高等教育教材

数字系统逻辑设计

欧阳星明　主编

欧阳星明　唐九飞　孙百勇　编著

韦　敏　主审

電子工業出版社

Publishing House of Electronics Industry

北京 · BEIJING

图书在版编目（CIP）数据

数字系统逻辑设计/ 欧阳星明主编. —北京：电子工业出版社，2004.3
高等教育教材
ISBN 7-5053-9442-8

Ⅰ. 数… Ⅱ. 欧… Ⅲ. 数字系统-逻辑设计-高等学校-教材 Ⅳ. TP302.2

中国版本图书馆 CIP 数据核字（2003）第 112506 号

策划编辑：刘宪兰
责任编辑：刘宪兰 特约编辑：逄积仁
印　　刷：
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
经　　销：各地新华书店
开　　本：787×980 1/16 印张：20.75 字数：429 千字
印　　次：2004 年 3 月第 1 次印刷
印　　数：5 000 册 定价：26.00 元

凡购买电子工业出版社的图书，如有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系。联系电话：（010）68279077。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。

前　言

QIANYAN

21 世纪是信息的时代，计算机技术是信息技术的核心，各行各业对计算机应用人才的需求越来越多。为了满足社会发展的需要，高等学校计算机教育正处在不断发展与变革的过程中。数字系统逻辑设计作为计算机科学与技术（类）各专业必修的一门重要专业基础课，无论从教学内容、教学方法与手段等方面都在不断更新与变化。本书是参照目前高等院校数字系统逻辑设计课程教学的基本要求，结合作者多年从事理论与实践教学的经验编写的。

本书对数字系统逻辑电路分析与设计的基本知识、基本理论和基本方法进行了系统介绍。针对各种不同规模的逻辑器件，举例说明逻辑电路分析与设计的全过程。目的在于培养学生对各种逻辑电路进行分析与设计的能力，为数字计算机和其他数字系统的硬件分析与设计打下坚实的基础。

全书分为七章：第 1 章介绍数字系统中采用的数制与编码；第 2 章介绍逻辑设计的理论基础——逻辑代数，以及实现各种逻辑运算的基本器件——逻辑门；第 3 章介绍组合逻辑电路分析与设计的基本方法及常用中规模组合逻辑器件；第 4 章介绍同步时序逻辑电路分析与设计的基本方法及常用中规模同步时序逻辑器件及应用；第 5 章介绍异步时序逻辑电路分析与设计的基本方法及常用中规模异步时序逻辑器件及应用；第 6 章介绍可编程逻辑器件 PLD（包括 PROM、EPROM、EEPROM、PLA、PAL、GAL 等）及 ISP 技术；第 7 章为综合应用举例，结合实际应用介绍简单数字系统的设计。

本书系统性强，概念清晰，例题丰富。内容力求深入浅出、通俗易懂，便于自学。为了满足学习的需要，每章后均配有小结、思考题和习题，书末附有习题解答和实验内容。

本书由欧阳星明主编，全书内容由欧阳星明、唐九飞、孙百勇共同完成。华中科技大学出版社社长韦敏先生担任本书主审，韦敏先生对书稿进行了认真、详细的审阅，提出了许多极为宝贵的意见，在此表示诚挚的谢意。在本书编写过程中得到了华中科技大学计算机学院、成人教育学院，以及电子工业出版社的领导和有关同志的关心、支持和帮助，在此一并表示衷心的感谢。

由于编者水平有限，加之时间仓促，书中错漏和不妥之处在所难免，恳请读者赐正。

编著者

2003 年 8 月于华工园

目　录

MULU

第 1 章　基本知识 …… (1)
1.1　数制及其转换 …… (2)
1.1.1　进位计数制 …… (2)
1.1.2　数制转换 …… (5)
1.2　带符号二进制数的代码表示 …… (8)
1.2.1　原码 …… (9)
1.2.2　反码 …… (9)
1.2.3　补码 …… (11)
1.3　几种常用的编码 …… (12)
1.3.1　十进制数的二进制编码（BCD 码） …… (12)
1.3.2　可靠性编码 …… (14)
1.3.3　字符编码 …… (17)
本章小结 …… (18)
思考题 …… (19)
习题 1 …… (19)
第 2 章　逻辑代数基础 …… (21)
2.1　逻辑代数的基本概念 …… (22)
2.1.1　逻辑变量及基本逻辑运算 …… (22)
2.1.2　逻辑函数及逻辑函数间的相等 …… (25)
2.1.3　逻辑函数的表示法 …… (26)
2.2　逻辑代数的基本等式和重要规则 …… (27)
2.2.1　基本等式 …… (27)
2.2.2　重要规则 …… (28)
2.2.3　复合逻辑 …… (29)
2.3　逻辑函数表达式的形式与变换 …… (31)
2.3.1　逻辑函数表达式的两种基本形式 …… (32)
2.3.2　逻辑函数表达式的两种标准形式 …… (32)
2.3.3　逻辑函数表达式的转换 …… (36)
2.4　逻辑函数化简 …… (38)
2.4.1　代数化简法 …… (38)

2.4.2 卡诺图化简法 ……（40）
2.5 逻辑门电路 ……（47）
2.5.1 简单门电路 ……（47）
2.5.2 复合门电路 ……（49）
2.5.3 逻辑门电路的性能指标 ……（53）
2.6 逻辑函数的实现 ……（56）
2.6.1 用与非门实现逻辑函数 ……（56）
2.6.2 用或非门实现逻辑函数 ……（57）
2.6.3 用与或非门实现逻辑函数 ……（58）
2.6.4 用异或门实现逻辑函数 ……（59）
本章小结 ……（60）
思考题 ……（60）
习题 2 ……（60）
第 3 章 组合逻辑电路 ……（63）
3.1 基本概念 ……（64）
3.2 组合逻辑电路分析 ……（64）
3.2.1 分析的一般步骤 ……（64）
3.2.2 分析举例 ……（65）
3.3 组合逻辑电路设计 ……（68）
3.3.1 设计方法概述 ……（69）
3.3.2 设计举例 ……（69）
3.3.3 设计中几个实际问题的处理 ……（72）
3.4 组合逻辑电路的险象 ……（79）
3.4.1 竞争现象与险象的产生 ……（80）
3.4.2 险象的消除 ……（81）
3.5 常用中规模组合逻辑器件及应用 ……（83）
3.5.1 二进制并行加法器 ……（83）
3.5.2 译码器 ……（87）
3.5.3 数据选择器 ……（91）
本章小结 ……（97）
思考题 ……（97）
习题 3 ……（98）
第 4 章 同步时序逻辑电路 ……（101）
4.1 概述 ……（102）
4.1.1 同步时序逻辑电路的结构和特点 ……（102）

4.1.2 同步时序逻辑电路的两种模型 …… (103)
4.1.3 同步时序逻辑电路的描述方法 …… (104)
4.2 存储元件——触发器 …… (106)
4.2.1 基本 R-S 触发器 …… (106)
4.2.2 常用时钟控制触发器 …… (110)
4.3 同步时序逻辑电路分析 …… (115)
4.3.1 分析的一般步骤 …… (115)
4.3.2 分析举例 …… (116)
4.4 同步时序逻辑电路的设计 …… (122)
4.4.1 设计的一般步骤 …… (123)
4.4.2 设计举例 …… (133)
4.5 常用中规模同步时序逻辑器件及应用 …… (140)
4.5.1 集成同步计数器 …… (140)
4.5.2 集成同步寄存器 …… (143)
本章小结 …… (147)
思考题 …… (148)
习题 4 …… (148)
第 5 章 异步时序逻辑电路 …… (153)
5.1 异步时序逻辑电路的结构模型 …… (154)
5.2 脉冲异步时序逻辑电路 …… (154)
5.2.1 电路工作特点与描述方法 …… (154)
5.2.2 脉冲异步时序逻辑电路的分析 …… (155)
5.2.3 脉冲异步时序逻辑电路的设计 …… (159)
5.3 电平异步时序逻辑电路 …… (161)
5.3.1 电路工作特点与描述方法 …… (161)
5.3.2 电平异步时序逻辑电路的分析 …… (165)
5.3.3 电平异步时序逻辑电路反馈回路间的竞争 …… (166)
5.4 电平异步时序逻辑电路的设计 …… (169)
5.4.1 电平异步时序逻辑电路设计的一般步骤和方法 …… (169)
5.4.2 电平异步时序逻辑电路设计举例 …… (171)
5.5 常用中规模异步时序逻辑器件及应用 …… (176)
5.5.1 典型芯片 …… (176)
5.5.2 应用举例 …… (178)
本章小结 …… (179)
思考题 …… (179)

习题 5 ………… (180)
第 6 章　可编程逻辑器件 ………… (185)
6.1　PLD 概述 ………… (186)
6.1.1　PLD 的发展 ………… (186)
6.1.2　PLD 的基本结构 ………… (186)
6.1.3　PLD 的电路表示法 ………… (187)
6.1.4　PLD 的分类 ………… (188)
6.2　可编程只读存储器 ………… (189)
6.2.1　半导体存储器 ………… (189)
6.2.2　可编程 ROM 的结构与类型 ………… (190)
6.2.3　可编程 ROM 的应用 ………… (195)
6.3　可编程逻辑阵列 PLA ………… (198)
6.3.1　PLA 的逻辑结构 ………… (198)
6.3.2　PLA 的应用 ………… (199)
6.4　可编程阵列逻辑 PAL ………… (203)
6.4.1　PAL 的基本逻辑结构 ………… (203)
6.4.2　PAL 的输出和反馈结构 ………… (205)
6.4.3　PAL 的应用 ………… (208)
6.5　通用阵列逻辑 GAL ………… (210)
6.5.1　GAL 的基本逻辑结构 ………… (211)
6.5.2　GAL 的输出逻辑宏单元 OLMC ………… (213)
6.5.3　GAL 器件的开发过程 ………… (220)
6.6　在系统编程技术简介 ………… (221)
6.6.1　ISP 技术的主要特点 ………… (222)
6.6.2　ISP 可编程逻辑器件的类型 ………… (223)
6.6.3　ISP 器件的开发过程 ………… (227)
本章小结 ………… (230)
思考题 ………… (230)
习题 6 ………… (231)
第 7 章　综合应用举例 ………… (233)
7.1　汽车尾灯控制器设计 ………… (234)
7.1.1　设计要求 ………… (234)
7.1.2　功能描述 ………… (234)
7.1.3　电路设计 ………… (235)
7.2　弹道计时器设计 ………… (238)

7.2.1 设计要求 (238)
7.2.2 功能描述 (239)
7.2.3 电路设计 (240)
7.3 数字钟电路设计 (243)
7.3.1 设计要求 (243)
7.3.2 功能描述 (243)
7.3.3 电路设计 (244)
本章小结 (247)
附录 A 习题参考解答 (249)
附录 B 实验 (293)
参考文献 (321)

第1章 基本知识

21 世纪是信息的时代，数字系统已成为人们日常生活的组成部分。广泛应用于科学计算、数据处理和过程控制等领域的数字计算机就是一种典型的数字系统。数字系统所处理的信息是离散形式的，一种给定基数的数制将提供由数字系统处理量化信息的一种方法。那么，数字系统中采用何种进位制，有哪些数据表示形式？本章将讨论数字系统中数据的表示方法，包括常用计数制及其转换、带符号二进制数的代码表示及常用的几种编码。

1.1 数制及其转换

1.1.1 进位计数制

数制是人们对数量计数的一种统计规律。日常生活中广泛使用的是十进制，而数字系统中使用的是二进制。

1．十进制

十进制中采用了 0, 1, ⋯, 9 共 10 个基本数字符号，进位规律是“逢十进一”。当用若干个数字符号并在一起表示一个数时，处在不同位置的数字符号，其值的含义不同。

例如，

666

6×10^2　　6×10^1　　6×10^0

同一个字符 6 从左到右所代表的值依次为 600、60、6。即 $(666)_{10}=6\times10^2+6\times10^1+6\times10^0$。

2．*R* 进制

广义地说，一种进位计数制包含着基数和位权两个基本的因素。

基数：指计数制中所用到的数字符号的个数。在基数为 R 的计数制中，包含 0, 1, ⋯, R-1 共 R 个数字符号，进位规律是“逢 R 进一”。称为 R 进位计数制，简称 R 进制。

位权：指在一种进位计数制表示的数中，用来表明不同数位上数值大小的一个固定常数。不同数位有不同的位权，某一个数位的数值等于这一位的数字符号乘以与该位对应的位权。R 进制数的位权是 R 的整数次幂。

例如，十进制数的位权是 10 的整数次幂，其个位的位权是 10^0，十位的位权是 10^1……。

一个 R 进制数 N 可以有两种表示方法：

（1）并列表示法(又称位置计数法)，其表达式为

$$(N)_R=(K_{n-1}K_{n-2}\Lambda\ K_1K_0\cdot K_{-1}K_{-2}\Lambda\ K_{-m})_R$$

（2）多项式表示法(又称按权展开法)，其表达式为

$$(N)_R=K_{n-1}\times R^{n-1}+K_{n-2}\times R^{n-2}+\Lambda+K_1\times R^1+K_0\times R^0+K_{-1}\times R^{-1}+K_{-2}\times R^{-2}+\Lambda+K_{-m}\times R^{-m}$$

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

式中，R——基数；n——整数部分的位数；

m——小数部分的位数；

K_i——R 进制中的一个数字符号，其取值范围为 $0 \leq K_i \leq R_{-1}(-m \leq i \leq n-1)$。

R 进制的特点可归纳如下：

（1）有 0, 1, …, R-1 共 R 个数字符号；

（2）“逢 R 进一”，“10”表示 R；

（3）位权是 R 的整数次幂，第 i 位的位权为 R^i $(-m \leq i \leq n-1)$。

3．二进制

基数 R=2 的进位计数制称为二进制。二进制数中只有 0 和 1 两个基本数字符号，进位规律是“逢二进一”。二进制数的位权是 2 的整数次幂。

任意一个二进制数 N 可以表示成

$$(N)_2 = (K_{n-1}K_{n-2}\Lambda\ K_1K_0 \cdot K_{-1}K_{-2}\Lambda\ K_{-m})_2$$

$$= K_{n-1}\times 2^{n-1} + K_{n-2}\times 2^{n-2} + \Lambda\ + K_1\times 2^1 + K_0\times 2^0 + K_{-1}\times 2^{-1} + K_{-2}\times 2^{-2} + \Lambda\ + K_{-m}\times 2^{-m}$$

$$= \sum_{i=-m}^{n-1} K_i 2^i$$

式中，n——整数位数；m——小数位数；

K_i——为 0 或 1，$-m \leq i \leq n-1$。

例如，一个二进制数 1011.01 可以表示成:

$$(1011.01)_2 = 1\times 2^3+0\times 2^2+1\times 2^1+1\times 2^0+0\times 2^{-1}+1\times 2^{-2}$$

二进制数的运算规则如下：

加法规则	0+0=0	0+1=1
	1+0=1	1+1=0（进位为 1）
减法规则	0-0=0	1-0=1
	1-1=0	0-1=1（借位为 1）
乘法规则	0×0=0	0×1=0
	1×0=0	1×1=1
除法规则	0÷1=0	1÷1=1

例如，二进制数 A=11001, B=101, 则 $A+B$、$A-B$、$A\times B$、$A\div B$ 的运算为

$$\begin{array}{r} 11001 \\ +\quad 101 \\ \hline 11110 \end{array} \qquad\qquad \begin{array}{r} 11011 \\ -\quad 101 \\ \hline 10110 \end{array}$$

$$\begin{array}{r} 11001 \\ \times\quad 101 \\ \hline 11001 \\ 00000 \\ +\,11001 \\ \hline 1111101 \end{array} \qquad\qquad \begin{array}{r} 101 \\ 101\overline{)11001} \\ \underline{-101} \\ 101 \\ \underline{-101} \\ 0 \end{array}$$

二进制的优点：运算简单，物理实现容易，存储和传送方便、可靠。

因为二进制中只有 0 **和** 1 **两个数字符号，可以用电子器件的两种不同状态来表示一位二进制数。例如，可以用晶体管的截止和导通表示** 1 **和** 0**，或者用电平的高和低表示** 1 **和** 0 **等。所以，在数字系统中普遍采用二进制。**

二进制的缺点：数的位数太长且字符单调，使得书写、记忆和阅读不方便。因此，人们在进行指令书写、程序输入和输出等工作时，通常采用八进制数和十六进制数作为二进制数的缩写。

4．八进制

基数 R=8 **的进位计数制称为八进制。八进制数中有** 0,1,…,7 **共** 8 **个基本数字符号，进位规律是“逢八进一”。八进制数的位权是** 8 **的整数次幂。**

任意一个八进制数 N **可以表示成**

$$\begin{aligned}(N)_8 &= (K_{n-1}K_{n-2}\cdots K_1K_0\cdot K_{-1}K_{-2}\cdots K_{-m})_8 \\ &= K_{n-1}\times 8^{n-1}+K_{n-2}\times 8^{n-2}+\cdots+K_1\times 8^1+K_0\times 8^0+K_{-1}\times 8^{-1}+K_{-2}\times 8^{-2}+\cdots+K_{-m}\times 8^{-m} \\ &= \sum_{i=-m}^{n-1} K_i 8^i\end{aligned}$$

式中，n**——整数位数；**m**——小数位数；**

K_i——0～7 **中的任何一个字符，**$-m \le i \le n-1$。

5．十六进制

基数 R=16 **的进位计数制称为十六进制。十六进制数中有** 0, 1, …, 9, A, B, C, D, E, F **共** 16 **个字符，其中，**A～F **分别表示十进制数的** 10～15**。进位规律为“逢十六进一”。十六进制数的位权是** 16 **的整数次幂。**

任意一个十六进制数 N **可以表示成**

$$(N)_{16}=(K_{n-1}K_{n-2}\cdots K_1K_0\cdot K_{-1}K_{-2}\cdots K_{-m})_{16}$$
$$=K_{n-1}\times16^{n-1}+K_{n-2}\times16^{n-2}+\cdots+K_1\times16^1+K_0\times16^0+K_{-1}\times16^{-1}+K_{-2}\times16^{-2}+\cdots+K_{-m}\times16^{-m}$$
$$=\sum_{i=-m}^{n-1}K_i16^i$$

式中，n——整数位数；m——小数位数；K_i——0～9、A～F 中的任何一个字符，$-m\le i\le n-1$。

表 1.1 列出了与十进制数 0～15 对应的二进制数、八进制数、十六进制数。

表 1.1　十进制数与二、八、十六进制数对照表

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	00	0	8	1000	10	8
1	0001	01	1	9	1001	11	9
2	0010	02	2	10	1010	12	A
3	0011	03	3	11	1011	13	B
4	0100	04	4	12	1100	14	C
5	0101	05	5	13	1101	15	D
6	0110	06	6	14	1110	16	E
7	0111	07	7	15	1111	17	F

1.1.2　数制转换

数制转换是指将一个数从一种进位制转换成另一种进位制。从实际应用出发，要求掌握二进制数与十进制数、八进制数和十六进制数之间的相互转换。

1．二进制数与十进制数之间的转换

1）二进制数转换为十进制数

方法：多项式替代法

将二进制数表示成按权展开式，并按十进制运算法则进行计算，所得结果即为该数对应的十进制数。

例如，$(10111.001)_2=(\ ?\ )_{10}$

$$(10111.001)_2=1\times2^4+1\times2^2+1\times2^1+1\times2^0+1\times2^{-3}$$
$$=16+4+2+1+0.125$$
$$=(23.125)_{10}$$

2）十进制数转换为二进制数

方法：基数乘除法

十进制数转换成二进制数时，应对整数和小数分别进行处理。

整数转换——采用“除 2 取余”的方法；

小数转换——采用“乘 2 取整”的方法。

（1）整数转换。“除 2 取余”法：将十进制整数 N 除以 2，取余数计为 K_0；再将所得商除以 2，取余数记为 K_1；依次类推，直至商为 0，取余数计为 K_{n-1} 为止。即可得到与 N 对应的 n 位二进制整数 $K_{n-1}\cdots K_1K_0$。

例如，$(37)_{10}=(\ ?\ )_2$

			余数	
2	37			
2	18	……	1 (K_0)	低位
2	9	……	0 (K_1)	
2	4	……	1 (K_2)	
2	2	……	0 (K_3)	
2	1	……	0 (K_4)	
	0	……	1 (K_5)	高位

即 $(37)_{10}=(100101)_2$

（2）小数转换。“乘 2 取整”法：将十进制小数 N 乘以 2，取积的整数部分记为 K_{-1}；再将积的小数部分乘以 2，取整数部分记为 K_{-2}；依次类推，直至其小数部分为 0 或达到规定精度要求，取整数部分记为 K_{-m} 为止。即可得到与 N 对应的 m 位二进制小数 $0.K_{-1}K_{-2}\cdots K_{-m}$。

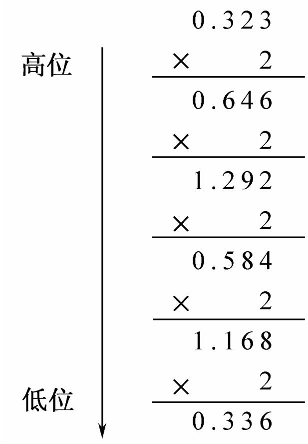
例如，$(0.8125)_{10}=(\ ?\ )_2$

	整数部分		
			0.8125
			× 2
高位	1 (K_{-1})	……	1.6250
			× 2
	1 (K_{-2})	……	1.2500
			× 2
	0 (K_{-3})	……	0.5000
			× 2
低位	1 (K_{-4})	……	1.0000

即 $(0.8125)_{10}=(0.1101)_2$

注意：当十进制小数不能用有限位二进制小数精确表示时，可根据精度要求，求出相应的二进制位数近似地表示。一般当要求二进制数取 m 位小数时，可求出 m+1 位，然后对最低位做 0 舍 1 入处理。

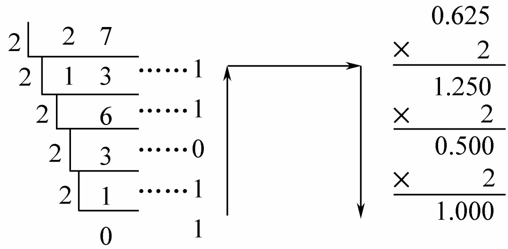
例如，$(0.323)_{10}=(\ ?\ )_2$**(保留 4 位小数)。**



即$(0.323)_{10}=(0.0101)_2$

若一个十进制数既包含整数部分，又包含小数部分，则需将整数部分和小数部分分别转换，然后用小数点将两部分结果连到一起。

例如，$(27.625)_{10} = (\ ?\ )_2$



即 $(27.625)_{10} = (11011.101)_2$

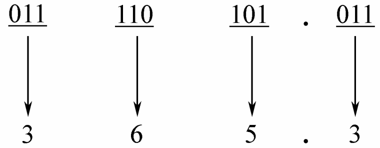
2．二进制数与八进制数、十六进制数之间的转换

1）二进制数与八进制数之间的转换

由于八进制的基本数字符号 0～7 **正好和** 3 **位二进制数的取值** 000～111 **对应，所以，二进制数与八进制数之间的转换可以按位进行。**

方法：以小数点为界，分别往高、往低每 3 **位为一组，最后不足** 3 **位时用** 0 **补充，然后写出每组对应的八进制字符，即为相应八进制数。**

例如，$(11110101.011)_2 = (\ ?\ )_8$



即 $(11110101.011)_2 = (365.3)_8$

八进制数转换成二进制数时，只需将每位八进制数用 3 位二进制数表示。

例如，$(46.5)_8 = (\ ？)_2$

$$\begin{array}{cccc} 4 & 6 & . & 5 \\ \downarrow & \downarrow & & \downarrow \\ \underline{100} & \underline{110} & . & \underline{101} \end{array}$$

即 $(46.5)_8 = (100110.101)_2$

2）二进制数与十六进制数之间的转换

二进制数与十六进制数之间的转换同样可以按位进行，只不过是 4 位二进制数对应 1 位十六进制数，即 4 位二进制数的取值 0000～1111 分别对应十六进制字符 0～F。

方法：以小数点为界，分别往高、往低每 4 位为一组，最后不足 4 位时用 0 补充，然后写出每组对应的十六进制字符即可。

例如，$(1101010.011)_2 = (\ ？)_{16}$

$$\begin{array}{cccc} \underline{0110} & \underline{1010} & . & \underline{0110} \\ \downarrow & \downarrow & & \downarrow \\ 6 & A & . & 6 \end{array}$$

即 $(1101010.011)_2 = (6A.6)_{16}$

十六进制数转换成二进制数时，只需将每位十六进制数用 4 位二进制数表示。

例如，$(8A.B)_{16} = (\ ？)_2$

$$\begin{array}{cccc} 8 & A & . & B \\ \downarrow & \downarrow & & \downarrow \\ \underline{1000} & \underline{1010} & . & \underline{1011} \end{array}$$

即 $(8A.B)_{16}=(10001010.1011)_2$

1.2　带符号二进制数的代码表示

为了标记一个数的正负，人们通常在一个数的前面用“＋”号表示正数，用“－”号表示负数。在数字系统中，符号和数值一样是用 0 和 1 来表示的，一般将数的最高位作为符号位，用 0 表示正，用 1 表示负。其格式为

$$\begin{array}{l} X_f\ X_{n-1}X_{n-2}\cdots X_1X_0 \\ \uparrow \\ \text{符号位} \end{array}$$

通常将用“+”、“-”表示正、负的二进制数称为符号数的真值，而把将符号和数值一起编码表示的二进制数称为机器数或机器码。

常用的机器码有原码、反码和补码 3 种。

1.2.1　原码

符号位用 0 表示正，1 表示负；数值位保持不变。原码表示法又称为符号—数值表示法。

1．小数原码

设二进制小数 X_1=+0.$x_{-1}x_{-2}\cdots x_{-m}$，$X_2$=-0.$x_{-1}x_{-2}\cdots x_{-m}$，则其原码为

$$[X_1]_原 = 0.x_{-1}x_{-2}\cdots x_{-m} \qquad [X_2]_原 = 1.x_{-1}x_{-2}\cdots x_{-m}$$

小数点前面的一位为符号位。

例如，若 X_1= +0.1011，X_2=-0.1011

则　$[X_1]_原$ = 0.1011　$[X_2]_原$=1.1011

小数“0”的原码可以表示成 0.0…0 或 1.0…0，分别表示+0.0…0 和-0.0…0。

2．整数原码

设二进制整数 X_1=+$x_{n-1}x_{n-2}\cdots x_0$, X_2=-$x_{n-1}x_{n-2}\cdots x_0$，则其原码为

$$[X_1]_原 = 0x_{n-1}x_{n-2}\cdots x_0 \qquad [X_2]_原 = 1x_{n-1}x_{n-2}\cdots x_0$$

即整数最高位为符号位。

例如，若 X_1=+1101 , X_2=-1101，则 X_1 和 X_2 的原码为

$$[X_1]_原 = 01101 \qquad [X_2]_原 = 11101$$

整数“0”的原码也有两种形式，即 00…0 和 10…0，分别表示+00…0 和-00…0。

原码的优点：简单易懂，求取方便。

原码的缺点：加、减运算不方便。当进行两数加、减运算时，要根据运算及参加运算的两个数的符号来确定是加还是减；如果是做减法，还需根据两数的大小确定被减数和减数，以及运算结果的符号。显然，这将增加运算的复杂性。

为了克服原码的缺点，引入了反码和补码。

1.2.2　反码

符号位与原码相同，即用 0 表示正，用 1 表示负；数值位与符号相关，正数反码的数值位和真值的数值位相同；而负数反码的数值位是真值的数值位按位变反。