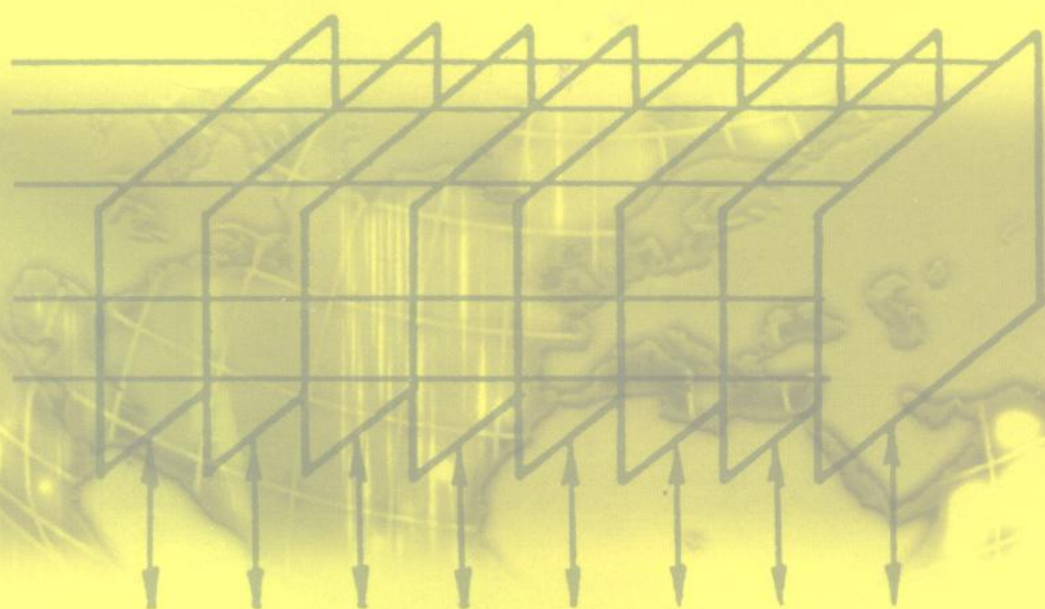


微型计算机 在检测技术及仪器中的应用

主编 阙沛文



上海交通大学出版社

457604

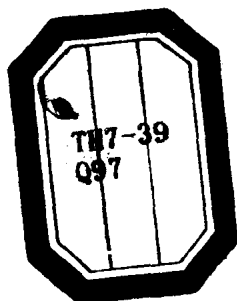
上海交通大学“九五”重点教材

微型计算机在检测技术及仪器中的应用

阙沛文 主编



00457604



上海交通大学出版社

内容提要

本书以 Intel MCS-51 单片机和 IBM-PC 系列机为典型样机,以检测仪器及检测系统为主要应用对象,详细论述了 MCS-51 单片机和 PC 系列机的原理以及它们在检测领域中的应用。本书共分 7 章,内容分别是:微机化检测仪器及系统导论,检测仪器及系统中信息的存储技术,检测仪器及系统中信息的输入/输出技术,MCS-51 单片机剖析,8088CPU 指令系统及汇编语言程序设计,MCS-51 单片机在检测仪器及系统中的应用,基于 PC 系列机的自动检测及仪器系统的设计。

本书可作高等院校检测技术及仪器以及非电类各专业的微机应用课程教材,也可供从事相关专业科技人员参考。

图书在版编目(CIP)数据

微型计算机在检测技术及仪器中的应用/阙沛文主编.
上海:上海交通大学出版社,2000
ISBN 7-313-02356-1

I. 微… II. 阙… III. ①计算机应用-仪器-检测
②微型计算机-基本知识 IV. TH7-39

中国版本图书馆 CIP 数据核字(1999)第 56222 号

微型计算机在检测技术及仪器中的应用

阙沛文 主编

上海交通大学出版社出版发行

(上海市番禺路 877 号 邮政编码 200030)

电话:64281208 出版人:张天蔚

常熟市印刷二厂印刷 全国新华书店经销

开本:787mm×1092mm 1/16 印张:22.25 字数:549 千字

2000 年 1 月第 1 版 2000 年 1 月第 1 次印刷

印数:1-800

ISBN 7-313-02356-1/TH·084 定价:36.00 元

版权所有 侵权必究

前 言

《微型计算机在检测技术及仪器中的应用》是测控技术及仪器专业的技术基础课教材,是一门将微型计算机原理与测控技术及仪器紧密结合的必修课程。尽管目前国内有关微型计算机原理的教科书和参考书很多,但大多数只是介绍微型计算机的基本原理的通用书籍,将微型计算机原理和测控技术及仪器专业应用相结合的教材还不多见。本教材是作者在1989年编写并由上海交通大学出版社出版的《微型计算机在精密仪器中的应用》教材的修订本。由于微型计算机技术的飞速发展,新机型、新技术、新应用等层出不穷,日新月异,因此,微型计算机原理及应用课程的教学内容需要不断更新和充实。本教材介绍的机型由原来的Z-80改为仪器仪表专业经常使用的单片机和16位/32位的PC系列微型计算机。这是比较实用的机型。通过本教材的学习,帮助学生在掌握基本原理时培养分析问题和解决问题的能力,通过学习举一反三,以适应微型计算机不断发展的形势。

本书第1章、第2章、第5章、第7章(第1、2、3、4节)由阙沛文编写,第3章、第4章、第6章(第3节)由罗海福编写,第6章(第1、2节)、第7章(第5节)由茅旭初编写,附录由阙沛文、罗海福编写。本书由阙沛文担任主编,负责全书内容的修改和最后定稿。以上作者均长期从事非计算机专业研究生和本科生的微机课程的教学工作。

由于编者水平的限制,加之时间比较仓促,书中一定存在错误和不妥之处,请读者不吝指正。

编者
1999年7月

目 录

第 1 章 微机化检测仪器及系统导论	(1)
1.1 微机化检测仪器概述	(1)
1.2 自动检测系统概述	(4)
1.3 微机化检测仪器及系统中数据的表示及编码	(5)
1.4 微机概述.....	(15)
1.5 微机化检测仪器及系统的设计过程.....	(20)
习题	(23)
第 2 章 微机化检测仪器及系统的信息存储技术	(24)
2.1 概述.....	(24)
2.2 检测仪器及系统中使用的半导体存储器.....	(25)
2.3 半导体存储器在检测仪器及系统中的连接.....	(37)
习题	(43)
第 3 章 检测仪器及系统中信息的输入/输出技术	(45)
3.1 检测仪器和系统的输入/输出接口电路概述	(45)
3.2 检测仪器及系统的传送方式.....	(47)
3.3 中断技术.....	(52)
习题	(57)
第 4 章 MCS-51 单片机剖析	(58)
4.1 单片机和 MCS-51 系列单片机的结构特点	(58)
4.2 MCS-51 单片机的结构分析	(59)
4.3 MCS-51 单片机的中断系统	(86)
4.4 MCS-51 单片机的时序	(89)
4.5 MCS-51 单片机的指令系统	(93)
4.6 MCS-51 汇编语言程序设计及实例.....	(119)
习题.....	(137)
第 5 章 Intel 8088CPU 指令系统及汇编语言程序设计	(142)
5.1 Intel 8088CPU 的内部结构及存储器的段结构	(142)
5.2 8088CPU 的指令系统	(152)
5.3 8088 汇编语言程序设计技术	(177)
习题.....	(202)
第 6 章 MCS-51 单片机在检测仪器及系统中的应用	(205)
6.1 单片机检测系统的数字 I/O 技术及显示技术	(205)
6.2 单片机检测仪器及系统的前向通道及后向通道(A/D,D/A)	(243)
6.3 微机化弹性元件测试仪的设计	(258)
习题.....	(281)

第 7 章 基于 PC 系列机的自动检测仪器及系统的设计	(282)
7.1 PC 系列机的 CPU 及基本组成	(282)
7.2 PC 系列机的存储器及其管理	(289)
7.3 PC 系列机的 I/O 及中断系统	(294)
7.4 PC 系列机的总线技术	(311)
7.5 基于 PC 系列机的检测仪器设计举例	(322)
7.6 基于 PC 系列机的自动检测系统的设计举例	(329)
习题	(334)
附录 1 MCS-51 系列单片机指令表	(336)
附录 2 DOS 系统功能调用 (INT 21H)	(339)
附录 3 BIOS 中断 (INT) 调用	(343)
附录 4 IBM PC/XT 的中断矢量表	(347)
参考文献	(348)

第 1 章 微机化检测仪器及系统导论

1.1 微机化检测仪器概述

检测技术及仪器应用于人类活动的各个方面,随着科学技术的不断发展,特别是微型计算机(以下简称微机)的出现,使检测仪器进入了一个新的发展时期。一些传统的检测仪器逐步地被微机化检测仪器所代替,这与微机化检测仪器具有自动化、多功能、高精度、高可靠性、体积小、质量轻、功耗小等优点是分不开的,它可以满足生产和科研中所提出的动态、快速、多参数、实时和自动测量以及数据处理的要求。

1.1.1 微机化检测仪器的结构

应用于各个领域的微机化检测仪器,尽管种类繁多、结构复杂程度差异很大,但它的中央控制部件都是由微机组成的。其一般结构形式如图 1.1 所示。

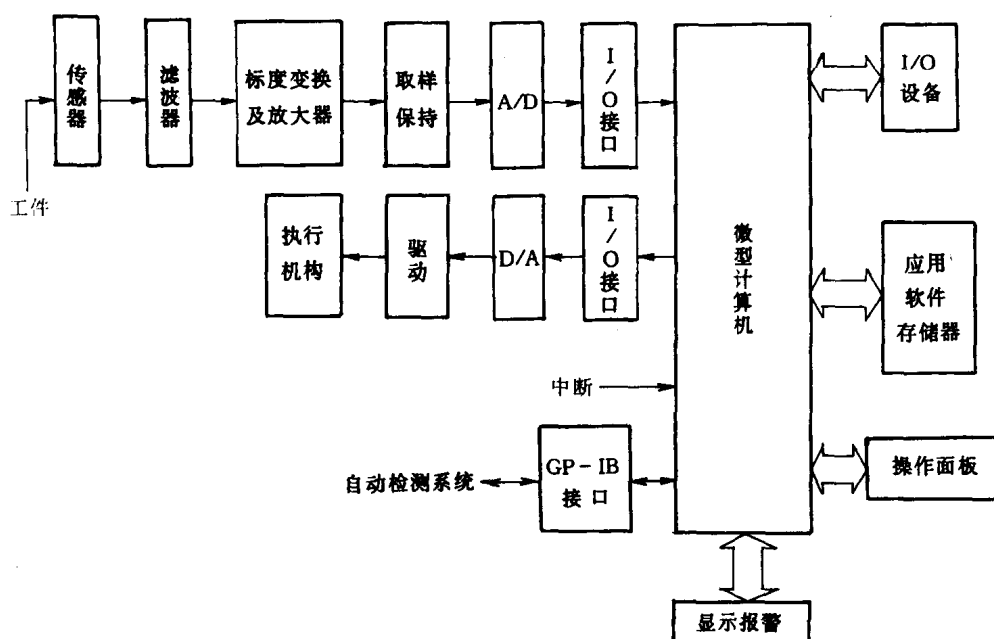


图 1.1 微机化测试仪器的结构

现将图 1.1 中各个组成部分的功能简述如下：

1. 检测仪器的中央控制部件

它是微机化仪器的控制中心,由微机组成。根据仪器的不同功能,微机形式也不同,目前

在检测仪器中广为应用的为三种形式:

(1) 单片机 由于单片机将 CPU, RAM, ROM(或外接 EPROM)计数器/定时器和多种 I/O(如并行接口 I/O 和 A/D 转换器)等电路集成在一块芯片上,因此体积小。在要求功能不太复杂、输入/输出比较少的场合往往采用单片机。

(2) 专用系统 这种系统为某种仪器的特定功能服务,系统中所用的 CPU、存储器、接口芯片及外围设备性能和数量都是根据仪器的使用要求来统一考虑的。设计者自行进行设计并装配成一台完整的仪器系统,其系统软件和应用软件也结合起来进行统一设计,因此要求系统设计人员具有较高的仪器专业知识及计算机应用方面的知识。这一系统研制的时间较长,适用于大批生产场合。

(3) 微机系统 较常用的微机系统有 IBM-PC 机,这种系统在实际应用中较侧重于软件设计,并不要求使用者对硬件有深入的了解。只要在它们的 I/O 总线槽口上安装 A/D 接口板、I/O 接口板等,通过软件对大量的测试数据进行采集、处理、统计、计算,并能将其结果进行显示、输出打印、绘制曲线等。我们称该仪器为插卡式个人仪器(PC 仪器)。

2. 仪器的输入部分

它包括传感器、滤波器、标度变换、取样保持器、A/D 转换器、接口电路及仪器操作面板等。

(1) 传感器 用来将被测对象的非电量信号,如角度、长度、圆度、形位、光洁度、压力、速度等信号变换成相应的电信号。

(2) 滤波器 由于传感器往往装在生产、试验现场,需要通过长线将信号送到仪器的控制中心,易受到外来干扰信号影响,为了消除干扰,一般需加接滤波器。

(3) 标度变换 传感器输出的信号可以是电流也可以是电压,信号的幅度有大有小。由于微机化仪器的输入是电压信号,其幅度一般是 0~5V,所以要进行电信号形式和幅度的变换。若仅仅是幅度变换,则可由放大器来完成。

(4) 取样保持 由于被测信号是变化的,而所需要测量的是某一时刻的信号,所以要对输入信号进行采样,并保持到将这一信号变成数字量为止。这个工作由取样保持器来完成。

(5) A/D 转换器 微机化仪器与一般仪器不同,虽然它也是接收来自传感器的电信号,但却不马上对这些信号进行处理,而是先将电信号经 A/D 转换器转换成数字信号,然后送给微机进行判别、分析和运算,从而得到所需要的结果。这一结果通常是用数字形式表示的,但也可以通过 A/D 转换器转换成模拟形式表示,同时也可以数字形式用可见的方式显示出来。

(6) 接口电路 接口电路是微机与外部世界联络的中转站,它负责将微机向各控制对象所发的指令传输出去并将控制对象的状态或中断请求信号传输给微机。这部分电路不易标准化。对微机化仪器来讲,若厂商提供的通用接口电路满足不了仪器系统的特殊要求时,往往要求设计者需要根据具体情况自行设计。

(7) 仪器操作面板 为了进行人-机联络,一般通过操作面板上的键盘或开关与微机交换一些必要的信息。

以上几部分可以根据仪器的功能选择其中一部分组成某一仪器系统。应该指出的是,输入部分质量对整个微机化仪器的优劣具有决定性意义,必须引起重视。如果输入部分的传感器、标度变换器、A/D 转换器等器件精度、线性度及稳定性不高的话,尽管微机对仪器具有自

校、自动补偿的功能,在一定条件下可以消除输入部分所引起的误差,对整个仪器系统的精度有所提高,但不能完全消除由于输入部分带来的误差,因此对精密仪器输入部分的要求比对一般仪器的要求要高一些。

3. 仪器的输出部分

它包括 D/A 转换器、驱动电路、执行机构、显示、报警等部分。

(1) D/A 转换器 由于微机化仪器往往要控制一些设备,例如马达、电磁阀、灯光、模拟记录器等,而这些设备往往是用电流或电压的变化作为控制输入的,所以要将微机发出的数字形式的控制命令转换成模拟信号。D/A 转换器就是用来实现这种转换的。

(2) 驱动电路 被控设备常常需要较大的控制功率,而微机化仪器通过 D/A 变化器输出‘出的功率往往达不到这个要求,所以必须加上驱动电路以提高激励功率。在常规仪器中也是如此。

(3) 执行机构 它是被控对象,如马达、电磁阀、灯光、模拟记录仪等。

(4) 显示报警 微机将处理后的信息以数字、字母或者图形显示出来,也可以显示出微处理器向人员提出的问题等。同时,当仪器发现情况异常或信号出错时,予以报警。

4. 应用软件存储器

指挥整个仪器工作的应用软件一般存放在应用软件存储器中,它由 ROM(EPROM)或磁盘组成。

5. GP-IB 接口

微机化仪器一般都带有这种接口,可以采用通用接口总线技术,方便地把各种微机化仪器组成一个自动化检测系统。

6. 总线

连接以上各个部分的通信线,如 GPIB 总线、VXI 总线等。

1.1.2 微机化检测仪器的功能特点

检测仪器引入微机后,除了能增强仪器的功能外,还提高了仪器的性能指标。具体讲,微机化检测仪器有以下功能特点:

(1) 实现了仪器操作自动化,因仪器的操作程序事先编制好存放在 ROM(EPROM)或磁盘中,使用人员只需在仪器面板上按动几个功能键,仪器就可以在微机控制下完成复杂的操作程序。用键盘代替了面板的开关和旋钮后,使仪器操作更简便。

(2) 利用微机内存储器的存储功能,可以对大量的数据包括多次测量或采样的数据、各种标准值、备用参数、仪器状态及运算结果进行存储。

(3) 利用微机具有的数据处理能力,对被测的量进行各种运算,如求最大值、最小值和定义各种参数,如角度、圆度、粗度、形状、波度、齿轮参数等;同时也能对误差和干扰进行处理,对测量信号进行数字滤波,以大大减少随机误差及测量误差。例如,通过信号进行多次测量,求

出算术平均值滤波或中值滤波等方法,可减小干扰误差。算术平均值滤波,就是将 n 次测量的结果相加后,取其算术平均值,计算公式为 $\bar{U}_y = \frac{1}{n} \sum_{i=1}^n U_i$ 。微机可以用很方便的算术程序来实现上述计算,完成算术平均值滤波。

(4) 对可分离的系统误差以及环境参数(如温度、湿度、气压等)误差进行自动补偿,提高仪器的精度。

(5) 利用微机的判断能力,可以对仪器进行自校、自诊断,从而提高仪器的稳定度、可靠性和保养性。

(6) 用软件代替硬件。只要改变存放在 ROM 或磁盘中的软件内容,就可以改变仪器的功能,而无需全面改变硬件设计。这种灵活性对批量小、品种多、更新快的仪器是极为重要的。

1.2 自动检测系统概述

在自动检测系统方面,早在 70 年代中期就形成了直至目前仍广泛使用的 GPIB 系统,即在每台测试仪器上设计一个 GPIB 接口,再采用 GPIB 总线技术将其连接成自动测试系统,这种系统以机架叠式微机化仪器为主要器件,构成开放的积木式检测系统。但它的数据线只有 8 位,系统最高数据传输率只有 1M byte/s,体积也较大,这种结构往往不能适应现代科技和生产对测试系统的需要,1987 年推出了一种新的总线技术——VXIbus,在 VXI 总线中地址线和数据线可升至 32 位,在数据线上数据传输率的上限为 40M bytes/s,VXIbus 系统集中了智能仪器、个人仪器和 GPIB 系统的许多特长,并且具有小型便携、使用灵活、数据传输率高、开放式模块化结构、标准化程度高、兼容性强、可扩展性好、器件可重复使用、便于充分发挥计算机效能、易于利用数字信号处理等新的原理和方法及构成虚拟仪器等优点并且便于接入计算机网,构成信息采集、传输和处理一体化的网络,因而得到了迅速发展和推广。这种被称为“未来仪器”或“未来系统”的设备目前就以很高速度进入测试市场。VXIbus 的标准范围也被 IEEE 确定为正式标准 IEEE1155。

以 VXI 总线系统为代表的开放式模块化系统从硬件方面为自动检测系统的组建提供了极大的方便。但是,任何使用微机的设备都必须在软件的支持下才能工作。在 VXI 总线系统中软件几乎没有传统的软件面板,设备的工作基本上靠程序控制。与硬件在世界范围内的开放及标准化相适应,在软件方面也要求有一个标准化的共同基础。在 VXIbus 及 GPIB 等自动测试系统中普遍采用程控领域的两个基本软件标准,即 IEEE488.2 和可编程仪器的标准命令 SCPI。可编程仪器的标准命令 SCPI 是建立在 IEEE488.2 的基础上的。当前用于 VXIbus 系统的各种软件,一般都与这两个标准兼容。

为了指挥微机化设备工作,最终都要使用机器语言,在传统方式下一般用 C、BASIC 等语言逐行编写测试程序,再由计算机转换成机器语言代码。但这种编程、调试过程与高度自动化的测试及对编程开发周期的要求极不适应。因此,世界各大公司都在改进编程和人机交互方面做了大量工作。其中基于图形的用户接口和软件开放环境是应用软件最流行的发展趋势,在这方面有代表性的是 HP 公司的 VEE(可视的工程环境)和 NI 公司的 LabVIEW 开发系统。NI 公司的 LabVIEW 开发系统可在 Windows 环境下工作,通过图形用户界面在分层次和模块化的虚拟仪器方框图中构成系统、输入参数、取得测试结果并自动生成测试程序。

在传统的机架层叠式智能仪器和 GPIB 系统中,微机虽然对仪器和系统进行了统一的管理,但它的作用并未充分发挥,实际上大量激励信号都可先由微机产生数字信号,再经过 D/A 变换产生所需模拟信号。大量的测试功能都可以通过对被测信号的采样,再经 A/D 转换得到数字信号,最后经计算机或数字信号处理器得到测量结果。这样就有可能摆脱由传统硬件构成一件件仪器再连成系统的概念,而变成由计算机资源、A/D 及 D/A 变换器等带共性的硬件资源和计算机应用软件三部分组成虚拟仪器和第三代自动测试系统。

以 VXI 总线系统为代表的新一代模块化检测和仪器系统,体现了测量与仪器技术的多方面重大发展,也是综合了很多先进技术的产物。

对微机化检测仪器及系统的设计者来讲,首先要对计算机原理及其结构、主要芯片、电路设计及软件都比较清楚,才能进行具体检测仪器及系统的设计,所以在以后的章节中,我们以目前用得比较多的 MCS-51 单片机及 IBM-PC 机为样机进行讲解。在理解和掌握一、两种机种以后,才能达到举一反三的目的,运用自如。

1.3 微机化检测仪器及系统中数据的表示及编码

微机化检测仪器及系统的控制中心是微机,因此它们所处理的数据及编码与微型计算机是相同的。众所周知,计算机所处理的信息,必须先经过信息数字化处理,即数据、文字符号、图形等各种信息都要经过编码,成为计算机可以识别和处理的数字信息。而计算机中一般采用二进制数,这是因为二进制数只有“0”和“1”两个数码,不仅易于物理实现、数据存储、传送和处理简便可靠,而且运算规则简单、节省设备,特别是采用二进制后,能方便地使用逻辑代数这一数学工具进行逻辑电路的设计、分析、综合,并使计算机具有逻辑性。因此有必要了解二进制表示及二进制与其他常用进位计数制间的转换方法。

1.3.1 进位计数制及数制转换

人们生活中最熟悉的是十进制,而生活中还有十二进制、十六进制等,其共同之处是采用进位计数制。

1. 进位计数制

进位计数制采用位置表示法,即处于不同位置的同一数字符号,所表示的数值不同。一般说来,如果数制只采用 R 个基本符号,则称基 R 数制,R 称为数制的“基数”或简称“基”(Radix);而数制中每一固定位置对应的单位值称为“权”(Weight)。

进位计数制的编码符合“逢 R 进位”的规则,各位的权是以 R 为底的幂,一个数可按权展开成多项式,例如“逢十进一”的十进制数 1992.5 可写为

$$1992.5 = 1 \times 10^3 + 9 \times 10^2 + 9 \times 10^1 + 2 \times 10^0 + 5 \times 10^{-1}$$

对 R 进制数 N,若用 n+m 个代码 D_i ($-m \leq i \leq n-1$) 表示,从 D_{n-1} 到 D_{-m} 自左至右排列,其按权展开多项式为:

$$N = D_{n-1}R^{n-1} + D_{n-2}R^{n-2} + \cdots + D_0R^0 + \cdots + D_{-m}R^{-m}$$

$$= \sum_{i=-m}^{n-1} D_i R^i$$

其中 D_i 为第 i 位代码,它可取 $0 \sim (R-1)$ 之间的任何数字符号; m 和 n 均为正整数, n 表示整数部分的位数, m 表示小数部分的位数; W_i 表示 D_i 位的权,它是以 R 为底的幂。

下面是计算机常用的进位计数制:

二进制 $R=2$ 基本符号 $0,1$

十进制 $R=10$ 基本符号 $0,1,2,3,4,5,6,7,8,9$

十六进制 $R=16$ 基本符号 $0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F$

其中,十六进制中的数符 $A \sim F$ 字母,分别对应十进制的 $10 \sim 15$,例如一个十六进制数 $8AE6$ 可写为:

$$8AE6H = 8 \times 16^3 + 10 \times 16^2 + 14 \times 16^1 + 6 \times 16^0$$

在数 $8AE6$ 后加的 H 是为识别十六进制数而加的标识字母。由于二进制书写长,难读难懂,为书写方便,计算机中经常使用十六进制。人们又习惯于十进制,而计算机内必须采用二进制,故上面三种数制是经常要用到的。为识别起见,二进制数尾加 B 作标识;十进制加 D 或省略;十六进制加 H 。使用三种进制必然产生各种数制间的相互转换问题。

2. 进位计数制间的相互转换

不同进位计数制间的数据转换,其实质是进行基数的转换,转换原则是根据:两个有理数相等,其指数部分与小数部分分别相等。故要分别进行转换。

(1) 二进制与十进制的相互转换。

① 二进制转换为十进制 其转换规则为“按权相加”,即只有把二进制数中数位是 1 的那些位的权值相加,其和就是等效的十进制数。二进制数整数与小数部分各位权值对应的十进制数值为:

2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
1024	512	256	128	64	32	16	8	4	2	1	0.5	0.25	0.125

例 1.1 将 $10101101.101B$ 转换为十进制数。

$$\begin{aligned} 10101101.101 &= 2^7 + 2^5 + 2^3 + 2^2 + 2^0 + 2^{-1} + 2^{-3} \\ &= 128 + 32 + 8 + 4 + 1 + 0.5 + 0.125 \\ &= 173.625D \end{aligned}$$

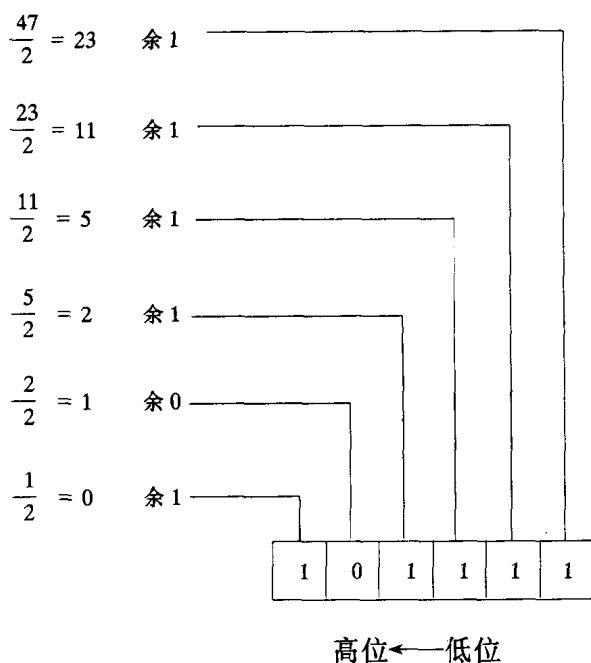
该转换规则同样适应于十六进制数转换为十进制数,只是相加各项是十六进制数符与数位权值之和。

例 1.2 将 $2A.68H$ 转为十进制数。

$$\begin{aligned} 2A.68H &= 2 \times 16^1 + 10 \times 16^0 + 6 \times 16^{-1} + 8 \times 16^{-2} \\ &= 32 + 10 + 0.375 + 0.03125 \\ &= 42.40625D \end{aligned}$$

② 十进制转换为二进制 整数部分转换——将十进制数连续用基数 2 去除,直到商数到 0 为止,每次除得的余数依次为二进制数由低到高的各位值,俗称“除二取余”法。

例 1.3 将 $47D$ 转换为二进制数

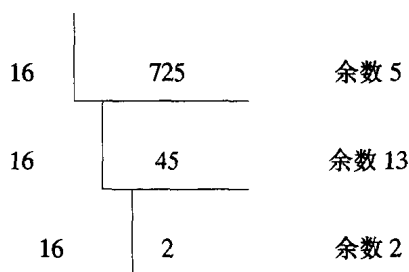


所以 $47D = 101111B$ 。

同理,将十进制数转换为 R 进制数,按“除 R 取余”规则即可。

例 1.4 将十进制数 $725D$ 转换为十六进制。

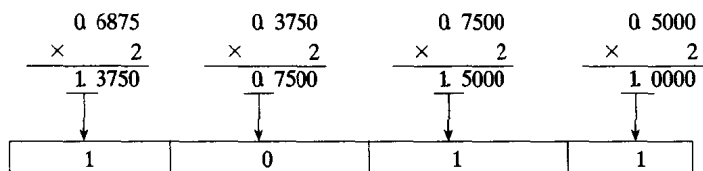
按“除 16 取余”方法进行。



转换结果: $725D = 2D5H$ 。

小数转换——将十进制小数部分连续乘以 2,每次所得乘积的整数部分,依次为二进制数从高到低的各位值,转换则称“乘 2 取整”法。

例 1.5 将 $0.6875D$ 转换为二进制数。



所以 $0.6875D = 0.1011B$ 。

需要注意的是,十进制小数常常不能准确地换算为等值的二进制数,有换算误差存在,转换后的二进制位数,根据字长限制取有限位的近似值。

因此,十进制小数转换为二进制,可按“乘 2 取整”方法进行。对于具有整数和小数的复合十进制数,只要将整数和小数部分分别按上述转换方法进行,最后将其二进制整数和小数部分用小数点连接即可。例如从例 3 和例 5 可得:

$$47.6875D = 101111.1011B$$

(2) 二进制与十六进制之间转换 由于十六进制既可简化书写,又便于记忆,而且与二进制之间转换方便、直观,因此在汇编语言及机器语言中指令、数据书写多采用十六进制。由于十六进制基数与二进制有内在联系,即: $2^4 = 16$ 。因此每位十六进制数可转换为 4 位二进制数,转换直接且方便。

将二进制数转换为十六进制数,只是将二进制数按四位一组划分进行转换。

例 1.6 把 10111001.01101 转换为十六进制数。

$$10111001.01101 = (\underset{B}{10\ 11})(\underset{9}{10\ 01}).(\underset{6}{01\ 10})(\underset{8}{10\ 00})B$$

所以 $10111001.01101B = 0B9.68H$ 。

反之,将十六进制数各位数码用相应四位二进制表示,即可转换为等效的二进制数。

例 1.7 把 5F.7A5H 转换为二进制数。

$$\begin{aligned} 5F.7A5H &= (0101)(1111).(0111)(1010)(0101)B \\ &= 01011111.01111010B \end{aligned}$$

1.3.2 数值数据表示

本节着重讨论两个问题:数值数据在机器内的定点和浮点表示方法;数据的编码方式。

1. 机器数的定点和浮点表示

把数值数据送入计算机处理,仅转换为二进制还不行,必须解决数的符号表示、小数点位置以及用有限设备表示的有效数值范围等问题。因此,计算机内数与实际数表示不一样。数在机器内部的表示形式称为机器数,而机器数对应的数值称为机器数的真值。机器数的符号是数字化处理的,用一位编码表示,通常用 0 表示正数,1 表示负数。机器的小数点位置是事先约定的,它在机器数表示格式中并不出现,但根据设计格式,计算机在运算处理中却清楚地知道其位置。按格式规定,机器数有定点和浮点两种表示形式。

(1) 定点表示 由于采用进位计数制,任何一个二进制数 N 都可表示为

$$N = 2^{\pm P} \times S$$

其中: P 是一个二进制数整数,称为数 N 的阶数(Exponent); 2 为底数, S 是二进制尾数,它表示该数的全部有效数字,而阶码 P 则指明了小数点位置,表明数值范围。

对任一数 $N = 2^{\pm P} \times S$,若阶码位置固定不变,则小数点位置是固定的,这种表示则称为数的定点表示法,该数称为定点数。计算机中定点数通常有两种约定:一是取 $P=0$,把小数点固定在尾数最高位之前,一是取 $P=n$ (n 为尾数的位数),则把小数点约定在尾数最末位之后,这两种情况阶码无需表示,前者即为定点小数,后者为定点整数,其格式如图 1.2 中(a)、(b)所示。

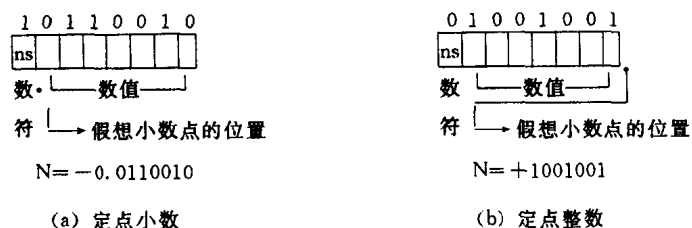


图 1.2 定点数表示

以上两种定点数的表示,计算机均可采用,目前在微机中多采用定点整数形式。这里需强调的是:小数点位置为假像位置,当计算机设计时将表示形式约定好,则各种部件及运算线路均按约定形式进行设计。

机器数字长确定后,其数值表示范围即可确定。例如对定点小数,机器数定长为 N 位,其中一位符号位, n 位有效数值,则 N 位定点小数表示范围为

$$\underbrace{1.11\dots1}_{n\text{位}} \quad (\text{最小值}) \text{ 到 } \underbrace{0.11\dots11}_{n\text{位}} \quad (\text{最大值})$$

对应真值表示范围为

$$-(1-2^{-n}) \leq X \leq 1-2^{-n}$$

例如, $N=16$ 位,则对应表示的真值范围为:

$$-(1-2^{-15}) \leq X \leq 1-2^{-15}$$

用 n 位设备存放定点整数,则所能表示的真值为负 n 位全 1 到正 n 位全 1,即:

$$-(2^n-1) \leq X \leq 2^n-1$$

若用 n 位表示无符号整数,则其整数值表示范围:

$$0 \sim 2^n-1$$

计算机中参加运算的数,若超过计算机所能表示的数值范围,则称之为溢出。这时,计算机要对溢出进行相应处理操作。

(2) 浮点表示 当阶码值固定不变时,数的小数点实际位置将根据阶码值相对浮动,这就构成数的浮点表示。浮点表示要把机器数分为两部分,一部分表示阶码,另一部分表示尾数,阶码和尾数均有各自的符号位。阶符表示数的实际小数点相对约定位置浮动方向:若阶符为负,实际小数点在约定小数点左边,反之在右边,其位置则由阶码值确定,而尾数符号代表了浮点数的符号。图 1.3 所示为浮点数的一种表示形式。

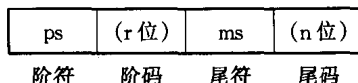


图 1.3 一种浮点数表示形式

若尾数用 n 位小数表示,浮点数的表示范围根据阶码位置 r 和尾数位数 n 决定,其浮点数表示范围为

$$-2^{(2^r-1)} \times (1-2^{-n}) \leq x \leq +2^{(2^r-1)} \times (1-2^{-n})$$

例如对 16 位机器字长,阶码 4 位(包括一位符号位),尾数 12 位(一位符号)则数值表示范围:

$$-2^{(2^3-1)} \times (1-2^{-11}) \leq x \leq +2^{(2^3-1)} \times (1-2^{-11})$$

$$-128 \times (1-2^{-11}) \leq x \leq +128 \times (1-2^{-11})$$

或

可以看出,要扩大数的表示范围,应增加阶码的位数;而要增加精度,就需要增加尾数的位数。对同样字长,浮点数比定点数表示的数值范围要大许多,但浮点数运算操作复杂。浮点数产生溢出实质上是阶码溢出。

2. 机器数的编码表示

数值数据在计算机内采用符号数字化处理后,机器可以表示并识别带符号的数据。为了改进运算方法,简化控制电路,人们研究出多种符号数的编码方式,如原码、反码、补码等。这里介绍最常用的原码和补码形式。

(1) 原码 前面已经讨论过最简单直观的机器数表示,即仅将符号位数字化表示为 0 或 1,数的绝对值与符号一起编码,或者说是“符号-绝对值表示”的编码,称为原码。

$$\begin{aligned} \text{例 1.8} \quad X = +0110111 \quad [X]_{\text{原}} &= 00110111 \\ X = -0110111 \quad [X]_{\text{原}} &= 10110111 \end{aligned}$$

其中 $[X]$ 原称为机器数, X 称为机器的真值。因此,原码直接可从真值求得,只要将符号位用 0 和 1 表示真值的正号和负号,真值的绝对值就是原码的数值部分。

当采用原码表示法时,编码简单直观,与真值转换方便,但也带来一些麻烦:一是引起机器中零的表示不唯一,因为 $[+0]_{\text{原}} = 000 \cdots 0$, $[-0]_{\text{原}} = 100 \cdots 0$,0 有二意性,给机器判零带来麻烦,必须在设计时约定好机器采用正零或负零;二是用原码进行四则运算时,负号位需单独处理,而且原码加减运算规则复杂。例如对有符号数的加法规则为:若两个数同号,两数相加,结果冠以共同的符号。若两个数异号,则由大数中减去小数,结果冠以大数的符号。而减法又有一套规则。

(2) 补码 为使数字化后的符号位能作为数参加运算,并解决减法转化为加法的问题,以简化计算机的运算线路,就产生了补码表示。

如果我们想把十进制减法变成加法做,用什么办法呢?减一个十进制数可采用加该数的十进制补数,然后丢其进位得到相减结果,一个数的十进制补数是用 10 减去该数得到的,例如,十进制减法:

$$9-2=7, \quad 2 \text{ 的补数是 } 10-2=8$$

$$\text{因此} \quad 9-2=9+8=17 \rightarrow 7 \quad (\text{丢掉进位})$$

这里,实际是把 10 看成一个模数,记作 M 。模数在物理上,是某种计量器的度量。在模数系统中:

$$9-2=9+8=17 \rightarrow 7 \quad (M \ 10)$$

上式之所以成立,是因为 2 与 8 对于模数 10 是互为补数的,即 $8=10-2$ 。

生活中使用的模数系统很多,例如时钟为 12,有下式成立:

$$8-3=8+9 \rightarrow 5 \quad (M \ 12)$$

因为 $9=12-3$ 。该式表明将时钟从 8 点倒拨 3 格和正拨 9 格效果一样,时针均指向 5 点,我们称 $+9$ 为 -3 在模 12 下的补码,即 $[-3] = 9 \ (M \ 12)$,这样就可以把减法转化为加法。

计算机中的机器数及其存储、运算设备,如寄存器,加法器等都有固定的长度,因此,都只能进行有模的运算。 n 位设备存放位二进制代码,则 2^n 就是其模数,因为两数相加求和时,如

果 n 位的最高位产生了进位,就会丢掉,这正是模数系统中相加的概念。所以,对任何一个二进制负数都可以找到对应于模数为 2^n 的正补数,从而将减法运算变成加法运算。同时由于 n 位字长中包括一位符号位,故补码运算中的符号和数一起运算。对补码的产生、定义及有关性质的证明,不准备进行过多的讨论,我们只需了解补码的形式及其运算特点。

① 补码的求法。由以上讨论可知,对一个二进制负数可用其模数与真值作加法(实际作减法,因真值为负)求得其补码,即按定义式求

$$[X]_{\text{补}} = \begin{cases} X & 0 \leq X \leq 2^n - 1 \\ 2^n + X & -(2^n - 1) \leq X \leq 0 \end{cases} \quad (M = 2^n)$$

但在机器中实现不方便,因机器中不存在数的真值表示,由于原码表示简单直观,因此在计算机中数是以原码存储的。我们可从数学推导得出一个简便方法,直接从原码求得负数补码。对正数,其补码和原码相同。负数的补码则是将原码符号位保留,其余各位取其反码,即 0 变 1, 1 变 0,再在最低位加 1。而正数的补码与原码相同。即

$$[X]_{\text{原}} \xrightarrow[\text{每位取反,末位加 1}]{\text{除符号位外}} = [X]_{\text{补}}$$

例 1.9 求 $X = +1010101$ 的补码。

X 是正数, $[X]_{\text{补}} = [X]_{\text{原}} = 01010101$

例 1.10 求 $X = -0101110$ 的补码。

X 是负数, $[X]_{\text{原}} = 10101110$

$$[X]_{\text{原}} = 11010001 + 1 = 11010010$$

② 补码的特点 机器数采用补码表示后具有如下几个特点:

(a) 正零、负零表示均为全零,机器零表示统一。

$$\begin{array}{ccc} [+0]_{\text{补}} = 00 \cdots 0 & [-0]_{\text{补}} = 11 \cdots 1 + 1 = \boxed{1} 00 \cdots 0 & \\ n \text{ 位} & \text{自动丢失} & n \text{ 位} \end{array}$$

所以 $[+0]_{\text{补}} = [-0]_{\text{补}} = 00 \cdots 0B$ 。

(b) 运算时符号位无需单独处理。符号位可作为数值一起参加运算,而且在不溢出的情况下,仍能得到正确的结果符号。这是由补码性质决定的。可用数学方法证明,在此我们不证明了。

(c) 采用补码进行加减法运算时,减法可用加法实现。其运算规则为

$$[x]_{\text{补}} + [y]_{\text{补}} = [x + y]_{\text{补}} \quad (M = 2^n)$$

$$[x]_{\text{补}} + [-y]_{\text{补}} = [x - y]_{\text{补}} \quad (M = 2^n)$$

该规则的严密证明略,我们只应用其结论。由于这种转化使得加减法运算只要一套加法设备即可,从而简化了硬件电路,这是补码表示法的一大优点。

在使用补码表示的计算机中,传送和运算处理过程中机器数均以补码形式出现,因此作减法时必须从 $[y]_{\text{补}}$ 求出 $[-y]_{\text{补}}$,求 $[-y]_{\text{补}}$ 方法也很简单,只要对 $[y]_{\text{补}}$ 的每一位(包括符号位)都求反,末位加 1 就得 $[-y]_{\text{补}}$,即

$[y]_{\text{补}}$ 与 $[-y]_{\text{补}}$ 是对模 2^n 互补,故也称 $[-y]_{\text{补}}$ 为 $[y]_{\text{补}}$ 的机器负数。

例 1.11 用补码求 $33 + (-15) = ?$ 。

$$[+33]_{\text{原}} = 00100001B \quad [+33]_{\text{补}} = 00100001B$$

$$[-15]_{\text{原}} = 10001111B \quad [-15]_{\text{补}} = 11110001B$$