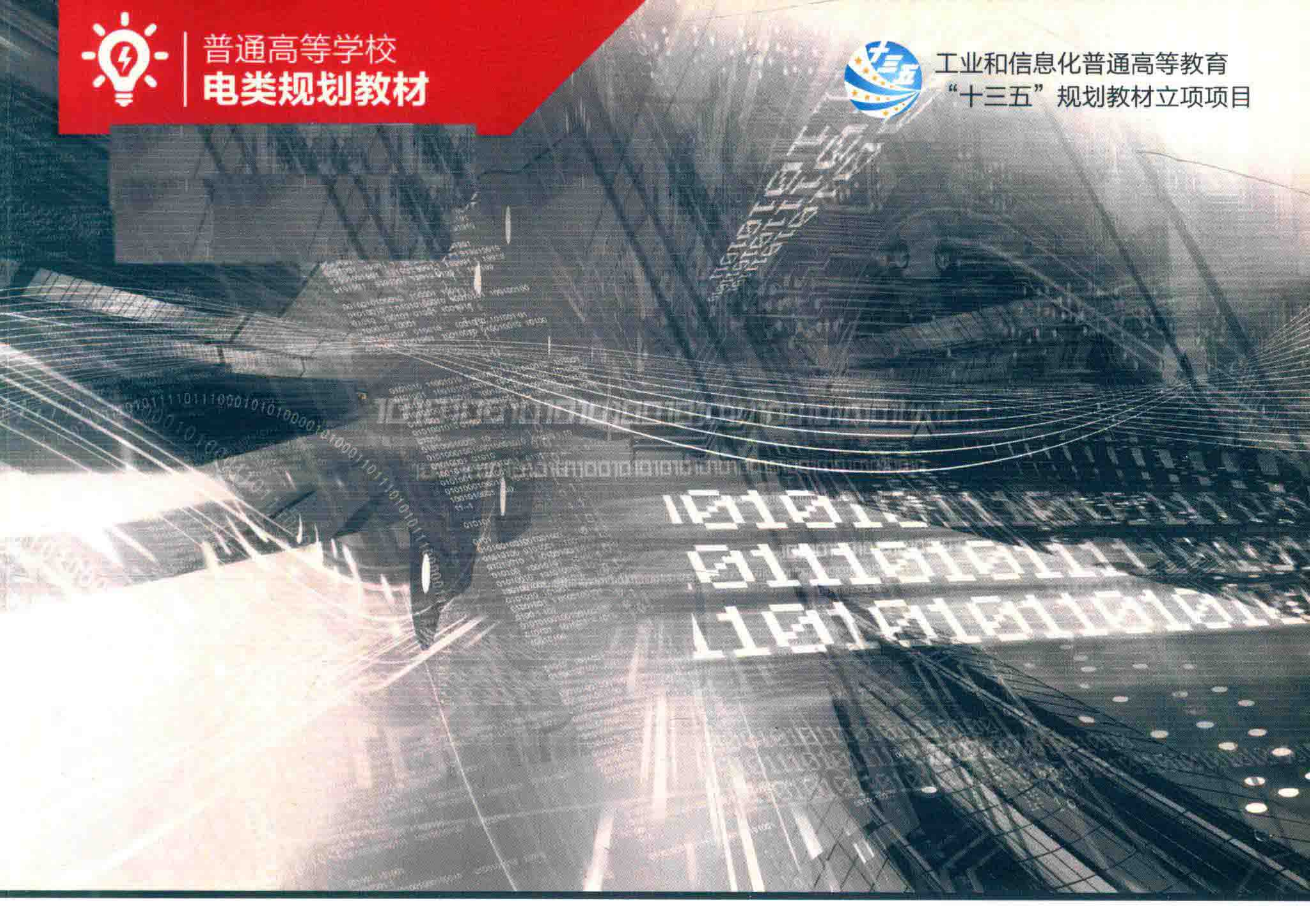
普通高等学校
电类规划教材

工业和信息化普通高等教育
“十三五”规划教材立项项目



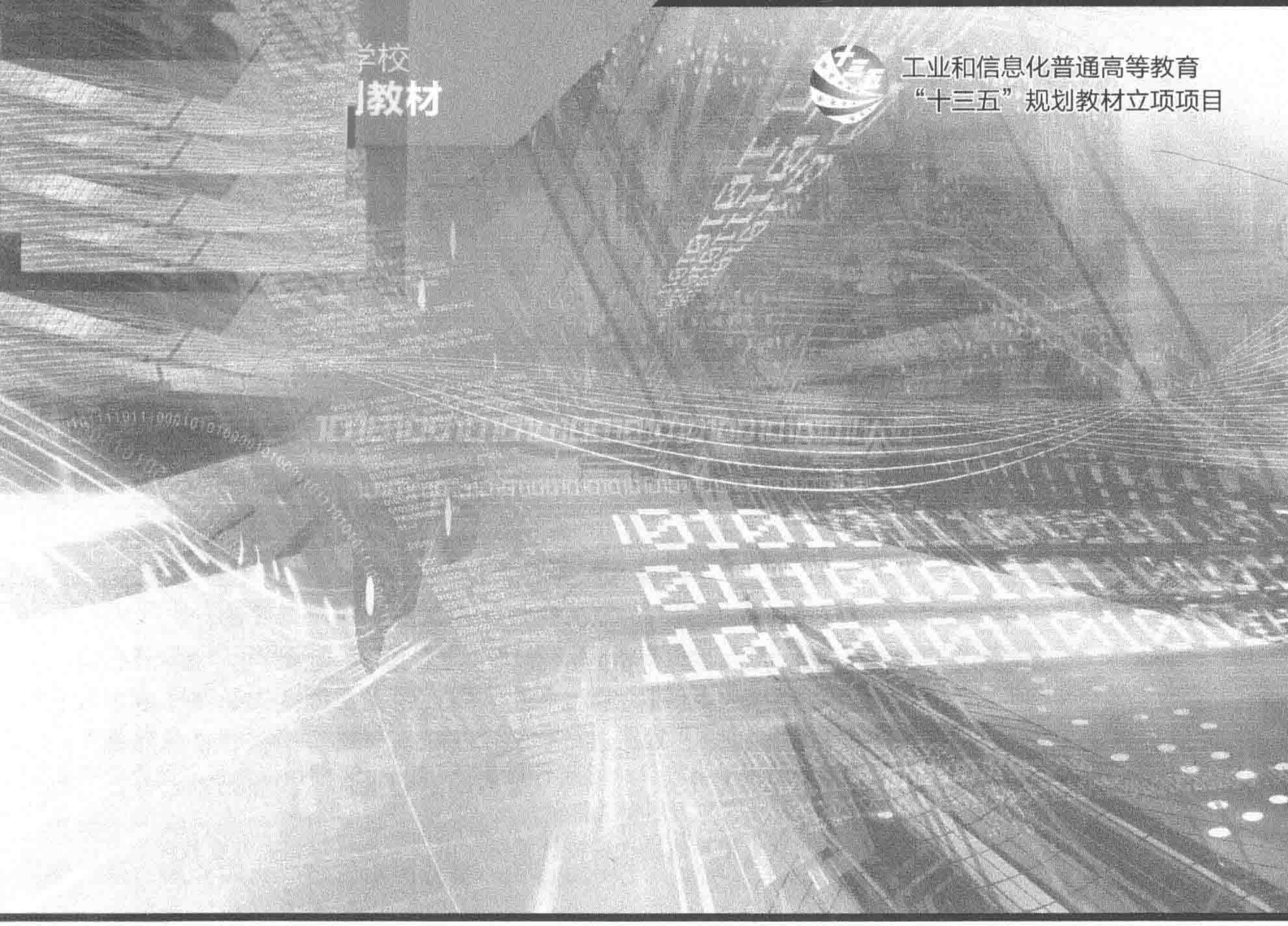
数字电子技术

◎刘琨 主编

◎李克勤 乔瑞芳 副主编

- 以应用为主，剪裁教学内容
- 以案例为驱动，构建知识应用架构
- 更新理念，紧跟课程发展趋势
- 内容充实，增加教材的可读性

中国工信出版集团 人民邮电出版社
POSTS & TELECOM PRESS



数字电子技术

◎刘琨 主编
◎李克勤 乔瑞芳 副主编

人民邮电出版社
北京

图书在版编目（CIP）数据

数字电子技术 / 刘琨主编. -- 北京 : 人民邮电出版社, 2017.8 (2017.10重印)
普通高等学校电类规划教材
ISBN 978-7-115-45373-0

Ⅰ. ①数… Ⅱ. ①刘… 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(2017)第176539号

内 容 提 要

本书根据近年来数字电子技术的新发展和作者多年的教学实践积累，针对数字电子技术课程教学基本要求和学习特点编写而成。本书内容包括数制与码制、逻辑代数、逻辑门电路、组合逻辑电路、触发器、时序逻辑电路、脉冲产生和整形电路、半导体存储器、可编程逻辑器件、A/D与D/A转换、数字电路综合案例等。本书适用于学时较少情况下的教学（50～65 学时）。为了方便教学和自学，本书配有实用的电子课件、思考题与习题参考答案。

本书可作为本科理工类院校大学生和其他层次院校计算机、电子、电气等相关专业教学用书，也可供相关专业工程技术人员参考。

◆ 主　　编　刘　琨
　副 主 编　李克勤　乔瑞芳
　责任编辑　李　召
　责任印制　陈　犇

◆ 人民邮电出版社出版发行　　北京市丰台区成寿寺路 11 号
　邮编　100164　　电子邮件　315@ptpress.com.cn
　网址　http://www.ptpress.com.cn
　北京京华虎彩印刷有限公司印刷

◆ 开本：787×1092　1/16
　印张：15.75　　　　　　2017 年 8 月第 1 版
　字数：382 千字　　　　　2017 年10 月北京第 3 次印刷

定价：45.00 元

读者服务热线：(010)81055256　印装质量热线：(010)81055316
反盗版热线：(010)81055315
广告经营许可证：京东工商广登字 20170147 号

前言

为了适应电子技术的高速发展和社会需求，高等教育对计算机、电子、电气等信息类人才的培养提出了更高要求，教材内容的更新和定位面临新的挑战。“数字电子技术”是计算机、电子、电气等信息类学生的专业基础课，也是实践性很强的技术基础课程。

按照教育部对“电子技术基础课程教学的基本要求”，学校要把学生培养成有一定理论基础，有较强实战能力，有足够创新意识的应用型人才，编者根据多年从事数字电子技术基础教学工作和教学改革实践积累的丰富经验，依照目前高等教育对数字电子技术课程教材更新的需要，纵观国内外相关课程教材，分析应用型专业人才特点，编写了本书。

本书具有以下几个特点。

1．以应用为主，筛选教学内容

本书定位于大众化高等教育背景下应用型、创新型人才培养目标，根据学生可能具备的认知状况、学生素质及能力要求，本着“保基础，重实践，少而精”的原则，对相关内容进行选择，剪裁了一部分纯学术研究内容。编者大幅度删减了对器件内部电路的结构分析和工作原理介绍，如译码器、编码器、数据选择器、计数器等中规模集成器件内部电路分析的相关内容，通过逻辑功能表、逻辑函数、引脚功能介绍等描述方法，讲述器件的逻辑功能及器件的外部特性，以便学生能够正确地使用器件。

本书适合较少学时教学，建议学时为 50～65。

2．以案例为驱动，构筑知识应用架构

以案例为驱动，每章安排例题，前 10 章每章最后安排一个实践案例，将每章知识重点以应用实例的形式加以总结。最后一章将本书主要的知识点整合到一个综合案例中，引导学生学有所用，增强了本书的应用性和实践性。

3．更新理念，紧跟课程发展趋势

本书更新了目前数字电子技术教学领域的一些理念，如第 5 章触发器中用锁存器代替了基本触发器，用电平触发的触发器代替了门控触发器并精心编排了触发器分类和编写顺序。在第 9 章可编程逻辑器件中，重点介绍目前主流使用的可编程逻辑器件——CPLD 和 FPGA 的结构，而将较早发明的 PAL 和 GAL 等内容简化。

4．结构充实，增加教材的可读性

增大图、表比例，强化知识对比和总结，并注重对“难点”内容进行细致推理解析和附图说明。每章设有“内容提要”和“本章小结”，并给出“基本教学要求”。本书附带思考题、

参考答案以及附录，利于学生阅读和自学。

本书统一使用国家标准的图形逻辑符号，附录 A 是逻辑符号对照表，方便学生查找对应的国际惯用图形逻辑符号。

本书分 11 章、2 个附录。由北京师范大学珠海分校刘琨任主编，北京理工大学珠海学院李克勤和吉林大学珠海学院乔瑞芳任副主编。第 1 章和第 4 章由北京理工大学珠海学院喻武龙编写，第 2 章、第 5 章、第 8 章和第 11 章由吉林大学珠海学院乔瑞芳编写，第 3 章、附录 A 和附录 B 由北京师范大学珠海分校彭宇帆编写，第 6 章和第 9 章由北京师范大学珠海分校刘琨编写，第 7 章和第 10 章由北京理工大学珠海学院宫鑫编写。第 2 版修订主要由刘琨、喻武龙、宫鑫和彭宇帆完成。全书由李克勤审稿，刘琨负责统编定稿。

编者参考了许多相关书籍，在此对本书选用参考文献的著作者致以真诚的感谢。

限于编者水平，加之时间仓促，书中难免有不妥之处，敬请业界同仁和读者批评指证。

编　者

2017 年 7 月

目录

第 1 章 数制与码制

内容提要

本章首先介绍模拟信号和数字信号的相关概念，然后详细介绍各种数制的概念，如原码、反码和补码的概念，常见的 BCD 编码、可靠性编码和 ASCII 码等相关内容，并以二进制为重点，讨论了各种进制数相互转换的方法以及有符号二进制数的运算方法。

基本教学要求

1．掌握进制数的概念和不同进制数之间相互转换的方法。

2．掌握原码、反码、补码的概念及相互转换的方法。

3．掌握有符号数的二进制补码运算方法。

4．了解常用的 BCD 码、可靠性码和 ASCII 码。

1.1 模拟信号与数字信号

信号是消息的表现形式，是运载消息的工具，通常表现为随时间变化的某些物理量。从广义上讲，它包含光信号、声音信号和电信号等。例如，古代人点燃烽火而产生的滚滚狼烟属于光信号，人与人之间交谈的话音属于声音信号，太空中的各种无线电波和电话网中的电流则属于电信号等。

按照在幅度上变化的连续性来分，信号可分为模拟信号和数字信号。

1.1.1 模拟信号

模拟信号是指用连续变化的物理量表示的信息，其信号的幅度随时间连续变化。在自然环境下，大多数物理信号都是模拟信号，如温度的变化不会在一瞬间从 25℃跳变到 26℃，而是经历了 25℃～26℃的所有值。图 1.1.1 是气象台记录某一天 24 小时的温度变化情况。这是一条光滑、连续的曲线，所记录的温度随时间变化而连续变化。其他常见的如广播中的声音信号、大气中的电磁波信号、汽车连续加速的信号等，都是值在值域范围内随时间连续变化。

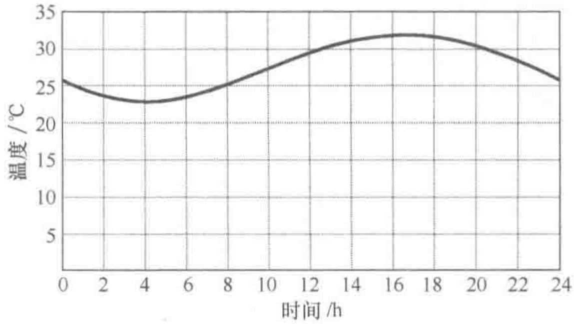


图 1.1.1　典型的模拟信号波形

1.1.2　数字信号

数字信号是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的信号。尽管自然界中大多数物理量是模拟信号，但有些物理量仍可以用数字信号来表示，如电子表的秒信号、生产流水线上记录零件个数的计数信号等。这些信号的变化发生在一系列离散的瞬间，其幅度取值也是离散的，如图 1.1.2 所示。

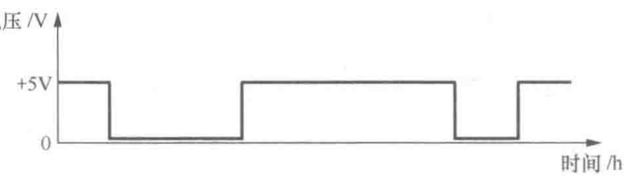


图 1.1.2　典型的数字信号波形

在各种数字信号中，二进制数字信号是一种常见的数字信号，只有两个离散值：0 和 1。

在数字电路中，经常采用 0 和 1 表示两种不同状态的逻辑关系，因此也称为二值数字逻辑。在二值数字逻辑关系中，0 和 1 不代表大小关系，而代表两种不同的状态。如 1 通常代表高电平，称为逻辑 1 或逻辑高电平；0 通常代表低电平，称为逻辑 0 或逻辑低电平。

1.2　进位计数制

1.2.1　数制的概念

在日常生活和工作中，人们经常使用各种进制的计数方式，如广泛使用的十进制数，钟表计时采用的六十进制数，数字系统中采用的二进制数、八进制数和十六进制数等。

进位计数制简称数制，其计数方法是把数划分为不同的数位，由低位到高位逐位累加，当某一数位累计到一定数量之后，产生向高位的进位，同时本位清零。

数制的标识一般可采用数字下标法和字母下标法。数字下标法是在数据结尾加上基数作为下标，如$(32.6)_{10}$，$(10110011)_2$，$(376)_8$，$(2AF6)_{16}$等。字母下标法是在数据结尾采用字母作为下标，其中 D(Decimal)代表十进制数，B(Binary)代表二进制数，O(Octal)代表八进制数，H(Hexadecimal)代表十六进制数，如$(32.6)_D$，$(10110011)_B$，$(376)_O$，$(2AF6)_H$等。

不同数制包含的可使用的数码符号是不同的，而数码符号的个数称为该数制的基数，记作 R 。例如，常见的二进制数码符号是 0 和 1，基数为 2；八进制数码符号是 0～7，基数为

8；十进制数码符号是 0～9，基数为 10；十六进制数码符号是 0～9，A～F，基数为 16。某位的数码为“1”时所表征的数值大小，称为该数位的权值。

利用基数和权值的概念，可以把任意 R 进制数值 D_R 用权值展开法公式展开，即将每位权值与该位的数码相乘后进行叠加，如式（1.2.1）所示。

$$\begin{aligned} D_R &= a_{n-1}\times R^{n-1}+a_{n-2}\times R^{n-2}+\cdots+a_0\times R^0+a_{-1}\times R^{-1}+\cdots+a_{-m}\times R^{-m} \\ &= \sum_{i=-m}^{n-1} a_i\times R^i \end{aligned} \tag{1.2.1}$$

在式（1.2.1）中，n 是整数部分的位数，m 是小数部分的位数，R 是基数，R^i 是第 i 位的权，a_i 是第 i 位的数码，因此某个数位上的数码 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。

1.2.2 十进制数

十进制（Decimal）数共有 0，1，2，3，4，5，6，7，8，9 十个数码符号，每位数累计不能超过 10，计满 10 就要向高位进 1，同时本位清 0；而从高位借来的 1 相当于低位的数 10，遵循“满十进一，借一当十”的运算规则。

对照权值展开式（1.2.1），任意一个十进制数值可展开如下：

$$\begin{aligned} D_{10} &= a_{n-1}\times 10^{n-1}+a_{n-2}\times 10^{n-2}+\cdots+a_0\times 10^0+a_{-1}\times 10^{-1}+\cdots+a_{-m}\times 10^{-m} \\ &= \sum_{i=-m}^{n-1} a_i\times 10^i \end{aligned} \tag{1.2.2}$$

在式（1.2.2）中，a_i 是第 i 位的数码，可以是 0～9 中的任何一个；10 为基数，10 的正次幂对应整数部分，10 的负次幂对应小数部分。

十进制数默认缺省/不做任何标记，或者可以用下标 10 或 D 表示十进制数，如十进制数 68.36 可以表示为 68.36，$(68.36)_{10}$ 或 $(68.36)_{\mathrm{D}}$，并可展开为：

$$(68.36)_{10} = 6\times 10^1+8\times 10^0+3\times 10^{-1}+6\times 10^{-2}$$

1.2.3 二进制数

十进制数虽然在日常生活和工作中应用很方便，但在数字系统中经常采用二进制（Binary）数。数字系统中存在两种不同工作状态的器件很多，如开关的接通和断开，晶体管的导通和截止，电位的高电平和低电平等。用二进制数的 0 和 1 来表示两种不同工作状态不仅可行，并且容易实现，因此二进制数在数字电路系统中得到广泛应用。

二进制数的数码符号只有 0 和 1，进位基数为 2，遵循“满二进一，借一当二”的运算规则。

对照权值展开法公式（1.2.1），任意一个二进制数可展开为：

$$\begin{aligned} D_2 &= a_{n-1}\times 2^{n-1}+a_{n-2}\times 2^{n-2}+\cdots+a_0\times 2^0+a_{-1}\times 2^{-1}+\cdots a_{-m}\times 2^{-m} \\ &= \sum_{i=-m}^{n-1} a_i\times 2^i \end{aligned} \tag{1.2.3}$$

在式（1.2.3）中，a_i 是第 i 位的数码，可以是 0 和 1 中的任何一个；2 为基数，2 的正次幂对应整数部分，2 的负次幂对应小数部分。

二进制数一般用下标 2 或 B 表示，如 $(1011.011)_2$ 或 $(1011.011)_{\mathrm{B}}$，并可展开为：

$$(1011.011)_2 = 1\times2^3 + 0\times2^2 + 1\times2^1 + 1\times2^0 + 0\times2^{-1} + 1\times2^{-2} + 1\times2^{-3}$$

在二进制数中，二进制的数位经常称为“位”，因此在二进制数$(1011.011)_2$中，整数有 4 位，小数有 3 位，并称最左边位为最高有效位（MSB），最右边位为最低有效位（LSB）。

【例 1.2.1】进行二进制数$(1010.011)_2+(111.001)_2$的运算。

解：列出二进制加法算式为

$$\begin{array}{r} 1010.011 \\ +\ \ 111.001 \\ \hline 10001.100 \end{array}$$

所以，$(1010.011)_2+(111.001)_2=(10001.100)_2$。

1.2.4 八进制数

八进制（Octal）数共有 0，1，2，3，4，5，6，7 这 8 个数码符号，进位基数为 8，每计满 8 就要向高位进 1，同时本位清 0；而从高位借来的 1，就相当于低位的数 8，遵循“满八进一，借一当八”的运算规则。

对照权值展开式（1.2.1），任意一个八进制数可展开如下：

$$\begin{aligned} D_8 &= a_{n-1}\times8^{n-1} + a_{n-2}\times8^{n-2} + \cdots + a_0\times8^0 + a_{-1}\times8^{-1} + \cdots a_{-m}\times8^{-m} \\ &= \sum_{i=-m}^{n-1} a_i \times 8^i \end{aligned} \tag{1.2.4}$$

在式（1.2.4）中，a_i是第i位的数码，可以是 0～7 中的任何一个。8 为基数，8 的正次幂对应整数部分，8 的负次幂对应小数部分。

八进制数一般用下标 8 或者 O 表示，如$(76.12)_8$或者$(76.12)_O$，并可展开为：

$$(76.12)_8 = 7\times8^1 + 6\times8^0 + 1\times8^{-1} + 2\times8^{-2}$$

1.2.5 十六进制数

为了简化对二进制数的书写、记忆和交流，经常将四位二进制数用一个十六进制（Hexadecimal）数表示。十六进制数共有 0，1，2，3，4，5，6，7，8，9，A，B，C，D，E，F 共 16 个数码符号，进位基数为 16，每计满 16 就要向高位进 1，同时本位清 0；而从高位借来的 1，就相当于低位的数 16，遵循“满十六进一，借一当十六”的运算规则。

对照权值展开式（1.2.1），任意一个十六进制数可展开如下：

$$\begin{aligned} D_{16} &= a_{n-1}\times16^{n-1} + a_{n-2}\times16^{n-2} + \cdots + a_0\times16^0 + a_{-1}\times16^{-1} + \cdots a_{-m}\times16^{-m} \\ &= \sum_{i=-m}^{n-1} a_i \times 16^i \end{aligned} \tag{1.2.5}$$

在式（1.2.5）中，a_i是第i位的数码，可以是 0～F 中的任何一个；16 为基数，16 的正次幂对应整数部分，16 的负次幂对应小数部分。

十六进制数一般用下标 16 或者字母 H 表示，如$(3A.24)_{16}$或者$(3A.24)_H$，并可展开为：

$$(3A.24)_{16} = 3\times16^1 + 10\times16^0 + 2\times16^{-1} + 4\times16^{-2}$$

【例 1.2.2】 完成十六进制数$(1234)_{16}+(5678)_{16}$的运算。

解：列出十六进制加法算式如下：

$$
\begin{array}{r}
0001\ 0010\ 0011\ 0100 \\
+\,0101\ 0110\ 0111\ 1000 \\
\hline
0100\ 1000\ 1010\ 1100
\end{array}
=
\begin{array}{r}
1234\text{H} \\
+\,5678\text{H} \\
\hline
68\text{ACH}
\end{array}
$$

所以，$(1234)_{16}+(5678)_{6}=(68\text{AC})_{16}$。

1.3　数制的转换

1.3.1　非十进制数转换成十进制数

非十进制数转换成十进制数可按照式（1.2.1）展开，即首先把非十进制数写成按权展开的多项式，然后按十进制数的计数规则对各项相加求和。

【例 1.3.1】　将二进制数$(11010.011)_2$转换成十进制数。

解：将要转换的二进制数按照式（1.2.1）按权展开相加。

$$
\begin{aligned}
(11010.011)_2 &= 1\times 2^4+1\times 2^3+0\times 2^2+1\times 2^1+0\times 2^0+0\times 2^{-1}+1\times 2^{-2}+1\times 2^{-3} \\
&= 16+8+2+0.25+0.125 \\
&= 26.375
\end{aligned}
$$

【例 1.3.2】　将十六进制数$(1\text{AF}.6)_{16}$转换成十进制数。

解：将要转换的十六进制数按照式（1.2.1）按权展开相加。

$$
\begin{aligned}
(1\text{AF}.6)_{16} &= 1\times 16^2+10\times 16^1+15\times 16^0+6\times 16^{-1} \\
&= 256+160+15+0.375 \\
&= 431.375
\end{aligned}
$$

1.3.2　十进制数转换成非十进制数

十进制数转换成非十进制数时，要对十进制数的整数和小数分别进行转换，然后将两个转换结果进行合并即可。

1. 整数部分转换

十进制整数D转换成R进制数可采取基数连除法，具体步骤如下：

① 将整数D除以基数R，记下所除结果的商和余数。

② 将上一步所得的商再继续除以基数R，记下所除结果的商和余数。

③ 重复做步骤②，直至商为 0 结束，并且每进行一次除法都要保留相应的余数。

④ 把保留下来的余数转换成相应的R进制数码，并按照跟运算过程相反的顺序把各个余数排列起来，得到整数D所对应的R进制数。

【例 1.3.3】　将$(367)_{10}$转换成十六进制数。

解：按照十进制整数转换成十六进制数的方法，对基数 16 进行连除并取余数。

$$
\begin{array}{r|l l l l l}
16 & 367 & \cdots & 余15=\text{F} & b_0 & \uparrow 低 \\
\hline
16 & 22 & \cdots & 余6=6 & b_1 & \\
\hline
16 & 1 & \cdots & 余1=1 & b_2 & 高 \\
\hline
 & 0 & & & &
\end{array}
$$

对于十六进制数，当所表示数值大于 9 时需要用字母 A～F 来表示，因此对连除后得到的余数按照从高到低的顺序进行排序，可得：$(367)_{10}=(16F)_{16}$。

2．小数部分转换

十进制小数 M 转换成 R 进制数可采用基数连乘法，具体步骤如下：

① 将十进制小数 M 乘以基数 R，记下所乘结果的整数部分，保留小数部分。

② 将上一步乘积所得结果中的小数部分再乘以基数 R，记下所乘结果的整数部分。

③ 重复步骤②，直至小数部分为 0 或者满足预定精度要求为止，并且每进行一次乘法都要保留相应的整数部分。

④ 将保留下来的整数部分转换成 R 进制的数码，并按照和运算过程相同的顺序排列起来，得到小数 M 所对应的 R 进制数。

【例 1.3.4】 将十进制小数（0.5625）$_{10}$ 转换成二进制数小数。

解：按照十进制小数转换成二进制小数的方法，对基数 2 进行连乘并取整数。

$0.5625\times2=1.125$ … 整数 1 ↓ 高
$0.125\times2=0.250$ … 整数 0
$0.250\times2=0.50$ … 整数 1
$0.50\times2=1.00$ … 整数 1 低

对连乘所得的整数按照从高到低的顺序排列，可得：$(0.5625)_{10}=(0.1001)_2$ 。

【例 1.3.5】 将十进制数$(18.48)_{10}$转换成二进制数小数（要求精确到小数点后 3 位）。

解：该十进制数既有整数部分又有小数部分，因此对于整数部分采取连除取余数法进行转换，对小数部分采取连乘取整数法进行转换，然后合并起来就可得到所求的结果。

2 | 18 … 余0 b_0 ↑ 低
2 | 9 … 余1 b_1
2 | 4 … 余0 b_2
2 | 2 … 余0 b_3
2 | 1 … 余1 b_4 高
0

$(18)_{10}=(10010)_2$

$0.48\times2=0.96$ …… 整数 0 ↓ 高
$0.96\times2=1.92$ …… 整数 1
$0.92\times2=1.84$ …… 整数 1 低

$(0.48)_{10}=(0.011)_2$

将转换后的整数部分和小数部分合并，可得：

$(18.48)_{10}=(10010.011)_2$

1.3.3 非十进制数之间的转换

不同进制数的转换不仅仅是进行十进制数与非十进制数的相互转换，在二进制数、八进制数

和十六进制数这些非十进制数之间也经常要相互转换，不同进制数之间的关系如表 1.3.1 所示。

表 1.3.1　　不同进制数的关系

十进制数	二进制数	八进制数	十六进制数	十进制数	二进制数	八进制数	十六进制数
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

1．二进制数与八进制数相互转换

二进制数转换为八进制数时，整数部分从右往左，每 3 位分成一组，最后剩余不足 3 位时可在左边补齐 0；小数部分从左往右，每 3 位一组，最后剩余不足 3 位时在右边补 0，使其补齐 3 位；最后将整数部分和小数部分的每一组都用等价的八进制数替换就得到所对应的八进制数。

【例 1.3.6】　将二进制数$(10100101.1101)_2$转换为八进制数。

解：将二进制数整数部分从右往左，每 3 位分成一组，小数部分从左往右，每 3 位一组，最后剩余不足 3 位时补 0 凑成 3 位，然后用相应的八进制数替换。

补齐3位　　　　补齐3位

$$\underbrace{\boxed{0}10}_{2}\,\underbrace{100}_{4}\,\underbrace{101}_{5}.\,\underbrace{110}_{6}\,\underbrace{1\boxed{00}}_{4}\qquad \begin{array}{l}\text{二进制}\\ \text{八进制}\end{array}$$

由上可得，$(10100101.1101)_2=(245.64)_8$。

八进制数转换为二进制数时，把八进制数的每一位数都转化为 3 位二进制数即可，其中最高位和最低位的 0 都可以省略。

【例 1.3.7】　将八进制数$(63.27)_8$转换为二进制数。

解：　把八进制数的每一位数都转化为 3 位二进制数即可。

$$\overbrace{110}^{6}\,\overbrace{011}^{3}.\,\overbrace{010}^{2}\,\overbrace{111}^{7}\qquad \begin{array}{l}\text{二进制}\\ \text{八进制}\end{array}$$

由上可得，$(63.27)_8=(110011.010111)_2$。

2．二进制数与十六进制数相互转换

二进制数转换为十六进制数与二进制数转换为八进制数类似，整数部分从右往左开始，每 4 位分成一组，最后剩余不足 4 位时可在左边补 0；小数部分从左往右，每 4 位一组，最后剩余不足 4 位时在右边补 0，使其补齐 4 位；最后将整数部分和小数部分的每一组都用等价的十六进制数替换就得到所对应的十六进制数。

【例 1.3.8】　将二进制数$(101011101.011)_2$ 转换为十六进制数。

解：　将二进制数整数部分从右往左开始，每 4 位分成一组。小数部分从左往右，每 4

位一组，最后剩余不足 4 位时补 0 凑成 4 位，然后用相应的十六进制数替换。

补齐4位　　补齐4位

$$\underbrace{\boxed{000}1}_{1}\,\underbrace{0101}_{5}\,\underbrace{1101}_{D}\,.\,\underbrace{011\boxed{0}}_{6}\qquad \begin{array}{l}\text{二进制}\\ \text{十六进制}\end{array}$$

由上可得，$(101011101.011)_2=(15D.6)_{16}$。

十六进制数转换为二进制数时，把十六进制数的每一位数都转化为 4 位二进制数即可，其中最高位和最低位的 0 都可以省略。

【例 1.3.9】 将十六进制数$(1A4.2D)_{16}$转换为二进制数。

解：

$$\overbrace{0001}^{1}\,\overbrace{1010}^{A}\,\overbrace{0100}^{4}\,.\,\overbrace{0010}^{2}\;\overbrace{1101}^{D}\qquad \begin{array}{l}\text{十六进制}\\ \text{二进制}\end{array}$$

由上可得，$(1A4.2D)_{16}=(110100100.00101101)_2$。

1.4 原码、反码和补码

在十进制数中，有符号数用符号“＋”表示正数，符号“−”表示负数。在计算机系统中，有符号数可分为符号位和数值位两部分。有符号数的最高位作为符号位，用 0 表示“＋”，用 1 表示“−”，其余为数值位部分。

1.4.1 机器数和真值

在计算机系统中，把由符号位和数值位构成的一个数称作为机器数，而数值位部分称为机器数的真值。

例如，X_1=+0101011 B，机器数为 00101011 B，真值为 0101011 B；X_2=−0101011 B，机器数为 10101011 B，真值为 0101011 B。

1.4.2 原码

原码是计算机对有符号数采用二进制表示的一种方法，由符号位和数值位组成。原码的最高位作为符号位，0 表示正数，1 表示负数。数值位表示数值绝对值的大小，绝对值的编码规则与前面讲的无符号数编码规则相同。常见的 8 位有符号数的数值范围是−127～＋127，16 位有符号数的数值范围−32767～+32767。如果运算结果超出可表示的有符号数的范围时，就会发生溢出，使计算结果出错。

例如，X_1=＋1001010B，X_2=−1001010B，则 X_1 原码是$[X_1]_{原}$=01001010 B，X_2 对应的原码是$[X_2]_{原}$=11001010 B。

在原码表示方法中，0 有两种不同的值，即$[+0]_{原}$=00000000 B，$[-0]_{原}$=10000000 B。

1.4.3 反码

反码也是计算机对有符号数采用二进制表示的一种方法，符号位与原码的符号位表示方法一样，对于正数，符号位为 0；对于负数，符号位为 1。反码的数值部分则不同，其值变化与它的符号位有关。对于正数，反码的数值部分和原码数值部分相同；对于负数，反码的数

值部分是原码的数值部分按位取反（即 1 变 0，0 变 1）。

例如：X_1=+ 0000111 B，$[X_1]_{反}$= 0 0000111 B，X_2=−0000111 B，$[X_2]_{反}$= 1 1111000 B。

由于 0 的原码有两种不同的值，因此 0 的反码也有两种不同的值，即$[+0]_{反}$=00000000，$[-0]_{反}$=11111111。

1.4.4 补码

二进制数的算术运算包含加、减、乘、除四种运算，计算机在执行乘法和除法运算时，实际上都是通过做加减法运算来实现的。当设计做减法运算的电路时，必须先比较两个数绝对值的大小，再将绝对值大的数减去绝对值小的数，最后在相减的结果前面加上绝对值较大的数的符号。减法运算逻辑电路复杂，运算速度也比做加法运算慢得多。

为了能使加、减法运算统一用加法运算来实现，人们通过补码来表示有符号数值。补码的引入使得两个数无论相加还是相减，均可通过加法运算来实现。因此，计算机中一切有符号数都是以补码的形式参与算术运算的。

补码也是一种二进制数码，又称为“对 2 的补数”，其表示方法与该数是正数还是负数有关。正数的补码与该数的原码和反码是一样的，对于负数，求补码的方法是：符号位不变，数值部分按位对原码取反，然后在最低位加 1，通常简称为“取反加 1”。

例如，X_1=+ 0000111 B，$[X_1]_{补}$= 0 0000111 B；X_2=−0000111 B，$[X_2]_{补}$= 1 1111001 B。

在补码表示法中，$[+0]_{补}$=0 0000000B，$[-0]_{补}$=$[-0]_{反}$+1=1 1111111+1=1 00000000B，舍弃进位后，$[-0]_{补}$=0 0000000B。由此可见，0 具有唯一的补码$[0]_{补}$=0 0000000B。

【例 1.4.1】 求二进制数 X_1 = ＋0001011B 和 X_2 = −0001011B 对应的原码、反码和补码。

解：对于原码、反码和补码来说，最高位都是符号位，其他位则与数值部分有关。如果该数为正数，则对应的原码、反码和补码按位相同。如果该数为负数，则反码是原码的按位取反，补码则是原码的按位取反加 1。

因此，二进制数＋0001011B 和−0001011B 所对应的原码、反码和补码分别表示如下：

$[X_1]_{原}$ = 00001011B， $[X_1]_{反}$ = 00001011B， $[X_1]_{补}$ = 00001011B。

$[X_2]_{原}$ = 10001011B， $[X_2]_{反}$ = 11110100B， $[X_2]_{补}$ = 111101011B。

1.4.5 有符号数的加减运算

通过采用补码来表示有符号数，可以把两个数的加、减法运算统一为加法运算，即把减法运算通过用$[A]_{补}+[-B]_{补}$的方法实现，因此机器在进行运算时符号位一律按加法参与运算。如果计算的结果产生了进位，则把进位丢弃，保留符号位和数值位即可。

【例 1.4.2】 计算 X_1=36+45，X_2=36−45 和 X_3=−36−45 的结果。

解：对于有符号数的算术运算，采取补码进行，因此先算出各个数的补码，再按补码运算方法进行计算。

（1）X_1=36+45

$$
\begin{array}{rl}
[+36]_{补} & 00100100 \\
+[+45]_{补} & 00101101 \\
\hline
[X_1]_{补} & 01010001
\end{array}
$$