第4章 存储器

大 纲



- (一)存储器的分类
- (二)存储器的层次化结构
- (三)半导体随机存取存储器
 - 1. SRAM存储器的工作原理
 - 2.DRAM存储器的工作原理
- (四)只读存储器
- (五)主存储器与CPU的连接
- (六)双口RAM和多模块存储器
- (七)高速缓冲存储器(Cache)
 - 1. 程序访问的局部性原理
 - 2. Cache的基本工作原理

- 3. Cache和主存之间的映射方式
- 4. Cache中主存块的替换算法
- 5. Cache写策略

(八) 虚拟存储器

- 1. 虚拟存储器的基本概念
- 2. 页式虚拟存储器
- 3. 段式虚拟存储器
- 4. 段页式虚拟存储器
- 5. TLB (快表)

Contents



4.1 概 述

回顾: 存储器基本术语

- 记忆单元 (存储基元 / 存储元 / 位元) (Cell)
 - 具有两种稳态的能够表示二进制数码0和1的物理器件
- 存储单元 / 编址单位 (Addressing Unit)
 - 主存中具有相同地址的位构成一个存储单元, 也称为一个编址单位
- 存储体/存储矩阵/存储阵列 (Bank)
 - 所有存储单元构成一个存储阵列
- 编址方式 (Addressing Mode)
 - 字节编址、按字编址
- 存储器地址寄存器 (Memory Address Register MAR)
 - 用于存放主存单元地址的寄存器
- 存储器数据寄存器 (Memory Data Register-MDR (MDR))
 - 用于存放主存单元中的数据的寄存器

4.1 概 述

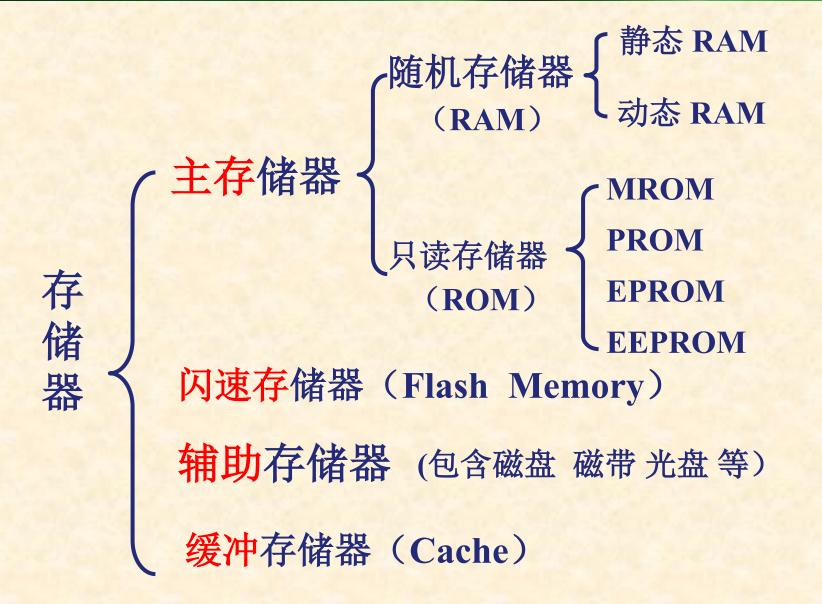
- 一、存储器分类
 - 1. 按存储介质分类
 - (1) 半导体存储器 双极型 静态MOS型 动态MOS型
 - (2) 磁表面存储器 磁盘 (Disk)、磁带 (Tape)
 - (3) 磁芯存储器 硬磁材料、环状元件 (破坏性读出)
 - (4) 光盘存储器

一、存储器分类

2. 按存取方式分类

- (1)存取时间与物理地址无关(随机访问)
 - · 每个单元读写时间一样,且与各单元所在位置无关。如:内存。
 - 随机存储器 在程序的执行过程中 可 读 可 写
 - 只读存储器 在程序的执行过程中 只读
- (2) 存取时间与物理地址有关(串行访问)
 - ◆ 顺序存取存储器 (Sequential Access Memory, SAM)
- 数据按顺序从存储载体的始端读出或写入,因而存取时间的长短与信息所在位置有关。例如:磁带。
- ◆ 直接存取存储器 (Direct Access Memory, DAM)
- 直接定位到要读写的数据块,在读写某个数据块时按顺序进行。例如:磁盘。
- ◆ 相联存储器(Associate Memory/Content Addressed Memory CAM)
 - 按内容检索到存储位置进行读写。例如: 快表。

3. 按在计算机中的作用分类



4. 按断电后信息的可保存性分类

- 非易失(不挥发)性存储器(Nonvolatile Memory)
 - 信息可一直保留, 不需电源维持。

(如: ROM、磁表面存储器、光存储器等)

- 易失(挥发)性存储器(Volatile Memory)
 - 电源关闭时信息自动丢失。(如:RAM、 Cache等)

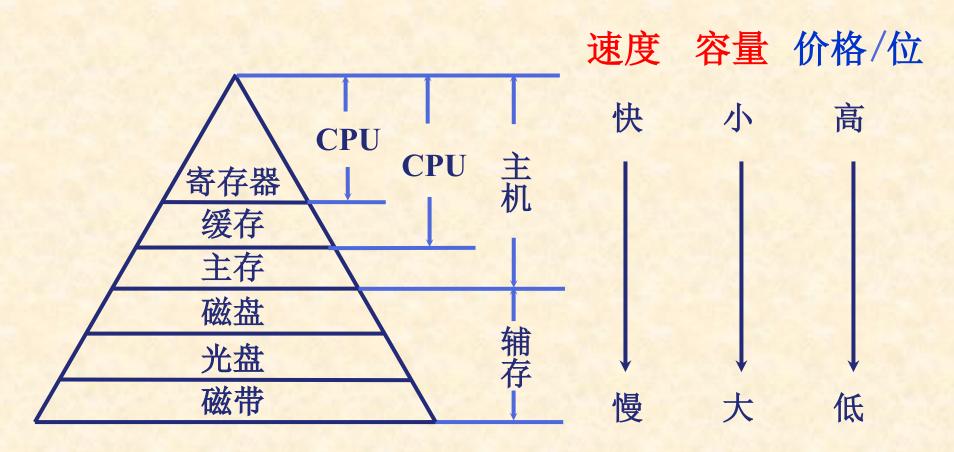
5. 按功能/容量/速度/所在位置分类

- 寄存器(Register)
 - 封装在CPU内,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(几十个)
- 高速缓存(Cache)
 - 位于CPU内部或附近,用来存放当前要执行的局部程序段和数据
 - 用SRAM实现,速度可与CPU匹配,容量小(几MB)
- 内存储器MM (主存储器Main (Primary) Memory)
 - 位于CPU之外,用来存放已被启动的程序及所用的数据
 - 用DRAM实现,速度较快,容量较大 (几GB)
- 外存储器AM (辅助存储器Auxiliary / Secondary Storage)
 - 位于主机之外,用来存放暂不运行的程序、数据或存档文件
 - 用磁表面或光存储器实现,容量大而速度慢

二、存储器的层次结构

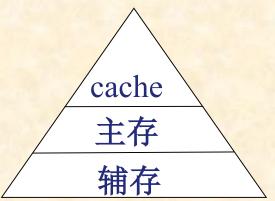


1. 存储器三个主要特性的关系

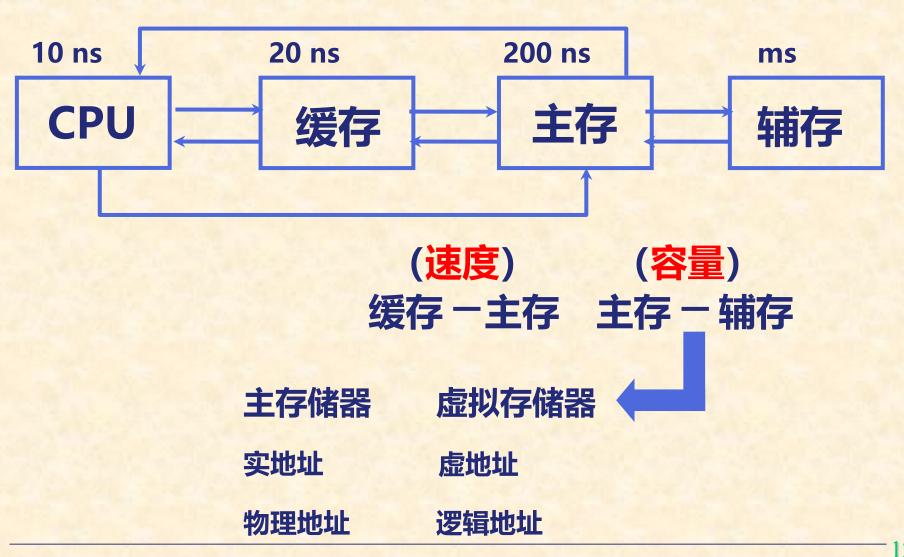


2. 三级存储器结构

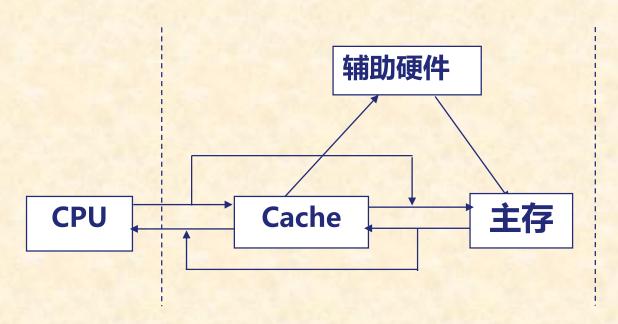
- 由高速缓冲存储器、主存储器和外存储器组成
 - 内存储器: CPU能直接访问的存储器
 - 包括高速缓冲存储器和主存储器
 - 外存储器: CPU不能直接访问的存储器,外存储器的信息必须调入内存储器后才能由CPU进行处理



缓存—主存层次和主存—辅存层次

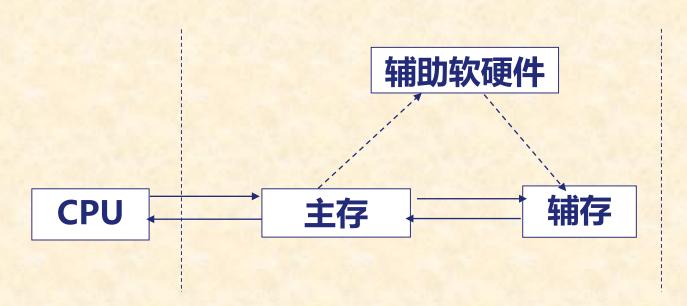


Cache-主存存储层次(Cache存储系统)



- · Cache存储系统是为了解决主存速度不足而提出来的。
- · 在Cache和主存之间增加辅助硬件,让它们构成一个整体
- ·从CPU看,速度接近Cache,容量是主存的容量,位价格接近于主存价格
- 由于Cache存储系统全部采用硬件来调度,因此对系统程序员和应用程序 员是透明的

主存 - 辅存存储层次 (虚拟存储系统)

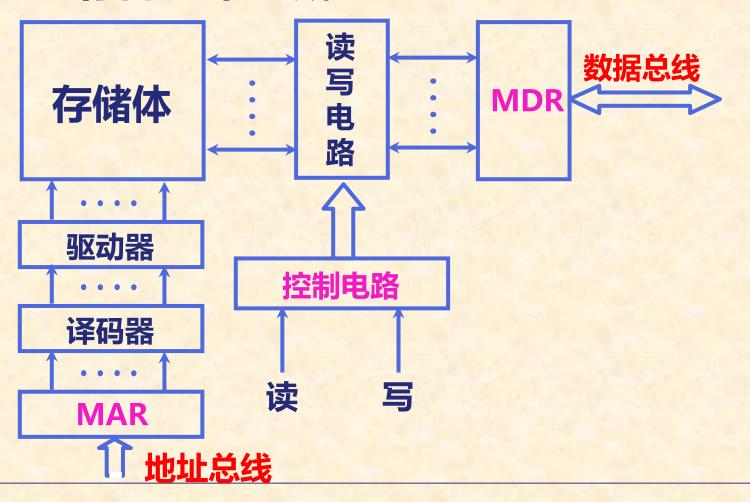


- 虚拟存储器是为了解决主存容量不足而提出来的。
- 在主存和辅存之间,增加辅存的软硬件,让它们构成一个整体
- 从CPU看,速度接近主存的速度,容量是虚拟的地址空间,每位价格 是接近辅存的价格
- 由于虚拟存储系统需要通过操作系统来调度,因此对系统程序员是不透明的,但对应用程序员是透明的

4.2 主存储器

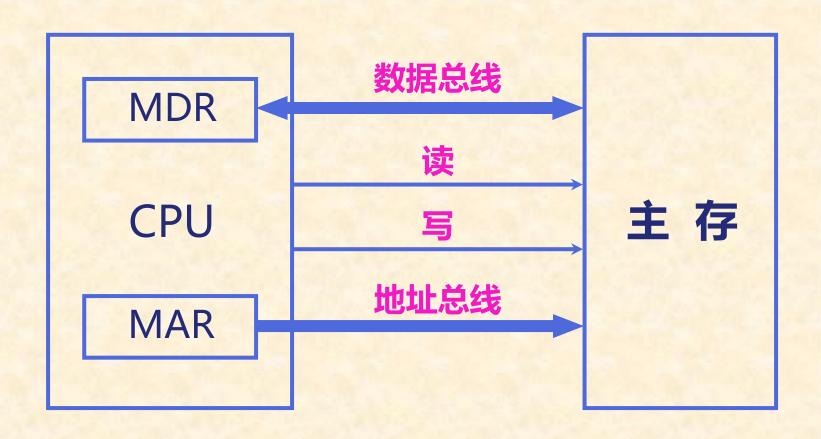
一、概述

1. 主存的基本组成



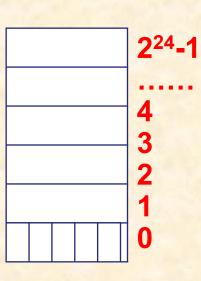
15

2. 主存和 CPU 的联系



3. 主存中存储单元地址的分配

设地址线 24 根 按字节 寻址 2²⁴ = 16 M 若字长为 16 位 按 字 寻址 8 M 若字长为 32 位 按 字 寻址 4 M



大端存储(Big

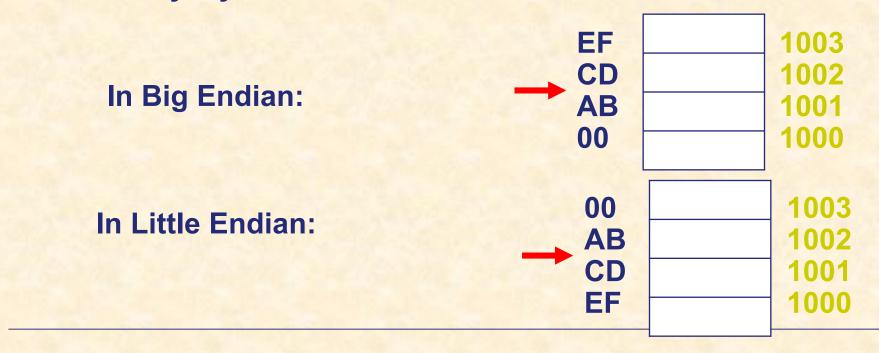
Endian : 将信息的 高位字节先送入存储 器的低位存储起来。 小端存储(Little Endian):将信息的低位字节先送入存储器的低位存储器的低位存储起来。

Big Endian versus Little Endian

Ex1: Memory layout of a number ABCDH located in 1000



Ex2: Memory layout of a number 00ABCDEFH located in 1000



Big Endian versus Little Endian

Ex3: Memory layout of a instruction located in 1000

假定**小端机器**中指令: mov AX, 0x12345(BX)

其中操作码mov为40H,寄存器AX和BX分别为0001B和0010B,立即数占32位,则存放顺序为:

MOV	AX BX	00012345H	
40 1 2	45 23 01	U I 1004	45 23
若在大端机器上,	则存放顺序	字如何? 23 1003 45 1002	01 00 12
40 1 2	00 01 23	12 1001 40 1000 地址	40

3. 主存中存储单元地址的分配

大端存储(Big Endian)

: 将信息的高位字节先送入存储器的低位存储起来。

3字节 2字节 1字节 0字节

字节册计

字地址		3-13	- TO-11	
0	0	1	2	3
4	4	5	6	7
8	8	9	10	11

高位字节 地址为字地址

小端存储(Little Endian)

: 将信息的低位字节先送入 存储器的低位存储起来。

3字节 2字	节 1字	芦	0字节
字地址	字节	地址	
0	1	0	
2	3	2	
4	5	4	

低位字节 地址为字地址

MSB:Most Significant Bit/Byte, 最高有效位/字节;

LSB:Least Significant Bit/Byte, 最低有效位/字节。

Alignment(对齐)

Alignment: 要求数据的地址是相应的边界地址

- 目前机器字长一般为32位或64位,而存储器地址按字节编址
- 指令系统支持对字节、半字、字及双字的运算,也有位处理指令
- 各种不同长度的数据存放时,有两种处理方式:
 - 按边界对齐 (假定字的宽度为32位,按字节编址)
 - 字地址: 4的倍数(低两位为0), 地址依次为: 0, 4, 8, 12······
 - 半字地址: 2的倍数(低位为0), 地址依次为:0, 2, 4, 8******
 - 字节地址: 任意, 地址依次为: 0, 1, 2, 3, 4 ·····
 - 不按边界对齐

坏处:可能会增加访存次数!

Alignment(对齐)

示例 假设数据顺序:字-半字-双字-字节-半字-.....

如: int i, short k, double x, char c, short j,......

		0字节	1字节	2字节	3字节
拉油田对文	00			i	
按边界对齐	04		/// k ////		blank
x: 2个周期	08	X	32318		eligi in a
	12				
j: 1个周期	16	C	blank		

则: &i=0; &k=4; &x=8; &c=16; &j=18;......

i的字地址要求边界对齐:只能取值为4的倍数(低两位为0),地址依次为: 0,4,8,12*****

K和j的半字地址要求边界对齐:只能取值2的倍数(低位为0),地址依次为:0,2,4,8******

x的double地址要求边界对齐:只能取值8的倍数(低三位为0),地址依次为: 0,8,16,24······

Alignment(对齐)

示例 假设数据顺序:字-半字-双字-字节-半字-.....

如: int i, short k, double x, char c, short j,.....

		0千巾	1十丁	2子丁	3千7
地計甲計文	00			i	
按边界对齐	04		/// <mark>k</mark> ////		blank
x: 2个周期	08	X			Ne
	12				
j: 1个周期	16	C	blank		

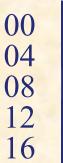
则: &i=0; &k=4; &x=8; &c=16; &j=18;......

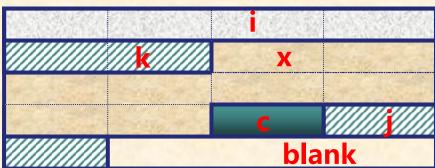
字节0 字节1 字节2 字节3

边界不对齐

x: 3个周期

j: 2个周期





增加了访存次数!

则: &i=0; &k=4; &x=6; &c=14; &j=15;......

4. 主存的性能指标

- (1) 存储容量 主存存放二进制代码的总数量
 - ·以字编址的计算机:字数与字长的乘积 存储容量=存储单元个数×存储字长

64K×16位: 64K个存储单元,每个存储单元字长为16位

•以字节编址的计算机,以字节数来表示

存储容量 = 存储单元个数×存储字长/8

64K×16位用字节数表示: 128K字节 (128KB)

4. 主存的技术指标

(2) 存储速度

- 存取时间T_A (Memory Access Time):从CPU送出内存单元的 地址码开始,到主存读出数据并送到CPU(或者是把CPU数据写入主存)所
 需要的时间(单位: ns, 1 ns = 10⁻⁹ s) 读出时间 写入时间
- 存取周期T_{MC} (Memory Cycle Time):连读两次访问存储器所需的最小时间间隔,它应等于存取时间加上下一存取开始前所要求的附加时间,因此,T_{MC}比T_A大(因为存储器由于读出放大器、驱动电路等都有一段稳定恢复时间,所以读出后不能立即进行下一次访问。)读周期 写周期

• $T_{MC} > T_A$,因为存储器在读写操作后,要有一段恢复内部状态的复原时间。对于破坏性读出的RAM,存取周期往往比存取时间要大的多,甚至可达到 $T_{MC} = 2T_A$,这是因为存储器中的信息读出后需要马上进行再生(重写)。

4. 主存的技术指标

(3) 存储器的带宽: 单位时间内存储器存取的信息量

单位:字/秒,字节/秒,位/秒

- 提高主存的带宽的措施
 - 缩短存取周期
 - 增加存储字长
 - 增加存储体

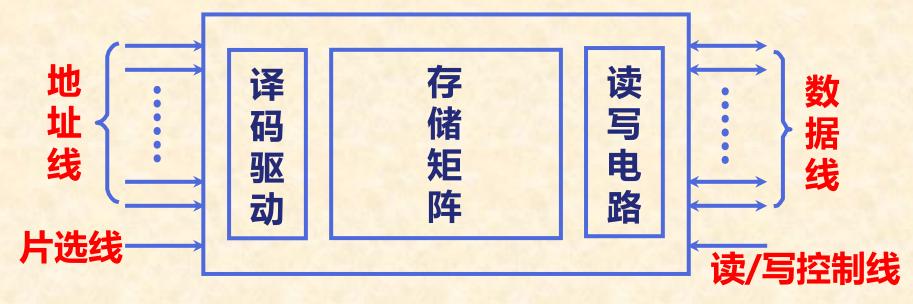
4. 主存的技术指标

- (4) 可靠性: 指在规定时间内, 存储器无故障读写的概率。
 - 平均无故障时间 (Mean Time Between Failures, MTBF) 用于衡量可靠性。
 - MTBF为两次故障之间的平均时间间隔。MTBF 越长,说明存储器可靠性越高。
 - (5) 功耗: 功耗既反映耗电又反映其发热程度。

二、半导体存储芯片简介



1. 半导体存储芯片的基本结构



地址线 (单向)	数据线 (双向)	芯片容量
10	4	1K × 4位
14	1	16K×1位
13	8	8K×8位

二、半导体存储芯片简介

1. 半导体存储芯片的基本结构



片选线 CS CE

读/写控制线 一线二用: WE (低电平写 高电平读)

一线一用: OE (允许读) WE (允许写)

存储芯片片选线的作用

- 由于一块集成芯片容量有限,要组成一个大的存储器一般需要将多个芯片连接使用。
- 在使用存储器某个地址时,某些芯片需要使用,其它芯片暂时不用,即片选。
- 当芯片的片选信号(CS)有效时,芯片被选中,此片所连的地址线才有效,才能对它进行读写操作

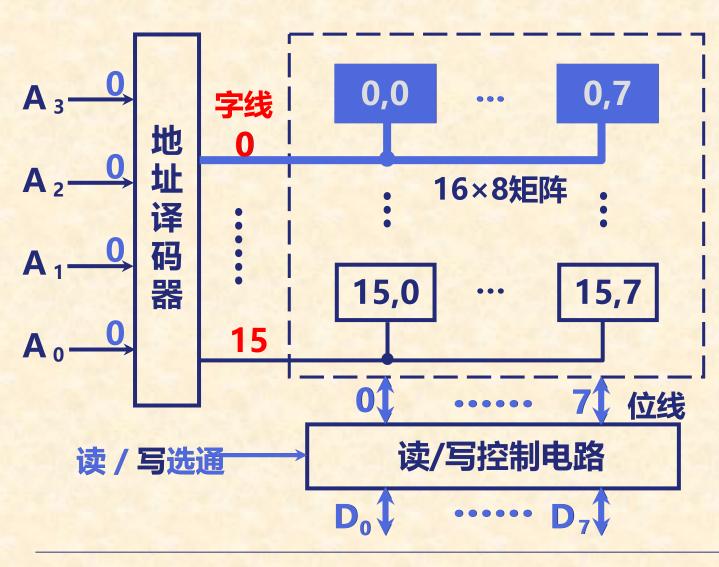
用 16K×1位的存储芯片组成 64K×8位的存储器 8片 8片 8片 8片 16K×1位 16K×1位 16K×1位 16K×1位

2. 半导体存储芯片的译码驱动方式

- 地址译码器设计方案
 - 线选法(单译码): 地址译码器只有一个,用一根字线直接选中一个存储单元的各位
 - 当地址线数较大,译码器复杂、开销大,使得存储器成本迅速增加,性能下降。
 - 重合法 (双译码): X地址译码器、Y地址译码器

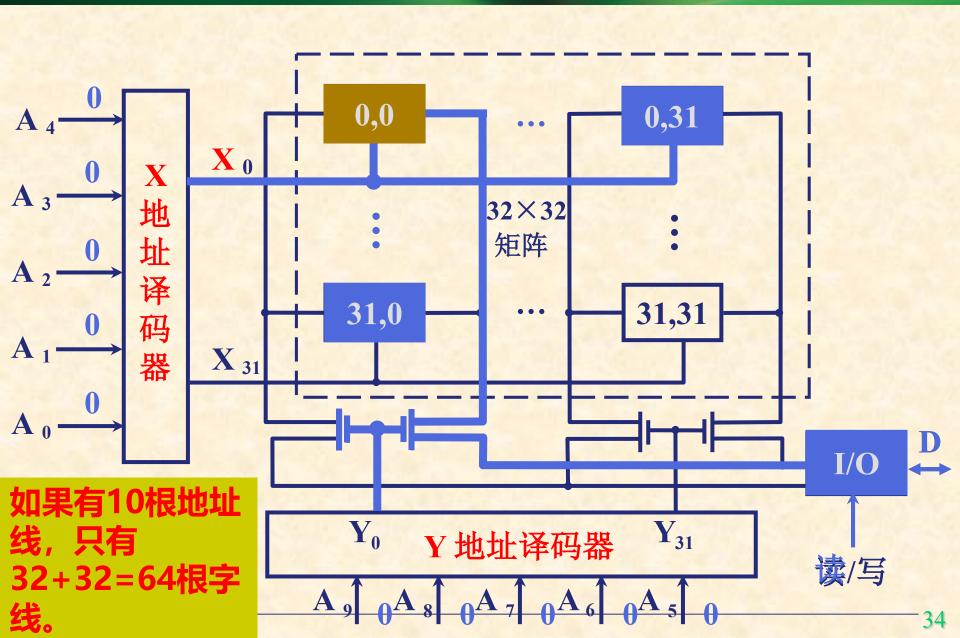
2. 半导体存储芯片的译码驱动方式

(1) 线选法

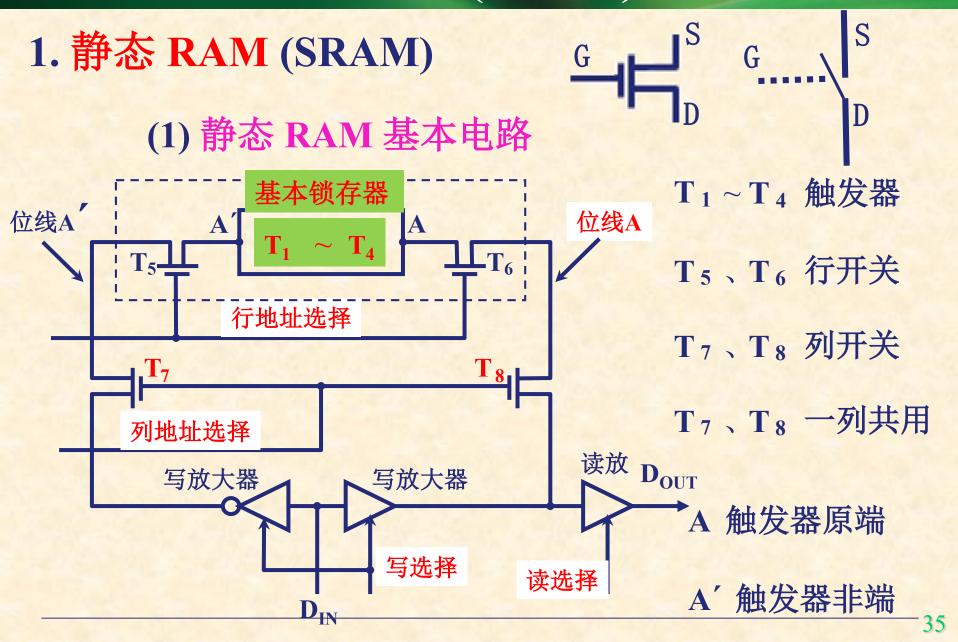


如果有10根地 址线,就有 1024根字线 ,字线的数量 比较大。

(2) 重合法

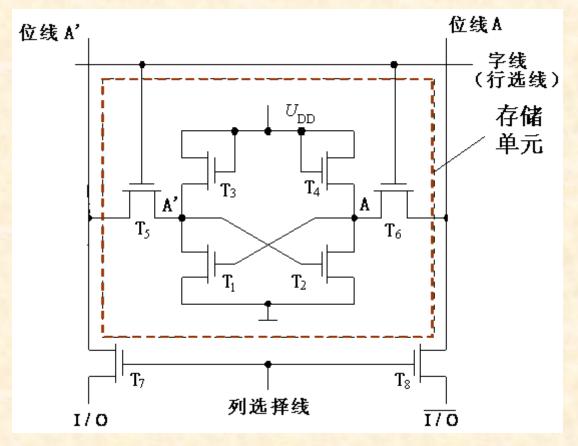


三、随机存取存储器(RAM)



六管静态MOS管电路

6管静态NMOS记忆单元



SRAM中数据保存在一对正负反馈门电路中,只要供电,数据就一直保持,不是破环性读出,也无需重写。即:无需刷新!

信息存储原理: 看作带时钟的RS触发器

写入时:

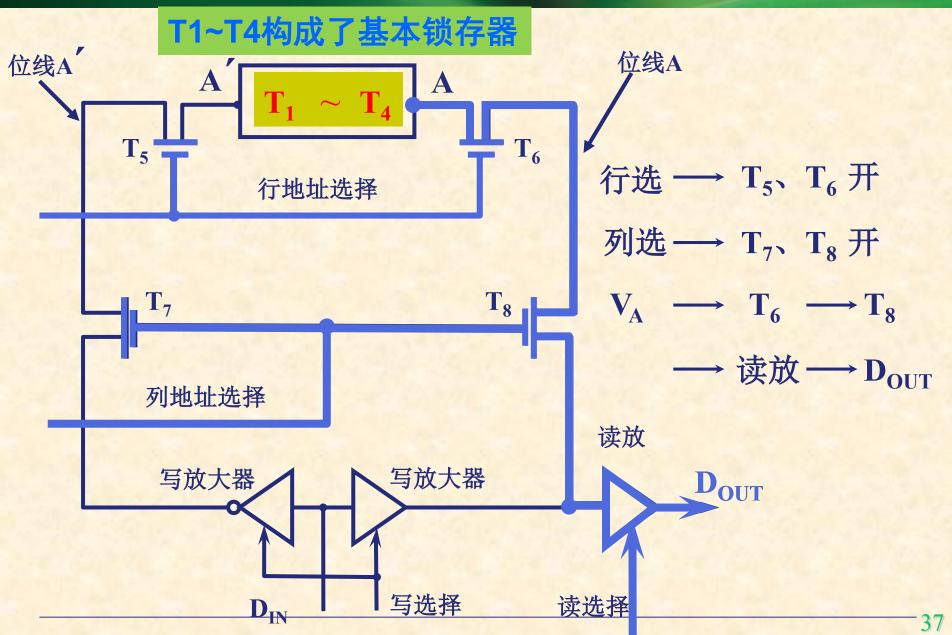
- 位线上是被写入的二进 位信息0或1
 - 置字线为1
- 存储单元(触发器)按位 线的状态设置成0或1

读出时:

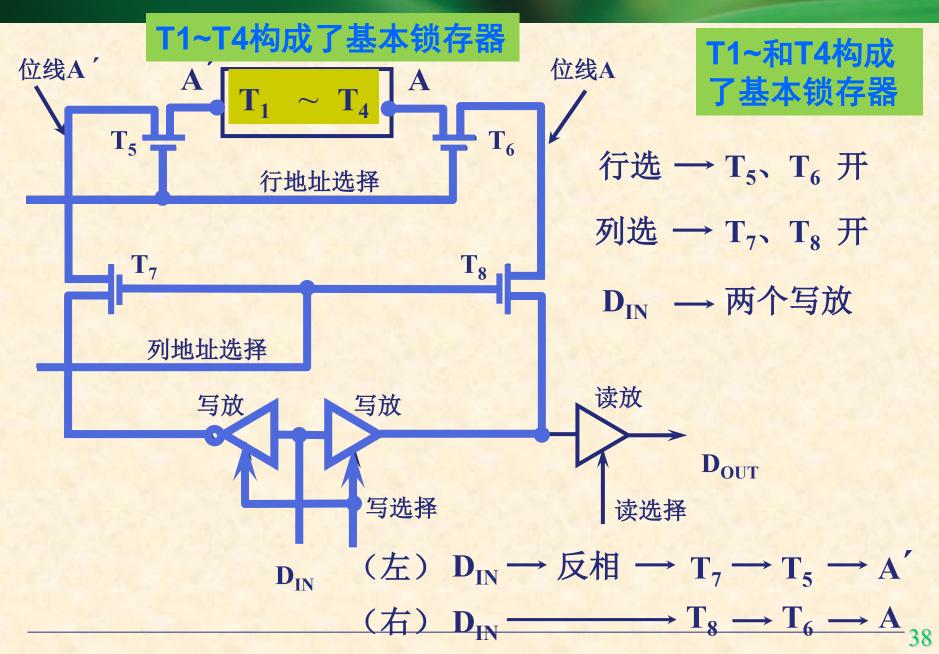
- 置2个位线为高电平
- 置字线为1
- 存储单元状态不同,位 线的输出不同

T1~T4构成了基本锁存器

①静态 RAM 基本电路的 读操作



② 静态 RAM 基本电路的 写 操作

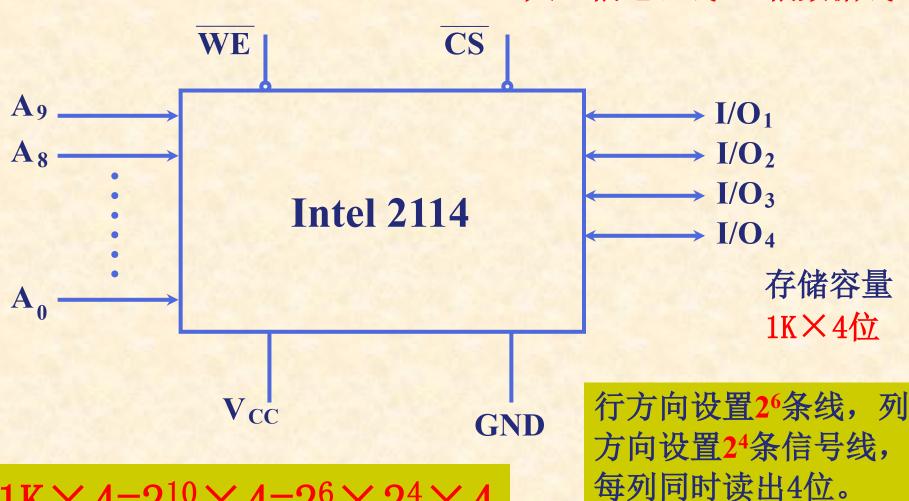


(2) 静态 RAM 芯片举例

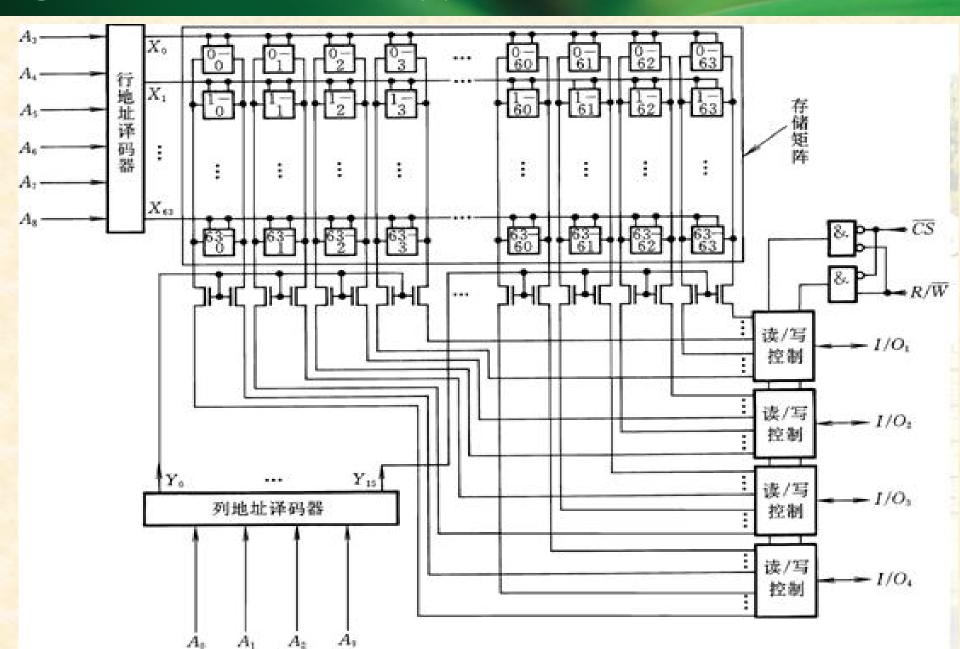
① Intel 2114 外特性 (1K×4)

 $1K \times 4 = 2^{10} \times 4 = 2^{6} \times 2^{4} \times 4$

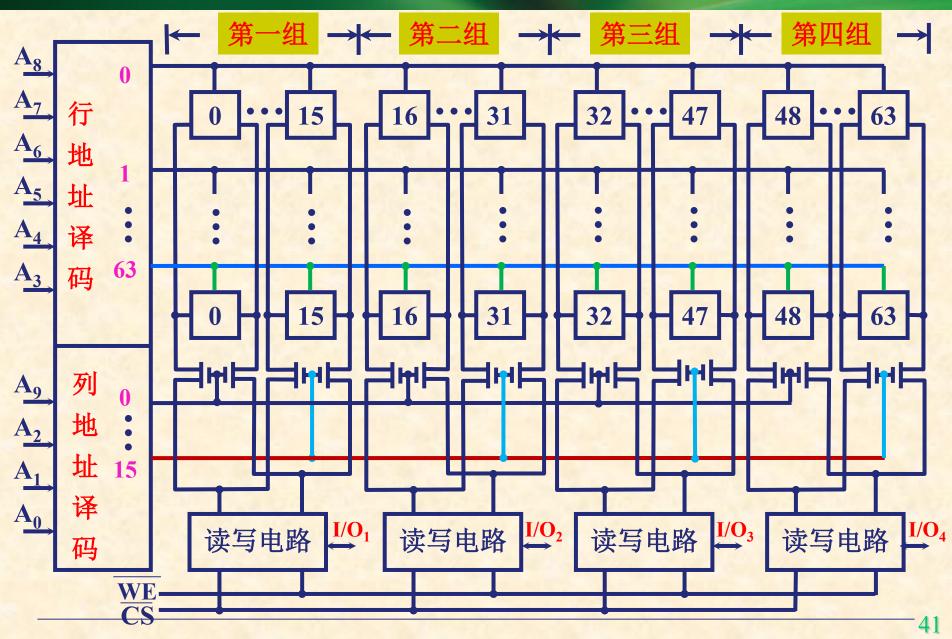
共10根地址线。4根数据线



② Intel 2114 RAM 矩阵

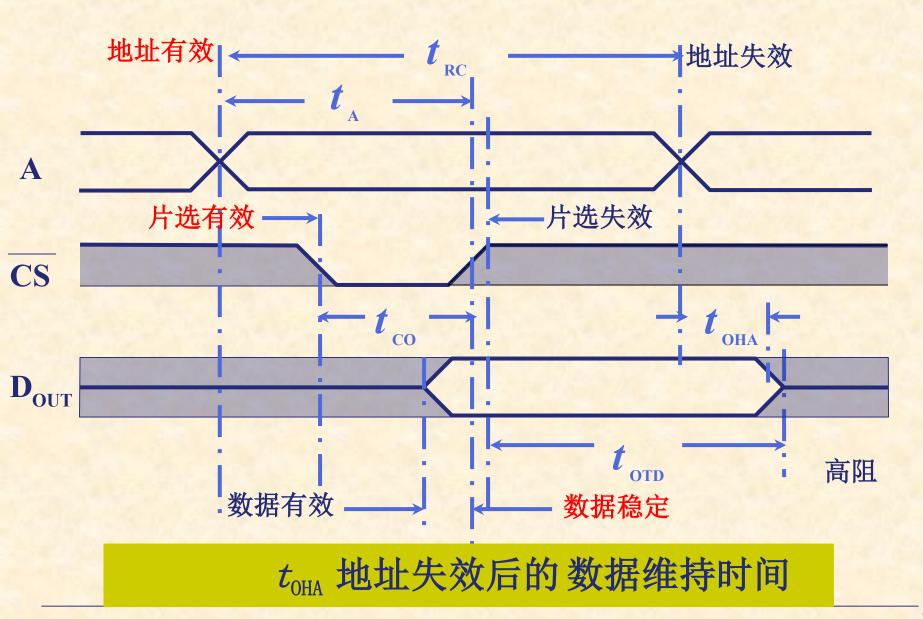


② Intel 2114 RAM 矩阵 读



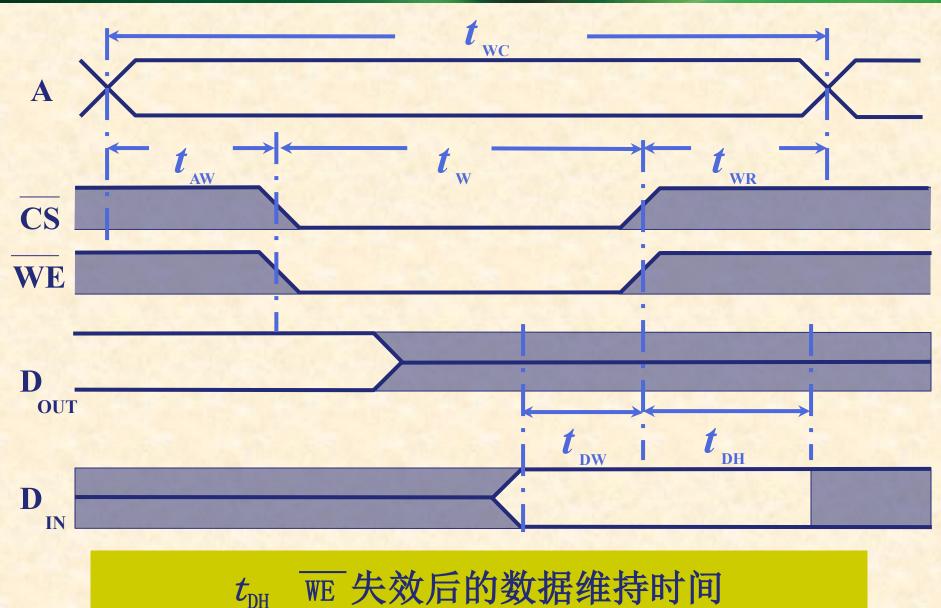
(3) 静态 RAM 读 时序





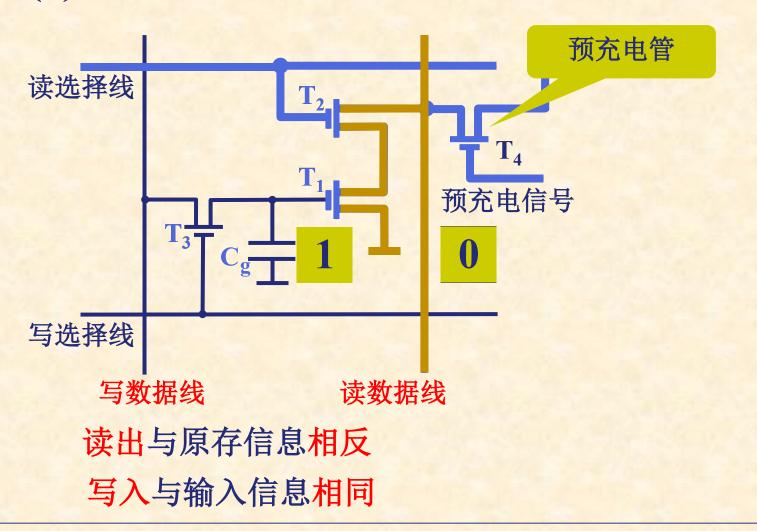
(4) 静态 RAM (2114) 写 时序





2. 动态 RAM (DRAM)

(1) 动态 RAM 基本单元电路

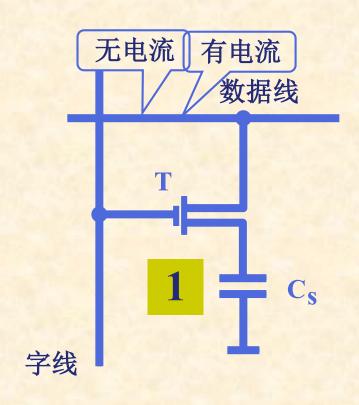


(1) 动态 RAM 基本单元电路

• 构造和表示:

信息记忆在电容C_s上, T为门控管, 控制数据的进出。其栅极接读/写选择线(字线),漏和源分别接数据线(位线)和记忆电容C_s。数据1或0以电容C_s上电荷量的有无来判别。

- 读写原理:在选择(字)线上加高电平,使T管导通。
 - •写"0"时,在数据线上加低电平,使C。上电荷对数据线放电;
 - •写"1"时,在数据线上加高电平,使数据线对C_s充电;
 - 读出时, 在数据线上有一读出电压。它与C₅上电荷量成正比。

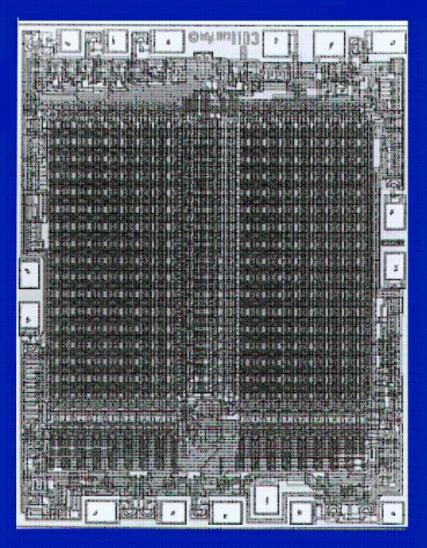


读出时数据线有电流 为 "1"

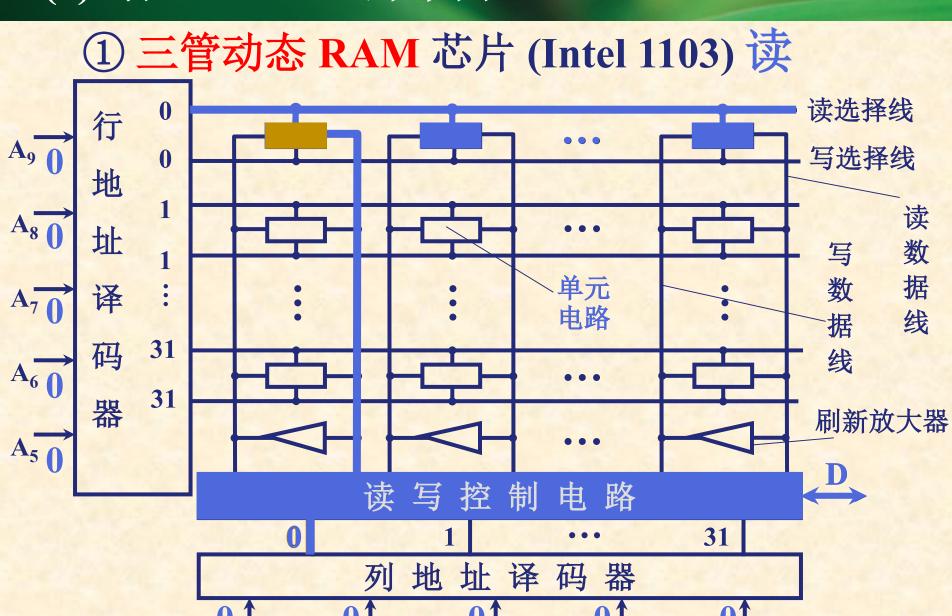
写入时Cs充电 为 "1" 放电 为" 0"

FIRST DRAM (1103 by Intel 1970)

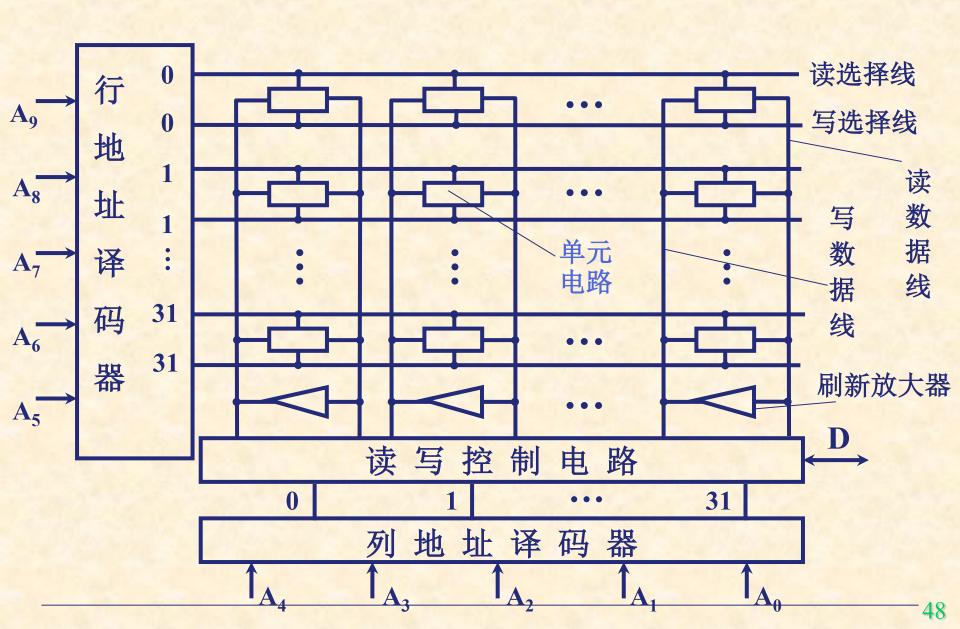
(5 V 8 μm 3 mm² 2600 μm²/cell)



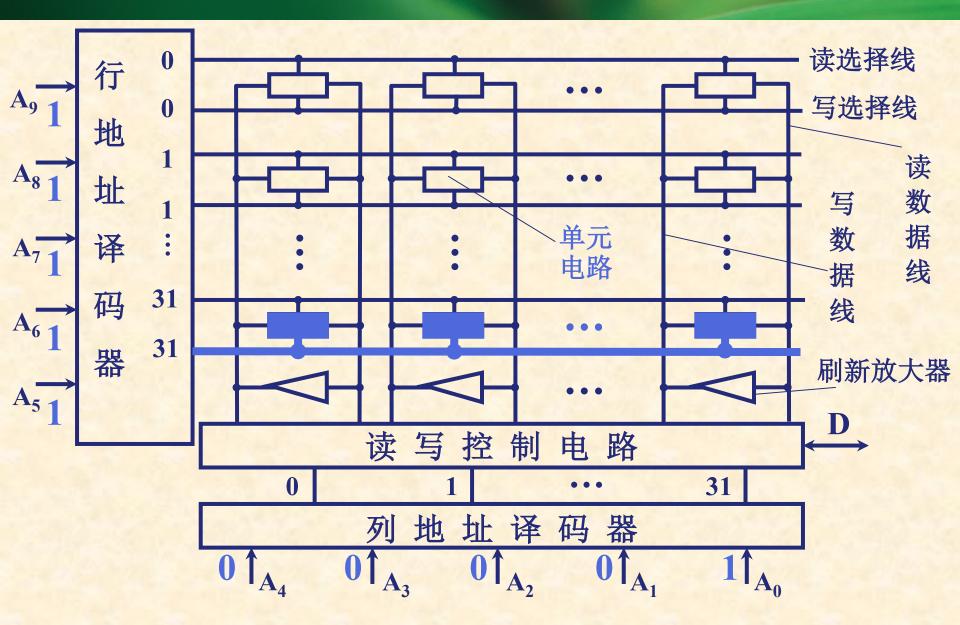
(2) 动态 RAM 芯片举例



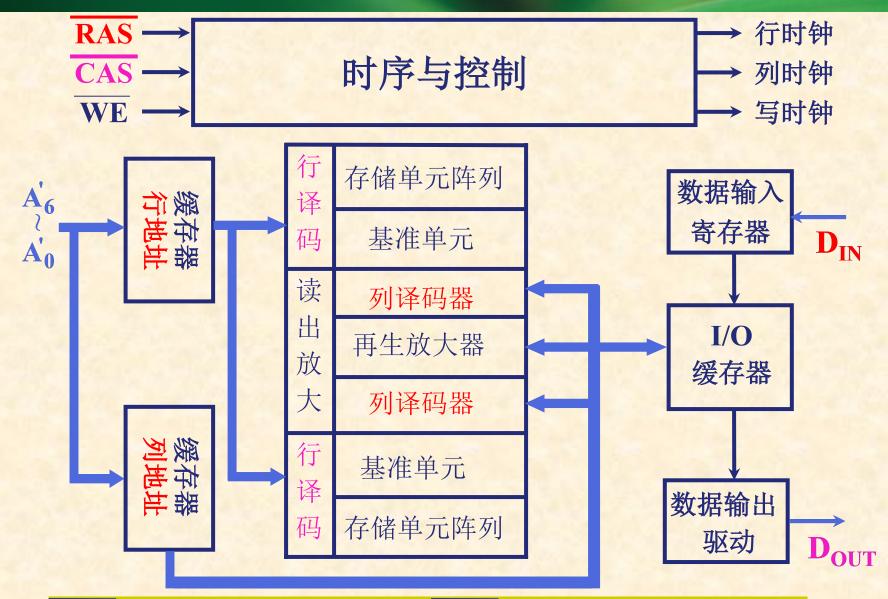
② 三管动态 RAM 芯片 (Intel 1103) 与



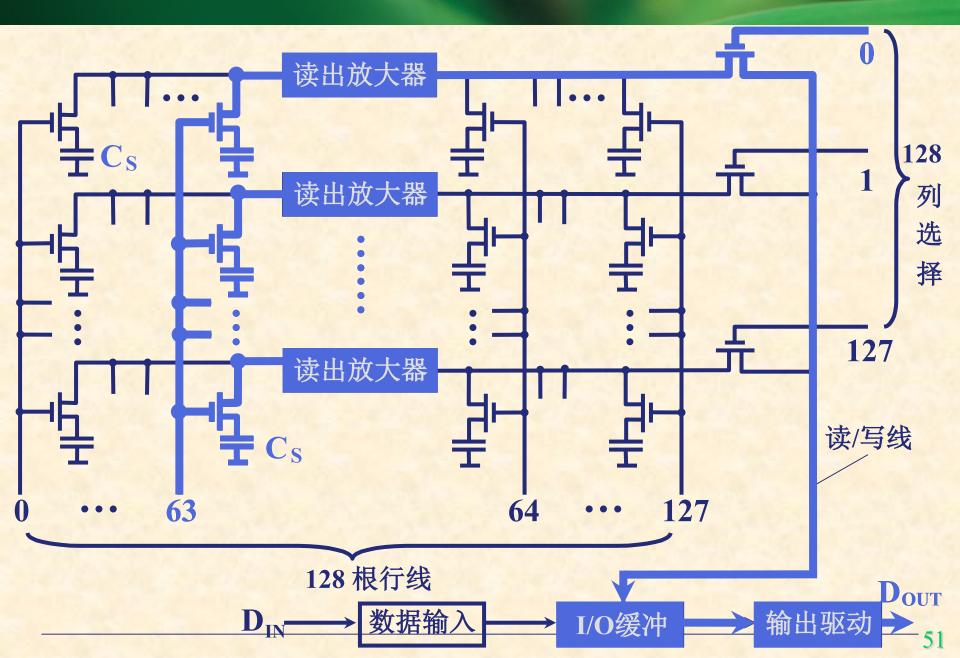
② 三管动态 RAM 芯片 (Intel 1103) 写



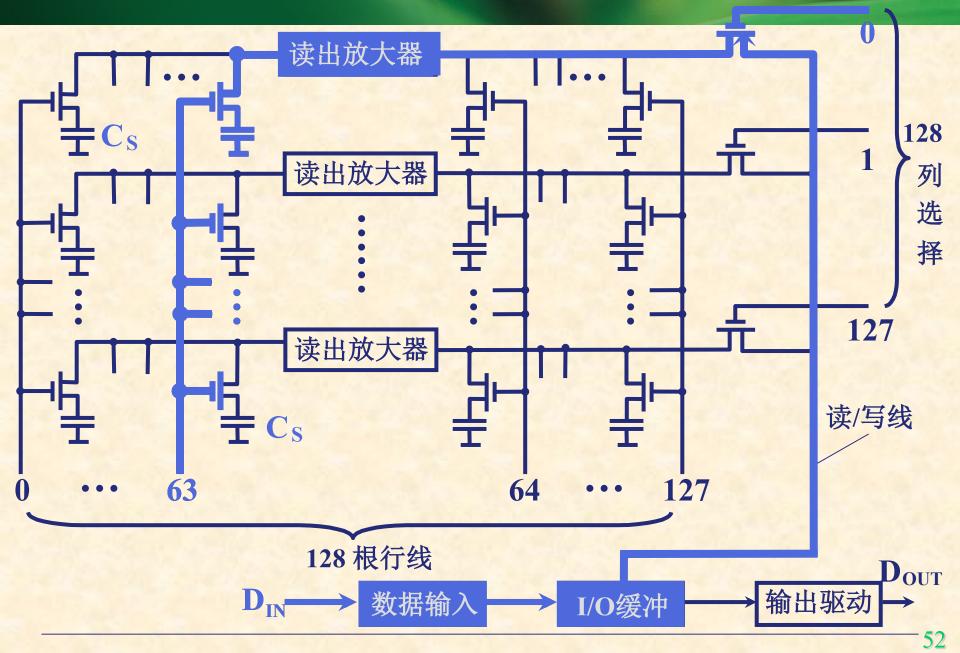
③ 单管动态 RAM 4116 (16K × 1位) 外特性



④ 4116 (16K × 1位) 芯片读原理



④ 4116 (16K × 1位) 芯片 写 原理



(3) 动态 RAM 时序

行、列地址分开传送

读时序

行地址 RAS 有效

写允许 WE 有效(高)

列地址 CAS 有效

数据 D_{out} 有效

写时序

行地址 RAS 有效

写允许 WE 有效(低)

数据 D_{IN} 有效

列地址 CAS 有效

(4) 动态 RAM 刷新

- 刷新是先将原存信息读出,再由刷新放大器 形成原信息并重新写入的再生过程。
 - 刷新过程对CPU是透明的
 - 刷新是按行进行的,每行中的记忆单元同时被刷新,故刷新操作时仅仅需要行地址,不需要列地址。
 - 刷新类似读操作,但有所不同。刷新操作只给 电容补充电荷,不需要信息输出。另外,刷新 不用加片选,整个存储器中所有芯片同时被刷 新。

(4) 动态 RAM 刷新

- 刷新周期:从上次对整个存储器刷新结束到下次对整个存储器全部刷新一遍为止的时间间隔,为电容数据有效保存期的上限(64ms)。
- 有三种刷新方式:集中式、分散式、异步刷新。
 - ① 集中刷新:

前一段时间正常读/写,后一段时间停止读/写,集中逐行刷新。

特点:集中刷新时间长,不能正常读/写(死区),很少使用。

② 分散刷新:

一个存储周期分为两段: 前一段用于正常读/写操作,后一段用于刷新操作。

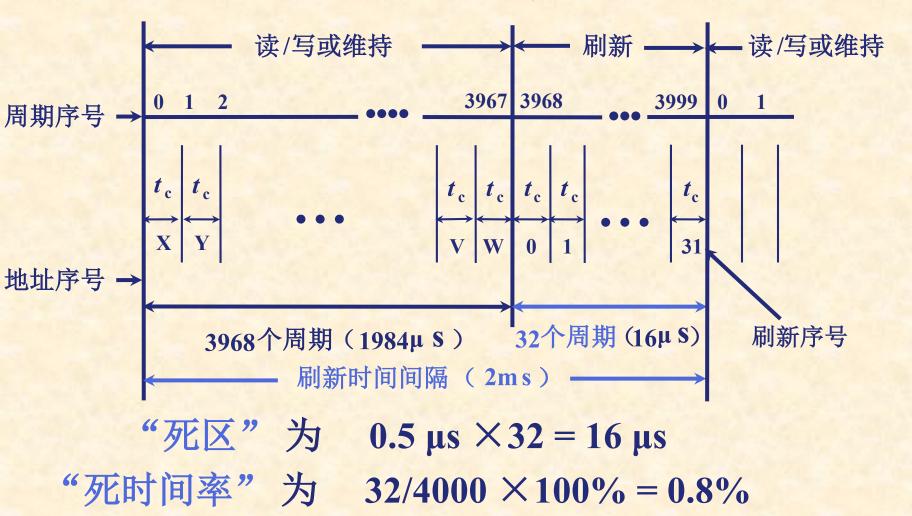
特点:不存在死区,但每个存储周期加长。很少使用。

③ 异步刷新:

结合上述两种方式。以4096行为例,在64ms时间内必须轮流对每一行刷新一次,即每隔64ms/4096=15.625µs刷新一行。

特点:结合前两种,效率高,用得较多。

①集中刷新(存取周期为0.5µs)以32×32矩阵为例



②分散刷新(存取周期为1µs)

以 128×128 矩阵为例 每次读写后,刷新一次



$$t_{\rm C} = t_{\rm M} + t_{\rm R}$$
 无 "死区" \downarrow 太宗 即第 (左取 思期)

(存取周期为 0.5 μs + 0.5 μs)

③ 异步刷新

• 是分散刷新与集中刷新的结合

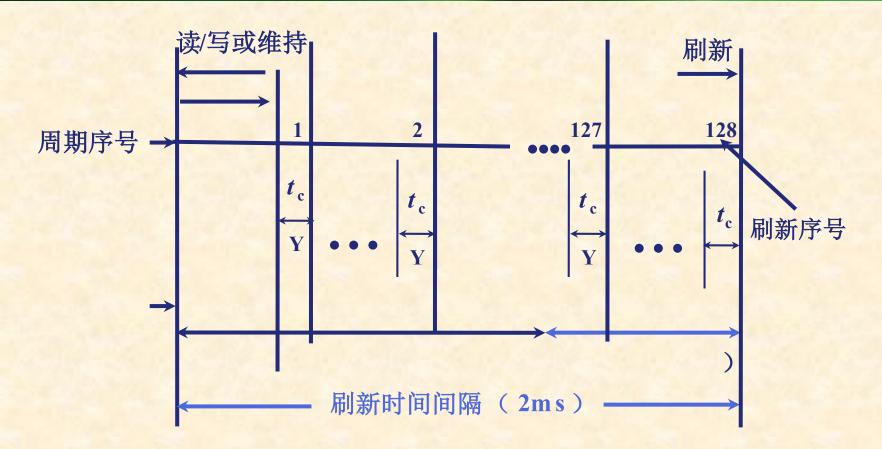
对于 128×128 的存储芯片 (存取周期为 0.5μs)

若每隔 2 ms 集中刷新一次 "死区" 为 64 μs 若异步刷新,将集中刷新里的128行刷新任务分到 128块内,每块内刷新一行,则每隔 15.6 μs 刷新一行(2000 μs ÷ 128= 15.6 μs)。

而且每行每隔 2 ms 刷新一次 "死区" 为 0.5 μs

将刷新安排在指令译码阶段,不会出现"死区"

③ 异步刷新



3. 动态 RAM 和静态 RAM 的比较

4.2

主存	DRAM	SRAM
存储原理	电容	触发器 缓存
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

四、只读存储器 (ROM)



1. 掩膜 ROM (MROM)

- •以元件的"有/无"表示"1"、"0"。
- 内容由半导体制造厂按用户提出的要求在芯片的生产过程中直接写入的,写入后内容无法改变。

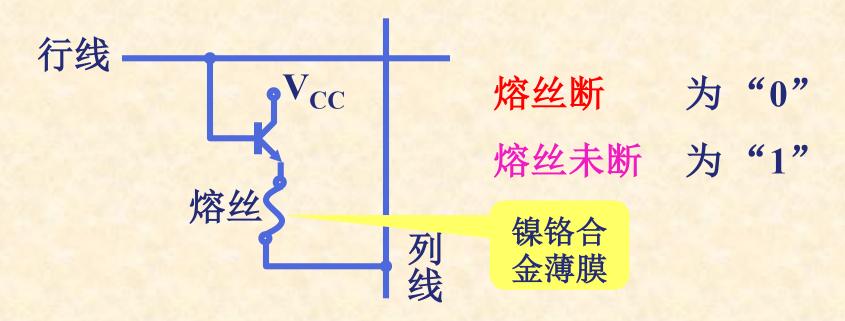
行列选择线交叉处有 MOS 管为"1"

行列选择线交叉处无 MOS 管为 "0"

- 优点:可靠性高,形成批量之后价格便宜;
- •缺点:用户对制造厂的依赖性过大,灵活性差。

2. PROM (一次性编程)

允许用户利用专门的设备(编程器)写入自己的程序, 但一旦写入后,其内容将无法改变。



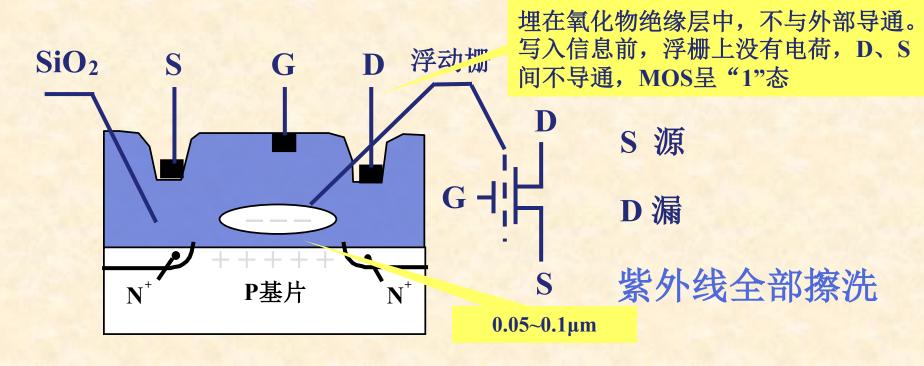
- 产品出厂时,所有记忆单元均制成"0"(或制成"1"),用户根据需要可自行将其中某些记忆单元改为"1"(或改为"0")。
- · 双极型PROM有两种结构,一种是熔丝烧断型,另一种是PN结击穿型,

3. EPROM (多次性编程)

浮动栅雪崩注入型MOS管

(1) N型沟道浮动栅 MOS 电路

25V(写入电压)、 50ms宽正脉冲



D端加正电压

D端不加正电压

形成浮动栅

不形成浮动栅

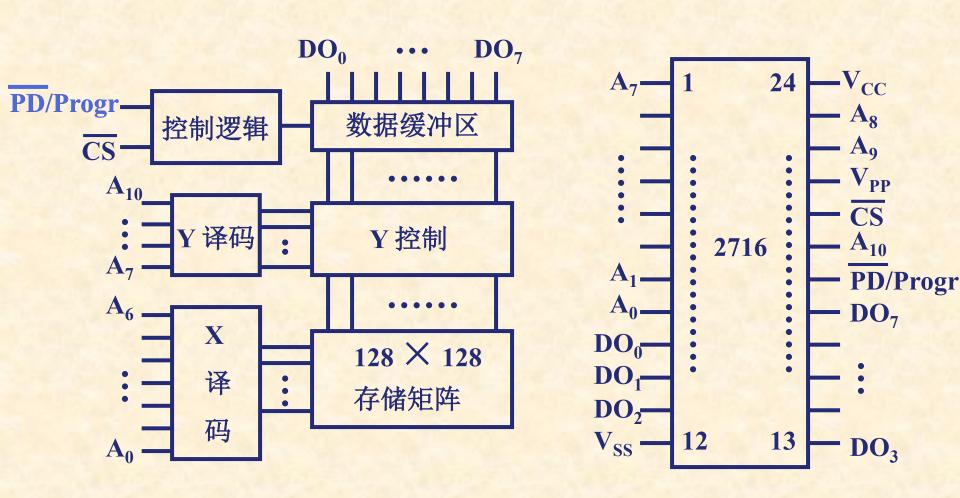
S与D不导通为"0"

S与D导通为"1"

3. EPROM (多次性编程)

- 可由用户利用编程器写入信息,可以多次改写。
- EPROM出厂时,存储内容为全"1",用户可以根据需要将其中某些记忆单元改为"0"。当需要更新存储内容时可以将原存储内容擦除(恢复全"1"),以便再写入新的内容。
- EPROM可分为: 紫外线擦除和电擦除
 - UVEPROM: 用紫外线灯制作的擦除器照射存储器芯片上的透明窗口, 使芯片中原存内容被擦除。
 - 由于是用紫外线灯进行擦除,所以只能对整个芯片擦除,而不能对芯片中个别需要改写的存储单元单独擦除。
 - 另外,为了防止存储的信息受日光中紫外线成分的作用而缓慢丢失,在 UVEPROM芯片写入完成后,必须用不透明的黑纸将芯片上的透明窗口封住。
 - EEPROM是采用电气方法来进行擦除的。
 - 在联机条件下既可以用字擦除方式擦除,也可以用数据块擦除方式擦除。以字擦除方式操作时,能够只擦除被选中的那个存储单元的内容;在数据块擦除操作时,可擦除数据块所有单元的内容。

(2) 2716 EPROM 的逻辑图和引脚



PD/Progr 功率下降 / 编程输入端 读出时为低电平

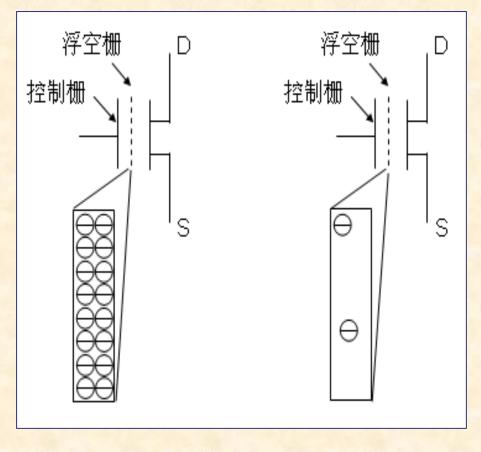


- 原因
 - ①EPROM的编程次数(寿命)是有限的: EEPROM因为氧化层被磨损,重复改写次数一般10万次
 - ②写入时间过长,即使对于EEPROM,擦除一个字节大约需要10ms,写入一个字节大约需要10us,比SRAM或DRAM的时间长100-1000倍。

4. Flash Memory (快擦型存储器)

- 闪速存储器是20世纪80年代中期出现的一种快擦写型存储器,它的主要特点是既可在不加电的情况下长期保存信息,又能在线进行快速擦除与重写,兼备了EEPROM和RAM的优点。
- 目前,大多数微型计算机的主板采用闪速存储器来存储BIOS程序,由于BIOS的数据和程序非常重要,不允许修改,故早期主板BIOS芯片多采用PROM或EPROM。

闪存 (Flash Memory)

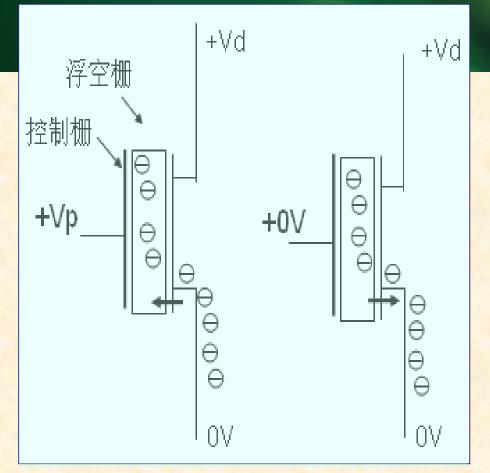


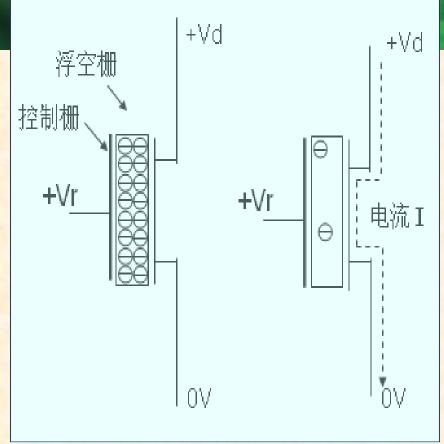
(a)"0"状态

(b) "1"状态

控制栅加足够正电压时,浮空栅储存大量负电荷,为"0"态;控制栅不加正电压时,浮空栅少带或不带负电荷,为"1"态

闪存的读取速度与DRAM 相近,是磁盘的100倍左右 ;写数据(快擦一编程) 则与硬盘相近





(a) 编程:写"0" (b) 擦除:写"1"

(a) 读"0"

(b) 读"1"

有三种操作:擦除、编程、读取

读快、写慢!

写入: 快擦(所有单元为1) - 编程(需要之处写0)

读出:控制栅加正电压,若状态为0,则读出电路检测不到电流;若状

态为1,则能检测到电流。