

第八章 半导体存储器和可编程逻辑器件

8.1 半导体存储器

8.2 可编程逻辑器件PLD



8.1 半导体存储器

半导体存储器的种类很多，按采用元件来分，有双极型和MOS型两大类。

双极型存储器以双极型触发器为基本存储单元，其工作速度快，但功耗大，主要用于对速度要求高的场合；MOS型存储器以MOS触发器或电荷存储结构为存储单元，它具有集成度高、功耗小、工艺简单等特点，主要用于大容量存储系统中。目前数字系统中主要选用MOS型存储器。按存取信息方式划分，有只读存储器ROM(Read Only Memory)和随机存取存储器RAM(Random Access Memory)两大类。

只读存储器ROM在正常工作时只能读出信息，而不能写入信息。ROM的信息是在制造时用专门的写入装置写入的并可长期保留，即断电后器件中的信息不会消失，因此也称为非易失性存储器。ROM又可分为掩膜ROM、可编程ROM(Programmable Read Only Memory, PROM)和可擦除的可编程ROM(Erasable Programmable Read Only Memory, EPROM)。

随机存取存储器RAM正常工作时可以随时写入或读出信息，但断电后器件中的信息也随之消失，因此也称为易失性存储器。RAM又可分为静态存储器SRAM(Static Random Access Memory)和动态存储器DRAM(Dynamic Random Access Memory)两类。DRAM的存储单元结构非常简单，其集成度远高于SRAM，但它的存取速度不如SRAM快。

存储器的存储容量和存取时间是反映系统性能的两个重要指标。存储容量指存储器所能存放的信息的多少，存储容量越大，说明存储的信息越多，系统的功能越强。存储器的容量一般用字数 N 同字长 M 的乘积即 $N \times M$ 来表示，如 $1K \times 8$ 表明该存储器有1024个存储单元，每一单元存放八位二进制信息。存取时间一般用读/写周期来描述。读/写周期越短，即存取时间越短，存储器的工作速度就越高。

8.1.1 只读存储器(ROM)

ROM的一般结构如图8-1所示。它主要由地址译码器、存储矩阵及输出缓冲器组成。

存储矩阵是存放信息的主体，它由许多存储单元排列组成。每个存储单元存放一位二进制代码(0或1)，若干个存储单元组成一个字。地址译码器有几条地址输入线($A_0 \sim A_{n-1}$)，则译码器输出线有 2^n 条($W_0 \sim W_{2^n-1}$)，每一条译码输出线 W_i 称为“字线”，它与存储矩阵中的一个“字”相对应。因此，每当给定一组输入地址时，译码器只有一条输出字线 W_i 被选中，即 $W_i=1$ ，该字线可以在存储矩阵中找到一个相应的字，并将字中的 m 位信息 $D_{m-1} \sim D_0$ 送至输出缓冲级。此时三态控制端使缓冲级工作，读出 $D_{m-1} \sim D_0$ 的数据。每条输出线 D_i 又称为“位线”，每个字中信息的位数称为“字长”。

ROM的存储单元可以用二极管构成，也可用双极型三极管或MOS管构成。

输出缓冲级是ROM的数据读出电路，通常用三态门构成，它不仅可以实现对输出数据的三态控制，以便与系统总线连接，还可提高存储器的负载能力。🔥

图8—2是具有两位地址输入和四位数据输出的ROM结构图，其存储单元用二极管构成。图中 $W_0 \sim W_3$ 四条字线分别选择存储矩阵中的四个字，每个字存放四位信息。制作时，若在某个字的某一位存入“1”，则在该字的字线 W_i 与位线 D_j 之间接入二极管；反之，就不接二极管。

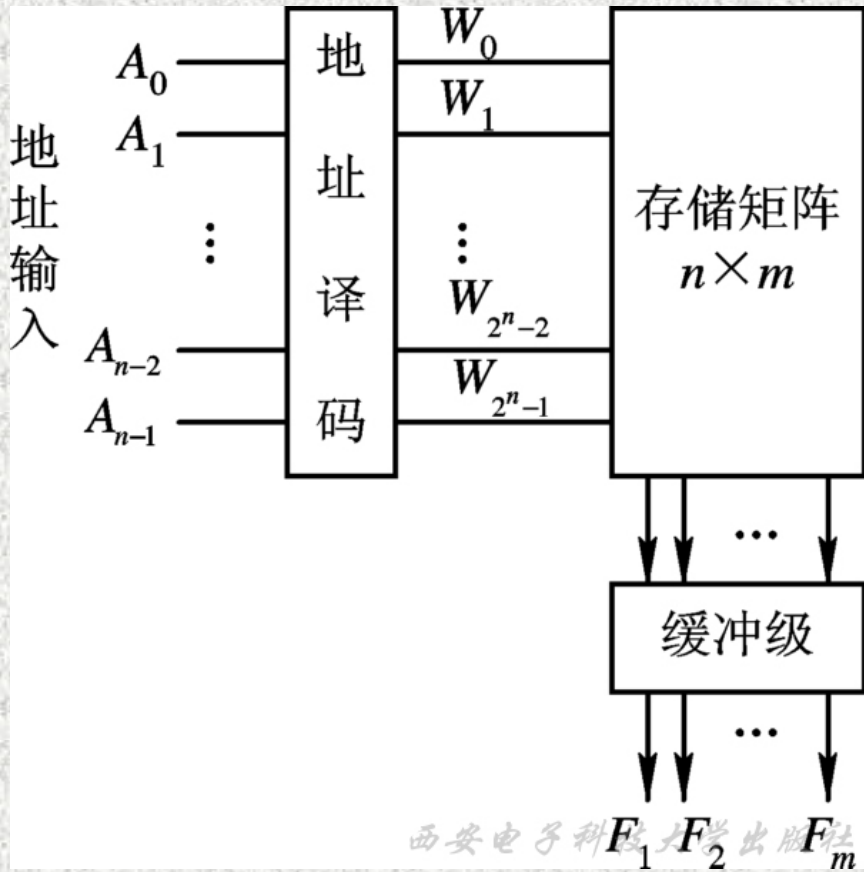


图8—1 n 字 m 位ROM结构

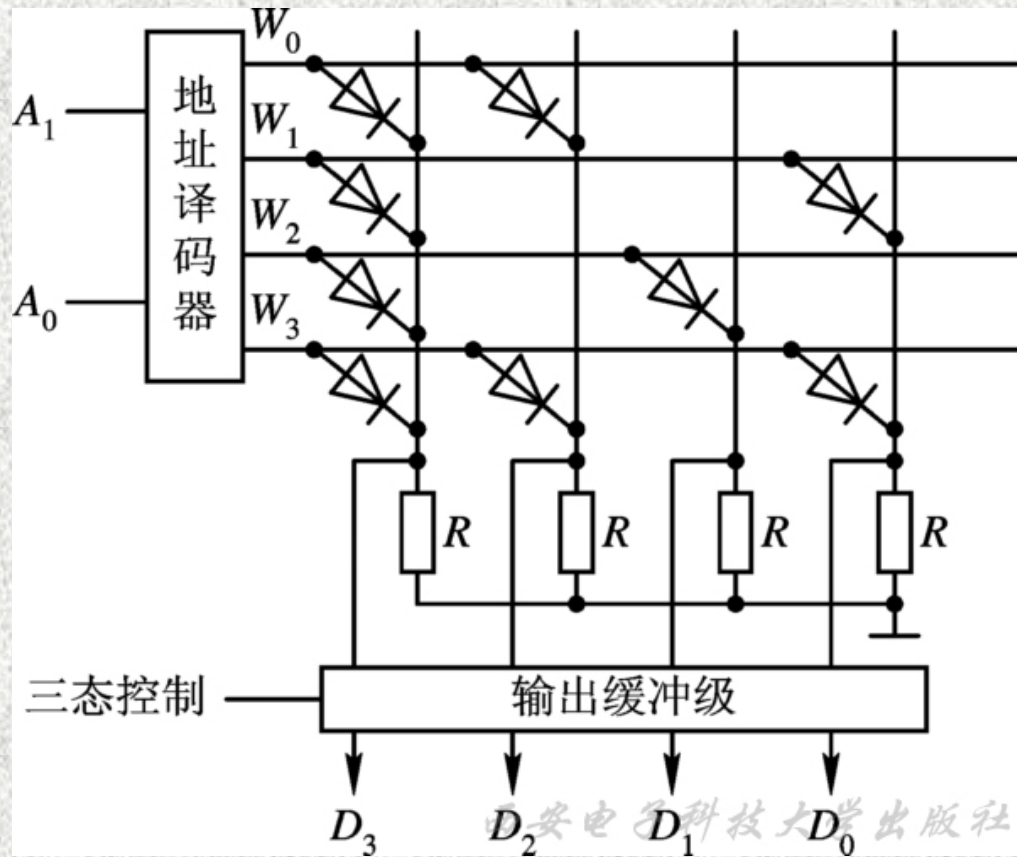


图8-2 二极管ROM结构图

读出数据时，首先输入地址码，同时输出缓冲级三态控制端，使输出缓冲级工作，则在数据输出端 $D_3\sim D_0$ 可以获得该地址对应字中所存储的数据。例如，当 $A_1A_0=00$ 时， $W_0=1, W_1=W_2=W_3=0$ ，所以 W_0 被选中，读出 W_0 对应字中的数据 $D_3D_2D_1D_0=1100$ 。当 A_1A_0 为01、10、11时，依次读出的数据是1001、1010、1101。该ROM存储的数据如表8-1所示。

。

表8—1 图8—2ROM的数据表

A_1	A_0	D_3	D_2	D_1	D_0
0	0	1	1	0	0
0	1	1	0	0	1
1	0	1	0	1	0
1	1	1	1	0	1

8.1.2 ROM在组合逻辑设计中的应用

用ROM实现组合逻辑的基本原理可从“存储器”和“与或逻辑网络”两个角度来理解。🔥

从存储器的角度看，只要把逻辑函数的真值表事先存入ROM，便可用ROM实现该函数。例如，在表8-1中，将输入地址 A_1A_0 视为输入变量，而将 D_3 、 D_2 、 D_1 、 D_0 视为一组输出逻辑变量，则 D_3 、 D_2 、 D_1 、 D_0 就是 A_1 、 A_0 的一组逻辑函数：

$$D_3 = \overline{A_1}\overline{A_0} + \overline{A_1}A_0 + A_1\overline{A_0} + A_1A_0 = m_0 + m_1 + m_2 + m_3$$

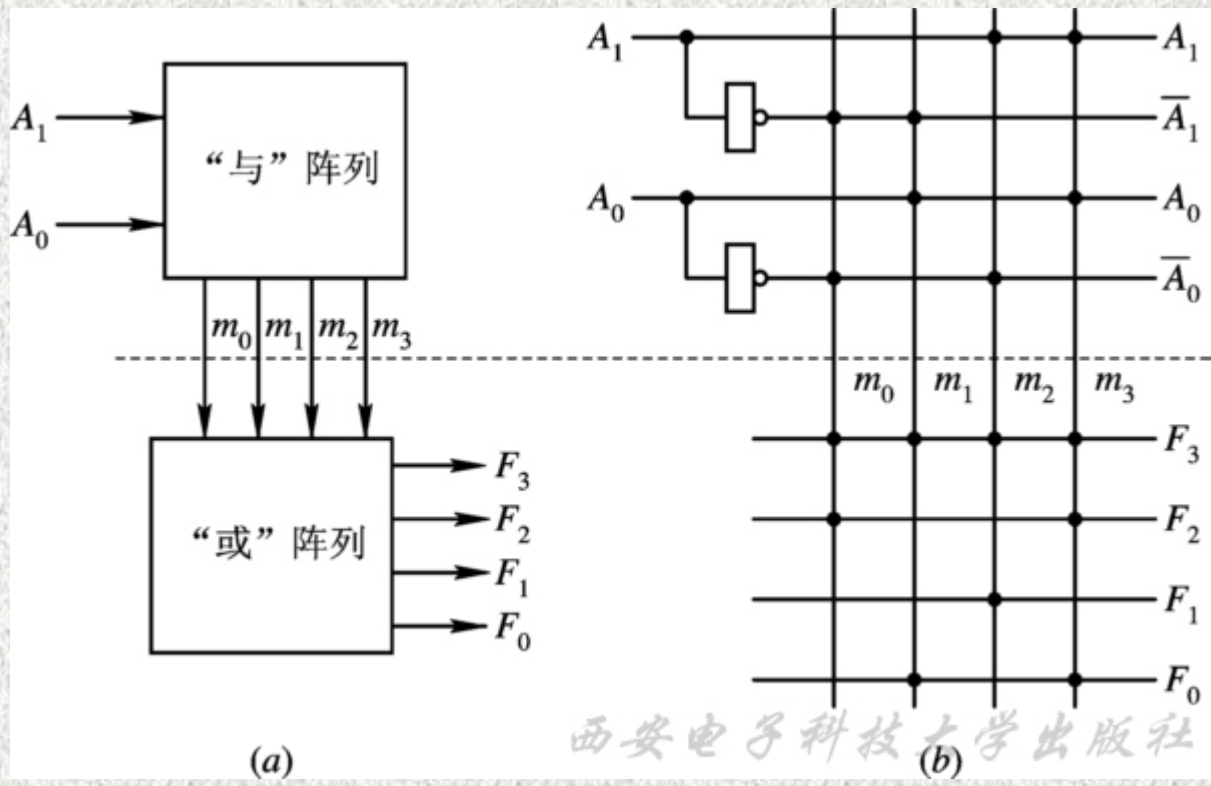
$$D_2 = \overline{A_1}\overline{A_0} + A_1A_0 = m_0 + m_3$$

$$D_1 = A_1\overline{A_0} = m_2$$

$$D_0 = \overline{A_1}A_0 + A_1A_0 = m_1 + m_3$$

可见，用ROM实现组合逻辑函数时，将逻辑函数的输入变量作为ROM的地址输入，将每组输出对应的函数值作为数据写入相应的存储单元中即可，这样按地址读出的数据便是相应的函数值。🔥

从与或逻辑网络的角度看，ROM中的地址译码器形成了输入变量的所有最小项，即实现了逻辑变量的“与”运算。ROM中的存储矩阵实现了最小项的“或”运算，即形成了各个逻辑函数，如上所述。基于这一分析，可以把ROM看做是一个与或阵列，如图8—3所示，其中(a)为ROM的框图，(b)为ROM的符号矩阵图。在图8—3(b)中，与阵列中的小圆点表示各逻辑变量之间的“与”运算，或阵列中的小圆点表示各最小项之间的“或”运算。该图中的小圆点是根据逻辑表达式画出的，因而它就是图8—2所示ROM的符号矩阵图。



西安电子科技大学出版社

图8—3 ROM的与或阵列图

(a)框图；(b)符号矩阵

由上可知，用ROM实现逻辑函数时，需列出它的真值表或最小项表达式，然后画出ROM的符号矩阵图。工厂根据用户提供的符号矩阵图，便可生产出所需的ROM。利用ROM不仅可实现逻辑函数(特别是多输出函数)，而且可以用作序列信号发生器和字符发生器，以及存放各种数学函数表(如快速乘法表、指数表、对数表及三角函数表等)。下面举例说明这些应用。

用ROM实现逻辑函数一般按以下步骤进行：❖

(1)根据逻辑函数的输入、输出变量数，确定ROM容量，选择合适的ROM。❖

(2)写出逻辑函数的最小项表达式，画出ROM阵列图。

(3)根据阵列图对ROM进行编程。❖

【例8-1】用ROM实现四位二进制码到格雷码的转换。



解 (1)输入是四位二进制码 $B_3 \sim B_0$ ，输出是四位格雷码，故选用容量为 $2^4 \times 4$ 的ROM。

(2) 列出四位二进制码转换为格雷码的真值表，如表8-2所示。由表可写出下列最小项表达式：

$$G_3 = \sum(8,9,10,11,12,13,14,15)$$

$$G_2 = \sum(4,5,6,7,8,9,10,11)$$

$$G_1 = \sum(2,3,4,5,10,11,12,13)$$

$$G_0 = \sum(1,2,5,6,9,10,13,14)$$

(3)画出四位二进制码转换为格雷码的转换器的ROM符号矩阵，如图8-4所示。

表8-2

四位二进制码转换为格雷码的真值表

二进制数 (存储地址)				格雷码 (存放数据)			
B_3	B_2	B_1	B_0	G_3	G_2	G_1	G_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0

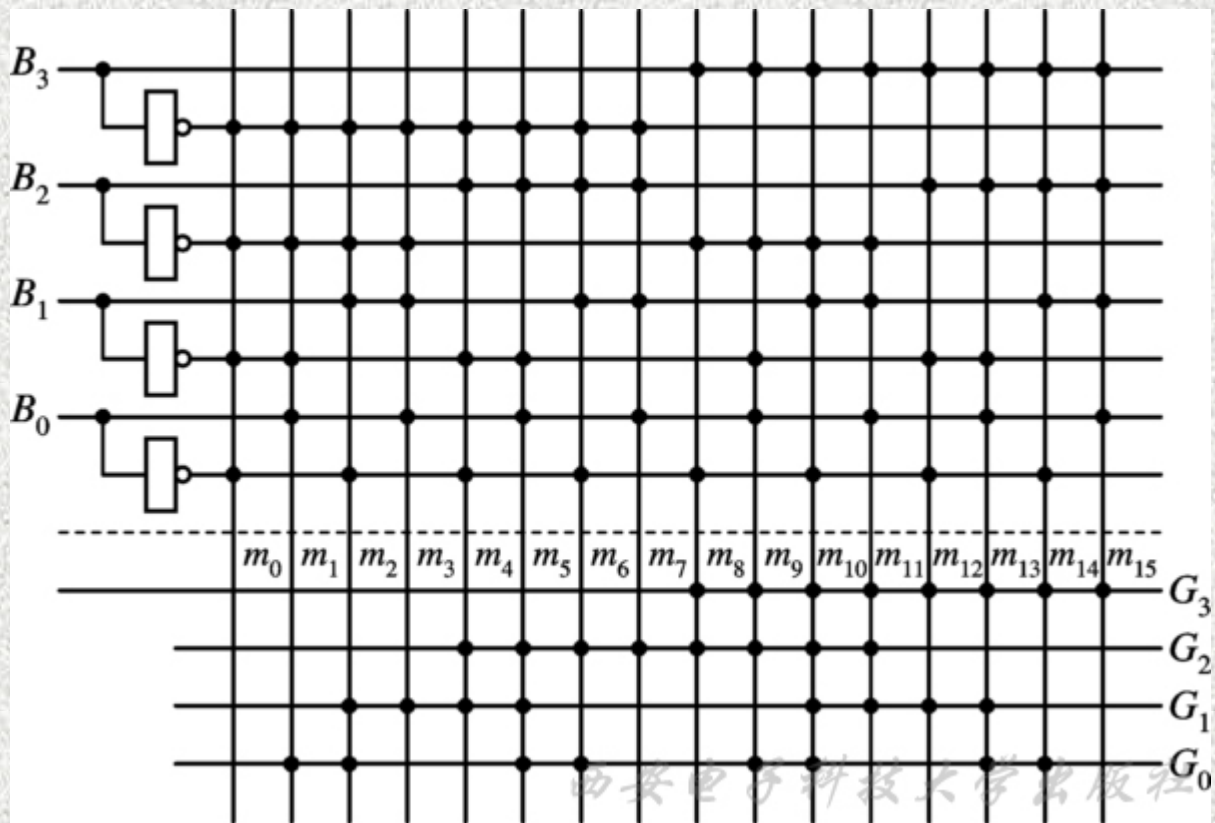


图8—4 四位二进制码转换为四位格雷码阵列图

8.1.3 ROM的分类❖

ROM的编程是指将信息存入ROM的过程。根据编程和擦除的方法不同，ROM可分为掩膜ROM、可编程ROM(PROM)和可擦除的可编程ROM(EPROM)三种类型。

1.掩膜ROM❖

掩膜ROM中存放的信息是由生产厂家采用掩膜工艺专门为用户制作的，这种ROM出厂时其内部存储的信息就已经“固化”在里边了，所以也称固定ROM。它在使用时只能读出，不能写入，因此通常只用来存放固定数据、固定程序和函数表等。

2.可编程ROM(PROM)🔥

PROM在出厂时，存储的内容为全0(或全1)，用户根据需要，可将某些单元改写为1(或0)。这种ROM采用熔丝或PN结击穿的方法编程，由于熔丝烧断或PN结击穿后不能再恢复，因此PROM只能改写一次。🔥

熔丝型PROM的存储矩阵中，每个存储单元都接有一个存储管，但每个存储管的一个电极都通过一根易熔的金属丝接到相应的位线上，如图8—5所示。用户对PROM编程是逐字逐位进行的。首先通过字线和位线选择需要编程的存储单元，然后通过规定宽度和幅度的脉冲电流将该存储管的熔丝熔断，这样就将该单元的内容改写了。

采用PN结击穿法的PROM的存储单元原理图如图8—6(a)所示，字线与位线相交处由两个肖特基二极管反向串联而成。正常工作时二极管不导通，字线和位线断开，相当于存储了“0”。若将该单元改写为“1”，可使用恒流源产生约100~150mA电流，使 VD_2 击穿短路，存储单元只剩下一个正向连接的二极管 VD_1 (见图8—6(b))，相当于该单元存储了“1”；未击穿 VD_2 的单元仍存储“0”。

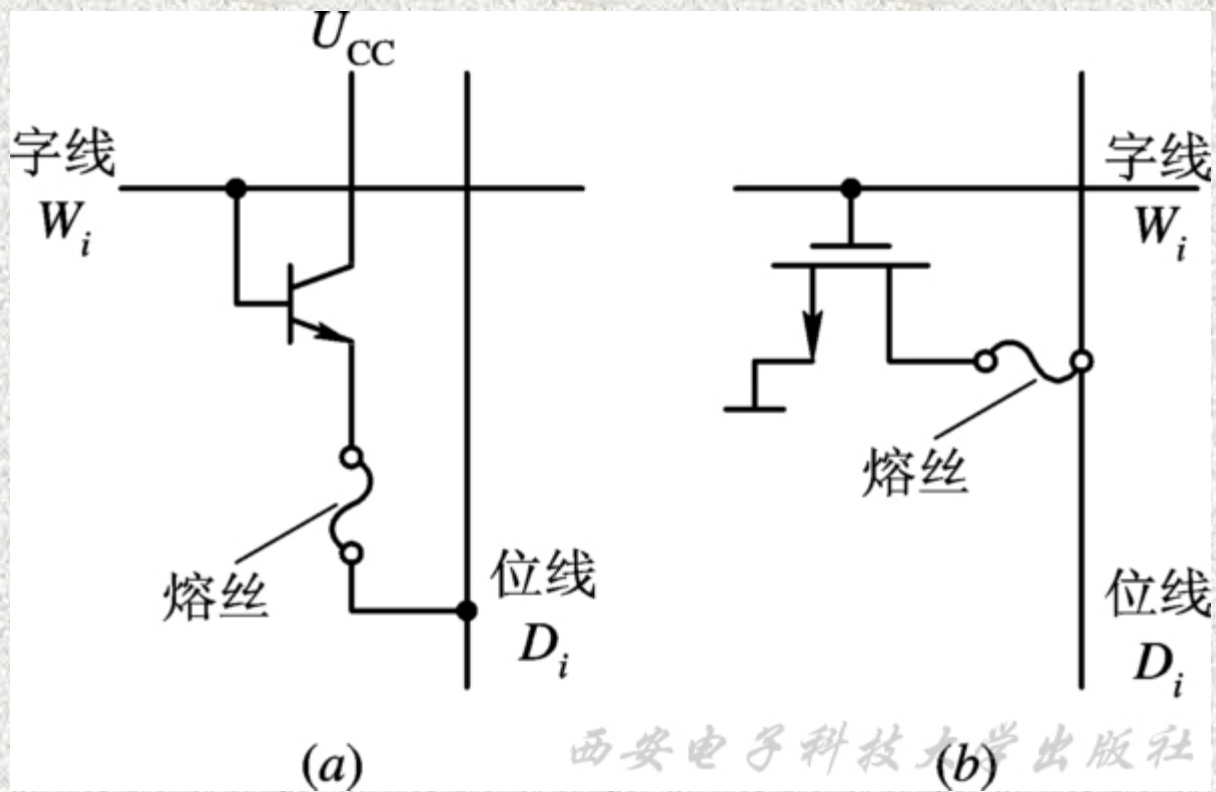


图8—5 熔丝型PROM的存储单元

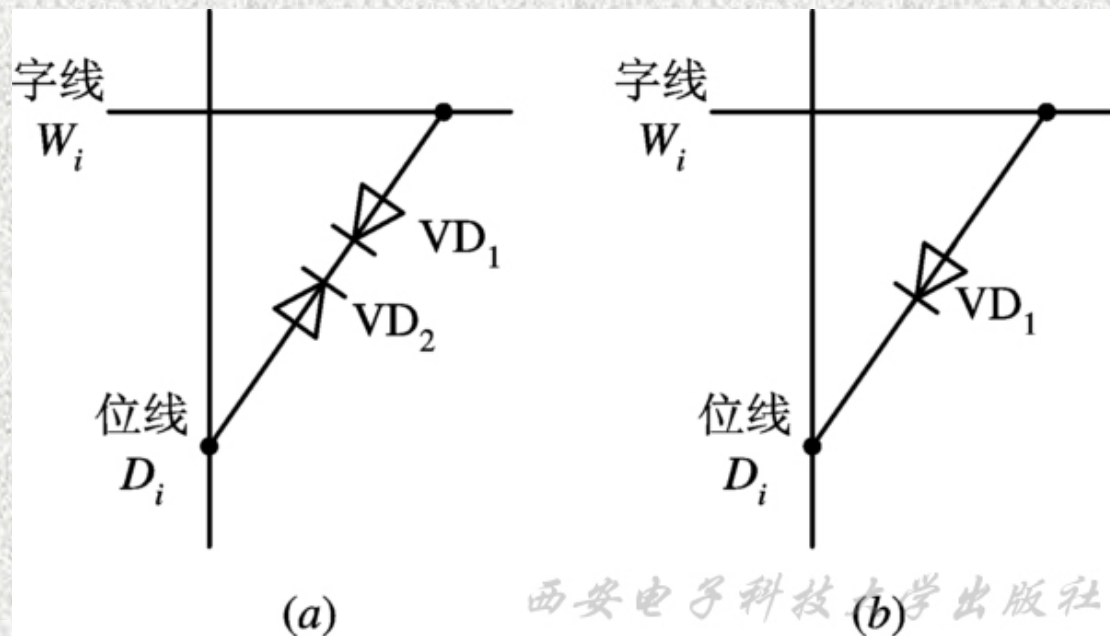









图8-6 PN结击穿型PROM的存储单元

3.可擦除的可编程ROM(EPROM)

EPROM利用特殊结构的浮栅MOS管进行编程，ROM中存储的数据可以进行多次擦除和改写。

最早出现的是用紫外线照射擦除的EPROM(Ultra  Violet Erasable Programmable Read  Only Memory, UVEEPROM)。不久又出现了用电信号可擦除的可编程ROM  (Electrically Erasable Programmable Read  Only Memory, E²EPROM)。后来又研制成功的快闪存储器(Flash Memory)也是一种用电信号擦除的可编程ROM。

(1) EPROM的存储单元采用浮栅雪崩注入MOS管(Floating-gate Avalanche-Injection  Metal-Oxide-Semiconductor, FAMOS管) 或叠栅注入MOS管(Stacked-gate  Injection  Metal-Oxide-Semiconductor, SIMOS管)。图8—7是SIMOS管的结构示意图和符号，它是一个N沟道增强型的MOS管，有 G_f 和 G_c 两个栅极。 G_f 栅没有引出线，而是被包围在二氧化硅(SiO_2)中，称之为浮栅， G_c 为控制栅，它有引出线。若在漏极D端加上约几十伏的脉冲电压，使得沟道中的电场足够强，则会造成雪崩，产生很多高能量的电子。此时若在 G_c 上加高压正脉冲，形成方向与沟道垂直的电场，便可以使沟道中的电子穿过氧化层面注入到 G_f ，于是 G_f 栅上积累了负电荷。由于 G_f 栅周围都是绝缘的二氧化硅，泄漏电流很小，所以一旦电子注入到浮栅之后，就能保存相当长的时间(通常浮栅上的电荷10年才损失30%)。

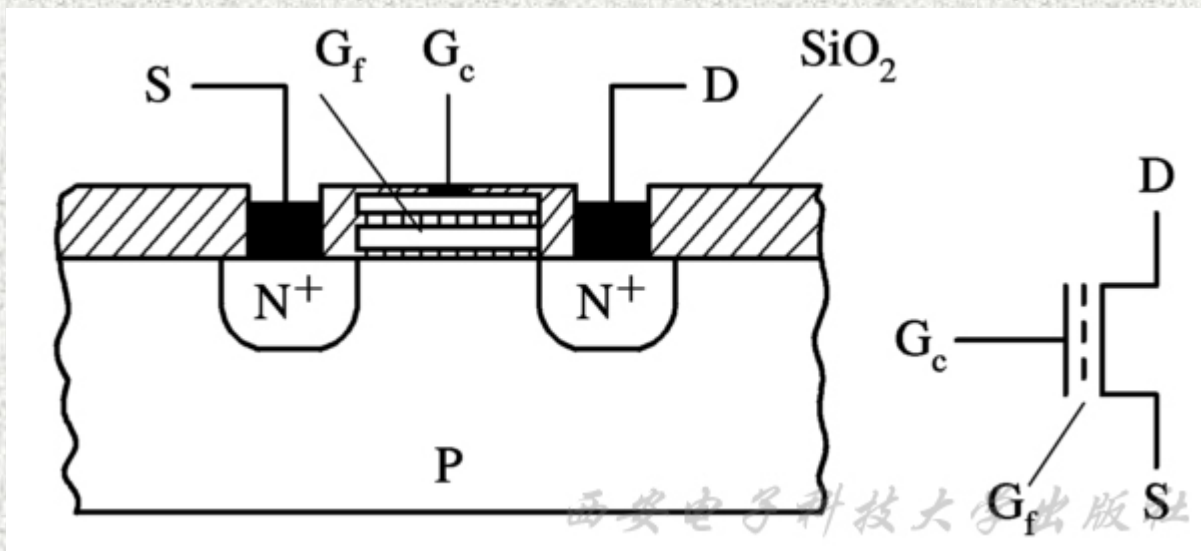



图8—7 SIMOS管的结构和符号

如果浮栅 G_f 上积累了电子，则使该MOS管的开启电压变得很高。此时给控制栅(接在地址选择线上)加+5V电压时，该MOS管仍不能导通，相当于存储了“0”；反之，若浮栅 G_f 上没有积累电子，MOS管的开启电压较低，因而当该管的控制栅被地址选中后，该管导通，相当于存储了“1”。可见，SIMOS管是利用浮栅是否积累负电荷来表示信息的。这种EPROM出厂时为全“1”，即浮栅上无电子积累，用户可根据需要写“0”。

擦除EPROM的方法是将器件放在紫外线下照射约20分钟，浮栅中的电子获得足够能量，从而穿过氧化层回到衬底中，这样可以使浮栅上的电子消失，MOS管便回到了未编程时的状态，从而将编程信息全部擦去，相当于存储了全“1”。

对EPROM的编程是在编程器上进行的，编程器通常与微机联用。

(2)E²PROM的存储单元如图8—8所示，图中V₂是选通管，V₁是另一种叠栅MOS管，称为浮栅隧道氧化层MOS管(Floating gate Tunnel Oxide MOS, Flotox管)，其结构如图8—9所示。Flotox管也是一个N沟道增强型的MOS管，与SIMOS管相似，它也有两个栅极——控制栅G_c和浮栅G_f，不同的是Flotox管的浮栅与漏极区(N⁺)之间有一小块面积极薄的二氧化硅绝缘层(厚度在2×10⁻⁸m以下)的区域，称为隧道区。当隧道区的电场强度大到一定程度(>10⁷V/cm)时，漏区和浮栅之间出现导电隧道，电子可以双向通过，形成电流。这种现象称为隧道效应。

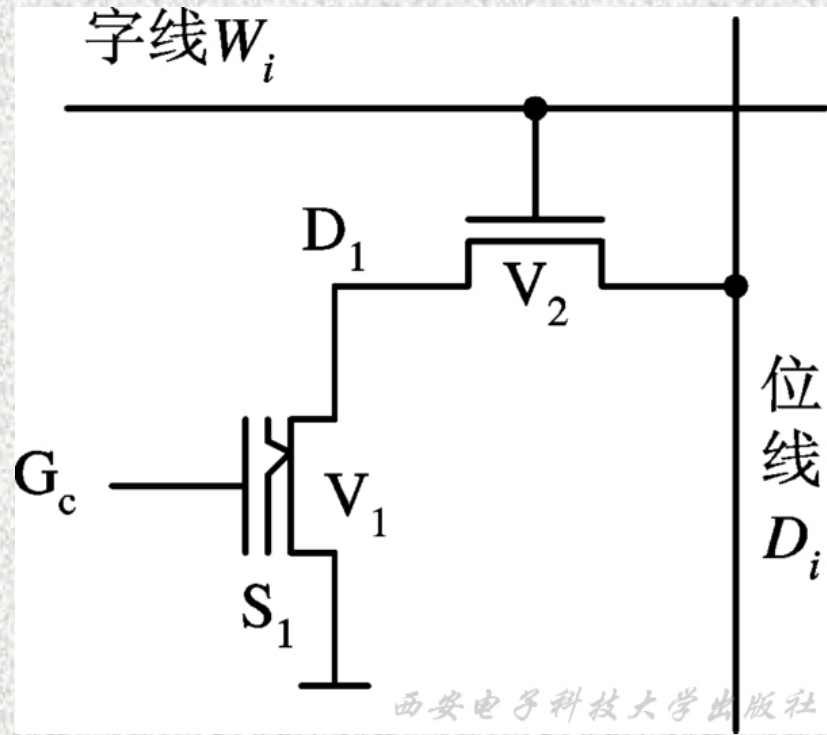


图8—8 E²PROM的存储单元

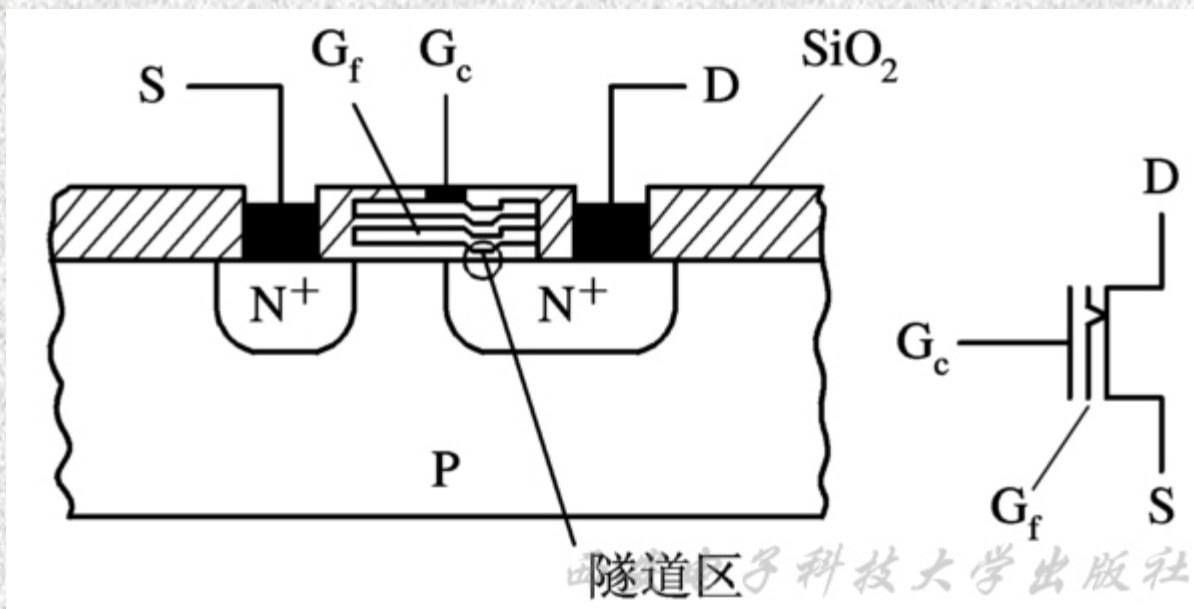


图8—9 Flotox管的结构和符号

(3)快闪存储器(Flash Memory)是新一代电信号擦除的可编程ROM。它既吸收了EPROM结构简单、编程可靠的优点,又保留了 E^2PROM 用隧道效应擦除快捷的特性,而且集成度可以做得很高。🔥

图8-10(a)是快闪存储器采用的叠栅MOS管示意图。其结构与 EPROM 中的SIMOS管相似,两者区别在于浮栅与衬底间氧化层的厚度不同。在EPROM中氧化的厚度一般为 $30\sim 40\text{nm}$,在快闪存储器中仅为 $10\sim 15\text{nm}$,而且浮栅和源区重叠的部分是源区的横向扩散形成的,面积极小,因而浮栅-源区之间的电容很小,当 G_c 和S之间加电压时,大部分电压将降在浮栅-源区之间的电容上。快闪存储器的存储单元就是用这样一只单管组成的,如图8-10(b)所示。

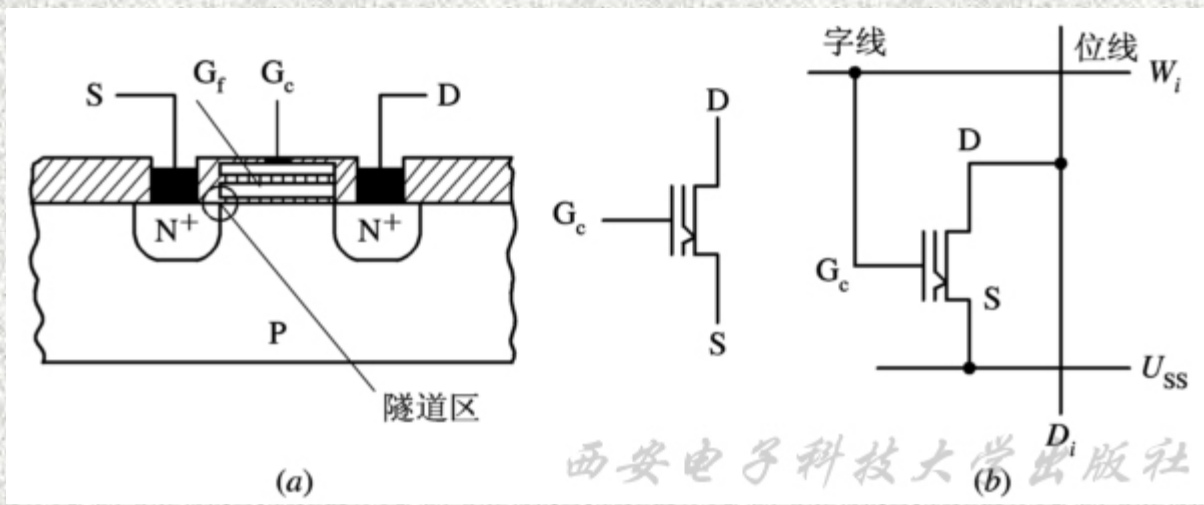


图8-10 快闪存储器

(a)叠栅MOS管；(b)存储单元


快闪存储器的写入方法和EPROM相同，即利用雪崩注入的方法使浮栅充电。🔥

在读出状态下，字线加上+5V，若浮栅上没有电荷，则叠栅MOS管导通，位线输出低电平；如果浮栅上充有电荷，则叠栅管截止，位线输出高电平。🔥

擦除方法是利用隧道效应进行的，类似于E²PROM写0时的操作。在擦除状态下，控制栅处于0电平，同时在源极加入幅度为12V左右、宽度为100ms的正脉冲，在浮栅和源区间极小的重叠部分产生隧道效应，使浮栅上的电荷经隧道释放。但由于片内所有叠栅MOS管的源极连在一起，因而擦除时是将全部存储单元同时擦除，这是不同于E²PROM的一个特点。🔥

快闪存储器具有集成度高、容量大、成本低和使用方便等优点。

8.1.4 随机存取存储器(RAM)

随机存取存储器也称随机存储器或随机读/写存储器，简称RAM。RAM工作时可以随时从任何一个指定的地址写入(存入)或读出(取出)信息。根据存储单元的工作原理不同，RAM分为静态RAM和动态RAM。 

1.静态随机存储器(SRAM)

1)基本结构

SRAM主要由存储矩阵、地址译码器和读/写控制电路三部分组成，其框图如图8—11所示。

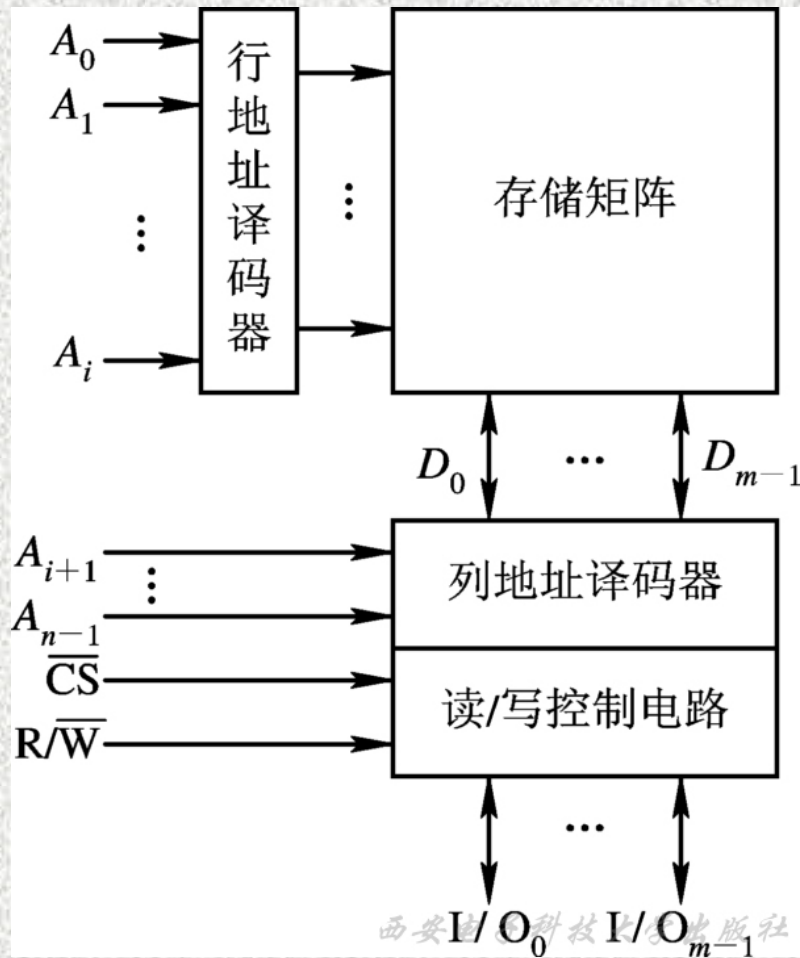


图8-11 SRAM的基本结构

2)SRAM的静态存储单元

静态RAM的存储单元如图8—12所示。

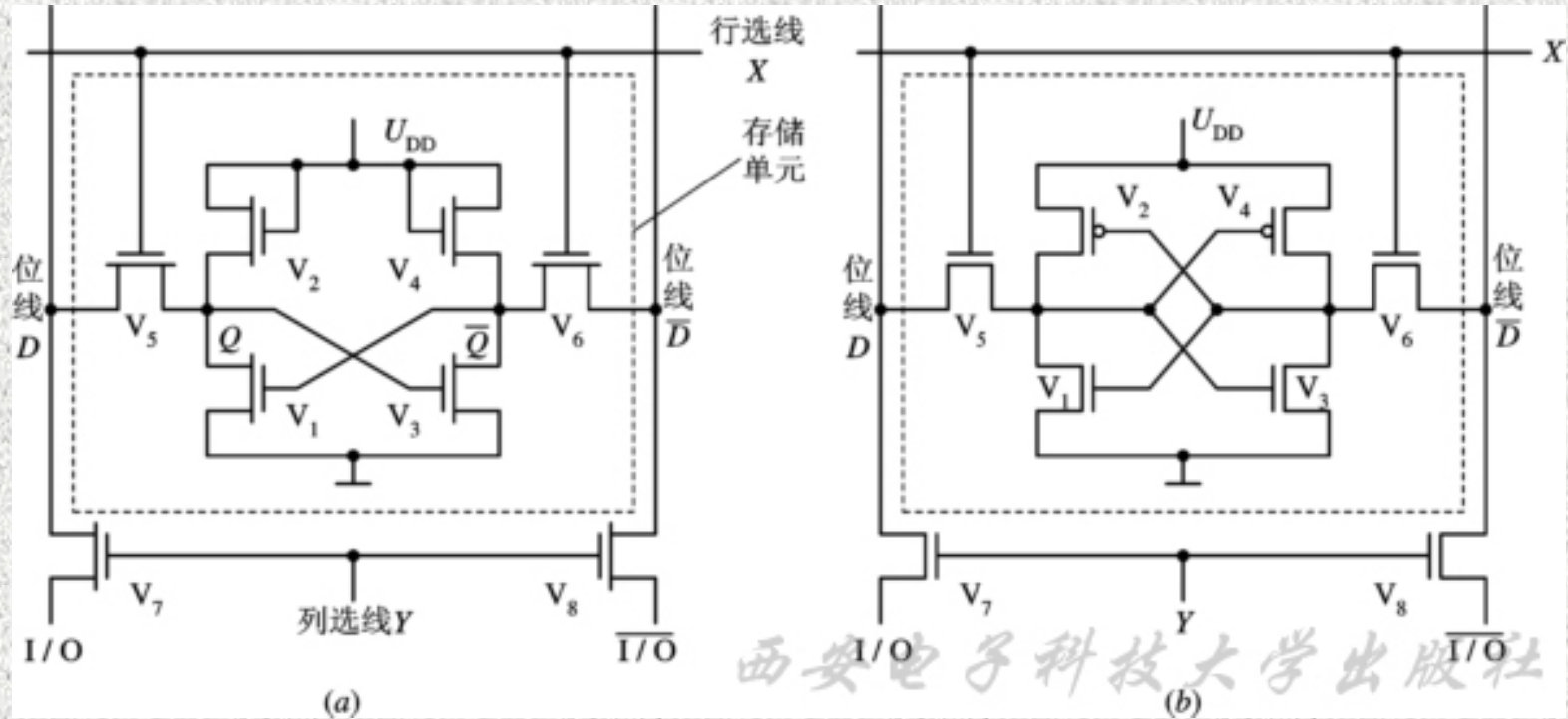


图8—12 (a)SRAM存储单元

(a)六管NMOS存储单元；(b)六管CMOS存储单元

2.动态随机存储器(DRAM)

动态RAM的存储矩阵由动态MOS存储单元组成。动态MOS存储单元利用MOS管的栅极电容来存储信息，但由于栅极电容的容量很小，而漏电流又不可能绝对等于0，所以电荷保存的时间有限。为了避免存储信息的丢失，必须定时地给电容补充漏掉的电荷。通常把这种操作称为“刷新”或“再生”，因此DRAM内部要有刷新控制电路，其操作也比静态RAM复杂。尽管如此，由于DRAM存储单元的结构能做得非常简单，所用元件少，功耗低，因而目前已成为大容量RAM的主流产品。

动态MOS存储单元有四管电路、三管电路和单管电路等。四管和三管电路比单管电路复杂，但外围电路简单，一般容量在4K以下的RAM(⊙)多采用四管或三管电路。图8-13(a)为四管动态MOS存储单元电路。图中， V_1 和 V_2 为两个N沟道增强型MOS管，它们的栅极和漏极交叉相连，信息以电荷的形式储存在电容 C_1 和 C_2 上， V_5 、 V_6 是同一列中各单元公用的预充管，是脉冲宽度为 $1\mu\text{s}$ 而周期一般不大于 $2(\text{⊙})\text{ms}$ 的预充电脉冲， C_{O1} 、 C_{O2} 是位线上的分布电容，其容量比 C_1 、 C_2 大得多。

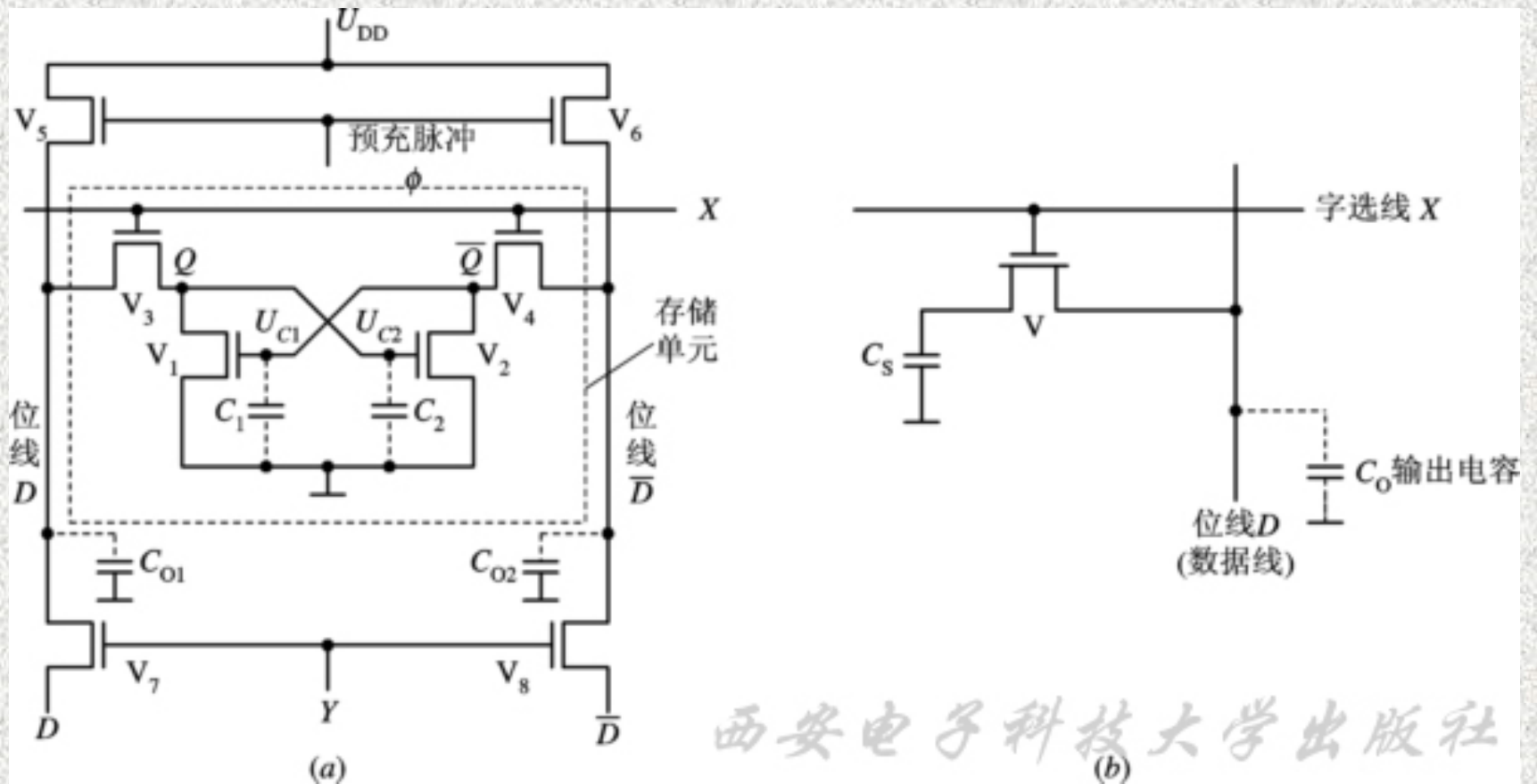


图8-13 动态MOS存储单元

若 C_1 被充电到高电位， C_2 上没有电荷，则 V_1 导通， V_2 截止，此时 $Q=0, Q=1$ ，这一状态称为存储单元的0状态；反之，若 C_2 充电到高电位， C_1 上没有电荷，则 V_2 导通， V_1 截止， $Q=1, Q=0$ ，此时称为存储单元的1状态。当字选线X为低电位时，门控管 V_3 、 V_4 均截止。在 C_1 和 C_2 上电荷泄漏掉之前，存储单元的状态维持不变，因此存储的信息被记忆。实际上，由于 V_3 、 V_4 存在着泄漏电流，电容 C_1 、 C_2 上存储的电荷将慢慢释放，因此每隔一定时间要对电容进行一次充电，即进行刷新。两次刷新之间的时间间隔一般不大于20ms。

在读出信息之前，首先加预充电脉冲 ϕ ，预充管 V_5 、 V_6 导通，电源 U_{DD} 向位线上的分布电容 C_{O1} 、 C_{O2} 充电，使 D 和 \bar{D} 两条位线都充到 U_{DD} 。预充脉冲消失后， V_5 、 V_6 截止， C_{O1} 、 C_{O2} 上的信息保持。

要读出信息时，该单元被选中(X 、 Y 均为高电平)， V_3 、 V_4 导通，若原来存储单元处于0状态($Q=0, Q=1$)，即 C_1 上有电荷， V_1 导通， C_2 无电荷， $(\ominus)V_2$ 截止，这样 C_{O1} 经 V_3 、 V_1 放电到0，使位线 D 为低电平，而 C_{O2} 因 $V_2(\ominus)$ 截止无放电回路，所以经 V_4 对 C_1 充电，补充了 C_1 漏掉的电荷，结果读出数据仍为 $D=1, D=0$ ；反之，若原存储信息为1($Q=1, Q=0$)， C_2 上有电荷，则预充电后 C_{O2} 经 V_4 、 V_2 放电到0，而 C_{O1} 经 V_3 对 C_2 补充充电，读出数据为 $D=0, D=1$ 。可见位线 D 、 \bar{D} 上读出的电位分别与 C_2 、 C_1 上的电位相同，同时每进行一次读操作，实际上也进行了一次补充充电，即刷新。

8.1.5 存储器容量的扩展 ♡

1.位数的扩展 ♡

存储器芯片的字长多数为一位、四位、八位等。当实际存储系统的字长超过存储器芯片的字长时，需要进行位扩展。



位扩展可以利用芯片的并联方式实现，图8-14是用八片 1024×1 位的RAM扩展为 1024×8 位RAM的存储系统框图。图中八片RAM的所有地址线、R/W、CS分别对应并接在一起，而每一片的I/O端作为整个RAM的I/O端的一位。

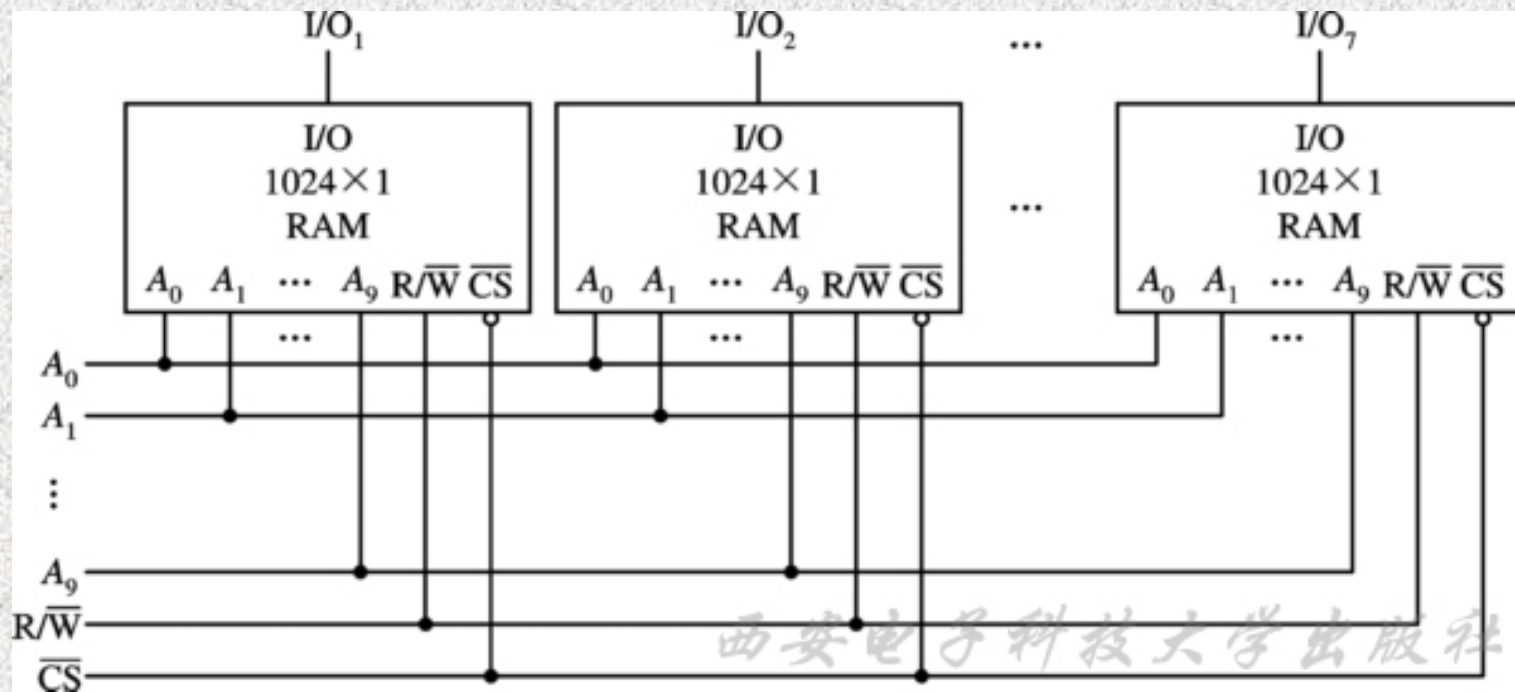


图8—14 RAM的位扩展连接法

2.字数的扩展

字数的扩展可以利用外加译码器控制芯片的片选(CS)输入端来实现。图8—15是用字扩展方式将四片 256×8 位的RAM扩展为 1024×8 位RAM的系统框图。图中，译码器的输入是系统的高位地址 A_9 、 A_8 ，其输出是各片RAM的片选信号。若 $A_9A_8=01$ ，则RAM(2)片的 $CS=\overline{0}$ ，其余各片RAM的CS均为1，故选中第二片。只有该片的信息可以读出，送到位线上，读出的内容则由低位地址 $A_7\sim A_0$ 决定。显然，四片RAM轮流工作，任何时候，只有一片RAM处于工作状态，整个系统字数扩大了四倍，而字长仍为八位。

ROM的字扩展方法与上述方法相同。

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/956210233240011004>