计算机程序设计

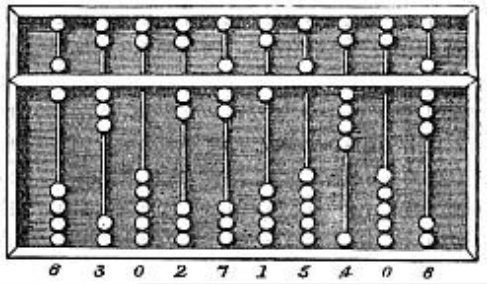
Computer Programming

计算机基础

主讲：李卫海

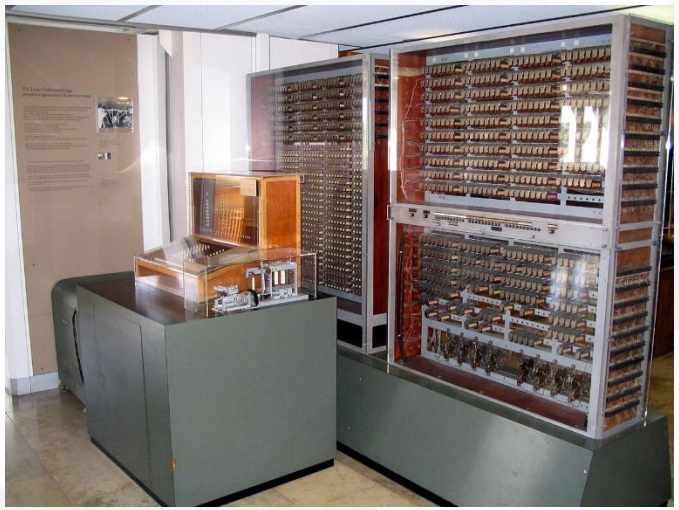
目录
CONTENTS

◎ 第一台计算机？



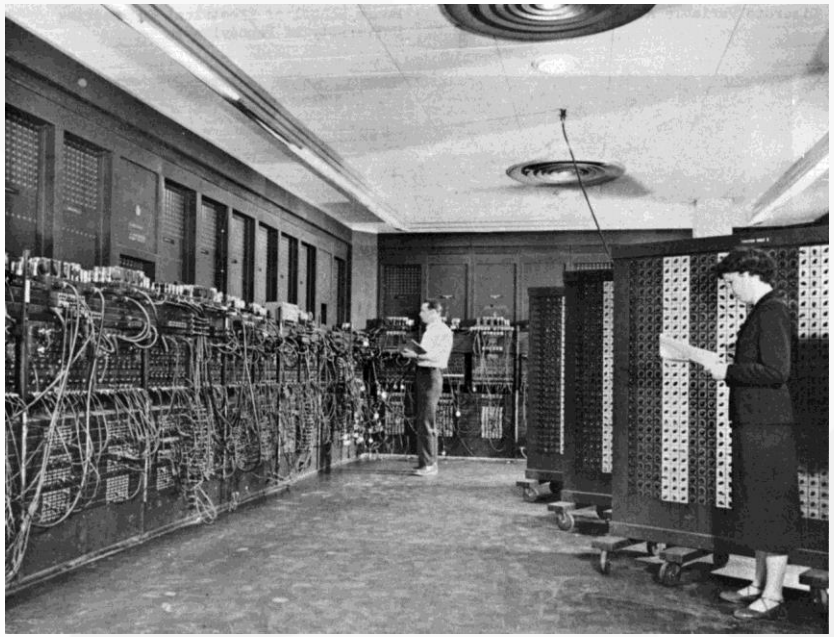
The Chinese Suanpan (算盘) (the number represented on this abacus is 6,302,715,408)
东汉记载，发明年代不明

The ancient Greek-designed Antikythera mechanism, dating between 150 and 100 BC, is the world's oldest analog computer.



Replica of Zuse's Z3, the first fully automatic, digital (electromechanical) computer.

◎ 第一台计算机？



ENIAC was the first electronic, Turing-complete device

1946年2月14日，世界上第一台电子可编程图灵计算机ENIAC(Electronic Numerical Integrator and Calculator)在美国宾夕法尼亚大学诞生

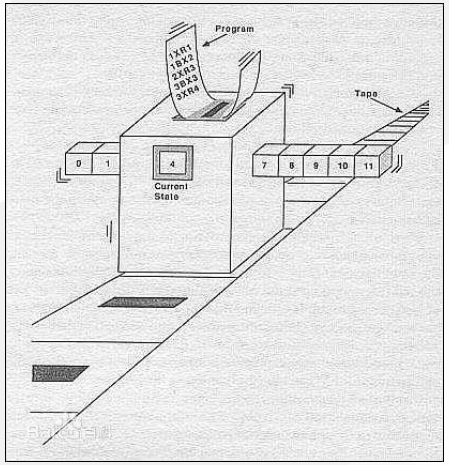
Mauchly博士和他的学生Eckert设计以真空管取代继电器，电子数字积分器与计算器)，用来计算炮弹弹道

这部机器使用了18800个真空管，占地1500平方英尺，重达30吨，每秒可进行5000次加法运算

图灵机

- 图灵机指一个抽象的机器，用来模拟人们用纸笔进行运算的过程。
  - 它有一条无限长的纸带，纸带分成了一个一个的小方格
  - 有一个控制器在纸带上移来移去。控制器内部有一组内部状态
  - 还有一些固定的程序。

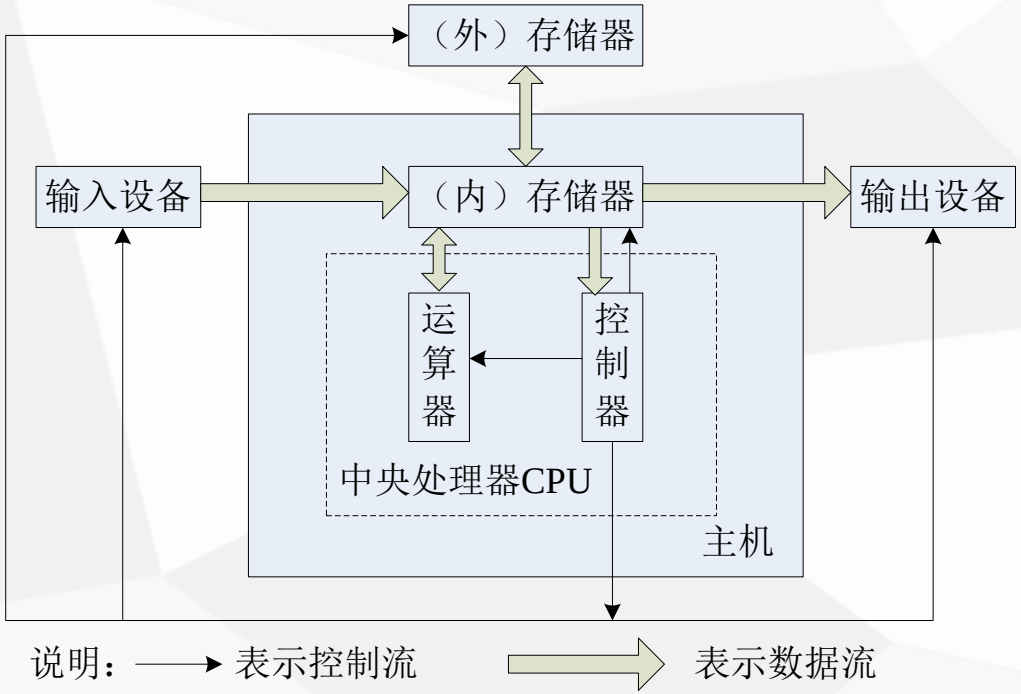
- 在每个时刻，控制器都从当前纸带上读入一个方格信息，然后结合自己的内部状态查找程序表，根据程序输出信息到纸带方格上，并转换自己的内部状态，然后进行移动。



计算机硬件——冯·诺伊曼结构

美籍匈牙利数学家冯·诺依曼

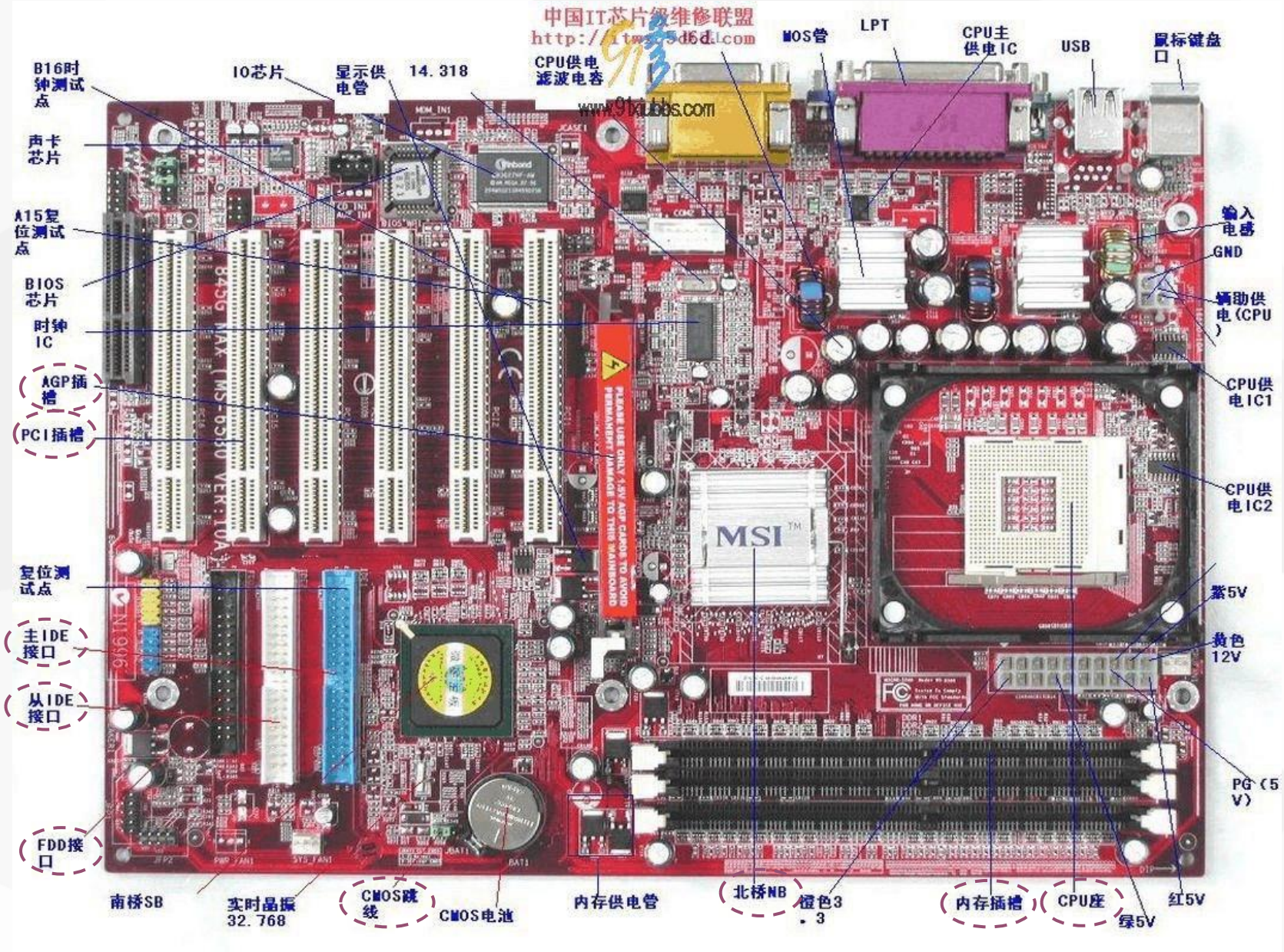
- 采用二进制编码形式表示数据和程序
- 要执行的程序和被处理的数据预先放入计算机中，计算机能够自动地从内存中取出指令执行
- 计算机由五大基本部件组成
  - 运算器Arithmetic Logic Unit
  - 控制器Control Unit
  - 存储器Memory Unit
  - 输入设备Input Device
  - 输出设备Output Device
- 计算机的基本工作原理
  - 控制流：控制器发给各部件
  - 数据流：数据、指令
    - 指令：完成一个基本操作的命令
    - 指令系统/指令级：指令集合
    - 程序：完成特定任务的指令序列



普林斯顿结构

哈佛结构：将程序指令存储和数据存储分开

◎ 微型计算机的硬件组成


B16时钟测试点
IO芯片
显示供电管
14.318
CPU供电滤波电容
MOS管
LPT
CPU主供电IC
USB
鼠标键盘口
声卡芯片
A15复位测试点
BIOS芯片
时钟IC
AGP插槽
PCI插槽
输入电感
GND
辅助供电(CPU)
CPU供电IC1
CPU供电IC2
复位测试点
主IDE接口
从IDE接口
FDD接口
紫5V
黄色12V
PG(5V)
南桥SB
实时晶振32.768
CMOS跳线
CMOS电池
内存供电管
北桥NB
橙色3.3
内存插槽
CPU座
绿5V
红5V

◎ 微型计算机的主要性能指标

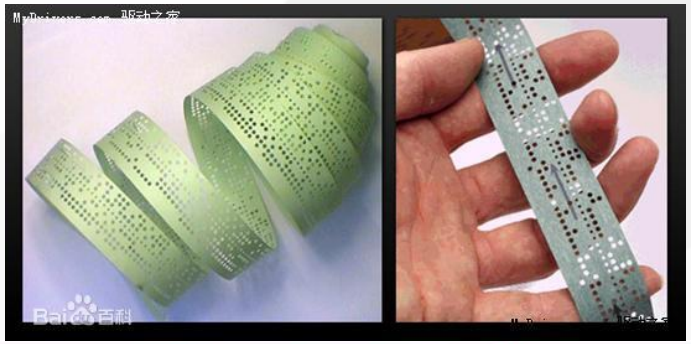
- 字长
  - 8位、12位、16位、32位、64位
- 速度
  - 运算速度 MIPS（Million Instructions Per Second)
  - 主频 MHz、GHz
- 内存容量
  - Byte、KB、MB、GB
- 存取周期
- 外部设备的配置和软件的配置

◎ 计算机软件

- 系统软件
  - System Software，管理计算机全部资源（硬件和应用软件等）的软件，包括操作系统（DOS、Windows、Linux、Unix、软件系统的核心）、语言处理程序、数据库管理系统、网络管理程序、工具与服务程序等
  - 文件系统
    - FAT、NTFS、EXT
- 应用软件
  - 提供给用户的程序，如办公软件（Office）、软件开发环境（如Visual Basic）、管理软件、设计软件（如AutoCAD）、财务软件、游戏软件等
- 软件的划分比较模糊，如数据库管理也可以算是应用软件，关键看提供给不同层次用户的配置

程序设计语言

- 低级语言
  - 面向机器的语言
    - 机器语言
    - 汇编语言
- 高级语言
  - 面向过程的语言
  - 面向对象的语言
    - Fortran
    - BASIC/QB/VB
    - Pascal
    - C/C++/C#
    - Java
    - Python
    - …
- 语言处理程序
  - 编译器：高级语言→机器语言→执行
  - 解释器：逐句翻译执行



arm64机器语言与汇编语言

```
BF C3 1E B8   stur   wzr, [x29, #-0x14]
BF 83 1E B8   stur   wzr, [x29, #-0x18]
A8 83 5E B8   ldur   w8, [x29, #-0x18]
1F 91 01 71   cmp    w8, #0x64
2A 01 00 54   b.ge   0x102b926a0
A8 83 5E B8   ldur   w8, [x29, #-0x18]
A9 C3 5E B8   ldur   w9, [x29, #-0x14]
28 01 08 0B   add    w8, w9, w8
A8 C3 1E B8   stur   w8, [x29, #-0x14]
A8 83 5E B8   ldur   w8, [x29, #-0x18]
08 05 00 11   add    w8, w8, #0x1
A8 83 1E B8   stur   w8, [x29, #-0x18]
F6 FF FF 17   b      0x102b92674
A8 C3 5E B8   ldur   w8, [x29, #-0x14]
E9 03 08 AA   mov    x9, x8
EA 03 00 91   mov    x10, sp
49 01 00 F9   str    x9, [x10]
00 00 00 B0   adrp   x0, 1
00 F8 3D 91   add    x0, x0, #0xf7e
D5 00 00 94   bl     0x102b92a0c
```

IA-64机器语言与汇编语言

```
C7 45 EC 00 00 00 00   movl   $0x0, -0x14(%rbp)
C7 45 E8 00 00 00 00   movl   $0x0, -0x18(%rbp)
83 7D E8 64            cmpl   $0x64, -0x18(%rbp)
0F 8D 17  00 00 00     jge    0x101a48b02
8B 45 E8               movl   -0x18(%rbp), %eax
03 45 EC               addl   -0x14(%rbp), %eax
89 45 EC               movl   %eax, -0x14(%rbp)
8B 45 E8               movl   -0x18(%rbp), %eax
83 C0 01               addl   $0x1, %eax
89 45 E8               movl   %eax, -0x18(%rbp)
E9 DF FF FF FF         jmp    0x101a48ae1
48 8D 3D 27 22 00 00   leaq   0x2227(%rip), %rdi
8B 75 EC               movl   -0x14(%rbp), %esi
B0 00                  movb   $0x0, %al
E8 D1 0C 00 00         callq  0x101a497e4
```

C语言

```
int sum = 0;
for (int i = 0; i < 100; i++)
{
    sum += i;
}
printf("sum:%d",sum);
```

计算机的信息存储

- 冯·诺依曼结构计算机中，所有的信息都采用二进制编码进行存储与处理，原因
  - 易于物理实现：仅有0、1，物理实现简单可靠
  - 运算简单：对R进制进行求和或求积运算时，其运算规则为R(R+1)/2种，而二进制仅需三种
  - 机器可靠性高：仅量变，无质变，抗干扰
  - 通用性强：不仅用于数值信息编码，也与逻辑真假对应，易于逻辑运算

◎ 计算机的信息存储

- 信息(程序源代码、数据等)在计算机中以二进制方式存储在内存、硬盘等存储器上，存储的最小单位是bit(位)，1位中的内容要么是0，要么是1

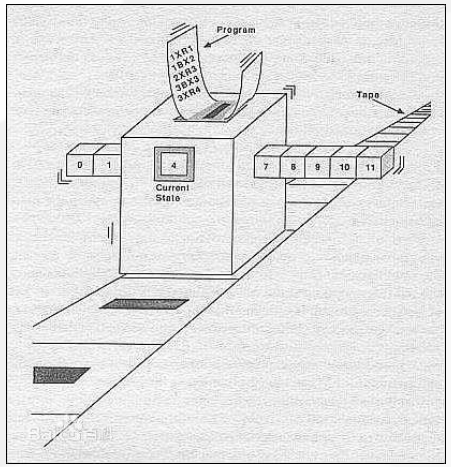
- 1bit太小，为便于使用，常以1个字节(Byte)作为基本存储单元计数，1Byte = 8bits，即1B = 8b

- 1KB=1024Bytes，即2^10个字节
  1MB=1024KB，即2^20个字节
  1GB=1024MB，即2^30个字节
  1TB=1024GB，即2^40个字节

计算机的信息存储

- 内存编址：以1MB大小的内存为例，以字节为基本存储单元，按顺序为这个存储空间编址，则有：

第一个字节的内容为0x96
内容的具体意义由操作系统和运行程序确定



1	0	0	1	0	1	1	0	← 地址为0000 0000 0000 0000 0000 (00000H)
1	1	1	0	0	0	0	0	← 地址为0000 0000 0000 0000 0001 (00001H)
0	0	1	0	0	1	0	1	← 地址为0000 0000 0000 0000 0010 (00002H)
⋮								
1	1	1	1	0	0	1	1	← 地址为1010 0111 1100 0001 0010 (A7C12H)
⋮								
1	1	0	0	1	0	0	0	← 地址为1111 1111 1111 1111 1110 (FFFFEH)
0	1	1	0	1	0	0	1	← 地址为1111 1111 1111 1111 1111 (FFFFFH)

1MB=2^20，20比特，00000H~FFFFFH

◎ 计算机的信息存储

- 32位、64位计算机
  - 32/64位计算机实际指CPU中的寄存器位数，也称计算机字长，是计算机一次基本操作能处理的最大数据位数
  - 32/64位同时也指数据总线宽度，即地址表示范围；最大的数据占用字节数
  - 32位计算机最多只能使用4G内存，2^32=4G
- 32位、64位操作系统
  - 操作系统是硬件和应用软件中间的一个平台
  - 32/64位操作系统针对32/64位的CPU设计指令（指令instruction是指挥机器工作的二进制指示和命令）
  - 32位操作系统最多只能管理4G内存
- 操作系统与计算机位数匹配才能高效工作
  - 64位计算机可装32位操作系统，此时高32位数据总线和寄存器位闲置
  - 32位计算机不能装64位操作系统

◎ 数的位置计数法及进制

- 采用R个基本符号计数，称为R数制，逢R进位， R为“基数”，每一位对应的单位称为“权”
- 一个数可按权展开成多项式，如十进制数19.08，可表示为

$$19.08 = 1 \times 10^1 + 9 \times 10^0 + 0 \times 10^{-1} + 8 \times 10^{-2} = \sum_{i=n-1}^{-m} a_i R^i$$

- 10^2、10^1、10^0、10^{-1} 等称为十进制的位权
- 此称为位置计数法，展开式称为按权展开式

数的位置计数法及进制

- 在不同进制下，同一个数代表的值不同，如

$$(11011)_{10}=1\times10^4+1\times10^3+0\times10^2+1\times10^1+1\times10^0=(11011)_{10}$$

$$(11011)_{2}=1\times2^4+1\times2^3+0\times2^2+1\times2^1+1\times2^0=(27)_{10}$$

$$(11011)_{8}=1\times8^4+1\times8^3+0\times8^2+1\times8^1+1\times8^0=(4617)_{10}$$

$$(11011)_{16}=1\times16^4+1\times16^3+0\times16^2+1\times16^1+1\times16^0=(69649)_{10}$$

- 不同进制的数，可用数字或字母脚标区别
  - 二进制Binary：B，如10110B或10110_B
  - 十六进制Hexadecimal：H，如9A0H、$9A0_H$、0x9A0或0X9A0
  - 八进制Octal：O，用Q代替以避免与数字0混淆，如275Q或275_Q
  - 十进制Decimal：D，也可用空标识，如356、356D、或356_D

进制转换

- R进制转十进制：直接按权展开求和

  $12345_Q=1*8^4+2*8^3+3*8^2+4*8^1+5*8^0$

  $AB12_H=10*16^3+11*16^2+1*16^1+2*16^0$

- 快速算法
  - 整数部分

101001B = 41

1 0 1 0 0 1

= = = = = = =

1 2 5 10 20 41

*2 *2 *2 *2 *2

  - 小数部分

0.101001B = 0.640625

0.1 0 1 0 0 1

= = = = = = =

0.640625 0.28125 0.5625 0.125 0.25 0.5

/2+0.5* /2+0.5* /2+0.5* /2+0.5* /2+0.5*

进制转换

- 十进制转R进制：
  - 整数部分采用除基数取余法
    - 最先得到的余数是整数的最低位
  - 小数部分采用乘基数取整法
    - 最先取到的整数是小数点后第一位
- 例如：

28转换为2进制

2	28	
2	14	0
2	7	0
2	3	1
2	1	1
	0	1

28=11100B

0.28转换为2进制

0.28*2=0.56　0
0.56*2=1.12　1
0.12*2=0.24　0
0.24*2=0.48　0
0.48*2=0.96　0
0.96*2=1.92　1
……

0.28=0.010001…B

进制转换

- 二进制与八进制、十六进制之间的转换
  - 3位二进制对应1位八进制

    $11\ 010\ 010_B \rightarrow 322_Q$

  - 4位二进制对应1位十六进制

    $1101\ 0010_B \rightarrow D2_H$

  - 反之亦然

计算机的信息存储

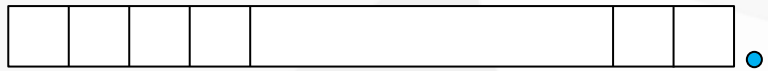
- 计算机中的数据
  - 数值数据：无符号数、有符号数
  - 非数值数据：文字、图像、视频、语音等

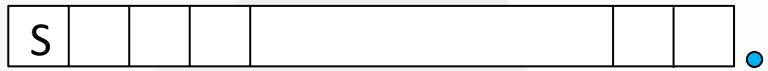
- 数据的类型一般是一定的，有约定的长度
  - 有符号数：最高位作为符号位，分为定点数、浮点数
  - 无符号数：无符号位，多用于表示非负整数、字符、地址及逻辑值等

◎ 定点数（纯整数、纯小数）

- 定点整数：小数点固定在最低位，纯整数
- 定点小数：小数点固定在符号位后的最高位，纯小数
- 符号位：最高位，0表示正，1表示负

纯整数000…0B～111…1B





纯小数0.000…0B～0.111…1B

表示范围：对应整数的表示范围除2^N

表示范围：对应整数的表示范围除2^{N-1}

- 以此表示的数称为“机器数”，原来的数称为“真值”
- 机器数的三种形式：原码、反码和补码

◎ 原码

- 符号位0为正，1为负
- 数值部分为真值的绝对值的二进制数
- 如 $[+100]_{原}$=01100100B

  $[-100]_{原}$=11100100B

机器数	0000 0000B	…	0111 1111B	1000 0000B	…	1111 1111B
原码真值	0	…	127	-0	…	-127

- 0的表示不唯一（+0=-0，即00000000B=10000000B）
- 表示范围：$-2^{N-1}+1 \sim 2^{N-1}-1$
- 加法运算时，符号相同可直接加，符号不同时，需要先比较绝对值然后大数减小数，结果的符号由大数决定。因此，原码不便于运算

◎ 反码

- 符号位不变，0为正，1为负
- 正数的反码与原码相同，负数的反码，其数值部分为原码的数值部分按位取反（相当于连同符号位一起取反）
- 如 $[+100]_{反}$=01100100B

  $[-100]_{反}$=10011011B

机器数	0000 0000B	…	0111 1111B	1000 0000B	…	1111 1111B
原码真值	0	…	127	-0	…	-127
反码真值	0	…	127	-127	…	-0

- 0的表示不唯一（+0=-0，即00000000B=11111111B）
- 表示范围： $-2^{N-1}+1 \sim 2^{N-1}-1$
- 反码同样不便于运算

◎ 补码

- 符号位不变，0为正，1为负
- 正数的补码与原码相同，负数的补码，其数值部分取反码再+1
- 如 $[+100]_{补}$=01100100B

  $[-100]_{补}$=10011011B+1B= 10011100B

机器数	0000 0000B	…	0111 1111B	1000 0000B	…	1111 1111B
原码真值	0	…	127	−0	…	−127
反码真值	0	…	127	−127	…	−0
补码真值	0	…	127	−128	…	−1

- 0有唯一的补码00000000B
- 补码10000000B是−128，但不存在原码和反码
- 表示范围： $-2^{N-1}\sim 2^{N-1}-1$

◎ 补码运算简单方便

- 补码运算的结果仍是补码

-2+3	10-67
$[-2]_{补}$=11111110B	$[+10]_{补}$=00001010B
$[+3]_{补}$=00000011B	$[-67]_{补}$=10111101B
+ ----------	+ ----------
$[+1]_{补}$=100000001B	$[-57]_{补}$=11000111B

- 减法可以视为补码加法，符号位当作数据位直接参与运算后仍然有效，简化了硬件电路

- 因乘法可用加法实现，除法可用减法实现，使用补码时，计算机中只要有加法器，就可以实现四则运算（后来为提高运算速度，增加了专门的乘法器和除法器）

◎ 补码运算的溢出

$$[-2]_{补}=11111110B$$
$$[+3]_{补}=00000011B$$
$$+\ \text{----------}$$
$$[+1]_{补}=100000001B$$

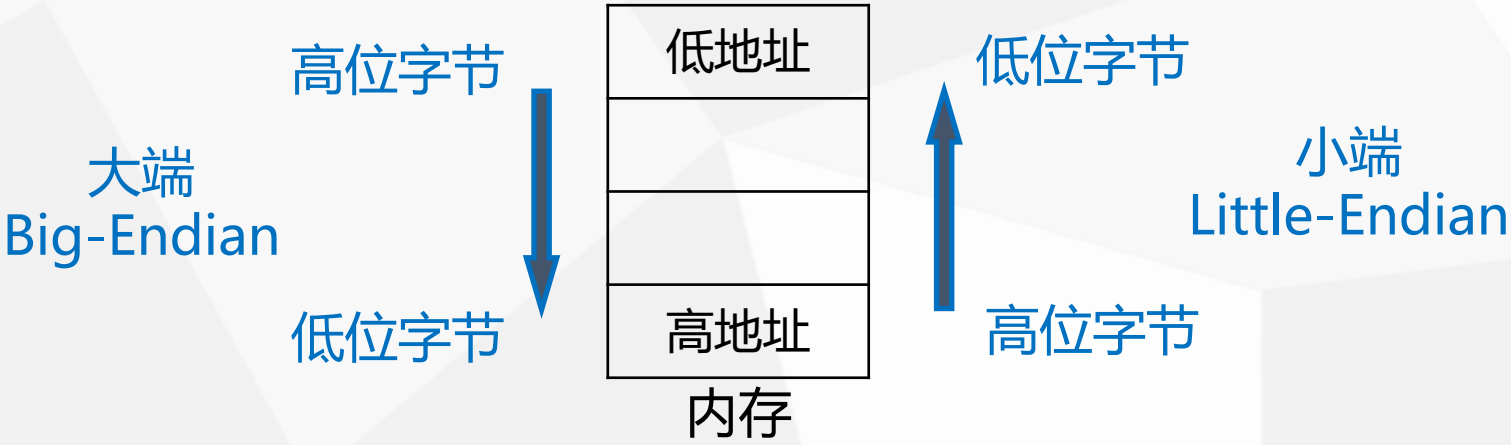
最高有效位进位

符号位进位

- [正]+[正]不溢出时，符号位为0，最高有效位和符号位不进位；溢出时，符号位变为1，最高有效位进位，符号位不进位
- [正]+[负]不会溢出。若结果为负，最高有效位不进位，符号位不进位；若结果为正，最高有效位进位，符号位进位
- [负]+[负]不溢出时，符号位为1，最高有效位和符号位进位；溢出时，符号位变为0，最高有效位不进位，符号位进位
- **溢出标志：最高有效位进位⊕符号位进位**

大端与小端

- 当一个整数占用多个字节时，这些字节在内存中的排列顺序



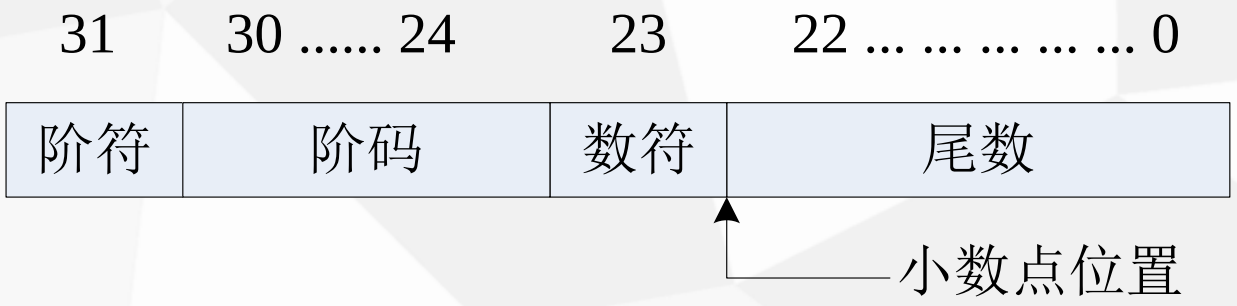
- 例如：12345678H，以大端方式保存时，在内存中从低地址到高地址依次保存为12H,34H,56H,78H；以小端方式保存时，在内存中从低地址到高地址依次保存为78H,56H,34H,12H
- 不同的系统采用的方式不同

◎ 浮点数（一般实数）

- 一般实数用科学计数法表示，采用二进制时

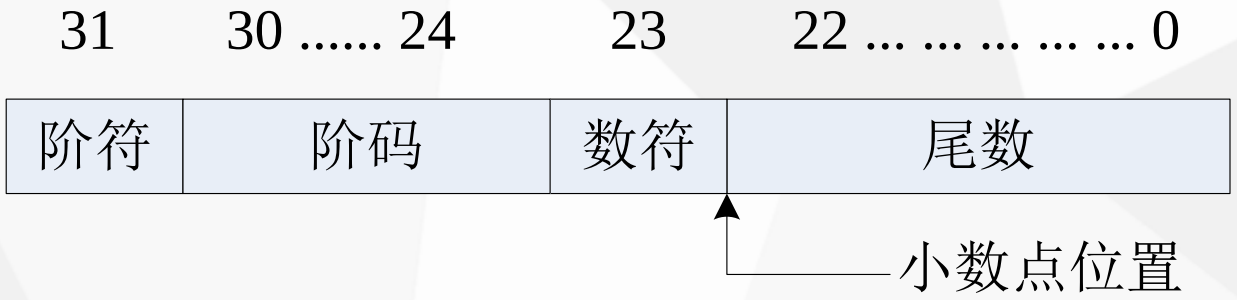
$$X = \pm M \times 2^{\pm E}$$

  - M是二进制纯小数(0.???...)，称为数X的尾数，代表了X的全部有效数字，其位数反映了数据的精度
  - E是二进制整数，称为X的阶码(阶次)，决定了数的范围
  - M、E有符号，可以采用原码或补码表示



有符号实数表示范围

- 浮点小数（非IEEE 754的一般形式）
  - 数符号用1位表示
  - 阶码E用r位有符号整数补码表示，范围-2^{r-1}～$2^{r-1}-1$
  - 尾数M用n位无符号定点原码表示，范围0～$1-2^{-n}$
  - 浮点数范围$-(1-2^{-n})*2^{k}$～$(1-2^{-n})*2^{k}$，其中$k=2^{r-1}-1$



r=8，k=127
n=23
表示范围 $-(1-2^{-23})*2^{127}$～$(1-2^{-23})*2^{127}$

IEEE standard 754

- 不同的计算机可能采用不同的浮点数表示
- IEEE standard 754标准（从高位到低位）：$X = \pm M \times 2^{\pm E}$　　M规格化处理：M=1.????
  - 单精度32位

31	30 …… 23	22 …… 0
符号位	指数（移码）	有效数（原码，舍弃最高位）

  - 双精度64位

63	62 …… 52	51 …… 0
符号位	指数（移码）	有效数（原码，舍弃最高位）

移码：真值加127/1023（8比特/11比特）后的原码

例：5.0：0 10000000001 01000000000000000000……

$1.01*2^2$　　2+1023　　最高位1不写入内存

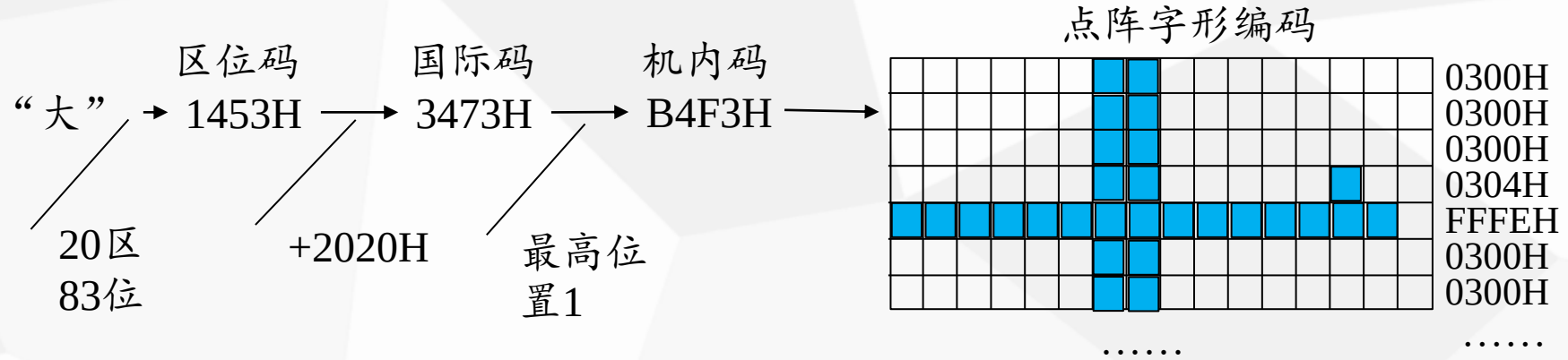
  - 移码为0：尾数*2^(-127)（或-1023);
  - 移码全1：尾数为0时代表INF，其它值时代表NAN

◎ 非数值数据

- 非数值数据包括西文字符、汉字字符以及声音、图像等信息，也必须用二进制数进行编码才能进行存储和处理
- 西文字符编码（ASCII码，教材附录B）
  - American Standard Code for Information Interchange，由ANSI (American National Standard Institute, 美国国家标准学会)制定
  - 标准ASCII码用7位二进制数表示128个字符，分为控制字符（0-31，127）和图形字符
  - 扩展ASCII码用8位二进制数表示256个字符，扩充出来的符号包括表格符号、计算符号、希腊、拉丁字母等
- Unicode码
  - 统一码、万国码、单一码，国际标准，以两个字节表示一个字符，可编码表示世界上几乎所有的书面语言
  - Unicode码的65536个可用编码中，39000个已用，其中21000个用于表示汉字
  - Microsoft Office即基于Unicode文字编码标准

◎ 非数值数据

- 汉字字符编码
  - 常用汉字数千个，目前采用两个**字节**编码
  - 区位码、国标码（有兴趣可自行了解）
  - 区分两个字节是两个ASCII字符还是一个汉字
    - 两个字节首位都为1是汉字，为0是ASCII字符，也称为机内码
  - 汉字输入法采用不同的编码方案，称为输入码，均需要转换为机内码
  - 另有显示或打印汉字所需的点阵字形编码



◎ 非数值数据

- 多媒体信息编码
  - 计算机中的媒体（medium）一般指传递信息的载体，如文本、图像、视频、声音等
  - 多种媒体的结合称为多媒体（Multimedia）
  - 多媒体信息的编码方法与媒体本身的特性（存储、传输软硬件要求等）有关，采用专有的编码格式
  - 涉及到压缩
  - 后续专业课会介绍部分编码格式

◎ 什么是计算思维？

- 百度百科：2006年，周以真提出，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、以及人类行为理解等涵盖计算机科学之广度的一系列思维活动。
  - 数学思维？物理思维？化学思维？

- 高斯：1+2+3+. . . . +100=？
- 专业技术←→计算技术
  - “此两者同出而异名，同谓之玄。玄之又玄，众妙之门。”
- 要花多大精力来优化？付出与收益的考量
  - 魂斗罗128KB←→王者荣耀3GB

编程的概念

- 程序
  - 计算机思维是人脑思维的量化和延伸
  - 计算机语言是工具，需要在拟定算法的基础上编写程序，才能正确解决问题
  - 程序是为完成预定任务，用某种计算机语言编写的一组指令序列；计算机按照程序规定的流程执行指令以完成任务
- 编程=数据结构+算法+程序设计语言
  - 首先根据问题设计算法，选择数据结构

  - 使用某一种程序设计语言进行编码

◎ 编程的概念

- 数据结构（Data Structure）
  - 算法实现的基础
  - 数据对象（整数、实数、字符、符号等）及其相互关系和构造方法
  - 描述程序中的数据间的组织形式和结构关系

  - 算法决定了用什么数据结构描述数据更佳
  - 语言决定了有什么数据结构可以使用

编程的概念

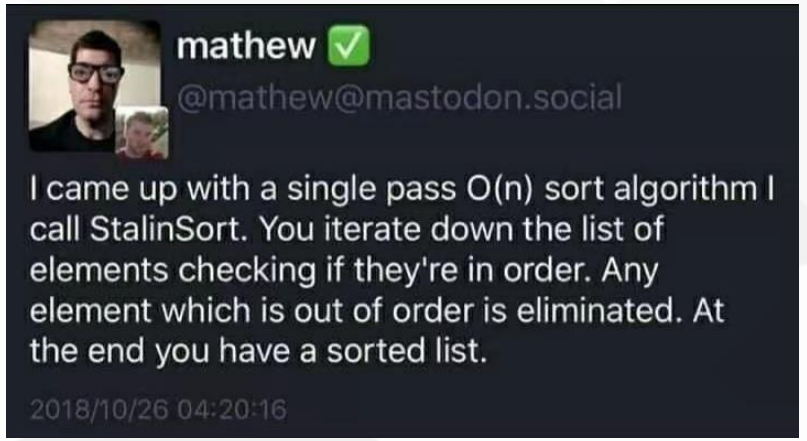
- 算法（Algorithm）
  - 问题的求解方法，由一组简单指令和规则组成，供计算机在有限的步骤内解决问题
  - 正确的算法要求
    - 组成算法的规则和步骤的意义应是唯一确定的，没有二义性；由这些规则指定的操作是有序的，并能给出问题的结果
  - 好的算法
    - 结果优，消耗资源少，计算过程快
    - 实现容易，原理易懂
  - 语言和数据结构限制了算法的选择

编程的概念

- 例如，求两个整数的最大公约数
  - 欧几里德算法（辗转相除法）

    问题描述：给定两个正整数m和n，求它们的最大公因子，即能同时整除m和n的最大正整数
    - 步骤1：以n除m，并令r为所得余数
    - 步骤2：若r=0，算法结束，n即为m和n的最大公因子
    - 步骤3：置n→m，r→n，返回步骤1

- 算法应当具备
  - 可行性：所有操作可行
  - 有穷性：可在有限步之后结束
  - 正确性：执行结果应当是正确的
  - 健壮性：对各种输入应有适当的处理
  - 可读性：思路清晰，表述明确，解释清楚
  - 高效和低存储量



算法的表示

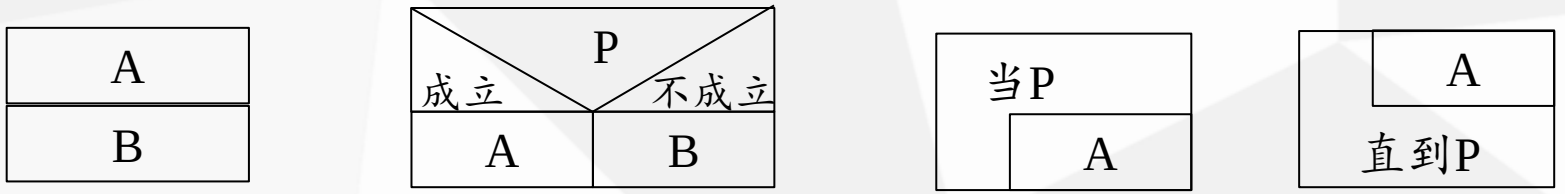
- 自然语言描述
  - 简单的通俗易懂，逻辑复杂的不易描述。不够严谨

- 伪代码表示
  - 介于程序设计语言和自然语言之间，帮助程序设计人员把注意力放在描述解决问题的思路，而不拘泥于具体的语言、语法
  - 保留程序设计语言的关键流程结构，细节用自然语言描述
  - 伪代码也有若干规范（如类Pascal，类C），但并无统一标准，一般应包含描述基本结构的语句
  - 规范：使用缩进，自然语言+某种常用语言的格式语句，忽略细节

◎ 算法的表示

- 流程图表示
  - 传统流程图

  - NS流程图



算法的设计原则

- 明确输入、输出
- 自顶向下，逐步求精
  - 积累经验：什么是能实现的，什么是难实现的，什么是不能实现的
- 模块化设计
  - 积累经验：是否要划分、如何划分
  - 影响团队合作、代码效率

◎ 算法的基本分类

- 直接法
- 枚举法、穷举法
- 递推法
- 递归法
- 贪心法
- 分支选择
- ……

- 程序设计重点在解决问题的思路，即设计解决方案，好的程序设计易于理解、交流和实现
- 程序设计语言是实现解决方案的具体手段，需要正确熟练掌握，以高质量完成预定任务
- 专业化的程序设计思想和方法是更为本质的知识和能力，是独立于具体程序设计语言的
- 具备了基本的程序设计能力后，掌握一门新的语言并不困难

某程序员对书法十分感兴趣，退休后决定在这方面有所建树。于是花重金购买了上等的文房四宝。

一日，饭后突生雅兴，一番磨墨拟纸，并点上了上好的檀香，颇有王羲之风范，又具颜真卿气势，定神片刻，泼墨挥毫，郑重地写下一行字：

```
Hello, World.
```

◎ 作业

1. 将下列十进制数转换成二进制、八进制、十六进制数（精确到小数点后4位）。

   123　　8962.5827

2. 将下列十六进制数转换为二进制数。

   85EH　　A7.2H　　387.15H

3. 将下列二进制数转换为十六进制数。

   10110111B　　10110.10110B