电子信息与电气学科规划教材·电子电气基础课程

数字电路与系统

（第2版）

王 兢 戚金清 主编
李小兵 王开宇 唐 洪 高仁璟 编

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY http://www.phei.com.cn

电子信息与电气学科规划教材·电子电气基础课程

数字电路与系统

（第2版）

王　兢　戚金清　主编

李小兵　王开宇　唐　洪　高仁璟　编

電子工業出版社
Publishing House of Electronics Industry
北京·BEIJING

内 容 简 介

数字电子技术是信息、通信、计算机、控制等领域工程技术人员必须掌握的基本理论和技能，本书主要讲解了数字逻辑基础，逻辑门电路，逻辑代数基础，组合逻辑电路，触发器，时序逻辑电路，脉冲波形的产生与变换，数字系统设计基础，数模与模数转换，半导体存储器及可编程逻辑器件，硬件描述语言 VHDL 等内容。全书包含大量例题和习题，便于学生理解所学概念。

本书不仅是一本面向信息与电气大类专业的本科生基础课教材，而且是电类工程技术人员的合适参考用书。

图书在版编目(CIP)数据

数字电路与系统/王兢，戚金清主编. —2 版. —北京：电子工业出版社，2011.2
电子信息与电气学科规划教材·电子电气基础课程
ISBN 978-7-121-12219-4

I. 数… II. ①王… ②戚… III. 数字集成电路－系统设计－高等学校－教材 IV. TN431.2

中国版本图书馆 CIP 数据核字（2011）第 216392 号

策划编辑：许菊芳
责任编辑：许菊芳　　特约编辑：龙继文
印　　刷：北京东光印刷厂
装　　订：三河市皇庄路通装订厂
出版发行：电子工业出版社
　　　　　北京市海淀区万寿路 173 信箱　邮编　100036
开　　本：787×1092　1/16　　印张：19　　字数：487 千字
印　　次：2011 年 2 月第 1 次印刷
印　　数：3000 册　　定价：35.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：（010）88258888。

序

本书在《数字电路与系统》第 1 版的基础上修订而成。在总结了三年多来使用本书教学和学习的经验的基础上，对本书进行了部分提高、增删和修改。为了保持数字电路基本知识和基础理论的连贯性，同时适应信息科学的迅速发展，本书保留了有关数制、逻辑代数、触发器原理、组合逻辑电路、时序逻辑电路、模数与数模转换等原有的基本内容。而对“半导体存储器及可编程逻辑器件”、“VHDL”等章节做了较多改动。同时，增加了让读者更多了解数字电路课程内容在后续课程的用处以及工程应用的背景等内容。

本书在上一版基础上的具体修改包括：（1）第 1 章中单列了“偏移码”一节；（2）第 2 章中删除了半导体二极管和晶体管的开关特性等模拟电路方面的内容，补充了常用芯片介绍；（3）第 10 章中删除了“可编程阵列逻辑 PAL”、“通用阵列逻辑电路 GAL”两节，修改了“复杂可编程逻辑器件 CPLD”，增补了新内容“CPLD/FPGA 设计方法与编程技术”；（4）第 11 章中补充了许多设计例题的仿真时序图；（5）全书各章都增加了例题。

节号前有“*”号的部分是建议作为选讲的内容。在学时较少的情况下，可以删减这些内容，删去这些内容不会影响本书理论体系的完整性和内容的连贯性。感兴趣的同学可自学这部分内容。

参加本次修订工作的人员包括：王兢（第 7、8、9 章），戚金清（第 2、5、10 章），李小兵（第 4 章），王开宇（第 6 章），唐洪（第 1、3 章），高仁璟（第 11 章）。全书由王兢统稿、定稿。作者向参与上一版写作的王洪玉和余隽老师表示衷心的感谢，向为本课程教学大纲提出宝贵意见的蔡惟铮教授表示真诚的感谢，向电子工业出版社的编辑以及帮助本书进行修改、出版、发行的同事们致以诚挚的谢意。

为便于教学，本书配有教学课件，感兴趣的读者可从华信教育资源网 www.hxedu.com.cn 免费下载。

修订后的教材中一定还会存在一些不妥和不完善之处，恳求读者批评指正。

编　者

目　录

第1章　数字逻辑基础

1.1　模拟电路与数字电路

自从人类发现了电现象，模拟电路就诞生了。随着制造工艺的进步，20 世纪初有了电子管，从此模拟电路能够完成功率放大、产生振荡等功能。结合模拟电路理论，模拟电路可进行算术运算、微积分运算，以至于最终造出了模拟计算机，实现了当时人们进行计算机仿真的愿望。然而，随着科技的飞速发展，许多军事和民用科学研究迫切需要进行大量快速计算，模拟计算机则表现出显著的局限性，不能满足人们日益增长的需求。主要表现为：（1）体积庞大、笨重、功耗高。电子管是在气密性封闭容器（一般为玻璃管）内利用电场对真空中的电子流作用以获得信号放大或振荡的。复杂的电路中，众多电子管在一起，必然体积庞大。虽然在出现晶体管后，模拟电路的体积大大减小，但是相对于集成电路而言，模拟电路在体积和功耗上仍然是不可比拟的。（2）可靠性差，容易受干扰。模拟电路的干扰源较多，如电压不稳定、元器件之间的耦合和非线性等。（3）模拟电路的数值计算能力弱，不能满足科学计算的需要。

数字技术早在 19 世纪末就已经有了工程应用，比如电报就是一个简单的二值数字系统。1846 年，英国数学家 George Boole 创立了布尔代数，从此数字逻辑电路有了分析方法和设计方法，为数字技术的后来发展奠定了理论基础。数字电路最初也经历了电子管时期，由电子管的导通和截止来表示数字信号。1946 年出现的第一台电子计算机，就是以电子管为基本元件制成的。但是，由于当时电子管性能指标的缺陷，这台计算机故障频发。从 20 世纪 60 年代开始，随着半导体制造工艺的突破，晶体管诞生了。由于晶体管体积小、功耗低、工作速度快、工作寿命长等特点，数字电路的体积大大缩小，有了小规模数字电路。从此，数字电路的发展进入了快车道。20 世纪 70 年代末，出现了集成电路，可以把成千上万的晶体管、电阻等元器件制造在一块面积很小的芯片上，并且价格大幅度下降，性能大幅度上升，数字技术开始进入国民经济的各行各业中。20 世纪 80 年代后，大规模、超大规模集成电路的生产技术日渐成熟，工作性能等指标取得突破性进展，诞生了微处理器，除了在各学科领域得到广泛使用外，还深入到人们生活的各个方面。进入 21 世纪，数字技术已经成为人们生活中的一部分，与人们朝夕相处，形影不离。数字技术伴随着人类迈进信息社会。

回顾模拟电路与数字电路的发展历程，不难看出数字电路与模拟电路相比，具有以下一些优点：

（1）集成度高、功耗低、计算能力强；

（2）抗干扰能力强，工作性能可靠；

（3）功能多样化，适应性强。

环顾身边的各种电路与系统可以发现：很少有系统是由全模拟电路构成的，也很少有系统是由全数字电路构成的。一般来讲，在一个系统中模拟电路与数字电路会各司其职，紧密配合共同完成设计者指定的任务。模拟电路与数字电路的联系纽带是模数（A/D）、数模（D/A）转换。形象地说，模拟电路、A/D 转换、数字电路三者之间的关系可以用一个鸡蛋来表示，模拟电路是蛋壳，数字电路是蛋黄，而 A/D 转换是介于蛋壳与蛋黄之间的蛋清。三者既有区别，又紧密联系。

1.2 模拟信号与数字信号

自然界中有许多物理量是模拟量，比如电压、电流、温度、压力、流量、速度等，它们在时间和幅度上都是连续变化的，在其变化范围内可以是任意的实数值，称这些连续变化的物理量为模拟量。表示模拟量的信号称为模拟信号，如速度信号常表示为 $v(t)$ 、压力信号常表示为 $p(t)$ 。

自然界中除了模拟量外，还有另外一些物理量在幅度上是离散的。物理量的幅度只能是有限集合中的某一个数值。这样的物理量称为数字量，表示数字量的信号称为数字信号。比如电灯的开关，每次只能处于“开”或“关”两个状态中的一个；二极管工作在导通和截止两个状态；算盘上每个算珠的位置是有限的，可以认为是数字信号；文字是由有限个字（字符）组成的，因此文字也可以认为是数字信号。除了自然界的数字信号，人们在科学研究中常常人为地制造出数字量，比如通信中的数字调制信号，每次发射的符号是有限符号集中的某一个。

1.3 数制

按照进位规则进行计数，即进位的制度，称为数制。一个数制所包含的数字符号的个数，称为该数制的基数（radix）。人们在日常生活中使用的是十进制，有时也采用十二进制、二十四进制、六十进制，比如用于计时的时钟。而在数字系统中多采用二进制，有时也采用八进制或十六进制。

1. 十进制（Decimal）

十进制有 10 个数字符号：0, 1, 2, 3, 4, 5, 6, 7, 8, 9，基数为 10，逢 10 进 1，即 $9+1=10$。任何一个十进制数都可以用这 10 个代码按一定规律排列起来表示。一个数的大小由它的数码大小和数码所在的位置决定。每个数码所处的位置称为“权”，权由基数的乘方表示，十进制的权由 $10^0, 10^1, 10^2, \cdots$ 以及 $10^{-1}, 10^{-2}, 10^{-3}, \cdots$ 表示。例如，8596.41 按权展开为

$$(8596.41)_{10} = 8\times10^3 + 5\times10^2 + 9\times10^1 + 6\times10^0 + 4\times10^{-1} + 1\times10^{-2}$$

数字电路的计数规则一般不直接采用十进制，因为构成计数电路的基本思路是把电路的状态与数码对应起来，如果采用十进制，则需要有 10 个不同的电路状态来与之对应，从而会使数字电路的结构复杂，错误概率增大，工作可靠性变差。数字电路通常采用二进制进行计数。

2．二进制（Binary）

二进制的基数为 2，只有两个数码 0 和 1，逢 2 进 1，即 1 + 1 = 10。二进制数各位的权为基数 2 的乘方（见表 1.1）。

表 1.1　二进制数的权

二进制位数	权	十进制表示	二进制位数	权	十进制表示	二进制位数	权	十进制表示
12	2^{11}	2048	6	2^5	32	−1	2^{-1}	0.5
11	2^{10}	1024	5	2^4	16	−2	2^{-2}	0.25
10	2^9	512	4	2^3	8	−3	2^{-3}	0.125
9	2^8	256	3	2^2	4	−4	2^{-4}	0.062 5
8	2^7	128	2	2^1	2	−5	2^{-5}	0.031 25
7	2^6	64	1	2^0	1	−6	2^{-6}	0.015 625

二进制数$(101101.101)_2$可表示为

$$(101101.101)_2 = 1\times2^5 + 1\times2^3 + 1\times2^2 + 1\times2^0 + 1\times2^{-1} + 1\times2^{-3}$$

数字电路中通常采用二进制，是因为二进制数只有 0 和 1 两个数码，正好对应于低电平和高电平两种电路状态。

3．八进制（Octal）

八进制的基数为 8，有 8 个数码 0, 1, 2, 3, 4, 5, 6, 7，逢 8 进 1, 即 7 + 1=10。八进制数各位的权为基数 8 的乘方。例如八进制数$(374.25)_8$按权展开为

$$(374.25)_8 = 3\times8^2 + 7\times8^1 + 4\times8^0 + 2\times8^{-1} + 5\times8^{-2}$$

4．十六进制（Hexadecimal）

十六进制的基数为 16，有 16 个数码 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F，其中 A~F 分别表示 10~15，逢 16 进 1，即 F + 1=10。各位的权为基数 16 的乘方。例如十六进制数$(\mathrm{D5E8.A3})_{16}$按权展开为

$$(\mathrm{D5E8.A3})_{16} = 13\times16^3 + 5\times16^2 + 14\times16^1 + 8\times16^0 + 10\times16^{-1} + 3\times16^{-2}$$

5．任意进制

r 进制的基数为 r，有 r 个数码 0, 1, 2, …, $(r-1)$，逢 r 进 1。各位的权为 r 的乘方。一个 r 进制数 N 可以按权展开为

$$\begin{aligned}(N)_r &= k_{n-1}r^{n-1} + k_{n-2}r^{n-2} + \cdots + k_1r^1 + k_0r^0 + k_{-1}r^{-1} + k_{-2}r^{-2} + \cdots + k_{-m}r^{-m} \\ &= \sum_{i=-m}^{n-1} k_i r^i\end{aligned}$$

式中，n 为整数部分的位数，m 为小数部分的位数，r^i 为各位的权，k_i 为系数，是各位的数码。注意，整数部分从右向左第 n 位的权为 r^{n-1}，系数为 k_{n-1}；小数部分从左向右第 m 位的权为 r^{-m}，系数为 k_{-m}。

例如，七进制数$(345.61)_7$按权展开为

$$(345.61)_7 = 3\times7^2 + 4\times7^1 + 5\times7^0 + 6\times7^{-1} + 1\times7^{-2}$$

表 1.2 列出了十进制数 0~17 所对应的二进制数、八进制数和十六进制数。

表 1.2 几种数制之间的关系对照表

十进制数	二进制数	八进制数	十六进制数
0	00000	0	0
1	00001	1	1
2	00010	2	2
3	00011	3	3
4	00100	4	4
5	00101	5	5
6	00110	6	6
7	00111	7	7
8	01000	10	8
9	01001	11	9
10	01010	12	A
11	01011	13	B
12	01100	14	C
13	01101	15	D
14	01110	16	E
15	01111	17	F
16	10000	20	10
17	10001	21	11

1.4 数制间的转换

1.4.1 任意进制转换成十进制

从 1.3 节可以看出，将各种进制数按权展开，就完成了向十进制数的转换。

【例 1.1】 将二进制数$(101011.011)_2$转换为十进制数。

解： $(101011.011)_2 = (1\times2^5 + 1\times2^3 + 1\times2^1 + 1\times2^0 + 1\times2^{-2} + 1\times2^{-3})_{10} = (43.375)_{10}$

【例 1.2】 将八进制数$(1047.5)_8$转换为十进制数。

解： $(1047.5)_8 = (1\times8^3 + 4\times8^1 + 7\times8^0 + 5\times8^{-1})_{10} = (551.625)_{10}$

【例 1.3】 将十六进制数$(A6.C)_{16}$转换为十进制数。

解： $(A6.C)_{16} = (10\times16^1 + 6\times16^0 + 12\times16^{-1})_{10} = (166.75)_{10}$

1.4.2 十进制转换成任意进制

转换原则如下：将十进制数的整数部分除以 r 取余数，直到商为 0，将余数逆序排列，得到 r 进制数的整数部分；将十进制数的小数部分乘以 r，取出乘积的整数部分，剩下的小数部分继续乘以 r，直到满足精度要求为止，将乘积的整数部分顺序排列获得 r 进制数的小数部分。

【例 1.4】　将十进制数 45.28 转换成二进制数（取 4 位小数）。

解：

除数	商	余	
2	45	……1	低位
2	22	……0	
2	11	……1	
2	5	……1	
2	2	……0	
2	1	……1	高位
	0		

	整数	乘积
		0.28 × 2
高位	0 ←	(0).56 × 2
	1 ←	(1).12 × 2
	0 ←	(0).24 × 2
低位	0 ←	(0).48

所以　$(45.28)_{10} = (101101.0100)_2$。

【例 1.5】　将十进制数 348.27 转换成八进制数（取两位小数）。

解：

除数	商	余	
8	348	……4	低位
8	43	……3	
8	5	……5	高位
	0		

	整数	乘积
		0.27 × 8
高位	2 ←	(2).16 × 8
低位	1 ←	(1).28

所以　$(348.27)_{10} = (534.21)_8$。

【例 1.6】　将十进制数 4021.78 转换成十六进制数（取两位小数）。

解：

除数	商	余	
16	4021	……5	低位
16	251	……11	
16	15	……15	高位
	0		

$$
\begin{array}{rr}
 & 0.78 \\
\times & 16 \\
\hline
 & 468 \\
 & 78 \\
\hline
\text{高位}\quad 12 \leftarrow & (12).48 \\
\times & 16 \\
\hline
 & 288 \\
 & 48 \\
\hline
\text{低位}\quad 7 \leftarrow & (7).68
\end{array}
$$

所以　$(4021.78)_{10} = (FB5.C7)_{16}$。

1.4.3　二进制与八进制间的转换

八进制数的基数 8 是 2 的幂，即 $8 = 2^3$，因此可用 3 位二进制数表示 1 位八进制数。将二进制数转换成八进制数时，先对二进数进行分组，以小数点为界，向左、右两侧每 3 位分成一组（不够 3 位添 0），然后将每组转换为 1 位八进制数，按组的顺序排列得到所对应的八进制数。将八进制数转换成二进制数时，只需将每个八进制数字展开成 3 位二进制数字即可。

【例 1.7】　将二进制数$(10111101.1101)_2$转换成八进制数。

解：

$$(\underline{010}\ \underline{111}\ \underline{101}.\underline{110}\ \underline{100})_2 = (275.64)_8$$

【例 1.8】　将八进制数$(3641.256)_8$转换成二进制数。

解：

$$(3641.256)_8 = (\underline{11}\ \underline{110}\ \underline{100}\ \underline{001}.\underline{010}\ \underline{101}\ \underline{11})_2$$

1.4.4 二进制与十六进制间的转换

十六进制数的基数 16 是 2 的幂，即 $16 = 2^4$，因此可用 4 位二进制数表示 1 位十六进制数。将二进制数转换成十六进制数时，以小数点为界，向左、右两侧每 4 位分成一组（不够 4 位添 0），每组转换为 1 位十六进制数。将十六进制数转换成二进制数时，只需将每个十六进制数字展开成 4 位二进制数字即可。

【例 1.9】 将二进制数$(101110110100100.1111011)_2$转换成十六进制数。

解：
$$(\underline{0101}\ \underline{1101}\ \underline{1010}\ \underline{0100}.\underline{1111}\ \underline{0110})_2 = (5DA4.F6)_{16}$$

【例 1.10】 将十六进制数$(B2E.57)_{16}$转换成二进制数。

解：
$$(B2E.57)_{16} = (\underline{1011}\ \underline{0010}\ \underline{1110}.\underline{0101}\ \underline{0111})_2$$

由于八进制数和十六进制数书写比二进制数方便，而且很容易与二进制数相互转换，因此在数字电路中有时也使用八进制数或十六进制数。

1.5 代码

代表信息的数码称为**代码**。本节介绍几种常用的二进制代码。

1.5.1 二-十进制代码

若被编码的信息量为 M，用于编码的二进制数为 n 位，则应有 $n \geqslant \log_2 M$ 或 $\ln M$，即 $2^n \geqslant M$。如果用二进制对 0~9 这 10 个十进制数进行编码，则令二进制数的位数为 n，应有 $n \geqslant \log_2 10$ 或 $\ln 10$，应取 $n = 4$。

用 4 位二进制数对 1 位十进制数进行编码，这种编码称为**二-十进制代码**（BCD，Binary Coded Decimal）。这种编码的方法有多种，常用的几种 BCD 码列于表 1.3 中。最常用的是 8421BCD 码，使用了 0000~1001 这 10 个 4 位二进制数，依次表示 10 个十进制数的代码，其中1010~1111 为禁用码。8421BCD 码保持了二进制数位权的特点，为有权码。此外，2421BCD 码、4221BCD 码、5421BCD 码等也是有权码。余 3 码是一种偏移码，是由 8421BCD 码加 3 后得到的。从表 1.3 可以看出，余 3 码的主要特点是：0 与 9、1 与 8、2 与 7、3 与 6、4 与 5 各组数中两数之和均为 1111，即各组数中两数互为反码。

表 1.3 几种 BCD 码

十进制	二进制	8421BCD	2421BCD	4221BCD	5421BCD	余 3 码
0	0000	0000	0000	0000	0000	0011
1	0001	0001	0001	0001	0001	0100
2	0010	0010	0010	0010	0010	0101
3	0011	0011	0011	0011	0011	0110
4	0100	0100	0100	0110	0100	0111
5	0101	0101	0101	0111	1000	1000
6	0110	0110	0110	1100	1001	1001
7	0111	0111	0111	1101	1010	1010
8	1000	1000	1110	1110	1011	1011
9	1001	1001	1111	1111	1100	1100

8421BCD 码与十进制之间的转换是直接完成的，例如

$$(0101\ 1000\ 0111.1001\ 0000\ 0100)_{8421BCD} = (587.904)_{10}$$

$$(3462.58)_{10} = (0011\ 0100\ 0110\ 0010.0101\ 1000)_{8421BCD}$$

注意，“8421BCD”作为下标应明确标注，不能省略，否则会与二进制数混淆。另外，8421BCD 码不能直接转换成二进制数，要先将其转换成十进制数，再由十进制数转换成二进制数。

1.5.2 格雷码

格雷码（Gray Code）有许多种，表1.4给出了典型格雷码的编码顺序。各种格雷码的共同特点是：任意两个相邻码之间只有 1 位不同。在典型的 n 位格雷码中，0 和最大数（2^n-1）之间也只有 1 位不同，所以它是一种循环码。格雷码的这个特点使它在传输过程中引起的误差较小。例如，7 的二进制码为 0111，8 的二进制码为 1000。在 7 和 8 的边界上，二进制码的 4 位数都发生变化，都处于模糊状态。而格雷码中 7 为 0100，8 为 1100，在二者边界上仅存在 1 位发生变化，带来的误差不会大于 1（即 7 和 8 之差）。

表 1.4 格雷码

十进制	二进制	格雷码	十进制	二进制	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

1.5.3 字符代码

在数字系统中，0和 1 不仅可以代表数，它们的组合还可以表示字母和符号的代码。ASCII 码就是一种常见的字符代码。ASCII 码是美国信息交换标准码（American Standard Code for Information Interchange）。ASCII 码一般有7 位信息码，不同的字符组合代表不同的含义。如 0001101 为信息 CR（Carriage Return，回车）；1111111 为信息 DEL（Delete，删除）；1000001 为信息 A；0100101 为信息%；等等。

1.6 二进制代码的表示法

本节介绍二进制代码的原码、反码和补码表示法。

一个二进制代码的原码就是其本身。把一个二进制代码的原码逐位求反，即 1 变为 0，0 变为 1，就得到该二进制代码的反码。显然，n 位二进制数 N 的反码等于 n 位最大数（n 个 1）与其原码之差，即

$$(N)_{反} = 2^n - 1 - N$$

将一个二进制代码的反码最低有效位加1，就得到该二进制代码的补码。一个 n 位二进制数 N，其补码$(N)_{补}$的定义为

$$(N)_{补} = 2^n - N$$

二进制数的补码可以直接从其原码求得，方法是：二进制数低位（包括小数部分）的右边第一个“1”保持不变（包含此1），向左依次求反。

反码的反码为原码，补码再求补为原码。

【例 1.11】 求二进制代码 11001 的原码、反码、补码。

解： 二进制代码的原码是该代码本身，即 $(11001)_{原码} = 11001$

二进制代码的反码是代码各位依次求反，即 $(11001)_{反码} = 00110$

二进制代码的补码是代码的反码末位加 1，即 $(11001)_{补码} = 00111$

*1.7　带符号二进制数的表示法

一个二进制数可以表示为正数或者负数，方法是在二进制数最高位之前加一个符号位，用 0 表示正数，1 表示负数，通常用逗号将符号位隔开。

1.7.1　二进制正数表示法

正数的原码表示法、反码表示法和补码表示法相同，均为符号位 0 加二进制数本身（即原码）。例如，$(+37)_{10} = 0,100101$。

1.7.2　二进制负数表示法

二进制负数的三种表示方法各有不同，规则如下。

原码表示法：符号位 1 加原码。

反码表示法：符号位 1 加反码。

补码表示法：符号位 1 加补码。

例如，$(37)_{10}$的二进制数为 100101，$(-37)_{10}$的三种二进制表示法分别如下。

原码表示法：1,100101

反码表示法：1,011010

补码表示法：1,011011

1.7.3　带符号二进制数的运算

在数字电路系统中，为了简化运算电路，减法运算用补码相加来完成，乘法运算用加法和移位来实现，除法运算用减法和移位来完成。因此，加法运算是数字电路的基本运算单元。例 1.12~例 1.14 说明了如何利用补码运算将减法化为加法来完成计算。

【例 1.12】 用二进制补码运算求$(1101)_2 - (1010)_2$。

解： 采用补码运算，首先化为带符号数相加的形式

$$(1101)_2-(1010)_2=(0,1101)_2+(1,1010)_2$$

对两数求补

$$[(0,1101)_2]_{补}=0,1101 \qquad [(1,1010)_2]_{补}=1,0110$$

然后两个补码相加并舍去进位

$$0,1101+1,0110=(1)\,0,0011=0,0011$$

这仍是计算结果的补码形式。对此结果再求一次补，得到计算结果的原码

$$(0,0011)_{补}=0,0011$$

所以$(1101)_2-(1010)_2=(0,0011)_2$。

【例 1.13】 用二进制补码运算求$(13)_{10}-(25)_{10}$。

解：$(13)_{10}$表示成补码为$(0,01101)_2$，$(-25)_{10}$表示成补码为$(1,00111)_2$。

将两个补码相加，$(0,01101)_2+(1,00111)_2=(1,10100)_2$。

这仍是计算结果的补码形式。对此结果再求一次补，得到计算结果的原码

$$(1,10100)_{补}=(1,01100)_2=(-12)_{10}$$

【例 1.14】 用二进制补码运算求$(-13)_{10}-(25)_{10}$。

解：$(-13)_{10}$表示成补码为$(1,10011)_2$，$(-25)_{10}$表示成补码为$(1,00111)_2$。

将两个补码相加，$(1,10011)_2+(1,00111)_2=(10,11010)_2$。由于两个被加数都是 6 位的（包括符号位），运算结果也应该是 6 位的。所以，$(10,11010)_2$的最高位溢出，运算结果为$(0,11010)_2$。此数是一个正数，显然运算结果是错误的。

$(-13)_{10}-(25)_{10}$的运算结果应该是$(-38)_{10}$。用二进制表示至少用 7 位数字（包括符号位）。可见，出错的原因是参与运算的字长不足。解决办法是增加字长，在两个被加数的原码最高位加一个 0，再按相同方法继续运算，过程如下。

用 7 位数字表示$(-13)_{10}$，原码为$(1,001101)_2$，补码为$(1,110011)_2$。同理，$(-25)_{10}$的原码为$(1,011001)_2$，补码为$(1,100111)_2$。两个补码相加，$(1,110011)_2+(1,100111)_2=(11,011010)_2$，高位溢出后为$(1,011010)_2$。

这仍是计算结果的补码形式。对此结果再求一次补，得到计算结果的原码

$$[(1,011010)_2]_{补}=(1,100110)_2=(-38)_{10}$$

1.8 偏移码

能够表示信号的幅值和极性的代码，称为双极性码。比如，1.7 节的带符号二进制数既可以表示数的大小也可以表示数的极性。常见的双极性码有原码、反码、补码，还有偏移码。偏移码因带符号的二进制数偏移而得名，它是将带符号二进制数的补码的符号位取反而得。因此，偏移码是最容易实现的双极性码之一。表 1.5 给出了几组原码、补码与偏移码之间的对应关系。

表 1.5 原码、补码与偏移码

十进制	原码	补码	偏移码
3	0,11	0,11	1,11
2	0,10	0,10	1,10
1	0,01	0,01	1,01
0	0,00	0,00	1,00
−0	1,00	1,00	0,00
−1	1,01	1,11	0,11
−2	1,10	1,10	0,10
−3	1,11	1,01	0,01

习题

1.1 什么是数字电路？与模拟电路相比，数字电路具有哪些特点？

1.2 举例说明我们身边的模拟信号和数字信号。

1.3 把下列二进制数转换成十进制数。

（1）$(11000101)_2$　（2）$(0.01001)_2$　（3）$(1010.001)_2$

1.4 把下列十进制数转换成二进制数。

（1）$(12.0625)_{10}$　（2）$(127.25)_{10}$　（3）$(101)_{10}$

1.5 把二进制数$(110101111.110)_2$分别转换成十进制数、八进制数和十六进制数。

1.6 把八进制数$(623.77)_8$分别转换成十进制数、十六进制数和二进制数。

1.7 把十六进制数$(2AC5.D)_{16}$分别转换成十进制数、八进制数和二进制数。

1.8 把十进制数$(432.13)_{10}$转换成五进制数。

1.9 用 8421BCD 码表示下列十进制数。

（1）$(42.78)_{10}$　（2）$(103.65)_{10}$　（3）$(9.04)_{10}$

1.10 把下列 8421BCD 码表示成十进制数。

（1）$(0101\ 1000)_{8421BCD}$　（2）$(1001\ 0011\ 0101)_{8421BCD}$

（3）$(0011\ 0100.0111\ 0001)_{8421BCD}$　（4）$(0111\ 0101.0110)_{8421BCD}$

1.11 把下列 8421BCD 码表示成二进制数。

（1）$(1000)_{8421BCD}$　（2）$(0011\ 0001)_{8421BCD}$

1.12 把$(1001\ 0011)_{8421BCD}$转换成 5421BCD 码，把$(1001\ 0011)_{5421BCD}$转换成 8421BCD 码。

1.13 填空。

（1）$(58.23)_{10}=(\qquad)_2=(\qquad)_8=(\qquad)_{8421BCD}$

（2）$(0001\ 1000\ 1001.0011\ 0101)_{8421BCD}=(\qquad)_{10}=(\qquad)_2$

1.14 填写下表中的空格。

原码	反码	补码	偏移码
1,0010			
	0,1010.01		
		1,11001.10	
1,0000			