微型计算机及其在电牵引中的应用

李　治　吴守箴　主编

西南交通大学出版社

前　言

本书是依据铁道部电力牵引与传动控制和铁道电气化专业教学指导委员会1985年南昌会议制定的1985—1990年教材编写规划而编写的。

该教材由西南交通大学李治和上海铁道学院吴守箴担任主编，西南交通大学贺威俊担任主审，西南交通大学于万聚和华东交通大学叶雳参加了该书的编写工作。编审者是依据“微型计算机及其在电牵引中的应用”编审小组1986年上海会议审定的编写大纲进行编写和审阅的。

根据编者多年教学经验一致认为，该书若以一种典型的微处理器为主线，作深入、系统的阐述，会比将各种处理器面面俱到的介绍方法收到更好的教学效果，这样可以达到举一反三的目的。因此本书选定了目前国内比较流行的Z80微处理器系列作为主要对象，力求讲深讲透，选材尽可能完整、详尽、实用和系统。

本书力求深入浅出，注重实用，每一部分都列举了一定数量的电路及编程实例，并在第十章专门介绍了微型计算机在电牵引中应用的许多实例。对各种大规模集成电路的内部结构只作简单介绍，而偏重于阐述这些电路的实际用法，特别强调了软件怎样通过硬件起作用，这对系统扩展和接口电路设计是很有用的。书中引用了许多程序段和接口电路，一部分取材于国内外资料和教科书，一部分则直接从我们近年来教学、科研的成果中提取。这些程序和电路均经过实际考验，可以作为读者应用中的借鉴。

书中的第一、二、三章由上海铁道学院吴守箴编写，第四章及第十章的第六节由西南交通大学于万聚编写，第五、九章及第十章的一、八节由华东交通大学叶雳编写，第六、七、八章及第十章的二、三、四节由西南交通大学李治编写，第十章的第五、七两节分别由西南交通大学刘学军和陈小川编写。

由于编者水平有限，书中难免存在缺点和错误，殷切希望广大读者批评指正。

编　者

1987年9月

内 容 简 介

本书共分十章，内容包括：微型计算机的基础知识；Z80 CPU结构、引脚功能和内部时序；多种半导体存贮器芯片及连接电路；以大量实例说明 Z80 指令系统功能、汇编语言程序设计方法和技巧；中断技术；从软硬件结合角度较详细地介绍了输入／输出接口、数／模和模／数转换电路；详细剖析了 TP 801 单板计算机的监控系统；书中最后介绍了几个微型计算机在电牵引中有代表性的应用实例。

本书可作为电力牵引与传动控制、铁道电气化和工业企业自动化等专业的教材及现场有关技术人员的参考书。

微型计算机及其在电牵引中的应用

WEIXING JISUANJI JIQI ZAI
DIANQIANYIN ZHONG DE YINGYONG

李 治 吴守箴 主编

*

西南交通大学出版社出版
（四川 峨眉）
四川省新华书店发行
西南交通大学出版社印刷厂印刷

*

开本：787×1092 1/16 印张：22.875
字数：582千字 印数：1～3000册
1988年7月第一版 1988年7月第一次印刷
ISBN 7—81022—047—0/TP 006
定 价：4.15元

目　录

第一章　概　论

第二章　微处理器

第三章　存贮器

第四章　Z80 CPU 指令系统

第五章　汇编语言程序设计

第六章　微型计算机的中断系统

第七章　输入/输出接口电路

第八章　模拟通道接口

第九章　TP801 单板机及其监控系统

第十章　微型计算机应用系统

第一章 概 论

第一节 微型计算机发展概况

自从1946年美国的“ENIAC”电子计算机问世以来，计算机科学的发展十分迅速。按其所用的电子器件来划分，至今已经历了电子管数字计算机、晶体管计算机、集成电路计算机和大规模集成电路计算机四代的演变。

随着电子工业和计算技术的飞速发展，可以把电子计算机中多种功能部件成套地制作成大规模集成电路LSI（指在单片硅片上可以集成1 000～20 000个晶体管的集成电路）。这样，只需要用数量较少的LSI，就可以构成一台微型计算机。

自从1971年美国的Intel 4004微型计算机问世以来，短短的十多年，微型计算机也已经历了四代演变。

1971—1973年为第一代。典型产品有Intel 4004、Intel 4040、Rockwell PPS-4等4位机和Intel 8008 8位低档机。芯片集成度为2 000晶体管/片。主要用来代替可编程序的高级台式计算机，或用作小型控制机，可装在电传动打字机、照相机、电视机及台秤等上，以增强装置的智能性。

1973—1978年为第二代。典型产品有Intel 8080、Motorola 6800、Rockwell PPS8、Zilog Z80和Intel 8085等8位机。集成度达8 000晶体管/片，基本指令执行时间为2μs，其功能已接近小型计算机，适用于智能终端、工业生产监测、过程控制、中小型事务处理等，这是微型计算机飞跃发展的新时期。

1978—1981年为第三代。典型产品有Intel 8086、Zilog Z8000、Motorola 68000等16位机。由于超大规模集成电路(VLSI)工艺的研制成功，集成度已达20 000晶体管/片，且64 K位的存贮器也相继产生，其微型计算机的性能已达到或超过中低档小型机（PDP11/45）的水平。适用于科学计算、大型事务处理、多处理机系统、计算机辅助设计和实时控制系统。

1981年以后为第四代。典型产品有Intel IAPX 432、Motorola 68020、Zilog Z80000、Intel 80386等32位机。集成度达10万晶体管/片，其性能可与高档小型机相匹敌。适用于大型数据处理应用系统、实时多用户多任务数据处理、局部网络等。

由于微型计算机体积小、可靠性高、功能强、价格低，因而得以迅猛发展。我国从1975年开始研制微型计算机，1977年试制成仿Intel 8080的DJS-50微型机，1978年用四片芯片组成8080和8228的DJS-51微型机，1979年制成单芯片的8080A及其DJS-052微型机。80年代我国已批量生产DJS-050系列和DJS-060（仿Motorola 6800）的微型机以及4位机芯片DJS-20等配套组件，并积极研制和开发16位微型机。

第二节　微型计算机的组成及其应用

一、微型计算机的组成

1．微型计算机的硬件构成

通常，我们用算盘来进行计算时，若要求回答 $15\times65+36\times18$ 的计算结果时，首先要通过人的眼睛读入原始数据，然后经大脑分析指挥，先用算盘计算 15×65，把计算的中间结果用笔记录在纸上，再用算盘计算 36×18，并把它和上一次的中间结果相加，这样就得到了最终结果，最后再用嘴来回答计算结果。

若要用一台计算机来完成上述计算过程，显然要有能代替算盘进行运算的部件——运算器；能代替笔和纸来存放题目、计算步骤、原始数据和中间结果的部件——存贮器和能代替大脑起指挥作用控制整个计算过程的器件——控制器；能取代眼睛读入原始数据的部件——输入设备以及能代替嘴来回答运算结果的部件——输出设备。因而，一台微型计算机系统硬件是由运算器、存贮器、控制器、输入设备和输出设备五大基本部件组成，如图 1--1 所示。

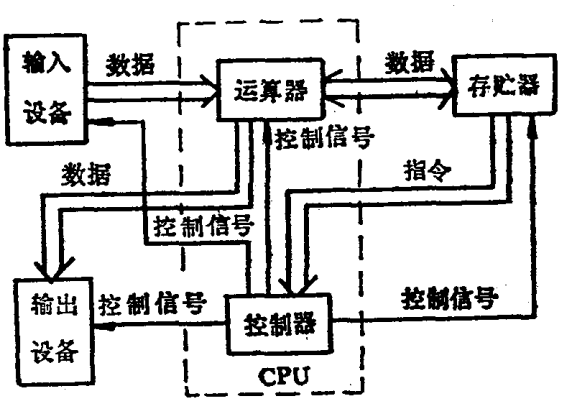

图 1—1　微型计算机系统硬件结构图

2．微型计算机的基本工作原理

为了使微型计算机能自动地进行计算，人们须用命令的形式让计算机执行一步步的操作，这种命令称为指令。如完成上例中 15×65 计算时，

(1) 把数 15 从存贮器的数据区中取出来，送至运算器——取数。

(2) 把数 65 从存贮器的数据区中取出来，送至运算器——取数。

(3) 两数相乘。

(4) 把结果送至存贮器的某一指定单元中——存数。

上述取数、相乘、存数等都是一种操作。通常，每一条指令对应着一种基本操作。计算机所能执行的全部指令，就是计算机的指令系统。指令包括操作码和操作数两大部分。操作码表示计算机执行什么操作，如取数，相加等。操作数表示参加操作的数的本身或操作数所在的地址，即操作的数到哪里去取或操作的结果放到哪里去等。在计算机自动进行计算以前，人们必须根据题意，按照机器能识别的一定格式编制成一条条指令，这一工作称为编程序，这些指令的集合就称为程序。用户为解决某一问题所编的程序，称为源程序。人们通过输入设备预先把一条条指令以数据的形式输入到存贮器中，计算机工作时再把这些指令一条条地取出来，加以译码并执行。

3．微处理器、微计算机和微型计算机系统

微型计算机中，通常把运算器和控制器做在一片或几片大规模集成电路上，合称为微处理器 MP (Micro Proccessor) 或中央处理单元 CPU (Central Proccessing Unit)，有的也称为微处理机 μP。

运算器是用来完成对数据进行基本的算术运算和逻辑运算的部件。例如在控制器的控制下进行“加”、“减”、逻辑“与”、逻辑“或”等运算。8位机一次运算8位二进制代码的数，在计算过程中，运算器不断地从存贮器中读出数据，经运算后，把结果写入存贮器中暂存起来或通过输入输出接口送至I/O设备。

控制器是对全机进行指挥操作的部件。它从存贮器中取出指令，进行译码并按指令的要求，在适当的时刻向各部件发出相应的控制信号，使整个系统有条不紊地工作。

存贮器是用来存贮程序、原始数据、中间结果和最终结果等二进制代码信息的部件。存贮器由许多存贮单元组成，每个单元都有一个唯一的编号称为单元地址，每个存贮单元中存放二进制代码的位数称为字长。例如某存贮器共有4 096（简称4 K）个单元，字长为8位，则其存贮器的容量为4 K×8位。常用的是半导体存贮器。

输入设备通常用来向计算机输入原始数据和程序，常用的有键盘、纸带读入机、卡片读入机等。输出设备通常用来将计算机运算的结果加以显示、打印、制表等，常用的有荧光数码显示管、CRT显示器、电传打字机、行打印机等。微型计算机中，通常把各种输入输出设备统称为计算机的外围（或外部）设备。

输入输出接口电路的作用是将计算机和输入输出设备连接起来，以实现计算机与外部设备工作速度的匹配、工作电平的转换、增大驱动能力、实现数据的串行或并行转换等。

在计算机中，基本上有两种信息在流动，一种是数据信息，另一种是控制信息（见图1—1）。

数据信息流：各种原始数据和程序要由输入设备输入至运算器，再存于存贮器中；存贮器中的指令也以数据形式由存贮器送入控制器中；在运算器运算过程中，数据从存贮器读入运算器进行运算，运算的结果存入存贮器中或经输出设备输出。

控制信息流：指令由控制器经过译码后变为各种控制信号；由控制器控制输入设备的启动或停止；控制运算器按规定一步步地进行各种运算和处理；控制存贮器的读或写；控制输出设备输出结果等。

微处理器、存贮器和I/O接口相互之间是通过系统总线进行信息交换的。系统总线包括数据总线、地址总线和控制总线。数据总线DB（Data Bus）是一组传输数据和指令的公共传输线，它既可以从CPU发出数据，也可由CPU接受数据，故称为双向总线。8位微型机有8根数据总线。地址总线AB（Address Bus）是一组寻找并确定存贮器或I/O接口地址信息的公共传输线，只能从CPU发出地址信息，故为单向总线。地址总线的数目由可寻找的存贮单元数来确定。控制总线CB（Control Bus）是一组由CPU向存贮器或I/O接口以及由存贮器或I/O接口向CPU传送控制信息的传输线。不同的微处理器，其控制总线的数目亦各不相同。

微处理器、存贮器和输入/输出接口组合在一起构成微型计算机（如图1—2所示），又称主机，其中CPU是核心部件。有的是把这三者都集成在一片硅片内，称为单片微型计算机，有的是把这三者组装在一块或多块印刷线路板上，称为单板或多板微型计算机。上述由各类芯片以及电源、输入输出设备等组成的计算机实体部分，即为微型计算机系统的硬件。但是，硬件只使计算机有了计算的可能性，要真正能进行计算还必须要有软件的配合。一台微型计算机再配上软件，就构成微型计算机系统。

为了运行、管理和诊断计算机故障所编制的各种程序的总和称为软件。计算机只能识别

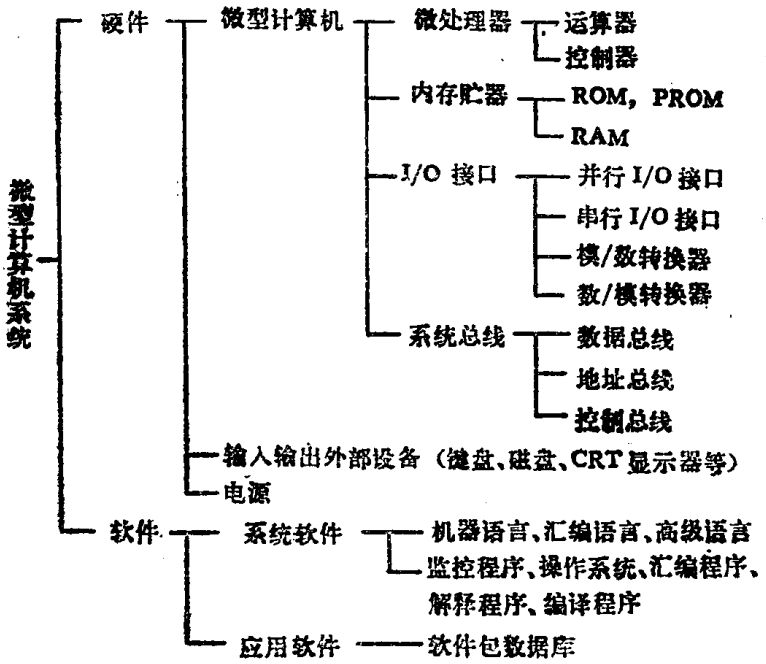

图 1—2 微型计算机系统的组成

“0”和“1”，所执行的指令只能是一连串“0”和“1”组合的形式，称之为机器语言。但用机器语言编制程序非常麻烦，易出错且不易理解和记忆，因而，人们就用一些助记符（通常是指令功能的英文词的缩写）来代替操作码，用一些符号来代替操作数，即用汇编语言来编写程序，以便于书写、理解和记忆，但在计算机执行前，必须将汇编语言写的源程序转换成用机器码所表示的程序（称为目标程序），完成这一功能的程序称为汇编程序。但用汇编语言编写程序时，须对机器的指令系统十分熟悉，且不同机型，其指令系统也不同，因此，为使用户编程更容易，不必了解具体的机器而编制出适用于各种机型，通用性更强的程序，这就出现了各种高级语言，如 BASIC、FORTRAN、PASCAL、COBOL 等，但是在计算机执行时仍需要有解释程序或编译程序，把高级语言的源程序转换成机器码表示的目标程序。

系统软件就是为了方便用户和充分发挥计算机效能，向用户提供的一系列软件。监控程序是实现对计算机监视控制的一套管理程序。应用软件是指编制解决用户各种实际问题的程序。应用软件可逐步标准化、模块化。解决各种典型问题的应用程序的组合称为软件包。

总之，计算机的硬件建立了计算机应用的物质基础，而软件则扩大了计算机的功能且便于用户使用。软件与硬件的结合，才是一个完整的计算机系统。

二、微型计算机的应用

微型计算机之所以得到迅速发展，在于它具有如下的独特优点。

1. 使用灵活的模块化结构

由于微型机实现了硬件功能积木化结构，组装灵活方便，大多数 I/O 接口芯片都是可编程的，能按不同任务进行程序控制，特别是采用了可擦去可编程只读存贮器（EPROM）后，用户可以更改程序以适应新的要求，使微型机能更灵活地适用于各个领域中。

2. 体积小、成本低

由于采用了大规模或超大规模集成电路，把计算机的功能集成在一块或数块芯片中，因

而元件数量减少，组装简单。这样，不仅体积小、重量轻，而且成本亦急剧下降，从而使微型机可作为一个部件组装在小型设备、终端设备和家用电器中，扩大了微型机的应用范围。

3. 可靠性高，耗电少

由于元件集成度高，大大减少了外部连线，提高了可靠性。微型机的功耗比小型机低60～150%，减少了发热，提高了使用寿命。

基于上述特点，微型机的应用已渗透到社会各领域，主要用于以下几方面：

工业：数据监视系统，数值计算，数据处理，工业用机器人，自动化机械，实时控制，技术诊断，自动测试设备，企业管理。

交通通讯：交通信号控制，数据传输控制，数据记录，终端设备，数字通讯系统。

测量仪器：各种智能仪器，自动测试设备，数值测量。

商业：现金出纳，销售终端设备，银行终端，检索系统。

国防：卫星导弹的监视跟踪系统，卫星通讯系统，无人驾驶飞机控制装置等。

办公与民用：办公系统自动化，家用电器，游戏机等。

微型机的应用范围已远远超过了原来计算机的应用范围，对国民经济和人民生活都产生了巨大的影响，其发展趋势是其功能可与价格昂贵的大型计算机相匹敌。

虽然16位机和32位机有着更强的功能和更高的处理速度，然而由于8位微型机软件齐全，性能价格比最优，在一个相当长的时间内，将处于主导地位，其中单板机，单片机在工业控制、智能仪器仪表和计算机外设控制器应用方面显示了无可比拟的优点。

第三节　微型计算机中的数制与码制

微型计算机的最基本功能是进行数的计算和处理加工，但是计算机只能识别“0”或“1”，那么要计算机处理的所有的数、字母、字符等将如何来表示呢？

一、微型计算机中常用的数制及其转换

1. 数的位置表示法及进位计数制

用一组数字（或符号）表示数时，如果每个数字表示的量不但取决于数字本身，而且还决定于它所在的位置，就称为位置表示法。在位置表示法中，对每一个数位赋予一定的位值，称为权。每个数位上的数字所表示的数值是这个数字和该位权的乘积。相应两位中高位的权与低位的权之比为一常数，称为基数或底数。基数的取值不同便得到不同的进位制数。按进位的方法进行计数，称为进位计数制。在计算机中，常用的是二进制和十六进制。

(1) 十进制数

日常生活中，最熟悉和常用的是十进制数。它有两个主要特点。

① 有十个不同的数字符号（又称数码），即0～9。

② 逢十进位。同一个数字符号在不同数位上所表示的值是不同的，例如，

$$(999.99)_{10} = 9\cdot10^{2} + 9\cdot10^{1} + 9\cdot10^{0} + 9\cdot10^{-1} + 9\cdot10^{-2}$$

任意一个十进制数 A 可以表示为

$$(A)_{10} = A_{n-1}\cdot 10^{n-1} + A_{n-2}\cdot 10^{n-2} + \cdots + A_1\cdot 10^1 + A_0\cdot 10^0 + A_{-1}\cdot 10^{-1} + \cdots + A_{-m}\cdot 10^{-m}$$

$$= \sum_{i=0}^{n-1} A_i\cdot 10^i + \sum_{i=-1}^{-m} A_i\cdot 10^i \qquad (1—1)$$

式（1—1）中前项是数的整数值，后项是数的小数值；n 为整数部分的位数，m 为小数部分的位数，n、m 均为正整数；i 为位数，A_i 表示数位上的数码，可取 0～9 中任一个，10^i 为第 i 位的权；10 为基数故称为十进制数。上式称为十进制数的位权展开式。

(2) 二进制数

与十进制数类似，二进制数也有两个主要特点。

① 有两个不同的数码，即 0 和 1 。

② 逢二进位。在不同的数位上，数码所表示的值是不同的。例如，

$$(1011.111)_2 = 1\cdot 2^3 + 0\cdot 2^2 + 1\cdot 2^1 + 1\cdot 2^0 + 1\cdot 2^{-1} + 1\cdot 2^{-2} + 1\cdot 2^{-3}$$

与十进制数类似，任意一个二进制数 B 可以表示为

$$(B)_2 = B_{n-1}\cdot 2^{n-1} + B_{n-2}\cdot 2^{n-2} + \cdots + B_1\cdot 2^1 + B_0\cdot 2^0 + B_{-1}\cdot 2^{-1} + \cdots + B_{-m}\cdot 2^{-m}$$

$$= \sum_{i=0}^{n-1} B_i\cdot 2^i + \sum_{i=-1}^{-m} B_i\cdot 2^i \qquad (1—2)$$

式（1—2）中 B_i 只能取“0”或“1”；n、m 为正整数，n 是整数部分的位数，m 是小数部分的位数；基数是 2 故称为二进制数。

微型计算机均采用二进制数，因为二进制具有如下优点：

① 二进制数只有两个数码，因此它的每一数位都可以用任何具有二个不同稳定状态的元件来表示。例如晶体管的导通和截止，电位的高和低，脉冲的有和无等。

② 二进制数的运算简单，满足下列规律：

0+0=0	0+1=1	1+0=1	1+1=0（有进位）
0−0=0	0−1=1（有借位）	1−0=1	1−1=0
0×0=0	0×1=0	1×0=0	1×1=1

③ 节省设备。要制造有十种状态的元件是很困难的，但若用具有二种状态元件的组合来选择十种不同的状态，则只要 4 个元件。

④ 逻辑运算也只用“0”和“1”两个符号，因而可利用逻辑线路来进行算术运算，这样可采用逻辑代数，为计算机的逻辑设计提供了便利的工具。

但二进制除了不习惯、不易懂以外，书写起来也不方便，为此在书写时常采用十六进制数。

(3) 十六进制数

与十进制数类似，十六进制数也有两个特点。

① 有十六个不同的数码，即 0～9 及 A、B、C、D、E、F。

② 逢十六进位。在不同数位上，数码所表示的值是不同的。例如，

$$(3AB.11)_{16} = 3\cdot 16^2 + A\cdot 16^1 + B\cdot 16^0 + 1\cdot 16^{-1} + 1\cdot 16^{-2}$$

与十进制数类似，任意一个十六进制数 C 可以表示为

$$(C)_{16}=C_{n-1}\cdot16^{n-1}+C_{n-2}\cdot16^{n-2}+\cdots+C_1\cdot16^1+C_0\cdot16^0+C_{-1}\cdot16^{-1}+\cdots+C_{-m}\cdot16^{-m}$$

$$=\sum_{i=0}^{n-1}C_i\cdot16^i+\sum_{i=-1}^{-m}C_i\cdot16^i \qquad (1—3)$$

式（1—3）中 C_i 可取 0～9、A～F 中的任一个；n、m 为正整数，n 是整数部分的位数，m 是小数部分的位数；16 为基数故称为十六进制。

综上所述，计数制的特点为：

(1) 基数为 X 则称为 X 进制数，每一位可在 X 个数码中取值。

(2) 逢 X 进位，每一个数位 i 的权是 X^i，小数点左边各位的权依次是基数 X 的正次幂，而右边各位的权依次是基数 X 的负次幂。因而任一数若向右移一位则相当于减小了 X 倍，若向左移一位则相当于增加了 X 倍。

区分一种数制的基本特征是基数或底数。微型计算机里为区分采用的数制，通常可在数的右下角注明数制或者在数字后面加一字母。例如 B(Binary)表示二进制数制；D(Decimal)或不加字母表示十进制数制；H (Hexadecimal) 表示十六进制数制。如 1011B 表示为二进制数，99 表示为十进制数，ABH 表示为十六进制数。

表 1—1 为十进制、二进制和十六进制数码的对照表。

表 1—1

十进制	二进制	十六进制	十进制	二 进 制	十六进制
0	0000	0	8	1000	8
1	0001	1	9	1001	9
2	0010	2	10	1010	A
3	0011	3	11	1011	B
4	0100	4	12	1100	C
5	0101	5	13	1101	D
6	0110	6	14	1110	E
7	0111	7	15	1111	F

2． 各进位计数制之间的转换

人们习惯于十进制，因此输入的原始数据和输出的结果希望是十进制数，但微型机只能识别二进制，而人们的书写往往又是十六进制，为此须熟悉各进位数制之间的转换。

(1) 任意进位制数转换为十进制数

这种转换很简单，只要将任意进位制数按权展开相加即可。例如，

$$1011.11B=1\cdot2^3+0\cdot2^2+1\cdot2^1+1\cdot2^0+1\cdot2^{-1}+1\cdot2^{-2}=11.75$$

$$3AB.11H = 3\cdot16^2 + A\cdot16^1 + B\cdot16^0 + 1\cdot16^{-1} + 1\cdot16^{-2} = 939.0664063$$

(2) 十进制整数转换为任意进制整数

若要把十进制整数 N 转换为 n 位的任意(X)进制整数，即 $N = (A_{n-1}A_{n-2}\cdots A_2A_1A_0)_X$，就应找到相应的 A_{n-1}，A_{n-2}，…，A_2，A_1，A_0 值。根据 X 进制整数的位权展开式，可写为

$$N = A_{n-1}\cdot X^{n-1} + A_{n-2}\cdot X^{n-2} + \cdots + A_2\cdot X^2 + A_1\cdot X^1 + A_0\cdot X^0 \qquad (1—4)$$

式 (1—4) 右边除了最后一项 A_0 以外均包含有基数 X 的因子，它们都能被 X 除尽，故该式两边同除以基数 X，所得余数即为 A_0，即

$$\frac{N}{X} = \underbrace{A_{n-1}\cdot X^{n-2} + A_{n-2}\cdot X^{n-3} + \cdots + A_2\cdot X^1 + A_1\cdot X^0}_{Q_1} \qquad \text{余数为 } A_0$$

其中，Q_1 为商，余数为 A_0，均为整数。同理，再将 Q_1 除以 X，得商 Q_2 和余数 A_1，

$$\frac{Q_1}{X} = \underbrace{A_{n-1}\cdot X^{n-3} + A_{n-2}\cdot X^{n-4} + \cdots + A_2\cdot X^0}_{Q_2} \qquad \text{余数为 } A_1$$

如此用 X 不断地去除商，直至商等于 0 为止，就可得到 A_{n-1}，A_{n-2}，…，A_2，A_1，A_0 各值。

由上可知，欲把十进制整数转换为 X 进制整数，可采用除 X 取余法。即用 X 不断地去除要转换的十进制数，直至商等于 0 为止，所得各次余数，依次排列即得 X 进制整数。值得注意的是：第一次除 X 所得的余数是最低有效位(LSD——Least Significant Digit)A_0，最后一次得到的是最高有效位 (MSD——Most Significant Digit) A_{n-1}。

例 1 已知十进制数 107，求其相应的二进制数。

解

```
2 | 107
  2 |  53 ……余 1 = A0  (LSD)
    2 |  26 ……余 1 = A1
      2 |  13 ……余 0 = A2
        2 |  6 ……余 1 = A3
          2 |  3 ……余 0 = A4
            2 |  1 ……余 1 = A5
              0 ……余 1 = A6  (MSD)
```

故得 $(107)_{10} = (A_6A_5A_4A_3A_2A_1A_0)_2 = 1101011B$

例 2 已知十进制数 3910，求其相应的十六进制数。

解

```
16 | 3910
   16 | 244 ……余 6 = A0  (LSD)
      16 | 15 ……余 4 = A1
         0 ……余 F = A2  (MSD)
```

故得 $(3910)_{10} = (A_2A_1A_0)_{16} = F46H$

(3) 十进制小数转换为任意进制小数

若要把十进制小数 M 转换为 m 位的任意 (X) 进制小数，即 $M = (0.A_{-1}A_{-2}\cdots A_{-m})_x$，

就应找到相应的 A_{-1}，A_{-2}，…，A_{-m} 的值。根据 X 进制小数的位权展开式可写为

$$M = A_{-1}\cdot X^{-1} + A_{-2}\cdot X^{-2} + A_{-3}\cdot X^{-3} + \cdots + A_{-m}\cdot X^{-m}$$

若将等式两边同乘以基数 X，得

$$XM = A_{-1} + \underbrace{A_{-2}\cdot X^{-1} + A_{-3}\cdot X^{-2} + \cdots + A_{-m}\cdot X^{-m+1}}_{D_1}$$

其中，A_{-1} 为整数部分，它正好等于所求任意（X）进制数的最高位，D_1 为小数部分，若再将 D_1 乘以基数 X，便得

$$XD_1 = A_{-2} + A_{-3}\cdot X^{-1} + \cdots + A_{-m}\cdot X^{-m+2}$$

其中，整数部分正好等于所求任意（X）进制数的次高位 A_{-2}。如此用 X 不断地去乘小数部分，直到小数部分等于 0 为止，就可得到 A_{-1}，A_{-2}，…，A_{-m} 各值。要注意的是：不断地乘以 X 时，不一定都能使小数部分等于 0，这时只要根据精度要求，取足够的位数即可。

由上可知：欲把十进制小数转换为 X 进制小数，可采用乘 X 取整法。用 X 不断地去乘要转换的十进制小数，直到乘积小数等于 0 或按精度要求所得足够位数为止，所得各次整数，依次排列即为 X 进制小数。要注意的是：第一次乘 X 所得的整数是最高有效位 A_{-1}，最后一次得到的是最低有效位 A_{-m}。

例 3 已知十进制小数 0.6875，求其相应的二进制小数。

解

	0.6875	
	× 2	
MSD→	1.3750	
	× 2	
	0.7500	
	× 2	
	1.5000	
	× 2	
LSD→	1.0000	

故得 $(0.6875)_{10} = (0.A_{-1}A_{-2}A_{-3}A_{-4})_2 = 0.1011\text{B}$

例 4 已知十进制小数 0.306，求其相应的二进制小数。

解

	0.306	
	× 2	
MSD→	0.612	
	× 2	
	1.224	
	× 2	
	0.448	
	× 2	
	0.896	
	× 2	
LSD→	1.792	（精度满足）

故得 $(0.306)_{10} = (0.A_{-1}A_{-2}A_{-3}A_{-4}A_{-5})_2 = 0.01001\text{B}$

可见十进制小数并不都能用有限位的二进制小数精确地来表示，这是二进制计数制的一个缺点。

例 5 已知十进制小数 0.65625，求其相应的十六进制小数。

解

```
          0 . 6 5 6 2 5
        ×           1 6
MSD→  |10| 5 0 0 0 0
      ×  |          1 6
LSD→  | 8| 0 0 0 0 0
```

故得 $(0.65625)_{10} = (0.A_{-1}A_{-2})_{16} = 0.A8H$

具有整数和小数部分的任一个十进制数转换为任意（X）进制数时，只要分别将其整数部分和小数部分转换为任意（X）进制数，然后用小数点把这两部分连起来即可。例如，

$(107.6875)_{10} = 1101011.1011B$

$(3910.65625)_{10} = F46.A8H$

(4) 二进制数与十六进制数之间的转换

由于 $2^4 = 16$，故 4 位二进制数相当于一位十六进制数。二进制数与十六进制数之间的相互转换是十分方便的。

① 二进制数转换为十六进制数

转换的方法是将其整数部分由小数点向左，每 4 位一组用“,”分开，最后不足 4 位的前面补 0；小数部分由小数点向右，每 4 位一组用“,”分开，最后不足 4 位的后面补 0，然后把每组的 4 位二进制数用相应的十六进制数代替即可。例如，

0001,1010,0101,1111.1000,1110B = 1A5F.8EH

② 十六进制数转换为二进制数

只要把每一位十六进制数用相应的 4 位二进制数来代替即可。例如，

AB15.3FH = 1010 1011 0001 0101.0011 1111B

二、微型计算机中带符号数的表示法

前面所提到的二进制数都指的是无符号数。但实际的数显然会有正有负，那么在微型计算机中符号是如何表示的呢？通常规定在数的前面增设一位符号位，符号位为“0”表示正数，符号位为“1”表示负数。

1. 机器数和真值

将一个数连同符号一起在机器中作为一个数的则称为机器数。而它的数值称为机器数的真值。

为了运算方便，在计算机中机器数有三种表示法。即原码、反码和补码。

2. 原 码

在二进制数中，符号位为“0”表示正数，符号位为“1”表示负数，其余各位为数值位，这种表示法称为原码表示法。例如，

$$X=+91 \quad \text{则} \quad [X]_{原}=\underbrace{0}_{\text{符号位}}\ \underbrace{1011011}_{\text{数值位}}\ \text{B}$$

$$X=-91 \quad \text{则} \quad [X]_{原}=\underbrace{1}_{\text{符号位}}\ \underbrace{1011011}_{\text{数值位}}\ \text{B}$$

原码表示法具有如下的性质：

(1) $[+0]_{原}=00000000\text{B}$，$[-0]_{原}=10000000\text{B}$，可见在二进制数原码表示法中有正零和负零之分。这两种零的表示形式，在机器中都应将它当作零来处理。

(2) 8位二进制原码所能表示的数值范围为 $+127\sim-127$。

采用原码表示的数直观易懂，且与真值转换方便，因此大部分机器在存贮数据或乘除运算时都采用原码。但原码不便于加减运算。当两数相加时，如果两数符号相同，则数值部分相加后和的符号位不变。当符号位不同的两数相加时，则必须先检查这两数数值的大小，然后用较大的数减去较小的数，再用较大数的符号作为和的符号，这样很麻烦，且使机器的控制线路复杂。为此人们找到了适合计算机进行加减运算的新的机器数的表示法，即反码和补码表示法。

3. 反 码

正数的反码和原码表示相同，即符号位用“0”表示，其余各位为数值位。负数的反码，其符号位用“1”表示，其余位为它的正数的按位取反。例如，

$$X=+127 \quad [X]_{反}=\underbrace{0}_{\text{符号位}}\ \underbrace{1111111}_{\text{数值位}}$$

$$X=-127 \quad [X]_{反}=\underbrace{1}_{\text{符号位}}\ \underbrace{0000000}_{\text{数值位}}$$

值得注意的是：上式中符号位为1时，其余各位（0000000）并不是 X 的值，一定要再把它们按位取反后，才是它的二进制值。又如 $[X]_{反}=10010100$，这是一个负数，它不等于 -20，而等于 $-1101011=-107$。

反码表示法具有如下的性质：

(1) $[+0]_{反}=00000000$，$[-0]_{反}=11111111$。可见在二进制反码中，“0”也有两种表示法。

(2) 8位二进制反码所能表示的数值范围为 $+127\sim-127$。

(3) 一个带符号数用反码表示时，当符号位为“0”时其余各位即为它的二进制值；当符号位为“1”时，将其余各位按位取反后才是它的二进制值。

4. 补 码

为了说明补码的概念，先以时钟为例。若现在是北京时间8点整，而时钟却指着11点整，要将时钟拨准有两种方法：一种是将时钟倒拨3小时（这种逆时钟拨法可视为做减法），相当于 $11-3=8$，另一种是将时钟顺时钟拨9小时（这种顺时钟拨法可视为做加法），相当于 $11+9=12$（自动丢失）$+8=8$。可见对时钟而言，$11-3$ 与 $11+9$ 是等价的，这是因为时针转一圈会自动丢失一个数12。这个自动丢失的数就叫做“模”。模（12）和某个数 X（例如 -3）相加所得的数（$12+(-3)=9$）就称为该数（-3）对模（12）的补数，用 $[X]_{补}$ 表示为

$$[X]_{补}=\text{模}+X$$

这样，某数减去小于模的数，可以用加上模数与该数的负数之和来代替，因此引入了补码后，减法就可以转换为加法了。

在字长为8位的微型机中，因第七位的进位是自然丢失的，故其模数为2^8。若$X=+1000000B$，则以2^8为模的补码（又称为对2的补码，简称补码）为

$$〔X〕_{补}=2^8+X=100000000+1000000=01000000=〔X〕_{原}$$

可见，一个正数的补码与原码相同，都等于其真值。

若$X=-0001010B$，则其补码为

$$〔X〕_{补}=2^8+X=2^8+(-0001010)=100000000-0001010$$
$$=11110110B$$

可见，一个负数的补码仍是一个负数。只有将该数求补后才是它的二进制值。

上述方法求补码时要做一次减法，很不方便。实际应用中可采用下列任一种由原码直接求补码的简便方法：

(1) 一个负数X的补码等于其原码的符号位不变，其余各位按位取反后再在最低位上加1。例如，

$X=-1010111B$　　$〔X〕_{原}=1\underbrace{1010111}_{取反}B$

则　$〔X〕_{补}=10101000+1=10101001B$

(2) 一个负数X的补码等于其原码的符号位不变，其余各位从最低位起到出现第一个“1”以前（包括第一个“1”），原码中的数字不变，以后逐位取反。例如，

$X=-1010100B$　　$〔X〕_{原}=1\underbrace{1010}_{取反}100B$（“1”以前不变）

则　$〔X〕_{补}=10101100B$

补码表示法具有如下性质：

(1) $〔+0〕_{补}=00000000B$，$〔-0〕_{补}=00000000B$，可见，在二进制补码表示中，零的表示是唯一的。

(2) 8位二进制补码所能表示的数值范围为$+127\sim-128$。

(3) 一个带符号数用补码表示时，当符号位为“0”时，其余各位即为它的二进制值；当符号位为“1”时，该数求补后才是它的二进制值。

(4) 当采用补码表示法时，可把减法运算转换为加法运算。例如，

$$65-10=65+(-10)=65+〔-10〕_{补}$$

$$\begin{array}{r} 65=01000001B \\ -)\ 10=00001010B \\ \hline 00110111B \end{array}$$

$$\begin{array}{r} 65=01000001B \\ +)\ 〔-10〕_{补}=11110110B \\ \hline \boxed{1}\ 00110111B \end{array}$$

（1为自动丢失）