第3讲 从加法自动计算看机器硬件实现
—逻辑运算与逻辑门电路

战德臣

哈尔滨工业大学计算学部教学委员会主任
国家教学名师

18686783018, dechen@hit.edu.cn

计算机是怎样完成计算的?

示例

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

计算机器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

计算机的本质是逻辑

【逻辑】与基本逻辑运算

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】一个命题由语句表述，即内容为“真”或为“假”的一个判断语句!

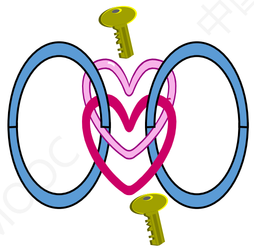
【命题的运算】如果命题由X, Y, Z等表示，其值可能为“真”或为“假”，则两个命题X, Y之间是可以进行运算的:

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

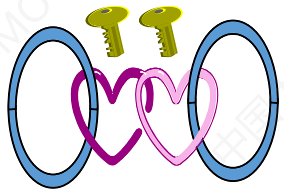
【或】运算(OR): 当X和Y都为假时, X OR Y也为假; 其他情况, X OR Y均为真。

【非】运算(NOT): 当X为真时, NOT X为假;当X为假时, NOT X为真。

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



“与”运算：两把钥匙都有才能开门



“或”运算：只要有任何一把钥匙便能开门

计算机的本质是逻辑

0和1的计算：基本逻辑运算

注: 1表示 **真**，0表示 **假**

用0和1来表示逻辑运算

- **【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

$$\begin{array}{r} 0 \\ \text{AND}\quad 0 \\ \hline 0 \end{array} \qquad \begin{array}{r} 0 \\ \text{AND}\quad 1 \\ \hline 0 \end{array} \qquad \begin{array}{r} 1 \\ \text{AND}\quad 0 \\ \hline 0 \end{array} \qquad \begin{array}{r} 1 \\ \text{AND}\quad 1 \\ \hline 1 \end{array}$$

$$\begin{array}{r} 0 \\ \text{OR}\quad 0 \\ \hline 0 \end{array} \qquad \begin{array}{r} 0 \\ \text{OR}\quad 1 \\ \hline 1 \end{array} \qquad \begin{array}{r} 1 \\ \text{OR}\quad 0 \\ \hline 1 \end{array} \qquad \begin{array}{r} 1 \\ \text{OR}\quad 1 \\ \hline 1 \end{array}$$

$$\begin{array}{r} \text{NOT}\quad 0 \\ \hline 1 \end{array} \qquad \begin{array}{r} \text{NOT}\quad 1 \\ \hline 0 \end{array}$$

$$\begin{array}{r} 0 \\ \text{XOR}\quad 0 \\ \hline 0 \end{array} \qquad \begin{array}{r} 0 \\ \text{XOR}\quad 1 \\ \hline 1 \end{array} \qquad \begin{array}{r} 1 \\ \text{XOR}\quad 0 \\ \hline 1 \end{array} \qquad \begin{array}{r} 1 \\ \text{XOR}\quad 1 \\ \hline 0 \end{array}$$

用基本逻辑运算实现复杂计算

【示例】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

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

$$\begin{array}{rr} & A_i \\ + & B_i \\ \hline C_{i+1} & S_i \end{array}$$

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

你能证明一下吗?

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

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

用基本逻辑运算实现复杂计算

【枚举-计算-验证】证明复杂逻辑运算的正确性

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

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

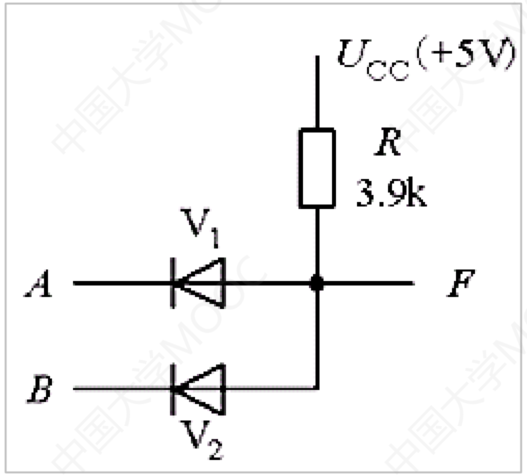
A_i	B_i	C_i	加法规则的S_i	逻辑运算规则的S_i	加法规则的C_{i+1}	逻辑运算规则的C_{i+1}
0	0	0	0	0	0	0
0	0	1	1	1	0	0
0	1	0	1	1	0	0
0	1	1	0	0	1	1
1	0	0	1	1	0	0
1	0	1	0	0	1	1
1	1	0	0	0	1	1
1	1	1	1	1	1	1

用电子技术实现基本逻辑运算

基本逻辑运算的电子实现：逻辑门

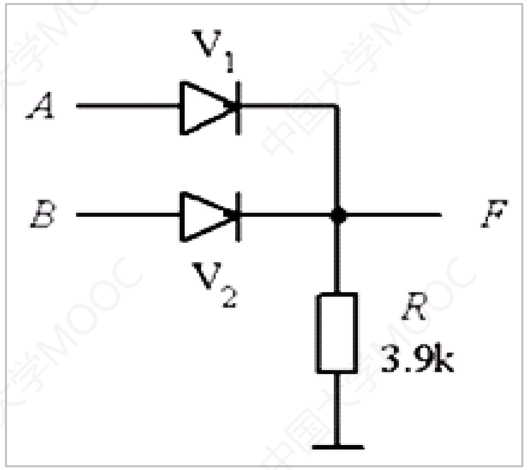
电信号：0V（低电平） 和 5V（高电平）

用二极管、三极管实现基本逻辑运算的电路



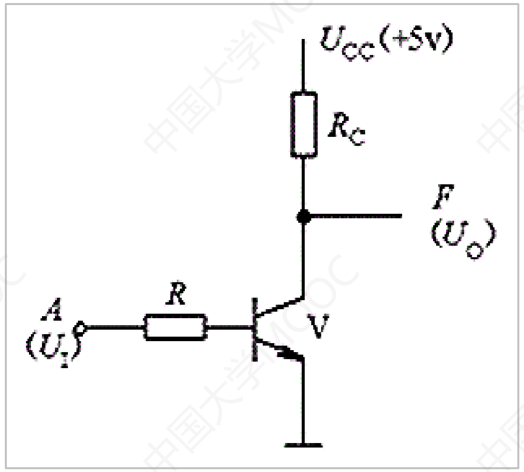
F = A AND B

【与门】电路



F = A OR B

【或门】电路



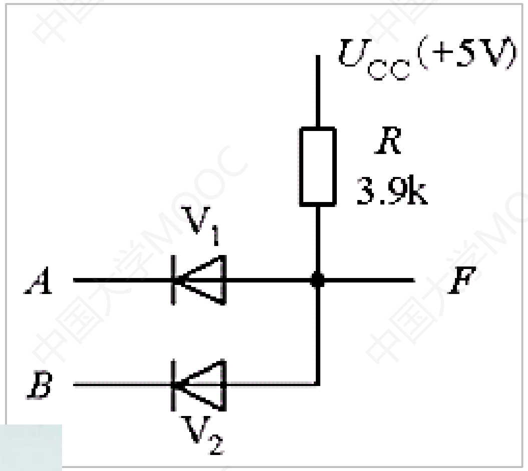
F = NOT A

【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】。

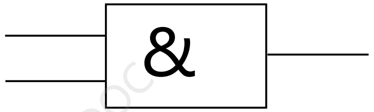
用电子技术实现基本逻辑运算

集成电路示意



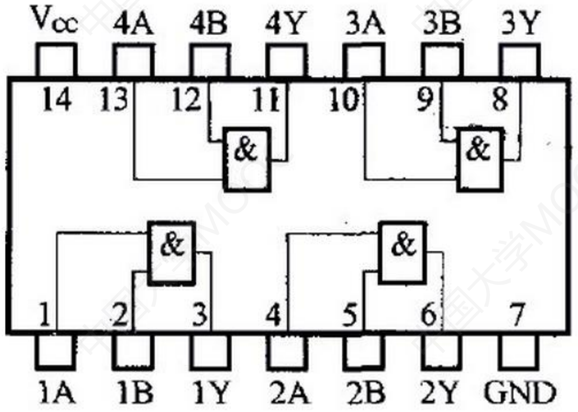
F = A AND B

【与门】电路



【与门】电路符号

左侧是输入，右侧是输出

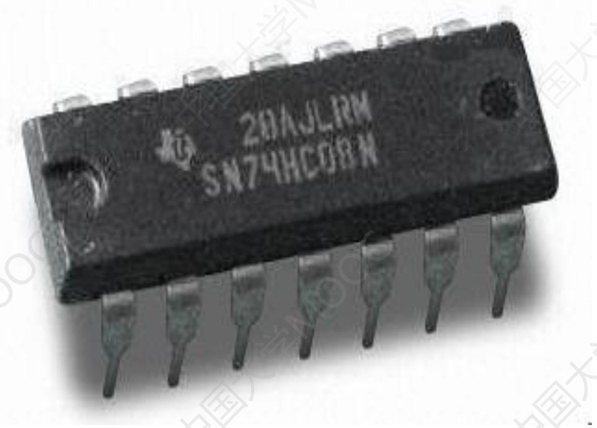


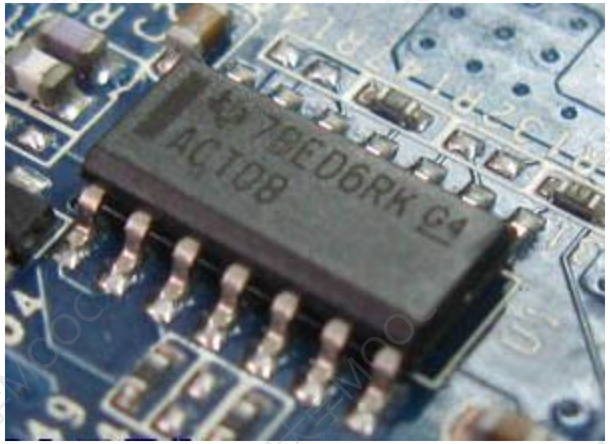
1Y = 1A AND 1B

2Y = 2A AND 2B

3Y = 3A AND 3B

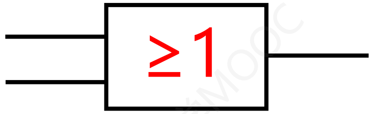
4Y = 4A AND 4B





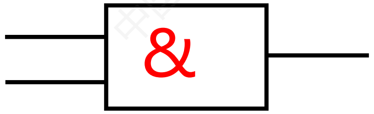
用电子技术实现基本逻辑运算

逻辑门的符号表示

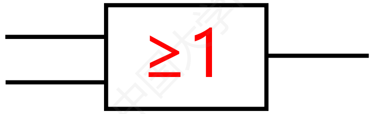


注意：**左侧是输入，右侧是输出**
输入是电信号的0或者1，
输出是对输入做相应的运算，也
是电信号的0或者1

注意门电路符号的左输入右输出

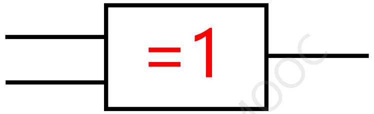


【与门】电路符号



【或门】电路符号

【非门】电路符号

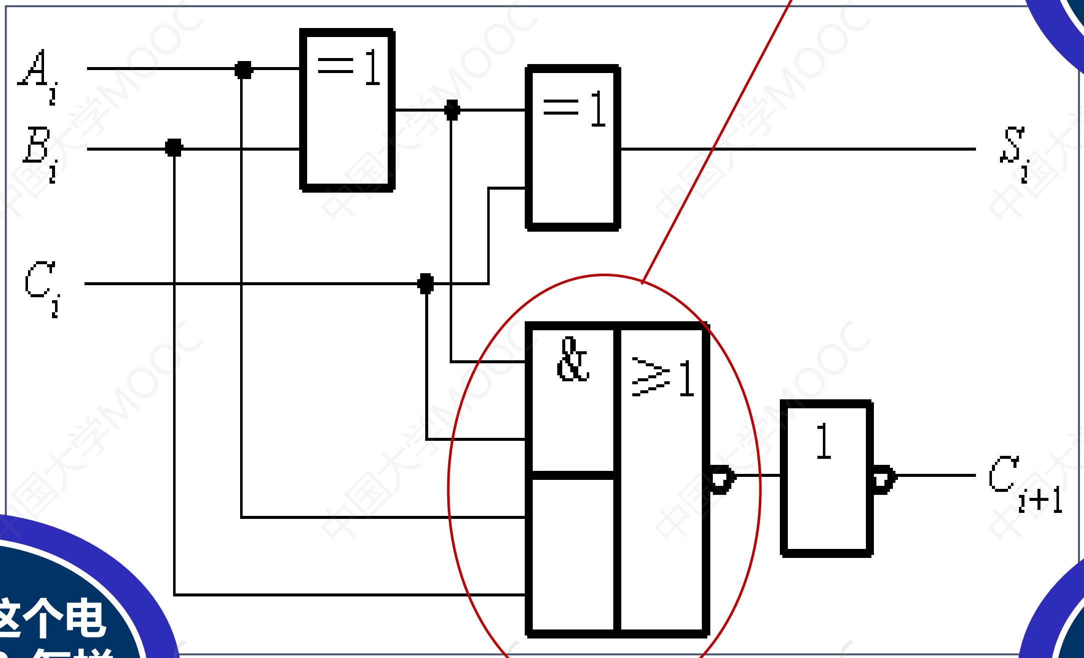


【异或门】电路符号

加法运算是这样实现的

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？

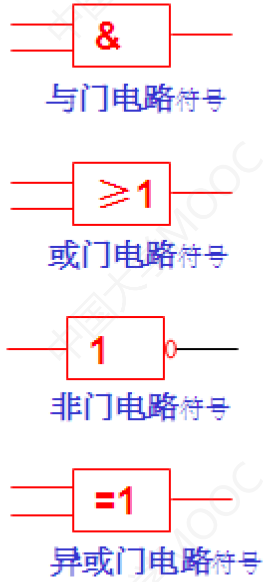
疑问2：为什么前面1个非门后面又有一个非门，有意义吗？

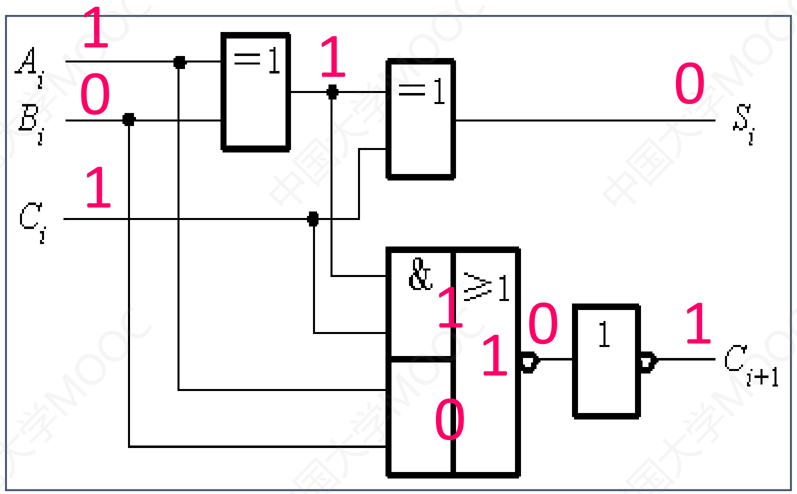
疑问3：这个电路正确吗？怎样验证其正确性呢？

加法运算是这样实现的

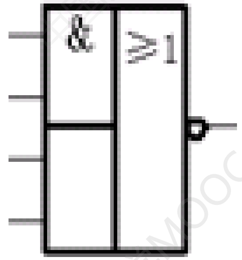
二进制加法运算用逻辑门电路的组合来实现

枚举-计算-验证：通过枚举所有可能的输入，可验证一位加法器实现的正确性





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



加法运算是这样实现的

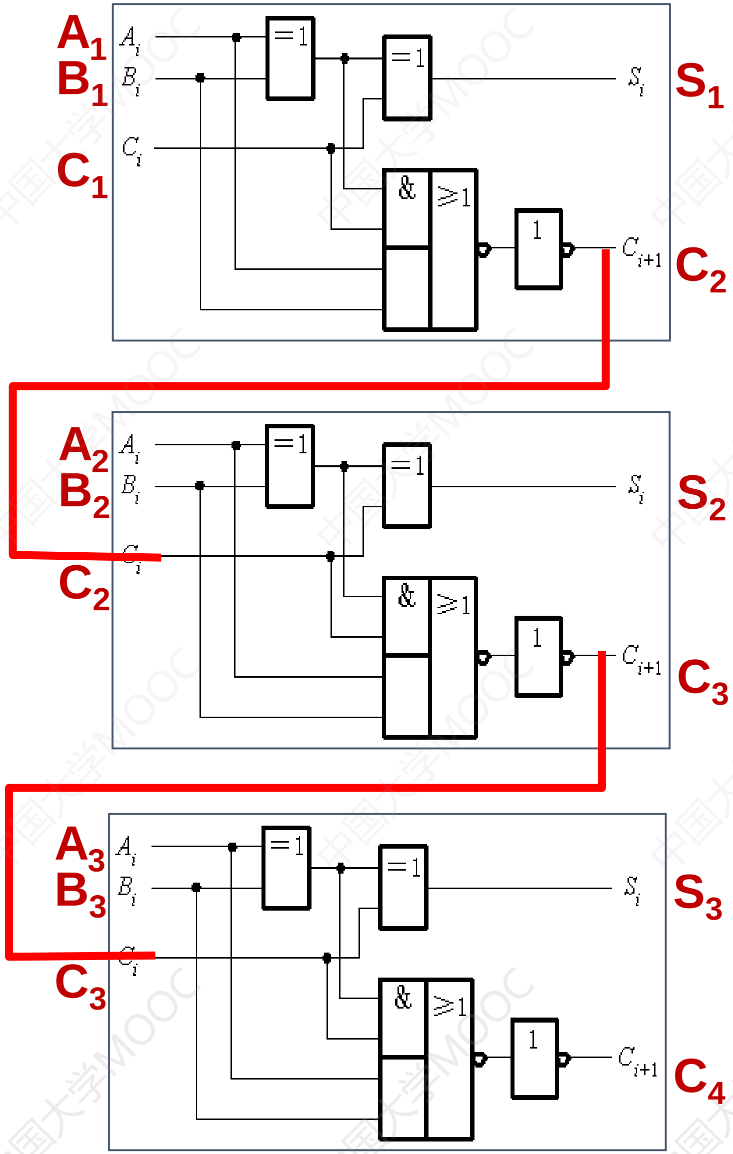
二进制加法运算用逻辑门电路的组合来实现

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/除法器等(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。

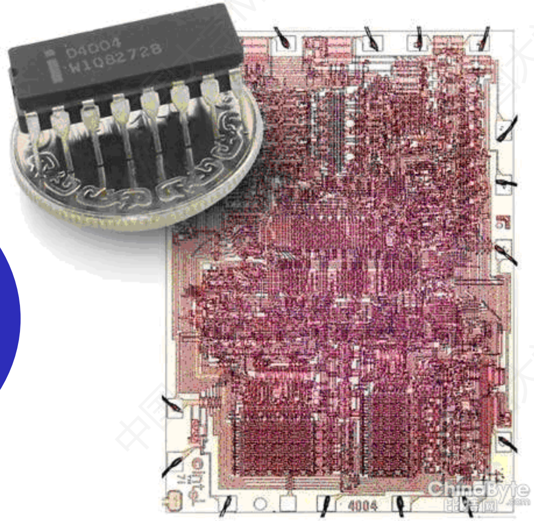
$$
\begin{array}{cccccc}
 & & A_3 & A_2 & A_1 \\
+ & & B_3 & B_2 & B_1 \\
\hline
 & C_4 & C_3 & C_2 & C_1 \\
 & & S_3 & S_2 & S_1
\end{array}
\qquad
\begin{array}{cccc}
 & 0 & 0 & 1 \\
+ & 0 & 1 & 1 \\
\hline
 & 1 & 1 & 0 \\
 & 1 & 0 & 0
\end{array}
$$

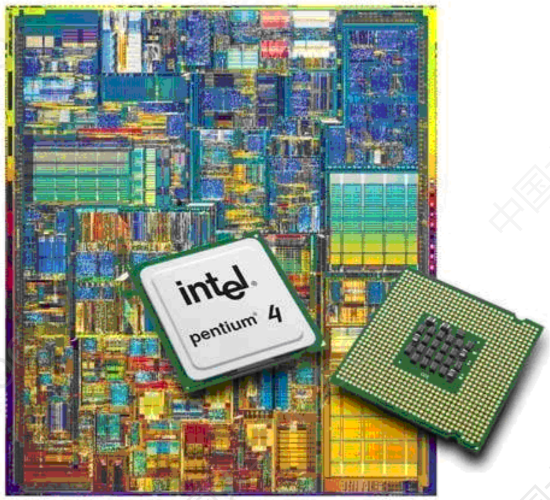


集成电路是这样复杂起来的

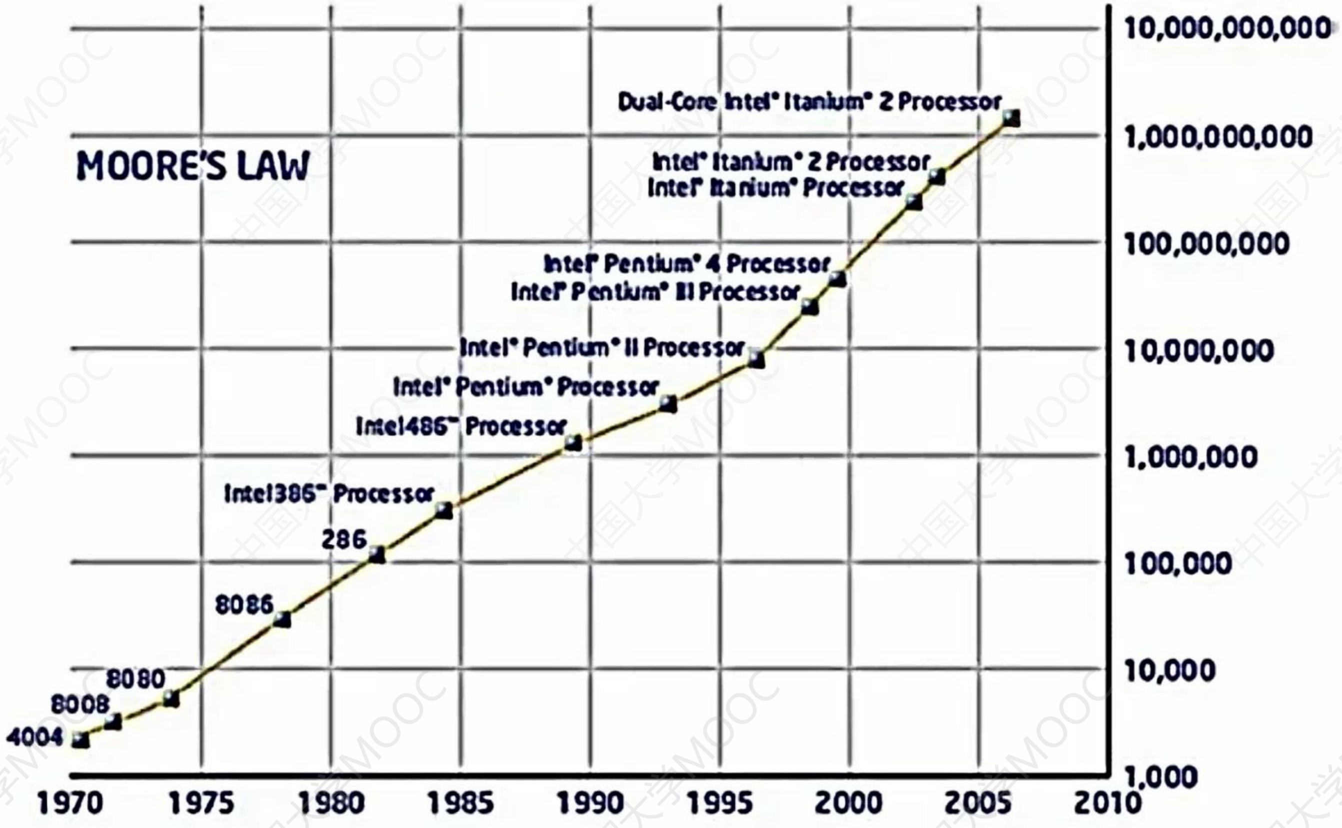
集成电路是这样的…

疑问：为什么需要这么多晶体管？



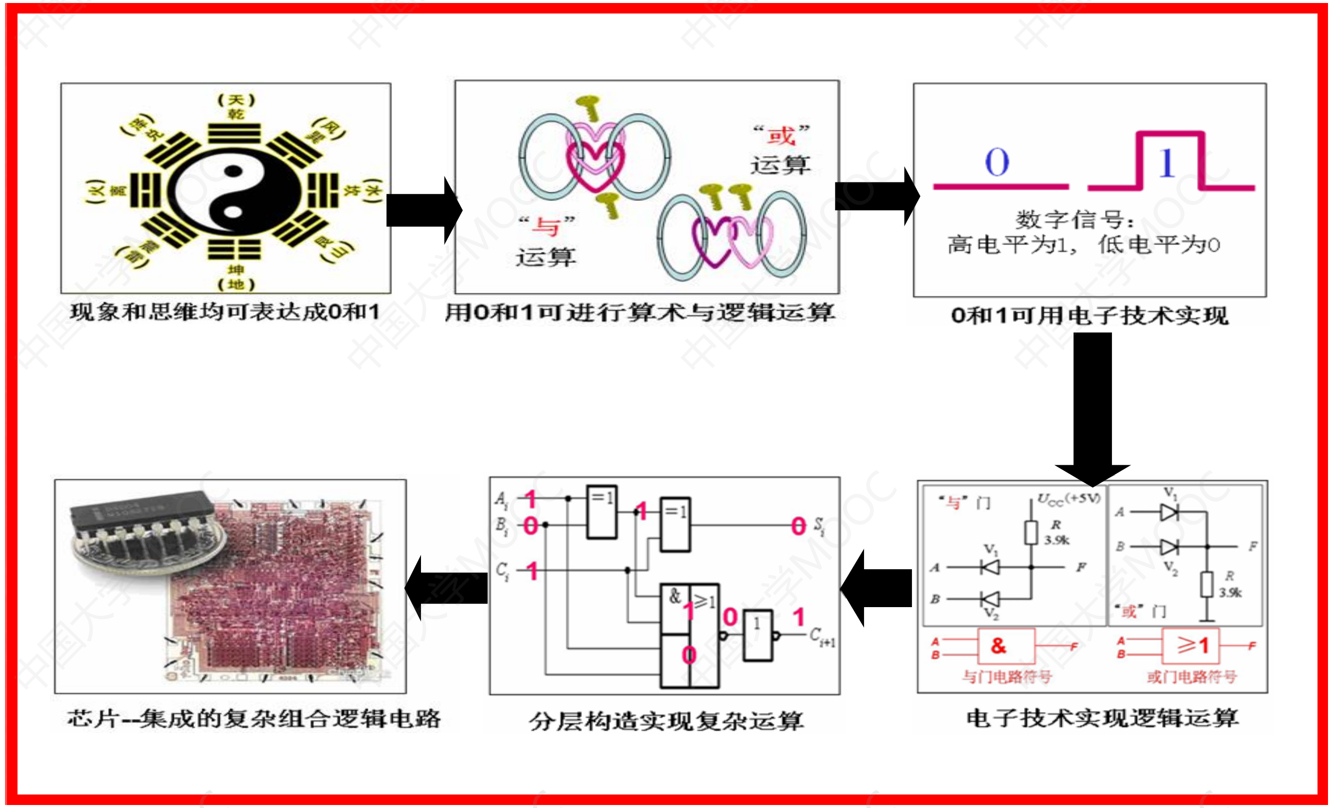


摩尔定律：芯片可容纳的晶体管数目大约每经过18个月便会增加一倍：2021年达到的规模一块芯片4.6万亿颗晶体管



计算机的本质是逻辑

小结：符号化、计算化与自动化



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