深入浅出系列规划教材

深入浅出微机原理与接口技术

（第2版）

何 超 主编

清华大学出版社

第1章 概述

1.1 计算机及其基本组成

20世纪最重要的科技成果，莫过于电子计算机的发明。1946年2月14日，世界上第一台电子数字积分计算机ENIAC(Electronic Numerical Integrator and Calculator)诞生在美国宾夕法尼亚大学。

电子计算机的出现在人类社会的各个领域引发了一场新的技术革命，其深远意义不亚于当年蒸汽机的诞生所带来的第一次工业革命。

1.1.1 信息社会和计算机

如今，“信息”二字已成为人们的口头禅，我们已进入信息化社会。

什么是信息化社会？信息化社会就是脱离工业化社会以后，信息起主要作用的，信息技术和信息产业在经济和社会发展中发挥主导作用的社会。与农业社会和工业社会不同，物质和能源不再是主要资源，信息成为更重要的资源，以开发和利用信息资源为目的信息经济活动迅速扩大，逐渐取代工业生产活动而成为国民经济活动的主要内容。换句话说，信息经济在国民经济中占据主导地位，并构成社会信息化的物质基础。

以计算机、微电子和通信技术为主的信息技术革命是社会信息化的动力和源泉。

计算机(Computer)是一种能够预先存储程序，并按照程序自动地、高速地、精确地进行大量数值计算和逻辑运算，处理各类海量数据信息的电子设备。它处理的对象是信息，处理的结果也是信息。从某种意义上说，计算机扩展了人类大脑的功能，因此经常把计算机称为“电脑”。

信息的表现形式是多种多样的。作为一种电子设备，计算机主要的处理对象就是电信号形式的信息。按照所处理的电信号的不同，计算机又可以分为模拟计算机与数字计算机。早期的计算机一般是模拟计算机，处理的电信号在时间上是连续的。现在的计算机大都是数字计算机，处理的电信号在时间上是离散的。同模拟机相比，数字计算机在数据精度、存储量和逻辑判断能力上都强于模拟计算机。随着数字计算机的发展，模拟计算机作为计算工具和通用仿真设备的作用，被数字计算机所取代。但是，作为专用仿真设备、教学与训练工具，模拟计算机还将继续发挥作用，并且也在不断发展中。

信息技术在生产资料、科研教育、医疗保健、企业和政府管理以及家庭中的广泛应用，对经济和社会发展产生了巨大而深刻的影响，从根本上改变了人们的生活方式、行为方式和价值观念。

1.1.2 计算机的分类

按照不同的标准，计算机有多种分类方法。20 世纪 90 年代之前，计算机的分类标准主要有以下几种，如图 1-1 所示。

- 计算机
  - 按原理分类
    - 数字计算机
    - 模拟计算机
    - 混合计算机
  - 按作用分类
    - 通用计算机
    - 专用计算机
  - 按规模分类
    - 巨型机
    - 大型机
    - 中型机
    - 小型机
    - 微型机

图 1-1 计算机分类示意图

但随着时间的推移和计算机技术的发展，分类标准也在发生着变化。现在则一般按照计算机的规模、运算速度、使用范围等综合考虑，可以把计算机分为：高性能计算机、微型计算机(PC)、工作站、服务器、嵌入式计算机 5 类。

1. 高性能计算机

高性能计算机简称 HPC(High Performance Computer)，它之所以被称为高性能计算机，主要是它跟微机和低档 PC 服务器相比而言具有性能、功能方面的巨大优势，主要用于处理数据量大、要求快速、高效的工作的场合。高性能计算机主要用于：

(1) 计算密集型应用，例如大型科学工程计算，数值模拟等，应用领域集中在石油、气象、核能、仿真等行业。

(2) 数据密集型应用，例如数字图书馆、数据仓库、数据挖掘、计算可视化等，应用领域集中在图书馆、银行、证券等行业。

(3) 通信密集型应用，例如协同工作、网格计算、遥控和远程诊断等，应用领域集中在网站、信息中心、搜索引擎、流媒体等行业。

近年来，我国高性能计算机的研制与产业化取得长足发展，以中国科学院、清华大学、国防科技大学、中国科技大学等为代表的国内主要科研单位和高校，已经研制成功了一系列的高性能计算机。

特别值得一提的是，2009 年 10 月 29 日，中国首台千万亿次超级计算机“天河一号”诞生。这台计算机每秒 1206 万亿次的峰值速度和每秒 563.1 万次的 Linpack 实测性能，使中国成为继美国之后世界上第二个能够研制千万亿次超级计算机的国家。并且中国能排进世界 500 强(TOP500)的超级计算机已经从 42 台增加到 62 台。

2011 年 6 月 21 日消息，一台日本计算机取得了 500 强超级计算机排行榜(每年两次，由国际超级计算机会议发布)排名第一的位置。日本 K 计算机每秒浮点运算次数为百万的四次方(8612 万亿次的运算速度)。

K 计算机的性能是使用 68 544 个 SPARC64 VIIIfx 处理器衡量的，每个处理器配置 8 个内核，一共有 548 352 个内核，几乎是其他世界 500 强超级计算机处理器内核数量的两倍。这台超级计算机仍在建造之中。据这台超级计算机的制造商富士通称，当它在 2012 年

11 月开始服务的时候，这台超级计算机将配置 8 万多个 SPARC64 VIIIfx 处理器。

据 Top500.org 称，与其他最近发布的超级计算机不同，K 计算机没有使用图形处理器或者其他加速器。但是，它使用的是这个排行榜中最强大的和最节能的系统。

2. 微型计算机（个人计算机 PC）

微型计算机又称为个人计算机。其种类很多，主要有台式机、笔记本和个人数字助理 PDA 3 种类型。

3. 工作站

工作站（Workstation），是一种以个人计算机和分布式网络计算为基础的一种高档的微型计算机，它可以提供比个人电脑更加强大的性能，具有强大的数据运算与图形、图像处理功能以及联网功能。

工作站主要面向专业应用领域，能够满足工程设计、动画制作、科学研究、软件开发、金融管理、信息服务、模拟仿真等专业领域的工作要求。

常见的工作站有计算机辅助设计（CAD）工作站（或称工程工作站）、办公自动化（OA）工作站和图像处理工作站等。不同任务的工作站有不同的硬件和软件配置。

工作站通常配有高分辨率的大屏幕显示器、容量很大的内存储器和外部存储器、具有小型计算机接口（SCSI）或者光纤路径磁盘存储系统、高端 3D 加速设备、单个或多个 64 位处理器以及设计优秀的冷却系统。另外，还有非常周到的修理/更换计划。

越来越多的计算机厂家在生产和销售各种工作站。

工作站可以工作在网络上，充当客户机或服务器。

4. 服务器

当一台计算机连接到网络时，这台计算机就成为网络的一个客户机。

服务器是一种在网络环境中的高性能计算机，运行网络操作系统，它侦听网络上的众多客户机提交的服务请求，并为这些客户机提供网络资源和服务（含文件服务、数据库服务和图形、图像处理以及打印、通信、安全、保密、系统管理和网络管理等应用程序服务），使其犹如工作站那样地进行操作。

相对于普通 PC 来说，服务器在稳定性、安全性、性能等方面都要求更高，因此它必须有高性能、高稳定性的主机，其 CPU、芯片组、内存、磁盘系统、网络硬件等都要比普通 PC 高一个档次。由于它主要用于后台服务，不一定需要独立的 I/O 设备。

服务器上一般还安装有数据库，便于工作站利用自身装有的应用软件帮服务器分流一些计算工作。

按照应用层次来划分，即按网络规模划分，服务器可以分为入门级服务器、工作组级服务器、部门级服务器和企业级服务器。

服务器的管理和服务有：文件、数据库、图形、图像以及打印、通信、安全、保密、系统管理和网络管理服务。

5. 嵌入式计算机

嵌入式计算机是指作为一个信息处理部件,嵌入到应用系统之中的计算机。它是以专门应用为中心,软硬件可增减的,适应应用系统对功能、可靠性、成本、体积、功耗等综合性严格要求的专用计算机系统。

正因为嵌入式计算机以专门应用为中心,它与通用型计算机最大的区别是,运行固化的软件(用户很难改变),通常制作成单片机或单板机的形式。嵌入式计算机应用最广泛,数量超过微型机。目前,广泛应用于各种家用电器之中,如电冰箱、数字电视机、数码照相机等。

1.1.3 计算机的基本组成

计算机最早的也是最基本的功能,是计算。欲制造代替人工作的计算机,应从分析人工计算过程出发,来"确定计算机的基本组成"和"怎样教会计算机"工作。

例如,我们接到一个除法题目,要先知道"被除数"和"除数",并把它们记录在纸上,接着就要按照已经学过的运算法则和步骤去计算,最后还要把计算结果的"商"和"余数"记录在纸上,出示给"提出任务的"人。这个过程中,起主要作用的,是人的"清醒的"大脑,它是完成计算任务的主体,是"运算器";其次,"给出题目"、"运算法则和步骤"、"记录"、"输出结果"等,是"一系列的指令";纸、记忆"运算法则和步骤"的大脑皮层,是"存储器";"提出任务的"人,是"输入设备",手是"输出设备"。整个过程,必须由"大脑"这个"控制器"指挥,而"一系列"指令就是"软件",即程序。

可见,尽管欲代替人工作的真实的计算机十分复杂,但计算机的基本组成,但总可归结为两大部分:硬件系统和软件系统。

硬件系统是完成计算工作的物理实体,由运算器、控制器、存储器、总线、输入设备和输出设备5部分组成,各部分起着不同的作用。

运算器是直接完成各种算术、逻辑运算的装置。运算器的主要技术指标是字长和运算速度,字长即能处理的数据长度,这决定计算精度,速度决定运算快慢。

控制器的任务是协调各部分硬件的工作,使计算机有序正确地完成各项作业。

软件系统是指导计算机工作的所有程序、数据和相关文件的集合,是计算机系统不可缺少的组成部分。微型计算机的软件系统包括系统软件和应用软件两部分。

系统软件是为了计算机能正常、高效工作所配备的各种管理、服务、监控和维护系统的程序及有关资料。系统程序软件的主要任务:一是更好地发挥计算机的效率;二是方便用户使用计算机。这就好比行政单位的内部管理的各项规章制度。系统软件的作用可以概括为两个接口:一是"人"与"机"的接口,二是"硬件"和"软件"的接口。系统软件包括3大部分:操作系统、编译系统和网络系统。

应用软件就是用户为解决各种实际问题而编写的计算机应用程序及有关资料。如办公软件、图形软件、计算机辅助设计软件……这就好比到银行存款取款,有一套具体事项、操作的顺序和注意事项。

1.1.4　微型计算机的硬件系统和软件系统

1. 微型计算机硬件系统的基本组成

微型计算机自然应有计算机同样的基本硬件组成，只不过要实现微型化。微型计算机的基本组成如下。

1）中央处理器

中央处理器（CPU）由运算器、控制器，再加上用于暂时存放“当前运算的中间结果和相关数据”的寄存器组组成，三者集成在一块芯片上，也称为“微处理器”。

CPU 技术是微型计算机的关键技术之一。在微型机不断向超轻、超薄方向发展的今天，要求 CPU 在保持高性能和高速度的同时，还要在设计上考虑低耗电、低发热等因素。

2）存储器

存储器在计算机中占有非常重要的地位，它是计算机存放信息的场所。计算机当前正在执行的程序和处理的数据都是存放在存储器中的。存储器分内存储器和外存储器两种，简称内存和外存。

内存和 CPU 组成计算机的主机，换句话说，内存和 CPU 是构成计算机的基本成分。

内存用来存储当前正在使用的或者要经常使用的程序和数据，因此，内存也称主存，CPU 在工作过程中要频繁地与主存交换信息，可以直接对它进行读、写。外存是在主机的外部，存放的信息相对 CPU 来说是不经常使用的信息。如果 CPU 要使用这些信息，必须通过专门的设备将信息先调入到内存中。

存储器按其工作特点，可以分为只读存储器（ROM）和随机存储器（RAM）两种。对于微型计算机而言，常见的内存有：RAM、EDO DRAM、SRAM、DDR RAM、RDRAM 等类型（详见第 6 章）。

3）输入输出设备

输入输出设备（I/O 设备）是计算机与人进行交流的平台，也被称为外部设备。输入设备将计算机程序、原始数据以计算机能识别的形式输入到计算机中；输出设备把计算机处理后的结果以人能识别的形式表示出来。常见的输入输出设备有显示器、鼠标、键盘、扫描仪和打印机等。

4）总线与接口

CPU 与每一个内部芯片及外部设备的连接和数据的交换都必须通过各种总线与 I/O 接口来实现，因此接口技术也是微型计算机的关键技术之一。

2. 微型计算机硬件系统的硬件结构

实际的微型计算机元件较多，为了追求体积轻薄、散热性强、性能稳定等目标，就必须想方设法把诸多元件整合在一个小小的空间内，于是一个叫做“主板”的器件应运而生。

主板实际上是一块装有各种插槽和接口，并带有连接导线的电路板。板上有CPU插槽、内存插槽、各种外设的功能卡（如显卡、声卡等）插槽，总线扩展插槽（ISA，PCI等扩展槽），以及用来与键盘、鼠标、软盘驱动器、硬盘驱动器、光盘驱动器等设备相连的接口电路（简称接口）；还有为了方便扩展外设而安排的串行接口（如COM1，COM2）、并行接口（如打印机接口LPT1）等。

因此微型计算机也可以看成以主板为中心，把相应的外设连接起来构成的系统。

为了简洁和接线方便，主板上的连接导线做成各种不同的系统总线，比较常用的有：ISA总线、PCI总线和AGP总线等。

自从引入PCI总线以后，总线控制与转换部件——芯片组（即南北桥）成了构成主板的核心组成部分，它是CPU与周边设备连接的桥梁。

主板装在主机箱内。其上已插入了CPU、内存、芯片组，和显卡、声卡、网卡等功能卡。除主板外，电源、光驱、硬盘等都装在主机箱内。

可以说，一台通用微机最基本的部件就是主机箱；再加上一些不方便装在箱内的外部设备（包括鼠标、键盘、显示器、音箱、打印机和扫描仪等）。

3. 微型计算机的软件系统

如前所述，同所有的计算机系统一样，微型计算机的软件系统包括系统软件和应用软件两部分。

随着CPU、主板等硬件技术，软件开发技术的快速发展，微型计算机的组成部分也在不断变换、更新中。微型计算机的硬件系统和软件系统示意图见图1-2所示。

<table>
<tr><td rowspan="13">微型计算机系统</td><td rowspan="9">硬件系统</td><td rowspan="7">主机箱</td><td rowspan="6">主板</td><td rowspan="3">CPU</td><td>运算器</td></tr>
<tr><td>控制器</td></tr>
<tr><td>寄存器组</td></tr>
<tr><td colspan="2">内存、功能卡</td></tr>
<tr><td colspan="2">输入/输出接口电路</td></tr>
<tr><td colspan="2">系统总线（含芯片组）</td></tr>
<tr><td colspan="3">电源、光驱、硬盘等</td></tr>
<tr><td colspan="3" rowspan="2">外部设备</td><td>输入设备（如键盘、鼠标等）</td></tr>
<tr><td>输出设备（显示器、打印机等）</td></tr>
<tr><td rowspan="4">软件系统</td><td colspan="3" rowspan="3">系统软件</td><td>操作系统</td></tr>
<tr><td>编译系统</td></tr>
<tr><td>网络系统</td></tr>
<tr><td colspan="4">应用软件</td></tr>
</table>

图1-2 微型计算机系统的组成

1.2 进位计数制

1.2.1 数制

1. 数制定义

数制，也称计数制，是指用一组固定的符号和统一的规则来表示数值的方法。按进位的方法进行计数，称为进位计数制。在日常生活和计算机中采用的是进位计数制。

日常生活中，人们通常以十进制来进行计算；也用六十进制计算时间，即 60 分钟为一个小时；用十二进制来计算年度，即 12 个月为一年。

但计算机内部一般采用二进制，这是因为：

(1) 技术实现简单，计算机是由逻辑电路组成的，逻辑电路通常只有两个状态，开关的接通与断开，这两种状态正好可以用“1”和“0”表示，或者反之；

(2) 运算规则简单，有利于简化计算机内部结构，提高运算速度；

(3) 适合逻辑运算，逻辑代数是逻辑运算的理论依据，二进制只有两个数码，正好与逻辑代数中的“真”和“假”相吻合；

(4) 易于与其他进制互相转换；

(5) 因为二进制数的每位数据只有 0 和 1 两个状态，当受到一定程度的干扰时，不易改变。表明二进制数据具有抗干扰能力强、可靠性高的优点。

通常在数制中，都涉及 4 个概念：数位、数码、基数和位权(权重)。

(1) 数位：指数码在一个数中的位置。如十进制的个位、十位等。

(2) 数码：一个数制中表示基本数值大小的不同数字符号。如八进制有 8 个数码：0、1、2、3、4、5、6、7。

(3) 基数：某种数制中所拥有基本数码的个数。如十进制的基数为 10，二进制的基数为 2。运算规则为逢“基数”进一(加法运算)，借一当“基数”(减法运算)。

(4) 位权：一个数值中某一位上的 1 所表示数值的大小。如十进制数 128，1 的位权是 100(10^2)，2 的位权是 10(10^1)，8 的位权是 1(10^0)，所以十进制数中千位、百位、十位、个位上的权可表示为 10^3、10^2、10^1、10^0。

2. 常用的进位计数制

(1) 十进制(Decimal Notation)。十进制数 X 一般简记为$(X)_{10}$或 XD。

(2) 二进制(Binary Notation)。二进制数 X 一般简记为$(X)_2$或 XB。

(3) 八进制(Octal Notation)。八进制数 X 一般简记为$(X)_8$或 XQ。

(4) 十六进制(Hexadecimal Notation)。十六进制数用 16 个数码(0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F)记数，基数为 16，权为 16^n(n 为数位)，运算规则为逢 16 进一(加法运算)，借一当 16(减法运算)。十六进制数 X 一般简记为$(X)_{16}$或 XH。在 16 个数码中，A、B、C、D、E、F 分别对应十进制数中的 10、11、12、13、14、15。

知道了数位、数码、基数和位权这 4 个概念后，就可以用一组有序数码，或者以位权展

开多项式的求和形式来表示一个数据。即:

$$N_R = K_{n-1} \times R^{n-1} + K_{n-2} \times R^{n-2} + \cdots + K_1 \times R^1 + K_0 \times R^0 + K_{-1} \times R^{-1} + K_{-2} \times R^{-2} + \cdots + K_{-m} \times R^{-m} = \sum_{i=n-1}^{-m} K_i \times R^i$$

式中:R 表示基数,K_i 表示数码,i 表示 K_i 的位序号,R^i 表示位权。

例如:$(200)_{10} = 200D = 2\times10^2 + 0\times10^1 + 0\times10^0$

$(520.919)_{10} = 520.919D$
$= 5\times10^2 + 2\times10^1 + 0\times10^0 + 9\times10^{-1} + 1\times10^{-2} + 9\times10^{-3}$

$(11001.01)_2 = 11001.01B$
$= 1\times2^4 + 1\times2^3 + 0\times2^2 + 0\times2^1 + 1\times2^0 + 0\times2^{-1} + 1\times2^{-2}$

$(115.23)_8 = 115.23Q = 1\times8^2 + 1\times8^1 + 5\times8^0 + 2\times8^{-1} + 3\times8^{-2}$

$(8AC)_{16} = 8ACH = 8\times16^2 + A\times16^1 + C\times16^0$

$(6D.3E)_{16} = 6D.3EH = 6\times16^1 + D\times16^0 + 3\times16^{-1} + E\times16^{-2}$

4 种进制间的等值对照关系如表 1-1 所示。

表 1-1 四种进制之间的等值对照关系

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	8	1000	10	8
1	0001	1	1	9	1001	11	9
2	0010	2	2	10	1010	12	A
3	0011	3	3	11	1011	13	B
4	0100	4	4	12	1100	14	C
5	0101	5	5	13	1101	15	D
6	0110	6	6	14	1110	16	E
7	0111	7	7	15	1111	17	F

但在新一代计算机中,例如生物计算机(DNA 计算机),已开始考虑使用四进制数制。四进制是以 0、1、2、3 为基数,采用逢四进一的计算原则,它与其他数制之间的转换可以参考八进制、十六进制的转换方法。据说,采用四进制最大的好处是能节省一半的运算单元,并能提高系统的整体运算速度。例如某台计算机需要 20 万个运算单元,在采用了四进制后,只需 10 万个运算单元就能得到相同的效果。

3. 计算机中数的运算与存储单位

现在的计算机大多是二进制的计算机,运算与存储二进制数据经常使用到以下几个单位。

1) 位

计算机存储数据的最小单位,表示一个二进制位,简记为 bit(中文叫"比特")。计算

机中最直接、最基本的操作就是对二进制位的操作。每一位只能表示0或1两种状态。

2）字节

8位二进制数字，构成一个字节(Byte)，简称B，即1B=8bit。由于常用的英文字符用8位二进制就可以表示，所以通常就将8位称为一个字节。一般情况下，一个汉字国标码占用两个字节。

3）字

在计算机技术中，通常把CPU能进行一次最基本的运算的二进制数的位数叫字长，该二进制数，称为字(Word)。一个字由若干字节组成。例如一个字由两个字节组成，则该字字长为16位。它标志着CPU的计算精度和计算速率。字长越长，CPU的数据处理能力(精度越高，寻址空间越大，同一时间内传送信息量越大，计算速率越大，指令越丰富)越强。CPU按照其处理信息的字长可以分为：8位微处理器、16位微处理器、32位微处理器以及64位微处理器等。

后面将会看到，CPU的字长决定于其数据总线宽度。

1.2.2 数在不同进制之间的转换

实际运算中常常涉及数在不同进制之间的相互转换，现在讨论这些转换应遵循的转换原则。

1. *N*(*N*=2、8、16)进制数转换成十进制数

将一个N进制数转换成十进制数比较容易。其转换规则为：依照上述的位权展开法，将N进制数以多项式形式展开，然后对数位上的数进行求和运算。

例如：把二进制数$(1101.01)_2$、$(46.23)_8$、$(5B.6A)_{16}$分别转换成十进制数。可按以下方式展开为每个N进制数的数字乘以N的相应次幂，然后相加即可。

$$
\begin{aligned}
(1101.01)_2 &= 1\times2^3+1\times2^2+0\times2^1+1\times2^0+0\times2^{-1}+1\times2^{-2}\\
&= 1\times8+1\times4+0\times2+1\times1+0\times0.5+1\times0.25\\
&= (13.25)_{10}
\end{aligned}
$$

$$
\begin{aligned}
(46.23)_8 &= 4\times8^1+6\times8^0+2\times8^{-1}+3\times8^{-2}\\
&= 4\times8+6\times1+2\times0.125+3\times0.016\\
&= (38.297)_{10}
\end{aligned}
$$

$$
\begin{aligned}
(5B.6A)_{16} &= 5\times16^1+B\times16^0+6\times16^{-1}+A\times16^{-2}\\
&= 5\times16+B\times1+6\times0.0625+A\times0.004\\
&= (91.415)_{10}
\end{aligned}
$$

2. 十进制数转换成*N*进制数

十进制数转换成N进制数较为复杂，一般需要将十进制数的整数部分与小数部分分开进行转换。

其转换规则如下。

整数部分按照"除基逆序取余"法,即将十进制整数部分除以基数 N,取出商的余数部分作为"转换结果的整数部分"的最低位,再将所得商除以 N。重复操作,直至商为0为止。将所得的各步余数逆序排列就是转换结果的整数部分。

小数部分按照"乘 N 顺序取整"法,即将十进制小数部分乘以 N,取出积的整数部分作为"转换结果的小数部分"的最高位,重复操作,直到积的小数部分为0或达到精度要求的位数为止。将各次所得整数顺序排列即为转换结果的小数部分。

例如:把十进制数 $(197.85)_{10}$ 分别转换成二进制数、八进制数和十六进制数。对该数的整数和小数部分分别进行除以 N 和乘以 N 的运算。

(1) 转换成二进制数。

整数部分:

2 | 197		
2 | 98	……余 1	低
2 | 49	……余 0	↑
2 | 24	……余 1	
2 | 12	……余 0	
2 | 6	……余 0	
2 | 3	……余 0	
2 | 1	……余 1	
0	……余 1	高

小数部分:

$0.85\times2=1.7$ 取整数 1 高

$0.7\times2=1.4$ 取整数 1 ↓

$0.4\times2=0.8$ 取整数 0

$0.8\times2=1.6$ 取整数 1 低

……

组合可得,$(197.85)_{10}=(11000101.1101)_2$

注意:在小数部分的转换中,可能会出现小数部分不能为0的情况,即转换过程无限,此时,我们可按要求保留一定精度的小数即可。

(2) 转换成八进制数。

整数部分:

8 | 197		
8 | 24	……余 5	低
8 | 3	……余 0	↑
0	……余 3	高

小数部分:

$0.85\times8=6.8$ 取整数 6 高

$0.8\times8=6.4$ 取整数 6 ↓

$0.4\times8=3.2$ 取整数 3 低

……

组合可得，$(197.85)_{10}=(305.663)_8$

(3) 转换成十六进制数。

整数部分：

16 | 197			
16 | 12	……余 5	低	
0	……余 C	高	

小数部分：

$0.85\times16=13.6$	取整数 D	高
$0.6\times16=9.6$	取整数 9	低
……		

组合可得，$(197.85)_{10}=(\mathrm{C5.D9})_{16}$

3. 八进制数、十六进制数与二进制数之间的转换

二进制数与八进制数和十六进制数存在一种对应关系，这是因为 $2^3=8,2^4=16$。所以它们之间的转换规则为：每 1 位八进制数对应于 3 位二进制数，每 1 位十六进制数对应于 4 位二进制数。

1) 二进制数与八进制数之间的转换

二进制数转换成八进制数方法：以小数点为界，分别向左向右，每 3 位二进制数字转换为 1 位八进制数字，不够 3 位用 0 补齐，然后按顺序连接得出转换结果。

例如：把 $(10111011.10101)_2$ 转换成八进制数：

转换成八进制数					
分组：	10	111	011.	101	01
补 0：	**0** 10	111	011.	101	01 **0**
转换：	2	7	3.	5	2

得出 $(10111011.10101)_2=(273.52)_8$

反过来操作，容易实现八进制数字转换成二进制数。

如：把 $(265.31)_8$ 转换成二进制数：

转换成二进制数					
分组：	2	6	5.	3	1
转换：	010	110	101.	011	001

得出 $(265.31)_8=(10110101.011001)_2$

2) 二进制数与十六进制数之间的转换

二进制数转换成十六进制数方法：以小数点为界，分别向左向右，每 4 位二进制数字转换为 1 位十六进制数字，不够 4 位用 0 补齐，然后按顺序连接得出转换结果。

例如：把 $(1011110110110.111001)_2$ 转换成十六进制数：

转换成十六进制数						
分组：	1	0111	1011	0110 .	1110	01
补 0：	**000** 1	0111	1011	0110 .	1110	01 **00**
转换：	1	7	B	6 .	D	4

得出 $(1011110110110.111001)_2=(\mathrm{17B6.D4})_{16}$

反过来操作，容易实现十六进制数转换成二进制数。

如：把$(B91.97)_{16}$转换成二进制数。

转换成二进制数　分组：B　9　1　.　9　7

转换：1011　1001　0001.1001　0111

得出$(B91.97)_{16}=(101110010001.10010111)_2$

如果需要完成八进制数与十六进制数之间的转换，可以先把八进制数转换成十进制数，再转换成十六进制数，或者把八进制数转换成二进制数，再转换成十六进制数。反之亦然。

1.3 微型计算机中数的编码和字符的表示

数据、信息处理是计算机的一个重要应用。人们用各种方法来表示数据、反映信息，例如文字、图表、数字都可以用来记录信息。

对于数字计算机而言，所接收的信息都要先转化为数据。数据既有数值数据也有非数值数据。数值数据用来表示数量意义，非数值数据表示文字、符号、图形、语言等。虽然二者表示的含义不一样，但在计算机中都是采用不同进制形式的编码来表示它们。

计算机中常用的编码有很多，不同的编码，其编码规则不同，具有不同的特性及应用场合。

1.3.1 二进制数值数据的编码

1. 真值数和机器数

真值数是未经计算机编码的二进制原始数据。如：正数+0101011和负数-1001101。

计算机只能用0和1来表示数据，怎样区分正数和负数呢？又怎样区分无符号数和带符号数呢？

对于带符号数，为了区分正数和负数，把一个数的最高位定义为数的符号位，称为数符。数符值为“0”表示该数为正，数符值为“1”表示该数为负；而这个数的其他位称为数值位，如图1-3所示。

$$\underbrace{1}_{\text{数符}}\ \underbrace{1\ 0\ 0\ 1\ 1\ 0\ 1}_{\text{数值位}}$$

图1-3　8位二进制表示一个数

通常，把这种在计算机中使用且符号位被正负化了的数称为机器数。

用机器数的最高位代表符号(0为正，1为负)，其数值位为真值数的绝对值。

例如：若真值数分别为：-1001101　　+0101011

则机器数分别为：11001101　　0 0101011

在计算机中，也使用无符号数，此时，不设符号位，故无符号数的机器数和真值相同。n位二进制无符号整数的可表示的数据范围是：$0\sim(2^n-1)$；n位二进制有符号整数的可

表示的数据范围是：$(-2^{n-1}-1)\sim(2^{n-1}-1)$。

例如，$n=8$ 时，无符号整数的范围：0～255，有符号整数的范围：－128～127。

$n=16$ 时，无符号整数的范围：0～65535，有符号整数的范围：－32768～32767。

在计算机中，常用无符号数表示存储器的地址。

计算机无法识别有符号数还是无符号数，决定者是用户。用户使用计算机时，要采用不一样的指令，分别处理有符号数与无符号数。

2. 机器数的原码、反码和补码、移码

在机器数中，符号位和数值位都由 0 和 1 表示，因此，在存储这些数据的时候必须考虑符号位的处理，这种处理方法就是符号位和数值位的编码方法。根据符号位和数值位的编码方式不同，机器数分为原码、补码和反码。其构成方法如下（注意正数和负数的不同）：

1）正数

若一个机器数是正数，则它的原码、补码和反码都是它本身，且符号位为 0。

例如：假设某机器为 8 位机，即一个数据用 8 位（二进制）来表示，＋28 的原码为 00011100，则它的原码、补码和反码都是它本身 00011100，其中最高位是符号位，后 7 位是数值位。

2）负数

若一个机器数是负数，则它的原码是它本身，符号位为 1；反码则保持符号位不变，其余数值位的数逐位取反。补码则是将它的原码除符号位以外，数值位各位取反（即 0 变为 1，1 变为 0），最后在末位加 1，即在反码的末位加 1。

例如－28 的原码为 10011100。

－28 的补码为 11100100（首位不动，其余逐位取反，得到 11100011，在末尾加 1，得 11100100）。

－28 的反码为 11100011。

显然原码较为简单直观，就是在讨论真值数和机器数中讲过的带符号数的表示方法，即：最高位为符号位，用 0 表示正数，1 表示负数，数值部分用二进制数的绝对值形式表示。

原码易于在机器数和真值数之间转换。但如果对原码进行加、减法运算时，则比较繁琐，为了方便加、减法运算，可以用补码表示数据。

补码表示数据，可以避免符号判断的问题，简化加、减法的运算。在计算机中，有符号数一般是用补码形式表示的。引入补码的意义在于可把加减法都统一到加法上来，从而使运算器中根本无须设计减法器，简化了运算器的结构。

至于移码，其构成方法是：移码：除了符号位与补码相反以外，其他都相同。

3. 原码、反码和补码、移码的数学理论

本节讨论中所举例子，均假设是 8 位机，$n=8$。

1）原码的数学定义

若记定点整数（见 1.3.3）的 n 位二进制形式为 $X_{n-1}X_{n-2}\cdots X_1X_0$（$X_{n-1}$ 为符号位，0

表示正数,1表示负数),则其原码的数学定义是:

$$[X]_{原}=\begin{cases}X & 0\leqslant X\leqslant 2^{n-1}-1\\ 2^{n-1}-X=2^{n-1}+|X| & -(2^{n-1}-1)\leqslant X\leqslant 0\end{cases}$$

其中 X 为真值,而 n 为整数的位数。

原码的表示范围:

$$-(2^{n-1}-1)\sim 2^{n-1}-1$$

实例验证:有真数 $X=+11011$,$Y=-11011$,当 $n=8$ 时,高位不足部分添0,补足到机器数的位数。由原码构造方法。

$$[X]_{原}=00011011=X;$$

而

$$\begin{aligned}[Y]_{原}&=10011011=10000000+0011011\\&=10000000-(-0011011)\\&=2^7-Y=2^7+|Y|\end{aligned}$$

若记定点小数(见1.3.3节)的二进制形式为 $X_0.X_{-1}X_{-2}\cdots X_{-n}$($X_0$ 为符号位,0表示正数,1表示负数),则其原码的数学定义是:

$$[X]_{原}=\begin{cases}X & 0\leqslant X<1\\ 1-X=1+|X| & -1<X\leqslant 0\end{cases}$$

其中 X 为真值,而 n 为小数的位数。

实例验证:有真数 $X=+0.11011$,$Y=-0.11011$,低位不足部分添0补足到机器数的位数。则

$$[X]_{原}=0.1101100=X$$

$$[Y]_{原}=1.1101100=1+0.1101100=1+|Y|=1-Y$$

注意:0的原码有$[+0]_{原}$$[-0]_{原}$之分。

$$[+0]_{原}=0.0\cdots0;$$

$$[-0]_{原}=1.0\cdots0。$$

另外,由于符号位被数值化,如果按照纯粹的二进制代码形式,会造成原码表示的负数大于正数的误会,如:011011<111011;0.1101100<1.1101100。

2) 反码的数学定义

若记定点整数(见1.3.3节)的二进制形式为 $X_{n-1}X_{n-2}\cdots X_1X_0$($X_{n-1}$ 为符号位,0表示正数,1表示负数),则其反码的数学定义是:

$$[X]_{反}=\begin{cases}X & 0\leqslant X<2^{n-1}-1\\ 2^n-1+X & -(2^{n-1}-1)<X\leqslant 0\end{cases}$$

其中 X 为真值,而 n 为整数的位数。

反码的表示范围:

$$-(2^{n-1}-1)\sim 2^{n-1}-1$$

实例验证:有 $X=+11011$,$Y=-11011$,高位不足部分添0补足到机器数的位数(此处设是8位机)。

则

$$[X]_{反} = 00011011 = X$$
$$[Y]_{反} = 11100100 = 100000000 - 1 + (-0011011) = 2^8 - 1 + Y$$

若定点小数(见 1.3.3 节)的二进制原码形式为 $X_0.X_{-1}X_{-2}\cdots X_{-n}$($X_0$ 为符号位,0 表示正数,1 表示负数),则其反码的数学定义是:

$$[X]_{反} = \begin{cases} X & 0 \leqslant X < 1 \\ 2 - 2^{-n} + X & -1 < X \leqslant 0 \end{cases}$$

其中 X 为真值,而 n 为小数的位数。

实例验证:有 $X=+0.11011$,$Y=-0.11011$,低位不足部分添 0 补足到机器数的位数(此处设是 8 位机)。

则

$$\begin{aligned} [X]_{反} &= 0.1101100 = X \\ [Y]_{反} &= 1.0010011 = 1.0000000 + 0.0010011 \\ &= 1.0000000 + (1.0000000 - 0.1101101) \\ &= 10.0000000 - \mathbf{0.00000001} - 0.1101101 \\ &= 2 - 2^{-8} + Y \end{aligned}$$

注意:0 的反码有$[+0]_{反}$,$[-0]_{反}$之分。

$$[+0]_{反} = 0.0\cdots0;$$
$$[-0]_{反} = 1.1\cdots1。$$

另外,由于符号位被数值化,如果按照纯粹的二进制代码形式,会造成反码表示的负数大于正数的误会,如:011011<100100;0.1101100<1.0010011。

3) 补码的数学定义

若定点整数(见 1.3.3 节)的二进制形式为 $X_{n-1}X_{n-2}\cdots X_1X_0$($X_{n-1}$ 为符号位,0 表示正数,1 表示负数),则其补码的数学定义是:

$$[X]_{补} = \begin{cases} X & 0 \leqslant X < 2^{n-1} - 1 \\ 2^n + X & -2^{n-1} \leqslant X < 0 \end{cases}$$

其中 X 为真值,而 n 为整数的位数。

补码的表示范围:

$$-2^{n-1} \sim 2^{n-1} - 1$$

实例验证:有 $X=+11011$,$Y=-11011$,高位不足部分添 0 补足到机器数的位数(此处设是 8 位机)。

则

$$\begin{aligned} [X]_{补} &= 00011011, \\ [Y]_{补} &= 11100101 = 10000000 + 01100101 \\ &= 1'00000000 + (-00011011) = 2^8 + Y \end{aligned}$$

若定点小数(见 1.3.3 节)的二进制形式为 $X_0.X_{-1}X_{-2}\cdots X_{-n}$($X_0$ 为符号位,0 表示正数,1 表示负数),则其补码的数学定义是:

$$[X]_{补} = \begin{cases} X & 0 \leqslant X < 1 \\ 2+X & -1 < X \leqslant 0 \end{cases}$$

其中 X 为真值。

实例验证：有 $X=+0.11011$，$Y=-0.11011$，低位不足部分添 0 补足到机器数的位数(此处设是 8 位机)。

则

$$[X]_{补} = 0.1101100$$

$$[Y]_{补} = 1.0010100 = 10.0000000 + (-0.1101100) = 2+Y$$

注意：0 的补码只有一个。$[0]_{补码}=[+0]_{补码}=[-0]_{补码}=0.000\cdots0$

另外，对于 8 位二进制数码，因为

$$[-1]_{补}+[-127]_{补}=1111\ 1111+1000\ 0001=1000\ 0000。$$

上式中，最高位有进位 1，限于位数，舍去。

故可规定 $[-128]_{补}=1000\ 0000$，使 8 位二进制数的补码表示的整数范围比原码和反码都要多一个 -2^7；推而广之，对于 n 位二进制数的补码，可表示的整数范围比原码和反码都要多一个 -2^{n-1}，并规定在形式上，有 $[-2^{n-1}]_{补}=2^{n-1}$。

4) 移码的数学定义

移码用于浮点数的指数部分，只有整数形式。

若定点整数(见 1.3.3 节)的二进制原码形式为 $X_{n-1}X_{n-2}\cdots X_1X_0$($X_{n-1}$ 为符号位，0 表示正数，1 表示负数)，则其移码的数学定义是：

$$[X]_{移} = 2^n + X \quad -2^n \leqslant X < 2^n$$

其中 X 为真值，而 n 为整数的位数。例如，$X=+11011$，$Y=-11011$，则 $[X]_{移}=111011$，$[Y]_{移}=000101$。

移码的表示范围：

$$-2^n \sim 2^n-1$$

即移码与补码只差一个符号位，即正数的移码符号位为 1；负数的移码符号位为 0，其他都相同。

注意：

- 按上述编码方法，数值 0 的原码，反码或补码，比较特殊：

  $$[0]_{原码} = 0.000\cdots0$$

  $$[-0]_{原码} = 1.000\cdots0$$

  0 的补码只有一个。$[0]_{补码}=[+0]_{补码}=[-0]_{补码}=0.000\cdots0$

  0 的移码只有一个。$[0]_{移码}=[+0]_{移码}=[-0]_{移码}=1.0\cdots0$。

  这里的点，不代表小数点，而是表示其左边是符号位。

- 反码的反码为原码；补码的补码也为原码。即：

  $$[[N]_{反码}]_{反码} = [N]_{原码} \quad [[N]_{补码}]_{补码} = [N]_{原码}$$

  后一式不能用于求 $[-2^{n-1}]_{原}$，因为 -2^{n-1} 超出了原码和反码的定义范围，故 $[-2^{n-1}]_{原} \neq [[-2^{n-1}]_{补}]_{补}$，例如 $[[-2^7]_{补}]_{补} \neq -2^7$。