



高等学校电子与通信类专业“十一五”规划教材

数字电路与逻辑设计

主编 白 静
主审 王 荣



西安电子科技大学出版社
<http://www.xduph.com>

高等学校电子与通信类专业“十一五”规划教材

数字电路与逻辑设计

主编 白 静

主审 王 荣

西安电子科技大学出版社

2009

内 容 简 介

本书详细介绍了数字电路与逻辑设计的基本理论及设计方法。对课程内容进行了优化,力求在知识体系、理论教学与实践教学等方面充分体现本课程的基础性、先进性和实用性。

本书在注重知识连贯性的基础上,精简了传统分立元件、小规模集成电路等内容,增加了可编程逻辑器件 PLD、硬件描述语言 VHDL 以及课程设计等内容,能较好地适应现代数字技术的发展要求和本课程教学的实际需要。

全书共十章,主要内容包括数字逻辑基础、集成逻辑门电路、组合逻辑电路、集成触发器、时序逻辑电路、脉冲产生电路、半导体存储器、可编程逻辑器件、硬件描述语言 VHDL,以及典型的数字电路课程设计。

本书概念清楚、选材典型、深入浅出、语言流畅、通俗易懂,并配有大量例题帮助学习和理解。可作为高等学校电子信息、通信工程、电气工程、自动化等专业的教材,也可供相关工程技术人员阅读和参考。

☆ 本书配有电子教案,需要者可与出版社联系,免费提供。

图书在版编目(CIP)数据

数字电路与逻辑设计/白静主编. —西安:西安电子科技大学出版社,2009.8
高等学校电子与通信类专业“十一五”规划教材
ISBN 978-7-5606-2256-9

I. 数… II. 白… III. 数字电路—逻辑设计—高等学校—教材 IV. TN79

中国版本图书馆 CIP 数据核字(2009)第 070988 号

策 划 寇向宏
责任编辑 寇向宏
出版发行 西安电子科技大学出版社(西安市太白南路2号)
电 话 (029)88242885 88201467 邮 编 710071
网 址 www.xduph.com 电子邮箱 xdupfxb001@163.com
经 销 新华书店
印刷单位 陕西天意印务有限责任公司
版 次 2009年8月第1版 2009年8月第1次印刷
开 本 787毫米×1092毫米 1/16 印张 21.25
字 数 497千字
印 数 1~4000册
定 价 30.00元

ISBN 978-7-5606-2256-9/TN·0511

XDUP 2548001-1

*** 如有印装问题可调换 ***

本社图书封面为激光防伪覆膜,谨防盗版。

西安电子科技大学出版社
高等学校电子与通信类专业“十一五”规划教材
编审专家委员会名单

主任：杨震（南京邮电大学校长、教授）

副主任：张德民（重庆邮电大学通信与信息工程学院副院长、教授）

秦会斌（杭州电子科技大学电子信息学院院长、教授）

通信工程组

组长：张德民（兼）

成员：（成员按姓氏笔画排列）

王晖（深圳大学信息工程学院副院长、教授）

巨永锋（长安大学信息工程学院副院长、教授）

成际镇（南京邮电大学通信与信息工程学院副院长、副教授）

刘顺兰（杭州电子科技大学通信工程学院副院长、教授）

李白萍（西安科技大学通信与信息工程学院副院长、教授）

张邦宁（解放军理工大学通信工程学院卫星系系主任、教授）

张瑞林（浙江理工大学信息电子学院院长、教授）

张常年（北方工业大学信息工程学院院长、教授）

范九伦（西安邮电学院信息与控制系系主任、教授）

姜兴（桂林电子科技大学信息与通信学院副院长、教授）

姚远程（西南科技大学信息工程学院副院长、教授）

康健（吉林大学通信工程学院副院长、教授）

葛利嘉（中国人民解放军重庆通信学院军事信息工程系系主任、教授）

电子信息工程组

组长：秦会斌（兼）

成员：（成员按姓氏笔画排列）

王荣（解放军理工大学通信工程学院电信工程系系主任、教授）

朱宁一（解放军理工大学理学院基础电子学系系主任、工程师）

李国民（西安科技大学通信与信息工程学院院长、教授）

李邓化（北京信息工程学院信息与通信工程系系主任、教授）

吴谨（武汉科技大学信息科学与工程学院电子系系主任、教授）

杨马英（浙江工业大学信息工程学院副院长、教授）

杨瑞霞（河北工业大学信息工程学院院长、教授）

张雪英（太原理工大学信息工程学院副院长、教授）

张彤（吉林大学电子科学与工程学院副院长、教授）

张焕君（沈阳理工大学信息科学与工程学院副院长、副教授）

陈鹤鸣（南京邮电大学光电学院院长、教授）

周杰（南京信息工程大学电子与信息工程学院副院长、教授）

欧阳征标（深圳大学电子科学与技术学院副院长、教授）

雷加（桂林电子科技大学电子工程学院副院长、教授）

项目策划：毛红兵

策划：曹 昶 寇向宏 杨 英 郭 景

前 言

“数字电路与逻辑设计”是通信、电子信息类专业的主要基础课程，也是当前发展最快的学科之一。本书是针对体现“面向 21 世纪，注重应用能力培养”的教学特点，结合作者多年的教学和实践经验进行编写的，力求在内容、体系、理论与实践教学方面充分体现新世纪对本课程的教学要求。

本书具有以下特点：

(1) 内容上具有完整的理论体系。本书的理论体系符合数字技术的发展趋势，能够引导学生掌握扎实的理论基础。

(2) 注重基本理论、基本分析和设计方法，重点突出。在内容编排上，遵从先易后难、由浅入深、循序渐进的原则，对重点、难点进行深入分析和讨论，通过大量典型例题的学习，加深对重要的知识点和考点的理解。

(3) 注重当前数字电子技术的发展要求，增加了可编程逻辑器件 PLD 和硬件设计语言 VHDL 等内容。详细介绍了 PLA、PAL、GAL、FPGA、CPLD 等器件的基本工作原理及应用，并通过实例介绍了用 VHDL 设计实现组合逻辑电路和时序逻辑电路。

(4) 理论教学与实践教学相结合。针对本课程具有实践性强的特点，为培养学生学以致用能力，本书在理论分析的基础上，给出了典型的数字电路与逻辑设计课程设计实例，以培养学生分析问题和解决问题的能力。

(5) 选材典型，结构清晰，文字简练，通俗易懂。本书适合作为高等学校电子信息、通信工程、电气工程、计算机科学与技术、自动化等专业的本科教材，也可作为报考电子、信息、通信专业的硕士研究生考生的复习参考用书。

全书共十章，由白静任主编，张雪英任副主编。第一章到第七章、第十章及附录由白静编写，第八章和第九章由张雪英编写。解放军理工大学的王荣老师担任本书的主审。在此书完成之际，谨向对本书提供过帮助的人们以及给与我们支持的家人表示诚挚的谢意！

限于编者水平，书中难免存在不足之处，恳请读者批评指正。

编 者

2009 年 1 月

目 录

第一章 数字逻辑基础	1
1.1 数制及码制	1
1.1.1 模拟量与数字量	1
1.1.2 数制及其转换	2
1.1.3 码制	6
1.1.4 算术运算和逻辑运算	8
1.2 逻辑代数	9
1.2.1 基本逻辑运算	9
1.2.2 逻辑代数的基本定律	16
1.2.3 逻辑代数的基本规则	18
1.2.4 常用公式	20
1.3 逻辑函数的表示方法	22
1.3.1 不同的表示方法及其转换	22
1.3.2 逻辑函数的两种标准形式	25
1.3.3 逻辑函数的常见表达式	29
1.4 逻辑函数的简化	29
1.4.1 公式化简法	29
1.4.2 卡诺图化简法	31
1.4.3 有无关项逻辑函数的化简	38
本章小结	40
思考题	40
习题	41
第二章 集成逻辑门电路	45
2.1 半导体器件的开关特性	45
2.1.1 晶体二极管的开关特性	45
2.1.2 晶体三极管的开关特性	48
2.1.3 MOS管的开关特性	51
2.2 简单分立元件逻辑门电路	54
2.2.1 二极管门电路	54
2.2.2 三极管门电路	56
2.3 TTL集成逻辑门电路	58
2.3.1 TTL与非门的电路结构和工作原理	58
2.3.2 TTL与非门主要外部特性	61
2.3.3 TTL其他类型门电路	69
2.3.4 TTL改进系列门电路	74
2.4 CMOS逻辑门电路	77

2.4.1	CMOS 反相器工作原理	77
2.4.2	CMOS 反相器主要特性	78
2.4.3	CMOS 其他类型门电路	81
2.4.4	CMOS 电路的正确使用	85
本章小结	86
思考题	86
习题	87
第三章	组合逻辑电路	91
3.1	组合逻辑电路的特点	91
3.2	SSI 器件构成的组合逻辑电路的分析	92
3.2.1	分析步骤	92
3.2.2	分析举例	92
3.3	常用的 MSI 组合逻辑电路	94
3.3.1	加法器	94
3.3.2	编码器	97
3.3.3	译码器	101
3.3.4	数据选择器	109
3.3.5	数值比较器	113
3.4	组合逻辑电路的设计	116
3.4.1	采用 SSI 器件设计组合逻辑电路	116
3.4.2	采用 MSI 器件实现组合逻辑函数	120
3.5	组合逻辑电路中的竞争 - 冒险现象	126
3.5.1	竞争 - 冒险现象形成的原因	126
3.5.2	冒险的判断及消除方法	129
本章小结	132
思考题	133
习题	133
第四章	集成触发器	137
4.1	基本 RS 触发器	137
4.1.1	基本 RS 触发器的电路结构与工作原理	137
4.1.2	逻辑功能描述方法	139
4.2	钟控触发器	141
4.2.1	钟控 RS 触发器	141
4.2.2	钟控 D 触发器	143
4.2.3	钟控触发方式的空翻现象	144
4.3	主从触发器	145
4.3.1	主从 RS 触发器	145
4.3.2	主从 JK 触发器	146
4.3.3	主从触发器的特点	150
4.4	边沿触发器	150
4.4.1	维持 - 阻塞 D 触发器	150
4.4.2	边沿 JK 触发器	152
4.4.3	边沿 T 触发器	154

4.4.4 CMOS 边沿触发器	156
本章小结	157
思考题	158
习题	158
第五章 时序逻辑电路	162
5.1 时序逻辑电路的特点与分类	162
5.2 SSI 器件构成的时序逻辑电路的分析	163
5.2.1 分析步骤	163
5.2.2 分析举例	163
5.3 常用的时序逻辑电路	167
5.3.1 寄存器	167
5.3.2 移位寄存器	169
5.3.3 计数器	176
5.3.4 移存型计数器	190
5.4 时序逻辑电路的设计	193
5.4.1 一般设计步骤	194
5.4.2 采用 SSI 器件设计同步时序电路	194
5.4.3 采用 SSI 器件设计异步时序电路	199
5.4.4 采用 MSI 计数器件设计任意模值计数器	203
5.4.5 序列信号发生器	209
5.4.6 顺序脉冲发生器	212
本章小结	213
思考题	214
习题	214
第六章 脉冲产生电路	219
6.1 矩形脉冲基本特性	219
6.2 脉冲电路	219
6.3 555 定时器电路结构及功能	220
6.4 施密特触发器	221
6.4.1 门电路构成的施密特触发器	221
6.4.2 集成施密特触发器	223
6.4.3 555 定时器构成的施密特触发器	224
6.4.4 施密特触发器的应用	224
6.5 单稳态触发器	225
6.5.1 门电路构成的单稳态触发器	225
6.5.2 集成单稳态触发器	228
6.5.3 555 定时器构成的单稳态触发器	231
6.5.4 单稳态触发器的应用	232
6.6 多谐振荡器	233
6.6.1 电容正反馈多谐振荡器	234
6.6.2 环形振荡器	235
6.6.3 石英晶体多谐振荡器	236
6.6.4 由施密特触发器构成的多谐振荡器	237

6.6.5	555 定时器构成的多谐振荡器	238
6.6.6	多谐振荡器的应用	239
	本章小结	240
	思考题	240
	习题	241
第七章	半导体存储器	244
7.1	半导体存储器概述	244
7.1.1	半导体存储器的特点	244
7.1.2	半导体存储器的分类	244
7.1.3	半导体存储器的主要技术参数	245
7.2	随机存取存储器(RAM)	246
7.2.1	RAM 的结构	246
7.2.2	静态 RAM(SRAM)	246
7.2.3	动态 RAM(DRAM)	248
7.3	只读存储器(ROM)	249
7.3.1	固定 ROM	249
7.3.2	可编程 ROM(PROM)	251
7.3.3	可擦除可编程 ROM(EPROM)	252
7.3.4	电可擦除可编程 ROM(EEPROM)	253
7.3.5	快闪存储器(Flash Memory)	254
7.4	存储器容量的扩展	255
7.4.1	位扩展	255
7.4.2	字扩展	256
7.5	用 ROM 存储器实现组合逻辑函数	258
	本章小结	260
	思考题	261
	习题	261
第八章	可编程逻辑器件	263
8.1	可编程逻辑器件(PLD)概述	263
8.1.1	PLD 的基本结构	263
8.1.2	PLD 的分类	264
8.2	可编程逻辑阵列器件(PLA)	265
8.2.1	PLA 的基本结构	265
8.2.2	PLA 的应用	266
8.3	可编程阵列逻辑器件(PAL)	267
8.3.1	PAL 的基本结构	267
8.3.2	PAL 的应用	270
8.4	通用阵列逻辑器件(GAL)	272
8.4.1	GAL 的基本结构	272
8.4.2	GAL 的输出逻辑宏单元(OLMC)	274
8.4.3	GAL 的应用	278
8.5	复杂可编程逻辑器件(CPLD)	279
8.5.1	CPLD 的基本结构	279

8.5.2 CPLD的通用逻辑块(GLB)	281
8.6 现场可编程门阵列器件(FPGA)	282
8.6.1 FPGA的基本结构	282
8.6.2 FPGA的CLB模块和IOB模块	283
8.6.3 FPGA内部互连资源(IR)	285
8.7 可编程逻辑器件的编程	286
本章小结	288
思考题	289
习题	289
第九章 硬件描述语言 VHDL	290
9.1 硬件描述语言概述	290
9.2 VHDL的基本结构	291
9.2.1 VHDL的基本程序结构	291
9.2.2 VHDL实体、结构体、配置、程序包及库	292
9.3 VHDL的基本元素	296
9.3.1 关键字及标志符	296
9.3.2 数据类型	297
9.3.3 数据对象	298
9.3.4 表达式与运算符	299
9.4 VHDL语句	300
9.4.1 顺序语句	300
9.4.2 并行语句	304
9.5 VHDL设计举例	307
9.5.1 用VHDL设计组合逻辑电路	307
9.5.2 用VHDL设计时序电路	310
本章小结	312
思考题	313
习题	313
第十章 数字电路课程设计	315
10.1 概述	315
10.2 数字电路课程设计的基本步骤	315
10.3 典型数字电路课程设计举例	316
10.3.1 数字式竞赛抢答器	316
10.3.2 四路彩灯显示系统	319
10.3.3 三位数字显示计时系统	321
附录 A 国产半导体集成电路型号命名法	323
附录 B 常用逻辑图形符号对照表	325
参考文献	327

第一章

数字逻辑基础

1.1 数制及码制

1.1.1 模拟量与数字量

在自然界中,存在着形形色色的物理量,尽管它们的性质各异,但就其变化规律的特点而言,可分为两大类:模拟量和数字量。

模拟量:在时间和数值上都具有连续变化特点的物理量叫做模拟量。自然界广泛存在着许多物理量都是模拟量,如温度、压力、距离、时间等。

模拟信号:表示模拟量的电信号叫做模拟信号。在工程应用中,为了测量、传递和处理这些物理量,常把它们通过传感器转换成与之成比例的电压值(或电流值),这些时间连续、幅值也连续的电信号表示和模拟了实际的物理量。例如:正弦波信号、话音信号等就是典型的模拟信号。

模拟电路:工作在模拟信号下的电子电路称为模拟电路。

数字量:在时间和数量上的取值是不连续的、离散的,只能按有限个或可数的量化单位取值,这类物理量叫做数字量。例如:某一实际距离的值为 $3869.82526\cdots\text{km}$,若取量化单位为 1 km ,则代表此距离的数字量为 3870 km ,若量化单位为 1 m ,则数字量为 $3\ 869\ 825\text{ m}$ 。量化单位的选择取决于所要求的精度。

数字信号:表示数字量的信号称为数字信号。数字信号是一种脉冲信号(Pulse Signal),脉冲信号具有边沿陡峭、持续时间短的特点。广义讲,凡是非正弦波形状的信号都可称为脉冲信号。例如:矩形波、方波、锯齿波等信号就是典型的数字信号。

数字电路:处理数字信号的电路称为数字电路。

同一物理量可以用连续的模拟信号表示,也可用离散的数字信号表示。同模拟信号相比,数字信号具有传输可靠、易于存储、抗干扰能力强、稳定性好等优点。因此,数字电路的应用愈来愈广泛。

在数字电路中,只采用 0 、 1 两种数字表示数字信号,一个 0 或一个 1 通常称为 1 bit ,有时也将一个 0 或一个 1 的持续时间称为一拍。“ 0 ”在数字电路中可代表低电平、开关的闭合,也可代表无脉冲信号等;“ 1 ”可代表高电平、开关的断开,也可代表有脉冲信号等。

数字电路中把只由高、低两种逻辑电平组成的信号称为数字信号,或数字逻辑信号,这种信号只能由数字电路进行处理。注意,数字逻辑信号不同于数字信号处理中所说的数字信号。对

于数字信号处理系统来说,数字信号是一组离散数据,可通过运算对其进行任何处理。

1.1.2 数制及其转换

1. 数制

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计数进位制,简称数制(Number System)。日常生活中最常用的是十进制,数字电路及计算机等设备中还经常使用二进制、八进制和十六进制。对于任何一个数,可以用不同的进制来表示。

1) 十进制(Decimal)

在十进制中,采用0~9十个数码,任何一个十进制数都可以用这十个数码按一定规律并列在一起来表示,计数规则为“逢十进一,借一当十”。例如,十进制数749.25可表示成

$$749.25 = 7 \times 10^2 + 4 \times 10^1 + 9 \times 10^0 + 2 \times 10^{-1} + 5 \times 10^{-2}$$

上式中的 10^2 、 10^1 、 10^0 …称为十进制数数位的位权值。十进制数各个数位的位权值是10的幂。“10”称为十进制数的基数。

对于任意一个十进制数 N ,均可按位权展开为

$$\begin{aligned} (N)_{10} &= (a_{n-1}a_{n-2}\cdots a_1a_0 \cdot a_{-1}a_{-2}\cdots a_{-m})_{10} \\ &= a_{n-1} \times 10^{n-1} + a_{n-2} \times 10^{n-2} + \cdots + a_1 \times 10^1 + a_0 \times 10^0 \\ &\quad + a_{-1} \times 10^{-1} + a_{-2} \times 10^{-2} + \cdots + a_{-m} \times 10^{-m} \\ &= \sum_{i=-m}^{n-1} a_i \times 10^i \end{aligned} \quad (1-1)$$

这种表示方法称为多项式表示法或按位权展开式。上式中, a_i 为十进制数第 i 位的数码,它可以是0~9这十个数码中的任意一个; n 表示整数部分的位数, m 表示小数部分的位数,因此 i 包含从 $n-1 \sim 0$ 的所有正整数和从 $-1 \sim -m$ 的所有负整数。一般可用下角标10或D表示十进制数,如 $(12)_{10}$ 、 $(20)_D$ 等。

若以 R 取代式(1-1)中的10,可得到任意 R 进制数的位权展开式为

$$\begin{aligned} (N)_R &= (a_{n-1}a_{n-2}\cdots a_1a_0 \cdot a_{-1}a_{-2}\cdots a_{-m})_R \\ &= a_{n-1} \times R^{n-1} + a_{n-2} \times R^{n-2} + \cdots + a_1 \times R^1 + a_0 \times R^0 + a_{-1} \times R^{-1} + a_{-2} \times R^{-2} \\ &\quad + \cdots + a_{-m} \times R^{-m} = \sum_{i=-m}^{n-1} a_i \times R^i \end{aligned} \quad (1-2)$$

式中, a_i 为 R 进制数第 i 位的数码; R^i 为 R 进制数第 i 位的位权值。 R 称为计数制的基数或称为计数的模(mod),一般用下角标 R 表示数 N 是 R 进制。

2) 二进制(Binary)

在二进制中,只采用0和1两个数码,计数规则为“逢二进一,借一当二”。二进制的基数为2,每个数位的位权值为2的幂。任意一个二进制数的位权展开式为

$$(N)_2 = \sum_{i=-m}^{n-1} a_i \times 2^i \quad (1-3)$$

式中, a_i 为第 i 位的0或1数码; 2^i 为第 i 位的位权值。例如,二进制数11101.101按位权展开式为

$$(11101.101)_2 = 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3}$$

二进制数 N 一般用下角标 2 或 B 表示, 如 $(101)_2$ 、 $(110.1)_B$ 等。

3) 八进制(Octal)

在八进制中, 采用 0~7 八个数码, 计数规则为“逢八进一, 借一当八”。八进制的基数为 8, 其位权展开式为

$$(N)_8 = \sum_{i=-m}^{n-1} a_i \times 8^i \quad (1-4)$$

八进制数 N 一般用下角标 8 或 O 表示, 如 $(76)_8$ 、 $(35.1)_O$ 等。

4) 十六进制(Hexadecimal)

在十六进制中, 采用 0~9、A(10)、B(11)、C(12)、D(13)、E(14)、F(15) 共十六个数码, 计数规则为“逢十六进一, 借一当十六”。十六进制的基数为 16, 其位权展开式为

$$(N)_{16} = \sum_{i=-m}^{n-1} a_i \times 16^i \quad (1-5)$$

十六进制数 N 一般用下角标 16 或 H 表示, 如 $(E12)_{16}$ 、 $(2B)_H$ 等。

2. 不同数制的转换

1) R 进制—十进制转换

将 R 进制(R 为二、八、十六)数转换为等值的十进制数, 其步骤为

- (1) 将 R 进制数按位权展开, 见式(1-2);
- (2) 将展开式按十进制运算规则相加, 即可得到等值的十进制数。

【例 1-1】 将二进制数 $(11011.101)_2$ 转换成等值的十进制数。

解: 二进制数

1	1	0	1	1.	1	0	1	
位权	2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}

$$\begin{aligned} (11011.101)_2 &= 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} \\ &= 16 + 8 + 0 + 2 + 1 + 0.5 + 0 + 0.125 = (27.625)_{10} \end{aligned}$$

【例 1-2】 将八进制数 $(157.304)_8$ 转换成等值的十进制数。

解: $(157.304)_8 = 1 \times 8^2 + 5 \times 8^1 + 7 \times 8^0 + 3 \times 8^{-1} + 0 \times 8^{-2} + 4 \times 8^{-3}$
 $= 64 + 40 + 7 + 0.375 + 0.0078125 = (111.3828125)_{10}$

【例 1-3】 将十六进制数 $(F4.C)_{16}$ 转换成等值的十进制数。

解: $(F4.C)_{16} = 15 \times 16^1 + 4 \times 16^0 + 12 \times 16^{-1} = 240 + 4 + 0.75 = (244.75)_{10}$

2) 十进制—R 进制转换

将十进制数转换为等值的 R (R 为二、八、十六)进制数, 需将十进制数的整数部分和小数部分分别进行转换, 然后再将它们合并起来。

整数部分转换时, 采用除基取余法, 具体步骤如下:

(1) 将十进制整数除以 R 进制的基 R , 并对每次得到的商再依次除以 R , 直到商等于 0 为止。

(2) 将每次得到的余数按倒序写出来, 即第一次的余数作为 R 进制整数的最低有效位 (Least Significant Bit, LSB), 最后一位的余数作为 R 进制整数的最高有效位 (Most Significant Bit, MSB), 所得数值即为等值 R 进制整数。

【例 1-4】 将十进制数 $(83)_{10}$ 转换成等值的二进制数。

解：将十进制数 83 依次除以二进制数的基数 2，并取其余数，转换过程如下：

2 $\overline{)83}$	商	余数	
2 $\overline{)41}$	←	1
2 $\overline{)20}$		1
2 $\overline{)10}$		0
2 $\overline{)5}$		0
2 $\overline{)2}$		1
2 $\overline{)1}$		0
0		1

↑ LSB
 ↓ MSB

因此

$$(83)_{10} = (1010011)_2$$

【例 1-5】 将十进制数 $(93)_{10}$ 转换成等值的十六进制数。

解：将十进制数 93 依次除以十六进制数的基数 16，并取其余数，转换过程如下：

16 $\overline{)93}$	商	余数	
16 $\overline{)5}$	←	D
0		5

↑
 ↓

因此

$$(93)_{10} = (5D)_{16}$$

十进制小数部分转换时，采用乘基取整法，即将十进制小数依次乘以 R ，取每次得到的乘积的整数部分构成十进制小数的各位数，直到小数部分为 0 或达到一定的精度为止。第一次乘积的整数作为二进制小数的最高有效位，最后一次乘积的整数作为二进制小数的最低有效位。

【例 1-6】 将十进制数 $(0.375)_{10}$ 转换成二进制数。

解：	$0.375 \times 2 = 0.75$	取整：	$b_{-1} = 0$	↓	MSB
	$0.75 \times 2 = 1.5$		$b_{-2} = 1$	↓	
	$0.5 \times 2 = 1.0$		$b_{-3} = 1$	↓	LSB

b_i 表示小数点后第 i 次乘积的整数部分。因此

$$(0.375)_{10} = (0.011)_2$$

有整数和小数的十进制数转换成 R 进制数时，将整数和小数部分分别进行转换，然后将结果合并起来。

例如，十进制数 $(83.375)_{10}$ 转换为二进制数时，综合例 1-4 和例 1-6 的转换结果，可得

$$(83.375)_{10} = (1010011.011)_2$$

十进制小数部分的转换有一个精度问题，不可能准确地完全转换，只要满足所要求的精度即可。

【例 1-7】 将十进制数 $(0.46)_{10}$ 转换成二进制数。

(1) 要求转换误差不大于 2^{-8} ；

(2) 要求精度达到 0.1%。

解：(1) 要求误差不大于 2^{-8} ，只需保留至小数点后第八位，计算过程如下：

$$\begin{array}{llll}
 0.46 \times 2 = 0.92 & b_{-1} = 0 & 0.92 \times 2 = 1.84 & b_{-2} = 1 \\
 0.84 \times 2 = 1.68 & b_{-3} = 1 & 0.68 \times 2 = 1.36 & b_{-4} = 1
 \end{array}$$

$$\begin{array}{ll} 0.36 \times 2 = 0.72 & b_{-5} = 0 \\ 0.44 \times 2 = 0.88 & b_{-7} = 0 \end{array} \quad \begin{array}{ll} 0.72 \times 2 = 1.44 & b_{-6} = 1 \\ 0.88 \times 2 = 1.76 & b_{-8} = 1 \end{array}$$

因此

$$(0.46)_{10} \approx (0.01110101)_2$$

(2) 由于二进制数的小数点后第九位为 $2^{-9} = 1/512 \approx 0.2\%$, 第十位为 $2^{-10} = 1/1024 < 0.1\%$, 所以要达到 0.1% 的精度, 需保留至小数点后第十位。

接(1)的计算过程, 有

$$\begin{array}{ll} 0.76 \times 2 = 1.52 & b_{-9} = 1 \\ 0.52 \times 2 = 1.04 & b_{-10} = 1 \end{array}$$

因此

$$(0.46)_{10} \approx (0.0111010111)_2$$

3) 二进制—八进制、八进制—二进制转换

二进制数转换为八进制数时, 由于三位二进制数恰好有八个状态, 所以将三位二进制数直接用一位八进制数代替。划分原则为: 以小数点为中心, 整数部分从低到高每三位一组, 最高位不足三位其前添零补齐; 小数部分从高到低每三位一组, 最低位不足三位其后添零补齐。

八进制数转换为二进制数时, 将每位八进制数直接展开成三位二进制数即可。

4) 二进制—十六进制、十六进制—二进制转换

二进制数转换为十六进制数时, 由于四位二进制数恰好有十六个状态, 所以将四位二进制数直接用一位十六进制数代替。划分原则为: 以小数点为中心, 整数部分从低到高每四位一组, 最高位不足四位其前添零补齐; 小数部分从高到低每四位一组, 最低位不足四位其后添零补齐。

十六进制数转换为二进制数时, 将每位十六进制数直接展开成四位二进制数即可。

5) 八进制—十六进制、十六进制—八进制转换

八进制数转换为十六进制数时, 以二进制为桥梁, 先将八进制数转换为二进制数, 再将二进制数转换为十六进制数。

同理, 十六进制数转换为八进制数时, 先将十六进制数转换为二进制数, 再将二进制数转换为八进制数。

【例 1-8】 将二进制数 $(10011101100.0011101111)_2$ 分别转换成八进制数和十六进制数。

解: 转换过程如下:

二进制数	<u>010</u>	<u>011</u>	<u>101</u>	<u>100.</u>	<u>001</u>	<u>110</u>	<u>111</u>
八进制数	2	3	5	4.	1	6	7

因此

$$(10011101100.0011101111)_2 = (2354.167)_8$$

二进制数	<u>0100</u>	<u>1110</u>	<u>1100.</u>	<u>0011</u>	<u>1011</u>	<u>1000</u>
十六进制数	4	E	C.	3	B	8

因此

$$(10011101100.0011101111)_2 = (4EC.3B8)_{16}$$

【例 1-9】 将十六进制数 $(BE.29D)_{16}$ 转换成八进制数。

解：转换过程如下：

$$\begin{array}{rcccccc}
 (BE.29D)_{16} = & (1011 & 1110. & 0010 & 1001 & 1101)_2 \\
 \text{十六进制数} & B & E & . & 2 & 9 & D \\
 \text{二进制数} & & 010 & 111 & 110. & 001 & 010 & 011 & 101 \\
 \text{八进制数} & & 2 & 7 & 6 & . & 1 & 2 & 3 & 5
 \end{array}$$

因此

$$(BE.29D)_{16} = (276.1235)_8$$

1.1.3 码制

数码不仅可以表示数量上的大小，而且还可用来表示特定的事物。例如“865”路公交车，学号 060016 等，这些数码已没有表示数量大小的含意，只是一种人们事先约定而赋予特定事物的代号。这种类型的数码称为代码。

在编制代码时所遵循的规则称为码制。

1. 二—十进制代码(BCD 代码)

在数字系统中，常用 0、1 两种数码的组合作为代码，称为二进制码。二进制码可以是多位数的，若用 4 位二进制码表示 1 位十进制数的代码，称为二—十进制代码，简称 BCD (Binary Coded Decimal) 代码。BCD 代码是用 4 位二进制码的 10 种组合表示十进制数 0~9 十个数码。

4 位二进制码有 $2^4 = 16$ 种组合，当用这些组合表示十进制数 0~9 时，需在 16 种组合中选用 10 种组合，表 1.1 列出了几种常用的 BCD 代码。

表 1.1 常用 BCD 代码

十进制数	8421 码	余 3 码	2421 码	5421 码	631-1 码	余 3 循环码	移存码
0	0000	0011	0000	0000	0011	0010	0001
1	0001	0100	0001	0001	0010	0110	0010
2	0010	0101	0010	0010	0101	0111	0100
3	0011	0110	0011	0011	0111	0101	1001
4	0100	0111	0100	0100	0110	0100	0011
5	0101	1000	1011	1000	1001	1100	0111
6	0110	1001	1100	1001	1000	1101	1111
7	0111	1010	1101	1010	1010	1111	1110
8	1000	1011	1110	1011	1101	1110	1100
9	1001	1100	1111	1100	1100	1010	1000

BCD 代码分为有权码和无权码两类。

1) 有权 BCD 码

有权 BCD 码是指 4 位二进制数码都有确定的位权值。如表 1.1 中的 8421 码、2421 码、5421 码、631-1 码等。对于有权 BCD 码，可根据位权展开求得所代表的十进制数。

$$\text{例如：} [1001]_{8421\text{码}} = 1 \times 8 + 0 \times 4 + 0 \times 2 + 1 \times 1 = (9)_{10}$$

$$[1110]_{2421\text{码}} = 1 \times 2 + 1 \times 4 + 1 \times 2 + 0 \times 1 = (8)_{10}$$

$$[0111]_{631-1\text{码}} = 0 \times 6 + 1 \times 3 + 1 \times 1 + 1 \times (-1) = (3)_{10}$$

最常用的有权BCD码是8421码,其位权值与自然二进制数的位权值8、4、2、1一致,所以也称为自然权码。

2421码各位的权依次为2、4、2、1,其组成特点是,2421码的前5个码与8421码一致,后5个码是8421码加上 $(6)_{10} = (0110)_2$ 得到。另一显著特点是,将任意一个十进制数D的2421码的各位取反,正好是与9互补的那个十进制数 $(9-D)$ 的代码。例如,将4的2421代码0100取反,得到的代码1011,与 $9-4=5$ 的2421码1011一致,即4和5、0和9、1和8、2和7、3和6互为反码。这种特性称为自补特性。余3码、631-1码等也具有自补特性,这在数字系统中非常有用。

5421码各位的权依次为5、4、2、1,其组成特点是,5421码的前5个码与8421码一致,后5个码是8421码加上 $(3)_{10} = (0011)_2$ 得到。另一特点是最高位连续5个0后连续5个1。当计数器采用这种代码时,最高位可产生对称方波输出。

2) 无权BCD码

无权BCD码没有确定的位权值,不能按位权展开来求所代表的十进制数。如表1.1中的余3码、余3循环码、移存码等。但这些代码各有其特点,可应用于不同场合。

余3码是每个8421码加上 $(3)_{10} = (0011)_2$ 得到的。用余3码进行加减运算比8421码方便。余3循环码的两个相邻代码仅有一个数码不同,利用这种特性设计的计数器不会发生冒险现象。

3) 用BCD码表示十进制数

在BCD代码中,4位二进制代码仅表示一位十进制数,对一个多位十进制数编制代码,需要用与十进制位数相同的几组BCD代码来表示。

【例1-10】用8421码、2421码、余3码分别表示十进制数869。

解: $[869]_{10} = [1000\ 0110\ 1001]_{8421\text{码}} = [1011\ 1001\ 1100]_{\text{余3码}}$
 $= [1110\ 1100\ 1111]_{2421\text{码}}$

如果用8421码表示R进制数,应先将R进制数转换成十进制数,再用相应的几组BCD码表示出来。例如: $(1101.1)_2 = (13.5)_{10} = (0001\ 0011.0101)_{8421\text{码}}$ 。

2. 格雷码

格雷码(Gray Code)是另一种无权码,表1.2列出了一种典型的四位格雷码与相应的十进制码以及二进制码的对应关系。

表 1.2 四位格雷码与十进制码及二进制码的对应关系

十进制码	二进制码	格雷码	十进制码	二进制码	格雷码
0	0000	0000	8	1000	1100
1	0001	0001	9	1001	1101
2	0010	0011	10	1010	1111
3	0011	0010	11	1011	1110
4	0100	0110	12	1100	1010
5	0101	0111	13	1101	1011
6	0110	0101	14	1110	1001
7	0111	0100	15	1111	1000