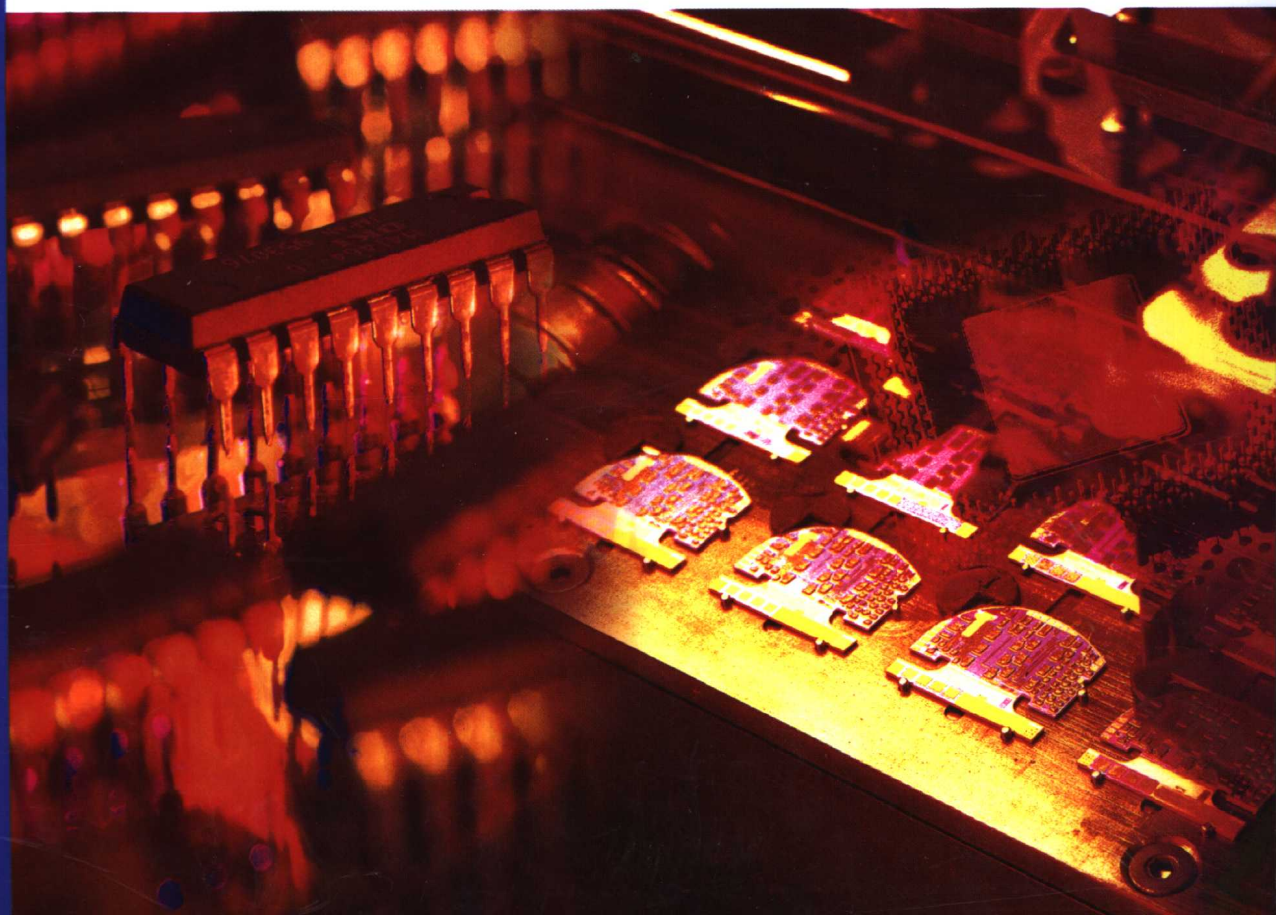


杨恢先 黄辉先 等编著

单片机

原理及应用

国防科技大学出版社



单片机原理及应用

单片机原理及应用

杨恢先 黄辉先 何凤庭 编著
戴永 袁建国

国防科技大学出版社

·长沙·

MJS80 | 01

内 容 简 介

本书深入浅出地介绍了 MCS-51 单片机的原理及应用技术。全书共分 8 章,内容包括:单片微型计算机概述、MCS-51 系列单片机硬件结构、MCS-51 指令系统、汇编语言程序设计、MCS-51 的硬件资源、MCS-51 系列单片机的基本扩展技术、单片机的 C 语言简介及单片机应用系统的设计。各章均精选了一定数量的习题。另外,本书还提供了较为实用的附录供学习时查阅。本书内容新颖、丰富和实用,具有一定的先进性、系统性和实用性。

本书可作为本科电类、计算机类、机械类及其他理工科专业的单片机课程教材,适合自学,也可供从事单片机开发与应用的工程技术人员的参考。

图书在版编目(CIP)数据

单片机原理及应用/杨恢先等编著. —长沙:国防科技大学出版社,2003.3
ISBN 7-81024-929-0

I.单… II.杨… III.单片微型计算机—高等院校—教材 IV.TP368.1

中国版本图书馆 CIP 数据核字(2003)第 006053 号

国防科技大学出版社出版发行
电话:(0731)4572640 邮政编码:410073
E-mail: gfkdcbs@public.cs.hn.cn
责任编辑:潘生 罗青 责任校对:肖滨
新华书店总店北京发行所经销
国防科技大学印刷厂印装

*

开本:787×1092 1/16 印张:22 字数:508 千
2003 年 3 月第 1 版第 1 次印刷 印数:1—7000 册

*

定价:28.00 元

前 言

以大规模集成电路为主组成的微型计算机——单片微型计算机(Single chip micro-computer),简称为单片机,又称为嵌入式微控制器(Embedded microcontroller)。它的诞生是计算机发展史上一个新的里程碑。

由于单片机具有功能强、体积小、可靠性好和价格便宜等独特优点,已成为传统工业技术改造和新产品更新换代的理想机种,具有广泛的发展前景。单片机技术的应用,使得许多领域的技术水平和自动化程度得以大大提高。可以说,当今世界正面临着一场以单片机(微电脑)技术为标志的新技术革命,人们渴望迅速走进单片机应用与开发的大门。作者特地结合自身的教学和科研实践编著了本书,奉献给广大读者。

由于 MCS-51 单片机的模块化结构比较典型,应用灵活,目前在国内外单片机应用中占有重要的地位,因此本书仍以 MCS-51 单片机为主线来展开讨论,但所论述的原理方法,同样适用于其他类型的单片机。

本书共分 8 章。全面而翔实地论述了 MCS-51 单片机的系统结构、系统资源、单片机应用系统的扩展原理和方法。全书结构紧凑,章节编排合理,有一定的先进性、系统性和实用性;文句力求简练、深入浅出和通俗易懂。第一章为单片机概述,简要地介绍了单片机的发展过程及应用领域;第二章至第六章系统地介绍了 MCS-51 单片机的系统结构、指令系统、汇编语言程序设计、内部硬件资源及基本应用、系统基本扩展技术;第七章简单介绍了单片机的 C 语言;第八章介绍了单片机应用系统的基本设计方法。

本书第三、六章由杨恢先编写,第四、七章由黄辉先编写,第五章由何凤庭编写,第八章由戴永编写,第一、二章由袁建国编写。国防科技大学梁加红、廖瑛两位教授详细审阅了全书,并提出了许多宝贵意见,在此表示衷心感谢。

本书在编写过程中,得到了湘潭大学教务处、信息工程学院、材料与光电物理学院领导的关怀与支持,在此一并表示感谢。

在本书的编写的过程中,杨穗、王子菡、代秋芳、徐又又、汤红忠、肖业伟、吴明辉等同志整理了部分文字和插图等资料,在此一并致谢。

限于作者水平,书中错误和不妥之处在所难免,敬请读者批评指正。

编 者

2002 年 7 月

目 录

第一章 单片微型计算机概述及预备知识	(1)
1.1 单片机的发展历史和发展趋势	(1)
1.1.1 单片机的发展历史	(1)
1.1.2 单片机的发展趋势	(2)
1.2 计算机的数制和码制及数制之间转换	(3)
1.2.1 计算机的数制	(3)
1.2.2 计算机的码制	(4)
1.2.3 各数制间的转换	(7)
1.3 计算机中数的表示方法	(10)
1.4 计算机的二进制数运算	(13)
1.4.1 算术运算	(13)
1.4.2 逻辑运算	(15)
1.5 常用的单片机产品	(16)
1.6 单片机的应用领域及系统结构	(17)
1.6.1 单片机的特点	(17)
1.6.2 单片机的应用领域	(18)
1.6.3 单片机应用系统的结构	(19)
习 题	(20)
第二章 MCS-51 系列单片机硬件结构	(21)
2.1 总体结构	(21)
2.1.1 8051 的总体结构	(21)
2.1.2 MCS-51 系列单片机的一般结构	(22)
2.2 MCS-51 单片机的外部引脚	(23)
2.3 MCS-51 单片机的时钟和时钟电路	(25)
2.4 MCS-51CPU 时序	(26)
2.5 MCS-51 单片机的复位和复位电路	(27)
2.6 MCS-51 单片机的存贮器	(29)
2.6.1 程序存贮器	(29)
2.6.2 数据存贮器	(30)
2.6.3 特殊功能寄存器	(32)
2.6.4 位存贮器	(34)

习 题	(35)
第三章 MCS-51 指令系统	(36)
3.1 汇编语言指令格式	(36)
3.1.1 汇编语言格式	(36)
3.1.2 汇编伪指令	(36)
3.2 MCS-51 的寻址方式	(40)
3.3 MCS-51 的指令系统	(44)
3.3.1 数据传送类指令	(45)
3.3.2 算术运算类指令	(49)
3.3.3 逻辑运算类指令	(54)
3.3.4 位操作类指令	(56)
3.3.5 控制转移类指令	(59)
习 题	(66)
第四章 汇编语言程序设计	(68)
4.1 汇编语言程序结构及基本设计方法	(68)
4.1.1 程序设计步骤	(68)
4.1.2 程序结构及基本设计方法	(68)
4.2 简单和分支程序设计	(69)
4.2.1 简单程序设计	(69)
4.2.2 分支程序设计	(70)
4.2.3 散转程序设计	(73)
4.3 循环和查表程序设计	(74)
4.3.1 循环程序设计	(74)
4.3.2 查表程序设计	(79)
4.4 子程序和运算程序设计	(81)
4.4.1 子程序设计	(81)
4.4.2 运算程序设计	(85)
4.4.3 代码转换程序	(97)
习 题	(100)
第五章 MCS-51 单片机的硬件资源	(101)
5.1 MCS-51 的并行 I/O 口及其应用	(101)
5.1.1 MCS-51 内部并行 I/O 口	(101)
5.1.2 MCS-51 内部并行 I/O 口的应用	(104)
5.2 MCS-51 的中断系统	(106)
5.2.1 中断概述	(106)

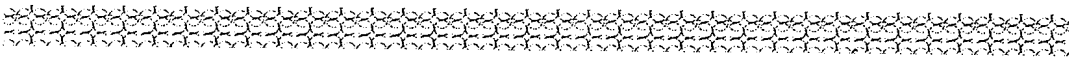
5.2.2	MCS-51 中断系统与控制	(108)
5.2.3	中断响应	(111)
5.2.4	中断请求的撤除	(112)
5.2.5	中断系统的初始化	(114)
5.2.6	MCS-51 单片机的单步工作方式	(115)
5.2.7	外部中断源的扩展	(116)
5.3	MCS-51 的定时器/计数器	(118)
5.3.1	定时方法概述	(118)
5.3.2	定时器/计数器的定时和计数功能	(118)
5.3.3	定时器/计数器的控制寄存器	(119)
5.3.4	定时器/计数器的工作方式	(120)
5.4	MCS-51 的串行接口及串行通信	(127)
5.4.1	串行通信基础	(127)
5.4.2	MCS-51 的串行口及控制寄存器	(131)
5.4.3	MCS-51 的串行通信工作方式	(134)
习 题		(149)

第六章 MCS-51 单片机的基本扩展技术

6.1	MCS-51 并行扩展总线	(151)
6.1.1	MCS-51 并行扩展总线方法	(151)
6.1.2	地址译码方法	(153)
6.1.3	总线驱动能力及扩展方法	(155)
6.2	MCS-51 存贮器的扩展	(157)
6.2.1	程序存贮器的扩展	(157)
6.2.2	数据存贮器的扩展	(159)
6.2.3	存贮器扩展举例	(160)
6.3	MCS-51 输入输出并行接口的扩展	(160)
6.3.1	采用 8255A 扩展并行 I/O 接口	(162)
6.3.2	采用 8155 扩展并行 I/O 接口	(170)
6.3.3	采用锁存器扩展并行 I/O 接口	(176)
6.3.4	采用 MCS-51 的串行口扩展并行 I/O 接口	(179)
6.4	D/A 转换器接口的扩展	(181)
6.4.1	梯形电阻式 D/A 转换器原理	(181)
6.4.2	D/A 芯片 DAC0832 的扩展	(182)
6.4.3	D/A 芯片 AD7520 的扩展	(186)
6.5	A/D 转换器接口的扩展	(188)
6.5.1	双积分式 A/D 转换器原理	(189)
6.5.2	双积分式 A/D 芯片 MC14433 的扩展	(189)

6.5.3 逐次逼近式 A/D 转换器原理	(193)
6.5.4 逐次逼近式 A/D 芯片 ADC0809 的扩展	(194)
6.6 显示、键盘及接口	(196)
6.6.1 LED 显示器及接口	(196)
6.6.2 LCD 显示器及接口	(200)
6.6.3 键盘及接口	(207)
6.7 专用键盘、显示接口芯片 8279 的扩展	(213)
6.8 微型打印机接口	(223)
习 题	(233)
第七章 单片机的 C 语言简介	(234)
7.1 C 语言的基本知识	(234)
7.1.1 C 语言的特点与程序结构	(234)
7.1.2 C 语言的标识符和关键字	(235)
7.1.3 单片机 C51 程序的编译连接与调试	(237)
7.2 C51 程序设计的基本语法	(238)
7.2.1 C 语言的数据类型	(238)
7.2.2 常量	(240)
7.2.3 变量及存储模式	(241)
7.2.4 运算符与表达式	(242)
7.3 C51 程序设计	(246)
7.3.1 表达式语句	(246)
7.3.2 复合语句	(247)
7.3.3 流程控制语句	(247)
7.4 函数	(253)
7.4.1 函数的定义	(253)
7.4.2 函数的调用形式	(254)
7.4.3 中断服务函数与寄存器组定义	(255)
7.4.4 变量的存储方式	(257)
7.5 模块化程序设计	(259)
7.5.1 基本概念	(259)
7.5.2 模块设计原则	(260)
7.5.3 常用算法简介	(261)
7.6 C51 编程实例	(262)
7.6.1 8051 串行口扩展的矩阵键盘	(262)
7.6.2 8051 单片机并行接口扩展	(264)
7.6.3 D/A 转换接口及其驱动程序	(266)

第八章 单片机系统的设计及应用举例	(268)
8.1 单片机应用条件	(268)
8.1.1 单片机可应用场合的确定	(268)
8.1.2 单片机应用层次的确定	(268)
8.1.3 单片机应用地位的确定	(268)
8.2 单片机系统设计要领	(269)
8.2.1 对象特性分析要领	(269)
8.2.2 硬件体系设计要领	(269)
8.2.3 算法设计及优化要领	(270)
8.2.4 软件设计及优化要领	(270)
8.3 单片机系统的设计过程	(270)
8.3.1 单片机系统的总体设计	(270)
8.3.2 单片机系统的硬件设计	(272)
8.3.3 单片机系统的软件设计	(273)
8.3.4 单片机系统的可靠性设计	(276)
8.4 单片机与软盘驱动器接口技术及应用	(282)
8.4.1 软盘驱动器的选择及接口电路设计	(283)
8.4.2 软盘驱动器接口命令组成及单片机控制程序设计	(283)
8.5 MCS-51 单片机用于电子配料秤	(297)
8.5.1 电子配料秤的工作原理	(297)
8.5.2 单元电路分析	(299)
8.5.3 调试程序	(306)
8.5.4 精度计算	(306)
8.6 在单片机控制系统中实现可控硅(SCR)的过零控制	(310)
8.6.1 数字实现 SCR 过零控制的方法	(310)
8.6.2 单片机温度控制系统实例	(313)
附录 A: MCS-51 指令表	(324)
附录 B: MCS-51 指令助记符注释表	(329)
附录 C: 常用的缩写符号	(331)
附录 D: ASCII 码表	(332)
附录 E: C51 的库函数	(333)
附录 F: 常用集成电路引脚图	(336)
参考文献	(341)



第一章 单片微型计算机概述及预备知识

自从 1946 年美国宾夕法尼亚大学研制了世界上第一台电子计算机 ENIAC(Electronic Numerical Integrator And Computer)以来,计算机的发展经历了四个时代(电子管时代、晶体管时代、集成电路时代、大规模及超大规模集成电路时代)。现代的计算机都是大规模集成电路计算机,它们具有功能强、结构紧凑、系统可靠等特点,其发展趋势是巨型化、微型化、网络化及智能化。微型化是计算机发展的重要方向,把计算机的运算器、控制器、存贮器、输入/输出(I/O)接口四个组成部分集成在一个硅片内,于是就出现了以一个大规模集成电路为主组成的微型计算机——单片微型计算机(Single chip microcomputer),简称为单片机,由于单片机的重要应用领域为智能化电子产品,一般需要嵌入仪器设备内,故又称为嵌入式微控制器(Embedded microcontroller)。目前生产单片机的厂商很多,产品型号规格数不胜数,性能亦各具特色。本书以我国应用较早、且仍占有较大市场份额的 Intel 公司 MCS-51 系列单片机为例加以介绍。

1.1 单片机的发展历史和发展趋势

1.1.1 单片机的发展历史

单片机的发展可以分为四个阶段:

第一阶段(1974~1976年):单片机初级阶段。因工艺限制,此阶段的单片机采用双片的形式而且功能比较简单。例如仙童公司生产的 F8 单片机,只包括了 8 位 CPU(Central Processing Unit,称为中央处理器,由计算机的运算器和控制器组成,是计算机的核心),64 个字节的 RAM(Random Access Memory,随机存取存贮器)和两个并行口,需要加一块具有 1KB ROM(Read Only Memory,只读存贮器)、定时器/计数器和两个并行口的 3851 芯片才能组成一台完整的计算机。

第二阶段(1976~1978年):低性能单片机阶段。此阶段的单片机已为一台完整的计算机,但内部资源不够丰富,以 Intel 公司生产的 MCS-48 系列为代表,片内集成了 8 位 CPU、8 位定时器/计数器、RAM 和 ROM 等,但无串行口,中断系统也比较简单,片内 RAM 和 ROM 容量较小且寻址范围不大于 4KB。

第三阶段(1978~):高性能单片机阶段。此阶段的单片机内部资源丰富,以 Intel 公司生产的 MCS-51 系列为代表,片内集成了 8 位 CPU、16 位定时器/计数器、串行 I/O 口、多级中断系统、RAM 和 ROM 等,片内 RAM 和 ROM 容量加大,寻址范围可达 64KB。

有的型号内部还带有 A/D 转换器。

第四阶段(1982~):8 位单片机的巩固发展及 16 位、32 位单片机推出阶段。16 位单片机以 Intel 公司生产的 MCS-96 系列为代表,在片内带有多通道 A/D 转换器和高速输入/输出(HSI/HSO)部件,中断处理和实时处理能力很强。

目前单片机的品种众多,其中性能优良的 8 位单片机在今后若干年内仍将是工业检测、控制应用领域中的主角。

1.1.2 单片机的发展趋势

单片机的发展趋势是向高性能、大容量、微型化、外围电路内装化等方面发展。

1. CPU 的改进

(1)采用双 CPU 结构。以提高处理速度和处理能力。

(2)增加数据总线宽度。以提高数据处理速度和能力。

(3)采用流水线结构。指令以队列形式出现在 CPU 中,且具有很快的运算速度,尤其适合于实时数字信号处理。

(4)串行总线结构。飞利浦公司开发了一种新型总线——I²C 总线(Inter-ICbus),该总线采用三条数据线代替现行的 8 位数据总线,从而大大地减少了单片机的引线,降低了单片机的成本,特别适用于电子仪器设备的微型化。

2. 存储器的的发展

(1)增大存储容量。新型单片机片内 ROM 一般可达 4KB 至 8KB,有的甚至可达 128KB。片内 RAM 可达 256 字节。片内存储器存储容量的增大有利于外围扩展电路的简化,从而提高产品的稳定性,降低产品的成本。

(2)片内 EPROM 开始 E²PROM 化。片内 EPROM 由于需要高压编程写入、紫外线擦抹删除,存在诸多不便。采用可电改写的 E²PROM 后,不需要紫外线擦抹,只需重新写入。特别是能在 +5V 下读写的 E²PROM,既有静态 RAM 读写操作简便的优点,又有在掉电时数据不会丢失这一 ROM 的优点。片内 E²PROM 的使用不仅会对单片机结构产生影响,而且会大大简化应用系统的组成结构,从而提高产品的稳定性,降低产品的成本。由于 E²PROM 中数据写入后能永久保持,因此有的单片机将它作为片内 RAM 使用,甚至有的单片机将 E²PROM 用作片内通用寄存器使用。

(3)程序保密化。一般 EPROM 中的程序很容易被复制,为防止复制,某些公司开始采用 KEPROM(Keyed access EPROM)编程写入,对片内 EPROM 或 E²PROM 采用加锁方式。加锁后,无法读出其中的程序,防止应用系统程序被抄袭。

3. 片内 I/O 的改进

一般单片机都有较多的并行口,以满足外围设备、芯片扩展的需要,并配有串行口,以满足多机通信功能的要求。

(1)提高并行口的驱动能力。这样可减少外围驱动芯片。有的单片机能直接输出大电流和高电压,以便能直接驱动 LED 和 VFD(荧光显示器)等。

(2)增加 I/O 口的逻辑控制功能。中、高档单片机的位处理系统能够对 I/O 口线进

行位寻址及位操作,加强了 I/O 口线控制的灵活性。

(3)特殊的串行接口功能。为单片机构成网络系统提供更便利的条件。

4. 外围电路内装化

随着集成电路集成度的不断提高,有可能把众多的外围功能电路集成到单片机芯片内。除了一般必须具备的 ROM、RAM、定时器/计数器、中断系统外,为满足检测、控制功能更高的要求,片内集成的部件还可能有 A/D 转换器、D/A 转换器、DMA 控制器、中断控制器、锁相环、频率合成器、字符发生器、声音发生器、CRT 控制器、译码驱动器等。由于集成工艺在不断地改进和提高,能集成于片内的外围电路也可以是大规模的,把所需的外围电路全部集成到单片机内,即系统的单片化是目前单片机发展的趋势。

5. 低功耗化

在 8 位单片机中有半数以上产品已 CMOS 化,CMOS 单片机具有功耗小的优点。为了充分发挥低功耗的特点,这类单片机普遍设置有空闲和掉电两种工作方式。如采用 CHMOS 工艺的 MCS—51 系列单片机 80C51BH/80C31/87C51 在正常运行时(5V, 12MHz),工作电流为 16mA;同样条件下在空闲方式工作时,工作电流为 3.7mA;而在掉电方式(2V)工作时,工作电流仅为 50nA。

1.2 计算机的数制和码制及数制之间的转换

所谓数制是指数的形式,是人们利用符号计数的一种科学方法。数制是人类在长期的生存斗争和社会实践中逐步形成的。数制有很多种,常用的数制有十进制、二进制、八进制和十六进制四种。计算机都是以二进制形式进行算术运算和逻辑操作的,因此,用户在键盘上输入的十进制数字和符号命令,计算机都必须先把它们转换成二进制形式进行识别、运算和处理,然后再把运算或处理的结果还原成十进制数字和符号供显示或打印。为此,必须事先为这些数字、符号等进行二进制编码,当然所需编码的数字、符号等越多,编码的二进制数字位数也越长。

1.2.1 计算机的数制

进位计数制的特征可概括如下:

- 有一个固定的基数 r , 数的每一位只能取大于等于 0, 小于 r 的数字, 即符号集是 $\{0, 1, 2, \dots, r-1\}$;
- 逢 r 进位, 它的第 i 个数位对应于一个固定的值 r^i , r^i 称为该位的“权”。小数点左边各位的权是基数 r 的正次幂, 依次为 $0, 1, 2, \dots, m$ 次幂, 小数点右边各位的权是基数 r 的负次幂, 依次为 $-1, -2, \dots, -n$ 次幂。 r 进制数在计数过程中, 当它的某位计满 r 时就要向它临近的高位进 1。

一般用 $(\quad)_r$ 表示括号内的数为 r 进制数。将 r 进制数按权展开, 表达式为:

$$a_m \times r^m + a_{m-1} \times r^{m-1} + \cdots + a_1 \times r^1 + a_0 \times r^0 + a_{-1} \times r^{-1} + a_{-2} \times r^{-2} + \cdots + a_n \times r^{-n} = \sum_{i=-n}^m a_i \times r^i$$

由此看来我们可以用任意数制来表示数,但因长期以来形成的习惯和计算处理的方便,一般只用十进制、二进制、八进制和十六进制四种数制来表示数。

1. 十进制数

十进制数的基数 $r = 10$,逢十进位,符号集为 $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$,其权为: $\cdots, 10^2, 10^1, 10^0, 10^{-1}, 10^{-2}, \cdots$ 对于十进制数,因人们已经习惯,一般不用括号和基数的方式来表示,如十进制数 $(123.456)_{10}$ 就写成 123.456。有时也用 D(Decimal)表示。

例 1-1 $(123.45)_{10} = 123.45 = 123.45D$

$$= 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0 + 4 \times 10^{-1} + 5 \times 10^{-2}$$

2. 二进制数

二进制数的基数 $r = 2$,符号集为 $\{0, 1\}$,其权为: $\cdots, 2^2, 2^1, 2^0, 2^{-1}, 2^{-2}, \cdots$ 常用 B(Binary)表示。

例 1-2 $(1011.011)_2 = 1011.011B$

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

3. 八进制数

八进制数的基数 $r = 8$,符号集为 $\{0, 1, 2, 3, 4, 5, 6, 7\}$,其权为: $\cdots, 8^2, 8^1, 8^0, 8^{-1}, 8^{-2}, \cdots$ 常用 O(Octadic)表示。

例 1-3 $(7016.235)_8 = 7016.235O$

$$= 7 \times 8^3 + 0 \times 8^2 + 1 \times 8^1 + 6 \times 8^0 + 2 \times 8^{-1} + 3 \times 8^{-2} + 5 \times 8^{-3}$$

4. 十六进制数

十六进制数的基数 $r = 16$,符号集为 $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$,其权为: $\cdots, 16^2, 16^1, 16^0, 16^{-1}, 16^{-2}, \cdots$ 常用 H(Hexadecimal)表示。

例 1-4 $(12AB.F9)_{16} = 12AB.F9H$

$$= 1 \times 16^3 + 2 \times 16^2 + 10 \times 16^1 + 11 \times 16^0 + 15 \times 16^{-1} + 9 \times 16^{-2}$$

十进制、二进制、八进制和十六进制之间的数码对照表如表 1-1 所示。

1.2.2 计算机的码制

在日常生活中,编码问题是经常会遇到的。例如电话号码、房间编号、学号等,这些编码问题的共同特点是采用十进制数字来编码的。而计算机只能识别 0 和 1 两种符号,但计算机处理的信息却有多种形式,如数字、文字符号和命令、图形等。要表示这么多各种各样的信息并且能识别它们,就必须对这些信息进行编码。即以上那些各种各样的信息只有按特定的规则用二进制编码后才能在计算机中表示和识别。

下面介绍几种计算机中常用的编码。

表 1-1 十进制、二进制、八进制和十六进制之间的数码对照表

整 数				小 数			
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	0000B	000	0H	0	0B	00	0H
1	0001B	010	1H	0.5	0.1B	0.40	0.8H
2	0010B	020	2H	0.25	0.01B	0.20	0.4H
3	0011B	030	3H	0.125	0.001B	0.10	0.2H
4	0100B	040	4H	0.0625	0.0001B	0.040	0.1H
5	0101B	050	5H	0.03125	0.00001B	0.020	0.08H
6	0110B	060	6H	0.015625	0.000001B	0.010	0.04H
7	0111B	070	7H	...	...	...	...
8	1000B	100	8H				
9	1001B	110	9H				
10	1010B	120	AH				
11	1011B	130	BH				
12	1100B	140	CH				
13	1101B	150	DH				
14	1110B	160	EH				
15	1111B	170	FH				

一、BCD 码

BCD(Binary Coded Decimal)码为用二进制表示的十进制数。BCD 码是为了满足人们使用十进制来计数,同时又满足计算机处理信息需要二进制编码的要求,而设计的一种编码。因为任何十进制数都是 0~9 这 10 个数字的组合,这样必须对十进制的 0~9 这 10 个数字进行二进制编码,即 BCD 码具有二进制和十进制两种数制编码的某些特征。

1.8421 BCD 码

BCD 码用 4 位二进制码来表示 0~9 的十进制数。一般采用 8421 方式编码,即组成的 4 位二进制数码的权分别为 8、4、2、1,编码 0~9 这 10 个数字的 8421 BCD 码的编码表如表 1-2 所示。

用 4 位二进制数字来完全编码共有 16 种组合,但 BCD 码只用了其中 0000B~1001B 的基本代码系统,而 1010B~1111B 则未被使用,这部分在 8421 BCD 码中为非法码或冗余码。10 以上的所有十进制数至少需要 2 位 BCD 码。因此,BCD 码数是由 BCD 码构成的,是以二进制形式出现的,是逢十进位的,即它不是一个真正的二进制数,因为二进制数是逢二进位的。例如:十进制数 45 的 BCD 码形式为 01000101B(即 45H),而它的等值二进制数为 00101101B(即 2DH)。

BCD 码具有转换方便,容易阅读的优点,但也具有位长较用纯二进制表示的十进制数更长的缺点,导致电路复杂性增加,运算速度也慢一些。

表 1-2 8421 BCD 码编码表

十进制数	8421 BCD 码	十进制数	8421 BCD 码
0	0000B	8	1000B
1	0001B	9	1001B
2	0010B	10	00010000B
3	0011B	11	00010001B
4	0100B	12	00010010B
5	0101B	13	00010011B
6	0110B	14	00010100B
7	0111B	15	00010101B

需要指出的是,用二进制编码表示十进制数的编码方式并不是只有 8421 编码一种,只是 8421 编码比较直观和符合人们的习惯。另外,还有 2421、5211、余 3 码等编码方式,因为 2421、5211 编码表示的十进制数不唯一,而余 3 码编码不太符合习惯,所以一般采用 8421 编码方式。

2. BCD 码加减法运算

所谓 BCD 码加法运算是指两个 BCD 码数按“逢十进位”原则进行加法运算,其和也是一个 BCD 码数。但因计算机只能进行二进制数加法运算,进位是“逢十六进位”,这样当两个 BCD 码数进行加法运算时在两个 BCD 码之间也只能按“逢十六进位”的原则进行,即不可能满足 BCD 码数按“逢十进位”原则进行加法运算的规则。因此,计算机在进行 BCD 码加法运算时,必须对因采用二进制加法的原则进行运算的结果进行调整,使两个紧邻的 BCD 码数之间真正能够做到“逢十进位”。

在进行 BCD 码数加法运算过程中,计算机对因采用二进制数加法运算原则的运算结果进行调整的原则是:若和的低 4 位大于 9 或低 4 位向高 4 位有进位,则低 4 位需加 6 调整;若和的高 4 位大于 9 或高 4 位向最高位有进位,则高 4 位需加 6 调整。在第三章将会学到这种调整的过程是通过特定的指令(DA A)由微处理器内部的十进制调整电路自动完成的。两个 BCD 码运算调整结果的具体示例将在第三章介绍十进制调整指令(DA A)时讨论。

同样对于两个 BCD 码数进行减法运算时,因相同的原因也要进行十进制调整。在进行 BCD 码数减法运算过程中,计算机对因采用二进制数减法运算原则的运算结果进行调整的原则是:若差的低 4 位大于 9 或低 4 位向高 4 位有借位,则低 4 位需减 6 调整;若差的高 4 位大于 9 或高 4 位向最高位有借位,则高 4 位需减 6 调整。

二、ASCII 码(字母数字代码)

计算机不仅要处理数字信息,而且还要处理大量字母、字母数字和符号信息,为此需

对这些信息进行二进制编码,以供计算机识别、存贮和处理。

ASCII 码(American Standard Code for Information Interchange),即美国信息交换标准码。ASCII 码为一种 8 位代码,但采用 7 位代码编码(一般其最高位为 0),共可编码为 128 个字符编码。其中 96 个为图形字符,32 个为控制字符。96 个图形字符中包括十进制数字字符(0~9)10 个、大小写英文字母 52 个及其他字符 34 个。这 96 个字符都可以在 CRT 上显示、通过打印机在打印纸上打印,其编码可以存贮、传送和处理。32 个控制字符包括回车符、换行符、退格符、设备控制符和信息分隔符等,这类字符没有特定形状,其编码可以存贮、传送和起某种控制作用,但这些字符本身是不能在 CRT 上显示和通过打印机在打印纸上打印的。

ASCII 码表见附录 D。

我国于 1980 年制订了“信息处理交换用的 7 位编码字符集”,即国家标准 GB1988 - 80。其中除了用人民币符号 ¥ 代替美元符号 \$ 外,其余代码与含义都和 ASCII 码相同。

1.2.3 各数制间的转换

计算机是采用二进制数操作的,但人们已习惯了使用十进制数,这就要求常用的几种不同数制之间能相互转换。

一、进制数 $\leftrightarrow$ 十进制数

1. r 进制数 $\rightarrow$ 十进制数

将 r 进制数转换成十进制数只要将 r 进制数中出现非 0 的位按权展开相加即可。例如:

$$11010.01B = 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^1 + 1 \times 2^{-2} = 26.25$$

又例如:

$$\begin{aligned} 3FEA.1CH &= 3 \times 16^3 + 15 \times 16^2 + 14 \times 16^1 + 10 \times 16^0 + 1 \times 16^{-1} + 12 \times 16^{-2} \\ &= 16362.109375 \end{aligned}$$

2. 十进制数 $\rightarrow$ r 进制数

一个十进制数可以分成两个部分:即整数部分和小数部分。将其转换成 r 进制数需对整数部分和小数部分分开来转换,然后再合并起来。

对十进制整数部分转换成 r 进制数的方法一般是将十进制数整数部分依次除以所需要转换进制数的基数(r),并记录余数就能够转换成 r 进制的数。比如:为了将十进制数的整数部分转换成相应的二进制数,只要把十进制数的整数部分依次除以 2 并记下每次所得的余数(这种情况下余数总是为 1 或 0),所得的余数即为相应的二进制数的整数部分。这种方法称为余数法。

例 1-5 试求出十进制数 125 的二进制数。

解:采用余数法来求转换的结果。

2	215		余 1	最低位
2	107		余 1	
2	53		余 1	
2	26		余 0	
2	13		余 1	
2	6		余 0	
2	3		余 1	
2	1		余 1	最高位

把所得的余数按箭头方向从高到低排列起来便可得到:

$$215D = 11010111B$$

例 1-6 试求出十进制数 125 的十六进制数。

解:采用余数法来求转换的结果。

16	215		余 7	最低位
16	13		余 D	最高位

把所得的余数按箭头方向从高到低排列起来便可得到:

$$215D = 0D7H \text{ (注意:十六进制数的最高位如果大于等于 10,则在其前面添 0)}$$

对十进制小数部分转换成 r 进制数的方法一般是将十进制数小数部分依次乘以所需要转换进制数的基数(r),并记录溢出数(即整数部分)就能够转换成 r 进制的数。比如:为了将十进制数的小数部分转换成相应的二进制数,只要把十进制数的小数部分依次乘以 2 并记下每次所得的整数(这种情况下整数总是为 1 或 0),所得的整数即为相应的二进制数。这种方法称为取整法。

例 1-7 试把十进制小数 0.3125 转换为二进制小数。

解:采用取整法来求转换的结果。

0.3125	
$\times \quad 2$	
0.625	取得整数为 0 最高位
$\times \quad 2$	
1.250	取得整数为 1
0.250	
$\times \quad 2$	
0.500	取得整数为 0
$\times \quad 2$	
1.000	取得整数为 1 最低位

把所得的整数按箭头方向从高到低排列起来便可得到:

$$0.3125D = 0.0101B$$

例 1-8 试把十进制小数 0.3125 转换为八进制小数。