

本节内容

补充：数字电 路基础知识

逻辑门电路

本节总览

处理二进制的
逻辑运算

逻辑门电路

基本逻辑运算 —— 与、或、非

复合逻辑运算 —— 与非、或非、异或、同或

其他补充

Tips: 本科期间学过《数字电路》的同学可以跳过本视频

算数运算 V.S. 逻辑运算

算数运算——对我们熟悉的数字进行运算，如：加、减、乘、除、幂次方

基本算
数运算

复合算
数运算

逻辑运算——对逻辑值 真/假（二进制 1/0）进行运算，如：与、或、非、异或

基本逻辑
运算

复合逻辑
运算

注1：无论是算数运算、还是逻辑运算，都要有输入、输出。

注2：《离散数学》是专门研究逻辑运算的一门数学学科。

基本逻辑运算：与

逻辑运算

与 AND

表达式

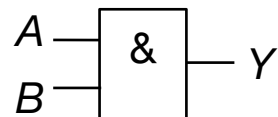
$$Y = A \cdot B$$

也可简写为
 $Y = AB$

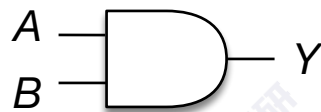
真值表

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

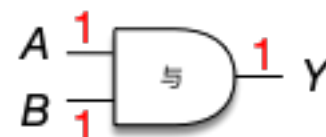
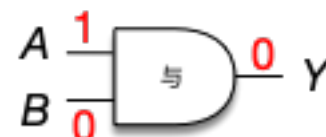
门电路图形符号



国标画法
GB/T 4728.02-1996



国际常用画法



“与门”
示例

基本逻辑运算：或

逻辑运算

或 OR

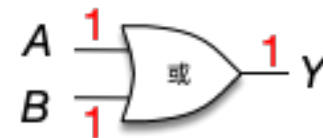
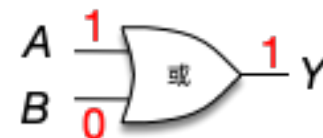
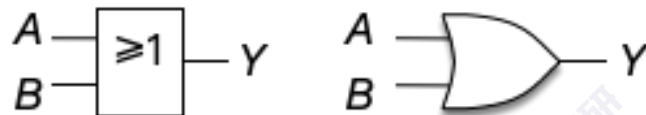
表达式

$$Y = A + B$$

真值表

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

门电路图形符号



“或门”
示例

基本逻辑运算：非

逻辑运算

非 NOT

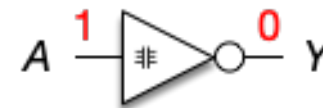
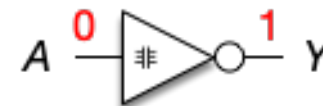
表达式

$$Y = \bar{A}$$

真值表

A	Y
0	1
1	0

门电路图形符号



“非门”
示例

复合逻辑运算：与非

逻辑运算

与非 NAND

NOT-AND

表达式

$$Y = \overline{A \cdot B}$$

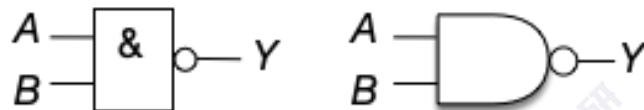
也可简写为
 $Y = \overline{AB}$

$$Y = \overline{A \cdot B}$$

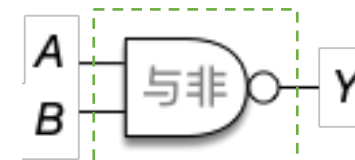
真值表

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

门电路图形符号



简化



复合逻辑运算：或非

逻辑运算

或非 NOR

NOT-OR

表达式

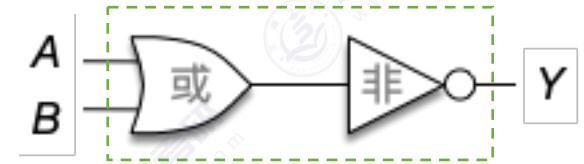
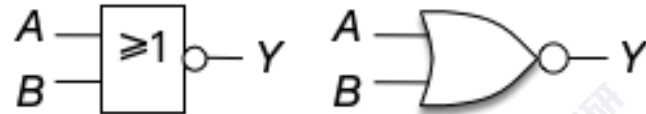
$$Y = \overline{A + B}$$

$$Y = \overline{A + B}$$

真值表

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

门电路图形符号



简化



复合逻辑运算：异或

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

逻辑运算

异或 XOR

exclusive OR

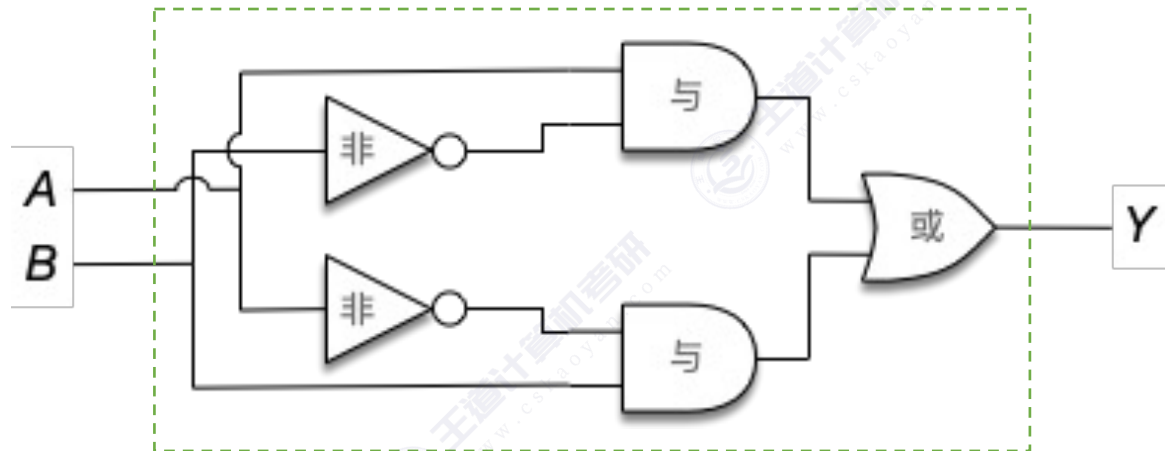
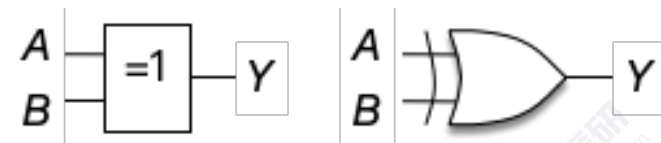
表达式

$$Y = A \oplus B$$

真值表

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

门电路图形符号



简化



异或门：两个输入相“异”时，输出为1

“封装”思想：屏蔽电路部件的内部实现细节，仅对外暴露出输入/输出引脚。使用者仅需关心该部件的功能即可。

Tips: 对比程序的“封装”思想→函数

复合逻辑运算：同或

逻辑运算

同或 XNOR

exclusive NOT-OR

表达式

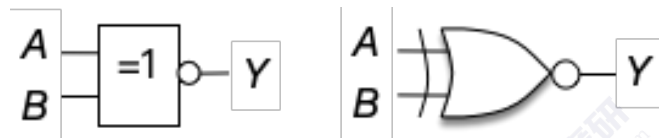
$$Y = A \odot B$$

效果等价于
 $Y = \overline{A \oplus B}$

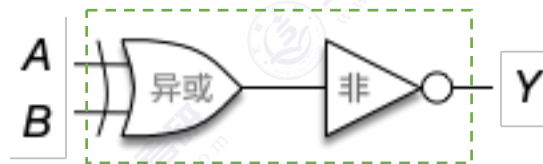
真值表

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

门电路图形符号



$$Y = A \odot B = \overline{A \oplus B}$$





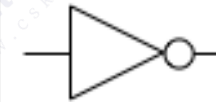
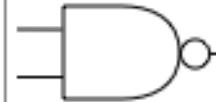



简化



注：有的地方会翻译为“**异或非门**”

同或门：两个输入相“同”时，输出为1

逻辑门电路基础总结

逻辑运算	与	或	非	与非	或非	异或	同或																																																																																																
逻辑表达式	$Y=A \cdot B$	$Y=A+B$	$Y=\overline{A}$	$Y=\overline{A \cdot B}$	$Y=\overline{A+B}$	$Y=A \oplus B$	$Y=A \odot B$																																																																																																
英文缩写	AND	OR	NOT	NAND	NOR	XOR	XNOR																																																																																																
门电路																																																																																																							
真值表	<table><tr><td>A</td><td>B</td><td>Y</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	Y	0	0	0	0	1	0	1	0	0	1	1	1	<table><tr><td>A</td><td>B</td><td>Y</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	Y	0	0	0	0	1	1	1	0	1	1	1	1	<table><tr><td>A</td><td>Y</td></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	A	Y	0	1	1	0	<table><tr><td>A</td><td>B</td><td>Y</td></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	Y	0	0	1	0	1	1	1	0	1	1	1	0	<table><tr><td>A</td><td>B</td><td>Y</td></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	Y	0	0	1	0	1	0	1	0	0	1	1	0	<table><tr><td>A</td><td>B</td><td>Y</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	Y	0	0	0	0	1	1	1	0	1	1	1	0	<table><tr><td>A</td><td>B</td><td>Y</td></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	Y	0	0	1	0	1	0	1	0	0	1	1	1
A	B	Y																																																																																																					
0	0	0																																																																																																					
0	1	0																																																																																																					
1	0	0																																																																																																					
1	1	1																																																																																																					
A	B	Y																																																																																																					
0	0	0																																																																																																					
0	1	1																																																																																																					
1	0	1																																																																																																					
1	1	1																																																																																																					
A	Y																																																																																																						
0	1																																																																																																						
1	0																																																																																																						
A	B	Y																																																																																																					
0	0	1																																																																																																					
0	1	1																																																																																																					
1	0	1																																																																																																					
1	1	0																																																																																																					
A	B	Y																																																																																																					
0	0	1																																																																																																					
0	1	0																																																																																																					
1	0	0																																																																																																					
1	1	0																																																																																																					
A	B	Y																																																																																																					
0	0	0																																																																																																					
0	1	1																																																																																																					
1	0	1																																																																																																					
1	1	0																																																																																																					
A	B	Y																																																																																																					
0	0	1																																																																																																					
0	1	0																																																																																																					
1	0	0																																																																																																					
1	1	1																																																																																																					
门电路符号特征	门电路输入端是直线	门电路输入端为曲线	小三角+小圆圈	与门的输出端加个小圆圈	或门的输出端加个小圆圈	或门的输入端加条曲线	异或门的输出端加小圆圈																																																																																																
门电路逻辑运算特征	当且仅当输入全1时，输出才为1	当且仅当输入全0时，输出才为0	0变1，1变0	仅当输入全1时，输出才为0	当输入全0时，输出才为1	输入相异时输出为1	输入相同时输出为1																																																																																																

题外话：异或运算的妙用

异或运算的妙用：n bit 进行异或，若有奇数个 1，则异或结果为 1；若有偶数个 1，则异或结果为 0

$$1 \oplus 0 \oplus 0 \oplus 1 \oplus 1 \oplus 0 \oplus 1 = 0$$

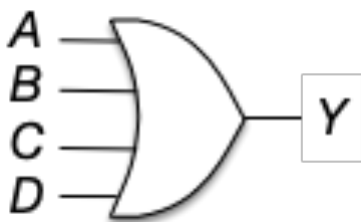
$$1 \oplus 0 \oplus 1 \oplus 0 \oplus 1 \oplus 1 \oplus 1 = 1$$

可用于实现奇偶校验、二进制加法

妙啊

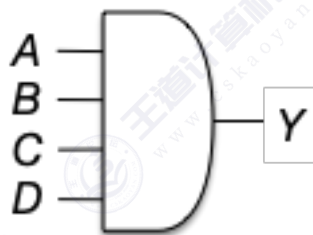


补充1: 门电路的变形画法



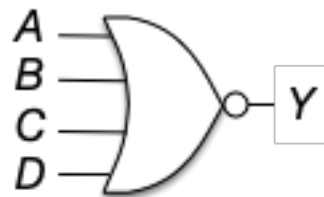
多输入“或门”

$Y=A+B+C+D$, 当且仅当所有输入都为0时, 输出才为0



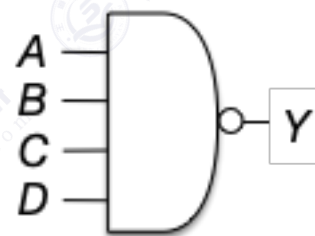
多输入“与门”

$Y=A \cdot B \cdot C \cdot D$, 当且仅当所有输入都为1时, 输出才为1



多输入“或非门”

$Y=\overline{A+B+C+D}$, 当且仅当所有输入都为0时, 输出才为1



多输入“与非门”

$Y=\overline{A \cdot B \cdot C \cdot D}$ 当且仅当所有输入都为1时, 输出才为0

补充2：逻辑运算的优先级、常见公式

三种基本逻辑运算的优先级：非 > 与 > 或

注1：与、或可以类比乘法、加法

注2：如果有括号，括号的优先级更高

注3：非运算符下面，可理解为“隐含一个括号”

Eg 1: $Y=AB+CD$ → 先与后或

Eg 2: $Y= A(B+C)D$ → 先算括号内的或、再算两个与

Eg 3: $Y= \overline{A}B+C$ → 先非、再与、最后或

Eg 4: $Y=\overline{AB}+C$ → 先与、再非、最后或。等价于 $\overline{(AB)} + C$

逻辑运算的常见公式（详见离散数学）：

$A(C+D) = AC+ AD$ ——分配律

$ABC=A(BC)$ ——与运算的结合律

$A+B+C=A+(B+C)$ ——或运算的结合律

$\overline{A+B} = \overline{A} \cdot \overline{B}$ ——反演律

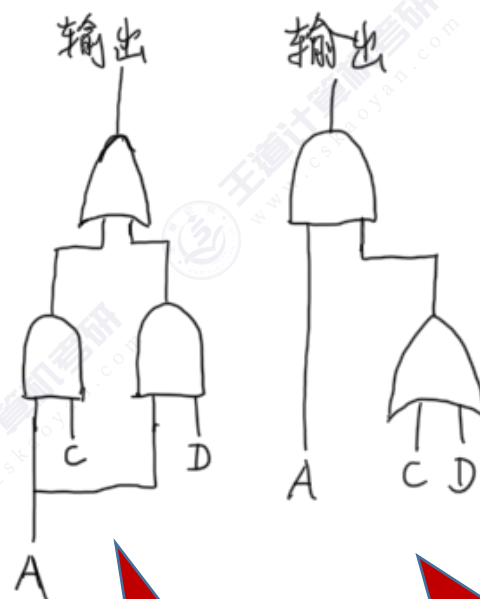
$\overline{A \cdot B} = \overline{A} + \overline{B}$ ——反演律

有何意义？

数学使人卸下伪装



Eg: 用电路实现复合逻辑运算 $AC+AD$

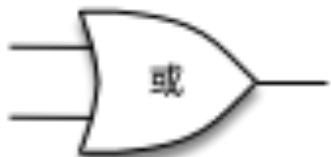


Tips: 本质上
逻辑表达式是
对电路的数学
化描述，简化
逻辑表达式，
就是在简化电
路设计，就是
在省钱。

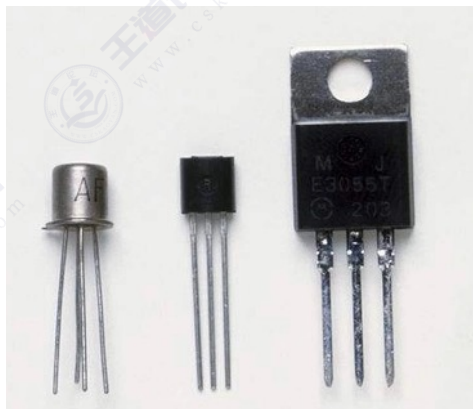
$AC+AD$

$AC+AD = A(C+D)$

拓展：芯片制程



门电路：使用门电路设计电路，可实现二进制运算



门电路由“晶体管”实现

关键制造设备：光刻机



某芯片采用 **10nm制程**——意味着该芯片内部的**晶体管栅极宽度为10nm**。晶体管越小，功耗越低、单位面积内晶体管数量更多（可实现更复杂的硬件功能）

注：头发丝直径约为 $0.1\text{mm} = 100,000\text{nm}$



公众号：王道在线



b站：王道计算机教育



抖音：王道计算机考研