



国家示范性高职院校建设项目成果

高等职业教育教学改革系列规划教材·电子信息类

单片机原理 及接口技术

陈贵银 祝 福 主 编



任务驱动



行动导向



工学结合



学生主体



过程考核

本书提供多媒体课件、习题答案和源程序文件等，可登录

www.hxedu.com.cn 免费下载



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

国家示范性高职院校建设项目成果
高等职业教育教学改革系列规划教材·电子信息类

单片机原理及接口技术

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书以流行单片机芯片 AT89C51 为主体介绍了单片机的原理、常用单片机接口技术及应用。全书共 9 章，在硬件基础知识、汇编软程序设计的基础上，按照单片机产品的开发流程，介绍了调试工具软件 Keil C 与调试仿真工具软件 Protues。用 11 个实训将产品开发的流程理念充分贯穿于其中。达到在稳固掌握基础原理的基础上再应用与开发。融教、学、做一体于教材中。针对有些专业还专门设置了课程设计（或综合训练）。本书也特别编写了 8 个课题，方便老师与同学们完成该项训练。

本书内容精练，实例丰富，用实训来强化原理的基础知识，知识点与技能点相结合，既实现了知识的全面性和连贯性，又做到了理论与实践内容的融合贯通，体现了应用性人才培养的特点。特别适合作为高职高专院校电子信息类、计算机类、自动化类、机电类及控制类各专业的单片机技术课程教材，也可作为职工大学、函授大学、中职学校的教材及单片机应用开发人员的参考书。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有，侵权必究。

图书在版编目（CIP）数据

单片机原理及接口技术/陈贵银，祝福主编. —北京：电子工业出版社，2011.1

高等职业教育教学改革系列规划教材. 电子信息类

ISBN 978-7-121-12288-0

I. ①单… II. ①陈…②祝… III. ①单片微型计算机—基础理论—高等学校：技术学校—教材②单片微型计算机—接口—高等学校：技术学校—教材 IV. ①TP368.1

中国版本图书馆 CIP 数据核字（2010）第 222455 号

策划编辑：田领红

责任编辑：田领红 特约编辑：徐 岩

印 刷：北京市李史山胶印厂
装 订：

出版发行：电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本：787×1 092 1/16 印张：19.75 字数：506 千字

印 次：2011 年 1 月第 1 次印刷

印 数：4 000 册 定价：33.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至 zllts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：（010）88258888。

前 言

目前单片机已经渗透到了人们生活中的各个领域，导弹的导航装置、飞机上各种仪表的控制、计算机的网络通信与数据传输、工业自动化过程的实时控制和数据处理、广泛使用的各种智能 IC 卡、民用豪华轿车的安全保障系统、录像机、摄像机、全自动洗衣机的控制，以及程控玩具、电子宠物等，这些都离不开单片机，更不用说自动控制领域的机器人、智能仪表和医疗器械了。科技越发达，智能化的东西就越多，使用的单片机也就越多。因此，学习单片机技术越来越成为社会发展的需求。单片机原理及接口技术课程也成为高等学校重要的基础课或专业课。

全书共 9 章，按照基础理论—实训操作—总结的顺序，主要介绍 AT89C51 单片机的硬件基础知识、软程序设计、调试工具软件 Keil C 与软硬结合的调试仿真工具软件 Protues 的融合应用，将产品开发的流程理念充分贯穿于其中。第 1 章到第 5 章均是针对单片机最小应用系统的学习，第 6 章到第 8 章主要是针对常用接口部分的扩展应用学习，第 9 章专门介绍 2 个单片机应用的综合实例。从专用芯片到 Keil 的调试，从而完成软件设计，用工具软件 Protues 先进行产品的仿真，再用面包板（实验 PCB 或教学实验板）进行制作。教学时可融合于前面的每一章节中。本书从原理到应用，以应用实例为主，每个例子均已实践检测。每一章节均体现“教、学、做”的思想在里面，符合当前工学结合的理念与思路。

本教材是针对学生重点学习单片机基础并能快速入门产品开发，学会简单应用制作相应的单片机产品，并能有一定的扩展学习的资源空间接口部分。引入调试软件 Keil C 与单片机仿真软件 Protues 的相关应用，产品开发的流程理念贯穿于其中，达到在稳固掌握基础理念的基础上再应用与开发。每一章节中都配有练习，小结中阐明讲授重点，也就是学生学习的重点及方法指导。针对有些专业还专门设置了课程设计（或综合训练）。本书也特别编写了 8 个课题，方便老师与同学们完成该项训练。

通过对本教材的学习，将使读者达到以下目标。

- (1) 了解单片机的组成、内部结构和特点，获得其硬件和软件的必要基础知识；
- (2) 在初步掌握 MCS-51 单片机指令系统的基础上，掌握汇编语言程序的分析，能根据实际工作要求进行一般的程序设计和应用；
- (3) 能基本掌握 Keil（程序调试软件）与 Protues（虚拟软件与硬件的连机仿真）两个软件界面操作与应用；
- (4) 基本掌握单片机内部硬件资源和常用外围电路的初步应用方法；
- (5) 掌握中小型单片机应用电路的软硬件设计和调试。通过很多实例的讲解，能完全掌握整个单片机系统的开发过程。

本书可作为高职高专院校电子信息类、计算机类、自动化类、机电类及控制类各专业的单片机技术课程教材，也可作为职工大学、函授大学、中职学校的教材及单片机应用开发人员的参考书。

本书由陈贵银、祝福主编，鄢磊、郑火胜、何新洲副主编，章国华主审。第 3、4、5、9 章及课程设计和附录由陈贵银编写；第 1、7 章由祝福编写；第 2 章由何新洲编写；第 6 章由郑火胜与陈贵银编写；第 8 章由鄢磊编写，祝福对部分程序进行了调试工作；在此向关心和支持本书编写工作的人士表示衷心的感谢。

为便于教材使用，本书还配有多媒体课件、教材中有关例题的 Proteus 设计文件、汇编源程序及对应机器语言文件和习题答案。请有此需要的教师登录华信教育资源网 (www.hxedu.com.cn) 免费注册后再进行下载，如果有问题请在网站留言板留言或与电子工业出版社联系 (E-mail: tianlh@phei.com.cn)。

由于编者水平有限，不妥之处在所难免，真诚希望广大读者批评指正。

编 者

2010 年 8 月

目 录

第 1 章 单片机的基础知识	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 二进制数的算术运算	4
1.1.4 二进制编码	6
1.1.5 逻辑电路	7
1.2 单片机概论	9
1.2.1 嵌入式系统、单片机	9
1.2.2 单片机的发展概况	10
1.2.3 应用广泛的 AT89 系列单片机	11
1.3 单片机应用研发工具和教学实验装置	12
1.3.1 单片机应用研发工具	12
1.3.2 单片机课程教学实验装置	14
1.3.3 AT89C51 单片机研发工具操作演示	15
小结	15
练习题 1	15
第 2 章 AT89C51 单片机芯片的硬件结构	17
2.1 AT89C51 单片机的逻辑结构及信号引脚	17
2.1.1 AT89C51 单片机的结构框图	17
2.1.2 AT89C51 单片机芯片内部结构	18
2.1.3 AT89C51 的信号引脚	19
2.1.4 时钟与复位电路	21
2.2 AT89C51 的内部存储器	23
2.2.1 内部数据存储器低 128 单元	24
2.2.2 内部数据存储器高 128 单元	26
2.2.3 MCS-51 的堆栈	29
2.2.4 内部程序存储器	29
2.3 实训 1: 单片机复位、晶振、ALE 信号的观测	30
2.3.1 电路安装	30
2.3.2 信号观测	30
小结	31

练习题 2	32
第 3 章 单片机的指令系统及程序设计	34
3.1 基本概念	34
3.1.1 指令、指令系统、机器代码	34
3.1.2 程序、程序设计、机器语言	35
3.1.3 汇编语言、汇编语言指令格式	35
3.1.4 汇编（编译）和编程（固化）	36
3.1.5 汇编语言常用符号	37
3.2 MCS-51 单片机指令的寻址方式	38
3.2.1 MCS-51 单片机指令的寻址方式	38
3.2.2 MCS-51 单片机指令寻址方式小结	42
3.3 汇编语言的指令系统	42
3.3.1 数据传送指令	42
3.3.2 算术运算指令	46
3.3.3 逻辑运算类及移位类指令	51
3.3.4 控制转移类指令	55
3.3.5 位操作指令	60
3.4 MCS-51 单片机伪指令	61
3.4.1 常量和标号	61
3.4.2 常用伪指令	63
3.4.3 手工汇编和机器汇编	65
3.5 汇编语言程序设计	66
3.5.1 顺序程序设计	67
3.5.2 分支程序设计	69
3.5.3 循环程序设计	71
3.5.4 子程序设计	75
3.6 经典功能模块汇编语言程序设计实例	77
3.6.1 定时程序	77
3.6.2 查表程序	78
3.6.3 码制转换程序	79
3.6.4 算术运算程序	81
3.7 实训 2：软件调试仿真器 Keil u Vision 及其应用	82
3.7.1 Keil u Vision 快速入门	82
3.7.2 用 Keil 设计延时子程序并仿真调试	89
3.7.3 用 Keil 设计分支结构程序并仿真调试	90
3.7.4 用 Keil 设计查表程序并仿真调试	91
3.8 实训 3：跑马灯的 Proteus 设计与仿真	91

3.8.1	Proteus ISIS 窗口与基本操作	92
3.8.2	Proteus ISIS 库元件的认识	95
3.8.3	跑马灯 Proteus 电路原理图的设计	96
3.8.4	跑马灯 Proteus 与 Keil u Vision 的联调与仿真	101
	小结	102
	练习题 3	104
第 4 章	AT89C51 I/O 及其应用举例	108
4.1	I/O 口结构与负载能力	108
4.1.1	I/O 口结构与工作原理	108
4.1.2	I/O 口负载能力	112
4.2	实训 4: I/O 的简单应用	112
4.2.1	单片机控制数码管静态显示	112
4.2.2	单片机用开关控制的 LED 显示	115
4.2.3	单片机用开关控制数码管静态显示	117
4.3	单片机与矩阵式键盘的接口技术	119
4.3.1	基础知识	119
4.3.2	接口电路与程序设计	121
4.3.3	运行与思考	124
4.4	实训 5: I/O 口应用	124
4.4.1	单片机与 LED 数码管的动态显示	124
4.4.2	单片机与 LED 数码管的动态显示 Proteus 仿真	126
4.4.3	编程器的使用	127
4.4.4	制作实训及思考	129
	小结	129
	练习题 4	130
第 5 章	AT89C51 单片机的中断与定时	131
5.1	中断	131
5.1.1	数据传送控制方式	131
5.1.2	中断基本概念	132
5.2	MCS-51 单片机中断系统结构	132
5.2.1	MCS-51 的中断	132
5.2.2	与中断控制有关的寄存器	134
5.2.3	中断响应过程	137
5.2.4	中断程序设计举例	139
5.3	MCS-51 的定时器/计数器	147
5.3.1	定时方法	147

5.3.2	定时和计数功能	147
5.3.3	定时/计数器的控制	148
5.3.4	定时器/计数器初值的计算	150
5.3.5	定时器/计数器的工作方式	151
5.3.6	定时器/计数器的应用举例	153
5.4	实训 6: 定时/计数器与中断综合应用举例	158
5.4.1	基于 AT89C51 的 60s 倒计时装置的设计与仿真	158
5.4.2	基于 AT89C51 的按键发声装置的设计与仿真	161
	小结	163
	练习题 5	164
第 6 章	AT89C51 单片机存储器的扩展技术	166
6.1	存储器概述	166
6.1.1	存储器的类型	166
6.1.2	存储器的性能指标与分级结构	167
6.1.3	常用的只读存储器芯片	169
6.1.4	常用的随机读/写存储器芯片	171
6.2	MCS-51 单片机存储器的扩展	172
6.2.1	AT89C51 存储器扩展的三总线	172
6.2.2	扩展存储器的编址技术	173
6.2.3	程序存储器的扩展	175
6.2.4	数据存储器的扩展	177
6.2.5	存储器的综合扩展	179
6.3	实训 7: 用 E ² PROM 扩展 AT89C51 单片机 ROM、RAM	180
6.3.1	E ² PROM 存储器 2864A	180
6.3.2	E ² PROM 扩展 ROM、RAM 电路设计	181
6.3.3	E ² PROM 扩展 ROM、RAM 程序设计	181
6.3.4	运行与思考	182
	小结	183
	练习题 6	183
第 7 章	AT89C51 单片机的串行接口技术	185
7.1	AT89C51 单片机之间的串行通信接口技术及仿真	185
7.1.1	串行通信的基本原理	185
7.1.2	AT89C51 串行通信基础知识	188
7.1.3	硬件接口电路设计	190
7.1.4	软件接口程序设计	191
7.1.5	单片机之间的串行通信 Protues 仿真	195

7.2	AT89C51 单片机与 PC 机间的通信接口技术及仿真	196
7.2.1	RS-232C 总线标准	196
7.2.2	硬件接口电路设计	197
7.2.3	软件接口程序设计	198
7.2.4	单片机与 PC 机间的串行通信 Protues 仿真	199
	小结	201
	练习题 7	201
第 8 章	单片机 I/O 扩展及应用	203
8.1	可编程通用并行接口 8255A	203
8.1.1	8255A 的组成与接口信号	203
8.1.2	8255A 的工作方式和控制字	205
8.1.3	三种工作方式的功能	206
8.1.4	实训 8: 用 8255 设计交通信号灯管理仿真	208
8.2	可编程的多功能接口 8155	211
8.2.1	8155 的组成及接口信号	211
8.2.2	8155 的命令状态字	212
8.2.3	8155 与 MCS-51 单片机的连接	214
8.3	键盘/显示控制寄存器 8279	216
8.3.1	8279 的组成与接口信号	216
8.3.2	8279 的操作命令	219
8.3.3	实训 9: 8279 在键盘/显示接口中的应用仿真	223
8.4	ADC0809 (ADC0808) 与 DAC0832 的接口技术	226
8.4.1	ADC0809 的基础知识	227
8.4.2	实训 10: ADC0808 接口电路与程序设计仿真	228
8.4.3	DAC0832 的基础知识	231
8.4.4	实训 11: 单片机与 DAC0832 的接口技术仿真	233
8.5	单片机控制步进电机接口技术	235
8.5.1	基础知识	235
8.5.2	接口电路设计与仿真	236
8.5.3	运行与思考	238
8.6	单片机控制直流电机的接口技术	239
8.6.1	基础知识	239
8.6.2	接口电路设计与仿真	240
8.6.3	运行与思考	242
8.7	单片机与字符型 LCD 显示器的接口技术	243
8.7.1	基础知识	243
8.7.2	接口的程序设计与仿真	245

8.7.3 运行与思考	248
小结	248
练习题 8	249
第 9 章 单片机实际应用制作	250
9.1 基于单片机和 DS1302 的电子时钟	250
9.1.1 功能与操作	252
9.1.2 电子时钟的硬件设计	253
9.1.3 电子时钟的软件设计	253
9.1.4 技术要点	259
9.1.5 电子时钟的 Protues 仿真	260
9.1.6 电路安装及现象观察	261
9.2 基于单片机和 DS18B20 的数字温度计	261
9.2.1 功能与操作	264
9.2.2 数字温度计的硬件设计	264
9.2.3 数字温度计的软件设计	264
9.2.4 数字温度计的 Protues 仿真	269
9.2.5 电路安装及现象观察	270
小结	270
课程设计项目任务书	271
课题 1: 简易的交通信号灯控制器设计与制作	271
课题 2: 定时闹铃的仿真、设计与制作	275
课题 3: 直流电机 PWM 调速控制器的设计、仿真与制作	278
课题 4: 数控电源的设计、仿真与制作	280
课题 5: 数字电压表的设计、仿真与制作	281
课题 6 单片机间的多机通信的设计、仿真与制作	283
课题 7: LCD 简易时钟的设计、仿真与制作	284
课题 8: LCD 密码锁的设计、仿真与制作	287
课程设计报告参考内容	291
附录 A AT89S51 相对 AT89C51 增加的功能	294
附录 B ASCII 码表	300
附录 C MCS-51 指令表	301
参考文献	305

第 1 章

单片机的基础知识

【内容提要】

本章主要介绍了计算机中的数和编码、单片机的发展概况及主芯片 AT89C51 单片机、单片机应用研发工具和教学实验装置等相关内容；在计算机中的数和编码中介绍了计算机中的数制、符号数的表示、二进制数的算术运算、逻辑电路、二进制的编码等数的基础理论知识；在单片机的概论中介绍了单片机主流系列及其相关组成部件的发展；在单片机应用研发工具和教学实验装置中介绍了常用的开发工具及开发流程。本章为后面学习 MCS-51 系列的单片机奠定了坚实的基础。

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数几乎都是采用二进制表示。但是，二进制数书写起来太长，且不便阅读和记忆。目前大部分微型机是 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，所以微型机广泛采用十六进制数来缩写二进制数。十六进制数用 0~9、A~F 共 16 个数码表示十进制数 0~15。1 个 8 位的二进制数用 2 位十六进制数表示，1 个 16 位的二进制数用 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，需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。它们之间的关系见表 1-1。

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

为了区别上述 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) 或不带字母表示十进制数制，H (hexadecimal) 表示十六进制数制。

1. 二进制数和十六进制数间的相互转换

二进制整数转换为十六进制数，只需从右向左将二进制数分为每 4 位 1 组，每组用 1 位十六进制数表示，左边不足 4 位应在左边加 0，以凑成 4 位 1 组。例如：

$$1\ 1111\ 1101\ 0110\text{B} \rightarrow 0001\ 1111\ 1101\ 0110\text{B} = 1\text{FD6H}$$

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。例如：

$$3\text{AC8H} = 0011\ 1010\ 1100\ 1000\text{B}$$

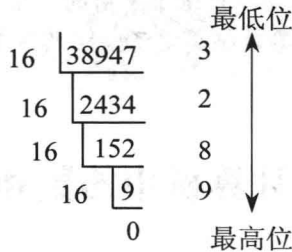
2. 十六进制数和十进制数间的相互转换

$$1\text{F3DH} = (1 \times 16^3) + (15 \times 16^2) + (3 \times 16^1) + (13 \times 16^0) = 4096 + 3840 + 48 + 13 = 7997$$

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。

如下式所示：

即 $38947 = 9823\text{H}$



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，较常用的有原码、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，通常称前者为真值，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，这样表示的数称为原码。例如：

$$X_1 = +105 = +1101001\text{B} \quad [X_1]_{\text{原}} = 01101001\text{B}$$

$$X_2 = -105 = -1101001B \quad [X_2]_{原} = 11101001B$$

其中，最高位为符号位，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。

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，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）。例如：

$$\begin{aligned} X_1 = +105 = +1101001B & \quad [X_1]_{反} = 01101001B \\ X_2 = -105 = -1101001B & \quad [X_2]_{反} = 10010110B \end{aligned}$$

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。负数求补码方法有两种，分别如下：

方法一：该数的反码加 1，可以得到该数的补码。例如：

$$\begin{aligned} X_1 = +105 = +1101001B & \quad [X_1]_{补} = 01101001B \\ X_2 = -105 = -1101001B & \quad [X_2]_{补} = 10010111B \end{aligned}$$

方法二：把与其绝对值相等的正数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。例如：

$$[-105]_{补} = 10010111B$$

对其求补，从右向左扫描，第一位就 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，而需将该数求补，才得到它的绝对值。

当数采用补码表示时，就可以把减法转换为加法。例如：

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

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：

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

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。例如：

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

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，得其真值为-0010 0010B，即-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，化作补码相加就不会有进位。

1.1.3 二进制数的算术运算

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~(2ⁿ-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。例如：

$$127+160=7\text{FH}+0\text{A0H}$$

为了区分数字和符号，写字母开头的十六进制数，前面应添加一个 0，如 0A0H。

$$\begin{array}{r}
 0111\ 1111 \\
 + 1010\ 0000 \\
 \hline
 10001\ 1111 = 11FH = 256 + 16 + 15 = 287
 \end{array}$$

↑ 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。例如：

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

反过来相减，即 10-192，运算过程如下：

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

↑ 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：Cy=0（无借位），结果为正；Cy=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。例如：

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

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 0110 0101B 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（从第 0 位开始计）0 看做符号位，也看做数值而将进位看做数的符号，结果为 -1001 1011B=-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，即

(1) 进位或借位是指无符号数运算结果的最高位向更高位进位或借位。通常多位二进制数将其拆成两部分或三部分或更多部分进行运算时，数的低位部分均无符号位，只有最高部分的最高位才为符号位。运算时，低位部分向高位部分进位或借位。由此可知，进位主要用于无符号数的运算，这与溢出主要用于符号数的运算是有区别的。

(2) 溢出与补码运算中的进位丢失也应加以区别，例如：

$$\begin{array}{r}
 -50-5=55 \\
 1100\ 1110 \\
 +\ 1111\ 1011 \\
 \hline
 1\ 1100\ 1001 = -0011\ 0111B = -55 \\
 \uparrow \text{进位}
 \end{array}$$

两个负数相加，结果为负数是正确的。这里虽然出现了补码运算中产生的进位，但由于和数并未超出 8 位二进制数 -128~127 的范围，因此无溢出。那么，如何来判别有无溢出呢？

设符号位向进位位的进位为 C_y ，数值部分向符号位的进位为 C_s ，则溢出：

$$O = C_y \oplus C_s$$

$O=1$ ，有溢出； $O=0$ ，无溢出。

下面用 M 、 N 两数相加来证明。设 MS 和 NS 为两个加数的符号位， RS 为结果的符号位，则表 1-2 所列的真值表。由真值表得逻辑表达式：

$$O = C_y \oplus C_s$$

表 1-2 符号、进位、溢出的真值表

MS	NS	RS	C_y	C_s	O
0	0	0	0	0	0
0	0	1	1	0	1
0	1	0	1	1	0
0	1	1	0	0	0
1	0	0	1	1	0
1	0	1	0	0	0
1	1	0	0	1	1
1	1	1	1	1	0

再来看 $105+50$ 、 $-105-50$ 和 $-50-5$ 三个运算有无溢出。

$ \begin{array}{r} 0110\ 1001 \\ +\ 0011\ 0010 \\ \hline 1001\ 1011 \\ C_y=0\ C_s=1 \\ O=0 \oplus 1=1 \text{ 有溢出} \end{array} $	$ \begin{array}{r} 1001\ 0111 \\ +\ 1100\ 1110 \\ \hline 1\ 0110\ 0101 \\ C_y=1\ C_s=0 \\ O=1 \oplus 0=1 \text{ 有溢出} \end{array} $	$ \begin{array}{r} 1100\ 1110 \\ +\ 1111\ 1011 \\ \hline 1\ 1100\ 1001 \\ C_y=1\ C_s=1 \\ O=1 \oplus 1=1 \text{ 无溢出} \end{array} $
--	---	---

1.1.4 二进制编码

如上所述，计算机中数是用二进制表示的，而计算机又应能识别和处理各种字符，如大小写英文字母、标点符号、运算符号等。这些又如何表示呢？在计算机里，字母、各种符号及指挥计算机执行操作的指令，都是用二进制数的组合来表示的，称为二进制编码。