



国外经典教材·电子信息

PEARSON
Prentice
Hall

Programmable Controllers Using the Allen-Bradley SLC-500 Family (Second Edition)

可编程序控制器 原理与设计 (第2版)

(美) David A. Geller 著
于 玲 译
潘丽萍 审校



清华大学出版社

国外经典教材·电子信息

可 编 程 序 控 制 器 原 理 与 设 计

(第2版)

(美) David A. Geller 著

于 玲 译

潘丽萍 审校

清华大学出版社

北 京

内 容 简 介

罗克韦尔自动化旗下Allen-Bradley公司的SLC-500系列可编程序控制器体积小、功能强、性价比高,已经在各行业得到了广泛应用。本书结合作者多年的工程实践和教学经验,详细介绍了SLC-500系列可编程序控制器的基本组成、指令系统与编程。通过本书的阅读,读者可了解可编程序控制器的原理,进而设计电气控制系统和编写自动化控制软件。

全书注重实际,强调应用,是一本工程性较强的应用类图书,可作为大专院校工业自动化、电气工程及其自动化、应用电子、计算机应用、机电一体化及其他相关专业的教材,也可作为自学和培训教材供工程技术人员使用,对MicroLogix 1200系列的用户也有很大的参考价值。

Simplified Chinese edition copyright © 2005 by PEARSON EDUCATION ASIA LIMITED and TSINGHUA UNIVERSITY PRESS.

Original English language title from Proprietor's edition of the Work.

Original English language title: Programmable Controllers Using the Allen-Bradley SLC-500 Family, Second Edition by David A. Geller, Copyright © 2005

EISBN: 0-13-113052-8

All Rights Reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as Pearson Education.

This edition is authorized for sale only in the People's Republic of China (excluding the Special Administrative Region of Hong Kong and Macao).

本书中文简体翻译版由Pearson Education授权给清华大学出版社在中国境内(不包括中国香港、澳门特别行政区)出版发行。

北京市版权局著作权合同登记号 图字: 01-2004-6479

版权所有,翻印必究。举报电话:010-62782989 13501256678 13801310933

本书封面贴有Pearson Education(培生教育出版集团)激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

可编程序控制器原理与设计(第2版)/(美)盖勒(Geller, D. A.)著;于玲译;潘丽萍审校.—北京:清华大学出版社,2006.3

书名原文:Programmable Controllers Using the Allen-Bradley SLC-500 Family, Second Edition
(国外经典教材·电子信息)

ISBN 7-302-12288-1

I. 可… II. ①盖… ②于… ③潘… III. 可编程序控制器—教材 IV. TN312

中国版本图书馆CIP数据核字(2005)第158853号

出 版 者:清华大学出版社 地 址:北京清华大学学研大厦
http://www.tup.com.cn 邮 编:100084
社 总 机:010-62770175 客户服务:010-62776969

文稿编辑:文开棋

封面设计:久久度文化

印 刷 者:北京市清华园胶印厂

装 订 者:三河市金元印装有限公司

发 行 者:新华书店总店北京发行所

开 本:185×260 印张:18.25 字数:432千字

版 次:2006年3月第1版 2006年3月第1次印刷

书 号:ISBN 7-302-12288-1/TN·297

印 数:1~3500

定 价:34.00元

译者序

可编程控制器(PLC)是专门为工业环境下应用而设计的工业控制计算机,已经成为电气控制系统中应用最广的核心装置。它不仅可以实现复杂的逻辑控制,还能够完成各种顺序或定时的闭环控制功能。它具有多方面的优势,抗干扰能力强、可靠性高、稳定性好、体积小,能在恶劣环境下长时间、不间断的运行。另外,它编程简单,维护方便,并配有各类通讯接口与模块,可实现各种级别的连接。

在可编程控制器市场,罗克韦尔自动化的可编程控制器技术非常成熟,是业内最值得信赖的品牌之一。它旗下的Allen-Bradley公司所生产的SLC-500系列处理器,体积小,功能强,配置灵活,性价比高,特别适合中、小型应用,但也有很多大型工业应用也采用了这一系列的控制器中。从食品加工到钢铁制造,从环保、水处理、造纸到石化,从单机应用到分布式控制应用,都可以见到它的踪影。例如,北方某2X300MW机组电站的循环水控制系统、上海宝钢集团上钢一厂连铸连轧项目的大型空压站、汕头市梅溪桥闸、旺隆水电站、广东省恩平市水处理厂、自来水工程计算机数据遥测调度系统以及汉星显示器公司的控制系统等,都采用SLC-500作为核心部件。

Allen-Bradley公司的SLC-500系列处理器在国内已经引起了极大的关注,很多读者迫切需要这方面的书补充新的知识。本书正是基于这个需求引进的。作者结合自己多年的实践和教学经验,详细介绍了如何用SLC-500进行控制系统的设计。首先介绍基本的二进制数学基础,接着讲述PLC控制系统所涉及到的硬件设备,包括输入输出模块、与输入输出模块相连的典型输入输出设备、控制器模块,说明这些硬件设备之间如何进行连接,包括直接的接线连接、内存和地址之间的转换映射。然后介绍PLC的软件,包括各种寻址方式和梯形图的指令格式以及一些常用的梯形图指令。

全书特别强调基于团队的设计方法,这是大多数公司目前普遍采用的一种方法。作者还从项目管理者的角度出发,详细描述了装置的设计过程,并通过多个不同的例子来说明在装置的类型、运行方式、操作方式以及警报级别不同的情况下,如何设计出合理的PLC控制程序。

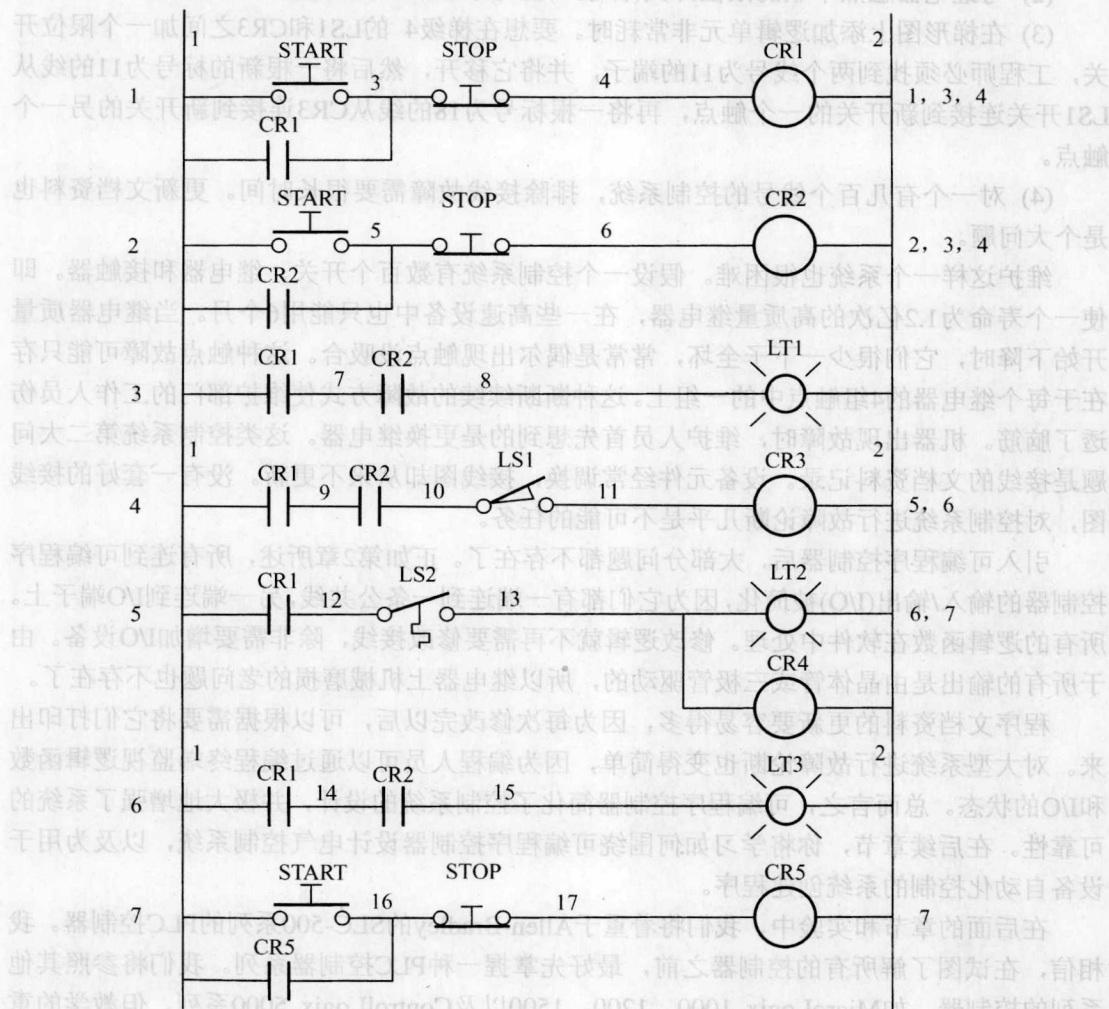
本书通俗易懂,示例丰富,并在每章最后提供复习题,十分适合初次进行工程实践或者不具备较多控制知识的读者。对于经验丰富的工程师,本书中有关装置和控制程序的设计思想和设计方法的论述也会使其受益匪浅。虽然本书以Allen-Bradley公司的SLC-500系列PLC为例,但其他型号或者其他PLC产品的用户也能从中得到较大的收获。

全书由于玲翻译、潘丽萍审校。在本书的译校过程中,得到浙江大学罗克韦尔自动化实验室主任黄海龙老师的大力帮助,在此表示诚挚的谢意。由于译者水平所限,难免有错漏之处,望读者不吝指正。

译者
于浙江大学

前言

早在20世纪70年代微处理器诞生之前，所有机器的控制都由继电器来完成。控制系统由所有逻辑元件(继电器、开关、按钮)之间的独立连线构成。设计这类系统时，设计工程师要画电气接线图，标出所有部件的接线方法。图L.1给出了这样一个例图。从例图中可以看到，这类电气图有垂直方向的主电源线和两根主电源线之间水平方向的逻辑连线。这种布局类似一个梯子，鉴于此，我们称之为梯形图。



图L.1 梯形图示例

注意图L.1电源线外侧的数字。左边的数字是梯级号。横跨电源线的每条线路都被视为一条单一的梯级，即使它不止一个分支，如梯级1、2和7就有多个起动分支。电源线右侧的数字表示在哪些梯级中用了这个控制继电器的触点。通过这种方式，电气工程师就可以很

容易地识别线路。这个例子总共只有7个梯级。大型设备的梯形图中通常有好几百条梯级。

你可能还注意到,图中用到的每一根线都有标号。要对于一个包含几百个电线标号的大系统进行故障诊断,这样做是必要的。因为每根线都是连着的,标号就贴在每根线的终端。

尽管这种控制系统概念已经沿用了很多年,但还存在一些比较大的设计和维护问题,例如:

(1) 每一控制继电器最多有4组触点。如果需要使用超过4组的触点,就需要增加一个并联的继电器。

(2) 与继电器触点串联的限位开关(梯级4)在每个触点上都需要长串的电线的。

(3) 在梯形图上添加逻辑单元非常耗时。要想在梯级4的LS1和CR3之间加一个限位开关,工程师必须找到两个线号为11的端子,并将它移开,然后将一根新的标号为11的线从LS1开关连接到新开关的一个触点,再将一根标号为18的线从CR3连接到新开关的另一个触点。

(4) 对一个有几百个线号的控制系统,排除接线故障需要很长时间。更新文档资料也是个大问题。

维护这样一个系统也很困难。假设一个控制系统有数百个开关、继电器和接触器。即使一个寿命为1.2亿次的高质量继电器,在一些高速设备中也只能用6个月。当继电器质量开始下降时,它们很少一下子全坏,常常是偶尔出现触点没吸合。这种触点故障可能只存在于每个继电器的4组触点中的一组上。这种断断续续的故障方式使维护部门的工作人员伤透了脑筋。机器出现故障时,维护人员首先想到的是更换继电器。这类控制系统第二大问题是接线的文档资料记录。设备元件经常调换,接线图却从来不更新。没有一套好的接线图,对控制系统进行故障论断几乎是不可能的任务。

引入可编程序控制器后,大部分问题都不存在了。正如第2章所述,所有连到可编程序控制器的输入/输出(I/O)被简化,因为它们都有一端连到一条公共线,另一端连到I/O端子上。所有的逻辑函数在软件中处理。修改逻辑就不再需要修改接线,除非需要增加I/O设备。由于所有的输出是由晶体管或三极管驱动的,所以继电器上机械磨损的老问题也不存在了。

程序文档资料的更新要容易得多,因为每次修改完以后,可以根据需要将它们打印出来。对大型系统进行故障论断也变得简单,因为编程人员可以通过编程终端监视逻辑函数和I/O的状态。总而言之,可编程序控制器简化了控制系统的设计,并极大地增强了系统的可靠性。在后续章节,你将学习如何围绕可编程序控制器设计电气控制系统,以及为用于设备自动化控制的系统创建程序。

在后面的章节和实验中,我们将着重于Allen-Bradley的SLC-500系列的PLC控制器。我相信,在试图了解所有的控制器之前,最好先掌握一种PLC控制器系列。我们将参照其他系列的控制器,如MicroLogix 1000, 1200, 1500以及ControlLogix 5000系列。但教学的重点和所有的实验都将基于SLC-500系列。

在准备本书以指导学生如何正确使用可编程序控制器的过程中,我们还设计制造了一个教学设备,用在课程的实验部分。这个设备安装在一个金属的运输箱内以保护部件,且便于存放(图I.2)。不结合实际的运动部件来讲PLC控制系统和编程无益于浪费时间。为了让学生体会实时系统编程,我们还在PLC箱的盖子上装了一个运动面板(图I.3)。箱盖上还装了5个气缸、5个气阀、一个空气调节器、9个接近开关以及一个带凸轮和凸轮开关的调速

(timing) 马达。在以后的章节中我们会经常提到这个实验箱。所有设计的实验和程序实例都可以在这个仿真装置上运行。如果选用本书作为教材, 建议你做一个类似的装置, 其电气图见附录A。

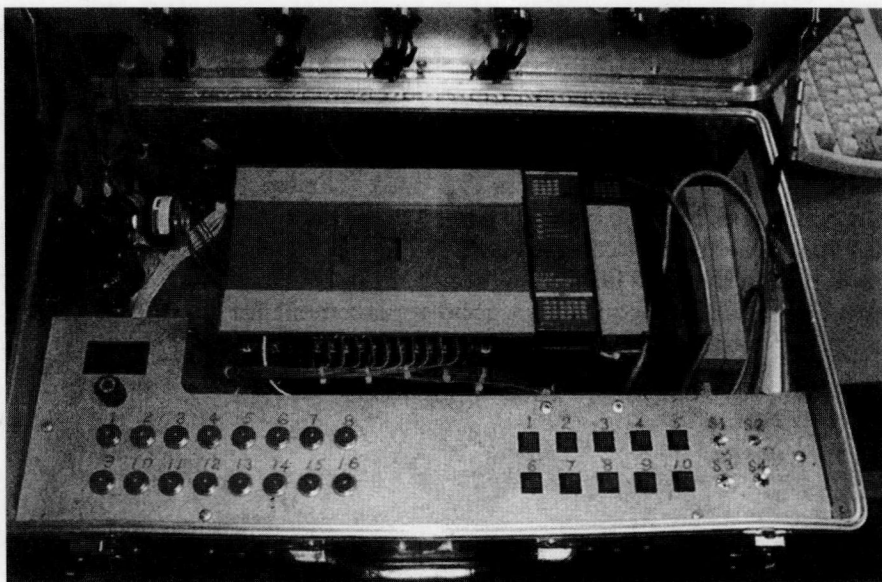


图1.2 PLC控制器教学实验箱

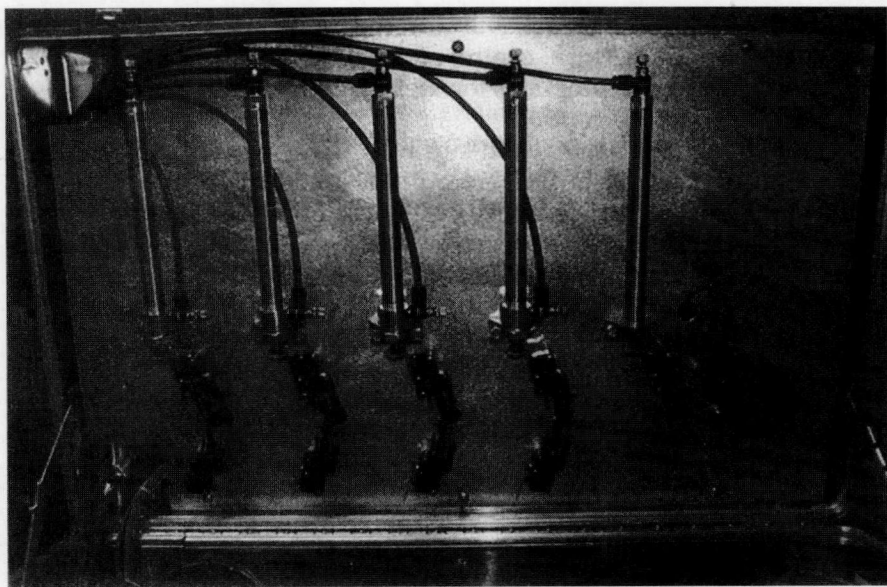


图1.3 PLC控制器教学设备包的运动面板

本书第1版出版后, Allen-Bradley还推出了MicroLogix 1200和1500。我们用的教学装置是基于SLC-500的, 但在新的教学装置中, 可以采用MicroLogix 1200系列的PLC。稍微修改一下I/O地址, 就可以在1200系列上运行本书所有的实验和程序。1762-L40BWA完全可以替代我们用的1747-L40E。详情请与作者联系, 邮件地址dgeller@oakton.edu。

目 录

第1章 二进制运算	1	5.3 SLC-500控制器系统	49
1.1 基本的二进制函数	2	5.4 结论	55
1.2 二进制逻辑函数	4	复习题	55
1.3 数字系统	7	第6章 PLC系统的设计过程	57
1.4 结论	8	6.1 与工程师进行协调	58
复习题	8	6.2 生成I/O清单	59
第2章 输入/输出接线	9	6.3 制作机器的功能图表	59
2.1 输入/输出符号	9	6.4 再次和工程师们协商	59
2.2 输入模块	10	6.5 设计电气系统并画出 布线图	60
2.3 输出模块	13	6.6 生成梯形图和HMI程序	60
2.4 系统设计	16	6.7 调试I/O布线	60
2.5 硬件接线	19	6.8 调试程序	61
2.6 初始化和调试	19	6.9 准备最后的文档	61
2.7 结论	20	6.10 结论	61
复习题	20	第7章 PLC软件开发	62
第3章 开关和位置传感器设备	21	7.1 RSLogix-500开发软件	62
3.1 开关设备	23	7.2 结论	73
3.2 电感式接近传感器	26	复习题	73
3.3 电容式接近传感器	29	实验设计	74
3.4 光电传感器	32	第8章 基本的机器控制功能	77
3.5 超声式传感器	34	8.1 气缸和气阀	77
3.6 结论	35	8.2 分度器驱动装置	81
复习题	37	8.3 假想的机器示例	84
第4章 梯形图的梯级逻辑	37	8.4 编程疑难	88
4.1 基本的梯形逻辑函数符号	37	8.5 结论	88
4.2 锁定和解锁输出类型	40	复习题	88
4.3 装置举例	41	实验设计	89
4.4 结论	44	第9章 定时器指令	92
复习题	44	9.1 TON指令	93
第5章 PLC模型、模式、I/O扫描 和存储配置	45	9.2 TOF指令	94
5.1 Allen-Bradley PLC系统	45	9.3 RTO指令	94
5.2 处理器系统的选择	48	9.4 在机器控制中使用 定时器指令	95

9.5 定时器的其他应用	98	13.9 结论	191
9.6 故障诊断与设置	100	复习题	192
9.7 结论	103	实验设计	192
复习题	104	第14章 移位寄存器、FIFO和 LIFO指令	196
实验设计	104	14.1 移位寄存器	196
第10章 计数器指令	108	14.2 FIFO指令	198
10.1 故障维修与调整	112	14.3 结论	201
10.2 程序举例	113	复习题	201
10.3 结论	114	实验设计	201
复习题	115	第15章 定序器的功能	203
实验设计	115	15.1 SQO指令	204
第11章 逻辑和数学指令	118	15.2 SQC指令	209
11.1 比较指令	118	15.3 SQL指令	218
11.2 逻辑和数据移动指令	122	15.4 结论	219
11.3 数学指令	125	复习题	219
11.4 数据文件指令	129	实验设计	220
11.5 特殊寻址模式	131	第16章 诊断程序	221
11.6 结论	134	16.1 诊断分级	221
复习题	135	16.2 错误检测编程	223
实验设计	135	16.3 例题	231
第12章 基本的机器控制编程	137	16.4 结论	245
12.1 控制程序的设计	138	复习题	245
12.2 机器举例	141	实验设计	246
12.3 结论	153	第17章 HMI显示系统	249
复习题	154	17.1 地址分配	250
实验设计	154	17.2 界面设计	251
第13章 连续运行的机器	160	17.3 开始面板构建项目	253
13.1 三态系统	160	17.4 报警条	267
13.2 装配线程序	163	17.5 修改例题	271
13.3 系统时序	163	17.6 结论	277
13.4 跟踪分度装配线上的 产品数据	170	复习题	278
13.5 装配线程序说明	171	实验作业	278
13.6 例题	176	附录 PLC教学实验箱的I/O列表 和连线图	279
13.7 异步工作台牵引装配线	189		
13.8 凸轮和编码器驱动机器	191		

第1章 二进制运算

学习目标

学完本章后，读者应该掌握：

- (1) 二进制系统
- (2) 二进制、十六进制和十进制系统间的相互转换
- (3) 逻辑函数(如与、或和非)
- (4) 二进制加法和减法
- (5) 带点分隔符的十进制记数法

要进行PLC系统的设计和编程，用户需要有一定的二进制运算基本知识。本书包含的基本二进制函数有加、减、与、或以及异或。我们还会涉及到十六进制和带点分隔符的十进制系统。

读者朋友应该很熟悉日常所用的十进制系统。它是一个位加权系统，数字中的每一位都是它右边这一位的十倍。在十进制系统中我们用十个不同的符号来代表这一位的值(0到9)。例如，图1.1给出的3526这个数可以如图所示按位进行分解。

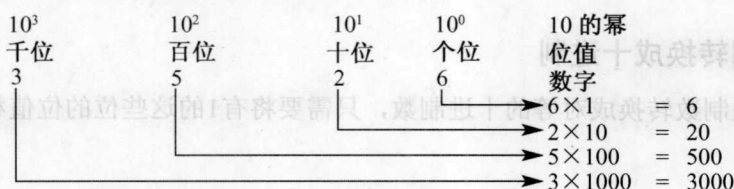


图1.1 十进制中10的幂

在二进制系统中，同样有位加权系统。二进制系统是基于2的幂，所以每一位的值是它右边这一位的2倍(见图1.2)。二进制系统只有两个符号：0和1。二进制常用于计算机，因为数字系统只有两种状态。关状态用0V表示，即二进制中的0。开状态用5V表示，即二进制中的1。二进制计数级数的前4位如图1.3所示。

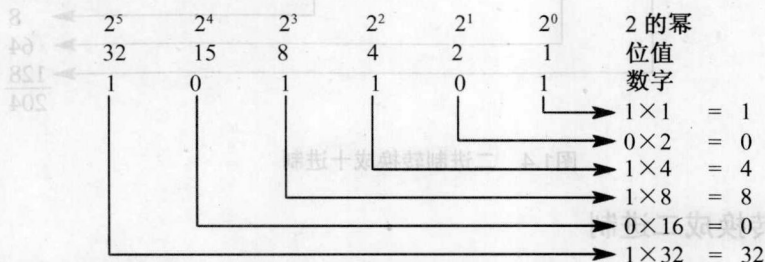


图1.2 二进制系统

二进制位的值				十进制	
8	4	2	1		
0	0	0	0	=	0
0	0	0	1	=	1
0	0	1	0	=	2
0	0	1	1	=	3
0	1	0	0	=	4
0	1	0	1	=	5
0	1	1	0	=	6
0	1	1	1	=	7
1	0	0	0	=	8
1	0	0	1	=	9
1	0	1	0	=	10
1	0	1	1	=	11
1	1	0	0	=	12
1	1	0	1	=	13
1	1	1	0	=	14
1	1	1	1	=	15

图1.3 0到15的二进制级数

1.1 基本的二进制函数

1.1.1 二进制转换成十进制

将一个二进制数转换成对等的十进制数，只需要将有1的这些位的位值相加，如图1.4所示。

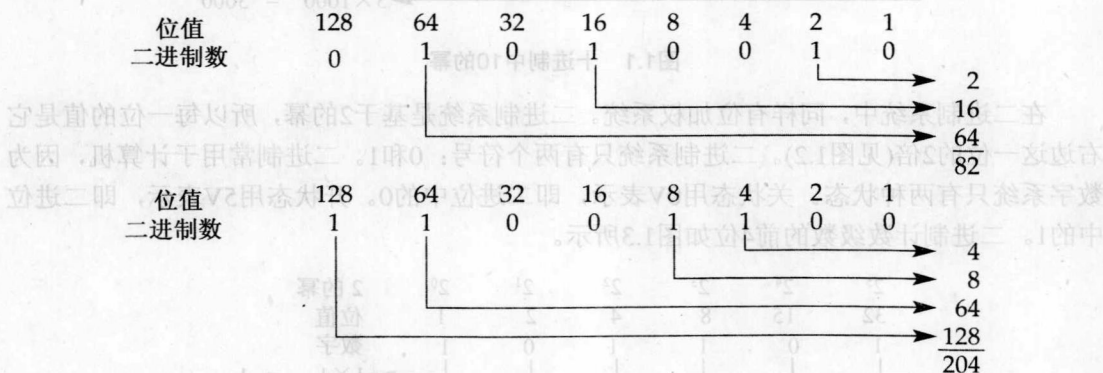


图1.4 二进制转换成十进制

1.1.2 十进制转换成二进制

十进制换成二进制有好几种方法，这里只介绍其中的一种。将十进制转换成二进制时，先将二进制的位值逐渐增大，直到位值大于要转换的十进制数为止。然后从十进制数中减去所允许的最高的二进制位值。对于每一个能减去的位值，就在相应的位上置1。对于每一个不能减去的位值，就在相应的位上置0。图1.5给出了一个例子。

注意，像十进制一样，在二进制中，最后的数字也省略前面的零。一种快速检查答案

的方法是看它是奇数还是偶数。如果转换的十进制数是奇数，那么二进制数最小的有效位应该为1；如果你转换的十进制数是偶数，那么二进制数最小的有效位应该为0。

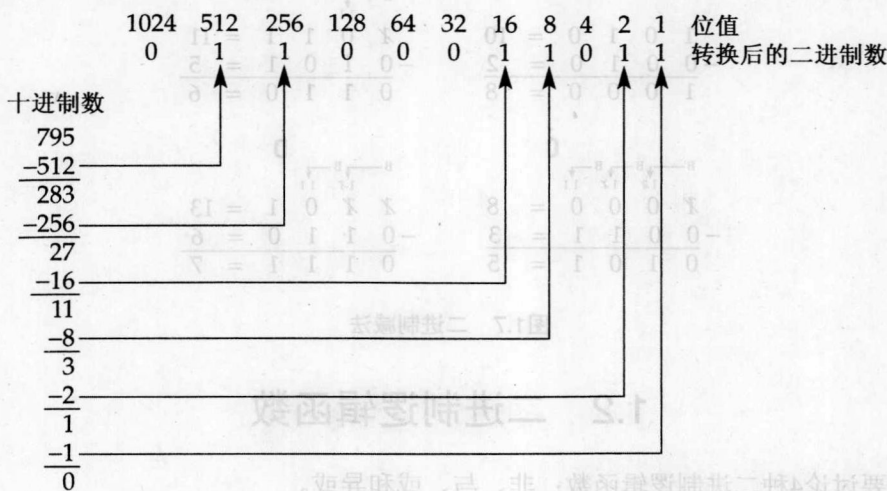


图1.5 十进制转换成二进制

1.1.3 二进制加法

在十进制加法中，如果位值的和大于9，就会产生一个进位。在二进制加法中，如果位值的和大于1，就会产生一个进位。图1.6展示了几个例子。

注意，在例子D中，第二位等于3，所以转化为一个进位1，并在和中保留一个1。

A					B				
1	0	0	1	= 9	1	0	1	1	= 11
+0	1	1	0	= 6	+0	1	1	0	= 6
1	1	1	1	= 15	1	0	0	0	= 17

C					D				
0	1	0	1	= 5	0	0	1	1	= 3
0	1	1	0	= 6	1	0	1	1	= 11
+1	0	1	1	= 11	+1	1	1	0	= 14

图1.6 二进制加法

1.1.4 二进制减法

在十进制减法中，当减数的位值大于它被减数上对应的位值时，可以从它的左边位借一个10。在二进制减法中，我们同样可以从左边位借位，但借的值是2。图1.7展示了几个例子。

例子A中没有借位。例子B的第三位有一个借位，注意，有两个1从第四位借到第三位。一个1被减去，还有一个1写到这一位的差中。例子C中第一位就需要借位，但只能从第四位借。注意，每个借位都给其右边的位两个1，其中的一个1又被借给下一位，直到第一位中有两个1。第一位从借来的两个1中减去一个1，剩下的一个1移到这一位的差中。第二位

从借来的保留的一个1中减去1, 所以剩下0移到这一位的差中。

$$\begin{array}{r}
 \text{A} \\
 \begin{array}{r}
 1\ 0\ 1\ 0 = 10 \\
 -0\ 0\ 1\ 0 = 2 \\
 \hline
 1\ 0\ 0\ 0 = 8
 \end{array} \\
 \\
 \text{B} \\
 \begin{array}{r}
 \begin{array}{c} \text{B} \rightarrow \\ 1 \end{array} \\
 \begin{array}{r}
 \cancel{1}\ 0\ 1\ 1 = 11 \\
 -0\ 1\ 0\ 1 = 5 \\
 \hline
 0\ 1\ 1\ 0 = 6
 \end{array}
 \end{array} \\
 \\
 \text{C} \\
 \begin{array}{r}
 \begin{array}{c} \text{B} \rightarrow \text{B} \rightarrow \text{B} \rightarrow \\ 1\ \cancel{1}\ \cancel{1}\ 1 \end{array} \\
 \begin{array}{r}
 \cancel{1}\ 0\ 0\ 0 = 8 \\
 -0\ 0\ 1\ 1 = 3 \\
 \hline
 0\ 1\ 0\ 1 = 5
 \end{array}
 \end{array} \\
 \\
 \text{D} \\
 \begin{array}{r}
 \begin{array}{c} \text{B} \rightarrow \text{B} \rightarrow \\ 1\ \cancel{1}\ 1 \end{array} \\
 \begin{array}{r}
 \cancel{1}\ \cancel{1}\ 0\ 1 = 13 \\
 -0\ 1\ 1\ 0 = 6 \\
 \hline
 0\ 1\ 1\ 1 = 7
 \end{array}
 \end{array}
 \end{array}$$

图1.7 二进制减法

1.2 二进制逻辑函数

本书要讨论4种二进制逻辑函数: 非、与、或和异或。

1.2.1 “非”函数(反码)

图1.8给出的符号是非门或反码逻辑函数。逻辑符号的右边是真值表, 其中显示了每种输入状态对应的输出状态。

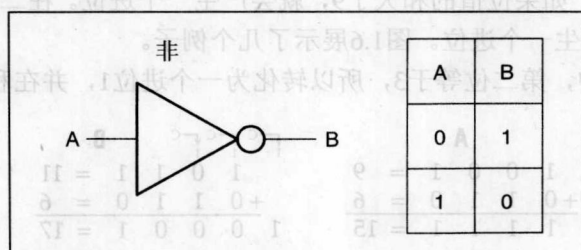


图1.8 非门(反码)的逻辑符号及其真值表

反码函数用于二进制运算。图1.9的例子说明一个寄存器内容的反码。要找到任何一个二进制数的反码, 必须对该数的每一位取反。

寄存器 X	01100101	10111100
反码	10011010	01000011

图1.9 反码

1.2.2 “与”(AND)函数

图1.10展示了“与”的逻辑符号及其真值表。该逻辑符号有两个输入(A和B)和一个输出C。从真值表可以看出, 只有A和B都为1时, C才为1。与前面一样, 真值表给出了所有可能的输入和输出状态。

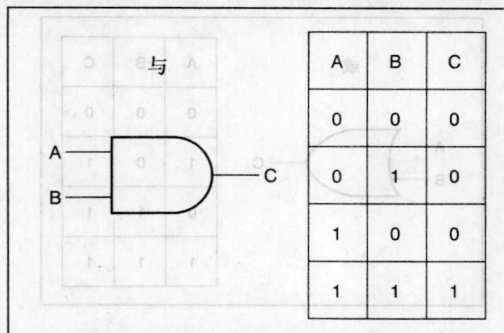


图1.10 “与”逻辑符号及其真值表

“与”函数在逻辑设计中可以多于两个输入，但只有当所有的输入为1时，输出才为1。“与”函数也用于二进制运算。编程人员通常用这个函数将寄存器的一部分清零，而保留寄存器中其余部分的内容。也就是说，我们可以将寄存器的某一位或一些位清零，但不影响寄存器中的其余位。任何与1相“与”的位保持不变，与0相“与”的位变为0。考虑图1.11所示的例子。



图1.11 二进制与函数

从这两个例子可以发现，左边这个例子是将寄存器X中的低4位清零，保留高4位不变。右边这个例子只是将单个的位(3)清零，其余7位保持不变。

“与”函数的另一个用途是测试寄存器中的某一个位是否为1。这个函数的使用方法是，将寄存器与一个除被测试位外其余位全为0的字进行“与”运算，在被测试位置置1。如果最后结果全为0，说明这一位没被置位，如果结果不为0，则说明这一位被置位，图1.12给出了一个例子。



图1.12 与函数用作位测试

左边的例子中，位4被置位，结果不是0。右边的例子中，位4没被置位，结果为0。检查执行AND指令后结果是否为0，就可以确定有疑问的位是否被置位。

1.2.3 “或”(OR)函数

图1.13给出的符号是“或”的逻辑函数。它有两个输入A和B，一个输出C。从真值表可知，当A或B中的一个或两个同时为1时，输出C为1。真值表展示了这种逻辑函数所有可能的输入和输出情况。

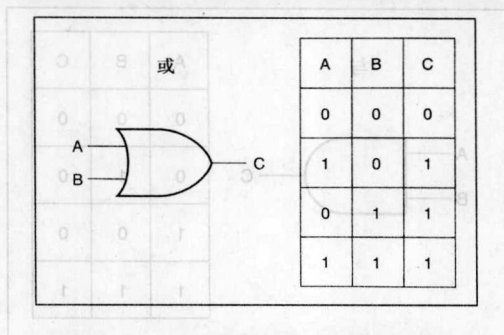


图1.13 “或”逻辑符号及其真值表

在逻辑设计中，“或”函数也可以有两个以上的输入。不管有多少个输入，输入中任何一个为1，输出就为1。“或”函数在二进制运算中用于将寄存器中的一个位或一组位置位，但不改变寄存器中的其余位。这个逻辑函数的几个例子如图1.14所示，任何与1相“或”的位变为1，任何与0相“或”的位保持不变。

	置位位	置位位
寄存器 X	10110010	01001100
OR 函数	00001100	00100000
寄存器 X 中的结果	10111110	01101100

图1.14 “或”逻辑函数的例子

在这些例子中，左边的例子有2个位被置位，右边的例子有1位被置位。

1.2.4 “异或”(XOR)函数

图1.15展示了“异或”的逻辑符号以及真值表。注意，除了最后这个状态，其余的与“或”函数是一样的。如果A或B为1，但不是两个同时为1时，“异或”的输出为1。

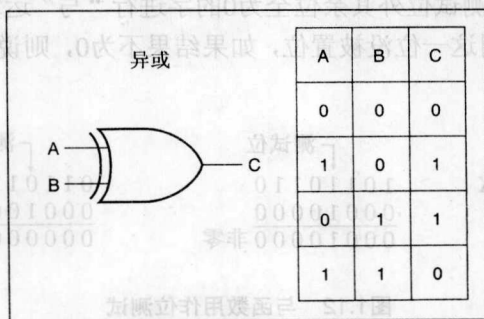


图1.15 “异或”逻辑符号及其真值表

异或逻辑函数在二进制运算中用来比较两个寄存器的值是否相等。例如有一个计数寄存器，我们想知道什么时候计数值到达预置值，这时就可以用“异或”函数。如果把计数器的值与预置值相“异或”，结果为0，那就表明两个寄存器的值相等；其他任何值都表示两个值不等。图1.16给出了两个例子。

寄存器 X	10110010	01001100
XOR 函数	10001100	01001100
结果	00111110	00000000
	相等	

图1.16 “异或”逻辑函数的例子

1.3 数字系统

在使用二进制数时，大多数人都害怕面对一长串的1和0。PLC中的大部分地址都是16位的。假设你想要同事看某一个PLC地址，并大声地说出下列二进制数字：

1001 0111 0011 1010

和大部分人一样，你会觉得口头表达这么长的二进制数非常困难。鉴于这个原因同时也为了方便地显示二进制数，工程师们开发了一种可选的数字系统，即十六进制。很重要的一点是，这种可选的数字系统并不改变二进制数的值，它只是更易于数字的显示和表达。

1.3.1 十六进制系统

在十六进制系统中，所有的二进制数按4位一组划分，然后每一组赋一个单个的数值。请看图1.17显示的十六进制数字级数。注意，9以后就变成了字母字符。

要把一个二进制数变为十六进制，首先将这个二进制数按4位一组进行分组，然后给每一组赋一个十六进制的值。图1.18展示了几个例子。

二进制	十六进制	十六进制	二进制
0000=0		1000=8	
0001=1		1001=9	
0010=2		1010=A	
0011=3		1011=B	
0100=4		1100=C	
0101=5		1101=D	
0110=6		1110=E	
0111=7		1111=F	

图1.17 二进制和十六进制转换表

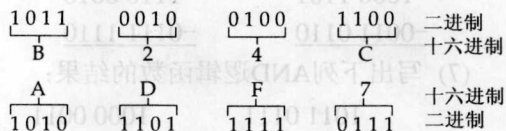


图1.18 二进制与十六进制之间的转换

我想再一次强调，二进制的值并没有改变，它只不过采用了一种更清晰的表示方法。

要将十六进制数转换回二进制，逆转这个过程，将每个十六进制数转换成对应的二进制位形式即可。

1.3.2 带点分隔符的十进制记数法

随着对PLC系统的深入了解，肯定会遇到需要将几个PLC系统连接起来的情况。让几个PLC系统相互通信的一种方法是通过Ethernet(以太网)将它们连在一起。每个以太网设备都需要一个独立的IP地址。IP地址是32位的二进制数，通常用带点分隔符的十进制记数法表示。用带点分隔符的十进制记数法来表示一个32位的二进制数时，要先将这个二进制数拆成4个8位二进制组，然后每8位转换成0到255之间的一个十进制数。4个十进制数之间用句点分开(见图1.19)。网络的IP地址经常需要在二进制和带点分隔符的十进制之间相互转换。

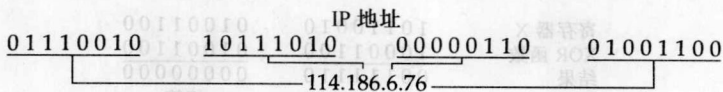


图1.19 带点分隔符的十进制记数法

1.4 结 论

结束本章时,我想声明一下,本章并没有完全讨论所有二进制数学的内容。如果希望更全面地了解,有很多这方面的专著可读。这一章对于以前学过二进制数学的读者来说,权当是复习;对于第一次接触这方面知识的读者而言,只是一个基本的起点。

复 习 题

- (1) 将下列二进制数转换为十进制: 1011 0110 0111 0101 1010 0011
- (2) 将下列十进制数转换为二进制: 1592 279 763
- (3) 将下列二进制数转换为十六进制: 1010 0111 0101 1011 1000 0011 1111 1100
- (4) 将下列十六进制数转换为二进制: FA67 EDA7
- (5) 对下列二进制数进行相加:

$$\begin{array}{r} 1000\ 0110 \\ +0110\ 0101 \\ \hline \end{array} \quad \begin{array}{r} 0111\ 1101 \\ +1001\ 1011 \\ \hline \end{array}$$

- (6) 对下列二进制数进行相减:

$$\begin{array}{r} 1000\ 1101 \\ -0011\ 0110 \\ \hline \end{array} \quad \begin{array}{r} 1110\ 0010 \\ -0111\ 1110 \\ \hline \end{array}$$

- (7) 写出下列AND逻辑函数的结果:

$$\begin{array}{l} 1011\ 0111 \\ \text{AND } 0000\ 1111 \end{array} \quad \begin{array}{l} 1000\ 0011 \\ \text{AND } 1111\ 1101 \end{array} \quad \begin{array}{l} 1010\ 0011 \\ \text{AND } 0010\ 0000 \end{array}$$

- (8) 写出下列OR逻辑函数的结果:

$$\begin{array}{l} 1100\ 0011 \\ \text{OR } 0001\ 0000 \end{array} \quad \begin{array}{l} 1110\ 0000 \\ \text{OR } 0000\ 1111 \end{array} \quad \begin{array}{l} 1010\ 1010 \\ \text{OR } 0000\ 0101 \end{array}$$

- (9) 写出下列XOR逻辑函数的结果:

$$\begin{array}{l} 1010\ 0011 \\ \text{XOR } 1010\ 0011 \end{array} \quad \begin{array}{l} 1110\ 0110 \\ \text{XOR } 1111\ 0101 \end{array} \quad \begin{array}{l} 1101\ 0011 \\ \text{XOR } 1101\ 0011 \end{array}$$

- (10) 将下列二进制数转换成带点的十进制记数法:

$$11001010\ 10101110\ 10000011\ 01110101$$

- (11) 将下列带点分隔符的十进制数转换成二进制的形式:

$$245.134.85.18$$