SHUZI DIANZI JISHU

数字电子技术

江晓安 编

西安电子科技大学出版社

数字电子技术

江晓安 编

西安电子科技大学出版社
2000

（陕）新登字 010 号

内 容 简 介

本书内容包括：数制与编码；基本逻辑运算及集成逻辑门；布尔代数与逻辑函数化简；组合逻辑电路；触发器；同步时序电路分析与设计；常用时序逻辑部件；脉冲产生电路和定时电路；数/模、模/数转换电路；大规模集成电路；可编程逻辑器件 PLD 共十一章。每章均有例题和练习题。本书与《模拟电子技术》配套使用。教学时数 60 学时左右（不含实验）。

编者积 30 多年的教学经验，综合有关专业的大纲要求，写出适应面较宽的教材。本教材适用于高等工科院校有关专业本科、专科、自学考试、夜大、函大、职大学生，也可供从事电子技术方面工作的工程技术人员学习参考。

数字电子技术

江晓安　编

责任编辑　王绍菊

西安电子科技大学出版社出版发行

西安兰翔印刷厂印刷

新华书店经销

开本 787×1092　1/16　印张 18.5　字数 438 千字

1993 年 8 月第 1 版　2000 年 4 月第 9 次印刷　印数 68 001～76 000

ISBN 7-5606-0231-2/TN・0071　　定价：14.90 元

前　言

电子技术分为模拟电子技术和数字电子技术，数字电子技术是当前发展最快的学科之一。就逻辑器件而言，已经从40年代的电子管、50年代的晶体管、60年代的小规模集成电路（SSI），发展到现在的中规模集成电路（MSI）、大规模集成电路（LSI）和超大规模集成电路（VLSI）。近几年又出现了可编程逻辑器件，为数字电路设计，提供更加完善、方便的器件。相应地，数字电路的设计过程和方法也在不断地演变和发展。由于半导体技术的迅速发展，微型计算机的广泛应用，所以数字电子技术在现代科学技术领域中占有很重要的地位，在各个领域中得到广泛的应用。

本书共分十一章。第一章讲解了数字电路中所用数制和编码；第二章讲述了TTL和MOS门电路；第三章是全书的学习基础，讲述了数字技术的数学基础——布尔代数及逻辑函数的化简；第四章讲述组合逻辑电路的分析与设计，重点介绍常用组合逻辑部件（MSI）的原理和应用；第五章为触发器，它是学习时序电路的基础；第六章为同步时序电路的分析与设计；第七章介绍常用的时序逻辑部件，主要讲述了计数器和移位寄存器的设计、分析及集成计数器、移位寄存器的应用；第八章讲述脉冲产生电路和定时电路，主要讲述555定时电路及其应用；第九章为数/模、模/数转换电路；第十章介绍了大规模集成电路ROM和PLA在数字电路中的应用；第十一章介绍可编程逻辑器件（PLD）。

在内容选取和安排上，编写时突出基本概念，基本理论和基本方法。并为读者提供独立分析和设计逻辑电路的工具，即主要讲述分析和设计的方法，不追求系统性和完整性。如逻辑函数的标准式，只讲述常用的最小项标准式，而最大项的概念就不引入；时序电路，我们只讲述同步时序电路的分析与设计，而异步时序电路的分析与设计就不介绍。为了适应科学技术的发展，本书除了讲述传统的逻辑技术外，还用较多的篇幅讲述了MSI和LSI在数字技术中的应用。

为便于读者自学，着重讲清思路，交待方法，并附有例题和练习题，文字上力求叙述流畅，说理清楚。同时也编写了《数字电子技术学习指导书》，配合本教材使用。

本教材适合高等工业学校有关专业本科、专科生《数字电子技术》课程的教材，也适用于自学考试、夜大、函大、职大的学生选用。

本书的出版得到西安电子科技大学王和平同志、吕建伟同志、付长进同志的支持和帮助，在此表示谢意。

由于编者水平有限，时间又较紧，书中一定有不少错误和不妥之处，欢迎读者指正。

编　者

1993年3月

目　录

前言

第一章　数制与编码

§1　进位计数制 …… 1
一、十进制 …… 1
二、二进制 …… 1
三、八进制和十六进制 …… 2
§2　数制转换 …… 3
一、其它进制与十进制数相互转换 …… 3
二、二进制与八进制、十六进制的相互转换 …… 6
§3　二进制数的算术运算 …… 7
一、加法运算 …… 7
二、减法运算 …… 7
三、乘法运算 …… 7
四、除法运算 …… 8
§4　数的原码、反码及补码表示 …… 8
一、机器数与真值 …… 8
二、原码、反码及补码 …… 8
三、原码、反码及补码的算术运算 …… 10
四、溢出的概念及补码运算中溢出的判断 …… 12
§5　编码 …… 13
一、二—十进制(BCD)码 …… 13
二、奇偶校验码 …… 16
三、字符码 …… 17
练习题 …… 19

第二章　基本逻辑运算及集成逻辑门

§1　基本概念 …… 21
一、逻辑变量与逻辑函数 …… 21
二、真值表 …… 22
§2　三种基本逻辑运算 …… 22
一、逻辑乘("与"运算)—— AND …… 22
二、逻辑加("或"运算)—— OR …… 23
三、逻辑非(NOT) …… 24
§3　常用的复合逻辑 …… 25
一、"与非"逻辑 …… 25
二、"或非"逻辑 …… 26
三、"与或非"逻辑 …… 26
四、"异或"逻辑及"同或"逻辑 …… 26
五、正负逻辑 …… 27
§4　集成逻辑门 …… 28
一、TTL 与非门 …… 29
二、ECL 电路和 I^2L 电路 …… 42
三、MOS 集成逻辑门 …… 48
练习题 …… 54

第三章　布尔代数与逻辑函数化简

§1　基本公式和规则 …… 58
一、基本公式 …… 58
二、基本法则 …… 60
三、几个常用公式 …… 61
§2　逻辑函数的代数法化简 …… 62
一、逻辑函数的"与或"式和"或与"式 …… 62
二、逻辑函数与逻辑图 …… 62
三、逻辑函数化简的原则 …… 63
四、逻辑函数的形式和逻辑变换 …… 63
五、与或逻辑函数的化简 …… 64
§3　卡诺图化简 …… 67
一、卡诺图化简的基本原理 …… 67
二、逻辑函数的标准式——最小项 …… 67
三、卡诺图结构 …… 69
四、逻辑函数的卡诺图表示法 …… 70
五、最小项合并规律 …… 71
六、与或逻辑化简 …… 71
七、其它逻辑形式的化简 …… 74
八、无关项及无关项的应用 …… 77
九、输入只有原变量没有反变量的逻辑函数化简 …… 79
十、多输出函数的化简 …… 85
练习题 …… 86

第四章　组合逻辑电路

§1　组合逻辑电路的分析 …… 89
§2　组合逻辑电路的设计 …… 91
§3　常用组合逻辑 …… 93
一、半加器与全加器 …… 94
二、编码器与译码器 …… 99

三、数据选择器及多路分配器…… 112
四、数字比较器与检验电路…… 119
§4 组合电路中的竞争与冒险 …… 123
一、竞争现象…… 123
二、冒险现象…… 124
三、冒险现象的判别…… 124
四、冒险现象的消除…… 125
练习题…… 127
第五章 触发器
§1 时序电路概述 …… 131
一、时序电路特点…… 131
二、时序电路分类…… 131
三、状态表和状态图…… 132
§2 基本触发器 …… 135
一、基本 R－S 触发器 …… 135
二、时钟控制的 R－S 触发器 …… 137
三、D 触发器…… 139
四、T 触发器…… 140
五、JK 触发器 …… 141
六、基本触发器的空翻和振荡现象…… 142
§3 集成触发器 …… 144
一、主从触发器…… 144
二、维持阻塞触发器…… 147
三、边沿触发器…… 148
四、CMOS 触发器…… 150
五、触发器的相互转换…… 153
练习题…… 155
第六章 同步时序电路的分析与设计
§1 同步时序电路的分析方法 …… 159
一、分析步骤…… 159
二、分析举例…… 159
§2 同步时序电路的设计 …… 162
一、建立原始状态图…… 162
二、状态化简…… 165
三、状态分配…… 170
四、设计举例…… 171
练习题…… 176
第七章 常用时序逻辑部件
§1 计数器 …… 178
一、同步计数器的分析与设计…… 178
二、异步计数器的分析与设计…… 187
三、集成计数器功能分析及其应用…… 192
§2 寄存器与移位寄存器 …… 205
一、寄存器…… 205
二、移位寄存器…… 206
三、集成移位寄存器…… 208
§3 序列信号发生器 …… 219
一、移位型序列信号发生器…… 219
二、计数型序列信号发生器…… 221
练习题…… 223
第八章 脉冲产生电路和定时电路
§1 概述 …… 227
§2 555 定时电路 …… 227
一、555 定时电路的基本组成 …… 228
§3 单稳态电路 …… 229
一、电路组成…… 229
二、工作原理…… 229
§4 多谐振荡器 …… 231
一、电路组成…… 232
二、工作原理…… 232
§5 施密特电路 …… 234
一、电路组成…… 234
二、工作原理…… 234
练习题…… 235
第九章 数模、模数转换电路
§1 DAC …… 237
一、DAC 的基本概念 …… 237
二、DAC 的电路形式及工作原理 …… 239
三、DAC 电路中的模拟开关 …… 243
§2 ADC …… 244
一、ADC 的组成 …… 244
二、ADC 电路 …… 246
练习题…… 252
第十章 大规模集成电路
§1 ROM 及其应用…… 253
一、ROM 的原理与实现方法 …… 253
二、ROM 在组合逻辑设计中的应用 …… 254
三、可编程只读存贮器…… 259
四、ROM 容量的扩展 …… 259
§2 PLA 及其应用 …… 261
练习题…… 270
第十一章 可编程逻辑器件 PLD
§1 PLD 器件的基本结构 …… 272
一、PROM 的基本结构 …… 273

二、FPLA 的基本结构 …… 273
三、PAL 的基本结构 …… 274
四、GAL 的基本结构 …… 275
§2 PLD 器件的应用举例 …… 278
参考文献 …… 285

第一章 数制与编码

数字设备及计算机存在两种不同类型的运算，逻辑运算和算术运算。逻辑运算实际上是实现某种控制功能，而算术运算是对数据进行加工。算术运算的对象是数据，因此对数的基本特征和性质应有所了解。同时，数字设备中，是采用二进制数，因而，在数字设备中表示的数、字母、符号等等都要以特定的二进制码来表示——这就是二进制编码。所以这一章将对数的一些基本知识进行介绍，同时还将介绍一些常用的编码。

§1 进位计数制

目前计数通常是采用进位计数法。进位计数法是将数划分为不同的数位，按位进行累计，累计到一定数量之后，又从零开始，同时向高位进位。由于位数不同，则同样的数码在不同的数位中所表示的数值是不同的，低位数值小，高位数值大。进位计数法使用较少的数码就能表示较大的数。

在生产实践中人们使用各种进位制，如10、12、16、60等进制，但人们最熟悉的是十进制。在数字设备中，机器只认识二进制，由于二进制书写长，所以在数字设备中又常采用八进制和十六进制。

每个数位规定使用的数码符号的总数，称为进位基数，又称进位模数，用 R 表示。若每位数码用 a_i 表示，n 为整数的位数，m 为小数的位数，则进位计数制表示数的式子为

$$N = a_{n-1}\,a_{n-2}\cdots a_i\cdots a_1\,a_0\,a_{-1}\,a_{-2}\cdots a_{-m} \tag{1.1}$$

当某位的数码为1时所表征的数值，称为该数位的权值。权值随数位的增加呈指数规律增加，最低位的权值为1，第 i 位的权值为 R^i。这样，第 i 位数码 a_i 所表示的绝对值就是数码 a_i 乘上该位数的权值，即 $a_i \times R^i$。(1.1) 式可写成下述按权展开式

$$N = a_{n-1}R^{n-1} + \cdots + a_iR^i + \cdots + a_0R^0 + a_{-1}R^{-1} + \cdots + a_{-m}R^{-m} \tag{1.2}$$

该式对任何进位制均是适用的。

一、十进制

十进制是人们最熟悉的一种数制，它的进位规则是“逢十进一”。每位数码用下列十个符号之一表示，即0，1，2，3，4，5，6，7，8，9。

例如一个多位十进制数为

$$N = (1989.524)_D$$

下标D表示十进制数。根据位权的概念写出按权展开式：

$$N = 1 \times 10^3 + 9 \times 10^2 + 8 \times 10^1 + 9 \times 10^0 + 5 \times 10^{-1} + 2 \times 10^{-2} + 4 \times 10^{-3}$$

二、二进制

二进制是目前数字设备、计算机采用的数制。它的进位规则是“逢二进一”，每位数码只有下列两个符号：0，1。

一个多位二进制数表示如下：

$$N = (1101.01)_B$$

下标 B 表示为二进制。其按权展开式为

$$N = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2}$$

为便于理解和熟悉二进制，下面列出几个十进制数和二进制数的关系式：

$$(1001)_B = (9)_D$$

$$(111)_B = (7)_D$$

$$(1101.01)_B = (13.25)_D$$

二进制书写起来太长，故在数字设备和计算机中，常采用八进制或十六进制，它可有效地缩短字长。因 $8 = 2^3$，$16 = 2^4$ 故一位八进制数相当于三位二进制数，一位十六进制数相当于四位二进制数，这样就分别将字长缩短 3 倍和 4 倍。

三、八进制和十六进制

八进制的进位规则是“逢八进一”，每位数码用下列八个符号之一表示：0，1，2，3，4，5，6，7。

一个多位八进制数表示如下：

$$N = (367.42)_O$$

下标 O 表示为八进制。其按权展开式为

$$N = 3 \times 8^2 + 6 \times 8^1 + 7 \times 8^0 + 4 \times 8^{-1} + 2 \times 8^{-2}$$

十六进制的进位规则是“逢十六进一”，每位数码用下列 16 个符号之一表示：0，1，2，3，4，5，6，7，8，9，A，B，C，D，E，F。

一个多位十六进制数表示如下：

$$N = (19AF.EB)_H$$

下标 H 表示为十六进制。其按权展开式为

$$N = 1 \times 16^3 + 9 \times 16^2 + A \times 16^1 + F \times 16^0 + E \times 16^{-1} + B \times 16^{-2}$$

为便于比较，表 1-1 列出不同数制的对照关系。

由于二进制机器实现起来十分容易，而十进制人们熟悉，八进制和十六进制可压缩字长。因此，这几种数制都会用到，必然会遇到不同数制之间的转换问题。

表 1-1　几种进位制对照表

十进制数	二进制数	八进制数	十六进制数
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5

十进制数	二进制数	八进制数	十六进制数
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	*A*
11	1011	13	*B*
12	1100	14	*C*
13	1101	15	*D*
14	1110	16	*E*
15	1111	17	*F*
16	10000	20	10
17	10001	21	11
20	10100	24	14
32	100000	40	20
100	1100100	144	64

§2 数制转换

一、其它进制与十进制数相互转换

其它进制数转换为十进制数用加权法，即将其它进制数写成按权展开式，然后各项相加，则得相应的十进制数。

例 1 $N=(1011.011)_B=(?)_D$

按权展开

$$\begin{aligned}N&=1\times2^3+0\times2^2+1\times2^1+1\times2^0+0\times2^{-1}+1\times1\times2^{-2}+1\times2^{-3}\\&=8+2+1+0.25+0.125\\&=(11.375)_D\end{aligned}$$

今后数码为 0 的那些项可以不写。

例 2 $N=(101011.1001)_B=(?)_D$

按权展开

$$\begin{aligned}N&=1\times2^5+1\times2^3+1\times2^1+1\times2^0+1\times2^{-1}+1\times2^{-4}\\&=32+8+2+1+0.5+0.0625\\&=(43.5625)_D\end{aligned}$$

例 3 $N=(153.07)_O=(?)_D$

$$N=1\times8^2+5\times8^1+3\times8^0+7\times8^{-2}$$
$$=64+40+3+0.109$$
$$=(107.109)_D$$

例 4　$N=(E93.A)_H$

$$N=14\times16^2+9\times16^1+3\times16^0+10\times16^{-1}$$
$$=3584+144+3+0.625$$
$$=(3731.625)_D$$

十进制数转换为其它进制数，分为整数和小数两部分，它们的转换方法不同。

整数转换，采用基数除法，即将待转换的十进制数除以将转换为新进位制的基数，取其余数，其步骤如下：

(1) 将待转换十进制数除以新进位制基数 R，使其余数作为新进位制数的最低位(LSB)；

(2) 将前步所得之商再除以新进位制基数 R，记下余数，作为新进位制数的次低位；

(3) 重复步骤 2，将每次所得之商除以新进位制基数，记下余数，得到新进位制数相应的各位，直到最后相除之商为 0，这时的余数即为新进位制数的最高位(MSB)。

例 5　$(241)_D=(?)_B$

2⌊241	……余数为	1	LSB	b_0
2⌊120	……	0		b_1
2⌊60	……	0		b_2
2⌊30	……	0		b_3
2⌊15	……	1		b_4
2⌊7	……	1		b_5
2⌊3	……	1		b_6
2⌊1	……	1	MSB	b_7
0				

即　$(241)_D=(11110001)_B$

例 6　$(357)_D=(?)_O$

8⌊357	……余数为	5	LSB	o_0
8⌊44	……	4		o_1
8⌊5	……	5	MSB	o_2
0				

即　$(357)_D=(545)_O$

例 7　$(367)_D=(?)_H$

16⌊367	……余数	15 = F	LSB	h_0
16⌊22	……	6		h_1
16⌊1	……	1	MSB	h_2
0				

即　$(367)_D=(16F)_H$

纯小数部分的转换，采用基数乘法，即将待转换的十进制的纯小数，逐次乘以新进位制基数 R，取乘积的整数部分作为新进位制的有关数位。步骤如下：

(1)将待转换的十进制纯小数乘以新进位制基数 R，取其整数部分作为新进位制纯小数的最高位；

(2)将前步所得小数部分再乘以新进位制基数 R，取其积的整数部分作为新进位制小数的次高位；

(3) 重复前一步，直到小数部分变成 0 时，转换结束。或者小数部分虽未变成 0，但新进位制小数的位数已达到预定的要求(如位数的要求或者精度的要求)时，转换也可结束。

例 8　$(0.875)_D=(?)_B$

$$
\begin{array}{r}
0.875 \\
\times\quad 2 \\
\hline
1.750 \\
\quad 2 \\
\hline
1.500 \\
\quad 2 \\
\hline
1.000
\end{array}
\qquad
\begin{array}{ll}
\cdots\cdots\text{整为 } 1 & b_{-1} \\
\cdots\cdots\cdots\cdots 1 & b_{-2} \\
 & \\
 & \\
\cdots\cdots\cdots\cdots 1 & b_{-3}
\end{array}
$$

即　$(0.875)_D=(0.111)_B$

例 9　$(0.39)_D=(?)_B$

$$
\begin{aligned}
&0.39\times 2=0.78 && b_{-1}=0\\
&0.78\times 2=1.56 && b_{-2}=1\\
&0.56\times 2=1.12 && b_{-3}=1\\
&0.12\times 2=0.24 && b_{-4}=0\\
&0.24\times 2=0.48 && b_{-5}=0\\
&0.48\times 2=0.96 && b_{-6}=0\\
&0.96\times 2=1.92 && b_{-7}=1\\
&0.92\times 2=1.84 && b_{-8}=1\\
&0.84\times 2=1.68 && b_{-9}=1\\
&0.68\times 2=1.36 && b_{-10}=1\\
&\vdots
\end{aligned}
$$

即　$(0.39)_D=(0.110001111\cdots)_B$

此例中不能用有限位数实现准确的转换。转换后的小数究竟取多少位合适呢？实际中常用指定转换位数，如指定转换为八位，则$(0.39)_D=(0.01100011)_B$；也可根据转换精度确定位数。如此例要求转换精度优于 0.1%，即引入一个小于 $1/2^{10}=1/1024$ 的舍入误差，则转换到第十位时，转换结束。

如果是一个有整数又有小数的数，则整数小数应分开转换，再相加得转换结果。

例 10　$(52.375)_D=(?)_B$

整数为 52 按整数转换方法——基数除法

$$
\begin{array}{lll}
2\underline{|52} & \cdots\cdots\cdots\cdots 0 & b_0=0 \\
2\underline{|26} & \cdots\cdots\cdots\cdots 0 & b_1=0 \\
2\underline{|13} & \cdots\cdots\cdots\cdots 1 & b_2=1 \\
2\underline{|6} & \cdots\cdots\cdots\cdots 0 & b_3=0 \\
2\underline{|3} & \cdots\cdots\cdots\cdots 1 & b_4=1 \\
2\underline{|1} & \cdots\cdots\cdots\cdots 1 & b_5=1 \\
0 &
\end{array}
$$

即　$(52)_D=(110100)_B$

小数为 0.375 按基数乘法转换

$$
\begin{array}{ll}
0.375\times 2=0.75 & b_{-1}=0 \\
0.75\times 2=1.5 & b_{-2}=1 \\
0.5\times 2=1.0 & b_{-3}=1
\end{array}
$$

所以　$(0.375)_D=(0.011)_B$

即　$(52.375)_D=(110100.011)_B$

至于十进制数转换为八进制、十六进制，读者可根据上述方法自己练习。

二、二进制与八进制、十六进制的相互转换

由于二进制与八进制和十六进制之间正好满足 2^3 和 2^4 关系，因此它们之间的转换十分方便。

二进制转换为八进制、十六进制时，将二进制数由低位向高位每三位或每四位一组，若最高位一组不足位，则在有效位左边加 0，然后按每组二进制数转换为八进制数或十六进制数。

例 11　$(111010101)_B=(?)_O=(?)_H$

$111010101=111/010/101=(725)_O$

$111010101=0001/1101/0101=(1D5)_H$

八进制、十六进制转为二进制是上述的逆过程，分别将每位八进制数或十六进制用二进制代码写出来，然后写成相应的二进制数。

例 12　$(563)_O=(?)_B$　$(563)_H=(?)_B$

$(563)_O=101/110/011=(101110011)_B$

$(563)_H=0101/0110/0011=(010101100011)_B$

当要求将八进制和十六进制相互转换时，可通过二进制来完成。

例 13　$(8FC)_H=(?)_O$

$(8FC)_H=1000/1111/1100=(100011111100)_B$

$=100/011/111/100$

$=(4374)_O$

§3　二进制数的算术运算

同十进制数一样，二进制数也可进行加、减、乘、除四则运算，且运算规则也相同，所不同的是进位基数。由于二进制只有两个数字符号（0、1），所以二进制的运算比十进制的运算简单得多，且易于用数字电路来实现。

一、加法运算

和十进制一样，二进制数进行加法运算时进行按权对位相加，其加法规则是：

$0+0=0$　　$0+1=1$

$1+0=1$　　$1+1=10$

即“逢二进一”。

例 14　求 $(0111)_B$ 与 $(0011)_B$ 之和

```
  0 1 1 1      (7)
  0 0 1 1      (3)
+1 1 1          进位
-------------------
  1 0 1 0     (10)
```

其中最低位只是被加数与加数相加，不考虑低位向本位的进位，我们称为“半加”；而其它位还要考虑低位向本位的进位，即被加数＋加数＋低位向本位的进位，我们称为“全加”。

二、减法运算

二进制数相减时，也要先按权对位，然后同位数相减，其减法规则是：

$1-1=0$　　$1-0=1$

$0-0=0$　　$0-1=1$　（向高位借 1 当 2）

例 15　求 $(1101)_B$ 和 $(0110)_B$ 之差

```
  1 1 0 1      (13)
  0 1 1 0       (6)
-1 1
-------------------
  0 1 1 1       (7)
```

需要指出的是，在数字系统或计算机中，为减少设备量，而是先将二进制数表示成它的反码或补码形式，然后用加法代替减法运算。这方面的知识在本章 §4 介绍。

三、乘法运算

二进制乘法运算方法与十进制乘法相同，其乘法规则是：

$0\times0=0$　　$0\times1=0$

$1\times0=0$　　$1\times1=1$

例 16　求 $(1101)_B$ 与 $(0101)_B$ 的乘积

$$
\begin{array}{r}
1101 \quad (13) \\
\times \quad 0101 \quad (5) \\
\hline
1101 \\
0000 \\
1101 \\
0000 \\
\hline
1000001 \quad (65)
\end{array}
$$

四、除法运算

二进制除法与十进制除法类似，但是二进制数因只有两个值，故其商不是 1 就是 0，当商为 1，则被除数减除数。

例 17　求$(1111)_B$与$(101)_B$之商

$$
\begin{array}{r|l}
 & 11 \\
\hline
101 & 1111 \\
 & 101 \\
\hline
 & 101 \\
 & 101 \\
\hline
 & 000
\end{array}
$$

此例中余数为 0，正好除尽。如在规定位数除不尽，则商的位数由给定精度来确定。

§4　数的原码、反码及补码表示

前面所提到的二进制数，没有提到符号问题，故是一种无符号数。但在实际中数显然会有正有负，那么在数字设备中“+”、“−”符号是如何表示的呢?

一、机器数与真值

按我们习惯表示方法正 5 用+5 表示，二进制数为+101；负 5 用−5 表示，二进制数为−101。在数字设备中“+”、“−”也要数值化，一般将数的最高位设为符号位，“0”表示“+”、“1”表示为“−”。如

$$
\begin{array}{cc}
+101 & \rightarrow 0101 \\
-101 & \rightarrow 1101 \\
\text{(真值)} & \text{(机器数)}
\end{array}
$$

为了区分“+”、“−”号数值化前后的两个对应数，引入真值和机器数两个术语。连同符号位在一起的数称为机器数；而它的数值称为真值。

为了运算方便，即将减法运算变为加法运算，常用的机器数有原码、反码和补码三种形式。

二、原码、反码及补码

将数的真值形式中正数符号用符号位 0 表示，负数符号用符号位 1 表示时，叫做数的

原码形式、简称原码。如绝对值为 9 的数，它的真值形式和原码形式如下所示(用八位数码表示，最高位为符号位)：

数　　真值　　原码

$+9=+0001001=00001001$

$-9=-0001001=10001001$

原码的优点是易于辨认，因为它的数值部分就是该数的绝对值，而且与真值和十进制的转换十分方便。但在采用原码进行运算时，如两个异号数相减，则首先判定那个数的绝对值大，绝对值大的作为被减数，小的数作为减数，所得差数的符号与绝对值大的数的符号一致。这样数字设备就要增加判数大小的设备，且要用减法器来完成减法运算，显然增加了设备量。为了减少设备量，将减法变为加法运算，就引进了反码和补码形式。

与原码相比较，反码也是在数位左面加上一位符号位，0 代表正数，1 代表负数。与原码不一样的是，反码数位的形成与它的符号位有关：对于正数，反码与原码相同；对于负数，反码数位由原码数位逐位求反而得。这就是反码的由来。需要指出的是国外文献称反码为 1 的补码(1′ s complement)，国内已习惯称反码。

反码定义可用下式表示：

$$A^{*}=\begin{cases}A & (A\text{ 为正数时})\\ (2^{n}-1)+A & (A\text{ 为负数时})\end{cases}$$

式中 A 为真值，A^{*} 为 A 的反码。如 $+9$ 用四位二进制数表示

数　　原码　　反码

$+9=01001=01001$

-9 用四位二进制数表示

数　　原码

$-9=11001=A$

反码

$$\begin{aligned}A^{*}&=2^{4}-1+A\\&=(10000-1)+(-1001)=1111-1001\\&=0110\end{aligned}$$

即 $A^{*}=10110$。注意观察 A^{*} 与 A 的关系，符号位不变将 A 逐位取反即得 A^{*}。

机器数的第三种表示是补码形式。对于正数，原码、反码和补码的表示是相同的，对于负数表示则不相同。补码的最高位仍然是符号位，0 表示正数，1 表示负数。

补码定义可用下式表示：

$$A^{**}=\begin{cases}A & (A\text{ 为正数时})\\ (2^{n}+A) & (A\text{ 为负数时})\end{cases}$$

式中 A 为真值，A^{**} 为 A 的补码形式。

例 18　求 $+6$ 和 -6 的四位补码形式。

按定义正数 $A^{**}=A$ 故 $+6$ 的补码形式与原码一样 $A^{**}=0110$。

负数 $A^{**}=2^{n}+A$，在四位设备中，

$$2^{4}=10000,\qquad A=-0110$$

则　$[-6]_{补}=10000-0110=1010$

由上述可看出求补码的过程是做减法运算，若用机器来实现。又必须用减法运算设备。为此总结出求补的另外方法，就是“求反加 1”。如

$$-6=-110 \rightarrow 1110 \xrightarrow{\text{求反}} 1001 \xrightarrow{+1} 1010$$

真值　　原码　　反码　　补码

其结果与按定义求出的结果相同。需指出的是，求反过程中，符号位不变；加 1 只是对最低位加 1。在国外文献中称补码为 2 的补码(2′ s complement)。

对于反码和补码形式，当符号位为 1 时，一定要注意后面几位表示的不是此数的数值，如要知道此数的大小，一定要再求其反码或补码，此时后面的数位才表示其数值。如一个反码表示的数 10110，这是一个负数，它不等于 $(-6)_D$，而等于 $10110 \xrightarrow{\text{求反}} 11001$ 为 $(-9)_D$。一个补码表示的数 10111，这也是一个负数，它不等于 $(-7)_D$，而等于 $10111 \xrightarrow{\text{求反加 1}}$，11001 为 $(-9)_D$。这是在使用补码和反码时需注意的问题。

为了使读者对原码、反码和补码的关系有进一步了解，我们列出图 1-1。

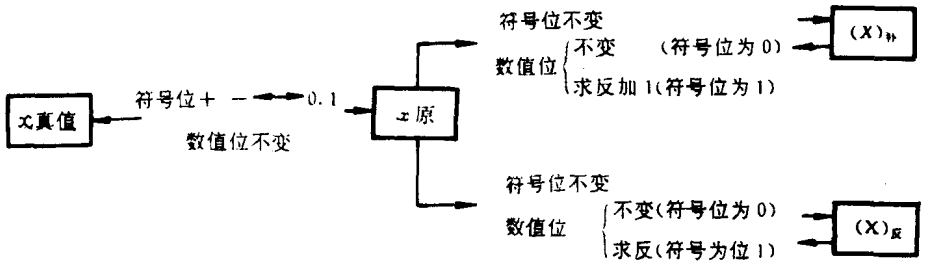

图　1-1

三、原码、反码及补码的算术运算

机器数有三种表示法，它们形成规则不同，算术运算的方法也不相同，下面通过例子来说明它们的不同之处。

例 19　已知 $X=+1101$，$Y=+0110$，用原码、反码及补码计算 $Z=X-Y$。

(*a*) 原码运算。

采用原码运算时，需将真值表示为原码：

$$[X]_{原}=01101 \qquad [Y]_{原}=00110$$

首先判别相减的两数是同号还是异号。若如同号，则进行减法；若为异号，则进行加法。本例 X、Y 同号，故进行减法。其次判别 X、Y 谁大谁小，以确定谁为被减数。本例 $|X|>|Y|$，故 X 为被减数，结果符号应与 $[X]_{原}$ 相同。

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

即　$[Z]_{原}=00111$　其真值 $Z=+0111$。