



第3章 存储系统2





3.4 只读存储器和闪速存储器

- 只读存储器ROM
- 闪速存储器





只读存储器ROM

■ ROM (Read Only Memory)

- ◆ 只能读出，不能写入
- ◆ 具有不易失性
- ◆ 只读存储器写入数据的过程，称为对其进行编程





只读存储器分类 (1)

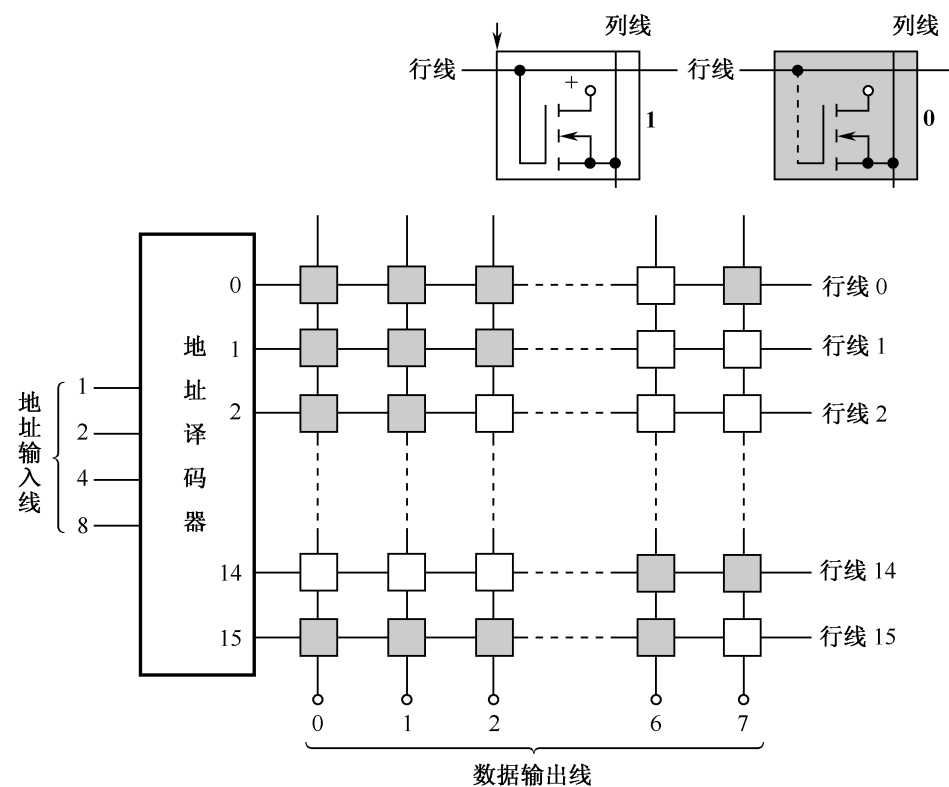
根据编程方法的不同来划分

- 掩模式只读存储器
- 一次编程只读存储器 (PROM)
- 多次编程只读存储器
 - ◆ 光擦编程只读存储器 (EPROM)
 - ◆ 电擦编程只读存储器 (EEPROM)
 - ◆ 电改写只读存储器 (EAROM)



掩模式只读存储器

- 在芯片制造过程中确定其内容
- 使用时，只能读出，不能再进行修改
- 优点：可靠性高、集成度高、价格便宜、适宜大批量生产
- 缺点：不能重写
- 只能专用，用户可向生产厂家定做





一次编程只读存储器 (PROM)

- 产品出厂时，所有存储元均制成“0”（或均为“1”）
- 用户根据需要可自行将其中某些存储元改为“1”（或改为“0”）
- 双极型PROM
 - ◆ 熔丝烧断型PROM
 - ◆ PN结击穿型PROM



多次编程只读存储器

- **EPROM:** 可以用紫外光照射擦除原来写入的数据，编程（即：写入数据）时需要相对较高的电压



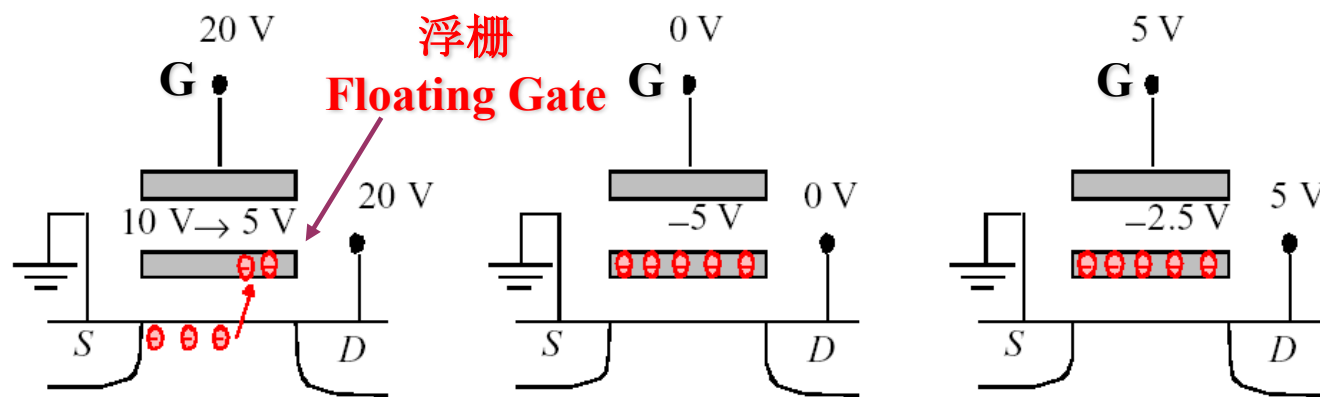
- **EEPROM:** 可以用电擦除原来写入的数据，但擦除、编程时均需要相对较高的电压

光擦可编程只读存储器—编程与读出

工作原理：存储单元为MOSFET+浮空栅，浮空栅被氧化层包裹隔离，可利用雪崩注入（Avalanche Injection）效应向其注入负电荷。未注入负电荷时存储单元为1，注入负电荷后为0。

编程：在漏、源极加高压（如+20V）、控制栅极加高压正脉冲（如50ms宽、20V）后，电子获得能量变热，穿过氧化层进入浮栅，注入负电荷可长期保存。

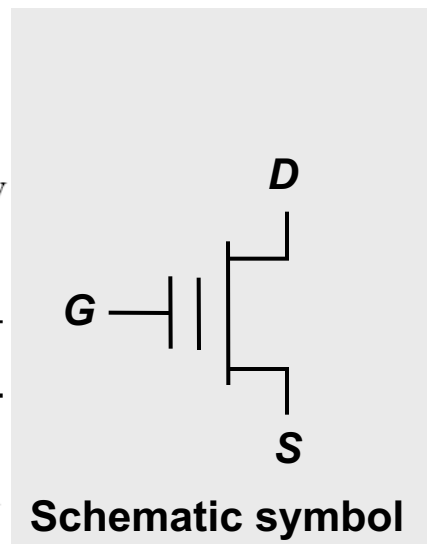
读出：凡注入负电荷的单元，其开启电压 V_T 变高，因此，在正常+5V电压下不能使其导通。



Avalanche injection.

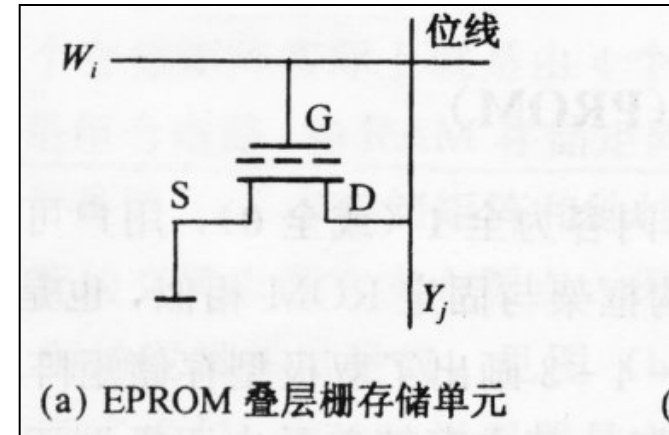
Removing programming voltage leaves charge trapped.

Programming results in higher V_T .



光擦可编程只读存储器—擦除

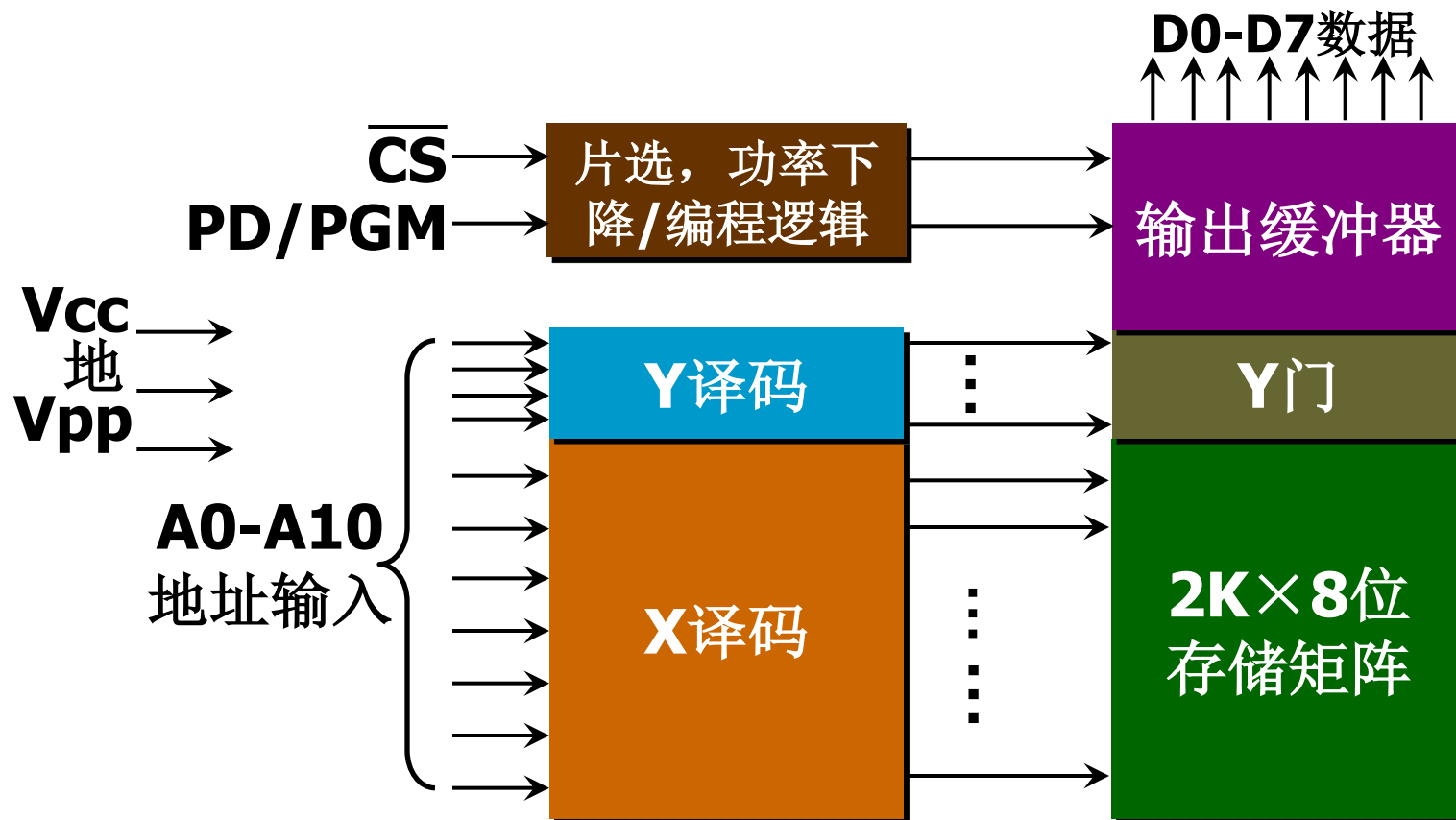
- 这种EPROM做成的片子封装上方有一个石英玻璃窗口。
- 在强紫外线光照射下，使氧化层变得具有导电性，且浮栅上的电子变得更活跃，从而穿过氧化层回到衬底中，使电路恢复起始状态，即把原先写入的“0”信息擦去。
- 擦除过程是对**所有**存储单元进行的。不能实现选择性的擦除。





EPRROM实例 Intel2716

- 容量 $2k \times 8$
- 片子正常工作电源为+5V，脱机编程时给V_{pp}加+25V电源
- 11条地址线中，7条用于行译码，4条用于列译码





举例 (1)

- CPU的地址总线A15-A0，数据总线D7-D0，控制总线中与主存有关的信号有 $\overline{\text{MREQ}}$ ，R/ $\overline{\text{W}}$ 信号。

主存地址空间分配如下：0-8191为系统程序区，由ROM芯片组成；8192-32767为用户程序区；最后(最大地址)2K地址空间为系统程序工作区。上述地址为十进制，按字节编址。

现有如下存储器芯片：EPROM：8K×8位(控制端仅有 $\overline{\text{CS}}$)；SRAM：16K×1位，2K×8位，4K×8位，8K×8位。请从上述芯片中选择适当芯片设计该计算机主存储器，画出主存储器逻辑框图，注意画出选片逻辑（可选用门电路及3:8译码器74LS138）与CPU的连接，说明选哪些存储器芯片，选多少片。





举例 (2)

解：根据给定条件，选用EPROM：8K×8位芯片1片。
SRAM：8K×8位芯片3片，2K×8位芯片1片。

地址区间：

0—8191=0000 0000 0000 0000—0001 1111 1111 1111

8192—32767=0010 0000 0000 0000—0111 1111 1111 1111

63488—65535=1111 1000 0000 0000—1111 1111 1111 1111

选择EPROM时，需要用到3:8译码器的 $\overline{Y0}$ 输出端

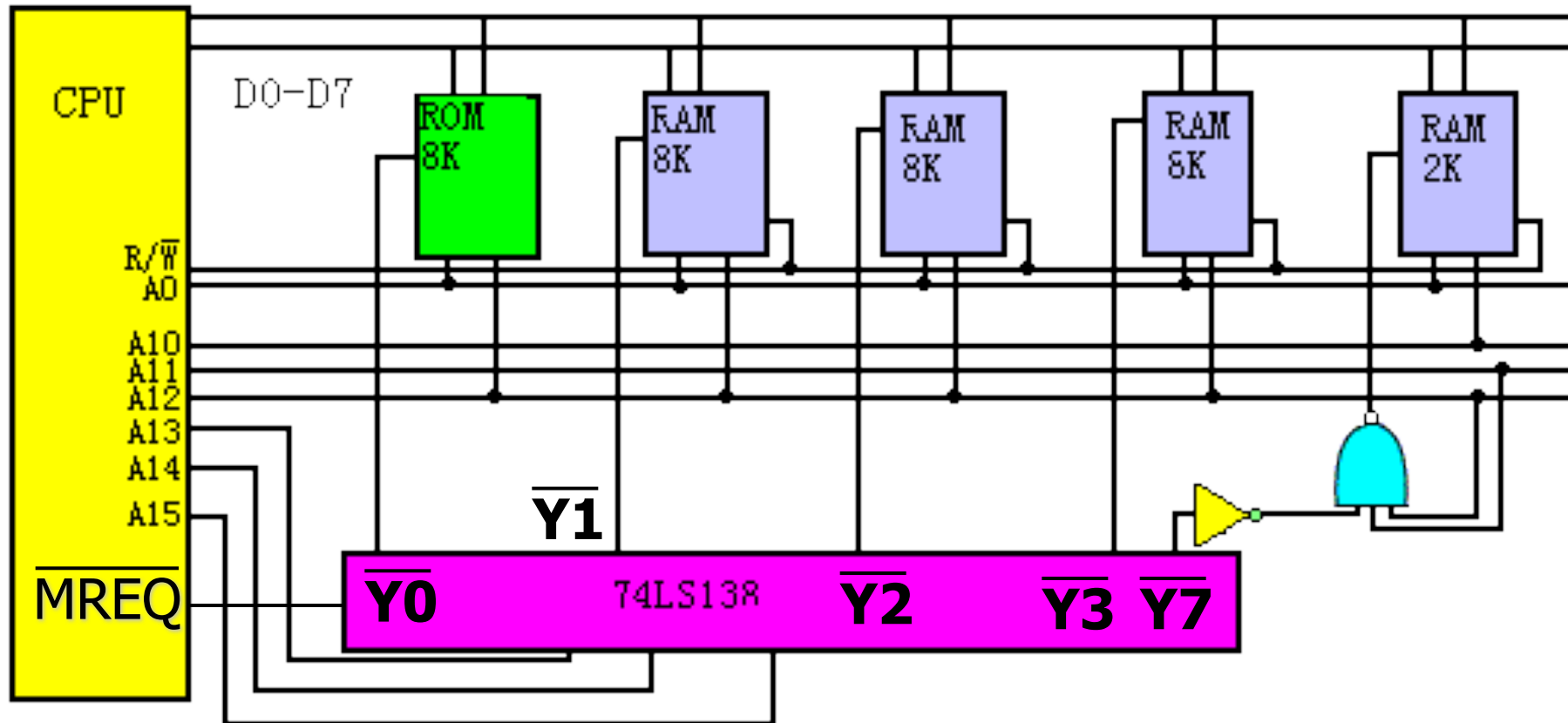
选择3片8K×8位SRAM时，用 $\overline{Y1}/\overline{Y2}/\overline{Y3}$ 输出端

选择2K×8位SRAM时，用 $\overline{Y7}$ 输出端以及A11和A12地址线





举例 (3)





选择题

某计算机主存容量为64KB，其中ROM区为4KB，其余为RAM区，按字节编址。现要用 $2K \times 8$ 位的ROM芯片和 $4K \times 4$ 位的RAM芯片来设计该存储器，则需要上述规格的ROM芯片数和RAM芯片数分别是

- A. 1、15
- B. 2、15
- C. 1、30
- D. 2、30





UV EPROM的型号举例

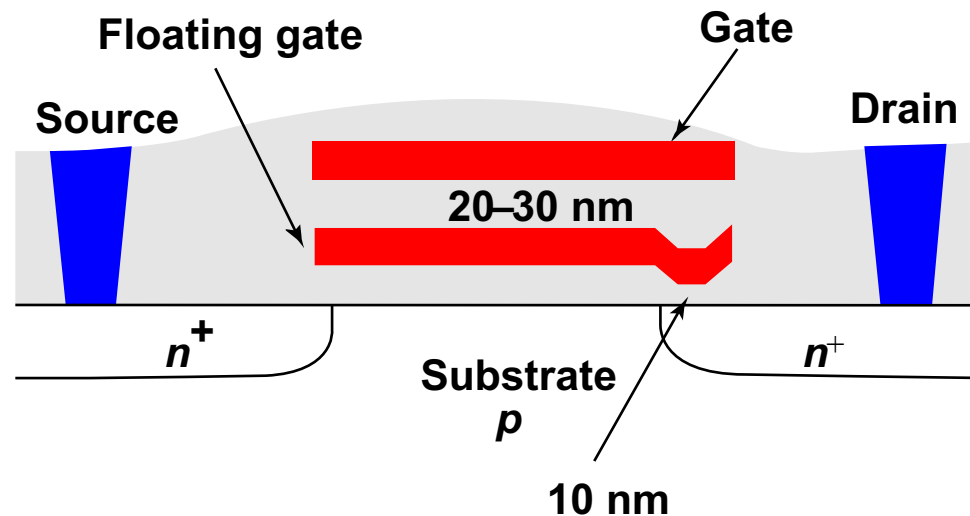
2716	16kbit	2k × 8bit
2732	32kbit	4k × 8bit
2764	64kbit	8k × 8bit
27128	128kbit	16k × 8bit
27256	256kbit	32k × 8bit
27512	512kbit	64k × 8bit
27010	1024kbit	128k × 8bit
27020	2048kbit	256k × 8bit
27040	4096kbit	512k × 8bit
27c040	CMOS EPROM	





EEPROM (1)

- Electrical Erasable PROM, 简称E²PROM
- 存储单元称为Floating-Gate Tunnelling Oxide Transistor (FLOTOX)
- 结构与普通的浮栅晶体管类似，关键改进点为：浮栅的一小部分区域与沟道及漏极的距离小于10nm





EEPROM (2)

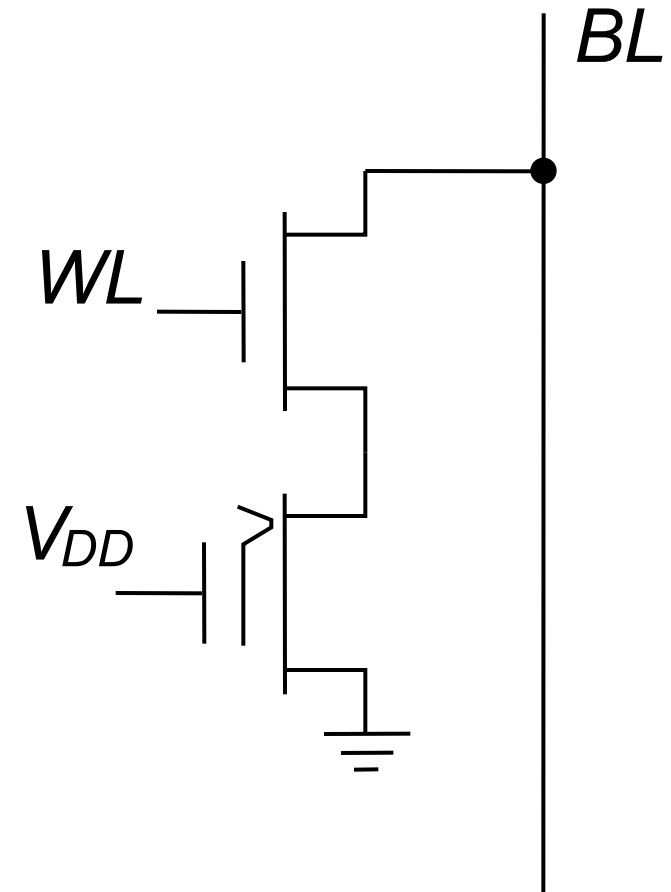
- 大约10v的 V_{GD} 加载到薄的绝缘层上，电子就可以利用这种F-N隧道（Fowler-Nordheim Tunnelling）效应进入浮栅或从浮栅中释放出来。
- 注入电子到浮栅中将升高开启电压 V_T ；反之，则降低。因此，开启电压 V_T 的大小依赖于浮栅中初始电荷和施加的编程电压大小。





EEPROM (3)

- 在实现中，增加相邻的场效应管来实现存取
- 与EPROM相比，集成度低
 - ◆ 2 晶体管/per cell
- 寿命长
 - ◆ 10^5 erase/write cycles.





闪存存储器（1）

■ 闪存分为两类

- ◆ NOR闪存

- ◆ NAND闪存

- 1988年Intel公司发明了NOR闪存，擦除写入时间较长，但有完整的地址/数据接口，能随机存取任一可擦除单元。适用于存储偶尔需要修改程序代码或数据的场合。擦除次数为10,000到100,000次。是在EPROM和EEPROM的基础上发展起来的，既有传统的热电子隧道效应的变成机制，又有EEPROM的F-N隧道效应的擦除特点。存储容量比NAND闪存小很多

- 1989年Toshiba公司发明了NAND闪存，擦除写入时间较快，与NOR闪存比较具有较高的密度和较低的位成本，10倍的擦除次数。但I/O接口只允许顺序存取数据。适用于大容量存储设备，例如各种存储卡、U盘、固态硬盘SSD等





闪速存储器 (2)

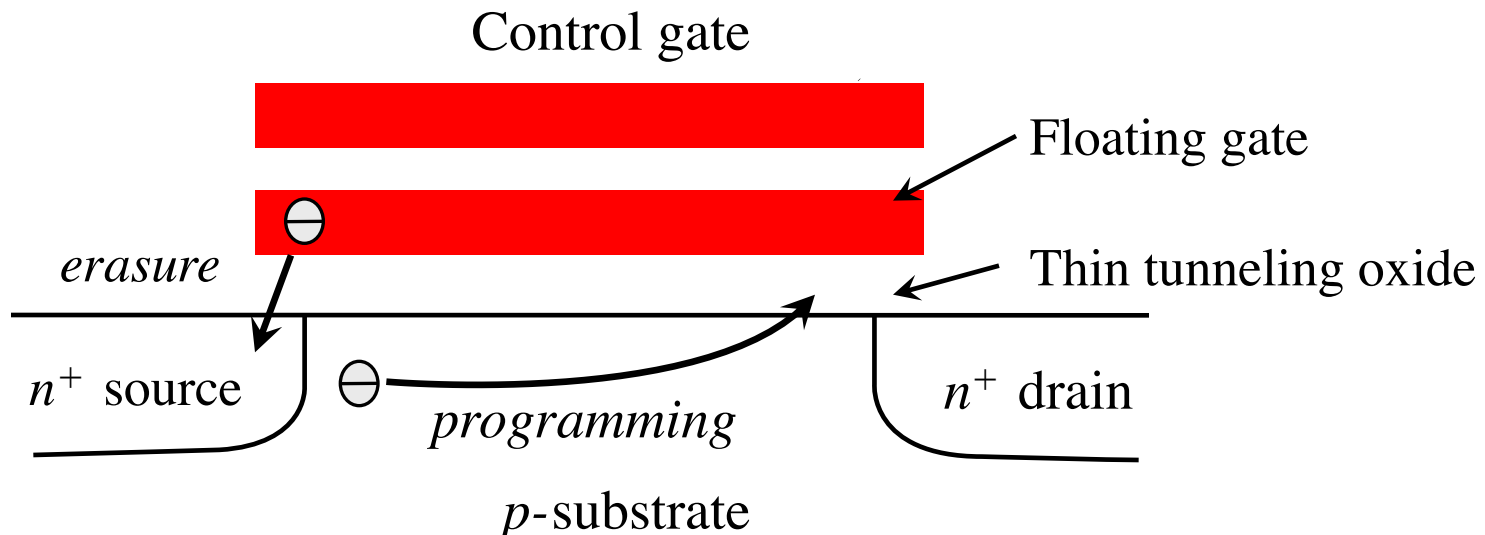
- 闪速存储器是目前唯一的具有大容量、非易失性、低价格、可在线改写和较高速度几个特性共存的存储器
- 之所以称为FLASH，是因为擦除速度快，擦除整个存储矩阵所花时间，与EEPROM擦除一个存储单元的时间相同
- Flash比SRAM成本低，比SRAM集成度高，且信息具有不易失性
- 可以取代DRAM吗？
 - ◆ 擦写次数 10^6 而DRAM可读写次数为 10^{15} 次
 - ◆ 速度：45ns 而DRAM在10ns以内





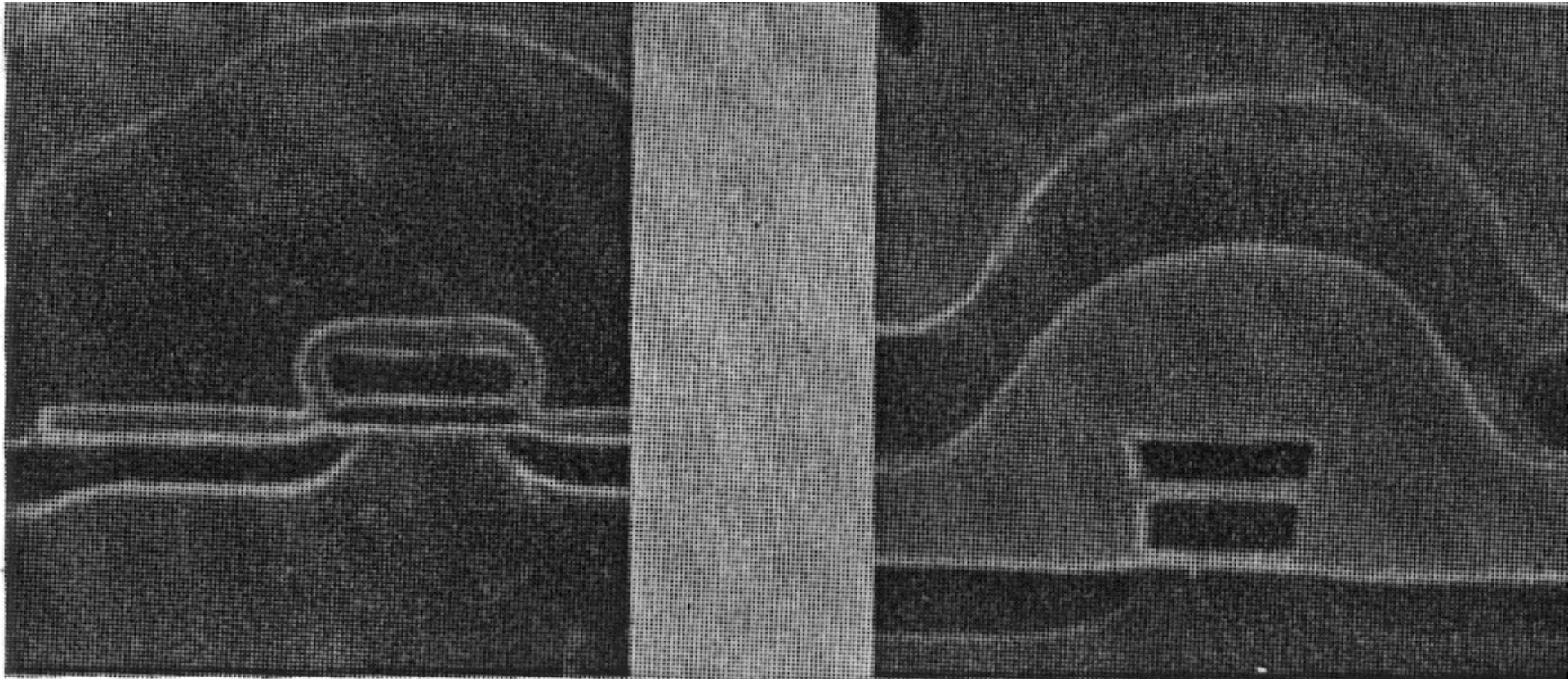
Flash EEPROM

- Flash EEPROM类似浮栅，但是具有很薄的栅极氧化层。使用先进的处理工艺制造精确
- 编程时利用雪崩热电子注入效应
- 擦除时利用F-N隧道效应
 - ◆ 栅极（接地）/源极（+12V）用于快速擦除





Non-Volatile Memory Cells



Flash — 1990

Thin Layers

Excellent performance

EPROM — 1983

Thick Layers

Limited performance





闪存的特点

- 按区块（sector）或页面（page）组织
 - ◆ 可进行字节、区块、页面或整芯片的擦除和编程
- 具有内部编程控制逻辑
 - ◆ 内部设有命令寄存器和状态寄存器
- 具有在系统编程能力
 - ◆ 目前的闪存内部可以自行产生编程电压（VPP）
 - ◆ 在工作状态下，在系统中就可实现编程操作。
- 具有软件和硬件保护能力
- 采用**命令方式**可以使闪存进入各种不同工作状态
 - ◆ 例如整片擦除、页面擦除、整片编程、字节编程、分页编程、进入保护方式、读识别码等





闪存的工作方式

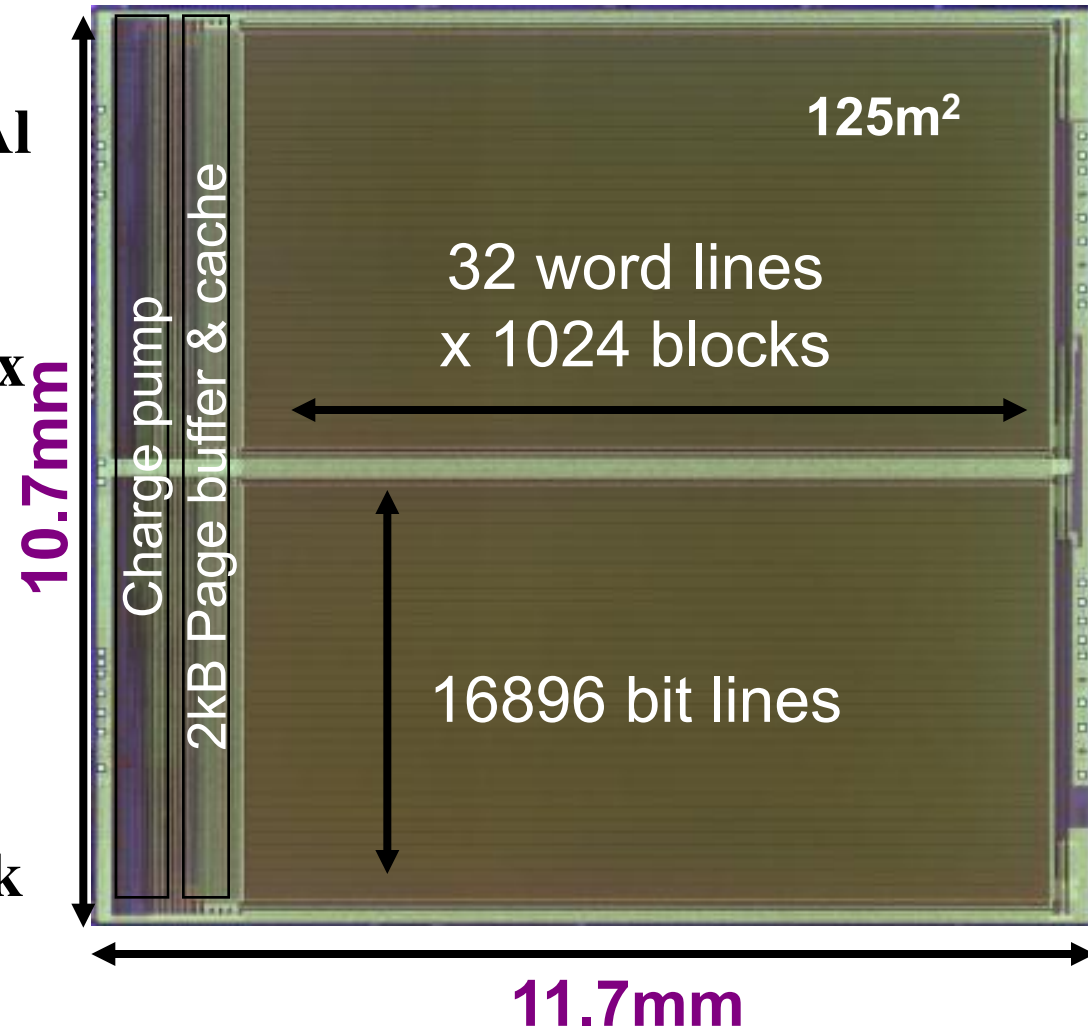
- 上电后，芯片内部的状态机使器件处于**读出操作状态**
- 在读出状态下，闪存的读出操作与其它的各种**ROM**芯片相同。
- 只有在执行了特定的命令序列之后，才可进入其它状态，进行芯片擦除、区块擦除、编程写入、软件数据保护或者读标识码等操作。
- 为防止状态机的误动作，闪存的各种命令是以向特定地址写入特定内容的“命令序列”方式定义的。





1Gbit NAND Flash Memory

- Technology 0.13 μm p-sub CMOS triple-well
1poly, 1polycide, 1W, 2Al
- Cell size 0.077 μm^2
- Chip size 125.2mm²
- Organization 2112 x 8b x 32 x 1024 block x 2
- Power supply 2.7V-3.6V
- Cycle time 50ns
- Read time 25 μs
- Program time 200 μs / page
- Erase time 2ms / block





闪存的应用

- 特点：可在线写入数据，具有ROM的非易失性
- 可以取代全部的UV EPROM和大部分的E²PROM
- 闪存的主要用途：
 - ◆ 存储监控程序、引导程序等基本不变或不经
常改变的程序；可使用NOR闪存
 - ◆ 保存掉电时需要保持的系统配置等不常改变
的数据；可使用NOR闪存
 - ◆ 固态硬盘：采用NAND闪存





选择题

下列有关RAM和ROM的叙述中，正确的是（ ）

- I、 RAM是易失性存储器， ROM是非易失性存储器
 - II、 RAM和ROM都是采用随机存取的方式进行信息访问
 - III、 RAM和ROM都可用作Cache
 - IV、 RAM和ROM都需要进行刷新
- A: 仅I和II
- B: 仅II和III
- C: 仅I,II, III
- D: 仅II, III, IV





3.5 并行存储器

- 双端口存储器
- 多模块交叉存储器





提高存储系统访问速度的途径

- 芯片技术—提高单个芯片的访问速度
 - ◆ 选用更高速的半导体器件
 - ◆ 改善芯片内部结构和对外接口方式
- 结构技术—改进存储器与CPU之间的连接方式
 - ◆ 单体多字存储器
 - ◆ 多体交叉存储器（即：多体并行）
- 系统结构技术—从整个存储系统的角度采用分层存储结构
 - ◆ cache
 - ◆ 虚拟存储器

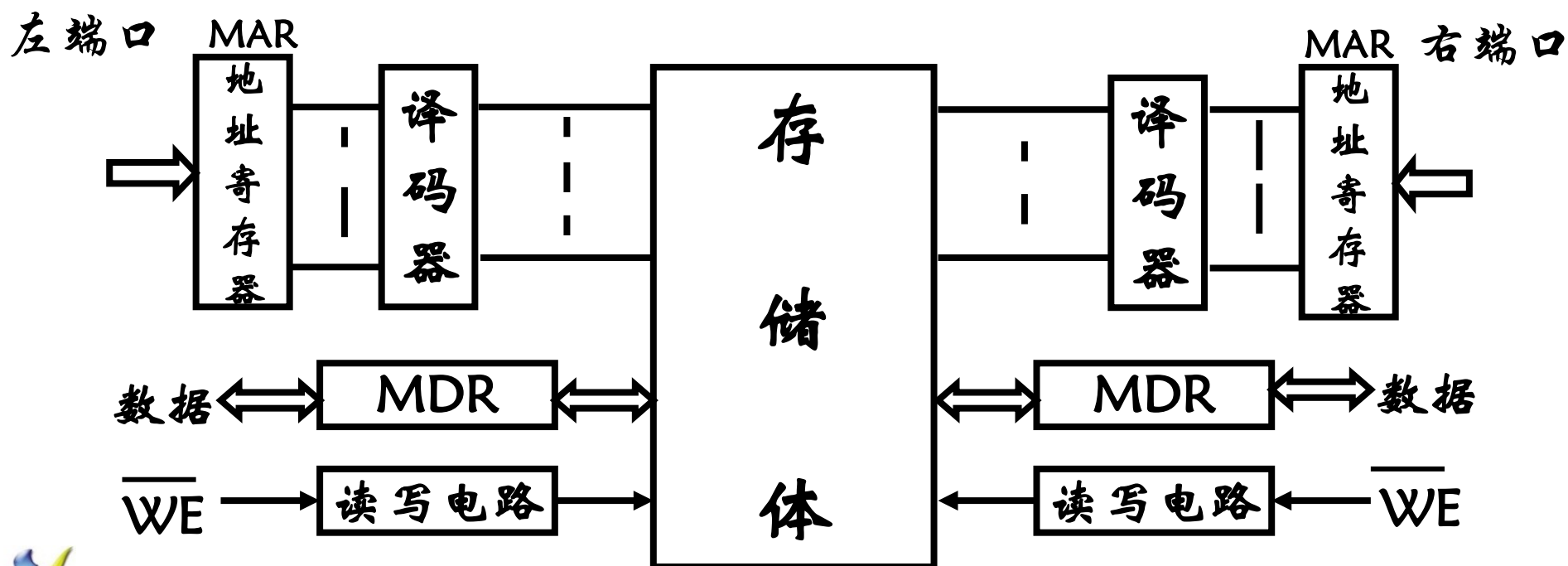




双端口存储器DPRAM

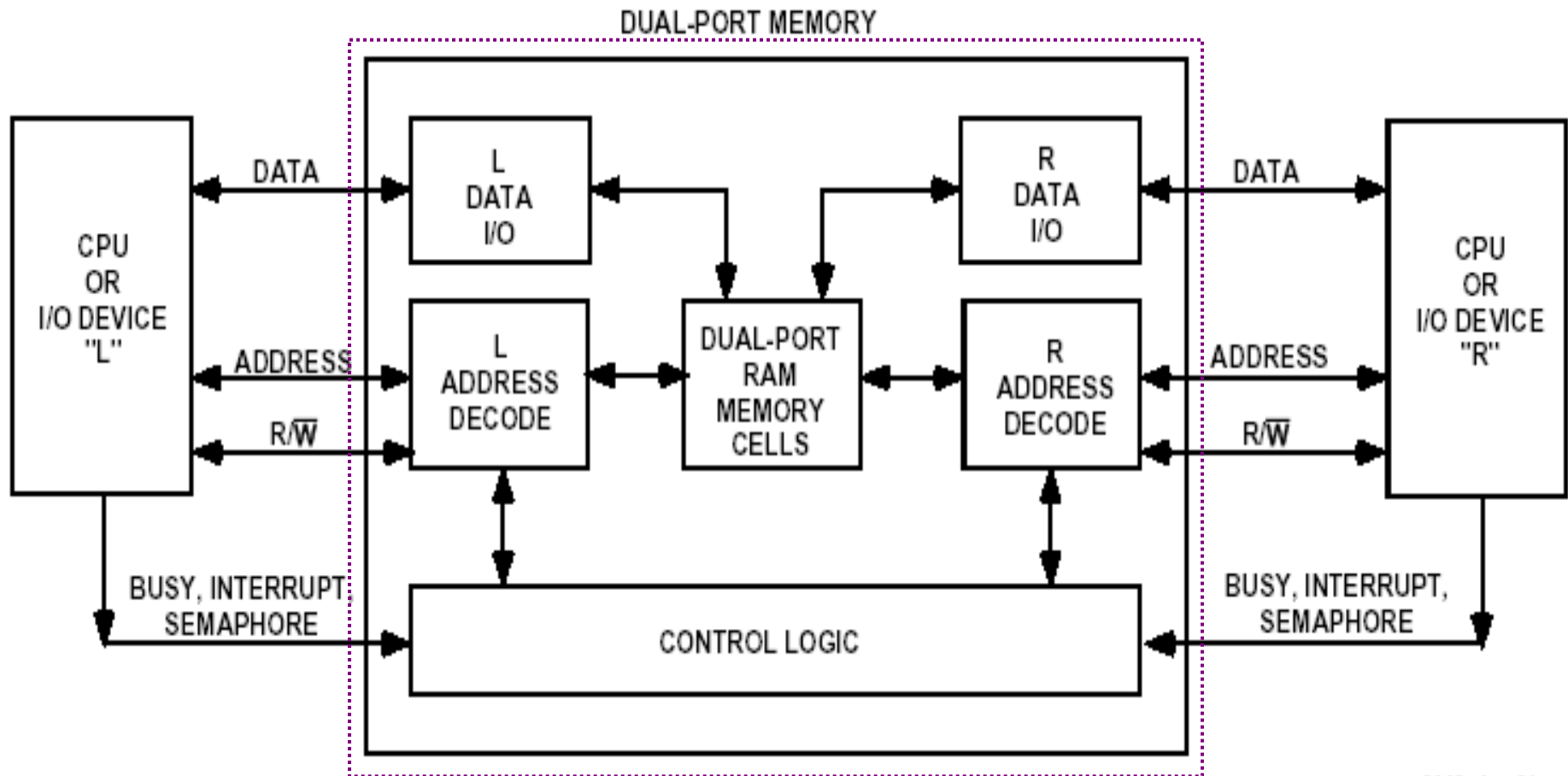
■ 双端口存储器（Dual Port RAM）

- ◆ 双端口存储器是指同一个存储器具有**两组相互独立**的读写控制线、地址线和数据线
- ◆ 提供了两个相互独立的端口（左端口和右端口），可以进行并行的独立操作





DPRAM的应用



2648 drw 01

Figure 1. Dual-Port Memory Block Diagram





双端口存储器实例

■ 双端口存储器IDT7133

- ◆ 容量为 $2K \times 16$ 位SRAM
- ◆ 两个相互独立的端口，即左端口和右端口
- ◆ 左、右端口分别具有各自的地址线、数据线和控制线
- ◆ 图中下标L表示左端口，R表示右端口，LB表示低位字节，UB表示高位字节

管脚说明

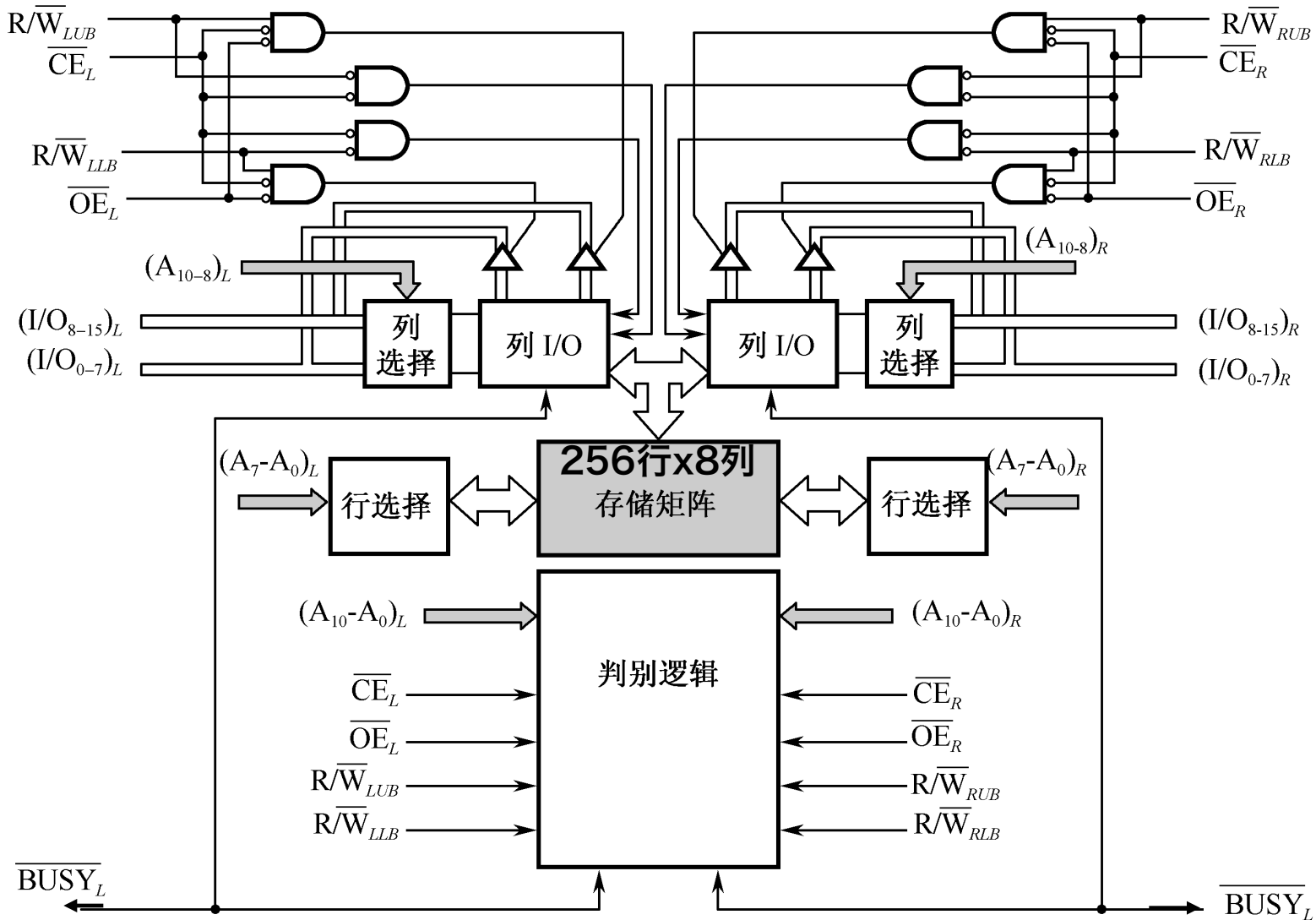
Pin Names

Left Port	Right Port	Names
$\overline{CE_L}$	$\overline{CE_R}$	Chip Enable
R/\overline{W}_{LUB}	R/\overline{W}_{RUB}	Upper Byte Read/Write Enable
R/\overline{W}_{LLB}	R/\overline{W}_{RLB}	Lower Byte Read/Write Enable
$\overline{OE_L}$	$\overline{OE_R}$	Output Enable
$A_{0L} - A_{10L}$	$A_{0R} - A_{10R}$	Address
$I/O_{0L} - I/O_{15L}$	$I/O_{0R} - I/O_{15R}$	Data Input/Output
$\overline{BUSY_L}$	$\overline{BUSY_R}$	Busy Flag
V_{CC}		Power
GND		Ground





IDT7133功能方框图





读写操作

■ 无冲突读写控制

- ◆ 当两个端口存取的存储单元的**地址不相同**时，读写操作不会产生冲突

■ 有冲突的读写控制

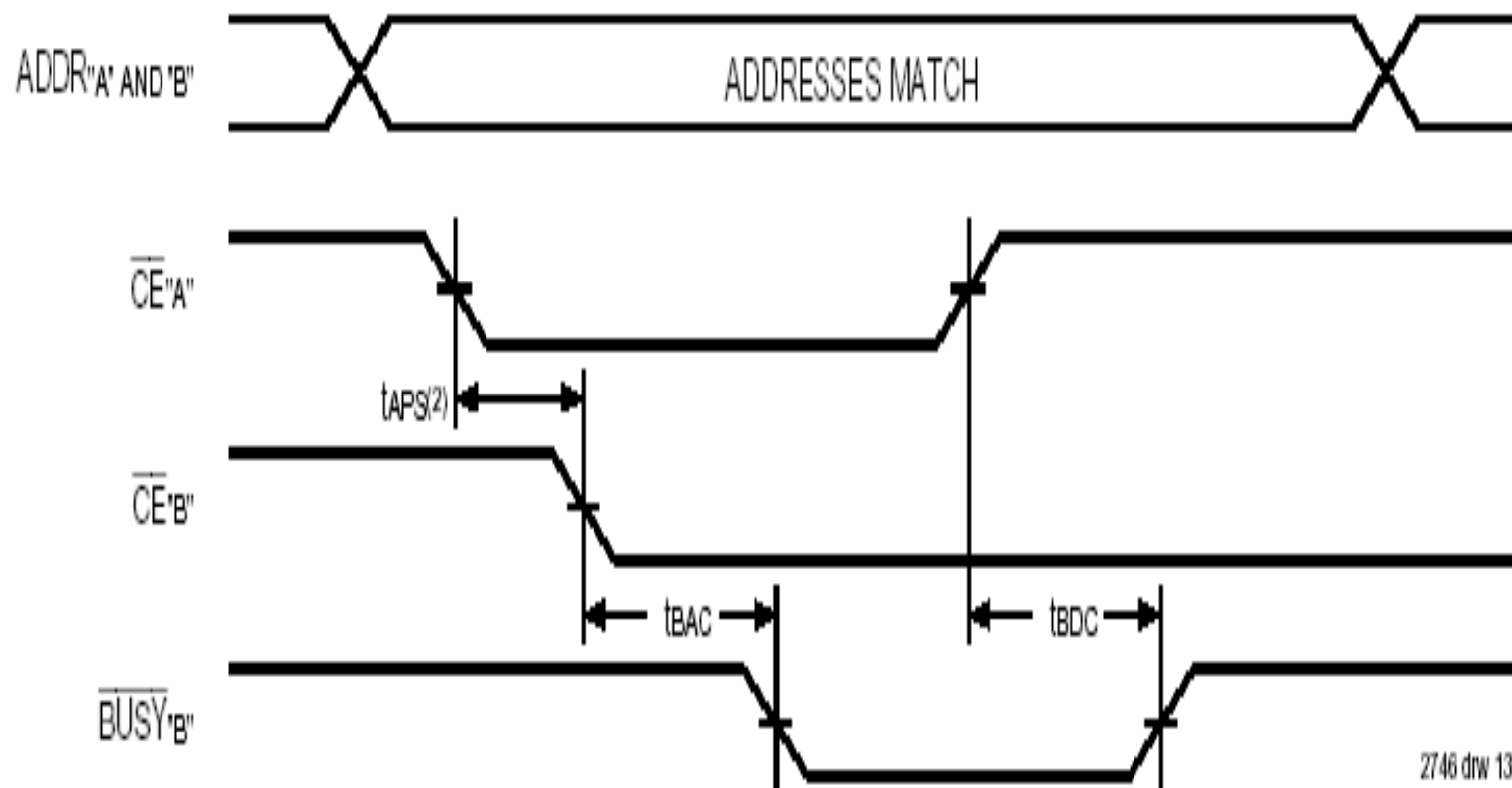
- ◆ 当两个端口同时存取存储器**同一**存储单元时，便发生读写冲突
- ◆ 为了解决读写冲突，设置了 $\overline{\text{BUSY}}$ 标志线
- ◆ 由判断逻辑部件来决定对哪个端口优先进行读写操作，另一个端口的 $\overline{\text{BUSY}}$ 标志有效，读写操作延迟执行。
- ◆ 判断逻辑部件的判断方式：
 - **CE判断**：如果地址匹配且在CE之前有效，片上的控制逻辑在 CE_L 和 CE_R 之间进行判断来选择端口
 - **地址有效判断**：如果CE在地址匹配之前变低，片上的控制逻辑在左、右地址间进行判断来选择端口





/CE控制的仲裁

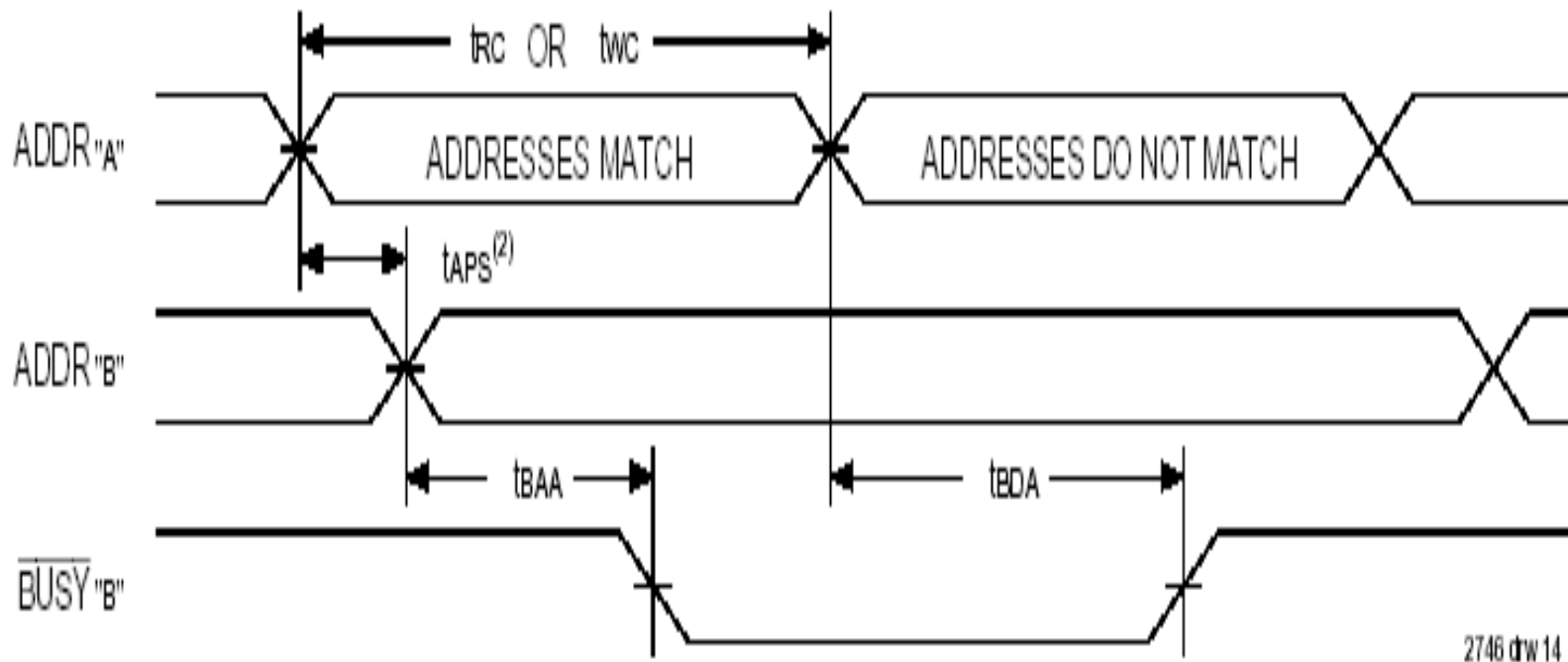
Timing Waveform of \overline{BUSY} Arbitration Controlled by \overline{CE} Timing⁽¹⁾





地址控制的仲裁

Timing Waveform of $\overline{\text{BUSY}}$ Arbitration Controlled by Addresses⁽¹⁾





多模块存储器

■ 存储器的模块化组织

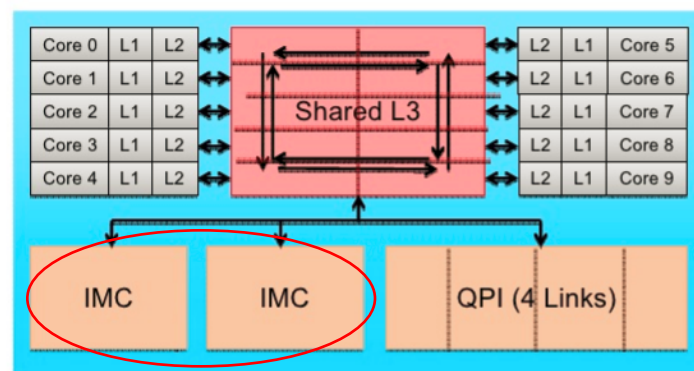
- ◆ 由若干个模块组成的主存储器是线性编址的
- ◆ 然而，各模块地址安排有2种方式

□ 顺序方式

□ 交叉方式(Interleaving)

也称为多通道方式

Intel® Xeon® E7 Processor Architecture



Cache Architecture

- 64K L1 Cache
- 256K L2 Cache
- 30MB 10 slice shared Last Level cache (L3) (compared to 24MB 8 slice L3 on Xeon® 7500)

- 2 integrated memory controllers
 - Scalable Memory Interconnect (SMI) with support for up to 8 DDR channels
- 4 Quick Path Interconnect (QPI) system interconnect links



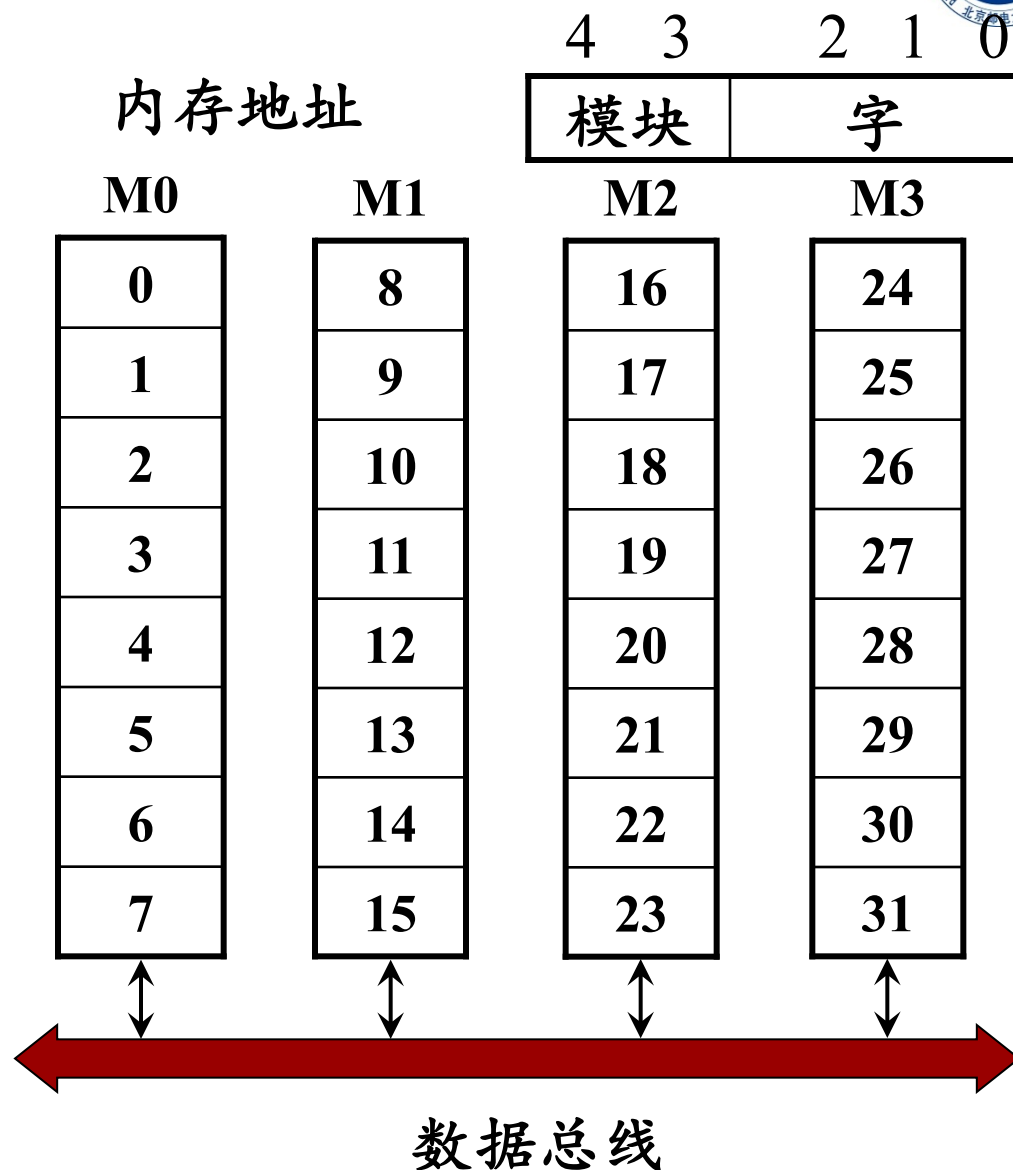


顺序方式

- 设存储器容量为32字，分成四个模块
- 访问地址按顺序分配给每一个模块
- 地址寄存器5位，其中高2位选择4个模块中的一个，低3位选择每个模块中的8个字

特点：

- 只能**串行访问**各模块，存储器的带宽受到了限制
- 某一模块出现故障时，其他模块可以照常工作
- 增加模块可扩充存储器容量



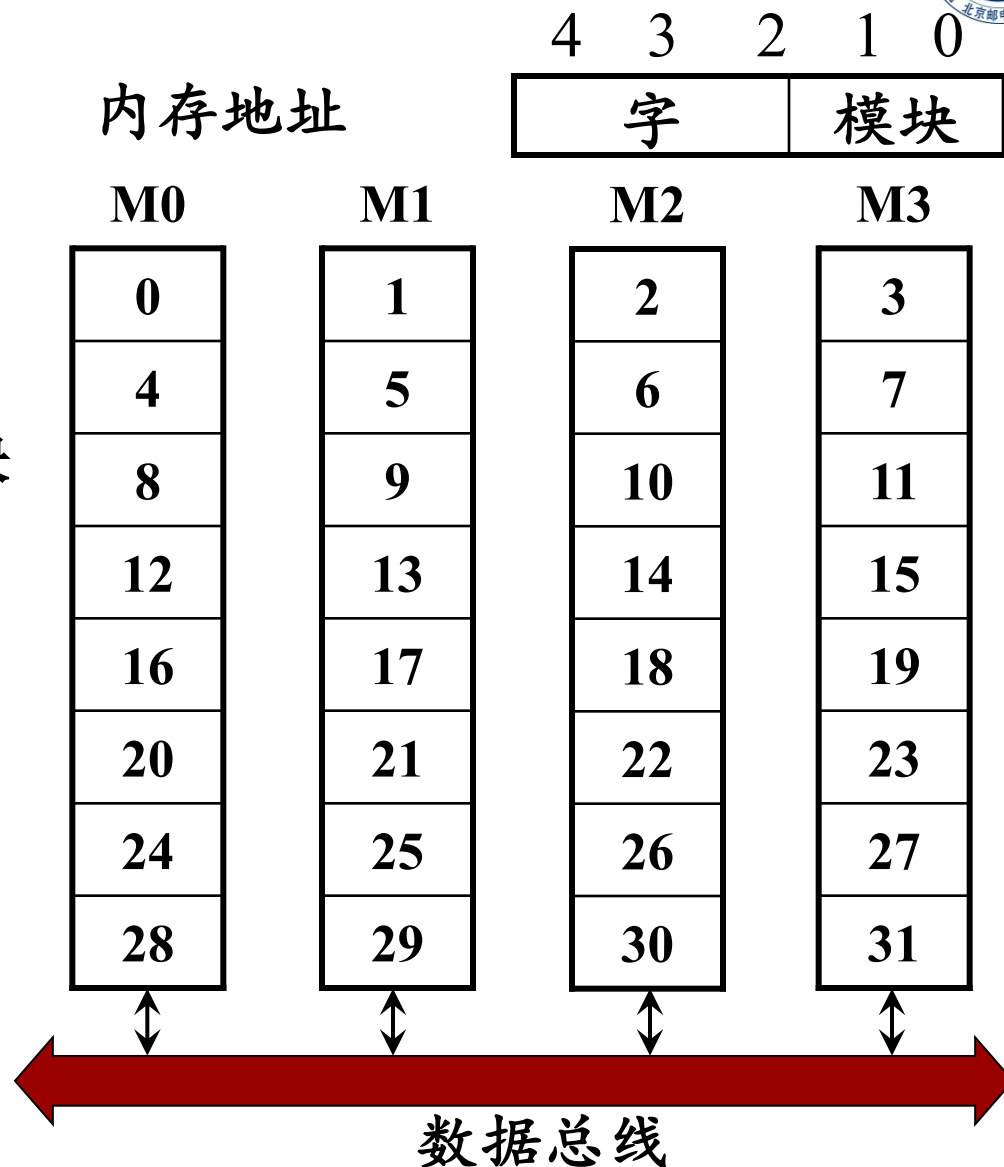


交叉方式

- 设存储器容量为32字，分成四个模块
- 线性地址0、1、2、3依次分配给M0、M1、M2、M3，.....
- 地址寄存器5位，低2位选择模块，而高3位选择模块中的字

特点：

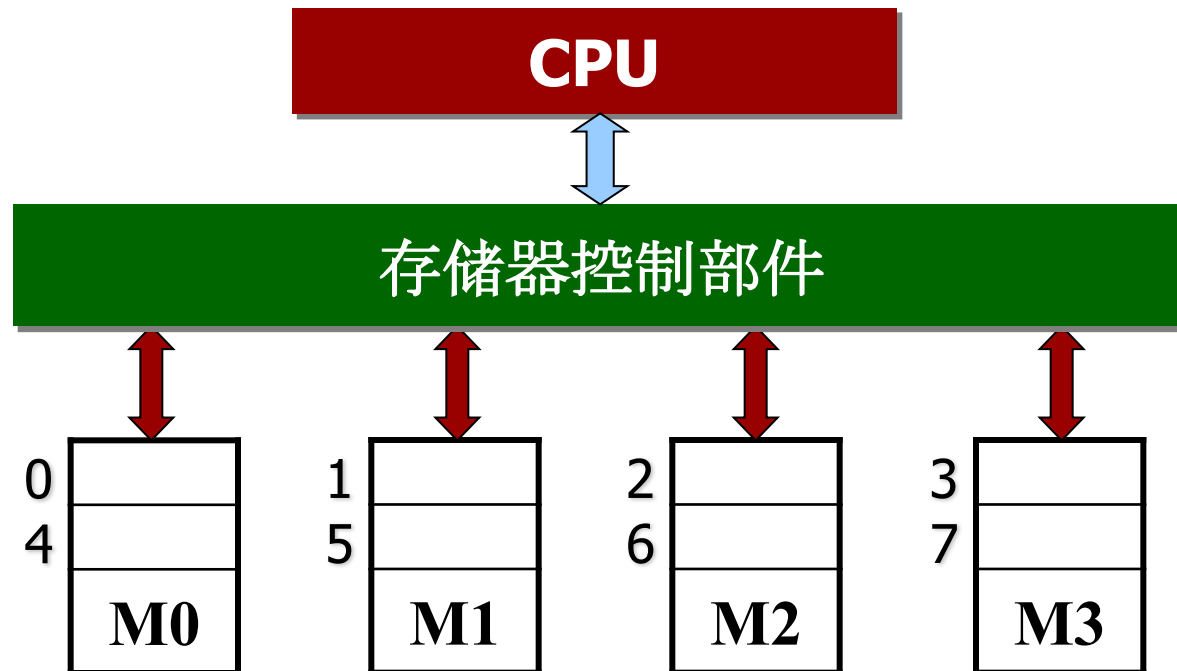
- 地址码的低位字段选择不同的模块，高位字段确定相应模块内的存储字
- 连续地址分布在相邻的不同模块内，同一个模块内的地址都是不连续的
- 对连续字的成块传送可实现多模块流水式并行存取，大大提高存储器的带宽





多模块交叉存储器的基本结构

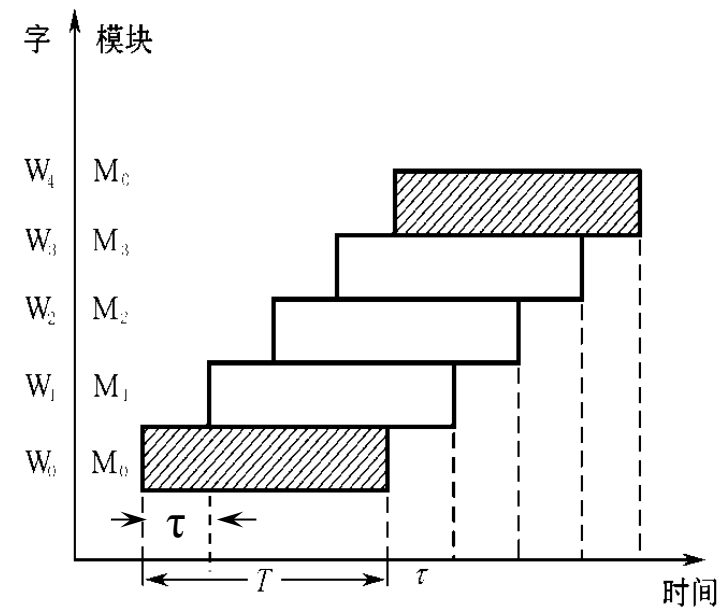
- 主存被分成4个独立、容量相同的模块M0/M1/ M2/M3
- 每个模块均有**自己的**读写控制电路、地址寄存器和数据寄存器，以相同的方式与CPU传送信息
- CPU同时访问四个模块，由存储器控制部件控制它们分时使用数据总线进行信息的传递





多模块交叉存储器的特点 (1)

- 对一个存储模块来说，从CPU给出访存命令直到读出信息仍然使用了一个存取周期 T
- CPU来可以连续访问四个模块
- 各个模块的读写过程几乎重叠，是一种存储器并行访问方式。 τ 为总线传送周期





多模块交叉存储器的特点 (2)

- 假设模块字长等于数据总线宽度，且模块存取一个字的存储周期为 T ，总线传送周期为 τ ，存储器的交叉模块数为 m
- 那么， $m = T / \tau$ 被为交叉存取度，连续读取 m 个字所需的时间为 $t_1 = T + (m-1) \tau$
- 对于顺序方式存储器，连续读取 m 个字所需的时间为 $t_2 = mT$

	1	2	3	4	5	6	7	8	9	10	11	12
Bank 0	0	0	0		4	4	4					
Bank 1		1	1	1		5	5	5				
Bank 2			2	2	2		6	6	6			
Bank 3				3	3	3		7	7	7		
Bus				0	1	2	3	4	5	6	7	





例4

- 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 $\tau=50\text{ns}$ 。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：

$$q=64\text{位} \times 4=256\text{位}$$

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：

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

$$t_1=T+(m-1)\tau=200\text{ns}+3 \times 50\text{ns}=350\text{ns}=3.5 \times 10^{-7}\text{s}$$

顺序存储器和交叉存储器的带宽分别是：

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

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





二模块交叉存储器举例

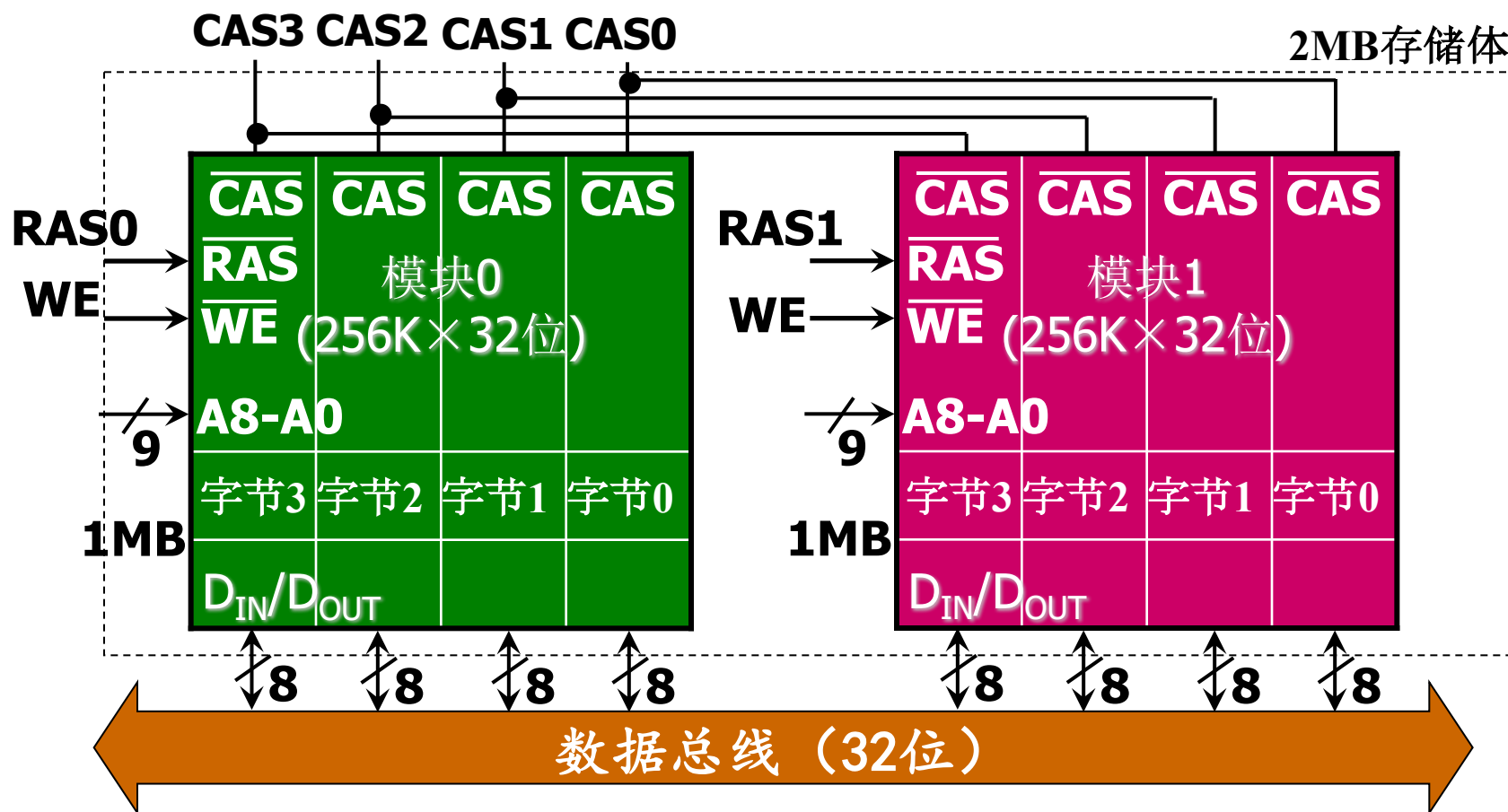
- 数据总线的宽度为32位，地址总线宽度为24位(16MB)
- 共有8个2MB存储体，地址高3位地址用于存储体选择(字扩展)
- 每个存储体由2个模块组成。每个模块的容量为1MB($256K \times 32$ 位)，由8片 $256K \times 4$ 位的DRAM芯片组成(位扩展)，存储体的总容量为2MB($512K \times 32$ 位)
- A20-A3的18位地址用于模块中256K个存储字的选择，分为行、列地址两部分送至芯片的9位地址引脚
- A2用于模块选择，A2=0，RAS0有效，选中模块0；A2=1，RAS1有效，选中模块1。连续的32位存储字交叉分布在两个模块上，偶地址在模块0，奇地址在模块1



二模块交叉存储器举例

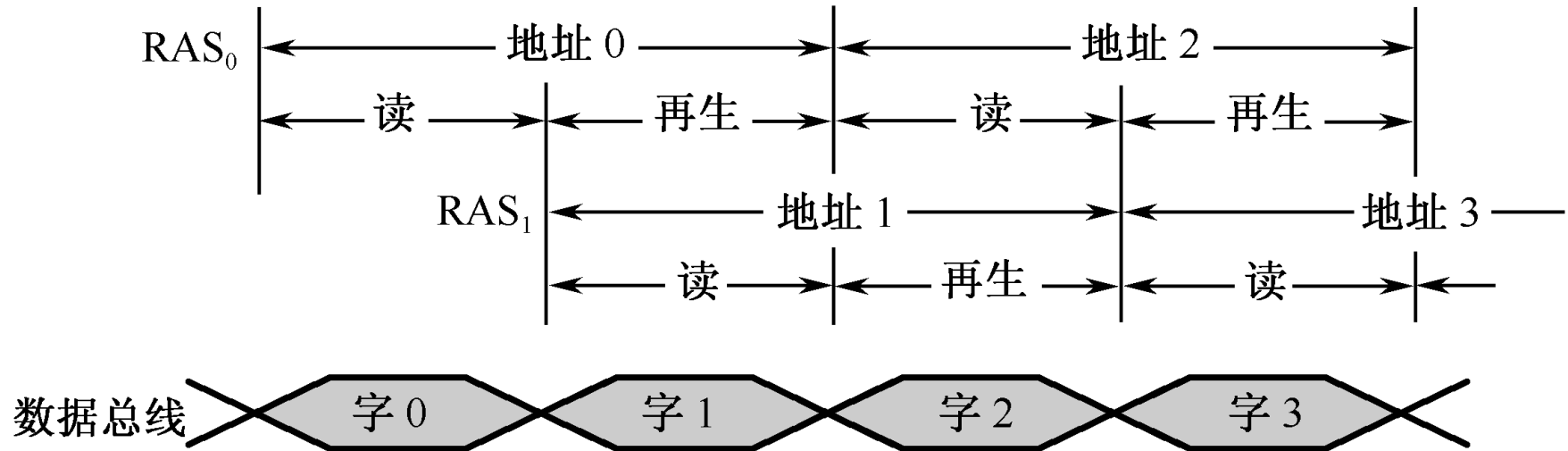
存储地址

$A_{23} A_{21} A_{20}$	$A_3 A_2$	$BE_{3 \sim 0}$
体	块内字地址	块 字节允许





零等待存取

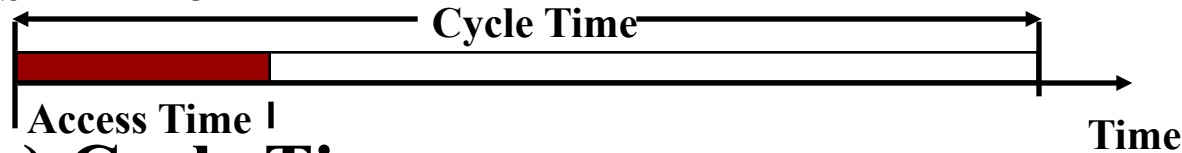




Main Memory Performance

■ DRAM (Read/Write) Cycle Time \gg DRAM (Read/Write) Access Time

◆ 2:1; why?



■ DRAM (Read/Write) Cycle Time :

◆ How frequent can you initiate an access?

◆ Analogy: A little kid can only ask his father for money on Saturday

■ DRAM (Read/Write) Access Time:

◆ How quickly will you get what you want once you initiate an access?

◆ Analogy: As soon as he asks, his father will give him the money

■ DRAM Bandwidth Limitation analogy:

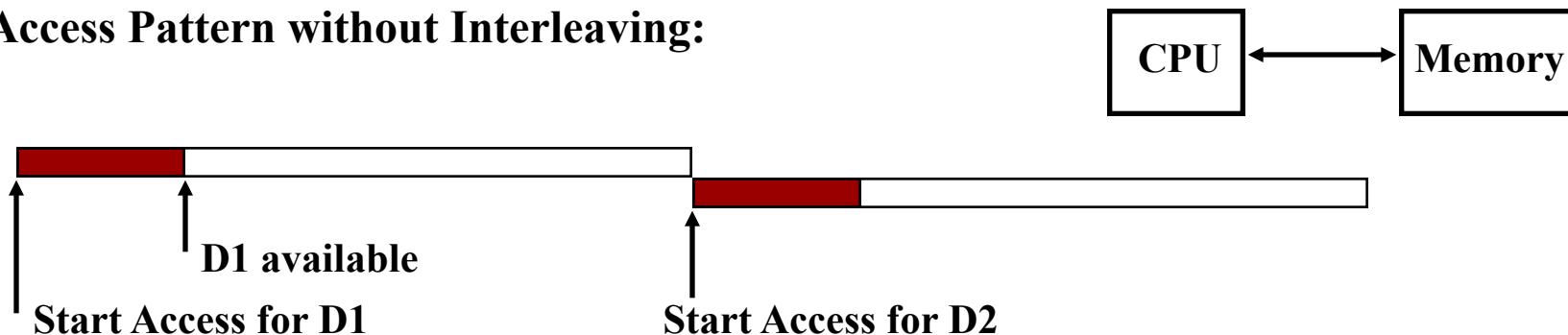
◆ What happens if he runs out of money on Wednesday?



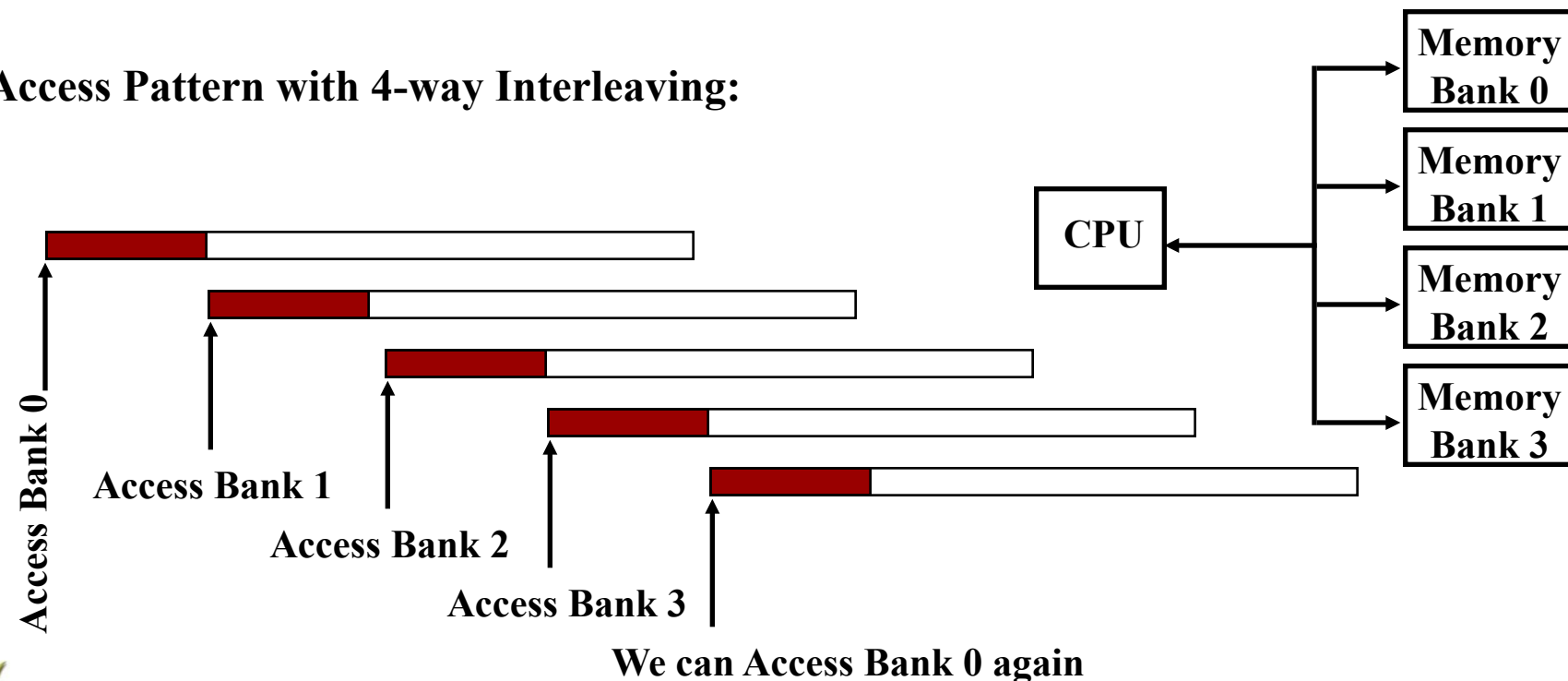


Increasing Bandwidth - Interleaving

Access Pattern without Interleaving:

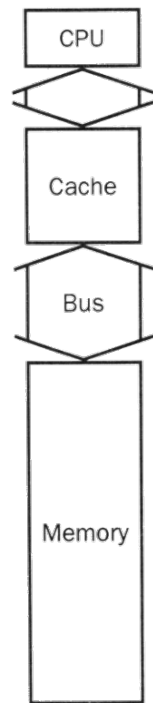


Access Pattern with 4-way Interleaving:

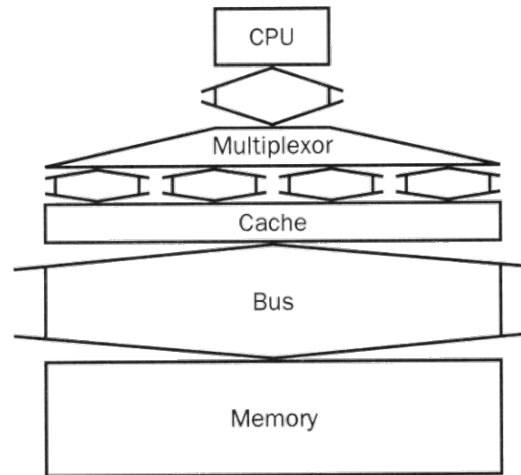




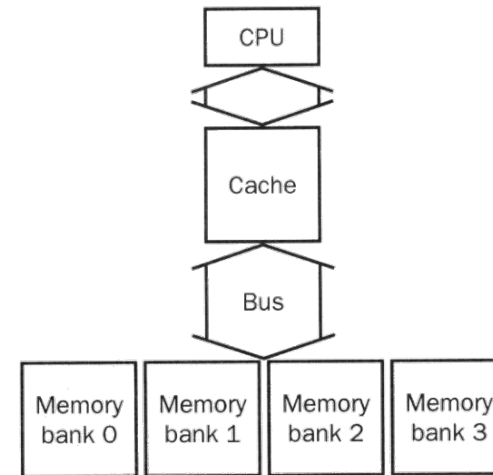
Main Memory Performance



a. One-word-wide memory organization



b. Wide memory organization



c. Interleaved memory organization

- **Wide:**

- CPU/Mux 1 word;
Mux/Cache, Bus,
Memory N words
(Alpha: 64 bits & 256 bits)

- **Interleaved:**

- CPU, Cache, Bus 1 word;
Memory N Modules
(4 Modules); example is
word interleaved

■ **Simple:**

- ◆ CPU, Cache, Bus, Memory same width
(32 bits)

