

STC大学计划推荐教材
全国大学生电子设计竞赛参考教材
“蓝桥杯”全国软件专业人才设计与创业大赛参考教材

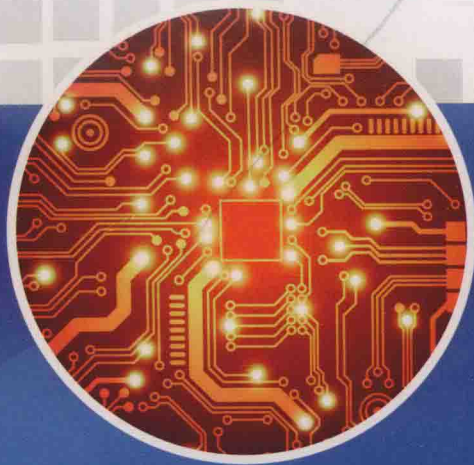
单片机原理 与接口技术

——基于可仿真的STC8系列单片机

丁向荣 | 编著

姚永平 | 主审

- ★ 以STC新一代产品STC8A8K64S4A12单片机为载体
- ★ 采用“汇编语言+C语言”双语言教学
- ★ 精选工程训练实例，配有类型丰富的习题
- ★ 适用于院校教学和大赛培训



 中国工信出版集团

 电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

STC 大学计划推荐教材

全国大学生电子设计竞赛参考教材

“蓝桥杯”全国软件专业人才设计与创业大赛参考教材

单片微机原理与接口技术

——基于可仿真的 STC8 系列单片机

丁向荣 编著

姚永平 主审

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书以 STC8 系列单片机中的 STC8A8K64S4A12 单片机为介绍对象,采用“汇编语言+C 语言”双语言教学,精选工程训练实例,设计类型多样化的习题。教材内容包括微型计算机基础、STC8A8K64S4A12 单片机增强型 8051 内核、STC 系列单片机的应用系统的开发工具、STC8A8K64S4A12 单片机的指令系统与汇编语言程序设计、C51 与 C51 程序设计、STC8A8K64S4A12 单片机的存储器与应用编程、STC8A8K64S4A12 单片机的定时/计数器、STC8A8K64S4A12 单片机中断系统、STC8A8K64S4A12 单片机的串行接口、人机对话接口的应用设计、STC8A8K64S4A12 单片机的比较器、STC8A8K64S4A12 单片机的 A/D 转换模块、STC8A8K64S4A12 单片机的 PCA 模块、STC8A8K64S4A12 单片机的增强型 PWM 模块、STC8A8K64S4A12 单片机的 SPI 接口、STC8A8K64S4A12 单片机的 I²C 通信接口,以及 STC8A8K64S4A12 单片机的低功耗设计与可靠性设计等内容。

本书可作为高等学校电子信息类、电子通信类、自动化类、计算机应用类专业“单片机原理与应用”或“微机原理”课程的教材,也可作为电子设计竞赛、单片机应用工程师考证的培训教材。此外,本书也是传统 8051 单片机应用工程师升级转型的参考书籍。

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:基于可仿真的 STC8 系列单片机 / 丁向荣编著. —北京:电子工业出版社, 2020.3
ISBN 978-7-121-38723-4

I. ①单… II. ①丁… 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—高等学校—教材 IV. ①TP368.1

中国版本图书馆 CIP 数据核字(2020)第 041184 号

责任编辑:郭乃明 特约编辑:田学清

印 刷:三河市良远印务有限公司

装 订:三河市良远印务有限公司

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:787×1092 1/16 印张:30.5 字数:780.8 千字

版 次:2020 年 3 月第 1 版

印 次:2020 年 3 月第 1 次印刷

定 价:79.00 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, 88258888。

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

本书咨询联系方式:(010) 88254561, QQ34825072。

前言

单片机技术是现代电子系统设计、智能控制的核心技术，与其相关的专业有应用电子、电子信息、电子通信、物联网技术、机电一体化、电气自动化、工业自动化、计算机应用等。本书是作者以其三十余年单片机应用经历和教学经验为基础，精心打造的以 STC 系列单片机中的 STC8A8K64S4A12 单片机为介绍对象的单片机课程教材。

STC 系列单片机传承于 8051 单片机，其在传统 8051 单片机的基础上注入了新鲜血液，焕发了新的“活力”。STC 的创造者深圳市宏晶科技有限公司（以下简称宏晶科技），对 8051 单片机进行了全面的技术改革与创新：采用了 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，增强了产品抗干扰能力；进行了特别加密设计；全面提升了工作速度，指令运行速度提高为原来的 24 倍；大幅提高了集成度，集成了 A/D 功能模块、CCP/PCA/PWM（PWM 还可当成 DAC 使用）功能模块、SPI 接口、I²C 接口，以及更多的 UART、更多的定时器、WDT、内部高精度时钟源（±1% 温漂，工作温度为 -40~+85℃，可省掉昂贵的外部晶振）、内部高可靠复位电路（可省掉外部复位电路）、大容量 SRAM、大容量 EEPROM、大容量 Flash 程序存储器等。STC 系列单片机的在线下载编程、在线仿真功能，以及分系列的资源配置，增加了单片机型号的可选择性，使用户可根据单片机应用系统的功能要求选择合适的单片机，从而降低了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的开发更加简单、高效，提高了单片机应用产品的性能价格比。

作者根据多年单片机教学经验发现，学生在学习单片机课程时普遍存在一个现象，即明白单片机课程很重要，但觉得很难，不知道怎么学。对此，作者认为学生在学习单片机课程前要明白如下三件事。

（1）有什么用？单片机技术是现代电子系统设计的核心技术，学习单片机就是利用单片机设计具有智能化、自动化功能的单片机应用系统。

（2）要学什么？简单地说，就是学习单片机有哪些资源，以及如何使用这些资源。

（3）怎么学？单片机的学习应该分为三个方面：一是掌握一种编程语言（C 语言或者汇编语言）；二是掌握单片机应用系统的开发工具及辅助工具（Keil C 集成开发环境、STC-ISP 在线编程软件与 Proteus 仿真软件）；三是学习单片机的各种资源特性与编程技巧。

本书将 STC8 系列单片机中的 STC8A8K64S4A12 单片机作为主讲机型，系统地介绍了

STC8A8K64S4A12 单片机的硬件结构、指令系统与应用编程。本书力求体现实用性、应用性与易学性，以提高读者的工程设计能力与实践能力为目标，主要具有以下几方面特点。

(1) 选用的单片机机型贴近生产实际：STC 系列单片机在我国 8 位单片机市场中占有率极高，STC8A8K64S4A12 单片机更是其中的佼佼者。

(2) 采用双语言编程：本书的应用编程大部分采用了汇编语言和 C 语言（C51 语言）。学习汇编语言更有利于加强学生对单片机的理解，但 C 语言的功能、结构，以及用其编写的程序的可读性、可移植性、可维护性相对而言都有非常明显的优势。

(3) 理论联系实际：本书第 3 章专门介绍了单片机应用系统的开发工具，这些开发工具的应用贯穿了程序的编辑、编译、下载与调试过程。本书在讲述相关知识时，强调单片机知识的应用性与实践性，不论是一条指令，还是一个程序段，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够开发与制作有具体意义的单片机应用系统，因此，本书在大多数章节中都配置了工程训练实例。

(5) 为了便于读者更好地理解教学内容与教学要求，本书设置了多样化的习题，如填空、选择、判断、问答与程序设计。

(6) 为便于读者学习与应用开发，本书在附录中提供了实用的技术资料。

(7) 本书所含的工程训练实例是基于 STC 官方发布的 STC8 学习板开发的，本书是宏晶科技 STC 大学推广计划的指定教材。

本书由丁向荣编著，宏晶科技在技术上给予了大力支持和帮助，在教材的编写过程中，作者与 STC 单片机的创始人姚永平先生进行了沟通与交流，并且邀请姚永平先生担任了本书的主审，对全书进行了认真审阅，提出了宝贵意见，确保了教材内容的系统性与正确性。在此，对所有提供过帮助的人表示感谢！

由于编者水平有限，书中难免有疏漏和不妥之处，敬请读者不吝指正！相关勘误或更新信息也会动态地公布在 STC 官网上。如需程序源代码或有其他建议，可发电子邮件到 dingxiangrong65@163.com，与作者进行沟通与交流。

编者

2019.10 于广州

目 录

第 1 章 微型计算机基础.....	1
1.1 数制与编码.....	1
1.1.1 数制及其转换方法.....	1
1.1.2 微型计算机中数的表示方法.....	4
1.1.3 微型计算机中常用编码.....	6
1.2 微型计算机原理.....	7
1.2.1 微型计算机的基本组成.....	7
1.2.2 指令、程序与编程语言.....	9
1.2.3 微型计算机的工作过程.....	10
1.2.4 微型计算机的应用形态.....	11
本章小结.....	11
习题与思考题.....	12
第 2 章 STC8A8K64S4A12 单片机增强型 8051 内核.....	13
2.1 单片机概述.....	13
2.1.1 单片机的概念.....	13
2.1.2 常见单片机.....	13
2.1.3 STC8 系列单片机.....	14
2.2 STC8A8K64S4A12 单片机资源概述与引脚功能.....	17
2.3 STC8A8K64S4A12 单片机的内部结构.....	22
2.3.1 内部结构框图.....	22
2.3.2 CPU 结构.....	22
2.4 STC8A8K64S4A12 单片机的存储结构.....	24
2.5 STC8A8K64S4A12 单片机的并行 I/O 口.....	29
2.5.1 并行 I/O 口的工作模式.....	29
2.5.2 并行 I/O 口的结构.....	30
2.5.3 并行 I/O 口的使用注意事项.....	32

2.6	STC8A8K64S4A12 单片机的时钟与复位	34
2.6.1	时钟	34
2.6.2	复位	36
	本章小结	38
	习题与思考题	39
第 3 章	STC 系列单片机应用系统的开发工具	41
3.1	Keil μ Vision4 集成开发环境	41
3.2	STC 系列单片机在线编程与在线仿真	50
3.2.1	STC 系列单片机在线编程电路	50
3.2.2	单片机应用程序的下载与运行	52
3.2.3	Keil μ Vision4 与 STC 仿真器的在线仿真	53
3.2.4	STC-ISP 在线编程软件的其他功能	55
3.3	Proteus 仿真软件简介	56
3.3.1	工程训练 3.1 Keil C 集成开发环境的操作使用	56
3.3.2	工程训练 3.2 STC 系列单片机的在线调试与在线仿真	64
3.3.3	工程训练 3.3 用 Proteus 仿真软件实现单片机应用系统的仿真	67
	本章小结	74
	习题	75
第 4 章	STC8A8K64S4A12 单片机的指令系统与汇编语言程序设计	78
4.1	STC8A8K64S4A12 单片机的指令系统	78
4.1.1	概述	78
4.1.2	数据传送类指令	84
4.1.3	算术运算类指令	89
4.1.4	逻辑运算与循环移位类指令	95
4.1.5	控制转移类指令	98
4.1.6	位操作类指令	104
4.2	汇编语言程序设计	108
4.2.1	汇编语言程序设计基础	108
4.2.2	基本程序结构与程序设计举例	113
4.2.3	工程训练 4.1 LED 数码管的驱动与显示 (汇编语言版)	122
	本章小结	125
	习题与思考题	126
第 5 章	C51 与 C51 程序设计	131
5.1	C51 基础	131
5.1.1	C51 数据类型	133
5.1.2	C51 的变量	135

5.1.3	8051 单片机特殊功能寄存器变量的定义	137
5.1.4	8051 单片机位寻址区 (20H~2FH) 位变量的定义	138
5.1.5	函数的定位	138
5.1.6	中断服务函数	139
5.1.7	函数的递归调用与再入函数	140
5.1.8	在 C51 中嵌入汇编语言程序	140
5.2	C51 程序设计	141
5.2.1	C51 程序框架	141
5.2.2	C51 程序设计举例	145
5.2.3	工程训练 5.1 LED 数码管驱动与显示 (C 语言版)	146
	本章小结	148
	习题与思考题	149
第 6 章	STC8A8K64S4A12 单片机的存储器与应用编程	151
6.1	程序存储器	151
6.2	基本 RAM	153
6.3	扩展 RAM (XRAM)	154
6.4	EEPROM	158
6.4.1	工程训练 6.1 片内扩展 RAM 的测试	164
6.4.2	工程训练 6.2 EEPROM 的测试	165
	本章小结	169
	习题与思考题	169
第 7 章	STC8A8K64S4A12 单片机的定时/计数器	172
7.1	定时/计数器 T0、T1 的结构和工作原理	172
7.2	定时/计数器 T0、T1 的控制	174
7.3	定时/计数器 T0、T1 的工作方式	175
7.4	定时/计数器 T0、T1 的应用举例	179
7.4.1	定时/计数器 T0、T1 的定时应用	179
7.4.2	定时/计数器 T0、T1 的计数应用	181
7.4.3	定时/计数器 T0、T1 的综合应用	182
7.5	定时/计数器 T2	185
7.5.1	定时/计数器 T2 的电路结构	185
7.5.2	定时/计数器 T2 的控制寄存器	186
7.6	定时/计数器 T3、T4	187
7.6.1	定时/计数器 T3、T4 的电路结构	187
7.6.2	定时/计数器 T3、T4 的控制寄存器	187

7.7	可编程时钟输出功能	189
7.7.1	定时/计数器 T0~T4 的可编程时钟输出	189
7.7.2	可编程时钟的应用举例	190
7.7.3	工程训练 7.1 定时/计数器的定时应用	192
7.7.4	工程训练 7.2 定时/计数器的计数应用	194
7.7.5	工程训练 7.3 定时/计数器的综合应用	196
7.7.6	工程训练 7.4 可编程时钟输出	198
	本章小结	200
	习题与思考题	201
第 8 章	STC8A8K64S4A12 单片机中断系统	204
8.1	中断系统概述	204
8.1.1	中断系统的几个概念	204
8.1.2	中断的技术优势	205
8.1.3	中断系统需要解决的问题	206
8.2	STC8A8K64S4A12 单片机中断系统的简介	206
8.2.1	中断请求	206
8.2.2	中断响应	215
8.2.3	中断服务与中断返回	218
8.3	STC8A8K64S4A12 单片机中断系统的中断应用举例	219
8.3.1	定时中断的应用	219
8.3.2	外部中断的应用	223
8.4	STC8A8K64S4A12 单片机外部中断源的扩展	224
8.4.1	工程训练 8.1 定时中断的应用编程	227
8.4.2	工程训练 8.2 外部中断的应用编程	228
	本章小结	230
	习题与思考题	231
第 9 章	STC8A8K64S4A12 单片机的串行接口	234
9.1	串行通信基础	234
9.2	STC8A8K64S4A12 单片机的串行接口 1	237
9.2.1	串行接口 1 的控制寄存器	237
9.2.2	串行接口 1 的工作方式	239
9.2.3	串行接口 1 的波特率	243
9.2.4	串行接口 1 的应用举例	244
9.3	STC8A8K64S4A12 单片机与计算机的通信	256
9.3.1	单片机与计算机 RS-232 串行通信接口设计	256
9.3.2	STC8A8K64S4A12 单片机与计算机的串行通信程序设计	258

9.4	STC8A8K64S4A12 单片机串行接口 1 的中继广播方式	261
9.5	STC8A8K64S4A12 单片机串行接口 2*	261
9.6	STC8A8K64S4A12 单片机串行接口 3*	263
9.7	STC8A8K64S4A12 单片机串行接口 4*	264
9.7.1	工程训练 9.1 STC8A8K64S4A12 单片机间的双机通信	266
9.7.2	工程训练 9.2 STC8A8K64S4A12 单片机与计算机间的串行通信	269
	本章小结	272
	思考与提高题	273
第 10 章	人机对话接口的应用设计	276
10.1	单片机应用系统的设计和开发流程	276
10.1.1	单片机应用系统的设计原则	276
10.1.2	单片机应用系统的开发流程	277
10.1.3	单片机应用系统工程报告的编制	280
10.2	键盘接口与应用编程	283
10.3	LCD 接口与应用编程	291
10.3.1	LCD 模块概述	291
10.3.2	点阵字符型 LCD 模块 LCD1602	291
10.3.3	点阵图形型 LCD 模块 LCD12864	299
10.3.4	工程训练 10.1 STC8A8K64S4A12 单片机与矩阵键盘的接口与应用	309
10.3.5	工程训练 10.2 STC8A8K64S4A12 单片机与 LCD12864 (含中文 字库) 的接口与应用	311
	本章小结	322
	习题与思考题	322
第 11 章	STC8A8K64S4A12 单片机的比较器	326
11.1	比较器的内部结构与控制寄存器	326
11.2	比较器的应用	329
	本章小结	332
	思考与提高题	332
第 12 章	STC8A8K64S4A12 单片机的 A/D 转换模块	334
12.1	A/D 转换模块的结构	334
12.2	A/D 转换模块的控制	335
12.3	A/D 转换模块的应用	338
	本章小结	343
	思考与提高题	344

第 13 章 STC8A8K64S4A12 单片机的 PCA 模块	346
13.1 PCA 模块的结构.....	346
13.2 PCA 模块的控制.....	347
13.3 PCA 模块的工作模式与应用编程.....	350
13.3.1 捕获模式与应用编程.....	350
13.3.2 16 位软件定时器模式与应用编程.....	353
13.3.3 高速脉冲输出模式与应用编程.....	355
13.3.4 脉宽调制模式与应用编程.....	356
13.3.5 工程训练 13.1 PCA 模块的软件定时器应用.....	361
13.3.6 工程训练 13.2 PCA 模块的 PWM 应用.....	364
本章小结.....	366
习题与思考题.....	367
第 14 章 STC8A8K64S4A12 单片机的增强型 PWM 模块	369
14.1 增强型 PWM 模块的结构.....	369
14.2 增强型 PWM 模块的控制.....	370
14.3 增强型 PWM 模块的应用编程.....	375
本章小结.....	382
习题与思考题.....	382
第 15 章 STC8A8K64S4A12 单片机的 SPI 接口	384
15.1 SPI 接口的结构.....	384
15.2 SPI 接口的控制.....	386
15.3 SPI 接口的通信方式.....	391
15.4 SPI 接口的应用编程.....	393
本章小结.....	397
习题与思考题.....	398
第 16 章 STC8A8K64S4A12 单片机的 I²C 的串行总线和 I²C 通信接口	399
16.1 I ² C 串行总线.....	399
16.2 I ² C 通信接口.....	407
本章小结.....	420
思考与提高题.....	420
第 17 章 STC8A8K64S4A12 单片机的低功耗设计与可靠性设计	423
17.1 低功耗设计.....	423
17.2 可靠性设计.....	428
本章小结.....	431
习题.....	431

附录 A	ASCII 码表.....	434
附录 B	STC8A8K64S4A12 系列单片机指令系统表.....	435
附录 C	STC8 系列单片机特殊功能寄存器一览表	439
附录 D	STC8 单片机学习板模块电路	446
附录 E	STC8A8K64S4A12 单片机内部接口硬件切换控制.....	455
附录 F	C51 常用头文件与库函数.....	460
附录 G	C 语言编译常见错误信息一览表	468
附录 H	C51 的模块化编程与 C51 库函数的制作.....	473

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，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相关课程中讲解，但由于数制与编码知识与当前课程的联系并不密切，所以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这方面的知识不太扎实。因此，我们在下文将对相关知识进行梳理。

1.1.1 数制及其转换方法

数制就是计数的方法，通常采用进位计数制，在学习与应用微型计算机的过程中，常用的数制有二进制、十进制和十六进制。在日常生活中采用的是十进制计数方法；微型计算机只能识别和处理数字信息，因此，微型计算机硬件电路采用的是二进制计数方法，但为了更好地记忆与描述微型计算机的地址、程序及运算数字，一般采用十六进制计数方法。

1. 各种数制及其表示方法

二进制、十进制与十六进制的计数规则与表示方法如表 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

数制	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 借一当二	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 借一当十	10	10^i	0、1、2、3、4、5、 6、7、8、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	$()_{10}$ 通常默认表示
十六进制	逢十六进一 借一当十六	16	16^i	0、1、2、3、4、5、 6、7、8、9、A、B、 C、D、E、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	$()_{16}$

注： i 是各进制数码在数字中的位置， i 值以小数点为界，往左依次为 0,1,2,3,⋯，往右依次为 -1,-2,-3,⋯。

2. 数制之间的转换

数值在任意进制之间的相互转换，其整数部分和小数部分必须分开进行。各进制数的相互转换关系如图 1.1 所示。

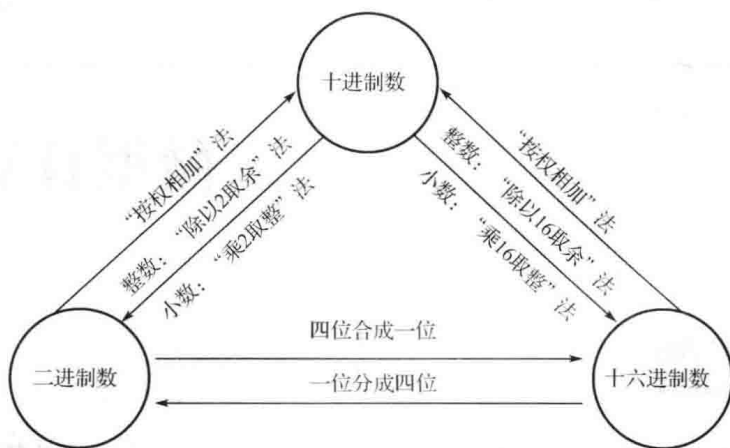


图 1.1 各进制数的相互转换关系

1) 二进制数、十六进制数转换成十进制数

将二进制数、十六进制数按权值展开式展开，所得数相加，即十进制数。

2) 十进制数转换成二进制数

十进制数转换成二进制数要将数值分成整数部分与小数部分进行转换，整数部分和小数部分的转换方法是完全不同的。

(1) 十进制数的整数部分转换成二进制数的整数部分——“除以 2 取余”法，将所得余数倒序排列即可得到二进制数的整数部分，如下所示：

2	84	余数	二进制数码
2	42	0	b ₀
2	21	0	b ₁
2	10	1	b ₂
2	5	0	b ₃
2	2	1	b ₄
2	1	0	b ₅
2	0	1	b ₆

∴ (84)₁₀=(1010100)₂。

(2) 十进制数的小数部分转换成二进制数的小数部分——“乘 2 取整”法，将整数部分顺序排列，即可得到二进制数的小数部分，如下所示：

	整数	0.6875
b ₋₁	1 ←	0.3750
b ₋₂	0 ←	0.7500
b ₋₃	1 ←	0.5000
b ₋₄	1 ←	0.0000

∴ (0.6875)₁₀=(0.1011)₂。

将上述两部分合起来，则有

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

3) 二进制数与十六进制数相互转换

(1) 二进制数转换成十六进制数。

以小数点为界，往左、往右每4位二进制数为一组，每4位二进制数用1位十六进制数表示，往左高位不够用0补齐，往右低位不够用0补齐，如：

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

(2) 十六进制数转换成二进制数。

先将每位十六进制数用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，如：

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

3. 数制转换工具

利用计算机附件中的计算器（科学型）可实现各数制之间的相互转换。单击任务栏中的“开始”按钮，依次单击“所有程序”→“附件”→“计算器”，即可打开“计算器”窗口，单击菜单栏中的“查看”菜单，选择“科学型”，此时计算器界面即科学型计算器界面，如图1.2所示。

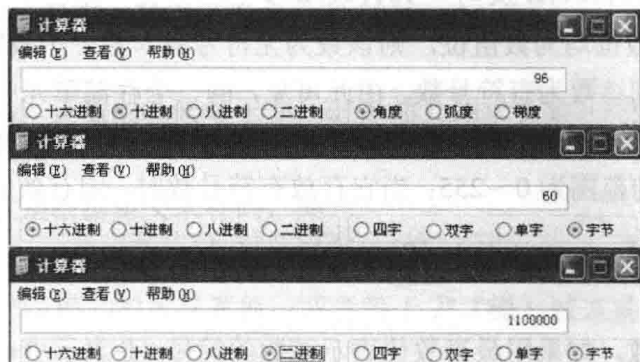


图 1.2 科学型计算器界面

转换方法：先选择被转换数制类型，并在文本框中输入要转换的数字，再选择目标转换数制类型，此时，文本框中的数字就是转换后的数字。例如，将96转换为十六进制数、二进制数的步骤为，先选择数制类型为十进制，再在文本框中输入96，然后选择数制类型为十六进制，此时，文本框中看到的数字即转换后的十六进制数60；再选择数制类型为二进制，此时，文本框中看到的数字即转换后的二进制数1100000，如图1.2所示。

4. 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

数学中数的正和负用符号“+”和“-”表示，计算机中如何表示数的正和负呢？在计算机中数据是存放在存储单元内的。每个存储单元是由若干二进制位组成的，其中每一数位或是 0 或是 1，而数的符号或为“+”或为“-”，因此，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就被数码化了，但从表示形式上看，符号位与数值位没有区别。

设有两个数 x_1, x_2 ：

$$x_1 = +1011011B, \quad x_2 = -1011011B$$

它们在计算机中分别表示为

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

其中，带下画线部分为符号位，字长为 8 位。为了区分这两种形式的数，我们把机器中以数码形式表示的数称为机器数（ $x_1 = \underline{0}1011011B$ 及 $x_2 = \underline{1}1011011B$ ），把原来以一般书写形式表示的数称为真值（ $x_1 = +1011011B$ 及 $x_2 = -1011011B$ ）。

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，其他数位为数值位，则该数为有符号数。由此可见，同一个存储单元中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，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，该数的范围为 -128~+127）。

2. 原码

对于一个二进制数，如果用最高数位表示该数的符号（0 表示“+”，1 表示“-”），其余各数位表示数值本身，则称这种表示方法为原码表示法：

若 $x = \pm x_1 x_2 \cdots x_{n-1}$ ，则 $[x]_{\text{原}} = x_0 x_1 x_2 \cdots x_{n-1}$ 。其中， x_0 为原机器数的符号位，它满足：

$$x_0 = \begin{cases} 0, & x \geq 0 \\ 1, & x < 0 \end{cases}$$

3. 反码

如果 $[x]_{\text{原}} = 0x_1 x_2 \cdots x_{n-1}$ ，则 $[x]_{\text{反}} = [x]_{\text{原}}$ 。

如果 $[x]_{\text{原}} = 1x_1 x_2 \cdots x_{n-1}$ ，则 $[x]_{\text{反}} = \overline{1x_1 x_2 \cdots x_{n-1}}$ 。

也就是说，正数的反码与其原码相同，而负数的反码保持原码的符号位不变，各数值位按位取反。

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，假定现在是北京时间 8 点整，而一只表却指向 10 点整。为了校正此表，可以采用倒拨和顺拨两种方法：倒拨就是

逆时针减少 2 小时, 把倒拨视为减法, 相当于 $10-2=8$, 时针指向 8; 顺拨就是将时针顺时针拨 10 小时, 时针同样指向 8, 把顺拨视为加法, 相当于 $10+10=12$ (自动丢失) $+8=8$, 其中自动丢失的数 (12) 就称为模 (mod), 上述加法称为“按模 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, 就可以将原来的减法 $10-2=8$ 转化为加法 $10+10=12$ (自动丢失) $+8=8 \pmod{12}$ 了。

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 , 则它能表示的无符号数最大值为 2^n-1 , 逢 2^n 进 1 (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 2^n (模) 加上其真值或减去其真值的绝对值。

在补码表示法中, 0 只有一种表示形式, 即 0000...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。下面介绍求负数补码的 3 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

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

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, 这也可以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, 则它加 1 后就变成 0, 并产生向次低位的进位。如果反码的次低位也为 1, 则它同样变成 0, 并产生向其高位的进位 (这相当于在传递进位), 依次类推, 进位一直传递到第 1 个为 0 的位为止, 于是得到这样的转换规律: 从反码的最低位起直到第一个为 0 的位之前 (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的, 所以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。

特别要指出的是, 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, 此时可以认为该结果是以原码形式表示的 (正数的补码即原码); 若结果的符号位为“1”, 则表示结果为负数, 此时可以认为该结果是以补码形式表示的, 若用原码来表示该结果, 还需要对结果求补 (除符