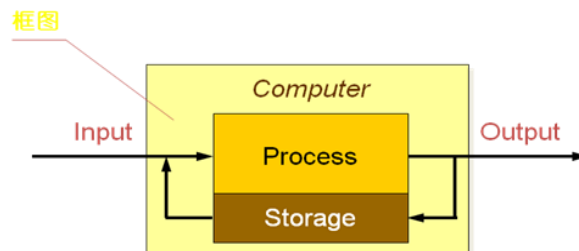


计算机与计算机科学

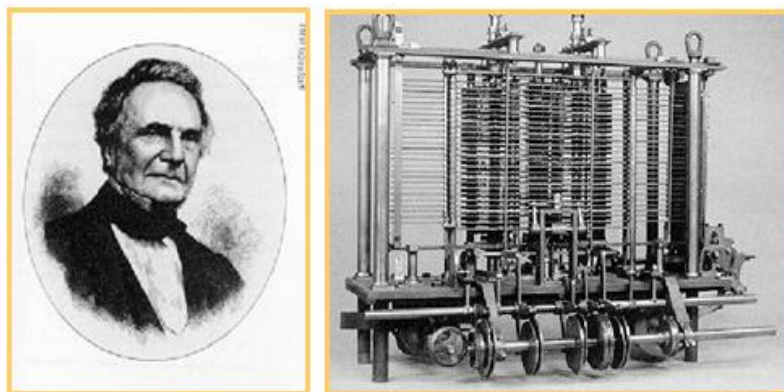
1.1 计算机概述

1.什么是计算机？

计算机是一种能够接收和存储信息，并把存储在内部的程序对输入信息进行加工、处理，得到人们所期望的结果，然后把处理结果输出的高度自动化设备。



巴比奇在马洪发明的逻辑演示器的影响下，于 1822 年开始设计差分机（difference engine），其目标是能计算具有 20 位有效数字的 6 次多项式的值。这是第一台可自动进行数学变换的机器，因此他被称为“计算之父”。



计算理论的奠基人（图灵）

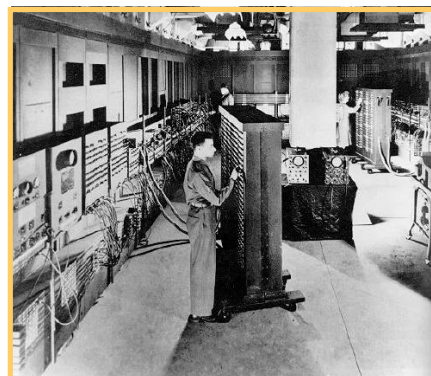
- Alan Turing (1912~1954) 1936 年上研究生时发表的一篇文章中提出了图灵机 (Turing Machine)，奠定了计算机的理论基础。
- 第二次世界大战中，Turing 领导的小组制造出了破译德军 Enigma 密码的计算机，并成功地完成了任务。
- Turing 与 Church 合作给出了数学证明，断言未来计算机能够象人那样具有思维能力（因而汉语中有了“电脑”）。
- 计算机学科的最高荣誉是 ACM（美国计算机学会）图灵奖。
- 图灵测验：



一个人在不接触对象的情况下，进行一系列的提问，如果他根据这些回答无法判断对象是人还是机器，则这种计算机具有与人相当的智力。

世界上第一台电子数字计算机

- 1946 年，在宾夕法尼亚大学 Moore 学院，J. Eckert 和 J. Mauchly 为实现弹道计算研制成功了 ENIAC（“电子数字积分计算机”，*Electronic Numerical*



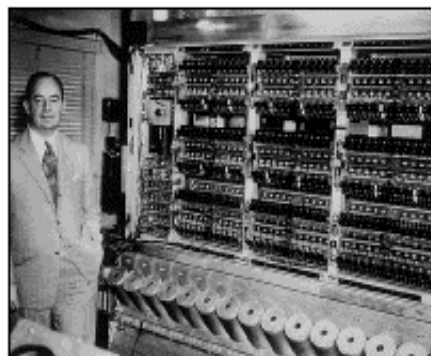
Integrator and Computer), 它用了 19000 个电子管, 重 30 吨, 耗电 200 千瓦。

- 现在所谓的计算机就是电子数字计算机。

计算机体系结构的创始人(冯·诺依曼)

● 1946 年, John von Neumann (1903~1957) 发表了第一篇关于电子计算机程序存储的论文, 描述了如何用(可被存储、读出和执行的)数字来表示逻辑操作(程序)。至今, 大多数计算机采用的都是 von Neumann 体系结构 (*von Neumann architecture*)。

● 时至今日, 所有的计算机都没有突破冯·诺依曼机的基本结构。



● 特征

冯诺依曼机的组成: 存储器+运算器+控制器

其主要特征为: 数字计算机的数制采用二进制; 计算机应该按照程序顺序执行。

● 功能

把需要的程序和数据送至计算机中。

必须具有长期记忆程序、数据、中间结果及最终运算结果的能力。

能够完成各种算术、逻辑运算和数据传送等数据加工处理的能力。

能够根据需要控制程序走向, 并能根据指令控制机器的各部件协调操作。

能够按照要求将处理结果输出给用户。

● 基本组成部件

- 1, 输入数据和程序的输入设备
- 2, 记忆程序 and 数据的存储器
- 3, 完成数据加工处理的运算器
- 4, 控制程序执行的控制器
- 5, 输出处理结果的输出设备

计算机的组成部分及功能

由运算器, 控制器, 存储器, 输入装置和输出装置五大部件组成计算机, 每一部件分别按要求执行特定的基本功能。

● 运算器或称算术逻辑单元 (Arithmetical and Logical Unit)

运算器的主要功能是对数据进行各种运算。这些运算除了常规的加、减、乘、除等基本的算术运算之外, 还包括能进行“逻辑判断”的逻辑处理能力, 即“与”、“或”、“非”这样的基本逻辑运算以及数据的比较、移位等操作。

● 存储器 (Memory unit)

存储器的主要功能是存储程序和各种数据信息, 并能在计算机运行过程中高速、自动地完成程序或数据的存取。存储器是具有“记忆”功能的设备, 它用具有两种稳定状态的物理器件来存储信息。这些器件也称为记忆元件。由于记忆元件只有两种稳定状态, 因此在计算机中采用只有两个数码“0”和“1”的二进制来表示数据。记忆元件的两种稳定状态分别表示为“0”和“1”。日常使用的十进制数必须转换成等值的二进制数才能存入存储器中。计算机中处理的各种字符, 例如英文字母、运算符号等, 也要转换成二进制代码才能存储和操作。

存储器是由成千上万个“存储单元”构成的, 每个存储单元存放一定位数(微机为上 8 位)的二进制数, 每个存储单元都有唯一的编号, 称为存储单元的地址。“存储单元”

是基本的存储单位，不同的存储单元是用不同的地址来区分。

计算机采用按地址访问的方式到存储器中存数据和取数据，即在计算机程序中，每当需要访问数据时，要向存储器送去一个地址指出数据的位置，同时发出一个“存放”命令（伴以待存放的数据），或者发出一个“取出”命令。这种按地址存储方式的特点是，只要知道了数据的地址就能直接存取。但也有缺点，即一个数据往往要占用多个存储单元，必须连续存取有关的存储单元才是一个完整的数据。

计算机在计算之前，程序和数据通过输入设备送入存储器，计算机开始工作之后，存储器还要为其它部件提供信息，也要保存中间结果和最终结果。因此，存储器的存数和取数的速度是计算机系统的一个非常重要的性能指标。

● 控制器 (Control Unit)

控制器是整个计算机系统的控制中心，它指挥计算机各部分协调地工作，保证计算机按照预先规定的目标和步骤有条不紊地进行操作及处理。

控制器从存储器中逐条取出指令，分析每条指令规定的是什么操作以及所需数据的存放位置等，然后根据分析的结果向计算机其它部分发出控制信号，统一指挥整个计算机完成指令所规定的操作。因此，计算机自动工作的过程，实际上是自动执行程序的过程，而程序中的每条指令都是由控制器来分析执行的，它是计算机实现“程序控制”的主要部件。

通常把控制器与运算器合称为中央处理器 (Central Processing Unit-CPU)。工业生产中总是采用最先进的超大规模集成电路技术来制造中央处理器，即 CPU 芯片。它是计算机的核心部件。它的性能，主要是工作速度和计算精度，对机器的整体性能有全面的影响。

● 输入设备(Input device)

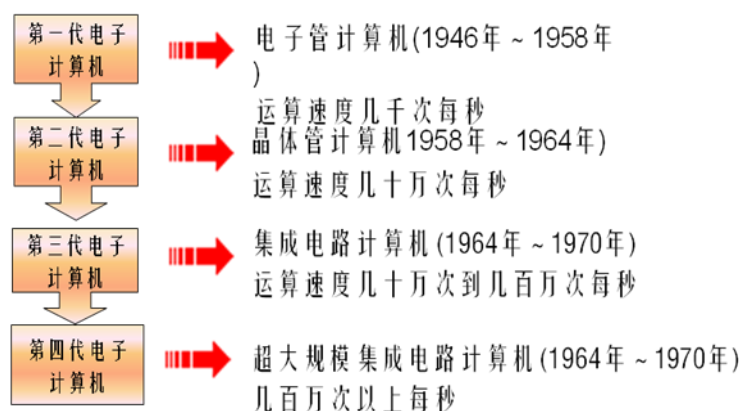
用来向计算机输入各种原始数据和程序的设备叫输入设备。输入设备把各种形式的信息，如数字、文字、图像等转换为数字形式的“编码”，即计算机能够识别的用 1 和 0 表示的二进制代码(实际上是电信号)，并把它们“输入”(INPUT)到计算机内存储起来。键盘是必备的输入设备、常用的输入设备还有鼠标器、图形输入板、视频摄像机等。

● 输出设备(Output device)

从计算机输出各类数据的设备叫做输出设备。输出设备把计算机加工处理的结果（仍然是数字形式的编码）变换为人或其它设备所能接收和识别的信息形式如文字、数字、图形、声音、电压等。常用的输出设备有显示器、打印机、绘图仪等。

通常把输入设备和输出设备合称为 I/O 设备（输入 / 输出设备）

1.2 计算机的发展



(一) 计算机的特点

1. 运算速度快
2. 精确度高
3. 存储容量大
4. 自动化程度高
5. 通用性强

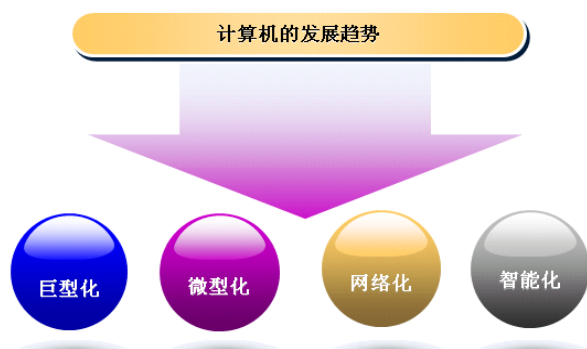
(二) 计算机的类别

1. 微型计算机（微机，Microcomputer）
 - 1) 台式计算机（Desktop）
 - 2) 膝上型电脑（Laptop）/ 笔记本电脑（Notebook）
 - 3) 工作站（Workstation）
 - 4) 掌上型电脑（Palmtop）、个人数字助理（PDA, Personal Digital Assistant）
2. 小型计算机（小型机，Minicomputer）
3. 大型计算机（大型机，Mainframe computer）
4. 超级（巨型）计算机（Supercomputer）
5. 专用计算机（Special-purpose computer）/ 嵌入式计算机（Embedded computer）

(三) 计算机的应用



(四) 计算机发展趋势



第一章 数据的表示、编码和运算

一、 进制（数制）

○ 即进位计数值。就是用进位的方法进行计数。

○ 三要素：

数码:一组用来表示某种数制的符号。

基数: 数制所使用的数码个数称为“基数”或“基”，常用“R”表示，称 R 进制。

位权:指数码在不同位置上的权值。在进位计数制中，处于不同数位的数码代表的数值不同。

○ 不同进制数的表示方法

○ 数制之间的转换

1. 常用的进位计数制

1) 十进制（Decimal System）

由 0、1、2、...、8、9 十个数码组成，即基数为 10。

2) 二进制（Binary System）

由 0、1 两个数码组成，即基数为 2。

3) 八进制（Octal System）

由 0 ...、7 八个数码组成，即基数为 8。

4) 十六进制（Hexadecimal System）

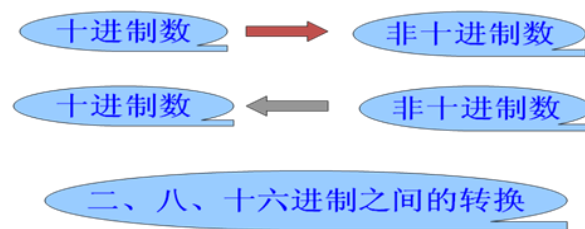
由 0、...、9、A、...、F 十六个数码组成，即基数为 16。

在计算机里，通常用数字后面紧跟一个英文字母来表示该数的数字，
十进制一般用 D，二进制用 B，八进制用 O，十六进用 H 来表示，

十进制、二进制、八进制、十六进制之间的对应关系

十进制	二进制	八进制	十六进制	十进制	二进制	八进制	十六进制
0	0000	0	0	9	1001	11	9
1	0001	1	1	10	1010	12	A
2	0010	2	2	11	1011	13	B
3	0011	3	3	12	1100	14	C
4	0100	4	4	13	1101	15	D
5	0101	5	5	14	1110	16	E
6	0110	6	6	15	1111	17	F
7	0111	7	7	16	10000	20	10
8	1000	10	8	17	10001	21	11

二、进制之间的转换



非十进制数 \rightarrow 十进制数

位权法：把各非十进制数按权展开求和

$$\text{转换公式: } (F)_{10} = a_1 \times x^{n-1} + a_2 \times x^{n-2} + \dots + a_{m-1} \times x^1 + a_m \times x^0 + a_{m+1} \times x^{-1} + \dots$$

示例：

$$\begin{aligned} (1011.1)_2 &= 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} \\ &= 8 + 0 + 2 + 1 + 0.5 \\ &= (11.5)_{10} \end{aligned}$$

十进制整数 \rightarrow 非十进制整数

- 整数部分和小数部分采用不同的方法
- 整数部分采用除基数逆向取余法
- 小数部分采用乘基数正向取整法

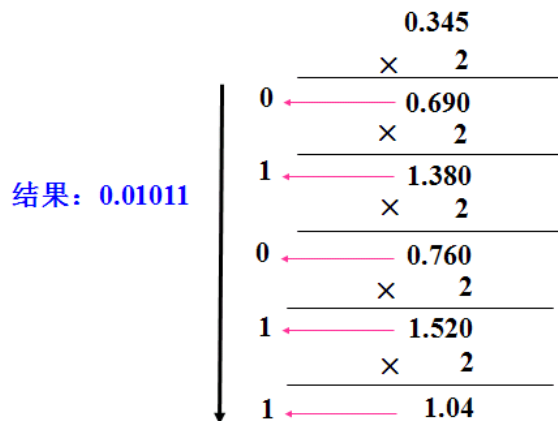
例：100.345 D = () B

整数部分：除基数逆向取余法

整数部分：除基数逆向取余法

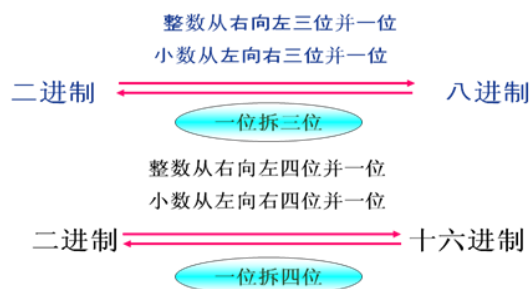
2	100	余数	
2	50	0	↑ 结果：1100100
2	25	0	
2	12	1	
2	6	0	
2	3	0	
2	1	1	
2	0	1	
	0	1	

小数部分：乘基数正向取整法

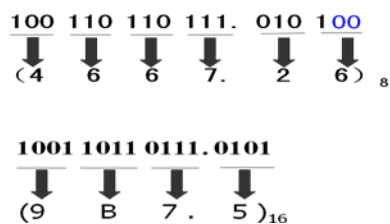


- ◆ 十进制小数并不是都能够用有限位的其他进制数精确地表示,这时应根据精度要求转换到一定的位数为止,此时可以采用0舍1入的方法进行处理(类似于十进制中的四舍五入的方法)作为其近似值。
- ◆ 如果一个十进制数既有整数部分,又有小数部分,则应将整数部分和小数部分分别进行转换

非十进制之间的转换



例: 100110110111.0101B=()O=()H



思考

□ 计算机内部为何采用二进制?

- 易于用物理元件表示:计算机是由逻辑电路组成,而逻辑电路通常只有两个状态。
- 运算规则简单
- 可靠性高:两个状态表示的二进制两个数码,数字传输和处理不容易出错。
- 逻辑性强:计算机工作原理是建立在逻辑运算基础上的,逻辑代数是逻辑运算的理论依据。

三、计算机中数据的单位

□ 位 (b):位是计算机存储信息的最小单位

□ 字节(B):字节是信息处理的基本单位, 一个字节由八位二进制数组成, 即 $1\text{Byte}=8\text{bit}$ 。

○ $1\text{KB}=1024\text{B}=2^{10}\text{B}$

○ 另外还有 MB、GB、TB

➤ 千 字 节 (KB): $1\text{KB}=2^{10}$ 字节=1024B

➤ 兆 字 节 (MB): $1\text{MB}=2^{20}$ 字节=1024KB

➤ 千兆字节 (GB): $1\text{GB}=2^{30}$ 字节=1024MB

➤ 兆兆字节 (TB): $1\text{TB}=2^{40}$ 字节=1024GB

□ 字长:字是 CPU 通过数据总线一次存取、加工和传送数据的长度。一个字通常由一个或若干个字节组成, 字长越长, 性能越强。

常用的字长有 8 位、16 位、32 位、64 位。

四、计算机中的信息表示

1) 机器数的定义

通常, 机器数是把符号"数字化"的数,是数字在计算机中的二进制表示形式。

2) 机器数的特点

机器数有两个基本特点:

一: 数的符号数值化。

实用的数据有正数和负数, 由于计算机内部的硬件只能表示两种物理状态 (用 0 和 1 表示), 因此实用数据的正号“+”或负号“-”, 在机器里就用一位二进制的 0 或 1 来区别。通常这个符号放在二进制数的最高位, 称符号位, 以 0 代表符号“+”, 以 1 代表符号“-”。因为有符号占据一位, 数的形式值就不等于真正的数值, 带符号位的机器数对应的数值称为机器数的真值。例如二进制真值数-011011, 它的机器数为 1011011。

二: 二进制的位数受机器设备的限制。

机器内部设备一次能表示的二进制位数叫机器的字长, 一台机器的字长是固定的。字长 8 位叫一个字节 (Byte), 现在机器字长一般都是字节的整数倍, 如字长 8 位、16 位、32 位、64 位。

3) 机器数的分类

根据小数点位置固定与否, 机器数又可以分为定点数和浮点数。通常, 使用定点数表示整数, 而用浮点数表示实数。整理如下:

(1) 整数。

认为整数没有小数部分, 小数点固定在数的最右边。整数可以分为无符号整数和有符号整数两类。无符号整数的所有二进制位全部用来表示数值的大小; 有符号整数用最高位表示数的正负号, 而其他位表示数值的大小。例如十进制整数-65 的计算机内表示可以是 11000001。

(2) 实数。

实数的浮点数表示方法是: 把一个实数的范围和精度分别用阶码和尾数来表示。在计算机中, 为了提高数据表示精度, 必须唯一地表示小数点的位置, 因此规定浮点数必须写成规范化的形式, 即当尾数不为 0 时, 其绝对值大于或者等于 0.5 且小于 1 (注: 因为是二进制数, 要求尾数的第 1 位必须是 1)。例如设机器字长为 16 位, 尾数为 8 位, 阶码为 6 位, 则二进制实数-1101.010 的机内表示为 0000100111010100。

4) 机器数与真值

不带符号的数是数的绝对值, 在绝对值前加上表示正负的符号就成了符号数。直接用正号“+”和负号“-”来表示其正负的二进制数叫做符号数的真值。在计算机中不仅用 0, 1 编码的形式表示一个数的数值部分, 正、负号亦同样用 0, 1 编码表示。把符号数值

化以后，就能将它用于机器中。我们把一个数在机器内的表示形式称为机器数。而这个数本身就是该机器数的真值。“01101”和“11101”是两个机器数，而它们的真值分别为+1101 和-1101。

5) 整数在计算机中的表示

- 在计算机中，按照既定的二进制位数（称为码长），
 - 最左边的那一位（称为符号位）用来表示一个整数的正负号：0 表示正数，1 表示负数。
 - 符号位之后的那些位（称为数值位），用来表示这个整数的绝对值。
- 在计算机中，数可以有三种不同的二进制表示方法（差别在于负数之数值位的表示不同）：
 - 原码表示
 - 反码表示
 - 补码表示

6) 原码表示

原码：

将数的真值形式中“+”号用“0”表示，“-”号用“1”表示时，叫做数的原码形式，简称原码。若字长为 n 位，原码一般可表示为：

(1) 当 X 为正数时 $[X]$ 原和 X 一样，即 $[X]$ 原 = X 。当 X 为负数时。由于 X 本身为负数，所以，实际上是将 $|X|$ 数值部分绝对值前面的符号位上写成“1”即可。

原码表示法比较直观，它的数值部分就是该数的绝对值，而且与真值、十进制数的转换十分方便。但是它的加减法运算较复杂。当两数相加时，机器要首先判断两数的符号是否相同，如果相同则两数相加，若符号不同，则两数相减。在做减法前，还要判断两数绝对值的大小，然后用大数减去小数，最后再确定差的符号，换言之，用这样一种直接的形式进行加运算时，负数的符号位不能与其数值部分一道参加运算，而必须利用单独的线路确定和的符号位。要实现这些操作，电路就很复杂，这显然是不经济实用的。为了减少设备，解决机器内负数的符号位参加运算的问题，总是将减法运算变成加法运算，也就引进了反码和补码这两种机器数。

- 在给定码长后，根据一个整数的正负填写符号位，再将这个整数之绝对值的二进制表示，按照数值位的长度在前面补足必要的 0 后，就得到这个整数的原码表示。

若码长为 8，则 $123_{(10)}$ 的原码表示是：

01111011

$-123_{(10)}$ 的原码表示是：

11111011

若码长为 16，则 $123_{(10)}$ 的原码表示是：

000000001111011

$-123_{(10)}$ 的原码表示是：

100000001111011

原码中 0 有两种表达方式（+0、-0）

7) 反码表示

如前所述，为了克服原码运算的缺点，采用机器数的反码和补码表示法。若字长为 n 位，反码可表示为：即对正数来说，其反码和原码的形式相同；对负数来说，反码为其原码的数值部分各位变

- 规定：
- 一个正整数的反码表示与其原码表示相同；

- 一个负整数的反码表示: 对其原码表示的数值位进行按位变反(按位将 1 换成 0、将 0 换成 1) 的结果。
- 例如 (若码长为 8):
 $(26)_{(反)} = (26)_{(原)} = 0\ 0011010$
 $(-26)_{(反)} = 11100101 \quad (10011010 \rightarrow 11100101)$
- 0 也有两种反码表示:
 00000000
 11111111

8) 补码表示

补码是根据同余的概念引入的, 我们来看一个减法通过加法来实现的例子。假定现在是在北京时间 6 点整, 有一只手表却是 8 点整, 比北京时间快了 2 小时, 校准的方法有两种, 一种是倒拨 2 小时, 一种是正拨 10 小时。若规定倒拨是做减法, 正拨是做加法, 那么对手表来讲减 2 与加 10 是等价的, 也就是说减 2 可以用加 10 来实现。这是因为 8 加 10 等于 18, 然而手表最大只能指示 12, 当大于 12 时 12 自然丢失, 18 减去 12 就只剩 6 了。这说明减法在一定条件下, 是可以用法来代替的。这里“12”称为“模”, 10 称为“-2”对模 12 的补数。推广到一般则有:

$$A - B = A + (-B + M) = A + (-B)_{补}$$

可见, 在模为 M 的条件下, A 减去 B, 可以用 A 加上 -B 的补数来实现。这里模 (module) 可视为计数器的容量, 对上述手表的例子, 模为 12。

总之, 正数的原码、反码和补码是完全相同的; 负数的原码、反码和补码其形式各不相同。另外, 特别要注意的是, 对于负数的反码和补码 (即符号位为 1 的数), 其符号位后边的几位数表示的并不是此数的数值。如果要想知道此数的大小, 一定要求其反码或补码才行。

- 回到给定码长的二进制表示上来:
 例如, 当码长为 8 (即数值位数为 7), 则

$$26_{(10)} = 0011010$$

那么, 要得到 $-26_{(10)}$, 就是求一个二进制数 c: 使得:

$$c + 0011010 = 0000000$$

这样的 c 就是 $|-26_{(10)}|$ 的二进制表示:

$$1100110$$

因为:

$$\begin{array}{r} 1100110 \\ +) 0011010 \\ \hline 0000000 \end{array}$$

补码表示

- 规定:
 - 一个正整数的补码表示与它的原码表示相同;
 - 一个负整数的补码表示: 符号位为 1, 数值位是其绝对值的求补结果。
- 对于一个负整数, 怎样求它的补码表示?
 - 一条简单规则: 对其原码表示的数值位按位变反后加 1。
 - 例: 当码长为 8, 求 $-26_{(10)}$ 的补码表示 (11100110):
 - 原码表示是: 10011010
 - 按位变反后: 11100101
 - 加 1 后得到: 11100110, 即得到其补码表示。

计算机中为什么使用补码表示数

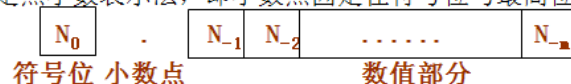
- 使符号位能与有效值部分一起参加运算,从而简化运算规则.
- 使减法运算转换为加法运算,进一步简化计算机中运算器的线路设计

9) 定点数和浮点数

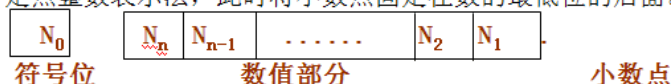
□ 定点表示法:

在机器中, 小数点位置固定的数称为定点数。

1、定点小数表示法, 即小数点固定在符号位与最高位之间。



2、定点整数表示法, 此时将小数点固定在数的最低位的后面。



定点数的运算规则比较简单, 但不适宜对数值范围变化比较大的数据进行运算。

□ 浮点表示法

浮点数可以扩大数的表示范围。浮点数由两部分组成, 一部分用以表示数据的有效位, 称为尾数; 一部分用于表示该数的小数点位置, 称为阶码。一般阶码用整数表示, 尾数大多用小数表示。一个数 N 用浮点数表示可以写成:

$$N = M R^e$$

M 表示尾数, e 表示指数, R 表示基数。基数一般取 2, 8, 16。一旦机器定义好了基数, 就不能再改变了。因此, 在浮点数表示中基数不出现, 是隐含的。

- ❖ 规格化的浮点数: 为了提高浮点数表示的精度通常规定其尾数的最高位必须是非零的有效位, 称为浮点数的规格化形式。

五、字符的编码

1) 字符编码: 目前采用的字符编码主要是 ASCII 码, 它是 American Standard Code for Information Interchange 的缩写 (美国标准信息交换代码), 已被国际标准化组织 ISO 采纳, 作为国际通用的信息交换标准代码。ASCII 码是一种西文机内码, 有 7 位 ASCII 码和 8 位 ASCII 码两种, 7 位 ASCII 码称为标准 ASCII 码, 8 位 ASCII 码称为扩展 ASCII 码。7 位标准 ASCII 码用一个字节 (8 位) 表示一个字符, 并规定其最高位为 0, 实际只用到 7 位, 因此可表示 128 个不同字符。同一个字母的 ASCII 码值小写字母比大写字母大 32。

ASCII 码表完整版

ASCII 值	控制字符	ASCII 值	控制字符	ASCII 值	控制字符	ASCII 值	控制字符
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	HT	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	X	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	TB	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