高等学校电子信息类规划教材

数字电子技术基础

王海光 主 编

西安电子科技大学出版社
http://www.xduph.com

高等学校电子信息类规划教材

数字电子技术基础

主　编　王海光
副主编　林　凡　陈赐海

西安电子科技大学出版社

内 容 简 介

本书是福建省本科高校专业综合改革试点项目的改革成果之一，是按照教育部“专业综合改革试点”项目的内涵和要求，为主动适应海峡西岸经济区电子信息及相关产业和行业对应用型人才的需求，提高电子信息类专业学生工程实践和创新的能力，对传统数字电子技术基础课程的教学内容进行改写、整合而成的。

全书共10章，内容包括数字电路基础、集成门电路、组合逻辑电路、触发器、时序逻辑电路、半导体存储器、可编程逻辑器件、脉冲产生和整形电路、数/模与模/数转换、数字系统设计初步等。

本书可作为本科应用型普通高校电子信息类、电气信息类及其他相近专业学生的专业基础课教材或课外学习参考书，也可供有关工程技术人员参考。

图书在版编目(CIP)数据

数字电子技术基础/王海光主编. —西安：西安电子科技大学出版社，2015.2

高等学校电子信息类规划教材

ISBN 978-7-5606-3610-8

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

中国版本图书馆CIP数据核字(2015)第028849号

策　　划　邵汉平

责任编辑　王　瑛

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　陕西华沐印刷科技有限责任公司

版　　次　2015年2月第1版　2015年2月第1次印刷

开　　本　787毫米×1092毫米　1/16　印　张　21

字　　数　499千字

印　　数　1～3000册

定　　价　38.00元

ISBN 978-7-5606-3610-8/TN

XDUP 3902001-1

* * * **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。

前　言

《国务院关于支持福建(省)加快建设海(峡)西(岸)经济区的若干意见》及福建省“十二五”规划纲要均明确指出，要将东部沿海地区建设成为我国先进制造业重要基地。科技含量高、经济效益好、资源消耗低、环境污染少的电子信息产业，将成为海峡西岸经济区的重点发展产业，这必将迫切需要大量的应用型电子信息类人才。闽南师范大学是福建省重点建设的普通高校，所在地漳州的电子信息产业规模全省排行第三。面对电子信息产业的迅速发展、产业的不断升级，原来的电子信息类专业人才培养模式已不能很好地适应未来电子信息产业的发展要求，现有的专业内涵、教学观念、教学内容、课程体系等都与应用型本科人才培养目标不相适应。因此，如何根据社会需求不断探索应用型本科人才培养机制，找准人才培养目标定位，更新教学观念，调整课程教学内容，改进教学方法，以培养满足社会需求的应用型高级专门人才，是地方性普通高等学校面临的一个重要的理论与实践课题。

改革须充分考虑到高等教育的发展态势、学校的类型、层次与人才规格定位、人才培养服务行业对人才的需求、专业的发展态势及特点。由于多年来招生规模的不断加大，如今我国的高等教育已从精英教育走向大众化教育，不同层次高校录取的生源基础素质差距也在不断加大，因材施教的规律要求不同层次的高校须调整本校的培养目标。重点高校以研究型大学为办学方向，以培养研究型人才为主；高职高专院校以培养实用型、职业型技术人才为主，着重于学好职业知识技能；而地方性普通高等学校以培养应用型人才为主，毕业生中的大部分将直接走向社会，毕业生去向多元化。此外，随着研究生培养规模的扩大，本科毕业生逐渐淡出大型高新技术产品的研发，更多地从事成果转化、产品改造与技术革新、生产管理和市场服务等一般性技术工作，因此地方性普通高等学校的电子信息类专业既不能像重点高校那样注重数理及学科基础，又不能像高职高专院校那样只注重实际操作，需要在学科基础与工程基础之间找到合适的平衡点，这个平衡点就是要坚持数理及学科基础够用、工程技术基础扎实的原则，加强工程基础和实践教学，培养学生科学的思维方法及创新能力、信息获取及处理能力，以提高学生的工程素质和实践能力。

基于以上考虑，本书在编写过程中理论知识以“必需”和“够用”为度，删除了与应用无直接关系的理论介绍和推导，简化了复杂电路结构与工作原理的描述，加强了有关应用方面的介绍，同时补充了一些现已盛行的新技术、新方法。比如在第 1 章“数字电路基础”中

除介绍数字电路的基本概念、分析工具和传统描述工具外，还简要介绍了数字电路的一种现代描述工具——VHDL，以适应现代数字电子系统设计对工程师的要求；第2章“集成门电路”则简化了集成门电路工作原理的讨论，补充讨论了一些门电路在实际应用中面临的问题，比如“门电路带负载时的接口电路”，突出对集成电路技术参数解读和正确使用的综合能力培养；第3章“组合逻辑电路”、第4章“触发器”和第5章“时序逻辑电路”在简要介绍相关逻辑部件的基础知识后，都各设一节来介绍和讨论相关常用数字逻辑部件的应用问题并给出实例；第6章“半导体存储器”和第7章“可编程逻辑器件”通过对有关基本单元进行高度概括、提炼和抽象，概念性地介绍其工作原理，大幅删除了内部复杂电路结构的介绍，在不影响器件使用理论指导的情况下明显减轻了学生的学习负担；第8章“脉冲产生和整形电路”、第9章“数/模与模/数转换”和第10章“数字系统设计初步”通过精选整合具有代表性的内容介绍相关知识，在确保知识结构完整性的基础上压缩篇幅，以利学生在有限学时内从容修完该门课程。

本书由闽南师范大学物理与信息工程学院的多位老师合作完成。其中：第1、2章由林凡老师执笔；第3、4和8章由陈赐海老师执笔；第5、6、7、9和10章由王海光老师执笔；郑锦良老师负责各章课后习题的编写；余超群老师负责本书课件的制作(有兴趣的读者可登录出版社网站，免费下载)。全书由王海光老师组织和统稿。

本书在编写过程中得到了闽南师范大学物理与信息工程学院周小方、庄榕榕老师及厦门大学机电工程系林育兹老师的很多帮助，也引用了诸多学者、专家的著作和他们的研究成果，这里一并向他们表示衷心的感谢。同时，也向热情支持和关心本书出版的西安电子科技大学出版社邵汉平编辑表示感谢。

由于编者水平有限，书中一定存有不妥之处，敬请广大读者批评指正，编者不胜感激！

编　者

2014年9月

目　录

第 1 章　数字电路基础

本章讨论的主要问题

常用的数制有哪些？如何相互转换？

二进制整数的原码、补码、反码如何表示？如何采用补码进行带符号数的运算？

8421BCD 码的特点是什么？

什么是与、或、非运算？

利用逻辑函数反演规则、对偶规则时，应如何处理变换的优先顺序和式中的非符号？

逻辑函数的常用表示方法有哪些？如何相互转换？

什么是最小项和最小项表达式？

使用卡诺图化简逻辑函数的依据是什么？卡诺图化简的优点是什么？

什么是无关项？在逻辑函数中如何表示？

多输出逻辑函数的化简目标是什么？

如何用 VHDL 描述简单的数字电路，并利用 MAX＋plusⅡ 进行编译综合与仿真？

概　　述

电子系统中处理的信号，按其变化规律与时间的联系可分为两类：模拟信号和数字信号。

模拟信号是指在时间和数值上都连续变化的信号，如自然界中的时间、温度、压力、速度等。模拟信号的特点是其变化总是连续的、平滑的，在任何一个极短的时间内都不发生突变。将模拟信号转换为电信号后，电信号的幅度是随时间连续变化的。用于传递、加工和处理模拟信号的电路称为模拟电路。

数字信号是指在时间和数值上是不连续的信号，其变化总是发生在一系列离散的瞬间，数量大小和每次的增减变化都是某一个最小单位的整倍数，小于这个最小单位的数值是没有物理意义的。如工厂流水线上的计件信号，计件数在一些离散的瞬间产生，是不连续的，且产品的数量也只能一个单位一个单位地增加。处理数字信号的电路称为数字电路。当强调处理电路功能的完整性和实用性时，也称其为数字系统。

本章将首先讨论数字系统中数的表示方法及其相互转换和常用的几种编码；然后介绍逻辑代数的基本概念、公式、定理等，阐述逻辑函数的表示方法及化简方法，为读者学习与掌握数字电路的分析和设计奠定基本的数学基础；最后简要介绍 VHDL 的基本结构、元素和语句，为读者开启进入现代数字电路设计的大门。

1.1 数制和码制

数制(number system)是用于表示数值大小的各种方法的统称。表示数量大小时，一位数码往往不够用，需采用进位计数的方法组成多位数码，多位数码中每一位的构成及从低位到高位的进位规则即为数制，也称为位置计数制(positional number representation)。组成数制的两个基本要素是进位基数和数位权值，简称基数(base 或 radix)和位权(power)。

(1) 基数：在一个数位上，规定使用的数码符号的个数，亦称模数，记作 R。例如，十进制中每个数位规定使用的数码符号为 0，1，2，…，9，共 10 个，故其基数 $R=10$。

(2) 位权：某个数位上数码为 1 时所表征的数值，简称“权”。各数位的权值均可表示成 R^i 的形式，i 是各数位的序号。i 按如下方法确定：整数部分，以小数点为起点，自右向左依次为 0，1，2，…，$n-1$(n 为整数部分的位数)；小数部分，以小数点为起点，自左向右依次为 -1，-2，…，$-m$(m 为小数部分的位数)。其中，m、n 均属于排除零的自然数集，即

$$m \in \mathbf{N}_+,\ n \in \mathbf{N}_+,\ \mathbf{N}_+ = \{1, 2, \cdots\}$$

某个数位上的数码 a_i 所表示的数值等于数码 a_i 与该位的权值 R^i 的乘积。

R 进制数位置计数表示如下：

$$(N)_R = a_{n-1}a_{n-2}\cdots a_2a_1a_0.a_{-1}a_{-2}\cdots a_{-m}$$

也可按权展开成多项式形式：

$$(N)_R = a_{n-1}R^{n-1} + a_{n-2}R^{n-2} + \cdots + a_2R^2 + a_1R^1 + a_0R^0 + a_{-1}R^{-1} + a_{-2}R^{-2} + \cdots + a_{-m}R^{-m}$$

$$= \sum_{i=-m}^{n-1} a_iR^i$$

数字设备除了处理二进制数外，有时候还需要处理其他数字甚至字母或符号。这些字母、数字、符号也必须用二进制数来表示。这种用二进制数按一定规则表示给定字母、数字、符号等信息的各种方法称为码制(code system)。

1.1.1 几种常用的数制

1. 十进制(Decimal)

十进制是日常生活和工作中最常用的进位计数制，使用的数码为 0，1，2，…，9，基数 R 为 10，计数规律为“逢十进一”。十进制数用下标 D 或 10 标记，有时也默认不做标记。任意一个十进制数可以按权展开为 $(N)_D = \sum_{i=-m}^{n-1} a_i10^i$。例如：

$$(568.457)_D = 5\times10^2 + 6\times10^1 + 8\times10^0 + 4\times10^{-1} + 5\times10^{-2} + 7\times10^{-3}$$

十进制数难以在数字电路中实现，因为使一个电路或者电子器件具有严格区分的 10 个状态来与十进制的 10 个数码一一对应是较困难的，所以在数字电路中一般不使用十进制。

2. 二进制(Binary)

二进制中，每个数位上使用的数码为 0 和 1，其进位基数 R 为 2，计数规律为“逢二进一”。二进制数用下标 B 或 2 标记。任意一个二进制数可以按权展开为 $(N)_B=\sum_{i=-m}^{n-1}a_i2^i$。例如：

$$(101.01)_B=1\times2^2+0\times2^1+1\times2^0+0\times2^{-1}+1\times2^{-2}$$

1 位二进制数称为一个比特(bit)，4 位二进制数称为半字节(nibble)，8 位二进制数称为一个字节(byte)。二进制各位的权值为 2^i，最左侧的位权最大，为最高有效位(MSB，Most Significant Bit)，最右侧的位权最小，为最低有效位(LSB，Least Significant Bit)。表 1.1.1 列出了二进制的部分位权。

表 1.1.1　二进制的部分位权

位置	11	10	9	8	7	6	5	4	3	2	1	0	−1	−2
位权	2048	1024	512	256	128	64	32	16	8	4	2	1	0.5	0.25

二进制数只有 0 与 1 两个数码，很容易用电路元件的状态来表示，如三极管的饱和与截止、继电器的接通与断开、电平的高与低等。由于状态简单，机器实现容易，可靠性也较高。与十进制数相比，二进制数位数多，不便书写和阅读，常用八进制数和十六进制数进行缩写。

3. 八进制(Octonary)

八进制中，每个数位上使用的数码为 0，1，2，3，4，5，6，7，其进位基数 R 为 8，计数规律为“逢八进一”。八进制数用下标 O 或 8 标记。任意一个八进制数可以按权展开为 $(N)_O=\sum_{i=-m}^{n-1}a_i8^i$。例如：

$$(352.14)_O=3\times8^2+5\times8^1+2\times8^0+1\times8^{-1}+4\times8^{-2}$$

由于 $2^3=8$，因而 3 位二进制数可以用 1 位八进制数表示。换句话说，1 位八进制数可以表示 3 位二进制数。

4. 十六进制(Hexadecimal)

十六进制中，每个数位上使用的数码为 0，1，2，…，9，A，B，C，D，E，F，其进位基数 R 为 16，计数规律为“逢十六进一”。十六进制数用下标 H 或 16 标记。任意一个十六进制数可以按权展开为 $(N)_H=\sum_{i=-m}^{n-1}a_i16^i$。例如：

$$\begin{aligned}(BD2.3C)_H&=B\times16^2+D\times16^1+2\times16^0+3\times16^{-1}+C\times16^{-2}\\&=11\times16^2+13\times16^1+2\times16^0+3\times16^{-1}+12\times16^{-2}\end{aligned}$$

由于 $2^4=16$，因而 4 位二进制数可以用 1 位十六进制数表示，从而实现缩写。

在计算机系统中，二进制主要用于机器内部的数据处理，八进制和十六进制主要用于书写程序，十进制主要用于运算最终结果的输出。表 1.1.2 为四种常用数制的对照关系。

表 1.1.2　四种常用数制的对照关系

十进制	二进制	八进制	十六进制	十进制	二进制	八进制	十六进制
0	0	0	0	9	1001	11	9
1	1	1	1	10	1010	12	A
2	10	2	2	11	1011	13	B
3	11	3	3	12	1100	14	C
4	100	4	4	13	1101	15	D
5	101	5	5	14	1110	16	E
6	110	6	6	15	1111	17	F
7	111	7	7	16	10000	20	10
8	1000	10	8	17	10001	21	11

1.1.2　常用数制间的相互转换

数制的转换是一个数从一种进制形式转换为等值的另一种进制形式，其实质是权值的转换。

1. 将 *R* 进制数转换为十进制数

将 R 进制(包括二进制、八进制和十六进制)数转换为十进制数时，只要将 R 进制数按权展开，然后将各项数值相加，便可得到等值的十进制数。

【例 1.1】 将二进制数$(1101.11)_2$转换为十进制数。

解

$$(1101.11)_2=1\times2^3+1\times2^2+0\times2^1+1\times2^0+1\times2^{-1}+1\times2^{-2}$$
$$=8+4+0+1+0.5+0.25=(13.75)_{10}$$

【例 1.2】 将八进制数$(352.14)_8$转换为十进制数。

解

$$(352.14)_8=3\times8^2+5\times8^1+2\times8^0+1\times8^{-1}+4\times8^{-2}$$
$$=192+40+2+0.125+0.0625=(234.1875)_{10}$$

【例 1.3】 将十六进制数$(6B.8)_{16}$转换为十进制数。

解

$$(6B.8)_{16}=6\times16^1+11\times16^0+8\times16^{-1}$$
$$=96+11+0.5=(107.5)_{10}$$

2. 将十进制数转换为 *R* 进制数

将十进制数转换为 R 进制(包括二进制、八进制和十六进制)数时，通常采用基数乘除法，需要将十进制数的整数部分和小数部分分别进行转换，具体如下。

1) 整数部分

对于十进制数，其整数部分可写成

$$(N)_D=a_{n-1}R^{n-1}+a_{n-2}R^{n-2}+\cdots+a_1R^1+a_0R^0 \tag{1.1.1}$$

其中，a_{n-1}，a_{n-2}，…，a_1，a_0为欲转换的 R 进制数的各位数字。将式(1.1.1)两边同除以 R，得

$$\frac{1}{R}(N)_{\mathrm{D}} = a_{n-1}R^{n-2} + a_{n-2}R^{n-3} + \cdots + a_1R^0 + \frac{a_0}{R} \tag{1.1.2}$$

此时，余数为 a_0；将式(1.1.2)两边再同时除以 R，余数为 a_1。不难推知，只要将十进制数连续除以 R，直至商为零，即可反序依次获得 a_{n-1}，a_{n-2}，…，a_1，a_0。

2）小数部分

类似地，对于十进制数，其小数部分可写成

$$(N)_{\mathrm{D}} = a_{-1}R^{-1} + a_{-2}R^{-2} + \cdots + a_{-m}R^{-m} \tag{1.1.3}$$

其中，a_{-1}，a_{-2}，…，a_{-m}为欲转换的R 进制数的各位数字。将式(1.1.3)两边同乘以 R，得

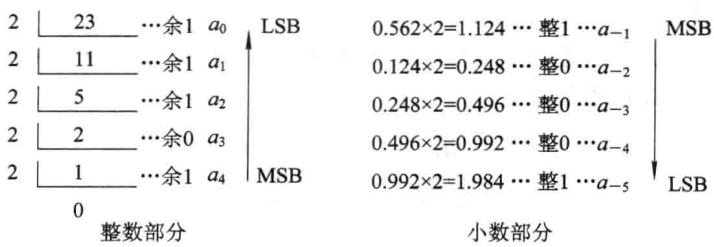
$$R(N)_{\mathrm{D}} = a_{-1}R^0 + a_{-2}R^{-1} + \cdots + a_{-m}R^{-m+1} \tag{1.1.4}$$

此时，乘积的整数为 a_{-1}；将式(1.1.4)两边再同时乘以 R，乘积的整数为 a_{-2}。只要将十进制数连续乘以 R，直至满足误差要求进行“四舍五入”为止，即可顺序依次获得 a_{-1}，a_{-2}，…，a_{-m}。

综上，十进制数转换为 R 进制数的方法可以总结为：整数部分，“除 R 倒取余”；小数部分，“乘 R 正取整”。值得注意的是，从二进制数、八进制数和十六进制数转换为十进制数，或者十进制数转换为二进制整数，都能做到完全准确。但把十进制小数转换为其他进制小数时，除少数可以完全准确外，大多存在误差，这时就需要根据设计精度的要求进行“四舍五入”。

【例 1.4】 将十进制数$(23.562)_{10}$转换为二进制数，要求转换误差 ε 不大于 2^{-6}。

解　题目要求转换误差不大于 2^{-6}，即要求保留小数点后 6 位。转换过程如下：



由于最后的小数 0.984＞0.5，根据“四舍五入”的原则，a_{-6}应为 1。因此

$$(23.562)_{10} \approx (10111.100011)_2$$

十进制数转换为八进制数和十六进制数的方法是：整数部分，“除 8/16 倒取余”；小数部分，“乘 8/16 正取整”，此处不再赘述。

需要指出的是，在实际工作中，若十进制数值较大，转换为二进制时，不必逐次乘除以 2，可通过八或十六进制过渡转换，使转换过程得到简化。另外，熟记二进制数的位权值(如表 1.1.1)，将十进制数与其位权值进行对比，从而获得相应的二进制数，也是常用的方法。

3. 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 $8=2^3$ 和 $16=2^4$，即 3 位二进制数相当于 1 位八进制数，4 位二进制数相当于 1 位十六进制数。

1）二进制数转换为八进制数和十六进制数

二进制数转换为八进制数的方法是：以小数点为中心，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)，然后写出每组等值的八进制数。

二进制数转换为十六进制数的方法是：以小数点为中心，分别向左、向右将二进制数按每 4 位一组分组(不足 4 位的补 0)，然后写出每组等值的十六进制数。

【例 1.5】 将$(11101110010.1011)_2$分别转换为八进制数和十六进制数。

解

$$(11101110010.1011)_2=(\underline{011}\,\underline{101}\,\underline{110}\,\underline{010}.\underline{101}\,\underline{100})_2=(3562.54)_8$$

$$(11101110010.1011)_2=(\underline{0111}\,\underline{0111}\,\underline{0010}.\underline{1011})_2=(772.B)_{16}$$

可见，十六进制数比二进制数位数少，识别方便，在编程中更为常用。

2) 八进制数和十六进制数转换为二进制数

八进制数和十六进制数转换为二进制数的方法与前面介绍的步骤相反，只要按位的高低顺序将每 1 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 1.6】 分别求出$(275.43)_8$、$(6B8.A4)_{16}$的等值二进制数。

解

$$(275.43)_8=(\underline{010}\,\underline{111}\,\underline{101}.\underline{100}\,\underline{011})_2=(10111101.100011)_2$$

$$(6B8.A4)_{16}=(\underline{0110}\,\underline{1011}\,\underline{1000}.\underline{1010}\,\underline{0100})_2=(11010111000.101001)_2$$

1.1.3 二进制数的算术运算

当二进制数用于表示数值大小时，它们之间可以进行算术运算。二进制数的加、减、乘、除四种运算的运算规则与十进制数的运算规则基本相同，也可借助十进制数的运算竖式，其区别仅在于二进制数的进位和借位是“逢二进一，借一当二”。下文为简化书写，不再加注表示二进制数的下标。

【例 1.7】 分别求出二进制数 1010 和 0110 进行加、减、乘、除四种运算的结果。

解 依题意，运算过程如下：

```
  1010        1010
 +0110       -0110
 -----       -----
 10000        0100

  1010          1.101…
 ×0110      110)1010
 -----        110
  0000        ----
  1010        1000
  1010         110
 -----        ----
 0000         1000
-------        110
 0111100       ----
               10
```

由上述运算过程可以看出，二进制数的乘法运算可由被乘数左移与加法运算完成；二进制数的除法运算可由除数右移与减法运算完成。如果能将减法操作转化为某种形式的加法操作，那么二进制数的算术运算就可全部用“移位”和“加法”来实现，从而使数字电路或系统得到简化。

在算术运算中，通常用“+”表示正数(positive number)，用“−”表示负数(negative number)。而在数字系统中，则是将一个数的最高位作为符号位，用“0”表示正数，用“1”表示负数，称为机器数。常用的机器数有原码、反码和补码三种表示方法。

1. 原码

用原码表示带符号的二进制数时，符号位用“0”表示正，用“1”表示负，数值位保持不变。

【例 1.8】 已知 $X=(15)_{10}$，$Y=(-15)_{10}$，字长 $n=5$，求 X 和 Y 的原码。

解

$$[X]_{原}=01111,\quad [Y]_{原}=11111$$

字长为 5 时，0 的原码有两种表示：$[0]_{原}=00000$，$[0]_{原}=10000$。

采用原码表示带符号的二进制数简单易懂，但需要根据参加运算的两个数的符号来确定最终的运算是加法还是减法。若是减法，还需要根据两个数的大小确定被减数和减数，以及判断运算结果的符号，增加了运算的复杂性。

2. 反码

反码的符号位与原码的相同，用“0”表示正，用“1”表示负。正数反码的数值位与原码的数值位相同，而负数反码的数值位是将原码的数值位按位取反。可以验证，一个数反码的反码就是这个数本身。

【例 1.9】 已知 $X=(65)_{10}$，$Y=(-65)_{10}$，字长 $n=8$，求 X 和 Y 的原码、反码。

解

$$[X]_{原}=01000001,\quad [Y]_{原}=11000001$$
$$[X]_{反}=01000001,\quad [Y]_{反}=10111110$$

用反码进行加、减运算时，运算规则如下：

$$[X+Y]_{反}=[X]_{反}+[Y]_{反}$$
$$[X-Y]_{反}=[X]_{反}+[-Y]_{反}$$

反码比原码运算方便，可用加法代替减法，符号位相加后，若出现进位，则送回到最低位相加(循环进位)。但是数值 0 在反码系统中有+0 和−0 之分，给运算器的设计带来了麻烦。

3. 补码

为说明补码原理，先讨论日常生活中的例子。时钟是逢 12 进位，12 点也可看作 0 点。当将时针从 10 点调整到 5 点时有以下两种方法：一种方法是将时针向逆时针方向拨 5 格，相当于做减法，即 10−5=5；另一种方法是将时针向顺时针方向拨 7 格，相当于做加法，即 10+7=12+5。由于时钟以 12 为模，在这个前提下，当总和超过 12 时，可将 12(进位)舍去。于是，减 5 相当于加 7。同理，减 4 可表示成加 8……也称 7 为−5 对模 12 的补码，8 为−4 对模 12 的补码……

补码是计算机中使用最多的一种编码。用补码表示带符号的二进制数时，正数的补码与原码、反码相同；负数的补码是对应的反码在最低位加 1。可以验证，一个数补码的补码就是这个数本身。0 的补码只有一种表示方式。

【例 1.10】 已知 $X=(48)_{10}$，$Y=(-48)_{10}$，字长 $n=8$，求 X 和 Y 的原码、反码、补码。

解

$$[X]_{原}=00110000,\quad [Y]_{原}=10110000$$
$$[X]_{反}=00110000,\quad [Y]_{反}=11001111$$
$$[X]_{补}=00110000,\quad [Y]_{补}=11010000$$

采用补码进行加、减运算时，可将加、减运算通过统一加法实现，得数仍用补码表示，其运算规则如下：

$$[X+Y]_{补}=[X]_{补}+[Y]_{补}$$
$$[X-Y]_{补}=[X]_{补}+[-Y]_{补}$$

【例 1.11】 用二进制补码计算 15+11、16−12、−16+11、−17−17，字长 $n=6$。

解 求出 15、11、16、−12、−16、−17 的补码，然后将各个补码进行加法运算，字长 $n=6$，符号位溢出不保留。具体运算过程如下：

+15	001111	+16	010000
+11	001011	−12	110100
+26	011010	+4	(1) 000100

−16	110000	−17	101111
+11	001011	−17	101111
−5	111011	−34	(1) 011110

观察例 1.11 可发现，−17−17 的计算结果出现错误。在两个同符号数相加时，若它们的绝对值之和超过有效字长所能表示的最大值，则产生溢出，导致计算结果出错。解决溢出问题的办法是进行位扩展。如将例 1.11 中的字长调整为 $n=7$，增加一个数值位即可。表 1.1.3 给出了 4 位二进制整数的原码、补码、反码的对应关系。

表 1.1.3　4 位二进制整数的原码、补码、反码的对应关系

十进制整数	二进制整数	原码	反码	补码	十进制整数	二进制整数	原码	反码	补码
+0	+0000	0000	0000	0000	−0	−0000	1000	1111	0000
+1	+0001	0001	0001	0001	−1	−0001	1001	1110	1111
+2	+0010	0010	0010	0010	−2	−0010	1010	1101	1110
+3	+0011	0011	0011	0011	−3	−0011	1011	1100	1101
+4	+0100	0100	0100	0100	−4	−0100	1100	1011	1100
+5	+0101	0101	0101	0101	−5	−0101	1101	1010	1011
+6	+0110	0110	0110	0110	−6	−0110	1110	1001	1010
+7	+0111	0111	0111	0111	−7	−0111	1111	1000	1001

1.1.4　几种常用的编码

数字系统中除了处理数值以外，还需要处理文字、符号以及一些特定的操作。这就需要将这些信息用二进制的数字符号来表示。用按一定规律排列的多位二进制数码表示十进制数值、字母符号、特定操作等的过程称为编码。这些特定的二进制数码称为代码。此时，这些二进制数码不表示数量大小的含义，也不存在“进位”。n 位的二进制数码共有 2^n 种不同的组合，可以用来代表 2^n 种不同的信息。

1. 二－十进制编码(BCD 码)

用二进制数码表示十进制中的 0～9 这 10 个字符，这些二进制数码称为二－十进制码，简称 BCD(Binary Coded Decimal)码。十进制中的 10 个字符需要 4 位二进制数码来表示，而 4 位二进制数码可有 16 种组合，可以指定其中任意 10 种来表示十进制的 10 个字

符。因此，BCD 码的编码方案形式多样，常用的有 8421 码、2421 码、5421 码、余 3 码等，如表 1.1.4 所示。

表 1.1.4　几种常用的 BCD 码

十进制数	有权码			无权码
	8421 码	2421 码	5421 码	余 3 码
0	0000	0000	0000	0011
1	0001	0001	0001	0100
2	0010	0010	0010	0101
3	0011	0011	0011	0110
4	0100	0100	0100	0111
5	0101	1011	1000	1000
6	0110	1100	1001	1001
7	0111	1101	1010	1010
8	1000	1110	1011	1011
9	1001	1111	1100	1100
各位位权	8421	2421	5421	—

1) 8421 码

8421 码是有权码，各位的权值分别为 8、4、2、1，因为每位上的权固定不变，也称为衡权代码。需要注意的是，该权值虽与自然二进制码的权值相同，但表示不同概念。8421 码只选取了 4 位自然二进制代码的前 10 种组合 0000～1001，其余 6 种组合 1010～1111 是无效的，将它们称为“伪码”(pseudo code)。

用 8421 码表示十进制数时，只要把十进制数的每一位数码分别用 8421 码取代即可。反之，若要获得 8421 码代表的十进制数，只要把 8421 码以小数点为起点，分别向左、向右每 4 位分一组，再写出每一组代码代表的十进制数，并保持原排序即可。

【例 1.12】 求$(312.89)_{10}$的 BCD 码。

解

$$(312.89)_{10}=(0011\ 0001\ 0010.1000\ 1001)_{8421BCD}$$

【例 1.13】 求$(10010111.01100100)_{8421BCD}$所表示的十进制数。

解

$$(10010111.01100100)_{8421BCD}=(97.64)_{10}$$

2) 余 3 码

余 3 码由是 8421 码加 3(0011)得来的。余 3 码的各位无固定权值，属于无权码。若将两个余 3 码相加，其和将比所表示的十进制数对应的二进制数多 6。因此，将余 3 码用于加法运算时，若两数之和正好为 10，等于二进制的 16，并自动产生了向高位的进位。

余 3 码中 0 和 9、1 和 8、2 和 7、3 和 6、4 和 5 的代码互为反码。这种特性称为自补性。具有自补性的代码称为自补码。余 3 码对 10 求补容易，有利于简化 BCD 码的减法运算。

3) 2421 码和 5421 码

2421 码和 5421 码是有权码，各位的权值分别为 2、4、2、1 和 5、4、2、1。这两种编码方案都不是唯一的，表 1.1.4 给出了其中一种方案。2421 码也是一种自补码。5421 码的较