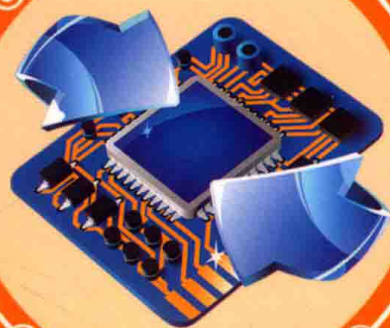


新工科人才培养·电气信息类应用型系列规划教材

# 数字电子技术与仿真

SHUZI DIANZI JISHU YU FANGZHEN

刘海珊 李洪芹◎主编



中国铁道出版社有限公司  
CHINA RAILWAY PUBLISHING HOUSE CO., LTD.

新工科人才培养·电气信息类应用型系列规划教材

# 数字电子技术与仿真

刘海珊 李洪芹◎主编

中国铁道出版社有限公司  
CHINA RAILWAY PUBLISHING HOUSE CO., LTD.

## 内 容 简 介

本书采用先“逻辑”后“电路”的次序,首先讲解数制、码制和逻辑代数等基础知识,接着重点讲解组合逻辑电路和时序逻辑电路的分析与设计方法,再介绍触发器、时序逻辑电路的分析与设计、门电路与半导体存储器的电路结构、工作原理,数字电子电路及系统设计,并采用当今数字设计实验验证的新方法(Proteus 仿真软件)来描述数字电路。全书内容安排循序渐进,由浅入深,并设有适量例题帮助读者理解和掌握重点、难点。

本书适合作为高等院校电气信息类、机电类、轨道交通等专业的教材,也可作为电气工程技术人员及相关领域工程师的参考书。

### 图书在版编目(CIP)数据

数字电子技术与仿真/刘海珊,李洪芹主编. —北京:  
中国铁道出版社有限公司, 2021. 1  
新工科人才培养·电气信息类应用型系列规划教材  
ISBN 978-7-113-27456-6

I. ①数… II. ①刘…②李… 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 (2020) 第 240745 号

书 名: 数字电子技术与仿真  
作 者: 刘海珊 李洪芹

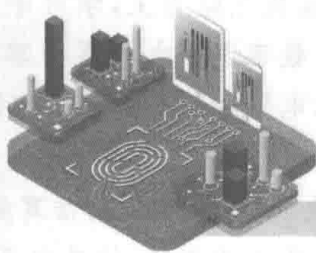
策 划: 曹莉群  
责任编辑: 陆慧萍 绳 超  
封面设计: 刘 莎  
责任校对: 张玉华  
责任印制: 樊启鹏

编辑部电话: (010) 63549508

出版发行: 中国铁道出版社有限公司 (100054, 北京市西城区右安门西街 8 号)  
网 址: <http://www.tdpress.com/51eds/>  
印 刷: 国铁印务有限公司  
版 次: 2021 年 1 月第 1 版 2021 年 1 月第 1 次印刷  
开 本: 787 mm×1 092 mm 1/16 印张: 15.25 字数: 389 千  
书 号: ISBN 978-7-113-27456-6  
定 价: 42.00 元

版权所有 侵权必究

凡购买铁道版图书,如有印制质量问题,请与本社教材图书营销部联系调换。电话:(010) 63550836  
打击盗版举报电话:(010) 63549461



# 前言

为了积极推进新工科建设下的基础教学改革,按照教育部《普通高等学校本科专业类教学质量国家标准》对数字电子技术理论教学的基本要求,结合自动化、电气工程及其自动化、电子信息工程、轨道交通信号与控制、材料工程等专业对数字电子技术的课程要求,联合业内具有丰富数字电子技术教学经验和实践应用能力的教师,专门编写了本书。

本书主要包括逻辑代数基础、组合逻辑电路、触发器、时序逻辑电路、门电路与半导体存储器、数字电子电路及系统设计、基于 Proteus 的数字电子技术仿真实验等内容。本书内容简明易学,以帮助初学者比较轻松地掌握基本内容,特别适合用于翻转式或混合式教学,也适合相关专业的自学者自学。

本书主要特色:

## 1. 结构清晰

全书各章按照知识点编排,既相互独立,又相互联系,力求顺序合理,逻辑性强,可读性强,使学生更易学习和掌握。

## 2. 突出虚实结合

为了进一步丰富教学内容并增强实用性,特别引入了基于 Proteus 的数字电子技术仿真实验,软硬结合,注重能力的培养,这也使学习变得生动有趣。第 6 章数字电子电路及系统设计和第 7 章基于 Proteus 的数字电子技术仿真实验的内容突出工程设计及应用。先介绍数字电路设计的基本要求,然后给出数字电路设计的步骤及实例。每个实验给出完整的实验要求和设计过程。学生可以按照书中所讲述的内容操作,顺利完成仿真任务。实验过程非常接近实际操作的效果,便于培养学生的创新能力及工程实际的应用能力。

## 3. 理论联系实际

本书为“新工科人才培养·电气信息类应用型系列规划教材”之一。其内容设计简洁,同时拓展考研相关的知识点,适当引入工程应用内容以突出针对性和实用性,帮助学生提升理解和分析能力。书中安排了适量的例题和习题,以帮助学生进一步理解和掌握相关的知识点。题型多样,具有一定的启发性、灵活性和实践性。浓缩教学内容,用言简意赅的语言精讲要点、重难点,便于学生理解记忆。

本书适用于高等院校电气信息类、机电类、轨道交通等专业数字电子技术基础课程的教学。而数字电子技术课程在教学要求中是“入门性质的技术基础课”,课程的主要任务是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并在其相关专业中的应用打下基础。本课程学习过程中可从以下几方面入手:

(1) 数字电子技术的数学基础是逻辑代数,此部分与其他课程的关联较少,必须要掌握逻辑代数的基本知识、理论,如数制和码制等,这是本书最基本的内容。

(2)数字电路中以集成电路为主,学习中重点掌握各集成器件的逻辑功能及输入和输出特性。但也不必对所有数字逻辑电路逐一掌握,只需要学会分析、设计数字逻辑电路的一般方法。尤其是组合逻辑电路和时序逻辑电路的分析方法和设计方法,这是数字电子技术课程的核心内容。

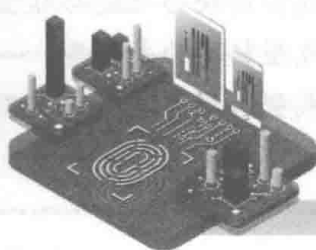
(3)结合 Proteus 软件,重视相关的仿真实验。通过仿真实验,进一步加深对书中基本电路、基本分析方法、基本理论的理解,培养电子电路实际操作能力。软件仿真便于较快地明确目标,节省时间,不受实验设备、场地的限制。在利用软件对数字电子电路进行辅助设计时,还可进一步通过实验操作和硬件安装、调试,感受工程应用的特点,积累实践经验,提高实验能力。

本书由上海工程技术大学电气类电学基础教学团队编写,由刘海珊和李洪芹任主编,参与编写工作的还有田瑾、张振华、邹睿和高飞。在编写过程中,每位教师都将其多年的教学经验倾囊而出,力求做到条理清晰、语言准确、文字简洁、图表规范。本书在编写过程中也得到了同行专家的帮助,在此致以诚挚的感谢!

限于编者水平,书中难免存在不妥之处,恳请广大读者给予批评指正。

编者

2020年7月



# 目 录

<b>第 1 章 逻辑代数基础</b> .....	<b>1</b>
1.1 数制与码制 .....	2
1.1.1 常用的数制 .....	2
1.1.2 不同数制之间的转换 .....	4
1.1.3 码    制 .....	6
1.2 逻辑运算 .....	9
1.2.1 基本逻辑运算 .....	9
1.2.2 组合逻辑运算 .....	11
1.3 逻辑代数的基本公式和基本规则 .....	12
1.3.1 逻辑代数的基本公式 .....	12
1.3.2 基本规则 .....	13
1.4 逻辑函数的表示方法 .....	15
1.4.1 逻辑真值表 .....	15
1.4.2 逻辑函数式 .....	16
1.4.3 逻辑图 .....	16
1.4.4 波形图 .....	16
1.4.5 各种表示方法之间的相互转换 .....	16
1.5 逻辑函数表达式的两种标准形式 .....	18
1.5.1 最小项 .....	18
1.5.2 最大项 .....	18
1.5.3 最小项和最大项的性质 .....	19
1.5.4 逻辑函数的与或标准表达式和或与标准表达式 .....	20
1.5.5 逻辑函数的卡诺图表示法 .....	21
1.6 逻辑函数的化简 .....	25
1.6.1 同一逻辑函数表达式形式的多样性 .....	26
1.6.2 公式法化简 .....	26
1.6.3 卡诺图法化简 .....	28
1.6.4 具有无关项的逻辑函数表达式的化简 .....	30
1.7 逻辑函数表达式形式的变换 .....	33
1.7.1 五种类的逻辑函数表达式 .....	33

1.7.2	与或型转换为与非与非型 .....	34
1.7.3	与或型转换为或与型 .....	34
1.7.4	与或型转换为或非或非型 .....	35
1.7.5	与或型转换为与或非型 .....	35
小 结	.....	35
习 题	.....	36
<b>第 2 章</b>	<b>组合逻辑电路 .....</b>	<b>39</b>
2.1	组合逻辑电路分析 .....	40
2.2	组合逻辑电路设计 .....	44
2.3	编 码 器 .....	47
2.4	译 码 器 .....	51
2.4.1	二进制译码器——3 线-8 线译码器 .....	52
2.4.2	代码转换译码器——8421BCD 码(4 线-10 线)译码器 .....	56
2.4.3	显示译码器 .....	57
2.5	数据选择器 .....	60
2.6	加 法 器 .....	66
2.6.1	半 加 器 .....	66
2.6.2	全 加 器 .....	67
2.6.3	串行多位加法器 .....	67
2.7	数值比较器 .....	68
2.8	组合逻辑电路中的竞争-冒险现象 .....	69
小 结	.....	71
习 题	.....	71
<b>第 3 章</b>	<b>触 发 器 .....</b>	<b>76</b>
3.1	锁存器和双稳态触发器 .....	77
3.1.1	RS 锁存器 .....	77
3.1.2	钟控 RS 触发器 .....	78
3.1.3	边沿触发 D 触发器 .....	79
3.1.4	边沿触发 JK 触发器 .....	80
3.1.5	T 与 T' 触发器 .....	82
3.1.6	触发器之间的转换 .....	83
3.2	施密特触发器 .....	85
3.2.1	门电路组成的施密特触发器 .....	85
3.2.2	集成施密特触发器 .....	87
3.2.3	施密特触发器的应用 .....	88
3.3	单稳态触发器 .....	89

3.3.1	由门电路构成的微分型单稳态触发器	89
3.3.2	集成单稳态触发器	92
3.3.3	单稳态触发器的应用	95
3.4	多谐振荡器	97
3.5	555 定时器及其应用	98
3.5.1	555 定时器内部结构和工作原理	98
3.5.2	555 定时器构成的施密特触发器	101
3.5.3	555 定时器构成的多谐振荡器	102
3.5.4	555 定时器构成的单稳态触发器	103
3.5.5	555 时基集成电路的分类	104
3.5.6	基于 555 时基集成电路的实际应用举例	109
小 结		113
习 题		114
<b>第 4 章</b>	<b>时序逻辑电路</b>	<b>122</b>
4.1	时序逻辑电路概述	122
4.2	时序逻辑电路分析	124
4.2.1	同步时序逻辑电路分析	124
4.2.2	异步时序逻辑电路分析	130
4.3	同步时序逻辑电路设计	131
4.4	中规模集成时序逻辑电路	138
4.4.1	计数器	138
4.4.2	计数进制的改变	140
4.4.3	计数器的应用实例	144
4.5	寄存器	144
4.5.1	数码寄存器	145
4.5.2	移位寄存器	146
4.6	时序逻辑电路中的竞争-冒险现象	148
小 结		149
习 题		149
<b>第 5 章</b>	<b>门电路与半导体存储器</b>	<b>155</b>
5.1	门电路基础	155
5.1.1	半导体二极管门电路	156
5.1.2	半导体三极管门电路	158
5.1.3	集电极开路门电路	159
5.1.4	三态门	161
5.2	半导体存储器	163

5.2.1	只读存储器 .....	164
5.2.2	随机存储器 .....	166
5.2.3	存储器的扩展 .....	166
5.2.4	用存储器实现组合逻辑函数 .....	169
小 结	.....	171
习 题	.....	171
<b>第 6 章</b>	<b>数字电子电路及系统设计 .....</b>	<b>174</b>
6.1	典型数字电子系统的组成 .....	174
6.2	任意进制计数器设计 .....	176
6.2.1	$N$ 进制计数器设计要点 .....	176
6.2.2	计数电路设计 .....	176
6.3	555 时基电路及其应用 .....	183
6.3.1	555 时基电路的内部结构 .....	183
6.3.2	555 时基电路的基本工作模式 .....	184
6.3.3	555 集成电路的典型应用 .....	188
6.4	数字电子钟设计 .....	189
6.4.1	数字电子钟系统概述 .....	189
6.4.2	单元电路设计与分析 .....	190
6.5	智力竞赛抢答器设计 .....	195
6.5.1	智力竞赛抢答器系统概述 .....	195
6.5.2	智力竞赛抢答器系统原理和组成框图 .....	195
6.5.3	智力竞赛抢答器单元电路设计 .....	196
小 结	.....	203
习 题	.....	203
<b>第 7 章</b>	<b>基于 Proteus 的数字电子技术仿真实验 .....</b>	<b>205</b>
7.1	Proteus 简介 .....	206
7.2	Proteus 电路设计软件使用方法 .....	206
7.2.1	安装环境 .....	206
7.2.2	安装步骤 .....	207
7.2.3	Proteus 仿真实验界面 .....	208
7.3	典型数字电路仿真实验 .....	211
7.3.1	集成门电路逻辑功能测试实验 .....	211
7.3.2	组合逻辑电路仿真实验 .....	216
7.3.3	编码器及其应用实验 .....	218
7.3.4	译码器及其应用实验 .....	219
7.3.5	字段译码器逻辑功能测试及其应用实验 .....	220

7.3.6	半加器和全加器实验 .....	223
7.3.7	触发器及其应用实验 .....	225
7.3.8	时序逻辑电路设计实验 .....	228
7.3.9	计数、译码及显示电路实验 .....	229
7.3.10	集成移位寄存器应用实验 .....	230
小 结	.....	232
习 题	.....	232
附录 A	图形符号对照表.....	233
参考文献	.....	234

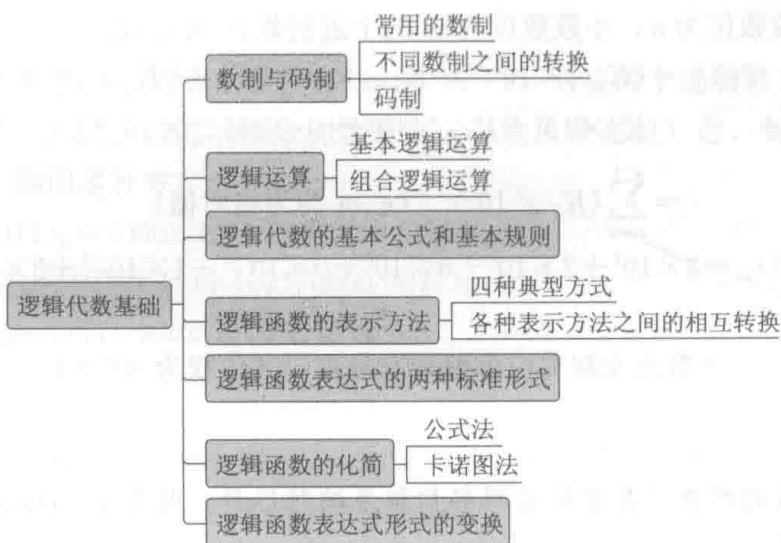
# 第 1 章

## 逻辑代数基础

### 引言

本章主要介绍数制和码制的基本概念和数字电路中常用的数制和编码，并重点讨论不同数制之间的转换方法。同时介绍了二进制数的基本运算和反码、补码的定义及运算。最后讨论了各种不同的编码方式。详细介绍了逻辑代数的基本运算、基本定律和基本运算规则，然后介绍逻辑函数的表示方法及逻辑函数的代数化简法和卡诺图化简法。逻辑代数有其自身独立的规律和运算法则，而不同于普通代数。

### 内容结构



### 学习目标

通过本章内容的学习，应该能够做到：

- (1) 熟练掌握数制和码制；
- (2) 灵活运用逻辑代数的基本公式和运算规则；
- (3) 学会逻辑函数的表示方法及其相互转换；
- (4) 熟练掌握逻辑函数表达式的两种标准形式；
- (5) 掌握逻辑函数的公式化简和卡诺图化简法，以及包含无关项的逻辑函数的化简；
- (6) 掌握逻辑函数表达式的四种类型及相互转换。



## 1.1 数制与码制

电子系统中的信号可以分为两大类：模拟量和数字量。模拟量是随时间连续变化的物理量。其特点是具有连续性。表示模拟量的信号称为模拟信号，工作在模拟信号下的电子电路称为模拟电路。

数字量是时间、幅值上不连续的物理量。其特点是具有离散性。表示数字量的信号称为数字信号，工作在数字信号下的电子电路称为数字电子电路。数字信号通常用数码来表示，数码可以通过数制表示数值的大小，如二进制码、十进制码和十六进制码等。用数码通过码制来表示不同的事物或事物的不同状态，码制是指不同的编码方式，如各种 BCD 码、循环码等。

### 1.1.1 常用的数制

#### 1. 十进制数

十进制是最常见、最广泛使用的一种数制，可表示数的基本数符为 0~9，是以 10 为基数的进位计数制。进位规则是“逢十进一，借一当十”。其中位权以基数为底，数位序数为指数的幂。

通常，一个整数数位为  $n$ ，小数数位为  $m$  的十进制数  $N$  可写成：

$$\begin{aligned} (N)_{10} &= (K_{n-1} \times 10^{n-1} + K_{n-2} \times 10^{n-2} + \dots + K_1 \times 10^1 + \\ &\quad K_0 \times 10^0 + K_{-1} \times 10^{-1} + \dots + K_{-m} \times 10^{-m}) \quad (1-1) \\ &= \sum_{i=m}^{n-1} (K_i \times 10^i) \quad (n, m \text{ 均为绝对值}) \end{aligned}$$

比如： $(328.013)_{10} = 3 \times 10^2 + 2 \times 10^1 + 8 \times 10^0 + 0 \times 10^{-1} + 1 \times 10^{-2} + 3 \times 10^{-3}$



小数点左起首位称整数部分位，其位权为  $10^0 = 1$



#### 注意：

考虑到十进制数的特点，其分析过程起始位置的特殊性，此方法又称位置计数法，即从小数点处开始左右分析。

一个数码的进制表示，可用数字下标来表示，如  $(N)_2$  表示二进制； $(N)_{10}$  表示十进制； $(N)_8$  表示八进制； $(N)_{16}$  表示十六进制。

有时也用字母作为下标，如  $(N)_B$  表示二进制，B 即 Binary； $(N)_D$  表示十进制，D 即 Decimal； $(N)_O$  表示八进制，O 即 Octal； $(N)_H$  表示十六进制，H 即 Hexadecimal。

#### 2. 二进制数

二进制数可表示数的基本数符为 0 或 1，是以 2 为基数的进位计数制。进位规则是“逢二进一，借一当二”。比如，两个一位二进制数 1 相加时，产生高位进位 1，其结果为二进制数 10。

$$\begin{array}{r} 1 \\ +1 \quad 1 \\ \hline 1 \quad 0 \end{array}$$

通常，一个整数数位为  $n$ ，小数数位为  $m$  的二进制数  $N$  可写成：

$$\begin{aligned} (N)_2 &= (B_{n-1} \times 2^{n-1} + B_{n-2} \times 2^{n-2} + \cdots + B_1 \times 2^1 + B_0 \times 2^0 + \\ &\quad B_{-1} \times 1^{-1} + \cdots + B_{-m} \times 1^{-m})_{10} \\ &= \sum_{i=m}^{n-1} (D_i \times 2^i) \end{aligned} \quad (1-2)$$

$$\begin{aligned} \text{例如：} (11011.11)_2 &= 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} \\ &= 16 + 8 + 0 + 2 + 1 + 0.5 + 0.25 = (27.75)_{10} \end{aligned}$$



### 拓展知识：

计算机存储器的基本存储单位是存储单元，每个单元存放一字节二进制数。存储器由许多存储单元组成，各存储单元给予编号，称为存储地址，采用若干字节二进制表示。例如，80X86CPU 处理的信息以字节为单位。

一字节表示 8 位二进制数，其数值范围是 00000000~11111111，即 0~255；

双字节表示 16 位二进制数，其数值范围是 00……00~11……11，即 0~65 535；

四字节表示 32 位二进制数，其存储范围为 0~ $2^{32}$ ，约 4 GB；

八字节表示 64 位二进制数，其存储范围为 0~ $2^{64}$ ，性能优于 32 位。

### 3. 十六进制数

十六进制数的基数是 16，可表示数符为 0~9、A~F，对应的十进制数为 0~15，是多位二进制数的一种简明表示形式。二进制整数从最低位（小数从最高位）起，每四位用一位十六进制数表示；若缺位，则用零补齐。

例如：(110110011)<sub>B</sub>=(0001 1011 0011)<sub>B</sub>=(1B3)<sub>H</sub>

(5FB.03)<sub>H</sub>=(0101 1111 1011.0000 0011)<sub>B</sub>

(F9AD)<sub>H</sub>=(1111 1001 1010 1101)<sub>B</sub>



### 注意：

目前在计算机上常用的是 8 位、16 位和 32 位二进制数表示和计算，由于 8 位、16 位和 32 位二进制数都可以用 2 位、4 位和 8 位十六进制数表示，故在编程时用十六进制书写非常方便。

不同进制数的对照表见表 1-1。

表 1-1 不同进制数的对照表

十进制数	二进制	八进制	十六进制
00	0000	00	0
01	0001	01	1
02	0010	02	2
03	0011	03	3
04	0100	04	4
05	0101	05	5
06	0110	06	6
07	0111	07	7

续表

十进制数	二进制	八进制	十六进制
08	1000	10	8
09	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

## 1.1.2 不同数制之间的转换

### 1. 十进制与二进制之间的相互转换

十进制可以与二进制、八进制、十六进制之间相互转换。

整数部分：将十进制整数转换为二进制采用除 2 取余法，再将所取得的余数倒排列就得到对应的二进制数。

即将十进制整数除以 2，得到一个商和一个余数；再将得到的商除以 2，又得到一个商和一个余数；以此类推，直至商等于 0 为止，再将所得到的余数倒排列就得到对应的二进制数。

例如：将  $(12)_{10}$  转换为二进制，整数部分采用除 2 取余的方法。

$$\begin{array}{r}
 2 \overline{) 12} \dots\dots \text{余数为 } 0 = k_0 \\
 2 \overline{) 6} \dots\dots \text{余数为 } 0 = k_1 \\
 2 \overline{) 3} \dots\dots \text{余数为 } 1 = k_2 \\
 2 \overline{) 1} \dots\dots \text{余数为 } 1 = k_3 \\
 \quad \quad \quad 0
 \end{array}
 \begin{array}{l}
 \uparrow \\
 \uparrow \\
 \uparrow \\
 \uparrow
 \end{array}$$

所以， $(12)_{10} = (1100)_2$ 。

小数部分：将十进制小数转换为二进制采用乘 2 取整法，再将所得到的整数部分顺排列就得到对应的二进制数。

即将十进制小数逐次乘以 2，以乘积到 1 为止（或到有效位数为止），再将每次所取得的积的整数部分按各自出现的顺序依次排列就得到对应的二进制数。

$$\begin{array}{r}
 0.8125 \\
 \times \quad 2 \\
 \hline
 1.6250 \dots\dots \text{整数部分为 } 1 = k_{-1} \\
 0.6250 \\
 \times \quad 2 \\
 \hline
 1.2500 \dots\dots \text{整数部分为 } 1 = k_{-2} \\
 0.2500 \\
 \times \quad 2 \\
 \hline
 0.5000 \dots\dots \text{整数部分为 } 0 = k_{-3}
 \end{array}$$

$$\begin{array}{r} 0.5000 \\ \times \quad 2 \\ \hline 1.000\cdots\cdots \end{array}$$

整数部分为  $1 = k_{-4}$

故  $(0.8125)_{10} = (0.1101)_2$ 。

一个十进制数既有整数又有小数，则分别转换后相加即可。参照上述计算结果则有：

$$(12.8125)_{10} = (1100.1101)_2$$



### 注意：

若小数乘 2 无法使尾数为零，则可根据精度要求求出足够位数。另外，无论整数部分还是小数部分， $k_i$  的取值顺序一定不能错。

## 2. 十进制与八进制、十六进制之间的相互转换

数制转换实质就是将一个数从一种进位制转化为等值的另一种进位制。比如，将十进制转换为八进制、十六进制，对于整数部分和小数部分分别采取与前述十进制转二进制相同的方法。整数部分转换除以基数 8 或 16，得到的余数几位  $k_i$ ，以此类推，反复将每次得到的商除以基数直到商为零，就可以得到整数部分的每一个系数。小数部分转换乘以基数 8 或 16，所得乘积的整数部分即为  $k_{-m}$ ，以此类推，将每次乘以基数得到的乘积的小数部分再乘以基数，直到小数部分为零；或小数部分不为零，但已满足误差要求进行“四舍五入”为止。从而求出八进制或十六进制小数部分的每一个系数。

反之，将二进制、八进制、十六进制转换为十进制，只需要按照前文所述内容按位权展开即可，求各位数值之和即可得到相应的十进制数。

## 3. 二进制与八进制之间的相互转换

从表 1-1 中可以看到，每位八进制数均可用三位二进制数表示。故而，以小数点为界，将二进制数的整数部分从低位开始，小数部分从高位开始，每三位一组，首（尾）位不足三位的补零，然后每组三位二进制数用一位八进制数表示。

**例 1.1** 将  $(11101.010101)_2$  转换为八进制。

$$\begin{array}{cccc} (011 & 101. & 010 & 101)_2 \\ \downarrow & \downarrow & \downarrow & \downarrow \\ = (3 & 5. & 2 & 5)_8 \end{array}$$

反之，若将八进制转换为二进制，见例 1.2。

**例 1.2** 将  $(62.45)_8$  转换为二进制。

$$\begin{array}{cccc} (6 & 2. & 4 & 5)_8 \\ \downarrow & \downarrow & \downarrow & \downarrow \\ = (110 & 010. & 100 & 101)_2 \end{array}$$

## 4. 二进制与十六进制之间的相互转换

由于二进制的位数比较多，不便于书写和记忆，通常可用十六进制数或者八进制数来描述二进制数。从表 1-1 中可以看到，每位十六进制数均可用四位二进制数表示。故而，以小数点为界，将二进制数的整数部分从低位开始，小数部分从高位开始，每四位一组，首位不足四位的补零，然后每组四位二进制数用一位十六进制数表示。

**例 1.3** 将  $(1011110.1011001)_2$  转换为十六进制。

$$\begin{array}{cccc} (0101 & 1110. & 1011 & 0010)_2 \\ \downarrow & \downarrow & \downarrow & \downarrow \\ = (5 & E. & B & 2)_H \end{array}$$

反之，若将十六进制转换为二进制，见例 1.4。

**例 1.4** 将  $(9F1C.04A)_H$  转换为二进制。

$$\begin{array}{ccccccc} ( & 9 & F & 1 & C & . & 0 & 4 & A )_H \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ = ( & 1001 & 1111 & 0001 & 1100 & . & 0000 & 0100 & 1010 )_2 \end{array}$$



### 注意:

- (1) 二进制转换为十进制：一般直接按照公式直接展开即可。
- (2) 十进制转换为八进制（十六进制）：一般先把十进制转换为二进制，再转换为八（十六）进制。
- (3) 八进制（十六进制）转换为十进制：先把八（十六）进制转换为二进制，再转换为十进制。

## 1.1.3 码 制

### 1. 原码、反码和补码

各种数制都有原码、反码和补码之分。前面介绍的十进制数和二进制数都属于原码。二进制的反码和补码很重要，因为它们允许表达负数。

在数字电子计算机中，二进制数的正负号也用 0 和 1 表示，以最高位作为符号位，正数为 0，负数为 1。

例如： 0 0100 表示 +4                      1 0011 表示 -3

$$(0\ 1010111)_2 = (+87)_{10}$$

↓

符号位

$$(1\ 1010111)_2 = (+87)_{10}$$

↓

符号位

为了简化运算电路，在数字电路中两数相减的运算是用它们的补码相加来完成的。二进制数的补码是这样定义的：最高位为符号位，正数为 0，负数为 1；正数的补码和它的原码相同；负数的补码为原码的数值位逐位求反，然后在最低位上加 1。

例如：计算  $(1001)_2 - (0101)_2$ ，根据二进制减法运算规则有：

$$\begin{array}{r} 1001 \\ -0101 \\ \hline 0100 \end{array}$$

在采用补码运算时，首先求出  $(1001)_2$  和  $-(0101)_2$  的补码，即正数的补码是符号位加数值位，正数的符号位为 0，数值位为正数本身；负数的符号位为 1，数值位为正数逐位求反后再加 1，即

$$[+1001]_{\text{补}} = 0\ 1011$$

↓

符号位

$$[-0101]_{\text{补}} = 1\ 1011$$

↓

符号位

然后将两个补码相加并舍去进位：

$$\begin{array}{r} 01001 \\ +10101 \\ \hline 100100 \end{array}$$

这样就把减法运算转换为加法运算。

## 2. 二进制编码

用文字、符号或数码来表示特性对象的过程称为编码。数字系统中常用的二进制编码，即用二进制代码表示相关的对象。这里重点介绍二-十进制码 BCD 码。

在数字系统中，一般是采用二进制数进行运算的，但是由于人们习惯采用十进制，因此常需要进行十进制数和二进制数之间的转换。为了便于数字系统处理十进制数，经常采用编码的方法，即以若干位二进制码来表示一位十进制数，这种代码为二进制编码的十进制数，简称二-十进制码或 BCD 码 (binary coded decimal codes)。常见的 BCD 码如表 1-2 所示，有 8421 码、5421 码、2421 码和余 3 码等。实际应用中一般计算问题的原始数据大多是十进制数，人们为计算机设计了一种用二进制数为它编码，该编码称 BCD 码。

一位的 BCD 码可以用四位二进制表示：0001 (1) ~ 1001 (9)。

一字节的 BCD 码可表示数值范围为 0~99。

### 1) 8421 码

8421 码是一种有权码，0~9 的 8421 码与其二进制码完全相同，所以求一个数的 8421 码就将这个数按位转化为二进制 (四位，不足前面补 0)。例如，十进制的 123，二进制为 1111001，8421 码为 (0001 0010 0011)<sub>BCD8421</sub>。和四位自然二进制码不同的是，8421BCD 码只选用了四位二进制码中前 10 组代码，即 0000~1001 分别代表它所对应的十进制数，余下的六组代码不用。

### 2) 5421 码

5421 码是一种有权码，四位二进制码的权依次为 5, 4, 2, 1。从十进制的 0~9 转换为 5421 码，就是按照每一位的权凑出所要的数字，在凑 5421 码的时候，先用大的数，比如 9，9=5+4，所以 9 的 5421 码就是 1100。但是四位二进制码可以表示 16 个数，十进制中只有 10 个，就会有 6 个用不到 (不允许出现)，这 6 个分别是 0101、0110、0111、1101、1110、1111。

### 3) 2421 码

2421 码也是一种有权码，四位二进制码的权依次为 2, 4, 2, 1。从十进制的数制转换到 2421 码也是凑的思路。但是出现了两个 2，并且还有 4 (=2+2) 就注定了这个规律要比 5421 码稍微复杂一些。2421 码中的 10 个数码中，0 和 9，1 和 8，2 和 7，3 和 6，4 和 5，代码的对应位恰好一个是 0 时，另一个就是 1，即互为反码。可见 2421BCD 码具有对 9 互补的特点，它是一种对 9 的自补代码 (即只要对某一组代码各位取反就可以得到 9 的补码)，这在运算电路中使用较为方便。