# 数字逻辑设计

## 学习指导与题解

曾 洁 李会勇◎编著

SHUZI LUOJI SHEJI
XUEXI ZHIDAO
YU TIJIE

电子科技大学出版社

# 数字逻辑设计
## 学习指导与题解

SHUZI LUOJI SHEJI
XUEXI ZHIDAO
YU TIJIE

曾洁 李会勇◎编著

电子科技大学出版社

**图书在版编目（CIP）数据**

数字逻辑设计学习指导与题解 / 曾洁，李会勇编著.
—成都：电子科技大学出版社，2012. 9

ISBN 978-7-5647-1041-5

Ⅰ. ①数… Ⅱ. ①曾… ②李… Ⅲ. ①数字逻辑－逻辑设计－高等学校－教学参考资料 Ⅳ. ①TP302.2

中国版本图书馆 CIP 数据核字（2011）第 222922 号

**数字逻辑设计学习指导与题解**

曾　洁　李会勇　编著

| | |
|---|---|
| **出　　版：** | 电子科技大学出版社（成都市一环路东一段 159 号电子信息产业大厦　邮编：610051） |
| **策划编辑：** | 杨仪玮 |
| **责任编辑：** | 万晓桐　谢应成 |
| **主　　页：** | www.uestcp.com.cn |
| **电子邮箱：** | uestcp@uestcp.com.cn |
| **发　　行：** | 新华书店经销 |
| **印　　刷：** | 成都市新都华兴印务有限公司 |
| **成品尺寸：** | 185mm×260mm　**印张** 16　**字数** 399 千字 |
| **版　　次：** | 2011 年 9 月第一版 |
| **印　　次：** | 2011 年 9 月第一次印刷 |
| **书　　号：** | ISBN 978-7-5647-1041-5 |
| **定　　价：** | 35.0 元 |

# 前　　言

《数字逻辑设计及应用》是教育部电子信息科学与电气信息类专业的一门重要基础骨干课程，是研究数字系统设计的入门课程。通过本课程的学习，学生能够掌握数字逻辑电路的基本理论和基本分析方法，为学习后续课程准备必要的电路知识。课程主要介绍数字逻辑电路的分析设计方法和基本的系统设计技巧，培养同学综合运用知识分析解决问题的能力和在工程性设计方面的基本素养。主要内容包括：数字逻辑基础、逻辑门电路、组合逻辑电路、常用组合逻辑功能器件、时序逻辑电路、常用时序逻辑功能器件、半导体存储器和可编程逻辑器件、数模和模数转换。

为了配合本课程的学习，特组织了在《数字逻辑设计及应用》课程教学方面有丰富经验的教师编写了这本《数字逻辑设计学习指导与题解》。本书内容编排主要按照常见的《数字逻辑设计及应用》教材内容体系进行组织，每一章主要由知识要点、例题解析、自测自评等部分组成，首先通过对各章相关的基本概念、运算规则、分析与设计方法、典型器件特点及应用等做出简明清晰的概括，便于学生很快地复习，然后通过典型例题的讲解，对每章涉及的重点知识点和难点进行示范性分析、讨论和解答，给学生以启发与思考，促进学生自学。最后通过习题练习对所学知识进行巩固与提高。本书最后列出了部分电子科技大学《数字逻辑设计及应用》课程的期末考试题以及全国研究生入学考试电子科技大学信号与信息处理（信息获取与探测）专业课程中的数字电路部分试题。

本参考书对电子信息科学与电气信息类专业学习数字电路相关课程的本科生或准备考研的学生，以及教师和工程技术人员均有一定的参考作用。

本书在编写出版过程中，得到了电子科技大学电子工程学院领导的大力支持，也得到《数字逻辑设计及应用》课程组张扬教授、唐广副教授以及其他成员的帮助，在此表示诚挚的感谢。

由于编者水平有限，书中难免有错误和不足之处，希望读者批评指正。

曾　洁　李会勇

# 目　　录

# 第一章 数制与码制

## 1.1 学 习 要 点

本章要求掌握：

1．按位计数制的概念。

2．十进制、二进制、八进制和十六进制数的表示方法以及它们之间的相互转换。

3．无符号非十进制数的加减运算方法。

4．有符号数的表达：符号—数值码（或称为原码），二进制补码（或直接称为补码）、二进制反码（或直接称为反码）表示以及它们之间的相互转换。

5．补码表示的带符号数的加减运算及溢出的判断。

6．BCD 码、格雷码（Gray code）的特点以及它们与二进制数之间的转换关系。

### 1.1.1 数制

1．*数值表示法*

数制是指多位数码中每一位的构成方法以及从低位到高位的进位规则。

对于一个具有 $n$ 位整数，$m$ 位小数的 $r$ 进制数 $N$，有 $(N)_r=\sum_{i=-m}^{n-1}k_i\times r^i$。式中，$k_i$ 是数码符号 0，1，…，$r$–1 中的任何一个；$r$ 称为基数。

常用的数制包括十进制、二进制、八进制和十六进制等。其中，

（1）十进制数基数为 10，可用数码符号为 0、1、2、3、4、5、6、7、8、9；其中第 $i$ 位的权为 $10^i$；

（2）二进制数基数为 2，可用数码符号为 0、1；第 $i$ 位的权为 $2^i$；

（3）八进制的基数为 8，可用数码符号为 0、1、2、3、4、5、6、7；第 $i$ 位的权为 $8^i$；

（4）十六进制的基数为 16，可用数码符号为 0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F；第 $i$ 位的权为 $16^i$。

*思考*:

（1）$r$ 进制数左移一位后等于多少？（答案：原来的数×$r$）

（2）$r$ 进制数右移一位后等于多少？（答案：原来的数÷$r$）。

（3）计算（11/32）$_{10}$ = ?$_2$，（11×4）$_{10}$ = ?$_2$ （提示：因为 32=$2^5$，（11/32）$_{10}$ 相当于将二进制数 1011 右移 5 位，结果为 $0.01011_2$。（11×4）$_{10}$ 相当于将二进制数 1011 左移 2 位，

结果为 $101100_2$）

2. 常用进制间的转换

（1）$r$ 进制数→十进制数

采用定义求和法：将 $r$ 进制数按公式 $D=\sum_{i=-n}^{p-1} d_i \times r^i$ 展开，再将所有各项的数值按十进制数相加即可。

例 1：完成下面的数制转换

（a）$10100.1101_2 = ?_{10}$　　（b）$67.24_8 = ?_{10}$　　（c）$12010_3 = ?_{10}$

解：（a）$10100.1101_2=1\times2^4+1\times2^2+1\times2^{-1}+1\times2^{-2}+1\times2^{-4}= 20.8125_{10}$

（b）$67.24_8=6\times8^1+7\times8^0+2\times8^{-1}+4\times8^{-2} = 55.3125_{10}$

（c）$12010_3=1\times3^4+2\times3^3+1\times3^1= 138_{10}$

（2）十进制数 →$r$ 进制数

采用基数乘除法，即：

整数部分：除以 $r$ 取余数，一直除到商为 0，然后按出现的先后顺序，由低到高排成转换后的整数；

小数部分：乘以 $r$ 取整数，一直到得到纯小数 0.0 或达到要求的精度，然后按出现的先后顺序，由高到低排成转换后的小数（注：$d$ 位 $r$ 进制小数的精度为 $r^{-d}$）。

例 2：完成下面的数制转换

（a）$125.375_{10} = ?_2$　　（b）$3489.35_{10}= ?_8$

**分析：十进制数转换为任意进制数，可以采用基数乘除法，将整数部分和小数部分分别转换后，再将转换结果相加。**

解：（a）整数部分：

125÷2=62 余数 1　（最低位）

62÷2=31 余数 0

31÷2=15 余数 1

15÷2=7 余数 1

7÷2=3 余数 1

3÷2=1 余数 1

1÷2=0 余数 1　（最高位）

所以，$125_{10}= 1111101_2$ ；

小数部分：

0.375×2=0.75　取整数 0　（最高位）

0.75×2=1.5　取整数 1

0.5×2=1.0　取整数 1　（最低位）

所以，$0.375_{10}= 0.011_2$ ；

所以，$125.375_{10}=1111101.011_2$ 。

（b）整数部分：

$3489÷8=436$ 余数 1 （最低位）
$436÷8=54$ 余数 4
$54÷8=6$ 余数 6
$6÷8=0$ 余数 6 （最高位）

所以，$3489_{10}=6641_8$

**转换位数的确定：一般来说，题目都会指明精确的程度，比如说精确到小数点后第几位。若没有指明，则只要转换后的 *r* 进制数的精度不低于要转换的十进制数的精度即可。本题中十进制数的精度小于 $10^{-2}$，所以转换后取八进制数小数点后三位，转换精度为小于 $8^{-3}$。**

小数部分：

$0.35×8=2.8$ 取整数 2 （最高位）
$0.8×8=6.4$ 取整数 6
$0.4×8=3.2$ 取整数 3 （最低位）

所以，$0.35_{10}= 0.263_8$

所以，$3489.35_{10}= 6641.263_8$

（3）二进制数↔八进制数，二进制数↔十六进制数

采用位数替换法。具体方法为：

1．保持小数点不变；每位八进制数对应 3 位二进制数；每位十六进制数对应 4 位二进制数；

2．由二进制转换时，从小数点开始向左向右分组，位数不足的须在整数部分前和小数部分后用 0 补足。

例 3：完成下面的数制转换

（a）$10100.1101_2 = ?_{16}$ （b）$101111.0111_2 = ?_8$
（c）$F3A5_{16} = ?_2$ （d）$67.24_8= ?_2$
（e）$5436.15_8 = ?_{16}$ （f）$ABCD_{16} = ?_8$

**分析：二、八、十六进制数间的相互转换，可采用“位数置换法”直接转换。十六进制和八进制之间的相互转换可以将其先变成二进制数后再进行转换。**

解：（a）$10100.1101_2 = \underline{0001}\ \underline{0100}.\underline{1101}_2 = 14.D_{16}$
（b）$101111.0111_2 =\underline{101}\ \underline{111}.\underline{011}\ \underline{100}_2 =57.34_8$
（c）$F3A5_{16} =1111\ 0011\ 1010\ 0101_2$
（d）$67.24_8=110\ 111.010\ 100_2$
（e）$5436.15_8=101\ 100\ 011\ 110.001\ 101_2=1011\ 0001\ 1110.0011\ 01_2=B1E.34_{16}$
（f）$ABCD_{16}=1010\ 1011\ 1100\ 1101_2=1\ 010\ 101\ 111\ 001\ 101_2= 125715_8$

### 1.1.2 *N* 进制无符号数的运算

#### 1. 二进制数的加法和减法

基本规则：$0+0=0$ $1+0=1$ $1+1=10$ $10-1=1$。

多位二进制数相加减时，可以列出竖式进行运算。运算要点和十进制数的类似，即小数点对齐；从低位向高位逐位进行运算。

进位和借位规则为：逢 2 进 1，借 1 当 2。

2. *N* 进制数的加法和减法

可以采用竖式进行运算；运算时需注意进位和借位规则为：逢 *N* 进 1，借 1 当 *N*。

思考：根据二进制数的运算法则，列出一位全加器、一位全减器的真值表。

提示：（1）一位全加器完成被加数 *X*、加数 *Y* 和进位输入 *CIN* 的加法运算 *X*+*Y*+*CIN*，输出为和 *S* 和进位输出 *COUT*，故真值表中应包括 3 个输入和 2 个输出。

（2）一位全减器完成被减数 *X*、减数 *Y* 和借位输入 *BIN* 的减法运算 *X*–*Y*–*BIN*，输出为差 *D* 和借位输出 *BOUT*。

例 4：完成下面的二进制数运算，指出所有的进位或借位：

（a）110101+11001　　　（b）11011101–1100011

解：列竖式计算结果（其中 C 为计算时产生的进位，B 为计算时产生的借位。）

$$\text{(a)}\quad \begin{array}{r} \text{C: } 1\,1\,0\,0\,0\,1\,0 \\ 1\,1\,0\,1\,0\,1 \\ +\quad 1\,1\,0\,0\,1 \\ \hline 1\,0\,0\,1\,1\,1\,0 \end{array} \qquad\qquad \text{(b)}\quad \begin{array}{r} \text{B: } 1\,1\,0\,0\,0\,1\,0\,0 \\ 1\,1\,0\,1\,1\,1\,0\,1 \\ -\quad 1\,1\,0\,0\,0\,1\,1 \\ \hline 0\,1\,1\,1\,1\,0\,1\,0 \end{array}$$

可得：（a）110101+11001=1001110　　　（b）11011101 − 1100011=1111010

### 1.1.3　带符号数的表示

1. 原码

符号数的原码表示法，又称为符号—数值（S-M）表示法。规定原码的最高位用来表示数的符号，其后各位用来表示数的绝对值。对正数，符号位用 0 表示；对负数，符号位用 1 表示。对于 0，有两种表示（+0、–0），所以 *n* 位二进制原码的表示范围为$-(2^{n-1}-1)\sim+(2^{n-1}-1)$。

2. 补码

规定正数的补码表示和其原码表示相同，负数的补码表示是其对应正数的补码表示逐位求反后再加 1。这样规定的目的是保证两个相加为 0 的符号数，其补码表示之和也为 0。所以，零的补码表示只有一种，*n* 位二进制补码的表示范围为$-2^{n-1}\sim+(2^{n-1}-1)$。

*N* 位二进制补码表示的展开式为 $(N)_{补}=-k_{n-1}\times 2^{n-1}+\sum_{i=0}^{n-2}k_i\times 2^i$ 。和 *N* 位二进制无符号数的展开式相比，只是最高位的权值由无符号数时的 $2^{n-1}$，变为$-2^{n-1}$。

思考：$11010_{补}=?_{10}$（答案：–16+8+2= − 6）

3. 反码

规定正数的反码表示和其原码表示相同，负数的反码表示是其对应正数的反码表示逐位求反。反码表示可以说是求补码表示过程中的一个中间值。零的反码表示有两种（全

0 和全 1），所以 $n$ 位二进制反码的表示范围为–（$2^{n-1}$–1）～+（$2^{n-1}$–1）。

综上所述，对有符号数的二进制原码、反码、补码表示，有：

（1）正数的原码、反码、补码表示相同，符号位都为 0。

（2）对于负数，不同码制表示下的码不同，但符号位都为 1。

（3）由正数的原码表示得到对应负数的原码表示只需将其符号位改为 1。

（4）由正数 $D$ 求负数–$D$ 的反码表示：

符号位不变，其余在该负数对应的原码基础上按位取反；或在|$D$|的原码基础上按位取反（包括符号位）。

（5）由正数 $D$ 求负数–$D$ 的补码表示：该负数对应的反码+1。

（6）对于零，原码和反码各有两种表达形式，但补码只有一种。

例 5：某十进制数的等值二进制原码、补码、反码表示（不一定是这个顺序）分别是 1010101、1101010、1010110，求该十进制数是多少？

解：该数的三种表示中符号位均为 1，所以该数为负数。1010101+1=1010110。根据原码、补码、反码表示的特点可知，1010101 为反码表示，1010110 为补码表示，1101010 为原码表示。无符号二进制数 101010 对应的十进制数为 $42_{10}$，所以该十进制数为$-42_{10}$。

### 4. 各种表示方法之间的转换

（1）由无符号数到符号数：

首先添加符号位：在最高有效位（MSB）前添加一位；

因为无符号数是正数，改为符号数时，添加的符号位应为 0。

（2）符号数改变符号：

改变符号意味着符号数发生了变化，相当于在原来的符号数前面加一个负号（–）；

按三种表达方式（码制）的不同，改变符号可以分为：

1）原码表示（S-M 码）改变最高位（符号位）；

2）反码表示改变每一位（取反）；

3）补码表示改变每一位，然后在最低位加 1（取补）。

注意：（1）取补操作需要忽略最高位的进位（保持位数不变）；

（2）为便于区分，任何符号数必须用下标注明采用的码制。

（3）三种表达方式之间的转换：

对于正数，不同表达方式结果相同，直接改下标即可。

对于负数，先按转换前的表达方式将其改为对应的正数，修改下标后，再按转换后的表达方式将其改为负数。

（4）符号数增加位数（位数扩展）的方式：

原码表示：在符号位之后加 0。

补码与反码表示：在符号位之前增加与符号位相同的位。

例 6：已知 $A_2$=1011，$B_{原}$=1011，$C_{补}$=1011，$D_{反}$=0101；写出 A、B、C、D 和–A、–B、–C、–D 各种码制的 8 位符号数。

解：从已知条件中可知，A 为无符号数，B、C、D 已经是符号数；A、D 为正数，B、 C 为负数。

$A_{原}=A_{反}=A_{补}=0000\ 1011$　$-A_{原}=1000\ 1011$　$-A_{反}=1111\ 0100$　$-A_{补}=1111\ 0101$

$B_{原}=1000\ 0011$　$-B_{原}=-B_{反}=-B_{补}=0000\ 0011$　$B_{反}=1111\ 1100$　$B_{补}=1111\ 1101$

$C_{补}=11111011$　$-C_{原}=-C_{反}=-C_{补}=0000\ 0101$　$C_{原}=1000\ 0101$　$C_{反}=1111\ 1010$

$D_{原}=D_{反}=D_{补}=0000\ 0101$　$-D_{原}=1000\ 0101$　$-D_{反}=1111\ 1010$　$-D_{补}=1111\ 1011$

### 1.1.4 二进制补码的加法和减法

补码运算的优点为：（1）数值（包括符号位）单调变化；（2）0 具有唯一的表达形式；（3）运算时只考虑加法，减法可采用代数和的方式进行运算。

补码运算过程中会产生溢出。溢出是指运算结果超出表示的位数而导致结果错误。异号数相加绝不会溢出；同号数相加可能会溢出。

溢出的判断方法：同号数相加发生符号位变化（或者，向符号位的进位与符号位产生的进位不同）。

思考：根据溢出的定义，推出无符号数运算的溢出判断规则。（答案：两无符号数相加，最高位有进位表示溢出；两无符号数相减，最高位有借位表示溢出。）

例 7：已知 $A_{补}=001011$，$B_{补}=111011$，$C_{补}=101011$，$D_{补}=011001$；求：$(A-B)_{补}$、$(C+D)_{补}$、$(A+D)_{补}$。

**分析：（1）补码表示的数在进行减法运算时，可以将减法运算变为加法运算，方法是将减数逐位求反再在最低位加 1（即将减数取补）。**

**（2）运算后需要判断结果是否有效，即判断是否溢出。**

解：$(A-B)_{补}=A_{补}-B_{补}=A_{补}+(-B_{补})=001011+000101=010000$（无溢出）

$(C+D)_{补}=C_{补}+D_{补}=101011+011001=1\ 000100$（无溢出，运算结果为 000100）

$(A+D)_{补}=A_{补}+D_{补}=001011+011001=100100$（溢出，结果无效）

### 1.1.5 编码

1. 二进制编码

$m$ 位二进制串可以表达最多 $2^m$ 种不同的对象；表达 $n$ 种不同对象至少需要 $b=\lceil \log_2 n \rceil$ 位二进制数据串。（$\lceil \log_2 n \rceil$ 为上限函数，表示取大于或等于括号内数值的最小整数，即 $b$ 为满足 $2^b \geqslant n$ 关系的最小整数。）

2. 编码与数制的区别

在数制表达中，二进制串表达具体数量，可以比较大小，小数点前的 MSB 和小数点后的 LSB 的 0 通常可以去掉（有符号数除外）；在码制表达中，二进制串表达的是对象的名称，不能比较大小，MSB 和 LSB 的 0 也不能去掉。

3. BCD 码

BCD 码称为十进制数的二进制编码。一般是利用 4 位二进制串表示 0～9 十个数字，有很多种不同的编码方法。常用的几种编码有 8421BCD 码、2421BCD 码、余 3 码，它们与十进制数字之间的对应关系如表 1-1。其中的 8421BCD 码、2421BCD 码的各位权值

是固定的，称为有权码；余 3 码为无权码。

表 1-1 常用的 BCD 码

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

例 8：完成下面的变换：

（1）$47.86_{10}=?_{8421BCD}=?_{余3码}$　　（2）$001101010111_{8421BCD}=?_{2421BCD}=?_{10}$

（3）$110001011_2=?_{8421BCD}$

**分析：十进制数到 BCD 码的相互转换，按照 1 位十进制数对应 4 位码的原则完成，小数点照写。二进制数和 BCD 码的相互转换，需要先变为十进制数后再进行变换。**

解：（1）$47.86_{10}$=0100 0111.1000 0110 $_{8421BCD}$ =0111 1010.1011 1001 $_{余3码}$

（2）$001101010111_{8421BCD}$=0011 1011 $1101_{2421BCD}$ =$357_{10}$

（3）$110001011_2$=256+128+8+2+1=$395_{10}$=0011 1001 0101 $_{8421BCD}$

4. 格雷（GRAY）码

格雷码的特点是连续数值变化时码字（相邻码字）之间只有 1 位不同。

由 $n$ 位二进制数（自然码）直接得到 $n$ 位 Gray 码的方法为：对 $n$ 位二进制码从右到左编号 0～$n$−1；若二进制码第 $i$ 位和 $i$+1 位相同，则 Gray 码第 $i$ 位为 0，否则为 1；二进制码第 $n$+1 位当做 0 处理。

思考：总结由 $n$ 位 Gray 码直接得到 $n$ 位二进制数（自然码）的方法。（提示：使用异或运算定理）

例 9：9 的四位 Gray 码是（　　　）。325 的十位 Gray 码为（　　　　）。

解：9 的四位二进制编码为 1001，按照上面所说的构成方法，四位 Gray 码为 1101。

325 的十位二进制编码为 0101000101，所以十位 Gray 码为 0111100111。

5. 奇偶校验码（可靠性编码）

奇/偶校验是数据传送时采用的一种检验错误的方式，可分为奇校验和偶校验两种。

在传送每一个字节的时候需要另外附加一位作为校验位，如果是采用奇校验，则需保证传输数据中“1”的个数为奇数的要求，也就是说，当实际数据中“1”的个数为偶数的时候，这个校验位就是“1”，否则这个校验位就是“0”，这样，在接收方收到数据时，将按照奇校验的要求检测数据中“1”的个数，如果是奇数，表示传送正确，否则表示传送错误。比如，0100101 的奇校验码就是 0 0100101，其中最高位“0”为校验位。

同理，对于偶校验，当实际数据中“1”的个数为偶数的时候，这个校验位就是“0”，否则这个校验位就是“1”，这样就可以保证传送数据满足偶校验的要求。在接收方收到

数据时，将按照偶校验的要求检测数据中“1”的个数，如果是偶数个“1”，表示传送正确，否则表示传送错误。

思考：已知 8 位数据为 11010100，校验位为 $C$，则奇校验时 $C$=？，偶校验时 $C$=？（答案：奇校验时 $C$=1，偶校验时 $C$=0）

## 1.2 例题解析

例 10：下面每个算术运算至少在某一种计数制中是正确的。试确定每个运算中操作数的基数可能是多少？

（a）1234+5432=6666　　（b）41/3=13

（c）33/3=11　　（d）23+44+14+32=223

（e）302/20=12.1　　（f）$\sqrt{41}=5$

**分析：数的大小不会随着所用进制的变化而变化。对任意基数 $r$ 表示的数，其对应的十进制数的大小可以由公式 $D=\sum_{i=-n}^{p-1} d_i \times r^i$ 得到。设操作数基数为 $r$，将等号两边变为十进制数的运算，等式依然成立。若基数为 $r$，则各位数码可以是 0～$r$–1 中的正整数，也就是说，各位上出现的数码必定小于 $r$。**

解：（a）等式左边=$1\times r^3+2\times r^2+3\times r^1+4\times r^0+5\times r^3+4\times r^2+3\times r^1+2\times r^0=6\times r^3+6\times r^2+6\times r^1+6\times r^0$，

等式右边= $6\times r^3+6\times r^2+6\times r^1+6\times r^0$=等式左边，

又因为出现的数码最大值为 6，所以，基数为大于 6 的整数。

（b）由 $\frac{4\times r+1}{3}=1\times r+3$，可得 $4\times r+1=3\times r+9$，所以基数 $r$ 为 8。

（c）由 $\frac{3\times r+3}{3}=1\times r+1$，可得 $3\times r+3=3\times r+3$，等式始终成立；又因为式中出现的数码最大值为 3，所以基数 $r$ 为大于 3 的整数。

（d）由等式 $2\times r+3+4\times r+4+1\times r+4+3\times r+2=2\times r^2+2\times r^1+3\times r^0$，可得 $r$=5 或 $r$=–1，基数为 5。

（e）由 $\frac{3\times r^2+2}{2\times r}=1\times r+2+r^{-1}$，可得基数 $r$ 为 4。

（f）由 $\sqrt{4\times r+1}=5$，可得基数 $r$ 为 6。

例 11：已知在某数制系统中，$5x^2-50x+125=0$ 的解为 $x=5$ 和 $x=8$。请问此数制系统是多少进制？

解：$5x^2-50x+125=0$ 为非十进制下的等式，将它变为十进制（$r$ 为进制基数），即为 $5x^2-5\times r\times x+1\times r^2+2\times r+5=0$，该式对 $x$=5 和 $x$=8 均成立。

对 $x$=5 等式成立，可得：$r$=13 或 $r$=10；

对 $x$=8 等式成立，可得：$r$=13 或 $r$=25；

所以：$r$=13。此数制系统是 13 进制。

例 12：完成下面十六进制数的运算。

（a）F35B+27E6　　　　（b）1B90F−C44E

**分析：多位 *r* 进制数相加减时，可以列出竖式进行运算。运算方法类似十进制数的运算，将小数点对齐，从低位向高位逐位运算。加法时逢 *r* 进 1，减法时借 1 当 *r*。**

解：

（a）
$$\begin{array}{r} \text{F}\ 3\ 5\ \text{B} \\ +\quad 2\ 7\ \text{E}\ 6 \\ \hline 1\ 1\ \text{B}\ 4\ 1 \end{array}$$

（b）
$$\begin{array}{r} 1\ \text{B}\ 9\ 0\ \text{F} \\ -\quad \text{C}\ 4\ 4\ \text{E} \\ \hline \text{F}\ 4\ \text{C}\ 1 \end{array}$$

例 13：试证明：通过符号位扩展，可以用更多的数位来表示二进制补码数。也就是说，对给定的 $n$ 位二进制补码数，当 $m>n$ 时，$x$ 的 $m$ 位二进制补码表示就等于在 $x$ 的 $n$ 位补码左位添加（$m-n$）个符号位，即扩展的各位均以符号位填充。

**分析：补码表示的数 $D=d_{n-1}d_{n-2}\dots d_1d_0$，其对应的十进制数值大小为 $D=-2^{n-1}\cdot d_{n-1}+\sum\limits_{i=0}^{n-2}d_i\cdot 2^i$。因此，只需证明在位扩展前后补码表示所对应的十进制数的数值大小相等即可。**

证明：已知 $X=x_{n-1}x_{n-2}\dots x_1x_0$，$Y=y_{m-1}y_{m-2}\dots y_1y_0$，$m=n+d$，并且 $y_{m-1}=y_{m-2}=\dots=y_{n-1}=x_{n-1}$，$y_{n-1}y_{n-2}\dots y_1y_0=x_{n-1}x_{n-2}\dots x_1x_0$，因为 $X$、$Y$ 均为补码表示，所以有：

$$X=-2^{n-1}\cdot x_{n-1}+\sum_{i=0}^{n-2}x_i\cdot 2^i,$$

$$\begin{aligned} Y&=-2^{m-1}\cdot y_{m-1}+\sum_{i=0}^{m-2}y_i\cdot 2^i=-2^{m-1}\cdot y_{m-1}+2^{m-2}\cdot y_{m-2}+\dots+2^{n-1}\cdot y_{n-1}+\sum_{i=0}^{n-2}y_i\cdot 2^i \\ &=-2^{m-1}\cdot x_{n-1}+2^{m-2}\cdot x_{n-1}+\dots+2^{n-1}\cdot x_{n-1}+\sum_{i=0}^{n-2}x_i\cdot 2^i \\ &=(-2^{m-1}+2^{m-2}+\dots+2^{n-1})\cdot x_{n-1}+\sum_{i=0}^{n-2}x_i\cdot 2^i=-2^{n-1}\cdot x_{n-1}+\sum_{i=0}^{n-2}x_i\cdot 2^i=X \end{aligned}$$

原题得证。

例 14：写出下面每个十进制数的 8 位原码（S-M 码）、二进制补码、二进制反码表示：+18、−49、−100、−128、0。

**分析：对正数来说，规定其原码表示、二进制补码表示、二进制反码表示相同，符号位为 0。对负数，规定其原码表示为对应正数的原码表示的符号位取反，其二进制补码表示为对应正数的补码表示取补（逐位取反后加 1），其二进制反码为对应正数的反码表示取反（逐位取反）。**

**8 位二进制原码和反码表示数的范围为−127～+127，所以−128 超出其表示范围，无法表示。虽然−128 在 8 位二进制补码的表示范围在−128～+127 内，但不能由对应正数直接得到。补码表示将 *n* 位二进制数按最高位（符号位）为 0 和为 1 分为数量相等的两部分，因为数字 0 的表示符号位为 0，所以负数比正数多一个，这一个就是 $-2^{n-1}$，对应的**

**码为 1 后面有 *n*–1 个 0。*n*=8 时即为–128，对应码为 10000000。**

解：结果如表题 1-1 所示。

表题 1-1　例 14 的解

| 十进制数 | +18 | –49 | –100 | –128 | 0 |
|---|---|---|---|---|---|
| 原码表示 | 00010010 | 10110001 | 11100100 | 无法表示 | 00000000 或 10000000 |
| 二进制补码表示 | 00010010 | 11001111 | 10011100 | 10000000 | 00000000 |
| 二进制反码表示 | 00010010 | 11001110 | 10011011 | 无法表示 | 00000000 或 11111111 |

例 15：采用 6 位二进制补码完成下列运算：

（1）3–2　　　（2）–15–24

解：将 3、2、–15、24 分别用 6 位补码表示，然后将减法运算变为加法。

（1）$(3)_{补}-(2)_{补}=(3)_{补}+(-2)_{补}$

3–2 ：　0 00011–0 00010 = 0 00011+ 1 11110 = 0 00001　　（无溢出）

（2）–15–24 ：1 10001–0 11000 = 1 10001+1 01000 = 0 11001　（有溢出）

例 16：已知 $A=+(1011)_2$，$B=-(1101)_2$，求 $(A+B)_{补}$，$(A\times B)_{补}$

**分析：求两数和的补码可以先求两数的补码，再求和。而求两数乘积的补码，可以先求两数乘积的原码，再转换为补码。**

解：$A_{补}$=01011，$B_{补}$=10011，

所以，$(A+B)_{补}$=01011+10011=11110。

因为 1011×1101=10001111，A 为正数，B 为负数，所以 A×B 应为负数。

$(A\times B)_{原码}$=110001111，所以 $(A\times B)_{补码}$=101110001。

例 17：指出下面 8 位二进制补码相加时是否发生溢出。

（a）10111001+11010110　　（b）00100110+01011010

**分析：补码数的加法运算法则与二进制数加法运算法则相同，可用竖式完成。最高位的进位可忽略。**

解：

（a）
$$\begin{array}{r} 1\ 0\ 1\ 1\ 1\ 0\ 0\ 1 \\ +\ 1\ 1\ 0\ 1\ 0\ 1\ 1\ 0 \\ \hline 1\ 0\ 0\ 0\ 1\ 1\ 1\ 1 \end{array}$$

加数符号相同，和的符号与加数符号相同，无溢出。

（b）
$$\begin{array}{r} 0\ 0\ 1\ 0\ 0\ 1\ 1\ 0 \\ +\ 0\ 1\ 0\ 1\ 1\ 0\ 1\ 0 \\ \hline 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0 \end{array}$$

加数符号相同，和的符号与加数符号不同，有溢出。

例 18：试用 8421BCD 码完成下列十进制数的加法运算：

（1）5+9　　　（2）58+36

**分析：8421BCD 码的加法运算法则类似于 4 位无符号二进制数加法，但如果结果超过 1001（9）则需要修正，方法是将结果再加 0110（6）。**