Xidian University

计算机导论与程序设计

1 计算机体系结构及其编码方式

讲授内容

- 图灵机原理-什么是计算？
- 二进制的产生和运算
- 数的表示：数据的编码的基本原理

计算学科的定义

- 报告《计算作为一门学科》对计算学科作了如下定义：
  - 计算学科是对描述和变换信息的算法过程进行的系统的研究，包括理论、分析、设计、效率、实现和应用等。

计算学科的研究包括从算法与可计算性的研究到硬件和软件的实际实现问题的研究。

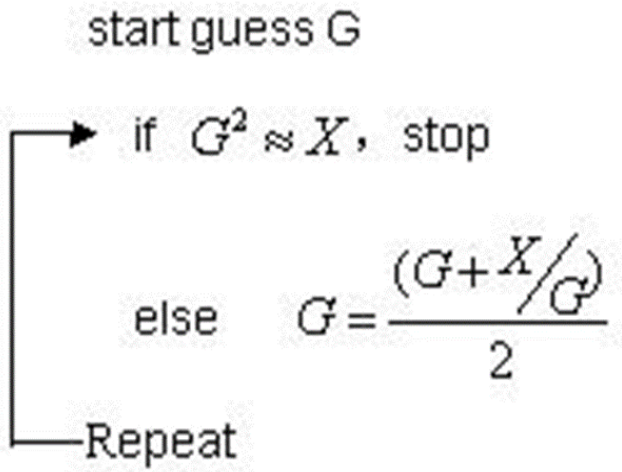
其不但包括从总体上对算法和信息处理过程进行研究，也包括对满足给定规格要求的有效并且可靠的软硬件的设计和实现。

计算机科学与技术的研究范畴

- 计算机理论：离散数学、算法分析理论、形式语言与自动机理论、形式语义学等。
- 计算机硬件：元器件与存储介质、微电子技术、计算机组成原理、微机原理与接口技术、计算机体系结构等。
- 计算机软件：程序设计语言、数据结构、程序设计、编译原理、操作系统、数据库原理、软件工程、程序设计方法学等。
- 计算机网络：网络结构、数据通信与网络协议、网络服务、网络安全等。
- 计算机应用：数值计算（科学计算）、图形学与图像处理、多媒体、计算可视化与虚拟现实、信息系统、办公自动化、计算机辅助设计（测试、制造、教学等）、计算机仿真、人工智能等。

计算学科的根本问题

- 计算学科的根本问题：什么能被（有效地）自动进行。“自动进行”是被构造出来的，构造性是计算学科的基本特征，递归和迭代是最基本的两种构造性方法。
- 计算学科与数学的不同： 数学家强调“是什么(What is it)”的问题，重点放在数学本身的性质上，数学的灵魂是定义，定理和证明是其精髓。计算机科学家则不同，他们不仅要知道“是什么”的问题，更要解决“怎么做（How to do it）”的问题，也即“能行性”问题。比如，平方根的求解算法。



求平方根的亚历山大时代的海伦算法，这是计算机科学；数学则强调证明，比如证明$\sqrt{2}$为无理数

计算学科的认知模型——计算学科二维定义矩阵

3 个过程 / 学科知识领域	抽　象	理　论	设　计
1．算法和复杂性(Algorithms and Complexity, AL)			
2．体系结构和组织(Architecture and Organization, AR)			
3．计算科学(Computational Science, CN)			
4．离散结构(Discrete Structures, DS）			
5．图形学与可视化(Graphics and Visual Computing, GV）			
6．人机交互(Human-Computer Interaction, HC)			
7．信息保障与安全(Information Assurance and Security, IAS）			
8．信息管理(Information Management, IM)			
9．智能系统(Intelligent Systems, IS）			
10．网络与通信(Networking and Communication, NC）			
11．操作系统(Operating Systems, OS)			
12．基于平台的开发(Platform-Based Development, PBD)			
13．并行与分布式计算(Parallel and Distributed Computing, PD)			
14．程序设计语言(Programming Languages, PL)			
15．软件开发基础(Software Development Fundamentals ,SDF）			
16．软件工程(Software Engineering, SE)			
17．计算机系统基础(System Fundamentals, SF)			
18．社会问题与专业实践(Social Issues and Professional Practice, SP)			

计算学科二维定义矩阵（1）

- 定义矩阵的纵向维是“学科分支领域”。
  - 将整个学科划分为若干分支领域有助于我们对学科的理解。
  - 这些学科分支领域将随着计算技术的发展而变化。

3 个过程 / 学科知识领域	抽 象	理 论	设 计
1．算法和复杂性(Algorithms and Complexity, AL)			
2．体系结构和组织(Architecture and Organization, AR)			
3．计算科学(Computational Science, CN)			
4．离散结构(Discrete Structures, DS)			
5．图形学与可视化(Graphics and Visual Computing, GV)			
6．人机交互(Human-Computer Interaction, HC)			
7．信息保障与安全(Information Assurance and Security, IAS)			
8．信息管理(Information Management, IM)			
9．智能系统(Intelligent Systems, IS)			
10．网络与通信(Networking and Communication, NC)			
11．操作系统(Operating Systems, OS)			
12．基于平台的开发(Platform-Based Development, PBD)			
13．并行与分布式计算(Parallel and Distributed Computing, PD)			
14．程序设计语言(Programming Languages, PL)			
15．软件开发基础(Software Development Fundamentals ,SDF)			
16．软件工程(Software Engineering, SE)			
17．计算机系统基础(System Fundamentals, SF)			
18．社会问题与专业实践(Social Issues and Professional Practice, SP)			

计算学科二维定义矩阵（2）

- **定义矩阵的横向维是“3个过程”。**
  - 也称为3个学科形态，即：抽象、理论和设计。
  - 这3个过程反映了计算领域中人们的认识是从感性认识（抽象）到理性认识（理论），再由理性认识（理论）回到实践（设计）的科学思维方法。
  - 抽象、理论和设计3个过程之间相互作用，推动了计算学科及其分支领域的发展。

3个过程 / 学科知识领域	抽 象	理 论	设 计
1．算法和复杂性(Algorithms and Complexity, AL)			
2．体系结构和组织(Architecture and Organization, AR)			
3．计算科学(Computational Science, CN)			
4．离散结构(Discrete Structures, DS)			
5．图形学与可视化(Graphics and Visual Computing, GV)			
6．人机交互(Human-Computer Interaction, HC)			
7．信息保障与安全(Information Assurance and Security, IAS)			
8．信息管理(Information Management, IM)			
9．智能系统(Intelligent Systems, IS)			
10．网络与通信(Networking and Communication, NC)			
11．操作系统(Operating Systems, OS)			
12．基于平台的开发(Platform-Based Development, PBD)			
13．并行与分布式计算(Parallel and Distributed Computing, PD)			
14．程序设计语言(Programming Languages, PL)			
15．软件开发基础(Software Development Fundamentals ,SDF)			
16．软件工程(Software Engineering, SE)			
17．计算机系统基础(System Fundamentals, SF)			
18．社会问题与专业实践(Social Issues and Professional Practice, SP)			

计算学科二维定义矩阵（2）

- 该二维定义矩阵是对计算学科的一个高度概括。
- 因此，我们将学科的认知问题具体为学科二维定义矩阵的认知问题，从而使学科的认知具体化。

3 个过程 / 学科知识领域	抽 象	理 论	设 计
1．算法和复杂性(Algorithms and Complexity, AL)			
2．体系结构和组织(Architecture and Organization, AR)			
3．计算科学(Computational Science, CN)			
4．离散结构(Discrete Structures, DS)			
5．图形学与可视化(Graphics and Visual Computing, GV)			
6．人机交互(Human-Computer Interaction, HC)			
7．信息保障与安全(Information Assurance and Security, IAS)			
8．信息管理(Information Management, IM)			
9．智能系统(Intelligent Systems, IS)			
10．网络与通信(Networking and Communication, NC)			
11．操作系统(Operating Systems, OS)			
12．基于平台的开发(Platform-Based Development, PBD)			
13．并行与分布式计算(Parallel and Distributed Computing, PD)			
14．程序设计语言(Programming Languages, PL)			
15．软件开发基础(Software Development Fundamentals ,SDF)			
16．软件工程(Software Engineering, SE)			
17．计算机系统基础(System Fundamentals, SF)			
18．社会问题与专业实践(Social Issues and Professional Practice, SP)			

从计算机科学到计算思维

科学思维主要分三大类：

1. 理论思维：以数学为基础：理论源于数学，理论思维支撑着所有的学科领域。正如数学一样，定义是理论思维的灵魂，定理和证明则是它的精髓。公理化方法是最重要的理论思维方法。

2. 实验思维：以物理等学科为基础：实验思维的先驱应当首推意大利著名的物理学家、天文学家和数学家伽利略，他开创了以实验为基础具有严密逻辑理论体系的近代科学，被人们誉为“近代科学之父”。

3. 计算思维：以计算机科学为基础：计算思维又叫构造性思维，以设计与构造为特征。具体而言，可以说，计算思维是通过约简、嵌入、转化和仿真等方法，把一个看来困难的问题重新阐释成一个我们知道问题怎样解决的思维方法；计算思维又可以认为是一种递归思维，是一种并行思维，是一种……

计算思维

计算思维的定义：计算思维是运用计算机科学的基础概念进行问题求解，系统设计，以及人类行为理解的涵盖了计算机科学之广度的一系列思维活动。

摘自：Jeannette M. Wing. Computational Thinking. Communications of the ACM, 20

计算思维的本质（两个“A”）

- 抽象（Abstraction）
- 自动化（Automation）

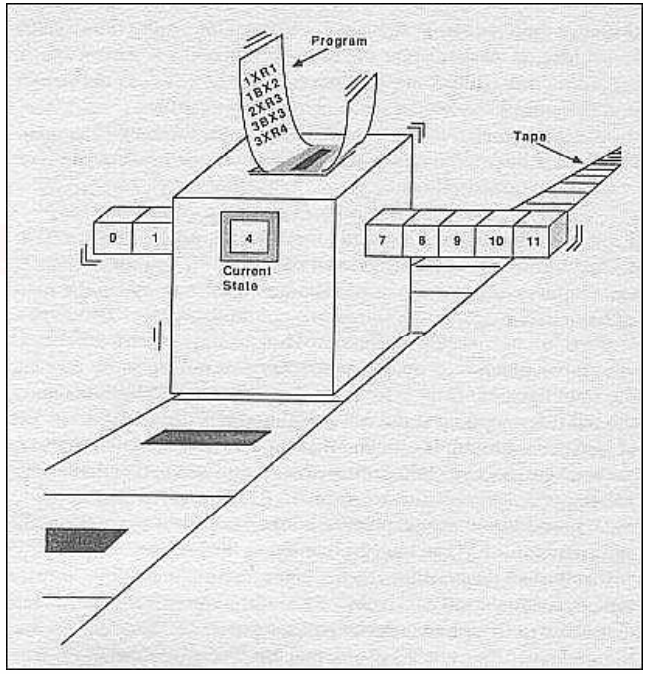
Note：NBA的统计分析系统

伟大的图灵机

- 抽象计算模型(思想模型)
  - 一切具有计算功能的机器的基础
  - 计算机和人工智能的发展一直没有彻底脱离图灵机模型
- 图灵和超越图灵?
  - 网络，云计算，量子计算?

图灵机组成

- **一条无限长的带：**带上划上格子，每个格子中可以写一个符号；所有允许出现的符号属于一个预先规定好的字母表。
- **一个读写头：**每次可以从带上读出一个符号，也可以擦去或改写这个符号；读写头可以左移一格、右移一格或者保持不动。
- **一个控制器：**控制器里存有一个程序（Program）（程序就是指令（Instructions）的序列）；控制器在每个时刻处于一定的状态，叫做机器状态；
  - 当读写头从带上读出一个符号后，控制器就根据这个符号和当时的机器状态，参照程序作出反应，即指挥读写头进行书写或者移动，并决定是否改变机器状态。



图灵机原理–指令序列

当前状态	输入	输出	动作	下一时刻状态
B	1	1	前移	C
A	0	1	不动	B
C	0	0	后移	A
…	…	…	…	…

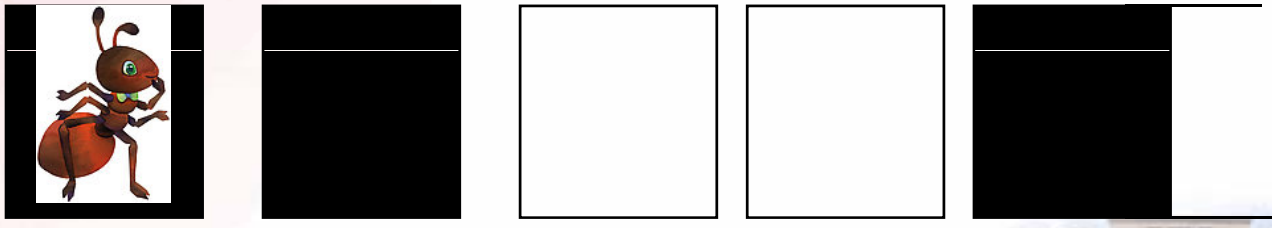
如何理解图灵机模型

- **图灵机原理**：根据读取信息和内部状态进行查表就可以确定下一时刻的内部状态和输出动作
  - 只要变化程序（规则表），它就可以完成任何计算工作
  - 这么简单的图灵机模型能否完成复杂任务?

小虫模型

程序1

输入	输出
黑色	前移
白色	后移



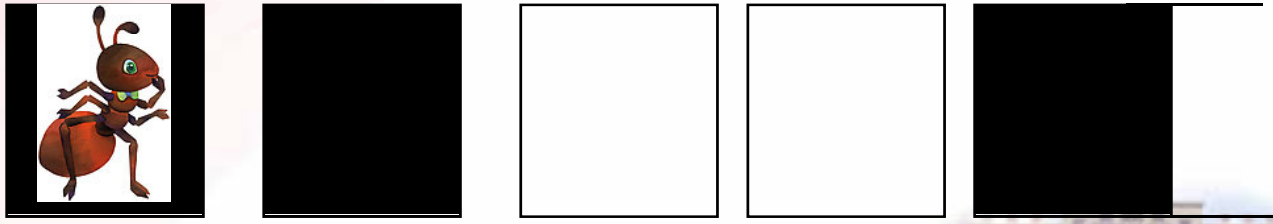
小虫模型

程序1

输入　　输出

黑色　　前移

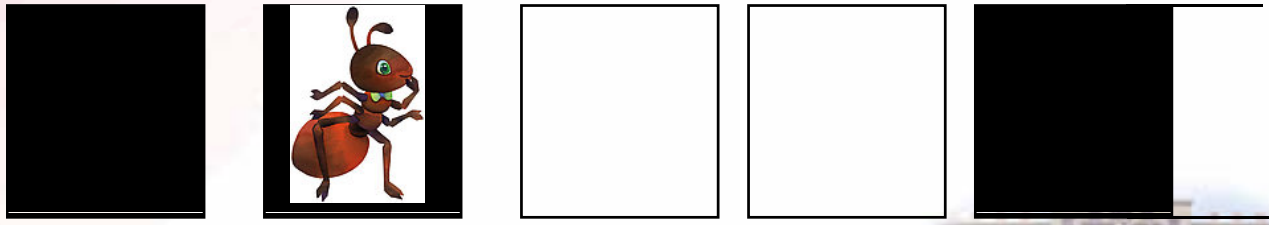
白色　　后移



小虫模型

程序1

输入	输出
黑色	前移
白色	后移



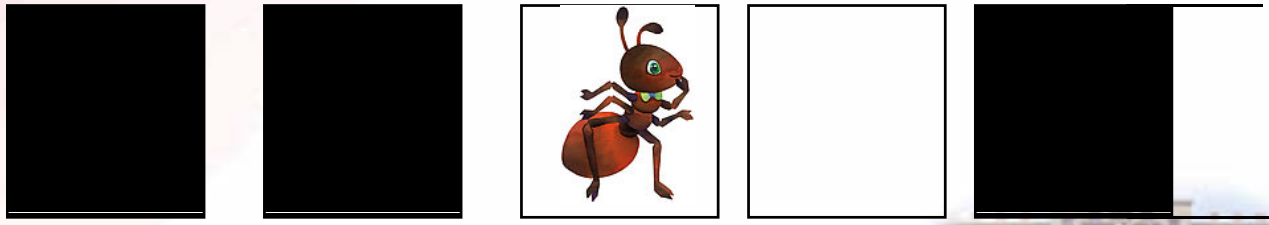
小虫模型

程序1

输入　　输出

黑色　　前移

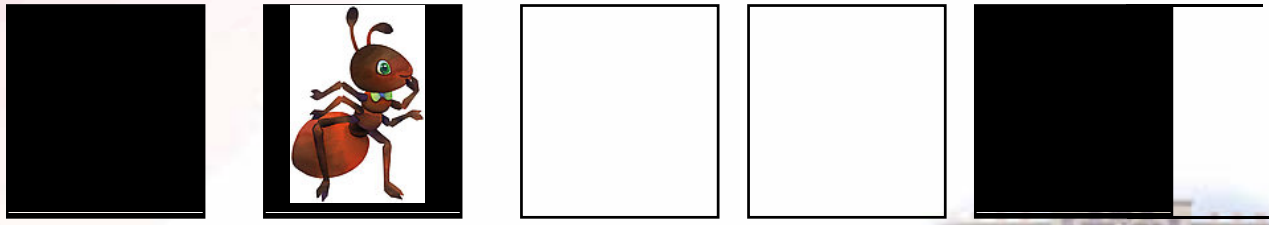
白色　　后移



小虫模型

程序1

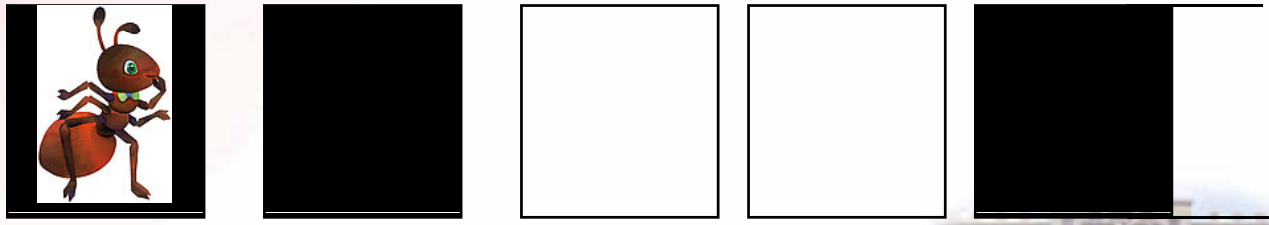
输入	输出
黑色	前移
白色	后移



小虫模型

程序2

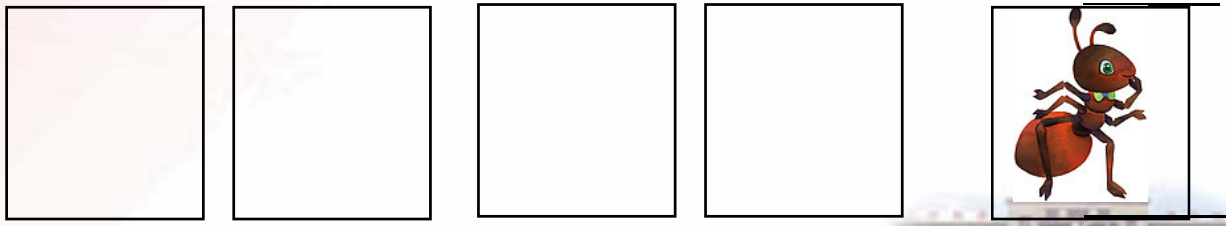
输入	输出
黑色	涂白
白色	前移



小虫模型

程序2

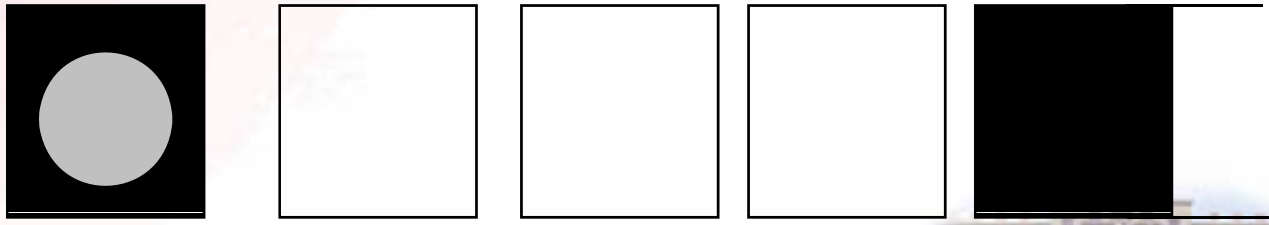
输入	输出
黑色	涂白
白色	前移



小虫模型

程序3

输入	内部状态	输出	下一时刻状态
黑色	饥饿	涂白	吃饱
黑色	吃饱	后移	饥饿
白色	饥饿	涂黑	饥饿
白色	吃饱	前移	吃饱



用图灵机进行计算

字 母 表：{ 1, b }

机器状态：{ q1, q2, q3 }

读写头

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

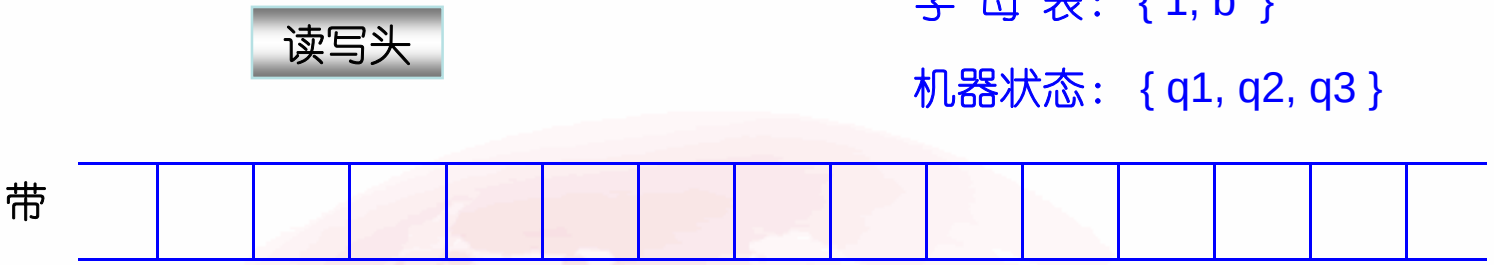
用图灵机进行计算


读写头
字 母 表：{ 1, b }
机器状态：{ q1, q2, q3 }
带
当前机器状态
控制器
程序
要输出的符号
当前读入的符号
读写头动作
下一个机器状态
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

用图灵机进行计算

字 母 表：{ 1, b }

机器状态：{ q1, q2, q3 }



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

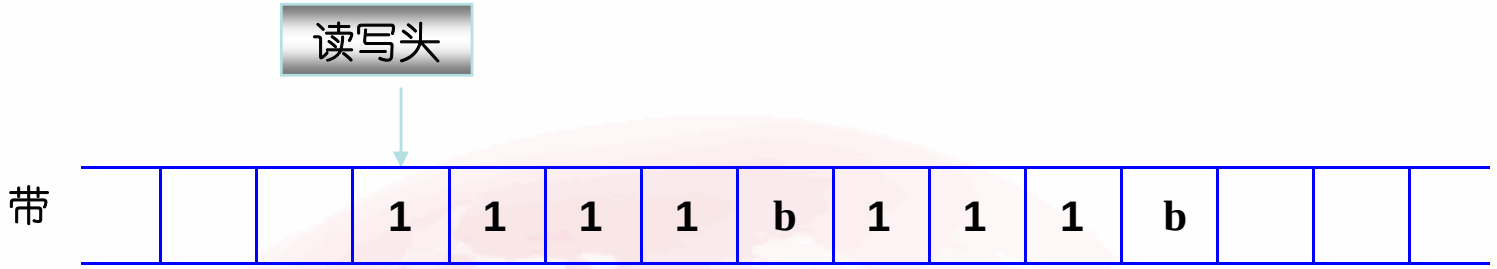
2）判断读入的符号

3）写一个符号

4）控制读写头动作

5）设置下一机器状态

用图灵机进行计算



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

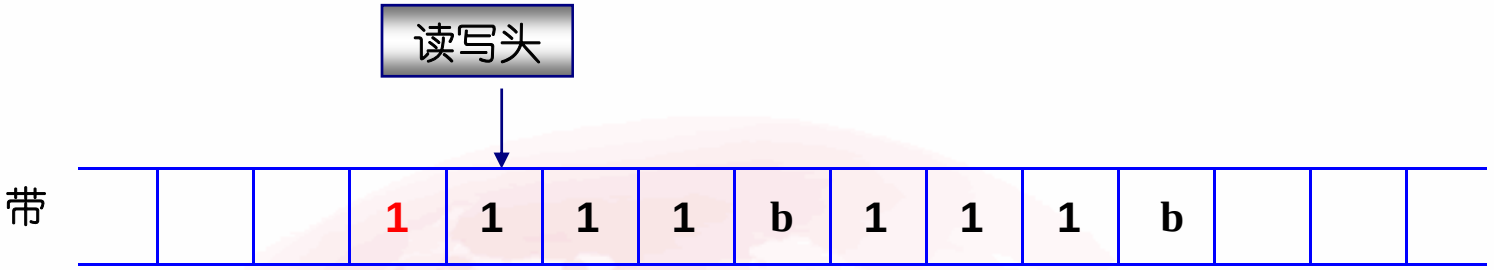
2）判断读入的符号

3）写一个符号

4）控制读写头动作

5）设置下一机器状态

用图灵机进行计算



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

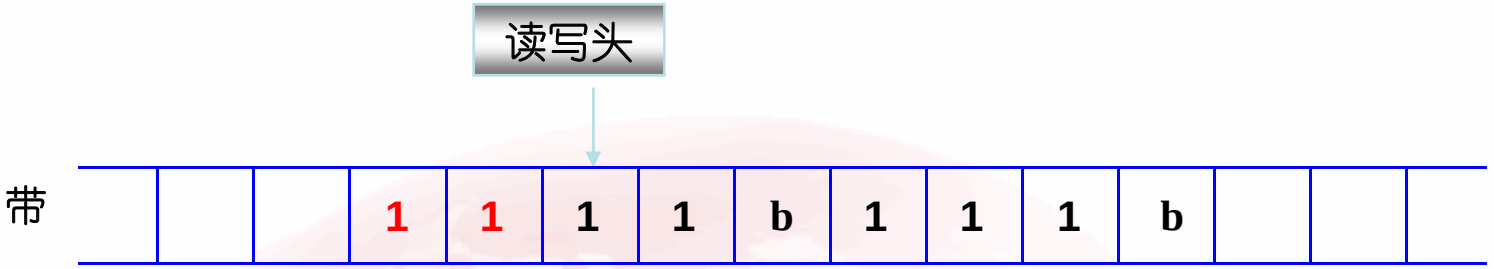
2）判断读入的符号

3）写一个符号

4）控制读写头动作

5）设置下一机器状态

用图灵机进行计算



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

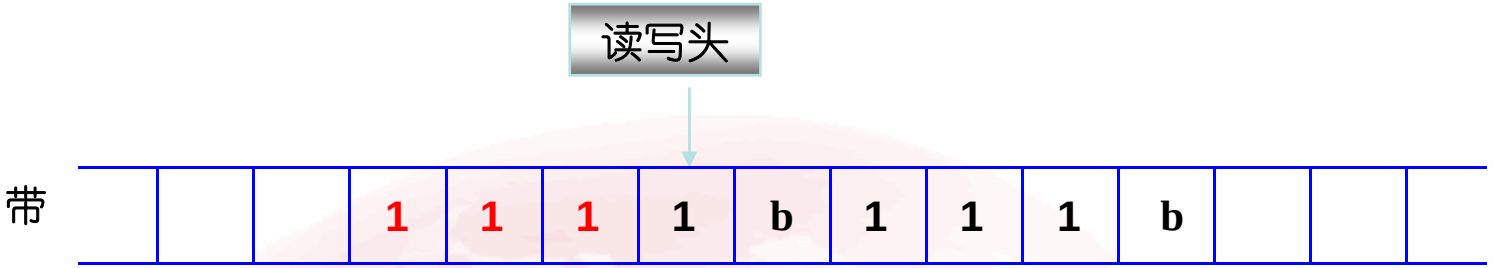
2）判断读入的符号

3）写一个符号

4）控制读写头动作

5）设置下一机器状态

用图灵机进行计算



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

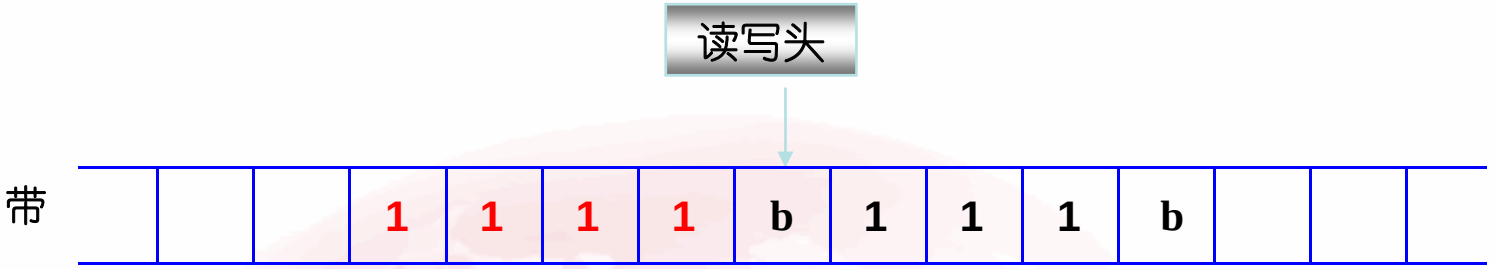
2）判断读入的符号

3）写一个符号

4）控制读写头动作

5）设置下一机器状态

用图灵机进行计算



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

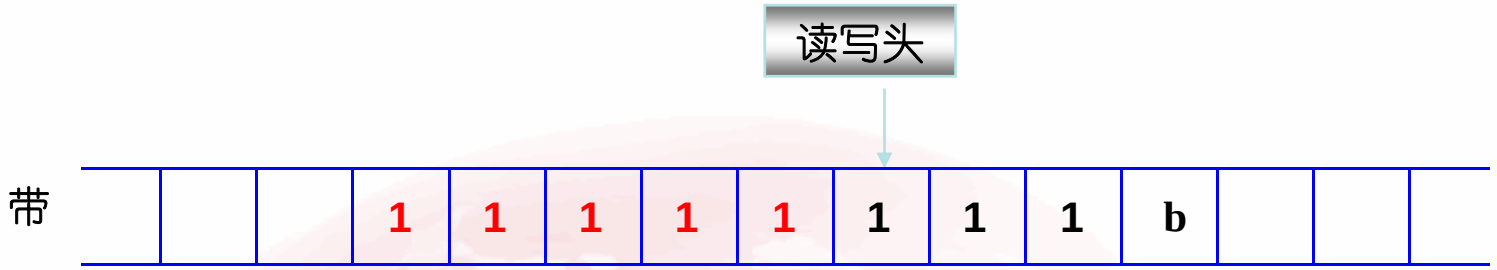
2）判断读入的符号

3）写一个符号

4）控制读写头动作

5）设置下一机器状态

用图灵机进行计算



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

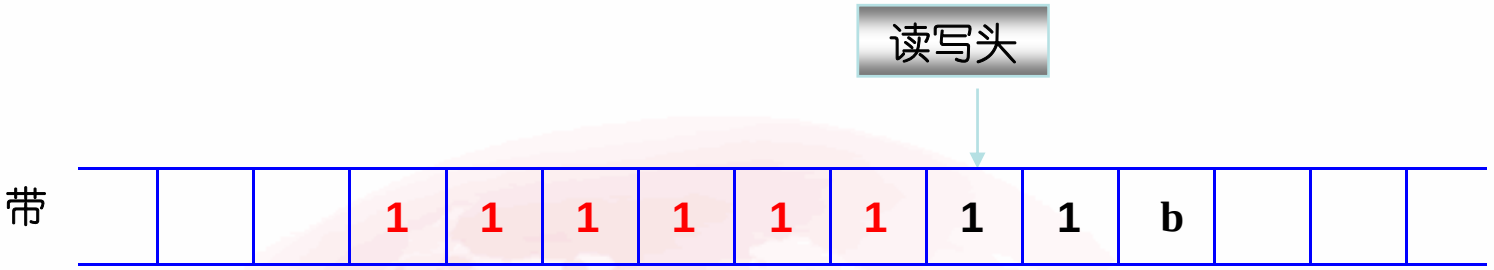
2）判断读入的符号

3）写一个符号

4）控制读写头动作

5）设置下一机器状态

用图灵机进行计算



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

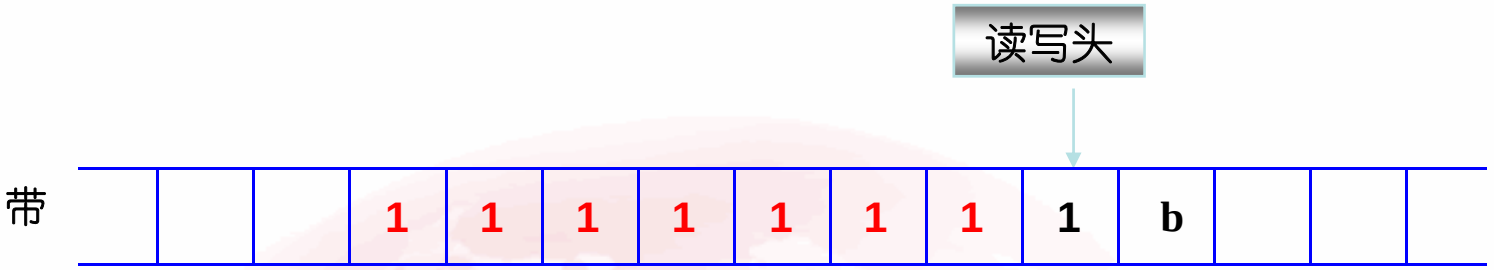
2）判断读入的符号

3）写一个符号

4）控制读写头动作

5）设置下一机器状态

用图灵机进行计算



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

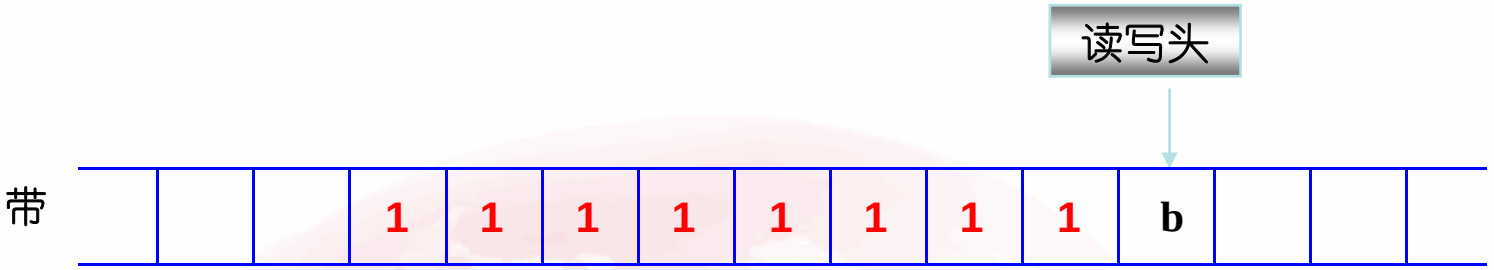
2）判断读入的符号

3）写一个符号

4）控制读写头动作

5）设置下一机器状态

用图灵机进行计算



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

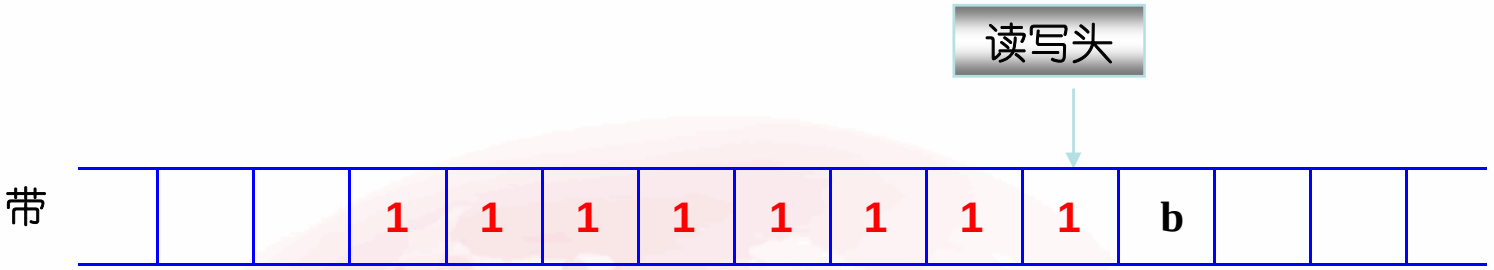
2）判断读入的符号

3）写一个符号

4）控制读写头动作

5）设置下一机器状态

用图灵机进行计算



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

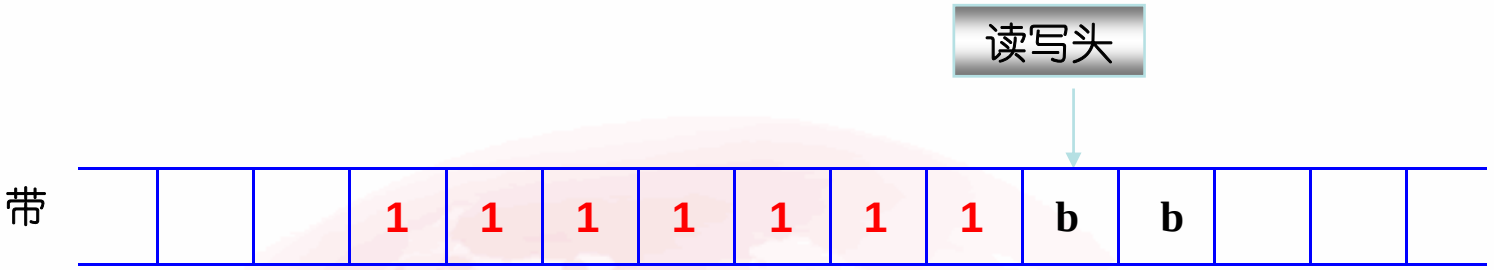
2）判断读入的符号

3）写一个符号

4）控制读写头动作

5）设置下一机器状态

用图灵机进行计算



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

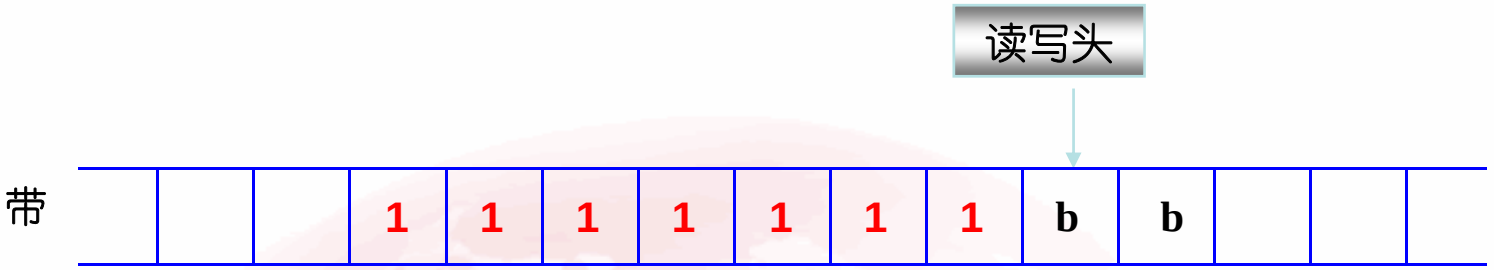
2）判断读入的符号

3）写一个符号

4）控制读写头动作

5）设置下一机器状态

用图灵机进行计算



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

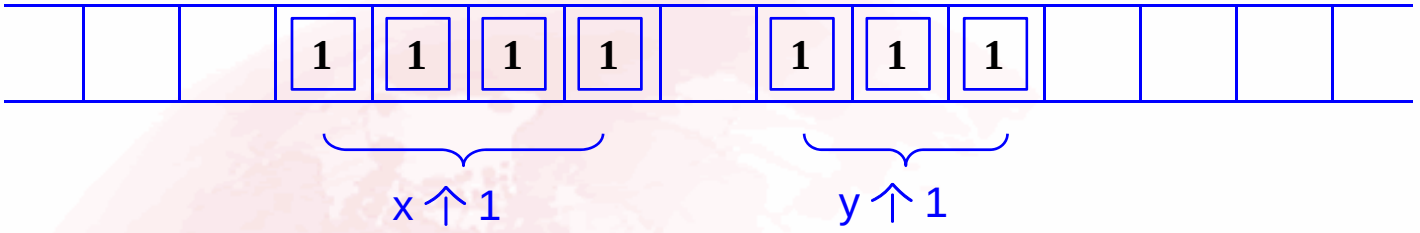
4）控制读写头动作

5）设置下一机器状态

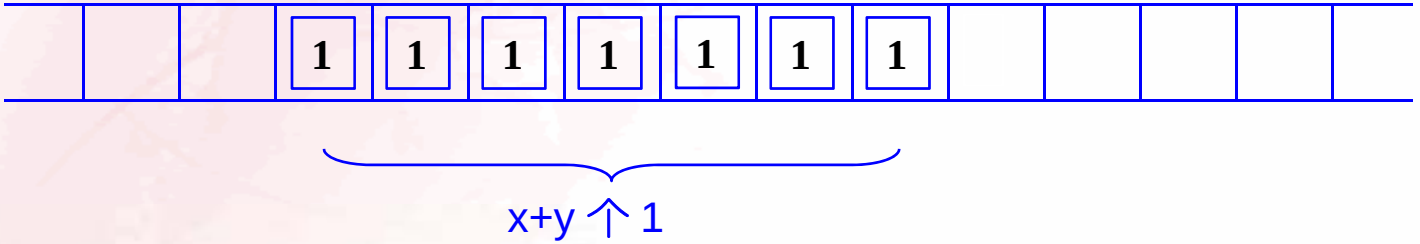
用图灵机进行计算

- 这个例子中，是在用图灵机进行什么计算？

开始时：



结束时：



- 这是在进行任意两个大于 0 的整数的相加。

从图灵机我们看到了什么？

- 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。因此，有条件用真正的机器来实现图灵机。
- 依据程序，可以对符合字母表要求的任意符号序列进行计算。因此，同一个图灵机可以进行规则相同、对象不同的计算，具有数学概念上的函数 f（x）的计算能力。
- 如果开始的状态（读写头的位置、机器状态）不同，那么计算的涵义与计算的结果就可能不同。在按照每条指令进行计算时，都要参照当前的机器状态，计算后也可能改变当前的机器状态。

  口状态（States）是计算机科学中非常重要的一个概念。

我们还看到了什么？

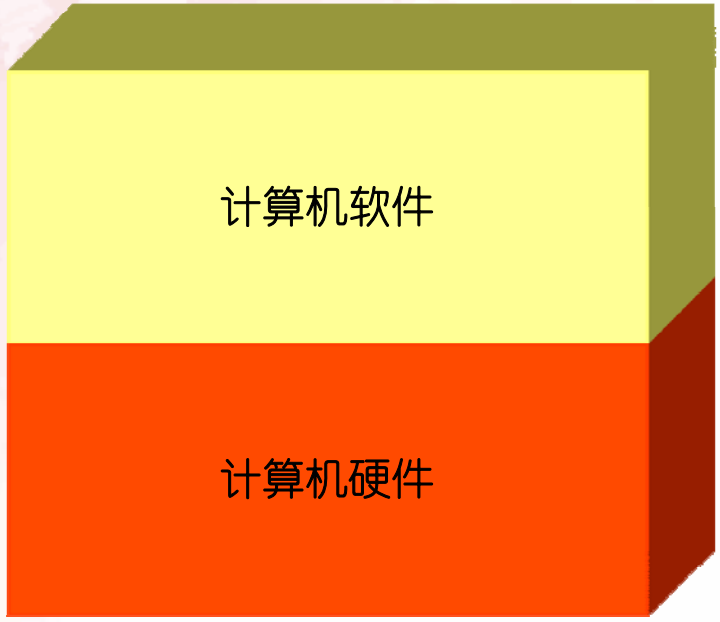
- 计算的对象、中间结果和最终结果都在带上，程序则在控制器中。这意味着什么？
- 如果把这样的图灵机做成一台计算机，由于程序是固定 ，那么这样的计算机就只能完成规则固定的计算（但输入可以多样化），因此是一台专用计算机。
- 于是想到，把计算用的程序也放在带上，而控制器中的程序能够从带上把计算用的程序中的指令逐条地读进来，再按照其要求进行计算（这个过程叫做解释（Interpretation），以后的《计算机组成原理》和《编译原理》等课程中会学到）。
  - 具有这种能力的图灵机叫做通用图灵机。
  - 这就是 von Neumann 体系结构的基本思想。

感兴趣同学

- 图灵停机问题
- 机器能否具有与人一样的智能？
- 人工智能是否在广义上可行？

从图灵机到计算机系统

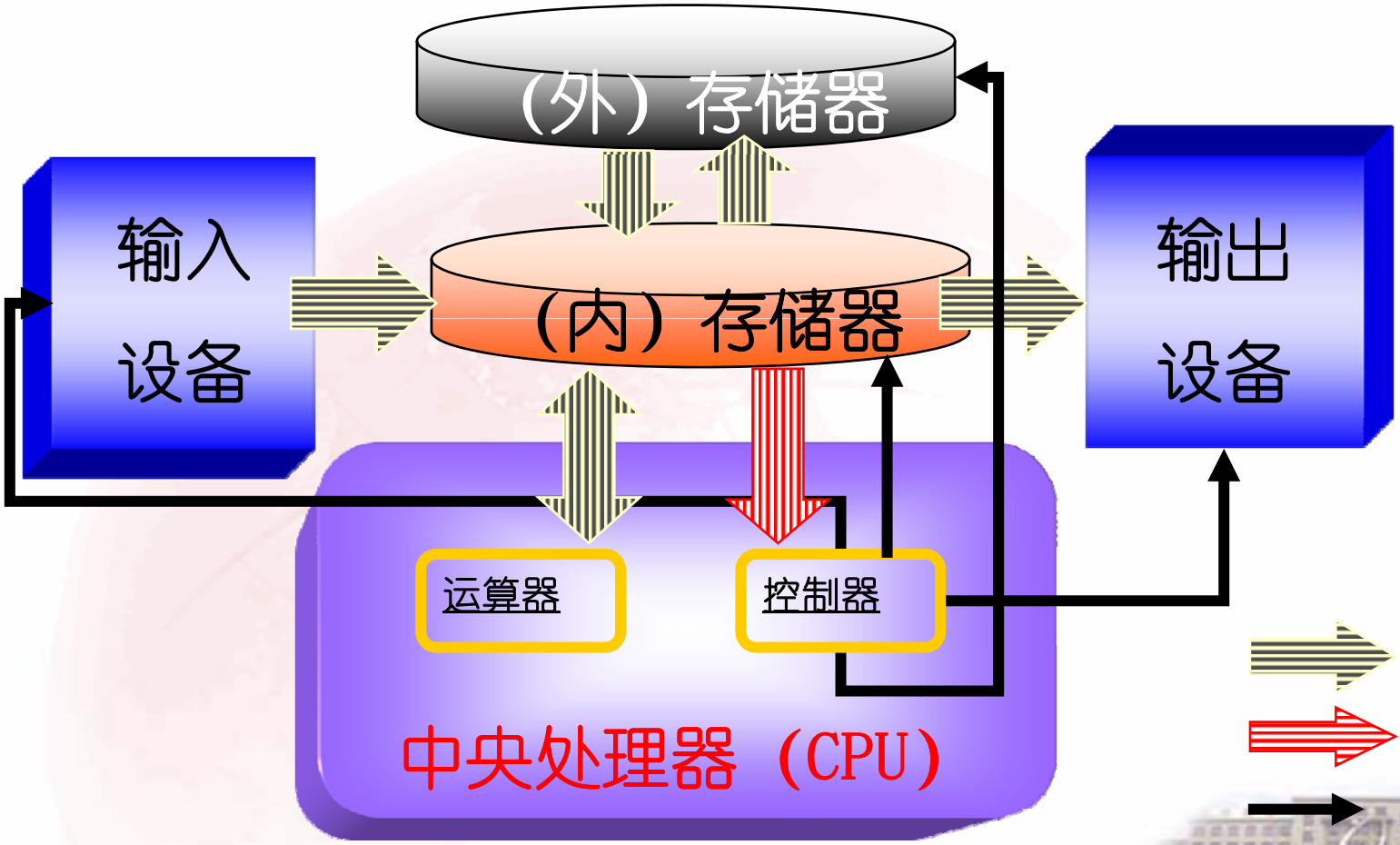
- 计算机系统是由计算机硬件和软件组成的
- 从功能上看，计算机软件极大地提高和扩展了计算机硬件的能力，计算机硬件为软件的存在提供了物质基础a

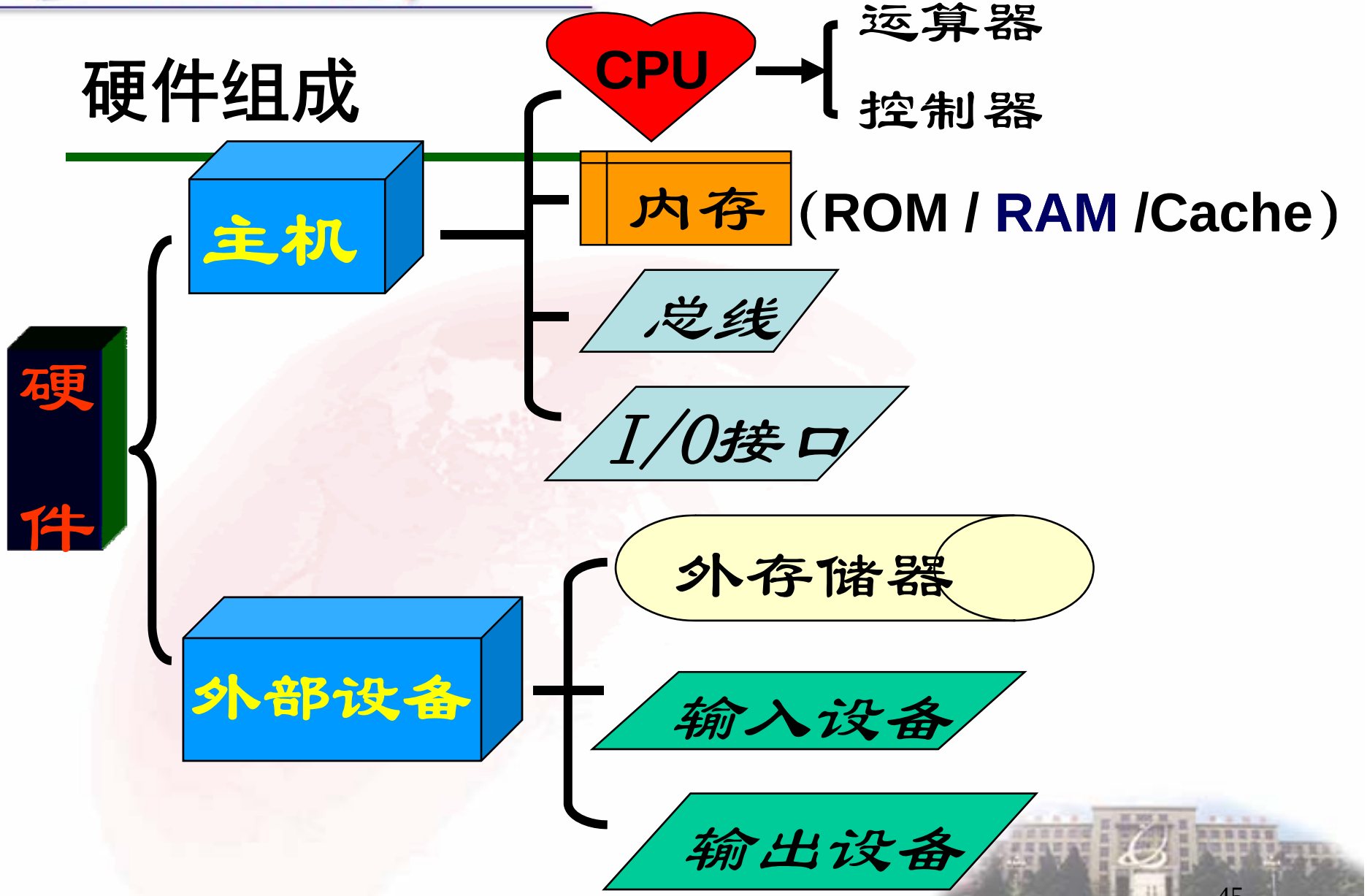


1、软件实现丰富各种功能

2、各自发展

计算机硬件基本组成




硬件组成
硬件
主机
CPU
运算器
控制器
内存
(ROM / RAM /Cache)
总线
I/O接口
外部设备
外存储器
输入设备
输出设备

Dev C++　　VC6.0　C4droid

a