

软件工程专业导论

战德臣

哈尔滨工业大学 教授·博士生导师
教育部大学计算机课程教学指导委员会委员

Research Center on Intelligent
Computing for Enterprises & Services,
Harbin Institute of Technology

第2讲 计算思维起步

--符号化、计算化与自动化

战德臣

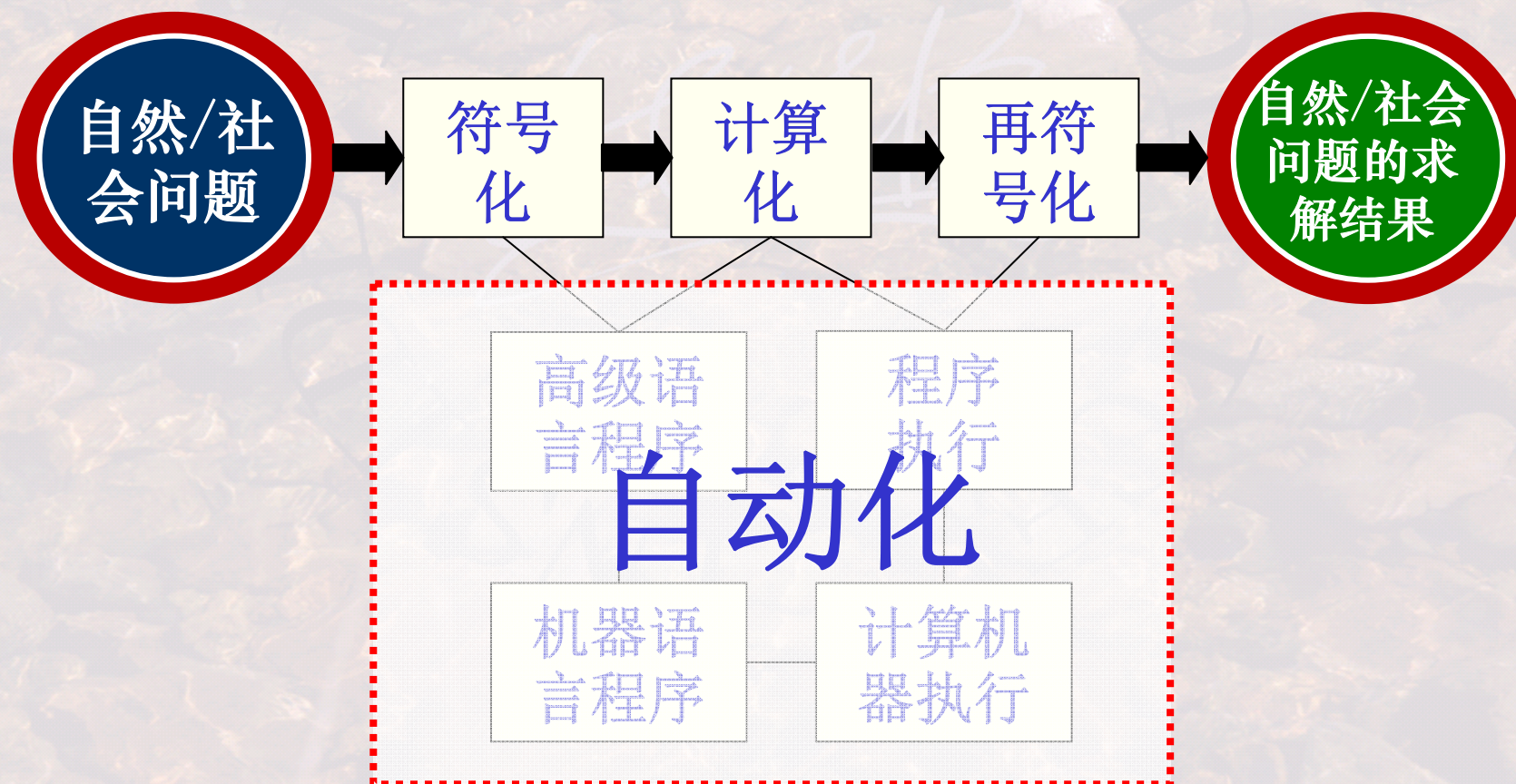
哈尔滨工业大学 教授·博士生导师
教育部大学计算机课程教学指导委员会委员

Research Center on Intelligent
Computing for Enterprises & Services,
Harbin Institute of Technology

●软件工程学科的基本思维模式

软件研究的终极目的是解决社会/自然问题，如何解决呢？

◆将社会/自然问题用符号表达，基于符号进行计算，将计算用软件来实现，是解决社会/自然问题的基本思维模式



符号化-计算化

----示例1：思维能否计算？

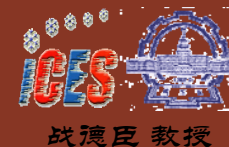
战德臣

哈尔滨工业大学 教授.博士生导师
教育部大学计算机课程教学指导委员会委员

Research Center on **I**ntelligent
Computing for **E**nterprises & **S**ervices,
Harbin **I**nstitute of **T**echnology

符号化-计算化--示例1：思维能否计算？

(1)典型的思维模式--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**。**命题**由语句表述，即内容为“真”或为“假”的一个判断语句！**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。

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。

能否进行符号化计算呢？



符号化-计算化--示例1：思维能否计算？

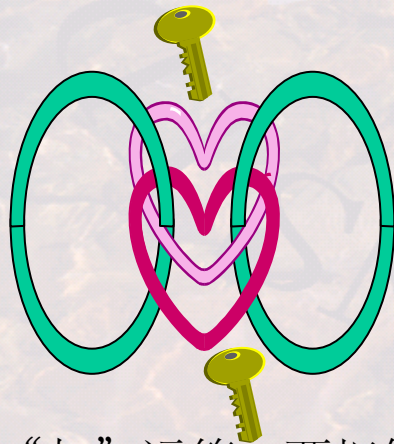
(2)逻辑可以符号化并进行基于符号的计算

逻辑的符号化及基本运算

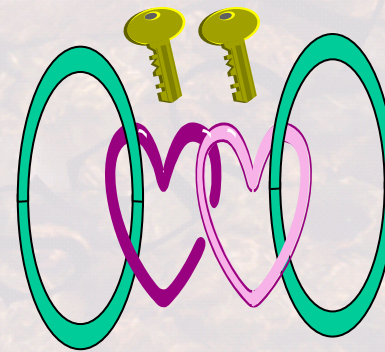
一个命题由 X, Y, Z 等表示，其值可能为“真”或为“假”。

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：只要有任何一把钥匙便能开门

符号化-计算化--示例1：思维能否计算？

(3)逻辑运算是基于0和1的符号计算？

用0和1来表示逻辑运算

■ “与”运算 **AND**：

有0为0，全1为1

	0	0	1	1
AND	0	1	0	1
	0	0	0	1

■ “或”运算 **OR**：

有1为1，全0为0

	0	0	1	1
OR	0	1	0	1
	0	1	1	1

■ “非”运算 **NOT**：

非0则1，非1则0

NOT	0	1
	1	0

■ “异或”运算 **XOR**：

相同为0，不同为1

	0	0	1	1
XOR	0	1	0	1
	0	1	1	0

注：1表示 真，0表示 假

这些是最基本最重要的运算，可要记住哟...

符号化-计算化--示例1：思维能否计算？

(4)逻辑中的推理可以基于符号计算来进行？



用符号表达推理--进行计算

三个符号**A**、**B**、**C**表达三个命题，其值为0或1

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

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

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

复杂吗？耐心看看，只是基本运算的组合？还复杂吗？...

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：

$\langle A=1, B=0, C=0 \rangle$ 。

符号化-计算化--示例1：思维能否计算？

(5)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](#)

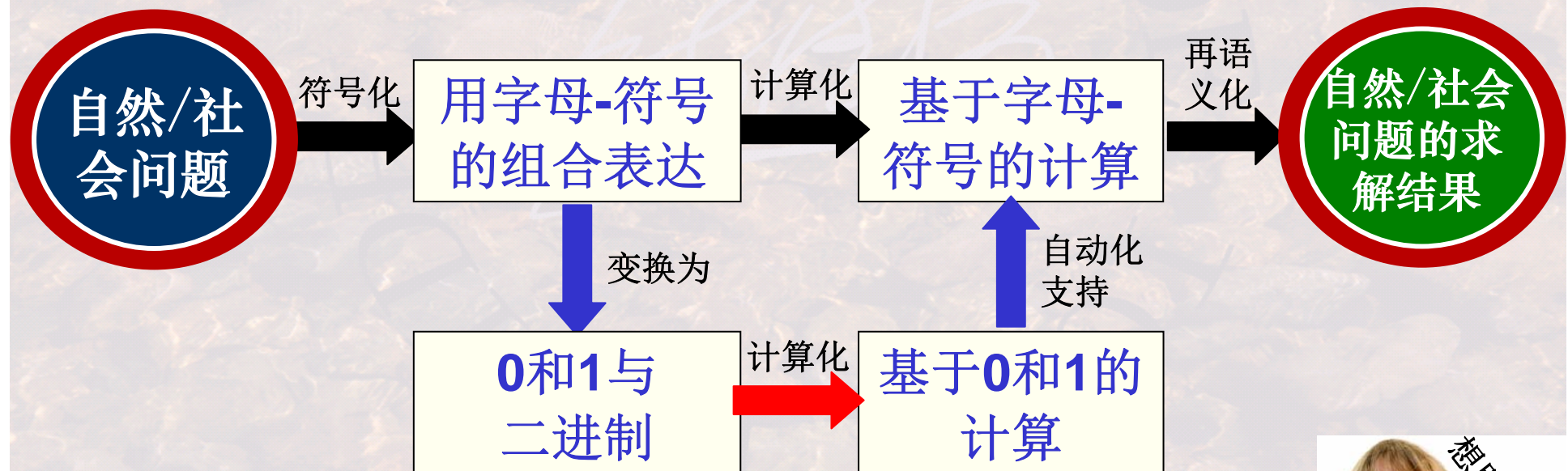
典型概念：**布尔量，布尔值，布尔运算，布尔操作**

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

符号化-计算化--示例1：思维能否计算？

(6)由示例推而广之--符号化-计算化思维

符号化-计算化



符号化-计算化

----示例2：图像能否隐藏信息？

战德臣

哈尔滨工业大学 教授.博士生导师
教育部大学计算机课程教学指导委员会委员

Research Center on **I**ntelligent
Computing for **E**nterprises & **S**ervices,
Harbin **I**nstitute of **T**echnology

符号化-计算化--示例2：图像能否隐藏信息？

(1)信息隐藏问题

➤这幅图像背后其实是隐藏着信息的,你相信吗？

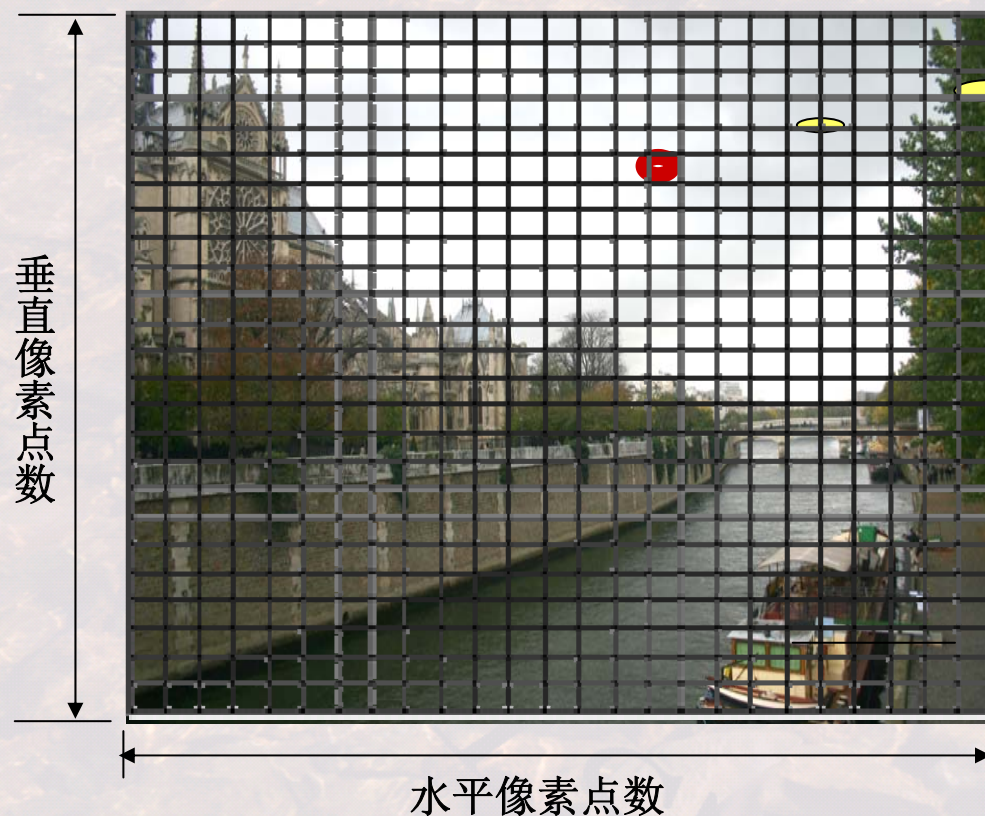


这是怎么做到的呢？



符号化-计算化--示例2：图像能否隐藏信息？

(2)图像是如何表示的



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

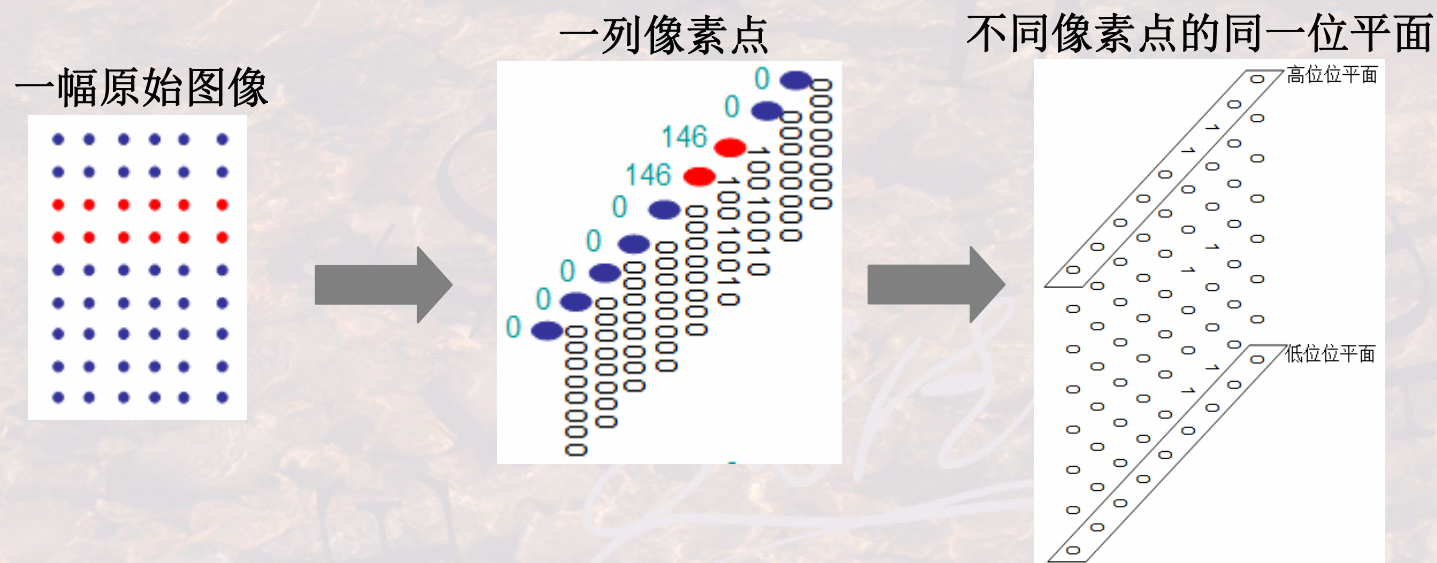
0010 --16种组合

10010010 --256种组合

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

符号化-计算化--示例2：图像能否隐藏信息？

(3)图像的位平面—低位位平面和高位位平面及其作用



十进制

146

二进制

1 0 0 1 0 0 1 0
7 6 5 4 3 2 1 0
2⁷ 2⁶ 2⁵ 2⁴ 2³ 2² 2¹ 2⁰

10010010

—— 高位

—— 低位

10010010 (146)
00010010 (18)

10010010 (146)
10010011 (147)

符号化-计算化--示例2：图像能否隐藏信息？

(5)基于0和1的“位操作”

基于0和1的计算：位操作

- “与” 运算 **AND**:

有0为0，全1为1

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

- “或” 运算 **OR**:

有1为1，全0为0

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

- “非” 运算 **NOT**:

非0则1，非1则0

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

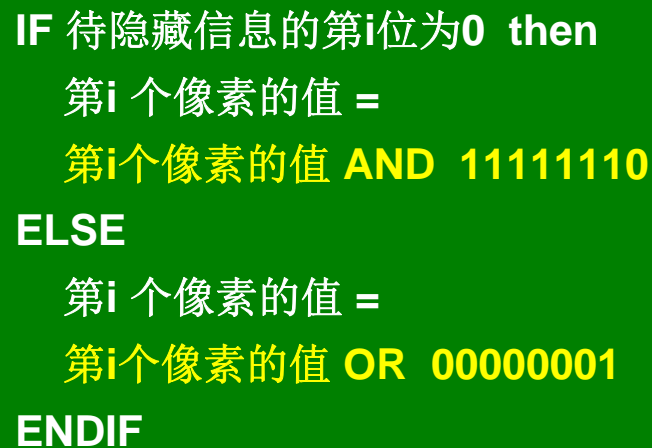
- “异或” 运算 **XOR**:

相同为0，不同为1

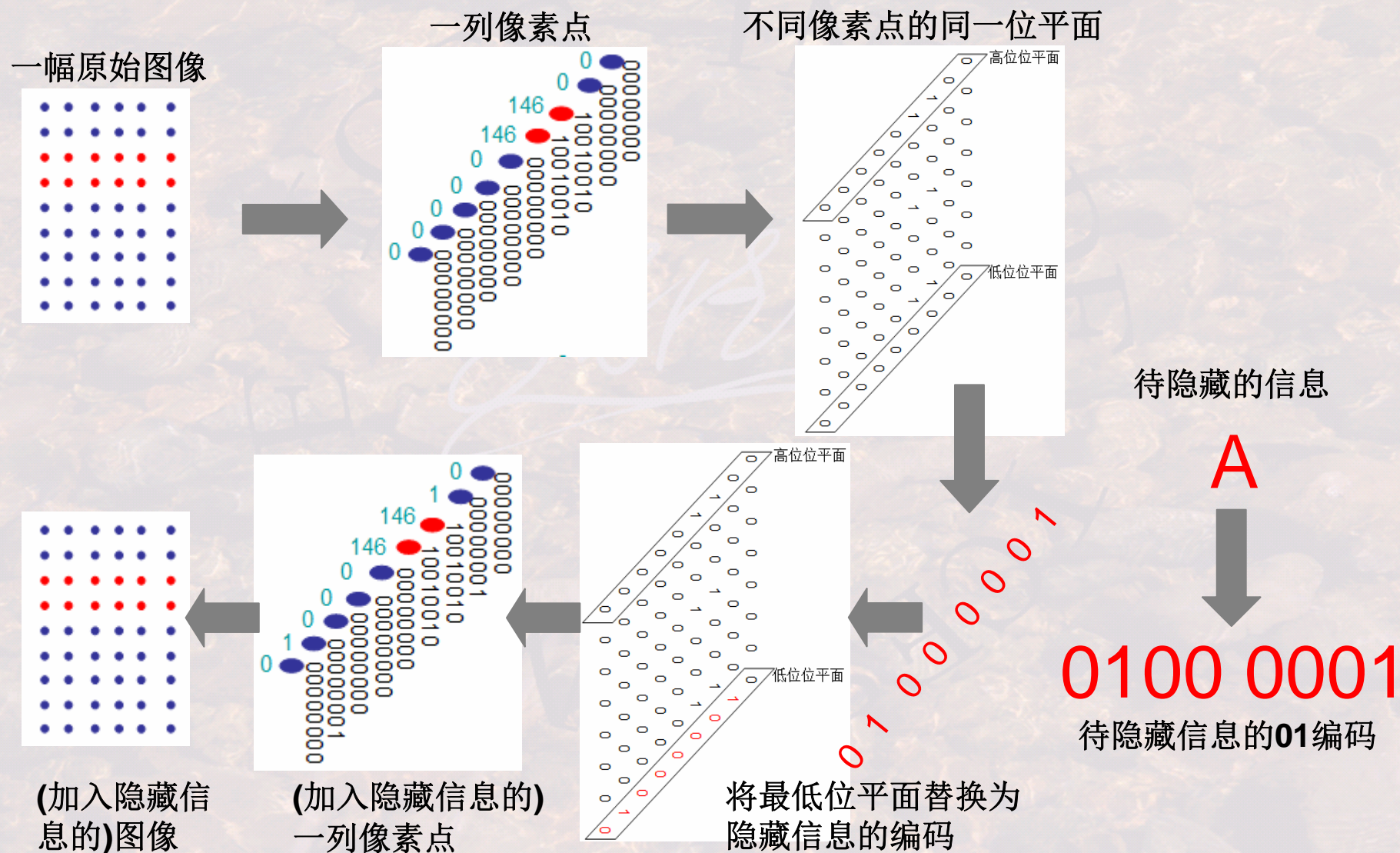
$$\begin{array}{r} 10001110 \\ \text{AND } 00010000 \\ \hline 00000000 \end{array}$$

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

ICES 战德臣 教授



符号化-计算化--示例2：图像能否隐藏信息？ (7)完整的思路？



符号化-计算化--示例2：图像能否隐藏信息？ (8)能否隐藏信息呢？

➤这幅图像背后其实是隐藏着信息的？你相信吗？



符号化-计算化--示例2：图像能否隐藏信息？

(8)能否隐藏信息呢？

➤这幅图像背后是否也隐藏着什么信息呢？



间谍
信息？

违法
信息？

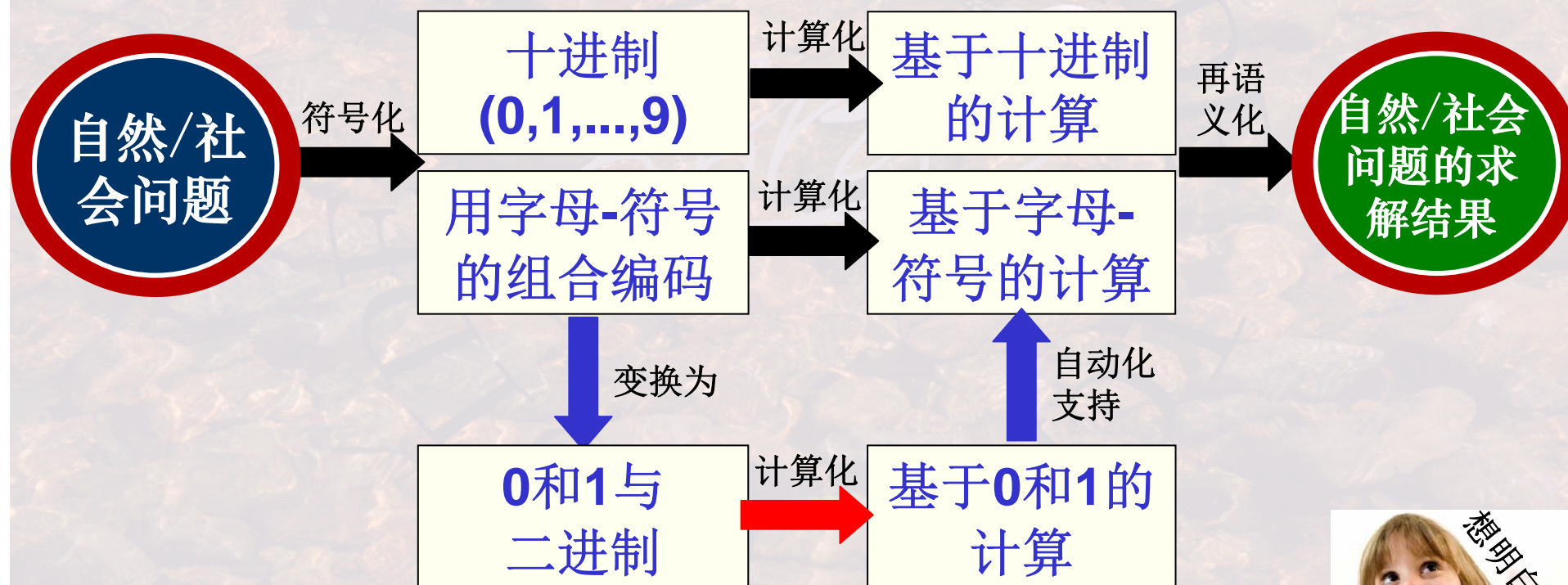
恶意
程序？

隐私
信息？

相互要
交换的
信息？

符号化-计算化--示例2：图像能否隐藏信息？ (10)推而广之-符号化计算化？

符号化-计算化



计算化-自动化

----能否由硬件实现计算？

战德臣

哈尔滨工业大学 教授.博士生导师
教育部大学计算机课程教学指导委员会委员

Research Center on **I**ntelligent
Computing for **E**nterprises & **S**ervices,
Harbin **I**nstitute of **T**echnology

计算化-自动化：能否由硬件实现计算？

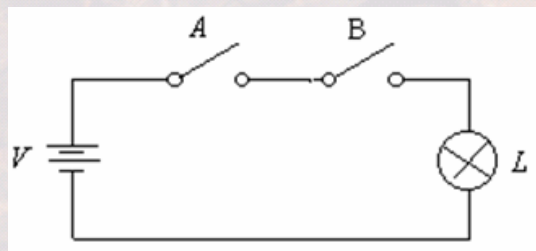
(1)如何用电信号及电子元件表达0和1？

实现0和1：电信号和继电器开关

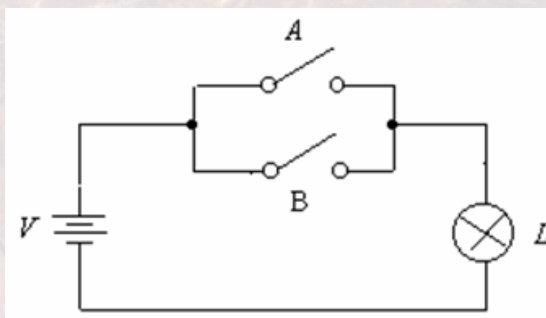


数字信号：高电平为1，低电平为0

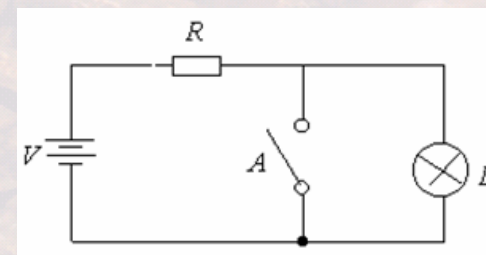
◆用继电器开关实现基本逻辑运算



“与”运算电路



“或”运算电路



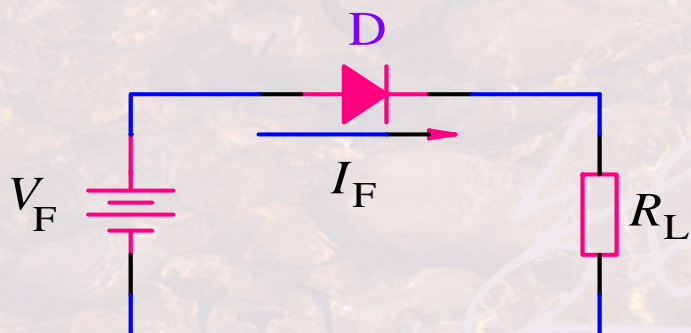
“非”运算电路

计算化-自动化：能否由硬件实现计算？

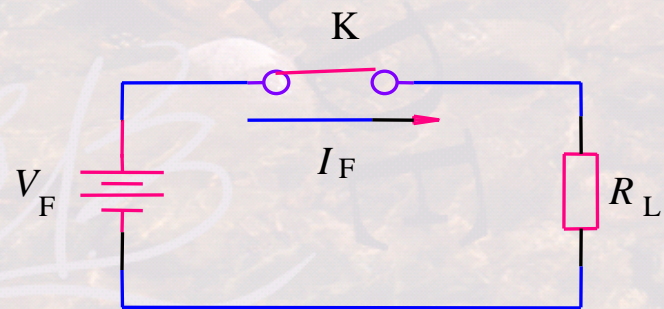
(2)处理0和1的基本元件？

实现0和1的基本元器件：二极管

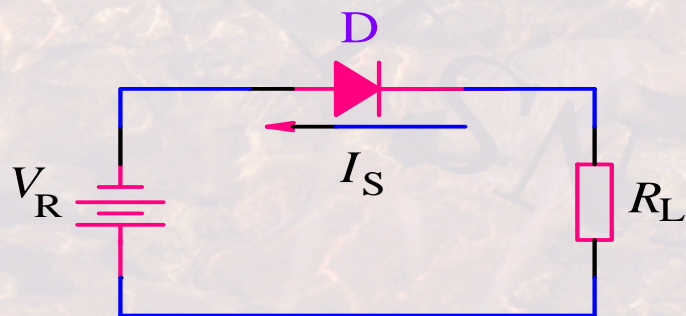
◆二极管的基本特性



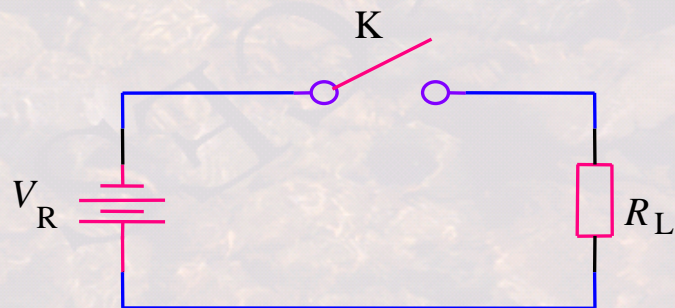
(a)



(b)



(a)



(b)

计算化-自动化：能否由硬件实现计算？

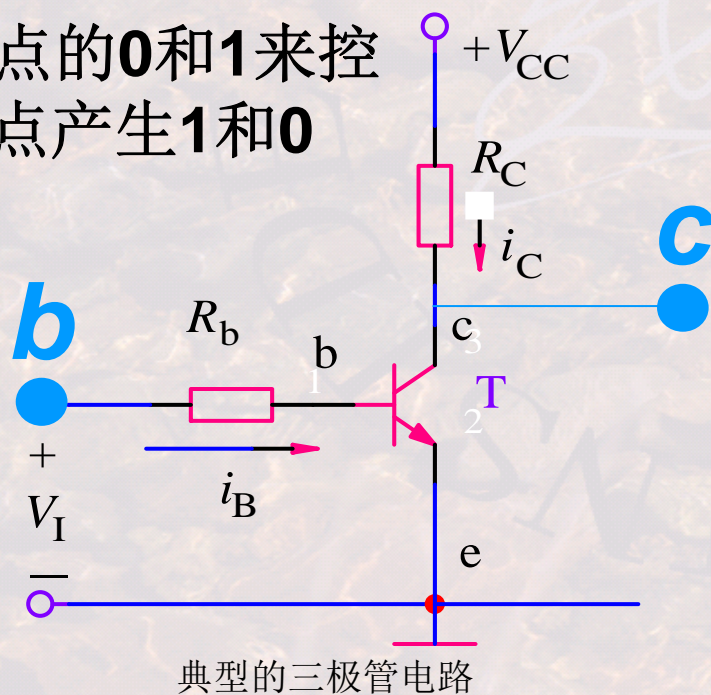
(2)处理0和1的基本元件？

实现0和1的基本元器件：三极管

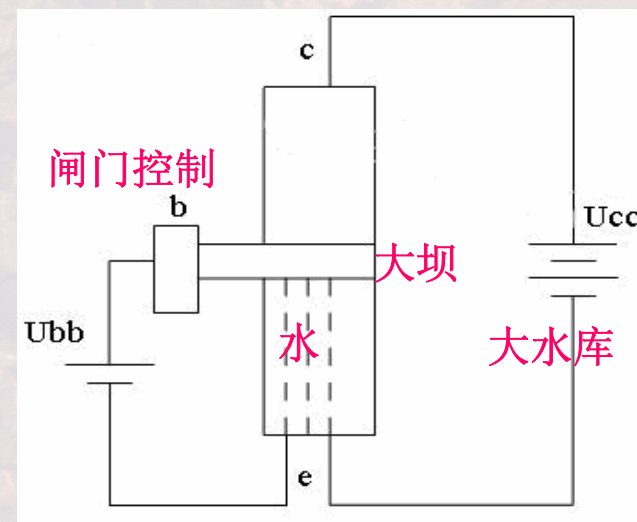
◆ 三极管的基本特性：

- 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



第一个三极管试验装置

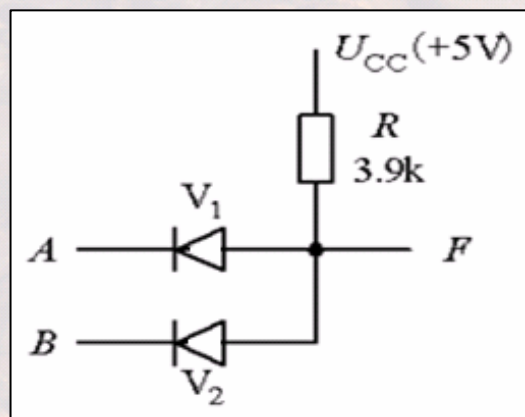


计算化-自动化：能否由硬件实现计算？

(3)用基本电子元件实现基本逻辑运算？

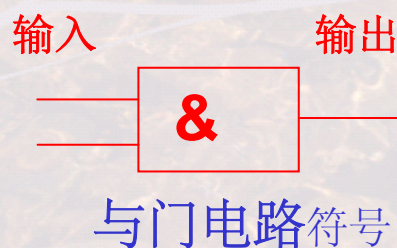
用二极管、三极管可实现基本的集成电路：与门、或门和非门

◆这些电路被封装成集成电路(芯片)，即所谓的门电路。



“与”门电路

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

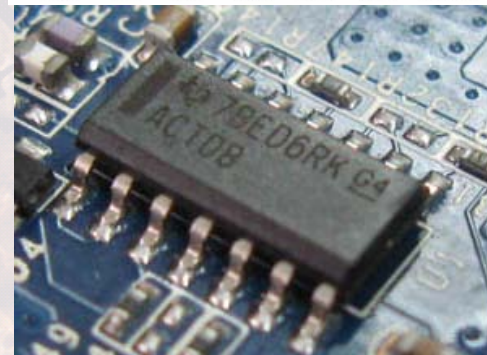
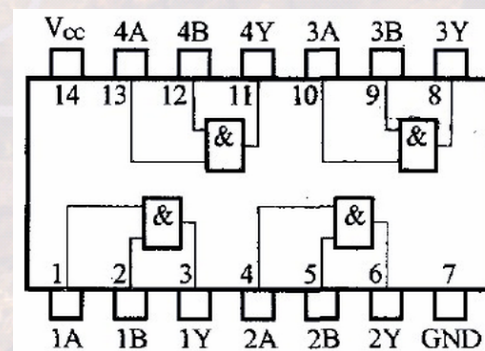


$$1Y = 1A \text{ and } 1B$$

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

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

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

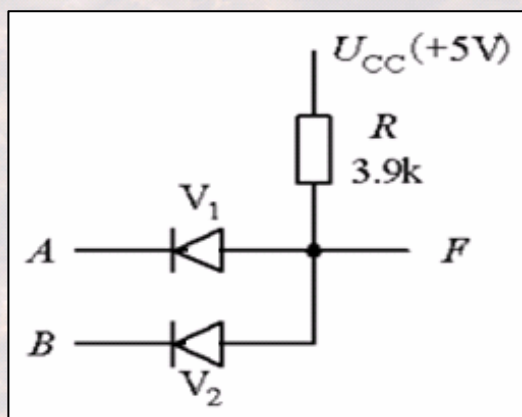


计算化-自动化：能否由硬件实现计算？

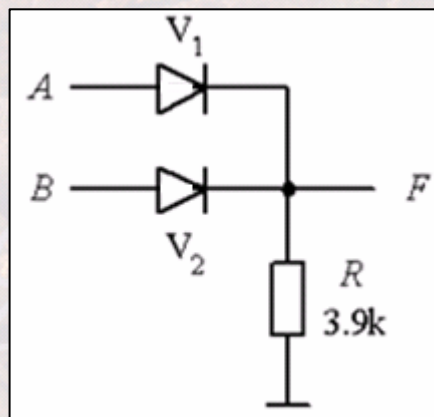
(3)如何用基本电子元件实现基本逻辑运算？

用二极管、三极管可实现基本的集成电路：与门、或门和非门

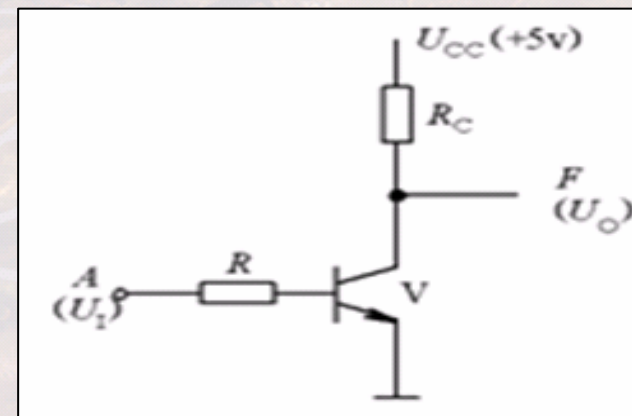
◆ 这些电路被封装成集成电路(芯片)，即所谓的门电路。



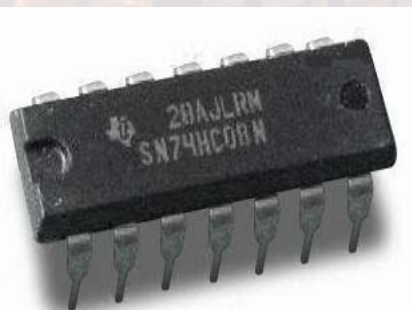
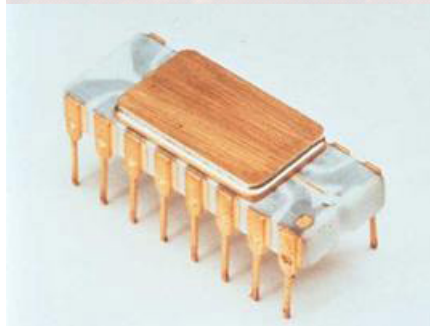
“与” 门电路



“或” 门电路



“非” 门电路



计算化-自动化：能否由硬件实现计算？

(4)门电路的符号化-计算化

基本门电路的符号表示及其特性

◆**与门电路**：是实现逻辑与运算的集成电路，即：只有当两个输入端为高电平(1)时，则输出端为高电平(1)；否则，输出端为低电平(0)。

◆**或门电路**：是实现逻辑或运算的集成电路，即：只有当两个输入端为低电平(0)时，则输出端为低电平(0)；否则，输出端为高电平(1)。

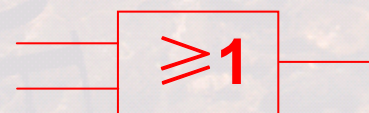
◆**非门电路**：是实现逻辑非运算的集成电路，即：当输入端为高电平(1)时，则输出端为低电平(0)；输入端为低电平(0)时，则输出端为高电平(1)。

◆**异或门电路**：是实现逻辑异或运算的集成电路，即：当两个输入端同为高电平(1)或同为低电平(0)时，则输出端为低电平(0)；否则，输出端为高电平(1)。

输入(左进) 输出(右出)



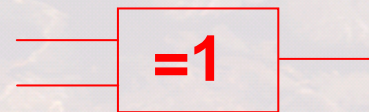
与门电路符号



或门电路符号



非门电路符号



异或门电路符号

为什么要符号化呢？



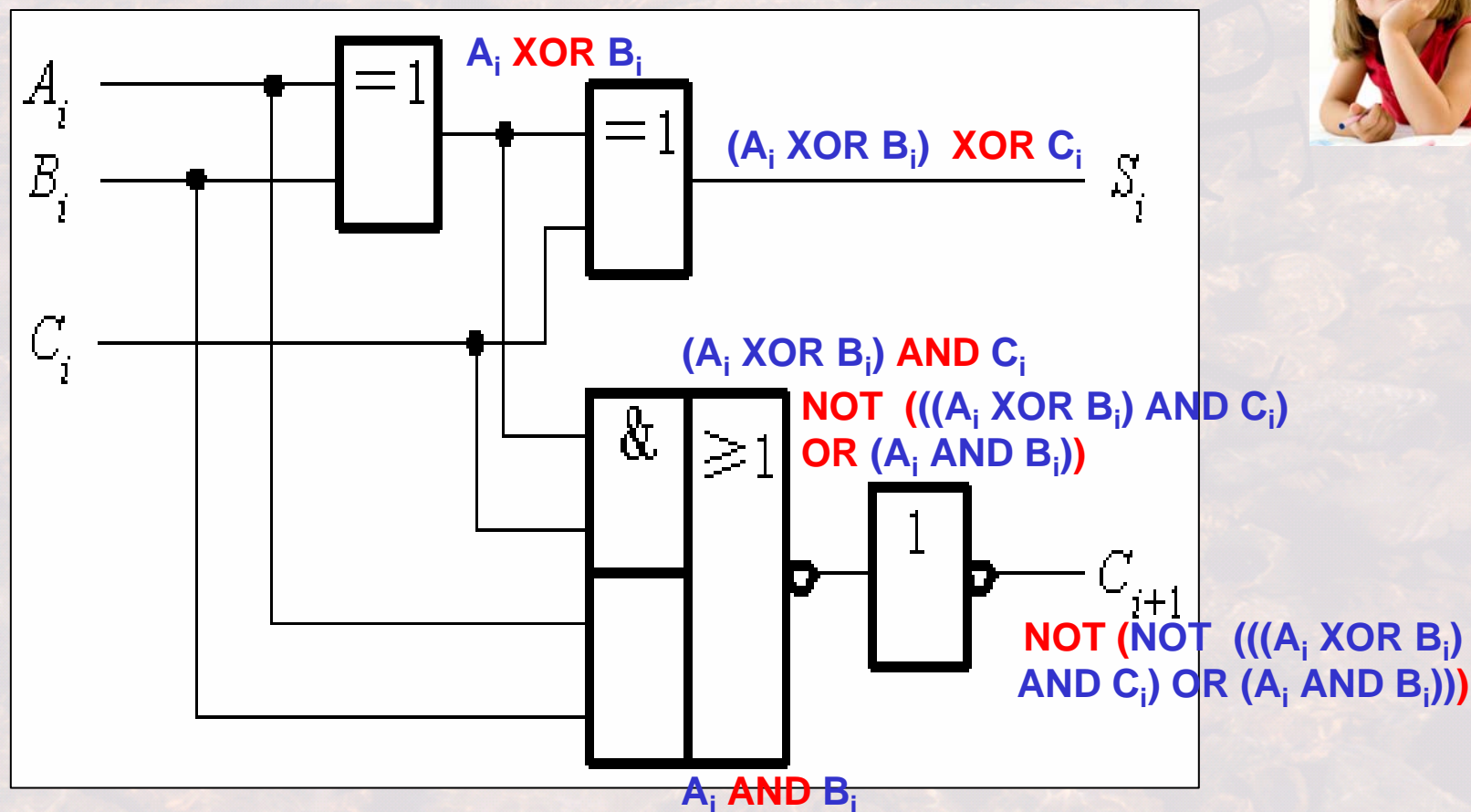
计算化-自动化：能否由硬件实现计算？

(5) 如何用已实现的基本逻辑运算(门电路)来实现更复杂的运算？

基于门电路的复杂组合逻辑电路

◆ 示例1：一位加法器的示例。

这是否是电路版的逻辑运算组合表达式呢

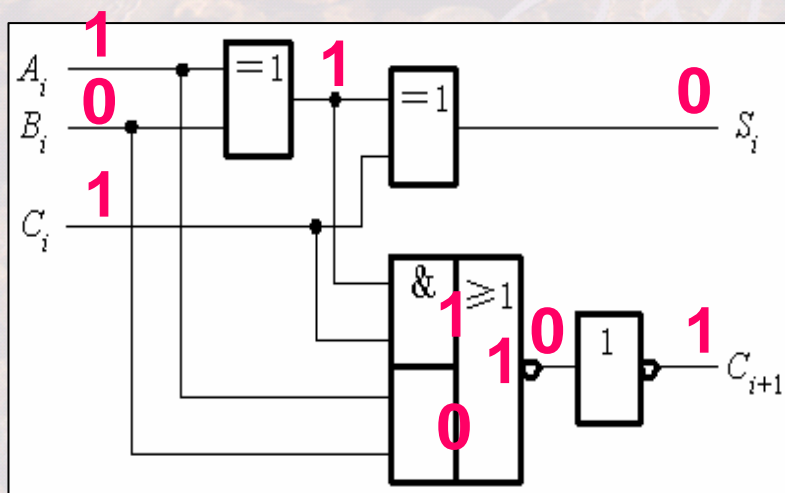


计算化-自动化：能否由硬件实现计算？

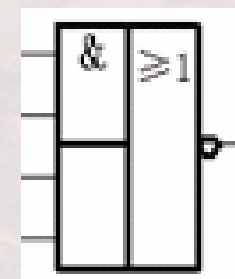
(5)如何用已实现的基本逻辑运算(门电路)来实现更复杂的运算？

基于门电路的复杂组合逻辑电路

◆可验证一位加法器实现的正确性。



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

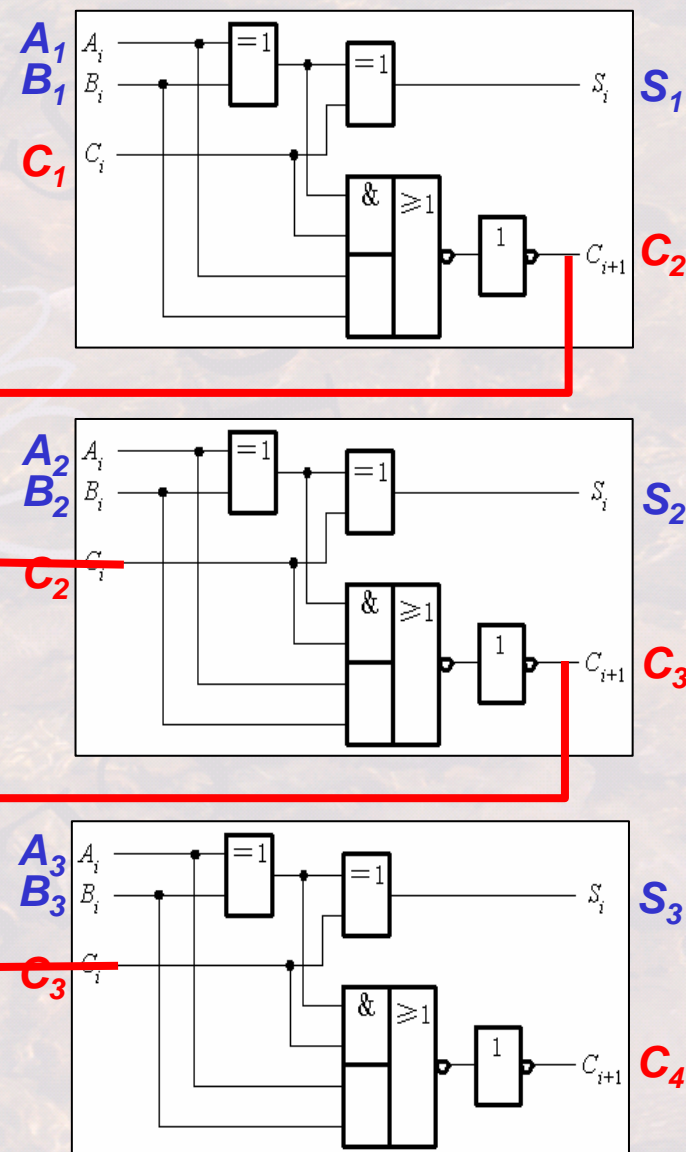


(5)如何用已实现的基本逻辑运算(门电路)来实现更复杂的运算?

基于门电路的复杂组合逻辑电路

- ◆ 示例：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。

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



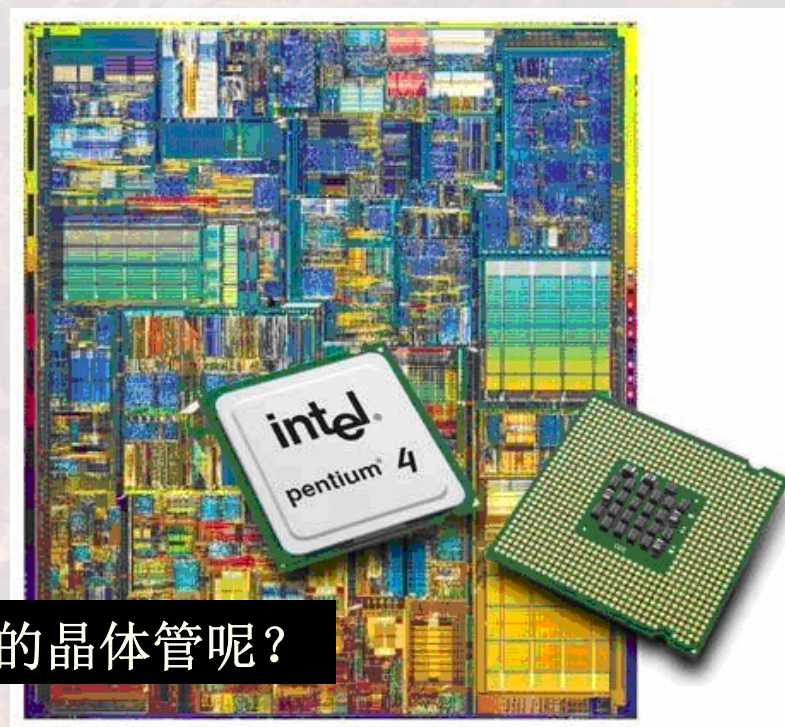
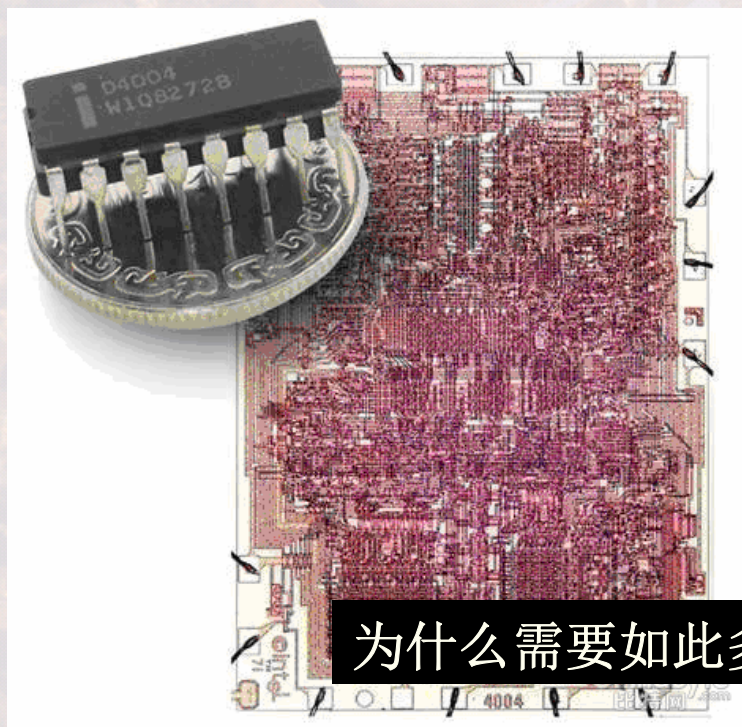
计算化-自动化：能否由硬件实现计算？

(6)复杂硬件部件的实现是否都可以这样呢？

复杂部件的硬件实现(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Intel4004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？

计算化-自动化

-----怎样进行复杂计算？

战德臣

哈尔滨工业大学 教授.博士生导师
教育部大学计算机课程教学指导委员会委员

Research Center on **I**ntelligent
Computing for **E**nterprises & **S**ervices,
Harbin **I**nstitute of **T**echnology

假设前面能实现加法运算了，也能实现减法、乘除法运算

---能执行运算的部件，称为“**运算器**”

---那能实现复杂的运算吗？怎样实现复杂的运算呢？

$$8 \times 3^2 + 2 \times 3 + 6$$

$$a \times x^2 + b \times x + c$$

计算化-自动化：怎样进行复杂计算？

(2)机器级算法？

可在机器上执行的求解问题的操作规则及步骤, 被称为可执行的算法。

$$\text{计算 } 8 \times 3^2 + 2 \times 3 + 6 = ((8 \times 3) + 2) \times 3 + 6$$

计算方法1

- Step1:** 取出数3至运算器中
- Step2:** 乘以数3在运算器中
- Step3:** 乘以数8在运算器中
- Step4:** 存结果 8×3^2 在存储器中
- Step5:** 取出数2至运算器中
- Step6:** 乘以数3在运算器中
- Step7:** 加上 (8×3^2) 在运算器中
- Step8:** 加上数6在运算器中

计算方法2

- Step1:** 取出数3至运算器中
- Step2:** 乘以数8在运算器中
- Step3:** 加上数2在运算器中
- Step4:** 乘以数3在运算器中
- Step5:** 加上数6至运算器中

这个算法机器
能够执行吗？

能执行这种算
法的机器是怎
样的？



计算化-自动化：怎样进行复杂计算？

(3)冯·诺依曼计算机与存储程序？

冯·诺依曼(Von.Neumann)计算机

◆1944~1945年间，冯·诺伊曼提出“存储程序”的计算机设计思想，并进行了实践，现代计算机普遍来讲属于冯·诺伊曼机体系。

◆冯·诺伊曼机的基本思想：

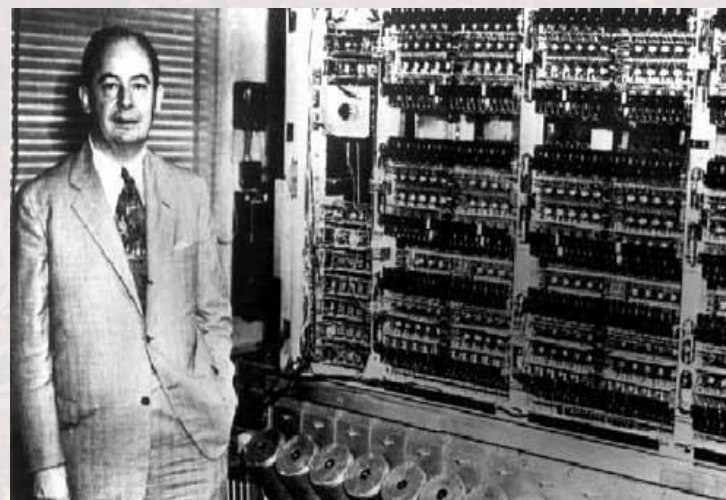
- 运算和存储分离

- 存储程序：指令和数据以同等地位事先存于存储器，可按地址寻访，连续自动执行。

- 五大部件构成：运算器、控制器、存储器、输入设备和输出设备

- 指令和数据用二进制表示，指令由操作码和地址码组成

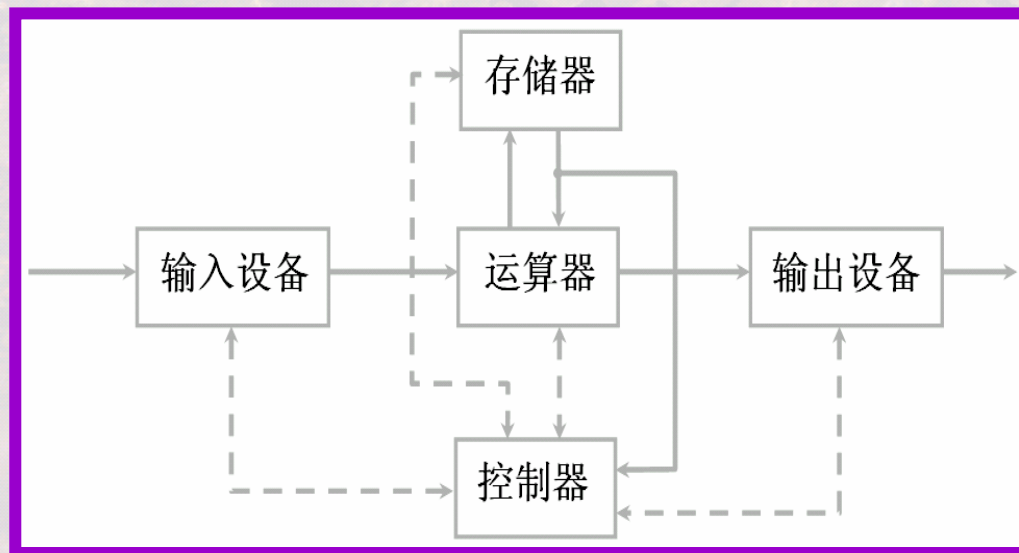
- 以运算器为中心，控制器负责解释指令，运算器负责执行指令



计算化-自动化：怎样进行复杂计算？

(4)冯·诺依曼计算机的结构？

以运算器和以存储器为中心的冯·诺依曼计算机



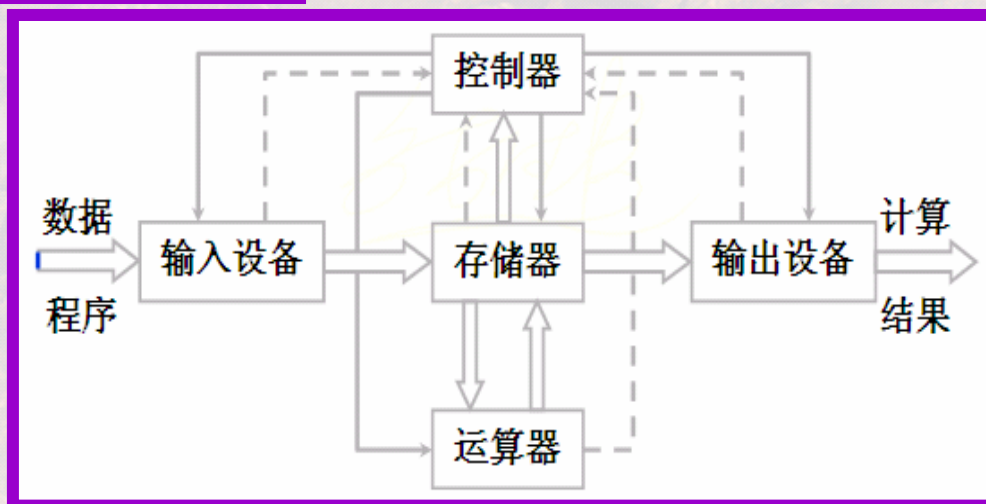
什么是“程序”，
“指令”和
“数据”？

控制器有什么
作用呢？

“运算”为什么
要和“存储”分
离呢？

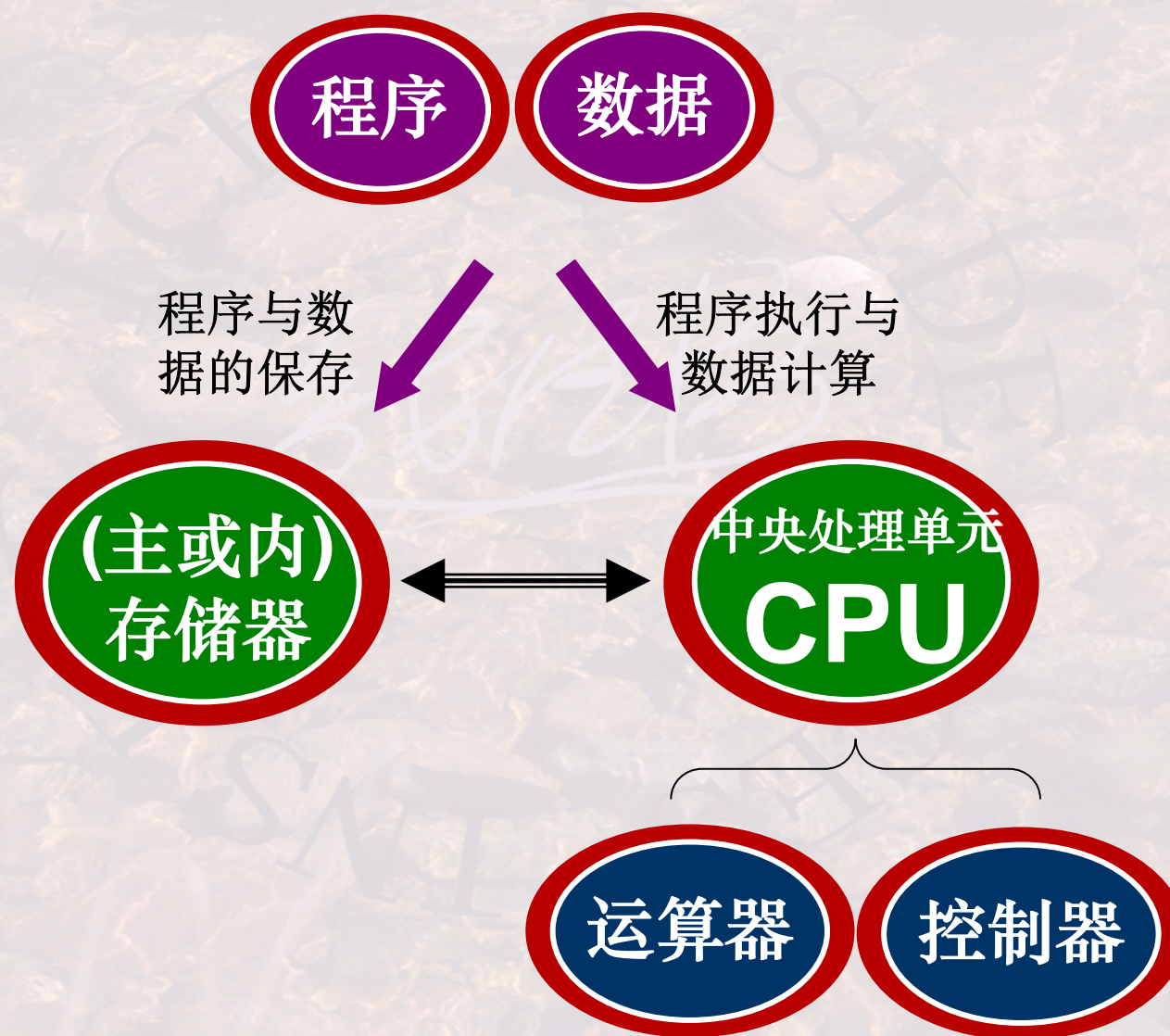


为什么要以存
储器为中心呢？



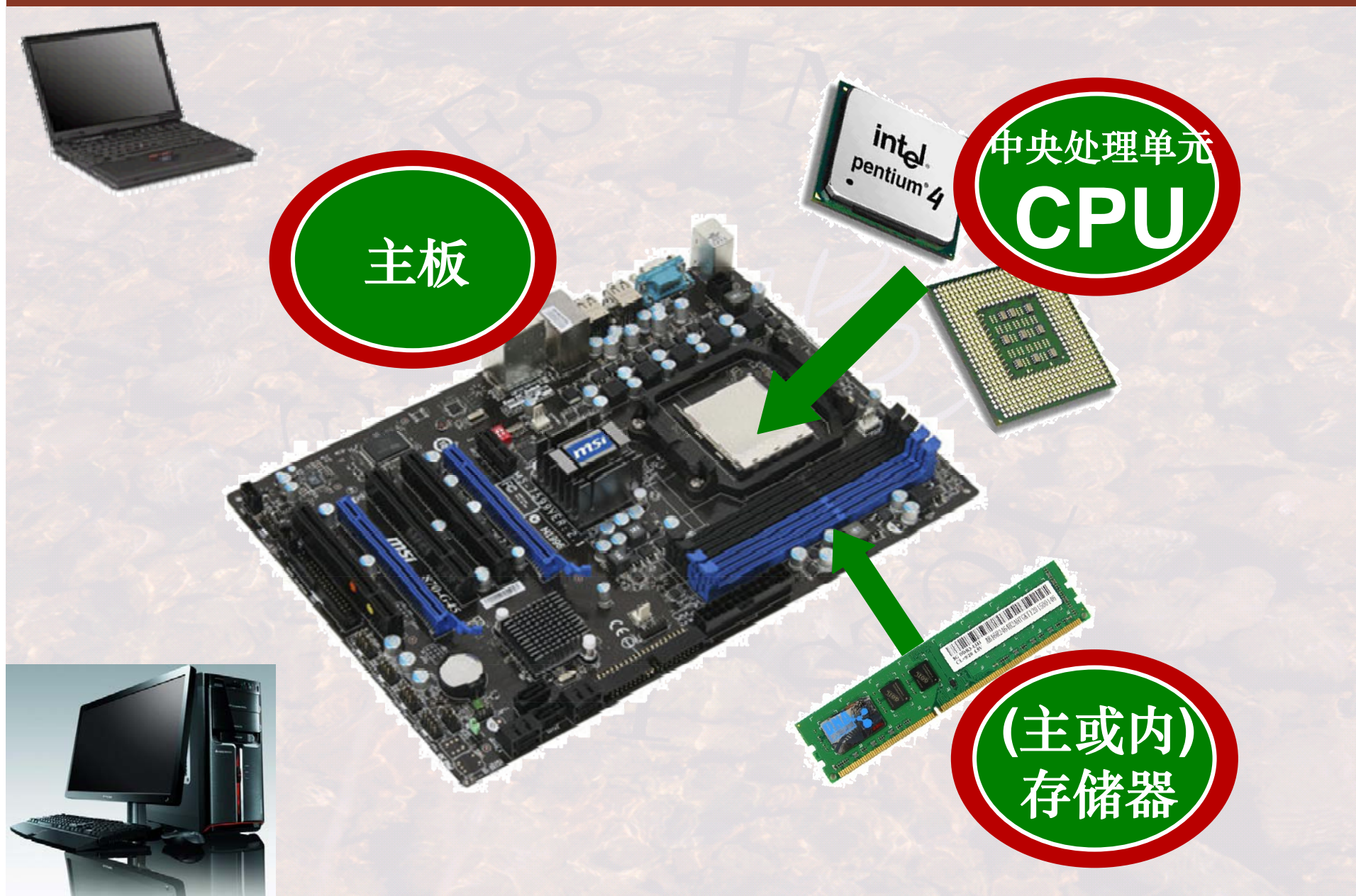
计算化-自动化：怎样进行复杂计算？

(5)现代计算机-计算环境



计算化-自动化：怎样进行复杂计算？

(5)现代计算机-计算环境



计算化-自动化：怎样进行复杂计算？

(6)机器指令系统？

机器指令-机器语言

◆ 机器指令是CPU可以直接分析并执行的指令，由0和1编码表示。

◆ 指令 \approx 操作码 + 地址码；

操作码

地址码

000001 00 00000111

000100 00 00001010

(如取数，加法等操作) (操作中的数据的来源)

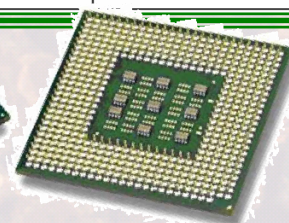
000001 0000000100

000001 0000001100

000001 0000001000

机器指令		对应的功能
操作码	地址码	
取数	α	α 号存储单元的数 取出送到运算器；
000001	0000000100	
存数	β	运算器中的数 存储到 β 号存储单元；
000010	0000010000	
加法	γ	运算器中的数 加上 γ 号存储单元的数，结果保留在运算器；
000011	0000001010	
乘法	δ	运算器中的数 乘以 δ 号存储单元的数，结果保留在运算器；
000100	0000001001	
打印		打印指令
000101	0000001100	
停机		停机指令
000110		

这是否也是
符号化呢？



计算化-自动化：怎样进行复杂计算？ (7)用机器指令表达机器级算法？

$$((8 \times 3) + 2) \times 3 + 6$$

计算方法2

机器级 算法

- Step1:** 取出数3至运算器中
Step2: 乘以数8在运算器中
Step3: 加上数2在运算器中
Step4: 乘以数3在运算器中
Step5: 加上数6至运算器中

机器 级程序

000001	0000001000
000100	0000001001
000011	0000001010
000100	0000001000
000011	0000001011
000010	0000001100
000101	0000001100
000110	

机器指令		对应的功能
操作码	地址码	
取数	α	α 号存储单元的数 取出送到运算器；
000001	0000000100	
存数	β	运算器中的数 存储到 β 号存储单元；
000010	0000010000	
加法	γ	运算器中的数 加上 γ 号存储单元的数，结果保留在运算器；
000011	0000001010	
乘法	δ	运算器中的数 乘以 δ 号存储单元的数，结果保留在运算器；
000100	0000001001	
打印		打印指令
000101	0000001100	
停机		停机指令
000110		

机器 指令

“3” 存储在8号存储单元
 “8” 存储在9号存储单元
 “2” 存储在10号存储单元
 “6” 存储在11号存储单元

计算化-自动化：怎样进行复杂计算？

(8)将机器级程序和数据装载进存储器中？



存储器

计算 $8 \times 3^2 + 2 \times 3 + 6$ 的程序；
计算 $ax^2 + bx + c$ 的程序。

机器级程序

对应的十进制地址	存储单元的地址	存储单元的内容		说明
		操作码	地址码	
0	00000000 00000000	000001	0000001000	指令：取出 8 号存储单元的数(即 3)至运算器中
1	00000000 00000001	000100	0000001001	指令：乘以 9 号存储单元的数(即 8)得 8×3 在运算器中
2	00000000 00000010	000011	0000001010	指令：加上 10 号存储单元的数(即 2)得 $8 \times 3 + 2$ 在运算器中
3	00000000 00000011	000100	0000001000	指令：乘以 8 号存储单元的数(即 3) 得 $(8 \times 3 + 2) \times 3$ 在运算器中
4	00000000 00000100	000011	0000001011	指令：加上 11 号存储单元的数(即 6)得 $8 \times 3^2 + 2 \times 3 + 6$ 至运算器中
5	00000000 00000101	000010	0000001100	指令：将上述运算器中结果存于 12 号存储单元
6	00000000 00000110	000101	0000001100	指令：打印
7	00000000 00000111	000110		指令：停机
8	00000000 00001000	000000	0000000011	数据：数 3 存于 8 号单元
9	00000000 00001001	000000	00000001000	数据：数 8 存于 9 号单元
10	00000000 00001010	000000	00000000010	数据：数 2 存于 10 号单元
11	00000000 00001011	000000	000000000110	数据：数 6 存于 11 号单元
12	00000000 00001100			数据：存放结果

程序与数据以同等地位存于存储器中

计算化-自动化

----程序怎样被执行？

战德臣

哈尔滨工业大学 教授.博士生导师
教育部大学计算机课程教学指导委员会委员

Research Center on **I**ntelligent
Computing for **E**nterprises & **S**ervices,
Harbin **I**nstitute of **T**echnology

计算化-自动化：程序怎样被执行

(1)装配一台简单的计算机—存储器

- 存储单元地址
- 存储单元内容

存储器
(内存)

内容 00000100 00001000

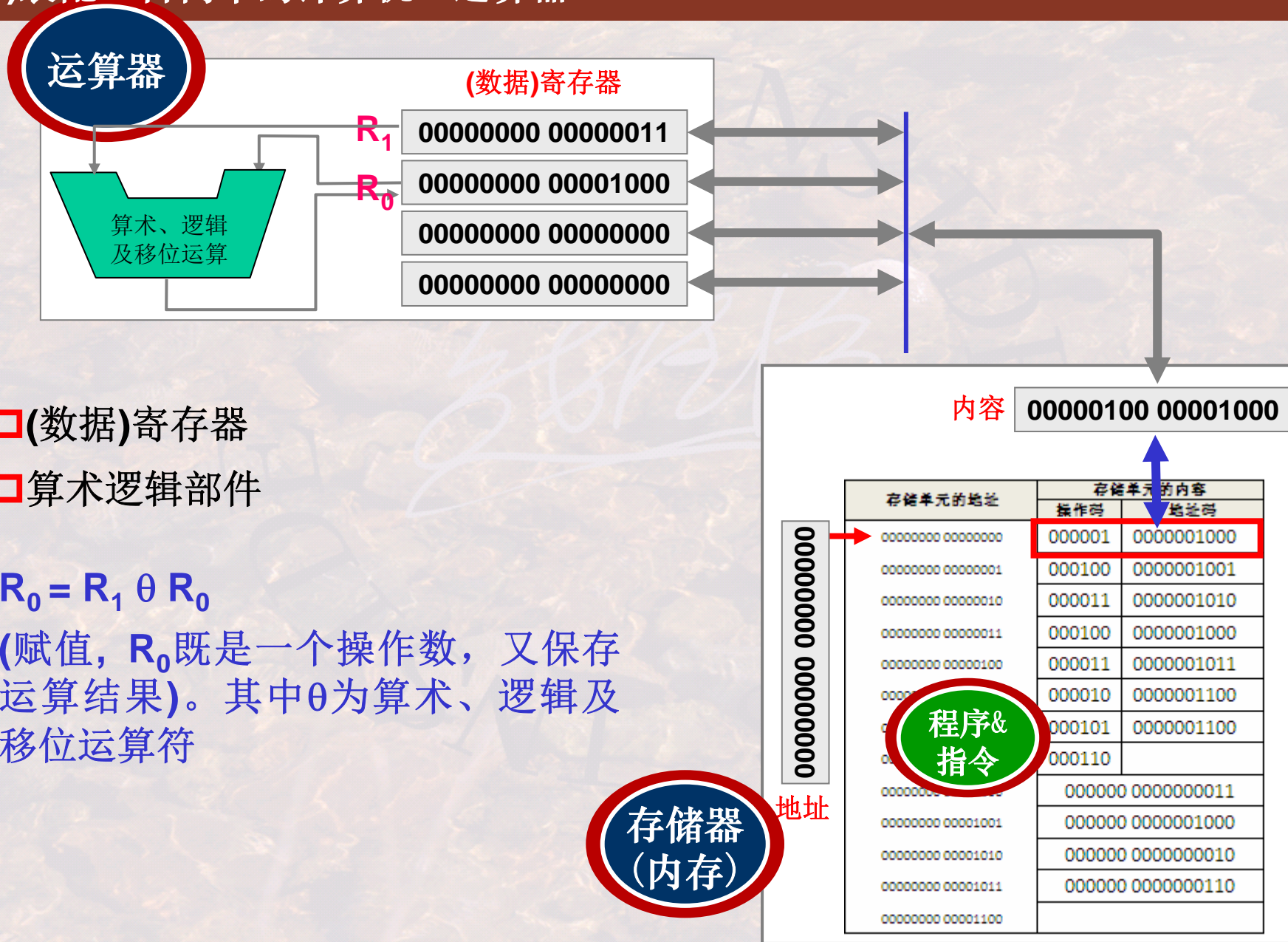
存储单元的地址	存储单元的内容	
	操作码	地址码
00000000 00000000	000001	0000001000
00000000 00000001	000100	0000001001
00000000 00000010	000011	0000001010
00000000 00000011	000100	0000001000
00000000 00000100	000011	0000001011
00000000 00000101	000010	0000001100
00000000 00000110	000101	0000001100
00000000 00000111	000110	
00000000 00001000		000000 0000000011
00000000 00001001		000000 0000001000
00000000 00001010		000000 0000000010
00000000 00001011		000000 00000000110
00000000 00001100		

地址 00000000 00000000

程序&指令

计算化-自动化：程序怎样被执行

(2)装配一台简单的计算机—运算器



□(数据)寄存器

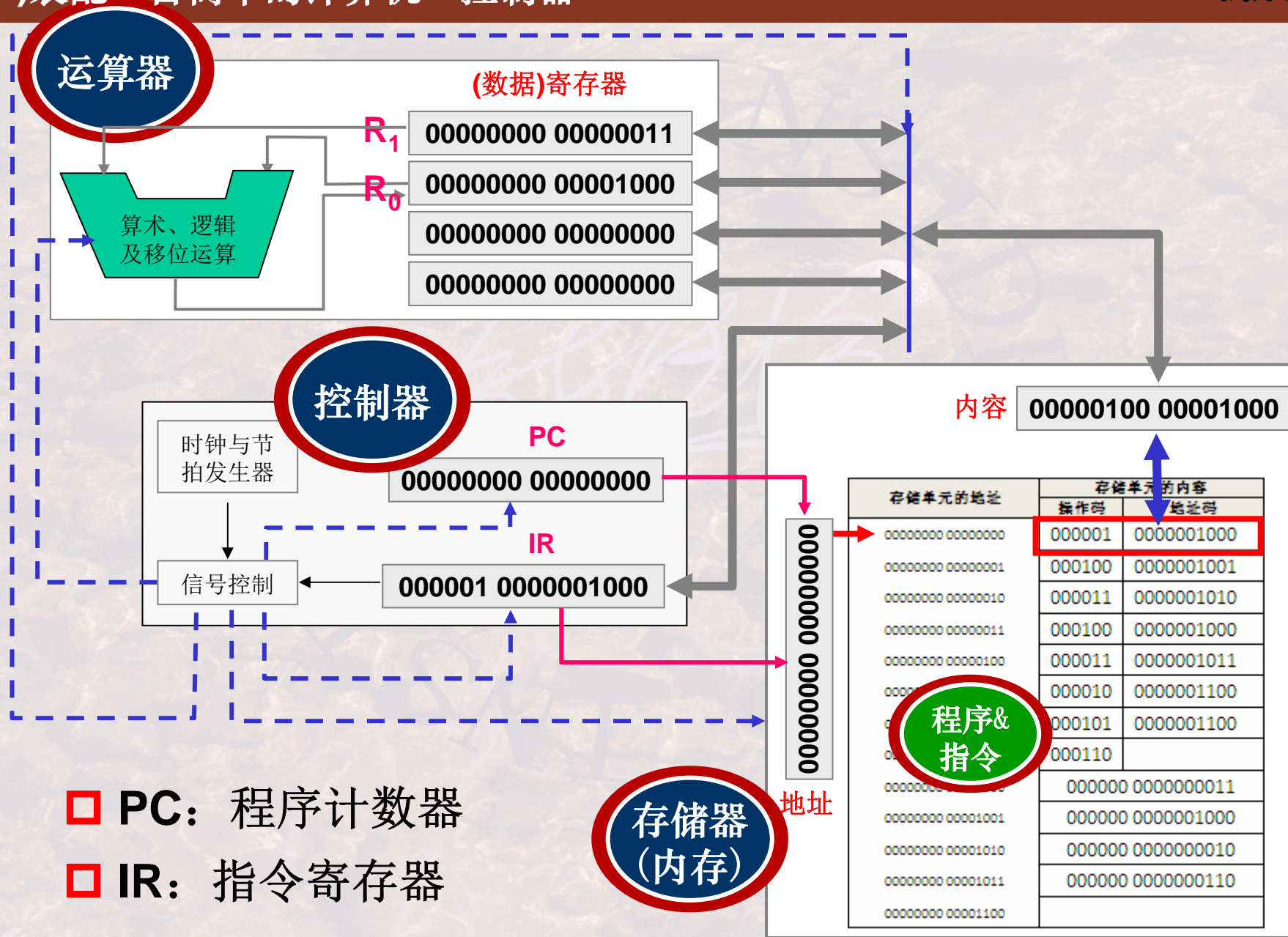
□算术逻辑部件

$$R_0 = R_1 \theta R_0$$

(赋值, R_0 既是一个操作数, 又保存运算结果)。其中 θ 为算术、逻辑及移位运算符

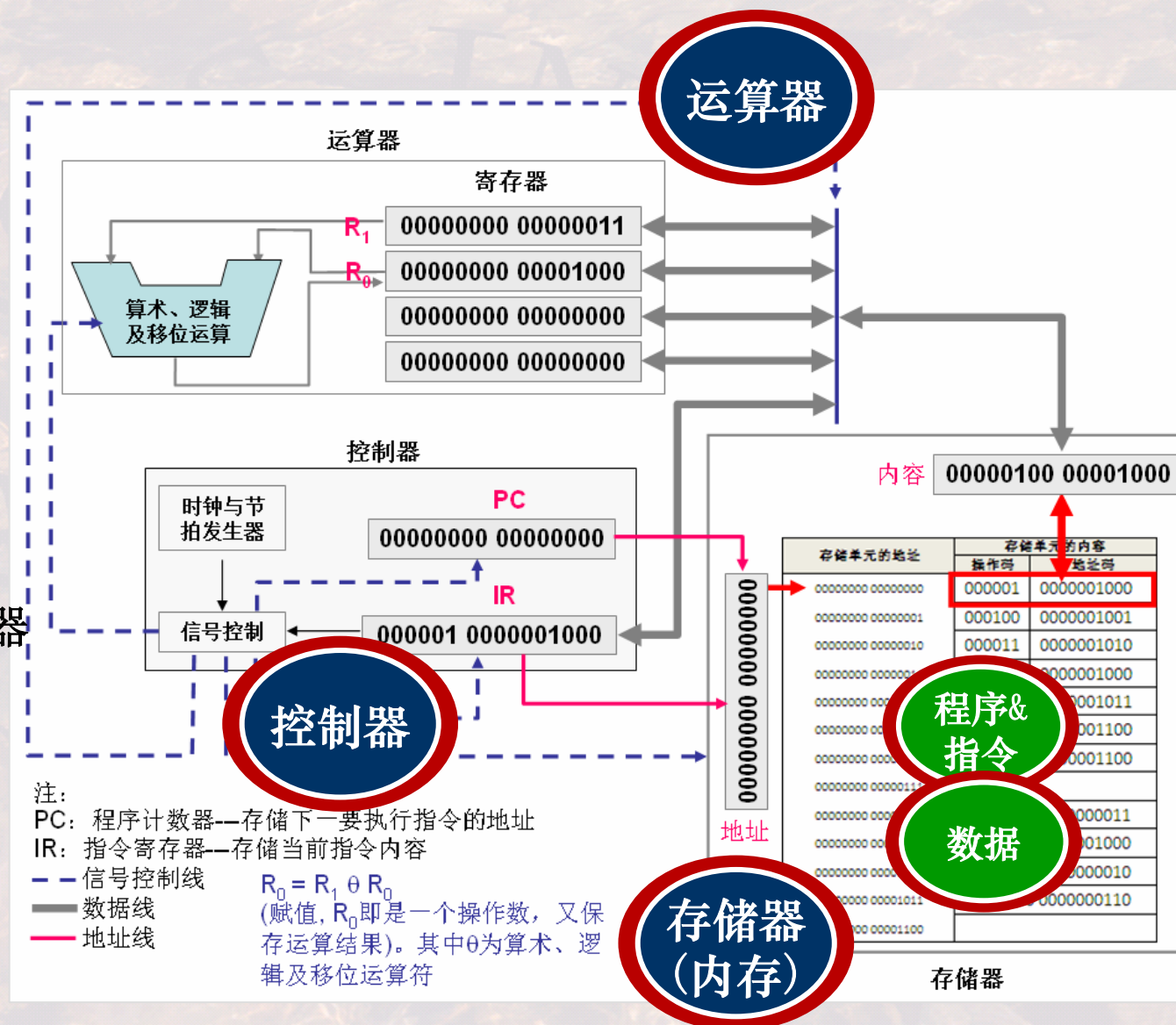
计算化-自动化：程序怎样被执行

(3)装配一台简单的计算机—控制器



计算化-自动化：程序怎样被执行 (4)装配一台简单的计算机—关键要素

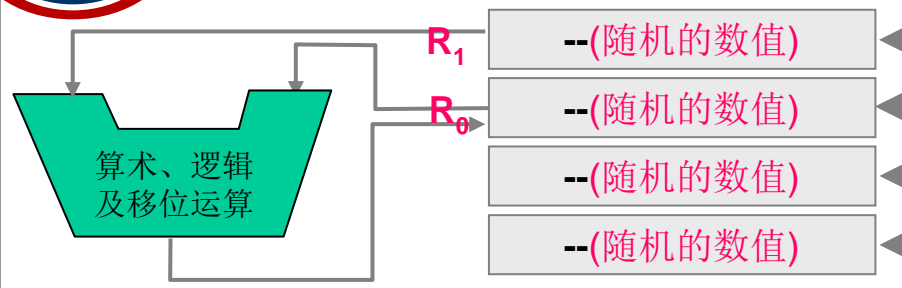
- 寄存器
- 算术逻辑部件
- 程序计数器PC
- 指令寄存器
- 信号控制器
- 时钟与信号发生器
- 存储单元地址
- 存储单元内容



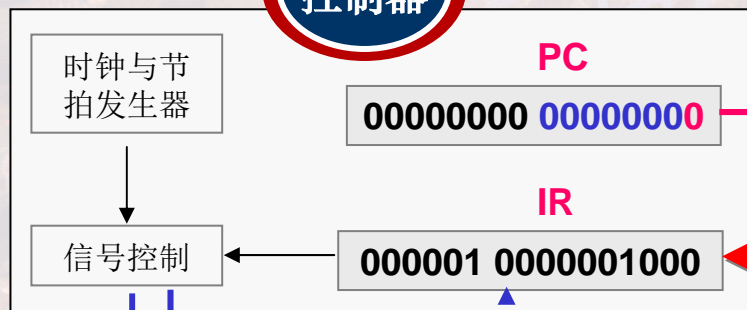
计算化-自动化：程序怎样被执行

(5)第1条指令的读取

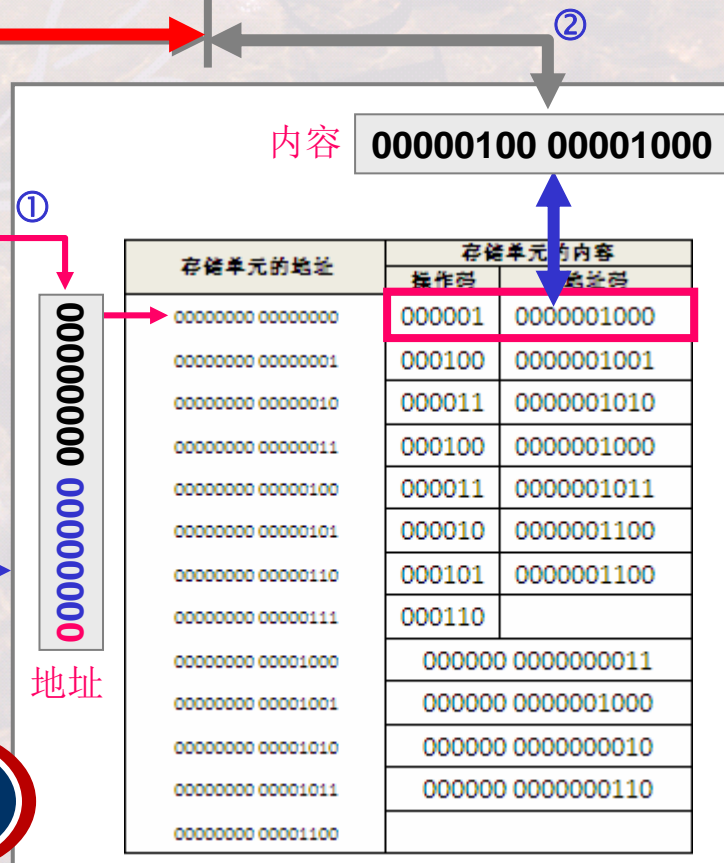
运算器



控制器



存储器

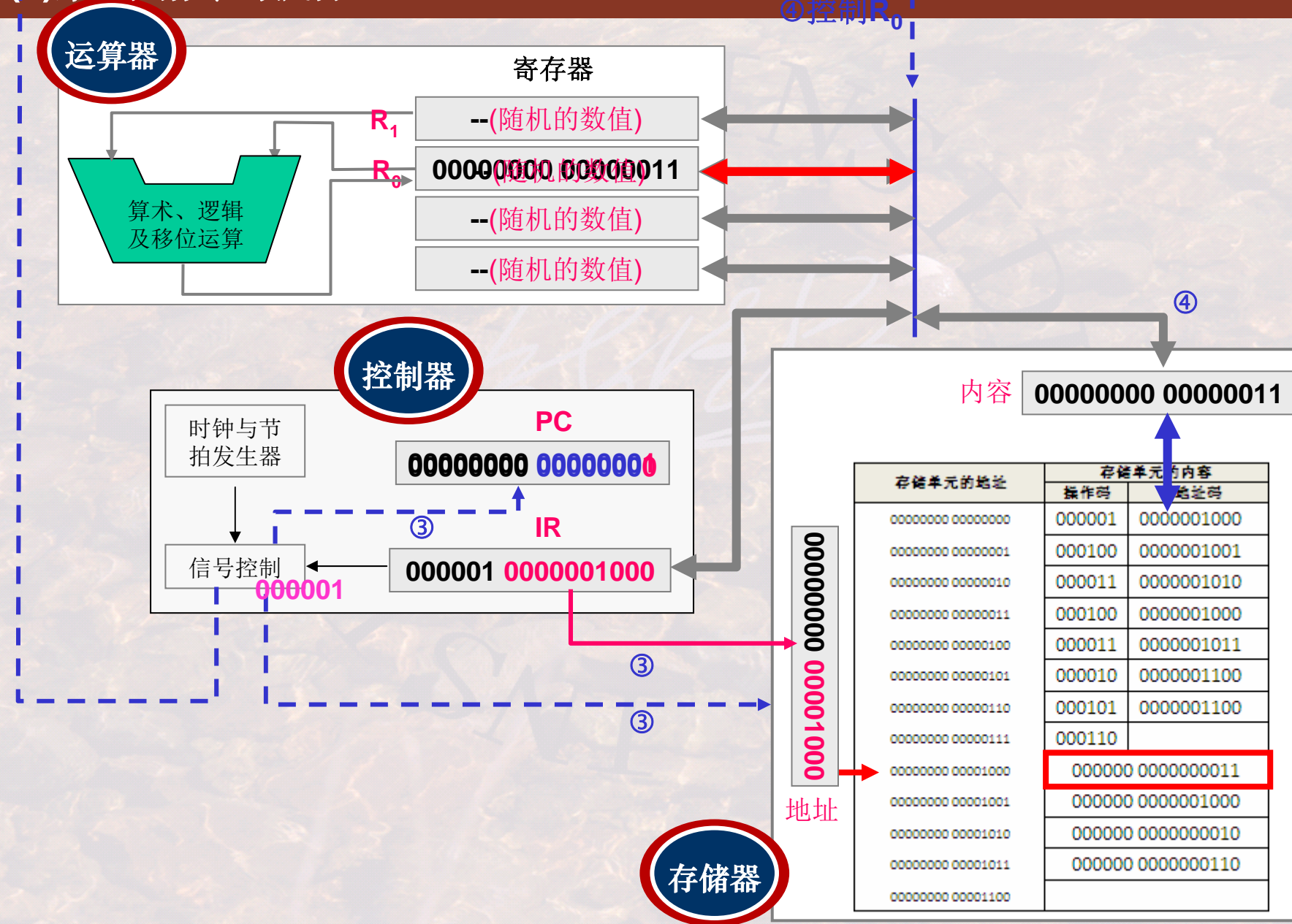


计算化-自动化：程序怎样被执行



战德臣 教授

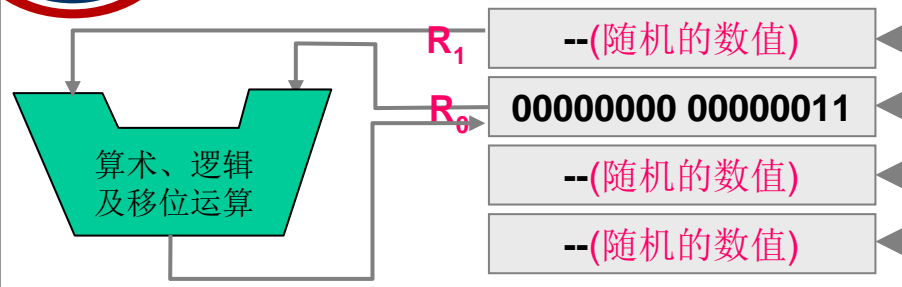
(6)第1条指令的执行



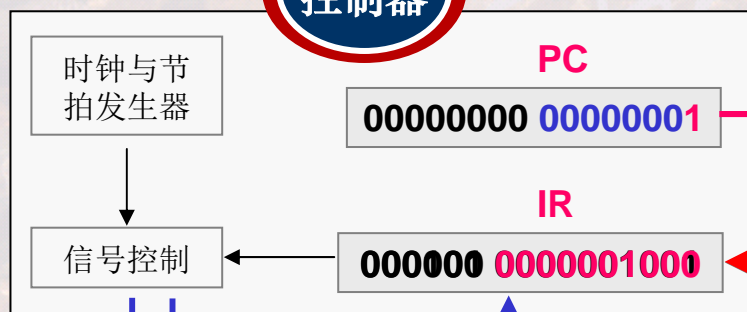
计算化-自动化：程序怎样被执行

(7)第2条指令的读取

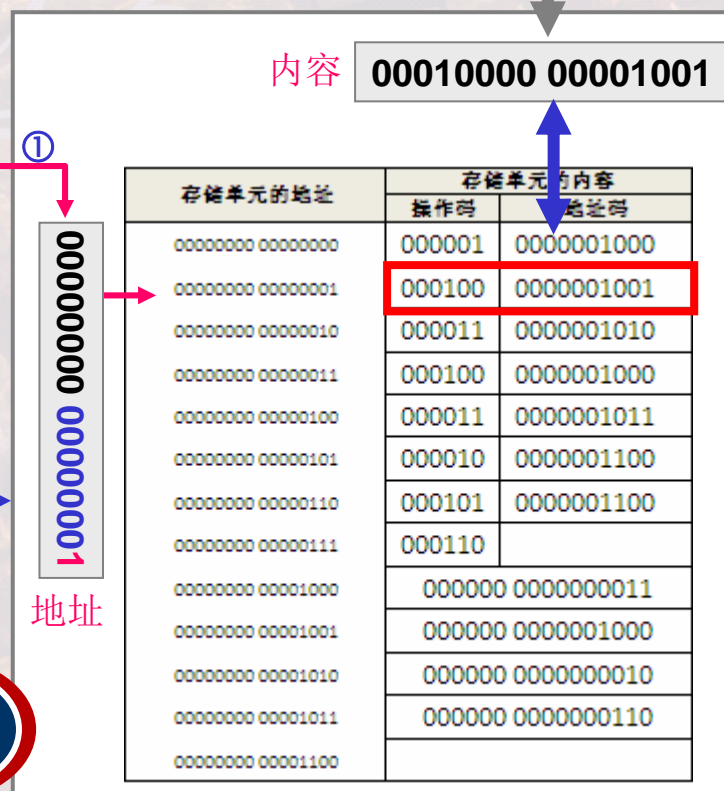
运算器



控制器

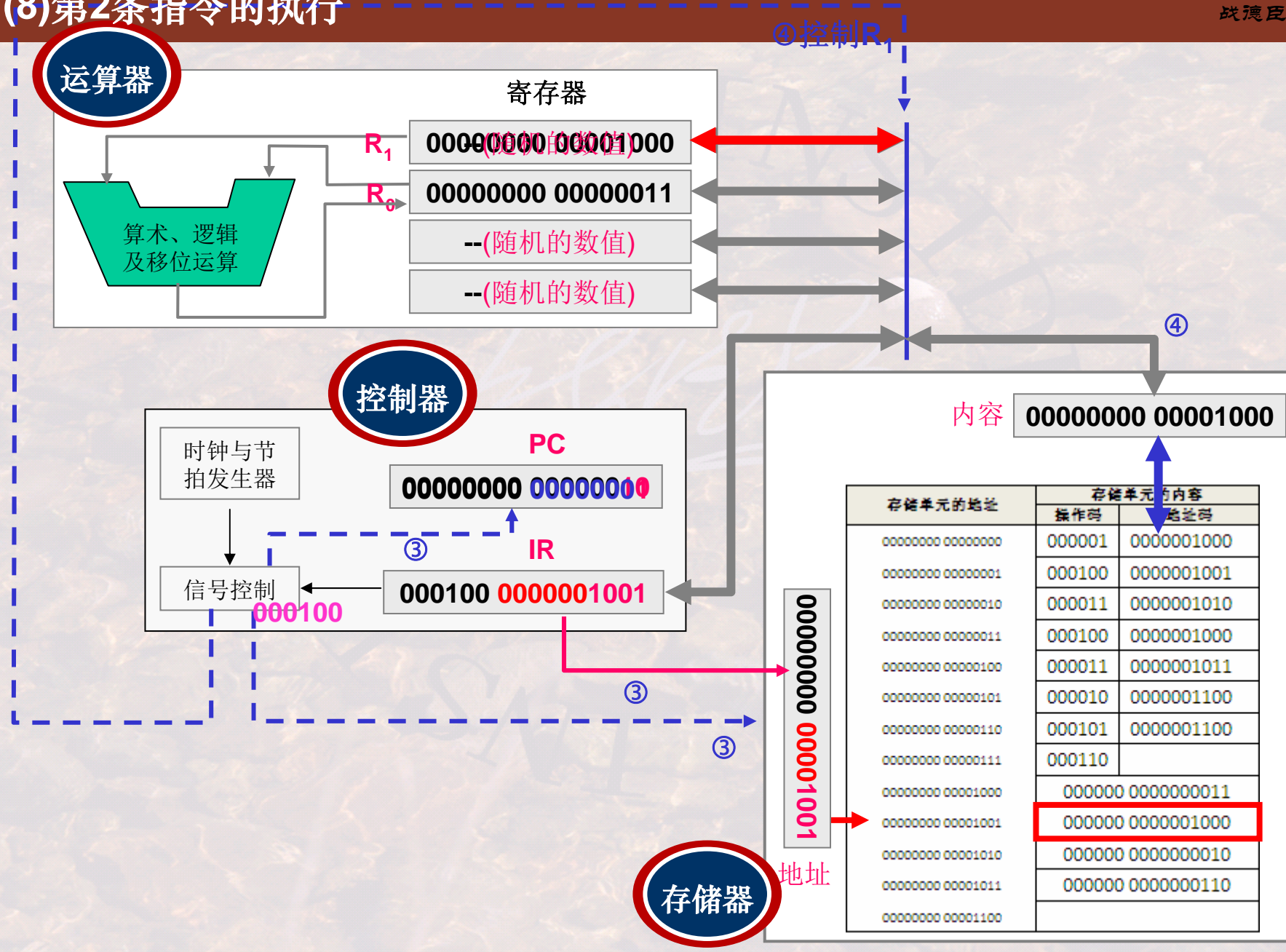


存储器



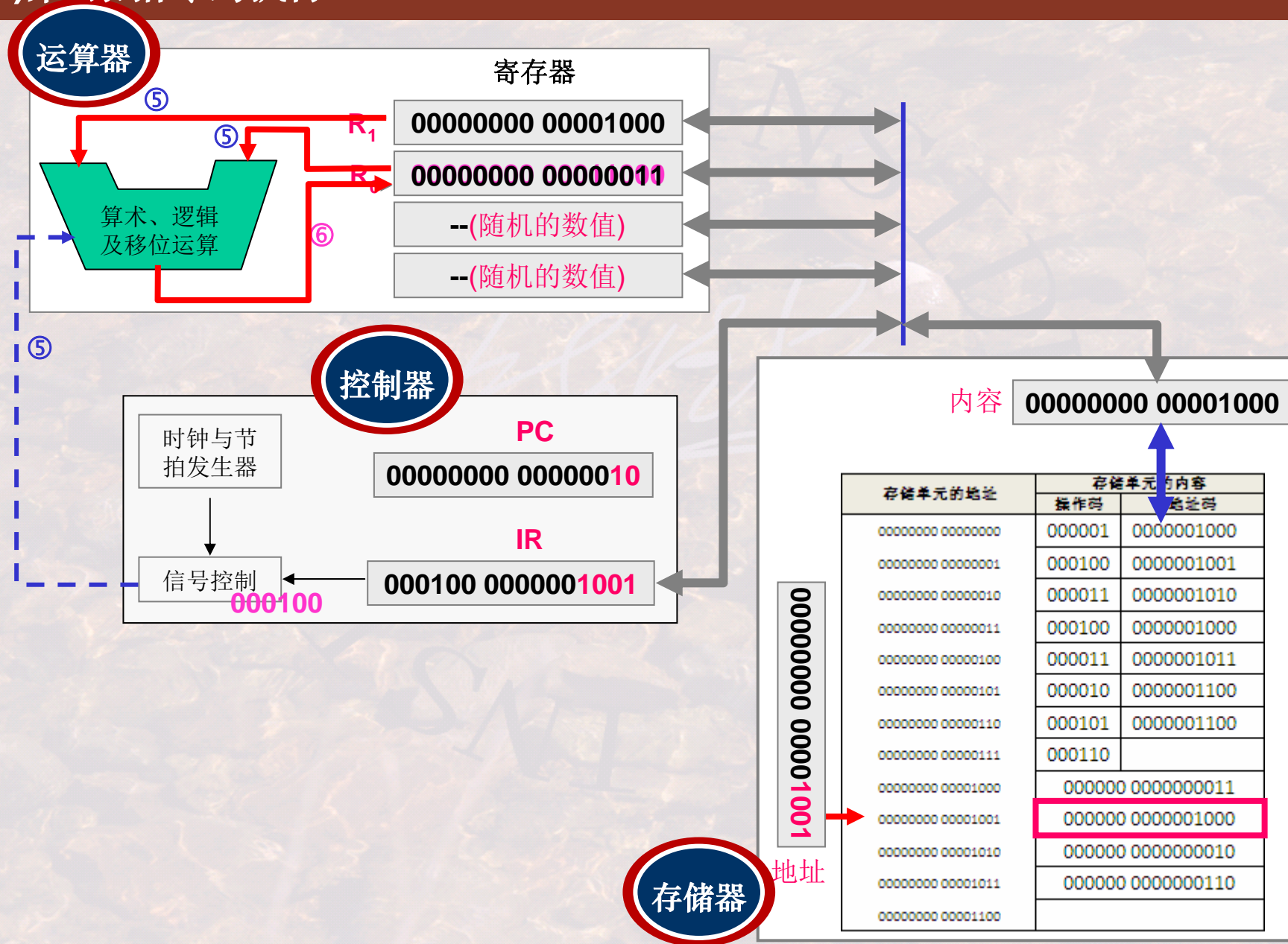
计算化-自动化：程序怎样被执行

(8)第2条指令的执行



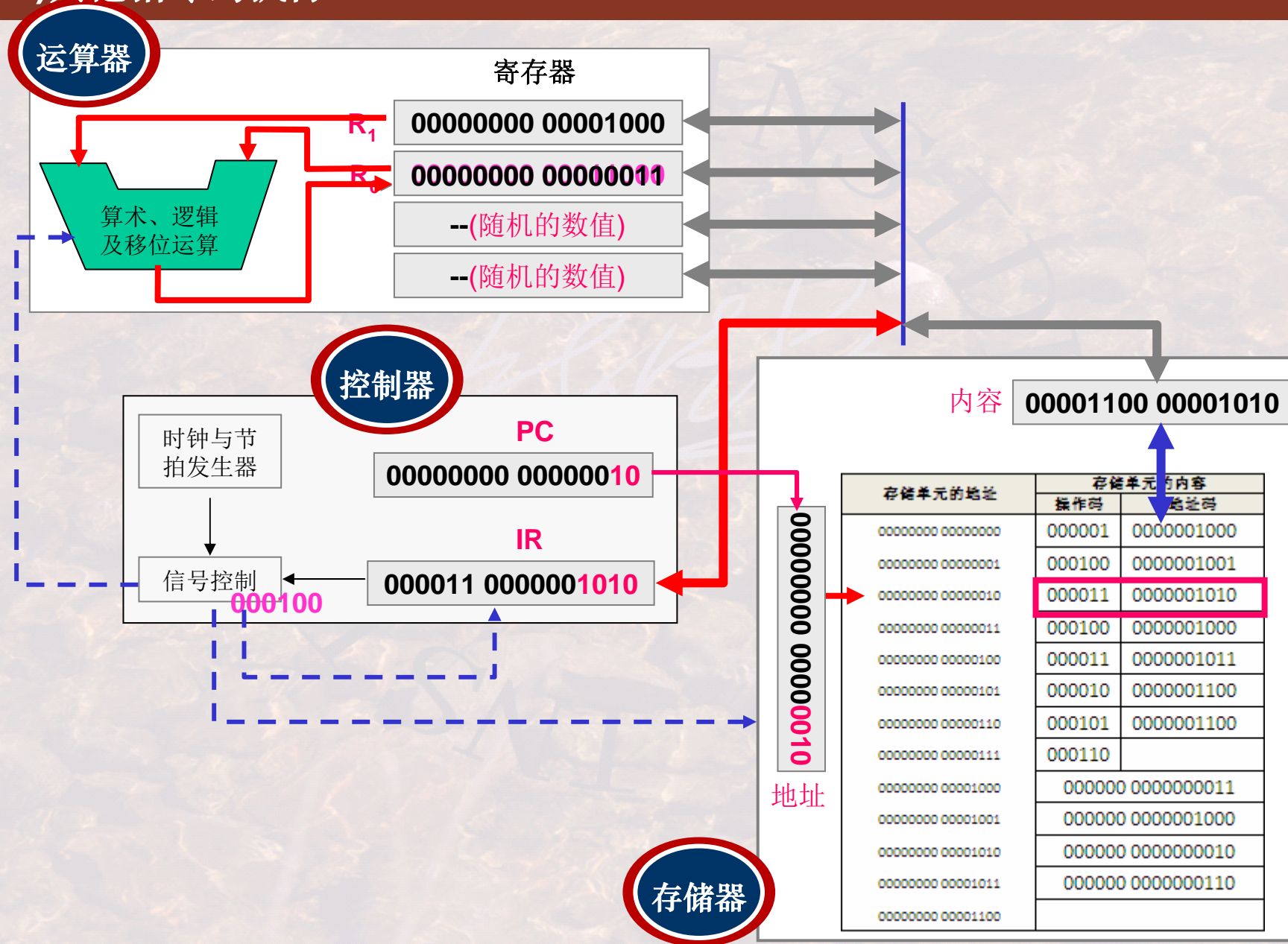
计算化-自动化：程序怎样被执行

(9)第2条指令的执行



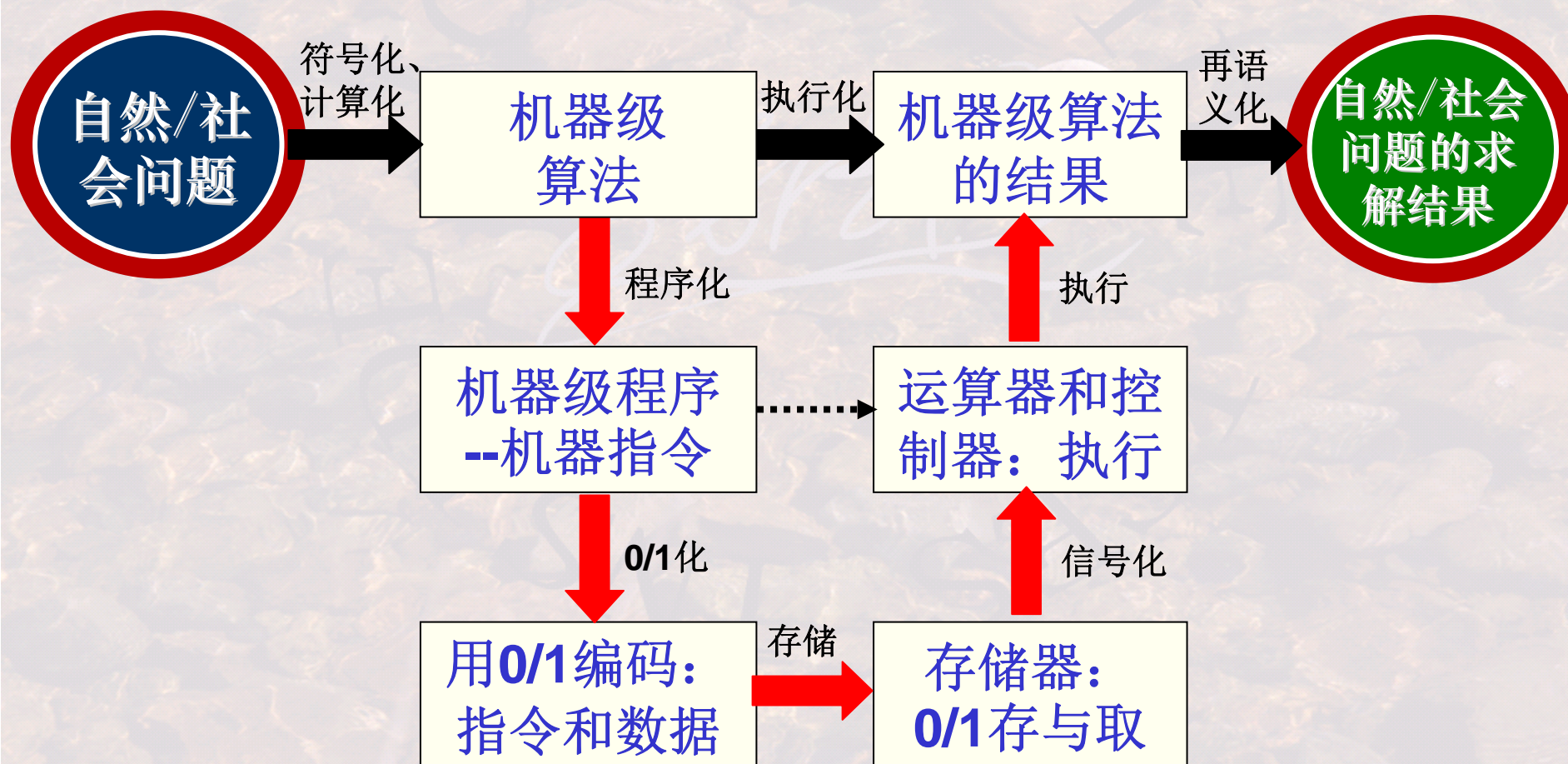
计算化-自动化：程序怎样被执行

(10)其他指令的执行



计算化-自动化：程序怎样被执行

(11)推而广之：用机器语言进行问题求解



计算化-自动化

-----能否更方便地编写程序：高级语言？

战德臣

哈尔滨工业大学 教授.博士生导师
教育部大学计算机课程教学指导委员会委员

Research Center on **I**ntelligent
Computing for **E**nterprises & **S**ervices,
Harbin **I**nstitute of **T**echnology

计算化-自动化：能否更方便的编写程序

(1)由机器语言到汇编语言

计算机语言--汇编语言

◆用符号编写程序 ==> 翻译 ==> 机器语言程序

◆人们提供了用助记符编写程序的规范/标准。同时开发了一个翻译程序，实现了将符号程序自动转换成机器语言程序的功能。

计算7+10并存储的程序

```
10000110 00000111
10001011 00001010
10010111 00000110
11110100
```

这种程序怎么能看懂？看不懂又能怎么办呢？

操作码	地址码
100001	1000000111
↓	
MOV A, 7	



计算7+10并存储的程序

```
MOV A, 7
ADD A, 10
MOV (6), A
HLT
```

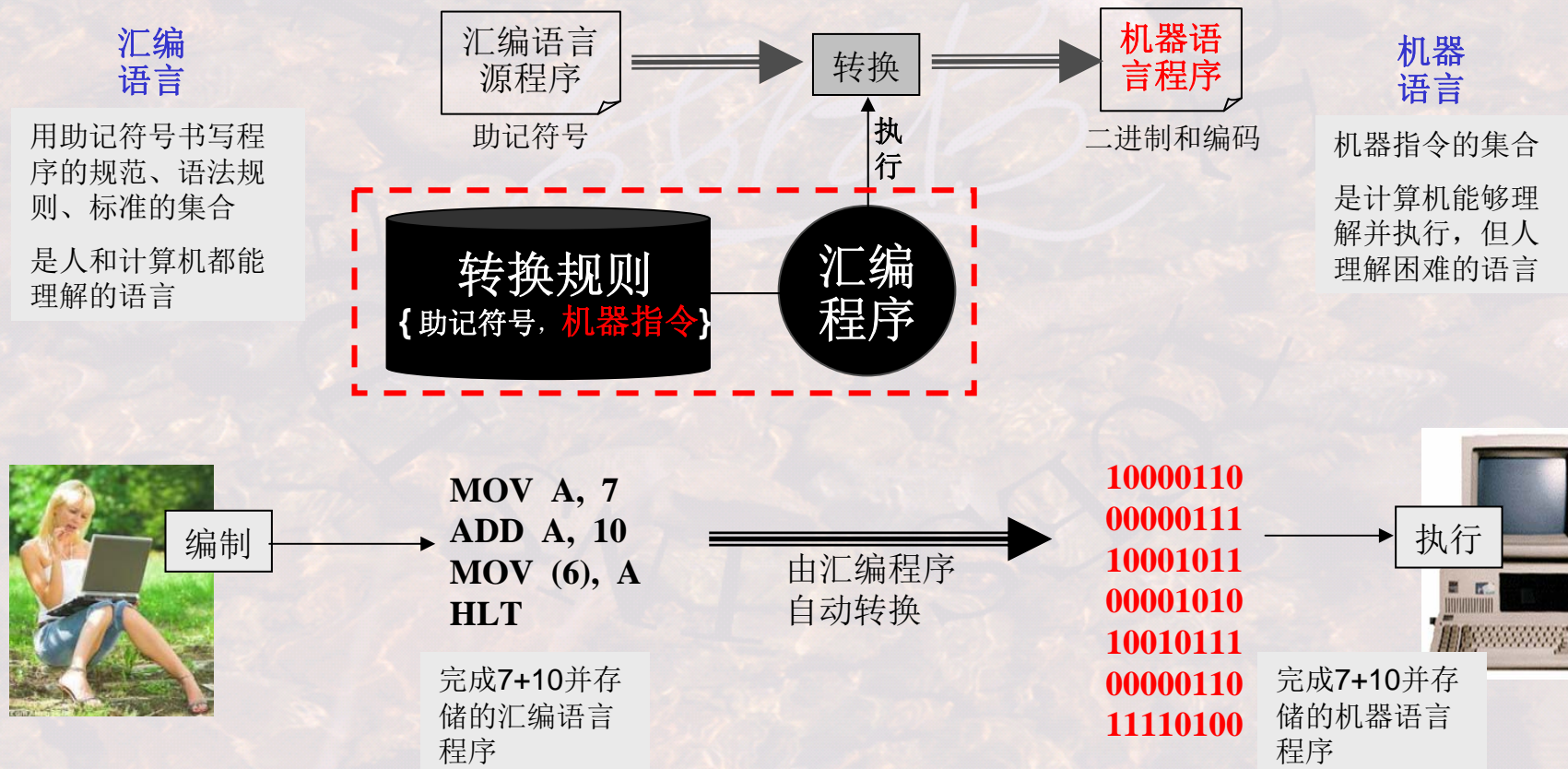
汇编
语言

计算化-自动化：能否更方便的编写程序

(1)由机器语言到汇编语言

计算机语言---汇编语言---汇编程序(编译器)

◆汇编语言程序处理过程



计算机语言---高级语言

◆人们提供了类似于自然语言方式、以语句为单位书写程序的规范/标准。并开发了一个翻译程序，实现了将语句程序自动翻译成机器语言程序的功能。



计算7+10并存储的程序

```
Result = 7+10;  
Return
```

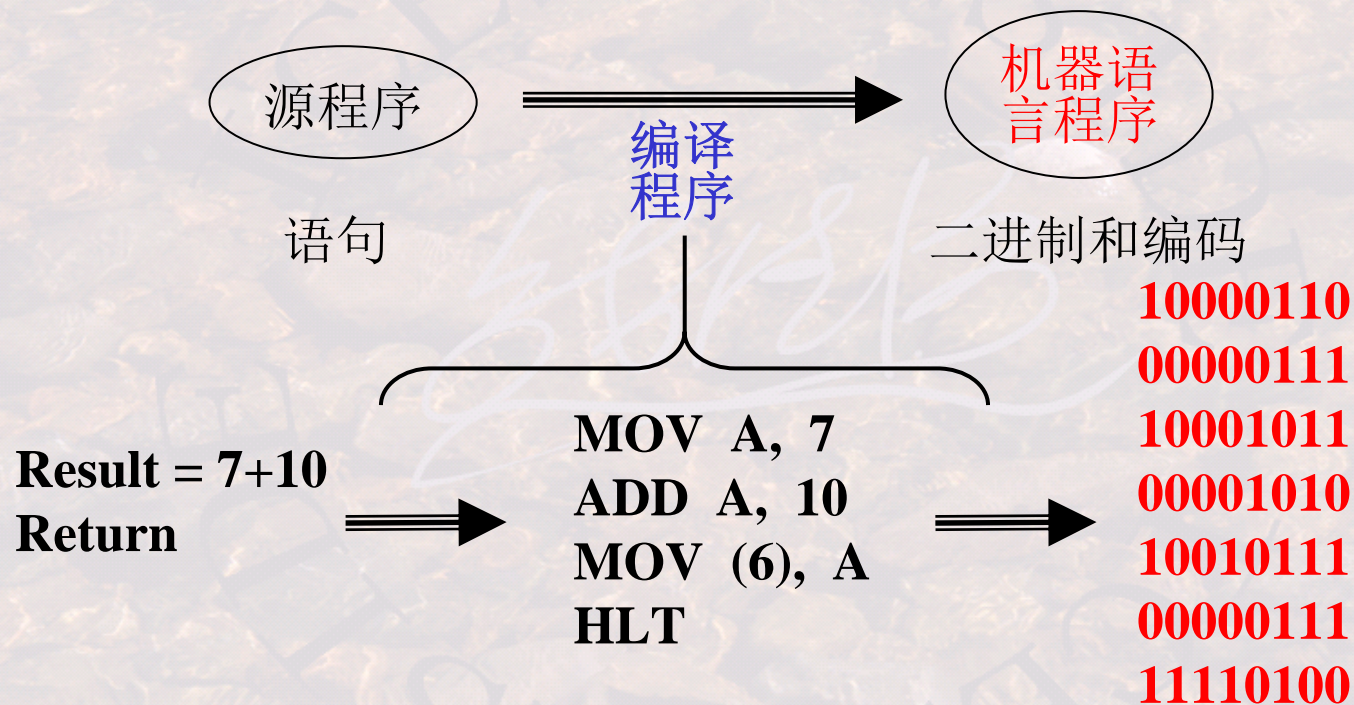
◆高级语言：机器无关性；一条高级语言语句往往可由若干条机器语言语句实现且不具有对应性

◆汇编语言：机器相关性；汇编语言语句和机器语言语句有对应性

计算化-自动化：能否更方便的编写程序

(2)由汇编语言到高级语言

高级语言程序处理过程示意



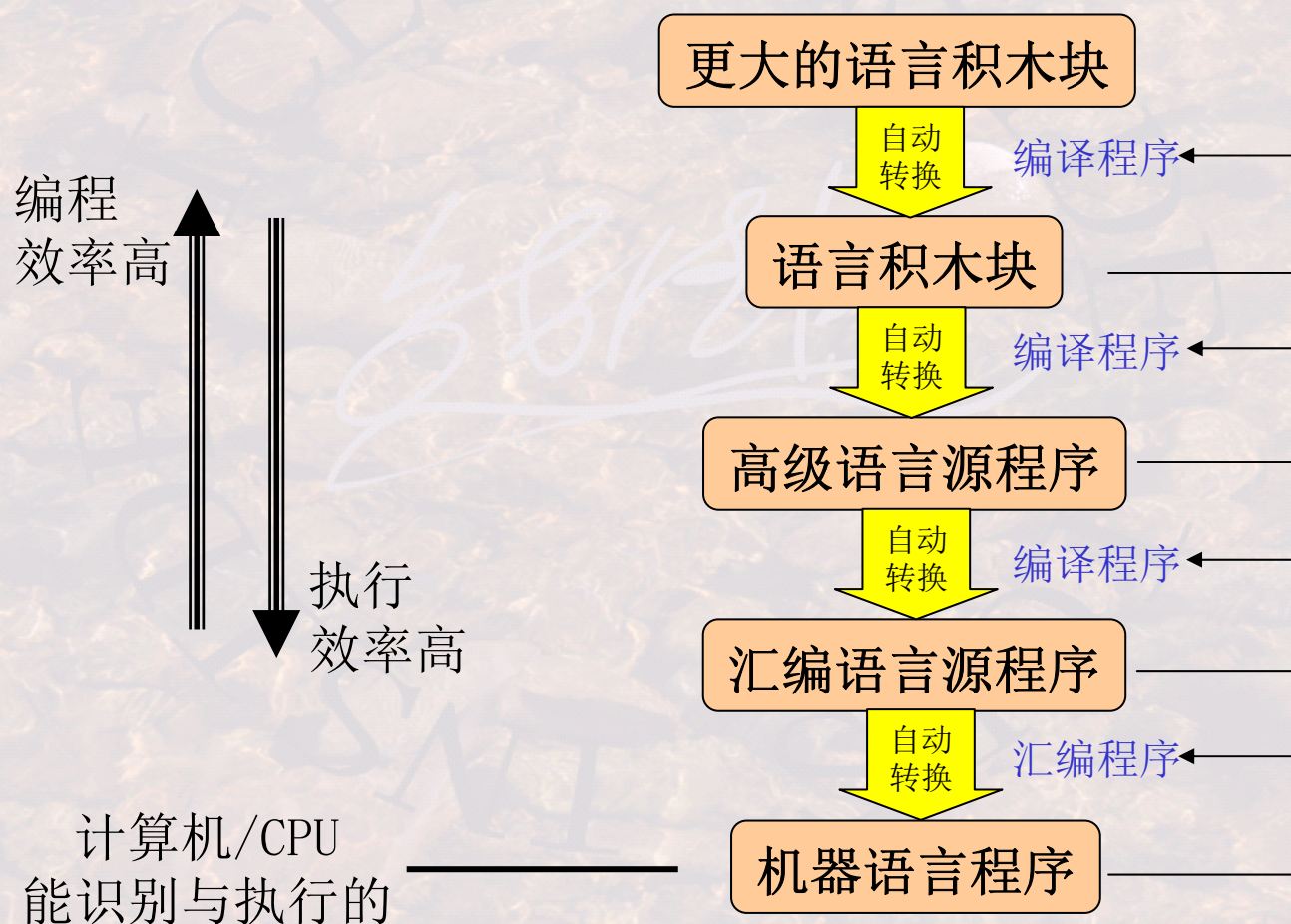
◆所有计算机语言的程序都必须转换成机器语言程序，机器才能够执行。--这种转换，是由“编译”程序来实现的。

计算化-自动化：能否更方便的编写程序

(3)计算机语言的发展



计算机语言发展的基本思维



计算化-自动化：能否更方便的编写程序

(4)用高级语言编写程序？

计算 ax^2+bx+c

其中 a, x, b, c 是变量。

变量的地址是由编译程序在编译过程中自动分配的，也即是说编译器根据当时编译的情况，分配 a, x, b, c 为8号，9号，10号，11号存储单元，并产生上述的机器指令程序

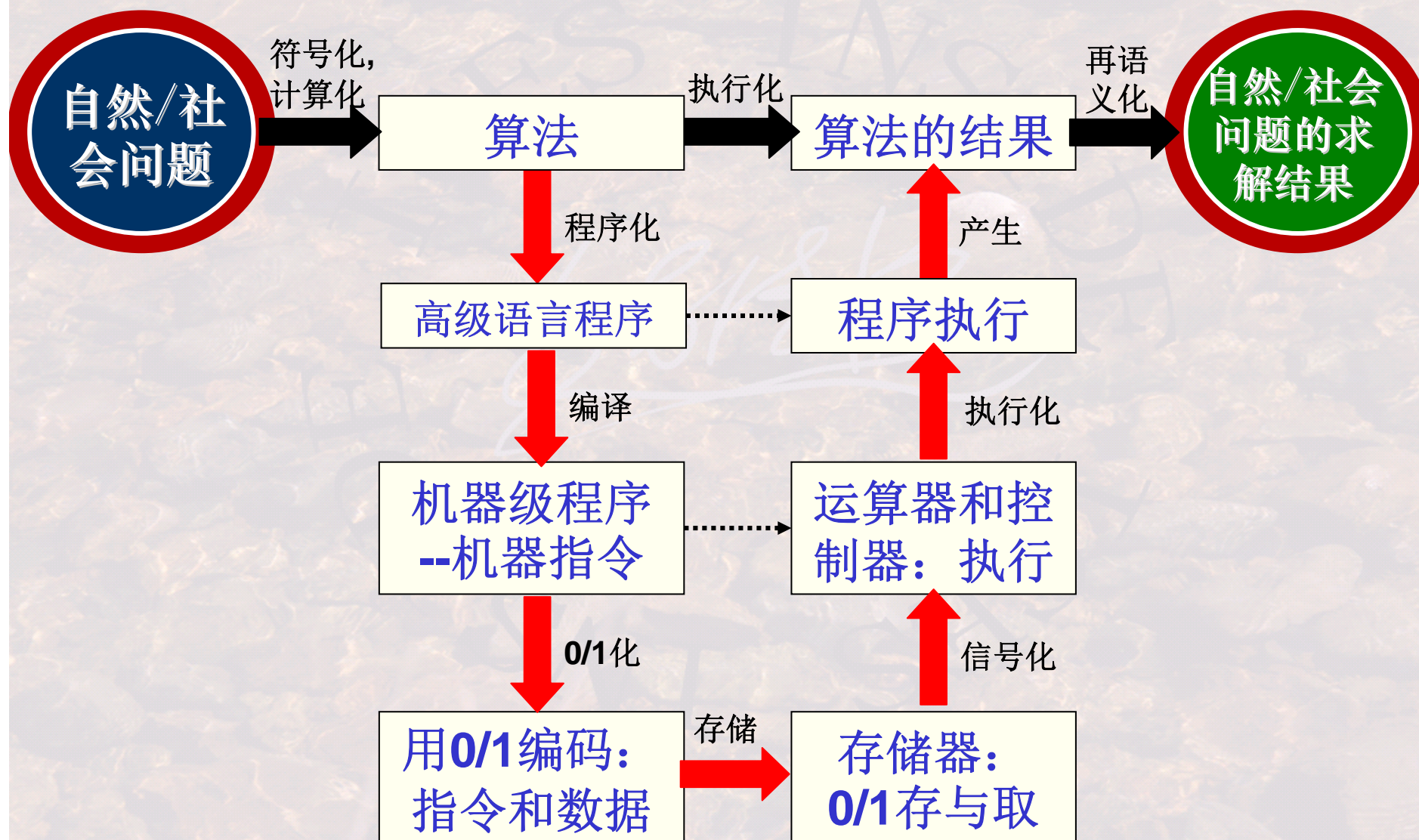
这种程序怎么编写呢？我能够编写程序吗？



```
Main() {  
    int result;    //定义变量 result  
    int x;         //定义变量 x  
    int a;         //定义变量 a  
    int b;         //定义变量 b  
    int c;         //定义变量 c  
    x = 3;         //将 3 赋值给 x  
    //数据赋值过程中也可在运行过程中进行  
    a = 8;         //将 8 赋值给 a  
    b = 2;         //将 2 赋值给 b  
    c = 6;         //将 6 赋值给 c  
    result = (((a * x) * x) + (b * x)) + c;  
    //计算  $ax^2 + bx + c$  并赋值给 result  
    printf("%d", result); //打印 result 的值  
}
```


计算化-自动化：能否更方便的编写程序

(5)推而广之：用高级语言进行问题求解



第2讲 计算思维起步

--符号化、计算化与自动化

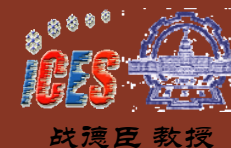
战德臣

哈尔滨工业大学 教授·博士生导师
教育部大学计算机课程教学指导委员会委员

Research Center on Intelligent
Computing for Enterprises & Services,
Harbin Institute of Technology

计算思维起步--符号化、计算化与自动化

•回顾本讲学习了什么？



“符号化-计算化-自动化”是最基本的抽象与自动化机制

“符号化-计算化-自动化”是软件工程学科人才的基本思维模式

“符号化-计算化-自动化”是软件工程学科能力培养的第一步

