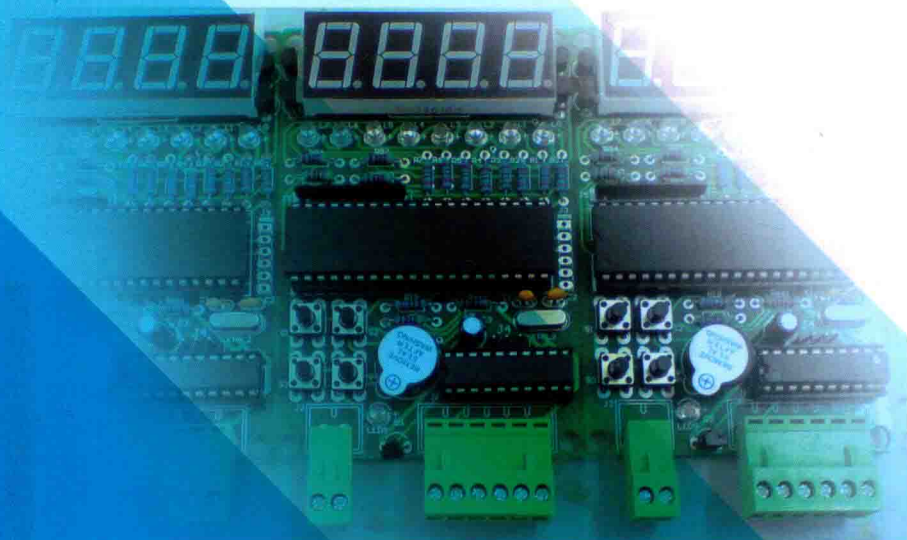




普通高等教育“十三五”规划教材

MCS-51单片机基础及其 在材料加工中的应用

徐向前 周好斌 主编



中国石化出版社

[HTTP://WWW.SINOPEC-PRESS.COM](http://www.sinopec-press.com)

普通高等教育“十三五”规划教材

MCS-51 单片机基础及其 在材料加工中的应用

徐向前 周好斌 主编

中国石化出版社

内 容 提 要

本书系统介绍了 MCS-51 单片机的基本原理及其在材料加工中的基本应用。单片机的基本原理内容涵盖单片机的结构、组成原理;指令系统和汇编程序的编写;C 语言的开发、编译和应用;单片机的接口(AD/DA、键盘与显示)应用及主要功能部件(定时计数器、中断)的应用。单片机在材料加工中的基本应用是根据单片机的特点、原理、应用方法,给出了其在材料加工中的具体应用示例,具有较强的实用性和指导性。

本书可以作为高等理工院校非计算机专业微机原理及接口技术课程的教材,也可以供工程技术人员参考。

图书在版编目(CIP)数据

MCS-51 单片机基础及其在材料加工中的应用 / 徐向前,
周好斌主编. —北京:中国石化出版社, 2017. 9
ISBN 978-7-5114-4655-8

I. ①M… II. ①徐…②周… III. ①单片微型计算机
IV. ①TP368.1

中国版本图书馆 CIP 数据核字(2017)第 223466 号

未经本社书面授权,本书任何部分不得被复制、抄袭,或者以任何形式或任何方式传播。版权所有,侵权必究

中国石化出版社出版发行

地址:北京市朝阳区吉市口路 9 号

邮编:100020 电话:(010)59964500

发行部电话:(010)59964526

<http://www.sinopec-press.com>

E-mail:press@sinopec.com

北京科信印刷有限公司印刷

全国各地新华书店经销

*

787×1092 毫米 16 开本 11.25 印张 266 千字
2017 年 9 月第 1 版 2017 年 9 月第 1 次印刷
定价:28.00 元

前 言

单片机具有结构简单、功能强、集成度高、容易掌握、应用灵活和价格低的优点，在工业控制、机电一体化、智能仪表、汽车工业、材料成型技术、通信技术、家用电器等诸多领域中有着广泛的应用。单片机的应用使得机电设备的技术水平和自动化程度有了大幅度的提高。因此高等理工科院校的师生和工程技术人员掌握或者了解单片机的原理和应用技术显得十分必要。

针对非计算机专业的师生和工程技术人员的特点，作者结合从事单片机原理及应用教学和科研体会，参考了大量相关书刊、资料，编写了本书。其主要目的是将单片机的基本原理和应用方法介绍给读者。

单片机具有完整的计算机结构，并且现在各种机型越来越多，功能越来越强，应用方式越来越灵活。本书主要讲述 MCS-51 单片机的基本原理和在材料加工中的基本应用。单片机的基本原理内容包括单片机的结构、组成原理，指令系统和汇编程序的编写，单片机的接口技术，定时计数器、中断的应用以及 C 语言的开发、编译和应用。结合作者的科研体会，将单片机在材料加工中的基本应用引入，包括单片机系统设计方法，编程语言的应用以及编译和调试工具等。根据现在的工程应用，在内容上，力求从易到难、循序渐进；在方法上，从基本原理入手，结合应用和设计进行编写。

本书注重单片机编程语言应用内容的介绍，包括汇编语言和 C 语言。C 语言已成为目前单片机编程语言的主要高级语言之一，而使用汇编语言编程的工程技术人员逐年减少。因此，本书在介绍了基本的汇编指令系统后，进行了 C 语言的应用情况叙述。这样，在理解单片机的结构和指令后，应用 C 语言编程，贴近了工程实际，具有了一定的应用能力。单片机基础部分的应用范例中采用了汇编和 C 语言两种语言编写的程序。这样在理解汇编语言的基础上，也学会了 C 语言的程序，为今后的学习和工作打下一定基础。同时给出了工程应用的实例，特别是在材料加工中的应用。

本书第 1 章至第 6 章由徐向前编写，第 7 章和第 8 章由周好斌编写。在编写过程中，参考了有关教材、期刊、资料，在此表示感谢！

目 录

1 单片机概述	(1)
1.1 单片机的概念	(1)
1.2 单片机的发展历史及趋势	(1)
1.3 数值基础	(2)
1.3.1 十进制、二进制和十六进制数	(2)
1.3.2 数的定点与浮点表示	(5)
1.3.3 机器数的表示	(7)
1.3.4 计算机中常用的编码	(8)
思考题	(9)
2 8051 单片机的结构和原理	(10)
2.1 8051 单片机内部结构及特点	(10)
2.1.1 基本组成	(10)
2.1.2 内部结构	(11)
2.2 8051 单片机引脚及功能	(13)
2.3 存储器结构和配置	(15)
2.3.1 程序存储器空间	(16)
2.3.2 数据存储器空间	(16)
2.3.3 专用寄存器	(18)
2.4 8051 单片机的并行 I/O 接口	(21)
2.4.1 P0 口结构及功能	(22)
2.4.2 P1 口结构及功能	(23)
2.4.3 P2 口结构及功能	(23)
2.4.4 P3 口结构及功能	(25)
2.5 8051 单片机时钟电路与时序	(26)
2.5.1 片内时钟信号	(26)
2.5.2 CPU 时序	(27)
2.5.3 复位电路	(29)
思考题	(31)

3 MCS-51 单片机指令系统	(32)
3.1 单片机指令系统概述	(32)
3.1.1 指令分类	(32)
3.1.2 寻址方式	(33)
3.2 8051 单片机指令系统	(35)
3.2.1 数据传送指令	(36)
3.2.2 算术运算指令	(40)
3.2.3 逻辑运算指令	(44)
3.2.4 控制转移指令	(46)
3.2.5 位处理指令	(52)
3.2.6 伪指令	(54)
思考题	(55)
4 单片机的中断和定时/计数器	(57)
4.1 中断的概念	(57)
4.2 8051 单片机系统中断结构和中断控制	(58)
4.2.1 中断源	(58)
4.2.2 中断控制寄存器	(60)
4.3 中断响应及处理过程	(64)
4.3.1 中断响应条件	(66)
4.3.2 中断响应过程	(66)
4.3.3 中断响应时间	(68)
4.4 8051 单片机中断应用案例	(69)
4.5 8051 单片机定时器/计数器	(72)
4.5.1 定时器/计数器概述	(73)
4.5.2 定时器/计数器的方式控制	(74)
4.5.3 定时器/计数器的 4 种工作方式	(75)
4.5.4 关于计数初始值的计算	(82)
思考题	(85)
5 单片机的串行口	(86)
5.1 串行通讯基本知识	(86)
5.1.1 同步通讯和异步通讯	(87)
5.1.2 单工、半双工和全双工通讯	(89)

5.1.3	编码方式	(89)
5.1.4	波特率	(90)
5.2	8051 单片机的串行口结构	(90)
5.2.1	内部硬件结构	(90)
5.2.2	串行通讯控制寄存器	(91)
5.2.3	电源控制器	(93)
5.3	8051 单片机串行口的 4 种工作方式	(93)
5.3.1	串行口方式 0	(93)
5.3.2	串行口方式 1	(95)
5.3.3	串行口方式 2 和方式 3	(96)
5.4	波特率设定和计算	(97)
5.5	8051 单片机串口应用举例	(99)
	思考题	(103)
6	单片机的扩展及接口技术	(104)
6.1	总线概念	(104)
6.2	常用存储器地址分配方法	(106)
6.2.1	线选法	(106)
6.2.2	译码法	(109)
6.3	8051 单片机程序、数据存储器的扩展	(110)
6.4	8255A 可编程接口芯片扩展 I/O 口	(114)
6.5	8051 单片机与 D/A、A/D 转换器的接口	(117)
6.5.1	单片机与 D/A 转换器的接口	(118)
6.5.2	单片机与 A/D 转换器的接口	(120)
	思考题	(123)
7	单片机应用系统设计方法	(124)
7.1	概述	(124)
7.2	8051 单片机应用系统设计	(124)
7.2.1	总体设计	(124)
7.2.2	硬件设计	(125)
7.2.3	软件设计	(126)
7.2.4	可靠性设计	(127)
7.2.5	单片机应用系统的调试、测试	(131)

7.3 C51 编程简介	(132)
7.3.1 8051 单片机 C51 语言简介	(132)
7.3.2 C51 的基本语法	(133)
7.3.3 C51 编译器	(133)
7.3.4 Keil C51 开发系统简介	(134)
7.3.5 C51 应用举例	(135)
7.4 单片机集成开发环境——Keilc 的使用指导	(137)
7.4.1 如何建立一个 C 项目	(137)
7.4.2 如何进行工程详细设置	(140)
7.5 实例——频率可调的方波信号发生器	(142)
7.5.1 系统设计	(143)
7.5.2 程序设计	(146)
8 单片机在材料加工控制中的应用实例	(150)
8.1 超薄不锈钢自动点焊设备	(150)
8.1.1 储能自动点焊设备机械结构	(150)
8.1.2 储能自动点焊设备设备的控制系统	(151)
8.1.3 试验结果	(155)
8.2 铝制散热器封头自动焊设备	(156)
8.2.1 设备控制系统总方案	(156)
8.2.2 控制系统硬件详细设计	(159)
8.2.3 控制系统软件详细设计	(163)
8.2.4 焊接工艺参数试验	(170)
参考文献	(172)



单片机概述

1.1 单片机的概念

无论规模大小、性能高低,计算机的硬件系统都是由运算器、存储器、输入设备、输出设备以及控制器等单元组成。在通用计算机中,这些单元被分成若干块独立的芯片,通过电路连接而构成一台完整的计算机。而单片机技术则将这些单元全部集成到一块集成电路中,即一块芯片就构成了一个完整的计算机系统。这成为当时这一类芯片的典型特征,因此,就以 Single Chip Microcomputer 来称呼这一类芯片,中文译为“单片机”。这在当时是一个准确的表达。但随着单片机技术的不断发展,“单片机”已无法确切地表达其内涵,国际上逐渐采用 MCU(Micro Controller Unit)来称呼这一类计算机,并成为单片机界公认的、最终统一的名词。但国内由于多年来一直使用“单片机”的称呼,已约定俗成。所以目前仍采用“单片机”这一名词。

单片微型计算机(Single Chip Microcomputer),简称单片机,即把 CPU、存储器(RAM、ROM)、中断系统、串并口接口电路和定时计数器集成在一块芯片上,组成计算机系统。

单片机在控制领域中,主要用于实现各种测试和控制功能,因此通常也称为微控制器 MCU(Micro Controller Unit)。但单片机作为嵌入方式使用时,通常作为核心芯片,因此也可以称为嵌入式微控制器 EMCU(Embedded Micro Controller Unit)。

单片机可以分为通用型和专用型两类。通用型单片机是一种基本芯片,片内资源较多,功能较全,能够适用不同用户的设计和应用。也就是,用户通过对单片机内部资源的了解,可以进行单片机最小应用系统的软硬件设计,从而组成控制系统,完成各种控制功能。

专用型单片机通常是针对某种特定产品进行设计和生产的,特点是针对性强,数量大。比如有些智能仪表读写器上用的单片机,就是如此。

1.2 单片机的发展历史及趋势

20 世纪 70 年代,美国仙童公司首先推出了第一款单片机 F-8,随后 Intel 公司推出了 MCS-48 单片机系列,其他一些公司如 Motorola、Zilog 等也先后推出了自己的单片机,取得了一定的成果,这是单片机的起步与探索阶段。总的来说,这一阶段的单片机性能较弱,属于低、中档产品。随着集成技术的提高以及 CMOS 技术的发展,单片机的性能也随之改善,高性能的 8 位单片机相继问世。1980 年 Intel 公司推出了 8 位高档 MCS-51 系列单片机,性能得到很大的提高,应用领域也大为扩展。这是单片机的完善阶段。1983 年 Intel 公司推出了 16 位 MCS-96 系列单片机,加入了更多的外围接口,如模/数转换器(ADC)、看门狗

(WDT)、脉宽调制器(PWM)等,其他一些公司也相继推出了各自的高性能单片机系统。随后许多用在高端单片机上的技术被下移到8位单片机上,这些单片机内部一般都有非常丰富的外围接口,强化了智能控制器的特征,这是8位单片机与16位单片机的推出阶段。

近年来,Intel、Motorola 等公司又先后推出了性能更为优越的32位单片机,单片机的应用达到了一个更新的层次。随着科学技术的进步,早期的8位中、低档单片机逐渐被淘汰。但8位单片机并没有消失,尤其是以8051为内核的单片机,不仅没有消失,还呈现快速发展的趋势。

Intel 公司在1976年推出了8位单片机 MCS-48 系列,当时以控制功能较全、价格较低,且体积较小等主要特点,得到了较为广泛的应用,也为单片机的发展打下了基础。在 MCS-48 系列单片机的成功应用的影响下,半导体芯片生产厂商开始研制和发展自己的单片机系列。在这个时期世界各地出现了50多个系列的单片机,300多个品种,其中较为有名的是 Motorola 公司的6801、6802 系列和 Zilog 公司的 Z-8 系列。日本的 NEC、日立等公司也研制出了各自的单片机。

我国应用较为广泛的单片机是 MCS-51 系列单片机。MCS-51 系列单片机的基础是 MCS-48,是20世纪80年代发展起来的。MCS-51 单片机虽然是8位单片机,但功能比 MCS-48 强很多,且品种较为齐全,兼容性好,软硬件资源较为丰富。很快成为了单片机的主流,直到现在 MCS-51 单片机仍然为主流单片机。Intel 公司,1983年推出的 MCS-96 系统的单片机是16位单片机的典型代表。

1.3 数值基础

在计算机中各种信息的表示和处理都有具体的数据形式。在单片机中通常采用二进制表示数据。所谓二进制形式,是指每位数码只取两个值,要么是“0”,要么是“1”,数码最大值只能是1,超过1就应向高位进位。

为什么要采用二进制形式呢?这是因为二进制最简单。它仅有两个数字符号,这就特别适合用电子元器件来表示。制造有两个稳定状态的元器件一般比制造具有多个稳定状态的元器件要容易得多。

计算机内部采用二进制表示各种数据,对于单片机而言。其主要的数据类型分为数值数据和逻辑数据两种。下面分别介绍数制概念和各种数据的机内表示、运算等知识。

按进位的原则进行计数,称为进位计数制,简称“数制”。数制有多种,在计算机中常使用的有十进制、二进制和十六进制。

1.3.1 十进制、二进制和十六进制数

人们最常用的数是十进制数,计算机中采用的是二进制数,有时为了简化二进制数据的书写,也采用八进制和十六进制表示方法。下面将分别介绍这几种常用的进制。

(1) 十进制数(Decimal System)

十进制数是大家熟悉的,用0、1、2、…、8、9十个不同的符号来表示数值,它采用的是“逢十进一,借一当十”的原则。每一位的位权都是以10为底的指数函数,由小数点向左,各数位按位权展开。依次为 10^0 、 10^1 、 10^2 、 10^3 等;由小数点向右,各数的位权分别为 10^{-1} 、 10^{-2} 、 10^{-3} 等。这里的 10^0 、 10^1 、 10^2 、 10^{-1} 、 10^{-2} 在数学上称为权。所使用的数码的个

数称为基。因此十进制数的基数是 10。“权”和“基”是进位计数制中的两个要素。

例如 526.23，如果按各数的位权来展开，可以写成下面的形式：

$$526.23 = 5 \times 10^2 + 2 \times 10^1 + 6 \times 10^0 + 2 \times 10^{-1} + 3 \times 10^{-2}$$

(2) 二进制数(Binary System)

基数为 10 的计数制叫十进制；基数为 2 的计数制叫作二进制。二进制数的计算规则是“逢二进一，借一当二”。二进制数的基为“2”，即其使用的数码为 0、1，共 2 个。二进制制的权是以 2 为底的幂。比如下面这个数：

二进制	1	1	0	1	1	1
	2^5	2^4	2^3	2^2	2^1	2^0
十进制	32	16	8	4	2	1

其各位的权为 1、2、4、8、16、32，即以 2 为底的 0 次幂、1 次幂、2 次幂等。

$$(110111)_2 = 1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = (55)_{10}$$

(3) 八进制(Octave System)

八进制数采用基数为 8 的计数制。八进制数主要采用 0、1、2、…、7 这 8 个阿拉伯数字。八进制数的运算规则为“逢八进一，借一当八”。八进制数制的权是以 8 为底的幂。比如下面这个数：

八进制	1	0	3	5	2	4
	8^5	8^4	8^3	8^2	8^1	8^0

$$(103524)_8 = 1 \times 8^5 + 0 \times 8^4 + 3 \times 8^3 + 5 \times 8^2 + 2 \times 8^1 + 4 \times 8^0 = (34644)_{10}$$

(4) 十六进制数(Hexadecimal System)

十六进制数采用基数为 16 的计数制。但只有 0~9 这十个数字，所以用 A、B、C、D、E、F 这 6 个字母来分别表示 10、11、12、13、14、15，字母不区分大小写。十六进制数的运算规则为“逢十六进一，借一当十六”。十六进制数制的权是以 16 为底的幂。比如下面这个数：

十六进制	4	A	0	7	F	1
	16^5	16^4	16^3	16^2	16^1	16^0

$$(4A07F1)_{16} = 4 \times 16^5 + A \times 16^4 + 0 \times 16^3 + 7 \times 16^2 + F \times 16^1 + 1 \times 16^0 = (4851679)_{10}$$

在数字后面加上(2)、(8)、(10)或者(16)是指二进制、八进制、十进制和十六进制。也有用字母表示这些数制的，B 代表二进制，H 代表十六进制，O 代表八进制，D 代表十进制。通常十进制数的 D 或者 10 可以省略不写。

(5) 进制间的转换

由于人们习惯于十进制数，而计算机内部采用的是二进制数，因此就产生了各种进制之间的转换规则。下面将分两方面来介绍一下各个进制之间的转换问题。

① 十进制数与二进制数之间的相互转换

十进制数转换成二进制 将十进制数转换成二进制数时，要把整数部分和小数部分分别进行转换，然后再把转换之后的结果相加。整数部分采用“除 2 取余”的方法，也就是只要

将它一次一次地被 2 除，直到商为 0 为止，得到的余数“自下而上”(从最后一个余数)读取，这就是二进制数的整数部分。小数部分采用“乘 2 取整”的方法，也就是将它一次一次地被 2 乘，取乘积的整数部分，再取其小数部分被 2 乘，直到小数部分为 0 结束，得到的整数“自上而下”读取，这就是二进制数的小数部分。

【例 1.1】 将 $(136)_{10}$ 转换为二进制数。

2	136	余数为 0
2	68	余数为 0
2	34	余数为 0
2	17	余数为 0
2	8	余数为 1
2	4	余数为 0
2	2	余数为 0
2	1	余数为 0
	0	余数为 1

转化结果为： $(136)_{10} = (10001000)_2$

【例 1.2】 将十进制数 0.625 转换成二进制数

0.625	
× 2	1
1.250	
× 2	0
0.500	
× 2	1
1.000	

转换结果为： $(0.625)_{10} = (0.101)_2$

需要说明的是，有的十进制小数不能精确地转换成二进制小数，这样乘积的小数部分就永远不能为 0，此时可以根据精度的要求，将它转换到所需的位数即可。

十进制数到二进制数的转换过程可以推广到十进制数和八进制数、十六进制数之间的转换。也就是将“除 2 取余”和“乘 2 取整”相应地转换为“除 M 取余”和“乘 M 取整”(M 即可代表 8 或 16)。

二进制数转换成十进制数 将二进制数转换成十进制数就相对简单点，可以将二进制数按“权”展开，相加即可。

【例 1.3】 将二进制数 11101.101 转换成为十进制数。

$$\begin{aligned}
 \text{解: } (11101.101)_2 &= 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} \\
 &= 16 + 8 + 4 + 0 + 1 + 0.5 + 0.25 + 0.125 \\
 &= (29.875)_{10}
 \end{aligned}$$

② 二进制数与八进制、十六进制之间的相互转换

计算机中采用的是二进制数，但二进制数的最大缺点就是显示和书写不方便。在实际应用中，人们经常把八进制数和十六进制数作为二进制数的辅助计数方式。

二进制数和八进制数互换 二进制数转换成八进制数时，只要从小数点位置开始，向左或向右每 3 位二进制划分为一组(不足 3 位时可补 0)，然后写出每一组二进制数所对应的八进制数码即可。而二进制数和十六进制数则是将 4 位二进制数作为一组对应十六进制数码，

进行相应的转换。表 1.1 给出了一组基本对应关系。

表 1.1 常用进制间对应关系

十进制数	二进制数	八进制数	十六进制数
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	13	D
14	1110	16	E
15	1111	17	F

【例 1.4】 将二进制数 $(11110101.111)_2$ 转换成八进制数：
011 110 101.111
3 6 5 7
即二进制数 $(11110101.111)_2$ 转换成八进制数是 $(365.7)_8$ 。反过来，将每位八进制数分别用 3 位二进制数表示，就可完成八进制数到二进制数的转换。

二进制数和十六进制数互换 二进制数转换成十六进制数时，只要从小数点位置开始，向左或向右每 4 位二进制划分为一组（不足 4 位时可补 0），然后写出每一组二进制数所对应的十六进制数码即可。

【例 1.5】 将二进制数 $(10110110110.0111)_2$ 转换成十六进制数：
0101 1011 0110.0111
5 B 6 7
即二进制数 $(10110110110.0111)_2$ 转换成十六进制数是 $(5B6.7)_{16}$ 。反过来，将每位十六进制数分别用 4 位二进制数表示，就可完成十六进制数和二进制数的转换。

八进制数、十六进制数和十进制数的转换：这三者转换时，可用二进制数作为媒介，先把待转换的数转换成二进制数，然后将二进制数转换成要求转换的数制形式。

1.3.2 数的定点与浮点表示

由于计算机所处理的数据（二进制数表示）可能既有整数部分，又有小数部分。这就提出了一个小数点位置如何表示的问题，所以就出现了数的定点表示和浮点表示方法。用定点

表示法表示的数就是定点数，而用浮点方法表示的数就是浮点数。

(1) 定点数表示法

定点表示法中约定机器中所有数据的小数点位置固定不变。一般采用两种简单的约定：定点整数和定点小数。

定点整数约定小数点在数值位的最低位之后，此时计算机中所表示的数一律为整数。计算机中的整数有正整数(也称不带符号的整数)和整数两大类(也称带符号的整数)。带符号的整数必须使用一个二进位作为其符号位，一般总是最高位(最左面的一位)，0表示“+”(正数)，1表示“-”(负数)。其余各位则用来表示数值的大小，例如： $00101011 = +43$ ； $10101011 = -43$ 。

负数反码：原码各位数取反。负数补码：原码各位数取反加1。正整数无论采用原码、反码还是补码表示，其编码都是相同的，并无区别(图 1.1)。



图 1.1

所有定点整数绝对值的表示范围： $1 \leq |X| \leq 2^n - 1$ 用定点表示法表示数据的机器称为定点计算机，定点计算机目前多采用定点小数的表示方法。

定点小数是用最高位表示符号，其他 $n-1$ 位二进制数表示数值部分，将小数点定在数值部分的最高位左边，因此任何一个小数可以表示为如图 1.2 所示的形式。

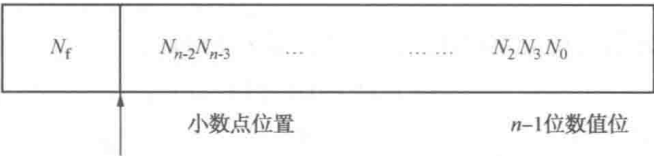


图 1.2

(2) 浮点数表示法

浮点数是指小数点在数据中的位置可以左右移动的数据。一个实数可以表示成一个纯小数和一个乘幂的积，例如： $56.725 = 10^2 \times (0.56725)$ 。其中指数部分用来指出实数中小数点的位置，括号括出的是一个纯小数。二进制数的情况完全类同。任意一个实数，在计算机内部都可以用指数(这是整数)和尾数(这是纯小数)来表示，这种用指数和尾数来表示实数的方法叫做浮点表示法。所以，在计算机中实数也叫作浮点数。而整数则叫作定点数。浮点数在机器中的表示格式如图 1.3 所示。

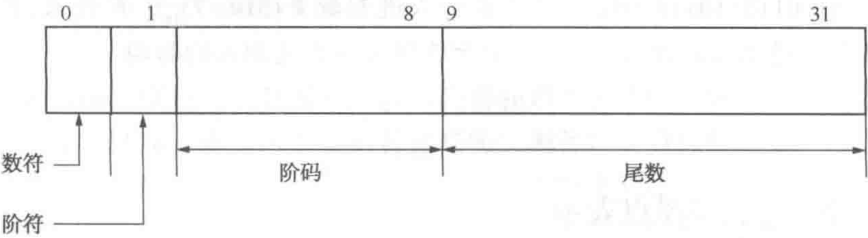


图 1.3

目前计算机系统对阶码多数采用移码表示，由于浮点数 N 的范围主要由阶码决定。阶

码相当于定点数中所取的比例因子，但它作为浮点数的一部分，说明小数点可以浮动。有效数的精度则主要由尾数决定。为了充分利用尾数的有效位数，一般采取规格化的办法，即让尾数的绝对值限定在一个范围内。如果阶码以 2 为底，则满足 $0.5 \leq |N| \leq 1$ ；如果尾数用补码表示，对正数规格化浮点数的尾数最高位等于 1；对负数规格化尾数为 0。因此，规格化的浮点数，其尾数最高位永远是符号位的反码。

1.3.3 机器数的表示

一个二进制数值数据，包括二进制表示的定点小数、定点整数和浮点小数。这里所指的表示方法是如何方便地表示正数、零和负数。常用的编码方式有三种：原码、反码、补码。

在日常生活中人们习惯用正负号加上绝对值来表示一个数的大小。数值数据在计算机中的编码表示称为机器数，而把机器数所代表的实际数值称为机器数的真值。

例如： $S = +0.110110$ 是真值， $S = 0.110110$ 为机器数

$D = -0.010011$ 是真值， $D = 1.010011$ 为机器数

前面提到的符号数的表示方法是一种最简单的表示方法，为原码表示法。除原码以外，还有补码和反码等表示方法。在介绍这些表示方法之前，先介绍模的概念与性质。

模是指一个计量系统的计数范围，计算机也可以看成是一个计量器，它也有一个计量范围，即都存在一个“模”。

例如：时钟的计量范围是 $0 \sim 11$ ，模 = 12。表示 n 位的计算机计量范围是 $0 \sim (2^n - 1)$ ，模 = 2^n 。

模实质上是计量器发生“溢出”的量，它的值在计量器上表示不出来，计量器上只能表示模的余数。任何有模的计量器，均可将减法转化为加法运算。

例如：设当前时针指向 10 点，而准确时间是 6 点，调整时间可有两种拨法。一种是倒拨 4 小时，即： $10 - 4 = 6$ ；另一种是顺拨 8 小时： $10 + 8 = 12 + 6 = 6$ 。在以 12 为模的系统中，加 8 和减 4 效果是一样的，因此凡是减 4 运算，都可以用加 8 来代替。

对模而言，8 和 4 互为补数。实际上以 12 为模的系统中，11 和 1、10 和 2、9 和 3、7 和 5、6 和 6 都有这个性质，共同的特点是两者相加等于模。对于计算机，其概念和方法完全一样。 n 位计算机，设 $n = 8$ ，所能表示的最大数是 11111111，若再加 1 变为 100000000 (9 位)，但因只有 8 位，最高位 1 自然丢失，又变成 00000000，所以 8 位二进制系统的模为 2^8 。在这样的系统中减法问题也可以化成加法问题，只需把减数用相应的补数表示就可以了。把补数用到计算机对数的处理上，就是补码。这个问题在讲到补码的时候再具体讲述。

下面分别以定点小数和定点整数为例给出三种编码的定义。

(1) 原码表示法

原码表示法中最高位表示符号，其中如果符号位为 0 表示该数为正，符号位为 1 则表示该数为负。

例如： $N_1 = +1001010$ $N_2 = -1001010$

其原码记为 $[N_1]_{\text{原}} = [+1001010]_{\text{原}} = 01001010$

$[N_2]_{\text{原}} = [-1001010]_{\text{原}} = 11001010$

0 的原码有两种表示形式，即 +0 和 -0：

$[+0]_{\text{原}} = 00000000$ $[-0]_{\text{原}} = 10000000$

原码的表示方法简单易懂，而且与真值转换方便，但是在做加法运算时就遇到了麻烦。当两个数相加时，如果是同号，则数值相加，符号不变；如果是异号，数值部分实际上是相

减,而且必须比较两个数哪个绝对值大,才能确定减数与被减数,这件事在手算时比较容易,而在计算机中这是一件繁琐的工作。为了便于计算机进行加减法运算,需要使用补码。

用二进制原码表示的数中,所用的二进制位数越多,所能表示的数的范围就越大。如8位二进制原码表示的范围是 $-128 \sim +127$;16位二进制原码表示的范围是 $-32768 \sim +32767$ 。

原码的特点:所有正数的原码,最左边的一位是0,负数是1。

例如: $X = +0.1011$ $[X]_{\text{原}} = 01011$

$X = -0.1011$ $[X]_{\text{原}} = 11011$

(2) 反码表示法

正数的反码和原码相同,负数的反码是保持负数原码的符号位不变,而其余各位取相反码即为机器数的反码的表示形式。反码表示法中最左边一位是符号位。当 X 为正数时,由于 $0 \leq X < 2^n$,反码表示法的最左边一位是0;当 X 为负数时, $2^n - 1$ 为 n 位1, $-2^n < X \leq 0$,其尾数不会大于 $n - 1$ 个1,此时反码表示法的最左边一位是1。反码也可以看作是以 $2^n - 1$ 为模的补码。

例如: $X_1 = +1010011$ $X_2 = -1010011$

则其反码记为 $[X_1]_{\text{反}} = [+1010011]_{\text{反}} = 01010011$

$[X_2]_{\text{反}} = [-1010011]_{\text{反}} = 10101100$

0的反码有两种表示形式,即+0和-0:

$[+0]_{\text{反}} = 00000000$ $[-0]_{\text{反}} = 11111111$

(3) 补码表示法

前面已经介绍过模的概念,在这里就不再叙述了。计算机的运算部件都有一定的字长限制,因此它的运算也是一种模运算。对于定点小数,可以在模为2的前提下,实现正负数数间的互补。

例如: $X = +100101$ $[X]_{\text{补}} = 0\ 100101$

$X = -100101$ $[X]_{\text{补}} = 1\ 011011$

补码的计算:

① 补码的和等于和的补码,符号位和数值位一样参加运算,不必单独处理,即

$$[X]_{\text{补}} + [Y]_{\text{补}} = [X + Y]_{\text{补}}$$

② 补码相减: $[X]_{\text{补}} - [Y]_{\text{补}} = [X]_{\text{补}} + [-Y]_{\text{补}}$, $[Y]_{\text{补}} \rightarrow [-Y]_{\text{补}}$,符号位连同数值位一起取反加1。

1.3.4 计算机中常用的编码

在计算机中,所有的信息都采用二进制表示,如大小写的英文字母、标点符号、运算符等,也必须采用二进制编码来表示,因为这样计算机才能进行识别,下面来了解一下计算机常用的两种编码。

(1) ASCII 码

人们需要计算机处理的信息除了数值外,还有字符或字符串。但在计算机中,所有信息都用二进制代码表示。为了在计算机中能够表示不同的字符,为使计算机使用的数据能共享和传递,必须对字符进行统一的编码。这样人们可以通过 n 位二进制代码来表示不同的字符,这些字符的不同组合就可表示不同的信息。常用的编码方式为美国标准信息交换码(American Standard Code for Information Interchange, ASCII),它是使用最广泛的一种编码。

基本的 ASCII 码有 128 个，每一个 ASCII 码与一个 8 位(bit)二进制数对应，其第 7 位是 0，称为基本的 ASCII 码，相应的十进制数是 0~127。如，数字“0”的编码用十进制数表示就是 48。另外 128 个扩展的 ASCII 码，最高位都是 1，用于表示一些图形符号，是扩展 ASCII 码。

(2) BCD 码

计算机中采用二进制数表示，但二进制不是很直观，所以在计算机的输入输出时通常用十进制数表示。不过这样的十进制数要采用二进制的编码来表示。这样的二进制数编码具有十进制数的特点，但形式上是二进制数。BCD 码是一种用 4 位二进制数字来表示 1 位十进制数字的编码，也称为二进制编码表示的十进制数(Binary Code Decimal, BCD)，简称 BCD 码。表 1.2 列出了十进制数 0~15 的 BCD 码。

表 1.2 BCD 编码表

十进制数	BCD 码	十进制数	BCD 码
0	0000	8	1000
1	0001	9	1001
2	0010	10	0001 0000
3	0011	11	0001 0001
4	0100	12	0001 0010
5	0101	13	0001 0011
6	0110	14	0001 0100
7	0111	15	0001 0101

BCD 码有两种格式：

- ① 压缩 BCD 码格式(Packed BCD Format)。用 4 个二进制位表示 1 个十进制位，就是用 0000B~1001B 来表示十进制数 0~9。例如：十进制数 4256 的压缩 BCD 码表示为 0100 0010 0101 0110。
- ② 非压缩 BCD 码格式(Unpacked BCD Format)。用 8 个二进制位表示 1 个十进制位，其中，高四位无意义，一般用××××表示，低四位和压缩 BCD 码相同。例如：十进制数 4256 的非压缩 BCD 码表示为××××0100××××0010××××0101××××0110。

 思考题

- 1. 单片机与一般微型计算机的结构上的主要区别是什么？
- 2. 什么叫原码、反码及补码？
- 3. 十进制、二进制、八进制和十六进制的概念及相互转化？
- 4. 计算机中常用的编码有哪些？