NJUAI 南京大学 人工智能学院

SCHOOL OF ARTIFICIAL INTELLIGENCE, NANJING UNIVERSITY

第2章 数字逻辑基础

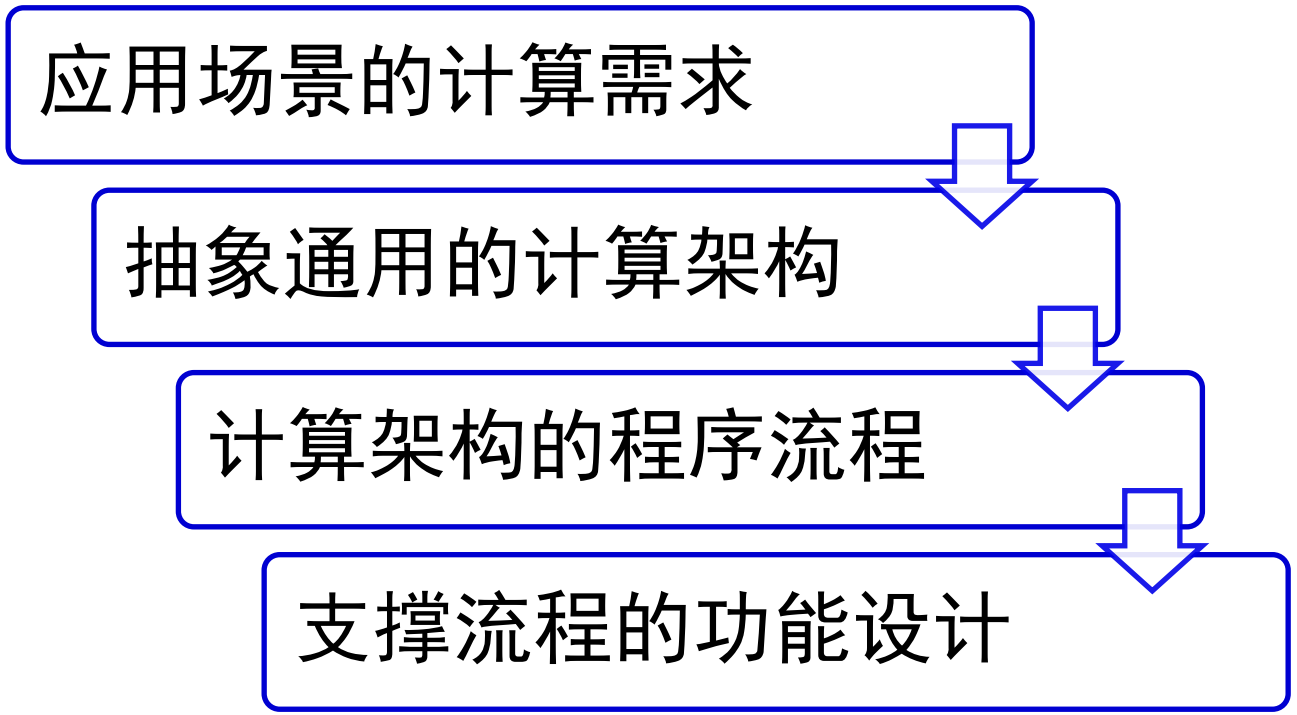
第二讲 布尔代数

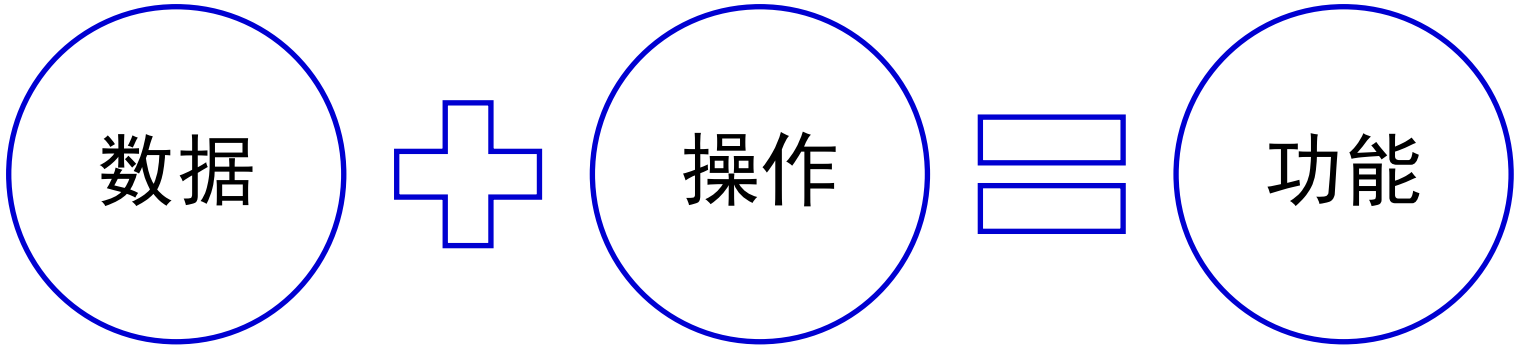
第一讲 逻辑门和数字抽象

第三讲 逻辑关系描述

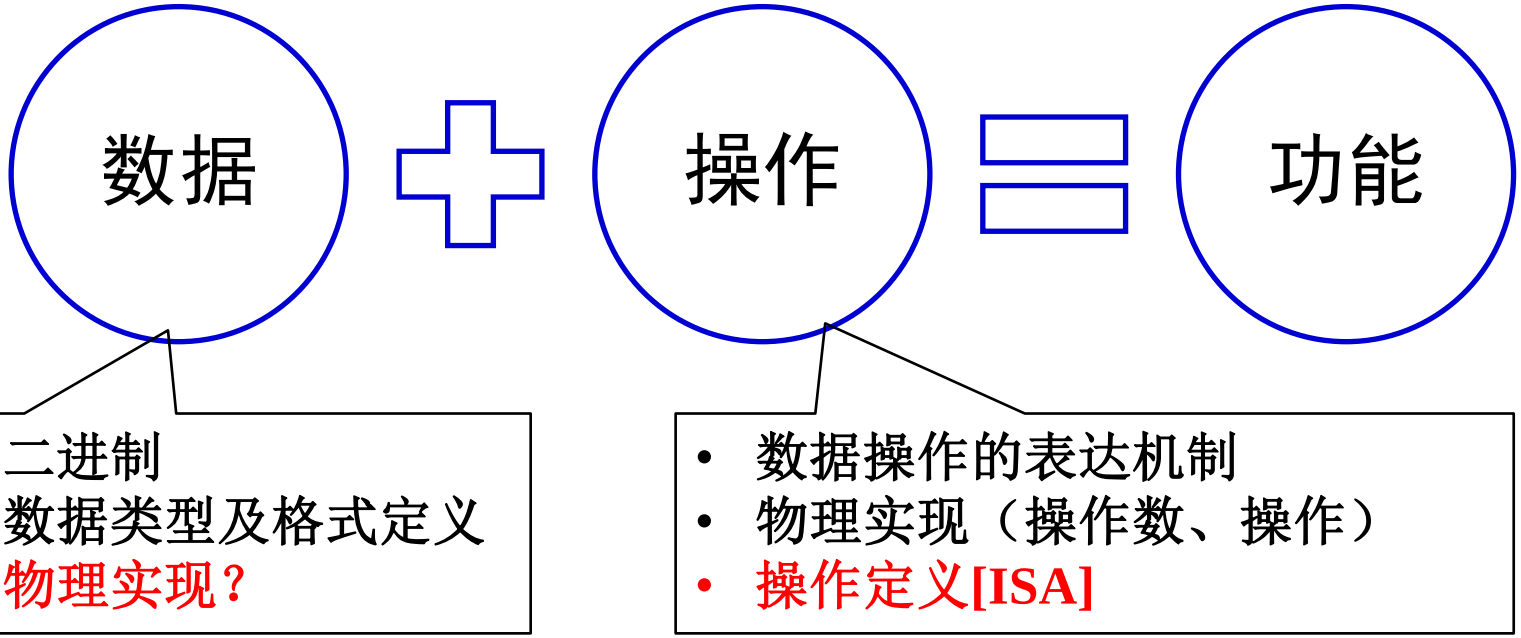
第四讲 逻辑函数化简

章节导引





章节导引



- 数据操作可以抽象为：

$$F(X1,X2...Xn)=Y$$

其中Xi和Y都是二进制向量，我们还可以把这个公式改写成：

$$fi(X1,X2...Xn)=yi \rightarrow fi(z1,z2,z3...zm)=yi$$

其中，yi 和zi 是二进制位变量，fi 就是一个（0，1）定义域的函数。

- 操作 F 可以用布尔代数来刻画。

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

- 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
  - *An Investigation of the Laws of Thought，on which are Founded the Mathematical Theories of Logic and Probabilities*
  - “基于人类逻辑思考的本性”，将人类思想转换成符号，并指出这些符号只需要两个值：真、假
- 布尔代数在代数学（代数结构）、逻辑演算、集合论、拓扑空间理论、测度论、概率论、泛函分析等数学分支中均有应用。

2 布尔代数

- 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；
  - 只有两种取值：“真”或“假”；
    - “真”记作“1”，数字电路中表示为高电平；
    - “假”记作“0”，数字电路中表示为低电平；
  - 0和1不表示数值的大小，只表示完全相反的两种状态。
- 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。
- 逻辑函数：表明输入和输出变量之间的逻辑关系。

2 布尔代数

◆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，符号“•”、“∧”,Verilog：“&”

或运算：析取、逻辑加，符号“+”、“∨”,Verilog：“|”

非运算：否定、取反，符号“ ¯”、“¯/~/¬”,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“•”，不建议省略。

2.1 公理系统

- 用符号X、Y、Z表示逻辑变量。
- 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$\overline{x}=1$； (A2D)如果$x=1$，则$\overline{x}=0$
- 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)

2.2 定理

- 单变量定理
  - 一致性 (T1) $X \cdot 0 = 0$ (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

2.2 定理

- 二变量和三变量定理
  - 交换律 (T6) X+Y=Y+X　　(T6D) X•Y=Y•X
  - 结合律 (T7) (X+Y)+Z=X+(Y+Z)
    (T7D) (X•Y) •Z=X•(Y•Z)
  - 分配律 (T8) X•Y +X•Z=X •(Y+Z)
    (T8D) (X+Y) •(X+Z)=X+Y•Z
  - 吸收律 (T9) X+ X•Y =X　　(T9D) X•(X+Y)=X
  - 组合律 (T10) X•Y + X•$\overline{Y}$ =X (T10D) (X+Y) •(X+$\overline{Y}$) =X
  - 一致律 (T11) X•Y+ $\overline{X}$ • Z+Y • Z =X • Y+ $\overline{X}$ • Z
    (T11D) (X+Y) •($\overline{X}$ +Z) •(Y+Z)=(X+Y) •($\overline{X}$ +Z)

与算术运算规则不同！

称为一致项/冗余项　在组合电路用来消除时序冒险。

2.2 定理

- 证明定理T9（方法有多种）

$$X+X\cdot Y = X\cdot 1+X\cdot Y$$
$$= X\cdot(1+Y)$$
$$= X\cdot 1$$
$$=X$$

- 证明T9D

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

请证明以下公式：

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

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

2.2 定理

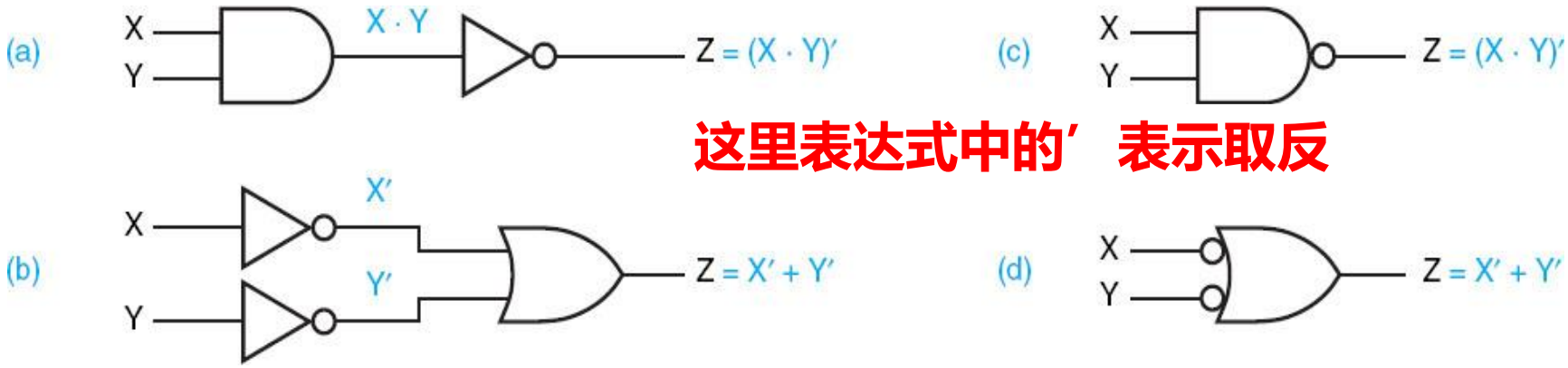
- n变量定理
  - 广义同一律 (T12)　　X+X+…+X=X
    (T12D)　X •X • …•X = X
  - 德•摩根定理De Morgan’ s Theorem
    (T13)　$\overline{X_1 \cdot X_2 \cdot \cdots \cdot X_n} = \overline{X_1} + \overline{X_2} + \cdots + \overline{X_n}$
    (T13D)　$\overline{X_1 + X_2 + \cdots + X_n} = \overline{X_1} \cdot \overline{X_2} \cdot \cdots \cdot \overline{X_n}$
  - 广义德•摩根定理
    (T14)　$\overline{F(X_1, X_2, \cdots, X_n, +, \cdot)} = F(\overline{X_1}, \overline{X_2}, \cdots, \overline{X_n}, \cdot, +)$
  - 香农定理　　用于多变量函数的实现
    (T15)　$F(X_1, X_2, \cdots, X_n) = X_1 \cdot F(1, X_2, \ldots, X_n) + \overline{X_1} \cdot F(0, X_2, \ldots, X_n)$
    (T15D) $F(X_1, X_2, \cdots, X_n) = [X_1 + F(0, X_2, \ldots, X_n)] \cdot [\overline{X_1} + F(1, X_2, \ldots, X_n)]$

n个变量　　n-1个变量

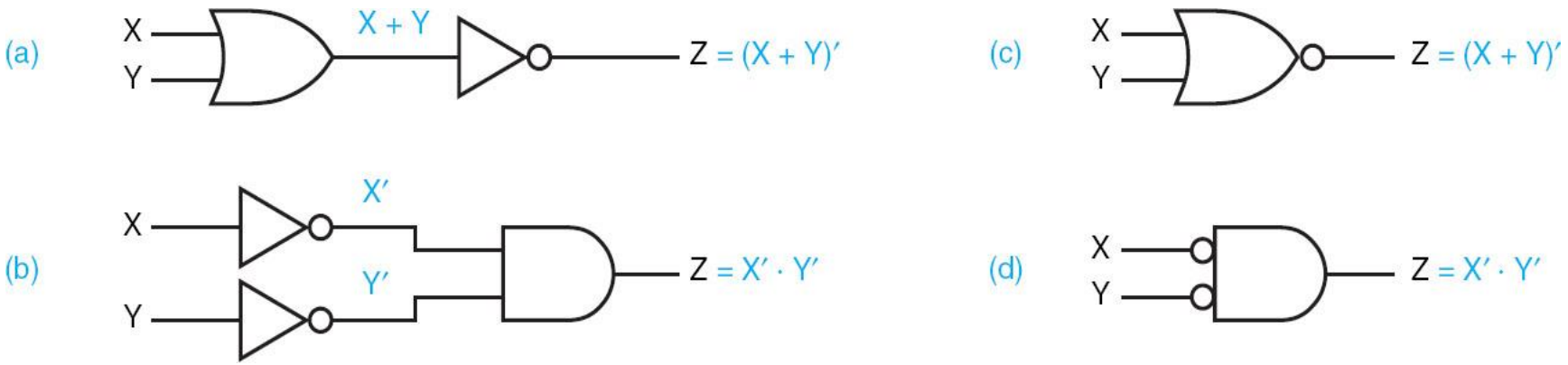
2.2 定理

注意反相圈的位置，逻辑门的符号和逻辑门的名称之间的关系。

- **德•摩根定理的应用**



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

2.2 定理

- 德•摩根定理的应用例 f(a,b,c)布尔函数变换

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

2.3 对偶定律

- 对于任何一个逻辑表达式Y，若将其中的“·”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

$$X+X\cdot Y=X \quad \xrightarrow{\text{对偶式}} \quad X\cdot(X+Y)=X$$

2.4 反演定律*

- **反演定律：**对于任意一个逻辑表达式Y，若将其中所有的“•”与“+”互换，“0”和“1”互换，原变量与反变量互换，则得到的结果就是原函数的反函数$\overline{Y}$ 。
  - 需遵守“先括号，然后与，最后或”的运算优先次序
  - 不属于单个变量上的取反运算保留不变
  - DeMorgan定理是反演定理的一个特例

注意：与数学概念中的“反函数”有区别，也有人称之为补函数。

南京大学 人工智能学院
SCHOOL OF ARTIFICIAL INTELLIGENCE, NANJING UNIVERSITY

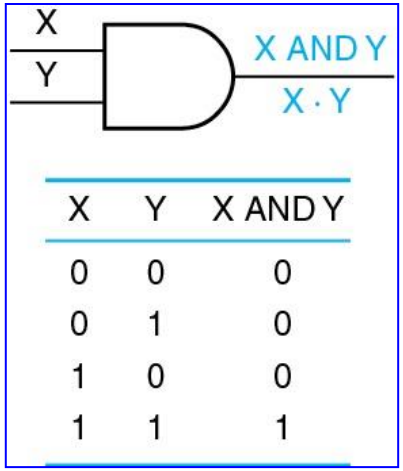
第一讲 逻辑门和数字抽象

- 数据操作可以用布尔函数表达。
- 布尔函数由几个基本操作构建
- 基本操作的物理实现？

- **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，是输入和输出信号间逻辑关系的物理实现电路，能够对输入信号实现基础的共性逻辑操作。
  - 一个或多个输入信号
  - 一个输出信号
  - 具有允许或禁止信号传输的功能，也称为门电路。
- 输入信号和输出信号之间逻辑关系使用真值表或者逻辑表达式来描述。
  - 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其对应的输出信号值。
  - 逻辑表达式就是用逻辑运算符来连接逻辑变量。
    - 与、或、非运算有自己的运算符



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

1.1 逻辑门

- 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。
    - 输入信号、输出信号称为逻辑变量。
    - 输入信号的取值是0或1，逻辑运算的结果也是0或1。
- 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。

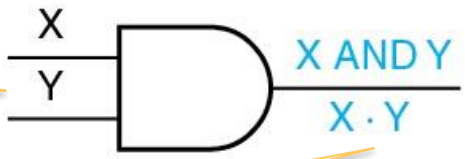
1.1 逻辑门

- 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“•”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“—”表示，称为非运算或者取反运算。

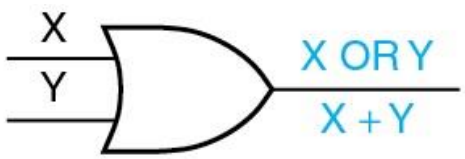
逻辑符号

逻辑表达式

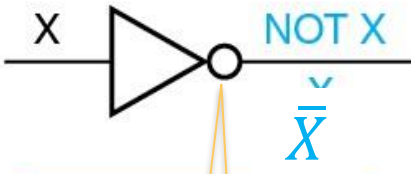
真值表



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

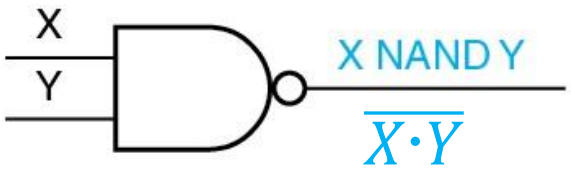


X	NOT X
0	1
1	0

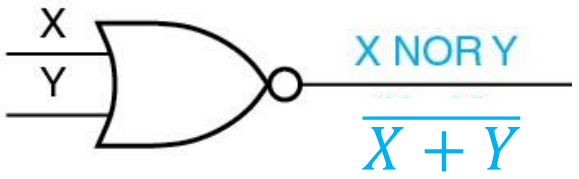
反相圈

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，输信号出才为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

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

X	Y	X⊕Y
0	0	0
0	1	1
1	0	1
1	1	0

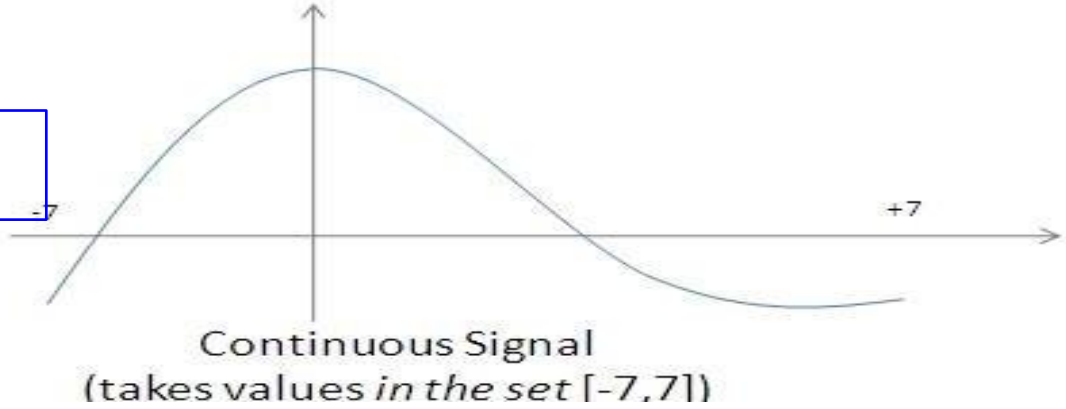
(a)异或门

X	Y	X⊙Y
0	0	1
0	1	0
1	0	0
1	1	1

(b)同或门

1.2 数字抽象

- 模拟信号如何变成数字量



◆ **模拟信号**

- **连续的**
  - **时间上的连续: 任意时刻有一个相对应的值**
  - **数值上的连续: 任意时刻可以是在一定范围内的任意值**
- **例如：电压，电流，温度，亮度，颜色等**
- **缺点**
  - **精度受限，很难度量**
  - **不稳定，容易受噪声的干扰**
  - **难以保存**
- **优点：表示事物比较准确**

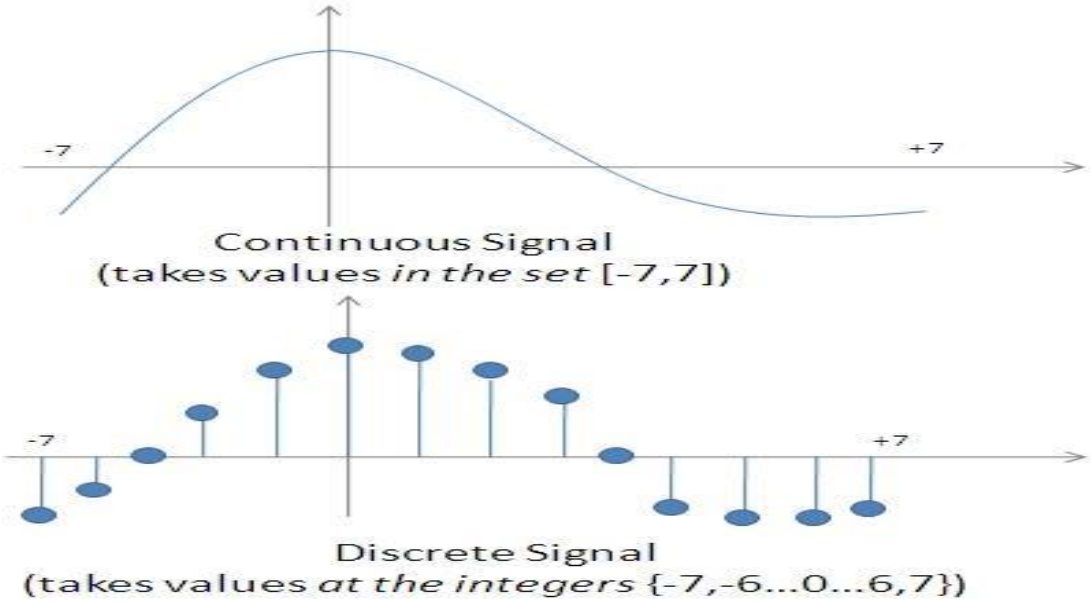
自然世界是模拟的！

1.2 数字抽象

- **数字信号**
  - **非连续（离散）**
    - **时间上离散**

      只在某些时刻有定义
    - **数值上离散**

      只能是有限集合中的一个值
  - **如数字声音、图像、视频、电影特技等**
  - **优点**
    - **结果重复**
    - **设计简单**
    - **功能灵活**
    - **可编程HDL**



数字系统随处可见！

数字系统在底层实现时是模拟信号！

核心：知道何时需考虑模拟特性！

模拟信号 → **采样、量化、编码** → **数字信号**

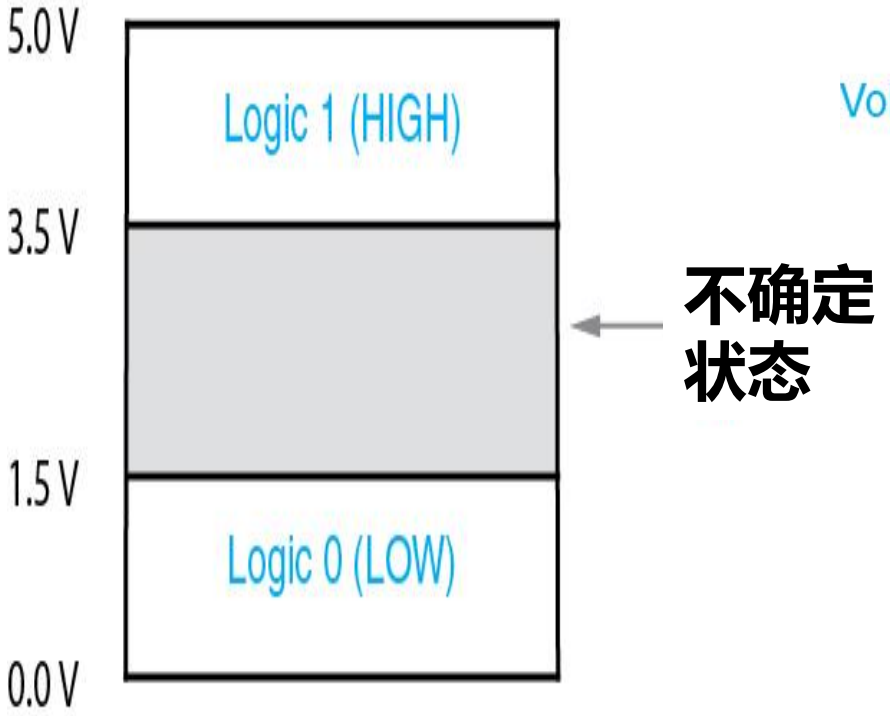
1.2 数字抽象

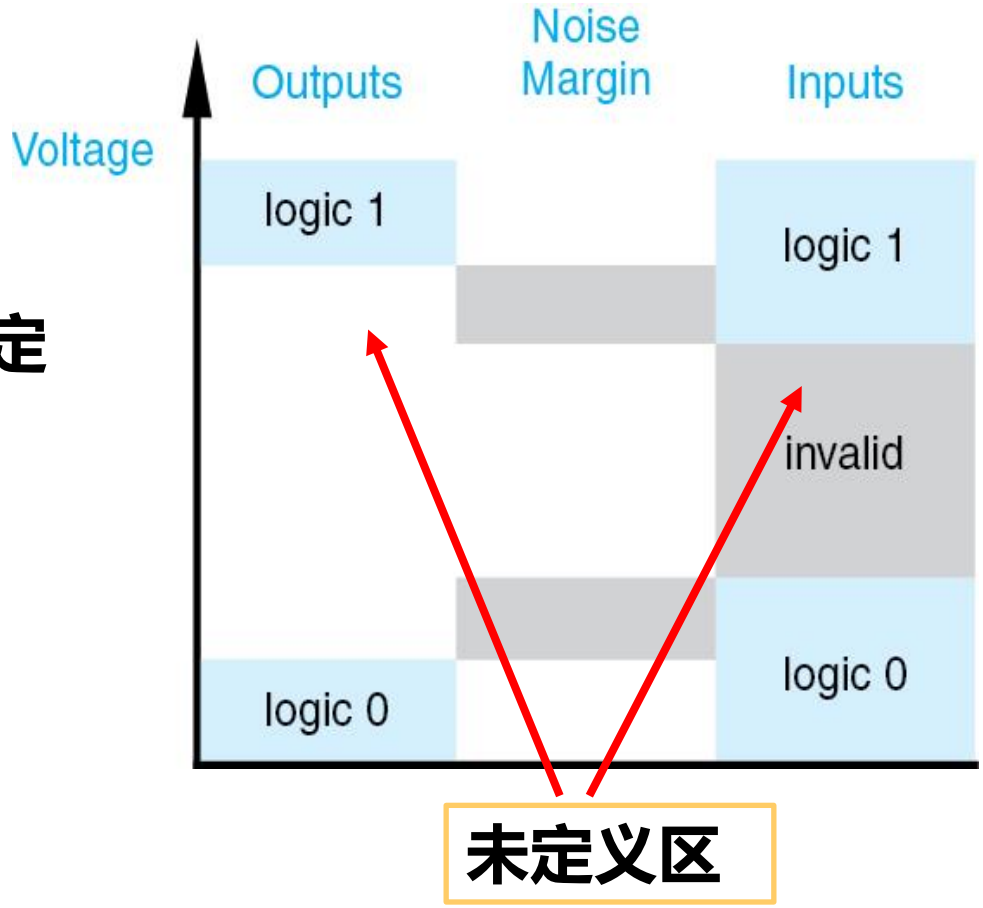
- 基本逻辑信号0和1的物理表征：
  - 逻辑抽象：将物理量实际值的无穷集映射为两个子集，对应于两个状态或两个逻辑值0和1，从而隐藏模拟量的物理特性。
  - 明确可检测：物理量如何明确可靠地检测出0、1状态?
    - 设定阈值范围/未定义区。

逻辑值	正逻辑 Postitive 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

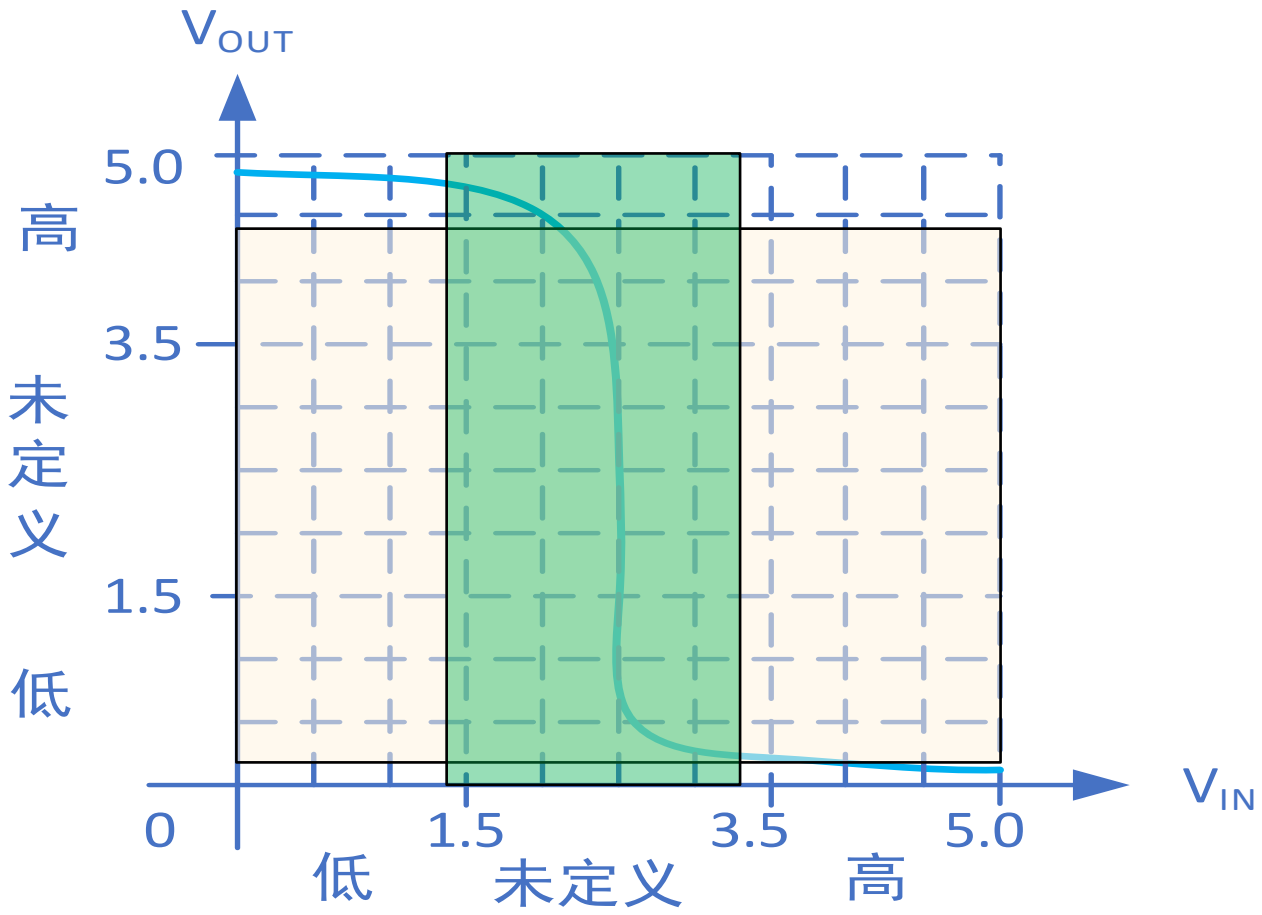
- 输入、输出电平的逻辑采样





1.2 数字抽象

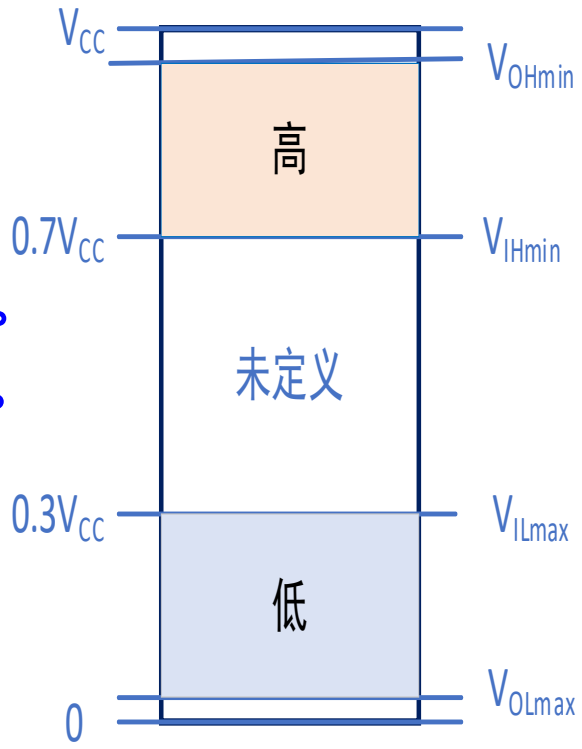
- 逻辑门电路的输出电压受到**负载**及**噪声**的影响。
- 输出电压必须能够被其他逻辑门的输入端**准确识别**。



非门典型的输入- 输出传输特性图

1.2 数字抽象

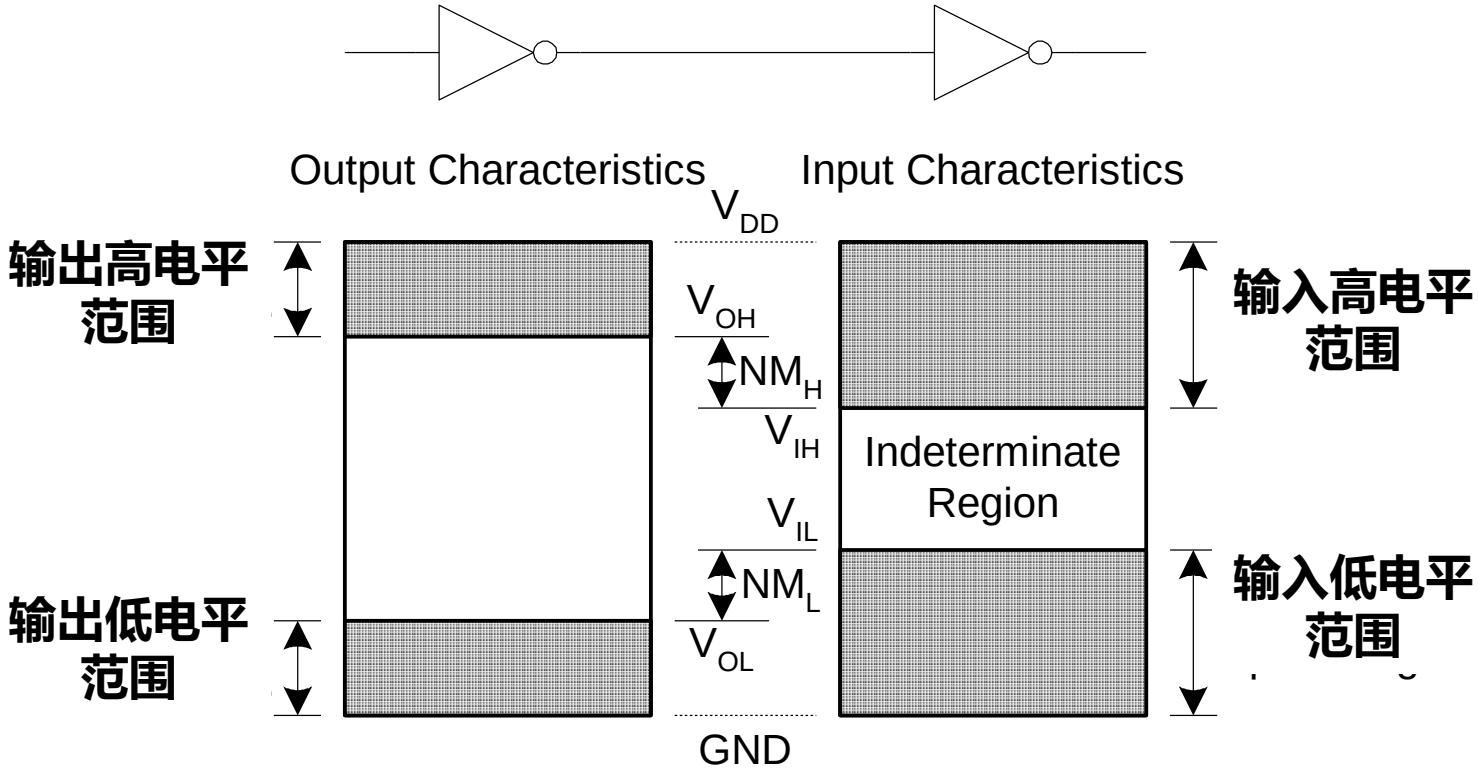
- 输入电压主要由晶体管的**开关阈值电压**决定，而输出电压则主要由晶体管**导通时的电阻**决定。
  - V_{OHmin}：输出为高态时的最小输出电压值。
  - V_{OLmax}：输出为低态时的最大输出电压值。
  - V_{IHmin}：确保能被识别为高态的最小输入电压值。
  - V_{ILmax}：确保能被识别为低态的最大输入电压值。



- V_{CC}/V_{DD}称为电源电压，典型值为5.0V±10%
- GND称为地线。
- 电路电平参数的典型数值如下：
  - V_{OHmin}：V_{CC}-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。

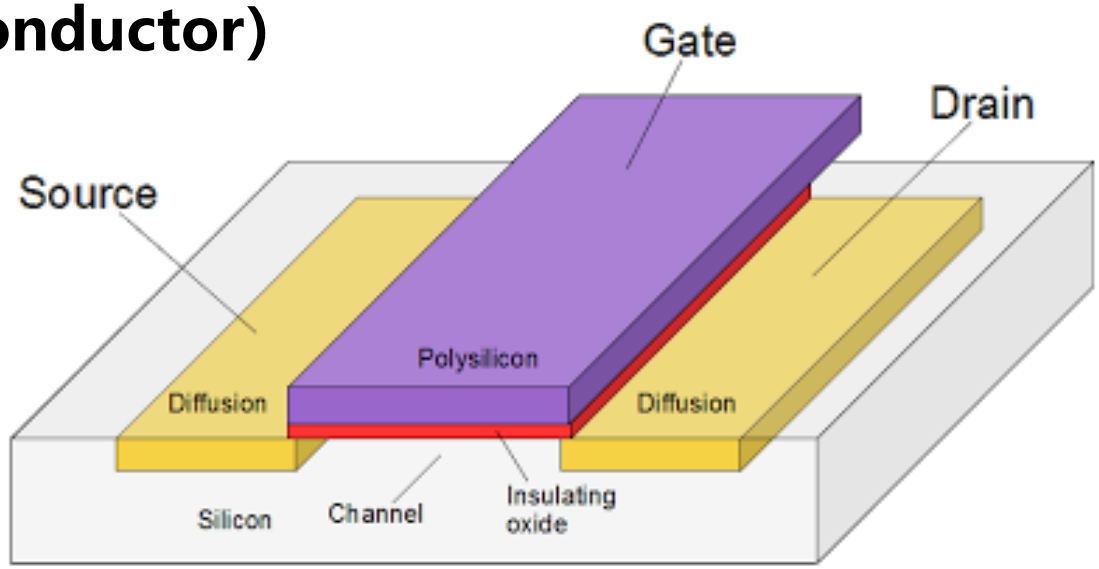
1.2 数字抽象

- **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最差输出电压被破坏，成为不可被输入端识别的值。**
  - **高态直流噪声容限$NM_H=V_{OHmin}-V_{IHmin}$**
  - **低态直流噪声容限$NM_L=V_{ILmax}-V_{OLmax}$**

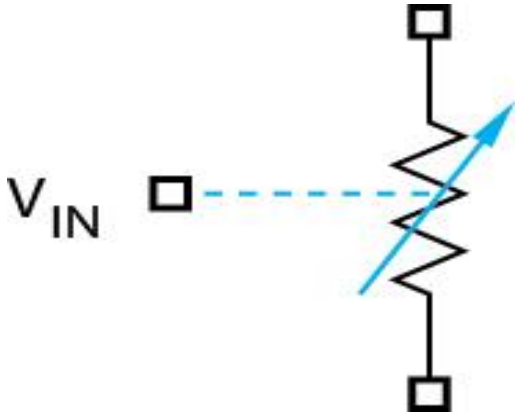


1.3 CMOS晶体管

- 基于金属氧化物半导体场效应晶体管的CMOS（Complementary Metal-Oxide Semiconductor)
- MOS晶体管三极：
  - 栅极gate
  - 源极source
  - 漏极drain



- 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。
- MOS晶体管分为：
  - n沟道型NMOS，N型杂质有磷或者锑
  - p沟道型PMOS，P型杂质有硼或者铟

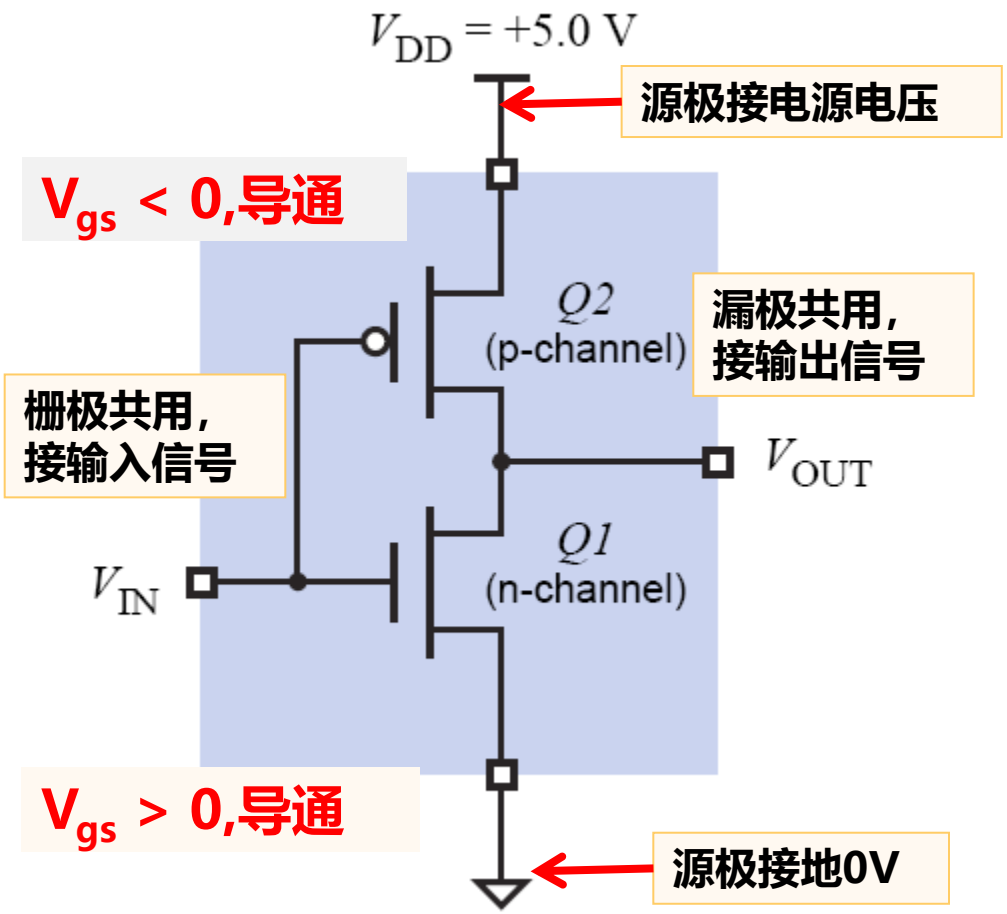


1.3 CMOS晶体管

- CMOS 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，分别连接到输入和输出信号。
  - 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，从而改变漏极的输出电压值。
  - 可以看成电压控制开关
- 常用CMOS门电路
  - 反相器 / 与非门 / 或非门 / 与或非门 / 或与非门

1.3 CMOS晶体管

- **非门使用一对CMOS 晶体管实现**



原理图

V_{IN}=L

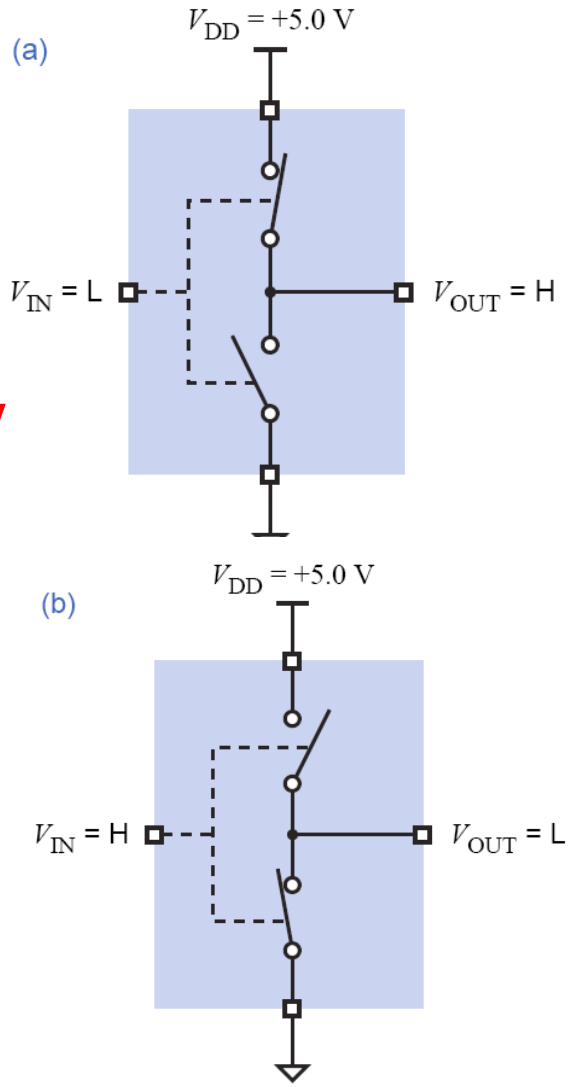
V_{gs1} = 0

V_{gs2} =-5V

V_{IN}=H

V_{gs1} =5V

V_{gs2} = 0



开关模型

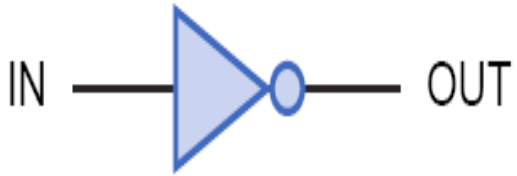
1.3 CMOS晶体管

(b)

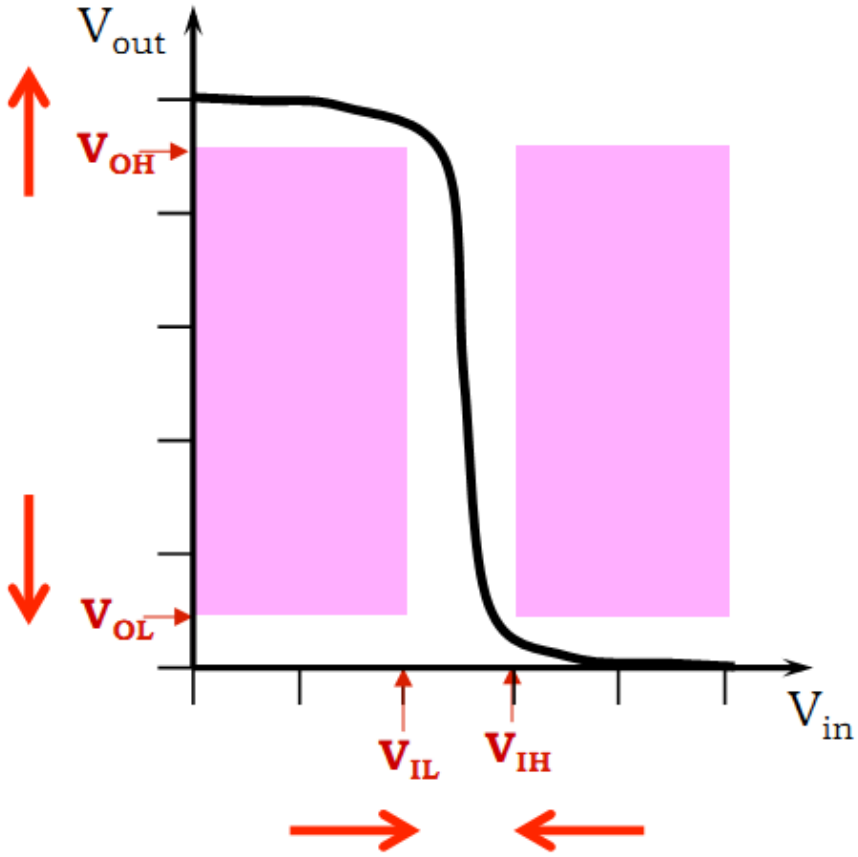
V_{IN}	$Q1$	$Q2$	V_{OUT}
0.0 (L)	off	on	5.0 (H)
5.0 (H)	on	off	0.0 (L)

功能表

(c)

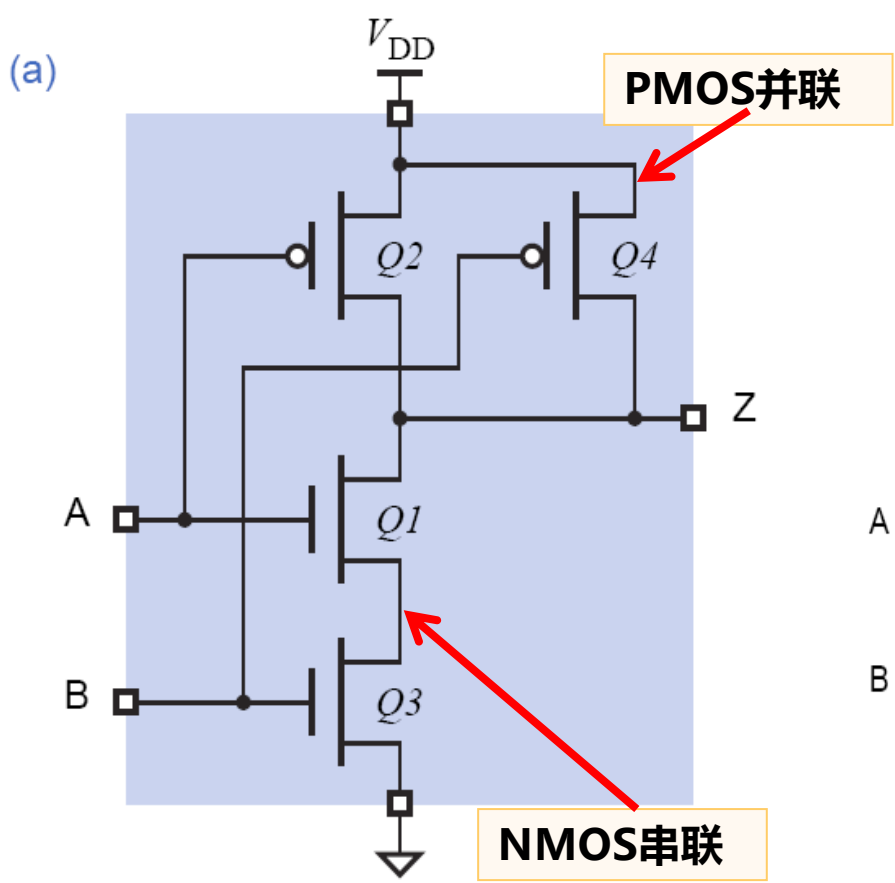


逻辑符号

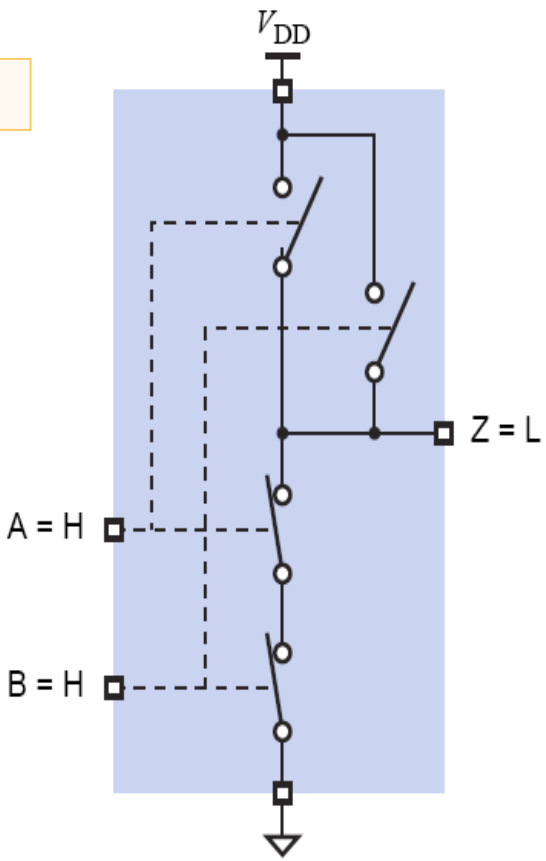


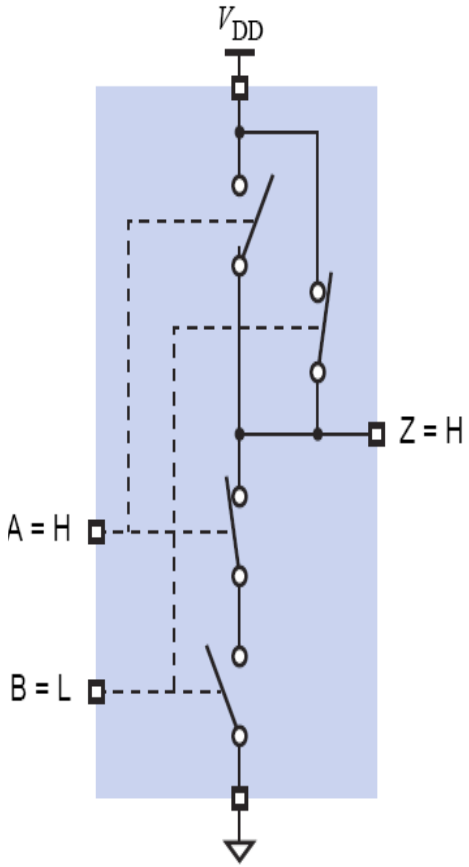
1.3 CMOS晶体管

- 2输入与非门使用两对CMOS晶体管实现
  - NMOS管串联
  - PMOS管并联



原理图



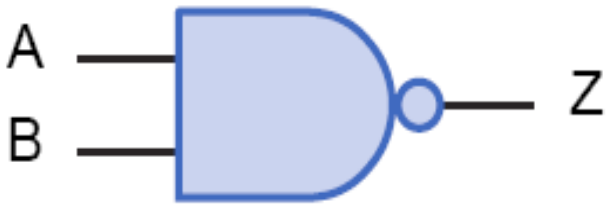


开关模型

1.3 CMOS晶体管

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



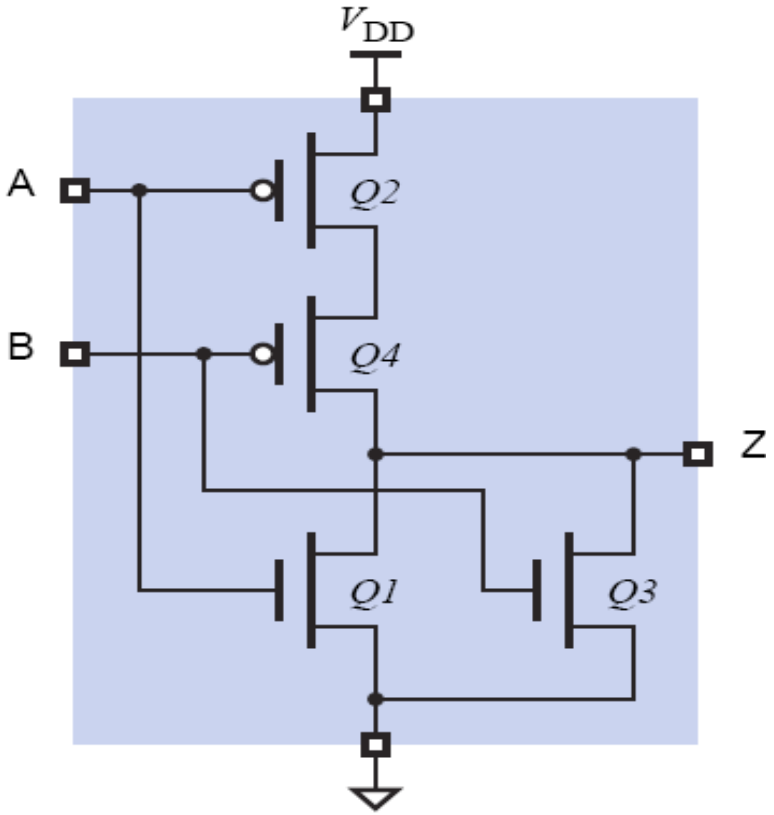
逻辑符号

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

- 2输入或非门使用两对CMOS晶体管实现
  - NMOS管并联
  - PMOS管串联



原理图

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

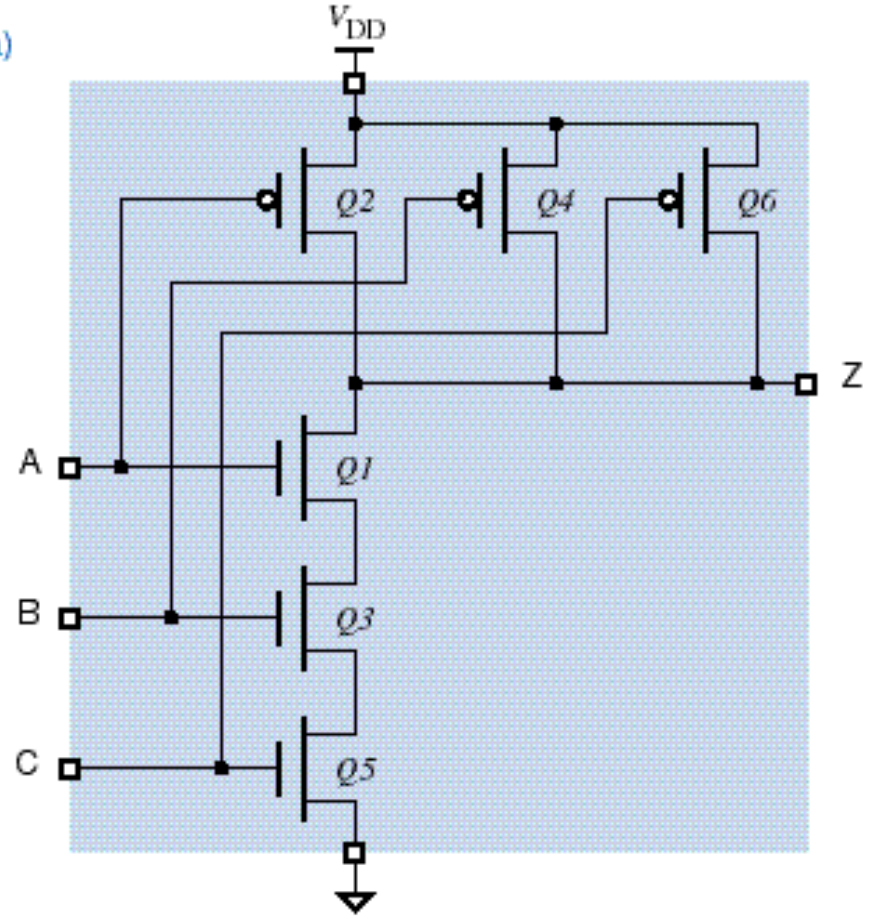
逻辑符号

CMOS的或门如何得到?

1.3 CMOS晶体管

- 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

(a)

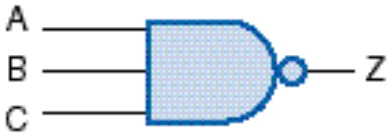


3 输入与非门

(b)

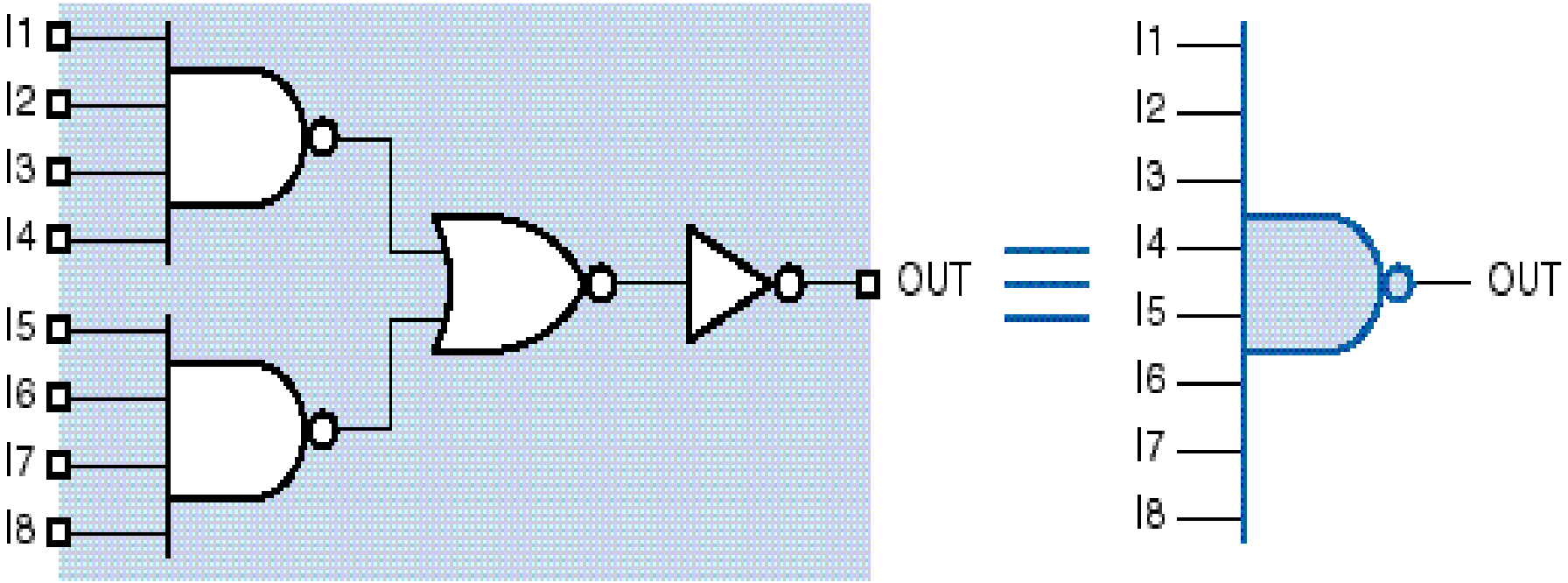
A	B	C	Q1	Q2	Q3	Q4	Q5	Q6	Z
L	L	L	off	on	off	on	off	on	H
L	L	H	off	on	off	on	on	off	H
L	H	L	off	on	on	off	off	on	H
L	H	H	off	on	on	off	on	off	H
H	L	L	on	off	off	on	off	on	H
H	L	H	on	off	off	on	on	off	H
H	H	L	on	off	on	off	off	on	H
H	H	H	on	off	on	off	on	off	L

(c)



1.3 CMOS晶体管

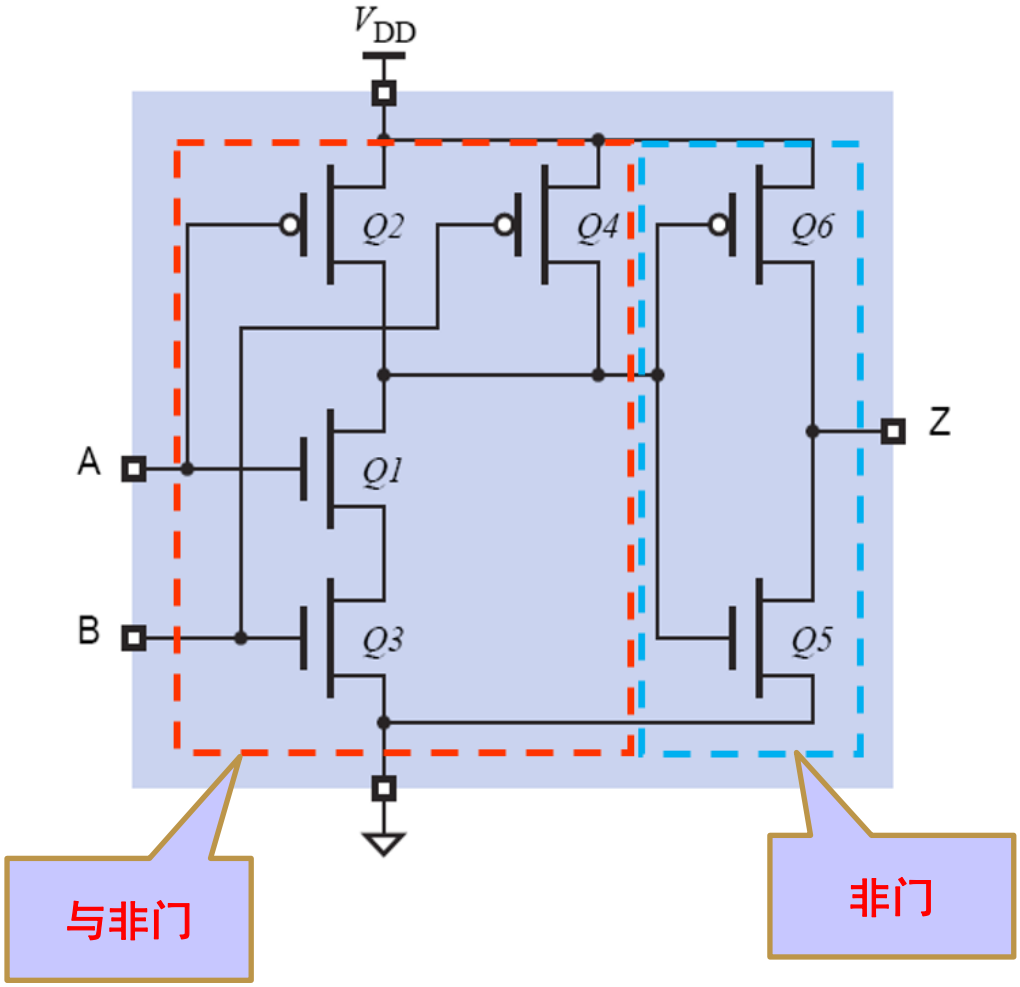
- 受扇入系数等电气特性的限制，输入端不能无限制增加。
- 一般输入端数目不超过8 个。
- 输入端较多的门电路可用输入端较少的门电路级联而构成，速度更快、体积更小



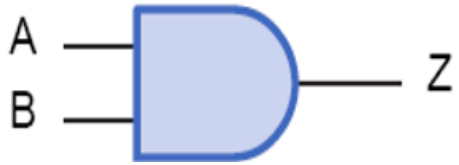
8 输入与非门

1.3 CMOS晶体管

- 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。



A	B	Q1	Q2	Q3	Q4	Q5	Q6	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

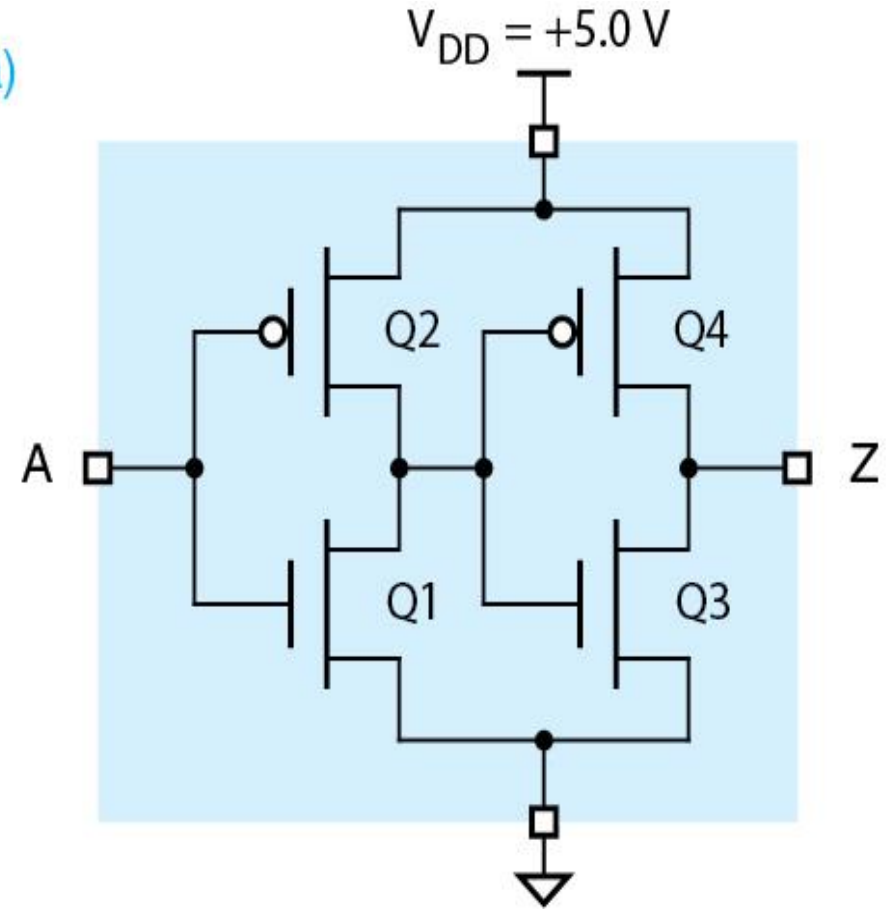


2 输入与门

1.3 CMOS晶体管

- 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

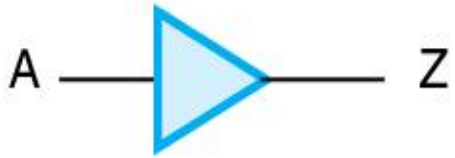
(a)



(b)

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

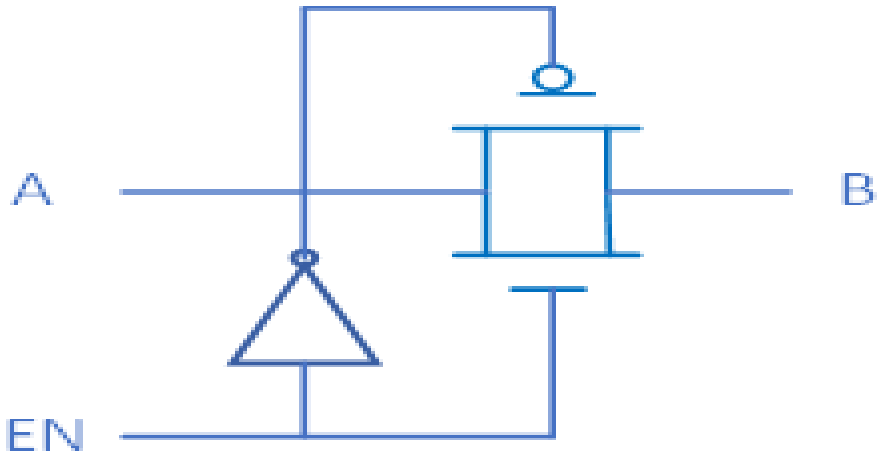
(c)



缓冲器

1.3 CMOS晶体管

- **传输门**（transmission gate）由一对CMOS 晶体管以及控制信号EN构成。
  - 信号EN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。
- 传输门的传播延迟非常短、电路简单，可双向传输。

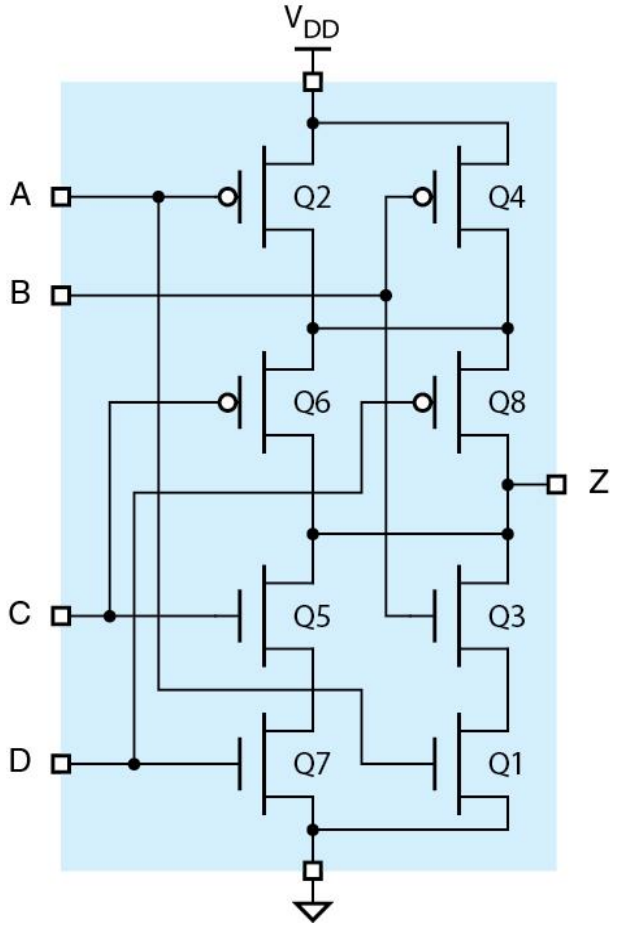


传输门

1.3 CMOS晶体管*

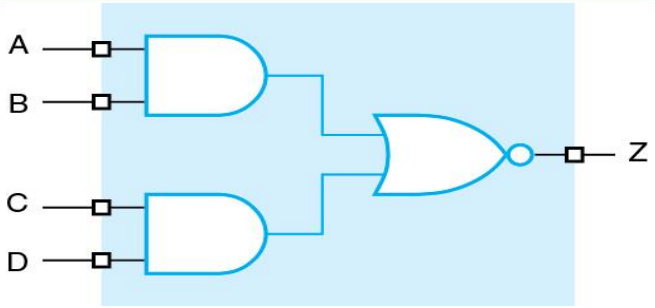
- 用单级晶体管实现两级逻辑与或非门。

(a)



(b)

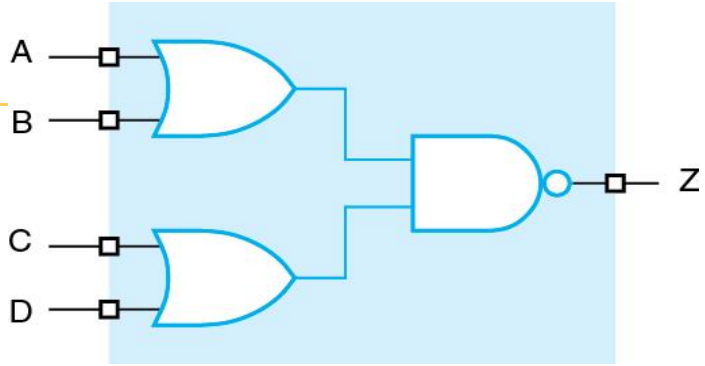
A	B	C	D	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Z
L	L	L	L	off	on	off	on	off	on	off	on	H
L	L	L	H	off	on	off	on	off	on	on	off	H
L	L	H	L	off	on	off	on	on	off	off	on	H
L	L	H	H	off	on	off	on	on	off	on	off	L
L	H	L	L	off	on	on	off	off	on	off	on	H
L	H	L	H	off	on	on	off	off	on	on	off	H
L	H	H	L	off	on	on	off	on	off	off	on	H
L	H	H	H	off	on	on	off	on	off	on	off	L
H	L	L	L	on	off	off	on	off	on	off	on	H
H	L	L	H	on	off	off	on	off	on	on	off	H
H	L	H	L	on	off	off	on	on	off	off	on	H
H	L	H	H	on	off	off	on	on	off	on	off	L
H	H	L	L	on	off	on	off	off	on	off	on	L
H	H	L	H	on	off	on	off	off	on	on	off	L
H	H	H	L	on	off	on	off	on	off	off	on	L
H	H	H	H	on	off	on	off	on	off	on	off	L



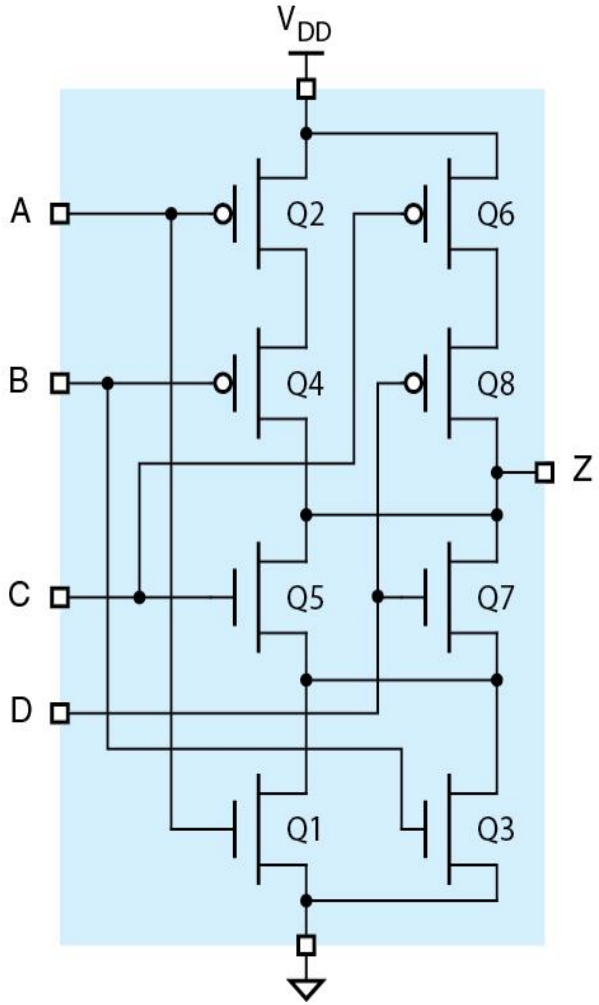
任何输入组合下，Z都不能同时与V_{DD}和地线相连。

1.3 CMOS晶体管*

- 用单级晶体管实现两级逻辑或与非门。



(a)

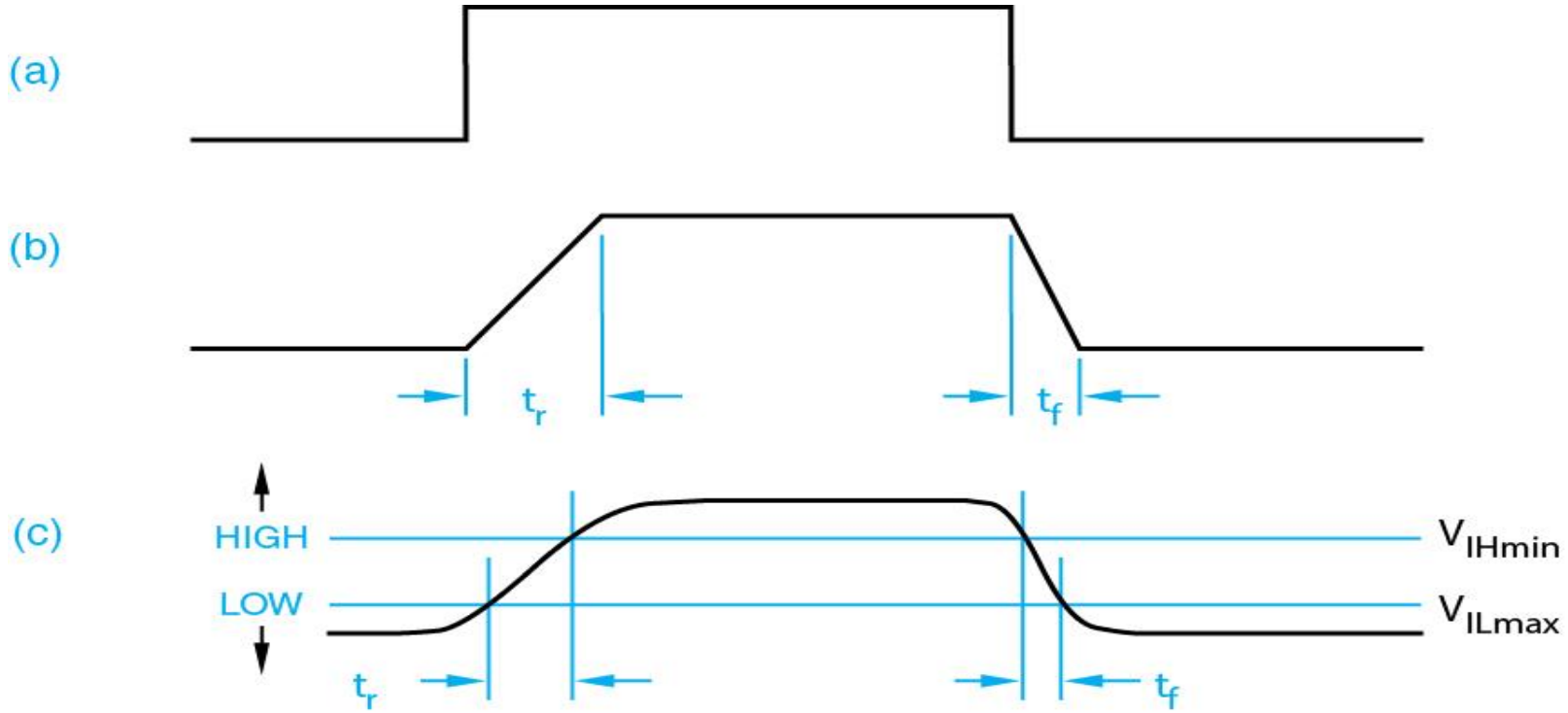


(b)

A	B	C	D	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Z
L	L	L	L	off	on	off	on	off	on	off	on	H
L	L	L	H	off	on	off	on	off	on	on	off	H
L	L	H	L	off	on	off	on	on	off	off	on	H
L	L	H	H	off	on	off	on	on	off	on	off	H
L	H	L	L	off	on	on	off	off	on	off	on	H
L	H	L	H	off	on	on	off	off	on	on	off	L
L	H	H	L	off	on	on	off	on	off	off	on	L
L	H	H	H	off	on	on	off	on	off	on	off	L
H	L	L	L	on	off	off	on	off	on	off	on	H
H	L	L	H	on	off	off	on	off	on	on	off	L
H	L	H	L	on	off	off	on	on	off	off	on	L
H	L	H	H	on	off	off	on	on	off	on	off	L
H	H	L	L	on	off	on	off	off	on	off	on	H
H	H	L	H	on	off	on	off	off	on	on	off	L
H	H	H	L	on	off	on	off	on	off	off	on	L
H	H	H	H	on	off	on	off	on	off	on	off	L

1.4 CMOS电路电气特性

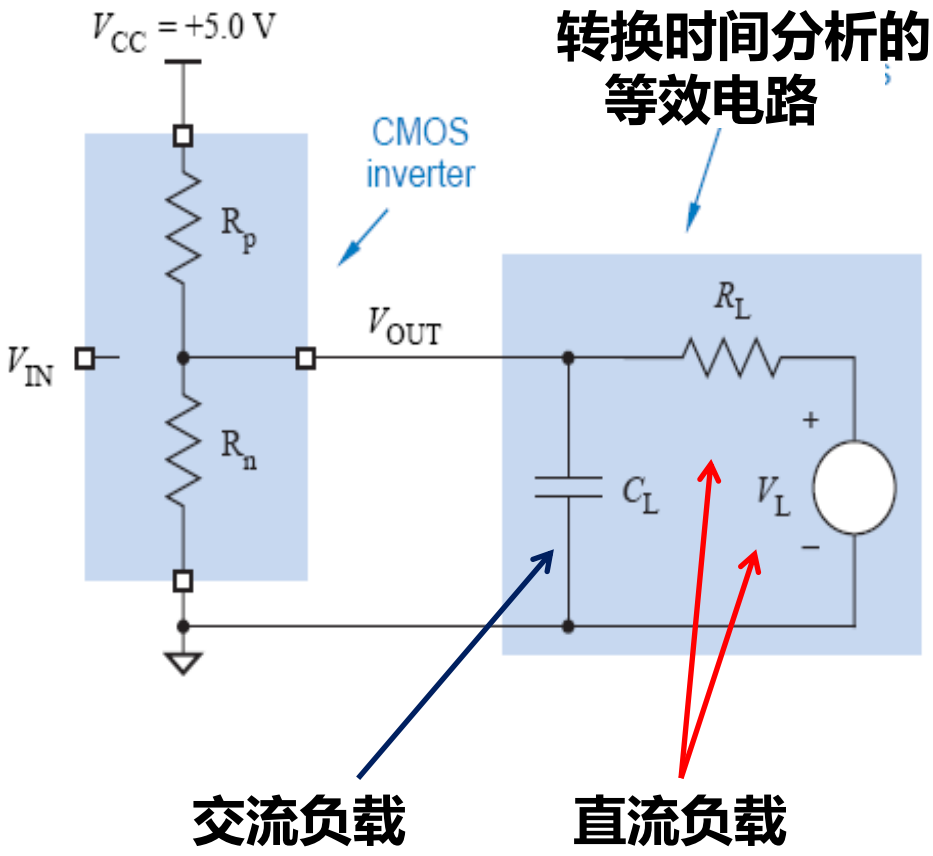
- 转换时间transition time： 逻辑电路的输出信号从一种状态转换到另一种状态所需的时间
  - 上升时间rise time t_r：从低态到高态。
  - 下降时间fall time t_f：从高态到低态。



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

- 影响转换时间的因素
  - 晶体管的导通电阻
  - 负载电容
- 寄生电容的来源
  - 输出电路，2-10pF
  - 输出和其它输入的连线电容，1pF/英寸或更多
  - 输入电路，2-15pF
- 称电容负载或交流负载

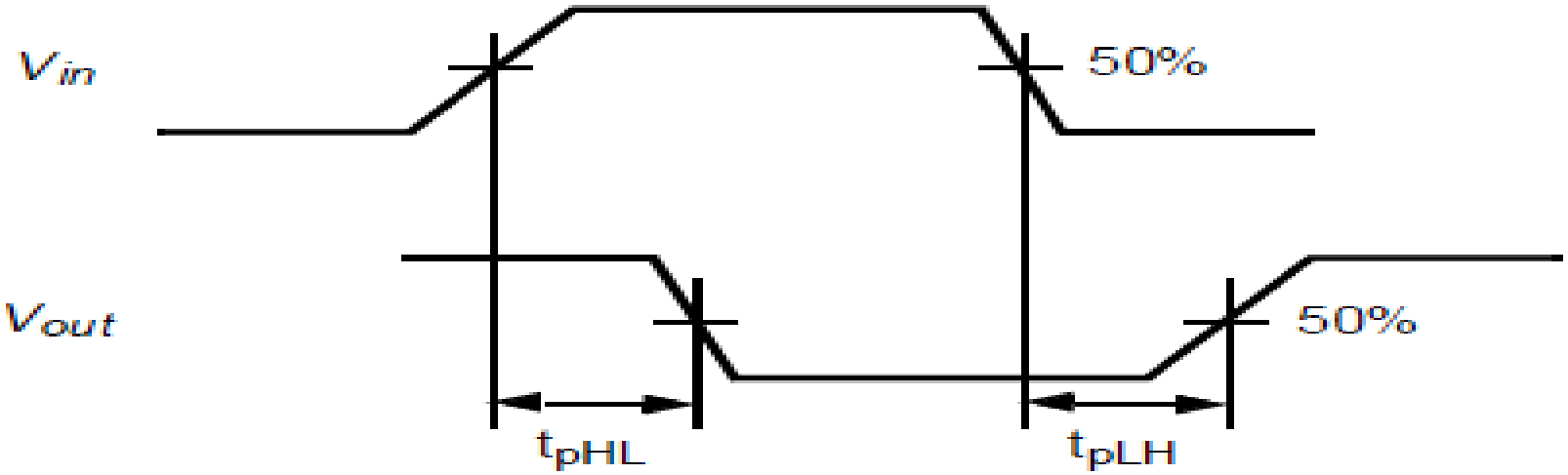


只驱动CMOS输入时，直流负载可忽略

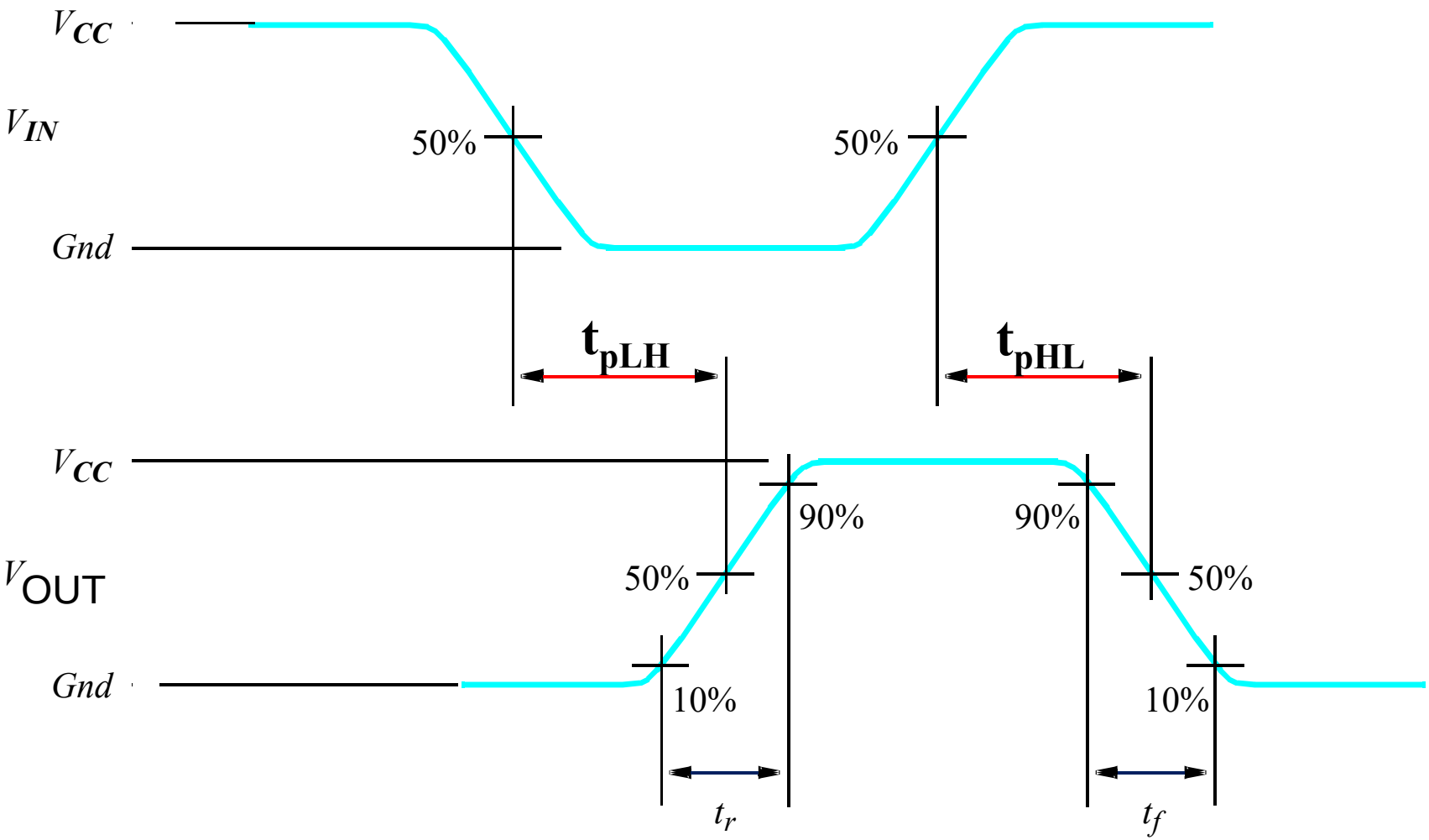
交流负载决定了输出状态转换时的电压和电流，以及从一个状态转换到另一个状态所需的时间。

1.4 CMOS电路电气特性

- **传播延迟**(t_p，propagation delay)是指从输入信号变化到产生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



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}
- 动态功耗：$P_D=(C_L+C_{PD})\times V_{CC}^2\times f$，f 为输出信号转换的频率
- 在CMOS电路的应用中， CV^2f功率是总功率的主要成分

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

- 从基本的两变量、三变量函数（逻辑门），过渡到多变量逻辑关系的刻画和实现

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$X_1, X_2, \cdots, X_n$是n个变量，每个变量取值0 或者取值1。

$F(X_1, X_2, \cdots, X_n)$ 是$X_1, X_2, \cdots, X_n$的一个逻辑函数，其取值由$X_1, X_2, \cdots, X_n$的取值决定。

$X_1,X_2,\cdots,X_n$	$F(X_1,X_2,\cdots,X_n)$
0 0 ⋯ 0	0
0 0 ⋯ 1	0
⋯	⋯
0 1 ⋯ 0	1
0 1 ⋯ 1	0
⋯	⋯
1 0 ⋯ 0	1
⋯	⋯
1 1 ⋯ 1	0

3.1 逻辑函数

- 每一组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组合
- 两个输入变量的函数F(x,y)可能的输出值如下：

X	Y	$F_i(X, Y)$															
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

三个输入变量有多少种可能的逻辑函数f(x,y,z)?

3.2 真值表与波形图

- 刻画逻辑关系的方法

- **真值表 true table：用二维表的形式列出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。**

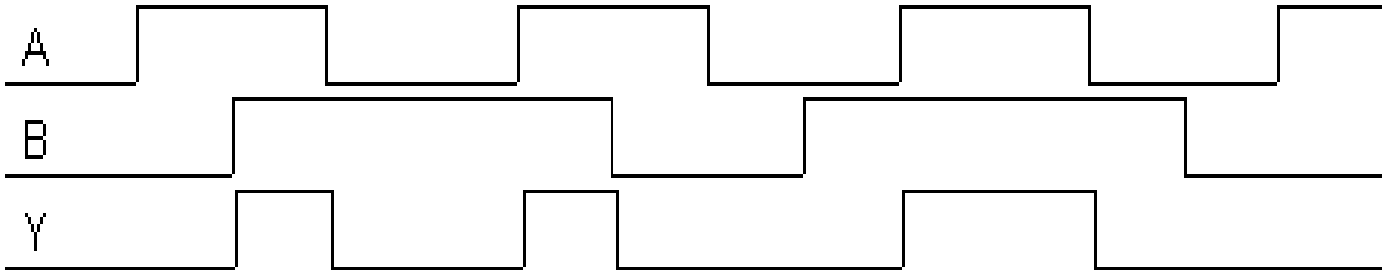
真值表

ABC	Y
000	0
001	0
010	0
011	1
100	0
101	1
110	1
111	1

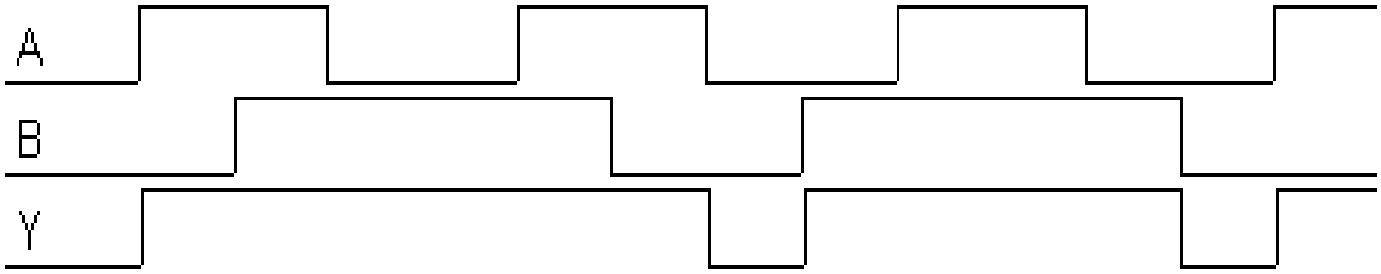
$$Y = A \cdot B + B \cdot C + A \cdot C$$

3.2 真值表与波形图

- **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



$Y=A \cdot B$



$Y=A+B$

3.3 逻辑函数的标准表示

- **描述逻辑关系的方法：**
  - **自然语言描述**
  - **逻辑函数**
    - 简洁，需要标准化
  - **真值表**
    - 精准
  - **波形图**
    - 直观
  - **其它…**

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

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：$X\cdot Y+\bar{X}\cdot\bar{Y}\cdot Z$。
- **“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：$(X+Y)\cdot(\bar{X}+\bar{Y}+Z)$。

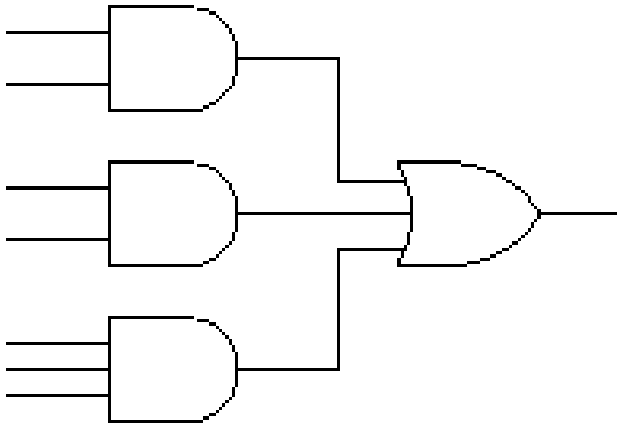
3.3 逻辑函数的标准表示

- **与-或表达式**：积之和表达式SOP

$$f(A,B,C) = A \bullet B + \overline{A} \bullet C + A \bullet \overline{B} \bullet \overline{C}$$

与项

乘积项

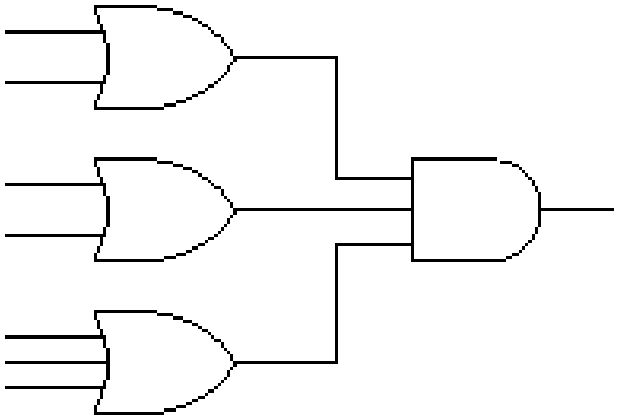


- **或-与表达式**：和之积表达式POS

$$f(A,B,C) = (A + B) \bullet (A + \overline{C}) \bullet (\overline{A} + B + C)$$

求和项

或项



3.3 逻辑函数的标准表示

- **标准乘积项**：每个逻辑变量出现且仅出现一次的**乘积项**。
  - n 个变量的标准项共有2^n 个。
  - 标准乘积项也称为**最小项**，每个最小项都对应真值表中的一个输入组合，赋值该输入组合后，最小项的运算结果为**1**，其它都为0。
  - 将该输入组合对应的0/1序列看成序号i，则可用m_i 表示该最小项，i 称为该最小项的编号。
    - 如：$\overline{A} \bullet \overline{B} \bullet C$只有当输入001时，结果为1，最小项编号为$m_1$
- **标准求和项**：每个逻辑变量出现且仅出现一次的**求和项**。
  - n 个变量的标准项共有2^n 个。
  - 标准求和项也称为**最大项**，每个最大项都对应真值表中的一个输入组合，赋值该输入组合后，最大项的运算结果为**0** ，其它都为1 。
  - 将该输入组合对应的0/1序列编码看成序号i，则可用M_i 表示该最大项，i 称为该最大项的编号。
    - 如：$\overline{A} + \overline{B} + C$ 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	ABC	Y	最小项	最大项
0	000	1	$\overline{A}\bullet\overline{B}\bullet\overline{C}$	$A+B+C$
1	001	0	$\overline{A}\bullet\overline{B}\bullet C$	$A+B+\overline{C}$
2	010	0	$\overline{A}\bullet B\bullet\overline{C}$	$A+\overline{B}+C$
3	011	0	$\overline{A}\bullet B\bullet C$	$A+\overline{B}+\overline{C}$
4	100	0	$A\bullet\overline{B}\bullet\overline{C}$	$\overline{A}+B+C$
5	101	0	$A\bullet\overline{B}\bullet C$	$\overline{A}+B+\overline{C}$
6	110	1	$A\bullet B\bullet\overline{C}$	$\overline{A}+\overline{B}+C$
7	111	1	$A\bullet B\bullet C$	$\overline{A}+\overline{B}+\overline{C}$

标准和/最小项列表：使得函数输出为1的最小项之和。

$$Y=\overline{A}\bullet\overline{B}\bullet\overline{C}+A\bullet B\bullet\overline{C}+A\bullet B\bullet C$$
$$=\sum m\,(0,6,7)$$

标准积/最大项列表：使得函数输出为0的最大项之积。

$$Y=(A+B+\overline{C})\bullet(A+\overline{B}+C)$$
$$\bullet(A+\overline{B}+\overline{C})\bullet(\overline{A}+B+C)$$
$$\bullet(\overline{A}+B+\overline{C})=\prod M(1,2,3,4,5)$$

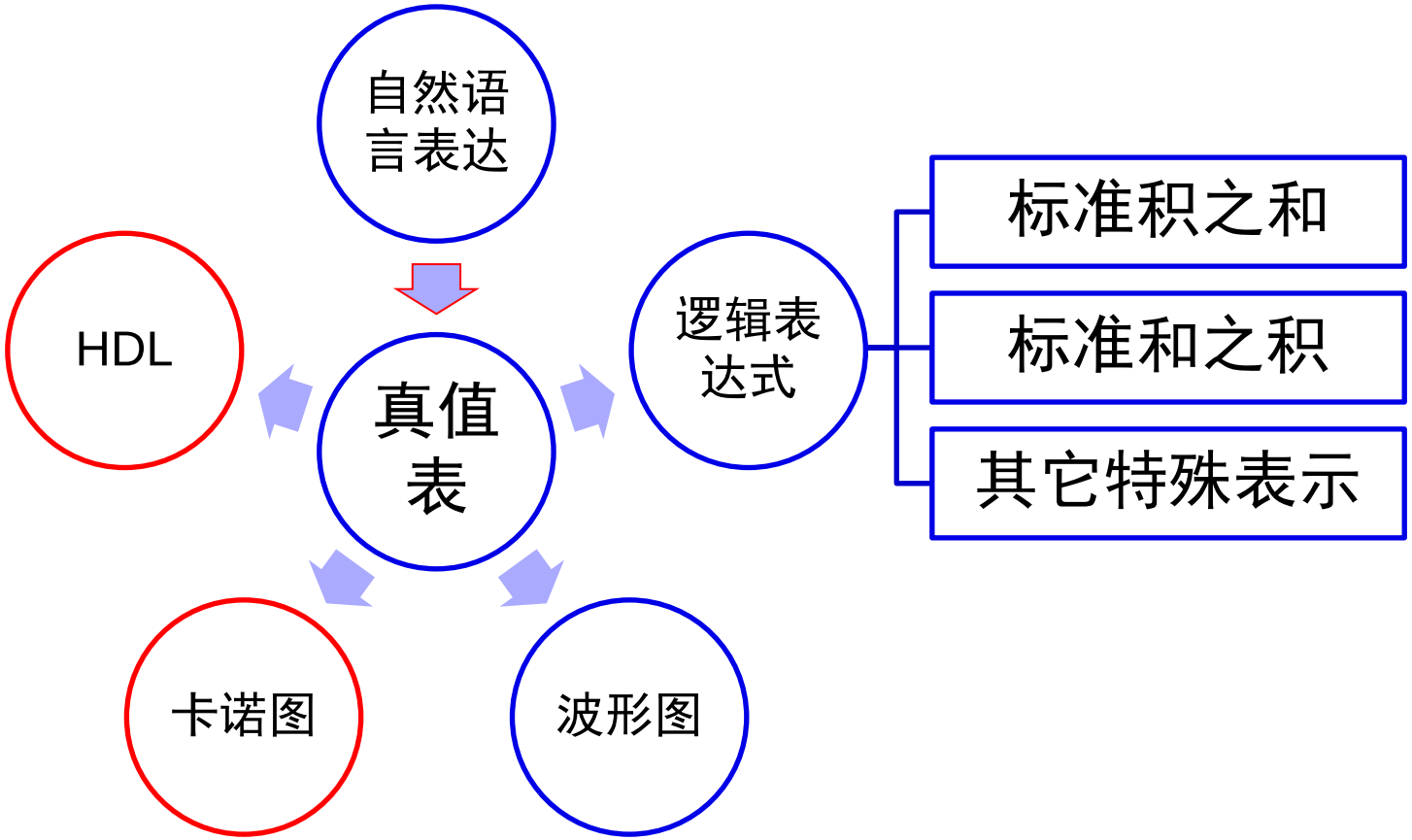
真值表的表示是唯一的，因此标准项列表的表示也是唯一的。

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 个变量的逻辑函数其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，…，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。
  - $f(A,B,C)=\Sigma\, m\,(0,1,2,3) = \Pi\, M(4,5,6,7)$
  - $f(X,Y) = \Sigma\, m\,(1) = \Pi\, M(0,2,3)$
  - $f(A,B,C) = \Sigma\, m\,(0,6,7) = \Pi\, M(1,2,3,4,5)$

3.3 逻辑函数的标准表示

- **逻辑关系的表现形式**



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

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

4 逻辑函数的化简与变换

- 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
  - 门越小速度越快；可以减少电路层级数。

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。

- 数字系统辅助设计工具EDA中都包含化简功能模块
  - FPGA、PLD的资源毕竟是有限的
  - ASIC门电路的输入端的数量也是有限的

4.1 代数法化简

- 利用布尔代数的公理、定理、定律等，消去逻辑表达式式中多余的乘积项或多余的因子，进行化简。
  - 利用互补律T5： $\overline{X}+X=1$，可消去一个变量。
  - 利用吸收律T9：X+X•Y=X 和 $X+\overline{X}\bullet Y=X+Y$，可消去乘积项中一个因子。
  - 利用组合律T10：X•Y+X • $\overline{Y}$ =X 和 (X+Y)•(X+ $\overline{Y}$)=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。

4.1 代数法化简

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

  =w•x+ $\bar{w}$·(y•z+y•z+x•y•z)　　(根据T8和T9)

  =w•x+ $\bar{w}$·(z+x•y•z)　　(根据T10)

  =w•x+ $\bar{w}$·(z+x•y)　　(根据T9)

  =w•x+ $\bar{w}$·z+ $\bar{w}$·x•y　　(根据T9)

  =(w+ $\bar{w}$·y)•x+ $\bar{w}$·z　　(根据T8)

  =(w+y)•x+ $\bar{w}$·z　　(根据T9)

  =w•x+x•y+ $\bar{w}$·z　　(根据T8)

- 和原表达式相比，化简后减少了2个与门、11个输入端。
  - 门数：　5与门+1或门　→ 3与门+1或门
  - 输入端：　15一级+5二级　→ 6一级+3二级

4.1 代数法化简

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

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

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

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

$$\mathbf{X\cdot Y+\bar{X}\cdot Z+Y\cdot Z=X\cdot Y+\bar{X}\cdot Z}$$

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

是最简表达式吗？

4.1 代数法化简

- 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

如何证明是最简表达式？

- 判别逻辑表达式是否为最小化
  - 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）相邻的小方格输入逻辑相邻

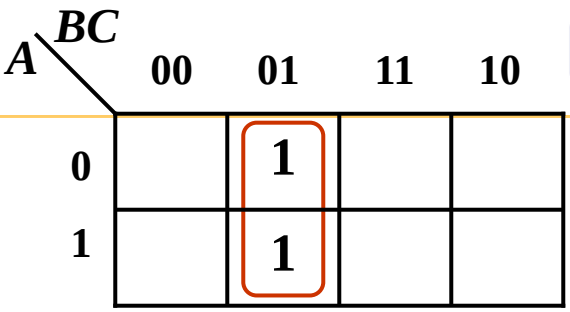
A \ B	0	1
0	$\overline{A}\overline{B}$	$\overline{A}B$
1	$A\overline{B}$	AB

A \ B	0	1
0	m_0	m_1
1	m_2	m_3

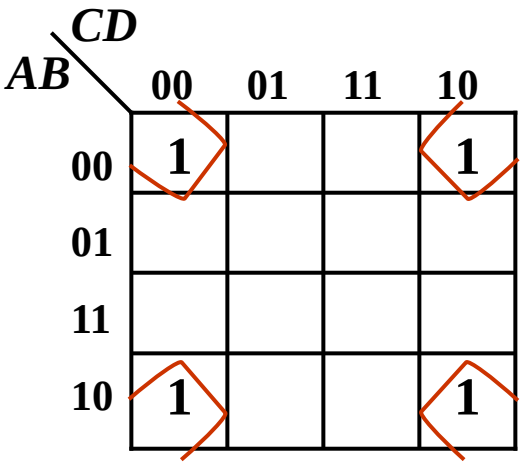
AB \ CD	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

4.2 卡诺图化简

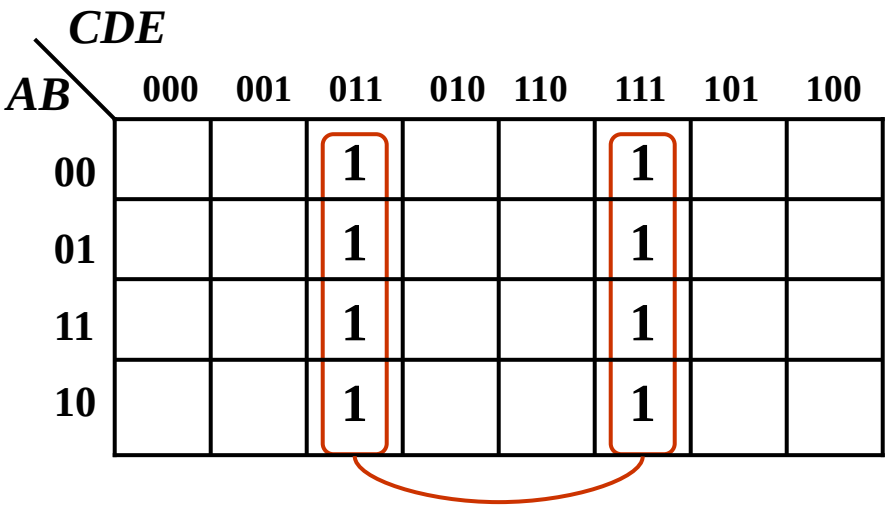
- 卡诺图每个单元对应一个最小项，其中标注该最小项在真值表中的输出值，如果输出为1，则称为“1单元”
- 若两个“1单元”相邻，则表示两个最小项仅1个变量不同，该变量在一个方格中为原变量，在另一个方格中则为反变量。根据T10，这两个最小项可合并为一个乘积项，并消去那个不相同的变量
- 相邻单元数越多可消去的变量数越多
- 相邻2^i个“1单元”的最小项可以合并成一个乘积项，并消去i个不同的变量
- 使用一个方框来标注可以合并的“1单元”，这个方框称为卡诺圈



$\overline{B} \cdot C$



$\overline{B} \cdot \overline{D}$



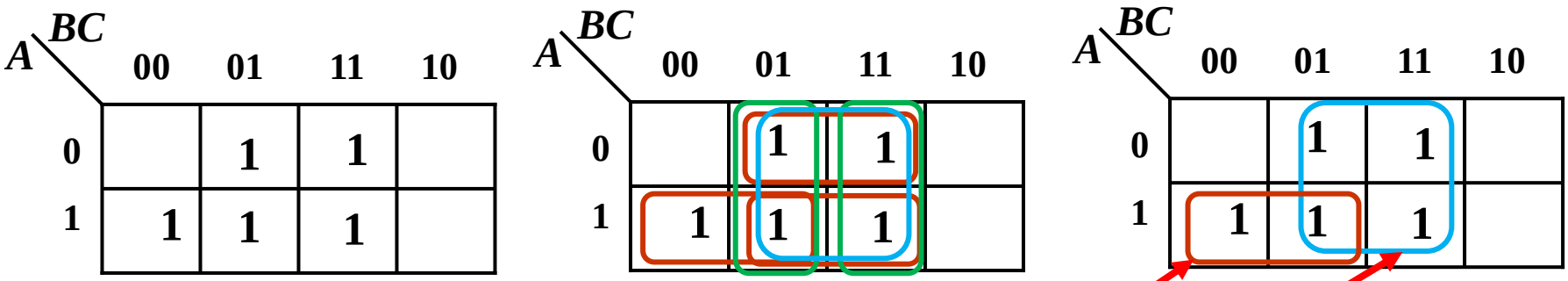
$D \cdot E$

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。
- 质蕴涵项(prime implicant)：没有被该逻辑函数的其它蕴涵项所覆盖的蕴涵项。
- 实质蕴涵项(essential prime implicant)：覆盖的最小项中至少有一个最小项没有被其他质蕴涵项所覆盖的质蕴涵项。
- 质蕴涵项覆盖的最小项越多越可能是实质蕴涵项。
- 如果逻辑函数的所有最小项都被一组的质蕴涵项所覆盖，则该组蕴涵项被称为函数的一个覆盖(Cover)，一定包含了所有的实质蕴涵项。
- 最小覆盖是包含质蕴涵项数是最少的，并且质蕴涵项中的变量总数也是最少的。
- 逻辑函数化简问题就转化为寻找该函数的最小覆盖问题。

4.2 卡诺图化简

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



蕴涵项：

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

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

含有四个最小项的蕴涵项：$\{C\}$

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

实质蕴涵项 = $\{A\cdot\overline{B},\ C\}$

最小覆盖 = $A\cdot\overline{B}+C$

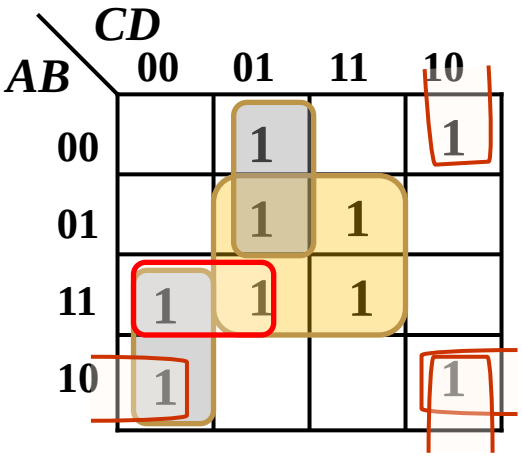
4.2 卡诺图化简

- 逻辑函数卡诺图化简的关键：找出和选择质蕴涵项，一般步骤如下：

1. 根据逻辑表达式，列出真值表，构建卡诺图
2. 找出卡诺图中的所有质蕴涵项，从覆盖范围最大的质蕴涵项开始
3. 找出所有的实质蕴涵项
4. 在剩余的质蕴涵项中寻找一个最小的覆盖，该覆盖包含那些没有被实质蕴涵项所覆盖的最小项
5. 合并3、4两步的结果，生成函数的最简逻辑表达式

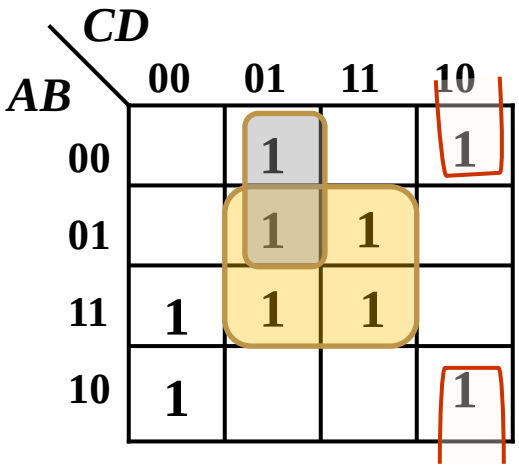
4.2 卡诺图化简

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



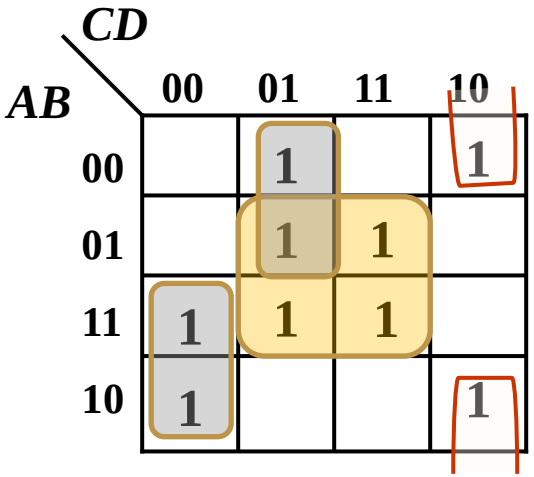
找出所有质蕴涵项

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



选择所有的实质蕴涵项

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



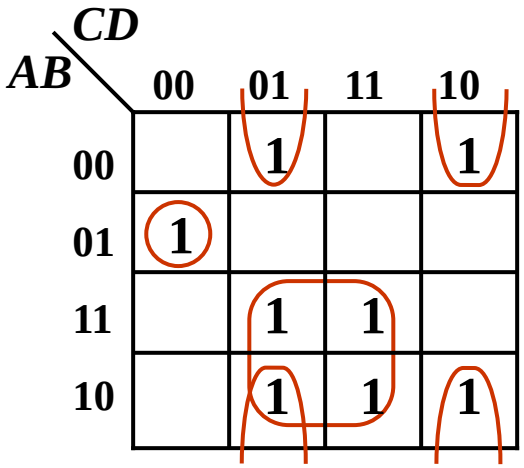
选择最小的覆盖子集

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

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

4.2 卡诺图化简

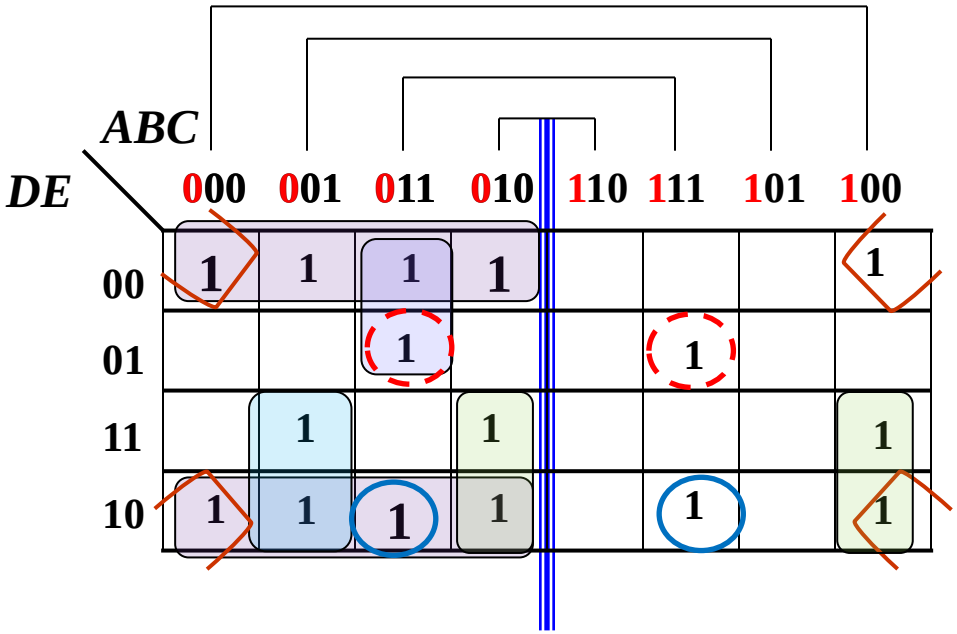
$$Y(A,B,C,D)=\sum m(1,\quad 2,\quad 4,\quad 9,\quad 10,\quad 11,\quad 13,\quad 15)$$



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

4.2 卡诺图化简

$$F(A,B,C,D,E) = \sum m(0,2,4,6,7,8,10,11,12,13,14,16,18,19,29,30)$$



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

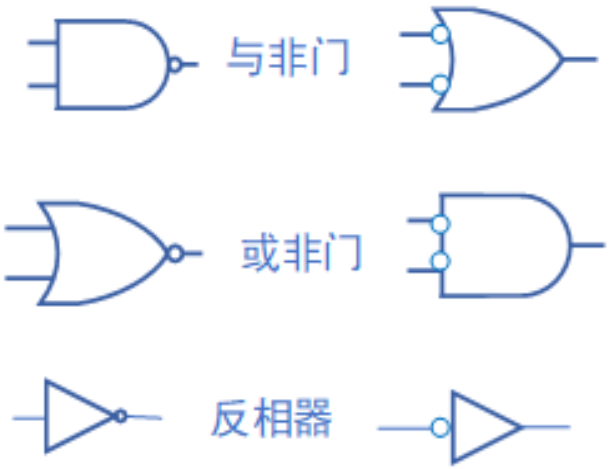
4.2 卡诺图化简

- 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。

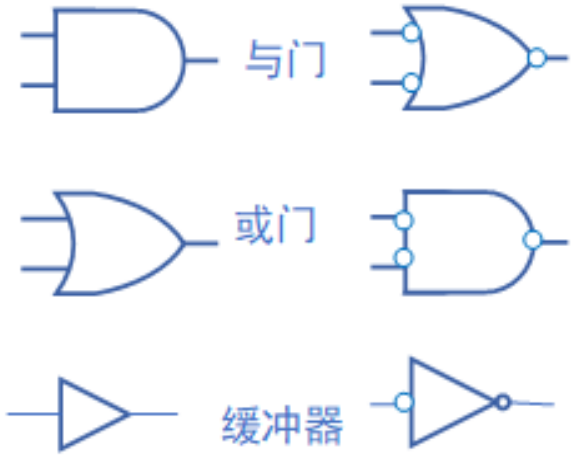
在现代数字系统设计中，大多采用计算机程序自动实现逻辑函数的化简。

4.3 逻辑函数变换

- **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。



(a)反相门的等效符号

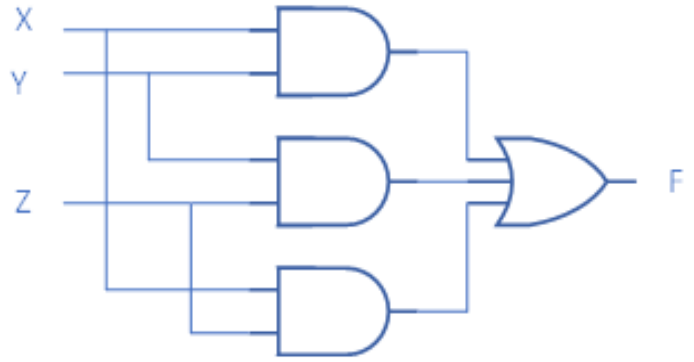


(b)非反相门的等效符号

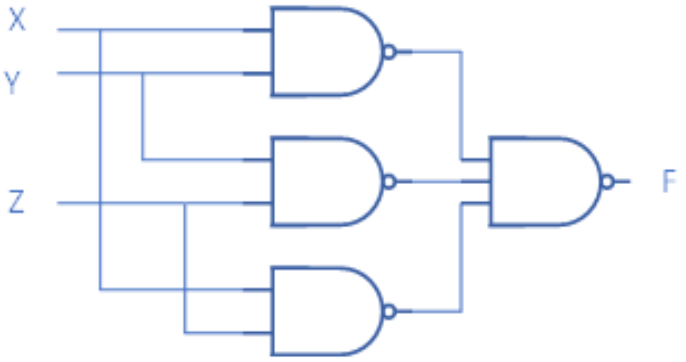
4.3 逻辑函数变换

- 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该逻辑函数。
  - 例如：F(X,Y,Z)=X•Y+Y•Z+X•Z，转换为：

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



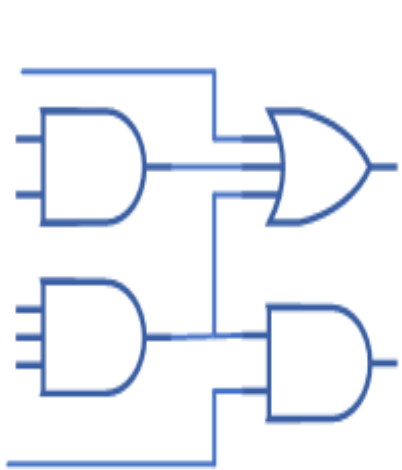
(a)与-或电路



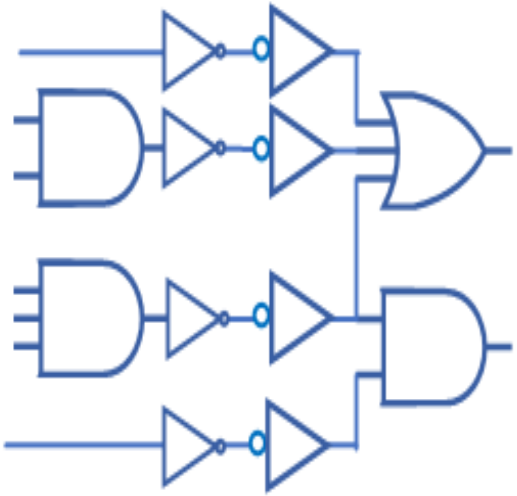
(b)与非-与非电路

4.3 逻辑函数变换

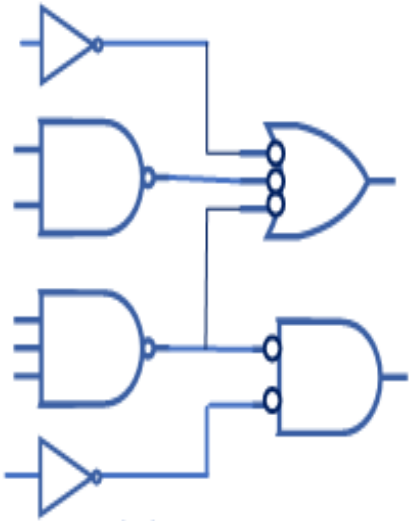
- 对于任何积之和表达式都可使用“与-或”电路和“与非-与非”电路这两种方法实现。
- 在第一级的输出和第二级的输入之间加入**一对反相器**，来实现用反相门替代与门和或门。



(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

4.3 逻辑函数变换*

- 利用布尔代数可以实现不同形式逻辑表达式之间的变换。
- 公式之间可等价变换，逻辑电路图的不同设计，实现也不同
- 示例：与或表达式转换成或与表达式

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

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

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

$$F = 1 \bullet 1 \bullet (X + \overline{Y} + \overline{Z}) \bullet (Z + \overline{X}) \bullet (Z + Y) \bullet 1$$

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