

南建辉 熊 鸣 王军茹 编著

MCS-51 单片机

原理及应用实例



清华大学出版社

MCS-51 单片机原理及应用实例

南建辉 熊 鸣 王军茹 编著

清华大学出版社

北 京

内 容 简 介

本书系统、全面地介绍了 MCS-51 系列单片机应用系统的基本原理、系统扩展和 design 方法。本书参考了单片机开发应用的新成果, 结合应用实例, 阐述系统的扩展和组成方法, 实用性强。全书共 9 章。内容包括 MCS-51 单片机组成结构与工作原理、指令系统与程序设计、外部功能扩展与接口技术、应用系统的设计与调试、提高可靠性和抗干扰的措施以及应用实例剖析等。附录中介绍了 MCS-51 指令及其操作码、特殊功能寄存器及其位地址分布, 并给出了本书所用主要芯片的引脚图。

本书具有较丰富的应用实例,深入浅出,阐述清晰。本书适宜从事单片机应用和开发,特别是控制、测试和智能仪器等领域的工程技术人员阅读,也可作为各类单片机中、高级培训班教材,或大专院校相关专业本科生和研究生的教学参考用书。

版权所有，翻印必究。

本书封面贴有清华大学出版社激光防伪标签，无标签者不得销售。

图书在版编目(CIP)数据

MCS-51 单片机原理及应用实例/南建辉, 熊鸣, 王军茹编著.—北京: 清华大学出版社, 2003
ISBN 7-302-07718-5

I. M... II. ①南...②熊...③王... III. 单片机微型计算机, MCS-51 IV. TP368.1

中国版本图书馆 CIP 数据核字 (2003) 第 111079 号

出版者: 清华大学出版社 地址: 北京清华大学学研大厦
http://www.tup.com.cn 邮编: 100084
社总机: 010-62770175 客户服务: 010-62776969

组稿编辑：张瑜

文稿编辑：刘颖

封面设计：陈刘源

印刷者：北京市人民文学印刷厂

装 订 者: 北京市密云县京文制本装订厂

发 行 者：新华书店总店北京发行所

开 本: 185 × 260 印张: 22.25 字数: 526 千字

版次: 2004 年 3 月第 1 版 2004 年 3 月第 1 次印刷

书 号: ISBN 7-302-07718-5/TP · 5650

印 数: 1 ~ 5000

定 价：32.00 元

本书如存在文字不清、漏印以及缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770175-3103 或(010)62795704

前 言

随着电子技术的日益进步,微型计算机的发展突飞猛进。其发展之一就是微处理器及其外围芯片,如程序存储器、数据存储器、并行、串行 I/O 口、定时器/计数器、中断控制器及其他控制部件集成在一个芯片之中,制成单片机(Single Chip Microcomputer)。而近年来推出的一些高档单片机还包含有许多特殊功能单元,例如 A/D、D/A 转换器、调制解调器、通信控制器、锁相环、DMA 和浮点运算单元等。因此,只要外加一些扩展电路及必要的通道接口就可以构成各种计算机应用系统,并具有集成度高、可靠性高、性能价格比高、适应温度范围宽、抗干扰能力强、小巧灵活、易于实现机电一体化等优点,已广泛应用于智能化仪器仪表的检测、控制以及生产设备自动化、家用电器等领域。

MCS-51 是一个独立的高性价比的 8 位单片机系列,具有一定的片内存贮器容量及外部寻址范围,含有全双工串行 I/O 口和 16 位定时器/计数器,并具备乘、除法运算功能,具有较高的运算速度,因而适合于复杂的应用场合。

8 位单片机,特别是高档 8 位单片机的出现是计算机工程应用史上的一个里程碑。近年来,虽然出现了功能更为强大的 16 位单片机,但 8 位单片机以其突出的性价比、成熟的开发和应用技术,仍在单片机应用领域占有非常重要的地位。因此,为满足广大单片机设计开发人员的需要,特编撰此书,奉献给读者。

本书选材注重实用性、系统性、先进性,有利于提高读者应用单片机技术解决实际问题的能力。书中给出了大量的实用电路,还提供了大量的实用子程序,并对程序功能、硬件环境、流程图、入口和出口参数等进行了说明,便于读者查阅和引用。

参与本书编写的人员还有王军茹、熊鸣和祁鲲。肖峻岭、黄敏、姜岩峰、王荣、吴国新、唐勉、张皆喜、古玉海、柏森、王小妮和王巧玲等也做了大量的工作,在此一并感谢。在编写过程中得到了自动化领域资深专家贾永乐教授的指导,特在此表示衷心的感谢。

由于作者的学识水平有限,书中难免有错误或不妥之处,恳请广大读者批评指正。

编者

2003 年 11 月

目 录

第 1 章 单片机的基础知识	1	2.9 串行通信	53
1.1 单片机的发展及应用	1	2.9.1 串行通信及基础知识	53
1.1.1 单片机的发展概况	1	2.9.2 串行接口结构及 控制寄存器	55
1.1.2 单片机的应用	2	2.9.3 串行口工作方式	57
1.1.3 Intel 系列单片机	2	2.9.4 实例分析	63
1.2 计算机运算基础	5	2.10 复位电路	65
1.2.1 数制	5	第 3 章 8051 单片机的指令系统	68
1.2.2 数的表示方法	8	3.1 指令系统概述	68
1.2.3 数的运算方法	13	3.1.1 寻址方式	68
1.3 几个基本概念	14	3.1.2 寻址方式中常用符号注释	72
第 2 章 MCS-51 单片机的组成及 结构分析	17	3.2 MCS-51 单片机指令系统	73
2.1 微型计算机结构及指令执行过程	17	3.2.1 数据传送指令	74
2.1.1 微型计算机结构	17	3.2.2 算术操作类指令	77
2.1.2 单片机结构	18	3.2.3 逻辑操作类指令	80
2.1.3 指令执行过程	18	3.2.4 控制程序转移类指令	82
2.2 MCS-51 单片机内部结构	20	3.2.5 布尔变量操作类指令	89
2.3 引脚描述	23	第 4 章 程序设计	92
2.4 MCS-51 存储器配置	26	4.1 MCS-51 伪指令	92
2.4.1 程序存储器	26	4.2 MCS-51 汇编语言	94
2.4.2 数据存储器	27	4.2.1 汇编语言程序的格式	95
2.5 时钟电路和 CPU 时序	30	4.2.2 汇编程序的汇编	95
2.6 输入输出电路	32	4.3 汇编语言程序设计	96
2.7 中断	34	4.3.1 程序流程图	97
2.7.1 中断技术	34	4.3.2 简单程序设计	97
2.7.2 MCS-51 单片机中断 系统	39	4.3.3 分支程序设计	98
2.7.3 实例分析	43	4.3.4 循环程序设计	101
2.8 定时器/计数器	45	4.3.5 查表程序设计	105
2.8.1 定时器/计数器控制	46	4.3.6 子程序设计	108
2.8.2 定时器工作方式	47	4.4 程序实例	110
2.8.3 实例分析	50	4.4.1 平均滤波程序	110
		4.4.2 定点数乘法运算程序	112

4.4.3 浮点数乘法运算程序	114	5.7.1 8253 芯片介绍	177
第 5 章 MCS-51 单片机的扩展应用	118	5.7.2 8253 和 MCS51 单片机 的接口设计	181
5.1 程序存储器的扩展	118	5.8 单片机外部功能扩展中常用的 串行标准接口	182
5.1.1 外部程序存储器的 操作时序	118	5.8.1 RS232C/RS449 标准 接口	182
5.1.2 地址锁存器及 EEPROM 芯片介绍	119	5.8.2 RS-423/RS-422/RS-485 标准接口	186
5.2 外部数据存储器的扩展	123	第 6 章 MCS-51 单片机的接口技术	190
5.2.1 外部程序存储器的 操作时序	124	6.1 模拟量输入通道接口设计	190
5.2.2 常用 RAM 芯片及 译码芯片	125	6.1.1 A/D 转换器的基本原理 与分类	190
5.2.3 典型 EEPROM 扩展 电路	129	6.1.2 A/D 转换器的技术指标	193
5.3 快速擦存储器在 MCS-51 单片机 系统中的应用	134	6.1.3 8 位 A/D 转换器与 MCS-51 系列单片机 的接口设计	194
5.3.1 Flash 的主要性能特点	135	6.1.4 12 位 A/D 转换器与 MCS-51 系列单片机 的接口设计	199
5.3.2 典型器件的封装和引脚	137	6.1.5 A/D 转换器的应用实例	202
5.3.3 基本工作原理和结构	140	6.2 模拟量输出通道接口设计	206
5.3.4 Flash 和单片机接口及 擦除编程操作	143	6.2.1 D/A 转换器的基本原理	206
5.4 用可编程外围芯片 PSD8XX/9XX 扩展外围功能	146	6.2.2 D/A 转换器的技术指标	208
5.4.1 可编程外围芯片 PSD8XX/9XX 的结构原理	148	6.2.3 8 位 D/A 转换器与 MCS-51 系列单片机 的接口设计	210
5.4.2 用 PSD8XX/9XX 和 51 单片 机设计 ISP 嵌入式系统	150	6.2.4 12 位 D/A 转换器与 MCS-51 系列单片机 的接口设计	214
5.5 并行 I/O 口的扩展	154	6.2.5 D/A 转换器的应用实例	217
5.5.1 I/O 口扩展概述	154	6.3 人机交互接口设计	219
5.5.2 简单 I/O 口的扩展	155	6.3.1 键盘接口设计及其应用 实例	220
5.5.3 8155 可编程并行接口 的扩展	157	6.3.2 LED 接口设计及其应用 实例	225
5.5.4 8255 可编程并行口的扩展	163	6.3.3 LCD 接口设计及其应用 实例	229
5.6 串行 I/O 口的扩展	168		
5.6.1 8251 芯片介绍	169		
5.6.2 8251 和 MCS51 单片机 的接口设计	175		
5.7 定时器/计数器的扩展	176		

第7章 单片机应用系统的设计与**调试234****7.1 应用系统设计方法234**

7.1.1 总体方案论证235

7.1.2 系统硬件设计239

7.1.3 系统软件设计243

7.2 MCS-51 开发系统245**7.3 应用系统的调试255**

7.3.1 硬件调试255

7.3.2 软件调试256

7.3.3 脱机运行调试256

第8章 单片机应用系统的可靠性与**抗干扰性257****8.1 应用系统的可靠性257**

8.1.1 系统可靠性概述257

8.1.2 系统可靠性估算259

8.1.3 提高系统可靠性260

8.2 应用系统的抗干扰267

8.2.1 干扰源及其耦合通道267

8.2.2 抗干扰原则268

8.2.3 抗干扰措施268

第9章 单片机应用系统实例剖析274**9.1 数据采集及显示系统274**

9.1.1 模拟输入子系统的设计274

9.1.2 数据处理278

9.1.3 显示子系统278

9.1.4 系统整合279

9.2 步进电动机的控制程序设计282

9.2.1 步进电机的工作原理及

特性 282

9.2.2 步进电机控制系统原理 284

9.2.3 步进电机与单片机的接口
及程序设计 2879.2.4 步进电机步数及速度的
确定方法 291

9.2.5 步进电机的变速控制 292

9.3 PC 机与 8051 间的通信技术 2949.3.1 PC 机与 8051 间的并行
通讯 2949.3.2 PC 机与 8051 间的串行
通讯 297**9.4 用 8051 单片机实现 CAN
总线节点 309**

9.4.1 CAN 总线协议介绍 310

9.4.2 CAN 通讯控制器 SJA1000 312

9.4.3 CAN 总线系统智能节点
硬件电路设计 3149.4.4 CAN 总线系统智能节点
软件设计 315**附录 A 本书所选用的芯片引脚图 319****附录 B MCS-51 特殊功能寄存器总述 328****附录 C MCS-51 系列单片机指令
一览表 330****参考文献 344**

第 1 章 单片机的基础知识

单片机又称为微控制器，在工业控制中占据很重要的地位。目前，8 位单片机仍占有单片机市场 60% 以上的份额，促进了 8 位单片机朝着高性能和多功能化方向发展。本章从单片机的发展概况讲起，叙述了单片机的应用领域以及 Intel 系列单片机的发展历程。本章还详细讲述了数制的表示方法和数的运算，解释了总线、数据、地址、指令和接口等基本概念。

1.1 单片机的发展及应用

单片机也称为单片微型计算机，是把微型计算机的各部件——中央处理器、存储器、输入输出接口电路、定时器/计数器等制作在一块集成电路芯片上，构成一个完整的微型计算机。

1.1.1 单片机的发展概况

随着超大规模集成电路的出现以及微电子工艺水平的提高，近 10 余年来单片机有了长足的发展。世界上各个著名的集成电路芯片制造商纷纷推出各自的产品，各种类型和型号的单片机犹如雨后春笋般相继问世。

1976 年 Intel 公司首先推出的 MCS-48 系列单片机是一种功能较简单、寻址范围很有限的低性能 8 位单片机，除了一些传统的应用领域(如键盘控制器)外，这类单片机在很大程度上已被 1980 年 Intel 公司推出的 MCS-51 系列高档 8 位单片机所取代。在 MCS-51 系列单片机的内核 8051/80C51 的基础上，Intel 公司、Philips 公司、Siemens 公司等纷纷推出了各种派生芯片。此类派生来的单片机功能丰富、使用灵活，性能价格比远高于 MCS-51 系列单片机，是目前世界上使用得最广的一类单片机。它的发展之路也是其他系列单片机发展的共同道路，它是沿着以下两条路发展的：

- 改进集成电路制造工艺，提高芯片的工作速度，降低工作电压和降低功耗。早期 8051 单片机的最高振荡器频率为 12 MHz，一个机器周期为 1 μ s。Intel 公司后来推出的 80C51ZX 的一个机器周期仅为 1/6 μ s。早期 8051 单片机的工作电压都为 5 V，而 Philips 公司的 80CL51/80CL410 可在 1.8 V 的低电压环境下工作。80CL51/80CL410 是全静态设计的，当芯片采用外部时钟时，可工作于直流状态，即可把外部时钟完全关掉仍能保持芯片的内部状态；当时钟重新加上时，芯片将继续正常工作。当外部时钟停止时，芯片的消耗电流只有 1 μ A。
- 在保留共同的 CPU 体系结构、最基本的外设装置(如异步串行接口、定时器等)和

一套公用的指令系统的基础上, 根据不同的应用领域, 把不同的外设装置集成到芯片内, 在同一家族内繁衍滋生出各种型号的单片机。

Intel 公司在 8051/80C51 的基础上, 已经衍生出 10 种共 50 多个型号的芯片, Philips 公司在 80C51 的基础上, 衍生出 20 多种近 50 个型号的芯片, 使用户可根据系统设计的要求选用合适的型号, 而不必重新熟悉指令系统和 CPU 的结构。除此之外, 还有 Motorola 公司的 MC68HC11(6801 增强型)、Zilog 公司的 Super8 等单片机, 它们代表了单片机发展的方向, 在单片机领域起着越来越大的作用。

随着单片机应用范围的扩大, 人们对单片机的性能要求越来越高, 1983 年 Intel 公司又推出 MCS-96 系列高性能的 16 位单片机, 它的处理速度更高, 功能更齐备。但由于其价格昂贵, 使这种单片机在我国的应用受到限制。1988 年 Intel 公司研制出了 8098 准 16 位单片机, 它属于 MCS-96 系列的子系列, 其性能与 MCS-96 系列的 16 位单片机基本相同, 而价格却与 MCS-51 系列 8 位单片机接近, 因而也受到人们的青睐。

1.1.2 单片机的应用

单片机是一种集成度很高的微型计算机, 在一块小芯片上就集成了一台计算机所具备的功能。单片机具有超小型化、结构紧凑、高可靠性、高抗干扰能力和高性能价格比的优势, 特别适用于实时工业测量控制、智能化仪器仪表、计算机外设和家用电器控制等应用系统。其应用范围如表 1.1 所示。

单片机的发展趋势是: 增加存储容量, 片内 EPROM 开始 EEPROM 化, 存储器编码保密化, 片内 I/O 多功能化及低功耗 CMOS 化。

表 1.1 单片机的应用范围

范 围	具体应用
工业方面	电机控制; 工业机器人; 过程控制; 数字控制; 智能传感器
民用方面	电子玩具、字典、记事簿; 高级电子游戏机; 录像机; 激光驱动、红外线驱动; 照相机、空调机、防盗控制; 调制解调器
汽车方面	点火装置; 变速器控制; 防滑刹车; 排气控制; 避雷控制; 节能控制; 保安控制
仪器仪表方面	智能仪器; 医疗器械; 色谱仪; 示波器
导航与控制方面	导弹控制; 鱼雷制导导弹控制; 智能武器装置; 航天导航系统; 电子干扰装置
数据处理方面	图表终端、图文传真机; 彩色与黑白复印机; 温氏硬盘驱动器; 磁带机、打印机、打字机
电讯方面	智能线路运行控制

1.1.3 Intel 系列单片机

Intel 公司自首创 8048 单片机以来, 其单片机产品的发展十分迅速, 现已形成系列产品, 目前我国主要使用 Intel 公司产品。

1. MCS-48 系列单片机

MCS-48 系列单片机是 Intel 公司的早期产品, 主要用于较简单的应用系统。该系列产品以 8048/8748/8035 为基本类型, 以 8049/8749/8039 为提高机型, 8050 为增强型。前一时我国主要使用 8035/8039、8748/8749 系列单片机, 现在则主要采用 MCS-51 系列单片机。

2. MCS-51 系列单片机

Intel 公司在 MCS-48 系列单片机的基础上, 采用 HMOS 技术, 研制出了 8 位高档的 MCS-51 系列单片机。该机型在性能上与 MCS-48 相比有了很大的改进和提高, 主要反映在以下几个方面:

- 程序存储器 片内程序存储器容量扩大了一倍, 外部程序存储器的寻址空间扩大到 64 KB。
- 数据存储器 片内数据存储器扩大了一倍, 外部数据存储器的空间达 64 KB。
- 并行 I/O 口线 由 MCS-48 的 27 线增加到 32 线, 且可进行位处理。
- 定时器/计数器 MCS-48 只有一个 13 位的定时器/计数器, 而 MCS-51 设有两个 16 位的定时器/计数器(8052 为 3 个), 且可编程设定多种工作方式。
- 串行 I/O 口 MCS-48 没有专门的 I/O 口, MCS-51 设有一个全双工串行 I/O 口, 可编程设定 4 种工作方式。
- 工作寄存器区 MCS-48 有两个通用工作寄存器区, 而 MCS-51 设有 4 个 8 位的通用工作寄存器区, 可适应多级中断和子程序嵌套, 这样可避免寄存器内容进栈保护操作, 提高了中断响应速度, 加速了子程序的调用。
- 中断 MCS-48 只有内部和外部各一个中断源, 而 MCS-51 设有两个内部中断源和两个外部(8052 为 3 个)中断源。还有一个串行接口中断源, 可编程设定中断优先级。
- 堆栈 MCS-48 的堆栈设置在内部 RAM 的固定区段, 且深度只有 8 级(共 16 个单元)。而 MCS-51 堆栈位置可编程设定, 深度可在允许范围内选用。
- 指令系统 MCS-51 指令系统增强了加、减、乘、除、比较、堆栈操作, 因而运算功能大大加强。灵活的跳转指令不仅能充分满足了实际应用的需要, 而且可尽量减少程序对存储空间占用, 当主频为 12 MHz 时, 大部分指令的执行时间为 1 μ s, 部分为 2 μ s。乘、除法指令的执行时间为 4 μ s。
- 布尔处理机 MCS-51 内部设有可直接进行位寻址的存储器、位处理指令、位处理累加器和运算器等, 因而成为一种功能极强的位处理机。这为控制方面的应用和逻辑运算提供了很大方便。

由此可见, MCS-51 系列单片机具有很强的功能, 适用范围广, 既可构成功能很强的复杂系统, 也可组成较简单的应用系统。

在制造工艺上, 也有些产品采用 CMOS 和 HMOS 相结合的 C-HMOS 工艺, 如 80C51 和 80C31 等, 这类产品既保持了 HMOS 高速和高组装密度的特点, 又具有 CMOS 低功耗的优点。

采用 C-HMOS 工艺的单片机具有掉电保护和冻结两种独特的节电运行方式。

在掉电保护方式下, 单片机的功耗可降低到最小值。掉电时, 由备用电源供电, 仅保

证对数据存储器 RAM 供电, 以保持其内容不丢失, 其他部分则停止工作。整个单片机仅吸收电流约 $10\ \mu\text{A}$ 左右。

在冻结运行方式下, 用软件使单片机进入冻结运行状态。这时 CPU 停止工作, 只留定时器和中断部分继续工作, 从而降低了单片机的功耗。结束冻结状态来自定时中断或外部中断信号。

MCS-51 系列的各类单片机产品功能参数如表 1.2 所示。

表 1.2 MCS-51 系列各产品性能参数

型号	8051	80C51	8751	8031	80C31	8052	8032	8044
特点								
程序存储器(B)	4 K (ROM)	4 K (ROM)	4 K (EPROM)	—	—	8 K (ROM)	—	4 K (ROM)
数据存储器(B)	128	128	128	128	128	256	256	192
程序存储器扩展 (片外, B)	60 K	60 K	60 K	64 K	64 K	64 K	60 K	64 K
数据存储器扩展 (片外, B)	64 K	64 K	64 K	64 K	64 K	64 K	64 K	64 K
最高时钟频率(MHz)	12	12	12	12	12	12	12	12
典型指令执行时间(μs)	1	1	1	1	1	1	1	1
16 位定时器/ 计数器数	2	2	2	2	2	3	3	1
并行 I/O 口(位)	32	32	32	16	16	32	16	32
串行 I/O 口(位)	同步方式: 异步方式: 9 位或 10 位可编程							HDLC/ SDLC
中断线	5	5	5	5	5	6	6	5
电源功耗 (i_{ccmax} , mA)	125	24	185	175	24	160	160	200
可编程电源方式								
控电方式	10	$50\ \mu\text{A}$	20	10	$50\ \mu\text{A}$	10	10	30
冻结方式	—	3.0	—	—	3.0	—	—	—

表中 8044 型是高性能新型机种, 设有 8044、8744、8344 三种产品, 由双 CPU 组成, 具有 HDLC/SDLC 通信功能, 能构成廉价的分布式控制局域网, 最大数据传输率可达 2.4Mb/s 。

3. MCS-96 系列单片机

Intel 公司于 1983 年研制出功能更强的、字长为 16 位的 MCS-96 系列单片机, 集成度达 12 万只晶体管以上。它除了对片内存储器容量、并行 I/O 口、定时器/计数器、中断源、运算速度等诸功能均有较大扩展外, 其主要特点是: 设有 6 条高速输出引脚、4 条高速输

入引脚、8 路(或 4 路)10 位 A/D 转换输入、256 级可变占空比的脉宽调制输出, CPU 支持位、字节、字、双字等数据类型的处理。因此, 它具有高速信息处理能力和高速 I/O 功能, 适用于复杂的高性能要求的应用场合。

MCS-96 系列产品列于表 1.3 中。

表 1.3 MCS-96 系列产品

型号	8096-A4	8096-A6	8396-A4	8096-04	8396-04	8096-D6	8396-D4	8396-D6
特点								
程序存储器 (片内 ROM)	—	—	8KB×8	8KB×8	—	—	8KB×8	8KB×8
并行 I/O 线	29	40	29	40	33	48	33	48
A/D 转换	4 路	8 路	4 路	8 路	—	—	—	—
引 脚	48	68	48	68	68	48	48	68
封 装	双列直插	扁平	双列直插	扁平	双列直插	扁平	双列直插	扁平

1.2 计算机运算基础

在计算机中, 数字是用一串由“0”和“1”组成的二进制代码来表示的。二进制有一整套独特的运算规则。本节主要介绍计数方法、数的表示法和运算方法。在计数方法中, 主要介绍各种进位计数制及不同进位计数制之间的转换。在数的表示法中, 主要介绍真值与机器数、定点与浮点数的表示方法、原码、补码、反码及数的编码方法。在运算方法中, 主要介绍定点加减法补码运算及逻辑运算。

1.2.1 数制

1. 进位计数制

按照进位的原则进行计数的数制, 称为进位计数制。

➤ 十进制(Decimal)

十进制计数制中, 是根据“逢十进一”的原则进行计数的。一个十进制数, 其数值是由数字 0、1、2……9 来表示的。数字所处的位置不同, 代表的数的大小也就不同。从右起的第一位是个位, 第二位是十位, 第三位是百位……依次类推。个、十、百、千等在数学上叫做“权”, 十进制数的权是以 10 为底的幂。所使用的数字的个数称为基, 十进制数中为 10。每一位上的数字与该位的权的乘积表示了该位数值的大小。任何一个十进制数 N_D , 可表示为:

$$\begin{aligned}
 N_D &= d_{n-1} \times 10^{n-1} + d_{n-2} \times 10^{n-2} + \cdots + d_1 \times 10^1 + d_0 \times 10^0 + \cdots + d_{-1} \times 10^{-1} + \cdots \\
 &\quad + d_{-m} \times 10^{-m} \\
 &= \sum_{i=-m}^{n-1} d_i \times 10^i
 \end{aligned}$$

例如: $1973 = 1 \times 10^3 + 9 \times 10^2 + 7 \times 10^1 + 3 \times 10^0$

➤ 二进制(Binary)

如果电子计算机使用十进制数,每位数的10个数字需要电路中的10个状态与之对应,这在实际中很难实现。而仅用电路的两个稳定状态就很容易地表示每位数的两个数字,所以计算机内部都使用二进制。

二进制是按照“逢二进一”的原则进行计数的。二进制数的基为2,即其使用的数字为0、1,共两个。二进制数的权是以2为底的幂。一个二进制数 N_B 可表示为:

$$\begin{aligned}
 N_B &= b_{n-1} \times 2^{n-1} + b_{n-2} \times 2^{n-2} + \cdots + b_1 \times 2^1 + b_0 \times 2^0 + b_{-1} \times 2^{-1} + \cdots + b_{-m} \times 2^{-m} \\
 &= \sum_{i=-m}^{n-1} b_i \times 2^i
 \end{aligned}$$

例如: $1011.101 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3}$

➤ 十六进制(Hexadecimal)

十六进制是按“逢十六进一”的原则进行计数的。十六进制数的基为16,即基数字共有16个:0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F,其中A、B、C、D、E、F分别代表十进制数中的10、11、12、13、14、15。十六进制的权为以16为底的幂。一个十六进制数 N_H 可表示为:

$$\begin{aligned}
 N_H &= h_{n-1} \times 16^{n-1} + h_{n-2} \times 16^{n-2} + \cdots + h_1 \times 16^1 + h_0 \times 16^0 + h_{-1} \times 16^{-1} + \cdots + h_{-m} \times 16^{-m} \\
 &= \sum_{i=-m}^{n-1} h_i \times 16^i
 \end{aligned}$$

h_i 为0~F之间的16个数字之一。十六进制以16为基数,第*i*位的权为 16^i ,例如:

$$\begin{aligned}
 ADC.7 &= A \times 16^2 + D \times 16^1 + C \times 16^0 + 7 \times 16^{-1} \\
 &= 10 \times 16^2 + 13 \times 16^1 + 12 \times 16^0 + 7 \times 16^{-1}
 \end{aligned}$$

因使用多种进位计数制,通常用下标予以区别,如2、8、10、16,更常用的是用B、O、D、H表示二、八、十、十六进制数。例如:

$$(1975)_D = (7B7)_H = (11110110111)_B$$

十进制数也可以不写下标,如 $101 = (101)_D$

任何进制数都可表示为如下形式:

$$\begin{aligned}
 N &= a_{n-1} \times r^{n-1} + a_{n-2} \times r^{n-2} + \cdots + a_1 \times r^1 + a_0 \times r^0 + a_{-1} \times r^{-1} + \cdots + a_{-m} \times r^{-m} \\
 &= \sum_{i=-m}^{n-1} a_i \times r^i
 \end{aligned}$$

其中, r 为基数, a_i 为0、1…… $r-1$ 中的任一数字。 N 为整数位数, m 为小数位数。

上式是不同进制之间转换的基础。一般来说,如果要将一个 r_1 进制的数转换为 r_2 进制的数,按照上式展开,再按 r_2 进制的运算法则进行计算即可完成。

常用计数制表示数的方法如表1.4所示。

表 1.4 常用计数制表示数的方法

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. 数值的转换方法

由于我们习惯用十进制计数，所以研究问题时，总是用十进制来考虑和书写的。当考虑成熟后，要把它变成计算机能够“认识”的形式，即把问题中的所有十进制数转换为二进制代码；计算机运算完毕得到二进制结果，又需要转换为十进制数，以十进制的形式显示出来。二进制数书写比较麻烦，经常用十六进制表示，这就需要二进制数与十六进制数的转换。

► 十进制数转换为二进制数的方法

十进制整数转换为二进制整数，通常采用“除二取余法”，即将十进制整数不断除以2，并取余数，直至商为0，第一次所得余数为二进制最低整数位系数。下面举例说明：

【例 1.1】将十进制数 53 转换为二进制数。

解：设 53 所对应的二进制表示如下：

$$(53)_D = b_{n-1} \times 2^{n-1} + b_{n-2} \times 2^{n-2} + \cdots + b_1 \times 2^1 + b_0 \times 2^0$$

如果将两边同除以二进制的基数 2，由上式可见， $53/2 = \text{整数} + \text{余数}/2$ ，所得余数必然等于 b_0 。再依次除 2 取余，便可以得到转换后的二进制各数位的系数 b_1 、 b_2 等。转换过程如下：

$$\begin{array}{rcl}
 \underline{53} & \cdots & \text{余 } 1(b_0) \\
 \underline{26} & \cdots & \text{余 } 0(b_1) \\
 \underline{13} & \cdots & \text{余 } 1(b_2) \\
 \underline{6} & \cdots & \text{余 } 0(b_3) \\
 \underline{3} & \cdots & \text{余 } 1(b_4) \\
 \underline{1} & \cdots & \text{余 } 1(b_5) \\
 0 & &
 \end{array}$$

可得转换结果： $(53)_D = (110101)_B$

十进制小数转换为二进制纯小数，通常采用“乘二取整法”。第一次乘 2 所得整数为转换进制的最高小数系数。

【例 1.2】将十进制小数 0.625 转换为二进制。

解：小数转换后也为小数，可写成如下转换式：

$$(0.625)_D = b_{-1} \times 2^{-1} + \dots + b_{-m} \times 2^{-m}$$

当上式两边同乘以 2 后, 两边整数部分与小数部分应分别相等, 则 0.625 乘 2 所得整数必为 b_{-1} , 将余下小数部分继续乘 2, 可得 b_{-2} 、 b_{-3} 等转换过程如下:

$$2 \times 0.625 = 1.25 \quad \text{取整为 } 1(b_{-1})$$

$$2 \times 0.25 = 0.5 \quad \text{取整为 } 0(b_{-2})$$

$$2 \times 0.5 = 1 \quad \text{取整为 } 1(b_{-3})$$

可得转换结果: $(0.625)_D = (0.101)_B$

转换过程有可能小数部分永不为 0, 可根据精度要求决定转换后的小数位。

➤ 二进制数转换为十进制数的方法

将二进制数转换为十进制数, 只要将二进制数各位的权乘以各位的数字(0 或 1)再相加即可。例如:

$$\begin{aligned} (1011.1010)_B &= 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} + 0 \times 2^{-4} \\ &= (11.625)_D \end{aligned}$$

➤ 二进制与十六进制数之间的转换方法

由于 $2^4=16$, 可见每 4 位二进制数与一位十六进制数对应, 这就使二进制数与十六进制数之间的转换变得简单。

● 二进制数转换为十六进制数

对二进制整数, 只要自右向左将每 4 位分为一组, 不足 4 位在左边用 0 补足; 对二进制小数, 只要从小数点后自左向右将每 4 位二进制数分为一组, 不足 4 位, 在右边用 0 补足 4 位, 然后将每组二进制数用相应的十六进制数代替, 即可完成转换。例如:

$$(3AF.2)_H = 001110101111.0010 = (1110101111.001)_B$$

● 十六进制数转换为二进制数

只要将每 1 位十六进制数用 4 位相应的二进制数表示即可完成转换。例如:

$$(1101011.11)_B = 01101011.1100 = (6B.C)_H$$

1.2.2 数的表示方法

1. 带符号数的表示方法

➤ 机器数与真值

在计算机中, 不仅所有的数都是用“0”和“1”两个数字的组合来表示, 数的符号也只能用“0”和“1”来表示。通常, 将一个有符号数的最高位用作符号位, 以“0”表示“+”号, 以“1”表示“-”号。一个数在机器中的表示形式称为机器数, 而把机器数所代表的实际数值称为该机器数的真值。例如, 一个 8 位机器数与其真值的关系可表示如下:

$$\text{机器数: } x_1 = 01010100$$

$$\text{真值: } x_1 = (+1010100)_B = +84$$

$$x_2 = 11010100$$

$$x_2 = (-1010100)_B = -84$$

➤ 原码

原码(True Form)表示法约定: 符号位为“0”, 表示正数; 符号位为“1”, 表示负数, 其余位与实际数值位相同。如上例所述, 即为 8 位原码表示。通常写成如下形式:

$$x_1 = +84 \quad [x_1]_{\text{原}} = 01010100$$

$$x_2 = -84 \quad [x_2]_{\text{原}} = 11010100$$

原码表示简单直观, 可直接看出数值大小, 但是它有两个缺点:

- 0 的原码有两种表示形式, 即 +0 和 -0:
 $[+0]_{\text{原}} = 00000000 \quad [-0]_{\text{原}} = 10000000$
- 执行加减运算很复杂。计算机中如果用原码做加减运算, 先必须检查参加运算的两个数的符号, 以决定实际应做加法还是减法。如果是做减法, 还要先比较两个数的绝对值大小, 用大数减小数才能保证结果正确。最后还要判断结果的符号。

➤ 补码

补码表示法克服了原码的两个缺点, 能使符号数的加减运算非常方便, 这使计算机内的运算电路及运算过程大为简化。

补码表示法约定: 正数的补码与原码相同; 负数的补码符号位为 1, 而数值位为原码数值各位取反, 且在最低位上加 1。

例如, 前述中 $x_1 = +84$, $x_2 = -84$, 用 8 位补码表示为:

$$[x_1]_{\text{补}} = [x_1]_{\text{原}} = 0100100, [x_2]_{\text{补}} = 10101011 + 1 = 10101100$$

0 的补码是惟一的: $[+0]_{\text{补}} = [-0]_{\text{补}} = 00000000$, 数值为 0 的最小负数 10000000 定为 -128, 因此 8 位补码的表示范围为 -128 ~ +127。

补码的主要优点是加减运算方便, 可将有符号数的减法运算转为加法。1.2.3 节将详细介绍。

日常生活中有不少补码的应用, 如钟表对时。假设标准时间是 7 点整, 而一只钟表的指针指向 9 点整, 这时钟表对时有两种方式:

- 倒拨 2 小时: $9 - 2 = 7(\text{点})$
- 顺拨 10 小时: $9 + 10 = 12 + 7 = 7(\text{点})$

由于钟表的 12 点与 0 点重合, 12 被自然丢弃。所以 $(9-2)$ 与 $(9+10)$ 是等价的, 等价的条件是以 12 为模。所谓模是一个系统的量程, 即此系统所能表示的最大数。对模为 12 的系统来说, 2 与 10 互为补数(补码), 减 2 的运算可用加 10 的运算代替。此外, 3 与 9、4 与 8、6 与 6 等也在模为 12 的系统中互为补数。

上面的例子中, $x_1 - x_2 = x_1 + (\text{模} - x_2) = x_1 + (-x_2)_{\text{补}}$ 。因此一个负数的补数, 是用模减去该数的绝对值来求得。对一个 8 位的二进制数, 模为 $2^8 = 256$, 同样一个负数 x 的补码可用下式求得:

$$[x]_{\text{补}} = 256 - |x|$$

问题是这种求法仍未避免减法, 但根据二进制的特点, $256 - |x| = 255 - |x| + 1$, 而 $255 - |x|$ 的运算结果与 x 数值各位取反的结果完全一样。

2. 数的定点与浮点表示

在计算机中, 小数有两种表示法, 即定点表示法和浮点表示法。所谓定点法, 就是小数点在数中的位置是固定不变的; 所谓浮点表示法, 就是小数点在数中的位置是浮动的。

➤ 定点表示法

通常, 对于意义一个二进制数总可以表示为正整数(或纯小数)和一个 2 的整数次幂的

乘积。例如，二进制数 N 可写成：

$$N = 2^P \times S$$

其中， S 称为数 N 的尾数， P 称为数 N 的阶码。此处 P 和 S 都是用二进制表示的数。尾数 S 表示了数 N 的全部有效数字，阶码 P 指明了小数点的位置。

假定 $P=0$ ，且尾数 S 为纯整数，这时定点数只能表示整数，即

符号位	尾数 S
-----	--------

假定 $P=0$ ，且尾数 S 为纯小数，这时定点数只能表示小数，即

符号位	尾数 S
-----	--------

这两种表示方法在计算机中均有采用。但是对一台具体机器而言，采用哪种方法，都是实现约定。

➤ 浮点表示法

如果阶码是个可变的数值，称这种表示法为数的浮点表示法，这样的数称为浮点数。设：

$$N = 2^P \times S$$

其中，阶码 P 为整数，可为正数，也可用负数；用 P_f 表示阶码的符号位，当 $P_f=0$ 时，表示阶码为正， $P_f=1$ 表示阶码为负。同样，尾数 S 用 1 位二进制数 S_f 表示尾数的符号。

浮点数在机器中表示方法如下：

7	6	5	4	3	2	1	0	
数符	阶符	阶 码						Addr
尾数高 8 位								Addr+1
尾数低 8 位								Addr+2

3. 常用编码

计算机只能识别 0 和 1 两种符号，而计算机处理的信息却有多种形式，例如数字、标点符号、运算符号、各种命令、文字和图形等。要表示这么多的信息并识别它们，就必须对这些信息进行编码。换句话说，以上这众多的信息只有按特定的规则用二进制编码后才能在机器中表示和识别。比如，在操作键盘时，敲击的是字母、数字或功能符号，可输入机器内部的却是相应的一组二进制编码。

计算机中根据信息对象不同，编码的方式(即码制)也不同。常见的码制有二—十进制(BCD)码和 ASCII 码等。

➤ 二—十进制 BCD 码

人们习惯使用十进制，于是计算机的输入和输出通常也采用十进制实现人机交互，不过计算机内部是用二进制编码来表示十进制数的。因为一个十进制数有 0~9 共 10 个数字，至少需要 4 位二进制编码才能表示一位十进制数。4 位二进制数可表示 16 种不同的状态，用它来表示一位十进制数时就要丢掉 6 种状态。根据所用 10 种状态与 10 个十进制数字对应关系的不同，产生了各种二—十进制(BCD)。最常用的方法是以 0000~1001 分别表示 0~