DIANZIKEJIDAXUECHUBANSHE
XILIEJIAOCAI

高等学校
电子信息类 规划教材

大专计算机

数字电路与逻辑设计

顾藏知　陈战平　编著

电子科技大学出版社
UESTC PUBLISHING HOUSE

高等专业学校
电子信息类 规划教材

数字电路与逻辑设计

顾藏知 陈战平 编著

电子科技大学出版社

高等专业学校
电子信息类 规划教材

数字电路与逻辑设计

顾藏知　陈战平　编著

出　　版：电子科技大学出版社　(成都建设北路二段四号，邮编：610054)
责任编辑：吴艳玲
发　　行：新华书店经销
印　　刷：四川建筑印刷厂
开　　本：787×1092　1/16　　印张　18.75　　字数　456千字
版　　次：1999年7月第1版
印　　次：2000年3月第二次印刷
书　　号：ISBN 7—81065—171—4/TN·12
印　　数：4001—7000册
定　　价：21.00元

前　言

本教材系按电子工业部的《1996—2000年全国电子信息类专业教材编审出版规划》，由大专计算机专业教学指导委员会编审、推荐出版。本教材列入上述规划的部级重点教材。本教材由大专计算机专业教学指导委员会副主任，南京动力高等专科学校顾藏知副教授担任主编，主审由东南大学无线电工程系教授李兆宏担任，责任编委由上海机械高等专科学校唐俊杰副教授担任。

本教材的参考时数为60学时，其主要内容为：逻辑代数、逻辑函数、组合逻辑电路的分析和设计、时序逻辑电路的分析和设计、存储器电路、脉冲的产生和整形、数/模转换和模/数转换，除了以上内容外，又增加了实用性较强的线路板计算机辅助设计和可编程器件的应用。从培养跨世纪人才的总目标出发，为了适应计算机迅速发展的形势，根据大专计算机专业教学实践性、应用性、针对性的要求，本教材在编写过程中，特别注意加大教材的改革力度，特别注意介绍新技术，特别注意吸收多年教学实践中的成果，希望同行在使用本教材的同时能对此有所认同，也希望学生在使用本教材的同时，会有所受益。为了实现大专计算机专业教学指导委员会多出精品教材的要求，本教材还要在今后使用的过程中不断修改，精益求精，力求以高标准满足教、学两方面的要求。

本教材在使用过程中应注意本学科实践性强的特点，要配合相关的实验以加强学生的理解，要介绍相关的产品以加强学生的实践观念，在课程结束后还应安排课程实践环节，综合运用所学到知识，这是学好本课程的重要辅助手段。

为了配合电化教学，节省宝贵的课堂教学时间，加大课堂教学的信息量，本教材的所有插图都精心绘制，并配有教学挂图和投影薄膜，可供教师选用。

本教材由顾藏知副教授编写大纲和确定编写的框架及总体思路，由陈战平老师编写，最后由顾藏知副教授统稿，另由张春良老师编写了全部习题，经顾藏知副教授审定。参加审阅工作的还有南京大学计算机科学系系主任张福炎教授，他为本书提出许多宝贵意见，在此表示诚挚的感谢。由于编者水平有限，书中难免还存在一些缺点和错误，殷切希望广大读者批评指正。

编者

1998年7月24日

第一章　数制与码制

§1-1　绪　　论

一、什么是数字电路

电子电路的工作信号，可分为模拟信号和数字信号两大类。

模拟信号在时间上和数值上的变化是连续的。如图 1-1-1 所示。

而数字信号是指在时间上和数值上都是离散的。信号的变化只发生在一系列离散的瞬间，并且信号的数值是一些离散的电平值（高电平或低电平），如图 1-1-2 所示。数字信号是用一系列离散的数量来表示某个实际变化的全过程。如图 1-1-3（a）、（b）所示。

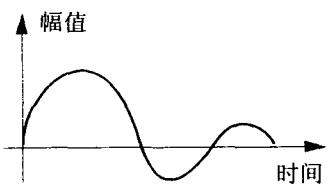


图 1-1-1　模拟信号

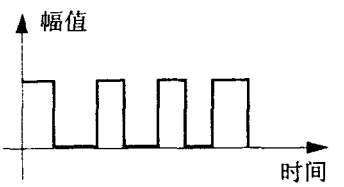
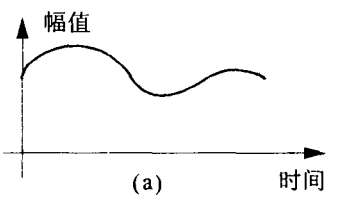


图 1-1-2　典型数字脉冲信号



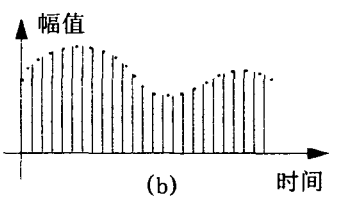


图 1-1-3　数字信号表示连续变化的模拟信号

用数字信号完成对数字量进行算术运算和逻辑运算的电路称为数字电路，或数字系统。由于它具有逻辑运算和逻辑处理功能，所以又称为数字逻辑电路。数字逻辑电路按功能大致可以分为以下两类：

1. 组合逻辑电路

简称组合电路，它是由最基本的逻辑门电路组合而成。组合电路的特点是：输出值只与当时的输入值有关，即输出唯一地由当时的输入值决定。电路没有记忆功能，输出状态随着输入状态的变化而变化。它类似于电阻性电路。

如课程中所要介绍的加法器、译码器、编码器、数据选择器等都是用最基本的逻辑门按一定的逻辑功能组合而成的电路。组合电路的基本组成单元是各种基本逻辑门。

2. 时序逻辑电路

简称时序电路，它是由最基本的逻辑门电路加上反馈逻辑回路（输出到输入）或器件组合而成的电路，与组合电路最本质的区别在于时序电路具有记忆功能。电路的特点是：输出不仅取决于当时的输入值，而且还与电路过去的状态有关。它类似于含储能元件的电感或电容的电路。

如课程中将要介绍的触发器、锁存器、计数器、移位寄存器、储存器等电路，都是时序电路的典型器件。时序电路是由最基本的触发器电路和组合电路所构成的。

近年来随着半导体技术和材料科学领域各种相关技术的飞速发展，以及集成电路制造工艺日新月异的进步，数字电路已经成为现代电子技术的极其重要的组成部分。与数字电路相关的应用技术如：电子计算机、数字仪器、数字通信、数控装置、雷达、电视等也迅猛发展起来。作为现代科学技术的杰出成就之一的数字电子计算机就是在数字电子技术的基础上发展起来的。数字电路技术的发展不仅推动了计算机的普及和应用，而且还带动了其他领域电子技术的腾飞，所以了解数字电路的功能和应用以便能进行合理的选择，正确的运用，是数字时代技术人员所应该具有的基本素质。所以，数字电路是计算机及其相关应用专业的一门必修的专业基础核心课程。

二、数字电路的特点

数字电路之所以能得到越来越广泛的应用，是和它具有的特点分不开的。

1. 同时具有算术运算和逻辑功能

数字电路是以二进制逻辑代数为数学基础，使用二进制数字信号，既能进行算术运算又能方便地进行逻辑运算（与、或、非、判断、比较、处理等），因此极其适合于运算、比较、存储、传输、控制、决策等应用。相比之下如果采用模拟系统完成同样的功能，从数学运算到电路实现所需要的设备较数字系统相比要复杂得多。

2. 实现简单，系统可靠

以二进制作为基础的数字逻辑电路，简单可靠，准确性高。因为在数字系统中，信号只有两个基本的离散量——“0”和“1”。对应逻辑代数二进制数的“0”和“1”状态，“0”和“1”可以表示任何事物的两个对立面（有、无，高、低，是、否，好、坏等等）。再用这两个最基本的状态组合，可以表达客观事物间的任何复杂关系，并能进行推理、运算。在物理电路中，它们对应着半导体二极管、三极管电子元件的导通和截止这两个不同的开关状态，所以，工作速度快，抗干扰能力强，在传递、加工和处理信息时不易出错，增加了系统的可靠性和准确性。

3. 集成度高，实现容易

集成度高，体积小，功耗低是数字电路突出的优点之一。电路的设计、维修、维护灵活方便，随着集成电路技术的高速发展，数字逻辑电路的集成度越来越高，集成电路块的功能随着小规模集成电路（SSI），中规模集成电路（MSI），大规模集成电路（LSI），超大规模集成电路（VLSI）的发展也从元件级、器件级、部件级、板卡级上升到系统级。电路的设计组成只需采用一些标准的集成电路块单元联接而成。对于非标准的特殊电路还可以使用可编程序逻辑阵列电路，通过编程的方法实现任意的逻辑功能。

三、数字电路的基本逻辑关系

从数字电路基本逻辑关系图 1-1-4 中可以了解数字电路的基本概貌，数字电路的基本组合构架，及数字电路最终应用系统的组成概貌。

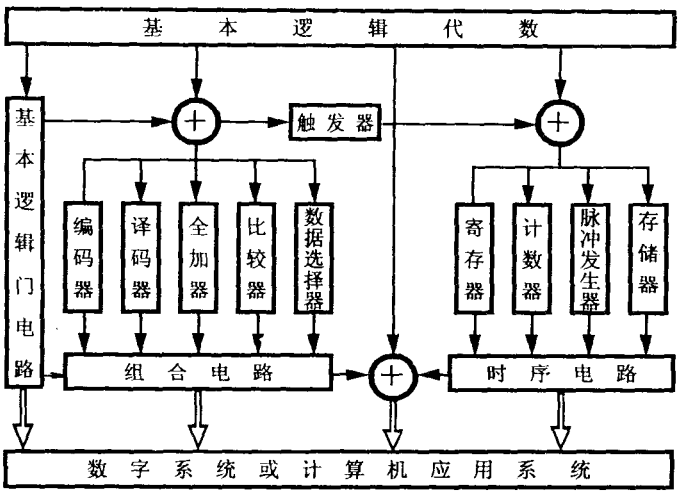


图 1-1-4　数字电路基本逻辑关系

从图中可以看出数字电路的数学描述以逻辑代数为基础，基本逻辑门电路实现了逻辑代数的基本运算功能（与、或、非），基本逻辑门按基本逻辑代数的运算规律又分别构成了标准应用电路，编码器、译码器、全加器、比较器和数据选择器等组合电路。

另一分支为：基本逻辑门电路按照逻辑代数规则构成基本触发器电路，由触发器作为基本电路构成了寄存器、计数器、存储器、脉冲发生器等另一类时序电路。

以基本电路和标准电路为基础，按照组合电路和时序电路的设计原则和方法可构成数字电路应用系统。

这是本课程所要讲解的主要内容框架。

四、数字电路与逻辑设计的基本方法

数字逻辑电路的理论基础是逻辑代数，逻辑代数最大的特点是二值代数，使用逻辑代数的二值变量和二值函数的概念，研究、分析和设计数字电路输入和输出之间的逻辑关系。对于一个给定的逻辑电路，研究它的逻辑功能，也就是对已知的逻辑电路，找出它的逻辑功能，并用逻辑函数来描述它的工作，称为逻辑分析。根据给定的逻辑要求，先确定要完成的逻辑功能，再求出相应的逻辑电路，称为逻辑设计。逻辑分析和逻辑设计是互逆的。

本课程主要讨论逻辑电路的基本组成原理，介绍如何分析和设计逻辑电路的方法。

研究数字逻辑电路，首先要理解和掌握逻辑代数的基本运算规则。在此基础上学会和掌握实用的分析和设计方法，培养具有处理实际电路问题的能力。

由于集成电路的高速发展，数字逻辑电路的设计已经标准化、模块化和系统化，因此在掌握逻辑代数的基本运算规则和分析设计方法的基础上，就可以选用成熟的已有基本模块来构成电路，寻求最好的设计，以满足全面的性能指标。经典的开关理论和最小化设计

思想，仍然是分析和设计逻辑电路的基础，对于逻辑设计仍然具有十分重要的指导意义。

在学习本课程时要注意在具体的数字电路与基本逻辑设计方法之间应以设计方法为主，在具体的设计步骤与所依据的基本原理和概念之间，应以原理和概念为主，在集成电路的内部工作原理与外部特性之间，应以外部特性为主，以便抓住重点。我们希望通过本课程的学习，能为数字电路应用及逻辑设计打下良好的基础。

§1-2 数制与转换

数字电路和计算机中数的表示形式决定了计算机的结构和性能。如图 1-2-1 所示。

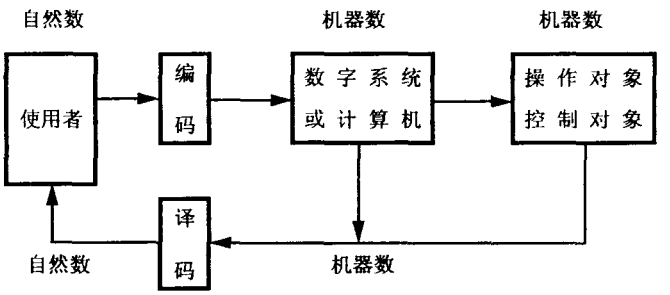


图 1-2-1 人机交流内部信号流程

数字系统、计算机系统都是一种能快速可靠地执行某种操作，实现某些功能的工具。人类的自然语言无法在机器内部运行，而机器高速运行输出的结果人类也无法直接判读。所以在人机接口处必须加以转换，以使两者之间得以沟通。

二进制在数字系统和电子计算机中得到广泛使用，因为它具有以下特点：

1. 具有完整的理论

二进制是二值计数，逻辑代数是二值代数。逻辑代数与二进制之间存在着对应关系，因此用逻辑代数的理论作为设计二进制系统的数学工具。

2. 容易实现

由于二进制只取两个数码“0”和“1”两个数值，因此它有一个很大的优点，就是它的每一位都可以用任何具有两个不同稳定状态的元件来表示，而这种元件的制造比需要多个稳定状态的元件要容易得多。这样，在数字电路或计算机中，数的存储和传送，就可用简单而可靠的方式进行，如脉冲的有无，电位的高低，信号的正负，晶体管的导通和截止等等方式来实现。

3. 运算简单

二进制的运算规则非常简单，实现这种运算对应的物理电路和控制线路也就简单。

二进制加法规则为

$$0+0=0$$

$$0+1=1+0=1$$

$$1+1=10$$

二进制的乘法规则为

$0\times0=0$

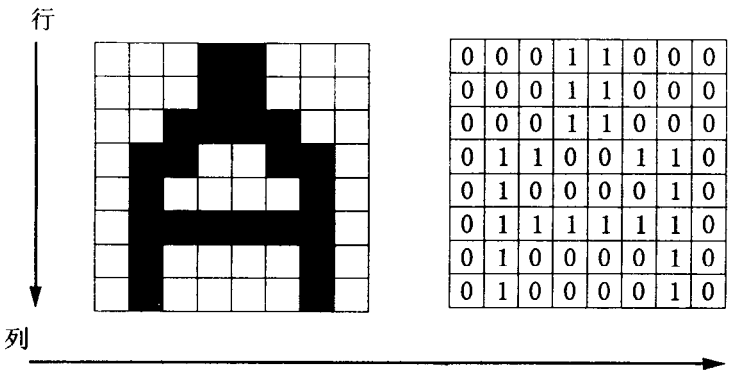
$0\times1=1\times0=0$

$1\times1=1$

4. 节省设备

在任何一个系统中采用什么数制，直接影响到所需配置的各种设备。不适当的数制将使设备量大大增加。从理论上说，e 进制最节约硬件设备投资，二进制次之。由于 e 进制对应的物理状态不容易在技术上实现，所以至今未能付诸实用。而二进制无论从数学物理模型还是大规模的技术实现都已完全成熟，所以说实际上二进制的设备最省，也最切实可行。

另外二进制数可以表示几乎所有的信息，如图 1-2-2 所示。



0	0	0	1	1	0	0	0
0	0	0	1	1	0	0	0
0	0	0	1	1	0	0	0
0	1	1	0	0	1	1	0
0	1	0	0	0	0	1	0
0	1	1	1	1	1	1	0
0	1	0	0	0	0	1	0
0	1	0	0	0	0	1	0

图 1-2-2　二进制数信息图

从图中可以看出单位面积内只要用足够的二进制位组合，便能表示出文字、图像、图形、色彩等任何信息。从图中还能看出：按照图中行、列的关系可以方便地进行高速运算、传送、存储等等一系列操作。

但是，这些都是在电路内部进行，在图 1-2-1 中所示的任何形式的信息数据最终的结果还将转换成自然信息方式提供给使用者。

下面首先讨论各种不同的进位计数制之间的转换方法；然后讨论计算机中进行二进制运算时所要用到的原码、补码、反码的表示方法，最后讨论各种常用的几种编码。

一、十进制计数

十进制是最常用的记数表示方式。我们研究的目的是要找出这种计数方式最基本的特征和它们的数学模型，以便用硬件或软件编程的方式来实现其运算功能、转换功能、存储方式。

记数符号为：0，1，2，3，4，5，6，7，8，9（该进位制中可能用到的全部数码的个数为 10）

记数基数为：10

进位方式：逢十进一

十进制计数的表示方法有以下几种形式：

1. 位置表示法

位置表示法是由数码所在位置来确定数的大小。

例如：数 1 9 9 8 . 5 2 3

从左到右分别表示：

千位，百位，十位，个位，十分之一位，百分之一位，千分之一位

1　9　9　8　5　2　3

$$(N)_{10}=(K_{n-1}K_{n-2}\cdots K_1K_0\cdot K_{-1}K_{-2}\cdots K_{-m})_{10}$$

2. 多项式表示法（按权展开式）

例如：数 1998.523 表示为：

$$1\times10^3+9\times10^2+9\times10^1+8\times10^0+5\times10^{-1}+2\times10^{-2}+3\times10^{-3}$$

$$(N)_{10}=K_{n-1}(10)^{n-1}+K_{n-2}(10)^{n-2}+\cdots+K_1(10)^1+K_0(10)^0+K_{-1}(10)^{-1}+K_{-2}(10)^{-2}+\cdots+K_{-m}(10)^{-m}$$

3. 和式表示法

$$(N)_{10}=\sum_{i=-m}^{n-1}K_i(10)^i$$

式中　K_i 是 0，1，2，3，…，9 中的任意一个数码；

m 是正整数，表示小数的位数；

n 是正整数，表示整数的位数；

$(10)^i$ 是第 i 位系数 K_i 的“权”，等号右边括号中的 10 表示进位制基数。

二、二进制记数

记数符号为：0，1（该进位制中可能用到的全部数码的个数为 2）

记数基数为：2

进位方式：逢二进一

二进制记数的表示方法有三种形式：

1. 位置表示法：由数码所在位置来确定数的大小。

例如：数 111011.101

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

2. 多项式表示法（按权展开式）

例如：数 111011.101 表示为：

$$1\times2^5+1\times2^4+1\times2^3+0\times2^2+1\times2^1+1\times2^0+1\times2^{-1}+0\times2^{-2}+1\times2^{-3}$$

$$(N)_2=K_{n-1}(2)^{n-1}+K^{n-2}(2)^{n-2}+\cdots+K_1(2)^1+K_0(2)^0+K_{-1}(2)^{-1}+K_{-2}(2)^{-2}+\cdots+K_{-m}(2)^{-m}$$

3. 和式表示法

$$(N)_2=\sum_{i=-m}^{n-1}K_i(2)^i$$

式中　K_i 是 0，1 中的任意一个数码；

m 是正整数，表示小数的位数；

n 是正整数，表示整数的位数；

$(2)^i$ 是第 i 位系数 K_i 的“权”，等号右边括号中的 2 表示进位制基数。

三、任意进制记数

记数符号为：0，1，j−1（该进位制中可能用到的全部数码的个数为 j）

记数基数为：j

进位方式：逢 j 进一

任意进制记数的表示方法有三种形式：

1. 位置表示法

$$(N)_j = (K_{n-1}K_{n-2}\cdots K_1K_0.K_{-1}K_{-2}\cdots K_{-m})_j$$

2. 多项式表示法（按权展开式）：

$$(N)_j = K_{n-1}(j)^{n-1} + K_{n-2}(j)^{n-2} + \cdots + K_1(j)^1 + K_0(j)^0 + K_{-1}(j)^{-1} + K_{-2}(j)^{-2} + \cdots + K_{-m}(j)^{-m}$$

3. 和式表示法：

$$(N)_j = \sum_{i=-m}^{n-1} K_i(j)^i$$

式中 K_i 是 0，1…j 中的任意一个数码；

m 是正整数，表示小数的位数；

n 是正整数，表示整数的位数；

$(j)^i$ 是第 i 位系数 K_i 的“权”，等号右边括号中的 j 表示进位制基数。

当 j＝2，就是二进制；

当 j＝8，就是八进制；

当 j＝10，就是十进制；

当 j＝16，就是十六进制，十六进制数有十六个不同的数码，除了 0～9 这十个数码外，还增加了 A，B，C，D，E，F 分别与十进制中的 10，11，12，13，14，15 相对应。

表 1-2-1 列出了几种常用进位制整数数列前面的一部分数码。

表 1-2-1 部分整数数列

j=10	j=2	j=4	j=8	j=16
0	0	0	0	0
1	1	1	1	1
2	10	2	2	2
3	11	3	3	3
4	100	10	4	4
5	101	11	5	5
6	110	12	6	6
7	111	13	7	7
8	1000	20	10	8
9	1001	21	11	9
10	1010	22	12	A
11	1011	23	13	B
12	1100	30	14	C
13	1101	31	15	D
14	1110	32	16	E
15	1111	33	17	F
16	10000	100	20	10
17	10001	101	21	11

从以上分析，可以看出不同的进位制具有的共同点如下：

每一个进位制数都有一个固定的基数 j，它们的每一个数位可能取 j 个不同的数码，而且是逢 j 进一的，即每一位计满 j 就向高位进一。

进位制数都能写成几种不同形式的记数方式，它的每一个数码 K 都对应于一个固定的“权”位，相应的展开式也称为进位制数按权展开式，权的高低按从左到右顺序分配。若一个数 $(N)_j$ 既有整数又有小数时，当小数点向左移一位，则等于原数缩小了 j 倍，即乘以 1/j，当小数点向右移一位时，则等于原数增大了 j 倍，即乘以 j 倍。

表示同样的数值，二进制需要的位数较多，虽然机器处理起来速度很高，但是读写起来很不方便，所以常用到八进制、十六进制数来表示，以缩短表示的数据长度，因为八进制、十六进制数都是二进制 2^N 的倍数，相互之间的换算十分方便。

在计算机的人机接口输入、输出时，同样也需要将机器语言的二进制数转换成十进制数。以下讨论各种数制之间的转换方法。

四、数制转换

1. 二进制与十进制数之间的转换

把二进制数转换成十进制数，只需把二进制数中 K＝1 的那些数位的权相加求和即可。

例如：

$$\begin{aligned}N_2 &=(1\,1\,1\,0\,1\,1.1\,0\,1)_2\\ &=1\times 2^5+1\times 2^4+1\times 2^3+0\times 2^2+1\times 2^1+1\times 2^0\\ &\quad +1\times 2^{-1}+0\times 2^{-2}+1\times 2^{-3}\\ &=32+16+8+0+2+1+0.5+0+0.125\\ &=59.625\end{aligned}$$

把二进制数中 K＝1 数位的权全部列出，包括整数和小数，然后相加得出相应的十进制数。将二进制数转换成十进制数，其实质就是求 2 的幂之和。

对于位数较多的数可用计算机按它的数学模型编程加以实现，对整数和小数分别用逐位乘 2 或乘 2^{-1} 相加的方法实现转换（连乘法）。

逐位乘 2 相加适用于整数：

$$\begin{aligned}(111011)_2 &=((((1\times 2+1)\times 2+1)\times 2+0)\times 2+1)\times 2+1\\ &=(59)_{10}\end{aligned}$$

只要由高到低，逐位乘 2 相加，最低位不需再乘 2，因为该位的权是 $2^0=1$。

逐位乘 2^{-1} 相加适用于小数：

$$\begin{aligned}(.101)_2 &=(((1\times 2^{-1}+0)\times 2^{-1}+1)\times 2^{-1}\\ &=(0.625)_{10}\end{aligned}$$

用连乘法就是分别在整数和小数部分逐位求出对应的权并相加之。如 $(111011)_2$ 中的最高位 1 的权是 2^5，在连乘法中该位的 1 连乘了五个 2，从而形成了该位的权 2^5，$(.101)_2$ 中最低位 1 的权是 2^{-3}，在连乘法中，该位的 1 连乘了三个 2^{-1}，从而形成了该位的权 2^{-3}。这样可以用权位的幂作为软件编程的循环控制变量，而实现编程转换。

2. 十进制与二进制数之间的转换

与二进制转换成十进制相对应，十进制数也可以通过逐位连除分解相应的二进制数的数码，从十进制数中由高到低提取 2 的幂，也就是从高到低分解出二进制数各位的权，确定二进制数各权位的数 K 值取 0 还是 1，实现十进制到二进制的转换。

例如：将十进制数 N＝625.625 转换成二进制数。

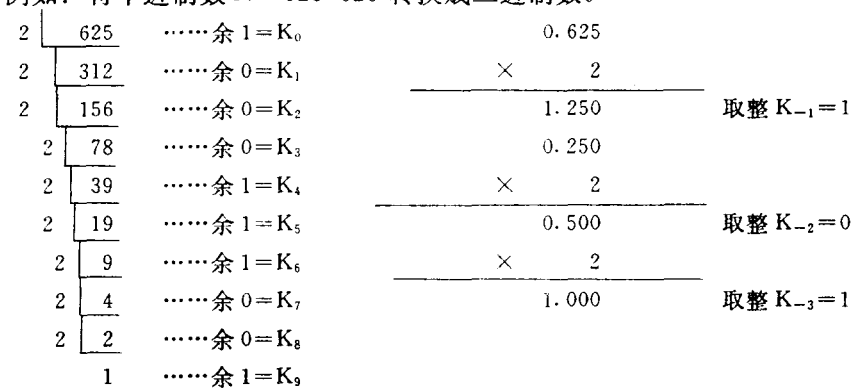


图 1-2-3　十进制转二进制整数短除取余小数取整

一个二进制数多项式表示方法为：

$$(N)_2=K_{n-1}(2)^{n-1}+K_{n-2}(2)^{n-2}+\cdots+K_1(2)^1+K_0(2)^0$$
$$+K_{-1}(2)^{-1}+K_{-2}(2)^{-2}+\cdots+K_{-m}(2)^{-m}$$

其中，整数部分的权位提取通过不断地除 2 取余确定从 K_0 到 K_{N-1}的值。

$K_9=1$，$K_8=0$，$K_7=0$，$K_6=1$，$K_5=1$，$K_4=1$，$K_3=0$，$K_2=0$，$K_1=0$，$K_0=1$

其中，小数部分的权位提取通过不断地除 2^{-1}即乘以 2 取整确定从 K_{-1}到 K_{-m}的值。

$$K_{-1}=1, K_{-2}=0, K_{-3}=1$$

则：

$$\begin{aligned}625.625&=1(2)^9+0(2)^8+0(2)^7+1(2)^6+1(2)^5+1(2)^4+0(2)^3+0(2)^2+0(2)^1\\&\quad+1(2)^0+1(2)^{-1}+0(2)^{-2}+1(2)^{-3}\\&=512+0+0+64+32+16+0+0+0+1+0.5+0+0.125\\&=(1001110001.101)_2\end{aligned}$$

在十进制小数转换成二进制小数时，整个过程可能无限制地进行下去，对于计算机来说，数的运算精度取决于处理器的字长，所以无限循环的小数对实际的机器运算处理没有实际意义。一般以计算机字长的为限制，只取机器字长位数的近似值。

3. 二进制数与基数为 2 的倍数 2^j 进制数之间的转换

二进制作为机器语言运行速度很快，但是二进制由于位数较多，实际读写表示时不方便，很容易出现错误，因此常常使用其 2 的倍数 2^j 进制数来表示、运算、处理。二进制数转换成基数为 2^j 进制数，首先要根据 j 的值进行分组，方法为：

以小数点为中心，整数部分从右到左按 j 的数值进行分组，最后不满一组的数用高位补零的方法补足一组。

对小数部分的分组，以小数点为起点从左向右按 j 的数值进行分组，最低位不足的一组的数位用尾数补零的方法补足一组。

举例说明：

例如：将二进制数转换成八进制数

因为八进制数的 j＝3，其过程为：先将二进制数以小数点为中心，按三位进行分组（高位不足一组补零，低位不足一组补零，见有下横线的零），再写出与各组对应的数码。

$$11101101001001101.1101011 = \underline{0}11\ 101\ 101\ 001\ 001\ 101.110\ 101\ 1\underline{00}$$
$$= (355115.654)_8$$

例如：将二进制数转换成十六进制数

其过程为：先将二进制数以小数点为中心，按四位进行分组，再写出与各组对应的数码。

$$11101101001001101.1101011 = \underline{0001}\ 1101\ 1010\ 0100\ 1101.1101\ 011\underline{0}$$
$$= (1DA4D.D6)_{16}$$

4. 基数为 2 的倍数 2^j 进制数与二进制数之间的转换

转换方法与上述过程相反，可以直接将 2^j 进制数写成对应位数的二进制数。

例如：将十六进制数转换成二进制数，每位十六进制数用四位二进制数替换即可。

1	D	A	4	D	.	D	6
0001	1101	1010	0100	1101	.	1101	0110

例如：将八进制数转换成二进制数，每位八进制数用三位二进制数替换即可。

3	5	5	1	1	5	.	6	5	4
011	101	101	001	001	101	.	110	101	100

以上的转换是可逆的。二进制可沟通八进制和十六进制之间的转换。在二、八、十和十六进制之间可以相互利用，以最简捷的方法进行转换。

例如：将二进制数（110101101）转换为十进制数，可以先将二进制数转换为八进制数，再用按权相加方法获得十进制数：

$$(110101101)_2 = (110\ 101\ 101)_2$$
$$= (655)_8$$
$$= 6 \times 8^2 + 5 \times 8^1 + 5 \times 8^0$$
$$= 6 \times 64 + 5 \times 8 + 5$$
$$= (429)_{10}$$

例如：将二进制数转换为十进制数，也可先将二进制数转换为十六进制数，再用按权相加方法获得十进制数：

$$(110101101)_2 = (1\ 1010\ 1101)_2$$
$$= (1AD)_{16}$$
$$= 1 \times 16^2 + A \times 16^1 + D \times 16^0$$
$$= 1 \times 256 + 10 \times 16 + 13$$
$$= (429)_{10}$$

例如：将十六进制数转换成八进制数，可先将十六进制转换成二进制数，再将二进制数转换为八进制数：

$$(DAE936)_{16} = (1101\ 1010\ 1110\ 1001\ 0011\ 0110)_2$$
$$= (110\ 110\ 101\ 110\ 100\ 100\ 110\ 110)_2$$

$=(66564466)_8$

八进制和十六进制与二进制数相比书写简短，易辨读，与二进制数之间的转换方便，所以在程序设计、指令书写、数据地址分配中十六进制用得特别广泛。另外，在数字系统和计算机应用中，由于受到计算机字长和存储器基本字节的影响，十六进制数的应用最为频繁，按字节（8 个二进制位）存储器的一个单元可以存放两个十六进制数，计算机的中央处理器单元（CPU）字长都是按 2^N 的规则递增，如：8 位，16 位，32 位，64 位等等。在通信系统中数据传送通常也是按 2^N 规律变化。所以，二－十六进制转换应熟练掌握。

§1-3 编　　码

编码是用各种数字、文字、图形、图像、符号及不同数码表示对应的某个具体信息状态，在数字系统和计算机系统中是以二进制数码来表示所有的信息状态，将每个二进制码赋予特定含义的过程被称为编码。

一、机器数的原码、补码和反码

为了表示数据的符号，即二进制的正负数，把符号和数值统一用由 0、1 两个数字符号表示，下面介绍在机器中有符号数的各种表示形式。

1. 机器数与真值

和十进制一样，二进制中的符号位也是置于所有数值位之前。规定用“＋”表示正数并且可以省略，用“－”表示负数。

例如　　N＝＋1101000　　M＝－1101000

在机器中通常用一位符号加数值部分来表示一个完整的数，用加符号“0”表示正数，加符号“1”表示负数。

在机器中由于存储器和运算器的限制，通常有符号数都是以字节方式存储和运算，所以有符号数通常是 1 位符号位＋7 位数值位，或 1 位符号位＋15 位数值位（2 个字节）等这种排列方式来表示一个有符号数。

例如　　N＝［01101000］　　M＝［11101000］

在机器中数值与符号已经全部被数码化了，把符号位和数值位一起编码来表示对应的二进制数，把机器中以代码形式出现的数称为机器数，而以原来一般书写形式表示的数称为机器数的真值。

为了使得编码的表示形式能便于机器进行运算和处理，又可将机器数表示为原码、补码及反码的形式。

2. 原码表示法

原码是机器数的简单表示法。符号位是最高位，用 0 表示正数，用 1 表示负数，数值部分用真值表示。

正数的原码：数值位前面加“0”表示的正号。

负数的原码：数值位前面加“1”表示的负号。

例如：　　X＝＋1011010

　　Y＝－1011010

X，Y 的原码表示为：

$[X]_{原}=01011010$

$[Y]_{原}=11011010$

当 X 为正时，$[X]_{原}$ 和 X 数值的区别是增加了一位符号位，通常在最高位增加 0 位对数值的大小没有影响，所以 $[X]_{原}$ 等于 X 本身。

当表示数 Y 为负值时，原码和原数的区别增加了一位 1 作符号位，所以原码不等于原数。

在机器中做算术运算时，除了数值部分进行运算，另外符号位也要进行运算。若是乘、除运算，符号按异或方式进行运算（即 $0\oplus0=0$，$0\oplus1=1$，$1\oplus0=1$，$1\oplus1=0$）确定，若是做加、减运算则需要判断两数的符号，相加时如是同号，则数值相加，符号不变；如是异号，数值部分相减，还需比较两数绝对值的大小，用大数减小数，差的符号和绝对值大的数相同。这通常在软件编程时用比较判断来实现。其数字模型为：

当 $0\leqslant X<2^{n-1}$ 时　　$[X]_{原}=X$

当 $-2^{n-1}<X\leqslant 0$ 时　　$[X]_{原}=2^{n-1}-X$

例如：数 $X=-1011010$

则原码为：

$$\begin{aligned}[X]_{原}&=11011010\\&=10000000+1011010\\&=10000000-(-1011010)\\&=2^{8-1}-X\end{aligned}$$

为了判断两数是同号还是异号，以及比较两数绝对值的大小，必须增加机器的计算时间，这样降低了计算速度。为此寻求新的机器数表示方法。

3. 反码表示法

反码也是符号位数值化后的机器数表示方法。符号位的表示方法和含义与原码相同。正数和负数的反码按如下规则组成。

正数的反码：与原码相同。

负数的反码：符号位为 1 不变，将数值位的各位取反。

$[-1000100]_{反}=10111100$

$[-1111001]_{反}=10000110$

例如：　$X=+1011010$　　$[X]_{原}=01011010$　　$[X]_{反}=01011010$

　　$X=-1011010$　　$[X]_{原}=11011010$　　$[X]_{反}=10100101$

数字模型为：

当 $0\leqslant X<2^{n-1}$ 时　　$[X]_{反}=X$

当 $-2^{n-1}<X\leqslant 0$ 时　　$[X]_{反}=(2^n-1)+X$

例如：$X=-1011010$

则反码为：

$$\begin{aligned}[X]_{反}&=(2^n-1)+(-1011010)\\&=(2^8-1)-1011010\\&=11111111-1011010\end{aligned}$$

$=10100101$

反码中的特点有：

（1）0 的表示方法不唯一

$[+0]_{反}=00000000$

$[-0]_{反}=11111111$

（2）两个反码运算时，当符号位有进位时，必须循环进位，即将符号位产生的进位加到最低位。

由于反码循环进位也给机器运算增加麻烦，为此提出了补码。

4．补码表示法

补码是机器数的另一种表示形式。引进补码是为了将加减法运算都用加法实现，即用求和的方式来求差，数的符号也作为数值处理一起参加运算，这为机器的运算处理带来很大方便，在现代计算机中均采用补码运算而使 CPU 的设计简单化。

补码的基本原型是求模取余运算，即对于两个整数 X、Y，用某一正数 M 去同除这两个整数 X、Y，若所得的余数相同，则称 X、Y 对模数 M 是同余。记为：

$$X=Y\ (\mathrm{MOD}\ M)$$

最简单的例子是钟表，设 X＝1，Y＝13，模数 M＝12，则

$$X=Y(\mathrm{MOD}\ M)$$

$$1=13(\mathrm{MOD}\ 12)$$

也可记为：

$$X+KM=X\quad(\mathrm{MOD}\ M)\quad K=0,1,2\cdots$$

所以对钟表 12 点来说：

$$-1+12=11=-1(\mathrm{MOD}\ 12)$$

其含义是 12 点差一小时等于 11 点，则 11 和－1 模 12 同余。

对于 M＝10 为模的十进制数：

$$-1+10=9=-1\ (\mathrm{MOD}\ 10)$$

$$-2+10=8=-2\ (\mathrm{MOD}\ 10)$$

这里 9 是－1 的补码，8 是－2 的补码……，1 是－9 的补码。在机器中做减法运算时，可以用补码把减法运算转换成加法运算来完成。

例如：设 X＝9，Y＝－3，X－Y

$$F=9-3$$

$$=9+(-3)=9+7=\boxed{1}6\ (\mathrm{MOD}\ 10)$$

其中进位$\boxed{1}$丢弃，结果相同。

在数字电路和计算机中数是以二进制来表示，并且都有一定的长度，若设长度为 n 位，则它的模数就是 2^n。因此补码 $[X]_{补}$ 的数字模型为：

当 $0\leqslant X<2^{n-1}$ 时　　$[X]_{补}=X$

当 $-2^{n-1}<X\leqslant 0$ 时　　$[X]_{补}=(2^n)+X$

即 $[X]_{补}=2^n+X\ (\mathrm{MOD}\ 2^n)$

例如：X＝＋1011010