《计算概论A》课程

计算机的基本原理

李　戈

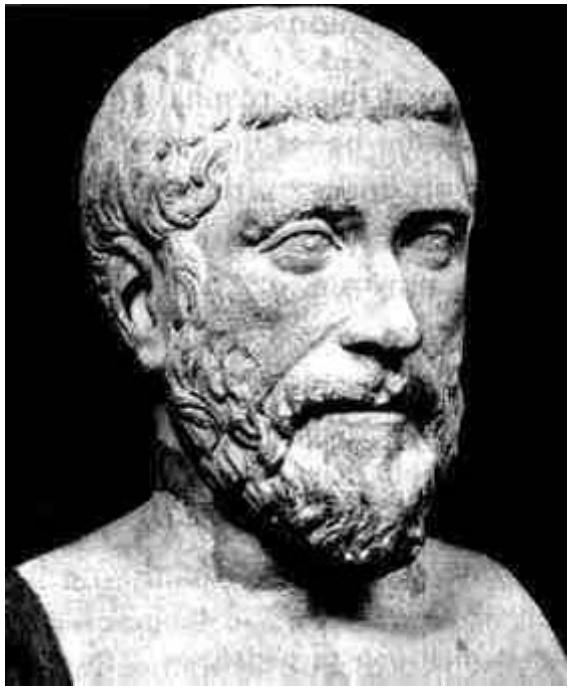
北京大学 信息科学技术学院 软件研究所

2010年9月10日

北京大学

图灵机的由来

- 第一次数学危机
  - 现代意义下的数学来源于公元前500年左右古希腊的毕达哥拉斯学派。他们认为“万物皆数”，“一切数均可表成整数或整数之比”是这一学派的数学信仰.
- “希帕索斯悖论”
  - 毕达哥拉斯证明了勾股定理，也同时发现了某些直角三角形的三边比不能用整数来表达，也就是勾长或股长与弦长是不可通约的。
- 危机的缓解：
  - 到十九世纪下半叶，实数理论建立后，无理数本质被彻底搞清，无理数在数学中合法地位的确立，才真正彻底、圆满地解决了第一次数学危机。



北京大学

图灵机的由来

- 第二次数学危机
  - 十七世纪，牛顿与莱布尼兹各自独立发现了微积分，但两人的理论都建立在无穷小分析之上，而对基本概念无穷小量的理解与运用却是混乱的。
- “贝克莱悖论”
  - 无穷小量在牛顿的理论中“一会儿是零，一会儿又不是零”。贝克莱嘲笑无穷小量是“已死量的幽灵”。
- 危机的缓解：
  - 19世纪末，柯西、魏尔斯特拉斯、戴德金、康托尔各自经过独立的研究，重建微积分学基础，都将分析基础归结为实数理论，数学分析的无矛盾性问题归纳为实数论的无矛盾性，使微积分学建立在牢固可靠的基础之上。

北京大学

图灵机的由来

- 第三次数学危机
  - 十九世纪下半叶，康托尔创立了著名的集合论。数学家们发现，从自然数与康托尔集合论出发可建立起整个数学大厦。
  - 集合论成为现代数学的基石。“一切数学成果可建立在集合论基础上”.
  - 1900年，国际数学家大会上，法国著名数学家庞加莱就曾兴高采烈地宣称：“…借助集合论概念，我们可以建造整个数学大厦…今天，我们可以说绝对的严格性已经达到了…”

北京大学

图灵机的由来

- “罗素悖论”
  - 在塞尔维亚有一位理发师，他宣称：他只给所有不给自己理发的人理发，不给那些给自己理发的人理发。可是当他自己要理发时，却陷入了尴尬境地。
  - 若他不给自己理发，根据他的第一个条件，则应该给自己理发；若给自己理发，根据他第二个条件，他不该给自己理发。总之，无论理不理发，都违背了自己的诺言。
  - S由一切不是自身元素的集合所组成。然后罗素问：S是否属于S呢？如果S属于S，根据S的定义，S就不属于S；反之，如果S不属于S，同样根据定义，S就属于S。无论如何都是矛盾的。

德国数学家、逻辑学家弗雷格：

“一位科学家不会碰到比这更难堪的事情了，在他的工作即将结束时，其基础崩溃了。罗素先生的一封信正好把我置于这个境地。”

北京大学

图灵机的由来

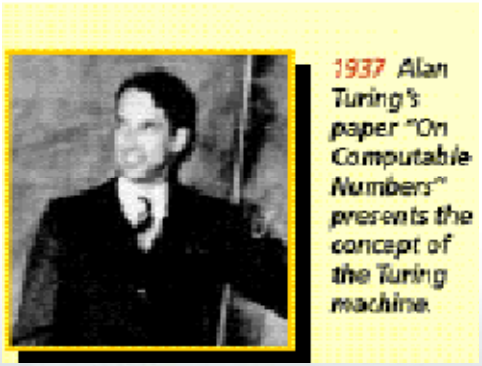
- 哥德尔不完备性定理：
  - 哥德尔于1931年成功地证明：任何一个数学系统，只要它是从有限的公理和基本概念中推导出来的，并且从中能推证出自然数系统，就可以在其中找到一个命题，对于它我们既没有办法证明，又没有办法推翻。
  - 哥德尔不完全定理的证明结束了关于数学基础的争论，宣告了把数学彻底形式化的愿望是不可能实现的。

什么是可计算的 (Computable)?

- 可计算：在可以预先确定的时间和步骤之内能够具体进行的计算。
  - 在电子数字计算机出现之前，数理逻辑学家们就开始研究可计算问题了。他们的思路是：
    - *为计算建立一个数学模型，称为计算模型，然后证明，凡是这个计算模型能够完成的任务，就是可计算的任务。*
  - 图灵机就是这样的一个计算模型。
    - *给定符号序列A，如果用图灵机能够得出对应的符号序列B，那么从A 到B 就是可计算的。*

图灵与图灵机

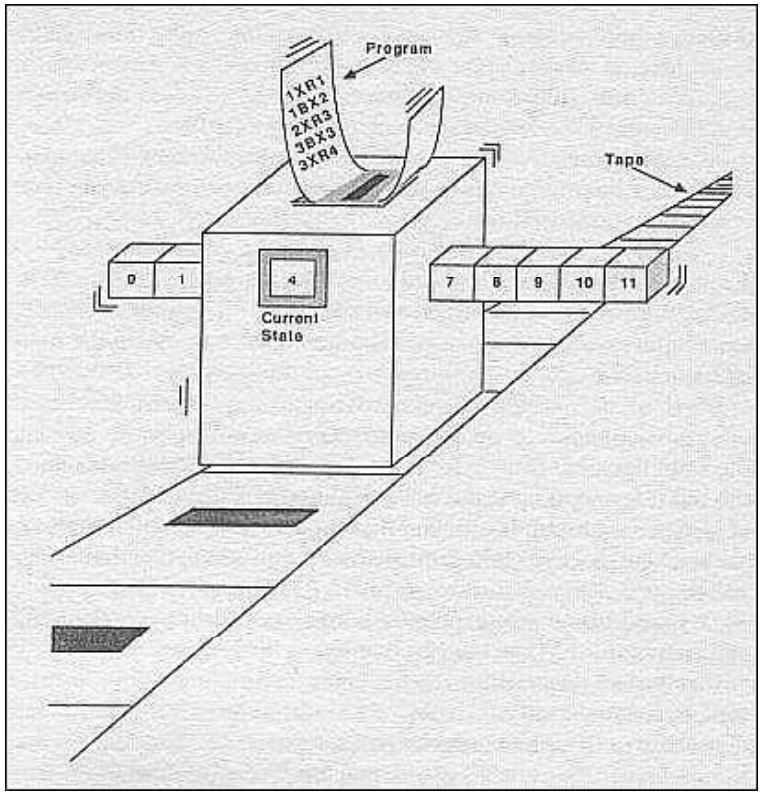
- 英国数学家艾伦•图灵（**Alan Turing**）
  - **1936**年，图灵在其著名的论文《论可计算数在判定问题中的应用》一文中提出了一种理想的计算机器的数学模型——图灵机（**Turing Machine**）。
  - 一切可能的机械式计算过程都能由图灵机实现



北京大学

图灵机

- 图灵机的组成
  - 一条双向都可无限延长的被分为一个个小方格的存储带
  - 一个有限状态控制器
  - 一个读写磁头组成

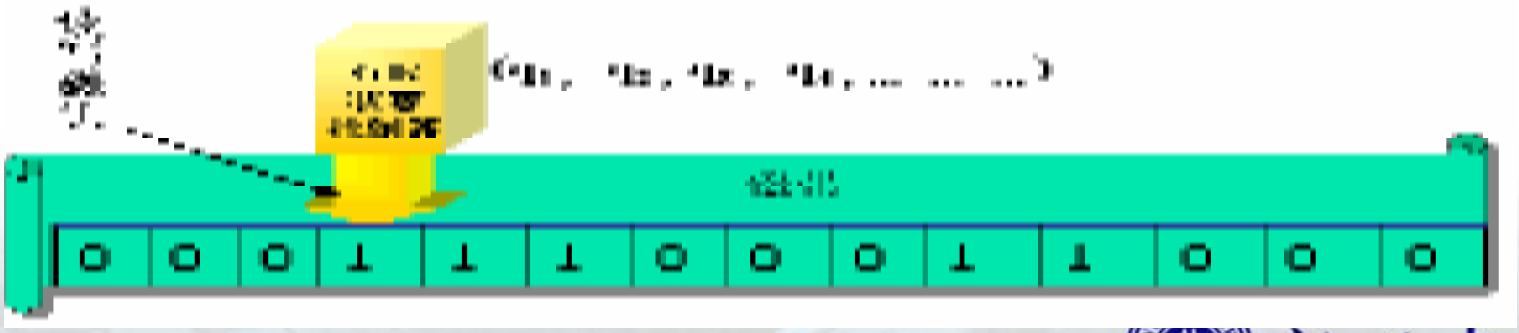


- 设：初始时存储带上的符号序列是输入I；
- 运行结束时带上的符号序列是输出O；
- 图灵机T实现的计算就是T(I) = O

北京大学

每一步工作的过程

- 读“读写磁头”当前所扫描的存储带方格的存储内容si；
- 根据si的值、“有限控状态制器”当前内部状态qi 和 运算法则
  - 决定在当前方格写入新的内容si’
  - 并决定“读写磁头”由当前位置或左移(一格)、或右移(一格)、或不移动、或停机
  - 并决定“有限控状态制器”新的控制状态qi+1
- 进入下一步工作，按同样的方式工作；如此周而复始，直到遇到命令机器停机。



图灵机程序

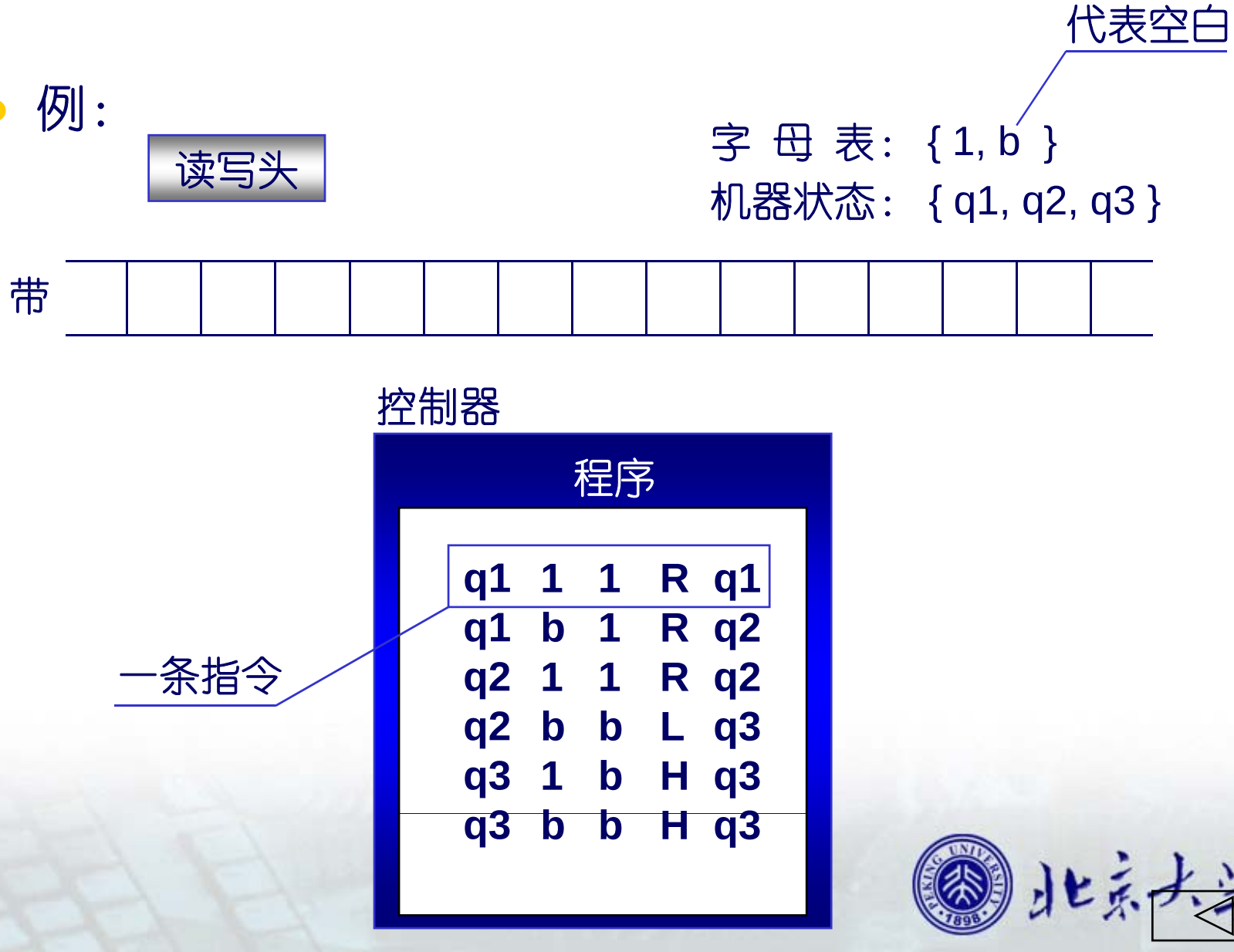
- 图灵机程序可以由一个五元组序列来定义：

  <q, b, a, m, q'>

  - **q**：当前状态
  - **q'**：下一状态
  - **b**：当前方格中的符号
  - **a**：当前方格中修改后的符号
  - **m**：磁头移动的方向，左移**L**、右移**R**，不动**H**
- **<q2, 1, 0, R, q5>**
  - 当前控制器处于状态**q2**，磁头所指的磁带方格内容为 **1**
  - 现在要采取的动作是：将该磁带方格的内容改写为 **0**
  - 磁头向右移动一格，控制器内状态变为**q5**

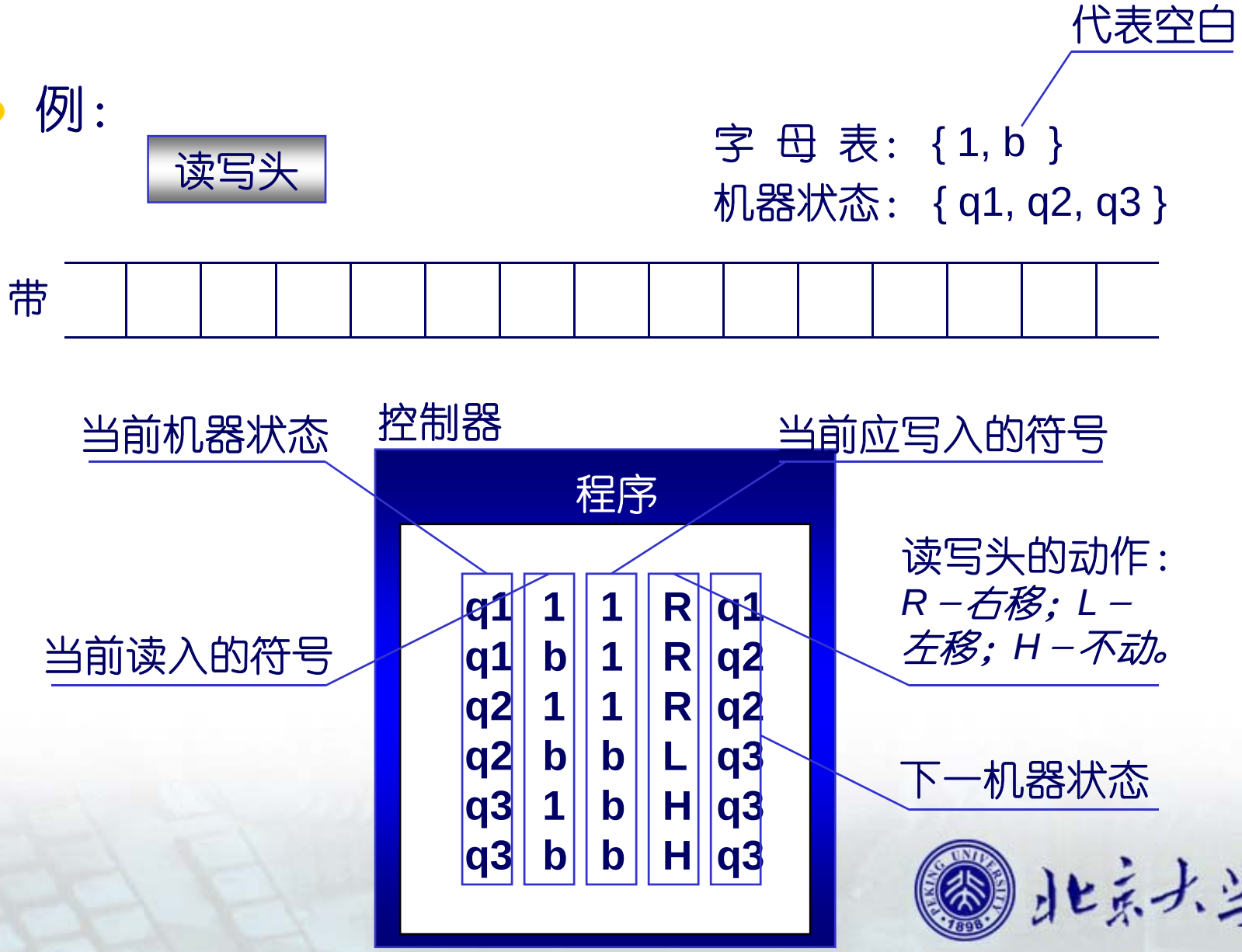
用图灵机来进行计算

- 例：


代表空白
读写头
字 母 表：{ 1, b }
机器状态：{ q1, q2, q3 }
带
控制器
程序
q1 1 1 R q1
q1 b 1 R q2
q2 1 1 R q2
q2 b b L q3
q3 1 b H q3
q3 b b H q3
一条指令

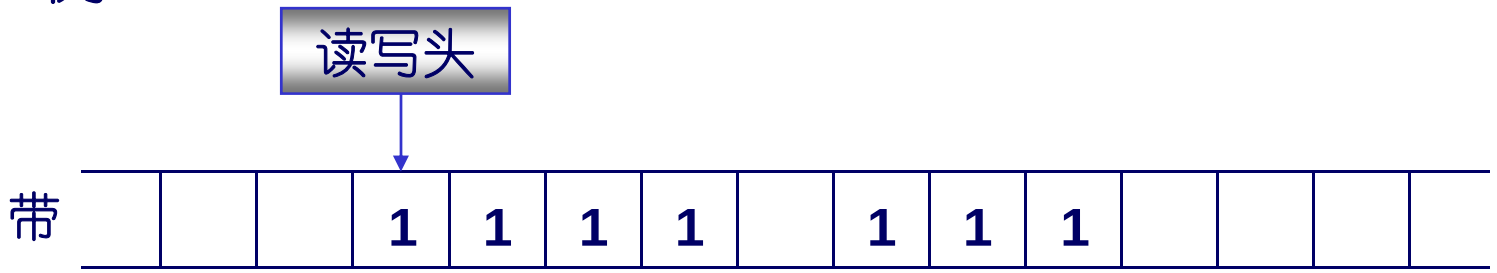
用图灵机来进行计算

- 例：


代表空白
字 母 表：{ 1, b }
机器状态：{ q1, q2, q3 }
读写头
带
当前机器状态
控制器
当前应写入的符号
程序
q1 1 1 R q1
q1 b 1 R q2
q2 1 1 R q2
q2 b b L q3
q3 1 b H q3
q3 b b H q3
当前读入的符号
读写头的动作：
R－右移；L－左移；H－不动。
下一机器状态

用图灵机来进行计算

- 例：



当前机器状态：q1

控制器

程序

q1	1	1	R	q1
q1	b	1	R	q2
q2	1	1	R	q2
q2	b	b	L	q3
q3	1	b	H	q3
q3	b	b	H	q3

指令各部分的合作：

1）在当前机器状态下

2）判断读入的符号

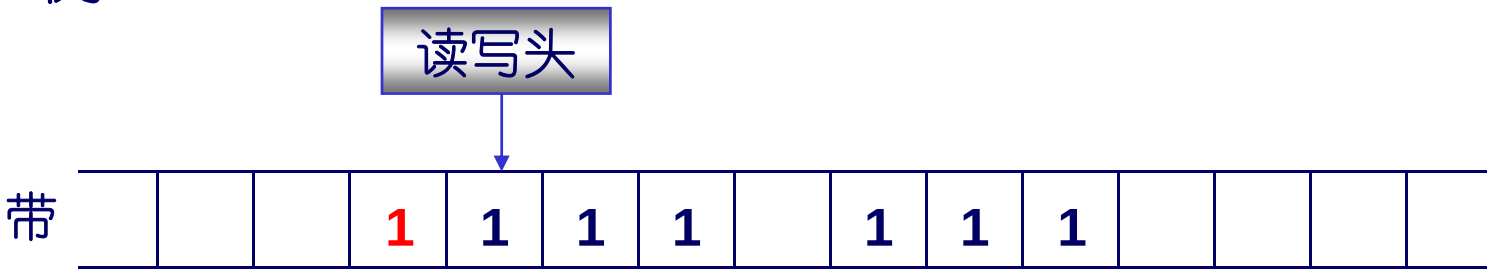
3）写一个符号

4）控制读写头动作

5）设置下一机器状态

用图灵机来进行计算

- 例：



当前机器状态：q1

控制器

程序

q1	1	1	R	q1
q1	b	1	R	q2
q2	1	1	R	q2
q2	b	b	L	q3
q3	1	b	H	q3
q3	b	b	H	q3

指令各部分的合作：

1）在当前机器状态下

2）判断读入的符号

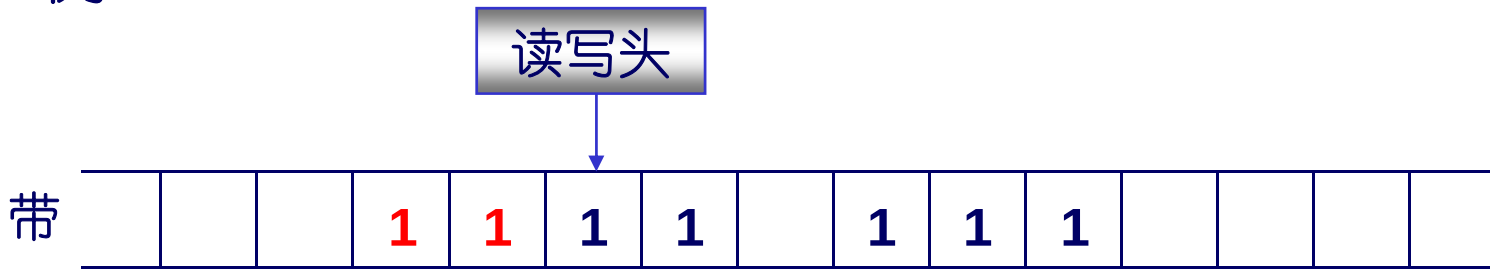
3）写一个符号

4）控制读写头动作

5）设置下一机器状态

用图灵机来进行计算

- 例：



当前机器状态：**q1**

控制器

程序

q1	1	1	R	q1
q1	b	1	R	q2
q2	1	1	R	q2
q2	b	b	L	q3
q3	1	b	H	q3
q3	b	b	H	q3

指令各部分的合作：

1）在当前机器状态下

2）判断读入的符号

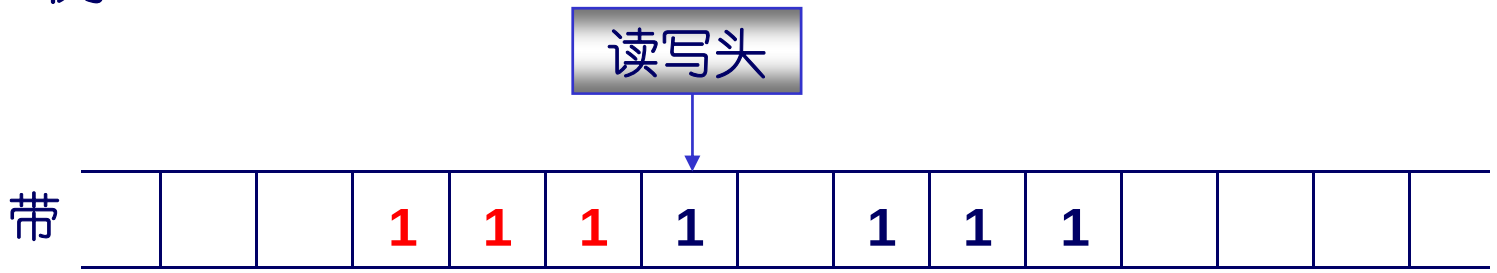
3）写一个符号

4）控制读写头动作

5）设置下一机器状态

用图灵机来进行计算

- 例：



当前机器状态：q1

控制器

程序

q1	1	1	R	q1
q1	b	1	R	q2
q2	1	1	R	q2
q2	b	b	L	q3
q3	1	b	H	q3
q3	b	b	H	q3

指令各部分的合作：

1）在当前机器状态下

2）判断读入的符号

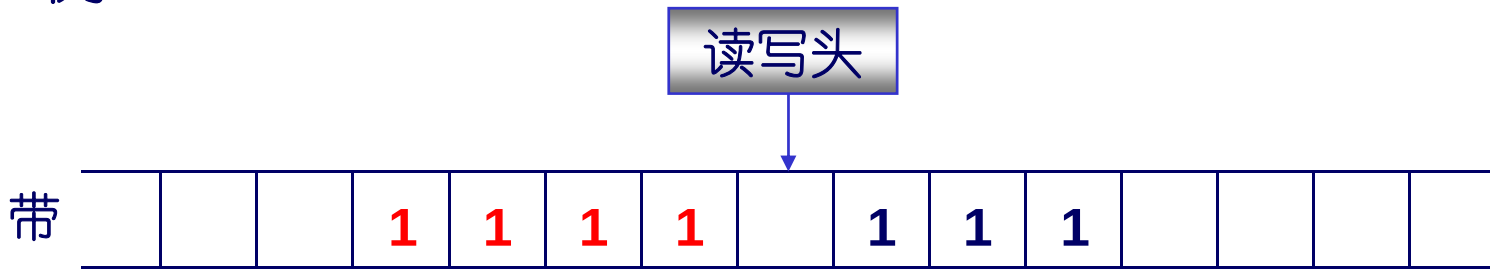
3）写一个符号

4）控制读写头动作

5）设置下一机器状态

用图灵机来进行计算

- 例：



当前机器状态：**q1**

控制器

程序

q1	1	1	R	q1
q1	b	1	R	q2
q2	1	1	R	q2
q2	b	b	L	q3
q3	1	b	H	q3
q3	b	b	H	q3

指令各部分的合作：

1）在当前机器状态下

2）判断读入的符号

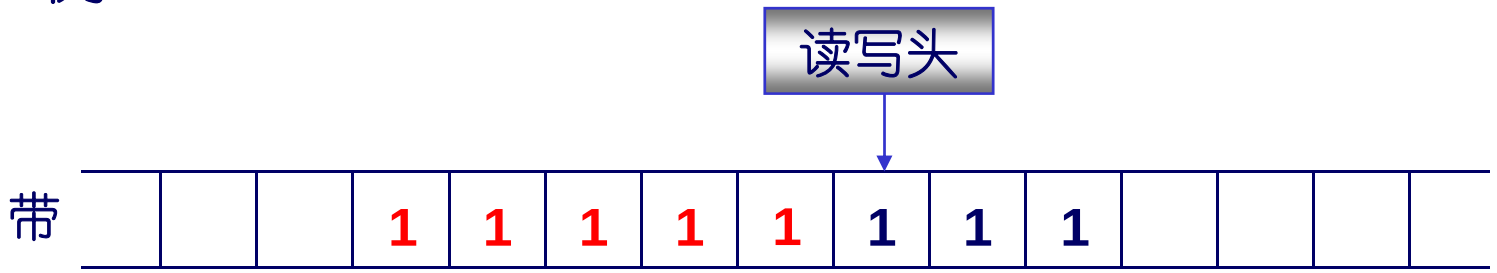
3）写一个符号

4）控制读写头动作

5）设置下一机器状态

用图灵机来进行计算

- 例：



当前机器状态：q2

控制器

程序

q1	1	1	R	q1
q1	b	1	R	q2
q2	1	1	R	q2
q2	b	b	L	q3
q3	1	b	H	q3
q3	b	b	H	q3

指令各部分的合作：

1）在当前机器状态下

2）判断读入的符号

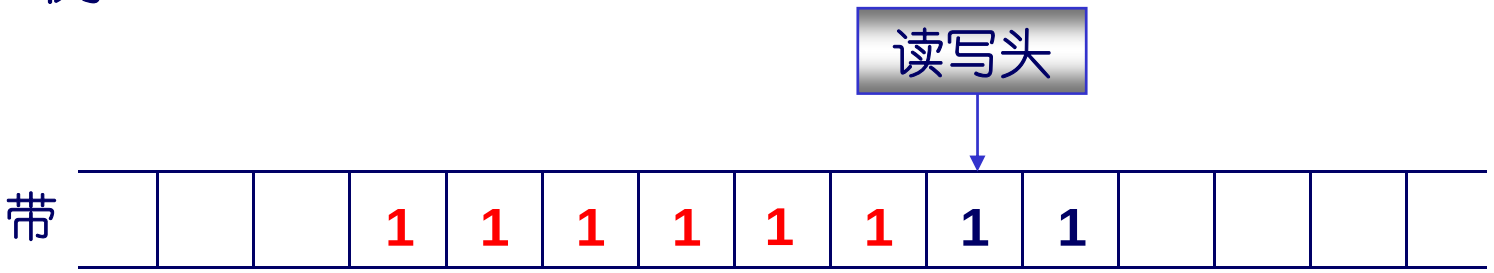
3）写一个符号

4）控制读写头动作

5）设置下一机器状态

用图灵机来进行计算

- 例：



当前机器状态：q2

控制器

程序

q1	1	1	R	q1
q1	b	1	R	q2
q2	1	1	R	q2
q2	b	b	L	q3
q3	1	b	H	q3
q3	b	b	H	q3

指令各部分的合作：

1）在当前机器状态下

2）判断读入的符号

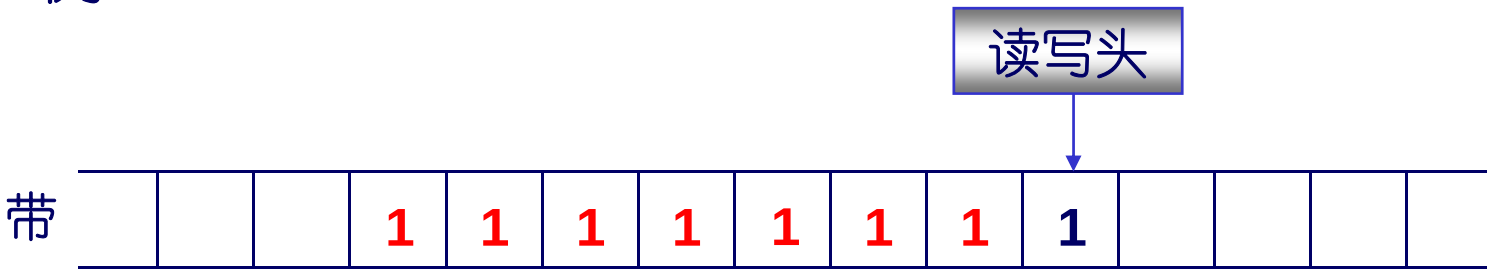
3）写一个符号

4）控制读写头动作

5）设置下一机器状态

用图灵机来进行计算

- 例：



当前机器状态：q2

控制器

程序

q1	1	1	R	q1
q1	b	1	R	q2
q2	1	1	R	q2
q2	b	b	L	q3
q3	1	b	H	q3
q3	b	b	H	q3

指令各部分的合作：

1）在当前机器状态下

2）判断读入的符号

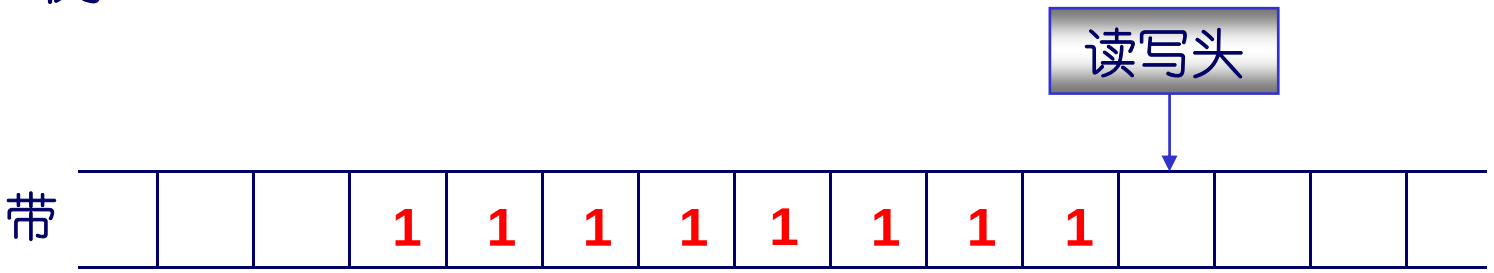
3）写一个符号

4）控制读写头动作

5）设置下一机器状态

用图灵机来进行计算

- 例：



当前机器状态：**q2**

控制器

程序

q1	1	1	R	q1
q1	b	1	R	q2
q2	1	1	R	q2
q2	b	b	L	q3
q3	1	b	H	q3
q3	b	b	H	q3

指令各部分的合作：

1）在当前机器状态下

2）判断读入的符号

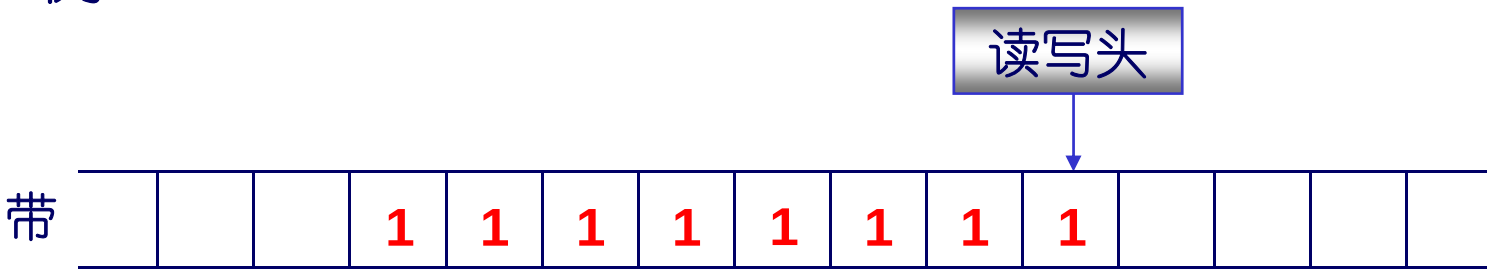
3）写一个符号

4）控制读写头动作

5）设置下一机器状态

用图灵机来进行计算

- 例：



当前机器状态：**q3**

控制器

程序

q1	1	1	R	q1
q1	b	1	R	q2
q2	1	1	R	q2
q2	b	b	L	q3
q3	1	b	H	q3
q3	b	b	H	q3

指令各部分的合作：

1）在当前机器状态下

2）判断读入的符号

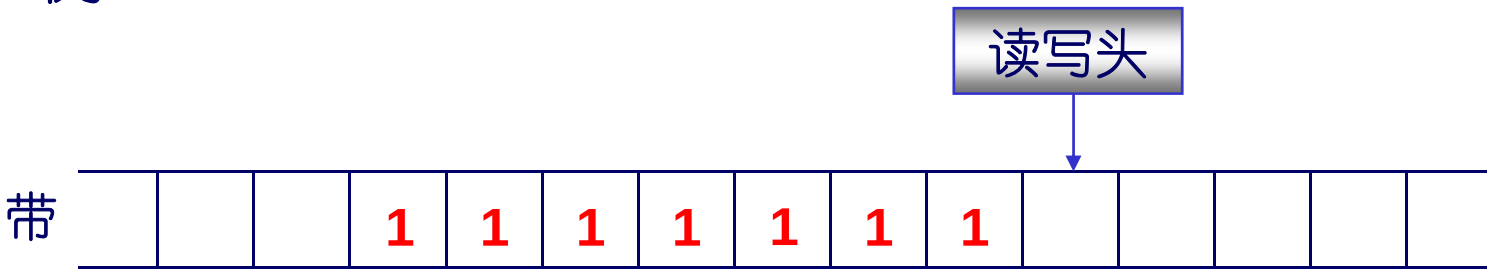
3）写一个符号

4）控制读写头动作

5）设置下一机器状态

用图灵机来进行计算

- 例：



当前机器状态：**q3**

控制器

程序

q1	1	1	R	q1
q1	b	1	R	q2
q2	1	1	R	q2
q2	b	b	L	q3
q3	1	b	H	q3
q3	b	b	H	q3

指令各部分的合作：

1）在当前机器状态下

2）判断读入的符号

3）写一个符号

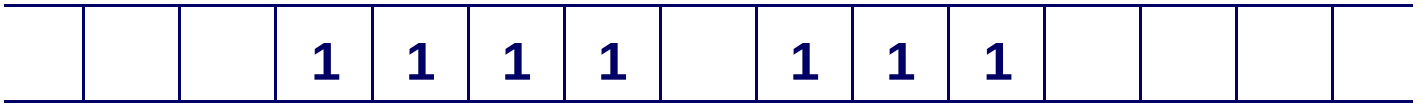
4）控制读写头动作

5）设置下一机器状态

用图灵机来进行计算

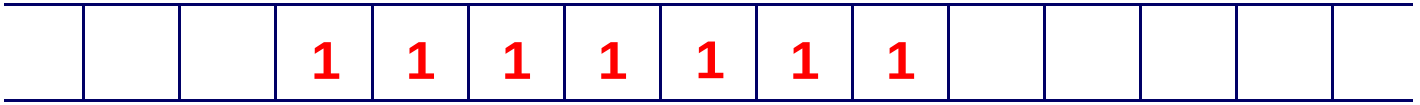
- 这个例子中，是在用图灵机进行什么计算？

开始时：



x 个 1　　y 个 1

结束时：



x+y 个 1

- 这是在进行任意两个大于 0 的整数的相加。

图灵机的特点

- 具有可实现性：
  - 图灵机在一定程度上反映了人类最基本的、最原始的计算能力，它的基本动作非常简单、机械、确定。
- 具有函数F(x)的计算能力：
  - 同一个图灵机可以进行规则相同、对象不同的计算。输入不同的开始的状态（读写头的位置、机器状态），获得相应的输出状态（计算结果）。
- 程序并非必须顺序执行
  - 虽然程序只能按线性顺序来表达指令序列，但程序的实际执行轨迹可以与表示的顺序不同。

图灵机是不是万能的?

哪些问题是图灵机不能计算的?

图灵机的用途

- 图灵的论点：
  - 一个问题是可判定的
    - *当且仅当能够找到一个图灵机程序，使得对于该问题的输入，图灵机能够给出确定的输出，即图灵机能够停机。*
  - 哪些问题是不可判定的呢？
    - *与“图灵机停机问题”等价的问题，都是不可判定的。*

图灵机停机问题

- 图灵机停机问题：
  - 求一台“万能的”图灵机 **H**, 把任意一台图灵机 **M** 输入给 **H**, 它都能判定 **M** 最终是否停机, 即： 输出一个明确的 **“yes”** 或 **“no”** 的答案。
  - 求一个判断任意一个图灵机是否停机的一般方法。
  - 图灵在**1936**年证明，图灵机的停机问题是不可判定的，即不存在一个图灵机能够判定任意图灵机对于任意输入是否停机。

利用“康托尔的对角线删除法则”的证明

- 等价描述问题：
  - 求一个程序P(X,Y)，它可以判断当任意一个程序X输入程序Y时是否存在死循环，或者说是否停机。
- 证明：
  - 假设这样的P(X,Y)是存在的。则可以为所有的程序编号：1, 2, 3, ……，对数据Y也进行编号1, 2, 3, ……

	1	2	3	4	5	6 ……	Y ……
1	yes	no	no	yes	yes	no ……	yes ……
2	no	no	yes	yes	no	yes ……	no ……
…	……						
X	no	yes	no	yes	yes	no ……	yes ……
…	……						

利用“康托尔的对角线删除法则”的证明

- 找出对角线序列
  - 把上表对角线上的元素挑选出来，得到序列：
    yes,no,no,yes, ……
- 构造对角线删除序列
  - 根据这个序列可以构造一个反序列：
    no,yes,yes,no, ……
- 对角线删除序列是否在表中的某一行上出现？

	1	2	3	4	5	6 ……	Y ……
1	yes	no	no	yes	yes	no ……	yes ……
2	no	no	yes	yes	no	yes ……	no ……
…	……						
X	no	yes	no	yes	yes	no ……	yes ……
…	……						

不可计算问题的判定

- 不可计算的问题：
  - 图灵机不能解的问题也就是一切计算机不能解的问题，也叫做不可计算的问题。
  - 存在一类问题：人类能构造出来，而图灵机是不能解
  - 计算机的计算能力存在极限。
- 计算等价性原理，
  - 所有不可计算问题都和图灵停机问题是计算等价的
  - 能够被归结为图灵停机问题的问题都是不可判定的

如何让图灵机变成计算机?

—— 关于字母表

计算机里的数字应该如何表示?

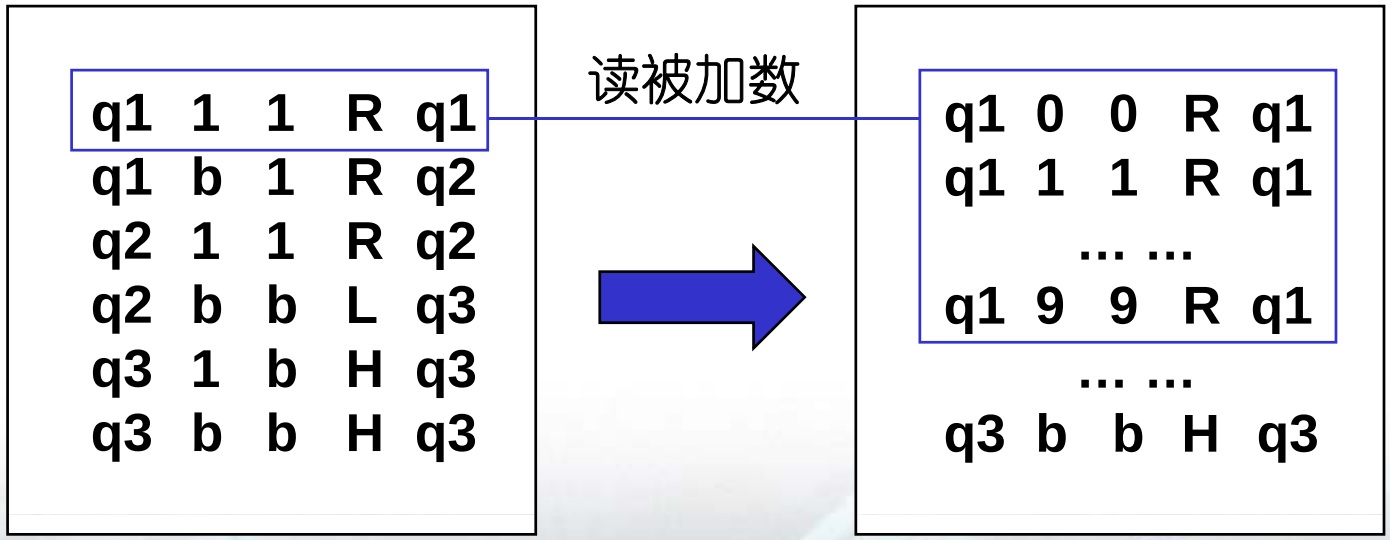
北京大学

字母表的问题

- 字母表是 { 1, b }
  - 符号 “b”
    - *表示计算对象和计算结果的边界（Blank）*
  - 符号 “ 1”
    - *表示计算对象和计算结果的数值*
- 字母表的问题：
  - 使用符号 “1”表示数字 1
    - *数值 1,000,000 应由一百万个 1 来表示*
    - *读入这一个数，读写头就要移动一百万次*
  - 显然不合理、不实际

字母表的问题

- 如果使用“十进制”
  - 字母表中包含11个符号： { 0, 1, …, 9, b }
  - 用于图灵机控制的程序要大量增加
  - 确定当前指令也需要更多的时间



字母表的问题

- 怎样用机器来表示带上的符号？
  - 字母表中的符号越多，用机器表示的困难就越大
  - 有限状态控制器的状态越多，机器表示困难就越大
- 字母表与状态：
  - 字母表中符号的最优数量，是欧拉常数 **e**（**2.7182818284590...**），取整后为 **3**。
  - 与具有两个状态的电子元件相比，具有三个状态的电子元件在制造上更困难，可靠性更低。

十进制、二进制、十六进制

- 十进制：
  - 计数符号：**0、1、2、3、4、5、6、7、8、9**
  - 基数：**10**　　$256 = 2\times10^2 + 5\times10^1 + 6\times10^0$
- 二进制：
  - 基数符号：**0、1**
  - 基数：**2**　　$10110 = 1\times2^4 + 0\times2^3 + 1\times2^2 + 1\times2^1 + 0\times2^0$
- 十六进制：
  - 基数符号：**0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F**
  - 基数：**16**

$$(ABCD)_{16} = A\times16^3 + B\times16^2 + C\times16^1 + D\times16^0$$

北京大学

十进制 转换为二进制

- 将123（10）转换成等值的二进制数：

除以2的商（取整）	余数
123/2 = 61	1
61/2 = 30	1
30/2 = 15	0
15/2 = 7	1
7/2 = 3	1
3/2 = 1	1
1/2 = 0	1

- 自下而上地依次将余数加以汇集，即得到对应的二进制数：**1111011**

北京大学

二进制 转换为 八进制、十六进制

- 从 二进制 到 八进制
  - 从右向左，每三位进行一次转换，即从二进制数的值转换成等值的八进制数字。
- 从 二进制 到 十六进制
  - 从右向左，每四位进行一次转换，即从二进制数的值转换成等值的十六进制数字。
- 例：转换 **1111011（2）**
  - $1111011_{(2)} = 173_{(8)}$
  - $1111011_{(2)} = 7B_{(16)}$

小结一下

- 已经解决的问题：
  - 所有的数字运算都可以转换为二进制数的计算；
- 接下来的问题：
  - 怎么让机器（或电路）完成二进制的计算?

如何让图灵机变成计算机?

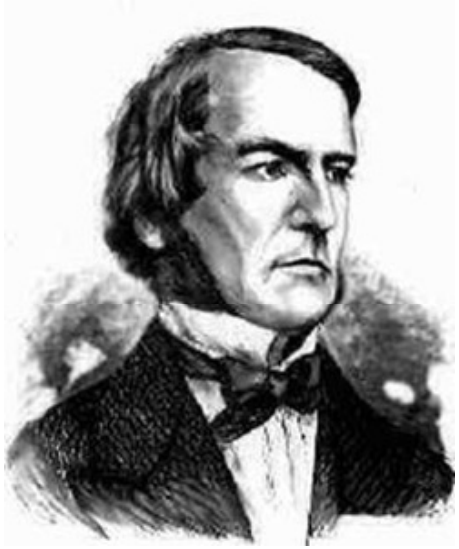
—— 关于控制器

【需要一种新的计算理论】

北京大学

控制器的问题

- 控制器的基本要求
  - 控制器必须有逻辑判断能力
  - 控制器必须有逻辑计算能力
- 布尔代数：
  - 基本状态：是、非
  - 基本计算：与、或、非



英国数学家布尔（G.Boole）

1854年：布尔发表《思维规律的研究——逻辑与概率的数学理论基础》，并综合其另一篇文章《逻辑的数学分析》，创立了一门全新的学科－布尔代数，为计算机的开关电路设计提供了重要的数学方法和理论基础。

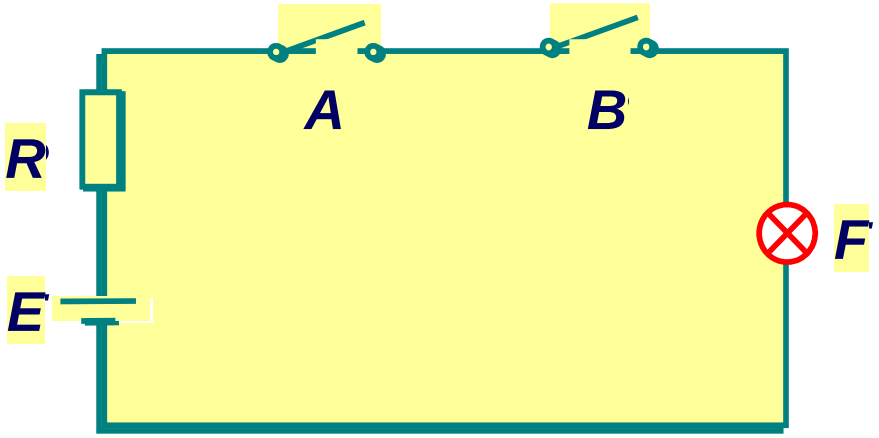
北京大学

布尔运算的基本逻辑

- 基本逻辑运算
  - 与逻辑（与运算、逻辑乘）
  - 或逻辑（或运算、逻辑加）
  - 非逻辑（非运算、逻辑反）
- 常用复合逻辑
  - “与非逻辑”
  - “或非逻辑”
  - “与或非”逻辑
  - “同或” “异或”逻辑

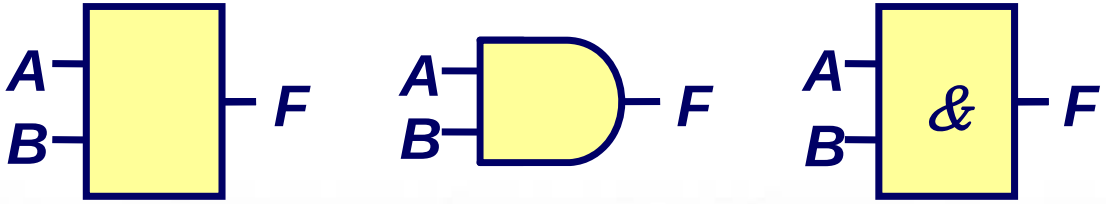
与逻辑（与运算、逻辑乘）

- 逻辑函数表达式：F= A•B



真值表

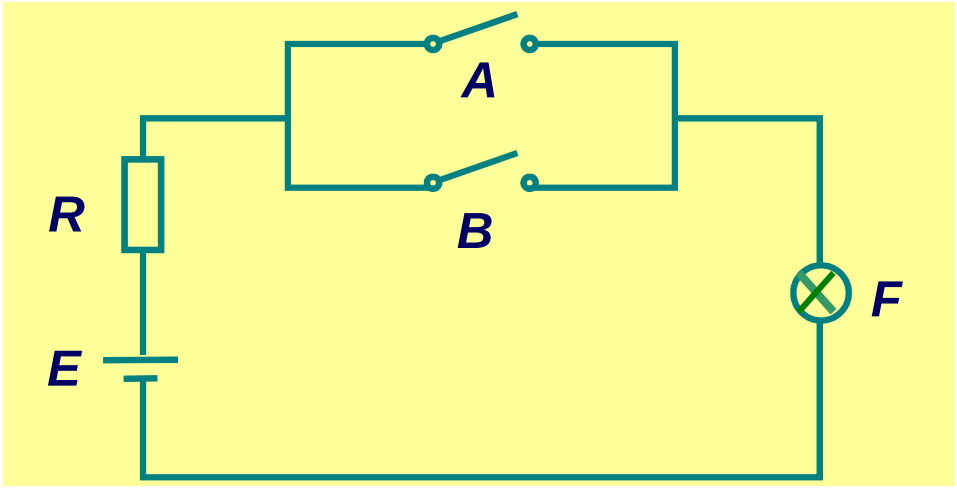
AB	F
00	0
01	0
10	0
11	1



（a）国家标准　（b）国外流行　（c）国际标准

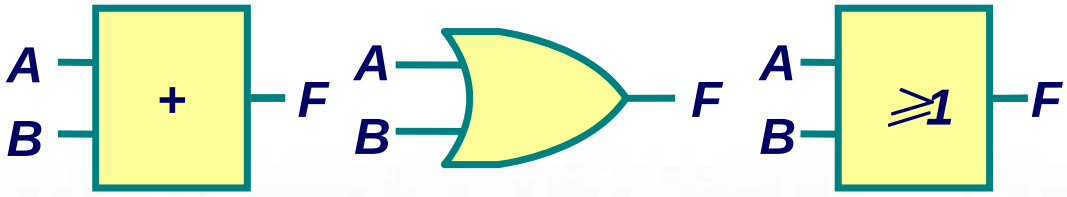
或逻辑(或运算、逻辑加)

- 逻辑函数表达式：F= A+B



真值表

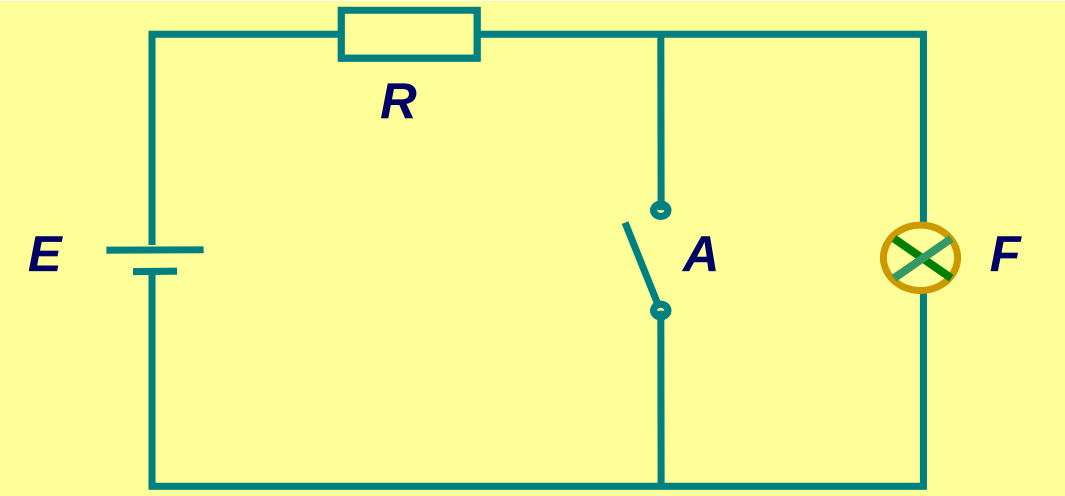
AB	F
00	0
01	1
10	1
11	1



（a）　（b）　（c）

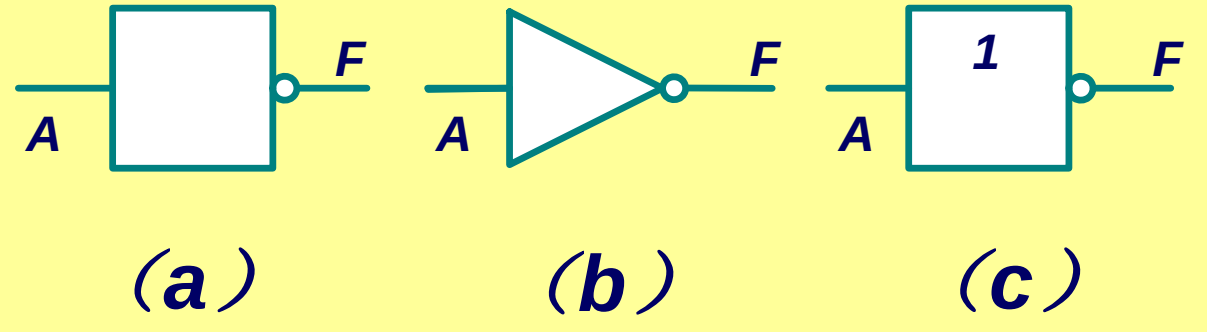
非逻辑(非运算、逻辑反)

- 逻辑函数表达式：$F = \overline{A}$



真值表

A	F
0	1
1	0



“同或” “异或”逻辑

- “异或” “同或”逻辑
  - 输入二变量相异为“1”，相同为“0”，称为“异或”F1。
  - 输入二变量相异为“0”，相同为“1”，称为“同或”F2。

异或

同或

$$F_1 = \overline{A}\,B + A\,\overline{B} = A \oplus B$$

$$F_2 = \overline{A}\,\overline{B} + A\,B = A \odot B$$

由真值表可得出下式 ：

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

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

即 $$\overline{\overline{A}\,B + A\,\overline{B}} = \overline{A}\,\overline{B} + A\,B$$

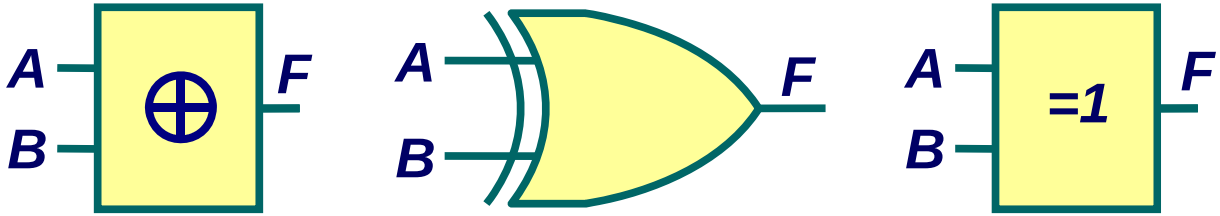
或 $$\overline{A\,B + A\,B} = \overline{A}\,B + A\,\overline{B}$$

A B	F_1	F_2
0 0	0	1
0 1	1	0
1 0	1	0
1 1	0	1

北京大学

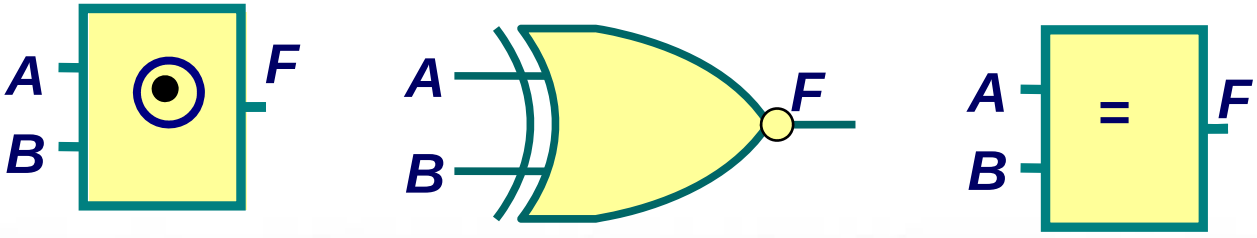
“异或” “同或”逻辑

异或逻辑符号



（a） （b） （c）

同或逻辑符号



（a） （b） （c）

加法器

举例：A=1101，B=1001，计算A+B

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

1 0 0 1

加法运算的基本规则：

（1）逢二进一。

（2）最低位的相加，不需考虑进位。

（3）其余各位都是三个数相加，包括加数、被加数和低位来的进位。

（4）任何位相加都产生两个结果：本位和、向高位的进位。

半加器

- 两个一位二进制数相加，只求本位和，不考虑低位的进位信号。

二进制加法：{ 变量只取0和1;
逢二进位。

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

本位加数

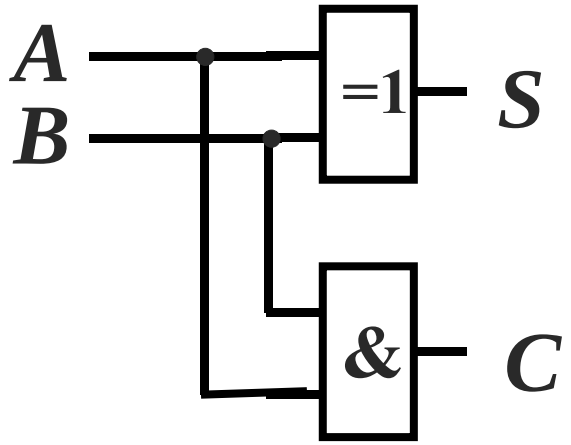
C :进位

S:本位和

北京大学

半加器

逻辑图



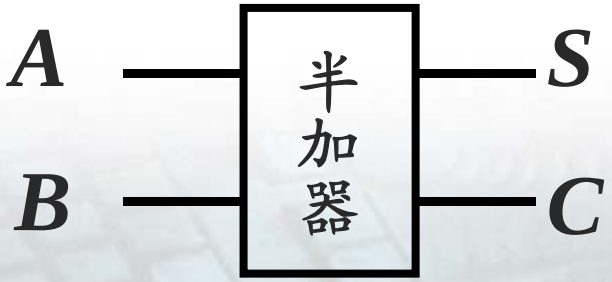
真值表

A	B	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

A---加数　　**B**---被加数

S---本位和　　**C**---进位

逻辑符号



逻辑式

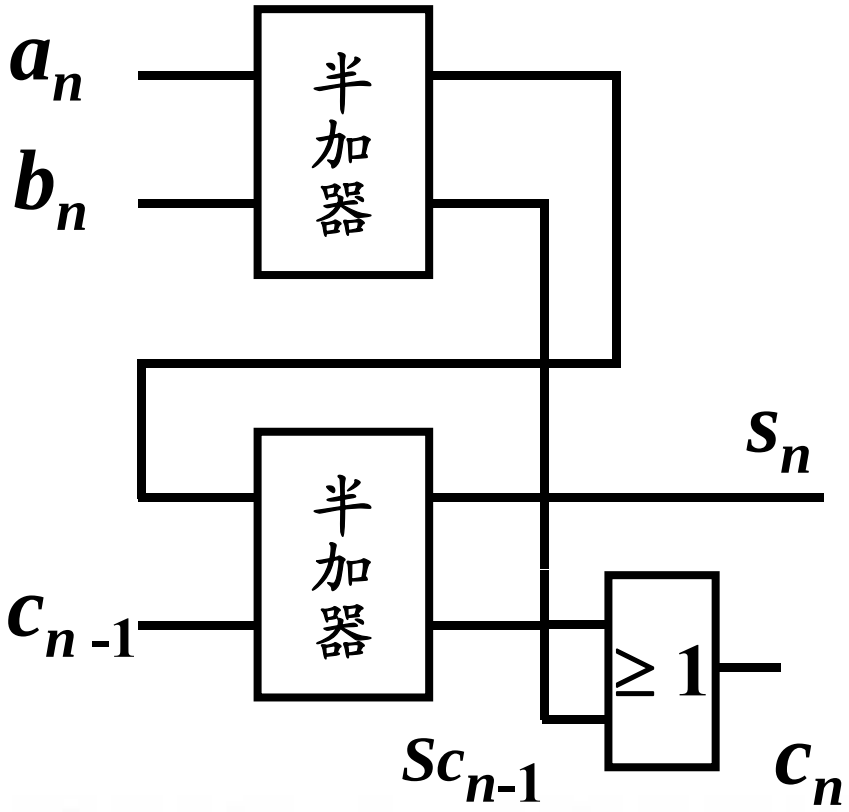
$$S=\bar{A}B+A\bar{B}=A\oplus B$$

$$C=AB$$

北京大学

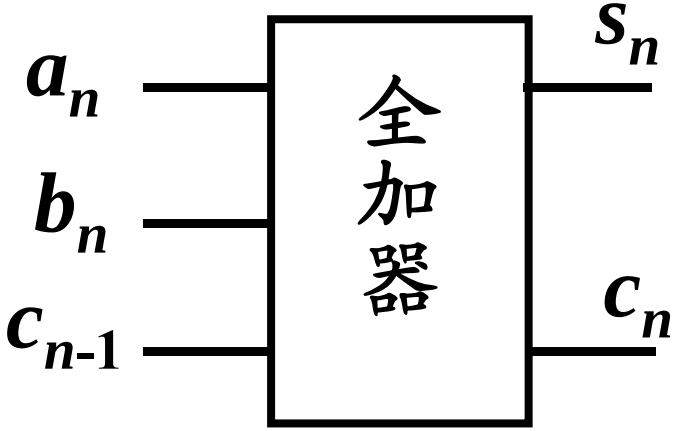
全加器

逻辑图



$$s_n = s\bar{c}_{n-1} + \bar{s}c_{n-1}$$

$$c_n = sc_{n-1} + a_n b_n$$



逻辑符号

a_n---加数；b_n---被加数；c_{n-1}---低位的进位

s_n---本位和；c_n---进位

北京大学

再小结一下

- 已经解决的问题：
  - 所有的数字运算都可以转换为二进制数的计算；
  - 所有的二进制数的计算都可以根据“布尔代数”理论转换为基本的“布尔运算”；
- 接下来的问题：
  - 基本的“布尔运算”能由电路完成吗?

数字逻辑电路

- **基本原理：**

  数的表示：

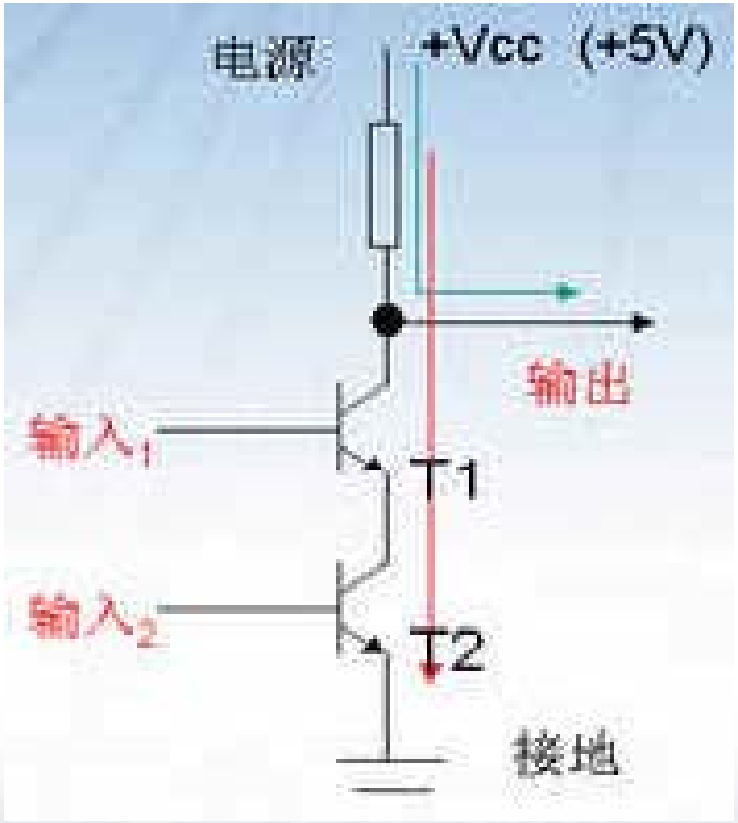
  – 利用单向导电性或电压高低表示是、非

  数的运算：

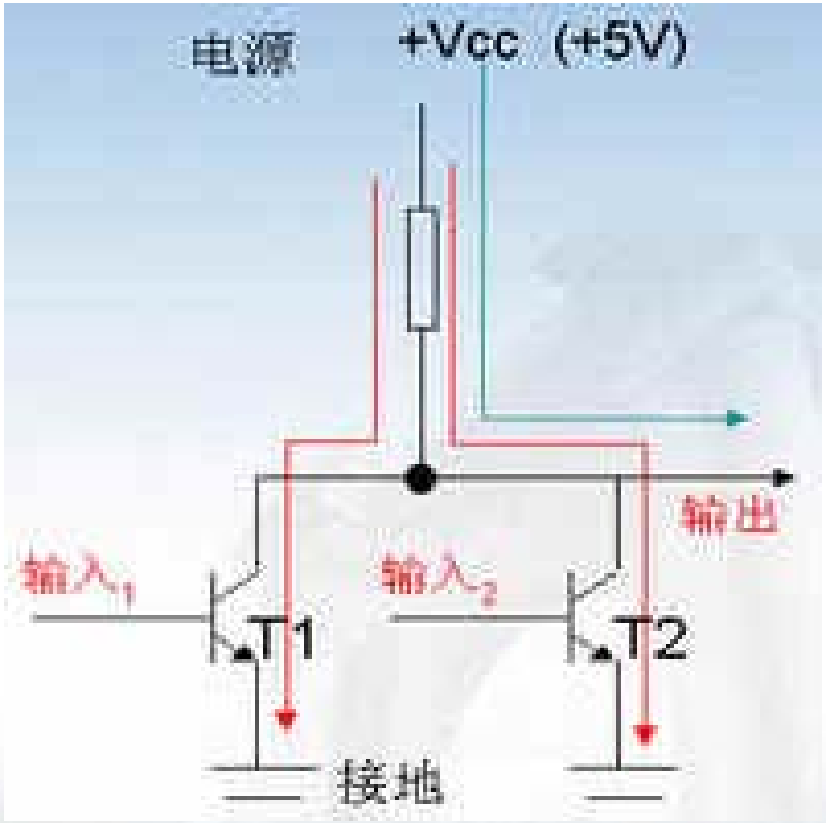
  – 利用电路之间的转换关系表达与、或、非

与非、或非电路的实现

- 与非门电路

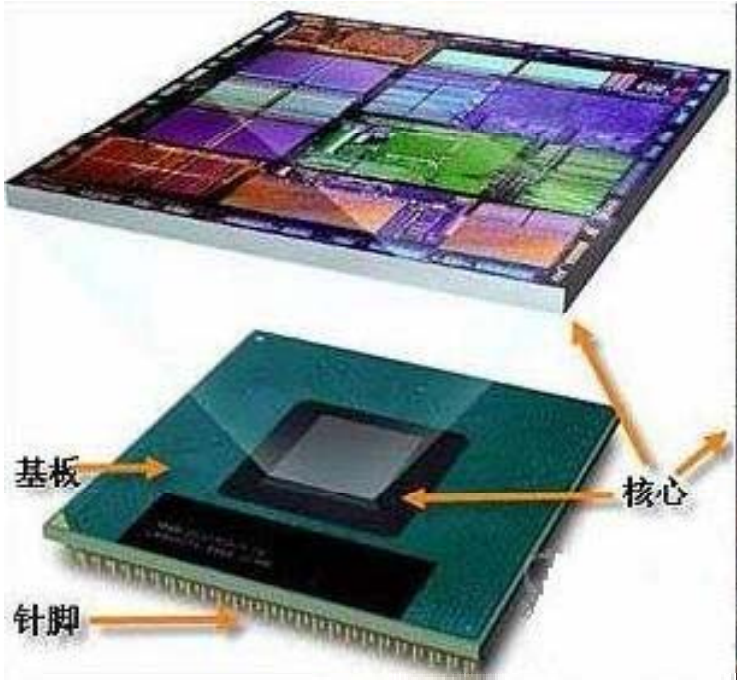


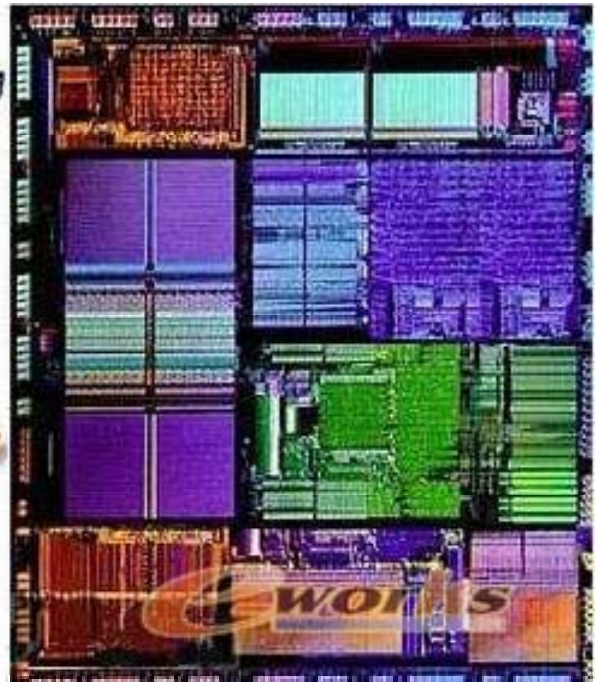
- 或非门电路

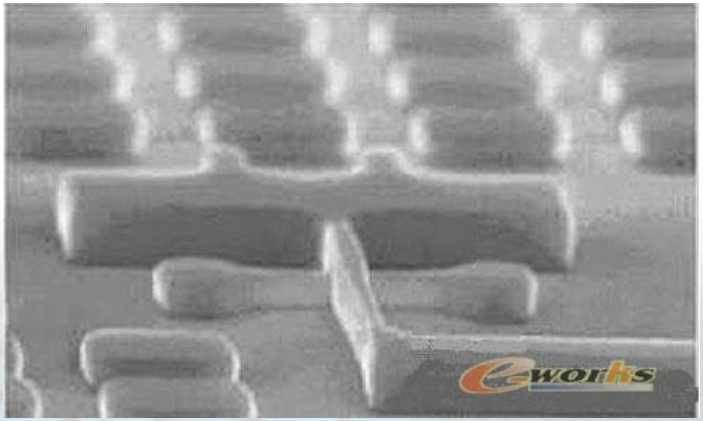


北京大学

CPU







再小结一下

- 已经解决的问题：
  - 所有的数字运算都可以转换为二进制数的计算；
  - 所有的二进制数的计算都可以根据“布尔代数”理论转换为基本的“布尔运算”；
  - 基本的“布尔运算”都可以由基本电路完成；
- 接下来的问题：
  - 程序怎么让CPU运行起来?

作 业

1. 在互联网上搜集关于CPU相关指标的资料，并确定自己所使用机器CPU的型号和各项性能指标。

2. 在互联网上搜集关于主板相关指标的资料，并确定自己所使用机器的主板型号和各项性能指标。

3. 请按照目前的市场行情给出4000元 或 7000元或10000元的个人计算机硬件配置，需要给出每个部件名称、品牌、型号、价格。

好好想想，有没有问题?

谢　谢　！

北京大学