

第5章 可编程逻辑器件

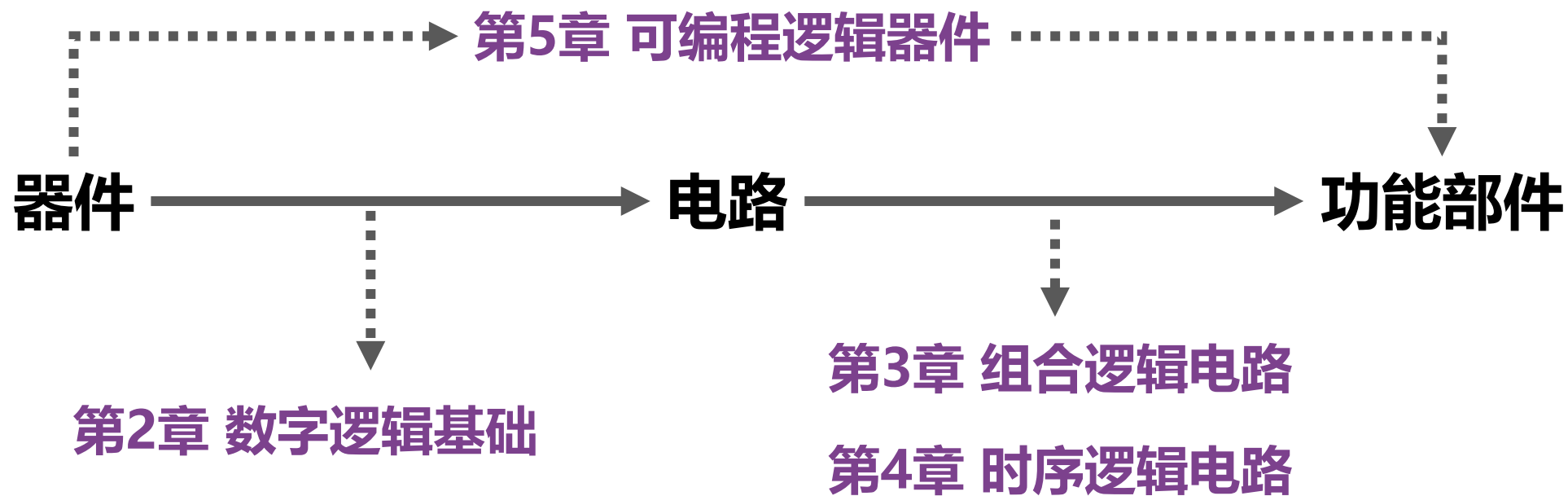
第一讲 可编程逻辑器件概述

第二讲 存储器阵列

第三讲 FPGA和ASIC

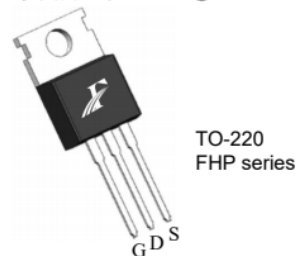


第5章 可编程逻辑器件

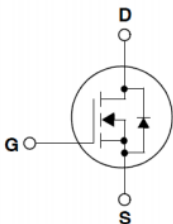


MOS器件

封装形式 Package

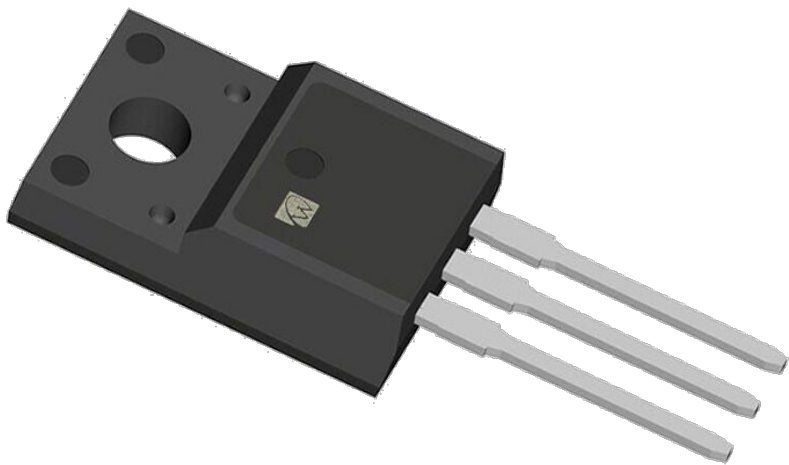


等效电路 Equivalent Circuit



绝对最大额定值 ABSOLUTE RATINGS (Tc=25°C)

项目 Parameter	符号 Symbol	数值 Value	单位 Unit
最高漏极-源极直流电压 Drain-Source Voltage	V _{DSS}	85	V
连续漏极电流* Drain Current -continuous *	I _D (T _c =25°C), Silicon Limited	147	A
	I _D (T _c =25°C), Package Limited	120	A
	I _D (T _c =100°C)	93	A
最大脉冲漏极电流 (注 1) Drain Current - pulse (note 1)	I _{DM}	480	A
最高栅源电压 Gate-Source Voltage	V _{GS}	±20	V
单脉冲雪崩能量 (注 2) Single Pulsed Avalanche Energy (note 2)	E _{AS}	364	mJ
雪崩电流 (注 1) Avalanche Current (note 1)	I _{AR}	27	A
重复雪崩能量 (注 1) Repetitive Avalanche Current (note 1)	E _{AR}	25	mJ
二极管反向恢复最大电压变化速率 (注 3) Peak Diode Recovery dv/dt (note 3)	dv/dt	5.0	V/ns
耗散功率 Power Dissipation	P _D (T _C =25°C)	184	W
	-Derate above 25°C	1.47	W/°C



固定逻辑 器件/芯片

概述

SM74HC138 是一款高速 CMOS 器件，引脚兼容低功耗肖特基 TTL（LSTTL）系列。

SM74HC138 有三个地址数据输入端（A0、A1、A2）和八个有效译码为低的输出端（ $\overline{Y0} - \overline{Y7}$ ）；

SM74HC138 有三个使能控制端（ $\overline{E_1}$ 、 $\overline{E_2}$ 、 E_3 ），当 $\overline{E_1}$ 、 $\overline{E_2}$ 为低电平且 E_3 为高电平时，八个译码输出端才有译码输出，否则八个译码输出端将全为高。

SM74HC138 通常应用于单个三地址数据输入八译码输出的 3-8 译码器，也可根据使能信号特点用两个 SM74HC138 实现四地址数据输入和 16 译码输出的 4-16 译码器，应用中未使用的使能端要处在译码有效输出使能电平状态。

SM74HC138 主要应用于消费类电子产品。

特性说明

- ◆ 采用 CMOS 工艺
- ◆ 低功耗
- ◆ 工作电压：3.0V—5.0V
- ◆ 封装形式：SOP16、TSSOP16、QFN16(3*3)

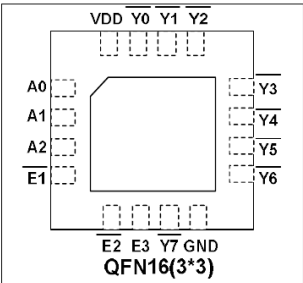
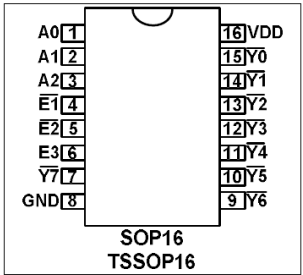
应用领域

- ◆ 适用于数字电路中的 3—8 译码功能

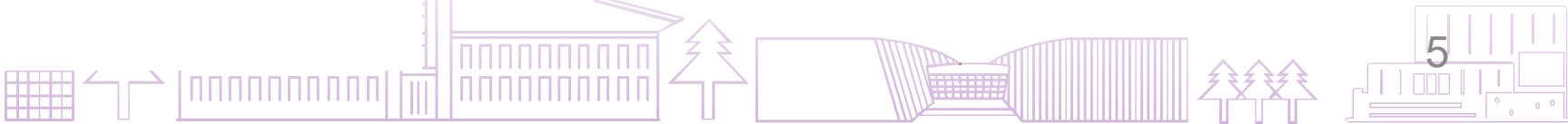
封装信息

产品名称	封装形式	塑封体尺寸 (mm)	脚间距 (mm)
SM74HC138D	SOP16	10.0*3.94*1.45	1.27
SM138TS	TSSOP16	5.0*6.4*0.9	0.635
MW138N	QFN16(3*3)	3*3*0.85	0.5

管脚定义



3-8译码器芯片74HC138



可编程逻辑器件 (PLD)

```
module ztj(clk,reset,state_in,com_out);
input clk,reset;
input state_in;
output [3:0] com_out;
reg[3:0] com_out;
parameter s0=0,s1=1,s2=2,s3=3;
reg[3:0] c_st,next_st;
always@(posedge clk or negedge reset)
begin
if(!reset) c_st<=s0;
else c_st<=next_st;
end
always@(c_st or state_in)
begin
case(c_st)
s0:begin
com_out<=4'b0111;
if(state_in==1'b0)
next_st<=s1;
else next_st<=s3;
end
s1:begin
```

现场可编程门阵列 (FPGA)



第5章 可编程逻辑器件

第一讲 可编程逻辑器件概述

第二讲 存储器阵列

第三讲 FPGA和ASIC



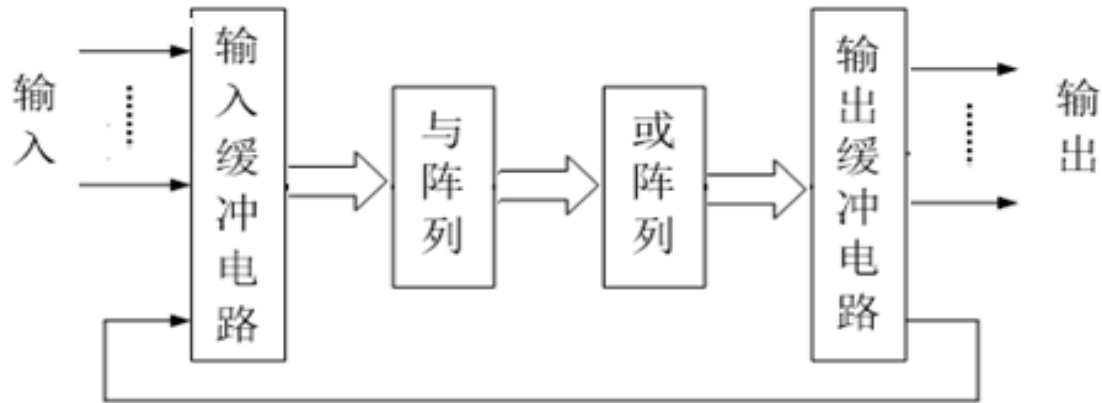
可编程逻辑器件

- 固定逻辑标准芯片曾被广泛使用，但**功能单一**，不能随电路设计的需求而任意改变，逐渐被**可编程逻辑器件**（Programmable Logic Device, PLD）取代
- PLD是一种用于实现逻辑电路的**通用器件**，包含多个逻辑单元，可根据需要通过**编程开关**连接进行编程，以构成不同功能的逻辑电路
- PLD的结构主要由**与阵列**和**或阵列**构成

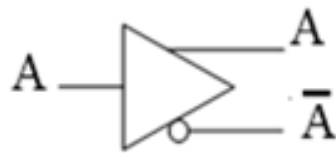


PLD器件

PLD结构框图：



互补缓冲器

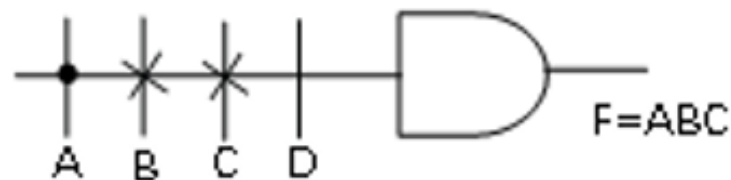


阵列连线

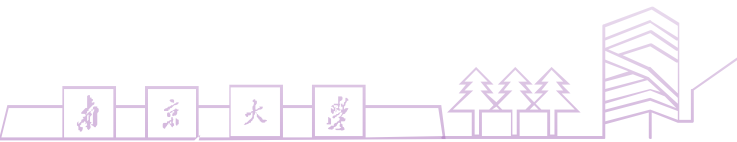
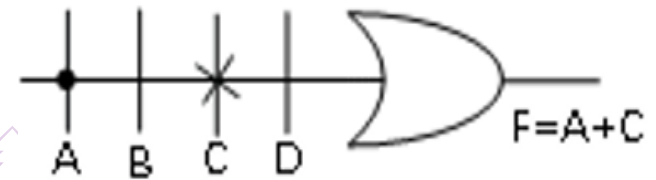


PLD中基本电路符号：

与阵列表示

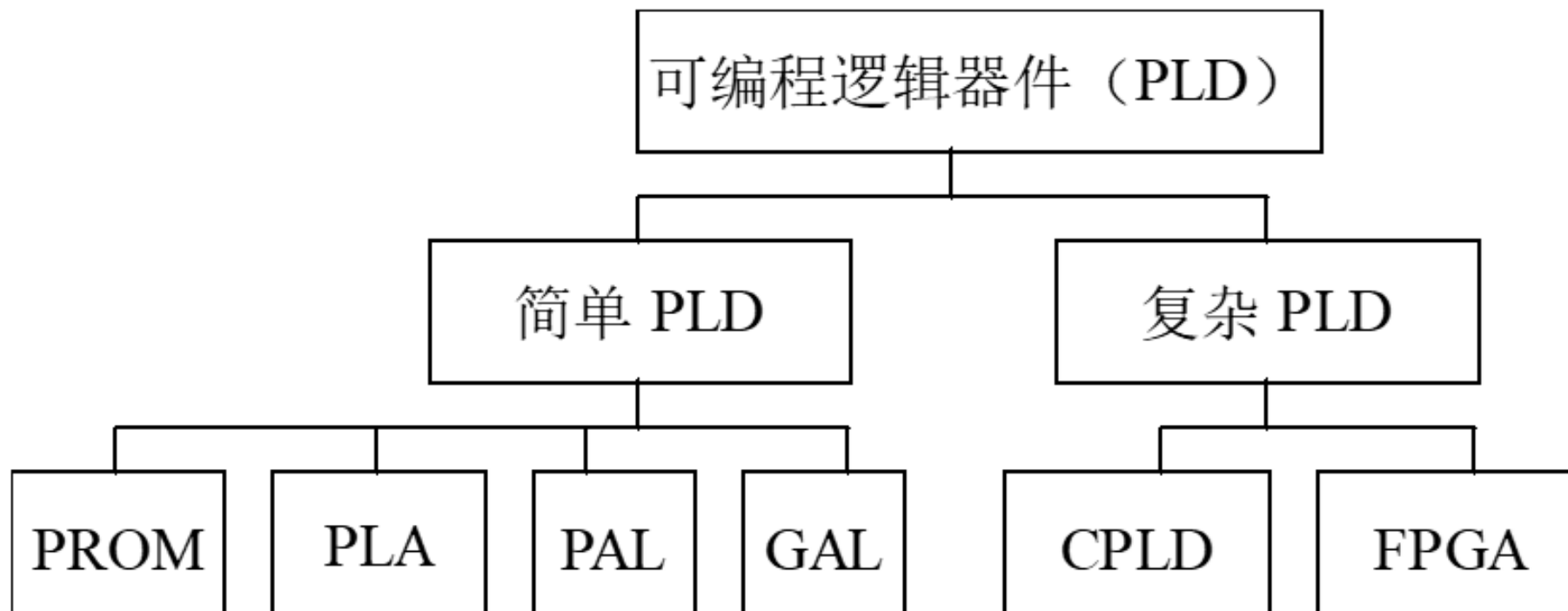


或阵列表示



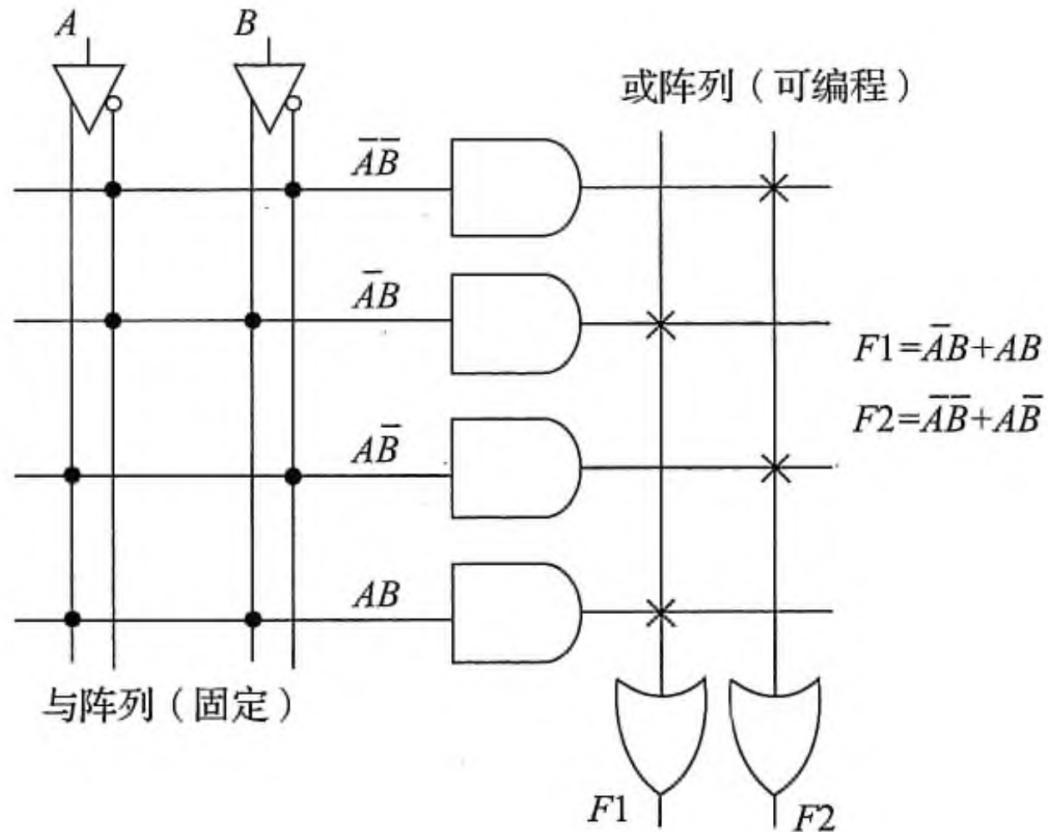
PLD器件

PLD分类



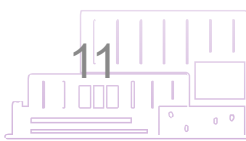
PLD器件

可编程只读存储器（Programmable Read Only Memory, PROM）
是一种与阵列固定、或阵列可编程的简单PLD。



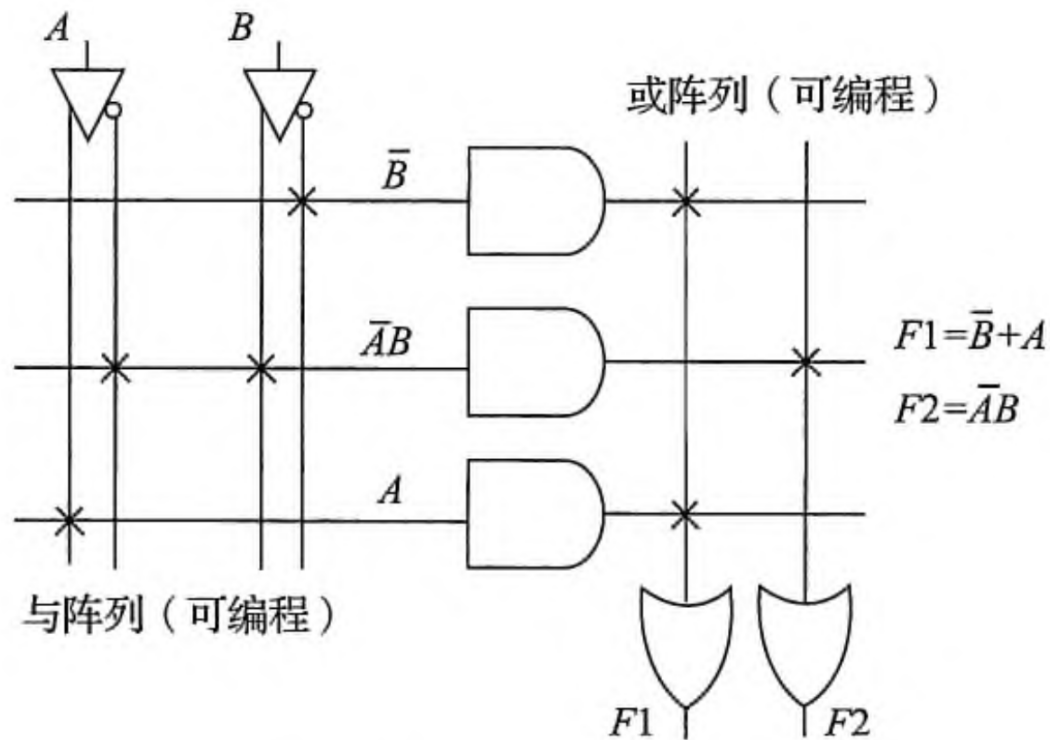
任何逻辑函数转换成标准与-或表达式后，可用PROM来实现

与阵列的水平线输出对应标准与-或表达式中的标准乘积项，即最小项。



PLD器件

可编程逻辑阵列（Programmable Logic Array, PLA）是一种与阵列、或阵列都可编程的逻辑阵列。

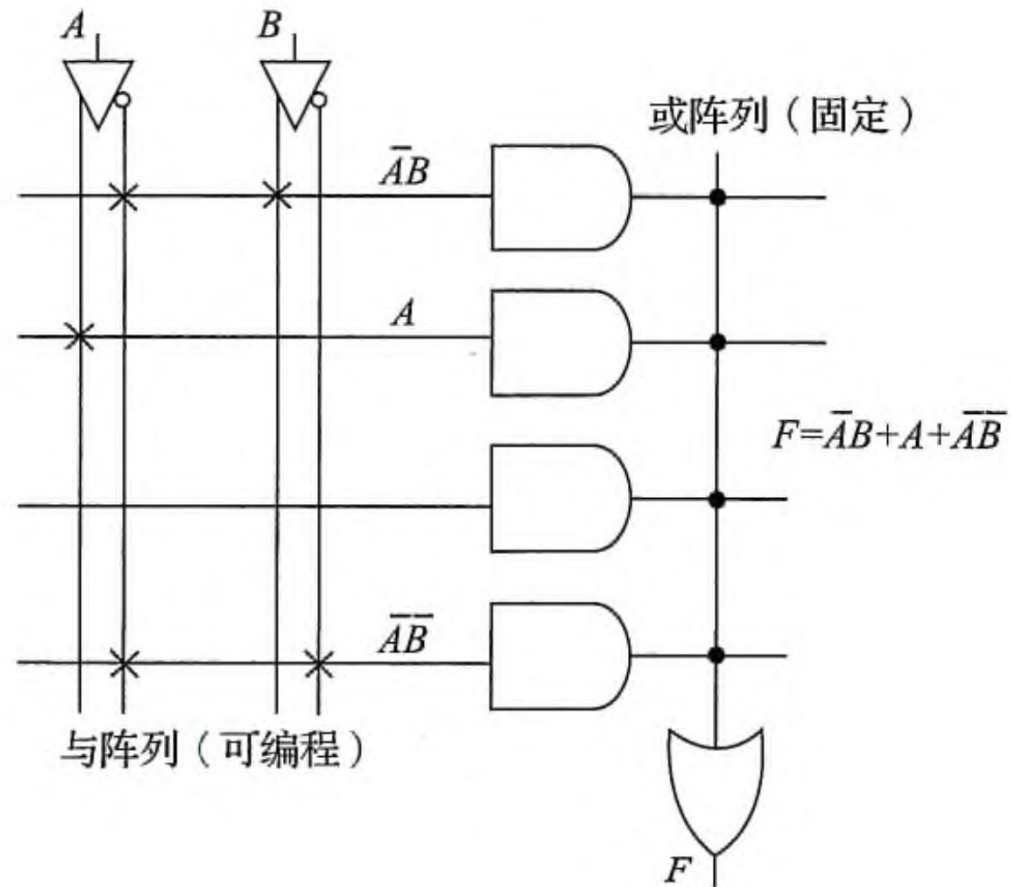


无须像PROM那样将逻辑函数转换成标准与-或表达式，而只要化简成最简与-或表达式即可，可节省编程资源。



PLD器件

可编程阵列逻辑（Programmable Array Logic, PAL）是一种与阵列可编程、或阵列固定的逻辑阵列。



PLD器件

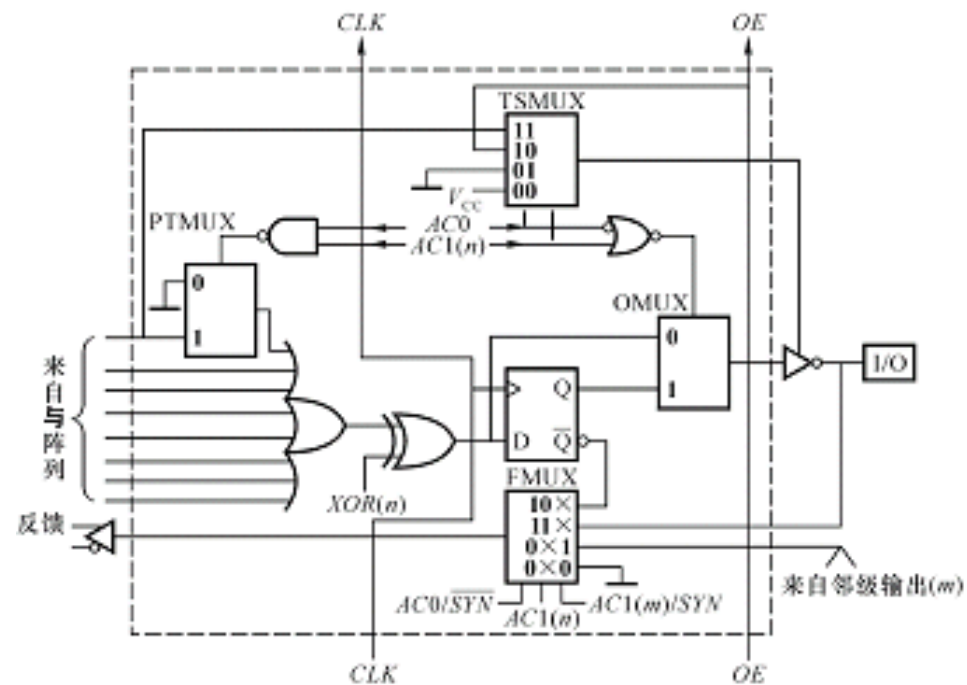
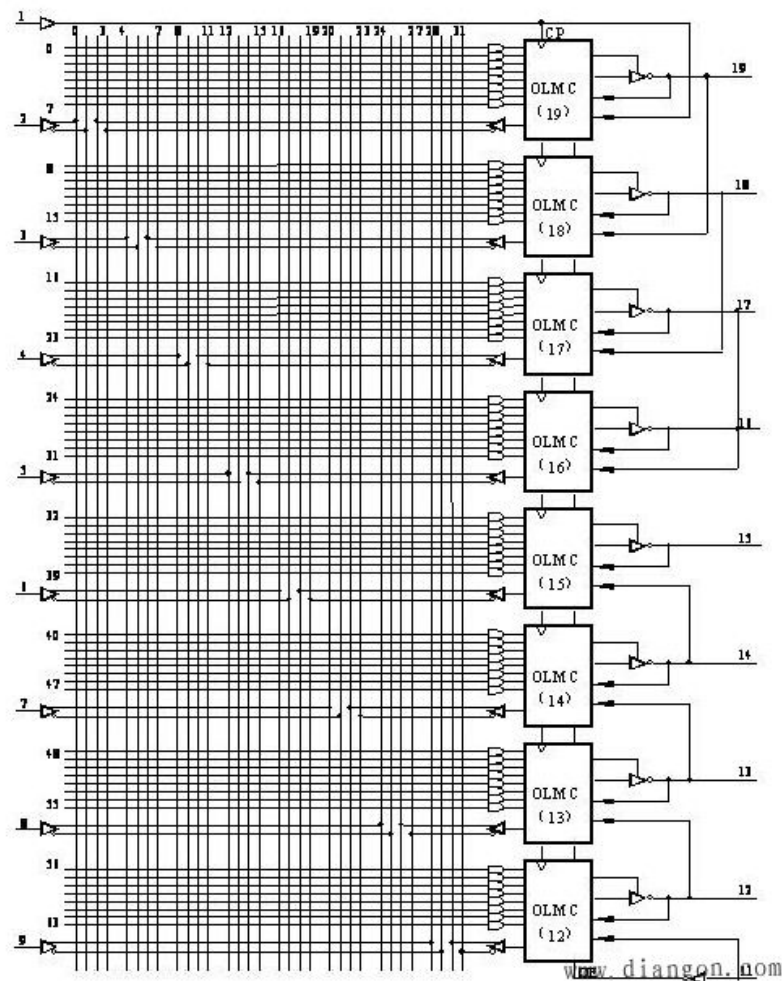
通用阵列逻辑（Generic Array Logic, GAL）

- PAL的逻辑结构简单，或阵列固定，灵活性差。LATTICE公司推出了一种新型的可编程逻辑器件—GAL。
- 它与PAL最大的差别是其输出逻辑宏单元（Output Logic Macro Cell, OLMC）可以编程定义。
- OLMC设置成不同的工作状态，可适用不同的功能需求。
- GAL具有电可擦写、可重复编程和设置加密位等特点。



PLD器件

通用阵列逻辑 (Generic Array Logic, GAL)



OLMC电路*



PLD器件

复杂可编程逻辑器件（Complex PLD, CPLD）主要包括：

- 逻辑阵列块（Logic Array Block, LAB）
- I/O控制块：用于和芯片的I/O引脚互连
- 可编程互联阵列（PIA）
- 每个LAB由4~20个宏单元（Macrocell）构成。宏单元包括：
 - 可编程逻辑阵列
 - 乘积项选择矩阵
 - 可编程寄存器：可以编程实现D、JK或钟控SR触发器等。
- 宏单元有多种配置方式，也可级联使用
- PIA用于连接所有宏单元，并与芯片时钟、复位、使能等引脚连
- CPLD集成度远高于PAL和GAL
- CPLD通常提供带片内RAM/ROM的嵌入式存储器阵列



PLD器件

现场可编程门阵列（Field Programmable Gate Array, FPGA）是一种高集成度的复杂可编程逻辑器件，可通过EDA软件对其进行配置和编程，可反复擦写。

- 实现乘法器、寄存器、地址发生器等硬件电路
- 通过使用框图或者Verilog HDL来设计
- FPGA可无限地重新编程，加载一个新的设计方案只需几百毫秒



第5章 可编程逻辑器件

第一讲 可编程逻辑器件概述

第二讲 存储器阵列

第三讲 FPGA和ASIC



存储器阵列

- 存储器可用于存储数字电路中的数据。
 - 寄存器用来存储少量数据，速度更快
 - 存储器阵列用来存储大量数据，速度较寄存器慢
- 在CPLD和FPGA芯片中通常会提供片内存储器阵列
- 存储器阵列中每位数据对应一个记忆单元（cell），称为存储元



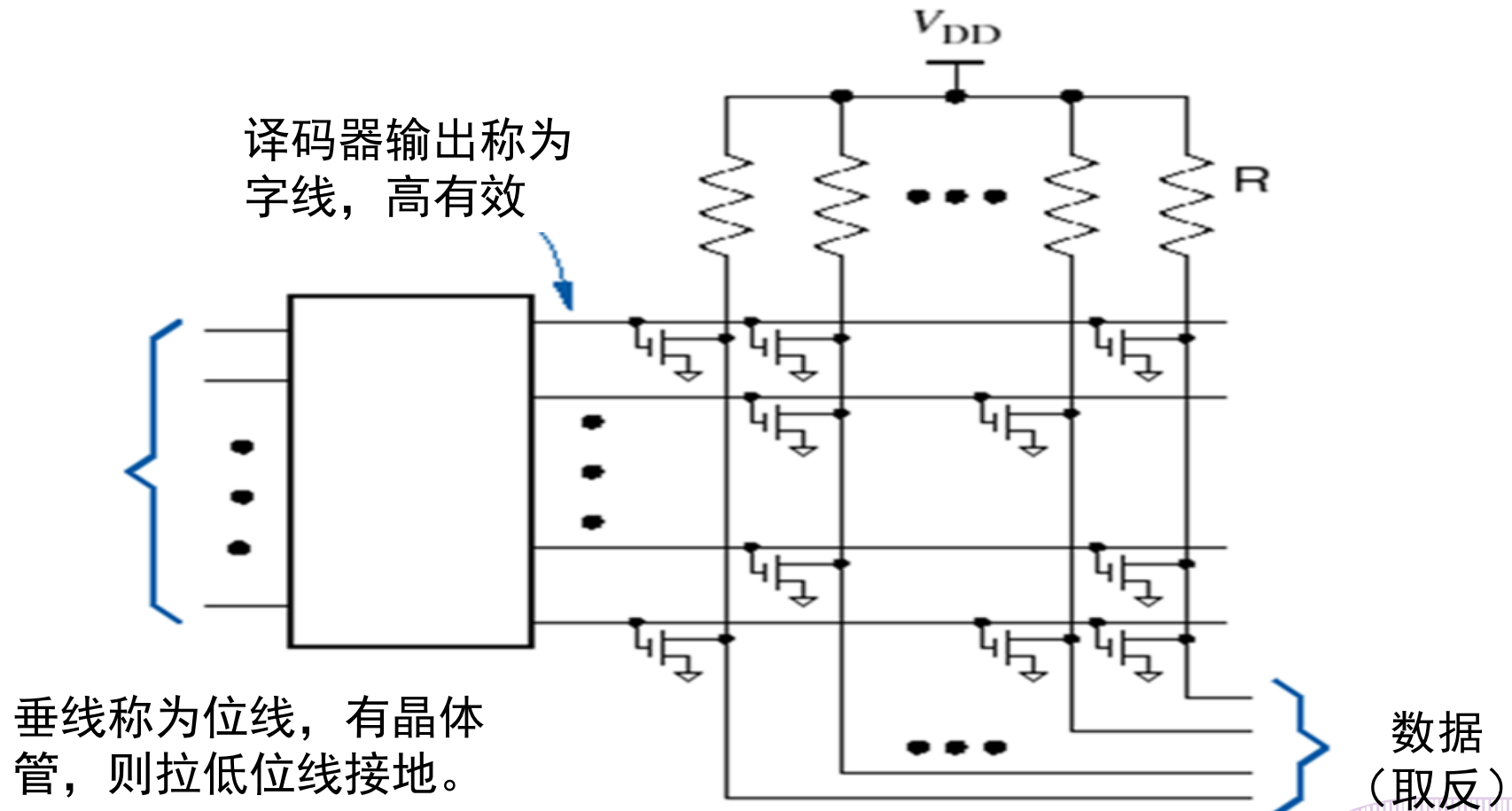
存储器阵列

- 按功能可分为：只读存储器 (Read-only Memory, ROM) 和随机存取存储器 (Random-access Memory, RAM)
 - ROM属于非易失性存储器，即使电源断电，ROM中存储的数据也不会消失。根据工艺的不同，分为：
 - 掩膜只读存储器MROM
 - 一次可编程只读存储器PROM
 - 光擦除可编程只读存储器EPROM
 - 电擦除可编程只读存储器EEPROM (E²PROM)
 - RAM属于易失性存储器，一旦电源断电，RAM中存储的数据就消失。
 - 静态RAM (Static RAM, SRAM)
 - 动态RAM (Dynamic RAM, DRAM)



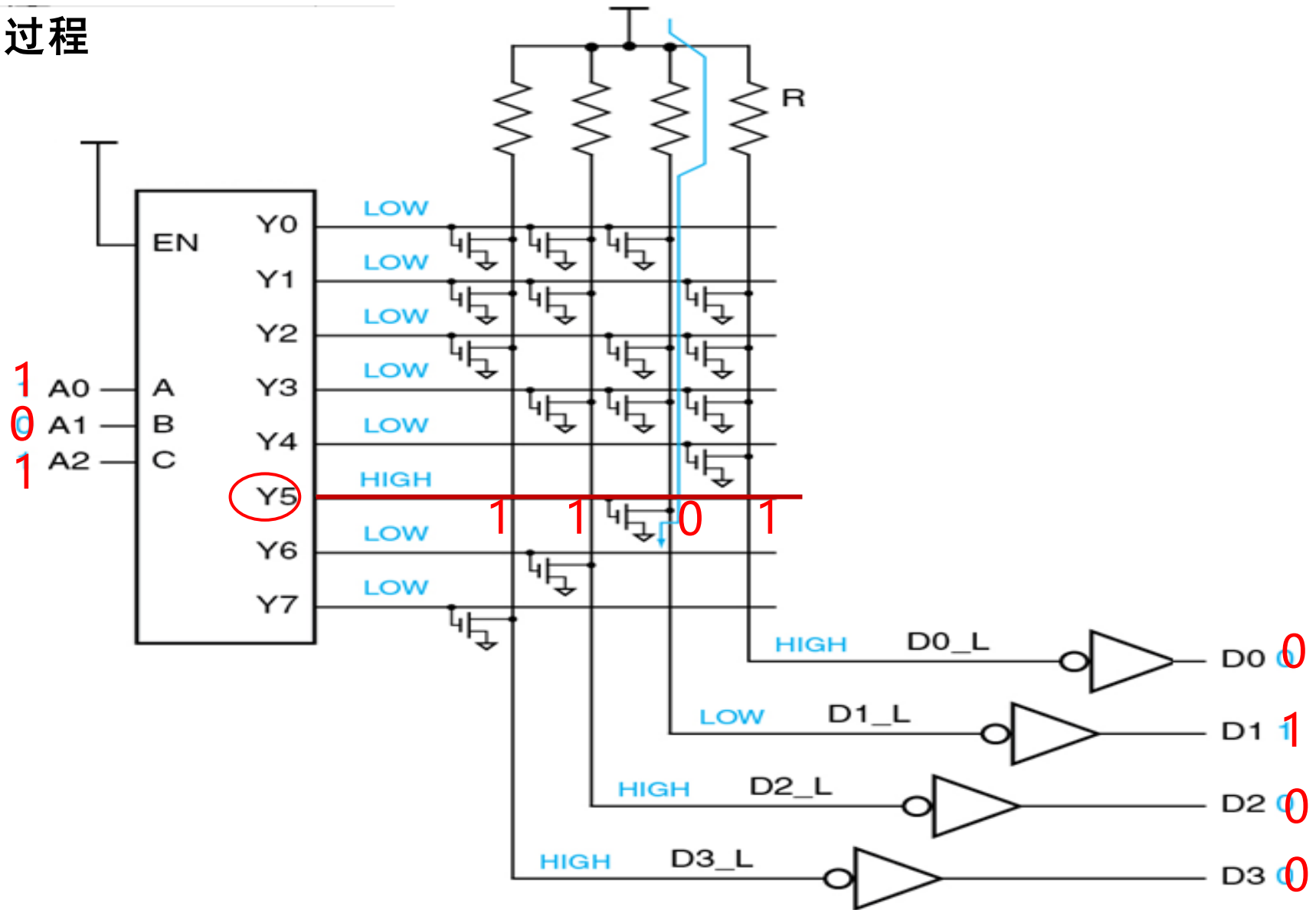
存储器阵列

- ROM存储阵列根据MOS晶体管的有无来区分存储0和1
- 不同类型的ROM，主要区别在于MOS晶体管的特性不同



存储器阵列

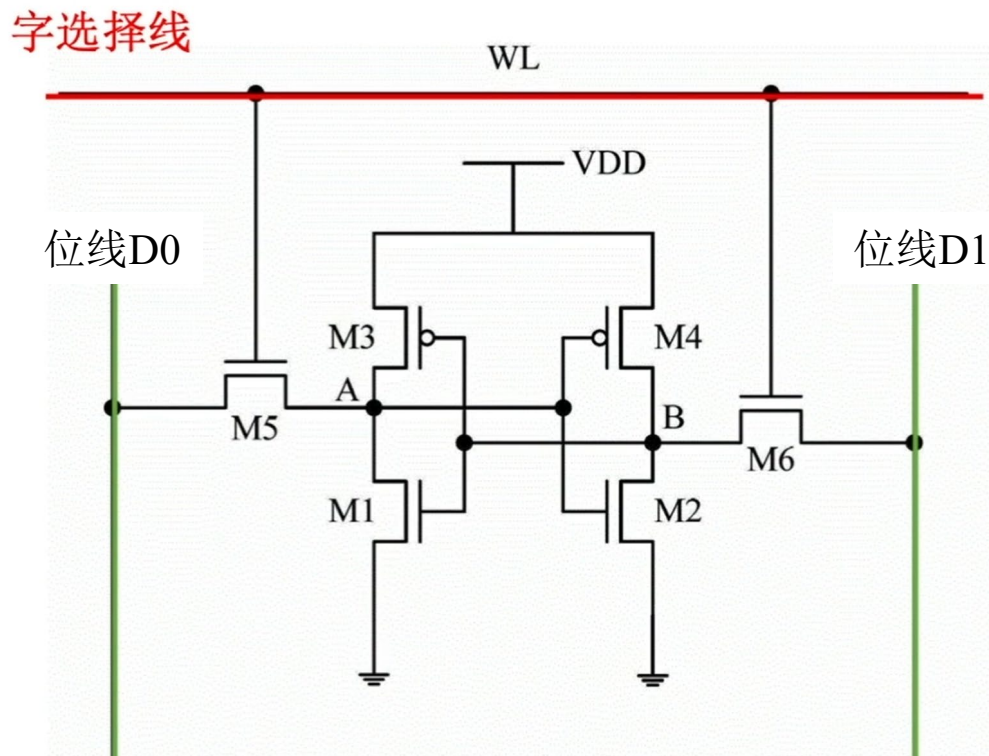
➤ ROM读取过程



存储器阵列

静态存储器SRAM：只要保持电源，存储单元中存放数据就保持不变

- 读写速度快、价格高、功耗大、集成度低，无需刷新。
- 存储单元使用6个MOS晶体管来实现



M1+M3、M2+M4分别构成了反相器，构成了双稳态元件，M5和M6为门控管

字选择线W为低电平时，M5与M6截止，双稳态元件与外界隔离，保持状态不变。

字选择线W为高电平时，M5与M6导通，可以进行状态读写

存储器阵列

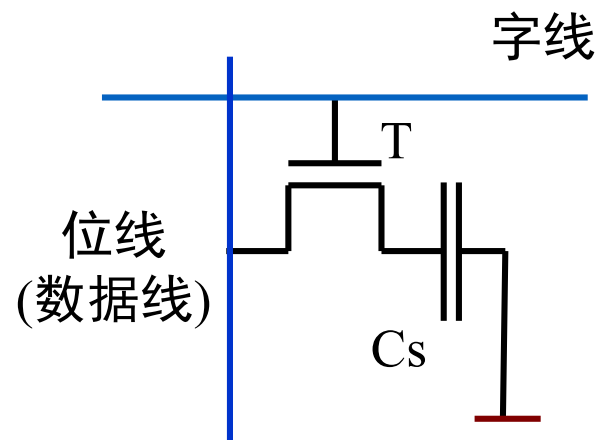
动态存储器DRAM：单MOS管，电容上存有大量电荷为1，否则0

读写原理：字线上加高电平，使T管导通。

写“0”时，数据线加低电平，使 C_s 上电荷对数据线放电；

写“1”时，数据线加高电平，使数据线对 C_s 充电；

读出时，数据线上有一读出电压。它与 C_s 上电荷量成正比。



优点：电路元件少，功耗小，集成度高，用于构建主存储器

缺点：速度慢，是破坏性读出（需读后再生），需定时刷新

刷新：DRAM的一个重要特点是，数据以电荷的形式保存在电容中，电容的放电使得电荷通常只能维持几十个毫秒左右，相当于1M个时钟周期左右，因此要定期进行刷新（读出后重新写回），按行进行（所有芯片中的同一行一起进行），刷新操作所需时间通常只占1%~2%左右。



第5章 可编程逻辑器件

第一讲 可编程逻辑器件概述

第二讲 存储器阵列

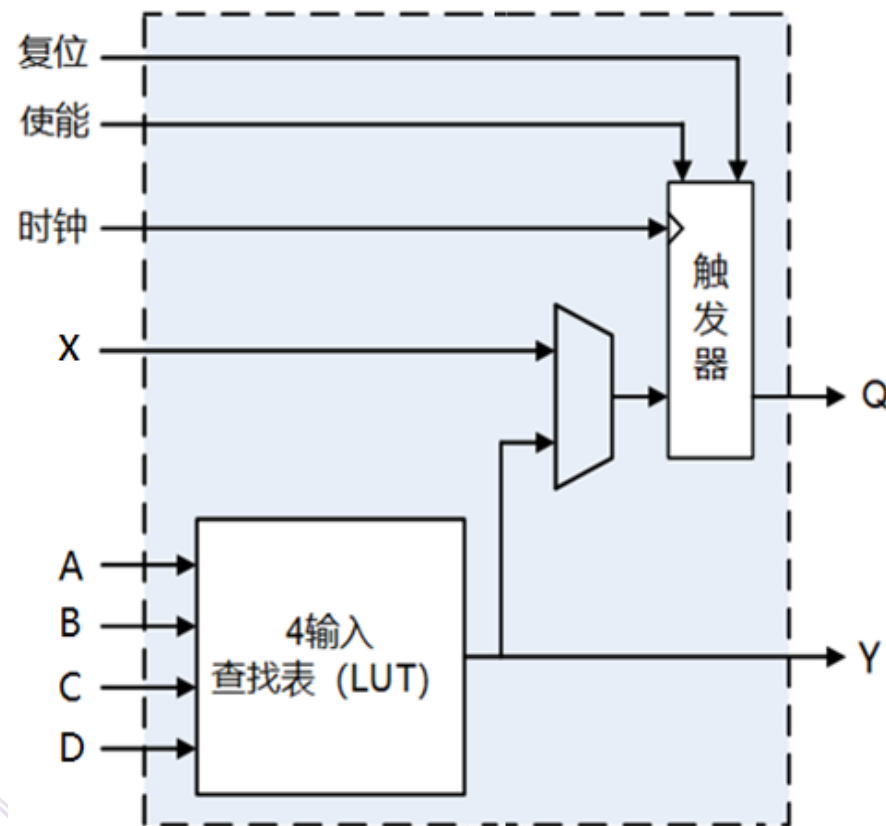
第三讲 FPGA和ASIC



FPGA设计概述

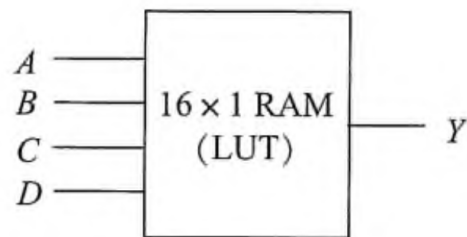
现场可编程门阵列（Field Programmable Gate Array, FPGA） 是一种高集成度的复杂可编程逻辑器件，可通过EDA软件对其进行配置和编程，可反复擦写。

- FPGA内部包含大量可配置逻辑块CLB，它由若干查找表（Look-Up Table, LUT）及多路选择器、进位链、触发器FF等附加逻辑组成。
- 可对CLB进行不同配置。如右图：
可对MUX编程配置为输出是LUT实现的组合逻辑电路结果Y；也可配置为输出是时序逻辑电路结果Q。
- LUT本质上是一个RAM，多采用SRAM实现。

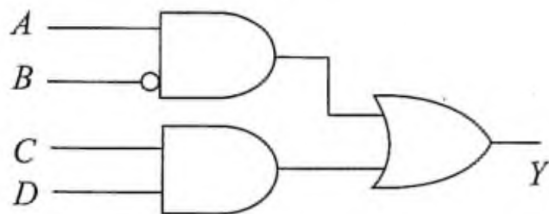


FPGA设计概述

- 函数发生器通过查找表LUT实现，其中的内容可编程配置
- LUT存储单元中存放函数输出值，用于实现一个小规模逻辑函数



a) 4输入LUT



$$Y = A\bar{B} + CD$$

b) LUT实现的逻辑功能

A	B	C	D	Y
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

c) 逻辑功能对应的真值表

0000	0
0001	0
0010	0
0011	1
0100	0
0101	0
0110	0
0111	1
1000	1
1001	1
1010	1
1011	1
1100	0
1101	0
1110	0
1111	1

d) RAM中存储的内容

专用集成电路ASIC

专用集成电路（Application-Specific Integrated Circuit, ASIC）

是一种应特定用户要求和特定电子系统的需要而设计、制造的集成电路。

- 全定制：设计者完成所有设计，速度更快
- 半定制：使用标准库里的标准逻辑单元（标准单元）
- FPGA和ASIC目前都是电子设计领域的主流产品。
- ASIC面向特定用户的需求，具有体积小、功耗低、可靠性高、性能高、保密性高、成本低等优点，一般用于**批量大的专用**产品中。
- FPGA可编程特性使其应用非常灵活，但芯片内部逻辑门的使用率大幅降低，导致功耗高、速度慢、资源冗余且价格昂贵，一般用于**小批量产品设计**中。



第5章 可编程逻辑器件

第一讲 可编程逻辑器件概述

第二讲 存储器阵列

第三讲 FPGA和ASIC



第5章 可编程逻辑器件

