

# PLC

## 编程和故障排除

韩兵 编



化学工业出版社

# PLC

## 编程和故障排除

韩兵 编



化学工业出版社

北京

图书在版编目 (CIP) 数据

PLC 编程和故障排除/韩兵编. —北京: 化学工业出版社, 2008. 12

ISBN 978-7-122-03794-7

I. P… II. 韩… III. 可编程序控制器-程序设计  
IV. TP332.3

中国版本图书馆 CIP 数据核字 (2008) 第 150988 号



---

责任编辑: 宋 辉  
责任校对: 顾淑云

文字编辑: 王 洋  
装帧设计: 张 辉

---

出版发行: 化学工业出版社 (北京市东城区青年湖南街 13 号 邮政编码 100011)

印 刷: 北京云浩印刷有限责任公司

装 订: 三河市宇新装订厂

787mm×1092mm 1/16 印张 13 字数 318 千字 2009 年 1 月北京第 1 版第 1 次印刷

---

购书咨询: 010-64518888 (传真: 010-64519686) 售后服务: 010-64518899

网 址: <http://www.cip.com.cn>

凡购买本书, 如有缺损质量问题, 本社销售中心负责调换。

---

定 价: 30.00 元

版权所有 违者必究

# 前言

PLC（可编程控制器）技术与 PLC 控制系统在工业生产与民用工程自动化应用中占支配地位。随着 PLC 产品不断更新和功能不断提高，各种类型、不同功能和大小规模的 PLC 控制系统在各个领域得到了更广泛的应用。进入 21 世纪以来，几乎所有的工业产品都离不开自动化技术，例如，在运动控制过程中无论是选择哪些变量进行监控，都可以采用光电 PLC 控制系统来实现。由于可编程控制器具有数字控制、网络通信和抗干扰能力强的优点，PLC 成为大小控制系统的核心部件。PLC 在航空航天工程、能源工程、机器人系统、生产流水线、制造过程、交通车辆控制、工程测量系统、安全监控系统、环境工程中起着实时控制、测量转换、采集数据传送和网络信息处理等功能。因此，PLC 的正常可靠工作对工业生产和居民生活至关重要。

PLC 应用对象广泛，PLC 控制系统结构复杂，全球重要企业的 PLC 产品众多，从事控制系统应用的工程技术人员开发、应用和维护 PLC 有很多问题需要解决。大多数 PLC 应用工程师是由企业进行专门的培训来掌握 PLC 的，而这往往只针对单一系统产品。为了帮助现场工程师和技术人员进一步掌握 PLC 技术和应用，我们编写了此书。全书包括 PLC 编程技术、先进的小规模 PLC、PLC 控制系统应用技术、PLC 系统通信应用技术、PLC 安装与在线维护、PLC 故障诊断与故障排除等内容。

袁洪基、袁振明也参加了本书编写的部分工作，在此表示感谢，并对提供资料的单位和个人表示感谢。

由于编者水平有限，时间仓促，不足之处在所难免，恳请读者批评指正。

编者

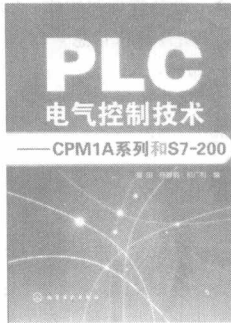
## PLC 电气控制技术——CPM1A 系列和 S7-200

夏田、陈婵娟、祁广利 编

本书从实际工程应用需要出发,以广泛应用的 OMRON 高性能小型机 CPM1A 以及 SIEMENS 公司高性能小型机 S7-200 为背景,系统讲述了可编程程序控制器的基本组成原理、工作方式、系统配置、PLC 指令系统与编程、PLC 控制系统的设计方法和应用,简要介绍了 CX-P 编程软件、PLC 的网络通信知识并介绍了电气控制的基础知识。相关章节附有习题及思考题。内容取材新颖,由浅入深,循序渐进,理论与实际相结合。

本书可作为高等院校工业自动化、电气工程及其自动化、机电一体化、机械设计制造及其自动化、数控技术等本科专业的教材,也可供相关工程技术技术人员参考。

(书号:978-7-122-01663-8, 16 开, 33 元)



## 触摸屏技术及应用

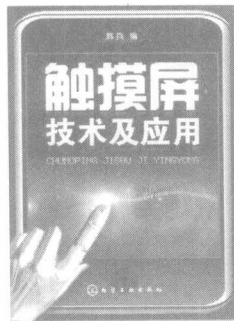
韩兵 编

触摸屏是一种最新的电脑输入设备,它可以让用户只用手指轻轻地碰计算机显示屏上的图符或文字,就能实现对主机的操作,这样摆脱了键盘和鼠标操作,使人与机交互更为直截了当。它具有坚固耐用、反应速度快、节省空间、易于交流等优点。

本书主要介绍了触摸屏的工作原理、软件设计、通信、触摸屏 PLC 控制系统等内容,在此基础上,向读者介绍了几种常用的触摸屏产品,给用户选择产品提供参考,最后以触摸屏在不同行业中的应用实例向读者完整地介绍了触摸屏的使用情况。

本书可供触摸屏系统开发、应用的人员阅读,也可作为高等院校电类、机电一体化专业的教学参考。

(书号:978-7-122-02961-4, 16 开, 32 元)



## PLC 技术及应用

刘光起、周亚夫 主编

本书介绍了小型可编程控制器的基本结构及工作原理,对应用较广泛的松下、三菱和西门子三种类型的可编程控制器的指令系统、编程方法和技巧作了较为详细的介绍,同时选取了大量的典型控制系统实例进行系统分析和程序设计,为 PLC 控制系统分析和设计提供了思路和方法。

为了使初学者能更快更好地掌握 PLC 的应用,在编写时力求由浅入深、通俗易懂,减少了纯理论性的分析介绍,注重实用性。

本书适于 PLC 的入门级读者学习使用,也可作为中、高职院校相关专业的教学参考书。

(书号:978-7-122-01220-3, 大 32 开, 18 元)



## 人机界面设计与应用

李方园 编

人机界面是系统和用户之间进行交互和信息交换的媒介,它实现信息的内部形式与人类可以接受形式之间的转换。

本书先从人机界面的基本概念讲起,理论上由浅入深,使读者对人机界面有一个较全面的认识。接着对人机界面在轻工业、机械工业、流程工业、公用设备以及电子仪器产品中的应用做了大量的案例分析,细致实用。最后,还针对不同品牌的人机界面系统进行了应用答疑,给出了作者的应用经验和实践技巧。

本书适用于从事人机界面产品设计、开发、使用与维护的技术人员,也可作为相关院校自动化专业学生的教学参考用书。

(书号:978-7-122-01401-6, 16 开, 36 元)



<b>第 1 章 PLC 编程技术</b> .....	1
1.1 PLC 技术概述 .....	1
1.2 PLC 的梯形图编程 .....	2
1.2.1 PLC 编程基本知识 .....	2
1.2.2 梯形图指令 .....	20
1.3 PLC 的指令集编程 .....	29
1.3.1 标准触点指令实现 .....	30
1.3.2 与堆栈有关的指令实现 .....	31
1.3.3 立即触点指令实现 .....	32
1.3.4 输出指令及取反指令实现 .....	33
1.3.5 定时器功能及指令实现 .....	34
1.3.6 计数器功能及指令实现 .....	38
1.4 PLC 编程顺序与状态图 .....	40
1.4.1 PLC 顺序功能图 .....	40
1.4.2 顺序功能图转换实现规则 .....	40
1.4.3 使用启保停电路的编程方法 .....	41
1.5 先进的 PLC 编程应用软件 .....	41
1.5.1 欧姆龙 CX-Programmer 编程软件 .....	41
1.5.2 西门子 STEP 7 编程软件 .....	47
<b>第 2 章 先进的小规模 PLC</b> .....	53
2.1 新的灵巧 PLC 技术 .....	53
2.2 生产包装喷涂 PLC 应用实例 .....	56
2.2.1 喷涂流量控制系统 .....	56
2.2.2 安全性仪表 (阀岛) .....	56
2.2.3 现场总线控制仪表 .....	57
2.2.4 生产过程 PLC 控制 .....	57
2.3 交通灯控制 PLC 应用实例 .....	59
2.3.1 红绿灯 PLC 控制程序编制 .....	60
2.3.2 红绿灯控制梯形图程序 .....	63
<b>第 3 章 PLC 控制系统应用技术</b> .....	65
3.1 大规模 PLC 控制系统 .....	65
3.1.1 大型 PLC 的技术特点 .....	65
3.1.2 大型 PLC 的网络通信 .....	65
3.2 PLC 控制系统应用技术 .....	67
3.2.1 集散控制系统 PLC 应用技术 .....	67

3.2.2	现场总线控制系统 PLC 应用技术	68
3.3	PLC 控制的组态应用技术	70
3.3.1	监控组态软件建立	70
3.3.2	PLC 控制系统的设备组态	72
3.4	PLC 热风炉控制应用实例	74
3.4.1	热风炉工作原理	74
3.4.2	热风炉控制系统的硬件设计方案	77
3.4.3	PLC 控制系统软件设计	80
3.5	PLC 机床加工控制应用实例	93
<b>第 4 章</b>	<b>PLC 系统通信应用技术</b>	<b>105</b>
4.1	PLC 通信系统构成	105
4.2	PLC 高级通信协议应用	106
4.2.1	PLC 通信方法	106
4.3	PLC 各种通信应用的建立	108
4.3.1	计算机 PLC 串行通信	108
4.3.2	PLC 实时监控软件	110
4.4	轴承检测 PLC 控制系统应用实例	112
4.4.1	检测和分拣系统结构	112
4.4.2	轴承部件匀速测量方案	112
4.4.3	轴承部件内外径测量方案	114
4.4.4	LS-7000 激光测量传感器	114
4.4.5	LS-7000 与 PLC 通信	118
4.5	电厂除灰 PLC 监控网络应用实例	119
4.5.1	集散控制系统应用	120
4.5.2	除灰 PLC 集散控制系统构建	121
4.5.3	PLC 控制系统配置	124
4.5.4	PLC 控制系统设计	126
4.5.5	PLC 电气连接与编程	130
<b>第 5 章</b>	<b>PLC 安装与在线维护</b>	<b>135</b>
5.1	PLC 的结构与功能	135
5.1.1	PLC 的模块化结构	135
5.1.2	PLC 组成模块的功能	135
5.2	PLC 控制系统安装运行	136
5.2.1	PLC 电源配置	136
5.2.2	PLC 模块的安装与拆卸	137
5.2.3	PLC 接地和接线	138
5.2.4	PLC 抑制电路的设计	139
5.3	PLC 控制系统的在线维护	140

5.3.1	在线 PLC 软件调试维护	140
5.3.2	PLC 控制系统的日常维护	143
5.4	PLC 控制系统的紧急停车	145
5.4.1	紧急停车系统的基本原理	145
5.4.2	PLC 紧急停车控制系统	149
<b>第 6 章</b>	<b>PLC 故障诊断与故障排除</b>	<b>153</b>
6.1	PLC 应用的故障报警	153
6.1.1	PLC 应用的错误信息	153
6.1.2	PLC 应用的故障诊断处理	154
6.2	PLC 一般故障的检查与处理	156
6.2.1	PLC 一般故障现象及修复办法	156
6.3	PLC 控制系统故障诊断	160
6.3.1	PLC 控制系统故障分析	160
6.3.2	纸机 PLC 故障的判断和处理	160
6.4	PLC 控制系统故障排除流程	162
6.4.1	PLC 总体诊断故障排除	162
6.4.2	PLC 电源诊断故障排除	162
6.4.3	PLC 运行诊断故障排除	162
6.4.4	PLC 输入/输出故障诊断排除	164
6.5	汽车涂装系统故障诊断排除实例	166
6.5.1	汽车生产涂装线功能和结构	166
6.5.2	汽车涂装线电气控制	169
6.5.3	故障树分析在设备故障中的应用	170
6.5.4	动作流程图在设备故障分析中的应用	173
6.5.5	复杂 PLC 系统的故障分析	174
6.5.6	设备故障分类及原因分析	176
6.5.7	涂装线设备故障排除	179
6.5.8	PLC 现场总线系统故障排除	187
6.6	基于 PLC 的小型水电站系统故障监测实例	188
6.6.1	水电站可控硅励磁系统原理	188
6.6.2	PLC 励磁调节器的总体结构	189
6.6.3	发电机励磁系统故障与排除	190
6.6.4	水电站励磁系统故障监测系统	192
	<b>参考文献</b>	<b>197</b>



# 第1章

## PLC 编程技术

### 1.1 PLC 技术概述

可编程逻辑控制器 (programmable logic controller, PLC), 简称可编程控制器, 是以嵌入式微处理器为核心, 具有数字逻辑或模拟输入输出模块, 专为工业和工程复杂环境而设计的数字控制装置。经过长期的实际应用, 它已经成为技术通用和标准化的控制器, 是综合了微计算机技术、自动化技术和网络通信技术的新一代工业产品。它采用了专门设计的模块化硬件结构, 其控制功能通过执行控制程序来完成, 具有可靠性高、适应工业现场的高温、高湿度、冲击和振动等恶劣环境的特点, 是机械制造控制、化工过程控制和能源工程控制等工业控制应用最普遍使用的工具, 在工业自动化、民用与环境工程领域得到了广泛的应用。随着可编程控制器的发展, 它不仅能完成编辑、逻辑控制和数字通信, 而且能实现模拟量与数字量的相互转换。可编程控制器不但具有存储程序的存储器, 还在内部对数据进行存储。它可执行逻辑运算、顺序控制、定时、记数和算术操作的指令, 通过数字量或模拟量的输入输出来控制各种类型的机械设备或生产过程; 还具有液晶显示功能, 通过触摸屏可实现人机对话, 设定控制系统的参数和状态。

可编程控制器 (PLC) 的工作基本原理是采用程序扫描技术来实现逻辑控制功能。扫描是一种形象化的术语, 用来描述可编程序控制器内部的 CPU 的工作过程。所谓扫描就是依次对各种规定的操作项目全部进行访问和处理。PLC 运行时, 用户程序中有众多的操作需要执行, 但是一个 CPU 每一个时刻只能执行一个操作而不能同时执行多个操作, 因此 CPU 按程序的顺序依次执行各个操作。这种在处理多个作业时依次按顺序处理的工作方式称为顺序扫描工作方式。由于扫描是周而复始、无限循环的, 每扫描一个循环所用的时间即为扫描周期。顺序扫描的工作方式是 PLC 的基本工作方式, 它简单直观, 方便用户程序设计, 为 PLC 的可靠运行提供了有利保证。一方面, 所扫描的指令被执行后其结果马上就可以被后面将要扫描的指令所利用; 另一方面, 还可以通过 CPU 设置定时器来监视每次扫描时间是否超过规定时间, 避免由于 CPU 内部故障使程序执行进入死循环。

PLC 的工作过程基本上是用户的梯形图程序的执行过程, 即在系统软件的控制下顺次扫描各输入点的状态, 按用户程序解算控制逻辑, 然后顺序向各个输出点发出相应的控制信号。除此之外, 为提高工作的可靠性和及时地接收外来的控制命令, 每个扫描周期还要进行故障自诊断和处理、与编程器或计算机的通信请求。因此, PLC 工作过程分为以下五步。

① 自诊断 自诊断功能可使 PLC 系统在发生故障时能尽快地修复, 为此 PLC 每次扫描用户程序以前都对 CPU、存储器、输入/输出模块等进行自诊断, 若自诊断正常便继续进行扫描, 而一旦发现故障或异常现象, 则转入处理程序, 保留现行工作状态, 关闭全部输出, 然后停机并显示出错的信息。

② 网络通信 自诊断正常后 PLC 就扫描编程器、上位机等通信接口，如有通信请求便响应处理。在与编程器通信过程中，编程器把指令和修改的参数发送给主机，主机把要显示的状态、数据、错误码进行相应指示，编程器还可以向主机发送运行、停止、清内存等监控命令。在与上位机通信过程中，PLC 将接收上位机发出的指令，并进行相应的操作，把现场状态、PLC 的内部工作状态、各种数值参数发送给上位机，并执行启动、停机、修改参数等命令。

③ 输入现场状态 完成上两步工作后 PLC 便扫描各个输入点，读入各点的状态和数据，如开关的通断状态及形成现场的内存映像。这一过程也称为输入采样或输入刷新。在一个扫描周期内，内存映像的内容不变，即使外部实际开关状态已经发生了变化，也只能在下一个扫描过程中的输入采样时刻进行刷新，解算用户逻辑所用的输入值是各输入点的内存映像值，而不是当时现场的实际值。



图 1-1 PLC 的扫描工作过程

④ 解算用户逻辑 解算用户逻辑即执行用户程序。一般是从小地址的最低地址存放的第一条程序开始，在无跳转的情况下按存储器地址的递增方向顺序地扫描用户程序，按用户程序进行逻辑判断和算术运算，因此称为解算用户逻辑。解算过程中所用的计数器、定时器、内部继电器等编程元件的内数据为相应存储单元的即时值，而输入继电器、输出继电器则用的是内存映像值。在一个扫描周期内，不管外部实际情况是否已经变化，某个输入信号的状态对整个用户程序是一致的，不会造成结果混乱。

⑤ 输出结果 将扫描过程中解算出的最新结果送到输出模块，取代前一次扫描解算的结果，称为输出刷新。解算用户逻辑时，每一步所得到的输出信号被存入输出信号寄存表，并未发送到输出模块，相当于输出信号被输出门阻隔，待全部解算完成后打开输出门并输出，所用输出信号由输出状态表送到输出模块，输出模块触发开关动作。在依次完成上述五个步骤操作后，PLC 又开始进行下一次扫描，如此不断的反复循环扫描，实现对全过程及设备的连续控制，直至接收到停止命令、停电或出现故障为止。PLC 的扫描工作过程如图 1-1 所示。

## 1.2 PLC 的梯形图编程

### 1.2.1 PLC 编程基本知识

设计 PLC 程序有三种方法：经验设计法、继电器电路图转换法和顺序控制设计法。

经验设计法在一些典型电路的基础上，根据被控对象对控制系统的具体要求，不断修改、完善和调试，最后得到一个满意的结果。这种设计方法没有规律可循，最后的结果不是唯一的，设计所用的时间和设计质量与设计者的经验有很大的关系，该方法一般只用于较简单的梯形图设计。

继电器电路图转换法用于将继电器控制改造为 PLC 控制。由于原有的继电器控制系统经过长期的使用和考验，已被证明能完成系统要求的控制功能，因此可以将继电器电

路图翻译成梯形图，即用 PLC 的外部硬件接线图和梯形图程序实现继电器控制系统的功能。

顺序控制设计法是建立在顺序功能图基础上的、标准化的编程方法。用它设计出的程序很容易调试、修改和阅读。对于控制要求很复杂的系统，使用该方法可以节约大量的时间，获得满意的设计效果。

### (1) 基本步骤

编写程序可分为以下九个基本步骤。可以复制输入/输出 I/O 分配表和程序代码表中给出的表格来帮助编程。

① 首先列出一个所有 I/O 设备和已经分配给它们的 I/O 点的目录，并且准备一张写出分配给每一个 I/O 设备 I/O 位的表格。

② 如果使用 LR 位链接两个 PLC，那么准备一张写出这些使用的位的表格，LR 位为 PLC 的指定 I/O 内存位。

③ 确定工作位用什么字，并准备一张使用工作位分配的表格。

④ 同样准备一张 TC 编号和跳转编号的表格，这样在使用时就能够分配它们。一个 TC 编号在程序中仅能定义一次；跳转编号 01~99 在每个程序中也只能使用一次。TC 编号为定时器/计数器编号数据区。

⑤ 画梯形图。

⑥ 将程序输入到 CPU 单元。当使用手持编程器时，需将梯形图程序转换为助记符程序形式。

⑦ 检查程序有无语法错误，若有，需更正错误。

⑧ 运行该程序以检查是否存在运行错误，若有，需更正错误。

⑨ 当整个控制系统安装好并准备使用时，运行该程序并按要求仔细调试。

梯形图和转换为助记代码的基础知识在基本指令梯形图中描述。用手持编程器输入程序的有关说明可查阅 PLC 使用手册，如欧姆龙 PLC 的 CPM1、CPM1A、CPM2A、CPM2C 操作手册和 SRM1 主机控制单元手册，若通过 SSS 输入程序，则可参考 SSS 操作手册。

调试说明见 CPM1、CPM1A、CPM2A、CPM2C 操作手册、SRM1 主机控制单元手册和 SSS 操作手册。

### (2) 指令术语

有两类基本指令用于梯形图编程：一类指令用来代表梯形图上的条件，它们仅在将程序转换为助记代码时，以指令的形式使用；另一类是梯形图右侧的指令，按照通向它们指令行上的条件来执行。

大部分指令至少有一个与它们相关的操作数。操作数表示或给出可以完成指令的数据。有时这些作为实际数字的形式输入，但是通常它们是包含将使用数据的数据区字或位的地址，例如，一个有表示为源操作数的 IR000 的传送指令，将 IR000 的内容转移到某些其他位置，这个位置也将被表示为一个操作数。由一个操作数指定的位地址称为一个操作数位；一个操作数指定的一个字地址称为一个操作数字。如果输入的实际数值是个常数，那么在它前面加“#”表示不是一个地址。

### (3) 基本指令梯形图

一个梯形图是由左边一条垂直向下的线以及若干条与之相交并向右延伸的分支线组成。

在左边的这条线就叫做母线；分支线称指令行或回路。沿指令行设置各种条件，它们连向右边的其他指令。这些条件的逻辑组合决定右边的指令何时执行和怎样执行。图 1-2 是一个梯形图例子。

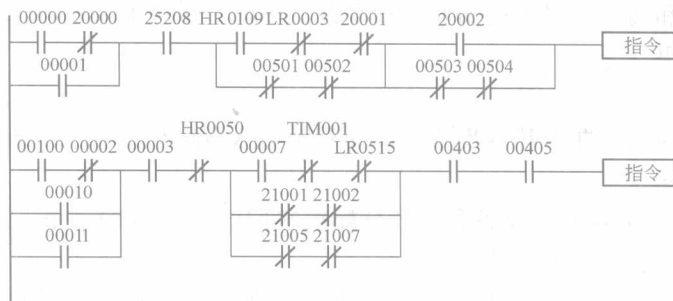


图 1-2 一个梯形图例子

如图 1-2 所示，指令行可以分支，也可以汇合。行上垂直的一对线叫做条件。无斜线穿过的条件称为常开条件，对应着 LOAD（载入）、AND（与）和 OR（或）指令。有斜线穿过的条件称为常闭条件，对应着 LOAD NOT（载入非）、AND NOT（与非）和 OR NOT（或非）指令。每个条件上方的数字表示指令的操作数位。它确定后面指令的执行条件的每个条件相关位的状态。

当用软件显示梯形图时，在梯形图的右边将给出另外一条母线，它与右侧指令相连。这不会改变这个梯形图的任何功能。在右侧的指令和右边母线之间不能放置任何条件，即在右侧的所有指令必须直接连接到右边的母线上。

下面将说明每个条件对应的指令的运行方法。但是，在考虑这些问题之前，必须先解释一些基本的术语。

#### ① 基本术语 常开和常闭条件如图 1-3 所示。



图 1-3 梯形图的常开和常闭条件

梯形图中的每个条件是“ON”还是“OFF”，取决于分配给它的操作数位的状态。如果操作数位为“ON”，则常开条件为“ON”；如果操作数位为“OFF”，则常开条件为“OFF”；如果操作数位为“OFF”，则常闭条件为“ON”；如果操作数位为“ON”，则常闭条件为“OFF”。一般来说，要在某位为“ON”时产生动作，应使用常开条件，而要在某位为“OFF”时产生动作，应使用一个常闭条件。

a. 执行条件。在梯形图编程中，一个指令前面“ON”和“OFF”条件的逻辑组合确定了指令执行的组条件，指令在此条件下执行。无论这个条件是“ON”还是“OFF”，都叫做该指令的执行条件。除了 LOAD（载入）指令外，所有指令都有执行条件。

b. 操作数位。指定任何梯形指令的操作数位可以是在 IR、SR、HR、AR、LR 或 TC 区域内的任何位。这也就是说：梯形图中的条件可由 I/O 位、标志位、工作位、定时器/计数器等来决定。载入和输出指令也能使用 TR 区域的位，但只能用在一些特殊用途。详情参

见指令行分支。

c. 逻辑块。与指令行相连的条件间的关系决定了条件对应的指令。任何一组能产生一个逻辑结果的条件组合称作逻辑块。尽管可以在没有实际分析单独逻辑块的情况下写出梯形图，但理解逻辑块对于更有效地编程是必要的。而且在程序要以助记代码（助记符）形式输入时，逻辑块是至关重要的。

d. 指令块。指令块由梯形图内部相连的所有指令组成。因此，一个指令块是由在没有任何相交垂直线情况下在梯形图中画出一条水平线的地方和能画出同类型水平线的下一个位置之间的所有指令组成。

② 助记符 梯形图不能直接通过手持编程器输入到 PC 中，此时需要 SSS。为了用手持编程器输入程序，则需要将梯形图转换为助记符形式。助记符可以提供同梯形图完全一样的信息，但这种形式可以直接被送入 PC。实际上可以直接用助记符进行编程，但它并不受初学者的欢迎，也不适合复杂程序。同样，不管使用何种编程设备，程序总是以助记符形式存储在内存里，所以理解助记符是非常重要的。把手持编程器作为编程设备以及完全理解在程序中的助记符有着重要的意义。记住，如果通过 SSS 输入程序则不需要使用助记符（用 SSS 时也可使用助记符）。程序被输入到程序内存中的地址。在程序内存中的地址与其他内存区域的地址略有不同，因为每个地址不必存有相同数量的数据。相反，每个地址存有一个指令和该指令需要的所有定义及操作数。

因为有些指令不需要操作数，而有些则最多需要三个操作数，所以程序内存地址可以是 1~4 个字长。程序内存地址从 00000 开始直到程序内存容量用完为止。每个地址第一个字定义了指令。而任何用于指令的定义也包含在第一个字中。如果一条指令仅要求单个位操作数（无定义），那么这个位操作数和该指令在同一指令行上编程。被一条指令要求的字中的其余部分包含着操作数，该操作数规定了什么样的数据将被使用。当转换为助记符形式时，除了梯形图外，所有指令以同样的形式写出，一个字对应一行，如同它们在梯形图符号中出现一样。表 1-1 给出一个助记符的例子。

表 1-1 助记符的例子

地 址	指 令	操 作 数	地 址	指 令	操 作 数
00000	LD	HR 0001			CM 0000
00001	AND	00001			HR 00
00002	OR	00002	00008	AND	25505
00003	LD NOT	00100	00009	OUT	20000
00004	OR	00101	00010	MOV(21)	
00005	AND LD				CM 0000
00006	MOV(21)				CM 0500
		000	00011	LD	00502
		CM 0000	00012	AND	00005
00007	CMP(20)		00013	OUT	20001

③ 梯形指令 梯形指令是在梯形图上对应于条件的指令。梯形指令既可独立于逻辑块指令（下面介绍），又可与之组合。它形成了执行所有其他指令的基本执行条件。

a. LOAD 和 LOAD NOT 指令。启动梯形图中任何逻辑块的第一条指令便是 LOAD 和 LOAD NOT 指令，它们都需要一条助记符。下面例子中的“指令”是一个代表，它可以是后面要介绍的右侧指令中的任何一条。指令，如图 1-4 所示。

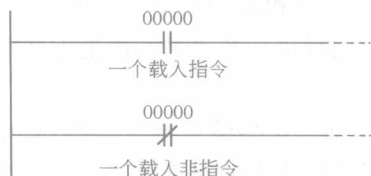


图 1-4 LOAD 和 LOAD NOT 指令

b. AND 和 AND NOT 指令。当指令行上的条件是唯一条件且为“ON”时，右侧的指令执行条件为“ON”。对于 LOAD 指令（即一个常开条件），当 IR00000 为“ON”时，执行条件为“ON”；对于 LOAD NOT 指令（即一个常闭条件），当 00000 为“ON”时，执行条件为“OFF”。

当同一指令行上存在串联两个或更多条件时，那么第一个条件对应于一条 LOAD 或 LOAD NOT 指令；余下条件对应于 AND 或 AND NOT 指令。如图 1-5 所示的指令 3 个条件，按顺序（从左至右）分别对应一条 LOAD 指令、一条 AND NOT 指令和一条 AND 指令。而每条指令都需要一个助记符行。



图 1-5 AND 和 AND NOT 指令

仅当所有三个条件都为“ON”时指令才有一个“ON”执行条件，即当 IR00000 为“ON”、IR00100 为“OFF”和 LR0000 为“ON”时，执行指令。

连续的 AND 指令可以分开单独考虑，每个 AND 指令都带有执行条件的逻辑与（即达到那点的所有条件的总和）和与指令的操作数位的状态。如果这些条件都为“ON”，则将为下条指令而产生一个“ON”执行条件；如果两者中有任一为“OFF”，则结果将为“OFF”。当 AND 指令串联时，第一条 AND 指令的执行条件是在指令行上的第一个条件。当串联时，每一条 AND NOT 指令将对其执行条件和它的操作数位的状态非进行逻辑“与”。若 OR 和 OR NOT 指令的两个或更多条件放置在相互独立的指令行上，且这些条件并联相接，那么第一个条件对应于一条 LOAD 或 LOAD NOT 指令；其他条件对应于 AND 或 AND NOT 指令。图 1-6 示出了三个条件，分别对应（按顺序至上而下）LOAD NOT、OR NOT 及 OR 指令。而且，其中每一条指令都需要一行助记符。

c. AND 和 OR 指令的组合。当三个条件中任何一个为“ON”，即当 IR00000 为“OFF”、IR00100 为“OFF”，或 LR0000 为“ON”时，指令会有一个“ON”执行条件。或和或非指令的作用可单独考虑，都在它的执行条件和或指令的操作数位的状态之间做逻辑或运算，如图 1-6 所示。

如果这些条件中任一为“ON”，则将为下个指令而产生一个“ON”执行条件。当 AND

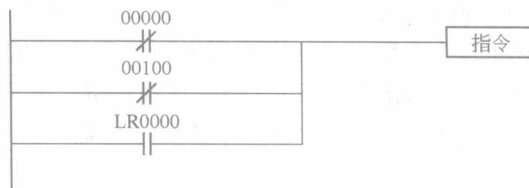


图 1-6 或和或非指令

和 OR 指令在更为复杂的图中组合在一起时，它们有时可以单独考虑，每条指令在执行条件和操作数位的状态上完成一个逻辑操作，如图 1-7 所示。仔细研究此图，可发现助记符和梯形图遵循同一逻辑。



图 1-7 AND 和 OR 指令的组合

这里，对 IR00000 和 IR00001 两者进行 AND 操作，以作为 OR 指令的执行条件（即 IR00200 状态）。这个 OR 操作结果作为 IR00002 的执行条件，然后与 IR00002 的状态进行 AND 操作，而此结果又作为 IR00003 状态的执行条件，和它的状态非进行 AND 操作。在更复杂的梯形图中，在为最后一条指令确定一个执行条件之前必须考虑逻辑块，此时要用 AND LOAD 和 OR LOAD 指令。但在考虑更为复杂的梯形图之前，将要看一下完成一个简单的输入/输出程序所需要的指令。

④ 输出和输出非 最简单的输出执行条件的组合结果的方法是直接用 OUT 和 OUT NOT 指令输出。这些指令用于根据执行条件，控制指定的操作位的状态。使用 OUT 指令，只要执行条件为“ON”，则操作位将置“ON”；执行条件为“OFF”，则操作位将置“OFF”。使用 OUT NOT 指令，只要执行条件为“OFF”，则操作位将置“ON”；执行条件为“ON”，则操作位将置“OFF”。这些情况如图 1-8 所示。在以助记符形式编程时，上述每条指令都需一行。

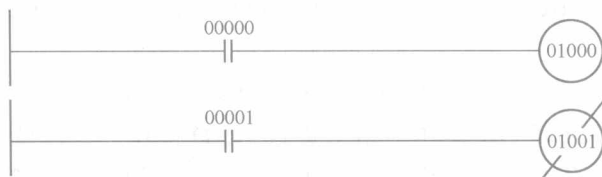


图 1-8 输出和输出非

在图 1-8 中，只要 IR00000 为“ON”则 IR01000 将变为“ON”；只要 IR00001 为“ON”则 IR01001 变为“OFF”。在此，IR00000 和 IR00001 是输入位，而 IR01000 和 IR01001 是分配给受 PC 控制的单元输出位，即通过分配在 IR00000 和 IR00001 的输入点进来的信号分别控制分配在 IR01000 和 IR01001 的输出点。

OUT、OUT NOT 指令与定时指令结合可以控制位输出“ON”或“OFF”的时间长短。

⑤ 结束指令 用来完成一个简单的程序的最后一条指令就是 END（结束）指令。当

CPU 单元扫描程序时，它执行所有的指令，直到第一条 END 指令为止，然后返回到程序开始，再次执行。END 指令可以放在程序的任意位置上，特别在调试程序时，在第一条 END 指令后的指令不会执行（除非把它删掉）。END 指令后面的数是指令功能代码，当输入大部分指令到 PC 中时使用指令功能代码。END 指令不需要操作数，且 END 指令的指令行上不能设置条件。END（结束）指令应用如图 1-9 所示。

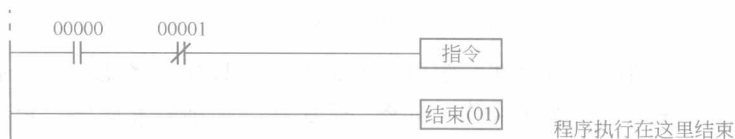


图 1-9 END（结束）指令应用

如果在程序中没有 END 指令，那么程序就根本不执行。

#### ⑥ 逻辑块指令

以上介绍了所有编写简单输入/输出程序必需的指令。在完成基本梯形图内容并转入向 PC 输入程序之前，介绍逻辑块指令（AND LOAD 和 OR LOAD），这些指令即使对简单梯形图，有时也是需要的。

逻辑块指令不需要对应梯形图上特定的条件，它们描述的是逻辑块之间的关系。AND LOAD 指令对两个逻辑块产生逻辑与的执行条件。OR LOAD 指令对两个逻辑块产生逻辑或的执行条件。

虽然图 1-10 所示出的梯形图看上去比较简单，但需要一条 AND LOAD 指令。

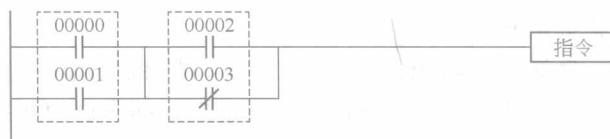


图 1-10 需要一条 AND LOAD 指令图例

a. AND LOAD。如图 1-10 所示，两个逻辑块用虚线表示。这个例子说明当左边的逻辑块中的任一条件是“ON”（即当 IR00000 或 IR00001 为“ON”），同时右边的逻辑块中的任一条件是“ON”（即当 IR00002 为“ON”或 IR00003 为“OFF”时），将产生一个“ON”执行条件。

但是，仅 AND 和 OR 指令是不能把上述梯形图转换为助记符形式的。如果对 IR00002 与 IR00000 和 IR00001 之间 OR 操作的结果进行 AND 操作，这样就无法进行 IR00002 和 IR00003 之间的 OR NOT 操作，而变成最终在 IR00003、IR00002 和第一个 OR 到 AND 操作的结果之间进行 OR NOT 操作。而程序要做的是独立地进行 OR（NOT）操作，然后将所有的结果结合起来。要做到这一点，可在一个指令行的中间使用 LOAD 或 LOAD NOT 指令。当在这种方法中使用 LOAD 或 LOAD NOT 指令时，当前执行条件保存在一个特定缓冲区里，而且逻辑过程重新开始。为了将当前执行条件与先前没用的执行条件结合，将使用一个 AND LOAD 或 OR LOAD 指令。这里 LOAD 是指使最后一个未用的执行条件被取出。除了指令行上第一个条件外，使用 LOAD 或 LOAD NOT 指令能产生一个未使用的执行条件。根据助记符指令来分析如图 1-10 所示的梯形图，IR00000 的条件是一条 LOAD 指令，下面的条件是 IR00000 和 IR00001 的 OR 指令。IR00002 的条件是另一条 LOAD 指令，



其下面的条件是一条 OR NOT 指令，即 IR00002 的状态和 IR00003 非状态的一条 OR 指令。为了达到右侧指令的执行条件，必须对两个逻辑块执行逻辑与（AND）操作，AND LOAD 实现了这个功能。AND LOAD 指令本身不需要操作数，因为它按先前确定的执行条件操作。

b. OR LOAD。如图 1-11 所示，上面逻辑块与下面逻辑块之间需要一条 OR LOAD 指令。当 IR00000 为“ON”和 IR00001 为“OFF”时，或当 IR00002 和 IR00003 都为“ON”的时候，都将为右侧的指令产生一个“ON”执行条件。除了当前执行的条件和最后一个未使用执行条件之间进行的是 OR 操作外，OR LOAD 指令的助记符的操作同 AND LOAD 指令一样。

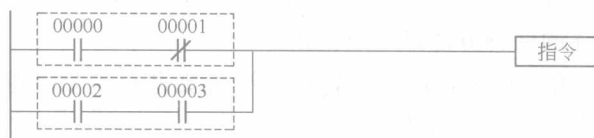


图 1-11 需要一条 OR LOAD 指令的例子

当然，有些梯形图也会同时需要 AND LOAD 和 OR LOAD 指令。

c. 逻辑块指令串联。用串联逻辑块指令对梯形图编程，梯形图必须划分为若干个逻辑块。每块采用一条 LOAD 或 LOAD NOT 指令编写第一个条件，然后使用 AND LOAD 或 OR LOAD 指令对这些逻辑块进行逻辑组合。用 AND LOAD 或 OR LOAD 指令时，可以有两途径实现上述目的。一条途径是在前面两个逻辑运算后用逻辑块指令编写，然后在每增加一个逻辑块后面用逻辑块指令编写；另一条途径是对所有要结合的逻辑块编写，即先用 LOAD 或 LOAD NOT 指令开始每个逻辑块编写，然后用逻辑块指令把它们组合起来。在此过程中，用于最后一对逻辑块的指令将首先被结合，然后前面的一逻辑块被结合，逐次递推回第一个逻辑块。尽管上述两种方法产生完全相同的结果，但第二种对所有逻辑块指令一起编写的方法仅用在有八个（或少于八个）逻辑块组合的情况，也即要用七条（或七条以下）逻辑块指令的情况。如图 1-12 所示的梯形图转化为助记符时，需要用 AND LOAD 指令，因为有三对并联条件串联连接，如图 1-12 所示。



图 1-12 三对并联条件串联连接情况

用第二种方法进行编程时，可以组合的逻辑块最多为八个。而采用第一种方法编程时，相连逻辑块的数目没有限制。

如图 1-13 所示，梯形图转化为助记符时，需要用 OR LOAD 指令，因为有三对串联条件并联连接。

每对条件中的第一个条件用分配的位操作数转换成 LOAD，然后和该对中另一个条件进行与操作。前面两个逻辑块可用接下来的一条 OR LOAD 指令先编写，而最后一个逻辑块用另一条 OR LOAD 指令编写，或者首先对前面三个逻辑块编写，接着连续用两条 OR LOAD 指令。