

第2章 数字逻辑基础

第一讲 逻辑门和数字抽象

第二讲 布尔代数

第三讲 逻辑关系描述

第四讲 逻辑函数化简

第一讲 逻辑门和数字抽象

◆ 逻辑门

- 逻辑关系、真值表、逻辑门符号

◆ 数字抽象

- 模拟信号与数字信号
- 直流噪声容限

◆ CMOS晶体管

- PMOS和NMOS
- 常用CMOS门电路

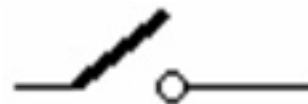
◆ CMOS电路电气特性

1.1 逻辑门

◆ 逻辑门电路(logic gate)是最基础的数字电路，具有**允许或禁止**信号传输的功能，也称为门电路。

- 一个或多个输入信号

- 一个输出信号：表明输入信号间的**逻辑关系**



开关

◆ 逻辑门都有自己特有的**图形符号**

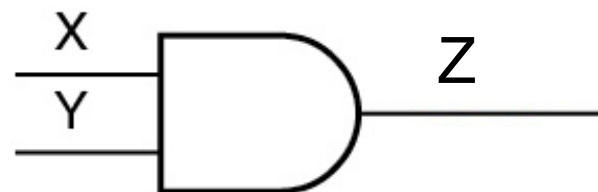
- 左边：输入信号

- 右边：输出信号

- 使用标识符来命名输入和输出信号

如X、Y、Z、INPUT 等

每根线是一个比特

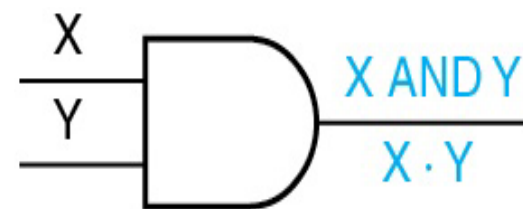


XYZ都是逻辑符号
(逻辑变量)

◆ 输入信号、输出信号称为逻辑变量（取值是**0**或**1**）

1.1 逻辑门

- ◆ 输入信号和输出信号之间的逻辑关系使用**真值表**或者**逻辑表达式**来描述。
- ◆ **逻辑表达式**就是用**逻辑运算符**来连接**逻辑变量**
- ◆ **真值表**是一个**二维表**
 - 表头左侧是输入信号，右侧是输出信号；
 - 按顺序列出**所有可能的**输入组合和该输入组合**对应的**输出信号值。
- ◆ 最基本的逻辑运算是**与、或、非**三种运算
 - 这三种运算可以表示任意组合逻辑关系。
 - 逻辑门分别称为与门、或门和非门，统称为基本逻辑门。

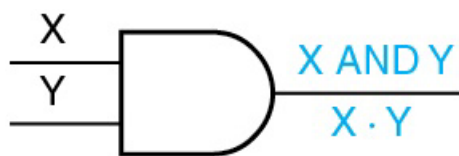


X	Y	X AND Y
0	0	0
0	1	0
1	0	0
1	1	1

1.1 基本逻辑门：与，或，非

◆基本逻辑门

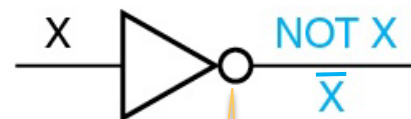
- **与门AND**：当且仅当所有输入信号为1时，输出信号才为1，运算符用乘点号 “ \cdot ” 表示，称为与运算或者逻辑乘运算。
- **或门OR**：只要有一个输入信号为1时，输出信号就为1。运算符用加号 “ $+$ ” 表示，称为或运算或者逻辑加运算。
- **非门NOT**：输出信号是输入信号的相反值，也称反相器。运算符用上横线 “ $\bar{}$ ” 表示，称为非运算或者取反运算。



X	Y	X AND Y
0	0	0
0	1	0
1	0	0
1	1	1



X	Y	X OR Y
0	0	0
0	1	1
1	0	1
1	1	1



X	NOT X
0	1
1	0

反相圈

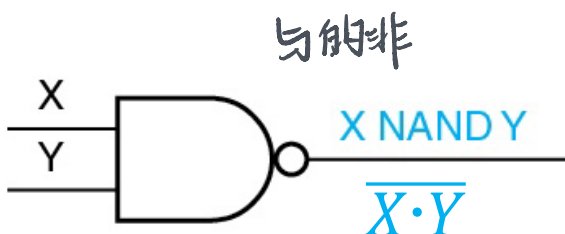
1.1 逻辑门：与非，或非

◆与非门 NAND

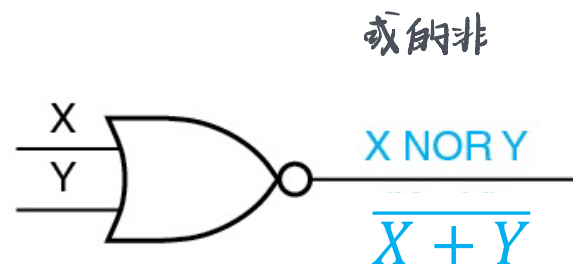
- 只要有一个输入信号为**0**，输出信号就为**1**。逻辑表达式用与运算加上横线来表示。

◆或非门 NOR

- 当且仅当所有输入信号为**0**时，输出信号才为**1**。逻辑表达式用或运算加上横线来表示。



X	Y	X NAND Y
0	0	1
0	1	1
1	0	1
1	1	0



X	Y	X NOR Y
0	0	1
0	1	0
1	0	0
1	1	0

1.1 逻辑门：异或，同或

◆异或门XOR

- 当两个输入不同时，输出为1。运算符用“ \oplus ”表示，逻辑表达式： $X \oplus Y = \bar{X} \cdot Y + X \cdot \bar{Y}$

◆同或门NXOR

- 当两个输入相同时，输出为1。也称为异或非门或等价关系门。运算符用“ \odot ”表示。 $X \odot Y = \bar{X} \cdot \bar{Y} + X \cdot Y$



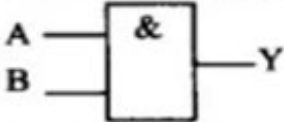
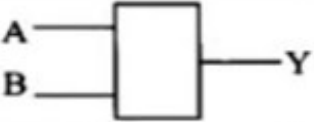
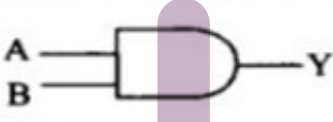
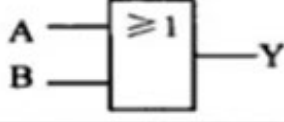
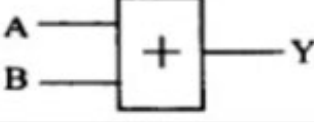
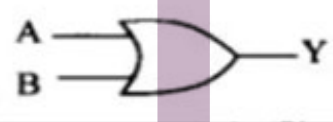
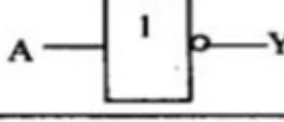
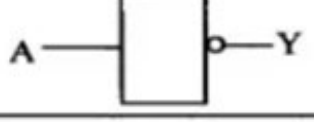
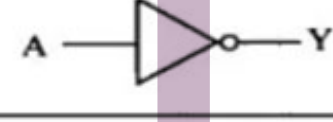
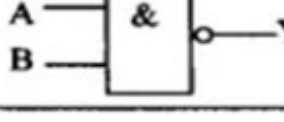
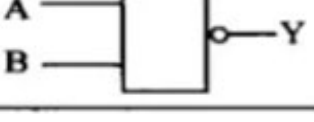

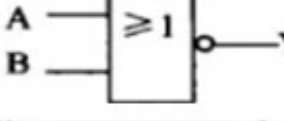
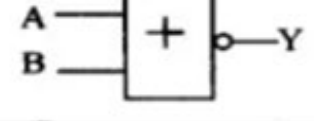
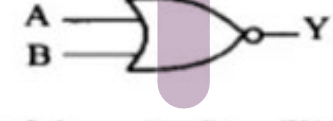
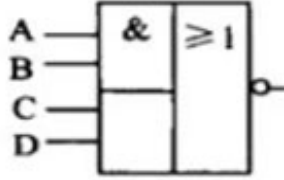
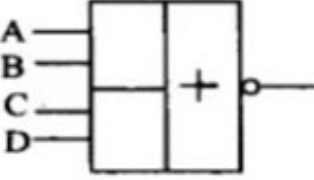
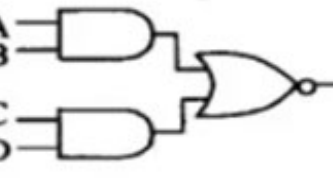
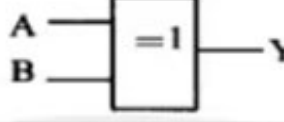

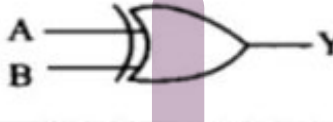
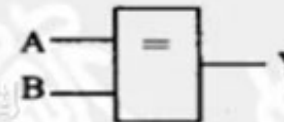
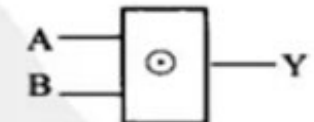

X	Y	$X \oplus Y$
0	0	0
0	1	1
1	0	1
1	1	0

(a)异或门



X	Y	$X \odot Y$
0	0	1
0	1	0
1	0	0
1	1	1

(b)同或门

名称	国标符号	曾用符号	国外流行符号
与			
或			
非			
与非			
或非			
与或非			
异或			
同或			

教材和
PPT中
都使用
国际流
行符号

有些出
版社的
教材使
用国标

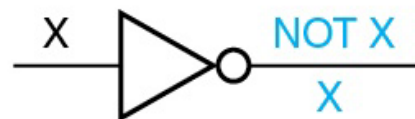
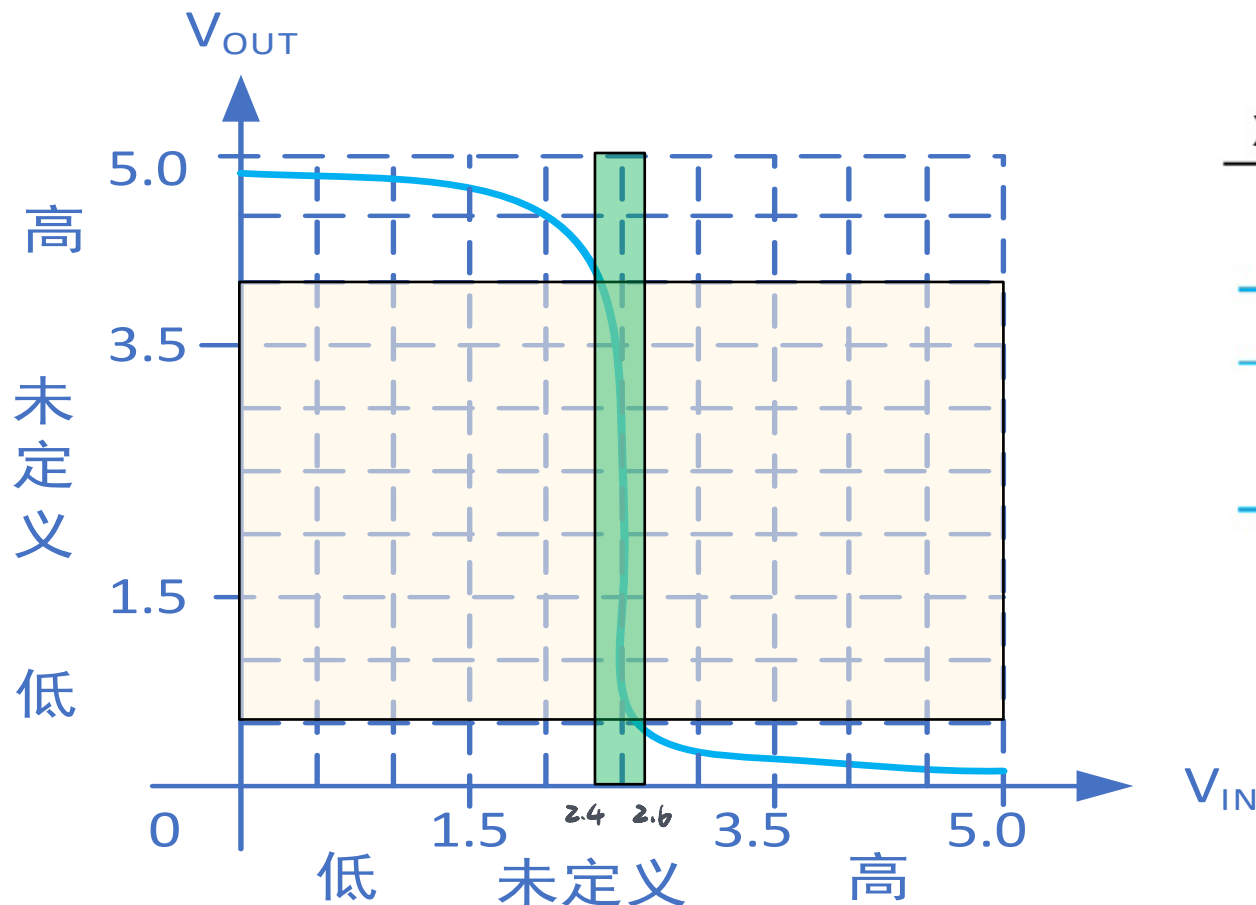
1.2 数字抽象

- ◆ **数字抽象（逻辑采样）**：将某个物理量的实际值集映射为两个子集，对应于两个**状态**或两个逻辑值**0**和**1**
- ◆ 0和1不表示数值的大小，而表示两种相反的状态。如电平高与低、电路导通与截止、灯亮与灭，开关的开与关，等
- ◆ 在数字系统中，将一定范围内的电压映射到两个状态：高态（high）和低态（low），并用0和1来表示
 - 设定**阈值范围/未定义区**。

逻辑值	正逻辑 Positive Logic	负逻辑 Negative Logic
0	低电平L	高电平H
1	高电平H	低电平L

1.2 数字抽象

- ◆ 由于受到**负载及噪声**的影响，输出电压可能不能保持稳定，但它必须能被其他逻辑门的输入端**准确识别**。



X	NOT X
0	1
1	0

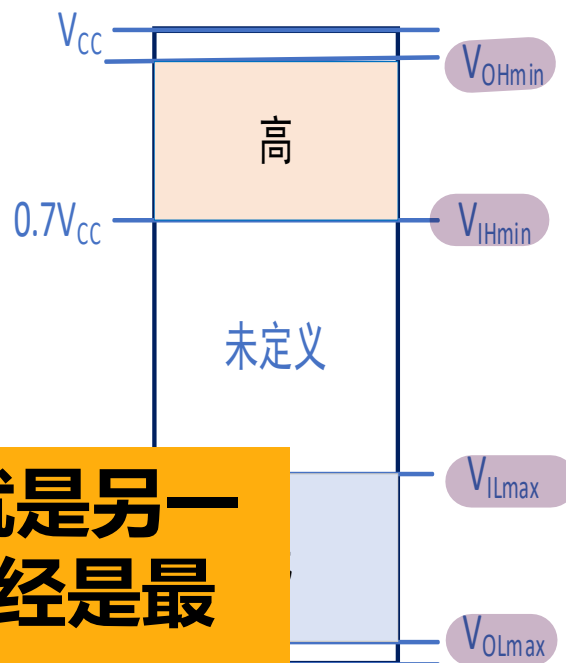
当输入电压在2.4
~ 2.6V之间时，
输出处于不确定
(未定义) 状态

非门典型的输入-输出传输特性图

1.2 数字抽象

◆ 输入电压主要由晶体管（CMOS）的开关阈值电压决定，而输出电压则主要由晶体管导通时的电阻决定。

- V_{IHmin} : 确保能被识别为高态的最小输入电压值。
- V_{ILmax} : 确保能被识别为低态的最大输入电压值。
- V_{OHmin} : 输出为高态时的最小输出电压值。
- V_{OLmax} : 输出为低态时的最大输出电压值。



◆ 供电轨道

- V_{CC}/V_{DD}
- GND称为

注意：一个逻辑门A的输出就是另一个逻辑门B的输入（除非A已经是最后一个逻辑门）

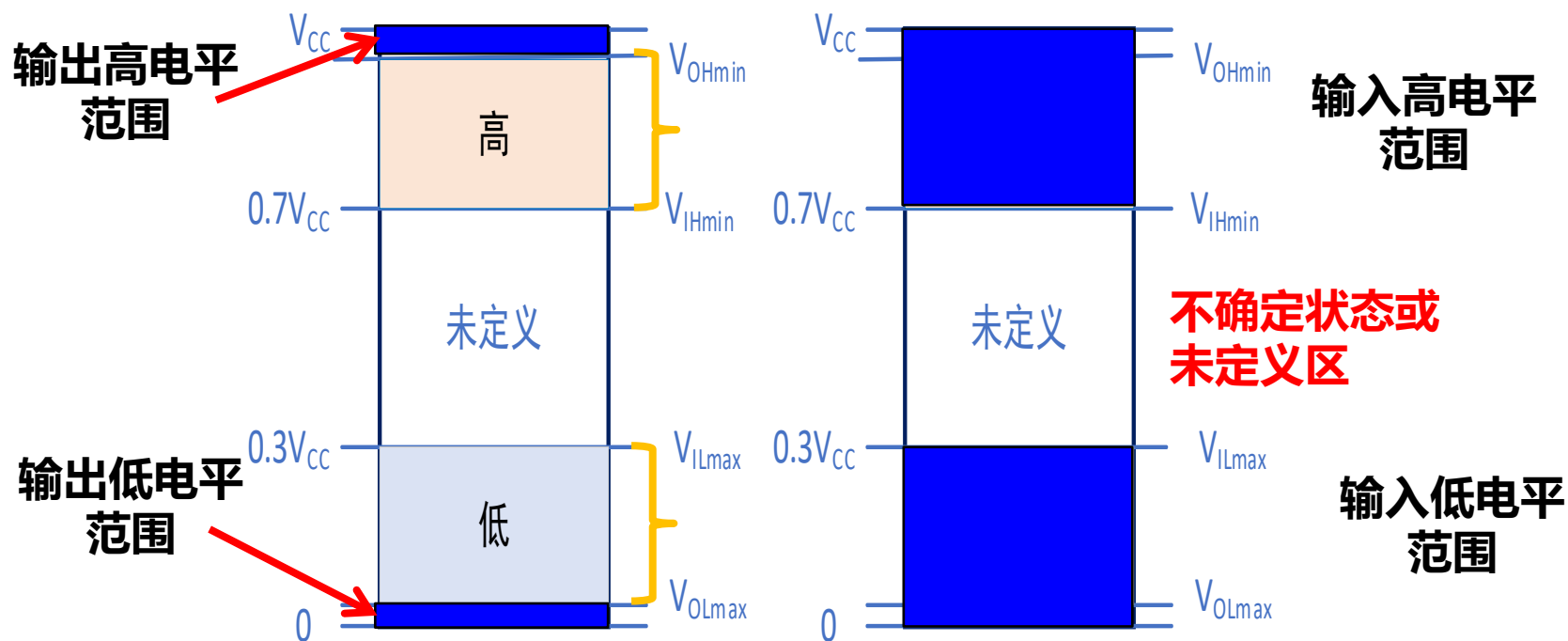
◆ 电平参数的典型数值如下：

- V_{OHmin} : V_{CC} 减0.1V, V_{CC} 最小值是4.5V, 减去0.1V, 得到4.4V。
- V_{OLmax} : 地线GND (0V) +0.1V。
- V_{IHmin} : V_{CC} 的70%, 约为3.15V。
- V_{ILmax} : V_{CC} 的30%, 约为1.35V。

(*) 1.2 数字抽象

◆ **直流噪声容限DC noise margin**是一种对噪声程度的度量，表示多大的噪声会使输出电压被破坏，成为不可被输入端识别的值。

- 高态直流噪声容限 $NM_H = V_{OHmin} - V_{IHmin}$ 约为 $4.4 - 3.15 = 1.25V$
- 低态直流噪声容限 $NM_L = V_{ILmax} - V_{OLmax}$ 约为 $1.35 - 0.1 = 1.25V$

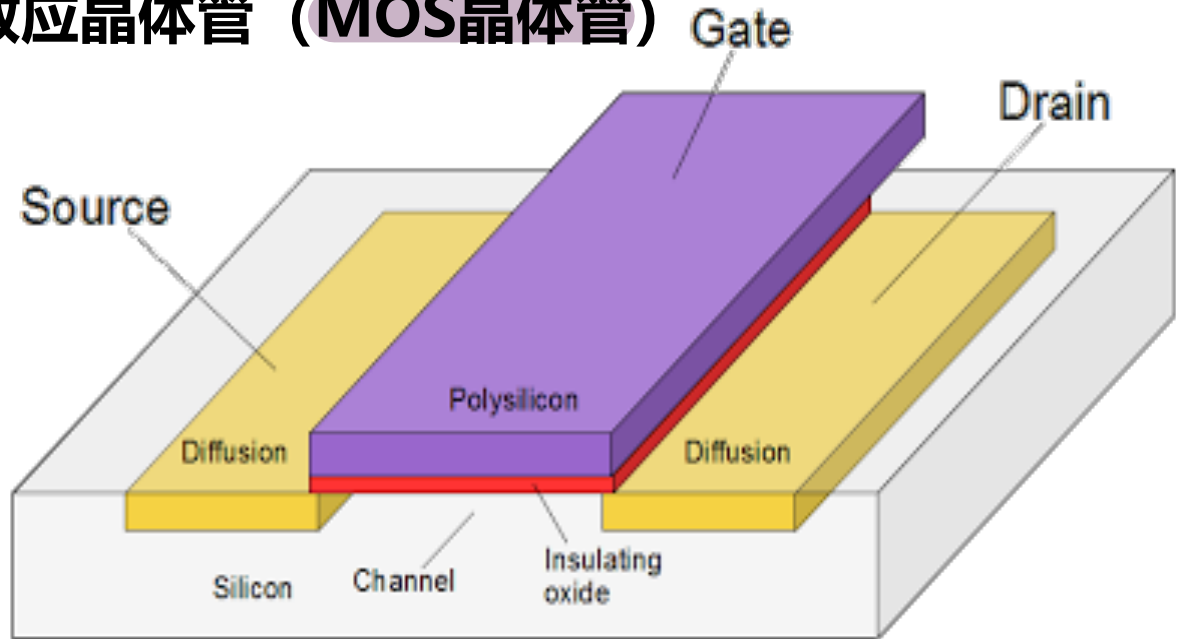


1.3 CMOS晶体管——MOS

◆ 金属氧化物半导体场效应晶体管（MOS晶体管）

◆ 也称三极晶体管

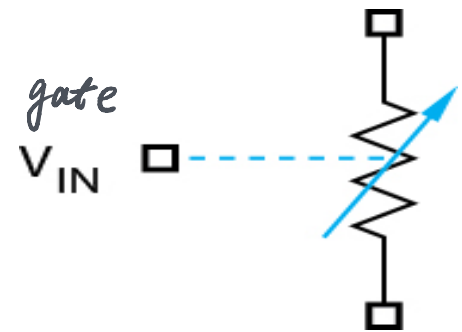
- 栅极gate
- 源极source
- 漏极drain



◆ MOS晶体管分为：

- n沟道型NMOS, (Negative)
- p沟道型PMOS, (Positive)

◆ MOS晶体管可被模型化为一种3端子压控电阻导体，将电压加到一个端子，来控制其他两个端子间的电阻。

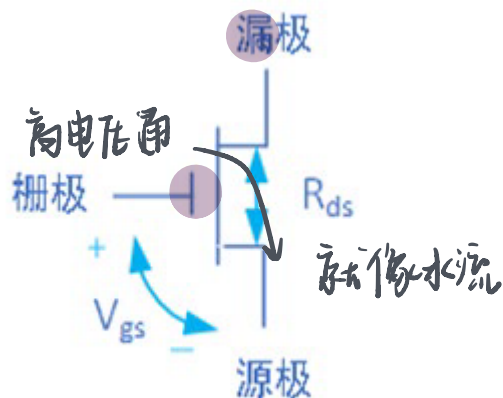


△ 1.3 CMOS晶体管——NMOS和PMOS

◆ 栅极和源极之间电压 V_{gs} 控制源极和漏极间电阻 R_{ds} 的大小

V_{gs} 等于栅、源极电位之差。若栅、源极电位分别为0、+5V，则 $V_{gs} = -5V$

晶体管状态：电阻很小时为导通状态；电阻很大时为截止状态



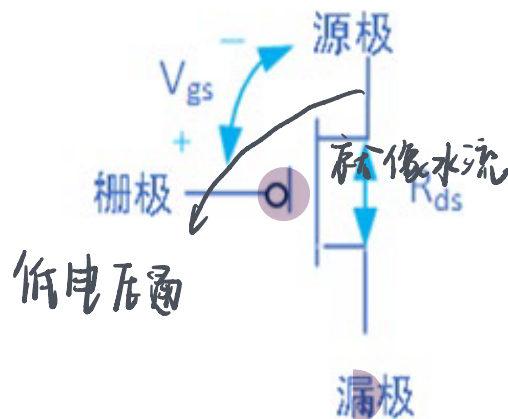
(a) NMOS 晶体管符号

NMOS: 当 $V_{gs} > 0$,

栅极为高电平, 导通

PMOS: 当 $V_{gs} < 0$,

栅极为低电平, 导通



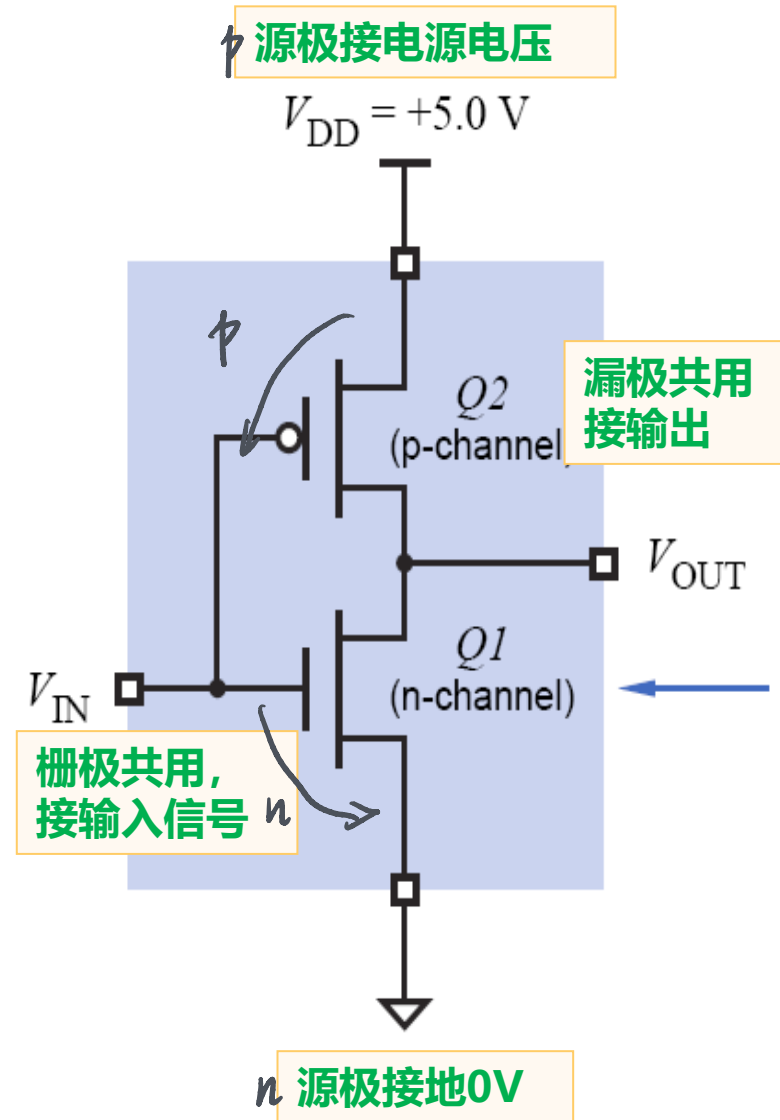
(b) PMOS 晶体管符号

NMOS: 当 $V_{gs} \leq 0$, R_{ds} 很大;
随着 V_{gs} 的增大, R_{ds} 逐步下降
通常 $V_{gs} \geq 0$

PMOS: 当 $V_{gs} \geq 0$, R_{ds} 很大;
随着 V_{gs} 的降低, R_{ds} 逐步下降
通常 $V_{gs} \leq 0$

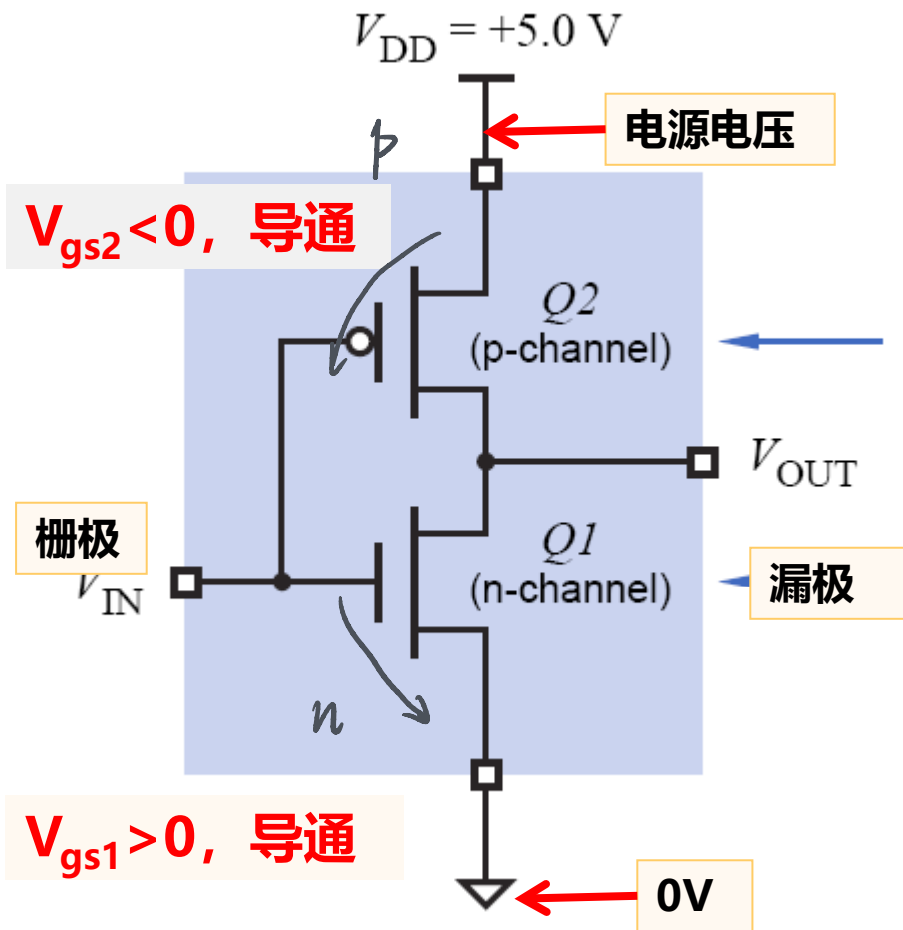
1.3 CMOS晶体管

- ◆ CMOS (Complementary Metal-Oxide Semiconductor) 晶体管以互补的形式共用一对NMOS 和PMOS 晶体管
- ◆ 栅极和漏极共用，分别连接输入和输出
 - NMOS 晶体管的源极连接地线GND
 - PMOS 晶体管的源极连接电源电压 V_{DD}
 - 通过改变栅极的输入电压值 (使得Q1和Q2只会有一个导通)，从而改变漏极的输出电压值
 - 可以看成电压控制开关
 - 输入电压由CMOS开关阈值电压决定，输出电压由晶体管导通时的电阻决定
- ◆ 常用CMOS门电路
 - 反相器/与非门/或非门



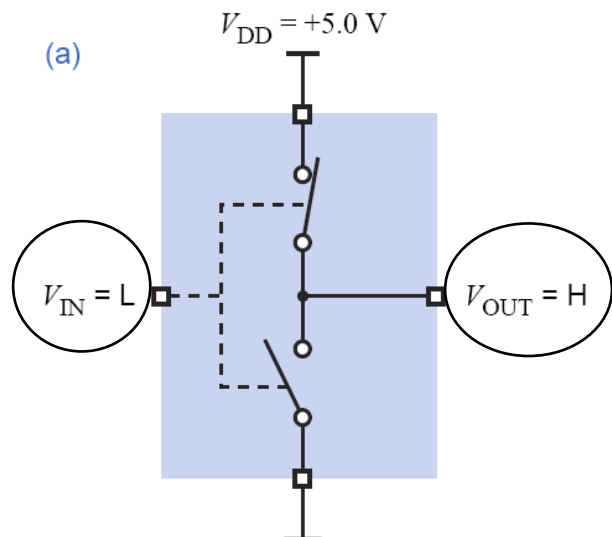
1.3 CMOS晶体管——实现非门

- 非门使用一对CMOS 晶体管实现

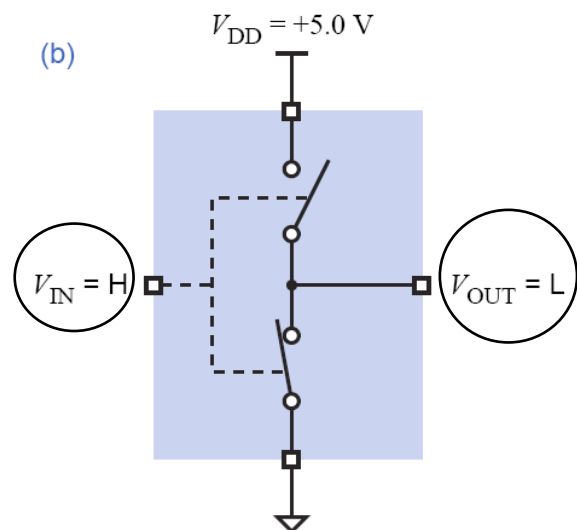


原理图

$$\begin{aligned} V_{gs2} &= -5\text{V} \\ V_{IN} &= L \\ V_{gs1} &= 0 \end{aligned}$$



$$\begin{aligned} V_{gs2} &= 0 \\ V_{IN} &= H \\ V_{gs1} &= 5\text{V} \end{aligned}$$



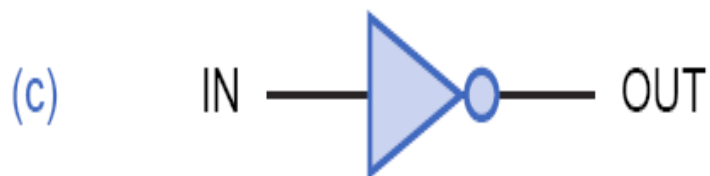
开关模型

1.3 CMOS晶体管——实现非门

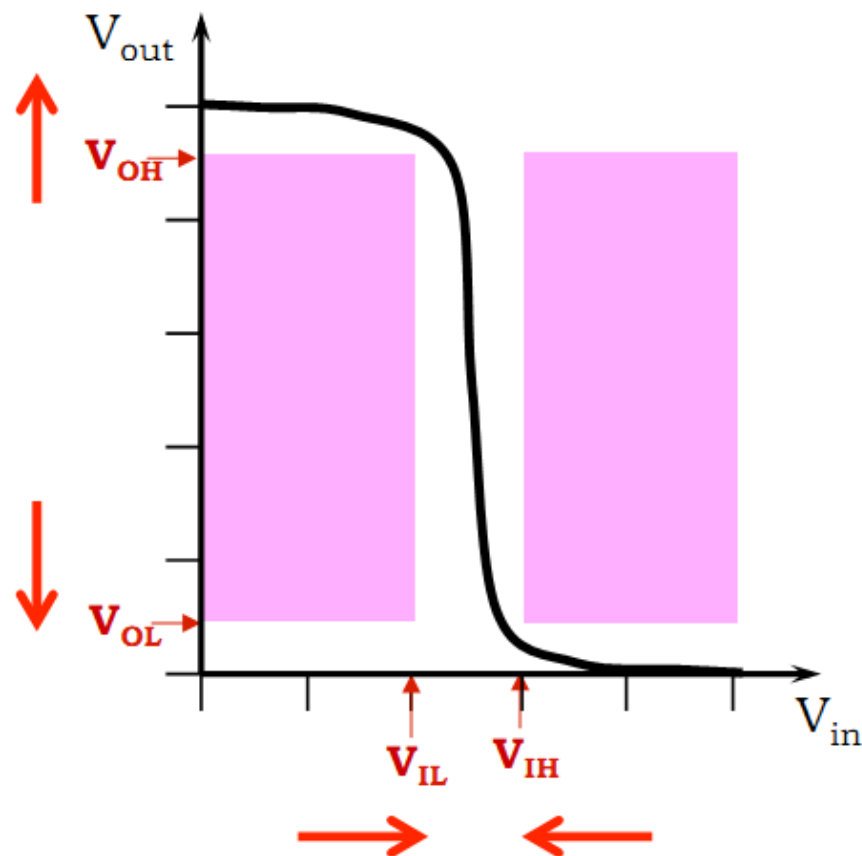
(b)

	N		P	
	V_{IN}	$Q1$	$Q2$	V_{OUT}
	0.0 (L)	off	on	5.0 (H)
	5.0 (H)	on	off	0.0 (L)

功能表



图形符号



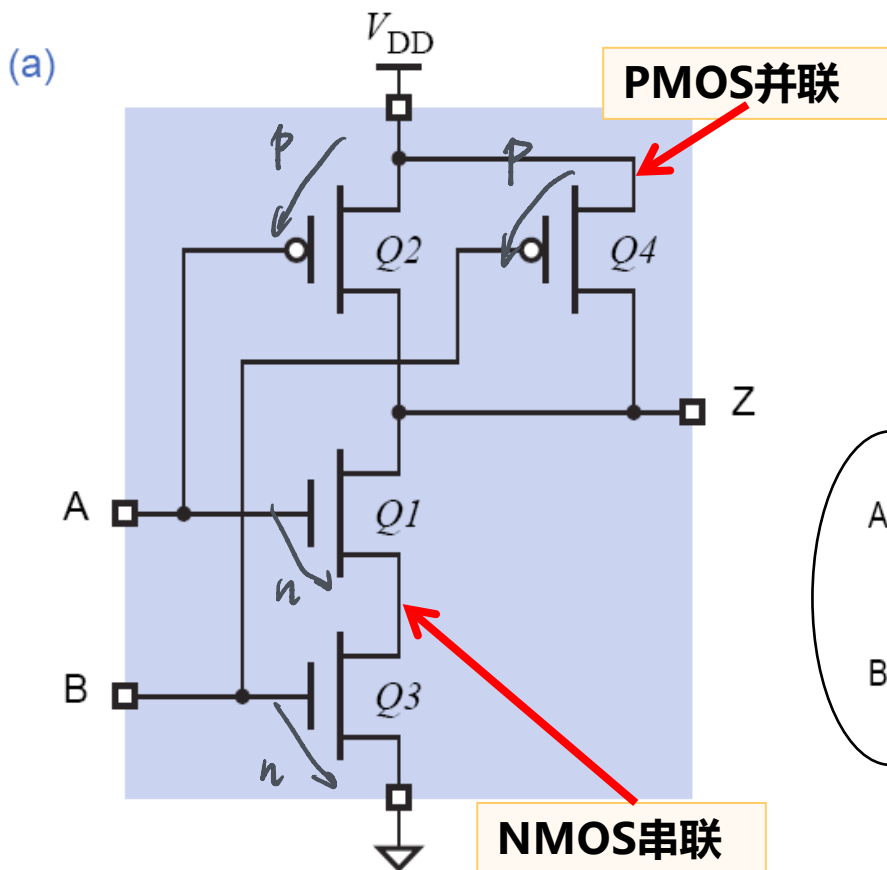
1.3 CMOS晶体管——实现与非门

■ 2输入与非门使用两对CMOS晶体管实现

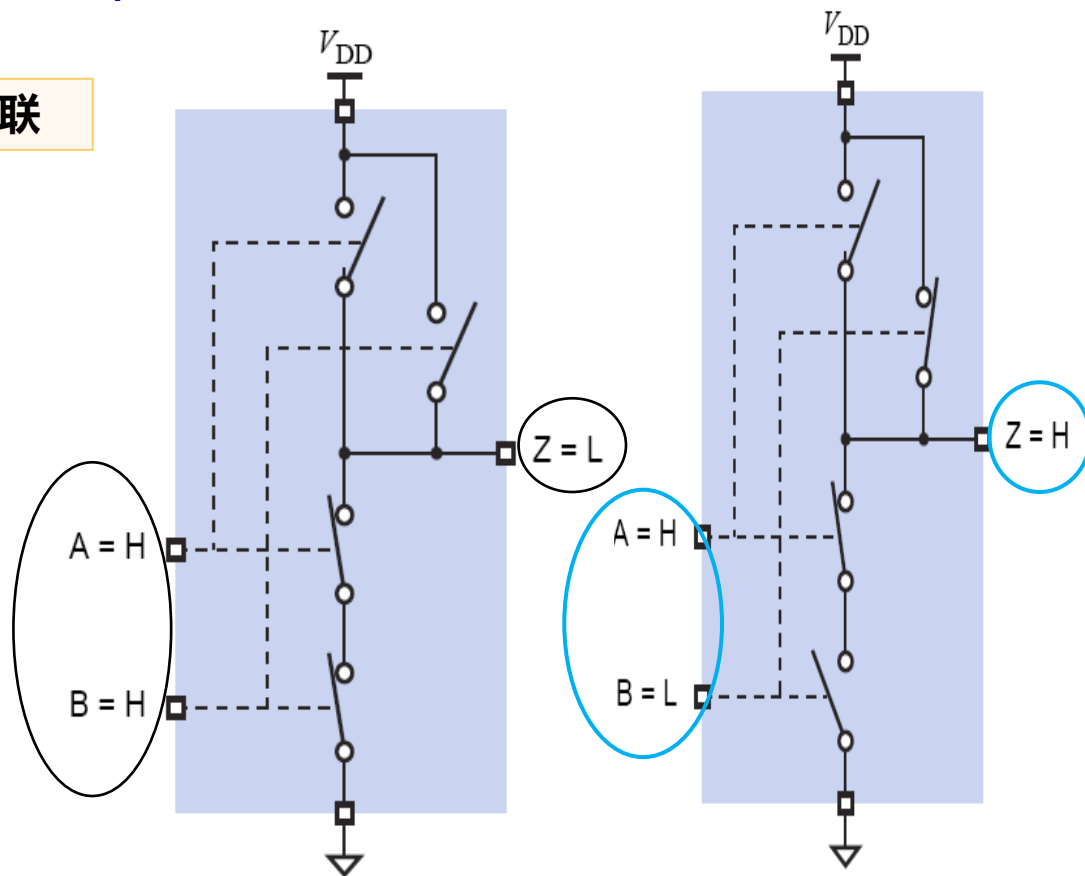
- NMOS管串联 (Q1、Q3)

- PMOS管并联 (Q2、Q4)

A和B中只要有一个是低电平L，输出就一定是高电平H



原理图



开关模型

1.3 CMOS晶体管——实现与非门

A	B	Q1	Q2	Q3	Q4	Z
L	L	off	on	off	on	H
L	H	off	on	on	off	H
H	L	on	off	off	on	H
H	H	on	off	on	off	L

功能表



图形符号

2输入与非门真值表

A	B	Z
0	0	1
0	1	1
1	0	1
1	1	0

1.3 CMOS晶体管——实现或非门

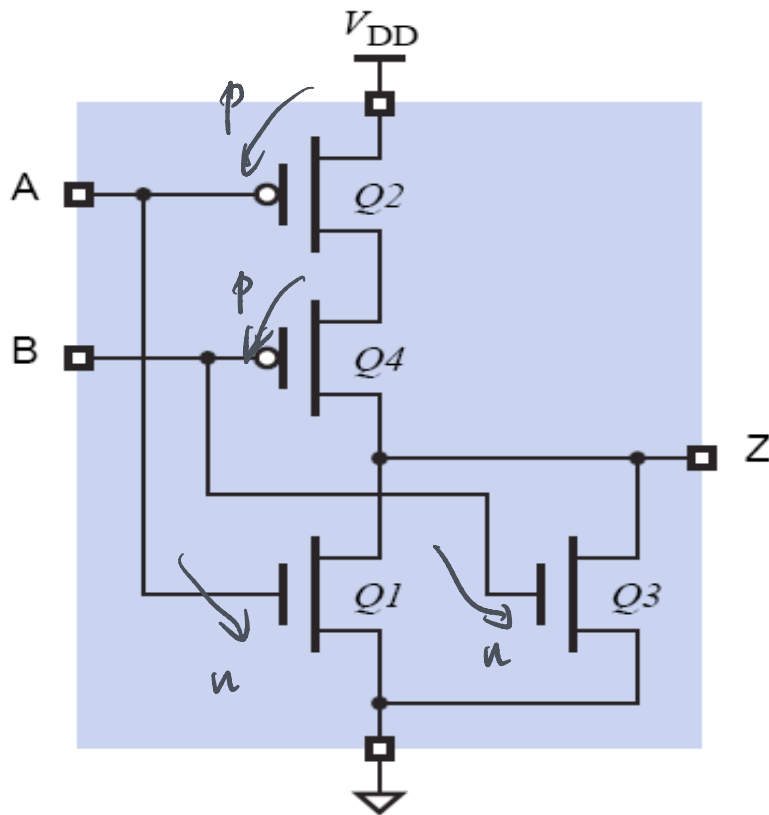
■ 2输入或非门使用两对CMOS晶体管实现

- NMOS管并联 (Q1, Q3)
- PMOS管串联 (Q2, Q4)

A和B中只要有一个是高电平H，输出就一定是低电平L

功能表

A	B	Q1	Q2	Q3	Q4	Z
L	L	off	on	off	on	H
L	H	off	on	on	off	L
H	L	on	off	off	on	L
H	H	on	off	on	off	L



原理图



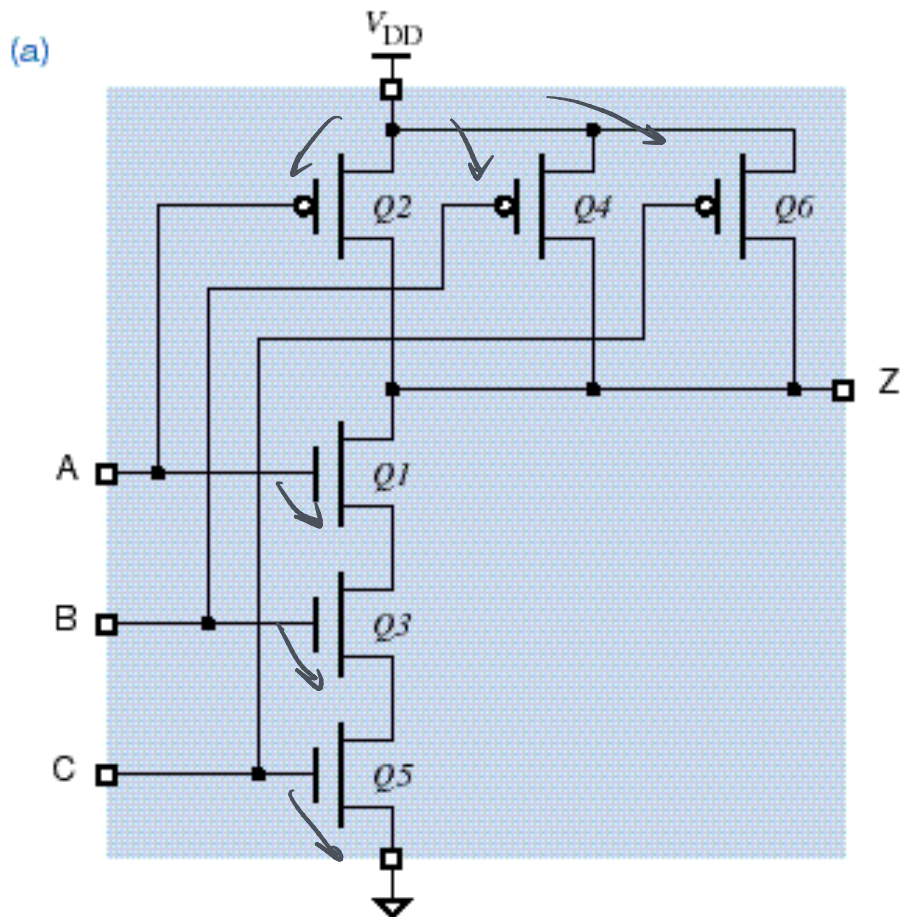
图形符号

CMOS的与门和或门如何得到?

1.3 CMOS晶体管——k输入

- 使用k对NMOS和PMOS晶体管通过串-并联结构构造一个k输入CMOS与非门/或非门
- 3输入与非门包含3对CMOS晶体管

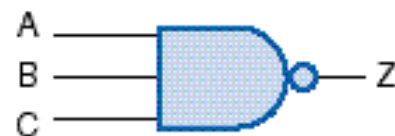
ABC中只要有一个是低电平L，输出就一定是高电平H



(b)

A	B	C	Q1	Q2	Q3	Q4	Q5	Q6	Z
L	L	L	off	on	off	on	off	on	H
L	L	H	off	on	off	on	on	off	H
L	H	L	off	on	on	off	off	on	H
L	H	H	off	on	on	off	on	off	H
H	L	L	on	off	off	on	off	on	H
H	L	H	on	off	off	on	on	off	H
H	H	L	on	off	on	off	off	on	H
H	H	H	on	off	on	off	on	off	L

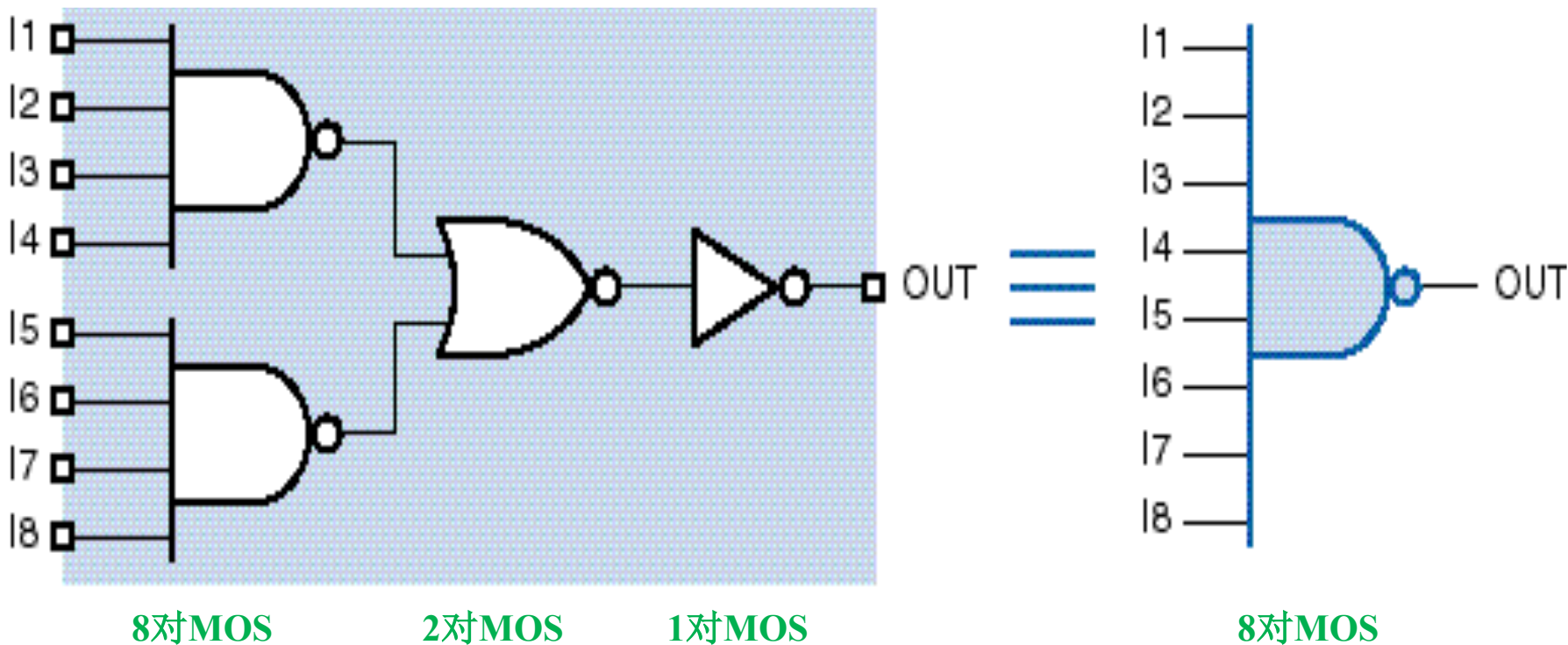
(c)



3 输入与非门

1.3 CMOS晶体管——级联

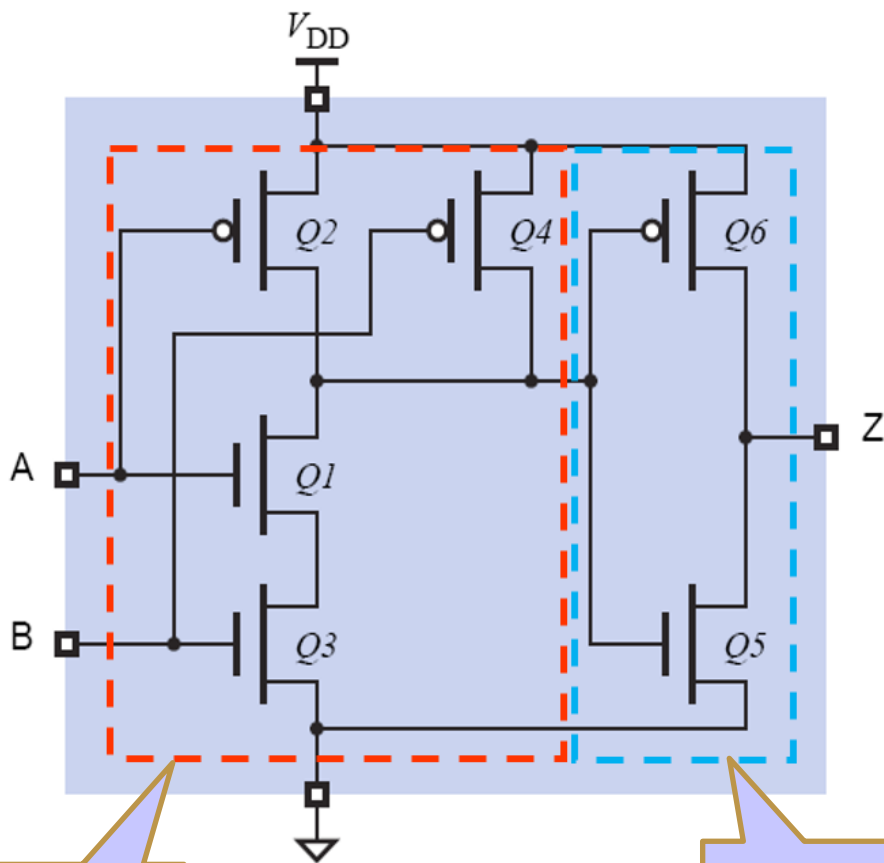
- 受电气特性的限制，输入端不能无限制增加。
- 一般输入端数目小于等于5，不超过8个。
- 输入端较多的门电路可用输入端较少的门电路级联而构成，速度更快、体积更小



8 输入与非门

1.3 CMOS晶体管——级联实现与门

- 通过与非门级联非门实现与门。
- 2输入与门使用了3对CMOS晶体管。



与非门

非门

A	B	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Z
L	L	off	on	off	on	on	off	L
L	H	off	on	on	off	on	off	L
H	L	on	off	off	on	on	off	L
H	H	on	off	on	off	off	on	H

与门比与非门慢



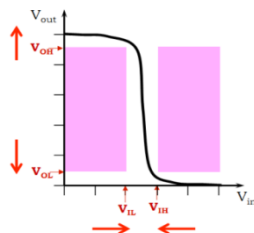
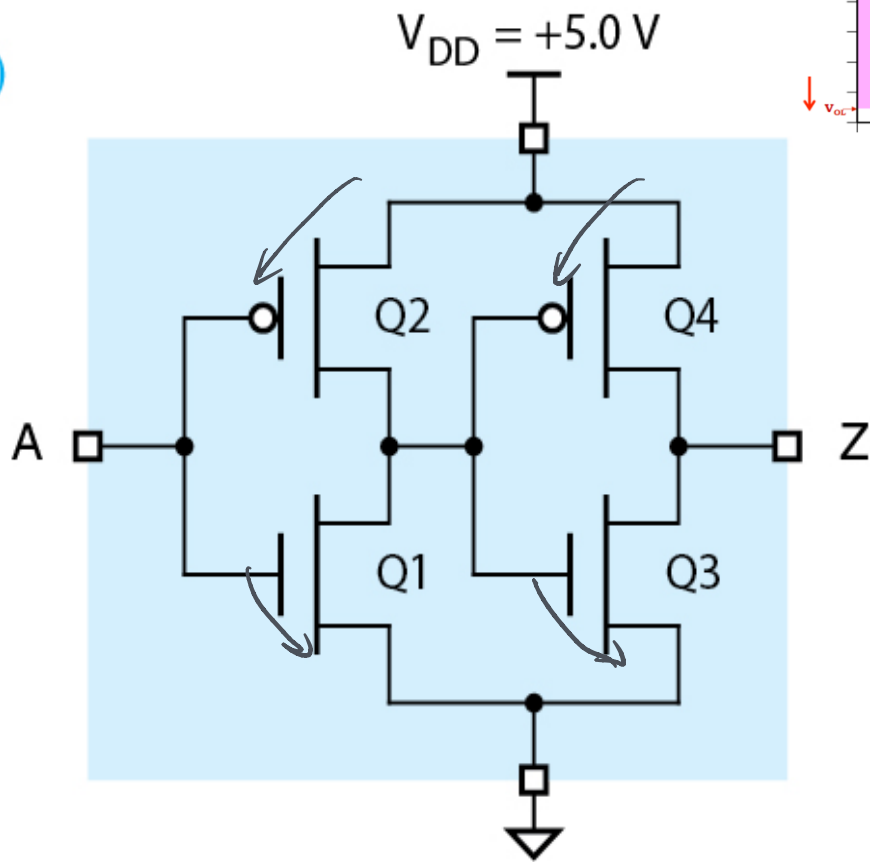
2 输入与门

同理，2 输入或门可以通过一个或非门级联一个非门构成

1.3 CMOS晶体管——级联实现缓冲器

- 两级非门实现缓冲器，将一个“弱”信号转换为具有相同逻辑值的“强”信号

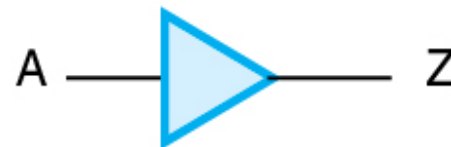
(a)



(b)

A	Q1	Q2	Q3	Q4	Z
L	off	on	on	off	L
H	on	off	off	on	H

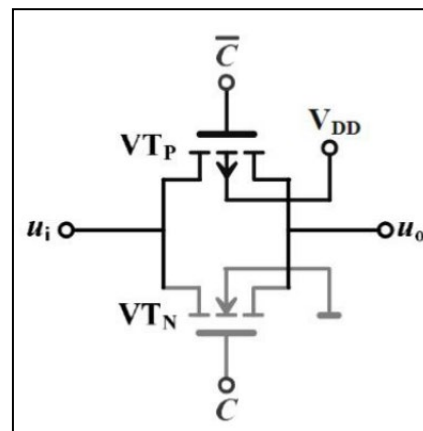
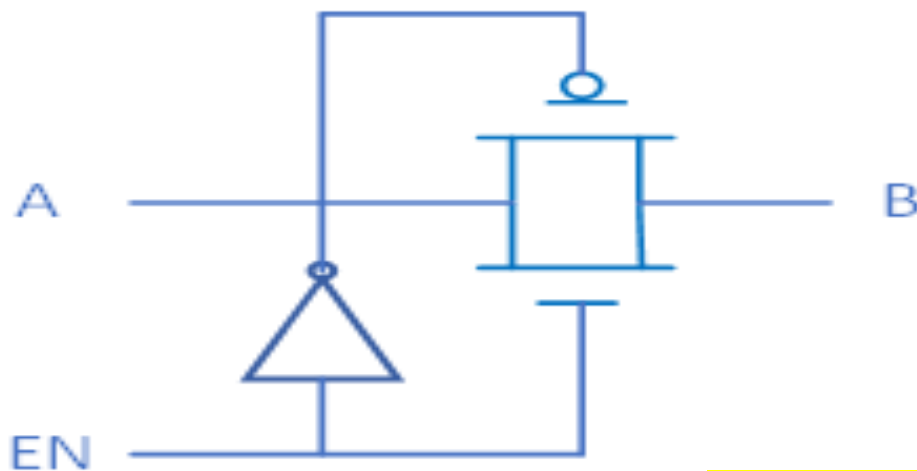
(c)



复杂电路都可以由简单电路级联而成
前一级电路的输出，就是后一级电路的输入

1.3 CMOS晶体管——实现传输门

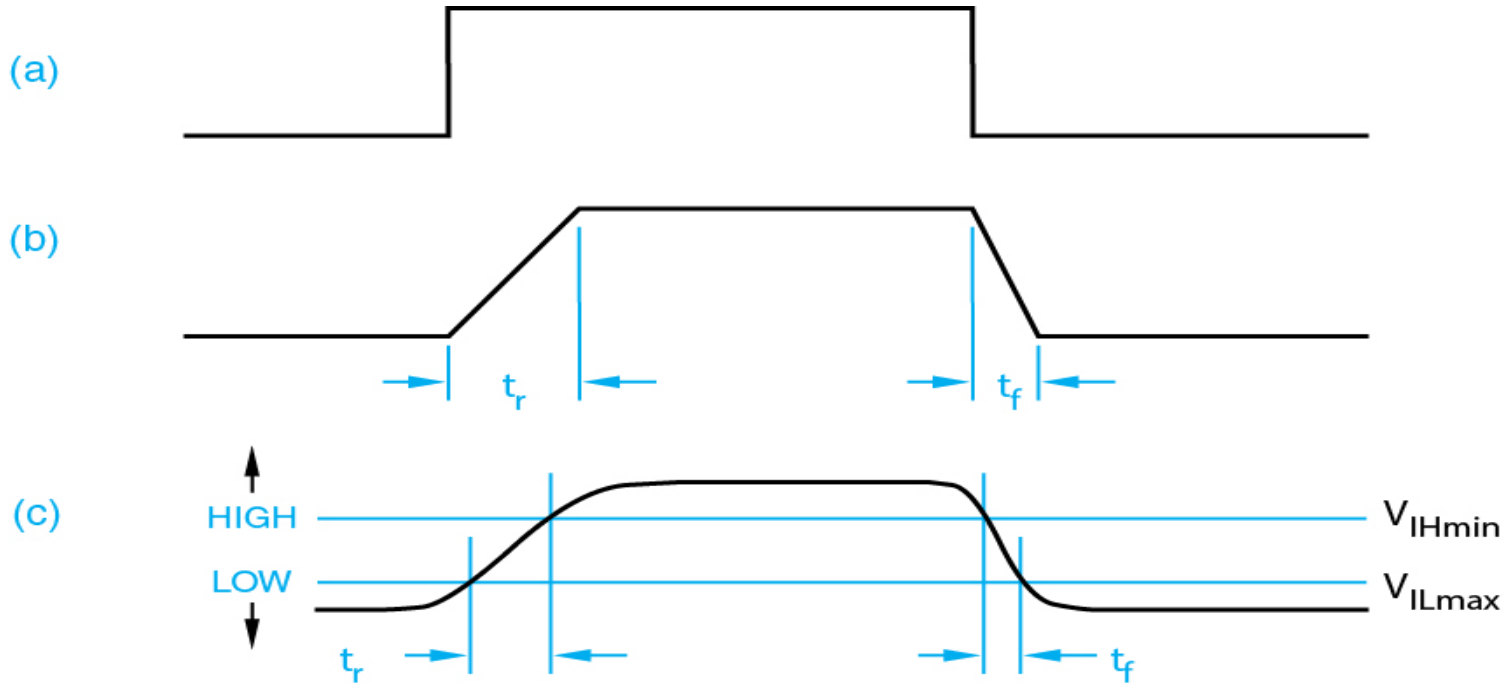
- ◆ **传输门** (transmission gate) 由一对CMOS 晶体管以及控制信号EN构成。
 - 信号EN (使能端) 用于控制晶体管的导通与截止, 其功能相当于一个逻辑控制开关。
 - 当EN为高态时, 若传输的是低态信号, 则NMOS管导通; 否则是PMOS管导通。若EN为低态时, 两个MOS管都截止。
- ◆ 传输门的传播延迟非常短、电路简单, 可双向传输。



用于多路选择器、触发器等

1.4 CMOS电路电气特性

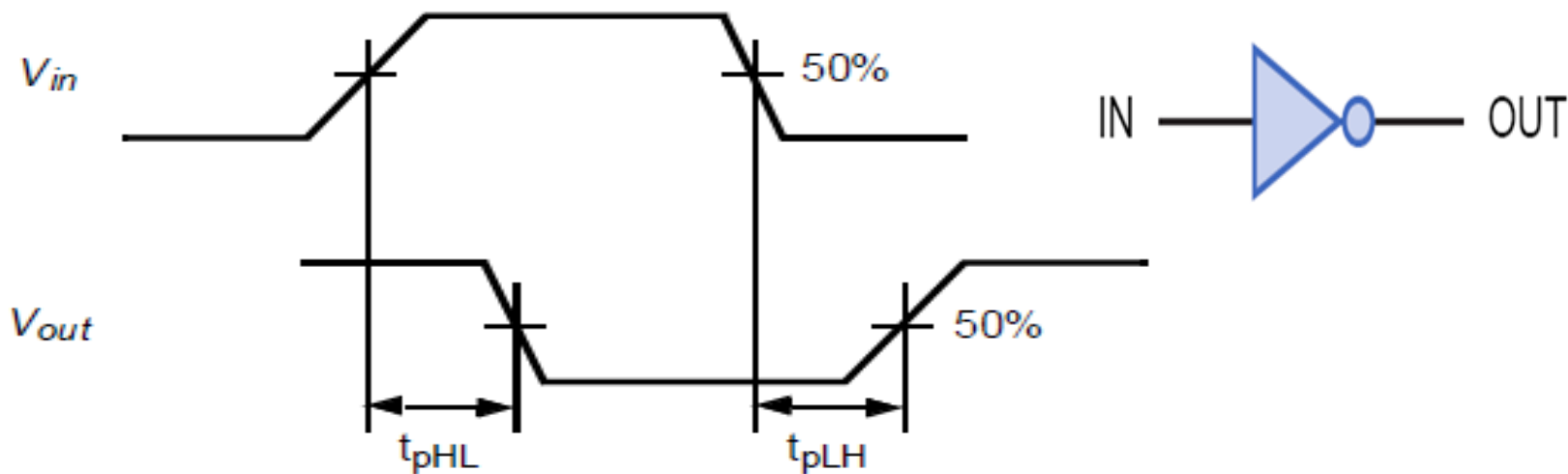
- ◆ **转换时间transition time**：逻辑电路的**输入信号（或输出信号）**从一种状态转换到另一种状态所需的时间
 - 上升时间rise time t_r ：从低态到高态。
 - 下降时间fall time t_f ：从高态到低态。



转换时间 (a)理想状态 (b)近似状态 (c)实际状态

1.4 CMOS电路电气特性

- ◆ **传播延迟**(t_p , propagation delay)指从**输入**信号变化到引起**输出**信号变化所需的时间。
- ◆ **信号通路**signal path: 是指一个特定**输入**信号到逻辑元件的特定**输出**信号所经历的电气通路。
- ◆ t_{pHL} : **输入**变化引起相应**输出**从高到低变化的时间。
- ◆ t_{pLH} : **输入**变化引起相应**输出**低到高变化的时间。



(*) 1.4 CMOS电路电气特性

- ◆ 数字电路在输出信号保持不变时的功率损耗称为**静态功耗**，通常CMOS电路的静态功耗很低，常忽略。
- ◆ 在输出信号**高低状态转换时的功率损耗**称为**动态功耗**。主要来源：
 - 输出端上的电容性负载 C_L
 - CMOS电路内部的功耗电容 C_{pD}
- ◆ 在CMOS电路的应用中， 动态功耗是总功率的主要成分

第一讲小结

◆ 逻辑门

- 门符号、逻辑运算符、真值表、逻辑表达式
- 与、或、非、与非、或非、异或、同或（等价）



◆ 数字抽象

- 物理量的数字化
- 电平参数的典型数值

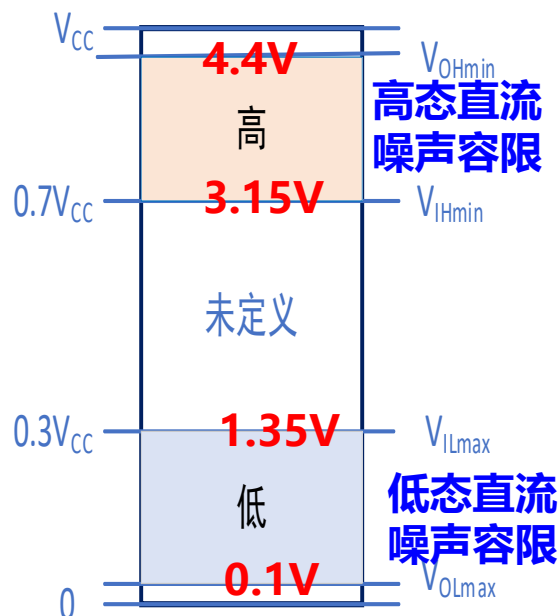
X	Y	X AND Y
0	0	0
0	1	0
1	0	0
1	1	1

◆ CMOS晶体管

- PMOS和NMOS
- 常用CMOS门电路
- 非、与非、或非
- 与（与非-非）、或
- 缓冲器、传输门

◆ CMOS电路电气特性

- 传播延迟
- 动态功耗



第二讲 布尔代数

- ◆公理系统
- ◆定理
- ◆对偶定律
- ◆反演定律

2 布尔代数：基本概念

- ◆ 乔治·布尔George Boole (1815–1864) , 英国数学家, 1854年发明了一种二值代数系统, 称为**开关代数**或**布尔代数**
- ◆ 1938年美国科学家香农提出用布尔代数分析并描述继电器电路的特性, 用**0、1**表示继电器接触状况 (打开/闭合) , 奠定了数字电路的理论基础
- ◆ **逻辑量**: 逻辑**变量**和逻辑**常量**{0,1}
- ◆ **逻辑变量**: 在数字系统中表示某个状态
 - 通常用**字母或字符串**来表示;
- ◆ 只有两种**逻辑值** (逻辑量的取值): “**真**” 或 “**假**”
 - “真” 记作 “**1**” , 数字电路中通常为**高电平**
 - “假” 记作 “**0**” , 数字电路中通常为**低电平**
 - 0和1不表示数值的大小, 只表示完全相反的两种状态

2 布尔代数：基本概念

- ◆ **逻辑运算**：在布尔代数中，有**与**、**或**、**非**三种基本逻辑运算。

与运算：合取、**逻辑乘**，运算符“ \cdot ”、“ \wedge ”，Verilog：“&”

或运算：析取、**逻辑加**，运算符“ $+$ ”、“ \vee ”，Verilog：“|”

非运算：否定、**取反**，运算符“ $-$ ”、“ \neg ”、“ \sim ”，Verilog：“~”

运算优先顺序：

- (1) 圆括号
- (2) **非**运算：一元运算
- (3) **与**运算：二元运算
- (4) **或**运算：二元运算

逻辑乘的符号在单符号变量中可省略“ \cdot ”，不建议省略。

- ◆ **逻辑表达式**：用**逻辑运算符**将**逻辑量**连接起来的代数式。其运算结果是一个逻辑值（不强调输入和输出），例如 $A+B$
- ◆ **逻辑函数**：表明输入和输出变量之间的逻辑关系，例如 $C=A+B$

2.1 公理系统

◆ 用符号X、Y、Z表示逻辑变量的状态。

◆ 公理1:

• (A1)如果 $x \neq 1$, 则 $x=0$; (A1D)如果 $x \neq 0$, 则 $x=1$

◆ 公理2:

• (A2)如果 $x=0$, 则 $\bar{x} = 1$; (A2D)如果 $x=1$, 则 $\bar{x} = 0$

◆ 常量运算公理

• $0 \cdot 0 = 0$ (A3) $1 + 1 = 1$ (A3D)

• $1 \cdot 1 = 1$ (A4) $0 + 0 = 0$ (A4D)

• $0 \cdot 1 = 1 \cdot 0 = 0$ (A5) $1 + 0 = 0 + 1 = 1$ (A5D)

布尔代数的公理和定理基本上成对出现
只要将与、或运算符以及0和1互换即可

2.2 对偶定律

- ◆ 对于任何一个逻辑表达式Y，若将其中的“ \cdot ”与“ $+$ ”互换，“0”和“1”互换，则得到Y的对偶式 Y^D ，称Y与 Y^D 互为对偶式。
- ◆ 对偶定律：若两个逻辑表达式相等，则它们的对偶式也相等。
 - 在保持运算优先次序不变的前提下

$$X + X \cdot Y = X$$

对偶式

$$X \cdot (X + Y) = X$$

2.3 定理

◆ 单变量定理

- 一致性 (T1) $X+0 = X$ (T1D) $X \cdot 1 = X$
- 空元素 (T2) $X+1 = 1$ (T2D) $X \cdot 0 = 0$
- 同一律 (T3) $X+X = X$ (T3D) $X \cdot X = X$
- 还原律 (T4) $\overline{\overline{X}} = X$
- 互补律 (T5) $X+\overline{X} = 1$ (T5D) $X \cdot \overline{X} = 0$

◆ 可用完备归纳法证明

2.3 定理

◆ 二变量和三变量定理

• 交换律 (T6) $X+Y=Y+X$ (T6D) $X\cdot Y=Y\cdot X$

• 结合律 (T7) $(X+Y)+Z=X+(Y+Z)$

(T7D) $(X\cdot Y)\cdot Z=X\cdot(Y\cdot Z)$

• 分配律 (T8) $X\cdot Y+X\cdot Z=X\cdot(Y+Z)$ x 分配给 Y, Z

(T8D) $(X+Y)\cdot(X+Z)=X+Y\cdot Z$

与算术运算
规则不同!

• 吸收律 (T9) $X+X\cdot Y=X$ (T9D) $X\cdot(X+Y)=X$

各种定理都可以用来进行逻辑表达式的化简

2.3 定理

◆证明定理（吸收律）T9（方法有多种）

$$\begin{aligned}X + X \cdot Y &= X \cdot 1 + X \cdot Y \\&= X \cdot (1 + Y) \\&= X \cdot 1 \\&= X\end{aligned}$$

$$= X + X \cdot Y + \bar{X} \cdot Y \quad \text{T9}$$

◆证明T9D

$$\begin{aligned}X \cdot (X + Y) &= X \cdot X + X \cdot Y \\&= X + X \cdot Y \\&= X\end{aligned}$$

$$= X + (X + \bar{X}) \cdot Y \quad \text{T8}$$

$$= X + Y$$

请证明以下公式：

(a) $X + \bar{X} \cdot Y = X + Y$

(b) $X \cdot (\bar{X} + Y) = X \cdot Y$

$$= X \cdot (X + Y) \cdot (\bar{X} + Y) \quad \text{T9D}$$

$$= X \cdot (X \cdot \bar{X} + Y) \quad \text{T8D}$$

$$= X \cdot Y$$

2.3 定理

◆ 二变量和三变量定理

• 交换律 (T6) $X+Y=Y+X$ (T6D) $X \cdot Y=Y \cdot X$

• 结合律 (T7) $(X+Y)+Z=X+(Y+Z)$

(T7D) $(X \cdot Y) \cdot Z=X \cdot (Y \cdot Z)$

• 分配律 (T8) $X \cdot Y+X \cdot Z=X \cdot (Y+Z)$

(T8D) $(X+Y) \cdot (X+Z)=X+Y \cdot Z$

与算术运算
规则不同!

• 吸收律 (T9) $X+X \cdot Y=X$ (T9D) $X \cdot (X+Y)=X$

• 组合律 (T10) $X \cdot Y+X \cdot \bar{Y}=X$ (T10D) $(X+Y) \cdot (X+\bar{Y})=X$

• 一致律 (T11) $X \cdot Y+\bar{X} \cdot Z+Y \cdot Z=X \cdot Y+\bar{X} \cdot Z$

(T11D) $(X+Y) \cdot (\bar{X}+Z) \cdot (Y+Z)=(X+Y) \cdot (\bar{X}+Z)$

$Y \cdot Z$ 称为一致项/冗余项, 若 $Y \cdot Z$ 为1, 则 $X \cdot Y$ 和 $\bar{X} \cdot Z$ 必有一个为1

同理: $Y+Z$ 为冗余项

2.3 定理

◆ n变量定理

• 广义同一律 (T12) $X + X + \dots + X = X$

(T12D) $X \cdot X \cdot \dots \cdot X = X$

• 德·摩根定理 De Morgan's Theorem

(T13) $\overline{X_1 \cdot X_2 \cdot \dots \cdot X_n} = \overline{X_1} + \overline{X_2} + \dots + \overline{X_n}$

(T13D) $\overline{\overline{X_1} + \overline{X_2} + \dots + \overline{X_n}} = X_1 \cdot X_2 \cdot \dots \cdot X_n$

• 广义德·摩根定理 保持与、或运算顺序不变

(T14) $\overline{F(X_1, X_2, \dots, X_n, +, \cdot)} = F(\overline{X_1}, \overline{X_2}, \dots, \overline{X_n}, \cdot, +)$

• 香农定理 用于多变量函数的实现

(T15) $F(X_1, X_2, \dots, X_n) = X_1 \cdot F(1, X_2, \dots, X_n) + \overline{X_1} \cdot F(0, X_2, \dots, X_n)$

(T15D) $F(X_1, X_2, \dots, X_n) = [X_1 + F(0, X_2, \dots, X_n)] \cdot [\overline{X_1} + F(1, X_2, \dots, X_n)]$

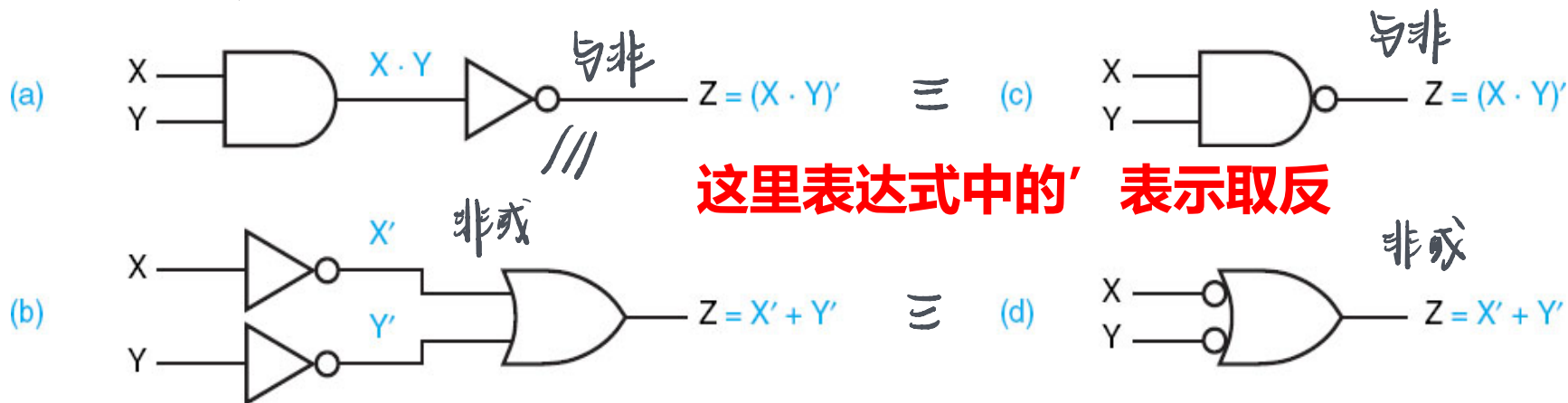
n个变量

n-1个变量

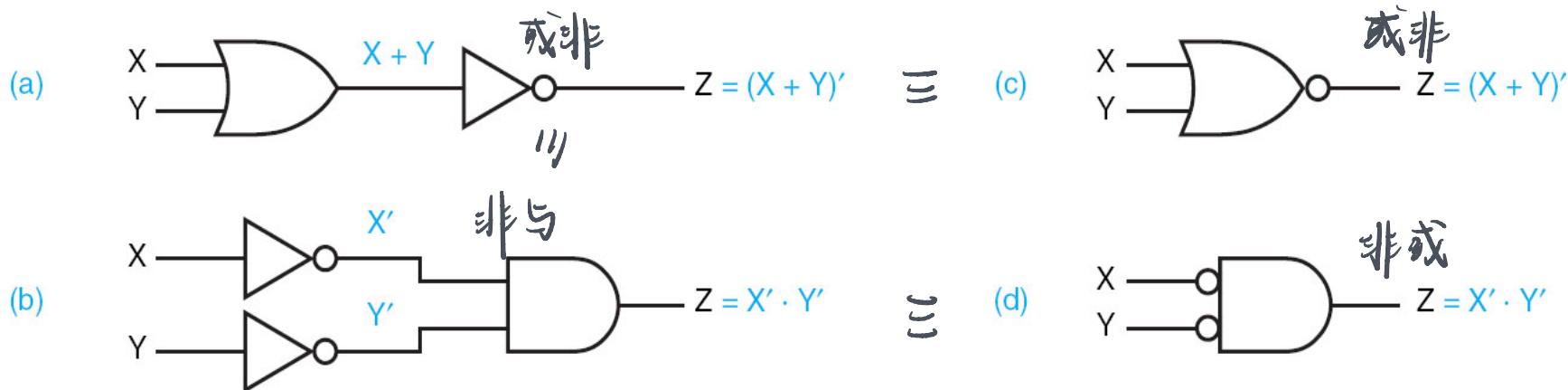
2.3 定理

等效电路具有相同的功能
实现与非逻辑时，可用与门接非门，也可以用非门接或门，也可以直接用与非门实现

◆ 德·摩根定理的应用



根据T13: a) 与-非, b) 非-或, c) 与非门逻辑符号, d) 与非门等效电路



根据T13D: a) 或非-非, b) 非-与, c) 或非门逻辑符号, d) 或非门等效电路

2.3 定理

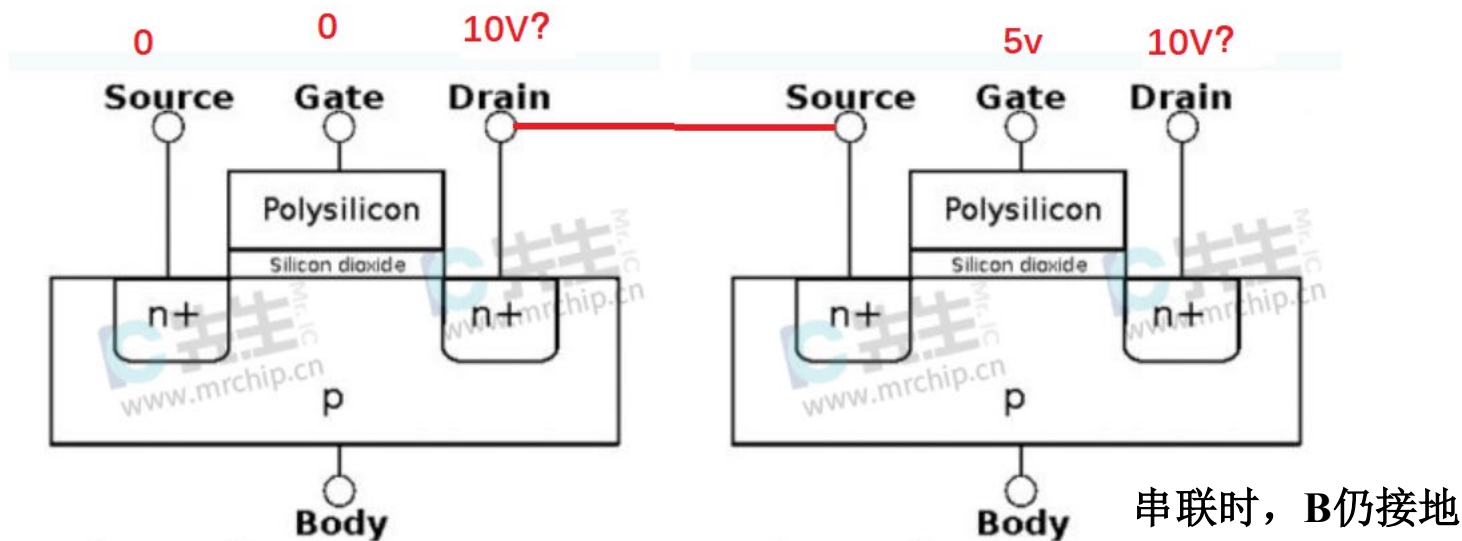
◆ 德·摩根定理的应用例（注：这里省略了很多“与”运算符）

$$\begin{aligned}\overline{a(b+c) + \bar{a}b} &= \overline{a(b+c)} \cdot \overline{\bar{a}b} \\&= (\bar{a} + \overline{(b+c)}) \cdot (\bar{\bar{a}} + \bar{b}) \\&= (\bar{a} + \bar{b} \cdot \bar{c}) \cdot (a + \bar{b}) \\&= (\bar{a} + \bar{b} \cdot \bar{c}) \cdot a + (\bar{a} + \bar{b} \cdot \bar{c}) \cdot \bar{b} \\&= \bar{a}a + \bar{b}\bar{c}a + \bar{a}\bar{b} + \bar{b}\bar{c}\bar{b} \\&= a\bar{b}\bar{c} + \bar{a}\bar{b} + \bar{b}\bar{c} \\&= (a\bar{c} + \bar{a} + \bar{c})\bar{b} \\&= \bar{b}(\bar{a} + \bar{c})\end{aligned}$$

T9的应用

(*) NMOS (增强型) 的一些补充 (无需掌握)

- ◆ 基底 (衬底), SGD、B四个电极,
- ◆ 除了SD之间没有, 其它任意两极之间存在寄生电容
- ◆ S端和B端共同接地, D端给出初始电压
- ◆ G端从0开始不断增压, 随着不同电容的充放电等工作过程, NMOS从截止、到电阻逐渐减小 (两个N+区域之间的电流增大)、到导通
- ◆ 导通时, 就是D端电压降至与S端趋同且稳定的状态



另外, CMOS传输门详解: <https://zhuanlan.zhihu.com/p/565873624>

第三讲 逻辑关系描述

- ◆逻辑函数
- ◆真值表与波形图
- ◆标准范式表示

3.1 逻辑函数

- ◆ **逻辑函数**是反映输入变量和输出变量之间逻辑关系的**表达式**。
 - 将一组取值范围在 $\{0, 1\}$ 之中的输入变量**唯一**映射到在同样取值范围中的输出变量。

设 X_1, X_2, \dots, X_n 是 n 个变量,
每个变量取值0 或1。

$F(X_1, X_2, \dots, X_n)$ 是 $X_1,$
 X_2, \dots, X_n 的一个逻辑函数,
其取值由 X_1, X_2, \dots, X_n 的取
值决定—— **n 个变量取值的所有排列组合, 共 2^n 种情况**

X_1, X_2, \dots, X_n	$F(X_1, X_2, \dots, X_n)$
0 0 \dots 0	0
0 0 \dots 1	0
\dots	\dots
0 1 \dots 0	1
0 1 \dots 1	0
\dots	\dots
1 0 \dots 0	1
\dots	\dots
1 1 \dots 1	0

3.1 逻辑函数

- ◆ 每一个输入组合都有一个确定的输出值
- ◆ 每个逻辑函数都有一组确定的输出（分别对应各种输入组合）
- ◆ 两个输入变量（4种输入组合）的函数 $F(x,y)$ 可能的16种输出值如下：

X	Y	$F_i(X, Y)$															
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

输入: 2^n
输出: 2^{2^n}

三个输入变量有多少种可能的输入组合? $2^3=8$ 种

三个输入变量有多少种可能的逻辑函数 $f(x,y,z)$?

$2^8=256$ 种 2^{2^3}

3.2 (1) 真值表

◆ 真值表 true table:

- 用二维表的形式列出一个逻辑函数所有的输入组合和对应的输出值。
- ◆ 标题栏左侧为输入组合，右侧对应输出。
- ◆ 输入组合通常按照递增顺序排列，输出写在相邻的列中。
- ◆ n 个输入变量逻辑函数的真值表有 2^n 行。
- ◆ 当 n 较大时，真值表将变得十分巨大而失去使用价值。

该真值表对应的逻辑函数是

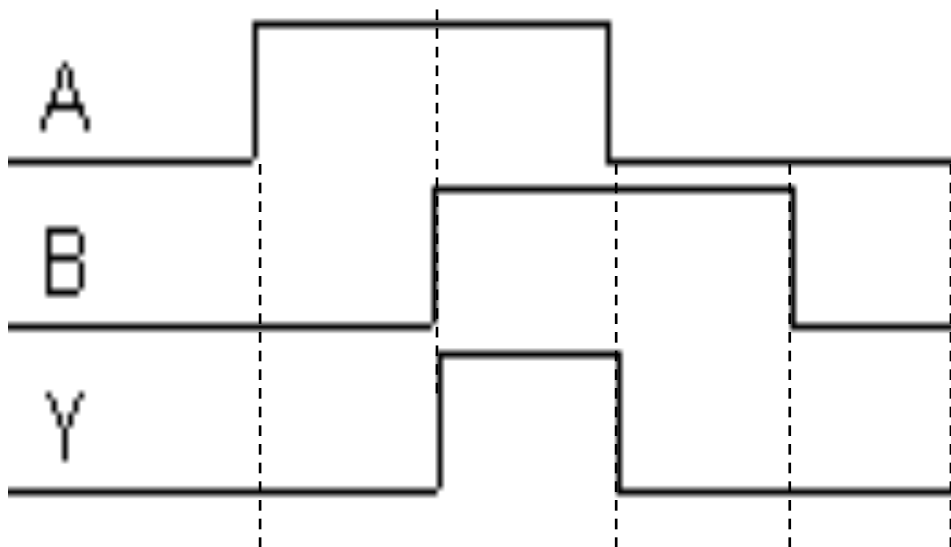
$$Y = A \cdot B + B \cdot C + A \cdot C$$

真值表

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

3.2 (2) 波形图

- ◆ **波形图**描绘了逻辑函数输出变量对于输入变量的变化所产生的响应。
在理想状态下，忽略时间延迟。
 - 横轴表示时间
 - 纵向用**横线的高低**来表示逻辑值大小
- ◆ 完整的波形图至少需要列出所有的输入组合和所对应的输出值



$$Y = A \cdot B$$

3.3 逻辑函数的标准表示

- ◆ **乘积项**：包含1个或1个以上逻辑变量的与项。例如， X 、 $X \cdot Y$ 和 $\bar{X} \cdot \bar{Y} \cdot Z$ 都是乘积项。
- ◆ **求和项**：包含1个或1个以上逻辑变量的或项。例如， X 、 $X+Y$ 和 $\bar{X} + \bar{Y} + Z$ 都是求和项。
- ◆ **“与-或”表达式或积之和表达式** (Sum of Product, SOP)：多个乘积项的或运算。例如： $X \cdot Y + \bar{X} \cdot \bar{Y} \cdot Z$ 。
先与后或
- ◆ **“或-与”表达式或和之积表达式** (Product of Sum, POS)：多个求和项的与运算。例如： $(X+Y) \cdot (\bar{X} + \bar{Y} + Z)$ 。
先或后与

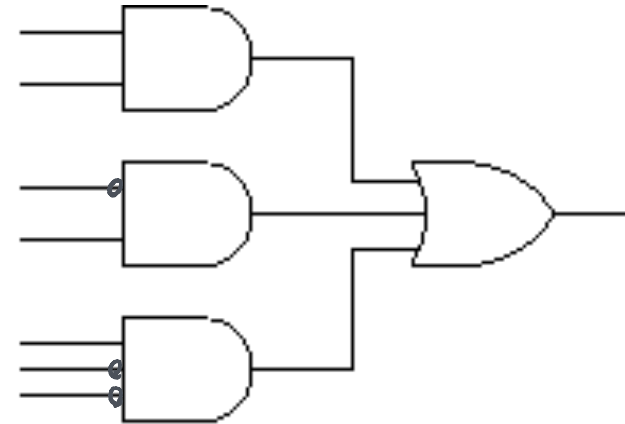
3.3 逻辑函数的标准表示

◆ 与-或表达式：积之和表达式SOP

$$f(A, B, C) = A \cdot B + \bar{A} \cdot C + A \cdot \bar{B} \cdot \bar{C}$$

与项

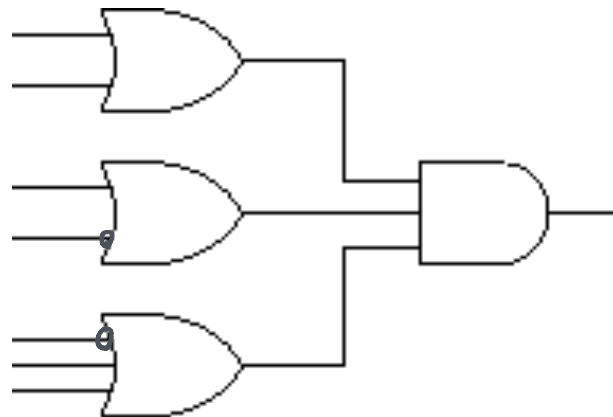
乘积项



◆ 或-与表达式：和之积表达式POS

$$f(A, B, C) = (A + B) \cdot (A + \bar{C}) \cdot (\bar{A} + B + C)$$

求和项
或项



3.3 逻辑函数的标准表示

- ◆ **标准乘积项（标准求和项）**：每个逻辑变量出现且仅出现一次的乘积项（求和项）。n 个变量的标准项共有 2^n 个。

$$A \cdot B \quad 11$$

$$A \cdot \overline{B} \quad 10$$

$$\overline{A} \cdot B \quad 01$$

$$\overline{A} \cdot \overline{B} \quad 00$$

当每个逻辑变量取值定下来之后，只有一个乘积项的输出值为1

$$A + B \quad 00$$

$$A + \overline{B} \quad 01$$

$$\overline{A} + B \quad 10$$

$$\overline{A} + \overline{B} \quad 11$$

当每个逻辑变量取值定下来之后，只有一个求和项的输出值为0

$$A \cdot B \cdot C$$

$$A \cdot B \cdot \overline{C}$$

$$A \cdot \overline{B} \cdot C$$

$$A \cdot \overline{B} \cdot \overline{C}$$

$$\overline{A} \cdot B \cdot C$$

$$\overline{A} \cdot B \cdot \overline{C}$$

$$\overline{A} \cdot \overline{B} \cdot C$$

$$\overline{A} \cdot \overline{B} \cdot \overline{C}$$

3.3 逻辑函数的标准表示

- ◆ **标准乘积项**（**标准求和项**）：每个逻辑变量出现且仅出现一次的乘积项（**求和项**）。 n 个变量的标准项共有 2^n 个。
- ◆ **标准乘积项**也称为**最小项**，每个最小项对应真值表中一个输入组合，赋值该输入组合后，最小项的运算结果为**1**。
- ◆ 若某输入组合对应的二进制数值为 i ，则用 m_i 表示该最小项， i 称为该最小项的编号。
 - 如： $\bar{A} \cdot \bar{B} \cdot C$ 只有当输入**001**时，结果为**1**，最小项编号为 m_1
- ◆ **标准求和项**也称为**最大项**，每个最大项对应真值表中一个输入组合，赋值该输入组合后，最大项的运算结果为**0**。
- ◆ 若某输入组合对应的二进制数值为 i ，则用 M_i 表示该最大项， i 称为该最大项的编号。
 - 如： $\bar{A} + \bar{B} + C$ 只有输入**110**时，结果为**0**，最大项编号为 M_6

3.3 逻辑函数的标准表示

真值表

序号	A B C	Y	最小项	最大项
0	000	1	$\bar{A} \cdot \bar{B} \cdot \bar{C}$	$A + B + C$
1	001	0	$\bar{A} \cdot \bar{B} \cdot C$	$A + B + \bar{C}$
2	010	0	$\bar{A} \cdot B \cdot \bar{C}$	$A + \bar{B} + C$
3	011	0	$\bar{A} \cdot B \cdot C$	$A + \bar{B} + \bar{C}$
4	100	0	$A \cdot \bar{B} \cdot \bar{C}$	$\bar{A} + B + C$
5	101	0	$A \cdot \bar{B} \cdot C$	$\bar{A} + B + \bar{C}$
6	110	1	$A \cdot B \cdot \bar{C}$	$\bar{A} + \bar{B} + C$
7	111	1	$A \cdot B \cdot C$	$\bar{A} + \bar{B} + \bar{C}$

标准和/最小项列表：使得函数输出为1的最小项之和。

$$Y = \bar{A} \cdot \bar{B} \cdot \bar{C} + A \cdot B \cdot \bar{C} + A \cdot B \cdot C$$

$$= \sum m(0, 6, 7)$$

标准积/最大项列表：使得函数输出为0的最大项之积。

$$Y = (A + B + \bar{C}) \cdot (A + \bar{B} + C) \cdot (A + \bar{B} + \bar{C}) \cdot (\bar{A} + B + C) \cdot (\bar{A} + B + \bar{C})$$

$$= \prod M(1, 2, 3, 4, 5)$$

真值表的表示是唯一的，因此标准项列表的表示也是唯一的。

3.3 逻辑函数的标准表示

- ◆ 函数的最小项列表和最大项列表等价且可相互转换。
- ◆ n 个变量的逻辑函数的最小项列表编号集合与最大项列表编号集合之并集为 n 位编号全集 $\{0, 1, \dots, 2^n - 1\}$, 且这两个集合之交集为 0, 它们为互补关系。
- ◆ 两个列表之间可方便转换, 只需对相应的编号集合求补即可。
 - $f(A,B,C) = \sum m(0,1,2,3) = \prod M(4,5,6,7)$
 - $f(X,Y) = \sum m(1) = \prod M(0,2,3)$
 - $f(A,B,C) = \sum m(0,6,7) = \prod M(1,2,3,4,5)$

标准表示并非最简表示

第四讲 逻辑函数的化简与变换

- ◆代数法化简
- ◆卡诺图化简
- ◆等效逻辑符号表示
- ◆逻辑函数变换

4 逻辑函数的化简与变换

化简的目的

◆ 减少输入变量的数目

- 最小项和最大项的数目随变量数呈指数级增长

◆ 减少门的数目

- 可以使用更小的集成电路器件

◆ 减少电路的规模

- 1. 通常不考虑输入反相器的成本。
2. 通常从真值表或标准范式开始化简。

4.1 代数法化简

- ◆ 利用布尔代数的公理、定理、定律等，消去逻辑表达式中**多余的乘积项或多余的因子**，进行化简。
 - 利用互补律T5: $\bar{X} + X = 1$ ，可消去一个变量。
 - 利用吸收律T9: $X + X \cdot Y = X$ 和 $X + \bar{X} \cdot Y = X + Y$ ，可消去乘积项中一个因子。
 - 利用组合律T10: $X \cdot Y + X \cdot \bar{Y} = X$ 和 $(X + Y) \cdot (X + \bar{Y}) = X$ ，可消去一个变量。
 - 利用一致律T11，可消去冗余的乘积项。
- ◆ 如果表达式的层级超过了两级，则先转换为两级。
- ◆ 如有整体取反运算，则先转换为单变量取反运算。


$$((A+B) \cdot C) + D = AC + BC + D$$

与或，或与

4.1 代数法化简

◆化简: $F(w,x,y,z) = w \cdot x + w \cdot x \cdot y + \bar{w} \cdot y \cdot z + \bar{w} \cdot \bar{y} \cdot z + \bar{w} \cdot x \cdot y \cdot \bar{z}$ 。

$$\begin{aligned} &= w \cdot x + w(y \cdot z + \bar{y} \cdot z + x \cdot \bar{y} \cdot z) && \text{—(根据T8和T9)} \\ &= w \cdot x + w(z + x \cdot y \cdot \bar{z}) && \text{(根据T10)} \\ &= w \cdot x + w(z + x \cdot y) && \text{(根据T9)} \\ &= w \cdot x + w \cdot z + w \cdot x \cdot y && \text{(根据T8)} \\ &= (w + w \cdot y) \cdot x + w \cdot z && \text{(根据T8)} \\ &= (w + y) \cdot x + w \cdot z && \text{(根据T9)} \\ &= w \cdot x + x \cdot y + w \cdot z && \text{(根据T8)} \end{aligned}$$

◆和原表达式相比，化简后减少了2个与门、11个输入端。

原来第一级有5个与门，15个输入端；第二级是5个与门的输出作为输入端，接一个或门
化简后第一级有3个与门，6个输入端；第二级是3个与门的输出作为输入端，接一个或门

4.1 代数法化简

◆化简 $f(A,B,C,D) = \sum m(1,3,5,7)$ 小项之和 $f(A,B,C,D) = \sum m(1,3,5,7)$

$$\begin{aligned} &= \bar{A} \cdot \bar{B} \cdot \bar{C} \cdot D + \bar{A} \cdot \bar{B} \cdot C \cdot D + \bar{A} \cdot B \cdot \bar{C} \cdot D + \bar{A} \cdot B \cdot C \cdot D \\ &= \bar{A} \cdot \bar{B} \cdot D + \bar{A} \cdot C \cdot D + \bar{A} \cdot B \cdot D \\ &= \bar{A} \cdot D + \bar{A} \cdot C \cdot D = \bar{A} \cdot D \end{aligned}$$

◆优点

- 不受变量数目的限制
- 化简比较直观

◆缺点

- 没有一定的规律和步骤，技巧性很强
- 难以判断化简结果是否最简

◆判别逻辑表达式是否为最简

- 乘积项（与项）最少
- 每个乘积项中因子（逻辑变量）最少

△ 4.2 卡诺图化简

◆ **卡诺图** (Karnaugh map) : 真值表的图形化表示, 把能化简的最小项通过**相邻项合并**的可视化方式标识出来。

- n 个变量的卡诺图是一个含有 2^n 个单元的矩阵图
- **每一行和每一列的编号**对应逻辑变量的输入组合, 0 表示反变量, 1 表示原变量
- 编号按照**格雷码**的顺序排列, 即相邻编号只有**1**位不同
- 根据格雷码的规则, 空间位置上 (**上下、左右或首尾**) **相邻**的小方格具有**逻辑相邻性**

		B	
		0	1
A	0	$\overline{A}\overline{B}$	$\overline{A}B$
	1	$A\overline{B}$	AB

		B	
		0	1
A	0	m_0	m_1
	1	m_2	m_3

每个单元对应一个最小项

		CD			
		00	01	11	10
AB	00	0	1	3	2
	01	4	5	7	6
	11	12	13	15	14
	10	8	9	11	10

4.2 卡诺图化简

◆ 行、列可互换

		Y	
		0	1
X	0	m_0	m_1
	1	m_2	m_3

		X	
		0	1
Y	0	m_0	m_2
	1	m_1	m_3

(a) 二变量

		Z	
		0	1
XY	00	0	1
	01	2	3
	11	6	7
	10	4	5

		X			
		00	01	11	10
Z	0	m_0	m_2	m_6	m_4
	1	m_1	m_3	m_7	m_5

(b) 三变量

		YZ			
		00	01	11	10
WX	00	0	1	3	2
	01	4	5	7	6
	11	12	13	15	14
	10	8	9	11	10

		W			
		00	01	11	10
YZ	00	m_0	m_4	m_{12}	m_8
	01	m_1	m_5	m_{13}	m_9
	11	m_3	m_7	m_{15}	m_{11}
	10	m_2	m_6	m_{14}	m_{10}

(c) 四变量

4.2 卡诺图化简

- ◆ 每个单元标注对应最小项在真值表中的输出值，若为1，称之“1单元”
- ◆ 若两个“1单元”相邻，则表示两个最小项仅1个变量相反。根据T10，这两个最小项可合并为一个乘积项，并消去那个具有相反取值的变量
(T10) $X \cdot Y + X \cdot \bar{Y} = X$
- ◆ 相邻单元数越多可消去的变量数越多
- ◆ 相邻 2^i 个“1单元”可以合并成一个乘积项，并消去 i 个不同的变量
- ◆ 使用一个方框来标注可以合并的“1单元”，这个方框称为卡诺圈

		BC			
		00	01	11	10
A	0		1		
	1		1		

$\bar{B} \cdot C$

A取值相反

		CD			
		00	01	11	10
AB	00	1			1
	01				
	11				
	10	1			1

$\bar{B} \cdot \bar{D}$

A和C取值相反

		CDE							
		000	001	011	010	110	111	101	100
AB	00			1			1		
	01			1			1		
	11			1			1		
	10			1			1		

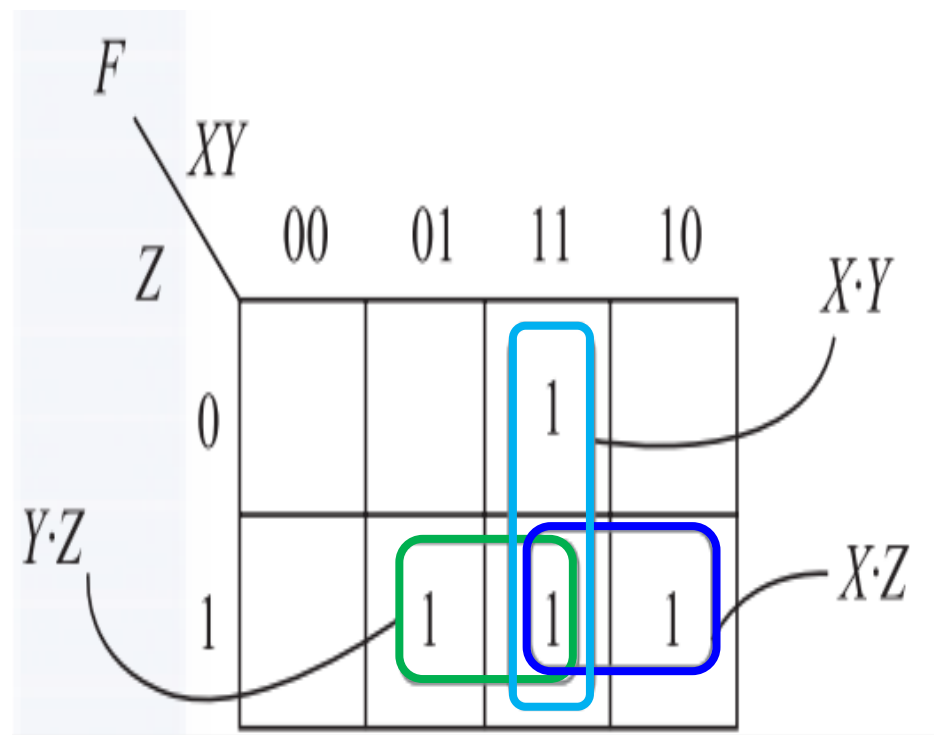
A、B、C取值相反

$D \cdot E$

相邻

4.2 卡诺图化简

序号	X	Y	Z	$F(X, Y, Z)$
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1



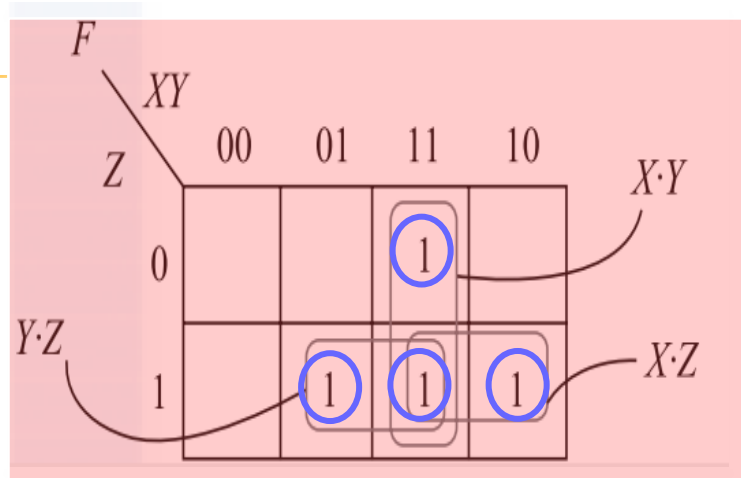
$$F(X, Y, Z) = Y \cdot Z + X \cdot Z + X \cdot Y$$

简化后的表示也并非一定是：最简表示

4.2 卡诺图化简

- ◆ **蕴涵项**是一个乘积项，**覆盖**了逻辑函数的1个或多个最小项。卡诺圈：化简前+化简后

如， $F(X, Y, Z) = \sum m(3, 5, 6, 7)$ 的蕴涵项有： $\bar{X} \cdot Y \cdot Z$ 、 $X \cdot Y \cdot Z$ 和 $Y \cdot Z$ 等， $Y \cdot Z$ 等价于 $\bar{X} \cdot Y \cdot Z + X \cdot Y \cdot Z$ ，因而 $Y \cdot Z$ **覆盖**了最小项 $\bar{X} \cdot Y \cdot Z$ 和 $X \cdot Y \cdot Z$ 。因此， $Y \cdot Z$ 是蕴涵项。



卡诺图化简后

- ◆ **质蕴涵项**(prime implicant)：不能被逻辑函数的其它蕴涵项所**覆盖**的蕴涵项。如， $F(X, Y, Z) = \sum m(3, 5, 6, 7)$ 的质蕴涵项有： $Y \cdot Z$ 、 $X \cdot Z$ 、 $X \cdot Y$ ，而其他蕴含项都可以被另外的蕴含项所覆盖，因此，不是质蕴含项
- ◆ **实质蕴涵项**(essential prime implicant)：覆盖的最小项中**至少有一个最小项**是没有被其他质蕴涵项所覆盖的质蕴涵项。如， $F(X, Y, Z) = \sum m(3, 5, 6, 7)$ 的实质蕴涵项有： $Y \cdot Z$ 、 $X \cdot Z$ 、 $X \cdot Y$ 。 $Y \cdot Z$ 覆盖的 $\bar{X} \cdot Y \cdot Z$ 、 $X \cdot Z$ 覆盖的 $X \cdot \bar{Y} \cdot Z$ 、 $X \cdot Y$ 覆盖的 $X \cdot Y \cdot \bar{Z}$ 都没有被其他质蕴含项所覆盖。
- ◆ 质蕴涵项覆盖的最小项越多（即卡诺圈越大）越可能是实质蕴涵项。

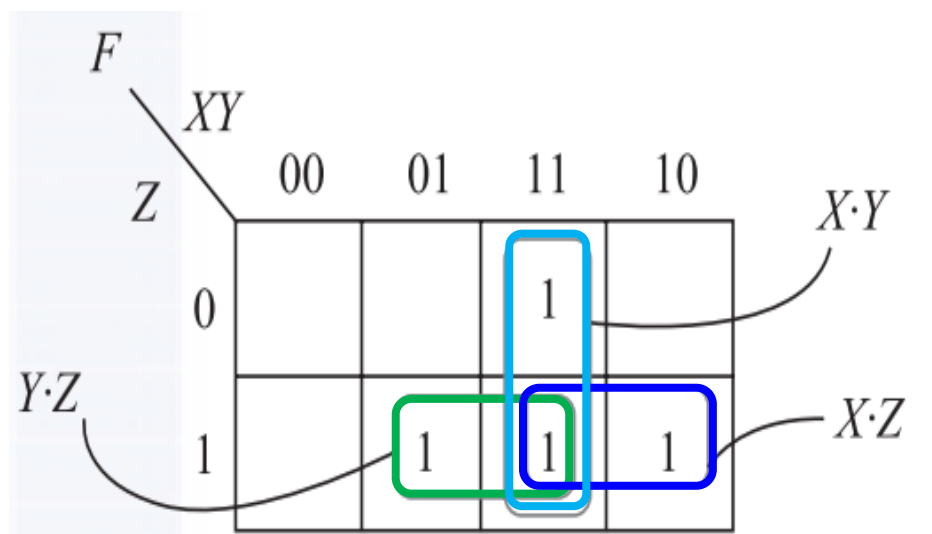
4.2 卡诺图化简

- ◆ 若逻辑函数的所有最小项都被一组质蕴涵项所覆盖，则该组质蕴涵项称为**函数的一个覆盖 (Cover)**，它一定包含**所有的实质蕴涵项**。

如， $\{Y \cdot Z, X \cdot Z, X \cdot Y\}$ 是 $F(X, Y, Z) = \sum m(3, 5, 6, 7)$ 的一个覆盖，包含所有实质蕴含项

- ◆ **最小覆盖**：包含**质蕴涵项数最少**，并且质蕴涵项中的**变量总数也是最少**。
- ◆ 逻辑函数化简问题就转化为寻找函数的**最小覆盖**问题。

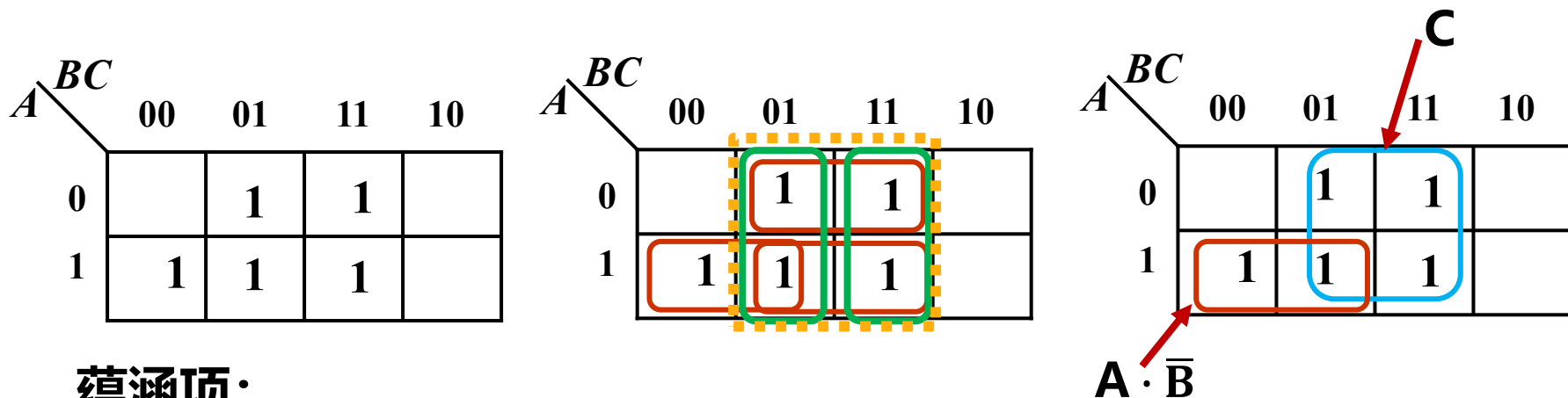
$$F(X, Y, Z) = Y \cdot Z + X \cdot Z + X \cdot Y$$



最小覆盖对应的逻辑表达式称为：最简逻辑表达式

4.2 卡诺图化简

- 确定逻辑函数 $F(A,B,C)=\sum m(1,3,4,5,7)$ 的最小覆盖的方法



蕴涵项:

最小项: $\{\bar{A} \cdot \bar{B} \cdot C, \bar{A} \cdot B \cdot C, A \cdot \bar{B} \cdot \bar{C}, A \cdot \bar{B} \cdot C, A \cdot B \cdot C\}$

含有两个最小项的蕴涵项: $\{A \cdot \bar{B}, \bar{A} \cdot C, A \cdot C, \bar{B} \cdot C, B \cdot C\}$

含有四个最小项的蕴涵项: $\{C\}$

被蕴含项C覆盖

质蕴涵项, 没有被覆盖的蕴涵项: $\{A \cdot \bar{B}, C\}$

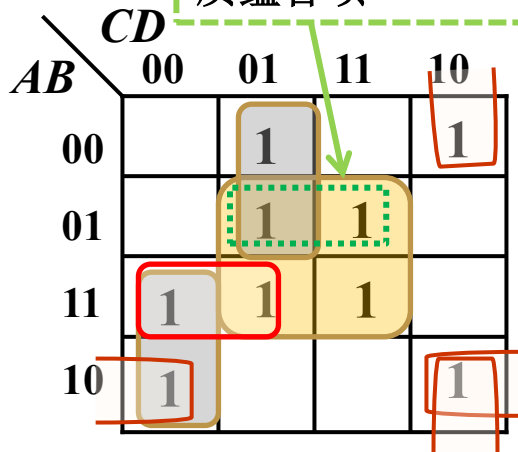
实质蕴涵项 = $\{A \cdot \bar{B}, C\}$ 。 $A \cdot \bar{B}$ 覆盖了不能被C覆盖的 m_4 ; C覆盖了不能被 $A \cdot \bar{B}$ 覆盖的 m_1 、 m_3 、 m_7

最小覆盖 = $A \cdot \bar{B} + C$

4.2 卡诺图化简

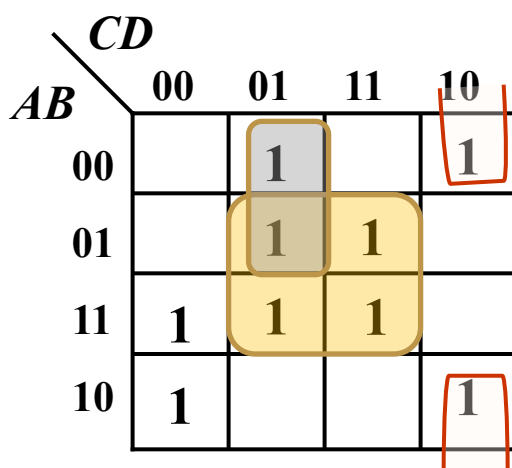
◆ $F(A,B,C,D) = \sum m(1,2,5,7,8,10,12,13,15)$

这是蕴含项，但不是
质蕴含项



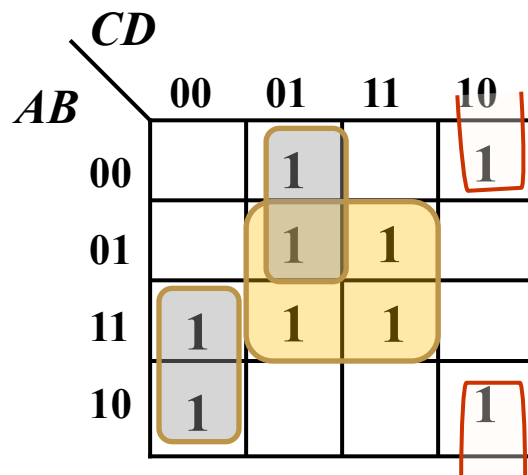
找出所有质蕴涵项

$$B \cdot D, \bar{A} \cdot \bar{C} \cdot D, \bar{B} \cdot C \cdot \bar{D} \\ A \cdot B \cdot \bar{C}, A \cdot \bar{C} \cdot \bar{D}, A \cdot \bar{B} \cdot \bar{D}$$



选择所有的实质蕴涵项

$$B \cdot D, \bar{A} \cdot \bar{C} \cdot D, \bar{B} \cdot C \cdot \bar{D}$$



在剩余质蕴含项中
选择**最小覆盖子集**

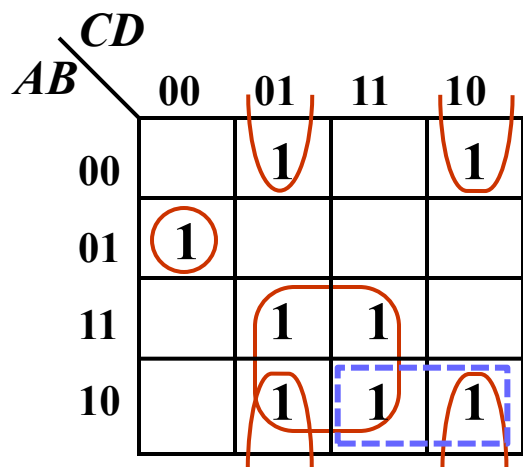
$$A \cdot \bar{C} \cdot \bar{D}$$

合并实质蕴含项和最小覆盖子集，得到最简逻辑表达式：

$$F(A,B,C,D) = B \cdot D + \bar{A} \cdot \bar{C} \cdot D + \bar{B} \cdot C \cdot \bar{D} + \mathbf{A \cdot \bar{C} \cdot \bar{D}}$$

4.2 卡诺图化简

$$Y(A, B, C, D) = \sum m(1, 2, 4, 9, 10, 11, 13, 15)$$



这是质蕴含项，但不是实质蕴含项

$$Y = A \cdot D + \overline{B} \cdot \overline{C} \cdot D + \overline{B} \cdot C \cdot \overline{D} + \overline{A} \cdot B \cdot \overline{C} \cdot \overline{D}$$

4.2 卡诺图化简

- ◆ 利用对偶性原理，卡诺图也可以用来化简**和之积表达式**，只需要将真值表中输出值为0的最大项对应的单元标注为0，然后合并相邻的“0 单元”，得到求和的质蕴涵项。
- ◆ 卡诺图化简优点
 - 方便、直观、容易掌握
- ◆ 卡诺图化简缺点
 - 受到变量数量的约束，当变量数大于6时，卡诺图绘制以及相邻关系的识别将变得非常复杂，从而导致难以直观化简。

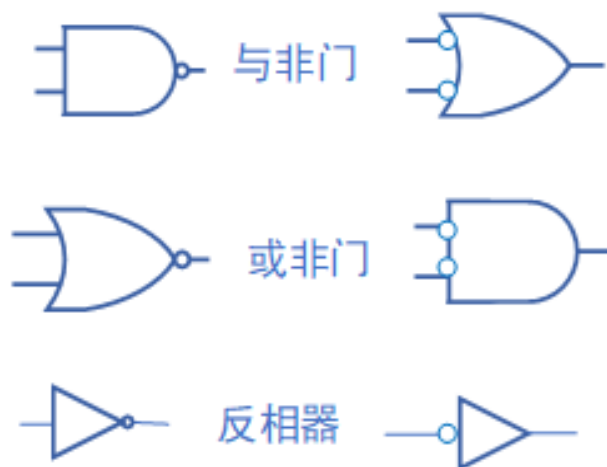
在现代数字系统设计中，大多采用编程实现**逻辑函数的自动化简**。

4.3 逻辑函数变换——实际电路实现方式

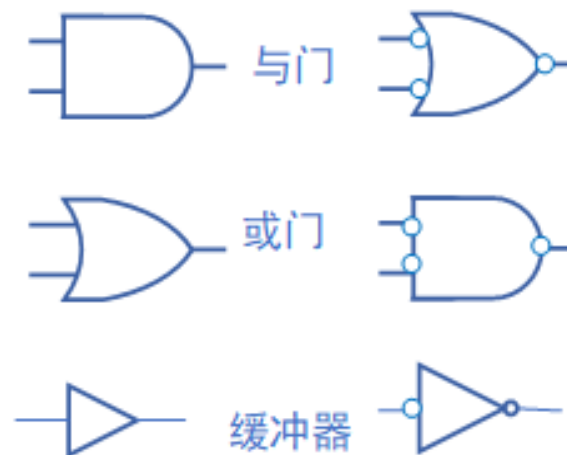
◆ **等效逻辑符号**：功能相同，符号不同。

- 反相输出门利用一次德·摩根定理，转换为非反相输出门。
- 非反相输出门**两次取反**，利用德·摩根定理**转换下层的取反运算**，可得到反相输出门。

$$XY \rightarrow \overline{\overline{XY}}$$



(a)反相门的等效符号

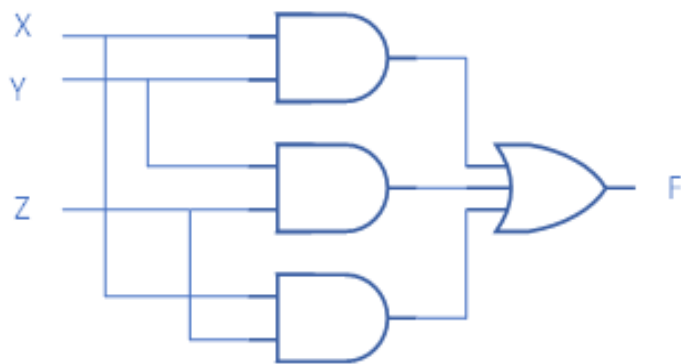


(b)非反相门的等效符号

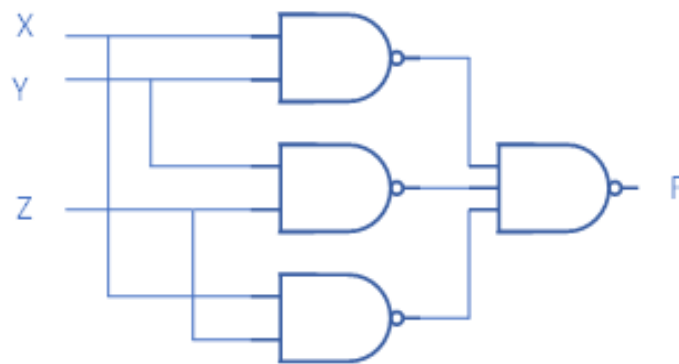
4.3 逻辑函数变换——实际电路实现方式

- ◆ 在数字电路中，**与非门和或非门**通常比**与门和或门**的执行**速度快**。
- ◆ 将“与-或”表达式转换为“与非-与非”表达式
 - 将“与-或”表达式整体**两次取反**，然后运用**德摩根定律转换下层的取反运算**，就可以得到“与非-与非”表达式
 - 使用**与非门**替代**与门**和**或门**来实现逻辑函数
 - 例如： $F(X,Y,Z)=X \cdot Y + Y \cdot Z + X \cdot Z$ ，转换为：

$$F(X,Y,Z)=\overline{\overline{X \cdot Y + Y \cdot Z + X \cdot Z}}=\overline{\overline{X \cdot Y} \cdot \overline{Y \cdot Z} \cdot \overline{X \cdot Z}}$$



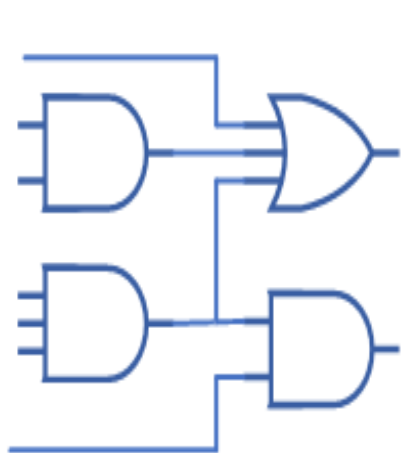
(a)与-或电路



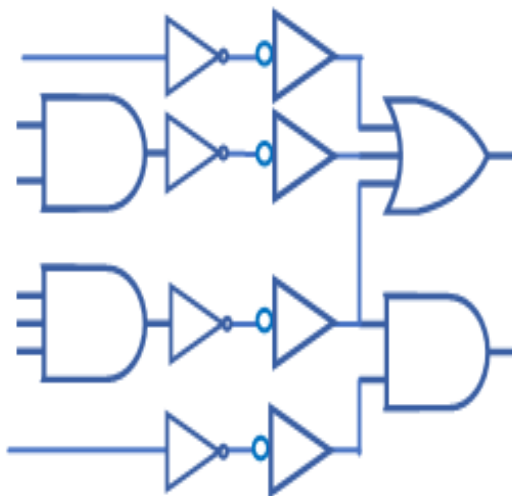
(b)与非-与非电路

4.3 逻辑函数变换

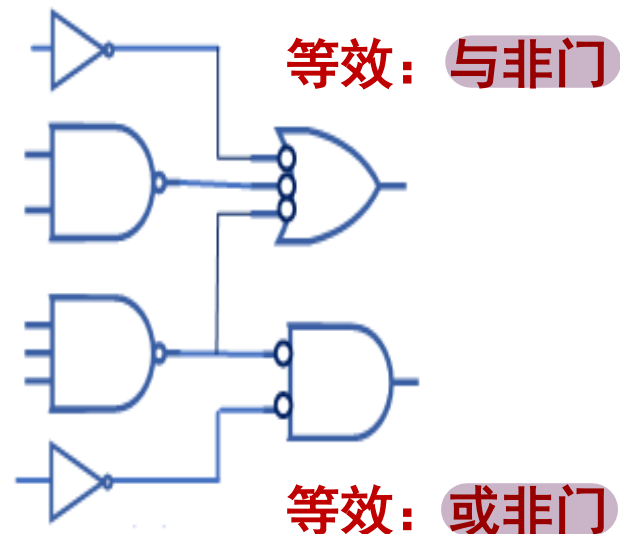
- ◆ 任何**积之和**表达式：都可使用“**与-或**”电路和“**与非-与非**”电路这两种方法实现。
- ◆ 任何**和之积**表达式：都可使用“**或-与**”电路和“**或非-或非**”电路这两种方法实现。
- ◆ 任何两级电路：在第一级的输出和第二级的输入之间加入**一对反相器**，来实现用反相门替代与门和或门。



(a) 初始电路



(b) 加入反相器对的电路



(c) 使用反相输出端和反向输入端的电路

第2章总结

- ◆ 现实世界的模拟信号需转换为数字信号。数字系统中的所有信号都是二值的，用0和1表示
 - ◆ **逻辑门**是最基础的数字电路，可通过**CMOS晶体管**实现
 - ◆ 最基本的逻辑运算有与、或、非三种运算，对应的逻辑门分别为与门、或门和非门
 - ◆ **布尔代数**是数字系统分析和设计的基础理论工具，对应的公理系统和定理可对逻辑表达式进行化简，实现逻辑函数间的相互转换
- ◆ 通常使用**真值表**、**波形图**以及**逻辑表达式**来描述逻辑变量间的关系
 - ◆ 可使用**代数法**、**卡诺图**等来化简逻辑表达式
 - ◆ 在实现数字系统时，为了提高速度、降低成本，通常利用**与非门**和**或非门**来构建电路。

作业：习题3、5、7(5)和(6)、8(1)和(6)、12、13(2)、13(5)
10月5日24:00之前提交。即将开放前两个实验。截止日期未定