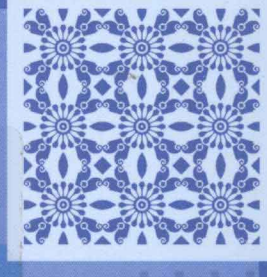
HZ BOOKS
华章教育

高等院校计算机教材系列

DIGITAL CIRCUIT AND LOGICAL DESIGN

数字电路与逻辑设计

胡全连 主编
郑燚 周新 李斌 参编
周琪云 主审



机械工业出版社
China Machine Press

高等院校计算机教材系列

DIGITAL CIRCUIT AND LOGICAL DESIGN

数字电路与逻辑设计

胡全连 主编
郑燚 周新 李斌 参编
周琪云 主审

机械工业出版社
China Machine Press

“数字电路与逻辑设计”是计算机软、硬件各专业的专业基础课。本书覆盖了数字电子技术的全部基础内容，系统地介绍了数字电路的分析与设计理论。本书主要内容有：数字系统基础知识、逻辑代数基础、组合逻辑电路、时序逻辑电路、半导体存储器、可编程逻辑器件、脉冲单元电路、模数及数模转换、Verilog HDL 语言及其编程应用。

本书可作为高等学校计算机及电气信息类各专业的教科书，也可供相关工程技术人员参考。

图书在版编目（CIP）数据

数字电路与逻辑设计 / 胡全连主编. —北京：机械工业出版社，2012.7
（高等院校计算机教材系列）

ISBN 978-7-111-38437-3

Ⅰ. 数…　Ⅱ. 胡…　Ⅲ. 数字电路—逻辑设计—教材　Ⅳ. TN79

中国版本图书馆 CIP 数据核字（2012）第 101954 号

机械工业出版社（北京市西城区百万庄大街 22 号　邮政编码　100037）
责任编辑：曾　珊
北京诚信伟业印刷有限公司印刷
2012 年 8 月第 1 版第 1 次印刷
185mm×260mm • 15.25 印张
标准书号：ISBN 978-7-111-38437-3
定价：29.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
客服热线：（010）88378991；88361066
购书热线：（010）68326294；88379649；68995259
投稿热线：（010）88379604
读者信箱：hzjsj@hzbook.com

前　言

本书是作者在总结多年从事“数字电路与逻辑设计”课程教学工作经验的基础上，结合电子技术的发展趋势及后续课程的需要编写的。考虑到“数字电路与逻辑设计”是计算机各相关专业的专业基础课，掌握数字系统的基本概念和基础知识，建立数字系统的分析与设计的思维方式，对后续课程的学习具有非常重要的影响；同时，考虑到教材面向的对象主要是没有学习过“电路分析”、“模拟电子电路”等先导课程的大学一年级学生，因此，本教材以逻辑代数理论、逻辑门外部性质为基础，以数字逻辑的基础理论及基本电路的分析、设计为重点，注重概念的清晰与内容的实际应用，强化各种规模逻辑器件的外部功能及其应用，淡化了其内部结构的理论分析，压缩了脉冲电路的内容。由于 Verilog HDL 语言简单易学，且在 IC 设计领域更为流行，因此本书以 Verilog HDL 语言为例介绍了硬件描述语言及其编程应用。

全书共分 8 章。第 1 章绪论，主要介绍常用的数制及其转换、数字系统中常用的编码。第 2 章逻辑代数基础，主要介绍逻辑代数中的基本规律、逻辑函数描述方法及逻辑函数的化简。第 3 章组合逻辑电路，主要介绍逻辑门、组合逻辑电路的分析与设计、常用组合逻辑芯片及其应用。第 4 章时序逻辑电路，主要介绍触发器、时序逻辑电路的分析与设计、常用时序逻辑芯片及其应用。第 5 章半导体存储器，主要介绍半导体存储器的工作原理及其应用。第 6 章可编程逻辑器件，主要介绍可编程逻辑器件的原理及其应用。第 7 章脉冲单元电路，主要介绍施密特触发器、单稳态触发器和多谐振荡器的基本原理以及 555 定时器的应用。第 8 章模数及数模转换，主要介绍 A/D、D/A 转换器的原理及应用。在本书附录部分介绍了 Verilog HDL 语言及其编程应用。

本教材由胡全连、郑燚、周新、李斌编著，具体分工如下：第 1 章、第 4 章由周新执笔，第 2 章、第 3 章由胡全连执笔，第 5 章、第 6 章由李斌执笔，第 7 章、第 8 章及附录部分由郑燚执笔。江西师范大学周琪云教授担任主审。本书在编写过程中，江西师范大学谢芳森教授提出了许多宝贵意见和建议，本书的出版还得到机械工业出版社编辑的大力支持和帮助，在此深表感谢。

作者在编写本教材的过程中，参阅了大量相关教材和专著，在此向各位原著作者致敬。

由于作者水平有限，加之时间仓促，书中难免存在一些缺点和错误，殷切希望广大读者批评指正。

作　者

2012 年 3 月

目　录

第1章 绪 论

本章主要介绍了数字电路的概念，数字电路中常用的数制与码制及其相关内容。数字是一种用来表示数的书写符号。不同的记数系统可以使用相同的数字，如十进制和二进制都会用到数字“0”和“1”。用数字信号完成对数字量进行算术运算和逻辑运算的电路称为数字电路，或称为数字系统。

1.1 概述

1.1.1 数字信号与数字电路

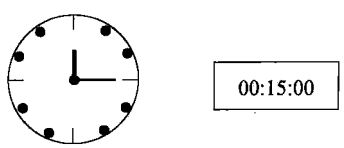
数字信号指幅度的取值是离散的，幅值表示被限制在有限个数值之内。比如二进制码的“0”和“1”就是一种数字信号。二进制码受干扰的影响小，易于数字电路进行处理，所以得到了广泛的应用。

时钟的表示方式大致可以分为模拟式（即指针式）和数字式。模拟式如图 1-1a 所示，用表盘上连续走动的长针和短针相对应，分别形成的角度（模拟量）表示时间。数字式是以秒、分为单位，如图 1-1b 所示，用离散的数字（数字量）表示时间。这两种表示方法充分体现了模拟和数字的特征。

信号分为模拟信号和数字信号这两种。

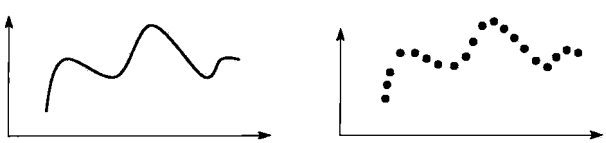
模拟信号：在时间上和数值上连续的信号，如图 1-2a 所示。对模拟信号进行传输、处理的电子线路称为模拟电路。

数字信号：在时间上和数值上不连续的（即离散的）信号，如图 1-2b 所示。对数字信号进行传输、处理的电子线路称为数字电路。



a）模拟式时钟　　b）数字式时钟

图 1-1　时钟



a）模拟信号　　b）数字信号

图 1-2　信号

数字电路或数字集成电路是由许多逻辑门组成的复杂电路。与模拟电路相比，它主要进行数字信号的处理（即信号由 0 与 1 两个状态表示），因此抗干扰能力较强。数字集成电路有各种门电路、触发器以及由它们构成的各种组合逻辑电路和时序逻辑电路。一个数字系统一般由控制部件和运算部件组成。在时钟脉冲的驱动下，控制部件控制运算部件完成所要执行的动作。通过模拟数字转换器、数字模拟转换器，数字电路可以和模拟电路相互连接。数字量电信号的表示不像模拟信号那样简单。通常使用电压高或低表示数字量。电压的高和低这两种状态分别对应于脉冲波的有和无，在水平坐标上纵向或横向排列，这些脉冲的组合可以表示较多的信息。例如，表现任意数值时，把该数值转换为由“1”和“0”构成的二进制数，“1”对应有脉冲，“0”对应无脉

冲，即用“1011…”这样的脉冲序列表示。将量化值转换为二进制值（例如，十进制值的 45 用二进制值表示为 00101101），“1”部分对应有脉冲，“0”部分对应无脉冲，就这样进行排列，做成对应各样本值的数字信号，从而实现用二进制值表现原信号。图 1-3 是把模拟信号的一个样本值用 8 个脉冲的有无表示的示意图。数字处理就是传送、加工如上所述的数字信号，进行各种各样的操作的过程。

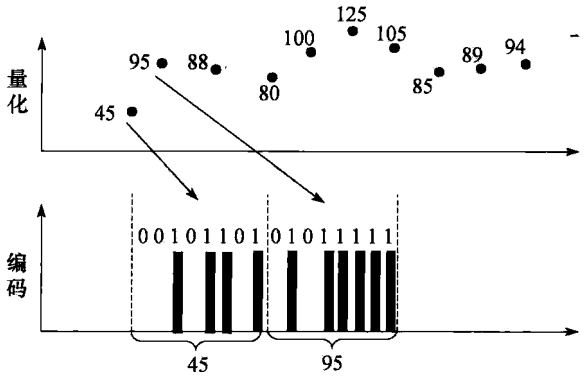


图 1-3 模拟信号的量化编码

电子设备处理信息从模拟方式转变成数字方式，其原因主要在以下几个方面：

- 稳定性好。数字电路不像模拟电路那样易受噪声的干扰。
- 可靠性高。数字电路中只需分辨出信号的有与无，故电路的组件参数可以允许有较大的变化（漂移）范围。
- 能长期存储。数字信息可以利用某种媒介（如磁带、磁盘、光盘等）进行长时期的存储。
- 便于计算机处理。数字信号的输出除了具有直观、准确的优点外，最主要的还是便于利用电子计算机来进行信息处理。
- 便于高度集成化。由于数字电路中基本单元电路的结构比较简单，而且又允许组件有较大的分散性，这就使得把众多基本电子单元集成在同一块硅片上成为可能，同时又能达到大批量生产所需要的合格率。

1.1.2 典型数字系统——数字电子计算机

数字电子计算机综合了数字技术的特点。它进行数字的逻辑性处理，只要得到数值就可以进行逻辑性描述，再复杂的运算也可以实现，如图 1-4 所示。

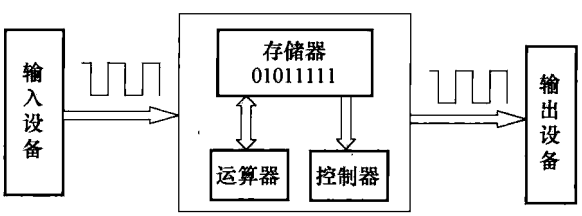


图 1-4 计算机进行数字处理的流程

1.2 数制及其转换

数制是使用一组数字符号来表示数的体系。最初的电子计算机由开关电路组成，只有两种状态：开（1）及关（0），并以此引申出各种复杂的逻辑变化。

0 和 1 的记数系统及数据处理方式是计算机和数字电子技术的基础，因此需要对 0 和 1 的记数系统及数据处理方式进行研究，掌握该系统的相关规律以及它和其他记数系统之间的关系，为理解计算机和数字电子系统的工作原理打好基础。

1.2.1 进位记数制

进位制：表示数时，仅用一位数码往往不够用，必须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以及从低位到高位的进位规则称为进位记数制，简称进位制。在相应的记数系统中，数字位置决定了它所表示的值。

基数：进位制的基数，就是在该进位制中可能用到的数码个数。

位权（位的权数）：在某一进位制的数中，每一位的大小都对应着该位上的数码乘上一个固

定的数，这个固定的数就是这一位的权数。权数是一个幂。

一般地，R 进制需要用到 R 个数码，基数是 R，运算规律为逢 R 进一。对数的表示法通常有两种：

1）并列表示法（位置记数法）：

$$(N)_R=(K_{n-1}K_{n-2}\cdots K_1K_0K_{-1}K_{-2}\cdots K_{-m})_R$$

2）多项式表示法（按权展开式）：

$$(N)=K_{n-1}R^{n-1}+K_{n-2}R^{n-2}+\cdots+K_1R^1+K_0R^0+K_{-1}R^{-1}+K_{-2}R^{-2}+\cdots+K_{-m}R^{-m}=\sum_{i=-m}^{n-1}K_iR^i$$

由按权展开式很容易将一个 N 进制数转换为十进制数。

1. 十进制

基数为 10，逢十进一，基本数码为 0、1、2、3、4、5、6、7、8、9，相邻高位是低位权的 10 倍。

位置记数法：$(S)_{10}=(a_{n-1}a_{n-2}\cdots a_1a_0a_{-1}a_{-2}\cdots a_{-m})_{10(或D)}$

按权展开式：$(S)_{10}=a_{n-1}\times10^{n-1}+a_{n-2}\times10^{n-2}+\cdots+a_1\times10^1+a_0\times10^0$

$$+a_{-1}\times10^{-1}+a_{-2}\times10^{-2}+\cdots+a_{-m}\times10^{-m}$$

图 1-5 为按权展开式的示意图。

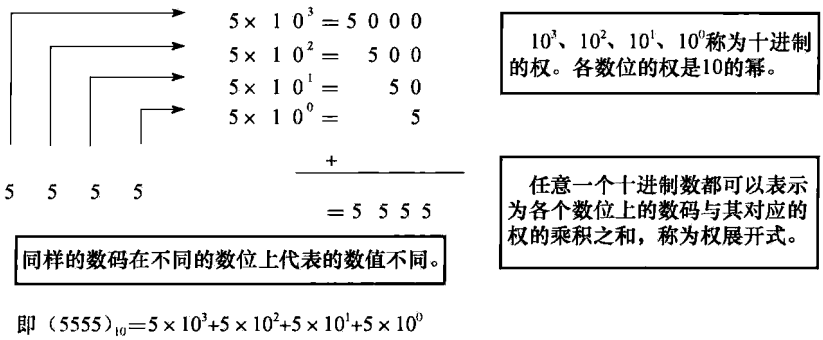


图 1-5　按权展开式的示意图

例 1.1　$(99.999)_{10}=9\times10^1+9\times10^0+9\times10^{-1}+9\times10^{-2}+9\times10^{-3}$

2. 二进制

基数为 2，逢二进一，基本数码为 0、1，相邻高位是低位权的 2 倍。

位置记数法：$(S)_2=(a_{n-1}a_{n-2}\cdots a_1a_0a_{-1}a_{-2}\cdots a_{-m})_{2(或B)}$

按权展开式：$(S)_2=a_{n-1}\times2^{n-1}+a_{n-2}\times2^{n-2}+\cdots+a_1\times2^1+a_0\times2^0$

$$+a_{-1}\times2^{-1}+a_{-2}\times2^{-2}+\cdots+a_{-m}\times2^{-m}$$

例 1.2　$(10011101.10)_B$

$$=1\times2^7+0\times2^6+0\times2^5+1\times2^4+1\times2^3+1\times2^2+0\times2^1+1\times2^0+1\times2^{-1}+0\times2^{-2}$$

二进制数只有 0 和 1 两个数码，即 0 和 1 的记数系统及数据处理方式，它的每一位都可以用电子元件来实现，且运算规则简单，相应的运算电路也容易实现。

- 运算规则：逢二进一，即 1+1=10
- 加法规则：0+0=0，0+1=1，1+0=1，1+1=10
- 乘法规则：0 • 0=0，0 • 1=0，1 • 0=0，1 • 1=1

3. 八进制

基数为 8，基本数码为 0～7。运算规则：逢八进一，即 7+1=10。

例 1.3 $(207.04)_{8(或O)}=2\times8^2+0\times8^1+7\times8^0+0\times8^{-1}+4\times8^{-2}=(135.0625)_{10}$

4. 十六进制

基本数码为 0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F（其中，A 代表 10，B 代表 11，C 代表 12，D 代表 13，E 代表 14，F 代表 15），合计 16 个基本符号来表示数字。相邻高位是低位权的 16 倍。基数是 16。运算规则：逢十六进一，即 F+1=10。

例 1.4 $(D8.A)_{16(或H)}=13\times16^1+8\times16^0+10\times16^{-1}=(216.625)_{10}$

1.2.2 数制间的相互转换

十进制是现实世界普遍采用的数据形式，而在计算机数字系统中普遍采用的是二进制数据形式，二进制数的缺点是书写时位数较长，不便于记忆和阅读。在研究二进制数据形式时，为了书写和阅读的方便，通常使用八进制和十六进制，因此就需要解决不同进制间的相互转换问题。

1. 十进制数与二进制数间的相互转换

（1）二进制数转换成十进制数（按权展开，相加得到）

例 1.5 $(1101001.11)_B$

$=1\times2^6+1\times2^5+1\times2^3+1\times2^0+1\times2^{-1}+1\times2^{-2}$

$=(105.75)_D$

例 1.6 $(1011011.01)_B$

$=1\times2^6+1\times2^4+1\times2^3+1\times2^1+1\times2^0+1\times2^{-2}$

$=32+16+8+2+1+0.25$

$=(59.25)_D$

（2）十进制数转换成二进制数

依据：两数相等，其整数部分和小数部分应分别相等。

1）整数部分：除 2 取余。

两数相等，则除 2 后它们也应相等，且它们的余数部分和整数部分应分别相等。

除基取余法：用目标数制的基数（$R=2$）去除十进制数，第一次相除所得余数为目的数的最低位 a_0，将所得商再除以基数，反复执行上述过程，直到商为“0”，所得余数为目的数的最高位 a_{n-1}。

设 $(A)_{10}=(B_{n-1}B_{n-2}\cdots B_1B_0)_2$，两边除以 2 得：$(A)_{10}/2=(B_{n-1}\times2^{n-2}+B_{n-2}\times2^{n-3}+\cdots+B_1\times2^0+B_0)/2$，所以，$B_0$ 相当于 a 除以 2 后的余数。

同理，通过对 $(A)_{10}/2$ 的商再进行除以 2 取余数，可求出 B_1 的值。如此反复，直到求出 B_{n-1} 的值。

例 1.7 要将十进制整数 47 转换为二进制整数，把它写成如图 1-6 所示形式。

2	47	余数
2	23	……1 (a_0)
2	11	……1 (a_1)
2	5	……1 (a_2)
2	2	……1 (a_3)
2	1	……0 (a_4)
	0	……1 (a_5)

图 1-6 十进制整数转换成二进制整数

2）小数部分：乘 2 取整，直到小数部分为 0 或达到所要求的精度。

两数相等，则乘 2 后它们也应相等，且它们的整数部分和小数部分应分别相等。

乘基取整法：小数乘以目标数制的基数（$R=2$），第一次相乘结果的整数部分为目的数的最高位 a_{-1}，将其小数部分再乘基数依次记

下整数部分 a_{-i}，反复进行下去，直到小数部分为“0”，或满足要求的精度为止（即根据设备字长限制，取有限位的近似值）。

设 $(0.A)_{10}=(0.B_{-1}B_{-2}\cdots B_{-n})_2$，两边乘以 2 得：$(0.A)_{10}\times 2=(B_{-1}+0.B_{-2}\cdots B_{-n})_2$，所以，$B_{-1}$为目的数的最高位。

同理，通过对其小数部分 $(0.A)_{10}\times 2$ 再进行乘以 2 取整数，可求出 B_{-2}的值。如此反复，直到求出 B_{-n}的值。

例 1.8　将十进制小数 0.8125 转换为二进制整数，把它写成如图 1-7 所示形式。

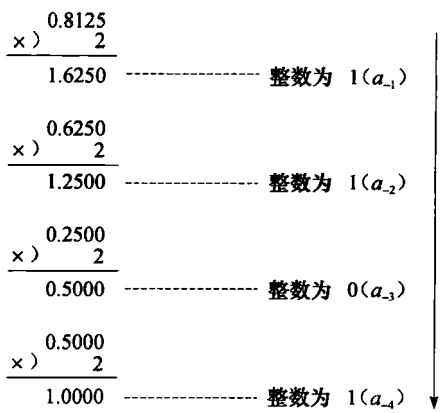


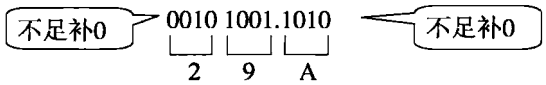
图 1-7　十进制小数转换成二进制整数

2. 二进制数与十六进制数间的相互转换

（1）二进制数转换成十六进制数

以小数点为中心，分别向左或向右每四位二进制数对应一位十六进制数，不足部分补 0。

例 1.9　$(101001.101)_B=(29.A)_H$



（2）十六进制数转换成二进制数

以小数点为中心，分别向左或向右每一位十六进制数对应四位二进制数。

例 1.10　$(F02C.6A)_H=(1111000000101100.01101010)_B$

3. 二进制数与八进制数间的相互转换

（1）二进制数转换成八进制数

以小数点为中心，分别向左或向右每三位二进制数对应一位八进制数，不足部分补 0。

例 1.11　$(101001.10)_B=(51.4)_O$

（2）八进制数转换成二进制数

以小数点为中心，分别向左或向右每一位八进制数对应三位二进制数。

例 1.12　$(502.67)_O=(101000.110111)_B$

表 1-1 给出了几种进制之间数据的对应规律。

表 1-1　几种进制之间数据的对应规律

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	0000	0	0	8	1000	10	8
1	0001	1	1	9	1001	11	9
2	0010	2	2	10	1010	12	A
3	0011	3	3	11	1011	13	B
4	0100	4	4	12	1100	14	C
5	0101	5	5	13	1101	15	D
6	0110	6	6	14	1110	16	E
7	0111	7	7	15	1111	17	F

1.3 带符号数的代码表示

当一个数送入计算机进行运算处理时，首先将其转换为二进制数，同时还要解决数据的正负问题。前面讨论的数据都没有考虑二进制数的符号，一般认为其为正数，事实上不带符号的数是数的绝对值，在绝对值前加上表示正负的符号（+/−）就成了带符号数。一个数由两部分组成：一部分是表示数的符号，另一部分是表示数的数值。

1.3.1 真值与机器码

1. 真值

一般地，直接用正号“+”和负号“−”来表示符号的二进制数，称为符号数的真值。数的真值形式是一种原始形式，在数字计算机中无法直接用（+/−）号表示数据，这给数字计算机的数据表示带来了困难。

2. 机器码

数字计算机中只能表示 0 和 1，能否用 0 和 1 这两个代码来表示正号“+”和负号“−”呢？当将符号数值化之后，便可以在计算机中使用它了。因此在计算机中使用的符号数便称为机器数。在研究解决数据运算问题的过程中，形成机器数的 3 种表示形式，即原码、反码和补码。在解决小数点表示的问题之前，把数据分为整数和小数单独表示：

符号位	数值部分

（1）*原码表示法*

将带符号数的数值部分用二进制数表示，符号部分用 0 表示“+”，用 1 表示“−”，这样形成的一组二进制数叫做原带符号数（也称真值）的原码。

1）当 S 为整数时：

$$[S]_{原}=\begin{cases}S & 0\leqslant S<2^{n-1}\\ 2^{n-1}-S & -2^{n-1}<S\leqslant 0\end{cases}$$

2）当 S 为小数时：

$$[S]_{原}=\begin{cases}S & 0\leqslant S<1\\ 1-S & -1<S\leqslant 0\end{cases}$$

例 1.13 N1＝+10011，N2＝−01010，求 $[N1]_{原}$ 和 $[N2]_{原}$。

解：$[N1]_{原}=010011$，$[N2]_{原}=2^5-(-01010)=101010$

在机器中的表示如下：

$[N1]_{原}$

0	1	0	0	1	1

$[N2]_{原}$

1	0	1	0	1	0

例 1.14 N1＝+0.10011，N2＝−0.01010，求 $[N1]_{原}$ 和 $[N2]_{原}$。

解：$[N1]_{原}=0.10011$，$[N2]_{原}=1-(-0.01010)=1.01010$

0 的原码表示形式有两种：

$$[+0]_{原}=00\cdots00 \text{ 或 } [+0]_{原}=0.0\cdots00$$

$$[-0]_{原}=10\cdots00 \text{ 或 } [-0]_{原}=1.0\cdots00$$

（2）*反码表示法*

反码：第 1 位表示符号位（0 表示正数，用 1 表示负数），其余各位表示数值部分。正数的反码与原码相同：符号位用 0 表示，数值位值不变。若是负数，则数值部分各位取反。

1）当 S 为整数时：

$$[S]_{反}=\begin{cases}S & 0\leqslant S<2^n\\ 2^{n+1}-1+S & -1<S\leqslant 0\end{cases}$$

$$[S]_{反}=\begin{cases}S & 0\leqslant S<1\\ 2-2^{-n}+S & -1<S\leqslant 0\end{cases}$$

2）当 S 为小数时：

例 1.15　N1＝＋10011，N2＝－01010，N3＝＋0.10011，N4＝－0.01010，求 $[N1]_{反}$ 和 $[N2]_{反}$，$[N3]_{反}$ 和 $[N4]_{反}$。

解：$[N1]_{反}$＝010011，$[N2]_{反}$＝110101，$[N3]_{反}$＝0.10011，$[N4]_{反}$＝1.10101

0 的反码表示形式有两种：

$$[+0]_{反}=00\cdots00\text{ 或}[+0]_{反}=0.0\cdots00$$

$$[-0]_{反}=11\cdots11\text{ 或}[-0]_{反}=1.1\cdots11$$

（3）补码表示法

在补码表示方法中，正数的表示同原码和反码的表示是一样的，而负数的表示却不相同。对于负数，将原码转变成补码的规则是：符号位不变，仍为 1，数值部分变反加 1，即逐位变反，在最低位加 1。一个 n 位的整数 S 和小数（包括一位符号位）的补码一般表达式为：

1）当 S 为整数时：

$$[S]_{补}=\begin{cases}S & 0\leqslant S<2^n\\ 2^{n+1}+S & -2^n<S\leqslant 0\end{cases}$$

2）当 S 为小数时：

$$[S]_{反}=\begin{cases}S & 0\leqslant S<1\\ 2+S & -1<S\leqslant 0\end{cases}$$

例 1.16　N1＝＋10011，N2＝－01010，N3＝＋0.10011，N4＝－0.01010，求 $[N1]_{补}$ 和 $[N2]_{补}$，$[N3]_{补}$ 和 $[N4]_{补}$。

解：$[N1]_{补}$＝010011，$[N2]_{补}$＝110101＋1＝110110

$[N3]_{补}$＝0.10011，$[N4]_{补}$＝1.10101＋0.00001＝1.10110

0 的补码表示形式只有一种：

$$[+0]_{补}=00\cdots00\text{ 或}[+0]_{补}=0.0\cdots00$$

$$[-0]_{补}=00\cdots00\text{ 或}[-0]_{补}=0.0\cdots00$$

1.3.2　机器码的运算

1. 原码运算

符号位不参加运算，只是数值部分进行运算，运算结果的符号位取绝对值大的数的符号。

例 1.17　N1＝－0.0011，N2＝0.1011，求 $[N1+N2]_{原}$。

解：$[N1+N2]_{原}=[(-0.0011)+0.1011]_{原}=0.1000$

2. 反码运算

符号位和数值部分一起运算，$[N1+N2]_{反}=[N1]_{反}+[N2]_{反}$；$[N1-N2]_{反}=[N1]_{反}+[-N2]_{反}$，符号位有向前进位，则最低位加 1。

例 1.18 N1=0.1001，N2=0.0011，求 $[N1+N2]_{反}$。

解：$[N1+N2]_{反}=[N1]_{反}+[N2]_{反}=0.1001+0.0011=0.1100$

3. 补码运算

符号位和数值部分一起运算，$[N1+N2]_{补}=[N1]_{补}+[N2]_{补}$；$[N1-N2]_{补}=[N1]_{补}+[-N2]_{补}$，符号位有向前进位，采用丢掉处理。

例 1.19 N1=−0.1100，N2=−0.0010，求 $[N1+N2]_{补}$。

解：$[N1+N2]_{补}=[N1]_{补}+[N2]_{补}=1.0100+1.1110=1.0110$

1.4 几种常用的代码

1.4.1 二—十进制编码

十进制数有 0～9 共 10 个数码，所以表示 1 位十进制数时，至少需要 4 位二进制数。但 4 位二进制数可以产生 $2^4=16$ 种组合，用 4 位二进制数表示 1 位十进制数，有 6 种组合是多余的。十进制数的二进制编码可以有许多种方法，即有许多种不同的编码方案。表 1-2 列举了目前常用的几种编码方案。

表 1-2 常用的几种编码方案

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

1. 8421 BCD 码

用 4 位自然二进制码中的前 10 个码字来表示十进制数码，因各位的权值依次为 8、4、2、1，故称 8421 BCD 码。由于 8421 码中的每一位的权是固定不变的，按权展开式如下：

$$S=a_3a_2a_1a_0=a_3\times2^3+a_2\times2^2+a_1\times2^1+a_0\times2^0$$

注意：在 8421 BCD 码中，不允许出现 1010～1111 这几个代码，因为在十进制中，没有数码同它们对应。

2. 余 3 码

余 3 码是一种特殊的 8421 码，它是由 8421 BCD 码加 3 后形成的，所以叫做余 3 码。例如，十进制数 7 在 8421 BCD 码中是 0111，在余 3 码中就成为 1010。余 3 码的各位无固定的权。

3. 2421 码

2421 码也是一种带权码，它的 0 和 9、1 和 8、2 和 7、3 和 6、4 和 5 互为反码，这一点和余 3 码相似。只要将 2421 码自身按位求反，就能方便地得到其“对 9 的补数”的 2421 码。2421 码用 4 位二进制数表示 1 位十进制数，其权值依次为 2、4、2、1。

1.4.2 可靠性编码

在代码的形成和传输过程中出现错误时，易于发现和矫正或可以避免错误发生的代码称为可靠性编码。表 1-3 列举了目前常用的几种可靠性编码方案。

表 1-3　几种可靠性编码方案

十进制数	8421 BCD码	格雷码	奇校验码	偶校验码
0	0000	0011	1 0000	0 0000
1	0001	0100	0 0001	1 0001
2	0010	0101	0 0010	1 0010
3	0011	0110	1 0011	0 0011
4	0100	0111	0 0100	1 0100
5	0101	1000	0 1011	1 1011
6	0110	1001	1 1100	0 1100
7	0111	1010	0 1101	1 1101
8	1000	1011	0 1110	1 1110
9	1001	1100	1 1111	0 1111

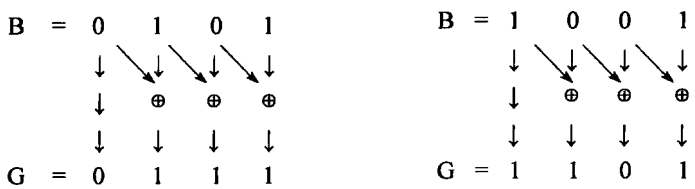
1. 格雷码

格雷码又叫循环码，它有多种编码形式，但它们有一个共同的特点，就是任意两个相邻的代码之间，格雷码仅有一位不同，其余各位均相同。

格雷码是一种无权码，它与二进制数之间的转换关系如下：

设某二进制数为 $B=B_nB_{n-1}\cdots B_1B_0$，其对应的格雷码为 $G=G_nG_{n-1}\cdots G_1G_0$，即 $G_n=B_n$，$G_i=B_{i+1}\oplus B_i(i=0,1,2,\cdots,n-1)$，$\oplus$是异或运算，运算特点是两数相同为 0，不同为 1。

例 1.20　把二进制数 0101 和 1001 转换成格雷码。过程如下：



2. 奇偶校验码

奇偶校验码是一种能检验出二进制信息在传送过程中是否出现错误的代码。这种代码由两部分组成：一部分是奇偶校验位，它使整个代码中 1 的个数按预先的规定成为奇数或偶数；另一部分是信息位，它是需要传送的信息本身。当信息位和校验位中 1 的总个数为奇数时，称为奇校验，而 1 的总个数为偶数时，称为偶校验。表 1-3 表示由 1 位奇偶校验位（首位）及 4 位信息位构成的 5 位奇偶校验码。

奇偶校验码的特点是使每 1 个代码中含有 1 的个数总是奇（偶）数。这样，一旦某一代码在传送过程中出现了误码，使 1 的个数不是奇（偶）数时，就会被发现。

1.4.3　字符编码

计算机处理的数据不仅有数码，还有字母、标点符号、运算符号及其他特殊符号。这些符号都必须用二进制代码来表示，计算机才能直接处理。通常，把用于表示各种字符的二进制代码称为字符代码。

国际上采用的 ASCII 码（美国标准信息交换码）是一种常用的字符代码，是目前国际上最通用的一种字符码。如表 1-4 所示。

表 1-4 ASCII 码编码表

LSD（低 4 位） $B_3B_2B_1B_0$		MSD（高 3 位） $B_6B_5B_4$							
		000	001	010	011	100	101	110	111
0	0000	NUL	DLE	SP	0	@	P	、	p
1	0001	SOH	DC1	!	1	A	Q	a	q
2	0010	STX	DC2	"	2	B	R	b	r
3	0011	ETX	DC3	#	3	C	S	c	s
4	0100	EOT	DC4	$	4	D	T	d	t
5	0101	ENQ	NAK	%	5	E	U	e	u
6	0110	ACK	SYN	&	6	F	V	f	v
7	0111	BEL	ETB	,	7	G	W	g	w
8	1000	BS	CAN	(	8	H	X	h	x
9	1001	HT	EM	)	9	I	Y	i	y
A	1010	LF	SUB	*	:	J	Z	j	z
B	1011	VT	ESC	+	;	K	[	k	{
C	1100	FF	FS	‘	<	L	\	l	|
D	1101	CR	GS	-	=	M	]	m	}
E	1110	SO	RS	.	>	N	↑	n	~
F	1111	SI	US	/	?	O	←	o	DEL

注：其中控制符以及特殊符号用字母缩写表示。

表中一些控制符的含义如下：

NUL Null，空白
SOH Start of Heading，标题开始
STX Start of Text，正文开始
ETX End of Text，正文结束
EOT End of Transmission，传输结束
ENQ Enquiry，询问
ACK Acknowledge，确认
BEL Bell，响铃（告警）
BS Backspace，退一格
HT Horizontal Tabulation，水平列表
LF Line Feed，换行
VT Vertical Tabulation，垂直列表
FF Form Feed，走纸
CR Carriage Return，回车
SO Shift Out，移出
SI Shift In，移入
DLE Data Link Escape，数据链路换码
DC1 Device Control 1，设备控制 1
DC2 Device Control 2，设备控制 2
DC3 Device Control 3，设备控制 3
DC4 Device Control 4，设备控制 4
NAK Negative Acknowledge，否认
SYN Synchronous Idle，同步空传
ETB End of Transmission Block，块结束
CAN Cancel，取消
EM End of Medium，纸尽
SUB Substitute，替换
ESC Escape，脱离
FS File Separator，文件分离符
GS Group Separator，字组分离符
RS Record Separator，记录分离符
US Unit Separator，单元分离符
SP Space，空格
DEL Delete，删除

计算机输出到打印机的字符码就采用 ASCII 码。ASCII 码采用 7 位二进制编码表示十进制符号、英文大小写字母、运算符、控制符以及特殊符号，ASCII 码也可以通过增加 1 位校验位的办法方便地扩展为 8 位，8 位在计算机中称为 1 个字节，这也是 ASCII 码采用 7 位编码的一个重要原因。

本章小结

二进制记数系统和数字编码是计算机和数字电路的基础。本章主要讲述数字系统的基本概念；二进制记数系统以及它和其他记数系统（如十进制、八进制和十六进制记数系统）之间的关