21 世纪高职高专电类系列规划教材

# 单片机原理与应用

主　编　陈权昌　李兴富
副主编　刘高潮　兰如波　谢　洪
主　审　陈可中

华南理工大学出版社
·广州·

## 内 容 简 介

本教材主要以高职高专院校电子信息及机电类专业学生为讲授对象，力求简练，易学易懂，概念准确，语言流畅，思路清晰，内容深入浅出，理论知识与实际应用紧密结合，例题与实例典型而又实用，便于教师授课与学生自学。本书以当前流行的标准型 AT89S51 单片机为例，共分 8 章，全面、详细地介绍单片微机系统的组成、工作原理、技术特点，单片机指令系统及程序设计，系统的扩展方法、应用实例等；还介绍了 AT89S 系列各种型号单片机的封装形式和型号选用指南。使学生通过本课程的学习，不但能比较系统地掌握当今流行的 AT89S 系列单片机的开发应用技术，更重要的是培养学生有效地提高对快速发展的微机技术的思维学习能力和掌握最佳的学习方法。

**图书在版编目 (CIP) 数据**

**单片机原理与应用/陈权昌，李兴富主编．—广州：华南理工大学出版社，**2007.8
(21 世纪高职高专电类系列规划教材)
ISBN 978-7-5623-2652-6

Ⅰ.单　Ⅱ.①陈…　②李…　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 (2007) 第 129998 号**

**总 发 行：**华南理工大学出版社 (广州五山华南理工大学 17 号楼，邮编 510640)
营销部电话：020－87113487　87111048 (传真)
E-mail：scutc13@scut.edu.cn　　http：//www.scutpress.com.cn
**责任编辑：**吴兆强
**印 刷 者：**广州市穗彩彩印厂
**开　　本：**787mm×1092mm　1/16　**印张：**17.25　**字数：**442 千
**版　　次：**2007 年 8 月第 1 版　2007 年 8 月第 1 次印刷
**印　　数：**1～3 000 册
**定　　价：**28.00 元

# 前 言

以往高校教材大多均以 MCS－51 单片机为主线介绍单片机的原理及应用技术，美国 ATMEL 公司将 Flash 存储器技术与 80C51 核相结合，新推出 AT89S 系列增强型高档单片机。AT89S 系列单片机与 MCS－51 系列单片机不仅全兼容，而且增加了许多新功能，如看门狗定时器 WDT、ISP 及 SPI 串行接口技术，双数据指针 DPTR，低功耗休闲状态及关电方式，关电方式下的中断恢复技术等，片内的 Flash 可允许在线重新编程，从而使单片机在电子智能产品的开发应用过程中极为方便、极易实现。AT89S 系列单片机是目前广泛应用的主流芯片之一。

本书着重于对 AT89S51 新型单片机原理及应用进行详细的全面论述。随着单片机技术的快速发展，功能越来越强，档次日益提高，使其应用领域不断扩大。单片机的应用技术是一项新型的软、硬件紧密相结合的工程技术，为了尽快地推广单片机实用技术，本书为广大高等职业技术学院师生及有关工程技术人员在单片机硬件和软件开发应用方面提供最新、最实用、最有参考价值的资料及内容。全书共分 8 章：其中第 1 章和第 2 章由广西机电职业技术学院的陈权昌、莫胜撼和林勇坚 3 位老师共同编写；第 3 章由桂林工学院南宁分院的刘高潮老师编写；第 4 章由广西电力职业技术学院的兰如波、罗黎老师编写；第 5 章、第 6 章由桂林航天工业高等专科学校的李兴富、梁强老师编写；第 7 章由广西职业技术学院的李显圣、封宇、明鑫等老师编写；第 8 章由柳州运输职业技术学院的谢洪、张永格老师编写。

由于作者水平有限，书中可能存在某些不足之处，敬请广大读者多多指正！

编　者

2007 年 3 月

21 世纪高职高专电类系列教材

# 编写委员会

# 总 序

随着科学技术的发展，电工电子技术的应用越来越广泛，并已渗透到人们日常生活的方方面面。掌握必要的电工电子知识已成为当代大学生特别是理工类大学生必备的素质之一。而电工电子技术的教学一直存在着学时少与内容多、基本内容与新技术、理论教学与实验教学三大矛盾。如何让学生在有限的时间内学到系统而扎实的电类知识，是摆在教育工作者面前的一个重要课题。

高职高专教育是以培养应用型、工艺型人才为目标的一种教育形式，目前正处于一个全新的发展时期，对它的研究也处于探索阶段。作为高职高专教育重要的一环，其教材的编写，需要认真对待和深入研究。

高职高专教材的编写，应在保证一定的理论教学的基础上，注重培养学生的实际动手能力，为社会培养出合格的应用型人才。但是，目前我国高职高专院校之间的教学条件、教学水平、学生层次、发展模式等均不平衡，硬性规定选用统一的“规划教材”、“精品教材”显然有悖科学规律，但每个学校的教材自成体系、“自编自用”则更不现实。那么，在教材的选用和编写过程中，如何既考虑学科的前瞻性，同时又兼顾各个学校发展水平不一的现实情况，是每一位教材编写者必须首先思考的问题。在基本相似的教学背景下，联合各种优秀的教学资源，在一定的地域范围内共同研究和探讨，共同编写有一定地域特色又富有创新性的教材，则不失为一种行之有效的方法。

出于以上考虑，在华南理工大学出版社的组织和策划下，我们联合了广西、贵州两省 10 余所高职高专院校共同编写了“21 世纪高职高专电类系列规划教材”。

为了出版一套高质量的“21 世纪高职高专电类系列规划教材”，华南理工大学出版社做了大量的前期组织准备工作。他们邀请了各个参编院校中富有教学经验且负责教学管理的专家、学者担任本系列教材的编委，多次召开编委会会议，就教材内容的定位、写作的要求、参编人员的要求及组成、主编的落实、写作大纲的确定等事项进行了具体而细致的商讨。在前期准备工作

基本就绪的基础上，召开了全体参编人员出版研讨会，讨论每种教材的写作大纲和具体分工。参编人员均为从事高职高专教学工作多年的老师，他们熟知高职高专的教学现状，对未来高职高专的发展方向有深刻的认识和研究。

全体参编人员按照编委会提出的“理论适度、注重实操、切合实际”的编写原则，以高度负责的态度对待教材的出版工作。我相信，“天道酬勤”，经过华南理工大学出版社的精心策划，经过广大作者的辛勤劳动，该套教材会成为一套比较理想的、切合高职高专教学实际的教材。该套教材的出版，对推动高职高专电类专业的教学改革具有积极的意义。

高职高专教育正处于一个探索和发展的阶段，我们编写的“21 世纪高职高专电类系列规划教材”肯定还存在一些疏漏与不足，我们将依据高职高专发展的趋势，充分把握科学发展的最新动态，不断修订和完善本系列教材。同时，我们也衷心希望使用本套教材的同仁们不吝赐教，更欢迎加入到本系列教材的后续出版工作或修订再版的作者队伍中来，共同促进高职高专人才培养事业的发展。

衷心祝愿本系列教材出版成功。

广西大学教授　**陈可中**

2007 年 5 月于南宁

# 目　　录

# 第 1 章　单片微型计算机概述

单片微型计算机具有功能强、体积小、价格低廉等众多独特的优点，应用已十分广泛，是现代技术不可缺少的重要部分。本章要求掌握计算机的数学基础：数制及其相互转换、带符号数的表示方法、补码运算原理及溢出的判别方法、ASCII 码和 BCD 码的应用等。

## 1.1　计算机数学基础

### 1.1.1　数制与转换

#### 1.1.1.1　数的几种常用进制

计算机最基本的功能就是对数进行计算和处理加工，人们最熟悉的是采用十进制数，但十进制数不能直接被计算机所识别，计算机仅能认识二进制数，这是因为数在计算机的电路中是以电子元件的物理状态来表示的。电子元件通常只有两种稳定的状态，例如晶体三极管的饱和导通与截止、触发器输出的高电平与低电平、信号电路传输脉冲的有与无等。在二进制中，只有 0 和 1 两个数，若以 1 代表高电平，则 0 代表低电平，利用二值电路来进行计数、比较及运算，简便可靠，极容易实现。二进制的运算规则最简单，因此迄今为止，所有计算机都是采用二进制数来进行算术运算和逻辑运算。但缺点是在使用二进制编写程序时既繁琐又容易出错，为此，人们在编程时又经常用到十进制、十六进制和八进制。

数制是利用符号来计数的科学方法，任何一种数制都有基数和权两个要素。所谓基数，就是在某一种进制中可能用到数码的个数，例如十进制的基数为 10，二进制的基数为 2，十六进制的基数为 16 等；所谓权，就是基数的某次幂，例如基数为 2 的正次幂 $2^0$，$2^1$，$2^2$，$2^3$，…，$2^n$ 等。

1. 十进制

十进制是以 10 为基数，逢十进一，借一当十的计数体制。使用的数码有 0，1，2，3，4，5，6，7，8，9 共十个，一般来说，任意一个十进制数 $D$，都可以表示为：

$$D = D_{n-1} \times 10^{n-1} + \cdots + D_1 \times 10^1 + D_0 \times 10^0 + D_{-1} \times 10^{-1} + \cdots + D_{-m} \times 10^{-m}$$

$$= \sum_{i=n-1}^{-m} D_i \times 10^i$$

$n$ 为整数的位数，$-m$ 为小数点的位数。$D_i$ 为第 $i$ 位的系数，可取 0～9 中的任一个，$10^i$ 为第 $i$ 位的权。例如：

$$(8963.36)_{10} = 8\times10^3 + 9\times10^2 + 6\times10^1 + 3\times10^0 + 3\times10^{-1} + 6\times10^{-2}$$

2. 二进制

二进制是以 2 为基数，逢二进一，借一当二的计数体制。使用的数码仅有 0 和 1 两

个，任意一个二进制数 $B$，其加权系数表示为：

$$B = \sum_{i=n-1}^{-m} B_i \times 2^i$$

$B_i$ 为第 $i$ 位的系数，可取 0 或 1；$2^i$ 为第 $i$ 位的权。例如：

$$(1011.1)_2 = 1\times2^3 + 0\times2^2 + 1\times2^1 + 1\times2^0 + 1\times2^{-1}$$

3. 八进制

八进制是以 8 为基数，逢八进一，借一当八的计数体制。使用的数码为 0，1，2，3，4，5，6，7 共八个，任意一个八进制数 $Q$，其加权系数表示为：

$$Q = \sum_{i=n-1}^{-m} Q_i \times 8^i$$

$Q_i$ 为第 $i$ 位的系数，可取 0～7 中的任一个；$8^i$ 为第 $i$ 位的权。例如：

$$(63.35)_8 = 6\times8^1 + 3\times8^0 + 3\times8^{-1} + 5\times8^{-2}$$

4. 十六进制

十六进制是以 16 为基数，逢十六进一，借一当十六的计数体制。使用的数码为 0，1，2，3，4，5，6，7，8，9，A，B，C，D，E，F 共十六个，任意一个十六进制数 $H$，其加权系数表示为：

$$H = \sum_{i=n-1}^{-m} H_i \times 16^i$$

$H_i$ 为第 $i$ 位的系数，可取 0～F 十六个数字中的任一个；$16^i$ 为第 $i$ 位的权。例如：

$$(A83.B5)_{16} = 10\times16^2 + 8\times16^1 + 3\times16^0 + 11\times16^{-1} + 5\times16^{-2}$$

1.1.1.2 不同进制数之间的相互转换

1. 十进制数转换为二进制数

对整数部分，连续除以 2 取余反排列，直到商为 0；对小数部分，连续乘以 2 取整正排列，直到乘积的小数部分为 0 或满足误差要求为止。例如：

$(36.335)_{10} =$ $(100100.01011)_2$ 要求保留 5 位小数

```
2 | 36  …  余0   最低位
2 | 18  …  余0
2 |  9  …  余1
2 |  4  …  余0      ↑
2 |  2  …  余0
2 |  1  …  余1   最高位
     0
```

```
      0.335
×       2
-------------
      0.670  …  取整0  最高位
×       2
-------------
      1.340  …  取整 1
      0.340
×       2
-------------
      0.680  …  取整 0      ↓
×       2
-------------
      1.360  …  取整 1
      0.360
×       2
-------------
      0.720  …  取整 0  最低位
```



由于要求保留 5 位小数，0.720 大于 0.5，因此按“四舍五入”原则，最低位取 1。当十进制数转为其他进制数时，同理采用上述类似方法进行转换。

2. 二进制与八进制之间的相互转换

由于 $2^3=8$，所以 1 位的八进制数可以表示为相应的 3 位二进制数，例如：

$(13.5)_8=(001011.101)_2$

八进制转为二进制：先将每位八进制数写成对应的 3 位二进制数，然后再按原来的顺序排列即可。

二进制转为八进制：对整数部分，从最低位开始按 3 位一组分取，不足 3 位前面补 0；对小数部分，则从最高位开始按 3 位一组，不足 3 位后面补 0。然后每组以其对应的八进制数代替，排列顺序不变。例如：

$(100010.11011)_2=(42.66)_8$

3. 二进制与十六进制之间的相互转换

由于 $2^4=16$，所以 1 位的十六进制数可以表示为相应的 4 位二进制数，它们之间的相互转换方法与上述类同，只是按 4 位分组。例如：

$(38A)_{16}=(001110001010)_2$

$(1001011101010010)_2=(9752)_{16}$

注：八进制数与十六进制数之间不能直接相互变换，必须通过二进制数作中间变量进行变换。例如：

$(376)_8=(011111110)_2=(0FE)_{16}$

$(ABC)_{16}=(101010111100)_2=(5274)_8$

4. 任意进制数转为十进制数

用按权展开求和的方法进行，例如：

$$(100011.101)_2=1\times2^5+1\times2^1+1\times2^0+1\times2^{-1}+1\times2^{-3}$$
$$=32+2+1+0.5+0.125$$
$$=(35.625)_{10}$$

$$(356)_8=3\times8^2+5\times8^1+6\times8^0$$
$$=192+40+6$$
$$=(238)_{10}$$

$$(AE)_{16}=10\times16^1+14\times16^0$$
$$=160+14$$
$$=(174)_{10}$$

5. 数制的表示符号

对于不同数制的数据，为在书写上能够加以区别，常在数后用字母符号来表示不同的数制，例如：

十进制：133D 或 133；

二进制：10000011B；

十六进制：3AH，87DCH，0DEH；（注：在汇编语言中首位为字符时，前面加 0）

八进制：365Q。

### 1.1.2 带符号数的表示方法

#### 1.1.2.1 机器数与真值

上述已提到计算机仅能对二进制数进行运算和处理，而二进制数包括有无符号数据和带符号数据两种，对于带符号的二进制数据，在计算机中正、负符号是如何表示呢？通常将一个二进制数的最高位用作为符号位，规定符号位用“0”表示正，用“1”表示负。这样，一个二进制数，连同符号位在内作为一个数据，称作机器数，是计算机所能识别的数；而用“+”、“-”号分别表示正数和负数，这样的数称为真值。以八位机为例：

+68 的机器数为 01000100B；

-68 的机器数为 11000100B。

↓ ↓

符号位 数值位

上面两个机器数的真值分别为 +1000100B 和 -1000100B。

在计算机中常用的机器数有原码、反码、补码三种形式，分别介绍如下。

1. 原码

对正数的符号位用 0 表示，负数的符号位用 1 表示时，这种表示法称为原码。如：

$[+68]_{原}=01000100B$

$[-68]_{原}=11000100B$

注意：8 位二进制原码能表示数的范围为：

$[+127]_{原}=01111111B$～ $[-127]_{原}=11111111B$

在原码中，0 有两种表示法，即：

$[+0]_{原}=00000000B$

$[-0]_{原}=10000000B$

原码表示法简单易懂，且与真值转换方便。但是，当两个异号数相加或两个同号数相减时，就要进行减法运算。而在计算机中的微处理器一般只有加法器，无减法器，所以，为了把减法运算变为加法运算，就引入了反码和补码。

2. 反码

⑴ 正数的反码表示法与正数的原码相等，最高位为符号位，其余则为数值位。例如：

$[+13]_{原}=00001101B$

$[+13]_{反}=00001101B$

↓ ↓

符号位 数值位



⑵ 负数的反码表示法，先保持其原码的符号位不变，然后数值则逐位取反而形成。例如：

$[-13]_{原}=10001101B$

$[-13]_{反}=11110010B$

反码所能表示的数值范围，对于八位机来说，只能在 -127～+127；对 0 也有两种表

示法，即：

$[+0]_{反}=00000000B$

$[-0]_{反}=11111111B$

3. 补码

正数的补码与正数原码相等；负数的补码为反码末位加 1 而形成。例如：

$[+13]_{补}=[+13]_{原}=00001101B$

$[-13]_{补}=[-13]_{反}+1=11110010+1=11110011B$

补码的数值位按位取反后末位加 1，也可转换为原码。例如：

已知 $[-13]_{补}=11110011B$，求 $[-13]_{原}$。

解：将补码除符号外逐一取反后得：10001100B，则：

$[-13]_{原}=10001100+1=10001101B$

上面已表述，在计算机中，对于带符号的数，采用补码表示法，目的是使减法转换为加法运算，从而使正、负数的加减运算全变为单纯的相加运算。

注：8 位二进制补码能表示数的范围为：

－128（10000000B）～＋127（01111111B）

在补码中，＋0 与－0 的表示法相同，即：

$[+0]_{补}=[-0]_{补}=00000000B$

1.1.2.2 进位与溢出

进位是指数值最高位的进位与借位，溢出是指超出计算机所能表示数的范围。由于采用了补码表示法，计算机在处理带符号数与不带符号数时同样对待，按原先的定义进行处理。一个二进制数究竟是带符号数还是不带符号数，完全由进行运算的人来事先确定，计算机是不知道的，计算机只是按照所编的程序执行运算。

当确定计算机不带符号数进行加减运算时，运算结果仍为无符号数，进位与溢出相一致，加法时若有进位，表示“和”超出了数表示的范围，产生了溢出；减法时若有借位，表示不够减，“差”为负数，超出了数表示的范围，也就产生溢出。现以八位机为例，运算结果的存储器能装入的数值范围为 0（00000000B）～255（11111111B），若超出，则有进位或借位，同时也产生溢出。例如：

127＋129＝256（＞255）

计算机采用二进制加法运算规则，从最低位开始逐位相加：

$$\begin{array}{r} 127=01111111B \\ +\ 129=10000001B \\ \hline 100000000B \end{array}$$

进位→

溢出

由于运算结果超出 8 位二进制数，有进位，也产生溢出（256＞255）。但用进位和结果数一起表示的数值仍可取得正确的“和”。

在减法运算时，若运算结果有借位，也必产生溢出，这时“差”出错。例如：

128－129＝－1（＜0）

计算机采用二进制减法运算规则，从最低位开始逐位相减：

$128 = 10000000B$

$-129 = 10000001B$

借位→ 111111111B

溢出

以上运算结果同样超出 8 位二进制数，有借位，也产生溢出 （$-1<0$），但是无法表示正确的“差”，结果出错。

当确定计算机对带符号数进行加减运算时，均采用补码形式进行，其结果也仍为补码，但进位与溢出并不一致。在上述的内容中我们已经知道了八位机补码能表示数的范围为 $-128$ （10000000B） ～ $+127$ （01111111B），只要运算结果不超过规定的数表示的范围，不产生溢出，则结果正确，否则结果出错。补码加、减法运算规则如下：

加法：$[X+Y]_{补}=[X]_{补}+[Y]_{补}$

减法：$[X-Y]_{补}=[X+(-Y)]_{补}=[X]_{补}+[-Y]_{补}$

例如：

1. 两个正数相加

(1) $36+64=100$

计算机采用补码运算：

$[36]_{补}=00100100B$

$+\ [64]_{补}=01000000B$

$01100100B$

两个正数相加，运算结果仍为正数，没有进位，也不产生溢出 （$100<+127$），表明结果正确。

(2) $115+32=147$

$[115]_{补}=01110011B$

$+\ [32]_{补}=00100000B$

$10010011B$

由于 $147>127$，计算机采用补码运算时，必然产生溢出，使符号位受破坏。即两个正数相加，运算结果变为负数，表明结果不正确。

2. 两个负数相加

(1) $(-36)+(-64)=-100$

$[-36]_{原}=10100100B$；　　$[-64]_{原}=11000000B$；

$[-36]_{反}=11011011B$；　　$[-64]_{反}=10111111B$；

$[-36]_{补}=11011100B$；　　$[-64]_{补}=11000000B$；

$[-36]_{补}=11011100B$

$+\ [-64]_{补}=11000000B$

进位→ 110011100B

丢掉

因为运算结果仍为补码形式，若转为原码形式，则将补码除符号外逐一取反后得：11100011B，最终得：$11100011+1=11100100B=[-100]_{原}$；

由于两个负数相加得负数，而且运算结果 $-100$ 大于 $-128$，虽有进位，但符号位正确，没产生溢出，结果正确。

(2) $(-93)+(-80)=-173(<-128)$

$[-93]_{原}=11011101B$；　　$[-80]_{原}=11010000B$；

$[-93]_{反}=10100010B$；　　$[-80]_{反}=10101111B$；

$[-93]_{补}=10100011B$；　　$[-80]_{补}=10110000B$；

$$\begin{array}{l}[-93]_{补}=10100011B\\ +\ [-80]_{补}=10110000B\\ \hline \text{进位}\rightarrow\ \ 101010011B\end{array}$$

由于 $-173<-128$，计算机采用补码运算时，也必然产生溢出，使符号位受破坏。即两个负数相加得正数，虽有进位，但结果不正确。

3. 异号数相减

例：$125-(-10)=125+10=135$

$[125]_{补}=[125]_{原}=01111101B$；　　$[10]_{补}=[10]_{原}=00001010B$

$$\begin{array}{l}[125]_{补}=01111101B\\ +\ [10]_{补}=00001010B\\ \hline 10000111B\end{array}$$

由于两个正数相加得负数，运算结果虽不产生进位，但符号位不正确有溢出，所以结果不正确。

1.1.2.3　溢出的判断

在上述几个例中，当两个用补码表示的数作加减运算时，如果是同号相加或异号相减，则运算结果可能会出现溢出，这时可以把运算结果数值的符号和参与运算的数值符号相比较，若出现负数加负数得正数，正数加正数得负数的情况，则可以认定运算结果有溢出，出错；如果是同号相减或异号相加，则可以推理，只能会使数值的绝对值越来越小，运算结果不可能产生溢出。

计算机是如何判断进位及溢出的状况呢？对于单片机，是采用程序状态寄存器 PSW 的进位（借位）标志 CY（CY = 1，表示有进位，CY = 0，无进位）和溢出标志位 OV（OV = 1，表示有溢出，否则无）的状态来进行判断的。

PSW

| CY | AC | F0 | RS1 | RS0 | OV | – | P |
|---|---|---|---|---|---|---|---|

对于 8 位二进制数据，一般称最低位为第 0 位，最高位为第 7 位，如下所示：

| 位 7 | 位 6 | → | 位 0 |
|---|---|---|---|

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|

当进行加法运算时，若在位 7、位 6 中均产生进位或均无进位，则 OV = 0，表示运算结果正确；若位 7、位 6 中仅有一位产生进位，而另一位无进位，则 OV = 1，表示两个正数相加，和为负数；或两个负数相加，和为正数的错误结果。

当进行减法运算时，若在位 7、位 6 中均产生借位或均无借位，则 OV = 0，表示运算