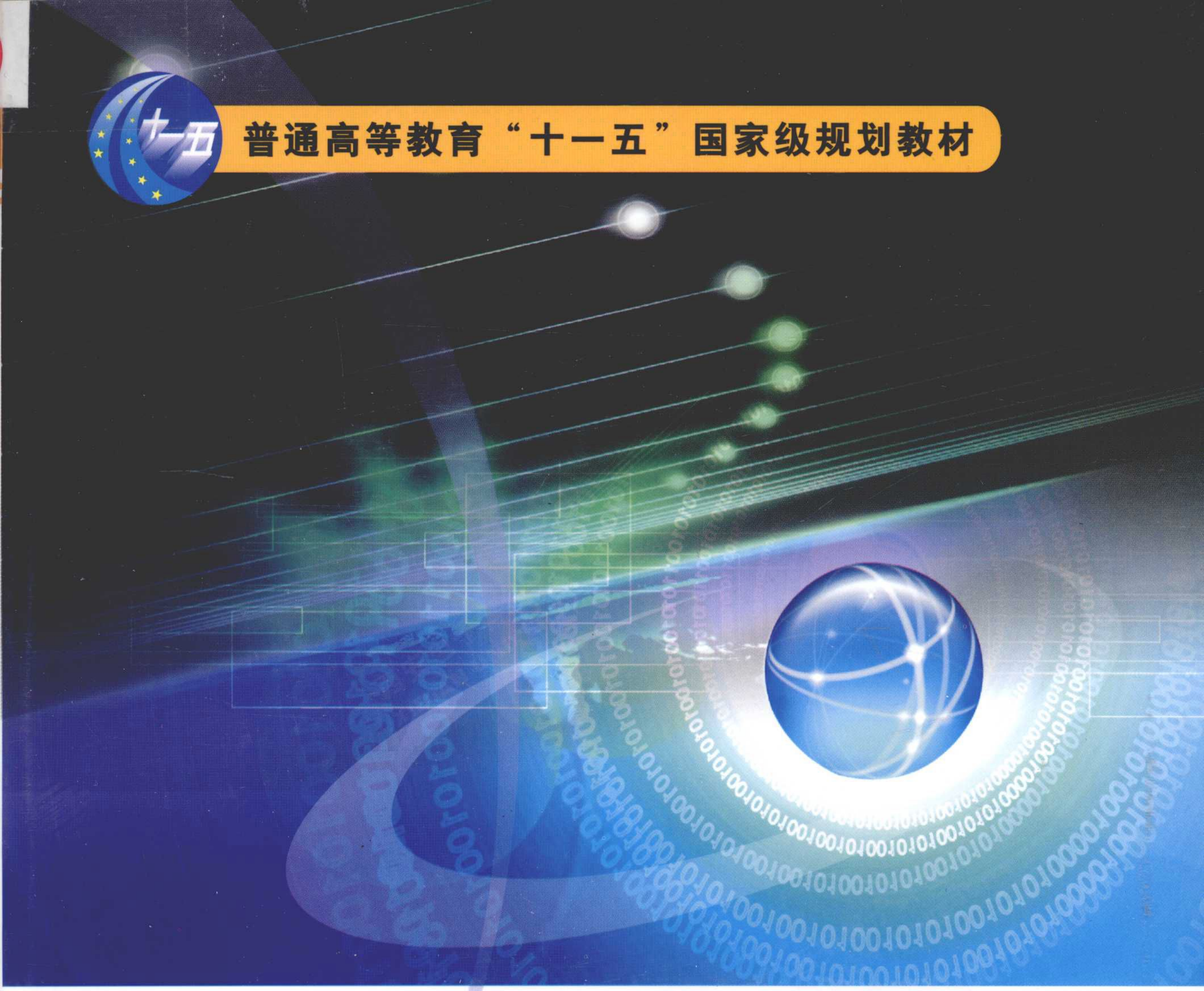
普通高等教育“十一五”国家级规划教材



# 数字电子技术基础

（第三版）

杨颂华　冯毛官　孙万蓉
初秀琴　胡力山　编著

西安电子科技大学出版社
http://www.xduph.com

普通高等教育“十一五”国家级规划教材

# 数字电子技术基础

## （第三版）

杨颂华　冯毛官　孙万蓉
初秀琴　胡力山　编著

西安电子科技大学出版社

## 内 容 简 介

本书为普通高等教育“十一五”国家级规划教材。

全书共分11章，主要内容包括：数制与编码、逻辑代数基础、集成逻辑门、组合逻辑电路、触发器、时序逻辑电路、脉冲波形的产生与整形、存储器和可编程逻辑器件、数/模和模/数转换器、VHDL硬件描述语言简介、VHDL数字系统设计实例等。书中各章均选用了较多的典型实例，并配有相当数量的习题，便于读者联系实际，灵活运用。

本书可作为高等学校通信、电子工程、自动控制、工业自动化、检测技术及电子技术应用等相关专业本科和专科生“数字电路”课程的基本教材和教学参考书，也可作为相关工程技术人员的参考书。

**★本书配有电子教案，需要者可登录出版社网站，免费下载。**

**图书在版编目(CIP)数据**

**数字电子技术基础**/杨颂华等编著. —3版.

—西安：西安电子科技大学出版社，2016.7

普通高等教育“十一五”国家级规划教材

ISBN 978-7-5606-4097-6

Ⅰ.①数… Ⅱ.①杨… 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(2016)第143197号**

策　　划　云立实

责任编辑　云立实

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　　电子信箱　xdupfxb001@163.com

经　　销　新华书店

印刷单位　陕西华沐印刷科技有限责任公司

版　　次　2016年7月第3版　2016年7月第18次印刷

开　　本　787毫米×1092毫米　1/16　印张 20.25

字　　数　481千字

印　　数　101 001～104 000册

定　　价　36.00元

ISBN 978-7-5606-4097-6/TN

**XDUP 4389003-18**

＊＊＊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。

# 第三版前言

《数字电子技术基础(第二版)》一书已出版发行六年有余。根据当前 EDA 技术的发展和教学改革的需要，为了更加便于教学和读者自学，我们经过教学实践经验的积累并广泛征集了读者的意见，对该书进行了本次修订。

本书第三版基本保留了第二版的特色和知识框架，仅对局部内容进行了修改和调整，具体做法是：

(1) 对第 6 章时序逻辑电路的内容进行了调整和修改，删减了小规模时序逻辑电路分析、设计的内容，保留同步时序逻辑电路分析、设计的基本方法和步骤；在典型时序逻辑电路的叙述中，主要强调各种电路的结构特点、状态变化规律和信号之间的时序关系，重点介绍中规模集成时序电路的分析和设计方法，减少集成时序电路内部结构的分析过程，增强集成时序电路逻辑框图、控制功能的认识，为学习硬件描述语言和系统设计打好基础。

(2) 为了便于使用 EDA 软件工具和阅读国外技术资料，第三版仍然保留了第二版采用的国际上流行的图形逻辑符号，但对电路图中的所有节点都重新进行了标注，使其更加清晰、明确。

(3) 对部分习题进行了调整。

本书第 2、8 章由杨颂华编写，第 5、6、9 章由冯毛官编写，第 1、7 章及各章关键词汉译英由孙万蓉编写，第 10、11 章由初秀琴编写，第 3、4 章由胡力山编写，全书由杨颂华进行修改和定稿。

本书由哈尔滨工业大学蔡惟铮教授主审，在修订出版过程中得到了孙肖子教授、江小安教授、云立实副编审等人的支持和帮助，在此表示衷心的感谢！

由于我们水平和时间有限，书中错误与疏漏之处在所难免，敬请同行及广大读者批评指正。

编 著 者

2016 年 1 月于西安电子科技大学

# 目　录





此为试读，需要完整PDF请访问：www.ertongbook.com

# 第1章 数制与编码

数字系统的基本功能是对数字信息进行加工和处理，如数的运算、传输和变换等，因此我们首先要对数的基本特征有所了解。

本章从常用的十进制数开始，分析推导各种不同数制的表示方法以及各种数制之间的转换方法，并着重讨论数字计算机和其他数字设备中广泛采用的二进制数，最后介绍几种常用的编码。

## 1.1 数字逻辑电路概述

自然界的各种物理量可分为模拟量和数字量两大类。模拟量在时间上是连续取值，在幅值上也是连续变化的。表示模拟量的信号称为模拟信号。处理模拟信号的电子电路称为模拟电路。数字量是一系列离散的时刻取值，数值的大小和每次的增减都是量化单位的整数倍，即它们是一系列时间离散、数值也离散的信号。表示数字量的信号称为数字信号。处理数字信号的电子电路称为数字电路。

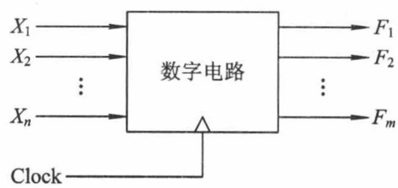


图 1.1.1 数字电路的一般框图

数字电路的一般框图如图 1.1.1 所示，它有 $n$ 个输入 $X_1$、$X_2$、…、$X_n$ 和 $m$ 个输出 $F_1$、$F_2$、…、$F_m$，此外还有一个定时信号，即时钟脉冲信号(Clock)。每一个输入 $X_i$ 和输出 $F_j$ 都是时间和数值上离散的二值信号，用数字 0 和 1 来表示。在数字电路和系统中，可以用 0 和 1 组成的二进制数码表示数量的大小，也可以用 0 和 1 表示两种不同的逻辑状态。当用 0 和 1 表示客观事物的两种对立状态时，它不表示数值，而表示逻辑 0 和逻辑 1，这两种对立的逻辑状态称为二值数字逻辑或简称为数字逻辑。数字电路的输出与输入之间满足一定的逻辑关系，因而数字电路也称为逻辑电路。

数字电路中的电子器件都工作在开关状态，电路的输出只有高、低两个电平，因而很容易实现二值数字逻辑。在分析实际电路时，逻辑高电平和逻辑低电平都对应一定的电压范围，不同系列的数字集成电路其输入、输出为高电平或低电平时所对应的电压范围是不同的(参考第 3 章)。一般用逻辑高电平(或接电源电压)表示逻辑 1 和二进制数的 1，用逻辑低电平(或接地)表示逻辑 0 和二进制数的 0。在数字电路中，当用高电平表示逻辑 1，用低电平表示逻辑 0 时称为正逻辑；当用低电平表示逻辑 1，用高电平表示逻辑 0 时称为负逻辑。通常情况下数字电路使用正逻辑。

数字电路的输入、输出逻辑电平随时间变化的波形称为数字波形。数字波形有两种类型：一种是电位型(或称非归零型)，另一种是脉冲型(或称归零型)。在波形图中，一定的时间间隔 $T$ 称为 1 位(1 bit)或一拍。电位型的数字波形在一拍时间内用高电平表示 1，用

低电平表示 0；脉冲型的数字波形则在一拍时间内以脉冲有无来表示 1 和 0。图 1.1.2 所示为 01001101100 序列信号的两种数字波形，其中图(a)为电位型的波形，图(b)是脉冲型的波形。

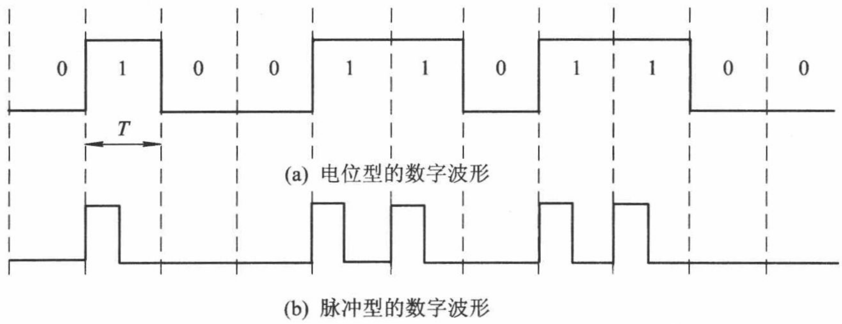



图 1.1.2 序列信号的两种数字波形

数字电路和系统的输入、输出逻辑关系(功能或行为)通常可以用文字、真值表、逻辑函数表达式、逻辑电路图、时序图、状态图、状态表等多种形式进行描述。此外，还可以采用硬件描述语言进行描述。各种描述形式将在后续章节介绍。

数字电路系统只能处理用二进制数表示的数字信号，而人们习惯用的十进制数不能直接被数字电路系统接收。因此，在人与数字电路系统交换信息时，需要把十进制数转换成二进制数；当数字系统运行结束时，为了便于人们阅读，又需要将二进制数转换成十进制数。所以为了便于信息交换和传输，我们需要研究各种数制之间的转换及不同的编码方式。

## 1.2 数　　制

### 1.2.1 进位计数制

按进位原则进行计数，称为进位计数制。每一种进位计数制都有一组特定的数字、符号，例如十进制数有 10 个数符，二进制数只有 2 个数符，而十六进制数有 16 个数符。每种进位计数制中允许使用的数符总数称为基数或底数。

在进位计数制中，任何一个数都由整数和小数两部分组成，并且具有两种书写形式：位置计数法和多项式表示法。

**1. 十进制数(Decimal)**

十进制数具有以下特点：

(1) 采用 10 个不同的数符 0、1、2、…、9 和一个小数点( . )。

(2) 进位规则是“逢十进一”。

若干个数符并列在一起可以表示一个十进制数。例如在 435.86 这个数中，小数点左边第一位 5 代表个位，它的数值为 5；小数点左边第二位 3 代表十位，它的数值为 $3\times10^1$；小数点左边第三位 4 代表百位，它的数值为 $4\times10^2$；小数点右边第一位的值为 $8\times10^{-1}$；小数点右边第二位的值为 $6\times10^{-2}$。可见，数符处于不同的数位，代表的数值是不同的。这里

$10^2$、$10^1$、$10^0$、$10^{-1}$、$10^{-2}$称为权或位权，即十进制数中各位的权是10的幂，因此435.86可表示为

$$435.86 = 4\times10^2+3\times10^1+5\times10^0+8\times10^{-1}+6\times10^{-2}$$

上式左边称为位置计数法或并列表示法，右边称为多项式表示法或按权展开法。

通常对于任何一个十进制数 $N$，都可以用位置计数法和多项式表示法写为

$$\begin{aligned}(N)_{10} &= a_{n-1}a_{n-2}\cdots a_1a_0.a_{-1}a_{-2}\cdots a_{-m}\\ &= a_{n-1}\times10^{n-1}+a_{n-2}\times10^{n-2}+\cdots+a_1\times10^1+a_0\times10^0+a_{-1}\times10^{-1}\\ &\quad+a_{-2}\times10^{-2}+\cdots+a_{-m}\times10^{-m}\\ &= \sum_{i=-m}^{n-1}a_i\times10^i\end{aligned}$$

式中：$n$ 代表整数位数；$m$ 代表小数位数；$a_i(-m\leqslant i\leqslant n-1)$表示第 $i$ 位数符，它可以是0、1、2、3、…、9中的任意一个；$10^i$ 为第 $i$ 位数符的权值。

上述十进制数的表示方法也可以推广到任意进制数。对于一个基数为$R(R\geqslant2)$的 $R$ 进制计数制，数 $N$ 可以写为

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

式中：$n$ 代表整数位数；$m$ 代表小数位数；$a_i$ 为第 $i$ 位数符，它可以是0、1、…、$R-1$ 个不同数符中的任何一个；$R^i$ 为第 $i$ 位数符的权值。

**2. 二进制数(Binary)**

二进制数的进位规则是“逢二进一”，其进位基数 $R=2$，每位数符的取值只能是0或1，每位的权是2的幂。表1.2.1列出了二进制位数、权和十进制数的对应关系。

**表 1.2.1 二进制位数、权和十进制数的对应关系**

| 二进制整数位 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 权 | $2^{12}$ | $2^{11}$ | $2^{10}$ | $2^9$ | $2^8$ | $2^7$ | $2^6$ | $2^5$ | $2^4$ | $2^3$ | $2^2$ | $2^1$ | $2^0$ |
| （十进制表示） | 4096 | 2048 | 1024 | 512 | 256 | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |

| 二进制小数位 | −1 | −2 | −3 | −4 | −5 | −6 |
|---|---|---|---|---|---|---|
| 权 | $2^{-1}$ | $2^{-2}$ | $2^{-3}$ | $2^{-4}$ | $2^{-5}$ | $2^{-6}$ |
| （十进制表示） | 0.5 | 0.25 | 0.125 | 0.0625 | 0.031 25 | 0.015 625 |

任何一个二进制数可表示为

$$\begin{aligned}(N)_2 &= a_{n-1}a_{n-2}\cdots a_1a_0.a_{-1}a_{-2}\cdots a_{-m}\\ &= a_{n-1}\times2^{n-1}+a_{n-2}\times2^{n-2}+\cdots+a_1\times2^1+a_0\times2^0+a_{-1}\times2^{-1}\\ &\quad+a_{-2}\times2^{-2}+\cdots+a_{-m}2^{-m}\\ &= \sum_{i=-m}^{n-1}a_i2^i\end{aligned}$$

例如：

$$\begin{aligned}(1011.011)_2 &= 1\times 2^3+0\times 2^2+1\times 2^1+1\times 2^0+0\times 2^{-1}+1\times 2^{-2}+1\times 2^{-3}\\ &=(11.375)_{10}\end{aligned}$$

二进制数具有以下特点：

(1) 二进制数只有0、1两个数符。在数字电路中利用一个开关器件就可以表示一位二进制数，其电路容易实现，且工作稳定可靠。

(2) 二进制数的算术运算和十进制数的算术运算规则相似，不同的是二进制数是“逢二进一”和“借一当二”，而不是“逢十进一”和“借一当十”。例如：

加法运算

$$\begin{array}{r} 1101.01\\ +\ \ 1001.11\\ \hline 10111.00 \end{array}$$

减法运算

$$\begin{array}{r} 1101.01\\ -\ \ 1001.11\\ \hline 0011.10 \end{array}$$

乘法运算

$$\begin{array}{r} 1101\\ \times\ \ 110\\ \hline 0000\\ 1101\phantom{0}\\ 1101\phantom{00}\\ \hline 1001110 \end{array}$$

除法运算

$$\begin{array}{rl} & \phantom{11}101\ \cdots\cdots\text{商}\\ 101 & \!\!\big)\overline{11011}\\ & \underline{101\phantom{00}}\\ & \phantom{0}111\\ & \phantom{0}\underline{101}\\ & \phantom{00}10\ \cdots\cdots\text{余数} \end{array}$$

从运算过程可看出，二进制乘法运算由左移被乘数与加法运算组成，而除法运算由右移被除数与减法运算组成。

**3. 八进制数(Octal)**

八进制数的进位规则是“逢八进一”，其基数 $R=8$，采用的数符是0、1、2、3、4、5、6、7，每位的权是8的幂。任何一个八进制数可以表示为

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

例如：

$$(376.4)_8=3\times 8^2+7\times 8^1+6\times 8^0+4\times 8^{-1}=3\times 64+7\times 8+6+0.5=(254.5)_{10}$$

**4. 十六进制数(Hexadecimal)**

十六进制数的特点如下：

(1) 采用的16个数符为0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F。符号A～F分别代表十进制数的10～15。

(2) 进位规则是“逢十六进一”，基数 $R=16$，每位的权是16的幂。

任何一个十六进制数可以表示为

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

例如：

$$(3AB.11)_{16}=3\times 16^2+10\times 16^1+11\times 16^0+1\times 16^{-1}+1\times 16^{-2}\approx(939.0664)_{10}$$

## 1.2.2　进位计数制之间的转换

### 1. 二进制数与十进制数之间的转换

1) 二进制数转换成十进制数——按权展开法

二进制数转换成十进制数时，只要将二进制数写成按权展开的多项式，然后按十进制数规则进行运算，所得结果便为相应的十进制数。例如：

此为试读，需要完整PDF请访问：www.ertongbook.com

$$(10110.11)_2=1\times2^4+1\times2^2+1\times2^1+1\times2^{-1}+1\times2^{-2}=(22.75)_{10}$$

同理，若将任意进制数转换为十进制数，则只需将数$(N)_R$写成按权展开的多项式表达式，并按十进制规则进行运算，便可求得相应的十进制数$(N)_{10}$。

2）十进制数转换成二进制数

十进制数转换为二进制数时，需要对其整数部分和小数部分分别进行转换。

(1) 整数转换——除2取余法。若将十进制整数$(N)_{10}$转换为二进制整数$(N)_2$，则按照转换前后相等的原则，可写成

$$\begin{aligned}(N)_{10}&=a_{n-1}\times2^{n-1}+a_{n-2}\times2^{n-2}+\cdots+a_1\times2^1+a_0\times2^0\\&=2(a_{n-1}\times2^{n-2}+a_{n-2}\times2^{n-3}+\cdots+a_2\times2^1+a_1)+a_0\\&=2Q_1+a_0\end{aligned}$$

将上式两边同除以2，所得的商为

$$Q_1=a_{n-1}\times2^{n-2}+a_{n-2}\times2^{n-3}+\cdots+a_2\times2^1+a_1\quad 余数为\ a_0$$

同理，将上式两边同除以2，得到的新商为

$$Q_2=a_{n-1}\times2^{n-3}+a_{n-2}\times2^{n-4}+\cdots+a_2\quad 余数为\ a_1$$

重复上述过程，直至得到的商为$Q_n=0$，余数为$a_{n-1}$，于是可得二进制整数的数符$a_0$、$a_1$、…、$a_{n-1}$。

例如，将$(57)_{10}$转换为二进制数：

| | | | |
|---|---|---|---|
| 2 | 57 | …… | 余数 |
| 2 | 28 | …… | $1=a_0$ |
| 2 | 14 | …… | $0=a_1$ |
| 2 | 7 | …… | $0=a_2$ |
| 2 | 3 | …… | $1=a_3$ |
| 2 | 1 | …… | $1=a_4$ |
| | 0 | …… | $1=a_5$ |

故

$$(57)_{10}\approx(111001)_2$$

(2) 小数转换——乘2取整法。若将十进制数小数$(N)_{10}$转换为二进制小数$(N)_2$，则可写成

$$(N)_{10}=a_{-1}\times2^{-1}+a_{-2}\times2^{-2}+\cdots+a_{-m}\times2^{-m}$$

将上式两边同时乘以2，便得到

$$2(N)_{10}=a_{-1}+(a_{-2}\times2^{-1}+\cdots+a_{-m}\times2^{-m+1})=a_{-1}+F_1$$

可见，$2(N)_{10}$乘积的整数部分就是$a_{-1}$，小数部分就是$F_1$。若将$2(N)_{10}$乘积的小数部分$F_1$再乘以2，则有

$$2F_1=a_{-2}+(a_{-3}\times2^{-1}+a_{-4}\times2^{-2}+\cdots+a_{-m}\times2^{-m+2})=a_{-2}+F_2$$

所得乘积整数部分就是$a_{-2}$，小数部分为$F_2$。显然，重复上述过程，便可求出二进制小数的各位数符$a_{-1}$、$a_{-2}$、…、$a_{-m}$。

例如，将$(0.724)_{10}$转换成二进制小数：

$$
\begin{array}{rll}
 & 0.724 & \\
\times & 2 & \qquad \text{整数} \\
\hline
 & 1.448 & \cdots\cdots\ 1=a_{-1} \\
 & 0.448 & \\
\times & 2 & \\
\hline
 & 0.896 & \cdots\cdots\ 0=a_{-2} \\
\times & 2 & \\
\hline
 & 1.792 & \cdots\cdots\ 1=a_{-3} \\
 & 0.792 & \\
\times & 2 & \\
\hline
 & 1.584 & \cdots\cdots\ 1=a_{-4}
\end{array}
$$

故

$$(0.724)_{10} \approx (0.1011)_2$$

应指出，小数部分乘 2 取整的过程不一定能使最后乘积为 0，因此转换值存在一定的误差。通常在二进制小数的精度已达到预定的要求时，运算便可结束。

将一个带有整数和小数的十进制数转换成二进制数时，必须将整数部分和小数部分分别按除 2 取余法和乘 2 取整法进行计算，然后将两者的转换结果合并起来。

同理，若将十进制数转换成任意 $R$ 进制 $(N)_R$，则整数部分转换采用除 $R$ 取余法，小数部分采用乘 $R$ 取整法。

**2. 二进制数与八进制数、十六进制数之间的相互转换**

八进制数和十六进制数的基数分别为 $8=2^3$，$16=2^4$，所以 3 位二进制数恰好相当于一位八进制数，4 位二进制数恰好相当于一位十六进制数，它们之间的相互转换是很方便的。

二进制数转换成八进制数的方法是从小数点开始，分别向左、向右将二进制数按每 3 位一组分组(不足 3 位的补 0)，然后写出每一组等值的八进制数。

例如，求 $(01101111010.1011)_2$ 的等值八进制数：

$$
\begin{array}{lcccccc}
\text{二进制} & \underline{001} & \underline{101} & \underline{111} & \underline{010}. & \underline{101} & \underline{100} \\
\text{八进制} & 1 & 5 & 7 & 2. & 5 & 4
\end{array}
$$

所以

$$(01101111010.1011)_2 = (1572.54)_8$$

二进制数转换成十六进制数的方法和二进制数转换成八进制数的方法相似，从小数点开始分别向左、向右将二进制数按每 4 位一组分组(不足 4 位补 0)，然后写出每一组等值的十六进制数。

例如，将 $(1101101011.101)_2$ 转换为十六进制数：

$$
\begin{array}{cccc}
\underline{0011} & \underline{0110} & \underline{1011}. & \underline{1010} \\
3 & 6 & \mathrm{B}. & \mathrm{A}
\end{array}
$$

所以

$$(1101101011.101)_2 = (36\mathrm{B.A})_{16}$$

八进制数、十六进制数转换为二进制数的方法可以采用与前面相反的步骤，即只要按原来的顺序将每一位八进制数(或十六进制数)用相应的 3 位(或 4 位)二进制数代替即可。

例如，分别求出 $(375.46)_8$、$(678.\mathrm{A5})_{16}$ 的等值二进制数：

| | | | |
|---|---|---|---|
| 八进制 | 3 7 5 . 4 6 | 十六进制 | 6 7 8 . A 5 |
| 二进制 | 011 111 101 . 100 110 | 二进制 | 0110 0111 1000 . 1010 0101 |

所以

$$(375.46)_8=(011111101.100110)_2$$

$$(678.A5)_{16}=(011001111000.10100101)_2$$

# 1.3 编　　码

在数字系统中，任何数据和信息都是用若干位“0”和“1”按照一定的规则组成的二进制码来表示的。$n$ 位二进制数码可以组成 $2^n$ 种不同的代码，代表 $2^n$ 种不同的信息或数据。因此，用若干位二进制数码按一定规律排列起来表示给定信息的过程称为编码。下面介绍数字系统中常用的编码及特性。

## 1.3.1 带符号数的编码

在数字系统中，需要处理的不仅有正数，还有负数。为了表示带符号的二进制数，在定点整数运算的情况下，通常以代码的最高位作为符号位，用 0 表示正，用 1 表示负，其余各位为数值位。代码的位数称为字长，它的数值称为真值。

带符号的二进制数可以用原码、反码和补码几种形式表示。

**1. 原码**

原码的表示方法是：符号位加数值位。

例如，真值分别为＋62 和－62，若用 8 位字长的原码来表示，则可写为

$$N=+62_D=+0111110_B \qquad [N]_{原}=00111110$$

$$N=-62_D=-0111110_B \qquad [N]_{原}=10111110$$

原码表示简单、直观，而且与真值转换方便，但用原码进行减法运算时，电路结构复杂，不容易实现，因此引入了反码和补码。

**2. 反码**

反码的表示方法是：正数的反码与其原码相同，即符号位加数值位；负数的反码是符号位为 1，数值位各位取反。

例如，真值分别为＋45 和－45，若用 8 位字长的反码来表示，则可写为

$$[+45]_{原}=00101101 \qquad [+45]_{反}=00101101$$

$$[-45]_{原}=10101101 \qquad [-45]_{反}=11010010$$

**3. 补码**

字长为 $n$ 的整数 $N$ 的补码定义如下：

$$[N]_{补}=\begin{cases}N & 0\leqslant N<2^{n-1}\\ 2^n+N & -2^{n-1}\leqslant N<0\end{cases} \qquad (\mathrm{mod}\ 2^n)$$

由于 $2^n-1$ 与 $n$ 位全为 1 的二进制数等值，而 $2^n$ 比 $2^n-1$ 多 1，所以求一个数的补码可以用以下简便方法：

(1) 正数和 0 的补码与原码相同。

(2) 负数的补码是将其原码的符号位保持不变，对数值位逐位求反，然后在最低位加 1。

此外，应注意以下几点：

• $n$ 位字长的二进制原码、反码、补码所表示的十进制数值范围是：

原码：$-(2^{n-1}-1)\sim+(2^{n-1}-1)$。

反码：$-(2^{n-1}-1)\sim+(2^{n-1}-1)$。

补码：$-2^{n-1}\sim+(2^{n-1}-1)$（不含$-0$）。

例如：4 位字长的原码、反码其数值表示范围均为$-7\sim+7$，而补码的范围则为$-8\sim+7$；$+0$ 的原码、反码、补码均为 0000，$-0$ 只有原码(1000)和反码(1111)，而没有补码；$-8$ 只有补码(1000)，而没有原码和反码。

• 如果已知一个数的补码，则可以用$\{[X]_{补}\}_{补}=[X]_{原}$求其原码和真值。

**【例 1.3.1】** 已知十进制数$+6$ 和$-5$，试分别用 4 位字长和 8 位字长的二进制补码来表示。

**解**：(1) $n=4$：

$$[+6]_{原}=0110 \qquad [+6]_{补}=0110$$
$$[-5]_{原}=1101 \qquad [-5]_{补}=1011$$

(2) $n=8$：

$$[+6]_{原}=00000110 \qquad [+6]_{补}=00000110$$
$$[-5]_{原}=10000101 \qquad [-5]_{补}=11111011$$

**【例 1.3.2】** 已知 4 位字长的二进制补码分别为 0011、1011、1000，试求出相应的十进制数。

**解**：(1) 因为$[X]_{补}=0011$，符号位为 0，所以$[X]_{原}=0011$，$X=+3$。

(2) 因为$[X]_{补}=1011$，符号位为 1，所以$[X]_{原}=[1011]_{补}=1101$，$X=-5$。

(3) 因为$[X]_{补}=1000$，符号位为 1，它是$n=4$ 时$-8$ 的补码，而$-8$ 没有原码和反码，所以 $X=-8$。

**4. 补码的运算**

在数字系统中，求一个数的反码和补码都很容易，而且利用补码可以方便地进行带符号二进制数的加、减运算。若 $X$、$Y$ 均为正整数，则 $X-Y$ 的运算可以通过$[X]_{补}+[-Y]_{补}$来实现，这样将减法运算变成了加法运算，因而简化了电路结构。

采用补码进行加、减法运算的步骤如下：

(1) 根据$[X\pm Y]_{补}=[X]_{补}+[\pm Y]_{补}$，分别求出$[X]_{补}$、$[\pm Y]_{补}$和$[X+Y]_{补}$。

(2) 补码相加时，符号位参与运算，若符号位有进位，则自动舍去。

(3) 根据$[X\pm Y]_{补}$的结果求出$[X\pm Y]_{原}$，进而求出 $X\pm Y$ 的结果。

**【例 1.3.3】** 试用 4 位字长的二进制补码完成下列运算：

① 7－5；② 3－4。

**解**：$[7]_{补}=0111$，$[-5]_{补}=1011$，$[3]_{补}=0011$，$[-4]_{补}=1100$。

① $[7]_{补}+[-5]_{补}$为

$$\begin{array}{r} 0111 \\ +\quad 1011 \\ \hline \text{舍去}\leftarrow(1)\ 0010 \end{array}$$

即$[7-5]_{补}=[7]_{补}+[-5]_{补}=0010$，符号位为0，所以$[7-5]_{原}=0010$，故$7-5=+2$。

② $[3]_{补}+[-4]_{补}$为

$$\begin{array}{r} 0011 \\ +\quad 1100 \\ \hline 1111 \end{array}$$

即$[3-4]_{补}=[3]_{补}+[-4]_{补}=1111$，符号位为1，所以$[3-4]_{原}=1001$，故$3-4=-1$。

必须指出，两个补码相加时，如果产生的和超出了有效数字位所表示的范围，则计算结果会出错，之所以发生错误，是因为计算结果产生了溢出，解决的办法是扩大字长。

### 1.3.2　二-十进制编码(BCD码)

二-十进制编码是用4位二进制码的10种组合表示十进制数0～9，简称BCD码(Binary Coded Decimal)。

这种编码至少需要用4位二进制数码，而4位二进制数码可以有16种组合。当用这些组合表示十进制数0～9时，有6种组合不用。从16种组合中选用10种组合，有

$$A_{16}^{10}=\frac{16!}{(16-10)!}\approx 2.9\times 10^{10}$$

种编码方案，但并不是所有的方案都有实用价值。表1.3.1列出了几种常用的BCD码的编码方式。

**表1.3.1　几种常用的BCD码的编码方式**

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

**1. 8421 BCD码**

8421 BCD码是最基本和最常用的BCD码，它和4位自然二进制码相似，各位的权值为8、4、2、1，故称为有权BCD码。和4位自然二进制码不同的是，8421 BCD码只选用了4位二进制码中的前10组代码，即用0000～1001分别代表十进制数的0～9，余下的6组代码1010～1111不用。

**2. 5421 BCD码和2421 BCD码**

5421 BCD码和2421 BCD码均属于有权BCD码，它们从高位到低位的权值分别为5、4、2、1和2、4、2、1。这两种BCD码的编码方案不是唯一的。例如：5421 BCD码中的

数码 5 既可以用 1000 表示，也可以用 0101 表示；2421 BCD 码中的数码 6 既可以用 1100 表示，也可以用 0110 表示。表 1.3.1 只列出了一种常用的编码方式。

表 1.3.1 所示的 2421 BCD 码的 10 个数码中，0 和 9、1 和 8、2 和 7、3 和 6、4 和 5 的代码对应恰好一个是 0 时，另一个就是 1。我们称 0 和 9、1 和 8 互为反码。因此 2421 BCD 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)，在运算电路中使用比较方便。

**3. 余 3 码**

余 3 码是 8421 BCD 码的每个码组加 3(0011)形成的。余 3 码也具有对 9 互补的特点，即它也是一种 9 的自补码，所以也常用于 BCD 码的运算电路中。

用 BCD 码可以方便地表示多位十进制数，其转换方法是每一位十进制数用一组 BCD 码代替，例如：

$$(579.8)_{10} = (0101\ 0111\ 1001\ .\ 1000)_{8421\ BCD} = (1000\ 1010\ 1100\ .\ 1011)_{余3码}$$

### 1.3.3 可靠性编码

代码在形成、传输过程中可能会发生错误。为了减少这种错误，出现了可靠性编码。常用的可靠性编码有以下两种。

**1. Gray 码(格雷码)**

Gray 码最基本的特性是任何相邻的两组代码中，仅有一位数码不同，即具有相邻性，因此又称单位距离码。此外，Gray 码的首尾两个码组也有相邻性，因此又称循环码。

Gray 码的编码方案有多种，典型的 Gray 码如表 1.3.2 所示。

**表 1.3.2 典型的 Gray 码**

| 十进制数 | 二进制码 | | | | Gray 码 | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | $B_3$ | $B_2$ | $B_1$ | $B_0$ | $G_3$ | $G_2$ | $G_1$ | $G_0$ | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | …一位反射对称轴 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | …二位反射对称轴 |
| 2 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | |
| 3 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | …三位反射对称轴 |
| 4 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | |
| 5 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | |
| 6 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | |
| 7 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | …四位反射对称轴 |
| 8 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | |
| 9 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | |
| 10 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | |
| 11 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | |
| 12 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | |
| 13 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | |
| 14 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | |
| 15 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | |

此为试读，需要完整PDF请访问：www.ertongbook.com