

 普通高等教育计算机规划教材

单片机原理 及应用教程

第 3 版

赵全利 主编

附赠电子教案、习题解答、源程序代码



下载网址 <http://www.cmpedu.com>



机械工业出版社
CHINA MACHINE PRESS

013026007

普通高等教育计算机规划教材

TP368.1-43
45-3

单片机原理及应用教程

第3版

赵全利 主 编
张会敏 刘英杰 忽晓伟 周 伟 等编著
刘瑞新 主 审



机械工业出版社



北航

C1632957

TP368.1-43
45-3

700390810

本书详细介绍了 51 系列及兼容单片机的硬件结构、指令系统及汇编语言程序设计、内部功能及接口、系统外部功能的扩展, Cx51 程序设计, 单片机系统仿真、开发及应用实例, 最后简要介绍了单片机实时操作系统的程序结构及应用。书中通过大量例题和单片机应用实例, 引导读者熟悉和理解单片机基本原理, 逐步掌握单片机应用系统设计开发的基本知识、方法和应用技能。各章后都配有习题, 以巩固学生所学的知识。书中所有程序均经过了上机调试。

本书可作为高等院校电子、控制、通信、机电、测控及信息类专业的教学用书, 也可作为高职高专同类专业的教学用书及各类工程技术人员的自学用书。

本书配套授课电子课件, 提供习题解答、源程序代码及 Proteus 仿真电路文件。需要的教师可登录 www.cmpedu.com 免费注册, 经审核通过后可直接下载, 或联系编辑索取 (QQ: 2399929378, 电话: 010-88379750)。

图书在版编目 (CIP) 数据

单片机原理及应用教程/赵全利主编. —3 版. —北京: 机械工业出版社, 2012. 12

普通高等教育计算机规划教材

ISBN 978-7-111-40995-3

I. ①单… II. ①赵… III. ①单片微型计算机-高等学校-教材
IV. ①TP368.1

中国版本图书馆 CIP 数据核字 (2012) 第 315017 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

责任编辑: 郝建伟 范文环

责任印制: 张楠

北京圣夫亚美印刷有限公司印刷

2013 年 3 月第 3 版·第 1 次印刷

184mm×260mm·19.25 印张·476 千字

0001—3000 册

标准书号: ISBN 978-7-111-40995-3

定价: 39.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

电话服务

网络服务

社服务中心: (010)88361066 教材网: <http://www.cmpedu.com>

销售一部: (010)68326294 机工官网: <http://www.cmpbook.com>

销售二部: (010)88379649 机工官博: <http://weibo.com/cmp1952>

读者购书热线: (010)88379203 封面无防伪标均为盗版

出版说明

信息技术是当今世界上发展最快、应用最广的关键技术之一，是推动经济增长和知识传播的重要引擎。在我国，随着国家信息化发展战略的贯彻实施，信息化建设已进入了全方位、多层次推进应用的新阶段。掌握计算机技术已成为 21 世纪人才应具备的基础素质之一。

为了推动计算机技术的发展，满足计算机学科教育的需求，机械工业出版社聘请了全国多所高等院校的一线教师，进行了充分的调研和讨论，针对计算机相关课程的特点，总结教学中的实践经验，组织出版了这套“普通高等教育计算机规划教材”。

本套教材具有以下特点：

- 1) 反映计算机技术领域的新发展和新应用。
- 2) 为了体现建设“立体化”精品教材的宗旨，本套教材为主干课程配备了电子教案、学习与上机指导、习题解答、多媒体光盘、课程设计和毕业设计指导等内容。
- 3) 针对多数学生的学习特点，采用通俗易懂的方法讲解知识，逻辑性强、层次分明、叙述准确而精炼、图文并茂，使学生可以快速掌握，并学以致用。
- 4) 符合高等院校各专业人才的培养目标及课程体系的设置，注重培养学生的应用能力，强调知识、能力与素质的综合训练。
- 5) 注重教材的实用性、通用性，适合各类高等院校、高等职业学校及相关院校的教学，也可作为各类培训班教材和自学用书。

希望计算机教育界的专家和老师能提出宝贵的意见和建议。衷心感谢计算机教育工作者和广大读者的支持与帮助！

机械工业出版社

前 言

为适应当前学习单片机的需要,作者从单片机应用开发的角度出发,在本书第2版的基础上进行了修改、调整和扩充,更加细化、完善地描述了单片机的重要知识点及程序代码,增设了单片机仿真、调试应用技术及大量的应用实例,凸显了51系列兼容机C语言编程的关键技术、ISP技术及单片机应用系统的开发过程。简要介绍了RTX51实时操作系统,以更加适应当前高等学校各相关专业及高职高专的教学需要。实践证明,随着计算机技术的深入发展及单片机性能、容量、功能的不断提高和扩充,以51系列8位兼容机组成的单片机应用系统仍能满足绝大多数应用领域的功能需求。51系列8位兼容机以其通用性强、价格低廉、设计开发方便灵活等特点而遍及各个领域。可以肯定,以51系列兼容机为主的8位单片机,在现在及以后的相当一段时期内仍将占据单片机应用的主导地位,有着广泛和稳定增长的市场。

本书以51系列兼容机为主流芯片、以单片机应用为主要目的,结合高等教育各相关专业的特点,系统介绍了单片机的结构原理及应用系统的组成与设计方法。为便于读者掌握本书的内容,各章均配有大量的例题与习题。本书实例取材于工程实际,内容覆盖面广,具有很高的参考价值。本书本着一切内容为应用服务的宗旨,在介绍单片机基本知识、基本结构和基本组成的基础上,通过大量例题和单片机应用实例,引导读者熟悉和理解单片机的基本原理,逐步掌握单片机应用系统设计开发的基本知识、方法和应用技能。本书是根据中国计算机学会高等教育学会最新审定的编写大纲编写的,所有例题和源程序都经上机调试通过,授课时间为60课时。本书可作为高等院校电子、自动化、通信、机电、测控及信息类专业“单片机原理与应用”课程的教学用书,也可作为高职高专同类专业的教学用书及各类工程技术人员自学用书。本书中带*的章节为选修内容。

为了配合本书的教学,机械工业出版社提供了电子教案、习题解答、源程序代码及仿真电路文件。读者可在机械工业出版社网站(www.cmpedu.com)下载。

本书由赵全利主编,张会敏、刘英杰、忽晓伟、周伟等编著。第1章、第5章的5.2至5.3节由赵全利编写;第9章由张会敏编写;第2章的2.2至2.3节、第8章由杜海龙编写;第2章的2.4至2.6节、第3章的3.1节、第11章及附录由刘英杰编写;第3章的3.2至3.6节由忽晓伟编写;第4章、第7章由谢泽奇编写;第5章的5.4至5.5节、第6章由周伟编写;第10章由赵俊伟编写;第2章的2.1节、第5章的5.1节及本书各章习题解答、电子课件、仿真电路图设计及软硬件调试验证、文字录入等工作由樊永良、张杰、陈瑞霞、周晓平、李会萍、吴永辉、宫德龙、刘克纯、骆秋容、缪丽丽、刘大学、陈文明、万兆君、孙明建、范龙、徐云林完成。全书由赵全利统稿,刘瑞新教授主审。本书编写过程中,得到了项仕标教授的指导,并提出了许多建设性意见,在此表示感谢。本书中涉及的元器件符号由于软件原因,部分保留了原有符号,以便于和软件相统一。如果需要,请读者参阅图象相关标准文献。

由于编者水平有限,书中难免有错误与不妥之处,请广大读者批评指正。

编 者

教 学 建 议

教 学 内 容	教 学 要 求	课 时
第 1 章 计算机基础知识概述	了解计算机常用数制及计算机系统组成特点；了解单片机的特点、应用及其发展概况；熟悉单片机应用系统的组成；掌握二进制与其他数制之间转换及编码	2
第 2 章 MCS-51 单片机 硬件结构	掌握单片机的基本结构、存储结构及各类存储器的地址空间；明晰特殊寄存器的功能和用法；熟练掌握单片机端口引脚功能和第二功能；了解 MCS-51 单片机工作方式；了解 MCS-51 单片机的时序；熟悉单片机最小系统结构	8（包括 2 个实验课时）
第 3 章 MCS-51 指令系统及 汇编语言程序设计	熟悉单片机指令系统；熟练掌握 7 种寻址方式、数据传送与交换指令、算术运算和逻辑运算指令、控制转移指令、位操作指令；理解伪指令的意义，掌握汇编语言程序基本结构；掌握顺序程序、分支程序、循环程序、位操作程序及子程序设计的思想和方法；熟练掌握单片机程序开发软件的使用方法和上机操作步骤	12（包括 4 个实验课时）
第 4 章 MCS-51 单片机的 中断系统	理解中断、中断源、中断优先级及中断嵌套概念；掌握 MCS-51 中断系统结构、中断相关控制字及初始化编程；熟悉外部中断源的扩展方法；了解单片机中断响应过程；熟练掌握应用中中断实现单片机一般控制功能的软、硬件设计	6（包括 2 个实验课时）
第 5 章 MCS-51 单片机定时 /计数器	熟悉定时/计数器结构和工作原理；掌握通过控制寄存器设置定时/计数器工作方式的方法和初始化编程；熟练掌握定时/计数器各种工作方式下的应用编程；能够结合中断使用定时/计数器实现单片机一般控制功能	6（包括 2 个实验课时）
第 6 章 单片机串行口及应用	了解串行通信的基本概念；掌握 MCS-51 单片机串行口结构、控制方法、串行口的工作方式及波特率设置；能够通过相应的控制寄存器来设置通信方式、结合中断和 I/O 接口编程实现与外部设备的串行通信；熟悉常用串行通信总线标准，了解其接口电路原理	6（包括 2 个实验课时）
第 7 章 单片机的系统扩展	理解单片机片外系统总线结构组成，掌握利用系统总线扩展存储器的方法；了解 EEPROM、FLASH、SRAM 等存储器的使用方法；熟练掌握并行单片机 I/O 端口扩展、地址空间编址、可编程并行接口芯片的使用方法及初始化编程	6（包括 2 个实验课时）
第 8 章 单片机 I/O 接口技术	了解显示器及接口电路、键盘及接口电路的原理；掌握常用显示器及接口电路、键盘及接口电路的软、硬件设计思想和实现方法；能够编写一般 I/O 接口控制程序；熟悉 A/D、D/A 技术的原理和常用芯片的使用方法；掌握 A/D、D/A 转换器与单片机的接口电路及编程	10（包括 4 个实验课时）
第 9 章 Cx51 程序设计及 Proteus 仿真	熟悉 Cx51 开发环境的使用方法；熟练掌握通过 C 语言编程实现对单片机的资源的访问；掌握 C 程序的编程方法及库函数的使用方法；熟悉 Proteus 软件仿真的使用方法，能够正确使用该软件对单片机应用系统进行仿真调试	8（包括 4 个实验课时）
第 10 章 单片机应用实例与 系统开发	了解单片机开发系统的基本功能；熟悉单片机系统的开发应用过程；能够使用单片机开发系统开发一般单片机应用系统	4（包括 2 个实验课时）
第 11 章	选修	2
复习		2
总学时		72

说明：

- 1) 建议选择相关教学内容在单片机仿真环境（上位机）或开发系统中进行，采用“课堂教学与实际操作”、“应用实例与开发系统”相结合的教学模式。
- 2) 建议相关章节的课外作业，要求学生通过软件环境完成硬件电路绘制、编程及仿真调试。
- 3) 任课教师可以根据实际情况有选择地对部分章节的内容酌情增减或进行调整。
- 4) 本课程建议学时为 72 学时（包括实验 24 学时），任课教师可根据实际情况调整理论课与实验课的比例。

目 录

出版说明

前言

第1章 计算机基础知识概述 1

1.1 数制与编码 1

1.1.1 计算机与二进制 1

1.1.2 数制及其转换 1

1.1.3 编码 5

1.2 计算机系统的组成 9

1.2.1 计算机的硬件组成 9

1.2.2 计算机的软件系统 11

1.2.3 计算机语言及程序设计 11

1.3 单片机与嵌入式系统的组成 15

1.3.1 单片机及其发展概况 15

1.3.2 单片机的特点和应用 15

1.3.3 嵌入式系统 17

1.3.4 单片机应用系统的组成 18

1.4 一个简单的单片机应用实例 19

1.5 习题 21

第2章 MCS-51 单片机硬件结构 22

2.1 MCS-51 单片机总体结构 22

2.1.1 MCS-51 单片机总体结构框图及
功能 22

2.1.2 MCS-51 引脚功能 26

2.2 MCS-51 存储结构及位处理器 29

2.2.1 MCS-51 存储器的特点 29

2.2.2 程序存储器 30

2.2.3 数据存储器 31

2.2.4 专用寄存器 34

2.2.5 位处理器 35

2.3 MCS-51 的工作方式 36

2.3.1 复位方式 36

2.3.2 程序执行方式 37

2.3.3 节电工作方式 37

2.4 MCS-51 单片机的时序 39

2.4.1 时钟 39

2.4.2 CPU 时序 39

2.5 单片机最小系统 42

2.6 习题 43

第3章 MCS-51 指令系统及汇编语言

程序设计 45

3.1 指令系统简介 45

3.1.1 指令格式 45

3.1.2 指令的分类及符号说明 46

3.2 寻址方式 47

3.3 指令系统 51

3.3.1 数据传送类指令 51

3.3.2 算术运算类指令 58

3.3.3 逻辑运算类指令 62

3.3.4 控制转移类指令 64

3.3.5 位操作类指令 70

3.4 汇编语言程序设计基础 73

3.4.1 汇编语言程序的组成 74

3.4.2 伪指令 74

3.4.3 程序设计步骤及技术 76

3.5 程序设计实例 77

3.5.1 汇编语言基本程序设计 77

3.5.2 延时程序设计 81

3.5.3 代码转换程序设计 82

3.5.4 查表程序设计 83

3.5.5 运算程序设计 83

3.5.6 排序程序设计 84

3.5.7 输入/输出程序设计 85

3.5.8 数字滤波程序设计 86

3.6 习题 87

第4章 MCS-51 单片机的中断系统 89

4.1 中断的概念 89

4.1.1 中断及中断源 89

4.1.2 中断嵌套 89

4.2 MCS-51 中断系统的结构及中断控制	90	6.1.4 奇偶校验	118
4.2.1 中断源和中断请求标志	90	6.2 MCS-51 单片机串行口	118
4.2.2 中断允许控制	92	6.2.1 串行口的结构	119
4.2.3 中断优先级控制	92	6.2.2 串行口控制	119
4.3 MCS-51 中断响应的过程	93	6.2.3 串行口的工作方式	120
4.3.1 中断响应	93	6.2.4 波特率的设置	122
4.3.2 中断处理	94	6.3 串行口的应用	124
4.3.3 中断返回	94	6.3.1 串行口方式 0 的应用	124
4.4 中断响应的时间	94	6.3.2 串行口在其他方式下的应用	127
4.5 中断响应后中断请求的撤除	95	6.3.3 双机通信	129
4.6 中断系统应用实例	95	6.3.4 多机通信	132
4.6.1 单步操作的中断实现	95	6.4 常用串行通信总线标准及接口电路	139
4.6.2 外部中断源的扩展	96	6.4.1 RS-232C 总线标准及接口电路	139
4.7 中断程序举例	97	6.4.2 RS-422/485 总线标准及接口电路	140
4.8 习题	101	6.5 习题	142
第 5 章 MCS-51 单片机定时/计数器	102	第 7 章 单片机的系统扩展	143
5.1 定时/计数器概述	102	7.1 单片机系统扩展概述	143
5.2 定时/计数器的控制	103	7.2 常用程序存储器及其扩展	144
5.2.1 定时/计数器的工作模式寄存器 TMOD	103	7.2.1 常用程序存储器	144
5.2.2 定时器控制寄存器 TCON	104	7.2.2 程序存储器的扩展	148
5.3 定时/计数器的工作模式	104	7.3 常用数据存储器及其扩展	151
5.3.1 工作模式 0	104	7.3.1 常用的数据存储器	151
5.3.2 工作模式 1	105	7.3.2 数据存储器的扩展	152
5.3.3 工作模式 2	105	7.4 I/O 口的扩展	154
5.3.4 工作模式 3	106	7.4.1 简单并行 I/O 口的扩展	154
5.4 定时/计数器应用实例	107	7.4.2 8155 可编程多功能接口的扩展	157
5.4.1 工作模式 0 的应用	107	7.5 单片机扩展系统外部地址空间的编址方法	163
5.4.2 工作模式 1 的应用	108	7.5.1 线选法	163
5.4.3 工作模式 2 的应用	108	7.5.2 译码法	164
5.4.4 工作模式 3 的应用	109	7.6 习题	165
5.4.5 定时/计数器的综合应用	110	第 8 章 单片机的 I/O 接口技术	167
5.5 习题	113	8.1 键盘及接口电路	167
第 6 章 单片机串行口及应用	115	8.1.1 键盘的分类	167
6.1 串行通信的基本概念	115	8.1.2 键盘的工作原理	167
6.1.1 异步通信和同步通信	115	8.1.3 键盘结构及扫描子程序	168
6.1.2 串行通信的制式	117		
6.1.3 波特率和发送/接收时钟	117		

8.1.4 键盘接口扩展设计	170	10.1.1 光电计数器	237
8.2 显示器及接口电路	172	10.1.2 时间顺序控制器	239
8.2.1 LED 状态显示	172	10.1.3 转速表	243
8.2.2 LED 数码显示	174	10.1.4 交通灯管理系统	246
8.2.3 七段 LED 数码管显示接口	176	10.2 单片机系统的开发应用过程 ...	250
8.2.4 键盘、显示器综合设计	177	10.2.1 总体论证	252
8.3 A-D、D-A 转换器与单片机的 接口	180	10.2.2 总体设计	252
8.3.1 D-A 转换器	181	10.2.3 硬件设计	252
8.3.2 A-D 转换器	184	10.2.4 软件设计	252
8.4 应用举例——简易 5 V 直流数字 电压表	191	10.2.5 联机调试	253
8.5 习题	193	10.2.6 下载测试	254
第 9 章 Cx51 程序设计及 Proteus 仿真	194	10.2.7 脱机运行	255
9.1 Keil C 简介与环境设置	194	10.3 单片机开发系统	255
9.2 Cx51 简介及特征	195	10.3.1 开发系统的基本功能	255
9.2.1 Cx51 的扩展	195	10.3.2 开发系统简介	256
9.2.2 存储区	196	10.4 单片机应用系统开发实例	258
9.2.3 存储模式	197	10.4.1 电动机调速系统	258
9.2.4 数据类型与自定义变量	197	10.4.2 实验及教学综合楼智能化管理 系统	268
9.2.5 指针	199	10.5 习题	271
9.2.6 函数	201	第 11 章* RTX51 实时操作系统简介 ...	272
9.3 Cx51 开发工具简介	205	11.1 RTX51 系统简介	272
9.4 Cx51 应用程序设计举例	209	11.2 RTX51 的程序结构	273
9.4.1 输入	211	11.3 RTX51 的功能函数	274
9.4.2 输出	213	11.4 RTX51 的任务调度与系统 配置	277
9.4.3 定时/计数器	214	11.4.1 RTX51 的任务调度	277
9.4.4 外部中断	216	11.4.2 RTX51 的系统配置	278
9.4.5 串行口通信	218	11.5 RTX51 的应用举例	279
9.5 Proteus 软件仿真	220	11.6 习题	281
9.5.1 Proteus 软件使用简介	220	附录	282
9.5.2 Keil C 与 Proteus 连接调试	223	附录 A MCS-51 指令表	282
9.5.3 Proteus 仿真举例	225	附录 B ASCII (美国标准信息交换码) 码表	285
9.6 习题	236	附录 C 典型汇编及模拟运行程序的 使用	286
第 10 章 单片机应用实例与系统 开发	237	附录 D 常用 C51 库函数	297
10.1 单片机应用系统实例	237	参考文献	300

第1章 计算机基础知识概述

1946年2月14日，在美国宾夕法尼亚（Pennsylvania）大学的一间大厅里，由美国陆军的一位将军按下一个按钮，一件对现代世界有巨大影响的事件发生了，世界上第一台电子数字计算机（取名为ENIAC）启动了。由于ENIAC是按照美籍匈牙利科学家冯·诺依曼（Von Neumann）提出的“以二进制存储信息”、“以存储程序”为基础的结构思想进行设计、制造和工作的，所以人们又称其为冯·诺依曼计算机。

六十多年来，计算机的应用已由传统的科学计算发展到用于信息处理、实时控制、辅助设计、智能模拟及现代通信网络等领域。计算机技术的迅速发展对人类社会进步产生了巨大的推动作用，尤其是微型计算机的出现及其在国民经济和人民生活的各个领域不断深入的广泛应用，正在改变着人们传统的生活和工作方式，人类已进入以计算机为主要工具的信息时代。

本章主要讲述计算机的基础知识，包括：常用的数制及不同数制相互间的转换、编码；计算机系统组成、接口、总线及程序设计的基本知识；单片机的概念、特点、应用领域及其嵌入式系统的概念。最后通过一个简单的单片机应用实例使读者初步了解单片机的工作概况。

1.1 数制与编码

在计算机中，任何命令和信息都是以二进制数据的形式存储的。计算机所执行的全部操作都归结为对数据的处理和加工，为了便于理解计算机系统的基本工作原理，掌握数字、字母等字符在计算机系统内的表示方法及处理过程，本节主要介绍有关计算机、数制和编码等方面的基础知识。

1.1.1 计算机与二进制

计算机是一种以二进制数据形式存储信息、以程序存储为基础、由程序自动控制的电子设备。在计算机中，由于所采用的电子逻辑器件仅能存储和识别两种状态的特点，计算机内部一切信息存储、处理和传输均采用二进制数的形式。二进制数是计算机硬件能直接识别并进行处理的唯一形式。

人们需要计算机所做的任何工作，都必须以计算机所能识别的指令形式转换为二进制数据送入计算机内存中，一条条有序指令的集合称为程序。计算机的工作过程也就是执行程序的过程，计算机所做的任何工作都是执行程序的结果。可以说，二进制数据存储信息和程序存储是计算机的基本工作机制。

1.1.2 数制及其转换

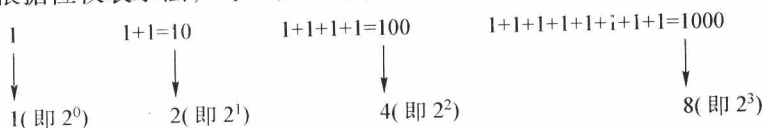
数制就是计数方式。

日常生活中常用的是十进制计数方式，计算机内部使用的是二进制数据，在向计算机输入数据及输出数据时，人们惯于用十进制、十六进制数据等，因此，计算机在处理数据时，必须进行各种数制之间的相互转换。

1. 二进制数

二进制数只有两个数字符号：0 和 1。计数时按“逢二进一”的原则进行计数。也称其基数为二。一般情况下，二进制数可表示为 $(110)_2$ 、 $(110.11)_2$ 、10110B 等。

根据位权表示法，每一位二进制数在其不同位置表示不同的值。例如：



对于 8 位二进制数（由低位 ~ 高位分别用 D0 ~ D7 表示），其各位所对应的权值为

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
D7	D6	D5	D4	D3	D2	D1	D0

对于任何二进制数，可按位权展开求和得到与之相应的十进制数，则有

$$(10)_2 = 1 \times 2^1 + 0 \times 2^0 = (2)_{10}$$

$$(11)_2 = 1 \times 2^1 + 1 \times 2^0 = (3)_{10}$$

$$(110)_2 = 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = (6)_{10}$$

$$(111)_2 = 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = (7)_{10}$$

$$(1111)_2 = 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = (15)_{10}$$

$$(10110)_2 = 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = (22)_{10}$$

例如，二进制数 10110111，按位权展开求和计算可得

$$\begin{aligned} (10110111)_2 &= 1 \times 2^7 + 0 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \\ &= 128 + 0 + 32 + 16 + 0 + 4 + 2 + 1 \\ &= (183)_{10} \end{aligned}$$

对于含有小数的二进制数，从小数点右边第一位小数开始向右各位的权值分别为

2^{-1}	2^{-2}	2^{-3}	2^{-4}	...
□	□	□	□	□

例如，二进制数 10110.101，按位权展开求和计算可得

$$\begin{aligned} (10110.101)_2 &= 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} \\ &= 16 + 4 + 2 + 0.5 + 0.125 \\ &= (22.625)_{10} \end{aligned}$$

必须指出：在计算机中，一个二进制数（如 8 位、16 位或 32 位）既可以表示数值，也可以表示一种符号的代码，还可以表示某种操作（即指令），计算机在程序运行时按程序的规则自动识别，这就是本节开始所讲的一切信息都是以二进制数据进行存储的。

2. 十六进制数

十六进制数是学习和研究计算机中二进制数一种比较方便的工具。计算机在信息输入/输出及书写相应程序或数据时，可采用简短的十六进制数表示相应的位数较长的二进制数。

十六进制数有 16 个数字符号，其中 0 ~ 9 与十进制数相同，剩余 6 个为 A ~ F，分别表

示十进制数的 10~15, 见表 1-1。十六进制数的计数原则是“逢十六进一”, 也称其基数为十六, 整数部分各位的权值由低位到高位分别为 16^0 、 16^1 、 16^2 、 16^3 ……。

$$\text{例如: } (31)_{16} = 3 \times 16^1 + 1 \times 16^0 = (49)_{10}$$

$$(2AF)_{16} = 2 \times 16^2 + 10 \times 16^1 + 15 \times 16^0 = (687)_{10}$$

为了便于区别不同进制的数据, 一般情况下可在数据后面跟一后缀:

二进制数用“B”作后缀, 如 00111010B。

十六进制数用“H”作后缀, 如 3A5H。

十进制数用“D”作后缀, 如 39D 或 39。

3. 不同数制之间的转换

前已述及, 计算机中的数只能用二进制表示, 十六进制数适合读写方便的需要, 日常生活中使用的是十进制数, 因此, 计算机必须根据需要对各种进制的数据进行转换。

(1) 二进制数转换为十进制数

任意二进制数均可按权值展开, 将其转化为十进制数。

$$\text{例如: } 10111\text{B} = 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 23\text{D}$$

$$10111.011\text{B} = 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3} \\ = 23.375\text{D}$$

(2) 十进制数转换为二进制数

1) 方法 1: 十进制数转换为二进制数, 可将整数部分和小数部分分别进行转换, 然后合并。其中整数部分可采用“除 2 取余法”进行转换。小数部分可采用“乘 2 取整法”进行转换。

例如: 采用“除 2 取余法”将 37D 转换为二进制数。

2 37	余数	低位 (第一次余数必为低位)
2 18	1	↓
2 9	0	
2 4	1	
2 2	0	
2 1	0	
0	1	
		高位 (直到商数等于 0 为止)

把所得余数由高到低排列起来可得:

$$37 = 100101\text{B}$$

例如: 采用“乘 2 取整法”将 0.625 转换为二进制小数。

0.625		
× 2		
1.250 ---- 取整数 1	高位 (第一次整数 1 必为二进制数小数权值的最高位)	↓
× 2		
0.500 ---- 取整数 0		
× 2		
1.000 ---- 取整数 1	低位	

把所得整数由高到低排列起来可得:

$$0.625 = 0.101\text{B}$$

同理，把 37.625 转换为二进制数，只需将以上转换合并起来便可得到

$$37.625 = 100101.101\text{B}$$

2) 方法 2：可将十进制数与二进制数的位权从高位到低位比较，若十进制数大于或等于二进制某位，则该位取“1”，否则该位取“0”，采用按位分割法进行转换。

例如：将 37.625 转换为二进制数。

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1
0	0	1	0	0	1	0	1

将整数部分 37 与二进制各位权值从高位→低位比较， $37 > 32$ ，则该位取 1，剩余 $37 - 32 = 5$ ，逐位比较，得 00100101B。

将小数部分 0.625 按同样方法，得 0.101B。

结果为 $37.625\text{D} = 100101.101\text{B}$ 。

(3) 二进制数与十六进制数的相互转换

在计算机进行输入、输出时，常采用十六进制数。十六进制数可看做是二进制数的简化表示。

因为 $2^4 = 16$ ，所以 4 位二进制数相当于 1 位十六进制数，二进制、十进制、十六进制对应数的转换关系见表 1-1。

表 1-1 二进制、十进制、十六进制数转换表

十 进 制	二 进 制	十 六 进 制
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

在将二进制数转换为十六进制数时，其整数部分可由小数点开始向左每 4 位为一组进行分组，直至高位。若高位不足 4 位，则补 0 使其成为 4 位二进制数，然后按表 1-1 的对应关

系进行转换。其小数部分由小数点开始向右每 4 位为一组进行分组，不足 4 位则末位补 0 使其成为 4 位二进制数，然后按表 1-1 的对应关系进行转换。

例如：1000101B = 0100 0101B = 45H

10001010B = 1000 1010B = 8AH

100101. 101B = 0010 0101. 1010B = 25. AH

需要将十六进制数转换为二进制数时，则为上述方法的逆过程。

例如：45. AH = 0100 0101. 1010 B

7ABFH = $\underbrace{0111}_7 \underbrace{1010}_A \underbrace{1011}_B \underbrace{1111}_F$

即 7ABFH = 11110101011111B

(4) 十进制数与十六进制数的相互转换

十进制数与十六进制数的相互转换可直接进行，也可先转换为二进制数，然后再把二进制数转换为十六进制数或十进制数。

例如：将十进制数 37D 转换为十六进制数。

37D = 100101B = 00100101B = 25H

例如：将十六进制数 41H 转换为十进制数。

41H = 01000001B = 65D

也可按位权展开求和的方式将十六进制数直接转换为十进制数，这里不再详述。

1.1.3 编码

计算机通过输入设备（如键盘）输入的信息和通过输出设备输出的信息，既有数字（数值型数据）、也有字符、字母、各种控制符号及汉字（非数值型数据）等。计算机内部所有数据均用二进制代码的形式表示，前面所提到的二进制数，没有涉及正、负符号问题，实际上是一种无符号数的表示，在实际问题中，有些数据是有正、负之分的。为此，需要对常用的数据及符号等进行编码，以表示不同形式的信息。这种以编码形式所表示的信息既便于存储，也便于由输入设备输入信息、输出设备输出相应的信息。

1. 二进制数的编码

(1) 机器数与真值

一个数在计算机中的表示形式叫做机器数，而这个数本身（含符号“+”或“-”）称为机器数的真值。

通常在机器数中，用最高位“1”表示负数，“0”表示正数（以下均以 8 位二进制数为例）。

例如，设两个数为 N1、N2，其真值为

N1 = 105 = +01101001B

N2 = -105 = -01101001B

则对应的机器数为

N1 = 0 1101001B（最高位“0”表示正数）

N2 = 1 1101001B（最高位“1”表示负数）

必须指出，对于一个有符号数，可因其编码不同而有不同的机器数表示法，如下面将要介绍的原码、反码和补码。

(2) 原码、反码和补码

1) 原码。按上所述, 正数的符号位用“0”表示, 负数的符号位用“1”表示, 其数值部分随后表示, 称为原码。

例如, 仍以上面 N1、N2 为例, 则

$$[N1]_{\text{原}} = 0\ 1101001B$$

$$[N2]_{\text{原}} = 1\ 1101001B$$

原码表示方法简单, 便于与真值进行转换。但在进行减法时, 为了把减法运算转换为加法运算 (计算机结构决定了加法运算), 必须引进反码和补码。

2) 反码、补码。在计算机中, 任何有符号数都是以补码形式存储的。对于正数, 其反码、补码与原码相同。

例如, $N1 = +105$

$$\text{则 } [N1]_{\text{原}} = [N1]_{\text{补}} = [N1]_{\text{反}} = 0\ 1101001B$$

对于负数, 其反码为: 原码的符号位不变, 其数值部分按位取反。

例如, $N2 = -105$

$$\text{则 } [N2]_{\text{原}} = 1\ 1101001B$$

$$[N2]_{\text{反}} = 1\ 0010110B$$

负数的补码为: 原码的符号位不变, 其数值部分按位取反后再加 1 (即负数的反码 + 1), 称为求补。

例如, $N2 = -105$

$$\begin{aligned}\text{则 } [N2]_{\text{补}} &= [N2]_{\text{反}} + 1 \\ &= 1\ 0010110B + 1 = 1\ 0010111B\end{aligned}$$

如果已知一个负数的补码, 可以对该补码再进行求补码 (即一个数的补码的补码), 即可得到该数的原码, 即 $[[X]_{\text{补}}]_{\text{补}} = [X]_{\text{原}}$, 而求出真值。

例如, 已知 $[N2]_{\text{补}} = 1\ 0010111B$

$$[N2]_{\text{原}} = 11101000B + 1 = 11101001B$$

由此可得真值: $N2 = -105$

对采用补码形式表示的数据进行运算时, 可以将减法转换为加法。

例如, 设 $X = 10$, $Y = 20$, 求 $X - Y$ 。

$X - Y$ 可表示为 $X + (-Y)$, 即 $10 + (-20)$ 。

$$[X]_{\text{原}} = [X]_{\text{反}} = [X]_{\text{补}} = 00001010B$$

$$[-Y]_{\text{原}} = 10010100B$$

$$[-Y]_{\text{补}} = [-Y]_{\text{反}} + 1 = 11101011B + 1 = 11101100B$$

$$\begin{aligned}\text{则有 } [X + (-Y)]_{\text{补}} &= [X]_{\text{补}} + [-Y]_{\text{补}} \\ &= 00001010B + 11101100B (\text{按二进制相加}) \\ &= 11110110B (\text{和的补码})\end{aligned}$$

再对 $[X + (-Y)]_{\text{补}}$ 求补码可得 $[X + (-Y)]_{\text{原}}$, 即

$$[X + (-Y)]_{\text{原}} = 10001001B + 1 = 10001010B$$

则 $X - Y$ 的真值为 $-10D$ 。

必须指出: 所有负数在计算机中都是以补码形式存放的。对于 8 位二进制数, 作为补码

形式，它所表示的范围为 $-128 \sim +127$ ；而作为无符号数，它所表示的范围为 $0 \sim 255$ 。对于 16 位二进制数，作为补码形式，它所表示的范围为 $-32\,768 \sim +32\,767$ ；而作为无符号数，它所表示的范围为 $0 \sim 65\,535$ 。因而，计算机中存储的任何一个数据，由于解释形式的不同，所代表的意义也不同，计算机在执行程序时自动地进行识别。

例如，某计算机存储单元的数据为 84H，其对应的二进制数表现形式为 10000100B，该数若解释为无符号数编码，其真值为 $128 + 4 = 132$ ；该数若解释为有符号数编码，最高位为 1 可确定为负数的补码表示，则该数的原码为 $11111011\text{B} + 1\text{B} = 11111100\text{B}$ ，其真值为 -124 ；该数若解释为 BCD 编码，其真值为 84D（下面介绍）；若该数作为 8051 单片机指令，则表示一条除法操作（见附录 A）。

2. 二-十进制编码

二-十进制编码又称为 BCD 编码。BCD 编码既具有二进制数的形式，以便于存储，又具有十进制数的特点，以便于进行运算和显示结果。在 BCD 码中，用 4 位二进制代码表示 1 位十进制数。常用的 8421BCD 码的对应编码见表 1-2。

表 1-2 二-十进制编码 (8421BCD 码)

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

例如，将 27 转换为 8421BCD 码：

$27\text{D} = (0010\ 0111)_{8421\text{BCD 码}}$

将 105 转换为 8421BCD 码：

$105\text{D} = (0001\ 0000\ 0101)_{8421\text{BCD 码}}$

因为 8421BCD 码中只能表示 0000B ~ 1001B (0 ~ 9) 这 10 个代码，不允许出现代码 1010B ~ 1111B（因其值大于 9），因而，计算机在进行 BCD 码加法（即二进制加法）的过程中，若和的低四位大于 9（即 1001B）或低四位向高四位有进位时，为保证运算结果的正确性，低四位必须进行加 6 修正。同理，若和的高四位大于 9（即 1001B）或高四位向更高四位有进位时，为保证运算结果的正确性，高四位也必须进行加 6 修正。

例如， $17 = (0001\ 0111)_{8421\text{BCD}}$

$24 = (0010\ 0100)_{8421\text{BCD}}$

$17 + 24 = 41$ 在计算机中的操作为

$$\begin{array}{r}
 0001\ 0111\text{B} \\
 +\ 0010\ 0100\text{B} \\
 \hline
 0011\ 1011\text{B} \leftarrow \text{个位超过 9, 结果错误。} \\
 +\ 0000\ 0110\text{B} \leftarrow \text{进行加 6 修正} \\
 \hline
 0100\ 0001\text{B} \leftarrow (01000001)_{8421\text{BCD}}=41\text{D}, \text{结果正确。}
 \end{array}$$

3. ASCII 码

以上介绍的是计算机中的数值型数据的编码，对于计算机中非数值型数据，如十进制数字符号：“0”，“1”，…，“9”（不是指数值）；

26 个大小写英文字母；

键盘专用符号：“#”、“\$”、“&”、“+”、“=”；

键盘控制符号：“CR”（回车）、“DEL”等。

上述这些符号在由键盘输入时不能直接装入计算机，必须将其转换为特定的二进制代码（即将其编码），以二进制代码所表示的字符数据的形式装入计算机。

ASCII（American Standard Code for Information Interchange）码是一种国际标准信息交换码。它利用 7 位二进制代码来表示字符，再加上 1 位校验位，故在计算机中用 1 个字节即 8 位二进制数来表示一个字符，这样有利于对这些数据进行处理及传输。常用字符的 ASCII 码表示见表 1-3。

表 1-3 常用字符的 ASCII 码

字 符	ASCII 码
0	00110000B (30H)
1	00110001B (31H)
2	00110010B (32H)
⋮	⋮
9	00111001B (39H)
A	01000001B (41H)
B	01000010B (42H)
C	01000011B (43H)
⋮	⋮
a	01100001B (61H)
b	01100010B (62H)
c	01100011B (63H)
⋮	⋮
CR（回车）	00001101B (0DH)

ASCII（美国标准信息交换码）码表见附录 B。

例如，字符“A”的 ASCII 码为 41H（65）；字符“B”的 ASCII 码为 42H（66）；字符“1”的 ASCII 码为 31H（49）；字符“2”的 ASCII 码为 32H（50）；Enter 键（回车）的 ASCII 码为 0DH（13）。