



华北电力大学

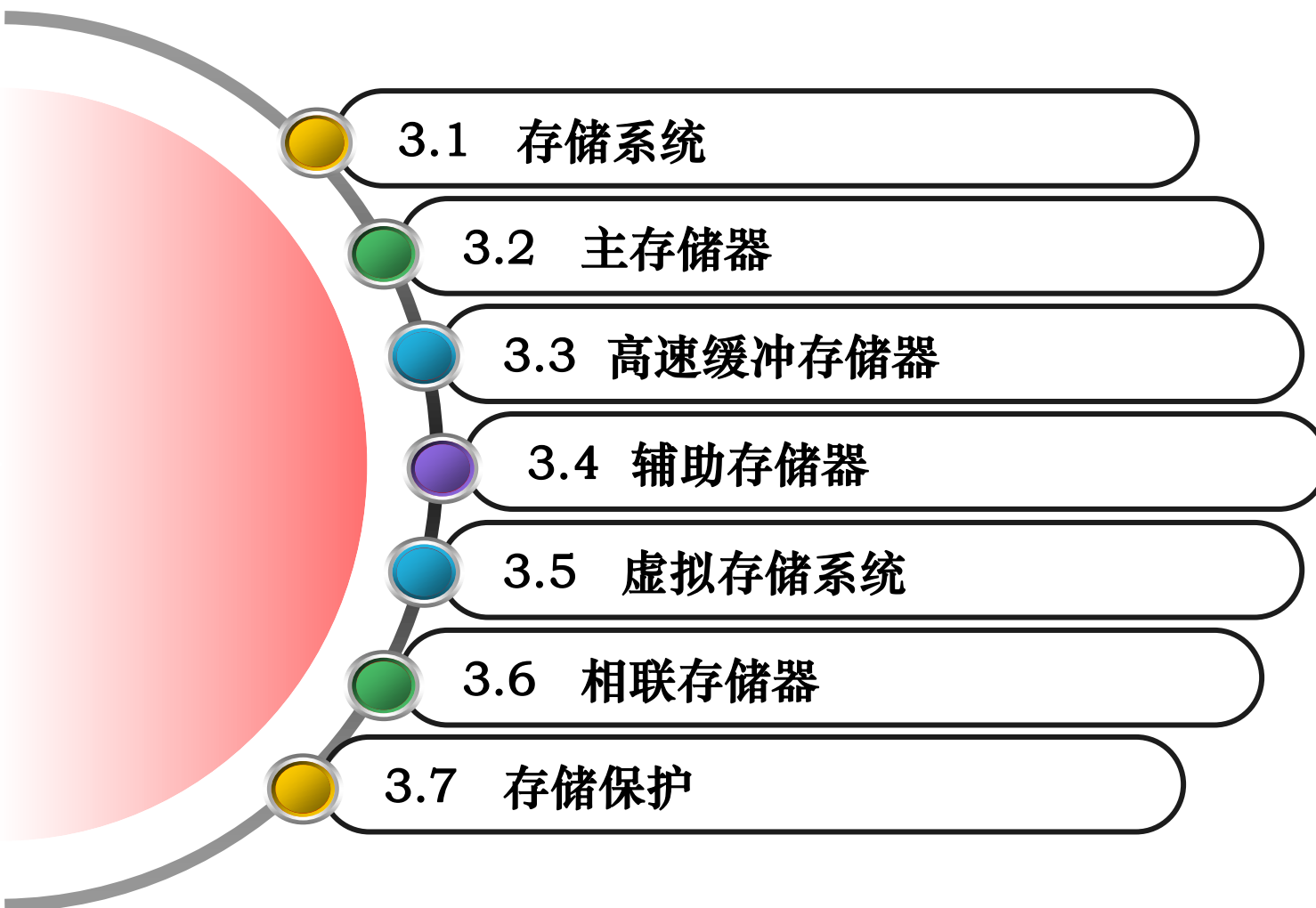
North China Electric Power University

# 计算机组成与结构

## 第三章

华北电力大计算机系

# 本章结构



## 3.1 存储系统概述

处理、传输、存储是信息技术的三大基石



**“新摩尔定律”** —— 从现在起每18个月，新增的存储量等于有史以来存储量之和！

1998年图灵奖获得者  
Jim Gray

# 3.1 存储系统概述

## IDC 《数据时代2025》 白皮书

全世界权威IT咨询公司研究报告预测：  
全球数据量将从 2018 年的 33ZB 增至 2025 年的 175ZB，达到5倍以上。



## 3.1 存储系统概述

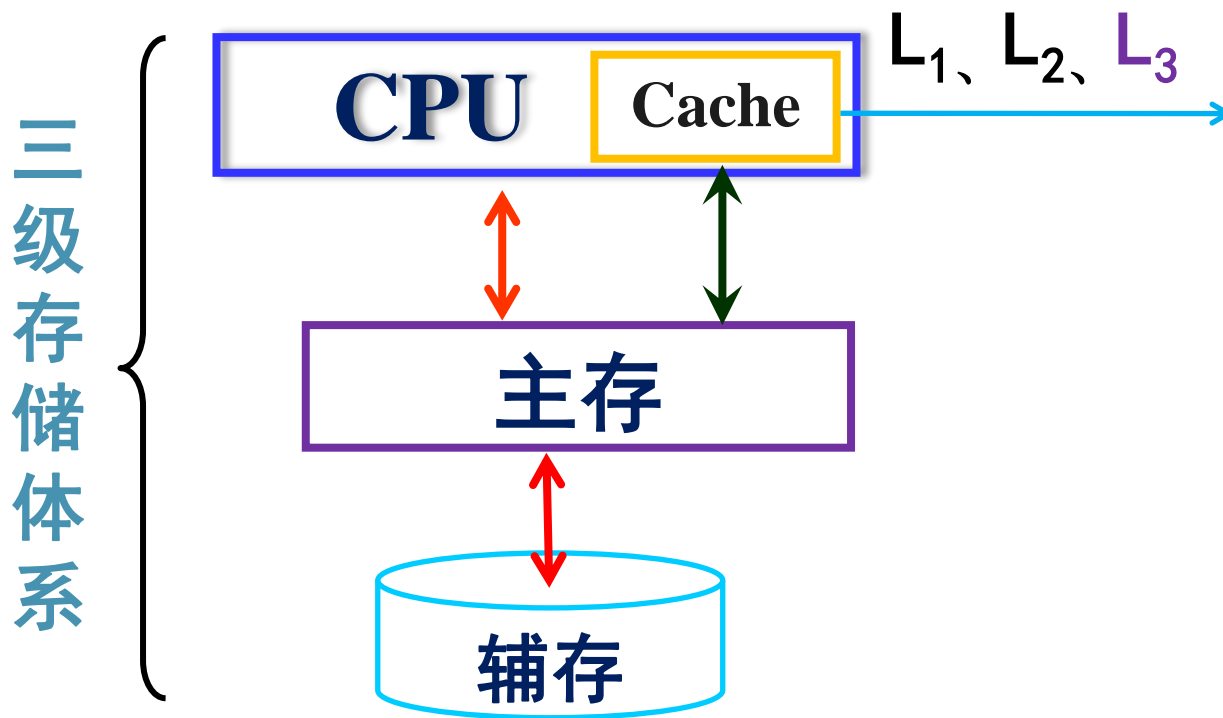
**本章需解决的主要问题：**

- (1) 存储器如何存储信息？**
- (2) 在实际应用中如何用存储芯片组成具有一定容量的存储器？**
- (3) 如何改进存储系统的性能？**

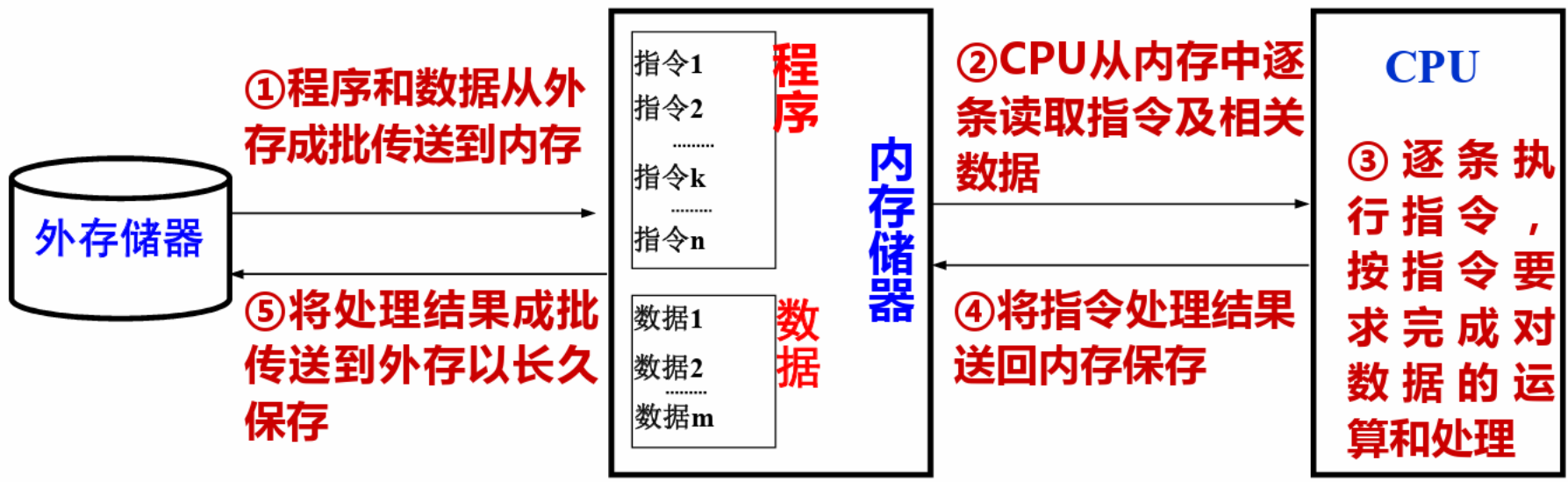


## 3.1 存储系统概述

计算机系统中的记忆设备，用于存放程序和数据



# 3.1 存储系统概述



## 外存储器（简称外存或辅存）

- 存取速度慢
- 成本低、容量很大
- 不与CPU直接连接，先传送到内存，然后才能被CPU使用。
- 用于长久存放系统中几乎所有的信息

## 内存储器（简称内存或主存）

- 存取速度快
- 成本高、容量相对较小
- 直接与CPU连接，CPU对内存中可直接进行读、写操作
- 用于临时存放正在运行的程序和数据

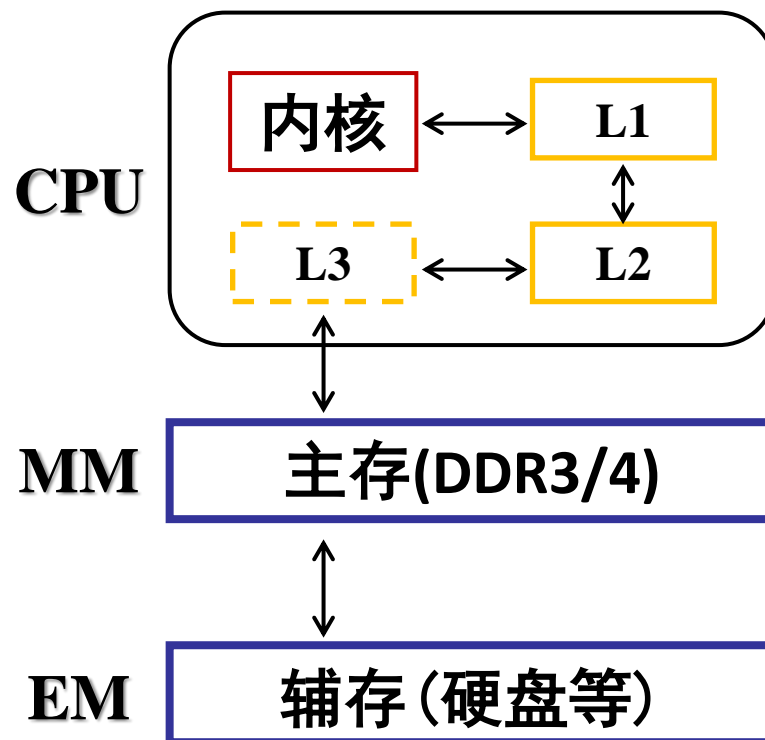


## 3.1 存储系统概述

### 高速缓冲存储器(Cache)

存放CPU在当前一小段时间内多次使用的程序和数据，以缓解CPU和主存的速度差异。

速度非常快  
容量却很小





## 3.1 存储系统概述

3.1.1

存储器的分类

3.1.2

存储系统的层次结构

## 3.1.1 存储器的分类

### 1. 按存储介质分类

#### □ 半导体存储器

- 速度快，用作内存
- 记忆原理：双稳态触发器、电容(静态、动态)

#### □ 磁表面存储器

- 用陶瓷、非磁性金属或塑料作载磁体，磁化后具有两种不同的剩磁状态记录信息“1”和“0”
- 非易失，容量大并且每位价格低。用作外存

#### □ 光盘存储器

- 有机玻璃作载磁体，利用磁化、晶态/非晶态表示信息
- 非易失，可靠性高，保存时间长，容量大且易于更换
- 存储速度比硬盘低一个数量级

#### □ 铁电存储器FeM(FRAM)、相变存储器RCM、阻变存储器ReRAM等

- 掉电数据不丢失(non-volatile)，速度快

## 3.1.1 存储器的分类

### 2.按存取方式分类

#### □ 随机访问存储器

- RAM和ROM
- 存储器任意单元可随时访问且访问所需时间相同
- 访问时间与存储单元所处的物理位置无关
- 速度快(ns)
- 主存和Cache

#### □ 顺序存储器

- 存储时以数据块为单位存储，顺序地记录在存储介质上
- 数据按顺序从存储载体的始端读出或写入，存取时间的长短与数据所在位置有关
- 速度慢、容量大、成本低，用作后援外存

## 3.1.1 存储器的分类

### 2.按存取方式分类

#### □ 直接存取存储器

- 介于随机和顺序存取之间
- 可以直接定位到要读写的数据块，存取时间的长短与数据所在位置有关
- 速度慢(ms)
- 磁盘

#### □ 相联存储器

- 按内容检索到存储位置进行读写
- 速度快(ns)，价格高
- 快表(TLB)

## 3.1.1 存储器的分类

### 3. 按在计算机中的作用分类

#### □ 寄存器

- 用于存放当前正在执行的指令和使用的数据
- 用触发器实现，速度快，容量小  $O(KB)$

#### □ 高速缓冲存储器(Cache)

- 位于CPU内部或位于主存和CPU之间
- 用来存放当前正在执行的局部程序段和数据
- 用SRAM实现，速度 $O(ns)$ 、容量 $O(MB)$

#### □ 主存储器

- 位于CPU之外
- 用来存放已被启动的程序及所用数据
- 用DRAM实现，速度 $O(ns)$ 、容量 $O(GB)$

## 3.1.1 存储器的分类

### 3. 按在计算机中的作用分类

#### □ 辅助存储器

- 不能由CPU的指令直接访问
- 用来存放暂不运行的程序、数据或存档文件
- 用磁、光等存储器件实现，容量大，但速度慢O(ms)

## 3.1.1 存储器的分类

### 4. 按存储器中信息的可保存性分类

#### □ 断电后是否丢失数据

##### ➤ 挥发性存储器(易失性存储器)

- 特点：断电后，信息即丢失
- SRAM、DRAM

##### ➤ 非挥发性存储器(非易失性/永久性存储器)

- 特点：断电后，信息不丢失
- ROM、磁盘、闪存

#### □ 读出后是否保持数据

##### ➤ 破坏性存储器

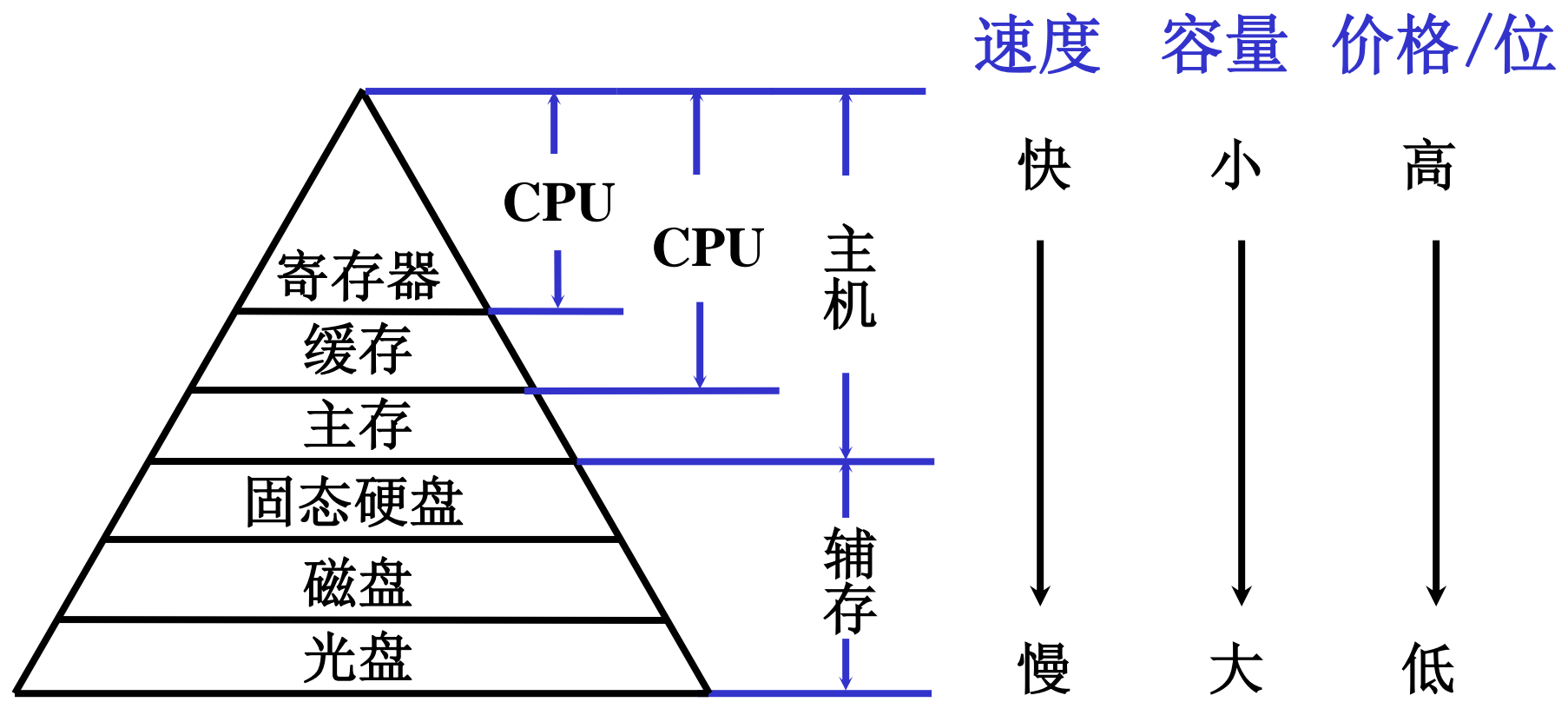
- 特点：读出时原存信息被破坏，需重写
- DRAM

##### ➤ 非破坏性存储器

- 特点：读出时，原存信息不被破坏
- SRAM



### 3.1.2 存储系统的层次结构



问题：你认为哪一种最适合做计算机的存储器呢？

单独用某一种存储器，都不能满足我们的需要！





### 3.1.2 存储系统的层次结构

内存：动态随机访问存储器      DRAM		
Year	容量	访存周期时间
1980	64 kbit	250 nsec
1983	256 kb	220 ns
1986	1 Mb	190 ns
1989	4 Mb	165 ns
1992	16 Mb	145 ns
1995	64 Mb	120 ns
1999	128 Mb	110 ns
2003	512 Mb	90 ns
2007	2 Gb	60 ns

容量 Capacity

DRAM: 4 x in 3 years

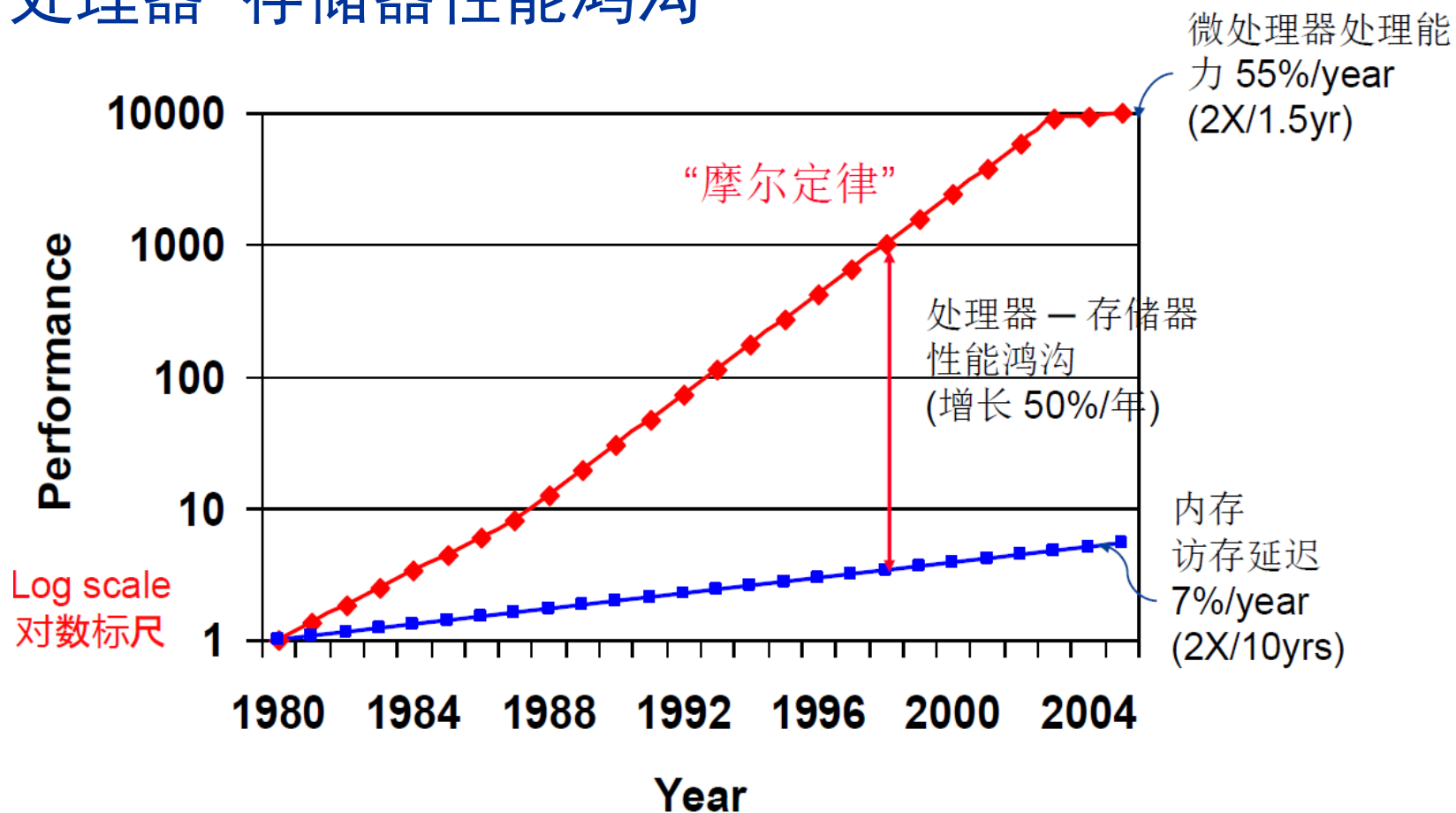
访存周期 (延迟: latency)

DRAM: 2 x in 10 years



## 3.1.2 存储系统的层次结构

### 处理器-存储器性能鸿沟



问题：主存增速与CPU增速不同步；指令执行期间多次访问存储器



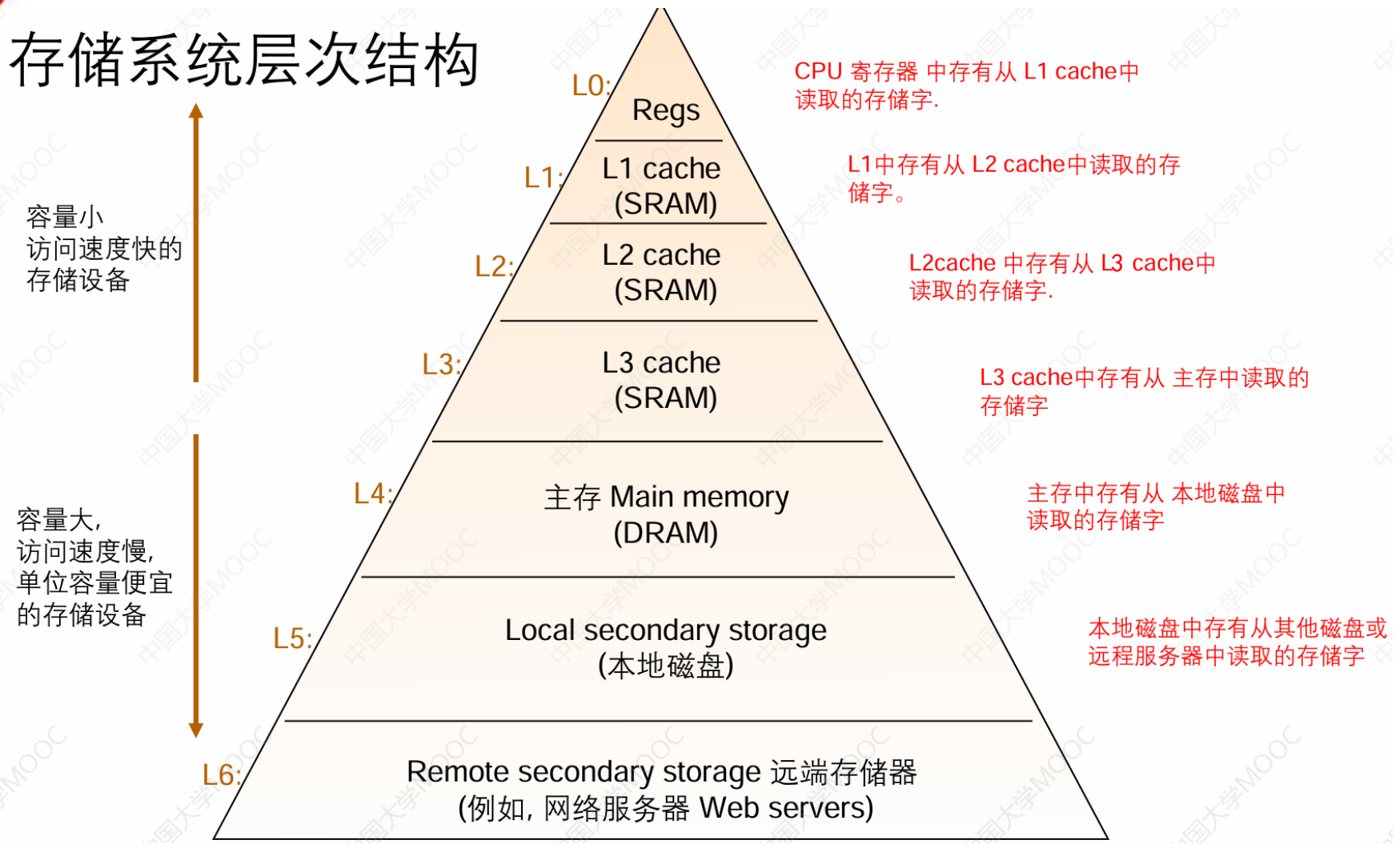
## 3.1.2 存储系统的层次结构

### 程序局部性原理

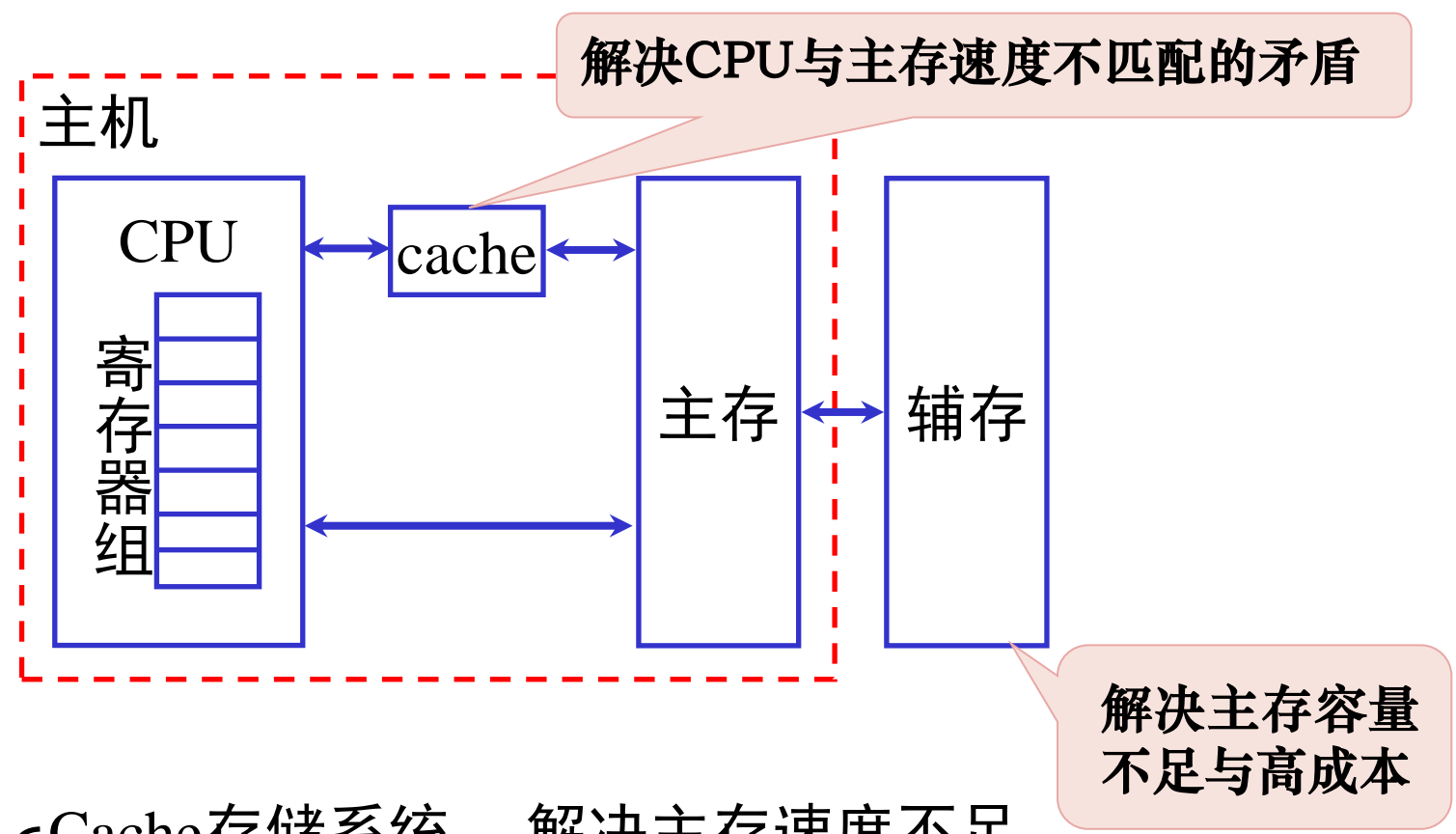
- 大量典型程序的运行情况分析结果显示：
  - 在较短时间间隔内，程序产生的访存地址往往集中在一个小范围内，这种现象称为**程序访问的局部性**；
    - **时间局部性**：刚被访问过的单元很有可能很快被再次访问
    - **空间局部性**：刚被访问过的单元其邻近单元很有可能很快被访问
- 程序具有**访问局部性特征**
  - 程序：指令按序存放，地址连续；循环程序段或子程序段重复执行；
  - 数据：连续存放，数组元素重复、按序访问
- 利用**程序局部性原理**构建**层次化的存储系统**。

# 3.1.2 存储系统的层次结构

存储系统层次结构



### 3.1.2 存储系统的层次结构



- Cache存储系统      解决主存速度不足
- 虚拟存储系统      解决主存容量不足

## 3.2 主存储器

3.2.1

主存储器

3.2.2

半导体存储器

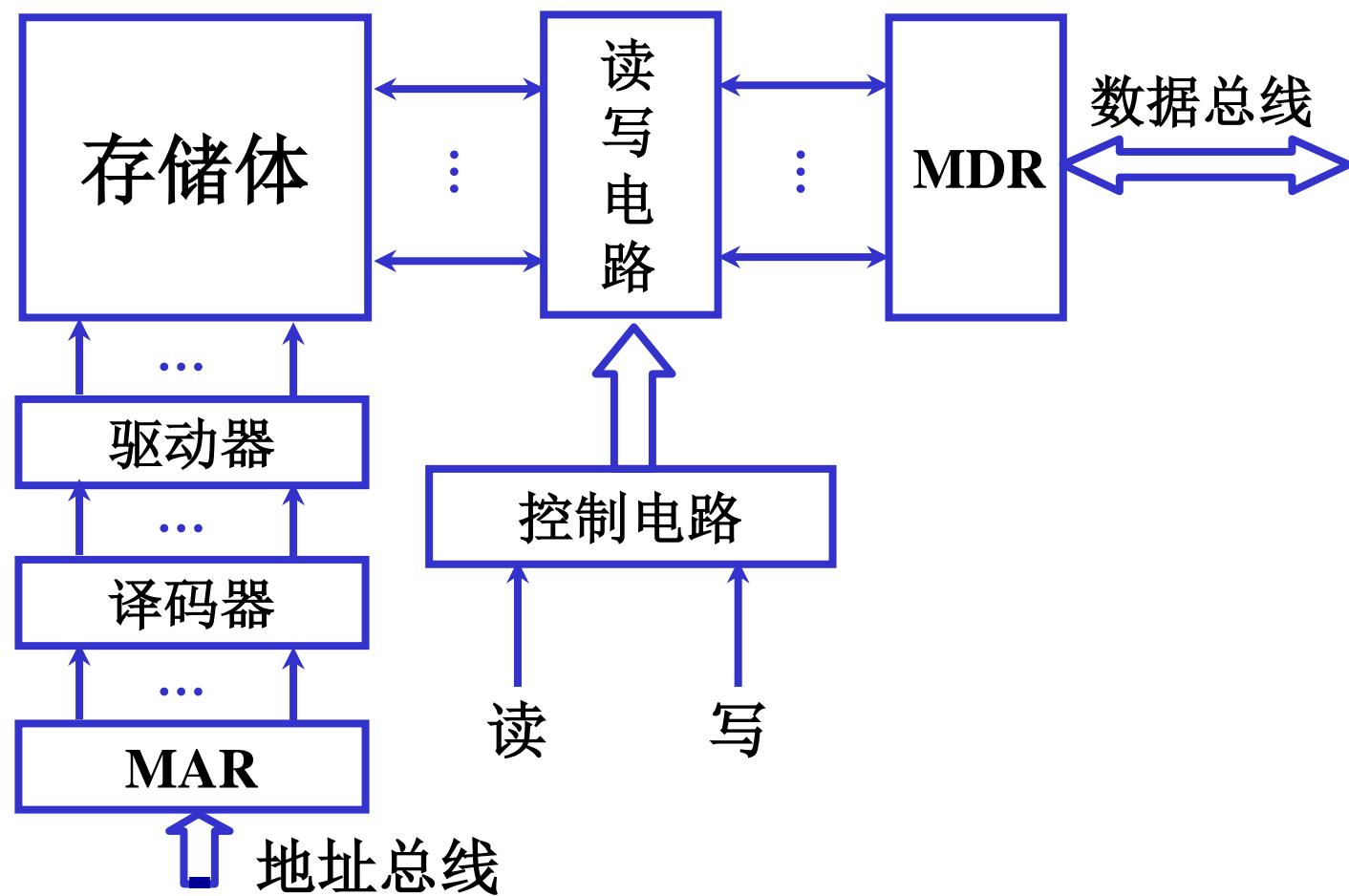
3.2.3

主存储器的设计与扩展



# 3.2.1 主存储器

## 1.主存的基本组成



## 3.2.1 主存储器

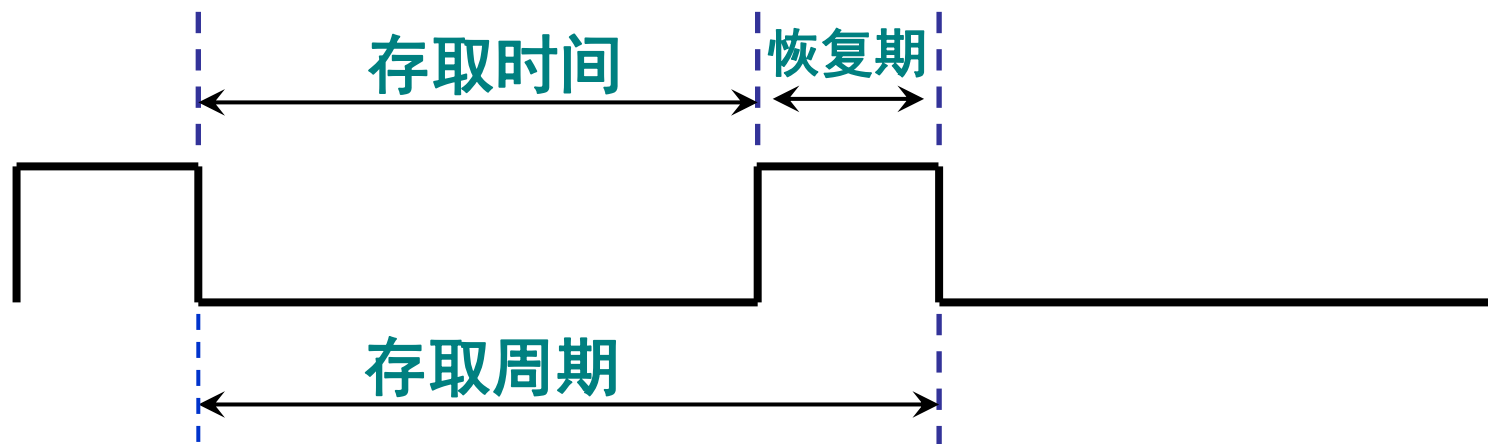
### 2.主存的技术指标

(1)存储容量 主存 存放二进制代码的总位数

常用单位: B(Byte)、KB( $2^{10}$ )、MB( $2^{20}$ )、GB( $2^{30}$ )、TB( $2^{40}$ )

(2)存取时间 从存储器收到读写命令, 到存储器读出 (写入) 信息所需要的时间,  $T_A$

(3)存取周期 存储器做连续访问操作过程中一次完整的存取操作所需的总时间,  $T_M$  (通常  $T_M > T_A$ )





## 3.2.1 主存储器

### 2.主存的技术指标

#### (4)数据传输率-R

单位时间内存取信息的数据量，也叫**带宽**或**频宽**

$$\text{数据传输率(R)} = \frac{\text{存储器的位宽}}{\text{存取周期}} \text{ bps}$$

**【例】**某双通道DDR-4内存传输频率为3200MHz，位宽64比特，则其有效带宽为：

$$R_{\text{DDR-4}} = (64\text{b} \times 3200\text{MHz} \div 8) \times 2 = 51.2\text{GBps}$$

## 3.2.1 主存储器

- 某存储器按字节编址，要求数据传输率达到 $8 \times 10^6$ 字节/秒，则应该选用存储周期为（D）的存储芯片。
- A. 800ns
- B. 250ns
- C. 200ns
- D. 120ns



### 3.2.2 半导体存储器



## 3.2.2 半导体存储器

**TTL**(Transistor-Transistor Logic), 晶体管-晶体管逻辑

输出 L:  $<0.8V$  ; H:  $>2.4V$ 。

输入 L:  $<1.2V$  ; H:  $>2.0V$

**ECL**(Emitter Couple Logic), 射极耦合逻辑

**MOS**(Metal-Oxide Semiconductor), 金属氧化物半导体, 即场效应管)

**CMOS**(Complementary Metal-Oxide Semiconductor),  
即 互补对称金属氧化物半导体

输出 L:  $<0.1 \times V_{cc}$  ; H:  $>0.9 \times V_{cc}$

输入 L:  $<0.3 \times V_{cc}$  ; H:  $>0.7 \times V_{cc}$

## 3.2.2 半导体存储器

### ※静态-动态存储器的特点

#### (1) 静态存储器(Static RAM)

主要包括：双极型、静态MOS型。

依靠双稳态电路内部交叉反馈机制存储信息。

功耗较大, 速度快, 常用作Cache。

#### (2) 动态存储器(Dynamic RAM)

主要包括：动态MOS型等；

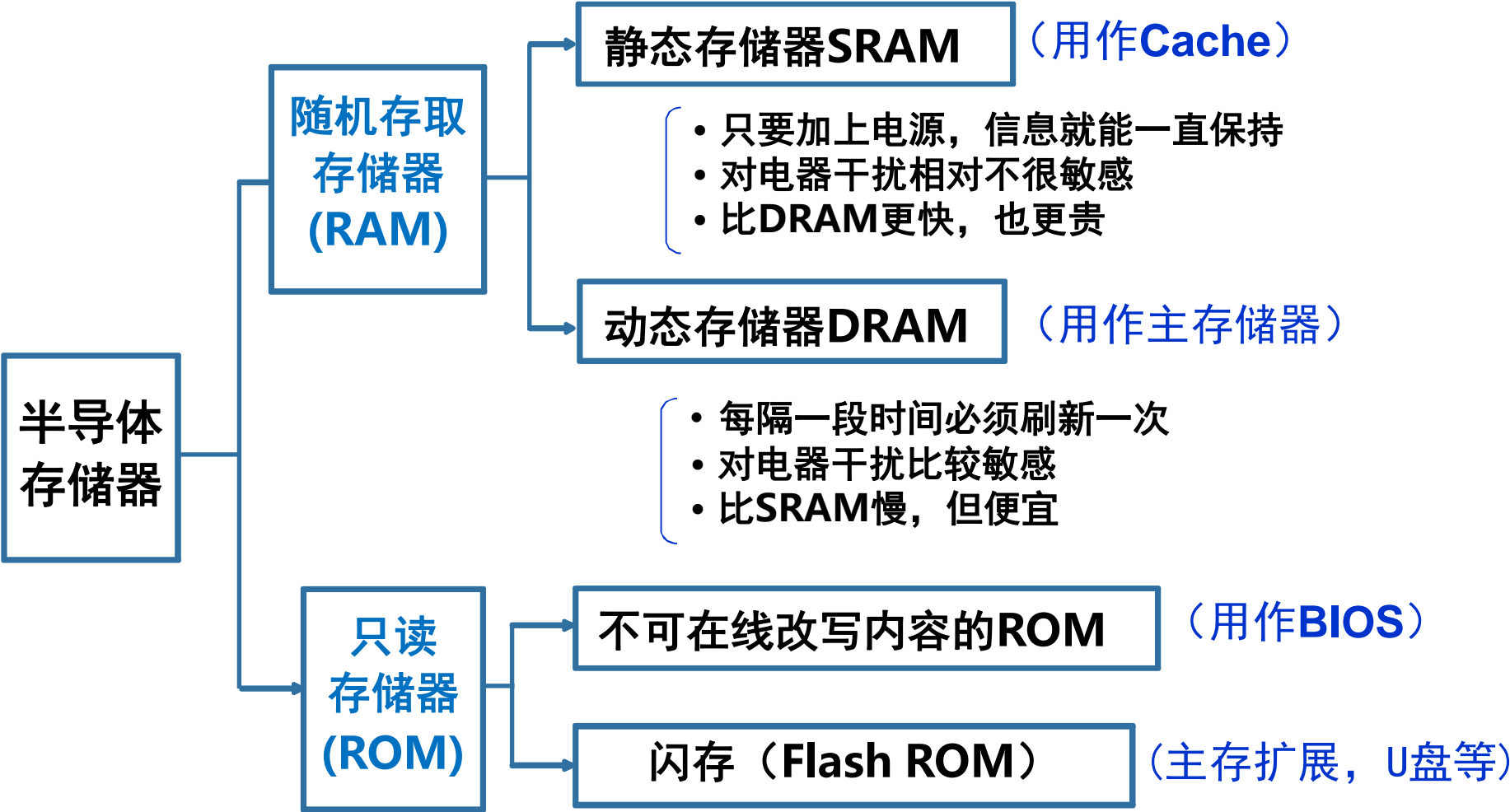
依靠电容存储电荷的原理存储信息。

功耗较小, 容量大, 速度较快, 常用作主存。



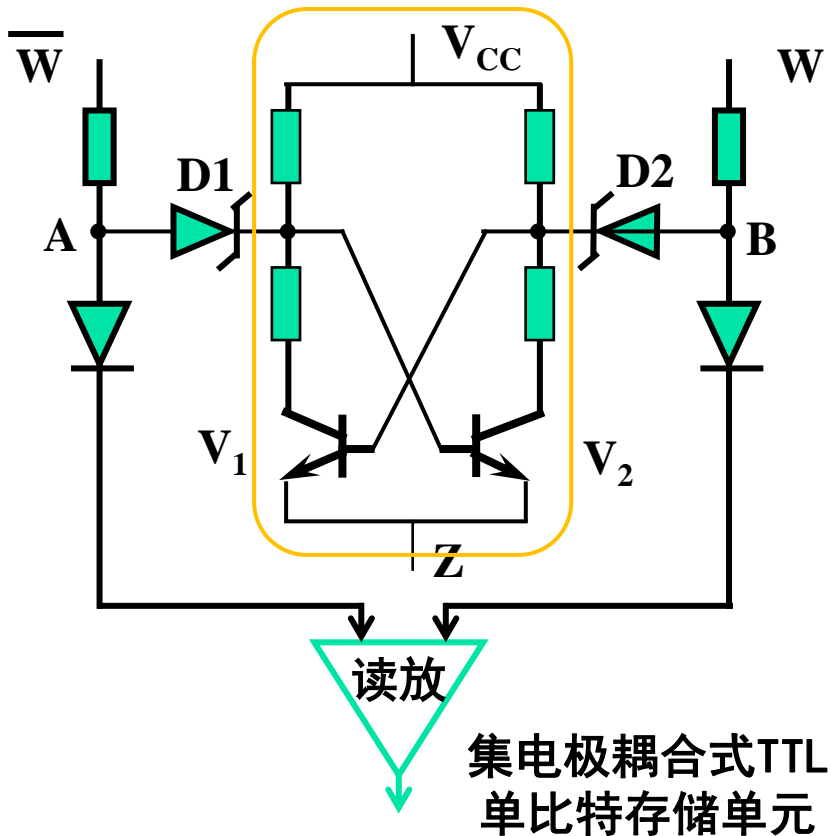
## 3.2.2 半导体存储器

内存由半导体存储器芯片组成，芯片有多种类型



# 3.2.2 半导体存储器存储原理

## (一) 双极型存储单元与芯片



$V_1$ 导通,  $V_2$ 截止—0

$V_1$ 截止,  $V_2$ 导通—1

选中: Z线 0.3V(负脉冲)

(1) 写入数据

0: 左低、右高

1: 左高、右低

(2) 数据保持(Z加高电平)

(3) 读出数据(Z加低电平)

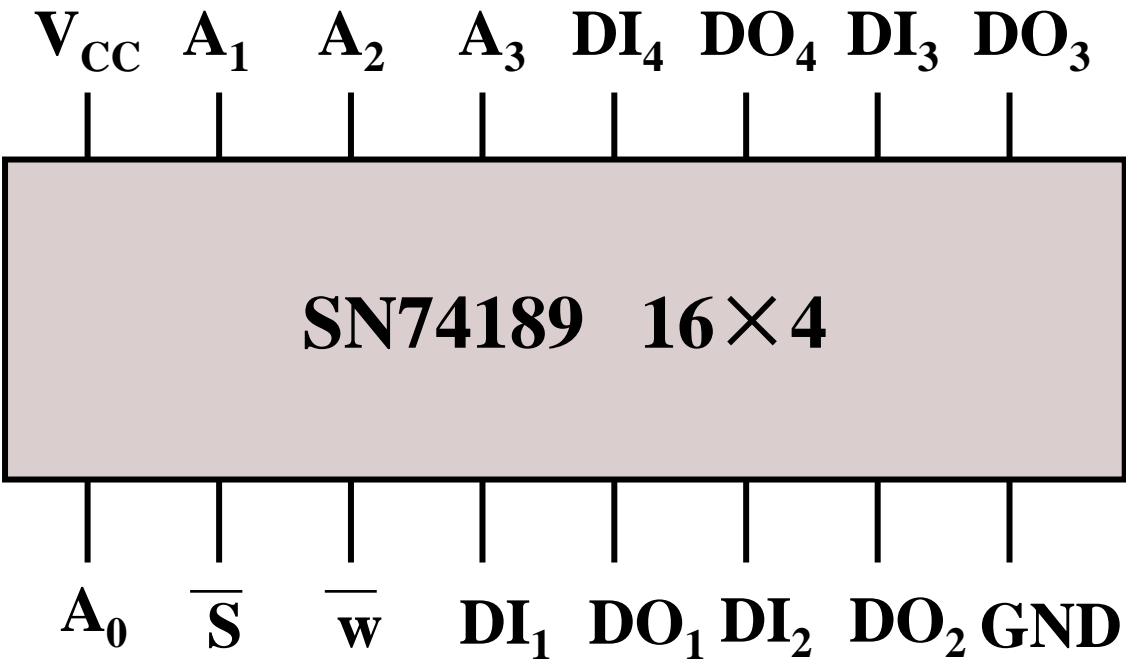
左低、右高→0

左高、右低→1



## 3.2.2 半导体存储器存储原理

### TTL型存储器芯片举例（SN74189）



- $V_{CC}$ : 电源
- $A_{0\sim3}$ : 地址
- $DI_{1\sim4}$ : 数据输入
- $DO_{1\sim4}$ : 数据输出
- GND: 接地线
- $\overline{S}$ : 片选信号
- $\overline{W}$ : 读\写信号

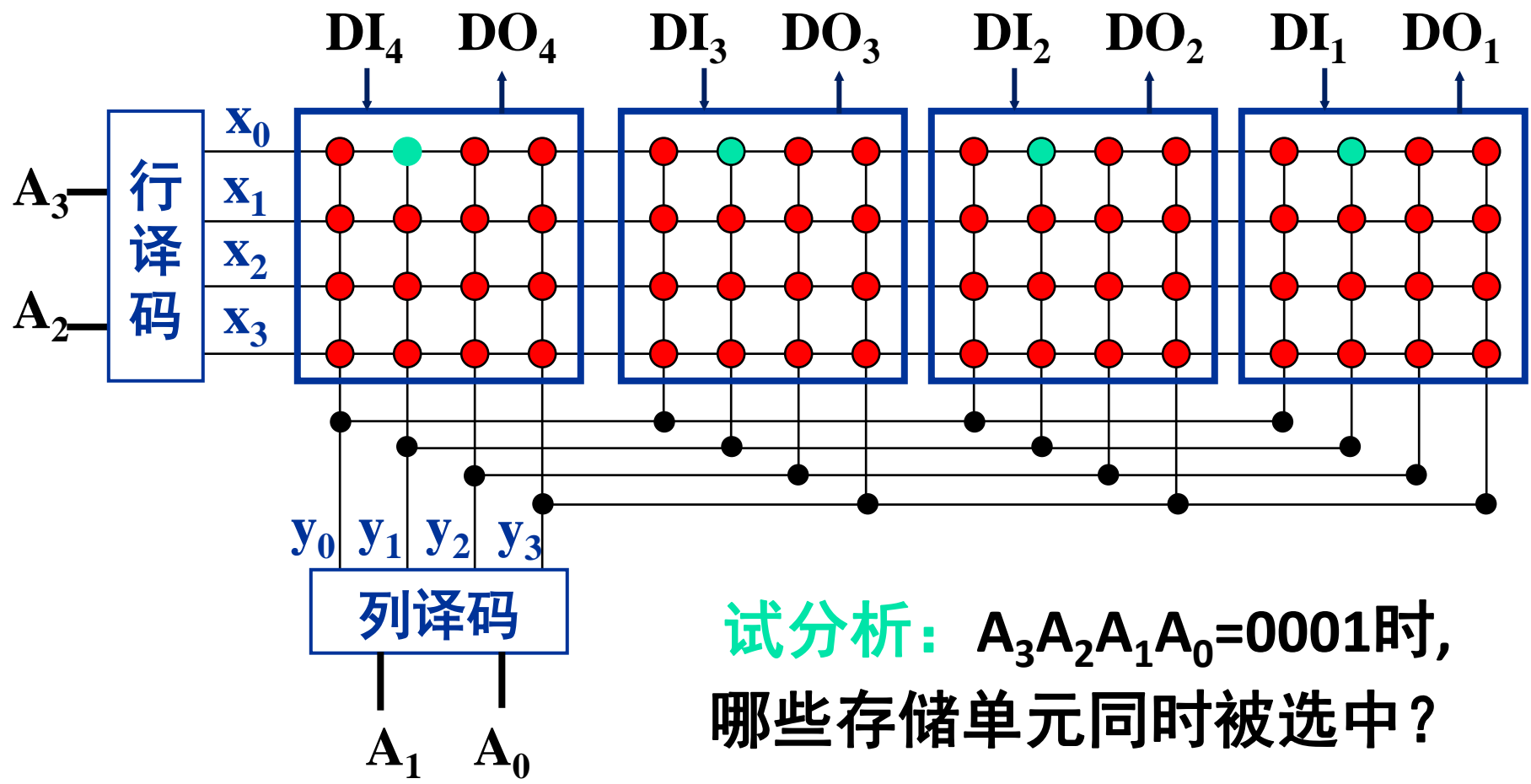
SN74189芯片引脚图





# 3.2.2 半导体存储器存储原理

## 内部四个位平面的行列译码结构



试分析：A<sub>3</sub>A<sub>2</sub>A<sub>1</sub>A<sub>0</sub>=0001时，  
哪些存储单元同时被选中？

3.2.2 半导体存储器存储原理

(二) 静态MOS存储单元与芯片

(1) 组成

T1、T3：MOS反相器

T2、T4：MOS反相器

触发器

T5、T6：控制门管

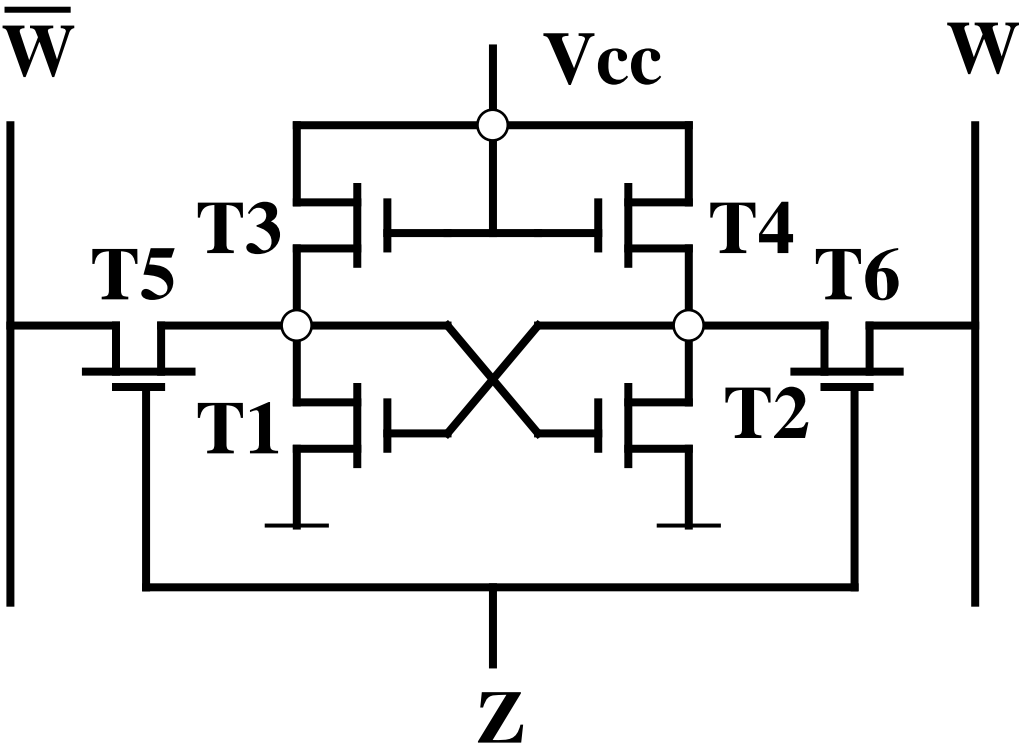
Z：字线，选择存储单元(高电平)

$\overline{W}$ 、W：位线，完成读写

(2) 定义

“0”：T1导通，T2截止；

“1”：T1截止，T2导通。





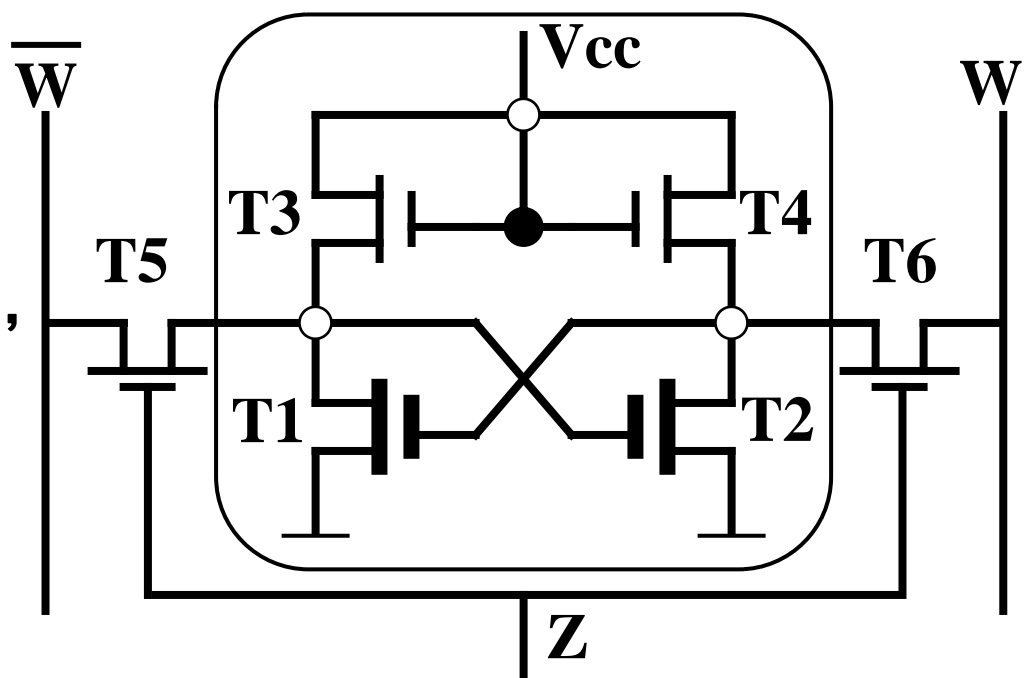
# 3.2.2 半导体存储器存储原理

## (3) 工作

**Z:** 加高电平, T5、T6导通  
选中该单元。

**写入:** 在 $\overline{W}/W$ 上分别加高/低电平,  
写1/0。

**读出:** 充电后根据 $\overline{W}/W$ 上有/无电  
流, 读出0/1。



## (4) 数据保持

**Z:** 加低电平, T5/T6截止, 该单元未选中, 保持原状态

T1导通, T2截止—0; T1截止, T2导通—1

只要电源正常, 保证向导通管提供电流, 便能维持一管导通、另一管截止的状态不变, 故称静态存储单元。

静态单元是非破坏性读出, 读出后不需重写。



# 3.2.2 半导体存储器存储原理

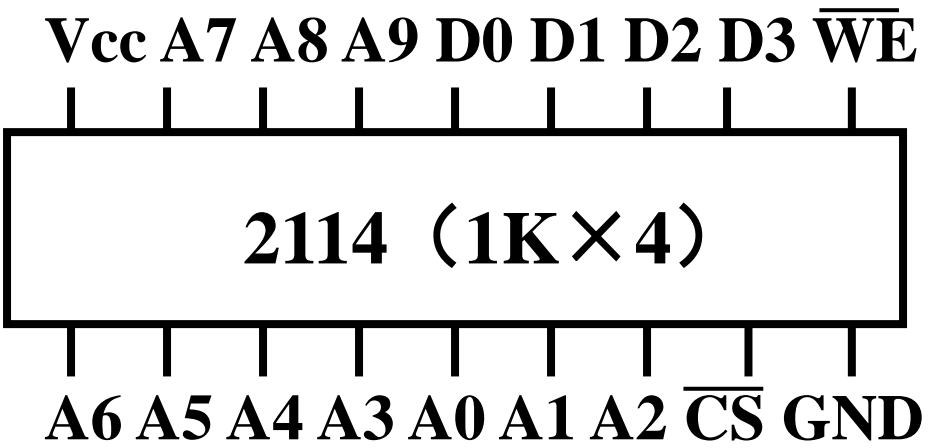
## 2. 存储芯片举例（SRAM芯片 Intel 2114）

地址端： A9~A0（入）

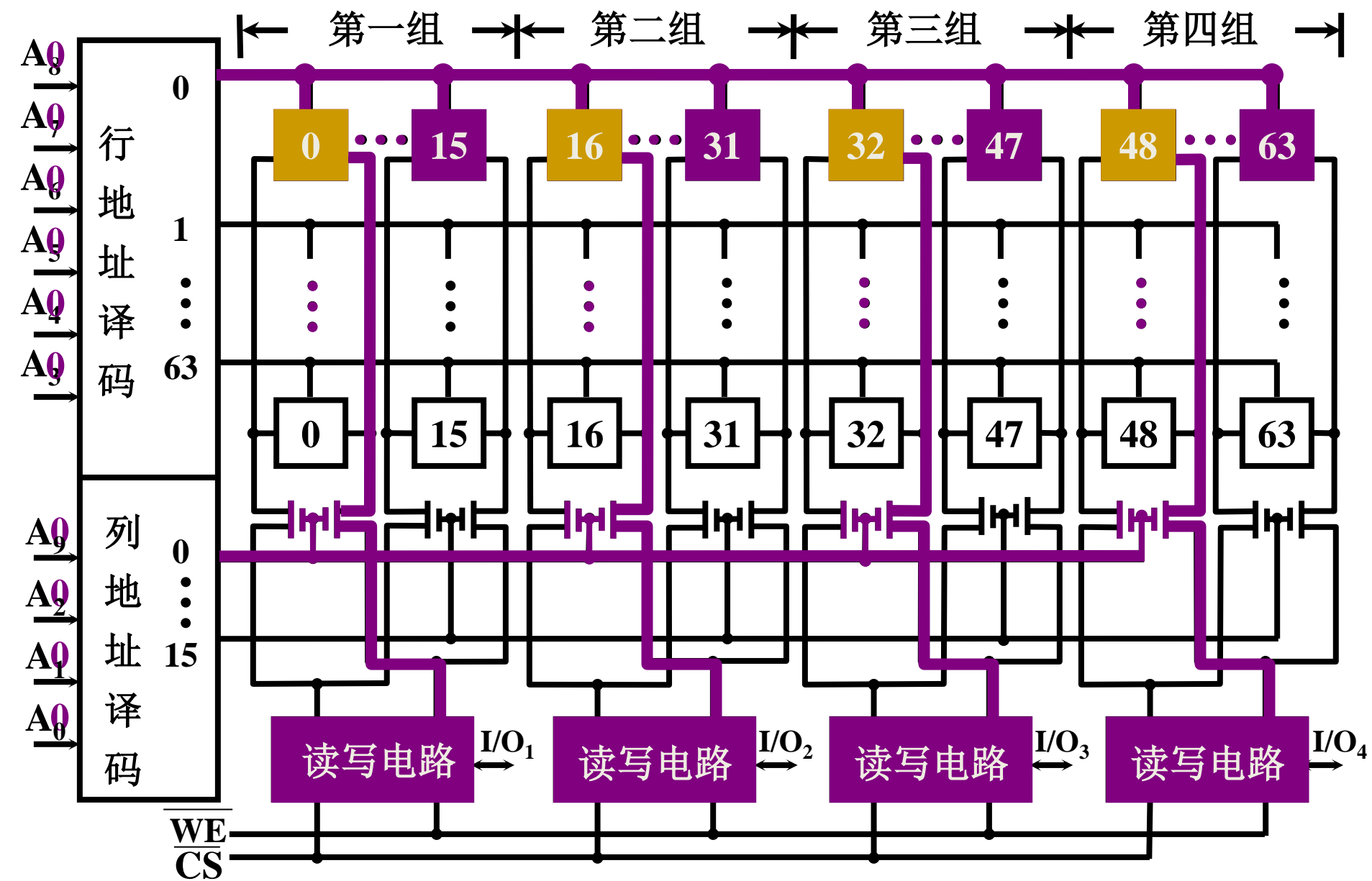
数据端： D3~D0（入/出）

控制端： $\left\{ \begin{array}{l} \text{片选} \overline{CS} \left\{ \begin{array}{l} = 0 \text{ 选中芯片} \\ = 1 \text{ 未选中芯片} \end{array} \right. \\ \text{写使能} \overline{WE} \left\{ \begin{array}{l} = 0 \text{ 写} \\ = 1 \text{ 读} \end{array} \right. \end{array} \right.$

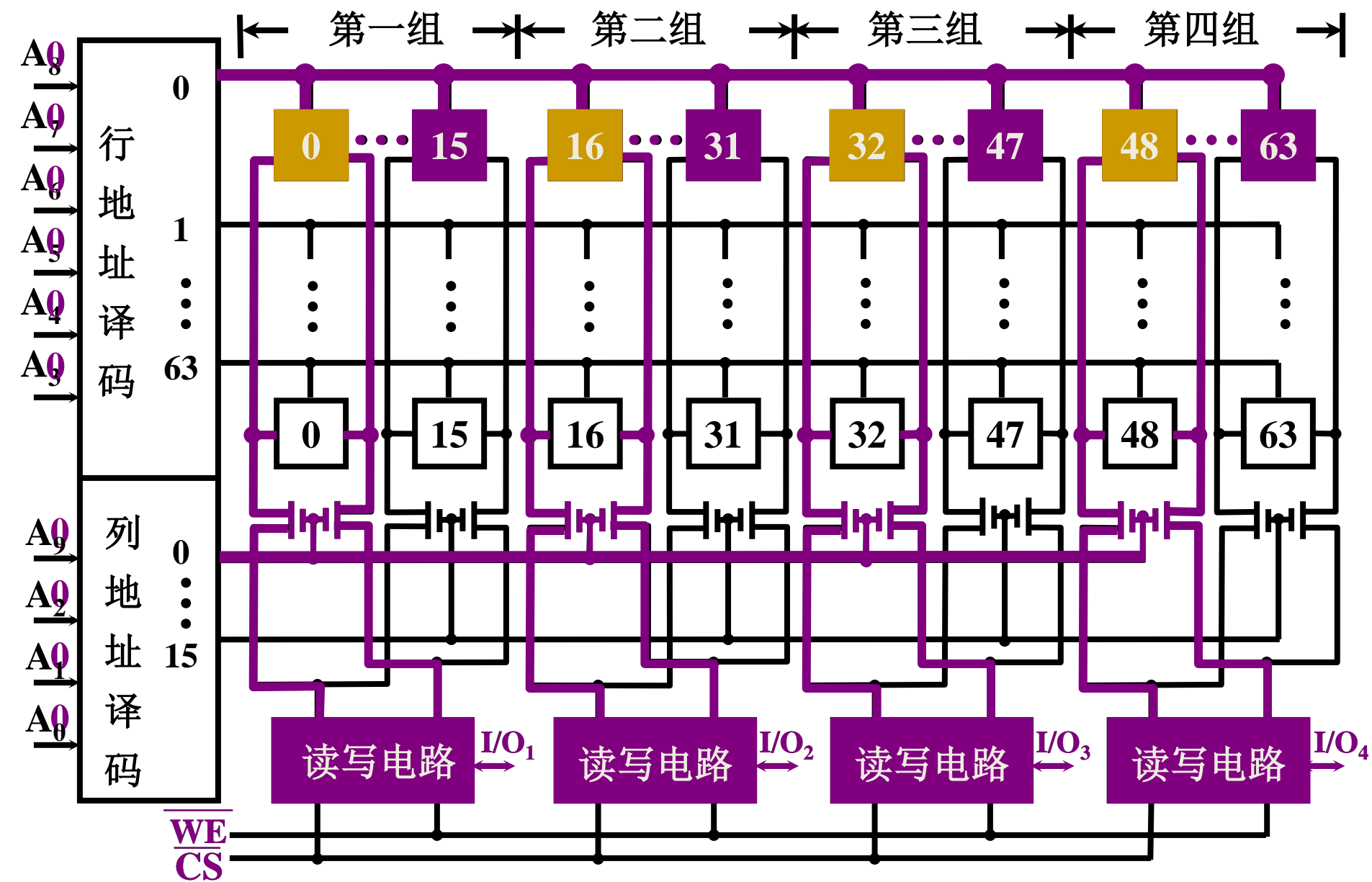
VCC：电源， GND：接地



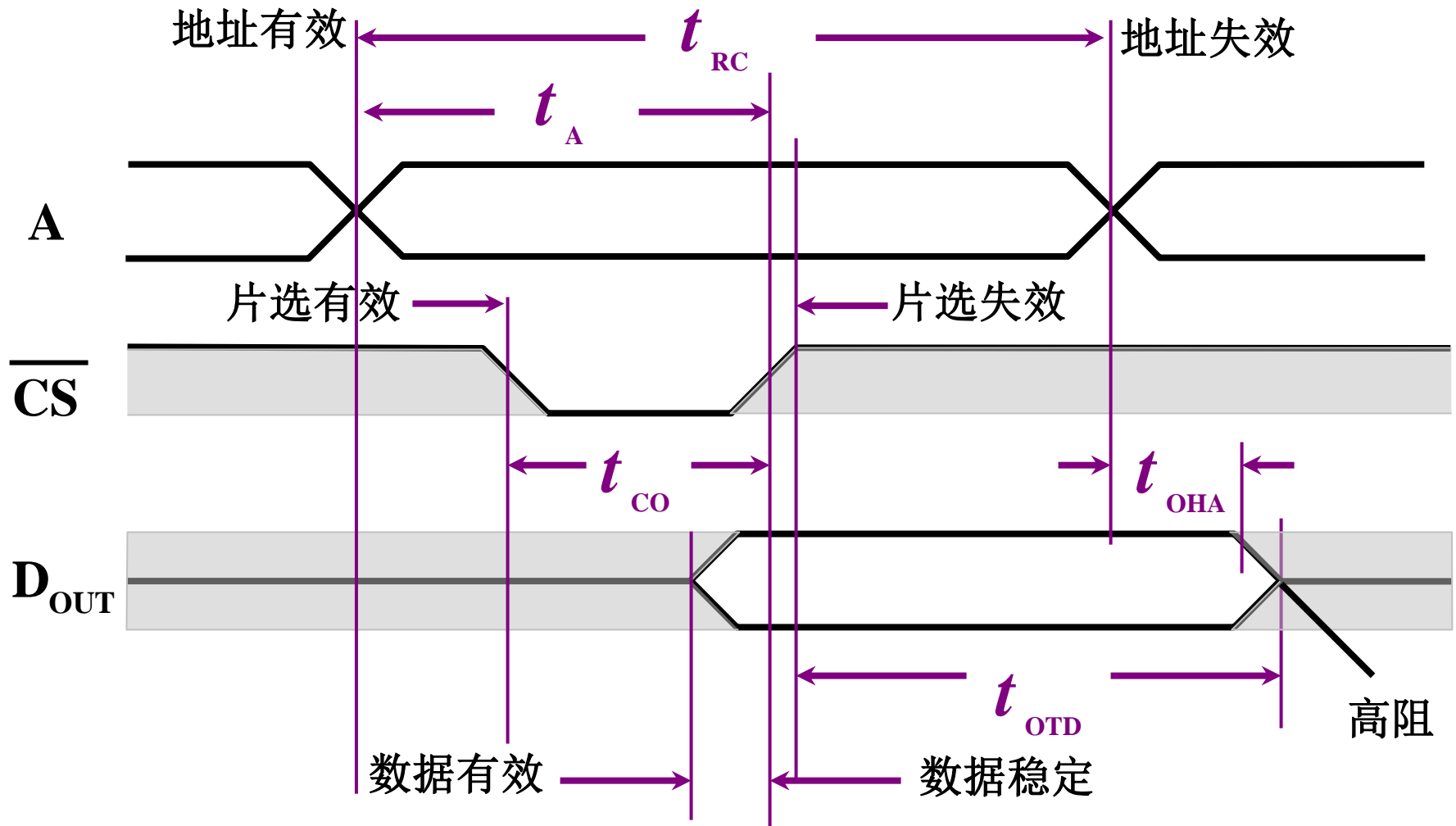
# Intel 2114 RAM 矩阵 (64 × 64) 读



# Intel 2114 RAM 矩阵 (64 × 64) 写

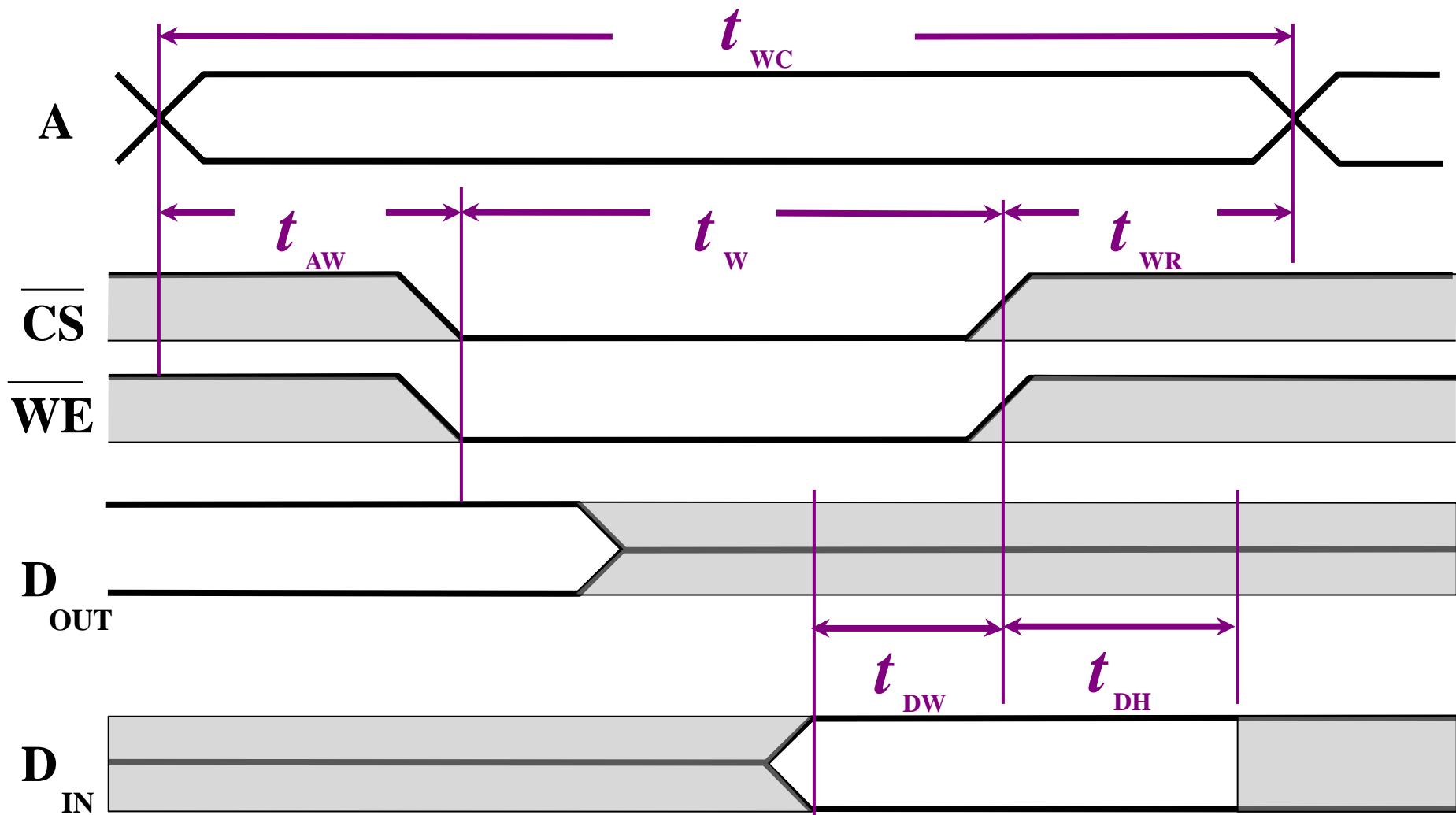


### 3. 读写时序 读周期



$t_{OHA}$  地址失效后的数据维持时间

# • 写周期

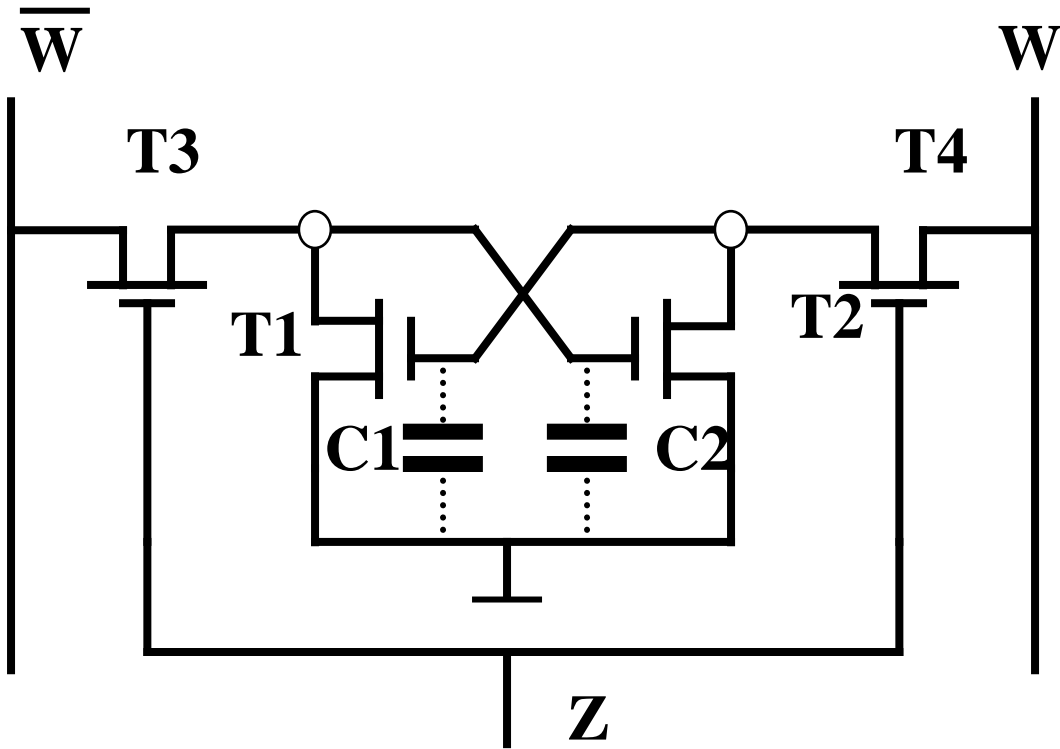


$t_{\text{DH}}$   $\overline{\text{WE}}$  失效后的数据维持时间



## 3.2.2 半导体存储器存储原理

### (三) 动态MOS存储单元与芯片



动态MOS四管存储单元

#### 1.四管单元电路

##### (1)结构

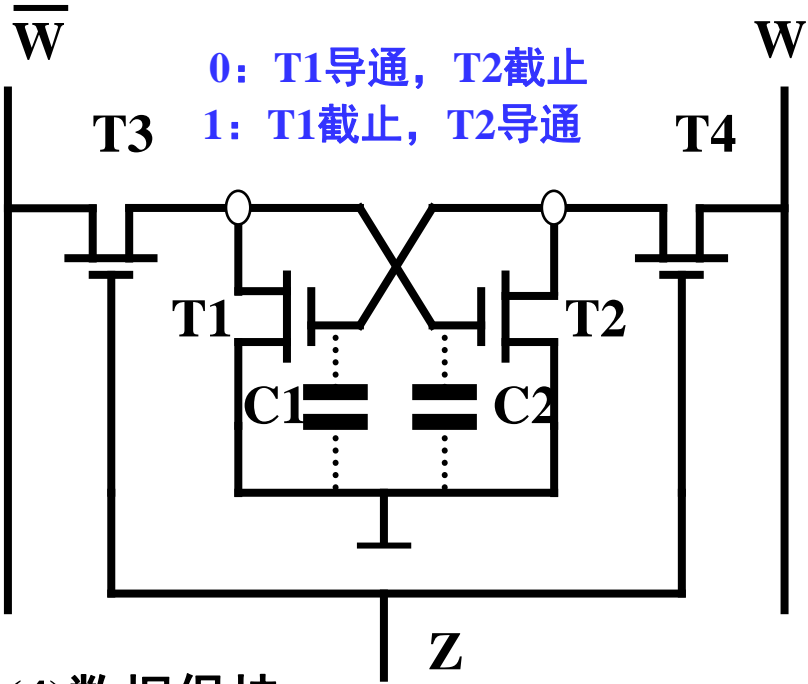
- T1、T2：记忆管
- C1、C2：栅极电容
- T3、T4：控制门管
- Z：字线
- $\overline{W}$ 、W：位线

##### (2)定义

- 0：T1导通，T2截止  
(C1带电荷，C2无电荷)
- 1：T1截止，T2导通  
(C1无电荷，C2带电荷)

## 3.2.2 半导体存储器存储原理

### (三) 动态MOS存储单元与芯片



#### (4) 数据保持

Z: 加低电平, T3/T4截止, 该单元未选中, 保持原状态。

需定期向电容补充电荷（动态刷新），故称动态存储器

四管单元是非破坏性读出，读出过程即可实现刷新。

#### (3) 工作

Z: 加高电平, T3、T4导通, 选中该单元。

写入: 在 $\overline{W}/W$ 上分别加高/低电平, 写入1/0。

读出:  $\overline{W}/W$ 先预充电至高电平, 断开充电回路, Z再加高电平, 再根据 $\overline{W}/W$ 上有/无电流, 读出0/1。

读出过程中, 通过 $\overline{W}/W$ 分别向C1和C2补充电荷。

## 3.2.2 半导体存储器存储原理

### (三) 动态MOS存储单元与芯片

#### 2. 单管动态存储单元

##### (1) 组成

C: 记忆单元    T: 控制门管

Z: 字线        W: 位线

##### (2) 定义

“0”: C无电荷, 电平V0 (低)

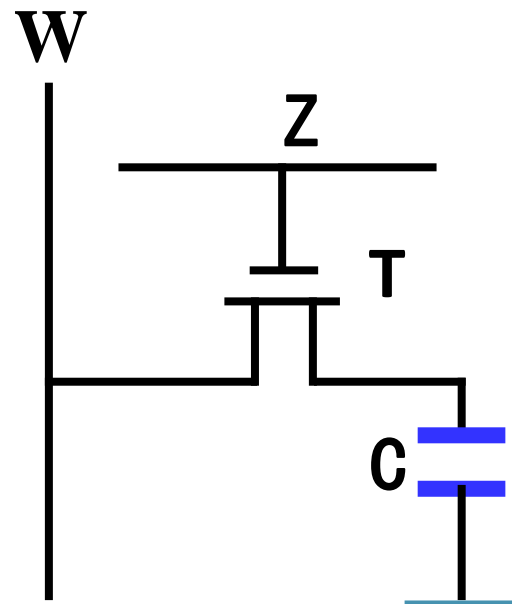
“1”: C有电荷, 电平V1 (高)

##### (3) 写入

Z加高电平, 使T导通, 在W上加高/低电平, 写1/0

##### (4) 保持

Z加低电平, 使T截止, 该单元未选中, 保持原状态。



## 3.2.2 半导体存储器存储原理

### (三) 动态MOS存储单元与芯片

#### (5) 读出数据

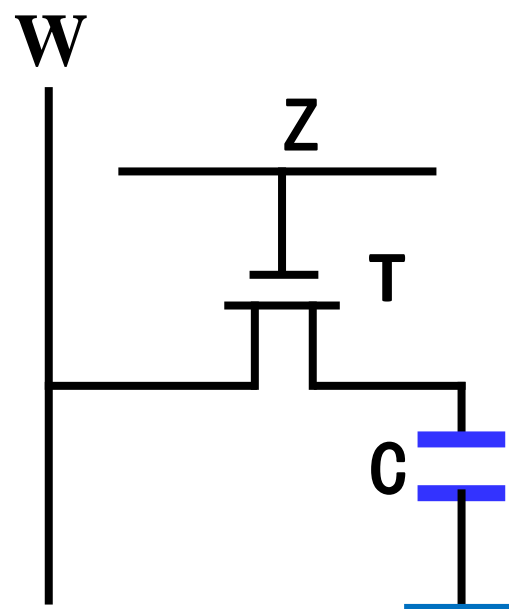
W预充至高电平后，断开充电回路，

Z加高电平,使T导通

根据W线上是/否产生读电流，判断读到数据 1/0

读操作，使C上电荷释放

单管单元是破坏性读出，读出后需重写。



数据以电荷形式保存在电容中，电容漏电使得电荷通常只能维持2ms左右，因此要定期在2ms内刷新(读出后重新写回)

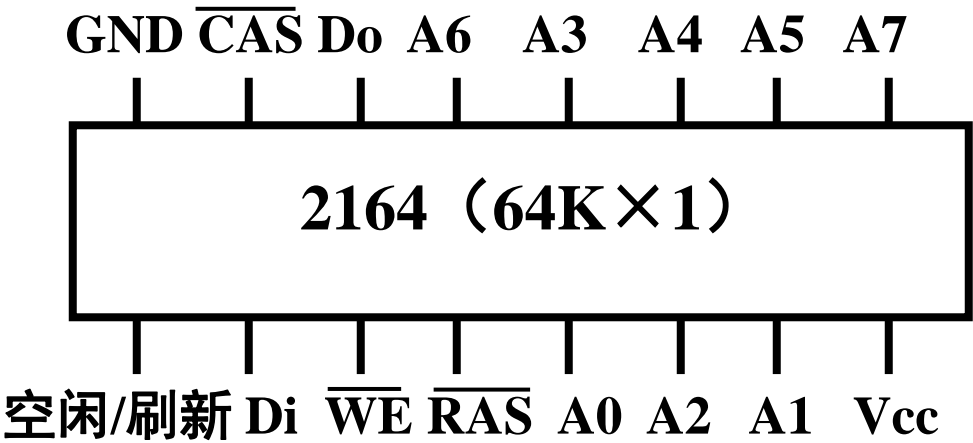
## 3.2.2 半导体存储器存储原理

### DRAM芯片2164

数据端：Di(输入),Do(输出)

地址端：A7~A0

(分时复用，可提供16位地址)



控制端：写使能信号( $\overline{WE}$ ): 0-写入; 1-读出;

行选 $\overline{RAS}=0$ 时：A<sub>7</sub>~A<sub>0</sub>为行地址(即高8位)

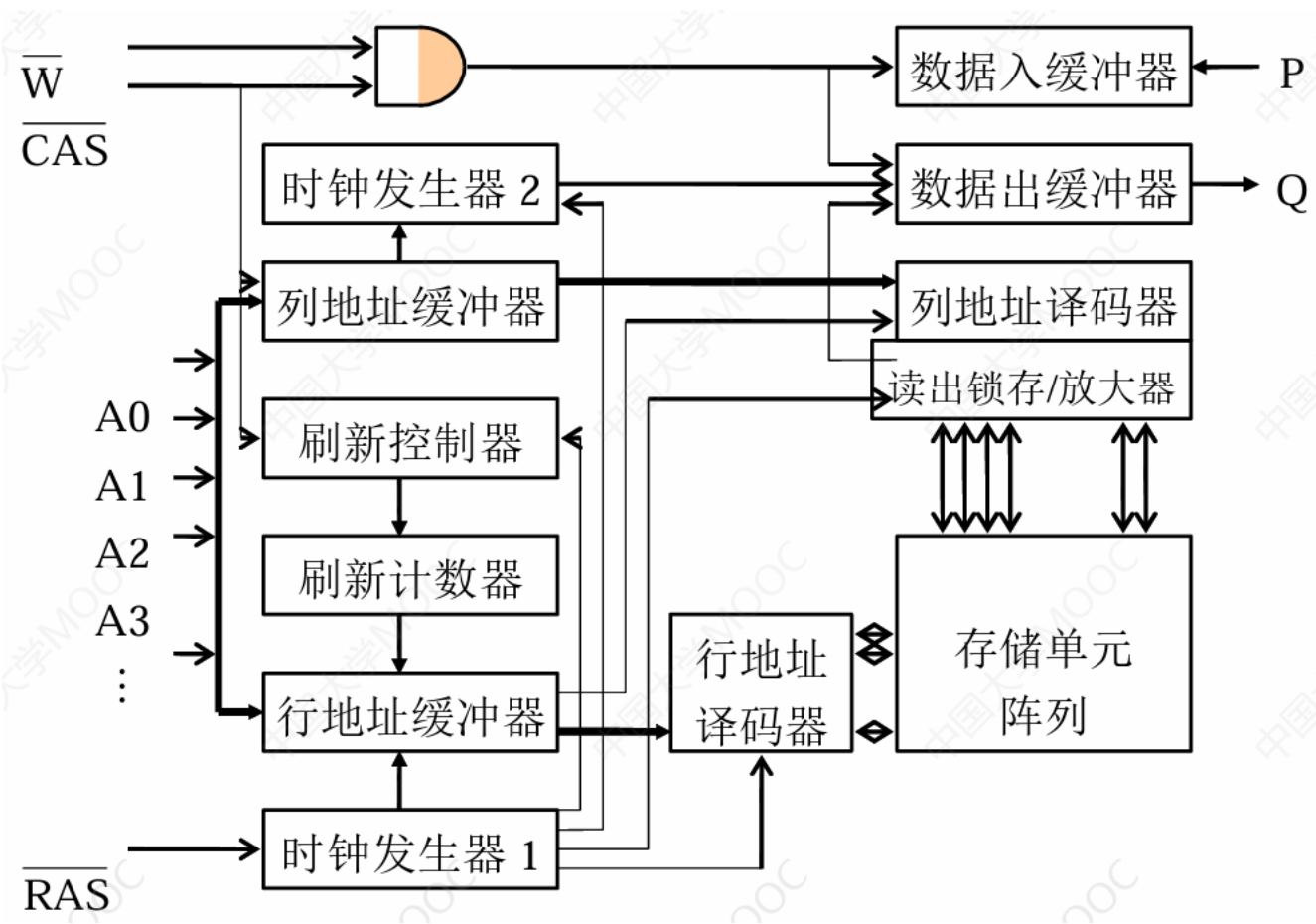
列选 $\overline{CAS}=0$ 时：A<sub>7</sub>~A<sub>0</sub>为列地址(即低8位)

动态刷新：行选信号送达，即可实现自动刷新;



# 3.2.2 半导体存储器存储原理

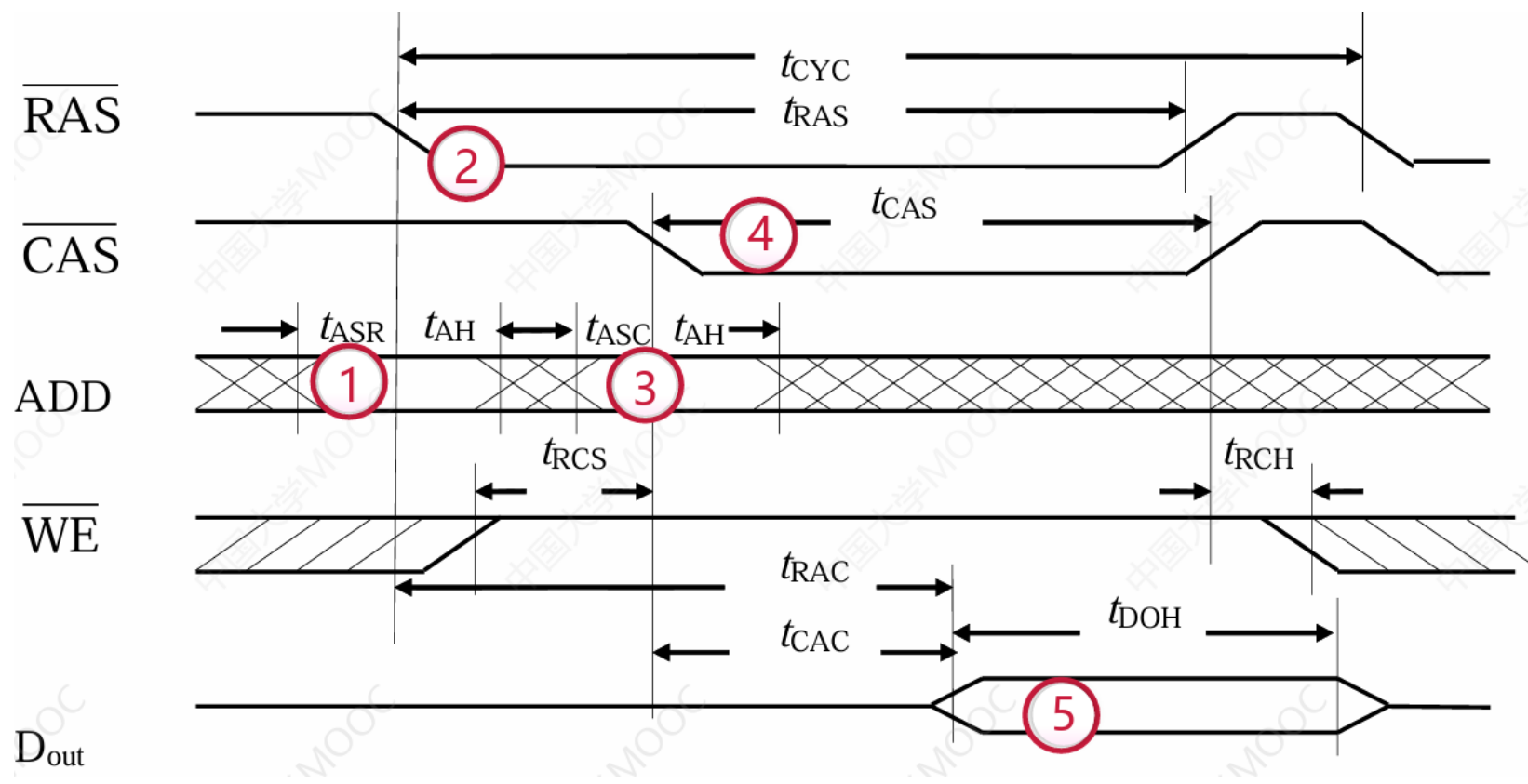
## DRAM芯片的结构



# 3.2.2 半导体存储器存储原理

## DRAM时序（读周期）

行地址有效→行地址选通→列地址有效→列地址选通→数据输出

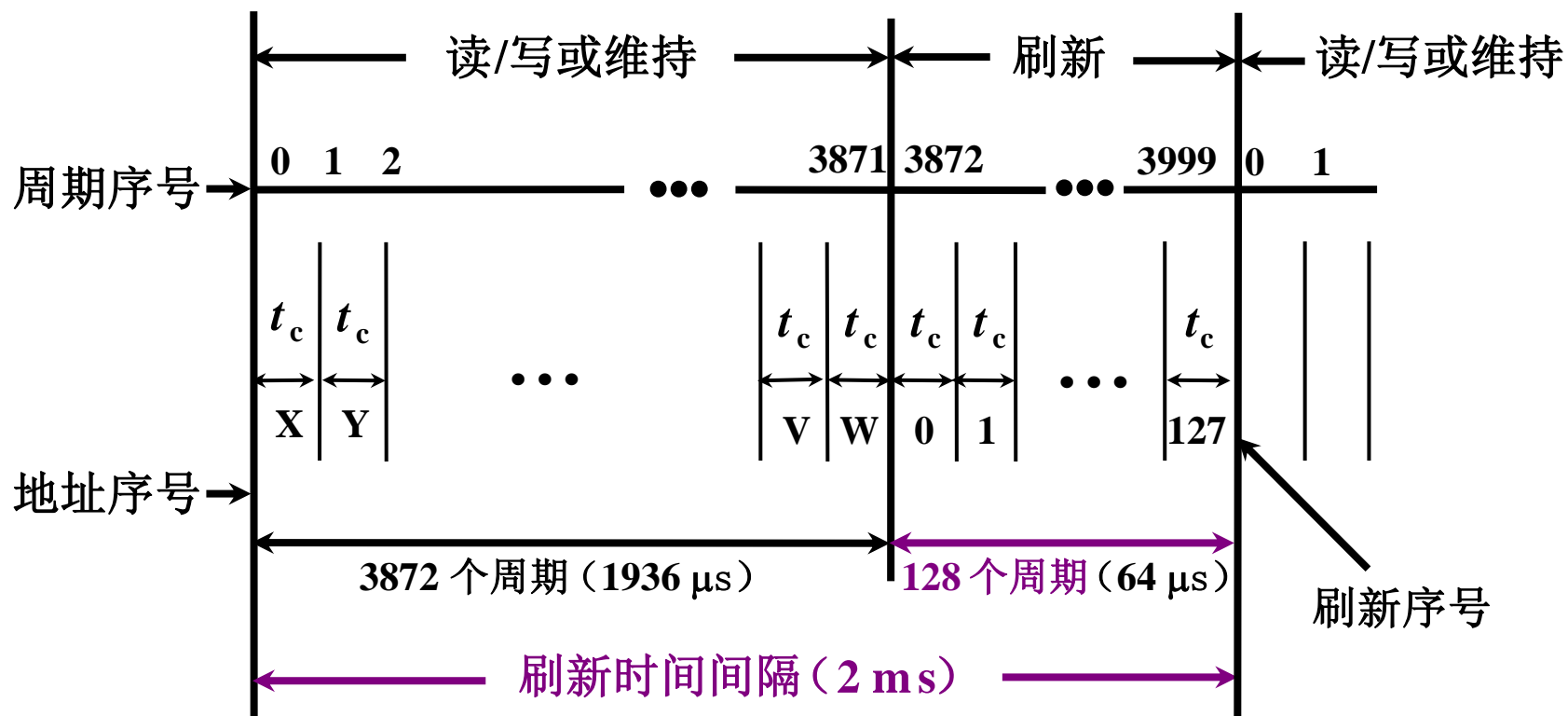




# DRAM的刷新方式

## 刷新与行地址有关

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



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

“死时间率” 为  $128/4000 \times 100\% = 3.2\%$

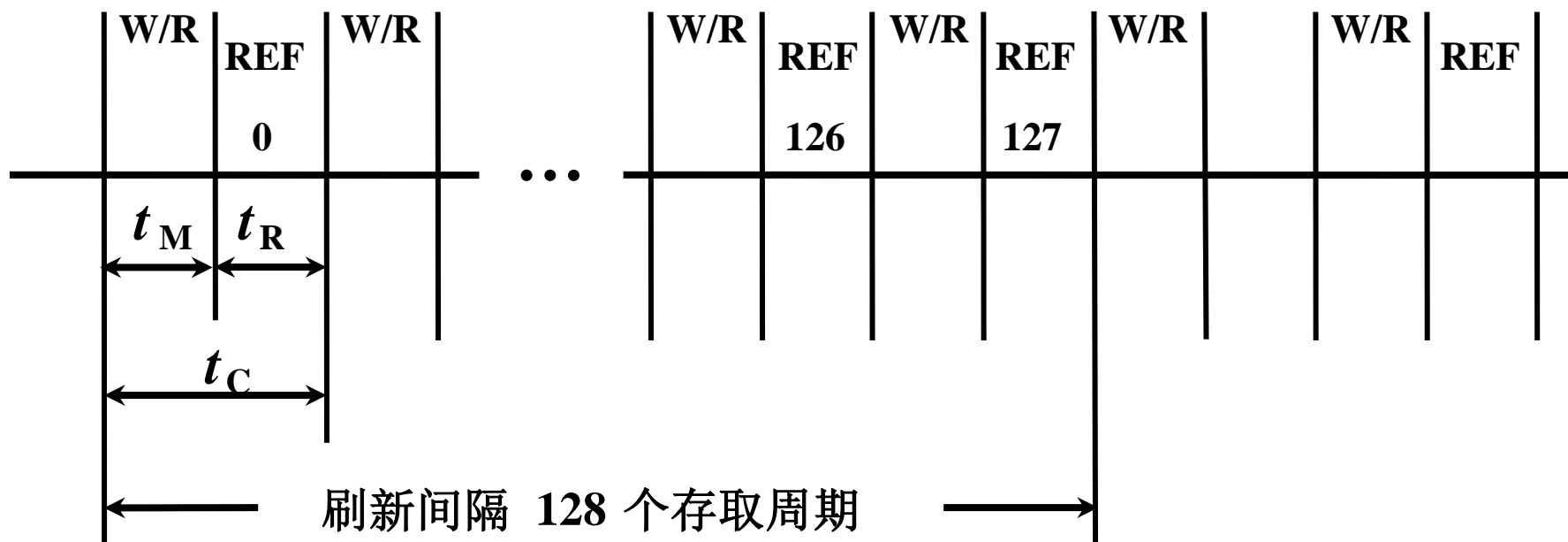




# DRAM的刷新方式

## ② 分散刷新（存取周期为 $1\ \mu\text{s}$ ）

以  $128 \times 128$  矩阵为例



$$t_C = t_M + t_R$$

无“死区”

读写 刷新

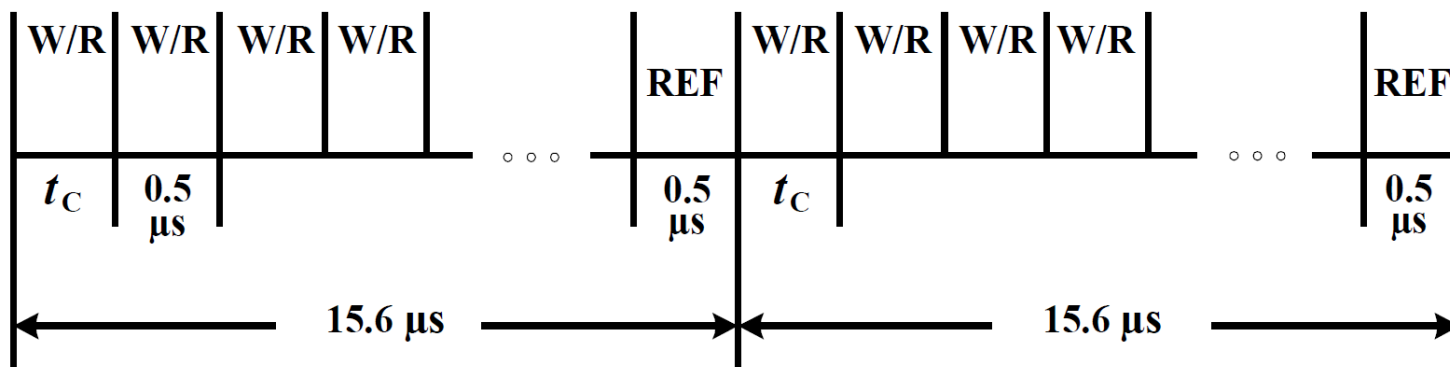
(存取周期为  $0.5\ \mu\text{s} + 0.5\ \mu\text{s}$ )

## DRAM的刷新方式

### 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



每行每隔  $2 \text{ ms}$  刷新一次 “死区” 为  $0.5 \mu\text{s}$

将刷新安排在指令译码阶段，不会出现 “死区”

## DRAM的刷新

**[例1]** 有一个 $16K \times 16$ 的存储器，由 $1K \times 4$ 的DRAM芯片（内部结构为 $64 \times 64$ ）构成，问：

(1) 采用异步刷新方式，如最大刷新间隔为 $2ms$ ，则相邻两行之间的刷新间隔是多少？

(2) 采用集中刷新方式，存储器刷新一遍最少用多少个刷新周期？设存储器的读写周期为 $0.5\mu s$ ，死区占多少时间？死时间率为多少？

分析：

$2ms$ 内刷新64行，相邻两行之间的刷新间隔为： $2ms/64=31.25\mu s$

集中刷新方式，刷新一遍最少用64个刷新周期

死区= $0.5\mu s \times 64=32\mu s$

死时间率= $32\mu s/2000\mu s \times 100\%=1.6\%$

## DRAM的刷新

**[例2]** 一个 $128 \times 128$ 结构的DRAM芯片，每隔2ms要刷新一次，且刷新是按顺序对所有128行的存储单元进行内部读操作和写操作实现的。设存取周期为 $0.5\mu\text{s}$ ，求刷新开销（即进行刷新操作的时间所占的百分比）

分析：

刷新过程是对每行的存储单元先读后写，所以每行的刷新

时间是： $0.5\mu\text{s} \times 2 = 1\mu\text{s}$

在2ms内进行128次刷新，需 $1\mu\text{s} \times 128 = 128\mu\text{s}$

刷新开销为： $128\mu\text{s} / 2\text{ms} \times 100\% = 6.4\%$



# 动态 RAM 和静态 RAM 的比较

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

## 思考

- 某一SRAM芯片，容量为512\*8位，考虑电源端和接地端，该芯片引脚最小数目为（ ）。  
• A 23    B 25    C 21    D 19
- 假定DRAM芯片中存储阵列的行数是r，列数是c，对于一个2K\*1位的DRAM芯片，为保证其引脚数最少，并尽量减少刷新开销，则r、c的取值分别是（ C ）  
• A 2048、1    B 64、32  
• C 32、64    D 1、2048



## 半导体只读存储器

### 1. 掩模 ROM ( MROM )

**基本原理：**根据存储信息的二进制代码，设计相应的光刻掩模（**1-有元件、0-无元件**）

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”

**特点：**存储的信息固定不变，**不可改写**

**应用：****字符点阵存储器、微程序存储器等。**



# 半导体只读存储器

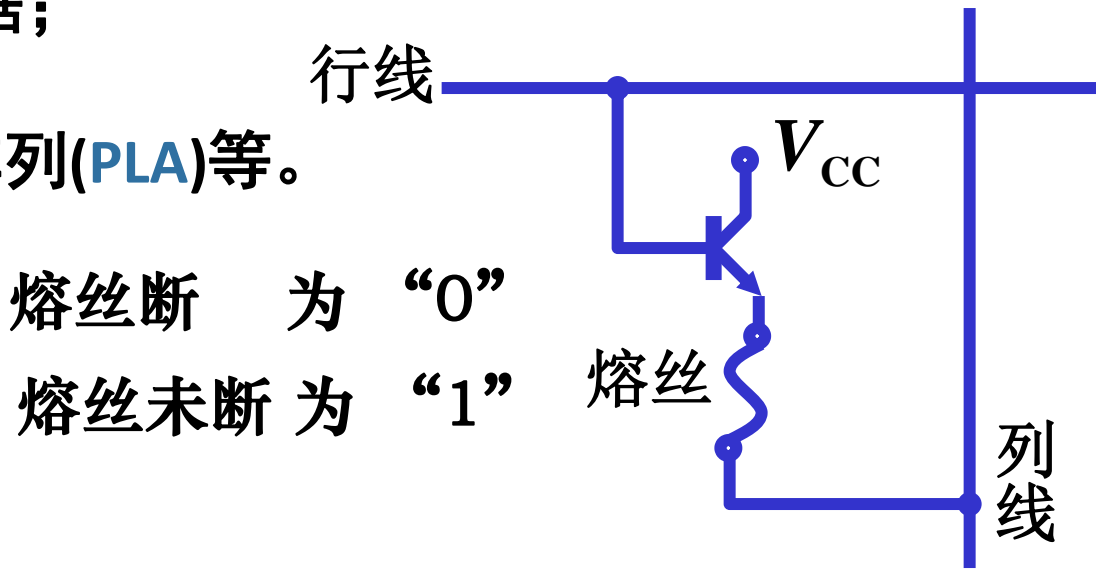
## 2. PROM (一次性编程)

**基本原理：** 芯片出厂时其存储的内容为**全0**，用户可通过专用的写入器将信息自行写入；

**结破坏型或者熔丝型**；

**芯片特点：** 写入操作是**不可逆的**，用户只能写入**1次**，无法再次重写数据；

**应用场合：** 可编程逻辑阵列(**PLA**)等。





## 半导体只读存储器

### 3. EPROM (多次性编程)

即擦除型可编程的只读存储器

基本原理：

※写入器在25v下写入数据，在5v下读数据；

※通过紫外线照射擦除数据；

芯片的特点：

※工作环境下存储芯片为只读模式；

※可擦写次数有限，通常几十次；

※需专用擦写器，只能芯片级擦除；



## 半导体只读存储器

### 4. EEPROM (多次性编程)

即电擦除型可编程只读存储器

基本原理：

※采用了更方便的高电压擦除数据的方式，可只对特定存储单元加高压形成电子隧道擦除其数据，其它单元数据保持不变；

※通常采用金属-氮-氧化硅的集成工艺；

芯片的特点：

※工作环境下存储芯片为只读模式；

※比EPROM更方便；

※需要专用擦写器，可实现比特级擦除；

# 半导体只读存储器

## 5. Flash Memory (闪速型存储器)

即闪存, 是一种快速擦写型的ROM

**基本原理:**

※沿用了EPROM的简单结构和浮栅/热电子注入写入方式,  
可**芯片级**擦除;

※兼备EEPROM的**比特级电擦除**特性;

**芯片的特点:**

※掉电时信息不丢失, 功耗低、存储密度高;

※**芯片级+比特级**数据擦除方式, 读写速度很快;

※可在计算机内实现擦写, **不需专用擦写器**;

**典型应用:** U盘、SSD固态硬盘等。

## 3.2.3 主存储器的设计与扩展

### 主存的设计目的

- 利用半导体存储芯片和其它逻辑芯片，构成所需要的存储器。
- 由存储芯片(  $mk \times n$  位/片)构成存储器( $Mk \times N$  位)
- 其它逻辑芯片：如地址寄存器、地址译码器、刷新控制电路等

### 主存储器的组织

- 存储器逻辑设计、如何与CPU连接、动态存储器的刷新、存储器的校验等。

## 3.2.3 主存储器的设计与扩展

### 1. 主存储器容量的扩展

#### 位扩展

- 存储芯片( $m \times k \times n$ 位/片)构成存储器( $m \times k \times N$ 位)

#### 字扩展

- 存储芯片( $m \times k \times n$ 位/片)构成存储器( $M \times k \times n$ 位)

#### 字位同时扩展

- 存储芯片( $m \times k \times n$ 位)构成存储器( $M \times k \times N$ 位)

注意:  $m < M, n < N$



## 3.2.3 主存储器的设计与扩展

### 1. 主存储器容量的扩展

#### (1) 位扩展

存储芯片( $m \times n$ 位/片)构成存储器( $m \times N$ 位)

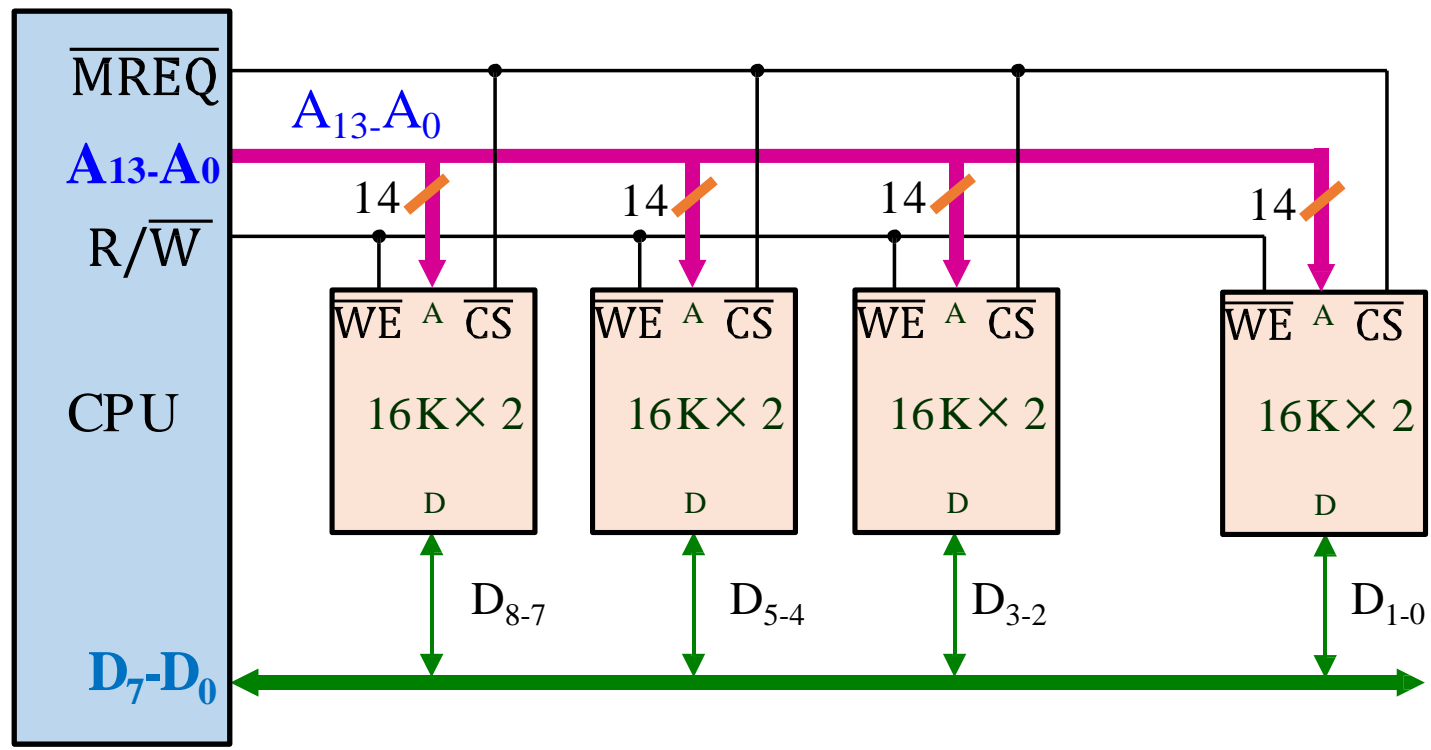
**特点：**字数不变(单元数不变)，位数扩展(字长加长)

- **芯片地址码**位数与**存储器的地址码**位数相同
- 需存储芯片数： $N/n$ 片
- 每个存储单元的位数增加，给出地址后，该存储单元中所含芯片均工作
- 芯片间各端点(引脚)如何连接？
  - \* **地址端、/CS、/WE、/OE端(若有)：**分别并接
  - \* **数据输入、输出端：**各位单独引出



### 3.2.3 主存储器的设计与扩展

[例1] 用  $16K \times 2$  的芯片组成  $16K \times 8$  的存储器。



## 3.2.3 主存储器的设计与扩展

### 1. 主存储器容量的扩展

#### (2) 字扩展

存储芯片( $m \times k \times n$ 位/片)构成存储器( $M \times k \times n$ 位)

**特点：**位数不变(字长不变)、扩充容量(存储单元数增加)；

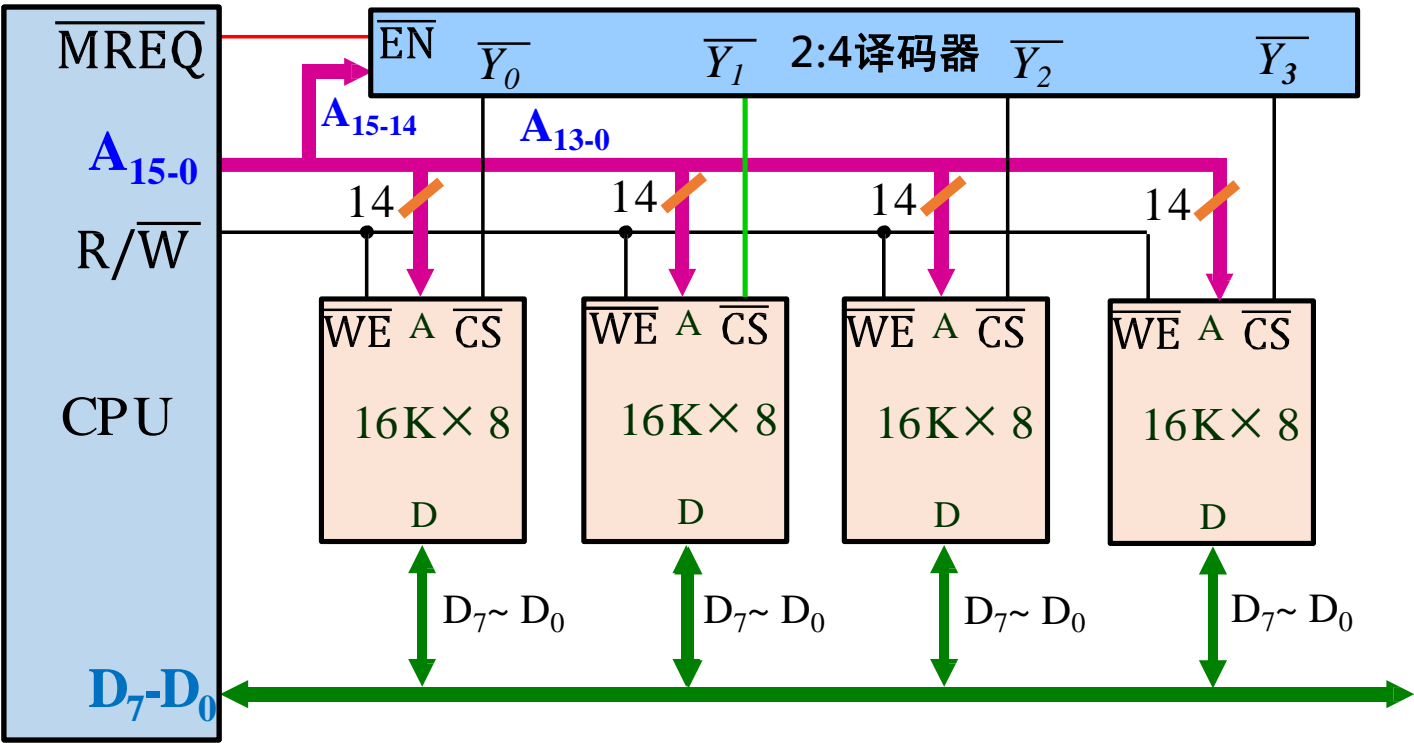
- 芯片和存储器二者的**存储单元的位数相同**
- **芯片地址码位数小于存储器的地址码位数**
- 需存储芯片数： $M/m$
- 给出地址后，选中芯片工作
- 芯片间各端点如何连接
  - \* **A、/WE、/OE(若有)、数据输入输出端：分别并接**
  - \* **/CS：单独引出，与增加的高位地址码(存储器地址码位数减芯片的地址码位数)的译码结果连接**





### 3.2.3 主存储器的设计与扩展

[例2] 用 $16\text{K} \times 8$  的芯片组成 $64\text{K} \times 8$  的存储器。



## 3.2.3 主存储器的设计与扩展

### 1. 主存储器容量的扩展

#### (3) 字、位同时扩展

存储芯片( $m \times n$ 位/片)构成存储器( $M \times N$ 位)

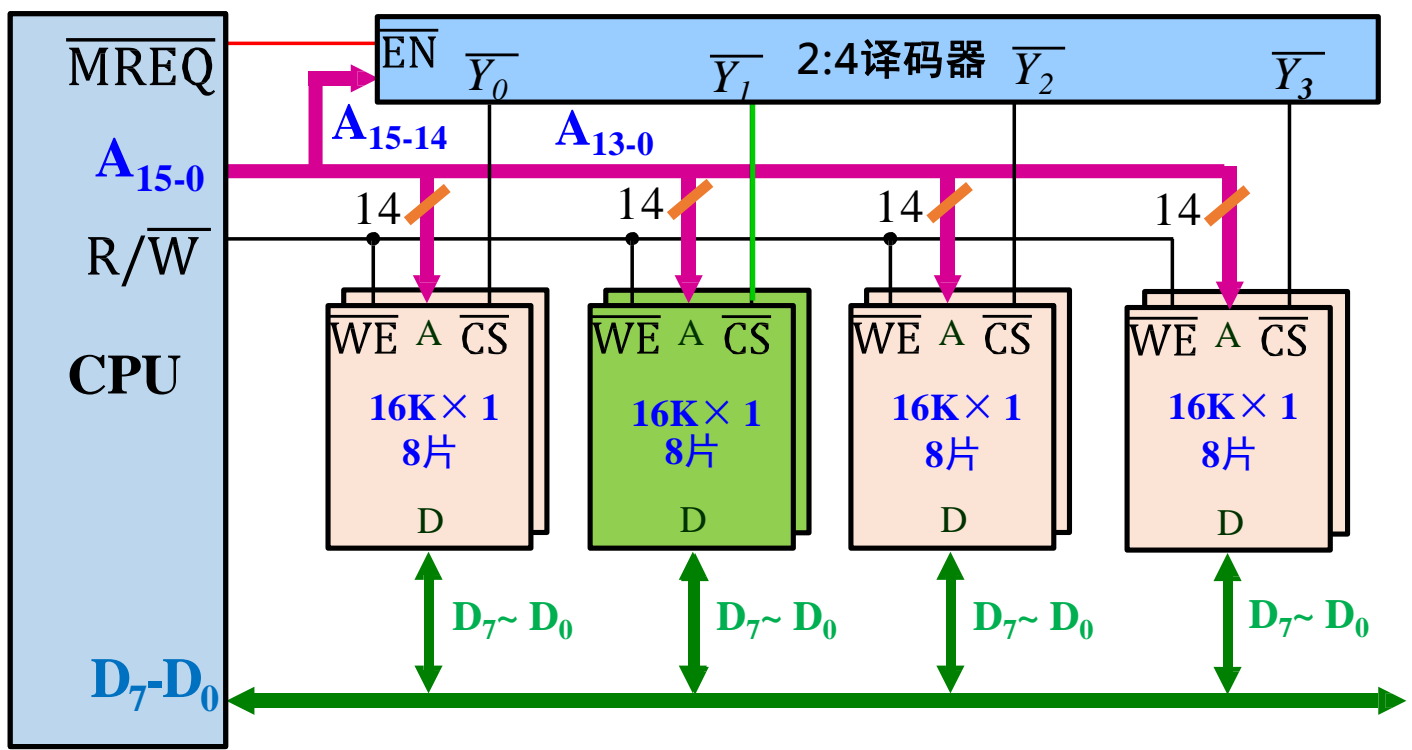
**特点：**存储单元数，字长同时增加

- 存储器地址码位数**多于**芯片地址码位数，存储器存储单元位数**大于**芯片存储单元位数。
- 给出地址后，同组芯片均工作
- 需存储芯片数： $[M/m][N/n]$ 片
- 芯片间各端点的连接
  - $A$ 、 $/WE$ 、 $/OE$ (若有)：分别并接
  - $/CS$ ：位向(同组)并接，字向(不同组)独立引出



### 3.2.3 主存储器的设计与扩展

[例3]用16K×1的存储芯片组成 64K×8的存储器。





## 3.2.3 主存储器的设计与扩展

### 2. 访存地址的译码方式

地址码：片内地址和选片地址两部分。

片内地址：低端地址码构成，长度取决于所选存储芯片的字数。

选片地址：高端地址码为选片地址，经译码产生选片信号。

#### (1) 全译码方式

选片地址部分全部参加译码。

- 实际使用的存储空间与CPU可访问的最大存储空间相同
- 对实际空间的地址范围有严格要求

### 3.2.3 主存储器的设计与扩展

①实际使用存储空间与CPU可访问最大存储空间相同。

**[例1]** CPU的访存地址16位 ( $A_{15} \sim A_0$ )，可访问最大存储空间64KB，选用16K $\times$ 4芯片构成。

分析：共需8片，构成4个小组，片内地址14位，选片地址2位。2位选片地址必须全部参加译码才能产生4个片选信号。4个小组的地址范围为：

$\left\{ \begin{array}{l} 0000H \sim 3FFFH \\ 4000H \sim 7FFFH \\ 8000H \sim BFFFH \\ C000H \sim FFFFH \end{array} \right.$



### 3.2.3 主存储器的设计与扩展

②实际使用存储空间小于CPU可访问最大存储空间，对实际空间地址范围有严格的要求。

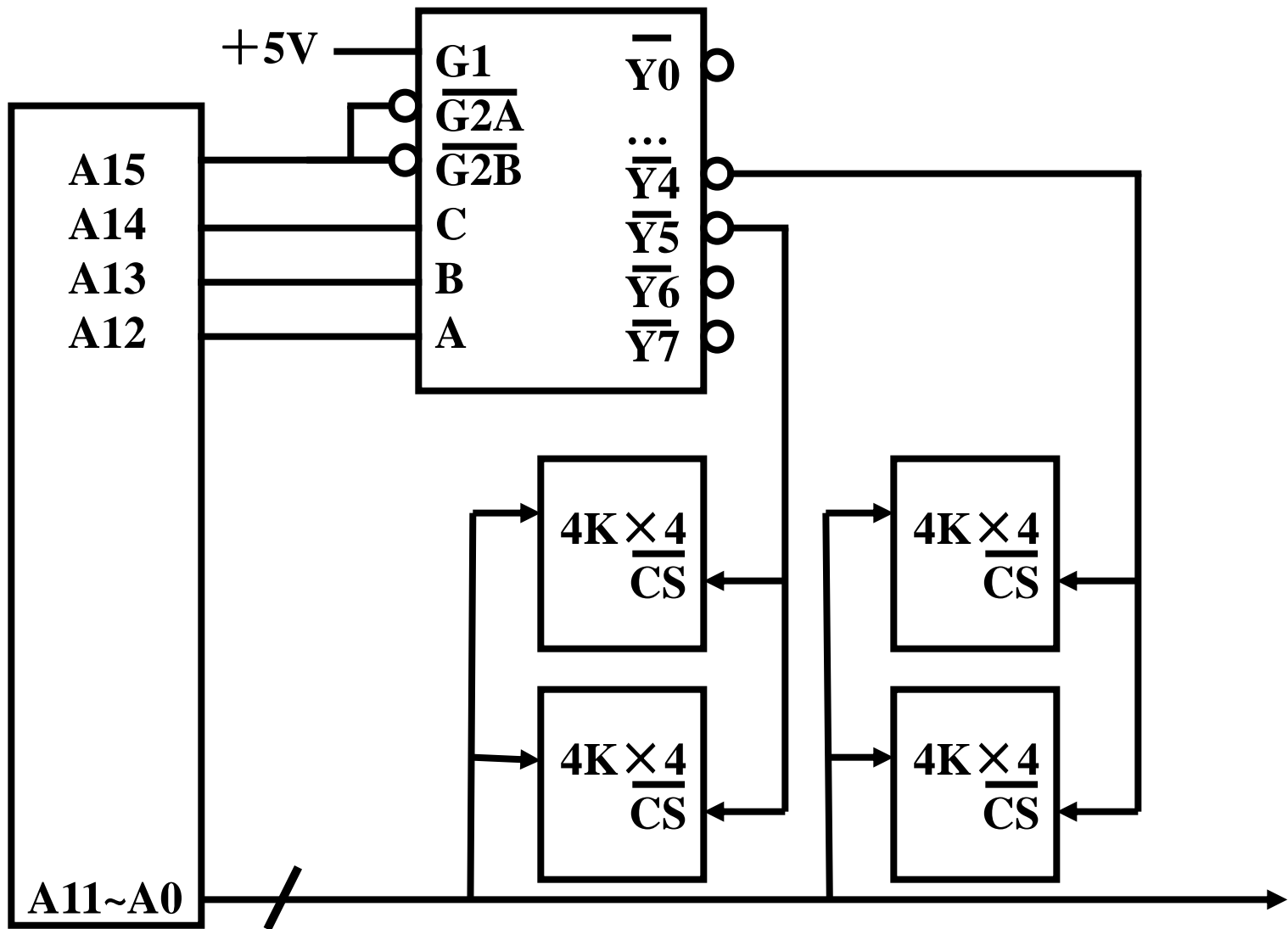
[例2]CPU的访存地址长16位（A15~A0），可访问的最大存储空间为64KB，而系统中实际使用的存储空间只有8KB，选用容量4K×4的芯片4片，并要求其地址范围必须在4000H~5FFFH范围内。

A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	地址范围
0	<u>1</u>	<u>0</u>	<u>0</u>	{	0	0	0	0	0	0	0	0	0	0	0	4000H
0	1	0	0		1	1	1	1	1	1	1	1	1	1	1	4FFFH
0	<u>1</u>	<u>0</u>	<u>1</u>	{	0	0	0	0	0	0	0	0	0	0	0	5000H
0	1	0	1		1	1	1	1	1	1	1	1	1	1	1	5FFFH

A    B    C

### 3.2.3 主存储器的设计与扩展

地址译码方式





## 3.2.3 主存储器的设计与扩展

### (2) 部分译码方式

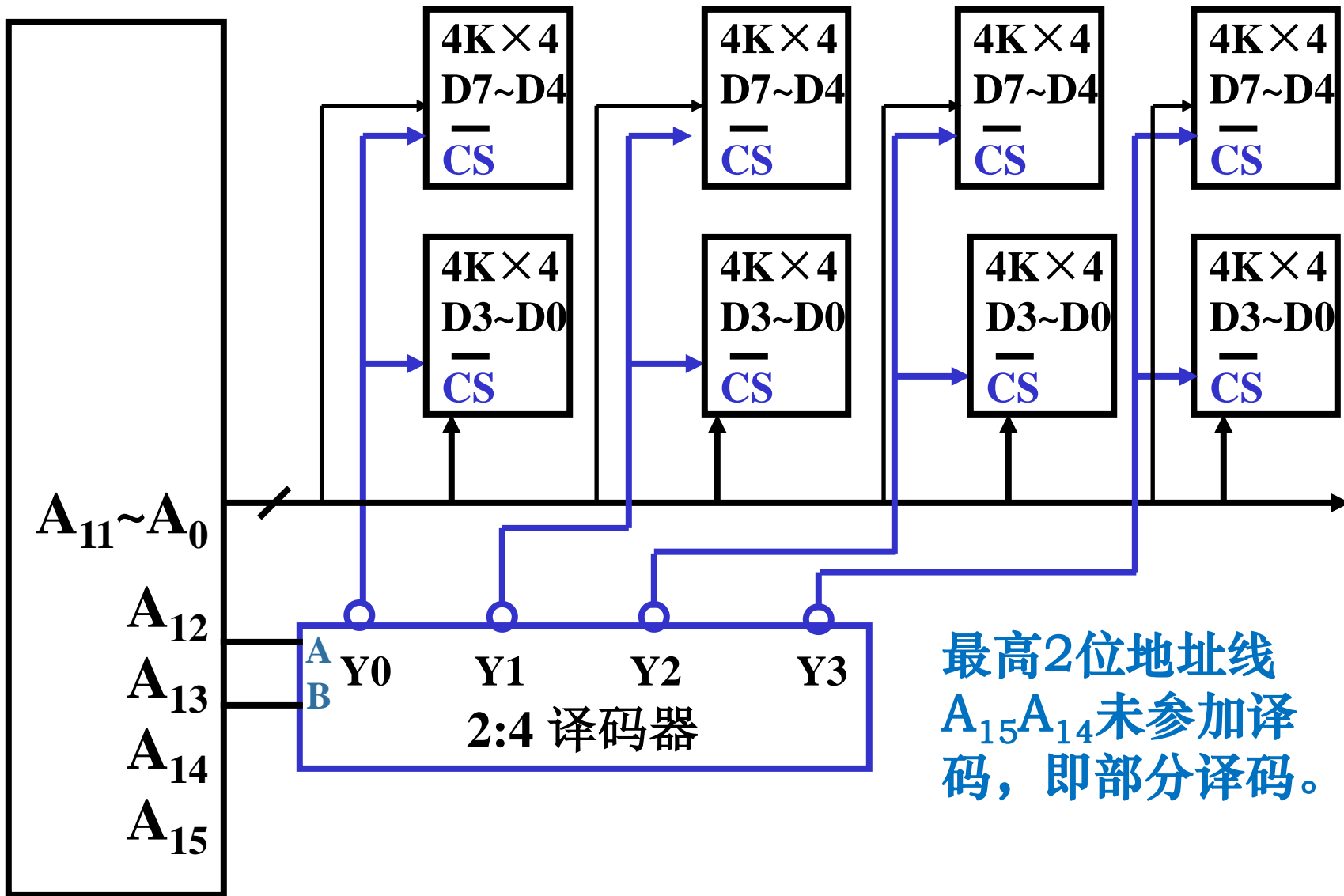
实际使用的存储空间小于CPU可访问的最大存储空间，而且对实际空间的地址范围**没有严格要求**。

**[例3]** CPU的访存地址长16位（A15~A0），可访问的最大存储空间为64KB，而系统中实际使用的存储空间只有16KB，选用容量4K×4的芯片扩展。



### 3.2.3 主存储器的设计与扩展

地址译码方式





# 3.2.3 主存储器的设计与扩展

## 地址分配

片选地址				片内地址												地址范围
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	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000H~ 0FFFH
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1000H~ 1FFFH
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	2000H~ 2FFFH
0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	
0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	3000H~ 3FFFH
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	

地址重叠



## 3.2.3 主存储器的设计与扩展

### 3. 主存储器设计举例

**[例1]**现有 $8K \times 4$ 的RAM和 $8K \times 8$ 的ROM。CPU输出地址线A0~A15，数据线D0~D7，欲构成24KRAM，16KROM存储器。地址范围：RAM 2000~7FFFH，ROM C000~FFFFH。

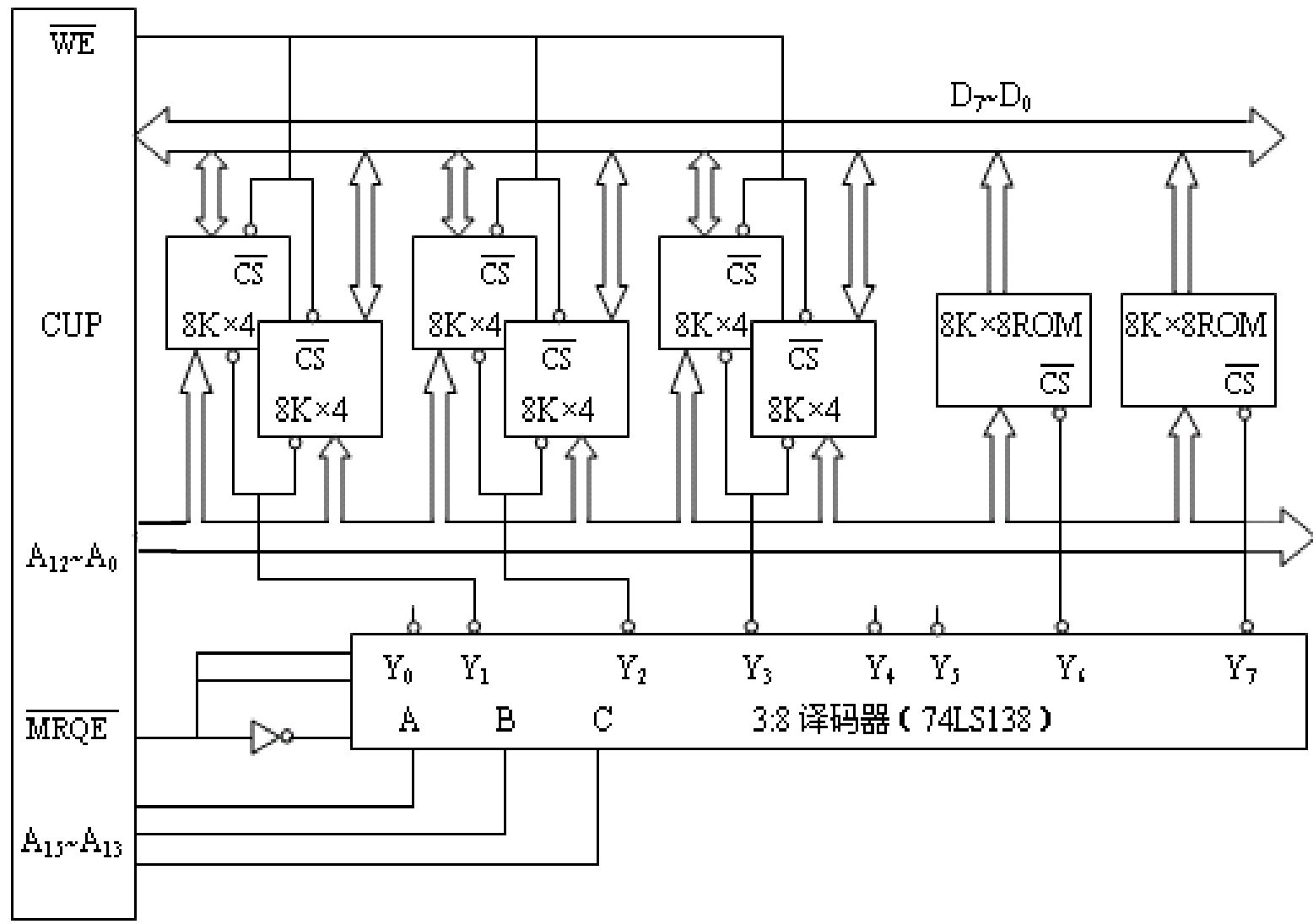
- RAM用 $8K \times 4$ 的芯片组成， $8K \times 4$ 芯片6片。
- ROM末地址 - 首地址 = FFFFH - C000H = 3FFFH;
- ROM容量： $2^{14} = 16K$ ，ROM用 $8K \times 8$ 芯片组成， $8K \times 8$ 的芯片2片。
- ROM的地址范围展开为1100 0000 0000 0000 ~
- 1111 1111 1111 1111，高位地址A15A14A13为110~111。

已给定地址空间，有严格的要求，属于全译码方式。地址无重叠区。



### 3.2.3 主存储器的设计与扩展

存储器的组成结构图及与CPU的连接图



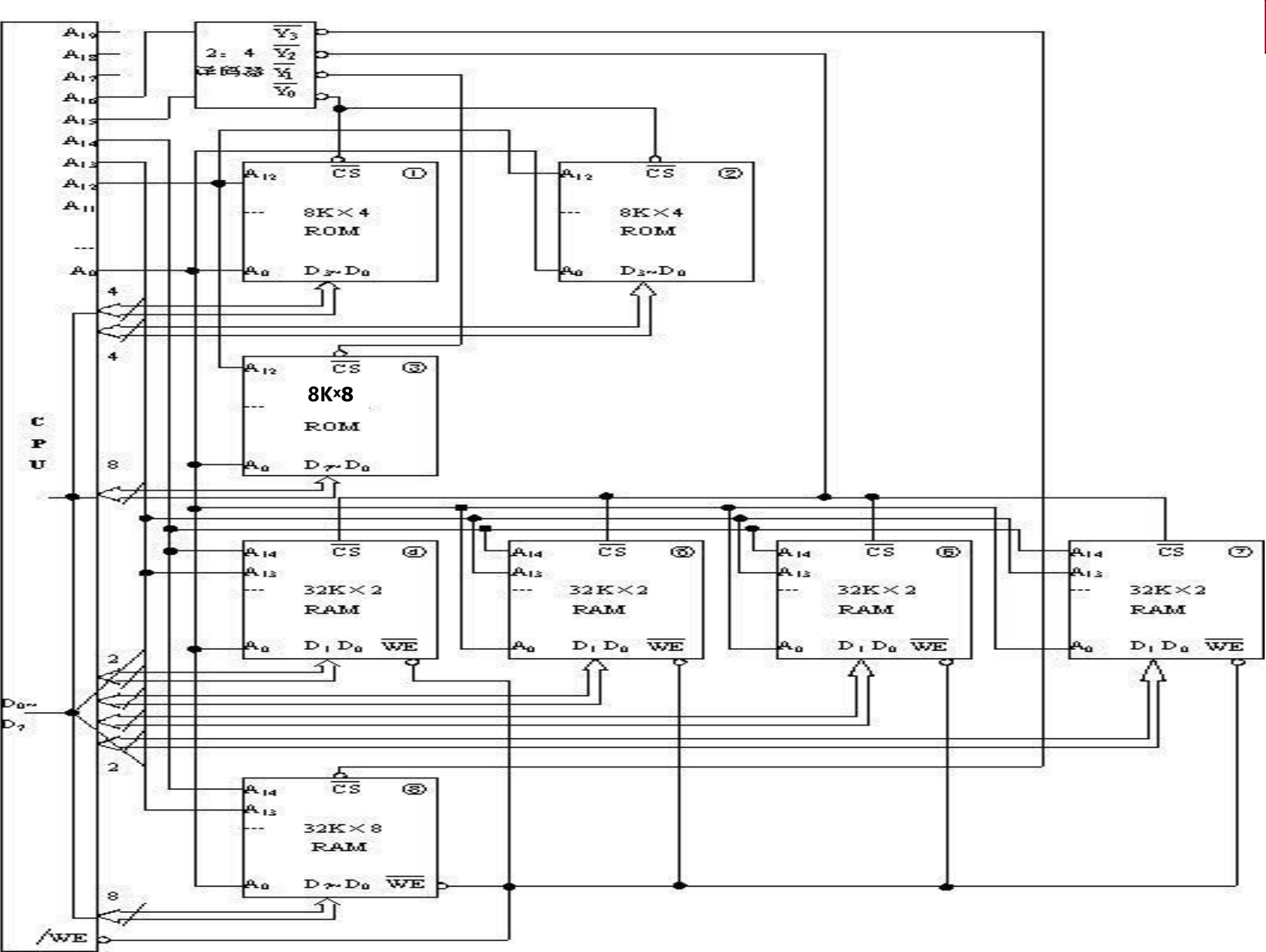


## 3.2.3 主存储器的设计与扩展

**[例2]** 某计算机系统中，CPU可输出20条地址线

(A19~A0)，8条数据线(D7~D0)和一条读/写控制线( $\overline{WE}$ )，主存储器按字节编址，由16KB的ROM和64KB的RAM组成。拟采用8K×4(位)的ROM芯片2片，8K×8(位)的ROM芯片1片，32K×2(位)的RAM芯片4片，32K×8(位)的RAM芯片1片。

主存储器可采用**部分译码**方式。





# 3.2.3 主存储器的设计与扩展

主存储器地址范围

所选 芯片	选片地址		片内地址	译码 输出	地址 范围	重叠区 个数	RAM ROM 片数
	A <sub>19</sub> A <sub>18</sub> A <sub>17</sub> A <sub>16</sub> 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>0</sub>				
①②	X X X 0 0	X X	0 0 0 ... 0 1 1 1 ... 1	=0	00000 ~ 01FFFH	32	8K×4的 ROM2个
③	X X X 0 1	X X	0 0 0 ... 0 1 1 1 ... 1	=0	08000 ~ 09FFFH	32	8K×8的 ROM1个
④⑤ ⑥⑦	X X X 1 0 X X X 1 0	0 0 1 1	0 0 0 ... 0 1 1 1 ... 1	=0	10000 ~ 17FFFH	8	32K×2的 RAM4个
⑧	X X X 1 1 X X X 1 1	0 0 1 1	0 0 0 ... 0 1 1 1 ... 1	=0	18000 ~ 1FFFFH	8	32K×8的 RAM1个

分析地址重叠？



# 思考

## 主存储器地址范围

所选芯片	RAM、ROM片数	地址范围
①②	8K×4的ROM2个	18000~19FFFH
③	8K×8的ROM1个	F8000~F9FFFH
④⑤⑥⑦	32K×2的RAM4个	98000~9FFFFH
⑧	32K×8的RAM1个	B8000~BFFFFH





## 3.2.3 主存储器的设计与扩展

**[例3]** 设CPU有16根地址线、8根数据线，并用MREQ作为访存控制信号（低电平有效），用WR作为读/写控制信号（高电平为读，低电平为写）。现有下列存储芯片：1K×4位RAM、4K×8位RAM、8K×8位RAM、2K×8位ROM、4K×8位ROM、8K×8位ROM及74138译码器和各种门电路。画出CPU与存储器的连接图，要求如下：

- 1) 主存地址空间分配为：  
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

### 3.2.3 主存储器的设计与扩展

**[例4]** 设CPU有16根地址线、8根数据线，并用MREQ作为访存控制信号（低电平有效），用WR作为读/写控制信号（高电平为读，低电平为写）。现 有下列存储芯片：

1K×4位RAM、4K×8位RAM、8K×8位RAM、2K×8位ROM、4K×8位ROM、8K×8位ROM及74138译码器和各种门电路。画出CPU与存储器的连接图。

1) 要求主存的地址空间：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，最大的4K地址空间为系统程序工作区。

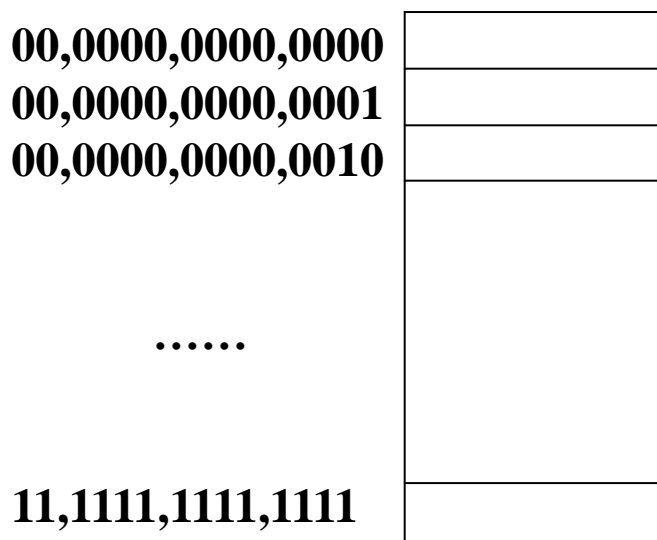
2) 详细画出存储芯片的片选逻辑并指出存储芯片的种类及片数。



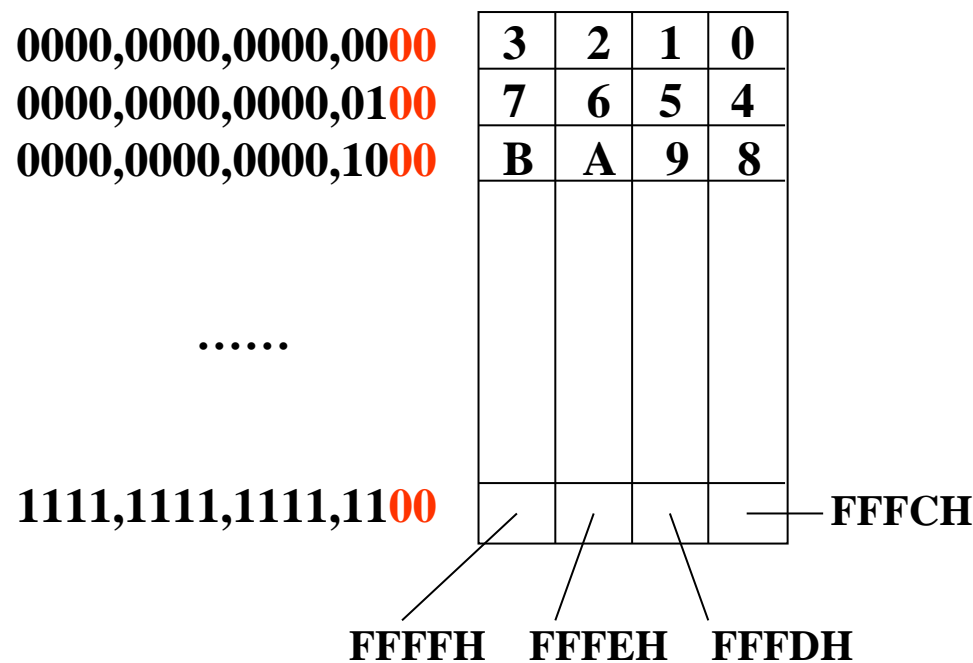
#### 4. 主存储器的编址方式

- 按字编址
- 按字节编址

**按字节编址，访存地址码长度16（位）。**



### (a) 按字编码

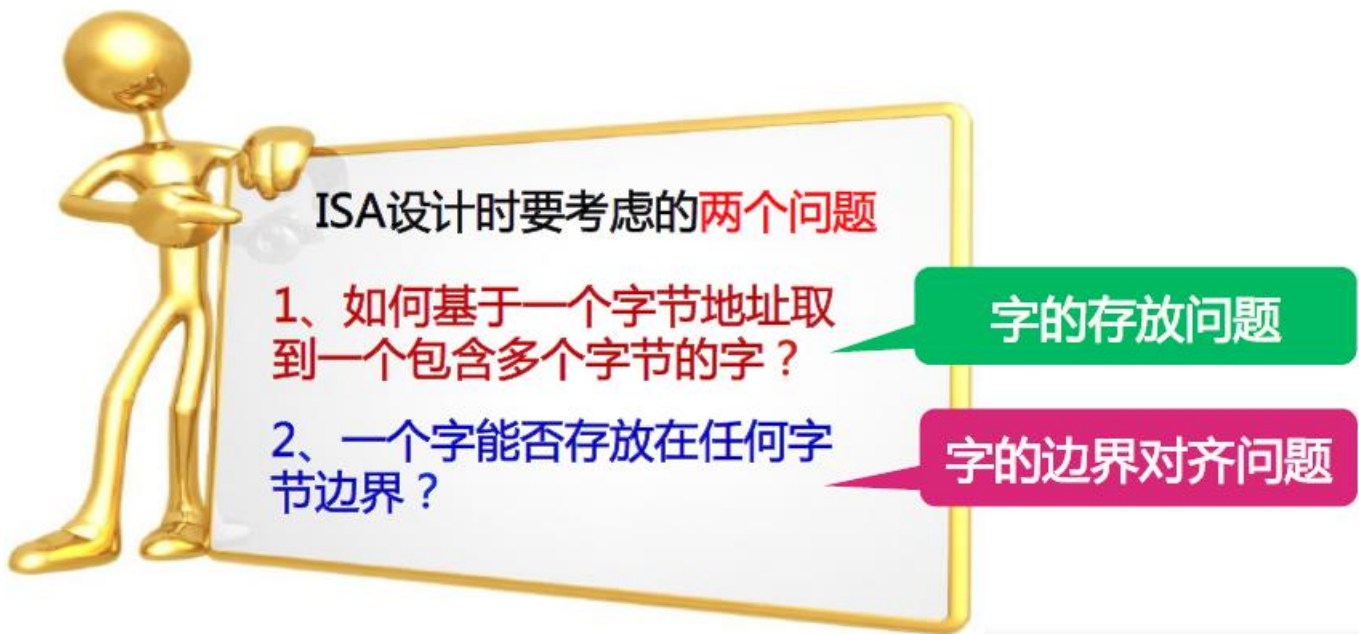


### (b) 按字节编码

## 3.2.4 主存中的数据组织

### 1. 存储字长

- 主存的一个存储单元所包含的二进制位数；
- 目前大多数计算机的主存按字节编址。



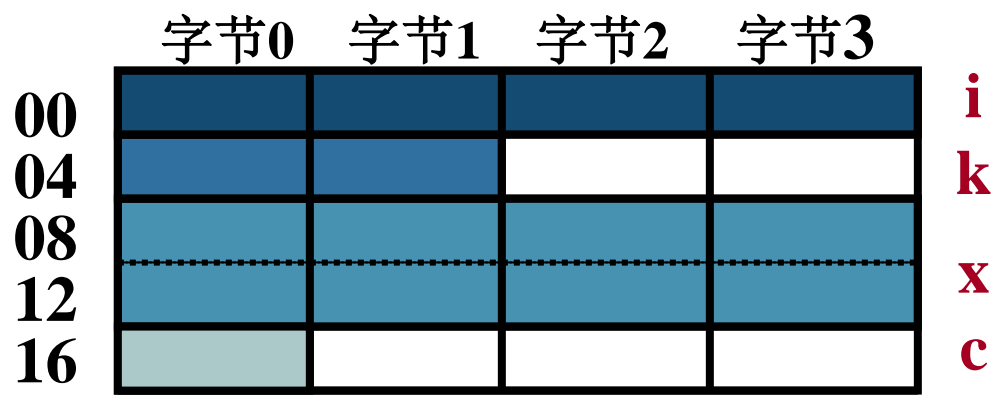


# 3.2.4 主存中的数据组织

## 2. 数据存储与边界的关系

### (1) 按边界对齐的数据存储

int i, short k, double x, char c （32位系统中）

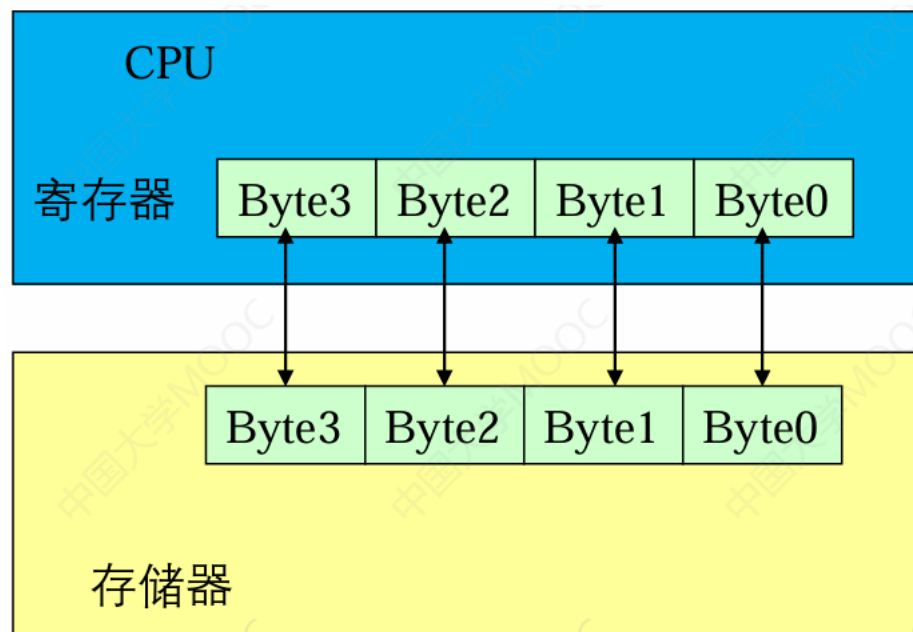


按字边界对齐    &i=0; &k=4; &x=8; &c=16

## 3.2.4 主存中的数据组织

### 对齐方式下的存储器访问

- 速度较高
- 接口较简单



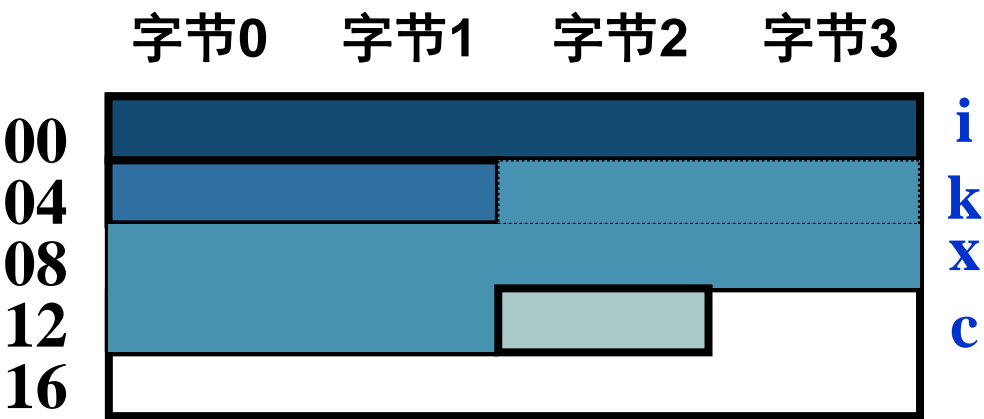


## 3.2.4 主存中的数据组织

### 2. 数据存储与边界的关系

#### (2) 不按边界对齐的数据存储

int i, short k, double x, char c (32位系统中)



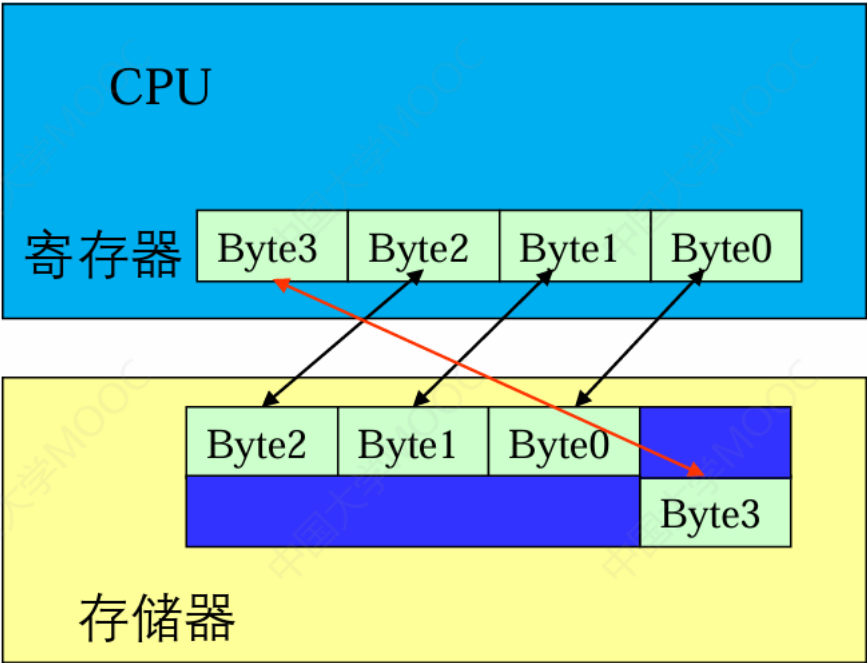
不按边界对齐    虽节省了空间，但增加了访存次数！  
需要在性能与容量间权衡！



### 3.2.4 主存中的数据组织

#### 不对齐方式下的存储器访问

- 节省空间
- 访存速度慢
- 接口复杂



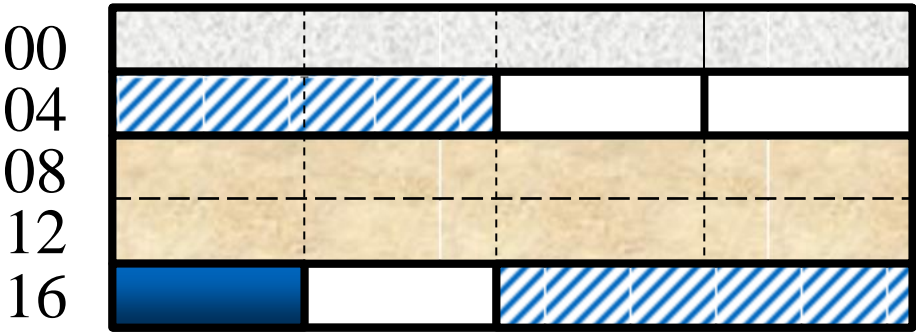




## 3.2.4 主存中的数据组织

### 2. 数据存储与边界的关系

#### (3) 边界对齐与存储地址的关系（以32位为例）



- ❑ 双字长数据边界对齐的起始地址的最末三位为000(8字节整数倍);
- ❑ 单字长边界对齐的起始地址的末二位为00(4字节整数倍);
- ❑ 半字长边界对齐的起始地址的最末一位为0(2字节整数倍)。

**数据的地址是相应的边界地址**



## 3.2.4 主存中的数据组织

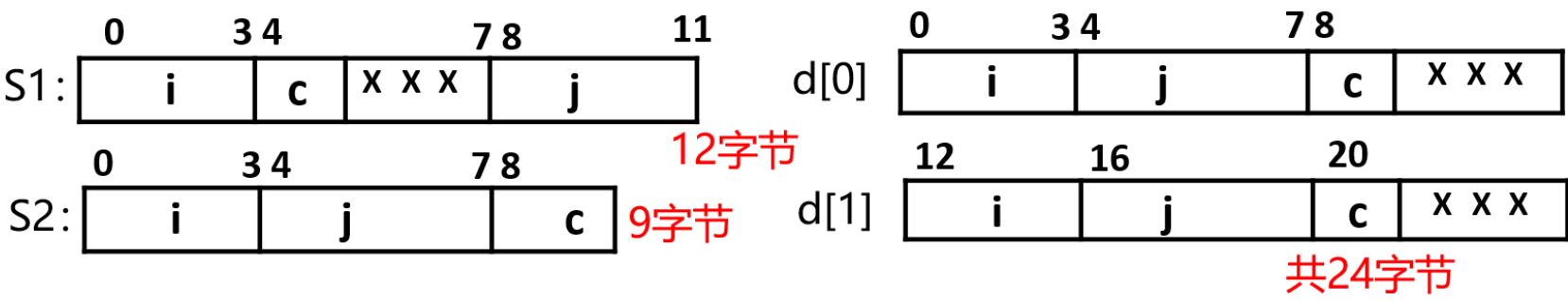
### 2. 数据存储与边界的关系

考虑下列两个结构声明（以32位为例）：

```
struct S1 {  
    int    i;  
    char   c;  
    int    j; };
```

```
struct S2 {  
    int    i;  
    int    j;  
    char   c; };
```

在对齐情况下，哪种结构声明更好？ struct S2 d[2]满足对齐要求的字节数？

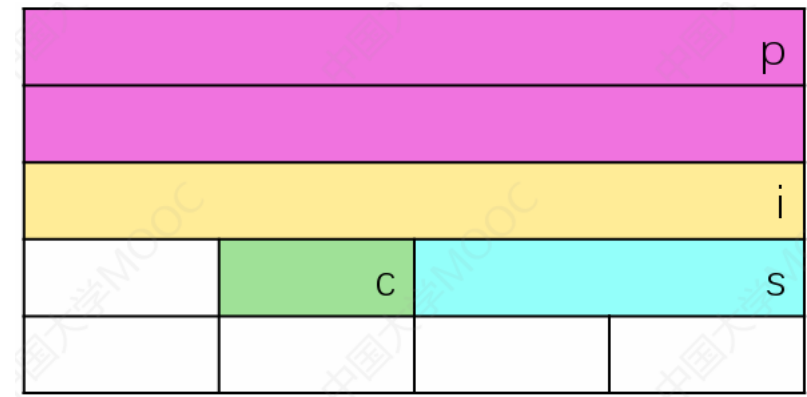
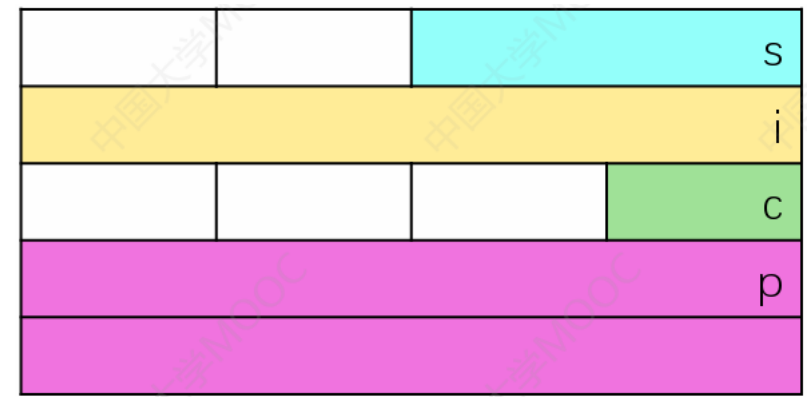




# 思考

试改变以下数据结构定义以减少对齐的开销

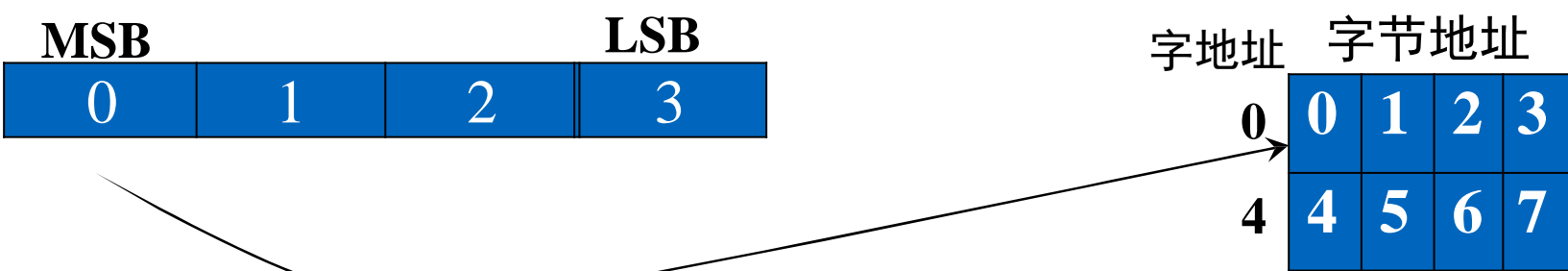
```
Struct Loose {  
    short s;    //16bit  
    int i;      //32bit  
    char c;     //8bit  
    double p;   //64bit  
};
```



# 3.2.4 主存中的数据组织

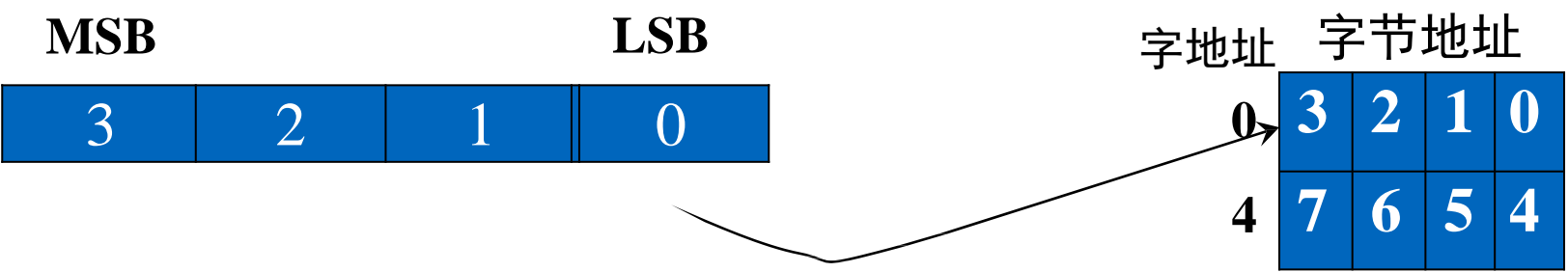
## 3. 大端与小端存储方式

Big-endian: 最高字节地址(MSB)是数据地址



IBM 360/370, Motorola 68k, MIPS, Sparc

Little-endian: 最低字节地址(LSB)是数据地址



有的机器两者都支持，但需要设定. Intel 80x86, DEC, VAX

## 3.2.4 主存中的数据组织

### 3. 大端与小端存储方式

设某程序执行前  $r0 = 0x\ 11223344$

执行下列指令：

$r1 = 0x100$

STR  $r0, [r1]$

LDRB  $r2, [r1]$

小端模式下:  $r2 = 0x44$

大端模式下:  $r2 = 0x11$

无论是大端还是小端，每个系统内部是一致的，但在系统间通信时可能会发生问题！因为顺序不同，需要进行顺序转换。

## 3.2.4 主存中的数据组织

### 3. 大端与小端存储方式

- ✗ 存放方式不同的机器之间程序移植或数据通信时，可能发  
生问题
  - 由于存放顺序不同，数据存储访问时，需要进行数据的顺序转换
  - 任何像音频、视频和图像等文件格式或处理程序都涉及字节的顺  
序问题
- 例： Little endian: GIF, PC Paintbrush, Microsoft RTF等
- Big endian: Adobe Photoshop, JPEG, MacPaint等

## 思考

某32位计算机，按**字节编址**，采用小端方式存放数据。  
假定有一个double型变量，则机器数表示为1122  
3344 5566 7788H，存放在0000 8040H开始的连续  
存储单元中，则存储单元0000 8046H中存放的是  
(**A**) ?

A. 22H    B. 33H    C. 77H    D. 66H



## 思考

在按字节编址，采用小端方式的32位计算机中，按边界对齐方式为以下C语言结构型变量a分配存储空间。

(short型占2个字节，int占4个字节)

```
Struct record {  
    short x1;  
    int   x2;  
} a;
```

若a的首地址为2020 FE00H，a的成员变量x2的机器数为1234 0000H，则其中34H所在存储单元的地址为 (D)。

A. 2020 FE03H

B. 2020 FE04H

C. 2020 FE05H

D. 2020 FE06H



## 3.2.5 高性能存储器

常用来改进存储系统性能的措施：

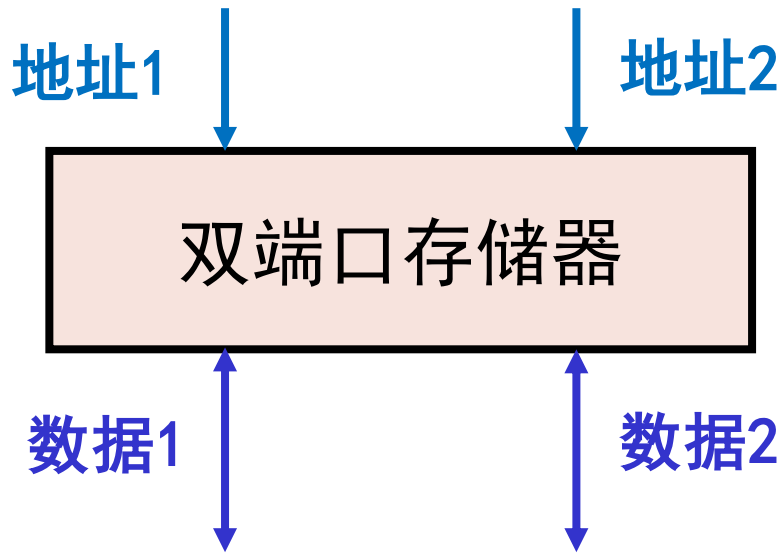
- ✓ 更高速主存或加长存储器字长
- ✓ 多端口存储器
- ✓ 采用多级Cache
- ✓ 采用交叉并行存储器

...

# 双端口存储器

[主存速度与CPU处理速度存在差距]

**双端口存储器：** 同一个存储器具有**两组独立的读写控制线路**，两个端口分别具有各自的地址线、数据线和控制线，可进行**独立的存取操作**。

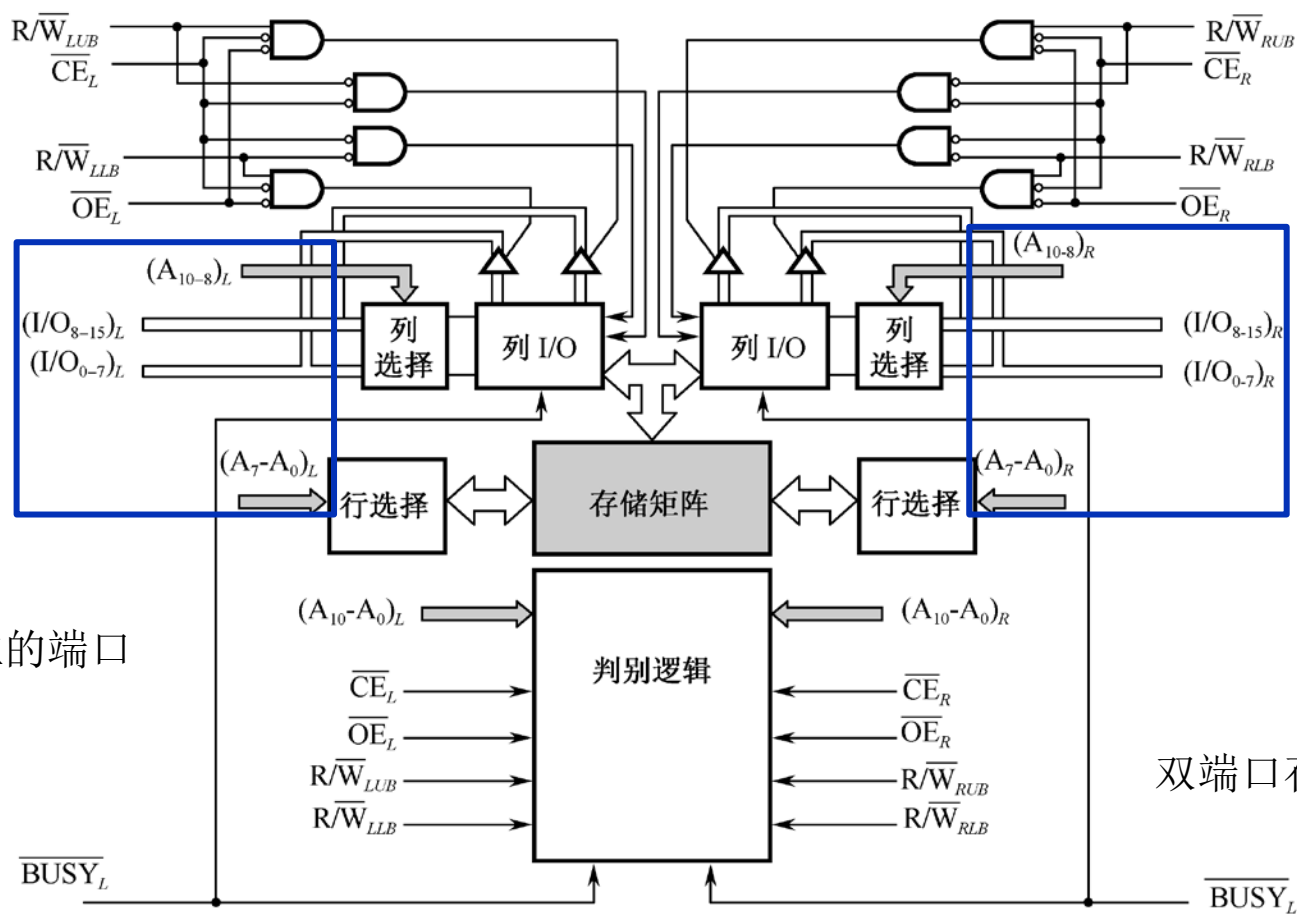


# 双端口存储器



存储容量**2K**  
字长**16**位

左右两套独立的端口

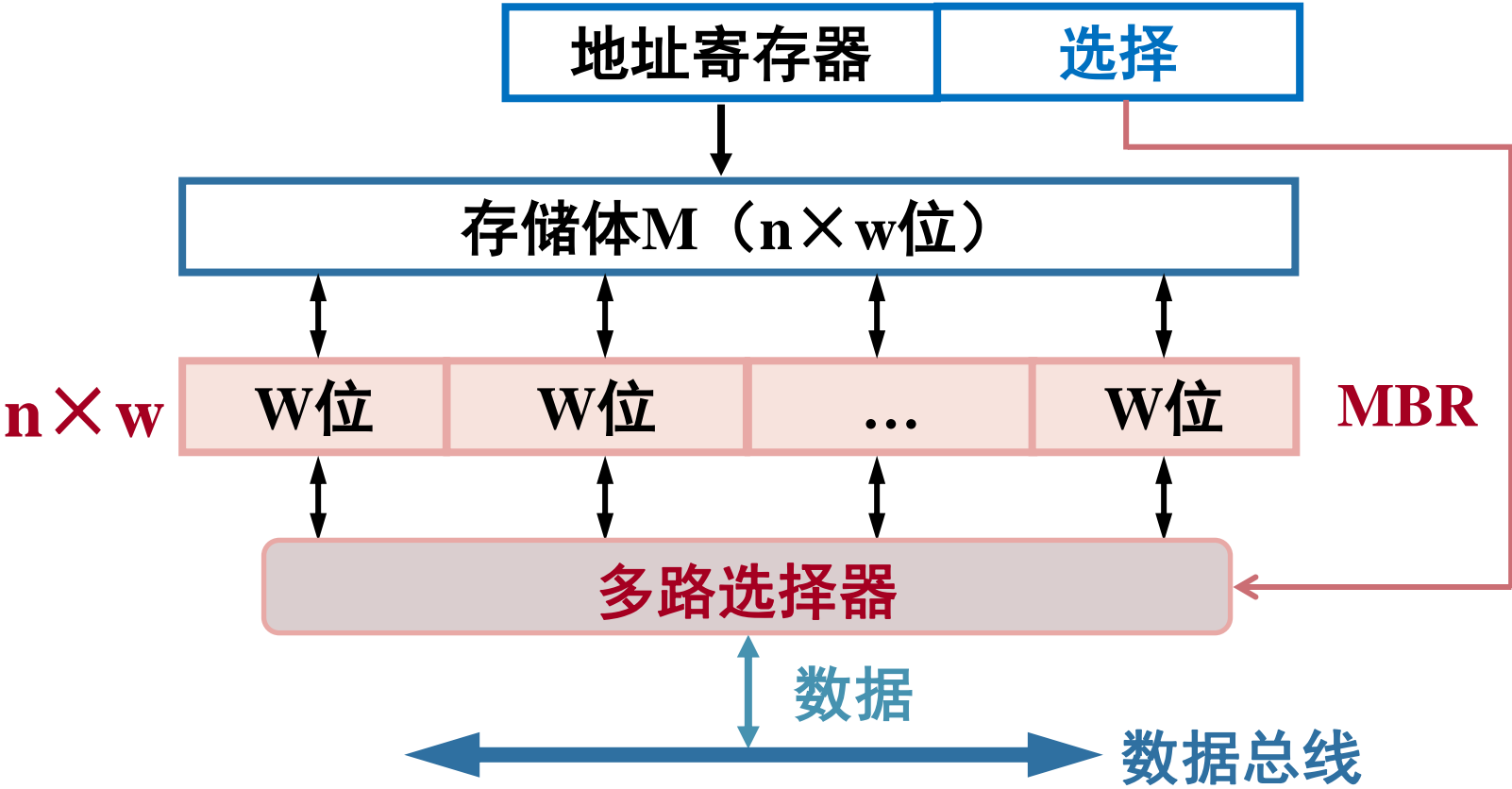


双端口存储器**IDT7133**

# 并行存储系统

## 1. 单体多字并行存储器

并行设置多个存储模块，在一个存取周期内，多个存储模块同时存取多个字以提高整体速度

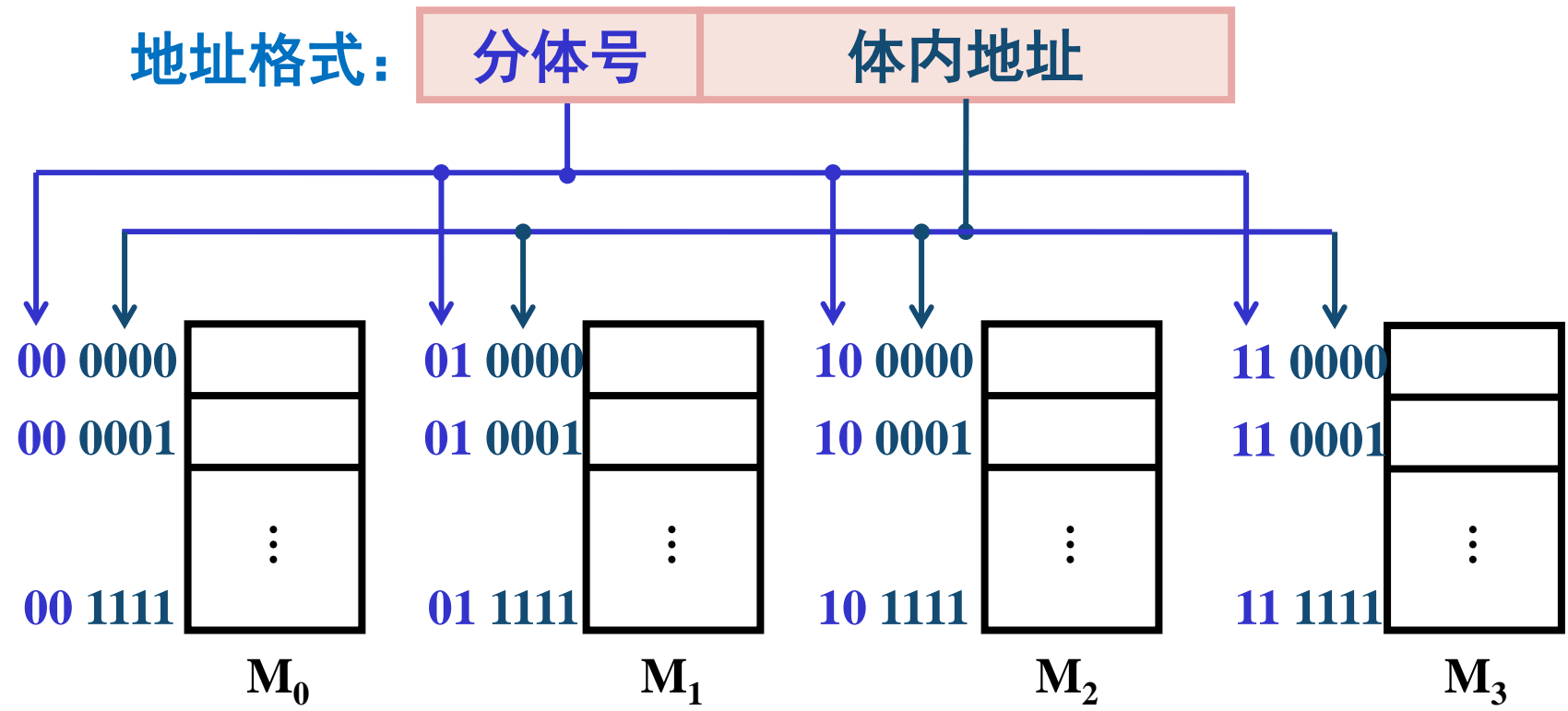


# 并行存储系统

## 2.多体交叉并行存储器

### (1)顺序编址方式

- ※有多个容量相同的独立存储模块；
- ※模块中的存储单元，依次分配连续的地址码；



## 并行存储系统

### 顺序编址方式的特点：

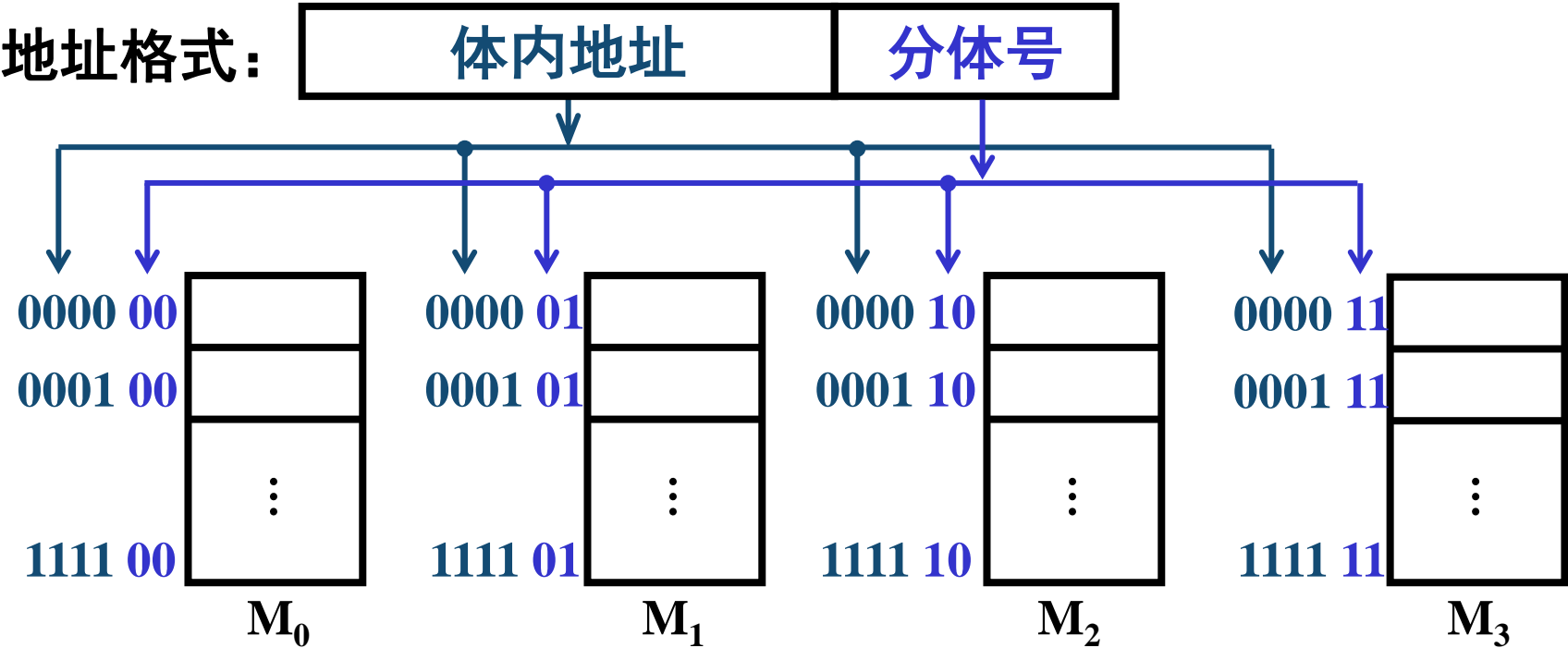
- ① 顺序访问地址连续的存储单元时，只能单模块工作（模块串行），其它模块处于空闲状态；
- ② 某个模块发生故障时，一般也不会影响其它模块正常工作，存储器容错性好；
- ③ 通过增加存储分体的数量，来扩展存储器容量比较方便，存储器扩展性好；
- ④ 模块之间是按串行方式工作的，会存储器的带宽受到限制；



# 并行存储系统

## (2) 交叉编址方式

- ※有多个容量相同的独立存储模块；
- ※连续地址被依次分配到不同模块中的存储单元；



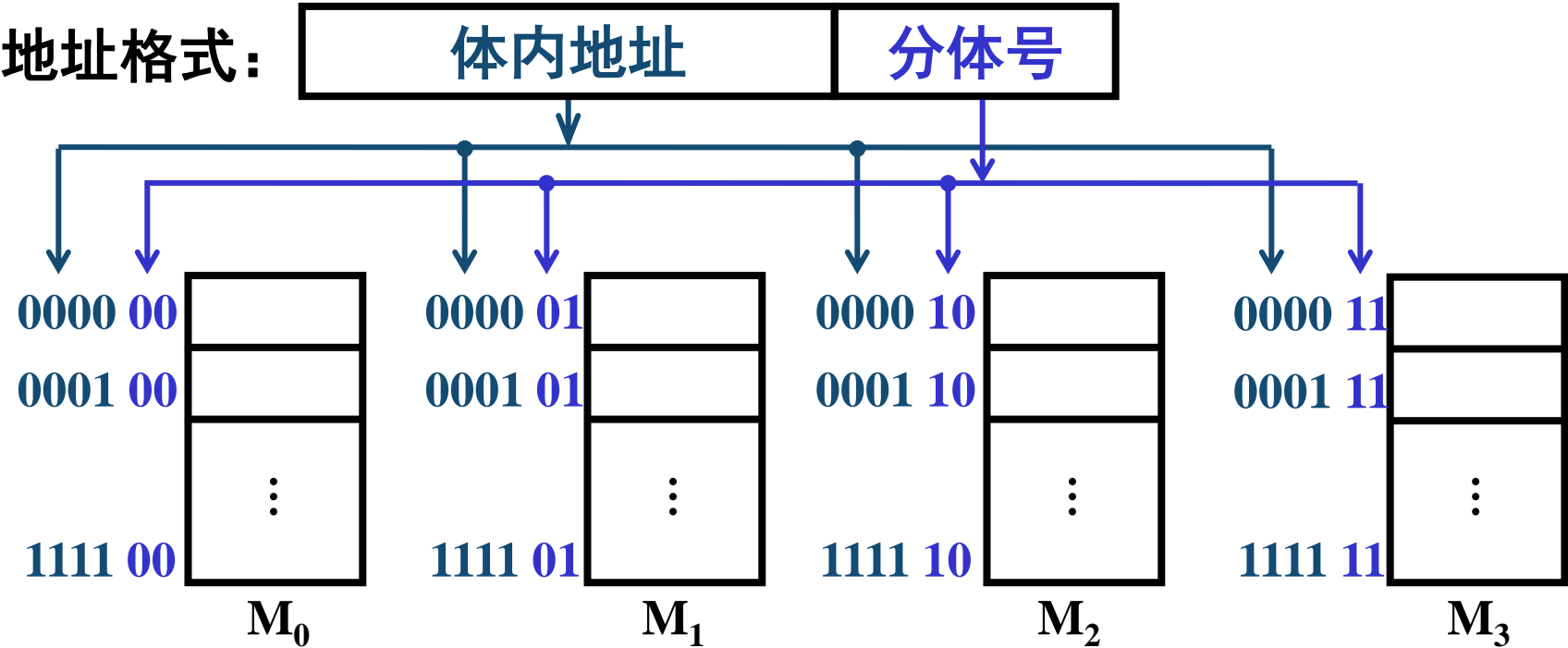
**[特点]** 存取周期不变时，流水式并行访存能提高带宽；



# 并行存储系统

## (2) 交叉编址方式

- ※有多个容量相同的独立存储模块；
- ※连续地址被依次分配到不同模块中的存储单元；



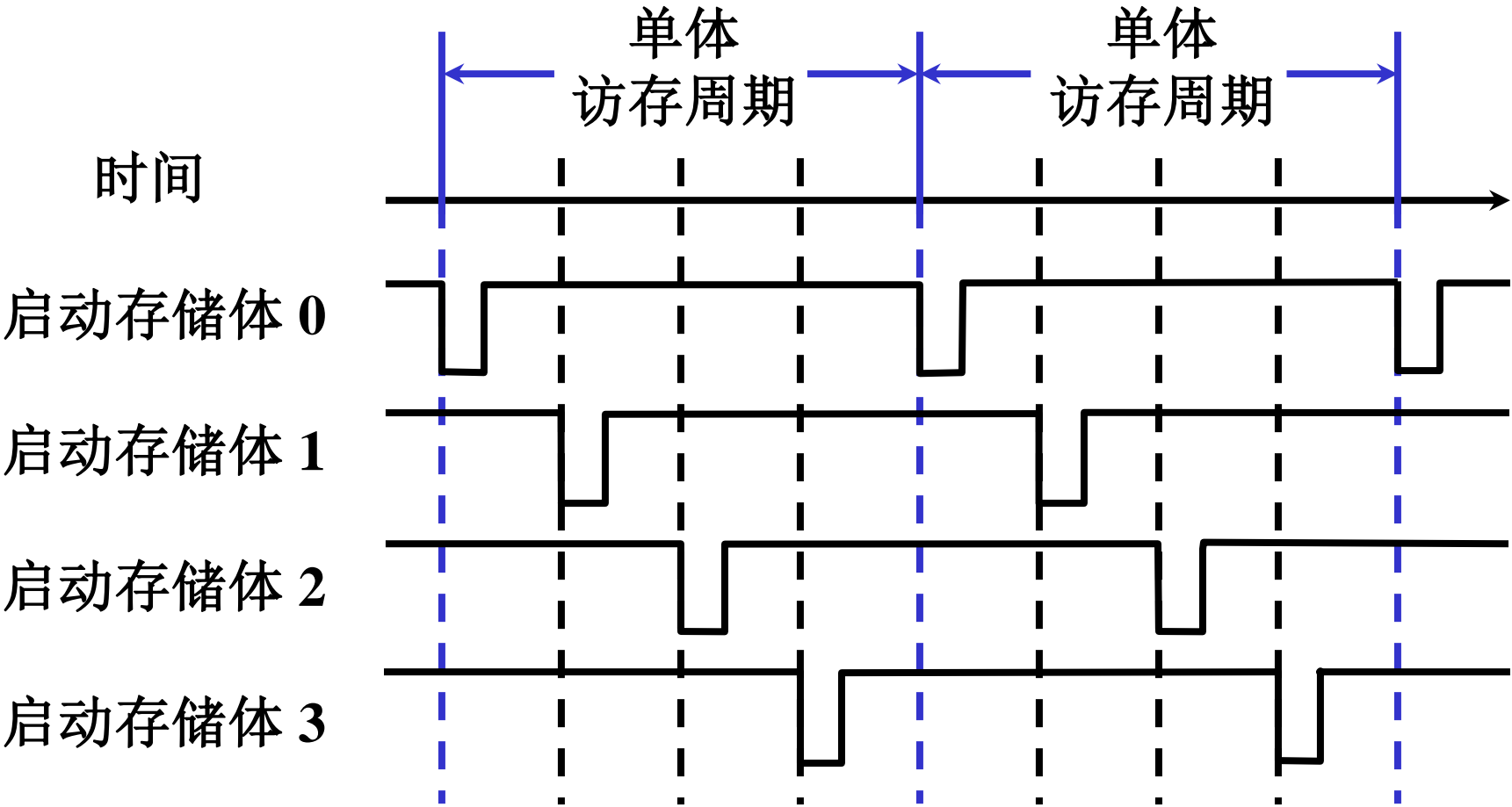
**[特点]** 存取周期不变时，流水式并行访存能提高带宽；





# 并行存储系统

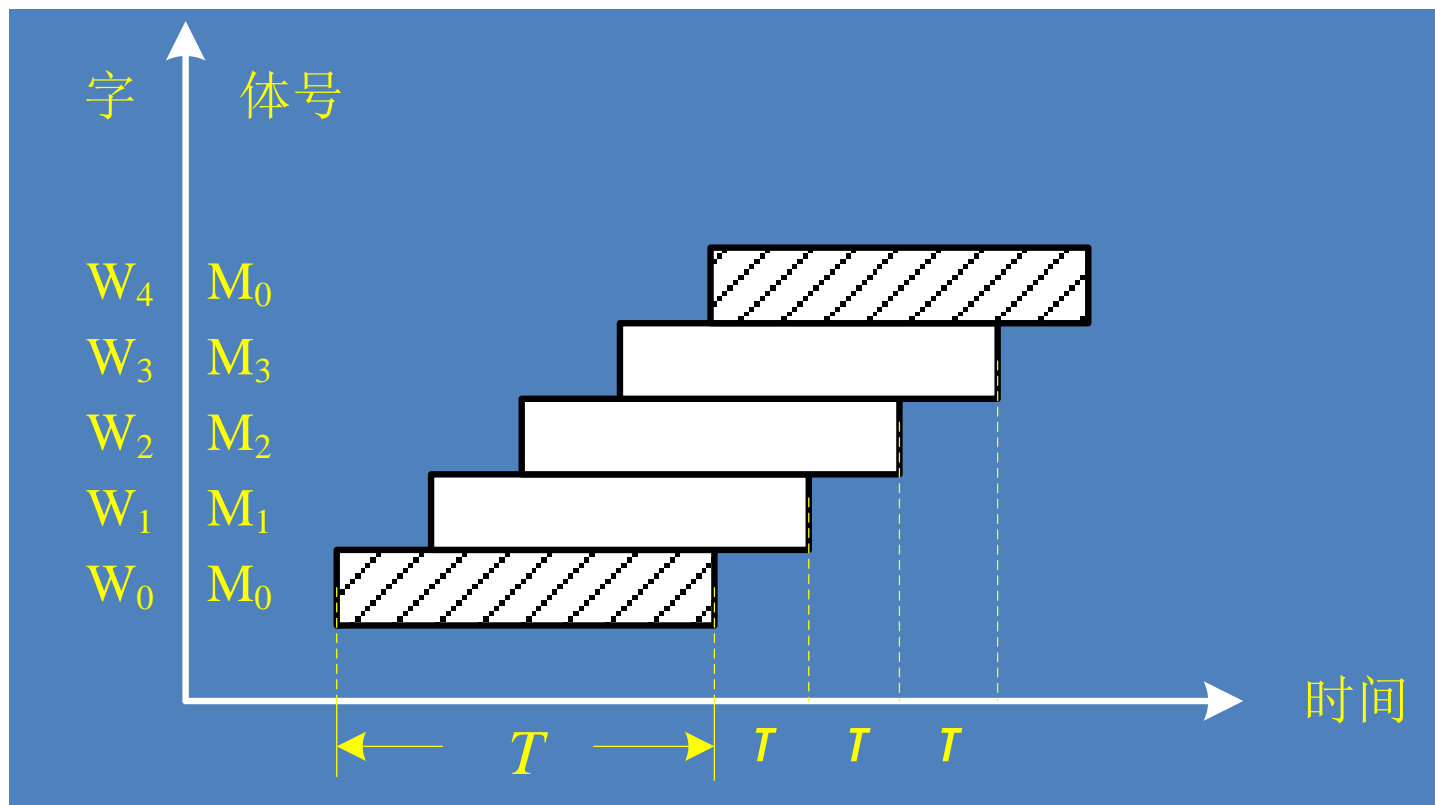
在不改变存取周期的前提下，增加存储器的带宽





# 并行存储系统

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  $T + (4 - 1)\tau$

## 并行存储系统

**[例1]** 设存储器容量为32字，字长64位，模块数 $m=4$ ，分别用顺序方式和交叉方式进行组织。存储周期 $T=200\text{ns}$ ，数据总线宽度为64位，总线传送周期 $\tau=50\text{ns}$ 。若连续读出4个字，问顺序存储器和交叉存储器的带宽各是多少？

解：顺序存储器和交叉存储器连续读出 $m=4$ 个字的信息总量都是：

$$q=64\text{b} \times 4=256\text{b}$$

顺序存储器和交叉存储器连续读出4个字所需的时间分别是：

$$t_1=T+(m-1)\tau=200\text{ns}+150\text{ns}=350\text{ns}=3.5 \times 10^{-7}\text{s}$$

$$t_2=mT=4 \times 200\text{ns}=800\text{ns}=8 \times 10^{-7}\text{s}$$

顺序存储器和交叉存储器的带宽分别是：

$$W_1=q/t_1=256\text{b} \div (3.5 \times 10^{-7})\text{s}=730\text{Mb/s}$$

$$W_2=q/t_2=256\text{b} \div (8 \times 10^{-7})\text{s}=320\text{Mb/s}$$



# RAM存储器介绍

## 1. 静态RAM

依靠存储单元中的交叉反馈电路形成互锁，通过电源对电路的持续供电以存储信息。

### ❖ SB SRAM

同步突发静态随机访问存储器

常作为Cache(高速缓存)

存储特点：能向CPU提供两种按突发地址访问存储器的方式。

(交替突发和线性突发)



## RAM存储器介绍

### ❖MP SRAM

多端口静态随机访问存储器

存储特点：提供多个读地址端口、写地址端口、数据I/O端口。

### ❖FIFO SRAM

先进先出的静态随机访问存储器

存储特点：支持以不同速率访问存储器，数据先进先出、后进后出。

(常用在接口电路，作为数据缓冲器)



## RAM存储器介绍

### 2.动态RAM

依靠存储单元中形成的栅级电容来保存信息，不需要持续电源，因此需要定期逐行刷新。

#### ❖EDO DRAM

扩展数据输出动态随机访问存储器

存储特点：行列地址译码后定位到存储单元，能整体锁定一列数据，保持该列在输出缓冲区中的数据开放,直到列改变或读周期切换。

可避免重复的同列寻址和数据读操作，速度快。

## RAM存储器介绍

### ❖ SDRAM

#### 同步动态随机访问存储器

存储特点：采用DRAM存储结构，但半导体工艺和工作机制有改进。

※同步时序，时钟上升沿触发。

※具备对芯片的行、列单元整体锁定功能。

※内部存储器采用了支持并行操作的分组结构,可交替与外部交互数据。

10多年前作为计算机主存,现在已基本弃用。

# RAM存储器介绍

## ❖DDR SDRAM

在SDRAM基础上，逐步发展起来的双倍数据率、同步动态、随机访问存储器。

存储特点：DRAM存储结构，同步时序，在时钟信号的上升和下降沿各触发 1 次存储读写操作。

※提供数据预读（bits）：



## ❖GDDR SDRAM

专用于显存的DDR存储器

“天河2号”的协处理配置GDDR5局存储器（8GB）