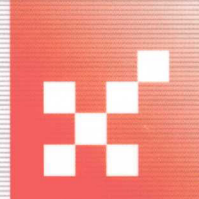




普通高等教育“十一五”国家级规划教材



数字电子技术基础

● 陈文楷 主编

 机械工业出版社
CHINA MACHINE PRESS



本书是普通高等教育“十一五”国家级规划教材，是教育部2002年

“十五”期间出版的国家重点教材，也是教育部2002年

普通高等教育“十一五”国家级规划教材

本书以CMOS逻辑门电路、三极管、集成运算放大器、数

数字电子技术基础

一种EDA软件的使用方法。

本书可作为电子信息类专业、通信工程专业、计算机专业、

www.cmpedu.com 注册下载或发邮件到 wpj@cmpedu.com 索取。

本书可作为电子信息类专业、通信工程专业、计算机专业、

用，也可供其他理工科相关专业师生参考。

主编 陈文楷
 参编 范秀娟
 主审 陆培新 彭斯福

图书在版编目(CIP)数据

数字电子技术基础 / 陈文楷主编. — 北京: 机械工业出版社, 2010.4
 ISBN 978-7-111-32832-8
 I. ①数… II. ①陈… III. ①数字电子技术—教材—高等学校
 IV. ①TN79
 中国版本图书馆CIP数据核字(2010)第071号
 责任编辑: 李莉
 封面设计: 王静
 北京机械工业出版社
 2010年4月第1版
 184mm×260mm·21
 印张: 18.4
 字数: 450千字
 ISBN 978-7-111-32832-8
 定价: 34.00元



机械工业出版社
 社址: 北京机械工业出版社
 电话: (010) 88381066
 邮购部: (010) 88379649
 发行部: (010) 88379649
 网址: <http://www.cmpedu.com>

本书是普通高等教育“十一五”国家级规划教材,是按照教育部2005年修订的“数字电子技术基础课程教学基本要求”编写的。

本书在教学内容的体系上做了一些改变,主要内容有数制与码制、逻辑代数基础、门电路、VHDL语言基础、组合逻辑电路、触发器、时序逻辑电路的分析与设计、存储器、可编程逻辑器件、脉冲波形的产生和整形、模数和数模转换器。

本书以CMOS逻辑门为主,减少了晶体管和小规模集成电路以及各种逻辑关系的内容。引入可编程逻辑器件和VHDL语言的内容,把数字电路与VHDL语言描述融合在一起。在学习数字电路的同时学习VHDL语言描述方法,学习教材内容的同时引入Quartus II仿真软件,使学生初步掌握一种EDA软件的使用方法。

本书配有免费电子课件,欢迎选用本书作教材的老师登录www.cmpedu.com注册下载或发邮件到wbj@cmpbook.com索取。

本书既可作为电气信息类、电子信息类、仪器仪表类等专业的教材使用,也可供其他理工科相关专业学生和社会读者阅读选用。

副主编 陈文楷 审主

图书在版编目(CIP)数据

数字电子技术基础/陈文楷主编. —北京:机械工业出版社,2010.3
普通高等教育“十一五”国家级规划教材
ISBN 978-7-111-29832-8

I. ①数… II. ①陈… III. ①数字电路-电子技术-高等学校-教材
IV. ①TN79

中国版本图书馆CIP数据核字(2010)第029308号

机械工业出版社(北京市百万庄大街22号 邮政编码100037)

策划编辑:王保家 责任编辑:王荣 版式设计:霍永明
责任校对:李秋荣 封面设计:张静 责任印制:乔宇
北京机工印刷厂印刷(三河市南杨庄国丰装订厂装订)

2010年4月第1版第1次印刷

184mm×260mm·21印张·523千字

标准书号:ISBN 978-7-111-29832-8

定价:34.00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

门户网:<http://www.cmpbook.com>

销售一部:(010)68326294

教材网:<http://www.cmpedu.com>

销售二部:(010)88379649

读者服务部:(010)68993821

封面无防伪标均为盗版

前 言

随着数字电子技术和电子设计自动化 (Electronic Design Automation, EDA) 技术应用的不断发展, 利用可编程逻辑器件 (PLD) 进行数字系统的开发已被广泛应用于通信、航天、医疗电子、工业控制等诸多领域。电子产品更新周期日益缩短, 新产品开发速度日益加快, 因而对电子设计自动化提出了更高的要求, 也有力地促进了 EDA 技术的发展和普及。与传统数字电路设计方法相比, PLD 具有功能强大、开发过程投资小、周期短、便于修改及开发工具智能化等特点。近年来随着电子产品市场的迅速发展和电子工艺不断改进, 低成本高性能的 PLD 不断推陈出新, 使 PLD 成为当今硬件设计方式的主流。为此, 本书针对当今快速发展的硬件及开发软件, 向读者讲述基本知识、基本理论和基本方法。

2005 年秋, 教育部电子信息科学与电气信息类基础课程教学指导分委员会主持重新修订了“数字电子技术基础课程教学基本要求”, 再次强调了本门课程的性质是“电子技术方面入门性质的技术基础课”, 其任务在于“使学生获得数字电子技术方面的基本知识、基本理论和基本技能, 为深入学习数字电子技术及其在专业中的应用打下基础”。

根据数字电子技术本身的发展状况和修订后教学大纲的基本要求, 考虑到延续教学和学习特点, 在保持传统的数字电子技术教学内容的基础上, 本书在内容、体系和风格上做了一些修改和补充。

本书共分 11 章, “数制和码制”单独编为第 1 章。第 2 章讲述“逻辑代数基础”。第 3 章“门电路”, 将 CMOS 电路放在前, 重点讲述 CMOS 逻辑门的功能和电气特性, 在逻辑门内部电路与逻辑功能的处理上着重强调了逻辑门的逻辑关系。第 4 章“VHDL 语言基础”讲述了 VHDL 语言的语法结构、特点, 用 VHDL 语言描述硬件电路的结构, VHDL 基本语句及应用等。第 5 章“组合逻辑电路”, 除了讲述常用的组合逻辑电路结构、工作原理之外, 增加了各种单元电路的 VHDL 设计实例, 如译码器、显示译码器、多路选择器、数值比较器、加法器及奇偶校验电路等, 并且对这些电路的描述语言进行了分析, 目的是使读者容易理解, 更快地掌握 VHDL 设计方法。第 6 章“触发器”、第 7 章“时序逻辑电路的分析与设计”、第 8 章“存储器”、第 9 章“可编程逻辑器件”, 像第 5 章一样在讲述硬件电路之后引入 VHDL 的文本描述程序, 在学习硬件电路的同时学习硬件描述语言。讲授本课程时, 在学习时的处理上要把硬件电路的结构及逻辑关系、工作原理作为重点, 在理解硬件原理的基础上学习 VHDL 语言将事半功倍。第 10 章讲述了波形发生和整形电路, 第 11 章讲述模数和数模转换技术。

本书在很多小节的末尾增加了复习思考题, 每一章后面都安排了习题, 习题包括 3 方面的内容, 包括硬件电路的分析和设计、仿真方法和 VHDL 语言设计。

本书第 9 章“可编程逻辑器件”采用了国际上流行的图形和逻辑符号, 其中基本运算和复合运算的符号采用了特定外形的图形符号。这种特定外形的图形符号已经补充到 1991 年修订的 IEEE/ANSI (The Institute of Electrical and Electronics Engineers/American National Standards Institute, 电气与电子工程师协会/美国国家标准化组织) 标准中。这些符号与清华

目 录

1.4.1	主从边沿触发器	14	3.2.3	CMOS 与非门和或非门	42
1.4.2	主从边沿 D 触发器	16	3.2.4	扩展输入门	43
1.4.3	边沿触发器的应用	17	3.2.5	驱动门	43
1.4.4	边沿触发器的应用	17	3.2.6	CMOS 与一或非门	44
1.4.5	边沿触发器的应用	17	3.3	CMOS 逻辑门的电特性	45
1.4.6	边沿触发器的应用	17	3.3.1	CMOS 逻辑门的静态特性	45
1.4.7	边沿触发器的应用	17	3.3.2	HC 和 HCT 系列	48
1.4.8	边沿触发器的应用	17	3.4	其他 CMOS 逻辑门	48
1.4.9	边沿触发器的应用	17	3.4.1	CMOS 异或逻辑门	48
1.4.10	边沿触发器的应用	17	3.4.2	CMOS 传输门	49
1.4.11	边沿触发器的应用	17	3.4.3	三态输出门	50
1.4.12	边沿触发器的应用	17	3.4.4	漏极开路输出门 (OD 门)	51
1.4.13	边沿触发器的应用	17	3.5	TTL 门电路	53
1.4.14	边沿触发器的应用	17	3.5.1	双极型晶体管的开关特性	53
1.4.15	边沿触发器的应用	17	3.5.2	TTL 反相门	54
1.4.16	边沿触发器的应用	17	3.5.3	TTL 系列其他类型的逻辑门	56
1.4.17	边沿触发器的应用	17	3.6	TTL 逻辑系列的电气特性	59
1.4.18	边沿触发器的应用	17	小结	60	
1.4.19	边沿触发器的应用	17	习题	61	
1.4.20	边沿触发器的应用	17	第 4 章 VHDL 语言基础	64	
1.4.21	边沿触发器的应用	17	内容提要	64	
1.4.22	边沿触发器的应用	17	4.1 概述	64	
1.4.23	边沿触发器的应用	17	4.1.1 EDA 技术和 HDL 的发展	64	
1.4.24	边沿触发器的应用	17	4.1.2 VHDL 和 Verilog HDL	65	
1.4.25	边沿触发器的应用	17	4.2 VHDL 程序结构	66	
1.4.26	边沿触发器的应用	17	4.3 实体和结构体	68	
1.4.27	边沿触发器的应用	17	4.4 用 Quartus II 开发数字系统	71	
1.4.28	边沿触发器的应用	17	4.4.1 Quartus II 集成环境开发软件	72	
1.4.29	边沿触发器的应用	17	4.4.2 Quartus II 集成开发软件的特点	72	
1.4.30	边沿触发器的应用	17	4.4.3 Quartus II 的基本开发流程	72	
1.4.31	边沿触发器的应用	17	4.5 VHDL 语法 Port、Mode、Type	78	
1.4.32	边沿触发器的应用	17	4.6 VHDL 信号的表示	92	
1.4.33	边沿触发器的应用	17	4.7 VHDL 程序结构语句	96	
1.4.34	边沿触发器的应用	17	4.7.1 程序结构语句	96	
1.4.35	边沿触发器的应用	17	4.7.2 并行语句结构	100	
1.4.36	边沿触发器的应用	17	4.7.3 顺序语句	104	
1.4.37	边沿触发器的应用	17	4.7.4 赋值语句	108	
1.4.38	边沿触发器的应用	17			
1.4.39	边沿触发器的应用	17			
1.4.40	边沿触发器的应用	17			
1.4.41	边沿触发器的应用	17			
1.4.42	边沿触发器的应用	17			
1.4.43	边沿触发器的应用	17			
1.4.44	边沿触发器的应用	17			
1.4.45	边沿触发器的应用	17			
1.4.46	边沿触发器的应用	17			
1.4.47	边沿触发器的应用	17			
1.4.48	边沿触发器的应用	17			
1.4.49	边沿触发器的应用	17			
1.4.50	边沿触发器的应用	17			
1.4.51	边沿触发器的应用	17			
1.4.52	边沿触发器的应用	17			
1.4.53	边沿触发器的应用	17			
1.4.54	边沿触发器的应用	17			
1.4.55	边沿触发器的应用	17			
1.4.56	边沿触发器的应用	17			
1.4.57	边沿触发器的应用	17			
1.4.58	边沿触发器的应用	17			
1.4.59	边沿触发器的应用	17			
1.4.60	边沿触发器的应用	17			
1.4.61	边沿触发器的应用	17			
1.4.62	边沿触发器的应用	17			
1.4.63	边沿触发器的应用	17			
1.4.64	边沿触发器的应用	17			
1.4.65	边沿触发器的应用	17			
1.4.66	边沿触发器的应用	17			
1.4.67	边沿触发器的应用	17			
1.4.68	边沿触发器的应用	17			
1.4.69	边沿触发器的应用	17			
1.4.70	边沿触发器的应用	17			
1.4.71	边沿触发器的应用	17			
1.4.72	边沿触发器的应用	17			
1.4.73	边沿触发器的应用	17			
1.4.74	边沿触发器的应用	17			
1.4.75	边沿触发器的应用	17			
1.4.76	边沿触发器的应用	17			
1.4.77	边沿触发器的应用	17			
1.4.78	边沿触发器的应用	17			
1.4.79	边沿触发器的应用	17			
1.4.80	边沿触发器的应用	17			
1.4.81	边沿触发器的应用	17			
1.4.82	边沿触发器的应用	17			
1.4.83	边沿触发器的应用	17			
1.4.84	边沿触发器的应用	17			
1.4.85	边沿触发器的应用	17			
1.4.86	边沿触发器的应用	17			
1.4.87	边沿触发器的应用	17			
1.4.88	边沿触发器的应用	17			
1.4.89	边沿触发器的应用	17			
1.4.90	边沿触发器的应用	17			
1.4.91	边沿触发器的应用	17			
1.4.92	边沿触发器的应用	17			
1.4.93	边沿触发器的应用	17			
1.4.94	边沿触发器的应用	17			
1.4.95	边沿触发器的应用	17			
1.4.96	边沿触发器的应用	17			
1.4.97	边沿触发器的应用	17			
1.4.98	边沿触发器的应用	17			
1.4.99	边沿触发器的应用	17			
1.4.100	边沿触发器的应用	17			

前言

第 1 章 数制与码制

内容提要

1.1 概述

1.2 数制的表示方法

1.3 十进制数与二进制数之间的转换

1.4 二进制数的算术运算

1.5 十六进制

1.6 二进制数的反码和补码

1.7 码制的表示方法

小结

习题

第 2 章 逻辑代数基础

内容提要

2.1 概述

2.2 逻辑代数的 3 种基本运算

2.3 逻辑代数的基本公式和常用公式

2.3.1 基本公式

2.3.2 若干常用公式

2.4 逻辑函数及其表示方法

2.4.1 逻辑函数

2.4.2 逻辑函数的表示方法

2.5 逻辑函数的两种标准形式

2.6 逻辑函数的公式化简法

2.7 逻辑函数的卡诺图化简法

2.7.1 逻辑函数的卡诺图表示法

2.7.2 用卡诺图化简逻辑函数

2.8 无关最小项在化简逻辑函数中的应用

小结

习题

第 3 章 门电路

内容提要

3.1 概述

3.2 CMOS 逻辑电路

3.2.1 MOS 晶体管的基本开关电路

3.2.2 CMOS 反相器

小结	108	6.4.1 主从式触发器的结构	167
习题	109	6.4.2 主从式 JK 触发器	169
第 5 章 组合逻辑电路	112	6.5 边沿触发的触发器	172
内容提要	112	6.5.1 边沿触发的方法	172
5.1 概述	112	6.5.2 边沿触发的 JK 触发器	173
5.2 组合逻辑电路的分析方法	112	6.6 触发器的动态特性和时间参数	174
5.3 组合逻辑电路的设计方法	114	6.7 VHDL 设计锁存器和触发器电路	175
5.4 加法器	115	6.7.1 库的概念及分类	175
5.4.1 半加器与全加器	115	6.7.2 VHDL 语言设计锁存器和触 发器	177
5.4.2 二进制加法器	118	6.7.3 边沿触发的 D 触发器设计	185
5.4.3 用 VHDL 实现加法器	121	6.7.4 异步置位/复位	187
5.5 译码器	129	6.7.5 同步置位/复位	189
5.5.1 二进制译码器	130	6.7.6 JK 触发器和 T 触发器设计	189
5.5.2 译码器的应用	133	小结	193
5.5.3 用 VHDL 语言设计译码器	134	习题	194
5.6 BCD 译码器和七段显示译码器	137	第 7 章 时序逻辑电路的分析与设计	198
5.6.1 BCD 译码器	137	内容提要	198
5.6.2 BCD—七段显示译码/驱动器	138	7.1 概述	198
5.7 多路选择器	142	7.2 时序逻辑电路的分析方法	199
5.7.1 多路选择器的概念	142	7.2.1 同步时序逻辑电路的分析	199
5.7.2 MSI 多路选择器	143	7.2.2 异步时序逻辑电路的分析	203
5.7.3 VHDL 多路选择器 (MUX) 设计	145	7.3 寄存器和移位寄存器	205
5.8 数值比较器	147	7.3.1 寄存器	205
5.8.1 4 位数值比较器	147	7.3.2 移位寄存器	206
5.8.2 中规模 (MSI) 4 位数值比 较器	147	7.4 IC 移位寄存器	208
5.8.3 VHDL 设计数值比较器	149	7.4.1 7495B 集成电路移位寄存器	208
5.9 编码器	152	7.4.2 双向移位寄存器	209
5.9.1 二进制编码器	152	7.4.3 通用移位寄存器 (74LS194)	209
5.9.2 优先编码器	153	7.5 寄存器与移位寄存器的 VHDL 设计	212
5.9.3 VHDL 优先编码器	156	7.5.1 4D 寄存器的设计	212
小结	157	7.5.2 移位寄存器的设计	213
习题	158	7.5.3 通用移位寄存器 (74LS194) 的 设计	214
第 6 章 触发器	162	7.5.4 循环移位寄存器的设计	216
内容提要	162	7.6 计数器	218
6.1 概述	162	7.6.1 异步计数器	219
6.2 SR 锁存器	162	7.6.2 同步计数器	222
6.3 同步式触发器	165	7.6.3 任意进制计数器	227
6.3.1 有使能控制端的 SR 锁存器	165	7.7 可逆计数器	231
6.3.2 同步式 SR 触发器	165	7.7.1 减法计数器	231
6.3.3 同步式 D 触发器	166	7.7.2 可逆计数器	232
6.4 主从式触发器	167	7.8 VHDL 计数器设计	234

7.8.1 二进制同步计数器的设计	234	9.1 概述	283
7.8.2 可逆计数器的设计	238	9.2 基本可编程逻辑器件	283
7.8.3 具有装数、进位输出功能的同 步计数器的设计	239	9.3 通用阵列逻辑 GAL	286
7.9 状态机的设计	241	9.3.1 GAL 的结构及原理	286
7.9.1 概述	241	9.3.2 GAL16V8 的结构及应用	288
7.9.2 状态机	242	9.4 HDPLD	292
7.9.3 状态机的设计方法与步骤	242	9.4.1 阵列扩展型 CPLD	294
7.9.4 摩尔型状态机的设计	243	9.4.2 FPGA 的内部结构	299
7.9.5 状态机的自启动设计	246	9.5 用 PLD 实现数字系统	305
7.9.6 米利型状态机的设计	248	小结	306
7.10 VHDL 实现状态机的设计	250	习题	306
7.10.1 摩尔型 VHDL 有限状态机的 设计	250	第 10 章 脉冲波形的产生和整形	307
7.10.2 米利型 VHDL 有限状态机的 设计	254	内容提要	307
7.10.3 状态机的自启动 VHDL 设计	258	10.1 概述	307
小结	262	10.2 施密特触发器	307
习题	262	10.3 555 多谐振荡器	310
第 8 章 存储器	267	10.4 单稳态触发器	313
内容提要	267	小结	315
8.1 概述	267	习题	315
8.2 随机存储器 (RAM)	267	第 11 章 模数—数模转换器	316
8.2.1 静态 RAM (SRAM)	269	内容提要	316
8.2.2 动态 RAM (DRAM)	271	11.1 概述	316
8.3 只读存储器 (ROM)	274	11.2 D/A 转换器	316
8.3.1 掩膜只读存储器	275	11.2.1 权电阻网络 D/A 转换器	316
8.3.2 可编程只读存储器	275	11.2.2 倒 T 形电阻网络 D/A 转换器	317
8.4 快闪存储器	278	11.2.3 双极性输出的 D/A 转换器	318
8.5 存储器扩展及应用	280	11.2.4 D/A 转换器的转换精度和转 换速度	320
8.5.1 位扩展方式	280	11.3 A/D 转换器	322
8.5.2 字扩展方式	281	11.3.1 逐次逼近型 A/D 转换器	322
小结	281	11.3.2 积分型 A/D 转换器	323
习题	282	11.3.3 A/D 转换器的几个主要参数	326
第 9 章 可编程逻辑器件	283	小结	326
内容提要	283	习题	327
		参考文献	328

第 1 章 数制与码制

内容提要

本章介绍数字电路中常用的数制和码制、数制和码制的表示方法及它们在实际中的应用,主要包括一些基本概念和术语、不同数制之间的转换方法、二进制数学运算的原理和方法。

1.1 概述

数字电子技术是研究数字信号的产生及数字信号处理的学科。数字信号指在时间和数值上是不连续的量值,它们的变化发生在一系列离散的瞬间,它们的数值大小和每次的增减变化都是某一最小单位的整数倍,而小于这个最小单位的数值是无法识别的。数字信号可以是在时间上持续极短的电压或电流波形,可以是矩形波、方波和尖峰波等,如图 1.1.1 所示。通常把工作数字信号下的电子电路称为数字电路。数字电子技术涉及的电路和系统只有两种可能的状态,分别用两个不同的值表示:高电压(又称高电平)用 1 表示,低电压(又称低电平)用 0 表示。用这两个状态也可以表示电流的有或无、开关的接通或断开、灯亮或灯灭。在数字系统中,这两种状态的组合(0 和 1)又被叫作数码,用来表示数、符号、字符和其他类型的信息,这两种状态的数制被叫作二进制,而 1 个二进制数字被称作位。表示事物的两个状态用的数字是 0 和 1。

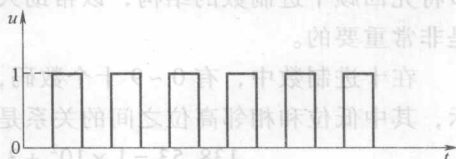


图 1.1.1 数字信号及表示方法

在数字电路中,两个不同的电压可表示成二进制中的两个数,1 表示高电压,0 表示低电压,这通常称作正逻辑。位的组合(即 1 和 0 的组合)被叫作码,多位码的组合用来表示数据、字母、符号、指令和任何其他可能使用的信息。

数字电路是由逻辑门运算或操作进行工作的,逻辑门的运算或操作表示成 1 和 0 的逻辑电平(一般是直流低电压 $0 \sim +5V$, 或者 $0 \sim +3V$), 1 代表高电平, 0 代表低电平。在实际的数字电路中,高电平“1”可以用一定范围(例如可以选取 $+3.5 \sim +5V$)内的电压值表示;同样,低电平“0”也可以用一定范围内的低电压(可以选取 $0 \sim +1.5V$)范围内的电压表示。数字电路正常工作的高电平和低电平不能重叠。图 1.1.2 说明数字电路中高低电平的一般范围,图中, V_H 代表高电压, V_L 代表低电压, $V_{H(max)}$ 表示高电压的最大值, $V_{H(min)}$ 表示高电压的最小值, $V_{L(max)}$ 表示低电压的最大值, $V_{L(min)}$ 表示低电压的最小值。 $V_{H(min)}$ 和 $V_{L(max)}$ 之间不允许有重叠的工作区,称为中间过渡区。具体工作情况在第 3 章中学习。

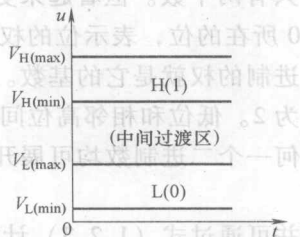


图 1.1.2 逻辑高、低电平示意图

1.2 数制的表示方法

逻辑运算是靠编码操作进行的, 数字电路和计算机综合了算术运算、代数运算、逻辑运算, 因此要学习数制和码制的表示方法及其运算方法。

二进制数和码是计算机和数字电子技术的基础。这一节介绍二进制数和其他数制的关系, 如十进制、十六进制和八进制等。而二进制数的算术运算是了解计算机和很多其他类型的数字系统工作的基础。

用数字表示物理量的大小时, 仅用一位数码往往不够用, 因此, 经常需要用进位计数的方法组成多位数码使用, 并把多位数码中每一位的构成方法以及从低位到高位进位的规则称为数制。

1. 十进制 十进制是人们非常熟悉的, 但它们的权值结构常不被人们所了解, 这一小节将先回顾十进制数的结构, 以帮助人们更进一步地了解二进制数的结构, 这在数字电路中是非常重要的。

在十进制数中, 有 0~9 十个数码, 所以计数的基数是 10。超过 9 的数必须用多位数表示, 其中低位和相邻高位之间的关系是“逢十进一”, 故称为十进制。例如

$$138.53 = 1 \times 10^2 + 3 \times 10^1 + 8 \times 10^0 + 5 \times 10^{-1} + 3 \times 10^{-2}$$

所以, 任意一个十进制数均可以展开为

$$D = \sum K_i \times 10^i \quad (1.2.1)$$

式中, K_i 是第 i 位的系数, 它可以是 0~9 这十个数码中的任何一个。如整数部分的位数是 n , 小数部分的位数为 m , 则 i 包含从 $n-1$ 到 0 所有正整数和从 -1 到 $-m$ 所有负整数。

若以 N 取代式 (1.2.1) 中的 10, 即可得到任意进制 (N 进制) 数展开式的普遍形式

$$D = \sum K_i N^i \quad (1.2.2)$$

式中, i 的范围与式 (1.2.1) 的规定相同; N 为计数的基数; K_i 为第 i 位的系数; N^i 为第 i 位的权。

2. 二进制 二进制是表示物理量的另一种简单方法。二进制没有十进制那么复杂, 它只有两个数。但看起来更困难, 因为不熟悉它。两个二进制数是 1 和 0, 在二进制数中 1 或 0 所在的位, 表示位的权, 或者是数的值。就像十进制数的位一样, 代表该位数的值, 这二进制的权就是它的基数。在二进制数中, 每一位仅有 0 和 1 两个可能的数, 所以计数的基数为 2。低位和相邻高位间的进位关系是“逢二进一”, 故叫作二进制。根据式 (1.2.2), 任何一个二进制数均可展开为

$$D = \sum K_i 2^i \quad (1.2.3)$$

并可通过式 (1.2.3) 计算出它所表示的十进制数大小。例如,

$$(1101.11)_2 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} = (13.75)_{10}$$

上式中分别使用下脚注的 2 和 10 表示括弧里的数是二进制数和十进制数, 有时也用 B (Binary) 和 D (Decimal) 代替 2 和 10 这两个脚注。

3. 十六进制 十六进制数的每一位有 16 个不同的数码, 分别用 0~9、A、B、C、D、E、F 表示。因此, 任意一个十六进制数均可展开为

$$D = \sum K_i 16^i \quad (1.2.4)$$

并由此式计算出它所表示的十进制数值。例如

$$(3E.2A)_{16} = 3 \times 16^1 + 14 \times 16^0 + 2 \times 16^{-1} + 10 \times 16^{-2}$$

$$= (48 + 14 + 2 \times 0.0625 + 10 \times 0.000153)_{10}$$

$$= (62.125153)_{10}$$

式中的下脚注 16 表示括号里的数是十六进制，下脚注 10 表示括号里的数是十进制，有时也用 H (Hexadecimal) 代替这个脚注。由于十六进制信息量大，而 8 位、16 位和 32 位的二进制数可以用 2 位、4 位或 8 位的十六进制数表示，因而用十六进制符号书写程序十分简便。目前在微型计算机中普遍采用 8 位、16 位和 32 位二进制并行运算。

1.3 十进制数与二进制数之间的转换

1. 二—十进制转换 由二进制数的表示方法知道，二进制计数是逢二进一，3 位二进制计数法最大计到 $2^3 = 8$ ，4 位二进制计数法最大计到 $2^4 = 16$ ， n 位二进制计数最大计数为 2^n 个数，表示成 $D = 2^n - 1$ 。

把二进制数转换为等值的十进制数称为二—十进制转换。转换时只要将二进制数按式 (1.1.3) 展开，然后把所有各项的数值按十进制数相加，就可以得到等值的十进制数了。转换方法是：将整数部分所有位为 1 的权值相加，就得到十进制数的整数部分；将小数部分所有为 1 的位取负指数权值后相加，就得到十进制数的小数部分。

[例 1.3.1] 求 $(1101011)_2$ 的十进制数。

$$\begin{aligned} \text{解：} (1101011)_2 &= 1 \times 2^6 + 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \\ &= (64 + 32 + 0 + 8 + 0 + 2 + 1)_{10} \\ &= (107)_{10} \end{aligned}$$

[例 1.3.2] 求 $(0.1011)_2$ 的十进制数。

$$\begin{aligned} \text{解：} (0.1011)_2 &= 0.0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} + 1 \times 2^{-4} \\ &= (0.0 + 0.5 + 0 + 0.125 + 0.0625)_{10} \\ &= (0.6875)_{10} \end{aligned}$$

[例 1.3.3] 求 $(1011.01)_2$ 的十进制数。

$$\begin{aligned} \text{解：} (1011.01)_2 &= 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} \\ &= (11.25)_{10} \end{aligned}$$

2. 十—二进制转换 十—二进制转换，就是把十进制数转换成等值的二进制数。

下面讨论整数部分的转换。有两种转换方法，一种是加权求和法，另一种是连续除以 2 的方法。首先介绍加权求和法，把十进制数分解成二进制权值相加。假设十进制数为 $(S)_{10}$ ，等值的二进制数为 $(k_n, k_{n-1}, \dots, k_0)_2$ ，则依式 (1.2.3) 可知

$$(S)_{10} = k_n 2^n + k_{n-1} 2^{n-1} + \dots + k_1 2^1 + k_0 2^0 \quad (1.3.1)$$

只要依次确定 k_n, \dots, k_0 的系数取 1 或者 0 即可。

将式 (1.3.1) 改写成下式就为连续除以 2 方法。

$$\begin{aligned} (S)_{10} &= k_n 2^n + k_{n-1} 2^{n-1} + \dots + k_1 2^1 + k_0 2^0 \\ &= 2(k_n 2^{n-1} + k_{n-1} 2^{n-2} + \dots + k_1) + k_0 \quad (1.3.2) \end{aligned}$$

上式表明,若将 $(S)_{10}$ 除以 2, 则得到的商为 $k_n 2^{n-1} + k_{n-1} 2^{n-2} + \dots + k_1$, 而余数即 k_0 。同理, 将式 (1.3.2) 中的商除以 2 得到新的商可写成

$$k_n 2^{n-2} + k_{n-1} 2^{n-3} + \dots + k_1 = 2(k_n 2^{n-3} + k_{n-1} 2^{n-4} + \dots + k_2) + k_1 \quad (1.3.3)$$

由式 (1.3.3) 不难看出, 若将 $(S)_{10}$ 除以 2 所得的商再次除以 2 所得余数即 k_1 。

依次类推, 反复将每次得到的商再次除以 2, 就可求得二进制数的每一位了。

例如, 将 $(145)_{10}$ 转换成二进制数可如下进行

2	145	余数 = 1 = k_0 (LSB)
2	72	余数 = 0 = k_1
2	36	余数 = 0 = k_2
2	18	余数 = 0 = k_3
2	9	余数 = 1 = k_4
2	4	余数 = 0 = k_5
2	2	余数 = 0 = k_6
2	1	余数 = 1 = k_7 (MSB)
	0		

于是, $(145)_{10} = (1001\ 0001)_2$ 。

对于十进制小数转换成二进制数, 也有两种转换方法。一种是加权求和法, 另一种是连续乘以 2 的方法。

应用加权求和法对十进制的小数部分进行转换, 要记住小数的最高位权值是 0.5, 即 2^{-1} ; 后面各位的权值依次减半, 列出小数点后面 4 位的权值依次为 0.5, 0.25, 0.125, 0.0625。

若 $(S)_{10}$ 是一个十进制的小数, 则对应的二进制小数为

$$(S)_{10} = k_{-1} 2^{-1} + k_{-2} 2^{-2} + \dots + k_{-m} 2^{-m} \quad (1.3.4)$$

只要根据权值的位置依次确定 k_{-1}, \dots, k_{-m} 的系数取 1 或者 0 即可。

[例 1.3.4] 将 $(0.625)_{10}$ 转换成二进制小数。

$$\begin{aligned} \text{解: } (0.625)_{10} &= 0.5 + 0.125 = 0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} \\ &= (0.101)_2 \end{aligned}$$

将式 (1.3.4) 两边同乘以 2, 得到

$$2(S)_{10} = k_{-1} + (k_{-2} 2^{-1} + \dots + k_{-m} 2^{-m+1}) \quad (1.3.5)$$

上式说明, 将小数 $(S)_{10}$ 乘以 2 得到乘积的整数部分系数 k_{-1} 。同理, 将乘积的小数部分再乘以 2 又可得到

$$2(k_{-2} 2^{-1} + \dots + k_{-m} 2^{-m+1}) = k_{-2} + (k_{-3} 2^{-1} + \dots + k_{-m} 2^{-m+1}) \quad (1.3.6)$$

亦即, 乘积的整数部分就是 k_{-2} 。

用连续乘以 2 的方法将十进制小数转换成二进制小数的步骤归纳如下: 首先, 把十进制小数乘以 2, 得到第 1 个积; 第二步, 用乘积的小数部分乘以 2, 得到第 2 个积; 第三步, 用乘积的小数部分乘以 2, 得到第 3 个积, 直到积的小数部分乘以 2 等于 0 为止, 或者得到预期的十进制数为止, 每次乘积得到的进位排列起来就得到要转换的二进制小数。第一个进位为最高位, 最后一个进位为最低位。

1.4 二进制数的算术运算

与十进制算术运算相同，二进制数的算术运算（加、减、乘、除）很简单，只要记住各种运算的4条基本规则就可以。

1. 二进制加法 有进位的二进制加法的4条基本规则如下：

	二进制		十进制
A	0000	0 + 0 = 0	0
B	0001	0 + 1 = 1	1
C	0010	1 + 1 = 10	2
D	0011	1 + 1 + 1 = 11	3

[例 1.4.1] 计算 011 + 001 = ?
解: 011 + 001 = 100

2. 二进制减法 有借位的二进制减法的4条基本规则如下：

	二进制		十进制
A	0 - 0 = 0		
B	1 - 1 = 0		
C	1 - 0 = 1		
D	1 - 1 = 0 (有借位1)		

[例 1.4.2] 101 - 011 = ?
解: 101 - 011 = 010

3. 二进制乘法 4条基本规则如下：

二进制	十进制
0 × 0 = 0	
0 × 1 = 0	
1 × 0 = 0	
1 × 1 = 1	

[例 1.4.3] 101 × 111 = ?
解: 101 × 111 = 100011

4. 二进制除法 与十进制除法相同。

[例 1.4.4] 110 ÷ 11 = ?
解: 110 ÷ 11 = 10

1.5 十六进制

十六进制数有16个数，主要用于数码显示和微处理机的汇编语言编程。因为计算机只理解1和0的指令编码，用十六进制或八进制表示16位的指令方便得多，它们与二进制的相互转换很容易。

十六进制数的基数为16，16个数字由9个阿拉伯字符和6个英文字母组成，因为每个十六进制数用4位二进制数表示，如表1.5.1所示。

十六进制数的计数方法和其他进制计数一样，也是连续计数。计数方法如下：
10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, 1C, 1D, 1E, 1F; 十六进制数的商

20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 2A, 2B, 2C, 2D, 2E, 2F; ...

两个十六进制的计数可以计到 FF_{16} , 相当于十进制的 255, 如果需要超过这个数, 如 3 位十六进制数 100_{16} 是十进制数 256, 101_{16} 是十进制数 257, 依次类推。最大的 3 位十六进制数 FFF_{16} , 是十进制数 4095, 最大的 4 位十六进制数 $FFFF_{16}$, 是十进制数 65535。

表 1.5.1 十进制—二进制—十六进制转换表

十进制	二进制	十六进制	十进制	二进制	十六进制
0	0000	0	8	1000	8
1	0001	1	9	1001	9
2	0010	2	10	1010	A
3	0011	3	11	1011	B
4	0100	4	12	1100	C
5	0101	5	13	1101	D
6	0110	6	14	1110	E
7	0111	7	15	1111	F

1. 二—十六进制转换 把二进制数转换成等值的十六进制数称为二—十六进制转换。

由于 4 位二进制数恰好有 16 个状态, 而把这 4 位二进制数看作一个整体时, 它的进位输出又正好是逢十六进一, 所以只要从低位到高位将每 4 位二进制数分为一组并代之以等值的十六进制数, 即可得到对应的十六进制数。

[例 1.5.1] 将 $(1100\ 1010\ 0101\ 0110)_2$ 转换成十六进制数。

解: $(1100\ 1010\ 0101\ 0110)_2 = (CA56)_{16}$

2. 十六—二进制转换 十六—二进制转换是把十六进制数转换成等值的二进制数。转换时只需将十六进制数的每一位用等值的 4 位二进制数代替就行了。

[例 1.5.2] 将下列各数分别转换为二进制数。

(1) $(10A8)_{16}$ (2) $(C8F.6C)_{16}$ (3) $(9746)_{16}$

解:

(1) $(10A8)_{16} = (1\ 0000\ 1010\ 1000)_2$

(2) $(C8F.6C)_{16} = (1100\ 1000\ 1111.0110\ 1100)_2$

(3) $(9746)_{16} = (1001\ 0111\ 0100\ 0110)_2$

式 (1) 中, 最高位的前 3 位是 0 (未标出) 与后面的 1 组成一个 4 位二进制数组。

3. 十六—十进制转换 将十六进制数转换为十进制数时, 可以是先将十六进制数转换成二进制数, 然后再从二进制数转换为十进制数。如下面例子的转换过程

$$\begin{aligned} (1B3)_{16} &= (0001\ 1011\ 0011)_2 = 2^8 + 2^7 + 2^5 + 2^4 + 2^1 + 2^0 \\ &= 256 + 128 + 32 + 16 + 2 + 1 = (435)_{10} \end{aligned}$$

另一种方法可根据式 (1.2.4) 将各位按权展开后相加求得。例如,

只将 (1) $(E8)_{16} = 14 \times 16^1 + 8 \times 16^0 = 224 + 8 = (232)_{10}$

再将 (2) $(B2F8)_{16} = 11 \times 16^3 + 2 \times 16^2 + 15 \times 16^1 + 8 \times 16^0$

$$= 11 \times 4096 + 2 \times 256 + 15 \times 16 + 8 \times 1$$

再将 (3) $(9746)_{16} = 9 \times 16^3 + 7 \times 16^2 + 4 \times 16^1 + 6 \times 16^0$

$$= 9 \times 4096 + 7 \times 256 + 4 \times 16 + 6 \times 1$$

4. 十—十六进制转换 十进制数到十六进制数的转换是用十进制数重复除以 16, 得到的商就是十六进制数。这个过程与十进制转换成二进制的过程类似。值得注意的是, 当商有

小数时，小数部分要乘以除数得到余数。

[例 1.5.3] 将十进制数 650 用重复除以 16 的方法转换成十六进制数。

解：

$$\begin{array}{l}
 \frac{650}{16} = 40.625 \longrightarrow 0.625 \times 16 = A \\
 \frac{40}{16} = 2.5 \longrightarrow 0.5 \times 16 = 8 \\
 \frac{2}{16} = 0.125 \longrightarrow 0.125 \times 16 = 2
 \end{array}$$

(十六进制数)

当商的整数等于 0 时停止

MSB LSB

5. 十六进制数的加法 两个十六进制数直接相加即可，但要记住十六进制数 0~9 与十进制数 0~9 等值，十六进制 A~F 与十进制 10~15 等值。当两个十六进制数相加时，遵循下面的规则。

(1) 同一列的两个十六进制数相加，把两个十六进制数看成十进制数相加即可。例如， $5_{16} = 5_{10}$ 和 $C_{16} = 12_{10}$ 。

(2) 若两个数的和是等于或小于 15_{10} ，则相应的 16 进制数保留。

(3) 两个数的和大于 15_{10} ，则超过 16_{10} 的部分向高位进 1。

[例 1.5.4] 计算下面十六进制数的加法。

$$\begin{array}{r}
 \text{DF}_{16} + \text{AC}_{16} \\
 + \text{AC}_{16} \\
 \hline
 \text{18B}_{16}
 \end{array}$$

右列 $F_{16} + C_{16} = 15_{10} + 12_{10} = 27_{10}$
 $27_{10} - 16_{10} = 11_{10} = B_{16}$ ，进位 1

左列 $D_{16} + A_{16} = 13_{10} + 10_{10} + 1_{10} = 24_{10}$
 $24_{10} - 16_{10} = 8_{10} = 8_{16}$ ，进位 1

6. 十六进制数的减法 由于十六进制数可以表示成二进制数，也可以表示成二进制的补码。例如， $C9_{16}$ 表示成二进制数为 11001001，这个二进制数的补码为 00110111，写成十六进制是 37_{16} 。

十六进制数的减法可以把十六进制数表示成二进制数，然后用二进制补码做加法实现十六进制数减法。

[例 1.5.5] 计算下面十六进制数的减法。

(1) $84_{16} - 2A_{16}$ (2) $C3_{16} - 0B_{16}$

解：

(1) $2A_{16} = (00101010)_2$

$2A_{16}$ 的补码 = $(11010110)_2 = D6_{16}$

$$\begin{array}{r}
 84_{16} \\
 + D6_{16} \\
 \hline
 15A_{16}
 \end{array}$$

同不悉兹，商率示示素量字数用... 舍弃进位 1，得到补码和

... $84_{16} - 2A_{16} = 5A_{16}$...

(2) $0B_{16} = (00001011)_2$

$0B_{16}$ 的补码: $0B_{16} = 11110101 = F5_{16}$

$$\begin{array}{r} \text{C3}_{16} \\ + \text{F5}_{16} \\ \hline 1\text{B8}_{16} \end{array}$$

加
舍弃进位1, 得到补码和

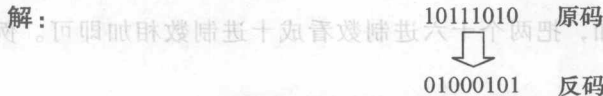
$\text{C3}_{16} - 0B_{16} = \text{B8}_{16}$

1.6 二进制数的反码和补码

二进制数的反码和补码是很重要的, 它们可以表示负数。二进制的补码运算通常用于计算机和数字系统中处理负数。

1. 二进制数的反码 求二进制数的反码, 把所有为1的位变为0, 为0的位变为1就得到反码。

[例 1.6.1] 用1变为0, 0变为1的方法求10111010的反码。



求反码的运算在数字电路中是很容易实现的, 只要用一组非门对原码求反即可。如图 1.6.1 所示。

2. 二进制数的补码 二进制数的补码是这样定义的: 左边最高位为符号位, 0表示正数、1为负数; 正数的补码和它的原码相同; 负数的补码可通过将原码的数值位逐位求反, 然后在最低位上加1得到。一个数的补码是用它的反码加1得到: 补码 = 原码的反码 + 1。

求二进制补码的运算在数字电路中也很容易实现, 如图 1.6.2 所示, 只要用一组非门对原码求反, 用加法器再加1即得到二进制补码。

[例 1.6.2] 求10110110的补码。

解:

10110110	二进制数
01001001	反码
+ 1	加1
01001010	补码

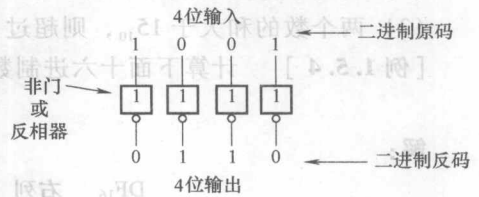


图 1.6.1 二进制反码的实现方法

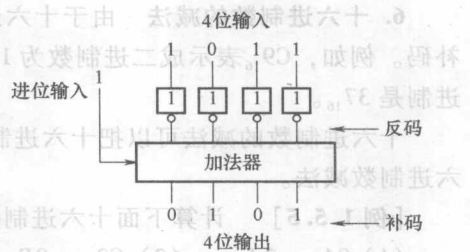


图 1.6.2 二进制补码的实现方法

1.7 码制的表示方法

利用数字符号作为某一特定的信息代号称作码制。码是用数字量表示的事物, 这些不同事物的代号叫作码。不同的数码不仅可以表示数量的不同大小, 用于计数或运算, 而且还能

用来表示不同的事物。这时数码已没有表示数量大小的含义，而是代表某种物理意义，只是表示不同事物的代号而已。码也可以表示事物的逻辑关系，这些数码称为代码。在编制代码时总要遵循一定的规则，这些规则就叫作码制。

码的种类很多，如有权码（8421码、2421码），无权码（余3码、格雷码），ASCII（American Standard Code for Information Interchange）码。

二进制码表示的十进制数，又称BCD（Binary-Code-Decimal）码，就是用4位二进制数码表示一位十进制数的方法，因为BCD码只有0~9的编码状态，这10个编码状态，又可以构成多种不同的码制。通常将这些代码通称为二—十进制代码。

表1.7.1中列出了几种常见的BCD代码，它们的编码规则各不相同，其中8421码是BCD代码中最常用的一种。在这种编码方式中，每一位二值代码的1都代表了一个固定数值，把每一位的1代表的十进制数加起来，得到的结果就是它所代表的十进制数。由于代码中从左到右每一位的1分别表示权重为8、4、2、1，所以把这种代码叫做8421码。每一位1代表的十进制数称为这一位的权。8421码中每一位的权是固定不变的，所以它属于恒权代码。

表 1.7.1 几种常见的 BCD 代码

十进制数 \ 编码种类	8421 码	余 3 码	2421 码	余 3 循环码	格雷码
0	0000	0011	0000	0010	0000
1	0001	0100	0001	0110	0001
2	0010	0101	0010	0111	0011
3	0011	0110	0011	0101	0010
4	0100	0111	0100	0100	0110
5	0101	1000	1011	1100	0111
6	0110	1001	1100	1101	0101
7	0111	1010	1101	1111	0100
8	1000	1011	1110	1110	1100
9	1001	1100	1111	1010	1101
权	8421		2421		

余3码的编码规则与8421码不同，如果把每一个余3码看作4位二进制数，则它的数值要比它所表示的十进制数码多3，故将这种代码称为余3码。

如果将两个余3码相加，所得的和将比十进制数和所对应的二进制数多6。因此，在用余3码作十进制运算时，若两数之和为10，正好等于二进制数的16，于是便从高位自动产生一个进位。

此外，从表1.7.1中还可以看出，0和9、1和8、2和7、3和6、4和5的余3码是互为相反的码，这对于求10的补码是很方便的。余3码不是恒权代码。如果试图把每个代码视为二进制数，并使它等效的十进制数与所表示的代码相等，那么代码中每一位的1所代表的十进制数在各个代码中不能是固定的。

2421码是一种恒权代码，它的0和9、1和8、2和7、3和6、4和5互为反码，这个特点和余3码相仿。

余3循环码是一种变权码，每一位的1在不同代码中并不代表固定的数值。它的主要特点是相邻的两个代码之间仅有一位的状态不同。因此，按余3循环码接成计数器时，每次状态转换过程中只有一个触发器翻转，译码时不会发生竞争冒险现象。

BCD加法 BCD码是一种数值码，并且能够用于算术运算。加法是最重要的运算，因为