



21世纪高职高专系列教材

单片机原理与应用

中国机械工业教育协会 组编



机械工业出版社
China Machine Press

21 世纪高职高专系列教材

单片机原理与应用

中国机械工业教育协会 组编

主 编 厦门大学 陈文芄

副主编 金华职业技术学院 马汝星

参 编 天津理工学院职业技术学院 陈岚
厦门大学 郭光真 大连理工大学 徐志祥

主 审 江苏理工大学 赵德安



机械工业出版社

本书是根据高等职业技术教育(理工类)的教学要求来编写的。全书共分12章,前6章着重介绍 MCS—51 单片机的硬件结构和指令系统,第7章介绍常用的数据运算和处理程序,其中给出了较多的实用子程序,第10章和第11章介绍了单片机的人机对话界面。最后在第12章中介绍了 MCS—51 在智能仪器、仪表等方面的应用实例。

本书力求文字简练,通俗易懂。内容上重点突出,并采用了大量的设计实例作为例题。

本书可作为高等职业技术学院、高等院校专科、职工大学、业余大学、夜大学、函授大学、成人教育学院等大专层次相应课程的教材,也可作为广大自学者的自学用书。

图书在版编目(CIP)数据

单片机原理与应用/中国机械工业教育协会组编.

—北京:机械工业出版社,2001.8

21 世纪高职高专系列教材

ISBN 7-111-08398-9

I. 单... II. 中... III. 单片微型计算机—高等学校:技术学校—教材 IV. TP368.1

中国版本图书馆 CIP 数据核字(2001)第 042335 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

责任编辑:郑文斌 版式设计:张世琴 责任校对:申春香

封面设计:姚毅 责任印制:郭景龙

北京铭成印刷有限公司印刷·新华书店北京发行所发行

2001 年 8 月第 1 版·第 1 次印刷

787mm×1092mm $\frac{1}{16}$ ·12.5 印张·307 千字

0 001—4 000 册

定价:19.00 元

MAU56/06

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

本社购书热线电话(010)68993821、68326677-2527

21 世纪高职高专系列教材编委会名单

编委会主任 中国机械工业教育协会 郝广发

编委会副主任 (单位按笔画排)

山东工程学院 仪垂杰

大连理工大学 唐志宏

天津大学 周志刚

甘肃工业大学 路文江

江苏理工大学 杨继昌

成都航空职业技术学院 陈玉华

机械工业出版社 陈瑞藻 (常务)

沈阳工业大学 李荣德

河北工业大学 檀润华

武汉船舶职业技术学院 郭江平

金华职业技术学院 余党军

编委会委员 (单位按笔画排)

广东白云职业技术学院 谢瀚华

山东省职业技术教育师资培训中心 邹培明

上海电机技术高等专科学校 徐余法

天津中德职业技术学院 李大卫

天津理工学院职业技术学院 沙洪均

日照职业技术学院 李连业

北方交通大学职业技术学院 佟立本

辽宁工学院职业技术学院 李居参

包头职业技术学院 郑 刚

北京科技大学职业技术学院 马德青

北京建设职工大学 常 莲

北京海淀走读大学 成运花

江苏理工大学 吴向阳

合肥联合大学 杨久志

同济大学 孙 章

机械工业出版社 李超群 余茂祚 (常务)

沈阳建筑工程学院 王宝金

佳木斯大学职业技术学院 王跃国

河北工业大学 范顺成

哈尔滨理工大学工业技术学院 线恒录

洛阳大学 吴 锐

洛阳工学院职业技术学院 李德顺

南昌大学 肖玉梅

厦门大学 朱立秒

湖北工学院高等职业技术学院 吴振彪

彭城职业大学 陈嘉莉

燕山大学 刘德有

序

1999年6月中共中央国务院召开第三次全国教育工作会议,作出了“关于深化教育改革,全面推进素质教育的决定”的重大决策,强调教育在综合国力的形成中处于基础地位,坚持实施科教兴国的战略。决定中明确提出要大力发展高等职业教育,培养一大批具有必备的理论知识和较强的实践能力,适应生产、建设、管理、服务第一线急需的高等技术应用性专门人才。为此,教育部召开了关于加强高职高专教学工作会议,进一步明确了高职高专是以培养技术应用性专门人才为根本任务;以适应社会需要为目标;以培养技术应用能力为主线设计学生的知识、能力、素质结构和培养方案;以“应用”为主旨和特征来构建课程和教学内容体系;高职高专的专业设置要体现地区、行业经济和社会发展的需要,即用人的需求;教材可以“一纲多本”,形成有特色的高职高专教材系列。

“教书育人,教材先行”,教育离不开教材。为了贯彻中共中央国务院以及教育部关于高职高专人才培养目标及教材建设的总体要求,中国机械工业教育协会、机械工业出版社组织全国部分有高职高专教学经验的职业技术学院、普通高等学校编写了这套《21世纪高职高专系列教材》。教材首批80余本(书目附书后)已陆续出版发行。

本套教材是根据高中毕业3年制(总学时1600~1800)、兼顾2年制(总学时1100~1200)的高职高专教学计划需要编写的。在内容上突出了基础理论知识的应用和实践能力的培养。基础理论课以应用为目的,以必需、够用为度,以讲清概念、强化应用为重点;专业课加强了针对性和实用性,强化了实践教学。为了扩大使用面,在内容的取舍上也考虑到电大、职大、业大、函大等教育的教学、自学需要。

每类专业的教材在内容安排和体系上是有机联系、相互衔接的,但每本教材又有各自的独立性。因此各地区院校可根据自己的教学特点进行选择使用。

为了提高质量,真正编写出有显著特色的21世纪高职高专系列教材,组织编写队伍时,采取专门办高职的院校与办高职的普通高等院校相互协作编写并交叉审稿,以便实践教学和理论教学能相互渗透。

机械工业出版社是我国成立最早、规模最大的科技出版社之一,在教材编辑出版方面有雄厚的实力和丰富的经验,出版了一大批适用于全国研究生、大学本科、专科、中专、职工培训等各种层次的成套系列教材,在国内享有很高的声誉。我们相信这套教材也一定能成为具有我国特色的、适合21世纪高职高专教育特点的系列教材。

中国机械工业教育协会

前 言

本书是高等职业技术教育理工类教学用书。

作为高职教育用书,我们本着学以致用原则。全书的指导思想是突出应用,基本概念以讲清为度。因此本书在内容的选取及编写上,有如下特点:

1. 突出重点。对 MCS-51 单片机的结构,指令系统做了比较详细的介绍。并把定时器、串行接口及中断系统等几个重要内容单独成章,其目的是为了强化读者对单片机整体结构的认识,并使读者能够灵活的应用定时器、串行接口及单片机的中断功能。本书的另一个重点是 MCS-51 的指令系统,本书是按指令的功能归类介绍指令的。为了帮助读者理解指令的功能,每一条指令都附有例子以帮助理解。

2. 内容实用。编写中采用了大量的设计实例作为例题,目的是使读者能通过成功的实例学习单片机。书中的例子大多来自单片机的实践应用,这些例子可作为读者日后工作的参考。对于在软件设计中常用到的各种实用程序及其硬件接口和典型的设计方案,书中都做了详细的介绍。因此,读者学完本书后,可以把本书作为参考方案,完成自己的设计工作。

全书共分 12 章,前面 6 章着重介绍 MCS-51 单片机的硬件结构和指令系统,第 7 章介绍常用的数据运算和处理程序,给出较多的实用子程序。第 8 章、第 9 章介绍了单片机的扩展使用方法。第 10 章、第 11 章介绍了单片机的人机对话界面。最后在第 12 章中介绍了 MCS-51 在智能仪器、仪表等方面的应用实例。本书力求文字简练,通俗易懂。

本书参编人员为:

第 1 章 天津理工大学成教院 陈岚;第 2 章 金华职业技术学院理工学院 马汝星;第 3 章、第 4 章 厦门大学 郭光真;第 5 章、第 6 章 厦门大学 陈文芾;第 7 章、第 8 章 金华职业技术学院理工分院 马汝星;第 9 章、第 10 章 大连理工大学 徐志祥;第 11 章、第 12 章 天津理工大学成教院 陈岚。

本书由江苏理工大学赵德安教授主审。

限于作者的水平和时间限制,书中难免有疏漏和不妥之处,敬请读者批评指正,不胜感激。

编 者

目 录

序 前言

第1章 基础知识 1

1.1 计算机的数制 1

1.1.1 进位计数制 1

1.1.2 数制之间的转换 2

1.1.3 计算机中数的表示法 4

1.1.4 二进制数的运算 7

1.2 计算机的数据及编码 10

1.2.1 数据的单位及存储形式 10

1.2.2 数据的二进制编码 10

1.3 计算机基础知识 13

1.3.1 计算机硬件基础知识 13

1.3.2 计算机软件概述 14

复习思考题 15

第2章 单片机结构 17

2.1 单片机硬件结构特点 17

2.1.1 单片机分类 17

2.1.2 单片机的应用 17

2.1.3 单片机的内部结构 17

2.1.4 单片机的工作条件及安全保障 18

2.2 MCS-51 处理器 21

2.2.1 算术逻辑单元(ALU) 21

2.2.2 控制器 21

2.2.3 CPU 的时序 21

2.3 MCS-51 内部寄存器 23

2.3.1 数据存储器 23

2.3.2 程序存储器 25

2.3.3 特殊功能寄存器 25

2.4 MCS-51 的输出输入

结构(I/O 口) 27

2.4.1 P0 口 27

2.4.2 P1 口 28

2.4.3 P2 口 28

2.4.4 P3 口 29

2.5 MCS-51 的引脚功能 29

复习思考题 30

第3章 指令系统 31

3.1 概述 31

3.1.1 MCS-51 单片机助记符语言 31

3.1.2 MCS-51 单片机指令代码 33

3.1.3 常用符号说明 34

3.2 寻址方式 35

3.2.1 寄存器寻址 35

3.2.2 直接寻址 35

3.2.3 立即寻址 35

3.2.4 寄存器间接寻址 36

3.2.5 变址寻址 36

3.2.6 位寻址 36

3.2.7 相对寻址 37

3.3 数据传送指令 37

3.3.1 内部 RAM 低 128 字节的数据传送 37

3.3.2 特殊功能寄存器的数据传送 39

3.3.3 外部 RAM 的数据传送 39

3.3.4 从程序存储器读数据 40

3.3.5 数据交换指令 41

3.3.6 堆栈指令 41

3.3.7 不同存储空间之间的数据传送 42

3.4 算术运算指令 42

3.4.1 加法指令 42

3.4.2 减法指令 45

3.4.3 乘法指令 46

3.4.4 除法指令 47

3.5 逻辑运算指令 47

3.5.1 单操作数逻辑运算指令 47

3.5.2 双操作数逻辑运算指令 49

3.6 控制转移指令 51

3.6.1 无条件转移指令 51

3.6.2 条件转移指令 52

3.6.3 位条件转移指令	53	7.2 分支及循环程序的设计	87
3.6.4 子程序调用和返回指令	54	7.2.1 分支程序的设计	87
复习思考题	55	7.2.2 循环程序的设计	90
第4章 定时/计数器	57	7.3 延时程序的设计	98
4.1 定时/计数器工作原理	57	7.3.1 单循环延时程序	98
4.2 定时/计数器工作方式	57	7.3.2 较长时间的定时程序	98
4.2.1 有关的特殊功能寄存器	57	7.3.3 提高延时精度	98
4.2.2 定时/计数器的工作方式	58	7.4 查表程序	99
4.3 定时/计数器编程	60	7.5 码制转换程序	101
4.3.1 计数器初值的确定	60	7.5.1 整数十翻二	101
4.3.2 编程举例	61	7.5.2 整数二翻十	103
复习思考题	64	7.6 子程序设计	104
第5章 串行接口	65	7.6.1 子程序的概念	104
5.1 单片机的串行通信	65	7.6.2 子程序的设计	105
5.1.1 并行传送与串行传送	65	7.7 定时应用程序	107
5.1.2 串行通信的速率	66	7.8 中断程序	108
5.2 MCS-51 串行口的结构	68	7.8.1 中断控制程序	108
5.2.1 与串口有关的寄存器	68	7.8.2 中断服务程序	108
5.2.2 串行接口的工作方式	69	7.9 串行接口应用程序	110
5.2.3 MCS-51 串行口的应用	72	复习思考题	111
5.3 多CPU通信	75	第8章 存储器的扩展	113
复习思考题	77	8.1 存储器的地址空间分配	113
第6章 中断系统	78	8.2 ROM 的扩展	113
6.1 单片机的中断控制	78	8.2.1 只读存储器概述	113
6.1.1 中断请求源	79	8.2.2 程序存储器扩展原理及时序	114
6.1.2 中断允许	80	8.2.3 典型只读存储器	115
6.1.3 中断优先级	80	8.2.4 程序存储器扩展举例	118
6.2 中断响应	81	8.3 RAM 的扩展	119
6.2.1 响应中断的过程	81	8.3.1 数据存储器的扩展原理及时序	120
6.2.2 中断响应的的时间	82	8.3.2 典型随机存储器芯片	121
6.2.3 外部中断请求的持续时间	83	8.3.3 外部数据存储器扩展举例	122
6.2.4 中断申请的撤除	83	复习思考题	124
6.3 多外中断源系统设计	84	第9章 MCS-51 I/O 口的扩展	126
6.3.1 用软件查询方式扩展外部中断源	84	9.1 采用中小规模集成电路扩展	126
6.3.2 用优先编码器扩展外部中断源	85	9.1.1 简单 I/O 扩展	126
复习思考题	86	9.1.2 多片 I/O 扩展电路	128
第7章 程序设计	87	9.2 采用专用 I/O 接口芯片扩展	129
7.1 算术逻辑运算程序	87	9.2.1 用 8243 芯片扩展 I/O 口	129
		9.2.2 用 8255 芯片扩展 I/O 口	131

9.3 用串行口扩展 I/O 口	137	单片机接口电路	166
9.3.1 扩展并行输入口	137	11.1.6 A/D 转换程序设计	169
9.3.2 扩展并行输出口	139	11.2 D/A 接口	170
复习思考题	140	11.2.1 D/A 转换原理	170
第 10 章 单片机的人—机接口	141	11.2.2 D/A 转换器的主要技术指标	172
10.1 开关接口	141	11.2.3 集成 D/A 转换器及接口电路	173
10.1.1 拨键开关与单片机的接口	141	11.2.4 D/A 转换的典型应用举例	177
10.1.2 拨盘开关与单片机的接口	142	复习思考题	177
10.1.3 按钮与单片机的接口	143	第 12 章 单片机与外电路的	
10.2 显示器接口电路设计	144	控制接口	179
10.2.1 LED 显示器及接口电路设计	144	12.1 单片机中常用的隔离技术	179
10.2.2 LCD 液晶显示器及接口电路设计	149	12.1.1 磁隔离技术	179
10.3 键盘接口	156	12.1.2 光隔离技术	179
10.3.1 独立式按键电路	156	12.2 常用的外围驱动电路	180
10.3.2 矩阵式键盘电路	156	12.2.1 扩大驱动电流的方法	180
10.3.3 中断方式键盘电路	159	12.2.2 单片机控制强电电路的开关器件	182
10.3.4 串行接口非编码键盘电路	159	12.3 由 MCS-51 单片机控制的	
10.3.5 具有锁定功能的键盘电路	160	温度测控系统	185
10.3.6 双功能键电路设计	160	12.3.1 系统的构成和特点	185
复习思考题	162	12.3.2 键盘扫描及显示电路	185
第 11 章 A/D、D/A 接口	163	12.3.3 温度测量及模数转换电路	186
11.1 A/D 接口	163	12.3.4 温度控制功率调整电路	187
11.1.1 A/D 转换原理	163	12.3.5 主要元器件性能简介	187
11.1.2 A/D 转换的主要技术指标	164	12.3.6 编程要点	187
11.1.3 多通道 A/D 转换器 ADC0809 及其与 单片机接口电路	164	12.4 由 MCS-51 系列单片机控制的	
11.1.4 12 位 A/D 转换器 AD574 及其与 单片机接口电路	166	红外遥控装置	188
11.1.5 双积分型 A/D 转换器 MC14433 及其与		12.4.1 工作原理	188
		12.4.2 硬件电路	188
		12.4.3 软件设计	189

第1章 基础知识

1.1 计算机的数制

1.1.1 进位计数制

所谓数制，就是指数的制式，是人们利用符号计数的一种科学方法。一个数值，可以用不同进制的数表示。常用的数制有十进制、二进制、八进制和十六进制，计算机采用的是二进制。

为什么计算机进行数的计算时要用二进制数而不用十进制数呢？这是因为在计算机中的数是以电路的物理状态表示的，电路只有两个稳定状态：高电平与低电平或者导通与截止。这两个稳定状态可对应表示二进制的两个数 0 和 1。所以，计算机中的数是以二进制形式表示并进行运算的。

二进制数适用于计算机，但对于人们来说，二进制数书写繁琐，难以记忆。为了克服这个缺点，引进了八进制和十六进制。由于 $8 = 2^3$ ， $16 = 2^4$ ，一位八进制的数可表示 3 位二进制的数，一位十六进制的数则可表示 4 位二进制的数，例如：二进制数 11 011 010 = 332 八进制数，二进制数 1101 1010 = DA 十六进制数。八进制和十六进制常用作二进制的压缩形式，在程序设计中使用。十进制、二进制、八进制和十六进制的对应关系如表 1-1 所示。

表 1-1 二进制数与其他进制数对照表

十进制数	二进制数	十六进制数	八进制数
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	8	10
9	1001	9	11
10	1010	A	12
11	1011	B	13
12	1100	C	14
13	1101	D	15
14	1110	E	16
15	1111	F	17
16	0001 0000	10	20

1. 数制的基数和进制 每种计数制有一个固定的数 N ，在数制中，每一位都能取 N 个不同的数字，每一位都是逢 N 进一，这种计数制为 N 进制，我们把 N 称为基数。

(1) 十进制(Decimal)，基数为 10，共有 10 个数码：0~9，最大数码为基数 10 减 1，即 9，最小数码为 0，在计数过程中，当它的某位计满 10 时就向它邻近的高位进 1。

(2) 十六进制(Hexadecimal), 基数为 16, 共有 16 个数码: 0、1、2、…、9、A、B、C、D、E、F, 其中 A~F 六个数码代表十进制的 10~15, 计数规则是逢十六进一。

(3) 八进制(Octal), 基数为 8, 共有 8 个数码: 0~7, 最大数码为基数 8 减 1, 即 7, 最小数码为 0, 计数规则是逢八进一。

(4) 二进制(Binary), 基数为 2, 它有 2 个数码: 0、1, 计数规则是逢二进一。

在阅读和书写不同数制的数时, 如果不加注上标识符, 各种数制就容易混淆。通常采用两种方法标记: 一种是在数的右下脚标注数制代号, 如 $(1100)_2$, $(1100)_{10}$ 分别表示二进制和十进制的数; 另一种是在数的后面用大写字母 B、O、D、H 分别表示二进制、八进制、十进制和十六进制, 如: 1101B, 1101H, 1101D 等。十进制数可以不加标识符, D 标记或下脚标 10 省略。

2. 数制的位权 任何一个数制的多位数, 处在某一位上的“1”所表示的数值的大小称为该位的位权, 对于 N 进制数, 整数部分第 i 位的位权为 N^{i-1} , 小数部分第 j 位的位权为 N^{-j} 。例如:

(1) 十进制数, 各位的位权是以基数 10 为底的幂。

$$(491.28)_{10} = 4 \times 10^2 + 9 \times 10^1 + 1 \times 10^0 + 2 \times 10^{-1} + 8 \times 10^{-2}$$

式中, 指数 10^2 , 10^1 , 10^0 , 10^{-1} , 10^{-2} 称为该位的位权。10 为它的基数, 第 3 位的位权为 100, 第 2 位的位权为 10, 第 1 位的位权为 1。

(2) 二进制数, 各位的位权是以基数 2 为底的幂。

$$(1011.1)_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1}$$

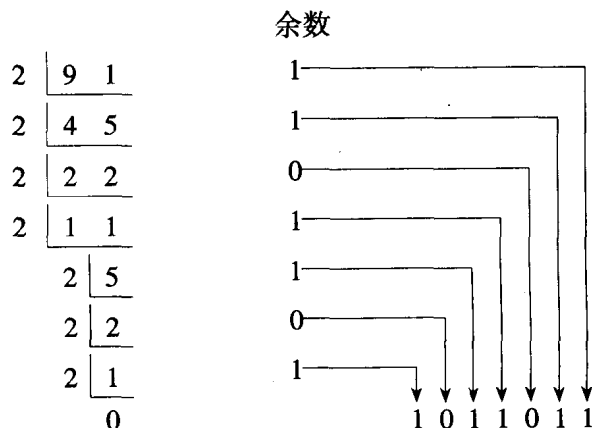
式中, 指数 2^3 , 2^2 , 2^1 , 2^0 , 2^{-1} 为该位的位权。2 为基数, 第 4 位的位权为 8, 第 3 位的位权为 4, 第 2 位的位权为 2, 第 1 位的位权为 1。

1.1.2 数制之间的转换

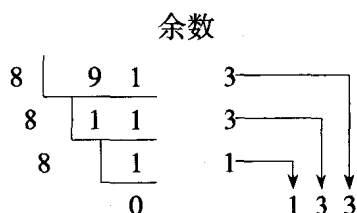
1. 十进制数转换成非十进制数 十进制数转换成二进制数、八进制数或十六进制数, 整数转换中采用除基数“2”、“8”、“16”取余的方法; 小数转换中采用乘基数“2”、“8”、“16”取整的方法。

例 1-1 试求出十进制数 91 的二进制数、八进制数和十六进制数。

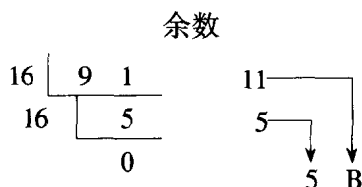
解 十进制数转换成二进制数、八进制数和十六进制数整数时, 采用“除 2 取余”、“除 8 取余”、“除 16 取余”直到商为 0, 第一个余数是转换后数的最低位, 最后一个余数是最高位。计算过程如下:



即十进制数转换成二进制数： $91 = 1011011\text{B}$



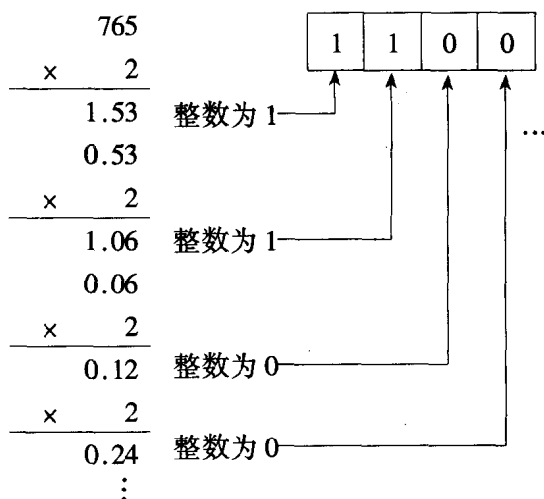
即十进制数转换成八进制数： $91 = (133)_8$



即十进制数转换成十六进制数： $91 = 5\text{BH}$

例 1-2 把十进制小数 0.765 转换成二进制小数。

解 把十进制小数转换成二进制小数时，采用“乘 2 取整”，第一个乘积的整数为转换结果的最高位，最后一个乘积的整数是最低位。运算过程为：



即 $0.765 \approx 0.11\text{B}$

对同时有整数和小数两部分的十进制数，其转换成二进制数的方法是：将它的整数和小数部分分别转换后，再合并起来。例如：将例 1-1 和例 1-2 合并起来，得到：

$$91.765 \approx 1011011.11\text{B}$$

例 1-2 中，还可采用“乘 8 取整”、“乘 16 取整”将 0.765 转换成八进制小数和十六进制小数，第一个乘积的整数为转换结果的最高位，最后一个乘积的整数是最低位。

2. 非十进制数转换成十进制数 转换方法是把非十进制数按权展开求和即可。例如：

二进制数转换成十进制数：

$$11010.01\text{B} = 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} = 26.25$$

八进制数转换成十进制数：

$$(30.7)_8 = 3 \times 8^1 + 0 \times 8^0 + 7 \times 8^{-1} = 24.875$$

十六进制数转换成十进制数：

$$3AH = 3 \times 16^1 + 10 \times 16^0 = 58$$

3. 二进制数、八进制数、十六进制数之间的转换

(1) 二进制数与八进制数之间的转换，由于 $2^3 = 8$ ，所以 3 位二进制数相当于 1 位八进制数，只要把 3 位二进制数转换为 1 位八进制数；反之，把 1 位八进制数转换为 3 位二进制数，即可实现二进制数与八进制数之间的转换。例如：

八进制数转换为二进制数：

$$\begin{array}{cccc} 6 & 7. & 2 & 1 \\ \downarrow & \downarrow & \downarrow & \downarrow \\ 110 & 111. & 010 & 001 \end{array}$$

$$\text{即：}(67.21)_8 = (110\ 111.010\ 001)_2$$

二进制数转换为八进制数：

$$\begin{array}{ccc} 011 & 101 & .\ 11\ \boxed{0} \\ \downarrow & \downarrow & \downarrow \\ 3 & 5 & .\ 6 \end{array}$$

$$\text{即：}011\ 101.11B = (35.6)_8$$

(2) 二进制数与十六进制数之间的转换，由于 $2^4 = 16$ ，所以 4 位二进制数相当于 1 位十六进制数，只要把 4 位二进制数转换为 1 位十六进制数；反之，把 1 位十六进制数转换为 4 位二进制数，即可实现二进制数与十六进制数之间的转换。例如：

二进制数转换为十六进制数：

$$\begin{array}{cccc} \boxed{000}\ 1 & 1011 & 1110 & .\ 01\ \boxed{00} \\ \downarrow & \downarrow & \downarrow & \downarrow \\ 1 & B & E & .\ 4 \end{array}$$

$$\text{即：}1\ 1011\ 1110.01B = 1BE.4H$$

十六进制数转换为二进制数：

$$\begin{array}{cccc} 7 & A & 5 & .\ 3 \\ \downarrow & \downarrow & \downarrow & \downarrow \\ 0111 & 1010 & 0101 & .\ 0011 \end{array}$$

$$\text{即：}7A5.3H = 111\ 1010\ 0101.0011B$$

1.1.3 计算机中数的表示法

我们知道计算机中采用二进制数，那么，一个二进制数在机器中具体是怎样表示的呢？数有正负之分，整数小数之分，也就是带符号数或带小数点数。为表示这样的数，通常采用两种表示方法：一是用数值的数码表示，即原码、反码和补码表示；二是用数值中的小数点表示，即定点数和浮点数表示。

1. 带符号数的表示 计算机区分二进制数的正负是把二进制数的最高位作为符号位加以识别。一个 8 位二进制数，符号位 $D_7 = 0$ ，表示正数； $D_7 = 1$ ，表示负数，如图 1-1 所示。习惯上，把存放在计算机中、连同符号位在内的二进制数称为机器数，而不包括符号位在

的 7 位二进制数称为机器数的真值。

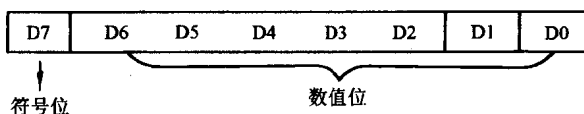


图 1-1 8 位微机中的带符号数

例如：在字长为 8 位的微型计算机中，+76 的机器数为 0 1001100B，最高位是符号位 $D_7 = 0$ ，表示正数。真值为 +1001100B；-76 的机器数为 1 1001100B，最高位是符号位 $D_7 = 1$ ，表示负数。真值为 -1001100B。

机器数通常有原码、反码和补码三种表示形式。

(1) 原码(true form)，凡是正数的符号位用 0 表示，负数的符号位用 1 表示，而数值位保持原样的机器数称为原码。

例如：X = +12 $[X]_{\text{原}} = 00001100\text{B}$

X = -12 $[X]_{\text{原}} = 10001100\text{B}$

当 0 这个数有 +0 和 -0 时，它的原码不相同。

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

原码表示方法简单易懂，与真值转换方便，8 位二进制原码表示数的范围是 -127 ~ +127，见表 1-2。但是，两个异号数相加或两个同号数相减，就要作减法运算，由于机器中一般只有加法器，没有减法器，所以，为将减法运算变为加法运算就引入了反码和补码。

(2) 反码(one's complement)，正数的反码和原码相同；负数的反码是符号位不变，数值位按位求反(即绝对值按位求反)。

例如：X = +12 $[X]_{\text{反}} = [X]_{\text{原}} = 00001100\text{B}$

X = -12 $[X]_{\text{原}} = 10001100\text{B}$

$[X]_{\text{反}} = 11110011\text{B}$

$[+0]$ 和 $[-0]$ 的反码形式为

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

8 位二进制反码表示数的范围是 -127 ~ +127，见表 1-2。

(3) 补码(two's complement)，正数的补码与正数的原码相同；负数的补码是由它的反码 +1 得到(即绝对值按位求反加 1)。

例如：X = +12 $[X]_{\text{补}} = [X]_{\text{原}} = 00001100\text{B}$

X = -12 $[X]_{\text{补}} = [X]_{\text{反}} + 1 = (11110011 + 1)\text{B} = 11110100\text{B}$

8 位二进制补码表示数的范围是 -128 ~ +127， $[+0]_{\text{补}} = [-0]_{\text{补}} = 00000000\text{B}$ ，如表 1-2 所示。

表 1-2 机器数的原码、反码和补码

二进制数	原码	反码	补码
0000 0000	+0	+0	± 0
0000 0001	+1	+1	+1
0000 0010	+2	+2	+2
⋮	⋮	⋮	⋮

(续)

二进制数	原码	反码	补码
0111 1101	+125	+125	+125
0111 1110	+126	+126	+126
0111 1111	+127	+127	+127
1000 0000	-0	-127	-128
1000 0001	-1	-126	-127
1000 0010	-2	-125	-126
⋮	⋮	⋮	⋮
1111 1101	-125	-2	-3
1111 1110	-126	-1	-2
1111 1111	-127	-0	-1

采用补码作加、减法运算时,符号位和数值位一起参加运算,符号位的进位略去不计。作加法运算时,两数之和的补码等于两数补码之和,即 $[X+Y]_{\text{补}}=[X]_{\text{补}}+[Y]_{\text{补}}$;作减法运算时,两数之差的补码等于两数补码之和,即 $[X-Y]_{\text{补}}=[X]_{\text{补}}+[-Y]_{\text{补}}$ 。

例 1-3 已知 $X=+29$, $Y=+5$ 试求 $X+Y$ 的二进制数值。

解 $[29]_{\text{原}}=0001\ 1101\text{B}$ $[5]_{\text{原}}=0000\ 0101\text{B}$

$[29]_{\text{补}}=[29]_{\text{原}}=0001\ 1101\text{B}$ $[5]_{\text{补}}=[5]_{\text{原}}=0000\ 0101\text{B}$

于是有: $[29]_{\text{补}}+[5]_{\text{补}}$

$$\begin{array}{r} 0001\ 1101 \\ +) 0000\ 0101 \\ \hline 0010\ 0010 \end{array}$$

而 $[29+5]_{\text{补}}=[34]_{\text{原}}=[34]_{\text{补}}=0010\ 0010\text{B}$

所以 $[29+5]_{\text{补}}=[29]_{\text{补}}+[5]_{\text{补}}$

$29+5=10\ 0010\text{B}$

例 1-4 上例中,求 $X-Y$ 。

解 $[29]_{\text{补}}=[29]_{\text{原}}=0001\ 1101\text{B}$ $[-5]_{\text{原}}=1000\ 0101\text{B}$ $[-5]_{\text{补}}=1111\ 1011\text{B}$

于是有: $[29]_{\text{补}}+[-5]_{\text{补}}$

$$\begin{array}{r} 0001\ 1101 \\ +) 1111\ 1011 \\ \hline \boxed{1}\ 0001\ 1000 \end{array}$$

↓

自然丢失

$[29]_{\text{补}}+[-5]_{\text{补}}=0001\ 1000\text{B}$

而 $[29-5]_{\text{补}}=[24]_{\text{原}}=[24]_{\text{补}}=0001\ 1000\text{B}$

所以 $[29-5]_{\text{补}}=[29]_{\text{补}}+[-5]_{\text{补}}$ 两种运算方法的结果相同。

说明两点:

1) 一般计算机中运算采用补码,这样减法运算变成加法运算,运算器中只设置加法器,简化了计算机的硬件结构。

2) 在字长 8 位的计算机中,补码加法运算结果只保留 8 位,符号位的进位符号自然丢失,只要数值在 $-128 \sim +127$ 范围内,结论就是正确的,但超出这个范围,就会得到错误的

结论。

例如: $X = -90$, $Y = -53$

$$[-90]_{\text{原}} = 1101\ 1010\text{B}; [-90]_{\text{补}} = [-90]_{\text{反}} + 1 = 1010\ 0101 + 1 = 1010\ 0110\text{B},$$

$$[-53]_{\text{原}} = 1011\ 0101\text{B}; [-53]_{\text{补}} = [-53]_{\text{反}} + 1 = 1100\ 1010 + 1 = 1100\ 1011\text{B}$$

于是 $[-90]_{\text{补}} + [-53]_{\text{补}} = \boxed{1}\ 0111\ 0001\text{B}$ 进位符号 1 自然丢失

$$\text{则有 } [(-90) + (-53)]_{\text{补}} = [-90]_{\text{补}} + [-53]_{\text{补}} = 0111\ 0001\text{B}$$

$$(-90) + (-53) = +0111\ 0001 (\text{真值}) = +113$$

因为 $(-90) + (-53) = -143 < -128$, 发生溢出, 符号位受破坏, 所以结果不正确。

2. 定点数和浮点数 在微型计算机中, 二进制的小数点位置通常采用定点和浮点两种方法表示。采用定点数的称为定点机, 采用浮点数的称为浮点机, 单片机一般采用定点表示法, 是定点机。

(1) 定点数 是指小数点位置固定不变的数。小数点可以固定在数值位之前, 也可以固定在数值位之后。前者称为定点小数表示法, 后者称为定点整数表示法。如图 1-2 所示。

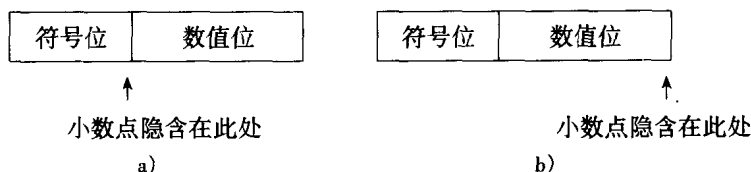


图 1-2 定点小数点位置

a) 定点小数表示 b) 定点整数表示

定点数的这两种表示方法在计算机中均采用, 究竟用那一种是事先约定好的。

(2) 浮点数 在采用浮点表示的二进制数中, 小数点位置是浮动的, 不固定的。对于任意一个十进制数, 可以用 $N = 10^P \times S$ 表示, 如:

$$1594.32 = 10^4 \times 0.159432;$$

同样, 对于任意一个二进制数, 也可以用 $N = 2^P \times S$ 表示, 如:

$$110.011\text{B} = 1000\text{B} \times 0.110011\text{B} = 2^{+11\text{B}} \times 0.110011\text{B}$$

式中, S 称为 N 的尾数; P 称为 N 的阶码; 2 称为阶码的底。 P 、 S 均用二进制数表示。并且尾数 S 是有符号的数时, 可用补码表示, 阶码 P 也可以为正数或负数, 也用补码表示。比较这两种表示方法, 浮点法的优点是可以表示较大范围的数, 如 16 位二进制数表示数的范围为 $-2^{31} \sim +2^{31}$, 缺点是运算规则复杂, 要对阶码和尾数分别运算; 定点法的优点是运算规则简单, 但表示数的范围有限, 如 16 位二进制数表示数的范围为 $0 \sim 2^{16}$ (无符号数), $-2^{15} \sim +2^{15}$ (有符号数)。因此, 计算机通常采用定点法。在定点机中, 为扩大数的范围, 利用增加数的范围的方法, 如 16 位、32 位、64 位等。

1.1.4 二进制数的运算

1. 二进制数的算术运算

(1) 加法运算, 二进制数加法运算规则是:

$$1) 0 + 0 = 0$$

$$2) 0 + 1 = 1 + 0 = 1$$

$$3) 1 + 1 = 10 (\text{向高位进位})$$

例如: $1010\ 0110\text{B} + 1101\ 1011\text{B}$ 的算式如下:

$$\begin{array}{r}
 \text{被加数} \quad 1010 \ 0110 \\
 +) \text{加数} \quad 1101 \ 1011 \\
 \hline
 \text{和} \quad \boxed{1} \ 1000 \ 0001
 \end{array}$$

两个二进制数相加时要注意低位的进位。

(2) 减法运算，二进制数减法运算规则是：

- 1) $0 - 0 = 1 - 1 = 0$
- 2) $1 - 0 = 1$
- 3) $0 - 1 = 1$ (向高位借位)

例如：1101 1011B + 1010 0110B 的算式如下：

$$\begin{array}{r}
 \text{被减数} \quad 1101 \ 1011 \\
 -) \text{减数} \quad 1010 \ 0110 \\
 \hline
 \text{差} \quad 0011 \ 0101
 \end{array}$$

两个二进制数相减时，低位向高位的借 1 应看作 2。

(3) 乘法运算，二进制数乘法运算规则是：

- 1) $0 \times 0 = 0$
- 2) $0 \times 1 = 1 \times 0 = 0$
- 3) $1 \times 1 = 1$

例如：1101B × 1011B 的算式如下：

$$\begin{array}{r}
 \text{被乘数} \quad 1101 \\
 \times) \text{乘数} \quad 1011 \\
 \hline
 \text{部分积} \quad 1101 \\
 \quad \quad 1101 \\
 \quad \quad 0000 \\
 \quad \quad 1101 \\
 \hline
 \text{乘积} \quad 10001111
 \end{array}$$

二进制数乘法特点是：乘数某位为 1，则将被乘数拷贝下来，乘数某位为 0，则不拷贝；换一位则左移一位。因此，二进制数乘法是采用边移位、边拷贝的方法实现的，即计算机完成“拷贝 + 移位”操作。在计算机内部，只需有加法、移位和判断电路，乘法运算通过程序就可以实现。

(4) 除法运算，二进制数除法运算规则是：

- 1) $0 \div 0 = 0$
- 2) $0 \div 1 = 0$ ($1 \div 0$ 无意义)
- 3) $1 \div 1 = 1$

例如：10101011B ÷ 110B 的算式如下：

除数	110	111 10101011 110 1001 110 110 110 11	商 被除数 余数
----	-----	--	--