

# 计算机科学基础

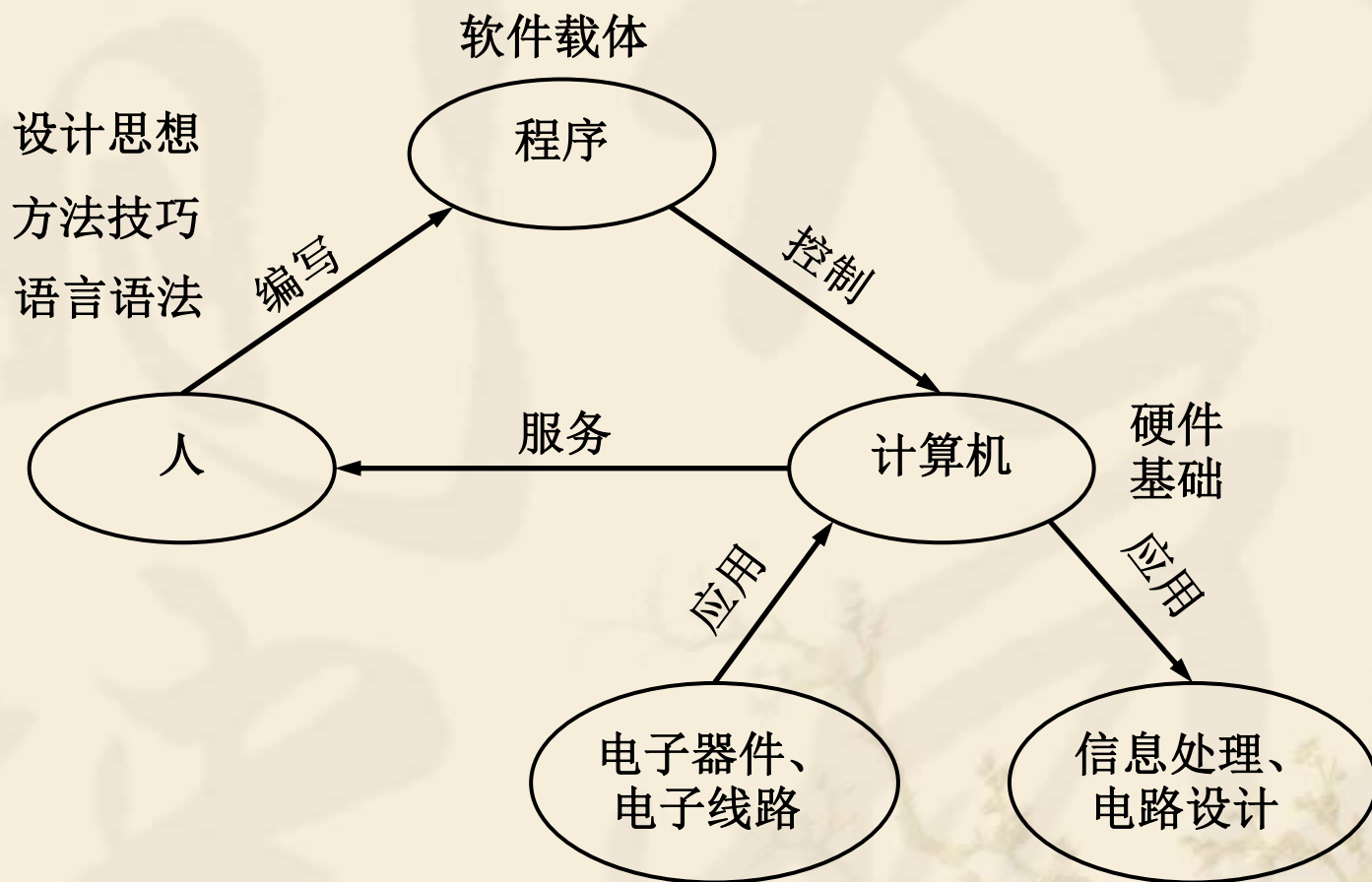
## ——程序设计与算法语言

### 课程预备

戚隆宁

Email: [longn\\_qi@seu.edu.cn](mailto:longn_qi@seu.edu.cn)

Tel: 13813839703



# 课程目标

- ❖ 掌握编写C++程序的基本语法；
- ❖ 掌握面向过程和面向对象的程序设计方法，实现算法解决实际问题；
- ❖ 掌握C++开发工具和程序的基本调试方法。

如何学好这门课程？

- 1) 多实践
- 2) 多提问

# 课程考核与交流

- ❖ 期中机试 (20%)
- ❖ 期末机试 (30%)
- ❖ 期末笔试 (40%)
- ❖ 平时作业 (10%)

C++教学讨论QQ群:

472355600

Email:

longn\_qi@seu.edu.cn

Tel:

13813839703



# 本章主要内容

- ❖ 计算机的发展历史
- ❖ 计算机的数据表示
- ❖ 计算机的基本组成
- ❖ 计算机的基本工作原理

# 本章主要内容

- ❖ 计算机的发展历史
- ❖ 计算机的数据表示
- ❖ 计算机的基本组成
- ❖ 计算机的基本工作原理

# 计算机的发展历史（1/8）

## ——手工计算



# 计算机的发展历史（2/8）

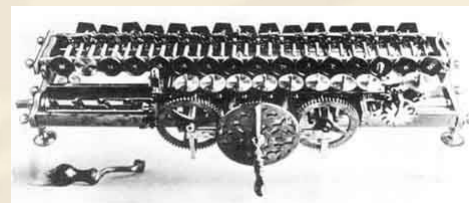
## ——机械计算



1623年  
契克卡德计算钟（德）



1642年  
帕斯卡加减机（法）



1674年  
莱布尼茨乘法机（德）



1843年  
巴贝奇差分机（英）



1874年  
鲍德温手摇计算机（美）

数分钟  
1次乘法

齿轮结构数字计算

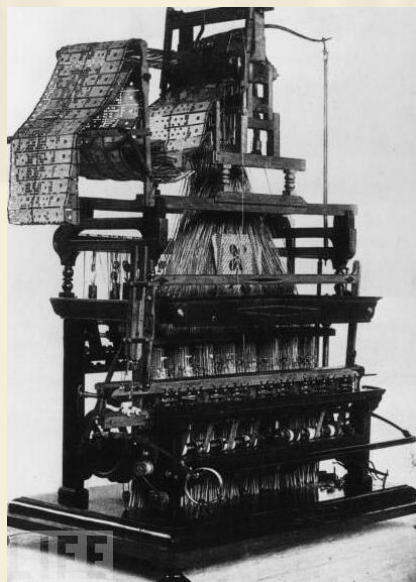


# 计算机的发展历史（3/8）

## ——自动计算（机电计算机）

穿孔纸带输入水银通电  
二进制表示数据和指令  
继电器电路计算

1秒3次加法  
6秒1次乘法



1801年

杰卡德提花织布机（法）



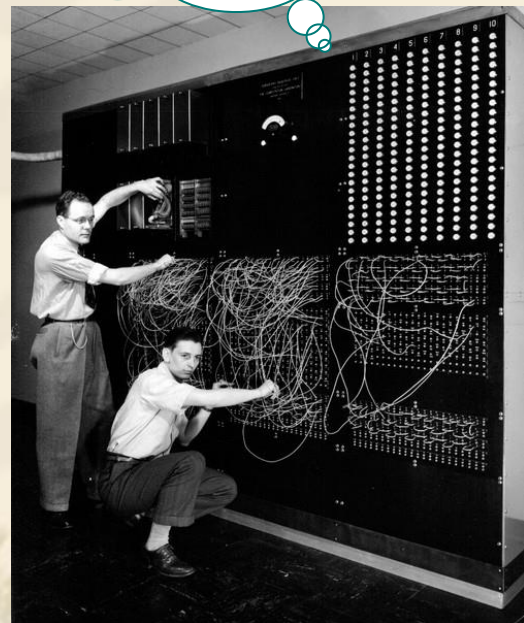
IBM

1890年

霍尔瑞斯制表机（美）

1935年

IBM穿孔卡片机（美）



1943年

Mark-I（美）

自动函数计算，接线编程，  
电传打印输出

# 计算机的发展历史（4/8）

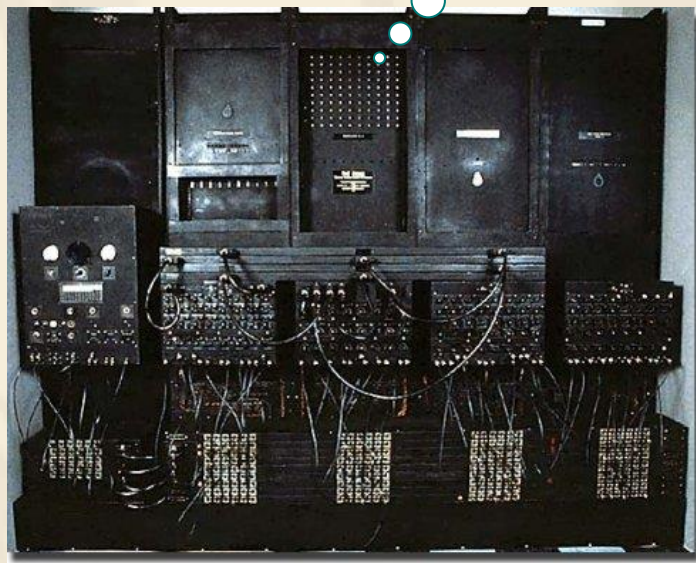
## ——电子管计算机

1秒400次  
乘法

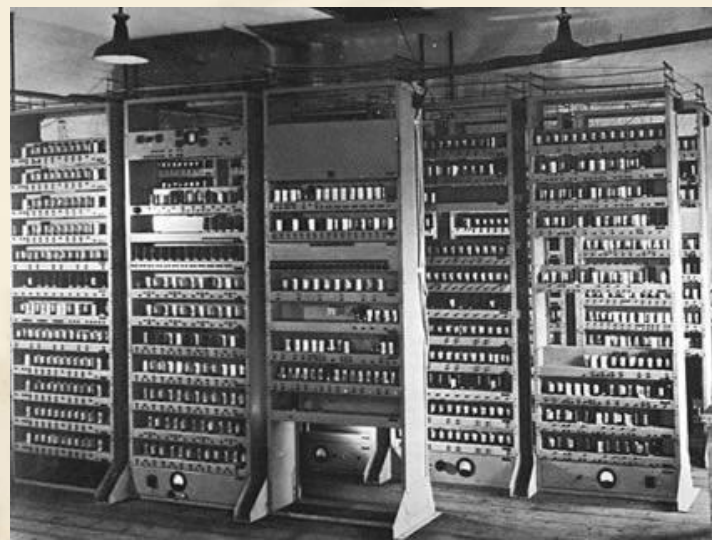
1500个继电器  
18800个电子管  
十进制计算  
接线编程



电子管



1946年  
ENIAC (美)



1949年  
EDSAC (英)

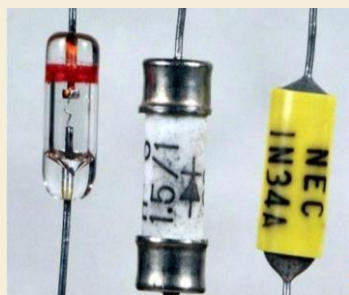
水银延迟线存储，存储程序  
汇编语言诞生



# 计算机的发展历史（5/8）

## ——晶体管计算机

800个晶体管，  
磁芯主存储器，  
磁盘磁带辅存



1947年  
锗晶体二极管



1秒8千次  
乘法

1955年  
TRADIC（美）

1秒2万4千  
次乘法



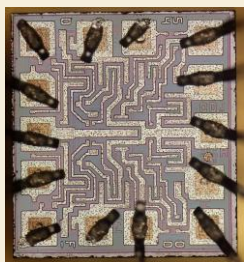
高级程序语言  
Fortran诞生

1958年  
IBM 7090（美）

# 计算机的发展历史（6/8）

## ——集成电路计算机

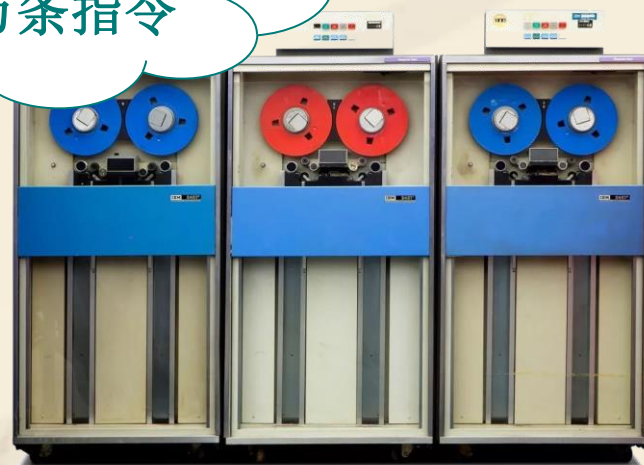
集成门电路  
半导体存储器



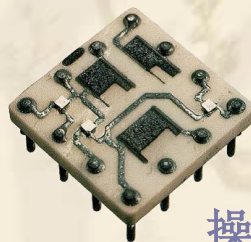
TTL集成逻辑门电  
路



1秒钟  
3.4万条指令



1964年  
IBM 360（美）



标准SLT封装  
采用印制电路板（PCB）

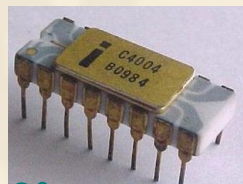
操作系统诞生  
硬件、软件兼容可扩展



# 计算机的发展历史（7/8）

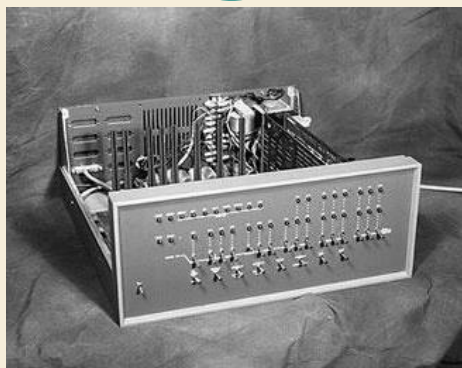
## ——大规模/超大规模集成电路计算机

商用微处理器诞生  
集成**2250**个晶体管



每秒**6万**  
条指令

1971年  
Intel 4004

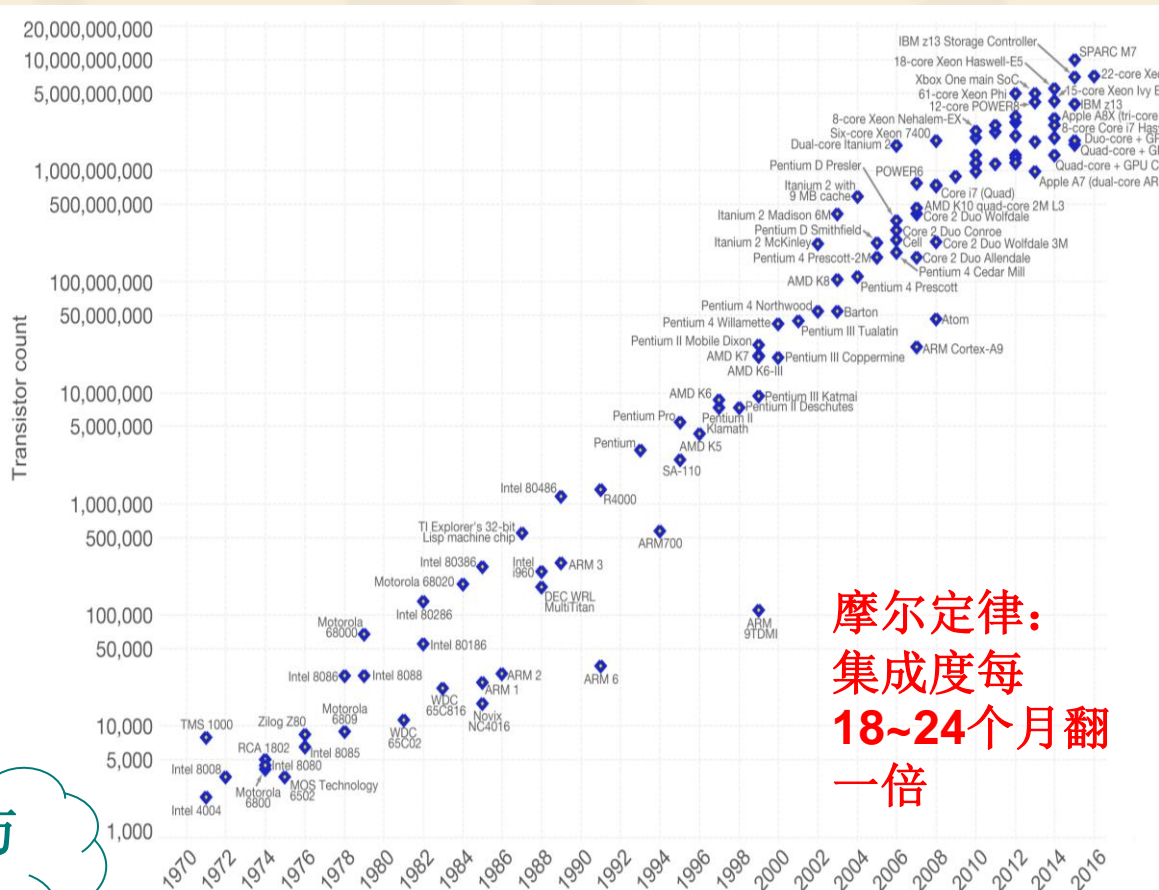


1975年

Altair 8800微机（美）

CPU: Intel 8080

每秒**64万**  
条指令



摩尔定律：  
集成度每  
18~24个月翻  
一倍

# 计算机的发展历史（8/8）

## ——现代计算机

- ❖ 小型微型计算机/个人计算机（十亿~百亿次/秒）
- ❖ 中型大型计算机/服务器（千亿~万亿次/秒）
  - ❧ HP Proliant（Intel Xeon E7-8800系列）、IBM Zaius（Power 9系列）
- ❖ 巨型计算机/超级计算机（千万亿次/秒）
  - ❧ 2020 No.1 日本富岳（Fugaku）41.5亿亿次；No.4 中国神威 9.3亿亿次/秒



# 本章主要内容

- ❖ 计算机的发展历史
- ❖ 计算机的数据表示★
- ❖ 计算机的基本组成
- ❖ 计算机的基本工作原理



# 计算机的数据表示（1/14）

## ❖ 计数的一般方法——位值计数法

∞ 位（A）、位权（W）、基数（B）  $W_n = B^n$

$$123 = 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0 \quad N = A_n W_n + \dots + A_0 W_0$$

1

百位

2

十位

3

个位

$A_n$

高位

...

$A_0$

低位

每个基本的有限计数方法称为**位**；每位规定一个权重值，称为**位权**，通过**基数**的幂来表示。

进制	基数	位计数范围	表示方法举例
十进制（Decimal）	10	0~9	101、101D、 $(101)_{10}$
二进制（Binary）	2	0、1	101B、 $(101)_2$
八进制（Octal）	8	0~7	101O、 $(101)_8$
十六进制（Hexadecimal）	16	0~9、A~F	101H、 $(101)_{16}$

## ❖ 计算机的世界是二进制的世界



# 计算机的数据表示（2/14）

## ❖ 进制间的转换

### ∞ 本质

- 等值转换
- 不同基数下的表达

### ∞ 转换原则

- 整数部分和小数部分分别进行转换

# 计算机的数据表示（3/14）

## ❖ 进制间的转换

↻ 其它进制转换为十进制

➤ 按位权乘累加

例子：

$$\begin{aligned} & (1011.01)_2 \\ &= 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} \\ &= 8 + 0 + 2 + 1 + 0 + 0.25 \\ &= (11.25)_{10} \end{aligned}$$

$$\begin{aligned} & (136.2)_8 \\ &= 1 \times 8^2 + 3 \times 8^1 + 6 \times 8^0 + 2 \times 8^{-1} \\ &= 64 + 24 + 6 + 0.25 \\ &= (94.25)_{10} \end{aligned}$$

$$\begin{aligned} & (1C.8)_{16} \\ &= 1 \times 16^1 + 12 \times 16^0 + 8 \times 16^{-1} \\ &= 16 + 12 + 0.5 \\ &= (28.5)_{10} \end{aligned}$$

# 计算机的数据表示 (4/14)

## ❖ 进制间的转换

### ∞ 十进制转换为其它进制

- 整数部分反复除以基数取余，余数逆序排列
- 小数部分反复乘以基数取整，整数顺序排列
- 注意小数精度

例子1:

$(26.375)_{10}$

$$\begin{array}{r} 2 \overline{) 26} \dots\dots\dots 0 \\ 2 \overline{) 13} \dots\dots\dots 1 \\ 2 \overline{) 6} \dots\dots\dots 0 \\ 2 \overline{) 3} \dots\dots\dots 1 \\ 1 \dots\dots\dots 1 \end{array}$$

$$(26)_{10} = (11010)_2$$

$$(26.375)_{10} = (11010.011)_2$$

$$\begin{array}{r} 2 \times 0.375 = 0.75 \dots\dots\dots 0 \\ 2 \times 0.75 = 1.5 \dots\dots\dots 1 \\ 2 \times 0.5 = 1 \dots\dots\dots 1 \end{array}$$

$$(0.375)_{10} = (0.011)_2$$

例子2:

$(0.4)_{10}$

$$\begin{array}{r} 2 \times 0.4 = 0.8 \dots\dots\dots 0 \\ 2 \times 0.8 = 1.6 \dots\dots\dots 1 \\ 2 \times 0.6 = 1.2 \dots\dots\dots 1 \\ 2 \times 0.2 = 0.4 \dots\dots\dots 0 \\ 2 \times 0.4 = 0.8 \dots\dots\dots 0 \end{array}$$

.....

$$(0.4)_{10} \approx (0.0110)_2 \text{ (保留4位)}$$

# 计算机的数据表示 (5/14)

## ❖ 进制间的转换

### ∞ 二进制转换为八进制和十六进制

- 三归一法和四归一法
- 位数不足，整数部分高位补0，小数部分低位补0

$$\begin{aligned} & A_{3k+2} \times 2^{3k+2} + A_{3k+1} \times 2^{3k+1} + A_{3k} \times 2^{3k} & A_{4k+3} \times 2^{4k+3} + A_{4k+2} \times 2^{4k+2} + A_{4k+1} \times 2^{4k+1} + A_{4k} \times 2^{4k} \\ & = 2^{3k}(A_{3k+2} \times 2^2 + A_{3k+1} \times 2^1 + A_{3k} \times 2^0) & = 2^{4k}(A_{4k+3} \times 2^3 + A_{4k+2} \times 2^2 + A_{4k+1} \times 2^1 + A_{4k} \times 2^0) \end{aligned}$$

例子：

$(10011.11)_2$

$(\underline{010} \ \underline{011}. \ \underline{110})_2$

$(\underline{0001} \ \underline{0011}. \ \underline{1100})_2$

$(010)_2 = (2)_8$

$(0001)_2 = (1)_{16}$

$(011)_2 = (3)_8$

$(0011)_2 = (3)_{16}$

$(110)_2 = (6)_8$

$(1100)_2 = (C)_{16}$

$(10011.11)_2 = (23.6)_8$

$(10011.11)_2 = (13.C)_{16}$



# 计算机的数据表示 (6/14)

## ❖ 进制间的转换

∞ 八进制和十六进制转为二进制

➤ 一拆三法和一拆四法

∞ 八进制和十六进制相互转换

➤ 通过二进制中转

**例子1：**

**$(34.2)_8$**

**$(3)_8 = (011)_2$**

**$(4)_8 = (100)_2$**

**$(2)_8 = (010)_2$**

**$(34.2)_8 = (011\ 100.010)_2$**

**例子2：**

**$(1B.4)_{16}$**

**$(1)_{16} = (0001)_2$**

**$(B)_{16} = (1011)_2$**

**$(4)_{16} = (0100)_2$**

**$(1B.4)_{16} = (0001\ 1011.0100)_2$**

# 计算机的数据表示（7/14）

## ❖ 计算机中数的表示（机器数）

- ∞ 有限的表示范围：位数、溢出
- ∞ 有限的表示精度
- ∞ 有符号数的表示：原码、反码、补码、移码
- ∞ 小数的表示：定点、浮点

# 计算机的数据表示 (8/14)

## ❖ 计算机中有符号数的表示 (原码)

∞ 最高位表示数正负：“0”为正，“1”为负；其余位表示数的绝对值大小。

❖ 正数不变，负数表示为  $2^{n-1} + |X|$ ， $(-2^{n-1} < X < 0)$

❖ 0 的表示不惟一；加减运算不便需要转换

$$[+57]_{\text{原}} = (00111001)_2 = (57)_{10}$$

$$[-57]_{\text{原}} = (10111001)_2 = (185)_{10}$$

$$[-1]_{\text{原}} = (10000001)_2 = (129)_{10}$$

$$[-127]_{\text{原}} = (11111111)_2 = (255)_{10}$$

$$[+127]_{\text{原}} = (01111111)_2 = (127)_{10}$$

$$[+0]_{\text{原}} = (00000000)_2, \quad [-0]_{\text{原}} = (10000000)_2 = (128)_{10} \text{ 不惟一,}$$

有“正零”和“负零”之分

# 计算机的数据表示（9/14）

## ❖ 计算机中有符号数的表示（反码）

⌘ 最高位表示数的正负：“0”为正，“1”为负；其余位，正数取原值，负数对 $|X|$ 各位取反

❖ 负数表示为  $2^n - 1 + X$ ,  $(-2^{n-1} < X < 0)$

⌘ 0的表示不惟一；加减运算简单

$$[+57]_{\text{反}} = (00111001)_2 = (57)_{10}$$

$$[-57]_{\text{反}} = (11000110)_2 = (198)_{10}$$

$$[-1]_{\text{反}} = (11111110)_2 = (254)_{10}$$

$$[-127]_{\text{反}} = (10000000)_2 = (128)_{10}$$

$$[+127]_{\text{反}} = (01111111)_2 = (127)_{10}$$

$$[+0]_{\text{反}} = (00000000)_2 \quad [-0]_{\text{反}} = (11111111)_2 = (255)_{10} \quad \text{不惟一}$$



# 计算机的数据表示（10/14）

## ❖ 计算机中有符号数的表示（补码）

∞ 最高位表示数的正负：“0”为正，“1”为负；其余位，正数取原值，负数取补（ $2^{n-1} - |X|$ ）

❖ 负数表示为  $2^n + X$ ，（ $-2^{n-1} \leq X < 0$ ）

∞ 0的表示惟一；加减运算简单；

$$[+57]_{\text{补}} = (00111001)_2 = (57)_{10}$$

$$[-57]_{\text{补}} = (11000111)_2 = (199)_{10}$$

$$[-1]_{\text{补}} = (11111111)_2 = (255)_{10}$$

$$[-127]_{\text{补}} = (10000001)_2 = (129)_{10}$$

$$[-128]_{\text{补}} = (10000000)_2 = (128)_{10}$$

$$[+127]_{\text{补}} = (01111111)_2 = (127)_{10}$$

$$[0]_{\text{补}} = [+0]_{\text{补}} = [-0]_{\text{补}} = (00000000)_2 \quad \text{惟一}$$

# 计算机的数据表示（11/14）

## ❖ 计算机中有符号数的表示（移码）

- ∞ 无符号位，真值直接加上固定偏移（如： $2^{n-1}+X$ ）。
- ∞ 0的表示惟一，但不是全0；比较运算简单，加减运算稍复杂；

$$[+57]_{\text{移}} = (10000000)_2 + (00111001)_2 = (10111001)_2 = (185)_{10}$$

$$[-57]_{\text{移}} = (10000000)_2 - (00111001)_2 = (01000111)_2 = (71)_{10}$$

$$[-1]_{\text{移}} = (10000000)_2 - (00000001)_2 = (01111111)_2 = (127)_{10}$$

$$[-127]_{\text{移}} = (10000000)_2 - (01111111)_2 = (00000001)_2 = (1)_{10}$$

$$[-128]_{\text{移}} = (10000000)_2 - (10000000)_2 = (00000000)_2 = (0)_{10}$$

$$[+127]_{\text{移}} = (10000000)_2 + (01111111)_2 = (11111111)_2 = (255)_{10}$$

$$[0]_{\text{移}} = (10000000)_2 = (128)_{10} \quad \text{惟一}$$

$(00000000)_2$

0

1  $(00000001)_2$

$(11111111)_2$  -1

补码 (n=8)

-64  $(11000000)_2$

64  $(01000000)_2$

$(10000001)_2$  -127

-128

127  $(01111111)_2$

$(10000000)_2$

移码(偏移 $2^7 = 128$ )

$(00000000)_2$   
-128

$(01111111)_2$  -1

$(10000001)_2$  1

$(11111111)_2$  127

-127

-64

0

64

$(00000001)_2$

$(01000000)_2$

$(10000000)_2$

$(11000000)_2$

# 计算机的数据表示（12/14）

## ❖ 计算机中小数的表示（定点数）

- ❧ 约定小数点固定在某个位置，即表示为定点数。  
只能处理定点数的计算机叫做“定点机”。
- ❧ 多用于小数范围和精度有限的场合（数字信号处理DSP）

例如：

把小数点位置固定在数的最高位之前，使机器所表示的数是纯小数。

这种表示中， $|X|_{\min}=2^{-n}$ ， $|X|_{\max}=1-2^{-n}$ （设尾数有n位）

$$(.1001)_2 = 2^{-1} + 2^{-4} = 0.5625$$

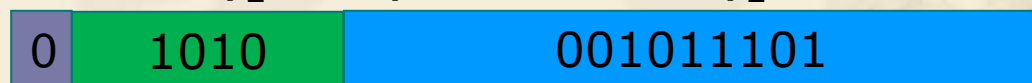


# 计算机的数据表示（13/14）

## ❖ 计算机中小数的表示（浮点数）

- ❧ 小数点位置可浮动，且在数据格式中显式给出。
- ❧ 浮点数均由数符、阶和尾数三个部分构成。
- ❧ 一般规定（IEEE 754/854标准，规约格式）
  - 数符表示浮点数的正负，0为正，1为负；
  - 尾数是原码表示的二进制定点小数，约定小数点在尾数最高位的左边，整数部分固定是1（省略）；
  - 阶为移码（偏移 $2^{n-1}-1$ ）表示的二进制整数，其隐含基数一般为2。
  - 0的尾数是0，阶也是0，符号位正负无关。

$$(1001.0111010)_2 = +(1.001011101)_2 \times 2^{(1010)_{2[\text{移}]}}$$



$$(-0.0001010011)_2 = -(1.010011000)_2 \times 2^{(0011)_{2[\text{移}]}}$$



	单精度	双精度
数符	1	1
阶	8	11
尾数	23	52

# 计算机的

## ❖ 计算机中

在计算机  
编码实现  
手段。

- 西文符
- 汉字符  
码等)
- 机内码
- 国际符

编码	字符	编码	字符	编码	字符	编码	字符
0	NUL	32	Space	64	@	96	`
1	SOH	33	!	65	A	97	a
2	STX	34	"	66	B	98	b
3	ETX	35	#	67	C	99	c
4	EOT	36	\$	68	D	100	d
5	ENQ	37	%	69	E	101	e
6	ACK	38	&	70	F	102	f
7	BEL	39	'	71	G	103	g
8	BS	40	(	72	H	104	h
9	TAB	41	)	73	I	105	i
10	LF	42	*	74	J	106	j
11	VT	43	+	75	K	107	k
12	FF	44	,	76	L	108	l
13	CR	45	-	77	M	109	m
14	SO	46	.	78	N	110	n
15	SI	47	/	79	O	111	o
16	DLE	48	0	80	P	112	p
17	DC1	49	1	81	Q	113	q
18	DC2	50	2	82	R	114	r
19	DC3	51	3	83	S	115	s
20	DC4	52	4	84	T	116	t
21	NAK	53	5	85	U	117	u
22	SYN	54	6	86	V	118	v
23	ETB	55	7	87	W	119	w
24	CAN	56	8	88	X	120	x
25	EM	57	9	89	Y	121	y
26	SUB	58	:	90	Z	122	z
27	ESC	59	;	91	[	123	{
28	FS	60	<	92	\	124	
29	GS	61	=	93	]	125	}
30	RS	62	>	94	^	126	~
31	US	63	?	95	_	127	DEL

过对其  
种处理

五笔字型  
(18030)、  
(546)

# 本章主要内容

- ❖ 计算机的发展历史
- ❖ 计算机的数据表示
- ❖ 计算机的基本组成
- ❖ 计算机的基本工作原理



# 计算机的基本组成（1/9）

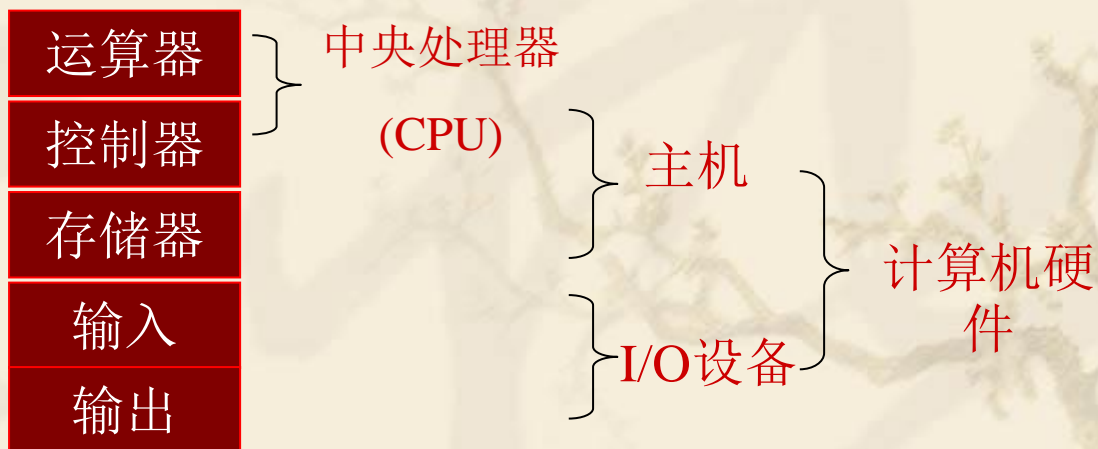
## ——冯·诺依曼计算机



**John von Neumann**  
冯·诺依曼

### 冯·诺依曼计算机特征

- 二进制存储程序和数据
- 顺序自动执行
- 由五个部分组成：运算器、控制器、存储器、输入和输出设备





# 计算机的基本组成（2/9）

## ——外设：输入设备

❖ **输入设备**是向计算机输入信息的装置，用于向计算机输入原始数据和处理数据的程序。

❖ **分类（按信息类型）**

❧ **数字和文字**输入设备（键盘、写字板等）



❧ **位置和命令**输入设备（鼠标器、触摸屏等）



❧ **图形**输入设备（扫描仪，数码相机等）



❧ **声音**输入设备（话筒，MIDI演奏器等）



❧ **视频**输入设备（摄像头、摄像机）



❧ **温度、压力**输入设备（温度、压力传感器）



# 计算机的基本组成（3/9）

## ——外设：输出设备

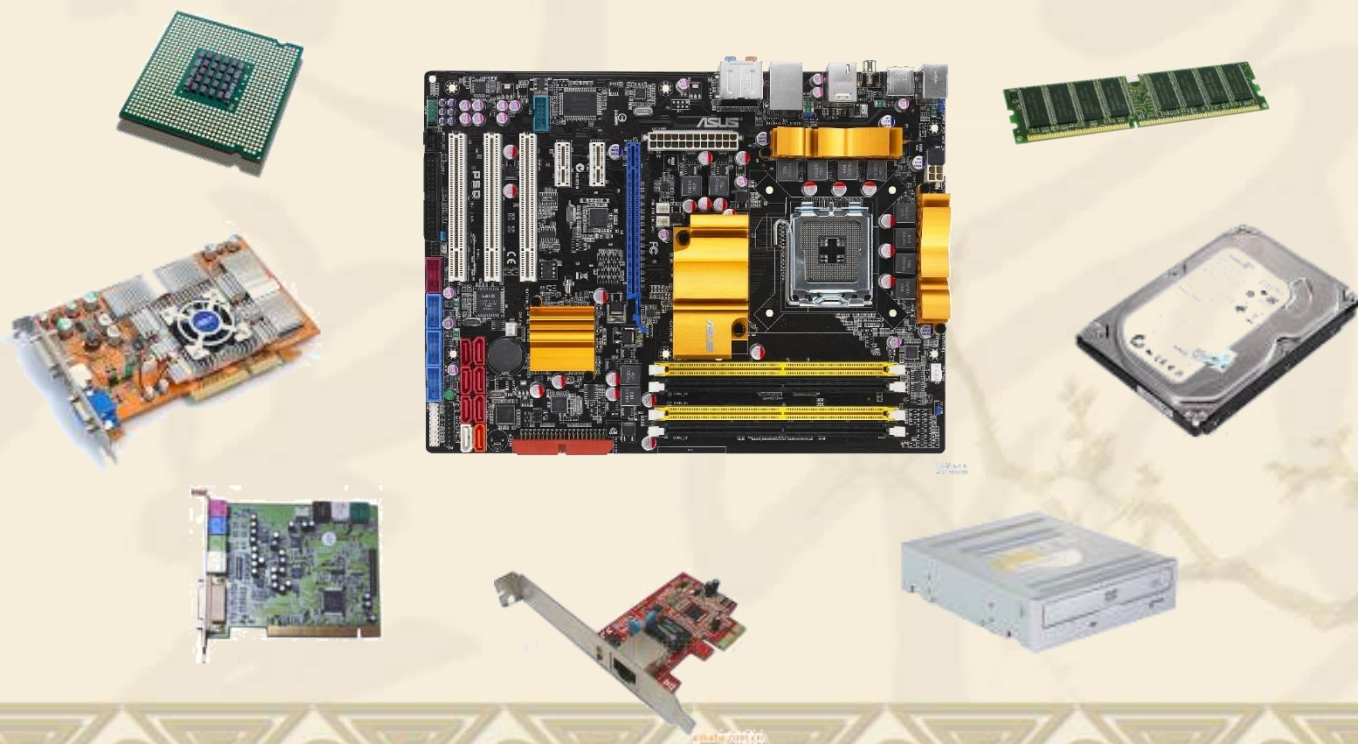
- ❖ **输出设备**主要用于将计算机处理过的信息保存起来或以人们能接受的数字、文字、符号、图形、图像和声音等形式显示、打印或播放出来。
- ❖ 常用的输出设备有显示器、音箱和打印机等。



# 计算机的基本组成（4/9）

## ——主机

- ❖ 主机主要包括处理器（以及各种协处理器）、存储器和连接它们的主板。

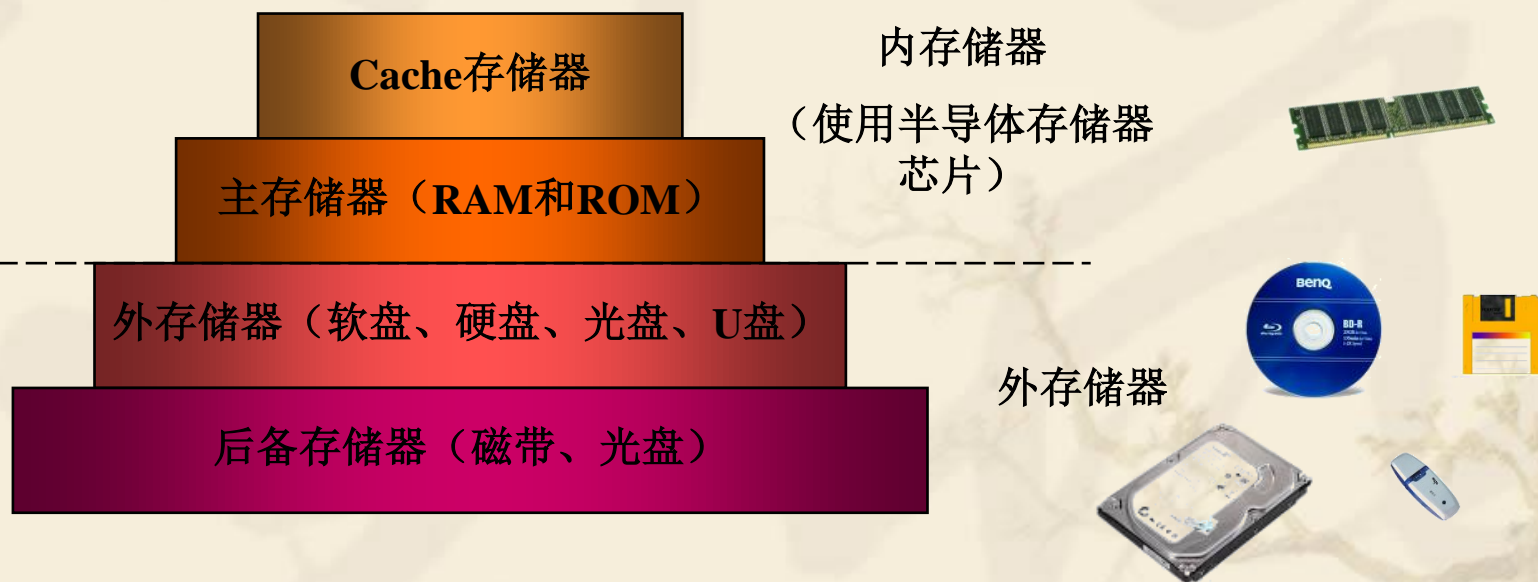




# 计算机的基本组成（5/9）

## ——主机：存储器

❖ **存储器（Memory）**：是具有记忆能力的部件，用来存储程序和数据。



冯·诺依曼计算机特征：存储程序（二进制）



# 计算机的基本组成（6/9）

## ——主机：存储器

### ❖ 信息的存储单位量纲

- ❧ 位(bit, 缩写为b), 二进制信息最小单位。
- ❧ 字节(Byte, 缩写为B), 信息存储基本单位。
- ❧ 字(Word)是可作为独立的信息单位进行处理的若干位的组合, 其包含的二进制位个数称为字长。

### ❖ 数量单位

- ❧ KB(Kilobytes)=  $2^{10}$  bytes = 1024bytes
- ❧ MB(Megabytes)=  $2^{10}$  KB
- ❧ GB(Gigabytes)=  $2^{10}$  MB
- ❧ TB(Terabytes)=  $2^{10}$  GB
- ❧ PB(Petabytes)=  $2^{10}$  TB

# 计算机的基本组成（7/9）

## ——主机：处理器

- ❖ 中央处理器（CPU，Central Processing Unit）是计算机系统的核心，负责运算和控制。



- ❖ 协处理器是辅助的运算和控制单元

- ❧ 浮点协处理器
- ❧ 图形协处理器
- ❧ 音频协处理器
- ❧ 总线控制器



# 计算机的基本组成（8/9）

## ——中央处理器的构成

- ❖ 运算器（ALU, Arithmetic Logic Unit）

- ∞ 进行算术和逻辑运算

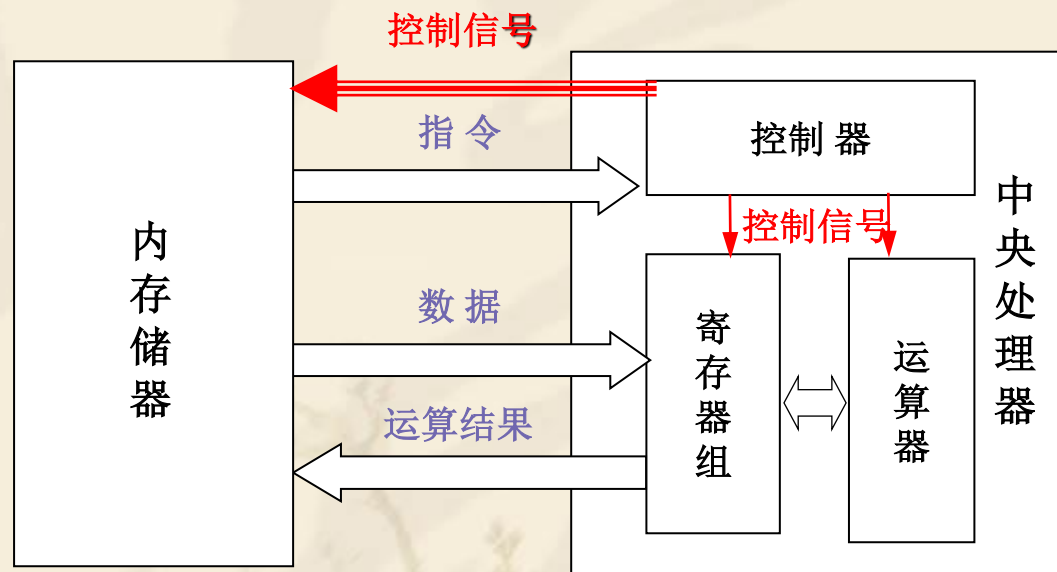
- ❖ 控制器（Controller）

- ∞ 从内存中取出指令

- ∞ 解读指令，并根据指令发出各种控制信号

- ❖ 寄存器（Register）

- ∞ 存放运算过程中的各种数据、地址或其他信息



冯·诺依曼计算机特征：自动执行

# 计算机的基本组成（9/9）

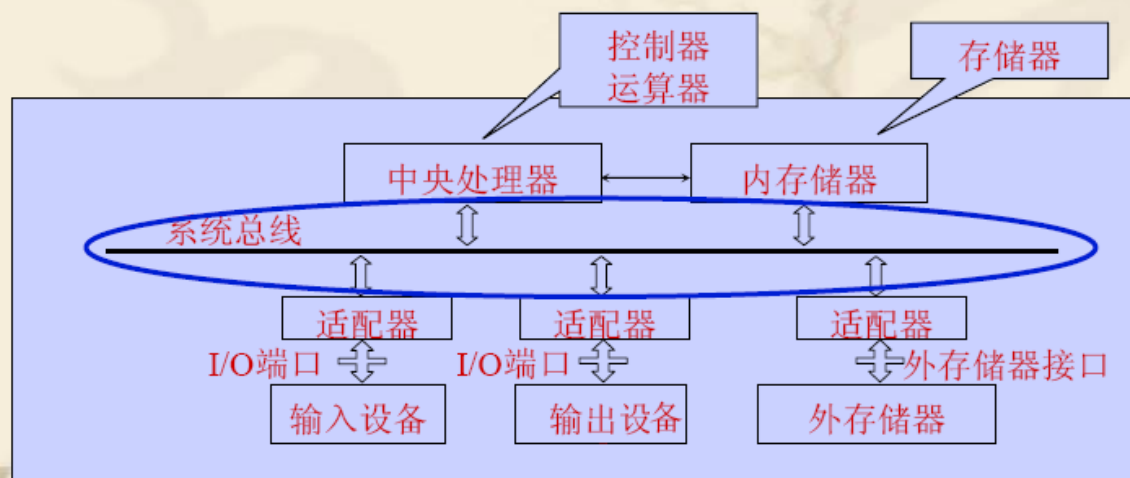
## ——主机：总线

❖ **总线（BUS）**是连接整个计算机硬件系统的公共通道，在各个部件间传递信息。

∞ 数据总线（DB, Data Bus）

∞ 地址总线（AB, Address Bus）

∞ 控制总线（CB, Control Bus）





# 本章主要内容

- ❖ 计算机的发展历史
- ❖ 计算机的数据表示
- ❖ 计算机的基本组成
- ❖ 计算机的基本工作原理

# 计算机的基本工作原理（1/3）

- ❖ 计算机基本工作原理：存储程序和程序控制
- ❖ 存储程序：计算机指令

∞ 操作码

∞ 操作数（寻址）

- 立即寻址
- 寄存器寻址
- 存储器寻址

指令格式

操作码	目的操作数	源操作数
-----	-------	------

<b>MOV</b>	<b>AL, 7</b>	<b><u>1011 0000 0000 0111</u></b>
<b>ADD</b>	<b>AL, 10</b>	<b><u>0000 0111 0000 1010</u></b>
<b>HLT</b>		<b><u>1111 0100</u></b>

# 计算机的基本工作原理（2/3）

## ❖ 计算机的指令系统

❧ 复杂指令系统（CISC）：IA-32/64（X86）

❧ 精简指令系统（RISC）：ARM、MIPS、PPC、SPARC

## ❖ X86指令系统

❧ 传送类指令（MOV、PUSH、POP、IN、OUT...）

❧ 运算类指令（ADD、SUB、MUL、DIV、CMP...）

❧ 逻辑类指令（AND、OR、NOT...）

❧ 转移类指令（JMP、CALL、RET...）

❧ 控制类指令（HLT、NOP、STC、CLC...）

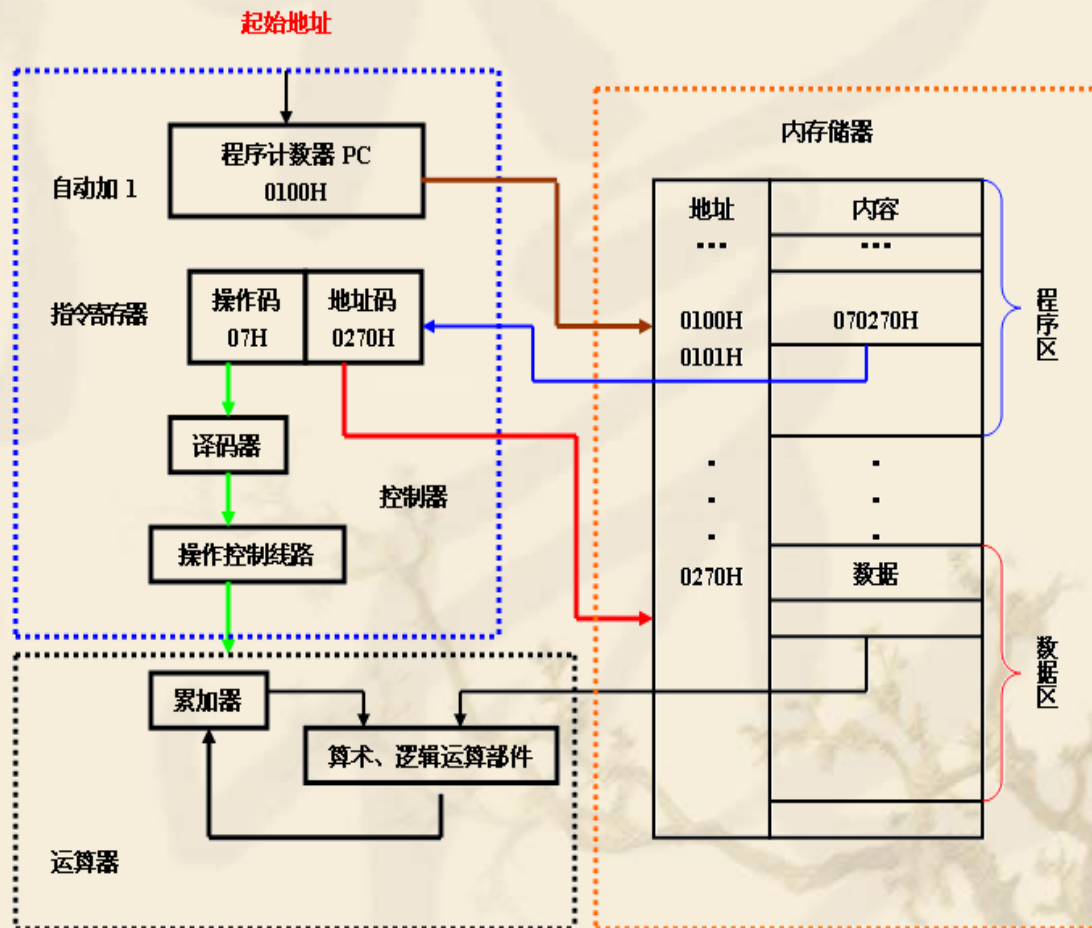
# 计算机的基本工作原理（3/3）

## ❖ 指令执行过程

- ❧ 取指令
- ❧ 分析指令
- ❧ 执行指令

## ❖ 寄存器功能

- ❧ 通用寄存器用于向运算器提供运算数据或保留运算结果
- ❧ 累加器A是可重复累加数据的通用寄存器
- ❧ 程序计数器PC存放将要执行的指令的地址
- ❧ 指令寄存器IR存放从内存中取出的指令





谢谢