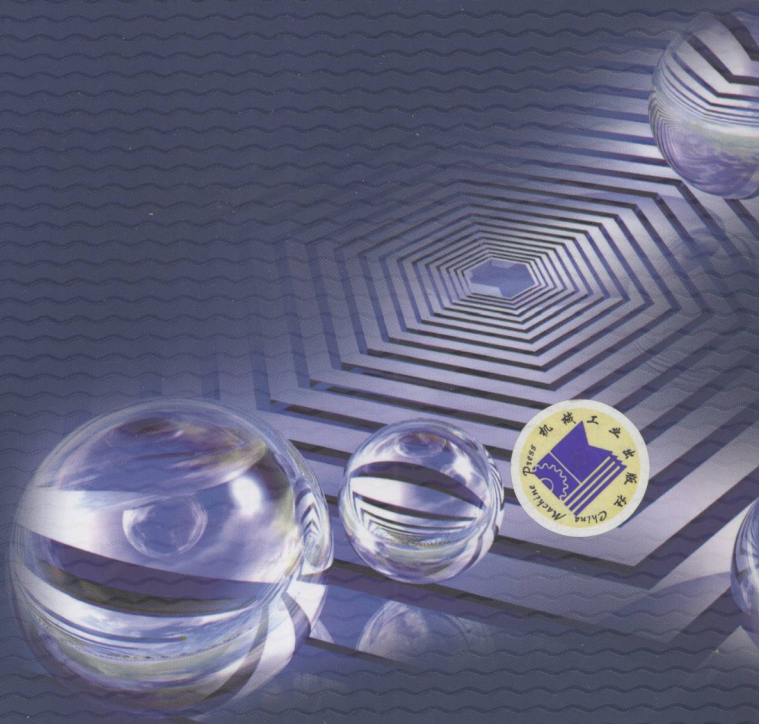
免费提供电子教案

高等院校规划教材
计算机科学与技术系列

数字逻辑原理与应用

郭 军 主编



机械工业出版社
CHINA MACHINE PRESS

高等院校规划教材·计算机科学与技术系列

数字逻辑原理与应用

郭　军　主　编
刘伟明　史　颖　刘　骊　谢　莹　参　编

机械工业出版社

本书从数字逻辑的基础知识——数制和编码入手，系统讲述了逻辑代数基础、逻辑门电路、组合逻辑电路和时序电路的相关知识，进而介绍了可编程逻辑器件的原理和设计方法，以及 Verilog 硬件描述语言。

本书内容全面新颖，既涵盖了数字逻辑传统的内容体系，又合理吸收了现代逻辑电路设计的新技术、新方法，许多内容是作者多年教学科研工作的总结。全书深入浅出，语言流畅，举例丰富，精选了大量习题，并配有电子教案。

本书可作为高等学校计算机、电子与通信等相关专业的教材，也可作为从事逻辑电路设计的工程技术人员的参考书。

图书在版编目（CIP）数据

数字逻辑原理与应用/郭军主编. —北京：机械工业出版社，2009.1
（高等院校规划教材·计算机科学与技术系列）
ISBN 978-7-111-25534-5

Ⅰ.数… Ⅱ.郭… Ⅲ.数字逻辑-高等学校-教材 Ⅳ.TP302.2

中国版本图书馆 CIP 数据核字（2008）第 174985 号

机械工业出版社（北京市百万庄大街 22 号 邮政编码 100037）
责任编辑：陈 皓 常建丽 版式设计：霍永明 责任校对：申春香
责任印制：杨 曦
三河市国英印务有限公司印刷
2009 年 1 月第 1 版第 1 次印刷
184mm×260mm·13 印张·317 千字
0 001— 3 000 册
标准书号：ISBN 978-7-111-25534-5
定价：24.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
销售服务热线电话：（010）68326294
购书热线电话：（010）88379639 88379641 88379643
编辑热线电话：（010）88379753 88379739
封面无防伪标均为盗版

出 版 说 明

计算机技术的发展极大地促进了现代科学技术的发展，明显地加快了社会发展的进程。因此，各国都非常重视计算机教育。

近年来，随着我国信息化建设的全面推进和高等教育的蓬勃发展，高等院校的计算机教育模式也在不断改革，计算机学科的课程体系和教学内容趋于更加科学和合理，计算机教材建设逐渐成熟。在“十五”期间，机械工业出版社组织出版了大量计算机教材，包括“21世纪高等院校计算机教材系列”、“21世纪重点大学规划教材”、“高等院校计算机科学与技术‘十五’规划教材”、“21世纪高等院校应用型规划教材”等，均取得了可喜成果，其中多个品种的教材被评为国家级、省部级的精品教材。

为了进一步满足计算机教育的需求，机械工业出版社策划开发了“高等院校规划教材”。这套教材是在总结我社以往计算机教材出版经验的基础上策划的，同时借鉴了其他出版社同类教材的优点，对我社已有的计算机教材资源进行整合，旨在大幅提高教材质量。我们邀请多所高校的计算机专家、教师及教务部门针对此次计算机教材建设进行了充分的研讨，达成了许多共识，并由此形成了“高等院校规划教材”的体系架构与编写原则，以保证本套教材与各高等院校的办学层次、学科设置和人才培养模式等相匹配，满足其计算机教学的需要。

本套教材包括计算机科学与技术、软件工程、网络工程、信息管理与信息系统、计算机应用技术以及计算机基础教育等系列。其中，计算机科学与技术系列、软件工程系列、网络工程系列和信息管理与信息系统系列是针对高校相应专业方向的课程设置而组织编写的，体系完整，讲解透彻；计算机应用技术系列是针对计算机应用类课程而组织编写的，着重培养学生利用计算机技术解决实际问题的能力；计算机基础教育系列是为大学公共基础课层面的计算机基础教学而设计的，采用通俗易懂的方法讲解计算机的基础理论、常用技术及应用。

本套教材的内容源自致力于教学与科研一线的骨干教师与资深专家的实践经验和研究成果，融合了先进的教学理念，涵盖了计算机领域的核心理论和最新的应用技术，真正在教材体系、内容和方法上做到了创新。另外，本套教材根据实际需要配有电子教案、实验指导或多媒体光盘等教学资源，实现了教材的“立体化”建设。本套教材将随着计算机技术的进步和计算机应用领域的扩展而及时改版，并及时吸纳新兴课程和特色课程的教材。我们将努力把这套教材打造成为国家级或省部级精品教材，为高等院校的计算机教育提供更好的服务。

对于本套教材的组织出版工作，希望计算机教育界的专家和老师能提出宝贵的意见和建议。衷心感谢计算机教育工作者和广大读者的支持与帮助！

机械工业出版社

前　言

“数字逻辑”是计算机、电子与通信等专业的重要专业基础课，也是电子计算机的基础理论之一。

21 世纪以来，我国高等教育教学理念进一步发展，素质教育得到了重视，提出了培养知识面宽、面向应用的高素质人才的教学目标。在这一背景下，数字逻辑课程与其他专业课一样，面临内容增加，学时压缩的矛盾。研究新形势下课程的教学内容，改革优化教学内容体系，是解决问题的基本途径。在此背景下，我们根据计算机教学指导委员会的教学大纲要求，并参考 IEEE/ACM－CS 相关最新教程编写了本书。在传统数字逻辑课程的基础上，增加了近年来涌现出的一些新技术、新方法。考虑到学时压缩问题，教材部分章节内容可选，以适应不同学时计划的要求。

全书共分 9 章，第 1 章为数制和编码，介绍了数字系统中常用的数制及转换、码制和编码。第 2 章为逻辑代数基础，介绍了逻辑代数、逻辑函数及函数化简。第 3 章为逻辑门电路，主要介绍了分立元件门电路和 TTL 门电路。第 4 章为组合逻辑电路，介绍了组合逻辑电路的分析和设计方法以及典型中规模器件的原理和应用。第 5 章为同步时序电路，主要介绍了同步时序电路的结构、分析和设计，同时对触发器和时序机也作了介绍。第 6 章为异步时序电路，介绍了脉冲异步电路和电平异步电路的分析和设计。第 7 章为简单可编程逻辑器件，主要介绍了可编程逻辑器件和 PLD 设计。第 8 章为复杂可编程逻辑器件，介绍了复杂可编程逻辑器件及其设计。第 9 章为数字系统设计初步，主要介绍了数字系统的组成和设计。

完成全部内容教学约需 54 学时，带 * 的内容难度较大，供有能力的读者选学。

本书是作者根据多年教学实践，充分考虑到理科计算机专业特点而编写的。其中 TTL 门电路和数字系统设计部分的内容，是为了加强对理科学生电路知识和工程实践能力培养而编写的。

为了便于教学，本书还配有电子教案，读者可以登录机械工业出版社的教材网站（http：//www. cmpedu. com）免费下载。

本书得到了西北大学新世纪教学基金的支持，编写过程得到了原计算机系硬件教研室全体教师的大力帮助，在此表示感谢。全书由郭军任主编并统稿，刘骊、刘伟明、史颖参与了初稿的编写，谢莹参与了稿件的修改和校订。

本书适合计算机、电子和网络通信等专业作为数字逻辑课程本科和专科教材使用，也可以供相关工程技术人员参考。

由于编者水平有限，书中难免有错误和不妥之处，敬请读者批评指正。读者对本书有任何意见与建议，可发电子邮件至：jsjfw@ mail. machineinfo. gov. cn。

编　者

目　录

出版说明
前言
第 1 章　数制和编码 …… 1
1.1　进位计数制 …… 1
1.1.1　进位计数制的要素 …… 1
1.1.2　二进制 …… 2
1.1.3　八进制和十六进制 …… 3
1.2　数制转换 …… 4
1.2.1　二进制数与十进制数的转换 …… 4
1.2.2　八进制数、十六进制数与二进制数的转换 …… 6
1.3　带符号数的代码表示 …… 7
1.3.1　机器数和真值 …… 7
1.3.2　带符号数的表示方法 …… 8
1.3.3　机器数的定点与浮点表示法 …… 12
1.4　常用的数字字符编码 …… 13
1.4.1　十进制数的代码表示 …… 13
1.4.2　字符的代码表示 …… 16
习题一 …… 17
第 2 章　逻辑代数基础 …… 18
2.1　逻辑代数的基本概念 …… 18
2.1.1　基本逻辑运算 …… 18
2.1.2　运算规则和复合运算 …… 20
2.2　逻辑代数的基本公式及规则 …… 21
2.2.1　逻辑函数的概念 …… 21
2.2.2　逻辑代数的基本公式 …… 22
2.2.3　逻辑代数的重要规则和定理 …… 23
2.3　逻辑函数的代数化简法 …… 24
2.3.1　逻辑函数的“与或”式和“或与”式 …… 24
2.3.2　代数化简法 …… 25
2.4　逻辑函数的卡诺图化简法 …… 27
2.4.1　最小项和最大项 …… 27
2.4.2　卡诺图化简法 …… 30
2.5　含任意项逻辑函数的化简 …… 35
2.5.1　任意项的产生 …… 35
2.5.2　逻辑函数的化简 …… 36

2.6 逻辑函数的表格化简法 …… 37
*2.7 二元决策图和多值逻辑函数 …… 39
习题二 …… 41
第3章 逻辑门电路 …… 43
3.1 集成逻辑电路的分类 …… 43
3.2 分立元件门电路 …… 44
3.2.1 正逻辑与负逻辑 …… 44
3.2.2 二极管“与”门电路 …… 44
3.2.3 二极管“或”门电路 …… 45
3.2.4 三极管“非”门电路 …… 45
3.3 TTL门电路 …… 46
3.3.1 TTL“与非”门的电路组成和工作原理 …… 46
3.3.2 集电极开路“与非”门（OC门） …… 48
3.3.3 三态输出“与非”门电路（TS门） …… 48
3.4 TTL“与非”门的主要外部特性 …… 49
3.5 逻辑门电路的符号与集成化逻辑门 …… 54
*3.6 MOS逻辑门 …… 55
3.6.1 MOS电路 …… 55
3.6.2 CMOS门电路 …… 56
习题三 …… 58
第4章 组合逻辑电路 …… 61
4.1 组合逻辑电路分析 …… 61
4.1.1 组合电路的特点和表示 …… 61
4.1.2 组合电路的分析步骤 …… 61
4.2 组合逻辑电路设计 …… 64
4.2.1 设计的基本步骤 …… 64
4.2.2 设计举例 …… 65
4.2.3 多输出组合逻辑电路的设计 …… 71
4.3 加法器 …… 72
4.3.1 一位加法器 …… 72
4.3.2 多位加法器 …… 76
4.3.3 集成化加法器及其应用 …… 78
4.4 译码器 …… 78
4.4.1 二进制译码器的功能原理 …… 79
4.4.2 集成化的译码器及其应用 …… 83
4.4.3 矩阵式译码器 …… 84
4.4.4 显示译码器 …… 85
4.5 数据选择器 …… 88
4.5.1 数据选择器的结构原理 …… 88

4.5.2 常见的数据选择器及其应用 …… 89
4.6 编码器 …… 92
4.7 数字比较器 …… 95
4.7.1 并行比较器的原理 …… 95
4.7.2 “分段比较”的原理 …… 97
4.8 组合逻辑电路的竞争与冒险 …… 98
4.8.1 竞争与冒险的产生 …… 99
4.8.2 判断冒险 …… 100
4.8.3 消除冒险 …… 101
习题四 …… 102
第5章 同步时序电路 …… 107
5.1 时序机简介 …… 107
5.1.1 时序机的定义 …… 107
5.1.2 时序机的状态表和状态图 …… 108
5.2 触发器 …… 109
5.2.1 RS 型触发器 …… 109
5.2.2 D 触发器 …… 112
5.2.3 JK 触发器 …… 113
5.2.4 T 触发器 …… 115
5.3 同步时序电路的结构与分析 …… 116
5.3.1 同步时序电路的结构 …… 116
5.3.2 同步时序电路的分析 …… 118
5.4 同步时序电路的设计 …… 122
5.4.1 建立原始状态图和状态表 …… 122
5.4.2 状态简化 …… 123
5.4.3 状态分配、求激励函数与输出函数 …… 128
*5.4.4 不完全确定状态的同步时序电路设计 …… 129
5.5 设计举例 …… 133
5.6 集成化的同步时序电路及其应用设计 …… 138
5.6.1 计数器 …… 138
5.6.2 寄存器 …… 139
5.6.3 节拍信号发生器 …… 142
习题五 …… 143
第6章 异步时序电路 …… 147
6.1 脉冲异步电路 …… 147
6.2 电平异步电路 …… 150
6.2.1 电路特点和描述方法 …… 150
6.2.2 电位异步电路的分析 …… 151
6.2.3 电位异步电路的设计 …… 153

*6.3 异步时序电路的险态 …… 155
习题六 …… 156
第7章 简单可编程逻辑器件 …… 158
7.1 可编程只读存储器 …… 158
7.2 可编程逻辑器件 …… 160
7.2.1 可编程逻辑阵列 …… 160
7.2.2 可编程阵列逻辑与通用阵列逻辑 …… 160
7.3 PLD设计方法及步骤 …… 166
7.3.1 PLD器件的设计步骤 …… 166
7.3.2 可编程器件设计软件简介 …… 167
7.3.3 可编程逻辑器件设计举例 …… 170
习题七 …… 174
第8章 复杂可编程逻辑器件 …… 175
8.1 复杂可编程逻辑器件 …… 175
8.2 可编程门阵列 …… 178
8.3 可编程逻辑器件设计简介 …… 180
8.3.1 设计步骤 …… 180
8.3.2 设计进入 …… 180
8.3.3 设计实现 …… 181
8.3.4 模拟仿真 …… 183
8.3.5 器件编程 …… 184
习题八 …… 184
第9章 数字系统设计初步 …… 185
9.1 数字系统的组成 …… 185
9.2 数字系统的设计 …… 186
9.2.1 数字系统的实现方法 …… 186
9.2.2 数字系统的设计过程 …… 187
9.2.3 数字系统的设计工具 …… 188
9.3 Verilog硬件描述语言简述 …… 188
9.3.1 Verilog语言的基本设计单元——模块 …… 189
9.3.2 结构化描述形式 …… 190
9.3.3 数据流描述方式 …… 190
9.3.4 行为描述方式 …… 192
9.3.5 混合设计描述方式 …… 193
9.3.6 设计模拟 …… 194
习题九 …… 195
附录 常用逻辑符号对照表 …… 196
参考文献 …… 197

第 1 章　数制和编码

本章以常用的十进制为例，引入进位计数制的概念，讨论计算机中的二进制计数方法，并推导出不同进位计数制之间的转换规则；介绍计算机中数值数据的表示方法，几种码制之间的关系以及十进制数的几种常用的编码。

1.1　进位计数制

所谓进位计数制，就是按照进位方式实现计数的一种规则。在日常生活中，人们使用了各种不同的进位计数制，除了最常用的十进制之外，还有六十进制（六十秒 = 一分钟），十二进制（十二个月 = 一年），而在计算机中，则采用二进制。

1.1.1　进位计数制的要素

1. 基数

我们知道，在十进制中共有十个数字符号：0、1、2、3、4、5、6、7、8、9。所有的十进制数都是由这些数字符号中的一个或几个组成的，我们把一种计数制中允许采用的所有基本数字符号的个数称为基数。那么，对于十进制来说，它的基数就是“10”；推广到一般的情况，则“**R**”进制数的基数为 R。十进制中每个数位计满 10 就向高位进 1，称为“逢十进一”，“R”进制则是“逢 R 进一”。

2. 权

同一个数字符号处在数中的不同位置时，代表的意义是不同的。比如，十进制数 11.1，虽然是由三个相同的数字符号“1”组成，但是这三个“1”所代表的数值不同。右边的代表“0.1”，中间的代表“1”，而最左边的代表“10”，这是由于它们的权不同。每个数字符号所表示的数值都等于该数字符号值乘以一个与数码所在位有关的常数，这个常数叫做“权”。权的大小是以基数为底，以数字符号所在位置的序号为指数的整数次幂。例如，十进制数 11.1 从左至右各位的权分别是：10^1，10^0，10^{-1}。11.1 可以表示为如下形式：

$$11.1 = 1 \times 10^1 + 1 \times 10^0 + 1 \times 10^{-1}$$

上式左边的表示方法称为位置记数法，等式右边称为多项式表示法或按权展开式。

一般来说，对于任意一个十进制数 N，可用位置记数法表示如下：

$$(N)_{10} = (k_{n-1}\, k_{n-2} \cdots k_1 k_0 . k_{-1} \cdots k_{-m})_{10}$$

或用按权展开式表示为

$$(N)_{10} = k_{n-1} \times 10^{n-1} + k_{n-2} \times 10^{n-2} + \cdots + k_1 \times 10^1 + k_0 \times 10^0 + k_{-1} \times 10^{-1} + \cdots + k_{-m} \times 10^{-m}$$

$$= \sum_{i=-m}^{n-1} k_i \times 10^i$$

这里，n 代表整数位，m 代表小数位，k_i 是十进制中 0 ~ 9 这 10 个数码中的任意一个，$0 \leqslant k_i \leqslant 9$，括号外下标 10 表示进位制的基数。

数字系统中使用的进位制并不限于十进制，对于任意的R进制数N，可用位置记数法表示为

$$(N)_R = (k_{n-1}\,k_{n-2}\cdots k_1 k_0 . k_{-1}\cdots k_{-m})_R$$

也可用按权展开式表示为

$$(N)_R = k_{n-1}\times R^{n-1} + k_{n-2}\times R^{n-2} + \cdots + k_1\times R^1 + k_0\times R^0 + k_{-1}\times R^{-1} + \cdots + k_{-m}\times R^{-m}$$

$$= \sum_{i=-m}^{n-1} k_i \times R^i$$

这里n代表整数位，m代表小数位，k_i是R进制中R个数码中的任意一个。

【例1-1】写出十进制数521.3的按权展开式。

解：$(521.3)_{10} = 5\times10^2 + 2\times10^1 + 1\times10^0 + 3\times10^{-1}$

1.1.2 二进制

在电子计算机中普遍使用基数为2的计数制，称为二进制。在二进制中，只有0和1两个数字符号，计数规则是由低位向高位“逢二进一”，即每位计满2就向高位进1。各位的权值是以2为底的连续整数幂，从右向左递增。

对于任意一个二进制数N，用位置计数法表示为

$$(N)_2 = (k_{n-1}\,k_{n-2}\cdots k_1 k_0 . k_{-1}\cdots k_{-m})_2$$

用按权展开式表示为

$$(N)_2 = k_{n-1}\times2^{n-1} + k_{n-2}\times2^{n-2} + \cdots + k_1\times2^1 + k_0\times2^0 + k_{-1}\times2^{-1} + \cdots + k_{-m}\times2^{-m}$$

$$= \sum_{i=-m}^{n-1} k_i \times 2^i \qquad (k_i \text{为0或1})$$

在计算机中采用二进制而不采用十进制，主要出于以下原因：

1）技术上容易实现。二进制数只有0和1两个数码，很多计算机所采用的元件都具有两个不同的稳定状态，这就可以用二进制的“0”和“1”来表示。例如，电平的“高”或“低”，脉冲信号的“有”和“无”等。

2）运算规则简单，使运算电路及控制简化。其运算规则如下：

加法规则

$0+0=0$　　$0+1=1$　　$1+0=1$　　$1+1=0$（且向相邻高位进1）

减法规则

$0-0=0$　　$0-1=1$（且向相邻高位借1）　　$1-0=1$　　$1-1=0$

乘法规则

$0\times0=0$　　$0\times1=0$　　$1\times0=0$　　$1\times1=1$

除法规则

$0\div1=0$　　$1\div1=1$

【例1-2】进行1011+1001运算。

解：

$$\begin{array}{r} 1\,0\,1\,1 \\ +\,1\,0\,0\,1 \\ \hline 1\,0\,1\,0\,0 \end{array}$$

（有进位）

【例1-3】进行1101－1011运算。

解：
$$\begin{array}{r} 1101 \\ -1011 \\ \hline 0010 \end{array}$$

二进制减法运算从低位起按位进行，在遇到0减1时，就要向相邻高位借1。

3）二进制中的“0”和“1”可与逻辑代数中逻辑变量的值“假”和“真”对应，能使计算机方便地进行逻辑运算。

1.1.3 八进制和十六进制

虽然计算机中采用二进制有诸多优点，比如运算规则简单，便于电路实现等。但是，用二进制表示一个数时，所用的位数比用十进制数表示的位数多很多，书写长，容易出错。因此，人们常采用八进制数和十六进制数来弥补二进制的缺点。这两种数制不但容易书写和阅读，而且，它们与二进制数之间的转换很容易。

在计算机中，常在数字后面加一个缩写的字母，用来标识其是几进制数，见表1-1。若数字后不加标识则默认是十进制数。

表1-1 常用进制的表示符号

	标 识	原 文	实 例
二进制	B	Binary	1011.1B
八进制	Q/O	Octal	275.6Q
十进制	D	Decimal	812.9D
十六进制	H	Hexadecimal	1A8.7H

1. 八进制

八进制数的基数是8，采用的数码是0～7，计数规则是从低位向高位“逢八进一”。对于相邻两位来说，高位的权值是低位权值的8倍。一个八进制数N的按权展开式如下：

$$(N)_8 = k_{n-1} \times 8^{n-1} + k_{n-2} \times 8^{n-2} + \cdots + k_1 \times 8^1 + k_0 \times 8^0 + k_{-1} \times 8^{-1} + \cdots + k_{-m} \times 8^{-m}$$

$$= \sum_{i=-m}^{n-1} k_i \times 8^i \qquad (k_i \text{ 为 } 0 \sim 7 \text{ 中的数})$$

八进制和二进制的对应关系见表1-2，可见，3位二进制数与1位八进制数存在一一对应关系。

表1-2 八进制和二进制的对应关系

八进制数	0	1	2	3	4	5	6	7
二进制数	000	001	010	011	100	101	110	111

2. 十六进制

十六进制数的基数为16，采用的数码是0，1，2，3，4，5，6，7，8，9，A，B，C，D，E，F。其中，A，B，C，D，E，F分别代表十进制数字10，11，12，13，14，15。计数规则是从低位向高位“逢十六进一”，对于相邻两位来说，高位的权值是低位权值的16倍。一个十六进制数N的按权展开式如下：

$$(N)_{16} = k_{n-1} \times 16^{n-1} + k_{n-2} \times 16^{n-2} + \cdots + k_1 \times 16^1 + k_0 \times 16^0 + k_{-1} \times 16^{-1} + \cdots + k_{-m} \times 16^{-m}$$

$$= \sum_{i=-m}^{n-1} k_i \times 16^i \qquad (k_i \text{ 为 } 0 \sim F \text{ 中的数})$$

十六进制、二进制和十进制之间的对应关系见表 1-3。

表 1-3　几种进制的对应关系

十进制数	二进制数	十六进制数
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

1.2　数制转换

一个数 N 从一种进制转换成另外一种进制称为数制转换。在计算机中，普遍使用二进制数，采用二进制数的数字系统只能处理二进制数或用二进制代码形式表示的其他进位制数，而人们习惯于使用十进制数，所以，在信息处理中首先必须把十进制数转换成计算机能加工处理的二进制数进行运算，然后还需要将二进制数的计算结果转换成人们习惯的十进制数。通常，这种进制之间的转换是由计算机按照规定的算法自动完成的。

1.2.1　二进制数与十进制数的转换

1. 二进制转换成十进制

二进制数转换成十进制数很简单，把二进制数按权展开，利用十进制运算法则，将式中各乘积项的积算出来，然后各项相加，即可得到与该二进制数相对应的十进制数。

【例 1-4】将二进制数 110010.01B 转换成十进制数。

解：$(110010.01)_2 = 1\times2^5 + 1\times2^4 + 0\times2^3 + 0\times2^2 + 1\times2^1 + 0\times2^0 + 0\times2^{-1} + 1\times2^{-2}$

$= 32 + 16 + 2 + 0.25$

$= (50.25)_{10}$

2. 十进制转换成二进制

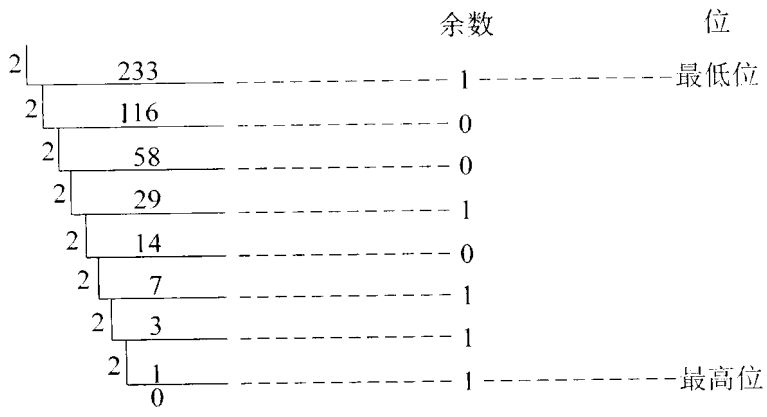
十进制数转换成二进制数时，整数和小数的转换方法不同。

（1）整数转换

整数转换通常采用“除2取余”法。从十进制数的按权展开式可知，把十进制数除以2，得到余数（1或0）即为相应二进制数的最低位 K_0，把得到的商再除以2，得到的余数即为二进制数的次低位 K_1。依次类推，继续上述过程，直至商为0，即可得到从低位到高位的余数序列，便构成对应的二进制数。

【例1-5】将十进制整数233转换为二进制整数，要把它写成如下形式：

解：



所以，$(233)_{10} = (11101001)_2$。

（2）纯小数转换

十进制数的小数部分通常采用“乘2取整”法进行转换，即先将十进制小数乘以2，取其整数部分1或0作为对应二进制小数的最高位 K_{-1}；然后，将乘积的小数部分再乘以2，并再取整数部分作为次高位 K_{-2}，重复上述过程即可得到相应的二进制数。读者可结合整数转换方法，自行分析其中的原理。

值得注意的是，并不是所有的十进制小数都能够用有限位的二进制小数表示，通常这类情况可以根据精度的要求，采用“0舍1入”的近似值表示。

【例1-6】将十进制小数0.625转换为二进制小数，需把它写成如下形式：

解：

十进制小数	积的整数部分	位
0.625 × 2 = 1.250	1	最高位
0.250 × 2 = 0.50	0	
0.50 × 2 = 1.0	1	最低位

所以，$(0.625)_{10} = (0.101)_2$。

一个既有整数又有小数部分的十进制数转换成二进制数，其转换步骤如下：①整数部分用除 2 取余法进行转换；②小数部分用乘 2 取整法进行转换；③将已转换好的两部分连接在一起就是所求的二进制数。

【例 1-7】将十进制数 233.625 转换为二进制数。

解：利用刚才的转换结果

$$(233)_{10} = (11101001)_2，(0.625)_{10} = (0.101)_2$$

所以，$(233.625)_{10} = (11101001.101)_2$。

【例 1-8】将十进制小数 0.913 转换为二进制小数，要求 4 位精度。

解：

十进制小数	积的整数部分	位
0.913 × 2		
1.826	1	最高位
0.826 × 2		
1.652	1	
0.652 × 2		
1.304	1	
0.304 × 2		
0.608	0	
0.608 × 2		
1.216	1	最低位

所以，$(0.913)_{10} = (0.1111)_2$

1.2.2 八进制数、十六进制数与二进制数的转换

八进制数的基数是 8（$8 = 2^3$），十六进制数的基数为 16（$16 = 2^4$）。二进制数、八进制数和十六进制数之间具有 2 的整指数倍关系，因而可直接进行转换。

1. 八进制转换为二进制

八进制数转换为二进制数，只要把每位八进制数用 3 位二进制数表示即可。

例如：$(745.21)_8$，我们把 7，4，5，2，1 各用 3 位二进制数表示：

7	4	5	.	2	1
111	100	101	.	010	001

所以，$(745.21)_8 = (111100101.010001)_2$

2. 二进制转换为八进制

将二进制数转换成八进制数，只需把二进制数以小数点为界，整数部分从右向左三位分为一组，最高位组不够三位时补0；小数部分从左向右三位分为一组，最低位组不够三位时也补0。然后，把每组二进制数用相对应的八进制数表示即可。

例如：把 $(11010.01)_2$ 转换成八进制的形式。

011　010　.　010

3　2　.　2

所以，$(11010.01)_2 = (32.2)_8$

3. 十六进制转换为二进制

十六进制数转换为二进制数，只要把每位十六进制数用4位二进制数表示即可。

例如：3C6.24H 转换成二进制的形式。

3　C　6　.　2　4

0011　1100　0110　.　0010　0100

所以，3C6.24H = 1111000110.001001B

4. 二进制转换为十六进制

将二进制数转换成十六进制数，则要把十六进制数以小数点为界，整数部分从右向左四位分为一组，最高位组不够四位时补0；小数部分从左向右四位分为一组，最低位组不够四位时也补0。然后，把每组二进制数用相对应的十六进制数表示即可。

例如：11100111101.01B 转换成十六进制数的形式。

0111　0011　1101　.　0100

7　3　D　.　4

所以，11100111101.01B = 73D.4H

从上述讨论可知，二进制在计算机中得到了普遍的应用，是计算机中数据的基本表示形式。八进制、十六进制比二进制表示形式简便，它们与二进制的转换很容易，因此，也是常用的数据表示形式。通常，十进制与八进制、十六进制转换时，可以以二进制为桥梁，先转换为二进制，再转换为八进制或十六进制。当然，也可以参考十进制转换为二进制的方法，通过“取余”和“取整”直接转换。

1.3 带符号数的代码表示

1.3.1 机器数和真值

前面讨论的数都没有考虑正负问题，一般认为是正数，但在实际的算术运算中总会出现负数，我们用符号“+”来表示正数，“-”来表示负数。在计算机中，怎样表示正数和负数呢？我们发现，数的符号是一个具有正、负两种值的离散信息，因此，可以用一位二进制数来表示符号，以0表示正号，以1表示负号。这样，一个带符号的数就由两部分组成：

部分表示数的符号，叫符号位；另一部分表示数的数值，叫数值位。对于一个 N 位二进制数，如果数的最高位为符号位，则剩下的 N－1 位就表示数的数值部分，而数在计算机中的这种二进制表示形式称为机器数。

例如：二进制数 +1101 在机器中的表示为

0	1	1	0	1

二进制数 －1101 在机器中的表示为

1	1	1	0	1

直接用正号“ + ”和负号“ － ”来表示带符号的二进制数，叫做符号数的真值。

例如：带符号机器数 01001 的真值为： +1001

带符号机器数 11001 的真值为： －1001

1.3.2 带符号数的表示方法

当符号数值化了之后，为了方便计算机的算术运算速度，人们提出了多种机器数的表示形式，常用的有原码、反码和补码。

1. 原码

原码表示法又称为“符号—数值表示法”。在以原码形式表示的正数和负数中，最高位表示符号位，对于正数，符号位记作 0，对于负数，符号位记作 1，其余各位表示数值部分。

（1）纯小数原码的表示

若二进制纯小数的真值为

$$X = \pm 0.X_1X_2\cdots X_{n-1}$$

则

当 X > 0 时，

$$[X]_{原} = 0.X_1X_2\cdots X_{n-1}$$

当 X < 0 时，

$$[X]_{原} = 1.X_1X_2\cdots X_{n-1}$$

例如，$X_1 = +0.1010$，$X_2 = -0.0111$，则

$$[X_1]_{原} = 0.1010 \qquad [X_2]_{原} = 1.0111$$

所以，纯小数原码一般定义如下：

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

（2）整数原码的定义

若二进制整数的真值为

$$X = \pm X_1X_2\cdots X_{n-1}$$

则

当 X > 0 时，

$$[X]_{原} = 0X_1X_2\cdots X_{n-1}$$

当 X < 0 时，