《数字电子技术基础》（第五版）教学课件

清华大学
阎石　王红

联系地址：清华大学 自动化系
邮政编码：100084

电子信箱：wang_hong@tsinghua.edu.cn
联系电话：(010)62792973

第七章　半导体存储器

第七章　半导体存储器

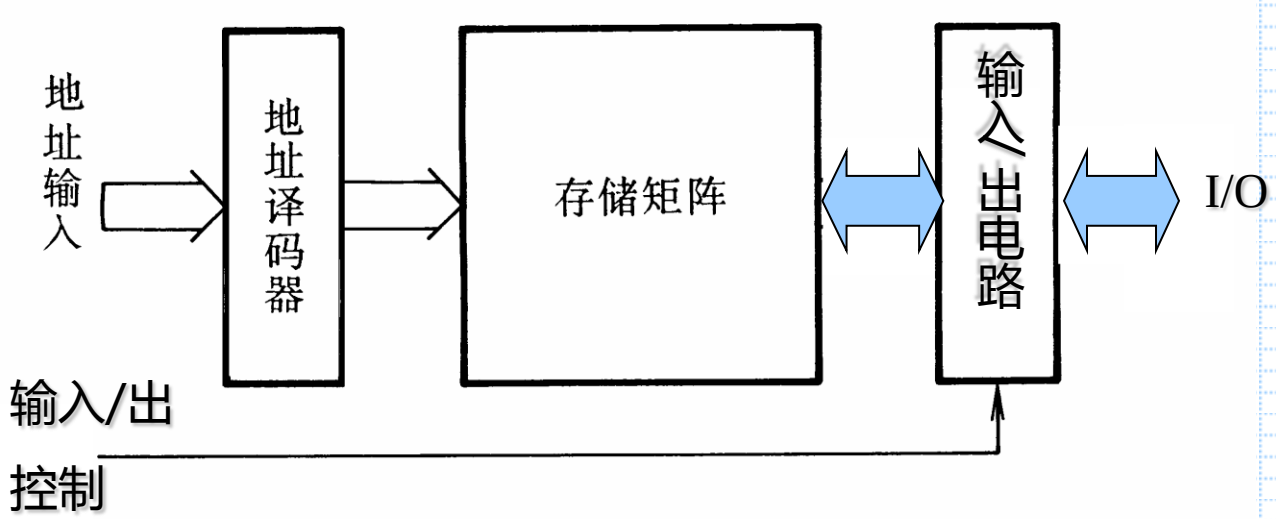
7.1 概述

能存储大量二值信息的器件

！单元数庞大

！输入/输出引脚数目有限

一、一般结构形式



二、分类

1、从存/取功能分：

①只读存储器 (Read-Only-Memory)

- 掩模***ROM***
- 可编程***ROM***
- 可擦除的可编程***EPROM***

②随机读/写 (Random-Access-Memory)

- 静态***RAM***
- 动态***RAM***

2、从工艺分：

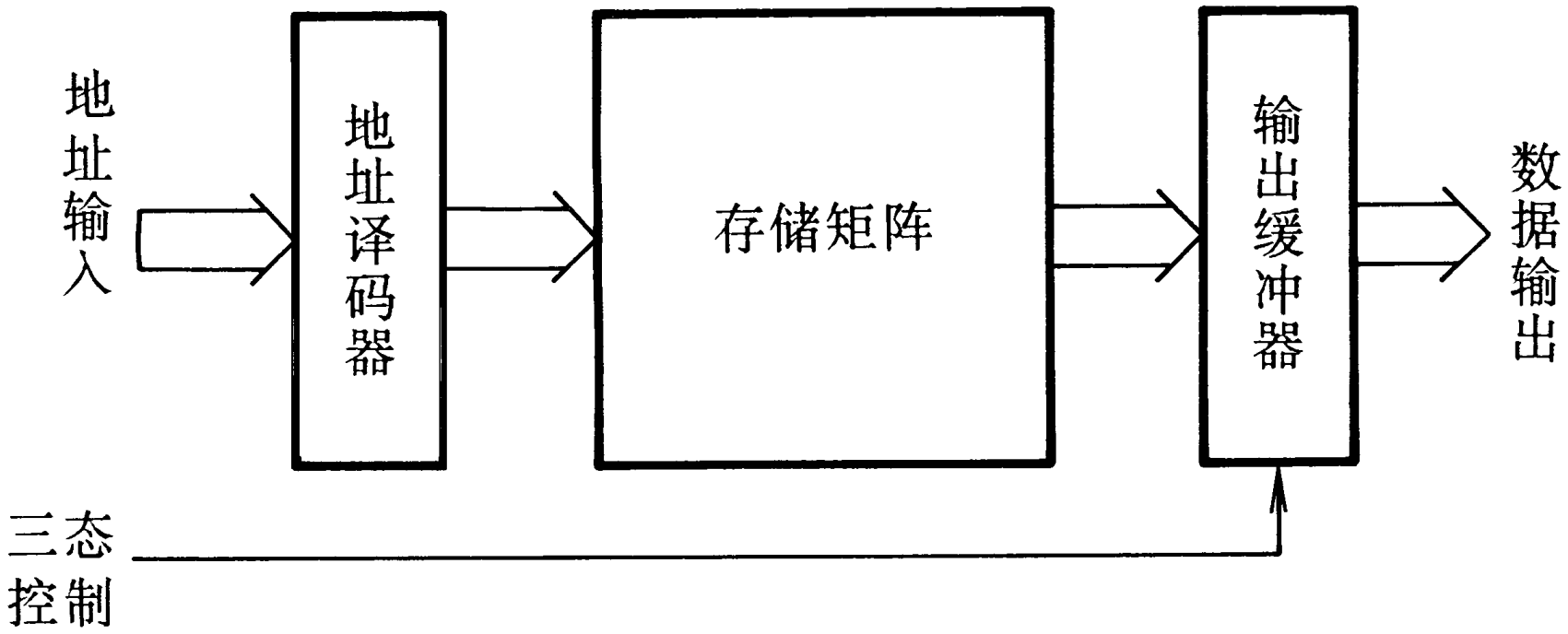
①双极型

②MOS型

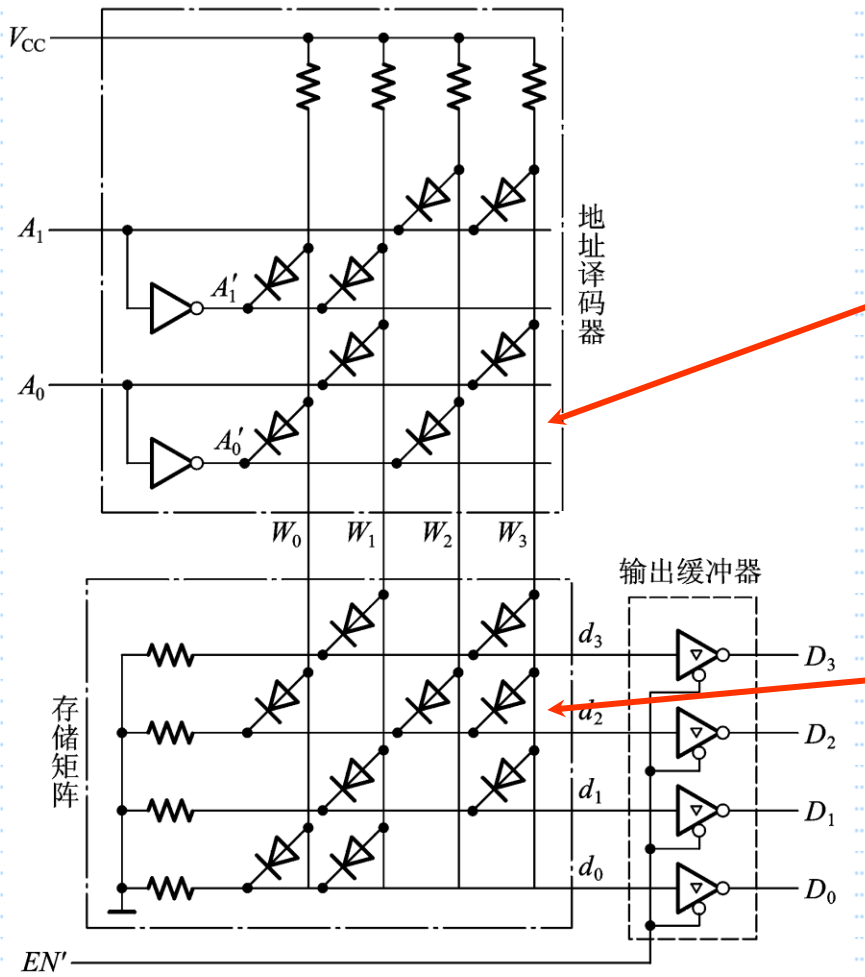
7.2 ROM

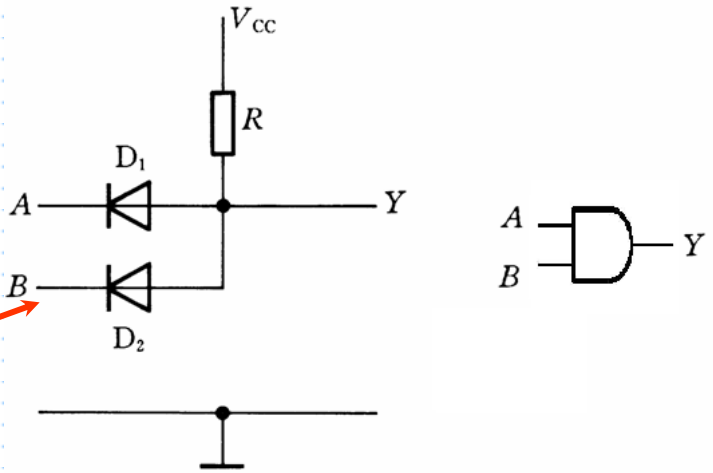
7.2.1 掩模ROM

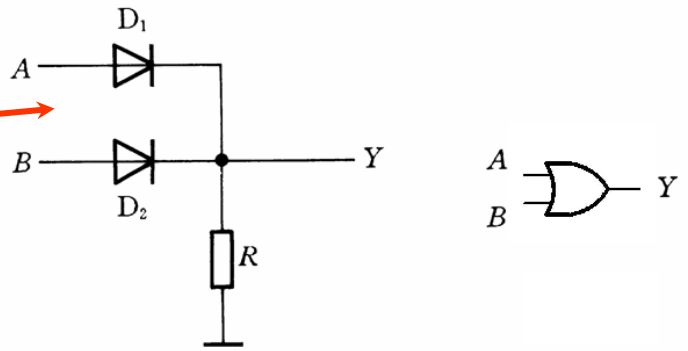
一、结构

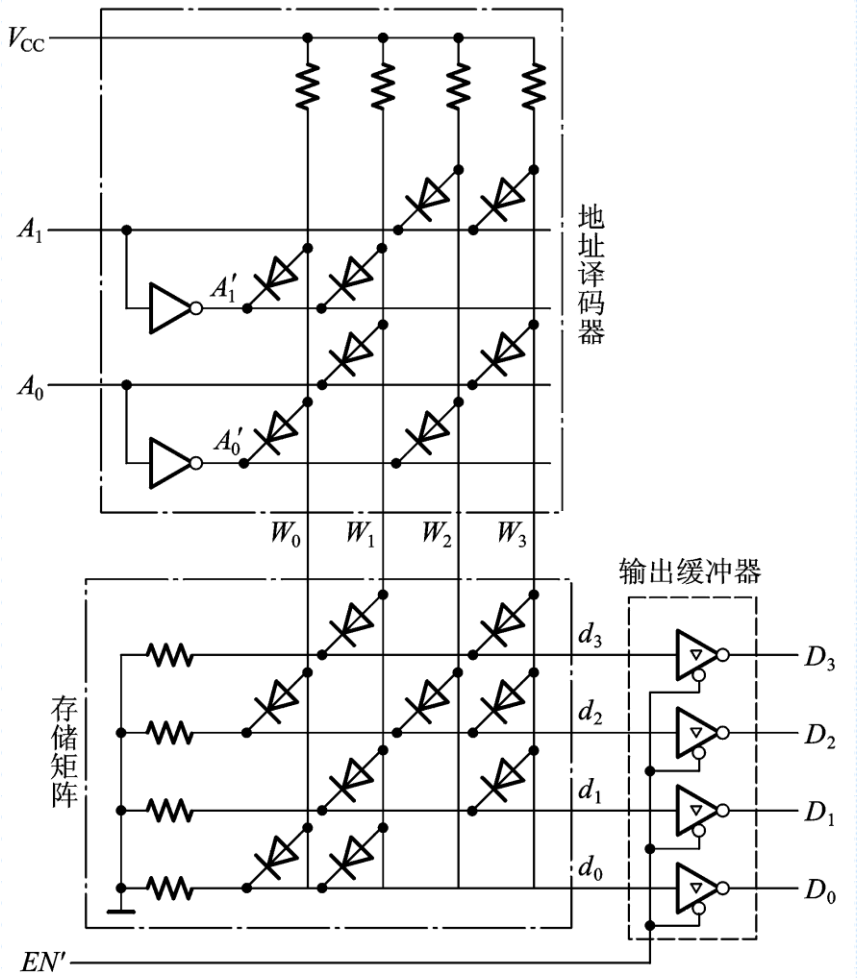


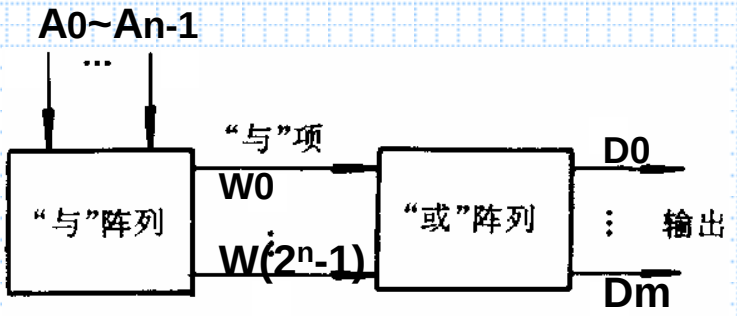
二、举例


V_{CC}
A_1
A_0
A_1'
A_0'
地址译码器
W_0
W_1
W_2
W_3
输出缓冲器
存储矩阵
d_3
d_2
d_1
d_0
D_3
D_2
D_1
D_0
EN'


V_{CC}
R
D_1
A
Y
B
D_2
A
B
Y


D_1
A
B
Y
D_2
R
A
B
Y

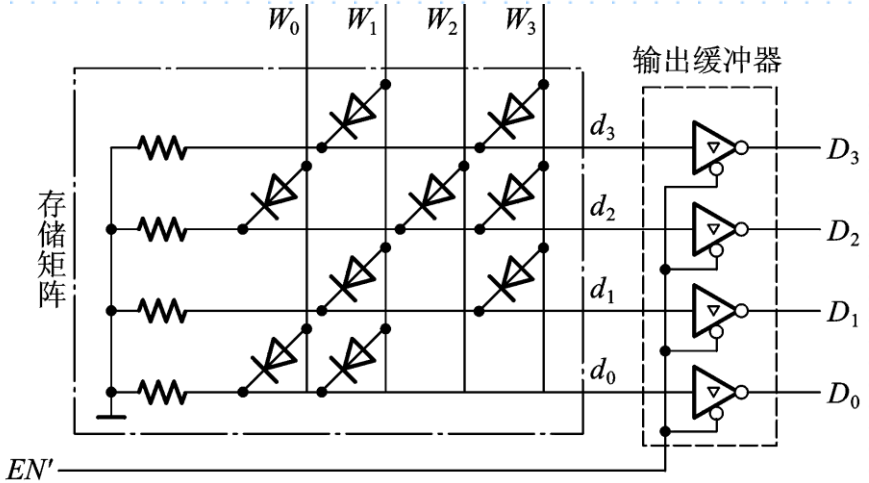


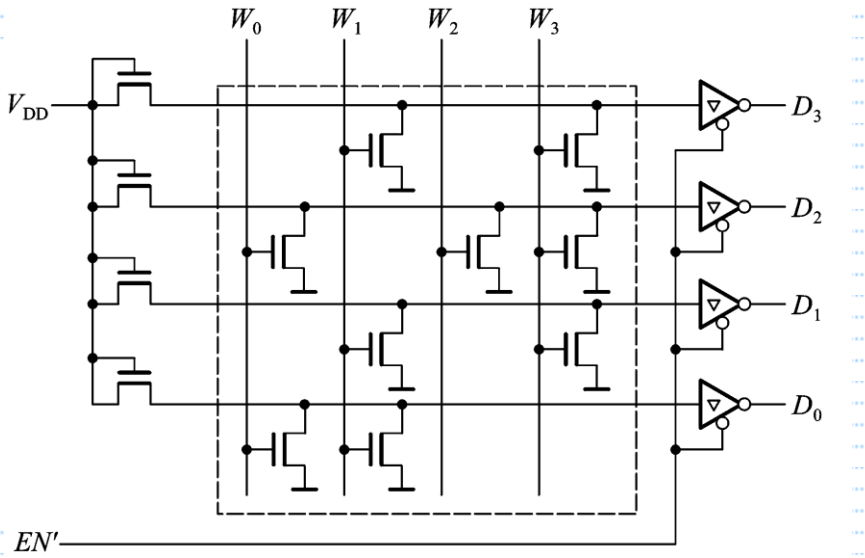


地址		数据			
A_1	A_0	D_3	D_2	D_1	D_0
0	0	0	1	0	1
0	1	1	0	1	1
1	0	0	1	0	0
1	1	1	1	1	0

两个概念：

- 存储矩阵的每个交叉点是一个“存储单元”，存储单元中有器件存入“1”，无器件存入“0”



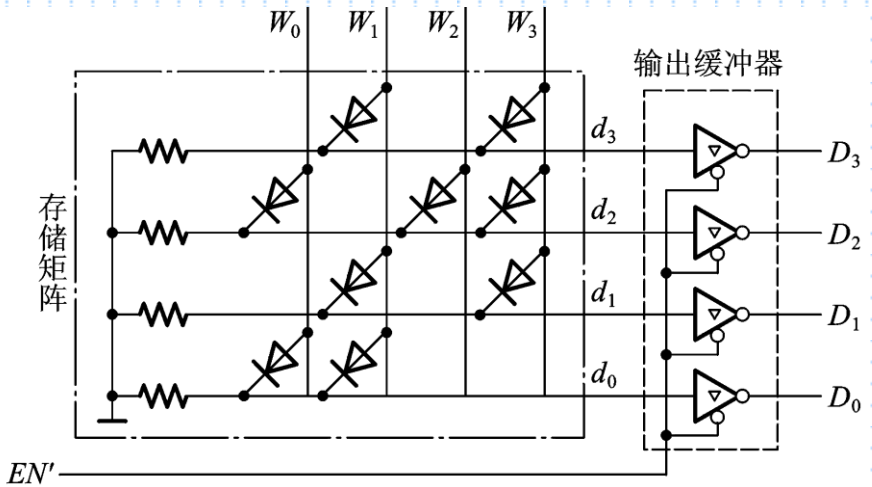


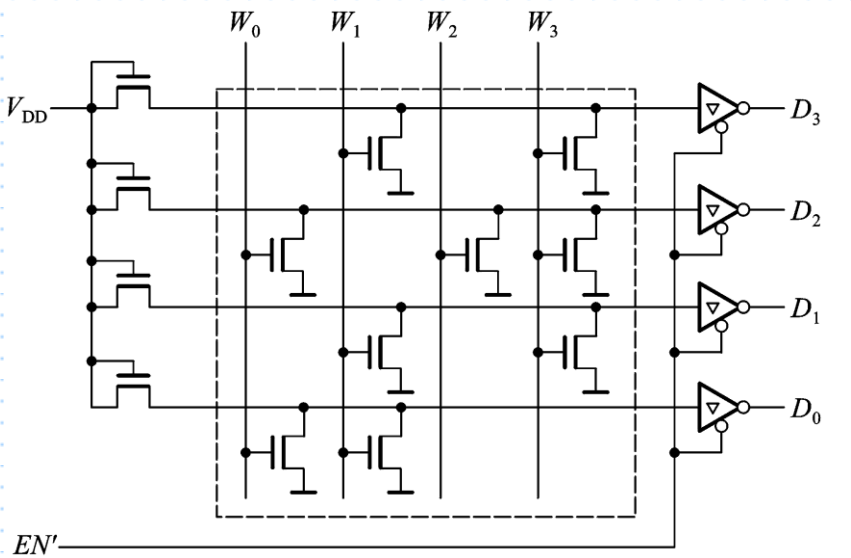
- 存储器的容量：“字数 × 位数”

掩模ROM的特点：

出厂时已经固定，不能更改，适合大量生产

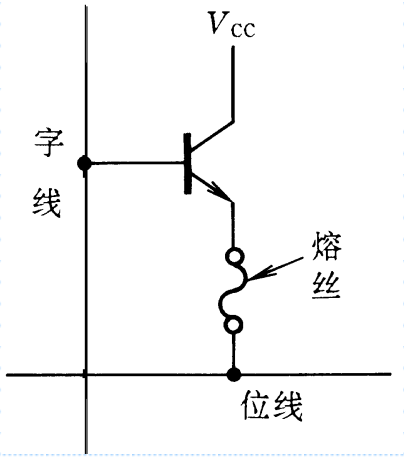
简单，便宜，非易失性

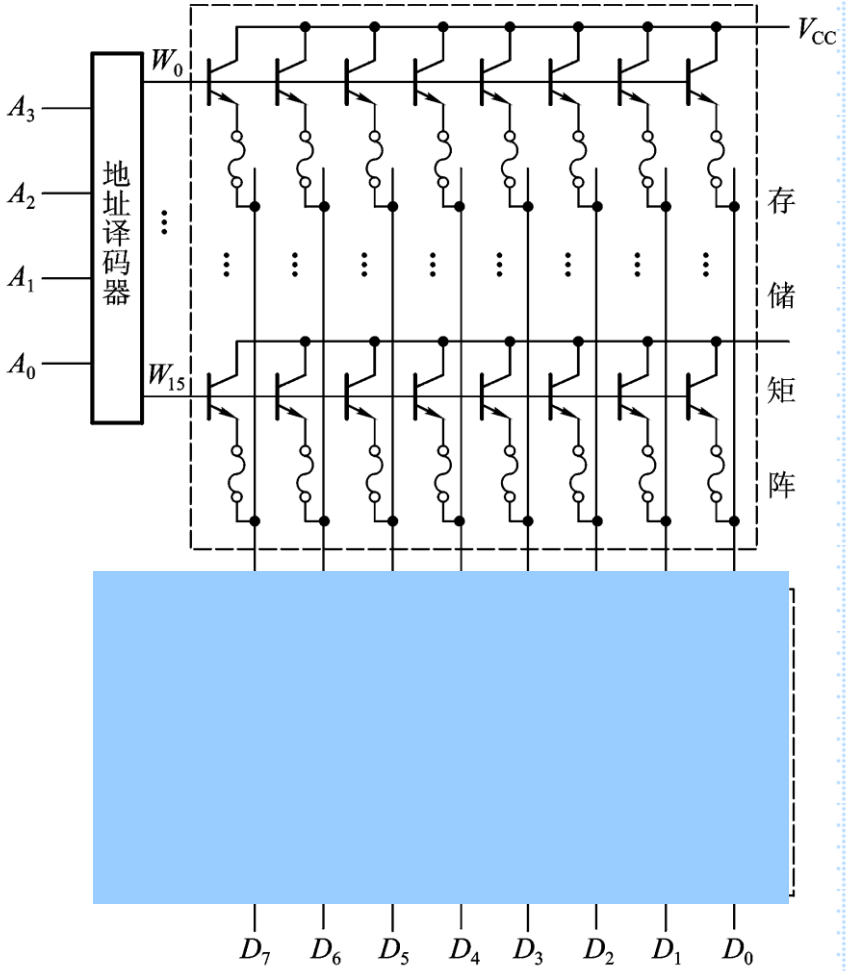




7.2.2 可编程ROM（PROM）

总体结构与掩模ROM一样，但存储单元不同





*熔丝由易熔合金制成

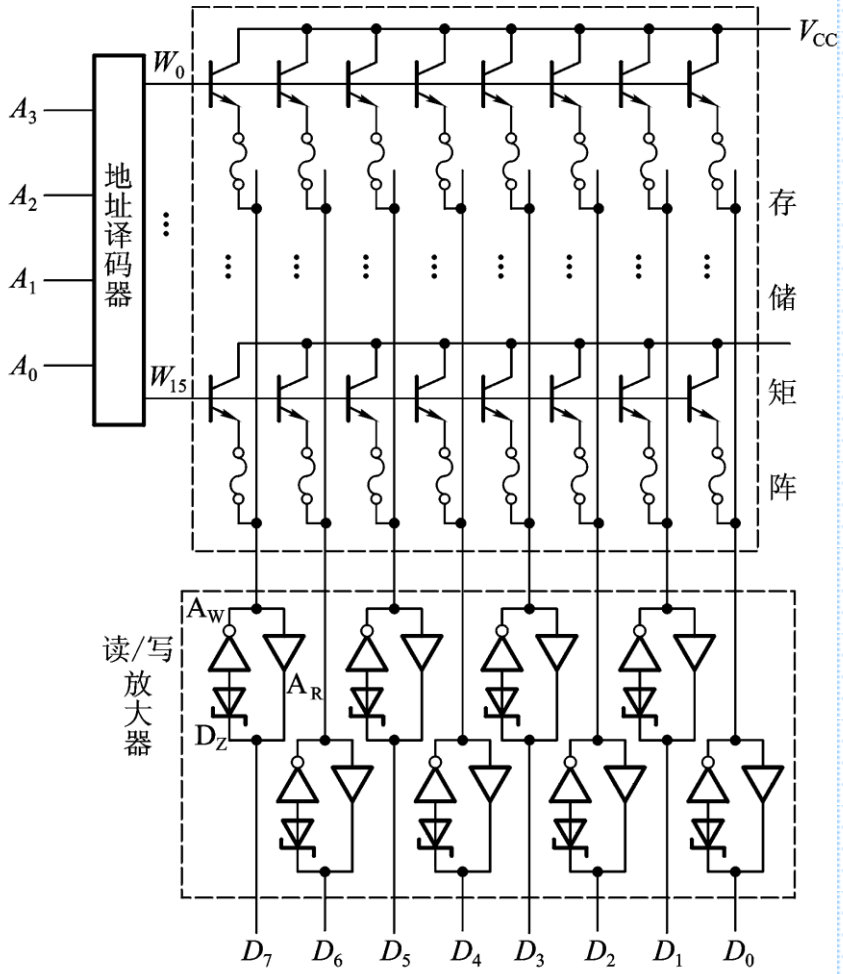
*出厂时，每个结点上都有

*编程时将不用的熔断

!!　是一次性编程，不能改写

7.2.2 可编程ROM（PROM）

总体结构与掩模ROM一样，但存储单元不同



写入时，要使用编程器

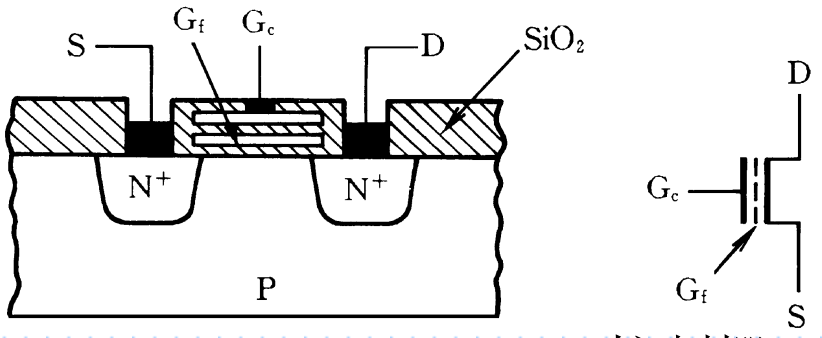
7.2.3 可擦除的可编程ROM（EPROM）

总体结构与掩模ROM一样，但存储单元不同

一、用紫外线擦除的PROM（UVEPROM）

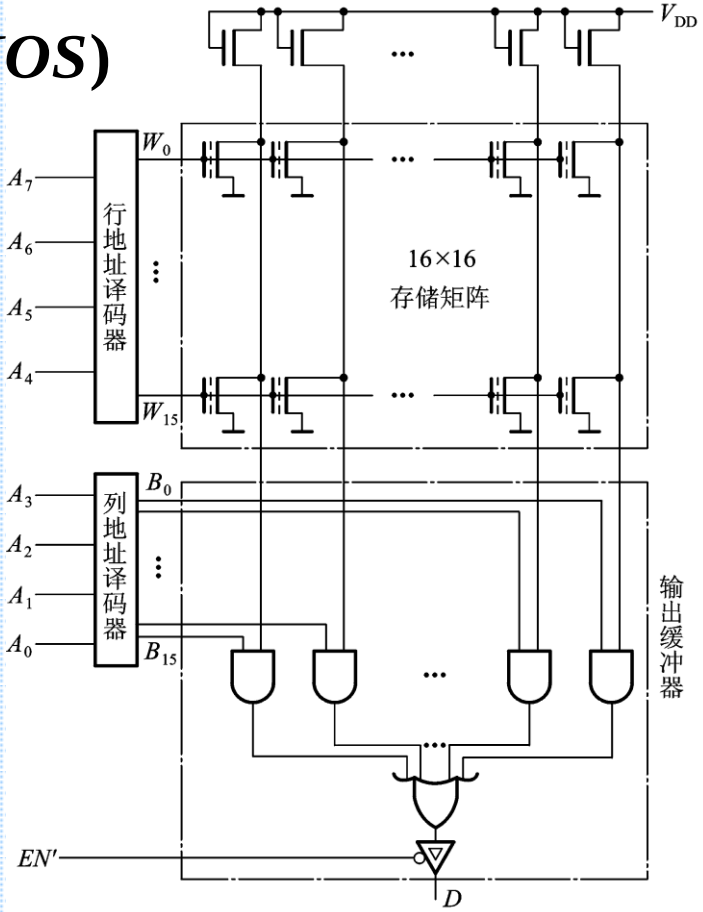
SIMOS(Stacked – gate Injuction MOS)

叠栅注入*MOS*管



G_c：控制栅

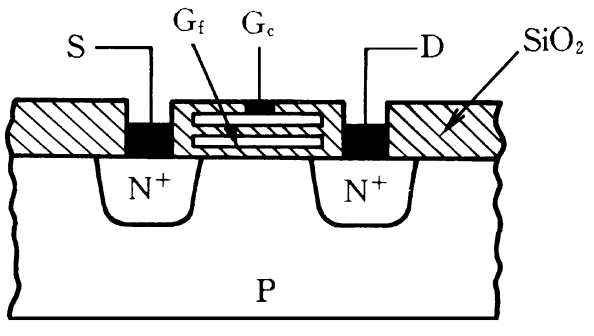
G_f：浮置栅

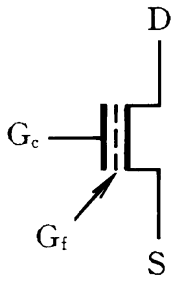


工作原理：

若G_f上充以负电荷，则G_c处正常逻辑高电平下不导通

若G_f上未充负电荷，则G_c处正常逻辑高电平下导通





“写入”：雪崩注入，$D-S$间加高压（20～25V），发生雪崩击穿

同时在G_c上加25V，50ms宽的正脉冲，

吸引高速电子穿过SiO_2到达G_f，形成注入电荷

“擦除”：通过照射产生电子–空穴对，提供泄放通道

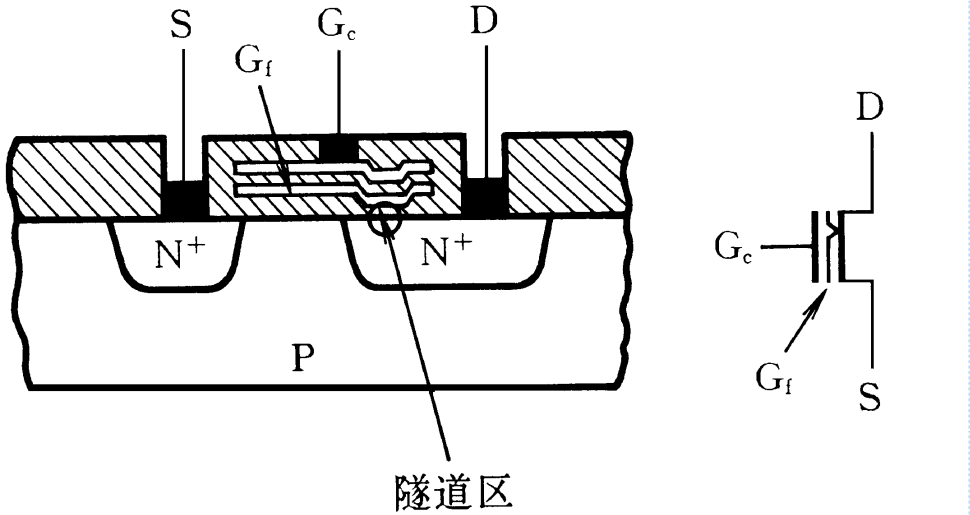
紫外线照射20～30分钟（阳光下一周，荧光灯下3年）

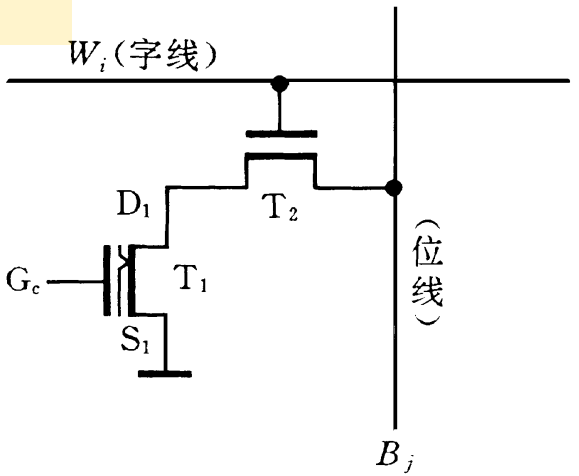
二、电可擦除的可编程ROM（E²PROM）

总体结构与掩模ROM一样，但存储单元不同

为克服*UVEPROM*擦除慢，操作不便的缺点

采用*FLOTOX*(浮栅隧道氧化层*MOS*管)

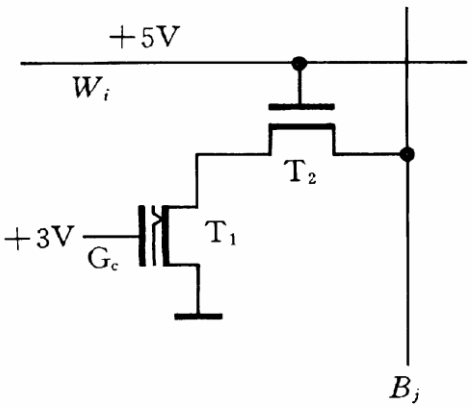




G_f与D之间有小的隧道区，SiO_2厚度$<2\times10^{-8}m$

当场强达到一定大小（$10^7V/cm$），电子会穿越隧道

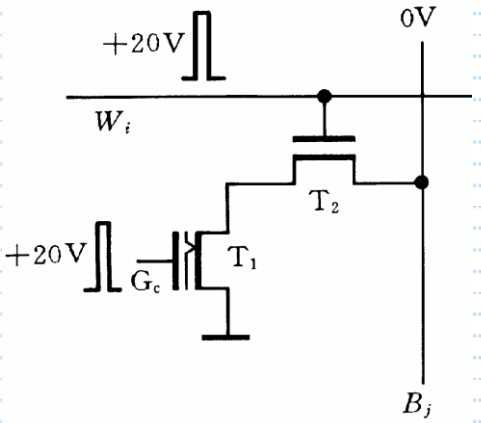
---“隧道效应”



工作原理：

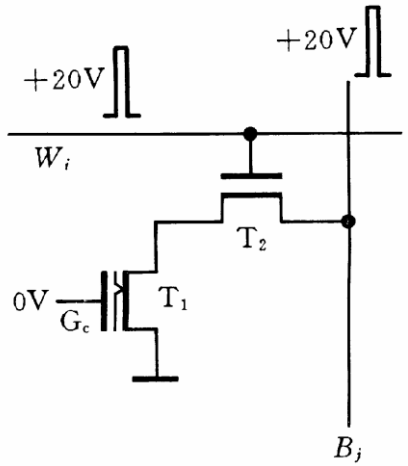
G_f充电荷后，正常读出G_C电压（3V）下，T截止

未充电荷时，正常读出G_C电压（3V）下，T导通



充电：W_i,G_C加20V,10ms的正脉冲，B_j接0

电子隧道区→G_f



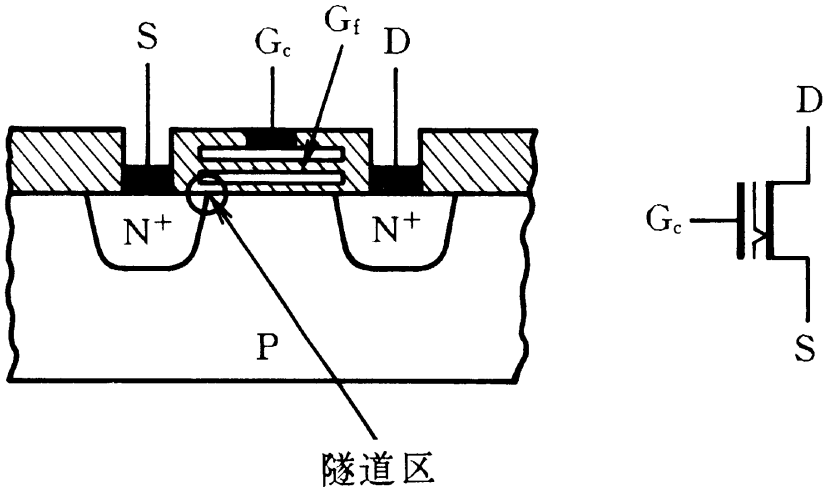
放电：G_C接0，W_i,B_j加正脉冲，

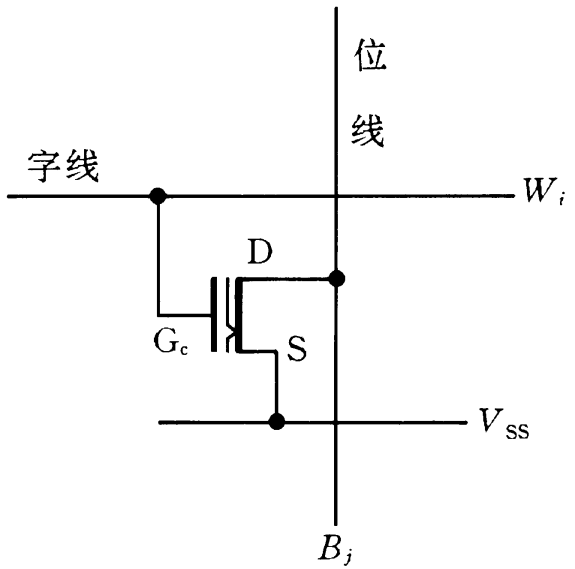
G_f上电荷经隧道区放电

三、快闪存储器（Flash Memory）

为提高集成度，省去T2（选通管）

改用叠栅MOS管（类似SIMOS管）





G_f与衬底间SiO_2更薄（10～15nm）

G_f与S区有极小的重叠区—（隧道区）

G_f放电，利用隧道效应

$G_c=0, V_{ss}$加12V,100ns的正脉冲

G_f上电荷经隧道区放电

*工作原理：

向G_f充电利用雪崩注入方式，

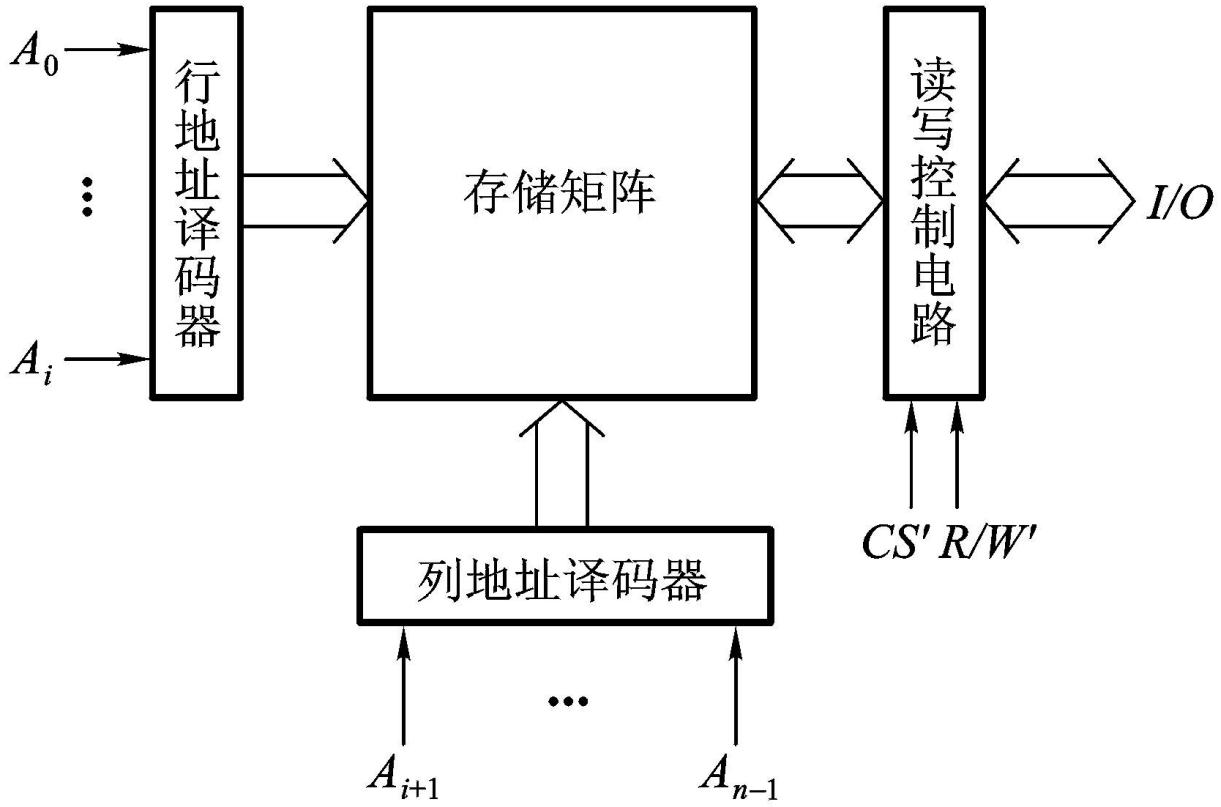
$D-S$加正压（6V），V_{ss}接0

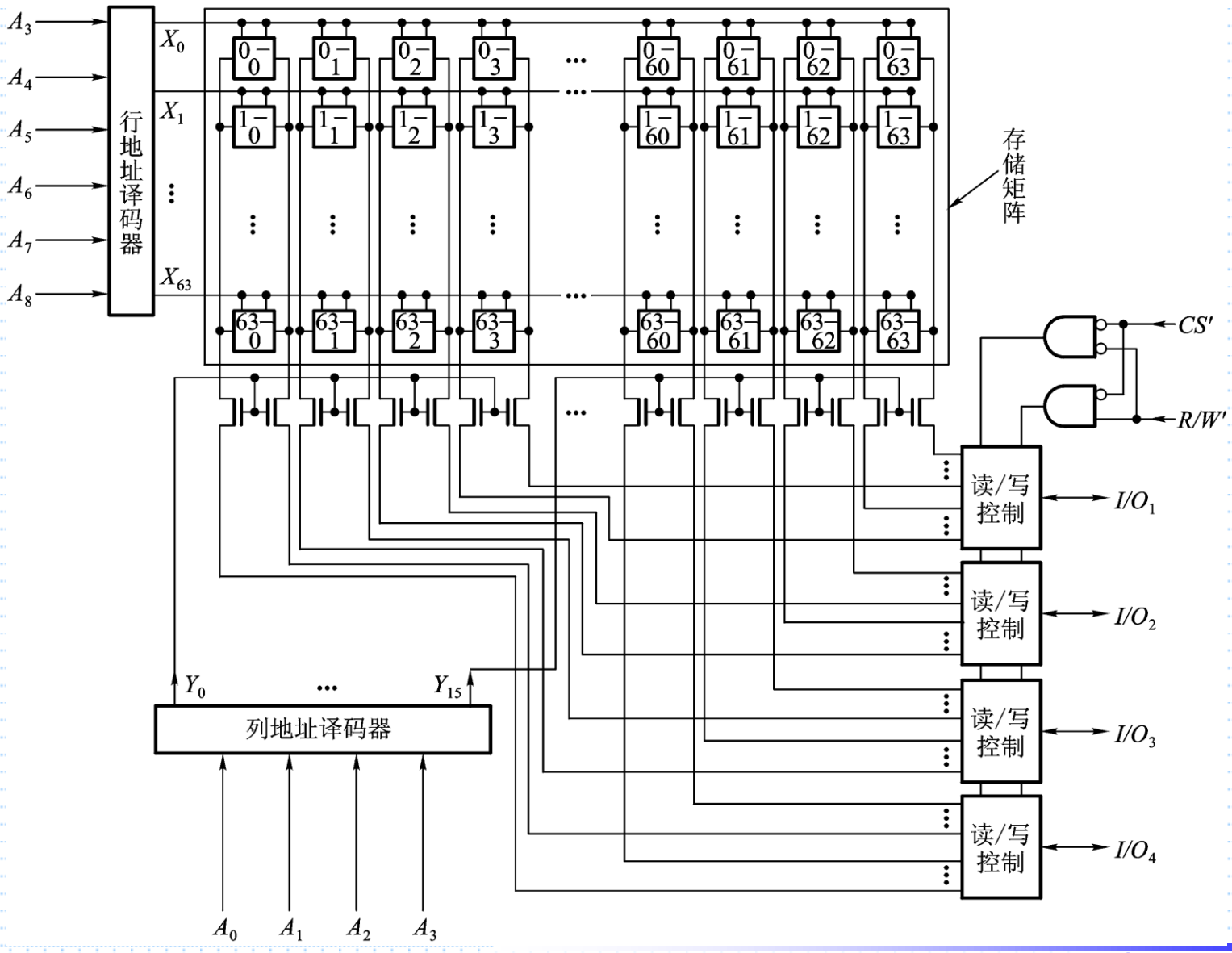
G_c加12V,10us的正脉冲

7.3 随机存储器RAM

7.3.1 静态随机存储器（SRAM）

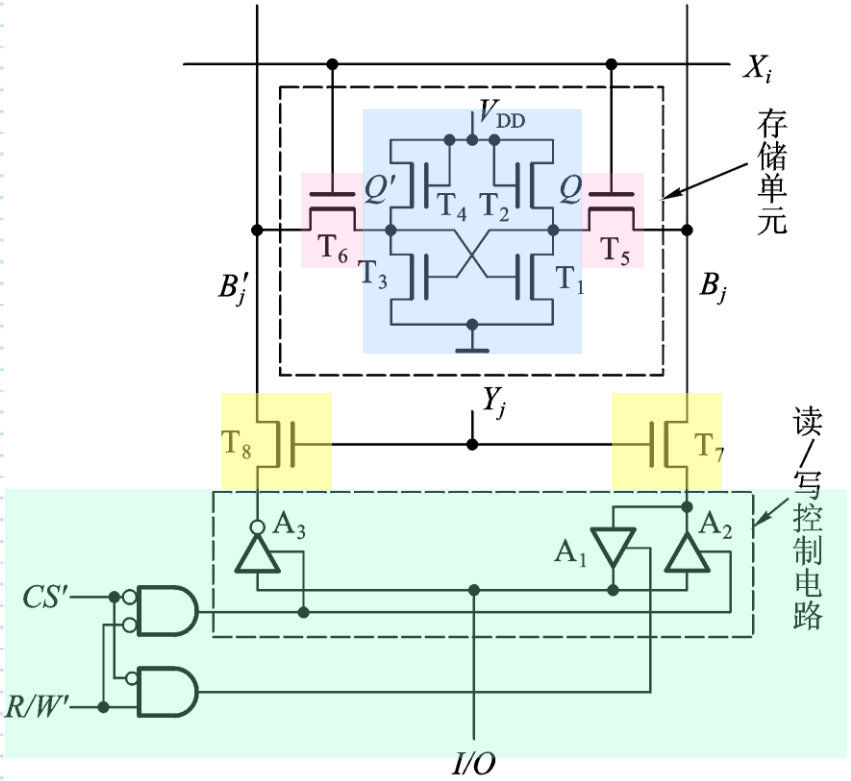
一、结构与工作原理




A_3
A_4
A_5
A_6
A_7
A_8
行地址译码器
X_0
X_1
X_{63}
存储矩阵
CS'
R/W'
读/写控制
I/O_1
I/O_2
I/O_3
I/O_4
Y_0
Y_{15}
列地址译码器
A_0
A_1
A_2
A_3

二、SRAM的存储单元

六管N沟道增强型MOS管



$T_1 \sim T_4$为基本RS触发器，作存储单元

$X_i = 1$时，能在1行中被选中，

T_5, T_6导通，Q、Q'与B_j、B_j'相通

当$CS' = 0$时，

若$R/W' = 1$，则A_1导通，A_2与A_3截止，

$Q \rightarrow I/O$，读操作

若$R/W' = 0$，则A_1截止，A_2与A_3导通，

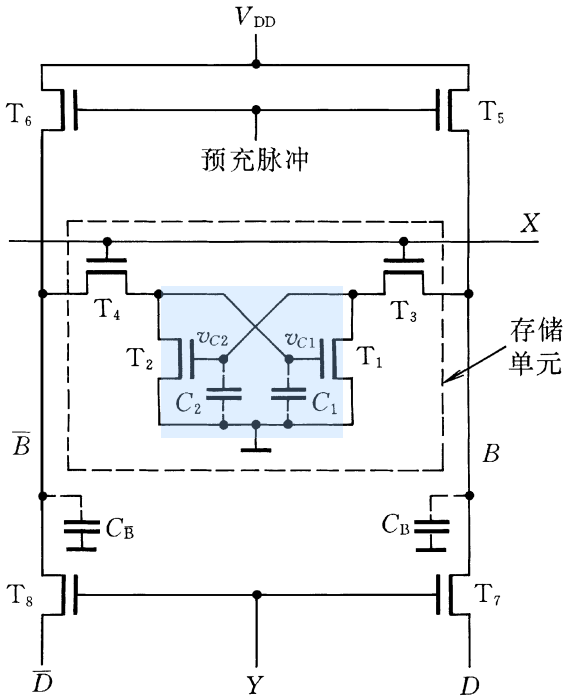
$I/O \rightarrow Q$，写操作

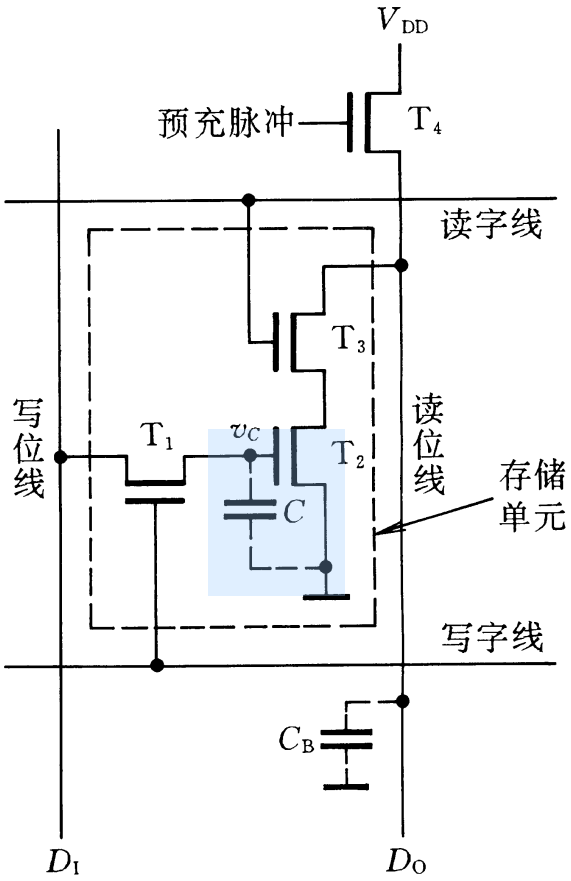
$Y_j = 1$时，所在列被选中，

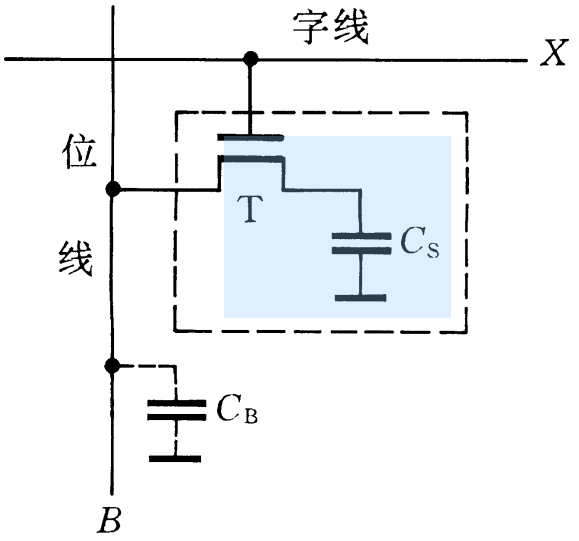
T_7, T_8导通，这时{第i行 / 第j列}单元与缓冲器相连

7.3.2* 动态随机存储器（DRAM）

动态存储单元是利用MOS管栅极电容可以存储电荷的原理







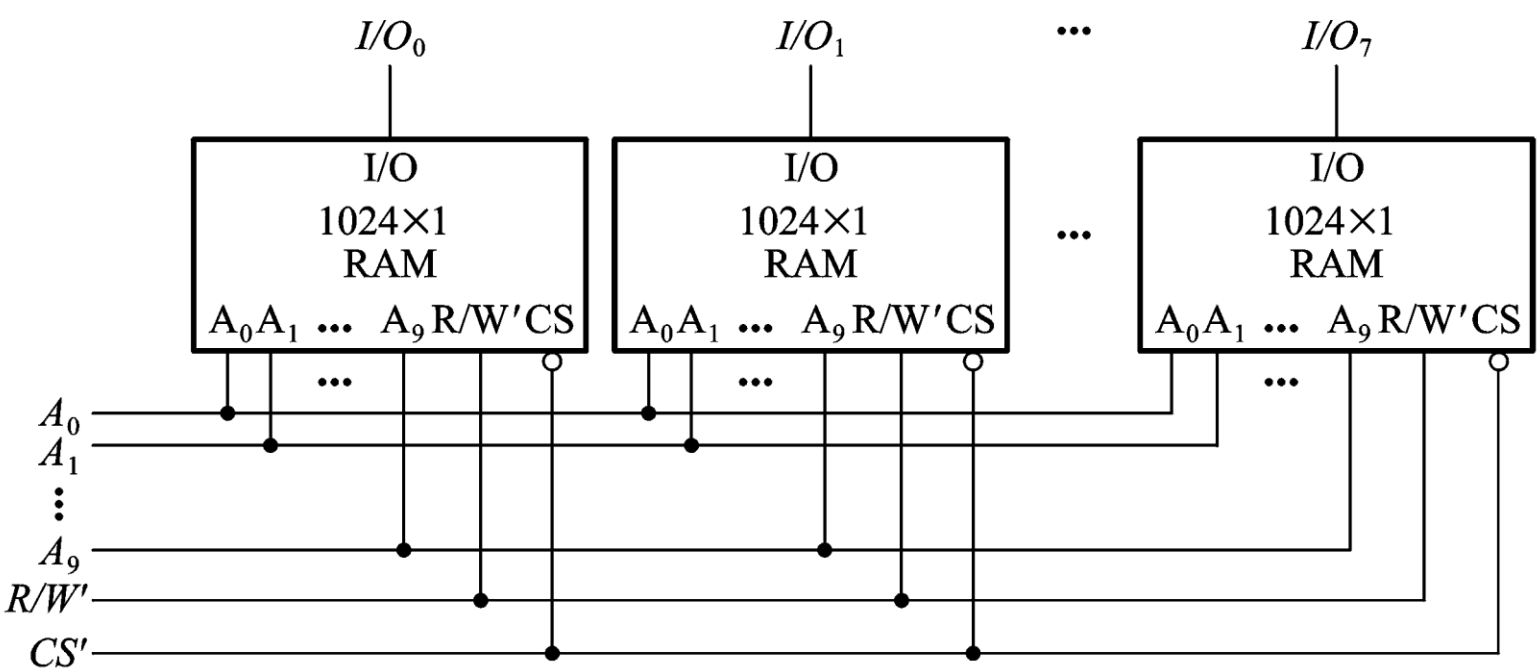
7.4 存储器容量的扩展

7.4.1 位扩展方式

适用于每片RAM,ROM字数够用而位数不够时

接法：将各片的地址线、读写线、片选线并联即可

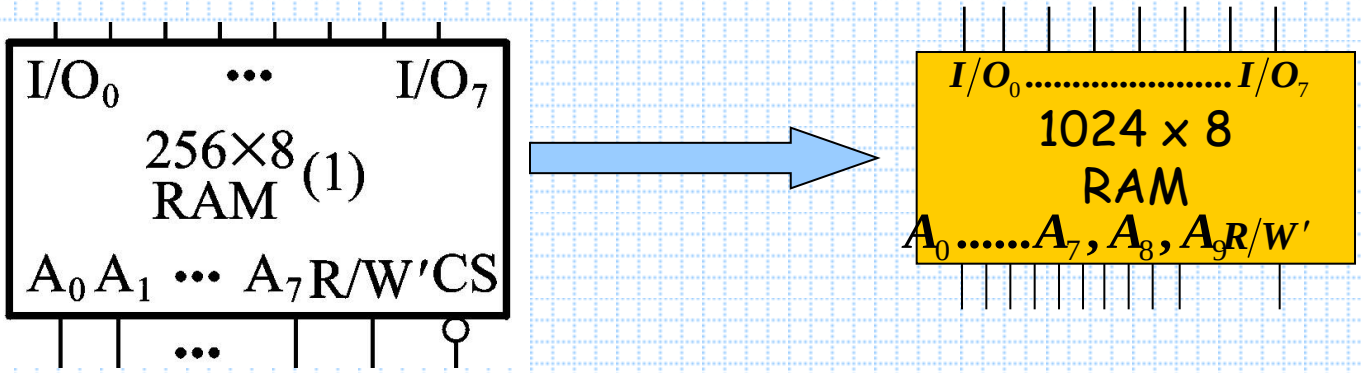
例：用八片1024 x 1位→1024 x 8位的RAM



7.4.2 字扩展方式

适用于每片RAM,ROM位数够用而字数不够时

例：用四片256 x 8位→1024 x 8位 RAM



数据线：$I/O_0 \sim I/O_7$

地址线：$A_0 \sim A_7$

读/写信号：R/W'

片选信号：CS'

数据线：$I/O_0 \sim I/O_7$

地址线：$A_0 \sim A_7, A_8, A_9$

读/写信号：R/W'

每一片提供256个字，需要256个地址($A_{0\sim7}$：0～0－－－－1～1)

用A_9，A_8两位代码区分四片

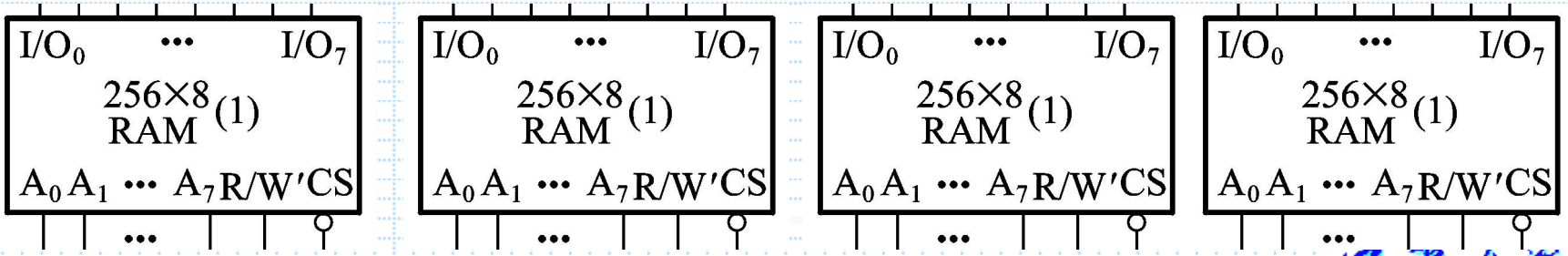
即将A_9A_8译成Y_0'～Y_3'，分别接四片的CS'

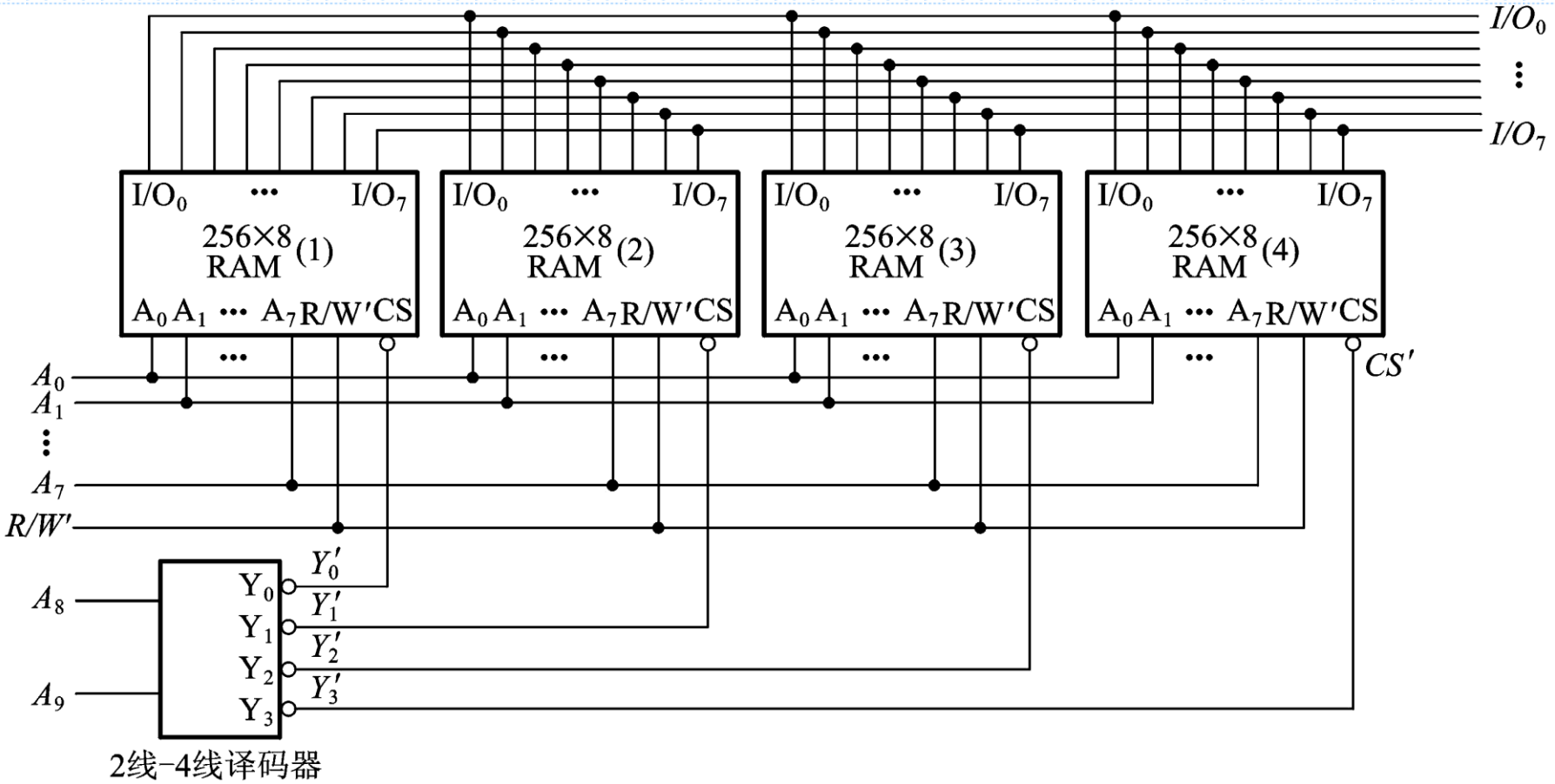
A_9	A_8	CS_1'	CS_2'	CS_3'	CS_4'
0	0	0	1	1	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	1	1	1	0

四片的地址分配就是：

00A_7～A_0，　01A_7～A_0，　10A_7～A_0，　11A_7～A_0

0～255　　256～511　　512～767　　768～1023

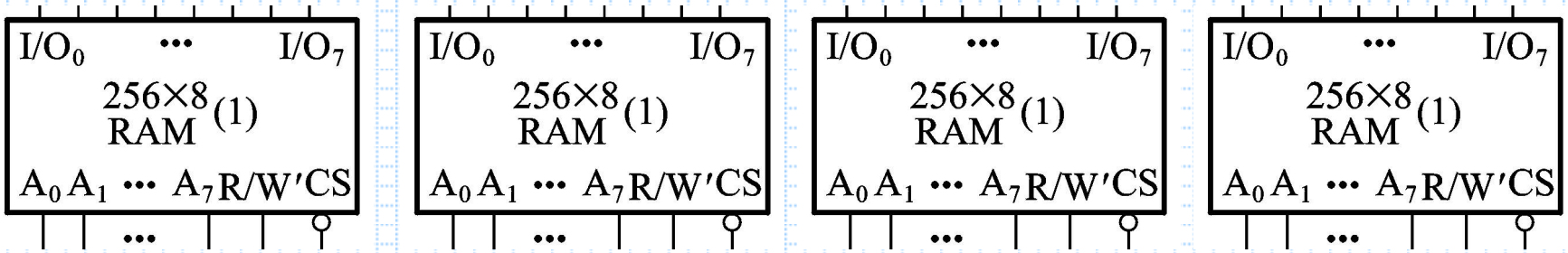


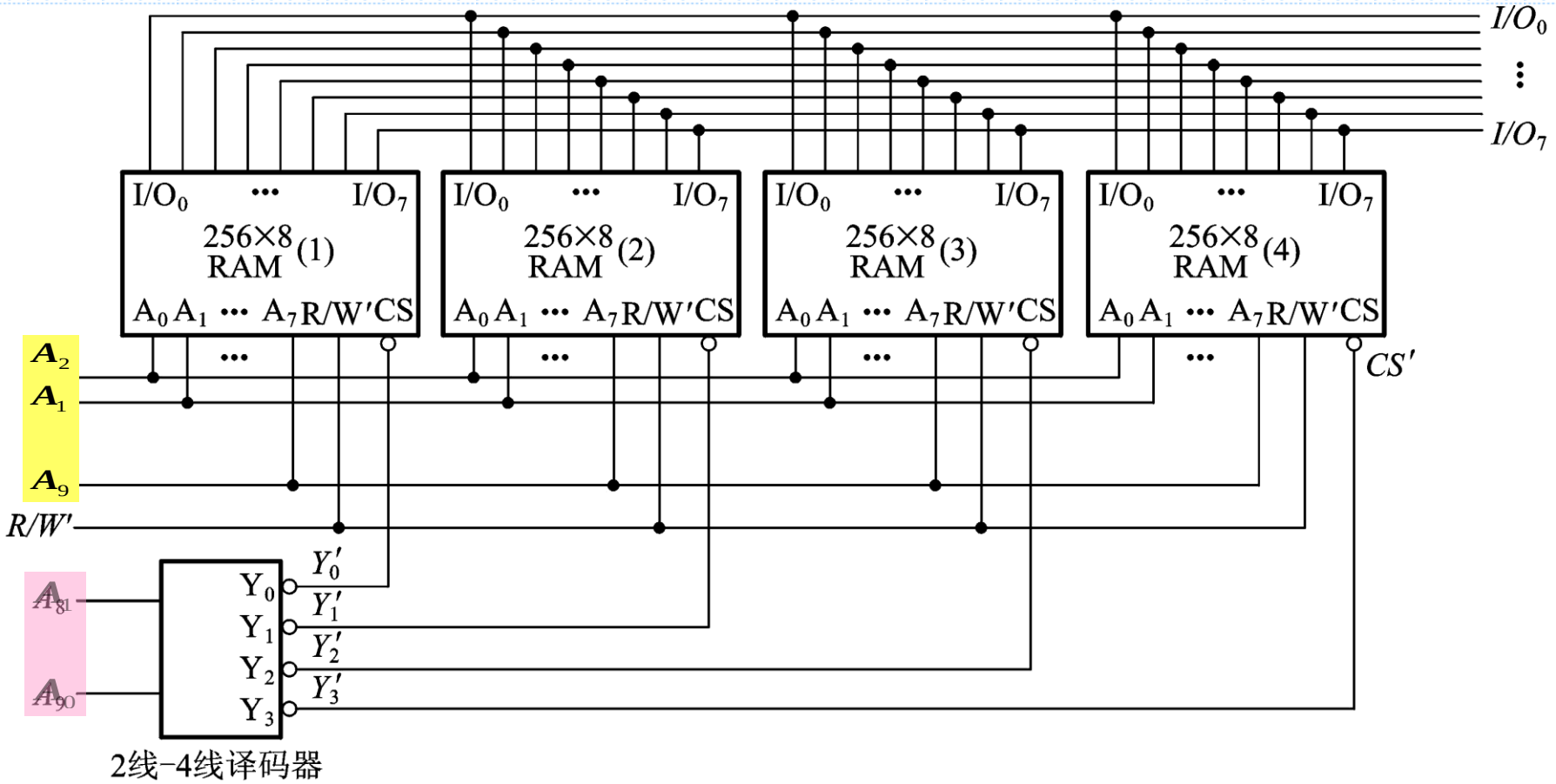

I/O0
I/O7
256×8 RAM (1)
256×8 RAM (2)
256×8 RAM (3)
256×8 RAM (4)
A0 A1 ··· A7 R/W′ CS
A0
A1
A7
R/W′
CS′
A8
A9
Y0
Y1
Y2
Y3
Y0′
Y1′
Y2′
Y3′
2线-4线译码器

A_1	A_0	CS'_1	CS'_2	CS'_3	CS'_4
0	0	0	1	1	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	1	1	1	0

四片的地址分配就是：

$A_9 \sim A_2 00$，　$A_9 \sim A_2 01$，　$A_9 \sim A_2 10$，　$A_9 \sim A_2 11$



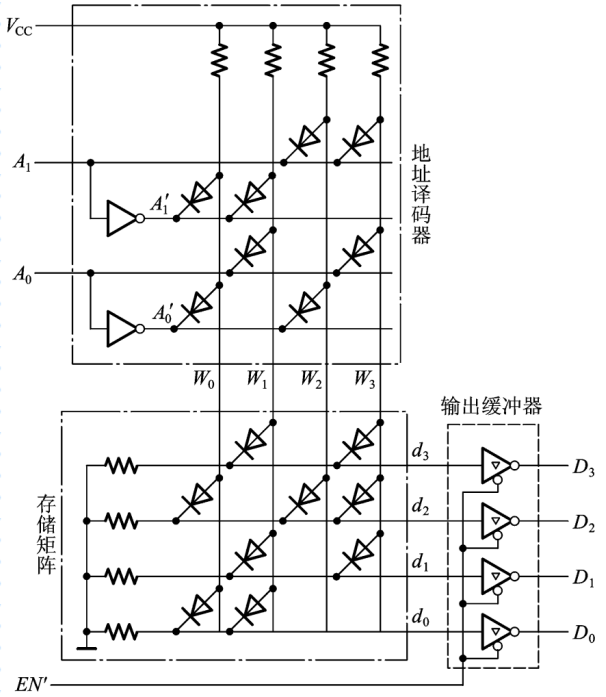

I/O0
I/O7
256×8 RAM (1)
256×8 RAM (2)
256×8 RAM (3)
256×8 RAM (4)
I/O0 ··· I/O7
A0 A1 ··· A7 R/W′ CS
A2
A1
A9
R/W′
CS′
A81
A90
Y0
Y1
Y2
Y3
Y0′
Y1′
Y2′
Y3′
2线-4线译码器

7.5 用存储器实现组合逻辑函数

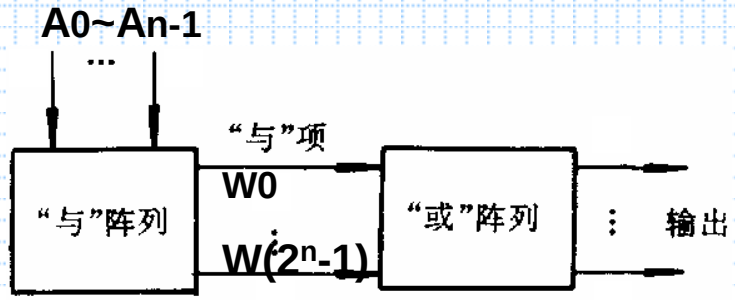
一、基本原理

从ROM的数据表可见：

若以地址线为输入变量，则数据线即为一组关于地址变量的逻辑函数



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

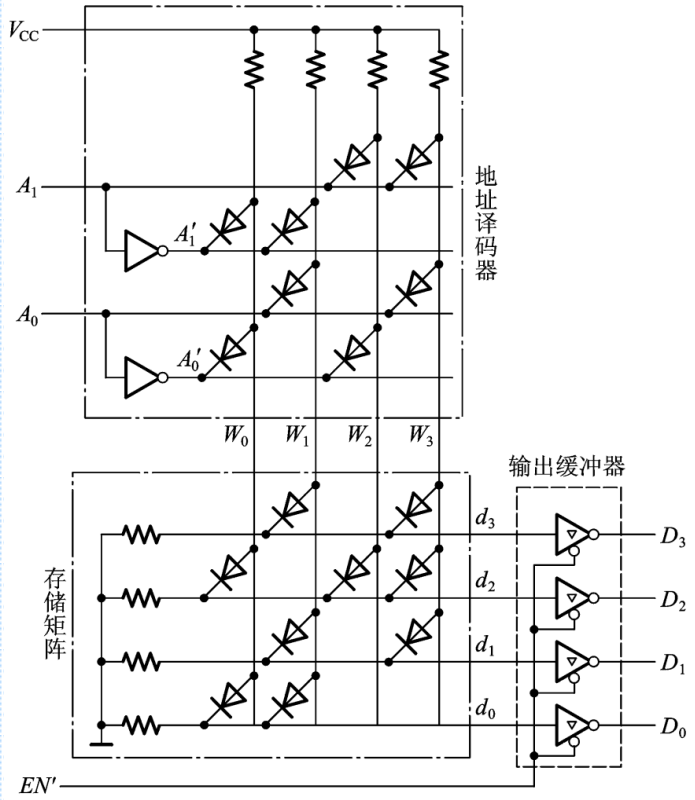


7.5 用存储器实现组合逻辑函数

一、基本原理

从ROM的数据表可见：

若以地址线为输入变量，则数据线即为一组关于地址变量的逻辑函数



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

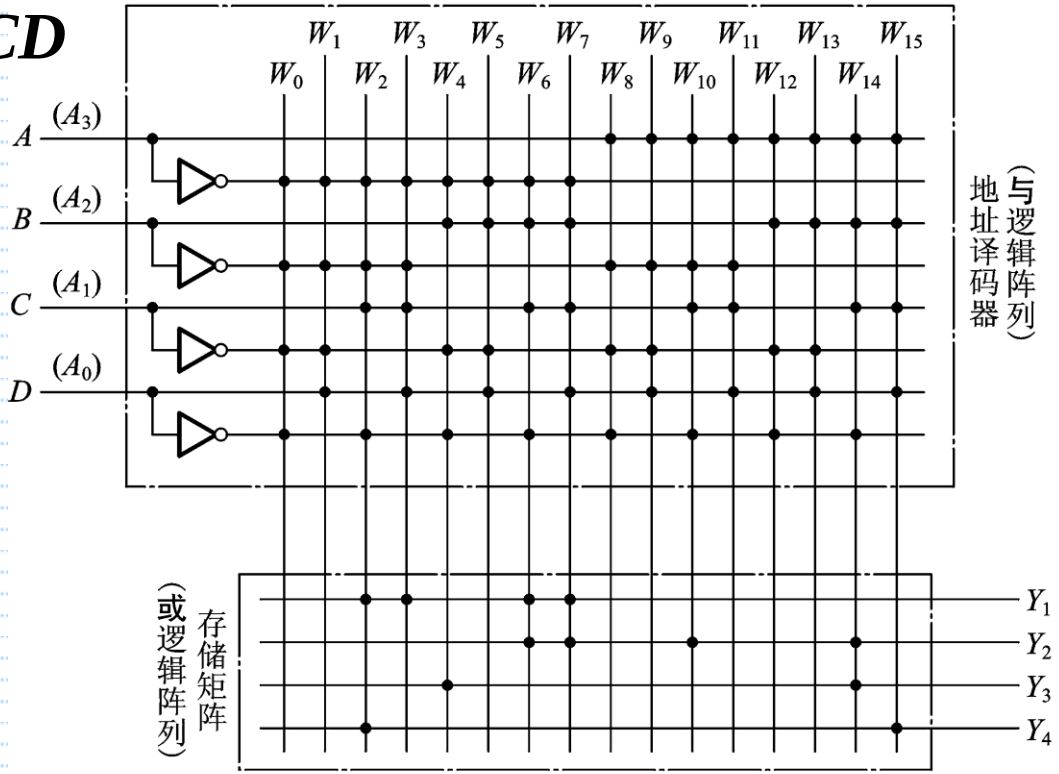
地　址		数　据			
A_1	A_0	D_3	D_2	D_1	D_0
0	0	0	1	0	1
0	1	1	0	1	1
1	0	0	1	0	0
1	1	1	1	1	0

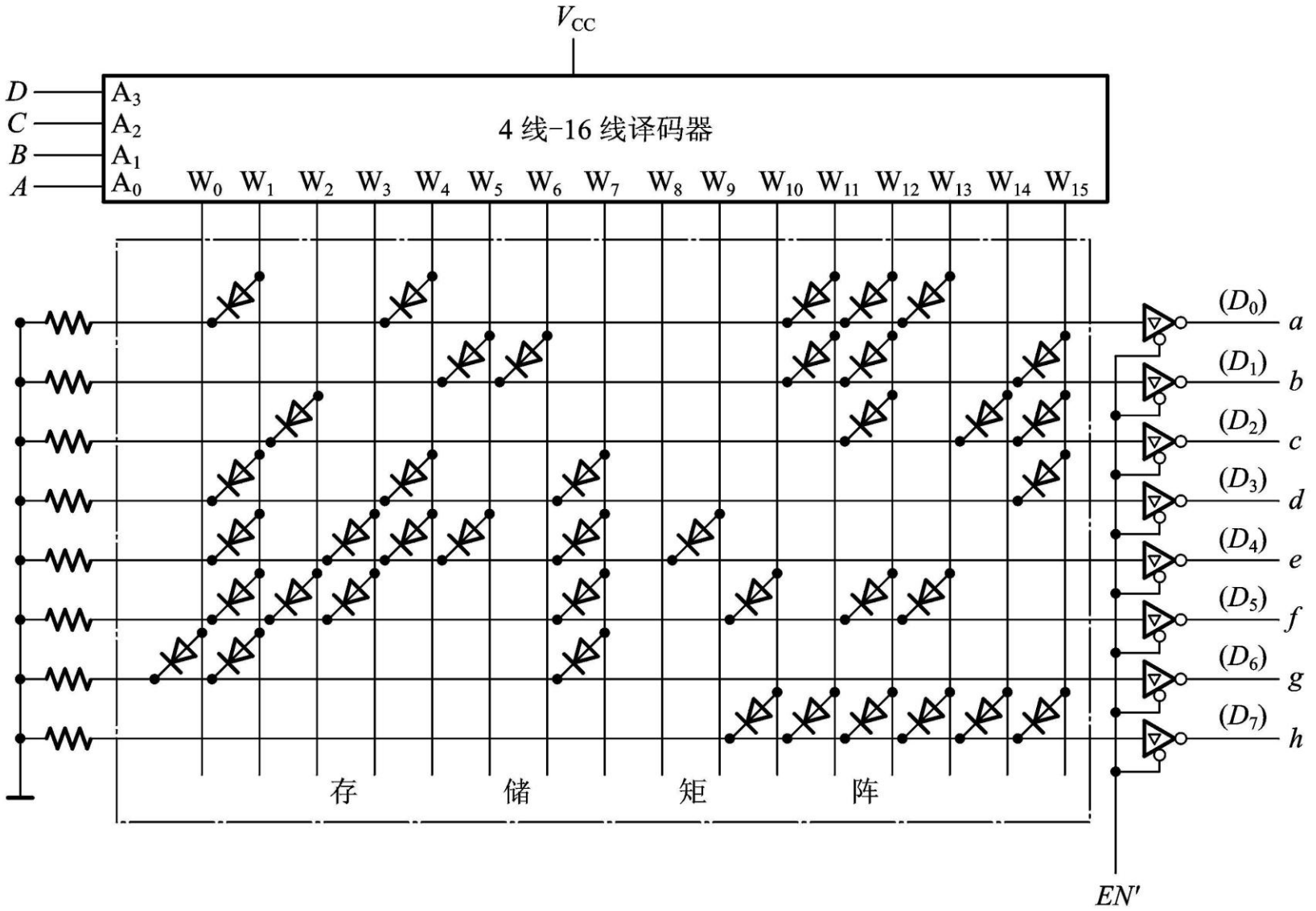
二、举例

用*ROM*产生：

$$\begin{cases} Y_1 = A'BC + A'B'C \\ Y_2 = AB'CD' + BCD' + A'BCD \\ Y_3 = ABCD' + A'BC'D' \\ Y_4 = A'B'CD' + ABCD \end{cases}$$

$$\begin{cases} Y_1 = \sum m(2,3,6,7) \\ Y_2 = \sum m(6,7,10,14) \\ Y_3 = \sum m(4,14) \\ Y_4 = \sum m(2,15) \end{cases}$$




V_{CC}
D
C
B
A
A_3
A_2
A_1
A_0
4 线-16 线译码器
W_0 W_1 W_2 W_3 W_4 W_5 W_6 W_7 W_8 W_9 W_{10} W_{11} W_{12} W_{13} W_{14} W_{15}
(D_0) a
(D_1) b
(D_2) c
(D_3) d
(D_4) e
(D_5) f
(D_6) g
(D_7) h
存 储 矩 阵
EN'