21世纪高职高专

21世纪高职高专电子信息类实用规划教材

单片机原理及应用（C51语言）

邓立新 主 编
董国增 曹月真 副主编

免费赠送电子课件

- 以89C51为典型机型，以Keil C51作为主要编程和调试工具，由浅入深地讲解了单片机的工作原理及应用技术等。
- 以项目为载体，以提高学生实际应用能力为目的，适当降低理论难度，丰富实践内容，将合适的应用实例与具体知识点相融合，做到学以致用。
- 将单片机技术的硬件和软件、理论和实践、情境化设计项目等进行了有机地结合，使读者能较完整地学习单片机技术及开发工具的使用，实现了教、学、做三合一。

清华大学出版社

21 世纪高职高专电子信息类实用规划教材

单片机原理及应用(C51 语言)

邓立新　主　编
董国增　曹月真　副主编

清华大学出版社
北 京

内 容 简 介

本书以89C51为典型机型，结合大量实例，并以Keil C51作为主要编程和调试工具，由浅入深地讲解了单片机的工作原理及应用技术。全书共分为9章，主要内容包括：单片机基础知识与数制编码、单片机汇编语言程序设计、单片机C51语言程序设计、中断系统与定时/计数器、串行接口、单片机接口技术、单片机应用系统开发。本书将单片机技术的硬件和软件、理论和实践、情境化设计项目等进行了有机的结合，使读者可以在接近实际开发的过程中较完整地学习单片机技术及开发工具的使用，实现了教、学、做的合一。

本书可作为高职高专、中等职业学校电类专业“单片机原理及应用”课程以及实践的教学用书，同时也非常适合自学单片机的读者使用。

图书在版编目(CIP)数据

单片机原理及应用(C51语言)/邓立新主编；董国增，曹月真副主编. --北京：清华大学出版社，2012
(21世纪高职高专电子信息类实用规划教材)
ISBN 978-7-302-28686-8

Ⅰ. ①单…　Ⅱ. ①邓…②董…③曹…　Ⅲ. ①单片微型计算机—C语言—程序设计—高等职业教育—教材　Ⅳ. ①TP368.1②TP312

中国版本图书馆CIP数据核字(2012)第084506号

责任编辑：李春明　郑期彤
封面设计：杨玉兰
责任校对：周剑云
责任印制：李红英

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦A座　　邮　　编：100084
社 总 机：010-62770175　　邮　　购：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
课 件 下 载：http://www.tup.com.cn,010-62791865
印 装 者：北京密云胶印厂
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：16.5　　**字　　数**：396千字
版　　次：2012年8月第1版　　**印　　次**：2012年8月第1次印刷
印　　数：1～4000
定　　价：32.00元

产品编号：043743-01

前　言

为了适应我国高职高专教学改革的需要，并结合高职高专学生的学习特点，使学生做到学有所得，本书采取了“以项目为载体”的编写思路，以及以项目引导教学的编写原则。经过与多所高职高专教学一线教师的深入切磋，我们对教学内容进行了整合，将合适的应用实例与具体知识点相融合，尽量做到学以致用，并适当降低了理论难度，丰富了实践内容。

本书以常用的 89C51 为典型机型讲述单片机原理及接口技术，以提高学生实际应用能力为目的丰富了常用串行口芯片扩展的内容。书中还利用一章的篇幅具体对两个设计实例进行了详尽的实施过程描述，可使学生熟悉设计过程中的思路与方法。本书采用 C51 语言作为设计语言，为学生能够早日掌握单片机的实际开发技术做出了较好的铺垫。

本书共分为 9 章，第 1、2 章介绍单片机的认知和初步应用，第 3、4 章介绍单片机的汇编语言及 C51 语言程序设计，第 5～7 章介绍单片机的中断系统、定时/计数器和串行接口，第 8、9 章介绍单片机接口技术和单片机应用系统开发。附录中的 ASCII 码表、51 单片机汇编指令速查表、Keil C51 常用库函数和 Keil C51 常见错误警告提示信息可供查询。全书参考学时为 84 学时，第 3 章可作为选学内容。

本书可作为高职高专、中等职业学校电类专业“单片机原理及应用”课程以及实践的教学用书，同时也非常适合自学单片机的读者作为参考。

本书由承德石油高等专科学校邓立新担任主编，承德石油高等专科学校董国增、衡水职业技术学院曹月真任副主编。具体编写分工如下：第 1 章、第 5 章、第 6 章和第 7 章由曹月真编写，第 2 章和第 3 章由钱彬编写，第 4 章由邓志辉编写，第 8 章由董国增编写，第 9 章和附录由邓立新编写。

由于作者水平及能力有限，加之时间仓促，书中难免出现错误和不妥之处，恳请读者批评指正，并欢迎来函来电探讨，一并感谢！作者 E-mail：dlx6969@126.com。

编　者

目　录

第 1 章

单片机的认知

教学目标

本章主要对当前几种主流单片机及其应用进行了较为详细的介绍，并简要介绍了计算机中的数制和编码的概念。通过本章的学习，读者应初步了解单片机的基本情况，熟练掌握二进制数、十进制数和十六进制数之间的相互转换及其 C 语言的表示方法，了解计算机中数的表示方法，并认识几种常用的编码。

1.1 单片机概述

1.1.1 单片机的概念

单片机又称单片微控制器，它不是完成某一个逻辑功能的芯片，而是集成了微处理器、存储器、输入/输出控制等接口电路的芯片。概括地讲：一块单片机芯片就是一台计算机。这种计算机可以作为一个信息处理部件，嵌入到应用系统和设备中，执行特定数据的处理和控制任务。对于嵌入式计算机，用户首先要对其进行编程，计算机再按照用户事先编制的程序，根据系统要求完成相应的数据处理和控制功能。

1.1.2 主流单片机介绍

51 系列单片机源于 Intel 公司的 MCS-51(Micro Controller System-51)系列单片机，这一系列单片机包括很多种类，如基本型的 8031、8051、8751、8951 和增强型的 8032、8052、8752、8952 等。在 Intel 公司将 MCS-51 系列单片机实行技术开放之后，许多公司，如 Atmel、Philips、NXP、Analog Devices、Dallas、Siemens、华邦、LG 以及我国的宏晶科技等都以 MCS-51 中的基础结构 8051 为基核推出了许多各具特色、具有优异性能的单片机。我们把以 8051 为基核推出的各种型号的兼容型单片机统称为 51 系列单片机。

Intel 公司 MCS-51 系列单片机的主要产品及其性能如表 1-1 所示，Atmel 公司的 AT89 系列单片机的主要产品及其性能如表 1-2 所示。

表 1-1 MCS-51 系列单片机的主要产品及其性能

子 系 列	芯片型号	片内存储器类型及容量		I/O 口/位	UART 串口/个	中断源/个	定时/计数器/个
		ROM/EPROM	RAM				
8X51/52 系列	8031	无	128B	32	1	5	2
	8051	4KB ROM	128B	32	1	5	2
	8052	8KB ROM	256B	32	1	6	3
8XC51/52 系列	80C31	无	128B	32	1	5	2
	80C51	4KB ROM	128B	32	1	5	2
	80C52	8KB ROM	256B	32	1	6	3
8X54/58 系列	80C54	16KB ROM	256B	32	1	6	3
	87C54	16KB EPROM	256B	32	1	6	3
	80C58	32KB ROM	256B	32	1	6	3
	87C58	16KB EPROM	256B	32	1	6	3+5PCA

表 1-2　Atmel 公司的 AT89 系列单片机的主要产品及其性能

子 系 列	芯片型号	片内存储器类型及容量		I/O 口/位	UART 串口/个	中断源/个	定时/计数器/个
		Flash ROM	RAM				
8 位 Flash 系列	AT89C51	4KB	128B	32	1	5	2
	AT89C52	8KB	256B	32	1	5	3
	AT89C51RC	32KB	512B	32	1	6	3
	AT89C1051	1KB	64B	15	1	—	2
	AT89C2051	2KB	128B	15	1	—	2
	AT89C4051	4KB	256B	15	1	—	2
ISP_Flash 系列	AT89S51	4KB	128B	32	1	5	2
	AT89S52	8KB	256B	32	1	5	3
I^2C_Flash 系列	AT89C51RB2	16KB	256B	32	1	6	3
	AT89C51ED2	32KB	256B	44	1	9	3

目前市场上的主流单片机有 51 系列、AVR 系列、PIC 系列，其中应用最广泛的 8 位单片机首推 Intel 公司的 51 系列。由于其产品硬件结构合理，指令系统规范，加之生产历史“悠久”，许多著名的芯片公司都购买了 51 系列芯片的核心专利技术，并在其基础上进行性能上的扩充，使芯片得到进一步的完善，形成了一个庞大的体系。

AVR 系列单片机是 Atmel 公司推出的较为新颖的单片机，其显著的特点为高性能、高速度、低功耗。它以时钟周期为指令周期，实行流水作业。在 51 系列中，所有的逻辑运算都必须在累加器 A 中进行，而 AVR 系列却可以在任意两个寄存器之间进行。

PIC 系列单片机是美国微芯公司(Microchip)的产品，是当前市场份额增长最快的单片机之一。其 CPU 采用 RISC 结构，分别有 33、35、58 条指令(视单片机的级别而定)，属精简指令集。而 51 系列有 111 条指令，AVR 系列有 118 条指令，都比 PIC 系列复杂。

1.1.3　单片机的应用

51 系列单片机凭借其品种多、体积小、兼容性强、性能价格比高、应用软件齐全、技术成熟等优势渗透到仪器仪表、家用电器、医用设备、航空航天、专用设备的智能化管理及过程控制等领域，几乎所有需要控制、通信和智能的领域，都可以找到单片机的身影。

1．在智能民用产品中的应用

从电视、电话、电冰箱、洗衣机、空调机、电饭煲到智能玩具、游戏机、收银机、家用防盗报警器、电子秤及其他音响视频器材等，单片机无所不在。单片机控制器的引入使产品功能大大增强，性能得到很大提高。

2．在智能仪器仪表上的应用

单片机具有体积小、功耗低、控制功能强、扩展灵活、微型化和使用方便等优点，广泛应用于仪器仪表中，结合不同类型的传感器，可实现对电压、功率、频率、湿度、温度、

流量、速度、厚度、角度、长度、硬度、元素、压力等物理量的测量。采用单片机控制可使仪器仪表数字化、智能化、综合化、柔性化、微型化，且功能比采用模拟或数字电路更加强大，例如功率计、示波器、各种分析仪等精密的测量设备。

3．在工业控制中的应用

单片机作为机电一体化设备控制器，可以简化机械产品的结构设计，实现智能生产和操作控制，并扩展原有设备的功能，还可以构成形式多样的控制系统、数据采集系统。例如车床、铣床、数控机床、工厂流水线的智能化管理，电梯智能化控制、各种报警系统，与计算机联网构成二级控制系统等。

4．在计算机网络和通信领域中的应用

单片机依靠串口、并口或者高速 USB 口等通信接口，可以很方便地与计算机进行数据通信，为在计算机网络和通信设备间的应用提供了极好的物质条件。现在的通信设备基本上都实现了单片机智能控制，例如手机、电话机、小型程控交换机、楼宇自动通信呼叫系统、列车无线通信、无线电对讲机等。

5．在医用设备领域中的应用

单片机在医用设备中的用途也相当广泛，例如医用呼吸机、分析仪、监护仪、超声诊断设备及病床呼叫系统等。

6．在各种大型电器中的模块化应用

某些专用单片机设计用于实现特定功能，从而在各种电路中进行模块化应用，而不要求使用人员了解其内部结构。例如音乐集成单片机，看似功能简单，微缩在纯电子芯片中(有别于磁带机的原理)，但需要复杂的类似于计算机的原理，如音乐信号以数字的形式存储于存储器中(类似于 ROM)，由微控制器读出，再转化为模拟音乐电信号(类似于声卡)。

在大型电路中，这种模块化应用极大地缩小了体积，简化了电路，降低了损坏、错误率，也便于更换。

此外，单片机在工商、金融、科研、教育、国防、航空航天等领域也都有着十分广泛的用途。

1.2 数制与编码

1.2.1 数制

1．二、八、十六进制数转换为十进制数

1) 二进制数转换为十进制数

二进制用阿拉伯数字 0、1 来表示，进位原则是逢二进一。

二进制数第 0 位的权值为 2 的 0 次方，第 1 位的权值为 2 的 1 次方……所以，设有一个二进制数 01011101，转换为十进制数为

$$1\times2^0+0\times2^1+1\times2^2+1\times2^3+1\times2^4+0\times2^5+1\times2^6+0\times2^7=93$$

0 与多少相乘都是 0，所以我们也可以只计算值为 1 的位，即

$$1\times2^0+1\times2^2+1\times2^3+1\times2^4+1\times2^6=93$$

【例 1.1】 将下列二进制数转换为十进制数。

(1) 101001　　(2)1110　　(3)1101.11　　(4)10.1010

解：(1) $1\times2^0+1\times2^3+1\times2^5=41$

(2) $1\times2^1+1\times2^2+1\times2^3=14$

(3) $1\times2^0+1\times2^2+1\times2^3+1\times2^{-1}+1\times2^{-2}=13.75$

(4) $1\times2^1+1\times2^{-1}+1\times2^{-3}=2.625$

2) 八进制数转换为十进制数

八进制用 0～7 八个数来表示，进位原则是逢八进一。

八进制数第 0 位的权值为 8 的 0 次方，第 1 位的权值为 8 的 1 次方，第 2 位的权值为 8 的 2 次方……所以，设有一个八进制数 357，转换为十进制为

$$7\times8^0+5\times8^1+3\times8^2=239$$

3)　十六进制数转换为十进制数

十六进制用 0～9 十个数字及 A、B、C、D、E、F 六个字母来表示，字母不区分大小写，进位原则是逢十六进一。

十六进制数第 0 位的权值为 16 的 0 次方，第 1 位的权值为 16 的 1 次方……所以，在第 N 位上，数值为 M 的数的大小可表示为 $M\times16^N$。

例如，设有一个十六进制数 4F5，转换为十进制数为

$$5\times16^0+F\times16^1+4\times16^2=1269(F\text{ 表示 }15)$$

总之，所有进制换算为十进制，关键在于各自的权值不同，例如十进制的 4321，我们可以写成 $1\times10^0+2\times10^1+3\times10^2+4\times10^3=4321$。

【例 1.2】 将下列十六进制数转换为十进制数。

(1) 4F5　　(2) 3BC　　(3)3BEF　　(4) 90.1

解：(1) $5\times16^0+15\times16^1+4\times16^2=1269$

(2) $12\times16^0+11\times16^1+3\times16^2=956$

(3) $15\times16^0+14\times16^1+11\times16^2+3\times16^3=15343$

(4) $9\times16^1+1\times16^{-1}=144.0625$

2．十进制数转换为二、八、十六进制数

1)　十进制数转换为二进制数

十进制数转换为二进制数时，由于整数和小数转换方法不同，所以先将十进制数的整数部分和小数部分分别转换后，再加以合并。

十进制整数转换为二进制整数采用“除 2 取余，逆序排列”的方法。具体过程是：将要转换的数除以 2，得到商和余数；将商继续除以 2，直到商为 0；最后将余数倒序排列，得到的数就是转换结果。

十进制小数转换为二进制小数采用“乘 2 取整，正序排列”的方法。具体过程是：将要转换的小数乘以 2，取出积的整数部分，再用余下的小数部分继续乘以 2，再取出积的整

数部分，如此进行，直到积的小数部分为零或者达到所要求的精度为止。

【例 1.3】 将十进制数 55 转换为二进制数。

解：

2 | 55 … 余 1，k_0=1

2 | 27 … 余 1，k_1=1

2 | 13 … 余 1，k_2=1

2 | 6 … 余 0，k_3=0

2 | 3 … 余 1，k_4=1

2 | 1 … 余 1，k_5=1

0

即得结果为 110111。

【例 1.4】 将十进制数 0.3125 转换为二进制数。

解： 0.3125

$\times \quad 2$

0.6250 …取0，$k_{-1}=0$

$\times \quad 2$

1.2500 …取1，$k_{-2}=1$

0.2500

$\times \quad 2$

0.5000 …取0，$k_{-3}=0$

$\times \quad 2$

1.0000 …取1，$k_{-4}=1$

即得结果为 0.0101。

需要说明的是，有些十进制小数无法准确地用二进制进行表达，所以转换时符合一定的精度即可，这也是为什么计算机的浮点数运算不准确的原因。

2) 十进制数转换为八进制数

十进制数转换为八进制数的方法同十进制数转换为二进制数。即以小数点为界，整数部分除以 8，然后取每次得到的商和余数，用商继续和 8 相除，直到商为零。然后将第一次得到的余数作为八进制的第 1 位，第二次得到的余数作为八进制的第 2 位，以此类推，最后得到的余数作为八进制的最高位。小数部分则先乘以 8，然后获得运算结果的整数部分，再将结果中的小数部分再次乘以 8，直到小数部分为零。然后把第一次得到的整数部分作为八进制小数的最高位，后续的整数部分依次作为低位，这样由各整数部分组成的数字就是转化后的八进制小数的值。

【例 1.5】 将十进制数 93 转换为八进制数。

解：

8 | 93 … 余 5，k_0=5

8 | 11 … 余 3，k_1=3

8 | 1 … 余 1，k_2=1

0

即得结果为 135。

【例 1.6】 将十进制数 0.3125 转换为八进制数。

解：0.3125

$$\begin{array}{r} \times \quad 8 \\ \hline 2.5000 \end{array} \quad \cdots \text{ 取2，} k_{-1}=2$$

$$0.5000$$

$$\begin{array}{r} \times \quad 8 \\ \hline 4.0000 \end{array} \quad \cdots \text{ 取4，} k_{-2}=4$$

即得结果为 0.24。

3) 十进制数转换为十六进制数

十进制数转换为十六进制数的方法同十进制数转换为二进制数。即以小数点为界，整数部分除以 16，然后取每次得到的商和余数，用商继续和 16 相除，直到商为零。然后把第一次得到的余数作为十六进制的第 1 位，第二次得到的余数作为十六进制的第 2 位，以此类推，最后得到的余数作为十六进制的最高位。小数部分则先乘以 16，然后获得运算结果的整数部分，再将结果中的小数部分再次乘以 16，直到小数部分为零。然后把第一次得到的整数部分作为十六进制小数的最高位，后续的整数部分依次作为低位，这样由各整数部分组成的数字就是转化后的十六进制小数的值。

【例 1.7】 将十进制数 93 转换为十六进制数。

解：

$$\begin{array}{r|l} 16 & 93 \quad \cdots \text{ 余 13，} k_0=\text{D} \\ \hline 16 & 5 \quad \cdots \text{ 余 5，} k_1=5 \\ \hline & 0 \end{array}$$

即得结果为 5D。

【例 1.8】 将十进制数 0.3125 转换为十六进制数。

解：0.3125

$$\begin{array}{r} \times \quad 16 \\ \hline 5.0000 \end{array} \quad \cdots \text{ 取5，} k_{-1}=5$$

即得结果为 0.5。

1.2.2 数制的 C 语言表述

1. 数制的 C 语言表述方法

数据在内存中以二进制补码形式存放。之所以用二进制形式存放是因为计算机容易识别二进制数，只用 0 和 1 表示，编码、译码速度快，存取速度快。但是二进制数太长了，没有人愿意面对很长的一串 0、1 组合进行思考或者操作。所以，C 语言没有提供输入二进制数的格式控制符。也就是说我们不能直接输入/输出二进制数。但是，C 语言同样可以表示二进制数，只是数据的存储方式我们看不到而已。

用八进制或者十六进制就可以解决这个问题。因为，进制越大，数的表示长度也就越短。又因为 2、8、16 分别是 2 的 1 次方、3 次方、4 次方，这一点使得三种进制之间可以非常直

接地相互转换。八进制数或十六进制数缩短了二进制数，但保持了二进制数的表达特点。

如果不使用特殊的书写形式，十六进制数与十进制数就会混淆。例如，9654 这个数就看不出是十进制数还是十六进制数。

同样，如果一个数是 856，我们可以断定它不是八进制数，因为八进制数中不可能出现 7 以上的阿拉伯数字。但如果这个数是 321 或 01234567，那么它可能是八进制数也可能是十进制数。

所以，C 语言规定，一个数如果要指明它采用八进制，必须在其前面加上一个 0，如 321 是十进制数，但 0123 则表示八进制数。这就是八进制数在 C 语言中的表达方法。

另外，在 C 语言中，十六进制数必须以 0x 开头(0x 中的 0 是数字 0，不是字母 o，其中的 x 不区分大小写)。例如 0x123 表示一个十六进制数，而 123 则表示一个十进制数。那么，对于同样一个数，我们在代码中可以用十进制表示，也可以用十六进制表示。例如在变量初始化时有

```
int a=100;
```

我们可以这样写：

```
int a = 0144; //0144 是用八进制形式表示的十进制数 100
```

还可以这样写：

```
int a=0x64;      //0x64 是用十六进制形式表示的十进制数 100
```

2. 八进制数、十六进制数在转义字符中的使用

前面提到，在 C 语言中用八进制表达一个数时，不能少了前面那个 0，否则计算机会认为其是十进制数。不过，在用于表达转义符的时候却不能加 0。C 语言中，我们常常用一个转义符‘\’加上一个特殊字母来表示某个字符，如‘\n’表示换行，而‘\t’表示 Tab 字符。转义符‘\’后面加上一个八进制数，用于表示 ASCII 码等于该值的字符。

比如，通过查询 ASCII 码表，可得知叹号字符‘!’的 ASCII 值是 33，那么可以把它转换为八进制数，即 41，然后用 ‘\41’来表示‘!’。由于是八进制数，所以本应写成‘\041’，但因为 C 语言规定不允许使用斜杠加十进制数的方式来表示字符，所以这里的 0 可以不写。

转义符也可以加上一个十六进制数来表示一个字符。于是，‘!’字符可以有以下表达方式：

```
'!'       //直接输入字符
'\41'     //用八进制表示，此时可以省略开头的 0
'\0x21' //用十六进制表示
```

一般，除了空字符用八进制数‘\0’表示以外，我们很少使用八进制数和十六进制数来表示一个字符。

1.2.3　原码、反码和补码

在数字中，数的正负是用“+”、“-”来表示的。在计算机中，数的正负在最高位分别用“0”和“1”表示。8 位微型计算机中位数的约定一般为：最高位 D7 表示正/负号，其中，D7=0 表示正数，D7=1 表示负数；其他 7 位表示数值，如图 1-1 所示。

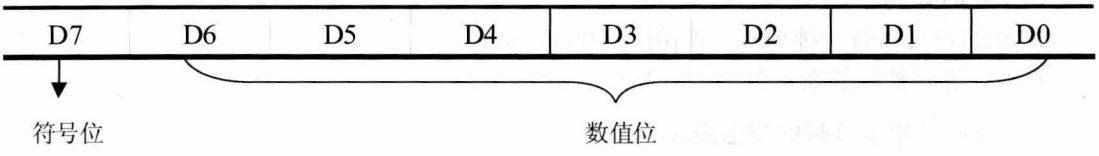

图 1-1　8 位有符号数的表示方法

例如，X_1=+1010110 B，X_2=−1011010 B，则在计算机中，X_1=01010110 B，X_2=11011010 B。为了区别原来的数与其在计算机中的表示形式，我们将数码化了的带符号数称为机器数，而把原来的数称为机器数的真值。

在计算机内，有符号数有 3 种表示法：原码、反码和补码。

1．原码

原码是将正数在符号位用 0 表示，负数在符号位用 1 表示，数值位保持不变的表示法。

1)　正数

正数的原码与原来的数相同。例如，$[+5]_{原}$= 0 0000101 B。

2)　负数

负数的原码是符号位为 1，数值位不变。例如，$[-5]_{原}$= 1 0000101 B

3)　0 的原码

数 0 的原码有两种形式，正 0 和负 0。

$[+0]_{原}$=0 0000000 B

$[-0]_{原}$=1 0000000 B

由于最高位为符号位，所以 8 位二进制原码的表示范围为−127～+127。

2．反码

1)　正数

正数的反码与原码相同。例如，$[+7]_{反}$= 0 0000111 B

2)　负数

负数的反码是符号位为 1，数值位按位取反。例如，$[-7]_{反}$= 1 1111000 B。

3)　0 的反码

数 0 的反码有两种形式，正 0 和负 0。

$[+0]_{反}$=0 0000000 B

$[-0]_{反}$=1 1111111 B

8 位二进制反码的表示范围为−127～+127。

3．补码

1) 正数

正数的补码与原码相同。例如，$[+7]_{补}$= 0 0000111 B

2) 负数

负数的补码是符号位为 1，数值位按位取反后再在最低位加 1，也就是“反码+1”。例如，$[-7]_{补}$=1 1111001 B。

3) 0 的补码

数 0 的补码只有一种形式，即$[0]_{补}$=0 0000000 B。

8 位二进制补码的表示范围为-128～+127。

4．有符号数运算时的溢出问题

如果计算机的字长为 n 位，n 位二进制数的最高位为符号位，其余 n-1 位为数值位，采用补码表示法时，可表示的数 X 的范围是$-2^{n-1} \leqslant X \leqslant 2^{n-1}-1$。

当 n=8 时，可表示的有符号数的范围为-128～+127。两个有符号数进行加法运算时，如果运算结果超出可表示的有符号数的范围时，就会发生溢出，使计算结果出错。

例如，(+72)+(+98)=?

将十进制数 72 和 98 转换为二进制数，运算过程如下：

```
   01001000  B    +72
+  01100010  B    +98
-------------------------
   10101010  B    -42
```

我们看到，两正数相加得数是负数。

再如，(-83)+(-80)=?

运算过程如下：

```
   10101101  B    -83
+  10110000  B    -80
-------------------------
   01011101  B    +93
```

我们看到，两负数相加得数是正数。

这两个例子的结果显然不正确，原因是发生了溢出。很显然，溢出只会出现在两个同符号数相加或两个异符号数相减的情况下。

对于加法运算，当次高位(数值部分最高位)形成进位加入最高位，而最高位(符号位)相加(包括次高位的进位)却没有进位输出时，或者反过来，当次高位没有进位加入最高位，但最高位却有进位输出时，都将发生溢出。因为这两种情况是：两个正数相加，结果超出了范围，形式上变成了负数；两负数相加，结果超出了范围，形式上变成了正数。

而对于减法运算，当次高位不需从最高位借位，但最高位却需借位(正数减负数，差超出范围)时，或者反过来，当次高位需从最高位借位，但最高位不需借位(负数减正数，差超出范围)时，也会出现溢出。

综上所述，进/借位标志位与辅助进/借位标志位的异或为 1 时将会发生溢出。

在计算机中，数据是以补码的形式存储的，而 C 语言中数据也是用补码进行存储和运算的。

5．补码的绝对值(称为真值)

若直接将 10111111 转换为十进制数，会发现结果并不是-65，而是 191。

事实上，在计算机内，如果一个二进制数最左边的位为 1，则我们可以判定它为负数，并且是用补码表示的。

若要得到一个负二进制数的绝对值(称为真值)，只要各位(包括符号位)取反，再加 1 即可。

如，二进制值 10111111(-65 的补码)，各位取反得 01000000，再加 1 得 01000001(+65 的补码)。

6．代数加减运算

1)　补码加法

$[X+Y]_{补}=[X]_{补}+[Y]_{补}$

【例 1.9】 X=+0110011，Y=-0101001，求$[X+Y]_{补}$。

解： $[X]_{补}$=00110011

$[Y]_{补}$=11010111

$[X+Y]_{补}=[X]_{补}+[Y]_{补}$= 00110011+11010111=00001010

注意： 因为计算机中运算器的位长是固定的，上述运算中产生的最高位进位将丢掉，所以结果不是 100001010，而是 00001010。

2)　补码减法

$[X-Y]_{补}=[X]_{补}-[Y]_{补}=[X]_{补}+[-Y]_{补}$

式中，$[-Y]_{补}$称为负补。求负补的方法是：所有位(包括符号位)按位取反，然后整个数加 1。

【例 1.10】 求 1+(-1)(十进制数)。

解： 1 的原码为 00000001，转换为补码为 00000001。

-1 的原码为 10000001，转换为补码为 11111111。

1+(-1)=0

00000001+11111111=00000000

00000000 转换为十进制数为 0，0=0，所以运算正确。

1.2.4　常用编码

我们输入计算机中的数据、字符、字符串等信息不能直接被计算机所识别，由于计算机只能识别 0 和 1，所以计算机中所有信息都要用二进制编码来表示。

1．8421BCD 码

用二进制编码表示十进制数的代码，简称 BCD 码(Binary Coded Decimal)，用标识符