高等专科学校试用教材

数字电路

肖雨亭 主编

机械工业出版社

GAOZ

高等专科学校试用教材

数字电路

肖雨亭 主编

机械工业出版社

内 容 简 介

本教材是根据1983年9月机械工业部系统高等专科学校工业电气自动化专业"数字电路"课程组制定的编写大纲编写的。

全书共有十章，其内容有：数字电路基础、逻辑代数、各种集成门电路、组合数字电路、集成触发器、时序数字电路、大规模集成电路介绍、脉冲波形的产生和变换、数模转换器及模数转换器，以及顺序控制器等。书中各章附有习题和思考题，书后附有与教材相配合的实验指导书。

本教材可作为高等专科学校工业电气自动化专业或其它相近专业的"数字电路"试用教材，亦可供有一定电工知识的工程技术人员自学参考。

数 字 电 路

肖雨亭 主编

*

机械工业出版社出版(北京阜成门外百万庄南里一号)

(北京市书刊出版业营业许可证出字第117号)

机械工业出版社印刷厂印刷

新华书店北京发行所发行·新华书店经售

*

开本 787×1092 1/16·印张20 3/4·字数 505千字

1985年11月北京第一版·1985年11月北京第一次印刷

印数 0,001—7,000 ·定价4.30元

*

统一书号：15033·6231

前　言

本教材是根据1983年9月机械工业部系统高等专科学校工业电气自动化专业“数字电路”课程组制定的编写大纲编写的。

在编写本教材时，考虑到目前数字电路的一个主要发展方向——集成化，所以教材是以中、小规模集成电路为主来组织内容，并适当介绍大规模集成电路，以期与国内技术的发展相适应。一些新器件，如ECL、I^2L在本教材中也得到反映。本教材的每章后都附有习题，供学生复习巩固所学知识用。

本书侧重于组合数字电路和时序数字电路，并通过对具体电路的分析和设计来达到介绍数字电路分析方法和设计方法的目的。对于逻辑门和触发器，则是把它们作为构成数字电路的基础器件来处理的。讲解它们的工作原理和某些内部电路结构的目的，是为了讲清楚这些逻辑门和触发器的外部特性和参数，以便正确地使用它们来组成各种数字电路。

本课程的一个特点是理论紧密结合实践，在书后附有与教材内容相配合的实验指导书，使学生达到进一步理解教材内容并培养动手能力。

本书共十章，第一、三、五、八章由肖雨亭同志执笔，第二、四、六章由孙光国同志执笔，第七、九、十章由赵文忠同志执笔，实验一、二、三、五、六、七、八、九、十、十一由徐友和同志执笔，实验四、十二、十三、十四由顾治本同志执笔，肖雨亭同志担任主编，张广慈同志担任主审。

本教材在修改和定稿前，许多兄弟学校对试用稿进行了教学试用，并提出了很多宝贵意见；实验指导书部分初稿，曾经上海航校吴铁才同志审阅，在此表示衷心感谢。

由于数字电路近年来发展十分迅速，编者水平有限，在内容的取舍上和叙述方法上难免有不当或错误之处，恳望读者提出批评与修改意见。

编　者

目 录

绪　言

一、数字电路

电子电路中的电信号可以分为两类。一类是在时间上和数值上都是离散的信号，即它们的变化发生在一系列离散的瞬间，而数值大小和增减的变化都采用数字的形式。这一类信号称为数字信号。另一类则是除数字信号以外的所有信号。称为模拟信号。模拟信号的幅度变化是连续的。工作于数字信号下的电路称为数字电路；工作于模拟信号下的电路称为模拟电路。比起模拟电路来，数字电路有许多优点。

首先，数字电路采用二进制数制，每一位只有0和1两种可能状态。因此，基本结构比较简单，而且对元件的要求也不太严格，只要能区分开0和1就足够了。所以允许电路元件和电源的参数有较大的误差。(模拟电路处理的是连续变化的信号，要准确复制它们的波形和幅度的变化，因此对元件质量和电源的稳定度要求很高，而且往往还要选用比较复杂的电路。)同时，数字电路又是利用信号（脉冲）的有、无来代表和传输0和1这样的数字信息的。只有环境干扰相当强才能改变信号的有无。因此抗干扰能力比较强。

其次，由于数字电路采用二进制数制，能够广泛应用逻辑代数这一工具。除了使数字电路能够对信号进行算术运算外，还能进行逻辑推演和逻辑判断等一定的“逻辑思维”能力。正是由于数字电路同时兼备这两方面的功能，才使得现代电子计算机的制造成为可能。

另外，由于数字电路简单，又允许元件参数有较大的离散性，因此便于集成。而集成电路又具有使用方便、可靠性高、价格低廉等一系列优点。因此使数字电路得到了愈来愈广泛的应用。

数字电路中所研究的问题和使用的分析方法，如前所述与模拟电路有很大的不同。归纳起来是：

在模拟电路中，主要是研究微弱信号的放大以及各种型式信号的产生、变换等。而在数字电路中，重点则在于研究各种数字电路输入输出状态之间的相互关系，即通常所说的逻辑关系，所以数字电路又叫逻辑电路。为了分析这些逻辑关系，需要使用一套新的分析方法，其中包括逻辑代数，特征方程以及状态转换图等。

数字电路中的主要元件是起开关作用的开关元件。因此，开关元件的发展对数字电路的发展起着重要作用。从1948年出现了第二代电子元件（晶体管）代替数字电路中的第一代电子元件（电子管）以来，1962年又出现了第三代电子元件（集成电路），从外观上看集成电路已分不出各种元件、半导体器件和电路的界线了，它使电子设备中最容易发生故障的各种连接线和焊点大大缩短和减少。因此集成电路不但体积小重量轻，而且工作可靠。最初的集成电路，是在一块硅片上制作10～100个元件的小规模集成电路（简称SSI)。随着半导体制造工艺的改进，已从小规模集成电路经历了一块硅片上制作100～1000个元件的中规模集成电路（简称MSI）阶段发展到可以在一块硅片上制作1000个元件以上的大规模集成电路（LSI），和可以在一块硅片上制作十万个元件以上的超大规模集成电路（SLSI），例如在一个硅片上能集成一个完整的微型计算机。随着集成度的提高，器件的性能进一步得到改善，

使用更加方便。反过来又促进了数字电路的进一步发展。

数字电路的应用范围十分广泛，它不仅应用于雷达、电视、通信、遥测遥控等方面，并且在近代的测量仪表中也日益普遍地采用了数字电路。在数字电路的基础上发展起来的电子数字计算机，是当代科学技术最杰出的成就之一。电子计算机的出现，标志着技术发展进入了一个新的历史阶段，在此以前的一切技术手段，归根到底都用于减轻和代替人的体力劳动，而电子计算机的出现则开始了用技术手段减轻和部分代替人的脑力劳动，它不仅成了近代自动控制系统中不可缺少的一个组成部分，而且几乎渗透到了国民经济和人民生活的一切领域之中，并在许多方面引起了根本性的变革。可以相信，随着我国集成电路技术的进一步发展和完善，数字电路的应用必将得到更快的发展和普及。

二、本课程所包括的内容

在说明此问题之前，首先对脉冲作一介绍。所谓脉冲，从广义上讲，除了正弦信号以外的信号统称脉冲信号。脉冲波形千变万化种类繁多，图 0-1 给出了几种常见的波形。数字电路处理的数字信号多是矩形脉冲或尖脉冲，下面介绍如图 0-2 所示实际的矩形电压信号的主要参数。

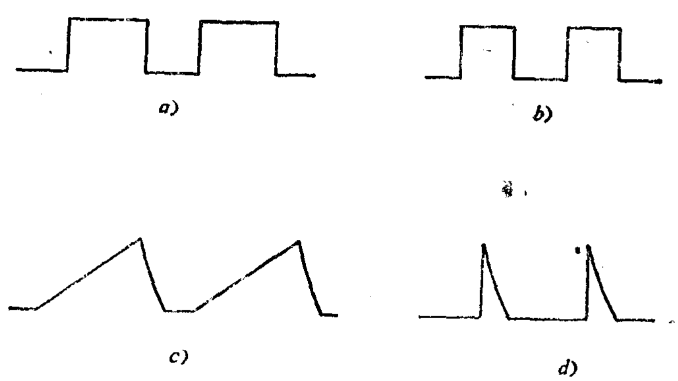

图0-1 几种常见脉冲波形

a）矩形脉冲 b）方波 c）锯齿波 d）尖脉冲

脉冲周期 T：在周期性连续脉冲中，相邻两个脉冲出现的时间间隔。

重复频率 f：在周期性连续脉冲中，每秒出现脉冲波形的次数。

脉冲幅度 V_m：脉冲电压波形变化的最大值。

脉冲上升时间（脉冲前沿）t_r：波形从 $0.1V_m$ 上升到 $0.9V_m$ 所需时间。

脉冲下降时间（脉冲后沿）t_f：波形从 $0.9V_m$ 下降到 $0.1V_m$ 所需时间。

脉冲宽度 t_p：常以波形 $0.5V_m$ 处前后沿时间间隔来计算。对于尖脉冲常用 $0.1V_m$ 处前后沿之间的时间间隔来计算，如图 0-3 所示。

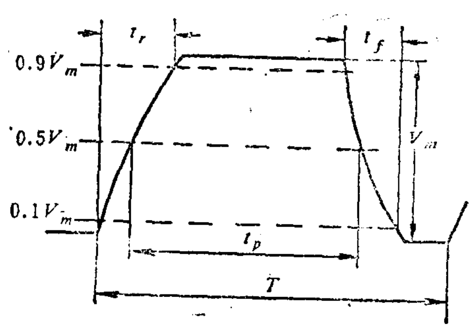

图0-2 矩形电压脉冲参数

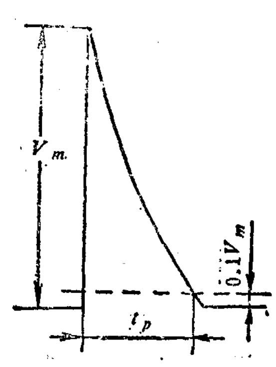

图0-3 尖脉冲的脉宽

现以简单的数字频率计为例(原理方框图见图 0-4)，概括地说明本课程所包含的内容。频率计是测量周期信号频率的仪器。测量时，首先将被测信号（不一定是正弦波）变换整形为规则的矩形脉冲，再将规则的矩形脉冲送给由“秒脉冲”控制的“门”电路，此脉冲通过被打开一秒钟的“门”电路后，再进入计数器，计数器计下一秒钟时间内矩形脉冲的个数，并将结果送到显示器，从而在显示器上读出被测信号的频率。频率计各部分的波形如图 0-5 所示。

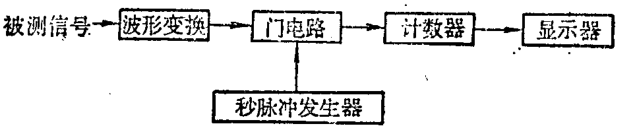

图0-4　数字频率计方框图

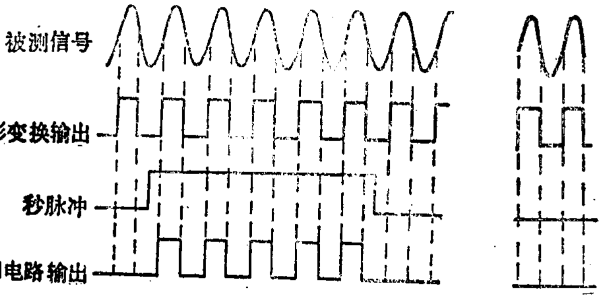

图0-5　数字频率计波形图

可见，本课程包括的内容有脉冲的产生、变换、控制、计数及显示等。本教材以数字电路为中心，按逻辑门→组合数字电路→触发器→时序数字电路的顺序进行讲述。以小规模集成电路为基础来说明各种电路的基本原理，分析方法或设计方法，在此基础上介绍中规模数字集成电路并简略介绍大规模集成电路。此外，还介绍数字电路的接口电路——数-模转换及模-数转换等电路。

第一章　数字电路基础

数字系统是以数字为对象进行运算或处理的。本章从十进制数开始，重点介绍数字系统中常用的二进制数、八进制数和十六进制数，以及二进制与其它进制之间的相互转换和机器数。此外，还介绍几种常见的 RC 电路和晶体管的开关特性及反相器，作为数字电路的基础。

§1-1　计数体制

一、十进制数

十进制数是用 0 ～ 9 十个基本数字符号表示数值，通常把这些数字符号称为数码。数码处于不同的位置（或称数位），有不同的位值（或称权）。例如十进制数 428，右起第一位为个位，其位值为 1——权为 1，右起第二位为十位，其权为 10，右起第三位为百位，其权为 100。相邻高位的权是低位权的十倍。因此 428 可写成：

$$428 = 4 \times 10^2 + 2 \times 10^1 + 8 \times 10^0$$

任意一个十进制整数 N（为方便起见，以正为例）都可表示为权的展开式：

$$N = K_{n-1} \times 10^{n-1} + K_{n-2} \times 10^{n-2} + \cdots\cdots + K_1 \times 10^1 + K_0 \times 10^0$$

$$= \sum_{i=n-1}^{0} K_i 10^i \tag{1-1}$$

式中　n——正整数；

K_i——0 ～ 9 十个数码中的任何一个，由 N 决定。

10称为十进制的基数。所谓某进位制的基数，就是在该进位制中可能用到数码的个数，所以十进制数的基数为10。十进制数的运算规律是“逢十进一”和“借一当十”。

二、二进制数

十进制数是日常使用最为普通的数制，但并不是唯一的数制。数制的选择和使用有一定的客观条件，如果说过去人们用十个指头计数产生了十进制，那么今天用电子元件的导通和截止两个状态来表示十进制就比较困难了，因此出现了二进制。

二进制数只用 0 和 1 两个数码来表示数值，并采用“逢二进一”和“借一当二”的运算规律。为了熟悉二进制数的表示方法，现列出部分十进制数和二进制数的对照表，见表 1-1。

表1-1　部分十进制数和二进制数对照表

十进制数	二进制数	十进制数	二进制数	十进制数	二进制数
0	0	7	111	14	1110
1	1	8	1000	15	1111
2	10	9	1001	$16 = 2^4$	10000
3	11	10	1010	$32 = 2^5$	100000
4	100	11	1011	$64 = 2^6$	1000000
5	101	12	1100	$128 = 2^7$	10000000
6	110	13	1101	$256 = 2^8$	100000000

从表 1-1 看出，在二进制中数码 1 所处的数位不同，所表示的十进制数也不同，即权不同。在二进制中，1 在某一位所表示的相当于十进制的数，称为这一位的“权”。因此，二进制从低位到高位的权分别是 2^0、2^1、2^2、2^3……，可见相邻高位的权是低位权的两倍。要表示的数值越大，需要二进制的位数越多。n 位二进制数最大能表示的数为 2^n-1，当 $n=3$ 时最大可表示的数为 $2^3-1=7$。

应用权的概念，可以把一个二进制数写成权的展开式。如：

$$1011 = 1\times 2^3 + 0\times 2^2 + 1\times 2^1 + 1\times 2^0$$

任意一个二进制整数 N，以正为例都可表示为

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

$$= \sum_{i=n-1}^{0} K_i \times 2^i \tag{1-2}$$

式中 K_i 只能取 0 或 1；2 为基数。

十进制数早已为人们所熟悉并广泛应用。但是在数字电路中却普遍采用二进制，这是由于二进制具有如下独特的优点：

（1）二进制数只有 0 和 1 两个数码，任何具有两个不同稳定状态的元件都可用来表示一位二进制数，因此表示二进制的方法容易实现。例如继电器触头的闭合和断开；晶体管的导通和截止；电位的高和低；脉冲的有和无等。只要规定其中的一种状态为 1，另一种状态为 0，就可以表示一位二进制数了。

（2）因为二进制只有两个数码，两两相加为 $0+0=0$，$1+0=0+1=1$，$1+1=10$；两两相乘为 $0\times 0=0$，$1\times 0=0\times 1=0$，$1\times 1=1$，一共六种情况。而十进制有十个数码，两两相加和两两相乘共有 110 种情况，因而用数字电路实现二进制的运算比较容易。

（3）若要表示 0～99之间的任一个数，如用二进制表示，则为七位（$2^7-1=127$），只要七个电子计数元件就可实现。而在十进制中则要两位十进制数，每位需要四个电子计数元件，共需八个电子计数元件才能完成。一般为了表示同一个数，十进制所需的元件要比二进制多20%。因此采用二进制可以节省设备。

（4）采用二进制可以使用逻辑代数这一数学工具，大大方便了数字电路的阅读和设计。

三、八进制与十六进制数

二进制数便于机器识别和运算，但二进制数的位数多，读起来困难，写起来太长。为了弥补二进制书写太长的缺点，常采用八进制和十六进制。

在八进制中，基数为 8。使用 0、1、2、3、4、5、6、7 这八个数码。相邻高位的权是低位权的八倍，采用“逢八进一”的规律。

任何一个八进制正整数 N，都可以写成权的展开式：

$$N = K_{n-1}\times 8^{n-1} + K_{n-2}\times 8^{n-2} + \cdots\cdots + K_1\times 8^1 + K_0\times 8^0$$

$$= \sum_{i=n-1}^{0} K_i \times 8^i \tag{1-3}$$

在十六进制中，基数为16。要求十六个数码，除十进制中使用的 0～9 十个数码外，还

补充A、B、C、D、E、F六个字母作为数码用。在十六进制中，相邻高位的权是低位权的十六倍。采用“逢十六进一”的规律。

任何一个十六进制正整数N，其权的展开式为

$$N=K_{n-1}\times16^{n-1}+K_{n-2}\times16^{n-2}+\cdots\cdots+K_1\times16^1+K_0\times16^0$$

$$=\sum_{i=n-1}^{0}K_i\times16^i \qquad (1\text{-}4)$$

表 1-2 列出了八进制和十六进制的等值二进制数和十进制数。

表 1-2

八进制			十六进制		
八进制数码	等值二进制数	等值十进制数	十六进制数码	等值二进制数	等值十进制数
0	000	0	0	0000	0
1	001	1	1	0001	1
2	010	2	2	0010	2
3	011	3	3	0011	3
4	100	4	4	0100	4
5	101	5	5	0101	5
6	110	6	6	0110	6
7	111	7	7	0111	7
10	001 000	8	8	1000	8
11	001 001	9	9	1001	9
12	001 010	10	A	1010	10
13	001 011	11	B	1011	11
14	001 100	12	C	1100	12
15	001 101	13	D	1101	13
16	001 110	14	E	1110	14
17	001 111	15	F	1111	15

四、二进制数与其它进制数的转换

（一）二进制与十进制的相互转换

二进制数转换为十进制数的方法很简单，只要写出二进制数的权展开式，然后按“权”相加，就可得到等值的十进制数。

例1-1 $(11010)_2=1\times2^4+1\times2^3+0\times2^2+1\times2^1+0\times2^0$

$=16+8+0+2+0$

$=(26)_{10}$

注意：$(N)_2$表示N为二进制数，$(N)_{10}$表示N为十进制数，同理$(N)_8$或$(N)_{16}$表示N为八进制或十六进制数。

下面介绍一种将十进制整数转换为二进制数的方法—除2取余法。

为了便于理解转换规则，先举一个实例。

例1-2 将十进制数41换算成二进制数。

设
$$(41)_{10}=(K_{n-1}K_{n-2}\cdots K_1K_0)_2$$

由于任意一个二进制数都可以写成权的展开式，因此上式可写成

$$(41)_{10}=K_{n-1}\times2^{n-1}+K_{n-2}\times2^{n-2}+\cdots\cdots+K_1\times2^1+K_0\times2^0$$

$$(41)_{10}=2(K_{n-1}\times 2^{n-2}+K_{n-2}\times 2^{n-3}+\cdots\cdots+K_1)+K_0$$

等式两边同除以 2 得

$$20+\frac{1}{2}=K_{n-1}\times 2^{n-2}+K_{n-2}\times 2^{n-3}+\cdots\cdots+K_1+\frac{K_0}{2}$$

两数相等，整数部分和小数部分必对应相等，因此得

$$\frac{1}{2}=\frac{K_0}{2},\ K_0=1,\quad 它正好是\frac{41}{2}的余数。$$

$$20=2(K_{n-1}\times 2^{n-3}+\cdots\cdots+K_2)+K_1$$

将上面整数的等式两端再除以 2，可得余数为 0，即 $K_1=0$。依次类推可得 K_2、K_3、… K_n 各值。其步骤可记成

$$\begin{array}{r|l}
2 & \underline{41} \\
2 & \underline{20}\cdots\cdots 余数=K_0=1 \\
2 & \underline{10}\cdots\cdots 余数=K_1=0 \\
2 & \underline{5}\ \cdots\cdots 余数=K_2=0 \\
2 & \underline{2}\ \cdots\cdots 余数=K_3=1 \\
2 & \underline{1}\ \cdots\cdots 余数=K_4=0 \\
 & 0\ \cdots\cdots 余数=K_5=1
\end{array}\ \uparrow$$

转换结果 $(41)_{10}=(101001)_2$

由此可知，将十进制数不断用“2”去除，直至商为 0，按从下往上的顺序写下余数，即为所求的二进制数。

（二）二进制与八进制的相互转换

因八进制的基数 8 是二进制基数 2 的 3 次幂。因此，二进制与八进制之间的换算是很方便的。

二进制转换为八进制的方法：从小数点起，把二进制数按每三位分组，然后写出每一组的等值八进制数，顺序排列起来即得该二进制数的等值八进制数。

例1-3 求 $(11101111010\cdot1011)_2$ 的等值八进制数。

解 011 101 111 010 · 101 100从小数点起，按每三位分组

3 5 7 2 · 5 4 每组的等值八进制数

转换结果

$$(11101111010\cdot1011)_2=(3572\cdot54)_8$$

八进制转换为二进制的方法可以采用与前面相反的步骤。

例1-4 求 $(712\cdot46)_8$ 的等值二进制数。

解 7 1 2 · 4 6 八进制数

111 001 010 · 100 110 每一位的等值二进制数

换算结果

$$(712\cdot46)_8=(111001010\cdot100110)_2$$

（三）二进制与十六进制的相互转换

因十六进制的基数16是二进制基数 2 的 4 次幂，因此，二进制与十六进制之间的转换也是方便的。

二进制数转换为十六进制数时，只要从小数点起把二进制数按每四位分组。然后写出每一组的等值十六进制数，顺序排列起来即得该二进制数的等值十六进制数。采用相反的步骤即可将十六进制数转换为等值二进制数。

例1-5 求$(11011011010 \cdot 001)_2$的等值十六进制数。

解 0110 1101 1010・0010 从小数点起按每四位分组

6 D A・2 每组的等值十六进制数

转换结果

$$(11011011010 \cdot 001)_2 = (6DA \cdot 2)_{16}$$

例1-6 求$(8FB \cdot 5)_{16}$的等值二进制数。

解 8 F B・5 十六进制数

1000 1111 1011・0101 每一位的等值二进制数

转换结果

$$(8FB \cdot 5)_{16} = (100011111011 \cdot 0101)_2$$

§1-2 机 器 码

电子数字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。二进制的二个数码1和0是用寄存器单元的两种不同状态（如电位的高、低）来表示。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。我们把这种符号位数码化了的数称为机器码，因此机器码是由符号位和数值两部分组成。机器码有三种常见的表示方法，即原码、反码和补码。

一、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二进制数码表示。数X的原码记作$[X]_{原}$，如

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

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

二、反码

规定一个数如果是正数，其反码与原码相同。如果是负数，则除符号位仍为“1”外，只要将原码中的各位数码凡“1”都换成“0”，凡“0”都换成“1”即可。数X的反码记作$[X]_{反}$，如

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

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

显然$[[X]_{反}]_{反} = [X]_{原}$。因此，当已知一数的反码，欲求其原码时，只要将其反码再求反即可。

三、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、减法运算时，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

结果的符号是参加运算两数的符号；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，而且运算速度也要降低。为了简化运算，人们研究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。

（一）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，快了两小时。为了将时间校准，很明显有两种校准方法，一是将表针倒拨两小时，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个小时，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，即有

$$X+12 \xlongequal{\text{在钟面上丢失12}} X$$

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。

因此，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1-7 设$X=+10011$即$(19)_{10}$、$Y=-00101$即$(-5)_{10}$

求 $X+Y$

（设寄存器为六位，即在运算中第六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$+01110$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，这对于六位寄存器来说等于不加。即

$$X+Y \xlongequal{\text{在六位寄存器中}} X+Y+1000000$$
$$=X+(1000000+Y)$$

因此，可将实际的减法运算变成$X+(1000000+Y)$的加法运算。$(1000000+Y)$称为$[Y]$的补码，记作$[Y]_{补}$。$(1000000)_2=(2^6)_{10}$为六位寄存器的模数。

$$[Y]_{补}=模数+Y \qquad (1-5)$$

上例的减法，可变为先求在六位寄存器时的$[Y]_{补}$。后再求$X+[Y]_{补}$。

$$[Y]_{补}=[-00101]_{补}=2^6+[-00101]$$

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

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。

$$X+[Y]_{补}=0\ \ 10011+1\ \ 11011=0\ \ 01110 \text{即}(14)_{10}$$

$$\begin{array}{rrl} & 0 & 10011\cdots\cdots X \\ +) & 1 & 11011\cdots\cdots[Y]_{补} \\ \hline \boxed{1} & 0 & 01110\cdots\cdots(14)_{10} \end{array}$$

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，是在寄存器具有有限位的条件下成立。上例寄存器为六位，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。

（二）补码的求法

如果直接按照补码定义用式(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
$$\begin{aligned}[Y]_{补}&=[-00101]_{补}\\&=1000000-00101\\&=111011\end{aligned}$$

为了避免作减法运算，将负数的求补公式(1-5)改写如下：
$$\begin{aligned}[Y]_{补}&=1000000+Y=111111+1+Y\\&=(111111+Y)+1\end{aligned}$$

将$Y=-00101$代入得

$$[Y]_{补}=(111111-00101)+1$$

而
$$\begin{array}{rr} & 111111 \\ -) & 00101 \\ \hline & 111010 \end{array}$$

从$(111111+Y)$的运算结果看，若Y为负数，凡Y数值的某一位为“0”时，其差的对应位为“1”，某一位为“1”时，其差的对应位为“0”，正好是Y的反码，即$(111111+Y)=[Y]_{反}$。

而在机器中实现一个数的反码是非常方便的。因此，可将求负数的补码$[Y]_{补}$用“求反加一”的办法，即先求反，然后在反码的最低位加1即可，故得

$$[Y]_{补}=[Y]_{反}+1 \tag{1-6}$$

如上例
$$Y=-00101$$
$$[Y]_{反}=1\ \ 11010$$