



## 课程性质与教学目标

**课程性质：** 深入了解计算机“内核” 的一门最关键的基础课程。

**教学目标：** 要求掌握对数字系统硬件进行分析、设计和开发的基本技能。

**教学时数：** 64学时  
课堂教学：48      实验：16

**考核：** (实验)  $\times 20\%$  + (作业+课堂+期中)  $20\%$  + 期末  $\times 60\%$

# 教学内容：

课程任务

原理、分析

典型单元设计

硬件编程语言（VHDL）

## § 1.1 数字系统

### 一、数字系统：

处理数字信号的实体。

电子电路中的信号

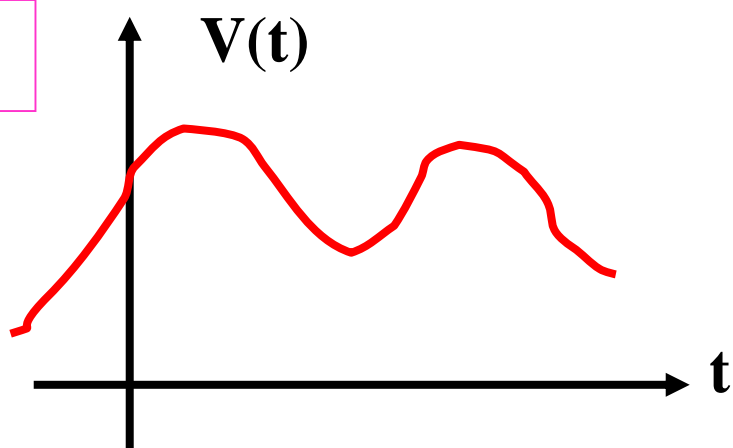
模拟信号

幅度随时间连续变化的信号

数字信号

幅度随时间离散变化的二值信号

模拟信号



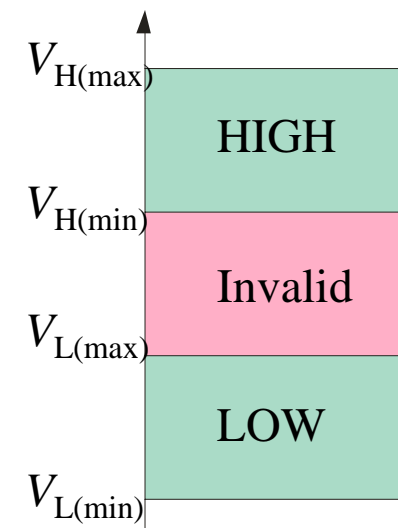
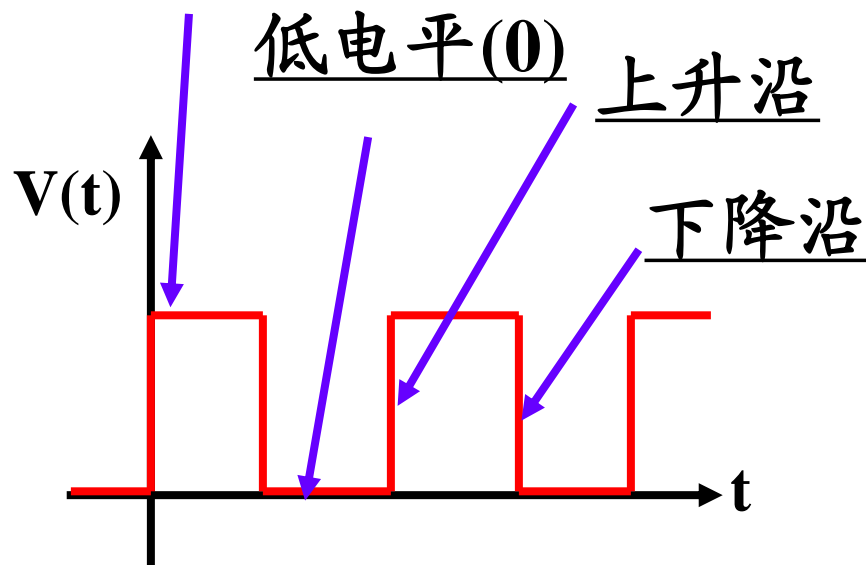
数字信号

高电平(1)

低电平(0)

上升沿

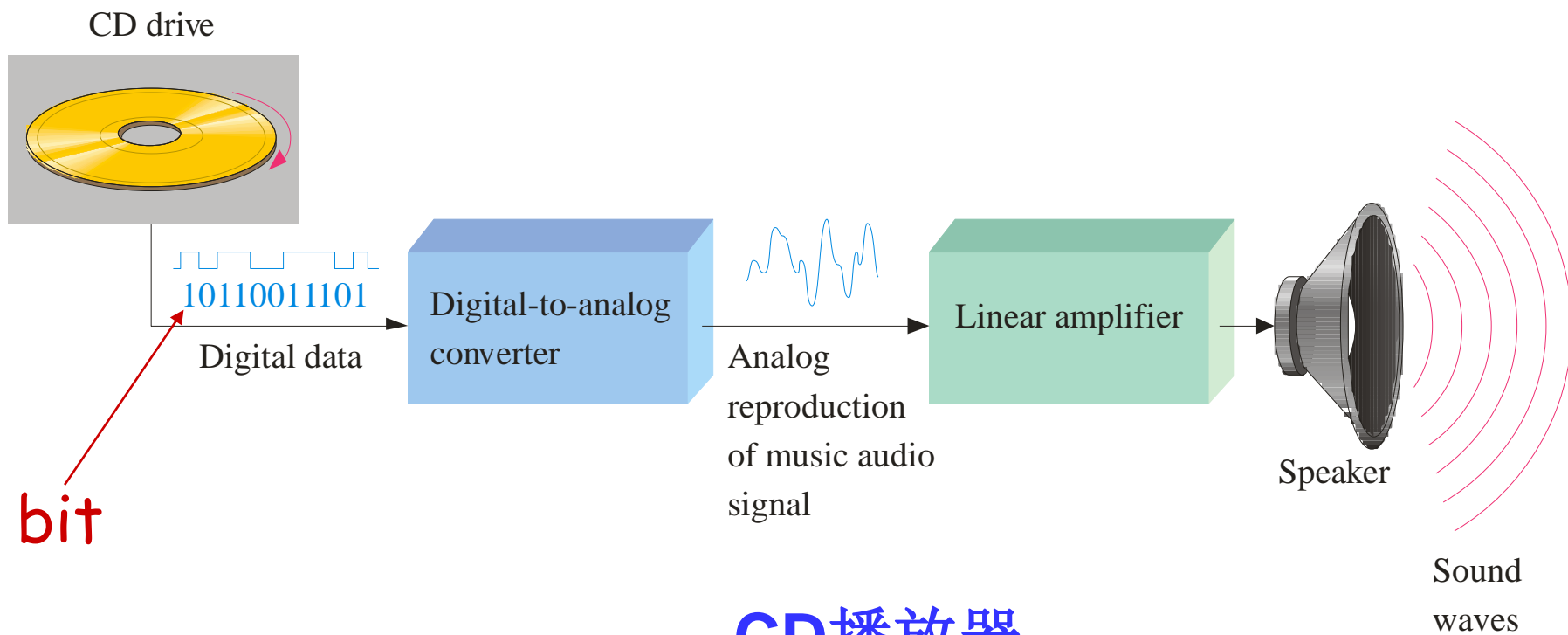
下降沿



负逻辑：高电平为“0”，低电平为“1”。

## 数字电路的特点:

- 1) 二值信号，抗干扰能力强，保密性好；
- 2) 通用性强，具有极强的信息处理和控制能力；



**CD播放器**

## 二、模拟电路与数字电路的区别

### 1、工作任务不同：

模拟电路：大小、相位、失真等；

数字电路：逻辑关系（因果关系）。

### 2、晶体管的工作状态不同：

模拟电路：工作在线性放大区,是放大管；

数字电路：工作在饱和或截止状态，是开关管。

数字系统与模拟信号联系时，必须经过模/数(A/D)、数/模(D/A) 电路。

### 3、数字电路含有两种运算:

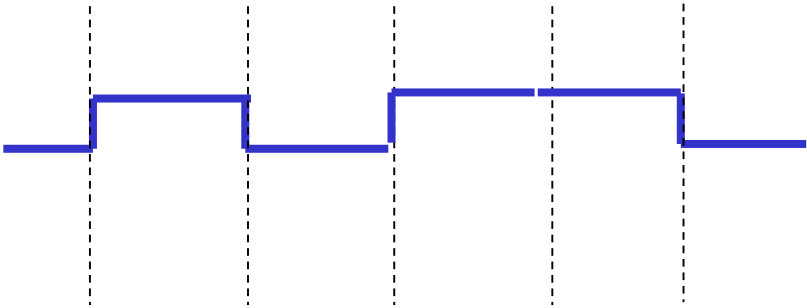
- 1) 数的算术运算;
- 2) 逻辑（控制）运算;

# 三、数字信号的表示

数字信号

电平信号

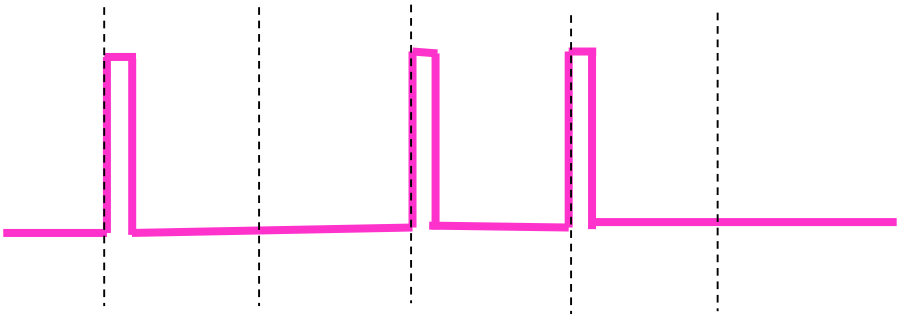
以一个节拍内信号是高、低电平来表示“1”或“0”



10110

脉冲信号

以一个节拍内有无脉冲来表示“1”或“0”





## § 1.2 数制与码制

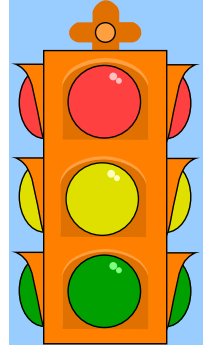
### 1、数制

- 多位数码中每一位的构成方法
- 从低位到高位进位的规则

复习

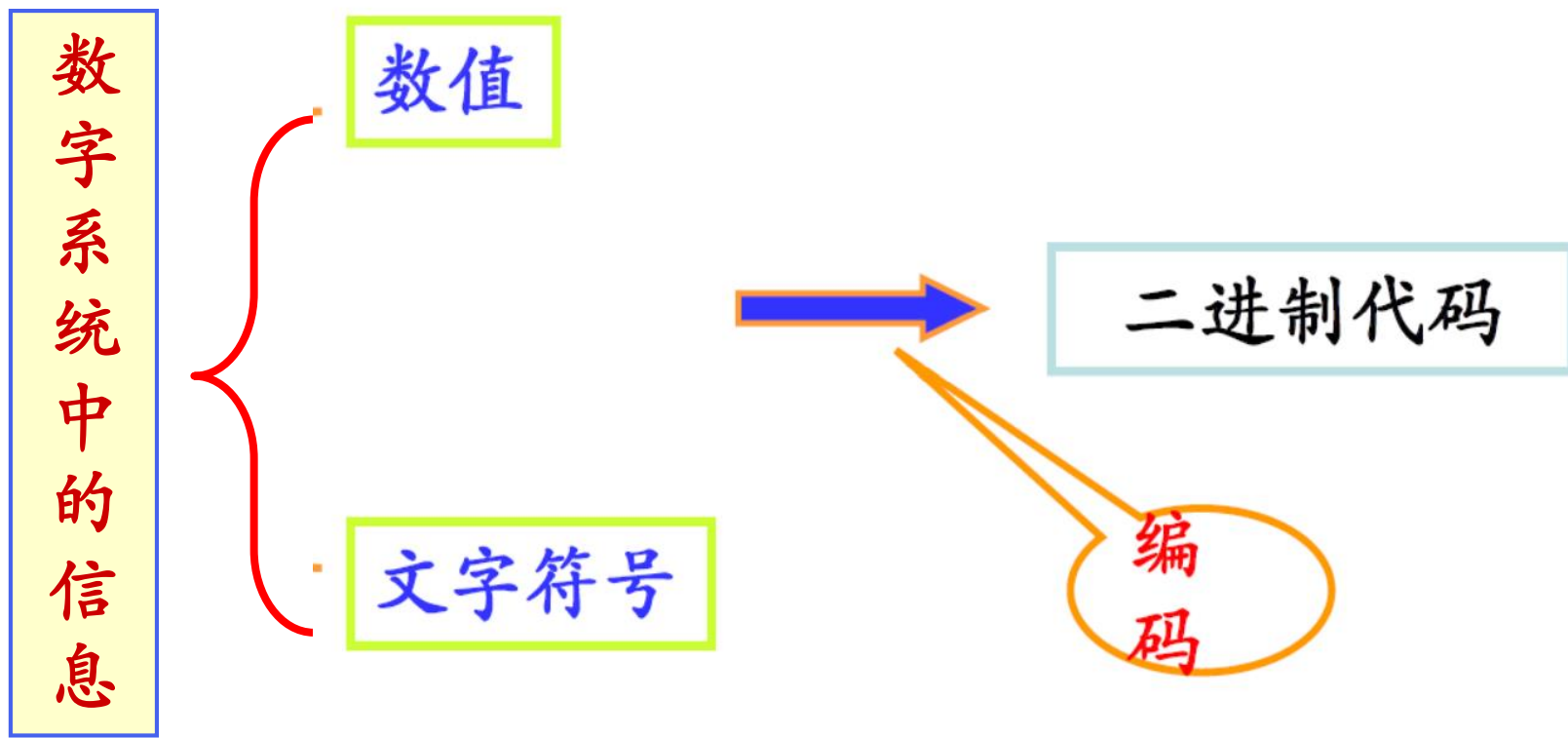
Decimal	Binary	Octal	Hexadecimal
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

## 二、常用编码



**编码：**用文字、符号或数码来表示某种信息的过程。





N项信息进行编码，要求二进制代码的位数 $n$ 应满足

$$2^n \geq N$$



ASCII 码 (用7位二进制数进行编码)

## 1. 二~十进制编码 (BCD码)

用四位二进制代码表示一位十进制数的计数符号的编码方法。

### 1.1. 有权 BCD 码

每一位都有固定权值的BCD码。

{ 8421码  
2421码  
5421码

#### 1) 8421码

用0000~1001 (自然二进制) 代表0~9。权值从高往低分别为8,4,2,1。  
冗余1010--1111。

8421码是一种人机联系时广泛使用的中间形式

十进制数	8421码
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

直接按位转换

$$(28)_{10} = (00101000)_{8421}$$

$$(28)_{10} = (11100)_2$$

Decimal	Binary	BCD
0	0000	0000
1	0001	0001
2	0010	0010
3	0011	0011
4	0100	0100
5	0101	0101
6	0110	0110
7	0111	0111
8	1000	1000
9	1001	1001
10	1010	00010000
11	1011	00010001
12	1100	00010010
13	1101	00010011
14	1110	00010100
15	1111	00010101

注意  
区别

## 8421码的特点:

- 1) 1010—1111为冗余码
- 2) 运算时按逢10进1的原则, 并且要进行调整。

例:  $8+9=17$

$$\begin{array}{r} 1000 \\ +) 1001 \\ \hline \text{进位} \leftarrow 10001 \\ +) 0110 \\ \hline 0111 \end{array}$$

例:  $7+6=13$

$$\begin{array}{r} 0111 \\ +) 0110 \\ \hline 1101 \\ +) 0110 \\ \hline 10011 \end{array}$$

调整原则: 有进位或出现冗余码 (和 $>9$ ) 时, +6调整。

判9加6

## 2) 2421 码

权值由高到低分别为：2, 4, 2, 1。

- 不允许出现0101~1010的6种状态。
- 取对9的自补码

执行十进制数相加时，能正确地产生进位信号。

十进制	2.4.2.1 码	
0	0000	
1	0001	
2	0010	
3	0011	
4	0100	
5	1011	
6	1100	
7	1101	
8	1110	
9	1111	



## 1.2. 无权 BCD 码 (余3码)

将8421码的每个码都加0011。

- 是一种无权码。
- 有六个冗余码。  
(0000、0001、0010、1101、1110、1111)
- 对9的自补码。

十进制	余 3 码	
0	0011	
1	0100	
2	0101	
3	0110	
4	0111	
5	1000	
6	1001	
7	1010	
8	1011	
9	1100	

执行十进制数相加时，能正确地产生进位信号。

## 2、可靠性编码

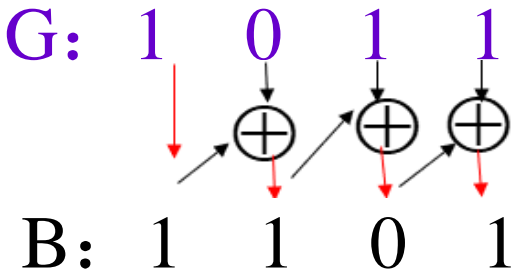
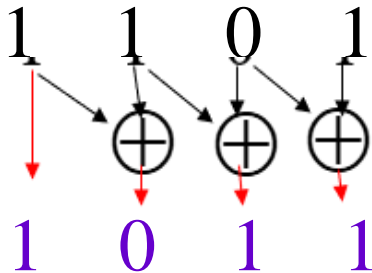
为减少错误的产生，或者能检测出错误的发生，所设定的码制。

### 2.1. 循环码（典型格雷码）

多种编码方案，相邻的两个代码只有1位取值不同。减少传输错误。

$$\begin{cases} G_{n-1} = B_{n-1} \\ G_i = B_i \oplus B_{i+1} \quad 0 \leq i \leq n-2 \end{cases}$$

例：13的格雷码：



Decimal	Binary	Gray code
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

十进制	余 3 码	10进制余3格雷码
0	0011	0010
1	0100	0110
2	0101	0111
3	0110	0101
4	0111	0100
5	1000	1100
6	1001	1101
7	1010	1111
8	1011	1110
9	1100	1010

## 2.2、奇偶校验码

用来检验在传送过程中是否产生错误的代码。码中1的个数强制为奇数（或偶数）。

### 1) 组成（两部分）：

校验码



信息位 —— 位数不限

“a”：1100001

检验位 —— 仅1位

奇校验：01100001

### 2) 编码方式（两种）：

编码方式



奇校验 —— 校验码中 “1”的个数为奇数

偶校验 —— 校验码中 “1”的个数为偶数

信息位 (7位)	采用奇检验的检验位 (1位)	采用偶检验的检验位 (1位)
1001101	<b>1</b> 1001101	<b>0</b> 1001101

### 3) 特点

- (1) 编码简单、容易实现 ；
- (2) 奇偶检验码只有检错能力，没有纠错能力 ；
- (3) 只能发现单错，不能发现双错 。

## § 1.3 逻辑函数及其描述工具

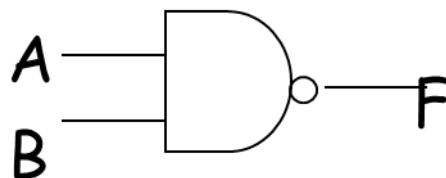
六种表示方法

真值表:

逻辑代数式

$$F = A\bar{B} + \bar{A}B$$

逻辑图:



卡诺图

波形图

硬件描述语言

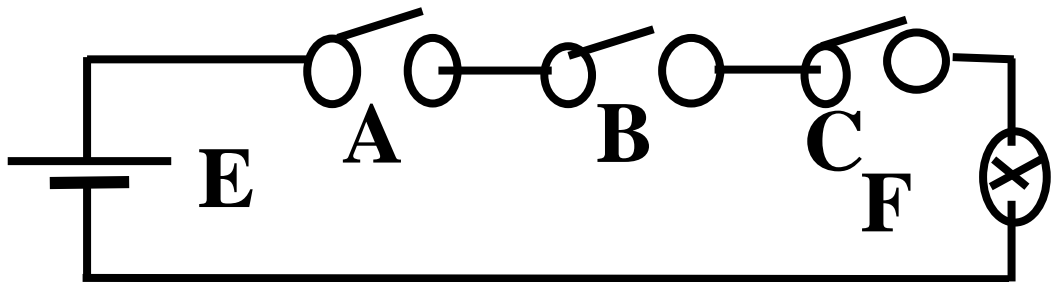
# 一、基本逻辑运算及逻辑门

**逻辑代数**表示逻辑思维，将逻辑归结为一种代数演算的数学工具。  
(George Boole、Shannon)。是数字系统的理论基础和重要数学工具！



逻辑代数不表示数量的大小关系，而是逻辑因果关系。  
它的三种基本运算式：与、或、非。

## 1. 与运算 (逻辑乘)



开关 (A, B, C) 断开为0、闭合为1;  
灯F灭为0; 亮为1;

A、B、C都具备时，事件F才发生。

### (1) 函数式:

输入与输出之间的关系写成与、或、非等运算的组合式。

$$F = A \cdot B \cdot C$$



## (2) 真值表:

将变量的所有取值组合与相应函数F 的值列成的表。

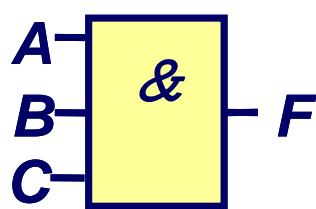
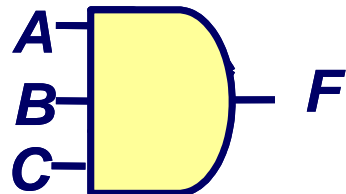
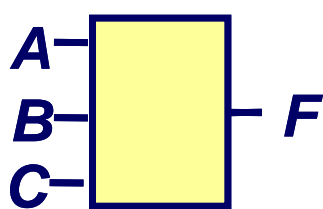
真值表具有唯一性。

逻辑功能：输入有0, 输出为0

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

## (3) 逻辑图:

用逻辑符号表示形成的逻辑电路图。

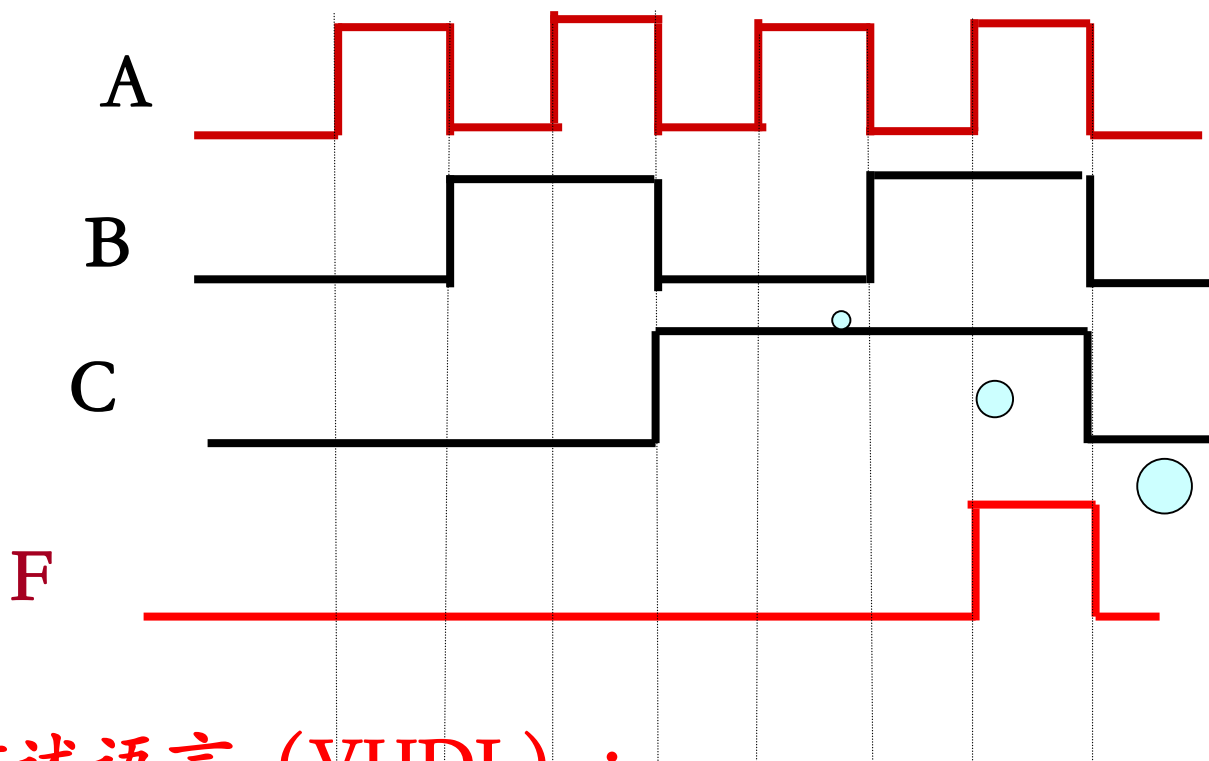


(a) 国家标准 (b) 国外流行 (c) 国际标准

#### (4) 波形图:

$$F = A \cdot B \cdot C$$

用电平的高低表示逻辑变量变化的图形。

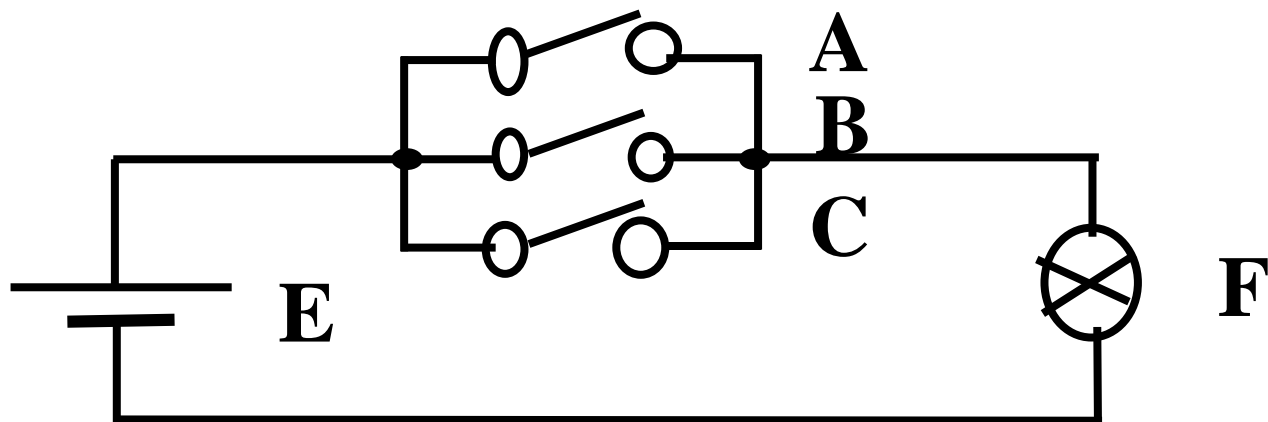


输入有变，  
输出可变

#### (5) 硬件描述语言 (VHDL) :

$$F <= A \text{ and } B \text{ and } C$$

## 2. 或运算

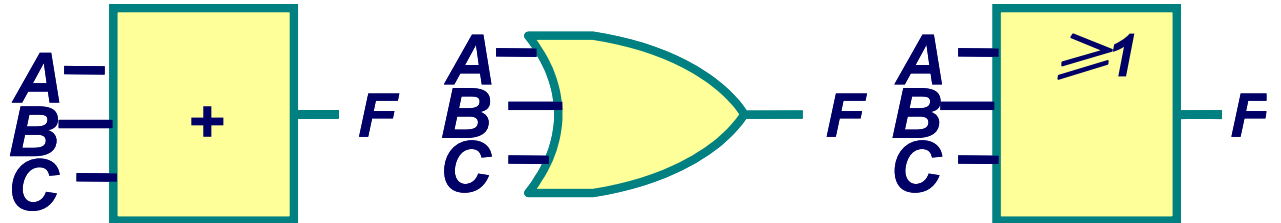


A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

**A**、**B**、**C**只要有一个具备时，事件**F**就发生。

$$\mathbf{F=A+B+C}$$

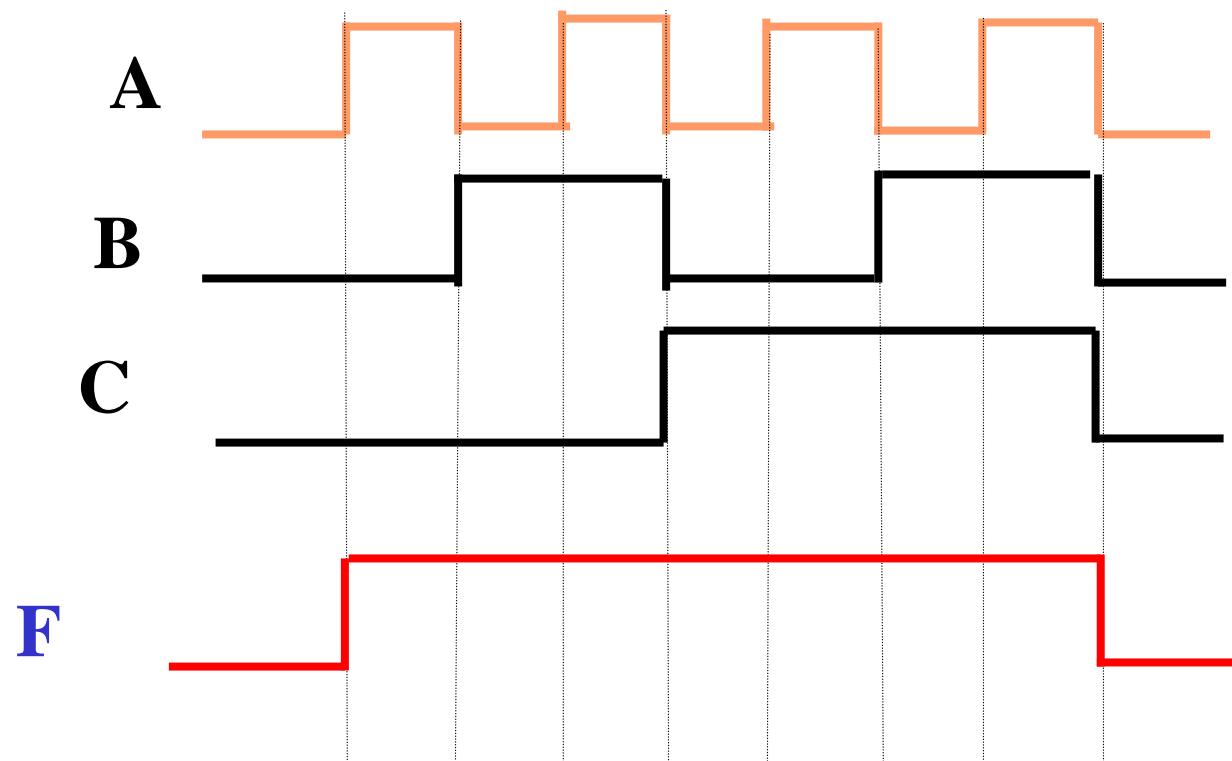
逻辑功能：输入有1，输出为1



(a)

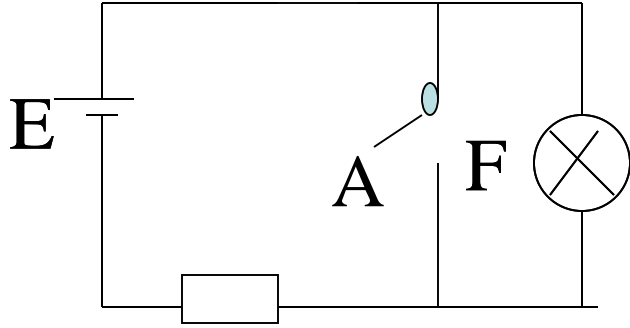
(b)

(c)

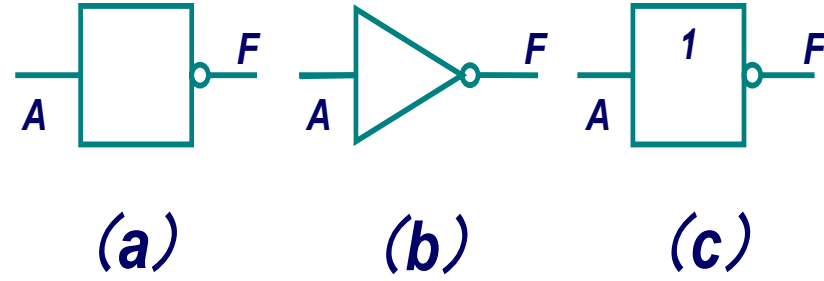


$F \leq A \text{ or } B \text{ or } C$

### 3. 非运算

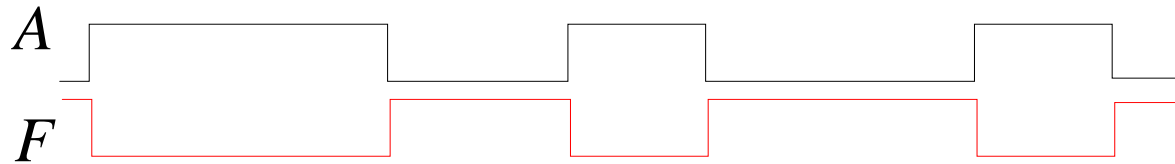


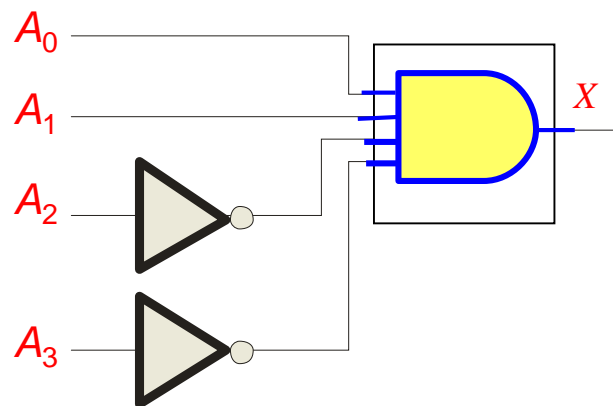
A	F
0	1
1	0



$$F = \overline{A}$$

**$F \leq \text{not } A$  (VHDL)**





**$A=0011$ 时 $x=1$ ，则？ =**

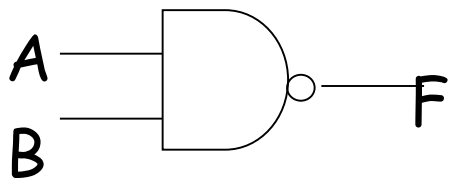
**译  
码**

## 4. 复合逻辑运算

### 1) 与非、或非逻辑

• 与非:

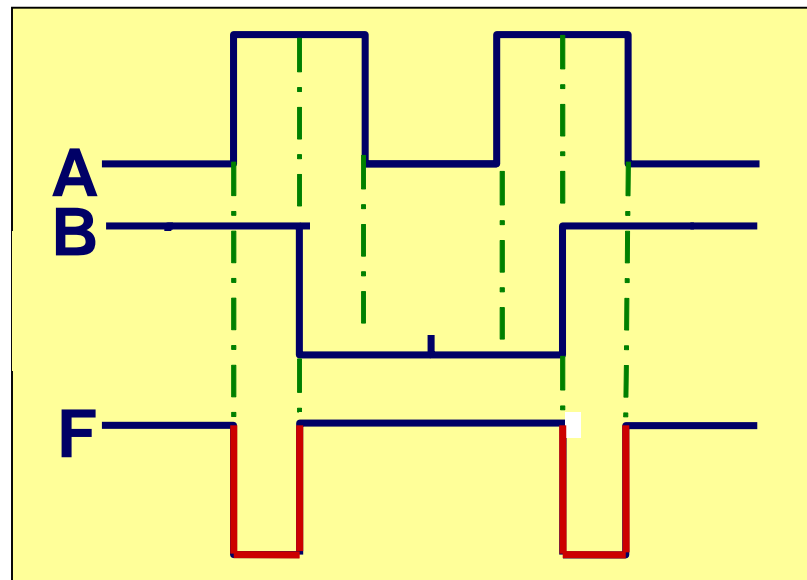
$$F = \overline{AB}$$



$F \leq A \text{ nand } B$  (VHDL)

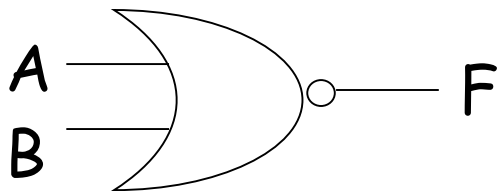
运算规则: 先与/后非

A	B	F
0	0	1
0	1	1
1	0	1
1	1	0



•或非:

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



$$F \leq A \text{ nor } B \quad (\text{VHDL})$$

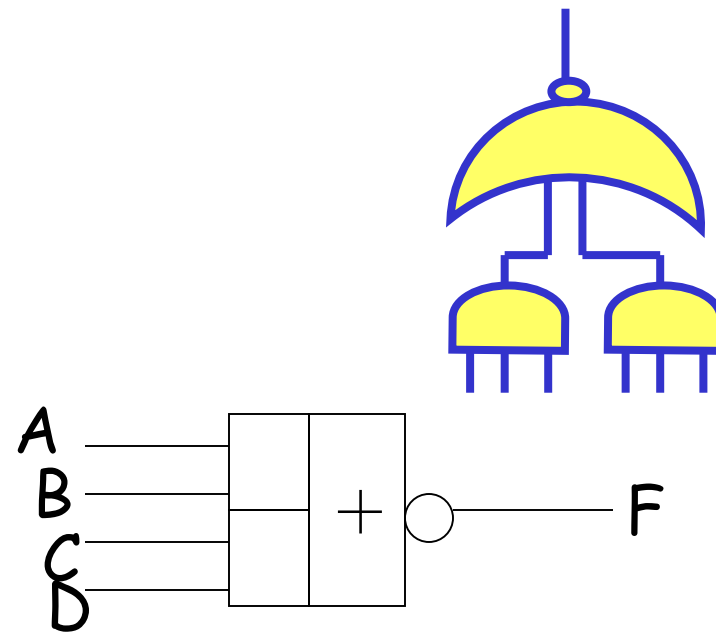
A	B	F
0	0	1
0	1	0
1	0	0
1	1	0

运算规则: 先或/后非

## 2) 与或非运算

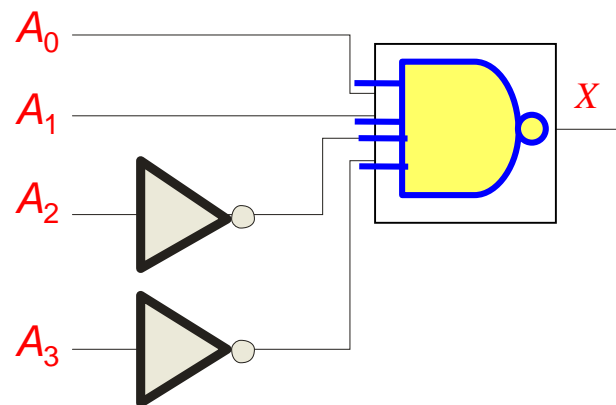
$$F = \overline{AB + CD}$$

运算规则: 先与/后或/再非



逻辑符号





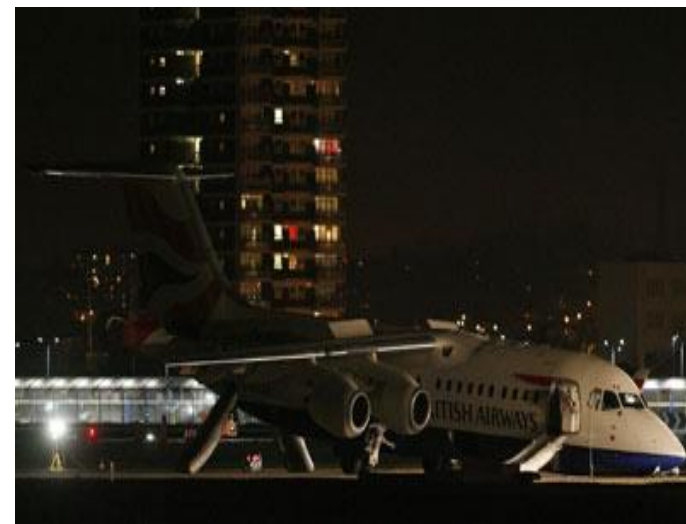
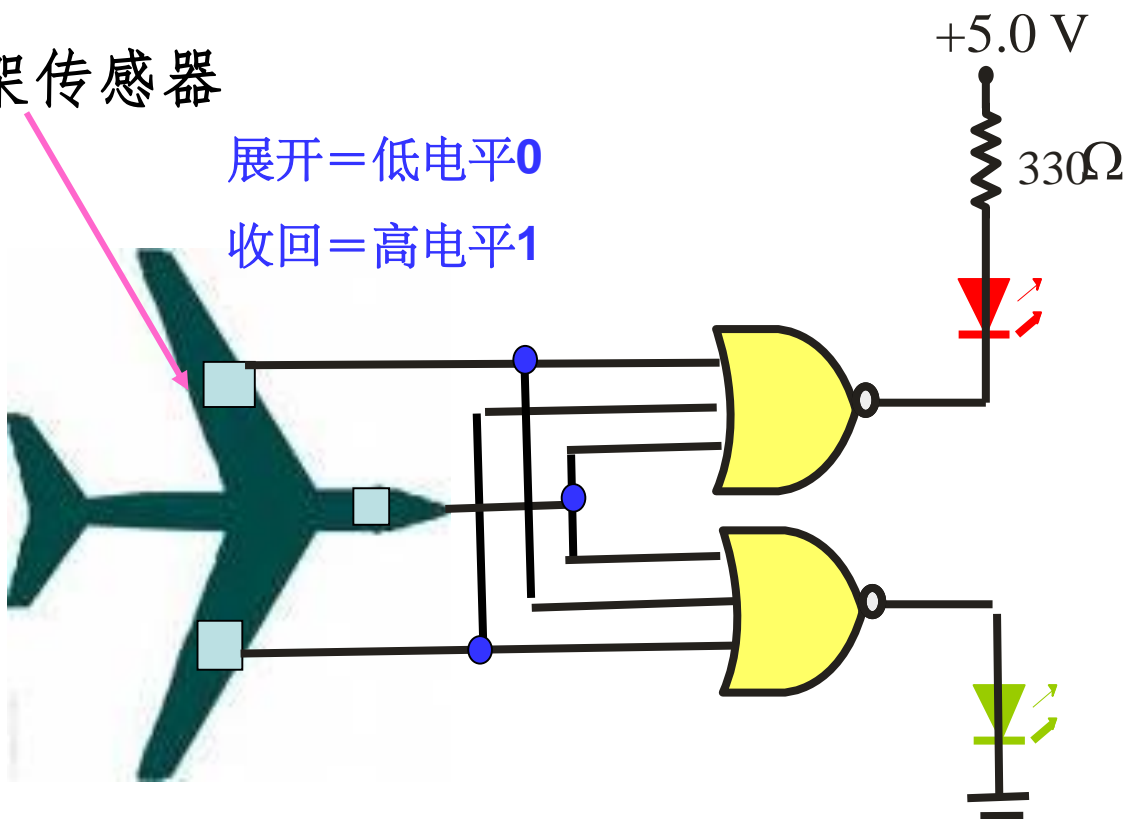
**$A=0011$ 时 $x=0$ ，则？ =**

# 飞机着陆功能检测系统

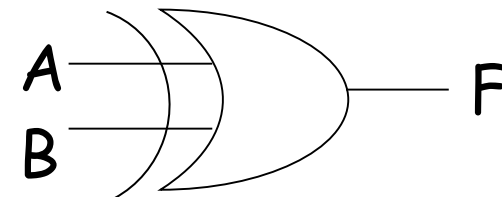
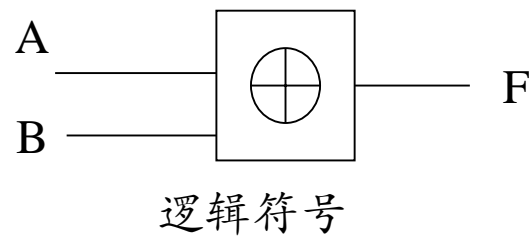
起落架传感器

展开=低电平0

收回=高电平1

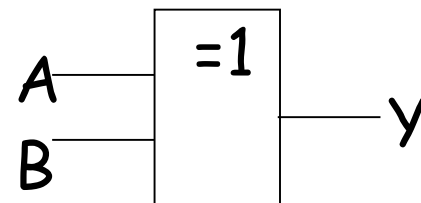


### 3) 异或逻辑



- 两变量异或:

A、B相异，输出为1。 “按位加”



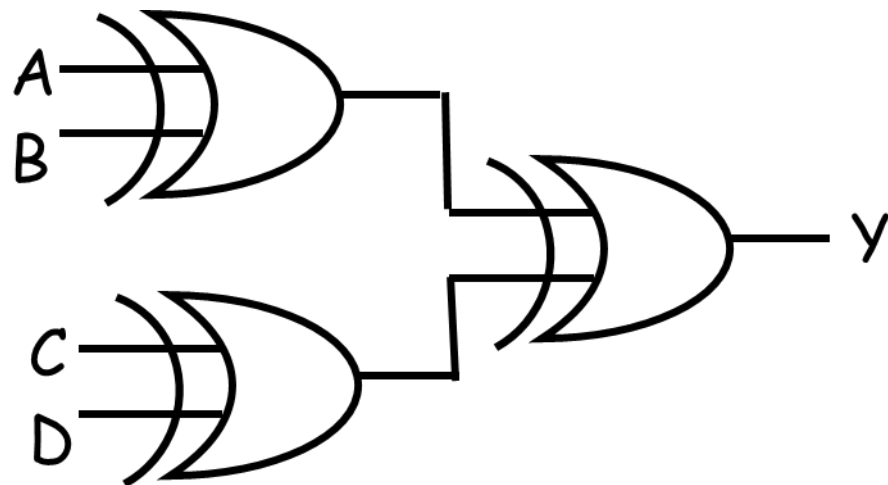
$$F = A \oplus B = A\bar{B} + \bar{A}B$$

$F \leq A \text{ xor } B$  (VHDL)

A	B	F
0	0	0
0	1	1
1	0	1
1	1	0

- 多变量的“异或”

$$Y = A \oplus B \oplus C \oplus D$$



- “异或”电路的特殊功能



奇偶检测电路

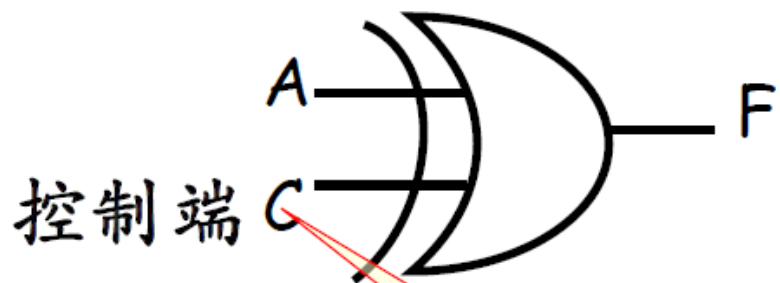
输入变量有奇数个1，则输出为1；

否则，输出为0。



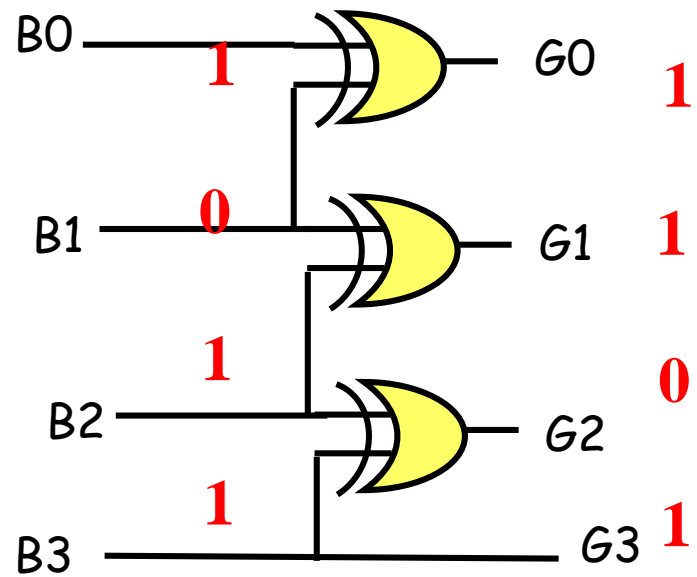
$A, \bar{A}$  控制电路:

$$A \oplus 0 = A \quad A \oplus 1 = \bar{A}$$



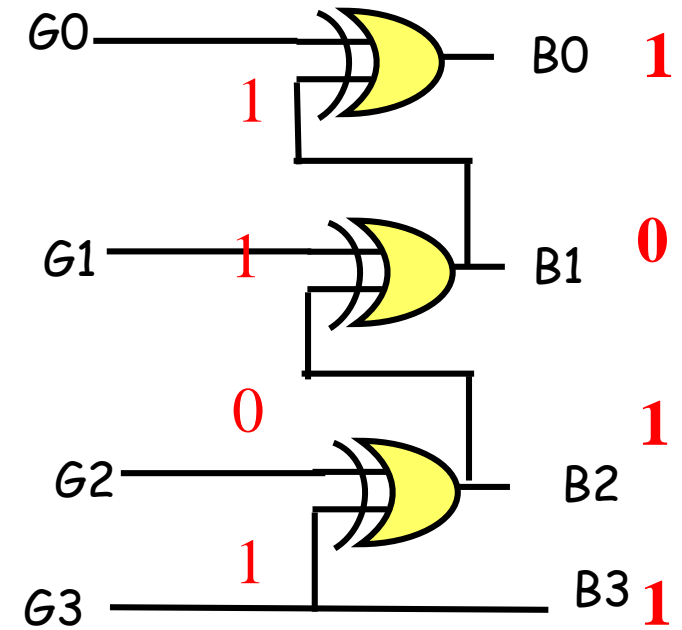
$C = 0$  时,  $F = A$

$C = 1$  时,  $F = \bar{A}$



## Binary to Gray

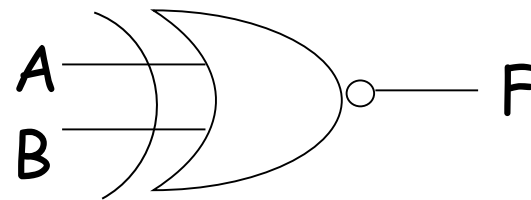
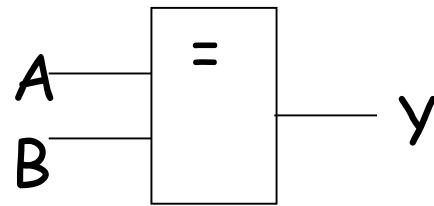
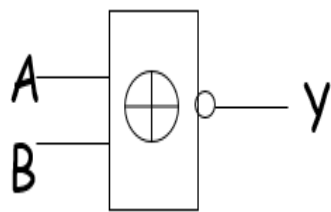
$$\begin{cases} G_3 = B_3 \\ G_2 = B_3 \oplus B_2 \\ G_1 = B_2 \oplus B_1 \\ G_0 = B_1 \oplus B_0 \end{cases}$$



## Gray to Binary

$$\begin{cases} B_3 = G_3 \\ B_2 = B_3 \oplus G_2 \\ B_1 = B_2 \oplus G_1 \\ B_0 = B_1 \oplus G_0 \end{cases}$$

#### 4) 同或（异或非）：



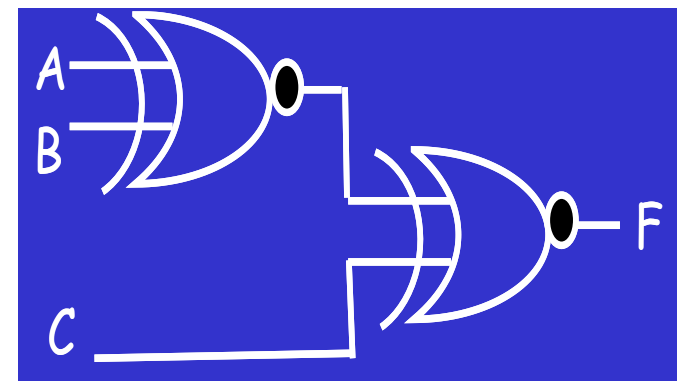
A、B相同，输出为1。

$$F = A \odot B = AB + \overline{A}\overline{B} = \overline{A \oplus B}$$

偶数个变量：同或、异或互为非；

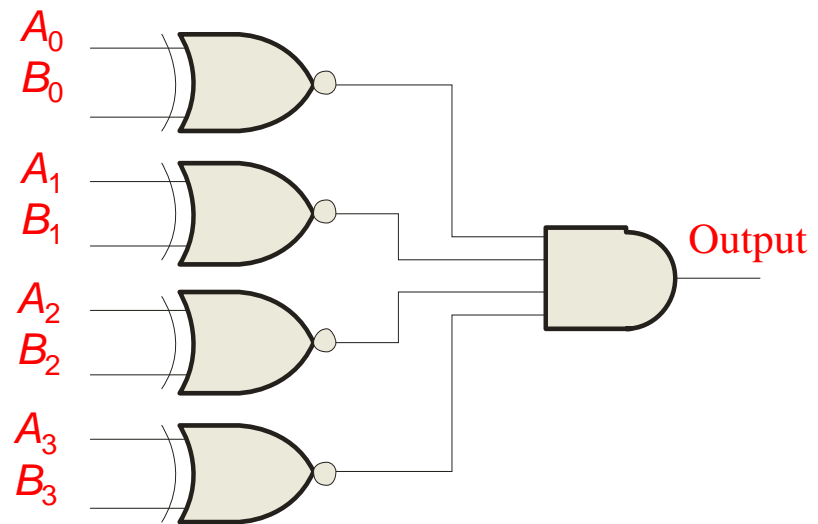
奇数个变量：同或、异或相等。

A	B	F
0	0	1
0	1	0
1	0	0
1	1	1



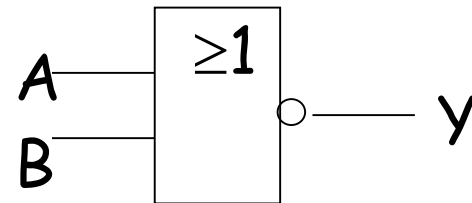
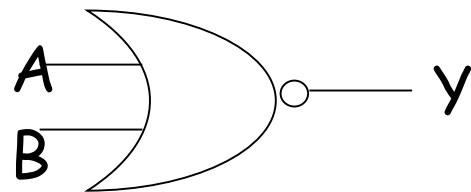
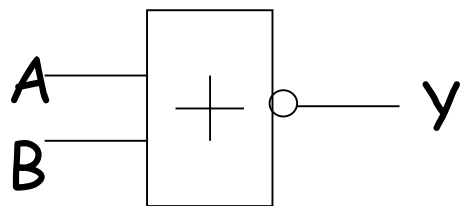
$A=B$  (?)

比较器

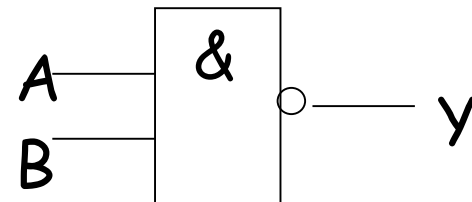
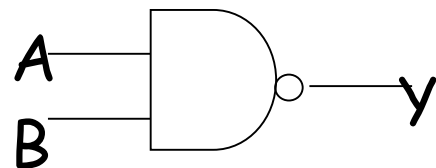
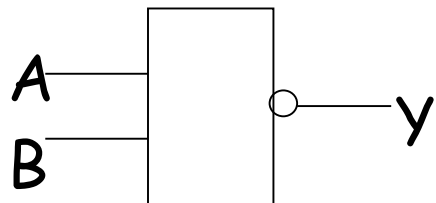




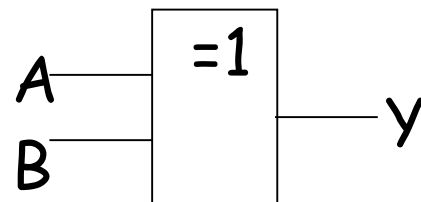
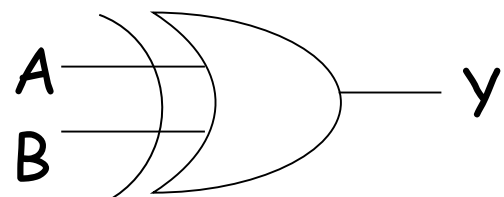
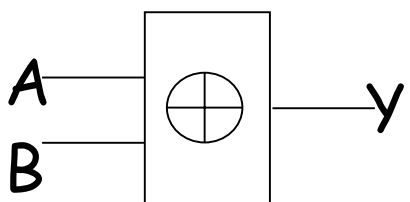
或非门



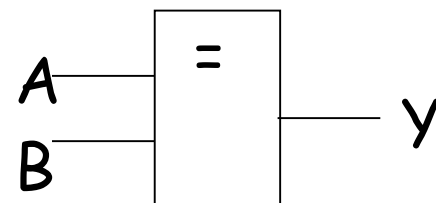
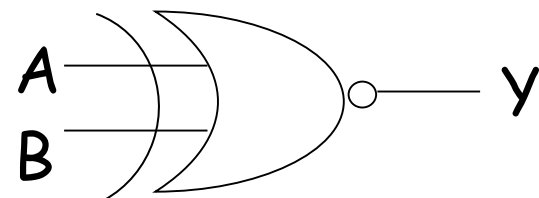
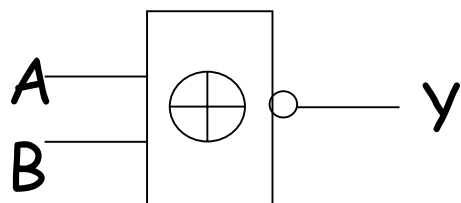
与非门



异或门



异或非门



(a)常用符号

(b)美、日常用符号

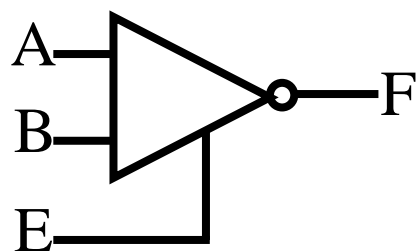
(c)国标符号

## 一. 三态门 (TSL 门)

**TSL 门 (Three State Logic)** : 输出有0、1两种状态外, 还有高阻状态 (禁止态)。

### 1. 基本概念:

(1) 使能端高电平有效:



日美常用符号

E (Enable) 使能端。

E	A	B	F
0	X	X	高阻
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

$E=1$  时, 输出  $F=\overline{AB}$ ;

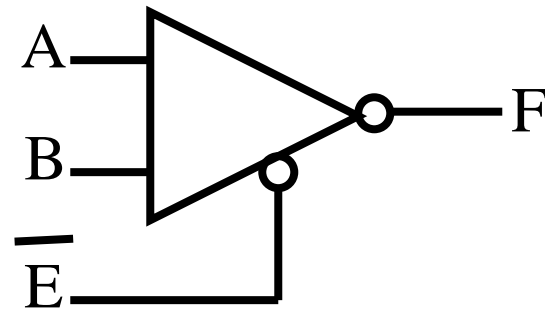
$E=0$  时, 与非门禁止, 输出呈现高阻状态。

(2) 使能端低电平有效:

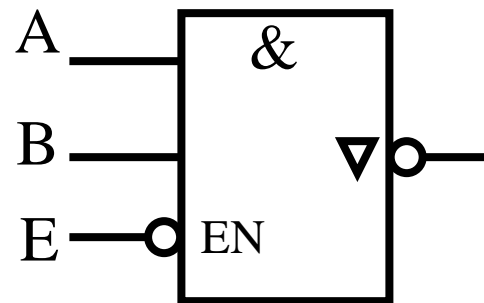
E	A	B	F
1	X	X	高阻
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0

当  $E=0$  时,  $F=\overline{AB}$ ;

当  $E=1$  时, 与非门禁止, 输出呈现高阻状态。



日美常用符号



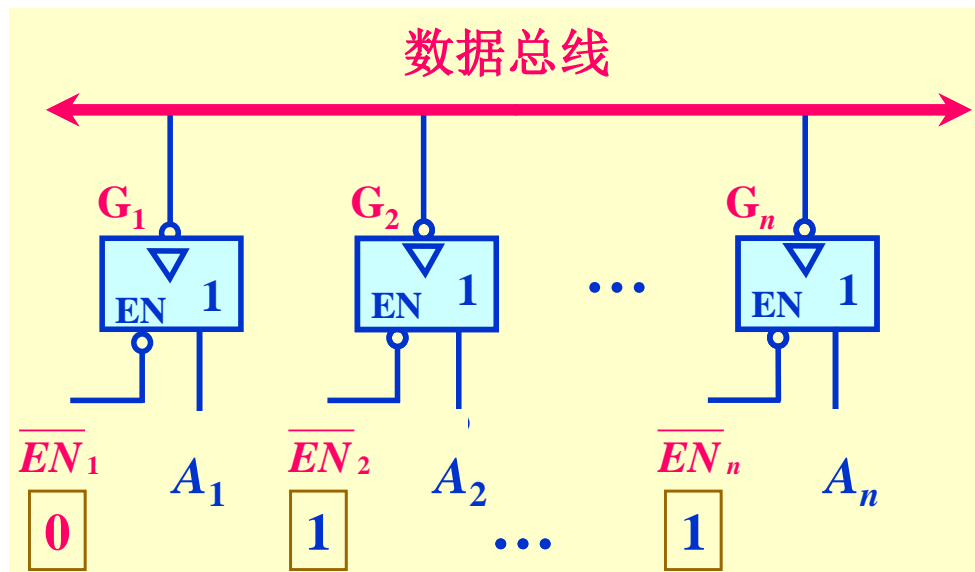
国标符号

## 2. 基本应用

是在数字系统中构成总线（Bus）。

### (1) 单向总线

信号的分时传送（选择传送）

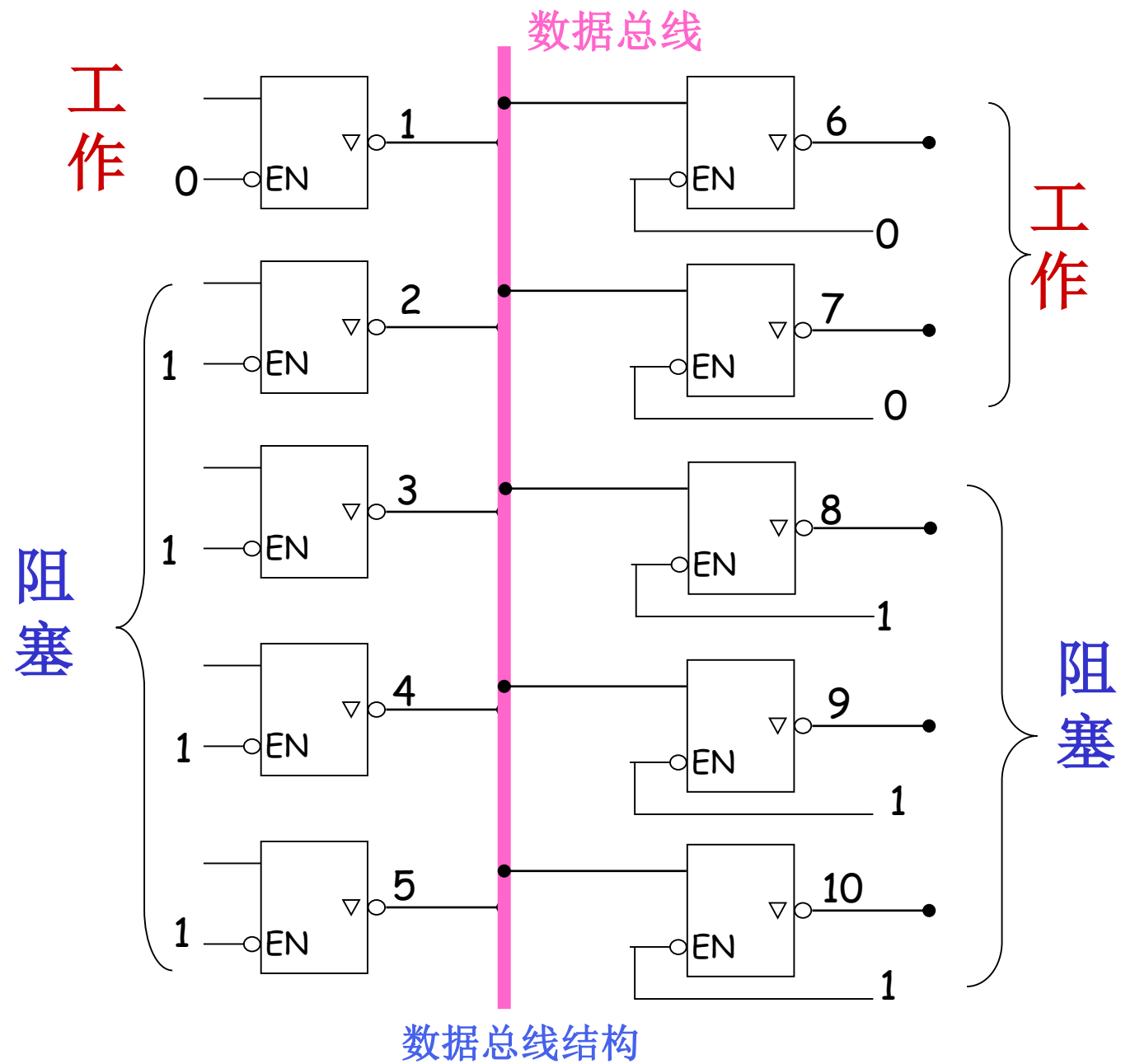


$E_1$	$E_2$	$E_3$	Y
0	1	1	$\overline{A_1}$
1	0	1	$\overline{A_2}$
1	1	0	$\overline{A_3}$
1	1	1	高阻

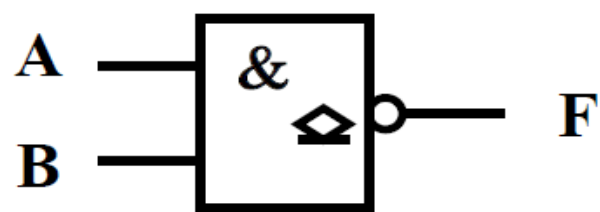
注意：

任何时刻，发送数据时，只允许一个三态门使能，其余为高阻态。





## 二. 集电极开路与非门 (OC 门 open-collector)

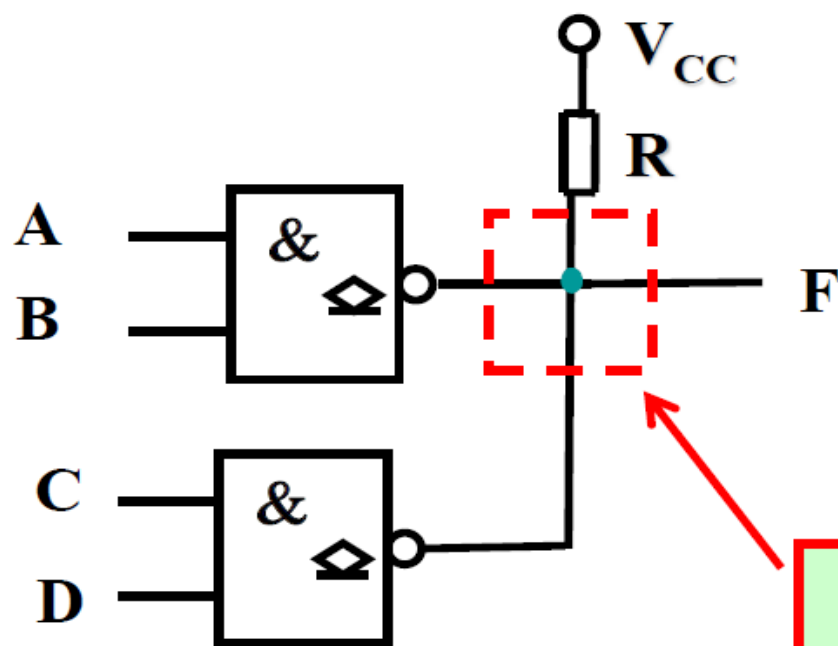


$$F = \overline{A B}$$

OC门使用时应注意两点:

① OC门输出端与电源间应接一个电阻。

② OC门并联使用时, 具有“线与”功能。



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

线与