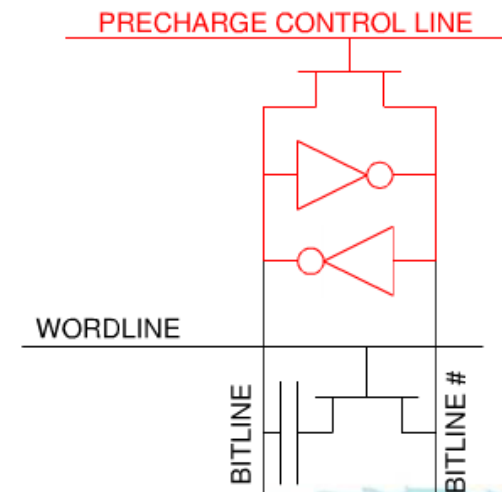
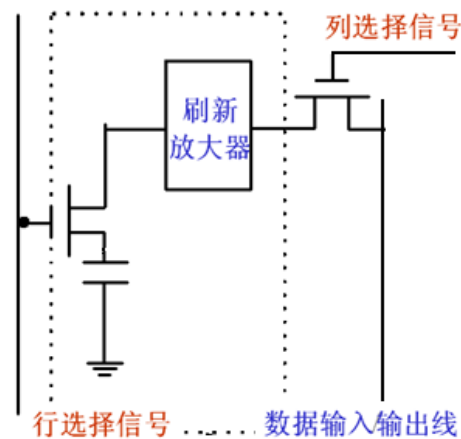
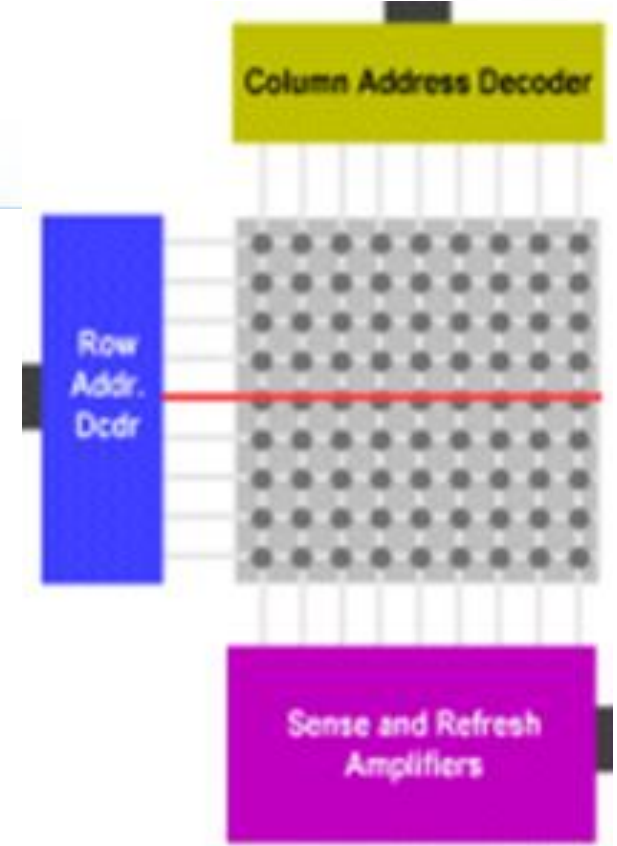
“死区”为  $0.5 \mu\text{s}$

“异步”？

将刷新安排在指令译码阶段，不会出现“死区”

# Sense Amplifiers

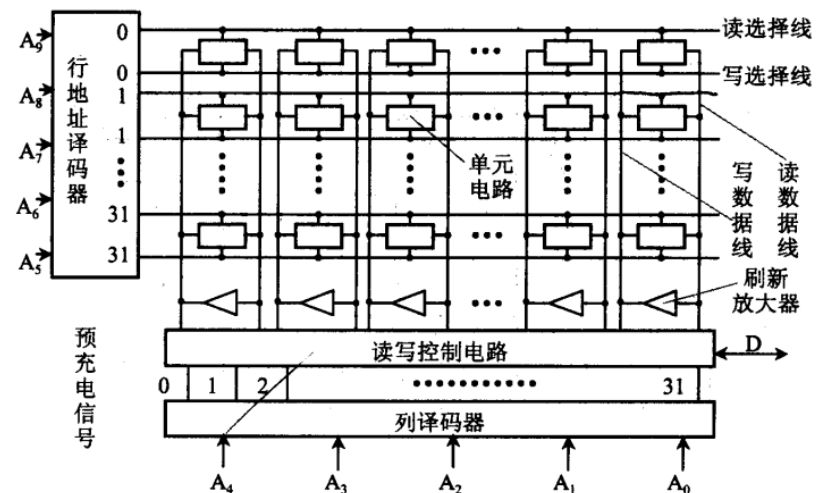
- 检测，放大
  - 当接入晶体管导通且存储电容器将其电荷置于位线上时电压的微小变化，并放大至数字信号“1”或“0”。
- Precharge: 预充电
- recharge: 再生
  - 单管为破坏性读
- Refresh: 刷新
- row buffer





# 动态RAM与静态RAM的比较

- DRAM的应用比SRAM要广泛 (主存)
  - DRAM集成度高、功耗为SRAM的1/6、价格为其1/4
  - DRAM由于使用动态元件 (电容), 速度比SRAM低
  - 单管DRAM需位线预充电和cell再生
- DRAM所涉及的操作 (命令)
  - 位线预充电 (Precharge)
  - cell再生 (recharge, restore)
  - 行选通 (RAS, )
    - 行激活Active
  - CAS
  - 读/写 (WE、OE)
  - 刷新 (refresh) : 2 ~ 64ms



# 内容



## 1. 主存储器的基本组成和技术指标



## 3. 存储容量的扩展、存储器与CPU的连接

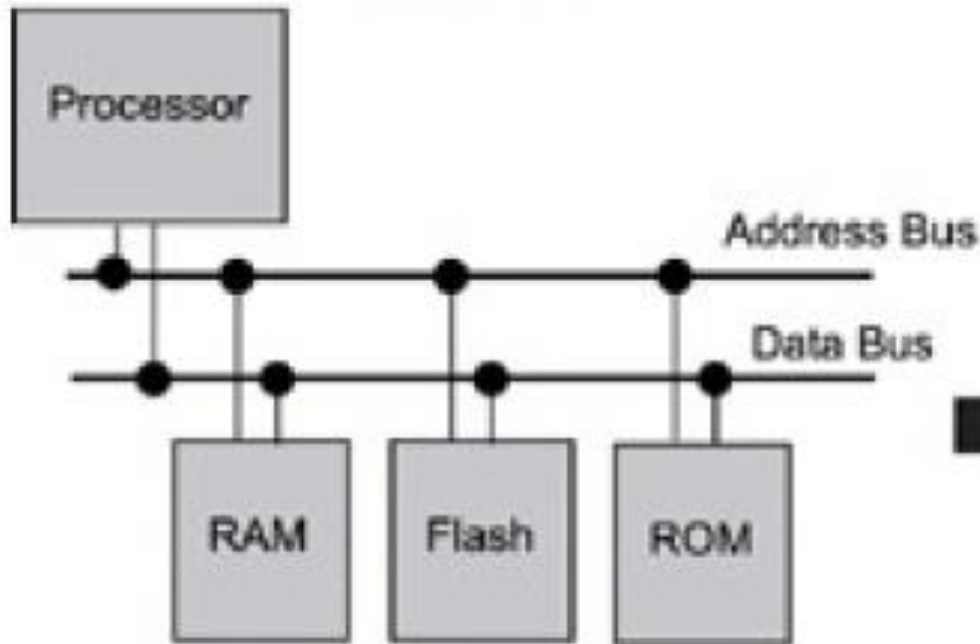
## 4. 存储器的校验—海明码

## 5. 提高访存速度的措施

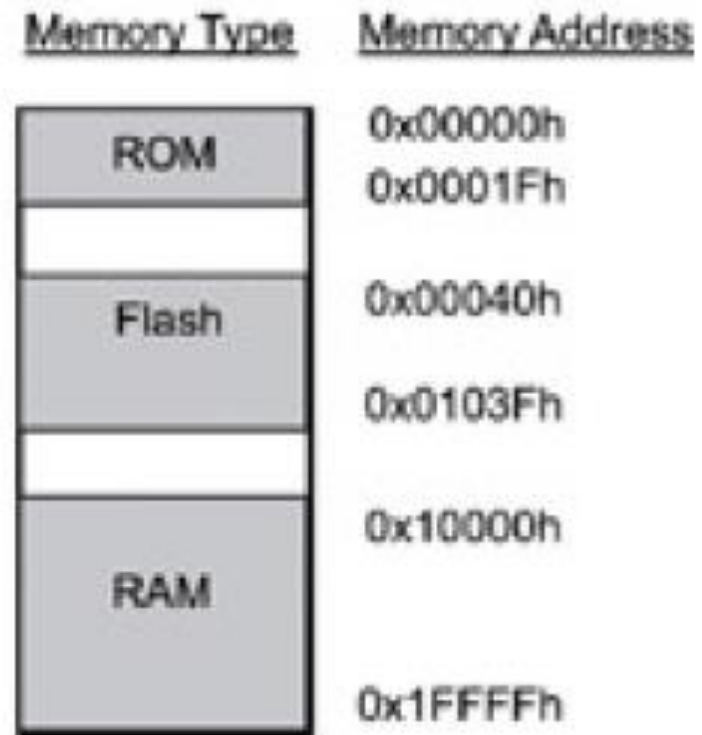
# 存储容量，地址空间



Schematic



Memory Map



- 内存容量，每一类容量？
- 需要多个芯片构成？
- 地址总线，数据总线？



# 存储容量的扩展

- **存储容量的扩展**：当一片RAM存储芯片不能满足存储容量需要时，就需要将若干片RAM存储芯片组合起来，构成满足存储容量要求的存储器。
- 三种扩展方法：
  1. **位扩展**：增加存储字长
  2. **字扩展**：增加存储字的数量
  3. **字位扩展**
- **存储芯片的连接**，主要完成三种线的连接
  1. 地址线 $A_0 \sim A_n$ 的连接 (**片内**, **片外**/片选)
  2. 数据线 $I/O_0 \sim I/O_n$  或  $D_0 \sim D_n$ 的连接
  3. 控制线的连接, 如片选 $CS$ 、读/写 $\overline{WE}$





# 位扩展法

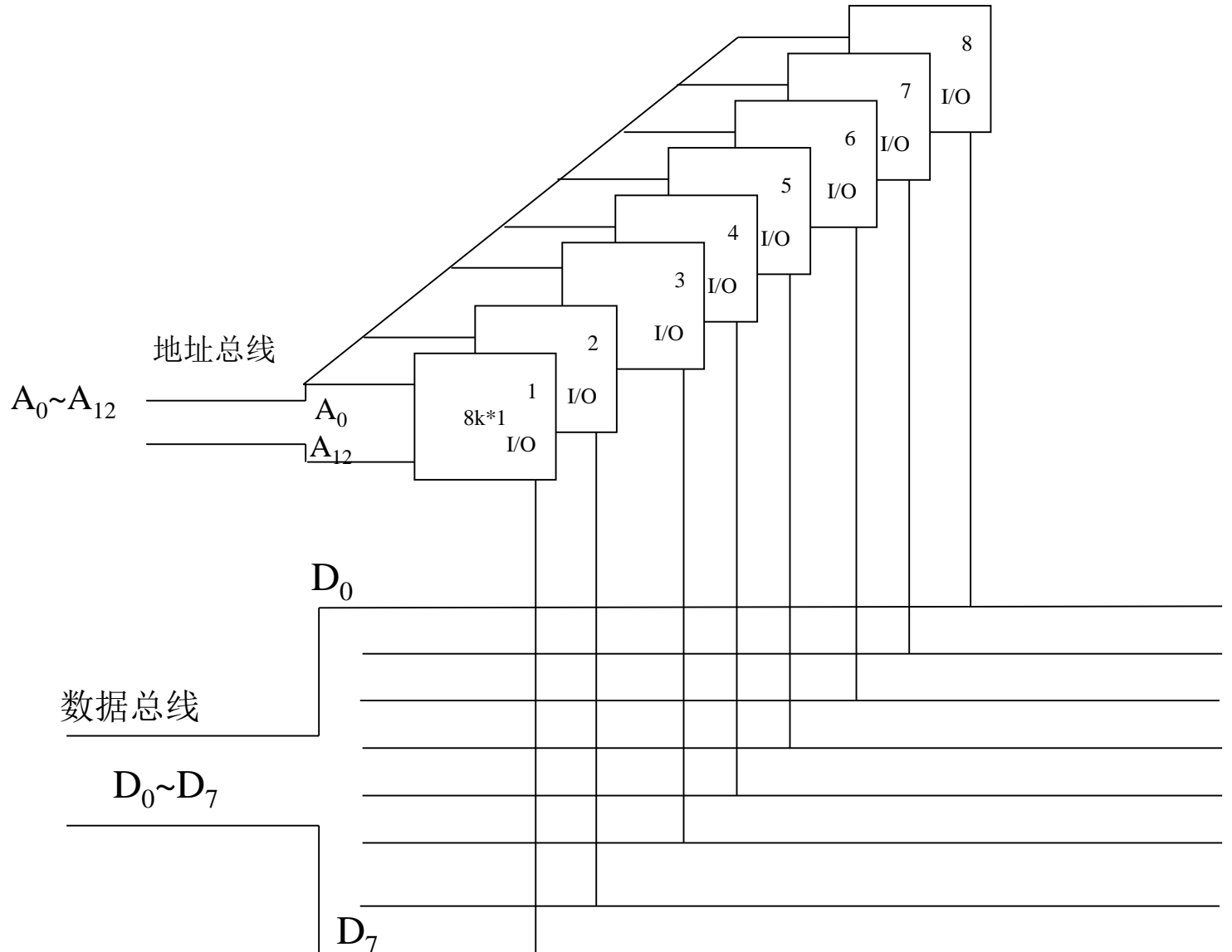
- **仅在字长（位数）扩展，字数不做扩展。**
  - 字数满足要求，而位数不够时，应采用位扩展。

**例：使用8K\*1位的RAM构成8K\*8位的存储器。**

分三步：

- (1) 使用8片8K\*1的RAM芯片，每片有13根 ( $A_0 \sim A_{12}$ ) 地址线，1根 (I/O) 数据线
- (2) 每片RAM的1位数据线分别接置数据总线的相应位 ( $D_0 \sim D_7$ )
- (3) 将8片地址线的相应位 ( $A_0 \sim A_{12}$ ) 并联后接至地址总线的相应位上。

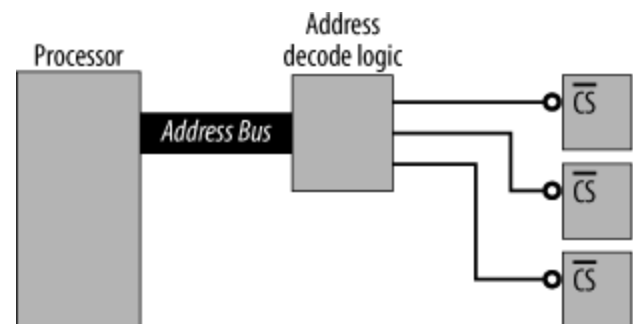
# 由8K\*1位的RAM构成8K\*8位的存储器—连接图



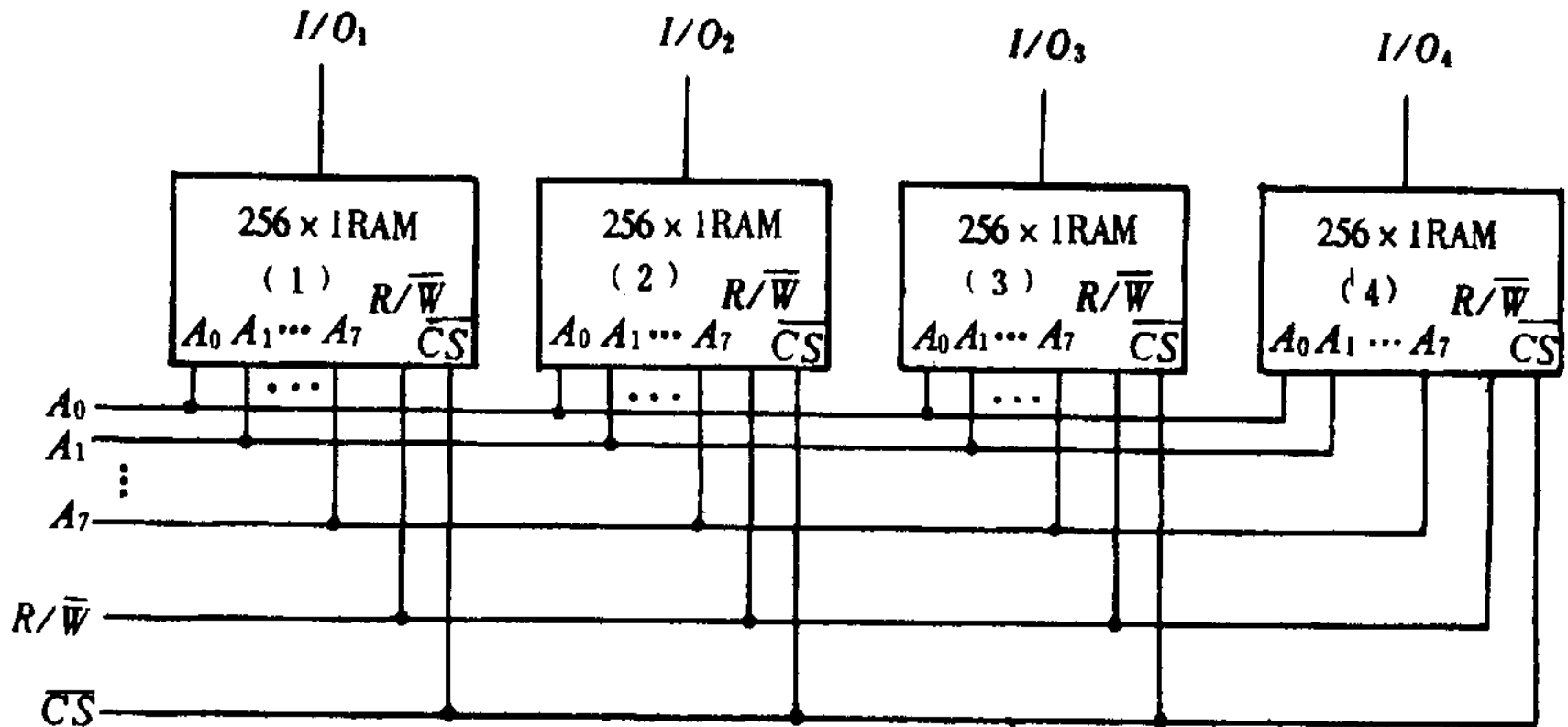


# 位扩展的原则

- 实现位扩展的原则是：
  - ①多个单片RAM的I/O端并行输出。
  - ②多个RAM的  $\overline{CS}$  端接到一起，作为RAM的片选端（同时被选中）；
  - ③地址端对应接到一起，作为RAM的地址输入端。
  - ④多个单片RAM的  $\overline{WE}$  端接到一起，作为RAM的读/写控制端（读/写控制端只能有一个）；



# 用4片256×1位的RAM扩展成256×4位的RAM的接线图

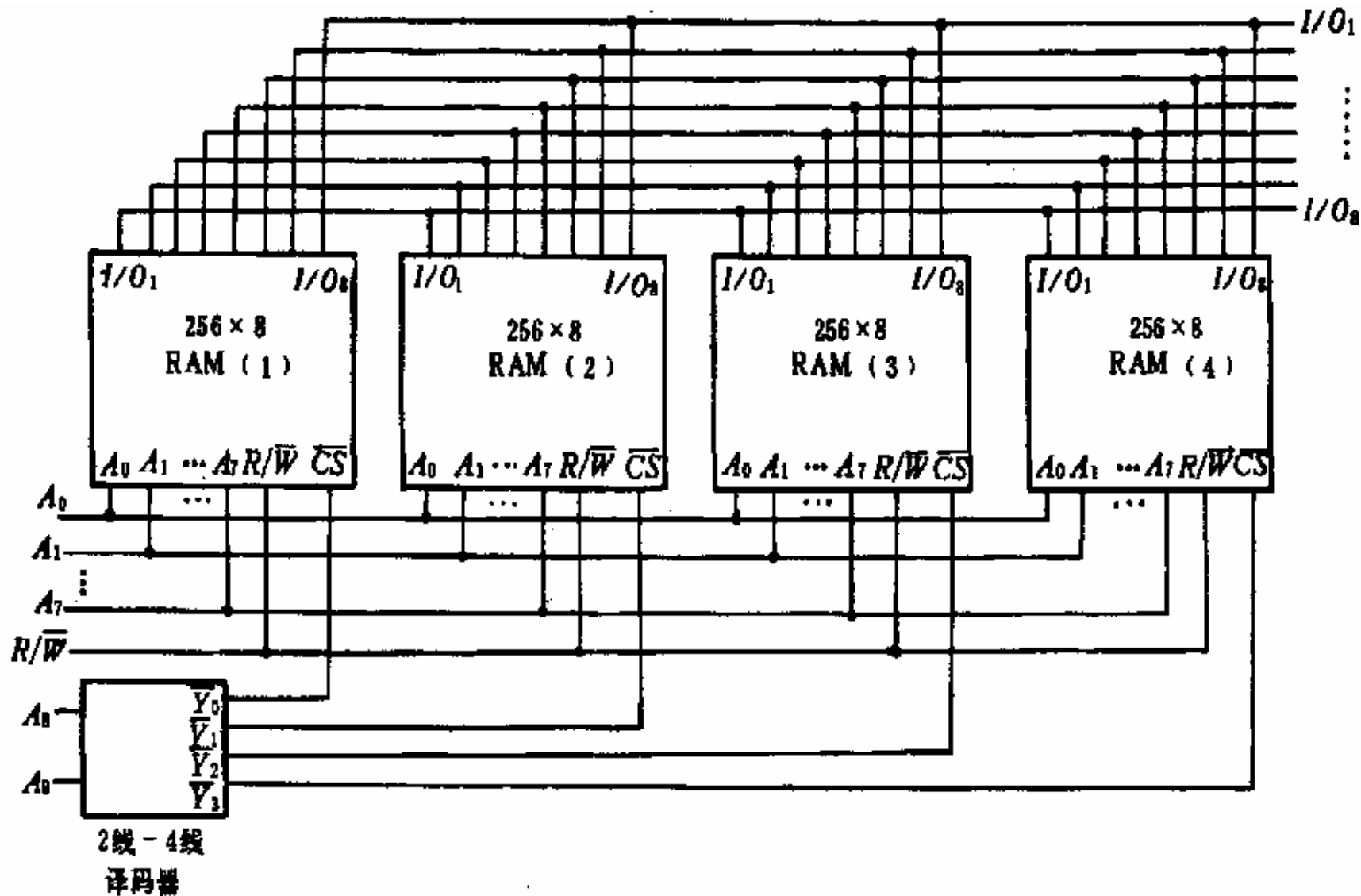




# 字扩展法

- **只在字向扩充，而位数不变。**
  - 字数增加，**地址线**数就得相应增加。
- **例：用 $256 \times 8$ 位RAM构成 $1024 \times 8$ 位RAM**
  - $256 \times 8$ 位RAM的地址线数为8条，而 $1024 \times 8$ 位RAM的地址线数为10条

# 字扩展法示意



# 字扩展法

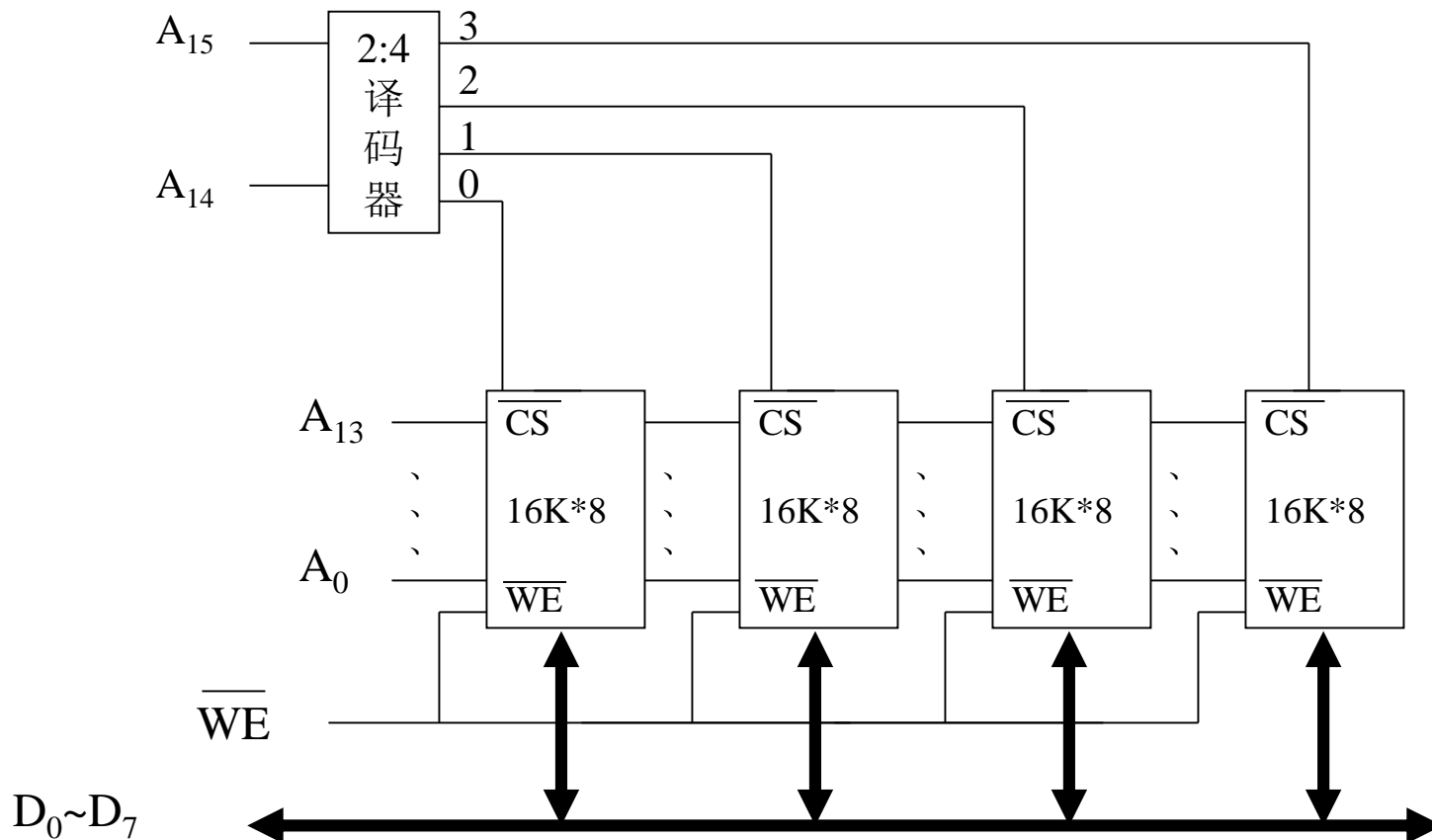


**例如：用16K\*8的存储芯片组成64K\*8位的存储器，并给出片选线CS和读写控制线WE的连接方法。**

连接方法：使用字扩展法

- (1) 使用4片16K\*8的存储芯片来组成
- (2) 每片有地址端14根 ( $A_0 \sim A_{13}$ )，数据端8根  $D_0 \sim D_7$ ，及片选  $\overline{CS}$ ，写允许  $\overline{WE}$  等引脚
- (3) 芯片数据端与数据总线  $D_0 \sim D_7$  相连
- (4) 地址总线的低位地址  $A_0 \sim A_{13}$  与各芯片的14位地址端相连，两位高位地址  $A_{14}, A_{15}$  经2-4译码器译码，4个输出分别与4个片选端  $\overline{CS}$  相连
- (5) 系统总线的  $\overline{WE}$  与各芯片的  $\overline{WE}$  相连

# 用16K\*8存储芯片组成64K\*8位存储器—连接图





# 各芯片地址分配见下表



## 地址空间分配表

地址 片号	$A_{15}A_{14}$	$A_{13}A_{12}A_{11}\dots A_1A_0$	说明
1	00 00	000...00 111...11	最低地址 0000H 最高地址 3FFFH
2	01 01	000...00 111...11	最低地址 4000H 最高地址 7FFFH
3	10 10	000..00 111...11	最低地址 8000H 最高地址 0BFFFH
4	11 11	000...00 111...11	最低地址 0C000H 最高地址 0FFFFH



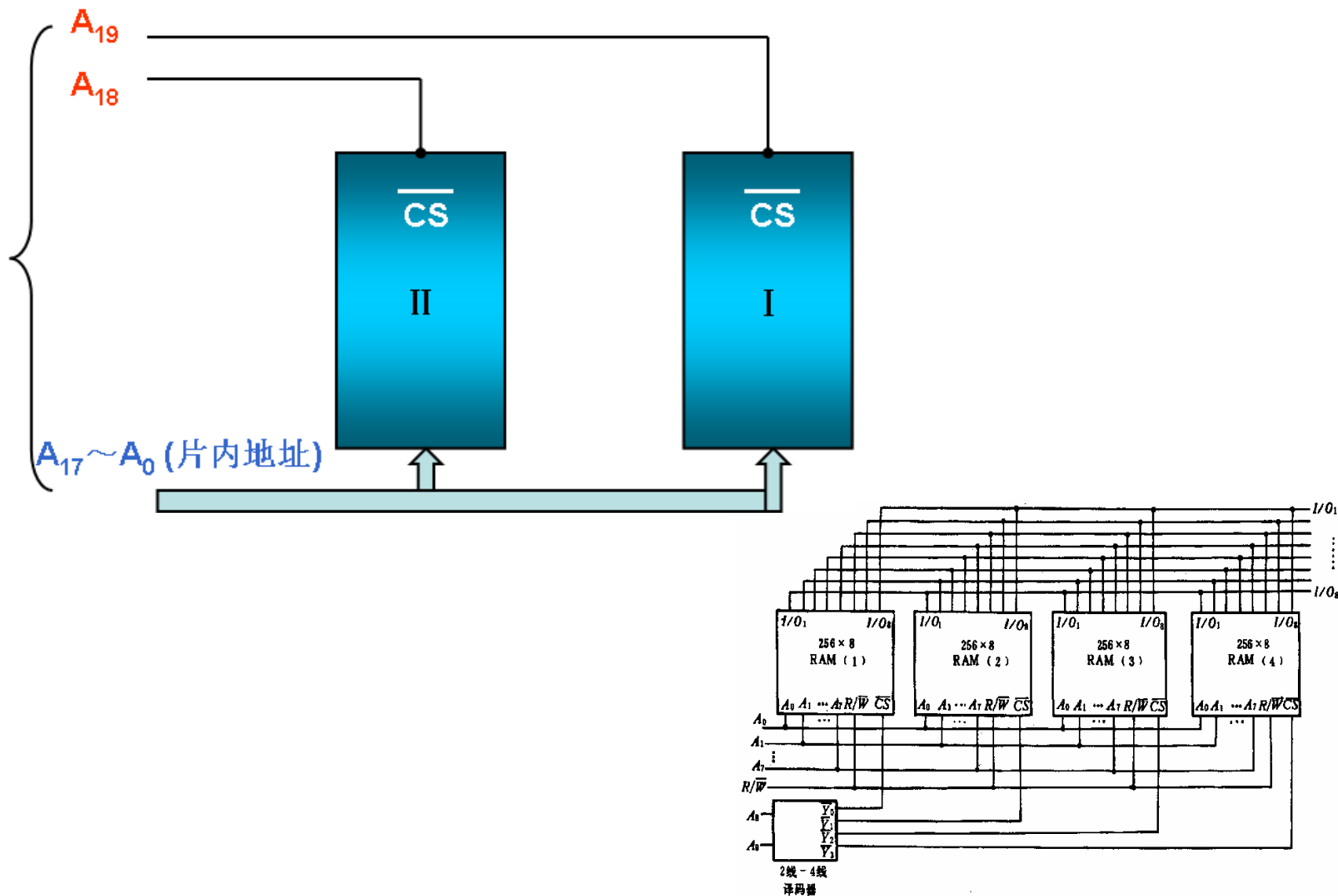
# 字扩展的原则

- 多个单片RAM的I/O端并接，作为RAM的I/O端
- 多片构成字扩展之后，每次访问只能选中一片，选中哪一片，由字扩展后多出的地址线决定。多出的地址线经输出低有效的译码器译码，接至各片RAM的  $\overline{CS}$  端；
- 低位地址端对应接到一起，作为低位地址输入端。
- $\overline{WE}$  端接到一起作为RAM的读/写控制端（读写控制端只能有一个）；

# 线选方式实现片选控制示例

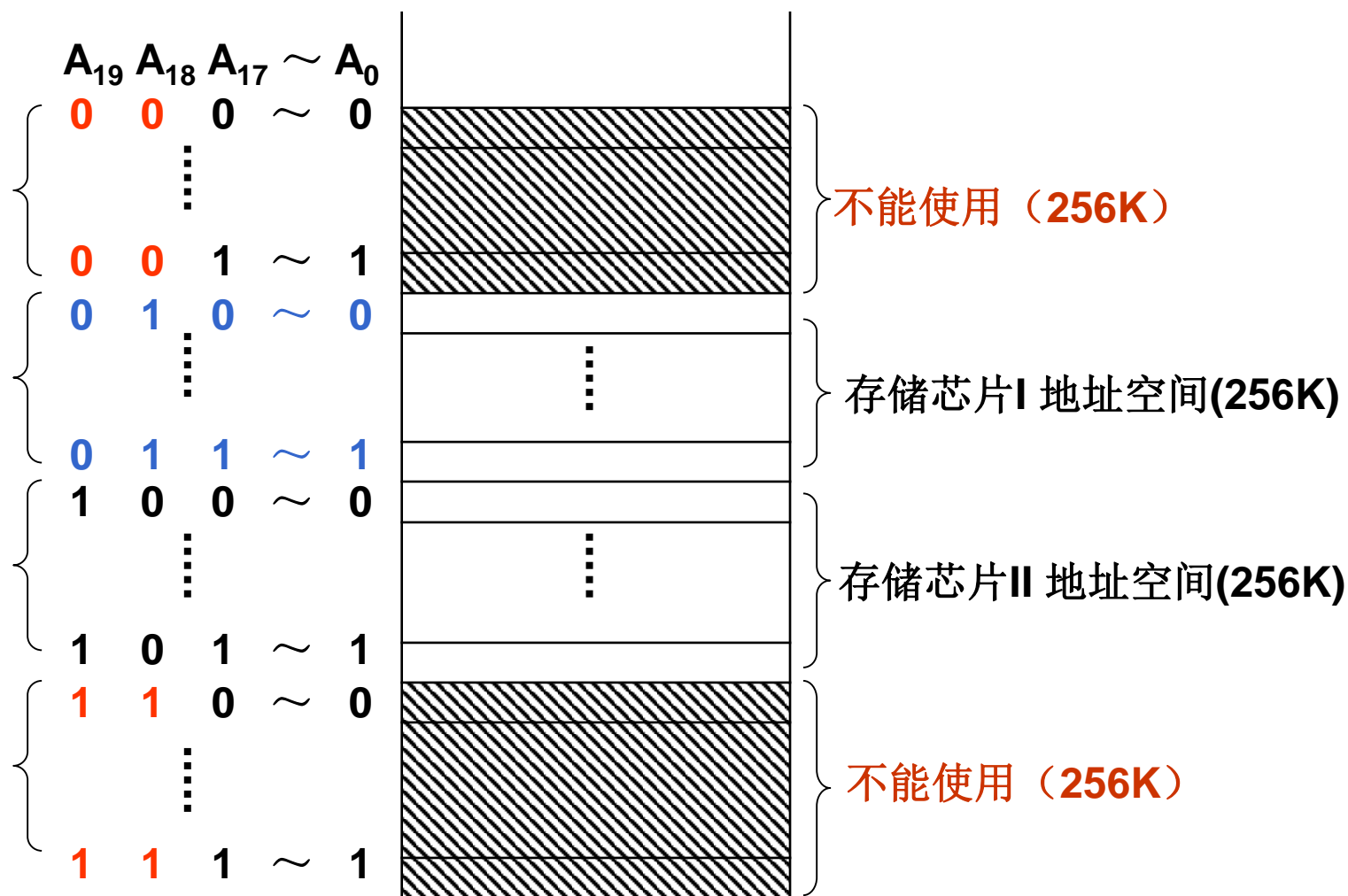


地址总线





## 线选方式的地址空间分布





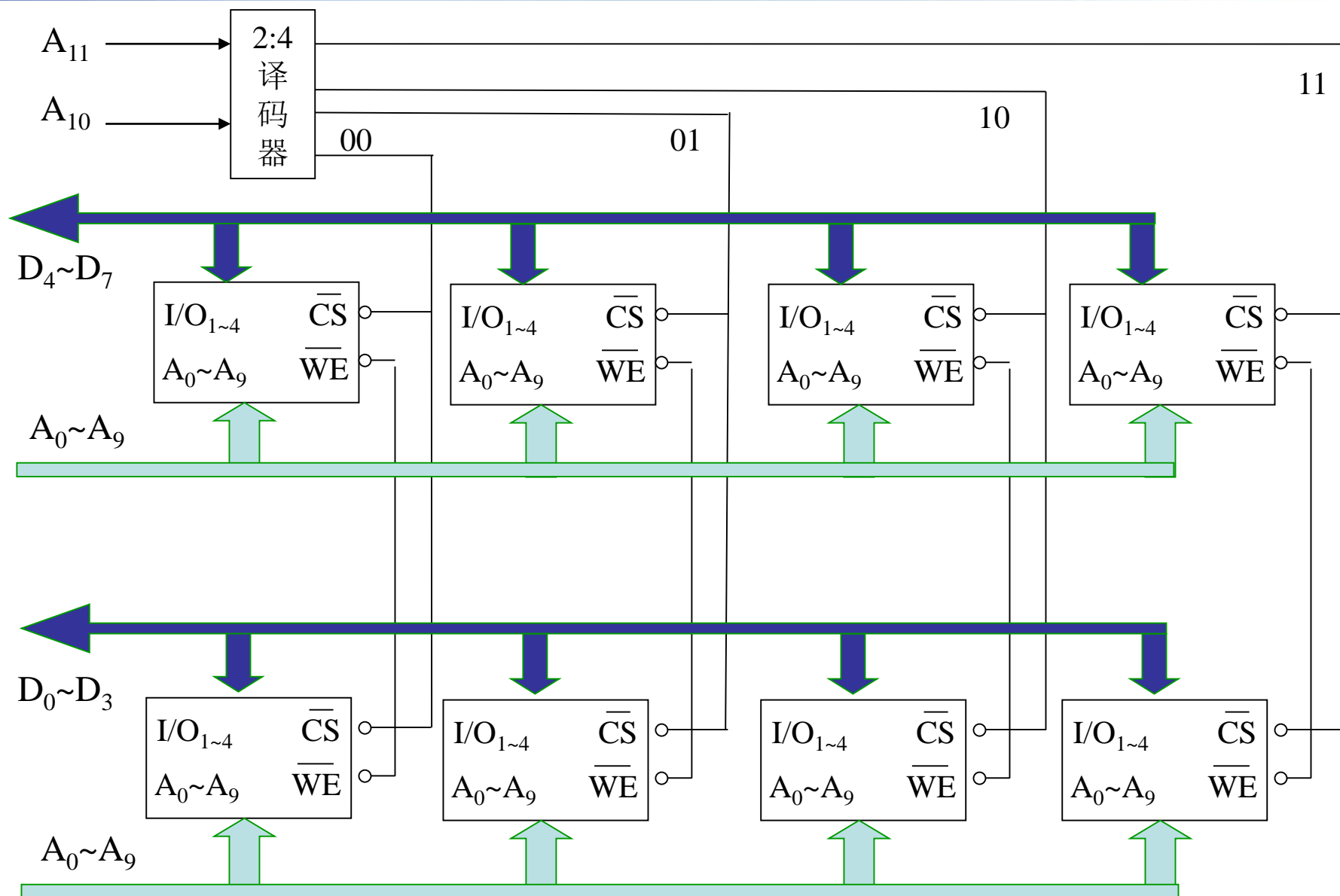
# 字、位同时扩展

假定存储器的容量为 $M*N$ 位,

- 若使用 $L*K$ 芯片 ( $L<M, K<N$ ) , 共需要  $(M/L) * (N/K)$ 个存储芯片, 要在字与位同时扩展。

例: 用2114SRAM构成 $4K*8$ 位的存储器模块,  
2114SRAM芯片是 $1K*4$ 位芯片, 有10根  
( $A_0 \sim A_7$ )地址端, 4根 ( $I/O_{1 \sim 4}$ ) 数据端。

# 由1K\*4SRAM构成的4K\*8存储器模块连接图





# 连接方法分析

1. 使8片2114SRAM芯片 ( $4/1 \times 8/4 = 8$ ) 。
2. 见图：将上面4个芯片的数据端与数据总线的高4位 ( $D_4 \sim D_7$ ) 相连；下面4个芯片的数据端与数据总线的低4位 ( $D_0 \sim D_3$ ) 相连。
3. 上、下4个芯片构成4K\*4RAM，共构成4K\*8位存储器模块。
4. 每一列芯片为一组，共分为4组，每一组芯片的  $\overline{CS}$ 端并连。
5. 地址总线的低地址线  $A_0 \sim A_9$  连接至每片的地址端；地址总线的高地址线，如  $A_{10}, A_{11}$  经过2-4译码有4个输出端，它们与4个组的CS相连，即  $A_0 \sim A_9$  作为片内单元选择， $A_{10}, A_{11}$  作为组间选择。



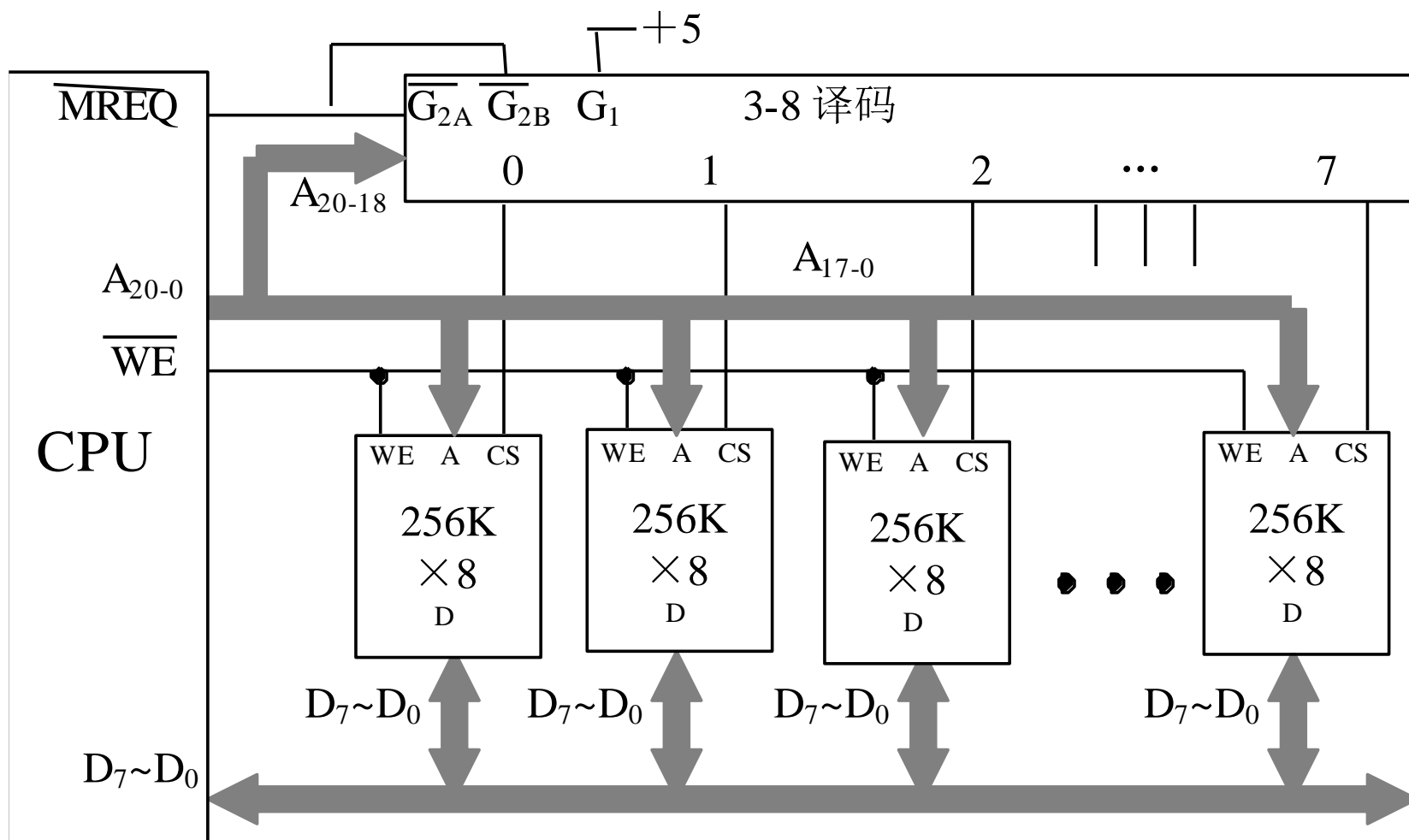




**例1 设有若干片256K×8位的SRAM芯片，问：**

- (1) 构成2MB的存储器需要多少块SRAM芯片？**
- (2) 该存储器需要多少字节地址线？**
- (3) 画出该存储器与CPU连接的结构图，设CPU的接口信号有地址信号、数据信号、控制信号MREQ和WE。**

- 解：**
- (1) 该存储器需要 $2048\text{K}/256\text{K} = 8$ 片SRAM芯片；
  - (2) 因为 $2^{21} = 2048\text{K}$ ，需要21条地址线。又 $2^{18} = 256\text{K}$ ，每个存储芯片有18根地址线。这样，高3位用于芯片选择，低18位作为每个存储器芯片的地址输入。
  - (3) 该存储器与CPU连接的结构图如下。





# 存储器与CPU的连接

## 1. 地址线的连接

- 低位地址线与存储芯片连接；高位地址线或用作存储芯片扩充时用，或作其他用法，如片选信号等。

## 2. 数据线的连接

- 必要时需要对存储芯片进行位扩展，使其数据位与CPU的数据线相等。

## 3. 读/写命令线的连接

- 直接与存储芯片的读写控制端相连，通常是高电平为读，低电平为写。



# 存储器与CPU的连接

## 4. 片选线的连接

- 片选信号的连接是CPU与存储芯片正确工作的关键。
- 片选有效信号与CPU的访存控制信号  $\overline{\text{MREQ}}$  (低电平有效) 有关。
  - $\overline{\text{MREQ}}$ 为低, 表示CPU访问存储器;  $\overline{\text{MREQ}}$ 为高, 表示CPU访问I/O, 此时不要求存储器工作。
- 片选信号与地址的高位有关, 未与存储芯片地址线连上的高位地址与访存控制信号共同作用产生存储器的片选信号。

## 5. 合理选择存储芯

- 类型ROM或RAM、数量的选择。



# 例4.1

- 题：设CPU有16根地址线，8根数据线，并用 $\overline{MREQ}$ 作访存控制信号（低电平有效），用 $\overline{WR}$ 作读写控制信号（高电平为读，低电平为写）。现有下列存储芯片：

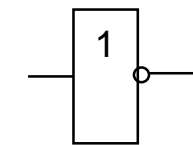
1KX4位RAM；4KX8位RAM；8KX8位RAM；

2KX8位ROM；4KX8位ROM；8KX8位ROM；

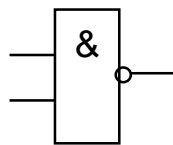
以及74LS138译码器和各种门电路（非门、与非门、或门）。

请画出CPU与存储器的连接图，要求：

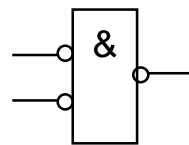
- ① 主存地址空间分配：6000H ~ 67FFH为系统程序区；6800H ~ 6BFFH为用户程序区。
- ② 合理选用上述存储芯片，说明各选几片？
- ③ 详细画出存储芯片的片选逻辑图。



非门



与非门



或门



# 例4.1

(1)先将16进制地址写成二进制地址码，并确其总容量。

A <sub>15</sub>	A <sub>14</sub>	A <sub>13</sub>	A <sub>12</sub>	A <sub>11</sub>	A <sub>10</sub>	A <sub>9</sub>	A <sub>8</sub>	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>
0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
.....															
0	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1
0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0
0	1	1	0	1	0	1	1	1	1	1	1	1	1	1	1

**6000H**  
 系统程序区  
 2kX8位  
**67FFH**  
**6800H**  
 用户程序区  
 1kX8位  
**6BFFH**

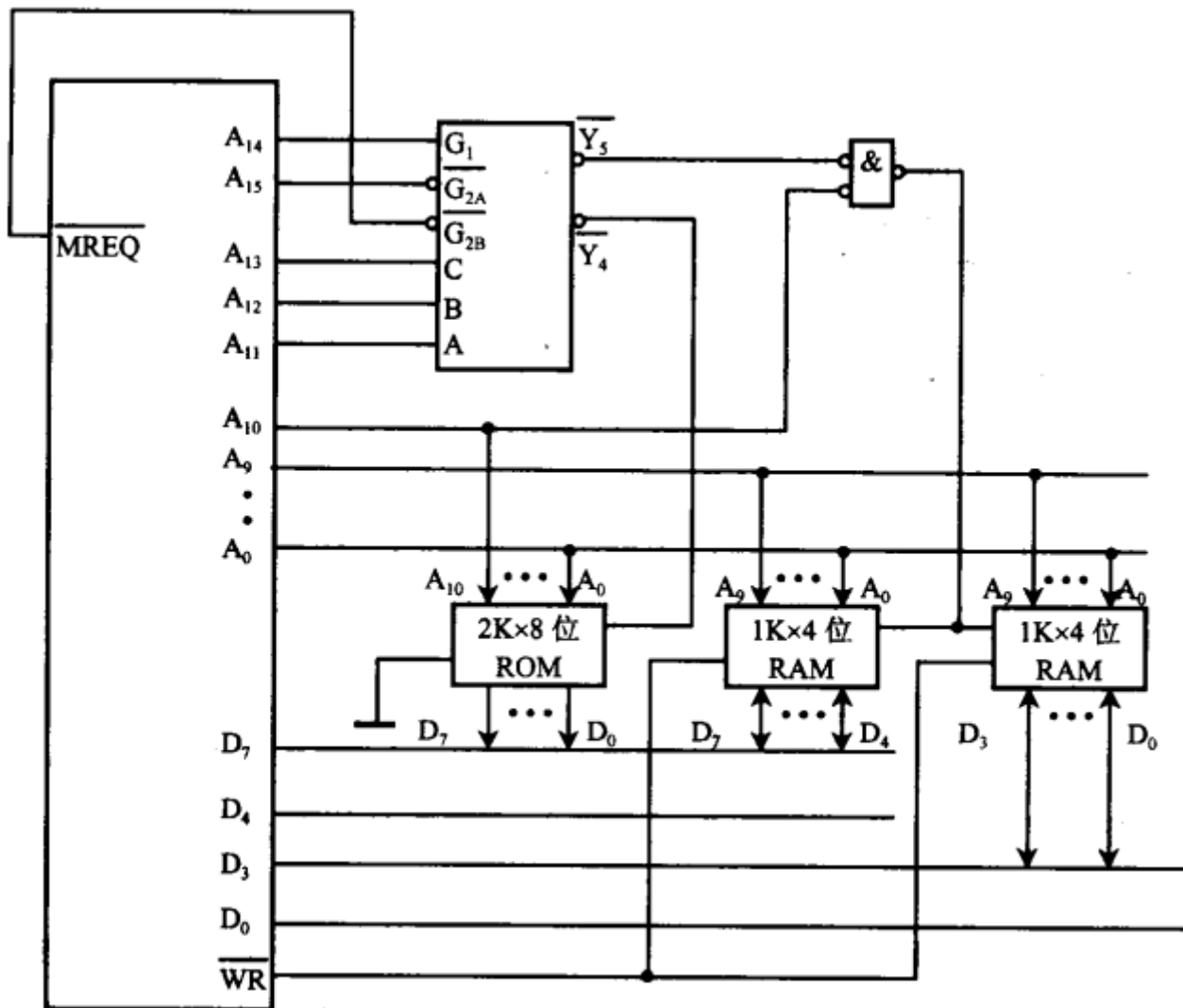
(2) 选器件 **可用芯片**

(3) **(5)ROM芯片的PD/progr引脚接地、ROM数据线单向连接数据总线；2片RAM的数据线分别连接数据总线高4位和低4位相连。**

(4)片选信号：A<sub>15</sub>和/MREQ接到/G<sub>2A</sub>和/G<sub>2B</sub>，A<sub>14</sub>接到G<sub>1</sub>。地址线A<sub>13</sub>~A<sub>11</sub>与3-8译码器的C,B,A引脚相连，3-8译码器的输出/Y<sub>4</sub>接到ROM的片选；/Y<sub>5</sub>和A<sub>10</sub>经或门接到2片RAM的片选。

# 例4.1

## •CPU与存储芯片的连接图





## 例4.2

- **题：** CPU及其他芯片假设同上题，画出CPU与存储器的连接图。要求主存的地址空间满足下述条件：最小8K地址为系统程序区，与其相邻的16K地址为用户程序区，最大4K地址空间为系统程序工作区。详细画出存储芯片的片选逻辑并指出存储芯片的种类及其片数。
- **注意：** 系统程序区用ROM，但是**系统程序工作区**用RAM。

### 可用芯片

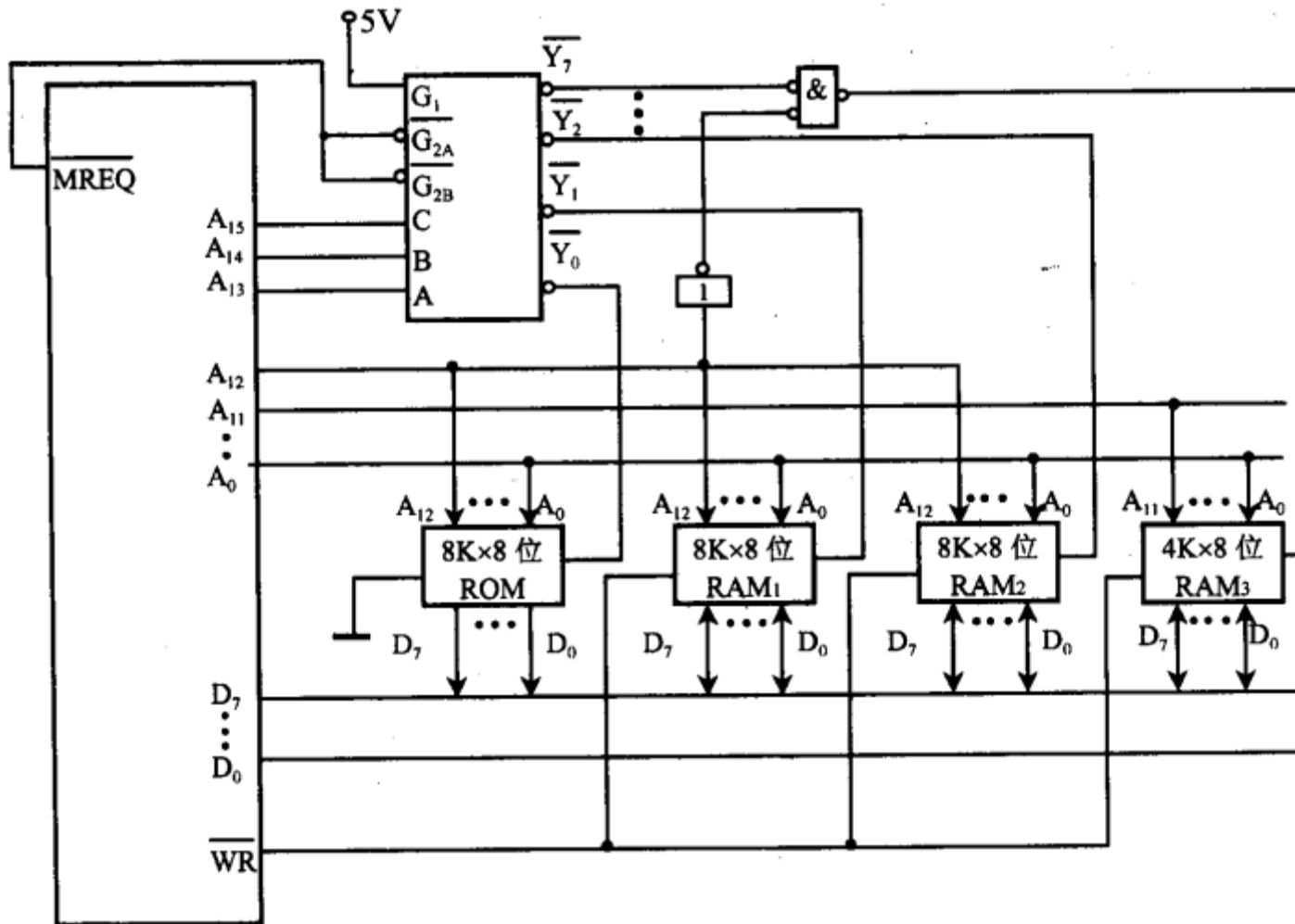
2KX8位ROM; 4KX8位ROM; 8KX8位ROM  
1KX4位RAM; 4KX8位RAM; 8KX8位RAM



# 例4.2



## •CPU与存储芯片的连接图

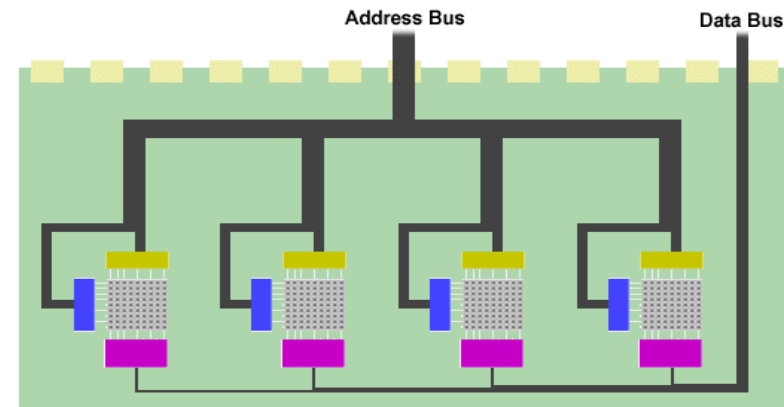
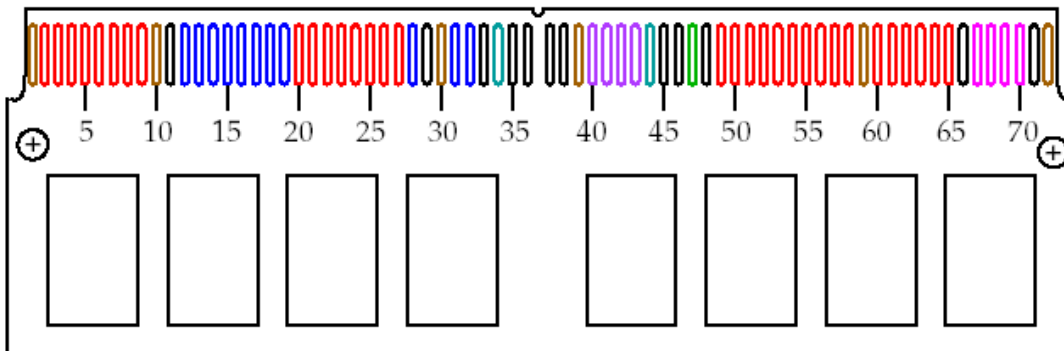
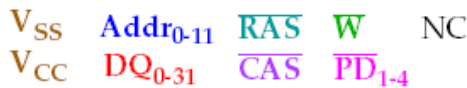
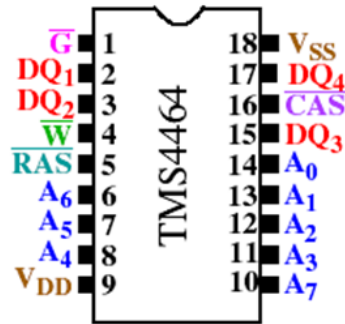


# Turning Bits Into Bytes: 内存条



- Larger DRAMs are available which are organized as  $1M \times 1$ ,  $4M \times 1$ ,  $16M \times 1$ ,  $64M \times 1$ ,  $256M \times 1$ .....
- DRAMs are typically placed on **SIMM boards**.
  - Single In-line Memory Modules
  - **30-pin** SIMMs (8bits data) come in  $1M \times 8$ ,  $1M \times 9$  (parity),  $4M \times 8$ ,  $4M \times 9$ .
  - **72-pin** SIMMs (32bits data) come in  $1/2/3/8/16M \times 32$  or  $1M \times 36$  (parity).

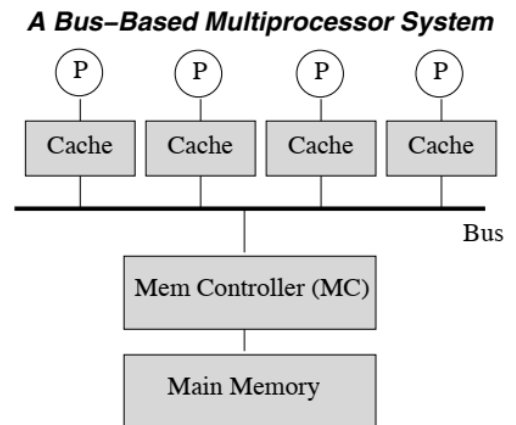
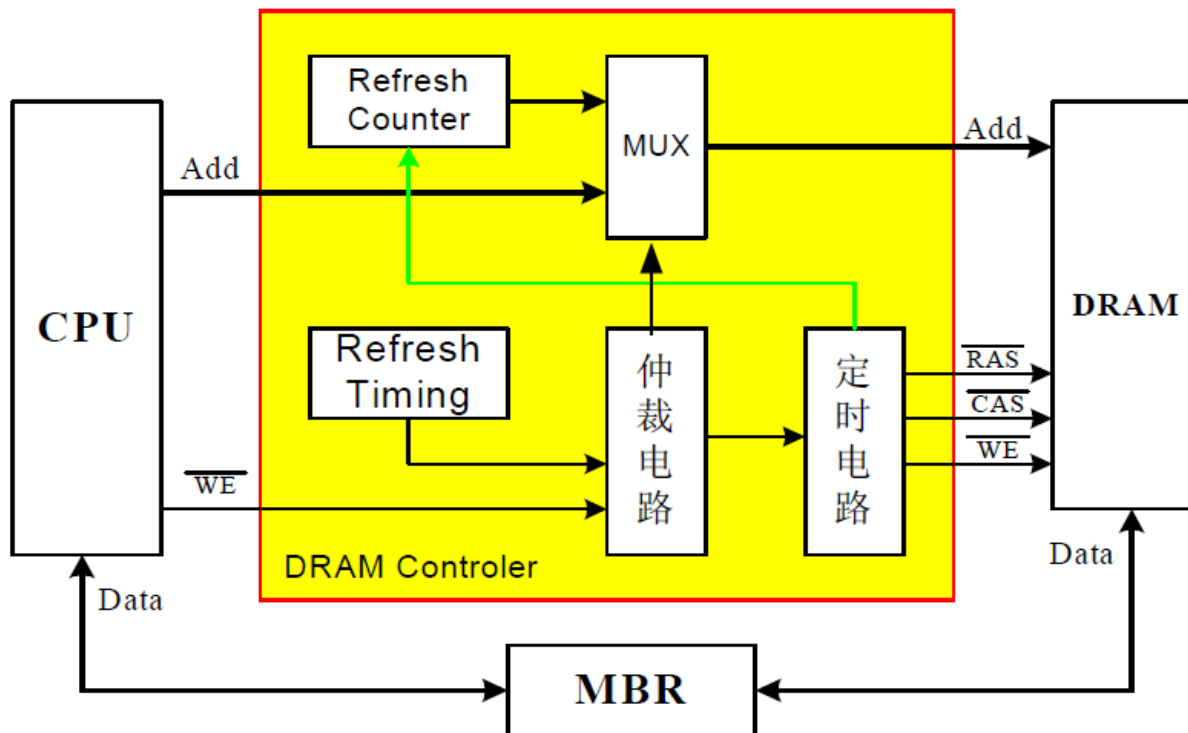
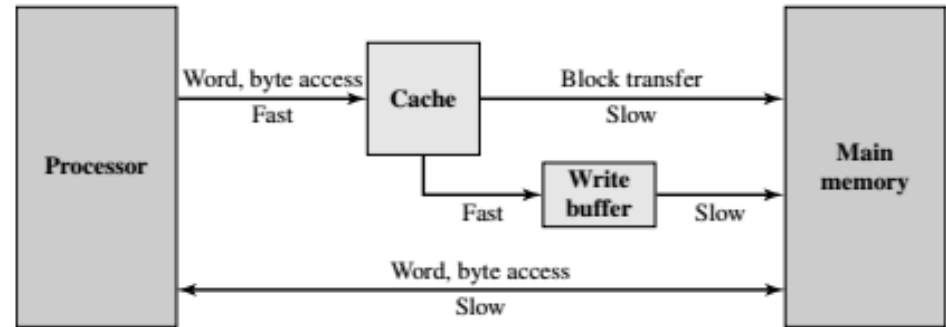
## • DIMM





# DRAM Controllers

- DRAM扩展的特殊性
  - 行地址与列地址复用：时序？
  - 刷新控制：提供刷新地址
    - 仲裁：刷新与CPU访存并发



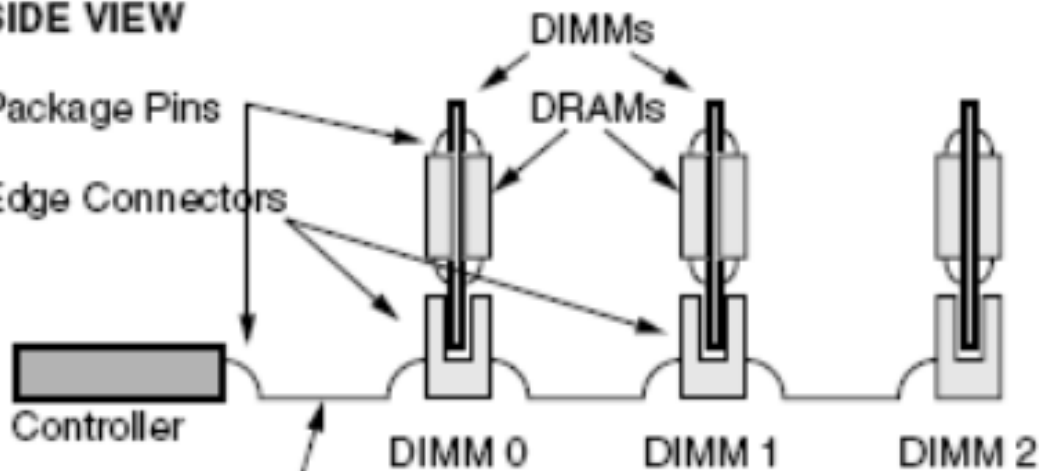
# 内存系统：Chip、多面Rank、多体bank



SIDE VIEW

Package Pins

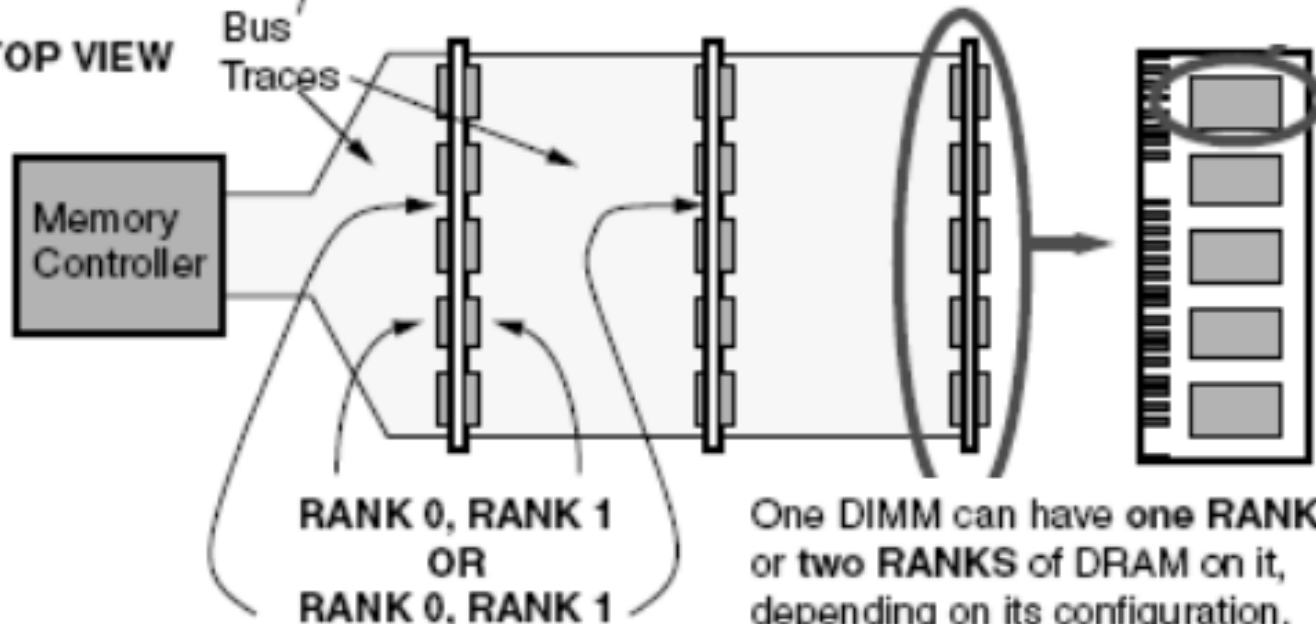
Edge Connectors



TOP VIEW

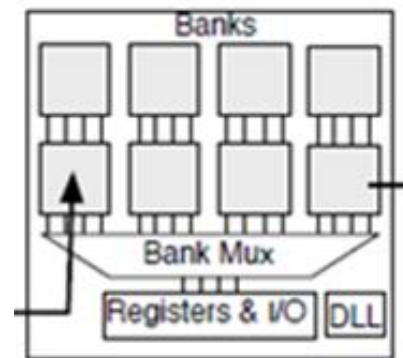
Bus Traces

Memory Controller



One DIMM can have one RANK or two RANKS of DRAM on it, depending on its configuration.

DRAM Chip



# 提高访存性能（带宽）的措施，RV\$5.2.2



- SDRAM

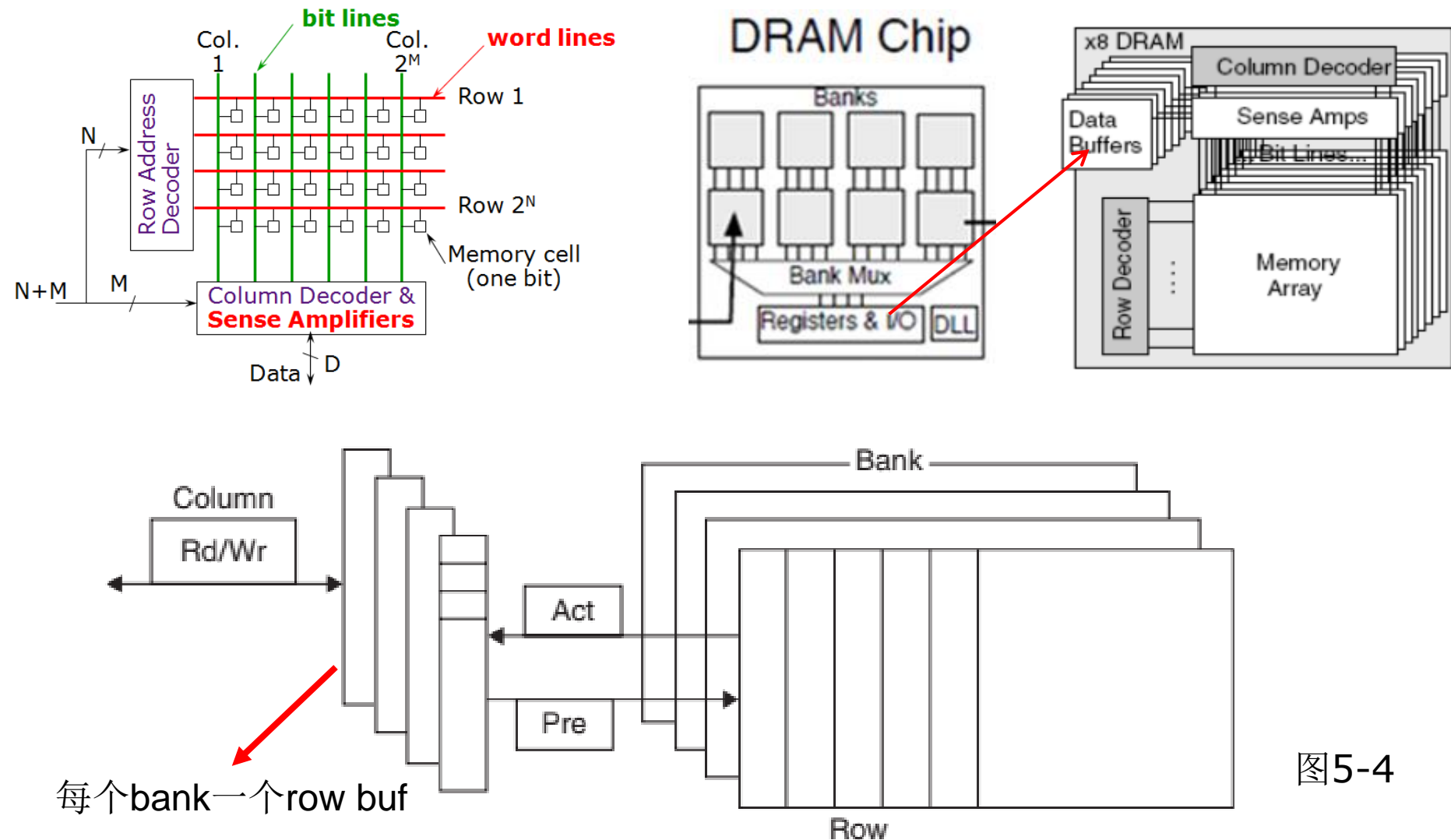
- 同步：与CPU周期同步
- DDR
- Bank Arch：高带宽
  - Bank可并行工作
  - 单体多字
  - 多体并行（地址交叉）
    - Burst

Year introduced	Chip size	\$ per GiB	Total access time to a new row/column	Average column access time to existing row
1980	64 Kibibit	\$1,500,000	250 ns	150 ns
1983	256 Kibibit	\$500,000	185 ns	100 ns
1985	1 Mebibit	\$200,000	135 ns	40 ns
1989	4 Mebibit	\$50,000	110 ns	40 ns
1992	16 Mebibit	\$15,000	90 ns	30 ns
1996	64 Mebibit	\$10,000	60 ns	12 ns
1998	128 Mebibit	\$4,000	60 ns	10 ns
2000	256 Mebibit	\$1,000	55 ns	7 ns
2004	512 Mebibit	\$250	50 ns	5 ns
2007	1 Gibibit	\$50	45 ns	1.25 ns
2010	2 Gibibit	\$30	40 ns	1 ns
2012	4 Gibibit	\$1	35 ns	0.8 ns

- 采用双端口存储器
- 采用层次化存储系统结构
  - Cacheline size
  - Non-Block Cache

RV图5-5

# Bank Arch: 多体, COD5 图5-4

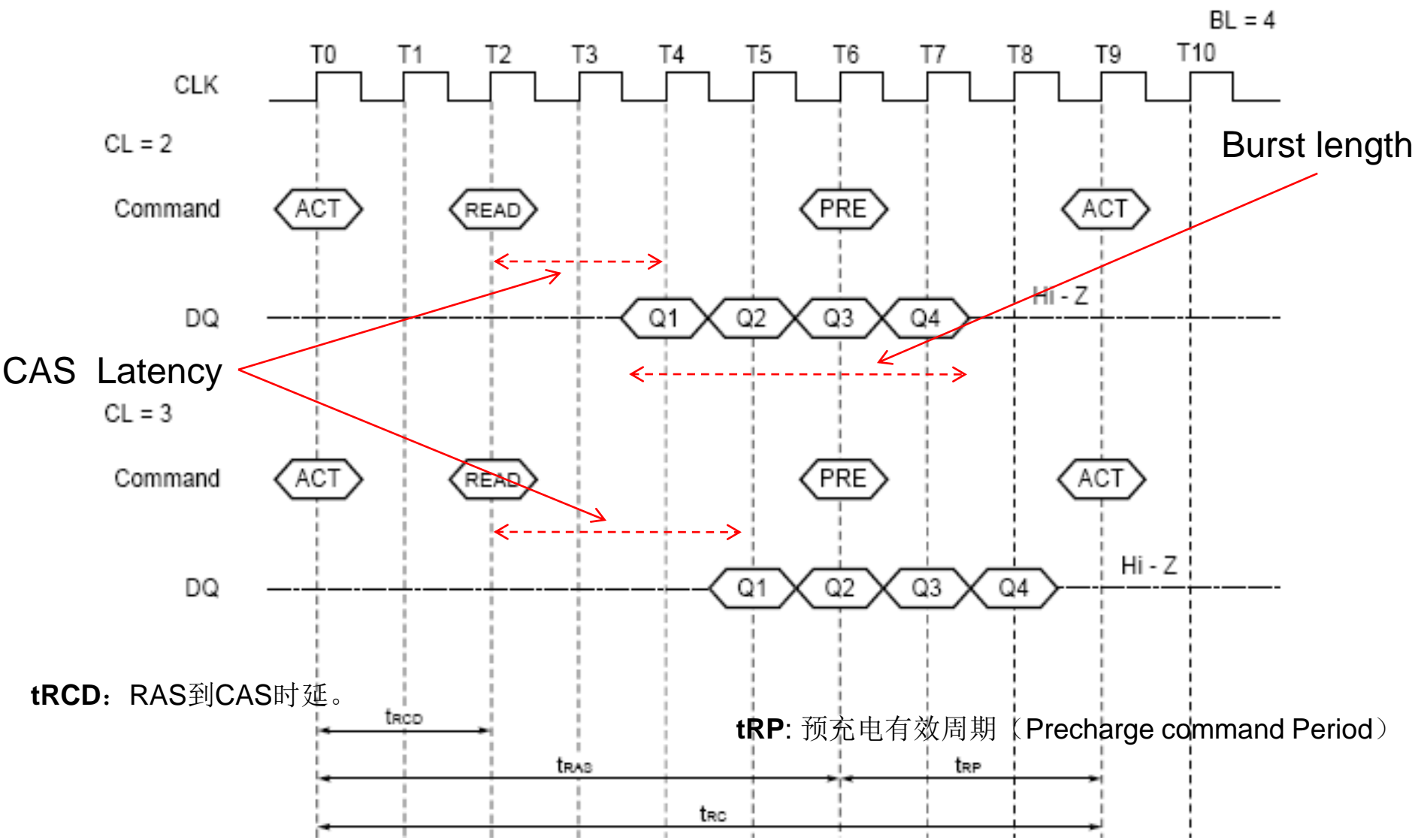


每个bank一个row buf

图5-4

Pre-charge (one Bank) ->RAS (Activate, row buf) ->CAS (Read/Write)

# SDR SDRAM时序 (Burst为基本操作)



**tRCD:** RAS到CAS时延。

**tRP:** 预充电有效周期 (Precharge command Period)

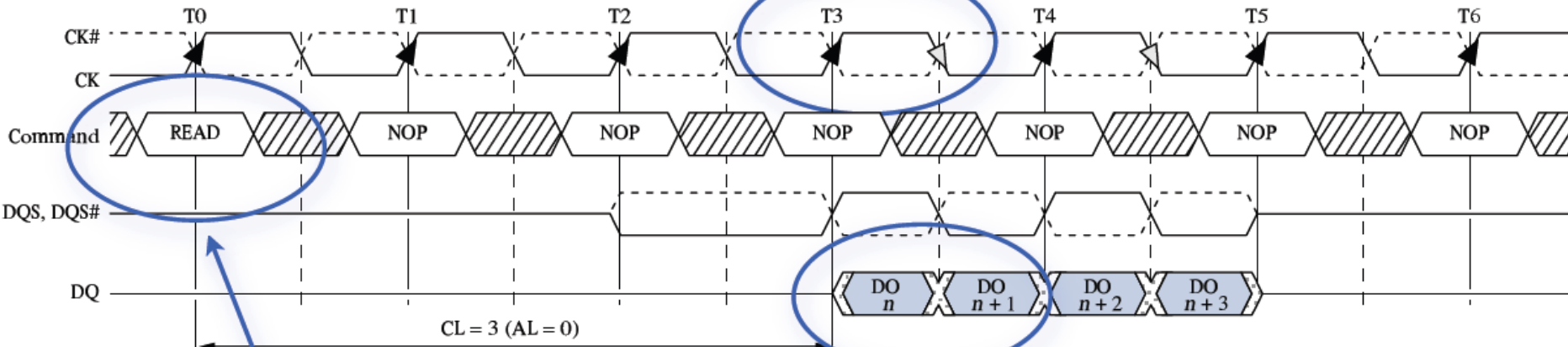
CPU knows when data will be ready, does not have to wait!

# DDR(Double Data Rate) SDRAM



A clocked bus:  
200 MHz clock,  
data transfers on  
both edges (**DDR**).

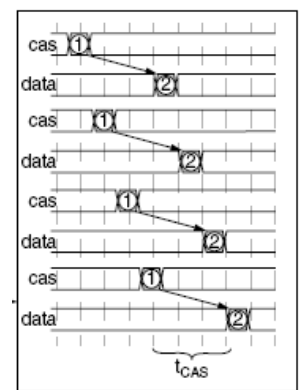
Note! This example is **best-case!**  
To access a new row, a slow **ACTIVE**  
command must run before the **READ**.



DRAM is controlled  
via commands  
(**READ, WRITE,  
REFRESH, ...**)

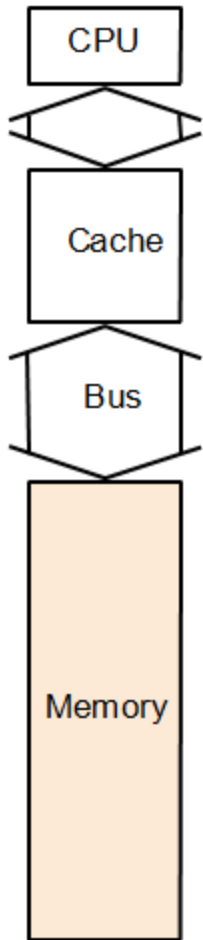
CAS latency

Synchronous  
data output.

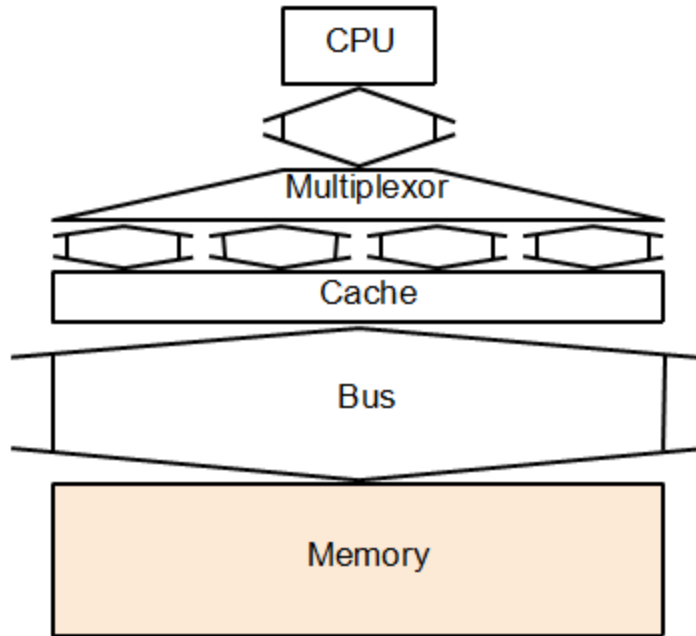




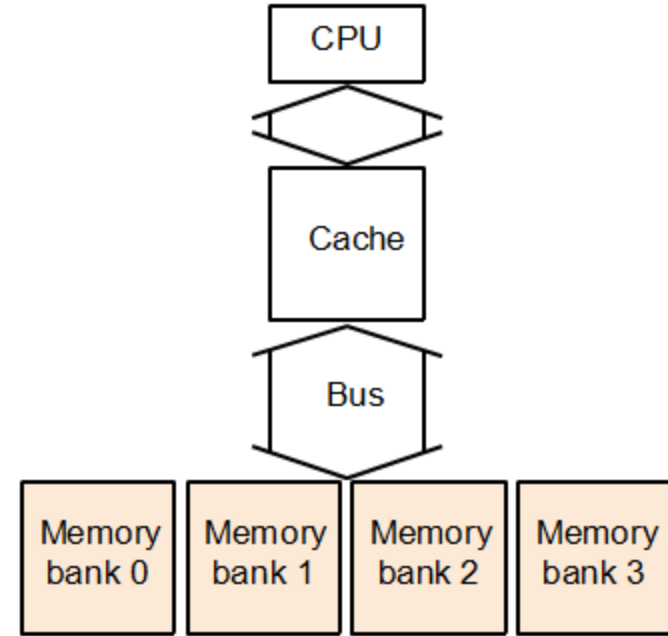
# Main Memory Organizations 模式



one-word wide  
memory organization



wide memory organization



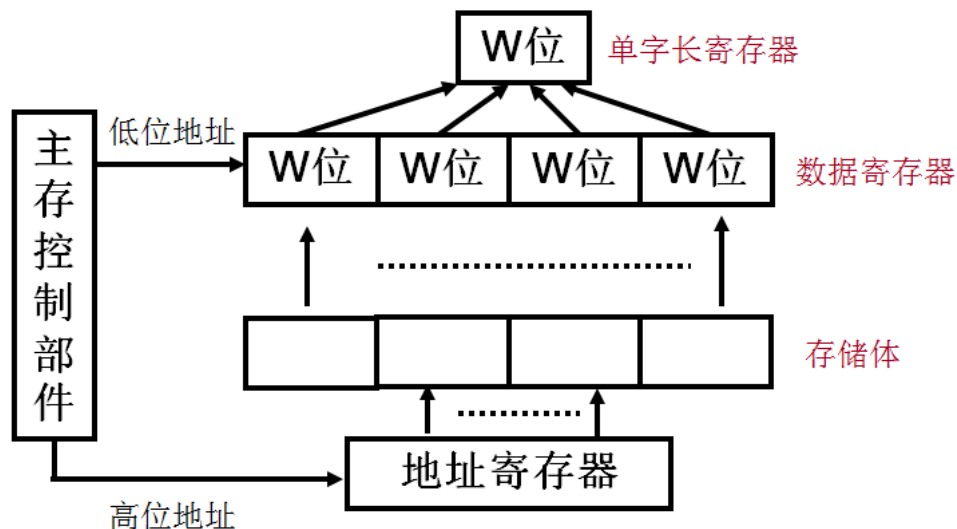
interleaved  
memory organization

- COD4\$5.2.5图5-11：支持Cache的存储系统（COD5无）
  - 单体单字 → 单体多字（wide） → 多体单（多）字（interleaved）
- 多体并行：用**低带宽**器件构成**高带宽**（wide）存储系统
  - Bank：并行、**交叉**（流水线？）



# 单体多字系统

- 系统中只有一个存储体，体内包含多字
  - 由单个存储芯片或多个存储芯片构成的一个独立存储器
  - 多字：例如，在一个周期内，从**同一地址开始**顺序读出4条指令字，再**逐条**将指令送至CPU执行。



唐图4.41

# 多体并行系统：interleaved memory

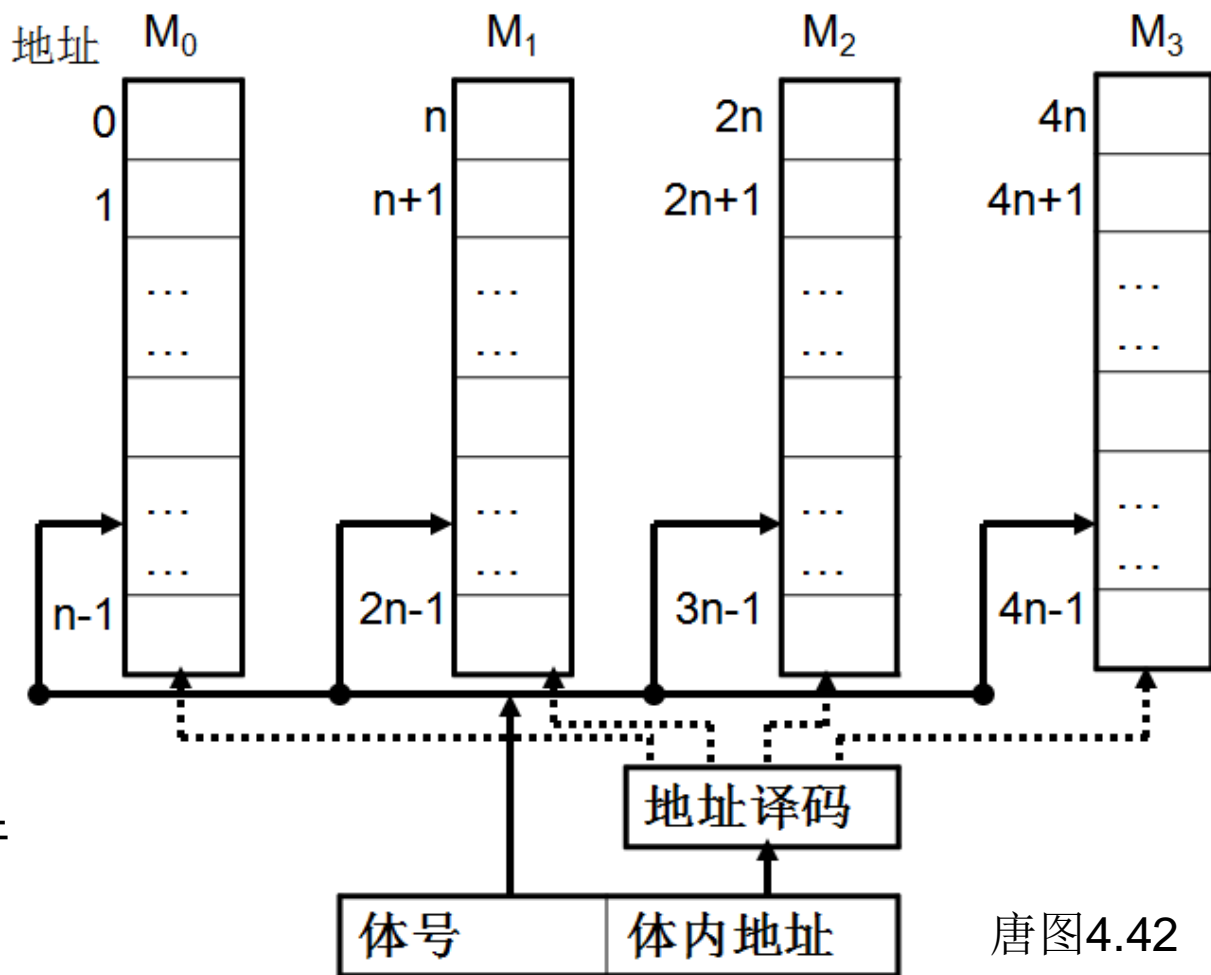


- 系统含多个存储体 (**rank/bank**)
  - 各体有自己的读写线路、地址寄存器和数据寄存器
  - 两种编址方式
    - 高位交叉编址：**扩容**，不同应用空间。也称“顺序编址”
    - 低位交叉编址：**带宽**，多体并行访问，多体交叉访问
- **并行访问：高位交叉编址**
  - 同时访问M个模块：同时启动，同时读出，并行工作
    - 每个模块一个字
    - 同时读出的M个字需在总线（单字）上**分时**传送
- **交叉访问：低位交叉编址**
  - M个模块按一定的顺序轮流启动各自的访问周期
  - Burst

# 多体并行系统—高位交叉编址



- 体选：高位地址
  - 顺序编址
- 方便扩容
  - 不同插槽对应不同存储体
- 支持并行访问
  - 多任务、多处理器系统
    - 多个任务并发执行
  - 程序执行、I/O
    - 某体用于程序执行，某体用于I/O



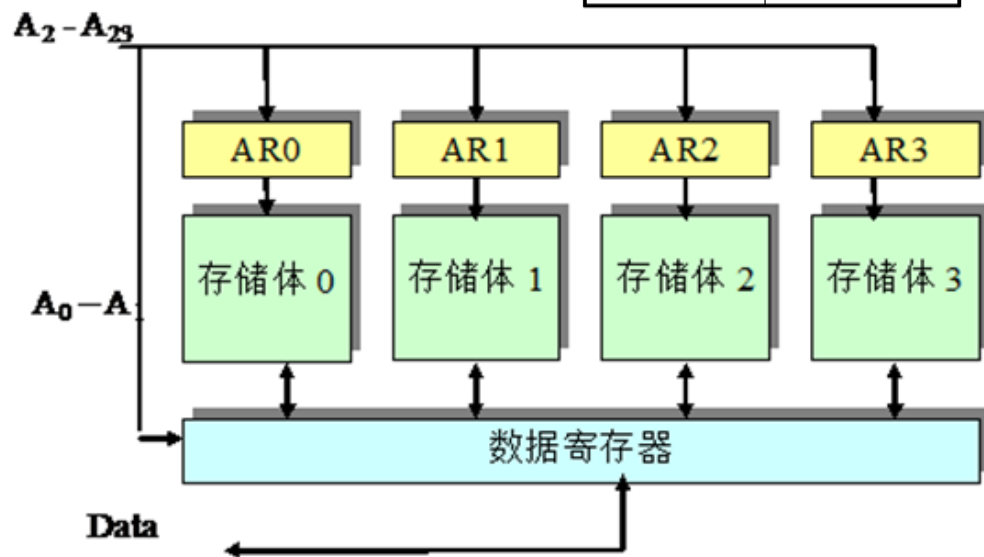
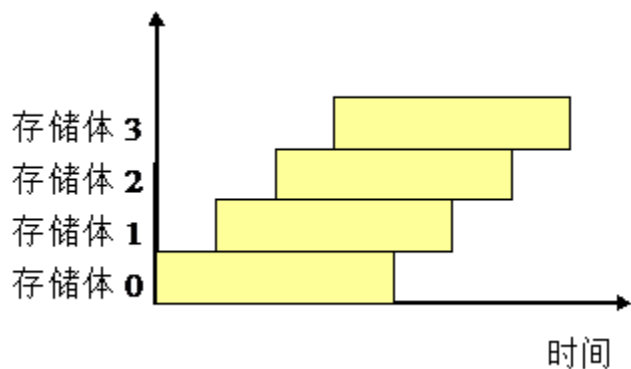
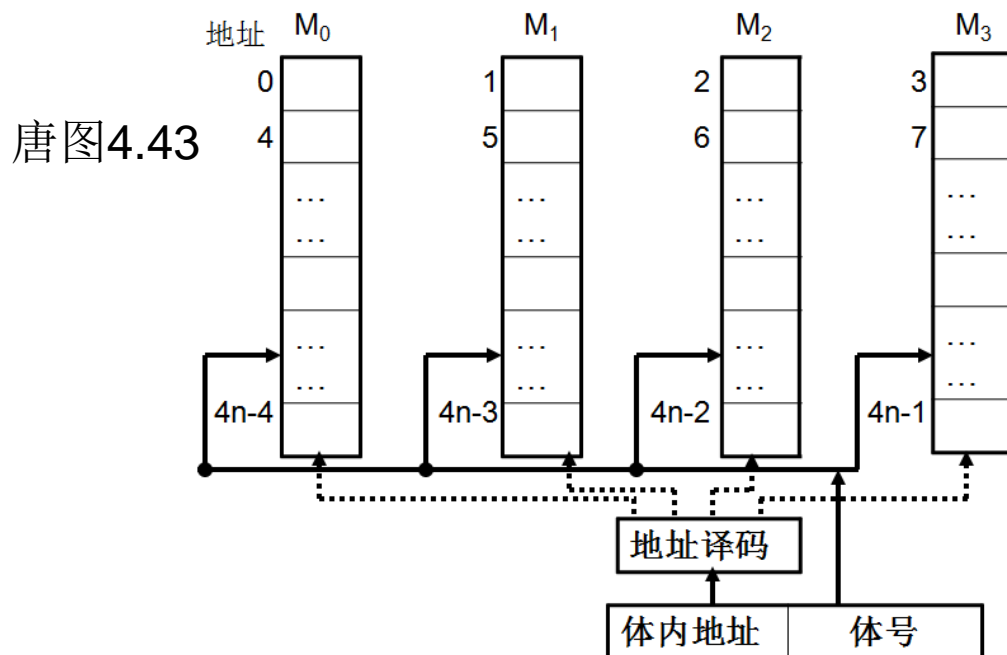
唐图4.42





# 多体并行系统—低位交叉编址

- 体选：低位地址
- 交叉访问
  - 轮流启动 (RR)
  - 多用：高带宽
    - 带宽提高M倍

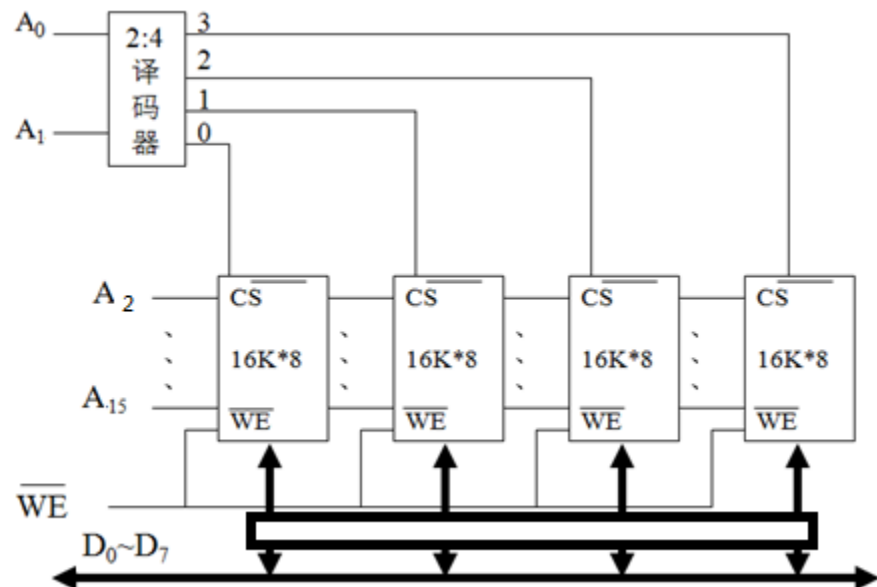
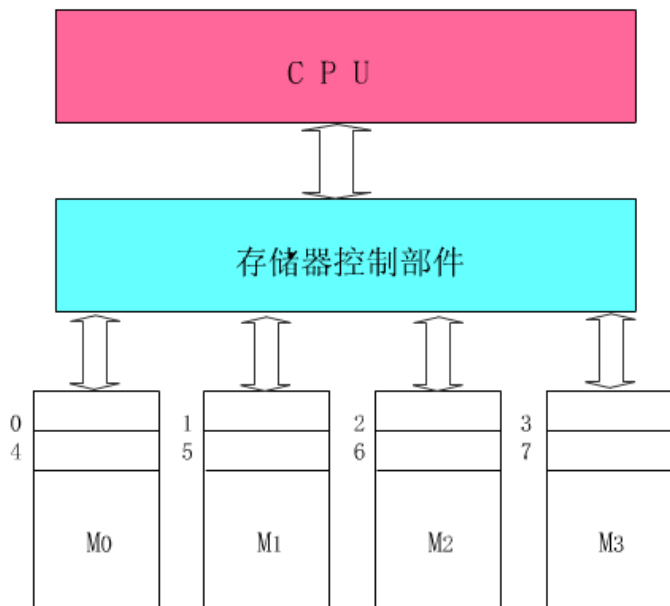


# 多体并行系统—低位交叉编址和访问



例：模四交叉个模块的编址序列

体号	体内编址序列	对应二进制地址最低二位
M1	0, 4, 8, 12, ..., $4j+0$ , ...	0 0
M2	1, 5, 9, 13, ..., $4j+1$ , ...	0 1
M3	2, 6, 10, 14, ..., $4j+2$ , ...	1 0
M4	3, 7, 11, 15, ..., $4j+3$ , ...	1 1





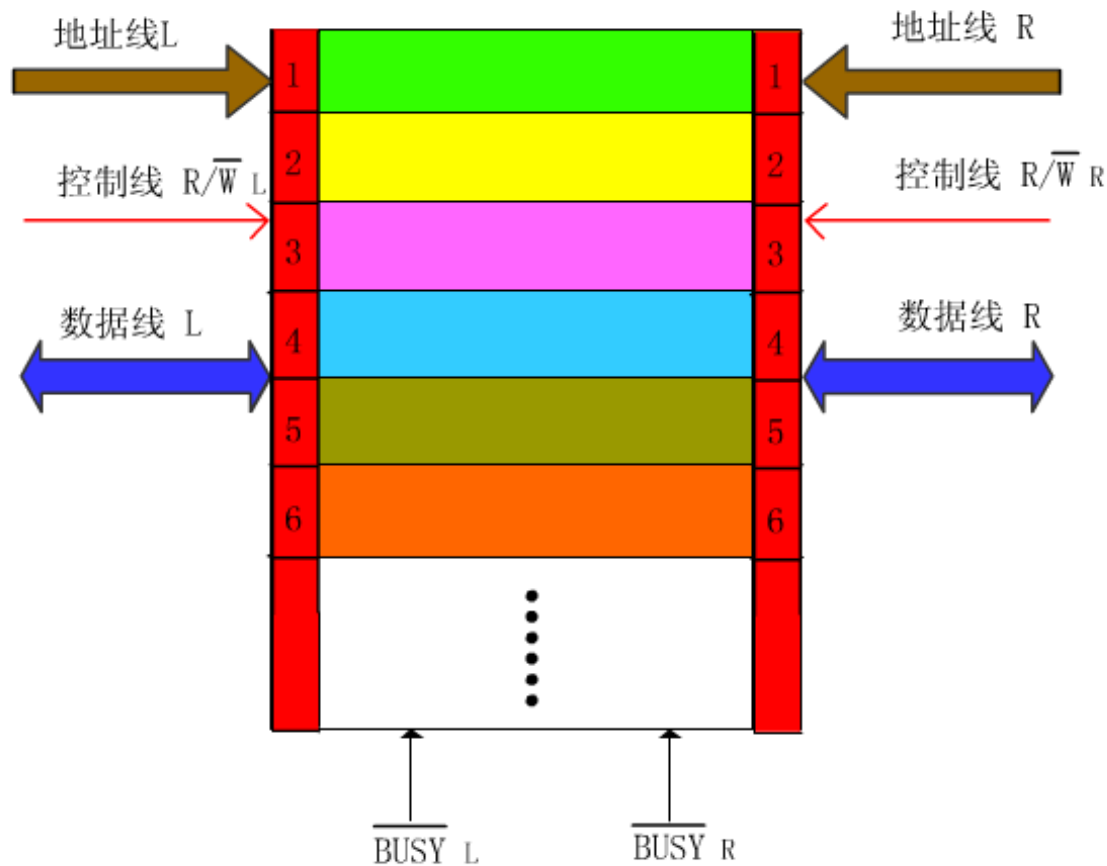
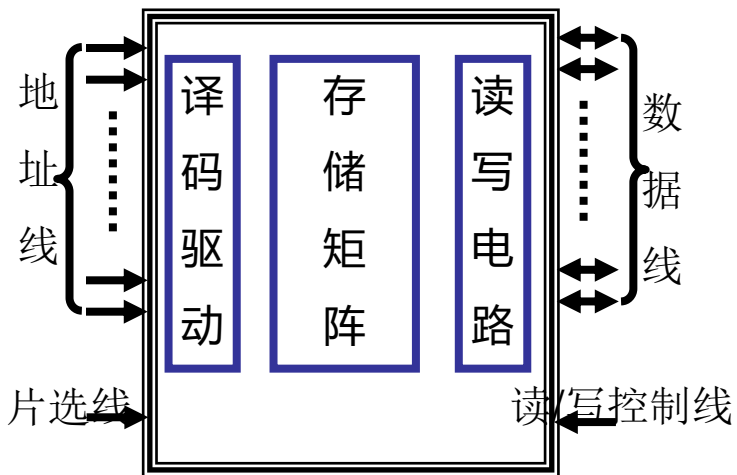
# 例：

在一个具有4个存储体的低位多体交叉存储器中，如果处理器的访存地址为以下十进制值。求该存储器比单体存储器的平均访问速率提高多少（忽略初启时的延迟）？

- (1) 0001、0002、0003、...、0100
- (2) 0002、0004、0006、...、0200
- (3) 0003、0006、0009、...、0300

- 解：**
- (1) 各个访问操作可以交叉进行，访问速率可达到单体存储器的4倍。
  - (2) 只有2个存储体交叉访问，访问速率可达到单体存储器的2倍。
  - (3) 访问的存储体分别是3、2、1、0、3、...，各属于不同的存储体，访问速率可达到单体存储器的4倍。

# 双端口存储器

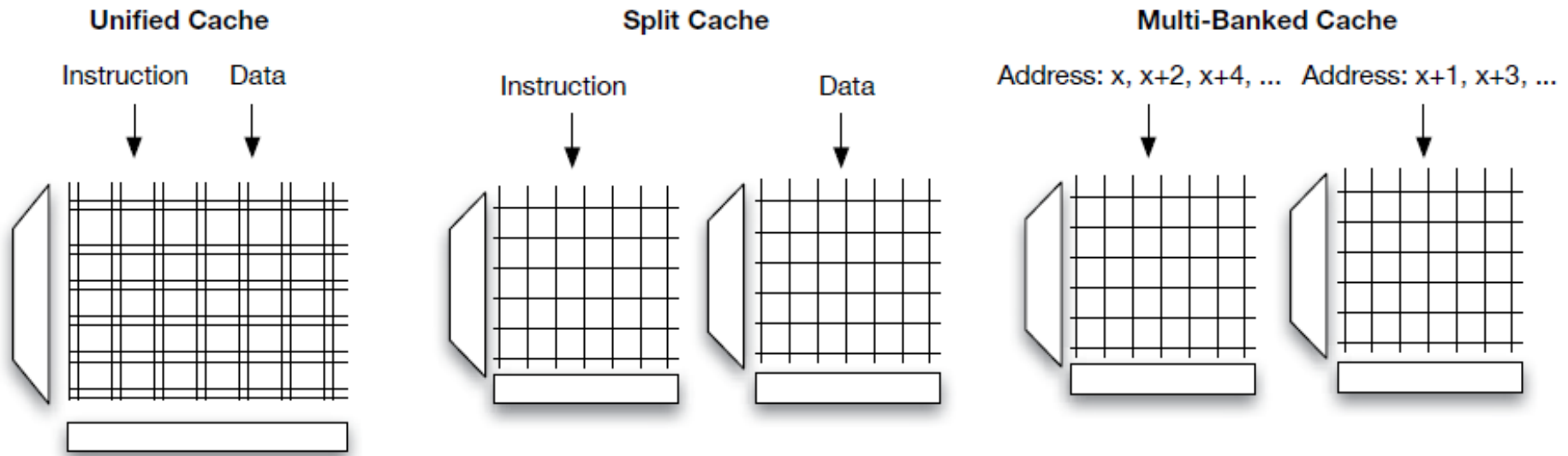


当两个端口同时存取存储器同一存储单元时，发生读写冲突。

因此，设置BUSY，由片上的判断逻辑决定对哪个端口优先进行读写操作，而暂时关闭另一个被延迟的端口。



# 单处理器Cache组织：分立，分体



- 组织：多端口，统一，分体，multiBanked
- 一致性

# 小结



- 主存储器的基本组成和技术指标
- 主存储器（半导体存储器）
  - 基本存储单元及操作
    - SRAM
    - DRAM: recharge、precharge、activate (RAS) 、CAS (R/W) , refresh
  - DRAM刷新方式：按行
- 存储器与CPU的连接
  - 位扩展, 字扩展
  - DRAM控制器：同步控制
- 提高访存性能
  - Bank Arch: 交叉编址, burst
- 作业（唐本）：4.6, 4.15, 4.24



Thank You