高等学校电子信息类专业“十二五”规划教材
新课改教材

# 数字逻辑电路基础

陈彦辉　冯毛官　胡力山　编著

西安电子科技大学出版社
http://www.xduph.com

高等学校电子信息类专业“十二五”规划教材

**新课改教材**

# 数字逻辑电路基础

陈彦辉　冯毛官　胡力山　编著

西安电子科技大学出版社

## 内 容 简 介

本书全面介绍了数字电路和数字系统中常用电路及其基本模块的工作原理、分析和设计方法。全书共8章，主要内容包括数字化处理基础、逻辑门、逻辑电路描述、组合逻辑电路、触发器、时序逻辑电路、数据转换与存储及综合设计开发示例。各章均选用了多个典型实例进行讲解，便于读者联系实际，提高分析和解决问题的能力，达到灵活应用的水平。

本书可以作为高等学校电子信息与通信工程学科所有专业“数字电路和逻辑设计”课程的基础教材或教学参考书，亦可供其他专业师生及相关工程技术人员参考。

**图书在版编目(CIP)数据**

**数字逻辑电路基础**/陈彦辉，冯毛官，胡力山编著. —西安：西安电子科技大学出版社，2014.8
高等学校电子信息类专业“十二五”规划教材新课改教材
ISBN 978-7-5606-3450-0

Ⅰ. ① 数… Ⅱ. ① 陈… ② 冯… ③ 胡… Ⅲ. ① 数字电路—逻辑电路—高等学校—教材
Ⅳ. ① TN79

**中国版本图书馆 CIP 数据核字(2014)第 178887 号**

策　　划　李惠萍
责任编辑　王　斌
出版发行　西安电子科技大学出版社(西安市太白南路2号)
电　　话　(029)88242885　88201467　　邮　　编　710071
网　　址　www.xduph.com　　电子邮箱　xdupfxb001@163.com
经　　销　新华书店
印刷单位　陕西天意印务有限责任公司
版　　次　2014年8月第1版　2014年8月第1次印刷
开　　本　787毫米×1092毫米　1/16　印张 13.75
字　　数　322千字
印　　数　1～3000册
定　　价　24.00元
ISBN 978-7-5606-3450-0/TN

**XDUP 3742001-1**

# 前　言

“数字电路与逻辑设计”是电子信息类专业的基础课程。近年来数字电路发展迅猛，可编程逻辑器件在诸如微型控制、数字信号处理、数字通信、图像处理、信源编码、模式识别等工程领域应用非常广泛。原先仅以电路图和 MSI 器件为基础的教材越来越难以适应学生进行工程实践的需求。为了适应当前数字电路技术发展现状，不少学校都开设了可编程逻辑器件及硬件描述语言等课程。这些课程通常侧重于语言编程，没有与“数字电路与逻辑设计”课程相结合，使许多学生重于语言，忽视了电路设计基本方法(数字描述、真值表、状态图、波形图)和模块化设计，电路设计不合理。

为了解决这一问题，我们根据“数字电路与逻辑设计”课程基本要求，配合西安电子科技大学通信工程学院“数字电路与逻辑设计”课程的改革，吸取研究生导师和企业技术专家的建议和需求，充分考虑近年来数字电路的发展新趋势，结合多年来的教学经验，用心编写了此书，作为“数字电路与逻辑设计”课程的教材。

本书采用电路符号和硬件描述语言两种方式来描述数字逻辑电路，电路符号直观易懂，便于结构设计；硬件描述语言简单易写，便于电路设计，从而培养学生“结构在心、电路在手”的数字电路设计能力。

本书从数字信息处理的流程出发，面向实际问题，着重提高读者解决实际问题的能力。在内容的安排上，注重既要使教师便于组织教学，又要便于读者阅读和自学实践。编写时力求做到深入浅出，突出重点。在确保基础理论的前提下，将重点放在电路的模块化设计和实际问题的解决分析上。

我们在总结多年的教学实践并结合当前教学改革需要的基础上，对本书的实例进行了精心设计，并增加了一定量的设计性实验与综合性实验，旨在进一步提高学生的综合设计与应用能力。

本书组织结构如下：

第 1 章为数字化处理基础知识，主要介绍数字电路作为数字化处理的实现工具，如何完成算术运算、数据保存和流程控制这三个基本功能。

第 2、3 章为逻辑代数基础，介绍了基本的逻辑门与逻辑电路的原理和构成。在这里除了介绍传统的逻辑描述方法外，还强调了工程中常用的波形描述。

第 4～6 章分别介绍了组合逻辑电路、触发器与寄存器和时序逻辑电路。根据当前数字电路的发展趋势，示例采用 Verilog 语言实现，便于读者进行实验；同时强化模块化的设计思路，弱化具体器件，通过精心设计的示例引导读者掌握解决问题的工程技巧。

第 7 章介绍了数据转换及存储，介绍了信息处理的获取、显示和存储三个关键环节的主要原理、器件及设计思路。

第 8 章介绍了数字系统设计实例，通过几个常用的小型数字系统设计例子，使读者了解数字系统的设计方法和设计过程，并从系统的角度对学过的知识进一步加深理解。

本书内容前后连续性较强，示例设计有一定的难度。建议读者按章节顺序学习，并通过仿真实验来辅助理解和掌握知识点。

本书由西安电子科技大学通信工程学院“数字电路与逻辑设计”课程的教学教师编写完成。其中，第2～4章由胡力山和陈彦辉合编，第5～7章由冯毛官和陈彦辉合编，第1和8章由陈彦辉编写，陈彦辉负责全书的修改和统稿工作。

由于编者水平有限，书中难免存在疏漏和不妥之处，恳请广大读者批评指正。

编　者

2014年4月

# 目　录


**第 1 章　数字化处理基础** …… 1
1.1　信息处理流程的实现 …… 1
1.1.1　基本算术运算 …… 1
1.1.2　数据保存 …… 1
1.1.3　流程控制结构 …… 2
1.1.4　流程与部件 …… 4
1.2　数制 …… 5
1.2.1　进位计数制 …… 5
1.2.2　进位计数制转换 …… 6
1.3　编码 …… 7
1.3.1　有符号数的编码 …… 7
1.3.2　二－十进制编码(BCD 码) …… 9
1.3.3　格雷码(GRAY 码) …… 9
1.3.4　奇偶校验码 …… 10
1.3.5　ASCII 码 …… 10
**第 2 章　逻辑门** …… 12
2.1　布尔逻辑 …… 12
2.1.1　运算与逻辑 …… 12
2.1.2　二值逻辑运算 …… 13
2.2　逻辑门的描述 …… 14
2.2.1　基本逻辑门 …… 14
2.2.2　复合逻辑门 …… 16
2.3　逻辑门的电路实现 …… 18
2.3.1　开关逻辑电路 …… 18
2.3.2　继电器逻辑电路 …… 19
2.3.3　半导体逻辑电路 …… 20
2.4　集成逻辑门 …… 26
2.4.1　TTL 集成逻辑门 …… 26
2.4.2　CMOS 集成逻辑门 …… 30
2.4.3　集成门电路系列 …… 32
2.4.4　接口驱动 …… 33
2.4.5　常用 74LS 系列集成逻辑门 …… 33
**第 3 章　逻辑电路描述** …… 37
3.1　逻辑函数 …… 37
3.1.1　常用形式 …… 37
3.1.2　标准形式 …… 40

3.1.3 具有无关项的逻辑函数 …… 45
3.2 函数化简 …… 46
3.2.1 代数化简 …… 46
3.2.2 卡诺图化简 …… 47
3.3 逻辑电路结构 …… 49
3.3.1 组合电路 …… 49
3.3.2 时序电路 …… 50
3.4 硬件描述语言 …… 52
3.4.1 模块结构 …… 52
3.4.2 运算量与运算符 …… 54
3.4.3 行为语句 …… 56
3.4.4 描述示例 …… 58
**第 4 章 组合逻辑电路** …… 62
4.1 电路分析与设计 …… 62
4.1.1 电路分析 …… 62
4.1.2 电路设计 …… 63
4.2 数据选择/分配器 …… 67
4.2.1 数据选择器 …… 67
4.2.2 数据分配器 …… 68
4.2.3 应用示例 …… 69
4.3 编/译码器 …… 71
4.3.1 编码器 …… 71
4.3.2 译码器 …… 73
4.3.3 应用示例 …… 75
4.4 加/减法器 …… 78
4.4.1 全加/减器 …… 78
4.4.2 多位加/减法器 …… 81
4.4.3 应用示例 …… 82
4.5 数值比较器 …… 84
4.5.1 功能结构 …… 84
4.5.2 应用示例 …… 85
4.6 竞争与冒险 …… 88
4.6.1 竞争与冒险 …… 88
4.6.2 逻辑冒险的判别 …… 89
4.6.3 冒险现象的消除 …… 89
**第 5 章 触发器** …… 91
5.1 输出反馈电路 …… 91
5.2 基本 RS 触发器 …… 92
5.3 钟控触发器 …… 94
5.3.1 电平触发钟控触发器 …… 95
5.3.2 边沿触发器 …… 99
5.4 触发器的应用 …… 100
5.5 寄存器和移位寄存器 …… 102

5.5.1 寄存器 …… 102
5.5.2 移位寄存器 …… 106
**第 6 章 时序逻辑电路** …… 113
6.1 时序电路的描述 …… 113
6.1.1 时序电路的分类 …… 113
6.1.2 功能描述 …… 115
6.1.3 典型时序逻辑电路 …… 120
6.2 同步时序电路的分析与设计 …… 121
6.2.1 同步时序电路的分析 …… 121
6.2.2 同步时序电路的设计 …… 125
6.3 异步时序电路的分析 …… 129
6.4 计数器 …… 131
6.4.1 模 $2^n$ 同步二进制计数器 …… 131
6.4.2 模 $M$ 二进制计数器 …… 135
6.4.3 同步十进制计数器 …… 143
6.5 状态设计 …… 144
6.5.1 建立原始状态图和状态表 …… 144
6.5.2 状态化简 …… 146
6.5.3 状态分配 …… 150
6.6 常用时序电路设计 …… 151
6.6.1 分频器 …… 151
6.6.2 序列发生器 …… 155
6.6.3 序列检测器 …… 159
**第 7 章 数据转换与存储** …… 162
7.1 数/模转换器 …… 162
7.1.1 基本工作原理 …… 162
7.1.2 主要电路形式 …… 163
7.1.3 主要技术指标 …… 165
7.1.4 应用示例 …… 165
7.2 模/数转换器 …… 166
7.2.1 基本工作原理 …… 166
7.2.2 主要电路形式 …… 168
7.2.3 主要指标 …… 171
7.2.4 应用示例 …… 172
7.3 数据存储 …… 173
7.3.1 存储器原理 …… 173
7.3.2 只读存储器(ROM) …… 174
7.3.3 随机存取存储器(RAM) …… 177
7.4 存储器的应用 …… 179
7.4.1 存储扩展 …… 179
7.4.2 组合逻辑实现 …… 182
7.4.3 队列存储结构 …… 186
7.4.4 堆栈存储结构 …… 189

**第 8 章　综合设计开发示例** …… 193
8.1　多数码管显示 …… 193
8.1.1　工作原理 …… 193
8.1.2　模块设计 …… 194
8.1.3　仿真波形 …… 195
8.2　交通信号灯控制器 …… 196
8.2.1　工作原理 …… 196
8.2.2　模块设计 …… 197
8.2.3　仿真波形 …… 198
8.3　键盘扫描 …… 199
8.3.1　工作原理 …… 199
8.3.2　模块设计 …… 201
8.4　串型外设接口(SPI) …… 202
8.4.1　工作原理 …… 202
8.4.2　主 SPI 模块设计 …… 204
8.4.3　从 SPI 模块设计 …… 205
8.4.4　SPI 仿真波形 …… 206
8.5　语音合成器 …… 207
8.5.1　工作原理 …… 207
8.5.2　模块设计 …… 209
8.5.3　仿真波形 …… 210
**参考文献** …… 212

# 第 1 章　数字化处理基础

信息处理的流程通常由算术运算、数据保存和流程控制这三个基本要素构成。这三个基本要素的实现结构是数字化处理系统设计的前提，它给出了数字化处理的基本功能单元、组成部件及连接关系。

数字逻辑电路是现代数字信息处理的主要工具。信息通过各种方式变成数值，通过对数值进行各种运算来实现信息的处理。

二进制数仅采用 0 和 1 这两个字符，便于电路实现。基于二进制数的数值表示、计算公式和信息表达方式都是数字化处理的操作基础。

## 1.1　信息处理流程的实现

信息处理流程通常由算术运算、数据保存、流程控制等三个基本要素构成。

### 1.1.1　基本算术运算

基本算术运算主要包括加、减、乘、除。这些基本算术运算通过组合实现各种数学表达式的计算。例如，$y=(x_1+x_2)x_3-(x_4+x_5)/x_6$可以采用如图 1.1 所示的结构来实现。

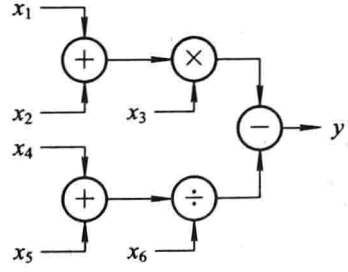



图 1.1　数学表达式的实现结构举例

任何一种处理都可以看成是一个部件，它有输入和输出，其功能可以用函数来表示。例如，输入为 $x$，输出为 $y$，该部件功能函数为 $y=f(x)$，$f$ 代表功能。基本算术运算所对应的部件分别称为加法器、减法器、乘法器和除法器。

### 1.1.2　数据保存

在处理过程中有许多计算结果需要暂时保存供后面运算使用。具有暂时保存功能的部件称为寄存器。并不是所有计算结果都需要保存，若中间结果立即被使用，则不需要保存。

例如，$y_1=f_1(x)$，$y=f_2(y_1)$，这里的 $y_1$ 不需要保存。又例如，$s=s+1$，这里的 $s$ 需要保存，每次计算是对 $s$ 的更新。

寄存器通常有一个触发输入，只有当触发输入有效时才会保存输入的数据。只要没有

新的触发输入，寄存器一直保存当前值。

寄存器还有两个特殊控制输入：一个是使数据为全 0，即清零或置 0；另一个是使数据为全 1，即置位或置 1。这两种控制通常用来初始化寄存器。

### 1.1.3 流程控制结构

流程控制结构主要有顺序结构、分支结构和循环结构。顺序结构就是按顺序逐一执行操作。下面主要介绍分支结构和循环结构。

**1. 分支结构**

分支结构是指根据不同条件执行不同的操作。例如：

$$y=\begin{cases} f_1(x) & x<a \\ f_2(x) & a\leqslant x<b \\ f_3(x) & b\leqslant x<c \\ f_4(x) & x\geqslant c \end{cases}$$

分支结构可以分为两个功能部分：条件判断和选择执行。前者用来决定哪个条件成立，生成相应的条件号；后者根据前者生成的条件号来决定哪个结果被输出。

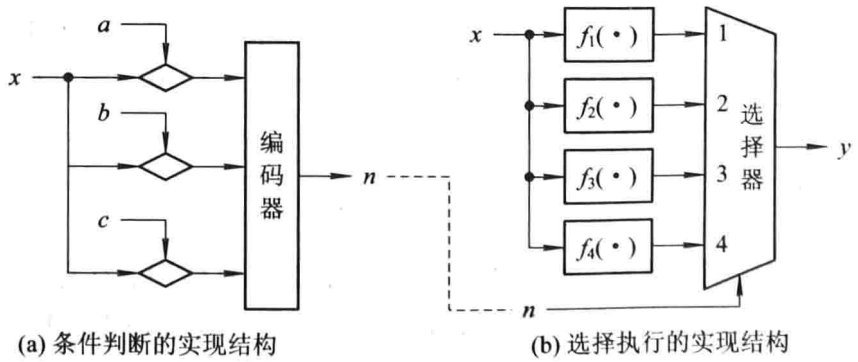
1）条件判断

条件判断可以描述为：当 $x<a$ 时，条件 1 成立，其他条件不成立；当 $a\leqslant x<b$ 时，条件 2 成立，其他条件不成立；当 $b\leqslant x<c$ 时，条件 3 成立，其他条件不成立；当 $x\geqslant c$ 时，条件 4 成立，其他条件不成立。

在条件判断中，$x$ 分别与三个数相比较，即 $x$ 与 $a$、$x$ 与 $b$、$x$ 与 $c$。比较运算实质是一种数学运算，差为 0 表明相等；差是正数表明是大于关系；差是负数表明是小于关系。进行比较运算的部件称为比较器。将 $x$ 与这三个数比较的结果进行组合，输出相应的条件号，该部件称为编码器。条件判断的实现结构如图 1.2(a)所示。

2）选择执行

选择执行可以描述为：当条件 1 成立时，$y=f_1(x)$；当条件 2 成立时，$y=f_2(x)$；当条件 3 成立时，$y=f_3(x)$；当条件 4 成立时，$y=f_4(x)$。采用如图 1.2(b)所示的描述方式，每个函数都对输入 $x$ 进行运算，根据条件号选择其所对应的函数计算结果作为输出。根据条件从若干个输入中选择其一作为输出的部件称为选择器。




(a) 条件判断的实现结构　　(b) 选择执行的实现结构

图 1.2　分支结构的实现结构

选择执行也可以写成加权和形式，即

$$y = c_1 f_1(x) + c_2 f_2(x) + c_3 f_3(x) + c_4 f_4(x)$$

当 $n=1$ 时，$c_1=1$，$c_2=c_3=c_4=0$；当 $n=2$ 时，$c_2=1$，$c_1=c_3=c_4=0$；当 $n=3$ 时，$c_3=1$，$c_1=c_2=c_4=0$；当 $n=4$ 时，$c_4=1$，$c_1=c_2=c_3=0$。$c_i$通常称为使能。带使能的函数 $y_i=f_i(c_i, x)$的表达式为

$$y_i = \begin{cases} f_i(x) & c_i = 1 \\ 0 & c_i = 0 \end{cases}$$

加权和形式的选择执行可以采用如图 1.3 所示的结构实现，其中译码器是将条件号转变为相应的使能信号。

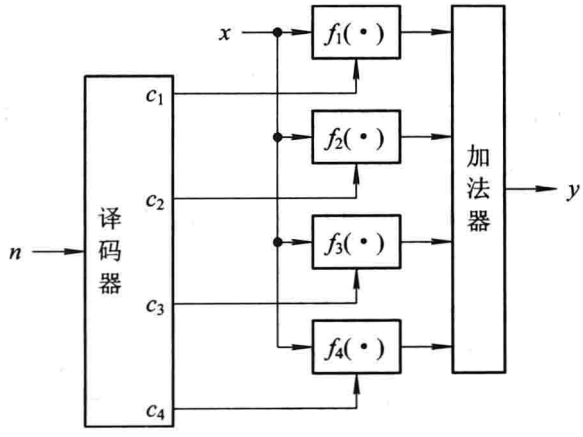



图 1.3　加权和形式的选择执行实现结构

**2. 循环结构**

有的处理操作需要对各种类型的操作反复执行，这种结构为循环结构。例如，阶乘运算 $y=x!$ 的计算流程如图 1.4 所示。

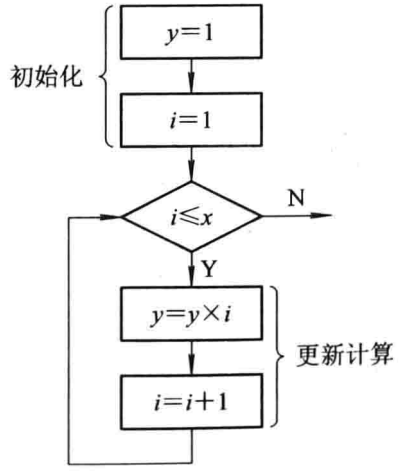



图 1.4　阶乘运算 $y=x!$ 的计算流程

循环结构通常分为初始化和循环体两部分。处理过程是先进行初始化，再根据条件反复执行循环体。初始化时对循环体内所用到的量赋初值，执行循环体时对这些量进行更新计算。

每次循环操作称为一个节拍，每个节拍中需要完成 $y=y\times i$ 和 $i=i+1$，$y$ 和 $i$ 每个节

拍更新一次，新的值与前一个节拍有关，因此 $y$ 和 $i$ 需要采用寄存器来保存，并采用时钟来表示节拍，每个节拍寄存器保存一次数据。阶乘循环计算流程的实现结构如图 1.5 所示。

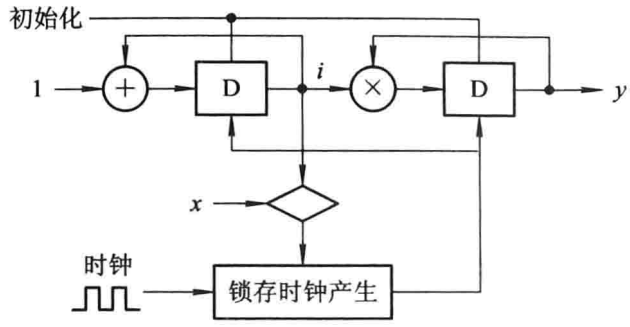



图 1.5 阶乘循环计算流程的实现结构

图 1.5 中的 D 框代表寄存器，初始化使 $i$ 和 $y$ 的寄存器置 1。若此时 $x \geqslant 1$，锁存时钟输出，$y$ 和 $i$ 的寄存器在每个时钟到达时更新一次。当 $i > x$ 时，没有锁存时钟输出，$y$ 和 $i$ 不再变化，此时输出的 $y$ 值为 $x!$。

以上计算中的 $i$ 实质是用来记录运算(节拍)的次数，故 $i$ 的实现结构是一个计数器，它在时钟的触发下每次加 1。

计数器的实现结构如图 1.6 所示，其中使能有效时选择器输出加法器的结果使之保存，相当于计数；使能无效时(未使能)选择保存现已保存的值，相当于保持。

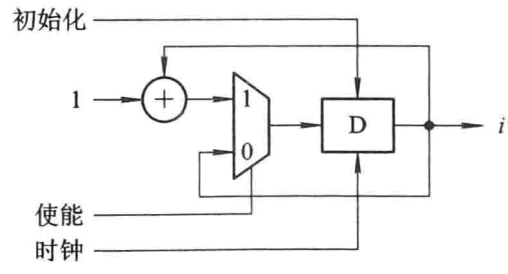



图 1.6 计数器的实现结构

图 1.6 中的计数器是受控的，初始化后共记录 $x$ 次后停止，故需将 $i$ 值与 $x$ 值进行比较，当 $i \leqslant x$ 时使能有效，其他无效。

阶乘的输出也受使能的控制，使能有效时计算，无效时保持。基于计数器的阶乘实现结构如图 1.7 所示。

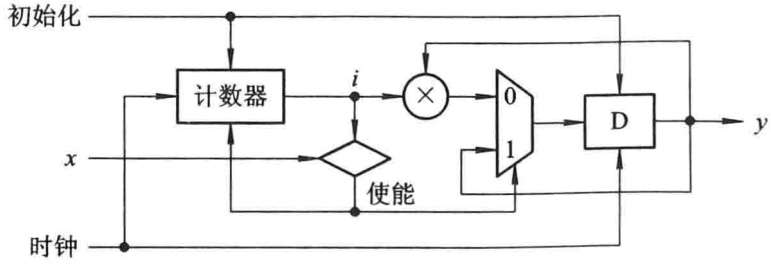



图 1.7 基于计数器的阶乘实现结构

### 1.1.4 流程与部件

信息处理流程可以采用算术运算器(加法器、减法器、乘法器和除法器)来处理数据；采用寄存器来保存数据；采用比较器、编码器、选择器、译码器等实现分支处理流程；采用寄存器、分支处理和计数器实现循环处理流程。

信息处理流程若出现反复操作或局部流程回转的现象，说明流程是有记忆的，需要采用寄存器来实现数值的保存。

# 1.2 数　制

## 1.2.1 进位计数制

按进位的原则进行计数称为进位计数制。$R$ 进制数有 $R$ 个允许使用的数符，它们也被称为基数或底数，若干个数符排列在一起表示一个 $R$ 进制数，进位规则是“逢 $R$ 进一”。

任何一个数都是由整数和小数两部分组成的。以小数点为界，左面为整数位，右面为小数位。假定数值 $N$ 有 $n$ 个整数位、$m$ 个小数位，该数采用位置计数法可写为

$$N=(a_{n-1}a_{n-2}\cdots a_1a_0.a_{-1}a_{-2}\cdots a_{-m})_R$$

其中，$a_i$ 是第 $i$ 位的数符，$-m\leqslant i<n$。

数符在不同位置所代表的数值不同，称数位所代表的值为权，它是 $R$ 的幂。整数位序号从 0 开始，从低位向高位递增，权依次为 $R^0$、$R^1$、$R^2$、…、$R^{n-1}$；小数位序号从 $-1$ 开始，从高位向低位递减，权依次为 $R^{-1}$、$R^{-2}$、…、$R^{-m}$。第 $i$ 位 $a_i$ 所表示的值为 $a_i\times R^i$。数值 $N$ 采用多项式表示法可写为

$$\begin{aligned}N&=a_{n-1}\times R^{n-1}+a_{n-2}\times R^{n-2}+\cdots+a_1\times R^1+a_0\times R^0+a_{-1}\times R^{-1}+a_{-2}\times R^{-2}\\&\quad+\cdots+a_{-m}\times R^{-m}\\&=\sum_{i=-m}^{n-1}a_i\times R^i\end{aligned}$$

其中，$R$ 的取值最小为 2，最为常用的为 10，$R$ 通常还取 16、8。

**1. 十进制数**

十个数符为 0、1、2、3、4、5、6、7、8、9，进位规则是“逢十进一”，可以采用下标法表示(如$(34.56)_{10}$)，也可以采用后缀法表示(如 345D)。十进制数可以省略下标或后缀，如 34.56、345。

**2. 十六进制数**

十六个数符为 0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F，其中 A、B、C、D、E、F 分别代表值 10、11、12、13、14、15，进位规则为“逢十六进一”，既可以采用下标法和后缀法(如$(158)_{16}$、158H)表示，也可以采用 0x158 来表示。

例如，数$(158)_{16}$可写为 158H 或 0x158，它的值为

$$(158)_{16}=1\times16^2+5\times16+8=256+80+8=344$$

数$(\mathrm{FE03.A})_{16}$的值为

$$\begin{aligned}(\mathrm{FE03.A})_{16}&=15\times16^3+14\times16^2+3+10\times16^{-1}=61\ 440+3584+3+0.625\\&=65\ 027.625\end{aligned}$$

**3. 八进制数**

八个数符为 0、1、2、3、4、5、6、7，进位规则为“逢八进一”，通常采用下标 8 和后缀 O 来表示。例如：

$$(1472)_8=1\times8^3+4\times8^2+7\times8+2=512+256+56+2=826$$

$$(473.72)_8=4\times8^2+7\times8+3+7\times8^{-1}+2\times8^{-2}$$
$$=256+56+3+0.875+0.031\ 25=315.906\ 25$$

**4. 二进制数**

两个数符为 0 和 1，进位规则为“逢二进一”，表示时采用下标 2 或后缀 B。例如：

$$(10110)_2=1\times2^4+1\times2^2+1\times2=16+4+2=22$$

$$1101.01\text{B}=1\times2^3+1\times2^2+0\times2+1+0\times2^{-1}+1\times2^{-2}$$
$$=8+4+1+0.25=13.25$$

二进制数只有两个数符，单位运算简单。二进制数运算规则与十进制数运算规则相似，只是“逢二进一”和“借一当二”，如图 1.8 所示。

```
  1010.01        1010.01          1010          110
+   110.11     -   110.11      ×    110     101/11111
----------     ----------     ----------     -101
  10001.00       11.10            0000        101
                                   1010        -101
                +  1010          ----------    ----
                ----------         111100        1
                  111100
```

图 1.8　二进制数运算规则示例

## 1.2.2　进位计数制转换

**1. 二进制数与十进制数之间的相互转换**

(1) 二进制数变为十进制数。在二进制数转换为十进制数时，只需要按权值展开即可。

(2) 十进制数变为二进制数。需要对数的整数部分和小数部分分别进行转换，具体方法如下：

① 整数部分采用除 2 取余法：将数值除以 2，将余数记为 $a_0$，若商不为 0，再将商除以 2，将余数记为 $a_1$，若商不为 0，再将商除以 2，……直到商为 0 为止。将余数从低向高排列，即可得到相应的二进制数。例如，将 345 转换为二进制数$(110101101)_2$，其计算过程如图 1.9 所示。

```
2 | 345
2 | 172  ----- 1
2 |  86  ----- 0
2 |  43  ----- 1
2 |  26  ----- 1
2 |  13  ----- 0
2 |   6  ----- 1
2 |   3  ----- 0
2 |   1  ----- 1
    0    ----- 1

      0.375          0.3
    ×     2        ×     2
  ----------     ----------
0 -----  .75    0 -----  .6
    ×     2        ×     2
  ----------     ----------
1 -----  .5     1 -----  .2
    ×     2        ×     2
  ----------     ----------
1 -----  .0     0 -----  .4
                     ×     2
                   ----------
                 0 -----  .8
                     ×     2
                   ----------
                 0 -----  .6
```

图 1.9　十进制数与二进制数之间的转换示例

② 小数部分采用乘 2 取整法：将数值乘以 2，将积的个位记为 $a_{-1}$，若积的小数部分不为 0，再将积的小数部分乘以 2，将积的个位记为 $a_{-2}$，若积的小数部分不为 0，再将积的小数部分乘以 2，……直到积的小数部分为 0 为止。将所得到的个位值从高到低排列，即可得到相应的二进制数。例如，将 0.375 转换为二进制数$(0.011)_2$，大多数值无法达到积的小数部分为 0。通常规定达到一定小数位即可，如 $0.3=(0.010001000\cdots)_2\approx(0.010001)_2$。

**2. 二进制数与八进制数之间的相互转换**

从小数点开始，分别向左、向右将二进制数每三位为一组，所有组的数值连接在一起，即构成其相应的八进制数。例如：

$$(001\quad 110\quad 101\quad 010.011\quad 110\quad 010)_2$$
$$=(\ 1\quad\ 6\quad\ 5\quad\ 2\ .\ 3\quad\ 6\quad\ 2\ )_8$$

从小数点开始，分别向左、向右将八进制数每位数符采用三位二进制数代替，所有位的二进制数连接在一起，即构成其相应的二进制数。例如：

$$(\ 3\quad\ 7\quad\ 6\quad\ 1\ .\ 0\quad\ 5\quad\ 7\ )_8$$
$$=(011\quad 111\quad 110\quad 001.000\quad 101\quad 111)_2$$

**3. 二进制数与十六进制数之间的相互转换**

从小数点开始，分别向左、向右将二进制数每四位为一组，所有组的数值连接在一起，即构成其相应的十六进制数。例如：

$$(0011\quad 1010\quad 1010.0111\quad 1001)_2$$
$$=(\ 3\quad\ A\quad\ A\ .\ 7\quad\ 9\ )_{16}$$

从小数点开始，分别向左、向右将十六进制数每位数符采用四位二进制数代替，所有位的二进制数连接在一起，即构成其相应的二进制数。例如：

$$(\ 7\quad\ F\quad\ 1\ .\ 1\quad\ 7\quad\ 8\ )_{16}$$
$$=(0111\quad 1111\quad 0001.0001\quad 0111\quad 1000)_2$$

## 1.3 编　　码

数字系统采用二进制数来表示数据信息。编码就是采用若干位二进制数来表示给定的信息，所规定的位数称为字长。下面介绍几种常用的编码。

### 1.3.1 有符号数的编码

数的编码分为无符号数编码和有符号数编码。无符号数是指数值是非负数，所有位都是数值，而有符号数是有正负之分的，将最高位作为符号位，0 表示正，1 表示负，其他位为数值。有符号数的编码可以采用原码、反码和补码这三种形式来表示。

**1. 原码**

原码就是在绝对值的二进制数值前面加上符号。

例如：

$$[+15]_{原}=01111,\ [-15]_{原}=11111$$

将(+15)+(−15)按原码进行二进制加法运算，即

$$\begin{array}{r} 01111 \\ +\ 11111 \\ \hline 101110 \end{array}$$

其结果是101110而不是0，所以原码不适合直接进行加法运算。

对于字长为$n$的原码，其正数范围为$[1, 2^{n-1}-1]$，负数范围为$[-(2^{n-1}-1), -1]$，共有$2^n-1$个编码，不存在只有最高位为1其他位为0的编码。对$n=4$的原码，正数原码为0001～0111，即1～7；负数原码为1001～1111，即－1～－7；不存在编码1000。

**2. 反码**

正数的反码与其原码相同，负数的反码是将其绝对值的数值按位取反后再与符号位连接而成。例如：

$$[+15]_{反} = 01111, \quad [-15]_{反} = 10000$$

对于字长为$n$的反码，其正数范围为$[1, 2^{n-1}-1]$，负数范围为$[-(2^{n-1}-1), -1]$，共有$2^n-1$个编码，不存在全1的编码。对$n=4$的反码，正数的反码为0001～0111，即1～7；负数的反码为1110～1000，即－1～－7；不存在编码1111。反码不适合直接进行加法运算。

**3. 补码**

对于字长为$n$的补码，正数的补码与其原码相同，负数的补码是$2^n$减去其绝对值所得到的二进制数。例如：

$$[+15]_{补} = 01111, \quad [-15]_{补} = 10001$$

需要注意的是，$-2^{n-1}$是可以编码的，它所对应的二进制数值为$2^{n-1}$。其正数范围为$[1, 2^{n-1}-1]$，负数范围为$[-2^{n-1}, -1]$，共有$2^n$个编码。对$n=4$的反码，正数的反码为0001～0111，即1～7；负数反码为1110～1000，即－1～－8。

将(＋15)＋(－15)按补码进行二进制加法运算，即

$$\begin{array}{r} 01111 \\ +\ 10001 \\ \hline 00000 \end{array}$$

取低5位的值为0，可以进行加法运算。

将(＋15)－(－15)按补码进行二进制减法运算，即

$$\begin{array}{r} 01111 \\ -\ 10001 \\ \hline 11110 \end{array}$$

结果为11110，按无符号数来看是30，但按有符号数来看是2。这主要是由于计算结果超过有效数字位即字长不够造成的。采用字长为6，$[+15]_{补}=001111$，$[-15]_{补}=110001$，两者的和和差都正确了，即

$$\begin{array}{r} 001111 \\ +\ 110001 \\ \hline 000000 \end{array} \qquad \begin{array}{r} 001111 \\ -\ 110001 \\ \hline 011110 \end{array}$$

补码是数字处理最为常用的表达方式。