

# 第3讲 符号化计算化自动化—— 一看计算机的本质

---

黄宏

华中科技大学计算机学院

[honghuang@hust.edu.cn](mailto:honghuang@hust.edu.cn)

# 第3讲 符号化、计算化与自动化——看计算机的本质

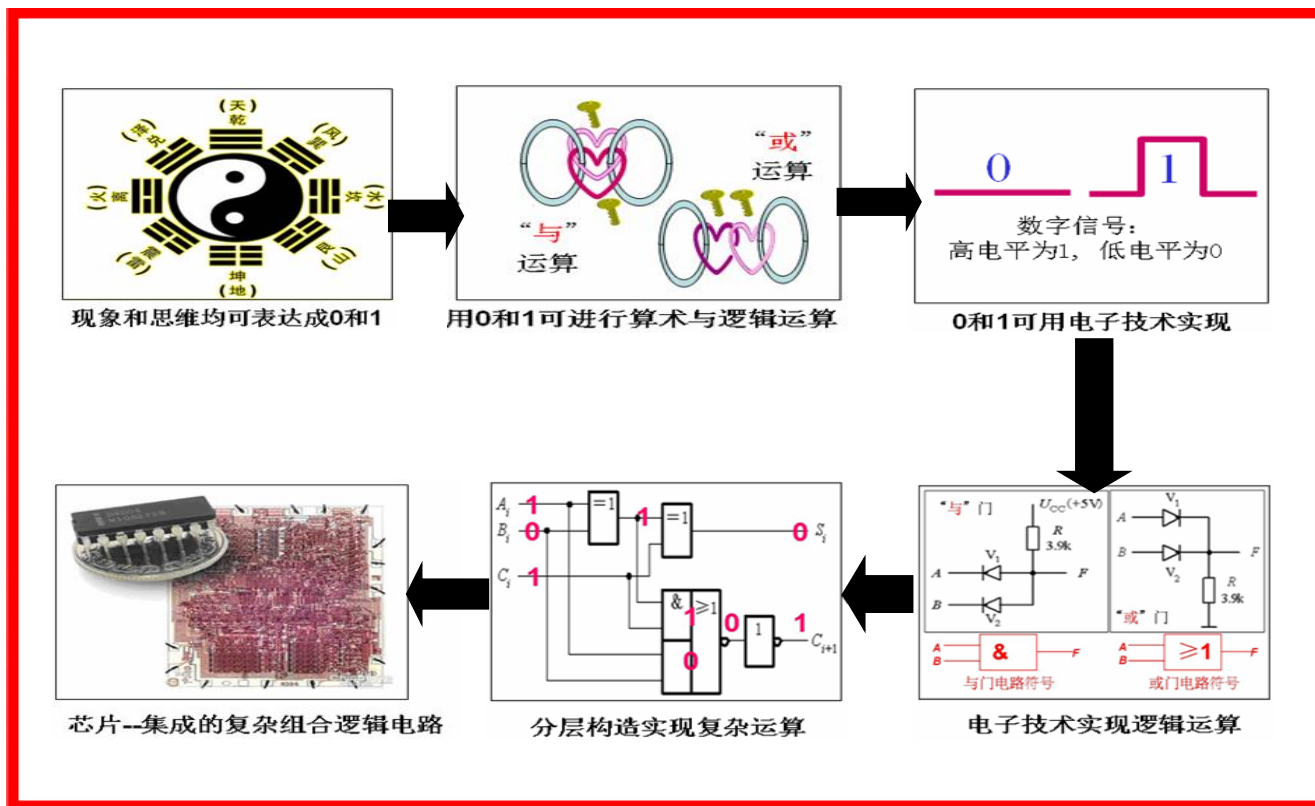
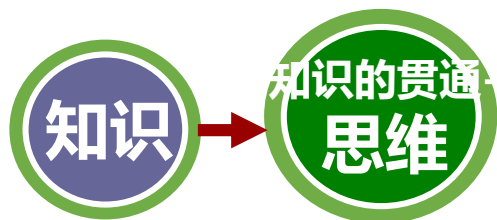
2

- 一、符号化、计算化与自动化——计算机的本质**
- 二、数值性信息的表达：计数制与机器数**
- 三、非数值性信息的表达：编码与组合**
- 四、0和1的计算：基本逻辑运算**
- 五、用电子技术实现0和1及其逻辑运算**
- 六、符号化-计算化的综合应用：利用图像隐藏信息**

# 符号化、计算化与自动化：0和1思维概述

3

## (1) 本讲要理解的计算思维



表层含义

深层含义

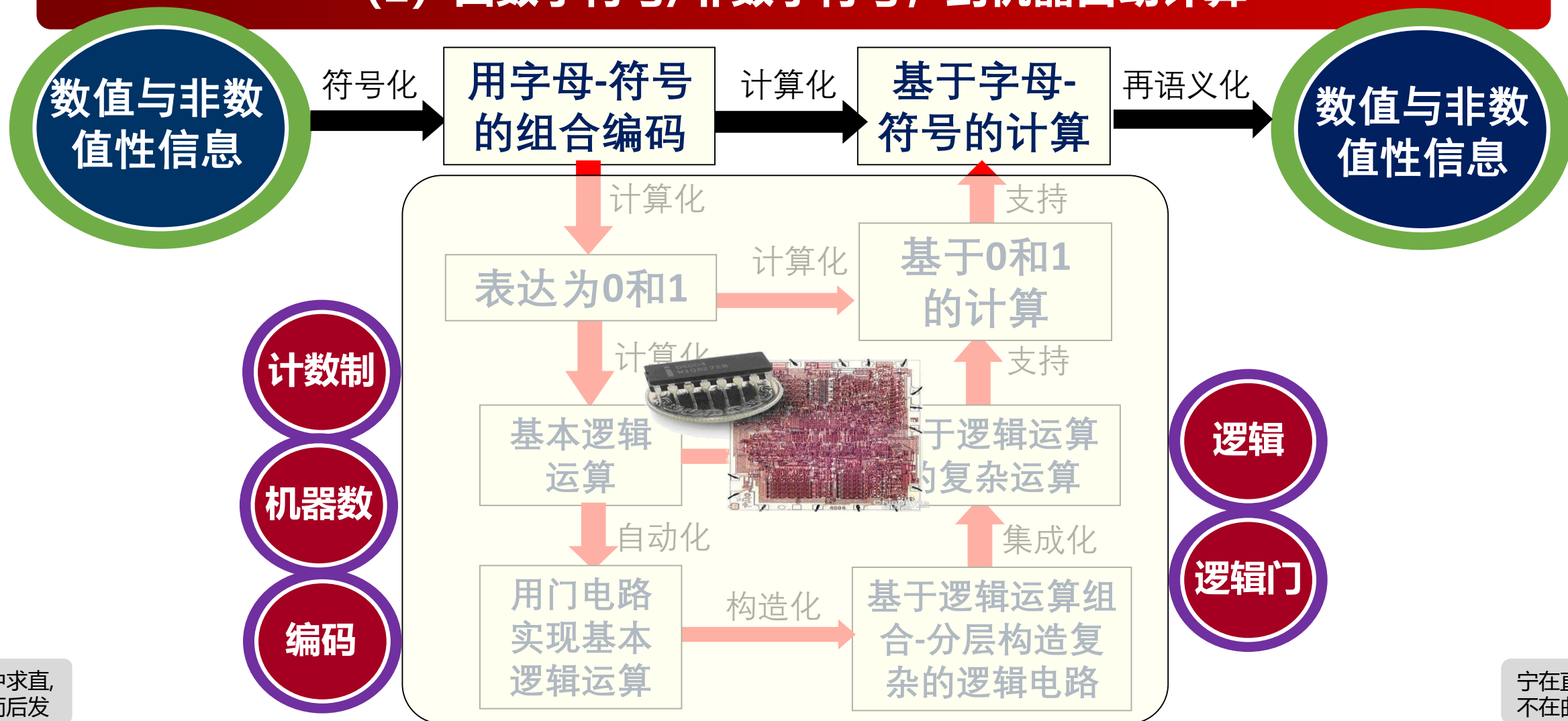
集成含义

语义符号化 → 符号计算化 → 计算0(和)1化 → 0(和) 1自动化 → 分层构造化 → 构造集成化

# 符号化、计算化与自动化：0和1思维概述

4

## (2) 由数学符号/非数学符号，到机器自动计算



# 第3讲 符号化、计算化与自动化——看计算机的本质

5

- 一、符号化、计算化与自动化——计算机的本质**
- 二、数值性信息的表达：计数制与机器数**
- 三、非数值性信息的表达：编码与组合**
- 四、0和1的计算：基本逻辑运算**
- 五、用电子技术实现0和1及其逻辑运算**
- 六、符号化-计算化的综合应用：利用图像隐藏信息**

# 进位计数制

6

## (1) 进位计数制与十进制

◆进位计数制：用数码和带有权值的数位来表示有大小关系的数值性信息的表示方法。

◆十进制

- 有0,1,2,3,4,5,6,7,8,9共十个数码
- 数码的位置规定了数码的等级“权/数位”： $10^i$
- 逢十进一、借一当十。高数位的1相当于低数位的10
- “十”----基值，十进制

$$\begin{array}{cccccc} (2 & 4 & 5 & . & 2 & 5)_{+} \\ | & | & | & & | & | \\ 2 & 1 & 0 & & -1 & -2 \\ 10^2 & 10^1 & 10^0 & & 10^{-1} & 10^{-2} \end{array}$$

$$(245.25)_{+} = 2 \times 10^2 + 4 \times 10^1 + 5 \times 10^0 + 2 \times 10^{-1} + 5 \times 10^{-2}$$

# 进位计数制

7

## (2) 二进制

- 有0,1共两个数码
- 数码的位置规定了数码的等级“权/数位”：  $2^i$
- 逢二进一、借一当二。高数位的1相当于低数位的2
- “二”----基值，二进制

$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$	.	$2^{-1}$	$2^{-2}$	—————	数位的权值
7	6	5	4	3	2	1	0	.	-1	-2	—————	数位

例如：  $(1\ 1\ 1\ 1\ 0\ 1\ 0\ 1\ .\ 0\ 1)_2$  ————— 二进制数

$$\begin{aligned} &= 1 \times 2^7 + 1 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 \\ &+ 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} = (245.25)_+ \end{aligned}$$

# 进位计数制

8

## (3) 任意进制 暨 r进制

- 有0,1,...,共r个数码
- 数码的位置规定了数码的等级“权/数位”： $r^i$
- 逢r进一、借一当r。高数位的1相当于低数位的r
- “r”----基值，r进制



$$\begin{array}{cccccccccccccccc} r^{n-1} & r^{n-2} & \dots & r^2 & r^1 & r^0 & . & r^{-1} & r^{-2} & \dots & r^{-m} & & \text{数位的权值} \\ n-1 & n-2 & \dots & 2 & 1 & 0 & . & -1 & -2 & \dots & -m & & \text{数位} \end{array}$$

$$(d_{n-1}d_{n-2}\dots\dots d_2d_1d_0 . d_{-1}d_{-2}\dots\dots d_{-m})_r \quad \text{——— r进制数}$$

$$= d_{n-1}r^{n-1} + d_{n-2}r^{n-2} + \dots + d_2r^2 + d_1r^1 + d_0r^0 + d_{-1}r^{-1} + d_{-2}r^{-2} + \dots + d_{-m}r^{-m}$$

$$= \sum_{i=-m}^{n-1} d_i r^i$$



## (4) 几种特殊的r进制

- ◆ 十六进制: 0,1,2,3,4,5,6,7,8,9,A(10), B(11), C(12), D(13), E(14), F(15)
- ◆ 八进制: 0,1,2,3,4,5,6,7
- ◆ 十进制: 0,1,2,3,4,5,6,7,8,9

$(365.2)_{+}$ ,  $(11011.01)_{=}$ ,  $(3460.32)_{八}$ ,  $(596.12)_{+六}$

# 进位计数制

10

## (5) 一个数值用不同进制表示，其数码串是不同的

$$\begin{aligned}(753.37)_8 &= 753.37_0 \\ &= 7 \times 8^2 + 5 \times 8^1 + 3 \times 8^0 + 3 \times 8^{-1} + 7 \times 8^{-2} \\ &= (491.484375)_+\end{aligned}$$

$$\begin{aligned}(753.37)_{16} &= 753.37_H = 0x753.37 \\ &= 7 \times 16^2 + 5 \times 16^1 + 3 \times 16^0 + 3 \times 16^{-1} + 7 \times 16^{-2} \\ &= (1875.2148)_+\end{aligned}$$

$$\begin{aligned}(753.37)_{12} &= 753.37_{12} \\ &= 7 \times 12^2 + 5 \times 12^1 + 3 \times 12^0 + 3 \times 12^{-1} + 7 \times 12^{-2} \\ &= (1071.2986)_+\end{aligned}$$

245的十进制表示记为：

245

245的二进制表示记为：

11110101

245的八进制表示记为：

365

245的十六进制表示记为：

F5

同一个数值，用不同进制表达，结果也是不同的

同一个数串，由于进制不同其所表达的数值大小也是不同的

# 进位计数制之间的转换

11

## (1) r进制转换为十进制

已知r进制的数码串 $d_m \dots d_{-n}$ , 求十进制的N

$$N = (d_{n-1}d_{n-2}\dots\dots d_2d_1d_0.d_{-1}d_{-2}\dots\dots d_{-m})_r$$
$$= d_{n-1}r^{n-1} + d_{n-2}r^{n-2} + \dots + d_2r^2 + d_1r^1 + d_0r^0 + d_{-1}r^{-1} + d_{-2}r^{-2} + \dots + d_{-m}r^{-m} = \sum_{i=-m}^{n-1} d_i r^i$$

$$\begin{array}{r} 7\ 6\ 5\ 4\ 3\ 2\ 1\ 0\ .\ -1\ -2 \\ \hline (1\ 1\ 1\ 1\ 0\ 1\ 0\ 1\ .\ 0\ 1)_{-} \\ = 1 \times 2^7 + 1 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 \\ + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} = (245.25)_{+} \end{array}$$

$$(F5.4)_{+_{\text{十六}}} = F \times 16^1 + 5 \times 16^0 + 4 \times 16^{-1} = (245.25)_{+}$$

# 进位计数制之间的转换

12

## (2) 十进制转换为r进制—整数部分转换

已知十进制整数N，求r进制的数码串 $d_{n-1}d_{n-2}\dots d_1d_0$

$$\begin{aligned} N &= (d_{n-1}d_{n-2}\dots d_2d_1d_0)_r \\ &= d_{n-1}r^{n-1} + d_{n-2}r^{n-2} + \dots + d_2r^2 + d_1r^1 + d_0r^0 \end{aligned}$$

$(N/r)$ 的余数为  $d_0$

$((N/r)/r)$ 的余数为  $d_1$

$((((N/r)/r)/r)$ 的余数为  $d_2$

...

$(\dots(((N/r)/r)/r)\dots/r)$ 的余数为  $d_{n-1}$

“除基取余”

$$(245)_{+} = (F5)_{+六}$$

$$\begin{array}{r|l} 16 & 245 \\ \hline 16 & 15 \text{ ----- } 5 \\ \hline & 0 \text{ ----- } 15(F) \end{array}$$

# 数值性信息的表达：计数制与机器数

13

## (2) 机器数：机器字长

【机器字长】是指机器内部进行数据处理、信息传输等的基本单元所包含的二进制位数，通常是8位、16位、32位、64位等。

例：(78)<sub>十</sub>转换成二进制数。不同字长的表示，结果是不同的

$$(1) \quad (78)_{十} = (1001110)_二$$

未考虑机器字长

$$(2) \quad (78)_{十} = (01001110)_二$$

8位机器字长

$$(3) \quad (78)_{十} = (00000000 \ 01001110)_二$$

16位机器字长

$$(4) \quad (78)_{十} = (00000000 \ 00000000 \ 00000000 \ 01001110)_二$$

32位机器字长

机器  
字长

数的表  
示范围

溢出

# 数值性信息的表达：计数制与机器数

14

## (3) 机器数：有符号数和无符号数

### 16位机器字长

无符号数

无符号机器数的表示范围： $0 \leq X < 2^{16}$ ，即0至65,535

(00000000 00000000) = ——— (11111111 11111111) =

数的符号也用0和1表达，0表“+”号，1表“-”号

有符号数

(**0**0000000 00000000) = ( **0**1111111 11111111) =

(**1**0000000 00000000) = ( **1**1111111 11111111) =

有符号数的  
大小？

有符号数的  
表示范围？

# 数值性信息的表达：计数制与机器数

15

## (4) 带符号的机器数：原码、反码和补码

十进制数	正数	+ 244	- 244	负数
二进制数	+ 1 1 1 1 0 1 0 0		- 1 1 1 1 0 1 0 0	
机器数-原码	0 1 1 1 1 0 1 0 0		1 1 1 1 1 0 1 0 0	
机器数-反码	0 1 1 1 1 0 1 0 0		1 0 0 0 0 1 0 1 1	
机器数-补码	0 1 1 1 1 0 1 0 0		加	0 0 0 0 0 0 0 1
			1 0 0 0 0 1 1 0 0	

# 数值性信息的表达：计数制与机器数

16

## (5) 机器数：原码、反码和补码

正数

原码、反码和补码是一样的

负数

原码、反码和补码表示的大小是不同的

原码、反码和补码的表示范围也是不同的

真实数值 (带符号的 $n$ 位 二进制数)	十进制数	机器数( $n+1$ 位二进制数,其中第 $n+1$ 位表符号,0 表示正号,1 表示负号)		
		原码	反码	补码
+ 11...11	$+(2^n-1)$	0 11...11	0 11...11	0 11...11
+ 10...00	$+2^{n-1}$	0 10...00	0 10...00	0 10...00
+ 00...00	+0	0 00...00	0 00...00	0 00...00
- 00...00	-0	1 00...00	1 11...11	0 00...00
- 10...00	$-2^{n-1}$	1 10...00	1 01...11	1 10...00
- 11...11	$-(2^n-1)$	1 11...11	1 00...00	1 00...01
-100...00	$-2^n$	-	-	1 00...00
		正数的原码、反码同补码形式是一样的。最高位为 0 表示正数		
		负数的最高位为 1,表示负数。其余同真实数值的二进制数。	负数的最高位为 1,表示负数。其余在真实数值的二进制数基础上逐位取反。	负数的最高位为 1,表示负数。其余在反码基础上最低位加 1 后形成的。它的负数不包括 0, 但包括 $-2^n$
		机器数由于受到表示数值的位数的限制,只能表示一定范围内的数。超出此范围则为“溢出”		



# 数值性信息的表达：计数制与机器数

17

## (6) 减法可变加法、符号位与数值位可一同运算

补码加减法：5位机器字长（1位符号位，4位数值位）

$$(+7) + (+3) = (+10)$$

$$\begin{array}{r} \text{---} \\ 0 \ 0111 \\ +) \ 0 \ 0011 \\ \hline 0 \ 1010 \end{array}$$

$$(10) + (-3) = (7)$$

$$\begin{array}{r} \text{---} \\ 0 \ 1010 \\ +) \ 1 \ 1101 \\ \hline 0 \ 0111 \end{array}$$

$$(-5) + (-7) = (-12)$$

$$\begin{array}{r} \text{---} \\ 1 \ 1011 \\ +) \ 1 \ 1001 \\ \hline 1 \ 0100 \end{array}$$

受字长限制自动舍弃

补码再求补  
即为原码

$$(-7) + (-12) = \text{溢出}$$

$$\begin{array}{r} \text{---} \\ 1 \ 1001 \\ +) \ 1 \ 0100 \\ \hline 0 \ 1101 \end{array}$$

# 第3讲 符号化、计算化与自动化——看计算机的本质

18

- 一、符号化、计算化与自动化——计算机的本质
- 二、数值性信息的表达：计数制与机器数
- 三、非数值性信息的表达：编码与组合
- 四、0和1的计算：基本逻辑运算
- 五、用电子技术实现0和1及其逻辑运算
- 六、符号化-计算化的综合应用：利用图像隐藏信息

# 非数值性信息的表达：编码与组合

19

## (1) 易经：古人研究自然现象/人事现象变化规律的方法

䷋ 阴 (六)

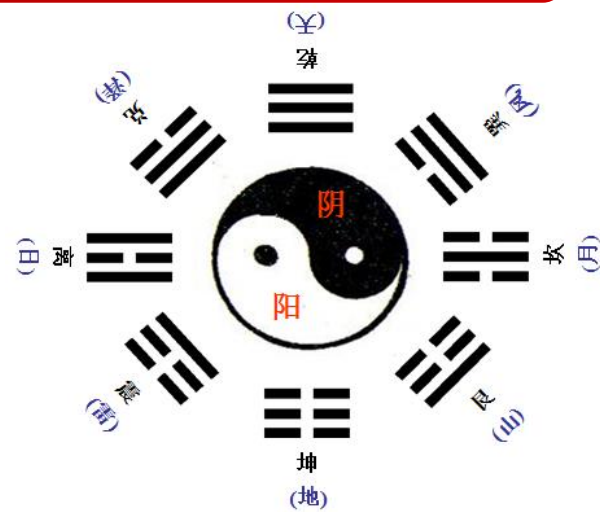
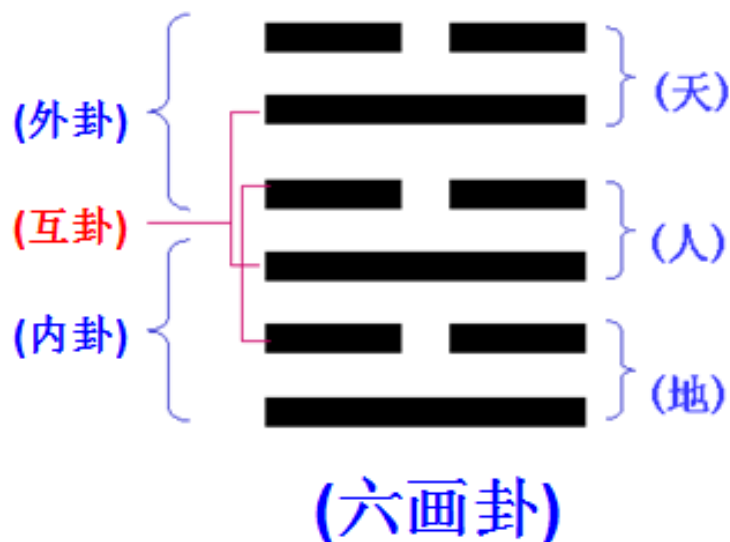
䷍ 阳 (九)

䷃ (九三爻)

䷖ (六二爻)

䷌ (九一爻)

(三画卦)



八纯	䷀ 乾	䷂ 震	䷃ 坎	䷄ 艮	䷁ 坤	䷊ 巽	䷝ 离	䷥ 兑
一世	䷗ 姤	䷌ 豫	䷻ 节	䷶ 贲	䷗ 复	䷗ 小畜	䷗ 旅	䷗ 困
二世	䷫ 遯	䷧ 解	䷌ 屯	䷌ 大畜	䷌ 临	䷌ 家人	䷌ 鼎	䷌ 萃
三世	䷥ 否	䷧ 恒	䷾ 既济	䷾ 损	䷾ 泰	䷾ 益	䷾ 未济	䷾ 咸
四世	䷗ 观	䷗ 升	䷗ 革	䷗ 睽	䷗ 大壮	䷗ 无妄	䷗ 蒙	䷗ 蹇
五世	䷗ 剥	䷗ 井	䷗ 丰	䷗ 履	䷗ 夬	䷗ 噬嗑	䷗ 涣	䷗ 谦
游魂	䷗ 晋	䷗ 大过	䷗ 明夷	䷗ 中孚	䷗ 需	䷗ 颐	䷗ 讼	䷗ 小过
归魂	䷗ 大有	䷗ 随	䷗ 师	䷗ 渐	䷗ 比	䷗ 蛊	䷗ 同人	䷗ 归妹

将现象抽象为符号，进行符号组合，利用符号组合及其变化表达自然现象，进行计算！——一卦是如何变化到另一卦的？

# 非数值性信息的表达：编码与组合

20

## (2) 非数值性信息表达：编码

### 编码

编码是以若干位数码或符号的不同组合来表示非数值性信息的方法，它是人为地将若干位数码或符号的每一种组合指定一种唯一的含义。

### ASCII码

### 英文字母与符号的0, 1型编码方法

规律性：英文字母与符号是按照二进制次序进行编码，连续排列

$b_6b_5b_4b_3b_2b_1b_0$   
**0** $b_6b_5b_4b_3b_2b_1b_0$

机器字长8位  
及其倍数

$b_6b_5b_4 \backslash b_3b_2b_1b_0$	000	001	010	011	100	101	110	111
0000	NUL	DLE	SP	0	@	P	`	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	\$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	.	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	]	l	
1101	CR	GS	-	=	M	\	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	_	o	DEL

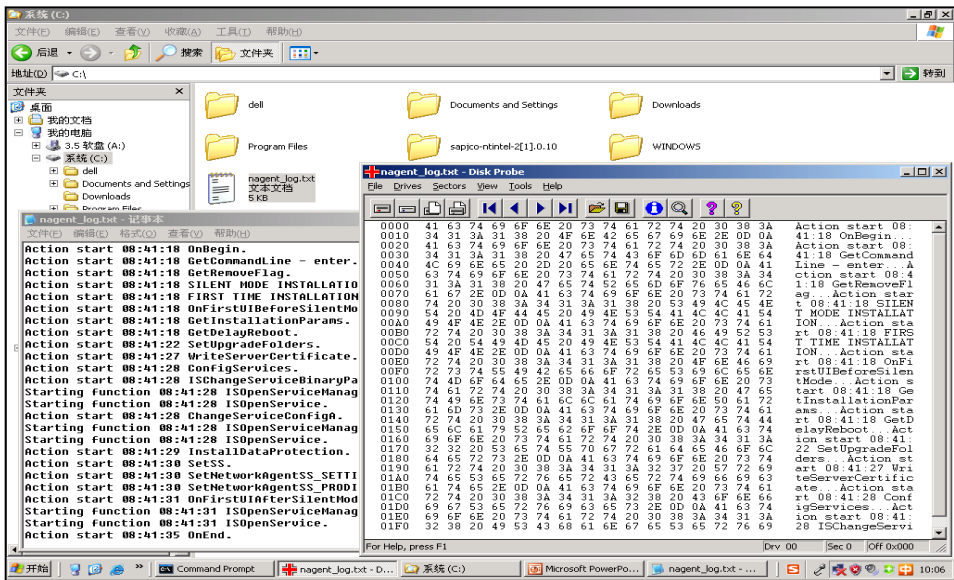
# 非数值性信息的表达：编码与组合

21

## (3) ASCII码存储的文件，即纯文本文件，是可以自动解读的

信息	机器内部存储	解析规则	十六进制显示存储内容
We are students	01010111 01100101 00100000 01100001 01110010 01100101 00100000 01110011 01110100 01110101 01100100 01100101 01101110 01110100 01110011	0/1串按8位分隔一个 字符，查找ASCII码 表映射成相应符号	57 65 20 61 72 65 20 73 74 75 64 65 6E 74 73

$b_7b_6b_5b_4$ $b_3b_2b_1b_0$	000	001	010	011	100	101	110	111
0000	NUL	DLE	SP	0	@	P	`	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	\$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	.	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	]	l	
1101	CR	GS	-	=	M	^	m	~
1110	SO	RS	>	>	N	_	n	~
1111	SI	US	/	?	O	-	o	DEL



# 非数值性信息的表达：编码与组合

22

## (4) 汉字的处理过程：由外码、到机器内码、到字形码

用0和1编码汉字,每个汉字在计算机内部由 2个字节表示

$b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0$     $b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0$

汉字  
内码

国标码   0 0 1 1 0 1 0 0   0 1 1 1 0 1 1 1  
↓   ↓  
(机)内码   1 0 1 1 0 1 0 0   1 1 1 1 0 1 1 1

0000001100000000  
0000001100000000  
0000001100000000  
0000001100000100  
1111111111111111  
0000001100000000  
0000001100000000  
0000001100000000  
0000001100000000  
0000001100000000  
0000001110000000  
0000011001000000  
0000110000100000  
0001100000110000  
0001000000011000  
0010000000001110  
1100000000000100

“大”

da

1 0 1 1 0 1 0 0   1 1 1 1 0 1 1 1

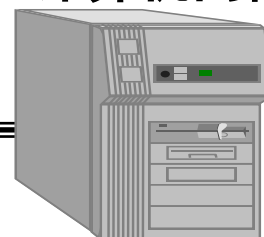
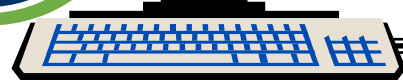
汉字外码  
/输入码

由外到内

计算机内部

由内到外

汉字字模  
点阵码



# 第3讲 符号化、计算化与自动化——看计算机的本质

23

- 一、符号化、计算化与自动化——计算机的本质
- 二、数值性信息的表达：计数制与机器数
- 三、非数值性信息的表达：编码与组合
- 四、0和1的计算：基本逻辑运算
- 五、用电子技术实现0和1及其逻辑运算
- 六、符号化-计算化的综合应用：利用图像隐藏信息

# 计算机是怎样完成计算的？

24

示例

$0+0 = 0$   
 $0+1 = 1$   
 $1+0 = 1$   
 $1+1 = 10$

计算机是怎样完成加法运算的，你知道吗？

1+1=2  
2+1=3   2+2=4  
3+1=4   3+2=5   3+3=6  
4+1=5   4+2=6   4+3=7   4+4=8  
5+1=6   5+2=7   5+3=8   5+4=9   5+5=10  
6+1=7   6+2=8   6+3=9   6+4=10   6+5=11   6+6=12  
7+1=8   7+2=9   7+3=10   7+4=11   7+5=12   7+6=13   7+7=14  
8+1=9   8+2=10   8+3=11   8+4=12   8+5=13   8+6=14   8+7=15   8+8=16  
9+1=10   9+2=11   9+3=12   9+4=13   9+5=14   9+6=15   9+7=16   9+8=17   9+9=18



# 0和1的计算：基本逻辑运算

25

## (1) 什么是逻辑？

逻辑是指事物因果之间所遵循的规律，是现实中普适的思维方式

- ◆ 逻辑的基本表现形式是命题与推理，推理即依据由简单命题的判断推导得出复杂命题的判断结论的过程。命题由语句表述，即内容为“真”或为“假”的一个判断语句！

**例如** 在一次中学生测验中，有三位老师做了预测：A.有人及格；B.有人不及格；C.全班都不及格。在考试后证明只有一个老师的预测是对的，请问谁对谁错？

求解过程：

命题A：“有人及格”；

命题B：“有人不及格”；

命题C：“全班都不及格”；

由题目假设和命题之间关系得出“已知”：A、B、C只有一个为真

如果A真，则C假；如果C真，则A假；

如果B真，而A、C可能有一个为真，与题矛盾，所以B为假。

如果B假，则“全班都及格”为真，而由此推断C为假。

由上“已知”，推理：A为真。

# 0和1的计算：基本逻辑运算

26

## (2) 基本逻辑运算

【命题】一个命题由语句表述，即内容为“真”或为“假”的一个判断语句！

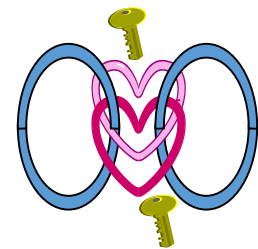
【命题的运算】如果命题由 $X$ ， $Y$ ， $Z$ 等表示，其值可能为“真”或为“假”，则两个命题 $X$ ， $Y$ 之间是可以进行运算的：

【与】运算(AND)：当 $X$ 和 $Y$ 都为真时， $X \text{ AND } Y$ 也为真；其他情况， $X \text{ AND } Y$ 均为假。

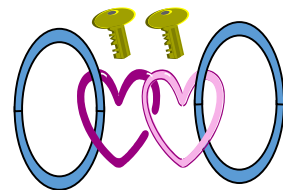
【或】运算(OR)：当 $X$ 和 $Y$ 都为假时， $X \text{ OR } Y$ 也为假；其他情况， $X \text{ OR } Y$ 均为真。

【非】运算(NOT)：当 $X$ 为真时， $\text{NOT } X$ 为假；当 $X$ 为假时， $\text{NOT } X$ 为真。

【异或】运算(XOR)：当 $X$ 和 $Y$ 都为真或都为假时， $X \text{ XOR } Y$ 为假；否则， $X \text{ XOR } Y$ 为真。



“与”运算：两把钥匙都有才能开门



“或”运算：只要有任意一把钥匙便能开门

# 0和1的计算：基本逻辑运算

27

## (3) 命题逻辑的符号化与计算

用0和1来表示逻辑运算

注：1表示 真，0表示 假

■ **【AND】**：“与”运算

有0为0，全1为1

■ **【OR】**：“或”运算

有1为1，全0为0

■ **【NOT】**：“非”运算

非0则1，非1则0

■ **【XOR】**：“异或”运算

相同为0，不同为1

<div>0 AND 0 0</div>	<div>0 AND 1 0</div>	<div>1 AND 0 0</div>	<div>1 AND 1 1</div>
<div>0 OR 0 0</div>	<div>0 OR 1 1</div>	<div>1 OR 0 1</div>	<div>1 OR 1 1</div>
<div>NOT 0 1</div>		<div>NOT 1 0</div>	
<div>0 XOR 0 0</div>	<div>0 XOR 1 1</div>	<div>1 XOR 0 1</div>	<div>1 XOR 1 0</div>

# 0和1的计算：基本逻辑运算

28

## (4) 怎样符号化逻辑并进行计算？

将逻辑表达为0和1及其运算

命题A: “有人及格”

命题B: “有人不及格”

命题C: “全班都不及格”

已知:  $((A \text{ AND } (\text{NOT } C)) \text{ OR } ((\text{NOT } A) \text{ AND } C)) = 1$

$(\text{NOT } B) \text{ AND } ((A \text{ AND } (\text{NOT } C)) \text{ OR } ((\text{NOT } A) \text{ AND } C))) = 1$

$(\text{NOT } B) \text{ AND } (\text{NOT } C) = 1$

组合形成所有可能解

$\{ \langle A=1, B=0, C=0 \rangle, \langle A=0, B=1, C=0 \rangle, \langle A=0, B=0, C=1 \rangle \}$

将上述可能解代入已知条件，使所有已知条件都满足的便是问题的解：

$\langle A=1, B=0, C=0 \rangle$ 。

1 ---真      0 ---假

一个命题用A、B等符号表达，其中符号的值可能为0，也可能为1

# 0和1的计算：基本逻辑运算

29

## (5) 二进制加法运算可用逻辑计算来实现

### 二进制的加法运算规则

$$\begin{array}{r} 0 \\ + 0 \\ \hline 0 \end{array} \quad \begin{array}{r} 1 \\ + 0 \\ \hline 1 \end{array} \quad \begin{array}{r} 0 \\ + 1 \\ \hline 1 \end{array} \quad \begin{array}{r} 1 \\ + 1 \\ \hline 0 \end{array}$$

不考虑进位

$$\begin{cases} S_i = A_i \text{ XOR } B_i \\ C_{i+1} = A_i \text{ AND } B_i \end{cases}$$

考虑进位

$$\begin{cases} S_i = (A_i \text{ XOR } B_i) \text{ XOR } C_i \\ C_{i+1} = ((A_i \text{ XOR } B_i) \text{ AND } C_i) \text{ OR } (A_i \text{ AND } B_i) \end{cases}$$

$$\begin{array}{r} A_i \\ B_i \\ + \\ \hline C_{i+1} \quad S_i \end{array}$$

$$\begin{array}{r} A_i \\ B_i \\ C_i \\ + \\ \hline C_{i+1} \quad S_i \end{array}$$

启示

乘除可转换为加减, 减法又可转换为加法, 而加法又可以用逻辑运算的组合来实现。机器只要能完成加法运算, 再由一个可执行组合的机构, 就可以实现任何运算。

# 用基本逻辑运算实现复杂计算

30

【枚举-计算-验证】证明复杂逻辑运算的正确性

考虑进位  $\begin{cases} S_i = (A_i \text{ XOR } B_i) \text{ XOR } C_i \\ C_{i+1} = ((A_i \text{ XOR } B_i) \text{ AND } C_i) \text{ OR } (A_i \text{ AND } B_i) \end{cases}$

$$\begin{array}{r} A_i \\ B_i \\ C_i \\ + \\ \hline C_{i+1} \quad S_i \end{array}$$

$A_i$	$B_i$	$C_i$	加法规则的 $S_i$	逻辑运算规则的 $S_i$	加法规则的 $C_{i+1}$	逻辑运算规则的 $C_{i+1}$
0	0	0	0	0	0	0
0	0	1	1	1	0	0
0	1	0	1	1	0	0
0	1	1	0	0	1	1
1	0	0	1	1	0	0
1	0	1	0	0	1	1
1	1	0	0	0	1	1
1	1	1	1	1	1	1

# 0和1的计算：基本逻辑运算

31

## (6) 逻辑研究有哪些?

可深入学习

1) Aristotle (亚里士多德)(公元前384—322)。古希腊哲学家：[形式逻辑](#)。

典型概念：命题，推理，三段论

2) Leibnitz (莱布尼茨)(1646—1716)。德国数学家：[数理逻辑](#)。

典型概念：谓词，谓词演算

3) Boole (布尔)(1815—1864)。英国数学家，[布尔代数](#)

典型概念：布尔量，布尔值，布尔运算，布尔操作

4) 其他：[时序逻辑](#)(Temporal Logics)、[模态逻辑](#)(Modal Logics)、[归纳逻辑](#)(Inductive Logics)、[模糊逻辑](#)(Fuzzy Logics)、[粗糙逻辑](#)(Rough Logics)、[非单调逻辑](#)等

# 第3讲 符号化、计算化与自动化——看计算机的本质

32

- 一、符号化、计算化与自动化——计算机的本质
- 二、数值性信息的表达：计数制与机器数
- 三、非数值性信息的表达：编码与组合
- 四、0和1的计算：基本逻辑运算
- 五、用电子技术实现0和1及其逻辑运算
- 六、符号化-计算化的综合应用：利用图像隐藏信息



# 0和1与电子技术实现-(1) 如何用电信号及电子元件表达0和1?

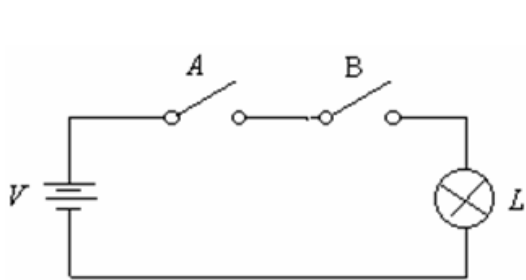
33

实现**0**和**1**的基本元器件：电信号和继电器开关

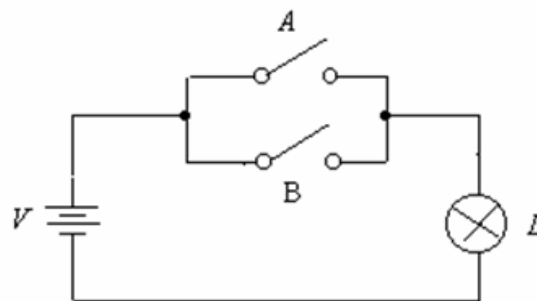


数字信号：高电平为1，低电平为0

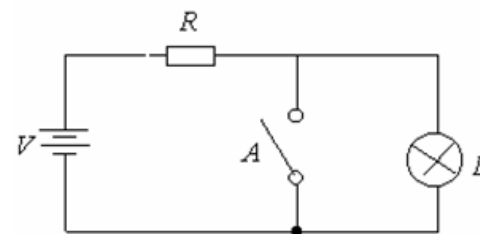
◆用继电器开关实现基本逻辑运算



“与”运算电路



“或”运算电路



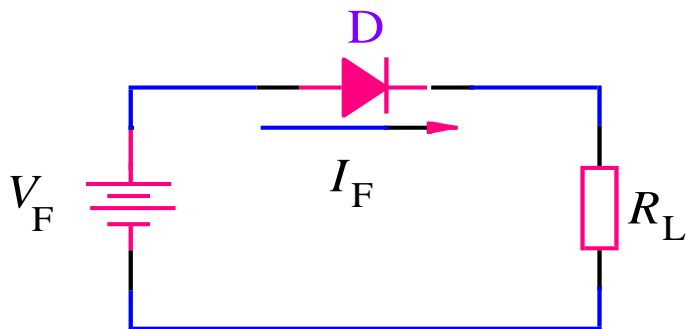
“非”运算电路

# 0和1与电子技术实现-(2) 处理0和1的基本元件？

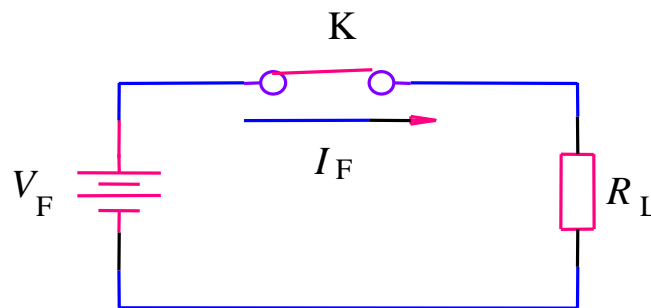
34

## 实现0和1的基本元器件：二极管

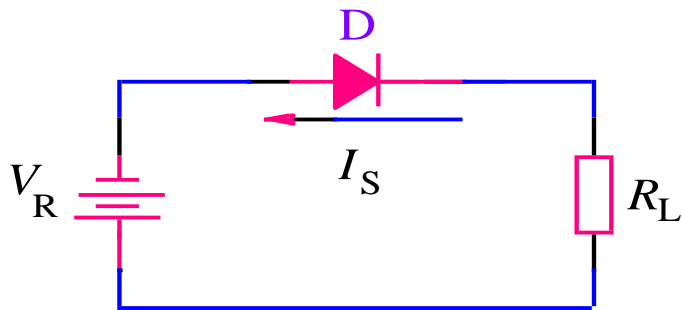
### ◆二极管的基本特性



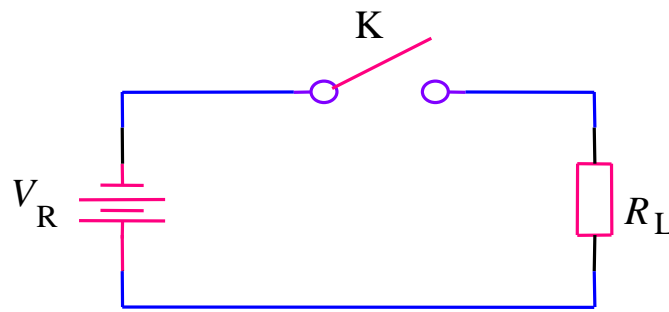
(a)



(b)



(a)



(b)

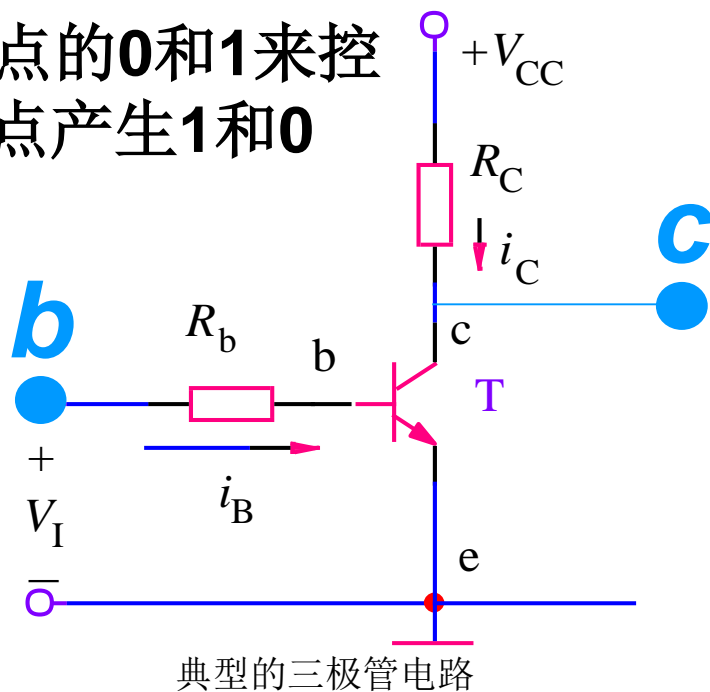
# 0和1与电子技术实现-(2) 处理0和1的基本元件？

## 实现0和1的基本元器件：三极管

### ◆ 三极管的基本特性：

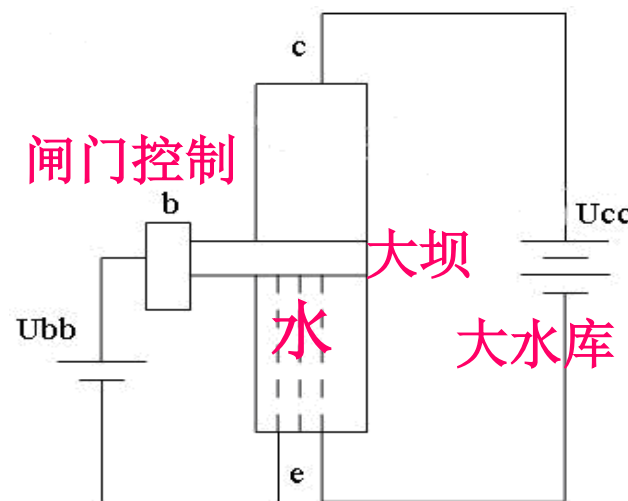
- 开关和放大
- 以较小的b极电流信号可控制较大的e极流过的电流--放大。

用b点的0和1来控制c点产生1和0



"The first transistor ever assembled, invented in Bell Labs in 1947." Photo and text from Porticus.org, www.porticus.org/bell/belllabs\_transistor.html. (Follow that link to see more historical documents and images about Bell Labs and the transistor.)

第一个三极管试验装置



# 用电子技术实现0和1及其逻辑运算

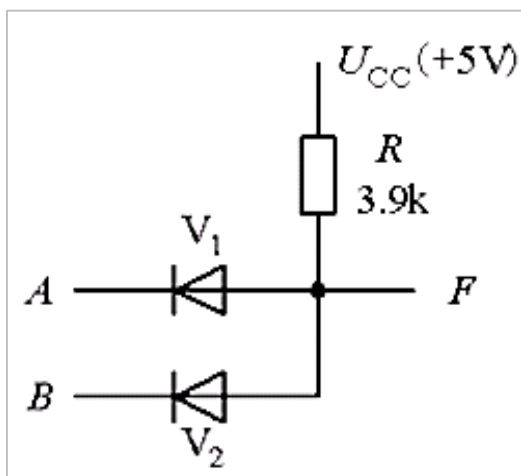
36

## (1) 基本逻辑运算的电子实现：逻辑门

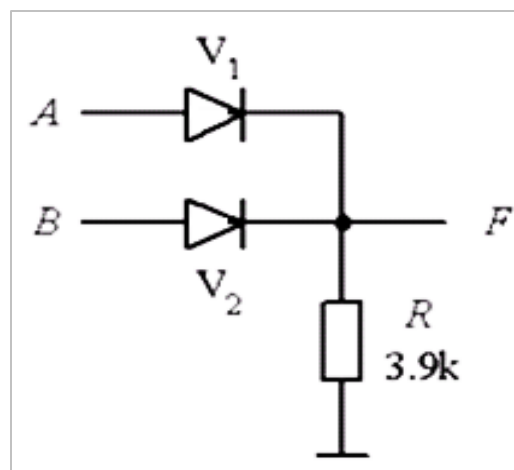
电信号：0V（低电平） 和 5V（高电平）



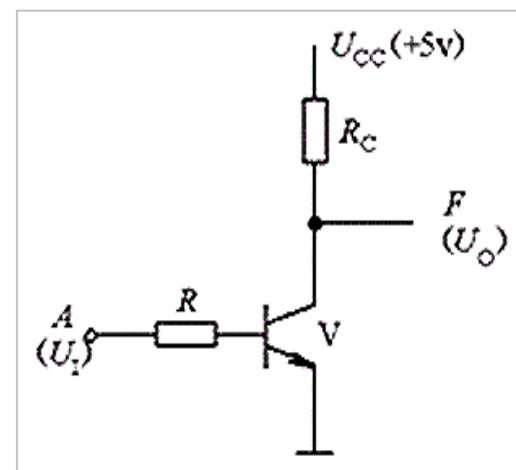
用二极管、三极管实现基本逻辑运算的电路



**$F = A \text{ and } B$**   
**【与门】电路**



**$F = A \text{ or } B$**   
**【或门】电路**



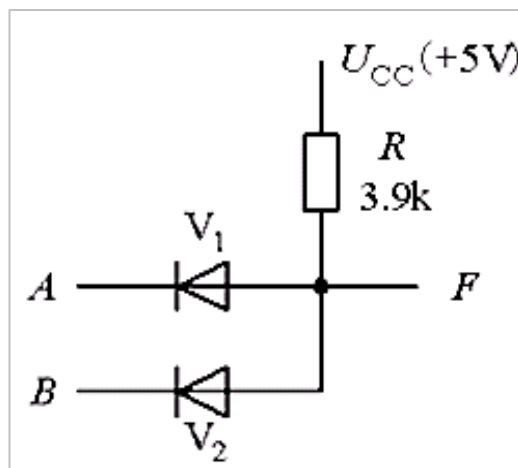
**$F = \text{not } A$**   
**【非门】电路**

这些电路被封装成集成电路(芯片)，即所谓的逻辑门，【与门】、【或门】和【非门】。

# 用电子技术实现0和1及其逻辑运算

37

## (2) 集成电路示意

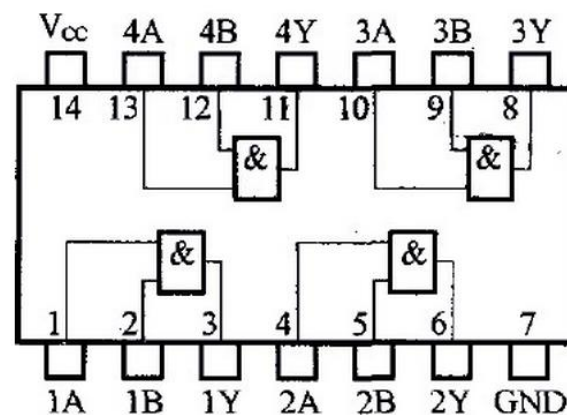


$$F = A \text{ and } B$$

【与门】电路



【与门】电路符号  
左侧是输入，右侧是输出

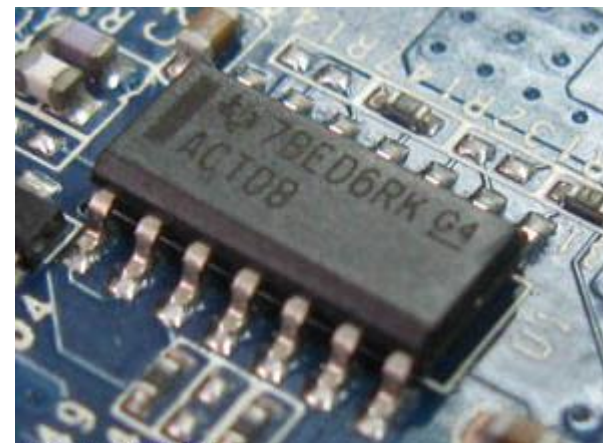
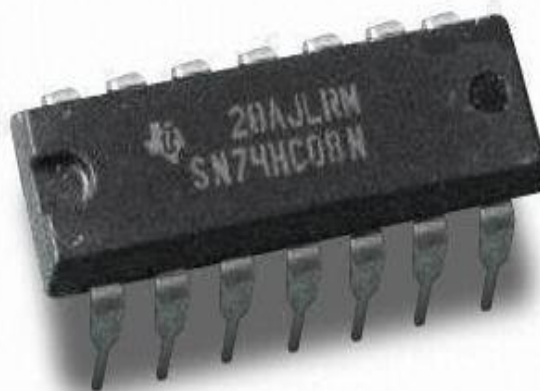


$$1Y = 1A \text{ and } 1B$$

$$2Y = 2A \text{ and } 2B$$

$$3Y = 3A \text{ and } 3B$$

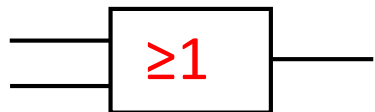
$$4Y = 4A \text{ and } 4B$$



# 用电子技术实现0和1及其逻辑运算

38

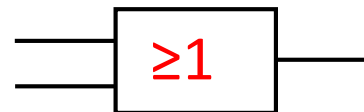
## (3) 逻辑门的符号表示



注意：左侧是输入，右侧是输出  
输入是电信号的0或者1，  
输出是对输入做相应的运算，也  
是电信号的0或者1



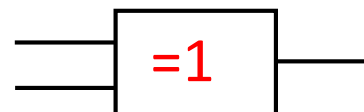
【与门】电路符号



【或门】电路符号



【非门】电路符号



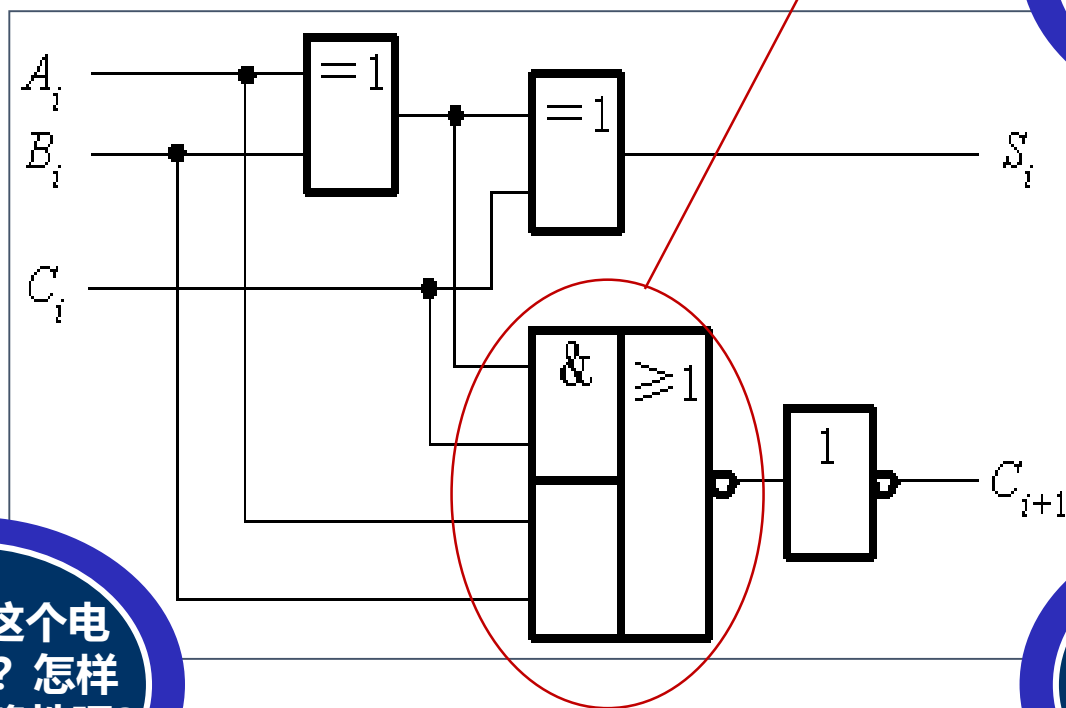
【异或门】电路符号

# 加法运算是这样实现的

39

二进制加法运算用逻辑门电路的组合来实现

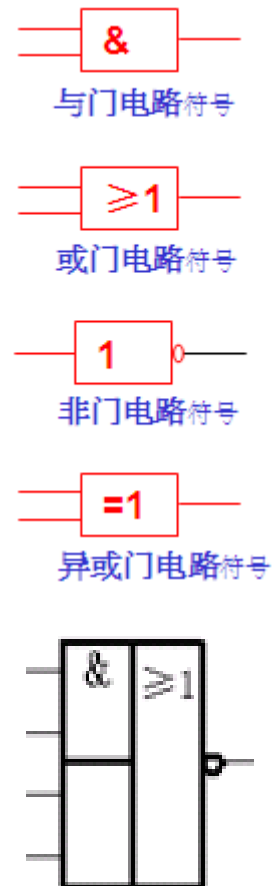
## 一位加法器的示例



疑问1：这个电路符号是什么？

疑问3：这个电路正确吗？怎样验证其正确性呢？

疑问2：为什么前面1个非门后面又有一个非门，有意义吗？

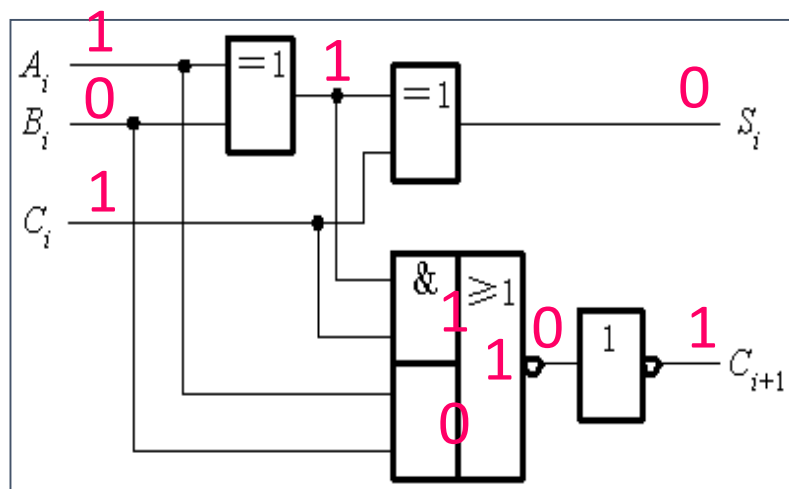


# 用电子技术实现0和1及其逻辑运算

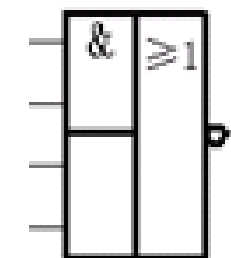
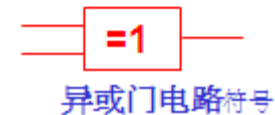
40

## (4) 二进制加法运算用逻辑门电路的组合来实现

通过枚举输入，可验证一位加法器实现的正确性



$$\begin{array}{r} A_i \\ B_i \\ C_i \\ + \\ \hline C_{i+1} \quad S_i \end{array}$$





# 用电子技术实现0和1及其逻辑运算

41

## (4) 二进制加法运算用逻辑门电路的组合来实现

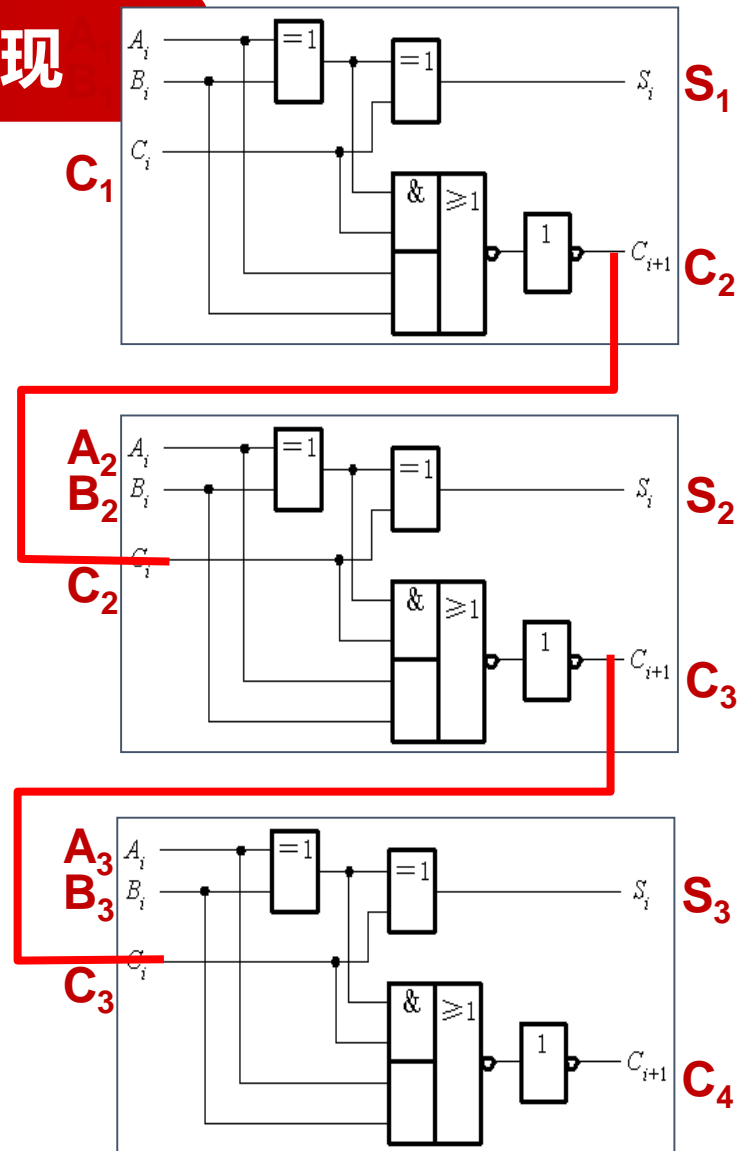
### 多位加法器的实现

- 用已验证正确的一位加法器，来实现更为复杂的多位加法器
- 用已验证正确的多位加法器，来实现更为复杂的乘法器/除法等(略)

【分层构造】低层电路已验证正确，可被封装起来；用已封装的已验证的低层电路可构造更为复杂的高层电路；如此一层层构造。

$$\begin{array}{r} \phantom{+} \phantom{A_3} \phantom{A_2} \phantom{A_1} \\ + \phantom{A_3} \phantom{A_2} \phantom{A_1} \\ \hline \phantom{+} \phantom{A_3} \phantom{A_2} \phantom{A_1} \end{array} \begin{array}{r} A_3 \phantom{A_2} \phantom{A_1} \\ B_3 \phantom{B_2} \phantom{B_1} \\ \hline C_4 \phantom{C_3} \phantom{C_2} \phantom{C_1} \\ S_3 \phantom{S_2} \phantom{S_1} \end{array}$$

$$\begin{array}{r} \phantom{+} \phantom{0} \phantom{0} \phantom{1} \\ + \phantom{0} \phantom{0} \phantom{1} \phantom{1} \\ \hline \phantom{+} \phantom{0} \phantom{0} \phantom{1} \end{array} \begin{array}{r} 0 \phantom{0} \phantom{1} \\ 0 \phantom{1} \phantom{1} \\ \hline 1 \phantom{0} \phantom{0} \end{array}$$



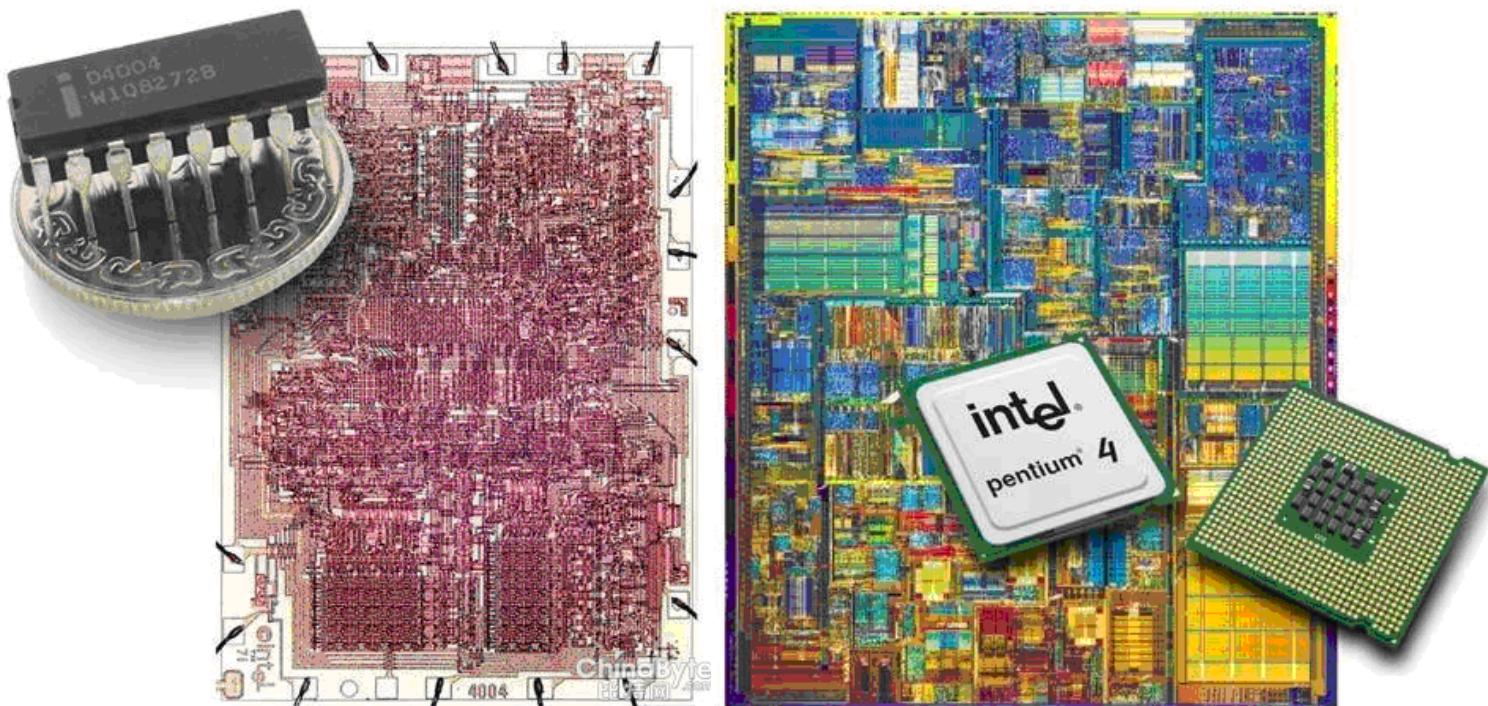
# 用电子技术实现0和1及其逻辑运算

42

## (7) 集成电路是这样的...

### 复杂的硬件部件：(芯片、主板)

◆ **微处理器芯片**即是复杂组合逻辑集成在一块板上并封装而成的电路：  
从Intel 4004在**12平方毫米**的芯片上集成了**2250颗**晶体管→到Pentium 4处理器内建了**4200万颗**晶体管，以及采用**0.18微米**的电路→再到英特尔的**45纳米**Core 2至尊/至强四核处理器上装载了**8.2亿颗**晶体管。

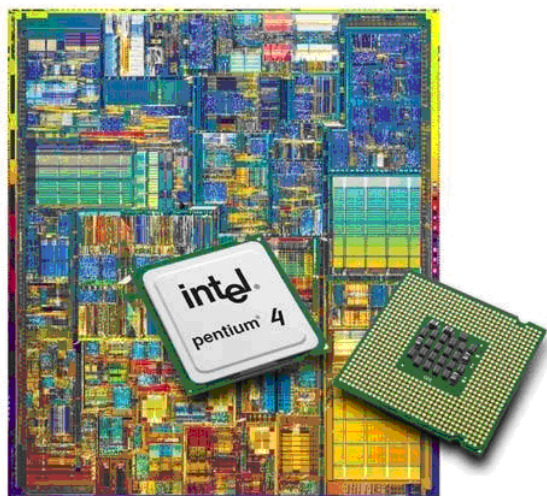
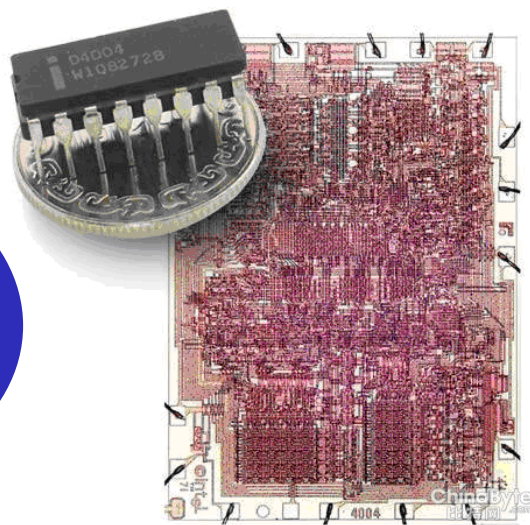


为什么需要如此多的晶体管呢？

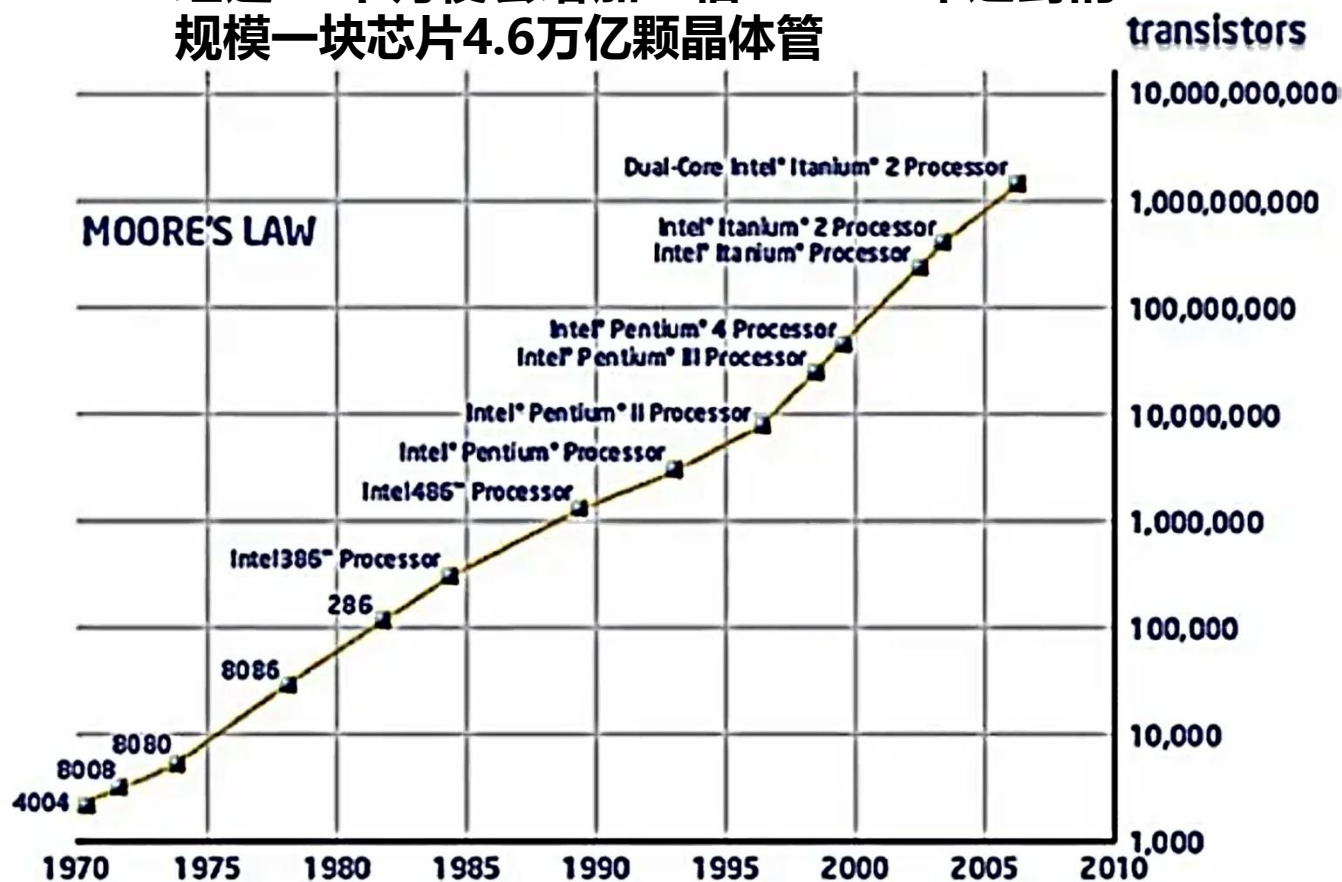
# 集成电路是这样复杂起来的

集成电路是这样的...

疑问：为什么需要这么多晶体管？



摩尔定律：芯片可容纳的晶体管数目大约每经过18个月便会增加一倍：2021年达到的规模一块芯片4.6万亿颗晶体管



# 用电子技术实现0和1及其逻辑运算

44

## (8) 练习题

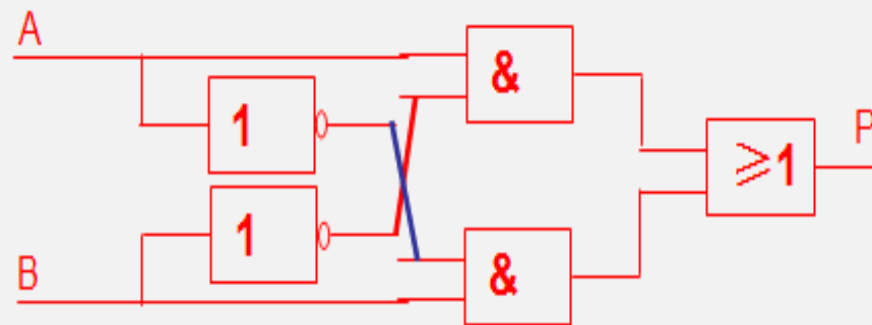
假设基本门电路的符号为



的正确的逻辑运算为\_\_\_\_\_。

- (A)  $P = (A \text{ AND } (\text{NOT } B)) \text{ AND } ((\text{NOT } A) \text{ OR } B) ;$
- (B)  $P = A \text{ XOR } B ;$
- (C)  $P = \text{NOT } (A \text{ AND } B) \text{ AND } (A \text{ AND } B) ;$
- (D)  $P = (A \text{ OR } B) \text{ AND } (A \text{ AND } (\text{NOT } B)) ;$

，已知电路如右图示意。问该电路所实现

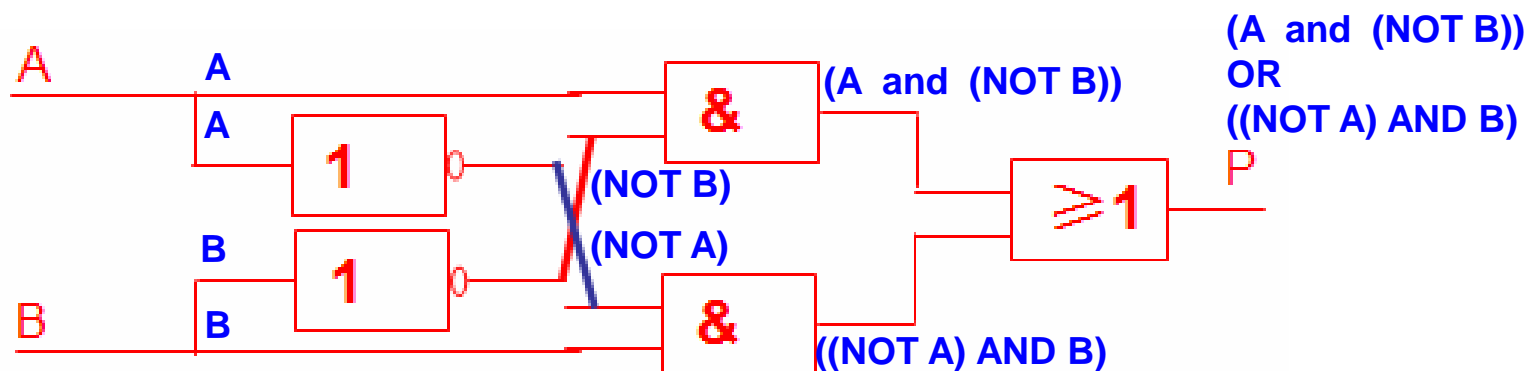




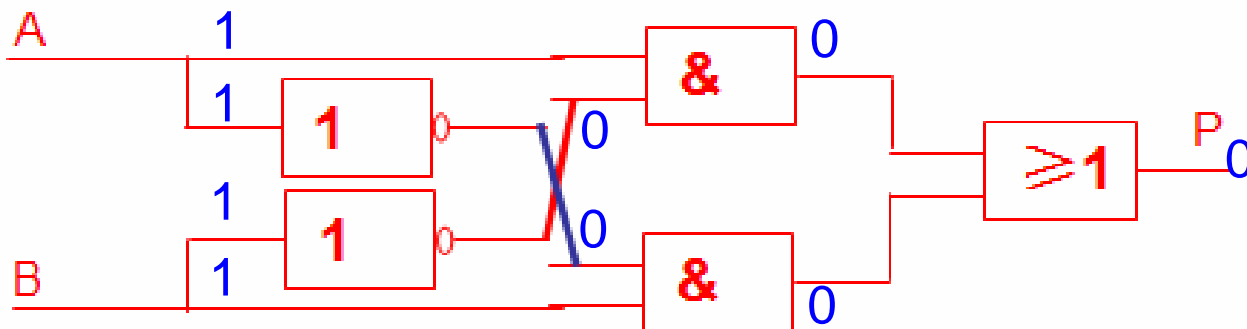
# 用电子技术实现0和1及其逻辑运算

45

## (8) 练习题的求解思维



A	B	A XOR B	(A AND (NOT B)) OR ((NOT A) and B)
0	0	0	0
0	1	1	1
1	0	1	1
1	1	0	0



运用计算思维  
求解：枚举-  
计算-验证

# 第3讲 符号化、计算化与自动化——看计算机的本质

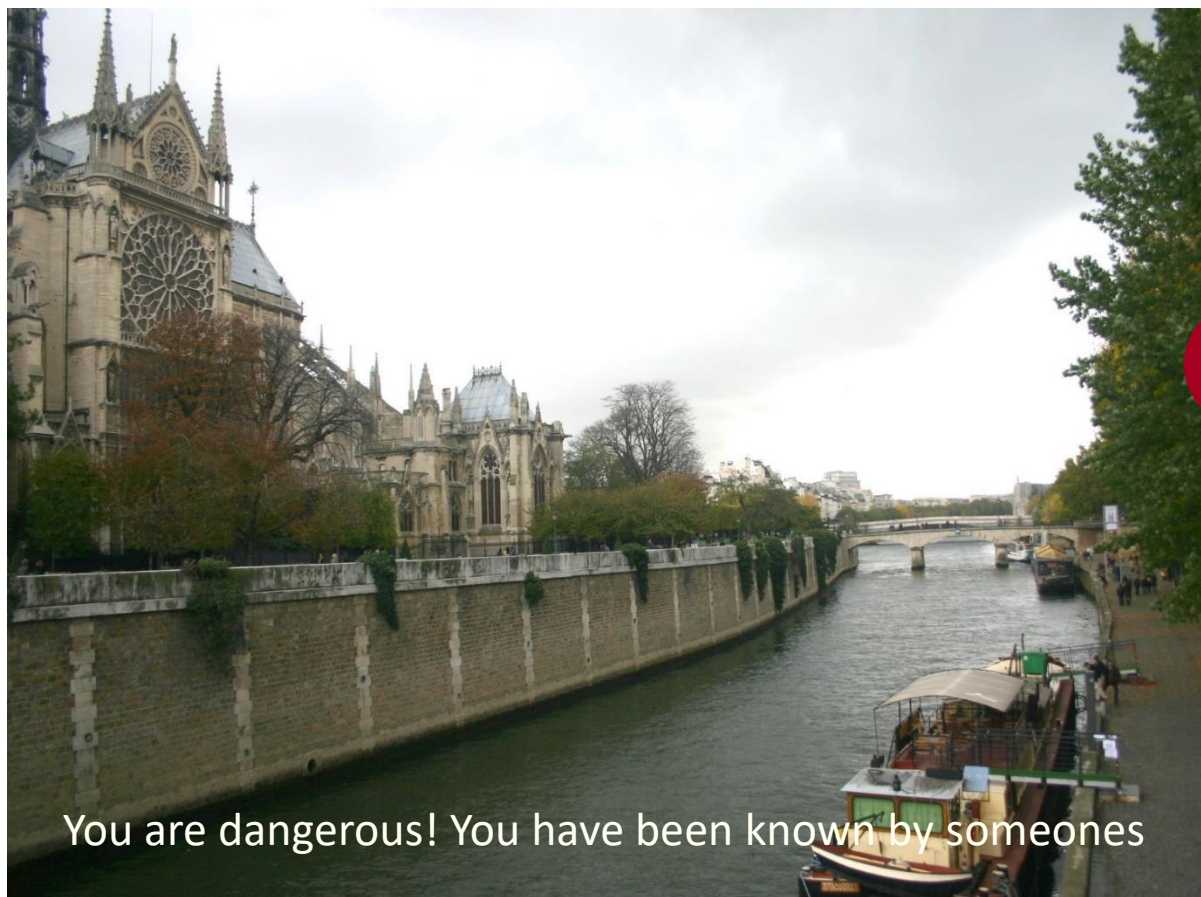
46

- 一、符号化、计算化与自动化——计算机的本质
- 二、数值性信息的表达：计数制与机器数
- 三、非数值性信息的表达：编码与组合
- 四、0和1的计算：基本逻辑运算
- 五、用电子技术实现0和1及其逻辑运算
- 六、符号化-计算化的综合应用：利用图像隐藏信息

# 符号化-计算化的综合应用：利用图像隐藏信息

47

(1) 这幅图像背后其实是隐藏着信息的, 你相信吗?



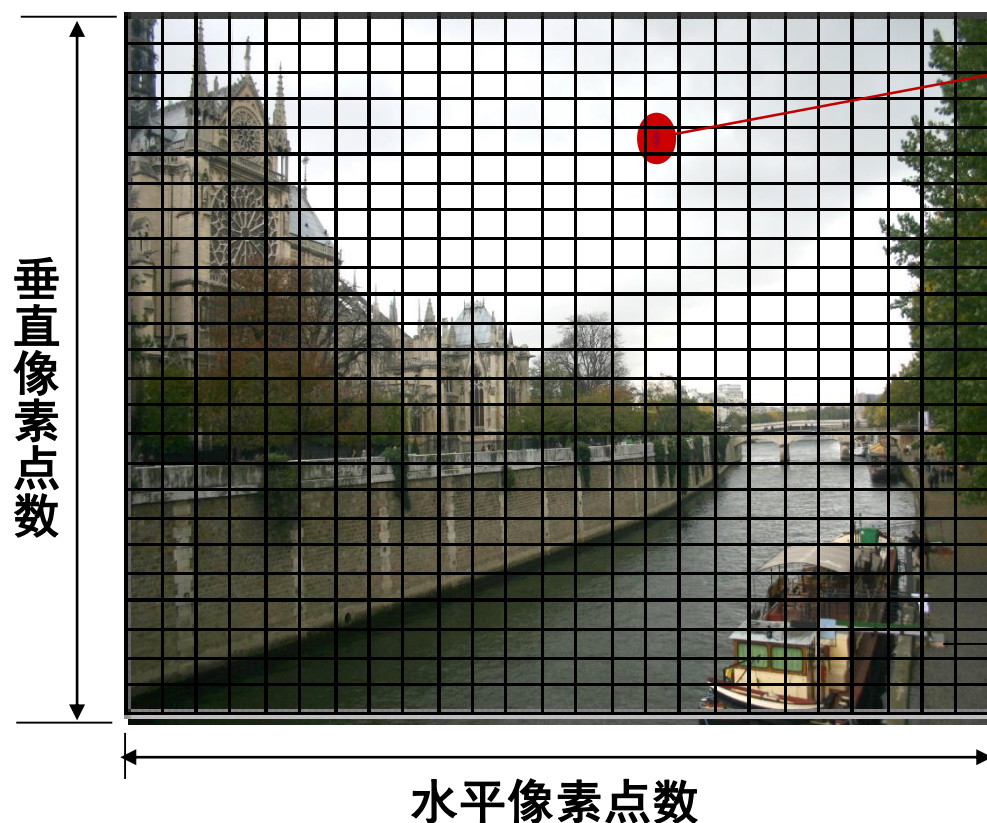
这是怎么做到的呢?



# 符号化-计算化的综合应用：利用图像隐藏信息

48

## (2) 图像是怎么表达的



像素点的位数

黑白-**1位**(0,1)

256级灰度-**8位**(0-255)

16色彩色-**4位**(0-15)

256色彩色-**8位**(0-255)

24位真彩色-**24位**

(红0-255、绿0-255、蓝0-255三原色)

0 --2种组合

0010 --16种组合

10010010 --256种组合

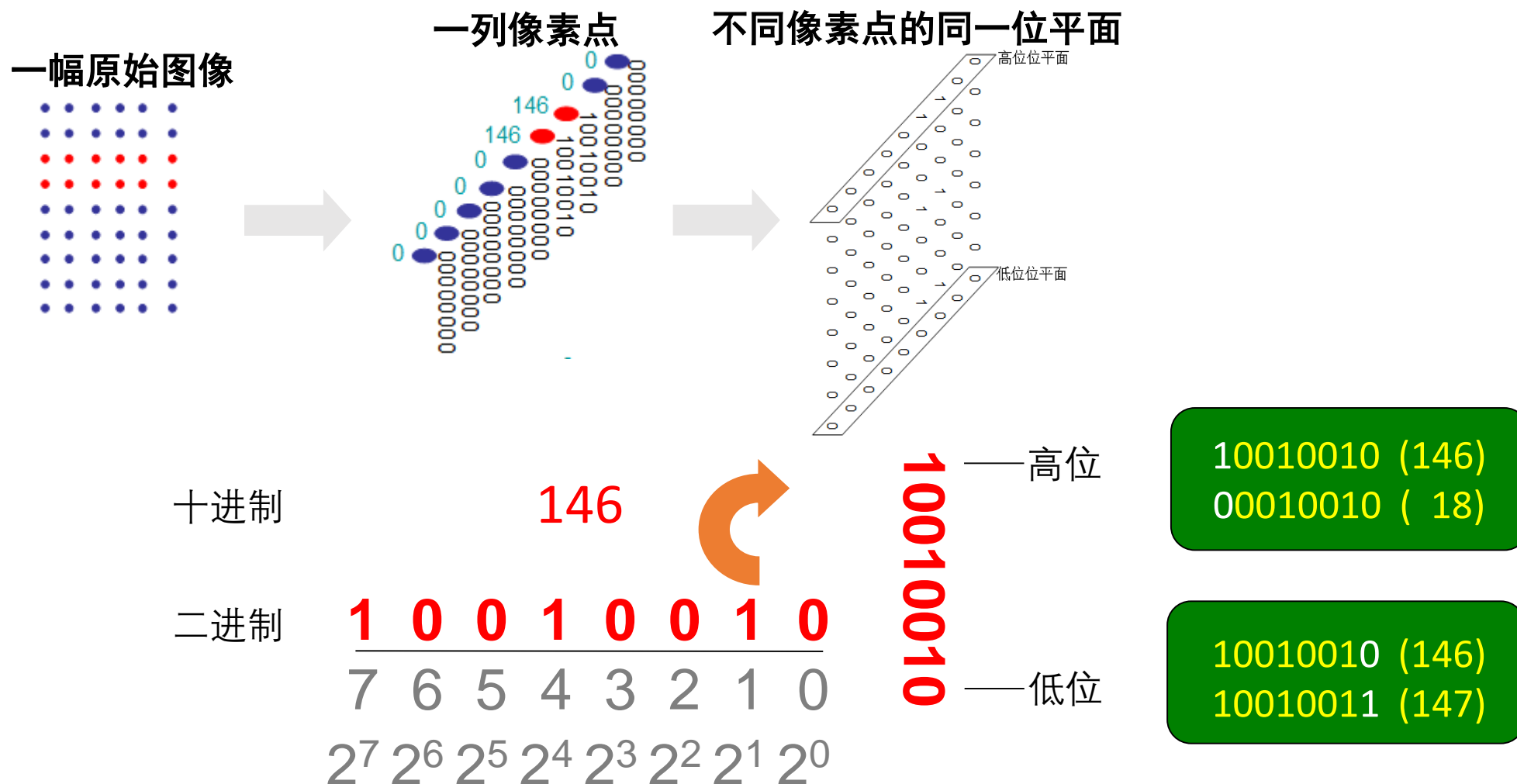
10010010 01010010 01010001 --256×256×256种组合



# 符号化-计算化的综合应用：利用图像隐藏信息

49

## (3) 图像的像素：高位和低位值的差别



#### (4) 待隐藏信息的01编码



## 怎样将一个二进制数的某一位设为0或1呢？

# 符号化-计算化的综合应用：利用图像隐藏信息

51

## (5) 判断信息位及设置信息位

将一个数的某一位设置为0

$$\begin{array}{r} 10001111 \\ \text{AND } 11111110 \\ \hline 10001110 \end{array}$$

$$\begin{array}{r} 10001111 \\ \text{AND } 11110111 \\ \hline 10000111 \end{array}$$

将一个数的某一位设置为1

$$\begin{array}{r} 10001110 \\ \text{OR } 00000001 \\ \hline 10001111 \end{array}$$

$$\begin{array}{r} 10000111 \\ \text{OR } 00001000 \\ \hline 10001111 \end{array}$$

判断一个数的某一位为0或1

$$\begin{array}{r} 10001110 \\ \text{AND } 00010000 \\ \hline 00000000 \end{array}$$

$$\begin{array}{r} 10001110 \\ \text{AND } 00001000 \\ \hline 00001000 \end{array}$$

**一幅原始图像**

**一列像素点**

**不同像素点的同一位平面**

**待隐藏的信息**

**待隐藏信息的01编码**

**将最低位平面替换为隐藏信息的编码**

IF 待隐藏信息的第  $i$  位为0 then  
 第  $i$  个像素的值 =  
 第  $i$  个像素的值 AND 11111110  
 ELSE  
 第  $i$  个像素的值 =  
 第  $i$  个像素的值 OR 00000001  
 ENDIF

0100 0001  
 AND 0000 1000  
 -----  
 =0, 则第  $i$  位为0  
 <>0, 则第  $i$  位为1

```
IF 待隐藏信息的第 i 位为 0 then
    第 i 个像素的值 =
        第 i 个像素的值 AND 11111110
ELSE
    第 i 个像素的值 =
        第 i 个像素的值 OR 00000001
ENDIF
```

待隐藏的信息

A

0100 0001

待隐藏信息的01编码

0100 0001

AND 0000 1000

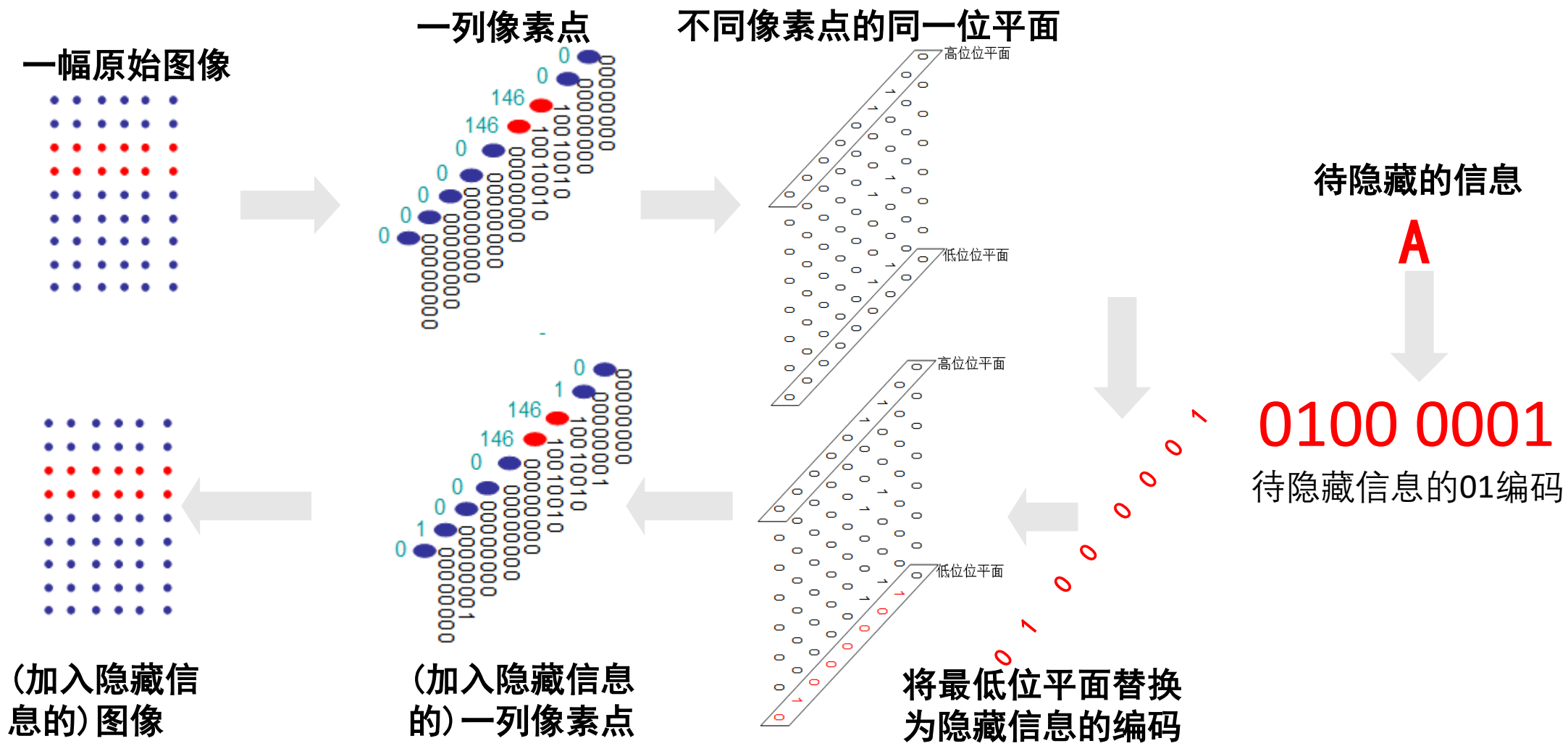
---

=0, 则第i位为0

<>0, 则第i位为1

# 符号化-计算化的综合应用：利用图像隐藏信息

## (7) 能否看出隐藏了信息呢?



# 符号化-计算化的综合应用：利用图像隐藏信息

54

(8) 这幅图像背后其实是隐藏着信息的, 你相信吗?



是这样做的  
呀?

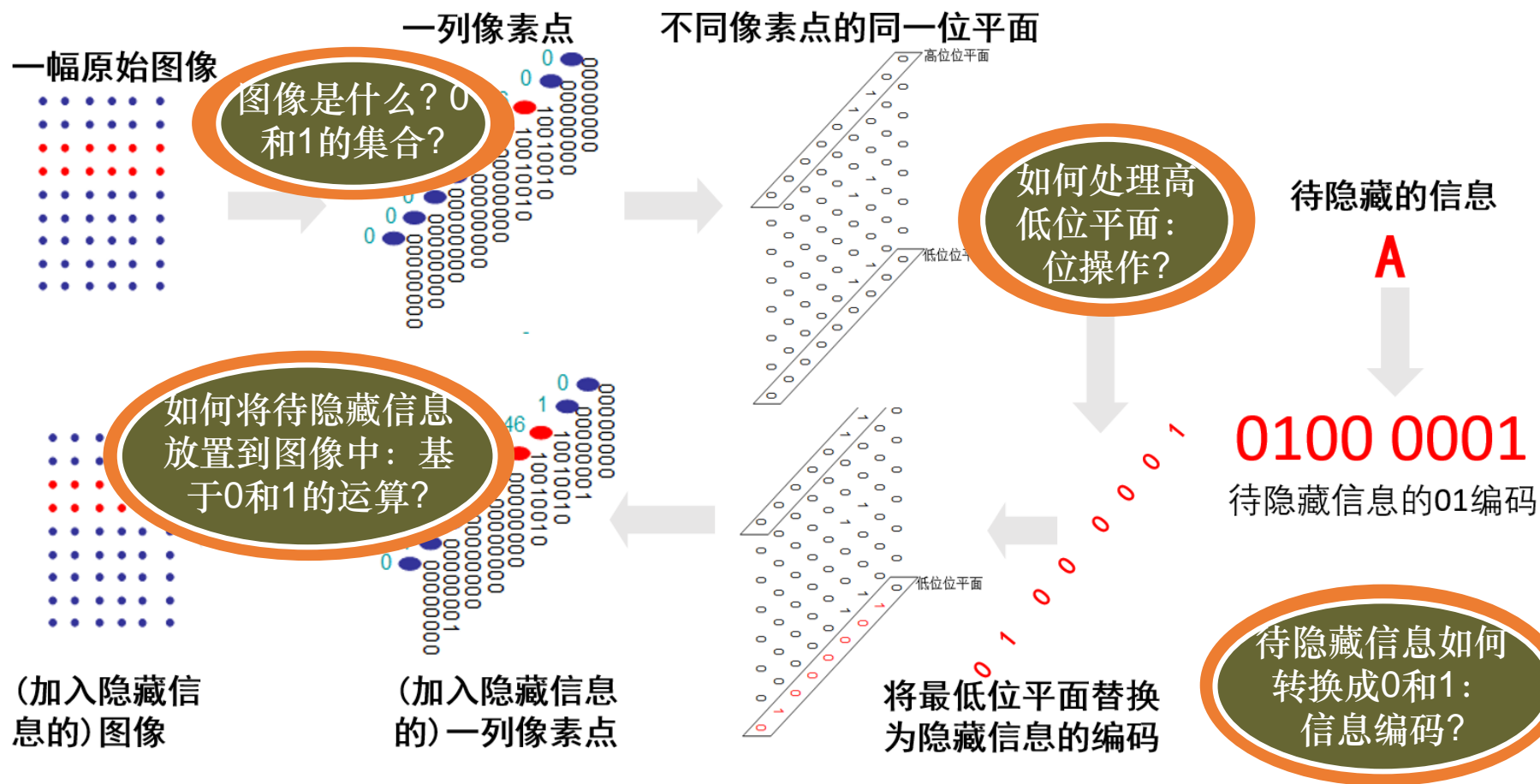




# 符号化-计算化的综合应用：利用图像隐藏信息

55

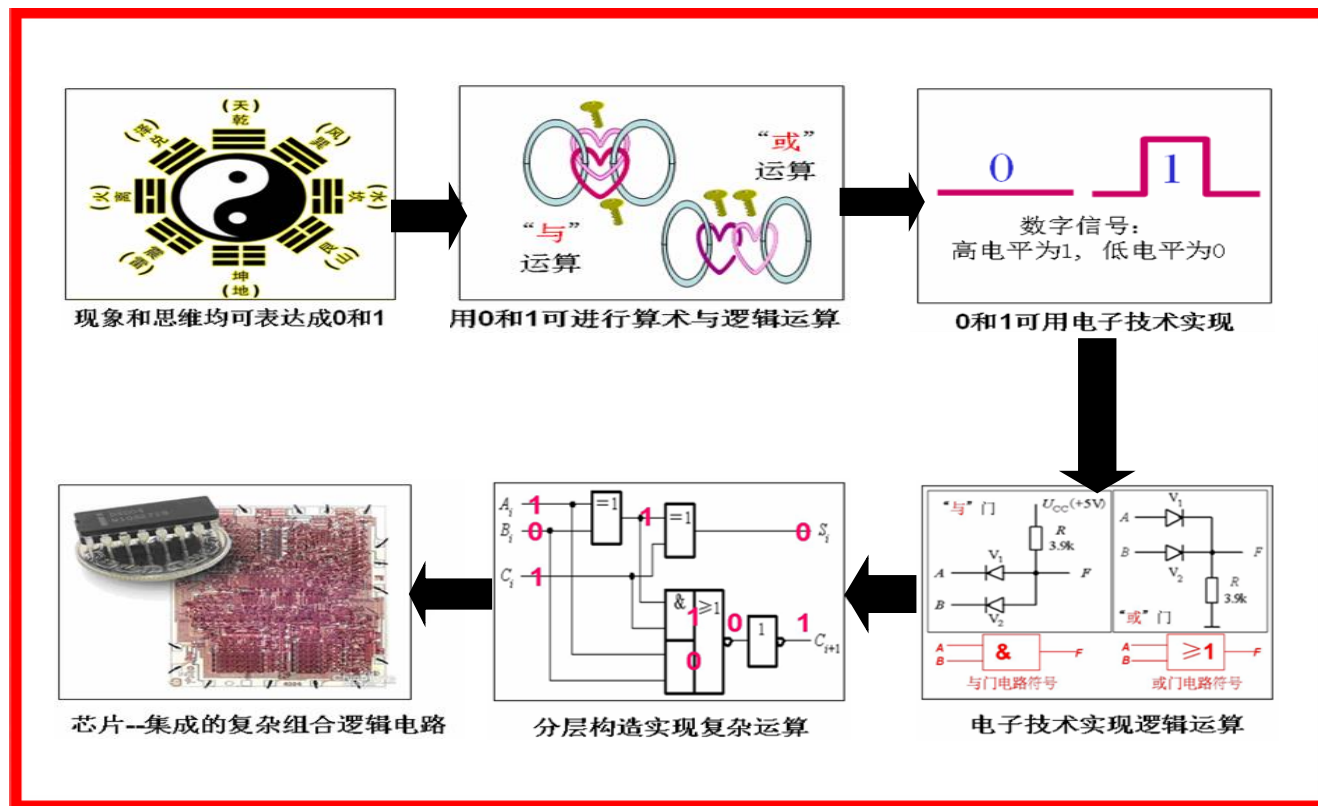
## (9) 图像隐藏信息：小结



# 符号化、计算化与自动化：0和1思维回顾

56

## 符号化、计算化与自动化：计算机的本质



语义符号化 → 符号计算化 → 计算0(和)1化 → 0(和) 1自动化 → 分层构造化 → 构造集成化