



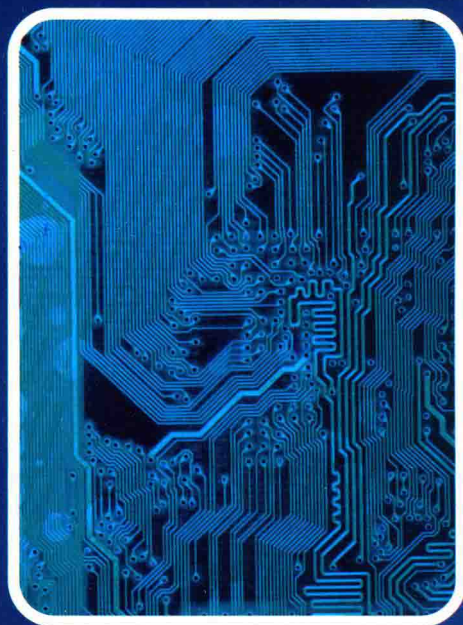
全国普通高等学校机械类“十三五”规划系列教材  
 全国普通高等学校机械类“十二五”规划系列教材

丛书顾问 ▶ 李培根 林萍华

# 单片机原理 及接口技术

(第二版)

杨术明 ▶ 主编



DANPIANJI YUANLI JI  
JIEKOU JISHU



JIXILEI SHISANWU



华中科技大学出版社

<http://www.hustp.com>

全国普通高等学校机械类“十三五”规划系列教材  
全国普通高等学校机械类“十二五”规划系列教材

# 单片机原理及接口技术 (第二版)

主 编 杨术明  
副主编 王 军 王艳春 周华茂  
        祝爱萍 李茂强  
参 编 王欣欣 杨小玲 翟晓华  
        薛 森



华中科技大学出版社  
中国·武汉

## 内 容 提 要

本书以 MCS-51 系列单片机为核心,系统介绍单片机的原理及应用,主要内容包括:单片机概述;单片机的硬件结构及原理;指令系统与汇编语言程序设计;定时器/计数器;中断系统;串行接口;MCS-51 系列单片机的系统扩展与接口技术;Keil C51 的应用程序设计基础;Proteus 虚拟仿真设计。

本书既立足于读者对单片机理论知识的掌握,也着眼于应用能力的提高。为此,引入了软件开发平台 Keil C51 软件和硬件开发平台 Proteus 软件,通过开发实例,引导读者掌握单片机系统的硬件设计、软件设计及仿真分析流程,提高读者对这门课程的学习兴趣。

本书是高等学校机械类及相关专业的教材,同时也可供计算机专业、高等职业教育相关专业作为教材及从事单片机开发应用方面的工程技术人员学习参考。

### 图书在版编目(CIP)数据

单片机原理及接口技术/杨术明主编. —2 版. —武汉:华中科技大学出版社,2018.7  
全国普通高等学校机械类“十三五”规划系列教材  
ISBN 978-7-5680-4355-7

I. ①单… II. ①杨… III. ①单片微型计算机-基础理论-高等学校-教材 ②单片微型计算机-接口技术-高等学校-教材 IV. ①TP368.1

中国版本图书馆 CIP 数据核字(2018)第 152861 号

单片机原理及接口技术(第二版)

Dānpiānjī Yuanli jì Jiekou Jìshù(Dì-er Bān)

杨术明 主编

策划编辑:余伯仲

责任编辑:刘 飞

封面设计:原色设计

责任监印:周治超

出版发行:华中科技大学出版社(中国·武汉)

电话:(027)81321913

武汉市东湖新技术开发区华工科技园

邮编:430223

录 排:武汉市洪山区佳年华文印部

印 刷:武汉市籍缘印刷厂

开 本:787mm×1092mm 1/16

印 张:15.5

字 数:407 千字

版 次:2018 年 7 月第 2 版第 1 次印刷

定 价:34.80 元



华中科技大学

本书若有印装质量问题,请向出版社营销中心调换  
全国免费服务热线:400-6679-118 竭诚为您服务  
版权所有 侵权必究

## 第二版前言

单片机技术的发展日新月异,内部资源不断扩展,功能日益增多,性能不断提高,应用范围越来越广。特别是在工业测量与控制、智能仪表、家用电器以及学生创新等方面,单片机已成为不可或缺的核心部件。

本书的特色是借助单片机集成开发环境 Keil 和 Proteus,将硬件设计、软件设计及仿真分析有机统一,为学习单片机原理,提高单片机开发效率提供系统的指导方法。本书自 2013 年出版以来,已被多所高校作为“单片机原理及接口技术”这门课程的指定教材,且已多次重印。根据读者的反馈意见,特对本书存在的问题进行了更正并对相关程序进行了修订。

在本书的撰写和修订过程中,参考了相关的国内外文献和教材,在此谨向作者表示衷心的感谢。

编 者

2018 年 5 月

# 第一版前言

本书是为普通高等学校培养基础扎实、知识面宽、具有创新实践能力的新世纪应用型人才而编写的,是全国普通高等学校机械类“十二五”规划系列教材。

单片机因集成度高、体积小、功耗低、功能强及性价比高等特点,被广泛应用于智能仪表、家用电器、通信设备、工业测控、汽车电子产品等领域。单片机的应用已经渗透到我们生活的方方面面,单片机应用技术已成为工程技术人员必须掌握的技术之一。

本书既立足于读者对单片机理论知识的掌握,又着眼于应用能力的提高。在介绍单片机硬件结构、单片机指令等理论知识的基础上,引入了软件开发平台 Keil C51 软件和硬件开发平台 Proteus 软件,通过开发实例,引导读者掌握单片机系统的硬件设计、软件设计及仿真分析方法,体现出理论和实践的有机统一,是提高单片机教学质量的一种尝试。

本书共分 9 章,主要内容为:单片机概述、单片机的硬件结构及原理、指令系统与汇编语言程序设计、定时器/计数器、中断系统、串行接口、MCS-51 系列单片机的系统扩展与接口技术、Keil C51 的应用程序设计基础和 Proteus 虚拟仿真设计。

本书由杨术明(宁夏大学)担任主编,王军(江西理工大学)、王艳春(蚌埠学院)、周华茂(江西农业大学)、祝爱萍(宁夏大学)担任副主编,主要编写成员还有翟晓华(晋中学院)、王欣欣(华北水利水电学院)、杨小玲(江西农业大学)。杨术明负责全书的统稿。在本书的编写过程中,参考了其他版本的同类教材及专家学者的文献资料,在此特向其编著者表示衷心的感谢!

由于编者水平有限,书中难免存在疏漏与不妥之处,敬请广大读者批评指正。

编 者

2012 年 5 月

# 目 录

<b>第 1 章 单片机概述</b> .....	(1)
1.1 数制及其运算 .....	(1)
1.2 BCD 码和 ASCII 码 .....	(6)
1.3 单片机的产生与发展 .....	(7)
1.4 MCS-51 系列单片机介绍 .....	(10)
1.5 基于 MCS-51 内核单片机简介 .....	(11)
1.6 单片机应用系统开发概述 .....	(13)
习题 .....	(16)
<b>第 2 章 单片机的硬件结构及原理</b> .....	(17)
2.1 MCS-51 系列单片机的内部结构 .....	(17)
2.2 MCS-51 系列单片机的引脚功能 .....	(18)
2.3 中央处理器 .....	(20)
2.4 存储器结构 .....	(22)
2.5 单片机的并行输入/输出口 .....	(26)
2.6 单片机的时钟与时序 .....	(29)
2.7 单片机的复位 .....	(31)
2.8 MCS-51 系列单片机的最小系统 .....	(32)
习题 .....	(33)
<b>第 3 章 指令系统与汇编语言程序设计</b> .....	(34)
3.1 Keil C51 开发工具简介 .....	(34)
3.2 MCS-51 系列单片机指令系统 .....	(37)
3.3 汇编语言程序设计 .....	(55)
习题 .....	(70)
<b>第 4 章 定时器/计数器</b> .....	(72)
4.1 定时器/计数器的结构及工作原理 .....	(72)
4.2 定时器/计数器工作方式和控制寄存器 .....	(73)
4.3 定时器/计数器工作方式 .....	(74)
4.4 定时器/计数器编程举例 .....	(78)
习题 .....	(85)
<b>第 5 章 中断系统</b> .....	(86)
5.1 中断概述 .....	(86)
5.2 MCS-51 系列单片机中断系统 .....	(86)
5.3 中断系统编程举例 .....	(95)

---

习题 .....	(99)
<b>第 6 章 串行接口</b> .....	(100)
6.1 串行通信的基本概念 .....	(100)
6.2 MCS-51 系列单片机的串行接口 .....	(101)
6.3 多机通信 .....	(107)
6.4 串行接口编程举例 .....	(108)
习题 .....	(110)
<b>第 7 章 MCS-51 系列单片机的系统扩展与接口技术</b> .....	(111)
7.1 单片机系统扩展概述 .....	(111)
7.2 MCS-51 系列单片机存储器的扩展技术 .....	(117)
7.3 输入/输出接口的扩展技术 .....	(125)
7.4 管理功能部件的扩展技术 .....	(133)
7.5 A/D 及 D/A 转换器的接口技术 .....	(161)
习题 .....	(173)
<b>第 8 章 Keil C51 的应用程序设计基础</b> .....	(175)
8.1 Keil C51 程序设计的基本语法 .....	(175)
8.2 Keil C51 程序的基本语句 .....	(182)
8.3 Keil C51 的函数 .....	(185)
8.4 Keil C51 的编译预处理 .....	(188)
8.5 Keil C51 编译器的绝对地址访问 .....	(190)
习题 .....	(191)
<b>第 9 章 Proteus 虚拟仿真设计</b> .....	(193)
9.1 Proteus 简介 .....	(193)
9.2 Proteus Schematic Capture 原理图设计与仿真 .....	(198)
9.3 Proteus 应用实例 .....	(205)
<b>附录 A Keil <math>\mu</math>Vision4 菜单及功能说明</b> .....	(227)
<b>附录 B MCS-51 系列单片机指令表</b> .....	(231)
<b>附录 C Keil C51 的库函数</b> .....	(236)
<b>参考文献</b> .....	(242)

# 第 1 章 单片机概述

## 1.1 数制及其运算

数制是人们对事物数量计数的一种统计规律。在日常生活中最常用的是十进制,但在计算机中,由于电子元件最易实现的是两种稳定状态:器件的“开”与“关”,电平的“高”与“低”,因此,采用二进制数“0”和“1”可以很方便地表示机内的数据运算与存储。在编程时,为了方便阅读和书写,人们还经常用八进制数或十六进制数来表示二进制数。虽然一个数可以用不同计数制形式表示它的大小,但该数的量值是相等的。

### 1.1.1 计数机中的数制及其相互转换

#### 1. 数制的基数和位权

当进位计数制采用位置表示法时,同一数字在不同的数位所代表的数值是不同的。每一种进位计数应包含以下两个基本的因素。

(1) 基数  $R$  (radix) 它代表计数制中所用到的数码个数,简称基。如:二进制计数中用到“0”和“1”两个数,而八进制计数中用到 0~7 共八个数。一般来说,基数为  $R$  的计数制(简称  $R$  进制)中,包含 0, 1, ...,  $R-1$  个数码,进位规律为“逢  $R$  进 1”。

(2) 位权  $W$  (weight) 在进位计数制中,某个数位的值是由这一位的数码值乘以处在这一位的固定常数决定的,通常把这一固定常数称为位权值,简称位权或权。各位的位权是以  $R$  为底的幂。如十进制的基数  $R=10$ ,则个位、十位、百位上的位权分别为  $10^0$ 、 $10^1$ 、 $10^2$ 。

**例 1-1** 二进制数 1010.11 的基数  $R$  为 2。

对应各位的位权  $W$  分别为  $2^3$ 、 $2^2$ 、 $2^1$ 、 $2^0$ 、 $2^{-1}$  和  $2^{-2}$ 。

#### 2. 常用数制简介

(1) 二进制数(binary) 二进制计数中用到“0”和“1”共两个数,后缀用“B”表示,进位规律为“逢 2 进 1”。

(2) 八进制数(octal) 八进制计数中用到 0~7 共八个数,因为字母“O”与数字“0”易混淆,所以后缀用“Q”表示,进位规律为“逢 8 进 1”。

(3) 十进制数(decimal) 十进制计数中用到 0~9 共十个数码,后缀用“D”表示,后缀“D”在使用时可以省略,此时数字默认为十进制数,规律为“逢 10 进 1”。

(4) 十六进制数 十六进制计数中用到 0~9 十个数码和 A、B、C、D、E、F 六个字母,其中 A 到 F 六个字母分别代表十进制数中的 10 到 15,后缀用“H”表示,进位规律为“逢 16 进 1”。

#### 3. 各种进制数转换为十进制数

各种进制数转换为十进制数的原则为:按位权展开相加。

**例 1-2** 将数 FFFFH, 735.4Q 及 10111100.101B 分别转换为十进制数。

$$\text{FFFFH} = 15 \times 16^3 + 15 \times 16^2 + 15 \times 16^1 + 15 \times 16^0 = 65535$$

$$735.4\text{Q} = 7 \times 8^2 + 3 \times 8^1 + 5 \times 8^0 + 4 \times 8^{-1} = 477.5$$



$$10111100.101B = 2^7 + 2^5 + 2^4 + 2^3 + 2^2 + 2^{-1} + 2^{-3} = 188.625$$

#### 4. 十进制数转换为二、八、十六进制数

十进制数转换为二、八、十六进制数时,需将整数部分和小数部分分开进行,转换原则为:整数部分除基取余,小数部分乘基取整。

整数部分转换步骤分为以下三步。

**步骤 1** 用基去除整数部分,得到商和余数,记余数为最终进制整数的最低位数码。

**步骤 2** 再用基继续去除上面得到的商,求出新的商和余数,余数又作最终进制整数的次低位数码。

**步骤 3** 重复步骤 2,直至商为零为止,整数转换结束。此时,余数作为转换后最终进制整数的最高位数码。

小数部分转换步骤分为以下三步。

**步骤 1** 用基去乘小数部分,记下乘积的整数部分,作为最终进制数小数的第 1 个数码。

**步骤 2** 再用基继续去乘上次得到的积的纯小数部分,得到新乘积的整数部分,记为最终进制数小数的次位数码。

**步骤 3** 重复步骤 2,直至乘积的小数部分为零,或者达到所需要的精度位数为止。此时,乘积的整数位作为最终进制数小数位的后一个数码。

**例 1-3** 将数 254.73D 转换为十六进制数,保留两位小数。

$\begin{array}{r} 16 \overline{) 254} \\ \underline{16 \quad 15} \\ 0 \end{array}$	余数 E F	低位 ↓ 高位	$\begin{array}{r} 0.73 \\ \times 16 \\ \hline 438 \\ 73 \\ \hline 11.68 \\ \times 16 \\ \hline 408 \\ 68 \\ \hline 10.88 \end{array}$	..... 小数部分为B ..... 小数部分为A	高位 ↓ 低位
--	--------------	---------------	---	------------------------------	---------------

即  $254.73D = FE.BAH$

由以上两个例子可以看出:其他进制数可以精确转换为十进制数,但十进制数不一定可以精确转换为其他进制数。

### 1.1.2 二进制数的运算

#### 1. 二进制数的算术运算

二进制数只有“0”和“1”两个数,其算术运算比较简单,加、减法分别遵循“逢二进一”和“借一当二”的原则。

##### 1) 二进制数的加法运算

一位二进制数的加法运算规则为:0+0=0,0+1=1,1+0=1,1+1=10(有进位)。

**例 1-4** 求 10101100B+110110B 的和。

解 被加数 10101100  
 加数 + 110110  
 和 11100010

即  $10101100B + 110110B = 11100010B$

##### 2) 二进制数的减法运算

一位二进制数的减法运算规则为:0-0=0,1-1=0,1-0=1,0-1=1(有借位)。

**例 1-5** 求  $11000111\text{B}-110110\text{B}$  的差。

$$\begin{array}{r} \text{解 被减数} \quad 11000111 \\ \text{减数} \quad \quad - \quad 110110 \\ \hline \text{差} \quad \quad \quad 10010001 \end{array}$$

即  $11000111\text{B}-110110\text{B}=10010001\text{B}$

3) 二进制数的乘法运算

一位二进制数的乘法运算规则为： $0 \times 0=0, 0 \times 1=0, 1 \times 0=0, 1 \times 1=1$ 。

**例 1-6** 求  $11001101\text{B} \times 1101\text{B}$  的积。

$$\begin{array}{r} \text{解 被乘数} \quad 11001101 \\ \text{乘数} \quad \quad \times \quad 1101 \\ \hline \quad \quad \quad 11001101 \\ \quad \quad 00000000 \\ \quad 11001101 \\ 11001101 \\ \hline \text{积} \quad 101001101001 \end{array}$$

即  $11001101\text{B} \times 1101\text{B}=101001101001\text{B}$

4) 二进制数的除法运算

**例 1-7** 求  $11101101\text{B} \div 1011\text{B}$  的商。

$$\begin{array}{r} \text{解} \quad \quad \quad 10101 \\ 1011 \overline{) 11101101} \\ \quad \underline{1011} \phantom{00} \\ \quad \quad 1111 \phantom{00} \\ \quad \quad \underline{1011} \phantom{00} \\ \quad \quad \quad 10001 \\ \quad \quad \quad \underline{1011} \\ \quad \quad \quad \quad 110 \end{array}$$

即  $11101101\text{B} \div 1011\text{B}=10101\text{B}$ , 余数为  $110\text{B}$

2. 二进制数的逻辑运算

1) 二进制数“与”运算

“与”运算又称逻辑乘, 运算符为“ $\cdot$ ”或“ $\wedge$ ”, 实现“有 0 就为 0, 全 1 才为 1”的逻辑运算。

“与”运算的规则为： $0 \wedge 0=0, 0 \wedge 1=0, 1 \wedge 0=0, 1 \wedge 1=1$ 。

**例 1-8** 若  $A=11010011\text{B}, B=10101010\text{B}$ , 求  $A \wedge B$ 。

解

$$\begin{array}{r} \quad \quad \quad 11010011 \\ \wedge \quad 10101010 \\ \hline \quad \quad \quad 10000010 \end{array}$$

即  $A \wedge B=11010011\text{B} \wedge 10101010\text{B}=10000010\text{B}$

2) 二进制数“或”运算

“或”运算又称逻辑加, 运算符为“ $+$ ”或“ $\vee$ ”, 实现“有 1 就为 1, 全 0 才为 0”的逻辑运算。

“或”运算的规则为： $0 \vee 0=0, 0 \vee 1=1, 1 \vee 0=1, 1 \vee 1=1$ 。

例 1-9 若  $A=10111101B, B=11010001B$ , 求  $A \vee B$ 。

解

$$\begin{array}{r} 10111101 \\ \vee 11010001 \\ \hline 11111101 \end{array}$$

即  $A \vee B = 10111101B \vee 11010001B = 11111101B$

3) 二进制数“非”运算

“非”运算又称逻辑非, 变量  $X$  的“非”运算记作  $\bar{X}$ , 实现对变量  $X$  取反的逻辑运算。“非”运算的规则为  $\bar{0}=1, \bar{1}=0$ 。

例 1-10 若  $X=11100001B$ , 求  $\bar{X}$ 。

解  $\bar{X} = \overline{11100001} = 00011110B$ 。

4) 二进制数“异或”运算

“异或”运算符为“ $\oplus$ ”, 实现“相同为 0, 相异为 1”的逻辑运算。“异或”运算的规则为  $0 \oplus 0 = 0, 0 \oplus 1 = 1, 1 \oplus 0 = 1, 1 \oplus 1 = 0$ 。

例 1-11 若  $A=10001111B, B=11010001B$ , 求  $A \oplus B$ 。

解

$$\begin{array}{r} 10001111 \\ \oplus 11010001 \\ \hline 01011110 \end{array}$$

即  $A \oplus B = 10001111B \oplus 11010001B = 01011110B$

### 1.1.3 符号数的表示

#### 1. 机器数与真值

计算机在数的运算中, 不可避免地会遇到正数和负数, 那么在计算机中正负号如何表示呢? 由于计算机只能识别“0”和“1”, 因此, 我们将一个二进制数的最高位用作符号位来表示这个数的正负。规定符号位用“0”表示正, 用“1”表示负。例如,  $X = -110101B, Y = +110101B$ , 则在计算机中  $X, Y$  用八位二进制数可分别表示如下。

	D7	D6	D5	D4	D3	D2	D1	D0
$X = -110101B$	1	0	1	1	0	1	0	1
	符号		数值部分					

	D7	D6	D5	D4	D3	D2	D1	D0
$Y = +110101B$	0	0	1	1	0	1	0	1
	符号		数值部分					

把一个二进制数连同符号位在内作为一个数, 这个数称为机器数, 如  $10110101B$ , 而一般书写形式的数, 即原来二进制数的数值称为该机器数的真值, 如  $-110101B$ 。计算机中机器数的表示方法有三种, 即原码、反码和补码。

#### 2. 数的码制

##### 1) 原码

正数的符号用“0”表示, 负数的符号用“1”表示, 数值部分用真值的绝对值来表示的二进制机器数称为数的原码, 用  $[X]_{原}$  表示, 正数的原码与其真值相同。

**例 1-12** 对于 8 位数据,

若  $X = +35$ , 则  $[X]_{\text{原}} = [+35]_{\text{原}} = 00100011\text{B}$

若  $X = -35$ , 则  $[X]_{\text{原}} = [-35]_{\text{原}} = 10100011\text{B}$

若  $X = 0$ , 则原码有两种表示法:  $[+0]_{\text{原}} = 00000000\text{B}$ ,  $[-0]_{\text{原}} = 10000000\text{B}$

由此可见, 当  $n$  为二进制的位数时, 原码与真值的关系为

$$[X]_{\text{原}} = \begin{cases} X, & 0 \leq X < 2^{n-1} \\ 2^{n-1} - X, & -2^{n-1} < X \leq 0 \end{cases}$$

八位二进制数( $n=8$ )原码所能表示的数的范围为  $-127 \sim 127$ 。

### 2) 反码

一个正数的反码等于该数的原码; 一个负数的反码, 等于该负数的原码符号位不变(即“1”), 数值位按位求反(即“0”变“1”, “1”变“0”)。反码用  $[X]_{\text{反}}$  表示。

**例 1-13** 对于 8 位数据,

若  $X = +35$ , 则  $[X]_{\text{反}} = [+35]_{\text{反}} = 00100011\text{B}$

若  $X = -35$ , 则  $[X]_{\text{反}} = [-35]_{\text{反}} = 11011100\text{B}$

若  $X = 0$ , 则反码也有两种表示法:  $[+0]_{\text{反}} = 00000000\text{B}$ ,  $[-0]_{\text{反}} = 11111111\text{B}$

由此可见, 当  $n$  为二进制的位数时, 反码与真值的关系为

$$[X]_{\text{反}} = \begin{cases} X, & 0 \leq X < 2^{n-1} \\ 2^{n-1} - 1 + X, & -2^{n-1} < X \leq 0 \end{cases}$$

八位二进制数( $n=8$ )反码所能表示的数的范围为  $-127 \sim 127$ 。

### 3) 补码

在日常生活中有许多“补”码的事例。如钟表, 假设标准时间为 5 点整, 而某钟表却指在 9 点, 若要把表拨准, 可以有两种拨法, 一种是倒拨 4 小时, 即  $9 - 4 = 5$ ; 另一种是顺拨 8 小时, 即  $9 + 8 = 5$ 。尽管将表针倒拨或顺拨不同的时数, 但却得到相同的结果, 即“ $9 - 4$ ”与“ $9 + 8$ ”的效果是一样的。这是因为钟表采用 12 进位, 超过 12 就从头算起, 即  $9 + 8 = 12 + 5$ , 12 称为钟表计时制的模(mod)。模(mod)为一个系统的量程或此系统所能表示的最大数, 它会自然丢掉, 如:  $9 - 4 = 5$  与  $9 + 8 = 12 + 5 \pmod{12}$  自然丢掉)是等价的。

通常称“ $-4$ ”和“ $+8$ ”是在模为 12 时的补数。于是, 引入补码后, 可使减法运算变为加法运算。

一般情况下, 任一整数  $X$ , 对于  $n$  位计算机, 在模为  $K$  时的补码可表示为

$$[X]_{\text{补}} = \begin{cases} X, & 0 \leq X < 2^{n-1} \\ 2^n + X, & -2^{n-1} \leq X \leq 0 \end{cases}$$

**例 1-14** 对于 8 位数据,

若  $X = +35$ , 则  $[X]_{\text{补}} = [+35]_{\text{补}} = 00100011\text{B}$

若  $X = -35$ , 则  $[X]_{\text{补}} = [-35]_{\text{补}} = 11011101\text{B}$

若  $X = 0$ , 则补码只有一种表示法, 即  $[+0]_{\text{补}} = [-0]_{\text{补}} = 00000000\text{B}$

八位二进制数( $n=8$ )补码所能表示的数的范围为  $-128 \sim 127$ 。

综上所述, 计算机中数的码制可归纳如下。

- (1) 正数的原码、反码、补码就是该数本身。
- (2) 负数的原码其符号位为 1, 数值位不变。
- (3) 负数的反码其符号位为 1, 数值位逐位求反。

(4) 负数的补码其符号位为 1,数值位逐位求反并在末位加 1。

注意:计算机中所有的符号数默认用补码表示;计算机中所能表示的符号数的范围为  $-2^{n-1} \sim +2^{n-1}-1$ ,  $n$  为机器数的位数;已知一个数的补码时,  $[\text{正数}]_{\text{真值}} = [\text{正数}]_{\text{补}}$ ,  $[\text{负数}]_{\text{真值}} = [\text{负数}]_{\text{补}}$  取反(符号位除外)+1。

## 1.2 BCD 码和 ASCII 码

### 1.2.1 BCD 码

二进制数以其物理易实现和数据传送、运算简单的优点,在计算机中得到了广泛应用,但二进制数不直观,也不符合人们的日常习惯,所以在计算机的输入和输出时,通常还是采用十进制数表示。为了既满足人们的习惯,又能让计算机接受,便引入了 BCD 码(binary coded decimal)。它用二进制编码来表示二进制数,这样的十进制数的二进制编码,既具有二进制数的形式,又具有十进制数的特点,便于传递处理。

一位十进制数有 0~9 共十个数,需要由四位二进制数来表示。四位二进制数有 16 种组合,取其前 10 种组合分别代表十个十进制数,最常用的方法是 8421 BCD 码,其中 8,4,2,1 分别为四位二进制数的位权值。表 1-1 给出了十进制数和 8421 BCD 码的对应关系。

**例 1-15** 写出 129.36 的 BCD 码。

**解** 根据表 1-1,可直接写出相应的 BCD 码,即

$$129.36 = (000100101001.00110110)_{\text{BCD}}$$

表 1-1 8421 BCD 编码表

十进制数	8421 BCD 码	十进制数	8421 BCD 码
0	0000	5	0101
1	0001	6	0110
2	0010	7	0111
3	0011	8	1000
4	0100	9	1001

### 1.2.2 ASCII 码

计算机除了能对二进制数运算外,还需要对各种各样的字符进行识别和处理,这就要求计算机首先能够表示这些字符。这些字符如下。

数字字符:0,1,⋯,9。

大小写的 26 个英文字母:A,B,⋯,Z;a,b,⋯,z。

专用字符:如+,−,\*,/,SP(空格)等。

各种标点符号。

非打印字符:CR(回车),LF(换行),BEL(响铃)等。

当用汇编语言或其他高级语音编写的源程序送入计算机时,就要键入很多字母、数字、标点及其他符号,这就要对字符进行编码。

在微型计算机中,目前国际上比较通用的是美国标准学会(ANSI)在1963年制定的美国国家信息交换标准字符码,简称ASCII码。标准的ASCII码采用七位二进制数对字符进行编码,共有128个元素,其中包括32个通用控制字符,10个十进制数码,52个大小写英文字母和34个专用符号。例如,字母“A”的ASCII码为41H,字母“b”的ASCII码为62H,0~9的ASCII码是30H~39H,详见表1-2。

表1-2 ASCII码表

高位 低位	000	001	010	011	100	101	110	111
0000	NULL	DLE	SP	0	@	P	,	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	\$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	'	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	
1101	CR	GS	-	=	M	]	m	}
1110	SO	RS	.	>	N	↑	n	~
1111	SI	US	/	?	O	←	o	DEL

在计算机中传输ASCII码时,通常采用八位二进制数,因此,最高有效位用做奇偶校验位,用于检查代码在传输过程中是否出现差错。

如果字母“W”的ASCII码采用偶校验,则在最左边的奇偶校验位上加一个“1”,即为11010111B,使其变成偶数个“1”。

如果“W”采用奇校验,则在最左边加一个“0”,即为01010111B,变成奇数个“1”。

### 1.3 单片机的产生与发展

随着大规模集成电路的出现及其发展,将计算机的CPU、RAM、ROM、定时器/计数器和多种I/O口集成在一片芯片上,形成了芯片级的计算机,因此单片机早期的含义称为单片微型计算机(single chip microcomputer),直译为单片机,沿用至今。准确反映单片机本质的定义应是微控制器(microcontroller)。目前国外大多数厂家、学者已普遍改用microcontroller一词,其缩写为MCU(microcontroller unit),与MPU(microprocessor unit)相对应。国内仍沿用单片机一词,但其含义应是microcontroller,而非microcomputer,这是因为单片机无论

从功能还是从形态来说,都是作为控制领域用计算机的要求而诞生的。

目前也有人根据单片机的结构和微电子设计特点将单片机称为嵌入式微处理器(embedded microprocessor)或嵌入式微控制器(embedded microcontroller)。本书我们仍沿用传统的称呼——单片机。

### 1.3.1 单片机的发展历史

单片机出现的历史并不长,但发展十分迅猛。它的产生与发展 and 微处理器的产生与发展大体同步,自 1971 年美国 Intel 公司首先推出 4 位微处理器以来,它的发展到目前为止大致可分为以下五个阶段。

(1) 第一阶段(1971—1976) 单片机发展的初级阶段。1971 年 11 月,Intel 公司首先设计出集成度为一块芯片上 2000 只晶体管的 4 位微处理器 Intel 4004,并配有 RAM、ROM 和移位寄存器,构成了第一台 MCS-4 微处理器。而后又推出了 8 位微处理器 Intel 8008,以及其他各公司相继推出的 8 位微处理器。它们虽说还不是单片机,但从此拉开了研制单片机的序幕。

(2) 第二阶段(1976—1980) 低性能单片机阶段。以 1976 年 Intel 公司推出的 MCS-48 系列为代表,采用将 8 位 CPU、8 位并行 I/O 口、8 位定时器/计数器、RAM 和 ROM 等集成于一块芯片上的单片结构,虽然其寻址范围有限(不大于 4 KB),也没有串行 I/O 口,RAM、ROM 容量小,中断系统也较简单,但功能可满足一般工业控制和智能化仪器、仪表等的需要。这种将 CPU 与计算机外围电路集成到一块芯片上的技术,标志着单片机与通用 CPU 的分道扬镳,在构成新型工业微控制器方面取得了成功,为进一步发展单片机开辟了成功之路。

(3) 第三阶段(1980—1983) 高性能单片机阶段。这一阶段推出的高性能 8 位单片机普遍带有串行口,有多级中断处理系统,多个 16 位定时器/计数器;片内 RAM、ROM 的容量加大,且寻址范围可达 64 KB,个别片内还带有 A/D 转换接口。其典型产品为 1980 年 Intel 公司推出的 MCS-51 系列单片机,其他代表产品有 Motorola 公司的 6801 和 Zilog 公司的 Z8 等。这类单片机拓宽了单片机的应用范围,使之能用于智能终端、局部网络的接口等。因而,它是目前国内外产品的主流,各制造公司还在不断地改进和发展它。

(4) 第四阶段(1983—20 世纪 80 年代末) 16 位单片机阶段。1983 年,Intel 公司又推出了高性能的 16 位单片机 MCS-96 系列,由于其采用了最新的制造工艺,芯片集成度高达每片 12 万只晶体管。CPU 为 16 位,支持 16 位算术逻辑运算,并具有 32 位除 16 位的除法功能,片内 RAM 和 ROM 容量更进一步增大;除两个 16 位定时/计数器外,还可设定 4 个软件定时器,具有 8 个中断源,片内带有多通道高精度 A/D 转换和高速输入、输出部件(HSIO),运算速度和控制功能也大幅度提高,具有很强的实时处理能力。

(5) 第五阶段(20 世纪 90 年代至今) 单片机在集成度、功能、速度、可靠性、应用领域等全方位向更高水平发展。如:CPU 的位数有 8 位、16 位、32 位,而结构上进一步采用双 CPU 结构或内部流水线结构,以提高处理能力和运算速度;时钟频率高达 20 MHz,使指令执行速度相对加快;提供新型的串行总线结构,为系统的扩展与配置打下了良好的基础;增加新的特殊功能部件(如:PWM 输出、监视定时器 WDT,可编程计数器阵列 PCA, DMA 传输、调制解调器、通信控制器,浮点运算单元等);半导体制造工艺的不断改进,使芯片向高集成化、低功耗方向发展;等等。以上这些方面的发展,使单片机在大量数据的实时处理、高级通信系统、数字信号处理、复杂工业过程控制、高级机器人及局域网络等方面得到大量应用。

### 1.3.2 单片机的特点

单片机的芯片集成度很高,它将微型计算机的主要部件都集成在一块芯片上,具有下列特点。

(1) 体积小,质量轻,价格便宜,功耗低。

(2) 根据工控环境要求设计,且许多功能部件集成在芯片内部,其信号通道受外界影响小,故可靠性高,抗干扰性能优于采用一般的CPU。

(3) 控制功能强,运行速度快。其结构组成与指令系统都着重满足工业控制要求,有极丰富的条件分支转移指令,有很强的位处理功能和I/O口逻辑操作功能。

(4) 片内存储器的容量不可能很大;引脚也较少,I/O引脚常不够用,且兼第二功能乃至第三功能,但存储器和I/O口都易于扩展。

### 1.3.3 单片机的应用

由于单片机具有上述显著的特点,其应用领域有很多,无论是工业部门,还是民用部门乃至事业部门,到处都有它的身影。现将单片机的应用大致归纳为以下几个方面。

#### 1. 在智能仪器仪表中的应用

仪器仪表是单片机应用最多,最活跃的领域之一。在各类仪器仪表中引入单片机,使仪器仪表智能化,提高测试的自动化程度和精度,简化仪器仪表的硬件结构,提高其性价比。

#### 2. 在机电一体化产品中的应用

机电一体化产品是指集机械技术、微电子技术、计算机技术于一体,使其成为具有智能化特征电子产品,它是机械工业发展的方向。

#### 3. 在实时过程控制中的应用

单片机广泛地用于各种实时过程控制系统中,例如工业过程控制、过程监测、航空航天、尖端武器、机器人系统等各种实时控制系统。用单片机实时进行数据处理和控制,使系统保持最佳工作状态,提高系统的工作效率和产品的质量。

#### 4. 在生活中的应用

目前,国内外各种家用电器已普遍采用单片机来代替传统的控制电路。例如洗衣机、电冰箱、空调机、微波炉、电饭煲、收音机、音响、电风扇及许多高级电子玩具都配上了单片机,从而提高了自动化程度,增强了功能。当前家电领域的主要发展趋势是模糊控制,以形成众多的模糊控制家电产品,而单片机正是这些产品的最佳选择。单片机将使人类生活更加方便舒适,丰富多彩。

#### 5. 在其他方面的应用

单片机除以上各方面的应用外,它还广泛应用于办公自动化领域、商业营销领域、汽车及通信系统、计算机外部设备、模糊控制等各领域中。

总之,单片机已成为计算机发展和应用的一个重要方面。

### 1.3.4 单片机发展的未来

在未来相当长的时期内,8位单片机仍是单片机的主流机型。这是因为,一方面8位廉价型单片机已基本完全替代了4位单片机;另一方面,8位增强型单片机在速度及功能上向现在的16位单片机挑战。因此未来的机型可能是8位机与32位机共同发展的时代。从应用而言,32位单



片机在相当长的时间内数量不会很多,现有的 16 位单片机仍有相当长的生命周期。

从单片机的结构功能上看,单片机的发展趋势将向着大容量,高性能,小容量、低价格和外围电路内嵌结构等几个方面发展。

(1) 大容量 片内存储器容量进一步扩大。以往单片机内的 ROM 为 1~4KB, RAM 为 64~128B,因此在某些复杂控制场合,使其存储器容量不够,不得不进行外部扩充。为适应这种应用场合的要求,可以加大片内存储器的容量。目前单片机内部的 ROM 可达 4~8KB, RAM 可达 256B,有的单片机片内 ROM 可达 128KB, RAM 可达 1MB,寻址可达 16MB。今后,随着工艺技术的不断发展,单片机片内存储器容量将进一步扩大。

(2) 高性能 主要是指进一步改进单片机 CPU 的性能,加快指令运算速度和提高系统控制的可靠性。第一代 8 位单片机片内 CPU 及寄存器都采用 16 位,内部总线也采用 16 位,有的还采用流水线技术,指令的执行速度可达 100ns,堆栈的空间可达 64KB,以支持 C 语言开发。片内 RAM 在 1MB 以上,存储器寻址可达 16MB。

(3) 小容量、低价格 与上述相反,这类单片机的用途是把以往用数字逻辑集成电路组成的控制电路单片化。

(4) 外围电路内嵌结构 随着集成度的不断提高,尽可能把众多的各种外围功能器件集成在片内,除存储器、定时器/计数器等以外,片内还可以集成 A/D、D/A、DMA 控制器、声音发生器、监视定时器、液晶显示驱动器、彩色电视机和录像机用的锁相电路等。

(5) 增强 I/O 口功能 为减少外部驱动芯片,进一步增加单片机并行口的驱动能力,有的单片机可直接输出大电流和高电压,以便直接驱动负载。为进一步加快 I/O 口的传输速度,有的单片机还设置了高速 I/O 口,以最快的速度触发外部设备,也可以以最快的速度响应外部事件。

## 1.4 MCS-51 系列单片机介绍

MCS-51 系列单片机共有 10 多种产品,可分为两大系列:MCS-51 子系列与 MCS-52 子系列。MCS-51 子系列中主要有 8031、8051、8751 三种类型。MCS-52 子系列也有三种类型:8032、8052、8752。各子系列配置如表 1-3 所示。

表 1-3 MCS-51 系列单片机配置一览表

系 列	片内存储器				定时器/ 计数器	并行 I/O	串行 I/O	中断源	制造工艺
	无 ROM	片内 ROM	片内 EPROM	片内 RAM					
MCS-51 子系列	8031	8051 4KB	8751 4KB	128B	2×16b	4×8b	1	5	HMOS
	80C31	80C51 4KB	87C51 4KB	128B	2×16b	4×8b	1	5	CHMOS
MCS-52 子系列	8032	8052 8KB	8752 8KB	256B	3×16b	4×8b	1	6	HMOS
	80C32	80252 8KB	87C252 8KB	256B	3×16b	4×8b	1	7	CHMOS