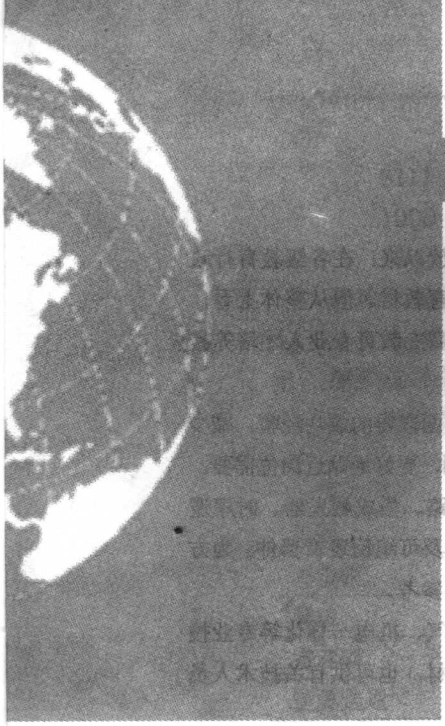
高 职 高 专 计 算 机 系 列 教 材

数字电子技术

常桂兰 王成安 任桂兰 主编

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE



高职高专计算机系列教材

数字电子技术

常桂兰　王成安　任桂兰　主编

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

教材建设是整个高职高专教育教学工作中的重要组成部分。改革开放以来，在各级教育行政部门、学校和有关出版社的共同努力下，各地已出版了一些高职高专教育教材。但从整体上看，教材建设仍落后于高职高专教育发展的需要。为贯彻执行教育部《高职高专教育专业人才培养目标及规划》，在有关部门的指导与配合下，我们编写了本部教材。

本教材汲取了近几年各院校在教学过程中与探索培养技术应用人才方面取得的成功经验，减少繁琐理论推导，注重结论与实践的应用，以培养能力为主，以应用为原则，更好地贴近岗位需要。

本书共分 8 章，内容包括数字电路基础、逻辑门电路、组合逻辑电路、集成触发器、时序逻辑电路、脉冲的产生与整形、数/模转换器和模/数转换器、半导体存储器及可编程逻辑器件。为方便教学，各章均有小结与习题。书后附有部分习题答案供学生和自学者参考，

本书可作为高职高专电气、电子、自动化、通信、计算机、汽车电气、机电一体化等专业技术基础课教材，也可以作为职工大学、业余大学同类专业的专业基础教材，也可供有关技术人员自学与参考。

图书在版编目（CIP）数据

数字电子技术/常桂兰，王成安，任桂兰主编. —北京：中国铁道出版社，2005.5（2007.8 重印）

（高职高专计算机系列教材）

ISBN 978-7-113-06534-8

Ⅰ.数… Ⅱ.①常… ②王… ③任… Ⅲ.数字电路：逻辑电路-高等学校：技术学校-教材 Ⅳ.TN79

中国版本图书馆 CIP 数据核字(2005)第 056491 号

书　　名：数字电子技术

作　　者：常桂兰　王成安　任桂兰

出版发行：中国铁道出版社（100054，北京市宣武区右安门西街 8 号）

策划编辑：严晓舟　商其坤

责任编辑：苏　茜　林菁菁

特邀编辑：李红玉

封面制作：白　雪

印　　刷：三河市国英印务有限公司

开　　本：787×1092　1/16　印张：16.25　字数：393 千

版　　本：2005 年 7 月第 1 版　　2007 年 8 月第 2 次印刷

印　　数：5 001 ~ 7 000 册

书　　号：ISBN 978-7-113-06534-8/TP・1502

定　　价：23.00 元

高职高专计算机系列教材

编 委 会

前　言

本书是在贯彻执行教育部《高职高专教育专业人才培养目标及规划》的教改中，积累了多年的教学改革与实践的经验，根据高职高专教育数字电子技术基础课程教学基本要求而编写的。可作为高职高专电气、电子、自动化、计算机、汽车电气、机电一体化等专业技术基础课教材，也可供从事电子技术的工作技术人员自学和参考。

为适应高职高专培养目标及现代科学技术的发展需求，本书以现代电子技术的基本知识与基本理论为主线，以应用为目的，删繁就简。使理论分析重点突出、实践性强。在内容安排上，以岗位需求和培养学生工作能力为目的，以应用为原则。编写内容与深度符合高职高专数字电子技术课程教育基本要求。在编写思路上注重知识更新，以小规模集成电路开始，逐步向大规模迈进，淡化内部结构与工作原理，注重器件外部功能特性和应用。本书的逻辑符号采用国家最新标准，在附录中有新旧符号对照表。书中每章后附有一定数量的习题，书后附有部分习题答案，有利于组织教学与学生自学。为提高学生识图能力，附录中有识图指导。本教材的参考学时为60～70学时。

全书共分8章。第1章 数字电路基础，介绍数的进制及转换，主要讲解逻辑代数及逻辑代数的化简。第2章 逻辑门电路，主要介绍各种门电路的组成及工作原理、门电路的应用及注意事项。第3章 组合逻辑电路，着重讲解组合逻辑电路的分析方法与设计方法，介绍几种常用的组合逻辑电路及其应用。第4章 集成触发器，主要讲解几种触发器的逻辑功能及特性。第5章 时序逻辑电路，着重讲解时序电路的分析方法，介绍几种实际应用的时序电路。随着EDA技术的出现，根据专业不同，时序电路的设计，可以作为选讲内容。第6章 介绍脉冲的产生与整形，重点介绍常用的整形电路和脉冲产生电路及其应用。第7章 介绍数/模转换器和模/数转换器，主要介绍D/A和A/D转换器的工作原理和技术指标，简单介绍集成DAC和ADC。第8章 半导体存储器及可编程逻辑器件，主要介绍各种存储器的工作原理，可编程逻辑阵列PAL和通用逻辑阵列GAL。全书各章节内容，各院校可根据专业不同，灵活取舍。

本书由常桂兰编写第1、2章；由任桂兰主持编写第3、4章及附录2，吴建军、王晓红参编；蒋新民编写第5、6章及附录3；王成安编写第7、8章及附录1，天津滨海职业技术学院任志娟参加了编写工作。全书由辽宁省2004年精品课主讲人常桂兰统稿。

本书在编写过程中，得到鞍山科技大学、辽宁机电职业技术学院、辽宁石油化工大学、沈阳师范大学等院校的有关领导和同行的帮助与支持，并提出了宝贵意见，另外，还得到中国铁道出版社的支持与帮助，在此，一并表示感谢。

尽管我们在编写过程中做了很多努力，但由于编者水平有限，加之时间仓促，书中难免会有许多疏漏和不妥之处，恳请使用本教材的师生和读者给予批评指正。

使用本教材者，中国铁道出版社计算机图书中心可提供电子教案。

编　者

2005年5月

目　录

第 1 章 数字电路基础

数字电路所讨论的是对数字量信息进行数值运算和逻辑加工的各种电路，它们是构成数字系统的基础。数字电路中的数字量信息是一种仅有两个可能取值的二值离散信号。因此，在学习数字电路之初，必须具备有关数字电路的基础知识。

本章首先介绍数字电路中常用的二进制数及其运算规律，然后阐明基本的逻辑概念、逻辑函数的表示方法以及逻辑代数的基本公式和常用公式等，最后以逻辑代数为工具讨论化简逻辑函数的几种方法。

1-1 数的进制及其转换

1-1-1 进位计数制

数是用来表示物理量的大小的。一个多位数是由一些数字符号（数码）按照一定的进位规则排列而成的。人们最习惯使用的是十进制数，但数字系统中的数，常常表现为二进制的形式，有时还采用八进制和十六进制形式。同一个数值的数，在不同场合下可以用不同进制的形式表示。

1. 十进制数

十进制数采用 0，1，2，3，…，9 十个不同的数码来表示任何一位数，十进制数的基数是 10，进位规律是“逢十进一”，各数码处在不同数位时，所代表的数值是不同的。例如：

$$192.85=1\times10^2+9\times10^1+2\times10^0+8\times10^{-1}+5\times10^{-2}$$

其中，10^2、10^1、10^0、10^{-1}、10^{-2} 分别称为十进制数各数位的权，都是 10 的幂。因此，对于任何一个十进制数，其数值都可表示为：

$$\begin{aligned}[N]_{10} &= k_{n-1}\times10^{n-1}+k_{n-2}\times10^{n-2}+\cdots+k_0\times10^0+k_{-1}\times10^{-1}\\ &\quad +k_{-2}\times10^{-2}+\cdots+k_{-m}\times10^{-m}\\ &=\sum_{i=-m}^{n-1}k_i\times10^i \end{aligned} \tag{1-1}$$

式（1-1）中，k_i 为基数 10 的第 i 次幂的系数；m、n 为正整数；$k_i\times10^i$ 称为加权系数。

2. 二进制数

二进制数只有两个数码 0 和 1，基数是 2，进位规律是“逢二进一”，每个数位的权是 2 的幂，同样，二进制数也可以按权展开

$$\begin{aligned}[N]_{2} &= k_{n-1}\times2^{n-1}+k_{n-2}\times2^{n-2}+\cdots+k_0\times2^0+k_{-1}\times2^{-1}\\ &\quad +k_{-2}\times2^{-2}+\cdots+k_{-m}\times2^{-m}\\ &=\sum_{i=-m}^{n-1}k_i\times2^i \end{aligned} \tag{1-2}$$

式（1-2）中，k_i 为基数 2 的第 i 次幂的系数；m，n 为正整数。

3. 八进制与十六进制

用二进制表示数时，数码串很长，书写和查错都很不方便，因此常用八进制和十六进制。

八进制数有 0，1，2，…，7 八个数码，基数是 8，进位规律是“逢八进一”，每个数位的权是 8 的幂。

十六进制数有 0，1，2，…，9，A，B，C，D，E，F 十六个数码，基数是 16，进位规律是“逢十六进一”，每个数位的权是 16 的幂。

八进制数和十六进制数都可以按权展开：

$$[N]_8 = k_{n-1}\times 8^{n-1} + k_{n-2}\times 8^{n-2} + \cdots + k_1\times 8^1 + k_0\times 8^0 = \sum_{i=0}^{n-1} k_i\times 8^i \tag{1-3}$$

$$[N]_{16} = k_{n-1}\times 16^{n-1} + k_{n-2}\times 16^{n-2} + \cdots + k_1\times 16^1 + k_0\times 16^0 = \sum_{i=0}^{n-1} k_i\times 16^i \tag{1-4}$$

1-1-2 不同进制之间的转换

1. 非十进制数转换成十进制数

正像上面讲述的那样，只要把式（1-2）R 进制的数在十进制计数体制内展开，再按十进制规则计算这个展开式的和，就能把一个非十进制数转换成十进制数。

例如：$(101.01)_2=1\times 2^2+0\times 2^1+1\times 2^0+0\times 2^{-1}+1\times 2^{-2}=(5.25)_{10}$

$(721.6)_8=7\times 8^2+2\times 8^1+1\times 8^0+6\times 8^{-1}=(465.75)_{10}$

$(4FA.8)_{16}=4\times 16^2+15\times 16^1+10\times 16^0+8\times 16^{-1}=(1274.5)_{10}$

2. 二进制和八、十六进制数之间的转换

八进制数中有 8 个数码 0～7，它的每一位数正好和一个 3 位二进制数相对应，即 $(000)_2=(0)_{10}=(0)_8$，…，$(111)_2=(7)_{10}=(7)_8$，一个二进制数要转换成八进制数时，只要把该二进制数按三位分为一组（从小数点处开始，分别向左、向右划分），每组就对应一位八进制数。按上述逆过程也可以把一个八进制数转换成二进制数。例如：

$$\begin{array}{cccc} (10 & 110 & 001. & 001)_2 \\ \Updownarrow & \Updownarrow & \Updownarrow & \Updownarrow \\ (2 & 6 & 1. & 1)_8 \end{array} \qquad 即(10110001.001)_2=(261.1)_8$$

十六进制中有 16 个数码 0～F，它的每一位正好和 4 位二进制数对应，即 $(0000)_2=(0)_{16}$…，$(1010)_2=(A)_{16}$=…，$(1111)_2=(F)_{16}$，因此二进制和十六进制数之间的转换也和上述二、八进制数的转换是类似的。例如：

$$(110\ 1010\ 0101.\ 01)_2=(6A5.4)_{16}$$

3. 十进制数转换成二进制数

十进制数转换成任意进制数都可用基数乘除法。

十进制整数转换成二进制数可采用“除 2 取余，逆序排列”法，其操作步骤如下：

（1）将给定的十进制数除以 2，余数便是二进制数的最低位。

（2）将上一步的商再除以 2，余数便是二进制数的次低位。

（3）重复步骤（2），直至商等于 0 为止。各次除得的余数，逆序排列，即可得到相应的二进制数。

【例 1-1】将十进制数$(53)_{10}$转换成二进制数。

解：

$$\begin{array}{r|l l}
2 & 53 & \cdots\cdots 1 \\
2 & 26 & \cdots\cdots 0 \\
2 & 13 & \cdots\cdots 1 \\
2 & 6 & \cdots\cdots 0 \\
2 & 3 & \cdots\cdots 1 \\
2 & 1 & \cdots\cdots 1 \\
 & 0 &
\end{array}$$

最后，得$(53)_{10}=(110101)_2$。

十进制小数可以用基数乘法转换成二进制小数，即所谓“乘 2 取整，顺序排列法”。下面通过一个例子说明具体的作法。

【例 1-2】将$(0.872)_{10}$转换成二进制数（误差 $e<\frac{1}{2^4}$）。

解：

$$\begin{array}{r}
0.872 \\
\times\quad 2 \\ \hline
\boxed{1}.744 \\
\times\quad 2 \\ \hline
\boxed{1}.488 \\
\times\quad 2 \\ \hline
\boxed{0}.976 \\
\times\quad 2 \\ \hline
\boxed{1}.952
\end{array}$$

最后，得$(0.872)_{10}=(0.1101)_2$，转换到第四位则误差小于$\frac{1}{2^4}$。

4. 二进制码

在数字系统的人机对话时，需要把十进制数值、不同的文字、符号用二进制数码来表示。建立这种与十进制数值、文字、符号一一对应的代码称为编码，常用的编码包括二—十进制码、格雷码以及字符代码等。

（1）二—十进制码

用二进制代码来表示一个给定的十进制数，称为二—十进制编码，简称 BCD 码（Binary Coded Decimal）。0 和 1 组成的 4 位二进制数有 $2^4=16$ 种组合方式，可任选其中 10 种来表示十进制数的 0～9 这十个数码，因此，编码的方案很多。表 1-1 给出了几种常用的二—十

进制编码。

因为4位二进制数代码共有16个不同的组合，用它对0～9十个十进制数编码，总有6个不用的状态，叫它无关状态，或称伪码，例如，8421码中的1010～1111为6个伪码。

表 1-1 常用的 BCD 码

十进制数	8421码	2421码	5421码	余3码	余3循环码
0	0000	0000	0000	0011	0010
1	0001	0001	0001	0100	0110
2	0010	0010	0010	0101	0111
3	0011	0011	0011	0110	0101
4	0100	0100	0100	0111	0100
5	0101	1011	1000	1000	1100
6	0110	1100	1001	1001	1101
7	0111	1101	1010	1010	1111
8	1000	1110	1011	1011	1110
9	1001	1111	1100	1100	1010

① 8421码、2421码、5421码

这几种代码的共同特点是，每一组代码中的每一位的权是固定不变的，称为恒权代码。其加权系数之和，即是所对应的十进制数。例如：

$$[1001]_{8421BCD}=[9]_{10}，[1001]_{5421BCD}=[6]_{10}$$

② 余3码

余3码所表示的十进制数比所对应的自然二进制码所代表的十进制数多“3”，余3码中的每一位没有固定的权，称为变权代码。余3码中，0和9的代码，1和8的代码，……，等都互为反码，是一种对9的自补代码。

③ 余3循环码

余3循环码也是一种变权代码，它从循环码（见表1-2）的第四个状态开始取10个状态代表十进制数。

表 1-2 循环码编码表

十进制数	循环码				十进制数	循环码			
	G_3	G_2	G_1	G_0		G_3	G_2	G_1	G_0
0	0	0	0	0	8	1	1	0	0
1	0	0	0	1	9	1	1	0	1
2	0	0	1	1	10	1	1	1	1
3	0	0	1	0	11	1	1	1	0
4	0	1	1	0	12	1	0	1	0
5	0	1	1	1	13	1	0	1	1
6	0	1	0	1	14	1	0	0	1
7	0	1	0	0	15	1	0	0	0

（2）循环码

循环码也称格雷（Gray）码，其编码表如表1-2所示。循环码的特点是任意两个相邻数所对应的代码之间仅有一位不同。

（3）字符编码

常用的字母和字符编码有 ASCII 码和 ISO 码。ASCII 码是美国标准信息交换码的简称，其编码表如表 1-3 所示。这是一组 8 位二进制代码，用 b_6～b_0 七位表示 2^7=128 个不同的信息，第 8 位 b_7 作为奇偶校验位。

表 1-3　ASCII 编码

字符 $b_6b_5b_4$ / $b_3b_2b_1b_0$	000	001	010	011	100	101	110	111
0000	NUL	DLE	SP	0	@	P	\	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	’	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	’	<	L	\	l	|
1101	CR	GS	−	=	M	]	m	}
1110	SO	RS	•	>	N	Λ	n	～
1111	SI	US	/	?	O	-	o	DEL

1-2　机器码

电子计算机普遍采用二进制。在机器中，数存放在由寄存单元组成的寄存器中。二进制的两个数码 1 和 0 是用寄存单元的两种稳定状态（如电位的高、低）来表示的。对于正号“+”或负号“−”，也只能用这两种不同状态来区别。因此，在机器中符号也就“数码化”了。并规定正数符号位用“0”表示，负数符号位用“1”表示，符号位放在一个数的最高位前面。符号位经“数码化”后的数称为机器码，因此，机器码是由符号位和数值两部分组成的。机器码有 3 种常见的表示方法，即原码、反码和补码。

1-2-1　原码

数的原码，其符号位表示该数的符号，而数值部分仍用原来的二进制数码表示。数 X 的原码记作$[X]_{原}$，例如：

$$X_1=+11001 \qquad [X_1]_{原}=[+11001]_{原}=0\ 11001$$

$$X_2=-11001 \qquad [X_2]_{原}=[-11001]_{原}=1\ 11001$$

1-2-2　反码

一个数如果是正数，其反码与原码相同。如果是负数，则除符号位仍为“1”外，将原码中的各位数码凡“1”换成“0”，凡“0”换成“1”即可。数 X 的反码记作$[X]_{反}$，例如：

$$X_1=+10011 \qquad [X_1]_{原}=0\ 10011 \qquad [X_1]_{反}=0\ 10011$$

$$X_2=-10011 \qquad [X_2]_{原}=1\ 10011 \qquad [X_2]_{反}=1\ 01100$$

显然$[[X]_{反}]_{反}=[X]_{原}$。因此，当已知一个数的反码，欲求其原码时，只要将其反码再求反即可。

1-2-3 补码

机器码用原码表示简单易懂，而且与数值换算方便。但是由于原码的符号位和数值是分别定义的，它们之间没有数值上的联系。所以运算结果的符号需要单独处理。例如，当两原码数进行加法运算时，首先要判别两数的符号，如果两数符号相同，则作加法运算，其结果的符号决定于参加运算的两数的符号；如果参加运算的两数为异号，则作减法运算，用绝对值大的数减去绝对值小的数，得到结果数，结果数的符号和两数中绝对值大的数的符号相同。减法运算电路比较复杂，而且运算速度也会降低。为了简化运算，人们研究了将符号和数值连在一起进行运算的方法，即将符号也看做一个数来进行运算，而不必单独处理。而且希望把减法运算变成加法运算，因而提出了补码。

1. 补码的概念

把减法化为加法来进行运算的例子，在日常生活中是经常遇到的。例如，校对时间，若标准时间是 6 点整，而时钟却指在 8 点整，快了 2 小时。为了将时间校准，很明显有两种校准方法。一是将表针倒拨 2 小时，这显然是一种减法运算，即

$$8-2=6$$

另一种办法是将表针正拨 10 小时，也同样可校准到 6 点，这种办法是加法运算，即

$$8+10=18=12+6 \xlongequal{\text{在钟面上}} 6$$

在钟面上仍是 6 点整。这里减“2”化为加“10”是有一定条件的，因为在钟面上正拨 12 小时，时钟的指针又回到原处，即对时钟来说加 12 等于不加。用数学式子表示，即

$$X+12 \xlongequal{\text{在钟面上}} X$$

于是

$$X+Y \xlongequal{\text{在钟面上}} X+Y+12=X+(12+Y)=X+[Y]_{补}$$

我们称$(12+Y)$是$[Y]$对 12 的补码，记作$[Y]_{补}$，12 称为模数。上式中 Y 是包含符号的数，若 Y 为负数，则实为减法运算。在只有有限个数的条件下，引进补码以后，可使减法运算化为加法运算。在二进制中，可利用存放二进制数的寄存器的位数是有限的，运算时可丢失最高位以上数码的特点，引进二进制负数的补码，从而可将减法运算化为加法运算。

【例 1-3】设 $X=+10011$ 即$(19)_{10}$，$Y=-00101$ 即$(-5)_{10}$

求：$X+Y$（设寄存器为 6 位，即在运算中第 6 位以上数码都会自动丢失）。

解 1：直接采用减法运算。因 X、Y 异号，且 $X>Y$，故实际上是将数值部分作减法运算，其结果与 X 符号相同，为正，即

$$X+Y=10011-00101=01110$$

结果为+0110 即$(14)_{10}$。

解 2：引进补码将减法变换为加法。如果将 $X+Y$ 加上 1000000，这对于 6 位寄存器来

说等于不加。即

$$X+Y \xlongequal{\text{在 6 位寄存器中}} X+Y+1000000=X+(1000000+Y)$$

因此，可将实际的减法运算变换成 $X+(1000000+Y)$的加法运算。$(1000000+Y)$称为$[Y]$的补码，记作$[Y]_{补}$。$(1000000)_2=(2^6)_{10}$ 为 6 位寄存器的模数。

$$[Y]_{补}=模数+Y \tag{1-5}$$

上例的减法，可变为先求在 6 位寄存器时的$[Y]_{补}$，再求 $X+[Y]_{补}$。

$$[Y]_{补}=[-00101]_{补}=1000000+(-00101)$$

$$\begin{array}{rl} & 1000000\cdots\cdots 模 \\ -) & 00101\cdots\cdots Y \\ \hline & 111011\cdots\cdots [Y]_{补} \end{array}$$

从上述负数求补过程看出，$[-00101]_{补}=1\ 11011$，它的符号位为“1”表示是负数的补码，而且它是求补运算的结果。所以有

$$X+[Y]_{补}$$

$$\begin{array}{rl} & 010011\cdots\cdots X \\ +) & 111011\cdots\cdots [Y]_{补} \\ \hline & \boxed{1}\ 0\ 01110\cdots\cdots [14]_{10} \end{array}$$

丢失

其结果与直接用减法运算结果相同。因此，引进补码后，运算结果的符号不用单独处理。符号位和数值一样同时参加运算，而且可将减法运算变成加法运算。这种变减为加的运算，只在寄存器具有有限位的条件下才成立。上例寄存器为 6 位，故 2^6 和零等效，其模数为 2^6。若寄存器为 n 位，则 2^n 和零等效，其模数为 2^n。

2. 补码的求法

直接按照补码的定义用式（1-5）求$[Y]_{补}$时，需作(2^n+Y)运算，若 Y 为负数，实际上仍要作减法运算。

如上例中

$$[Y]_{补}=[-00101]_{补}=1000000-00101=1\ 11011$$

为了避免作减法运算，将负数的求补公式（1-5）改写为

$$[Y]_{补}=1000000+Y=111111+1+Y=(111111+Y)+1$$

将 Y=−00101 代入得

$$[Y]_{补}=(111111-00101)+1$$

而

$$\begin{array}{rl} & 111111 \\ -) & 001101 \\ \hline & 111010 \end{array}$$

从$(111111+Y)$的运算结果看，若 Y 为负数，其结果正好是 Y 的反码，即$(111111+Y)=[Y]_{反}$。故负数的求补运算为

$$[Y]_{补}=[Y]_{反}+1 \tag{1-6}$$

如上例

$$Y=-00101$$

$$[Y]_{反}=1\ 11010$$

$$[Y]_{补}=[Y]_{反}+1=1\ 11010+0\ 00001=1\ 11011$$

这与直接按照补码定义求得结果相同。因此，对负数求补可用“求反加 1”的办法，即先求“反”，然后在反码的最低位加 1 即可。而在机器中实现一个数的反码和加 1 运算是很方便的。若 Y 是正数，从补码的定义用式（1-5）计算，显然正数的补码与原码相同。

【例 1-4】 设 $Y_1=+11001$，寄存器为 6 位，求$[Y_1]_{原}$、$[Y_1]_{反}$、$[Y_1]_{补}$。

解：因 Y_1 为正数，则$[Y_1]_{原}=[Y_1]_{反}=[Y_1]_{补}=0\ 11001$

【例 1-5】 设 $Y_2=-10110$，寄存器为 6 位，求$[Y_2]_{原}$、$[Y_2]_{反}$、$[Y_2]_{补}$。

解：$[Y_2]_{原}=1\ 10110$

$[Y_2]_{反}=1\ 01001$

$[Y_2]_{补}=[Y_2]_{反}+1=1\ 01010$

若已知负数的补码，则对补码求补可得原码。如例 1-5 中得到$[Y_2]_{补}=1\ 01010$，则$[[Y_2]_{补}]_{补}=[1\ 01010]_{补}=1\ 10110$，与$[Y_2]_{原}$相同。

3. 补码的加、减运算

若数码均以补码形式表示，称为补码系统。在补码系统中，加、减运算的结果也应是补码形式表示的数，并遵循两数之和的补码等于两数补码的和这一运算规则，即下列等式成立：

$$[X+Y]_{补}=[X]_{补}+[Y]_{补} \tag{1-7}$$

现分 3 种情况证明如下（设寄存器为 6 位）：

第一种情况：$X\geqslant0 \quad Y\geqslant0$

证明：因为 $X\geqslant0 \quad Y\geqslant0$ 则$[X+Y]\geqslant0$

由于正数的补码就是本身，故得

$$[X+Y]_{补}=X+Y$$

$$[X]_{补}=X$$

$$[Y]_{补}=Y$$

所以

$$[X+Y]_{补}=[X]_{补}+[Y]_{补}$$

第二种情况：$X<0 \quad Y<0$

证明：因为 $X<0 \quad Y<0$ 则 $X+Y<0$

所以 $[X]_{补}=X+1000000$

$[Y]_{补}=Y+1000000$

$[X+Y]_{补}=X+Y+1000000$

$[X]_{补}+[Y]_{补}=X+Y+1000000+1000000$

$10000000>X+Y+1000000+1000000>1000000$，丢失一个 1000000

则 $[X]_{补}+[Y]_{补}=X+Y+1000000$

所以 $[X+Y]_{补}=[X]_{补}+[Y]_{补}$

第三种情况：X 和 Y 符号不同

设 $X>0 \quad Y<0$ 则 $X+Y$ 有两种可能：