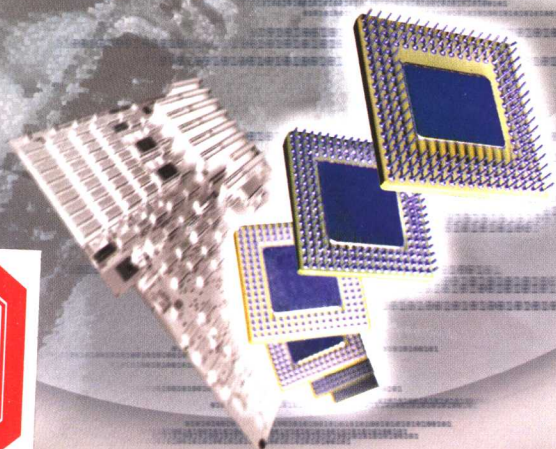


高等学校计算机科学与技术教材

单片机原理与应用

朱定华 戴汝平 编著



清华大学出版社

<http://www.tup.tsinghua.edu.cn>



北方交通大学出版社

<http://press.njtu.edu.cn>

高等学校计算机科学与技术教材

单片微机原理与应用

朱定华 戴汝平 编著

清华大学出版社
北方交通大学出版社
·北京·

内 容 简 介

《单片微机原理与应用》以 MCS-51 单片机为背景机,系统地介绍微型计算机的原理及应用,主要内容包
括计算机基础知识,汇编语言程序设计,MCS-51 单片机的内部接口,MCS-51 单片机的扩展方法,半导体存
储器,常用可编程接口芯片,A/D 和 D/A 转换芯片等;还较详细地介绍以 80C51 为基础结构并与其兼容的
8XC552 单片机和 C8051FXXX 单片机。

本书内容精练,实例丰富。其中大量的接口电路和程序是作者多年在科研和教学中反复提炼得出的,因
而本书应用性很强。本书内容系统全面,论述深入浅出,循序渐进,可作为大专院校“汇编语言程序设计”,“微
机原理”或“微机原理及接口技术”等课程的教学用书,也可以供从事电子技术、计算机应用与开发的科研人员
和工程技术人员学习参考,还适用于初学者自学使用。

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

单片微机原理与应用 / 朱定华, 戴汝平编著. —北京: 北方交通大学出版社, 2003.8
(高等学校计算机科学与技术教材)

ISBN 7-81082-137-7

I. 单… II. 朱… III. ① 单片微型计算机-基础-高等学校-教材 ② 单片微型计算机-接口-高等学校-教
材 IV. TP368.1

中国版本图书馆 CIP 数据核字(2003)第 039564 号

责任编辑: 韩 乐

印 刷 者: 北方交通大学印刷厂

出版发行: 北方交通大学出版社 邮编: 100044 电话: 010-51686045, 62237564

清 华 大 学 出 版 社 邮编: 100084

经 销: 各地新华书店

开 本: 787×1092 1/16 印张: 19.25 字数: 480 千字

版 次: 2003 年 8 月第 1 版 2003 年 8 月第 1 次印刷

印 数: 6000 册 定价: 26.00 元

前 言

当今微型计算机技术发展形成两大分支,一是以微处理器(Micro Processor Unit)为核心所构成的通用微机系统,另一大分支是微控制器(Micro Controller Unit),俗称单片机。面对这样“势均力敌”的两大分支,大专院校的“微机原理”课程究竟应选用何种机型为背景机来组织教学,也出现分歧。

作为 21 世纪的工科大学生,不仅要熟练地使用通用微机进行各种数据处理,还要把计算机技术运用到本专业领域或相关领域,即具有“开发”能力。由于新世纪的工科大专院校的大学生既要掌握通用微机,又要掌握单片机,所以有些学校在学习以通用微机为背景机的“微机原理”课程后,又开设“单片机及接口技术”选修课。单片机和通用微机作为微型机发展的两大分支,其基本结构、工作原理、控制思路及实现方法都非常类似,有了一个做基础,再学另一个就很容易了。选用 80x86 通用微机和 MCS-51 系列单片机为背景机来组织“微机原理”课程的教学都是可行的。

MCS-51 单片机应用于控制目的时,功能已足够强大,已能满足控制领域中多数场合的要求。MCS-51 单片机正朝着高速、高性能和多功能方向发展。Philips 公司开发的 8XC552 单片机和 Cygnal 公司开发的 C8051FXXX 单片机都以 80C51 为基础结构,并与 80C51 兼容。因此,选用 MCS-51 系列单片机作为“微机原理”课程的主要内容,既可满足教学内容稳定,实验设备成熟便宜,又不失其先进性与实用性。本书是以 Intel 公司的 MCS-51 系列单片机为背景机的“微机原理”课程的教材。本书在附录中较详细地介绍 8XC552 单片机和 C8051FXXX 单片机。

本书全面地介绍 MCS-51 单片机的结构原理和应用技术,共分 8 章,还有 5 个附录。第 1 章介绍微型计算机的基础知识,包括计算机中的数制和编码,逻辑单元和逻辑部件,微型计算机的结构和工作原理及 MCS-51 单片机的结构和主要系列产品的特性。第 2 章介绍 MCS-51 单片机常用的汇编指令和伪指令及指令的时序。第 3 章介绍汇编语言程序设计的基本技术。通过第 2 章和第 3 章的学习,使读者更透彻地了解汇编语言程序设计,为编程应用打下基础。第 4 章介绍 MCS-51 单片机的内部接口电路,包括中断系统、定时器、并行口和串行口,同时还介绍计算机间的通信。第 5 章介绍了 MCS-51 单片机的最小应用系统和扩展技术。第 6 章介绍存储器及其与微型计算机的接口技术。第 7 章介绍常用可编程接口芯片的功能与应用。第 8 章介绍 A/D 和 D/A 转换器与微型计算机的接口与应用。

本书内容精练,实例丰富。每章后均附有思考题与习题。编写本书时,注意了理论和实践相结合,力求做到既有一定的理论基础,又能运用理论解决实际问题;既掌握一定的先进技术,又着眼于当前的应用服务。

本课程的参考学时数为 64 学时(不含实验)。与本书配套的教材有《单片机原理及接口技术学习辅导》和《单片机原理及接口技术实验》。

本书由朱定华和戴汝平编写,参加本书编写工作的人员还有朱悦、戴颖颖、王根、周晓龙、王琳、赵捷、张禁静、林卫、王虹等。

由于作者水平有限，书中难免还存在一些不足和错误，殷切希望广大读者批评指正。

编著者
2003.8 于华中科技大学

目 录

第 1 章 微型计算机的基础知识	(1)
1.1 计算机中的数和编码	(1)
1.1.1 计算机中的数制	(1)
1.1.2 符号数的表示法	(2)
1.1.3 二进制数的加减运算	(4)
1.1.4 二进制数的逻辑运算与逻辑电路	(6)
1.1.5 二进制编码	(9)
1.1.6 BCD 数的加减运算	(10)
1.2 逻辑单元与逻辑部件	(11)
1.2.1 触发器	(11)
1.2.2 寄存器	(13)
1.2.3 移位寄存器	(14)
1.2.4 计数器	(14)
1.2.5 三态输出门与缓冲放大器	(15)
1.2.6 译码器	(16)
1.3 微型计算机的结构和工作原理	(16)
1.3.1 微型计算机常用的术语	(16)
1.3.2 微型计算机的基本结构	(17)
1.3.3 计算机的工作原理	(19)
1.4 MCS-51 单片机的基本组成和存储器配置	(20)
1.4.1 8051 单片机的基本组成	(20)
1.4.2 MCS-51 单片机的存储器	(21)
1.4.3 特殊功能寄存器	(23)
1.5 MCS-51 系列单片机	(27)
1.5.1 51 子系列和 52 子系列	(27)
1.5.2 单片机芯片的半导体工艺	(27)
1.5.3 AT89 系列单片机	(28)
习题与思考题	(28)
第 2 章 汇编语言与汇编程序	(30)
2.1 符号指令的寻址方式	(30)
2.1.1 寄存器寻址	(30)
2.1.2 立即寻址	(31)
2.1.3 直接寻址	(31)
2.1.4 间接寻址	(31)
2.1.5 变址寻址	(32)

2.1.6 位寻址·····	(33)
2.1.7 符号指令的操作数中使用的符号·····	(33)
2.2 常用指令·····	(33)
2.2.1 数据传送类指令·····	(33)
2.2.2 加减运算指令·····	(36)
2.2.3 逻辑运算及移位类指令·····	(38)
2.2.4 位操作指令·····	(41)
2.2.5 指令应用举例·····	(43)
2.3 伪指令·····	(45)
2.3.1 常量和标号·····	(45)
2.3.2 程序的定位和结束伪指令·····	(46)
2.4 指令的时序·····	(47)
2.4.1 指令周期、机器周期和状态 ·····	(47)
2.4.2 MCS-51 指令的时序 ·····	(47)
2.4.3 MCS-51 指令的执行过程 ·····	(49)
习题与思考题 ·····	(50)
第3章 程序设计的基本技术 ·····	(53)
3.1 顺序程序设计·····	(53)
3.1.1 乘法指令·····	(53)
3.1.2 BCD 数加法调整指令 DA A ·····	(54)
3.1.3 顺序程序设计举例·····	(56)
3.2 分支程序设计·····	(58)
3.2.1 条件转移指令·····	(58)
3.2.2 比较不等转移指令·····	(59)
3.2.3 无条件转移指令·····	(60)
3.2.4 应用举例·····	(61)
3.3 循环程序设计·····	(63)
3.3.1 减 1 非零转移指令 DJNZ ·····	(64)
3.3.2 单重循环程序设计举例·····	(65)
3.3.3 多重循环程序·····	(70)
3.4 子程序设计·····	(71)
3.4.1 子程序的概念·····	(72)
3.4.2 子程序的调用指令与返回指令·····	(74)
3.4.3 子程序及其调用程序设计举例·····	(74)
习题与思考题 ·····	(79)
第4章 MCS-51 单片机内部接口电路 ·····	(81)
4.1 接口的基本概念·····	(81)
4.1.1 接口电路的功能·····	(81)
4.1.2 接口控制原理·····	(82)

4.1.3 串行接口 UART、SPI 和 I ² C/SMBus	(83)
4.1.4 并行接口	(84)
4.2 中断及 MCS-51 单片机的中断系统	(85)
4.2.1 中断和中断处理过程	(85)
4.2.2 MCS-51 单片机的中断系统	(86)
4.2.3 多个外部中断源的系统设计	(90)
4.3 定时器	(91)
4.3.1 定时器的结构	(92)
4.3.2 定时器的工作方式	(93)
4.3.3 定时器应用举例	(96)
4.4 并行输入输出接口	(99)
4.4.1 P1 口	(99)
4.4.2 P3 口	(100)
4.4.3 P2 口	(101)
4.4.4 P0 口	(101)
4.4.5 并行输入输出接口应用举例	(102)
4.5 UART 串行输入输出接口	(109)
4.5.1 串行口的构成	(109)
4.5.2 串行口控制寄存器 SCON	(110)
4.5.3 电源控制寄存器 PCON	(111)
4.5.4 工作方式和波特率的设定	(111)
4.5.5 串行口应用举例	(113)
4.6 串行通信	(115)
4.6.1 双机通信	(115)
4.6.2 多机通信	(117)
4.6.3 MCS-51 单片机与 80x86 微型计算机的通信	(123)
习题与思考题	(128)
第 5 章 单片机的最小应用系统与外部扩展	(131)
5.1 单片机的最小应用系统	(131)
5.1.1 单片机的时钟电路	(131)
5.1.2 复位电路及复位状态	(132)
5.1.3 MCS-51 单片机引线及片外总线结构	(133)
5.1.4 89C51 单片机的最小应用系统	(135)
5.1.5 8031 单片机的最小应用系统	(135)
5.2 单片机的外部扩展	(136)
5.2.1 外部扩展芯片与地址总线的连接	(136)
5.2.2 外部扩展芯片与数据总线的连接	(138)
5.2.3 外部扩展芯片与控制总线的连接	(138)
5.3 用 TTL 或 CMOS 芯片扩展简单的 I/O 接口	(139)

5.3.1	用寄存器扩展简单的输出接口	(139)
5.3.2	用三态缓冲器扩展输入接口	(140)
5.3.3	三态缓冲寄存器扩展输入输出接口	(141)
5.3.4	应用举例	(142)
5.3.5	开关量的驱动与隔离	(147)
	习题与思考题	(149)
第 6 章	半导体存储器	(151)
6.1	存储器概述	(151)
6.1.1	存储器的类型	(151)
6.1.2	存储器的性能指标与分级结构	(151)
6.2	常用的存储器芯片	(152)
6.2.1	半导体存储器芯片的结构	(152)
6.2.2	随机读写存储器 RAM	(153)
6.2.3	只读存储器 ROM	(154)
6.3	存储器的扩展	(156)
6.3.1	数据存储器的扩展	(156)
6.3.2	程序存储器的扩展	(156)
6.3.3	程序存储器和数据存储器的扩展	(158)
6.4	串行 EEPROM 存储器及其应用	(158)
6.4.1	24LCXX	(159)
6.4.2	X25043/45	(172)
6.4.3	X84041	(180)
	习题与思考题	(184)
第 7 章	常用可编程接口芯片	(185)
7.1	可编程并行接口 8255	(185)
7.1.1	8255 的组成与接口信号	(185)
7.1.2	8255 的工作方式与控制字	(186)
7.1.3	三种工作方式的功能	(189)
7.2	可编程计数器/定时器 8253	(193)
7.2.1	8253 的组成与接口信号	(193)
7.2.2	计数器的工作方式及其与输入输出的关系	(195)
7.2.3	8253 的控制字和初始化编程	(196)
7.2.4	8253 的应用	(198)
7.3	可编程多功能接口 8155	(199)
7.3.1	8155 的组成与接口信号	(199)
7.3.2	8155 的命令状态字	(202)
7.3.3	8155 与 MCS-51 单片机的连接	(203)
7.4	键盘/显示控制器 8279	(204)
7.4.1	8279 的组成与接口信号	(204)

7.4.2	8279 的操作命令	(206)
7.4.3	8279 在键盘和显示器接口中的应用	(206)
	习题与思考题	(211)
第 8 章	模拟通道接口	(212)
8.1	数模转换器及其与微型计算机的接口	(212)
8.1.1	8 位数模转换芯片 DAC0832	(213)
8.1.2	12 位数模转换芯片 DAC1210	(220)
8.1.3	10 位 D/A 转换器 AD7520	(221)
8.2	模数转换器 ADC 及其与微型计算机的接口	(222)
8.2.1	8 位逐次逼近式 A/D 转换芯片 ADC0808	(222)
8.2.2	12 位逐次比较式数模转换芯片 AD574	(224)
8.3	串行模数转换器和数模转换器	(227)
8.3.1	串行模数转换器 TLC2543	(227)
8.3.2	串行输入 12 位电压输出数模转换器 TLV5616	(231)
	习题与思考题	(234)
附录 A	52 子系列单片机	(235)
A.1	内部 RAM	(235)
A.2	定时器 T2	(235)
A.2.1	定时器 T2 的中断	(235)
A.2.2	定时器 T2 的寄存器	(236)
A.2.3	定时器 T2 的工作方式	(237)
附录 B	8XC552 单片机	(241)
B.1	并行 I/O 端口	(241)
B.1.1	P1 口	(241)
B.1.2	P4 口	(241)
B.1.3	P5 口	(242)
B.2	串行 I/O 端口	(242)
B.2.1	8XC552 的串行口 1	(242)
B.2.2	8XC552 串行口 1 的特殊功能寄存器	(242)
B.2.3	8XC552 串行口 1 的工作方式	(243)
B.3	定时器	(244)
B.3.1	定时器 T2	(244)
B.3.2	定时器 T3	(247)
B.4	脉冲宽度调制器 PWM 和 D/A 转换器	(249)
B.5	A/D 转换器	(249)
B.6	中断系统	(250)
B.6.1	8XC552 新增加的中断源和中断标志位	(250)
B.6.2	8XC552 对中断请求的控制	(251)
B.6.3	中断服务程序的入口地址	(253)

B.7	8XC552 单片机引出线及其功能	(253)
B.7.1	8XC552 单片机引出线的排列	(253)
B.7.2	8XC552 单片机引出线的功能	(253)
附录 C	C8051FXXX 单片机	(255)
C.1	C8051FXXX 单片机的指令、时钟和复位	(255)
C.1.1	C8051FXXX 单片机的指令集	(255)
C.1.2	C8051FXXX 单片机的时钟	(255)
C.1.3	C8051FXXX 单片机的复位	(257)
C.2	C8051FXXX 单片机的中断系统	(258)
C.2.1	中断源和中断服务程序入口地址	(258)
C.2.2	特殊功能寄存器中的中断标志位	(259)
C.2.3	中断允许寄存器和中断优先级控制寄存器	(260)
C.3	片内存储器	(261)
C.3.1	程序存储器	(261)
C.3.2	内部数据存储器	(262)
C.3.3	片内的外部数据存储器(内部 XRAM)	(262)
C.3.4	外部数据存储器的访问	(264)
C.4	定时器	(264)
C.4.1	定时器 3	(264)
C.4.2	定时器 4	(265)
C.5	可编程计数器阵列 PCA	(265)
C.5.1	PCA 计数器	(265)
C.5.2	可编程计数器阵列 PCA	(265)
C.5.3	PCA 的特殊功能寄存器	(265)
C.6	串行口	(267)
C.6.1	UART 串行口	(267)
C.6.2	SMBus	(268)
C.6.3	SPI	(269)
C.7	模数转换器	(270)
C.7.1	模拟多路选择器	(270)
C.7.2	模数转换器 ADC 的工作方式	(271)
C.7.3	可编程增益放大器和模数转换器的转换速度	(272)
C.7.4	C8051F02X 的 8 位模数转换器 ADC1	(273)
C.8	数模转换器	(275)
C.9	比较器	(276)
C.9.1	比较器的工作原理和特殊功能寄存器	(276)
C.9.2	比较器 0 复位	(278)
C.10	可编程数字 I/O 和交叉开关	(278)
C.10.1	C8051F00X/01X 单片机的数字 I/O 和交叉开关	(278)

C.10.2	C8051F02X 单片机的 I/O 端口和交叉开关	(281)
C.11	C8051FXXX 单片机的引线图	(285)
C.11.1	C8051F00X/C8051F01X 系列单片机的引线图	(285)
C.11.2	C8051F02X 系列单片机的引线图	(286)
附录 D	MCS-51 指令系统表	(288)
附录 E	MCS-51 指令速查表	(292)

第 1 章 微型计算机的基础知识

1.1 计算机中的数和编码

1.1.1 计算机中的数制

计算机最早是作为一种计算工具出现的,所以它的最基本的功能是对数进行加工和处理。数在机器中是以器件的物理状态来表示的。一个具有两种不同的稳定状态且能相互转换的器件就可以用来表示 1 位(bit)二进制数。二进制数有运算简单,便于物理实现,节省设备等优点,所以目前在计算机中的数几乎全部采用二进制表示。但是二进制数书写起来太长,且不利于阅读和记忆;又加上目前大部分微型计算机是 8 位、16 位或 32 位的,都是 4 的整数倍,而 4 位二进制数即是 1 位十六进制数;所以微型计算机中的二进制数都采用十六进制数来表示。十六进制数用 0~9,A~F 等 16 个数码表示十进制数 0~15。因此,1 个 8 位的二进制数可用 2 位十六进制数表示,1 个 16 位的二进制数可用 4 位十六进制数表示等。这样书写方便,且便于阅读和记忆。然而人们最熟悉、最常用的是十进制数。为此,要熟练地掌握十进制数、二进制数和十六进制数间的相互转换。它们之间的关系如表 1-1 所列。

表 1-1 十进制数、二进制数及十六进制数对照表

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

为了区别十进制数、二进制数及十六进制数 3 种数制,可在数的右下角注明数制,或者在数的后面加一字母。如 B(binary)表示二进制数制;D(decimal)表示十进制数制;H(hexadecimal)表示十六进制数制。其中,十进制数后的字母 D 可以省略。

1. 二进制数和十六进制数间的相互转换

根据表 1-1 所示的对应关系即可实现它们之间的转换。

二进制整数转换为十六进制数,其方法是从右(最低位)向左将二进制数分组:每 4 位为 1 组,最后一组若不足 4 位则在其左边添加 0,以凑成 4 位 1 组,每组用 1 位十六进制数表示。如:

$$11111110001111\text{B} \rightarrow 1\ 1111\ 1100\ 0111\text{B} \rightarrow 0001\ 1111\ 1100\ 0111\text{B} = 1\text{FC7H}$$

十六进制数转换为二进制数,只需用 4 位二进制数代替 1 位十六进制数即可。如:

$$3\text{AB9H} = 0011\ 1010\ 1011\ 1001\text{B}$$

2. 十六进制数和十进制数间的相互转换

十六进制数转换为十进制数十分简单,只需将十六进制数按权展开相加即可。如:

$$\begin{aligned}
 1F3DH &= 16^3 \times 1 + 16^2 \times 15 + 16^1 \times 3 + 16^0 \times 13 = \\
 &4096 \times 1 + 256 \times 15 + 16 \times 3 + 1 \times 13 = \\
 &4096 + 3840 + 48 + 13 = 7997
 \end{aligned}$$

十进制整数转换为十六进制数可用除 16 取余法,即用 16 不断地去除待转换的十进制数,直至商等于 0 为止。将所得的各次余数,依倒序排列,即可得到所转换的十六进制数。如将 38947 转换为十六进制数,其方法及算式如下:

$$\begin{array}{r}
 16 \overline{) 38947} \quad 3 \\
 \underline{16 \quad 2434} \quad 2 \\
 \underline{16 \quad 152} \quad 8 \\
 \underline{16 \quad 9} \quad 9 \\
 0
 \end{array}$$

即 $38947 = 9823H$

1.1.2 符号数的表示法

1. 机器数与真值

二进制数与十进制数一样有正负之分。在计算机中,常用数的符号和数值部分一起编码的方法表示符号数。常用的有原码、反码和补码表示法。这几种表示法都将数的符号数码化。通常正号用“0”表示,负号用“1”表示。为了区分一般书写时表示的数和机器中编码表示的数,称前者为真值,后者为机器数,即数值连同符号数码“0”或“1”一起作为一个数就称为机器数,而它的数值连同符号“+”或“-”称为机器数的真值。把机器数的符号位也当做数值的数,就是无符号数。

为了表示方便,常把 8 位二进制数称为字节,把 16 位二进制数称为字,把 32 位二进制数称为双字。对于机器数应将其用字节、字或双字表示,所以只有 8 位、16 位或 32 位机器数的最高位才是符号位。

2. 原码

按上所述,数值用其绝对值,正数的符号位用 0 表示,负数的符号位用 1 表示,这样表示的数就称为原码。如:

$$\begin{aligned}
 X1 &= 105 = +1101001B & [X1]_{\text{原}} &= 01101001B \\
 X2 &= -105 = -1101001B & [X2]_{\text{原}} &= 11101001B
 \end{aligned}$$

其中最高位为符号,后面 7 位是数值。用原码表示时, +105 和 -105 的数值部分相同而符号位相反。

8 位原码数的数值范围为 FFH~7FH (-127~127)。原码数 00H 和 80H 的数值部分相同、符号位相反,它们分别为 +0 和 -0。16 位原码数的数值范围为 FFFFH~7FFFH (-32767~32767)。原码数 0000H 和 8000H 的数值部分相同、符号位相反,它们分别为 +0 和 -0。

1 个二进制符号数的扩展是指一个数从位数较少扩展到位数较多,如从 8 位(字节)扩展到 16 位(字),或从 16 位扩展到 32 位(双字)。对于用原码表示的数,它的正数和负数仅 1 位符号位相反,数值位都相同。原码数的扩展是将其符号位向左移至最高位,符号位移过之位即最高位与数值位间的所有位都填入 0。例如:68 用 8 位表示为 44H,用 16 位表示为 0044H; -68 用 8 位表示为 C4H,用 16 位表示为 8044H。

原码表示简单易懂,而且与真值的转换方便。但若是两个异号数相加,或两个同号数相减,就要做减法。为了把减运算转换为加运算,从而简化计算机的结构,就引进了反码和补码。

3. 反码

正数的反码与原码相同;负数的反码为它的绝对值(即与其绝对值相等的正数)按位取反(连同符号位)。如:

$$\begin{aligned} X1 &= 105 = +1101001B & [X1]_{\text{反}} &= 01101001B \\ X2 &= -105 = -1101001B & [X2]_{\text{反}} &= 10010110B \end{aligned}$$

4. 补码

正数的补码与原码相同;负数的补码为与它的绝对值相等的正数的补数。把一个数连同符号位按位取反再加1,可以得到该数的补数。如:

$$\begin{aligned} X1 &= 105 = +1101001B & [X1]_{\text{补}} &= 01101001B \\ X2 &= -105 = -1101001B & [X2]_{\text{补}} &= 10010111B \end{aligned}$$

求补数还可以直接求,方法是从最低位向最高位扫描,保留直至第一个“1”的所有位,以后各位按位取反。负数的补码可以由其正数求补得到。根据两数互为补数的原理,对补码表示的负数求补就可以得到其正数,即可得该负数的绝对值。如:

$$[-105]_{\text{补}} = 10010111B = 97H$$

对其求补,从右向左扫描,第一位就是1,故只保留该位,对其左面的七位均求反得:01101001,即补码表示的机器数97H的真值是-69H(=-105)。

一个用补码表示的机器数,若最高位为0,则其余几位即为此数的绝对值;若最高位为1,其余几位不是此数的绝对值,把该数求补,才得到它的绝对值。

8位补码数的数值范围为80H~7FH(-128~127)。16位补码数的数值范围为8000H~7FFFH(-32768~32767)。字节80H和字8000H的真值分别是-128(-80H)和-32768(-8000H)。补码数80H和8000H的最高位既代表了符号为负又代表了数值为1。

1个二进制补码数的符号位向左扩展若干位后,所得到的补码数的真值不变。对于用补码表示的数,正数的扩展应该在其前面补0,而负数的扩展,则应该在前面补1。例如:68用8位(二进制数)表示为44H,用16位表示为0044H;-68用8位表示为BCH,用16位表示为FFBCH。

当数采用补码表示时,就可以把减法转换为加法。例如:

$$\begin{aligned} 64 - 10 &= 64 + (-10) \\ [64]_{\text{补}} &= 40H = 0100\ 0000B \\ [10]_{\text{补}} &= 0AH = 0000\ 1010B \\ [-10]_{\text{补}} &= 1111\ 0110B \end{aligned}$$

做减法运算过程如下:

$$\begin{array}{r} 0100\ 0000 \\ - 0000\ 1010 \\ \hline 0011\ 0110 \end{array}$$

用补码相加过程如下:

$$\begin{array}{r}
 0100\ 0000 \\
 - 1111\ 0110 \\
 \hline
 1\ 0011\ 0110 \\
 \uparrow \\
 \text{进位自然丢失}
 \end{array}$$

结果相同,其真值为:54(36H=48+6)。

最高位的进位是自然丢失的,故做减法与用补码相加的结果是相同的。因此,在微型机中,凡是符号数一律是用补码表示的。一定要记住运算的结果也是用补码表示的。如:

$$\begin{aligned}
 34 - 68 &= 34 + (-68) \\
 34 &= 22\text{H} = 0010\ 0010\text{B} \\
 68 &= 44\text{H} = 0100\ 0100\text{B} \\
 -68 &= 1011\ 1100\text{B}
 \end{aligned}$$

做减运算过程如下:

$$\begin{array}{r}
 0010\ 0010 \\
 - 0100\ 0100 \\
 \hline
 1\ 1101\ 1110 \\
 \uparrow \\
 \text{借位自然丢失}
 \end{array}$$

用补码相加过程如下:

$$\begin{array}{r}
 0010\ 0010 \\
 - 1011\ 1100 \\
 \hline
 1101\ 1110
 \end{array}$$

结果相同。因为符号位为1,所以结果为负数。对其求补,得其真值:-00100010B,即为-34(-22H)。

由上面两个例子还可以看出,当数采用补码表示后,两个正数相减,若无借位,化为补码相加就会有进位;若有借位,化为补码相加就不会有进位。

1.1.3 二进制数的加减运算

计算机把机器数均当做无符号数进行运算,即符号位也参与运算。运算的结果要根据运算结果的符号,运算有无进(借)位和溢出等来判别。计算机中设置有这些标志位,标志位的值由运算结果自动设定。

1. 无符号数的运算

无符号数实际上是指参加运算的数均为正数,且整个数位全部用于表示数值。 n 位无符号二进制数的范围为 $0 \sim (2^n - 1)$ 。

① 两个无符号数相加,由于两个加数均为正数,因此其和也是正数。当和超过其位数所允许的范围时,就向更高位进位。如:

$$127 + 160 = 7\text{FH} + \text{A0H}$$

$$\begin{array}{r}
 0111\ 1111 \\
 - 1010\ 0000 \\
 \hline
 1\ 0001\ 1111 = 11\text{FH} = 256 + 16 + 15 = 287
 \end{array}$$

↑
进位

② 两个无符号数相减,被减数大于或等于减数,无借位,结果为正;被减数小于减数,有借位,结果为负。如:

$$\begin{array}{r}
 192 - 10 = \text{C0H} - 0\text{AH} \\
 1100\ 0000 \\
 - 0000\ 1010 \\
 \hline
 1011\ 0110 = \text{B6H} = 176 + 6 = 182
 \end{array}$$

反过来相减,即 $10 - 192$,运算过程如下:

$$\begin{array}{r}
 0000\ 1010 \\
 - 1100\ 0000 \\
 \hline
 1\ 0100\ 1010 = -10110110\text{B} = -\text{B6H} = -182
 \end{array}$$

↑
借位

由此可见,对无符号数进行减法运算,其结果的符号用进位来判别:CF=0(无借位),结果为正;CF=1(有借位)结果为负(对8位数值位求补得到它的绝对值)。

2. 符号数的运算

n 位二进制数,除去一位符号位,还有 $n-1$ 位表示数值,所能表示的补码的范围为: $-2^{n-1} \sim (2^{n-1}-1)$ 。如果运算结果超过此范围就会产生溢出。如:

$$\begin{array}{r}
 105 + 50 = 69\text{H} + 32\text{H} \\
 0110\ 1001 \\
 + 0011\ 0010 \\
 \hline
 1001\ 1011 = 9\text{BH} = 155 \text{ 或 } -65\text{H} = -101
 \end{array}$$

若把结果视为无符号数,为155,结果是正确的。若将此结果视为符号数,其符号位为1,结果为-101,这显然是错误的。其原因是和数155大于8位符号数所能表示的补码数的最大值127,使数值部分占据了符号位的位置,产生了溢出,从而导致结果错误。又如:

$$\begin{array}{r}
 -105 - 50 = -155 \\
 1001\ 0111 \\
 + 1100\ 1110 \\
 \hline
 1\ 0110\ 0101
 \end{array}$$

↑
进位

两个负数相加,和应为负数,而结果01100101B却为正数,这显然是错误的。其原因是和数-155小于8位符号数所能表示的补码数的最小值-128,也产生了溢出。若不将第7位(第7位~第0位)0看做符号,也看做数值而将进位看做数的符号,结果为-10011011B=-155,结果就是正确的。

因此,应当注意溢出与进位及补码运算中的进位或借位丢失间的区别。