

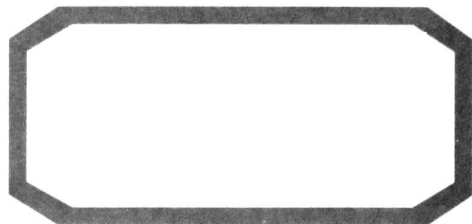
21世纪高等院校规划教材

单片机原理 及应用

主编 杨恢先 黄辉先

湘潭大学出版社

21世纪高等院校规划教材



单片机原理 及应用

主 编 杨恢先 黄辉先

副主编 欧青立 戴 永 何凤庭

参编人员（按姓氏笔画）

王仕果	王求真	尹艳群	李 卫
李玉声	李志斌	朱 江	张东波
宋 芳	肖登辉	冷爱莲	赵延明
范小春	谭乔来		

图书在版编目(CIP)数据

单片机原理及应用 / 杨恢先, 黄辉先主编. — 湘潭:
湘潭大学出版社, 2013.1

21 世纪高等院校规划教材

ISBN 978-7-81128-472-0

I. ①单… II. ①杨… ②黄… III. ①单片微型计算
机—高等学校—教材 IV. ①TP368.1

中国版本图书馆 CIP 数据核字 (2013) 第 026573 号

责任编辑: 丁立松

封面设计: 刘 扬

出版发行: 湘潭大学出版社

社 址: 湖南省湘潭市 湘潭大学出版大楼

电话(传真): 0731-58298966 邮编: 411105

网 址: <http://xtup.xtu.edu.cn>

印 刷: 湘潭市风帆印务有限公司

经 销: 湖南省新华书店

开 本: 787×1092 1/16

印 张: 19.5

字 数: 487 千字

版 次: 2013 年 1 月第 1 版 2013 年 1 月第 1 次印刷

书 号: ISBN 978-7-81128-472-0

定 价: 38.00 元

(版权所有 严禁翻印)

前 言

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,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。

单片机由于具有功能强、体积小、可靠性好和价格便宜等优点,已成为传统工业技术改造和新产品研发与更新的理想机种,发展前景广阔。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显著提高。可以说,当今世界正面临着一场以单片机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。编者特地结合自身的教学和科研实践编写了本书,奉献给广大读者。

本书的编写可以追溯到 2003 年,当时由国防科技大学出版社出版,后经较大修改后于 2007 年由人民邮电出版社出版。考虑到近年来串行外设的速度和精度上达到甚至超过同类并行接口芯片,而串行外设占用 I/O 口线和抗干扰能力等方面比并行外设更具优势,因而近年来单片机应用系统越来越多地采用串行外设接口技术,同时考虑到工程技术人员一般采用 C 语言编写单片机应用系统程序,为此我们在 2007 年版的基础上再次做了较大修改后重新出版,期望能得到广大读者的认可。

全书共分 9 章,全面翔实地论述了 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 1 章为单片机概述,简单介绍了单片机的发展过程及应用领域;第 2 章至第 6 章系统地介绍了 MCS-51 单片机的系统结构、指令系统与汇编语言程序设计、C51 程序设计、内部硬件资源及基本应用、系统基本扩展技术;第 7 章介绍了串行外设接口技术;第 8 章简单介绍了单片机应用系统的基本设计方法;第 9 章简单介绍了与 MCS-51 单片机兼容的常用单片机。

本书由杨恢先、黄辉先担任主编,欧青立、戴永、何凤庭担任副主编。宋斌、翟云龙、蔡勇勇、吴美勇、周杰文、任昆、喻世东、李璇等整理了部分文字和插图,编写和调试了部分程序。本书在编写过程中得到了湘潭大学教务处、湘潭大学信息工程学院、湘潭大学材料与光电物理学院、湖南科技大学信息与电气工程学院领导的支持,在此表示感谢。

限于编者水平,书中错误和不妥之处在所难免,敬请读者批评指正。

编 者

2013 年 1 月

目 录

第 1 章 单片微型计算机概述及预备知识

1.1 单片机的发展历史和发展趋势	(1)
1.2 计算机的数制和码制及数制之间的转换	(3)
1.3 计算机中数的表示方法	(9)
1.4 微型计算机的二进制数运算	(12)
1.5 常用的单片机产品	(15)
1.6 单片机的应用与选择	(16)
习 题	(17)

第 2 章 MCS—51 系列单片机硬件结构

2.1 逻辑结构及信号引脚	(18)
2.2 时钟电路与时序	(22)
2.3 工作方式	(26)
2.4 内部存储器	(29)
习 题	(35)

第 3 章 MCS—51 指令系统与汇编语言程序设计

3.1 汇编语言指令格式	(36)
3.2 寻址方式	(40)
3.3 指令系统	(44)
3.4 汇编语言程序设计方法	(64)
3.5 简单和分支程序设计	(65)
3.6 循环和查表程序设计	(70)
习 题	(76)

第 4 章 MCS—51 单片机的 C 语言简介

4.1 MCS—51 的 C 语言	(79)
4.2 C51 数据类型与存储方式	(80)
4.3 C51 数据的存储类型与存储关系	(82)
4.4 特殊功能寄存器(SFR)及其 C51 定义方法	(85)
4.5 并行接口及其 C51 定义方法	(86)

4.6 位变量及其 C51 定义方法	(87)
4.7 C51 构造数据类型	(88)
4.8 C51 的绝对地址访问	(90)
4.9 C51 的输入输出	(92)
4.10 C51 中断服务函数的定义方法	(94)
4.11 汇编语言与 C51 的混合编程	(95)
习 题	(97)

第 5 章 MCS-51 单片机的硬件资源及应用

5.1 并行 I/O 口及其应用	(98)
5.2 中断系统	(103)
5.3 定时器/计数器	(113)
5.4 串行接口及其串行通信	(121)
习 题	(144)

第 6 章 MCS-51 单片机的并行扩展接口

6.1 总线并行扩展	(145)
6.2 存储器接口扩展	(149)
6.3 输入/输出并行接口扩展	(153)
6.4 D/A 转换器接口扩展	(171)
6.5 A/D 转换器接口扩展	(177)
6.6 显示及键盘接口扩展	(186)
6.7 微型打印机接口扩展	(200)
习 题	(208)

第 7 章 MCS-51 单片机的串行外设接口技术

7.1 SPI 串行外设接口技术	(209)
7.2 IIC 总线外设接口技术	(215)
7.3 MicroWire 串行外设接口	(227)
7.4 1-Wire 单总线的基本原理	(232)
习 题	(241)

第 8 章 单片机应用系统的设计及应用举例

8.1 单片机应用条件	(242)
8.2 单片机应用系统设计要领	(243)
8.3 应用系统的设计过程	(244)
8.4 单片机应用系统调试	(255)
8.5 单片机应用系统设计与调试实例	(260)
习 题	(264)

第 9 章 MCS—51 兼容单片机简介

9.1	Atmel 公司 AT89、AT90 和 AT91 系列单片机简介	(265)
9.2	AT89C5X/LV5X 单片机原理	(268)
9.3	AT89CX051 单片机原理	(274)
9.4	Philips 公司 MCS—51 兼容单片机简介	(278)
9.5	Dalls—Maxim 公司 MCS—51 兼容单片机简介	(280)
9.6	其他公司的单片机	(283)
附录 A	MCS—51 指令表	(286)
附录 B	MCS—51 指令助记符注释表	(291)
附录 C	常用的缩写符号	(293)
附录 D	ASCII(美国信息交换标准码)表	(294)
附录 E	C51 的库函数	(295)
附录 F	常用集成电路引脚图	(298)
参考文献		(301)

第 1 章 单片微型计算机概述及预备知识

自 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 4 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、存储器、输入/输出(I/O)接口 4 个组成部分集成在一个硅片内,于是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,一般需嵌入仪器设备内,故又称嵌入式微控制器(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。本书以在我国应用较早、占有较大市场份额的 Intel 公司生产的 MCS-51 系列单片机为例来介绍单片机原理及应用。

1.1 单片机的发展历史和发展趋势

1.1.1 单片机发展历史

单片机的发展可以分为 4 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,随机存取存储器)和两个并行口,需要加一块具有 1 KB 的 ROM(Read Only Memory,只读存储器)、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 4 KB。

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 64 KB。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在工业检测、控制应用领域的主角。

1.1.2 单片机发展趋势

单片机的发展趋势是:向高性能化、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,尤其适合于作实时数字信号处理用。

(4) 串行总线结构。飞利浦公司开发了一种新型总线——IIC总线(Inter ICbus),该总线采用3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. 存储器的发展

(1) 增大存储容量。新型单片机片内ROM一般为4~8KB,有的甚至可达128KB。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EEPROM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EEPROM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EPROM,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EEPROM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EEPROM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EEPROM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EEPROM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,以满足多机通信等功能的要求。

(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

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,为适应检测、控制功能更高的要求,片内集成的部件还可有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重要趋势。

5. 低功耗化

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(5 V,12 MHz),工作电流为 16 mA;同样条件下在空闲方式工作时,工作电流为 3.7 mA;而在掉电方式(2 V)工作时,工作电流仅为 50 μ A。

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,比如现代人数数用的 10 进制;时钟计时用的 12 进制;用于早期衡器计量的 16 进制等。用于计算机的常用数制有二进制、八进制、十进制和十六进制 4 种。二进制是当前一般计算机所用的物理数制形式,其他数制均通过二进制的数据结构表达。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:

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

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,一般只用二进制、八进制、十进制和十六进制 4 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: $\dots,10^2,10^1,10^0,10^{-1},10^{-2},\dots$ 。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 $r = 8$, 其符号集为 $\{0, 1, 2, 3, 4, 5, 6, 7\}$, 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 $(12AB.F9)_8 = 12AB.F9H = 1 \times 16^3 + 2 \times 16^2 + 10 \times 16^1 + 11 \times 16^0 + 15 \times 16^{-1} + 9 \times 16^{-2}$

二进制、八进制、十进制和十六进制之间的数码对照表如表 1-1 所示。

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	00O	0H	0	0B	0O	0H
1	0001B	01O	1H	0.5	0.1B	0.4O	0.8H
2	0010B	02O	2H	0.25	0.01B	0.2O	0.4H
3	0011B	03O	3H	0.125	0.001B	0.1O	0.2H
4	0100B	04O	4H	0.0625	0.0001B	0.04O	0.1H
5	0101B	05O	5H	0.03125	0.00001B	0.02O	0.08H
6	0110B	06O	6H	0.015625	0.000001B	0.01O	0.04H
7	0111B	07O	7H	...	...	...	...
8	1000B	10O	8H				
9	1001B	11O	9H				
10	1010B	12O	AH				
11	1011B	13O	BH				
12	1100B	14O	CH				
13	1101B	15O	DH				
14	1110B	16O	EH				
15	1111B	17O	FH				

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

计算机处理的信息却有多种形式,例如: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。即上述各种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。

1. BCD 码

BCD(Binary Coded Decimal)码为用二进制表示的十进制数。BCD 码是为了满足人们利用计算机来进行十进制计数,同时又为能满足计算机处理信息需要二进制编码的需要,从而设计的一种编码。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 个数字符号进行二进制编码,即 BCD 码具有二进制和十进制两种数制编码的某些特征。

(1) 8421BCD 码

BCD 码用四位二进制码来表示 0~9 的十进制数。一般采用 8421 方式编码,即组成的四位二进制数码的权分别为 8、4、2、1,编码 0~9 这十个数字的 8421 BCD 码的编码表如表 1-2 所示。

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

用四位二进制数字来完全编码则共有 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 码中称为非法码或冗余码。10 以上的所有十进制数至少需要二位 BCD 码。因此,BCD 码数是由 BCD 码构成的,是以二进制形式出现的,每一位十进制数是用四位二进制数表示,当四位二进制数满 10,则向高位 BCD 码进位,即实现逢十进一,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。

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 2421、5211 编码表示的十进制数不唯一,而余 3 码编码不太符合编码习惯,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 3 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 3 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。

2. 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,为此需要对这些信息进行二进制编码,以供计算机识别、存储和处理。

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),故共可为 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符 0~9 共 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,其编码可以存储、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,其编码可以存储、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 码编码表见附录 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,其余代码和含义都与 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够相互转换。

1. r 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:

$11010.01B=1\times2^4+1\times2^3+1\times2^1+1\times2^{-2}=26.25$

又例如：

$3FEA.1CH=3\times16^3+15\times16^2+14\times16^1+10\times16^0+1\times16^{-1}+12\times16^{-2}=16362.109375$

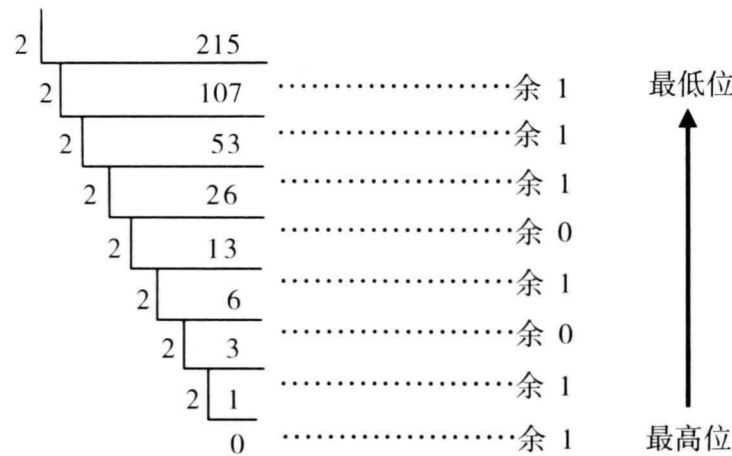
(2) 十进制数→r 进制数

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。例如：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)，所得的余数按规则排列起来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 215 的二进制数。

解：以下采用余数法来求转换的结果。

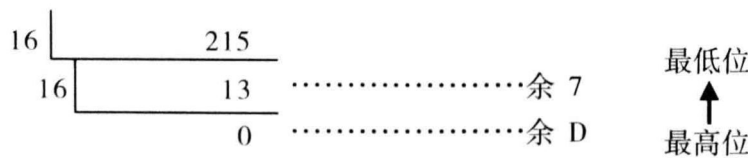


把所得的余数按箭头方向从高到低排列起来便可得到：

$215D=11010111B$

例 1—6 试求出十进制数 125 的十六进制数。

解：以下采用余数法来求转换的结果。



把所得的余数按箭头方向从高到低排列起来便可得到：

$125D=0D7H$

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。例如：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)，所得的整数按规则排列起来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:以下采用取整法来求转换的结果。

0.3125		
× 2	·····	取得整数为 0
0.625		
× 2	·····	取得整数为 1
1.250		
0.250		
× 2	·····	取得整数为 0
0.500		
× 2	·····	取得整数为 1
1.000		

最高位

↓

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。

0.3125		
× 8	·····	取得整数为 2
2.500		
0.500		
× 8	·····	取得整数为 4
4.000		

最高位

↓

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 = 0.24O=0.5H

如果一个十进制数既存在整数部分又存在小数部分,则将两部分分别转换后合并即可。

例 1-9 试求出十进制数 215.3125 的十六进制数。

解:将十进制整数部分 215 和小数部分 0.3125 分别转换成十六进制数后得:

215.3125=0D7.5H

应当指出:任何十进制整数都可以精确转换成一个二进制整数,但十进制小数却不一定可以精确转换成一个二进制小数。

2. 十六进制数↔二进制数

十六进制是计算机经常使用的一种数制,如指令机器码都是用十六进制表示的。部分十进制数和十六进制数之间的对应关系如表 1-1 所示。

(1) 二进制数→十六进制数

二进制转换成十六进制一般采用“四位合一位法”,即从二进制的小数点位置开始,左右两边每四位一组,不足四位以 0 补足,然后分别把每组用十六进制数码表示,并按序相连即可。

例 1-10 试将二进制数 1101111100011.100101B 转换为十六进制数。

解: 0001101111100011.10010100

□	□	□	□	□	□
---	---	---	---	---	---

则得

1101111100011.100101B=1BE3.94H

(2) 十六进制数→二进制数

将十六进制数的每位分别用四位二进制数码表示,然后把它们连成一体。

例 1—11 试将十六进制数 3AB.7A5H 转换为二进制数。

解: 3 A B . 7 A 5
 | | | | | |
 0011 1010 1011 0111 1010 0101

则得

3AB.7A5H=1110101011.011110100101B

1.3 计算机中数的表示方法

1.3.1 真值和机器数

一个数是由符号和数值两部分组成的。例如：

$N_1 = +1001010(+74)$

$N_2 = -1001010(-74)$

在计算机中数的符号也是用二进制码表示的，一般正数的符号用“0”表示，负数的符号用“1”表示。例如：

$N_1 = 01001010(+74)$

$N_2 = 11001010(-74)$

一个数在机器中的表示形式称为机器数，而把这个数本身称为真值。

1.3.2 带符号数的表示方法

在机器数表示方法中，用“0”表示正，用“1”表示负。这种表示数的方法，称为带符号数的表示方法。在机器中的一般表示形式为：

D_{n-1}	D_{n-2}	$D_{n-3} \cdots$	D_0
符号位	数值部分		

机器数最高位为符号位，其余的 $(n-1)$ 位为数值部分。例如：11001010B 有符号数为-74。

1.3.3 无符号数的表示方法

无符号数没有符号位，机器的全部有效位都用来表示数的大小。无符号数在机器中的一般形式为：

D_{n-1}	D_{n-2}	$D_{n-3} \cdots$	D_0
数值部分			

例如：11001010B 无符号数为 202。

1.3.4 数的定点和浮点表示

十进制数 485.23 也可以表示为 0.48523×10^3 ，还可表示为 0.048523×10^4 ，而在计算机内也有类似的两种数的表示方法，那就是定点数和浮点数。

1. 定点表示法

计算机内的定点数格式为：

符号位	数值部分	或	符号位	数值部分
-----	------	---	-----	------

小数点固定在数值部分的最高位之前或最低位之后。

2. 浮点表示法

浮点表示法即指小数点的位置是不固定的，而是浮动的。例如： $N_1 = 2^1 \times 0.1011$ 和 $N_2 = 2^3 \times 0.1011$ ，这两个数的有效数字相同，但小数点位置不一样。对于任何一个二进制数 N 都可以表示为：

$$N = \pm m \times 2^{\pm e}$$

其中， $m \geq 0$ 称为 N 的尾数； m 前面的符号称为数符； e 称为 N 的阶码，为非负整数，其前面的符号称为阶符（阶码和阶符决定 N 的小数点位置）。

计算机内浮点数格式为：

阶符	阶码	数符	尾数
----	----	----	----

由于一个数的浮点表示不是唯一的，为了使数据的有效位数最大，并使运算的精度尽可能高，计算机的浮点数采用规格化浮点数表示。规格化浮点数定义如下：

若 $N = \pm m \times 2^{\pm e}$ ，则

$$\frac{1}{2} \leq m < 1$$

1.3.5 原码、补码和反码

原码、补码和反码都是带符号数在机器中的表示方法。在介绍这 3 种编码方法之前，先介绍模的概念和性质。

我们把一个计量器的容量，称为模或模数，记为 M 或 $\text{mod } M$ 。例如：一个 n 位二进制计数器，它的容量为 2^n ，所以它的模为 2^n （即可表示 2^n 个不同的数）；又如：时钟可表示 12 个钟点，它的模为 12。

模具有这样的性质，当模为 2^n 时， 2^n 和 0 的表示形式是相同的。例如：一个 n 位二进制计数器，可以从 0 计数到 2^{n-1} ，如果再加 1，计数器就变成了零。所以， 2^n 和 0 在 n 位计数器中的表示形式是一样的。同样，时钟的 0 点和 12 点在钟表上的表示形式是相同的。

1. 原码

前面介绍的带符号数在机器中的表示方法，实际上就是原码表示法。原码表示方法是最简单的一种表示方法，只要把真值的符号部分用 0 或 1 表示即可。例如：

$$N_1 = +1001010$$
$$N_2 = -1001010$$

其原码记为：

$$[N_1]_{\text{原}} = 01001010$$
$$[N_2]_{\text{原}} = 11001010$$

由上述原码的表示形式，可将原码定义为：