

新编高等教育电子信息类规划教材

# 单片机原理与接口技术

丁向荣 主编 姚永平 主审

## ——基于STC15系列单片机



内容及编写特色:

1. 讲述可仿真、可编程、内置复位电路与时钟电路的8051单片机;
2. 将微机原理与单片机原理相融合, 汇编语言与C51“双语言”对照编程;
3. 精选应用实例, 强化单片机技术的实践性与应用性。

华信教育资源网上免费提供电子教案

新编高等教育电子信息类规划教材

# 单片机原理与接口技术

## ——基于 STC15 系列单片机

丁向荣 主编

姚永平 主审

電子工業出版社

**Publishing House of Electronics Industry**

北京·BEIJING

## 内 容 简 介

STC15 系列增强型 8051 单片机集成了上电复位电路与高精度 R/C 振荡器, 给单片机芯片加上电源就可跑程序; 集成了大容量的程序存储器、数据存储器以及 EEPRM, 集成了 A/D、PWM、SPI 等高性能接口部件, 可大大地简化单片机应用系统的外围电路, 使单片机应用系统的设计更加简捷, 系统性能更加高效、可靠。本教材以 STC15F2K60S2 单片机为主线, 系统地介绍了 STC15F2K60S2 单片机的硬件结构、指令系统与应用编程, 单片机应用系统的开发流程与接口设计, 同时提出多种实践模式: Keil C 集成开发环境、Proteus 仿真软件以及实物运行开发环境, 使得单片机的学习与应用变得更简单、更清晰。

本书可作为普通高校计算机类、电子信息类、电气自动化与机电一体化等专业的教学用书, 基础较好的高职高专也可选用本书。本书还可作为电子设计竞赛、电子设计工程师考证的培训教材。本书也是传统 8051 单片机应用工程师升级转型的重要参考书籍。

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

单片微机原理与接口技术——基于 STC15 系列单片机/丁向荣主编. —北京: 电子工业出版社, 2012. 8  
新编高等教育电子信息类规划教材

ISBN 978 - 7 - 121 - 17685 - 2

I. ① 单… II. ① 丁… III. ① 单片微型计算机 - 基础理论 - 高等职业教育 - 教材 ② 单片微型计算机 - 接口 - 高等职业教育 - 教材 IV. ① TP368. 1

中国版本图书馆 CIP 数据核字 (2012) 第 164128 号

策 划: 陈晓明 特约编辑: 张晓雪

责任编辑: 赵云峰

印 刷: 三河市鑫金马印装有限公司

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: 24. 75 字数: 634 千字

印 次: 2012 年 8 月第 1 次印刷

印 数: 3 000 册 定价: 46. 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, 联系及购电话: (010)88254888。

质量投诉请发邮件至 zlt@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

STC 系列单片机传承于 Intel8051 单片机，但在传统 8051 单片机框架基础上注入了新鲜血液，焕发出新的“青春”，在性能上进行了改进，在功能上进行了扩展。STC 单片机的在线下载编程功能（可节省仿真器、编程器）以及分系列的资源配置，增加了单片机型号的选择性，可根据单片机应用系统的功能要求选择合适的单片机，从而降低单片机应用系统的开发难度与开发成本，使得单片机应用系统更加简单、实效，提高了产品的性能价格比。STC 系列单片机相比传统的 8051 单片机具有强大的优势，它在我国应用市场中已占有较大的份额。传统的 8051 单片机教材与现实中使用的单片机存在较大的差距，本教材使单片机技术的教学与单片机技术的发展同步，保障单片机教学与单片机应用的无缝“链接”。

STC 系列单片机作为中国本土 MCU 的领航者，现已发展了 STC89/90 系列、STC10/11 系列、STC12 系列、STC15 系列，本教材选用 STC 最新系列 STC15 系列的 STC15F2K60S2 单片机作为主讲机型，系统地介绍了 STC15F2K60S2 单片机的硬件结构、指令系统与应用编程。

STC 系列单片机的指令系统和标准的 8051 内核完全兼容，原来讲解 8051 单片机的师资力量可以充分发挥以前讲解单片机原理及应用课程的经验；对于具有 8051 单片机知识的读者，也不存在转型困难的问题。

本教材力求实用性、应用性与易学性，以提高读者的工程设计能力与实践动手能力为目标。本书具有以下几方面的特点：

(1) 单片机机型贴近生产实际。STC 单片机是我国 8 位单片机应用中市场占有率最高的，更难能可贵的是，STC 单片机是我国本土的 MCU。

(2) 采用“双”语言编程。在绝大多数应用程序的编程中，采用汇编语言和 C 语言（C51）对照编程。采用汇编语言程序设计的学习更有利于加强对单片机的理解，而 C51 在功能、结构上以及可读性、可移植性、可维护性方面都有非常明显的优势。

(3) 理论联系实际。在学习单片机指令系统前的第 3 章就专门介绍了单片机应用的开发工具，贯穿程序的编辑、编译、下载与调试。强化单片机知识的应用性与实践性，不论是一条指令，或若干条指令，或一个程序段都可以用开发工具进行仿真调试或在线联机调试。

(4) 强化单片机应用系统的概念。学习单片机就是为了能开发与制作有具体意义的单片机应用系统，第 13 章介绍了单片机基本的外围接口技术与典型单片机应用系统的设计与开发。

(5) 在教材的编写中，直接与 STC 单片机的创始人姚永平先生进行密切沟通与交流，姚永平先生亲自担任本教材的主审，确保了教材内容的系统性与正确性。

(6) 开发与教材配套的 STC-15 型单片机通用开发板，建立与教材同步的 STC 教学资源网站（[www.stcstudy.com](http://www.stcstudy.com)）。

在本书的编写过程中，深圳宏晶科技有限公司技术部工程人员在技术上给予了大力支持和帮助，公司姚永平总经理对全书进行了认真审阅，并提出了宝贵意见。杨军博士给本教材提供了许多有益的建议，在此，对所有提供帮助的人表示感谢！

由于编者水平有限。书中定有疏漏和不妥之处，敬请读者不吝指正！垦请您的宝贵意见，可发电子邮件到：[dingxiangrong65@163.com](mailto:dingxiangrong65@163.com)，与作者进一步沟通与交流。

编者

2012.2 于广州

# 序

21 世纪全球全面进入了计算机智能控制/计算时代，而其中的一个重要方向就是以单片机为代表的嵌入式计算机控制/计算。由于最适合中国工程师/学生入门的 8051 单片机有 30 多年的应用历史，绝大部分工科院校均有此必修课，有几十万名对该单片机十分熟悉的工程师可以相互交流开发/学习心得，有大量的经典程序和电路可以直接套用，从而大幅降低了开发风险，极大地提高了开发效率，这也是宏晶科技基于 STC8051 系列单片机产品的巨大优势。

Intel 8051 技术诞生于 20 世纪 70 年代，不可避免地面临着落伍的危险，如果不对其进行大规模创新，我国的单片机教学与应用就会陷入被动局面。为此，宏晶科技对 STC8051 单片机进行了全面的技术升级与创新：全部采用 Flash 技术（可反复编程 10 万次以上）和 ISP/IAP（在系统可编程/在应用可编程）技术；针对抗干扰进行了专门设计，超强抗干扰；进行了特别加密设计，如宏晶 STC15 系列现无法解密；对传统 8051 单片机进行了全面提速，指令速度最快提高了 24 倍；大幅提高了集成度，如集成了 A/D、CCP/PCA/PWM（PWM 还可当 D/A 使用）、高速同步串行通信端口 SPI、高速异步串行通信端口 UART（如宏晶 STC15F2K60S2 系列集成了两个串行口，分时复用可当 5 组串口使用）、定时器（STC15F2K60S2 系列最多可实现 6 个定时器）、看门狗、内部高精度时钟（ $\pm 1\%$  温漂， $-40^{\circ}\text{C} \sim +85^{\circ}\text{C}$  之间，可彻底省掉外部昂贵的晶振）、内部高可靠复位电路（可彻底省掉外部复位电路）、大容量 SRAM（如 STC15F2K60S2 系列集成了 2KB 的 SRAM）、大容量 EEPROM、大容量 Flash 程序存储器等。

在中国民间草根企业掌握了 Intel 8051 单片机技术，以“初生牛犊不怕虎”的精神，击溃了欧美竞争对手之后，正在向 32 位前进的途中，此时欣闻官方国家队也已掌握了 Intel 80386 通用 CPU 技术，不由想起“老骥伏枥，志在千里”这句话，相信经过数代人艰苦奋斗，我们一定会赶上和超过世界先进水平！

明知山有虎，偏向虎山行。

感谢 Intel 公司发明了经久不衰的 8051 体系结构，感谢丁向荣老师的新书，保证了中国 30 年来的单片机教学与世界同步。

STC 宏晶科技：姚永平

www. STCMCU. com

2012 - 01 - 15

# 目 录

<b>第 1 章 微型计算机基础</b> .....	(1)
1.1 数制与编码 .....	(1)
1.1.1 数制及转换方法 .....	(1)
1.1.2 微型计算机中数的表示方法 .....	(3)
1.1.3 微型计算机中常用编码 .....	(5)
1.2 微型计算机的基本组成 .....	(6)
1.3 指令、程序与编程语言 .....	(8)
1.4 微型计算机的工作过程 .....	(8)
1.5 微型计算机的应用形态 .....	(10)
本章小结 .....	(10)
习题 1 .....	(11)
<b>第 2 章 STC15F2K60S2 单片机增强型 8051 内核</b> .....	(12)
2.1 单片机概述 .....	(12)
2.1.1 单片机的概念 .....	(12)
2.1.2 常见单片机 .....	(12)
2.1.3 STC 系列单片机 .....	(13)
2.2 STC15F2K60S2 系列单片机资源概述与引脚功能 .....	(14)
2.2.1 STC15F2K60S2 系列单片机资源与功能概述 .....	(14)
2.2.2 STC15F2K60S2 单片机引脚功能 .....	(14)
2.3 STC15F2K60S2 单片机的内部结构 .....	(18)
2.3.1 STC15F2K60S2 单片机的内部结构 .....	(18)
2.3.2 CPU 结构 .....	(19)
2.4 STC15F2K60S2 单片机的存储结构 .....	(20)
2.5 STC15F2K60S2 单片机的并行 I/O 口 .....	(25)
2.5.1 STC15F2K60S2 单片机的并行 I/O 口与工作模式 .....	(25)
2.5.2 STC15F2K60S2 单片机的并行 I/O 口的结构 .....	(25)
2.5.3 STC15F2K60S2 单片机并行 I/O 口的使用注意事项 .....	(27)
2.6 STC15F2K60S2 单片机的时钟与复位 .....	(29)
2.6.1 STC15F2K60S2 单片机的时钟 .....	(29)
2.6.2 STC15F2K60S2 单片机的复位 .....	(31)
本章小结 .....	(33)
习题 2 .....	(34)
<b>第 3 章 单片机应用的开发工具</b> .....	(35)
3.1 Keil $\mu$ Vision2 集成开发环境 .....	(35)
3.1.1 Keil $\mu$ Vision2 集成开发环境概述 .....	(35)

3.1.2 Keil C 集成开发环境下的程序编辑、编译与调试 .....	(36)
3.2 STC 系列单片机在线编程 .....	(40)
3.2.1 STC 系列单片机在系统可编程 (ISP) 典型应用线路图 .....	(40)
3.2.2 STC 系列单片机 PC 端下载软件的使用 .....	(42)
3.2.3 虚拟串口的应用 .....	(43)
3.2.4 STC 仿真器 .....	(44)
3.3 单片机学习的实践模式 .....	(46)
3.3.1 仿真模式 .....	(46)
3.3.2 利用 Proteus 模拟仿真软件进行调试 .....	(47)
3.3.3 在线系统调试模式 .....	(51)
本章小结 .....	(55)
习题 3 .....	(55)
<b>第 4 章 STC15F2K60S2 单片机的指令系统</b> .....	<b>(56)</b>
4.1 概述 .....	(56)
4.2 数据传送类指令 .....	(61)
4.3 算术运算类指令 (24 条) .....	(68)
4.4 逻辑运算类与循环移位类指令 (24 条) .....	(74)
4.5 控制转移类指令 (17 条) .....	(78)
4.6 位操作类指令 (17 条) .....	(84)
本章小结 .....	(89)
习题 4 .....	(89)
<b>第 5 章 STC15F2K60S2 单片机的程序设计</b> .....	<b>(93)</b>
5.1 汇编语言程序设计 .....	(93)
5.1.1 程序编制的方法和技巧 .....	(93)
5.1.2 程序的模块化设计 .....	(94)
5.1.3 伪指令 .....	(95)
5.2 基本程序结构与程序设计举例 .....	(98)
5.3 C51 程序设计 .....	(107)
5.3.1 C51 基础 .....	(108)
5.3.2 C51 程序设计 .....	(116)
本章小结 .....	(124)
习题 5 .....	(124)
<b>第 6 章 STC15F2K60S2 单片机存储器的应用</b> .....	<b>(126)</b>
6.1 STC15F2K60S2 单片机的程序存储器 .....	(126)
6.2 STC15F2K60S2 单片机的基本 RAM .....	(127)
6.3 STC15F2K60S2 单片机的扩展 RAM (XRAM) .....	(128)
6.4 STC15F2K60S2 单片机的 EEPROM (数据 Flash) .....	(132)
本章小结 .....	(142)
习题 6 .....	(142)
<b>第 7 章 STC15F2K60S2 单片机中断系统</b> .....	<b>(143)</b>
7.1 中断系统概述 .....	(143)
7.1.1 中断系统的几个概念 .....	(143)



7.1.2	中断的技术优势	(144)
7.1.3	中断系统需要解决的问题	(144)
7.2	STC15F2K60S2 单片机的中断系统	(145)
7.2.1	STC15F2K60S2 单片机的中断请求	(145)
7.2.2	STC15F2K60S2 单片机的中断响应	(151)
7.2.3	STC15F2K60S2 单片机中断应用举例	(154)
7.3	STC15F2K60S2 单片机外部中断的扩展	(155)
	本章小结	(158)
	习题 7	(158)
<b>第 8 章</b>	<b>STC15F2K60S2 单片机的定时/计数器</b>	<b>(159)</b>
8.1	STC15F2K60S2 单片机定时/计数器 (T0/T1) 的结构和工作原理	(159)
8.2	STC15F2K60S2 单片机定时/计数器 (T0/T1) 的控制	(160)
8.3	STC15F2K60S2 单片机定时/计数器 (T0/T1) 的工作方式	(162)
8.4	STC15F2K60S2 单片机定时/计数器 (T0/T1) 的应用举例	(167)
8.4.1	STC15F2K60S2 单片机定时/计数器 (T0/T1) 的定时应用	(167)
8.4.2	STC15F2K60S2 单片机定时/计数器 (T0/T1) 的计数应用	(171)
8.4.3	单片机秒表的设计	(173)
8.5	STC15F2K60S2 单片机的定时器 T2	(176)
8.5.1	STC15F2K60S2 单片机的定时器 T2 的电路结构	(176)
8.5.2	STC15F2K60S2 单片机的定时/计数器 T2 的控制寄存器	(176)
8.6	STC15F2K60S2 单片机的可编程时钟输出功能	(177)
8.6.1	STC15F2K60S2 单片机的可编程时钟: CLKOUT0、CLKOUT1、CLKOUT2	(177)
8.6.2	STC15F2K60S2 单片机可编程时钟的应用举例	(178)
	本章小结	(179)
	习题 8	(180)
<b>第 9 章</b>	<b>STC15F2K60S2 单片机的串行口</b>	<b>(182)</b>
9.1	串行通信基础	(182)
9.2	STC15F2K60S2 单片机的串行口 1	(184)
9.2.1	串行口 1 的控制寄存器	(185)
9.2.2	串行口 1 的工作方式	(187)
9.2.3	串行口的波特率	(193)
9.2.4	串行口的应用举例	(195)
9.3	STC15F2K60S2 单片机串行口 2	(209)
9.3.1	STC15F2K60S2 单片机串行口 2 控制寄存器	(209)
9.3.2	STC15F2K60S2 单片机串行口 2 的工作方式与波特率	(210)
9.4	STC15F2K60S2 单片机与 PC 机的通信	(210)
9.4.1	单片机与 PC 机 RS-232C 串行通信的接口设计	(210)
9.4.2	单片机与 PC 机 USB 总线通信的接口设计	(213)
9.4.3	单片机与 PC 机串行通信的程序设计	(214)
9.5	STC15F2K60S2 单片机串行口 1 的中继广播方式	(216)
9.6	STC15F2K60S2 单片机串行口硬件引脚的切换	(216)
	本章小结	(217)
	习题 9	(218)

<b>第 10 章</b>	<b>STC15F2K60S2 单片机的 A/D 转换</b>	(219)
10.1	STC15F2K60S2 单片机 A/D 模块的结构	(219)
10.2	STC15F2K60S2 单片机 A/D 模块的控制	(220)
10.3	STC15F2K60S2 单片机 A/D 模块的应用	(222)
	本章小结	(226)
	习题 10	(227)
<b>第 11 章</b>	<b>STC15F2K60S2 单片机 CCP/PCA/PWM 模块</b>	(228)
11.1	STC15F2K60S2 单片机的 CCP/PCA/PWM 模块的结构	(228)
11.2	PCA 模块的特殊功能寄存器	(229)
11.3	CCP/PCA 模块的工作模式与应用举例	(232)
11.4	PCA 模块功能引脚的切换	(245)
	本章小结	(246)
	习题 11	(246)
<b>第 12 章</b>	<b>STC15F2K60S2 单片机的 SPI 接口</b>	(247)
12.1	SPI 接口的结构	(247)
12.2	SPI 接口的特殊功能寄存器	(248)
12.3	SPI 接口的数据通信	(250)
12.4	SPI 接口的应用举例	(254)
12.5	SPI 接口功能引脚的切换	(263)
	本章小结	(264)
	习题 12	(264)
<b>第 13 章</b>	<b>单片机应用系统设计与接口技术</b>	(265)
13.1	单片机应用系统的开发流程	(265)
13.1.1	单片机应用系统的设计原则	(265)
13.1.2	单片机应用系统的开发流程	(265)
13.2	单片机人机对话接口设计	(269)
13.2.1	键盘接口与应用实例	(269)
13.2.2	LED 数码显示接口与应用实例	(279)
13.2.3	LCD 显示接口与应用实例	(286)
13.3	串行总线接口技术与应用设计	(312)
13.3.1	单总线数字温度传感器 DS18B20 与应用实例	(312)
13.3.2	I <sup>2</sup> C 串行总线原理与应用	(322)
13.3.3	I <sup>2</sup> C 总线应用实例——基于时钟芯片 PCF8563 的电子时钟的实现	(329)
13.4	电机控制与应用设计	(340)
13.4.1	直流电机的控制	(340)
13.4.2	步进电机的控制	(347)
13.4.3	步进电机与单片机的接口	(348)
13.5	STC15F2K60S2 单片机的低功耗设计	(356)
13.5.1	STC15F2K60S2 单片机的慢速模式	(356)
13.5.2	STC15F2K60S2 单片机的空闲（等待）模式与停机（掉电）模式	(356)
13.6	STC15F2K60S2 单片机的看门狗定时器	(364)
13.6.1	看门狗定时器	(364)

13.6.2 STC15F2K60S2 单片机的看门狗定时器 .....	(364)
13.6.3 STC15F2K60S2 单片机的看门狗定时器的使用 .....	(365)
本章小结 .....	(366)
习题 13 .....	(367)
附录 1 ASCII 码表 .....	(368)
附录 2 STC15F2K60S2 单片机指令系统表 .....	(369)
附录 3 STCISP 下载编程软件实用程序简介 .....	(373)
附录 4 STC-ISP 的自定义下载 .....	(376)
附录 5 STC15 系列单片机功能特性表 .....	(381)
参考文献 .....	(384)

# 第 1 章 微型计算机基础

## 1.1 数制与编码

数制与编码是微型计算机的基本数字逻辑基础，是学习微型计算机的必备知识。数制与编码的知识一般会在数字逻辑或计算机文化基础中学习，但往往由于数制与编码的知识，与当前课程的关系并非“不可或缺”，又比较枯燥。在微型计算机原理或单片机的教学中，教师普遍感觉到，学生在这方面的知识基础不扎实。在此，提纲挈领再理一理。

### 1.1.1 数制及转换方法

所谓数制就是计数的方法，通常采用进位计数制。在微型机算机的学习与应用中，主要有十进制、二进制和十六进制三种计数方法。日常生活中采用的是十进制；微型计算机只能识别和处理数字信息，微型计算机硬件电路采用的是二进制，但为了更好地记忆与描述微型计算机的地址和程序代码、运算数字，一般采用十六进制。

#### 1. 各种进位计数制及其表示方法

见表 1.1 所示。

表 1.1 二进制、十进制与十六进制的计数规则与表示方法

进位制	计数规则	基数	各位的权	数码	权值展开式	表示法	
						后缀字符	下标
二进制	逢二进一	2	$2^i$	0, 1	$(b_{n-1} \cdots b_1 b_0 . b_{-1} \cdots b_{-m})_2 = \sum_{i=-m}^{n-1} b_i \times 2^i$	B	$( )_2$
十进制	逢十进一	10	$10^i$	0, 1, ..., 9	$(d_{n-1} \cdots d_1 d_0 . d_{-1} \cdots d_{-m})_{10} = \sum_{i=-m}^{n-1} d_i \times 10^i$	D	$( )_{10}$
						通常缺省表示	
十六进制	逢十六进一	16	$16^i$	0, 1, ..., 9, A, ..., F	$(h_{n-1} \cdots h_1 h_0 . h_{-1} \cdots h_{-m})_{16} = \sum_{i=-m}^{n-1} h_i \times 16^i$	H	$( )_{16}$

#### 2. 数制之间的转换

任意进制之间相互转换，整数部分和小数部分必须分别进行。各进制的相互转换关系如图 1.1 所示。

(1) 二进制、十六进制转换为十进制。将二进制、十六进制数按权值展开式展开相加所得数，即为十进制数。

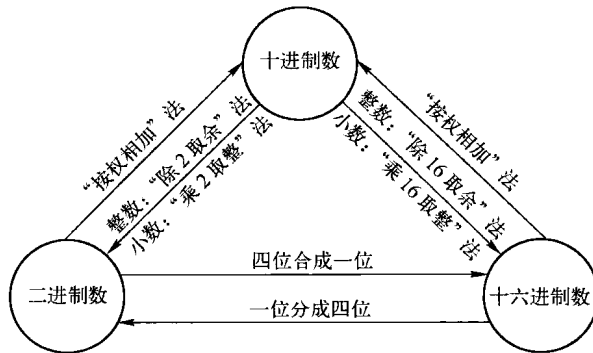


图 1.1 各进制的相互转换关系图

(2) 十进制转换为二进制。十进制转换为二进制要分成整数部分与小数部分，而且其转换方法完全不同。

① 整数部分——除 2 取余法，倒序排列，如下所示：

2	84	余数	二进制数码
2	42	..... 0	b <sub>0</sub>
2	21	..... 0	b <sub>1</sub>
2	10	..... 1	b <sub>2</sub>
2	5	..... 0	b <sub>3</sub>
2	2	..... 1	b <sub>4</sub>
2	1	..... 0	b <sub>5</sub>
	0	..... 1	b <sub>6</sub>

$$\therefore (84)_{10} = (1010100)_2$$

② 小数部分——乘 2 取整法，顺序排列，如下所示：

	0.6875	
	× 2	
b <sub>-1</sub>	1 ← ..... 1.3750	× 2
b <sub>-2</sub>	0 ← ..... 0.7500	× 2
b <sub>-3</sub>	1 ← ..... 1.5000	× 2
b <sub>-4</sub>	1 ← ..... 1.0000	

$$\therefore (0.6875)_{10} = (0.1011)_2$$

将上述两部分合起来，则有：

$$(84.6875)_{10} = (1010100.1011)_2$$

(3) 二进制与十六进制互转。

① 二进制转换为十六进制。以小数点为界，往左、往右 4 位二进制数为一组，每 4 位二进制数用 1 位十六进制表示，往左高位不够用 0 补齐，往右低位不够用 0 补齐，例如：

$$(111101.011101)_2 = (\underline{0011} \underline{1101} . \underline{0111} \underline{0100})_2 = (3D.74)_{16}$$

② 十六进制转换为二进制。每位十六进制数用 4 位二进制数表示，再将整数部分最高位的 0 去掉，小数部分最低位的 0 去掉，例如，

$$(3C20.84)_{16} = (\underline{0011} \underline{1100} \underline{0010} \underline{0000} . \underline{1000} \underline{0100})_2 = (11110000100000.100001)_2$$

数制转换工具：利用 PC 机附件中的计算器（科学型）可实现各数制间的相互转换。单击任务栏“开始”按钮，选择“所有程序”→“附件”→“计算器”，即可打开计算器工具，在计算器工具界面“查看”菜单栏中选择“科学型”，如图 1.2 所示。

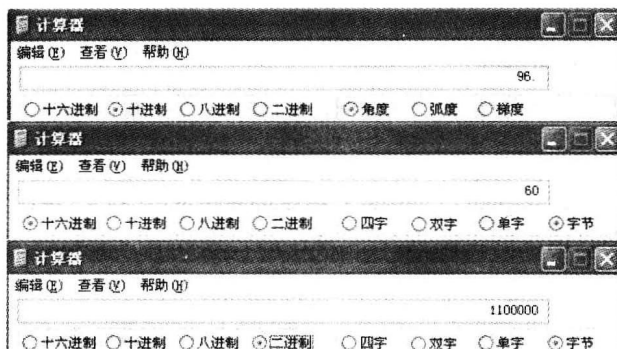


图 1.2 科学型计算器与各进制转换

转换方法：先选择被转换数制的类型，输入转换数字，再选择目标转换数制类型，此时看到的的就是转换后的数字。如 96 转换为十六进制、二进制，先选择数制类型为十进制，如图 1.2 上部所示，在输入框中输入数字 96，然后再选择数制类型为十六进制此时显示框中看到的数字即为转换后的十六进制数字 60，如图 1.2 中部所示；再选择数制类型为二进制，此时显示框中看到的数字即为转换后的二进制数字 1100000，如图 1.2 下部所示。

### 3. 二进制数的运算规则

(1) 加法运算规则。

$$0 + 0 = 0, 0 + 1 = 1, 1 + 1 = 0 \text{ (有进位)}$$

(2) 减法运算规则。

$$0 - 0 = 0, 1 - 0 = 1, 1 - 1 = 0, 0 - 1 = 1 \text{ (有借位)}$$

(3) 乘法运算规则。

$$0 \times 0 = 0, 1 \times 0 = 0, 1 \times 1 = 1$$

## 1.1.2 微型计算机中数的表示方法

### 1. 机器数与真值

数学中的正、负用符号“+”和“-”表示，计算机中是如何表示数的正、负呢？在计算机中数据存放在存储单元内，而每个存储单元则由若干二进制位组成，其中每一位或是 0 或是 1，刚好可以对应数的“+”号和“-”号，这样就可用一个数位来表示数的符号。在计算机中规定用“0”表示“+”，用“1”表示“-”。用来表示数的符号的数位被称为“符号位”（通常为最高数位），于是数的符号在计算机中已数码化了，但从表示形式上看符号位与数值位毫无区别。

设有两个数  $x_1, x_2$ ：

$$x_1 = +1011011 \text{ B}, x_2 = -1011011 \text{ B}$$

它们在计算机中分别表示为（带下画线部分为符号位，字长为 8 位）：

$$x_1 = \underline{0}1011011 \text{ B}, x_2 = \underline{1}1011011 \text{ B}$$

为了区分这两种形式的数，我们把机器中以编码形式表示的数称为机器数（上例中  $x_1$

= 01011011 B 及  $x_2 = 11011011\text{B}$ ), 而把原来一般书写形式表示的数称为真值 ( $x_1 = +1011011\text{ B}$  及  $x_2 = -1011011\text{B}$ )。

若一个数的所有数位均为数值位, 则该数为无符号数; 若一个数的最高数位为符号位而其他数位为数值位, 则该数为有符号数。由此可见, 对于同一存储单元, 它存放的无符号数和有符号数所能表示的数值范围是不同的 (如存储单元为 8 位, 当它存放无符号数时, 因有效的数值位为 8 位, 故该数的范围为 0 ~ 255; 当它存放有符号数时, 因有效的数值位为 7 位, 故该数的范围 (补码) 为 -128 ~ +127)。

## 2. 原码

对于一个  $n$  位二进制数, 如用最高数位表示该数的符号 (“0” 表示 “+” 号, “1” 表示 “-” 号), 其余各数位表示其数值本身, 则称为原码表示法。

若  $x = \pm x_{n-2} \cdots x_1 x_0$ , 则  $[x]_{\text{原码}} = x_{n-1} x_{n-2} \cdots x_1 x_0$

其中  $x_{n-1}$  为原机器数的符号位, 它满足:

$$x_{n-1} = \begin{cases} 0 & (x \geq 0 \text{ 时}) \\ 1 & (x < 0 \text{ 时}) \end{cases}$$

## 3. 反码

$$[x]_{\text{原}} = 0x_{n-2} \cdots x_1 x_0, \text{ 则 } [x]_{\text{反}} = [x]_{\text{原}}$$

$$[x]_{\text{原}} = 1x_{n-2} \cdots x_1 x_0, \text{ 则 } [x]_{\text{反}} = \overline{1x_{n-2} \cdots x_1 x_0}$$

也就是说, 正数的反码与其原码相同 (反码 = 原码), 而负数的反码为符号位保持不变, 数值位按位取反。

## 4. 补码

(1) 补码的引进。首先以日常生活中经常遇到的钟表 “对时” 为例来说明补码的概念。假定现在是北京标准时间八时整, 而一只表却指向十时整, 为了校正此表, 可以采用倒拨和顺拨两种方法: 倒拨就是反时针减少 2 小时 (把倒拨视为减法, 相当于  $10 - 2 = 8$ ), 时针指向 8; 还可将时针顺拨 10 小时, 时针同样也指向 8, 把顺拨视为加法, 相当于  $10 + 10 = 12$  (自动丢失)  $+ 8 = 8$ , 这自动丢失的数 (12) 就叫做摸 (mod), 上述的加法称为 “按摸 12 的加法”, 用数学式可表示为:

$$10 + 10 = 12 + 8 = 8 \pmod{12}$$

因时针转一圈会自动丢失一个数 12, 故  $10 - 2$  与  $10 + 10$  是等价的。称 10 和 -2 对模 12 互补, 10 是 -2 对模 12 的补码。引进补码概念后, 就可将原来的减法  $10 - 2 = 8$  转化为加法  $10 + 10$  (-2 的补码)  $= 12$  (自动丢失)  $+ 8 = 8 \pmod{12}$ 。

(2) 补码的定义。通过上面的例子不难理解计算机中负数的补码表示法。设寄存器 (或存储单元) 的位数为  $n$  位, 则它能表示的无符号数最大值为  $2^n - 1$ , 逢  $2^n$  进 1 (即  $2^n$  自动丢失)。换句话说, 在字长为  $n$  的计算机中, 数  $2^n$  和 0 的表示形式一样。若机器中的数以补码表示, 则数的补码以  $2^n$  为模, 即

$$[x]_{\text{补}} = 2^n + x \pmod{2^n}$$

若  $x$  为正数, 则  $[x]_{\text{补}} = x$ ; 若  $x$  为负数, 则  $[x]_{\text{补}} = 2^n + x = 2^n - |x|$ 。即负数  $x$  的补码等于模  $2^n$  加上其真值或减去其真值的绝对值。

在补码表示法中, 零只有唯一的表示形式:  $0000\cdots 0$ 。

(3) 求补码的方法。根据上述介绍可知, 正数的补码等于原码。下面介绍负数求补码的三种方法。

① 根据真值求补码。根据真值求补码就是根据定义求补码, 即有

$$[x]_{\text{补}} = 2^n + x = 2^n - |x|$$

即负数的补码等于  $2^n$  (模) 加上其真值, 或者等于  $2^n$  (模) 减去其真值的绝对值。

② 根据反码求补码 (推荐使用方法)。

$$[x]_{\text{补}} = [x]_{\text{反}} + 1$$

③ 根据原码求补码。负数的补码等于其反码加 1, 这也可理解为负数的补码等于其原码各位 (除符号位外) 取反并在最低位加 1。如果反码的最低位是 1, 它加 1 后就变成 0, 并产生向次最低位的进位; 如次最低位也为 1, 它同样变成 0, 并产生向其高位的进位 (这相当于在传递进位), 这进位一直传递到第 1 个为 0 的位为止。于是可得到这样的转换规律: 从反码的最低位起直到第一个为 0 的位以前 (包括第一个为 0 的位), 一定是 1 变 0, 第一个为 0 的位以后的位都保持不变, 由于反码是由原码求得, 因此可得从原码求补码的规律为: 从原码的最低位开始到第 1 个为 1 的位之间 (包括此位) 的各位均不变, 此后各位取反, 但符号位保持不变。

特别要指出, 在计算机中凡是带符号的数一律用补码表示且符号位参加运算, 其运算结果也是用补码表示, 若结果的符号位为 “0”, 则表示结果为正数, 此时可以认为它是原码形式表示的 (正数的补码即为原码); 若结果的符号位为 “1”, 则表示结果为负数, 它是补码形式表示的, 若是要用原码来表示该结果, 还需要对结果求补 (即除符号位外按位取反加 1), 即

$$[[x]_{\text{补}}]_{\text{补}} = [x]_{\text{原}}$$

### 1.1.3 微型计算机中常用编码

微型计算机不但要处理数值计算问题, 而且还要处理大量非数值计算问题, 因此除了直接给出二进制数外, 不论是十进制数还是英文字母、汉字以及某些专用符号都必须编成二进制代码, 这样它们才能被计算机识别、接收、存储、传送及处理。

#### 1. 十进制数的编码

在微型计算机中, 十进制数除了转换成二进制数外, 还可用二进制数对其进行编码: 用 4 位二进制数表示 1 位十进制数, 使它既具有二进制数的形式又具有十进制数的特点。二进制数又称为 BCD 码 (Binary - Coded Decimal), 它有 8421 码、5421 码、2421 码以及余 3 码等几种编码形式, 其中最常用的是 8421 码。8421 码与十进制数的对应关系见表 1.2 所示, 每位二进制数位都有固定的 “权”, 各数位的权从左到右分别为  $2^3$ 、 $2^2$ 、 $2^1$ 、 $2^0$ , 即 8、4、2、1, 这与自然二进制数的权完全相同, 故 8421BCD 码又称为自然权 BCD 码。其中 1010 ~ 1111 这 6 个代码, 是不允许出现的, 属非法 8421BCD 码。



表 1.2 8421BCD 码编码表

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

BCD 码低位与高位之间是“逢十进一”，而 4 位二进制数（即十六进制数）是“逢十六进一”，用二进制加法器进行 BCD 码运算时，如果 BCD 码运算的低、高位的和都在 0~9 之间，则其加法运算规则与二进制加法完全一样；如果相加后某位（BCD 码位，低 4 位或高 4 位）的和大于 9 或产生了进位，则此位应进行“加 6 调整”。通常在微型计算机中，都设置有 BCD 码的调整电路，机器执行一条十进制调整指令，机器就会自动根据刚才的二进制加法结果进行修正。BCD 码向高位借位是“借一当十”，而 4 位二进制数（1 位十六进制数）是“借一当十六”，因此在进行 BCD 码减法运算时，如果某位（BCD 码位）有借位时，必须在该位进行“减 6 调整”。

## 2. 字符编码

微型计算机需要进行非数值处理（如指令、数据的输入、文字的输入及处理等），必须对字母、文字以及某些专用符号进行编码。微型计算机系统的字符编码多采用美国信息交换标准代码——ASCII 码（American Standard Code for Information Interchange），ASCII 码是 7 位代码，共有 128 个字符，其中 94 个是图形字符，可在字符印刷或显示设备上打印出来，包括数字符号 10 个、英文大小写字母共 52 个以及其他字符 32 个，另外 34 个是控制字符，包括传输字符、格式控制字符、设备控制字符、信息分隔符和其他控制字符，这类字符不打印、不显示，但其编码可进行存储，在信息交换中起控制作用。其中，数字 0~9 对应的 ASCII 码为 30H~39H，英文大写字母 A~Z 对应的 ASCII 码为 41H~5AH，小写字母 a~z 对应的 ASCII 码为 61H~7AH，这些规律性对今后的码制转换的编程非常有用。

我国于 1980 年制定了国家标准 GB1988-80，即“信息处理交换用的 7 位编码字符集”，其中除了用人民币符号“¥”代替美元符号“\$”外，其余与 ASCII 码相同。

## 1.2 微型计算机的基本组成

随着集成电路技术的飞速发展，1971 年 1 月，Intel 公司的德·霍夫将运算器、控制器以及一些寄存器集成在一块芯片上，称为微处理器或中央处理单元（简称 CPU），形成了以微处理器为核心的总线结构框架。

如图 1.3 所示为微型计算机的组成框图，由微处理器、存储器（ROM、RAM）和输入/输出接口（I/O 接口）及连接它们的总线组成。微型计算机配上相应的输入/输出设备（如键盘、显示器）就构成了微型计算机系统。