计算概论：计算机的发展

北京大学 信息科学与技术学院 计算机系

计算机的发展

过去
- 数学危机到图灵机
- 图灵机原理
- 数的二进制表示
- 二进制的布尔运算

现在
- 历史上的计算设备
- 从电子管到云计算
- 从硬件到软件

未来
- 摩尔定律下的计算危机
- 量子计算

过去

- 数学危机到图灵机
- 图灵机原理
- 数的二进制表示
- 二进制的布尔运算

- 历史上的计算设备
- 从电子管到云计算
- 从硬件到软件

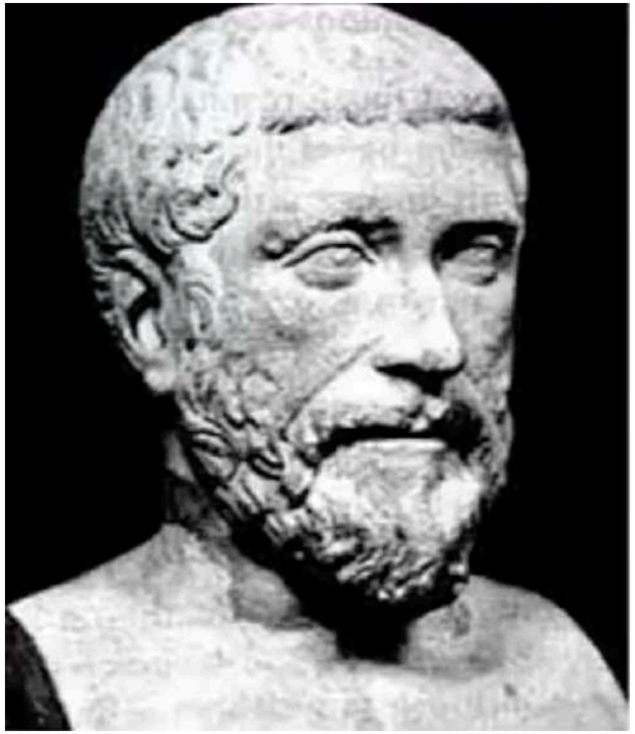
- 摩尔定律下的计算危机
- 量子计算

数学危机到图灵机

- **第一次数学危机背景**

  - 毕达哥拉斯学派（公元前500年，古希腊）

    相信：数是万物的本源，一切数均可以表达为整数或者整数之比



- 毕达哥拉斯证明了**勾股定理**

- 但证明勾股定理的过程中发现：

  某些直角三角形的三边之比**不能**用整数来表示的！
  （与当时数学的信仰产生了矛盾）

  为了数学的生存，毕达哥拉斯没有公布这个问题。。

数学危机到图灵机

- **第一次数学危机的出现**
  - 毕达哥拉斯的学生－希帕索斯（古希腊）

    但是，他的学生却把这个问题公布了出来，成为“希帕索斯悖论”

- 希帕索斯悖论

  问：边长为1的正方形，其对角线长度是多少呢？

  以当时的数学，不能用整数之比表示的。。

  我们今天知道，长度为“根号2”，它是无理数

数学危机到图灵机

- **第一次数学危机的出现**

  - 毕达哥拉斯的学生－希帕索斯（古希腊）

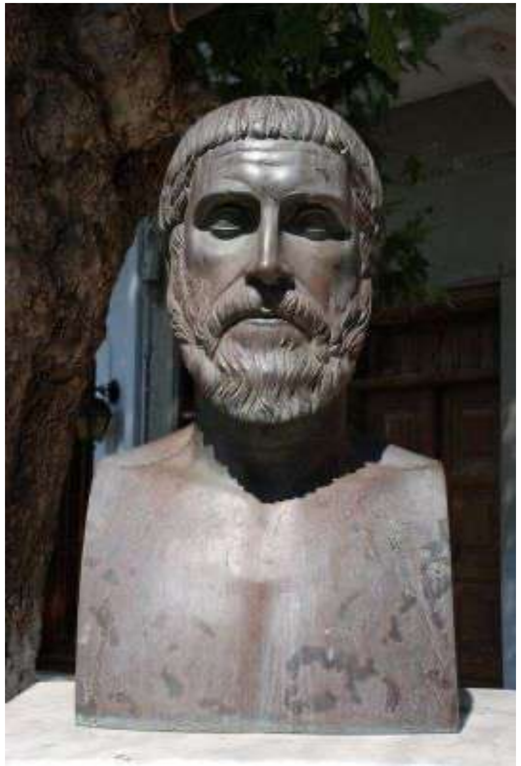
- 第一次数学危机出现
- 今天，希帕索斯被称为“发现无理数第一人”
- 挑战了当时的数学，推翻了毕达哥拉斯学派
- 背叛了老师，被在国外流浪多年
- 最终被毕达哥拉斯信徒扔进了地中海

数学危机到图灵机

- **第一次数学危机的缓解**

  - 欧多克索斯

  古希腊小亚细亚（今土耳其）



- 200年以后，欧多克索斯建立起一套完整的 **“比例论”**，避开了无理数这一逻辑上的丑闻。
- 使得当时的数学得到了形式上的完美，**缓解**了第一次数学危机。
- 解决方法是借助几何方法，通过**避免**直接出现无理数而实现的。
- 体现出几何学的强大，**推动**了几何学的发展。
- 但他并**没有真正地**解决第一次数学危机。

数学危机到图灵机

- **第一次数学危机真正的解决**

法国数学家柯西

- 直到19世纪下半叶，实数理论建立后，无理数本质才彻底清晰，
- 无理数在数学中合法地位确立，圆满地解决了第一次数学危机。

德国数学家克莱因

德国数学家兰道

德国数学家魏尔斯特拉斯

德国数学家戴德金

数学危机到图灵机

- **第二次数学危机：微积分**

牛顿

莱布尼兹

贝克莱

- 17世纪，牛顿和莱布尼兹分别独立发现了微积分，但两人的理论都建立在**“无穷小”**的分析之上。
- 很多人反对“无穷小”，受到了英国哲学家和基督教主教 --- 贝克莱强烈的反对，1734年的书中说到：

无穷小量一会儿是零，一会儿又不是零，到底是什么？嘲笑到无穷小量是“已死亡的幽灵”。

--《分析学家；或一篇致一位不信神数学家的论文；其中审查一下近代分析学的对象，原则及论断是不是比宗教的神秘、信仰的要点又更清晰的表达，或者更明显的推理》（书名很长）

数学危机到图灵机

- **第二次数学危机：贝克莱悖论**

  - 例子：求x^2的导数。

$$\frac{(x+\Delta x)^2-x^2}{\Delta x}$$
$$=2x+\Delta x$$
$$=2x$$

这时候， Δx 作为分母不为0

这时候， Δx 为0

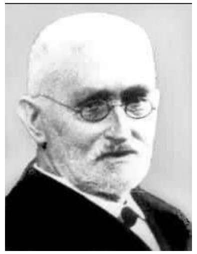
贝克莱

  - 挑起了第二次数学危机！

数学危机到图灵机

- **第二次数学危机的缓解**

法国数学家柯西



德国数学家戴德金

$$f(x) = \sum_{n=0}^{\infty} a^n \cos(b^n \pi x)$$

$$0 < a < 1$$

$$ab > 1 + \frac{3\pi}{2}$$

德国数学家魏尔斯特拉斯

- 第二次数学危机的缓解与第一次数学危机的解决是一起的
- 直到19世纪下半叶，实数理论建立后，建立起**极限论**的基本定理，缓解了第二次数学危机，但没有解决它。
- 发现**新问题**：
  - 魏尔斯特拉斯给出一个处处不可微的连续函数的例子，说明直观及几何思考不可靠，而必须给予严格的概念及推理。**（基于几何学的微积分不可靠）**
  - 推动数学家们更深入地研究数学分析的基础 --- 实数论问题，导致了**集合论**的诞生。

数学危机到图灵机

- **第二次数学危机的解决**

德国数学家康托尔

- 19世纪末，康托尔创立了集合论。当时受到了猛烈的工具，甚至导致他抑郁症。
- 但很快大家发现，从自然数与集合论出发，可以建立起整个数学大厦。**一切数学成果都可以建立在集合论的基础上**。
- 1900年，国际数学家大会上，法国数学家庞加莱：

“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已经达到了”

数学危机到图灵机

- **第三次数学危机**

英国数学家罗素

- 但就在这个时候，英国数学家罗素，讲了一个故事，让大家发现其实集合论也是有问题的。。。

- 一个故事：

塞尔维亚有一位理发师说：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。

问1：你不会理发，你可以找他来理发吗？
答1：可以。
把人分为两个集合：一个是会给自己理发，另一个是不会给自己理发。

问1：他要不要给自己理发呢？
如果给自己理发，他就不属于会给自己理发的集合。如果他不给自己理发，他到底属于哪个集合呢？**（罗素悖论）**

数学危机到图灵机

- **第三次数学危机：罗素悖论**

英国数学家罗素

- S由一切不是自身元素的集合所组成，问S是否属于S呢？
  它既不能属于，也不能不属于。。
- 这时候，大家发现，集合论也不是完美的。
- 德国数学家费雷格：

*一位数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。*

数学危机到图灵机

- **第三次数学危机的解决**
  - 哥德尔不完备性定理（1931年）

美国数学家哥德尔

- 任何一个数学系统，只要（1）它是从有限的公理和基本概念推导出来的，并且（2）从中能推导出自然数学系统。就可以在其中找到一个命题，对于它我们**即没有办法证明，又没有办法推翻**。
- 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**。

数学危机到图灵机

- **接下来的研究问题是？**
  - **可计算问题**：
  - 既然没有完美的系统，有任务是可以计算的，有的是不可以的，那哪些是可以计算的任务呢？？
  - 设函数f的定义域是D，值域是R，如果存在一种算法，对D中任意给定的X，都能计算出f(X)的值，则称函数f是可计算的。
  - 研究思路：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。不能完成的任务，就是不可计算问题。计算模型是一个**评价器**！

数学危机到图灵机

- **图灵提出评价模型：图灵机**

英国数学家、计算机科学家
艾伦.图灵

- 1936年，发表《论可计算数载判定问题中的应用》提出了一种理想的计算机的数学模型 --- 图灵机 Turing Machine
- 图灵机是当今计算机的基本原理模型，为了纪念他，美国计算机协会（ACM）于1966年设立了图灵奖（计算机领域的诺贝尔奖）

数学危机到图灵机

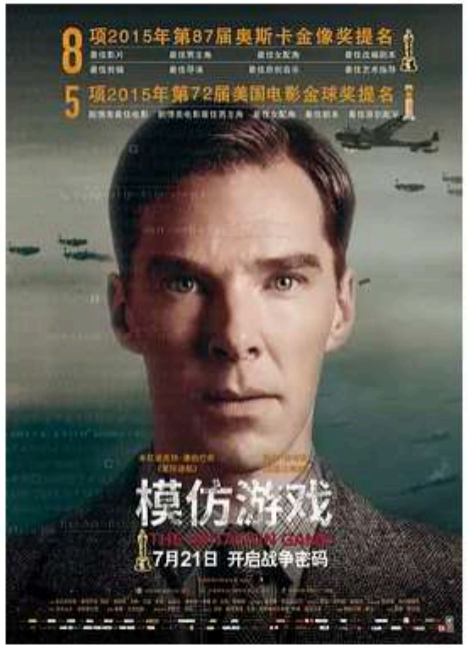
- **图灵本人**

英国数学家、计算机科学家
艾伦.图灵

- 1912年6月，生于伦敦
- 中学期间，获得国王爱德华六世数学金盾奖章
- 1935年，23岁时被选为剑桥大学国王学院院士
- 1936年，24岁时提出“图灵机”，发表于《伦敦数学会文集》
- 1938年，美国普林斯顿大学获得博士学位
- 1938-1945年二战期间，密码破译工作，英美密码破译部门总顾问
- 1946年战后，获得大不列颠帝国勋章
- 1950年，提出“图灵测试”
- 1950年10月，发表《机器能思考吗》论文，开启了人工智能的研究
- 1951年，被选为英国皇家学会会员，图灵家族第四位会员
- 1952年，40岁时编写出一个国际象棋程序
- 1954年，英年早逝

No.176 豆瓣电影Top250

模仿游戏 The Imitation Game (2014)



导演: 莫滕·泰杜姆

编剧: 格拉汉姆·摩尔 / 安德鲁·霍奇斯

主演: 本尼迪克特·康伯巴奇 / 凯拉·奈特莉 / 马修·古迪 / 罗里·金尼尔 / 艾伦·里奇 / 更多...

类型: 剧情 / 同性 / 传记 / 战争

制片国家/地区: 英国 / 美国

语言: 英语 / 德语

上映日期: 2015-07-21(中国大陆) / 2014-11-14(英国) / 2014-12-25(美国)

片长: 114分钟

又名: 解码游戏(港) / 模拟游戏

IMDb链接: tt2084970

豆瓣评分

8.7

454510人评价

5星 46.2%
4星 42.8%
3星 10.2%
2星 0.6%
1星 0.1%

好于 94% 传记片
好于 97% 同性片

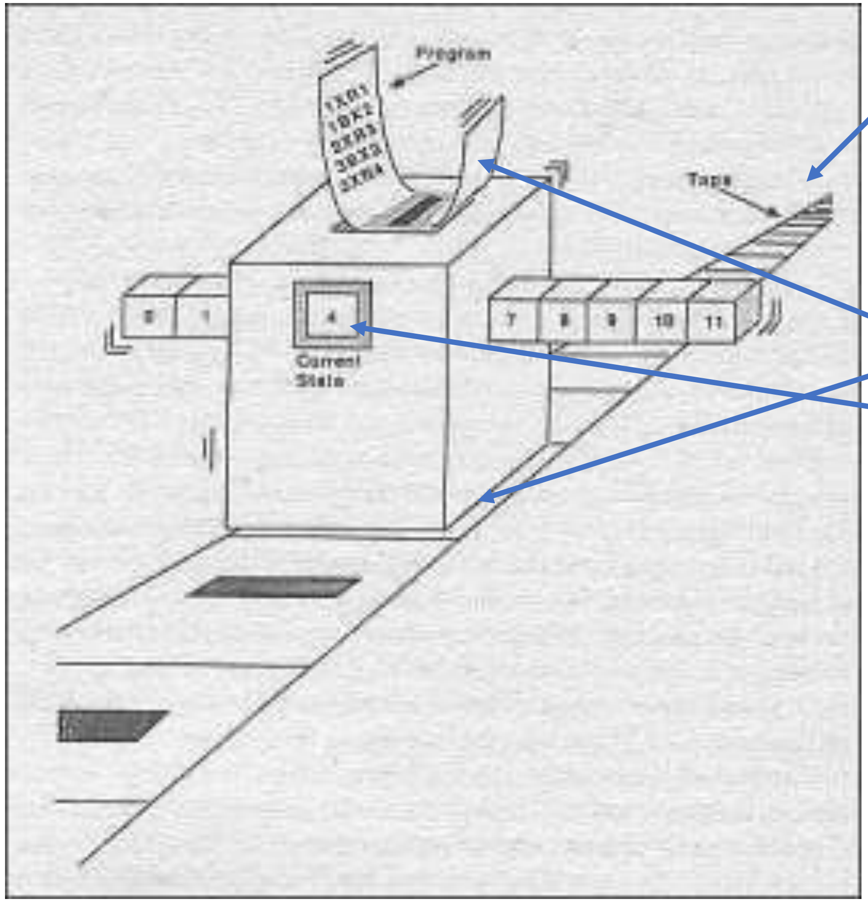
- 数学危机到图灵机
- 图灵机原理
- 数的二进制表示
- 二进制的布尔运算

- 历史上的计算设备
- 从电子管到云计算

- 摩尔定律下的计算危机
- 量子计算

图灵机原理

- 图灵机的**基本构成**

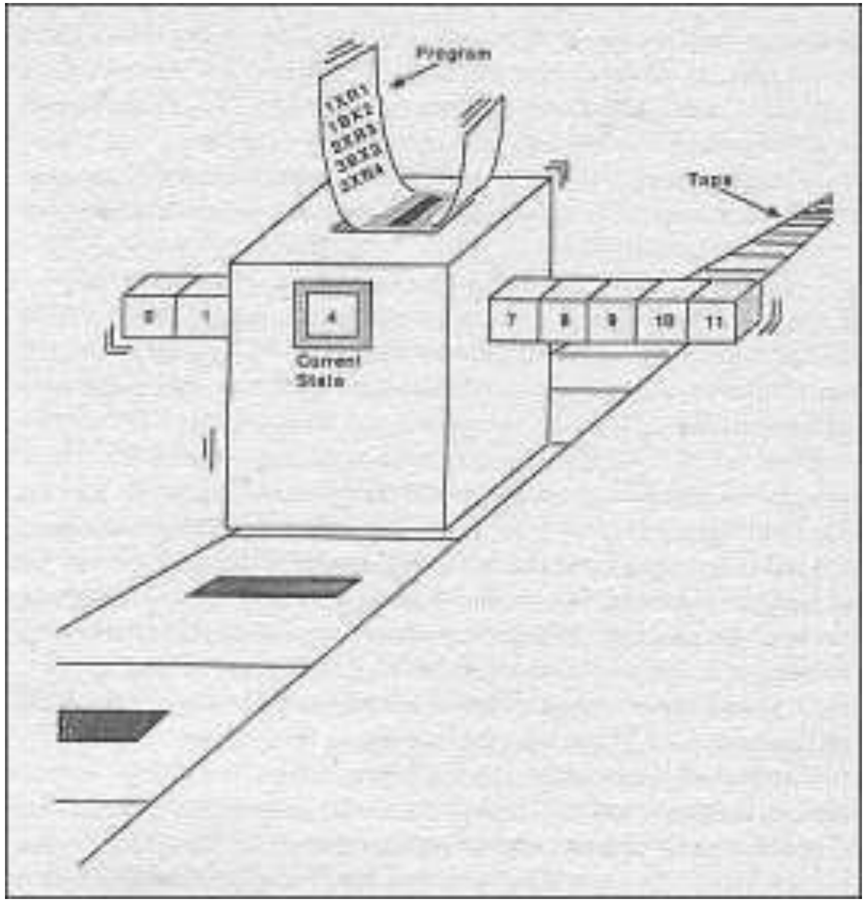


图灵机的结构

- **一条存储带**
  - 无限长
  - 上面有一个个小方格
  - 每个方格可存储一个数字/字母
- **一个控制器**
  - 读写头，可以读/写，更改存储带上的数字/字母
  - 可以接受设定好的程序语句
  - 可以存储当前的自身状态
  - 可以改变自身状态
  - 可以沿着存储带一个个地移动

图灵机原理

- 图灵机的运作机理

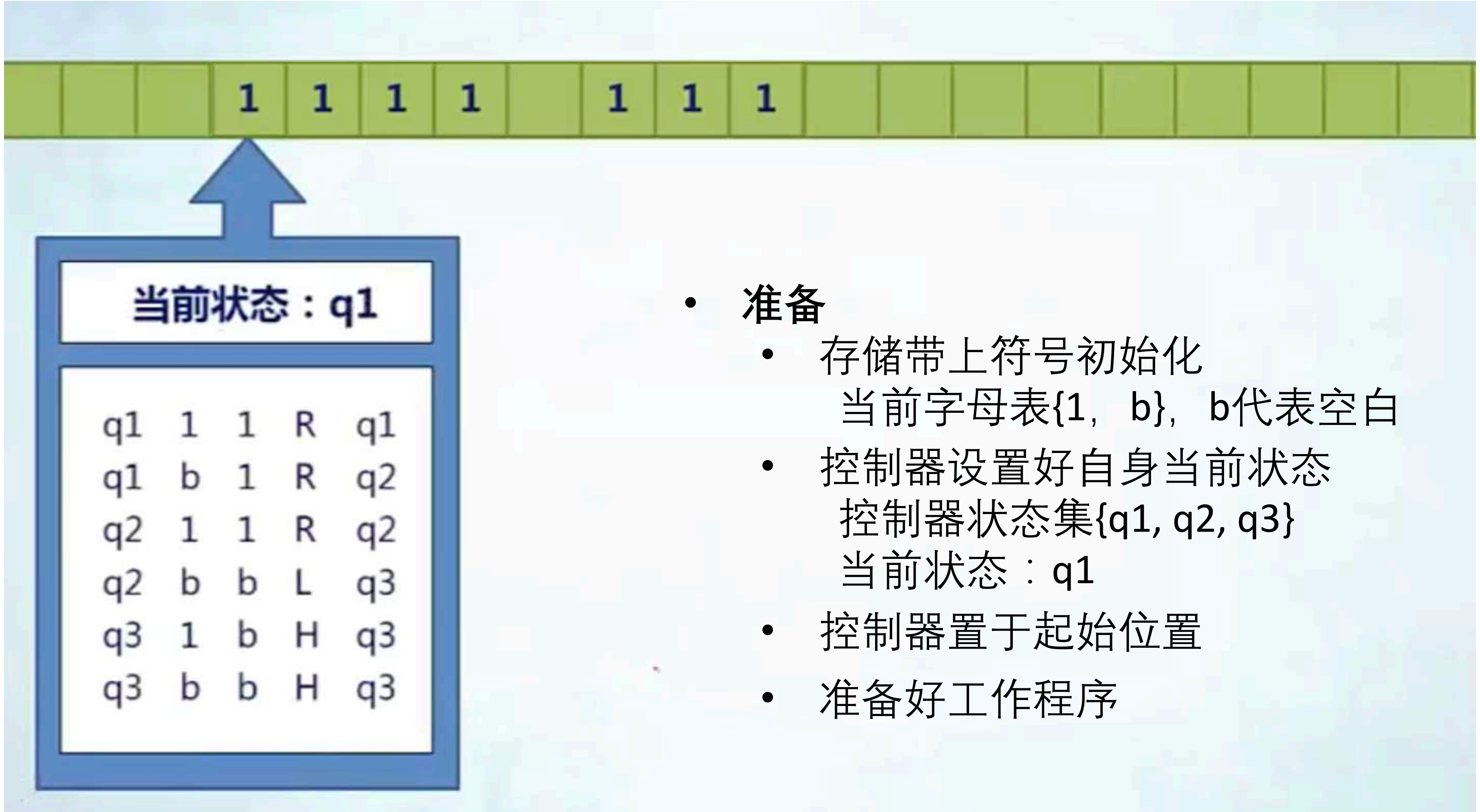


图灵机的工作

- **准备**
  - 存储带上符号初始化
  - 控制器设置好自身当前状态
  - 控制器置于起始位置
  - 准备好工作程序
- **反复执行以下工作直到停机**
  - 1）读写头读出存储带上当前方格中的字母/数字
  - 2）根据自身状态和所读到的字符，找到相应的程序语句
  - 3）根据程序语句，做3个动作：
    - 1. 在当前存储带方格上写入一个相应的字母/数字
    - 2. 变更自身状态至新状态
    - 3. 读写头向左或右移动一步

图灵机原理

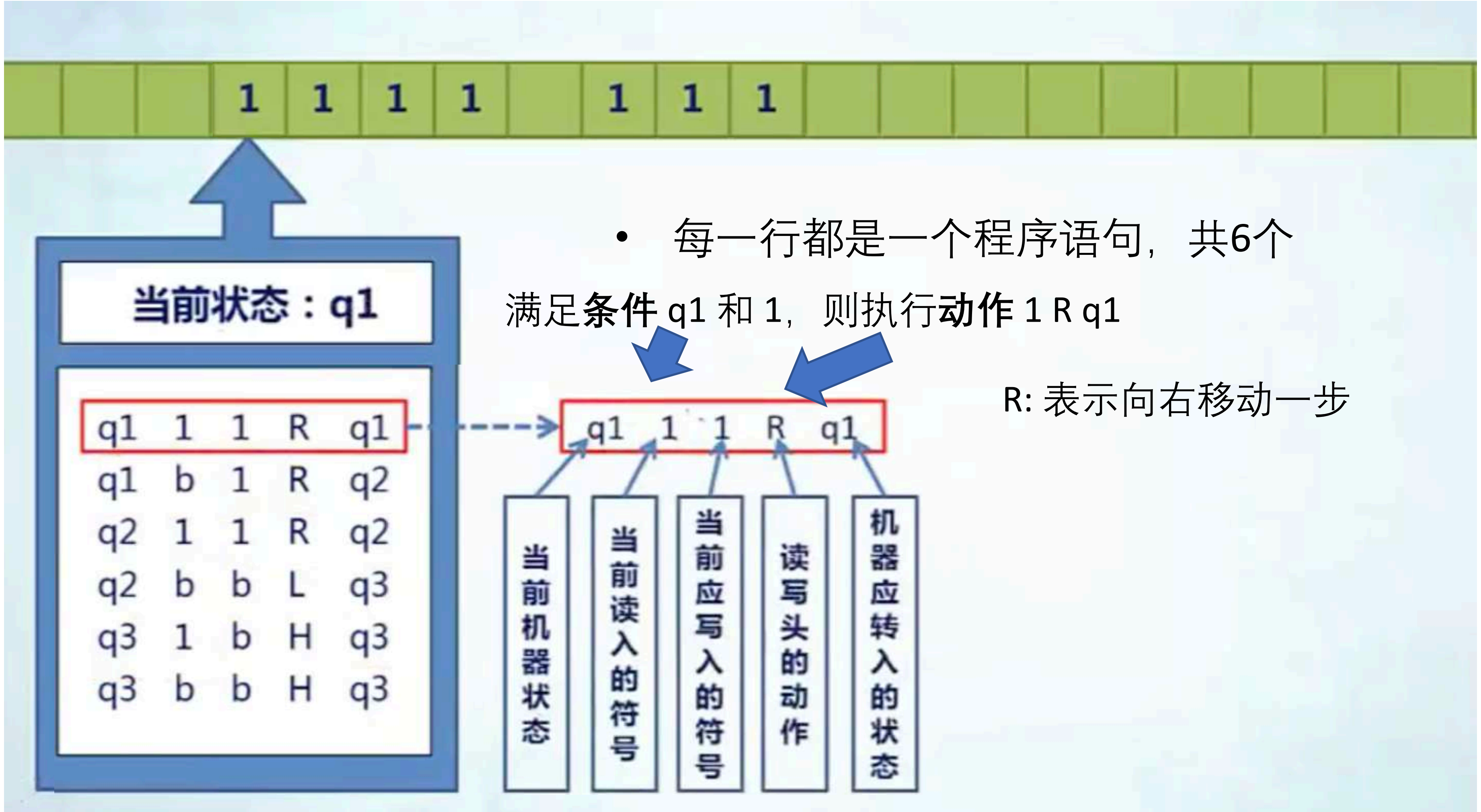
- 图灵机的示范例子



- **准备**
  - 存储带上符号初始化
    当前字母表{1，b}，b代表空白
  - 控制器设置好自身当前状态
    控制器状态集{q1, q2, q3}
    当前状态：q1
  - 控制器置于起始位置
  - 准备好工作程序

图灵机原理

- **图灵机的示范例子**



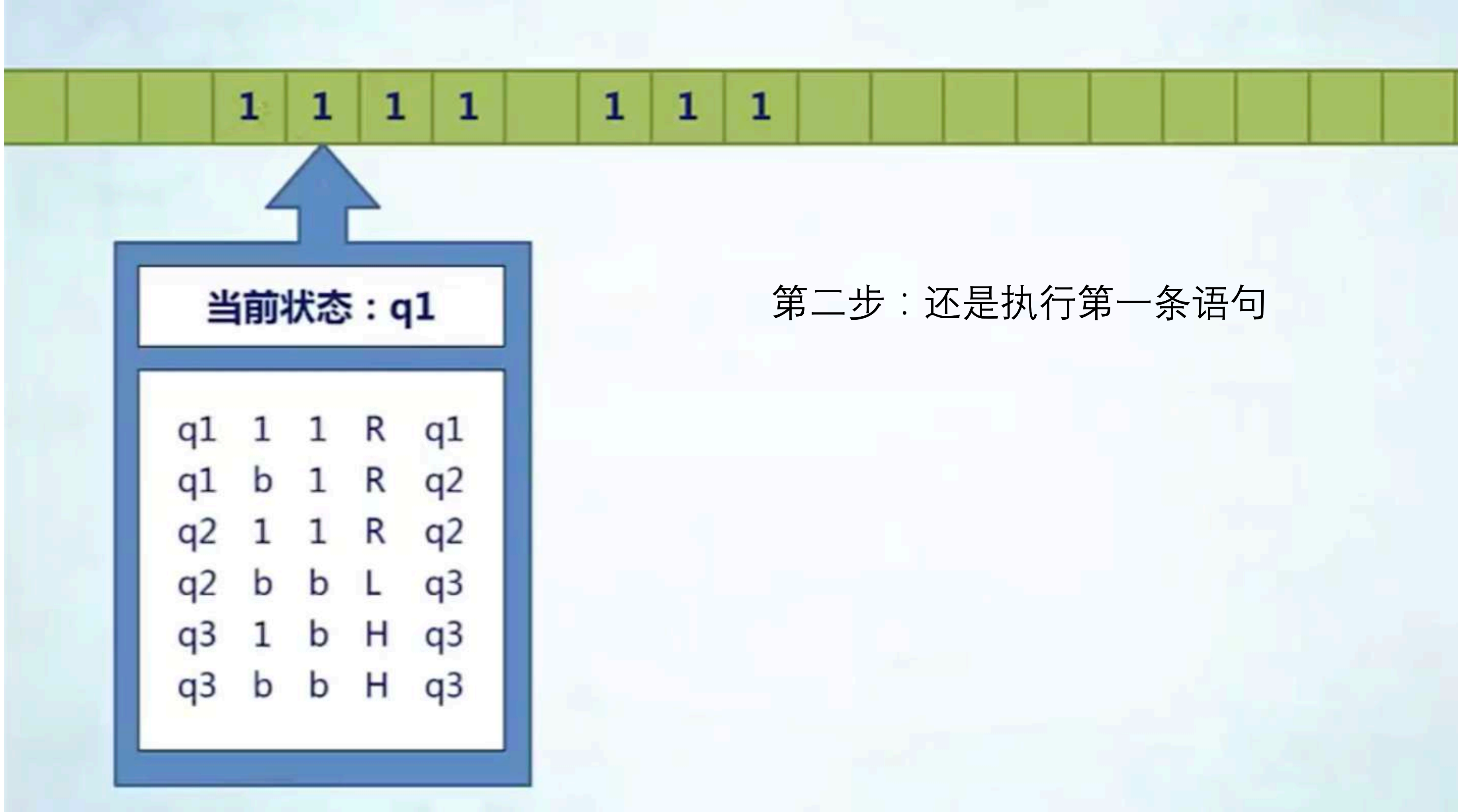
- 每一行都是一个程序语句， 共6个

满足**条件** q1 和 1， 则执行**动作** 1 R q1

R: 表示向右移动一步

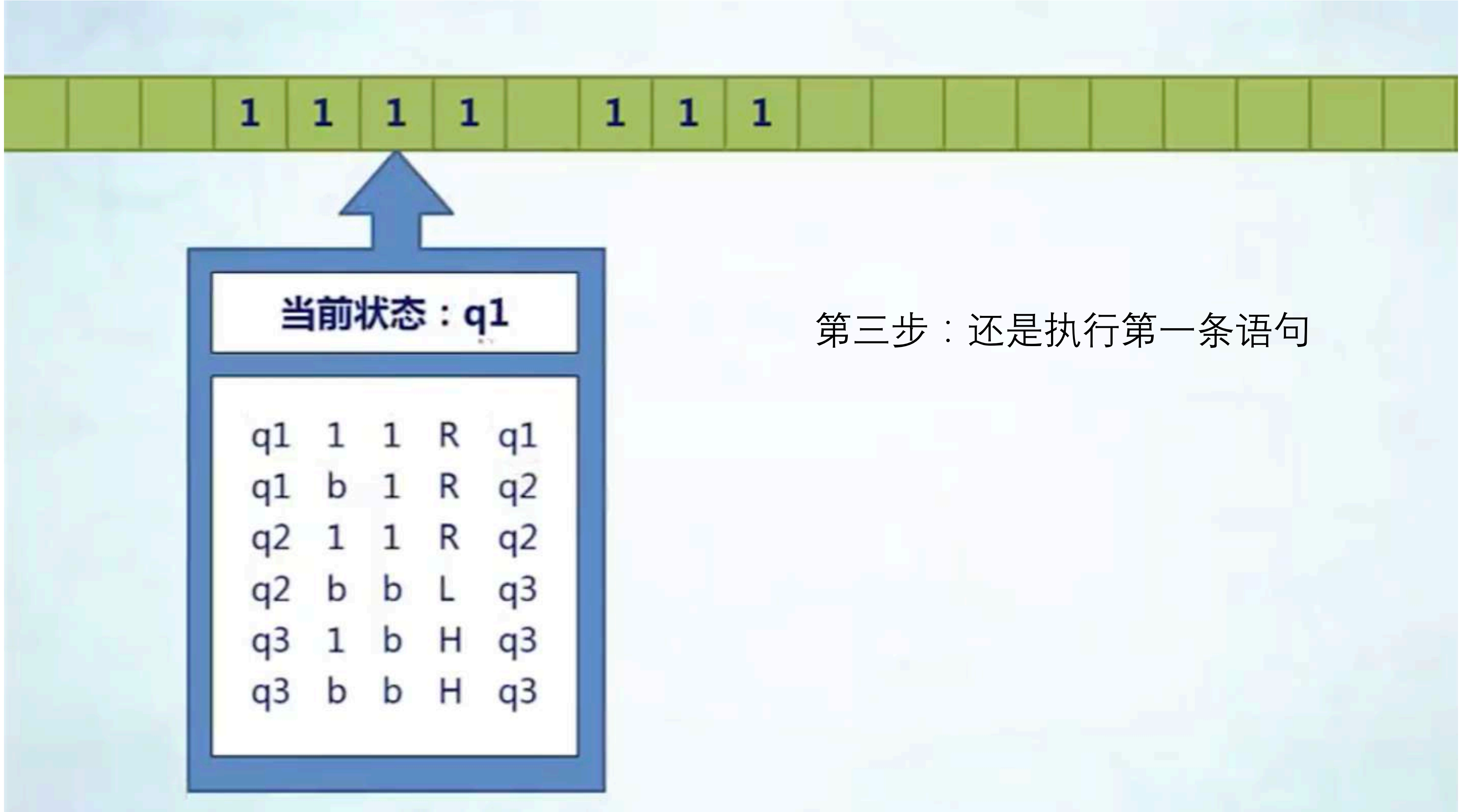
图灵机原理

- **图灵机的示范例子**



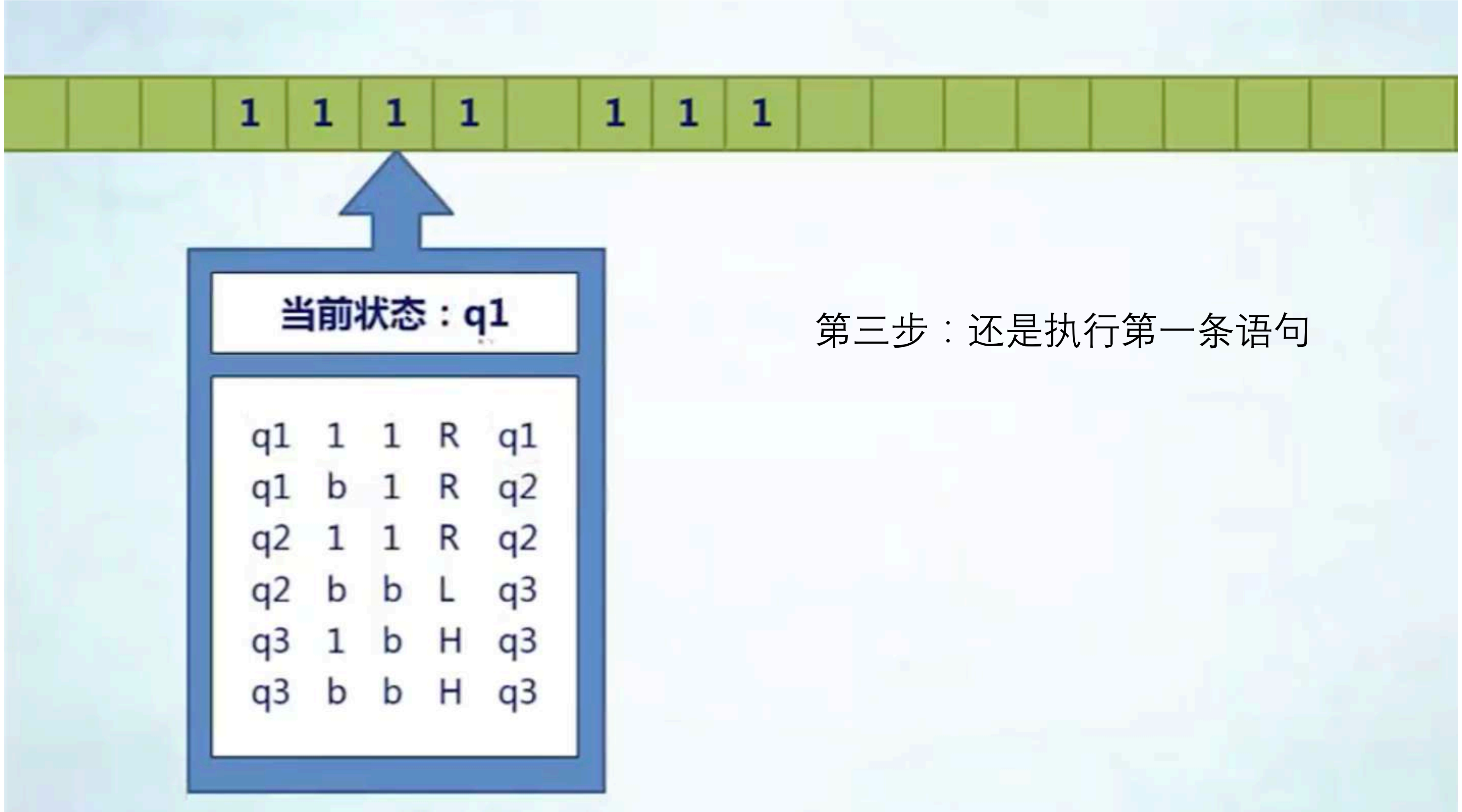
图灵机原理

- **图灵机的示范例子**



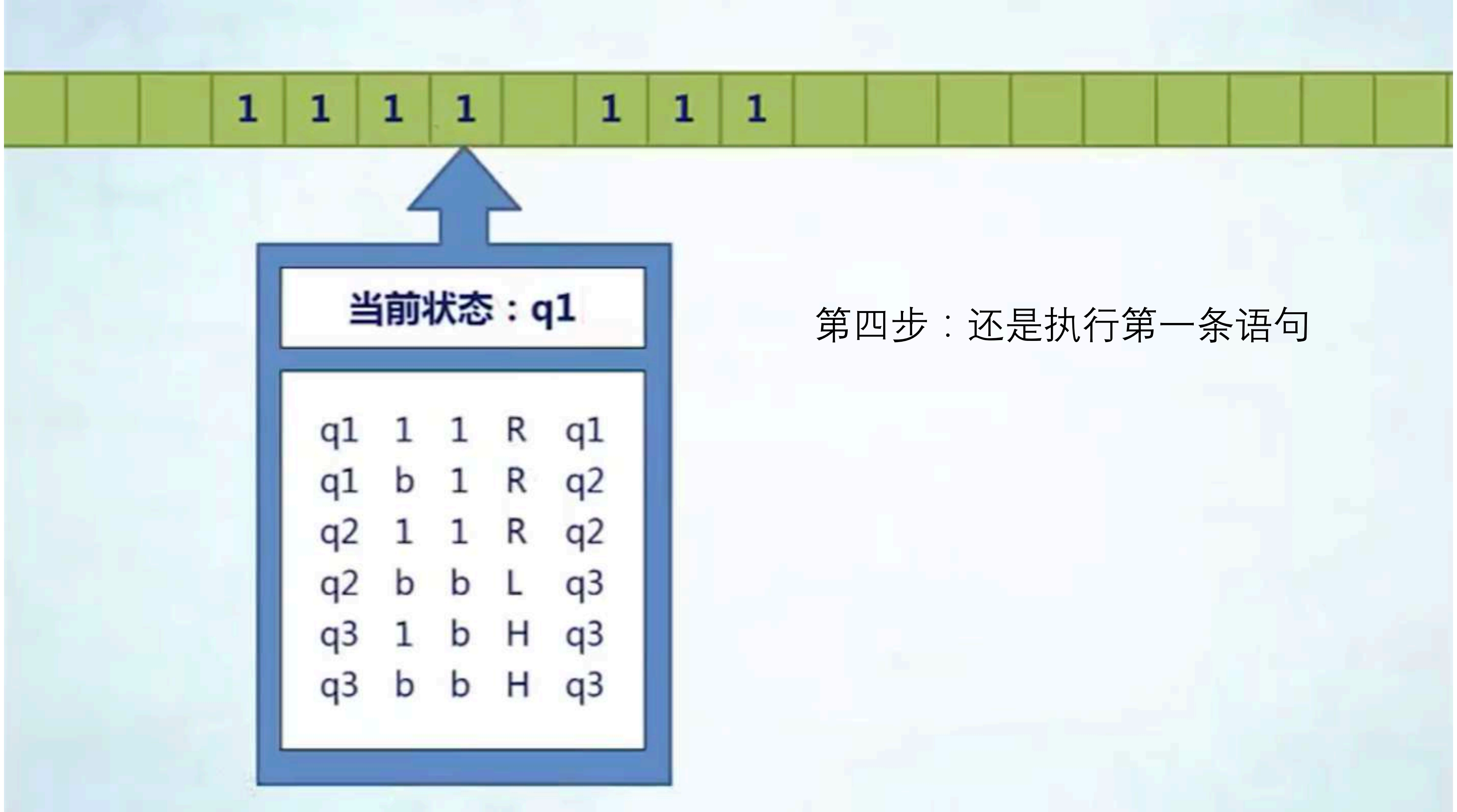
图灵机原理

- **图灵机的示范例子**



图灵机原理

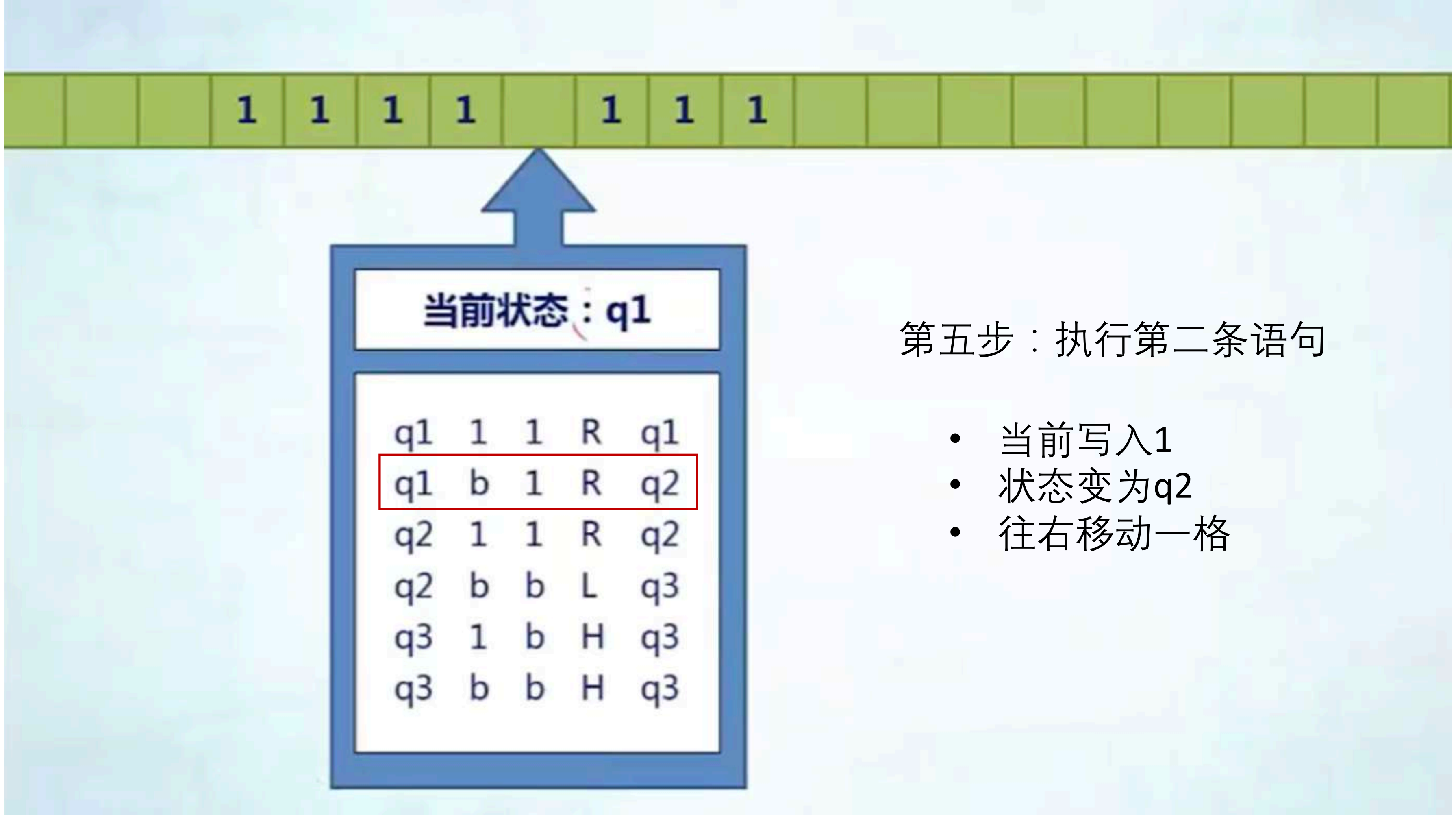
- **图灵机的示范例子**



第四步：还是执行第一条语句

图灵机原理

- **图灵机的示范例子**

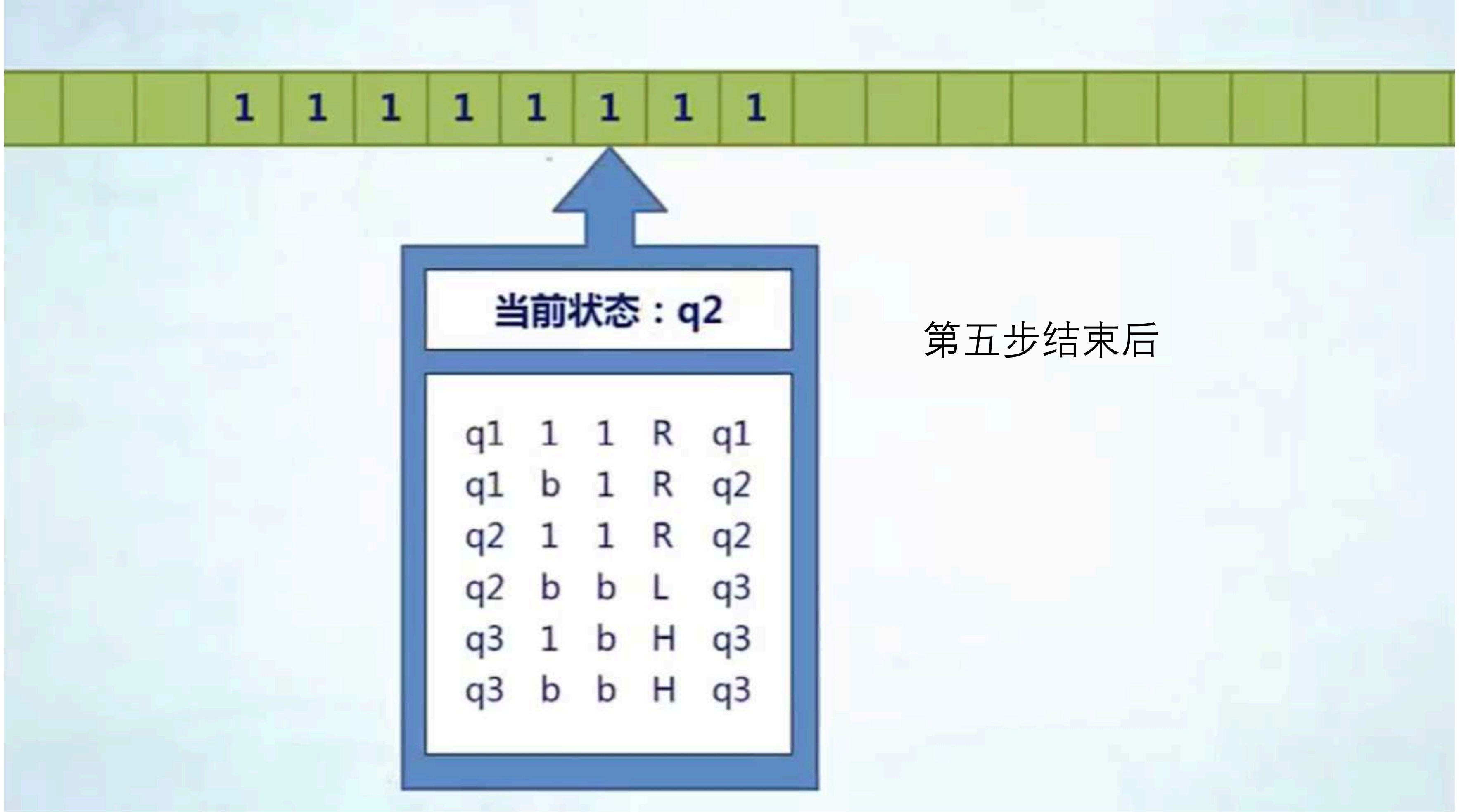


第五步：执行第二条语句

- 当前写入1
- 状态变为q2
- 往右移动一格

图灵机原理

- **图灵机的示范例子**



第五步结束后

图灵机原理

- **图灵机的示范例子**

很多步之后，成功停机了
不再移动了

刚刚做了什么？

4+3 = 7

Margaret Hamilton in 1969, standing next to listings of the software she and her MIT team produced for the Apollo project

图灵机原理

- **图灵机的意义**
  - 判定一个问题是否可计算的
  - 如果能找到存在一台这样的图灵机，则证明问题可计算，反之则反
- **图灵机为什么受到重视**
  - 简单，强大，容易实现

http://aturingmachine.com

图灵机原理

- **图灵机的理论意义**
  - 除了之前说过实现“判定一个问题是否可计算的”更重要的是如下
  - 给出一个可实现的通用计算模型
  - 引入“读写符号”和“状态改变”进行运算的思想
  - 证明了基于简单字母表完成复杂计算的能力
  - 引入存储区、程序、控制器等概念的模型

- 数学危机到图灵机
- 图灵机原理
- 数的二进制表示
- 二进制的布尔运算

- 历史上的计算设备
- 从电子管到云计算

- 摩尔定律下的计算危机
- 量子计算

数的二进制表示

- **问题：计算机为什么能算数呢？**
  - 为了回答“计算机为什么能算数”，我们需要回答3个问题：
    - 1. “数”在计算机中是如何表示的？

      不能表示，则谈何计算？
    - 2. 在逻辑上“数”是如何计算的？
    - 3. 在物理上“数”的计算是如何实现的？

先看第一个问题

数的二进制表示

4 3

1	1	1	1		1	1	1

7

1	1	1	1	1	1	1	

- 刚刚图灵机的例子中，我们用1的个数来表示“数”，并实现了“数的计算”
- 遇到的问题：用1的个数表示数，会有数值爆炸问题，1亿需要1亿个1来表示。做一次1亿的加法，需要把数移动1亿次。
- 采用了太少的符号来表示“数”了
- 如果使用“十进制”，字母表包含11个符号：
  {0, 1, 2, ..., 9, b}

优点：表达数就更容易了

缺点：程序条件会多很多，导致确定当前指令也需要更多的时间了

数的二进制表示

- **需要权衡：**
  - 字母表中的**符号越多**，读入**移动次数减少**，但**程序数量就越多**
  - 字母表中的**符号越少**，**程序量会减少**，但**读入移动次数就越多**

- **最优数量**
  - 科学家提出最优数量是欧拉常数e = 2.718281828 ... 取整后为**3**

  - **苏联**选择了3进制的数表示：-1 0 1

  - 但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

  - **美国**选择了2进制的数表示：0 1

数的二进制表示

- **十进制**

  计算符号：0、1、2、3、4、5、6、7、8、9　　基数：10

  $$256 = 2 \times 10^2 + 5 \times 10^1 + 6 \times 10^0$$

- **二进制**

  计算符号：0、1　　基数：2

  $$10110 = 1 \times 2^4 + 1 \times 2^2 + 1 \times 2^1$$

- **十六进制**

  计算符号：1、2、3、4、5、6、7、8、9、A、B、C、D、E、F　基数：16

  $$ABCD = A \times 16^3 + B \times 16^2 + C \times 16^1 + D \times 16^0$$

数的二进制表示

- **十进制 转 二进制**
  - 十进制123转为二进制数：除以2的商（整除）　　余数

除以2的商（整除）	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

  - 自上而下地依次将余数记录下来，得到结果：1111011

数的二进制表示

- **二进制 转 八进制**

  从右向左，**每3位**进行一次转换

  $1|111|011_{(2)} = 173_{(8)}$

- **二进制 转 十六进制**

  从右向左，**每4位**进行一次转换

  $111|1011_{(2)} = 7B_{(16)}$

已解决问题：数的表示
待解决问题：数的计算

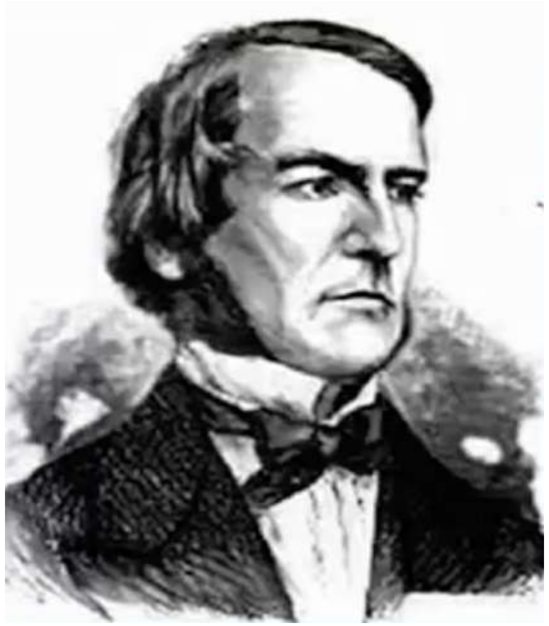
- 数学危机到图灵机
- 图灵机原理
- 数的二进制表示
- 二进制的布尔运算

- 历史上的计算设备
- 从电子管到云计算

- 摩尔定律下的计算危机
- 量子计算

二进制的布尔计算

- **布尔代数**



英国数学家布尔

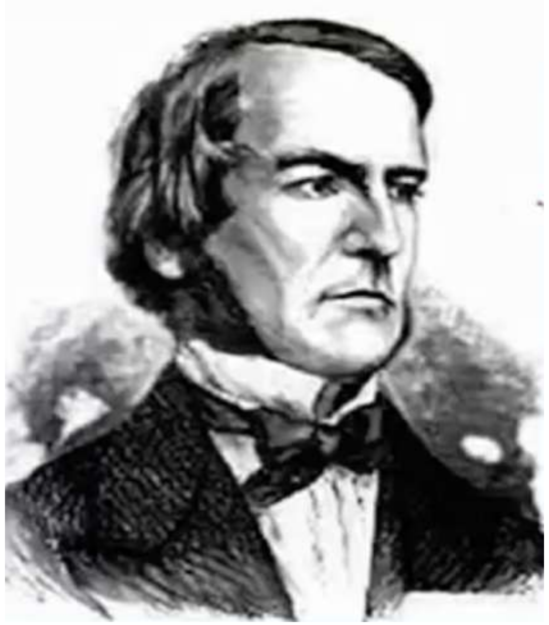
- 1854年，布尔发表《思维规律的研究－逻辑与概论的数学理论基础》并综合其另外一篇文章《逻辑的数学分析》，创立了一门全新的学科：布尔代数
- 为计算机的开关电路设计提供了重要的数学方法和理论基础。
- 十进制中，我们使用：加、减、乘、除，来进行数的计算

在布尔代数中，我们是否也是使用加、减、乘、除呢？

答案是否定的

二进制的布尔计算

- **布尔代数**



英国数学家布尔

- **基本逻辑运算**
  - 与 AND
  - 或 OR
  - 非 NOT
- **复合逻辑运算（由基本逻辑运算符组合而成）**
  - 同或 XNOR
  - 异或 XOR
  - 与非 NAND
  - 或非 NOR
  - 与或非 AND-NOR

二进制的布尔计算

- **基本逻辑运算**
    - **与 AND**
    - 或 OR
    - 非 NOT
- 复合逻辑运算
    - 同或 XNOR
    - 异或 XOR
    - 与非 NAND
    - 或非 NOR
    - 与或非 AND-NOR

- **与运算**：串联电路，两个开关都为1时，才能通电

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

$$F = A \cdot B$$

二进制的布尔计算

- **基本逻辑运算**
  - 与 AND
  - **或 OR**
  - 非 NOT
- 复合逻辑运算
  - 同或 XNOR
  - 异或 XOR
  - 与非 NAND
  - 或非 NOR
  - 与或非 AND-NOR

- **或运算**：并联电路，两个开关只要一个为1时，就能通电

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

$$F = A + B$$

二进制的布尔计算

- **基本逻辑运算**
  - 与 AND
  - 或 OR
  - **非 NOT**
- 复合逻辑运算
  - 同或 XNOR
  - 异或 XOR
  - 与非 NAND
  - 或非 NOR
  - 与或非 AND-NOR

- **非运算**：取反

A	F
0	1
1	0

$$F = \bar{A}$$

二进制的布尔计算

- 基本逻辑运算
  - 与 AND
  - 或 OR
  - 非 NOT
- 复合逻辑运算
  - 同或 XNOR
  - 异或 XOR
  - 与非 NAND
  - 或非 NOR
  - 与或非 AND-NOR

- **同或运算**：两个数相同为1，两个数不同为0

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

$$F = A \odot B$$

二进制的布尔计算

- 基本逻辑运算
  - 与 AND
  - 或 OR
  - 非 NOT
- 复合逻辑运算
  - 同或 XNOR
  - 异或 XOR
  - 与非 NAND
  - 或非 NOR
  - 与或非 AND-NOR

- **异或运算**：两个数相同为0，两个数不同为1

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

$$F = A \oplus B$$

二进制的布尔计算

- 基本逻辑运算
  - 与 AND
  - 或 OR
  - 非 NOT
- **复合逻辑运算**
  - 同或 XNOR
  - 异或 XOR
  - **与非 NAND**
  - **或非 NOR**
  - 与或非 AND-NOR

x_1	x_2	$f(x)$
0	0	1
0	1	0
1	0	0
1	1	0

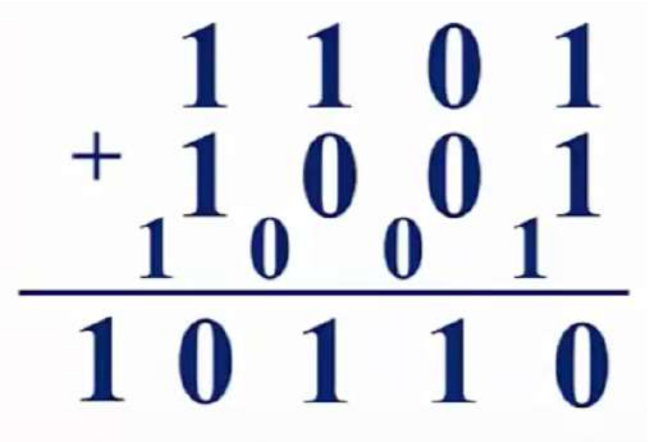
或非运算 NOR

x_1	x_2	$f(x)$
0	0	1
0	1	1
1	0	1
1	1	0

与非运算 NAND

二进制的布尔计算

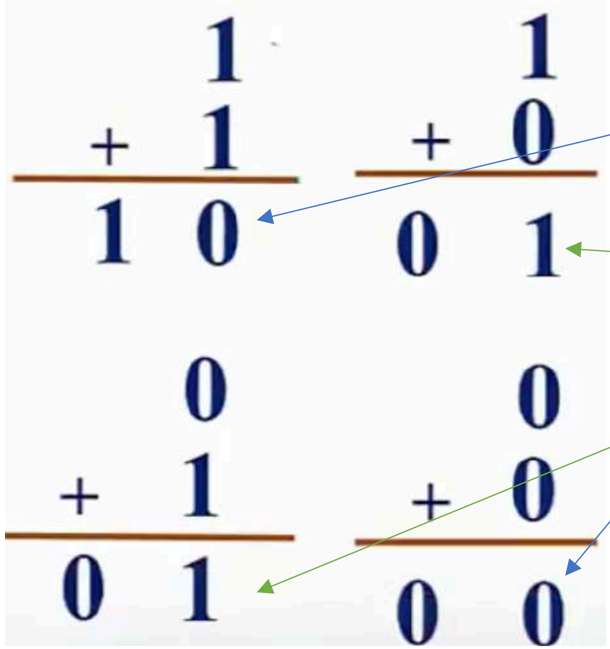
- **二进制的加法**
  - **A = 1101, B = 1001, 求 A + B**



- 通过进位，我们可以很容易得到结果
- 我们还能怎么计算呢？
- 如何用逻辑运算来实现的呢？

二进制的布尔计算

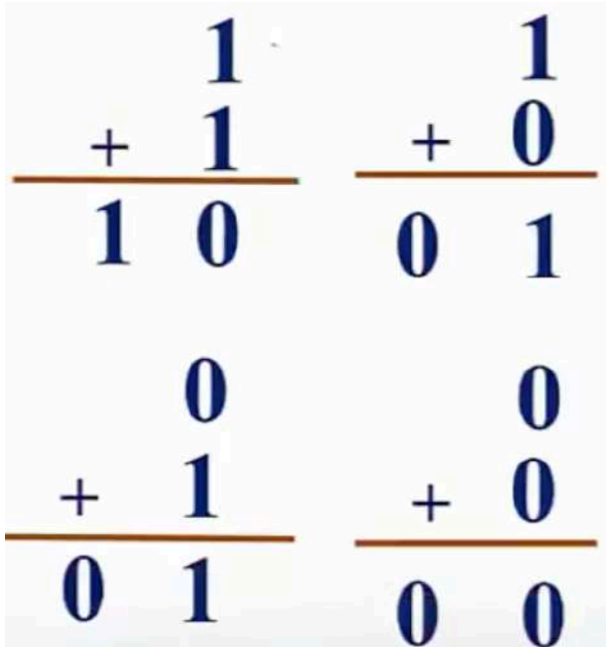
- **二进制的加法**
  - **A = 1101, B = 1001, 求 A + B**



- 根据布尔逻辑，我们左边可以发现：
  - 两个数字只要是一样的（0，0）或者（1，1），本位输出就一定是0
  - 两个数字只要是不一样时，本位输出就一定为1
- 显然，本位计算一个**异或运算**的结果！

二进制的布尔计算

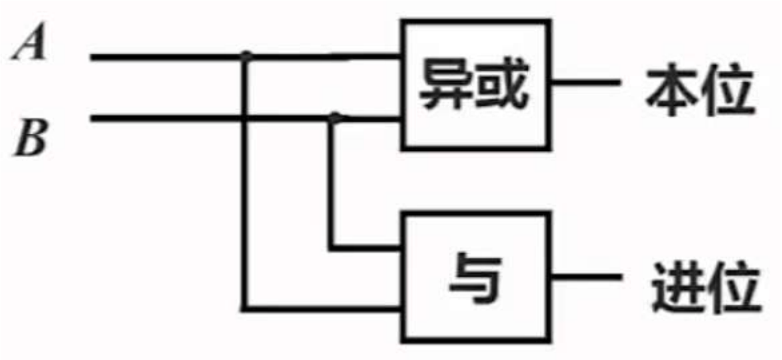
- **二进制的加法**
  - **A = 1101, B = 1001, 求 A + B**



- 根据布尔逻辑，我们左边**还能**发现：
  - 只有两个数字都为1时，进位才是1，否则位0
- 显然，进位计算是一个**与运算**的结果！

二进制的布尔计算

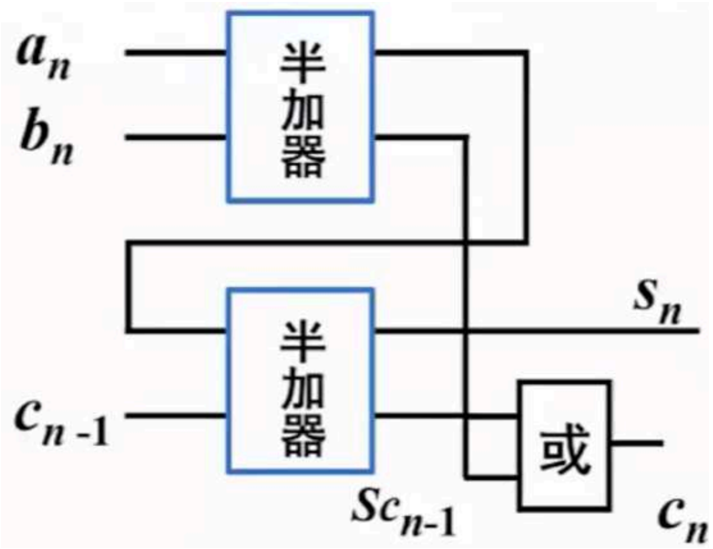
- **二进制的加法**



- 因此，二进制的加法，可以通过异或门电路和与门电路实现！
- 这个实现方式叫 **“半加器”** ，因为它只是计算了进位，但它没有考虑进位，它只能算一半，还不能做完整的加法
- 完成带进位的计算：
  - 两个半加器串联起来即可

二进制的布尔计算

- **二进制的加法**



- **多个半加器串联**起来即可实现带进位的计算：一个半加器计算的结果，作为下一个半加器计算结果的输入。

- 实现了 **"全加器"**

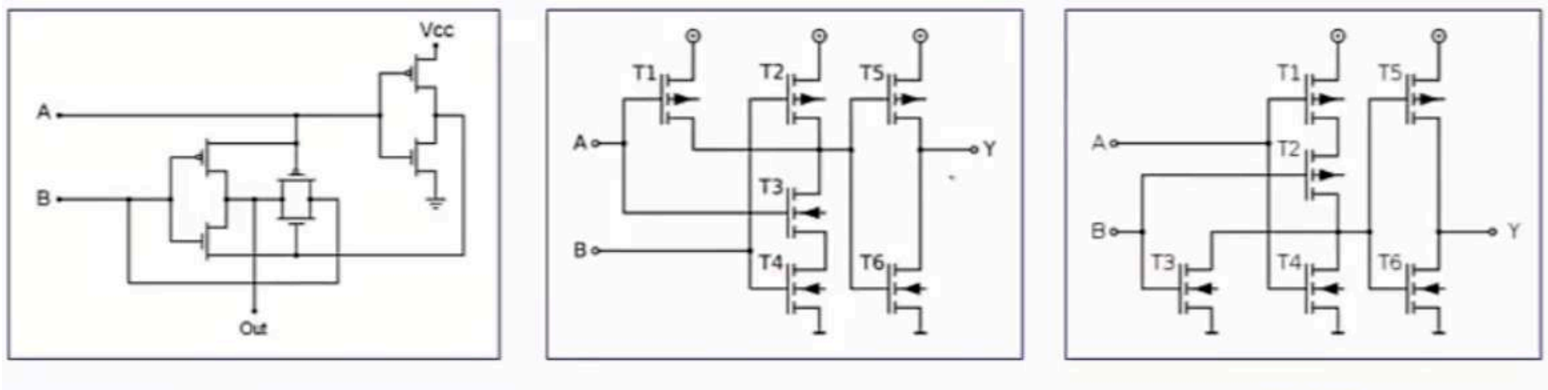
二进制的布尔计算

已解决问题：数的表示－二进制
已解决问题：数的计算－布尔代数
待解决问题：布尔运算如何实现？

二进制的布尔计算

- **布尔代数的硬件实现**

  - 全加器所需要的逻辑电路线路图，如下：



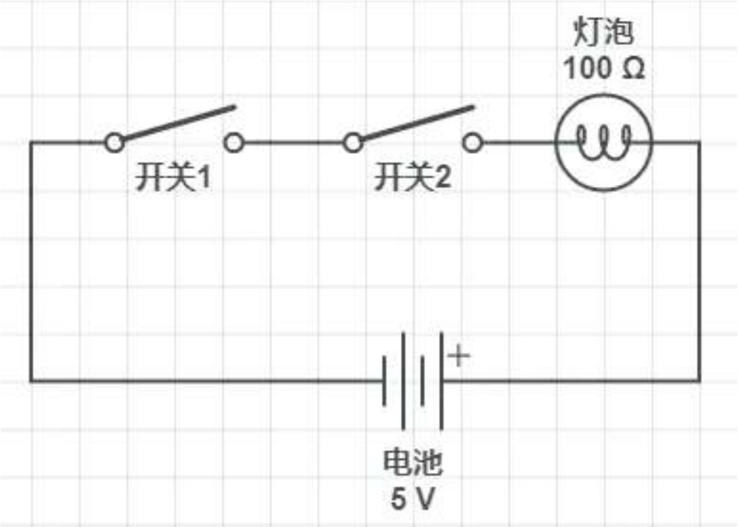
异或门　　与门　　或门

  - 参与运算的数，可以转为二进制表达
  - 二进制表达可以用基本布尔运算实现
  - 基本布尔运算，可以由电路实现

  - 因此：电路能算数！

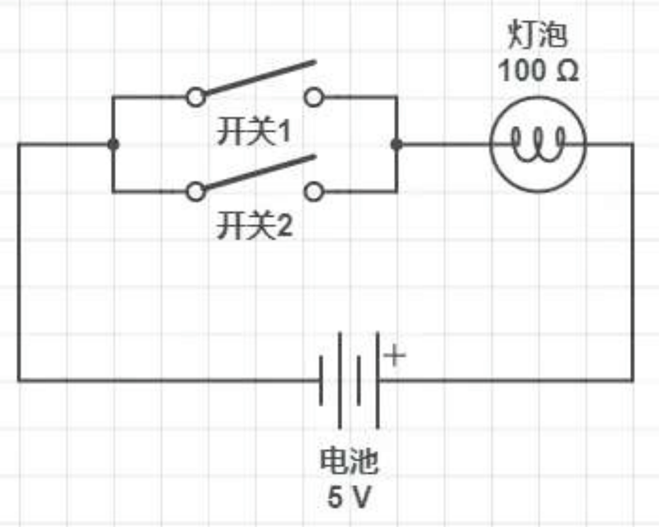
二进制的布尔计算

- **布尔代数的硬件实现**
  - 绝大多数计算设备的最小器件
  - 一切都从一个开关开始



两个开关实现“与门”操作

开关1	开关2	灯泡
0	0	0
0	1	0
1	0	0
1	1	1



两个开关实现“或门”操作

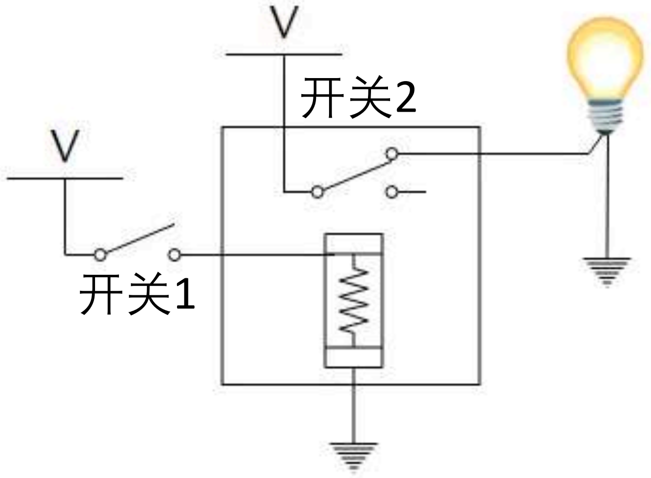
开关1	开关2	灯泡
0	0	0
0	1	1
1	0	1
1	1	1

  - 0表示开关断开，1表示开关闭合
  - 0 表示灯泡灭，1表示灯泡亮

二进制的布尔计算

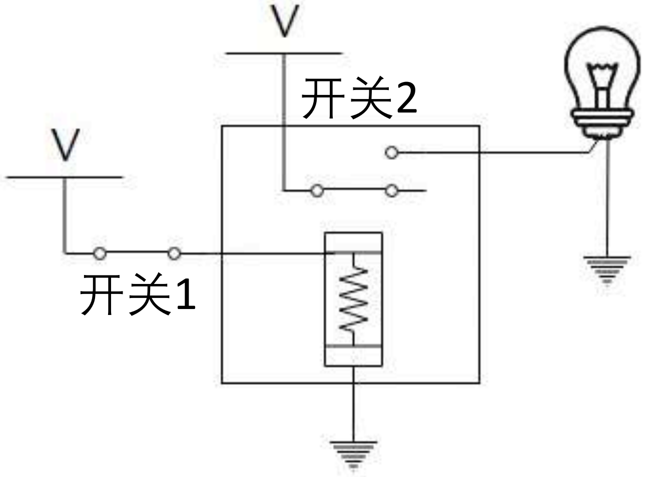
- **布尔代数的硬件实现**
  - 绝大多数计算设备的最小器件
  - 一切都从一个开关开始

  - 如果有一个可控开关
    - 比如继电器
    - 开关1控制开关2



开关1	灯泡
0	1
0	1
1	0
1	0

可控开关实现反相器（Inverter）



二进制的布尔计算

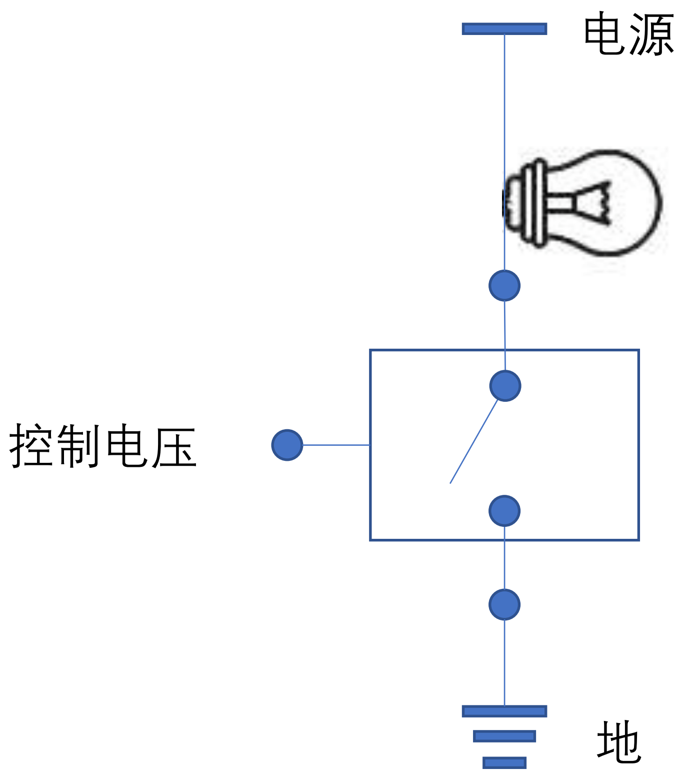
- **布尔代数的硬件实现**
  - 绝大多数计算设备的最小器件
  - 一切都从一个开关开始

  - 如果有一个可控开关
    - 控制电压
    - 低电平（0）闭合开关，高电平（1）断开开关

控制电压	灯泡
0	1
0	1
1	0
1	0

可控开关实现反相器（Inverter）

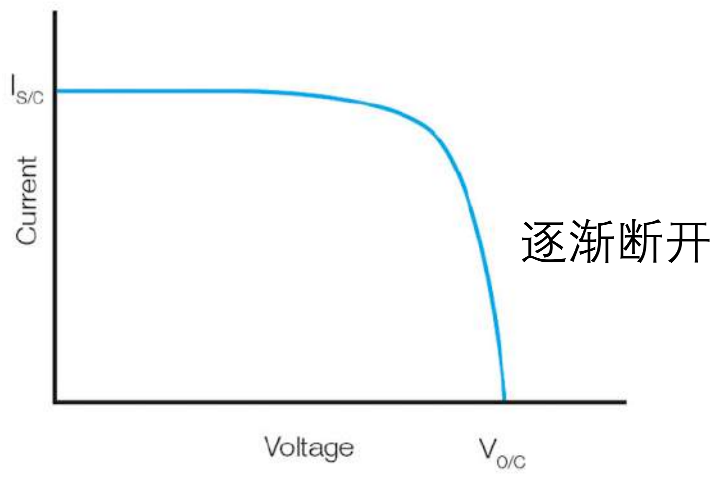
如果我们把它抽象成一个符号



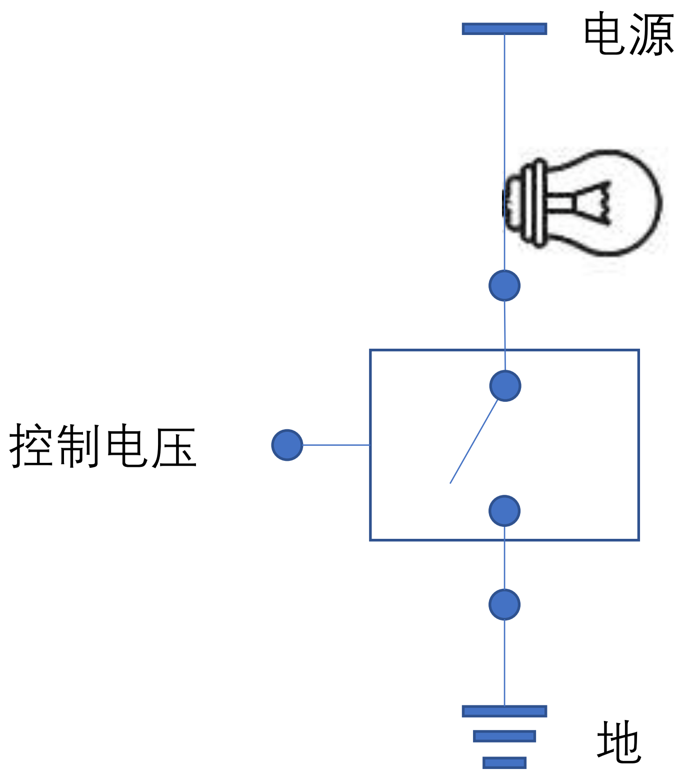
二进制的布尔计算

- **布尔代数的硬件实现**
  - 绝大多数计算设备的最小器件
  - 一切都从一个开关开始

  - 如果有一个可控开关
    - 控制电压
    - 低电平（0）闭合开关，高电平（1）断开开关
    - 实际开关器件并非那么理想



如果我们把它抽象成一个符号

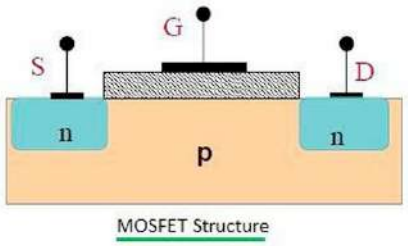


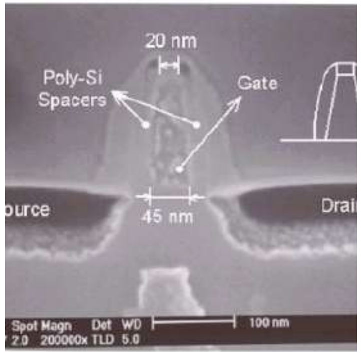
二进制的布尔计算

- **布尔代数的硬件实现**
  - 绝大多数计算设备的最小器件
  - 一切都从一个开关开始

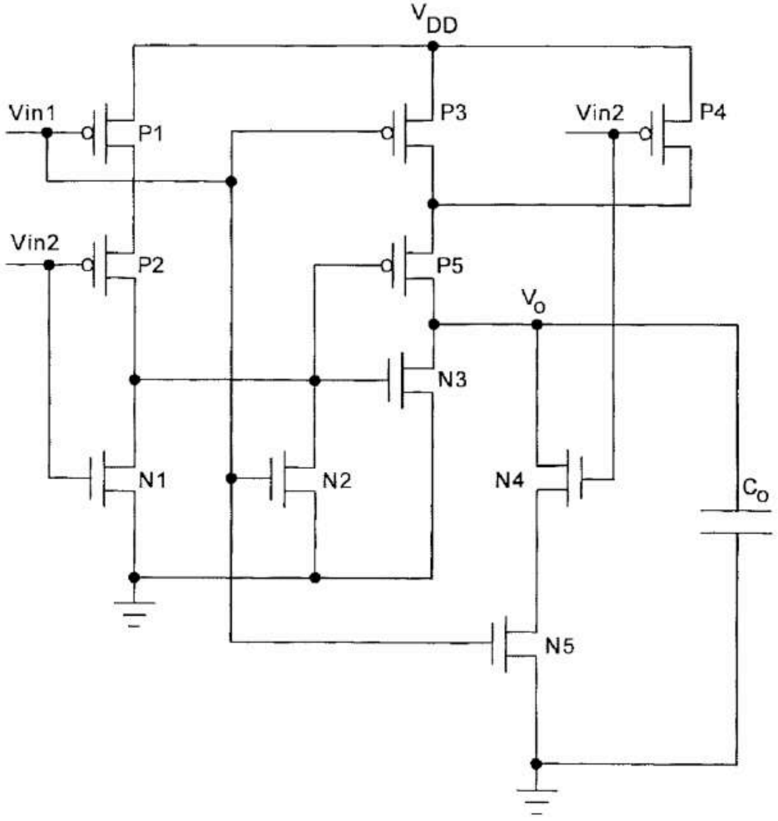
  - 如果有一个可控开关
    - 组合开关可以实现逻辑运算

分立的三极管器件

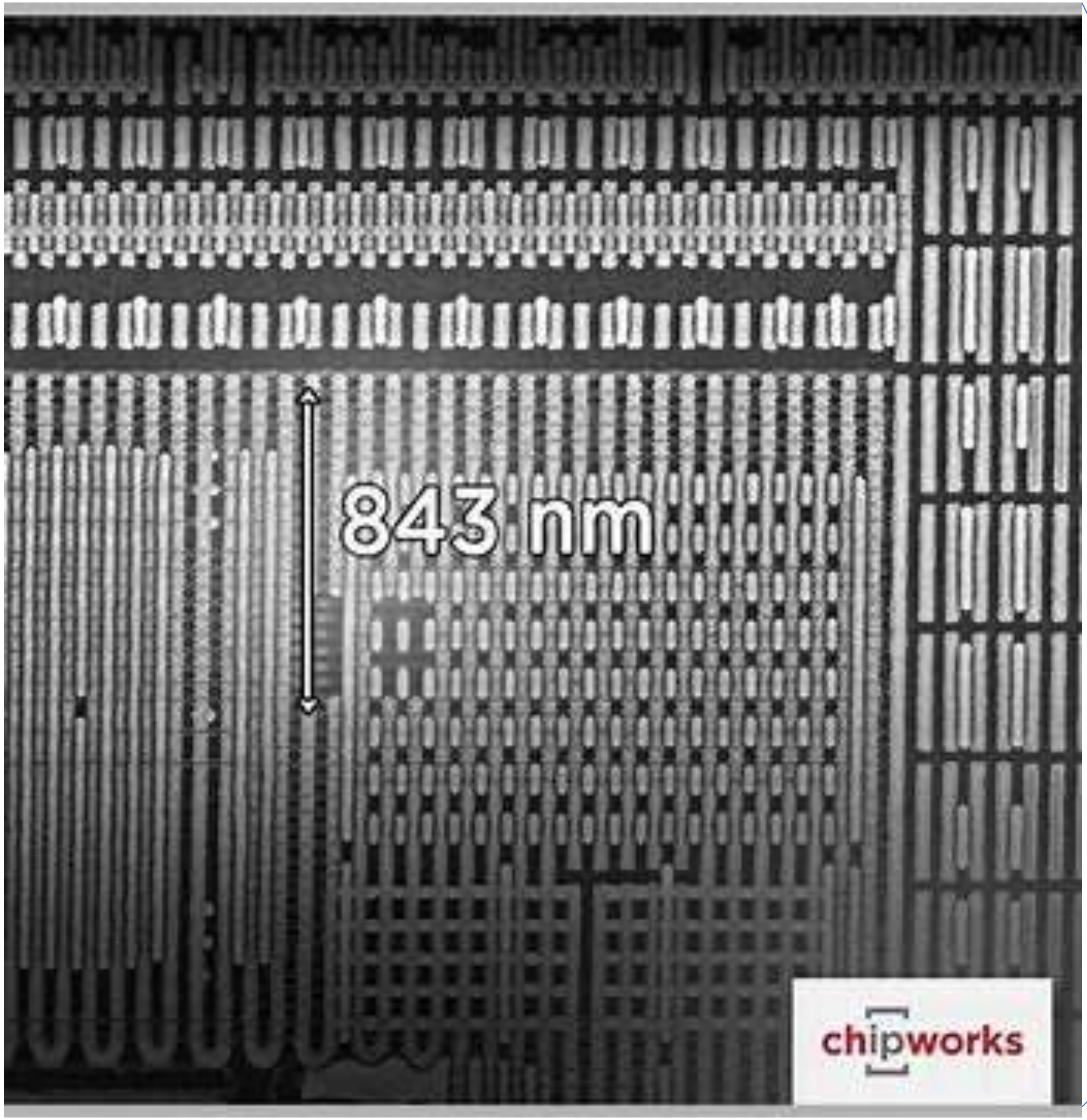




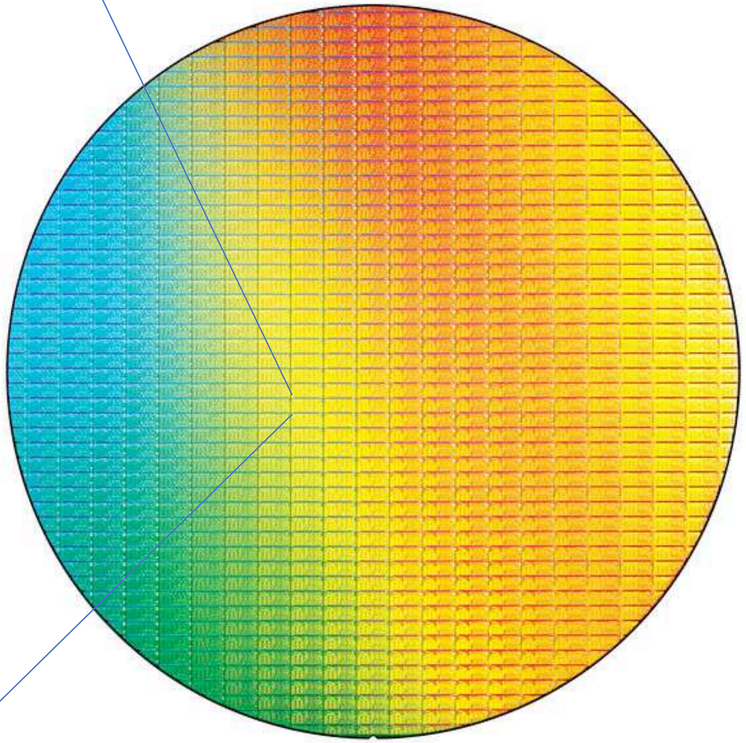
集成电路芯片中的晶体管



Vo = f(Vin1, Vin2)
猜猜f是什么运算



显微镜下的Intel CPU 14nm



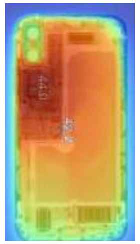
晶圆（Wafer）

Takeaways……

二进制 → 表示数

二进制 → 计算

开关 → 组成电路 → 二进制计算



手机又热又慢

想要算得更快、更省电、更复杂？

做更好的开关 …	微电子
做更高效的电路 …	集成电路设计
做更优秀的算法 …	计算机

各个领域交叉融合，无明显界限

人类文明的结晶

北京大学 PEKING UNIVERSITY

- 数学危机到图灵机
- 图灵机原理
- 数的二进制表示
- 二进制的布尔运算

现在

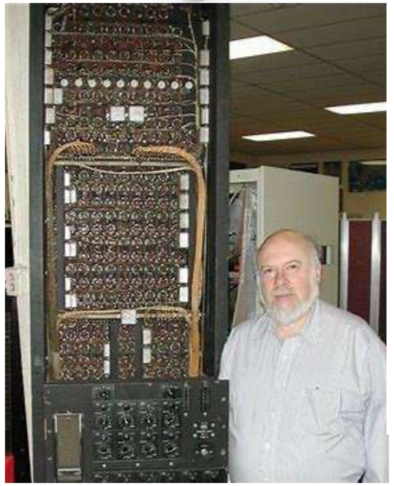
- 历史上的计算设备
- 从电子管到云计算

- 摩尔定律下的计算危机
- 量子计算

历史上的计算设备

- **早期计算机**
  - 手工计算器，1200 ~ 1600年
  - 机械计算器，1600 ~ 1930年
  - 计算机原型，1937 ~ 1946年

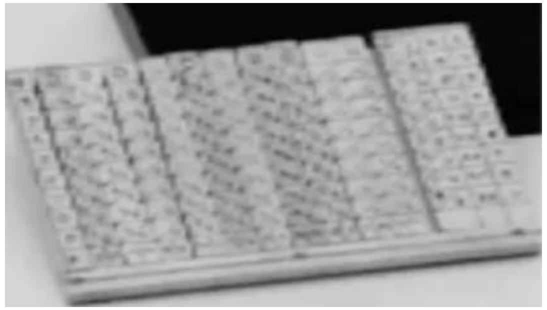
- **现代计算机**
  - 电子管计算机，1946年
  - 晶体管计算机，20世纪50年代后期
  - 集成电路计算机，1965年
  - 超大规模集成电路，20世纪70年代早期
  - … 个人电脑 … 云计算 …



历史上的计算设备

- **手工计算器**

算盘
古中国
公元前600年



英国Napier
乘除器
1617

英国Oughtred
计算尺
1621

- **作用**
  - 标记计算的过程
  - 记录计算的结果
  - 进行计算的辅助工具

- **缺点**
  - 不能记录计算法则
  - 不能设定计算步骤
  - 因此：

  我们把它们称为计算器，而不是计算机

历史上的计算设备

- **帕斯卡加法器（1642年）**

- **帕斯卡 法国数学家、物理学家、哲学家、散文家**
  - 12岁发现三角形内角和等于180度
  - 16岁参加巴黎数学家和物理学家小组
  - 17岁写成《圆锥截线论》震惊笛卡尔
  - 18岁开始设计计算器，帮助父亲计算税率
  - 19岁第三个模型在1642年研制成功！
  - ... 帕斯卡是压强单位 ...

- **普遍认为的第一台机械计算器**
  - 一种齿轮组成的装置，依靠发条转动，用专用的铁笔拨动转轮以输入数字
  - 以手为动力
  - 初期版本只能做6位加减法

历史上的计算设备

- **1958年考古发现了比帕斯卡更早的机械计算器：契克卡德计算机（1623年）**

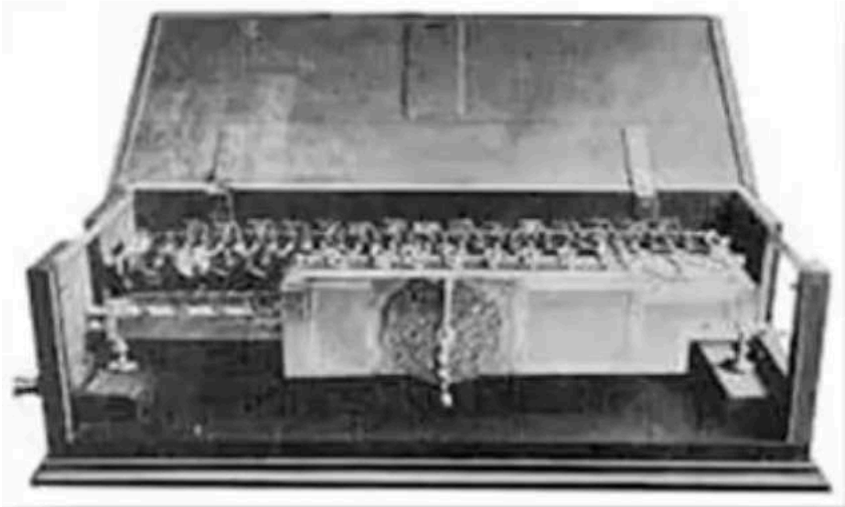
- **契克卡德 德国**
  - 德国图宾根大学教授
  - 1623年，为朋友 -- 天文学家开普勒制作了一种机械计算机
  - 可以进行6位数加减法，具有“溢出”装置
  - 附加一套圆柱形“纳皮尔算筹”可以进行乘除运算！

历史上的计算设备

- **帕斯卡加法器的改进（1673年）**

莱布尼兹

- **莱布尼茨 德国数学家 1647～1716**
  - 和牛顿先后独立发现微积分
  - 提出了二进制概念
  - 1673年，在帕斯卡加法器上，实现了连续运算（帕斯卡只能一次次算），可做乘除法，能进行四则运算的机械计算器，轰动欧洲
  - 计算结果可以达到16位



历史上的计算设备

- **工业革命时代：差分机**

巴贝奇

- **1822年英国巴贝奇制造出第一台差分机**
  - 能提高乘法速度和改进对数表等数字表的精度
  - 处理3个不同的5位数，计算精度达6位小数
- **巴贝奇差分机：机械的极限在哪里？**
  - 巴贝奇是资本家，有足够的资金
  - 第一台差分机的建造用了10年
  - 他向英国政府申请经费建造更强大的差分机：运算精度为20位

历史上的计算设备

- **工业革命时代：差分机**

巴贝奇

阿达.奥古斯塔

- **1834年，更厉害的差分机：巴贝奇提出“分析机”概念**
  - 不仅仅是能够制表的差分机，而是一种通用数学计算机
  - 100个变量的复杂算题，每个数可达25位
  - 第一次将机器分为：堆栈、运算器、控制器
  - 第一次企图用蒸汽动力实现计算过程
  - 第一次程序和数据存储于穿孔卡片上（一直沿用到1970年代）
  - 第一个程序员：阿达.奥古斯塔
    为分析机编制了历史上第一批计算程序
  - 但这第二台机器的建造也用了10年，失败了
    - 雇用了大量的工程师
    - 主要零件的误差达不到每英寸千分之一
    - 该机器目前放在伦敦皇家学院博物馆供人观赏
    - 但意义重大

历史上的计算设备

- **工业革命时代：美国的崛起 霍列瑞斯**

赫曼·霍列瑞斯

- **霍列瑞斯 美国统计学家 数据处理之父 德国侨民**
  - 1880年，美国进行全国人口普查，为当时5000万美国人登记
  - 手摇计算机太慢，1880年的人口普查需要到1887年才能完成
  - 1890年的普查要到1900年
  - **霍列瑞斯制表机**（电子穿孔卡片汇总）应运而生
  - 6个月完成了1890年的数据汇总
- **霍列瑞斯 成立 专业制表公司（1896年）**
  - 经营不善，几次被收购改组
  - 最终在1924年发展为国际商用机器公司（IBM）
- **IBM 的 IBM601穿孔卡片式计算机（1935年）**
  - 能在一秒钟内计算乘法运算

历史上的计算设备

- **电器革命时代：二战末期的德国**

- **楚泽 德国工程师 数字计算机之父**
  - 1934年开始研制Z1、Z2、Z3三种型号的计算机
  - 采用二进制运算
  - 以继电器为主要元件

- **Z3型号完成（1941年）**
  - 第一台**可编程**的电子计算机
  - 可处理7位指数，14位小数
  - 使用了大量的**真空管**
  - 每秒钟能进行3～4次加法运算
  - 3～5秒一次乘法运算（速度比IBM601慢）

历史上的计算设备

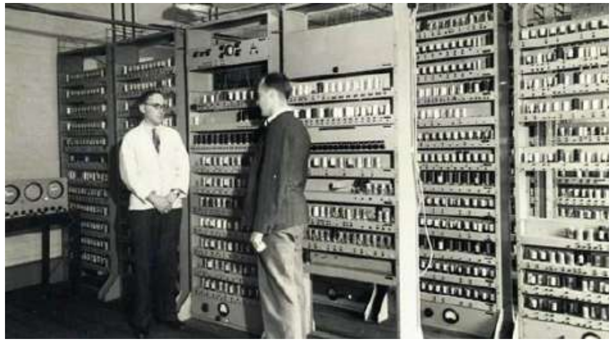
- **电器革命时代：第一台电子数字计算机**

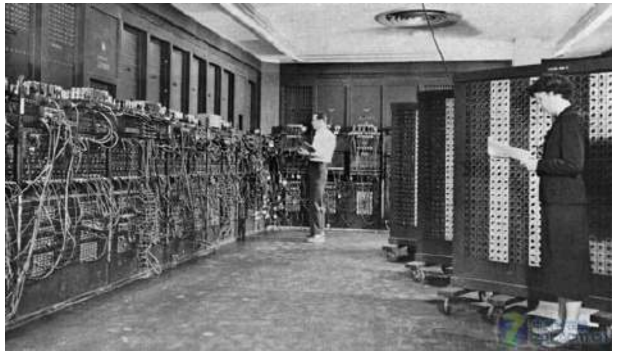
约翰·阿塔纳索夫

- **阿塔纳索夫 美国 保加利亚裔**
  - 1939年制造出了ABC计算机样机
  - 提出三条原则：ABC模型
    - 二进制逻辑进行数字运算
    - 电子技术实现控制、逻辑运算和算术，以保证速度
    - 计算功能和数更新存储功能相分离的结构
  - 1942年正式推出ABC计算机（与学生一起）
    - 采用二进制
    - 电子管作为数据的载体
    - 设计了逻辑电路
    - 磁鼓来存储数据，发明了**可重复**的数据存储方法！

历史上的计算设备

- **电器革命时代：普遍认为的“第一台计算机”**





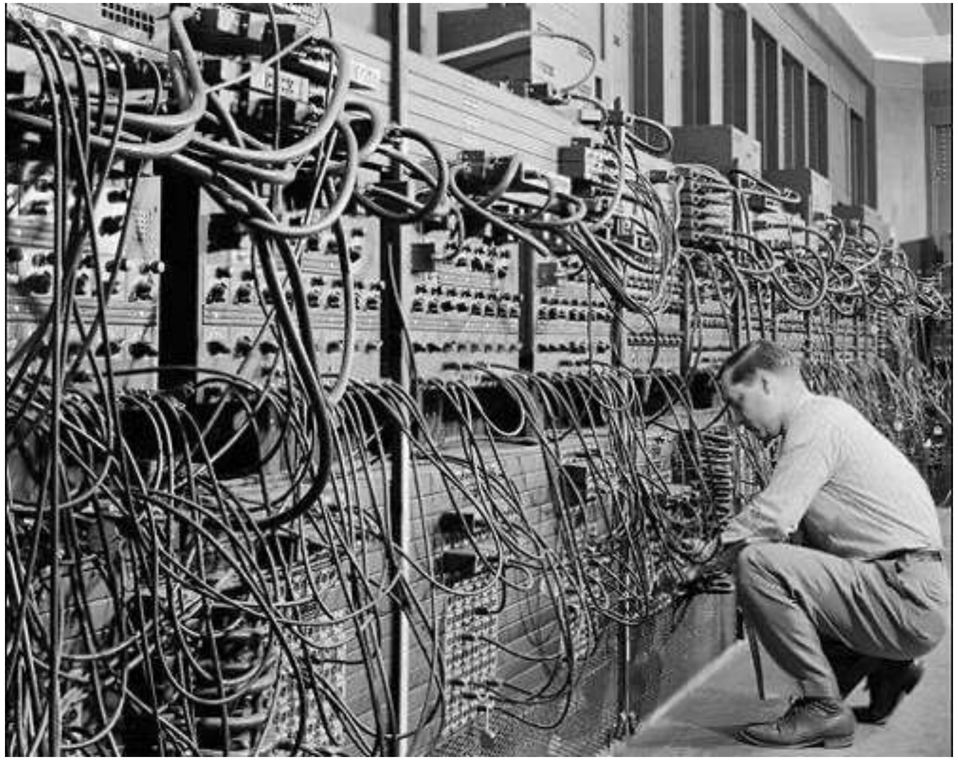
- **团队：**宾夕法尼亚大学莫尔小组工程师：**埃克特**（总工程师，25岁）、**莫克利**、戈尔斯坦、华人科学家朱传榘
- **ENIAC**（Electronic Numeric Integrator and Computer）
  - 电子数字积分计算机
  - 1943年开始，1945年完成（德国投降），1946年正式启动
  - 17,468个电子管、7,200个二极管、70,000+电阻器、10,000+电容、6,000+继电器，电路焊接点50万+，174,000瓦功耗
  - 30米长，3米高，30吨重，占地170平方米
  - 5,000次加法/秒，660次乘法/秒
- **地位**
  - 第一台图灵完全、且可编程的电子计算机，但它不是第一台电子计算机（ABC计算机更早，但不可编程、非图灵完全）
  - 被称为：第一台**通用**计算机

历史上的计算设备

- **到底谁是第一台计算机的制造者？**
    - **1973年 ENIAC vs ABC**
        - 美国法院判决ENIAC的发明，是由ABC的研究推导出来的
        - 原因：莫克利在1941年6月拜访过阿塔纳索夫，参观了ABC
        - 阿塔纳索夫被正式称为“电子计算机之父”
        - 但由于ENIAC规模比ABC大很多，且ENIAC可编程，业界普遍认为ENIAC是第一台计算机
        - 但其实德国Z1计算机也是1941年的。。

历史上的计算设备

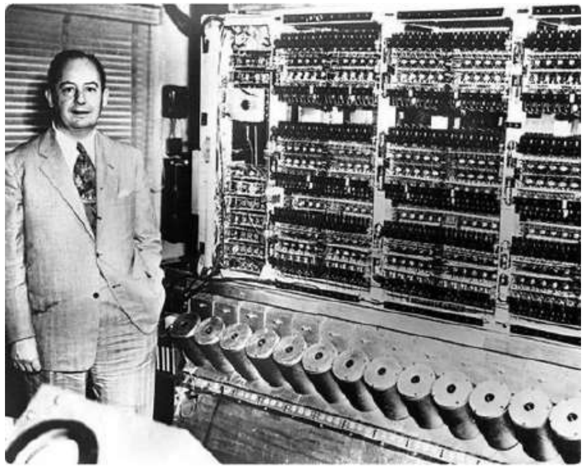
- **电器革命时代：普遍认为的“第一台计算机”**
  - **ENIAC的缺点**
    - 1）还不是存储程序的计算机
    - 2）编程通过手工插接线实现



历史上的计算设备

- **电器革命时代：从ENIAC到EDVAC**

- **冯.洛伊曼 美国 计算技术先驱**
  - 1945年3月，与莫尔学院的埃克特和莫克利讨论了2天
    - 拟定了存储程序式的电子计算机方案
    - 1945年6月发表《存储程序控制原理》



- **EDVAC（Electronic Discrete Variable Automatic Computer）**
  - 离散变量自动电子计算机
  - EDVAC于1952年制造完成
  - EDVAC是**第一台存储程序计算机**
  - EDVAC是**现代计算机的原型和范本**
    （目前的常用计算机仍然使用 冯.洛伊曼 结构）
  - ENIAC一样，EDVAC也是为<u>美国陆军阿伯丁试验场</u>的弹道研究实验室研制

历史上的计算设备

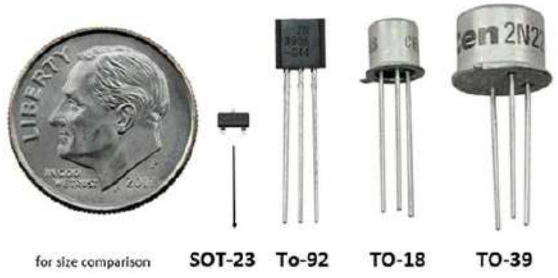
- **第一代计算机：电子管（真空管）**
  - 1940 ~ 1950年代
  - 电子管控制真空中电子流动
  - 设置为0或1两个状态
  - 速度比机械快，但**体积大、耗能高、容易坏**
  - ENIAC有17,468个电子管，第一年换了19,000个..
  - 只能用0/1进行编程

- **早期计算机**
  - 手工计算器，1200 ~ 1600年
  - 机械计算器，1600 ~ 1930年
  - 计算机原型，1937 ~ 1946年
- **现代计算机**
  - 电子管计算机，1946年 ← **我们讲到这里，目前还是用电子管**
  - 晶体管计算机，20世纪50年代后期
  - 集成电路计算机，1965年
  - 超大规模集成电路，20世纪70年代早期
  - ... 个人电脑 ... 云计算 ...

半导体技术的发展史

历史上的计算设备

- **第二代计算机：晶体管**



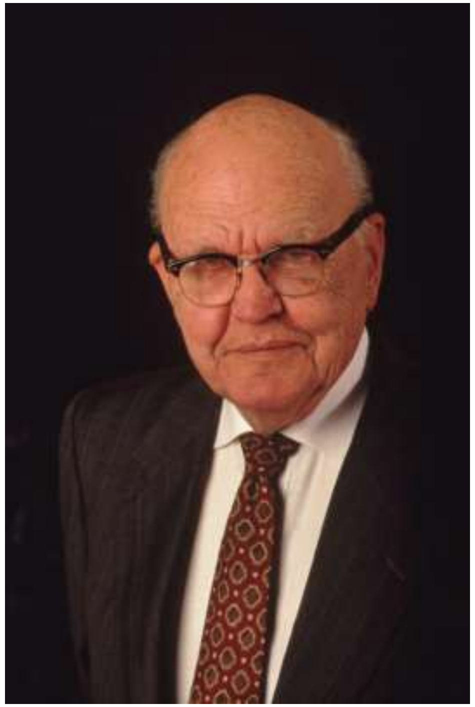
- 美国贝尔实验室于1947年发明了晶体管
  - 功能与电子管类似，但更小、更便宜、更省电、寿命长
- 20世纪50年代后期开始，**第二代计算机**使用**晶体管**存储数据
- 出现**操作系统**
  - 标准化的硬件资源管理
  - 但不可移植
- 出现**高级编程语言**
  - Fortran（现在还在使用）
  - Cobol

```
program main
  implicit none
  write(*,*) "Hello World!"
end
```

John Bardeen, William Shockley and Walter Brattain at Bell Labs, 1948

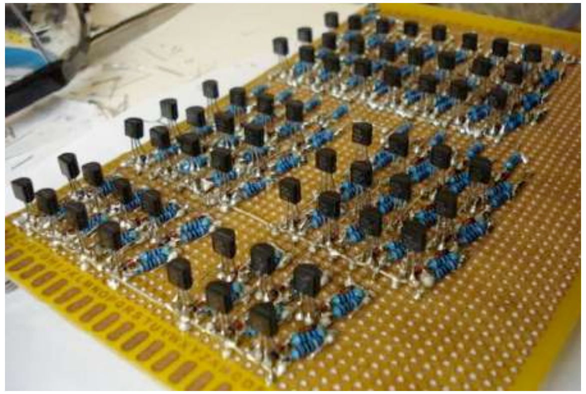
历史上的计算设备

- **第三代计算机：集成电路**

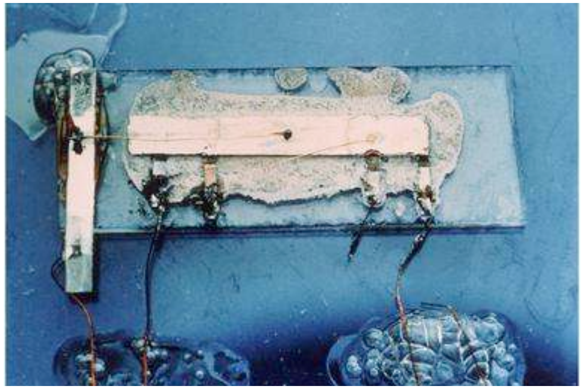


杰克.基尔比
Jack Kilby

- **基尔比 美国德州仪器公司工程师**
  - 1958年发明集成电路，集成电路近一步缩小计算机体积，将大量晶体管压在一个单独的微型芯片上。
  - 2000年获得了诺贝尔物理学奖
- 1965年开始，**第三代计算机**使用**集成电路**
- 那个时代还出现了**可移植的操作系统**
- 那个时代还出现了**C语言**



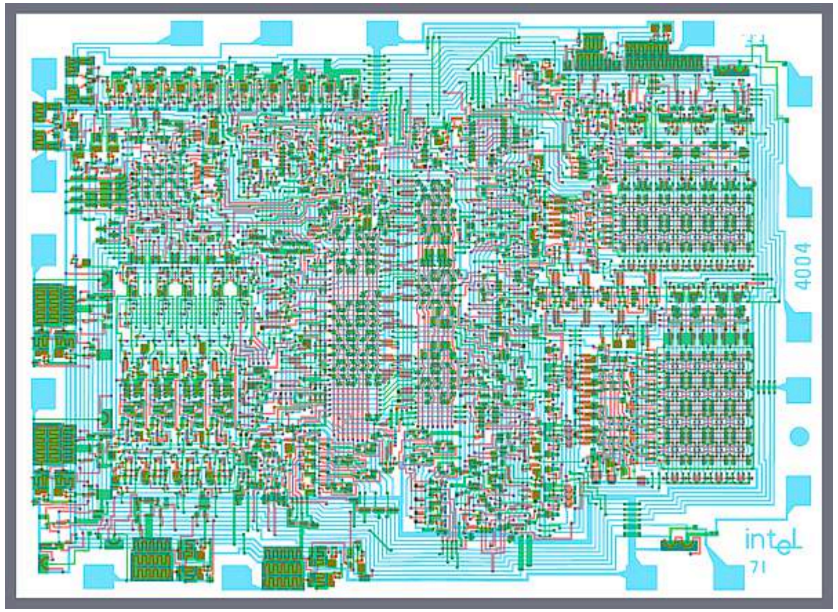
分立元件组成的电路



Jack Kilby发明的世界上第一个集成电路

历史上的计算设备

- **第四代计算机：超大规模集成电路（我们目前处于这个时代）**

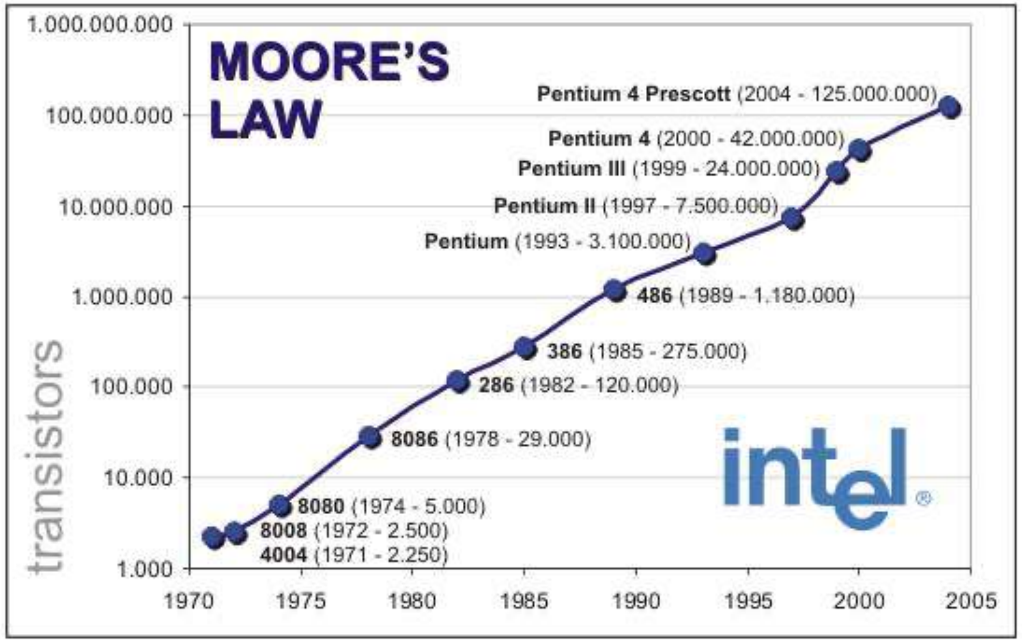


- 始于20世纪70年代
- 集成密度更大：更快、更小、更便宜
- 第一块微处理器：Intel 4004（1971年）
  - 2400个晶体管、计算能力与ENIAC相当
  - 尺寸：3x2mm
- 计算机发展出现瓶颈
  - 硬件、软件全方位出现瓶颈
  - 第五代计算机正在发展中...

Intel 4004

历史上的计算设备

- **第四代计算机：摩尔定律 for 晶体管和计算机**

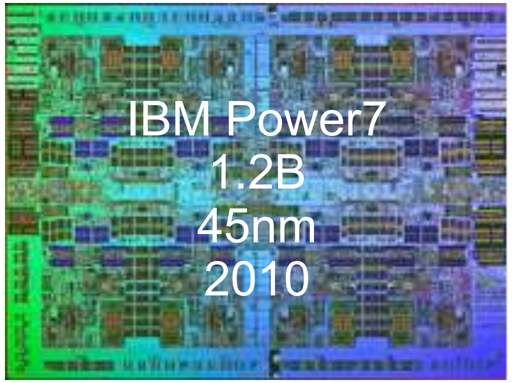


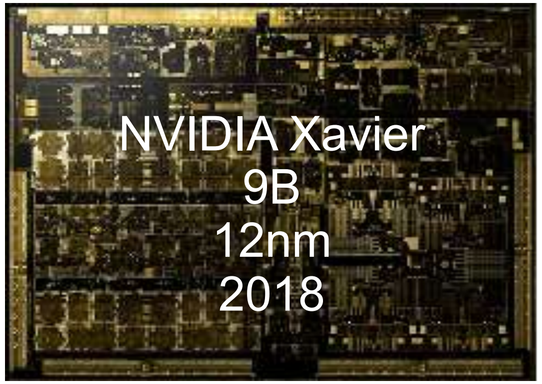
- **戈登摩尔 美国Intel公司创始人之一**
  - 芯片密度每18个月增加1倍
  - 1972年第一代Intel 4004才3000个晶体管
  - CPU性能价格比每18个月翻一翻
  - 速度越来越快，价格越来越便宜

- **硬件发展瓶颈解决**
  - 密度太高，
  - 出现多CPU的计算机

历史上的计算设备

- **第四代计算机：摩尔定律 for 晶体管和计算机**





- **戈登摩尔 美国Intel公司创始人之一**
  - 芯片密度每18个月增加1倍
  - 1972年第一代Intel 4004才3000个晶体管
  - CPU性能价格比每18个月翻一fa
  - 速度越来越快，价格越来越便宜

- **硬件发展瓶颈解决**
  - 密度太高，
  - 出现多CPU的计算机

- 已突破百亿晶体管

- 数学危机到图灵机
- 图灵机原理
- 数的二进制表示
- 二进制的布尔运算

现在
- 历史上的计算设备
- **从电子管到云计算**

- 摩尔定律下的计算危机
- 量子计算

计算机分类

- 微型计算机（microcomputer）
  - 工作站（Workstation）
    - 强大的图形功能
    - 用于计算机辅助设计
  - 个人桌面计算机（PC）
    - 主流Windows/Mac/Linux系统

- 服务器（小型机、中型机）
  - 速度快、存储容量大、稳定、多个处理器，为多个用户服务
  - 大多安装Linux系统
  - 速度较快，并行性好，一般用于专业领域的桌面计算机

计算机分类

- 大型计算机
  - 速度快、体积庞大、价格昂贵，用于一般的大型公司，银行和研究单位，具有很强的管理能力
- 巨型计算机
  - 功能最强，速度最快（几万亿次）
  - N万亿次以上浮点运算/秒
  - 天气预报、地震分析、人工智能、数据可视化、分子模拟
  - 国家科技水平的重要标志

关于超级计算机（巨型机）

- **TOP500列表（摘自Wikipedia）**
  - 针对全球已知最强大的电脑系统做出排名与详细介绍。此项目始于1993年并且一年出版两次最新的超级计算机排名列表。
  - 每年的第一次排名公布总是在六月份的国际超级计算机会议上，而第二次排名公布则是在十一月份的超级计算会议上。此项目主旨在提供一个可靠的基础去追踪与侦测高性能计算的趋势。
  - 由德国曼海姆大学的Hans Meuer、美国田纳西大学诺克斯维尔分校的Jack Dongarra以及美国劳伦斯伯克利国家实验室的Erich Strohmaier与Horst Simon等人共同汇编的
  - www.top500.org
- **2020年6月公布的TOP500列表中的前十名**

排名	名称	国家/地区	场所	安装年份	供应商	处理器核心数	Rmax (Tflops)	Rpeak (Tflops)	功率 (千瓦)
1	富岳	日本	RIKEN	2020	富士通	7,299,072	415,530.0	513,854.7	28,335
2	顶点	美国	橡树岭国家实验室	2018	IBM	2,414,592	148,600.0	200,794.9	10,096
3	Sierra	美国	劳伦斯利佛摩国家实验室	2018	IBM	1,572,480	94,640.0	125,712.0	7,438
4	神威 太湖之光	中国	国家超级计算无锡中心	2016	国家并行计算工程技术研究中心	10,649,600	93,014.6	125,435.9	15,371
5	天河-2A	中国	国家超级计算广州中心	2013	国防科大	4,981,760	61,444.5	100,678.7	18,482
6	HPC5	意大利	埃尼	2020	戴尔	669,760	35,450.0	51,720.8	2,252
7	Selene	美国	英伟达	2020	英伟达	277,760	27,580.0	34,568.6	1,344
8	Frontera	美国	TACC	2019	戴尔	448,448	23,516.4	38,745.9	-
9	Marconi-100	意大利	CINECA	2020	IBM	347,776	21,640.0	29,354.0	1,476
10	代恩特峰	瑞士	瑞士国家超级计算中心	2013	克雷公司	387,872	21,230.0	27,154.3	2,384

关于超级计算机（巨型机）

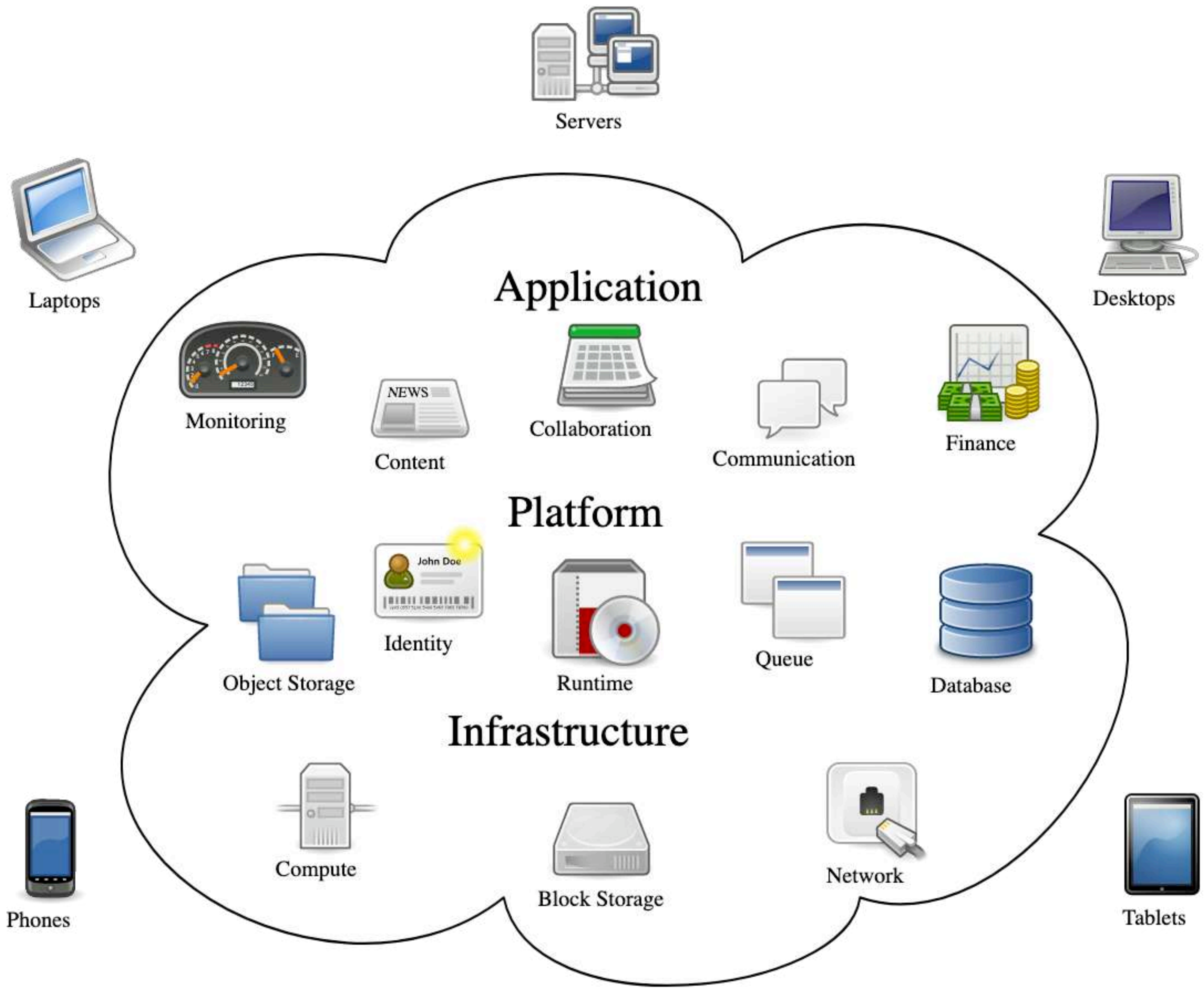
- **Green TOP500列表**
  - 全球最节能的巨型机
  - https://www.top500.org/lists/green500
- **2020年6月公布的Green TOP500列表中的前十名**

Rank	TOP500 Rank	System	Cores	Rmax (TFlop/s)	Power (kW)	Power Efficiency (GFlops/watts)
1	393	**MN-3** - MN-Core Server, Xeon 8260M 24C 2.4GHz, MN-Core, RoCEv2/MN-Core DirectConnect, Preferred Networks Preferred Networks Japan	2,080	1,621.1	77	21.108
2	7	**Selene** - DGX A100 SuperPOD, AMD EPYC 7742 64C 2.25GHz, NVIDIA A100, Mellanox HDR Infiniband, Nvidia NVIDIA Corporation United States	272,800	27,580.0	1,344	20.518
3	468	**NA-1** - ZettaScaler-2.2, Xeon D-1571 16C 1.3GHz, Infiniband EDR, PEZY-SC2 700Mhz, PEZY Computing / Exascaler Inc. PEZY Computing K.K. Japan	1,271,040	1,303.2	80	18.433
4	204	**A64FX prototype** - Fujitsu A64FX, Fujitsu A64FX 48C 2GHz, Tofu interconnect D, Fujitsu Fujitsu Numazu Plant Japan	36,864	1,999.5	118	16.876
5	26	**AiMOS** - IBM Power System AC922, IBM POWER9 20C 3.45GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM Rensselaer Polytechnic Institute Center for Computational Innovations (CCI) United States	130,000	8,339.0	512	16.285
6	6	**HPC5** - PowerEdge C4140, Xeon Gold 6252 24C 2.1GHz, NVIDIA Tesla V100, Mellanox HDR Infiniband, Dell EMC Eni S.p.A. Italy	669,760	35,450.0	2,252	15.740

Why We Need Green Computing?

- **How much energy does a google search consume?**
  - **A single Google query consumes as much energy as an 11-watt light bulb does in one hour**
  - **It consumes as much as preparing a cup of coffee**
  - **“performing two Google searches from a desktop computer can generate about the same amount of carbon dioxide as boiling a kettle” or about 7g of CO_2 per search.**
  - **(Google and you’ll damage the planet, Jan 11, 2009)**

云计算



Cloud computing

- 数学危机到图灵机
- 图灵机原理
- 数的二进制表示
- 二进制的布尔运算

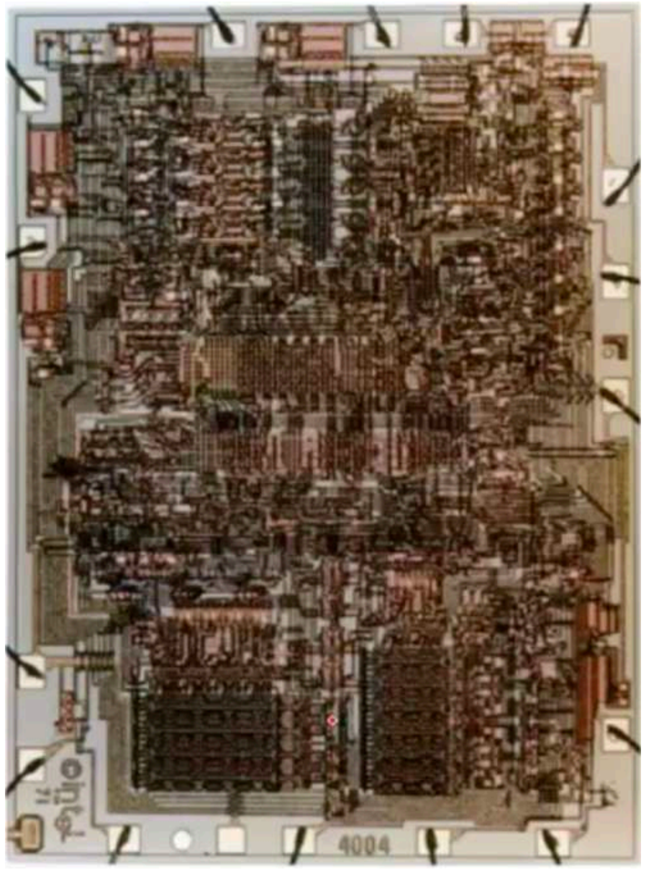
- 历史上的计算设备
- 从电子管到云计算

未来

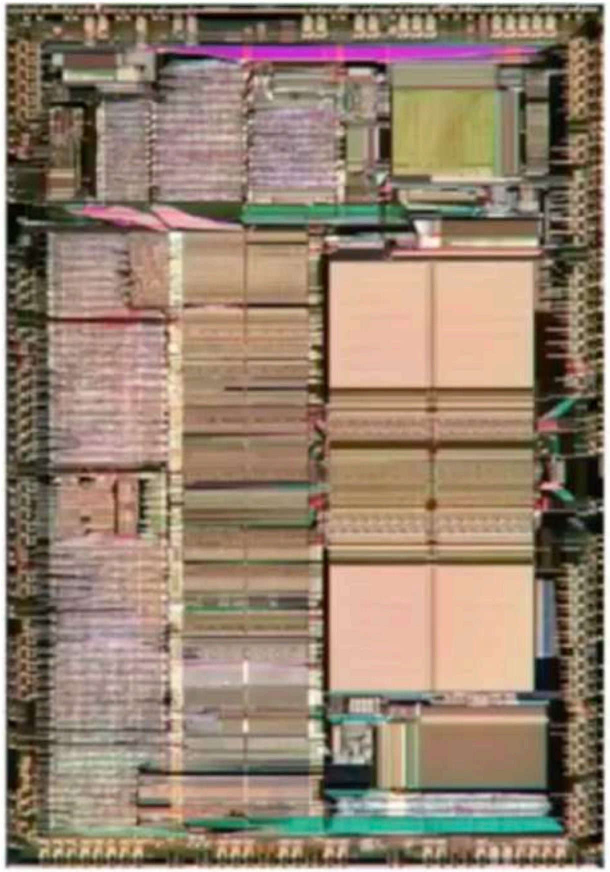
- 摩尔定律下的计算危机
- 量子计算

摩尔定律下的计算危机

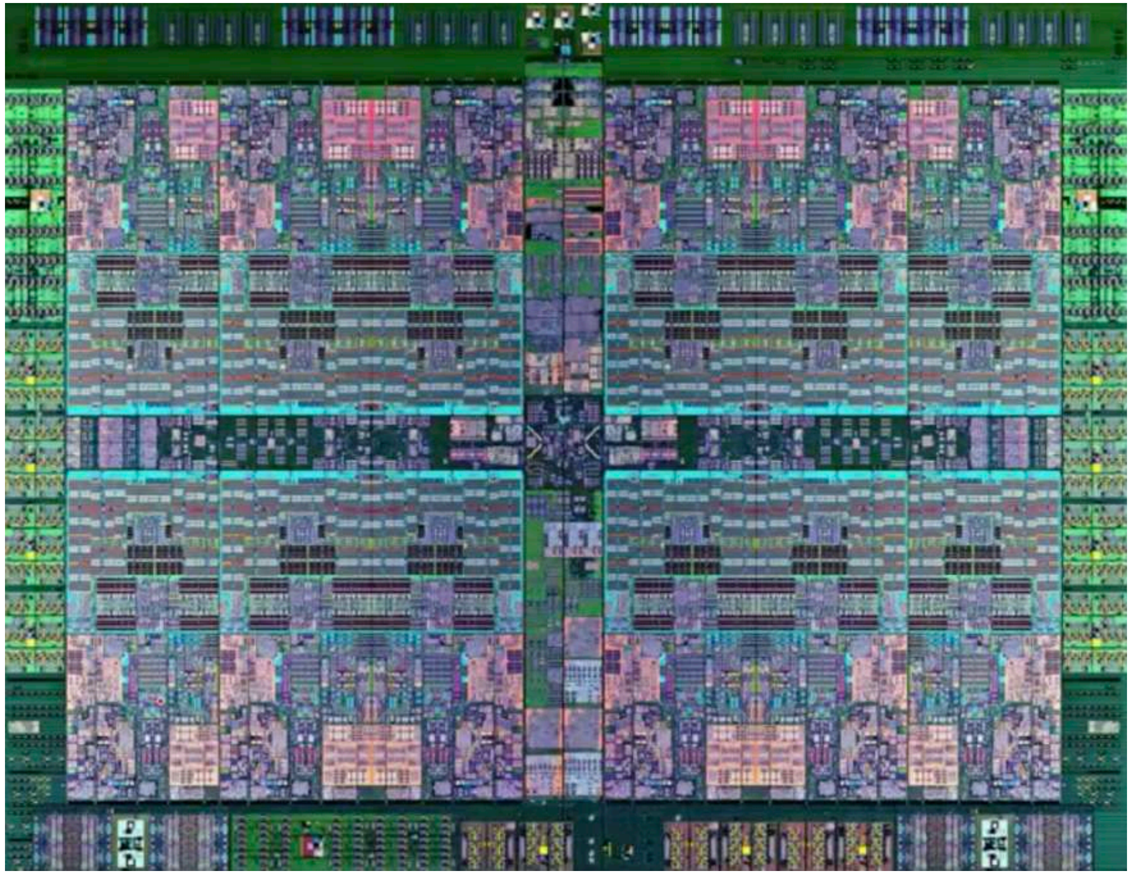
- **摩尔定律基本符合过去晶体管和计算机发展规律，芯片密度每18个月增加1倍**



Intel 4001（1971年）



Intel 80486（1989年）



IBM Power8（2013年）

摩尔定律下的计算危机

- **问：摩尔定律还能坚持多久？**

  - **晶体管大小的限制**
    - 1990年180nm
    - 2011年22nm
    - 2020年5nm
    - 很快就变成原子那么大了：原子直径大约为0.1nm
  - **电泄漏**
    - 晶体管不断缩小，电泄漏情形不断增加，芯片设计难度高
  - **散热**
    - 晶体管密度和速度增加，单位面积产生更多热能
    - 把芯片做的更大，反而用更多电，热量更多

- **答：摩尔定律将会失效！**

摩尔定律下的计算危机

- **急需：新的计算理论和计算模式！**
- **生物计算机**
- **DNA计算机**
- **光子计算机**
- **分子计算机**
- **量子计算机**

- 数学危机到图灵机
- 图灵机原理
- 数的二进制表示
- 二进制的布尔运算

- 历史上的计算设备
- 从电子管到云计算
- 从硬件到软件

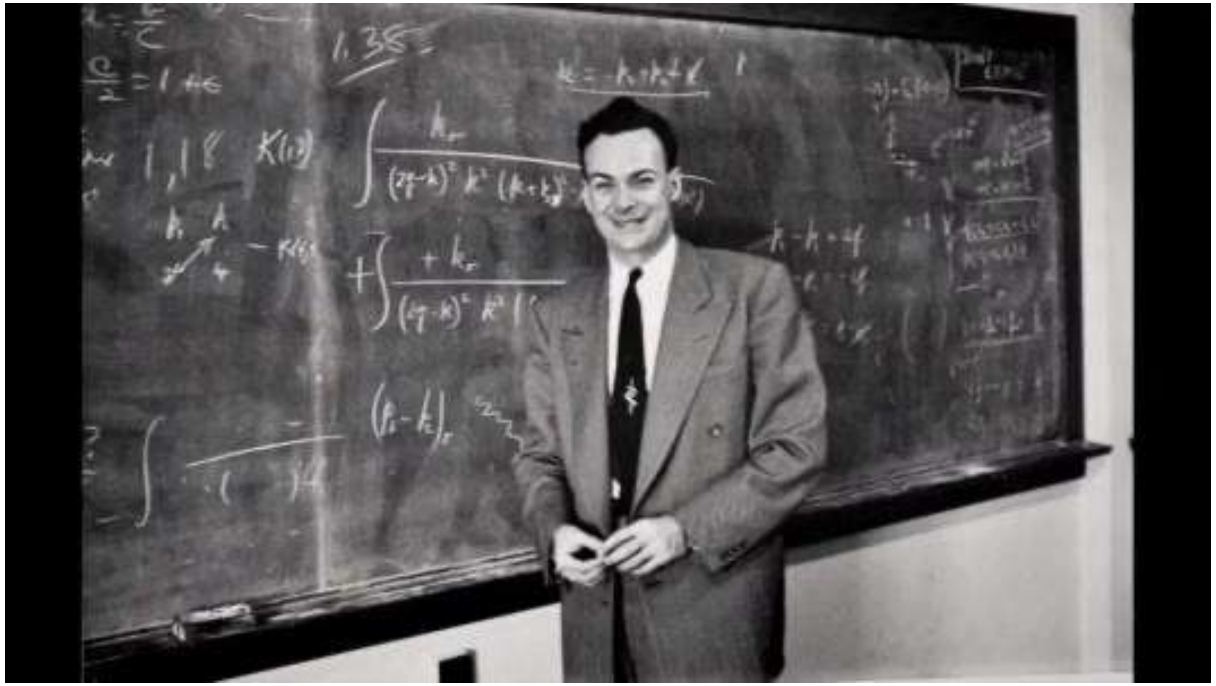
未来

- 摩尔定律下的计算危机
- 量子计算

量子计算

- **量子计算的提出**

- **1982年，理查德.费曼（物理学家）提出：利用量子体系实现通用计算**
  - 他发现传统计算机难以对物理世界的某些现象进行分析，所需要的计算量远远超过了传统计算机所能达到的能力
  - 能不能反过来，用一个可控的量子体系，来进行模拟计算呢？



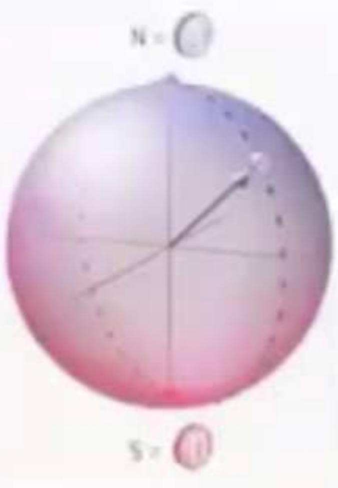
量子计算

- **量子计算的基本原理**

- **传统计算：一个比特某一时刻只能是0或者1**

- **量子计算：一个量子比特可以同时保持多种状态**

  一个量子比特可以同时存储多个数！



量子计算

- 量子计算的基本原理：存储的不同

- 传统计算：两个比特，可以表示00，01，10，11之中的一个组合

- 量子计算：两个比特，可以同时表示00，01，10，11
  传统计算如果要同时表示4个0和1，则需要8个比特

- 传统计算：N个比特，在某一时刻，只能存储2^N个数中的一个

- 量子计算：N个比特，在某一时刻，可以同时存储2^N个二进制数！

量子计算

- **量子计算的基本原理：计算的不同**

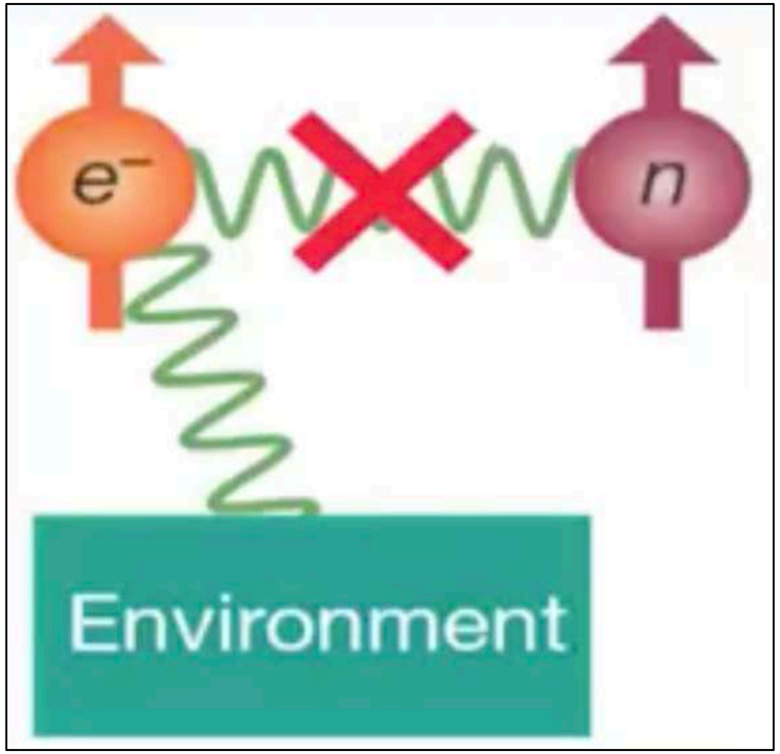
- **传统计算：接受一个输入数据，完成一次运算，输出一个结果**

- **量子计算：同时接受2^N个输入数据，同时完成2^N次运算，同时输出2^N个结果**
  (强大的并行计算能力)

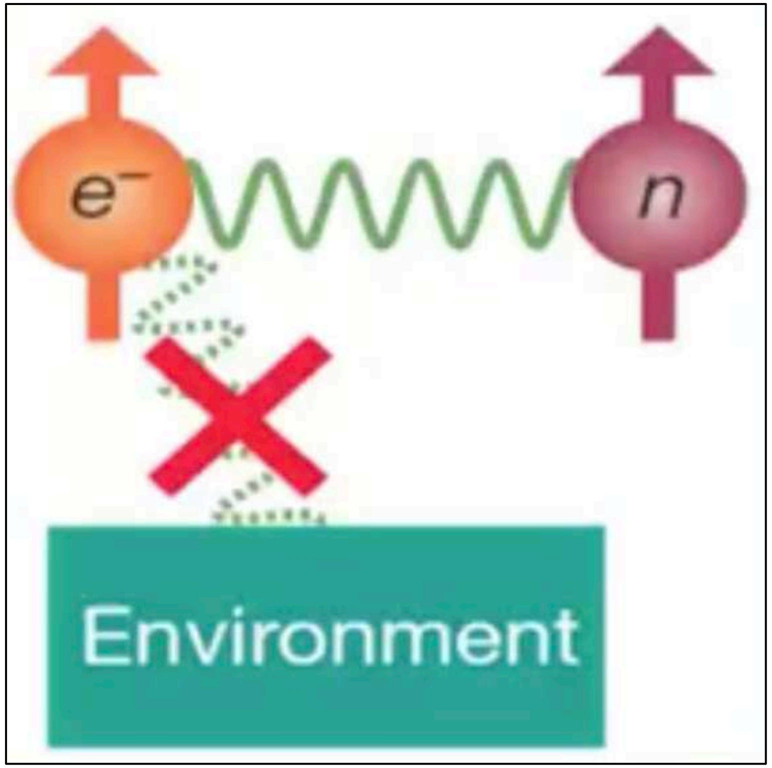
- **传统计算 vs 量子计算**
  - 理论上，300个量子比特能承载的数据是2^{300}，超过宇宙原子数量总和
  - 现实中，需要对计算过程进行纠错，需要多个量子比特才能获得一个可容错的逻辑比特。因此需要1000个物理量子比特才能超越传统计算机的计算能力。

量子计算

- 实现量子计算的**难点**



相矛盾



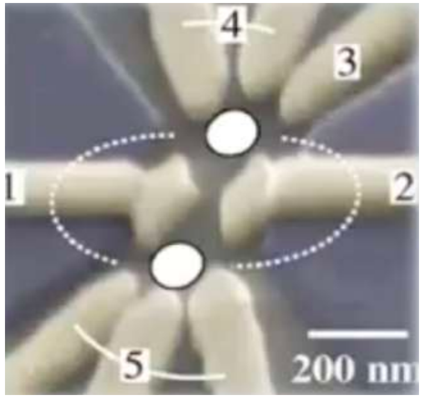
对计算原理而言
需要与外界环境隔离才能实现良好的相干性，进行计算

对实现而言
需要与外界环境良好耦合，才能控制并读出结果

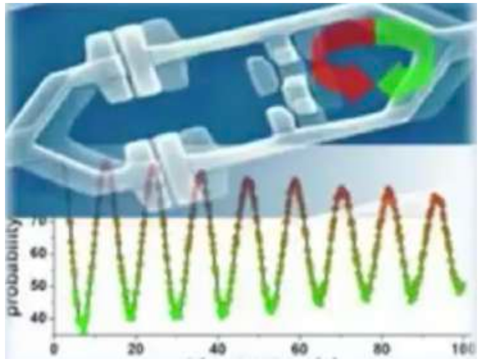
量子计算

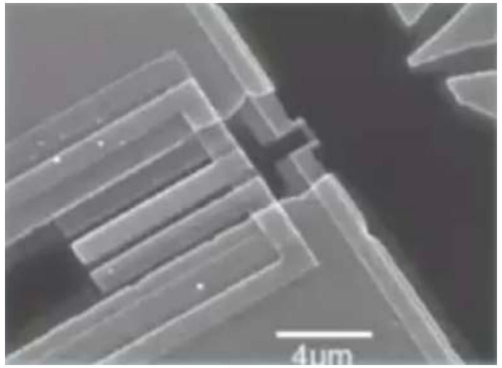
- 实现量子计算的各种努力

  - 基于核磁共振的系统



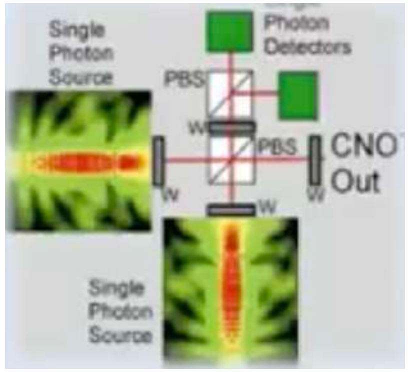
  - 基于超导量子比特的系统

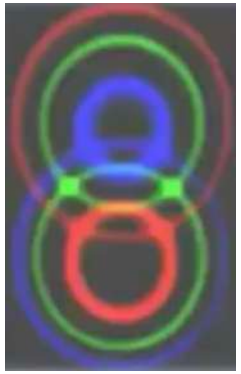




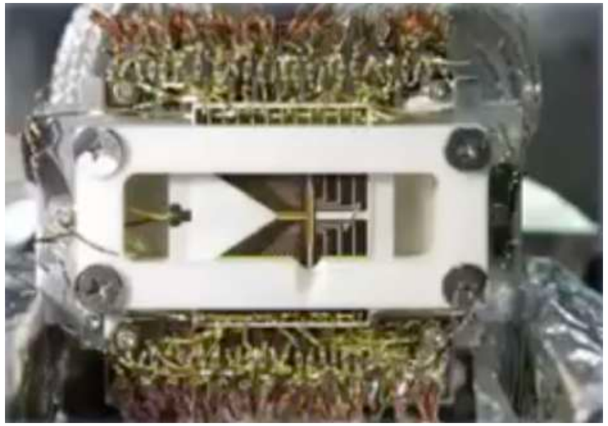
量子计算

- 实现量子计算的各种努力
  - 基于光量子比特的系统





  - 基于离子阱的系统（用以扑获量子）



计算机的发展

- 数学危机到图灵机
- 图灵机原理
- 数的二进制表示
- 二进制的布尔运算

- 历史上的计算设备
- 从电子管到云计算

- 摩尔定律下的计算危机
- 量子计算

谢谢