

高等院校通用教材

# 现代电气控制及 PLC应用技术

王永华 主编 宋寅卯 陈玉国 郑安平 副主编



 北京航空航天大学出版社  
<http://www.buaapress.com.cn>

责任编辑：金友泉

封面设计：陈鹤工作室

## 自动控制及测控原理教材

现代电气控制及 PLC 应用技术	王永华	37.00
测控电路及装置	孙传友等	26.50
测控系统原理与设计	孙传友等	28.50
自动控制原理	冯巧玲等	32.00
Verilog 数字系统设计教程	姜宇闻等	38.00
智能控制与 Lon 网络开发技术	马莉	26.00
智能传感与信息系统	梁威等	待出

ISBN 7-81077-346-1



9 787810 773461 >

ISBN 7-81077-346-1

定价：27.00 元

# 现代电气控制及 PLC 应用技术

王永华 主 编  
宋寅卯 陈玉国 郑安平 副主编

北京航空航天大学出版社

<http://www.buaapress.com.cn>

## 内 容 简 介

本书从实际工程应用和便于教学需要出发,介绍和讲解了继电接触式控制系统和可编程序控制器控制系统的工作原理、设计方法和实际应用。和其他同类的教材相比,本书主要有以下特点:(1)介绍了一些新型器件,讲解了软启动器和变频器的使用;(2)对传统的电气控制系统的内容进行了较大幅度的删节,给出并讲解了电气控制线路和可编程序控制器程序的“简单设计法”;(3)系统介绍和讲解了最新版本的 SIEMENS S7-200 CPU22\*(V1.21)系列可编程序控制器的原理和应用,并给出了大量实例并讲解其基本指令的用法和功能图(SFC)的编程;(4)对 S7-200 PLC 的功能指令和通信功能进行了详细的讲解,并简单介绍了 S7-200 PLC 新模块的特点;(5)附有思考题、练习题和实验指导书;(6)介绍了 S7-200 可编程序控制器上机编程软件的使用;(7)附有作者精心挑选并经修改过的 S7-200 PLC 资料速查表。

本书是作者在自己最近出版的同类教材的基础上精心修订和编写而成的,相信它会是一本值得大家使用的书。

本书可作为大专院校、电大和业余大学的自动控制、电气技术、机电一体化及相关专业的“电气控制及可编程序控制器”或类似课程的教材,也可供有关工程技术人员参考使用,同时它也是广大从事和电气控制技术专业有关的电工和技术人员的一本很好的自学教材。

## 图书在版编目(CIP)数据

现代电气控制及 PLC 应用技术/王永华等编著.

—北京:北京航空航天大学出版社,2003.9

ISBN 7-81077-346-1

I. 现… II. 王… III. ①电气设备—自动控制

②可编程序控制器—程序设计 IV. TM762

中国版本图书馆 CIP 数据核字(2003)第 045843 号

## 现代电气控制及 PLC 应用技术

王永华 主 编

宋寅卯 陈玉国 郑安平 副主编

责任编辑 金友泉

\*

北京航空航天大学出版社出版发行

北京市海淀区学院路 37 号(100083) 发行部电话:82317024 传真:82328026

<http://www.buaapress.com.cn>

E-mail: bhpess@263.net

北京市松源印刷有限公司印装 各地书店经销

\*

开本:787 mm×1 092 mm 1/16 印张:19.75 字数:506 千字

2003 年 09 月第 1 版 2003 年 09 月第 1 次印刷 印数:5 000 册

ISBN 7-81077-346-1 定价:27.00 元

## 前 言

目前高等院校已普遍将“工厂电气控制技术”和“可编程序控制器原理及应用”两门课程合并为“电气控制及可编程序控制器”一门课程。针对这种教学安排和电气控制技术及可编程序控制器的最新发展,我们组织编写了这本教材。

在编写过程中,编者力求做到语言通畅、叙述清楚、讲解细致,所有的内容都为了便于实际应用和教学,尽可能多地融进自己的经验和成果。本书还结合电气控制技术的最新发展,对一些传统的教学内容进行了较大篇幅的删节,增加了许多新的内容,重要的是本书在讲解可编程序控制器(PLC,全书简称)时,以现在最流行的、有较高性能价格比的 SIEMENS S7-200 CPU22\*(V1.21)系列小型 PLC 为对象,使大家接触到最新的 PLC 产品。

全书共分十章。第一章主要介绍低压电器的原理、用途和选用,其中增加了许多新型器件的内容。第二章讲解常用控制线路基本环节的设计和工作原理,并用较大篇幅,结合实例讲解了软启动器和新型变频器的应用,还特别讲解了电气控制线路的“简单设计法”,最后简单介绍了传统的电气控制线路分析基础。第三章叙述了可编程序控制器的基本概念和概况。第四章介绍了 S7-200 PLC 的内部元器件和寻址方式。第五章结合例子详细介绍了 S7-200 CPU22\*系列 PLC 的基本指令,讲解了常用典型电路的程序设计和 PLC 程序的“简单设计法”。第六章重点讲解 PLC 的功能图编程方法和 S7-200 PLC 顺序控制指令的使用,其中对使用 S7-200 PLC 顺序控制指令时的注意事项作了特别说明。第七章以简洁流畅的方法介绍了 S7-200 PLC 的功能指令,并给出了许多例子。第八章详细讲解了 PLC 的网络通信原理,介绍了 S7-200 PLC 的通信功能和通信指令的使用,另外还对最新的通信模块作了介绍。第九章讲解了 PLC 在实际应用中的设计步骤,并给出了两个大型的较全面的实例供大家参考,最后从诸多方面讲解了 PLC 在实际应用中应注意的问题。第十章是 S7-200 PLC 编程软件的使用简介。本书的附录还提供了实验指导书和作者精心挑选并修订过的 S7-200 PLC 的参考资料。

本书既可以供少学时(如 40 学时)使用,也可以供多学时(如 65 学时)使用。两者的区别在于是对第七章、第八章和第九章作简单介绍,还是详细讲解。另外实验的多少也有不同。

本书由王永华任主编,负责全书的组织、统稿和改稿。前言、绪论、第二章的 2.6 节、第四章、第五章、第六章、第七章、第九章的前两节及附录 B 由王永华编写;其余部分由宋寅卯、陈玉国、郑安平、杨存祥、张东初、刘慧琴和卢刚共同编写。

特别感谢北航出版社的金友泉老师、西门子(中国)有限公司的王东滨先生以及杨涟老师和作者指导毕业设计的学生们,他们为本书的出版、编写和校对等提供了许多帮助,另外本书部分章节的编写参考了有关资料(见参考文献),在此我们对这些同志和参考文献的作者表示衷心的感谢!

由于本书编者水平有限,编写时间仓促,虽已尽心尽力,多次修改,但书中难免仍有错误和不足之处,敬请读者批评指正。作者电子信箱地址:wyh@zzuli.edu.cn。

编 者

2003 年 5 月

# 目 录

## 绪 论

## 第一章 常用低压电器

1.1 电器的基本知识 .....	3
1.1.1 电器的定义和分类 .....	3
1.1.2 电磁式低压电器的基本 结构和工作原理 .....	4
1.2 接触器 .....	10
1.2.1 接触器的用途及分类 .....	10
1.2.2 接触器的结构及工作 原理 .....	10
1.2.3 接触器的技术参数 .....	11
1.2.4 接触器的选择 .....	12
1.3 继电器 .....	13
1.3.1 电磁式继电器 .....	13
1.3.2 热继电器 .....	15
1.3.3 时间继电器 .....	18
1.3.4 速度继电器 .....	19
1.3.5 温度继电器 .....	20
1.3.6 液位继电器 .....	21
1.3.7 固态继电器 .....	21
1.4 开关电器 .....	23
1.4.1 刀开关 .....	23
1.4.2 低压断路器 .....	23
1.5 熔断器 .....	26
1.5.1 熔断器的结构和分类 .....	26
1.5.2 熔断器的保护特性 .....	27
1.5.3 熔断器的技术参数 .....	28
1.5.4 熔断器的选择 .....	28
1.6 主令电器 .....	29
1.6.1 控制按钮 .....	29
1.6.2 转换开关 .....	31
1.6.3 行程开关 .....	32

1.6.4 接近开关 .....	32
1.6.5 光电开关 .....	33
1.6.6 指示灯 .....	34
本章小结 .....	35
思考题与练习题 .....	35

## 第二章 电气控制线路基础

2.1 电气控制线路图的图形、文字 符号及绘制原则 .....	37
2.1.1 常用电气图形符号和 文字符号 .....	37
2.1.2 电气控制线路图的绘制 原则 .....	41
2.2 三相笼型异步电动机的基本 控制线路 .....	43
2.2.1 全压启动控制线路 .....	43
2.2.2 正反转控制线路 .....	44
2.2.3 点动控制线路 .....	45
2.2.4 多点控制线路 .....	46
2.2.5 顺序控制线路 .....	46
2.2.6 自动循环控制线路 .....	47
2.3 三相笼型异步电动机降压 启动控制线路 .....	48
2.3.1 星形—三角形降压启动 控制线路 .....	48
2.3.2 自耦变压器降压启动 控制线路 .....	49
2.3.3 软启动器及其使用 .....	50
2.4 三相笼型异步电动机制动 控制线路 .....	56
2.4.1 反接制动控制线路 .....	56
2.4.2 能耗制动控制线路 .....	58
2.5 三相笼型异步电动机速度 控制线路 .....	61
2.5.1 基本概念 .....	61
2.5.2 变极调速控制线路 .....	61

2.5.3 变频调速与变频器的使用 .....	63
2.6 电气控制线路的简单设计法 .....	68
2.6.1 概 述 .....	68
2.6.2 简单设计法介绍 .....	68
2.6.3 简单设计法设计举例 .....	71
2.7 典型生产机械电气控制线路分析 .....	72
2.7.1 电气控制线路分析基础 .....	72
2.7.2 C650 卧式车床电气控制线路分析 .....	73
本章小结 .....	78
思考题与练习题 .....	78

### 第三章 可编程序控制器概述

3.1 PLC 的产生和定义 .....	80
3.1.1 PLC 的产生 .....	80
3.1.2 PLC 的定义 .....	81
3.2 PLC 的特点 .....	82
3.3 PLC 的应用和发展 .....	83
3.3.1 PLC 的发展状况 .....	83
3.3.2 PLC 的发展趋势 .....	83
3.3.3 PLC 的应用领域 .....	85
3.4 PLC 的分类 .....	86
3.5 PLC 的系统组成 .....	87
3.6 PLC 与继电器控制系统的区别 .....	90
3.7 PLC 的工作原理 .....	91
3.7.1 PLC 的工作方式与运行框图 .....	91
3.7.2 PLC 工作过程的中心内容 .....	93
3.7.3 PLC 对输入/输出的处理原则 .....	94
3.8 PLC 的编程语言和程序结构 .....	94

3.8.1 PLC 的编程语言 .....	94
3.8.2 PLC 的程序结构 .....	96
本章小结 .....	96
思考题 .....	97

### 第四章 S7-200 系列 PLC 的硬件系统及内部资源

4.1 概 述 .....	98
4.2 S7-200 系列 PLC 的硬件系统 .....	98
4.2.1 硬件系统基本构成 .....	99
4.2.2 主机结构及性能特点 .....	99
4.2.3 输入/输出的扩展 .....	100
4.3 S7-200 系列 PLC 的内部资源及寻址方式 .....	102
4.3.1 软元件 .....	102
4.3.2 CPU 存储区域的直接寻址 .....	105
4.3.3 CPU 存储区域的间接寻址 .....	108
本章小结 .....	109
思考题与练习题 .....	109

### 第五章 PLC 的基本指令及程序设计

5.1 PLC 的基本逻辑指令及举例 .....	110
5.1.1 逻辑取及线圈驱动指令 .....	110
5.1.2 触点串联指令 .....	111
5.1.3 触点并联指令 .....	111
5.1.4 串联电路块的并联连接指令 .....	112
5.1.5 并联电路块的串联连接指令 .....	112
5.1.6 置位、复位指令 .....	113
5.1.7 RS 触发器指令 .....	114
5.1.8 立即指令 .....	115

5.1.9 边沿脉冲指令 .....	116
5.1.10 逻辑堆栈操作指令 .....	117
5.1.11 定时器 .....	119
5.1.12 计数器 .....	123
5.1.13 比较指令 .....	126
5.1.14 NOT 及 NOP 指令 .....	128
5.2 程序控制指令 .....	128
5.2.1 结束及暂停指令 .....	128
5.2.2 看门狗指令 .....	129
5.2.3 跳转及标号指令 .....	130
5.2.4 循环指令 .....	131
5.2.5 子程序 .....	132
5.2.6 与 ENO 指令 .....	135
5.3 PLC 初步编程指导 .....	136
5.3.1 梯形图编程的基本规则 .....	136
5.3.2 LAD 和 STL 编程形式的区别 .....	137
5.4 典型的简单电路编程 .....	138
5.4.1 延时脉冲产生电路 .....	138
5.4.2 瞬时接通/延时断开电路 .....	139
5.4.3 延时接通/延时断开电路 .....	139
5.4.4 脉冲宽度可控制电路 .....	140
5.4.5 计数器的扩展 .....	141
5.4.6 长定时电路 .....	141
5.4.7 闪烁电路 .....	142
5.4.8 报警电路 .....	143
5.5 PLC 程序的简单设计法及应用举例 .....	145
5.5.1 PLC 程序的简单设计法 .....	145
5.5.2 应用举例 .....	146
本章小结 .....	150
思考题与练习题 .....	150

## 第六章 S7-200 PLC 顺序控制指令及应用

6.1 功能图的产生及基本概念 .....	153
6.1.1 功能图的产生 .....	153
6.1.2 功能图的基本概念 .....	153
6.1.3 功能图的构成规则 .....	154
6.2 顺序控制指令 .....	155
6.2.1 顺序控制指令介绍 .....	155
6.2.2 举例说明 .....	155
6.2.3 使用说明 .....	156
6.3 功能图的主要类型 .....	157
6.3.1 单流程 .....	157
6.3.2 可选择的分支和联接 .....	157
6.3.3 并行分支和联接 .....	158
6.3.4 跳转和循环 .....	160
6.4 顺序控制指令应用举例 .....	161
6.4.1 选择和循环电路举例 .....	161
6.4.2 并行分支和联接电路举例 .....	165
6.4.3 选择和跳转电路举例 .....	168
本章小结 .....	171
思考题与练习题 .....	171

## 第七章 S7-200 PLC 的功能指令

7.1 传送、移位和填充指令 .....	173
7.1.1 传送类指令 .....	173
7.1.2 移位与循环指令 .....	174
7.1.3 字节交换指令 .....	177
7.1.4 填充指令 .....	177
7.2 运算和数学指令 .....	178
7.2.1 加法指令 .....	178
7.2.2 减法指令 .....	178
7.2.3 乘法指令 .....	178
7.2.4 除法指令 .....	179

7.2.5 数学函数指令 .....	181
7.2.6 增/减指令 .....	183
7.2.7 逻辑运算指令 .....	184
7.3 表功能指令 .....	186
7.4 转换指令 .....	189
7.4.1 数据类型转换指令 .....	189
7.4.2 编码和译码指令 .....	192
7.4.3 段码指令 .....	192
7.4.4 ASCII 码转换指令 .....	193
7.4.5 字符串转换指令 .....	196
7.5 字符串指令 .....	197
7.6 时钟指令 .....	200
7.7 中 断 .....	201
7.7.1 几个基本概念 .....	202
7.7.2 中断指令 .....	204
7.7.3 中断程序 .....	205
7.8 高速计数器指令 .....	206
7.8.1 高速计数器介绍 .....	206
7.8.2 高速计数器指令 .....	208
7.8.3 高速计数器的使用 .....	208
方法 .....	208
7.9 高速脉冲输出指令 .....	212
7.9.1 几个基本概念 .....	212
7.9.2 高速脉冲指令及特殊 .....	212
寄存器 .....	212
7.9.3 PTO 的使用 .....	214
7.9.4 PWM 的使用 .....	218
7.10 PID 回路指令 .....	221
7.10.1 PID 算法 .....	221
7.10.2 PID 回路指令及使用 .....	221
本章小结 .....	225
练 习 题 .....	226

## 第八章 PLC 的网络通信技术 及应用

8.1 通信网络的基础知识 .....	227
8.1.1 数据通信方式 .....	227

8.1.2 网络概述 .....	230
8.2 S7-200 的通信与网络 .....	231
8.2.1 S7 系列 PLC 网络 .....	231
层次的结构 .....	231
8.2.2 S7-200 PLC 网络 .....	232
的通信协议 .....	232
8.2.3 网络配置实例 .....	236
8.2.4 网络部件 .....	238
8.3 S7-200 通信指令 .....	239
8.3.1 网络读/网络写指令 .....	239
8.3.2 发送与接收指令 .....	242
8.3.3 USS 通信指令 .....	247
8.4 S7-200 的通信扩展模块 .....	248
8.4.1 EM241 调制解调器 .....	248
模块 .....	248
8.4.2 CP243-1 工业以太网 .....	248
通信处理器模块 .....	248
本章小结 .....	249
思考题与练习题 .....	249

## 第九章 现代 PLC 控制系统 综合设计实例

9.1 PLC 控制系统设计步骤及 .....	250
内容 .....	250
9.1.1 分析评估及控制任务 .....	251
9.1.2 PLC 的选型 .....	251
9.1.3 I/O 地址分配 .....	252
9.1.4 系统设计 .....	252
9.1.5 系统调试 .....	252
9.2 双恒压无塔供水控制系统 .....	253
设计 .....	253
9.2.1 工艺过程 .....	253
9.2.2 系统控制要求 .....	254
9.2.3 控制系统的 I/O 点及 .....	254
地址分配 .....	254
9.2.4 PLC 系统选型 .....	255
9.2.5 电气控制系统原理图 .....	255

.....	255
9.2.6 系统程序设计 .....	258
9.3 薄刀式分切压痕机控制系统 .....	264
9.3.1 工艺过程 .....	264
9.3.2 系统控制要求 .....	264
9.3.3 控制系统的 I/O 点及地址分配 .....	265
9.3.4 PLC 系统选型 .....	265
9.3.5 电气控制系统原理图 .....	265
9.3.6 系统程序设计 .....	267
9.4 PLC 在工程应用中要注意的一些实际问题 .....	272
9.4.1 PLC 的安装 .....	272
9.4.2 电源的设计 .....	272
9.4.3 系统的接地 .....	273
9.4.4 电缆设计与铺设 .....	274
9.4.5 PLC 输出端的保护 .....	274
本章小结 .....	275
思考题与练习题 .....	275

## 第十章 编程软件的使用

10.1 编程软件安装 .....	276
10.1.1 系统要求 .....	276
10.1.2 软件安装 .....	276
10.1.3 硬件连接 .....	277
10.1.4 参数设置 .....	277
10.1.5 在线联系 .....	278
10.1.6 建立、修改 PLC 通信参数 .....	278
10.2 软件功能 .....	278
10.2.1 基本功能 .....	278
10.2.2 界面 .....	279
10.2.3 各部分功能 .....	279
10.2.4 系统组态 .....	281
10.3 编程 .....	281
10.3.1 程序文件操作 .....	281

10.3.2 编辑程序 .....	282
10.4 调试及运行监控 .....	286
10.4.1 选择扫描次数 .....	286
10.4.2 状态图表监控 .....	286
10.4.3 运行模式下的编辑 .....	287
10.4.4 程序监视 .....	288
本章小结 .....	289

## 附录 A 实验指导书

实验一 异步电动机可逆运行实验 .....	290
实验二 S7-200 PLC 编程软件使用实验 .....	291
实验三 抢答器程序设计实验 .....	291
实验四 人行道按钮控制交通灯程序设计实验 .....	292
实验五 水位控制程序设计实验 .....	294

## 附录 B S7-200 PLC 快速参考信息

表 B-1 S7-200 PLC 的 CPU 规范 .....	295
表 B-2 S7-200 PLC 的 CPU 输入规范 .....	296
表 B-3 S7-200 PLC 的 CPU 输出规范 .....	297
表 B-4 S7-200 PLC 的 CPU 存储器范围和特性总汇 .....	298
表 B-5 S7-200 PLC [CPU (V1.21)] 指令系统速查表 .....	299
表 B-6 常用特殊继电器 SM0 和 SM1 的位信息 .....	301
图 B-1 CPU 224 外围典型接线图 .....	302

## 参考文献

# 绪 论

## 1. 电气控制技术的发展

电气控制技术是随着科学技术的不断发展及生产工艺不断提出新的要求而得到飞速发展的。在控制方法上,主要是从手动控制到自动控制;在控制功能上,是从简单的控制设备到复杂的控制系统;在操作方式上,由笨重到轻巧;在控制原理上,从有触点的继电接触式控制系统到以计算机为核心的“软”控制系统。随着新的电器元件的不断出现和计算机技术的发展,电气控制技术也在持续发展。现代电气控制技术正是综合了计算机、自动控制、电子技术和精密测量等许多先进科学技术成果,并得到飞速发展。现在 PLC、CAD/CAM 和 Robot 组成了当代工业自动化技术的三大技术支柱。

我们知道,工业生产的各个领域,无论是过程控制系统还是电气控制系统,都有大量的开关量和模拟量信号。开关量又称为数字量,如电动机的启停,阀门的开闭,电子元件的置位与复位,按钮及位置检测开关的状态和定时器及计数器的状态等;模拟量又称为连续量,如温度、流量、压力和液位等。对这些信号和对象的处理就要使用自动控制的方法来完成。

20 世纪 70 年代以前,电气自动控制的任務基本上都由继电接触式控制系统完成。该系统主要由继电器、接触器和按钮等组成,它取代了原来的手动控制方式。由于这种控制系统具有结构简单、价格低廉、抗干扰能力强等优点,所以当时使用得十分广泛,至今仍在许多简单的机械设备中应用。但这种控制系统的缺点也是非常明显的,它采用固定的硬接线方式来完成各种控制逻辑,实现系统的各种控制功能,所以灵活性差;另外,由于机械式的触点工作频率低,易损坏,因此可靠性低。

社会的发展和进步对各行各业提出了越来越高的要求。机械加工企业为了提高生产效率和市场竞争力,采用了机械化流水线作业的生产方式,对不同的产品零件分别组成自动流水线。产品不断地更新换代,也同时要求相应的控制系统随之改变。在这种情况下,硬连接方式的继电接触式控制系统就不能满足经常更新的要求了。这是因为,一是成本高,二是周期长。后来出现的矩阵式顺序控制器和晶体管逻辑控制系统取代过继电接触式控制系统,由于这些控制装置仍是硬连接,装置体积大,功能少,本身存在某些不足,虽然提高了控制系统的通用性和灵活性,但均未得到广泛的应用。

随着大规模集成电路和微处理器的发展和应用,在 1969 年出现了世界上第一台以软件手段来实现各种控制功能的革命性控制装置——可编程序逻辑控制器。它把计算机的功能完备、通用性和灵活性好等优点和继电接触式控制系统的操作方便、简单易懂、价格低廉等优点结合起来,因此它是一种适应于工业环境的通用控制装置。现在的可编程序控制器和原来的控制系统相比,增加了算术运算、数据转换、过程控制、数据通信等功能,已可以完成大型而复杂的控制任务。可编程序控制器作为工业自动化的技术支柱之一,在工业自动控制领域占有十分重要的地位。

数控技术也是电气自动控制的一个重要分支。它综合了计算机、自动控制、伺服驱动系

统、精密检测与新型机械结构等多方面的最新技术成就。最近 20 多年来,机电一体化、机电光仪一体化等交叉学科的发展,使得数控技术也得到了飞速的发展。因此在机械制造、电气控制及自动控制领域内相继出现了直接数字(DDC)系统、柔性制造系统(FMS)、计算机集成制造系统(SIMS)、综合运用计算机辅助设计(CAD)、计算机辅助制造(CAM)、集散控制系统(DCS)、智能机器人和智能制造等高新技术。这些高新技术把整个自动控制和自动制造技术推动到了更高的水平。

说到自动控制技术的最新发展,我们必须提及现场总线控制系统(Fieldbus Control System, FCS)。FCS 是在计算机网络技术、通信技术和微电子技术飞速发展的基础上,与自动控制技术相结合的产物。它是继集散控制系统(DCS)之后的新一代控制系统,也是现在工业自动化研究开发热点之一。FCS 是用于现场仪表、设备之间以及现场与控制系统之间的一种全数字、双向串行、多节点的通信系统。它把具有数字计算和通信能力的现场仪表和执行器件连接成网络系统,按公开、规范的通信协议,在现场与上位机或网络之间实现数据传输和信息交换,FCS 适应了工业自动控制系统向分散化、智能化和网络化发展的方向,它的出现导致了传统的自动化仪表和控制系统在结构和功能上的重大变革。电气控制系统作为 FCS 中底层网络的重要组成部分,这就要求新型的检测器件、执行器件和智能控制器(如 PLC)必须具备和现场总线通信的能力。FCS 的发展和越来越多的应用宣告了工业自动控制系统一个新时代的到来。

## 2. 本课程的性质、内容和任务

本课程是一门实用性很强的专业课。电气控制技术在生产过程、科学研究和其他各个领域的应用十分广泛。本课程的主要内容是以电动机或其他执行电器为控制对象,介绍和讲解继电接触式控制系统和可编程序控制器控制系统的工作原理、设计方法和实际应用。其中可编程序控制器的飞速发展和其强大的功能使它已成为实现工业自动化的主要手段之一。所以本课程的重点是可编程序控制器,但这并不意味着继电接触式控制系统就不重要了。这是因为:首先,继电接触式控制在小型电气系统中还普遍使用,而且它是组成电气控制系统的基础;其次,尽管可编程序控制器取代了继电器,但它所取代的主要是逻辑控制部分,而电气控制系统中的信号采集和驱动输出部分仍然要由电气元器件及控制电路来完成。所以对继电接触式控制系统的学习是非常必要的。该课程的目标是让学生掌握一门非常实用的工业控制技术,以及培养和提高学生的实际应用和动手能力。

电气控制技术是电类专业学生所必须掌握的最基础的实际应用课程,具体要求是:

- (1) 熟悉常用控制电器的工作原理和用途,达到正确使用和选用的目的,同时要了解一些新型元器件的用途。
- (2) 熟练掌握电气控制线路的基本环节,并具备阅读和分析电气控制线路的能力,使之能设计简单的电气控制线路,较好地掌握电气控制线路的简单设计法。
- (3) 熟悉可编程序控制器的基本概况,深刻领会可编程序控制器的工作原理。
- (4) 熟练掌握可编程序控制器的基本指令系统和典型电路的编程,掌握可编程序控制器的程序设计方法;熟练掌握功能图的编程方法。掌握和熟悉可编程序控制器功能指令的使用。
- (5) 掌握和了解可编程序控制器的网络和通信原理,会编制简单的通信程序。
- (6) 了解可编程序控制器的实际应用程序的设计步骤和方法。

# 第一章 常用低压电器

低压电器是电力拖动控制系统、低压供配电系统的基本组成元件,其性能的优劣直接影响着系统的可靠性、先进性和经济性,是电气控制技术的基础。因此,必须熟练掌握低压电器的结构、工作原理并能正确使用。本章主要介绍常用低压电器的结构、工作原理以及使用方法等有关知识,同时根据电器发展状况,对新型电器元件予以简单介绍。

## 1.1 电器的基本知识

### 1.1.1 电器的定义和分类

电器就是根据外界施加的信号和要求,能手动或自动地断开或接通电路,断续或连续地改变电路参数,以实现了对电或非电对象的切换、控制、检测、保护、变换和调节的电工器械。低压电器通常指工作在交直流电压 1 200 V 以下的电器。采用电磁原理完成上述功能的低压电器称为电磁式低压电器。

电器的种类很多,分类方法也很多。常见的分类方法如图 1-1 所示。

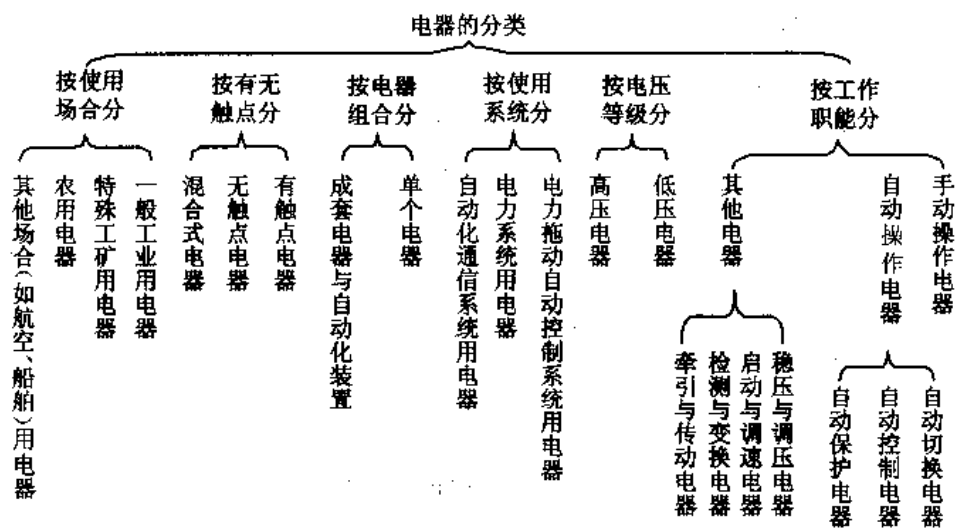


图 1-1 电器的分类

常用低压电器的分类如图 1-2 所示。

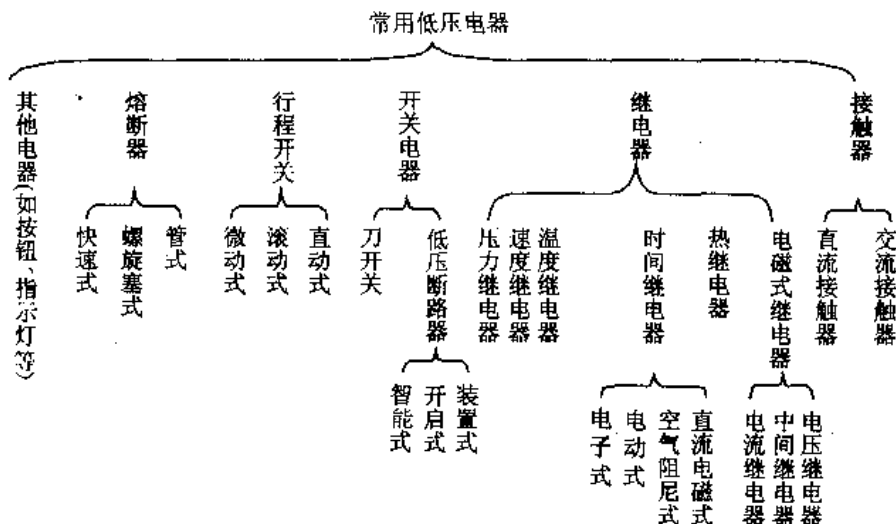


图 1-2 常用低压电器分类

### 1.1.2 电磁式低压电器的基本结构和工作原理

电磁式低压电器在电气控制线路中使用量最大,其类型也很多,各类电磁式低压电器在工作原理和构造上亦基本相同。就其结构而言,大都由三个主要部分组成,即触头、灭弧装置和电磁机构。

#### 1. 触 头

触头是一切有触点电器的执行部件,这些电器通过触头的动作来接通或断开被控制电路的。触头通常由动、静触点组合而成。

##### (1) 触点的接触形式

触点的接触形式有点接触(如球面对球面、球面对平面等)、线接触(如圆柱对平面、圆柱对圆柱等)和面接触(如平面对平面)三种,如图 1-3 所示。

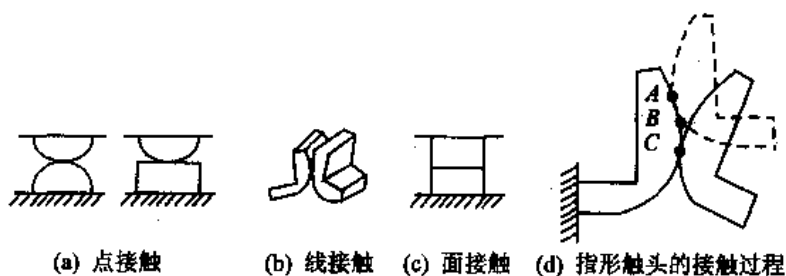


图 1-3 触点的接触形式

三种接触形式中,点接触形式的触点只能用于小电流的电器中,如接触器的辅助触点和继电器的触点;面接触形式的触点允许通过较大的电流,一般在接触表面上镶有合金,以减小触点接触电阻和提高耐磨性,多用于较大容量接触器的主触点;线接触形式的触点接触区域是一条直线,其触点在通断过程中有滚动动作,如图 1-3(d)所示。开始接触时,动静触点在 A 点接触,靠弹簧的压力经 B 点滚到 C 点,断开时作相反运动,这样可以清除触点表面的氧化膜。同时长期工作的位置是在 C 点而不是在易烧灼的 A 点,从而保证了触点的良好接触。这种滚

### (1) 多断点灭弧

在交流继电器和接触器中常采用桥式触头(如图 1-4 所示),这种触头有两个断点。交流电压在过零后,若一对断点处电弧重燃需要 150~250 V 电压,则两对断点就需要 300~500 V 电压。若断点电压达不到此值,则电弧因不能重燃而熄灭。一般交流继电器和小电流接触器采用桥式触点灭弧,而不再加设其他灭弧装置。

### (2) 磁吹式灭弧

这种灭弧的原理是使电弧处于磁场中间,电磁场力“吹”长电弧,使其进入冷却装置,加速电弧冷却,促使电弧迅速熄灭。

图 1-5 是磁吹式灭弧的原理图。其磁场由与触点电路串联的吹弧线圈 1 产生,当电流逆时针流经吹弧线圈时,其产生的磁通经铁心 3 和导磁夹板 5 引向触点周围。触点周围的磁通方向为由纸面流入,如图中“×”符号所示。由左手定则可知,电弧在吹弧线圈磁场中受一向上方向的力  $F$  的作用,电弧向上运动,被拉长并被吹入灭弧罩 6 中。引弧角 4 和静触点 8 相连接,引导电弧向上运动,将热量传递给灭弧罩壁,促使电弧熄灭。

这种灭弧装置是利用电弧电流本身灭弧,电弧电流越大,吹弧能力越强,且不受电路电流方向影响(当电流方向改变时,磁场方向随之改变,结果电磁力方向不变)。它广泛地应用于直流接触器中。

### (3) 灭弧栅

灭弧栅的原理如图 1-6 所示。灭弧栅片 1 是由镀铜薄钢片组成,灭弧栅是由许多灭弧栅片组成,片间距离为 2~3 mm,安放在触点上方的灭弧罩内(图中未画出灭弧罩)。一旦产生电弧,电弧周围产生磁场,导磁的钢片将电弧吸入栅片,电弧被栅片分割成许多串联的短电弧。交流电压过零时,电弧自然熄灭。电弧要重燃,两栅片间必须有 150~250 V 电弧压降。这样,一方面电源电压不足以维持电弧,同时由于栅片的散热作用,电弧自然熄灭后很难重燃。这是一种常用的交流灭弧装置。

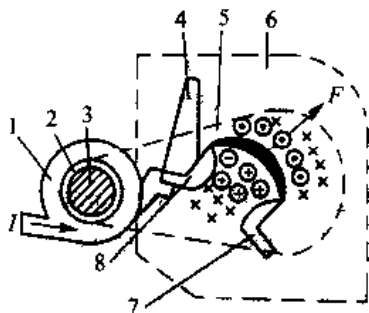


图 1-5 磁吹式灭弧原理图

1—吹弧线圈;2—绝缘套;3—铁心;4—引弧角  
5—导磁夹板;6—灭弧罩;7—动触点;8—静触点

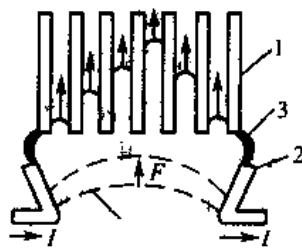


图 1-6 栅片灭弧原理图

1—灭弧栅片;2—触点;3—电弧

### (4) 灭弧罩

上面提到的磁吹式灭弧和灭弧栅灭弧都带有灭弧罩,它通常用耐弧陶土、石棉水泥或耐弧塑料制成。其作用一是分隔各路电弧,以防止发生短路。二是使电弧与灭弧罩的绝缘壁接触,使电弧迅速冷却而熄灭。

### 3. 电磁机构

电磁机构是电磁式低压电器的感测部件,它的作用是将电磁能量转换成机械能量,带动触头动作使之闭合或断开,从而实现电路的接通或分断。

电磁机构由磁路和激磁线圈两部分组成。磁路主要包括铁心、衔铁和空气隙。激磁线圈通以电流后激励磁场,通过气隙把电能转换为机械能,带动衔铁运动以完成触点的闭合或断开。

常用的磁路结构可分为三种形式,如图 1-7 所示。即衔铁沿棱角转动的拍合式铁心,如图 1-7(a)所示,这种形式广泛应用于直流电器中。衔铁沿轴转动的拍合式铁心,如图 1-7(b)所示,其铁心形状有 E 形和 U 形两种,此种结构多用于触点容量较大的交流电器中。衔铁直线运动的双 E 形直动式铁心,如图 1-7(c)所示,它多用于交流接触器、继电器中。

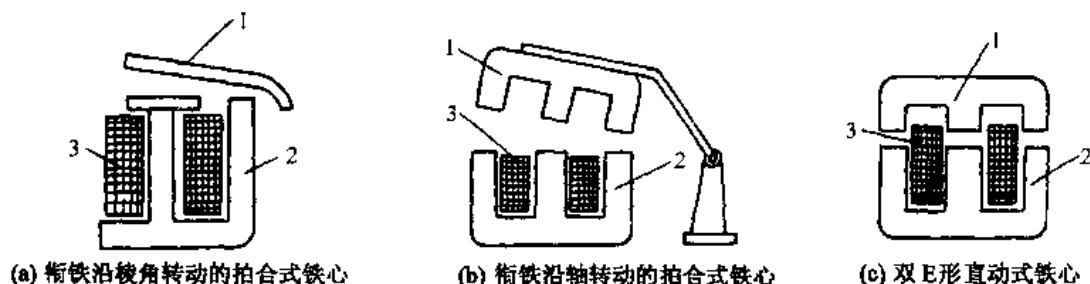


图 1-7 常用的磁路结构

1—衔铁;2—铁心;3—吸引线圈

激磁线圈的作用是将电能转换成磁场能量。按通入激磁线圈的电流种类不同,可分为直流线圈和交流线圈。与之对应的有直流电磁机构和交流电磁机构。

对于直流电磁机构,因其铁心不发热,只有线圈发热,所以通常直流电磁机构的铁心是用整块钢材或工程纯铁制成,而且它的激磁线圈做成高而薄的瘦高型,且不设线圈骨架,使线圈与铁心直接接触,易于散热。

对于交流电磁机构,由于其铁心存在磁滞和涡流损耗,这样铁心和线圈都发热,所以通常交流电磁机构的铁心用硅钢片叠铆而成,而且它的激磁线圈设有骨架,使铁心与线圈隔离并将线圈制成短而厚的矮胖型,这样有利于铁心和线圈的散热。

### 4. 电磁机构的工作原理

电磁机构的工作原理常用吸力特性和反力特性来表征。电磁机构使衔铁吸合的力与气隙长度的关系曲线称为吸力特性,电磁机构使衔铁释放(复位)的力与气隙长度的关系曲线称为反力特性。

(1) 反力特性 电磁机构使衔铁释放的力主要是弹簧的反力(忽略衔铁自身质量),弹簧的反力与其形变的位移  $X$  成正比,其反力特性可写成

$$F_{\text{反}} = K_1 X \quad (1-1)$$

考虑到常开触点闭合时超行程机构的弹力作用,上述反力特性如图 1-8 的曲线 3 所示,其中  $\delta_1$  为电磁机构气隙的初始值, $\delta_2$  为动、静触头开始接触时的气隙长度。由于超行程机构的弹力作用,反力特性在  $\delta_2$  处有一突变。

(2) 吸力特性 电磁机构的电磁吸力可以按下式求得

磁机构的磁通也发生相应的急剧变化,这会在激磁线圈中感生很大的反电势。此反电势可达线圈额定电压的 10~20 倍,易使线圈因过压而损坏。为此必须增加线圈放电回路,一般采用反并联二极管并加限流电阻来实现。

(3) 吸力特性与反力特性的配合 电磁机构欲使衔铁吸合,在整个吸合过程中,吸力都必须大于反力。但也不能过大,否则衔铁吸合时运动速度过大,会产生很大的冲击力,使衔铁与铁心柱端面造成严重的机械磨损。此外,过大的冲击力有可能使触点产生弹跳现象,导致触点的熔焊或磨损,降低触点的使用寿命。反映在特性图上就是要保持吸力特性在反力特性的上方且彼此靠近,如图 1-8 所示。对于直流电磁机构,当切断激磁电流以释放衔铁时,其反力特性必须大于剩磁吸力,才能保证衔铁可靠释放。

(4) 单相交流电磁机构短路环的作用 对于单相交流电磁机构,电磁吸力是一个两倍于电源频率的周期性变量。电磁机构在工作中,衔铁始终受到反力  $F_r$  的作用。由于交流磁通过零时吸力也为零,吸合后的衔铁在反力  $F_r$  作用下被拉开。磁通过零后吸力增大,当吸力大于反力时衔铁又被吸合。这样,在交流电每周期内衔铁吸力要两次过零,如此周而复始,使衔铁产生强烈的振动并发出噪声,甚至使铁心松散。因此必须采取有效措施予以克服。

具体办法是在铁心端部开一个槽,槽内嵌入称为短路环(或称分磁环)的铜环,如图 1-9 所示。短路环把铁心中的磁通分为两部分,即不穿过短路环的  $\varphi_1$  和穿过短路环的  $\varphi_2$ ,  $\varphi_2$  为原磁通与短路环中感生电流产生的磁通的叠加,且相位上  $\varphi_2$  滞后  $\varphi_1$ ,电磁机构的吸力  $F$  为它们产生的吸力  $F_1$ 、 $F_2$  的合力,如图 1-10 所示。此合力始终大于反力,所以衔铁的振动和噪声就消除了。

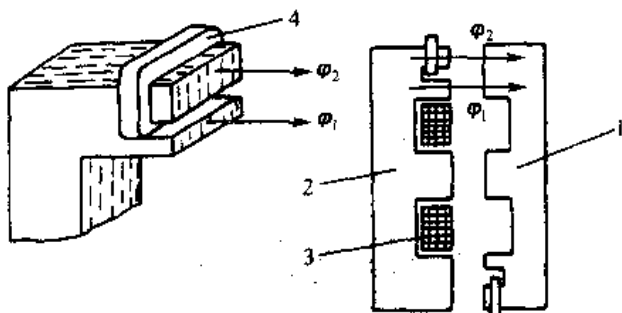


图 1-9 交流电磁铁的短路环  
1—衔铁;2—铁心;3—线圈;4—短路环

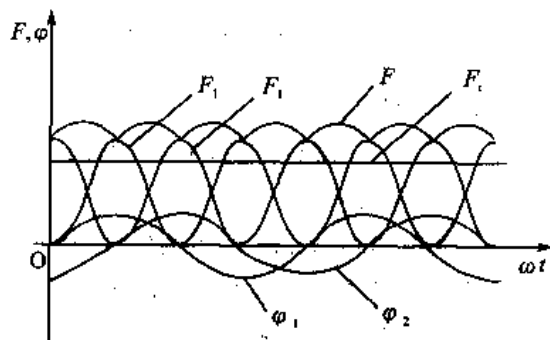


图 1-10 加短路环后的电磁吸力

短路环通常包围三分之二的铁心截面,它一般用铜、康铜或镍铬合金等材料制成。

(5) 电磁机构的输入/输出特性 电磁机构激磁线圈的电压(或电流)为其输入量,衔铁的位置为其输出量,衔铁的位置与激磁线圈的电压(或电流)的关系称为输入/输出特性。

若用  $y$  代表电磁机构的输出量,并将衔铁处于吸合位置记作  $y=1$ ,把衔铁处于释放位置记作  $y=0$ ;若用  $x$  表示电磁机构的输入量,使吸力特性处于反力特性上方的最小输入量以  $x_0$  表示,一般称为电磁机构的动作值。使吸力特性处于反力特性下方的最大输入量以  $x_1$  表示,一般称为电磁机构的返回值。

电磁机构的输入/输出特性如图 1-11 所示。当输

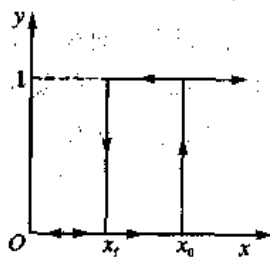


图 1-11 电磁机构的输入/输出特性

入量  $x$  由零增至  $x_0$  以前, 输出量  $y$  为零。当输入量  $x$  增至  $x_0$  时, 衔铁吸合, 输出量  $y$  为 1。如  $x$  再增大,  $y$  值保持不变。当  $x$  减小到  $x_1$  时, 衔铁释放, 输出量  $y$  降到零,  $x$  再减小,  $y$  值均为零。因此, 欲使衔铁吸合输入量必须大于等于  $x_0$ 。欲使衔铁释放输入量必须小于等于  $x_1$ 。

可见, 电磁机构的输入/输出特性为一矩形曲线, 此类矩形特性曲线也称为继电器特性。

## 1.2 接触器

接触器是用来接通或分断电动机主电路或其他负载电路的控制电器。用它可以实现频繁的远距离自动控制。由于它体积小, 价格低、寿命长、维护方便, 因而用途十分广泛。

### 1.2.1 接触器的用途及分类

接触器最主要的用途是控制电动机的启动、反转、制动和调速等, 因此它是电力拖动控制系统中最重要也是最常用的控制电器之一。它具有低电压释放保护功能。它具有比工作电流大数倍乃至十几倍的接通和分断能力, 但不能分断短路电流。它是一种执行电器, 即使在先进的可编程控制器应用系统中, 它一般也不能被取代。

接触器种类很多, 按驱动力不同可分为电磁式、气动式和液压式, 以电磁式应用最广泛。按接触器主触点控制的电路中电流种类分为交流接触器和直流接触器两种。按其主触点的极数(即主触点的个数)来分, 有单极、双极、三极、四极和五极等多种。本节介绍电磁式接触器。

### 1.2.2 接触器的结构及工作原理

#### 1. 接触器的结构

图 1-12 为交流接触器的结构剖面示意图。它由以下五个部分组成。

##### (1) 电磁机构

电磁机构由线圈、铁心和衔铁组成。铁心一般都是双 E 形衔铁直动式电磁机构, 有的衔铁采用绕轴转动的拍合式电磁机构。

##### (2) 主触点和灭弧系统

根据主触点的容量大小, 有桥式触点和指形触点两种结构形式, 且直流接触器和电流在 20 A 以上的交流接触器均装有灭弧罩, 有的还带有栅片或磁吹灭弧装置。

##### (3) 辅助触点

有常开和常闭辅助触点, 在结构上它们均为桥式双断点形式。其容量较小。接触器安装辅助触点的目的是使其在控制电路中起联动作用, 用于和接触器相关的逻辑控制。辅助触点不设灭弧装置, 所以它不能用来分合主电路。

##### (4) 反力装置

该装置由释放弹簧和触点弹簧组成, 且它们均不能进行弹簧松紧的调节。

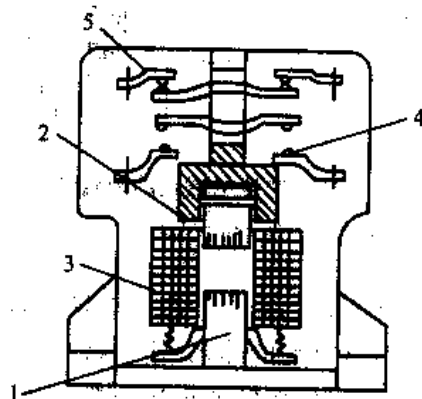


图 1-12 交流接触器的结构

1—铁心; 2—衔铁; 3—线圈;  
4—常开触点; 5—常闭触点

### (5) 支架和底座

它用于接触器的固定和安装。

## 2. 接触器的工作原理

当交流接触器线圈通电后,在铁心中产生磁通。由此在衔铁气隙处产生吸力,使衔铁产生闭合动作,主触点在衔铁的带动下闭合,于是接通了主电路。同时衔铁还带动辅助触点动作,使原来断开的辅助触点闭合,而原来闭合的辅助触点断开。当线圈断电或电压显著降低时,吸力消失或减弱,衔铁在释放弹簧作用下打开,主、辅触点又恢复到原来状态。这就是接触器的工作原理。

直流接触器的结构和工作原理与交流接触器基本相同,仅有电磁机构方面不同,在 1.1 节中有关部分已有阐述,这里不再赘述。

## 1.2.3 接触器的技术参数

### 1. 接触器的主要技术参数

#### (1) 额定电压

接触器铭牌上标注的额定电压是指主触点的额定电压。常用的额定电压等级如表 1-1 所列。

#### (2) 额定电流

接触器铭牌上标注的额定电流是指主触点的额定电流。常用额定电流等级如表 1-1 所列。表中的电流值是接触器安装在敞开式控制屏上,触点工作不超过额定温升,负荷为间断—长期工作制时的电流值。

#### (3) 线圈的额定电压

常用的额定电压等级如表 1-1 所列。选用时交流负载选用交流接触器,直流负载选用直流接触器,但交流负载频繁动作时可采用直流线圈的交流接触器。

表 1-1 接触器的额定电压和额定电流及线圈额定电压的等级表

名 称	直 流 接 触 器	交 流 接 触 器
额定电压/V	110,220,440,660	127,220,380,500,660
额定电流/A	5,10,20,40,60,100,150,250,400,600	5,10,20,40,60,100,150,250,400,600
线圈电压/V	24,48,110,220,440	36,110,127,220,380

#### (4) 接通和分断能力

指主触点在规定条件下能可靠地接通和分断的电流值。在此电流值下,接通时主触点不应发生熔焊,分断时主触点不应发生长时间燃弧。

接触器的使用类别不同对主触点的接通和分断能力的要求也不一样,而不同类别的接触器是根据其不同控制对象(负载)的控制方式所规定的。根据低压电器基本标准的规定,其使用类别比较多。但在电力拖动控制系统中,常见的接触器使用类别及其典型用途如表 1-2 所列。

表 1-2 常见接触器使用类别及其典型用途表

电 流 种 类	使 用 类 别	典 型 用 途
交流 (AC)	AC1	无感或微感负载、电阻炉
	AC2	绕线式电动机的启动和分断
	AC3	笼型电动机的启动和分断
	AC4	笼型电动机的启动、反接制动、反向和点动
直流 (DC)	DC1	无感或微感负载、电阻炉
	DC3	并励电动机的启动、反接制动、反向和点动
	DC5	串励电动机的启动、反接制动、反向和点动

接触器的使用类别代号通常标注在产品的铭牌或工作手册中。表 1-2 中要求接触器主触点达到的接通和分断能力为:

- ① AC1 和 DC1 类允许接通和分断额定电流;
- ② AC2、DC3 和 DC5 类允许接通和分断 4 倍的额定电流;
- ③ AC3 类允许接通 6 倍的额定电流和分断额定电流;
- ④ AC4 类允许接通和分断 6 倍的额定电流。

#### (5) 额定操作频率

指每小时(h)的操作次数。交流接触器最高为 600 次/h,而直流接触器最高为 1 200 次/h。操作频率直接影响到接触器的寿命,对于交流接触器还影响到线圈的温升。

### 2. 接触器的图形符号和文字符号

接触器在电路图中的图形符号如图 1-13 所示,文字符号为 KM。



图 1-13 接触器的图形和文字符号

### 1.2.4 接触器的选择

接触器使用广泛,其额定工作电流或额定控制功率随使用条件不同而变化,只有根据不同使用条件正确选用,才能保证接触器可靠运行。接触器选用主要依据以下几个方面。

#### 1. 根据所控制负载的工作任务选择相应使用类别的接触器

生产中广泛使用中小容量的笼型电动机,其中大部分负载是一般任务,它相当于 AC3 使用类别。对于控制机床电动机的接触器,其负载情况比较复杂,既有 AC3 类的也有 AC4 类的,还有 AC1 和 AC4 类混合的负载,这些都属于重任务范畴。如果负载明显地属于重任务类,则应选用 AC4 类接触器。如果负载为一般任务与重任务混合的情况,则应根据实际情况选用 AC3 或 AC4 类接触器。若确定选用 AC3 类接触器,它的容量应降低一级使用,即使这样,其寿命仍有不同程度的降低。

适用于 AC2 类的接触器,一般也不宜用来控制 AC3 及 AC4 类的负载,因为它的接通能力较低,在频繁接通这类负载时容易发生触点熔焊现象。

#### 2. 根据电动机(或其他负载)的功率和操作情况确定接触器主触点的电流等级

当接触器的使用类别与所控制负载的工作任务相对应时,一般应使主触点的电流等级与所控制的负载相当,或稍大一些。若不对应,例如用 AC3 类的接触器控制 AC3 与 AC4 混合类负载时,则须降低电流等级使用。

### 3. 接触器线圈的电压等级应与控制电路相同

接触器的线圈电压和额定电压是不同的概念,线圈电压应和控制电路的电压类型和等级相同。

目前我国常用的交流接触器主要有:CJ20、CJX1、CJX2、CJ12 和 CJ10 等系列。而引进产品应用较多的有:德国 BBC 公司制造的 B 系列,德国 SIEMENS 公司的 3TB 系列,法国 TE 公司的 LC1 系列等。常用的直流接触器有 CZ18、CZ21、CZ22、CZ10 和 CZ2 等系列,CZ18 系列是取代 CZ0 系列的新产品。

最后,再介绍一种实际工作中选择接触器的简单方法。接触器是电气控制系统中不可缺少的执行器件,而三相笼型电机也是最常用的被控对象。表 1-3 给出了代表性的新型 CJ20 系列接触器的主要参数,以及它所能控制的电动机的功率。从该表中的对应关系可以看出,对额定电压为 380 V 的接触器,如果知道了电动机的额定功率,则相应的接触器的额定电流数值约是电机功率数值的 2 倍。记住这个关系,对在实际工作中迅速选择接触器非常有用。

表 1-3 CJ20 系列交流接触器主要技术参数及可控制电动机最大功率一览表

型 号	额定电压/V	额定电流/A	可控制电动机最大功率/kW
CJ20-10	380	10	2.2
CJ20-25	380	25	11
CJ20-40	380	40	22
CJ20-63	380	63	30
CJ20-100	380	100	50
CJ20-160	380	160	85
CJ20-400	380	400	200
CJ20-630	380	630	300

## 1.3 继电器

继电器是根据某种输入信号来接通或断开小电流控制电路,实现远距离控制 and 保护的自动控制电器。其输入量可以是电流、电压等电量,也可以是温度、时间、速度、压力等非电量,而输出则是触头的动作或者是电路参数的变化。

继电器的种类很多,按输入信号的性质分为:电压继电器、电流继电器、时间继电器、温度继电器、速度继电器、压力继电器等。按工作原理分为:电磁式继电器、感应式继电器、电动式继电器、热继电器和电子式继电器等。按输出形式分为:有触点和无触点两类。按用途分为:控制用和保护用继电器等。本节介绍几种常用的继电器。

### 1.3.1 电磁式继电器

电磁式继电器结构简单、价格低廉、使用维护方便,广泛应用于控制系统中。

电磁式继电器的结构和工作原理与电磁式接触器相似,也是由电磁机构和触点系统组成。主要区别在于:继电器可对多种输入量的变化作出反应,而接触器只有在一定的电压信号

作用下动作;继电器用于切换小电流的控制电路和保护电路,而接触器用来控制大电流电路;继电器没有灭弧装置,也无主辅触点之分等。

### 1. 电压继电器

触点的动作与线圈的电压大小有关的继电器称为电压继电器。它用于电力拖动系统的电压保护和控制。使用时电压继电器的线圈与负载并联。其线圈的匝数多而线径细。按线圈电流的种类可分为交流和直流电压继电器;按吸合电压大小又可分为过电压和欠电压继电器。

对于过电压继电器,当线圈为额定电压时,衔铁不产生吸合动作,只有当线圈电压高于其额定电压的某一值时衔铁才产生吸合动作,所以称为过电压继电器。因为直流电路不会产生波动较大的过电压现象,所以在产品中没有直流过电压继电器。交流过电压继电器在电路中做电压保护作用。

对于欠电压继电器,当线圈的承受电压低于其额定电压时,衔铁就产生释放动作。它的特点是释放电压很低,在电路中做低电压保护。

电压继电器的图形符号如图 1-14(a)所示,文字符号为 KV。

电压继电器选用时,首先要注意线圈电流的种类和电压等级应与控制电路一致。另外,根据在控制电路中的作用(是过电压还是欠电压)选型。最后,要按控制电路的要求选触点的类型(是常开还是常闭)和数量。

### 2. 电流继电器

触点的动作与线圈电流大小有关的继电器叫做电流继电器。使用时电流继电器的线圈与负载串联。其线圈的匝数少而线径粗。根据线圈的电流种类分为交流和直流电流继电器;按吸合电流大小可分为过电流继电器和低(欠)电流继电器。

对于过电流继电器,正常工作时,线圈中流有负载电流,但不产生吸合动作。当出现比负载工作电流大的吸合电流时,衔铁才产生吸合动作,从而带动触点动作。在电力拖动系统中,冲击性的过电流故障时有发生,常采用过电流继电器做电路的过电流保护。

对于低电流继电器,正常工作时,由于电路的负载电流大于吸合电流而使衔铁处于吸合状态。当电路的负载电流降低至释放电流时,则衔铁释放。在直流电路中,由于某种原因而引起负载电流的降低或消失往往导致严重的后果(如直流电动机的励磁回路断线),因此在产品上有直流低电流继电器,而没有交流低电流继电器。

电流继电器图形符号如图 1-14(b)所示,文字符号为 KI。

电流继电器选用时首先要注意线圈电流的种类和等级应与负载电路一致。另外,根据对负载的保护作用(是过电流还是低电流)来选用电流继电器的类型。最后,要根据控制电路的要求选触点的类型(是常开还是常闭)和数量。

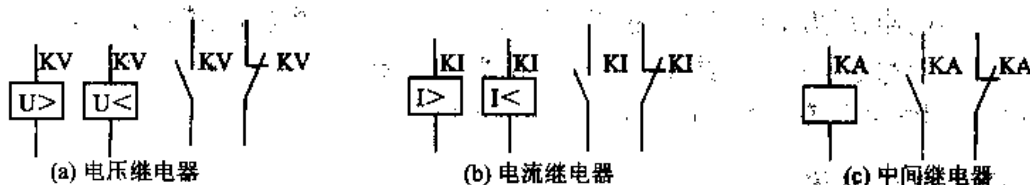


图 1-14 电磁式继电器的图形和文字符号

### 3. 中间继电器

在控制电路中起信号传递、放大、翻转和分路等中继作用的继电器称为中间继电器。它属

于电压继电器的一种,主要用于扩展触点数量,实现逻辑控制。中间继电器也有交、直流之分,可分别用于交流控制电路和直流控制电路。中间继电器的图形符号如图 1-14(c)所示,文字符号为 KA。

中间继电器的主要技术参数有额定电压、额定电流、触点对数以及线圈电压种类和规格等。选用时要注意线圈的电流种类和电压等级应与控制电路一致。另外,要根据控制电路的需求来确定触点的形式和数量。当一个中间继电器的触点数量不够用时,也可以将两个中间继电器并联使用,以增加触点的数量。

常用的中间继电器有 JZ7 系列,以 JZ7-62 为例,JZ 为中间继电器的代号,7 为设计序号,有 6 对常开触点,2 对常闭触点。表 1-4 为 JZ7 系列的主要技术数据。

表 1-4 JZ7 系列中间继电器的技术数据

型 号	触点额定 电压/V	触点额定 电流/A	触 点 对 数		吸引线圈 电压/V	额定操作 频率/(次/h)
			常 开	常 闭		
JZ7-44	500	5	4	4	交流 50 Hz 系列为 12,36,127,220,380	1 200
JZ7-62			6	2		
JZ7-80			8	0		

新型中间继电器触点在闭合过程中,其动、静触点间有一段滑擦、滚压过程。该过程可以有效地清除触点表面的各种生成膜及尘埃,减小了接触电阻,提高了接触可靠性。有的还装了防尘罩或采用密封结构,也是提高可靠性的措施。有些中间继电器安装在插座上,插座有多种型号可供选择;有些中间继电器可直接安装在导轨上,安装和拆卸均很方便。

### 1.3.2 热继电器

#### 1. 热继电器的作用和分类

在电力拖动控制系统中,当三相交流电动机出现长期带负荷欠电压运行、长期过载运行以及长期单相运行等不正常情况时,会导致电动机绕组严重过热乃至烧坏。为了充分发挥电动机的过载能力,保证电动机的正常启动和运转,而当电动机一旦出现长时间过载时又能自动切断电路,从而出现了能随过载程度而改变动作时间的电器,这就是热继电器。热继电器利用电流的热效应原理以及发热元件热膨胀原理设计,实现三相交流电动机的过载保护。但须指出的是,由于热继电器中发热元件有热惯性,在电路中不能做瞬时过载保护,更不能做短路保护。因此,它不同于过电流继电器和熔断器。

按相数来分,热继电器有单相、两相和三相式共三种类型,每种类型按发热元件的额定电流的不同又有不同的规格和型号。三相式热继电器常用于三相交流电动机做过载保护。按功能来分,三相式热继电器又有不带断相保护和带断相保护两种类型。

#### 2. 电动机的过载特性和热继电器的保护特性

因为热继电器的触点动作时间与被保护的电动机过载程度有关,所以在分析热继电器工作原理之前,首先要明确电动机在不超过允许温升的条件下,电动机的过载电流与电动机通电时间的关系。这种关系称为电动机的过载特性。

当电动机运行中出现过载电流时,必将引起绕组发热。根据热平衡关系,不难得出在允许

温升条件下,电动机通电时间与其过载电流的平方成反比的结论。根据这个结论,可以得知电动机的过载特性具有反时限特性,如图 1-15 中的曲线 1 所示。

为了适应电动机的过载特性而又起到过载保护作用,要求热继电器也应具有如同电动机过载特性那样的反时限特性。为此,热继电器中必须具有电阻性发热元件,利用过载电流通过电阻发热元件产生的热效应使感测元件动作,从而带动触点动作来实现保护作用。热继电器中通过的过载电流与热继电器触点动作时间的关系,称为热继电器的保护特性,如图 1-15 中的曲线 2 所示。考虑到各种误差的影响,电动机的过载特性和继电器的保护特性都不是一条曲线,而是一条带子。显见,误差越大,带子越宽;误差越小,带子越窄。

由图 1-15 中的曲线 1 可知,电动机出现过载时,工作在曲线 1 的下方是安全的。因此,热继电器的保护特性应在电动机过载特性的邻近下方。这样,如果发生过载,热继电器就会在电动机未达到其允许过载极限之前动作,切断电动机电源,使之免遭损坏。

### 3. 热继电器工作原理

热继电器主要由热元件、双金属片、触头系统等组成。双金属片是热继电器的感测元件,它由两种不同线膨胀系数的金属片用机械碾压而成。线膨胀系数大的称为主动层,小的称为被动层。图 1-16 是热继电器的结构原理图。热元件 3 串接在电动机的定子绕组中,电动机定子绕组电流即为流过热元件的电流。当电动机正常运行时,热元件产生的热量虽能使双金属片 2 弯曲,但还不足以使继电器动作。当电动机过载时,热元件产生的热量增大,使双金属片弯曲位移增大,经过一定时间后,双金属片弯曲到推动导板 4,并通过补偿双金属片 5 与推杆 14 将触点 9 和 6 分开,触点 9 和 6 为热继电器串于接触器线圈回路的常闭触点,断开后使接触器线圈失电,接触器的主触点断开电动机的电源以保护电动机。

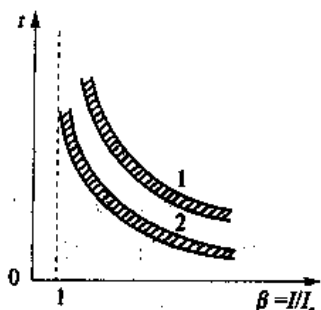


图 1-15 电动机的过载特性和热继电器的保护特性及其配合

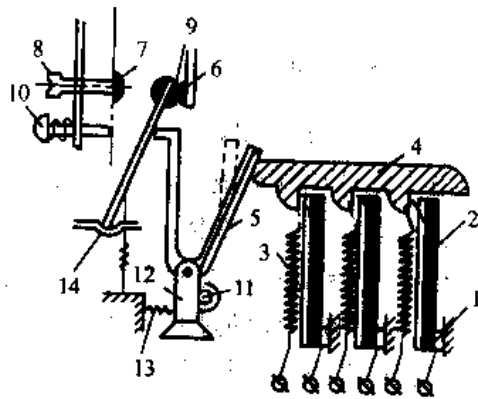


图 1-16 热继电器的结构原理图

调节旋钮 11 是一个偏心轮,它与支撑件 12 构成一个杠杆,13 是一压簧,转动偏心轮,改变它的半径即可改变补偿双金属片 5 与导板 4 的接触距离,因而达到调节整定热继电器动作电流的目的。此外,靠调节复位螺钉 8 来改变常开触点 7 的位置使热继电器能工作在手动复位和自动复位两种工作状态。调试手动复位时,在故障排除后要按下按钮 10 才能使动触点恢复与静触点 6 相接触的位置。

在电气原理图中,热继电器的热元件、触点的图形符号和文字符号如图 1-17 所示。

### 4. 带断相保护的热继电器

三相电动机的一根接线松开或一相熔丝熔断,是造成三相异步电动机烧坏的主要原因之

一。如果热继电器所保护的电动机为 Y 形接法,当线路发生一相断电时,另外两相电流便增大很多,由于线电流等于相电流,流过电动机绕组的电流和流过热继电器的电流增加的比例相同,因此普通的两相或三相热继电器可以对此作出保护。

如果电动机是  $\Delta$  形接法,发生断相时,由于电动机

的相电流与线电流不等,流过电动机绕组的电流和流过热继电器的电流增加比例不相同,而热元件又串联在电动机的电源进线中,按电动机的额定电流即线电流来整定,整定值较大。当故障线电流达到额定电流时,在电动机绕组内部,电流较大的那一相绕组的故障电流将超过额定相电流,便有过热烧毁的危险。所以  $\Delta$  形接法必须采用带断相保护的热继电器。

带有断相保护的热继电器是在普通热继电器的基础上增加一个差动机构,对三个电流进行比较。差动式断相保护装置结构原理如图 1-18 所示。热继电器的导板改为差动机构,由上导板 1、下导板 2 及杠杆 5 组成,它们之间都用转轴连接。图 1-18(a) 为通电前机构各部件的位置。图 1-18(b) 为正常通电时的位置,此时三相双金属片都受热向左弯曲,但弯曲的挠度不够,所以下导板向左移动一小段距离,继电器不动作。图 1-18(c) 是三相同同时过载时的情况,三相双金属片同时向左弯曲,推动下导板 2 向左移动,通过杠杆 5 使常闭触点立即打开。图 1-18(d) 是 C 相断线的情况,这时 C 相双金属片逐渐冷却降温,端部向右移动推动上导板 1

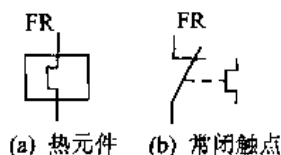


图 1-17 热继电器的图形和文字符号

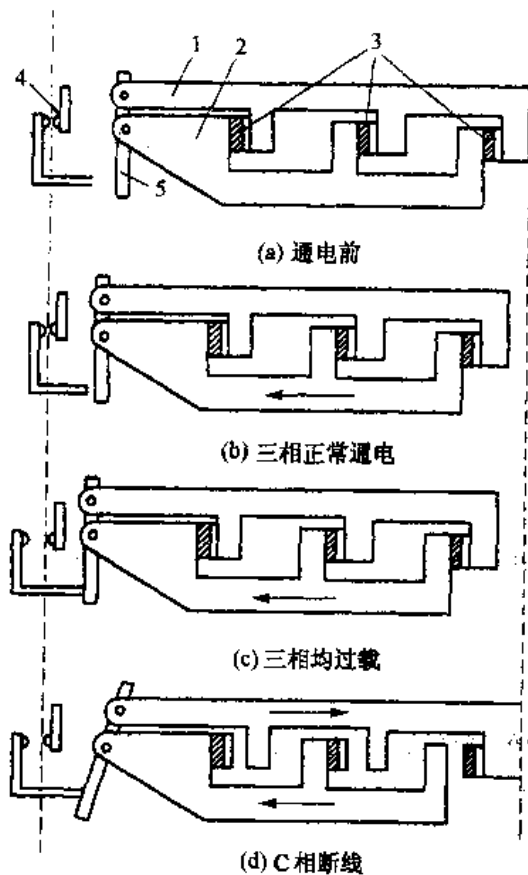


图 1-18 热继电器差动式断相保护机构动作原理图

1—上导板;2—下导板;3—双金属片;4—常闭触点;5—杠杆

向右移。而另外两相双金属片温度上升,端部向左弯曲,推动下导板 2 继续向左移动。由于上、下导板一左一右移动,产生了差动作用,通过杠杆放大作用,使常闭触点打开。由于差动作用,使热继电器在断相故障时加速动作,实现保护电动机的目的。

### 5. 热继电器的技术参数及常用产品

热继电器的主要技术参数有额定电压、额定电流、相数、热元件编号以及整定电流调节范围等。

热继电器的整定电流是指热继电器的热元件允许长期通过又不致引起继电器动作的电流值,对于某一热元件,可通过调节电流旋钮,在一定范围内调节其整定电流。

常用的热继电器有 JR20、JRS1、T、3UA 和 LR1-R 等系列, JR20、JRS1 系列具有断相保护功能。其中 T 系列是引进德国 ABB 公司技术生产, 3UA 系列是引进德国 SIEMENS 公司技术生产, LR1-D 系列是引进法国 TE 公司技术生产。每一系列的热继电器一般只能和相适应系列的接触器配套使用,如 JR20 系列热继电器与 CJ20 接触器配套使用, 3UA 系列热继电器与 3TB/3TF/3TW 等系列接触器配套使用, T 系列热继电器与 B 系列接触器配套使用等,但也有例外情况。

安装方式上有独立安装式(通过螺钉固定)、导轨安装式(在标准导轨上安装)和接插安装式(直接挂接在其配套的接触器上)。

### 6. 热继电器的选用

热继电器选用是否得当,直接影响着对电动机进行过载保护的可靠性。通常选用时应按电动机形式、工作环境、启动情况及负荷情况等几方面综合加以考虑。

#### (1) 原则上热继电器的额定电流应按电动机的额定电流选择

对于过载能力较差的电动机,其配用的热继电器(主要是发热元件)的额定电流可适当小些。通常,选取热继电器的额定电流(实际上是选取发热元件的额定电流)为电动机额定电流的 60%~80%。

#### (2) 在不频繁启动场合,要保证热继电器在电动机的启动过程中不产生误动作

通常,当电动机启动电流为其额定电流 6 倍以及启动时间不超过 6 s 且很少连续启动时,就可按电动机的额定电流选取热继电器。

#### (3) 当电动机为重复且短时工作制时,要注意确定热继电器的允许操作频率

因为热继电器的操作频率是很有限的,如果用它保护操作频率较高的电动机,效果很不理想,有时甚至不能使用。

对于可逆运行和频繁通断的电动机,不宜采用热继电器保护,必要时可以选用装入电动机内部的温度继电器。

## 1.3.3 时间继电器

从得到输入信号(线圈的通电或断电)开始,经过一定的延时后才输出信号(触点的闭合或断开)的继电器,称为时间继电器。

时间继电器的延时方式有两种,即通电延时和断电延时。

通电延时:接受输入信号后延迟一定的时间,输出信号才发生变化;当输入信号消失后,输出瞬时复原。

断电延时:接受输入信号时,瞬时产生相应的输出信号;当输入信号消失后,延迟一定的时

间,输出才复原。

时间继电器的图形符号如图 1-19 所示,文字符号为 KT。

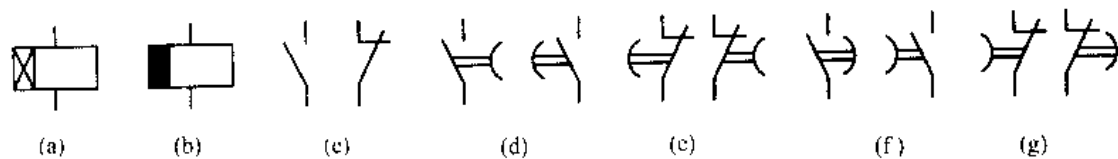


图 1-19 时间继电器的图形符号

(a) 通电延时线圈;(b) 断电延时线圈;(c) 瞬动触点;(d) 通电延时闭合常开触点;  
(e) 通电延时断开常闭触点;(f) 断电延时断开常开触点;(g) 断电延时闭合常闭触点

时间继电器按工作原理分类,有电磁式、电动式、空气阻尼式、电子式等。其中,电子式时间继电器近几年发展十分迅速,这类时间继电器除执行器件继电器外,均由电子元件组成,没有机械部件,因而具有寿命长、精度高、体积小、延时范围大、控制功率小等优点,已得到广泛应用。

电子式时间继电器的品种和类型很多。表 1-5 给出了 JS20 系列时间继电器主要技术参数,它有通电延时型、断电延时型、带瞬动触点的通电延时型等三种类型。

表 1-5 JS20 系列晶体管时间继电器的主要技术参数

产品名称	额定工作电压/V		延时等级/s
	交流	直流	
通电延时继电器	26,110,127,220,380	24,48,110	1,5,10,30,60,120,180,240,300,600,900
瞬动延时继电器	36,110,127,220		1,5,10,30,60,120,180,240,300,600
断电延时继电器	36,110,127,220,380	—	1,5,10,30,60,120,180

目前,一些厂家生产的 JS14P、JS14S、JSS1、JS11S 和 JS11 系列电子式时间继电器均采用拨码开关整定延时时间,采用显示器件直接显示定时时间和工作状态,具有直观、准确、使用方便等特点。其具体技术参数可查阅产品说明书。

此外,已有厂家引进了目前国际上最新式的 ST 系列超级时间继电器,其内部装有时间继电器专用的大规模集成电路,并使用高质量薄膜电容器和金属陶瓷可变电阻器,减少了元器件的数量,缩小了体积,增加了可靠性,提高了抗干扰能力。另外采用了高精度振荡电路和高频率分频电路,保证了高精度和长延时。因此,它是一种体积小、质量轻、可靠性极高的小型时间继电器。

#### 1.3.4 速度继电器

按速度原则动作的继电器,称为速度继电器。它主要应用于三相笼型异步电动机的反接制动中,因此又称为反接制动控制器。

感应式速度继电器主要由定子、转子和触点三部分组成,转子是一个圆柱型永久磁铁,定子是一个笼型空心圆环,由硅钢片叠制而成,并装有笼型绕组。

图 1-20 为感应式速度继电器原理示意图。其转子的轴与被控电机的轴相连接,当电动

机转动时,速度继电器的转子随之转动,到达一定转速时,定子产生的感应电流和力矩的作用下跟随转动;到达一定角度时,装在定子轴上的摆锤推动簧片(动触点)动作,使常闭触点打开、常开触点闭合;当电动机转速低于某一数值时,定子产生的转矩减小,触点在簧片作用下返回到原来位置,使对应的触点恢复到原来状态。

速度继电器的图形及文字符号如图 1-21 所示。

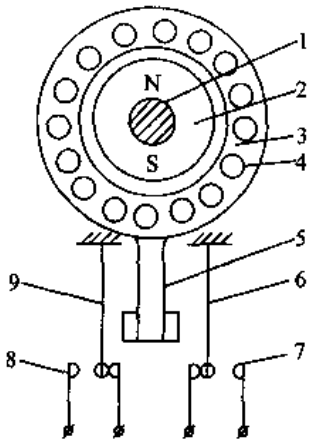


图 1-20 感应式速度继电器的原理示意图

1—转轴;2—转子;3—定子;4—绕组;  
5—摆锤;6、9—簧片;7、8—静触点

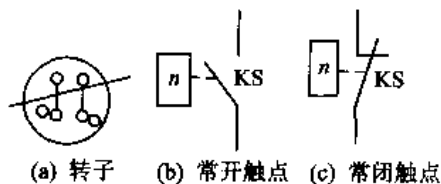


图 1-21 速度继电器的图形和文字符号

常用的感应式速度继电器有 JY1 和 JFZ0 系列。JY1 系列能在 3 000 r/min 以下可靠地工作。JFZ0-1 型适用于 300~1 000 r/min, JFZ0-2 型适用于 1 000~3 600 r/min。一般感应式速度继电器转轴在 120 r/min 左右时触点动作,在 100 r/min 以下时触点复位。

### 1.3.5 温度继电器

当电动机发生过电流时,会使其绕组温升过高,前已述及,热继电器可以起到保护作用。但当电网电压不正常升高时,即使电动机不过载,也会导致铁损增加而使铁心发热,这样也会使绕组温升过高;或者电动机环境温度过高以及通风不良等,也同样会使绕组温升过高。在这些情况下,若用热继电器则不能正常反映电动机的故障状态。为此,需要一种利用发热元件间接反映绕组温度并根据绕组温度进行动作的继电器,这种继电器称为温度继电器。

温度继电器大体上有两种类型:一种是双金属片式温度继电器;另一种是热敏电阻式温度继电器。以下介绍双金属片式温度继电器。

双金属片式 JW2 系列温度继电器结构组成如图 1-22 所示。在结构上它是封闭式的,其内部有盘式双金属片 2。双金属片受热后产生线膨胀,由于两层金属的线膨胀系数不同,且两层金属又紧密地贴合在一起,因此,使得双金属片向被动层一侧弯曲,由双金属片弯曲产生的机械力带动触点动作。

JW2 系列温度继电器的双金属片 2 左面为主动层,右面为被动层。动触点 8 铆在双金属片上,且经由导电片 3、外壳 1 与连接片 9 相连,静触点 6 与连接片 4 相连。当电动机发热部位温度升高时,产生的热量通过外壳 1 传导给其内部的双金属片,当达到一定温度时双金属片开始变形,双金属片及动触点向图中左方瞬动地跳开,从而控制接触器使电动机断电以达到过热

保护的。当故障排除后,发热部位温度则降低,双金属片也反向弹回而使触点重新复位。双金属片式温度继电器的动作温度是以电动机绕组绝缘等级为基础来划分的,它共有  $50^{\circ}\text{C}$ 、 $60^{\circ}\text{C}$ 、 $70^{\circ}\text{C}$ 、 $80^{\circ}\text{C}$ 、 $95^{\circ}\text{C}$ 、 $105^{\circ}\text{C}$ 、 $115^{\circ}\text{C}$ 、 $125^{\circ}\text{C}$ 、 $135^{\circ}\text{C}$ 、 $145^{\circ}\text{C}$  和  $165^{\circ}\text{C}$  共 11 个规格。继电器的返回温度因动作温度而异,一般比动作温度低  $5\sim 40^{\circ}\text{C}$ 。

双金属片式温度继电器用做电动机保护时,是将其埋设在电动机发热部位,如电动机定子槽内、绕组端部等,可直接反映该处发热情况,无论是电动机本身出现过载电流引起温度升高,还是其他原因引起电动机温度升高,温度继电器都可引起保护作用。不难看出,温度继电器具有“全热保护”作用。此外,双金属片式温度继电器因价格便宜,常用于热水器外壁、电热锅炉炉壁的过热保护。

双金属片式温度继电器的缺点是加工工艺复杂,且双金属片又容易老化。另外,由于体积偏大而多置于绕组的端部,故很难直接反映温度上升的情况,以致发生动作滞后的现象。同时,也不宜保护高压电动机,因为过强的绝缘层会加剧动作的滞后现象。

温度继电器的触点在电路图中的图形符号与中间继电器相同,文字符号为  $K\theta$ 。

### 1.3.6 液位继电器

某些锅炉和水柜需根据液位的高低变化来控制水泵电动机的启停,这一控制可由液位继电器来完成。

图 1-23 为液位继电器的结构示意图。浮筒置于被控锅炉或水柜内,浮筒的一端有一根磁钢,锅炉外壁装有一对触点,动触点的一端也有一根磁钢,它与浮筒一端的磁钢相对应。当锅炉或水柜内的水位降低到极限值时,浮筒下落,使磁钢端绕支点 A 上翘。由于磁钢同性相斥的作用,使动触点的磁钢端被斥下落,通过支点 B 使触点 1-1 接通,2-2 断开。反之,水位升高到上限位置时,浮筒上浮使触点 2-2 接通,1-1 断开。显然,液位继电器的安装位置决定了被控的液位,它主要用于不精确的液位控制场合。

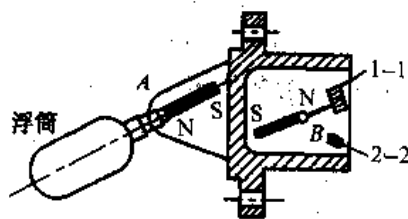


图 1-23 JYF-02 液位继电器结构示意图

### 1.3.7 固态继电器

固态继电器(简称 SSR)是采用固体半导体元件组装而成的一种新颖的无触点开关。由于固态继电器的接通和断开没有机械接触部件,因而具有控制功率小、开关速度快、工作频率高、使用寿命长、抗干扰能力强和动作可靠等一系列特点。目前,在许多自动控制装置中得到了广泛应用。

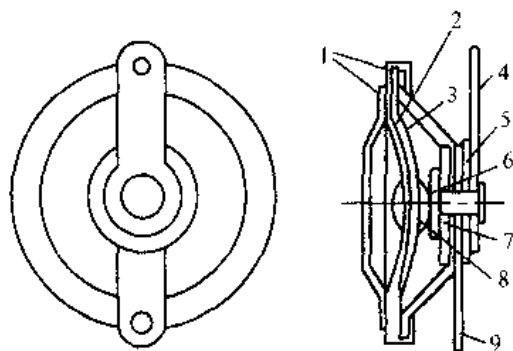


图 1-22 JW2 系列温度继电器

1—外壳;2—双金属片;3—导电片;4,9—连接片;  
5,7—绝缘垫片;6—静触点;8—动触点

### 1. 固态继电器的种类

固态继电器是四端器件,其中两端为输入端,两端为输出端,中间采用隔离器件,以实现输入与输出之间的隔离。

- ① 按切换负载性质分,有直流固态继电器和交流固态继电器。
- ② 按输入与输出之间的隔离分,有光电隔离固态继电器和磁隔离固态继电器。
- ③ 按控制触发信号方式分,有过零型和非过零型、有源触发型和无源触发型。

### 2. 固态继电器的工作原理

现以光电耦合式 SSR 为例来说明。如图 1-24 所示,当无信号输入时,光电耦合器中的光敏三极管  $V_3$  截止,  $V_4$  导通,  $VT_1$  控制极被箝在低电位而关断。当有信号输入时,光敏三极管  $V_3$  导通,  $V_4$  截止。当电源电压大于过零电压(约  $\pm 25\text{ V}$ ),  $A$  点电压大于  $V_5$  的  $V_{be5}$ ,  $V_5$  导通,  $VT_1$  由于控制极处于低电压而截止,输出端因  $VT_2$  控制极无触发信号而关断。当电源电压小于过零电压时,  $A$  点电压小于  $V_5$  的  $V_{be5}$ ,  $V_5$  截止,  $VT_1$  控制极通过  $R_5$ 、 $R_6$  分压获得触发信号,  $VT_1$  导通,这样  $VT_2$  控制极获得从  $R_8$ 、 $V_6$ 、 $VT_1$ 、 $V_9$ 、 $R_9$  和  $R_9$ 、 $V_8$ 、 $VT_1$ 、 $V_7$ 、 $R_8$  正反两个方向的触发脉冲,使  $VT_2$  导通,即输出端  $B$ 、 $C$  两点导通,接通负载电路。当输入信号取消后,  $V_4$  导通,  $VT_1$  关断,以此  $VT_2$  仍保持导通状态,直到负载电流随电源电压的下降到双向晶闸管维持电流以下而关断,从而切断负载电路。

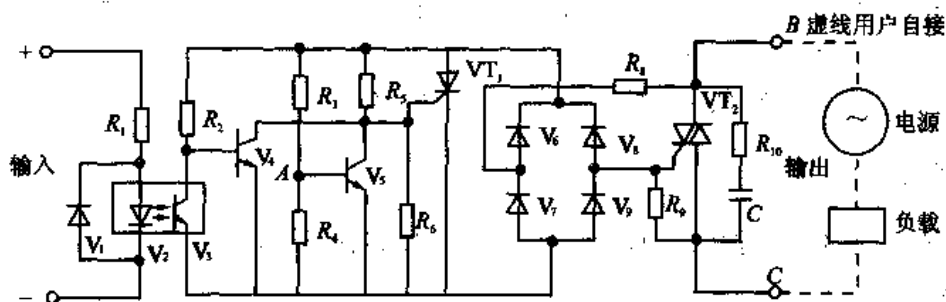


图 1-24 固态继电器的工作原理

输入端仅需一定量的电压和电流就能切换几安培甚至几百安培的大电流,且与晶体管、TTL、CMOS 电路有较好的兼容性,可直接与弱电回路连接。常用的产品有 DJ 系列固态继电器,其主要技术指标如表 1-6 所列。

表 1-6 DJ 系列固态继电器的主要技术参数

额定电压	额定电流/A	输出高电压	输出低电压	门限值
220 VAC50 Hz $\pm$ 10 %	1, 3, 5, 19	$\geq 95\%$ 电源电压	$\leq 5\%$ 电源电压	0.5~10 k $\Omega$
环境温度	开启时间	关闭时间	绝缘电阻	击穿电压
-10 $^{\circ}\text{C}$ ~+40 $^{\circ}\text{C}$	$\leq 1\text{ ms}$	$\leq 10\text{ ms}$	$\geq 100\text{ M}\Omega$	$\geq 2\,500\text{ VAC}$

### 3. 固态继电器使用的注意事项

- ① 固态继电器的选择应根据负载的类型(阻性、感性)来确定,并要采用有效的过压保护。
- ② 输出端要采用 RC 浪涌吸收回路或非线性压敏电阻吸收瞬变电压。
- ③ 过流保护应采用专门保护半导体器件的熔断器或用动作时间小于 10 ms 的自动开关。
- ④ 安装时采用散热器,要求接触良好,且对地绝缘。

⑤ 切忌负载侧两端短路,以免固态继电器损坏。

当然,尽管 SSR 有众多优点,但与传统的继电器相比,仍有不足之处,如漏电流大,接触电压大,触点单一,使用温度范围窄,过载能力差及价格偏高等。

## 1.4 开关电器

开关电器广泛用于配电系统和电力拖动控制系统,用做电源的隔离、电气设备的保护和控制。

### 1.4.1 刀开关

刀开关俗称闸刀开关,是一种结构最简单、应用最广泛的一种手动电器。主要用于接通和切断长期工作设备的电源及不经常启动及制动、容量小于 7.5 kW 的异步电动机。

刀开关主要由操作手柄、触刀、触点座和底座组成。依靠手动来实现触刀插入触点座与脱离触点座的控制。按刀数可分为单极、双极和三极。

刀开关在选择时,应使其额定电压等于或大于电路的额定电压。其电流应等于或大于电路的额定电流。当用刀开关控制电动机时,其额定电流要大于电动机额定电流的三倍。

刀开关在安装时,手柄要向上,不得倒装或平装,避免由于重力自由下落,而引起误动作和合闸。接线时,应将电源线接在上端,负载线接在下端,这样拉闸后刀片与电源隔离,防止可能发生的意外事故。

刀开关的图形、文字符号如图 1-25 所示。

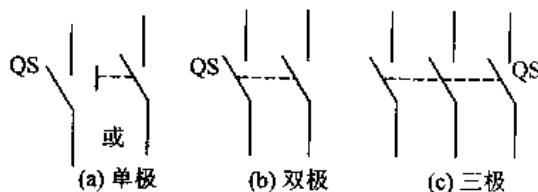


图 1-25 刀开关的图形、文字符号

### 1.4.2 低压断路器

低压断路器也称为自动空气开关,用于分配电能、不频繁地启动异步电动机以及对电源线路及电动机等的保护。当发生严重的过载、短路或欠电压等故障时能自动切断电路。它是低压配电线路应用非常广泛的一种保护电器。

#### 1. 低压断路器的结构及工作原理

低压断路器主要由三个基本部分组成:触头、灭弧系统和各种脱扣器。脱扣器包括过电流脱扣器、失压(欠电压)脱扣器、热脱扣器、分磁脱扣器和自由脱扣器。图 1-26 是低压断路器工作原理示意图。开关是靠操作机构手动或电动合闸的,触头闭合后,自由脱扣器机构将触头锁在合闸位置上。当电路发生故障时,通过各自的脱扣器使自由脱扣机构动作,自动跳闸实现保护作用。

##### (1) 过电流脱扣器

当流过断路器的电流在整定值以内时,过电流脱扣器 3 所产生的吸力不足以吸动衔铁。当电流超过整定值时,强磁场的吸力克服弹簧的拉力拉动衔铁,使自由脱扣机构动作,断路器跳闸,实现过流保护。

##### (2) 失压脱扣器

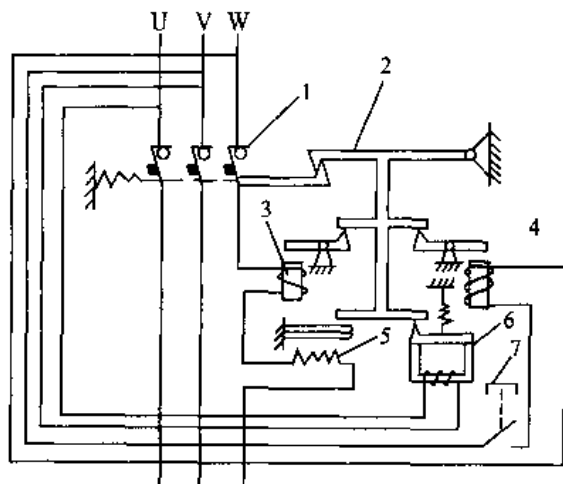


图 1-26 低压断路器的工作原理示意图

1—主触头；2—自由脱扣机构；3—过电流脱扣器；  
4—分磁脱扣器；5—热脱扣器；6—失压脱扣器；7—按钮

失压脱扣器 6 的工作过程与过流脱扣器恰恰相反。当电源电压在额定电压时，失压脱扣器产生的磁力足以将衔铁吸合，使断路器保持在合闸状态。当电源电压下降到低于整定值或降为零时，在弹簧的作用下衔铁释放，自由脱扣机构动作而切断电源。

### （3）热脱扣器

热脱扣器 5 的作用和工作原理与前面介绍的热继电器相同。

### （4）分磁脱扣器

分磁脱扣器 4 用于远距离操作。在正常工作时，其线圈是断电的，在需要远方操作时，使线圈通电，其电磁机构使自由脱扣机构动作，断路器跳闸。

低压断路器的图形、文字符号如图 1-27 所示。

## 2. 低压断路器的主要参数和类型

### （1）低压断路器的主要参数

① 额定电压 是指断路器在长期工作时的允许电压。通常它等于或大于电路的额定电压。

② 额定电流 是指断路器在长期工作时的允许持续电流。

③ 通断能力 是指断路器在规定的电压、频率以及规定的线路参数（交流电路为功率因数，直流电路为时间常数）下，所能接通和分断的短路电流值。

④ 分断时间 是指断路器切断故障电流所需的时间。

### （2）低压断路器的主要类型

① 敞开式低压断路器 它又称万能式低压断路器，具有绝缘衬底的框架结构底座，所有的构件组装在一起，用于配电网的保护。主要型号有 DW10 和 DW15 两个系列。

② 装置式低压断路器 它又称塑料外壳式低压断路器，具有用模压绝缘材料制成的封闭型外壳，将所有构件组装在一起。用做配电网的保护和电动机、照明电路及电热器等控制开

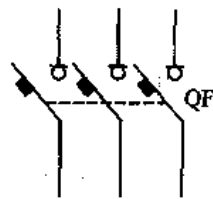


图 1-27 低压断路器的图形和文字符号

关。主要型号有 DZ5、DZ10、DZ20 等系列。

③ 模块化小型断路器 模块化小型断路器由操作机构、热脱扣器、电磁脱扣器、触头系统、灭弧室等部件组成,所有部件都置于一个绝缘壳中。在结构上具有外形尺寸模块化(9 mm 的倍数)和安装导轨化的特点,即单极断路器的模块宽度为 18 mm,凸颈高度为 45 mm,它安装在标准的 35 mm×15 mm 电器安装轨上,利用断路器后面的安装槽及带弹簧的夹紧卡子定位,拆卸方便。该系列断路器可作为线路和交流电动机等的电源控制开关及过载、短路等保护用。广泛应用于工矿企业、建筑及家庭等场所。常用型号有 C45、DZ47、S、DZ187、XA、MC 等系列。表 1-7 列出了 C45 系列断路器的主要技术参数。

表 1-7 C45 系列断路器主要技术参数

产 品 型 号	额定工作 电压/V	脱扣器 额定电 流/A	瞬时脱 扣电流 整定范围	操作频率 /(次/h)	机械寿命 /(万次)	电寿命 /(万次)	质量 /kg				
C45N-1A□P	单极 240 多极 415	1	4~7I <sub>N</sub>	I <sub>N</sub> ≤32 A 时,操作频 率为每小时 240 次, 每次循环中,断路器 应保持在断开位置至 少 13 s;当 I <sub>N</sub> >32 A 时,操作频率为每 小时 240 次,每次循 环中,断路器应保持 在断开位置至少 28 s	2	2	单极 0.127 双极 0.25 三极 0.385 四极 0.495				
C45N-3A□P		3									
C45N-6A□P		6									
C45N-10A□P		10									
C45N-16A□P		16									
C45N-20A□P		20									
C45N-25A□P		25									
C45N-32A□P		32									
C45N-40A□P		40	10~14I <sub>N</sub>								
C45N-50A□P		50									
C45N-63A□P		63									
C45AD-1A□P		1									
C45AD-3A□P		3									
C45AD-6A□P		6									
C45AD-10A□P		10									
C45AD-16A□P		16									
C45AD-20A□P		20									
C45AD-25A□P		25									
C45AD-32A□P		32									
C45AD-40A□P		40									

注:N:照明保护型;AD:电动机保护型。

④ 智能化断路器 传统的断路器的保护功能是利用了热磁效应原理,通过机械系统的动作来实现的。智能化断路器的特征是采用了以微处理器或单片机为核心的智能控制器(智能脱扣器)。它不仅具备普通断路器的各种保护功能,同时还具备实时显示电路中的各种电气参数(电流、电压、功率因数等),对电路进行在线监视、测量、试验、自诊断、通信等功能;能够对各种保护功能的动作参数进行显示、设定和修改。将电路动作时的故障参数存储在非易失存储

器中以便查询。智能化断路器原理框图如图 1-28 所示。

目前国内生产的智能化断路器有框架式和塑料外壳式两种。框架式智能化断路器主要用于智能化自动配电系统中的主断路器。塑料外壳式智能化断路器主要用在配电网中分配电能和作为线路及电源设备的控制与保护,也可用做三相笼型异步电动机的控制。主要型号有 DW45、DW40、DW914(AH)、DW18(AE-S)、DW48、DW19(3WE)、DW17(ME)等。

### 3. 低压断路器的选择

① 额定电流和额定电压应大于或等于线路、设备的正常工作电压和工作电流。

② 热脱扣器的整定电流应与所控制负载(比如电动机)的额定电流一致。

③ 欠电压脱扣器的额定电压等于线路的额定电压。

④ 过电流脱扣器的额定电流大于或等于线路的最大负载电流。对于单台电动机来说,可按下式计算

$$I_Z \geq k I_q \quad (1-8)$$

式中,  $k$  为安全系数,可取 1.5~1.7;  $I_q$  电动机的启动电流。

对于多台电动机来说,可按下式计算

$$I_Z \geq K(I_{q,\max} + I_e) \quad (1-9)$$

式中,  $k$  也可取 1.5~1.7;  $I_{q,\max}$  为最大一台电动机的启动电流;  $I_e$  为其中任意一台电动机的额定电流。

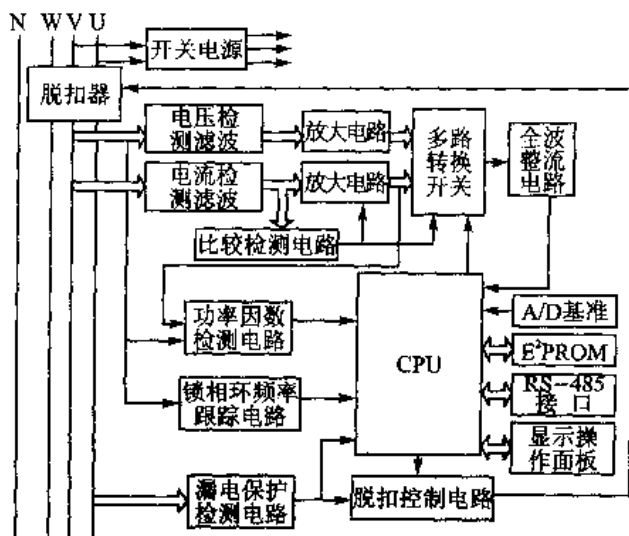


图 1-28 智能化断路器原理框图

## 1.5 熔断器

熔断器基于电流热效应原理和发热元件热熔断原理设计,具有一定的瞬动特性,用于电路的短路保护和严重过载保护。它具有结构简单、体积小、使用维护方便、分断能力较高、限流性能良好等特点,应用十分广泛。

### 1.5.1 熔断器的结构和分类

#### 1. 熔断器的结构

熔断器在结构上主要由熔断管(或盖、座)、熔体及导电部件等部分组成。其中熔体是主要部分,它既是感测元件又是执行元件。熔断管一般由硬质纤维或瓷质绝缘材料制成半封闭式或封闭式管状外壳,熔体则装于其内。熔断管的作用是便于安装熔体和有利于熔体熔断时熄灭电弧。熔体(又称为熔件)是由不同金属材料(铅锡合金、锌、铜或银)制成丝状、带状、片状或

笼状,它串接于被保护电路。熔断器的作用是当电路发生短路时,通过熔体的电流使其发热,当达到熔化温度时熔体自行熔断,从而分断故障电路。

## 2. 熔断器的分类

熔断器的种类很多。按结构来分有半封闭插入式、螺旋式、无填料密封管式和有填料密封管式。按用途来分有一般工业用熔断器、半导体器件保护用快速熔断器和特殊熔断器(如具有两段保护特性的快慢动作熔断器、自复式熔断器等)。

### (1) 插入式熔断器

常用的产品有 RC1A 系列,主要用于低压分支电路的短路保护,由于其分断能力较小,一般多用于民用和照明电路中。

### (2) 螺旋式熔断器

常用的产品有 RL6、RL7、RLS2 等系列。该系列产品的熔管内装有石英砂或惰性气体,用于熄灭电弧,具有较高的分断能力,并带熔断指示器,当熔体熔断时指示器自动弹出。其中 RL6、RL7 多用于机床配电线路中,RLS2 为快速熔断器,主要用于保护电力半导体器件。

### (3) 封闭管式熔断器

该种熔断器分为无填料、有填料和快速三种。RM10 系列为无填料熔断器。在低压电力网络成套配电设备中作短路保护和连续过载保护,其特点是可拆卸,即当熔体熔断后,用户可以按要求自行拆开,重新装入新的熔体。RT12、RT14、RT15 系列为有填料熔断器,技术参数符合国际电工电压熔断器标准,具有较大的分断能力,用于较大电流的电力输配电系统中,还可以用于熔断器式隔离器、开关熔断器等电器中。RS3 系列为快速熔断器,主要用于保护半导体器件。

### (4) 自复式熔断器

自复式熔断器是一种新型熔断器。它利用金属钠作熔体,在常温下,钠的电阻很小,允许通过正常的工作电流。当电路发生短路时,短路电流产生高温使钠迅速气化,气态钠电阻变得很高,从而限制了短路电流。当故障消除后,温度下降,金属钠重新固化,恢复其良好的导电性,其优点是能重复使用,不必更换熔体,但在线路中只能限制故障电流,而不能切断故障电路,一般与断路器配合使用。常用产品有 RZ1 系列。

## 1.5.2 熔断器的保护特性

熔断器的保护特性亦称熔化特性(或称安秒特性),是指熔体的熔化电流与熔化时间之间的关系。它和热继电器的保护特性一样,也具有反时限特性,如图 1-29 所示。在保护特性中有一熔断电流与不熔断电流的分界线,与此相应的电流就是最小熔化电流  $I_r$ 。当熔体通过电流等于或大于  $I_r$  时,熔体熔断。当熔体通过电流小于  $I_r$  时,熔体不熔断。根据对熔断器的要求,熔体在额定电流  $I_n$  时绝对不应熔断。

最小熔化电流  $I_r$  与熔体额定电流  $I_n$  之比称为熔断器的熔化系数,即  $K_r = I_r / I_n$ 。

熔化系数主要取决于熔体的材料和工作温度以及它的结构。当熔体采用低熔点的金属材料(如铅、

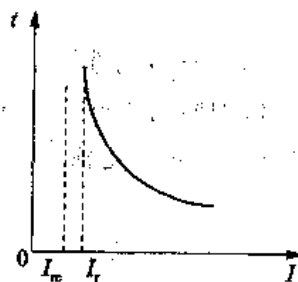


图 1-29 熔断器的保护特性

锡合金及锌等)时,熔化时所需热量小,故熔化系数较小,有利于过载保护。但它们的电阻系数较大,熔体截面积较大,熔断时产生的金属蒸气较多,不利于灭弧,故分断能力较低。当熔体采用高熔点的金属材料(如铝、铜和银等)时,熔化时所需热量大,故熔化系数大,不利于过载保护,而且可能使熔断器过热;然而,它们的电阻系数低,熔体截面积较小,有利于灭弧,故分断能力较高。由此看来,不同熔体材料的熔断器,在电路中起保护作用的侧重点是不同的。

### 1.5.3 熔断器的技术参数

#### (1) 额定电压

它指熔断器长期工作时和分断后能够承受的电压,其值一般等于或大于电气设备的额定电压。

#### (2) 额定电流

它指熔断器长期工作时,温升不超过规定值时所能承受的电流。为了减少熔断管的规格,熔断管的额定电流等级比较少,而熔体的额定电流等级比较多,即在一个额定电流等级的熔断管内可以分几个额定电流等级的熔体,但熔体的额定电流最大不能超过熔断管的额定电流。

#### (3) 极限分断能力

熔断器在规定的额定电压和功率因数(或时间常数)的条件下,能分断的最大电流值,而在电路中出现最大电流值一般是指短路电流值。所以,极限分断能力也反映了熔断器分断短路电流的能力。

熔断器的图形和文字符号如图 1-30 所示。



图 1-30 熔断器的图形和文字符号

### 1.5.4 熔断器的选择

熔断器的选择包括熔断器类型的选择和熔体额定电流的选择两大部分。

#### 1. 熔断器类型的选择

选择熔断器的类型时,主要依据负载的保护特性和短路电流的大小。例如,用于保护照明和电动机的熔断器,一般是考虑它们的过载保护,这时,希望熔断器的熔化系数适当小些。所以容量较小的照明线路和电动机宜采用熔体为铅锌合金的 RC1A 系列熔断器,而大容量的照明线路和电动机,除过载保护外,还应考虑短路时分断短路电流的能力。若短路电流较小时,可采用熔体为锡质的 RC1A 系列或熔体为锌质的 RM10 系列熔断器。用于车间低压供电线路的保护熔断器,一般是考虑短路时的分断能力。当短路电流较大时,宜采用具有高分断能力的 RLI 系列熔断器。当短路电流相当大时,宜采用有限流作用的 RT0 及 RT12 系列熔断器。

#### 2. 熔体额定电流的选择

① 用于保护照明或电热设备的熔断器,因负载电流比较稳定,熔体的额定电流一般应等于或稍大于负载的额定电流,即

$$I_{re} \geq I_e \quad (1-10)$$

式中:  $I_{re}$  为熔体的额定电流,  $I_e$  为负载的额定电流。

② 用于保护单台长期工作电动机(即供电支线)的熔断器,考虑电动机启动时不应熔断,即

$$I_{re} \geq (1.5 \sim 2.5) I_e \quad (1-11)$$

轻载启动或启动时间比较短时,系数可取近似 1.5。带重载启动或启动时间比较长时,系数可取近似 2.5。

③ 用于保护频繁启动电动机(即供电支线)的熔断器,考虑频繁启动时发热而熔断器也不应熔断,即

$$I_{re} \geq (3 \sim 3.5) I_e \quad (1-12)$$

式中:  $I_{re}$  为熔体的额定电流,  $I_e$  为电动机的额定电流。

④ 用于保护多台电动机(即供电干线)的熔断器,在出现尖峰电流时不应熔断。通常将其中容量最大的一台电动机启动,而其余电动机正常运行时出现的电流作为其尖峰电流。为此,熔体的额定电流应满足下述关系

$$I_{re} \geq (1.5 \sim 2.5) I_{e, \max} + \sum I_e \quad (1-13)$$

式中:  $I_{e, \max}$  为多台电动机中容量最大的一台电动机额定电流,  $\sum I_e$  为其余电动机额定电流之和。

⑤ 为防止发生越级熔断,上、下级(即供电干、支线)熔断器间应有良好的协调配合,为此,应使上一级(供电干线)熔断器的熔体额定电流比下一级(供电支线)大 1~2 个级差。

⑥ 熔断器额定电压的选择:熔断器的额定电压应等于或大于所在电路的额定电压。

## 1.6 主令电器

主令电器是自动控制系统中用于发送和转换控制命令的电器。主令电器用于控制电路,不能直接分合主电路。主令电器应用十分广泛,种类很多,本节介绍几种常用的主令电器。

### 1.6.1 控制按钮

控制按钮简称按钮,是一种结构简单使用广泛的手动电器,在控制电路中用于手动发出控制信号以控制接触器、继电器等。

控制按钮一般由按钮帽、复位弹簧、触点和外壳等部分组成,其结构如图 1-31 所示,图形和文字符号如图 1-32 所示。按钮中触点的形式和数量根据需要可以装配成 1 常开 1 常闭到 6 常开 6 常闭的形式。接线时,也可以只接常开或常闭触点。当按下按钮时,先断开常闭触点,而后接通常开触点。按钮释放后,在复位弹簧作用下使触点复位。

控制按钮可做成单式(一个按钮)、复式(两个按钮)和三联式(有三个按钮)的形式。为便于识别各个按钮的作用,避免误操作,通常将按钮帽做成不同颜色,以示区别,其颜色有红、绿、黄、蓝、白等。如红色表示停止按钮,绿色表示启动按钮等,如表 1-8 所列。另外还有形象化符号可供选用,如图 1-33 示。

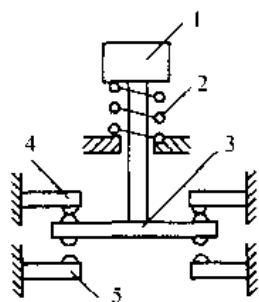


图 1-31 控制按钮结构示意图

1—按钮帽;2—复位弹簧;3—动触点;  
4—常闭触点;5—常开触点

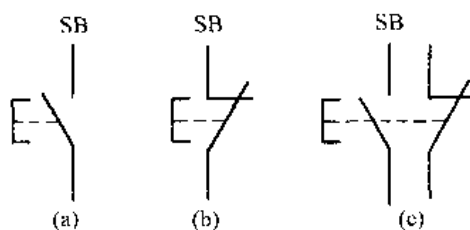


图 1-32 控制按钮的图形及文字符号

表 1-8 控制按钮颜色及其含义

颜色	含义	典型应用
红色	危险情况下的操作	紧急停止
	停止或分断	停止一台或多台电动机,停止一台机器的一部分,使电器元件失电
黄色	应急或干预	抑制不正常情况或中断不理想的工作周期
绿色	启动或接通	启动一台或多台电动机,启动一台机器的一部分,使电器元件得电
蓝色	上述几种颜色未包括的任一种功能	—
黑色、灰色、白色	无专门指定功能	可用于停止和分断上述以外的任何情况

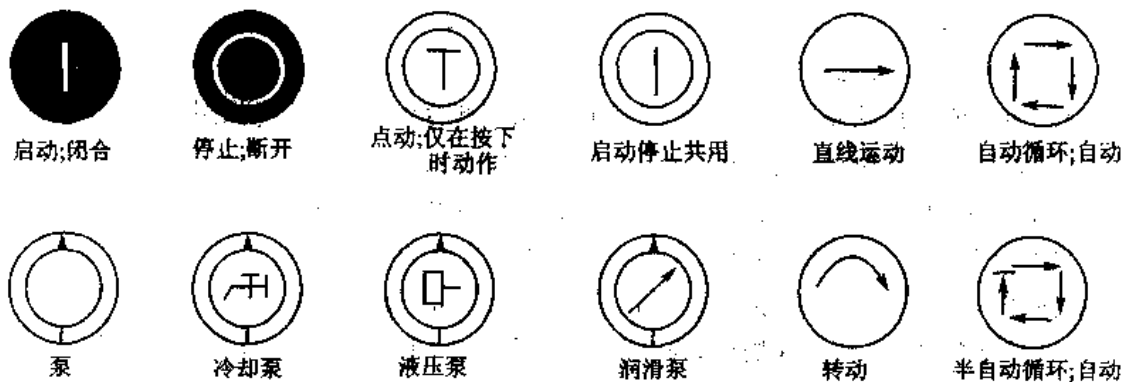


图 1-33 控制按钮的形象化符号

控制按钮在结构上有按钮式、自锁式、紧急式、钥匙式、旋钮式和保护式等,有些按钮还带有指示灯,可根据使用场合和具体用途来选用。

控制按钮的主要参数有外观型式及安装孔尺寸、触头数量及触头的电流容量,在产品说明书中都有详细说明。常用国产产品有 LAY3、LAY6、LA20、LA25、LA101、LA38、NP1 等系列。国外进口及引进产品品种亦很多,几乎所有大的国外低压电器厂商都有产品进入我国市场,并有一些结构新颖的新品种。

### 1.6.2 转换开关

转换开关是一种多档式、控制多回路的主令电器。广泛应用于各种配电装置的电源隔离、电路转换、电动机远距离控制等,也常作为电压表、电流表的换相开关,还可用于控制小容量的电动机。

目前常用的转换开关主要有两大类,即万能转换开关和组合开关。两者的结构和工作原理基本相似,在某些应用场合可以相互替代。转换开关按结构可分为普通型、开启型和防护组合型等。按用途又分为主令控制和控制电动机两种。

转换开关一般采用组合式设计,由操作结构、定位系统、限位系统、接触系统、面板及手柄等组成。接触系统采用双断点桥式结构,并由各自的凸轮控制其通断;定位系统采用棘轮棘爪式结构,不同的棘轮和凸轮可组成不同的定位模式,从而得到不同的开关状态,即手柄在不同的转换角度时,触头的状态是不同的。

转换开关是由多组相同结构的触点组件叠装而成,图 1-34 为 LW12 系列转换开关某一层的结构原理。LW12 系列转换开关由操作结构、面板、手柄和数个触头等主要部件组成,用螺栓组成为一个整体。触头底座由 1~12 层组成,其中每层底座最多可装 4 对触头,并由底座中间的凸轮进行控制。由于每层凸轮可做成不同的形状,因此,当手柄转到不同位置时,通过凸轮的作用,可使各对触头按所需要的规律接通和分断。

转换开关手柄的操作位置是以角度来表示的,不同型号的转换开关,其手柄有不同的操作位置。这可从电气设备手册中万能转换开关的“定位特征表”中查找到。

转换开关的触点在电路图中的图形符号如图 1-35 所示。由于其触点的分合状态是与操作手柄的位置有关,因此,在电路图中除画出触点圆形符号之外,还应有操作手柄位置与触点分合状态的表示方法。其表示方法有两种,一种是在电路图中画虚线和画“·”的方法,如图 1-35(a)所示,即用虚线表示操作手柄的位置,用有无“·”表示触点的闭合和断开状态。比如,在触点图形符号下方的虚线位置上画“·”,则表示当操作手柄处于该位置时,该触点是处于闭合状态;若在虚线位置上未画“·”,则表示该触点是处于断开状态。另一种方法是,在电路图中既不画虚线也不画“·”,而是在触点图形符号上标出触点编号,再用接通表表示操作手柄于不同位置时的触点分合状态,如图 1-35(b)所示。在接通表中用有无“×”来表示操作手柄不同位置时触点的闭合和断开状态。

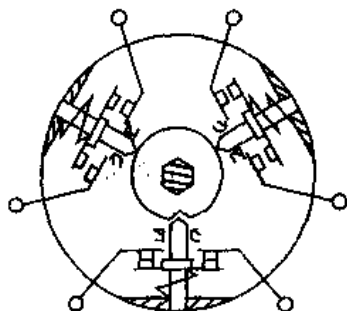
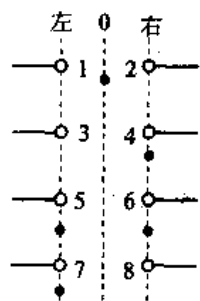


图 1-34 LW12 系列转换开关一层结构示意图



(a) 画“·”标记表示

触点	位置	
	左	右
1-2	×	
3-4		×
5-6	×	×
7-8	×	

(b) 接通表表示

图 1-35 转换开关的图形符号

转换开关的主要参数有型式、手柄类型、操作图型式、工作电压、触头数量及其电流容量等,在产品说明书中都有详细说明。常用的转换开关有 LW5、LW6、LW8、LW9、LW12、LW16、VK、HZ 等系列,另外还有许多品牌的进口产品也在国内得到广泛应用。

### 1.6.3 行程开关

行程开关又称为限位开关,是一种利用生产机械某些运动部件的碰撞来发出控制命令的主令电器。用于控制生产机械的运动方向、速度、行程大小或位置的一种自动控制器件。

行程开关广泛应用于各类机床、起重机械以及轻工机械的行程控制。当生产机械运动到某一预定位置时,行程开关通过机械可动部分的动作,将机械信号转换为电信号,以实现对其生产机械的控制,限制它们的动作和位置,借此对生产机械给以必要的保护。

行程开关按其结构可分为直动式(如 LX1、JLXK1 系列)、滚轮式(如 LX2、JLXK2 系列)和微动式(如 LXW-11、JLXK1-11 系列)三种。

直动式行程开关的动作原理与按钮相同。但它的缺点是分合速度取决于生产机械的移动速度,当移动速度低于  $0.4 \text{ m/min}$  时,触点分断太慢,易受电弧烧损。此时,应采用有盘形弹簧机构瞬时动作的滚轮式行程开关。当生产机械的行程比较小且作用力也很小时,可采用具有瞬时动作和微小行程的微动式行程开关。行程开关的图形和文字符号如图 1-36 所示。

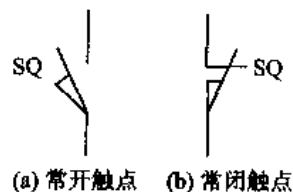


图 1-36 行程开关的图形和文字符号

行程开关的主要参数有动作行程、工作电压及触头的电流容量等,在产品说明书中都有详细说明。目前国内生产的行程开关有 LK3、3SE3、LS19、LSW、WL、LX、JLSK 等系列。其中 3SE3 系列为引进西门子公司技术生产。另外还有大量的国外进口及港台地区的产品,也得到了广泛的应用。

### 1.6.4 接近开关

随着电子技术的发展,出现了非接触式的行程开关,即接近开关。接近开关又称为无触点行程开关。当某种物体与之接近到一定距离时就发出动作信号,它不像机械行程开关那样需要施加机械力,而是通过其感辨头与被测物体间介质能量的变化来获取信号。接近开关的应用已远超出一般行程控制和限位保护的范畴,例如用于高速计数、测速、液面控制、检测金属体的存在、零件尺寸以及无触点按钮等。既使用于一般行程控制,其定位精度、操作频率、使用寿命和对恶劣环境的适应能力也优于一般机械式行程开关。

接近开关按工作原理可以分为高频振荡型、电容型、霍尔型等几种类型。

高频振荡型接近开关是以金属触发为原理,主要由高频振荡器、集成电路或晶体管放大电路和输出电路三部分组成。其基本工作原理是,振荡器的线圈在开关的作用表面产生一个交变磁场,当金属检测体接近此作用表面时,在金属检测体中将产生涡流,由于涡流的去磁作用使感辨头的等效参数发生变化,由此改变振荡回路的谐振阻抗和谐振频率,使振荡停止。振荡器的振荡和停振这两个信号,经整形放大后转换成开关信号输出。

电容型接近开关主要由电容式振荡器及电子电路组成。它的电容位于传感器表面,当物体接近时,因改变了其耦合电容值,从而产生振荡和停振使输出信号发生跳变。

霍尔型接近开关由霍尔元件组成,是将磁信号转换为电信号输出,内部的磁敏元件仅对垂直于传感器端面磁场敏感,当磁极 S 正对接近开关时,接近开关的输出产生正跳变,输出为高电平。若磁极 N 正对接近开关,输出产生正跳变,输出为低电平。

接近开关的图形和文字符号如图 1-37 所示。

接近开关的工作电压有交流和直流两种,输出形式有两线、三线 and 四线三种;有一对常开、常闭触点,晶体管输出类型有 NPN、PNP 两种;外形有方型、圆型、槽型和分离型等多种。接近开关的主要参数有动作行程、工作电压、动作频率、响应时间、输出形式以及触点电流容量等,在产品说明书中有详细说明。接近开关的产品种类十分丰富,常用的国产接近开关有 3SG、1J、CJ、SJ、AB 和 LXJ0 等系列,另外,国外进口及引进产品应用也非常广泛。

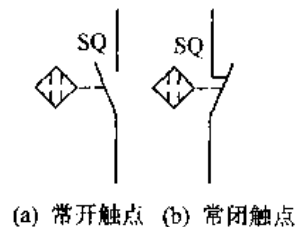


图 1-37 接近开关的图形和文字符号

### 1.6.5 光电开关

光电开关是接近开关的一种,它除克服了接触式行程开关存在的诸多不足外,还克服了接近开关的作用距离短、不能直接检测非金属材料等缺点。它具有体积小、功能多、寿命长、精度高、响应速度快、检测距离远以及抗电磁干扰能力强等优点。还可非接触、无损伤地检测和控制各种固体、液体、透明体、黑体、柔软体和烟雾等物质的状态和动作。目前,光电开关已被用做物位检测、液位控制、产品计数、宽度判别、速度检测、定长剪切、孔洞识别、信号延时、自动门传感、色标检出以及安全防护等诸多领域。

光电开关按检测方式可分为反射式、对射式和镜面反射式三种类型。表 1-9 给出了光电开关的检测分类方式及特点说明。

表 1-9 光电开关的检测分类方式及特点

检测方式	光 路	特 点
对射式	扩散	检测距离远,也可检测半透明物体的密度(透过率)
	狭角	光束发散角小,抗邻组干扰能力强
	细束	擅长检出细微的孔径、线型和条状物
	槽型	光轴固定不需调节,工作位置精度高
	光纤	适宜空间狭小、电磁干扰大、温差大、需防爆的危险环境

续表 1-9

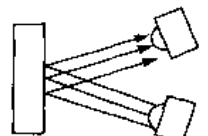
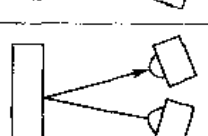
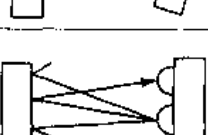
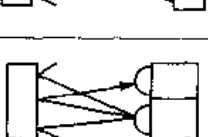
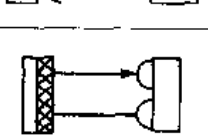
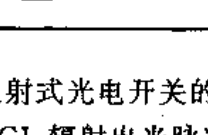
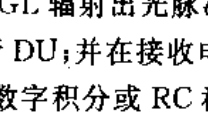
检测方式	光 路	特 点
限 距		工作距离限定在光束交点附近,可避免背景影响
狭 角		无限距型,可检测透明物后面的物体
反 射 式		检测透明体和不透明体
扩 散		颜色标记和孔隙、液滴、气泡检出,测电表、水表转速
光 纤		检测距离远,可检出所有物体,通用性强
镜面反射式		适宜空间狭小、电磁干扰大、温差大、需防爆的危险环境
镜面反射式		反射距离远,适宜远距检出,还可检出透明、半透明物体

图 1-38 所示是反射式光电开关的工作原理框图。图中,由振荡回路产生的调制脉冲经反射电路后,由发光管 GL 辐射出光脉冲。当被测物体进入受光器作用范围时,被反射回来的光脉冲进入光敏三极管 DU;并在接收电路中将光脉冲解调为电脉冲信号,再经放大器放大和同步选通整形,然后用数字积分或 RC 积分方式排除干扰,最后经延时(或不延时)触发驱动输出光电开关控制信号。

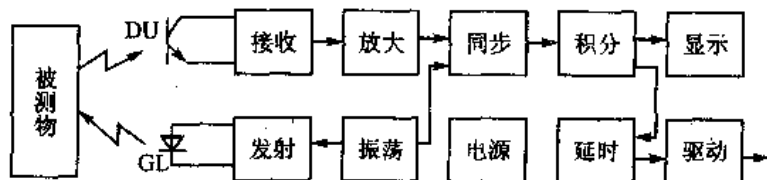


图 1-38 反射式光电开关的工作原理框图

光电开关一般都具有良好的回差特性,即使被检测物在小范围内晃动也不会影响驱动器的输出状态,从而可使其保持在稳定工作区。同时,自诊断系统还可以显示受光状态和稳定工作区,以随时监视光电开关的工作。

光电开关外形有方型、圆型等几种,主要参数有动作行程、工作电压、输出形式等,在产品说明书中有详细说明。光电开关的产品种类十分丰富,应用也非常广泛。

### 1.6.6 指示灯

指示灯在各类电器设备及电气线路中做电源指示及指挥信号、预告信号、运行信号、故障信号及其他信号的指示。

指示灯主要有壳体、发光体、灯罩等组成。外形结构多种多样,发光体主要有白炽灯、氖灯

和半导体型三种。发光颜色有黄、绿、红、白、蓝等五种,使用时按国标规定的用途选用,见表1-10。指示灯的图形和文字符号如图1-39所示。主要参数有安装孔尺寸、工作电压及颜色等。常用国产系列有AD11、AD30、XDJ1等。另外国外进口和合资生产的品种也很多,几乎生产低压电器的公司均有指示灯产品,如德国西门子、法国施耐德、日本和泉、日本富士等。

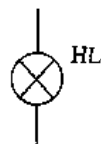


图1-39 指示灯的图形和文字符号

表1-10 指示灯的颜色及其含义

颜色	含义	解释	典型应用
红色	异常或警报	对可能出现危险和需要立即处理的情况进行报警	温度超过规定限制,设备的重要部分已被保护电器切断
黄色	警告	状态改变或变量接近其极限值	温度偏离正常值
绿色	准备、安全	安全运行条件指示或机械准备启动	设备正常运转
蓝色	特殊指示	上述几种颜色未包括的任意一种功能	—
白色	一般信号	上述几种颜色未包括的各种功能	—

## 本章小结

(1) 低压电器的种类繁多,本章主要介绍了接触器、继电器、开关电器、熔断器、主令电器等常用低压电器的用途、基本结构、工作原理及其主要技术参数和图形符号,为正确使用它们奠定了基础。

(2) 每一种电器都有它一定的使用范围,要根据使用的具体条件正确选用。在选用电器时,其技术参数是最主要的依据,其详细内容可参阅电器产品的技术手册及产品说明。

(3) 保护电器(如热继电器、熔断器、断路器等)及某些控制电器(如时间继电器、温度继电器、液位继电器等)的使用,除了要根据保护要求、控制要求正确选用电器的类型外,还要根据被保护、被控制电路的具体条件,进行必要的调整整定动作值,同时还要考虑各保护电器之间的配合特性的要求。

(4) 随着电器技术的发展,各种新型及引进电器不断出现。为优化系统,提高系统可靠性应尽量选用新型电器元件。

## 思考题与练习题

1. 何谓电磁式电器的吸力特性与反力特性? 吸力特性与反力特性之间应满足怎样的配合关系?
2. 单相交流电磁铁的短路环断裂或脱落后,在工作中会出现什么现象? 为什么?
3. 常用的灭弧方法有哪些?
4. 接触器的作用是什么? 根据结构特征如何区分交、直流接触器?

5. 交流接触器在衔铁吸合前的瞬间,为什么会在线圈中产生很大的电流冲击?直流接触器会不会出现这种现象?为什么?
6. 交流电磁线圈误接入直流电源,直流电磁线圈误接入交流电源,会发生什么问题?为什么?
7. 熔断器的额定电流、熔体的额定电流和熔体的极限分断电流三者有何区别?
8. 控制按钮、转换开关、行程开关、接近开关、光电开关在电路中各起什么作用?
9. 说明热继电器和熔断器保护功能的不同之处。
10. 热继电器在电路中的作用是什么?带断相保护和不带断相保护的三相式热继电器各用在什么场合?
11. 当出现通风不良或环境温度过高而使电动机过热时,能否采用热继电器进行保护?为什么?
12. 中间继电器与接触器有何异同?
13. 温度继电器为什么能实现全热保护?
14. 感应式速度继电器时怎样实现动作的?用于什么场合?
15. 简述固态继电器优缺点及使用时的注意事项。

## 第二章 电气控制线路基础

在各行各业广泛使用的电气设备和生产机械中,其自动控制线路大多以各类电动机或其他执行电器为被控对象,以继电器、接触器、按钮、行程开关、保护元件等器件组成的自动控制线路,通常称为电气控制线路。电气控制线路是根据一定的控制方式用导线把它们连接起来组成的。其作用是对被控对象实现自动控制,以满足生产工艺的要求和实现生产过程自动化。

生产工艺和生产过程不同,对控制线路的要求也不同。但是,无论哪一种控制线路,都是由一些比较基本的控制环节组合而成的。因此,只要掌握控制线路的基本环节以及一些典型线路的工作原理、分析方法和设计方法,就很容易掌握复杂电气控制线路的分析方法和设计方法。结合其具体的生产工艺要求,通过基本环节的组合,可设计出复杂的电气控制线路。

本章主要介绍应用广泛的三相笼型异步电动机的启动、调速、制动的基本控制线路和一些典型控制线路,其中重点讲解新型的电气控制装置软启动器和变频器的使用,最后讲解电气控制线路的简单设计法和电气控制线路分析基础。

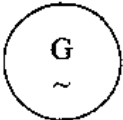



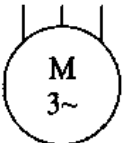
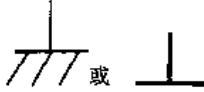
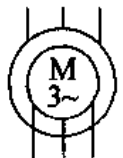


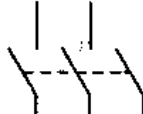



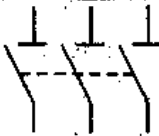
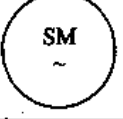


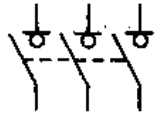
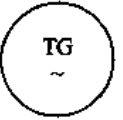


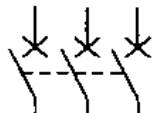
### 2.1 电气控制线路图的图形、文字符号及绘制原则

电气控制线路是由许多电器元件按一定的要求连接而成的。为了表达生产机械电气控制系统的结构、原理等设计意图,便于电气系统的安装、调试、使用和维修,将电气控制系统中各电器元件及其连接线路用一定的图形表达出来,这就是电气控制系统图。电气控制系统图一般有三种:电气原理图、电器布置图和电气安装接线图。在图上用不同的图形符号来表示各种电器元件,用不同的文字符号来说明图形符号所代表的电器元件的基本名称、用途、主要特征及编号等。各种图有其不同的用途和规定画法,应根据简明易懂的原则,采用统一规定的图形符号、文字符号和标准画法来绘制。




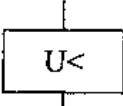

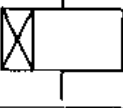

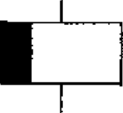
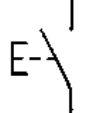
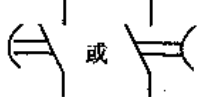
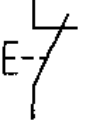
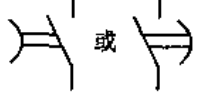
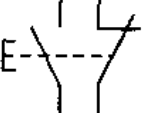
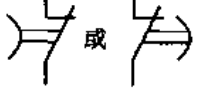

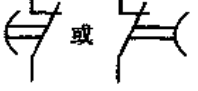

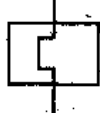
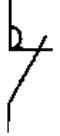
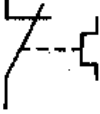
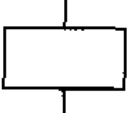

#### 2.1.1 常用电气图形符号和文字符号

电气控制系统图、电气元件的图形符号和文字符号必须符合国家标准规定。国家标准局参照国际电工委员会(IEC)颁布的标准,制定了我国电气设备有关国家标准:GB 4728—1984《电气图用图形符号》及 GB 6988—1987《电气制图》和 GB 7159—1987《电气技术中的文字符号制订通则》。规定从 1990 年 1 月 1 日起,电气控制线路中的图形和文字符号必须符合最新的国家标准。一些常用电气图形符号和文字符号如表 2-1 所列。


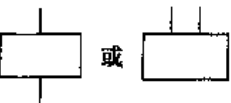

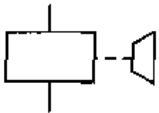
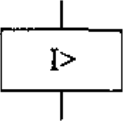
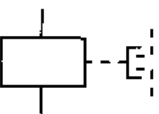




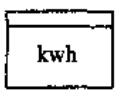

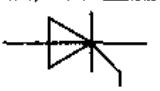



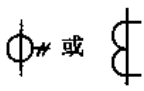

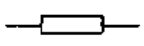


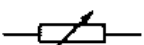
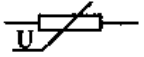
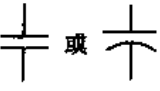
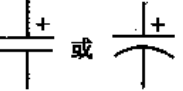
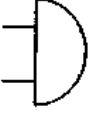

表 2-1 电气控制线路中常用图形符号和文字符号

名 称	图形符号 (GB 4728-1984)	文字符号 (GB 7159-1987)	名 称	图形符号 (GB 4728-1984)	文字符号 (GB 7159-1987)
交流发电机		GA	接地的一般符号		E
交流电动机		MA	保护接地		PE
三相笼型异步电动机		MC	接机壳或接地板		PU
三相绕线型异步电动机		MW	单极控制开关		SA
直流发电机		GD	三极控制开关		SA
直流电动机		MD	隔离开关		QS
直流伺服电动机		SM	三极隔离开关		QS
交流伺服电动机		SM	负荷开关		QS
直流调速发动机		TG	三极负荷开关		QS
交流调速发动机		TG	断路器		QF
步进电动机		M	三极断路器		QF

续表 2-1

名 称	图形符号 (GB 4728—1984)	文字符号 (GB 7159—1987)	名 称	图形符号 (GB 4728—1984)	文字符号 (GB 7159—1987)
双绕组 变压器		T	电压互感器		TV
位置开关 常开触点		SQ	欠压继电器 线圈		KV
位置开关 常闭触点		SQ	通电延时 (缓吸)线圈		KT
作双向机械 操作的位置 开关		SQ	继电延时 (缓放)线圈		KT
常开按钮		SB	延时闭合 常开触点		KT
常闭按钮		SB	延时断开 常开触点		KT
复合按钮		SB	延时闭合 常闭触点		KT
交流接触器 线圈		KM	延时断开 常闭触点		KT
接触器 常开触点		KM	热继电器 热元件		FR
接触器 常闭触点		KM	热继电器 常闭触点		FR
中间继电器 线圈		KA	熔断器		FU

续表 2-1

名 称	图形符号 (GB 4728—1984)	文字符号 (GB 7159—1987)	名 称	图形符号 (GB 4728—1984)	文字符号 (GB 7159—1987)
中间继电器 常开触点		KA	电磁铁		YA
中间继电器 常闭触点		KA	电磁制动器		YB
过流继电器 线圈		KA	电磁离合器		YC
电流表		PA	照明灯		EL
			信号灯		HL
电压表		PV	二极管		V
电度表		PJ	NPN 晶体管		V
晶闸管		V	PNP 晶体管		V
可拆卸端子		X	端 子		X
电流互感器		TA	控制电路用 电源整流器		VC
电阻器		R	电抗器	 或 	L
电位器		RP			
压敏电阻		RV			
电容器 一般符号		C	极性电容器		C
电铃		B	蜂鸣器		B

### 2.1.2 电气控制线路图的绘制原则

电气控制系统图一般有三种:电气原理图、电器布置图和电气安装接线图。由于它们的用途不同,绘制原则也有差别。根据学生学习的实际需要,这里重点介绍电气原理图。

电气原理图的目的是便于阅读和分析控制线路,应根据结构简单、层次分明清晰的原则,采用电器元件展开形式绘制。它包括所有电器元件的导电部件和接线端子,但并不按照电器元件的实际布置位置来绘制,也不反映电器元件的实际大小。

下面以图 2-1 所示的某机床的电气原理图为例,来说明电气原理图的规定画法和应注意的事项。

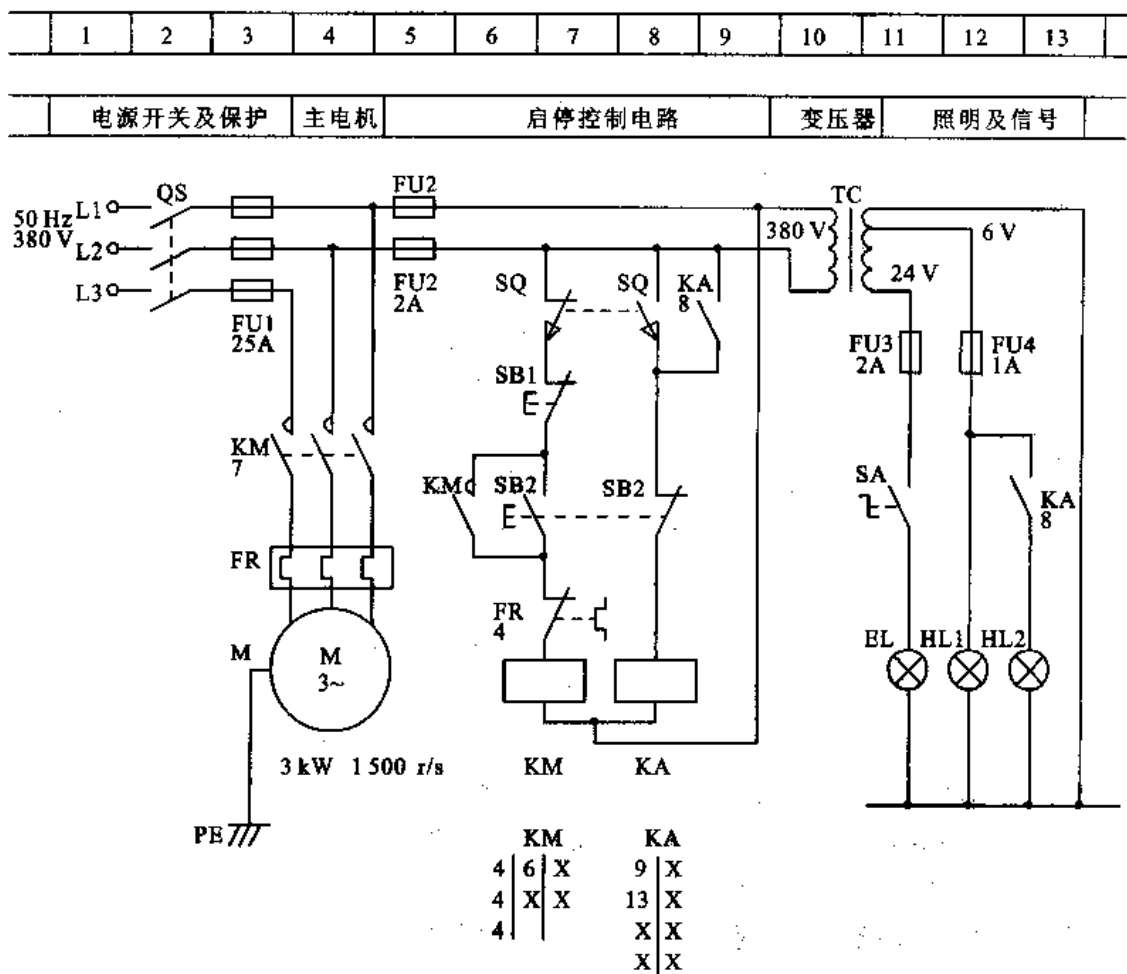


图 2-1 某机床电气原理图

#### 1. 绘制电气原理图时应遵循的原则

(1) 电气原理图一般分主电路和辅助电路两部分。主电路是电气控制线路中大电流通过的部分,包括从电源到电机之间相连的电器元件,一般由组合开关、主熔断器、接触器主触点、热继电器的热元件和电动机等组成。辅助电路是控制线路中除主电路以外的电路,其流过的电流比较小。辅助电路包括控制电路、照明电路、信号电路和保护电路。其中控制电路是由按钮、接触器和继电器的线圈及辅助触点、热继电器触点、保护电器触点等组成。

(2) 电气原理图中所有电器元件都应采用国家标准中统一规定的图形符号和文字符号

表示。

(3) 电气原理图中电器元件的布局,应根据便于阅读的原则安排。主电路安排在图面左侧或上方,辅助电路安排在图面右侧或下方。无论主电路还是辅助电路,均按功能布置,尽可能按动作顺序从上到下、从左到右排列。

(4) 电气原理图中,当同一电器元件的不同部件(如线圈、触点)分散在不同位置时,为了表示是同一元件,要在电器元件的不同部件处标注统一的文字符号。对于同类器件,要在其文字符号后加数字序号来区别。如两个接触器,可用 KM1、KM2 文字符号区别。

(5) 电气原理图中,所有电器的可动部分均按没有通电或没有外力作用时的状态画出。对于继电器、接触器的触点,按其线圈不通电时的状态画出;控制器按手柄处于零位时的状态画出;对于按钮、行程开关等触点,按未受外力作用时的状态画出。

(6) 电气原理图中,应尽量减少线条和避免线条交叉。各导线之间有电联系时,在导线交点处画实心圆点。根据图面布置需要,可以将图形符号旋转绘制,一般逆时针方向旋转  $90^\circ$ ,但文字符号不可倒置。

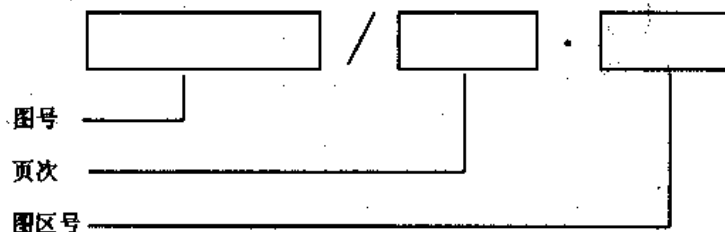
## 2. 图面区域的划分

图纸上方的 1、2、3... 等数字是图区的编号,它是为了便于检索电气线路,方便阅读分析从而避免遗漏设置的。图区编号也可设置在图的下方。

图区编号下方的文字表明它对应的下方元件或电路的功能,使读者能清楚地知道某个元件或某部分电路的功能,以利于理解全部电路的工作原理。

## 3. 符号位置的索引

符号位置的索引用图号、页次和图区编号的组合索引法,索引代号的组成如下:



图号是指当某设备的电气原理图按功能多册装订时,每册的编号,一般用数字表示。

当某一元件相关的各符号元素出现在不同图号的图纸上,而当每个图号仅有一页图纸时,索引代号中可省略“页号”及分隔符“·”。

当某一元件相关的各符号元素出现在同一图号的图纸上,而该图号有几张图纸时,可省略“图号”和分隔符“/”。

当某一元件相关的各符号元素出现在只有一张图纸的不同图区时,索引代号只用“图区”表示。

如图 2-1 图区 9 中的 KA 常开触点下面的“8”即为最简单的索引代号。它指出了继电器 KA 的线圈位置在图区 8。

图 2-1 中接触器 KM 线圈及继电器 KA 线圈下方的文字是接触器 KM 和继电器 KA 相应触点的索引。电气原理图中,接触器和继电器线圈与触点的从属关系使用右图表示。即在原理图中相应线圈下方,给出触点的图形符号,并在下面标明相应触点的索引代码,且对未使用的触点用“×”表明,有时也可采用省略的表示方法。

KM			KA		
4	6	×	9	×	
4	×	×	13	×	
4			×	×	
			×	×	

对接触器,上述表示法中各栏的含义如下所示:

左 栏	中 栏	右 栏
主触点所在的图区号	辅助常开触点所在的图区号	辅助常闭触点所在的图区号

对继电器,上述表示法中各栏的含义如下所示:

左 栏	右 栏
辅助常开触点所在的图区号	辅助常闭触点所在的图区号

## 2.2 三相笼型异步电动机的基本控制线路

三相笼型异步电动机由于结构简单、价格便宜、坚固耐用等优点获得了广泛的应用。在生产实际中,它的应用占到了使用电动机的 80 % 以上。所以本章主要讲解三相笼型异步电动机的控制线路。三相笼型异步电动机的控制线路大都由继电器、接触器和按钮等有触点的电器组成。本节介绍其基本的控制线路。

### 2.2.1 全压启动控制线路

图 2-2 所示为三相笼型异步电动机单向全压启动控制线路。主电路由刀开关 QS、熔断器 FU1、接触器 KM 的主触点、热继电器 FR 的热元件和电动机 M 构成。控制线路由热继电器 FR 的常闭触点、停止按钮 SB1、启动按钮 SB2、接触器 KM 常开触点以及它的线圈组成。这是最基本的电动机控制线路。

#### 1. 控制线路工作原理

启动时,合上刀开关 QS,主电路引入三相电源。按下启动按钮 SB2,接触器 KM 线圈通电,其常开主触点闭合,电动机接通电源开始全压启动,同时接触器 KM 的辅助常开触点闭合,使接触器 KM 线圈有两条通电路径。这样当松开启动按钮 SB2 后,接触器 KM 线圈仍能通过其辅助触点通电并保持吸合状态。这种依靠接触器本身辅助触点使其线圈保持通电的现象称为自锁。起自锁作用的触点称为自锁触点。

要使电动机停止运转,按停止按钮 SB1,接触器 KM 线圈失电,则其主触点断开。切断电动机三相电源,电动机 M 自动停车,同时接触器 KM 自锁触点也断开,控制回路解除自锁。松开停止按钮 SB1,控制电路又回到启动前的状态。

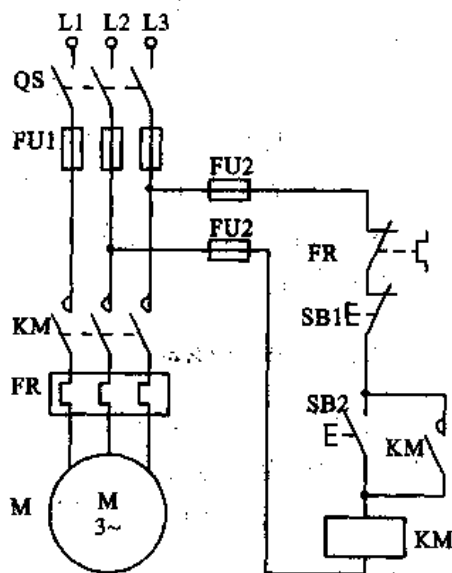


图 2-2 单向全压启动控制线路

## 2. 控制线路的保护环节

### (1) 短路保护

当控制线路发生短路故障时,控制线路应能迅速切除电源,熔断器 FU1 是作为主电路短路保护用的,但它达不到过载保护的目的。这是因熔断器的规格是根据电动机的启动电流大小作适当选择的;另一方面熔断器的保护特性分散性很大,即使是同一种规格的熔断器,其特性曲线也往往很不相同。熔断器 FU2 为控制线路的短路保护。

### (2) 过载保护

电动机长期超载运行,会造成电动机绕组温升超过其允许值而损坏,通常要采取过载保护。过载保护的特点是:负载电流越大,保护动作时间越快,但不能受电动机启动电流影响而动作。

过载保护由热继电器 FR 完成。一般来说,热继电器发热元件的额定电流按电动机额定电流来选取。由于热继电器热惯性很大,即使热元件流过几倍的额定电流,热继电器也不会立即动作,因此在电动机启动时间不长的情况下,热继电器是不会动作的。只有过载时间比较长时,热继电器动作,常闭触点 FR 断开,接触器 KM 线圈失电跳闸,主触点 KM 断开主电路,电动机停止运转,实现了电动机的过载保护。

### (3) 欠压和失压保护

在电动机正常运行时,如果因为电源电压的消失而使电动机停转,那么在电源电压恢复时电动机就可能自行启动,电动机的自启动可能会造成人身事故或设备事故。防止电源电压恢复时电动机自启动的保护也叫零电压保护。

在电动机正常运行时,电源电压过分降低会引起电动机转速下降和转矩降低,若负载转矩不变,使电流过大,造成电动机停转和损坏电机。由于电源电压过分降低可能会引起一些电器释放,造成电路不正常工作,可能会产生事故。因此需要在电源电压下降达到最小允许的电压值时将电动机电源切除,这样的保护称为欠电压保护。

图 2-2 电路中,依靠接触器本身实现欠压和失压保护。当电源电压低到一定程度或失电时,接触器 KM 的电磁吸力小于反力,电磁机构会释放,主触点把主电源断开,电动机停止运转。这时如果电源恢复,由于控制电路失去自锁,电动机不会自行启动。只有操作人员再次按下启动按钮 SB2,电动机才会重新启动。

以上这三种保护是三相笼型异步电动机常用的保护环节,它对保证三相笼型异步电动机安全运行非常重要。

## 2.2.2 正反转控制线路

各种生产机械常常要求具有上下、左右、前后等相反方向的运动,如机床工作台的往复运动,就要求电动机能可逆运行。由电动机原理可知,三相异步电动机的三相电源进线中任意两相对调,电动机即可反向运转。因此可借助正反向接触器改变定子绕组相序来实现正反向工作,其线路如图 2-3 所示。

当误操作即同时按正反向启动按钮 SB2 和 SB3 时,若采用图 2-3(a)所示线路,将造成短路故障,如图中虚线所示,因此正反向间需要有一种联锁关系。通常采用图 2-3(b)所示的电路,将其中一个接触器的常闭触点串入另一个接触器线圈电路中,则任一接触器线圈先带电后,即使按下相反方向按钮,另一接触器也无法得电,这种联锁通常称为互锁,即两者存在相互

制约的关系。图 2-3(b)所示的电路要实现反转运行,必须先停止正转运行,再按反向启动按钮才行,反之亦然。所以这个电路称为“正—停—反”控制。图 2-3(c)所示的电路可以实现不按停止按钮,直接按反向按钮就能使电动机反向工作。所以这个电路称为“正—反—停”控制。

工程上通常还使用带有机械互锁的可逆接触器,进一步保证两者不能同时通电,提高可靠性。带机械互锁装置的可逆接触器产品很多,如 TE 公司的 LC2-D 系列。

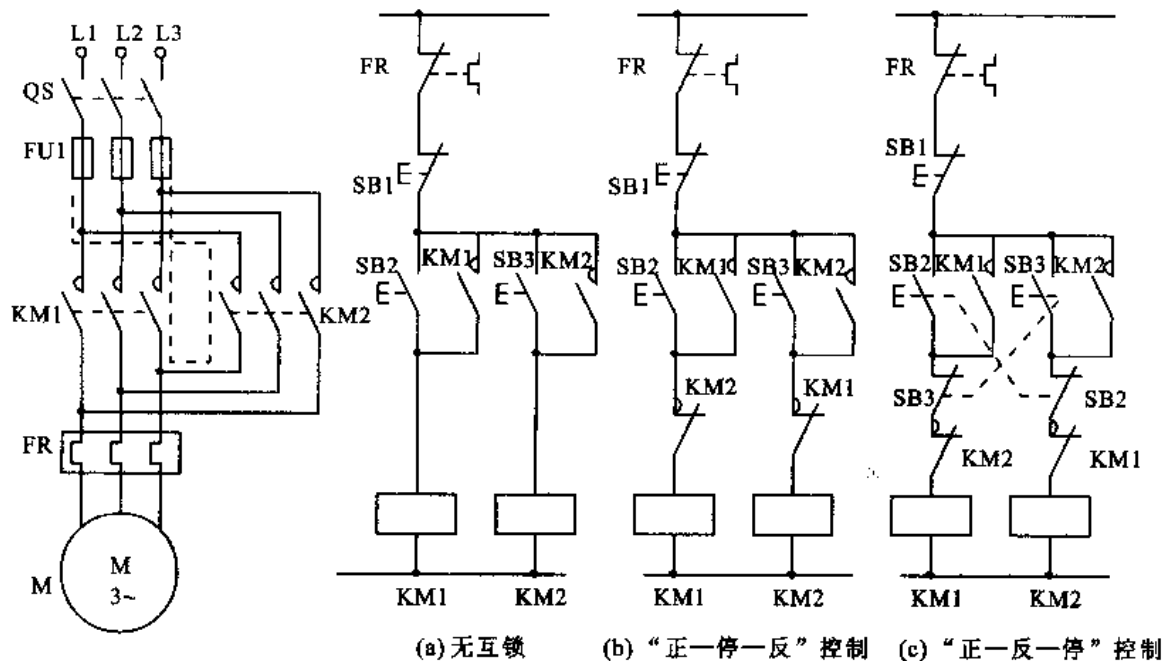


图 2-3 正反向工作的控制线路

### 2.2.3 点动控制线路

在生产实践中,有的生产机械需要点动控制,有的生产机械既需要按常规工作,又需要点动控制。图 2-4 所示为能实现点动的几种控制线路。

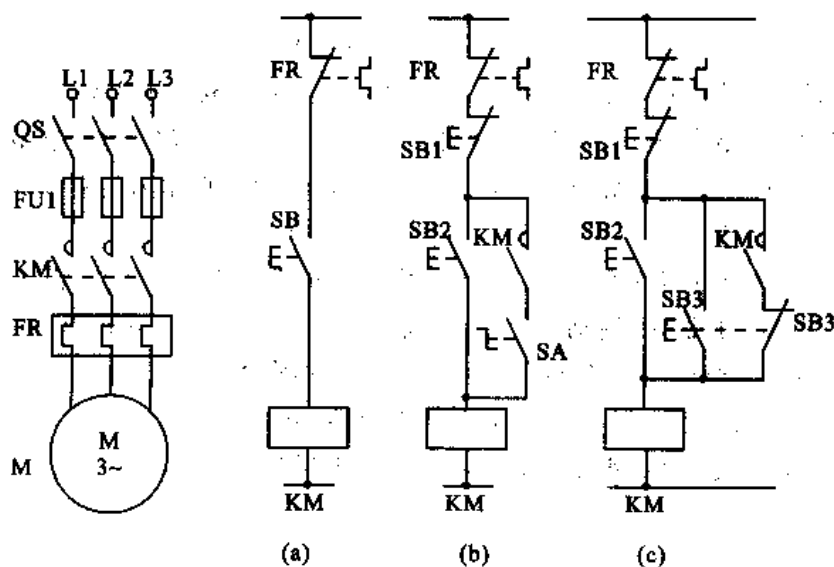


图 2-4 几种点动控制线路

图 2-4(a)所示是最基本的点动控制。启动按钮 SB 没有并联接触器 KM 的自锁触点,按下 SB, KM 线圈通电,电机启动运行;松开 SB, KM 线圈又断电释放,电机停止运转。

图 2-4(b)所示是带转换开关 SA 的点动控制线路。当需要点动控制时,只要把开关 SA 断开,由按钮 SB2 来进行点动控制。当需要正常运行时,只要把开关 SA 合上,将 KM 的自锁触点接入,即可实现连续控制。

图 2-4(c)中增加了一个复合按钮 SB3 来实现点动控制。需要点动控制时,按下点动按钮 SB3,其常闭触点先断开自锁电路,常开触点后闭合,接通启动控制电路, KM 线圈通电,衔铁被吸合,主触点闭合接通三相电源,电机启动运转;当松开点动按钮 SB3 时,其常开触点先断开,常闭触点后闭合, KM 线圈断电释放,主触点断开电源,电机停止运转。图中由按钮 SB2 和 SB1 来实现连续控制。

在读电气控制原理图时,大家一定要注意复合按钮常开触点和常闭触点的动作顺序。

#### 2.2.4 多点控制线路

有些机械和生产设备,由于种种原因,常要在两地或两个以上的地点进行操作。例如:重型龙门刨床,有时在固定的操作台上控制,有时需要站在机床四周用悬挂按钮控制;有些场合,为了便于集中管理,由中央控制台进行控制,但每台设备调整检修时,又需要就地进行机旁控制等。

要在两地进行控制,就应该有两组按钮,而且这两组按钮的连接原则必须是:常开按钮要并联,即逻辑“或”的关系;常闭停止按钮应串联,即逻辑“与非”的关系。图 2-5 所示就是实现两地控制的控制电路。这一原则也适用于三地或更多地点的控制。

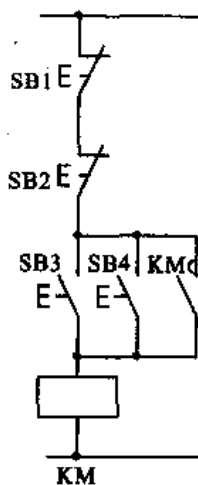


图 2-5 实现多地点控制线路

#### 2.2.5 顺序控制线路

生产实践中常要求各种运动部件之间能够按顺序工作。例如车床主轴转动时要求油泵先给齿轮箱提供润滑油,即要求保证润滑泵电动机启动后主拖动电动机才允许启动,也就是控制对象对控制线路提出了按顺序工作的联锁要求。如图 2-6 所示, M1 为油泵电动机, M2 为主拖动电动机。将控制油泵电动机的接触器 KM1 的常开辅助触点串入控制主拖动电动机的接触器 KM2 的线圈电路中,可以实现按顺序工作的联锁要求。

图 2-7 所示是采用时间继电器,按时间顺序启动的控制线路。主电路与图 2-6 中的主电路相同,线路要求电动机 M1 启动  $t$  秒后,电动机 M2 自动启动。可利用时间继电器的延时闭合常开触点来实现。按启动按钮 SB2,接触器 KM1 线圈通电并自锁,电动机 M1 启动,同时时间继电器 KT 线圈也通电。定时  $t$  秒到,时间继电器延时闭合的常开触点 KT 闭合,接触器 KM2 线圈通电并自锁,电动机 M2 启动,同时接触器 KM2 的常闭触点切断了时间继电器 KT 的线圈电源。

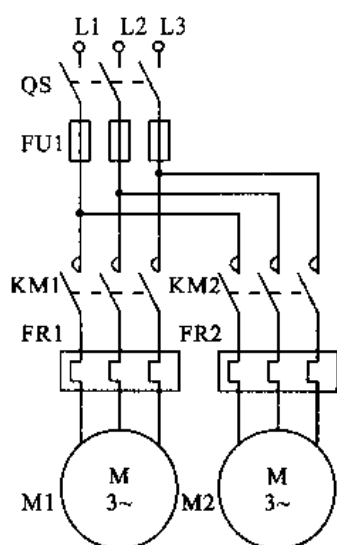


图 2-6 按顺序工作时的控制线路

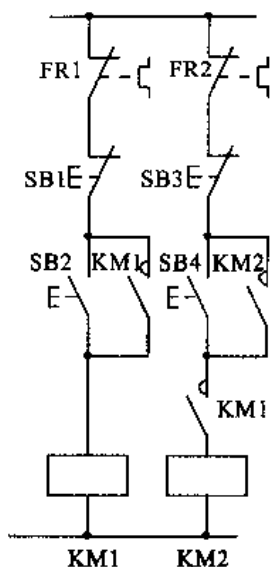
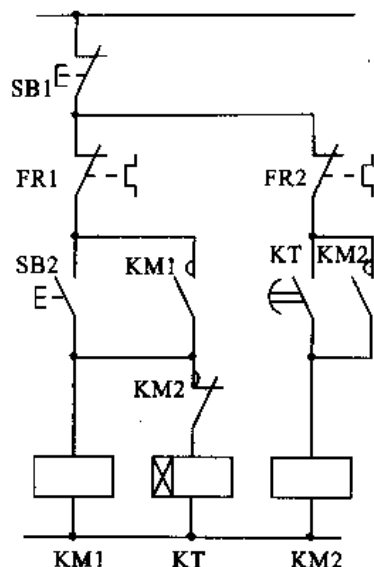


图 2-7 采用时间继电器的顺序启动控制线路



### 2.2.6 自动循环控制线路

在生产实践中,有些生产机械的工作台需要自动往复运动,如龙门刨床、导轨磨床等。图 2-8 为最基本的自动往复循环控制线路,它是利用行程开关实现往复运动控制的,这通常称为行程控制。

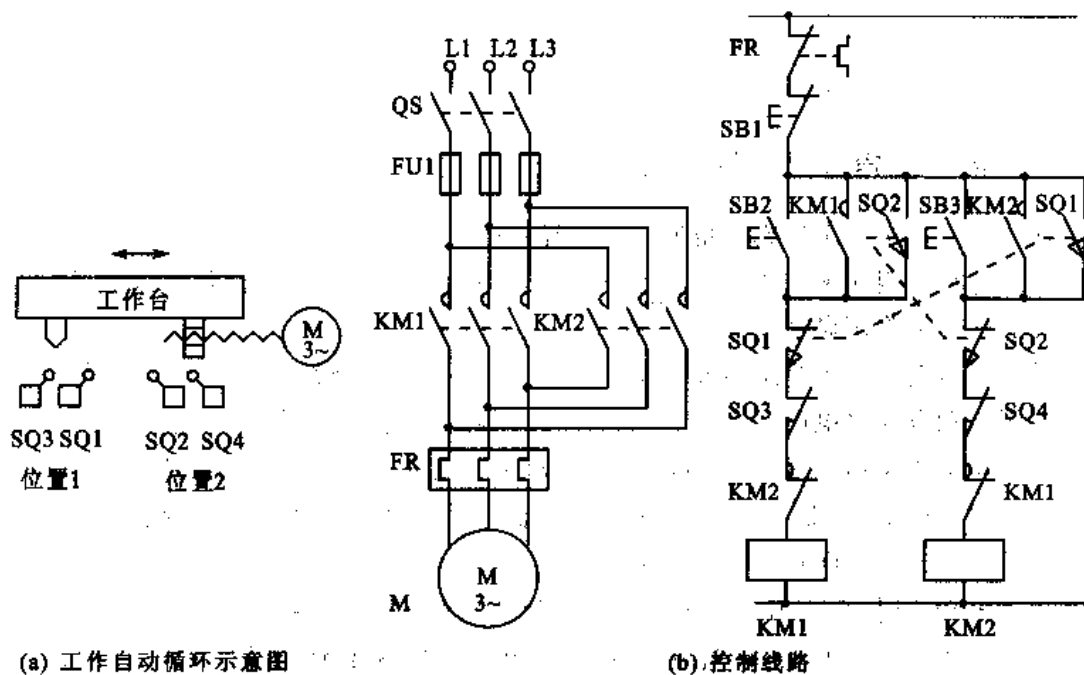


图 2-8 自动循环往复控制线路

限位开关 SQ1 放在左端需要反向的位置,而 SQ2 放在右端需要反向的位置,机械挡铁要装在运动部件上。启动时利用正向或反向启动按钮,如按正转按钮 SB2, KM1 通电吸合并自锁,电动机作正向旋转并带动工作台左移。当工作台移至左端并碰到 SQ1 时,将 SQ1 压下,

其常闭触点断开,切断 KM1 接触器线圈电路,同时,使其常开触点闭合,接通反转接触器 KM2 线圈电路。此时电动机由正向旋转变为反向旋转,带动工作台向右移动,直到压下 SQ2 限位开关,电动机由反转变为正转,工作台向左移动。因此工作台实现自动的往复循环运动。

由上述控制情况可以看出,运动部件每经过一个自动往复循环,电动机要进行两次反接制动,会出现较大的反接制动电流和机械冲击。因此这种电路只适用于电动机容量较小,循环周期较长、电动机转轴具有足够的刚性的拖动系统中。另外,在选择接触器容量时应比一般情况下选择的容量大一些。

除了利用限位开关实现往复循环之外,还可以做限位保护,如图 2-8 中的 SQ3、SQ4 分别为左、右超限限位保护用的行程开关。

机械式的行程开关容易损坏,现在多用接近开关或光电开关来取代行程开关实现行程控制。

## 2.3 三相笼型异步电动机降压启动控制线路

较大容量的笼型异步电动机(大于 10 kW)直接启动时,电流为其标称额定电流的 4~8 倍,启动电流较大,会对电网产生巨大冲击,所以一般都采用降压方式来启动。即启动时降低加在电动机定子绕组上的电压,启动后再将电压恢复到额定值,使之在正常电压下运行。因电枢电流和电压成正比,所以降低电压可以减小启动电流,防止在电路中产生过大的电压降,减少对线路电压的影响。

降压启动方式有定子电路串电阻(或电抗)、星形—三角形、自耦变压器、延边三角形和使用软启动器等多种。其中定子电路串电阻和延边三角形方法已基本不用,常用的方法是星形—三角形降压启动和使用软启动器。

### 2.3.1 星形—三角形降压启动控制线路

正常运行时定子绕组接成三角形的笼型异步电动机,可采用星形—三角形降压启动方式来限制启动电流。因功率在 4 kW 以上的三相笼型异步电动机均为三角形接法,因此都可以采用星形—三角形降压启动方式。

启动时将电动机定子绕组接成星形,加到电动机的每相绕组上的电压为额定值的  $1/\sqrt{3}$ ,从而减小了启动电流对电网的影响。当转速接近额定转速时,定子绕组改接成三角形,使电动机在额定电压下正常运转,图 2-9(a)所示为星形—三角形转换绕组连接图。星形—三角形降压启动线路如图 2-9(b)所示。这一线路的设计思想是按时间原则控制启动过程,待启动结束后按预先整定的时间换接成三角形接法。

当启动电动机时,合上刀闸开关 QS,按下启动按钮 SB2,接触器 KM、KM<sub>Y</sub> 与时间继电器 KT 的线圈同时得电,接触器 KM<sub>Y</sub> 的主触点将电动机接成星形并经过 KM 的主触点接至电源,电动机降压启动。当 KT 的延时时间到,KM<sub>Y</sub> 线圈失电,KM<sub>Δ</sub> 线圈得电,电动机主回路换接成三角形接法,电动机投入正常运转。

星形—三角形启动的优点在于,星形启动电流只是原来三角形接法直接启动时的  $1/3$ ,启动电流约为电动机额定电流的 2 倍左右,启动电流特性好、结构简单、价格低。缺点是启动转矩也相应下降为原来三角形的直接启动时的  $1/3$ ,转矩特性差。因而本线路适用于电动机空

载或轻载启动的场合。

工程上通常还可采用星形—三角形启动器来替代上述电路,其启动与上述原理相同。如 TE 公司的 LC3-D 系列产品。

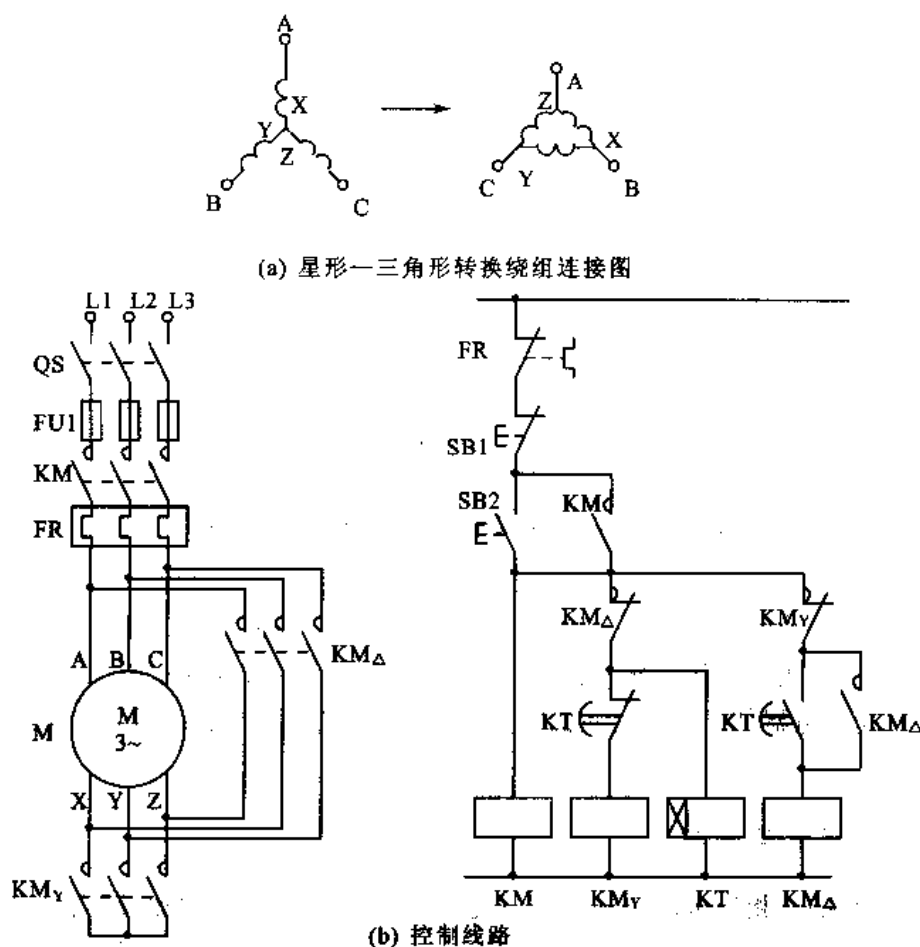


图 2-9 星形—三角形启动控制线路

### 2.3.2 自耦变压器降压启动控制线路

在自耦变压器降压启动的控制线路中,电动机启动电流的限制是靠自耦变压器降压来实现的。线路的设计思想也是采用时间继电器完成电动机由启动到正常运行的自动切换。启动时串入自耦变压器,启动结束时自动将其切除。

串自耦变压器降压启动的控制线路如图 2-10 所示。

当启动电动机时,合上刀开关 QS,按下启动按钮 SB2,接触器 KM1、KM3 与时间继电器 KT 的线圈同时得电,KM1、KM3 主触点闭合,电动机定子绕组经自耦变压器接至电源降压启动。当时间继电器 KT 延时时间到,一方面其常闭的延时触点打开,KM1、KM3 线圈失电,KM1、KM3 主触点断开,将自耦变压器切除;同时,KT 的常开延时触点闭合,接触器线圈 KM2 得电,KM2 主触点闭合,电动机投入正常运转。

串联自耦变压器启动的优点是,启动时对电网的电流冲击小,功率损耗小。缺点是,自耦变压器相对结构复杂,价格较高。这种方式主要用于较大容量的电动机,以减小启动电流对电

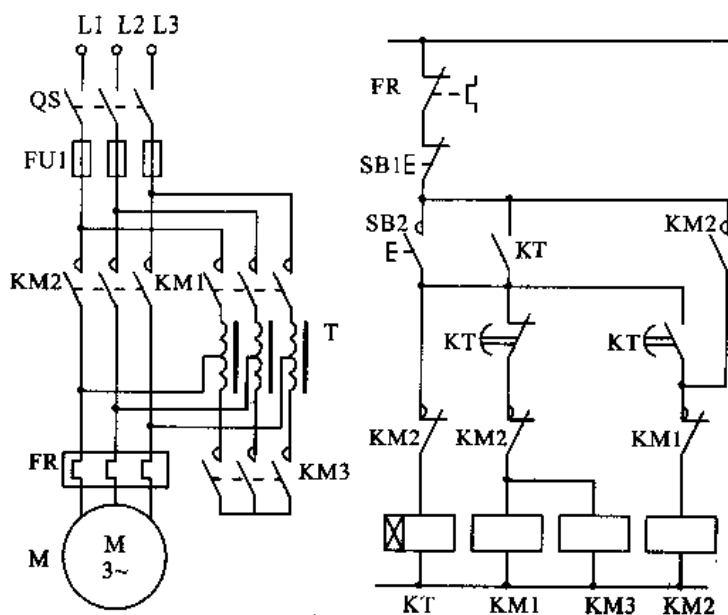


图 2-10 定子串自耦变压器降压启动控制线路

网的影响。

综合以上几种启动方法可见,一般均采用时间继电器及按照时间原则切换电压,以此实现降压启动。由于这种线路工作可靠,受外界因素(如负载,飞轮转动惯量以及电网电压)的影响较小,线路及时间继电器的结构都比较简单,因而在电动机启动控制线路中多采用时间控制其启动过程。

### 2.3.3 软启动器及其使用

前述几种传统的三相异步电动机的启动线路比较简单,不需要增加额外启动设备,但其启动电流冲击一般还很大,启动转矩较小而且固定不可调。电动机停机时都采用控制接触器触点断开,切掉电动机电源,电动机自由停车,这样也会造成剧烈的电网波动和机械冲击。如在直接启动方式下,启动电流为额定值的 4~8 倍,启动转矩为额定值的 0.5~1.5 倍;在定子串电阻降压启动方式下,启动电流为额定值的 4.5 倍,启动转矩为额定值的 0.5~0.75 倍;在星形—三角形启动方式下,启动电流为额定值的 1.8~2.6 倍,在星形—三角形切换时也会出现电流冲击,且启动转矩为额定值的 0.5 倍;而自耦变压器降压启动,启动电流为额定值的 1.7~4 倍,在电压切换时会出现电流冲击,启动转矩为额定值的 0.4~0.85 倍。因而上述这些方法经常用于对启动特性要求不高的场合。

在一些对启动要求较高的场合,可选用软启动装置,它采用电子启动方法。其主要特点是:具有软启动和软停车功能,启动电流、启动转矩可调节,另外还具有电动机过载保护等功能。

#### 1. 软启动器的工作原理

如图 2-11 所示为软启动器内部原理示意图。它主要由三相交流调压电路和控制电路构成。其基本原理是利用晶闸管的移相控制原理,通过控制晶闸管的导通角,改变其输出电压,达到通过调压方式来控制启动电流和启动转矩的目的。控制电路按预定的不同启动方式,通过检测主电路的反馈电流,控制其输出电压,可以实现不同的启动特性。最终软启动器输出全压,电动机全压运行。由于软启动器为电子调压并对电流实时检测,因此还具有对电动机和软启动器

本身的热保护、限制转矩和电流冲击、三相电源不平衡、缺相、断相等保护功能,并可实时检测并显示如电流、电压、功率因数等参数。

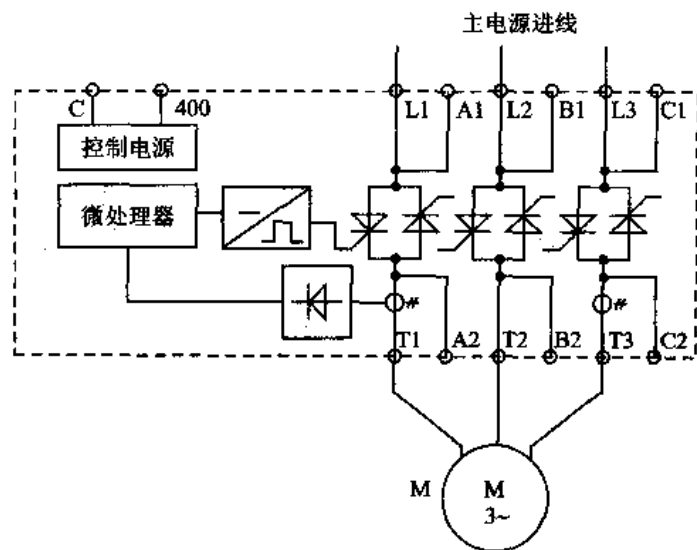


图 2-11 软启动器原理示意图

## 2. 软启动器的控制功能

异步电动机在软启动过程中,软启动器通过控制加到电动机上的电压来控制电动机的启动电流和转矩,启动转矩逐渐增加,转速也逐渐增加。一般软启动器可以通过改变参数设定得到不同的启动特性,以满足不同的负载特性要求。

### (1) 斜坡升压启动方式

斜坡升压启动特性曲线如图 2-12 所示。此种启动方式一般可设定启动初始电压  $U_0$  和启动时间  $t_1$ 。这种启动方式断开电流反馈,属开环控制方式。在电动机启动过程中,电压线性逐渐增加,在设定的时间内达到额定电压。这种启动方式主要用于一台软启动器并接多台电机或电动机功率远低于软启动器额定值的应用场合。

### (2) 转矩控制及启动电流限制启动方式

转矩控制及启动电流限制特性曲线如图 2-13 所示。此种启动方式一般可设定启动初始力矩  $T_0$ 、启动阶段力矩限幅  $T_{L1}$ 、力矩斜坡上升时间  $t_1$  和启动电流限幅  $I_{L1}$ 。这种启动方式引入电流反馈,通过计算间接得到负载转矩,属闭环控制方式。由于控制目标为转矩,故软启动器输出电压为非线性上升。图 2-13 同时给出启动过程中转矩  $T$ 、电压  $U$ 、电流  $I$  和电动机转速  $n$  的

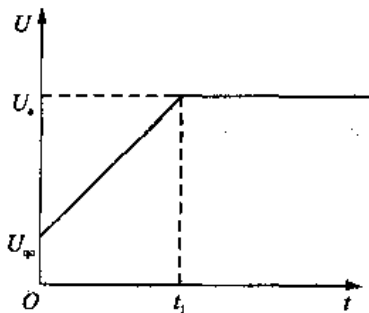


图 2-12 斜坡升压启动方式

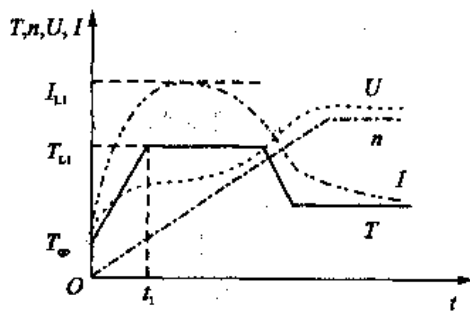


图 2-13 转矩控制及启动电流限制启动

曲线,其中转速曲线为恒加速度上升。

在电动机启动过程中,保持恒定的转矩使电动机转速以恒定加速度上升,实现平稳启动。在电动机启动的初始阶段,启动转矩逐渐增加,当转矩达到预先所设定的限幅值后保持恒定,直至启动完毕。在启动过程中,转矩上升的速率可以根据电动机负载情况调整设定。斜坡陡,转矩上升速率大,即加速度上升速率大,启动时间短。当负载较轻或空载启动时,所需启动转矩较低,可使斜坡缓和一些。由于在启动过程中,控制目标为电动机转矩,即电动机的加速度,即使电网电压发生波动或负载发生波动,通过控制电路自动通过增大或减小启动器的输出电压,也可以维持转矩设定值不变,保持启动的恒加速度。此种控制方式可以使电动机以最佳的启动加速度、以最快的时间完成平稳的启动,是应用最多的启动方式。

随着软启动器控制技术的发展,目前它大多采用转矩控制方式,也有采用电流控制方式,即电流斜坡控制及恒流升压启动方式。此种方式间接控制电动机电流来达到控制转矩目的,与转矩控制方式相比启动效果略差,但控制相对简单。

### (3) 电压提升脉冲启动方式

电压提升脉冲启动特性曲线如图 2-14 所示。此种启动方式一般可设定电压提升脉冲限幅  $U_{Li}$ 。升压脉冲宽度一般为 5 个电源周波,即 100 ms。在启动开始阶段,晶闸管在极短时间内按设定升压幅值启动,可得到较大的启动转矩,此阶段结束后,转入转矩控制及启动电流限制启动。该启动方法适用于重载并需克服较大静摩擦的启动场合。

### (4) 转矩控制软停车方式

当电动机需要停车时,立即切断电动机电源,属自由停车。传统的控制方式大都采用这种方法。但许多应用场合,不允许电动机瞬间停机。如高层建筑、楼宇的水泵系统,要求电动机逐渐停机,采用软启动器可满足这一要求。

软停车方式通过调节软启动器的输出电压逐渐降低而切断电源,这一过程时间较长且一般大于自由停车时间,故称为软停车方式。转矩控制软停车方式,是在停车过程中,匀速调整电动机转矩的下降速率,实现平滑减速。如图 2-15 所示为转矩控制软停车特性曲线。减速时间  $t_1$  一般是可设定的。

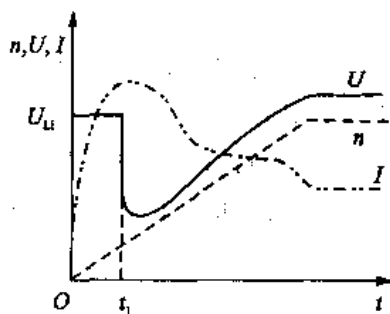


图 2-14 电压提升脉冲启动方式

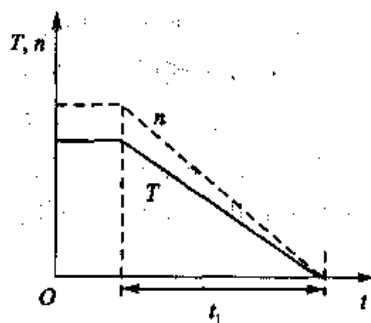


图 2-15 转矩控制软停车方式

### (5) 制动停车方式

当电动机需要快速停机时,软启动器具有能耗制动功能。在实施能耗制动时,软启动器向电动机定子绕组通入直流电,由于软启动器是通过晶闸管对电动机供电,因此很容易通过改变晶闸管的控制方式而得到直流电。如图 2-16 所示为制动停车方式特性曲线,一般可设定制动电流加入的幅值  $I_{Li}$  和时间  $t_1$ ,但制动开始到停车时间不能设定,时间长短与制动电流有关,应根据实

际应用情况,调节加入的制动电流幅值和时间来调节制动时间。

### 3. 软启动器的应用举例

目前国内外软启动器产品的技术发展很快,产品的型号很多,比较有代表性的有 ABB 公司的 PSA、PSD 和 PSDH 型;法国 TE 公司的 Altistart 46、Altistart 48 型;德国西门子的 3RW22 型;英国 CT 公司的 SX 型等。

下面以 TE 公司生产的 Altistart 46 型软启动器为例,介绍软启动器的典型应用。Altistart 46 型软启动器有标准负载和重型负载应用两大类,额定电流从 17~1 200 A 共 21 种额定值,电机功率 4~800 kW。其主要特点是:具有斜坡升压、转矩控制及启动电流限制、电压提升脉冲三种启动方式;具有转矩控制软停车、制动停车、自由停车三种停车方式;具有对电动机和软启动器本身的热保护、限制转矩和电流冲击、三相电源不平衡、缺相、断相和电动机运行中过流等保护功能并提供故障输出信号;具有实时检测并显示如电流、电压、功率因数等参数的功能,并提供模拟输出信号;提供本地端子控制接口和远程控制 RS-485 通信接口。

通过人机对话操作盘或通过 PC 机与通信接口连接,可显示和修改系统配置、参数。其主要参数设置范围如下。

- 启动电流可调节范围:额定值的 2~5 倍之间;
- 启动转矩可调节范围:额定值的 0.15~1.0 倍之间;
- 加速力矩斜坡时间可调节范围:1~60 s;
- 减速力矩斜坡可调节范围:1~60 s;
- 制动转矩可调节范围:额定值 0~100 %;
- 电压提升脉冲幅值可调节范围:额定电压的 50 %~100 %;
- 电动机运行时过流跳闸值可调节范围:额定值的 50 %~300 %。

#### (1) 电动机单向运行带旁路接触器、软启动、软停车或自由停车控制线路

如图 2-17 所示为三相异步电动机用软启动器启动控制线路。图中虚线框所示为软启动器,其中 C 和 400 为软启动器控制电源进线端子;L1、L2、L3 为软启动器主电源进线端子;T1、T2、T3 为连接电动机的出线端子;A1、A2、B1、B2、C1、C2 端子由软启动器三相晶闸管两端分别直接引出。当相对应端子短接时,相当于图 2-17 中 KM2 主触点闭合时,将软启动器内部晶闸管短接,但此时软启动器内部的电流检测环节仍起作用,即此时软启动器对电动机保护功能仍起作用。

PL 是软启动器为外部逻辑输入提供的 +24 V 电源;L+ 为软启动器逻辑输出部分的外接输入电源,在图中直接由 PL 提供。

STOP、RUN 分别为软停车和软启动控制信号,接线方式分为:三线制控制、二线制控制和通信远程控制。三线制控制,要求输入信号为脉冲输入型;二线制控制,要求输入信号为电平输入型;通信远程控制时,图 2-17 中将 PL 与 STOP 端子短接,启停要使用通信口远程控制。图 2-17 所示控制线路为三线制控制方式接线。

KA1 和 KA2 为输出继电器。KA1 为可编程输出继电器,可设置成故障继电器或隔离继电器。若 KA1 设置为故障继电器,则当软启动器控制电源上电时,KA1 闭合;当软启动器发生故障

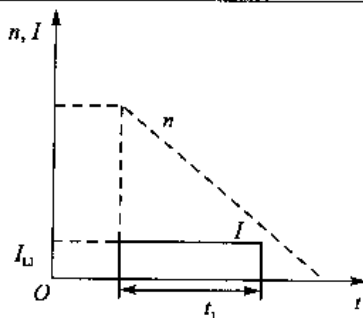


图 2-16 制动停车方式

时,KA1 断开。若 KA1 设置为隔离继电器,则当软启动器接收到启动信号时,KA1 闭合;当软启动器软停车结束时,或软启动器在自由停车模式下接收到停车信号时,或在运行过程中出现故障时,KA1 断开。KA2 为启动结束继电器,当软启动器完成启动过程后,KA2 闭合;当软启动器接收到停车信号或出现故障时,KA2 断开。

如图 2-17 所示为电动机单向运行、软启动、软停车或自由停车控制线路。KA1 设置为隔离继电器。此软启动器接有进线接触器 KM1。当开关 QS 合闸,按启动按钮 SB2,则 KA1 触点闭合,KM1 线圈上电,使其主触点闭合,主电源加入软启动器。电动机按设定的启动方式启动,当启动完成后,内部继电器 KA2 常开触点闭合,KM2 接触器线圈吸合,电动机转由旁路接触器 KM2 触点供电,同时将软启动器内部的功率晶闸管短接,电动机通过接触器由电网直接供电。但此时过载、过流等保护仍起作用,KA1 相当于保护继电器的触点。若发生过载、过流,则切断接触器 KM1 电源,则软启动器进线电源切除。因此电动机不需要额外增加过载保护电路。正常停车时,按停车按钮 SB1,停止指令使 KA2 触点断开,旁路接触器 KM2 跳闸,使电机软停车,软停车结束后,KA1 触点断开。按钮 SB3 为紧急停车用,当按下 SB3 时,接触器 KM1 失电,软启动器内部的 KA1 和 KA2 触点复位,使 KM2 失电,电机自由停转。

由于带有旁路接触器,该电路有如下优点:在电动机运行时可以避免软启动器产生的谐波;软启动器仅在启动和停车时工作,可以避免长期运行使晶闸管发热,延长了使用寿命。

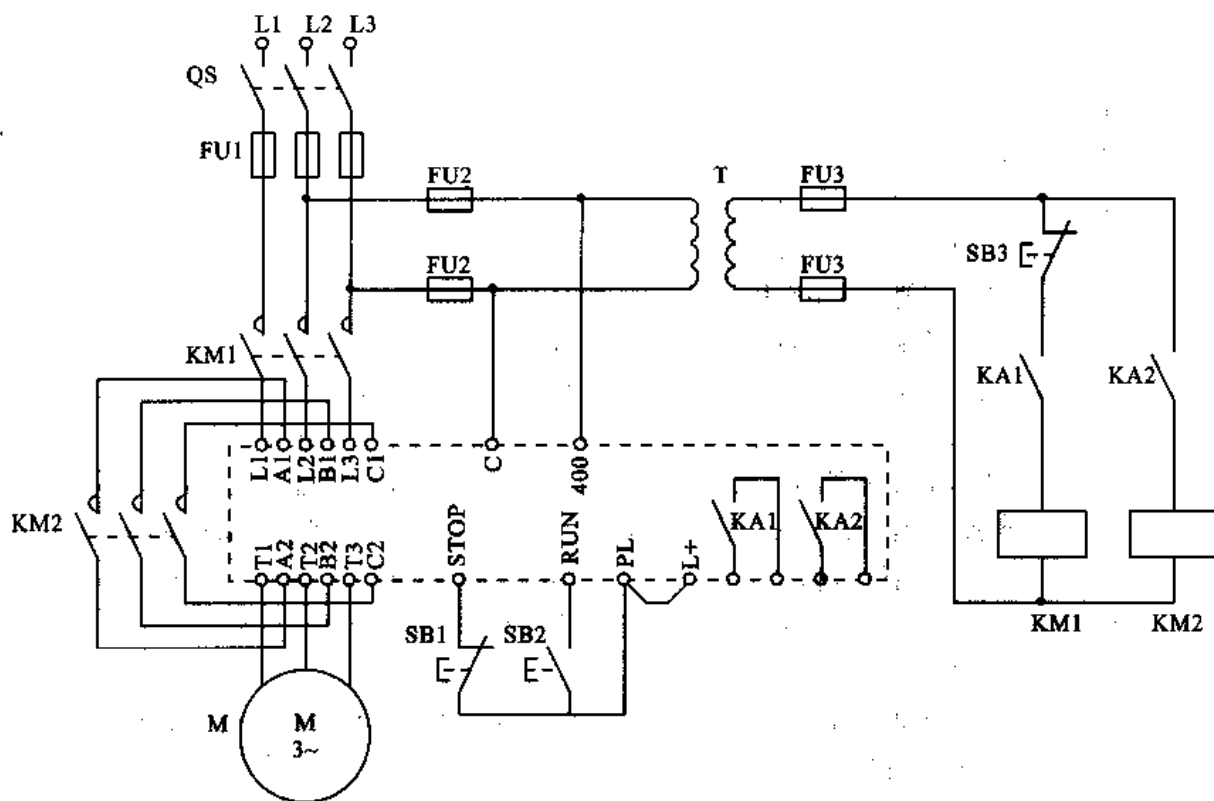


图 2-17 电动机单向运行、软启动、软停车或自由停车控制线路

## (2) 单台软启动器启动多台电动机

用一台软启动器对多台电动机进行软启动,可以降低控制系统的成本。通过设计适当的电路可以实现对多台电动机软启动、软停车控制,但不能同时启动或停机,只能一台台分别启动或停机。考虑到实现一台软启动器对多台电动机既能软启动又能软停车,控制线路比较复杂,而且

还要使用软启动器内部的一些特殊功能,所以下面仅介绍用一台软启动器对 2 台电动机进行软启动、自由停车控制的电路,如图 2-18 所示。软启动器的启动、停止采用二线制控制方式,即将 RUN 和 STOP 端子连接到一起,通过一控制触点 KA5 与 PL 端子相连。KA5 触点接通表示启动信号,断开表示停车信号。由于电动机启动结束后,由旁路接触器为电动机供电,图 2-18(a) 中主电路的接线方式将整个软启动器短接,因此软启动器的各种保护对电动机无效,故每台电动机要各自增加过载保护的热继电器。

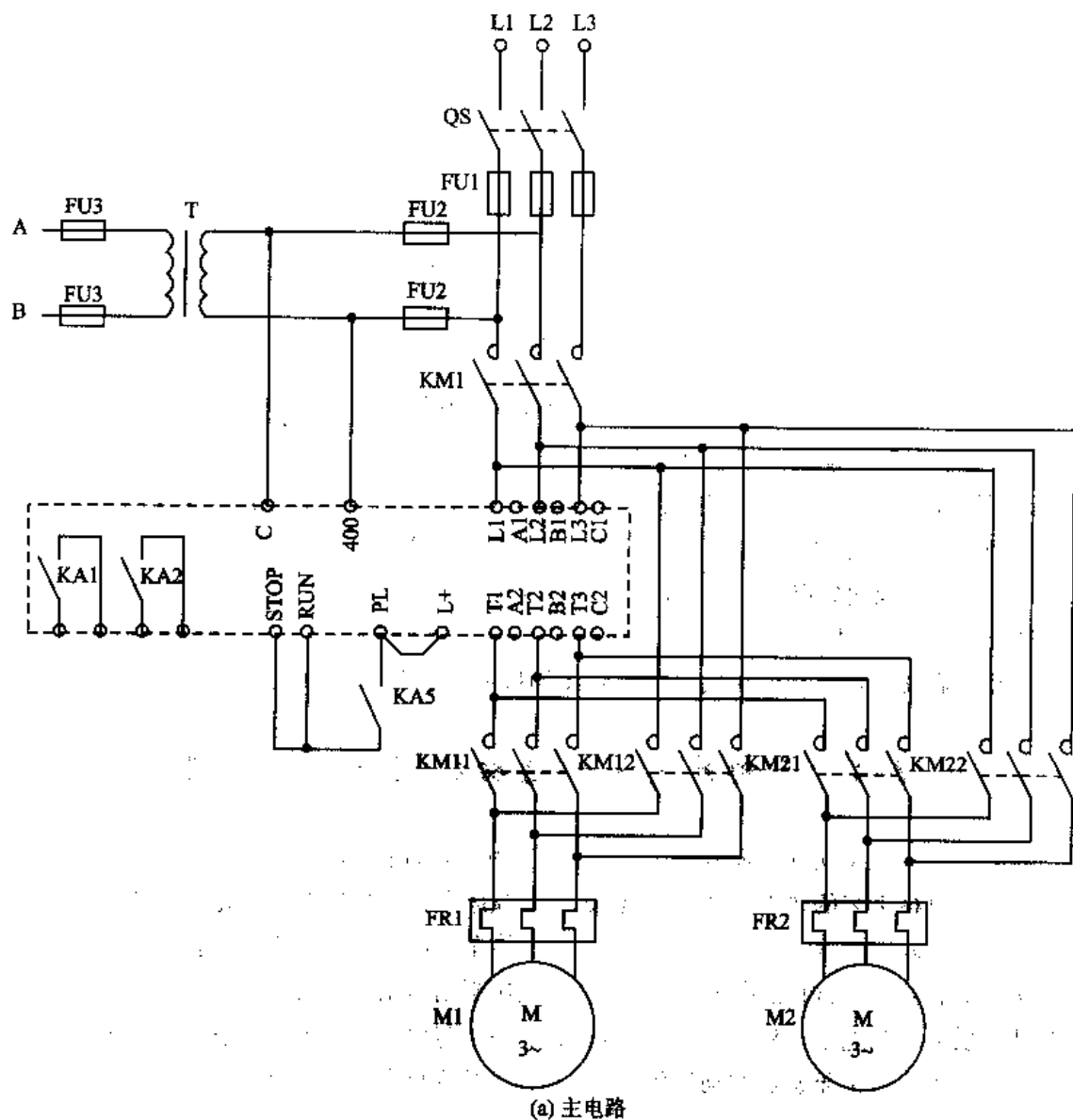


图 2-18 一台软启动器对 2 台电动机进行软启动、自由停车控制

工作原理如下:将 KA1 设置为隔离继电器。在图 2-18(b)控制电路中,按 SB2 按钮,接触器 KM1 线圈通电并自锁,软启动器的进线电源上电。若启动第一台电动机 M1,按启动按钮 SB12,接触器 KM11 线圈通电、中间继电器 KA5 线圈通电、启动信号加入软启动器,隔离继电器 KA1 触点接通、中间继电器 KA3 线圈通电,KA3 常开触点使 KM11 自锁,电动机软启动开始。当启动结束时,软启动器的启动结束继电器 KA2 触点闭合,中间继电器 KA4 线圈通电,KA4 常开触点使旁路接触器 KM12 线圈通电并自锁,此时 KM11 和 KM12 均接通。软启动器旁路后,使隔离继

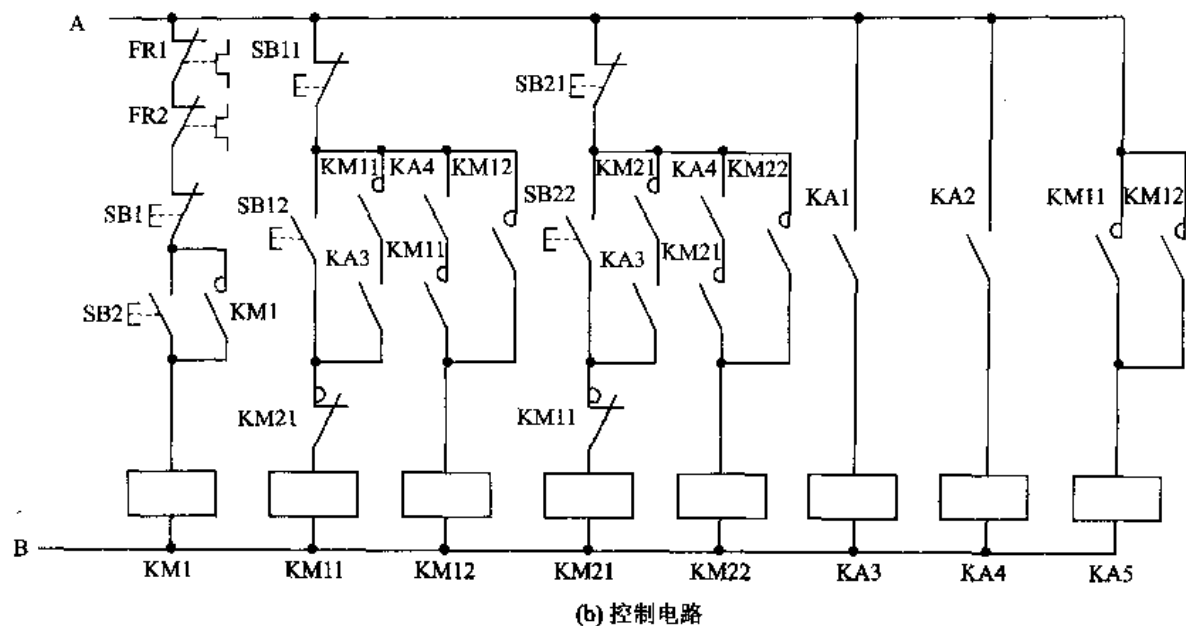


图 2-18 一台软启动器对 2 台电动机进行软启动、自由停车控制(续)

电器 KA1 触点断开、启动结束继电器 KA2 触点也断开, KA3 线圈断电, KA3 常开触点使 KM11 线圈自锁回路断开, KM11 线圈失电, 第一台电动机从软启动器上切除, 此时软启动器处于空闲状态。同理, 若对第二台电动机软启动, 则按启动按钮 SB22。若使第一台电动机停机, 按停止按钮 SB11, 则 KM12 线圈失电, 主触点断开, 电动机自由停机。第二台电动机的启、停控制过程分析与上述类似, 不再复述。

为防止软启动器带两台电动机同时启动, KM11 和 KM21 线圈回路增加有互锁触点。

## 2.4 三相笼型异步电动机制动控制线路

三相异步电动机从切除电源到完全停止旋转, 由于惯性的作用, 总要经过一段时间, 这往往不能适应某些机械工艺的要求。如万能铣床、卧式镗床和组合机床等。无论是从提高生产效率, 还是从安全及准确定位等方面考虑, 都要求能迅速停车, 因此要求对电动机进行制动控制。制动控制方法一般有两大类: 机械制动和电气制动。机械制动是用机械装置来强迫电动机迅速停车; 电气制动实质上是当电动机停车时, 给电动机加上一个与原来旋转方向相反的制动转矩, 迫使电动机转速迅速下降。由于机械制动比较简单, 下面我们着重介绍电气制动控制线路, 它包括反接制动和能耗制动。

### 2.4.1 反接制动控制线路

反接制动是利用改变电动机电源的相序, 使定子绕组产生相反方向的旋转磁场, 因而产生制动转矩的一种制动方法。

由于反接制动时, 转子与旋转磁场的相对速度接近于两倍的同步转速, 所以定子绕组中流过的反接制动电流相当于全电压直接启动时电流的两倍, 因此反接制动特点之一是制动迅速, 效果好, 但冲击大, 通常仅适用于 10 kW 以下的小容量电动机。为了减小冲击电流, 通常要求串接一

定的电阻以限制反接制动电流。这个电阻称为反接制动电阻。反接制动电阻的接线方法有对称和不对称两种接法,显然采用对称电阻接法可以在限制制动转矩的同时,也限制了制动电流,而采用不对称电阻的接法,只限制了制动转矩,未加制动电阻的那一相,仍具有较大的电流,因此一般采用对称接法。反接制动的另一要求是在电动机转速接近于零时,要及时切断反相序的电源,以防止电动机反向再启动。

### 1. 电动机单向运行反接制动控制线路

反接制动的关键在于电动机电源相序的改变,且当转速下降到接近于零时,能自动将电源切除,为此采用了速度继电器来检测电动机的速度变化。在  $120 \sim 3\,000 \text{ r/min}$  范围内速度继电器触点动作,当转速低于  $100 \text{ r/min}$  时,其触点恢复原位。

图 2-19 为带制动电阻的单向反接制动的控制线路。启动时,按下启动按钮 SB2,接触器 KM1 线圈通电并自锁,电动机 M 通电旋转。在电动机正常运转时,速度继电器 KS 的常开触点闭合,为反接制动做好了准备。停车时,按下停止按钮 SB1,其常闭触点断开,接触器 KM1 线圈断电,电动机 M 脱离电源。由于此时电动机的惯性转速还很高,KS 的常开触点仍然处于闭合状态,所以,当 SB1 常开触点闭合时,反接制动接触器 KM2 线圈通电并自锁,其主触点闭合,使电动机定子绕组得到与正常运转相序相反的三相交流电源,电动机进入反接制动状态,电动机转速迅速下降。当电动机转速低于速度继电器动作值时,速度继电器常开触点复位,接触器 KM2 线圈电路被切断,反接制动结束。

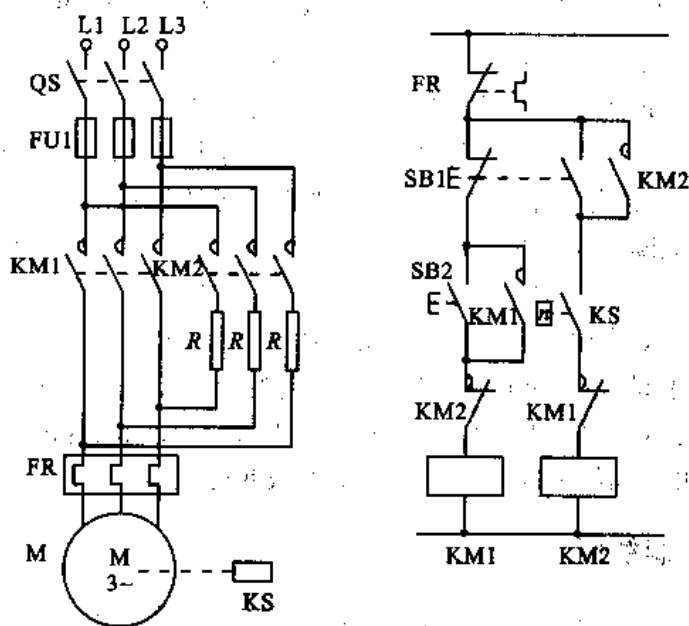


图 2-19 单向反接制动的控制线路

### 2. 具有反接制动电阻的可逆运行反接制动控制线路

图 2-20 所示为具有反接制动电阻的可逆运行反接制动控制线路。图中电阻  $R$  是反接制动电阻,同时也具有限制启动电流的作用。 $KS_1$  和  $KS_2$  分别为速度继电器 KS 的正转和反转常开触点。

该电路工作原理如下:按下正转启动按钮 SB2,中间继电器 KA3 线圈通电并自锁,其常闭触点打开,互锁中间继电器 KA4 线圈电路,KA3 常开触点闭合,使接触器 KM1 线圈通电,KM1 主

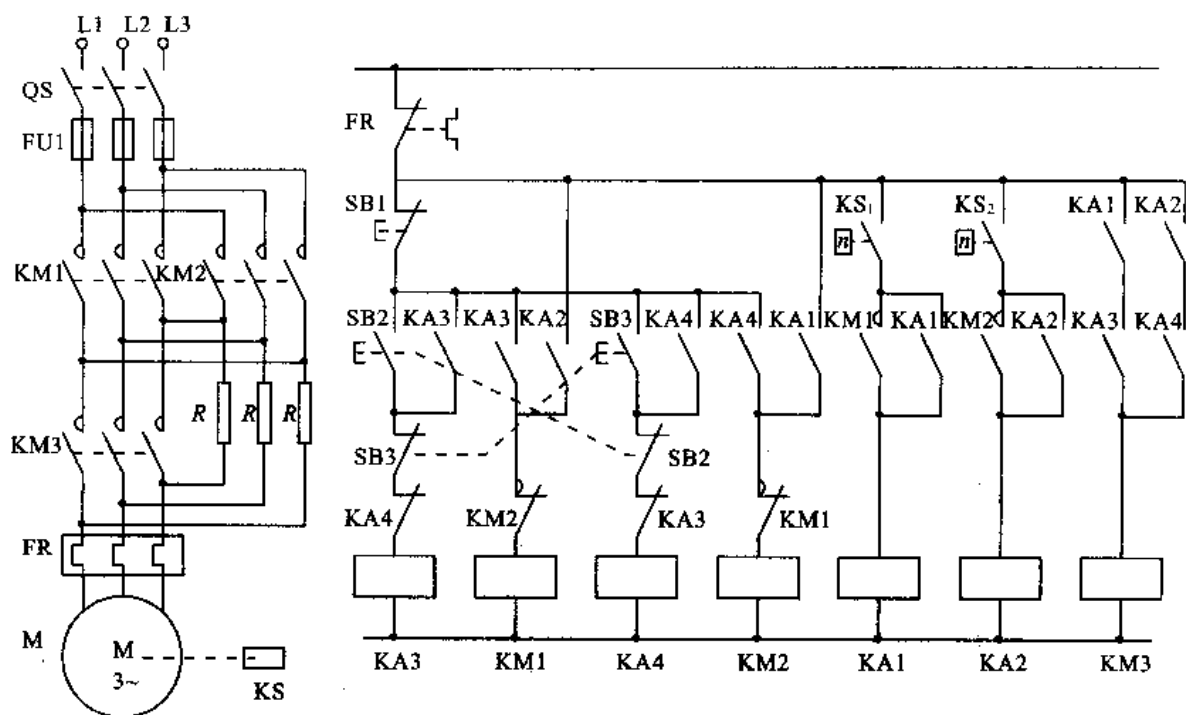


图 2-20 具有反接制动电阻的可逆运行反接制动的控制线路

触点闭合使定子绕组经电阻  $R$  接通正序三相电源,电动机  $M$  开始降压启动。当电动机转速上升到一定值时,速度继电器的正转常开触点  $KS_1$  闭合,使中间继电器  $KA_1$  通电并自锁,这时由于  $KA_1$ 、 $KA_3$  的常开触点闭合,接触器  $KM_3$  线圈通电,于是电阻  $R$  被短接,定子绕组直接加以额定电压,电动机转速上升到稳定工作转速。在电动机正常运转过程中,若按下停止按钮  $SB_1$ ,则  $KA_3$ 、 $KM_1$ 、 $KM_3$  三只线圈相继断电。由于此时电动机转子的惯性转速仍然很高,使速度继电器的正转常开触点  $KS_1$  尚未复原,中间继电器  $KA_1$  仍处于工作状态,所以在接触器  $KM_1$  常闭触点复位后,接触器  $KM_2$  线圈便通电,其常开触点闭合,使定子绕组经电阻  $R$  获得反相序三相交流电源,对电动机进行反接制动,电动机转速迅速下降。当电动机转速低于速度继电器动作值时,速度继电器常开触点复位,  $KA_1$  线圈断电,接触器  $KM_2$  释放,反接制动过程结束。

电动机反向启动和制动停车过程与正转时相同,故此处不再复述。

#### 2.4.2 能耗制动控制线路

所谓能耗制动,就是在电动机脱离三相交流电源之后,定子绕组上加一个直流电压,即通入直流电流,利用转子感应电流与静止磁场的作用以达到制动的目的。根据能耗制动时间控制原则,可用时间继电器进行控制,也可以根据能耗制动速度原则,用速度继电器进行控制。下面分别用单向能耗制动和正反向能耗制动控制线路为例来说明。

##### 1. 电动机单向运行能耗制动控制线路

图 2-21 为时间原则控制的单向能耗制动控制线路。在电动机正常运行时,若按下停止按钮  $SB_1$ ,电动机由于  $KM_1$  断电释放而脱离三相交流电源,而直流电源则由于接触器  $KM_2$  线圈通电使其主触点闭合而加入定子绕组,时间继电器  $KT$  线圈与  $KM_2$  线圈同时通电并自锁,于是电动机进入能耗制动状态。当其转子的惯性速度接近于零时,时间继电器延时打开的常闭触

点断开接触器 KM2 的线圈电路。由于 KM2 常开辅助触点的复位,时间继电器 KT 线圈的电源也被断开,电动机能耗制动结束。图中 KT 的瞬时常开触点的作用是为了当出现 KT 线圈断线或机械卡住故障时,电动机在按下按钮 SB1 后仍能迅速制动,两相的定子绕组不致于长期接入能耗制动的直流电流。所以,在 KT 发生故障后,该线路具有手动控制能耗制动的能力,即只要使停止按钮处于按下的状态,电动机就能实现能耗制动。

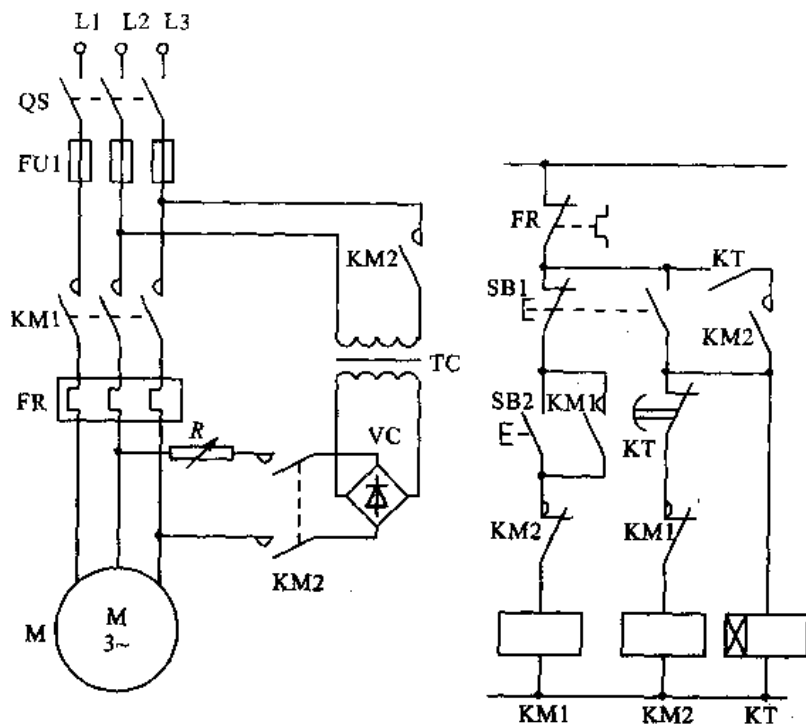


图 2-21 以时间原则控制的单向能耗制动线路

图 2-22 为速度原则控制的单向能耗制动控制线路。该线路与图 2-21 的控制线路基本相同,这里仅是在控制电路中取消了时间继电器 KT 的线圈及其触点电路,而在电动机轴伸端安装了速度继电器 KS,并且用 KS 的常开触点取代了 KT 延时打开的常闭触点。这样一来,该线路中的电动机在刚刚脱离三相交流电源时,由于电动机转子的惯性速度仍然很高,速度继电器 KS 的常开触点仍然处于闭合状态,所以接触器 KM2 线圈能够依靠 SB1 按钮的按下通电自锁。于是,两相定子绕组获得直流电源,电动机进入能耗制动。当电动机转子的惯性速度低于速度继电器 KS 动作值时,KS 常开触点复位,接触器 KM2 线圈断电释放,能耗制动结束。

## 2. 电动机可逆运行能耗制动控制线路

图 2-23 为电动机按时间原则控制的可逆运行的能耗制动控制线路。在其正常的正向运转过程中,需要停止时,可按下停止按钮 SB1,使 KM1 断电,KM3 和 KT 线圈通电并自锁。KM3 常闭触点断开,起着锁住电动机启动电路的作用;KM3 常开触点闭合,使直流电压加至定子绕组,电动机进行正向能耗制动。电动机正向转速迅速下降,当其接近于零时,时间继电器延时打开的常闭触点 KT 断开接触器 KM3 线圈电源。由于 KM3 常开辅助触点的复位,时间继电器 KT 线圈也随之失电,电动机正向能耗制动结束。反向启动与反向能耗制动其过程与上述正向情况相同。

电动机可逆运行能耗制动也可以以速度原则,用速度继电器取代时间继电器,同样能达到制

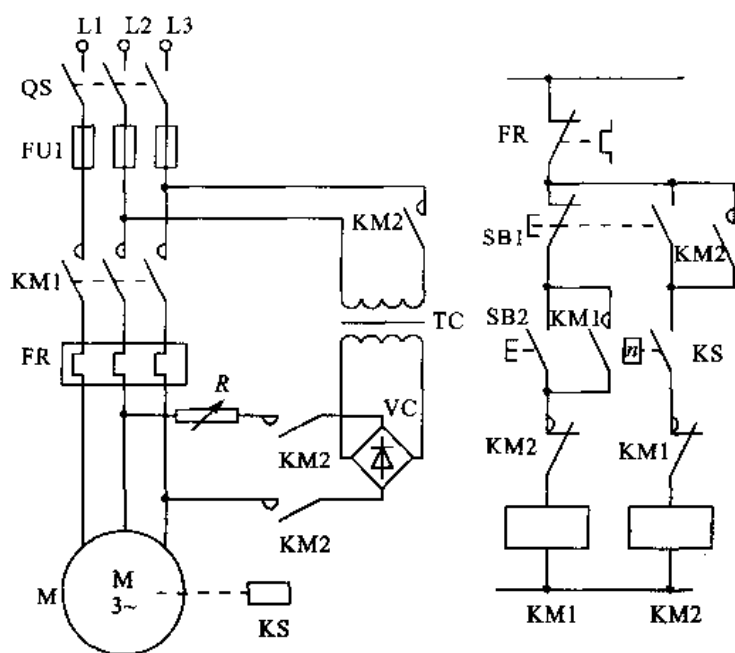


图 2-22 以速度原则控制的单向能耗制动控制线路

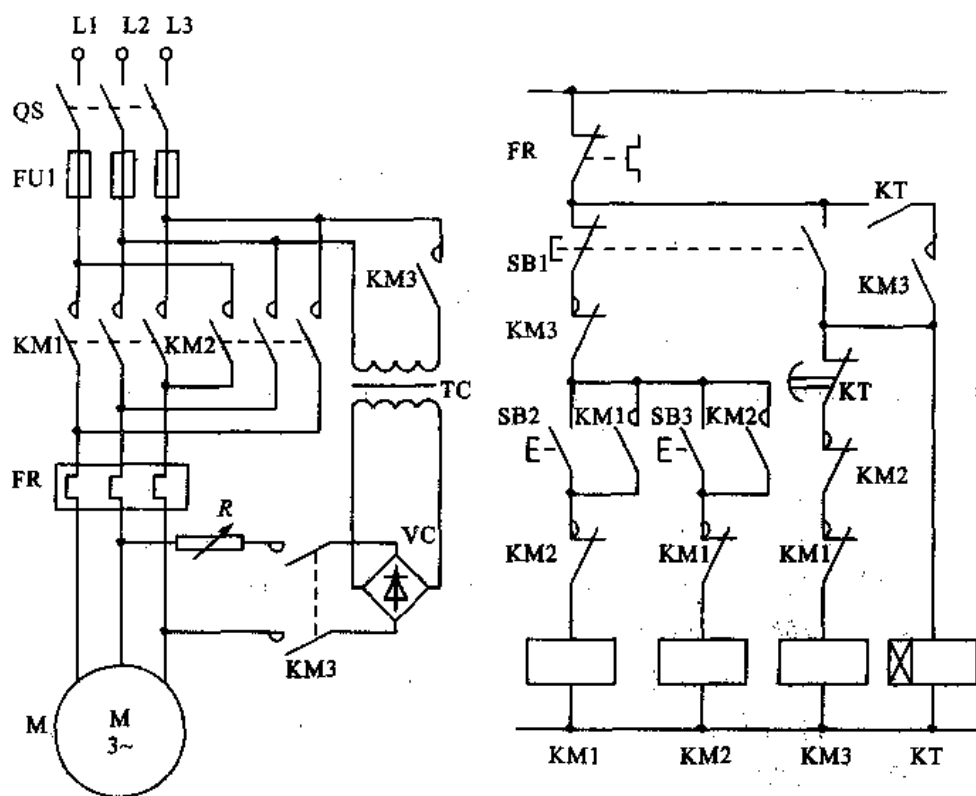


图 2-23 电动机可逆运行的能耗制动控制线路

动目的。该线路读者可自行分析,这里不再详细介绍。

按时间原则控制的能耗制动,一般适用于负载转速比较稳定的生产机械上。对于那些能够通过传动系统来实现负载速度变换或者加工零件经常变动的生产机械来说,采用速度原则控制

的能耗制动则较为合适。

能耗制动比反接制动消耗的能量少,其制动电流也比反接制动电流小得多,但能耗制动的制动效果不及反接制动明显,同时还需要一个直流电源,控制线路相对比较复杂,一般适用于电动机容量较大和启动、制动频繁的场所。

## 2.5 三相笼型异步电动机速度控制线路

在很多领域中,要求三相笼型异步电动机的速度为无级调节,其目的是实现自动控制、节能,以提高产品质量和生产效率。如钢铁行业的轧钢机、鼓风机,机床行业中的车床、机械加工中心等,都要求三相笼型异步电动机可调速。从广义上讲,电动机调速可分为两大类,即定速电动机与变速联轴器配合的调速方式和自身可调速的电动机。前者一般都采用机械式或油压式变速器,电气式只有一种即电磁转差离合器。其缺点是调速范围小和效率低。后者为电动机直接调速,其调速方法很多,如变更定子绕组极对数的变极调速和变频调速方式。变极调速控制最简单,价格便宜但不能实现无级调速。变频调速控制最复杂,但性能最好,随着其成本日益降低,目前已广泛应用于工业自动控制领域中。

### 2.5.1 基本概念

三相笼型异步电动机的转速公式为

$$n = n_0(1-s) = \frac{60f_1(1-s)}{p} \quad (2-1)$$

式中: $n_0$ 为电动机同步转速, $p$ 为极对数, $s$ 为转差率, $f_1$ 为供电电源频率。

对三相笼型异步电动机来讲,调速的方法有三种:改变极对数 $p$ 的变极调速、改变转差率 $s$ 的降压调速和改变电动机供电电源频率 $f_1$ 的变频调速。下面主要介绍变极调速和变频调速两种,其他调速方法可参考交流调速有关书籍。

### 2.5.2 变极调速控制线路

变极调速这一线路的设计思想是通过接触器触点改变电机绕组的接线方式来达到调速目的的。

变极电动机一般有双速、三速、四速之分,双速电动机定子装有一套绕组,而三速、四速则为两套绕组。

电动机变极采用电流反向法。下面以电动机一单相绕组为例来说明变极原理。如图2-24(a)表示极数等于4( $p=2$ )时的一相绕组的展开图,绕组由相同的两部分串联而成,两部分各称为半相绕组,一个半相绕组的末端X1与另一个半相绕组的首端A2相连接。图2-24(b)为绕组的并联连接方式展开图,则磁极数目减少一半,由4极变成2( $p=1$ )极。从图2-24(a)、(b)可以看出,串联时两个半相绕组的电流方向相同,都是从首端进、末端出;改成并联后,两个半相绕组的电流方向相反。当一个半相绕组的电流从首端进、末端出时,另一个半相绕组的电流便从末端进、首端出。因此改变磁极数目是将半相绕组的电流反向来实现的。

图2-24(c)和(d)为双速电动机三相绕组连接图。图(c)为三角形(四极,低速)与双星形(二极,高速)接法;图2-24(d)所示为星形(四极,低速)与双星形(二极,高速)接法。

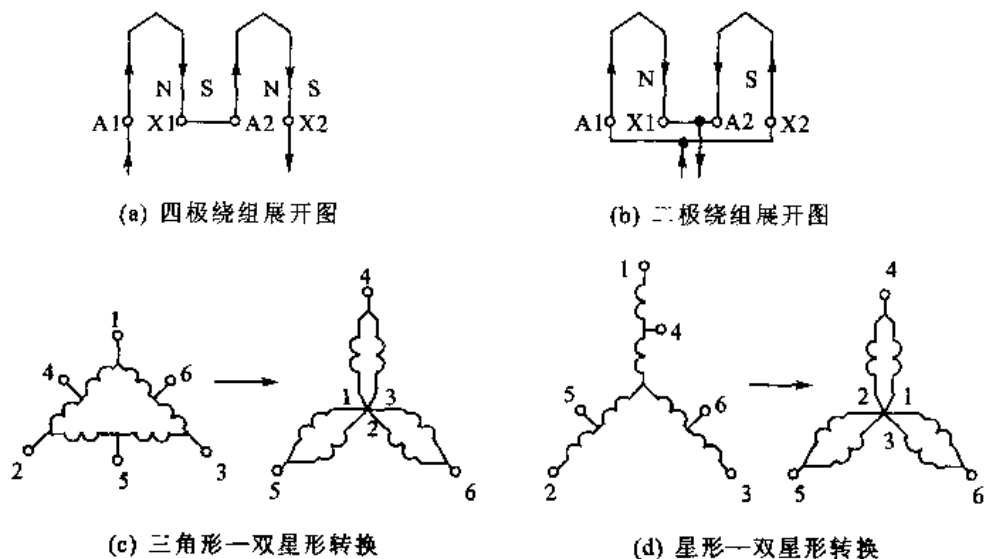


图 2-24 双速电动机改变极对数的原理

若低速运行时,电动机三相绕组端子的 1、2、3 端分别接入三相电源的 A、B、C。在高速运行时,4、5、6 端分别接入三相电源的 A、B、C。这会使电动机因变极而改变旋转方向,因此变极后必须改变绕组的相序。因为各相绕组在空间相差的机械角度是固定不变的,电角度则随磁极数目改变而改变。例如磁极数目减少一半,使各相绕组在空间相差的电角度增加一倍,原来相差  $120^\circ$  电角度的绕组,现在相差  $240^\circ$ 。如果相序不变,气隙磁场就要反转。

双速电动机调速控制线路如图 2-25 所示。图中接触器 KM1 工作时,电动机为低速运行;

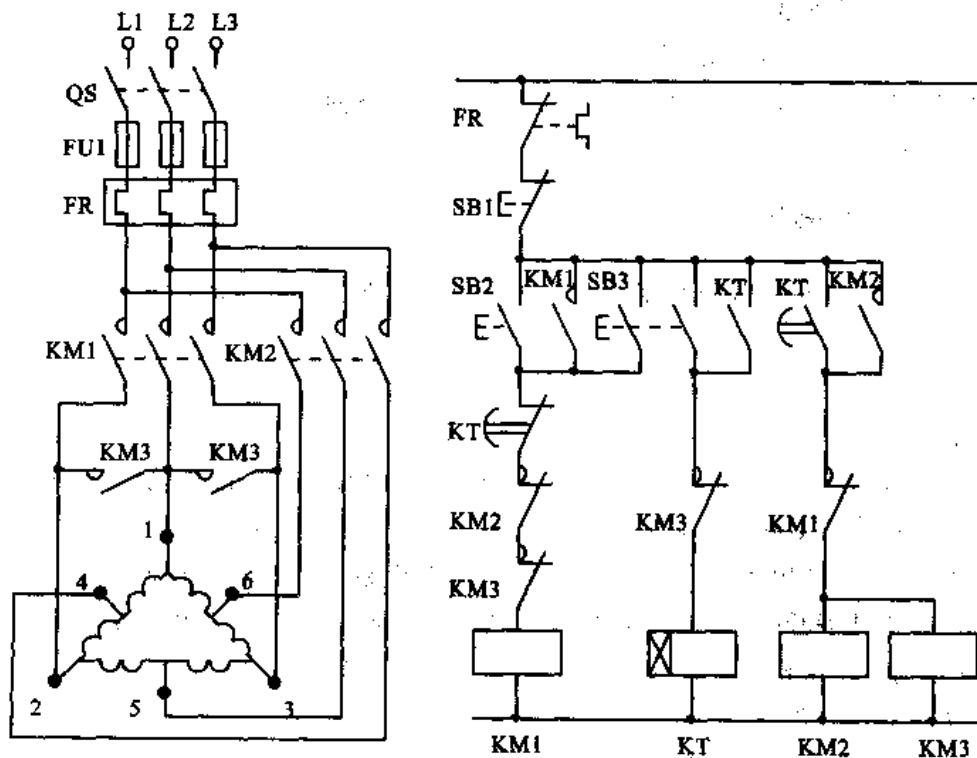


图 2-25 双速电动机调速控制线路

接触器 KM2、KM3 工作时,电动机为高速运行,注意变换后相序已改变。SB2、SB3 分别为低速和高速启动按钮。按低速按钮 SB2,接触器 KM1 通电并自锁,电动机接成三角形,低速运转;若按高速启动按钮 SB3 则直接启动,接触器首先使 KM1 通电自锁,时间继电器 KT 线圈通电自锁,电动机则先低速运转,当 KT 延时时间到,时间继电器 KT 首先延时并打开常闭触点,切断接触器 KM1 线圈电源,然后接触器 KM2、KM3 线圈通电自锁,KM3 的通电使时间继电器 KT 线圈断电,故自动切换使 KM2、KM3 工作,电动机高速运转,这样先低速后高速的控制,目的是限制启动电流。

双速电动机调速的优点是可以适应不同负载性质的要求。如需要恒功率时可采用三角形—双星形接法;如需要恒转矩调速时用星形—双星形接法。双速电机调速线路简单、维修方便;缺点是有级调速。变极调速通常要与机械变速配合使用,以扩大其调速范围。

### 2.5.3 变频调速与变频器的使用

近十多年来,随着控制技术和电力电子技术的发展,变频器的使用越来越广泛。一是由于变频调速的性能好,二是变频器的价格有了大幅度的降低。

变频调速的特点有:可以使用标准电机如不需维护的笼型电动机,可以连续调速,改变转速方向可通过电子回路改变相序实现。其优点是启动电流小,加减速速度可调节,电机可以高速化和小型化,防爆容易,保护功能齐全(如过载保护、短路保护、过电压和欠电压保护)等。

变频调速的应用领域非常广泛。它应用于风机、泵、搅拌机、挤压机、精纺机和压缩机,原因是节能效果显著。它应用于机床如车床、机械加工中心、钻床、铣床、磨床,主要目的是提高生产率和质量;它也广泛应用于其他领域,如各种传送带的多台电机同步、调速和起重机械等。

#### 1. 变频调速的基本概念

由式(2-1)可知,改变电源频率  $f_1$  就可改变电动机同步转速。异步电动机采用变频器进行调速控制时,为了避免电动机磁饱和,要控制电动机的磁通,同时抑制启动电流,这就需要根据电动机的特性对供电电压、电流、频率进行适当的控制,使电动机产生必需的转矩。

变频器的控制方式可分为两种,即开环控制和闭环控制。开环控制有 V/F 控制方式,闭环控制有矢量控制等方式。

##### (1) V/F 控制

异步电动机的转速由电源频率和极对数决定,所以改变频率,电动机就可以调速运转。但是频率改变时电动机内部阻抗也改变。仅改变频率,将会产生由弱励磁引起的转矩不足或由过励磁引起的磁饱和现象,使电动机功率因数和效率显著下降。

V/F 控制是这样一种控制方式,即改变频率的同时控制变频器输出电压,使电动机的磁通保持一定,在较广泛的范围内调速运转时,电动机的功率因数和效率不下降。这就是控制电压与频率之比,所以称为 V/F 控制。

作为变频器调速控制方式,V/F 控制比较简单,现多用于通用变频器。如风机和泵类机械的节能运行、生产流水线的传送控制和空调等家用电器中。

##### (2) 矢量控制

我们知道直流电动机的电枢电流控制方式,使直流电动机构成的传动系统的调速和控制性能非常优良。矢量控制按照直流电动机电枢电流控制思想,在交流异步电动机上实现该控制方法,并且达到与直流电动机具有相同的控制性能。

矢量控制是这样的一种控制方式,即将供给异步电动机的定子电流在理论上分成两部分:

产生磁场的电流分量(磁场电流)和与磁场相垂直、产生转矩的电流分量(转矩电流)。该磁场电流、转矩电流与直流电动机的磁场电流、电枢电流相当。在直流电动机中,利用整流子和电刷机械换向,使两者保持垂直,并且可分别供电。对异步电动机来讲,其定子电流在电动机内部,利用电磁感应作用,可在电气上分解为磁场电流和垂直的转矩电流。

矢量控制就是根据上述原理,将定子电流分解成磁场电流和转矩电流,任意进行控制。两者合成后,决定定子电流大小,然后供给异步电动机。

矢量控制方式使交流异步电动机具有与直流电动机相同的控制性能。目前采用这种控制方式的变频器已广泛应用于生产实际中。

## 2. 各种控制方式的变频器特性比较

### (1) V/F 控制变频器特点

- ① 它是最简单的一种控制方式,不用选择电动机,通用性优良。
- ② 与其他控制方式相比,在低速区内电压调整困难,故调速范围窄,通常在 1:10 左右的调速范围内使用。

- ③ 急加速、减速或负载过大时,抑制过电流能力有限。
- ④ 不能精密控制电动机实际速度,不适合于同步运转场合。

### (2) 矢量控制变频器特点

- ① 需要使用电动机参数,一般用做专用变频器。
- ② 调速范围在 1:100 以上。
- ③ 速度响应性极高,适合于急加速、减速运转和连续 4 象限运转,能适用任何场合。

## 3. 变频器的操作和显示

一台变频器应有可供用户方便操作的操作器和显示变频器运行状况及参数设定的显示器。用户通过操作器对变频器进行设定及运行方式的控制。通用变频器的操作方式一般有三种,即数字操作器、远程操作器和端子操作方式。变频器的操作指令可以由此三处发出。

### (1) 数字操作器和数字显示器

新型变频器几乎均采用数字控制,使数字操作器可以对变频器进行设定操作。如设定电动机的运行频率、运转方式、V/F 类型、加减速时间等。数字操作器有若干个操作键,不同厂商生产的变频器的操作器有很大的区别,但 4 个按键是必不可少的,即运行键、停止键、上升键和下降键。运行键控制电动机的启动,停止键控制电动机的停止,上升或下降键可以检索设定功能及改变功能的设定值。数字操作器作为人机对话接口,使得变频器参数设定与显示直观清晰,操作简单方便。

在数字操作器上,通常配有 6 位或 4 位数字显示器,它可以显示变频器的功能代码及各功能代码的设定值。在变频器运行前显示变频器的设定值,在运行过程中显示电动机的某一参数的运行状态,如电流、频率、转速等。

### (2) 远程操作器

远程操作器是一个独立的操作单元,它利用计算机的串行通信功能,不仅可以完成数字操作器所具有的操作功能,而且可以实现数字操作器不能实现的一些功能。特别是在系统调试时,利用远程操作器可以对各种参数进行监视和调整,比数字操作器功能强,而且更方便。

变频器的日益普及,使用场地相对分散,远距离集中控制是变频器应用的趋势,现在的变频器一般都具有标准的通信接口,用户可以利用通信接口在远处如中央控制室对变频器进行

集中控制,如参数设定、启动/停止控制、速度设定和状态读取等。

### (3) 端子操作

变频器的端子包括电源接线端子和控制端子两大类。电源接线端子包括三相电源输入端子,三相电源输出端子,直流侧外接制动电阻用端子以及接地端子。控制端子包括频率指令模拟设定端子,运行控制操作输入端子、报警端子、监视端子。不同类型的变频器端子的设置与排列有差别,但共同点很多。在后面的变频器应用中再详细介绍。

### 4. 变频器调速应用举例

目前实用化的变频器种类很多,下面以西门子 MICROMASTER 440 为例,简要说明变频器的使用。

MICROMASTER 440 是一种集多种功能于一体的变频器,它适用于电动机需要调速的各种场合。它可通过数字操作面板或通过远程操作器方式,修改其内置参数,即可工作于各种场合。主要特点是:内置多种运行控制方式;快速电流限制,实现无跳闸运行;内置式制动斩波器,实现直流注入制动;具有 PID 控制功能的闭环控制,控制器参数可自动整定;多组参数设定且可相互切换,变频器可用于控制多个交替工作的生产过程;多功能数字、模拟输入/输出,可任意定义其功能和具有完善的保护功能。

#### (1) 控制方式

变频器运行控制方式,即变频器输出电压与频率的控制关系。控制方式的选择,可通过变频器相应的参数设置选择。主要有以下 7 种控制方式。

① 线性 V/F 控制 变频器输出电压与频率为线性关系,用于恒定转矩负载。

② 带磁通电流控制(FCC)的线性 V/F 控制 在这种模式下,变频器根据电动机特性实时计算所需要的输出电压,以此来保持电动机的磁通处于最佳状态。此方式可提高电动机效率和改善电动机动态响应特性。

③ 平方 V/F 控制 变频器输出电压平方与频率为线性关系,用于变转矩负载,如风机和泵。

④ 特性曲线可编程的 V/F 控制 变频器输出电压与频率为分段线性关系,此种控制方式可应用于在某一特定频率下为电动机提供特定的转矩。

⑤ 带“能量优化控制(ECO)”的线性 V/F 控制 此方式的特点是变频器自动增加或降低电动机电压,搜寻并使电动机运行在损耗最小的工作点。

⑥ 无传感器矢量控制 用固有的滑差补偿对电机的速度进行控制。采用这一控制方式时,可以得到大的转矩、改善瞬态响应特性和具有优良的速度稳定性,而且在低频时可提高电动机的转矩。

⑦ 无传感器的矢量转矩控制 变频器可以直接控制电动机的转矩。当负载要求具有恒定的转矩时,变频器通过改变向电动机输出的电流,使转矩维持在设定的数值。另外,还有与纺织机械相关的 V/F 控制方式。

#### (2) 保护特性

过电压及欠电压保护、变频器过热保护、接地故障保护、短路保护、电动机过载保护和用 PTC 为传感器的电动机过热保护等。

#### (3) 变频器功能的方框图

如图 2-26 所示为变频器内部功能方框图。此变频器共有 20 多个控制端子,分为 4 类:

输入信号端子、频率模拟设定输入端子、监视信号输出端子和通信端子。

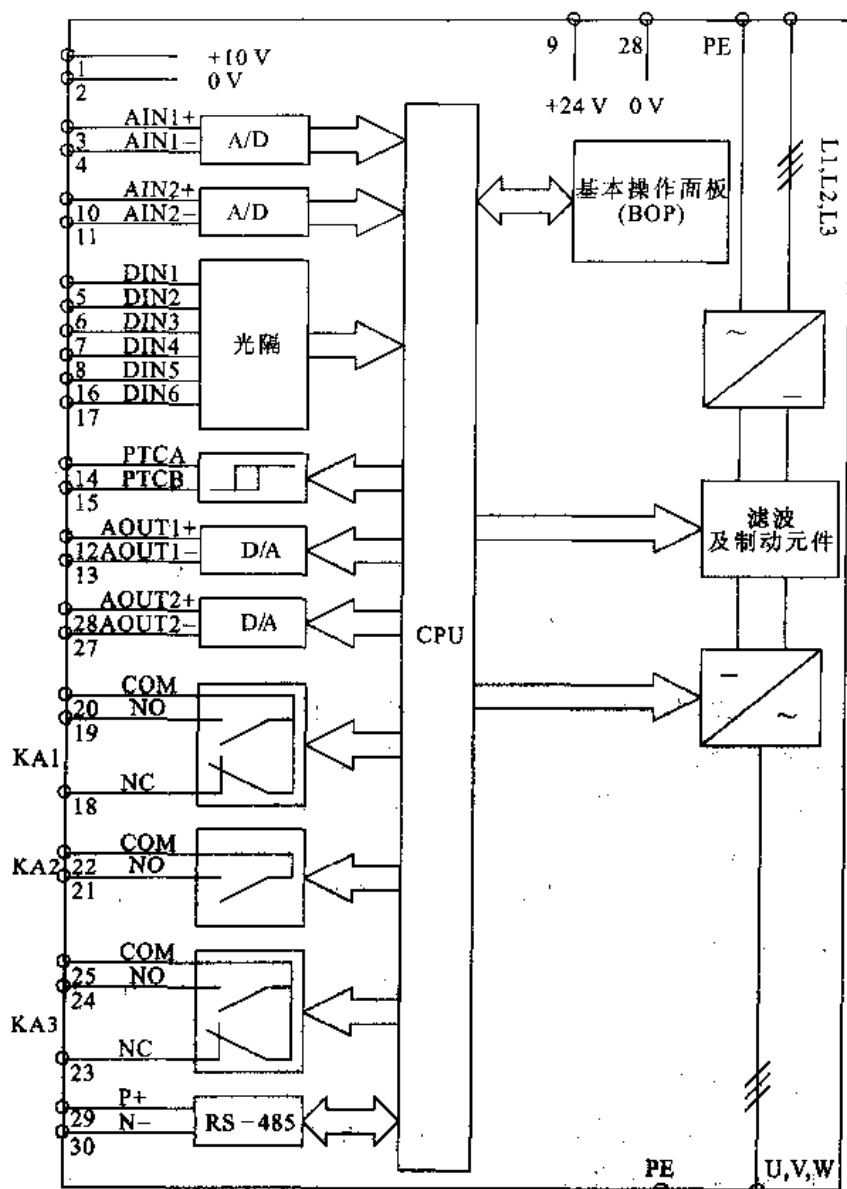


图 2-26 变频器内部功能方框图

DIN1~DIN6 为数字输入端子，一般用于变频器外部控制，其具体功能由相应设置决定。例如出厂时设置 DIN1 为正向运行、DIN2 为反向运行等，根据需要可通过修改参数可改变功能。使用输入信号端子可以完成对电动机的正反转控制、复位、多级速度设定、自由停车、点动等控制操作。PTC 端子用于电动机内置 PTC 测温保护，为 PTC 传感器输入端。

AIN1、AIN2 为模拟信号输入端子，分别作为频率给定信号和闭环时反馈信号输入。变频器提供了 3 种频率模拟设定方式：外接电位器设定、0~10 V 电压设定和 4~20 mA 电流设定。当用电压或电流设定时，最大的电压或电流对应变频器输出频率设定的最大值。变频器有两路频率设定通道，开环控制时只用 AIN1 通道，闭环控制时使用 AIN2 通道作为反馈输入，两路模拟设定进行叠加。

输出信号的作用是对变频器运行状态的指示，或向上位机提供这些信息。KA1、KA2、

KA3 为继电器输出,其功能也是可编程的,如故障报警、状态指示等。AOUT1、AOUT2 端子为模拟量输出  $0\sim 20\text{ mA}$  信号,其功能也是可编程的,用于输出指示运行频率、电流等。

P+、N- 为通信接口端子,是一个标准的 RS-485 接口。通过此通信接口,可以实现对变频器的远程控制,包括运行/停止及频率设定控制,也可以与端子控制进行组合完成对变频器的控制。

变频器可使用数字操作面板控制,也可使用端子控制,也可使用 RS-485 通信接口对其远程控制。

#### (4) 应用举例

如图 2-27 所示为使用变频器举例。此线路实现电动机的正反向运行、调速和点动功能。根据功能要求,首先要对变频器编程并修改参数。根据控制要求选择合适的运行方式,如线性 V/F 控制、无传感器矢量控制等;频率设定值信号源选择模拟输入。选择控制端子的功能,将变频器 DIN1、DIN2、DIN3 和 DIN4 端子分别设置为正转运行、反转运行、正向点动和反向点动功能。除此以外还要设置如斜坡上升时间、斜坡下降时间等参数,更详细的参数设定方法可参见变频器的使用手册。

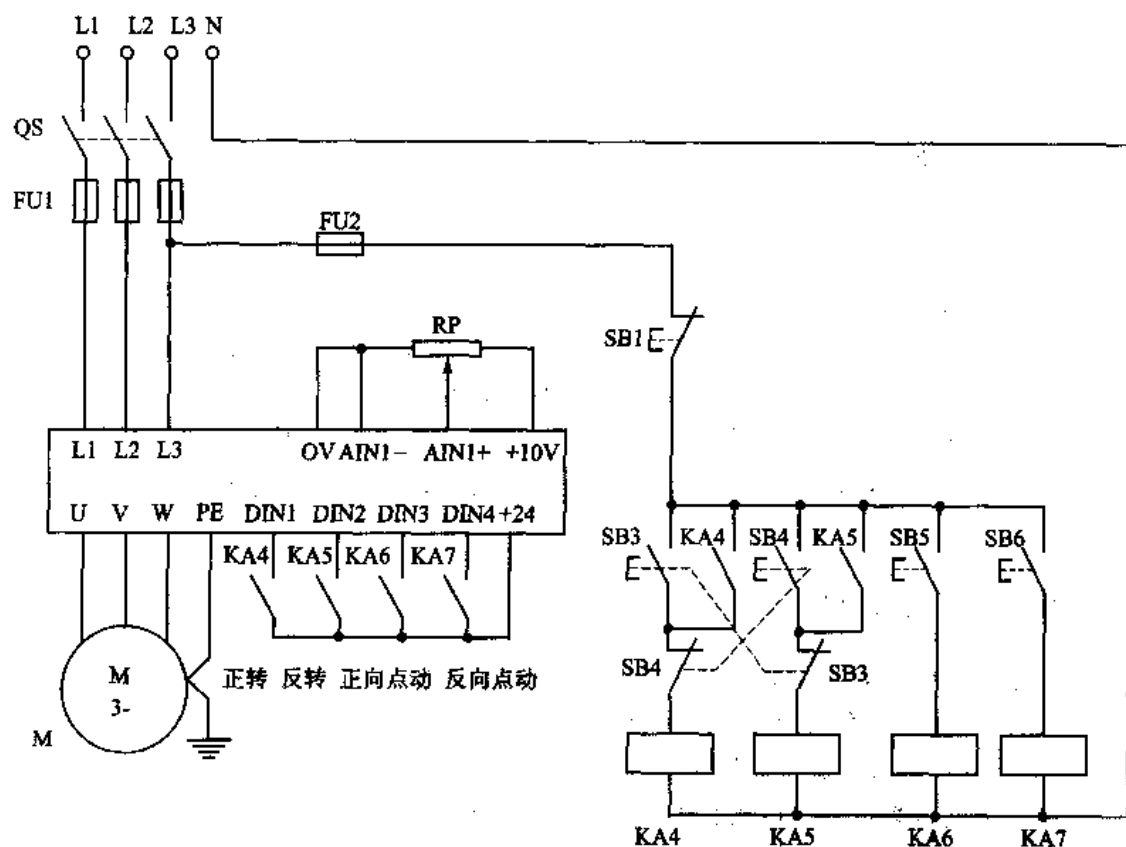


图 2-27 使用变频器的异步电动机可逆调速系统控制线路

在图 2-27 中,SB3、SB4 为正、反向运行控制按钮,运行频率由电位器 RP 给定。SB5、SB6 为正、反向点动运行控制按钮,点动运行频率可由变频器内部设置。按钮 SB1 为总停止控制。

## 2.6 电气控制线路的简单设计法

### 2.6.1 概 述

电气控制系统的设计一般包括确定拖动方案、选择电机容量和设计电气控制线路。电气控制线路的设计又分为主电路设计和控制电路设计。一般情况下我们所说的电气控制线路设计主要指的是控制电路的设计。

电气控制线路的设计通常有两种方法,即一般设计法和逻辑设计法。

一般设计法又称为经验设计法。它主要是根据生产工艺要求,利用各种典型的线路环节,直接设计控制电路。这种方法比较简单,但要求设计人员必须熟悉大量的控制线路,掌握多种典型线路的设计资料,同时具有丰富的经验,在设计过程中往往还要经过多次反复的修改、试验,才能使线路符合设计的要求。即使这样,设计出来的线路可能还不是最简,所用的电气触点不一定最少,所得出的方案也不一定是最佳方案。

逻辑设计法是根据生产工艺的要求,利用逻辑代数来分析、设计控制线路。用这种方法设计出来的线路比较合理,特别适合完成较复杂的生产工艺所要求的控制线路设计。但是相对而言,逻辑设计法难度较大,不易掌握,所设计出来的电路不太直观。

随着 PLC 的出现和 PLC 技术的飞速发展,其功能越来越强大,价格也越来越低。在电气控制技术领域,PLC 基本上全面取代了继电接触式控制系统,所以对传统的电气控制线路的设计方法也要适当地改进。这主要依据下面两点:

首先,对于简单的电气控制线路,考虑到成本问题,还要使用继电器组成控制系统,所以还是要进行电气控制线路设计的。其次,对于稍微复杂的电气控制线路,就要用 PLC 而不会再用继电器控制系统了,所以逻辑设计法使用得越来越少了。

基于上面的考虑,对于电气控制线路的设计和学习,我们想把一般设计法的简单和逻辑设计法的严谨结合起来,归纳出一种简单设计法。使用简单设计法可以完成大多数电气控制电路的设计。

### 2.6.2 简单设计法介绍

简单设计法遵从一般设计法的主要设计原则,利用逻辑设计法中继电器开关逻辑函数,把控制对象的启动信号、关断信号及约束条件找出,即可设计出控制电路。

#### 1. 一般设计法的几个主要原则

(1) 最大限度地满足生产机械和工艺对电气控制线路的要求。

(2) 在满足生产要求的前提下,控制线路力求简单、经济、安全可靠。应做到以下几点。

① 尽量减少电器的数量 尽量选用相同型号的电器和标准件,以减少备品量;尽量选用标准的、常用的或经过实际考验过的线路和环节。

② 尽量减少控制线路中电源的种类 尽可能直接采用电网电压,以省去控制变压器。

③ 尽量缩短连接导线的长度和数量 设计控制线路时,应考虑各个元件之间的实际接线。如图 2-28 所示,图(a)接线是不合理的,因为按钮在操作台或面板上,而接触器在电气柜内,这样接线就需要由电气柜二次引出接到操作台的按钮上。改为图(b)后,可减少一些引出线。

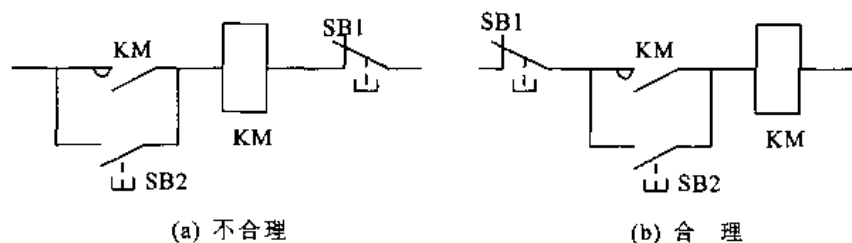


图 2-28 电器连接图

④ 正确连接触点 在控制电路中,应尽量将所有触点接在线圈的左端或上端,而线圈的右端或下端直接接到电源的另一根母线上(左右端和上下端是针对控制电路水平绘制或垂直绘制而言的)。这样可以减少线路内产生虚假回路的可能性,还可以简化电气柜的出线。

⑤ 正确连接电器的线圈 在交流控制电路中不能串联两个电器的线圈,如图 2-29(a)所示。因为每一个线圈上所分到的电压与线圈阻抗成正比,两个电器动作总是有先有后,不可能同时吸合。例如交流接触器 KM2 吸合,由于 KM2 的磁路闭合,线圈的电感显著增加,因而在该线圈上的电压降也显著增大,从而使另一接触器 KM1 的线圈电压达不到动作电压。因此两个电器需要同时动作时,其线圈应该并联起来,如图 2-29(b)所示。

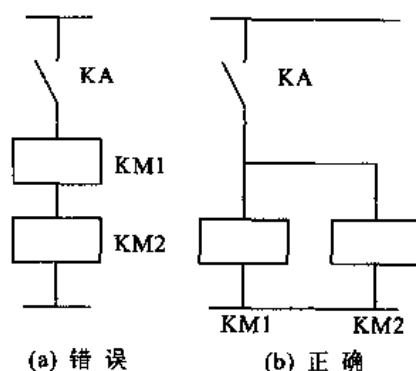


图 2-29 线圈的连接

⑥ 元器件的连接 应尽量减少多个元件依次通电后才接通另一个电器元件的情况。在图 2-30(a)中,线圈 KA3 的接通要经过 KA、KA1、KA2 三个常开触点。改接成图 2-30(b)后,则每一对线圈通电只需要经过一对常开触点,工作较可靠。

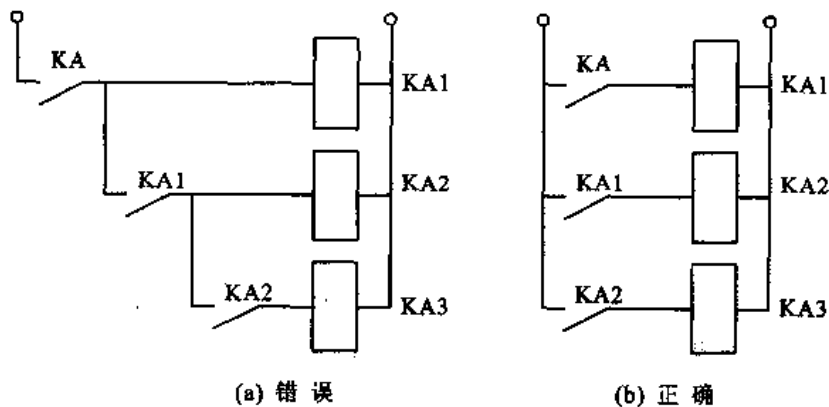


图 2-30 元器件的连接

⑦ 要注意电器之间的联锁和其他安全保护环节 在实际工作中,一般设计法还有许多要注意的地方,本书不再详细介绍。

## 2. 逻辑设计法中的继电器开关逻辑函数

我们知道逻辑设计法主要依据逻辑代数运算法则的化简办法来求出控制对象的逻辑方程,然后由逻辑方程画出电气控制原理图。其中继电器的开关逻辑函数以执行元件作为逻辑函数的输出变量,而以检测信号中间单元及输出逻辑变量的反馈触点作为逻辑变量,按一定规

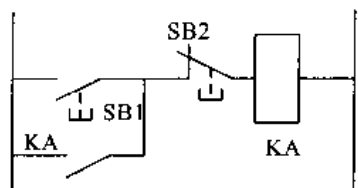


图 2-31 继电器开关逻辑函数

律列出其逻辑函数表达式。继电器开关逻辑函数是电气控制对象的典型代表。图 2-31 所示为它的开关逻辑函数（启—保—停电路）。

线路中 SB1 为启动信号按钮，SB2 为关断信号按钮，KA 的常开触点为自保持信号。它的逻辑函数为

$$F_{KA} = (SB1 + KA) \cdot \overline{SB2} \quad (2-2)$$

若把 KA 替换成一般控制对象 K，启动/关断信号换成一般形式 X，则式(2-2)的开关逻辑函数的一般形式为

$$F_K = (X_{开} + K) \cdot \overline{X_{关}} \quad (2-3)$$

扩展到一般控制对象：

$X_{开}$  为控制对象的开启信号，应选取在开启边界线上发生状态改变的逻辑变量； $X_{关}$  为控制对象的关断信号，应选取在控制对象关闭边界线上发生状态改变的逻辑变量。在线路图中使用的触点 K 为输出对象本身的常开触点，属于控制对象的内部反馈逻辑变量，起自锁作用，以维持控制对象得电后的吸合状态。

$X_{开}$  和  $X_{关}$  一般要选短信号，这样可以有效防止启/停信号波动的影响，保证了系统的可靠性。波形如图 2-32 所示。

在某些实际应用中，为进一步增加系统的可靠性和安全性， $X_{开}$  和  $X_{关}$  往往带有约束条件，如图 2-33 所示。

其逻辑函数为

$$F_K = (X_{开} \cdot X_{开约} + K) \cdot (\overline{X_{关}} + \overline{X_{关约}}) \quad (2-4)$$

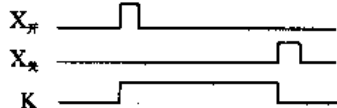


图 2-32 典型开关逻辑函数波形

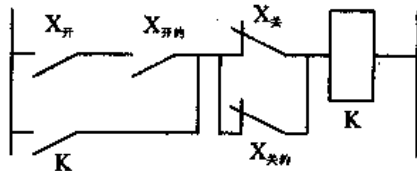


图 2-33 带约束条件的控制对象开关逻辑电路

式(2-4)基本上全面代表了控制对象的输出逻辑函数。由式(2-4)可以看出，对开启信号来说，开启的主令信号不止一个，还需要具备其他条件才能开启；对关断信号来说，关断的主令信号也不止一个，还需要具备其他的关断条件才能关断。这样就增加了系统的可靠性和安全性。当然  $X_{开约}$  和  $X_{关约}$  也不一定同时存在，有时也可能  $X_{开约}$  或  $X_{关约}$  不止一个，关键是要具体问题具体分析。

### 3. 简单设计法

把一般设计法中的重要设计原则和逻辑设计法中控制对象的开关逻辑函数结合起来使用就组成了简单设计法。简单设计法要求在设计控制线路时做到以下几点：

- (1) 找出控制对象的开启信号、关断信号；
- (2) 如果有约束条件，则找出相应的开启约束条件和关断约束条件；
- (3) 把各种已知信号带入式(2-4)中，写出控制对象的逻辑函数(熟练后可省去该步)；
- (4) 结合逻辑函数和一般设计法的设计原则，画出该控制对象的电气线路图；

(5) 最后根据工艺要求做进一步的检查工作,如加上保护和互锁环节等。

由此可以看出,简单设计法的核心内容是要找出控制对象的开启条件(短信号)和关断条件(短信号),然后所有的设计问题就很简单了。当然一些控制对象的开启条件和关断条件的短信号不容易找出来,这时就要采取一些其他技巧和措施配合使用才能解决问题。

需要指出的是,简单设计法设计出来的电路不一定是合理的。但稍复杂的电路已经被 PLC 取代了,所以现在对简单的电路进行最优化设计已不是最主要的问题,重要的是要理解电气控制线路设计的实质,力求用简单的方法对简单电控系统设计出较优质的电路。

### 2.6.3 简单设计法设计举例

**题目** 现有三台电动机 M1、M2、M3,要求启动顺序为:先启动 M1,经 T1 后启动 M2,再经 T2 后启动 M3;停车时要求:先停 M3,经 T3 后再停 M2,再经 T4 后停 M1。三台电机使用的接触器分别为 KM1、KM2 和 KM3。试设计该三台电动机的启/停控制线路。

**题目分析** 该系统有一个启动按钮(SB1)和一个停止按钮(SB2),另外要用四个时间继电器 KT1、KT2、KT3 和 KT4。其定时值依次为 T1、T2、T3 和 T4。工作顺序如图 2-34 所示。

**解题分析** 从图 2-34 中可以看出:M1 的启动信号为 SB1,停止信号为 KT4 计时到;M2 的启动信号为 KT1 计时到,停止信号为 KT3 计时到;M3 的启动信号为 KT2 计时到,停止信号为 SB2。

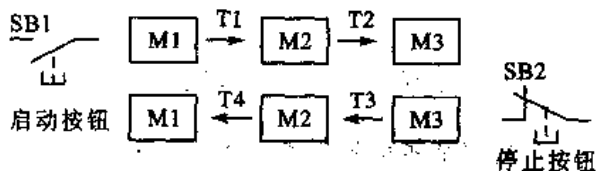


图 2-34 三台电机工作顺序

在设计时,考虑到启/停信号要用短信号,所以要注意对定时器及时复位。

该系统的电气控制线路原理如图 2-35 所示。

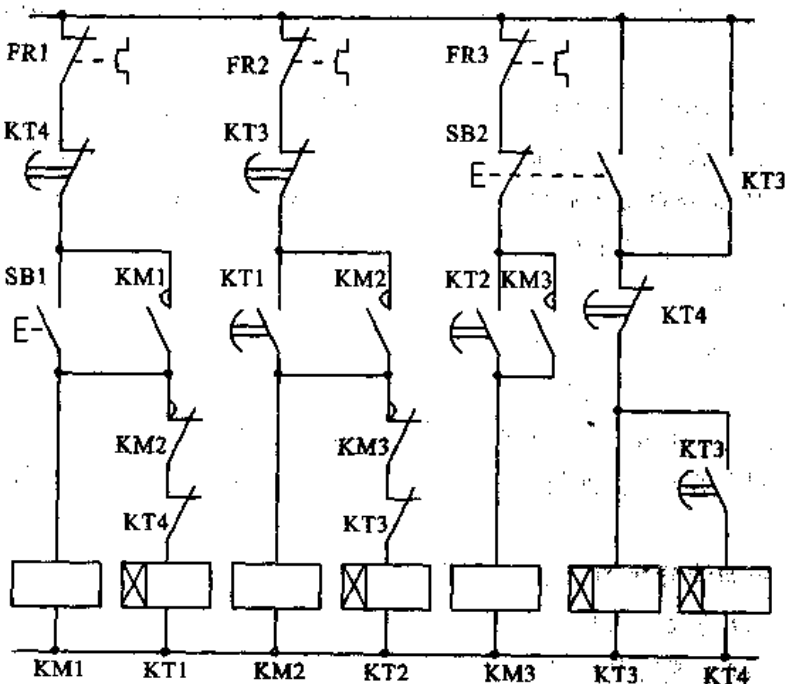


图 2-35 三台电机顺序启/停控制电路原理图

图 2-35 中的 KT1、KT2 线圈上方串联了接触器 KM2 和 KM3 的常闭触点,这是为了得到启动短信号而采取的措施;而 KT2、KT1 线圈上的常闭触点 KT3 和 KT4 的作用是为了防止 KM3 和 KM2 断电后,KT2 和 KT1 的线圈重新得电而采取的措施。因为若  $T_2 < T_3$  或  $T_1 < T_4$  时,有可能造成 KM3 和 KM2 重新启动。设计中的难点是找出 KT3、KT4 开始工作的条件,以及 KT1、KT2 的逻辑。本例中没有考虑时间继电器触点的数量是否够用的问题,实际选型时必须考虑这一点。

FR1~FR3 分别为三台电机的热继电器常闭触点,它是为了防止过载而采取的措施。若对过载没有太多要求,则可把它们去掉。

## 2.7 典型生产机械电气控制线路分析

在现代生产机械设备中,电气控制系统是重要的组成部分,本节将通过分析典型生产机械 C650 车床的电气控制线路,进一步介绍电气控制线路的组成以及各种基本控制线路在具体系统中的应用。同时,掌握分析电气控制线路的方法,从中找出规律,逐步提高阅读电气控制线路图的能力。

### 2.7.1 电气控制线路分析基础

#### 1. 电气控制线路分析的内容与要求

分析电气控制线路的具体内容和要求主要包括以下几个方面。

##### (1) 设备说明书

设备说明书由机械(包括液压部分)与电气两部分组成。在分析时首先要阅读这两部分说明书,了解以下内容:

① 设备的结构组成及工作原理、设备传动系统的类型及驱动方式、主要技术性能、规格和运动要求等。

② 电气传动方式,电机、执行电器的数目、规格型号、安装位置、用途及控制要求。

③ 设备的使用方法,各操作手柄、开关、旋钮、指示装置的布置及其在控制线路中的作用。

④ 与机械、液压部分直接关联的电器(行程开关、电磁阀、电磁离合器、传感器等)的位置、工作状态及其与机械、液压部分的关系,在控制中的作用等。

##### (2) 电气控制原理图

电气控制原理图是控制线路分析的中心内容。它一般由主电路、控制电路、辅助电路、保护及联锁环节,以及特殊控制电路等部分组成。

在分析电气原理图时,必须与阅读其他技术资料结合起来。例如,各种电动机及执行元器件的控制方式、位置及作用,各种与机械有关的位置开关、主令电器的状态等,只有通过阅读说明书才能了解。

#### 2. 电气原理图阅读分析的方法与步骤

在掌握了机械设备及电气控制系统的构成、运动方式、相互关系,以及掌握各电动机和执行电器的用途和控制方式等基本条件之后,即可对设备控制线路进行具体的分析。分析电气原理图的一般原则是:化整为零、顺藤摸瓜、先主后辅、集零为整、安全保护和全面检查。

通常,分析电气控制系统时,要结合有关技术资料将控制线路“化整为零”,即以某一电动

机或电器元件(如接触器或继电器线圈)为对象,从电源开始,自上而下,自左而右,逐一分析其接通及断开的关系(逻辑条件),并区分出主令信号、联锁条件和保护要求等。根据图区坐标标注的检索可以方便地分析出各控制条件与输出的因果关系。

电气原理图的分析方法与步骤如下:

① 分析主电路 无论线路设计还是线路分析都应从主电路入手,而主电路的作用是保证整机拖动要求的实现。从主电路的构成可分析出电动机或执行电器的类型、工作方式、启动、转向、调速和制动等基本控制要求。

② 分析控制电路 主电路的控制要求是由控制电路来实现的。运用“化整为零”、“顺藤摸瓜”的原则,将控制线路按功能不同划分成若干个局部控制线路,从电源和主令信号开始,经过逻辑判断,写出控制过程。如果控制线路较复杂,则可先排除照明、显示等与控制关系不密切的电路,以便集中精力进行分析。

③ 分析辅助电路 辅助电路包括执行元件的工作状态显示、电源显示、参数测定、照明和故障报警等部分。辅助电路中很多部分是由控制电路中的元件来控制的,所以在分析辅助电路时,还要回过头来对照控制电路进行分析。

④ 分析联锁与保护环节 生产机械对安全性和可靠性有很高的要求,实现这些要求,除了合理地选择拖动、控制方案以外,在控制线路中还应设置一系列电气保护装置和必要的电气联锁。在电气控制原理图的分析过程中,电气联锁与电气保护环节是一个重要内容,不能遗漏。

⑤ 分析特殊控制环节 在某些控制线路中,还设置了一些与主电路、控制电路关系不密切,且相对独立的某些特殊环节。如产品计数装置、自动检测系统、晶闸管触发电路和自动调温装置等。这些部分往往自成一个子系统,其读图和分析方法可参照上述分析过程,灵活运用所学过的电子技术、变流技术、自控系统、检测与转换等知识逐一分析。

⑥ 总体检查 经过“化整为零”,逐步分析了每一局部电路的工作原理以及各部分之间的控制关系之后,还必须用“集零为整”的方法,检查整个控制线路,看是否有遗漏。特别要从整体角度去进一步检查和理解各控制环节之间的联系,以达到清楚地理解原理图中每一个电气元器件的作用、工作过程及主要参数。

### 2.7.2 C650 卧式车床电气控制线路分析

卧式车床是一种应用极为广泛的金属切削加工机床,主要用来加工各种回转表面、螺纹和端面,并可通过尾架进行钻孔、铰孔和攻螺纹等切削加工。

卧式车床通常由一台主电动机拖动,经由机械传动链,实现切削主运动和刀具进给运动的输出,其运动速度由变速齿轮箱通过手柄操作进行切换。刀具的快速移动、冷却泵和液压泵等常采用单独的电动机驱动。不同型号的卧式车床,其主电动机的工作要求不同,因而具有不同的控制线路。下面以 C650 型卧式车床电气控制系统为例,进行电气控制线路分析。

#### 1. 机床的主要结构和运动形式

C650 卧式车床属于中型车床,可加工的最大工件回转直径为 1 020 mm,最大工件长度为 3 000 mm,机床的结构形式如图 2-36 所示。

C650 卧式车床主要由床身、主轴、刀架、溜板箱和尾架等部分组成。该车床有两种主要运动:一种是安装在床身主轴箱中的主轴转动,称为主运动;另一种是溜板箱中的溜板带动刀架的直线运动,称为进给运动。刀具安装在刀架上,与滑板一起随溜板箱沿主轴轴线方向实现进

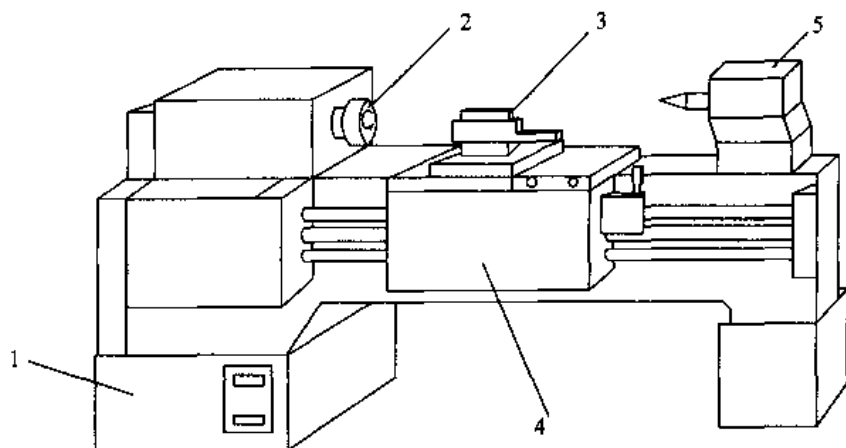


图 2-36 C650 卧式车床结构简图

1—床身；2—主轴；3—刀架；4—溜板箱；5—尾架

给移动，主轴的转动和溜板箱的移动均由主电机驱动。由于加工的工件比较大，加工时其转动惯量也比较大，需停车时不易立即停止转动，因此必须有停车制动的功能，较好的停车制动是采用电气制动方法。为了加工螺纹等工件，主轴需要正、反转，主轴的转速应随工件的材料、尺寸、工艺要求及刀具的种类不同而变化，所以要求在相当宽的范围内可进行速度调节。在加工过程中，还需提供切削液，并且为减轻工人的劳动强度和节省辅助工作时间，而要求带动刀架移动的溜板能够快速移动。

## 2. 电力拖动及控制要求

从车床的加工工艺出发，对拖动控制有以下要求：

① 主电动机 M1 完成主轴主运动和溜板箱进给运动的驱动，电动机采用直接启动的方式启动，可正反两个方向旋转，并可进行正反两个旋转方向的电气停车制动。为加工调整方便，还应具有点动功能。

② 电动机 M2 拖动冷却泵，在加工时提供切削液，采用直接启动及停止方式，并且为连续工作方式。

③ 主电动机和冷却泵电动机应具有必要的短路和过载保护。

④ 快速移动电动机 M3 拖动刀架快速移动，还可根据使用需要随时进行手动控制启停。

⑤ 应具有安全的局部照明装置。

## 3. 电气控制线路分析

C650 卧式车床的电气控制系统线路如图 2-37 所示，使用的电气元件符号与功能说明如表 2-2 所列。

表 2-2 电气元件符号及功能说明表

符 号	名 称 及 用 途	符 号	名 称 及 用 途
M1	主电动机	SB1	总停按钮
M2	冷却泵电动机	SB2	主电动机正向点动按钮
M3	快速移动电动机	SB3	主电动机正向启动按钮
KM1	主电动机正转接触器	SB4	主电动机反向启动按钮
KM2	主电动机反转接触器	SB5	冷却泵电动机停止按钮
KM3	短接限流电阻接触器	SB6	冷却泵电动机启动按钮
KM4	冷却泵电动机接触器	TC	控制变压器
KM5	快移电动机接触器	FU1~6	熔断器
KA	中间继电器	FR1	主电动机过载保护热继电器
KT	通电延时时间继电器	FR2	冷却泵电动机保护热继电器
SQ	快移电动机点动手柄开关	R	限流电阻
SA	照明灯开关	EL	照明灯
KS	速度继电器	TA	电流互感器
A	电流表	QS	隔离开关

### (1) 主电路分析

图 2-37 所示的主电路中有三台电动机,隔离开关 QS 将 380 V 的三相电源引入。电动机 M1 的电路接线分为三部分:第一部分由正转控制交流接触器 KM1 和反转控制交流接触器 KM2 的两组主触点构成电动机的正、反转接线;第二部分为电流表 A 经电流互感器 TA 接在主电动机 M1 的主回路上,以监视电动机绕组工作时的电流变化。为防止电流表被启动电流冲击损坏,利用时间继电器的延时动断触头(3 区),在启动的短时间内将电流表暂时短接掉;第三部分为串联电阻控制部分,交流接触器 KM3 的主触点(2 区)控制限流电阻 R(3 区)的接入和切除,在进行点动调整时,为防止连续的启动电流造成电动机过载,串入限流电阻 R,保证电路设备正常工作。速度继电器 KS 的速度检测部分与电动机的主轴同轴相联,在停车制动过程中,当主电动机转速低于 KS 的动作值时,其常开触点可将控制电路中反接制动的相应电路切断,完成制动停车。

电动机 M2 由交流接触器 KM4 控制其主电路的接通和断开,电动机 M3 由交流接触器 KM5 控制。

为保证主电路的正常运行,主电路中还设置了熔断器的短路保护环节和热继电器的过载保护环节。

### (2) 控制电路分析

① 主电动机正、反转启动与点动控制 当正转启动按钮 SB3 压下时,其两常开触点同时闭合,一常开触点(7 区)接通交流接触器 KM3 的线圈电路和时间继电器 KT 的线圈电路,时间继电器的常闭触点(3 区)在主电路中短接电流表 A,以防止电流对电流表的冲击;经延时断开后,电流表接入电路正常工作;KM3 的主触点(2 区)将主电路中限流电阻短接,其辅助动合触点(13 区)同时将中间继电器 KA 的线圈电路接通,KA 的常闭触点(9 区)将停车制动的基

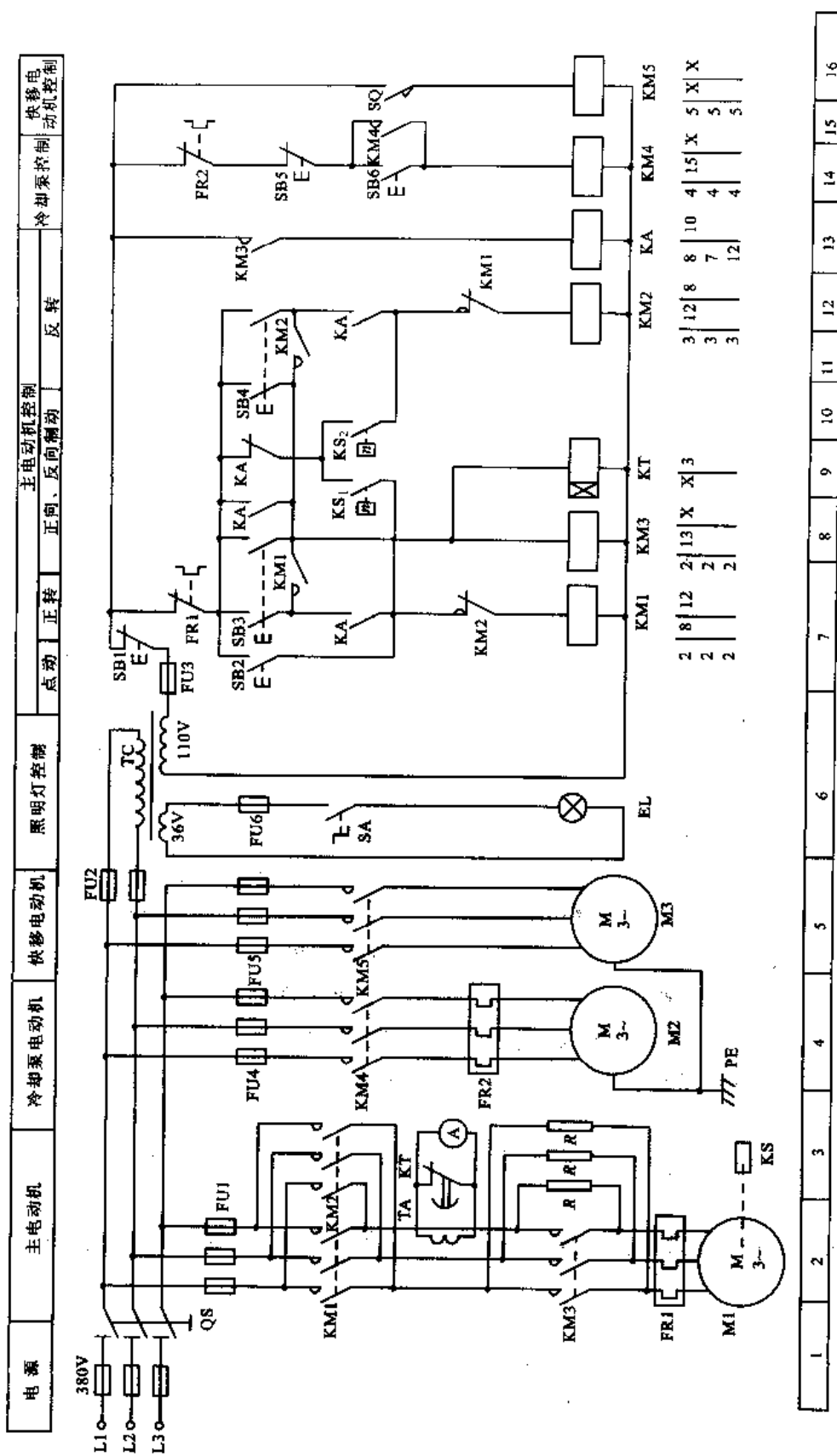


图 2-37 C650 车床控制线路

本电路切除,其动合触点(8区)与SB3的动合触点(7区)均在闭合状态,控制主电动机的交流接触器KM1的线圈电路得电工作并自锁,其主触点(2区)闭合,电动机正向直接启动并结束。KM1的自锁回路由它的常开辅助触点(7区)和KA的常开触点(9区)组成自锁回路,来维持KM1的通电状态。反向直接启动控制过程与其相同,只是启动按钮为SB4。

SB2为主电动机点动控制按钮。按下SB2点动按钮,直接接通KM1的线圈电路,电动机M1正向直接启动,这时KM3线圈电路并没有接通,因此其主触点不闭合,限流电阻R接入主电路限流,其辅助动合触点不闭合,KA线圈不能得电工作,从而使KM1线圈电路形不成自锁,松开按钮SB2,M1停转,实现了主电动机串联电阻限流的点动控制。

另外,接触器KM3的辅助触点数量是有限的,故在控制电路中使用了中间继电器KA。因为KA没有主触点,而KM3辅助触点又不够,所以用KM3来带一个KA,这样解决了在主电路中使用主触点,而控制电路辅助触点不够的问题。KA的线圈也可以直接和KM3的线圈并联使用。

② 主电动机反接制动控制电路 C650型卧式车床采用反接制动的方式进行停车制动,停车按钮按下后开始制动过程。当电动机转速接近零时,速度继电器的触点打开,结束制动。下面以原工作状态为正转时进行停车制动过程为例,说明电路的工作原理。

当电动机正向正常运转时,速度继电器KS的动合触点 $KS_2$ 闭合,制动电路处于准备状态,按下停车按钮SB1,切断控制电源,KM1、KM3、KA线圈均失电,此时控制反接制动电路工作与否的KA动断触点(9区)恢复原状闭合,与 $KS_2$ 触点一起,将反转交流接触器KM2的线圈电路接通,电动机M1接入反相序电流,反向启动转矩将平衡正向惯性转动转矩,强迫电动机迅速停车。当电动机速度降低到速度继电器的动作值时,速度继电器触点 $KS_2$ 复位打开,切断KM2的线圈电路,完成正转的反接制动。在反接制动过程中,KM3失电,所以限流电阻R一直起限制反接制动电流的作用。反转时的反接制动工作过程和正转时相似,此时在反转状态下, $KS_1$ 触点闭合,制动时,接通交流接触器KM1的线圈电路,进行反接制动。

③ 刀架的快速移动和冷却泵电动机的控制 刀架快速移动是由转动刀架手柄压动位置开关SQ,接通快速移动电动机M3的接触器KM5的线圈电路,KM5的主触点闭合,M3电动机启动运行,经传动系统驱动溜板带动刀架快速移动。

启动按钮SB6和停止按钮SB5控制接触器KM4线圈电路的通断,来完成冷却泵电动机M2的控制。

④ 辅助电路 开关SA可控制照明灯EL,且EL的电压为36V安全照明电压。

#### 4. C650卧式车床电气控制线路的特点

C650卧式车床电气控制线路的特点是:

(1) 主轴与进给电动机M1主电路具有正、反转控制、点动控制功能并设置有监视电动机绕组工作电流变化的电流表和电流互感器。

(2) 该机床采用反接制动的方法控制M1的正、反转制动。

(3) 能够进行刀架的快速移动。

## 本章小结

(1) 电气控制系统图主要有电气原理图、电器布置图和电气安装接线图。各种图纸有不同的用途和规定画法,重点应掌握电气原理图的规定画法及国家标准。

(2) 掌握电动机运行中的点动、连续运转、正反转、自动循环和调速等基本控制线路的特点以及各种电器及控制触点的逻辑关系。

(3) 较大容量的笼型异步电动机(大于 10 kW),一般都采用降压启动方式来启动,以避免过大的启动电流对电网及传动机械造成的冲击。常用的降压启动方法有星形—三角形、自耦变压器和使用软启动器等。要了解软启动器的概念和简单使用。启动过程中的控制方式通常采用时间控制原则。

(4) 常用的电气制动控制线路有反接制动和能耗制动。反接制动要避免反向再启动,要限制制动电流,一般采用速度控制原则。能耗制动可采用时间控制原则或速度控制原则。

(5) 了解变频调速的概念和变频器的简单使用。

(6) 重点掌握电气控制线路中常用的保护环节及其实现方法。

(7) 掌握电气控制线路的简单设计法原理。学会使用简单设计法设计控制电路。

(8) 掌握电气控制线路的分析基础。机床电气控制线路的复杂程度虽差异很大,但均是由电动机的启动、正反转、制动、点动控制、多电机启动的先后顺序控制等基本控制环节组成的。在对电气控制线路分析时,首先要对控制设备的结构组成、工作原理及运动要求等进行分析;其次,对复杂的控制线路要“化整为零”,按照主电路、控制电路和其他辅助电路等逐一分解、各个击破。

## 思考题与练习题

1. 三相笼型异步电动机在什么条件下可直接启动?试设计带有短路、过载、失压保护的三相笼型异步电动机直接启动的主电路和控制电路,对所设计的电路进行简要说明,并指出哪些元器件在电路中完成了哪些保护功能?
2. 某三相笼型异步电动机单向运转,要求采用自耦变压器降压启动。试设计主电路和控制电路,并要求有必要的保护。
3. 某三相笼型异步电动机单向运转,要求启动电流不能过大,制动时要快速停车。试设计主电路和控制电路,并要求有必要的保护。
4. 某三相笼型异步电动机可正反向运转,要求降压启动,快速停车。试设计主电路和控制电路,并要求有必要的保护。
5. 星形—三角形降压启动方法有什么特点并说明其适用场合?
6. 软启动器的启动和停车控制方式一般有哪些?与其他的启动方式相比有什么优点?
7. 三相笼型异步电动机有哪几种电气制动方式?各有什么特点和适用场合?
8. 三相笼型异步电动机的调速方法有哪几种?
9. 变频调速有哪两种控制方式?请简要说明。
10. 某机床主轴由一台三相笼型异步电动机拖动,润滑油泵由另一台三相笼型异步电动机拖动,均采用直接启动,工艺要求是:

- (1) 主轴必须在润滑油泵启动后,才能启动;
- (2) 主轴为正向运转,为调试方便,要求能正、反向点动;
- (3) 主轴停止后,才允许润滑油泵停止;
- (4) 具有必要的电气保护。

试设计主电路和控制电路,并对设计的电路进行简单说明。

- 11. M1 和 M2 均为三相笼型异步电动机,可直接启动,按下列要求设计主电路和控制电路:
  - (1) M1 先启动,经一段时间后 M2 自行启动;
  - (2) M2 启动后,M1 立即停车;
  - (3) M2 能单独停车;
  - (4) M1 和 M2 均能点动。
- 12. 设计一个控制线路,要求第一台电动机启动 10 s 后,第二台电动机自行启动,运行 5 s 后,第一台电动机停止并同时使第三台电动机自行启动,再运行 10 s,电动机全部停止。
- 13. 设计一小车运行控制线路,小车由异步电动机拖动,其动作程序如下:
  - (1) 小车由原位开始前进,到终端后自动停止;
  - (2) 在终端停留 2 min 后自动返回原位停止;
  - (3) 要求能在前进或后退途中任意位置都能停止或启动。
- 14. 简述分析电气原理图的一般步骤。
- 15. 对图 2-37 所示的 C650 型卧式车床的电气原理图,试分析和回答以下问题:
  - (1) 分析 C650 型卧式车床的工作过程;
  - (2) 写出 KM1 和 KM2 自锁回路的构成;
  - (3) 电流表 A 电路中的 KT 延时打开的常闭触点有何作用?
  - (4) KA 和 KM3 的逻辑相同,但它们能相互代替吗?

## 第三章 可编程序控制器概述

工业自动化控制中使用的可编程序控制器种类很多,不同厂家的产品各有特点,它们虽有一定的区别,但作为工业标准控制设备,可编程序控制器在结构组成、工作原理和编程方法等许多方面是基本相同的。本章主要介绍可编程序控制器的一般特性,重点讲解它的一般结构、工作原理和工作方式。

### 3.1 PLC 的产生和定义

#### 3.1.1 PLC 的产生

20 世纪 20 年代起,人们把各种继电器、定时器、接触器及其触点按一定的逻辑关系连接起来组成控制系统,控制各种生产机械,这就是大家所熟悉的传统继电接触器控制系统。由于它结构简单、容易掌握、价格便宜,在一定范围内能满足控制要求,因而使用面甚广,在工业控制领域中一直占主导地位。但是继电接触器控制系统有明显的缺点:设备体积大,可靠性差,动作速度慢,功能少,难于实现较复杂的控制,特别是由于它是靠硬连线逻辑构成的系统,接线复杂,当生产工艺或对象需要改变时,原有的接线和控制盘(柜)就要更换,所以通用性和灵活性较差。

到 20 世纪 60 年代,由于小型计算机的出现和大规模生产及多机群控的发展,人们曾试图用小型计算机来实现工业控制的要求,但由于价格高,输入、输出电路不匹配和编程技术复杂等原因,一直未能得到推广应用。

20 世纪 60 年代末期,美国的汽车制造业竞争激烈,各生产厂家的汽车型号不断更新,它必然要求生产线的控制系统亦随之改变,以及对整个控制系统重新配置。为抛弃传统的继电接触器控制系统的束缚,适应白热化的市场竞争要求,1968 年美国通用汽车公司(GM)公开招标,对汽车流水线控制系统提出具体要求,归纳起来是:

- (1) 编程方便,可现场修改程序;
- (2) 维修方便,采用插件式结构;
- (3) 可靠性高于继电器控制装置;
- (4) 体积小于继电器控制盘;
- (5) 数据可直接送入管理计算机;
- (6) 成本可与继电器控制盘竞争;
- (7) 输入可以是交流 115 V(美国电压标准);
- (8) 输出为交流 115 V,容量要求在 2 A 以上,可直接驱动接触器、电磁阀等;
- (9) 扩展时原系统改变最小;
- (10) 用户存储器至少能扩展到 4 KB。

以上就是著名的“GM 十条”。这些要求实际上提出了将继电器控制的简单易懂、使用方便、价格低廉的优点,与计算机的功能完善、灵活性、通用性好的优点结合起来,将继电器控制的硬连线逻辑转变为计算机的软件逻辑编程的设想。1969 年美国数据设备公司(DEC)根据上述要求,研制开发出世界上第一台可编程序控制器,并在 GM 公司汽车生产线上首次应用成功。当时人们把它称为可编程序逻辑控制器(Programmable Logical Controller),简称 PLC,它主要用来取代继电器逻辑控制,系统功能仅限于执行继电器逻辑、计时、计数等。

随着微电子技术的发展,20 世纪 70 年代中期出现了微处理器和微型计算机,人们将微机技术应用到 PLC 中,使得它能更多地发挥计算机的功能,不仅用逻辑编程取代硬连线逻辑,还增加了运算、数据传送和处理等功能,使其真正成为一种电子计算机工业控制设备。国外工业界在 1980 年正式命名其为可编程序控制器(Programmable Controller),简称 PC。但由于它和个人计算机(Personal Computer)的简称容易混淆,所以现在仍把可编程序控制器简称为 PLC。

进入 20 世纪 80 年代以来,随着大规模和超大规模集成电路等微电子技术的迅猛发展,以 16 位和 32 位微处理器构成的微机化 PLC 得到了惊人的发展,使 PLC 在概念、设计、性能价格比以及应用等方面都有了新的突破。不仅控制功能增强,功耗、体积减小,成本下降,可靠性提高,编程和故障检测更为灵活方便,而且远程 I/O 和通信网络、数据处理以及图像显示也有了长足的发展,所有这些已经使 PLC 应用于连续生产的过程控制系统,使之成为今天自动化技术的三大支柱之一。

### 3.1.2 PLC 的定义

PLC 一直在飞速发展,因此到现在为止,还未能对其下一个十分确切的定义。

国际电工委员会(IEC)曾于 1982 年 11 月颁发了可编程控制器标准草案第一稿,1985 年 1 月发表了第二稿,1987 年 2 月颁布了第三稿。该草案中对可编程控制器的定义是:

“可编程序控制器是一种数字运算操作的电子系统,专为工业环境而设计。它采用了可编程序的存储器,用来在其内部存储执行逻辑运算、顺序控制、定时、计数和算术运算等操作的指令,并通过数字式和模拟式的输入和输出,控制各种类型机械的生产过程。而有关的外围设备,都应按易于与工业系统联成一个整体,易于扩充其功能的原则设计。”

定义强调了 PLC 应直接应用于工业环境,它必须具有很强的抗干扰能力,广泛的适应能力和应用范围。这也是区别于一般微机控制系统的一个重要特征。

定义强调了 PLC 是“数字运算操作的电子系统”,它也是一种计算机。它是“专为在工业环境下应用而设计的”工业计算机。这种工业计算机采用“面向用户的指令”,因此编程方便。它能完成逻辑运算、顺序控制、定时、计数和算术运算等操作,它还具有“数字量和模拟量输入和输出”的能力,并且非常容易与“工业控制系统联成一体”,易于“扩充”。

应该强调的是,PLC 与以往所讲的鼓式、机械式的顺序控制器在“可编程”方面有质的区别。由于 PLC 引入了微处理机及半导体存储器等新一代电子器件,并用规定的指令进行编程,所以能灵活地修改程序,即它是用软件方式来实现“可编程”的目的。

## 3.2 PLC 的特点

现代工业生产过程是复杂多样的,它们对控制的要求也各不相同。PLC 一经出现就受到了广大工程技术人员的欢迎。

### 1. 抗干扰能力强,可靠性高

微机虽然具有很强的功能,但抗干扰能力差,工业现场的电磁干扰,电源波动,机械振动,温度和湿度的变化,都可以使一般通用微机不能正常工作。而 PLC 在电子线路、机械结构以及软件结构上都吸取了生产厂家长期积累的生产控制经验,主要模块均采用大规模与超大规模集成电路,I/O 系统设计有完善的通道保护与信号调理电路;在结构上对耐热、防潮、防尘、抗震等都有周到的考虑;在硬件上采用隔离、屏蔽、滤波、接地等抗干扰措施;在软件上采用数字滤波等抗干扰和故障诊断措施;所有这些使 PLC 具有较高的抗干扰能力。PLC 的平均无故障时间通常在几万小时以上,这是一般微机不能比拟的。

继电器接触器控制系统虽有较好的抗干扰能力,但使用了大量的机械触点,使设备连线复杂,且触点在开闭时易受电弧的损害,寿命短,系统可靠性差。而 PLC 采用微电子技术,大量的开关动作由无触点的电子存储器件来完成,大部分继电器和繁杂的连线被软件程序所取代,故寿命长,可靠性大大提高。

### 2. 控制系统结构简单,通用性强

PLC 及外围模块品种多,可由各种组件灵活组合成各种大小和不同要求的控制系统。在 PLC 构成的控制系统中,只需在 PLC 的端子上接入相应的输入/输出信号线即可,不需要诸如继电器之类的物理电子器件和大量而又繁杂的硬接线线路。当控制要求改变,需要变更控制系统的功能时,可以用编程器在线或离线修改程序,同一个 PLC 装置用于不同的控制对象,只是输入/输出组件和应用软件的不同。PLC 的输入/输出可直接与交流 220 V,直流 24 V 等强电相连,并具有较强的带负载能力。

### 3. 编程方便,易于使用

PLC 是面向用户的设备,PLC 的设计者充分考虑到现场工程技术人员的技能和习惯,PLC 程序的编制,采用梯形图或面向工业控制的简单指令形式。梯形图与继电器原理图相类似,这种编程语言形象直观,容易掌握,不需要专门的计算机知识和语言,只要具有一定的电工和工艺知识的人员都可在短时间学会。

### 4. 功能完善

PLC 的输入/输出系统功能完善,性能可靠,能够适应于各种形式和性质的开关量和模拟量的输入/输出。在 PLC 内部具备许多控制功能,诸如时序、计算器、主控继电器以及移位寄存器、中间寄存器等。由于采用了微处理器,它能够很方便地实现延时、锁存、比较、跳转和强制 I/O 等诸多功能,不仅具有逻辑运算、算术运算、数制转换以及顺序控制功能,而且还具备模拟运算、显示、监控、打印及报表生成功能。此外,它还可以和其他微机系统、控制设备共同组成分布式或分散式控制系统,还能实现成组数据传送、矩阵运算、闭环控制、排序与查表、函数运算及快速中断等功能。因此 PLC 具有极强的适应性,能够很好地满足各种类型控制的需要。

### 5. 设计、施工、调试的周期短

用继电器控制完成一项控制工程,必须首先按工艺要求画出电气原理图,然后画出继电器屏(柜)的布置和接线图等,进行安装调试,以后修改起来十分不便。而采用 PLC 控制,由于其硬软件齐全,为模块化积木式结构,且已商品化,故仅需按性能、容量(输入/输出点数、内存大小)等选用组装,而大量具体的程序编制工作也可在 PLC 到货前进行,因而缩短了设计周期,使设计和施工可同时进行。由于用软件编程取代了硬接线实现控制功能,大大减轻了繁重的安装接线工作,缩短了施工周期。因为 PLC 是通过程序完成控制任务的,采用了方便用户的工业编程语言,且都具有强制和仿真的功能,故程序的设计、修改和调试都很方便,这样可大大缩短设计和投运周期。

### 6. 体积小,维护操作方便

PLC 体积小,质量轻,便于安装。PLC 的输入/输出系统能够直观地反映现场信号的变化状态,还能通过各种方式直观地反映控制系统的运行状态,如内部工作状态、通信状态、I/O 点状态、异常状态和电源状态等,对此均有醒目的指示,非常有利于运行和维护人员对系统进行监视。

## 3.3 PLC 的应用和发展

### 3.3.1 PLC 的发展状况

PLC 诞生不久即显示了其在工业控制中的重要地位,如日本、德国、法国等国家相继研制成各自的 PLC。PLC 技术随着计算机和微电子技术的发展而迅速发展,由最初的一位机发展为 8 位机。随着微处理器 CPU 和微型计算机技术在 PLC 中的应用,形成了现代意义上的 PLC。现在的 PLC 产品已使用了 16 位、32 位高性能微处理器,而且实现了多处理器的多通道处理,通信技术使 PLC 的应用得到进一步发展。如今,可编程序控制器技术已非常成熟。

目前,世界上有 200 多个厂家生产可编程序控制器产品,比较著名的厂家有美国的 AB、通用(GE)、莫迪康(MODICON)、日本的三菱(MITSUBISHI)、欧姆龙(OMRON)、富士电机(FUJI)、松下电工、德国的西门子(SIEMENS)、法国的 TE、施耐德(SCHNEIDER)、韩国的三星(SUMSUNG)、LG 等(其中 MODICON 和 TE 已归到 SCHNEIDER 旗下)。

### 3.3.2 PLC 的发展趋势

PLC 总的发展趋势是向高集成度、小体积、大容量、高速度、易使用、高性能方向发展。具体表现在以下几个方面。

#### 1. 向小型化、专用化、低成本方向发展

20 世纪 80 年代初,小型 PLC 在价格上还高于小系统用的继电器控制装置。随着微电子技术的发展,新型器件大幅度的提高功能和降低价格,使 PLC 结构更为紧凑,相当于一本精装书的大小,操作使用十分简便。PLC 的功能不断增加,将原来大、中型 PLC 才有的功能部分地移植到小型 PLC 上,如模拟量处理、数据通信和复杂的功能指令等,但价格不断下降,真正成为现代电气控制系统中不可替代的控制装置。

## 2. 向大容量,高速度方向发展

大型 PLC 采用多微处理器系统,有的采用了 32 位微处理器,可同时进行多任务操作,处理速度提高,特别是增强了过程控制和数据处理的功能。另外,存储容量大大增加。

## 3. 智能型 I/O 模块的发展

智能 I/O 模块是以微处理器和存储器为基础的功能部件,它们的 CPU 与 PLC 的主 CPU 并行工作,占用主 CPU 的时间很少,有利于提高 PLC 的扫描速度。它们本身就是一个小的微型计算机系统,有很强的信息处理能力和控制功能,有的模块甚至可以自成系统,单独工作。它们可以完成 PLC 的主 CPU 难以兼顾的功能,简化了某些控制系统的系统设计和编程,提高了 PLC 的适应性和可靠性。智能 I/O 模块主要有模拟量 I/O、高速计数输入、中断输入、机械运动输入、热电偶输入、热电阻输入、条形码阅读器、多路 BCD 码输入/输出、模糊控制器、PID 回路控制和各种通信等模块。

## 4. 基于 PC 的编程软件取代编程器

随着计算机的日益普及,越来越多的用户使用基于个人计算机的编程软件。编程软件可以对 PLC 控制系统的硬件组态,即设置硬件的结构和参数,例如设置各框架各个插槽上模块的型号、模块的参数、各串行通信接口的参数等。在屏幕上可以直接生成和编辑梯形图、语句表、功能块图和顺序功能图程序,并可以实现不同编程语言的相互转换。程序被编译下载到 PLC,也可以将用户程序上传到计算机。程序可以存盘或打印,通过网络或 Modem 卡,还可以实现远程操作。

编程软件的调试和监控功能远远超过手持式编程器,例如在调试时可以设置执行用户程序的扫描次数,有的编程软件可以在调试程序时设置断点,有的具有跟踪功能,用户可以周期性地选择保存若干编程元件的历史数据,并可以将数据上传后存为文件。

通过与 PLC 通信,可以在梯形图中显示触点的通断和线圈的状态,查找复杂电路的故障非常方便。

## 5. PLC 编程语言的标准化

与个人计算机相比,PLC 的硬件、软件的体系结构都是封闭的而不是开放的。在硬件方面,各厂家的 CPU 模块和 I/O 模块互不通用。PLC 的编程语言和指令系统的功能和表达方式也不一致,因此各厂家的可编程序控制器互不兼容。为了解决这一问题,IEC(国际电工委员会)制定了可编程序控制器标准(IEC1131),其中的第 3 部分(IEC1131-3)是 PLC 的编程语言标准。标准中共有五种编程语言,其中的顺序功能图(SFC)是一种结构块控制程序流程图,梯形图和功能块图是两种图形语言,还有两种文字语言——语句表和结构文本。除了提供几种编程语言供用户选择外,标准还允许编程者在同一程序中使用多种编程语言,这使编程者能够选择不同的语言来适应特殊的工作。

目前已有越来越多的工控产品厂商推出了符合 IEC1131-3 标准的 PLC 指令系统或在 PC(个人计算机)上运行的软件包(软件 PLC)。如西门子公司的 STEP 7-Micro/WIN32 编程软件给用户提供了两套指令集,一套符合 IEC1131-3 标准,另一套指令集(SIMATIC 指令集)中的大多数指令也符合 IEC1131-3 标准。

## 6. PLC 通信的易用化

PLC 的通信联网功能使它能与个人计算机和其他智能控制设备交换数字信息,使系统形成一个统一的整体,实现分散控制和集中管理。通过双绞线、同轴电缆或光纤联网,信息可以

传送到几十公里远的地方,通过 Modem 和互联网可以与世界上其他地方的计算机装置通信。

为了尽量减少用户在通信编程方面的负担,PLC 厂商做了大量的工作,使设备之间的通信自动地周期性地地进行,不需要用户为通信编程,用户的工作只是在组成系统时做一些硬件或软件上的初始化设置。

### 7. 组态软件与 PLC 的软件化

个人计算机(PC)的价格便宜,有很强的数学运算、数据处理、通信和人机交互的功能。过去个人计算机主要用作 PLC、操作站或人机接口终端,工业控制现场一般使用工业控制计算机(IPC),这样相应地出现了应用于工业控制系统的组态软件,利用这些软件可以方便地进行工业控制流程的实时和动态监控,完成报警、历史趋势和各种复杂的控制功能,同时节约控制系统的设计时间,提高系统的可靠性。既然使用了 PC,为何不把 PLC 的功能也用软件在 PC 上来实现呢?这也就是软 PLC 产生的动机,再加上现在智能 I/O 终端的发展,更使得软 PLC 的开发出现了上升的势头。目前已有许多家厂商推出了在 PC 上运行的可实现 PLC 功能的软件包。

### 8. PLC 与现场总线相结合

IEC 对现场总线(Field bus)的定义是:“安装在制造和过程区域的现场装置与控制室内的自动控制装置之间的数字式、串行、多点通信的数据总线称为现场总线”。它是当前工业自动化的热点之一。现场总线以开放的、独立的、全数字化的双向多变量通信代替 0~10 mA 或 4~20 mA 的现场电动仪表信号。现场总线 I/O 集检测、数据处理、通信为一体,可以代替变送器、调节器、记录仪等模拟仪表。它不需要框架、机柜,可以直接安装在现场导轨槽上,现场总线 I/O 的接线极为简单,只需一根电缆,从主机开始,沿数据链从一个现场总线 I/O 连接到下一个现场总线 I/O。使用现场总线后,自控系统的配线、安装、调试和维护等方面的费用可以节约 2/3 左右,现场总线 I/O 与 PLC 可以组成功能强大的、廉价的 DCS 系统。

现场总线控制系统将 DCS 的控制站功能分散给现场控制设备,仅靠现场总线设备就可以实现自动控制的基本功能。例如,将电动调节阀及其驱动电路、输出特性补偿、PID 控制和运算、阀门自校验和自诊断功能集成在一起,再配上温度变送器就可以组成一个闭环温度控制系统,有的传感器中也植入了 PID 控制功能。现在功能强大的 PLC 也配有和现场总线联网的模块,使之可以就近挂接到现场总线上。

使用现场总线后,操作员可以在中央控制室实现远程监控,对现场设备进行参数调整,还可以通过现场设备的自诊断功能预测故障和寻找故障点。

## 3.3.3 PLC 的应用领域

PLC 的初期由于其价格高于继电器控制装置,使得其应用受到限制。但最近十多年来,PLC 的应用面越来越广,其主要原因是:一方面由于微处理器芯片及有关元件的价格大大下降,使得 PLC 的成本下降;另一方面 PLC 的功能大大增强,它也能解决复杂的计算和通信问题。目前 PLC 在国内外已广泛应用于钢铁、采矿、水泥、石油、化工、电力、机械制造、汽车、装卸、造纸、纺织、环保和娱乐等行业。PLC 的应用范围通常可分成以下 5 种类型。

### (1) 顺序控制

这是 PLC 应用最广泛的领域,也是最适合 PLC 使用的领域。它用来取代传统的继电器顺序控制。PLC 应用于单机控制、多机群控、生产自动线控制等。例如:注塑机械、印刷机械、

订书机械、包装机械、切纸机械、组合机床、磨床、装配生产线、电镀流水线及电梯控制等。

#### (2) 运动控制

PLC 制造商目前已提供了拖动步进电机或伺服电机的单轴或多轴位置控制模块,在多数情况下,PLC 把描述目标位置的数据送给模块,其输出移动一轴或数轴到目标位置。每个轴移动时,位置控制模块保持适当的速度和加速度,确保运动平滑。

相对来说,位置控制模块比 CNC 装置体积更小,价格更低,速度更快,操作更方便。

#### (3) 过程控制

PLC 还能控制大量的过程参数,例如:温度、流量、压力、液位和速度。PID 模块提供了使 PLC 具有闭环控制的功能,即一个具有 PID 控制能力的 PLC 可用于过程控制。当过程控制中某个变量出现偏差时,PID 控制算法会计算出正确的输出,把变量保持在设定值上。

#### (4) 数据处理

在机械加工中,PLC 作为主要的控制和管理系统用于 CNC 和 NC 系统中,可以完成大量的数据处理工作。

#### (5) 通信网络

PLC 的通信包括主机与远程 I/O 之间的通信、多台 PLC 之间的通信、PLC 和其他智能控制设备(如计算机、变频器、数控装置)之间的通信。PLC 与其他智能控制设备一起,可以组成“集中管理、分散控制”的分布式控制系统。

### 3.4 PLC 的分类

PLC 发展到今天,已经有多种形式,而且功能也不尽相同,分类时,一般按以下原则来考虑。

#### 1. 按 I/O 点数容量分类

一般而言,处理的 I/O 点数比较多,则控制关系比较复杂,用户要求的程序存储器容量比较大,要求 PLC 指令及其他功能比较多,指令执行的过程也比较快等。按 PLC 的输入、输出点数的多少可将 PLC 分为以下三类。

##### (1) 小型机

小型 PLC 的功能一般以开关量控制为主,小型 PLC 输入、输出总点数一般在 256 点以下,用户程序存储器容量在 4 K 字左右。现在的高性能小型 PLC 还具有一定的通信能力和少量的模拟量处理能力。这类 PLC 的特点是价格低廉,体积小巧,适合于控制单台设备和开发机电一体化产品。

典型的小型机有 SIEMENS 公司的 S7-200 系列、OMRON 公司的 CPM2A 系列、MITSUBISHI 公司的 FX 系列和 AB 公司的 SLC500 系列等整体式 PLC 产品。

##### (2) 中型机

中型 PLC 的输入、输出总点数在 256~2 048 点之间,用户程序存储器容量达到 8 K 字左右。中型 PLC 不仅具有开关量和模拟量的控制功能,还具有更强的数字计算能力,它的通信功能和模拟量处理能力更强大。中型机的指令比小型机更丰富,中型机适用于复杂的逻辑控制系统以及连续生产线的过程控制场合。

典型的中型机有 SIEMENS 公司的 S7-300 系列、OMRON 公司的 C200H 系列、AB 公

司的 SLC500 系列等模块式 PLC 产品。

### (3) 大型机

大型 PLC 的输入、输出总点数在 2 048 点以上,用户程序存储器容量达到 16 K 字以上。大型 PLC 的性能已经与工业控制计算机相当,它具有计算、控制和调节的功能,还具有强大的网络结构和通信联网能力,有些 PLC 还具有冗余能力。它的监视系统采用 CRT 显示,能够表示过程的动态流程,记录各种曲线, PID 调节参数等;它配备多种智能板,构成一台多功能系统。这种系统还可以和其他型号的控制器的互连,和上位机相连,组成一个集中分散的生产过程和产品质量控制系统。大型机适用于设备自动化控制、过程自动化控制和过程监控系统。

典型的大型 PLC 有 SIEMENS 公司的 S7-400、OMRON 公司的 CVM1 和 CS1 系列、AB 公司的 SLC5/05 等系列产品。

以上划分没有一个十分严格的界限,随着 PLC 技术的飞速发展,某些小型 PLC 也具有中型或大型 PLC 的功能,这也是 PLC 的发展趋势。

## 2. 按结构形式分

根据 PLC 结构形式的不同,PLC 主要可分为整体式和模块式两类。

### (1) 整体式结构

整体式结构的特点是将 PLC 的基本部件,如 CPU 板、输入板、输出板、电源板等紧凑地安装在一个标准机壳内,构成一个整体,组成 PLC 的一个基本单元(主机)或扩展单元。基本单元上设有扩展端口,通过扩展电缆与扩展单元相连,配有许多专用的特殊功能模块,如模拟量输入/输出模块、热电偶、热电阻模块、通信模块等,以构成 PLC 不同的配置。整体式结构的 PLC 体积小,成本低,安装方便。

微型和小型 PLC 一般为整体式结构。如西门子的 S7-200 系列。

### (2) 模块式结构

模块式结构的 PLC 是由一些模块单元构成,这些标准模块如 CPU 模块、输入模块、输出模块、电源模块和各种功能模块等,将这些模块插在框架上或基板上即可。各模块功能是独立的,外形尺寸是统一的,可根据需要灵活配置。

目前,中、大型 PLC 多采用这种结构形式。如西门子的 S7-300 和 S7-400 系列。

整体式 PLC 每一个 I/O 点的平均价格比模块式的便宜,在小型控制系统中一般采用整体式结构。但是模块式 PLC 的硬件组态方便灵活,I/O 点数的多少、输入点数与输出点数的比例、I/O 模块的使用等方面的选择余地都比整体式 PLC 大的多,维修时更换模块、判断故障范围也很方便,因此较复杂的、要求较高的系统一般选用模块式 PLC。

## 3.5 PLC 的系统组成

PLC 种类繁多,但其组成结构和工作原理基本相同。用 PLC 实施控制,其实质是按一定算法进行输入/输出变换,并将这个变换给以物理实现,应用于工业现场。PLC 专为工业现场应用而设计,采用了典型的计算机结构,它主要是由 CPU、电源、存储器和专门设计的输入/输出接口电路等组成。PLC 的结构框图如图 3-1 所示。

### 1. 中央处理单元(CPU)

中央处理单元(CPU)一般由控制器、运算器和寄存器组成,这些电路都集成在一个芯片

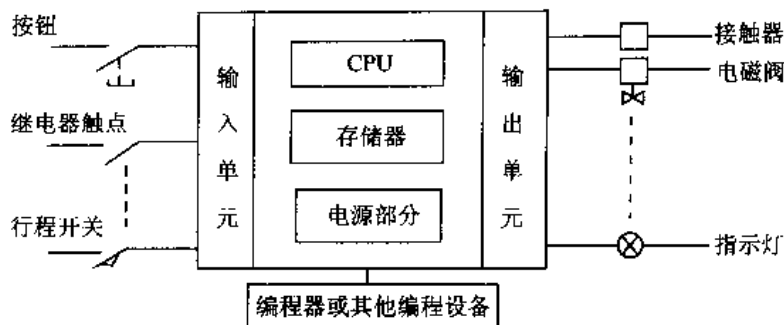


图 3-1 PLC 结构框图

内。CPU 通过数据总线、地址总线和控制总线与存储单元、输入/输出接口电路相连接。

与一般计算机一样,CPU 是 PLC 的核心,它按 PLC 中系统程序赋予的功能指挥 PLC 有条不紊地进行工作。用户程序和数据事先存入存储器中,当 PLC 处于运行方式时,CPU 按循环扫描方式执行用户程序。

CPU 的主要任务有控制用户程序和数据的接收与存储;用扫描的方式通过 I/O 部件接收现场信号的状态或数据,并存入输入映像寄存器或数据存储器中;诊断 PLC 内部电路的工作故障和编程中的语法错误等;PLC 进入运行状态后,从存储器逐条读取用户指令,经过命令解释后按指令规定的任务进行数据传送、逻辑或算术运算等;根据运算结果,更新有关标志位的状态和输出映像寄存器的内容,再经输出部件实现输出控制、制表打印或数据通信等功能。

不同型号的 PLC 其 CPU 芯片是不同的,有采用通用 CPU 芯片的,有采用厂家自行设计的专用 CPU 芯片的。CPU 芯片的性能关系到 PLC 处理控制信号的能力与速度,CPU 位数越高,系统处理的信息量越大,运算速度也越快。PLC 的功能是随着 CPU 芯片技术的发展而提高和增强。

## 2. 存储器

PLC 的存储器包括系统存储器和用户存储器两部分。

系统存储器用来存放由 PLC 生产厂家编写的系统程序,并固化在 ROM 内,用户不能直接更改。它使 PLC 具有基本的功能,能够完成 PLC 设计者规定的各项工作。系统程序质量的好坏,很大程度上决定了 PLC 的性能,其内容主要包括三部分。

第一部分为系统管理程序。它主要控制 PLC 的运行,使整个 PLC 按部就班地工作。

第二部分为用户指令解释程序。通过用户指令解释程序,将 PLC 的编程语言变为机器语言指令,再由 CPU 执行这些指令。

第三部分为标准程序模块与系统调用。它包括许多不同功能的子程序及其调用管理程序,如完成输入、输出及特殊运算等的子程序。PLC 的具体工作都是由这部分程序来完成的,这部分程序的多少也决定了 PLC 性能的高低。

用户存储器包括用户程序存储器(程序区)和数据存储器(数据区)两部分。用户程序存储器用来存放用户针对具体控制任务用规定的 PLC 编程语言编写的各种用户程序。用户程序存储器根据所选用的存储器单元类型的不同,可以是 RAM、EPROM 或 EEPROM 存储器,其内容可以由用户任意修改或增删。用户数据存储器可以用来存放(记忆)用户程序中所使用器件的 ON/OFF 状态和数值、数据等。它的大小关系到用户程序容量的大小,是反映 PLC 性能的重要指标之一。

PLC 使用的存储器类型有三种。

#### (1) 随机存取存储器(RAM)

用户可以用编程装置读出 RAM 中的内容,也可以将用户程序写入 RAM,因此 RAM 又叫读/写存储器。它是易失性的存储器,它的电源中断后,储存的信息将会丢失。

RAM 的工作速度高,价格便宜,改写方便。在关断可编程序控制器的外部电源后,可用锂电池保存 RAM 中的用户程序和某些数据。锂电池可用 2~5 年,需要更换锂电池时,由 PLC 发出信号通知用户。现在大部分 PLC 已不用锂电池来完成掉电保护功能了。

#### (2) 只读存储器(ROM)

ROM 的内容只能读出,不能写入。它是非易失的,它的电源消失后,仍能保存储存的内容。ROM 一般用来存放可编程序控制器的系统程序。

#### (3) 可电擦除可编程的只读存储器(EEPROM 或 E<sup>2</sup>PROM)

它是非易失性的,但是可以用编程装置对它编程,兼有 ROM 的非易失性和 RAM 的随机存取优点,但是将信息写入它所需的时间比 RAM 长得多。EEPROM 用来存放用户程序和需长期保存的重要数据。

### 3. 输入/输出单元

PLC 的输入和输出信号类型可以是开关量、模拟量和数字量。输入/输出单元从广义上分包含两部分:一是与被控设备相连接的接口电路,另一部分是输入和输出的映像寄存器。

输入单元接收来自用户设备的各种控制信号,如限位开关、操作按钮、选择开关、行程开关以及其他一些传感器的信号。通过接口电路将这些信号转换成中央处理器能够识别和处理的信号,并存到输入映像寄存器。运行时 CPU 从输入映像寄存器读取输入信息并进行处理,将处理结果放到输出映像寄存器。输出映像寄存器由输出点相对应的触发器组成,输出接口电路将其由弱电控制信号转换成现场需要的强电信号输出,以驱动电磁阀、接触器、指示灯等被控设备的执行元件。

下面简单介绍开关量输入/输出接口电路。

#### (1) 输入接口电路

为防止各种干扰信号和高电压信号进入 PLC,影响其可靠性或造成设备损坏,现场输入接口电路一般由光电耦合电路进行隔离。光电耦合电路的关键器件是光耦合器,一般由发光二极管和光电三极管组成。

通常 PLC 的输入类型可以是直流、交流和交直流。输入电路的电源可由外部供给,有的也可由 PLC 内部提供。

#### (2) 输出接口电路

输出接口电路通常有三种类型:继电器输出型、晶体管输出型和晶闸管输出型。每种输出电路都采用电气隔离技术,电源由外部提供,输出电流一般为 0.5~2 A,输出电流的额定值与负载的性质有关。

为使 PLC 避免受瞬间大电流的作用而损坏,输出端外部接线必须采用保护措施:一是输入和输出公共端接熔断器;二是采用保护电路,对交流感性负载,一般用阻容吸收回路,对直流感性负载用续流二极管。

由于输入和输出端是靠光信号耦合的,在电气上是完全隔离的,因此输出端的信号不会反馈到输入端,也不会产生地线干扰或其他串扰,因此 PLC 具有很高的可靠性和极强的抗干扰

能力。

#### 4. 电源部分

PLC 一般使用 220 V 的交流电源,内部的开关电源为 PLC 的中央处理器、存储器等电路提供 5V、12V、24V 等直流电源,使 PLC 能正常工作。

电源部件的位置形式可有多种,对于整体式结构的 PLC,通常电源封装到机壳内部;对于模块式 PLC,有的采用单独电源模块,有的将电源与 CPU 封装到一个模块中。

#### 5. 扩展接口

扩展接口用于将扩展单元以及功能模块与基本单元相连,使 PLC 的配置更加灵活以满足不同控制系统的需要。

#### 6. 通信接口

为了实现“人—机”或“机—机”之间的对话,PLC 配有多种通信接口。PLC 通过这些通信接口可以与监视器、打印机和其他的 PLC 或计算机相连。

当 PLC 与打印机相连时,可将过程信息、系统参数等输出打印;当与监视器(CRT)相连时,可将过程图像显示出来;当与其他 PLC 相连时,可以组成多机系统或联成网络,实现更大规模的控制;当与计算机相连时,可以组成多级控制系统,实现控制与管理相结合的综合控制。

#### 7. 编程器

编程器的作用是供用户进行程序的编制、编辑、调试和监视。

编程器有简易型和智能型两类。简易型的编程器只能联机编程,且往往需要将梯形图转化为机器语言助记符(语句表)后,才能输入。它一般由简易键盘和发光二极管或其他显示器件组成。智能型的编程器又称图形编程器,它可以联机编程,也可以脱机编程,具有 LCD 或 CRT 图形显示功能,可以直接输入梯形图和通过屏幕对话。

还可以利用 PC 作为编程器,PLC 生产厂家配有相应的编程软件,使用编程软件可以在屏幕上直接生成和编辑梯形图、语句表、功能块图和顺序功能图程序,并可以实现不同编程语言的相互转换。程序被编译后下载到 PLC,也可以将 PLC 中的程序上传到计算机。程序可以存盘或打印,通过网络,还可以实现远程编程和传送。现在已有些 PLC 不再提供编程器,而只提供微机编程软件了,并且配有相应的通信连接电缆。

#### 8. 其他部件

有些 PLC 还可配有 EPROM 写入器、存储器卡等其他外部设备。

### 3.6 PLC 与继电器控制系统的区别

PLC 的梯形图与继电器控制线路图十分相似,主要原因是 PLC 梯形图大致上沿用了继电器控制的电路元件符号和术语,仅个别之处有些不同。同时,信号的输入/输出形式及控制功能基本上也是相同的,但 PLC 的控制与继电器的控制又有根本的不同之处,主要表现在以下几个方面。

#### 1. 控制逻辑

继电器控制逻辑采用硬接线逻辑,利用继电器机械触点的串联或并联,及延时继电器的滞后动作等组合成控制逻辑,其接线多而复杂、体积大、功耗大、故障率高,一旦系统构成后,想再改变或增加功能都很困难。另外,继电器触点数目有限,每个只有 4~8 对触点。因此灵活性

和扩展性很差。而 PLC 采用存储器逻辑,其控制逻辑以程序方式存储在内存中,要改变控制逻辑,只需改变程序即可,故称为“软接线”。因此灵活性和扩展性都很好。

### 2. 工作方式

电源接通时,继电器控制线路中各继电器同时都处于受控状态,即该吸合的都应吸合,不吸合的都因受某种条件限制不能吸合,它属于并行工作方式。而 PLC 的控制逻辑中,各内部器件都处于周期性循环扫描过程中,属于串行工作方式。

### 3. 可靠性和可维护性

继电器控制逻辑使用了大量的机械触点,连线也多。触点开闭时会受到电弧的损坏,并有机械磨损,寿命短,因此可靠性和可维护性差。而 PLC 采用微电子技术,大量的开关动作由无触点的半导体电路来完成,体积小、寿命长、可靠性高。PLC 还配有自检和监督功能,能检查出自身的故障,并随时显示给操作人员,还能动态地监视控制程序的执行情况,为现场调试和维护提供了方便。

### 4. 控制速度

继电器控制逻辑依靠触点的机械动作实现控制,工作频率低,触点的开闭动作一般在几十 ms 数量级。另外,机械触点还会出现抖动问题。而 PLC 是由程序指令控制半导体电路来实现控制,属于无触点控制,速度极快,一般一条用户指令的执行时间在  $\mu\text{s}$  数量级,且不会出现抖动。

### 5. 定时控制

继电器控制逻辑利用时间继电器进行时间控制。一般来说,时间继电器存在定时精度不高,定时范围窄,且易受环境湿度和温度变化的影响,调整时间困难等问题。PLC 使用半导体集成电路做定时器,时基脉冲由晶体振荡器产生,精度相当高,且定时时间不受环境的影响,定时范围一般从 0.001 s 到若干天或更长。用户可根据需要在程序中设置定时值,然后由软件来控制定时时间。

### 6. 设计和施工

使用继电器控制逻辑完成一项控制工程,其设计、施工、调试必须依次进行,周期长,而且修改困难。工程越大,这一点就越突出。而用 PLC 完成一项控制工程,在系统设计完成以后,现场施工和控制逻辑的设计(包括梯形图设计)可以同时进行,周期短,且调试和修改都很方便。

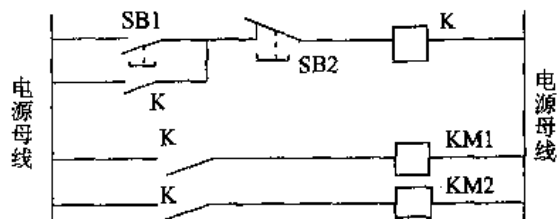
从以上几个方面的比较可知,PLC 在性能上比继电器控制逻辑优异,特别是可靠性高、通用性强、设计施工周期短、调试修改方便,而且体积小、功耗低、使用维护方便。但在很小的系统中使用时,价格要高于继电器系统。

## 3.7 PLC 的工作原理

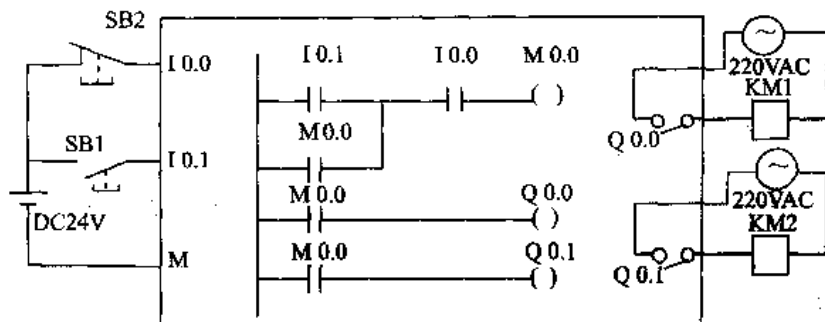
### 3.7.1 PLC 的工作方式与运行框图

众所周知,继电器控制系统是一种“硬件逻辑系统”,如图 3-2(a)所示,它的三条支路是并行工作的,当按下按钮 SB1,中间继电器 K 得电,K 的两个触点闭合,接触器 KM1、KM2 同时得电并产生动作。所以继电器控制系统采用的是并行工作方式。

而 PLC 是一种工业控制计算机,故它的工作原理是建立在计算机工作原理基础之上,即通过执行反映控制要求的用户程序来实现的,如图 3-2(b)所示。但是 CPU 是以分时操作方



(a) 继电器控制系统简图



(b) 用 PLC 实现控制功能的接线示意图

图 3-2 PLC 控制系统与继电器控制系统的比较

式来处理各项任务的,计算机在每一瞬间只能做一件事,所以程序的执行是按程序顺序依次完成相应各电器的动作,所以它属于串行工作方式。

概括而言,PLC 是按集中输入、集中输出,周期性循环扫描的方式进行工作的。每一次扫描所用的时间称为扫描周期或工作周期。CPU 从第一条指令执行开始,按顺序逐条地执行用户程序直到用户程序结束,然后返回第一条指令开始新一轮扫描。PLC 就是这样周而复始地重复上述循环扫描的。

PLC 工作的全过程可用图 3-3 所示的运行框图来表示。整个过程可分为三部分。

第一部分是上电处理。机器上电后对 PLC 系统进行一次初始化,包括硬件初始化, I/O 模块配置检查,停电保持范围设定及其他初始化处理等。

第二部分是扫描过程。PLC 上电处理完成以后进入扫描工作过程。先完成输入处理,其次完成与其他外设的通信处理,再次进行时钟、特殊寄存器更新。当 CPU 处于 STOP 方式时,转入执行自诊断检查。当 CPU 处于 RUN 方式时,还要完成用户程序的执行和输出处理,再转入执行自诊断检查。

第三部分是出错处理。PLC 每扫描一

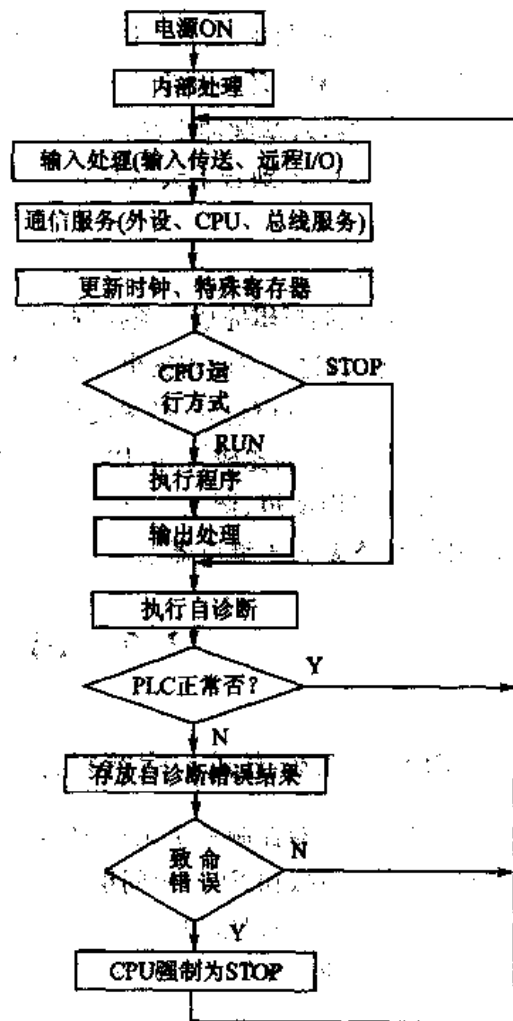


图 3-3 PLC 运行框图

次,执行一次自诊断检查,确定 PLC 自身的动作是否正常,如 CPU、电池电压、程序存储器、I/O 和通信等是否异常或出错。如检查出异常时,CPU 面板上的 LED 及异常继电器会接通,在特殊寄存器中会存入出错代码;当出现致命错误时,CPU 被强制为 STOP 方式,所有的扫描便停止。

PLC 运行正常时,扫描周期的长短与 CPU 的运算速度、与 I/O 点的情况、与用户应用程序的长短及编程情况等有关。通常用 PLC 执行 1 KB 指令所需时间来说明其扫描速度(一般 1~10 ms/KB)。值得注意的是,不同指令其执行时间是不同的,从零点几  $\mu\text{s}$  到上百  $\mu\text{s}$  不等,故选用不同指令所用的扫描时间将会不同。若用于高速系统要缩短扫描周期时,可从软硬件上同时考虑。

### 3.7.2 PLC 工作过程的中心内容

上面已经说明,PLC 是按图 3-3 所示的运行框图进行工作的,当 PLC 处于正常运行时,它将不断重复图中的扫描过程,不断循环扫描地工作下去。分析上述扫描过程,如果对远程 I/O 特殊模块和其他通信服务暂不考虑,这样扫描过程就只剩下“输入采样”、“程序执行”和“输出刷新”三个阶段了。这三个阶段是 PLC 工作过程的中心内容,理解透 PLC 工作过程的这三个阶段是学习好 PLC 的基础。下面就对这三个阶段进行详细的分析,PLC 典型的扫描周期如图 3-4 所示(不考虑立即输入、立即输出情况)。

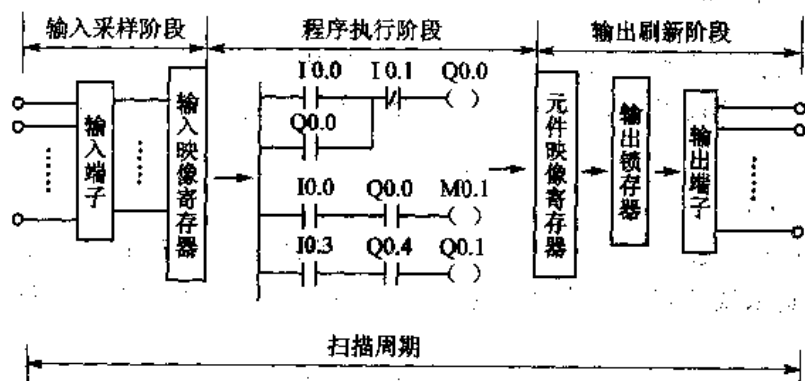


图 3-4 PLC 扫描工作过程

#### (1) 输入采样阶段

PLC 在输入采样阶段,首先扫描所有输入端子,并将各输入状态存入相对应的输入映像寄存器中。此时,输入映像寄存器被刷新。接着,进入程序执行阶段,在此阶段和输出刷新阶段,输入映像寄存器与外界隔离,无论输入信号如何变化,其内容保持不变,直到下一个扫描周期的输入采样阶段,才重新写入输入端的新内容。所以一般来说,输入信号的宽度要大于一个扫描周期,否则很可能造成信号的丢失。

#### (2) 程序执行阶段

根据 PLC 梯形图程序扫描原则,一般来说,PLC 按从左到右、从上到下的步骤顺序执行程序。当指令中涉及输入、输出状态时,PLC 就从输入映像寄存器中“读入”对应输入端子状态,从元件映像寄存器“读入”对应元件(“软继电器”)的当前状态。然后,进行相应的运算,运算结果再存入元件映像寄存器中。对元件映像寄存器来说,每一个元件(“软继电器”)的状态会随着程序执行过程而变化。

### (3) 输出刷新阶段

在所有指令执行完毕后,元件映像寄存器中所有输出继电器的状态(接通/断开)在输出刷新阶段转存到输出锁存器中,通过一定方式输出,最后经过输出端子驱动外部负载。

### 3.7.3 PLC 对输入/输出的处理原则

根据上述工作特点,可以归纳出 PLC 在输入/输出处理方面必须遵守的一般原则:

(1) 输入映像寄存器的数据取决于输入端子板上各输入点在上一刷新期间的接通和断开状态。

(2) 程序执行结果取决于用户所编程序和输入/输出映像寄存器的内容及其他各元件映像寄存器的内容。

(3) 输出映像寄存器的数据取决于输出指令的执行结果。

(4) 输出锁存器中的数据,由上一次输出刷新期间输出映像寄存器中的数据决定。

(5) 输出端子的接通和断开状态,由输出锁存器决定。

## 3.8 PLC 的编程语言和程序结构

### 3.8.1 PLC 的编程语言

PLC 为用户提供了完整的编程语言,以适应编制用户程序的需要。PLC 提供的编程语言通常有以下几种:梯形图、语句表、功能图和功能块图。下面以 S7-200 系列 PLC 为例加以说明。

#### 1. 梯形图(LAD)

梯形图(Ladder)编程语言是从继电器控制系统原理图的基础上演变而来的。PLC 的梯形图与继电器控制系统梯形图的基本思想是一致的,只是在使用符号和表达方式上有一定区别。

图 3-5 是典型的梯形示意图。左右两条垂直的线称为母线。母线之间是触点的逻辑连接和线圈的输出。

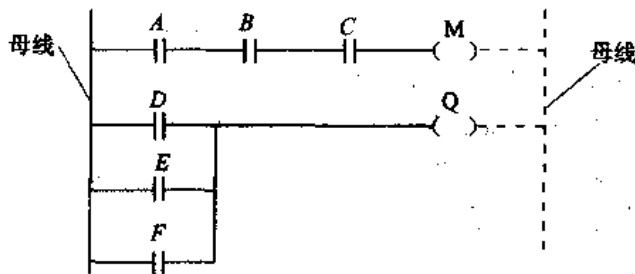


图 3-5 梯形图举例

梯形图的一个关键概念是“能流”(Power Flow),这只是概念上的“能流”。图 3-5 中,把左边的母线假想为电源“火线”,而把右边的母线(虚线所示)假想为电源“零线”。如果有“能

流”从左至右流向线圈,则线圈被激励。如没有“能流”,则线圈未被激励。

“能流”可以通过被激励(ON)的常开接点和未被激励(OFF)的常闭接点自左向右流。“能流”在任何时候都不会通过接点自右向左流。如图 3-5 中,当 A、B、C 接点都接通后,线圈 M 才能接通(被激励),只要其中一个接点不接通,线圈就不会接通;而 D、E、F 接点中任何一个接通,线圈 Q 就被激励。

要强调指出的是,引入“能流”的概念,仅仅是为了和继电接触器控制系统相比较,来对梯形图有一个深入的认识,其实“能流”在梯形图中是不存在的。

有的 PLC 的梯形图有两根母线,但大部分 PLC 现在只保留左边的母线了。在梯形图中,触点代表逻辑“输入”条件,如开关、按钮和内部条件等;线圈通常代表逻辑“输出”结果,如灯、电机接触器、中间继电器等。对 S7-200 PLC 来说,还有一种输出——“盒”,它代表附加的指令,如定时器、计数器和功能指令等,以后学习到指令时会有详细的介绍。

梯形图语言简单明了,易于理解,是所有编程语言的首选。

## 2. 语句表(STL)

语句表(Statements List)类似于计算机中的助记符语言,它是 PLC 最基础的编程语言。所谓语句表编程,是用一个或几个容易记忆的字符来代表 PLC 的某种操作功能。具体指令的说明在后面的章节有详细的介绍。

图 3-6 是一个简单的 PLC 程序,图(a)是梯形图程序,图(b)是相应的语句表。一般来说,语句表编程适合于熟悉 PLC 和有经验的程序员使用。

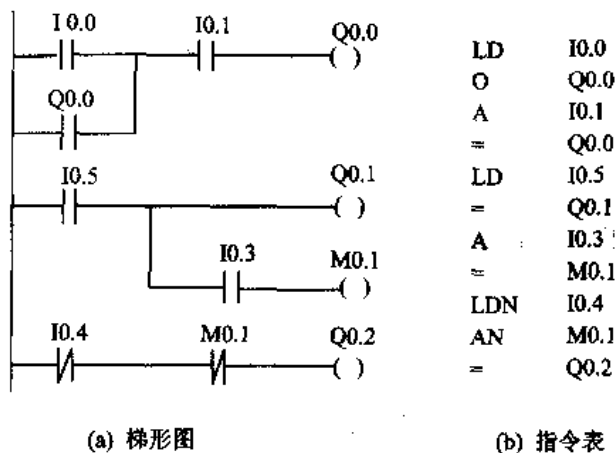


图 3-6 LAD 和 STL 应用举例

## 3. 顺序功能流程图(SFC)

顺序功能流程图(Sequence Function Chart)编程是一种图形化的编程方法,亦称功能图。使用它可以对具有并发、选择等复杂结构的系统进行编程,许多 PLC 都提供了用于 SFC 编程的指令。本书的第六章对这种编程方法有详细介绍。目前国际电工协会(IEC)也正在实施并发展这种语言的编程标准。

## 4. 功能块图(FBD)

S7-200 的 PLC 专门提供了功能块图(Function Block Diagrams)编程语言,利用 FBD 可以看到像普通逻辑门图形的逻辑盒指令。它没有梯形图编程器中的触点和线圈,但有与之等价的指令,这些指令是作为盒指令出现的,程序逻辑由这些盒指令之间的连接决定。也就是说,一个

指令(例如 AND 盒)的输出可以用来允许另一条指令(例如定时器),这样可以建立所需要的控制逻辑。这样的连接思想可以解决范围广泛的逻辑问题。FBD 编程语言有利于程序流的跟踪,但在目前使用较少。图 3-7 为 FBD 的一个简单使用例子。

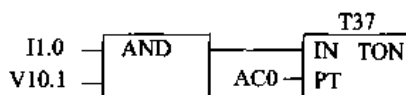


图 3-7 FBD 简单举例

### 3.8.2 PLC 的程序结构

控制一个任务或过程,是通过在 RUN 方式下,使主机循环扫描并连续执行用户程序来实现的,用户程序决定了一个控制系统的功能。程序的编制可以使用编程软件在计算机或其他专用编程设备中进行(如图形输入设备),也可使用手编器。

广义上的 PLC 程序由三部分构成:用户程序、数据块和参数块。

#### 1. 用户程序

用户程序是必选项。用户程序在存储器空间中也称为组织块,它处于最高层次,可以管理其他块,它是用各种语言(如 STL、LAD 或 FBD 等)编写的用户程序。不同机型的 CPU 其程序空间容量也不同。用户程序的结构比较简单,一个完整的用户控制程序应当包含一个主程序、若干子程序和若干中断程序三大部分。不同编程设备,对各程序块的安排方法也不同。程序结构示意图如图 3-8 所示。

#### 2. 数据块

数据块为可选部分,它主要存放控制程序运行所需的数据。

#### 3. 参数块

参数块也是可选部分,它存放的是 CPU 组态数据,如果在编程软件或其他编程工具上未进行 CPU 的组态,则系统以默认值进行自动配置。

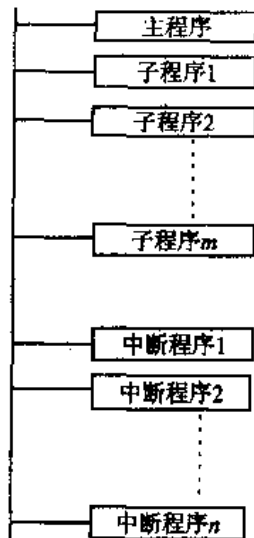


图 3-8 程序结构

## 本章小结

PLC 作为一种工业标准设备,虽然生产厂家众多,产品种类层出不穷,但它们都具有相同的工作原理,使用方法也大同小异。

(1) PLC 是计算机技术与继电接触式控制技术相结合的产物。它专为在工业环境下应用而设计,可靠性高,使用方便,应用广泛。

PLC 功能的不断增强,使 PLC 的应用领域不断扩大和延伸,应用方式也更加丰富。

PLC 从结构上可分为整体式和模块式;从容量上可分为小型、中型和大型。

(2) PLC 的组成部件有中央处理器(CPU)、存储器、输入/输出(I/O)接口和电源等。

(3) PLC 按集中采样、集中输出,按顺序周期性循环扫描用户程序的方式工作。

当 PLC 处于正常运行时,它将不断重复扫描过程,其工作过程的中心内容分为输入采样、程序执行和输出刷新三个阶段。

(4) PLC 是为取代继电接触式控制系统而产生的,因而两者存在着一定的联系。PLC 与继电接触式控制系统具有相同的逻辑关系,但 PLC 使用的是计算机技术,其逻辑关系用程序实现,而不是实际电路。

(5) 可用多种形式的编程语言来编写 PLC 的用户程序,而梯形图和语句表是两种最常用的编程语言。

### 思考题

1. PLC 有什么特点?
2. PLC 与继电接触式控制系统相比有哪些异同?
3. 构成 PLC 的主要部件有哪些? 各部分主要作用是什么?
4. PLC 是按什么样的工作进行工作的? 它的中心工作过程分哪几个阶段? 在每个阶段主要完成些什么控制任务?
5. 一般来说,PLC 对输入信号有什么要求?

## 第四章 S7-200 系列 PLC 的 硬件系统及内部资源

S7-200 系列 PLC 可以满足多种多样的自动化控制的需要。由于它具有紧凑的设计、良好的扩展性、低廉的价格、丰富的功能模块以及强大的指令系统,使得 S7-200 PLC 可以近乎完美地满足小规模的控制要求。本章主要对 S7-200 系列 PLC 的硬件系统及内部资源作一介绍,内容包括:

- S 系列 PLC 发展简介;
- S7-200 PLC 的硬件系统;
- S7-200 PLC 的寻址方式。

### 4.1 概 述

S7-200 PLC 是德国西门子公司生产的一种小型 PLC,其许多功能达到大、中型 PLC 的水平,而价格却和小型 PLC 的一样,因此,它一经推出,即受到了广泛的关注。特别是 S7-200 CPU22 \* 系列 PLC(它是 CPU21 \* 系列的替代产品),由于它具有多种功能模块和人机界面(HMI)可供选择,所以系统的集成非常方便,并且可以很容易地组成 PLC 网络。同时它具有功能齐全的编程和工业控制组态软件,使得在完成控制系统的设计时更加简单,几乎可以完成任何功能的控制任务。

S7 系列 PLC 还有 S7-300 和 S7-400 系列,它们分别为中大型 PLC,完全可以替代西门子早期的 S5 系列 PLC。S7 系列 PLC 的编程均使用 STEP7 编程语言。考虑到西门子公司的产品在中国应用非常广泛,其功能比较全面和典型,具有一定的代表性,因此本节以 S7-CPU22 \* 系列为例,详细介绍 S7-200 系列 PLC 的硬件系、内部元器件和寻址方式等。

需要说明的是,即使是 S7-CPU22 \* 系列 PLC,它们的功能也还在不断地提高和改进中,最新的 CPU 版本是 V1.21(2002 年年末),在这个版本中,加上了字符串处理功能。

### 4.2 S7-200 系列 PLC 的硬件系统

SIMATIC S7-200 系列是西门子公司前几年刚刚投入市场的小型可编程序控制器,可以单机运行,也可以进行输入/输出和功能模块的扩展。它价格低廉,结构小巧,可靠性高,运行速度快,继承和发挥了它在大、中型 PLC 领域的技术优势,有极丰富的指令集,具有强大的多种集成功能和实时特性,其性能价格比非常高,所以它在各行各业中的应用得到迅速推广,在规模不太大的控制领域是较为理想的控制设备。

### 4.2.1 硬件系统基本构成

S7-200 PLC 硬件系统的配置方式采用整体式加积木式,即主机中包含一定数量的输入/输出(I/O)点,同时还可以扩展 I/O 模块和各种功能模块。

一个完整的系统组成如图 4-1 所示。

① 基本单元 基本单元(Basic Unit)有时又称做 CPU 模块,也有的称之为主机或本机。它包括 CPU、存储器、基本输入/输出点和电源等,是 PLC 的主要部分。实际上它就是一个完整的控制系统,可以单独完成一定的控制任务。

② 扩展单元 主机 I/O 点数量不能满足控制系统的要求时,用户可以根据需要扩展各种 I/O 模块,所能连接的扩展单元的数量和实际所能使用的 I/O 点数是由多种因素共同决定的。

③ 特殊功能模块 当需要完成某些特殊功能的控制任务时,需要扩展功能模块。它们是完成某种特殊控制任务的一些装置。

④ 相关设备 相关设备是为充分和方便地利用系统的硬件和软件资源而开发和使用的一些设备,主要有编程设备、人机操作界面和网络设备等。

⑤ 工业软件 工业软件是为更好地管理和使用这些设备而开发的与之相配套的程序,它主要由标准工具、工程工具、运行软件和人机接口软件等几大类构成。

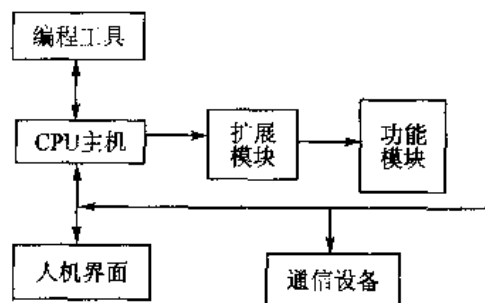


图 4-1 S7-200 PLC 系统组成

### 4.2.2 主机结构及性能特点

#### 1. 主机外形

CPU 22 \* 系列 PLC 主机(CPU 模块)的外形如图 4-2 所示。S7-200 的 CPU 模块包括一个中央处理单元、电源以及数字 I/O 点,这些都被集成在一个紧凑、独立的设备中。CPU 负责执行程序,输入部分从现场设备中采集信号,输出部分则输出控制信号,驱动外部负载。

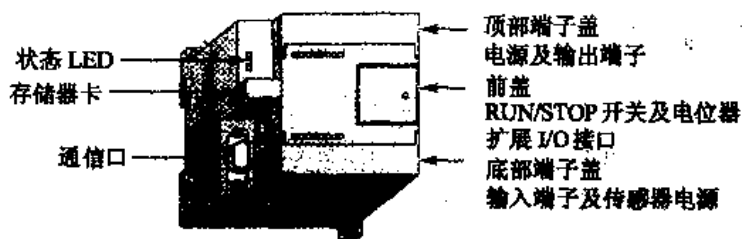


图 4-2 CPU22 \* 系列 PLC 的 CPU 外形图

从 CPU 模块的功能来看,S7-200 系列小型可程序控制器发展至今,经历了两代产品。

第一代产品的 CPU 模块为 CPU 21 \*, 主机都可进行扩展,它具有四种不同结构配置的 CPU 单元,即 CPU 212、CPU 214、CPU 215 和 CPU 216,现在已经停止生产。

第二代产品的 CPU 模块为 CPU 22 \*, 是在 21 世纪初投放市场的。其速度快,具有极强

的通信能力。它具有如下五种不同结构配置的 CPU 单元。

(1) CPU 221 它有 6 输入/4 输出, I/O 共计 10 点, 无扩展能力, 程序和数据存储容量较小, 有一定的高速计数处理能力, 非常适合于少点数的控制系统。

(2) CPU 222 它有 8 输入/6 输出, I/O 共计 14 点。和 CPU 221 相比, 它可以进行一定模拟量的控制和 2 个模块的扩展, 因此是应用更广泛的全功能控制器。

(3) CPU 224 它有 14 输入/10 输出, I/O 共计 24 点。和前两者相比, 存储容量扩大了一倍, 它可以有 7 个扩展模块, 有内置时钟, 它有更强的模拟量和高速计数的处理能力, 是使用得最多的 S7-200 产品。

(4) CPU 226 它有 24 输入/16 输出, I/O 共计 40 点, 和 CPU 224 相比, 增加了通信口的数量, 通信能力大大增强。它可用于点数较多、要求较高的小型或中型控制系统。

(5) CPU 226XM 这是西门子公司后来推出的一种增强型主机, 它在用户程序存储容量和数据存储容量上进行了扩展, 其他指标和 CPU 226 相同。

## 2. CPU 的主要特点和技术规范

S7-200 PLC 的电源电压有 (20.4~28.8)VDC 和 (85~264)VAC 两种, 主机上还集成了 24 V 直流电源, 可以直接用于连接传感器和执行机构。它的输出类型有晶体管 (DC)、继电器 (DC/AC) 两种输出方式。它可以用普通输入端子捕捉比 CPU 扫描周期更快的脉冲信号, 实现高速计数。2 路最大可达 20 kHz 的高频脉冲输出, 可用以驱动步进电机和伺服电机以实现准确定位任务。可以用模块上的电位器来改变它对应的特殊寄存器中的数值, 可以实时更改程序运行中的一些参数, 如定时器/计数器的设定值、过程量的控制参数等。实时时钟可用于对信息加注时间标记, 记录机器运行时间或对过程进行时间控制。

附录 B 中列出了 S7-200 PLC CPU 的主要技术规范, 包括 CPU 规范、CPU 输入规范和 CPU 输出规范。CPU 的技术数据对了解 PLC 的性能和进行 PLC 选择非常有用。

## 3. 存储系统

S7-200 系列 PLC 的存储系统由 RAM 和 EEPROM 两种类型存储器构成, CPU 模块内部配备一定容量的 RAM 和 EEPROM, 同时, CPU 模块支持可选的 EEPROM 存储器卡。超级电容和电池模块用于长时间保存数据, 用户数据可通过主机的超级电容存储若干天; 电池模块可选, 可使数据的存储时间延长到 200 天。各 CPU 的存储容量如附录 B 中表 B-4 所示。

### 4.2.3 输入/输出的扩展

当 CPU 的 I/O 点数不够用或需要进行特殊功能的控制时, 就要进行 I/O 的扩展。I/O 扩展包括 I/O 点数的扩展和功能模块的扩展。不同的 CPU 有不同的扩展规范, 它主要受 CPU 的功能限制。大家在使用时可参考 SIEMENS 的系统手册。

#### 1. I/O 扩展模块

用户可以使用主机 I/O 和扩展 I/O 模块。S7-200 系列 CPU 提供一定数量的主机数字量 I/O 点, 但在主机 I/O 点数不够的情况下, 就必须使用扩展模块的 I/O 点。

典型的数字量输入/输出扩展模块有:

输入扩展模块 EM221 有两种: 8 点 DC 输入、8 点 AC 输入。

输出扩展模块 EM222 有三种: 8 点 DC 晶体管输出, 8 点 AC 输出、8 点继电器输出。

输入/输出混合扩展模块 EM223 有六种: 分别为 4 点 (8 点、16 点) DC 输入/4 点 (8 点、16

点)DC 输出、4 点(8 点、16 点)DC 输入/4 点(8 点、16 点)继电器输出。

## 2. 功能扩展模块

当需要完成某些特殊功能的控制任务时,CPU 主机可以扩展特殊功能模块。如要求进行 PROFIBUS-DP 现场总线连接时,就需要 EM277 PROFIBUS-DP 模块。

典型的特殊功能模块有:

### (1) 模拟量输入/输出扩展模块

模拟量输入扩展模块 EM231 有 3 种:4 路模拟量输入、2 路热电阻输入和 4 路热电偶输入。

模拟量输出扩展模块 EM232 具有 2 路模拟量输出。

模拟量输入/输出扩展模块 EM235 具有 4 路模拟量输入/1 路模拟量输出。

### (2) 特殊功能模块

功能模块有 EM235 位控模块、EM277 PROFIBUS-DP 模块、EM241 调制解调器模块、CP243-1 以太网模块、CP243-2 AS-i 接口模块等。

功能模块性能的讲解请参见最新的 S7-200 PLC 的系统手册或本书以后的有关章节。

## 3. I/O 点数扩展和编址

CPU 22 \* 系列的每种主机所提供的本机 I/O 点的 I/O 地址是固定的,进行扩展时,可以在 CPU 右边连接多个扩展模块,每个扩展模块的组态地址编号取决于各模块的类型和该模块在 I/O 链中所处的位置。编址方法是同种类型输入或输出点的模块在链中按与主机的位置而递增,其他类型模块的有无以及所处的位置不影响本类型模块的编号。

例如,某一控制系统选用 CPU 224,系统所需的输入输出点数各为:数字量输入 24 点、数字量输出 20 点、模拟量输入 6 点和模拟量输出 2 点。

本系统可有多种不同模块的选取组合,并且各模块在 I/O 链中的位置排列方式也可能有多种,图 4-3 所示为其中的一种模块连接形式。表 4-1 所列为其对应的各模块的编址情况。

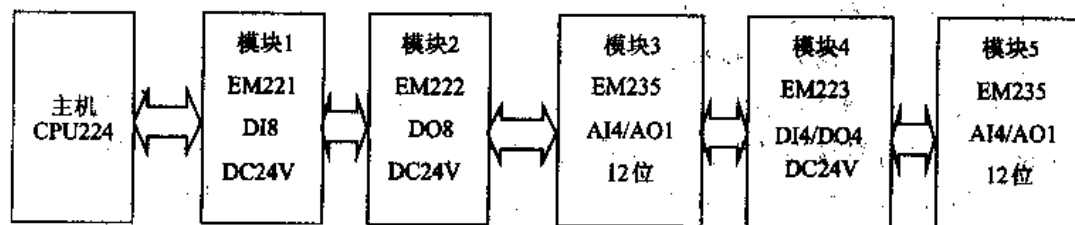


图 4-3 模块连接方式

表 4-1 各模块编址

主机 I/O		模块 1 I/O	模块 2 I/O	模块 3 I/O	模块 4 I/O	模块 5 I/O
I0.0	Q0.0	I2.0	Q2.0	AIW0 AQW0	I3.0 Q3.0	AIW8 AQW2
I0.1	Q0.1	I2.1	Q2.1	AIW2	I3.1 Q3.1	AIW10
I0.2	Q0.2	I2.2	Q2.2	AIW4	I3.2 Q3.2	AIW12
I0.3	Q0.3	I2.3	Q2.3	AIW6	I3.3 Q3.3	AIW14
I0.4	Q0.4	I2.4	Q2.4			
I0.5	Q0.5	I2.5	Q2.5			
I0.6	Q0.6	I2.6	Q2.6			
I0.7	Q0.7	I2.7	Q2.7			

续表 4-1

主机 I/O	模块 1 I/O	模块 2 I/O	模块 3 I/O	模块 4 I/O	模块 5 I/O
I1.0    Q1.0					
I1.1    Q1.1					
I1.2					
I1.3					
I1.4					
I1.5					

由此可见, S7-200 系统扩展对输入/输出的组态规则为:

- ① 同类型输入或输出点的模块进行顺序编址。
- ② 对于数字量, 输入/输出映像寄存器的单位长度为 8 位(1 个字节), 本模块高位实际位数未满足 8 位的, 未用位不能分配给 I/O 链的后续模块。
- ③ 对于模拟量, 输入/输出以 2 个字节(1 个字)递增方式来分配空间。

### 4.3 S7-200 系列 PLC 的内部资源及寻址方式

#### 4.3.1 软元件

##### 1. 软元件(软继电器)

用户使用的 PLC 中的每一个输入/输出、内部存储单元、定时器和计数器等都称为软元件。各元件有其不同的功能, 有固定的地址。软元件的数量决定了可编程控制器的规模和数据处理能力, 每一种 PLC 的软元件是有限的。

软元件是 PLC 内部的具有一定功能的器件, 这些器件实际上是由电子电路和寄存器及存储器单元等组成。例如, 输入继电器是由输入电路和输入映像寄存器构成; 输出继电器是由输出电路和输出映像寄存器构成; 定时器和计数器也都是由特定功能的寄存器构成。它们都具有继电器特性, 但没有机械性的触点。为了把这种元器件与传统电气控制电路中的继电器区别开来, 我们把它称为软元件或软继电器。这些软继电器的最大特点是其触点(包括常开触点和常闭触点)可以无限次使用, 并且它们的寿命长。

编程时, 用户只需要记住软元件的地址即可。每一软元件都有一个地址与之相对应, 软元件的地址编排采用区域号加区域内编号的方式。即 PLC 内部根据软元件的功能不同, 分成了许多区域, 如输入/输出继电器区、定时器区、计数器区、特殊继电器区等, 分别用 I、Q、T、C、SM 等来表示。

##### 2. 软元件介绍

###### (1) 输入继电器(I)

输入继电器一般都有一个 PLC 的输入端子与之对应, 它用于接收外部的开关信号。当外部的开关信号闭合, 则输入继电器的线圈得电, 在程序中其常开触点闭合, 常闭触点断开。这些触点可以在编程时任意使用, 使用次数不受限制。

在每个扫描周期的开始, PLC 对各输入点进行采样, 并把采样值送到输入映像寄存器。PLC 在接下来的本周期各阶段不再改变输入映像寄存器中的值, 直到下一个扫描周期的输入

采样阶段。

PLC 输入映像寄存器区的大小见表 B-4。实际输入点数不能超过这个数量,未用的输入映像区可以做其他编程元件使用,如可以当通用辅助继电器或数据寄存器,但这只有在寄存器的整个字节的所有位都未占用的情况下才可做他用,否则会出现错误的执行结果。

#### (2) 输出继电器(Q)

输出继电器一般都有一个 PLC 上的输出端子与之对应。当通过程序使得输出继电器线圈得电时,PLC 上的输出端开关闭合,它可以作为控制外部负载的开关信号。同时在程序中其常开触点闭合,常闭触点断开。这些触点可以在编程时任意使用,使用次数不受限制。

在每个扫描周期的输入采样、程序执行等阶段,并不把输出结果信号直接送到输出继电器,而只是送到输出映像寄存器,只有在每个扫描周期的末尾才将输出映像寄存器中的结果几乎同时送到输出锁存器,对输出点进行刷新。实际未用的输出映像区可做他用,用法与输入继电器相同。

#### (3) 通用辅助继电器(M)

通用辅助继电器的作用和继电器接触器控制系统中的中间继电器相同,它在 PLC 中没有输入/输出端与之对应,因此它的触点不能驱动外部负载。这是与输出继电器的主要区别。它主要起逻辑控制作用。

#### (4) 特殊继电器(SM)

有些辅助继电器具有特殊功能或用来存储系统的状态变量、有关的控制参数和信息,我们称其为特殊继电器。用户可以通过特殊标志来沟通 PLC 与被控对象之间的信息,如可以读取程序运行过程中的设备状态和运算结果信息,利用这些信息实现一定的控制动作。用户也可通过直接设置某些特殊继电器位来使设备实现某种功能。例如:

SM0.1 首次扫描为 1,以后为 0,常用来对程序进行初始化,属只读型;

SM1.2 当机器执行数学运算的结果为负时,该位被置 1,属只读型;

SMB28 和 SMB29 分别存储模拟调节器 0 和 1 的输入值,CPU 每次扫描时更新该值,属只读型。

常用特殊继电器的功能参见表 B-6。更多的 SM 的使用参见第八章和第九章。

#### (5) 变量存储器(V)

变量存储器用来存储变量。它可以存放程序执行过程中控制逻辑操作的中间结果,也可以使用变量存储器来保存与工序或任务相关的其他数据。在进行数据处理时,变量存储器会被经常使用。

#### (6) 局部变量存储器(L)

局部变量存储器用来存放局部变量。局部变量与变量存储器所存储的全局变量十分相似,主要区别在于全局变量是全局有效的,而局部变量是局部有效的。全局有效是指同一个变量可以被任何程序(包括主程序、子程序和中断程序)访问;而局部有效是指变量只和特定的程序相关联。

S7-200 PLC 提供 64 个字节的局部存储器,其中 60 个可以作暂时存储器或给子程序传递参数。主程序、子程序和中断程序都有 64 个字节的局部存储器可以使用。不同程序的局部存储器不能互相访问。机器在运行时,根据需要动态地分配局部存储器,在执行主程序时,分配给子程序或中断程序的局部变量存储区是不存在的,当子程序调用或出现中断时,需要为之

分配局部存储器,新的局部存储器可以是曾经分配给其他程序块的同一个局部存储器。

#### (7) 顺序控制继电器(S)

有些 PLC 中也把顺序控制继电器称为状态器。顺序控制继电器用在顺序控制或步进控制中。有关顺序控制继电器的使用请阅读第六章。

#### (8) 定时器(T)

定时器是可编程序控制器中重要的编程元件,是累计时间增量的内部器件。电气自动控制的大部分领域都需要用定时器进行时间控制,灵活地使用定时器可以编制出复杂动作的控制程序。

定时器的工作过程与继电接触式控制系统的时间继电器基本相同,但它没有瞬动触点。使用时要提前输入时间预设值。当定时器的输入条件满足时开始计时,当前值从 0 开始按一定的时间单位增加;当定时器的当前值达到预设值时,定时器触点动作,利用定时器的触点就可以得到控制所需的延时时间。

#### (9) 计数器(C)

计数器用来累计输入脉冲的个数,经常用来对产品进行计数或进行特定功能的编程。使用时要提前输入它的设定值(计数的个数)。当输入触发条件满足时,计数器开始累计它的输入端脉冲电位上升沿(正跳变)的次数;当计数器计数达到预定的设定值时,其常开触点闭合,常闭触点断开。

#### (10) 模拟量输入映像寄存器(AI)、模拟量输出映像寄存器(AQ)

模拟量输入电路用以实现模拟量/数字量(A/D)之间的转换,而模拟量输出电路用以实现数字量/模拟量(D/A)之间的转换。

在模拟量输入/输出映像寄存器中,数字量的长度为 1 个字长(16 位),且从偶数号字节进行编址来存取转换过的模拟量值,如 0、2、4、6、8 等。编址内容包括元件名称、数据长度和起始字节的地址,如:AIW6,AQW12 等。

PLC 对这两种寄存器的存取方式不同的是,模拟量输入寄存器只能进行读取操作,而对模拟量输出寄存器只能进行写入操作。

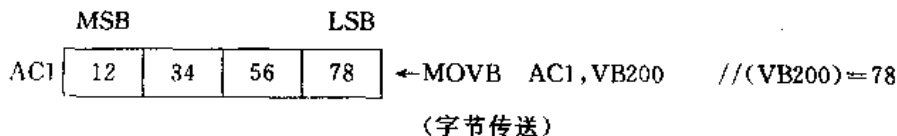
#### (11) 高速计数器(HC)

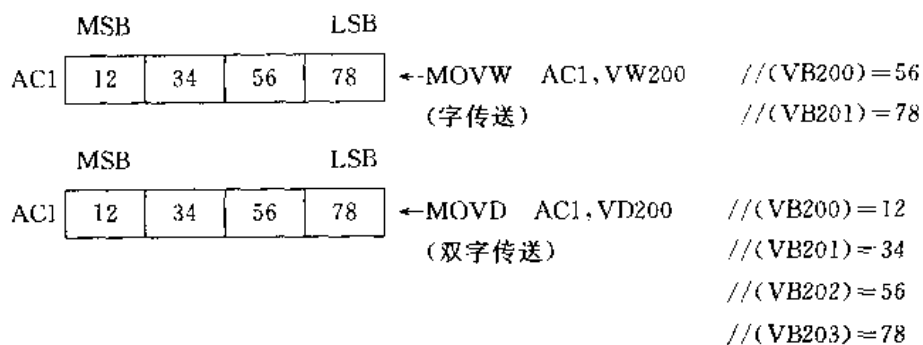
高速计数器的工作原理与普通计数器基本相同,它用来累计比主机扫描速率更快的高速脉冲。高速计数器的当前值是一个双字长(32 位)的整数,且为只读值。高速计数器的数量很少,编址时只用名称 HC 和编号,如:HC2。

高速计数器的编程使用比较复杂,在第七章将详细介绍。

#### (12) 累加器(AC)

S7-200 PLC 提供 4 个 32 位累加器,分别为 AC0、AC1、AC2、AC3。累加器(AC)是用来暂存数据的寄存器。它可以用来存放数据如运算数据、中间数据和结果数据,也可用来向子程序传递参数,或从子程序返回参数。使用时只表示出累加器的地址编号,如 AC0。累加器可进行读、写两种操作。累加器的可用长度为 32 位,数据长度可以是字节、字或双字,但实际应用时,数据长度取决于进出累加器的数据类型。具体使用分别如下所示:





### 4.3.2 CPU 存储区域的直接寻址

#### 1. 数据类型

##### (1) 数据类型及范围

S7-200 系列 PLC 的数据类型可以是字符串、布尔型(0 或 1)、整型和实型(浮点数)。实数采用 32 位单精度数来表示,数据类型、长度及范围如表 4-2 所列。

表 4-2 数据类型、长度及范围

基本数据类型	无符号整数表示范围		基本数据类型	有符号整数表示范围	
	十进制表示	十六进制表示		十进制表示	十六进制表示
字节 B(8 位)	0~255	0~FF	字节 B(8 位) 只用于 SHRB 指令	-128~127	80~7F
字 W(16 位)	0~65 535	0~FFFF	INT(16 位)	-32 768~32 767	8000~7FFF
双字 D(32 位)	0~4 294 967 295	0~FFFFFFFF	DINT(32 位)	-2 147 483 648~ 2 147 483 647	80000000~ 7FFFFFFF
BOOL(1 位)	0~1				
字符串	每个字符以字节形式存储,最大长度为 255 个字节,第一个字节中定义该字符串的长度				
实数(32 位)	-10 <sup>38</sup> ~10 <sup>38</sup> (IEEE32 浮点数)				

##### (2) 常数

在编程中经常会使用常数。常数数据长度可为字节、字和双字。在机器内部的数据都以二进制存储,但常数的书写可以用二进制、十进制、十六进制、ASCII 码或浮点数(实数)等多种形式。几种常数形式分别如表 4-3 所列。

注意:表中的#为常数的进制格式说明符,如果常数无任何格式说明符,则系统默认为十进制数。

表 4-3 常数表示方法

进 制	书写格式	举 例
十进制	十进制数值	1 052
十六进制	16#十六进制值	16#8AC6
二进制	2#二进制值	2#1010_0011_1101_0001

续表 4-3

进 制	书写格式	举 例
ASCII 码	'ASCII 码文本'	'Show terminals'
浮点数	ANSI/IEEE 754—1985 标准	(正数) - 1.175495E - 38 到 +3.402823E + 38 (负数) - 1.175495E - 38 到 -3.402823E + 38

## 2. 直接寻址

S7-200 PLC 的存储单元按字节进行编址,无论所寻址的是何种数据类型,通常应指出它所在存储区域内的字节地址。每个单元都有惟一的地址,这种直接指出元件名称的寻址方式称为直接寻址。S7-200 软元件的直接寻址的符号如表 4-4 所列。

表 4-4 S7-200 PLC 元件名称及直接编址格式

元件符号(名称)	所在数据区域	位寻址格式	其他寻址格式
I(输入继电器)	数字量输入映像区	Ax.y	ATx
Q(输出继电器)	数字量输出映像区	Ax.y	ATx
M(通用辅助继电器)	内部存储器区	Ax.y	ATx
SM(特殊继电器)	特殊存储器区	Ax.y	ATx
S(顺序控制继电器)	顺序控制继电器存储器区	Ax.y	ATx
V(变量存储器)	变量存储器区	Ax.y	ATx
L(局部变量存储器)	局部存储器区	Ax.y	ATx
T(定时器)	定时器存储器区	Ax	Ax(仅字)
C(计数器)	计数器存储器区	Ax	Ax(仅字)
AI(模拟量输入映像寄存器)	模拟量输入存储器区	无	Ax(仅字)
AQ(模拟量输出映像寄存器)	模拟量输出存储器区	无	Ax(仅字)
AC(累加器)	累加器区	无	Ax(任意)
HC(高速计数器)	高速计数器区	无	Ax(仅双字)

表 4-4 中:

A:元件名称,即该数据在数据存储器中的区域地址,可以是表 4-4 中的元件符号;

T:数据类型,若为位寻址,则无该项;若为字节、字或双字寻址,则 T 的取值应分别为 B、W 和 D;

x:字节地址;

y:字节内的位地址,只有位寻址才有该项。

### (1) 位寻址格式

按位寻址时的格式为:Ax.y,使用时必须指定元件名称、字节地址和位号,如图 4-4 是输入继电器(I)的位寻址格式举例。

可以进行这种方式位寻址的编程元件有:输入继电器(I)、输出继电器(Q)、通用辅助继电器(M)、特殊继电器(SM)、局部变量存储器(L)、变量存储器(V)和顺序控制继电器(S)。

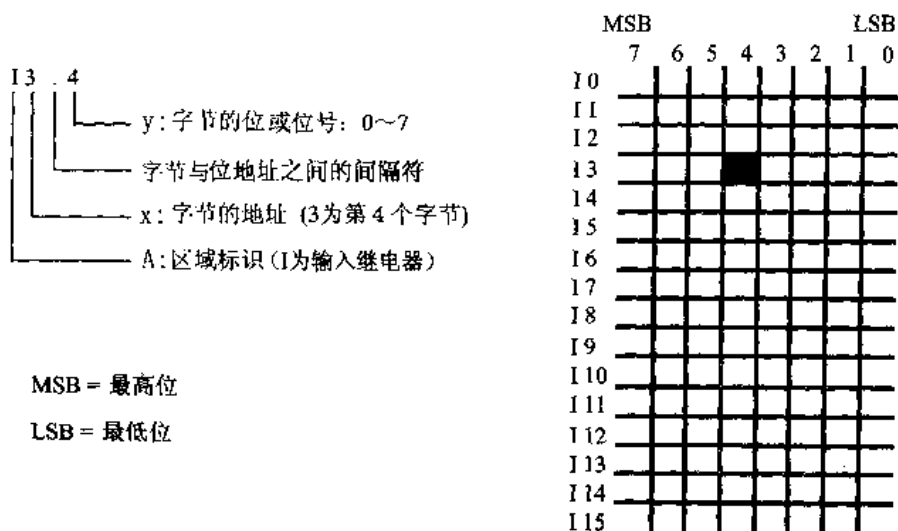


图 4-4 CPU 存储器中位数据表示方法举例(位寻址)

## (2) 特殊器件的寻址格式

存储区内另有一些元件是具有一定功能的器件,不用指出它们的字节,而是直接写出其编号。这类元件包括定时器(T)、计数器(C)、高速计数器(HC)和累加器(AC)。其中 T 和 C 的地址编号中均包含两个含义,如 T10,既表示 T10 的定时器位状态信息,又表示该定时器的当前值。

累加器(AC)的数据长度可以是字节、字或双字。使用时只表示出累加器的地址编号,如 AC0,数据长度取决于进出 AC0 的数据类型。

## (3) 字节、字和双字的寻址格式

对字节、字和双字数据,直接寻址时需指明元件名称、数据类型和存储区域内的首字节地址。如图 4-5 是以变量存储器(V)为例分别存取 3 种长度数据的比较。

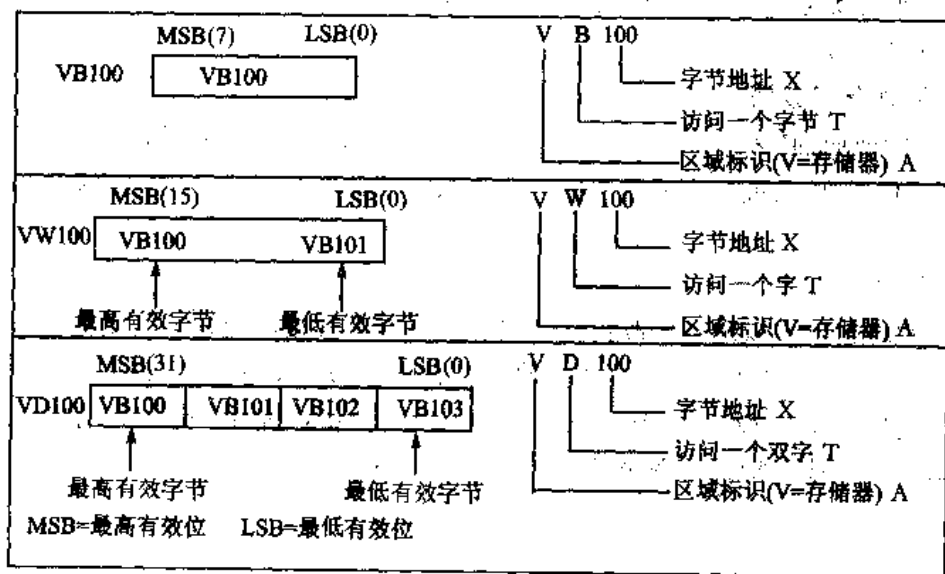


图 4-5 存取 3 种长度数据的比较

可以用此方式进行寻址的元件有输入继电器(I)、输出继电器(Q)、通用辅助继电器(M)、特殊标志继电器(SM)、局部变量存储器(L)、变量存储器(V)、顺序控制继电器(S)、模拟量输入映像寄存器(AI)和模拟量输出映像寄存器(AQ)。

### 4.3.3 CPU 存储区域的间接寻址

在直接寻址方式中,直接使用存储器或寄存器的元件名称和地址编号,根据这个地址可以立即找到该数据。

间接寻址方式是指数据存放在存储器或寄存器中,在指令中只出现所需数据所在单元的内存地址的地址。存储单元地址的地址又称为地址指针。这种间接寻址方式与计算机的间接寻址方式相同。间接寻址在处理内存连续地址中的数据时非常方便,而且可以缩短程序所生成的代码长度,使编程更加灵活。

可以用指针进行间接寻址的存储区有输入继电器(I)、输出继电器(Q)、通用辅助继电器(M)、变量存储器(V)、顺序控制继电器(S)、定时器(T)和计数器(C)。其中 T 和 C 仅仅是当前值可以进行间接寻址,而对独立的位值和模拟量值不能进行间接寻址。

使用间接寻址方式存取数据方法与 C 语言中的指针应用基本相同,其过程如下。

#### 1. 建立指针

使用间接寻址对某个存储器单元读、写时,首先要建立地址指针。指针为双字长,是所要访问的存储单元的 32 位的物理地址。可作为指针的存储区有变量存储器(V)、局部变量存储器(L)和累加器(AC1、AC2、AC3)。必须用双字传送指令(MOVD),将存储器所要访问单元的地址装入用来作为指针的存储器单元或寄存器,装入的是地址而不是数据本身。格式如下:

```
例: MOVD    &VB100, VD204
      MOVD    &VB10, AC2
      MOVD    &C2, LD16
```

其中:“&”为地址符号,它与单元编号结合使用表示所对应单元的 32 位物理地址;VB100 只是一个直接地址编号,并不是它的物理地址。指令中的第二个地址数据长度必须是双字长,如:VD、LD 和 AC 等。

#### 2. 用指针来存取数据

在操作数的前面加“\*”表示该操作数为一个指针。如图 4-6 所示,AC1 为指针,用来存放要访问的操作数的地址。在这个例子中,存于 VB200、VB201 中的数据被传送到 AC0 中去。

#### 3. 修改指针

连续存储数据时,可以通过修改指针后很容易存取其紧接的数据。简单的数学运算指令,如加法、减法、自增和自减等指令可以用来修改指针。在修改指针时,要记住访问数据的长度:存取字节时,指针加 1;存取字时,指针加 2;存取双字时,指针加 4。图 4-6 说明如何建立指针,如何存取数据及修改指针。图中具体指令的含义请参阅第七章。

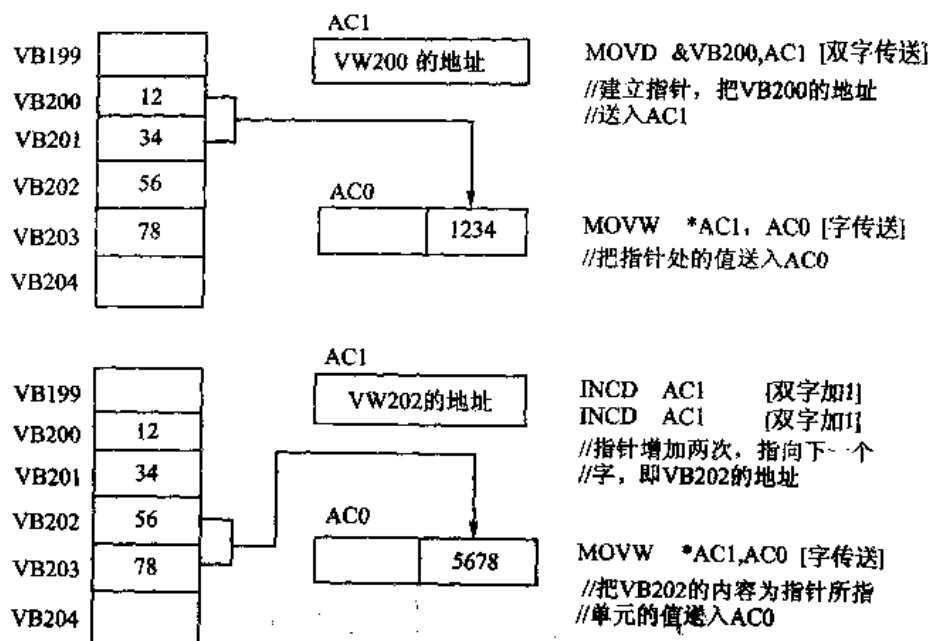


图 4-6 建立指针, 存取数据及修改指针

## 本章小结

本章以 SIMENS S7-200 系列 PLC 为对象, 详细介绍了其硬件结构、软元件及寻址方式。

(1) S7-200 系列 PLC 有五种 CPU 型号, 它们都是整体机, 除 CPU 221 外, 都可以进行 I/O 和功能模块的扩展, 进行 I/O 扩展或特殊功能模块扩展时必须遵循一定的原则。本系列 PLC 在许多方面, 如输入/输出、存储系统、高速处理、实时时钟、网络通信等方面, 具有自己的独特功能。

(2) 附录 B 中的几个表格非常有用, 应学会分析和参考 PLC 的技术性能指标表。这是衡量各种不同型号 PLC 产品性能的依据, 也是根据实际需求选择和使用 PLC 的依据。

(3) PLC 编程时用到的数据及数据类型可以是字符串、布尔型、整型和实型; 指令中常数可用二进制、十进制、十六进制、ASCII 码或浮点数据来表示。

(4) PLC 内部的编程元件有多种, 应当熟悉各种元器件和它们的直接寻址方式。在处理多个连续单元中的多个数据时, S7-200 系列 PLC 的间接寻址方式非常有用, 应掌握间接寻址方式的使用方法。

## 思考题与练习题

- 一个控制系统需要 12 点数字量输入、30 点数字量输出、7 点模拟量输入和 2 点模拟量输出。试问:
  - 可以选用哪种主机型号?
  - 如何选择扩展模块?
  - 各模块按什么顺序连接到主机? 请画出连接图。
  - 按(3)小题所画出的图形, 其主机和各模块的地址如何分配?
- PLC 中软继电器的主要特点是什么?
- S7-200 系列 PLC 主机中有哪些主要编程元件?
- 间接寻址包括几个步骤? 试举例说明。

## 第五章 PLC 的基本指令及程序设计

随着 PLC 的不断发展,厂家也为用户提供了如梯形图(LAD)、语句表(STL)、功能块图(FBD)和高级语言等编程语言。但不论从 PLC 的产生原因(主要替代继电器接触式控制系统)和广大电气工程技术人员的使用习惯来讲,梯形图和语句表一直是它的最基本,也是最常用的编程语言。本章以 S7-200 CPU22\* 系列 PLC 的指令系统为对象,用举例的形式来说明 PLC 的基本指令系统,然后介绍常用典型电路及环节的编程,最后深入浅出地为读者讲解 PLC 程序的简单设计法。

### 5.1 PLC 的基本逻辑指令及举例

在下面讲解指令和举例的时候,主要用到了 LAD 程序,S7-200 PLC 用 LAD 编程时以每个独立的网络块(Network)为单位,所有的网络块组合在一起就是梯形图程序,这也是 S7-200 PLC 的特点。在 S7-200 PLC 系统使用手册中,许多指令的介绍使用了堆栈的概念,这对初学者的学习带来一定的不便,在以下指令系统的讲解中,本书尽量避开较抽象的概念,力求用简单易懂的描述把指令系统叙述清楚。

#### 5.1.1 逻辑取及线圈驱动指令

逻辑取及线圈驱动指令为 LD、LDN 和 =。

LD(Load):取指令。用于网络块逻辑运算开始的常开触点与母线的连接。

LDN(Load Not):取反指令。用于网络块逻辑运算开始的常闭触点与母线的连接。

=(Out):线圈驱动指令。

图 5-1 所示为上述三条指令的用法。

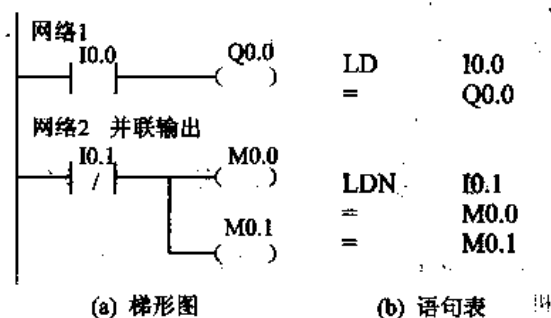


图 5-1 LD、LDN、= 指令使用举例

使用说明:

(1) LD、LDN 指令不止是用于网络块逻辑计算开始时与母线相连的常开和常闭触点,在分支电路块的开始也要使用 LD、LDN 指令,与后面要讲的 ALD、OLD 指令配合完成块电路

的编程。

(2) 并联的=指令可连续使用任意次。

(3) 在同一程序中不能使用双线圈输出,即同一个元器件在同一程序中只使用一次=指令。

(4) LD、LDN、=指令的操作数为:I、Q、M、SM、T、C、V、S和L。T和C也作为输出线圈,但在S7-200 PLC中输出时不是以使用=指令形式出现(见定时器和计数器指令)。

### 5.1.2 触点串联指令

触点串联指令为A、AN。

A(And):与指令。用于单个常开触点的串联连接。

AN(And Not):与反指令。用于单个常闭触点的串联连接。

图5-2所示为上述两条指令的用法。

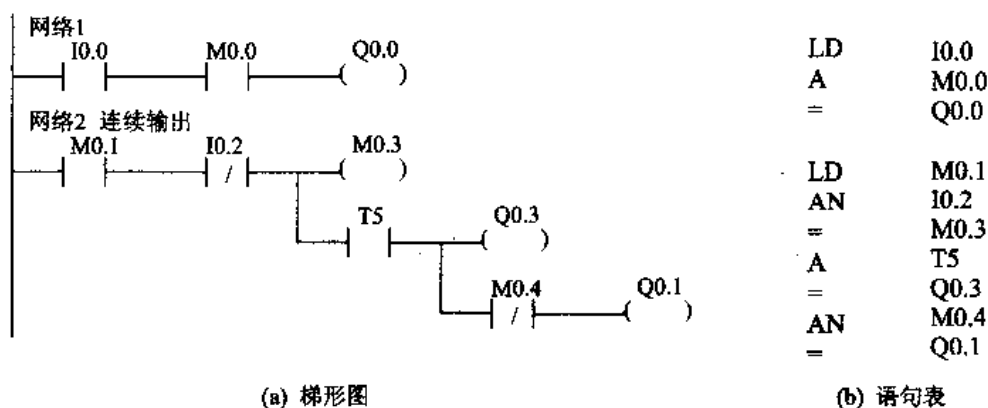


图 5-2 A、AN 指令使用举例

使用说明:

(1) A、AN 是单个触点串联连接指令,可连续使用。但在用梯形图编程时会受到打印宽度和屏幕显示的限制。S7-200 PLC 的编程软件中规定的串联触点使用上限为 11 个。

(2) 图 5-2 中所示的连续输出电路,可以反复使用=指令,但次序必须正确,不然就不能连续使用=指令编程了。图 5-3 所示的电路就不属于连续输出电路。

(3) A、AN 指令的操作数为:I、Q、M、SM、T、C、V、S和L。

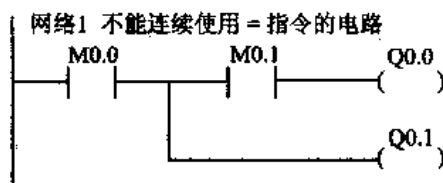


图 5-3 不可连续使用=指令的电路

### 5.1.3 触点并联指令

触点并联指令为O、ON。

O(OR):或指令。用于单个常开触点的并联连接。

ON(Or Not):或反指令。用于单个常闭触点的并联连接。

图 5-4 所示为上述两条指令的用法。

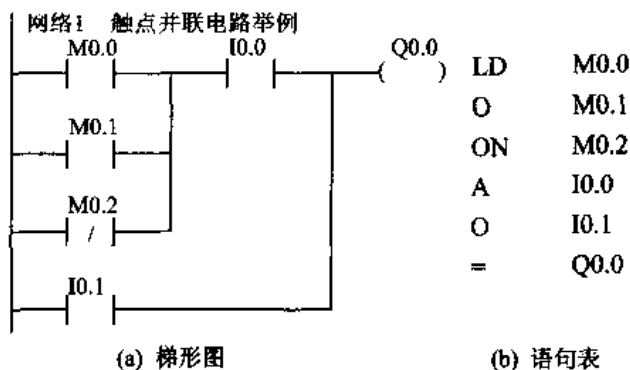


图 5-4 O、ON 指令使用举例

使用说明:

- (1) 单个触点的 O、ON 指令可连续使用。
- (2) O、ON 指令的操作数为: I、Q、M、SM、T、C、V、S 和 L。

#### 5.1.4 串联电路块的并联连接指令

串联电路块的并联连接指令为 OLD。

两个以上触点串联形成的支路叫串联电路块。

OLD(Or Load):或块指令。用于串联电路块的并联连接。

图 5-5 所示为 OLD 指令的用法。

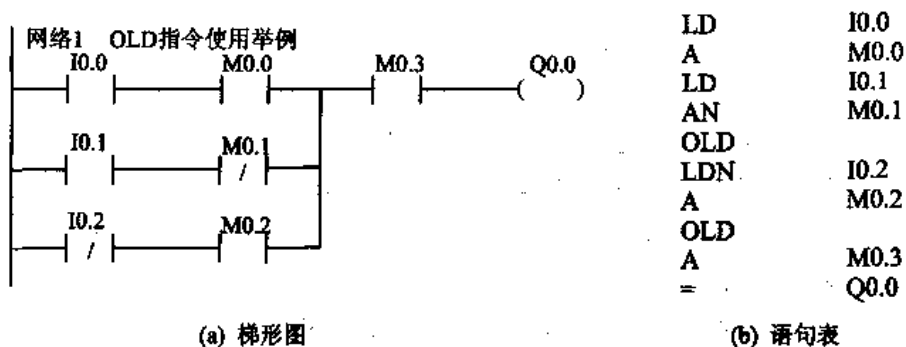


图 5-5 OLD 指令使用举例

使用说明:

- (1) 除在网络块逻辑运算的开始使用 LD 或 LDN 指令外,在块电路的开始也要使用 LD 和 LDN 指令。
- (2) 每完成一次块电路的并联时要写上 OLD 指令。
- (3) OLD 指令无操作数。

#### 5.1.5 并联电路块的串联连接指令

并联电路块的串联连接指令为 ALD。

两条以上支路并联形成的电路叫并联电路块。

ALD(And Load):与块指令。用于并联电路块的串联连接。

图 5-6 所示为 ALD 指令的用法。

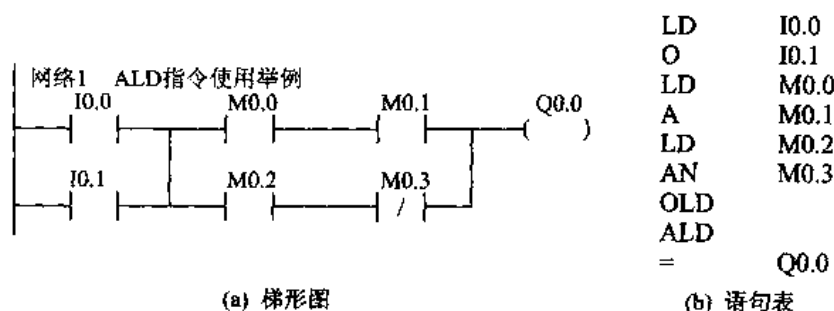


图 5-6 ALD 指令使用举例

使用说明:

- (1) 在块电路开始时要使用 LD 和 LDN 指令。
- (2) 在每完成一次块电路的串联连接后要写上 ALD 指令。
- (3) ALD 指令无操作数。

### 5.1.6 置位、复位指令

置位(Set)/复位(Reset)指令的 LAD 和 STL 形式以及功能如表 5-1 所列。

表 5-1 置位/复位指令的功能表

	LAD	STL	功 能
置位指令	bit N	S bit, N	从 bit 开始的 N 个元件置 1 并保持
复位指令	bit N	R bit, N	从 bit 开始的 N 个元件清零并保持

图 5-7 所示为 S/R 指令的用法。

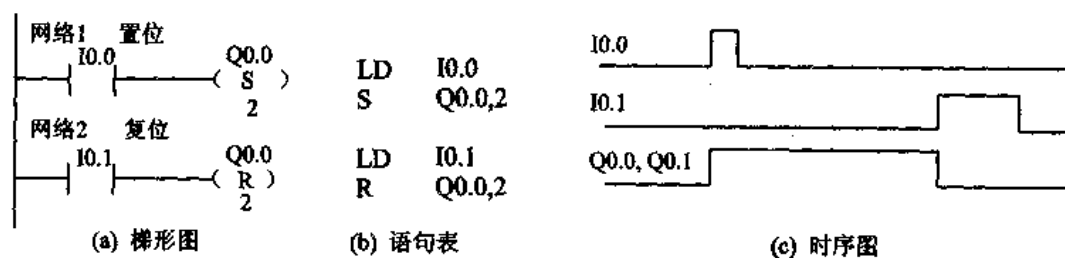


图 5-7 S/R 指令使用举例

使用说明:

- (1) 对位元件来说一旦被置位,就保持在通电状态,除非对它复位;而一旦被复位就保持在断电状态,除非再对它置位。
- (2) S/R 指令可以互换次序使用,但由于 PLC 采用扫描工作方式,所以写在后面的指令

具有优先权。如图 5-7 中,若 I0.0 和 I0.1 同时为 1,则 Q0.0、Q0.1 肯定处于复位状态而为 0。

(3) 如果对计数器和定时器复位,则计数器和定时器的当前值被清零。定时器和计数器的复位有其特殊性,具体情况大家可参考计数器和定时器的有关部分。

(4)  $N$  的常数范围为 1~255,  $N$  也可为:VB、IB、QB、MB、SMB、SB、LB、AC、常数、\* VD、\* AC 和 \* LD。一般情况下使用常数。

(5) S/R 指令的操作数为:I、Q、M、SM、T、C、V、S 和 L。

### 5.1.7 RS 触发器指令

RS 触发器指令在编程软件 Micro/WIN32 V3.2 版本中才有。它包括两条指令:

SR(Set Dominant Bistable):置位优先触发器指令。当置位信号(SI)和复位信号(R)都为真时,输出为真。

RS(Reset Dominant Bistable):复位优先触发器指令。当置位信号(S)和复位信号(R1)都为真时,输出为假。

RS 触发器指令的 LAD 形式如图 5-8 所示。图 5-8(a)为 SR 指令,图 5-8(b)为 RS 指令。Bit 参数用于指定被置位或者被复位的 BOOL 参数。RS 触发器指令没有 STL 形式,但可通过编程软件把 LAD 形式转换成 STL 形式,不过很难读懂。所以建议如果使用 RS 触发器指令最好使用 LAD 形式。

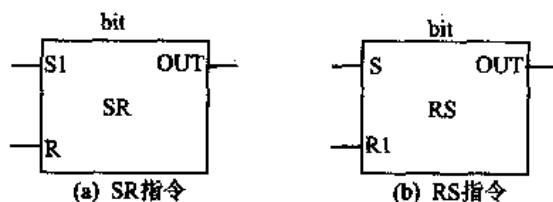


图 5-8 RS 触发器指令

RS 触发器指令的真值如表 5-2 所列。

表 5-2 RS 触发器指令的真值表

指 令	SI	R	输出(bit)
置位优先触发器指令 (SR)	0	0	保持前一状态
	0	1	0
	1	0	1
	1	1	1
指 令	S	R1	输出(bit)
复位优先触发器指令 (RS)	0	0	保持前一状态
	0	1	0
	1	0	1
	1	1	0

RS 触发器指令的输入/输出操作数为:I、Q、V、M、SM、S、T、C。Bit 的操作数为:I、Q、V、

M 和 S。这些操作数的数据类型均为 BOOL 型。

RS 触发器指令的使用举例如图 5-9 所示。图 5-9(b)为在给定的输入信号波形下产生的输出波形。

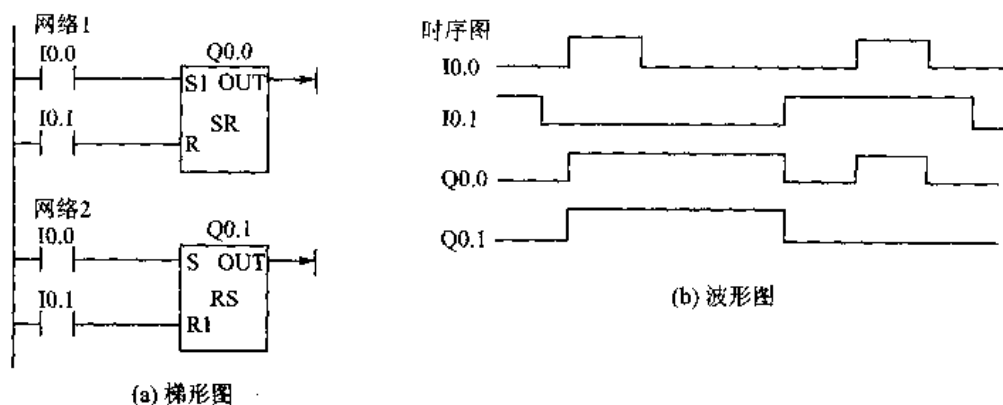


图 5-9 RS 触发器指令使用举例

### 5.1.8 立即指令

立即指令是为了提高 PLC 对输入/输出的响应速度而设置的,它不受 PLC 循环扫描工作方式的影响,允许对输入和输出点进行快速直接存取。当用立即指令读取输入点的状态时,对 I 进行操作,相应的输入映像寄存器中的值并未更新;当用立即指令访问输出点时,对 Q 进行操作,新值同时写到 PLC 的物理输出点和相应的输出映像寄存器。

立即指令的名称和使用说明如表 5-3 所列。

表 5-3 立即指令的名称和使用说明

指令名称	STL	LAD	使用说明
立即取	LDI bit		bit 只能为 I
立即取反	LDNI bit		
立即或	OI bit		
立即或反	ONI bit		
立即与	AI bit		
立即与反	ANI bit		
立即输出	=I bit		bit 只能为 Q
立即置位	SI bit, N		1. bit 只能为 Q 2. N 的范围: 1~128 3. N 的操作数同 S/R 指令
立即复位	RI bit, N		

图 5-10 所示为立即指令的用法。

在理解本例的过程中,一定要注意哪些地方使用了立即指令,哪些地方没有使用立即指令。要理解输出物理触点和相应的输出映像寄存器是不一样的概念,并且要结合 PLC 工作方

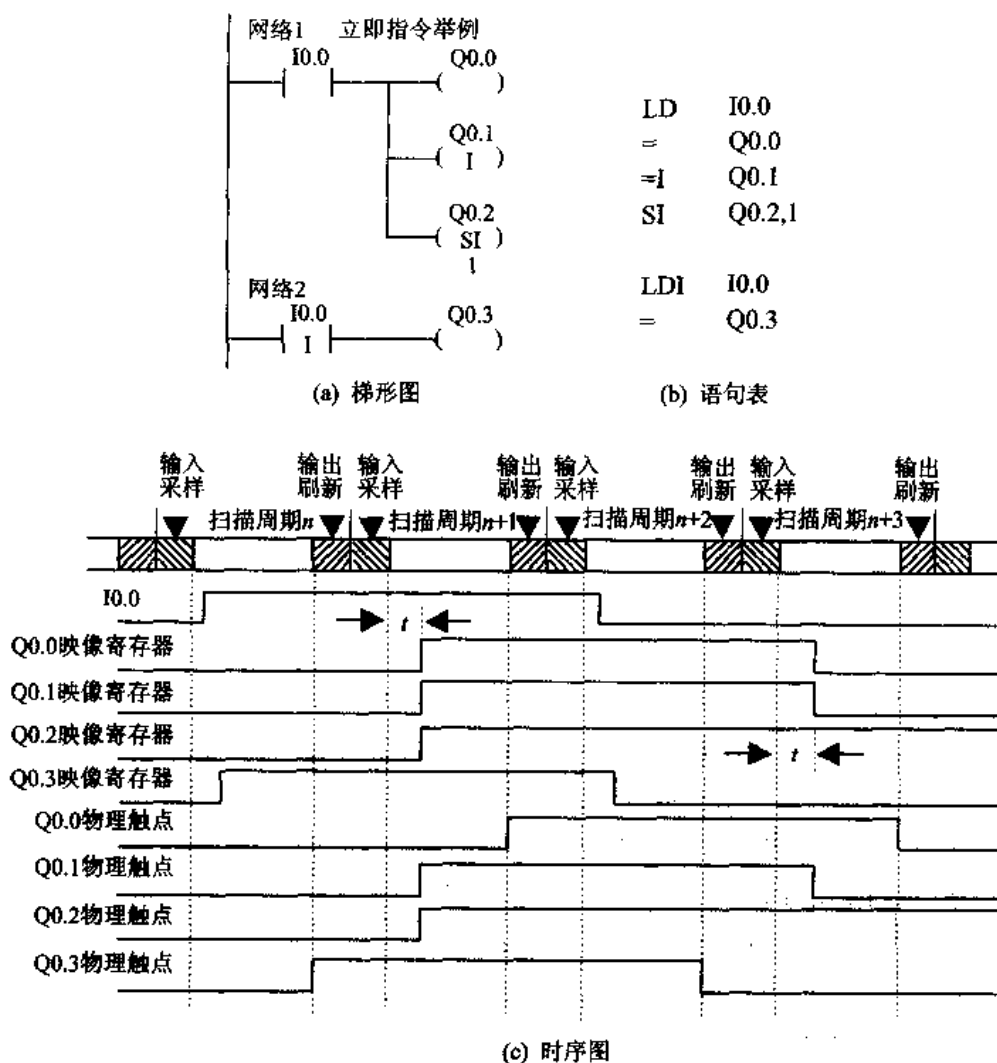


图 5-10 立即指令使用举例

式的原理来看时序图。图 5-10 中,  $t$  为执行到输出点处程序所用的时间,  $Q0.0$ 、 $Q0.1$ 、 $Q0.2$  的输入逻辑是  $I0.0$  的普通常开触点。 $Q0.0$  为普通输出, 在程序执行到它时, 它的映像寄存器的状态会随着本扫描周期采集到的  $I0.0$  状态的变化而改变, 而它的物理触点要等到本扫描周期的输出刷新阶段才改变;  $Q0.1$ 、 $Q0.2$  为立即输出, 在程序执行到它们时, 它们的物理触点和输出映像寄存器同时改变; 而对  $Q0.3$  来说, 它的输入逻辑是  $I0.0$  的立即触点, 所以在程序执行到它时,  $Q0.3$  的映像寄存器的状态会随着  $I0.0$  即时状态的变化而立即改变, 而它的物理触点要等到本扫描周期的输出刷新阶段才改变。

### 5.1.9 边沿脉冲指令

边沿脉冲指令为 EU(Edge Up)、ED(Edge Down)。

边沿脉冲指令的使用及说明如表 5-4 所列。

表 5-4 边沿脉冲指令使用说明

指令名称	LAD	STL	功 能	说 明
上升沿脉冲	$\uparrow P \downarrow$	EU	在上升沿产生脉冲	无操作数
下降沿脉冲	$\downarrow N \uparrow$	ED	在下降沿产生脉冲	

边沿脉冲指令 EU/ED 用法如图 5-11 所示。

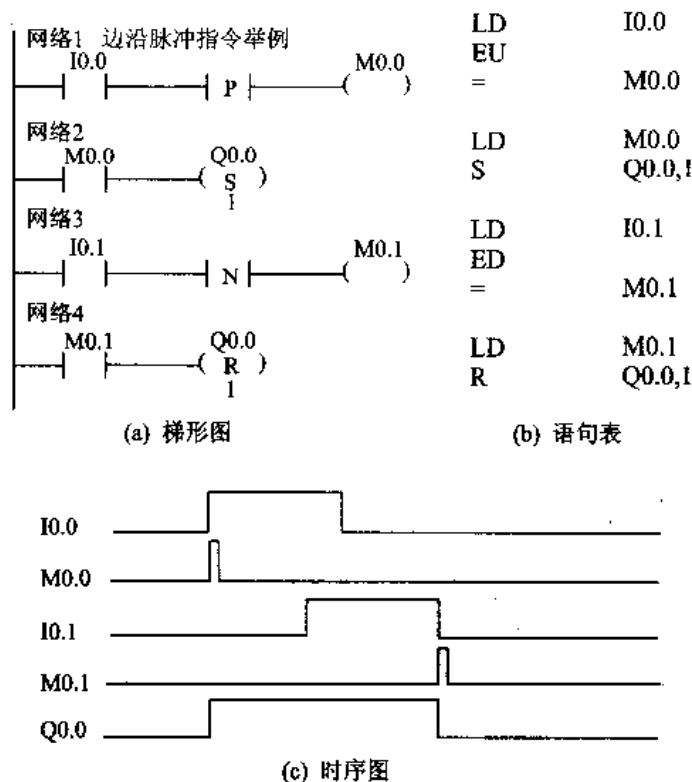


图 5-11 边沿脉冲指令 EU/ED 使用举例

EU 指令对其之前的逻辑运算结果的上升沿产生一个宽度为一个扫描周期的脉冲,如图中的 M0.0。ED 指令对逻辑运算结果的下降沿产生一个宽度为一个扫描周期的脉冲,如图中的 M0.1。脉冲指令常用于启动及关断条件的判定以及配合功能指令完成一些逻辑控制任务。

#### 5.1.10 逻辑堆栈操作指令

S7-200 系列 PLC 使用一个 9 层堆栈来处理所有逻辑操作,它和计算机中的堆栈结构相同。堆栈是一组能够存储和取出数据的暂存单元,其特点是“先进后出”。每一次进行入栈操作,新值放入栈顶,栈底值丢失;每一次进行出栈操作,栈顶值弹出,栈底值补进随机数。逻辑堆栈指令主要用来完成对触点进行的复杂连接。

西门子公司的系统手册中把 ALD、OLD、LPS、LRD、LPP 和 LDS 等指令都归纳为栈操作指令。其中 ALD(与块指令)和 OLD(或块指令)前面已经介绍过,下面分别介绍其余四条指令。

##### 1. 逻辑入栈 LPS、逻辑读栈 LRD 和逻辑出栈 LPP 指令

这三条指令也称为多重输出指令,主要用于一些复杂逻辑的输出处理。

LPS(Logic Push):逻辑入栈指令(分支电路开始指令)。从梯形图中的分支结构中可以从

形象地看出,它用于生成一条新的母线,其左侧为原来的主逻辑块,右侧为新的从逻辑块,因此可以直接编程。从堆栈使用上来讲,LPS 指令的作用是把栈顶值复制后压入堆栈。

LRD(Logic Read):逻辑读栈指令。在梯形图分支结构中,当新母线左侧为主逻辑块时,LPS 开始右侧的第一个从逻辑块编程,LRD 开始第二个以后的从逻辑块编程。从堆栈使用上来讲,LRD 读取最近的 LPS 压入堆栈的内容,而堆栈本身不进行 Push 和 Pop 工作。

LPP(Logic Pop):逻辑出栈指令(分支电路结束指令)。在梯形图分支结构中,LPP 用于 LPS 产生的新母线右侧的最后一个从逻辑块编程,它在读取完离它最近的 LPS 压入堆栈内容的同时复位该条新母线。从堆栈使用上来讲,LPP 把堆栈弹出一级,堆栈内容依次上移。

上述三条指令的用法如图 5-12、图 5-13 和图 5-14 所示。

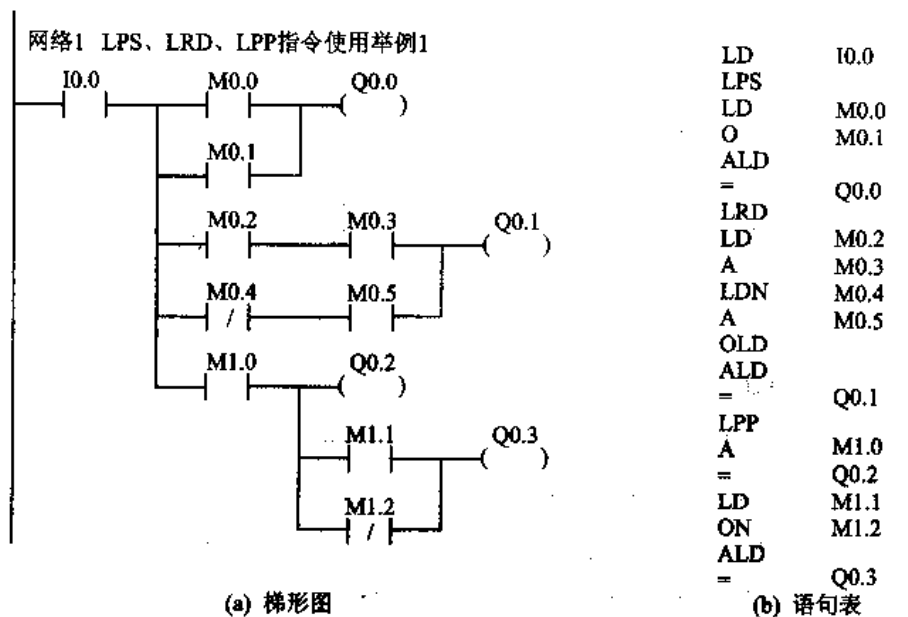


图 5-12 LPS、LRD、LPP 指令使用举例 1

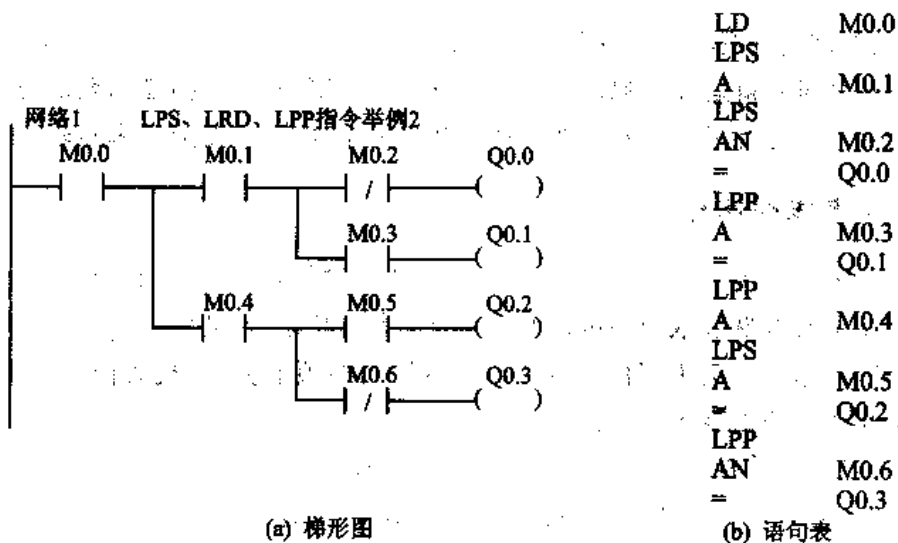


图 5-13 LPS、LRD、LPP 指令使用举例 2

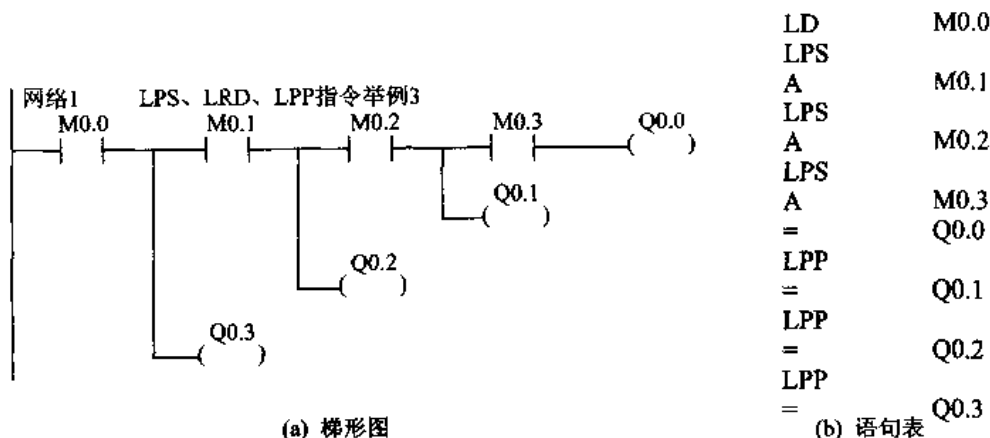


图 5-14 LPS、LRD、LPP 指令使用举例 3

使用说明:

(1) 由于受堆栈空间的限制(9 层堆栈), LPS、LPP 指令连续使用时应少于 9 次。

(2) LPS 和 LPP 指令必须成对使用, 它们之间可以使用 LRD 指令。

(3) LPS、LRD、LPP 指令无操作数。

## 2. 装入堆栈指令 LDS(Load Stack)

它的功能是复制堆栈中的第  $n$  个值到栈顶, 而栈底丢失。

该指令在编程中使用较少。

指令格式: LDS  $n$  ( $n$  为 0~8 的整数)

例如, 执行指令: LDS 3

该指令执行后堆栈发生变化的情况如表 5-5 所列。

表 5-5 LDS 指令使用举例

入栈前	入栈后
iv0	iv3
iv1	iv0
iv2	iv1
iv3	iv2
iv4	iv3
iv5	iv4
iv6	iv5
iv7	iv6
iv8	iv7

### 5.1.11 定时器

定时器是 PLC 中最常用的元器件之一。用好、用对定时器对 PLC 程序设计非常重要。定时器编程时要预置定时值, 在运行过程中当定时器的输入条件满足时, 当前值从 0 开始按一定的单位增加; 当定时器的当前值到达设定值时, 定时器发生动作, 从而满足各种定时逻辑控制的需要。下面从几个方面来详细讲解定时器的使用。

#### 1. 几个基本概念

##### (1) 种类

S7-200 PLC 为用户提供了三种类型的定时器: 接通延时定时器(TON)、有记忆接通延时定时器(TONR)和断开延时定时器(TOF)。

##### (2) 分辨率与定时时间的计算

单位时间的的时间增量称为定时器的分辨率。S7-200 PLC 定时器有 3 个分辨率等级: 1 ms、10 ms 和 100 ms。

定时器定时时间  $T$  的计算:  $T = PT \times S$ 。式中:  $T$  为实际定时时间,  $PT$  为设定值,  $S$  为分辨率。

例如: TON 指令使用 T97(为 10 ms 的定时器), 设定值为 100, 则实际定时时间为

$$T = 100 \times 10 \text{ ms} = 1\,000 \text{ ms}$$

定时器的设定值  $PT$ , 数据类型为 INT 型。操作数可为: VW、IW、QW、MW、SW、SMW、LW、AIW、T、C、AC、\* VD、\* AC、\* LD 和常数, 其中常数最为常用。

### (3) 定时器的编号

定时器的编号用定时器的名称和它的常数编号(最大数为 255)来表示, 即  $T^{***}$ 。如: T40。

定时器的编号包含两方面的变量信息: 定时器位和定时器当前值。

定时器位: 与其他继电器的输出相似。当定时器的当前值达到设定值  $PT$  时, 定时器的触点动作。

定时器当前值: 存储定时器当前所累计的时间, 它用 16 位符号整数来表示, 最大计数值为 32 767。

定时器的分辨率和编号如表 5-6 所列。

表 5-6 定时器分辨率和编号

定时器类型	分辨率/ms	最大当前值/s	定时器编号
TONR	1	32.767	T0, T64
	10	327.67	T1~T4, T65~T68
	100	3 276.7	T5~T31, T69~T95
TON, TOF	1	32.767	T32, T96
	10	327.67	T33~T36, T97~T100
	100	3 276.7	T37~T63, T101~T255

从上表可以看出 TON 和 TOF 使用相同范围的定时器编号。需要注意的是, 在同一个 PLC 程序中决不能把同一个定时器号同时用做 TON 和 TOF。例如在程序中, 不能既有接通延时(TON)定时器 T32, 又有断开延时(TOF)定时器 T32。

## 2. 定时器指令使用说明

三种定时器指令的 LAD 和 STL 格式如表 5-7 所列。

表 5-7 定时器指令的 LAD 和 STL 形式

格 式	名 称		
	接通延时定时器	有记忆接通延时定时器	断开延时定时器
LAD			
STL	TON T***, PT	TONR T***, PT	TOF T***, PT

### (1) 接通延时定时器 TON(On-Delay Timer)

接通延时定时器用于单一时间间隔的定时。上电周期或首次扫描时,定时器位为 OFF,当前值为 0。输入端接通时,定时器位为 OFF,当前值从 0 开始计时,当前值达到设定值时,定时器位为 ON,当前值仍连续计数到 32 767。输入端断开,定时器自动复位,即定时器位为 OFF,当前值为 0。

### (2) 记忆接通延时定时器 TONR(Retentive On-Delay Timer)

顾名思义,记忆接通延时定时器具有记忆功能,它用于对许多间隔的累计定时。上电周期或首次扫描时,定时器位为 OFF,当前值保持在掉电前的值。当输入端接通时,当前值从上次保持值继续计时;当累计当前值达到设定值时,定时器位为 ON,当前值可继续计数到 32 767。需要注意的是,TONR 定时器只能用复位指令 R 对其进行复位操作。TONR 复位后,定时器位为 OFF,当前值为 0。掌握好对 TONR 的复位及启动是使用好 TONR 指令的关键。

### (3) 断开延时定时器 TOF(Off-Delay Timer)

断开延时定时器用于断电后的单一间隔时间计时。上电周期或首次扫描时,定时器位为 OFF,当前值为 0。输入端接通时,定时器位为 ON,当前值为 0。当输入端由接通到断开时,定时器开始计时。当达到设定值时定时器位为 OFF,当前值等于设定值,停止计时。输入端再次由 OFF→ON 时,TOF 复位,这时 TOF 的位为 ON,当前值为 0。如果输入端再从 ON→OFF,则 TOF 可实现再次启动。

## 3. 应用举例

图 5-15 所示为三种类型定时器的基本使用举例,其中 T35 为 TON、T2 为 TONR、T36 为 TOF。

## 4. 定时器的刷新方式和正确使用

### (1) 定时器的刷新方式

在 S7-200 系列 PLC 的定时器中,1 ms、10 ms、100 ms 定时器的刷新方式是不同的,从而在使用方法上也有很大的不同。这和其他 PLC 是有很区别的。使用时一定要注意根据使用场合和要求来选择定时器。

① 1 ms 定时器 1 ms 定时器由系统每隔 1 ms 刷新一次,与扫描周期及程序处理无关。它采用的是中断刷新方式。因此,当扫描周期大于 1 ms 时,在一个周期中可能被多次刷新。其当前值在一个扫描周期内不一定保持一致。

② 10 ms 定时器 10 ms 定时器由系统在每个扫描周期开始时自动刷新,由于是每个扫描周期只刷新一次,故在一个扫描周期内定时器位和定时器的当前值保持不变。

③ 100 ms 定时器 100 ms 定时器在定时器指令执行时被刷新,因此,如果 100 ms 定时器被激活后,如果不是每个扫描周期都执行定时器指令或在一个扫描周期内多次执行定时器指令,则都会造成计时失准,所以在后面讲到的跳转指令和循环指令段中使用定时器时,要格外小心。100 ms 定时器仅用在定时器指令在每个扫描周期执行一次的程序中。

### (2) 定时器的正确使用

图 5-16 所示为正确使用定时器的一个例子。它用来在定时器计时时间到时产生一个宽度为一个扫描周期的脉冲。

结合各种定时器的刷新方式规定,从图中可以看出:

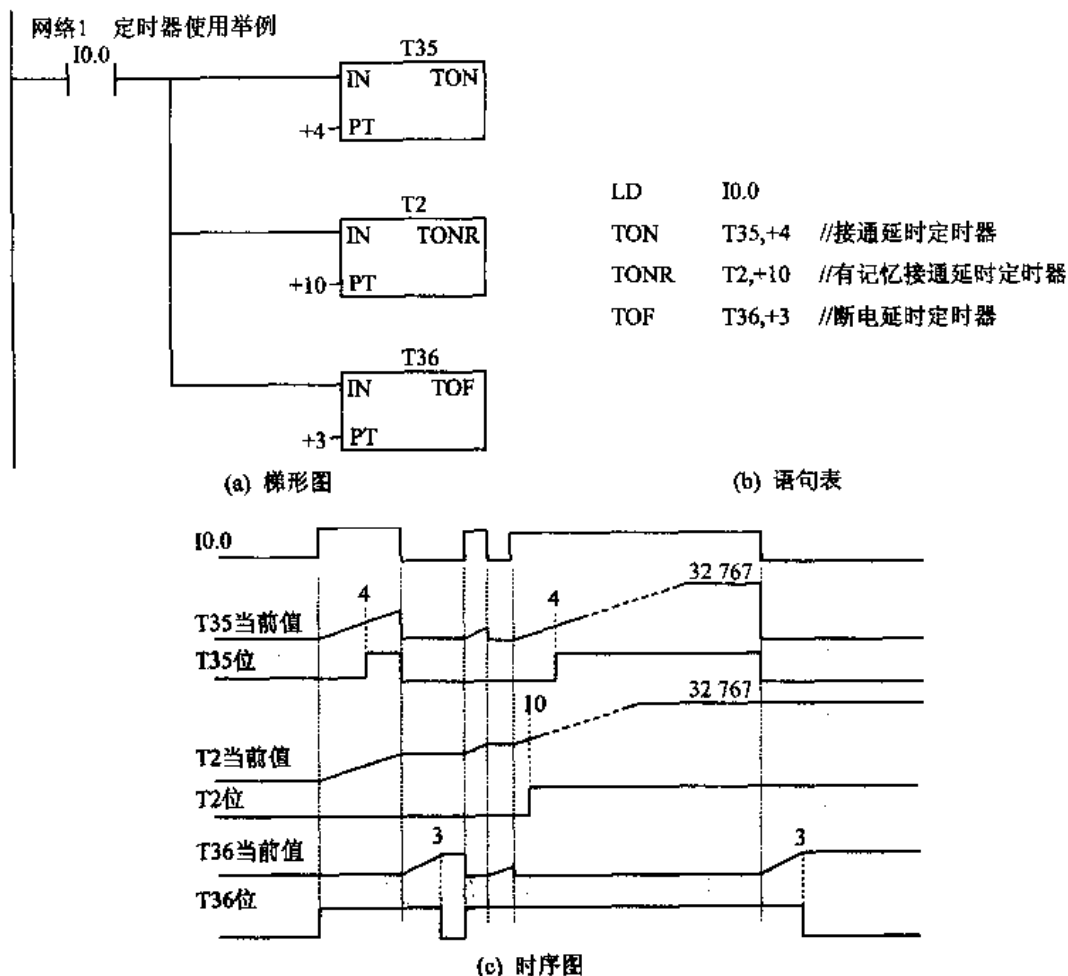


图 5-15 定时器基本使用举例

① 对 1 ms 定时器 T32, 在使用错误方法时, 只有当定时器的刷新发生在 T32 的常闭触点执行以后到 T32 的常开触点执行以前的区间时, Q0.0 才能产生宽度为一个扫描周期的脉冲, 而这种可能性是极小的。在其他情况, 则这个脉冲产生不了。

② 对 10 ms 定时器 T33, 使用错误方法时, Q0.0 永远产生不了这个脉冲。因为当定时器计时到时, 定时器在每次扫描开始时刷新。该例中 T33 被置位, 但执行到定时器指令时, 定时器将被复位(当前值和位都被置 0)。当常开触点 T33 被执行时, T33 永远为 OFF, Q0.0 也将为 OFF, 即永远不会被置位为 ON。

③ 100 ms 定时器在执行指令时刷新, 所以当定时器 T37 到达设定值时, Q0.0 肯定会产生这个脉冲。

改用正确使用方法后, 把定时器到达设定值产生结果的元器件的常闭触点用作定时器本身的输入, 则不论哪种定时器, 都能保证定时器达到设定值时, Q0.0 产生的宽度为一个扫描周期的脉冲。所以, 在使用定时器时, 要弄清楚定时器的分辨率, 否则, 一般情况下不要把定时器本身的常闭触点作为自身的复位条件。在实际使用时, 为了简单, 100 ms 的定时器常采用自复位逻辑, 而且 100 ms 定时器也是使用最多的定时器。

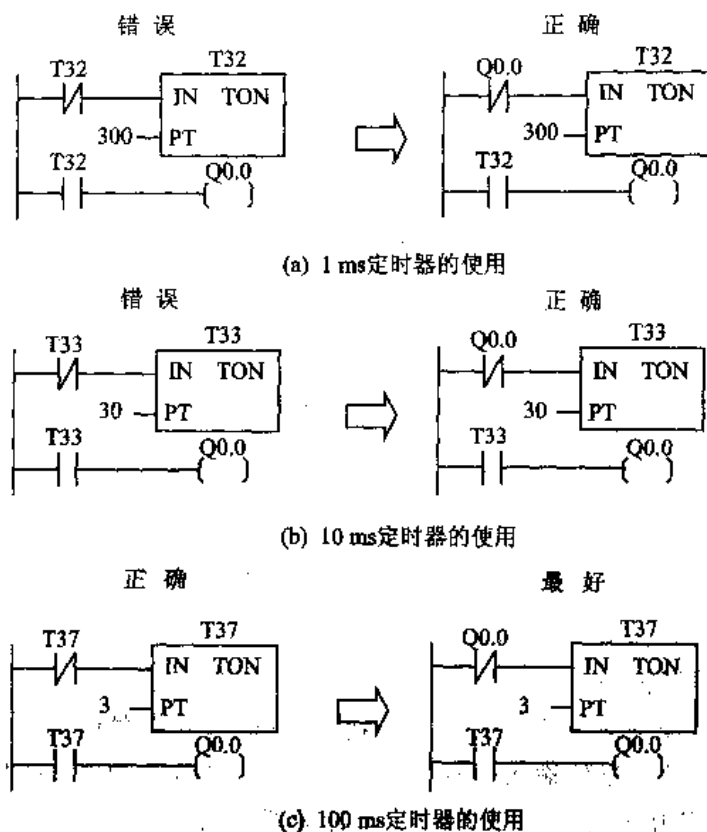


图 5-16 定时器的正确使用举例

### 5.1.12 计数器

计数器用来累计输入脉冲的次数,在实际应用中用来对产品进行计数或完成复杂的逻辑控制任务。计数器的使用和定时器基本相似,编程时输入它的计数设定值,计数器累计它的脉冲输入端信号上升沿的个数。当计数值达到设定值时,计数器发生动作,以便完成计数控制任务。

#### 1. 几个基本概念

##### (1) 种类

S7-200 系列 PLC 的计数器有 3 种:增计数器 CTU、增减计数器 CTUD 和减计数器 CTD。

##### (2) 编号

计数器的编号用计数器名称和数字(0~255)组成,即 C\*\*\*,如 C6。

计数器的编号包含两方面的信息:计数器的位和计数器当前值。

计数器位:计数器位和继电器一样是一个开关量,表示计数器是否发生动作的状态。当计数器的当前值达到设定值时,该位被置位为 ON。

计数器当前值:其值是一个存储单元,它用来存储计数器当前所累计的脉冲个数,用 16 位符号整数来表示,最大数值为 32 767。

##### (3) 计数器的输入端和操作数

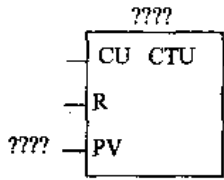
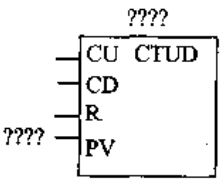
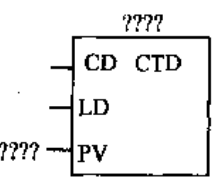
设定值输入:数据类型为 INT 型。寻址范围:VW、IW、QW、MW、SW、SMW、LW、AIW、

T、C、AC、\* VD、\* AC、\* LD 和常数。一般情况下使用常数作为计数器的设定值。

## 2. 计数器指令使用说明

计数器指令的 LAD 和 STL 格式如表 5-8 所列。

表 5-8 计数器的指令格式

格 式	名 称		
	增计数器	增减计数器	减计数器
LAD			
STL	CTU C* *,PV	CTUD C* *,PV	CTD C* *,PV

### (1) 增计数器 CTU(Count Up)

首次扫描时,计数器位为 OFF,当前值为 0。在计数脉冲输入端 CU 的每个上升沿,计数器计数 1 次,当前值增加一个单位。当前值达到设定值时,计数器位为 ON,当前值可继续计数到 32 767 后停止计数。复位输入端有效或对计数器执行复位指令,计数器自动复位,即计数器位为 OFF,当前值为 0。图 5-17 所示为增计数器的用法。

注意:在语句表中,CU、R 的编程顺序不能错误。

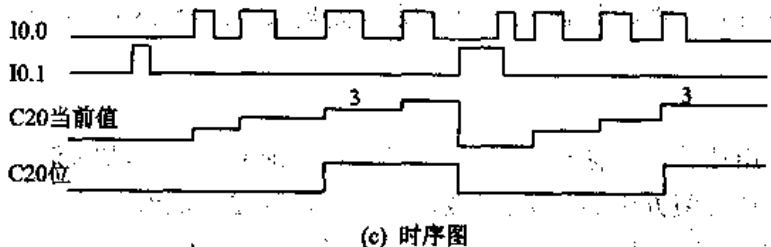
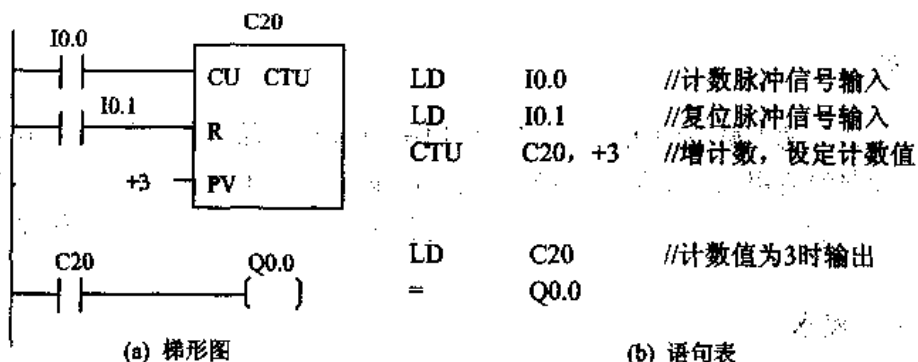


图 5-17 增计数器用法举例

### (2) 增减计数器 CTUD(Count Up/Down)

增减计数器有两个计数脉冲输入端:CU 输入端用于递增计数,CD 输入端用于递减计数。首次扫描时,计数器位为 OFF,当前值为 0。CU 输入的每个上升沿,计数器当前值增加 1 个

单位;CD 输入的每个上升沿,都使计数器当前值减小 1 个单位,当前值达到设定值时,计数器位置位为 ON。

增减计数器当前值计数到 32 767(最大值)后,下一个 CU 输入的上升沿将使当前值跳变为最小值(-32 768);当前值达到最小值-32 768 后,下一个 CD 输入的上升沿将使当前值跳变为最大值 32 767。复位输入端有效或使用复位指令对计数器执行复位操作后,计数器自动复位,即计数器位为 OFF,当前值为 0。图 5-18 所示为增减计数器的用法。

注意:在语句表中, CU、CD、R 的顺序不能错误。

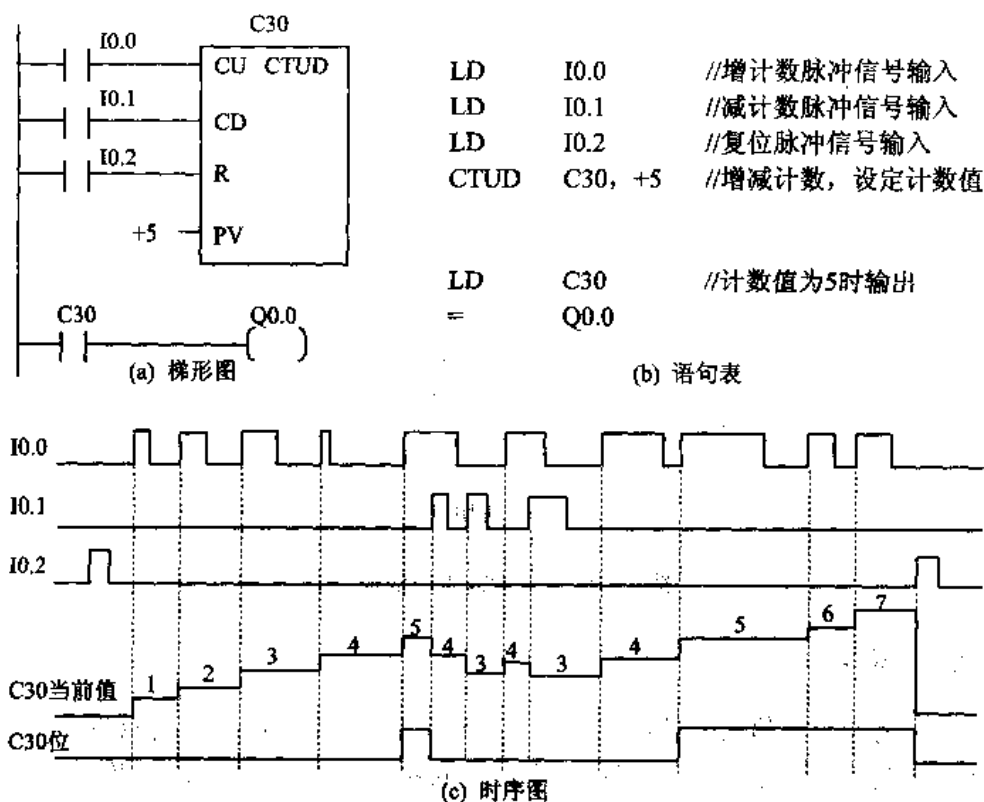


图 5-18 增减计数器用法举例

### (3) 减计数器 CTD(Count Down)

首次扫描时,计数器位为 ON,当前值为预设定值 PV。对 CD 输入端的每个上升沿计数器计数 1 次,当前值减少一个单位,当前值减小到 0 时,计数器位置位为 ON,复位输入端有效或对计数器执行复位指令,计数器自动复位,即计数器位 OFF,当前值复位为设定值。图 5-19 所示为减计数器的用法。

注意:减计数器的复位端是 LD,而不是 R。在语句表中,CD、LD 的顺序不能错误。

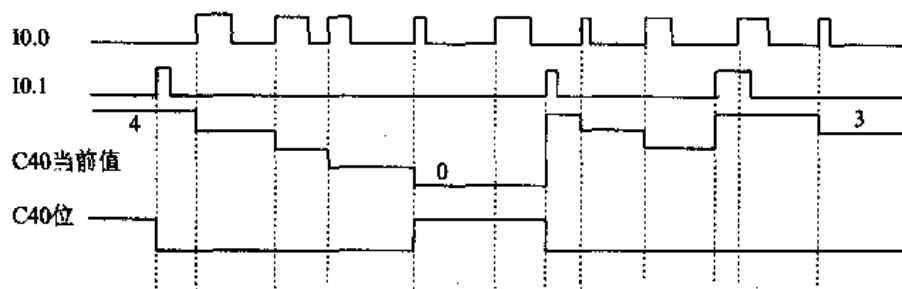
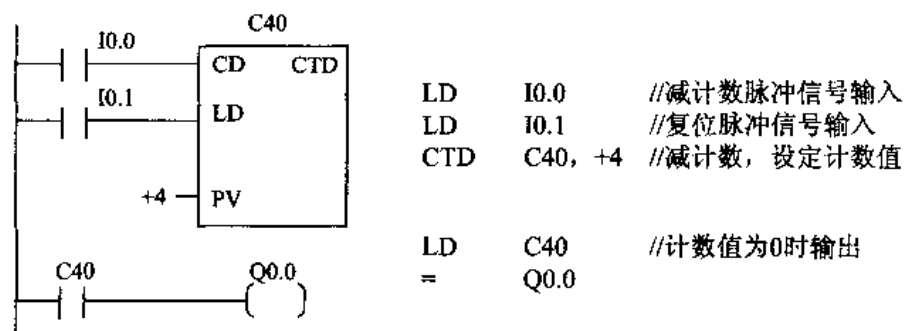


图 5-19 减计数器用法举例

### 5.1.13 比较指令

比较指令是将两个数值或字符串按指定条件进行比较,条件成立时,触点就闭合。所以比较指令实际上也是一种位指令。在实际应用中,比较指令为上、下限控制以及为数值条件判断提供了方便。

比较指令的类型有:字节比较、整数比较、双字整数比较、实数比较和字符串比较。

数值比较指令的运算符有: =、>、<、<=、>= 和 <> 等 6 种,而字符串比较指令只有 = 和 <> 两种。

对比较指令可进行 LD、A 和 O 编程。

比较指令的 LAD 和 STL 形式如表 5-9 所列。

说明:字符串比较指令在 PLC CPU1.21 和 Micro/WIN32 V3.2 以上版本中才有。

表 5-9 比较指令的 LAD 和 STL 形式

形 式	方 式				
	字节比较	整数比较	双字整数比较	实数比较	字符串比较
LAD (以=为例)	IN1 └─[=B]─┐ IN2	IN1 └─[=I]─┐ IN2	IN1 └─[=D]─┐ IN2	IN1 └─[=R]─┐ IN2	IN1 └─[=S]─┐ IN2
STL	LDB= IN1,IN2	LDW= IN1,IN2	LDD= IN1,IN2	LDR= IN1,IN2	LDS= IN1,IN2
	AB= IN1,IN2	AW= IN1,IN2	AD= IN1,IN2	AR= IN1,IN2	AS= IN1,IN2
	OB= IN1,IN2	OW= IN1,IN2	OD= IN1,IN2	OR= IN1,IN2	OS= IN1,IN2
	LDB<> IN1,IN2	LDW<> IN1,IN2	LDD<> IN1,IN2	LDR<> IN1,IN2	LDS<> IN1,IN2
	AB<> IN1,IN2	AW<> IN1,IN2	AD<> IN1,IN2	AR<> IN1,IN2	AS<> IN1,IN2
	OB<> IN1,IN2	OW<> IN1,IN2	OD<> IN1,IN2	OR<> IN1,IN2	OS<> IN1,IN2
	LDB< IN1,IN2	LDW< IN1,IN2	LDD< IN1,IN2	LDR< IN1,IN2	
	AB< IN1,IN2	AW< IN1,IN2	AD< IN1,IN2	AR< IN1,IN2	
	OB< IN1,IN2	OW< IN1,IN2	OD< IN1,IN2	OR< IN1,IN2	
	LDB<= IN1,IN2	LDW<= IN1,IN2	LDD<= IN1,IN2	LDR<= IN1,IN2	
	AB<= IN1,IN2	AW<= IN1,IN2	AD<= IN1,IN2	AR<= IN1,IN2	
	OB<= IN1,IN2	OW<= IN1,IN2	OD<= IN1,IN2	OR<= IN1,IN2	
	LDB> IN1,IN2	LDW> IN1,IN2	LDD> IN1,IN2	LDR> IN1,IN2	
	AB> IN1,IN2	AW> IN1,IN2	AD> IN1,IN2	AR> IN1,IN2	
	OB> IN1,IN2	OW> IN1,IN2	OD> IN1,IN2	OR> IN1,IN2	
	LDB>= IN1,IN2	LDW>= IN1,IN2	LDD>= IN1,IN2	LDR>= IN1,IN2	
	AB>= IN1,IN2	AW>= IN1,IN2	AD>= IN1,IN2	AR>= IN1,IN2	
	OB>= IN1,IN2	OW>= IN1,IN2	OD>= IN1,IN2	OR>= IN1,IN2	
IN1 和 IN2 寻址范围	IB, QB, MB, SMB, VB, SB, LB, AC, * VD, * AC, * LD, 常数	IW, QW, MW, SMW, VW, SW, LW, AC, * VD, * AC, * LD, 常数	ID, QD, MD, SMD, VD, SD, LD, AC, * VD, * AC, * LD, 常数	ID, QD, MD, SMD, VD, SD, LD, AC, * VD, * AC, * LD, 常数	(字符) VB, LB, * VD, * LD, * AC

字节比较用于比较两个字节型整数值 IN1 和 IN2 的大小,字节比较是无符号的。整数比较用于比较两个一个字长的整数值 IN1 和 IN2 的大小,整数比较是有符号的,其范围是 16#8000~16#7FFF。

双字整数比较用于比较两个双字长整数值 IN1 和 IN2 的大小。它们的比较也是有符号的,其范围是 16#80000000~16#7FFFFFFF。

实数比较用于比较两个双字长实数值 IN1 和 IN2 的大小,实数比较是有符号的。负实数范围为-1.175495E-38~-3.402823E+38,正实数范围是+1.175495E-38~+3.402823E+38。

字符串比较用于比较两个字符串数据的相同与否。字符串的长度不能超过 254 个字符。

图 5-20 所示为比较指令的用法。

从图 5-20 中可以看出:计数器 C30 中的当前值大于等于 30 时,Q0.0 为 ON;VD1 中的实数小于 95.8 且 I0.0 为 ON 时,Q0.1 为 ON;VB1 中的值大于 VB2 的值或 I0.1 为 ON 时,Q0.2 为 ON。

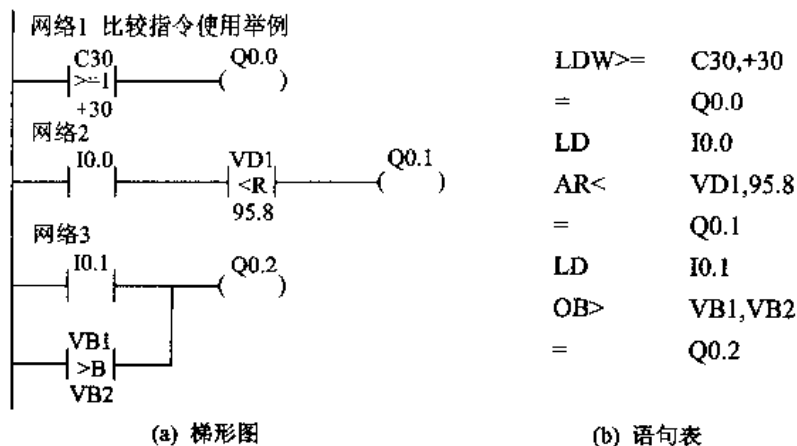


图 5-20 比较指令使用举例

### 5.1.14 NOT 及 NOP 指令

#### 1. 取反指令 NOT

将复杂逻辑结果取反,为用户使用反逻辑提供方便。该指令无操作数,其 LAD 和 STL 形式如下。


STL 形式:NOT

LAD 形式:——| NOT |——

#### 2. 空操作指令 NOP(No Operation)

该指令很少被使用,甚至在西门子公司的系统手册中都未介绍。该指令最有可能用在跳转指令的结束处,或在调试程序中使用。该指令对用户程序的执行没有影响,其 LAD 和 STL 形式如下。

STL 形式:NOP N

LAD 形式:——

N 的范围:0~255。

## 5.2 程序控制指令

程序控制类指令使程序结构灵活,合理使用该类指令可以优化程序结构,增强程序功能。这类指令主要包括结束、暂停、看门狗、跳转、子程序、循环和顺序控制等指令。因为顺序控制指令的使用非常多,也非常重要,所以我们把它单独作为一章,放到第六章中讲解。

### 5.2.1 结束及暂停指令

#### 1. 结束指令 END 和 MEND

结束指令分为有条件结束指令(END)和无条件结束指令(MEND)。两条指令在梯形图中以线圈形式编程。指令不含操作数。执行完结束指令后,系统结束主程序,返回到主程序起点。

使用说明:

(1) 结束指令只能用在主程序中,不能在子程序和中断程序中使用。而有条件结束指令可用在无条件的结束指令前结束主程序。

(2) 在调试程序时,在程序的适当位置插入无条件的结束指令可实现程序的分段调试。

(3) 可以利用程序执行的结果状态、系统状态或外部设置切换条件来调用有条件的结束指令,使程序结束。

(4) 使用 Micro/Win32 编程时,编程人员不需手工输入无条件的结束指令,该软件会自动在内部加上一条无条件的结束指令到主程序的结尾。

## 2. 停止指令 STOP

STOP 指令有效时,可以使主机 CPU 的工作方式由 RUN 切换到 STOP,从而立即中止用户程序的执行。STOP 指令在梯形图中以线圈形式编程。指令不含操作数。

STOP 指令可以用在主程序、子程序和中断程序中。如果在中断程序中执行 STOP 指令,则中断处理立即中止,并忽略所有挂起的中断,继续扫描程序的剩余部分,在本次扫描周期结束后,完成将主机从 RUN 到 STOP 的切换。

STOP 和 END 指令通常在程序中来对突发紧急事件进行处理,以避免实际生产中的重大损失。

结束指令和停止指令的用法如图 5-21 所示。

## 5.2.2 看门狗指令

WDR(Watchdog Reset)称为看门狗复位指令,也称为警戒时钟刷新指令。它可以把警戒时钟刷新,即延长扫描周期,从而有效地避免看门狗超时错误。WDR 指令在梯形图中以线圈形式编程,无操作数。

使用 WDR 指令时要特别小心,如果因为使用 WDR 指令而使扫描时间拖得过长(如在循环结构中使用 WDR),那么在中止本次扫描前,下列操作过程将被禁止:

- (1) 通信(自由口除外);
- (2) I/O 刷新(直接 I/O 除外);
- (3) 强制刷新;
- (4) SM 位刷新(SM0、SM5~SM29 的位不能被刷新);
- (5) 运行时间诊断;
- (6) 扫描时间超过 25 s 时,使 10 ms 和 100 ms 定时器不能正确计时;
- (7) 中断程序中的 STOP 指令。

注意:如果希望扫描周期超过 300 ms,或者希望中断时间超过 300 ms,则最好用 WDR 指令来重新触发看门狗定时器。

WDR 指令的用法如图 5-21 所示。

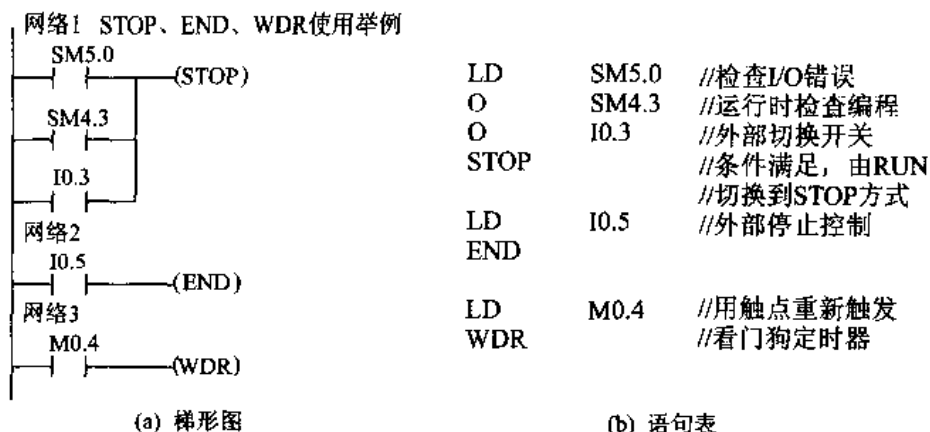


图 5-21 结束、停止及看门狗指令举例

### 5.2.3 跳转及标号指令

跳转指令可以使 PLC 编程的灵活性大大提高,使主机可根据对不同条件的判断,选择不同的程序段执行程序。

跳转指令 JMP(Jump to Label):当输入端有效时,使程序跳转到标号处执行。

标号指令 LBL(Label):指令跳转的目标标号。操作数  $n$  为 0~255。

使用说明:

(1) 跳转指令和标号指令必须配合使用,而且只能使用在同一程序块中,如主程序、同一个子程序或同一个中断程序。不能在不同的程序块中互相跳转。

(2) 执行跳转后,被跳过程序段中的各元器件的状态为:

① Q、M、S、C 等元器件的位保持跳转前的状态;

② 计数器 C 停止计数,当前值存储器保持跳转前的计数值;

③ 对定时器来说,因刷新方式不同而工作状态不同。在跳转期间,分辨率为 1 ms 和 10 ms 的定时器会一直保持跳转前的工作状态,原来工作的继续工作,到设定值后,其位的状态也会改变,输出触点动作,其当前值存储器一直累计到最大值 32 767 才停止。对分辨率为 100 ms 的定时器来说,跳转期间停止工作,但不会复位,存储器里的值为跳转时的值,跳转结束后,若输入条件允许,可继续计时,但已失去了准确计时的意义。所以在跳转段里的定时器要慎用。

跳转指令的使用方法如图 5-22 所示。

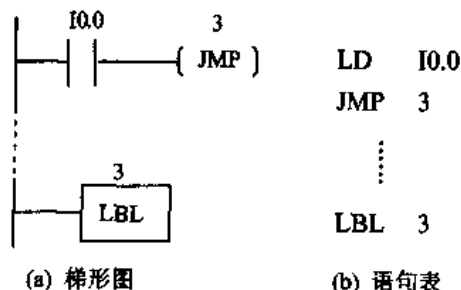


图 5-22 跳转指令使用举例

### 5.2.4 循环指令

循环指令的引入为解决重复执行相同功能的程序段提供了极大方便,并且优化了程序结构。特别是在进行大量相同功能的计算和逻辑处理时,循环指令非常有用。循环指令有两条:FOR 和 NEXT。

#### 1. 循环指令

循环开始指令 FOR:用来标记循环体的开始。

循环结束指令 NEXT:用来标记循环体的结束。无操作数。

FOR 和 NEXT 之间的程序段称为循环体,每执行一次循环体,当前计数值增 1,并且将其结果同终值作比较,如果大于终值,则终止循环。

循环指令的 LAD 和 STL 形式如图 5-23 所示。

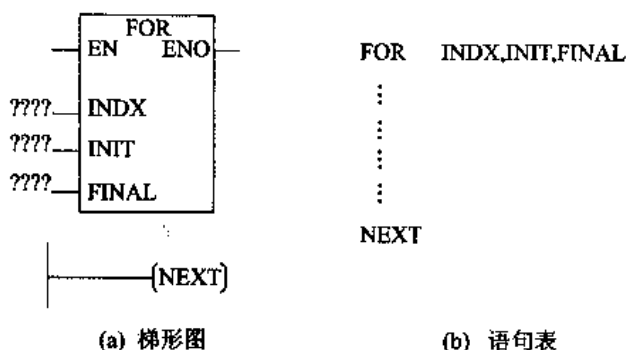


图 5-23 循环指令的 LAD 和 STL 形式

#### 2. 参数说明

从图 5-23(a)中可以看出,循环指令盒中有三个数据输入端:当前循环计数 INDX(index value or current loop count)、循环初值 INIT(starting value)和循环终值 FINAL(ending value)。在使用时必须给 FOR 指令指定当前循环计数(INDX)、初值(INIT)和终值(FINAL)。

INDX 操作数:VW、IW、QW、MW、SW、SMW、LW、T、C、AC、\* VD、\* AC 和 \* CD。这些操作数属 INT 型。

INIT 和 FINAL 操作数:VW、IW、QW、MW、SW、SMW、LW、T、C、AC、常数、\* VD、\* AC 和 \* CD。这些操作数属 INT 型。

循环指令使用举例如图 5-24 所示。当 I1.0 接通时,标为 A 的外层循环执行 100 次。当 I1.1 接通时,标为 B 的内层循环执行 2 次。

使用说明:

- (1) FOR、NEXT 指令必须成对使用。
- (2) FOR 和 NEXT 可以循环嵌套,嵌套最多为 8 层,但各个嵌套之间一定不可有交叉现象。
- (3) 每次使能输入(EN)重新有效时,指令将自动复位各参数。
- (4) 初值大于终值时,循环体不被执行。
- (5) 在使用循环指令时,要注意在循环体中对 INDX 的控制,这一点非常重要。

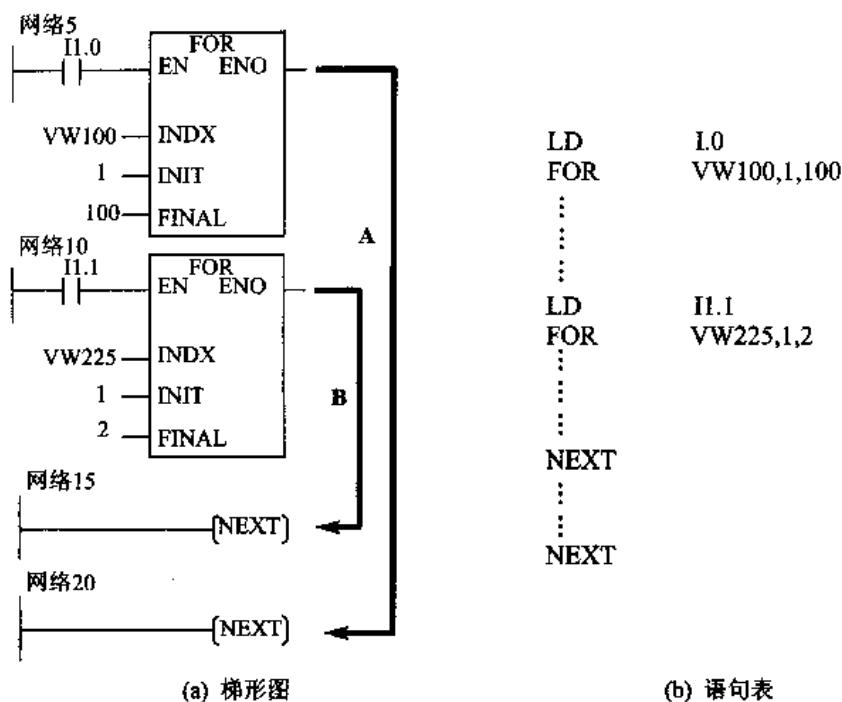


图 5-24 循环指令使用举例

### 5.2.5 子程序

子程序在结构化程序设计中是一种方便有效的工具。S7-200 PLC 的指令系统具有简单、方便、灵活的子程序调用功能。与子程序有关的操作有：建立子程序、子程序的调用和返回。

#### 1. 建立子程序

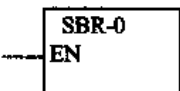
建立子程序是通过编程软件来完成的。可用编程软件“编辑”菜单中的“插入”选项选择“子程序”，以建立或插入一个新的子程序，同时，在指令树窗口可以看到新建的子程序图标，默认的程序名是 SBR\_N，编号 N 从 0 开始按递增顺序生成，也可以在图标上直接更改子程序的程序名，把它变为更能描述该子程序功能的名字。在指令树窗口双击子程序的图标就可进入子程序，并对它进行编辑。对于 CPU 226XM，最多可以有 128 个子程序；对其余的 CPU，最多可以有 64 个子程序。

#### 2. 子程序的调用

##### (1) 子程序调用指令(CALL)

在使能输入有效时，主程序把程序控制权交给子程序。子程序的调用可以带参数，也可以不带参数。它在梯形图中以指令盒的形式编程。指令格式如表 5-10 所列。

表 5-10 子程序调用指令格式

指 令	子程序调用指令	子程序条件返回指令
LAD		—(RET)
STL	CALL SBR_0	CRET

## (2) 子程序条件返回指令(CRET)

在使能输入有效时,结束子程序的执行,返回主程序中(返回到调用此子程序的下一条指令)。梯形图中以线圈的形式编程,指令不带参数。指令格式如表 5-10 所列。

## (3) 应用举例

图 5-25 所示的程序实现用外部控制条件分别调用两个子程序。

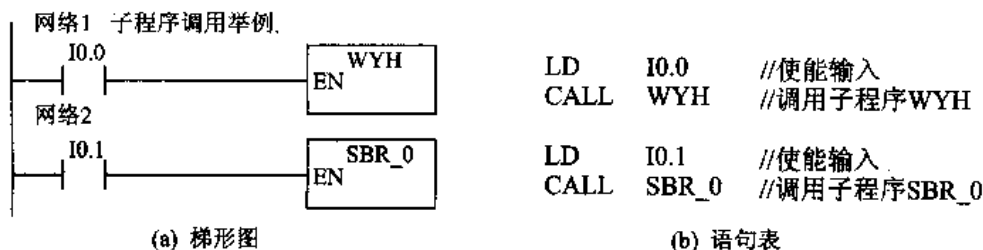


图 5-25 子程序调用举例

使用说明:

① CRET 多用于子程序的内部,由判断条件决定是否结束子程序调用,RET 用于子程序的结束。用 Micro/Win32 编程时,编程人员不需要手工输入 RET 指令,而是由软件自动在内部加到每个子程序结尾。

② 如果在子程序的内部又对另一子程序执行调用指令,则这种调用称为子程序的嵌套。子程序的嵌套深度最多为 8 级。

③ 当一个子程序被调用时,系统自动保存当前的堆栈数据,并把栈顶置 1,堆栈中的其他值为 0,子程序占有控制权。子程序执行结束,通过返回指令自动恢复原来的逻辑堆栈值,调用程序又重新取得控制权。

④ 累加器可在调用程序和被调用子程序之间自由传递,所以累加器的值在子程序调用时既不保存也不恢复。

⑤ 子程序的使用参见第七章、第八章和第九章的有关例题。

### 3. 带参数的子程序调用

子程序中可以有参变量,带参数的子程序调用极大地扩大了子程序的使用范围,增加了调用的灵活性。它主要用于功能类似的子程序块的编程。子程序的调用过程如果存在数据的传递,则在调用指令中应包含相应的参数。

#### (1) 子程序参数

子程序最多可以传递 16 个参数。参数在子程序的局部变量表中加以定义。参数包含下列信息:变量名、变量类型和数据类型。

① 变量名 变量名最多用 8 个字符表示,第一个字符不能是数字。

② 变量类型 变量类型是按变量对应数据的传递方向来划分的,可以是传入子程序(IN)、传入和传出子程序(IN/OUT)、传出子程序(OUT)和暂时变量(TEMP)等 4 种类型。4 种变量类型的参数在变量表中的位置必须按以下先后顺序。

IN 类型:传入子程序参数。参数可以是直接寻址数据(如 VB100)、间接寻址数据(如 \*AC1)、立即数(如 16#2344)或数据的地址值(如 &VB106)。

IN/OUT 类型:传入和传出子程序参数。调用时将指定参数位置的值传到子程序,返回

时从子程序得到的结果值被返回到同一地址。参数可以采用直接和间接寻址,但立即数(如 16 #1234)和地址值(如 &VB100)不能作为参数。

OUT 类型:传出子程序参数。它将从子程序返回的结果值送到指定的参数位置。输出参数可以采用直接和间接寻址,但不能是立即数或地址编号。

TEMP 类型:暂时变量参数。在子程序内部暂时存储数据,但不能用来与调用程序传递参数数据。

③ 数据类型 局部变量表中还要对数据类型进行声明。数据类型可以是:能流、布尔型、字节型、字型、双字型、整数型、双整型和实型。

能流:仅允许对位输入操作,是位逻辑运算的结果。在局部变量表中布尔能流输入处于所有类型的最前面。

布尔型:布尔型用于单独的位输入和输出。

字节、字和双字型:这 3 种类型分别声明一个 1 字节、2 字节和 4 字节的无符号输入或输出参数。

整数、双整数型:这 2 种类型分别声明一个 2 字节或 4 字节的有符号输入或输出参数。

实型:该类型声明一个 IEEE 标准的 32 位浮点参数。

#### (2) 参数子程序调用的规则

① 常数参数必须声明数据类型。例如,把值为 223 344 的无符号双字作为参数传递时,必须用 DW #223 344 来指明。如果缺少常数参数的这一描述,常数可能会被当作不同类型使用。

② 输入或输出参数没有自动数据类型转换功能。例如,局部变量表中声明一个参数为实型,而在调用时使用一个双字,则子程序中的值就是双字。

③ 参数在调用时必须按照一定的顺序排列,先是输入参数,然后是输入输出参数,最后是输出参数和暂时变量。

#### (3) 变量表的使用

按照子程序指令的调用顺序,参数值分配给局部变量存储器,起始地址是 L0.0。使用编程软件时,地址分配是自动的。在局部变量表中要加入一个参数,单击要加入的变量类型区可以得到一个选择菜单,选择“插入”,然后选择“下一行”即可。局部变量表使用局部变量存储器。

当在局部变量表中加入一个参数时,系统自动给各参数分配局部变量存储空间。

参数子程序调用指令格式:CALL 子程序名,参数 1,参数 2,...,参数 n。

#### (4) 程序实例

图 5-26 为一个带参数调用的子程序实例,其局部变量分配如表 5-11 所列。

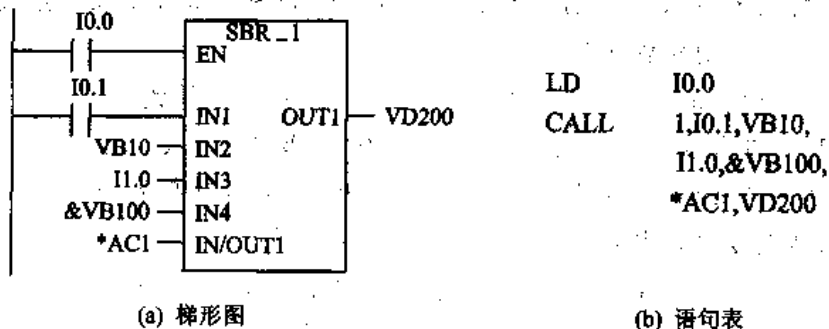


图 5-26 带参数子程序调用举例

表 5-11 局部变量表

L 地址	参数名	参数类型	数据类型	说 明
无	EN	IN	BOOL	指令使能输入参数
I0.0	IN1	IN	BOOL	第 1 个输入参数,布尔型
LB1	IN2	IN	BYTE	第 2 个输入参数,字节型
LB2.0	IN3	IN	BOOL	第 3 个输入参数,布尔型
LD3	IN4	IN	DWORD	第 4 个输入参数,双字型
LW7	IN/OUT1	IN/OUT	WORD	第 1 个输入/输出参数,字型
LD9	OUT1	OUT	DWORD	第 1 个输出参数,双字型

说明:图 5-26(b)的 STL 程序并不是从图 5-26(a)转换过来的,而是单独编写的。同样从图 5-26(b)也转换不成图 5-26(a)。编程软件使用 LB60~LB63 保存调用参数数据,所以在编程时要使用 LB60~LB63 中的一些位(如 LB60.0),才能实现带参数子程序的程序格式的转换。请参考 S7-200 系统手册。

### 5.2.6 与 ENO 指令

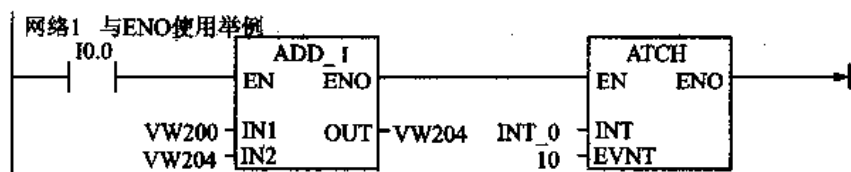
ENO 是 LAD 中指令盒的布尔能流输出端。如果指令盒的能流输入有效,则执行没有错误,ENO 就置位,并将能流向下传递。ENO 可以作为允许位表示指令成功执行。

STL 指令没有 EN 输入,但对要执行的指令,其栈顶值必须为 1。可用“与”ENO(AENO)指令来产生和指令盒中的 ENO 位相同的功能。

指令格式:AENO

AENO 指令无操作数,且只在 STL 中使用,它将栈顶值和 ENO 位的逻辑进行与运算,运算结果保存到栈顶。

AENO 指令使用较少。AENO 指令的用法如图 5-27 所示。



(a) 梯形图

```

LD      I0.0           //使能输入
+I      VW200,VW204    //整数加法,VW200+VW204=VW204
AENO                    //与ENO指令
ATCH    INT_0,10       //如果+I指令执行正确,则调用中断程序INT-0,中断事件号为10

```

(b) 语句表

图 5-27 AENO 指令用法举例

## 5.3 PLC 初步编程指导

### 5.3.1 梯形图编程的基本规则

梯形图编程的基本规则如下:

(1) PLC 内部元器件触点的使用次数是无限制的。

(2) 梯形图的每一行都是从左边母线开始,然后是各种触点的逻辑连接,最后以线圈或指令盒结束。触点不能放在线圈的右边,如图 5-28 所示。但如果是以有能量传递的指令盒结束时,可以使用 AENO 指令在其后面连接指令盒(较少使用),如图 5-27 所示。

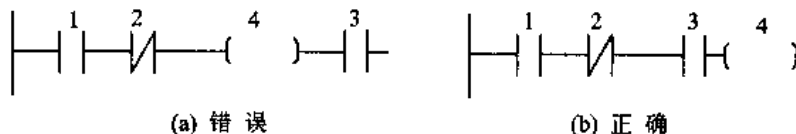


图 5-28 梯形图画法示例 1

(3) 线圈和指令盒一般不能直接连接在左边的母线上,如需要的话可通过特殊的中间继电器 SM0.0(常 ON 特殊中间继电器)完成,如图 5-29 所示。

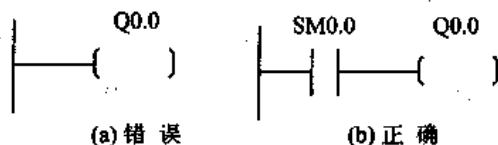


图 5-29 梯形图画法示例 2

(4) 在同一程序中,同一编号的线圈使用两次及两次以上称为双线圈输出。双线圈输出非常容易引起误动作,所以应避免使用。S7-200 PLC 中不允许双线圈输出。

(5) 在手工编写梯形图程序时,触点应画在水平线上,从习惯和美观的角度来讲,不要画在垂直线上,如图 5-30 所示。使用编程软件则不可能把触点画在垂直线上。

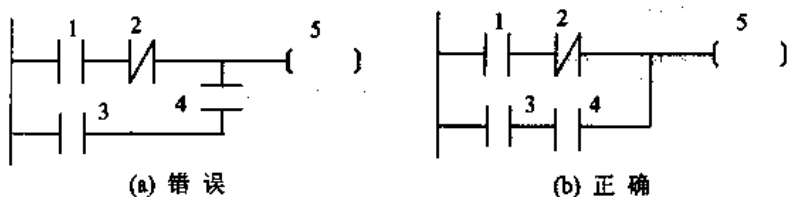


图 5-30 梯形图画法示例 3

(6) 不包含触点的分支线条应放在垂直方向,不要放在水平方向,以便于读图和图形的美观,如图 5-31 所示。使用编程软件则不可能出现这种情况。

(7) 应把串联多的电路块尽量放在最上边,把并联多的电路块尽量放在最左边,这样一是节省指令,二是美观,如图 5-32 所示。

(8) 图 5-33 所示为梯形图的推荐画法。

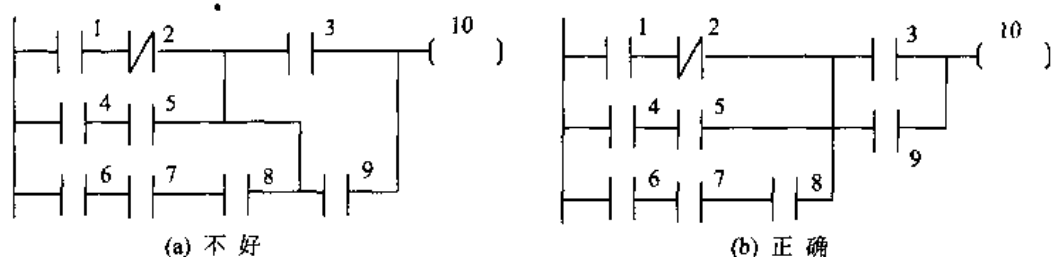
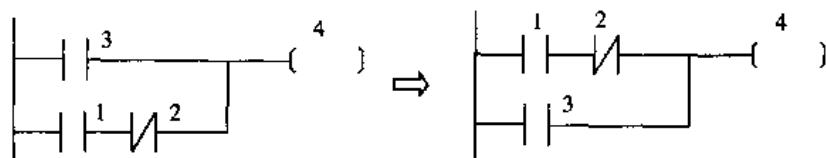
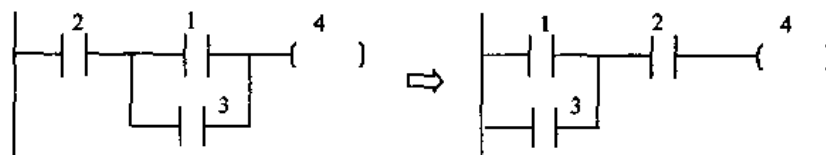


图 5-31 梯形图画法示例 4



(a) 把串联多的电路块放在最上边



(b) 把并联多的电路块放在最左边

图 5-32 梯形图画法示例 5

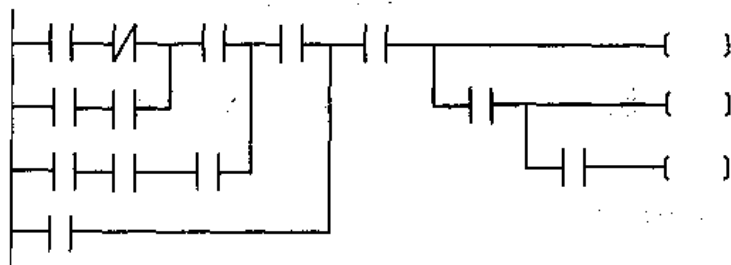


图 5-33 梯形图的推荐画法

### 5.3.2 LAD 和 STL 编程形式的区别

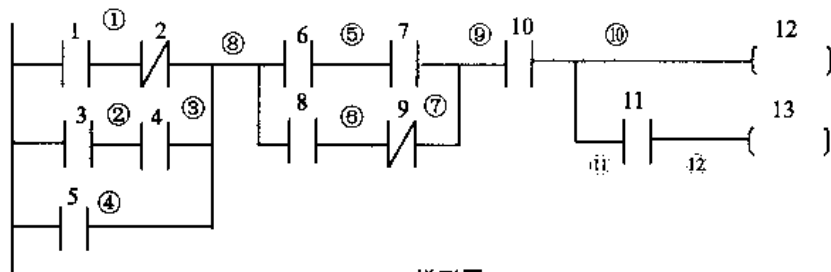
利用 PLC 指令对梯形图编程时,可以把整个梯形图程序看成由许多网络块组成,每个网络块均起始于母线。所有的网络块组合在一起就是梯形图程序,这是 S7-200 PLC 的特点。LAD 程序可以通过编程软件直接转换为 STL 形式。S7-200 PLC 用 STL 编程时,如果也以每个独立的网络块为单位,则 STL 程序和 LAD 程序基本上是一一对应的,而且两者可以通过编程软件相互转换;如果不以每个独立的网络块为单位编程,而是连续编写,则 STL 程序和 LAD 程序不能通过编程软件相互转换。大家在使用时要注意。

LAD 是使用最多的编程语言,它非常直观易懂,对每个人都适用;特别老练的人在某些情况下会直接使用 STL 形式编程,用 STL 形式编写的程序简短,但不直观。

但不管怎么说,大家对这两种形式的程序都应该熟悉。特别是会用手工把一个 LAD 程

序转换为 STL 程序,这对进一步理解 PLC 程序执行的原理有很大帮助。下面的一个典型例子说明了从 LAD 到 STL 的转换步骤。

对每一个独立的 LAD 网络块中的程序,可分成若干小块,对每个小块按照从左到右、从上到下的原则进行编程。然后将程序块联接起来,就完成了该网络块的 STL 编程。图 5-34 详细介绍了语句表编程的步骤。



(a) 梯形图

① LD	1	⑥ LD	8
AN	2	AN	9
② LD	3	⑦ OLD	
A	4	⑧ ALD	
③ OLD		⑨ A	10
④ O	5	⑩ =	12
⑤ LD	6	⑪ A	11
A	7	⑫ =	13

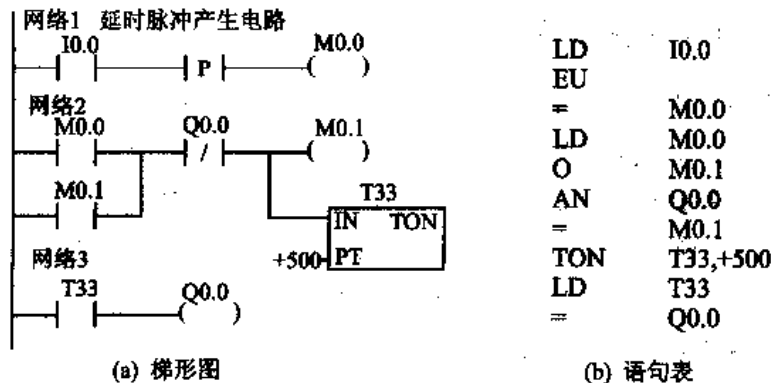
(b) 语句表编写顺序示范

图 5-34 语句表编程举例

## 5.4 典型的简单电路编程

### 5.4.1 延时脉冲产生电路

要求在有输入信号后,停一段时间产生一个脉冲。该电路常用在获取启动或关断信号。如图 5-35 所示为该电路的程序及时序图。



(a) 梯形图

LD	I0.0
EU	
=	M0.0
LD	M0.0
O	M0.1
AN	Q0.0
=	M0.1
TON	T33,+500
LD	T33
=	Q0.0

(b) 语句表

图 5-35 延时脉冲产生电路

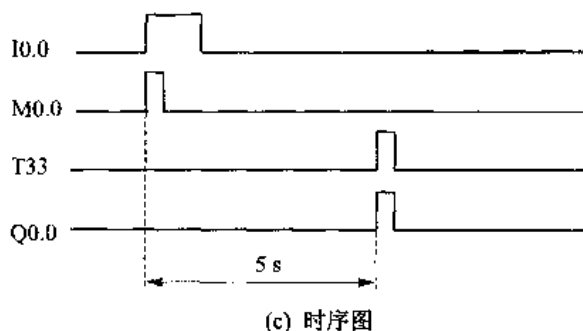


图 5-35 延时脉冲产生电路(续)

图 5-35 中利用脉冲指令在 I0.0 的上升沿产生一个计时启动脉冲,接下来就是一个非常典型的环节。因为定时器没有瞬动触点,不可能用自身的触点组成自锁回路,所以必须用一个中间继电器 M0.1 组成延时逻辑。T33 定时到时,产生一个宽度为一个扫描周期的脉冲 Q0.0,然后 Q0.0 使 T33 复位。

#### 5.4.2 瞬时接通/延时断开电路

该电路要求在输入信号有效时,马上有输出,而输入信号 OFF 后,输出信号延时一段时间才 OFF,图 5-36 所示为该电路的程序及时序图。

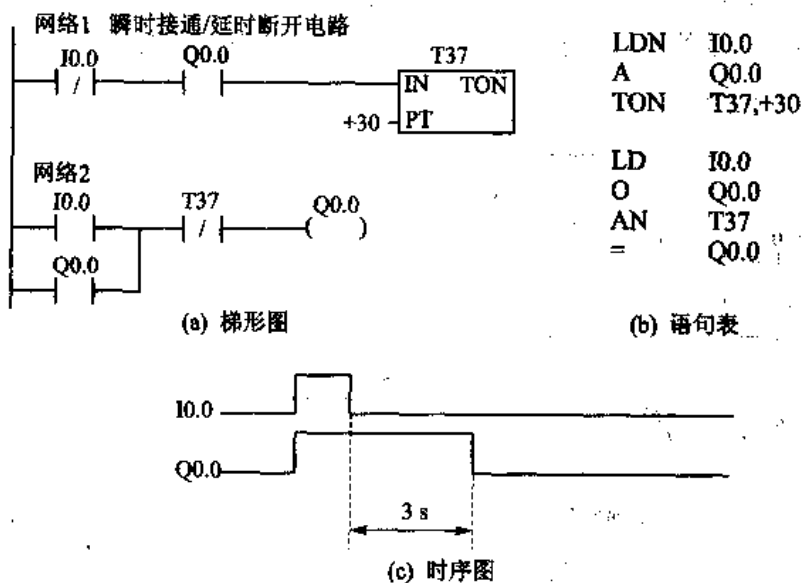


图 5-36 瞬时接通/延时断开电路

图 5-36 中,关键的问题是找出定时器 T37 的计时条件。本例中 T37 的计时条件是 I0.0 为 OFF 且 Q0.0 为 ON。因为 I0.0 变为 OFF 后, Q0.0 仍要保持通电状态 3 s,所以 Q0.0 的自锁触点是必须的。

#### 5.4.3 延时接通/延时断开电路

该电路要求有输入信号后,停一段时间输出信号才为 ON;而输入信号 OFF 后,输出信号延时一段时间才 OFF,如图 5-37 所示为该电路的程序及时序图。

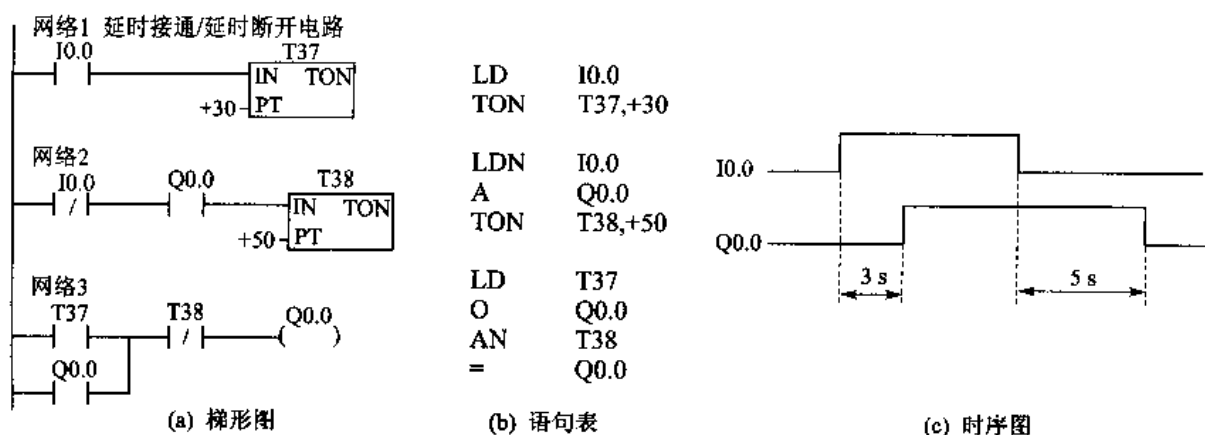


图 5-37 延时接通/延时断开电路

和瞬时接通/延时断开电路相比,该电路多加了一个输入延时。T37 延时 3 s 作为 Q0.0 的启动条件,T38 延时 5 s 作为 Q0.0 的关断条件。两个定时器配合使用实现该电路的功能。

#### 5.4.4 脉冲宽度可控制电路

在输入信号宽度不规范的情况下,要求在每一个输入信号的上升沿产生一个宽度固定的脉冲,该脉冲宽度可以调节。需要说明的是,如果输入信号的两个上升沿之间的距离小于该脉冲宽度,则忽略输入信号的第二个上升沿,图 5-38 所示为该电路的程序及时序图。

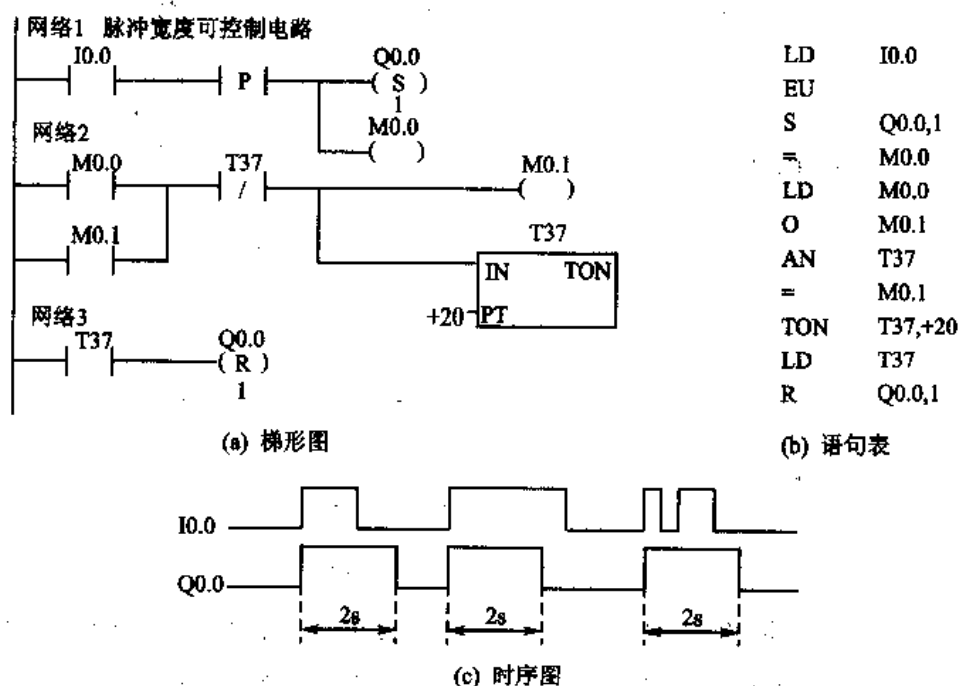


图 5-38 脉冲宽度可控制电路

该例中,我们使用了上升沿脉冲指令和 S/R 指令。关键是找出 Q0.0 的开启和关断条件,使其不论在 I0.0 的宽度大于或小于 2 s 时,都可使 Q0.0 的宽度为 2 s。定时器 T37 的计时输入逻辑在上升沿之间的距离小于该脉冲宽度时,对后产生的上升沿脉冲无效。T37 在计时到

后产生一个信号复位 Q0.0, 然后自己复位。该例中, 通过调节 T37 设定值 PT 的大小, 就可控制 Q0.0 的宽度。该宽度不受 I0.0 接通时间长短的影响。

### 5.4.5 计数器的扩展

如前所述, 一个计数器最大计数值为 32 767。在实际应用中, 如果计数范围超过该值, 就需要对计数器的计数范围进行扩展, 图 5-39 为计数器扩展电路的程序。

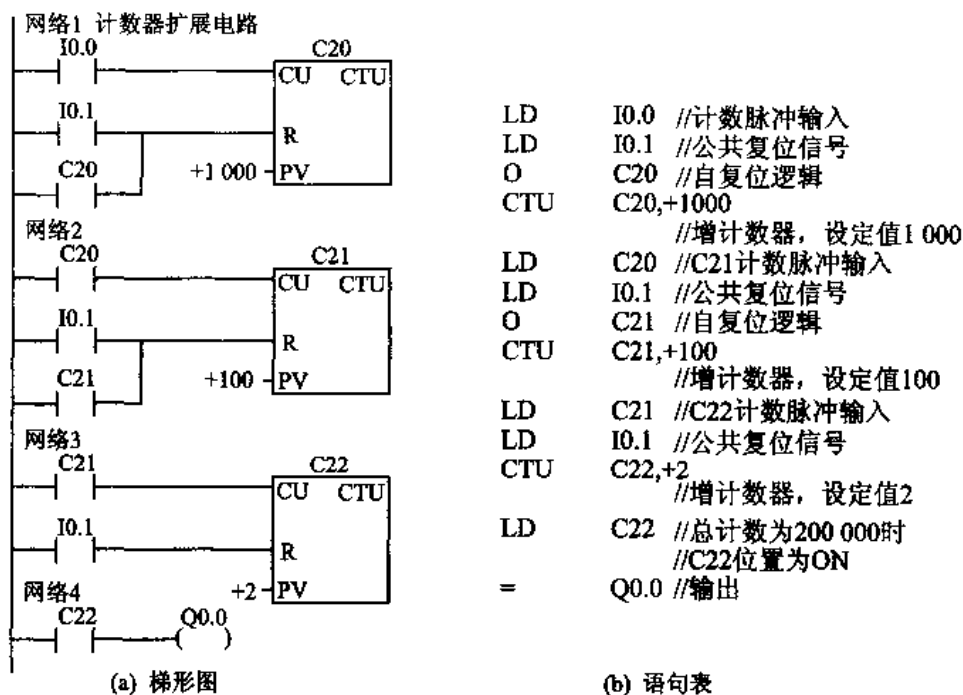


图 5-39 计数器的扩展电路

在图 5-39 中, 计数信号为 I0.0, 它作为 C20 的计数端输入信号, 每一个上升沿使 C20 计数 1 次; C20 的常开触点作为计数器 C21 的计数输入信号, C20 计数到 1 000 时, 使计数器 C21 计数 1 次; C21 的常开触点作为计数器 C22 的计数输入信号, C21 每计数到 100 时, C22 计数 1 次。这样当  $C_{\Sigma} = 1\,000 \times 100 \times 2 = 200\,000$  时, 即当 I0.0 的上升沿脉冲数到 200 000 时, Q0.0 才被置位。

使用时, 应注意计数器复位输入端逻辑的设计, 要保证能准确及时复位。该例中, I0.1 为外置公共复位信号。C20 计数到 1 000 时, 在使计数器 C21 计数 1 次之后的下一个扫描周期, 它的常开触点使自己复位; 同理, C21 计数到 100 时, 在使计数器 C22 计数 1 次之后的下一个扫描周期, 它的常开触点使自己复位。

### 5.4.6 长定时电路

S7-200 PLC 中的定时器最长定时时间不到 1 h, 但在一些实际应用中, 往往需要几小时甚至几天或更长时间的定时控制, 这样仅用一个定时器就不能完成该任务。下例表示在输入信号 I0.0 有效后, 经过 10 h 30 min 后将输出 Q0.0 置位。图 5-40 为该电路的梯形图程序。

在该例中, T37 每一分钟产生一个脉冲, 所以是分钟计时器。C21 每小时产生一个脉冲, 故 C21 为小时计时器。当 10 h 计时到时, C22 为 ON, 这时 C23 再计时 30 min, 则总的定时时

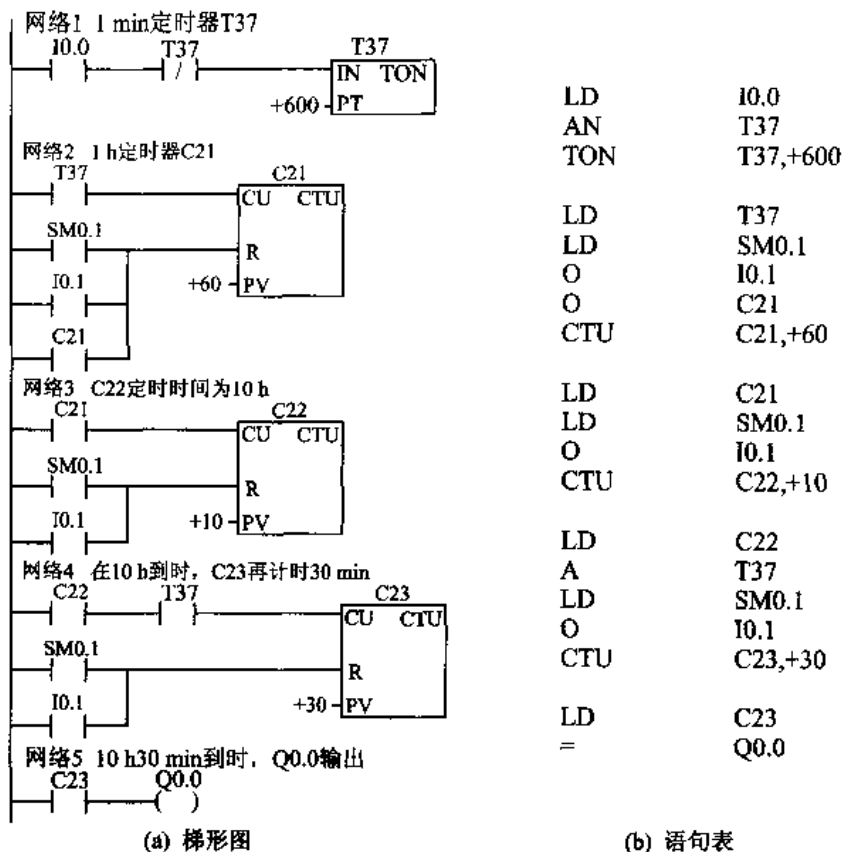


图 5-40 长延时电路

间为 10 h 30 min, Q0.0 置位成 ON。

在该例的计数器复位逻辑中,有初始化脉冲 SM0.1 和外部复位按钮 I0.1。初始化脉冲完成在 PLC 上电时对计数器的复位操作,如果所使用的计数器不是设置为掉电保护模式,则不需要初始化复位。另外,图中的 C21 有自复位功能。

在定时时间很长,定时精度要求不高的场合,如小于 1 s 或 1 min 的误差可以忽略不计,则可以使用时钟脉冲 SM0.4(1 min 脉冲)或 SM0.5(1 s 钟脉冲)来构成长延时电路。在学习完以后的“加 1 指令”后,也可以用功能指令完成长延时电路的程序设计。

#### 5.4.7 闪烁电路

闪烁电路也称为振荡电路,该电路用在报警、娱乐等场合。闪烁电路实际上就是一个时钟电路。它可以是等间隔的通断,也可以是不等间隔的通断。图 5-41 为一个典型闪烁电路的程序及时序图。在该例中,当 I0.0 有效时, T37 就会产生一个 1 s 通、2 s 断的闪烁信号。Q0.0 和 T37 一样开始闪烁。

在实际的程序设计中,如果电路用到闪烁功能,往往直接用两个定时器组成闪烁电路,如图 5-42 所示。这个电路不管其他信号如何,PLC 一经通电,它就开始工作。什么时候使用到闪烁功能时,把 T37 的常开触点(或常闭触点)串联上即可。通断的时间值可以根据需要任意设定。图 5-42 为一个 2 s 通、2 s 断的闪烁电路。

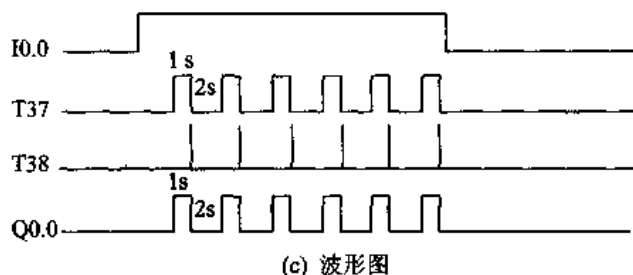
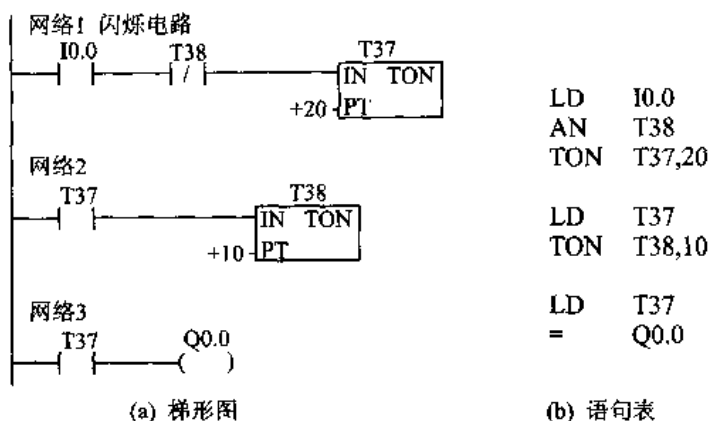


图 5-41 闪烁电路

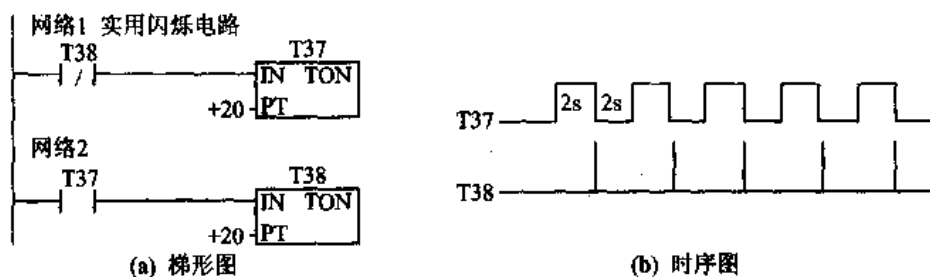


图 5-42 实际使用的闪烁电路

#### 5.4.8 报警电路

报警是电气自动控制中不可缺少的重要环节,标准的报警功能应该是声光报警。当故障发生时,报警指示灯闪烁,报警电铃或蜂鸣器鸣响。操作人员知道故障发生后,按消铃按钮,把电铃关掉,报警指示灯从闪烁变为长亮。故障消失后,报警灯熄灭。另外还应设置试灯、试铃按钮,用于平时检测报警指示灯和电铃的好坏。

图 5-43 为标准报警电路,图中的输入/输出信号地址分配如下:

输入信号 I0.0 为故障信号;I1.0 为消铃按钮;I1.1 为试灯、试铃按钮。

输出信号 Q0.0 为报警灯;Q0.7 为报警电铃。

在实际的应用系统中可能出现的故障一般有多种,这时的报警电路就不一样了。对报警指示灯来说,一种故障对应于一个指示灯,但一个系统只能有一个电铃。下面分析一个有两种故障的报警电路供大家在实际使用时参考。

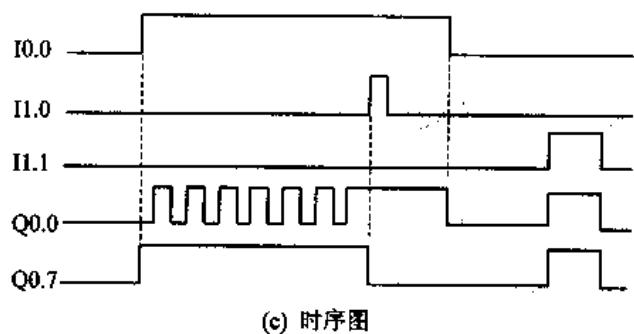
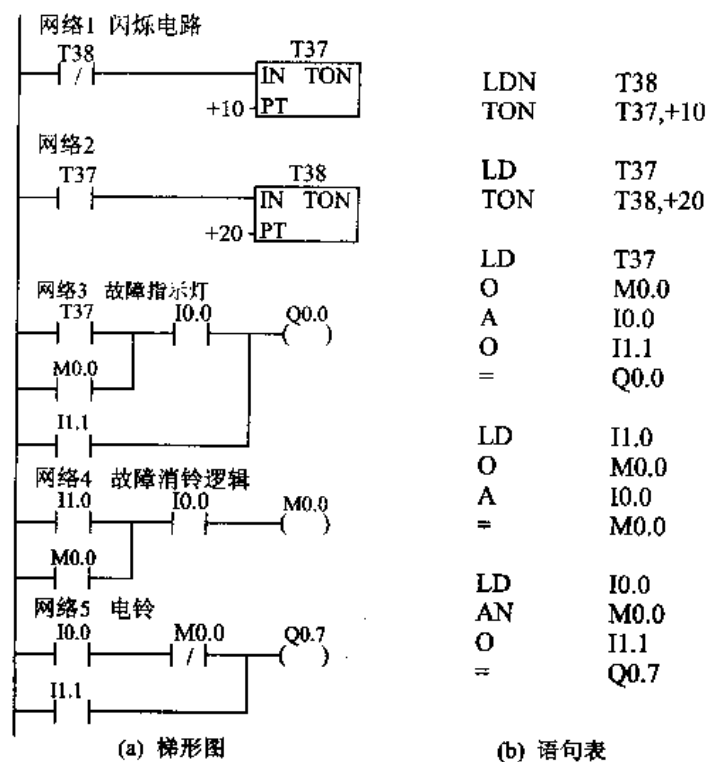


图 5-43 标准报警电路

图 5-44 为两种故障标准报警电路图,图中输入/输出信号地址的分配如下:

输入信号 I0.0 为故障 1;I0.1 为故障 2;I1.0 为消铃按钮;I1.1 为试灯、试铃按钮。

输出信号 Q0.0 为故障 1 指示灯;Q0.1 为故障 2 指示灯;Q0.7 为报警电铃。

在该程序的设计中,关键是当任何一种故障发生时,按消铃按钮后,不能影响其他故障发生时报警电铃的正常鸣响。

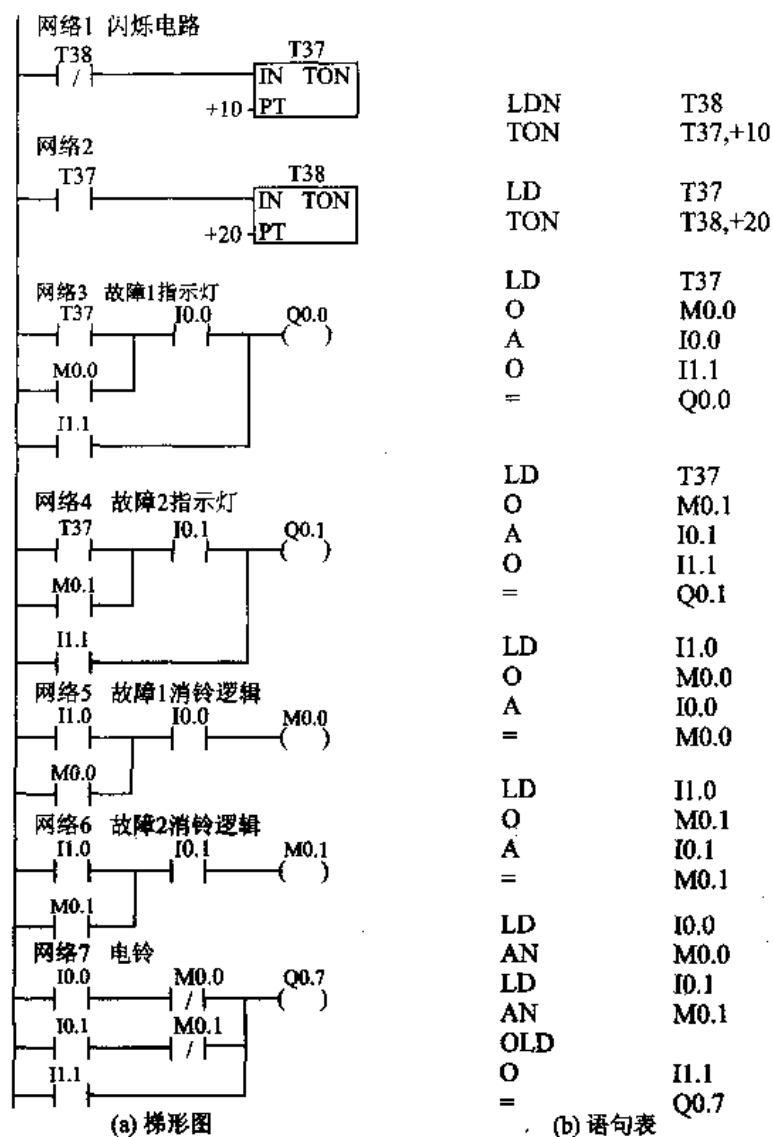


图 5-44 两种故障的标准报警电路

## 5.5 PLC 程序的简单设计法及应用举例

### 5.5.1 PLC 程序的简单设计法

PLC 的程序设计一般是凭设计者的经验来完成的。从事 PLC 程序设计时间越长的技术人员,其设计程序的速度也就越快,而且设计出的程序质量也越高。所有这一切都是靠长时间的探索积累换来的,所以经验设计法并不适合初学者使用。有关 PLC 的书籍和文献也介绍过 PLC 程序设计的其他方法,如状态表法、功能图法、流程图法等。对初学者来说这些方法也有一定的难度,本书将在第七章中介绍功能图法。在这里结合 2.6 节电气线路图的简单设计法,也给大家总结一个 PLC 程序的简单设计法,供参考使用。

从 2.6 节可知,在没有约束条件下,典型输出控制对象的基本逻辑函数可表示为

$$F_k = (X_{\text{开}} + K) X_{\text{关}} \quad (5-1)$$

式中:  $K$  为控制对象的当前状态,  $F_k$  为下一个状态值。在电气原理图中或梯形图中,  $K$  其实就是自锁触点,  $F_k$  就是输出线圈。  $X_{\text{开}}$  为启动条件,  $X_{\text{关}}$  为关断条件。为了安全性和可靠性, 要求  $X_{\text{开}}$  和  $X_{\text{关}}$  为短信号。具有启动和关断约束条件的输出对象的逻辑函数可表示为

$$F_k = (X_{\text{开}} \cdot X_{\text{开约}} + K)(X_{\text{关}} + X_{\text{关约}}) \quad (5-2)$$

式中:  $X_{\text{开约}}$  为启动约束条件,  $X_{\text{关约}}$  为关断约束条件。同理, 也要求  $X_{\text{开约}}$  和  $X_{\text{关约}}$  为短信号。

因为  $K$  是  $F_k$  的自锁触点, 所以式(5-1)和式(5-2)中的自锁触点  $K$  在电气原理图中和 PLC 的梯形图程序中就用  $F_k$  表示。对 PLC 来说, 设所有的输入信号均为常开触点接到 PLC 的输入端子, 则式(5-1)和式(5-2)对应的 PLC 梯形图程序如图 5-45(a)和(b)所示。当然启动或关断的约束条件不一定同时都有, 有时也可能有多个启动或关断约束条件, 大家在使用时应按照本书所讲的基本原则, 具体问题具体分析。

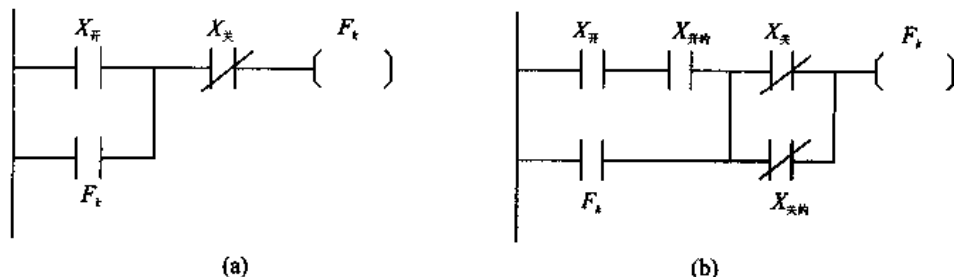


图 5-45 PLC 程序简单设计法的梯形图程序

PLC 的编程原理基本上同继电接触式系统的电气原理图设计, 所以对于 PLC 控制系统中的输出对象基本上可以按上面的方法来设计程序。不论是电气控制系统还是 PLC 控制系统, 编程的最终目的是控制输出对象, 输出对象的问题解决了, 基本的编程任务就完成了。

当然, 在编程时, PLC 与继电器相比有其特殊性和优越性, 这主要体现在:

- (1) 内部元器件的触点可以无限制地使用;
- (2) 大部分情况下, 基本上可以不考虑逻辑元器件的使用浪费问题;
- (3) 利用软件编程很容易找出控制对象的启动和关断所需要的短信号。

PLC 的这些特点在某些时候虽然增加了程序的长度, 但却大大方便了程序设计人员, 使他们能够设计出清晰、可靠的程序。

PLC 简单程序设计法的一般步骤和要求归纳如下:

- (1) 找出输出对象的启动条件和关断条件, 为了提高可靠性, 要求它们最好是短脉冲信号;
- (2) 如果该输出对象的启动或关断有约束条件, 则找出约束条件;
- (3) 一般情况下, 输出对象按照图 5-45(a)编程; 有约束条件时, 按图 5-45(b)编程;
- (4) 对程序进行全面检查和修改。

### 5.5.2 应用举例

**【例 5-1】** 电机顺序启/停电路。仍以 2.7.3 节的例题为例, 以便大家进行比较, 体会一下 PLC 程序设计的灵活和方便。

**要求** 3 台电机按启动按钮后,  $M1$ 、 $M2$ 、 $M3$  正序启动; 按停止按钮后, 逆序停止。动作之间要有一定间隔。

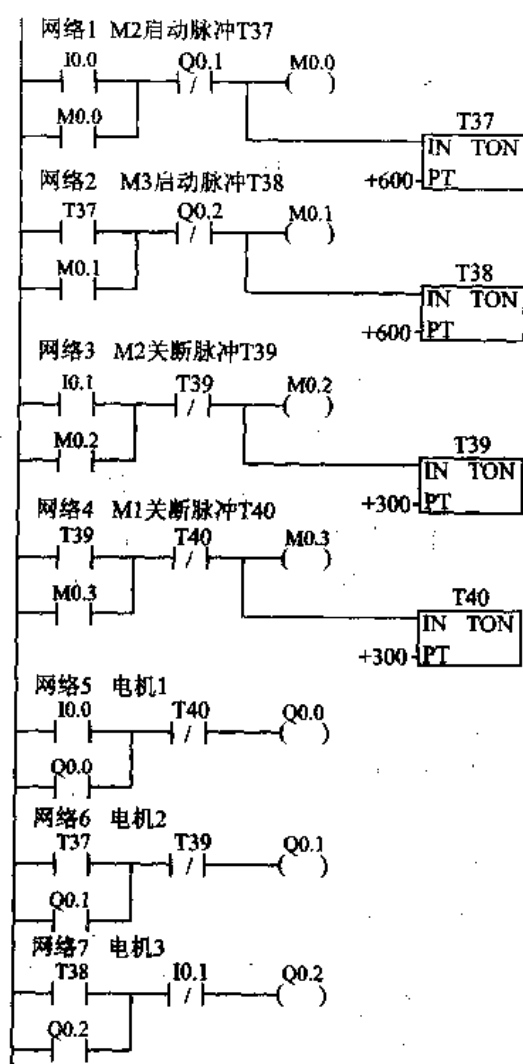
分析 先把题目中的输入/输出点找出来,分配好对应的 PLC 的 I/O 地址。

该题中有 2 个输入点,即启动按钮 I0.0,停止按钮 I0.1;有 3 个输出点,即电机 M1 的 Q0.0、电机 M2 的 Q0.1 和电机 M3 的 Q0.2。

注意:PLC 输出点实际上控制的是每个电机的接触器线圈。

方法一:

图 5-46 为电机顺序启停电路的梯形图及程序。若让 3 台电机启动的时间间隔为 1 min,停止时间间隔为 30 s,从图中可以看出,M1 的启动条件是启动按钮 I0.0,M3 的停止条件是停止按钮 I0.1,其他的启动和停止条件都是定时器所产生的脉冲信号(1 个扫描周期)。T39、T40 是 100 ms 的定时器,所以可以使用自复位来产生脉冲信号,自复位可以使编程简单。所以建议使用定时器时,如果允许,则尽量用 100 ms 的定时器(见定时器的正确使用一节)。



(a) 梯形图

```
LD      I0.0
O       M0.0
AN      Q0.1
=       M0.0
TON     T37,+600
```

```
LD      T37
O       M0.1
AN      Q0.2
=       M0.1
TON     T38,+600
```

```
LD      I0.1
O       M0.2
AN      T39
=       M0.2
TON     T39,+300
```

```
LD      T39
O       M0.3
AN      T40
=       M0.3
TON     T40,+300
```

```
LD      I0.0
O       Q0.0
AN      T40
=       Q0.0
```

```
LD      T37
O       Q0.1
AN      T39
=       Q0.1
```

```
LD      T38
O       Q0.2
AN      I0.1
=       Q0.2
```

(b) 语句表

图 5-46 电机顺序启停程序 1

方法二:

图 5-47 是用比较指令编写的程序,在程序中电机的启动和关断信号均为短信号(不一定

是一个扫描周期)。

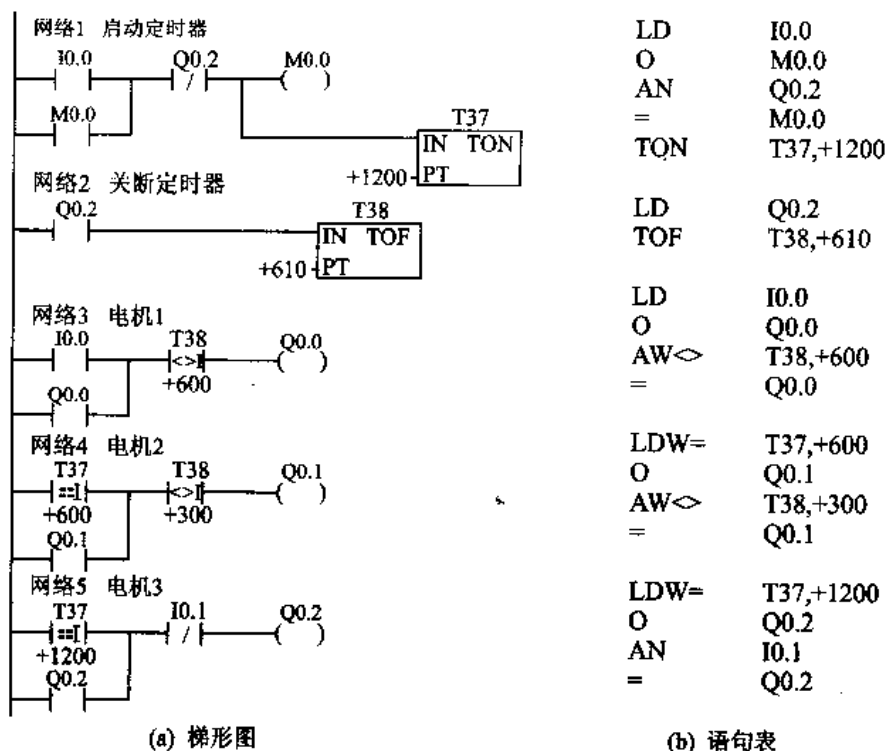


图 5-47 电机顺序启停程序 2

在图中,使用了一个断电延时定时器 T38,大家一定要了解断电延时定时器的特性,它计时到设定值后,当前值停在设定值处而不像通电延时定时器继续往前计时。所以 T38 的定时值我们设定为 610,这使得再次按启动按钮 I0.0 时,T38 不等于 600 的比较触点为闭合状态,M1 能够顺序启动。从图中也可以看出,使用一些复杂指令(如以后要讲的功能指令),可以使程序变得简单。

#### 【例 5-2】液体混合控制装置。

##### (1) 装置结构和工艺要求

图 5-48 为两种液体的混合装置结构图,SL1、SL2、SL3 为液位传感器,液面淹没时接通,两种液体(液体 A、液体 B)的流入和混合液体的流出分别由电磁阀 YV1、YV2、YV3 控制,M 为搅拌电机。控制要求如下:

① 初始状态 当装置投入运行时,容器内为放空状态。

② 启动操作 按下启动按钮 SB1,装置就开始按规定动作工作。液体 A 阀门打开,液体 A 流入容器。当液面到达 SL2 时,关闭液体 A 阀门,打开 B 阀门。当液面到达 SL3 时,关闭液体 B 阀门,搅拌电机开始转动。搅拌电机工作 1 min 后,停止搅动,混合液体阀门打开,开始放出混合液体。当液面下降到 SL1 时,SL1 由接通变为断开,再经过 20 s 后,容器放空,混合液体阀门 YV3 关闭,接着开始下一循环操作。

③ 停止操作 按下停止按钮后,要处理完当前循环周期剩余的任务后,系统停止在初始状态。

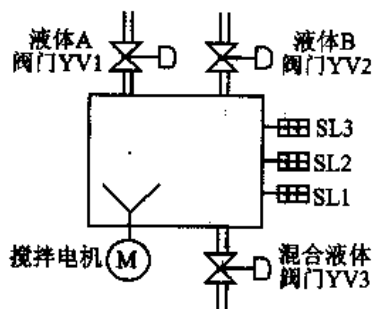


图 5-48 液体混合装置示意图

## (2) 系统输入/输出点及其对应的 PLC 地址

输入点:

启动按钮 SB1:I0.0

停止按钮 SB2:I0.1

液位传感器 SL1:I0.2

液位传感器 SL2:I0.3

液位传感器 SL3:I0.4

输出点:

液体 A 电磁阀 YV1:Q0.0

液体 B 电磁阀 YV2:Q0.1

搅拌电机接触器 KM:Q0.2

混合液体电磁阀 YV3:Q0.3

根据系统功能要求编写的 PLC 程序如图 5-49 所示。

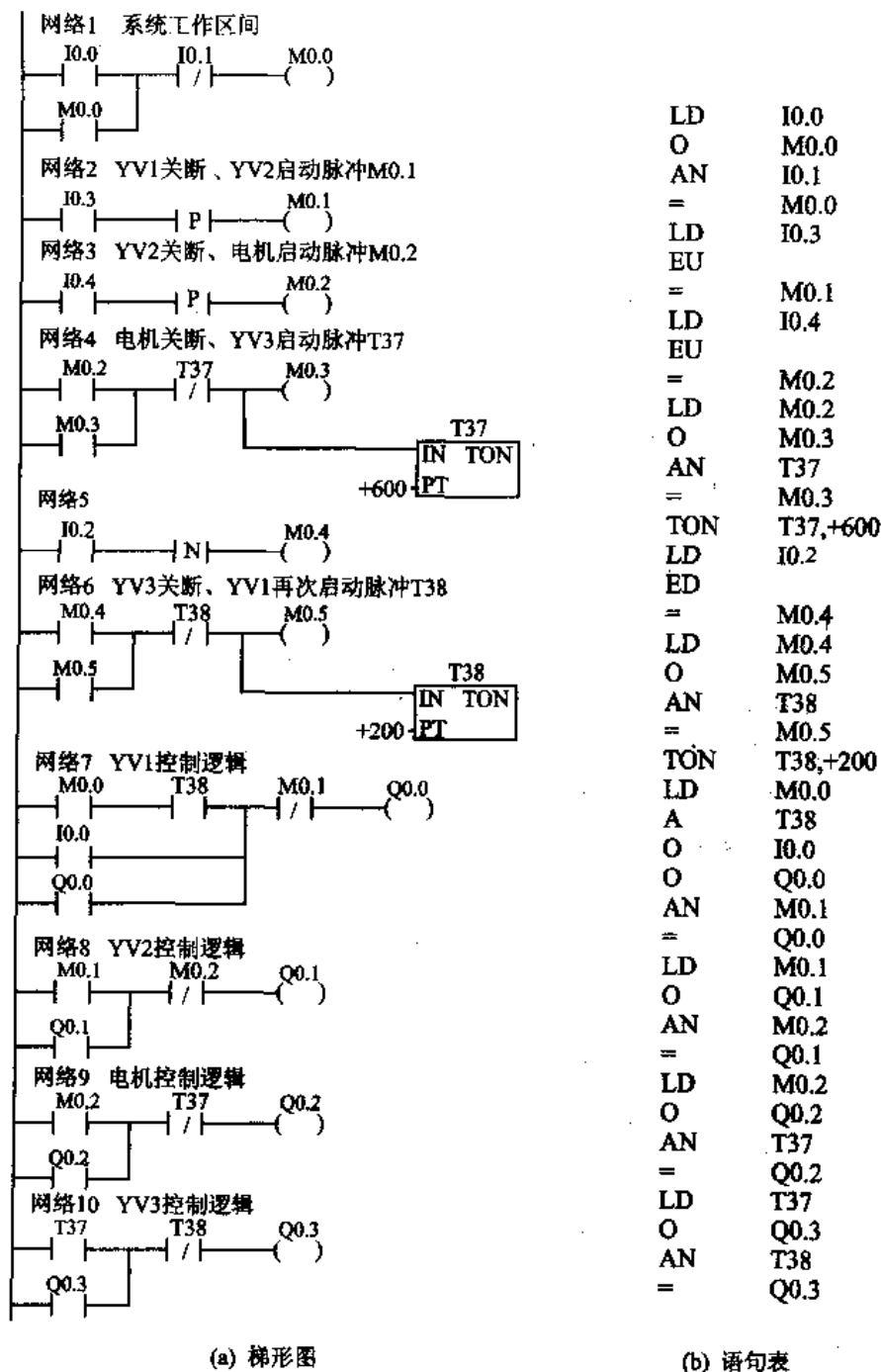


图 5-49 混合液体控制装置 PLC 程序

从该例中可以看出,对任何控制对象,如果准确地找出了它的可靠的开启和关断条件,则它的程序也就编写出来了。在该例中,YV1 的启动信号是启动按钮,关断信号是 SL2 的上升沿脉冲;YV2 的启动信号是 SL2 的上升沿脉冲,关断信号是 SL3 的上升沿脉冲;M 的启动信号是 SL3 的上升沿脉冲,关断信号是定时器 T37 计时到脉冲;YV3 的启动信号是定时器 T37 计时到脉冲,关断信号是定时器 T38 计时到脉冲。大家从使用中会体会到,启动及停止信号使用短脉冲信号,有效地避免了由于液面的波动所带来的不可靠隐患。需要注意的是,液体 A 阀 YV1 的启动条件除了启动按钮 I0.0 外,还有每次循环周期开始的启动条件 T38,而且 T38 还带有约束条件 M0.0。系统开始工作后,不按停止按钮 I0.1 时,M0.0 为 ON,在每次放完混合液体后,系统都可以自动进入新的工作循环。按过停止按钮 I0.1 后,M0.0 为 OFF,系统进行到最后一个动作,即混合液体放空后,由于  $M0.0 \cdot T38 = OFF$ ,所以不能进入新的循环,系统停止在初始状态。只有再次按下启动按钮后,系统才可重新开始工作。M0.0 的作用就像一个桥梁一样,不按停止按钮,桥梁处于接通状态;按过停止按钮后,桥就断了。另外,把  $M0.0 \cdot T38$  放在该网络块的最上边,则比较符合梯形图的编程规范。

## 本章小结

(1) 5.1 节重点讲解了 S7-200 PLC 的基本指令,这些指令是 PLC 编程的基础。要求大家熟练掌握各种指令在梯形图和语句表编程中的使用方法,理解指令的精髓,特别是要理解定时器和计数器的工作原理。

(2) 5.2 节重点讲解了 S7-200 PLC 的程序控制指令,这里的跳转、循环和子程序指令是常用指令,大家要知道跳转和循环时对元器件状态的影响。另外,编程软件会自动在主程序、子程序和中断程序结束时加上相应的结束指令,所以不需要人工处理。

(3) 5.3 节重点讲解了 PLC 编程的一些规则,这些规则很多是经验的总结。

(4) 5.4 节重点举例分析了多个典型环节的 PLC 程序,通过对典型电路的认识和学习,大家对 PLC 编程应有一个初步的掌握,大家应该记住一些典型环节的 PLC 程序。

(5) 5.5 节给大家总结的 PLC 程序的简单设计法能解决大部分编程问题,后面的举例是对简单设计法的进一步讲解,大家可在理解的基础上掌握。

## 思考题与练习题

1. 理解 S7-200 PLC 的程序设计和继电接触式控制系统电气原理图的区别。
2. S7-200 PLC 中共有几种分辨率的定时器? 它们的刷新方式有何不同? S7-200 PLC 中共有几种类型的定时器? 对它们执行复位指令后,它们的当前值和位的状态是什么?
3. S7-200 PLC 中共有几种形式的计数器? 对它们执行复位指令后,它们的当前值和位的状态是什么?
4. 写出图 5-50 所示梯形图的语句表。
5. 写出图 5-51 所示梯形图的语句表。
6. 已知输入信号 I0.0 的波形,画出图 5-52 梯形图程序中 M0.0、M0.1、M0.2 和 Q0.0 的波形。
7. 指出图 5-53 中的错误。
8. 设计图 5-54 所示二分频电路的梯形图程序。

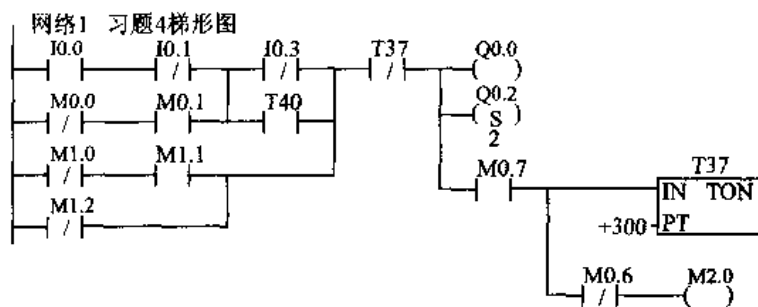


图 5-50 习题 4 梯形图程序

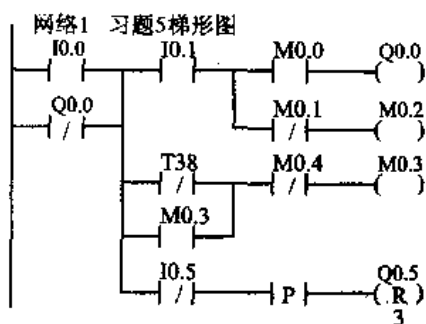


图 5-51 习题 5 梯形图程序

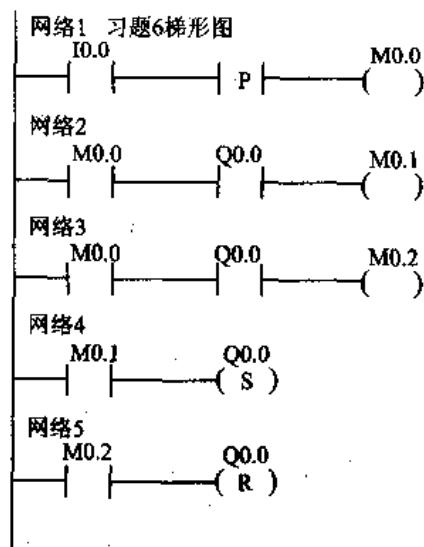


图 5-52 习题 6 的梯形图程序和输入信号波形

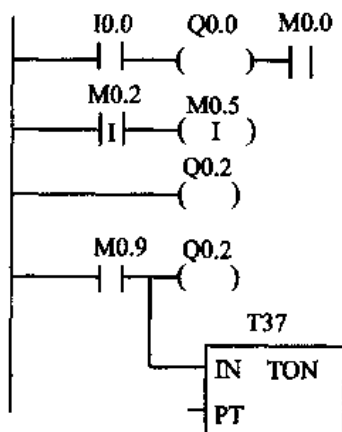


图 5-53 习题 7 梯形图

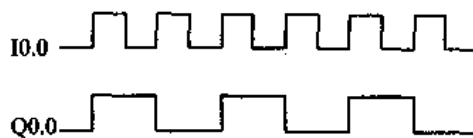


图 5-54 二分频电路波形图

9. 试设计一个 30 h 40 min 的长延时电路程序。

10. 试设计一个照明灯的控制程序。当按下接在 I0.0 上的按钮后,接在 Q0.0 上的照明灯可发光 30 s。

如果在这段时间内又有人按下按钮,则时间间隔从头开始。这样可确保在最后一次按完按钮后,灯光可维持 30 s 钟的照明。

11. 试设计一个抢答器电路程序。出题人提出问题,3 个答题人按动按钮,仅是最早按的人面前的信号灯亮。然后出题人按动复位按钮后,引出下一个问题。
12. 用简单设计法设计一个对锅炉鼓风机和引风机控制的梯形图程序。控制要求:
  - (1) 开机时首先启动引风机,10 s 后自动启动鼓风机;
  - (2) 停止时,立即关断鼓风机,经 20 s 后自动关断引风机。
13. 多个传送带启动和停止示意如图 5-55 所示。初始状态为各个电机都处于停止状态。按下启动按钮后,电动机 M1 通电运行,行程开关 SQ1 有效后,电动机 M2 通电运行,行程开关 SQ2 有效后,M1 断电停止。其他传动带动作类推。整个系统循环工作。按停止按钮后,系统把目前的工作进行完后停止在初始状态。

试设计其梯形图并写出语句表。

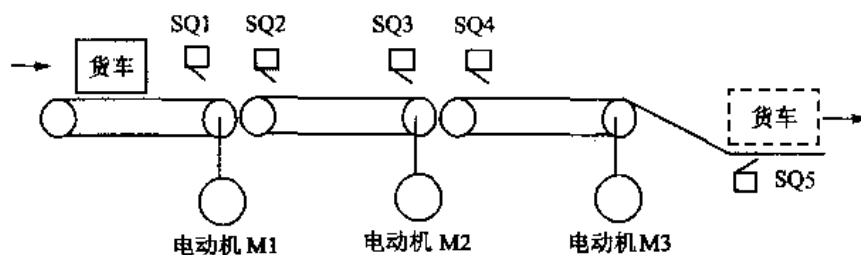


图 5-55 多个传送带控制示意图

14. 试设计第二章习题 11 的 PLC 控制系统程序,并和所设计的电气原理图进行比较。
15. 试设计第二章习题 12 的 PLC 控制系统程序,并和所设计的电气原理图进行比较。
16. 用简单设计法完成 5.4.4 一节中有关脉冲宽度可控制电路的梯形图编程。

## 第六章 S7-200 PLC 顺序控制指令及应用

### 6.1 功能图的产生及基本概念

#### 6.1.1 功能图的产生

应用上一章所介绍的基本指令和方法设计简单顺序控制问题的程序是可行的,但对于具有并发顺序和选择顺序的问题就显得力不从心了。因此,有必要进一步深入探讨解决更广泛的顺序类型问题的程序设计方法。

20 世纪 80 年代初,法国科技人员根据 PETRI NET 理论,提出了可编程序控制器设计的 Grafacet 法。Grafacet 法是专用于工业顺序控制程序设计的一种功能性说明语言,现在已成为法国国家标准(NFC03190)。IEC(国际电工委员会)也于 1988 年公布了类似的“控制系统功能图准备”标准(IEC848)。

现在多数 PLC 产品都有专为使用功能图编程所设计的指令,使用起来非常方便。在中小型 PLC 程序设计时,如果采用功能图法,首先要根据控制要求设计功能流程图,然后将其转化为梯形图程序。有些大型或中型 PLC 可直接用功能图进行编程。

#### 6.1.2 功能图的基本概念

功能图又称为功能流程图或状态转移图,它是一种描述顺序控制系统的图形表示方法,是专用于工业顺序控制程序设计的一种功能性说明语言。它能完整地描述控制系统的工作过程、功能和特性,是分析、设计电气控制系统控制程序的重要工具。

功能图主要由“状态”、“转移”及有向线段等元素组成。如果适当运用组成元素,就可得到控制系统的静态表示方法,再根据转移触发规则模拟系统的运行,就可以得到控制系统的动态过程。

##### 1. 状 态

状态是控制系统中一个相对不变的性质,对应于一个稳定的情形。状态的符号如图 6-1 所示。矩形框中可写上该状态的编号或代码。

(1) 初始状态 初始状态是功能图运行的起点,一个控制系统至少要有一个初始状态。初始状态的图形符号为双线的矩形框,如图 6-2 所示。在实际使用时,有时也是画单线矩形框,有时画一条横线表示功能图的开始。

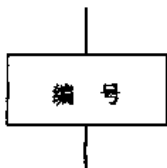


图 6-1 状态的图形符号



图 6-2 初始状态的图形符号

(2) 工作状态 工作状态是控制系统正常运行时的状态。根据系统是否运行,状态可分为动态和静态两种。动态是指当前正在运行的状态,静态是没有运行的状态。动态和静态的概念不在此深入讨论。

(3) 与状态对应的动作 在每个稳定的状态下,可能会有相应的动作。动作的表示方法如图 6-3 所示。

## 2. 转移

为了说明从一个状态到另一个状态的变化,要用转移概念,即用一个有向线段来表示转移的方向。两个状态之间的有向线段上再用一段横线表示这一转移。转移的符号如图 6-4 所示。

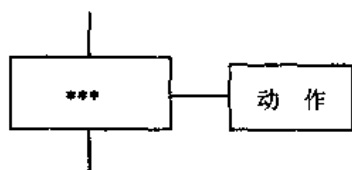


图 6-3 状态下动作的表示

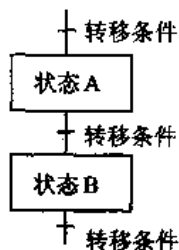


图 6-4 转移符号

转移是一种条件,当此条件成立时,称为转移使能。该转移如果能够使状态发生转移,则称为触发。一个转移能够触发必须满足:状态为动态及转移使能。转移条件是指使系统从一个状态向另一个状态转移的必要条件,通常用文字、逻辑方程及符号来表示。

## 6.1.3 功能图的构成规则

控制系统功能图的绘制必须满足以下规则:

- (1) 状态与状态不能相连,必须用转移分开;
- (2) 转移与转移不能相连,必须用状态分开;
- (3) 状态与转移、转移与状态之间的连接采用有向线段,从上向下画时,可以省略箭头;当有向线段从下向上画时,必须画上箭头,以表示方向;
- (4) 一个功能图至少要有有一个初始状态。

下面用一个例子来说明功能图的绘制。某一冲压机的初始位置是冲头抬起,处于高位;当操作者按动启动按钮时,冲头向工件冲击;到最低位置时,触动低位行程开关;然后冲头抬起,回到高位,触动高位行程开关,停止运行。图 6-5 所示为功能图表示的冲压机运行过程。冲压机的运行顺序可分为三个状态:初始、下冲和返回状态。从初始状态到下冲状态的转移须满足启动信号和高位行程开关信号同时为 ON 时才能发生;从下冲状态到返回状态,须满足低位行程开关为 ON 时才能发生。

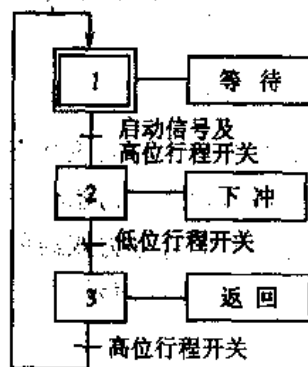


图 6-5 冲压机功能流程图

从该例可以进一步知道,功能图就是由许多个状态及连线组成的图形,它可以清晰地描述

系统的工序要求,使复杂问题简单化,并且使 PLC 编程成为可能,而且编程的质量和效率也会大大提高。


功能图有多种类型,具体举例说明见 6.3 节。

## 6.2 顺序控制指令

### 6.2.1 顺序控制指令介绍

顺序控制指令是 PLC 生产厂家为用户提供的可使功能图编程简单化和规范化的指令。S7-200 PLC 提供了三条顺序控制指令,它们的 STL 形式、LAD 形式和功能如表 6-1 所列。

表 6-1 顺序控制指令的形式及功能

STL	LAD	功 能	操作对象
LSCR bit (Load Sequential Control Relay)	bit 	顺序状态开始	S(位)
SCRT bit (Sequential Control Relay Transition)	bit —( SCRT )	顺序状态转移	S(位)
SCRE (Sequential Control Relay End)	—( SCRE )	顺序状态结束	无
CSCRE (Conditional Sequence Control Relay End)		条件顺序状态结束	无

从上表中可以看出,顺序控制指令的操作对象为顺控继电器 S, S 也称为状态器,每一个 S 位都表示功能图中的一种状态。S 的范围为: S0.0~S31.7。注意:我们使用的是 S 的位信息。

从 LSCR 指令开始到 SCRE 指令结束的所有指令组成一个顺序控制继电器(SCR)段。LSCR 指令标记一个 SCR 段的开始,当该段的状态器置位时,允许该 SCR 段工作。SCR 段必须用 SCRE 指令结束。当 SCRT 指令的输入端有效时,一方面置位下一个 SCR 段的状态器,以便使下一个 SCR 段开始工作;另一方面又同时使该段的状态器复位,使该段停止工作。由此可以总结出每一个 SCR 程序段一般有以下三种功能:

- (1) 驱动处理 即在该段状态器有效时,要做什么工作,有时也可能不做任何工作;
- (2) 指定转移条件和目标 即满足什么条件后状态转移到何处;
- (3) 转移源自动复位功能 状态发生转移后,置位下一个状态的同时,自动复位原状态。

注意:CSCRE 指令在 CPU V1.21 以上的版本中才有,而且只能进行 STL 形式编程,使用它可以结束正在执行的 SCR 段,使条件发生处和 SCRE 之间的指令不再执行。该指令不影响 S 位和堆栈。使用 CSCRE 指令后会改变正在进行的状态转移操作,所以要谨慎使用。

### 6.2.2 举例说明

在使用功能图编程时,应先画出功能图,然后对应于功能图画梯形图。图 6-6 所示为顺序控制指令使用的一个简单例子。

在该例中,初始化脉冲 SM0.1 用来置位 S0.1,即把 S0.1(状态 1)状态激活;在状态 1 的 SCR 段要做的工作是置位 Q0.4、复位 Q0.5 和 Q0.6、T37 同时计时。1s 计时到后状态发生转

移, T37 即为状态转移条件, T37 的常开触点将 S0.2(状态 2)置位(激活)的同时,自动使原状态 S0.1 复位。

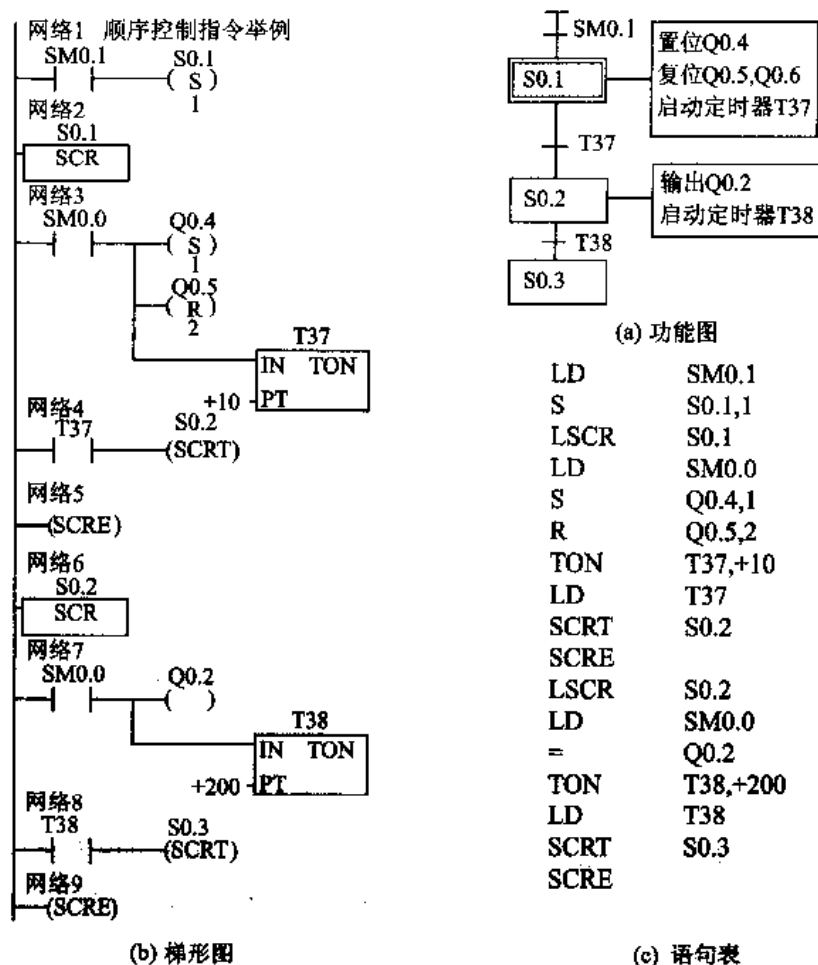


图 6-6 顺序控制指令使用举例

在状态 2 的 SCR 段,要做的工作是输出 Q0.2,同时 T38 计时,20 s 计时到后,状态从状态 2(S0.2)转移到状态 3(S0.3),同时状态 2 复位。

注意:在 SCR 段输出时,常用特殊中间继电器 SM0.0(常 ON 继电器)执行 SCR 段的输出操作。因为线圈不能直接和母线相连,所以必须借助于一个常 ON 的 SM0.0 来完成任务。

### 6.2.3 使用说明

(1) 顺控指令仅对元件 S 有效,顺控继电器 S 也具有一般继电器的功能,所以对它能够使用其他指令。

(2) SCR 段程序能否执行取决于该状态器(S)是否被置位,SCRE 与下一个 LSCR 之间的指令逻辑不影响下一个 SCR 段程序的执行。

(3) 不能把同一个 S 位用于不同程序中,例如:如果在主程序中用了 S0.1,则在子程序中就不能再使用它。

(4) 在 SCR 段中不能使用 JMP 和 LBL 指令,就是说不允许跳入、跳出或在内部跳转,但可以在 SCR 段附近使用跳转和标号指令。

(5) 在 SCR 段中不能使用 FOR、NEXT 和 END 指令。

(6) 在状态发生转移后,所有的 SCR 段的元器件一般也要复位,如果希望继续输出,可使用置位/复位指令,如图 6-6 中的 Q0.4。

(7) 在使用功能图时,状态器的编号可以不按顺序编排。

### 6.3 功能图的主要类型

#### 6.3.1 单流程

这是最简单的功能图,其动作是一个接一个地完成。每个状态仅连接一个转移,每个转移也仅连接一个状态。如图 6-7 所示为单流程的功能图、梯形图和语句表。

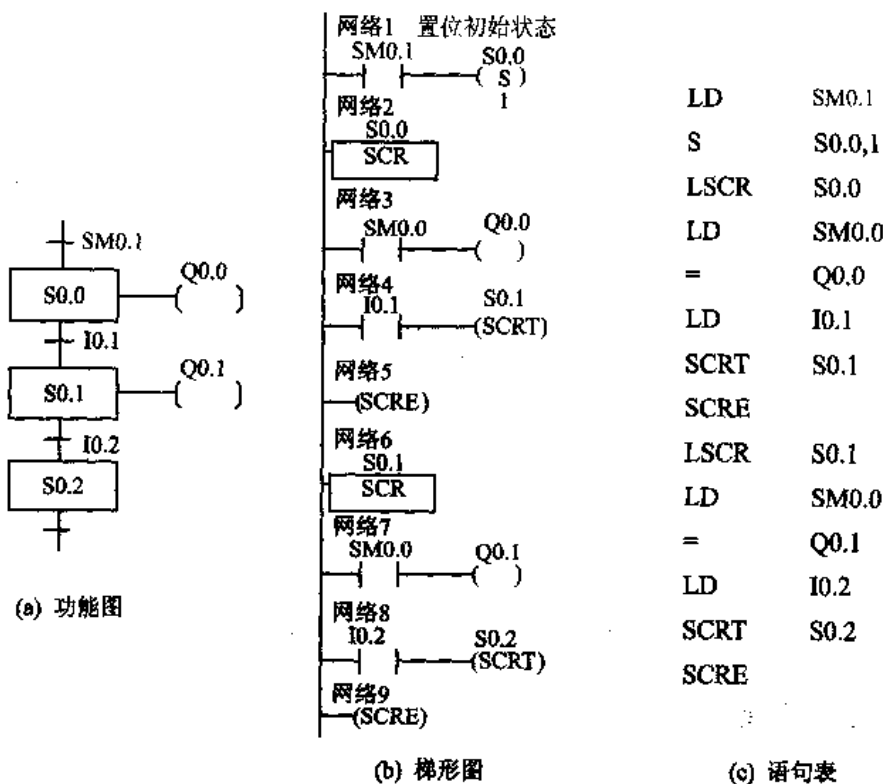


图 6-7 单流程功能图举例

#### 6.3.2 可选择的分支和联接

在生产实际中,对具有多流程的工作要进行流程选择或者分支选择。即一个控制流可能转入多个可能的控制流中的某一个,但不允许多路分支同时执行。到底进入哪一个分支,取决于控制流前面的转移条件哪一个为真。可选择分支和联接的功能图、梯形图如图 6-8 所示。

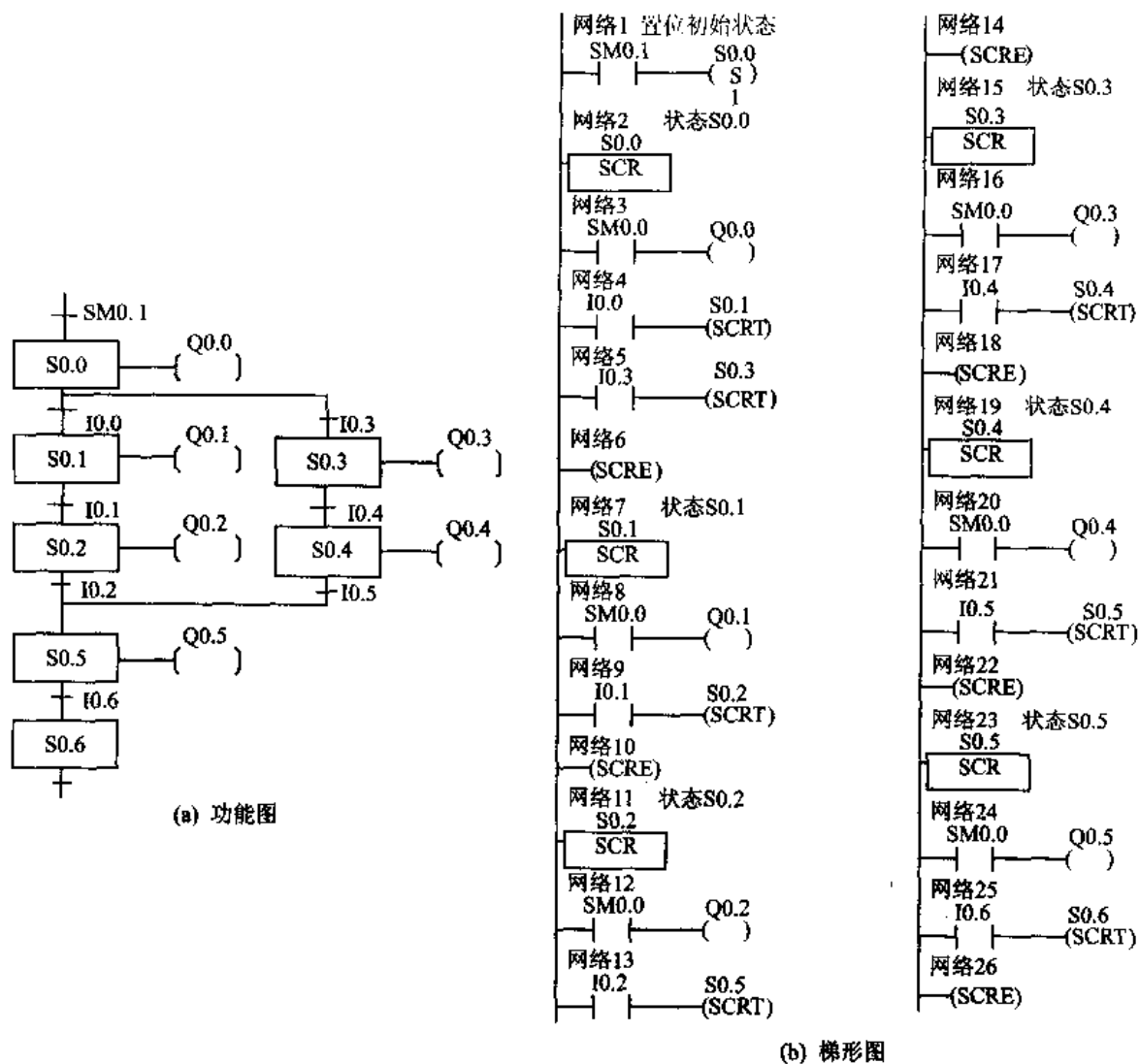


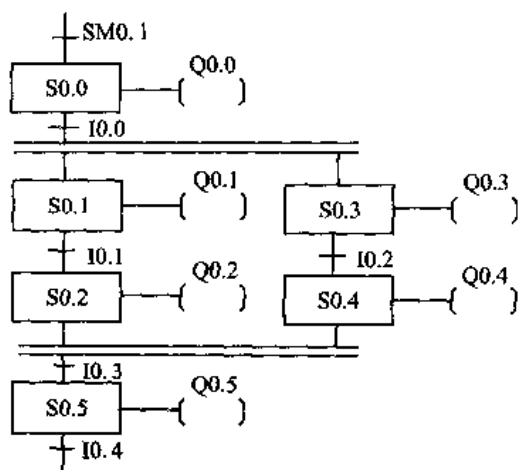
图 6-8 可选择的分支和联接功能图举例

### 6.3.3 并行分支和联接

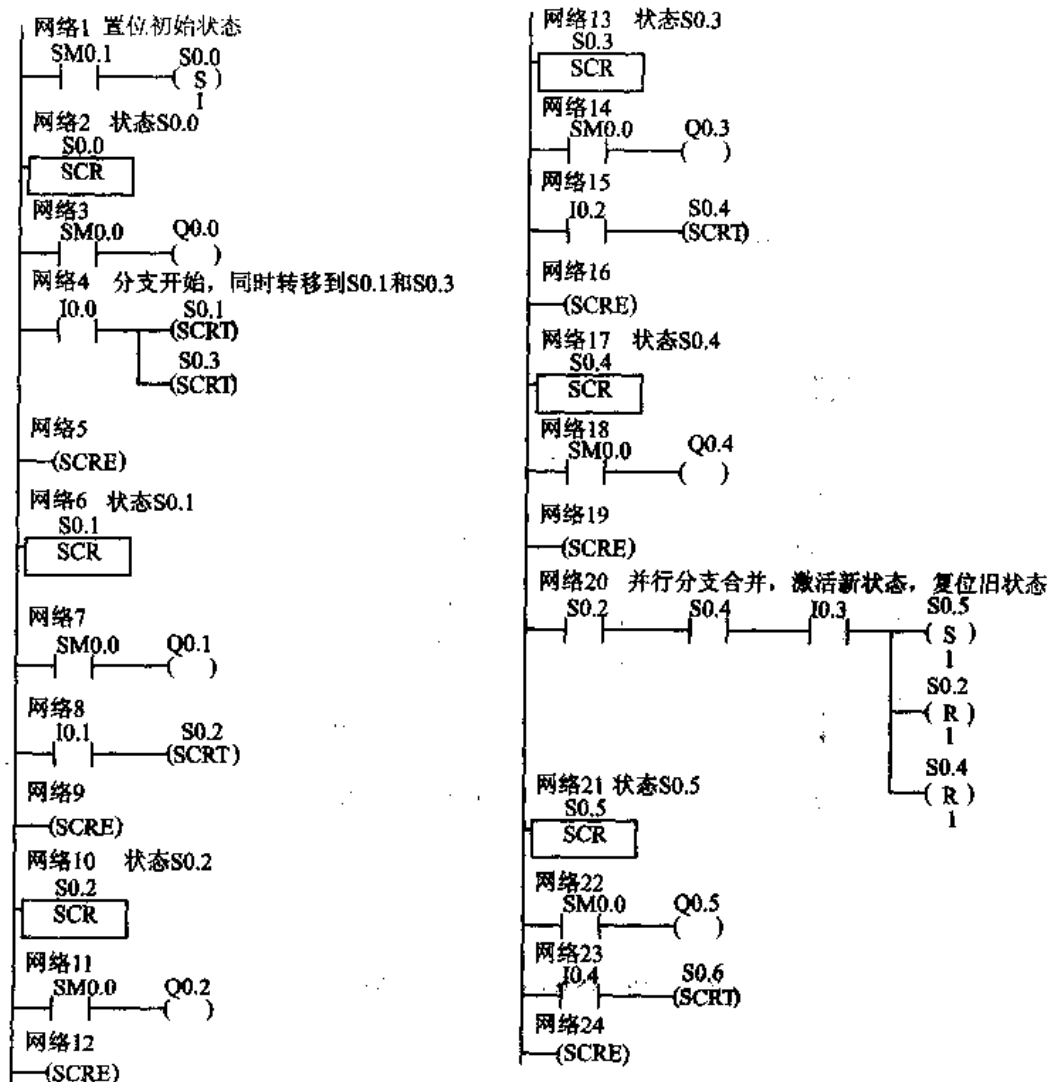
在许多实例中,一个顺序控制状态流必须分成两个或多个不同分支控制状态流,这就是并行分支或并发分支。当一个控制状态流分成多个分支时,所有的分支控制状态流必须同时激活。当多个控制流产生的结果相同时,可以把这些控制流合并成一个控制流,即并行分支的联接。在合并控制流时,所有的分支控制流必须都是完成了的。这样,在转移条件满足时才能转移到下一个状态。并发顺序一般用双水平线表示,同时结束若干个顺序也用双水平线表示。

图 6-9 所示为并行分支和联接的功能图和梯形图。需要特别说明的是,并行分支联接时要同时使状态转移到新的状态,完成新状态的启动。另外在状态 S0.2 和 S0.4 的 SCR 程序段中,由于没有使用 SCRT 指令,所以 S0.2 和 S0.4 的复位不能自动进行,最后要用复位指令对其进行复位。这种处理方法在并行分支的联接合并时会经常用到,而且在并行分支联接合并前的最后一个状态往往是“等待”过渡状态。它们要等待所有并行分支都为“真”后一起转移到新的状态。这时的转移条件永远为“真”,而这些“等待”状态不能自动复位,它们的复位就要使

用复位指令来完成了。具体应用参见 6.4.2 节。



(a) 功能图



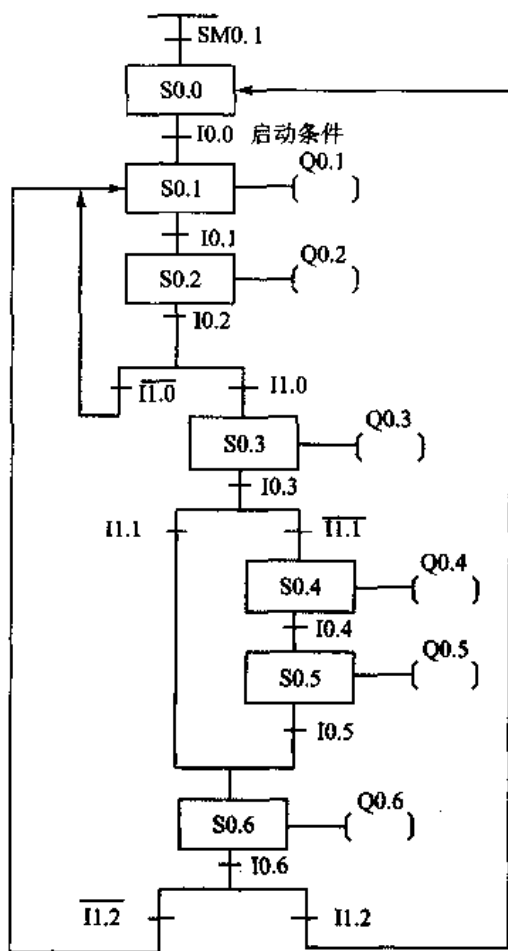
(b) 梯形图

图 6-9 并行分支和联接功能图举例

### 6.3.4 跳转和循环

单一顺序、并发和选择是功能图的基本形式。多数情况下,这些基本形式是混合出现的,跳转和循环是其典型代表。

利用功能图语言可以很容易实现流程的循环重复操作。在程序设计过程中可以根据状态的转移条件,决定流程是单周期操作还是多周期循环,是跳转还是顺序向下执行。图 6-10 为跳转和循环的功能图、梯形图和语句表。

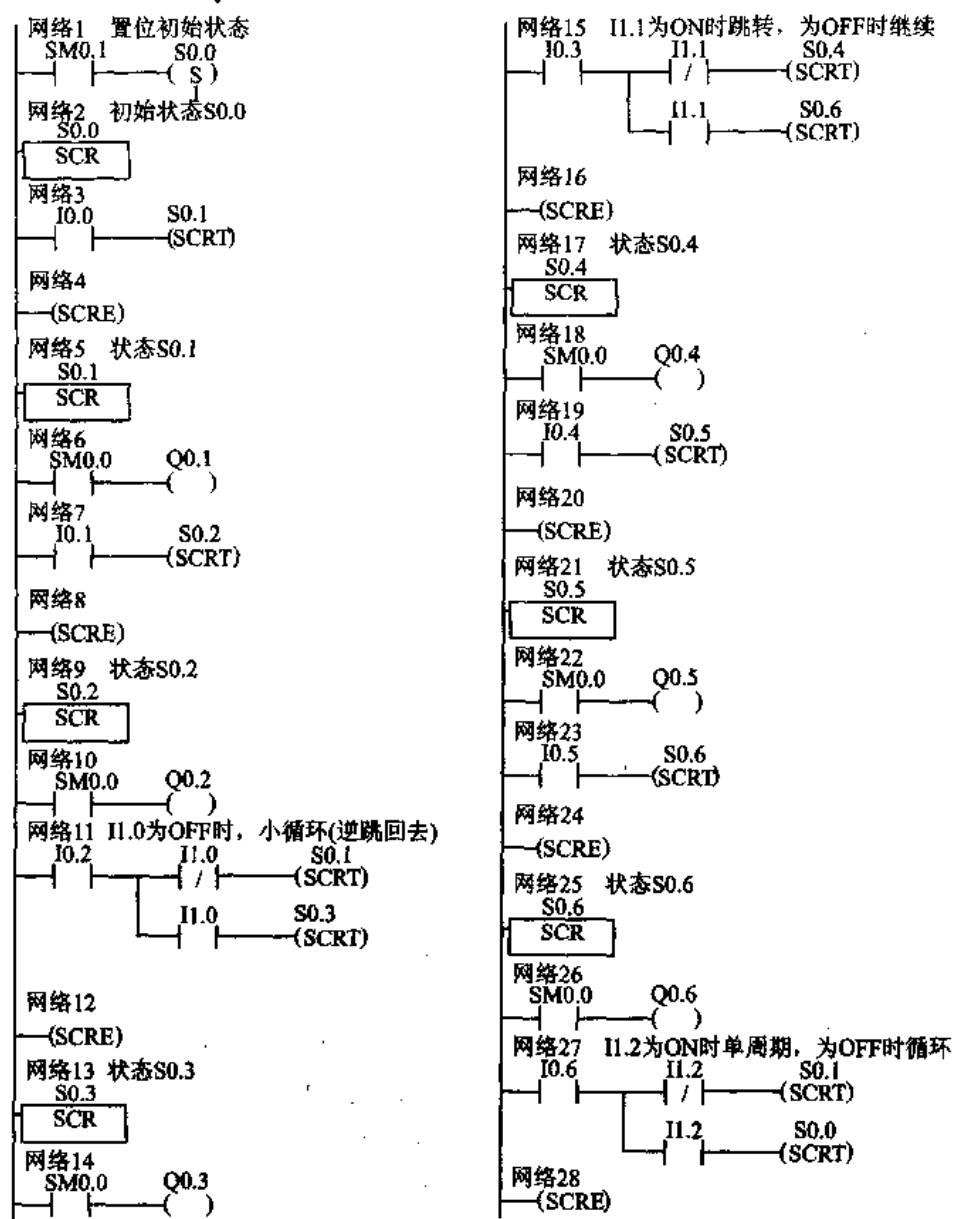


(a) 功能图

LD	SM0.1	SCRT	S0.4
S	S0.0,1	LPP	
LSCR	S0.0	A	I1.1
LD	I0.0	SCRT	S0.6
SCRT	S0.1	SCRE	
SCRE		LSCR	S0.4
LSCR	S0.1	LD	SM0.0
LD	SM0.0	=	Q0.4
=	Q0.1	LD	I0.4
LD	I0.1	SCRT	S0.5
SCRT	S0.2	SCRE	
SCRE		LSCR	S0.5
LSCR	S0.2	LD	SM0.0
LD	SM0.0	=	Q0.5
=	Q0.2	LD	I0.5
LD	I0.2	SCRT	S0.6
LPS		SCRE	
AN	I1.0	LSCR	S0.6
SCRT	S0.1	LD	SM0.0
LPP		=	Q0.6
A	I1.0	LD	I0.6
SCRT	S0.3	LPS	
SCRE		AN	I1.2
LSCR	S0.3	SCRT	S0.1
LD	SM0.0	LPP	
=	Q0.3	A	I1.2
LD	I0.3	SCRT	S0.0
LPS		SCRE	
AN	I1.1		

(b) 语句表

图 6-10 跳转和循环功能图举例



(c) 梯形图

图 6-10 跳转和循环功能图举例(续)

图中:I1.0 为 OFF 时进行局部循环操作,I1.0 为 ON 时则正常顺序执行;I1.1 为 ON 时正向跳转,I1.1 为 OFF 时则正常顺序执行;I1.2 为 OFF 时进行多周期循环操作,I1.2 为 ON 时则进行单周期循环操作。

## 6.4 顺序控制指令应用举例

### 6.4.1 选择和循环电路举例

#### 1. 题目

图 6-11 为一台分检大小球的机械臂装置。它的工作过程是:当机械臂处于原始位置时,

即上限开关 LS1 和左限位开关 LS3 压下,抓球电磁铁处于失电状态。这时按动启动按钮 SB1 后,机械臂下行,碰到下限位开关 LS2 后停止下行,且电磁铁得电吸球。如果吸住的是小球,则大小球检测开关 SQ 为 ON;如果吸住的是大球,则 SQ 为 OFF。1 秒钟后,机械臂上行,碰到上限位开关 LS1 后右行,它会根据大小球的不同,分别在 LS4(小球)和 LS5(大球)处停止右行,然后下行至下限位停止,电磁铁失电,机械臂把球放在小球箱里或大球箱里,1 秒钟后返回。如果不按停止按钮 SB2,则机械臂一直工作下去。如果按了停止按钮,则不管何时按,机械臂最终都要停止在原始位置。再次按动启动按钮后,系统可以再次从头开始循环工作。

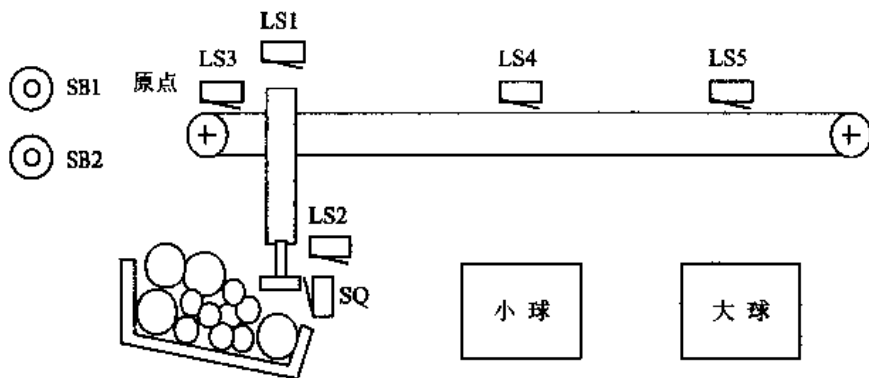


图 6-11 机械臂分检装置示意

## 2. 解 题

### (1) 输入输出点地址分配

输入点:

启动按钮 SB1	I0.0
停止按钮 SB2	I0.1
上限位开关 LS1	I0.2
下限位开关 LS2	I0.3
左限位开关 LS3	I0.4
小球右限位开关 LS4	I0.5
大球右限位开关 LS5	I0.6
大小球检测开关 SQ	I0.7

输出点:

原始位置指示灯 HL	Q0.0
抓球电磁铁 K	Q0.1
下行接触器 KM1	Q0.2
上行接触器 KM2	Q0.3
右行接触器 KM3	Q0.4
左行接触器 KM4	Q0.5

(2) 系统功能如图 6-12 所示,梯形图如图 6-13 所示。

### 3. 简要说明

(1) 由于大小球的不同,所以使用了分支选择电路,使机械臂能够在右行后在不同的位置下行,把大小球分别放进各自的箱子里去。

(2) 在机械手上、下、左、右行走的控制中,加上了一个软件联锁触点,替代了 SM0.0。

(3) 图 6-12 中的 M0.0 是一个选择逻辑,其功能如图 6-13 中的网络 1 所示,它相当于一个开关,控制着系统是进行单周期操作还是循环操作。

(4) S7-200 PLC 的顺控指令不支持直接输出(=)的双线圈操作。如果在图 6-12 中的状态 S0.1 的 SCR 段有 Q0.2(下行)输出,在状态 S1.0 的 SCR 段也有 Q0.2 输出,则不管在什么情况下,在前面的 Q0.2 永远不会有效。这是 S7-200 PLC 顺控指令设计方面的缺陷,为用户的使用带来了极大的不便。所以在使用 S7-200 PLC 的顺控指令时一定要不要有双线圈输

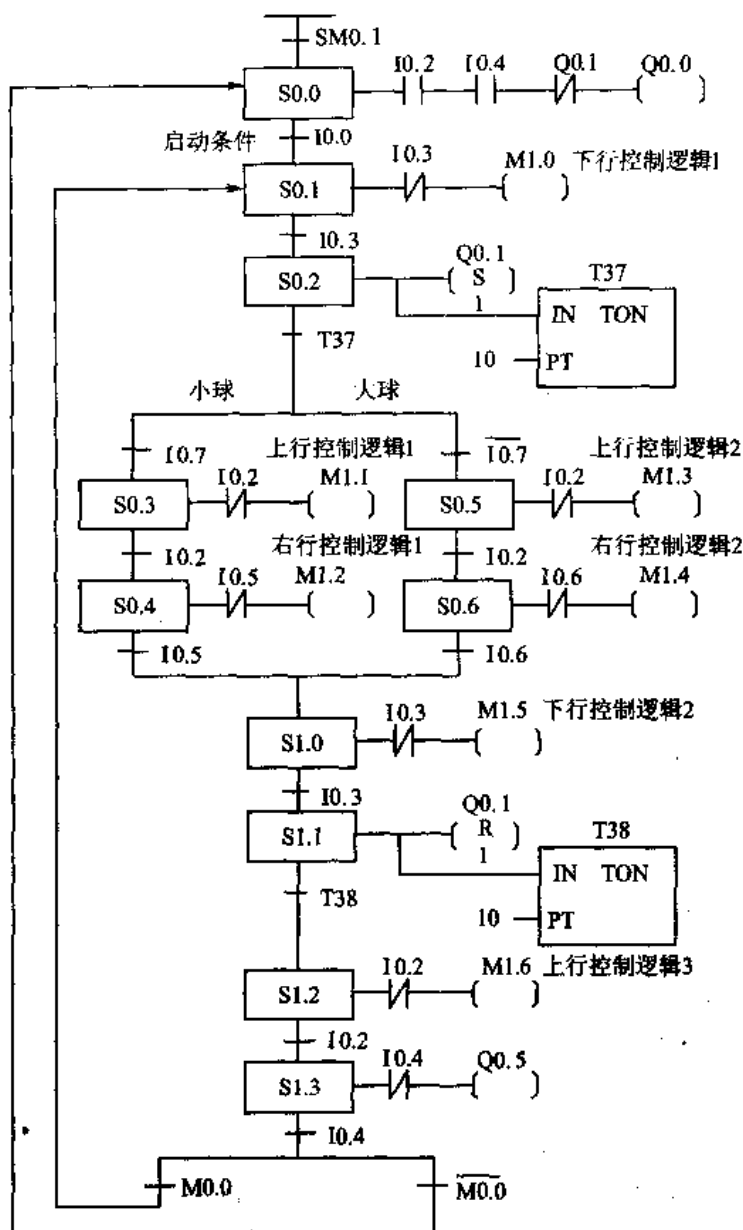


图 6-12 机械臂分检装置功能图

出。为解决这个问题,可采用本例的办法,用中间继电器逻辑过渡一下,如本例中的机械手进行上行、下行和右行的控制逻辑设计,凡是有重复使用的相同输出驱动,在 SCR 段中先用中间继电器表示其分段的输出逻辑,在程序的最后再进行合并输出处理。这是解决这一缺陷的最佳方法。左行时只有在状态 S1.7 中用到了 Q0.5,所以就不用中间过渡处理了。

注意:经作者验证,在 CPU V1.21 版本和 Micro/win32 V3.2 版本中该系统软件缺陷仍存在。

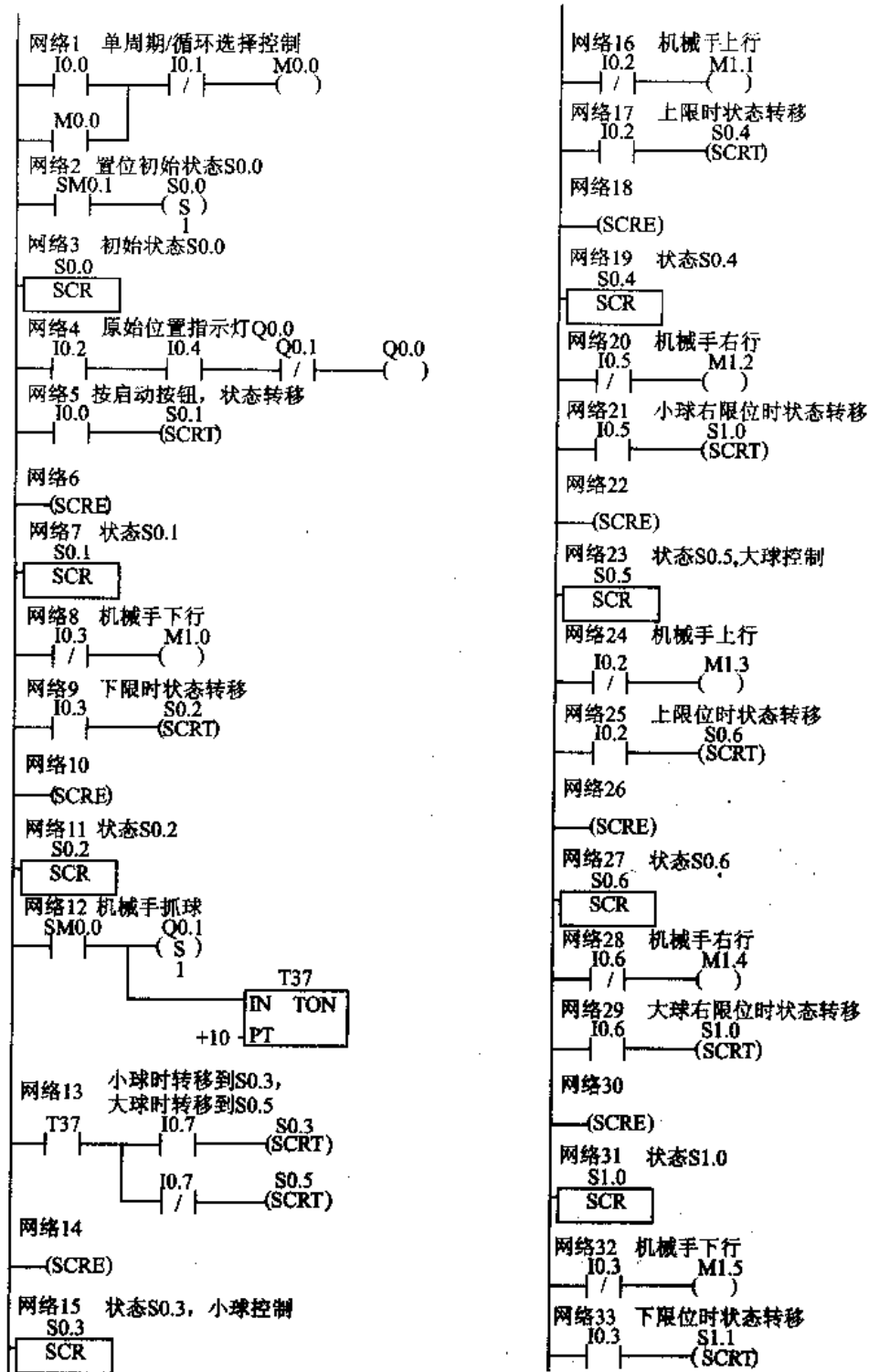


图 6-13 机械臂装置梯形图

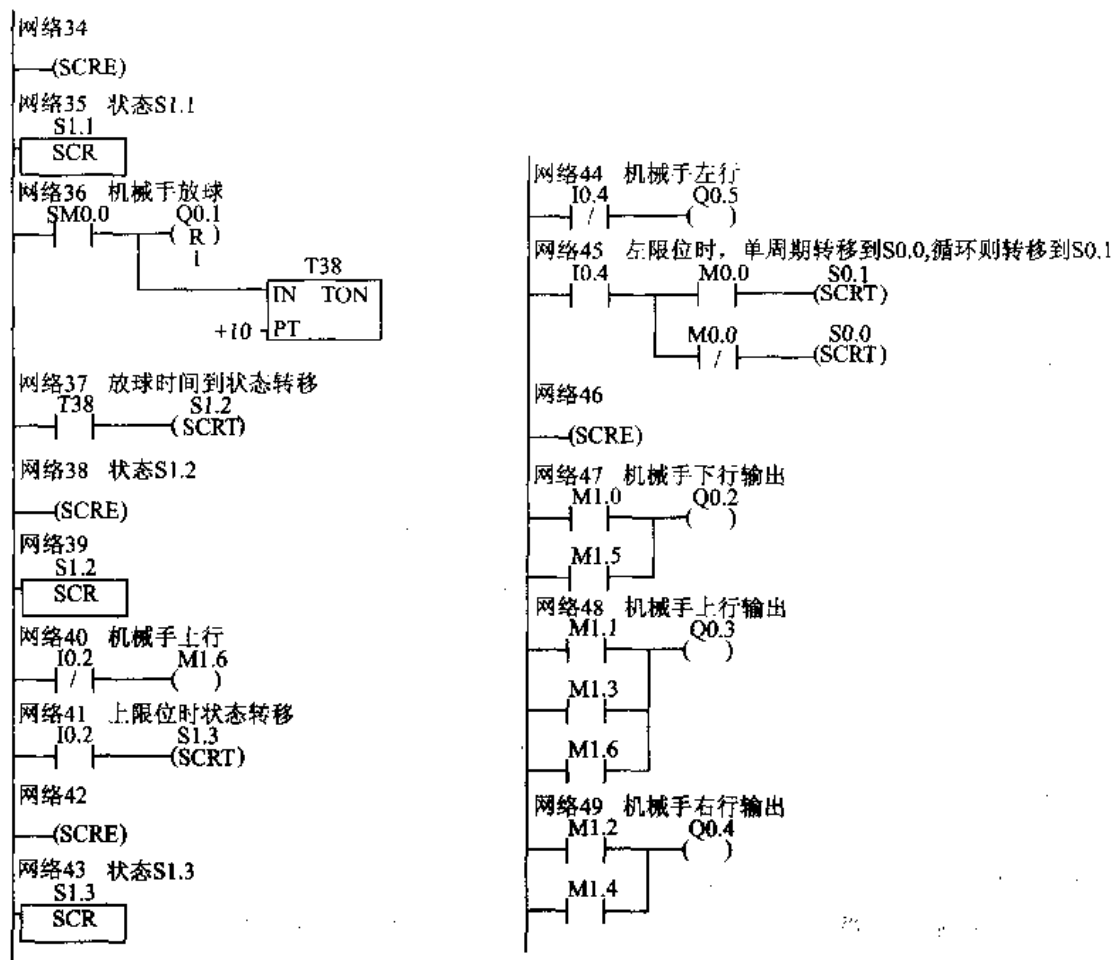


图 6-13 机械臂装置梯形图(续)

### 6.4.2 并行分支和联接电路举例

#### 1. 题目

某化学反应过程的装置由四个容器组成,容器之间用泵连接,以此来进行化学反应。每个容器都装有检测容器空满的传感器,2#容器还带有加热器和温度传感器,3#容器带有搅拌器。当1#、2#容器中的液体抽入3#容器时,启动搅拌器。3#、4#容器是1#、2#容器体积的两倍,可以由1#、2#容器的液体装满。化学反应过程如图6-14所示。

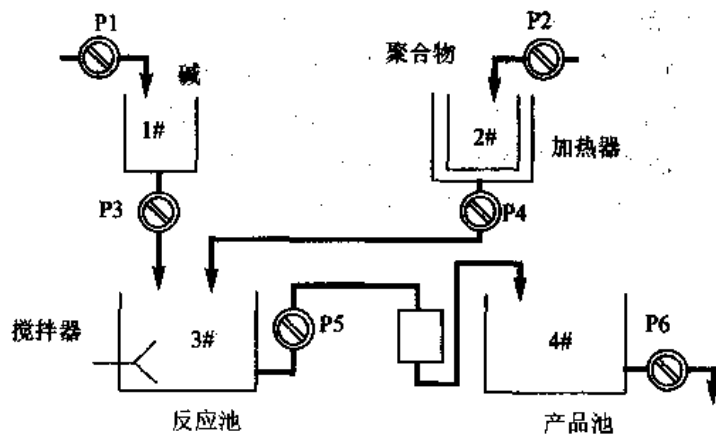


图 6-14 化学反应过程示意图

该化学反应过程的工作原理是:按动启动按钮后,1#、2#容器分别用泵 P1、P2 从碱和聚合物库中将其抽满。抽满后传感器发出信号,P1、P2 关闭,然后 2#容器加热到 60℃时,温度传感器发出信号,关掉加热器。P3、P4 分别将 1#、2#容器中的溶液送到 3#反应器中,同时启动搅拌器,搅拌时间为 60 s。一旦 3#满或 1#、2#空,则泵 P3、P4 停止并等待。当搅拌时间到,P5 将混合液抽到产品池 4#容器,直到 4#满或 3#空。成品用 P6 抽走,直到 4#空。至此,整个过程结束,再次按动启动按钮,新的循环可以开始。

## 2. 解 题

### (1) 输入/输出点地址分配

输入点:

手动启动按钮	I0.0
1#容器满	I0.1
1#容器空	I0.2
2#容器满	I0.3
2#容器空	I0.4
3#容器满	I0.5
3#容器空	I0.6
4#容器满	I0.7
4#容器空	I1.0
温度传感器	I1.1

输出点:

泵 P1 接触器	Q0.0
泵 P2 接触器	Q0.1
泵 P3 接触器	Q0.2
泵 P4 接触器	Q0.3
泵 P5 接触器	Q0.4
泵 P6 接触器	Q0.5
加热器接触器	Q0.6
搅拌器接触器	Q0.7

(2) 根据系统控制要求绘制的功能如图 6-15 所示。由功能图设计出的梯形图程序如图 6-16 所示。

### 3. 简要说明

(1) 初始状态设为 P1、P2、P3、P4、P5、P6 停,加热器停和搅拌器停,并且 4#容器空。在使用编程软件画梯形图时受宽度以及教材排版的限制,所以用 M0.0 和 M0.1 做了一下过渡。

(2) 该例中的关键是进行并行分支的合并处理,在一些并行分支合并时,由于各分支不一定同时结束,所以设计一些等待状态是必须的,也是合理的。对这些等待状态的复位处理要使用复位指令。

(3) 并行分支合并后转移到新的状态可以有转移条件,但有时看不到转移条件,其实这时的转移条件就是永远为“真”。即只要所有合并的分支最后一个状态都为 ON 时就可以转移。永远为“真”的条件在功能图上可以写出来,也可以不写出来。在该例的功能图中,我们都举了例子。分支状态 S2.0、S2.1 往状态 S0.3 转移时,就标出了转移条件“=1”,即为“真”的条件。而在 S2.2、S2.3、S2.4 往状态 S0.7 转移时就没有标出转移条件。

(4) 并行分支合并前的状态编号最好是连续的,如本例中的 S2.0、S2.1 和 S2.2、S2.3、S2.4,这样在最后对它们进行复位时只用一条复位指令就行了,这是一个小的使用技巧。

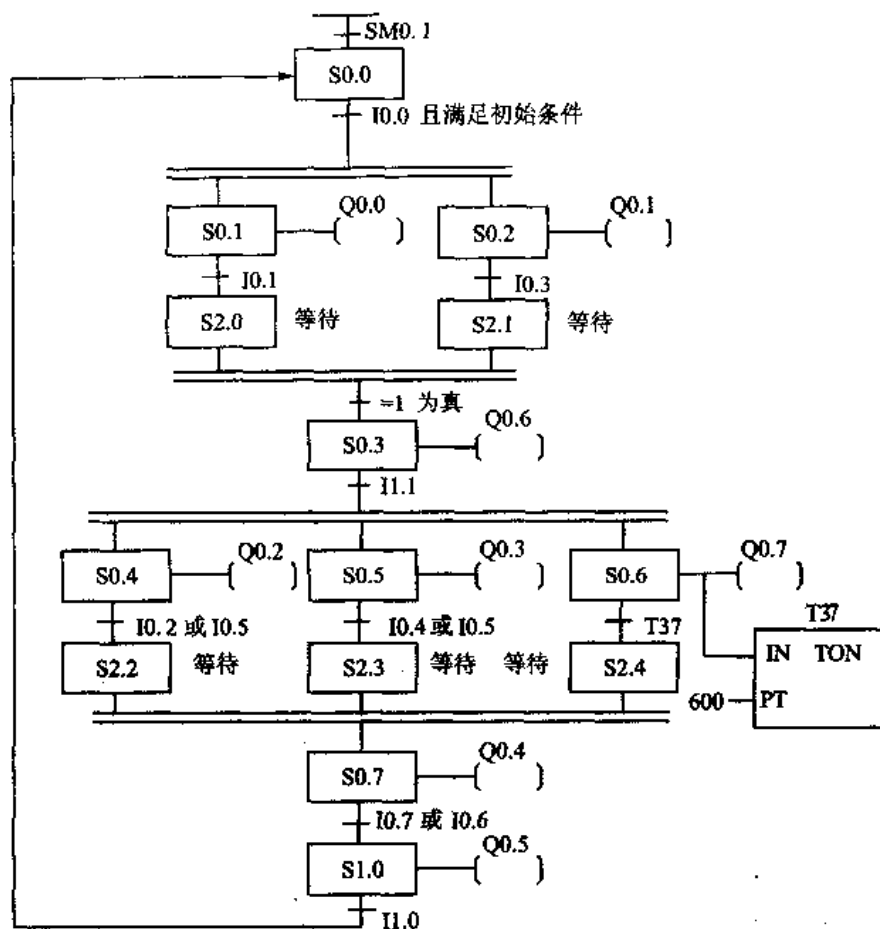


图 6-15 化学反应过程功能图

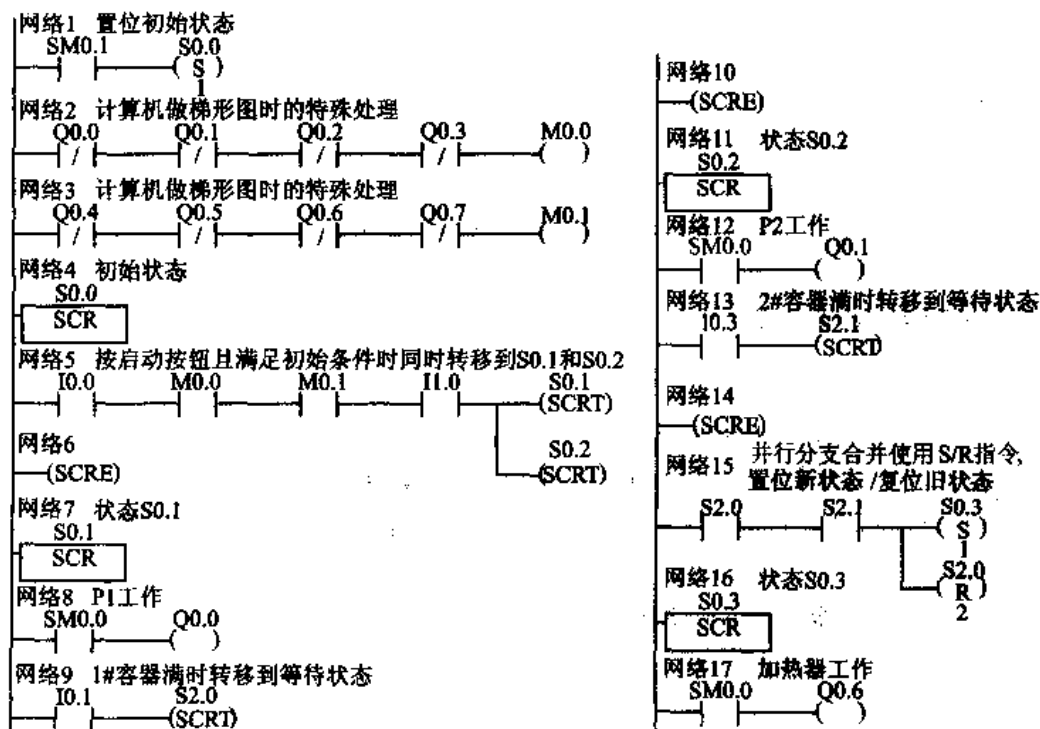


图 6-16 化学反应过程 PLC 控制系统梯形图程序

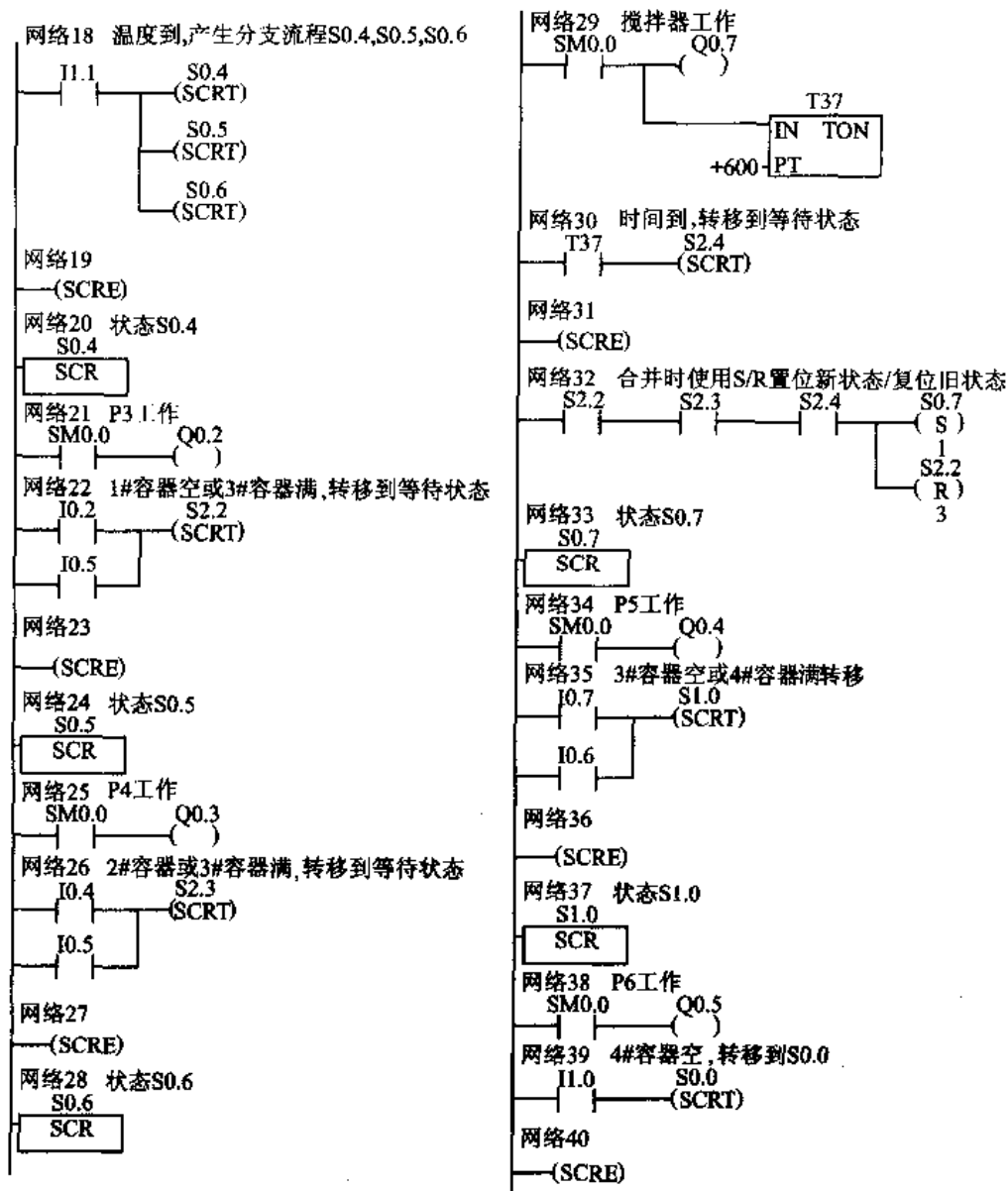


图 6-16 化学反应过程 PLC 控制系统梯形图程序(续)

### 6.4.3 选择和跳转电路举例

#### 1. 题目

如图 6-17 所示,三台电机在按下启动按钮后,每隔一段时间自动顺序启动;启动完毕后,按下停止按钮,每隔一段时间自动反向顺序停止。在启动过程中,如果按下停止按钮,则立即中止启动过程,对已启动运行的电机,马上进行反方向顺序停止,直到全部结束。

#### 2. 解题

该例控制系统的功能如图 6-18 所示,根据功能图设计的梯形图如图 6-19 所示。

PLC 的输入/输出地址分配如下:

启动按钮: I0.0;

停止按钮: I0.1;

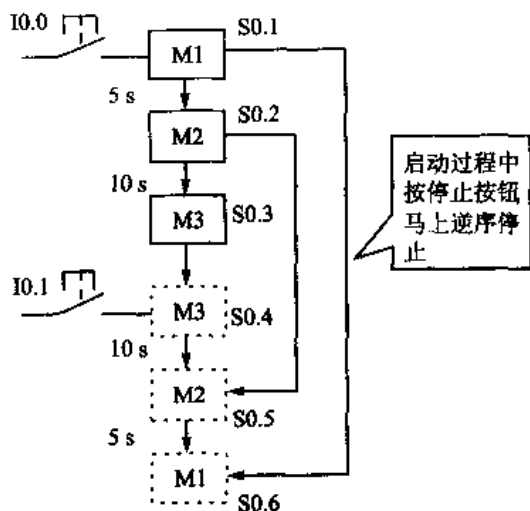


图 6-17 电机顺序启动/停止控制示意图

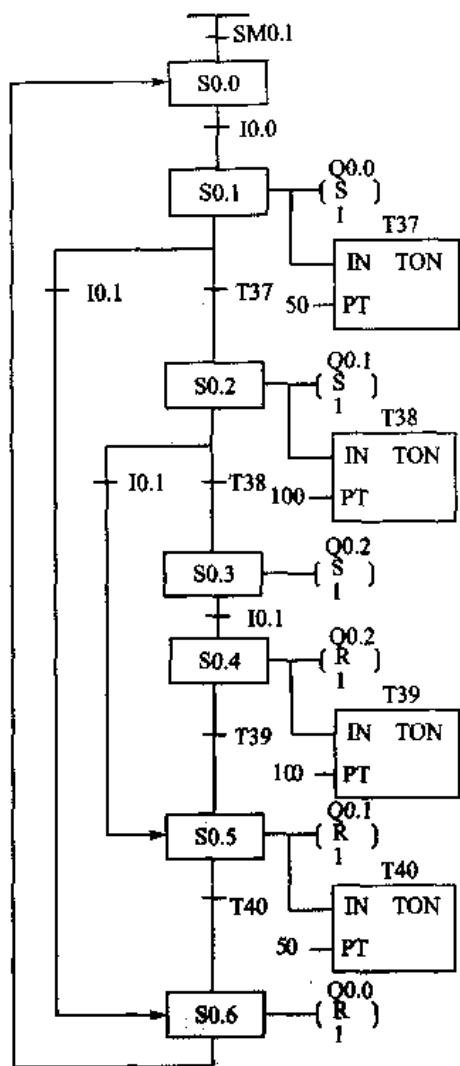


图 6-18 电机顺序启动/停止功能图

电机 M1: Q0.0;

电机 M2: Q0.1;

电机 M3: Q0.2。

### 3. 简要说明

(1) 在图 6-19 中加上了对所使用的顺序控制继电器 S 进行初始化的复位处理, 在 S7-200 PLC 中, S 不是掉电保持型的存储器, 所以不对它进行初始化复位也是可以的。

(2) 在启动过程中如果按下停止按钮, 则马上转移到相应的状态, 原状态随之复位, 定时器 T37 或 T38 也会复位。

(3) 在图 6-18 的最后一个状态 S0.6 后, 要激活初始状态 S0.0, 不然无法再次开始下一轮工作; 按本例设计, 则再次按下启动按钮后, 系统又可继续工作。

(4) 本例中最关键的是要设计好选择分支的条件和跳转的目标状态, 处理好结束状态的转移目标。

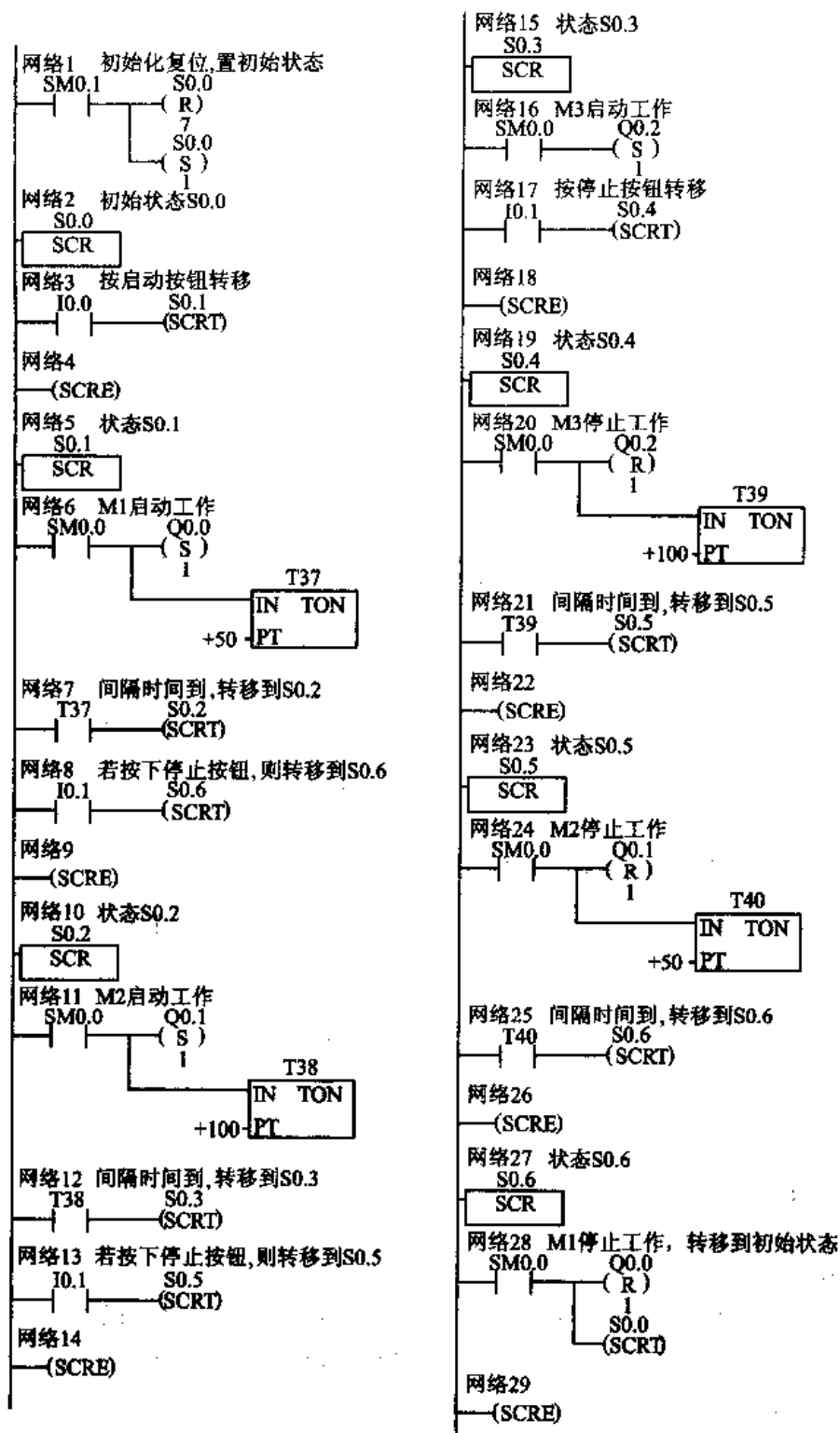


图 6-19 电机顺序启动/停止梯形图

## 本章小结

(1) 在 6.1 节中,主要讲述了为什么要使用功能图以及功能图主要解决什么问题。最后简要介绍了功能图的几个概念。

(2) 在 6.2 节中,主要介绍了 S7-200 PLC 所提供的顺序控制指令,这些指令是配合功能图来使用的。大家要理解 SCR 程序段的功能。

(3) 6.3 节给出了功能图的几种类型及使用方法。

(4) 6.4 节给出了三个应用实例。在这些应用实例中,希望大家学会其中的一些使用技巧,并注意理解简要说明中的内容。

(5) S7-200 PLC 的顺控指令的设计是有缺陷的,即它不支持双线圈输出,这为在不同的 SCR 段使用同一个线圈输出带来了不便,本书指出了该缺陷,并给出了最简单的解决方法。

## 思考题与练习题

1. 什么是功能图? 功能图主要由哪些元素组成?
2. 顺序控制指令段有哪些功能?
3. 功能图的主要类型有哪些?
4. 写出图 6-13、图 6-16 和图 6-19 的语句表程序。
5. 本书利用电气原理图、PLC 一般指令和功能图三种方法设计了“三台电动机顺序启动/停止”的例子,试比较它们的设计原理、方法和结果的异同。
6. 用功能图方法完成第五章 5.5.2 节应用举例中[例 5-2]的编程。要求画出功能图、梯形图,并写出语句表。
7. 图 6-20(a)为人行道和马路的信号灯系统。当行人过马路时,可按下分别安装在马路两侧的按钮 I0.0 或 I0.1,则交通灯(红灯,黄灯,绿灯三种类型)系统按图 6-20(b)中形式工作。在工作期间,任何按钮按下都不起作用。  
试设计该控制系统的功能图,并画出梯形图,写出语句表。

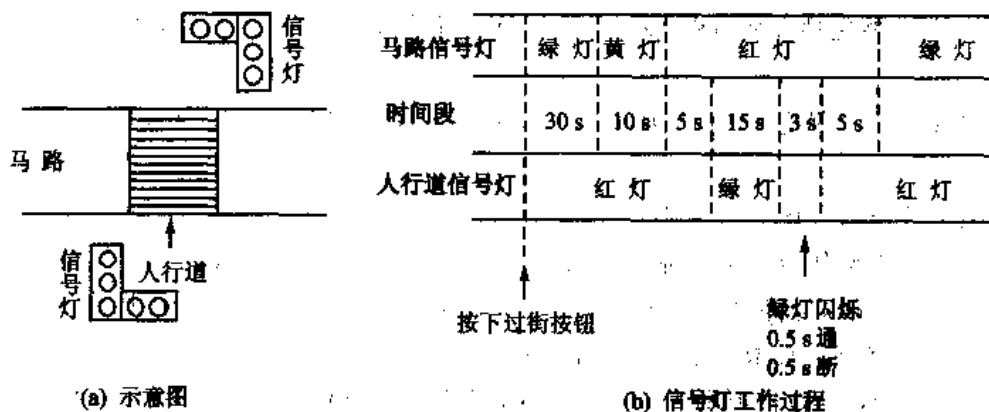


图 6-20 交通灯控制示意图

8. 用功能图方法完成第五章中习题 13 的程序设计。要求画出功能图、梯形图,写出语句表。

## 第七章 S7-200 PLC 的功能指令

随着计算机技术的发展,PLC除了有丰富的逻辑指令外,还有丰富的功能指令。实际上,现在的PLC就是一个计算机控制系统。为了满足工业控制的需要,PLC生产厂家为PLC增添了过程控制、数据处理和特殊功能的指令,这些指令我们称为功能指令(Function Instruction)。这些功能指令的出现,极大地拓宽了PLC的应用范围,增强了PLC编程的灵活性。

S7-200系列可编程序控制器的功能指令主要包括以下类型:

传送、移位及填充指令;

算术运算与逻辑运算指令;

数据转换指令;

高速处理指令;

通信指令;

PID指令。

在本章的功能指令介绍中,有以下几个方面的约定:

① 指令格式 给出了指令的梯形图和语句表格式。在所有的说明图中,上面的指令盒为LAD格式,下面为指令的STL格式。

② 功能描述 详细描述了指令的功能,讲解了使用中的注意事项。

③ 字符含义 B表示字节,W表示字,I表示整数,DW表示双字(LAD中),DI表示双整数(LAD中),D表示双字或双整数(STL中),R表示实数。

④ 数据类型 读者要特别注意指令的操作数形式。对操作数的内容,本书有如下约定:

字节型包括VB、IB、QB、MB、SB、SMB、LB、AC、\*VD、\*LD、\*AC和常数;

字型及INT型包括VW、IW、QW、MW、SW、SMW、LW、AC、T、C、\*VD、\*LD、\*AC和常数;

双字型及DINT型包括VD、ID、QD、MD、SD、SMD、LD、AC、\*VD、\*LD、\*AC和常数;

字符型字节包括VB、LB、\*VD、\*LD和\*AC。

操作数分输入操作数(IN)和输出操作数(OUT)。以上对操作数的概括只是一般总结,具体使用到每条指令时,可能会有微小的不同;另外,输入操作数(IN)和输出操作数(OUT)的相同数据类型的内容也会有微小不同,例如输出操作数(OUT)一般不包括常数。

⑤ EN与ENO 在梯形图中,S7-200 PLC用一个方框表示每一条功能指令,这些方框称为指令盒。我们假想梯形图的母线能提供一种能流,能流在梯形图中流动。每个指令盒都有一个使能输入端EN(Enable In)和一个使能输出端ENO(Enable Out)。当EN端有能流,即EN端有效时,该条功能指令才被执行;如果EN端有能流且该功能指令执行无误时,则ENO为1,即ENO能把这种能流传递下去,如果指令执行有误,则ENO为0,能流不能继续

传递。大家记住所有的功能指令只有在 EN 端有效时才被执行。

⑥ 标志位 由一些特殊继电器组成,如 SM1(见附录 B-4)。它们用来记录在执行功能指令时所产生的特殊信息。由于在教学时,PLC 的编程中较少使用标志位,因此除个别情况外,书中没有对功能指令影响标志位的情况进行说明。在实际使用时,读者可以查阅 S7-200 系统手册。

⑦ 使能信号 有些功能指令需要的是使能信号的上升沿,若使能信号不是一个扫描周期的脉冲信号,则可能会产生意想不到的结果。所以在使用功能指令时,大家要注意对输入使能信号的处理,这一点非常重要。

注意:本章有关字符串的指令在 CPU V1.21 和 Micro/win32 V3.2 版本以上才可使用。

## 7.1 传送、移位和填充指令

此类指令涉及对数据的非数值运算操作,主要包括传送、移位、字节交换、循环移位和填充等指令。

### 7.1.1 传送类指令

该类指令用来完成各存储单元之间进行一个或者多个数据的传送。可分为单一传送指令和块传送指令。

#### 1. 单一传送(Move)

单一传送包括字节传送、字传送和双字传送。

指令格式:LAD 和 STL 格式如图 7-1(a)所示。图中的□处可为 B、W、DW(LAD 中)、D(STL 中)或 R。

功能描述:使能输入有效时,把一个单字节数据(字、双字或实数)由 IN 传送到 OUT 所指的存储单元。

数据类型:输入输出均为字节(字、双字或实数)。

#### 2. 块传送(Block Move)

该类指令可用来进行一次多个(最多 255 个)数据的传送,它包括字节块传送、字块传送和双字块传送。

指令格式:LAD 及 STL 格式如图 7-1(b)所示。图中的□处可为 B、W、DW(LAD 中)、D(STL 中)或 R。

功能描述:把从 IN 开始的 N 个字节(字或双字)型数据传送到从 OUT 开始的 N 个字节(字或双字)存储单元。

数据类型:输入输出均为字节(字或双字),N 为字节。

#### 3. 字节立即传送(Move Byte Immediate)

字节立即传送指令就像位指令中的立即指令一样,用于输入和输出的立即处理。

##### (1) 传送字节立即读指令

指令格式:LAD 及 STL 格式如图 7-1(c)所示。

功能描述:立即读取单字节物理区数据 IN,并传送到 OUT 所指的字节存储单元。该指令用于对输入信号的立即响应。

操作数:输入为 IB,输出为字节。

## (2) 传送字节立即写指令

指令格式:LAD 及 STL 格式如图 7-1(d)所示。

功能描述:立即将 IN 单元的字节数据写到 OUT 所指的字节存储单元的物理区及映像区,它用于把计算出的 Q 结果立即输出到负载。

数据类型:输入为字节,输出为 QB。

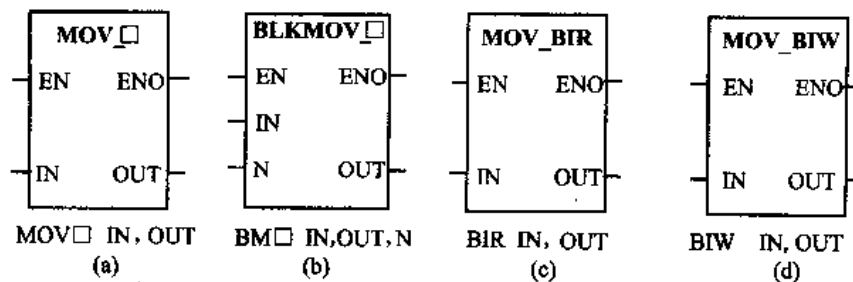


图 7-1 传送指令格式

### 【例 7-1】 传送类指令应用举例。

LD	I0.0	//I0.0 有效时执行下面操作
MOVB	VB100, VB200	//字节 VB100 中的数据送到字节 VB200 中
MOVW	VW110, VW210	//字 VW110 中的数据送到字 VW210 中
MOVD	VD120, VD220	//双字 VD120 中的数据送到双字 VD220 中
BMB	VB130, VB230, 4	//字节 VB130 开始的 4 个连续字节中的数据送到 //VB230 开始的 4 个连续字节存储单元中
BMW	VW140, VW240, 4	//字 VW140 开始的 4 个连续字中的数据送到字 //VW240 开始的 4 个连续字存储单元中
BMD	VD150, VD250, 4	//双字 VD150 开始的 4 个连续双字中的数据送到双字 //VD250 开始的 4 个连续双字存储单元中
BIR	IB1, VB270	//I1.0 到 I1.7 的物理输入状态立即送到 VB270 中,不受扫描 //周期的影响
BIW	VB280, QB0	//VB280 中的数据立即从 Q0.0 到 Q0.7 端子输出,不受扫描 //周期的影响

## 7.1.2 移位与循环指令

该类指令包括左移和右移、左循环和右循环。在该类指令中,LAD 与 STL 指令格式中的缩写表示是不同的。

### 1. 移位指令(Shift)

该指令有左移和右移两种。根据所移位数的长度不同可分为字节型、字型 and 双字型。移位数据存储单元的移出端与 SM1.1(溢出)相连,所以最后被移出的位被放到 SM1.1 位存储单元。移位时,移出位进入 SM1.1,另一端自动补 0。例如,在右移时,移位数据的最右端的位移入 SM1.1,则左端补 0。SM1.1 始终存放最后一次被移出的位,移位次数与移位数据的长度有关,如果所需移位次数大于移位数据的位数,则超出次数无效。如字左移时,若移位次数设定为 20,则指令实际执行结果只能移位 16 次,而不是设定值 20 次。如果移位操作使数据

变为 0, 则零存储器标志位(SM1.0)自动置位。

注意: 移位指令在使用 LAD 编程时, OUT 可以是和 IN 不同的存储单元, 但在使用 STL 编程时, 因为只写一个操作数, 所以实际上 OUT 就是移位后的 IN。

#### (1) 右移指令

指令格式: LAD 及 STL 格式如图 7-2(a)所示。图中□处可为 B、W、DW(LAD 中)或 D (STL 中)。

功能描述: 把字节型(字型或双字型)输入数据 IN 右移 N 位后, 再将结果输出到 OUT 所指的字节(字或双字)存储单元。最大实际可移位次数为 8 位(16 位或 32 位)。

数据类型: 输入输出均为字节(字或双字), N 为字节型数据。

#### (2) 左移指令

指令格式: LAD 及 STL 格式如图 7-2(b)所示。图中□处可为 B、W、DW(LAD 中)或 D (STL 中)。

功能描述: 把字节型(字型或双字型)输入数据 IN 左移 N 位后, 再将结果输出到 OUT 所指的字节(字或双字)存储单元。最大实际可移位次数为 8 位(16 位或 32 位)。

数据类型: 输入输出均为字节(字或双字), N 为字节型数据。

#### 【例 7-2】 移位指令举例。

```
LD      I0.0           //I0.0 有效时执行下面操作
SLB     VB0,2          //字节左移指令
SRW     VW10,3         //字右移指令
```

例题中若 VB0 中的内容为 00110101, 则执行 SLB 指令后, VB0 中的内容变为 11010100; 若 VW10 中的内容为 0011010100110101, 则执行 SRW 指令后, VW10 中的内容变为 0000011010100110。

### 2. 循环移位指令(Rotate)

循环移位指令包括循环左移和循环右移, 循环移位位数的长度分别为字节、字或双字。循环数据存储单元的移出端与另一端相连, 同时又与 SM1.1(溢出)相连, 所以最后被移出的位移到另一端的同时, 也被放到 SM1.1 位存储单元。例如在循环右移时, 移位数据的最右端位移入最左端, 同时又进入 SM1.1。SM1.1 始终存放最后一次被移出的位。移位次数与移位数据的长度有关, 如果移位次数设定值大于移位数据的位数, 则在执行循环移位之前, 系统先对设定值取以数据长度为底的模, 用小于数据长度的结果作为实际循环移位的次数。

#### (1) 循环右移指令

指令格式: LAD 及 STL 格式如图 7-2(c)所示。图中□处可为 B、W、DW(LAD 中)或 D (STL 中)。

功能描述: 把字节型(字型或双字型)输入数据 IN 循环右移 N 位后, 再将结果输出到 OUT 所指的字节(字或双字)存储单元。实际移位次数为系统设定值取以 8(16 或 32)为底的模所得的结果。

数据类型: 输入输出均为字节(字或双字), N 为字节型数据。

#### (2) 循环左移指令

指令格式: LAD 及 STL 格式如图 7-2(d)所示。图中□处可为 B、W、DW(LAD 中)或 D (STL 中)。

功能描述:把字节型(字型或双字型)输入数据 IN 循环左移 N 位后,再将结果输出到 OUT 所指的字节(字或双字)存储单元。实际移位次数为系统设定值取以 8(16 或 32)为底的模所得的结果。

数据类型:输入输出均为字节(字或双字),N 为字节型数据。

### 【例 7-3】 循环移位指令举例。

```
LD      I0.0           //I0.0 有效时执行下面操作
RRW     VW0.3          //循环右移指令
```

例题中若 VW0 中的内容为 1011010100110011,则执行 RRW 指令后,VW0 中的内容变为 0111011010100110。

### 3. 寄存器移位指令(Shift Register)

指令格式:LAD 及 STL 格式如图 7-2(e)所示。

功能描述:该指令在梯形图中有 3 个数据输入端,即 DATA 为数值输入,将该位的值移入移位寄存器;S\_BIT 为移位寄存器的最低位端;N 指定移位寄存器的长度。每次使能输入有效时,在每个扫描周期内,整个移位寄存器移动一位。所以要用边沿跳变指令来控制使能端的状态,不然该指令就失去了应用的意义。

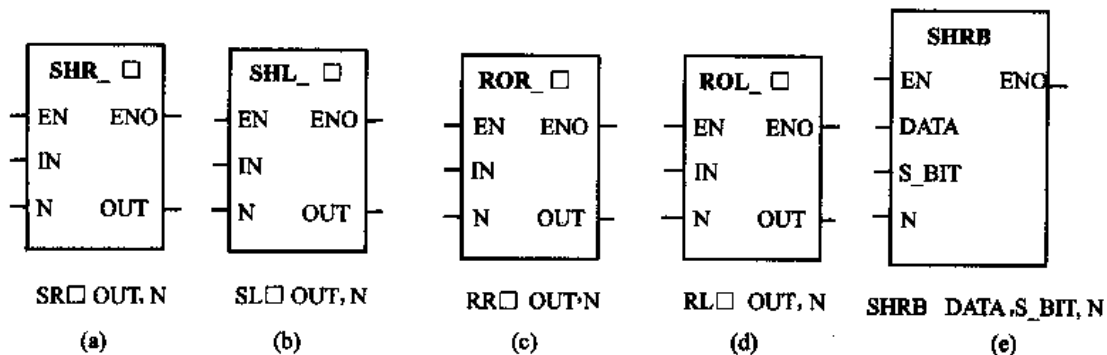


图 7-2 移位指令格式

移位寄存器存储单元的移出端与 SM1.1(溢出)相连,所以最后被移出的位放在 SM1.1 位存储单元。移位时,移出位进入 SM1.1,另一端自动补上 DATA 移入位的值。

移位方向分为正向移位和反向移位。正向移位时长度 N 为正值,移位是从最低字节的最低位(S\_BIT)移入,从最高字节的最高位移出;反向移位时长度 N 为负值,移位是从最高字节的最高位移入,从最低字节的最低位(S\_BIT)移出。

数据类型:DATA 和 S\_BIT 为 BOOL 型,N 为字节型,可以指定的移位寄存器最大长度为 64 位,可正可负。

最高位的计算方法:【N 的绝对值-1+(S\_BIT 的位号)】/8,余数即是最高位的位号,商与 S\_BIT 的字节号之和即是最高位的字节号。

例如,如果 S\_BIT 是 V33.4,N 是 14,则  $(14-1+4)/8 = 2$  余 1。所以,最高位字节号算法是:  $33+2 = 35$ ,位号为 1,即移位寄存器的最高位是 V35.1。

### 【例 7-4】 寄存器移位指令举例。

```
LD      I0.0
EU                               //在每个 I0.0 的上升沿移位 1 次
```

SHRB I0.5, V20.0, 5 //寄存器移位指令

SHRB 指令执行结果如表 7-1 所列。

表 7-1 指令 SHRB 执行结果

移位次数	I0.5 值	单元内容	位 SM1.1	说 明
0	1	10110101	X	移位前, 移位时从 VB20.4 移出
1	1	10101011	1	1 移入 SM1.1, I0.5 的值进入右端
2	0	10110111	0	0 移入 SM1.1, I0.5 的值进入右端
3	0	10101110	1	1 移入 SM1.1, I0.5 的值进入右端

### 7.1.3 字节交换指令

指令格式: LAD 及 STL 格式如图 7-3(a)所示。

功能描述: 字节交换指令(Swap Bytes)将字型输入数据 IN 的高字节和低字节进行交换。

数据类型: 输入为字。

【例 7-5】 字节交换指令举例。

LD I0.0 //I0.0 有效时执行下面操作

EU //在 I0.0 的上升沿执行

SWAP VW10 //字节交换指令

例题中若 VW10 中的内容为 1011010100000001, 则执行 SWAP 指令后, VW10 中的内容变为 0000000110110101。

### 7.1.4 填充指令

指令格式: LAD 及 STL 格式如图 7-3(b)所示。

功能描述: 填充指令(Memory Fill)将字型输入数据 IN 填充到从输出 OUT 所指的单元开始的 N 个字存储单元。

数据类型: IN 和 OUT 为字型, N 为字节型, 可取值范围为 1~255 的整数。

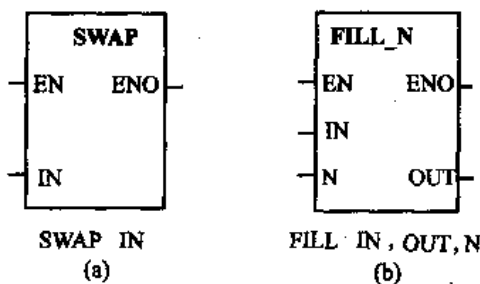


图 7-3 字节交换及填充指令格式

【例 7-6】 填充指令举例。

LD SM0.1 //初始化操作

FILL 10, VW100, 12 //填充指令

该例题的执行结果是将数据 10 填充到从 VW100 到 VW122 共 12 个字存储单元中。

## 7.2 运算和数学指令

运算功能的加入是现代 PLC 与以往 PLC 的最大区别之一,目前各种型号的 PLC 普遍具备较强的运算功能。和其他 PLC 不同,S7-200 对算术运算指令来说,在使用时要注意存储单元的分配。在用 LAD 编程时,IN1、IN2 和 OUT 可以使用不一样的存储单元,这样编写出的程序比较清晰易懂。但在用 STL 方式编程时,OUT 要和其中的一个操作数使用同一个存储单元,这样用起来较麻烦,编写程序和使用计算结果时都很不方便。LAD 格式程序转化为 STL 格式程序或 STL 格式程序转化为 LAD 格式程序时,会有不同的转换结果。所以建议大家在使用算术指令和数学指令时,最好用 LAD 形式编程。

注意:下面的运算指令 LAD 格式中的 IN1 和 STL 格式中的 IN1 不一定指的是同一个存储单元。

### 7.2.1 加法指令

加法指令(Add)是对有符号数进行相加操作。它包括整数加法、双整数加法和实数加法。

指令格式:LAD 及 STL 格式如图 7-4(a)所示。图中□处可为 I、DI(LAD 中)、D(STL 中)或 R。

功能描述:在 LAD 中, $IN1+IN2=OUT$ ;在 STL 中, $IN1+OUT=OUT$ 。

数据类型:整数加法时,输入输出均为 INT;双整数加法时,输入输出均为 DINT;实数加法时,输入输出均为 REAL。

### 7.2.2 减法指令

减法指令(Subtract)是对有符号数进行相减操作。它包括整数减法、双整数减法和实数减法。

指令格式:LAD 及 STL 格式如图 7-4(b)所示。图中□处可为 I、DI(LAD 中)、D(STL 中)或 R。

功能描述:在 LAD 中, $IN1-IN2=OUT$ ;在 STL 中, $OUT-IN1=OUT$ 。

数据类型:整数减法时,输入输出均为 INT;双整数减法时,输入输出均为 DINT;实数减法时,输入输出均为 REAL。

### 7.2.3 乘法指令

#### 1. 一般乘法指令(Multiply)

一般乘法指令是对有符号数进行相乘运算。它包括整数乘法、双整数乘法和实数乘法。

指令格式:LAD 及 STL 格式如图 7-4(c)所示。图中□处可为 I、DI(LAD 中)、D(STL 中)或 R。

功能描述:在 LAD 中, $IN1 \times IN2=OUT$ ;在 STL 中, $IN1 \times OUT=OUT$ 。

数据类型:整数乘法时,输入输出均为 INT;双整数乘法时,输入输出均为 DINT;实数乘法时,输入输出均为 REAL。

## 2. 完全整数乘法 (Multiply Integer to Double Integer)

将两个单字长(16 位)的符号整数 IN1 和 IN2 相乘,产生一个 32 位双整数结果 OUT。

指令格式:LAD 及 STL 格式如图 7-4(d)所示。

功能描述:在 LAD 中, $IN1 \times IN2 = OUT$ ;在 STL 中, $IN1 \times OUT = OUT$ ,32 位运算结果存储单元的低 16 位运算前用于存放被乘数。

数据类型:输入为 INT,输出为 DINT。

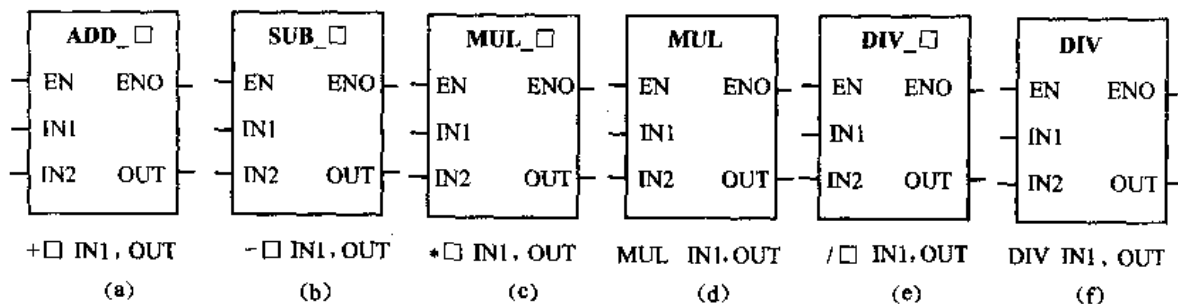


图 7-4 算术运算指令格式

## 7.2.4 除法指令

### 1. 一般除法指令 (Divide)

一般除法指令是对有符号数进行相除操作。它包括整数除法、双整数除法和实数除法。

指令格式:LAD 及 STL 格式如图 7-4(e)所示。图中□处可为 I、DI(LAD 中)、D(STL 中)或 R。

功能描述:在 LAD 中, $IN1/IN2 = OUT$ ;在 STL 中, $OUT/IN1 = OUT$ 。不保留余数。

数据类型:整数除法时,输入输出均为 INT;双整数除法时,输入输出均为 DINT;实数除法时,输入输出均为 REAL。

### 2. 完全整数除法 (Divide Integer to Double Integer)

将两个 16 位的符号整数相除,产生一个 32 位结果,其中,低 16 位为商,高 16 位为余数。

指令格式:LAD 及 STL 格式如图 7-4(f)所示。

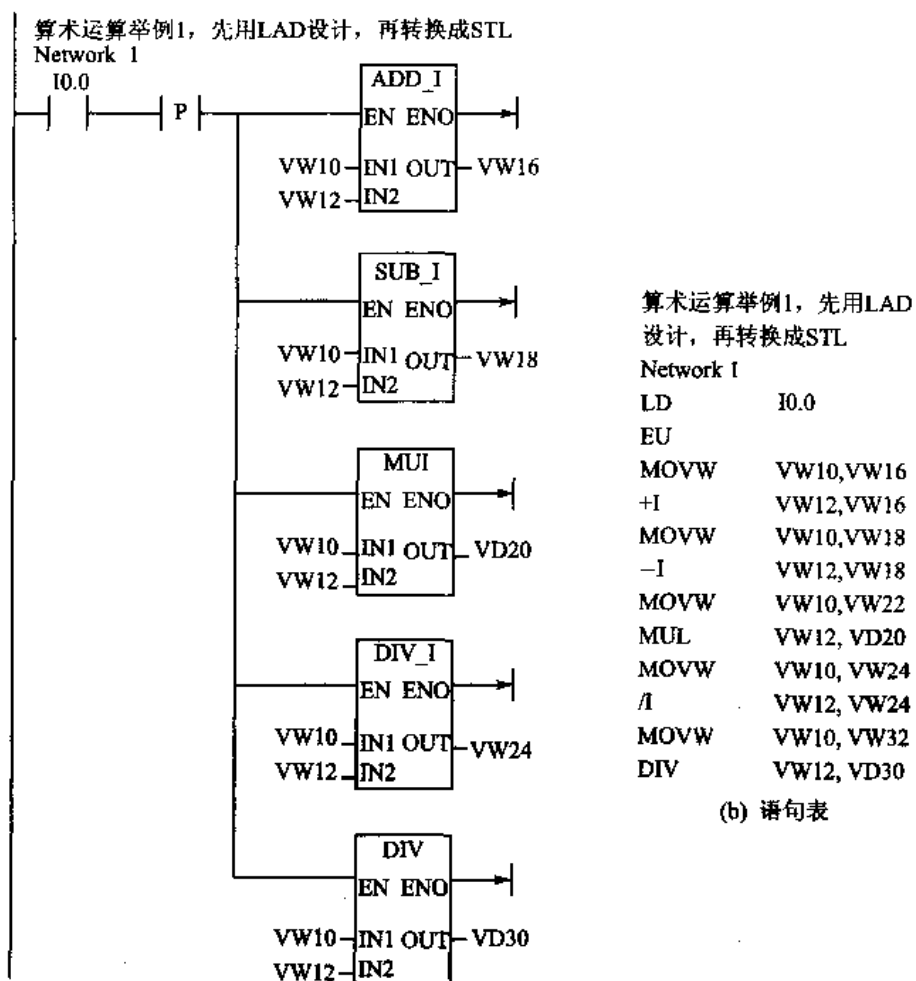
功能描述:在 LAD 中, $IN1/IN2 = OUT$ ;在 STL 中, $OUT/IN1 = OUT$ ,32 位结果存储单元的低 16 位运算前被兼用存放被除数。除法运算结果:商放在 OUT 的低 16 位字中,余数放在 OUT 的高 16 位字中。

数据类型:输入为 INT,输出为 DINT。

【例 7-7】 算术运算指令综合举例 1。该例先设计梯形图程序如图 7-5(a)所示,而图 7-5(b)是对应的语句表。

本例中若  $VW10=2\ 000$ ,  $VW12=150$ ,则执行完该段程序后,各有关结果存储单元的数值为:  $VW16=2\ 150$ ,  $VW18=1\ 850$ ,  $VD20=300\ 000$ ,  $VW24=13$ ,  $VW30=50$ ,  $VW32=13$ 。

【例 7-8】 算术运算指令综合举例 2。该例先设计语句表程序如图 7-6(a)所示,图 7-6(b)是对应的梯形图程序。大家可以比较和【例 7-7】的不同。



(b) 语句表

(a) 梯形图

图 7-5 算术运算指令实例 1

算术运算举例2, 先用STL设计, 再转换成梯形图

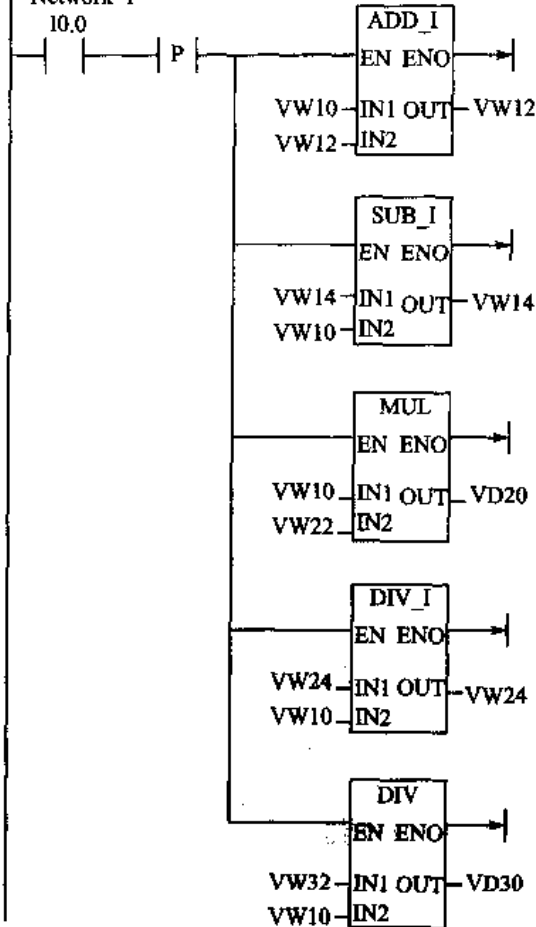
Network 1

```
LD    I0.0
EU
+I     VW10,VW12
-I     VW10,VW14
MUL    VW10,VD20
/I     VW10,VW24
DIV    VW10,VD30
```

(a) 语句表

算术运算举例2, 先用STL设计, 再转换成梯形图

Network 1



(b) 梯形图

图 7-6 算术运算指令实例 2

### 7.2.5 数学函数指令

S7-200 PLC 的数学函数指令有:平方根、自然对数、指数、正弦、余弦和正切。运算输入输出数据都为实数。结果大于 32 位二进制数表示的范围时产生溢出。

#### 1. 平方根指令(Square Root)

指令格式:LAD 及 STL 格式如图 7-7(a)所示。

功能描述:把一个双字长(32 位)的实数 IN 开平方,得到 32 位的实数结果送到 OUT。

数据类型:输入输出均为 REAL。

#### 2. 自然对数指令(Natural Logarithm)

指令格式:LAD 及 STL 格式如图 7-7(b)所示。

功能描述:将一个双字长(32 位)的实数 IN 取自然对数,得到 32 位的实数结果送到 OUT。

数据类型:输入输出均为 REAL。

当求解以 10 为底的常用对数时,可以用(/R)DIV-R 指令将自然对数除以 2.302 585 即可(LN10 的值约为 2.302 585)。

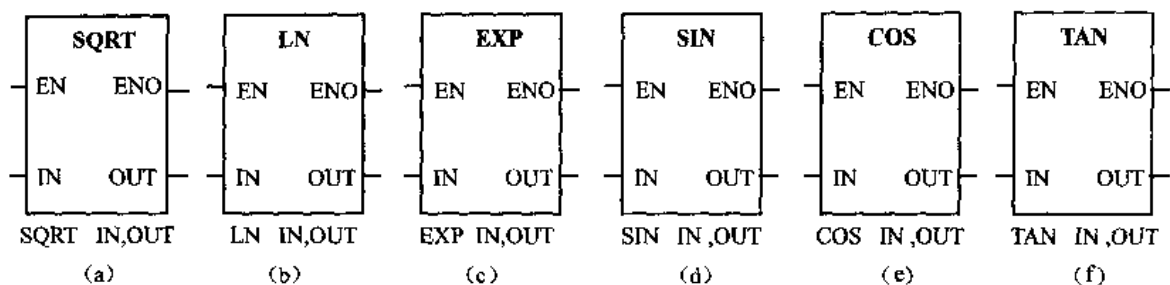


图 7-7 数学函数指令格式

【例 7-9】求以 10 为底的 50(存于 VD0)的常用对数,结果放到 AC0,运算程序如图 7-8 所示。

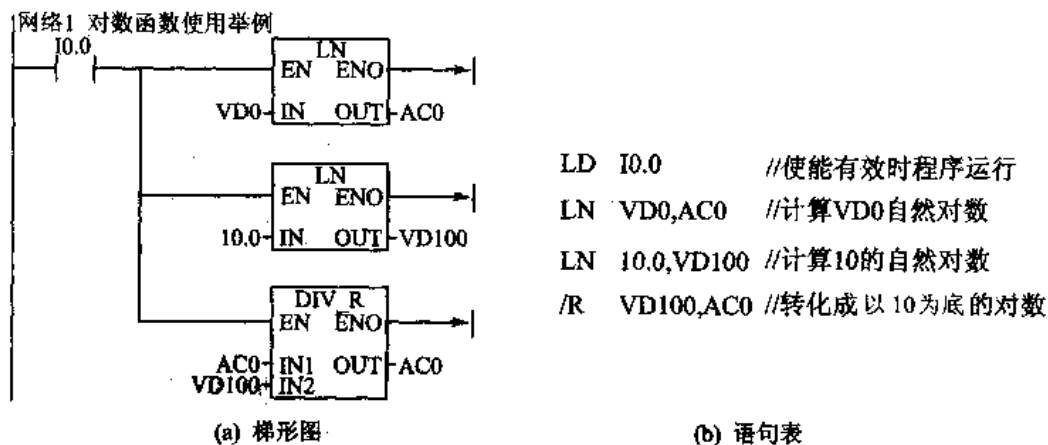


图 7-8 对数函数指令实例

### 3. 指数指令(Natural Exponential)

指令格式:LAD 及 STL 格式如图 7-7(c)所示。

功能描述:将一个双字长(32 位)的实数 IN 取以 e 为底的指数,得到 32 位的实数结果送到 OUT。

数据类型:输入输出均为 REAL。

可以用指数指令和自然对数指令相配合来完成以任意常数为底和以任意常数为指数的计算。

例如:18 的 6 次方 =  $18^6 = \text{EXP}(6 * \text{LN}(18))$

125 的 3 次方根 =  $125^{(1/3)} = \text{EXP}(1/3 * \text{LN}(125)) = 5$

### 4. 正弦(sine)、余弦(cosine)和正切(tan)指令

指令格式:LAD 及 STL 格式如图 7-7(d)、(e)和(f)所示。

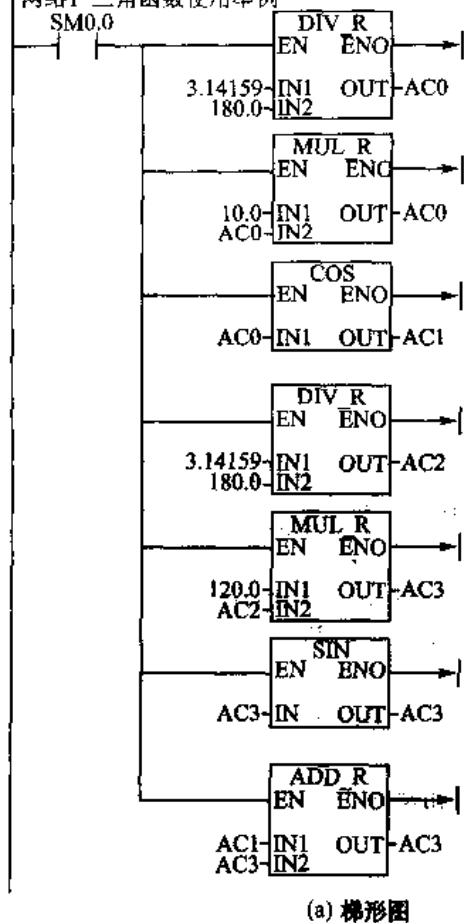
功能描述:将一个双字长(32 位)的实数弧度值 IN 分别取正弦、余弦、正切,各得到 32 位的实数结果送到 OUT。

数据类型:输入输出均为 REAL。

如果已知输入值为角度,要先将角度值转化为弧度值,方法是使用(\*R)MUL\_R 指令,把角度值乘以  $\pi/180^\circ$  即可。

【例 7-10】求  $\sin 120^\circ + \cos 10^\circ$  的值。程序如图 7-9 所示。

网络1 三角函数使用举例



(a) 梯形图

```

LD    SM0.0    //使能有效时程序运行
MOVW  3.14159,AC0 //计算10°的弧度值
/R    180.0,AC0
*R    10.0,AC0
COS   AC0,AC1    //计算cos10°
MOVW  3.14159,AC2 //计算120°的弧度值
/R    180.0,AC2
MOVW  120.0,AC3
*R    AC2,AC3
SIN   AC3,AC3    //计算sin120°
+R    AC1,AC3    //计算sin120° +cos10°

```

(b) 语句表

图 7-9 三角函数指令实例

### 7.2.6 增/减指令

增/减指令又称自增和自减指令。它是对无符号或有符号整数进行自动加1或减1的操作,数据长度可以是字节、字或双字。其中字节增减是对无符号数操作,而字或双字的增减是对有符号数操作。

#### 1. 增指令(Increment)

增指令包括字节增、字增和双字增指令。

指令格式:LAD及STL格式如图7-10(a)所示。图中□处可为B、W、DW(LAD中)或D(STL中)。

功能描述:在LAD中, $IN+1=OUT$ ;在STL中, $OUT+1=OUT$ ,即IN和OUT使用同一个存储单元。

数据类型:字节增指令输入输出均为字节;字增指令输入输出均为INT,双字增指令输入输出均为DINT。

#### 2. 减指令(Decrement)

减指令包括字节减、字减和双字减指令。

指令格式:LAD及STL格式如图7-10(b)所示。图中□处可为B、W、DW(LAD中)或

D(STL 中)。

功能描述:在 LAD 中, $IN-1=OUT$ ;在 STL 中, $OUT-1=OUT$ ,即 IN 和 OUT 使用同一个存储单元。

数据类型:字节减指令输入输出均为字节,字减指令输入输出均为 INT,双字减指令输入输出均为 DINT。

【例 7-11】 增减指令使用举例。梯形图程序如图 7-11(a)所示。图 7-11(b)为 LAD 对应的 STL 形式,请体会使用 LAD 和 STL 编程的不同。

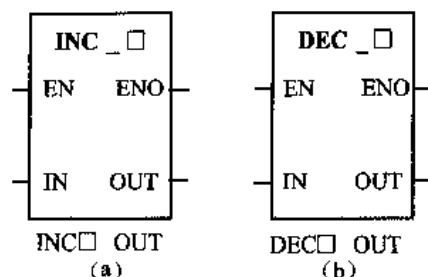


图 7-10 增减指令格式

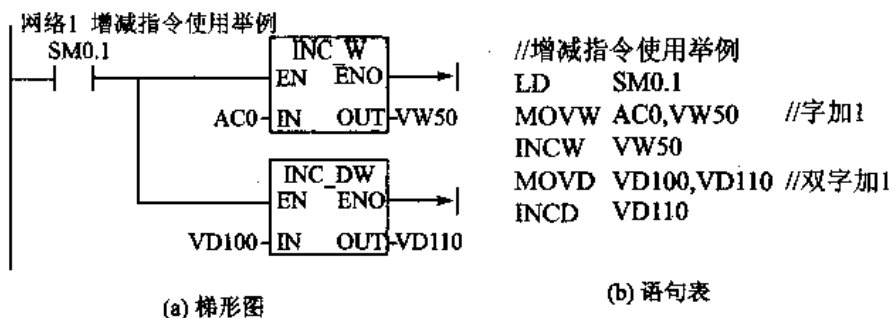


图 7-11 增减指令使用举例

### 7.2.7 逻辑运算指令

逻辑运算对逻辑数(无符号数)进行处理,按运算性质不同,有逻辑与、逻辑或、逻辑异或和取反等。参与运算的操作数可以是字节、字或双字。

#### 1. 逻辑与运算指令(Logical And)

它包括字节、字和双字的逻辑与运算指令。

指令格式:LAD 及 STL 格式如图 7-12(a)所示。图中□处可为 B、W、DW(LAD 中)或 D(STL 中)。

功能描述:把两个一个字节(字或双字)长的输入逻辑数按位相与,得到一个字节(字或双字)的逻辑数并输出到 OUT。在 STL 中 OUT 和 IN2 使用同一个存储单元。

数据类型:输入输出均为字节(字或双字)。

#### 2. 逻辑或运算指令(Logical Or)

它包括字节、字和双字的逻辑或运算指令。

指令格式:LAD 及 STL 格式如图 7-12(b)所示。图中□处可为 B、W、DW(LAD 中)或 D(STL 中)。

功能描述:把两个一个字节(字或双字)长的输入逻辑数按位相或,得到一个字节(字或双字)的逻辑数并输出到 OUT。在 STL 中 OUT 和 IN2 使用同一个存储单元。

数据类型:输入输出均为字节(字或双字)。

#### 3. 逻辑异或运算指令(Logical Exclusive Or)

它包括字节、字和双字的逻辑异或运算指令。

指令格式:LAD 及 STL 格式如图 7-12(c)所示。图中□处可为 B、W、DW(LAD 中)或 D(STL 中)。

功能描述:把两个一个字节(字或双字)长的输入逻辑数按位相异或,得到一个字节(字或双字)的逻辑数并输出到 OUT。在 STL 中 OUT 和 IN2 使用同一个存储单元。

数据类型:输入输出均为字节(字或双字)。

#### 4. 取反指令(Logic Invert)

它包括对字节、字和双字的逻辑取反指令。

指令格式:LAD 及 STL 格式如图 7-12(d)所示。图中□处可为 B、W、DW(LAD 中)或 D(STL 中)。

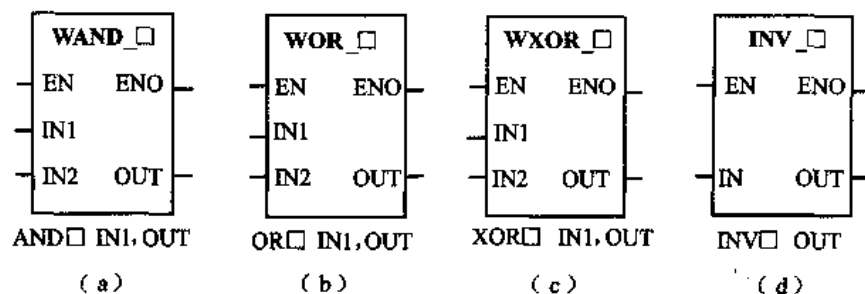


图 7-12 逻辑运算指令格式

功能描述:把两个一个字节(字或双字)长的输入逻辑数按位取反,得到一个字节(字或双字)的逻辑数并输出到 OUT。在 STL 中 OUT 和 IN 使用同一个存储单元。

数据类型:输入输出均为字节(字或双字)。

#### 【例 7-12】 逻辑运算指令使用举例。

```
LD      I0.0
EU                               //I0.0 上升沿时执行下面操作
ANDB    VB0, AC1                //字节逻辑与
ORB      VB0, AC0                //字节逻辑或
XORB     VB0, AC2                //字节逻辑异或
INVB     VB10                    //字节逻辑取反
```

该例题的执行结果如表 7-2 所列(各单元内容都用二进制数表示)。

表 7-2 指令执行情况表

指令	操作数	地址单元	单元长度(n 字节)	运算前值	运算结果值
ANDB	IN1	VB0	1	01010011	01010011
	IN2(OUT)	AC1	1	11110001	01010001
ORB	IN1	VB0	1	01010011	01010011
	IN2(OUT)	AC0	1	00110110	01110111
XORB	IN1	VB0	1	01010011	01010011
	IN2(OUT)	AC2	1	11011010	10001001
INVB	IN(OUT)	VB10	1	01010011	10101100

### 7.3 表功能指令

在 S7-200 PLC 指令系统中,一个表由表地址(表的首地址)指明。表地址和第二个字地址所对应的单元分别存放两个表参数(最大填表数 TL 和实际填表数 EC),之后是最多 100 个填表数据。

表只对字型数据存储,表的格式举例如表 7-3 所列。

表 7-3 数据表格式

单元地址	单元内容	说 明
VW100	0006	TL=6,最多可填 6 个数,VW100 为表地址
VW102	0004	EC=4,实际在表中存有 4 个数据
VW104	1203	数据 0
VW106	4467	数据 1
VW108	9086	数据 2
VW110	3592	数据 3
VW112	****	无效数据
VW114	****	无效数据

#### 1. 表存数指令(Add To Table)

指令格式:LAD 及 STL 指令格式如图 7-13(a)所示。

功能描述:该指令在梯形图中有 2 个数据输入端,即 DATA 为数值输入,指出将被存储的字型数据;TBL 为表格的首地址,用以指明被访问的表格。当使能输入有效时,将输入字型数据添加到指定的表格中。

表存数时,新存的数据添加在表中最后一个数据的后面。每向表中存一个数据,实际填表数 EC 会自动加 1。

数据类型:DATA 为 INT,TBL 为字。

【例 7-13】对表 7-3 执行程序:ATT VW200,VW100。

若指令执行前 VW200 中的内容为 222,则指令执行结果如表 7-4 所列。

表 7-4 指令 ATT 执行结果

操作数	单元地址	执行前内容	执行后单元内容	说 明
DATA	VW200	222	222	被填表数据地址
TBL	VW100	0006	0006	TL=6,最大填表数为 6,不变化
	VW102	0004	0005	EC 实际存表数由 4 加 1 变为 5
	VW104	1203	1203	数据 0
	VW106	4467	4467	数据 1
	VW108	9086	9086	数据 2
	VW110	3592	3592	数据 3
	VW112	****	222	将 VW200 中的数据填入表中
	VW114	****	****	无效数据

## 2. 表取数指令

从表中取出一个字型数据可有两种方式:先进先出式和后进先出式。一个数据从表中取出之后,表的实际填表数 EC 值减少 1。两种方式的指令在梯形图中有 2 个数据端:输入端 TBL 为表格的首地址,用以指明访问的表格;输出端 DATA 指明数值取出后要存放的目标单元。如果指令试图从空表中取走一个数值,则特殊标志寄存器位 SM1.5 置位。

### (1) 先进先出指令(First-In-First-Out)

指令格式:LAD 及 STL 格式如图 7-13(b)所示。

功能描述:从 TBL 指定的表中移出第一个字型数据并将其输出到 DATA 所指定的字存储单元。取数时,移出的数据总是最先进入表中的数据。每次从表中移出一个数据,剩余数据则依次上移一个字单元位置,同时实际填表数 EC 会自动减 1。

数据类型:DATA 为 INT, TBL 为字。

【例 7-14】对表 7-3 执行程序:FIFO VW100, AC0。

则指令执行结果如表 7-5 所列。

表 7-5 指令 FIFO 执行结果

操作数	单元地址	执行前内容	执行后内容	说明
DATA	AC0	任意数	1203	从表中取走的数据输出到 AC0
TBL	VW100	0006	0006	TL=6,最大填表数为 6,不变化
	VW102	0004	0003	EC 实际存表数由 4 减 1 变为 3
	VW104	1203	4467	数据 0,剩余数据依次上移一格
	VW106	4467	9086	数据 1
	VW108	9086	3592	数据 2
	VW110	3592	****	无效数据
	VW112	****	****	无效数据
	VW114	****	****	无效数据

### (2) 后进先出指令(Last-In-First-Out)

指令格式:LAD 格式如图 7-13(c)所示。

功能描述:从 TBL 指定的表中取出最后一个字型数据并将其输出到 DATA 所指定的字存储单元。取数时,移出的数据是最后进入表中的数据。每次从表中取出一个数据,剩余数据位置保持不变,实际填表数 EC 会自动减 1。

数据类型:DATA 为字, TBL 为 INT。

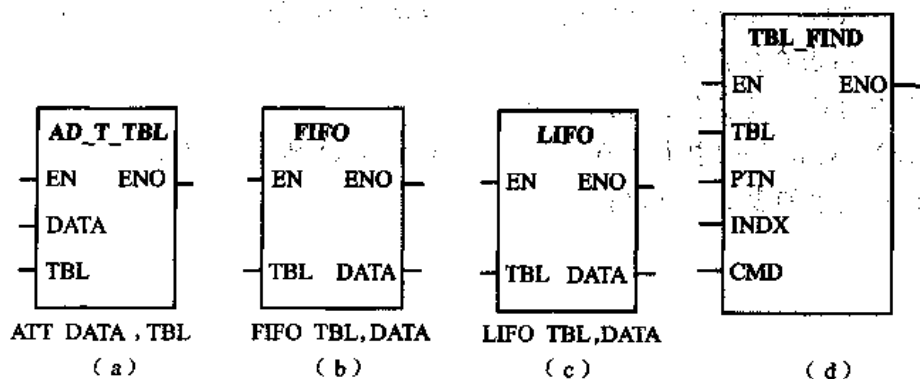


图 7-13 表功能指令格式

【例 7-15】对表 7-3 执行程序:LIFO VW100,AC0。

则指令执行结果如表 7-6 所列。

表 7-6 指令 LIFO 执行结果

操作数	单元地址	执行前内容	执行后内容	说 明
DATA	AC0	任意数	3592	从表中取走的数据输出到 AC0
TBL	VW100	0006	0006	TL=6,最大填表数为 6,不变化
	VW102	0004	0003	EC 实际存表数由 4 减 1 变为 3
	VW104	1203	1203	数据 0,剩余数据不移动
	VW106	4467	4467	数据 1
	VW108	9086	9086	数据 2
	VW110	3592	* * * *	无效数据
	VW112	* * * *	* * * *	无效数据
	VW114	* * * *	* * * *	无效数据

### 3. 表查找指令(Table Find)

通过表查找指令可以从数据表中找出符合条件数据的表中编号,编号范围为 0~99。

指令格式:LAD 格式如图 7-13(d)所示。

STL 格式:FND= TBL,PTN,INDX (查找条件: =PTN)  
 FND<> TBL,PTN,INDX (查找条件: <>PTN)  
 FND< TBL,PTN,INDX (查找条件: <PTN)  
 FND> TBL,PTN,INDX (查找条件: >PTN)

功能描述:在梯形图中有 4 个数据输入端,即 TBL 为表格的首地址,用以指明被访问的表格;PTN 是用来描述查表条件时进行比较的数据;CMD 是比较运算符“?”的编码,它是一个 1~4 的数值,分别代表 =、<>、<和>运算符;INDX 用来存放表中符合查找条件的数据的地址。

由 PTN 和 CMD 就可以决定对表的查找条件。例如,PTN 为 16 # 2555,CMD 为 3,则查找条件为“<16 # 2555”。

表查找指令执行之前,应先对 INDX 的内容清 0。当使能输入有效时,从 INDX 开始搜索表 TBL,寻找符合由 PTN 和 CMD 所决定的条件的数据,如果没有发现符合条件的数据,则 INDX 的值等于 EC。如果找到一个符合条件的数据,则将该数据的表中地址装入 INDX。

数据类型:TBL、INDX 为字,PTN 为 INT,CMD 为字节型常数。

表查找指令执行完成,找到一个符合条件的数据,如果想继续向下查找,必须先对 INDX 加 1,然后重新激活表查找指令。

在语句表中运算符直接表示,而不用各自的编码。

【例 7-16】对表 7-3 执行程序:FND> VW100,VW300,AC0。

指令的执行结果如表 7-7 所列。

表 7-7 表查找指令执行结果

操作数	单元地址	执行前内容	执行后内容	说 明
PTN	VW300	5000	5000	用来比较的数据
INDX	AC0	0	2	符合查表条件的单元地址
CMD	无	4	4	4 表示为 >
TBL	VW100	0006	0006	TL=6, 最大填表数, 不变化
	VW102	0004	0004	EC 实际存表数, 不变化
	VW104	1203	1203	数据 0
	VW106	4467	4467	数据 1
	VW108	9086	9086	数据 2
	VW110	3592	3592	数据 3
	VW112	****	****	无效数据
	VW114	****	****	无效数据

## 7.4 转换指令

转换指令是指对操作数的类型进行转换, 包括数据的类型转换、码的类型转换以及数据和码之间的类型转换。

### 7.4.1 数据类型转换指令

可编程序控制器中的主要数据类型包括字节、整数、双整数和实数。主要的码制有 BCD 码、ASCII 码、十进制数和十六进制数等。不同性质的指令对操作数的类型要求不同, 因此在指令使用之前需要将操作数转化成相应的类型, 转换指令可以完成这样的任务。

#### 1. 字节与整数

##### (1) 字节到整数(Byte to Integer)

指令格式: LAD 及 STL 格式如图 7-14(a) 所示。

功能描述: 将字节型输入数据 IN 转换成整数类型, 并将结果送到 OUT 输出。字节型是无符号的, 所以没有符号扩展位。

数据类型: 输入为字节, 输出为 INT。

##### (2) 整数到字节(Integer to Byte)

指令格式: LAD 及 STL 格式如图 7-14(b) 所示。

功能描述: 将整数输入数据 IN 转换成字节类型, 并将结果送到 OUT 输出。输入数据超出字节范围(0~255)时产生溢出。

数据类型: 输入为 INT, 输出为字节。

#### 2. 整数与双整数

##### (1) 双整数到整数(Double Integer to Integer)

指令格式: LAD 及 STL 格式如图 7-14(c) 所示。

功能描述: 将双整数输入数据 IN 转换成整数类型, 并将结果送到 OUT 输出。输出数据超出整数范围则产生溢出。

数据类型:输入为 DINT,输出为 INT。

(2) 整数到双整数(Integer to Double Integer)

指令格式:LAD 及 STL 格式如图 7-14(d)所示。

功能描述:将整数输入数据 IN 转换成双整数类型(符号进行扩展),并将结果送到 OUT 输出。

数据类型:输入为 INT,输出为 DINT。

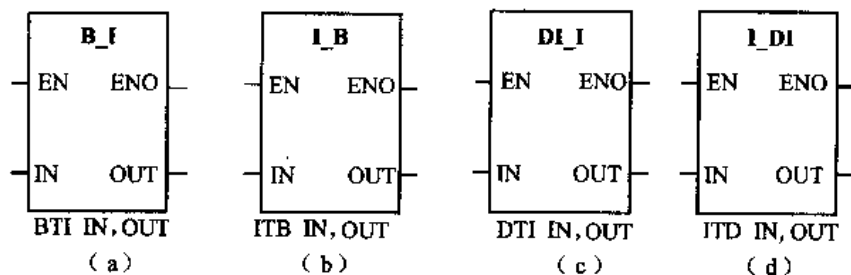


图 7-14 数据类型转换指令格式(1)

### 3. 双整数与实数

(1) 实数到双整数(Real to Double Integer)

实数转换为双整数,其指令有两条:ROUND 和 TRUNC。

指令格式:LAD 及 STL 格式如图 7-15(a)和(b)所示。

功能描述:将实型输入数据 IN 转换成双整数类型,并将结果送到 OUT 输出。两条指令的区别是:前者小数部分 4 舍 5 入,而后者小数部分直接舍去。

数据类型:输入为 REAL,输出为 DINT。

(2) 双整数到实数(Double Integer to Real)

指令格式:LAD 及 STL 格式如图 7-15(c)所示。

功能描述:将双整数输入数据 IN 转换成实数,并将结果送到 OUT 输出。

数据类型:输入为 DINT,输出为 REAL。

(3) 整数到实数(Integer to Real)

没有直接的整数到实数转换指令。转换时,先使用 I\_DI(整数到双整数)指令,然后再使用 DTR(双整数到实数)指令即可。

### 4. 整数与 BCD 码

(1) BCD 码到整数(BCD to Integer)

指令格式:LAD 及 STL 格式如图 7-15(d)所示。

功能描述:将 BCD 码输入数据 IN 转换成整数类型,并将结果送到 OUT 输出。输入数据 IN 的范围为 0~9 999。在 STL 中,IN 和 OUT 使用相同的存储单元。

数据类型:输入输出均为字。

(2) 整数到 BCD 码(Integer to BCD)

指令格式:LAD 及 STL 格式如图 7-15(e)所示。

功能描述:将整数输入数据 IN 转换成 BCD 码类型,并将结果送到 OUT 输出。输入数据 IN 的范围为 0~9 999。在 STL 中,IN 和 OUT 使用相同的存储单元。

数据类型:输入输出均为字。

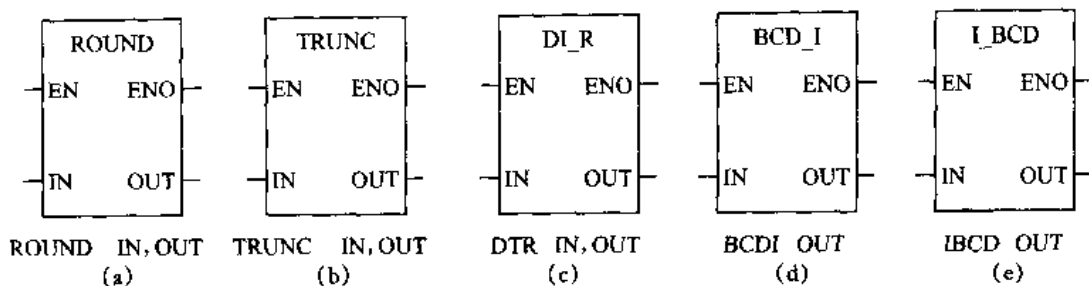


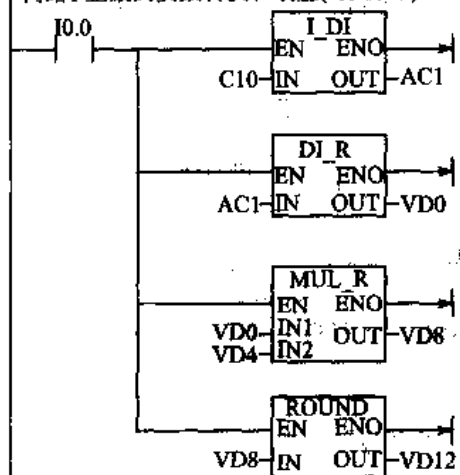
图 7-15 数据类型转换类指令格式(2)

【例 7-17】 转换指令使用举例。

网络 1: 将英寸长度转化成 cm 长度。其中 C10 中存放英寸长度, VD4 存放转换系数 2.54。网络 2: BCD 码与整数转换举例。程序如图 7-16 所示。

若 C10 为 101(英寸), VD4=2.54(常数), 则执行本程序后, VD0=101.0(实数), VD8=256.54(实数), VD12=257(整数)。对网络 2, 若 VW10=1 234(应当作 BCD 码), 则经过 BCD\_I 转换后, VW20=1234(即 16#04D2); 若 VW12=1 234, 则经过 I\_BCD 转换后, VW22=16#1234。

网络1 整数到实数转换和取整(ROUND)

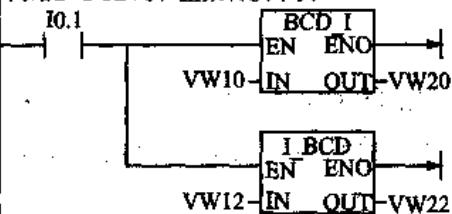


(a) 梯形图

NETWORK 1 //整数到实数转换和取整(ROUND)

```
LD      I0.0
ITD     C10, AC1  //将英寸值转化成双整数
DTR     AC1, VD0  //再转化成浮点数
MOVR    VD0, VD8  //与2.54相乘, 得到cm
*R      VD4, VD8
ROUND   VD8, VD12 //转化成整数
```

网络2 BCD码和整数转换举例



NETWORK 2 //BCD码和整数转换举例

```
LD      I0.1
MOVW    VW10, VW20
BCDI     VW20
MOVW    VW12, VW22
IBCD     VW22
```

(b) 语句表

图 7-16 数据类型转化程序举例

### 7.4.2 编码和译码指令

#### 1. 编码指令(Encode)

指令格式:LAD 及 STL 格式如图 7-17(a)所示。

功能描述:将字型输入数据 IN 的最低有效位(值为 1 的位)的位号输出到 OUT 所指定的字节单元的低 4 位,即用半个字节来对一个字型数据 16 位中的“1”位有效位进行编码。

数据类型:输入为字,输出为字节。

【例 7-18】 执行程序: ENCO VW0, VB10。

本例若 VW0 的内容为:0010101001000000,即最低为 1 的位是第 6 位,则执行编码指令后,VB10 的内容为:00000110(即 06)。

#### 2. 译码指令(Decode)

指令格式:LAD 及 STL 格式如图 7-17(b)所示。

功能描述:将字节型输入数据 IN 的低 4 位所表示的位号对 OUT 所指定的字单元的对应位置 1,其他位置 0。即对半个字节的编码进行译码,以选择一个字型数据 16 位中的“1”位。

数据类型:输入为字节,输出为字。

【例 7-19】 执行程序:DECO VB0, VB10。

本例若 VB0 的内容为:00000111(即 07),则执行译码指令后,VB10 的内容为:0000000010000000,即第 7 位为 1,其余位为 0。

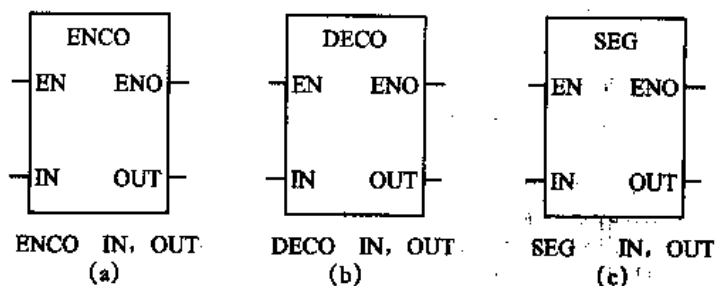


图 7-17 编码、译码及七段码指令格式

### 7.4.3 段码指令

指令格式:LAD 及 STL 格式如图 7-17(c)所示。

功能描述:段码指令(Segment)将字节型输入数据 IN 的低 4 位有效数字产生相应的七段码,并将其输出到 OUT 所指定的字节单元。该指令在数码显示时直接应用非常方便。七段码编码如表 7-8 所列。

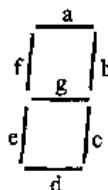
数据类型:输入输出均为字节。

【例 7-20】 执行程序:SEG VB10, QB0。

若设 VB10 = 05,则执行上述指令后,在 Q0.0~Q0.7 上可以输出 01101101。

表 7-8 七段码编码表

段显示	- g f e d c b a	段显示	- g f e d c b a
0	0 0 1 1 1 1 1 1	8	0 1 1 1 1 1 1 1
1	0 0 0 0 0 1 1 0	9	0 1 1 0 0 1 1 1
2	0 1 0 1 1 0 1 1	a	0 1 1 1 0 1 1 1
3	0 1 0 0 1 1 1 1	b	0 1 1 1 1 1 0 0
4	0 1 1 0 0 1 1 0	c	0 0 1 1 1 0 0 1
5	0 1 1 0 1 1 0 1	d	0 1 0 1 1 1 1 0
6	0 1 1 1 1 1 0 1	e	0 1 1 1 1 0 0 1
7	0 0 0 0 0 1 1 1	f	0 1 1 1 0 0 0 1



#### 7.4.4 ASCII 码转换指令

ASCII 码转换指令是将标准字符 ASCII 编码与 16 进制值、整数、双整数及实数之间进行转换。可进行转换的 ASCII 码为 30~39 和 41~46, 对应的十六进制数为 0~9 和 A~F。

##### 1. ASCII 码转换为 16 进制数指令(ASCII to HEX)

指令格式:LAD 及 STL 格式如图 7-18(a)所示。

功能描述:把从 IN 开始的长度为 LEN 的 ASCII 码转换为 16 进制数,并将结果送到 OUT 开始的字节进行输出。LEN 的长度最大为 255。

数据类型:IN、LEN 和 OUT 均为字节类型。

【例 7-21】 执行程序:ATH VB30,VB40,3。

在本例给定的输入条件下,则经过 ATH 后,结果如下:

ASCII 码表示:	'2'	'E'	'A'	
16 进制数表示:	32	45	41	ATH→ 2E AX
	VB30	VB31	VB32	VB40 VB41

注意:X 表示 VB41 的低四位(半个字节)未发生变化。

##### 2. 16 进制转换为 ASCII 码指令(HEX to ASCII)

指令格式:LAD 及 STL 格式如图 7-18(b)所示。

功能描述:把从 IN 开始的长度为 LEN 的 16 进制数转换为 ASCII 码,并将结果送到 OUT 开始的字节进行输出。LEN 的长度最大为 255。

数据类型:IN、LEN 和 OUT 均为字节类型。

【例 7-22】 执行程序:HTA VB10,VB20,4。

在本例给定的输入条件下,则经过 HTA 后,结果如下:

ASCII 码表示:			'2'	'4'	'A'	'E'
16 进制数表示:	24	AE	HTA→ 32	34	41	45
	VB10	VB11	VB20	VB21	VB22	VB23

### 3. 整数转换为 ASCII 码指令(Integer to ASCII)

指令格式:LAD 及 STL 格式如图 7-18(c)所示。

功能描述:把一个整数 IN 转换成一个 ASCII 码字符串。格式 FMT 指定小数点右侧的转换精度和小数点是使用逗号还是使用点号。转换结果放在 OUT 指定的 8 个连续的字节中。

数据类型:IN 为整数、FMT 和 OUT 均为字节类型。

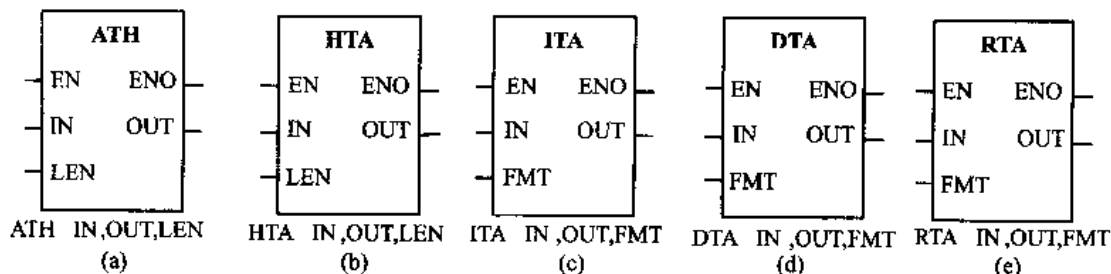


图 7-18 ASCII 码转换指令

FMT 操作数格式如图 7-19(a)所示。nnn 指定输出缓冲区中小数点右侧的位数,其有效范围是 0~5,如果 nnn=0,则没有小数;如果 nnn>5,则用 ASCII 码空格键填充整个缓冲区。c 指定用逗号(c=1)还是用点号(c=0)作为整数和小数部分的分隔符,FMT 的高 4 位必须为 0。图 7-19(b)给出了一个数值的例子,其格式位 c=0,nnn=011。

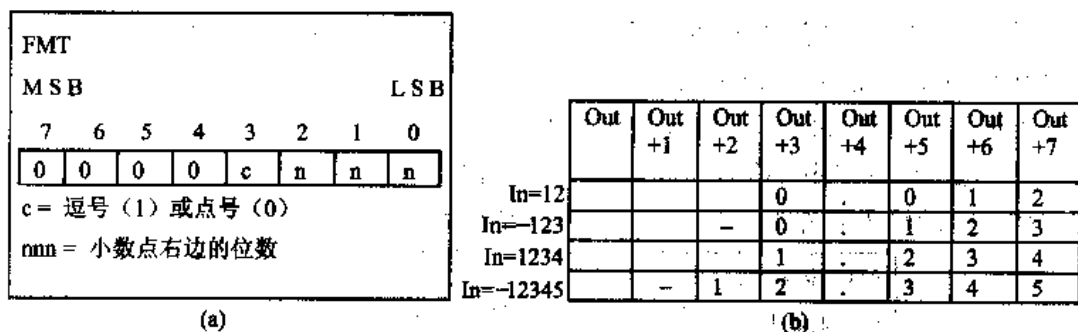


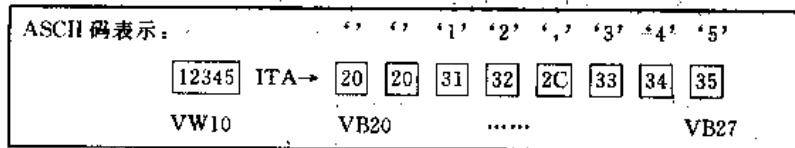
图 7-19 ITA 指令的 FMT 操作数格式及举例

输出缓冲区的格式符合下面的规则:

- (1) 正数写入 OUT 时没有符号;
- (2) 负数写入 OUT 时带负号;
- (3) 小数点左侧的 0(除去靠近小数点的那个 0)被隐藏;
- (4) OUT 中的数字右对齐。

【例 7-23】执行程序:ITA VW10,VB20,16#0B。

16#0B 表示用逗号作小数点,保留 3 位小数。在本例给定的输入条件下,则经过 ITA 后,结果如下:



#### 4. 双整数转换为 ASCII 码(Double Integer to ASCII)

指令格式:LAD 及 STL 格式如图 7-18(d)所示。

功能描述:把一个双整数 IN 转换成一个 ASCII 码字符串。格式 FMT 指定小数点右侧的转换精度和小数点是使用逗号还是使用点号。转换结果放在 OUT 指定的连续 12 个字节中。

数据类型:IN 为双整数、FMT 和 OUT 均为字节类型。

DTA 指令的 OUT 比 ITA 指令多 4 个字节,其余都和 ITA 指令一样。

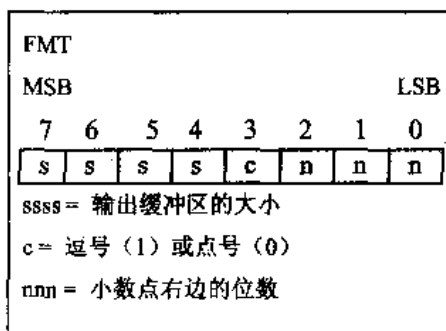
#### 5. 实数转换为 ASCII 码(Real to ASCII)

指令格式:LAD 及 STL 格式如图 7-18(e)所示。

功能描述:把一个实数 IN 转换成一个 ASCII 码字符串。格式 FMT 指定小数点右侧的转换精度和小数点是使用逗号还是使用点号,转换结果放在 OUT 开始的 3~15 个字节中。

数据类型:IN 为实数、FMT 和 OUT 均为字节类型。

FMT 的格式操作数如图 7-20(a)所示。ssss 指定 OUT 的大小,它的范围是 3~15。nnn 指定输出缓冲区中小数点右侧的位数,其有效范围是 0~5,如果 nnn=0,则没有小数;如果 nnn>5 或缓冲区过小,无法容纳转换数值时,则用 ASCII 码空格键填充整个缓冲区。c 指定是用逗号(c=1)还是用点号(c=0)作为整数和小数部分的分隔符。图 7-20(b)给出了一个例子,其 ssss=1000,nnn=001,c=1。



(a)

	Out	Out +1	Out +2	Out +3	Out +4	Out +5	Out +6	Out +7
In=1234.5			1	2	3	4	,	5
In=-1.23					-	1	,	2
In=5.67						5	,	7
In=-12345.1	-	1	2	3	4	5	,	1

(b)

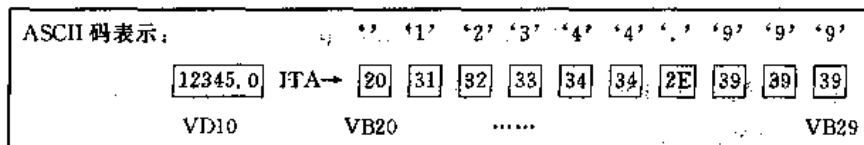
图 7-20 LAD 及 STL 格式、RTA 指令的 FMT 操作数格式及举例

除了 ITA 指令的 4 条规则外,RTA 指令输出缓冲区的格式还要符合下面的规则:

- (1) 小数部分的位数如果大于 nnn 指定的位数,则进行四舍五入,去掉多余的小数位。
- (2) 缓冲区的字节数应大于 3,且要大于小数部分的位数。

**【例 7-24】** 执行程序:RTA VD10,VB20,16#A3

16#A3 表示 OUT 的大小为 10 个字节,用点号作小数点,保留 3 位小数。在本例给定的输入条件下,则经过 RTA 后,结果如下:



注意:转换后的结果应为 12345.000,但因为转换精度的影响,有时会有误差,所以实际结果是 12344.999。大家可以进行实验验证。

### 7.4.5 字符串转换指令

字符串转换指令在 CPU V1.21 和 Micro/win32 V3.2 版本以上才可使用。

字符串是指全部合法的 ASCII 码字符串,这一点和上一节中的 ASCII 码范围不同。

#### 1. 数值转换为字符串

(1) 整数转换为字符串指令(Convert Integer to String)

ITS 指令的 LAD 及 STL 格式如图 7-21(a)所示。它和 ITA 指令基本上是一样的,惟一的区别是它转换结果放在从 OUT 开始的 9 个连续字节中,(OUT+0)字节中的值为字符串的长度。

(2) 双整数转换为字符串指令(Convert Double Integer to String)

DTS 指令的 LAD 及 STL 格式如图 7-21(b)所示。它和 DTA 指令基本上是一样的,惟一的区别是它的转换结果放在从 OUT 开始的 13 个连续字节中,(OUT+0)字节中的值为字符串的长度。

(3) 实数转换为字符串指令(Convert Real to String)

RTS 指令的 LAD 及 STL 格式如图 7-21(c)所示。它和 RTA 指令基本上是一样的,惟一区别是它的输出数据类型为字符串型字节,它的转换结果存放单元的第一个字节(OUT+0)中的值为字符串的长度,所以它的转换结果存放单元是从 OUT 开始的 ssss+1 个连续字节。

#### 2. 字符串转换为数值

字符串转换为数值包括 3 条指令:字符串转整数(Convert Substring to Integer)、字符串转双整数(Convert Substring to Double Integer)和字符串转实数(Convert Substring to Real)。

指令格式:LAD 及 STL 格式如图 7-21(d)、(e)和(f)所示。

功能描述:这 3 条指令将一个字符串 IN,从偏移量 INDX 开始,分别转换为整数、双整数和实数值,结果存放在 OUT 中。

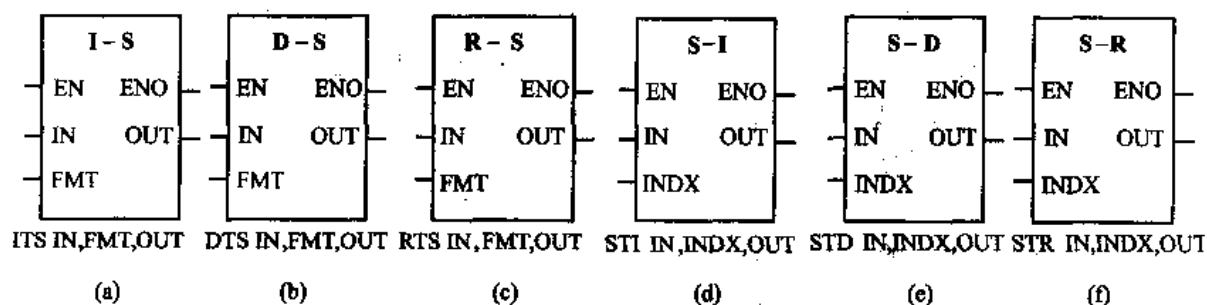


图 7-21 字符串转换指令

数据类型:这 3 条指令的 IN 均为字符串型字节,INDX 均为字节;STI 的 OUT 为 INT 型,STD 的 OUT 为 DINT 型,STR 的 OUT 为 REAL 型。

使用说明:

(1) STI 和 STD 将字符串转换为以下格式:【空格】【+或-】【数字 0~9】。STR 将字符串转换为以下格式:【空格】【+或-】【数字 0~9】、或,【数字 0~9】。

(2) INDX 的值通常设置为 1, 它表示从第一个字符开始转换。INDX 也可以设置为其他值, 从字符串的不同位置进行转换, 这可以被用于字符串中包含非数值字符的情况。例如输入字符串为“Temperature: 77.8”, 若 INDX 设置为 13, 则可以跳过字符串开头的“Temperature: ”。

(3) STR 指令不能用于转换以科学计数法或以指数形式表示的实数的字符串。指令不会产生溢出错误(SM1.1), 但是它会先将字符串转换到指数之前, 然后停止转换。例如: 字符串“1.234E6”转换为实数值为 1.234, 但不会有错误提示。

(4) 非法字符是指任意非数字(0~9)字符。在转换时, 当到达字符串的结尾或第一个非法字符时, 转换指令结束。

(5) 当转换产生的数值过大或过小以致使输出值无法表示时, 溢出标志(SM1.1)会置位。例如使用 STI 时, 若输入字符串产生的数值大于 32 767 或者小于 -32 768 时, SM1.1 就会置位。

(6) 当输入字符串中不包含可以转换的合法数值时, SM1.1 也会置位。例如字符串为空串或者为诸如“A123”等。

#### 【例 7-25】 执行程序:

```
LD    I0.0
STI   VB0,8,VW100
STD   VB0,8,VD120
STR   VB0,8,VD130
```

本例给定的输入条件 VB0 为:

VB0										VB13				
13	'w'	'e'	'i'	'g'	'h'	't'	' '	'5'	'8'	'.'	'8'	'k'	'g'	

当 I0.0 有效时, 程序执行结果如下:

VW100(整数) = 58;

VD120(双整数) = 58;

VD130(实数) = 58.8。

## 7.5 字符串指令

字符串指令在处理人机界面设计和数据转换时非常有用, 这是新版本的 PLC 才有的指令。

### 1. 字符串长度指令(String Length)

指令格式: LAD 及 STL 格式如图 7-22(a)所示。

功能描述: 把 IN 中指定的字符串的长度值送到 OUT 中。

数据类型: IN 为字符串型字节, OUT 为字节。

### 2. 字符串复制指令(Copy String)

指令格式: LAD 及 STL 格式如图 7-22(b)所示。

功能描述: 把 IN 中指定的字符串复制到 OUT 中。

数据类型:IN 和 OUT 均为字符串型字节。

### 3. 字符串连接指令 (Concatenate String)

指令格式:LAD 及 STL 格式如图 7-22(c)所示。

功能描述:把 IN 中指定的字符串连接到 OUT 中指定的字符串的后面。

数据类型:IN 和 OUT 均为字符串型字节。

### 4. 从字符串中复制字符串指令 (Copy Substring From String)

指令格式:LAD 及 STL 格式如图 7-22(d)所示。

功能描述:从 INDX 指定的字符开始,把 IN 中存储的字符串中的 N 个字符复制到 OUT 中。

数据类型:IN 和 OUT 均为字符串型字节,INDX 和 N 均为字节。

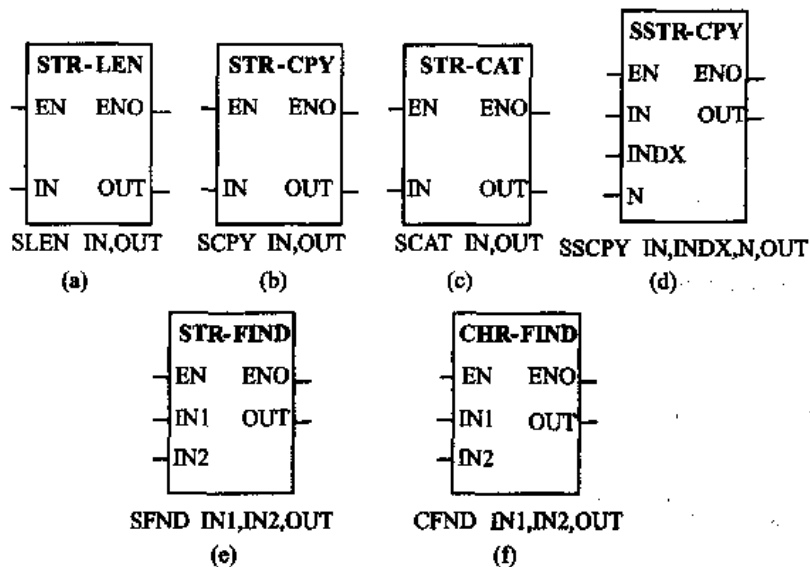


图 7-22 字符串指令

#### 【例 7-26】 执行程序:

LD I0.0

EU

SCAT VB20,VB0

SCPY VB0,VB100

SLEN VB100,AC0

SSCPY VB0,7,5,VB30

本例给定的输入条件 VB0 和 VB20 为:

VB0						VB6					
6	'H'	'E'	'L'	'L'	'O'	' '					

VB20						VB25					
5	'W'	'O'	'R'	'L'	'D'						

当 I0.0 有效时,程序执行结果如下:

VB0						VB11					
11	'H'	'E'	'L'	'L'	'O'	' '	'W'	'O'	'R'	'L'	'D'

VB100						VB111					
11	'H'	'E'	'L'	'L'	'O'	' '	'W'	'O'	'R'	'L'	'D'

AC0: 11

VB30				VB35	
5	'W'	'O'	'R'	'L'	'D'

**5. 字符串搜索指令 (Find String Within String)**

指令格式: LAD 及 STL 格式如图 7-22(e)所示。

功能描述: 在 IN1 字符串中寻找 IN2 字符串。由 OUT 指定搜索的起始位置。如果找到了相匹配的字符串, 则 OUT 中会存入这段字符串中首个字符的位置; 如果没有找到, OUT 被清零。

数据类型: IN1 和 IN2 均为字符串型字节, OUT 为字节。

**6. 字符搜索指令 (Find First Character Within String)**

指令格式: LAD 及 STL 格式如图 7-22(f)所示。

功能描述: 在 IN1 字符串中寻找 IN2 字符串中的任意字符。由 OUT 指定搜索的起始位置。如果找到了相匹配的字符, 则 OUT 中会存入相匹配的首个字符的位置; 如果没有找到, OUT 被清零。

数据类型: IN1 和 IN2 均为字符串型字节, OUT 为字节。

**【例 7-27】 执行程序:**

```

LD      I0.0
EU
MOVB    1, AC0
MOVB    1, AC1
SFND    VB0, VB10, AC0
CFND    VB0, VB30, AC1
STR      VB0, AC1, VD100

```

本例给定的输入条件 VB0、VB10 和 VB30 为:

VB0														VB14													
14	'T'	'e'	'm'	'p'	' '	' '	'9'	'8'	' '	'6'	'F'	' '	'O'	'K'													

VB10			VB12			VB30			VB42													
2	'O'	'K'				12	'1'	'2'	'3'	'4'	'5'	'6'	'7'	'8'	'9'	'0'	'+'	'-'				

当 I0.0 有效时, 程序执行结果如下:

AC0 = 13;

AC1 = 7;

VD100 = 98.6。

## 7.6 时钟指令

利用时钟指令可以实现调用系统实时时钟或根据需要设定时钟,这对于实现控制系统的运行监视、运行记录以及所有和实时时间有关的控制等十分方便。时钟操作有两种:读实时时钟和设定实时时钟。

### 1. 读实时时钟指令 (Read Real-Time Clock)

指令格式:LAD 及 STL 格式如图 7-23(a)所示。

功能描述:系统读当前时间和日期,并把它装入一个 8 字节的缓冲区。操作数 T 用来指定 8 个字节缓冲区的起始地址。

数据类型:T 为字节。

### 2. 设定实时时钟指令 (Set Real-Time Clock)

指令格式:LAD 及 STL 格式如图 7-23(b)所示。

功能描述:系统将包含当前时间和日期的一个 8 字节的缓冲区装入 PLC 的时钟中去。操作数 T 用来指定 8 字节缓冲区的起始地址。

数据类型:T 为字节。

时钟缓冲区的格式如表 7-9 所列。

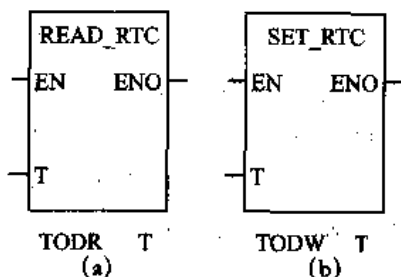


图 7-23 时钟指令格式

表 7-9 时钟缓冲区

字节	T	T+1	T+2	T+3	T+4	T+5	T+6	T+7
含义	年	月	日	小时	分钟	秒	0	星期
范围	00~99	01~12	01~31	00~23	00~59	00~59	0	00~07

注 意:

(1) 对于一个没有使用过时钟指令的 PLC,在使用时钟指令前,要在编程软件的“PLC”一栏中对 PLC 的时钟进行设定,然后才能开始使用时钟指令。时钟可以设定成和 PC 中的一样,也可用 TODW 指令自由设定,但必须先对时钟存储单元赋值后,才能使用 TODW 指令。

(2) 所有日期和时间的值均要用 BCD 码表示。如对于年而言,16#03 表示(20)03 年;对于小时而言,16#23 表示晚上 11 点。星期的表示范围是 0~7,1 表示星期一,依此类推,7 表示星期日,0 表示禁用星期。

(3) 系统不检查、不核实时钟各值的正确与否,所以必须确保输入的设定数据是正确的。例如,2 月 31 日虽为无效日期,但可以被系统接受。

(4) 不能同时主程序和中断程序中使用读写时钟指令,否则,将产生非致命错误,中断程序中的实时时钟指令将不被执行。

(5) 硬件时钟在 CPU224 以上的 PLC 中才有。

**【例 7-28】** 编写一段程序,要求可实现读写实时时钟,并以 BCD 码显示分钟。时钟缓冲区从 VB100 开始。该例程序如图 7-24 所示。

整个程序由主程序和子程序组成。主程序完成实时时钟的读取,并且进行分钟的显示。子程序完成时钟和日期的设置,可在需要的时候调用子程序,具体的时间可根据实际情况设置。日期和时间的设定数值也可以集中放到参数块中,从而简化程序设计。

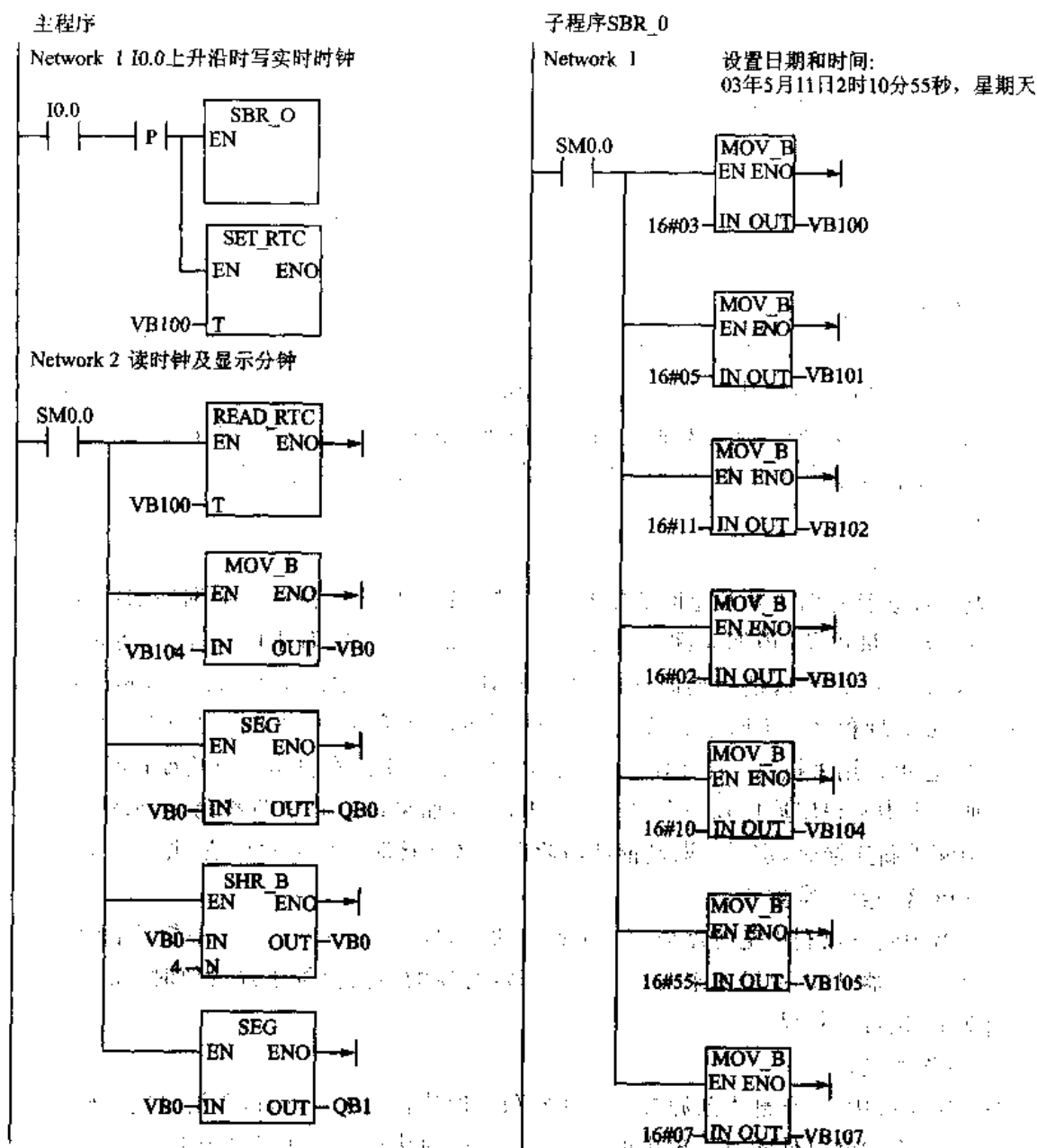


图 7-24 读写时钟程序

## 7.7 中 断

中断技术在处理复杂和特殊的控制任务时是必需的,它属于 PLC 的高级应用技术。中断是由设备或其他非预期的急需处理的事件引起的,它使系统暂时中断现在正在执行的程序,而转到中断服务程序去处理这些事件,处理完毕后再返回原程序执行。中断事件的发生具有随

机性,中断在可编程序控制器的实时处理、高速处理、通信和网络中非常重要。

### 7.7.1 几个基本概念

#### 1. 中断源及种类

中断源即中断事件发出中断请求的来源。S7-200 系列可编程序控制器具有最多可达 34 个中断源,每个中断源都分配一个编号加以识别,称为中断事件号。这些中断源大致分为三大类:通信中断、输入/输出中断和时基中断。

##### (1) 通信中断

可编程序控制器的通信口可由程序来控制,通信中的这种操作模式称为自由通信口模式。在这种模式下,用户可以通过编程来设置波特率、奇偶校验和通信协议等参数。

##### (2) 输入/输出中断

输入/输出中断包括外部输入中断、高速计数器中断和脉冲串输出中断。外部输入中断是系统利用 I0.0~I0.3 的上升沿或下降沿产生中断,这些输入点可用作连接某些一旦发生就必须引起注意的外部事件;高速计数器中断可以响应当前值等于预设值、计数方向改变、计数器外部复位等事件所引起的中断;脉冲串输出中断可以用来响应给定数量的脉冲输出完成所引起的中断。

##### (3) 时基中断

时基中断包括定时中断和定时器中断。定时中断可用来支持一个周期性的活动,周期时间以 1 ms 为计量单位,周期时间可以是 1~255 ms。对于定时中断 0,把周期时间值写入 SMB34;对于定时中断 1,把周期时间值写入 SMB35。每当达到定时时间值,相关定时器溢出,执行中断处理程序。定时中断可以用来以固定的时间间隔作为采样周期来对模拟量输入进行采样,也可以用来执行一个 PID 控制回路,另外定时中断在自由口通信编程时非常有用。

当把某个中断程序连接到一个定时中断事件上,如果该定时中断被允许,那就开始计时。当定时中断重新连接时,定时中断功能能清除前一次连接时的任何累计值,并用新值重新开始计时。理解这一点非常重要。

定时器中断可以利用定时器来对一个指定的时间段产生中断。这类中断只能使用分辨率为 1 ms 的定时器 T82 和 T96 来实现。当所用定时器的当前值等于预设值时,在主机正常的定时刷新中,执行中断程序。

#### 2. 中断优先级

在中断系统中,将全部中断源按中断性质和处理的轻重缓急进行,并给以优先权。所谓优先权,是指多个中断事件同时发出中断请求时,CPU 对中断响应的优先次序。中断优先级由高到低依次是:通信中断、输入/输出中断、时基中断。每种中断中的不同中断事件又有不同的优先权。所有中断事件及优先级如表 7-10 所列。

表 7-10 中断事件及优先级

组优先级	组内类型	中断事件号	中断事件描述	组内优先级
通信中断 (最高级)	通信口 0	8	通信口 0:接收字符	0
		9	通信口 0:发送完成	0
		23	通信口 0:接收信息完成	0
	通信口 1	24	通信口 1:接收信息完成	1
		25	通信口 1:接收字符	1
		26	通信口 1:发送完成	1
输入/输出中断 (次高级)	脉冲输出	19	PTO0 脉冲串输出完成中断	0
		20	PTO1 脉冲串输出完成中断	1
	外部输入	0	I0.0 上升沿中断	2
		2	I0.1 上升沿中断	3
		4	I0.2 上升沿中断	4
		6	I0.3 上升沿中断	5
		1	I0.0 下降沿中断	6
		3	I0.1 下降沿中断	7
		5	I0.2 下降沿中断	8
		7	I0.3 下降沿中断	9
	高速计数器	12	HSC0 当前值等于预设值中断	10
		27	HSC0 输入方向中断	11
		28	HSC0 外部复位中断	12
		13	HSC1 当前值等于预设值中断	13
		14	HSC1 输入方向改变中断	14
		15	HSC1 外部复位中断	15
		16	HSC2 当前值等于预设值中断	16
		17	HSC2 输入方向改变中断	17
		18	HSC2 外部复位中断	18
		32	HSC3 当前值等于预设值中断	19
		29	HSC4 当前值等于预设值中断	20
		30	HSC4 输入方向改变中断	21
		31	HSC4 外部复位中断	22
		33	HSC5 当前值等于预设值中断	23
时基中断 (最低级)	定时	10	定时中断 0	0
		11	定时中断 1	1
	定时器	21	T32 当前值等于预设值中断	2
		22	T96 当前值等于预设值中断	3

在 PLC 中,CPU 按先来先服务的原则响应中断请求,一个中断程序一旦执行,就一直执行到结束为止,不会被其他甚至更高优先级的中断程序所打断。在任何时刻,CPU 只执行一

个中断程序。中断程序执行中,新出现的中断请求按优先级排队等候处理。中断队列能保存的最大中断个数有限,如果超过队列容量,则会产生溢出,某些特殊标志存储器位被置位。中断队列、溢出位及队列容量如表 7-11 所列。

表 7-11 各主机的中断队列最大中断数

中断队列种类	中断队列溢出标志位	CPU221	CPU222	CPU224	CPU226/CPU226XM
通信中断队列	SM4.0	4 个	4 个	4 个	8 个
I/O 中断队列	SM4.1	16 个	16 个	16 个	16 个
时基中断队列	SM4.2	8 个	8 个	8 个	8 个

### 7.7.2 中断指令

中断调用即调用中断程序,使系统对特殊的内部事件做出响应。系统响应中断时自动保存逻辑堆栈、累加器和某些特殊标志存储器位,即保护现场。中断处理完成时,又自动恢复这些单元原来的状态,即恢复现场。

#### 1. 中断连接指令(Attach Interrupt)

指令格式:LAD 及 STL 格式如图 7-25(a)所示。

功能描述:将一个中断事件和一个中断程序建立联系,并允许这一中断事件。

数据类型:中断程序号 INT 和中断事件号 EVNT 均为字节型常数。

不同 CPU 主机的 EVNT 取值范围不同,如表 7-12 所列。

表 7-12 EVNT 取值范围

CPU 型号	CPU221	CPU222	CPU224	CPU226/CPU226XM
EVNT 取值范围	0~12,19~23,27~33	0~12,19~23,27~33	0~23,27~33	0~33

#### 2. 中断分离指令(Detach Interrupt)

指令格式:LAD 及 STL 格式如图 7-25(b)所示。

功能描述:切断一个中断事件和所有程序的联系,使该事件的中断回到不激活或无效状态,因而禁止了该中断事件。本指令主要用于对某一事件单独禁止中断。

数据类型:中断事件号 EVNT 为字节型常数。

#### 3. 开中断(Enable Interrupt)及关中断(Disable Interrupt)指令

指令格式:LAD 及 STL 格式如图 7-25(c)所示。

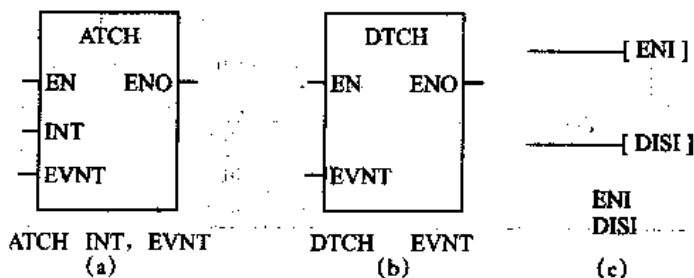


图 7-25 中断指令格式

ENI:开中断指令(中断允许指令)。全局开放(或允许)所有被连接的中断事件。梯形图中以线圈形式编程,无操作数。

DISI:关中断指令(中断禁止指令)。全局关闭(或禁止)所有被连接的中断事件。梯形图中以线圈形式编程,无操作数。

注 意:

(1) 多个事件可以调用同一个中断程序,但同一个中断事件不能同时指定多个中断服务程序。否则,在中断允许时,若某个中断事件发生,系统默认只执行为该事件指定的最后一个中断程序。

(2) 当系统由其他模式切换到 RUN 模式时,就自动关闭了所有的中断。

(3) 可以通过编程,在 RUN 模式下,用使能输入执行 ENI 指令来开放所有的中断,以实现中断事件的处理。全局关中断指令 DISI 使所有中断程序不能被激活,但允许发生的中断事件等候,直到使用开中断指令重新允许中断。

【例 7-29】编写一段程序完成一个数据采集任务,要求每 200 ms 采集一个数。本程序如图 7-26 所示。

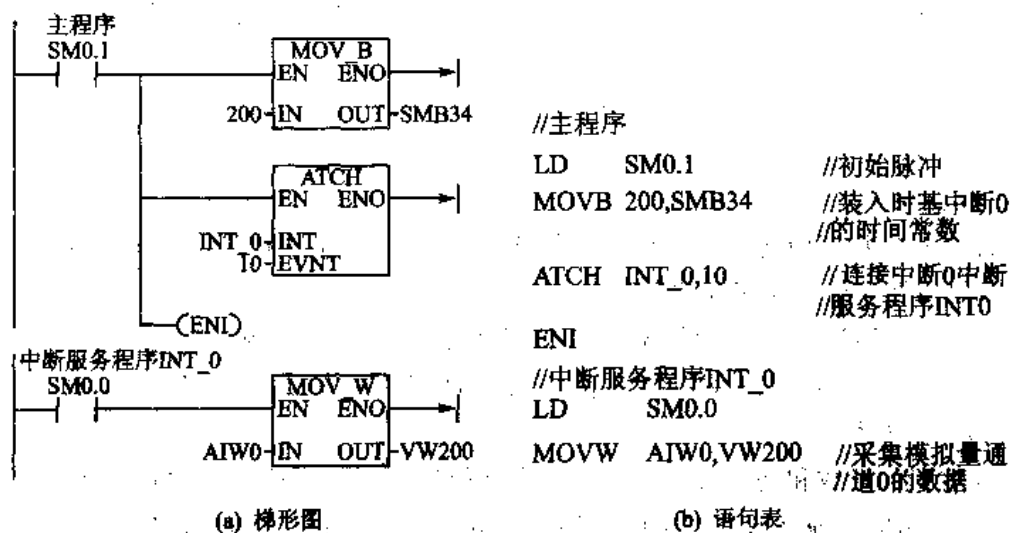


图 7-26 中断程序使用举例

### 7.7.3 中断程序

中断程序也称中断服务程序,是用户处理中断事件而事先编制的程序。编程时可以用中断程序入口处的中断程序标号来识别每个中断程序。

#### 1. 构成

中断程序必须由三部分构成:中断程序标号、中断程序指令和无条件返回指令。中断程序标号,即中断程序的名称,它在建立中断程序时生成;中断程序指令是中断程序的实际有效部分,对中断事件的处理就是由这些指令组合完成的,在中断程序中可以调用嵌套子程序;中断返回指令用来退出中断程序回到主程序。它有两条返回指令,一是无条件中断返回指令 RETI,程序编译时由软件自动在程序结尾加上 RETI 指令,而不必由编程人员手工输入。另一条是条件返回指令 CRET1,在中断程序内部用它可以提前退出中断程序。

## 2. 要 求

中断程序的编写要求是:短小精悍、执行时间短。用户应最大限度地优化中断程序,否则意外条件可能会导致由主程序控制的设备出现异常操作。

## 3. 编制方法

用编程软件,在“编辑”菜单下的“插入”中选择“中断”,则自动生成一个新的中断程序编号,进入该中断程序的编辑区,在此即可编写中断程序。

注 意:

(1) 在执行中断程序和中断程序调用的子程序时共用累加器和逻辑堆栈;

(2) 在中断程序中不能使用 DISI、ENI、HDEF、LSCR 和 END 指令。

中断程序应用实例可参见高速指令和 PID 指令部分。

# 7.8 高速计数器指令

一般来说,高速计数器 HSC 和编码器配合使用,在现代自动控制中实现精确定位和测量长度。它可用来累计比可编程序控制器的扫描频率高得多的脉冲输入,利用其产生的中断事件完成预定的操作。

## 7.8.1 高速计数器介绍

### 1. 数量及编号

高速计数器在程序中使用的地址编号用 HCn 来表示(在非程序中一般用 HSCn 表示),HC 表示编程元件名称为高速计数器,n 为编号。

不同型号的 PLC 主机,高速计数器的数量也不同,CPU221 和 CPU222 有 4 个,它们是 HC0 和 HC3~HC5;CPU224、CPU226 和 CPU226XM 有 6 个,它们是 HC0~HC5。

### 2. 中断事件类型

高速计数器的计数和动作可采用中断方式进行控制,与 CPU 的扫描周期关系不大,各种型号的 PLC 可用的高速计数器的中断事件大致分为 3 类:当前值等于预设值中断、输入方向改变中断和外部复位中断。所有高速计数器都支持当前值等于预设值中断。每个高速计数器的 3 种中断的优先级由高到低,不同高速计数器之间的优先级又按编号顺序由高到低。具体对应关系如表 7-10 所列。

### 3. 工作模式及输入点

高速计数器的使用共有四种基本类型:带有内部方向控制的单向计数器;带有外部方向控制的单向计数器;带有两个时钟输入的双向计数器和 A/B 相正交计数器。它的输入信号类型有:无复位或启动输入,有复位无启动输入或者既有启动又有复位输入。

每种高速计数器有多种工作模式,以完成不同的功能,高速计数器的工作模式与中断事件有密切关系。在使用一个高速计数器时,首先要使用 HDEF 指令给计数器设定一种工作模式。每一种 HSCn 的工作模式的数量也不同,HSC1 和 HSC2 最多可达 12 种,而 HSC5 只有一种工作模式。

选用某个高速计数器在某种工作模式下工作后,高速计数器所使用的输入端不是任意选择的,必须按系统指定的输入点输入信号。例如,如果 HSC0 在模式 4 下工作,就必须用 I0.0

为时钟输入端, I0.1 为增减方向输入端, I0.2 为外部复位输入端。

高速计数器输入点、输入/输出中断输入点都包括在一般数字量输入点编号范围内。同一个输入点只能用做一种功能, 如果程序使用了高速计数器, 则高速计数器的这种工作模式下指定的输入点只能被高速计数器使用。只有高速计数器不用的输入点才可以作为输入/输出中断或一般数字量输入点使用。例如, HSC0 在模式 0 下工作, 只用 I0.0 作时钟输入, 不使用 I0.1 和 I0.2, 则这两个输入端可作为它用。

高速计数器的输入点和工作模式如表 7-13 所列。

表 7-13 高速计数器的输入点和工作模式

模 式	描 述	输入点			
	HSC0	I0.0	I0.1	I0.2	
	HSC1	I0.6	I0.7	I0.2	I1.1
	HSC2	I1.2	I1.3	I1.1	I1.2
	HSC3	I0.1			
	HSC4	I0.3	I0.4	I0.5	
	HSC5	I0.4			
0	带有内部方向控制的 单相计数器	时钟			
1		时钟		复位	
2		时钟		复位	启动
3	带有外部方向控制的 单相计数器	时钟	方向		
4		时钟	方向	复位	
5		时钟	方向	复位	启动
6	带有增减计数时钟的 双相计数器	增时钟	减时钟		
7		增时钟	减时钟	复位	
8		增时钟	减时钟	复位	启动
9	A/B 相正交计数器	时钟 A	时钟 B		
10		时钟 A	时钟 B	复位	
11		时钟 A	时钟 B	复位	启动

对高速计数器的复位和启动有如下规定:

- (1) 当激活复位输入端时, 计数器清除当前值并一直保持到复位端失效。
- (2) 当激活启动输入端时, 计数器计数; 当启动端失效时, 计数器的当前值保持为常数, 并且忽略时钟事件。
- (3) 如果在启动输入端无效的同时, 复位信号被激活, 则忽略复位信号, 当前值保持不变; 如果在复位信号被激活的同时, 启动输入端被激活, 则当前值被清除。

### 7.8.2 高速计数器指令

#### 1. 定义高速计数器指令 (High-Speed Counter Definition)

指令格式: LAD 及 STL 格式如图 7-27(a) 所示。

功能描述: 为指定的高速计数器分配一种工作模式, 即用来建立高速计数器与工作模式之间的联系。每个高速计数器使用之前必须使用 HDEF 指令, 而且只能使用一次。

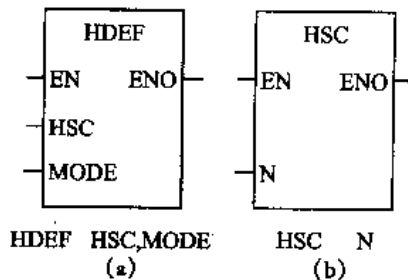
数据类型: HSC 表示高速计数器编号, 为 0~5 的常数, 属字节型; MODE 表示工作模式, 为 0~11 的常数, 属字节型。

#### 2. 高速计数器指令 (High-Speed Counter)

指令格式: LAD 及 STL 格式如图 7-27(b) 所示。

功能描述: 根据高速计数器特殊存储器位的状态, 并按照 HDEF 指令指定的工作模式, 设置高速计数器并控制其工作。

数据类型: N 表示高速计数器编号, 为 0~5 的常数, 属字型。



### 7.8.3 高速计数器的使用方法

每个高速计数器都有固定的特殊存储器与之相配合, 完成高速计数功能。具体对应关系如表 7-14 所列。

图 7-27 高速计数器指令格式

表 7-14 HSC 使用的特殊标志寄存器

高速计数器编号	状态字节	控制字节	当前值(双字)	预设值(双字)
HSC0	SMB36	SMB37	SMD38	SMD42
HSC1	SMB46	SMB47	SMD48	SMD52
HSC2	SMB56	SMB57	SMD58	SMD62
HSC3	SMB136	SMB137	SMD138	SMD142
HSC4	SMB146	SMB147	SMD148	SMD152
HSC5	SMB156	SMB157	SMD158	SMD162

#### 1. 状态字节

每个高速计数器都有一个状态字节, 程序运行时根据运行状况自动使某些位置位, 可以通过程序来读取相关位的状态, 用做判断条件实现相应的操作。状态字节中各状态位的功能如表 7-15 所列。

表 7-15 状态字节

状态位	SM×6.0~SM×6.4	SM×6.5	SM×6.6	SM×6.7
功能描述	不用	当前计数方向 0 减, 1 增	当前值=预设值 0 不等, 1 等	当前值>预设值 0<=, 1>

#### 2. 控制字节

每个高速计数器都对应一个控制字节。用户可以根据要求来设置控制字节中各控制位的

状态,如复位与启动输入信号的有效状态、计数速率、计数方向、允许更新双字值和允许执行 HSC 指令等,实现对高速计数器的控制。控制字节中各控制位的功能如表 7-16 所列。

表 7-16 控制位含义

控制位	功能描述	适用的计数器 HCn
SM××7.0	复位有效电平控制位:0,高电位有效;1,低电位有效	0,1,2,4
SM××7.1	启动有效电平控制位:0,高电位有效;1,低电位有效	1,2
SM××7.2	正交计数速率选择位:0,4x 计数速率;1,1x 计数速率	0,1,2,4
SM××7.3	计数方向控制位:0,减计数;1,增计数	0,1,2,3,4,5
SM××7.4	写计数方向允许控制:0,不更新;1,更新计数方向	0,1,2,3,4,5
SM××7.5	写入预设值允许控制:0,不更新;1,更新预设值	0,1,2,3,4,5
SM××7.6	写入当前值允许控制:0,不更新;1,更新当前值	0,1,2,3,4,5
SM××7.7	HSC 指令执行允许控制:0,禁止 HSC;1,允许 HSC	0,1,2,3,4,5

表中的前 3 位(0、1 和 2 位)只有在 HDEF 指令执行时进行设置,在程序中其他位置不能更改(默认值为:启动和复位为高电位有效,正交计数速率为 4x,即 4 倍输入时钟频率)。第 3 位和第 4 位可以在工作模式 0、1 和 2 下直接更改,以单独改变计数方向。后 3 位可以在任何模式下并在程序中更改,以单独改变计数器的当前值、预设值或对 HSC 禁止计数。

### 3. 使用高速计数器及选择工作模式步骤

选择高速计数器及工作模式包括两方面工作:根据使用的主机型号和控制要求,一是选用高速计数器;二是选择该高速计数器的工作模式。

#### (1) 选择高速计数器

例如,要对一高速脉冲信号进行增/减计数,计数当前值达到 1 200 产生中断,计数方向用一个外部信号控制,所用的主机型号为 CPU224。

**分析** 本控制要求是带外部方向控制的单相增/减计数,因此可用的高速计数器可以是 HSC0、HSC1、HSC2 或 HSC4 中的任何一个。如果确定为 HSC0,由于不要求外部复位,所以应选择工作模式 3。同时也确定了各个输入点:I0.0 为计数脉冲的时钟输入;I0.1 为外部方向控制(I0.1=0 时为减计数,I0.1=1 时为增计数)。

#### (2) 设置控制字节

在选择用 HSC0 的工作模式 3 之后,对应的控制字节为 SMB37。如果向 SMB37 写入 2#11111000,即 16#F8,则对 HSC0 的功能设置为:复位与启动输入信号都是高电位有效、4 倍计数频率、计数方向为增计数、允许更新双字值和允许执行 HSC 指令。

#### (3) 执行 HDEF 指令

执行 HDEF 指令时,HSC 的输入值为 0,MODE 的输入值为 3,指令如下:

```
HDEF    0,3
```

#### (4) 设定当前值和预设值

每个高速计数器都对应一个双字长的当前值和一个双字长的预设值。两者都是有符号整数。当前值随计数脉冲的输入而不断变化,运行时当前值可以由程序直接读取 HCn 得到。

本例中,选用 HSC0,所以对应的当前值和预设值分别存放到 SMD38 和 SMD42 中。如果

希望从 0 开始计数,计数值达到 1 200 时产生中断,则可以用双字传送指令分别将 0 和 120 装入 SMD38 和 SMD42 中。

(5) 设置中断事件并全局开中断

高速计数器利用中断方式对高速事件进行精确控制。

本例中,用 HSC0 进行计数,要求在当前值等于预设值时产生中断。因此,中断事件是当前值等于预设值,中断事件号为 12。用中断调用 ATCH 指令将中断事件号 12 和中断程序(假设中断子程序编号为 INT\_0)连接起来,并全局开中断。指令如下:

```
ATCH    INT_0,12
ENI
```

在 INT\_0 程序中,可完成 HSC0 当前值等于设定值时计划要做的工作。

(6) 执行 HSC 指令

以上设置完成并用指令实现之后,即可用 HSC 指令对高速计数器编程进行计数。本例中指令如下:

```
HSC    0
```

以上 6 步是对高速计数器的初始化,该过程可以用主程序中的程序段来实现,也可以用子程序来实现。高速计数器在投入运行之前,必须要执行一次初始化程序段或初始化子程序。

【例 7-30】 高速计数器应用实例。采用测频的方法测量电机的转速。

**分析** 所谓用测频法测量电机的转速是指在单位时间内采集编码器脉冲的个数,因此可以选用高速计数器对转速脉冲信号进行计数,同时用时基来完成定时。知道了单位时间内的脉冲个数,再经过一系列的计算就可以得知电机的转速。下面的程序只是整个程序中有关 HCS 的部分。

设计步骤:

① 选择高速计数器 HSC0,并确定工作方式 0。采用初始化子程序,用初始化脉冲 SM0.1 调用子程序。

② 令 SMB37=16#F8。其功能为:计数方向为增;允许更新计数方向;允许写入新当前值;允许写入新设定值;允许执行 HSC 指令。

③ 执行 HDEF 指令;输入端 HSC 为 0,MODE 为 0。

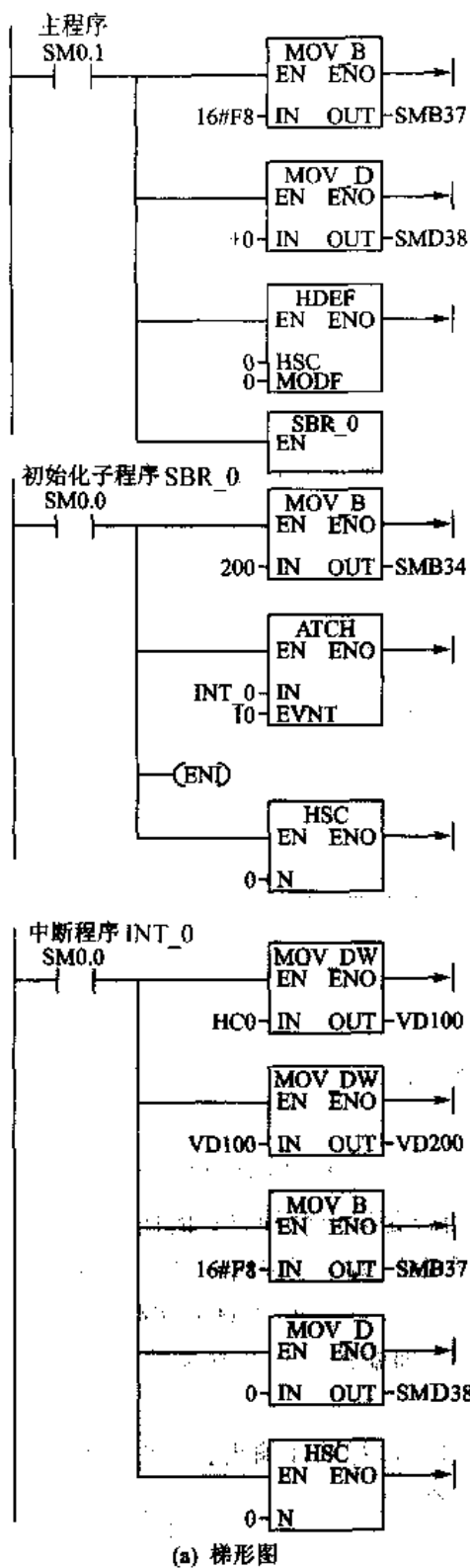
④ 装入当前值,令 SMD38=0。

⑤ 装入时基定时设定值,令 SMB34 = 200。

⑥ 执行中断连接 ATCH 指令,中断程序为 INT\_0,EVENT 为 10。执行中断允许指令 ENI,重新启动时基定时器,清除高速计数器的当前值。

⑦ 执行指令 HSC,对高速计数器编程并投入运行,输入值 IN 为 0。

主程序、初始化子程序和中断程序的梯形如图 7-28 所示。



主程序

```
LD SM0.1 //初始脉冲
MOVB 16#F8,SMB37
//F8H 送高速计数器0控制字节单元
MOVD +0,SMD38
//清高速计数器0的当前值单元
HDEF 0,0
//定义高速计数器0为工作方式0
CALL SBR_0 //调时基初始化子程序
```

时基初始化子程序 SBR\_0

```
LD SM0.0
MOVB 200,SMB34
//时基中断0定时时间常数单元送200
//即定时200 ms
ATCH INT_0,10
ENI //全局开中断
HSC 0 //启动高速计数器0
```

中断服务程序 INT\_0

```
LD SM0.0
MOVD HC0,VD100
//读高速计数器0的计数值到VD100
MOVB VD100,VD200
//数值送数据处理单元
MOVB 16#F8,SMB37
//重新初始化高速计数器0
MOVB 0,SMD38
//清高速计数器0的当前单元
HSC 0
//启动高速计数器
```

(b) 语句表

图 7-28 高速计数器使用举例程序

## 7.9 高速脉冲输出指令

高速脉冲输出功能是指在可编程序控制器的某些输出端产生高速脉冲,用来驱动负载实现精确控制。这在运动控制中具有广泛应用。使用高速脉冲输出功能时,PLC 主机应选用晶体管输出型,以满足高速输出的频率要求。

### 7.9.1 几个基本概念

#### 1. 高速脉冲输出的方式

高速脉冲输出有高速脉冲串输出 PTO 和宽度可调脉冲输出 PWM 两种方式。

PTO 可以输出一串脉冲(占空比 50 %),用户可以控制脉冲的周期和个数,如图 7-29(a)所示。PWM 可以输出一串占空比可调的脉冲,用户可以控制脉冲的周期和脉宽,如图 7-29(b)所示。

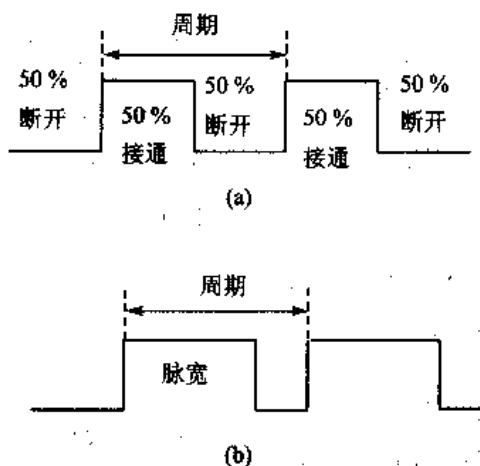


图 7-29 高速脉冲的输出方式

#### 2. 输出端子的确定

每种 PLC 主机最多可提供 2 个高速脉冲输出端。高速脉冲的输出端不是任意选择的,必须按系统指定的输出点 Q0.0 和 Q0.1 来选择,也可以是以上两种方式的任意组合。

高速脉冲输出点包括在一般数字量输出映像寄存器编号范围内。同一个输出点只能用做一种功能,如果 Q0.0 和 Q0.1 在程序执行时用做高速脉冲输出,则只能被高速脉冲输出使用,其通用功能被自动禁止,任何输出刷新、输出强制、立即输出等指令都无效。只有高速脉冲输出不用的输出点才可能做普通数字量输出点使用。

在 Q0.0 和 Q0.1 编程时用做高速脉冲输出,但未执行脉冲输出指令时,可以用普通位操作指令设置这两个输出位,以控制高速脉冲的起始和终止电位。

### 7.9.2 高速脉冲指令及特殊寄存器

#### 1. 脉冲输出指令(Pulse Output)

指令格式:LAD 和 STL 格式如图 7-30 所示。

功能描述:检测用程序设置的特殊存储器位,激活由控制位定义的脉冲操作,从 Q0.0 或 Q0.1 输出高速脉冲。高速脉

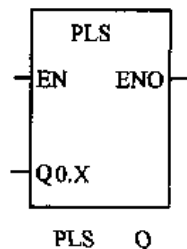


图 7-30 脉冲输出指令格式

冲串输出 PTO 和宽度可调脉冲输出 PWM 都由 PLS 指令激活输出。

数据类型:数据输入 Q 属字型,必须是 0 或 1 的常数。

## 2. 特殊标志寄存器

每个高速脉冲发生器对应一定数量的特殊寄存器,这些寄存器包括控制字节寄存器、状态字节寄存器和参数数值寄存器。它们用以控制高速脉冲的输出形式,反映输出状态和参数值。各寄存器的功能如表 7-17 所列。

表 7-17 相关寄存器功能表

Q0.0 的寄存器	Q0.1 的寄存器	名称及功能描述
SMB66	SMB76	状态字节,在 PTO 方式下,跟踪脉冲串的输出状态
SMB67	SMB77	控制字节,控制 PTO/PWM 脉冲输出的基本功能
SMW68	SMW78	周期值,属字型,PTO/PWM 的周期值,范围 2~65 535
SMW70	SMW80	脉宽值,属字型,PWM 的脉宽值,范围 0~65 535
SMD72	SMD82	脉冲数,属双字型,PTO 的脉冲数,范围 1~4 294 967 295
SMB166	SMB176	段号,多段管线 PTO 进行中的段的编号
SMW168	SMW178	多段管线 PTO 包络表起始字节的地址

### (1) 状态字节

用于 PTO 方式。每个高速脉冲输出都有一个状态字节,程序运行时根据运行状态使某位自动置位。可以通过程序来读取相关位的状态,用此状态作为判断条件实现相应的操作。状态字节中各状态位的功能如表 7-18 所列。

表 7-18 状态字节表

状态位	SM×6.0~SM×6.3	SM×6.4	SM×6.5	SM×6.6	SM×6.7
功能描述	不用	PTO 包络因增量计算错误终止 0:无错;1:终止	PTO 包络因用户命令终止 0:无错;1:终止	PTO 管线溢出 0:无溢出 1:溢出	PTO 空闲 0:执行中 1:空闲

### (2) 控制字节

每个高速脉冲输出都对应一个控制字节,通过对控制字节指定位的编程,设置字节中各控制位,如脉冲输出允许、PTO/PWM 模式选择、PTO 单段/多段选择、更新方式、时间基准和允许更新等。控制字节中各控制位的功能如表 7-19 所列。

表 7-19 控制位含义

Q0.0 控制位	Q0.1 控制位	功能描述
SM67.0	SM77.0	PTO/PWM 更新周期值;0,不更新;1,允许更新
SM67.1	SM77.1	PWM 更新脉冲宽度值;0,不更新;1,允许更新
SM67.2	SM77.2	PTO 更新输出脉冲数;0,不更新;1,允许更新
SM67.3	SM77.3	PTO/PWM 时间基准选择;0,μs 单位时基;1,ms 单位时基
SM67.4	SM77.4	PWM 更新方式;0,异步更新;1,同步更新

续表 7-19

Q0.0 控制位	Q0.1 控制位	功 能 描 述
SM67.5	SM77.5	PTO 单/多段方式:0,单段管线;1,多段管线
SM67.6	SM77.6	PTO/PWM 模式选择:0,选用 PTO 模式;1,选用 PWM 模式
SM67.7	SM77.7	PTO/PWM 脉冲输出:0,禁止;1 允许

例如,如果用 Q0.0 作为高速脉冲输出,则对应的控制字节为 SMB67。如果向 SMB67 写入  $2\#10101000$ ,即  $16\#A8$ ,则对 HSC0 的功能设置为:允许脉冲输出,多段 PTO 脉冲串输出,时基为 ms,不允许更新周期值和脉冲数。

### 7.9.3 PTO 的使用

高速脉冲串输出 PTO,状态字节中的最高位用来指示脉冲串输出是否完成。在脉冲串输出完成的同时可以产生中断,因而可以调用中断程序完成指定操作。

#### 1. 周期和脉冲数

周期:单位可以是  $\mu s$  或 ms,为 16 位无符号数据,周期变化范围是  $50\sim 65\,535\mu s$  或  $2\sim 65\,535\text{ ms}$ 。如果编程时设定周期单位小于最小值,系统默认则按最小值(2 个时间单位)进行设置。

脉冲数:用双字无符号数表示,脉冲数取值范围是  $1\sim 4\,294\,967\,295$  之间。如果编程时指定脉冲数为 0,则系统默认脉冲数为 1 个。

#### 2. PTO 的种类

PTO 方式中,可输出多个脉冲串,并允许脉冲串排队,以形成管线。当前输出的脉冲串完成之后,立即输出新脉冲串,这保证了脉冲串顺序输出的连续性。根据管线的实现方式,将 PTO 分为两种:单段管线和多段管线。

##### (1) 单段管线

管线中只能存放一个脉冲串的控制参数(即入口),一旦启动了一个脉冲串进行输出时,就需要用指令立即为下一个脉冲串更新特殊寄存器,并再次执行脉冲串输出指令。当前脉冲串输出完成之后,自动输出下一个脉冲串。重复这一操作可以实现多个脉冲串的输出。

单段管线中的各脉冲段可以采用不同的时间基准。单段管线输出多个高速脉冲时,编程复杂,而且有时参数设置不当会造成脉冲串之间的不平滑转换。

##### (2) 多段管线

多段管线是指在变量 V 存储区建立一个包络表。包络表中存储各个脉冲串的参数,相当于有多个脉冲串的入口。多段管线可以用 PLS 指令启动,运行时,CPU 自动从包络表中按顺序读出每个脉冲串的参数进行输出。

编程时必须装入包络表的起始变量(V 存储区)的偏移地址,运行时只使用特殊存储区的控制字节和状态字节即可。包络表的首地址代表该包络表,它放在 SMW168 或 SMW178 中。PTO 当前进行中的段的编号放在 SMB166 或 SMB176 中。

包络表格式由包络段数和各段构成。整个包络表的段数(1~255)放在包络表首字节中(8 位),接下来的每段设定占用 8 个字节,包括:脉冲初始周期值(16 位)、周期增量值(16 位)和脉冲计数值(32 位)。以包络 3 段的包络表为例,若 VBn 为包络表起始字节地址,则包络表的结

构如表 7-20 所列。表中周期单位为 ms。

表 7-20 3 段包络表格式

字节偏移地址	名称	描述
VB <sub>n</sub>	段标号	段数, 为 1~255, 数 0 将产生非致命性错误, 不产生 PTO 输出
VW <sub>n+1</sub>	段 1	初始周期, 取值范围为 2~65535
VW <sub>n+3</sub>		每个脉冲的周期增量, 符号整数, 取值范围为 -32 768~+32 767
VD <sub>n+5</sub>		输出脉冲数(1~4 294 967 295)
VW <sub>n+9</sub>	段 2	初始周期, 取值范围为 2~65 535
VW <sub>n+11</sub>		每个脉冲的周期增量, 符号整数, 取值范围为 -32 678~+32 767
VD <sub>n+13</sub>		输出脉冲数(1~4 294 967 295)
VW <sub>n+17</sub>	段 3	初始周期, 取值范围为 2~65 535
VW <sub>n+19</sub>		每个脉冲的周期增量, 符号整数, 取值范围为 -32 768~+32 767
VD <sub>n+21</sub>		输出脉冲数(1~4 294 967 295)

多段管线编程非常简单, 而且具有按照周期增量区的数值自动增减周期的功能, 这在步进电机的加速和减速控制时非常方便。

多段管线使用时的局限性是在包络表中的所有脉冲串的周期必须采用同一个基准, 而且当多段管线执行时, 包络表的各段参数不能改变。

### 3. 中断事件类型

高速脉冲串输出可以采用中断方式进行控制, 各种型号的 PLC 可用的高速脉冲串输出中断事件有两个, 即中断事件 19 和 20, 参见表 7-10。

### 4. PTO 的使用

使用高速脉冲串输出时, 要按以下步骤进行:

(1) 确定脉冲发生器及工作模式 它包括两方面工作: 根据控制要求, 一是选用高速脉冲串输出端(发生器); 二是选择工作模式为 PTO, 并且确定多段或单段工作模式。如果要求有多个脉冲串连续输出, 通常采用多段管线。

(2) 设置控制字节 按控制要求将控制字节写入 SMB67 和 SMB77 的特殊寄存器。

(3) 写入周期值、周期增量值和脉冲数 如果是单段脉冲, 对以上各值分别设置; 如果是多段脉冲, 则需要建立多段脉冲的包络表, 并对各段参数分别设置。

(4) 装入包络表的首地址 本步为可选, 只在多段脉冲输出中需要。

(5) 设置中断事件并全局开中断 高速脉冲串输出 PTO 可利用中断方式对高速事件进行精确控制。中断事件是高速脉冲输出完成, 中断事件号为 19 或 20。用中断调用 ATCH 指令把中断事件号 19 或 20 与中断子程序(假设中断子程序编号为 INT\_0)连接起来, 并全局开中断。程序如下:

```
ATCH INT_0,19
```

```
ENI
```

注意: 必须编写中断程序 INT\_0 与之相对应。

(6) 执行 PLS 指令 经以上设置并执行指令后, 即可用 PLS 指令启动高速脉冲串, 并由 Q0.0 或 Q0.1 输出。

以上 6 步是对高速脉冲输出的初始化。该过程可以用主程序中的程序段来实现,也可以用子程序来实现,这称为高速脉冲串初始化子程序。高速脉冲串在运行之前,必须要执行一次初始化程序段或初始化子程序。

### 【例 7-31】 PTO 应用实例。

#### (1) 控制要求

步进电机运行控制过程中,要从 A 点加速到 B 点后恒速运行,又从 C 点开始减速到 D 点,完成这一过程后用指示灯显示。电机的转动受脉冲控制,A 点和 D 点的脉冲频率为 2 kHz,B 点和 C 点的频率为 10 kHz,加速过程的脉冲数为 400 个,恒速转动的脉冲数为 4 000 个,减速过程脉冲数为 200 个,工作过程如图 7-31 所示。

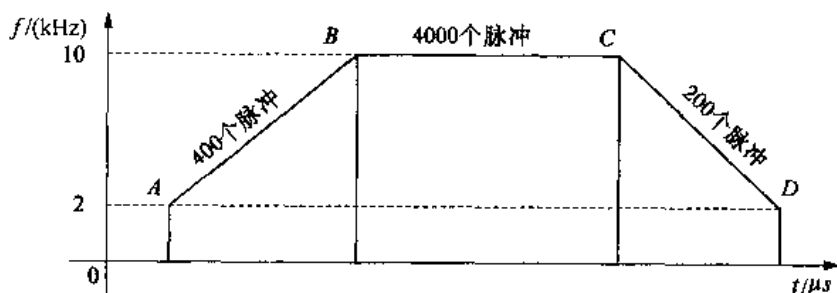


图 7-31 步进电机工作过程

#### (2) 解题分析

① 确定脉冲发生器及工作模式 本例要求 PLC 输出一定数量的多串脉冲。因此确定用 PTO 输出的多段管线方式。选择如下:选用高速脉冲串发生器为 Q0.0,并且确定 PTO 为 3 段脉冲管线(AB、BC 和 CD 段)。

② 设置控制字节 最大脉冲频率为 10 kHz,对应的周期值为 100 μs,因此时基选择为 μs 级。将 2#10100000,即 16#A0 写入控制字节 SMB67。功能为允许脉冲输出,多段 PTO 脉冲串输出,时基为 μs 级,不允许更新周期值和脉冲数。

③ 写入周期值、周期增量值和脉冲数 由于是 3 段脉冲,则需要建立 3 段脉冲的包络表,并对各段参数分别设置。包络表中各脉冲都是以周期为时间参数,所以必须先把频率换算为周期值(倒数计算即可)。给定段的周期增量按下式计算

$$\text{给定段的周期增量} = (T_{EC} - T_{IC}) / Q$$

式中:  $T_{EC}$  为该段结束周期时间;  $T_{IC}$  为该段初始周期时间;  $Q$  为该段的脉冲数量。

包络表结构如表 7-21 所列。

表 7-21 包络表内容

V 变量存储器地址	各块名称	实际功能	参数名称	参数值
VB400	段数	决定输出脉冲串数	总包络段数	3
VW401	段 1	电机加速阶段	初始周期	500 μs
VW403			周期增量	-1 μs
VD405			输出脉冲数	400

续表 7-21

V 变量存储器地址	各块名称	实际功能	参数名称	参数值
VW409	段 2	电机恒速运行阶段	初始周期	100 $\mu$ s
VW411			周期增量	0 $\mu$ s
VD413			输出脉冲数	4 000
VW417	段 3	电机减速阶段	初始周期	100 $\mu$ s
VW419			周期增量	2 $\mu$ s
VD421			输出脉冲数	200

④ 装入包络表首地址 将包络表的起始变量 V 存储器地址装入 SMW168 中。

⑤ 中断调用 高速输出完成时,调用中断程序,则信号灯变亮(本例中  $Q0.2=1$ )。脉冲输出完成,中断事件号为 19。用中断调用 ATCH 指令将中断事件 19 与中断子程序 INT\_0 连接起来,并全局开中断。

⑥ 执行 PLS 指令。

经以上设置并执行指令后,即可用 PLS 指令启动多段脉冲串,并由  $Q0.0$  输出。

本系统主程序、初始化子程序、包络表子程序和中断程序如图 7-32 所示。

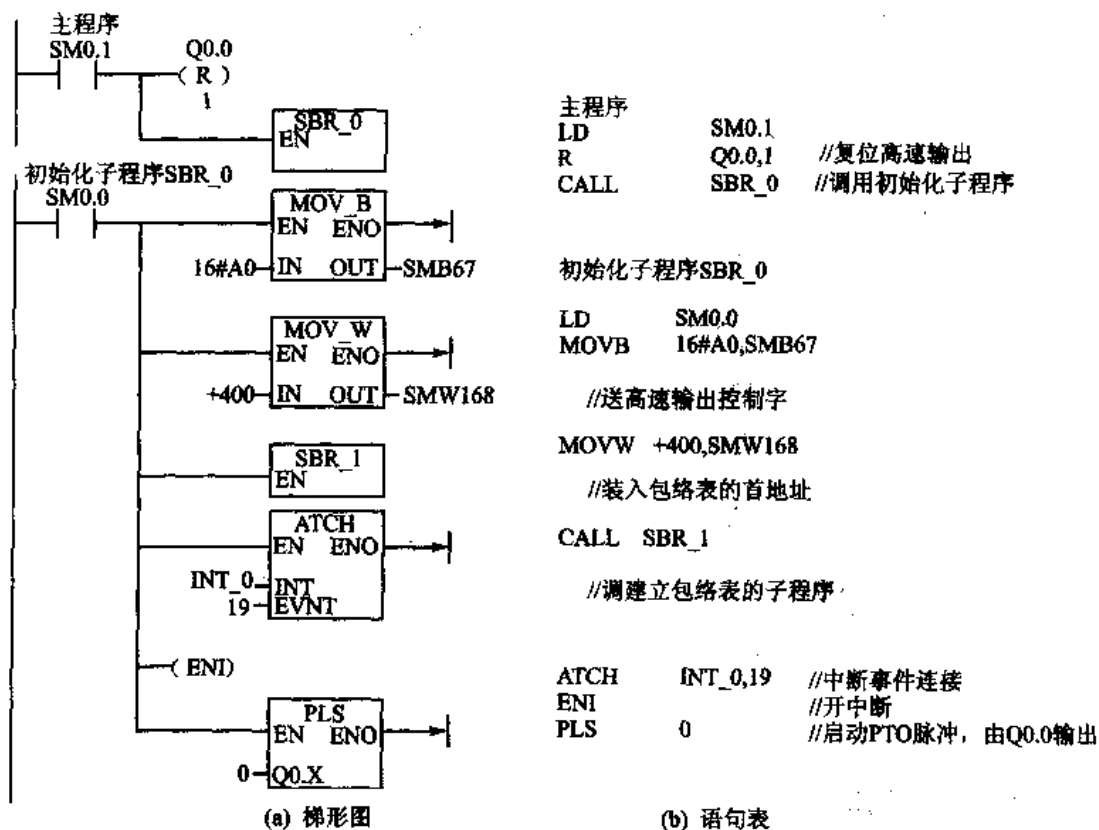


图 7-32 PTO 应用举例程序

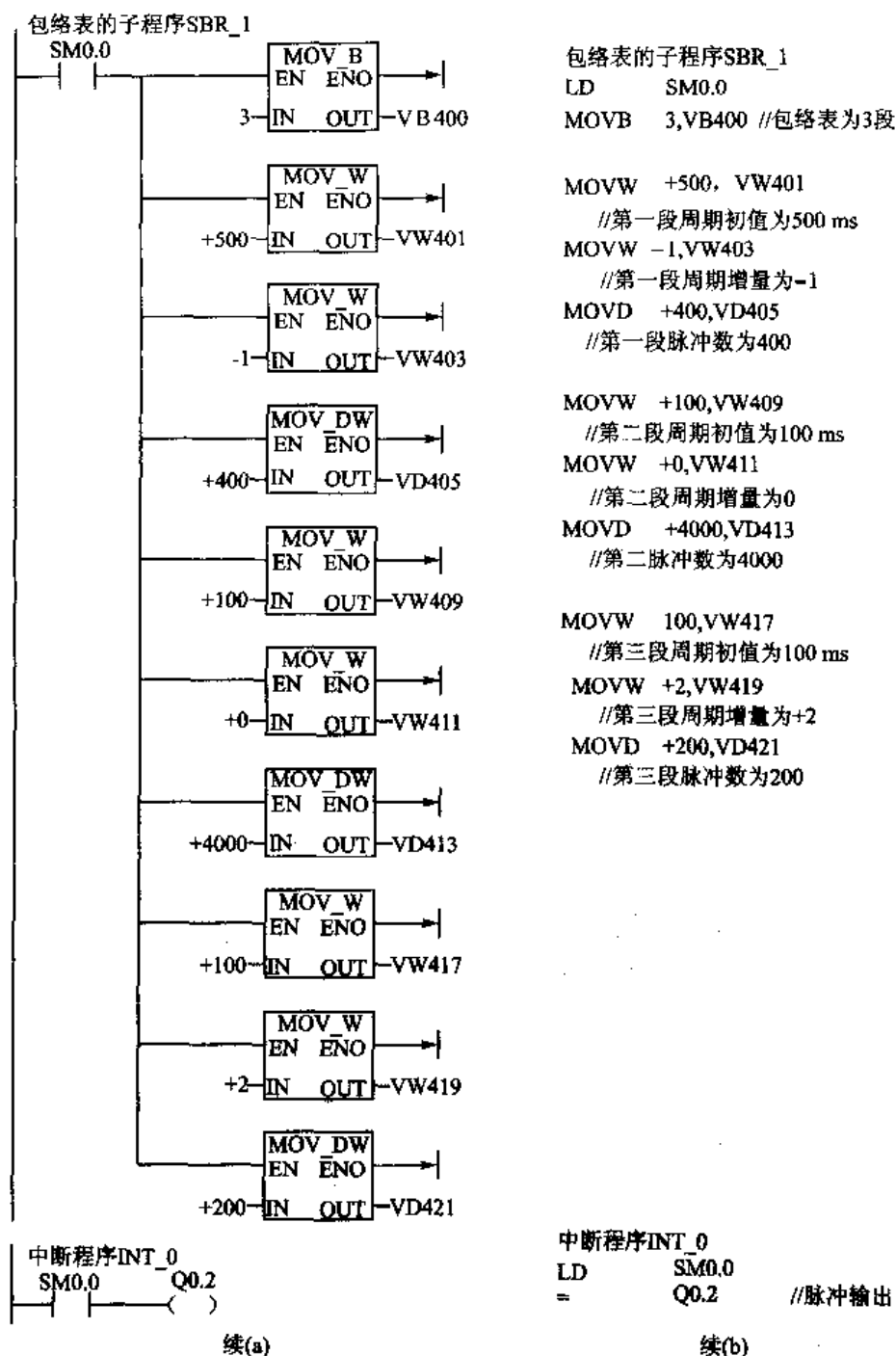


图 7-32 PTO 应用举例程序(续)

### 7.9.4 PWM 的使用

宽度可调脉冲输出 PWM, 用来输出占空比可调的高速脉冲。用户可以控制脉冲的周期和脉冲宽度, 完成特定的控制任务。

#### 1. 周期和脉冲宽度

周期: 单位可以是  $\mu\text{s}$  或  $\text{ms}$ ; 为 16 位无符号数据, 周期变化范围是  $50 \sim 65\,535\, \mu\text{s}$  或  $2 \sim 65\,535\, \text{ms}$ 。如果编程时设定周期单位小于 2, 系统默认则按 2 个时间单位进行设置。

脉冲宽度:单位可以是  $\mu\text{s}$  或  $\text{ms}$ ;为 16 位无符号数据,脉冲宽度变化范围是  $0\sim 65\,535\,\mu\text{s}$  或  $0\sim 65\,535\,\text{ms}$ 。

如果设定脉宽等于周期(使占空比为 100 %),则输出连续接通;如果设定脉宽等于 0(使占空比为 0 %),则输出断开。

## 2. 更新方式

有两种方式可改变高速 PWM 波形的特性:同步更新和异步更新。

同步更新:同步更新时,波形的变化发生在周期的边缘,形成平滑转换。在不需要改变时间基准的情况下,可以采用同步更新。

异步更新:在改变脉冲发生器的时间基准的情况下,必须采用异步更新。异步更新有时会引起脉冲输出功能被瞬时禁止,或波形不同步,引发被控制设备的振动。

由此可以看出,要尽可能采用 PWM 同步更新。为此要事先选一个适合于所有时间周期的时间基准。

## 3. PWM 的使用

使用高速脉冲串输出时,要按以下步骤进行。

① 确定脉冲发生器 它包括两方面工作,即根据控制要求,一是选用高速脉冲串输出端(发生器);二是选择工作模式为 PWM。

② 设置控制字节 按控制要求设置 SMB67 或 SMB77 特殊寄存器。

③ 写入周期值和脉冲宽度值 按控制要求将脉冲周期值写入 SMW68 或 SMW78 的特殊寄存器,将脉宽值写入 SMW70 或 SMW80 特殊寄存器。

④ 执行 PLS 指令 经以上设置并执行指令后,即可用 PLS 指令启动 PWM,并由 Q0.0 或 Q0.1 输出。

以上步骤是对高速计数器的初始化,它可以用主程序中的程序段来实现,也可以用子程序段来实现,这称为 PWM 初始化子程序。脉冲输出之前,必须要执行一次初始化程序段或初始化子程序。

**【例 7-32】** 设计一段程序,从 PLC 的 Q0.0 输出一串脉冲。该串脉冲脉宽的初始值为 0.5 s,周期固定为 5 s,其脉宽每周期递增 0.5 s,当脉宽达到设定的 4.5 s 时,脉宽改为每周期递减 0.5 s,直到脉宽减为零为止。以上过程重复执行。

**解题分析** 该题是 PWM 的典型应用,因为每个周期都有要求的操作,所以需要把 Q0.0 接到 I0.0,采用输入中断的方法完成控制任务。另外还要设置一个标志,来决定什么时候脉冲递增,什么时候脉冲递减。控制字设定为 16#DA,即 11011010,把它放到 SMB67 中,它表示输出端 Q0.0 为 PWM 方式,不允许更新周期,允许更新脉宽,时间基准单位为 ms 量级,同步更新,且允许 PWM 输出。

梯形图程序如图 7-33 所示,它包括主程序、子程序和中断程序。

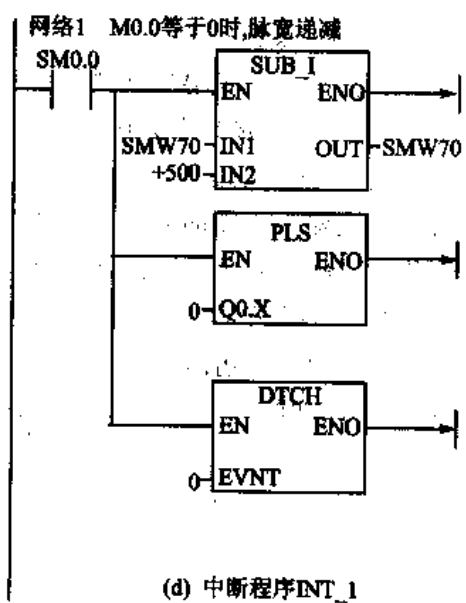
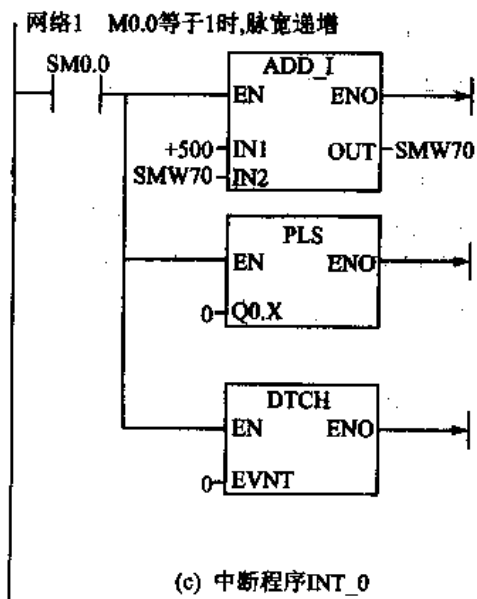
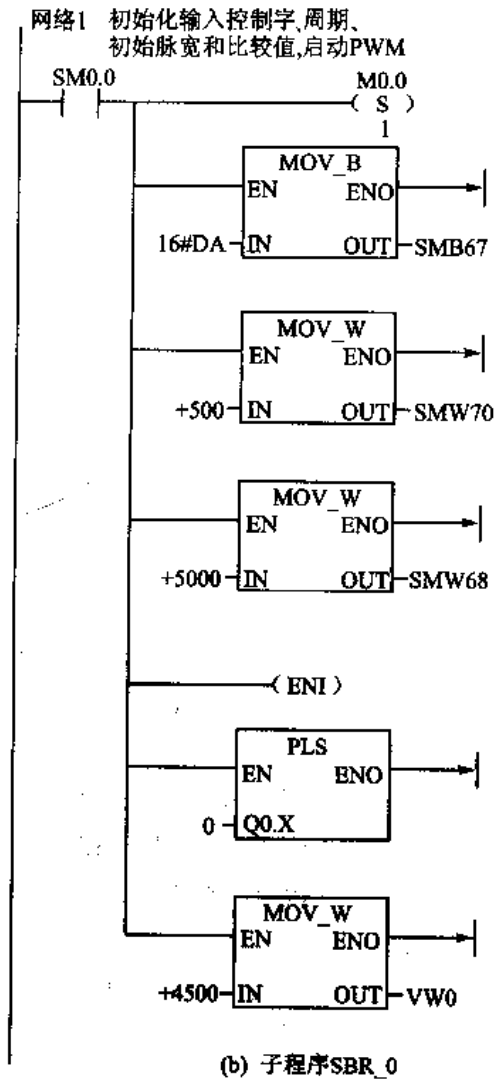
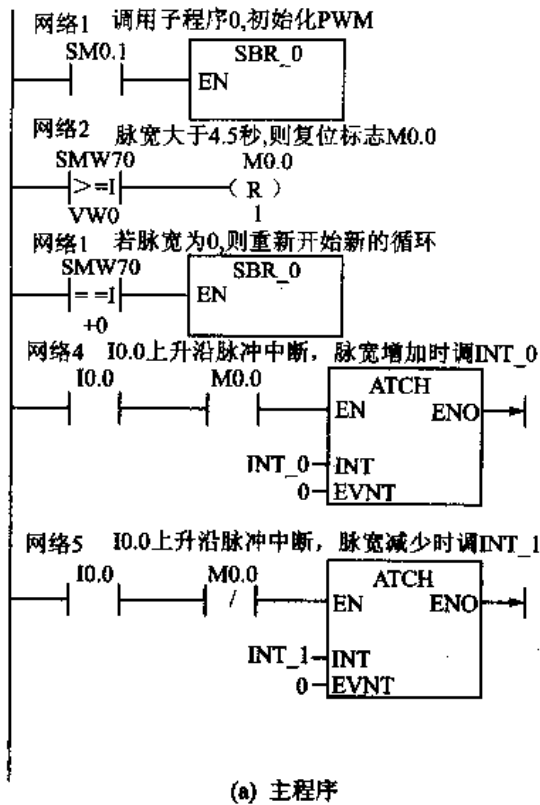


图 7-33 PWM 应用举例程序

## 7.10 PID 回路指令

在有模拟量的控制系统中,经常用到 PID 运算来执行 PID 回路的功能,PID 回路指令使这一任务的编程和实现变得非常容易。

### 7.10.1 PID 算法

如果一个 PID 回路的输出  $M$  是时间  $t$  的函数,则可以看做是比例项、积分项和微分项三部分之和。即

$$M(t) = K_c \cdot e + K_c \int_0^t e dt + M_0 + K_c \cdot de/dt$$

以上各量都是连续量,第一项为比例项,最后一项为微分项,中间两项为积分项。其中  $e$  是给定值与被控制变量之差,即回路偏差。 $K_c$  为回路的增益。用计算机处理这样的控制算式,即连续的算式必须周期性地采样并进行离散化,同时各信号也要离散化,公式如下

$$M_n = K_c \cdot (SP_n - PV_n) + K_c \cdot (T_i/T_s) \cdot (SP_n - PV_n) + MX + K_c \cdot (T_d/T_s) \cdot (PV_n - PV_{n-1})$$

公式中包含 9 个用来控制和监视 PID 运算的参数,在 PID 指令使用时要构成回路表,回路表的格式如表 7-22 所列。

表 7-22 PID 回路表

参 数	地址偏移量	数据格式	I/O 类型	描 述
过程变量当前值 $PV_n$	0	双字,实数	I	过程变量,0.0~1.0
给定值 $SP_n$	4	双字,实数	I	给定值,0.0~1.0
输出值 $M_n$	8	双字,实数	I/O	输出值,0.0~1.0
增益 $K_c$	12	双字,实数	I	比例常数,正、负
采样时间 $T_s$	16	双字,实数	I	单位为 s,正数
积分时间 $T_i$	20	双字,实数	I	单位为分钟,正数
微分时间 $T_d$	24	双字,实数	I	单位为分钟,正数
积分项前值 $MX$	28	双字,实数	I/O	积分项前值,0.0~1.0
过程变量前值 $PV_{n-1}$	32	双字,实数	I/O	最近一次 PID 变量值

### 7.10.2 PID 回路指令及使用

#### 1. PID 回路指令 (Proportional, Integral, Derivative Loop)

指令格式:LAD 和 STL 格式如图 7-34 所示。

功能描述:该指令利用回路表中的输入信息和组态信息,进行 PID 运算。

数据类型:回路表的起始地址 TBL 为 VB 指定的字节型数据;回路号 LOOP 是 0~7 的常数。

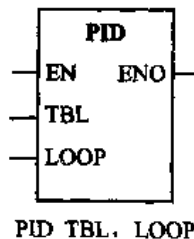


图 7-34 PID 指令格式

## 2. PID 回路号

用户程序中最多可有 8 条 PID 回路,不同的 PID 回路指令不能使用相同的回路号,否则,会产生意外的后果。

## 3. PID 指令的使用

使用 PID 指令的关键是对采集到的数值和计算出来的 PID 控制结果数据进行转换及标准化。下面是数值转换及标准化的步骤。

### (1) 回路输入量的转换及归一化

给定值和过程变量都是实际的工程量,其幅度、范围和测量单位都会不同,用可编程序控制器控制 PID 回路时,要把实际测量输入量、设定值和回路表中的其他输入参数进行标准化处理,即用程序把它们转化为 PLC 能够识别和处理的数据,也即把它们转化为无量纲的归一化纯量、浮点数的格式。

第一步,将工程实际值由 16 位整数转化为实数,程序如下:

```
XORD    AC0,AC0    //清累加器 AC0
ITD     AIW0,AC0    //把整数转化为双整数(设采集数据通道地址为 AIW0)
DTR     AC0,AC0    //把双整数转化为实数
```

第二步,将实数格式的工程实际值转化为【0.0,1.0】之间的无量纲相对值,用下式来完成这一过程:

$$R_{\text{Norm}} = (R_{\text{Raw}} / \text{Span}) + \text{Offset}$$

式中: $R_{\text{Norm}}$ 为工程实际值的归一化值; $R_{\text{Raw}}$ 为工程实际值的实数形式值,未归一化处理。标准化实数又分为双极性(围绕 0.5 上下变化)和单极性(以 0.0 为起点在 0.0 和 1.0 之间的范围内变化)两种。对于双极性,Offset 为 0.5;对于单极性,Offset 为 0,Span 表示值域的大小,通常单极性时取 32 000,双极性时取 64 000。

以下程序段用于将 AC0 中的双极性模拟量进行归一化处理(可紧接上面的程序):

```
/R      64000.0,AC0 //将 AC0 中的双极性模拟量值进行归一化
+R      0.5,AC0     //Offset 处理(双极性时)
MOVR    AC0,VD200   //将归一化结果存入 TABLE 中(设 TABLE 表地址为 VB200)
```

### (2) 回路控制输出转换为按工程量标定的整数值

程序执行时把各个标准化实数量用离散化 PID 算式进行处理,产生一个标准化实数运算结果,这一结果同样也要用程序将其转化为相应的 16 位整数,然后周期性地将其传送到指定的 AQW 输出,用以驱动模拟量的负载,实现模拟量的控制。这一转换实际上是归一化过程的逆过程。

第一步,用下式将回路输出转换为按工程量标定的实数格式:

$$R_{\text{scal}} = (M_n - \text{Offset}) \cdot \text{Span}$$

式中: $R_{\text{scal}}$ 为已按工程量标定的实数格式的回路输出; $M_n$ 为归一化实数格式的回路输出。程序如下:

```
MOVR    VD208,AC0   //将回路输出结果(设 TABLE 表地址为 VB200)放入 AC0
-R      0.5,AC0      //双极性场合时减去 0.5
*R      64 000,AC0   //将 AC0 中的值按工程量标定
```

第二步,将已标定的实数格式的回路输出转化为 16 位的整数格式,并输出。

```
TRUNC   AC0,AC0     //取整数
```

DTI      AC0,AC0      //双整数转换为整数  
MOVW    AC0,AQW0      //把整数值送到模拟量输出通道(设为 AQW0)

#### 4. 选择 PID 回路类型

在大部分模拟量的控制中,使用的回路控制类型并不是比例、积分和微分三者俱全。例如,只需要比例回路或只需要比例积分回路。通过对常量参数的设置,可以关闭不需要的控制类型。

关闭积分回路:把积分时间  $T_i$  设置为无穷大,此时虽然由于有初值  $MX$  使积分项不为零,但积分作用可以忽略。

关闭微分回路:把微分时间  $T_d$  设置为 0,微分作用即可关闭。

关闭比例回路:把比例增益  $K_c$  设置为 0,则只保留积分和微分项。

实际工作中,使用最多的是 PI 调节器。

说明:实际使用 PID 指令时,还有变量范围、控制方式等许多问题要具体考虑,所以更详细的内容请参考系统使用手册。

#### 【例 7-33】 PID 指令应用实例。

##### (1) 控制要求

某一水箱有一条进水管和一条出水管,进水管的水流量随时间不断变化,要求控制出水管阀门的开度,使水箱内的液位始终保持在水满时液位的一半。系统使用比例、积分及微分控制,假设采用下列控制参数值: $K_c$  为 0.4,  $T_s$  为 0.2 s,  $T_i$  为 30 min,  $T_d$  为 15 min。

##### (2) 解题分析

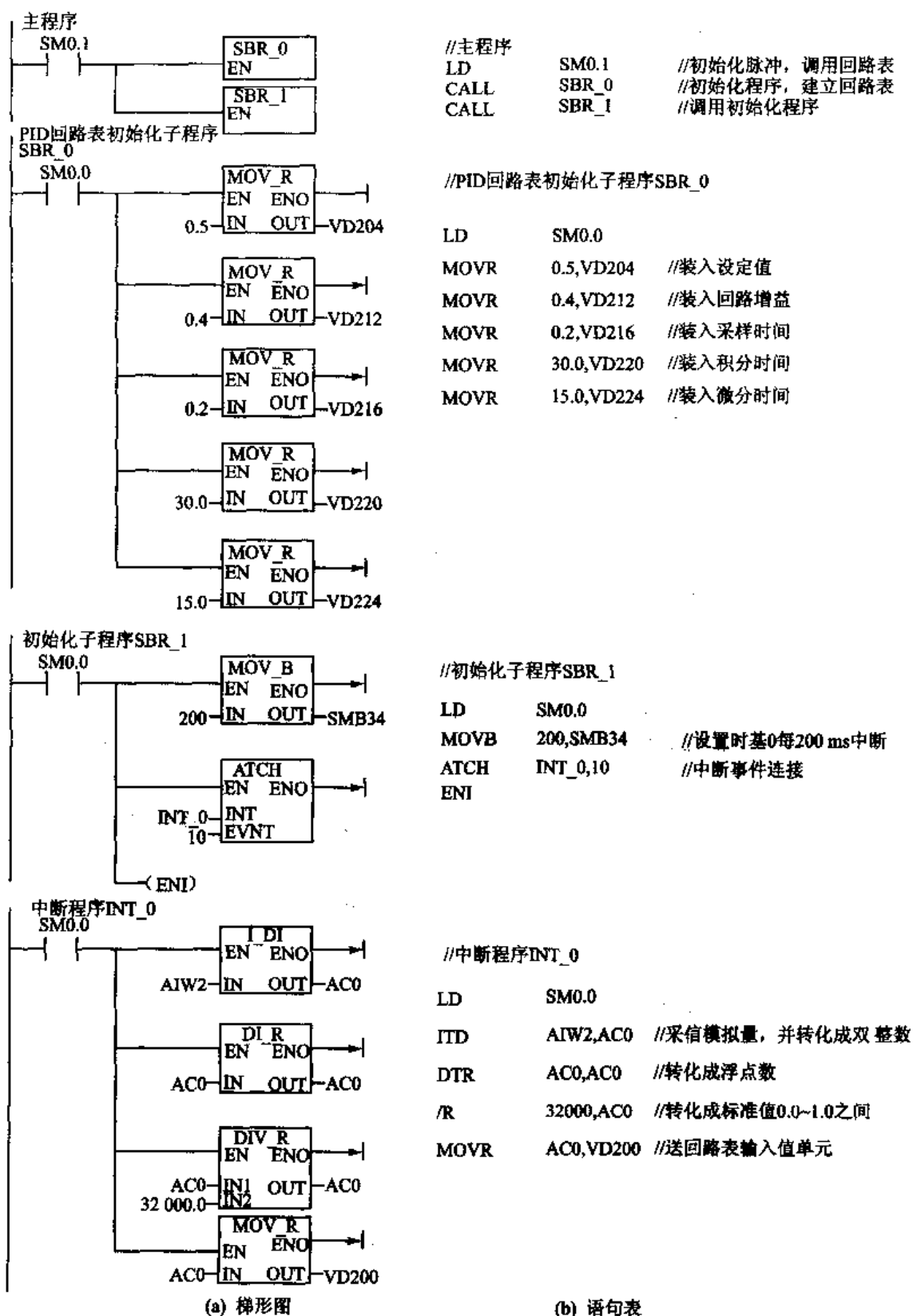
本系统标准化时可采用单极性方案,系统的输入来自液位计的液位测量采样;设定值是液位的 50 %,输出是单极性模拟量,用以控制阀门的开度,可以在 0 %~100 %之间变化。

##### (3) 程序实现

本程序只是模拟量控制系统的 PID 程序主干,对于现场实际问题,还要考虑诸多方面的影响因素。

本程序的主程序、回路表初始化子程序 SBR\_0、初始化子程序 SBR\_1 和中断程序 INT\_0 如图 7-35 所示。

本例中模拟量输入通道为 AIW2,模拟量输出通道为 AQW0。I0.4 是手动/自动转换开关信号,I0.4 为 1 时,为系统自动运行状态。

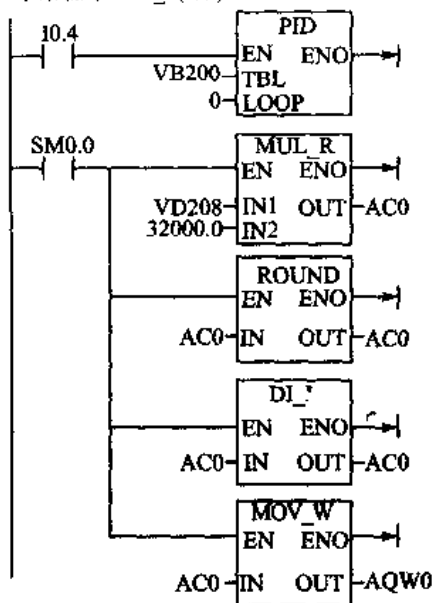


(a) 梯形图

(b) 语句表

图 7-35 PID 控制举例

中断程序 INT\_0(续)



(a) 梯形图

LD	I0.4	//手动/自动切换
PID	VB200,0	//执行PID指令
LD	SM0.0	
MOVR	VD208,AC0	//控制量输出值
*R	32000.0,AC0	//将控制量转化成实际值
ROUND	AC0,AC0	//即整数形式
DTI	AC0,AC0	//双整数到整数
MOVW	AC0,AQW0	//控制量输出

(b) 语句表

图 7-35 PID 控制举例(续)

## 本章小结

本章重点介绍了 S7-200 PLC 的功能指令及使用方法,给出了很多例题。通过学习,应了解功能指令在 PLC 中的主要作用,在使用时做到胸中有数。

(1) 数据处理指令主要涉及非数值运算的数据操作,包括传送类指令、移位指令、循环移位指令、字节交换指令、填充指令、数据类型转换指令和表功能指令等。在处理实际问题时,数据类型转换指令应用较多。

(2) 运算指令包括算术运算指令和逻辑运算指令,它们增强了 PLC 的数据处理能力,拓宽了 PLC 的应用领域。对 S7-200 来说,因为使用 STL 编程时存储单元的特殊性,所以不熟练时,建议尽量使用 LAD 方式处理运算指令。

学会使用这些指令的同时,还应学会结合数学方法灵活运用这些指令,以完成较为复杂的运算任务。

(3) 利用时钟指令可以方便地设定和读取系统的实时时钟,时钟指令使得对实时时间的处理变得非常容易,使用时要注意 PLC 本身时钟的设定,以及时钟数据的存储形式。

(4) 没有中断就实现不了复杂的控制任务,中断技术在可编程序控制器的人机联系、实时处理、通信处理和网络中占有重要地位。学习中断使用技术不仅需要深刻领会中断的实质内容,还要通过学习高速指令和通信指令等来掌握它的使用方法。

(5) 高速输入/输出指令和 PID 回路指令可以用来方便地完成特定的复杂控制任务,这些指令都用到了—定数量的内部特殊功能存储器,应掌握相应的控制参数、状态参数和变量值等的设定方法。

## 练 习 题

1. 写一段梯形图程序,实现将 VB20 开始的 100 个字节型数据送到 VB400 开始的存储区,这 100 个数据的相对位置在移动前后不发生变化。
2. 有一组数据存放在 VB300 开始的 10 个字节中,采用间接寻址方式设计一段程序,将这 10 个字节的数据存储到从 VB200 开始的存储单元中。
3. 用功能指令实现时间为 6 个月的延时,试设计梯形图程序。
4. 编写一段程序计算  $\sin 120^\circ + \cos 10^\circ$  的值。
5. 试设计一个记录某台设备运行时间的程序。I0.0 为该设备工作状态输入信号,要求记录其运行时的时、分、秒,并把秒值在 QB0 上显示。
6. 用时钟指令控制路灯的定时接通和断开,5 月 15 日到 10 月 15 日,每天 20:00 开灯,6:00 关灯;10 月 16 日到 5 月 14 日,每天 18:00 开灯,7:00 关灯,并可校准 PLC 的时钟。请编写梯形图程序。
7. 试设计一个计数器程序,要求如下:
  - (1) 计数范围是 0~255;
  - (2) 计数脉冲为 SM0.5;
  - (3) 输入 I0.0 的状态改变时,则立即激活输入/输出中断程序。中断程序 0 和 1 分别将 M0.0 置成 1 或 0;
  - (4) M0.0 为 1 时,计数器加计数;M0.0 为 0 时,计数器减计数。计数器的计数值在 PLC 输出端 QB0 显示。
8. 某一过程控制系统,其中一个单极性模拟量输入参数从 AIW0 采集到 PLC 中,通过 PID 指令计算出的控制结果从 AQW0 输出到控制对象。PID 参数表起始地址为 VB100。试设计一段程序完成下列任务:
  - (1) 每 200 ms 中断一次,执行中断程序;
  - (2) 在中断程序中完成对 AIW0 的采集、转换及归一化处理;完成回路控制输出值的工程量标定及输出。

## 第八章 PLC 的网络通信技术的应用

随着计算机网络技术的发展以及各企业对工厂自动化程度要求的不断提高,自动控制从传统的集中式向多元化分布式方向发展。为了适应这种形式的发展,世界各 PLC 生产厂家纷纷给自己的产品增加了通信及联网的功能,并研制开发出自己的 PLC 网络系统。如三菱的 MELSEC NET、CCLINK 网,欧姆龙的 Controller Link,西门子的 SINEC H1 局域网等。现在即使是微型和小型的 PLC 也都具有了网络通信接口。今后网络总的发展趋势是向高速、多层次、大信息吞吐量、高可靠性和开放式(即通信协议向国际标准或地区通用工艺标准靠近)的方向发展。

PLC 网络系统十分丰富,涉及面广,本章首先介绍一些通信网络基础知识,然后主要介绍西门子 S7-200 的网络系统。

### 8.1 通信网络的基础知识

#### 8.1.1 数据通信方式

不同的独立系统经由传输线路互相交换数据便是通信,构成整个通信的线路称之为网络。通信的独立系统可以是计算机、PLC 或其他有数据通信功能的数字设备,称为 DTE(Data Terminal Equipment)。传输线路的介质可以是双绞线、同轴电缆、光纤或无线电波等。

##### 1. 数据传输方式

##### (1) 并行通信(Parallel Communication)与串行通信(Serial Communication)

并行通信:所传送的数据的各位同时发送或接收。

串行通信:所传送的数据按顺序一位一位地发送或接收。

并行通信传递数据快,但由于一个并行数据有多少位二进制数就需要多少根传送线,所以通常用于近距离传输。在远距离传输时,会导致线路复杂,成本高,而且在传输过程中,容易因线路的因素使电压标准位发生变化(最常见的是电压衰减和信号互相干扰问题(Cross Talk),因此使得传输的数据发生错误。

串行通信只需一根到二根传输线,在长距离传送时,通信线路简单且成本低,但传递速度比并行通信速度低,故常用于长距离传送且速度要求不高的场合。近年来串行通信技术有了很快的发展,通信速度甚至可以达到 MB/s 的数量级,因此在分布式控制系统中得到了广泛的应用。

##### (2) 同步传送和异步传送

发送端与接收端之间的同步问题是数据通信中的一个重要问题。同步不好,轻者导致误码增加,重者使整个系统不能正常工作。为解决这一传送过程中的问题,在串行通信中采用了两种同步技术——异步传送和同步传送。

异步传送:异步传送也称起止式传送,它是利用起止法来达到收发同步的。

在异步传送中,被传送的数据编码成一串脉冲。字节传送的起始位由“0”开始;然后是被编码的字节,通常规定低位在前,高位在后,接下来是校验位(可省略);最后是停止位“1”(可以是1位、1.5位或2位)表示字节的结束。例如,传送一个 ASCII 字符(每个字节有7位),若选用2位停止位,那么传送这个七位的 ASCII 字符就需11位,其中包含1位起始位、1位校验位、2位停止位和7位数据位。其格式如图8-1所示。

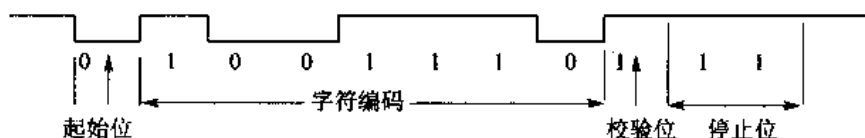


图 8-1 异步传送

异步传送就是按照上述约定好的固定格式,一帧一帧地传送,因此采用异步传送方式时,硬件结构简单,但是传送每一个字节就要加起始位、停止位,因而传送效率低,主要用于中、低速的通信。

同步传送:同步传送在数据开始处就用同步字符(通常为1~2个)来指示。由定时信号(时钟)来实现收发端同步,一旦检测到与规定的同步字符相符合,接下去就连续按顺序传送数据。在这种传送方式中,数据以一组数据(数据块)为单位传送,数据块中每字节不需要起始位和停止位,因而就克服了异步传送效率低的缺点,但同步传送所需的软、硬件价格是异步传送的8~12倍。因此通常在数据传递速率超过2 000 bps的系统中才采用同步传送方式。

## 2. 数据传送方向

在通信线路上按照传送的方向可以划分为单工、半双工和全双工通信方式,如图8-2所示。

### (1) 单工通信方式

单工通信就是指数据的传送始终保持同一个方向,而不能进行反向传送,如图8-2(a)所示。其中A端只能作为发送端发送数据,B端只能作为接收端接收数据。

### (2) 半双工通信方式

半双工通信就是指信息流可以在两个方向上传送,但同一时刻只限于一个方向传送,如图8-2(b)所示。其中A端和B端都具有发送和接收的功能,但传送线路只有一条,或者A端发送B端接收;或者B端发送A端接收。

### (3) 全双工通信方式

全双工通信能在两个方向上同时发送和接收,如图8-2(c)所示。A端和B端双方都可以一方面发送数据,一方面接收数据。

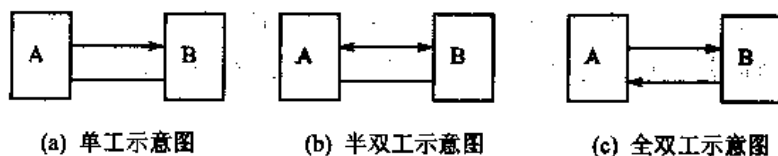


图 8-2 通信方向

### 3. 传送介质

目前普遍使用的传送介质有:同轴电缆、双绞线、光缆,其他介质如无线电、红外微波等在 PLC 网络中应用很少。其中双绞线(带屏蔽)成本低、安装简单;光缆尺寸小、质量轻、传输距离远,但成本高、安装维修需专用仪器,具体性能如表 8-1 所列。

表 8-1 传送介质性能比较

性 能	传 送 介 质		
	双 绞 线	同 轴 电 缆	光 缆
传送速率	9.6 KB/s~2 MB/s	1~450 MB/s	10~500 MB/s
连接方法	点到点 多点 1.5 km 不用中继器	点到点 多点 10 km 不用中继器(宽带) 1~3 km 不用中继器(基带)	点到点 50 km 不用中继器
传送信号	数字、调制信号、纯模拟信号 (基带)	调制信号,数字(基带),数 字、声音、图像(宽带)	调制信号(基带) 数字、声音、图像(宽带)
支持网络	星形、环形、小型交换机	总线型、环形	总线型、环形
抗干扰	好(需外屏蔽)	很 好	极 好
抗恶劣环境	好	好,但必须将电缆与腐蚀物 隔开	极好,耐高温和其他恶劣 环境

### 4. 串行通信接口

工业网络中,在设备或网络之间大多采用串行通信方式传送数据,常用的有以下几种串行通信接口。

#### (1) RS-232 接口

RS-232 接口是 1969 年由美国电子工业协会 EIA(Electronic Industries Association)所公布的串行通信接口标准。它既是一种协议标准,又是一种电气标准,它规定了终端和通信设备之间信息交换的方式和功能。

RS-232 接口是计算机普遍配备的接口,应用既简单又方便。它采用按位串行的方式,单端发送、单端接收,所以数据传送速率低,抗干扰能力差,传送波特率为 300、600、1 200、4 800、9 600、19 200 等。在通信距离近、传送速率和环境要求不高的场合应用较广泛。

#### (2) RS-485 接口

RS-485 接口的传输线采用差动接收和平衡发送的方式传送数据,有较高的通信速率(波特率可达 10 MB 以上)和较强的抑制共模干扰能力,输出阻抗低,并且无接地回路。这种接口适合远距离传输,是工业设备的通信中应用最多的一种接口。

#### (3) RS-422 接口

RS-422 接口传输线采用差动接收和差动发送的方式传送数据,也有较高的通信速率(波特率可达 10 MB 以上)和较强的抗干扰能力,适合远距离传输,工厂应用较多。

RS-422 与 RS-485 的区别在于 RS-485 采用的是半双工传送方式,RS-422 采用的是全双工传送方式;RS-422 用两对差分信号线,RS-485 只用一对差分信号线。

### 8.1.2 网络概述

#### 1. 网络结构概述

##### (1) 简单网络

多台设备通过传输线相连,可以实现多设备之间的信息交换,形成网络结构。图 8-3 就是一种最简单的网络结构,它由单个主设备和多个从设备构成。

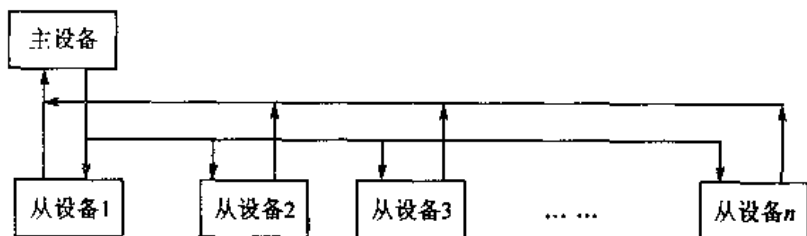


图 8-3 多设备通信(简单网络)

##### (2) 多级网络

现代大型工业企业中,一般采用多级网络的形式。可编程序控制器制造商常用生产金字塔结构来描述其产品可实现的功能。这种金字塔结构的特点是:上层负责生产管理,底层负责现场监测与控制,中间层负责生产过程的监控与优化。

国际标准化组织(ISO)对企业自动化系统确立了初步的模型,如图 8-4 所示。

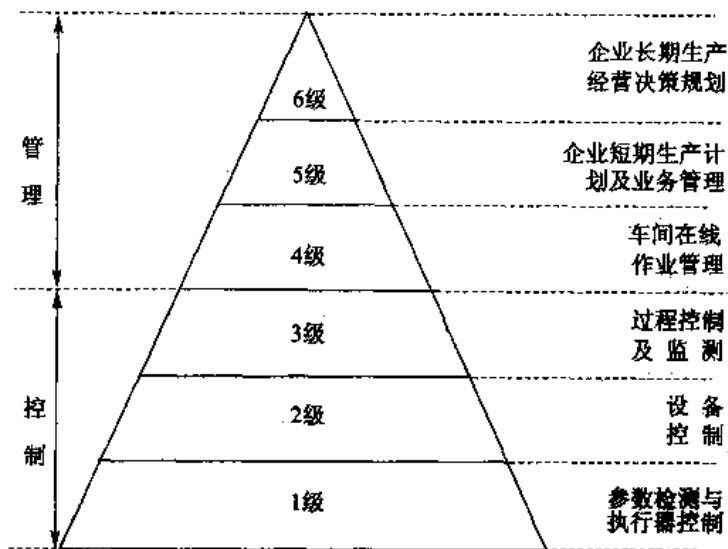


图 8-4 ISO 企业自动化系统模型

不同 PLC 厂家自动化系统的网络结构层数及各层的功能分布有所差异,但在工厂自动化系统中,都是 PLC 及其网络从上到下各层在通信基础上相互协调,共同发挥着作用。

实际工厂中一般采用 3~4 级子网构成复合型结构,而不一定都是这 6 级,不同的层采用相应的通信协议。

#### 2. 通信协议

通信双方就如何交换信息所建立的一些规定和过程,称为通信协议。在可编程序控制器网络中配置的通信协议分为两大类:一类是通用协议,一类是公司专用协议。

### (1) 通用协议

在网络金字塔的各个层次中,高层次子网中一般采用通用协议,如 PLC 网之间的互联及 PLC 网与其他局域网的互联,这表明工业网络向标准化和通用化发展的趋势。高层子网传送的是管理信息,与普通商业网络性质接近,同时要解决不同种类的网络互联。国际标准化组织 ISO(International Standard Organization)于 1978 年提出了开放系统互联 OSI(Open Systems Interconnection)的模型,它所用的通信协议一般为 7 层,如图 8-5 所示。

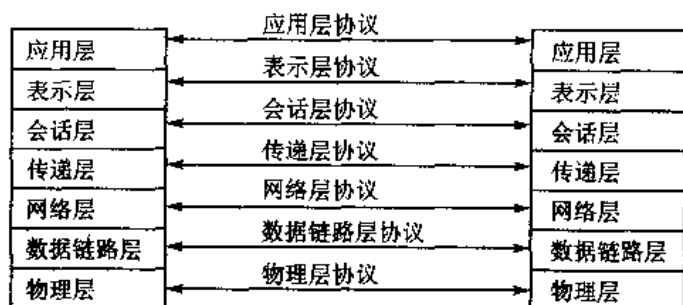


图 8-5 国际 OSI 企业自动化系统模型

在该模型中,最底层为物理层,实际通信就是通过物理层在物理互联媒体上进行的,上面的任何层都以物理层为基础,对等层之间可以实现开放系统互联。常用的通用协议有两种:一种是 MAP 协议,一种是 Ethernet 协议。

### (2) 公司专用协议

低层子网和中间层子网一般采用公司专用协议,尤其是最底层子网,由于传送的是过程数据及控制命令,这种信息较短,但实时性要求高。公司专用协议的层次一般只有物理层、链路层及应用层,而省略了通用协议所必须的其他层,信息传递速率快。

## 8.2 S7-200 的通信与网络

西门子 S7-200 系列 PLC 是一种小型整体结构形式的 PLC,内部集成的 PPI 接口为用户提供了强大的通信功能,其 PPI 接口(即编程口)的物理特性为 RS-485,根据不同的协议通过此接口与不同的设备进行通信或组成网络。

### 8.2.1 S7 系列 PLC 网络层次的结构

西门子公司的生产金字塔由 4 级组成,由下到上依次是过程测量与控制级、过程监控级、工厂与过程管理级、公司管理级。S7 系列的网络结构如图 8-6 所示。

西门子生产金字塔的 4 级子网由 3 级总线复合而成:

最底一级为 AS-I 级总线,负责与现场传感器和执行器的通信,也可以是远程 I/O 总线(负责 PLC 与分布式 I/O 模块之间的通信)。

中间一级是 Profibus 级总线,它是一种新型总线,采用令牌控制方式与主从轮询相结合的存取控制方式,可实现现场、控制和监控 3 级的通信。中间级也可采用主从轮询存取方式的主从式多点链路。

最高一级为工业以太网(Ethernet)使用通用协议,负责传送生产管理信息。

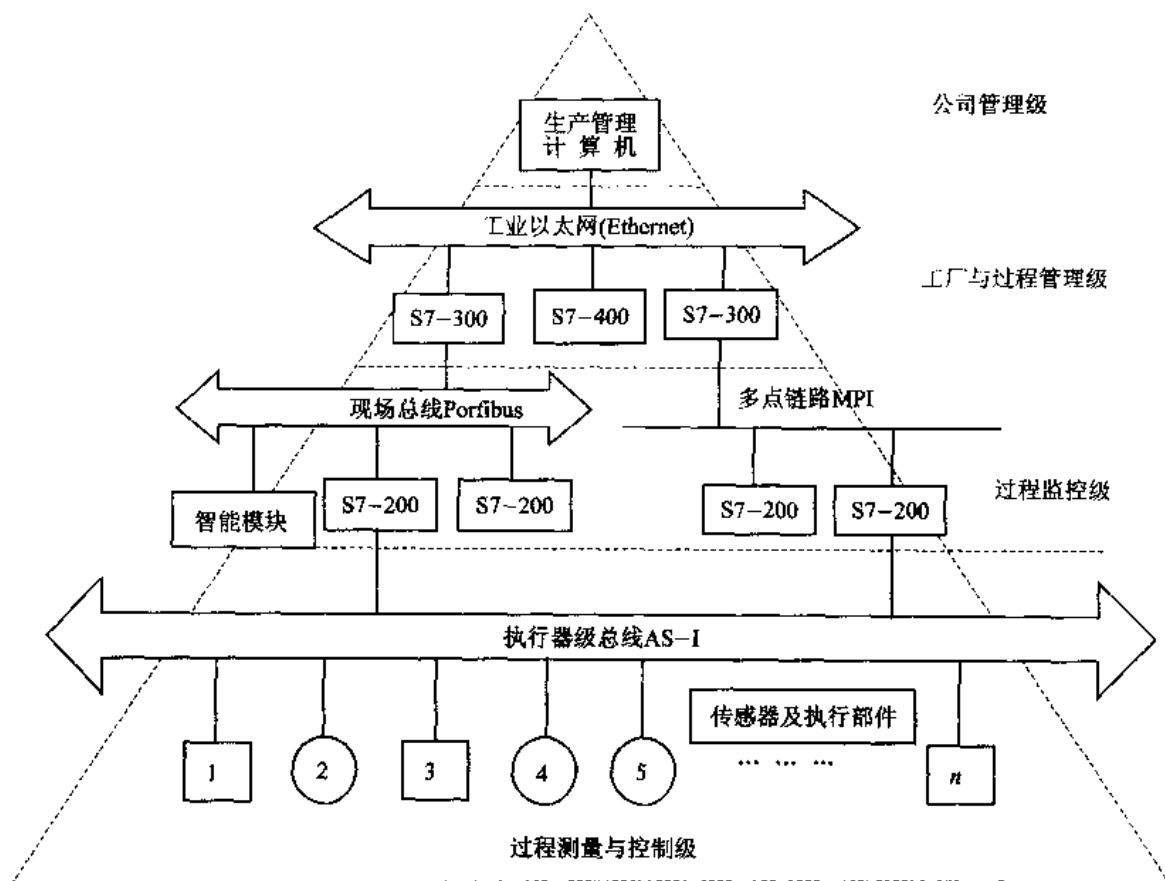


图 8-6 西门子的生产金字塔及网络

在对网络中的设备进行配置时，必须对设备的类型、在网络中的地址和通信的波特率进行设置。

在网络中的设备被定义为两类：主站和从站。主站设备可以对网络上其他设备发出请求，也可以对网络上的其他主站设备的请求作出响应。从站只响应来自主站的申请。典型的主站设备包括编程软件、TD200 等 HMI 产品和 S7-300、S7-400 PLC。从站设备只能对网络上主站的请求作出响应，自己不能发送通信请求。一般情况下，S7-200 PLC 被配置为从站。当 S7-200 需要从另外的 S7-200 读取信息时，S7-200 也可以定义为主站（点对点通信）。

在网络中的设备必须有惟一的地址，以保证数据发送到正确的设备或从正确的设备接收数据，S7-200 支持的网络地址为 0 到 126。对于有两个通信口（CPU 226）的 S7-200，每一个通信口可以有不同的地址。S7-200 的地址在编程软件的系统块（System Block）中设定。S7-200 的缺省地址是 2，编程软件的缺省地址是 0，操作面板（如 TD200、OP37 和 OP37）的缺省地址是 1。

数据通过网络传输的速度称为波特率，其单位通常为 KB 或 MB，表示单位时间内传输数据的多少，在同一网络中所有的设备必须被配置成相同的波特率。S7-200 波特率的配置在编程软件的系统块中完成。

### 8.2.2 S7-200 PLC 网络的通信协议

S7-200 CPU 支持多样的通信协议。根据所使用的 S7-200 CPU，网络可以支持一个或

多个协议,包括通用协议和公司专用协议。专用协议包括点到点(Point-to-Point)接口协议(PPI)、多点(Multi-Point)接口协议(MPI)、Profibus 协议、自由通信接口协议和 USS 协议。

PPI、MPI、Profibus 协议在 OSI 七层模式通信结构的基础上,通过令牌环网实现,令牌环网遵守欧洲标准 EN 50170 中的过程现场总线(Profibus)标准。这些协议都是异步、基于字符传输的协议,带有起始位、8 位数据、偶校验和一个停止位。通信帧由特殊的起始和结束字符、源和目的站地址、帧长度和数据完整性检查组成。如果使用相同的波特率,这些协议可以在一个网络中同时运行,而不相互影响。

网络通信通过 RS-485 标准双绞线实现,在一个网络段上允许最多连接 32 台设备。根据波特率不同,网络段的确切长度可以达到 1 200 m(3 936 ft)。采用中继器连接,各段可以在网络上连接更多的设备,延长网络的长度。根据不同的波特率,采用中继器可以把网络延长到 9 600 m(31 488 ft)。

### 1. PPI 协议

PPI 通信协议是西门子专门为 S7-200 系列 PLC 开发的一个通信协议。主要应用于对 S7-200 的编程、S7-200 之间的通信以及 S7-200 与 HMI 产品的通信。可以通过 PC/PPI 电缆或两芯屏蔽双绞线进行联网。支持的波特率为 9.6 KB/s、19.2 KB/s 和 187.5 KB/s。图 8-7 为 PPI 通信协议网络。

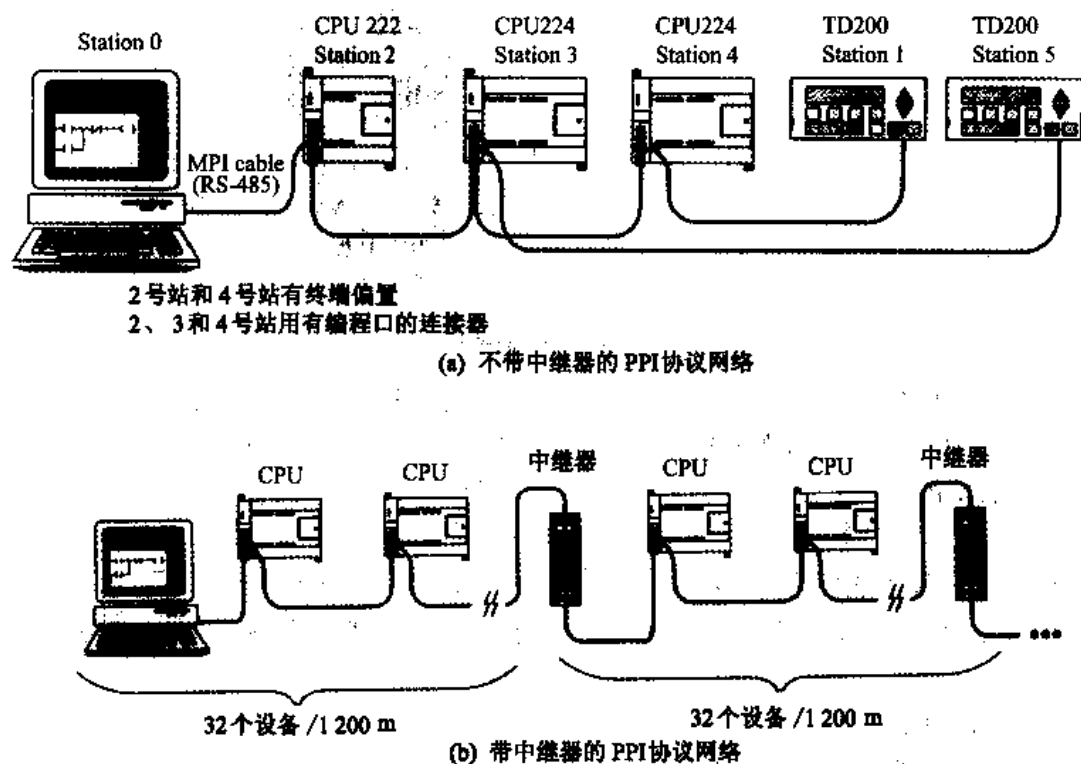


图 8-7 PPI 通信网络

PPI 是一个主/从协议。在这个协议中,S7-200 一般作为从站,自己不发送信息,只有当主站,如西门子编程器、TD 200 等 HMI,给从站发送申请时,从站才进行响应。

如果在用户程序中将 S7-200 设置(由 SMB30 设置)为 PPI 主站模式,则这个 S7-200 CPU 在 RUN 模式下可以作为主站。一旦被设置为 PPI 主站模式,就可以利用网络读

(NETR)和网络写(NETW)指令来读写另外一个 S7-200 中的数据。有关这些指令的详细描述,请参阅 8.3 节的通信指令。当 S7-200 CPU 作为 PPI 主站时,它还可以作为从站响应来自其他主站的申请。

PPI 通信协议是一个令牌传递协议,对于一个从站可以响应多少个主站的通信请求,PPI 协议没有限制,但是在不加中继器的情况下,网络中最多只能有 32 个主站,包括编程器、HMI 产品或被定义为主站的 S7-200。

## 2. MPI 协议

S7-200 可以通过通信接口连接到 MPI 网上,如图 8-8 所示,主要应用于 S7-300/400 CPU 与 S7-200 通信的网络中。应用 MPI 协议组成的网络,通信支持的波特率为 19.2 KB/s 或 187.5 KB/s。通过此协议,实现作为主站的 S7-300/400 CPU 与 S7-200 的通信。在 MPI 网中,S7-200 作为从站,从站之间不能通信,S7-300/400 作为主站,当然主站也可以是编程器或 HMI 产品。

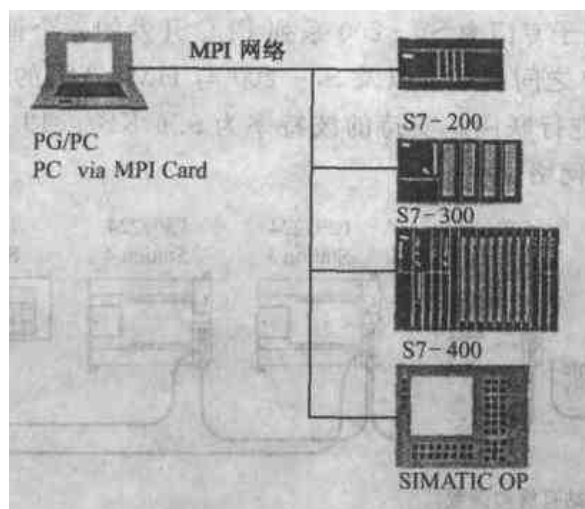


图 8-8 MPI 通信网络

MPI 协议可以是主/主协议或主/从协议,协议如何操作有赖于通信设备的类型。如果是 S7-300/400 CPU 之间通信,那就建立主/主连接,因为所有的 S7-300/400 CPU 在网站中都是主站。如果设备是一个主站与 S7-200 CPU 通信,那么就建立主/从连接,因为 S7-200 CPU 是从站。

应用 MPI 协议组成网络时,在 S7-300 和 S7-400 CPU 的用户程序中可以利用 XGET 和 XPUT 指令来读写 S7-200 的数据(指令的使用方法请参考 S7-300 或 S7-400 编程手册)。

## 3. Profibus 协议

Profibus 协议通常用于实现分布式 I/O 设备(远程 I/O)的高速通信。许多厂家生产类型众多的 Profibus 设备。这些设备包括从简单的输入或输出模块到电机控制器和可编程控制器。S7-200 CPU 可以通过 EM 277 Profibus-DP 扩展模块的方法连接到 Profibus-DP 协议支持的网路中。协议支持的波特率为 9 600 KB/s 到 12 MB/s。

Profibus 网络通常有一个主站和几个 I/O 从站,如图 8-9 所示。主站通过配置可以知道所连接的 I/O 从站的型号和地址。主站初始化网络时核对网络上的从站设备与配置的从站是否匹配。运行时主站可以像操作自己的 I/O 一样对从站进行操作,即不断地把数据写到从

站或从从站读取数据。当 DP 主站成功地配置一个从站时,它就拥有了该从站,如果在网络中有另外一个主站,它只能很有限地访问属于第一个主站的从站数据。

图 8-9 Profibus-DP 网络

#### 4. 用户自定义协议(自由口通信模式)

自由通信口(Freeport Mode)模式是 S7-200 PLC 一个很有特色的功能。S7-200 PLC 的自由口通信,即用户可以通过用户程序对通信口进行操作,自己定义通信协议(例如 ASCII 协议)。应用此种通信方式,使 S7-200 PLC 可以与任何通信协议已知、具有串口的智能设备和控制器(例如打印机、条形码阅读器、调制解调器、变频器、上位 PC 机等)进行通信,当然也可以用于两个 CPU 之间简单的数据交换,如图 8-10 所示。该通信方式使可通信的范围大大增大,使控制系统配置更加灵活、方便。当连接的智能设备具有 RS-485 接口时,可以通过双绞线进行连接;如果连接的智能设备具有 RS-232 接口时,可以通过 PC/PPI 电缆连接起来进行自由口通信。此时通信支持的波特率为 1.2~115.2 KB/s。

图 8-10 自由通信口方式与外设的连接

在自由口通信模式下,通信协议完全由用户程序控制。通过设定特殊存储字节 SMB30(端口 0)或 SMB130(端口 1)允许自由口模式,用户程序可以通过使用发送中断、接收中断、发送指令(XMT)和接收指令(RCV)对通信口操作。应注意的是,只有在 CPU 处于 RUN 模式时才能允许自由口模式,此时编程器无法与 S7-200 进行通信。当 CPU 处于 STOP 模式时,

自由口模式通信停止,通信模式自动转换成正常的 PPI 协议模式,编程器与 S7-200 恢复正常的通信。有关发送和接收指令的使用请参阅 8.3 节的说明。

### 5. USS 协议

USS 协议是西门子传动产品(变频器等)通信的一种协议,S7-200 提供 USS 协议的指令,用户使用这些指令可以方便地实现对变频器的控制。通过串行 USS 总线最多可接 30 台变频器(从站),然后用一个主站(PC,西门子 PLC)进行控制,包括变频器的启/停、频率设定、参数修改等操作,总线上的每个传动装置都有一个从站号(在传动设备的参数中设定),主站依靠此从站号识别每个传动装置。USS 协议是一种主-从总线结构,从站只是对主站发来的报文做出回应并发送报文。另外也可以是一种广播通信方式,一个报文同时发给所有 USS 总线传动设备。

## 8.2.3 网络配置实例

### 1. 单主站的 PPI 网络

编程设备通过 PC/PPI 电缆或通信卡(如 CP5611 等)与 S7-200 通信,完成对 S7-200 的编程、监控等操作,如图 8-11(a)所示;HMI 产品(如 TD200、TP 或 OP)通过标准 RS-485 电缆与 S7-200 通信,如图 8-11(b)所示,都是应用 PPI 协议组成的网络,而且图 8-11 中所示的两个网络中都是只有单一的主站,如编程设备(STEP7-Micro/WIN)、HMI 产品,在这两个网络中 S7-200 都是从站,只响应来自主站的请求。

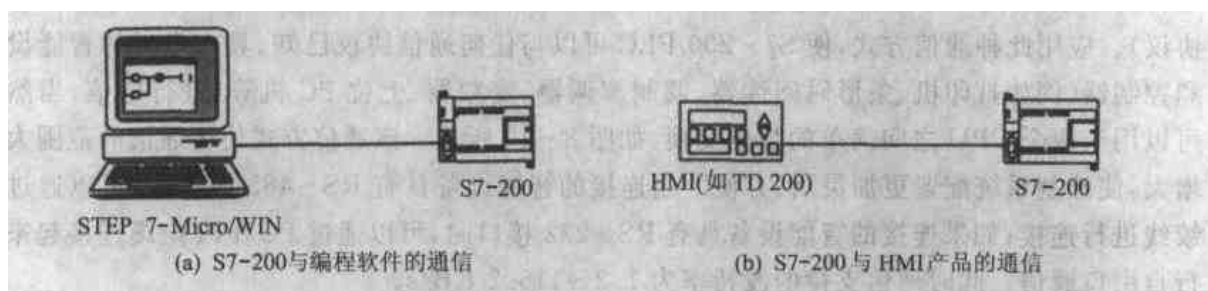


图 8-11 单主站的 PPI 网络

### 2. 多主站的 PPI 网络

图 8-12 所示为网络中有多主站的网络实例,编程设备通过 PC/PPI 电缆或通信卡与 S7-200 连接,HMI 产品与 S7-200 通过网络连接器及双绞线连接,网络应用 PPI 协议进行通信。



图 8-12 多主站的 PPI 网络

在网络中 S7-200 作为从站响应网络中所有主站的通信请求,任意主站均可以读写 S7-200 中的数据。如果一个 S7-200 在用户程序中被定义为 PPI 主站模式,则这个 S7-200 就可以应用网络读(NETR)和网络写(NETW)指令读写另外作为从站的 S7-200 中的数据,但与网络中其他主站(编程器或 HMI)通信时还是作为从站,即此时只能响应主站请求,不能发出请求。

因为 PPI 协议是一种主从通信协议,所以在网络中的多个主站之间不能相互通信。

### 3. 使用 S7-200、S7-300 和 S7-400 设备组成的 MPI 网络

图 8-13 所示是应用 MPI 协议组成的网络的实例,在网络中有多个主站,主站包括编程设备、S7-300(或 S7-400)以及 HMI 产品,又有从站 S7-200。网络通过通信卡(或 PC Adapter)、网络连接器和双绞线连接。在这种网络中 S7-200 只能作为从站,主站 S7-300 用 XGET 和 XPUT 指令实现对从站 S7-200 的读写操作,而且 S7-200 不能被定义为 PPI 主站模式。

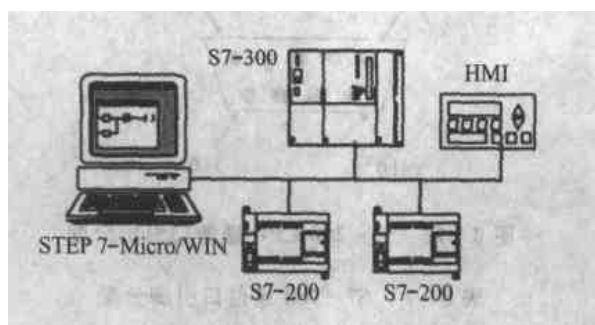


图 8-13 MPI 网络

MPI 是一种允许主-主通信和主-从通信的协议,所以作为主站的 S7-300、S7-400 之间也可以通信。

### 4. Profibus 网络配置

在这种网络中,S7-200 作为 S7-315-2DP 的一个从站,通过特殊扩展模块 EM277 连接到 Profibus 网络中,如图 8-14 所示。S7-315-2DP(一种具有一个 MPI 通信口和一个 Profibus-DP 通信口的 S7-300 CPU)作为主站。对从站 ET200,自己没有用户程序,其 I/O

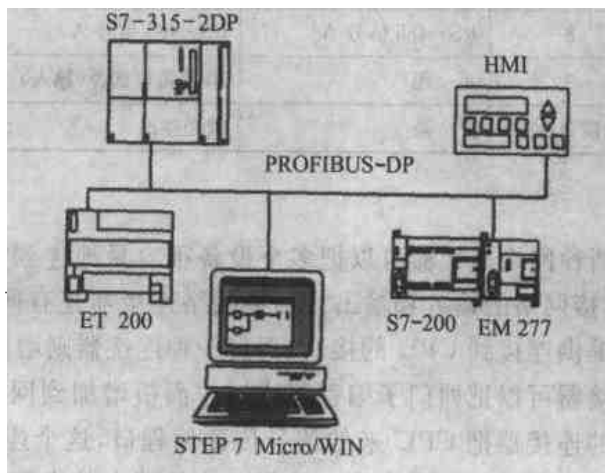


图 8-14 Profibus 网络

点直接作为主站的 I/O 点由主站直接进行读写操作,而且主站在网络配置时就将 ET200 的 I/O 点与主站本身的 I/O 点一起编址;对从站 S7-200 与主站的通信,是主站通过 EM277 来读写 S7-200 的 V 存储器来完成,通信的数据量为 1~128 个字节。

### 8.2.4 网络部件

网络部件可以把每个 S7-200 上的通信口连到网络总线。下面介绍通信口、网络总线连接器和用于扩展网络的中继器。

#### 1. 通信口

S7-200 CPU 上的通信口是符合欧洲标准 EN 50170 中 Profibus 标准的 RS-485 兼容 9 针 D 型连接器。图 8-15 是通信接口的物理连接口,表 8-2 给出了通信口插针对应关系的分配表。

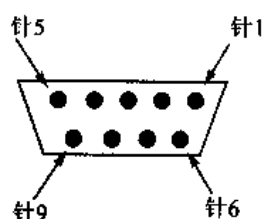


图 8-15 S7-200 CPU 通信口引脚分配

表 8-2 S7-200 通信口引脚分配

针 号	Profibus 名称	端口 0/端口 1
1	屏 蔽	逻辑地
2	24V 返回	逻辑地
3	RS-485 信号 B	RS-485 信号 B
4	发送申请	RTS(TTL)
5	5 V 返回	逻辑地
6	+5 V	+5 V, 100 $\Omega$ 串联电阻
7	+24 V	+24 V
8	RS-485 信号 A	RS-485 信号 A
9	不 用	10 位协议选择(输入)
连接器外壳	屏 蔽	机壳接地

#### 2. 网络连接器

利用西门子提供的两种网络连接器可以把多个设备很容易地连到网络中,两种连接器都有两组螺丝端子,可以连接网络的输入和输出。两种网络连接器还有网络终端匹配(电阻)选择开关。一种连接器仅提供连接到 CPU 的接口,而另一种连接器则增加了一个编程接口。

带有编程接口的连接器可以把西门子编程器或操作面板增加到网络中,而不用改动现有的网络连接。带编程口的连接器把 CPU 来的信号传到编程口,这个连接器对于连接从 CPU 取电源的设备(例如 TD200 或 OP37)很有用。编程口连接器上的电源引针到编程口,而不用另加电源。

### 3. 中继器

西门子提供连接到 Profibus 网络段的网络中继器,利用中继器可以延长网络距离;允许给网络加入设备;并且提供了一个隔离不同网络段的方法。在波特率是 9 600 时,Profibus 允许在一个网络段上最多有 32 个设备,最长距离是 1 200 m(3 936ft),每个中继器允许给网络增加另外的 32 个设备,而且可以把网络再延长 1 200 m(3 936ft)。网络中最多可以使用 9 个中继器,网络总长度可增加至 9 600 m。每个中继器为网络段提供终端匹配开关。

## 8.3 S7-200 通信指令

S7-200 的通信指令包括应用于 PPI 协议网络读写指令、用于自由通信模式的发送和接收指令以及用于控制变频器的 USS 协议指令。

### 8.3.1 网络读/网络写指令

#### 1. 网络读 NETR(Network Read)、网络写 NETW(Network Write)指令

网络读、网络写指令格式如图 8-16 所示。当 S7-200 被定义为 PPI 主站模式时,就可以应用网络读写指令对另外的 S7-200 进行读写操作。

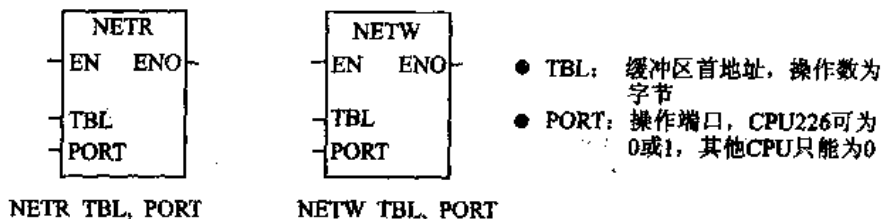


图 8-16 NETR/NETW 指令格式

应用网络读(NETR)通信操作指令,可以通过指令指定的通信端口(PORT)从另外的 S7-200 上接收数据,并将接收到的数据存储在指定的缓冲区表(TBL)中。

应用网络写(NETR)通信操作指令,可以通过指令指定的通信端口(PORT)向另外的 S7-200 写指令指定的缓冲区表(TBL)中的数据。

缓冲区(TBL)参数的定义如图 8-17 所示。

NETR 指令可以从远程站点上读最多 16 个字节的信息,NETW 指令则可以向远程站点写最多 16 个字节的信息。在程序中使用任意多条网络读写指令,但在任何同一时间,最多只能同时执行 8 条 NETR 或 NETW 指令、4 条 NETW 指令和 4 条 NETR 指令,或者 2 条 NETR 指令和 6 条 NETW 指令。

使用网络读写指令对另外的 S7-200 读写操作时,首先要将应用网络读写指令的 S7-200 定义为 PPI 主站模式(SMB30),即通信初始化,然后就可以使用该指令进行读写操作。

#### 2. 应用举例

一条生产线正在灌装黄油桶并将其送到四台包装机(打包机)上包装,打包机把 8 个黄油桶包装到一个纸箱中。一个分流机控制着黄油桶流向各个打包机。4 个 CPU 222 用于控制打包机,一个 CPU 224 安装了 TD200 操作机器人机界面,用于控制分流机。图 8-18 为系统组成示意图。

字节偏移量		7	6	5	4	3	0
字节 0	D	A	E	0	错误码		
字节 1	远程站的地址						
字节 2	远程站的数据指针						
字节 3							
字节 4							
字节 5							
字节 6	数据长度						
字节 7	数据字节 0						
字节 8	数据字节 1						
:	:						
字节 21	数据据字节 14						
字节 22	数据据字节 15						

通信操作的状态信息字节。其中：

D:操作是否完成      0=未完成 1=完成

A:有效(操作已被排队) 0=无效 1=有效

E:操作是否错误      0=无错误 1=错误

远程站的地址:要访问 PLC 的地址

远程站的数据指针:要访问数据的间接指针,  
如(&VB100)

数据长度:要访问的数据字节数

数据区:执行 NETR 后,从远程读到的数据放在这个数据区;

执行 NETW 后,要发送到远程站的数据要放在这个数据区。

图 8-17 网络读/写指令缓冲区参数定义

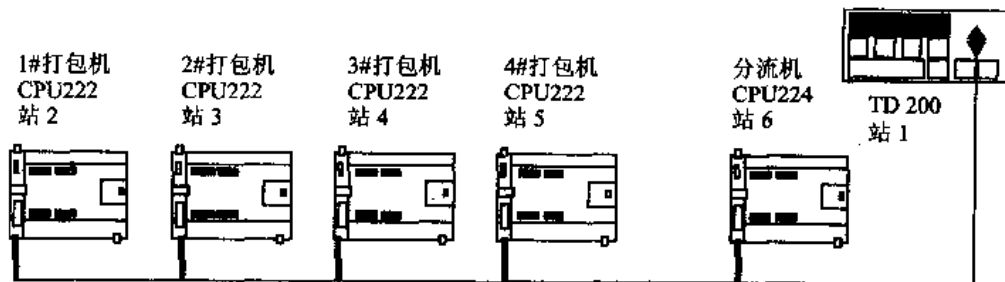


图 8-18 系统组成示意图

分流机对打包机的控制主要是负责将纸箱、粘结剂和黄油桶分配给不同的打包机,而分配的依据就是各个打包机的工作状态,因此分流机要实时地知道各个打包机的工作状态,另外,为了统计的方便各个打包机打包完成的数量应上传至分流机,以便记录和通过 TD200 查阅。

四个打包机(CPU 222)的站地址分别为 2、3、4 和 5,分流机(CPU 224)的站地址为 6,TD200 的站地址为 1,将各个 CPU 的站地址在系统块中设定好,随程序一块下载到 PLC 中,TD200 的地址在 TD200 中直接设定。

在这个例子中,6#站分流机的程序应包括控制程序、与 TD200 的通信程序以及与其他站的通信程序,其他站只有控制程序,下面给出的只是 6#站与其他站的通信程序,其他程序可根据控制要求编写。

在网络连接中 6#站所用的网络连接器带编程口,以便连接 TD200 和其他站,其他站用不带编程口的网络连接器。

假设各个打包机的工作状态存储在各自 CPU 的 VB100 中,其中:

V100.7 为打包机检测到错误;

V100.6~V100.4 为打包机错误代码;

V100.2 为粘结剂缺的标志,应增加粘结剂;

V100.1 为纸箱缺的标志,应增加纸箱;

V100.0 为没有可包装黄油桶的标志。

各个打包机已经完成的打包箱数分别存储在各自 CPU 的 VW101 中。

我们定义 6 号站分流机对各打包机接收和发送的缓冲区的起始地址分别为: VB200、VB210、VB220、VB230 和 VB300、VB310、VB320、VB330。

分流机读/写 1 号打包机(2 号站)的工作状态和完成打包数量的程序清单如图 8-19 所示。

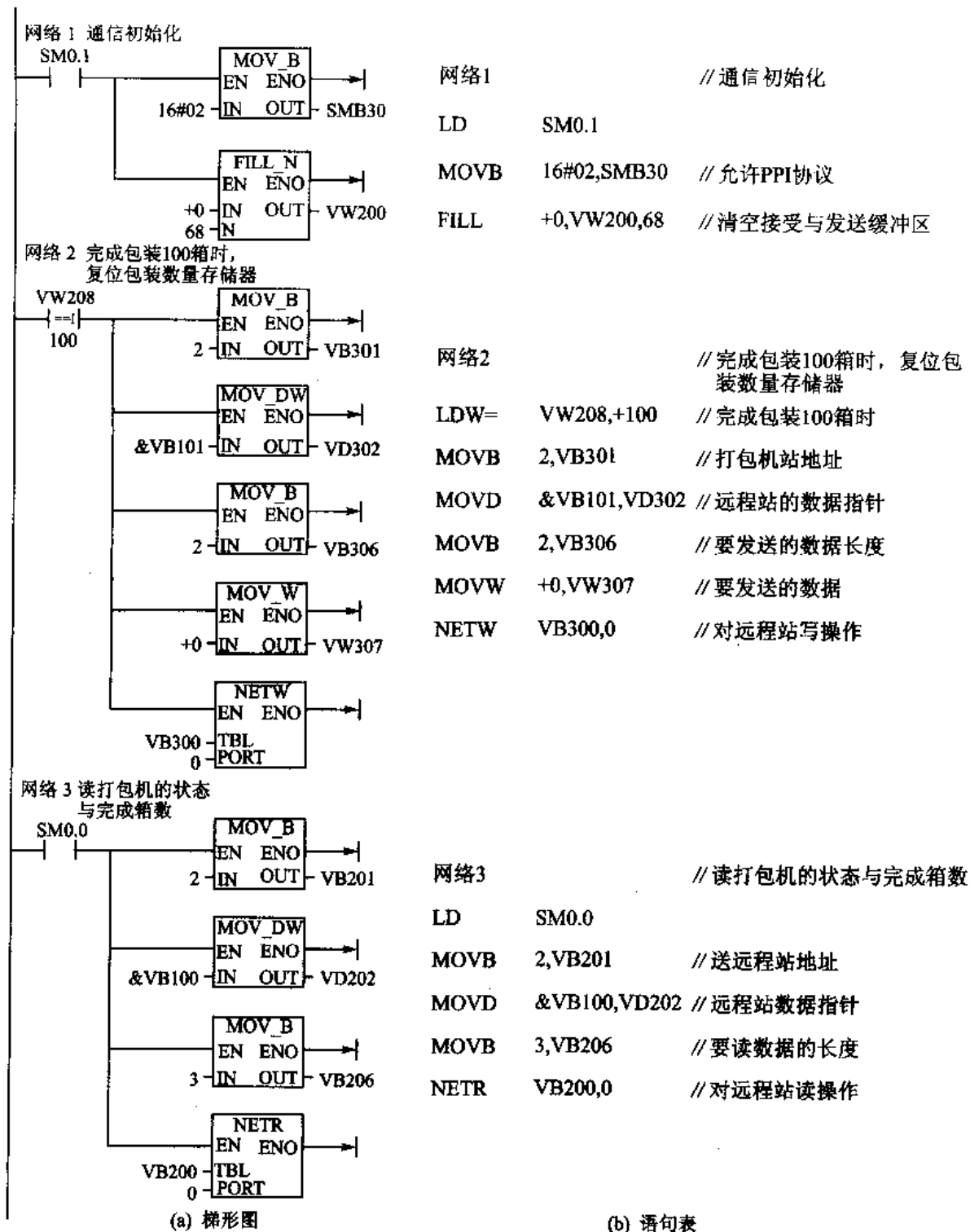


图 8-19 网络读写指令应用举例程序图

对其他站的读写操作程序只需将站地址号与缓冲区指针作相应的改变即可。

### 8.3.2 发送与接收指令

#### 1. XMT(Transmit)/RCV(Receive)发送与接收指令

XMT/RCV 指令格式如图 8-20 所示,XMT/RCV 指令用于当 S7-200 被定义为自由端口通信模式时,由通信端口发送或接收数据。

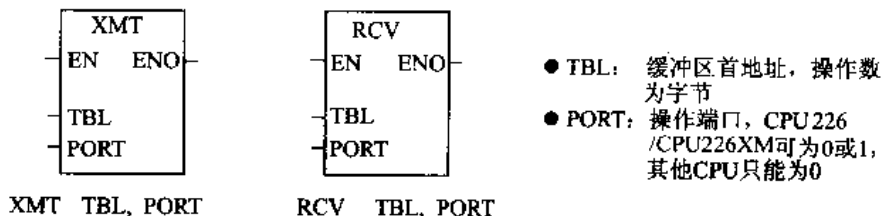


图 8-20 XMT/RCV 指令格式

应用发送指令(XMT),可以将发送数据缓冲区(TBL)中的数据通过指令指定的通信端口(PORT)发送出去,发送完成时将产生一个中断事件,数据缓冲区的第一个数据指明了要发送的字节数。

应用接收指令(RCV),可以通过指令指定的通信指定端口(PORT)接收信息并存储于接收数据缓冲区(TBL)中,接收完成也将产生一个中断事件,数据缓冲区的第一个数据指明了接收的字节数。

#### 2. 对自由端口模式的解释

CPU 的串行通信口可由用户程序控制,这种操作模式称为自由端口模式。当选择了自由端口模式时,用户程序可以使用接收中断、发送中断、发送指令(XMT)和接收指令(RCV)来进行通信操作。在自由端口模式下,通信协议完全由用户程序控制。SMB30(用于端口 0)和 SMB130(如果 CPU 有两个端口,则用于端口 1)用于选择波特率、奇偶校验、数据位数和通信协议。

只有 CPU 处于 RUN 模式时,才能进行自由端口通信。通过向 SMB30(端口 0)或 SMB130(端口 1)的协议选择区置 1,可以允许自由端口模式。处于自由端口模式时,PPI 通信被禁止,此时不能与编程设备通信(如使用编程设备对程序状态监视或对 CPU 进行操作)。在一般情况下,可以用发送指令(XMT)向打印机或显示器发送信息,其他的如同条码阅读器、重量计和焊机等的连接,在这种情况下,用户都必须编写用户程序,以支持自由端口模式下设备同 CPU 通信的协议。

当 CPU 处于 STOP 模式,自由端口模式被禁止,通信口自动切换为 PPI 协议的操作,重新建立与编程设备的正常通信。

注意:可以用反映 CPU 工作方式的模式开关的当前位置的特殊存储器 SM0.7 来控制自由端口模式的进入。当 SM0.7 为 0 时,模式开关处于 TREM 位置;当 SM0.7 为 1 时,模式开关处于 RUN 位置。只有模式开关位于 RUN 位置,才允许自由端口模式,为了使用编程设备对程序状态监视或对 CPU 进行操作,可以把模式开关改变到任何其他位置(如 STOP 或 TERM 位置)。

#### 3. 端口的初始化与控制字节

SMB30 和 SMB130 分别配置通信端口 0 和 1,为自由端口通信选择波特率、奇偶校验和数据位数。自由端口的控制字节定义如表 8-3 所列。

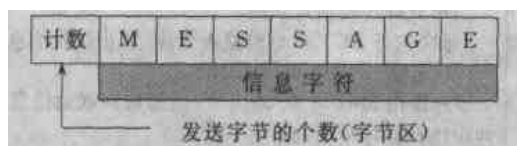
表 8-3 特殊存储器位 SMB30 和 SMB130

端口 0	端口 1	描述自由口模式控制字节
SMB30 格式	SMB130 格式	<div> <div>MSB</div> <div>LSB</div> <div> <div>P</div> <div>P</div> <div>D</div> <div>B</div> <div>B</div> <div>B</div> <div>M</div> <div>M</div> </div> </div>
SM30.6 和 SM30.7	SM130.6 和 SM130.7	PP: 校验选择 00=无奇偶校验;01=偶校验;10=无奇偶校验; 11=奇校验
SM30.5	SM130.5	D: 每个字符的数据位 0=每个字符 8 位;1=每个字符 7 位
SM30.2 到 SM30.4	SM130.2 到 SM130.4	BBB: 自由口波特率 000=38 400 波特;001=19 200 波特; 010=9 600 波特; 011=4 800 波特;100=2 400 波特; 101=1 200 波特; 110=115.2 K 波特;111=57.6 K 波特
SM30.0 和 SM30.1	SM130.0 和 SM130.1	MM: 协议选择 00=PPI/从站模式(默认设置);01=自由口协议 10=PPI/主站模式;11=保留

#### 4. 用 XMT 指令发送数据

用 XMT 指令可以方便地发送一个或多个字节缓冲区的内容,最多为 255 个字节。XMT 缓冲区的数据格式如表 8-4 所列。

表 8-4 XMT 缓冲区的格式



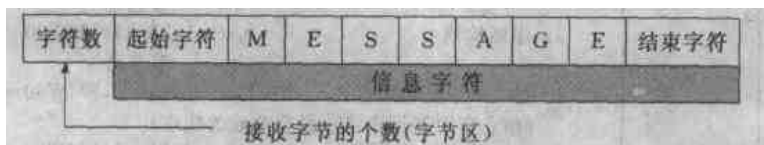
如果有一个中断服务程序连接到发送结束事件上,在发完缓冲区中的最后一个字符时,则会产生一个中断(对端口 0 为中断事件 9,对端口 1 为中断事件 26)。当然也可以不用中断来判断发送指令(如向打印机发送信息)是否完成,而是监视 SM4.5 或 SM4.6 的状态,以此来判断发送是否完成。

如果把发送字符数设置为 0,然后执行 XMT 指令,可以产生一个中断(BREAK)事件。发送 BREAK 的操作和发送任何其他信息的操作是一样的,当发送 BREAK 完成时,产生一个 XMT 中断,并且 SM4.5 或 SM4.6 反映了发送操作的当前状态。

#### 5. 用 RCV 指令接收数据

用 RCV 接收指令可以方便地接收一个或多个字节缓冲区的内容,最多为 255 个字节,这些字符存储在接收缓冲区中,接收缓冲区的格式如表 8-5 所列。

表 8-5 RCV 缓冲区格式



如果有一个中断程序连接到接收完成事件上,在接收到缓冲区中的最后一个字符时,则会产生一个中断(对端口 0 为中断事件 23,对端口 1 为中断事件 24)。当然也可以不使用中断,而是通过监视 SMB86(对端口 0)或 SMB186(对端口 1)状态的变化,进行接收信息状态的判断。

当接收指令没有被激活或接收已经结束时, SMB86 或 SMB186 为 1; 当正在接收时, 它们为 0。

使用接收指令时, 允许用户选择信息接收开始和信息接收结束的条件。如表 8-6 所列, 用 SMB86~SMB94 对端口 0 进行设置, 用 SMB186~SMB194 对端口 1 进行设置。应该注意的是, 当接收信息缓冲区超界或奇偶校验错误时, 接收信息功能会自动终止。所以必须为接收信息功能操作定义一个启动条件和一个结束条件。接收指令支持的启动条件有: 空闲线检测、起始字符检测、空闲线和起始字符检测、断点检测、断点和起始字符检测和任意字符检测。支持的结束信息的方式有: 结束字符检测、字符间隔定时器、信息定时器、最大字符记数、校验错误、用户结束或以上几种结束方式的组合。

表 8-6 特殊存储器字节 SMB86 到 SMB94, SMB186 到 SMB194

端口 0	端口 1	描 述									
SMB86	SMB186	<p>接收信息状态字节</p> <p>7 0</p> <table border="1"> <tr> <td>n</td> <td>r</td> <td>e</td> <td></td> <td></td> <td></td> <td>t</td> <td>c</td> <td>p</td> </tr> </table> <p>n: 1=用户通过禁止命令结束接收信息  r: 1=接收信息结束: 输入参数错误或缺少起始和结束条件  e: 1=收到结束字符  t: 1=接收信息结束: 超时  c: 1=接收信息结束: 字符数超长  p: 1=接收信息结束: 奇偶校验错误</p>	n	r	e				t	c	p
n	r	e				t	c	p			
SMB87	SMB187	<p>接收信息控制字节</p> <p>7 0</p> <table border="1"> <tr> <td>en</td> <td>sc</td> <td>ec</td> <td>il</td> <td>c/m</td> <td>tmr</td> <td>bk</td> <td></td> </tr> </table> <p>en: 0=禁止接收信息功能  1=允许接收信息功能  每次执行 RCV 指令时检查允许/禁止接收信息位  sc: 0=忽略 SMB88 或 SMB188  1=使用 SMB88 或 SMB188 的值检测起始信息  ec: 0=忽略 SMB89 或 SMB189  1=使用 SMB89 或 SMB189 的值检测结束信息  il: 0=忽略 SMB90 或 SMB190  1=使用 SMB90 或 SMB190 值检测空闲状态  c/m: 0=定时器是内部字符定时器  1=定时器是信息定时器  tmr: 0=忽略 SMW92 或 SMW192  1=当执行 SMW92 或 SMW192 时终止接收  bk: 0=忽略中断条件  1=使用中断条件来检测起始信息  信息的中断控制字节位用来定义识别信息的标准。信息的起始和结束需定义  起始信息 = <math>il * sc + bk * sc</math>  结束信息 = <math>ec + tmr + \text{最大字符数}</math>  起始信息编程:  1. 空闲检测: <math>il=1, sc=0, bk=0, SMW90&gt;0</math>  2. 起始字符检测: <math>il=0, sc=1, bk=0, SMW90</math>  被忽略  3. 中断检测: <math>il=0, sc=1, bk=1, SMW90</math>  被忽略  4. 对一个信息的响应: <math>il=1, sc=0, bk=0, SMW90=0</math>  (信息定时器用来终止没有响应的接收)  5. 中断一个起始字符: <math>il=0, sc=1, bk=1, SMW90</math>  被忽略  6. 空闲和一个起始字符: <math>il=1, sc=1, bk=0, SMW90&gt;0</math>  7. 空闲和起始字符(非法): <math>il=1, sc=1, bk=0, SMW90=0</math>  注意: 通过超时和奇偶校验错误(如果允许), 可以自动结束接收过程</p>	en	sc	ec	il	c/m	tmr	bk		
en	sc	ec	il	c/m	tmr	bk					

续表 8-6

端口 0	端口 1	描 述
SMB88	SMB188	信息字符的开始
SMB89	SMB189	信息字符的结束
SMB90 SMB91	SMB190 SMB191	空闲线时间段按毫秒设定。空闲线时间溢出后接收的第一个字符是新的信息的开始字符。SMB90(或 SMB190)是最高有效字节,SMB91(或 SMB191)是最低有效字节
SMB92 SMB93	SMB192 SMB193	中间字符/信息计时器溢出值按毫秒设定。如果超过这个时间段,则终止接收信息。SMB92(或 SMB192)是最高有效字节,SMB93(或 SMB193)是最低有效字节
SMB94	SMB194	要接收的最大字符数(1 到 255 字节) 注:这个范围必须设置到所希望的最大缓冲区大小,即使信息的字符数始终达不到

### 6. 使用字符中断控制接收数据

为了完全适应对各种通信协议的支持,可以使用字符中断控制的方式来接收数据。每接收一个字符时都会产生中断。在执行连接到接收字符中断事件上的中断程序前,接收到的字符存储在 SMB2 中,校验状态(如果允许的话)存储在 SM3.0 中。

SMB2 是自由端口接收字符缓冲区。在自由端口模式下,每一个接收到的字符都会被存储在这个单元中,以方便用户程序访问。

SMB3 用于自由端口模式,并包含一个校验错标志位。当接收字符的同时检测到校验错误时,SM3.0 被置位,该字节的所有其他位保留。用该位丢弃本信息或产生对本信息的否定确认。

注意: SMB2 和 SMB3 是端口 0 和端口 1 共用的。当接收的字符来自端口 0 时,执行与事件(中断事件 8)相连接的中断程序,此时 SMB2 中存储从端口 0 接收的字符,SMB3 中存储该字符的校验状态;当接收的字符来自端口 1,执行与事件(中断事件 25)相连接的中断程序,此时 SMB2 中存储从端口 1 接收的字符,SMB3 中存储该字符的校验状态。

### 7. 指令应用举例

本程序功能为上位 PC 机和 PLC 之间的通信,PLC 接收上位 PC 发送的一串字符,直到接收到回车符为止,PLC 又将信息发送回 PC 机。

程序清单如图 8-21 所示。

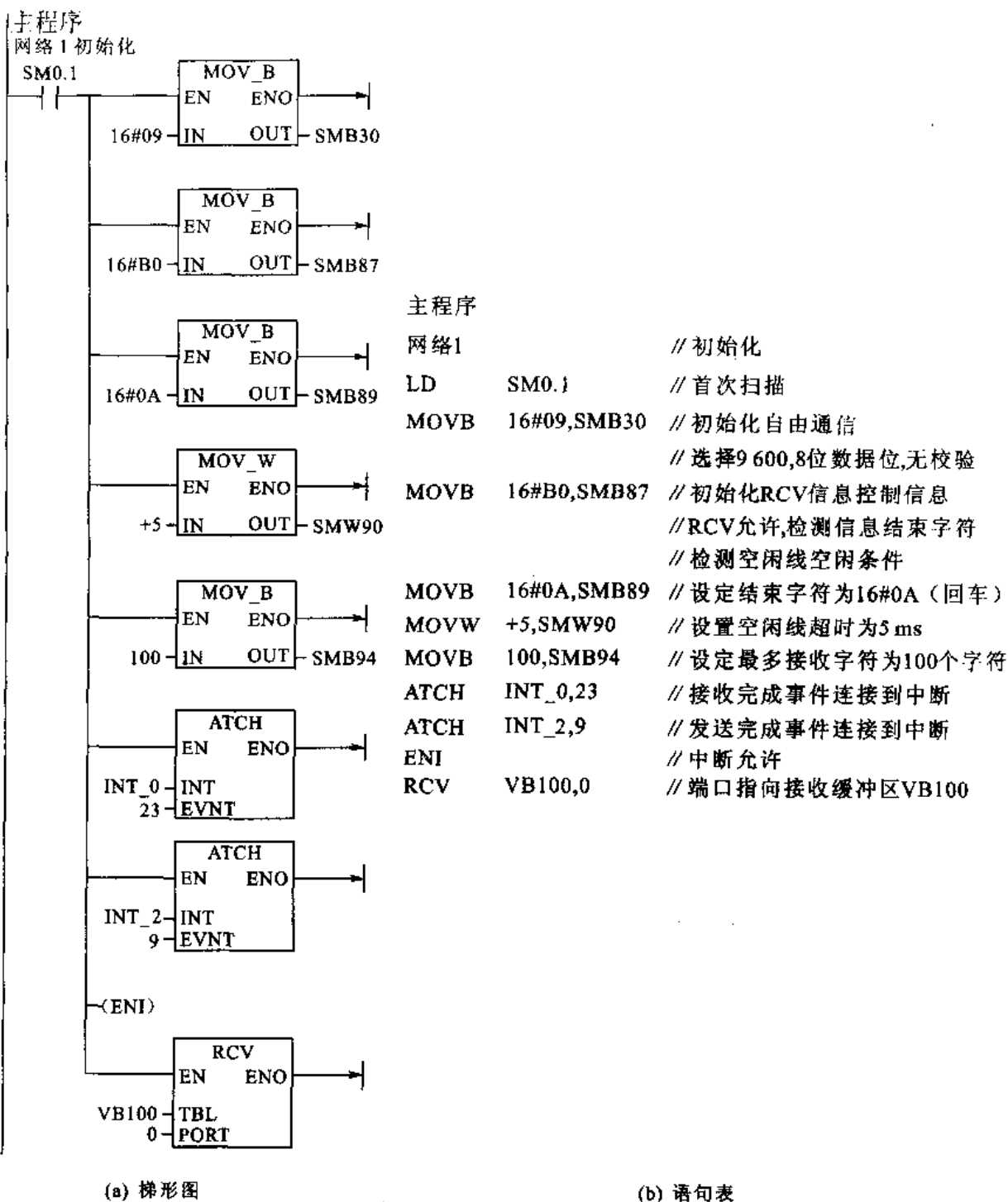


图 8-21 自由通信应用举例

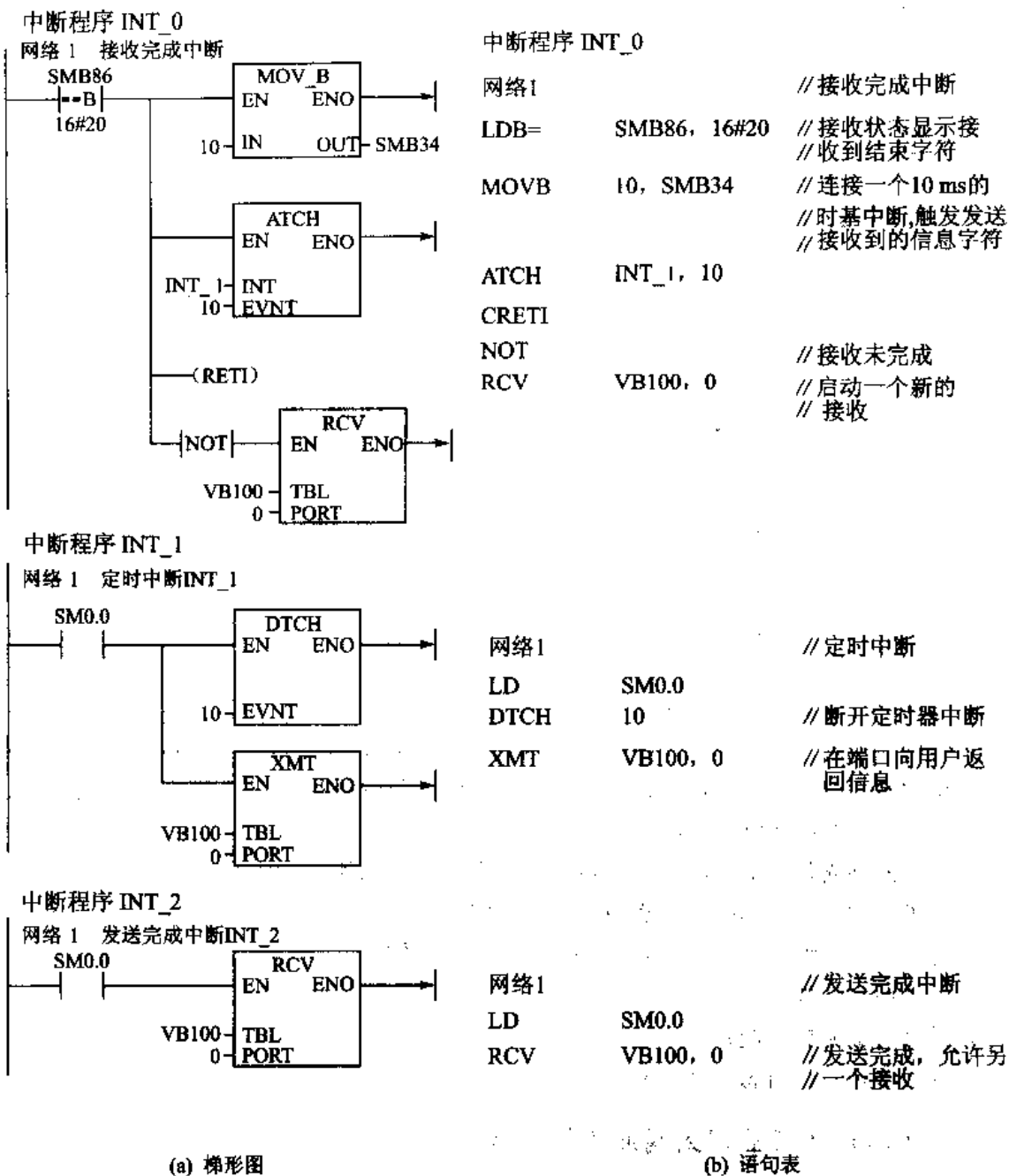


图 8-21 自由通信应用举例(续)

### 8.3.3 USS 通信指令

USS 通信指令用于 PLC 与变频器等驱动设备的通信及控制。

将 USS 通信指令置于用户程序中, 经编译后自动地将一个或多个子程序和 3 个中断程序添加到用户程序中。另外用户需要将一个 V 存储器地址分配给 USS 全局变量表的第一个存储单元, 从这个地址开始, 以后连续的 400 个字节的 V 存储器将被 USS 指令使用, 不能用作它用。

当使用 USS 指令进行通信时, 只能使用通信口 0, 而且 0 口不能用作它用, 包括与编程设

备的通信或自由通信。

使用 USS 指令对变频器进行控制时,变频器的参数应做适当的设定,USS 通信指令包括:

- USS\_INIT 初始化指令;
- USS\_CTRL 控制变频器指令;
- USS\_RPM\_W(D,R)读无符号字类型(双字类型、实数类型)参数指令;
- USS\_WPM\_W(D,R)写无符号字类型(双字类型、实数类型)参数指令。

限于篇幅,具体的变频器参数设定和 USS 指令的详细使用请参考变频器的操作手册和 S7-200 系统手册。

## 8.4 S7-200 的通信扩展模块

### 8.4.1 EM241 调制解调器模块

使用 EM241 调制解调器模块可以将 S7-200 直接连接到一个模拟电话线上,此时 S7-200 就具有了电话机所具有的部分功能。

该模块的主要特点和功能有:

- 提供标准的国际电话线接口,与电话线连接;
- 提供与支持 STEP7-Micro/WIN,通过调制解调器接口,连接到具有 EM241 扩展模块的 S7-200 上,实现对 S7-200 的编程和远程诊断;
- 支持 Modbus RTU 协议;
- 提供向预先设定的寻呼机发送数字或文本信息的功能;
- 提供向预先设定的手机发送短信息的功能;
- 允许 CPU 到 CPU 或 CPU 到 Modbus 的数据传送;
- 密码保护功能;
- 提供安全回拨功能;
- 调制解调器的组态存储在 CPU。

### 8.4.2 CP243-1 工业以太网通信处理器模块

CP243-1 是一种通信处理器,它可以将 S7-200 系统连接到工业以太网(IE)中。CP243-1 还可用于实现 S7-200 低端性能产品的以太网通信。因此,一台 S7-200 还可通过以太网与其他 S7-200、S7-300 或 S7-400 控制器进行通信,并可与基于 OPC 的服务器进行通信。

在开放的 SIMATIC NET 通信系统中,工业以太网可以用作协调级和单元级网络。在技术上,工业以太网是一种基于屏蔽同轴电缆、双绞线而建立的电气网络,或一种基于光纤电缆的光网络。

## 本章小结

建立在通信基础上的工厂自动化网络系统是目前工厂中常见的 PLC 应用形式。本章简要介绍了通信及工业网络的基本知识及实现方法。重点掌握通信参数设置方法、网络读及网络写指令、自由通信指令及 USS 指令的应用。

(1) 通信的基本方式有并行通信和串行通信两种,这两种通信方式有各自的优缺点和适用领域。串行通信又分为同步和异步两种,串行异步通信的双方必须建立两项约定:传送字符格式和波特率。

(2) 网络大致分为单级网络和多级网络。国际标准化组织对企业自动化系统初步建立一个 6 级的金字塔结构模型。通信协议分为两大类:通用协议和公司专用协议。

(3) 西门子网络金字塔一般由 4 个级别、3 级总线复合而成。通用协议采用 Ethernet 协议,专用协议可以是 PPI 协议、MPI 协议、Profibus 协议和自由口协议,通信类型可以是单主站型或多主站型。

(4) 网络通信中,主站与从站之间的数据以数据表的格式进行传送。网络指令有网络读指令和网络写指令。

(5) 当主机处于 RUN 方式时,可使用自由通信,可以设定波特率等通信参数;当主机切换为 STOP 方式时,自由口通信方式被自动终止。自由口通信用到内部普通寄存器和专用寄存器,通过数据缓冲区接收和发送一个或多个数据信息,利用中断进行控制。自由通信中用到的指令有自由口发送指令和自由口接收指令。

## 思考题与练习题

1. 网络通信时数据传输的方式有哪几种?它们各有什么特点?
2. 西门子 S7-200 PLC 支持的通信协议有哪几种?各有什么特点?
3. 如何理解自由口通信的功能?
4. 参照图 8-14,编写分流机读写 2# 打包机(站 3)的工作状态和完成打包数量的程序。
5. 利用自由口通信的功能和指令,设计一个计算机与 PLC 通信程序,要求上位计算机能够对 S7-200 PLC 中的 VB100~VB107 中的数据进行读写操作。(提示:在编制程序前,应首先指定通信的帧格式,包括起始符、目标地址、操作种类、数据区、停止符等的顺序和字节数;当 PLC 收到信息后,应根据指定好的帧格式进行解码分析,然后再根据要求做出响应。)

## 第九章 现代 PLC 控制系统综合设计实例

### 9.1 PLC 控制系统设计步骤及内容

学习了 PLC 的硬件系统、指令系统和编程方法以后,对设计一个较大的 PLC 控制系统时,要全面考虑许多因素,不管所设计的控制系统的大小,一般都要按图 9-1 所示的设计步骤进行系统设计。

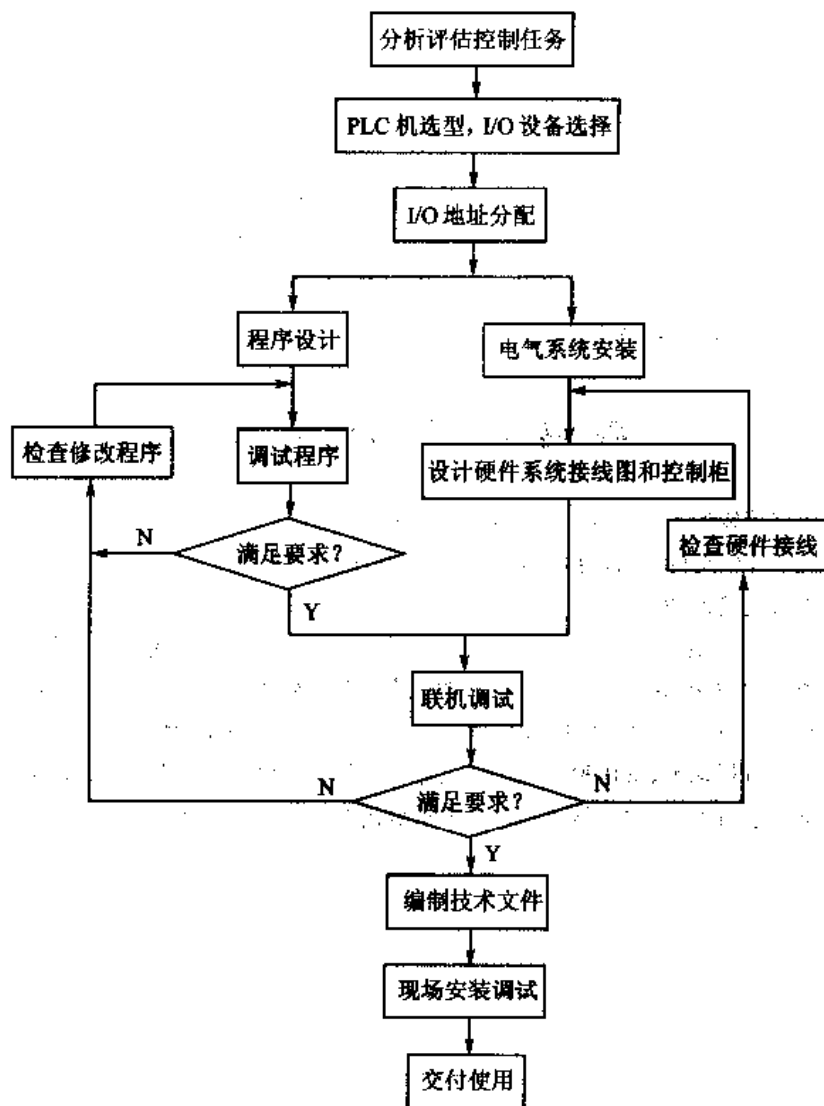


图 9-1 PLC 控制系统设计步骤

### 9.1.1 分析评估及控制任务

随着 PLC 功能的不断提高和完善, PLC 几乎可以完成工业控制领域的各项任务。但 PLC 还是有它最合适的场合, 所以在接到一个控制任务后, 要分析被控对象的控制过程和要求, 看看用什么控制装备 (PLC、单片机、DCS 或 IPC) 来完成该任务最合适。比如仪器及仪表装置、家电的控制器就要用单片机来做; 大型的过程控制系统大部分要用 DCS 来完成。而 PLC 最适合的控制对象是: 工业环境较差, 而对安全性、可靠性要求较高, 系统工艺复杂, 输入/输出以开关量为主的工业自控系统或装置。其实, 现在的可编程序控制器不仅处理开关量, 而且对模拟量的处理能力也很强。所以在很多情况下, 也可取代工业控制计算机 (IPC) 作为主控制器, 来完成复杂的工业自动控制任务。

控制对象及控制装置 (选定为 PLC) 确定后, 还要进一步确定 PLC 的控制范围。一般来说, 能够反映生产过程的运行情况, 能用传感器进行直接测量的参数, 控制逻辑复杂的部分都由 PLC 完成。另外, 如紧急停车等环节, 对主要控制对象还要加上手动控制功能, 这就需要在设计电气系统原理图与编程时统一考虑。

### 9.1.2 PLC 的选型

当某一个控制任务决定由 PLC 来完成, 选择 PLC 就成为最重要的事情。一方面是选择多大容量的 PLC, 另一方面是选择什么公司的 PLC 及外设。

对第一个问题, 首先要对控制任务进行详细的分析, 把所有的 I/O 点找出来, 包括开关量 I/O 和模拟量 I/O 以及这些 I/O 点的性质。I/O 点的性质主要指它们是直流信号还是交流信号, 它们的电源电压, 以及输出是用继电器型还是晶体管或是可控硅型。控制系统输出点的类型非常关键, 如果它们之中既有交流 220 V 的接触器、电磁阀, 又有直流 24 V 的指示灯, 则最后选用的 PLC 的输出点数有可能大于实际点数。因为 PLC 的输出点一般是几个一组共用一个公共端, 这一组输出只能有一种电源的种类和等级。所以一旦它们是交流 220 V 的负载使用, 则直流 24 V 的负载只能使用其他组的输出端了。这样有可能造成输出点数的浪费, 增加成本。所以要尽可能选择相同等级和种类的负载, 比如使用交流 220 V 的指示灯等。一般情况下继电器输出的 PLC 使用最多, 但对于要求高速输出的情况, 如运动控制时的高速脉冲输出, 就要使用无触点的晶体管输出的 PLC 了。知道这些以后, 就可以定下选用多少点和 I/O 是什么类型的 PLC 了。

对第二个问题, 则有以下几个方面要考虑:

① 功能方面 所有 PLC 一般都具有常规的功能, 但对某些特殊要求, 就要知道所选用的 PLC 是否有能力完成控制任务。如对 PLC 与 PLC、PLC 与智能仪表及上位机之间有灵活方便的通信要求; 或对 PLC 的计算速度、用户程序容量等有特殊要求; 或对 PLC 的位置控制有特殊要求等。这就要求用户对市场上流行的 PLC 品种有一个详细的了解, 以便做出正确的选择。

② 价格方面 不同厂家的 PLC 产品价格相差很大, 有些功能类似、质量相当、I/O 点数相当的 PLC 的价格能相差 40 % 以上。在使用 PLC 较多的情况下, 这样的差价当然是必须考虑的因素。

③ 个人喜好方面 有些工程技术人员对某种品牌的 PLC 熟悉, 所以一般比较喜欢使用

这种产品。另外,甚至一些政治因素或个人情感有时也会成为选择的理由。

PLC 主机选定后,如果控制系统需要,则相应的配套模块也就选定了。如模拟量单元、显示设定单元、位置控制单元或热电偶单元等。

### 9.1.3 I/O 地址分配

输入/输出信号在 PLC 接线端子上的地址分配是进行 PLC 控制系统设计的基础。对软件设计来说,I/O 地址分配以后才可进行编程;对控制柜及 PLC 的外围接线来说,只有 I/O 地址确定以后,才可以绘制电气接线图、装配图,让装配人员根据线路图和安装图安装控制柜。分配输出点地址时,要注意 9.1.2 节中所说的负载类型的问题。

在进行 I/O 地址分配时最好把 I/O 点的名称、代码和地址以表格的形式列写出来。

### 9.1.4 系统设计

系统设计包括硬件系统设计和软件系统设计。硬件系统设计主要包括 PLC 及外围线路的设计、电气线路的设计和抗干扰措施的设计等。软件系统设计主要指编制 PLC 控制程序。

选定 PLC 及其扩展模块(如需要的话)和分配完 I/O 地址后,硬件设计的主要内容就是电气控制系统原理图的设计,电气控制元器件的选择和控制柜的设计。电气控制系统原理图包括主电路和控制电路。控制电路中包括 PLC 的 I/O 接线和自动部分、手动部分的详细连接等,有时还要在电气原理图中标上器件代号或另外配上安装图、端子接线图等,以方便控制柜的安装。电气元器件的选择主要是根据控制要求选择按钮、开关、传感器、保护电器、接触器、指示灯和电磁阀等。

控制系统软件设计的难易程度因控制任务而异,也因人而异。对经验丰富的工程技术人员来说,在长时间的专业工作中,受到过各种各样的磨练,积累了许多经验,除了一般的编程方法外,更有自己的编程技巧和方法。但不管怎么说,平时多注意积累和总结是很重要的。

在程序设计时,除 I/O 地址列表外,有时还要把在程序中用到的中间继电器(M)、定时器(T)、计数器(C)和存储单元(V)以及它们的作用或功能列写出来,以便编写程序和阅读程序。

在编程语言的选择上,用梯形图编程还是用语句表编程或使用功能图编程,这主要取决于以下几点:

- (1) 有些 PLC 使用梯形图编程不是很方便(例如书写不便),则可用语句表编程,但梯形图总比语句表直观。
- (2) 经验丰富的人员可用语句表直接编程,就像使用汇编语言一样。
- (3) 如果是清晰的单顺序、选择顺序或并发顺序的控制任务,则最好是用功能图来设计程序。

软件设计和硬件安装可同时进行,这样做可以缩短工期。

### 9.1.5 系统调试

系统调试分模拟调试和联机调试。

硬件部分的模拟调试可在断开主电路的情况下,主要试一试手动控制部分是否正确。

软件部分的模拟调试可借助于模拟开关和 PLC 输出端的输出指示灯进行。需要模拟量信号 I/O 时,可用电位器和万用表配合进行。调试时,可利用上述外围设备模拟各种现场开

关和传感器状态,然后观察 PLC 的输出逻辑是否正确。如果有错误则修改后反复调试。现在 PLC 的主流产品都可在 PC 机上编程,并可在电脑上直接进行模拟调试。

联机调试时,可把编制好的程序下载到现场的 PLC 中。有时 PLC 也许只有这一台,这时就要把 PLC 安装到控制柜相应的位置上。调试时一定要先将主电路断电,只对控制电路进行联调即可。通过现场联调信号的接入常常还会发现软硬件中的问题,有时厂家还要对某些控制功能进行改进,这种情况下,都要经过反复测试系统后,才能最后交付使用。

系统完成后一定要及时整理技术材料并存档,不然日后会需要几倍的辛苦来做这件事。这也是工程技术人员良好的习惯之一。

## 9.2 双恒压无塔供水控制系统设计

本例综合了 PLC 在多方面的应用,既有开关量 I/O,也有模拟量 I/O;既有 PID 调节的典型使用,又有复杂的逻辑控制。另外,本例中还使用了变频器和电机软启动控制。

### 9.2.1 工艺过程

随着社会的发展和进步,城市高层建筑的供水问题日益突出。一方面要求提高供水质量,不要因为压力的波动造成供水障碍;另一方面要求保证供水的可靠性和安全性,在发生火灾时能够可靠供水。针对这两方面的要求,新的供水方式和控制系统应运而生,这就是 PLC 控制的恒压无塔供水系统。恒压供水包括生活用水的恒压控制和消防用水的恒压控制——即双恒压系统。恒压供水保证了供水的质量,以 PLC 为主机的控制系统丰富了系统的控制功能,提高了系统的可靠性。

下面以一个三泵生活/消防双恒压无塔供水系统为例来说明其工艺过程(已做过简化)。如图 9-2 所示,市网来水用高低水位控制器 EQ 来控制注水阀 YV1,它们自动把水注满储水水池,只要水位低于高水位,则自动往水箱中注水。水池的高/低水位信号也直接送给 PLC,作为低水位报警用。为了保证供水的连续性,水位上下限传感器高低距离不是相差很大。生活用水和消防用水共用三台泵,平时电磁阀 YV2 处于失电状态,关闭消防管网,三台泵根据生活用水的多少,按一定的控制逻辑运行,使生活供水在恒压状态(生活用水低恒压值)下进行;当有火灾发生时,电磁阀 YV2 得电,关闭生活用水管网,三台泵供消防用水使用,并根据用水量的大小,使消防供水也在恒压状态(消防用水高恒压值)下进行。火灾结束后,三台泵再改为生活供水使用。

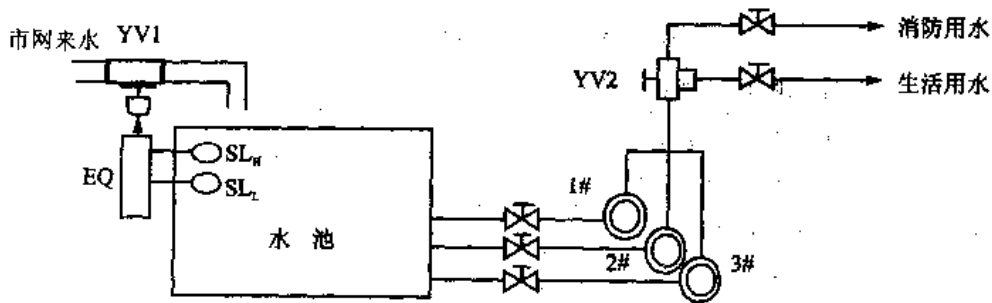


图 9-2 生活/消防双恒压供水系统工艺流程图

### 9.2.2 系统控制要求

对三泵生活/消防双恒压供水系统的基本要求是:

- (1) 生活供水时,系统应低恒压值运行,消防供水时系统应高恒压值运行;
- (2) 三台泵根据恒压的需要,采取“先开先停”的原则接入和退出;
- (3) 在用水量小的情况下,如果一台泵连续运行时间超过 3 h,则要切换到下一台泵,即系统具有“倒泵功能”,避免某一台泵工作时间过长;
- (4) 三台泵在启动时要有软启动功能;
- (5) 要有完善的报警功能;
- (6) 对泵的操作要有手动控制功能,手动只在应急或检修时临时使用。

### 9.2.3 控制系统的 I/O 点及地址分配

控制系统的输入/输出信号的名称、代码及地址编号如表 9-1 所列。水位上下限信号分别为 I0.1、I0.2,它们在水淹没时为 0,露出时为 1。

表 9-1 输入/输出点代码和地址编号

名 称	代 码	地 址 编 号
输 入 信 号		
手动和自动消防信号	SA1	I0.0
水池水位下限信号	SL <sub>L</sub>	I0.1
水池水位上限信号	SL <sub>H</sub>	I0.2
变频器报警信号	S <sub>0</sub>	I0.3
消铃按钮	SB9	I0.4
试灯按钮	SB10	I0.5
远程压力表模拟量电压值	U <sub>P</sub>	AIW0
输 出 信 号		
1#泵工频运行接触器及指示灯	KM1, HL1	Q0.0
1#泵变频运行接触器及指示灯	KM2, HL2	Q0.1
2#泵工频运行接触器及指示灯	KM3, HL3	Q0.2
2#泵变频运行接触器及指示灯	KM4, HL4	Q0.3
3#泵工频运行接触器及指示灯	KM5, HL5	Q0.4
3#泵变频运行接触器及指示灯	KM6, HL6	Q0.5
生活/消防供水转换电磁阀	YV2	Q1.0
水池水位下限报警指示灯	HL7	Q1.1
变频器故障报警指示灯	HL8	Q1.2
火灾报警指示灯	HL9	Q1.3
报警电铃	HA	Q1.4
变频器频率复位控制	KA(EMG)	Q1.5
控制变频器频率电压信号	V <sub>f</sub>	AQW0

### 9.2.4 PLC 系统选型

从上面分析可以知道,系统共有开关量输入点 6 个、开关量输出点 12 个;模拟量输入点 1 个、模拟量输出点 1 个。如果选用 CPU 224 PLC,也需要扩展单元;如果选用 CPU 226 PLC,则价格较高,浪费较大。参照西门子 S7-200 产品目录及市场实际价格,选用主机为 CPU 222(8 入/6 继电器输出)一台,加上一台扩展模块 EM 222(8 继电器输出),再扩展一个模拟量模块 EM 235(4AI/1AO)。这样的配置是最经济的。整个 PLC 系统的配置如图 9-3 所示。

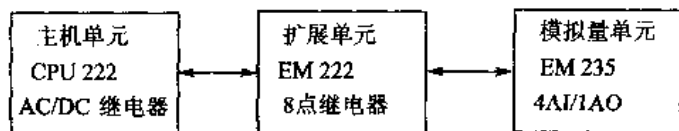


图 9-3 PLC 系统组成

### 9.2.5 电气控制系统原理图

电气控制系统原理图包括主电路图、控制电路图及 PLC 外围接线图。

#### 1. 主电路图

如图 9-4 所示为电控系统主电路。三台电机分别为 M1、M2、M3。接触器 KM1、KM3、KM5 分别控制 M1、M2、M3 的工频运行;接触器 KM2、KM4、KM6 分别控制 M1、M2、M3 的变频运行,FR1、FR2、FR3 分别为三台水泵电机过载保护用的热继电器;QS1、QS2、QS3、QS4 分别为变频器和三台水泵电机主电路的隔离开关;FU1 为主电路的熔断器,VVVF 为简单的一般变频器。

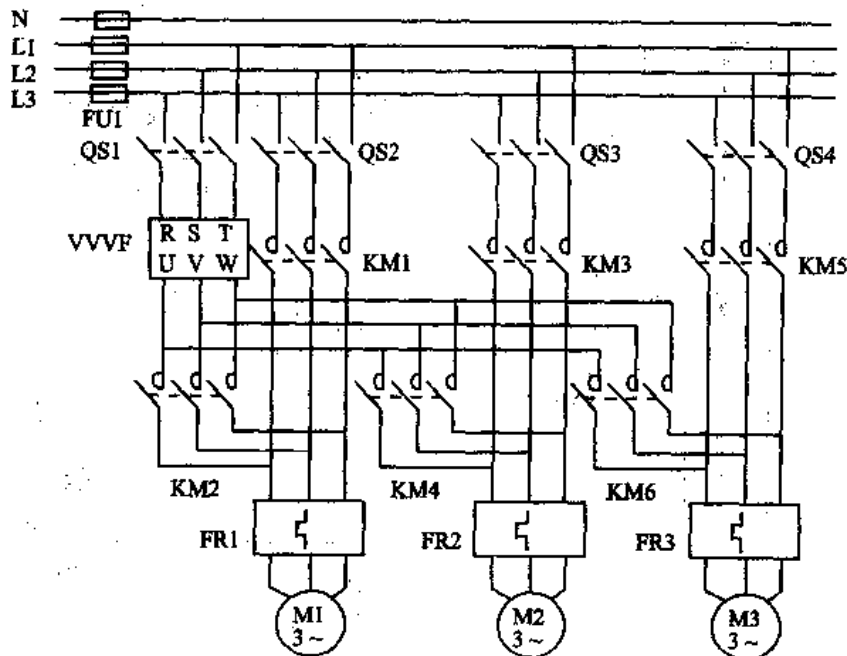


图 9-4 电控系统主电路

#### 2. 控制电路图

如图 9-5 所示为电控系统控制电路图。图中 SA 为手动/自动转换开关,SA 打在 1 的位

置为手动控制状态;打在 2 的状态为自动控制状态。手动运行时,可用按钮 SB1~SB8 控制三台泵的启/停和电磁阀 YV2 的通/断;自动运行时,系统在 PLC 程序控制下运行。

图中的 HL10 为自动运行状态电源指示灯。对变频器频率进行复位时只提供一个干触点信号,由于 PLC 为 4 个输出点为一组共用一个 COM 端,而本系统又没有剩下单独的 COM 端输出组,所以通过一个中间继电器 KA 的触点对变频器进行复频控制。图中的 Q0.0~Q0.5 及 Q1.0~Q1.5 为 PLC 的输出继电器触点,它们旁边的 4、6、8 等数字为接线编号,可结合图 9-6 一起读图。

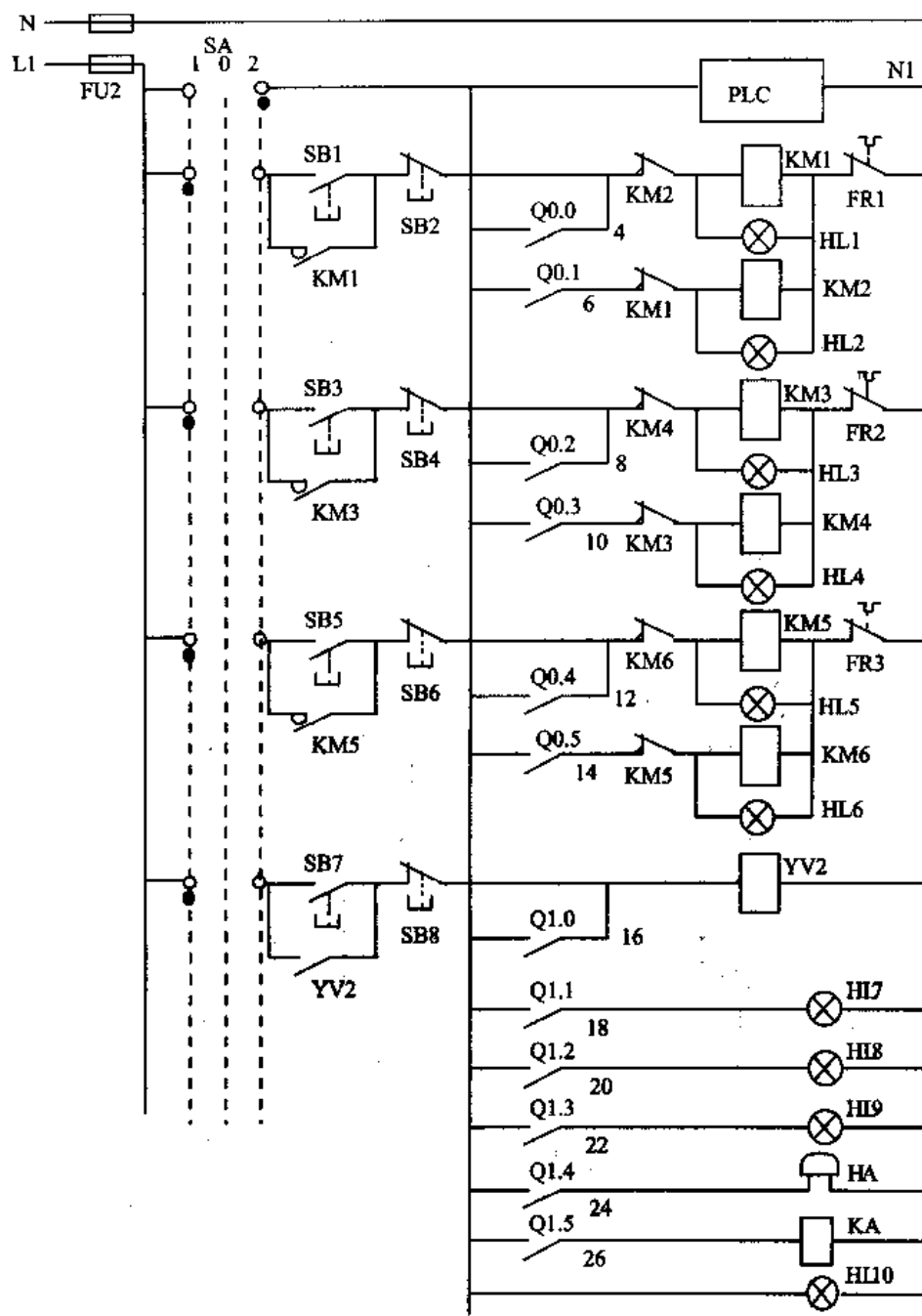


图 9-5 电控系统控制电路

## 3. PLC 外围接线图

如图 9-6 所示为 PLC 及扩展模块外围接线图。火灾时,火灾信号 SA1 被触动,IO.0 为 1。

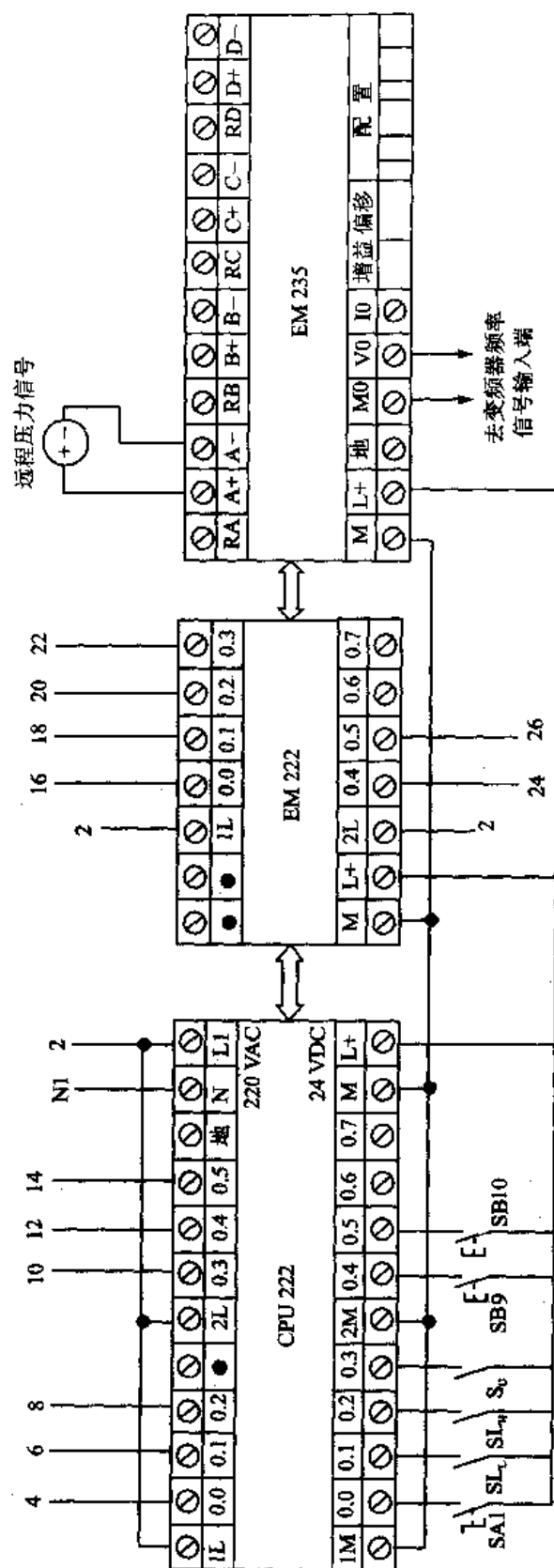


图 9-6 恒压供水控制系统 PLC 及扩展模块外围接线

本例只是一个教学例子,实际使用时还必须考虑许多其他因素,这些因素主要包括:

- (1) 直流电源的容量;
- (2) 电源方面的抗干扰措施;
- (3) 输出方面的保护措施;
- (4) 系统保护措施。

### 9.2.6 系统程序设计

本程序分为三部分:主程序、子程序和中断程序。

逻辑运算及报警处理等放在主程序。系统初始化的一些工作放在初始化子程序中完成,这样可节省扫描时间。利用定时器中断功能实现 PID 控制的定时采样及输出控制。生活供水时系统设定值为满量程的 70%,消防供水时系统设定值为满量程的 90%。在本系统中,只是用比例(P)和积分(I)控制,其回路增益和时间常数可通过工程计算初步确定,但还需要进一步调整以达到最优控制效果。初步确定的增益和时间常数为(参考本书 PID 指令的使用一节):

增益  $K_c = 0.25$ ;

采样时间  $T_s = 0.2 \text{ s}$ ;

积分时间  $T_i = 30 \text{ min}$ 。

程序中使用的 PLC 元器件及其功能如表 9-2 所列。

表 9-2 程序中使用的元器件及功能

器 件 地 址	功 能	器 件 地 址	功 能
VD100	过程变量标准化值	T38	工频泵减泵滤波时间控制
VD104	压力给定值	T39	工频/变频转换逻辑控制
VD108	PI 计算值	M0.0	故障结束脉冲信号
VD112	比例系数	M0.1	泵变频启动脉冲
VD116	采样时间	M0.3	倒泵变频启动脉冲
VD120	积分时间	M0.4	复位当前变频运行泵脉冲
VD124	微分时间	M0.5	当前泵工频运行启动脉冲
VD204	变频器运行频率下限值	M0.6	新泵变频启动脉冲
VD208	生活供水变频器运行频率上限值	M2.0	泵工频/变频转换逻辑控制
VD212	消防供水变频器运行频率上限值	M2.1	泵工频/变频转换逻辑控制
VD250	PI 调节结果存储单元	M2.2	泵工频/变频转换逻辑控制
VB300	变频工作泵的的泵号	M3.0	故障信号汇总
VB301	工频运行的泵的总台数	M3.1	水池水位下限故障逻辑
VD310	倒泵时间存储器	M3.2	水池水位下限故障消铃逻辑
T33	工频/变频转换逻辑控制	M3.3	变频器故障消铃逻辑
T34	工频/变频转换逻辑控制	M3.4	火灾消铃逻辑
T37	工频泵增泵滤波时间控制	—	—

双恒压供水系统的梯形图程序及程序注释如图 9-7 所示。对该程序有几点说明:

- (1) 因为程序较长,所以读图时请按网络标号的顺序进行;
- (2) 本程序的控制逻辑设计针对的是较少泵数的供水系统;

- (3) 本程序不是最优设计；  
 (4) 本程序已做过大量简化, 不能作为实际使用的程序。

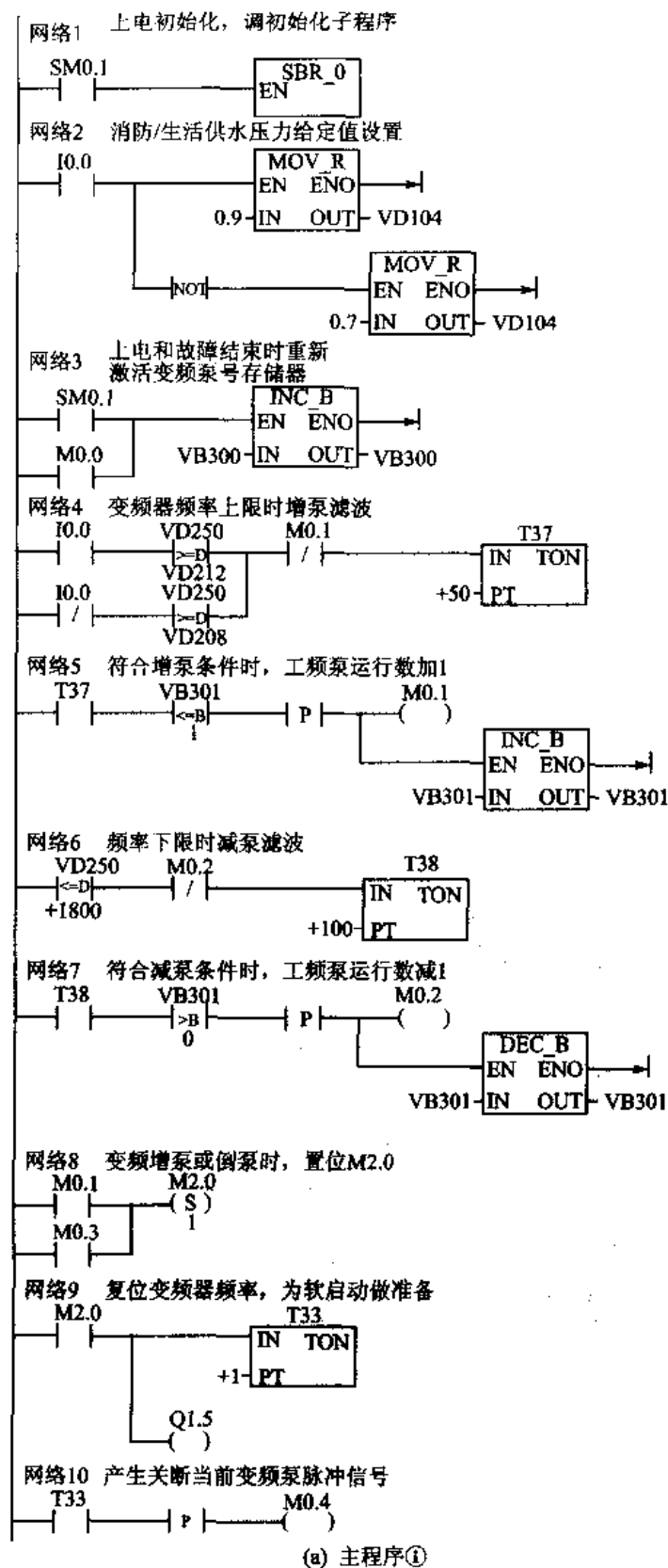


图 9-7 双恒压供水系统梯形图程序

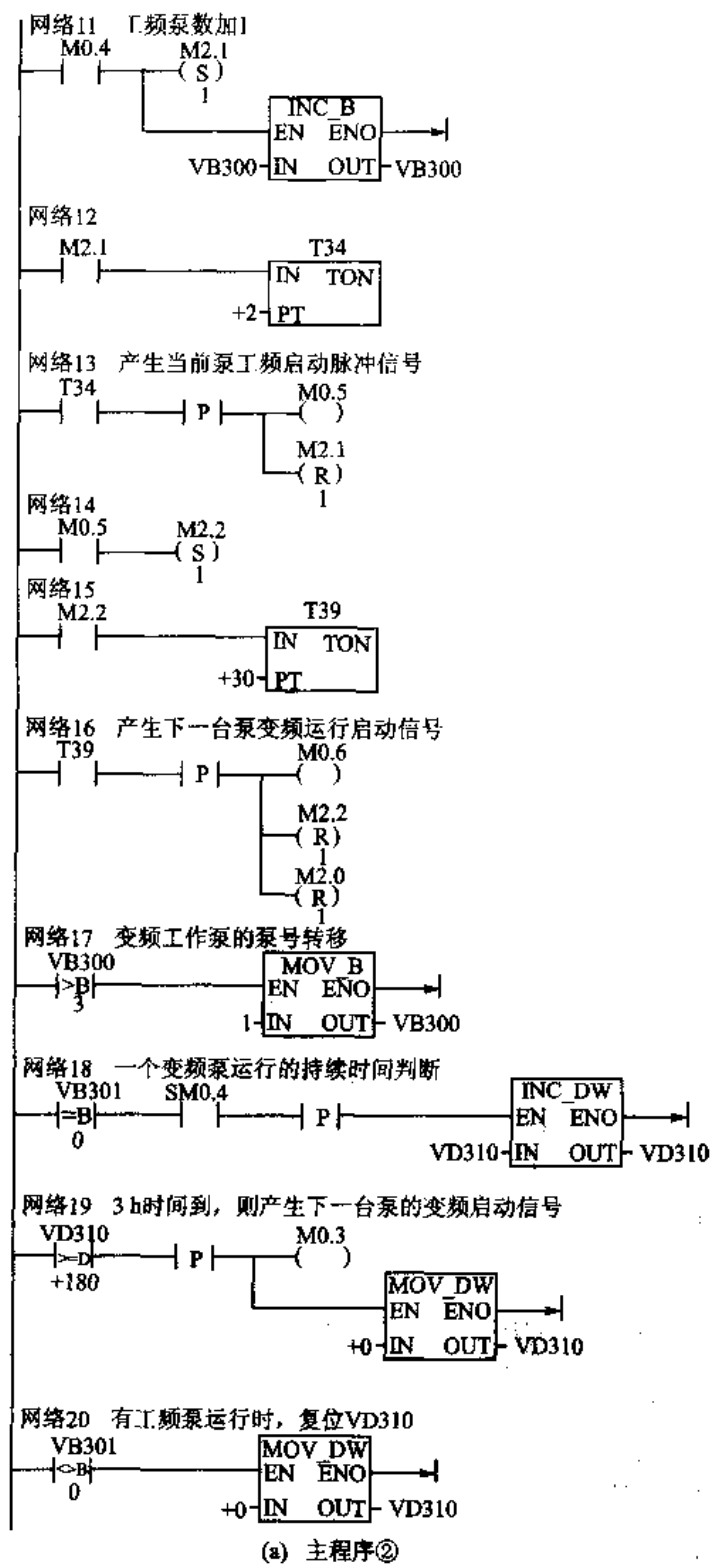


图 9-7 双恒压供水系统梯形图程序(续)

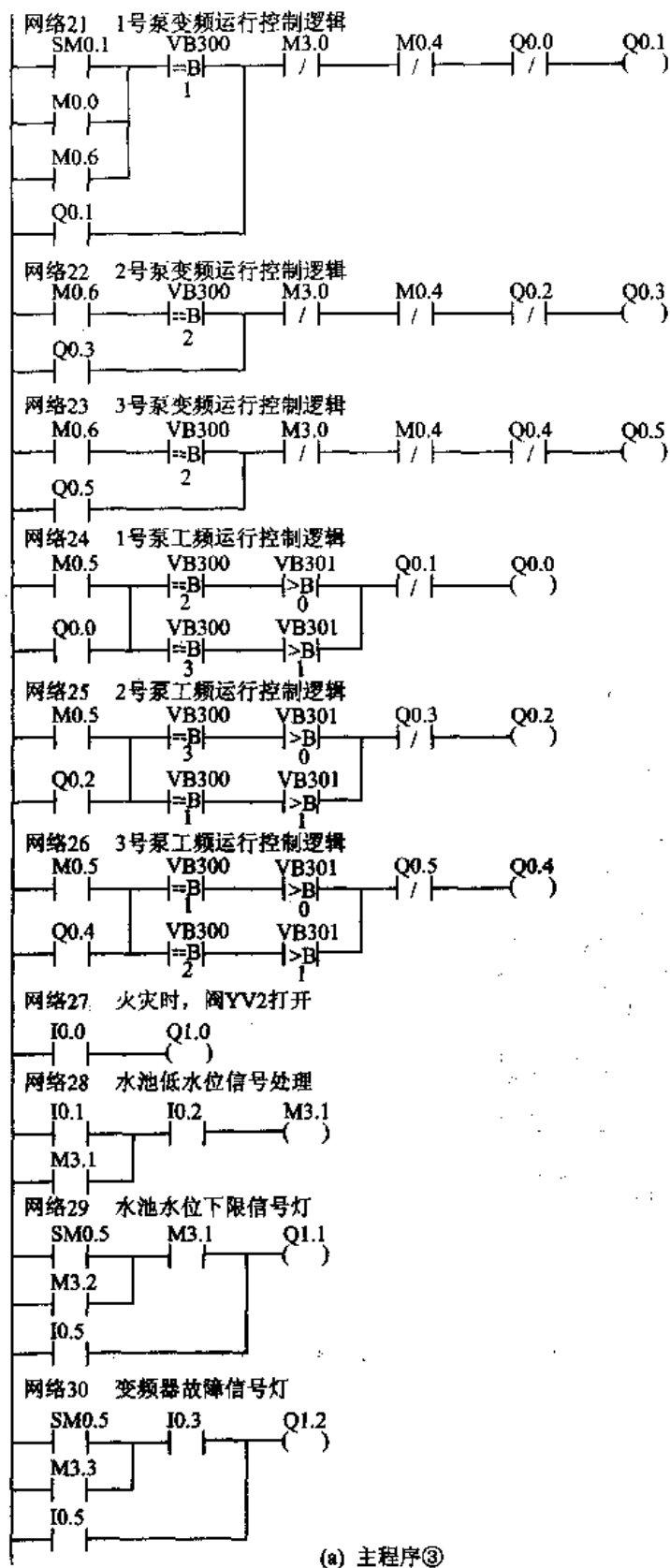


图 9-7 双恒压供水系统梯形图程序(续)

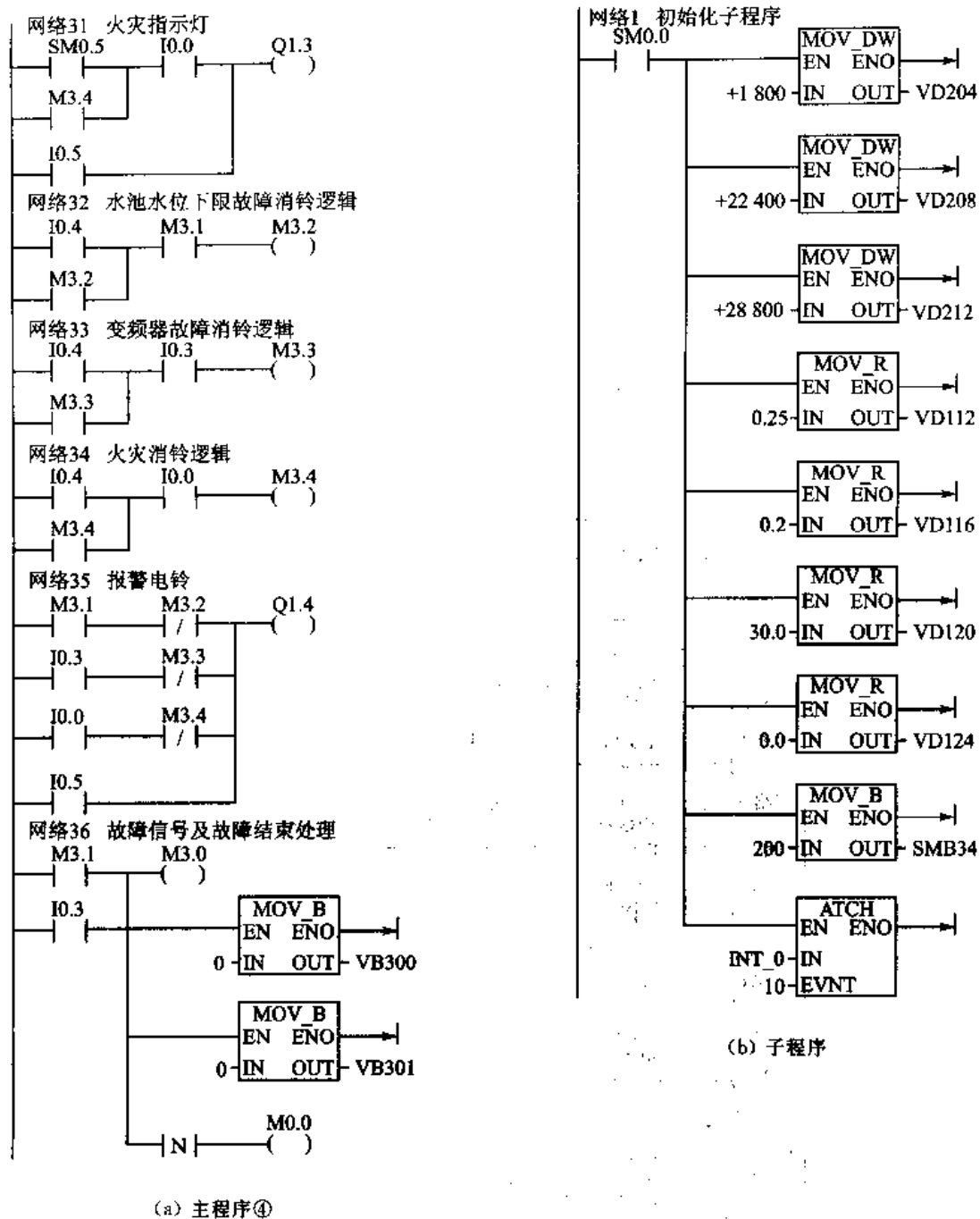
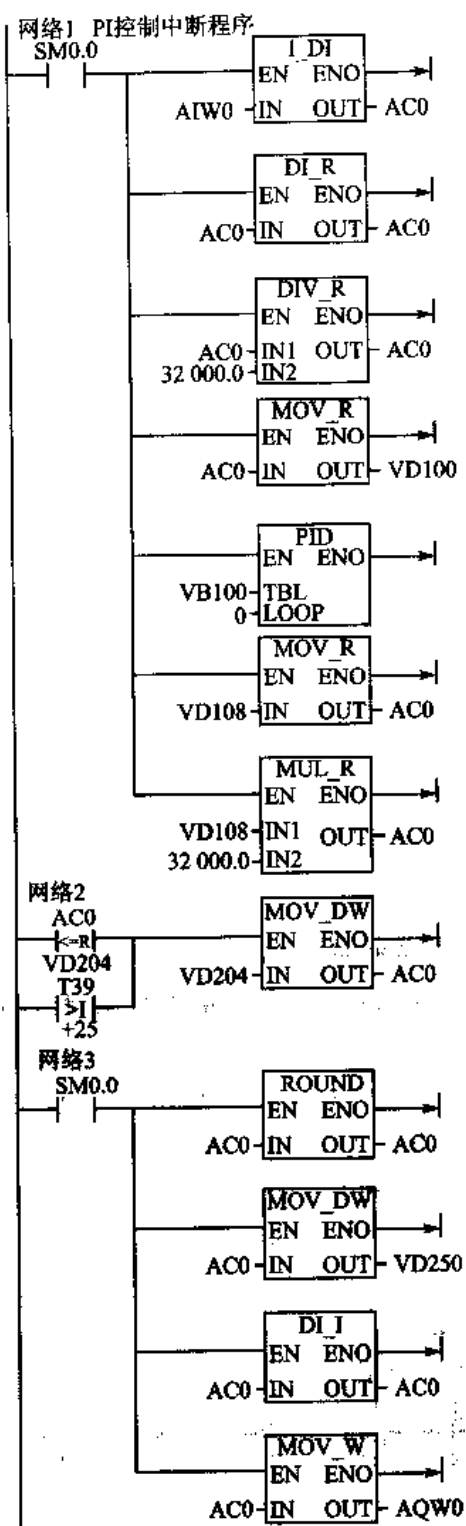


图 9-7 双恒压供水系统梯形图程序(续)



(c) 中断子程序

图 9-7 双恒压供水系统梯形图程序(续)

### 9.3 薄刀式分切压痕机控制系统

薄刀式分切压痕机是生产包装纸箱所用瓦楞纸生产线中重要的设备,分切出的纸板克服了传统的厚刀分切的诸多缺点,成品纸板边缘平整、光洁,无压扁现象,从而提高了纸板的整体质量,特别是对包装纸箱生产线上后序印刷质量带来了根本性的提高。

#### 9.3.1 工艺过程

瓦楞纸生产线生产流程如图 9-8 所示。整个工艺过程分为开卷、压痕、涂胶、粘合、烘干、分切和磨刀七部分,实际的控制系统非常复杂,下面主要讲述压痕、分切和磨刀的控制。

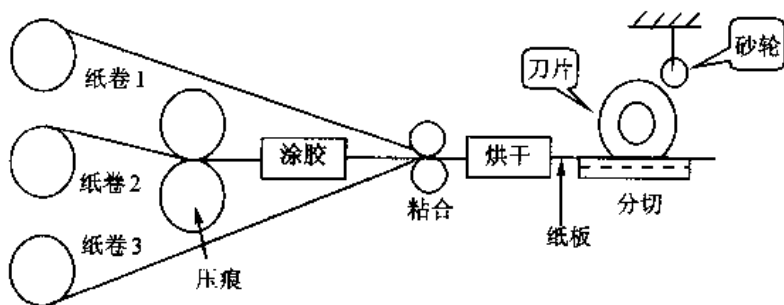


图 9-8 生产工艺流程示意图

#### 9.3.2 系统控制要求

(1) 压痕辊的线速度要求与送来纸板速度保持一致,即与主轴开卷机轴的线速度相等,由于开卷机的速度是可调的,所以压痕辊由变频器驱动控制,根据开卷机的速度调整范围,要求压痕轴的速度范围为  $0 \sim 1\,500 \text{ r/s}$ ,主轴的速度用测速机检测,输出的电压范围为  $0 \sim 10 \text{ V}$ 。

(2) 分切装置上有四片分切刀片,如图 9-9 所示,刀片位置可根据要求进行调整,刀片的速度应根据纸板的速度变化而变化。所以分切装置轴的驱动也采用变频器驱动,但又有最低转速和最高转速的限制,根据主轴速度和上下限的要求确定的刀片速度范围为  $300 \sim 1\,200 \text{ r/s}$ 。

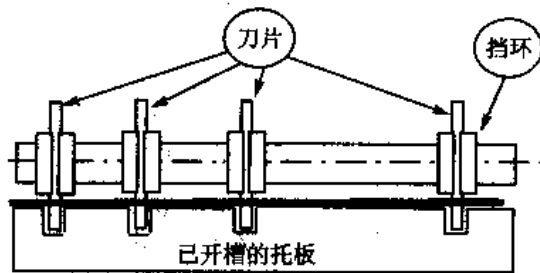


图 9-9 分切装置示意图

(3) 分切装置上的分切刀片在分切一段时间变钝后,磨刀装置应对刀片进行磨削锐化,以保持分切刀片刀刃的锐利,分切刀片由锐利变钝的时间与分切装置的转速以及切削纸板的多少有关,因此每次磨刀的时间间隔应与分切装置的速度保持一致,磨刀时间与粗设的磨刀间隔时间由 PLC 上的模拟电位器设定,设定刀片每次磨削时间的范围为  $500 \sim 1\,000 \text{ ms}$ ,磨刀时间

间隔为 5 000~10 000 ms。

磨刀装置上的四片磨刀砂轮由不同的电磁阀控制,分别对四片刀片进行磨削,磨刀要求有手动和自动两种控制方式。手动时,按一次磨刀按钮进行一次磨刀操作;自动时,分切只要运行,PLC 就根据粗设的磨刀间隔时间和刀片的速度微调的间隔时间,自动地进行磨刀操作。

每个刀片的磨刀装置可以人为地设定为工作和停止两种状态。

(4) 生产过程中会发生纸板跑偏现象,横向移动装置应随时自动地进行跟踪纠偏,也可以手动横向移动进行纠偏,用两个光电开关作为纠偏的位置检测,用两个行程开关进行限制横向移动的行程范围。

### 9.3.3 控制系统的 I/O 点及地址分配

控制系统的输入、输出信号及代码、地址编号如表 9-3 所列。

表 9-3 输入输出信号代码和地址编号表

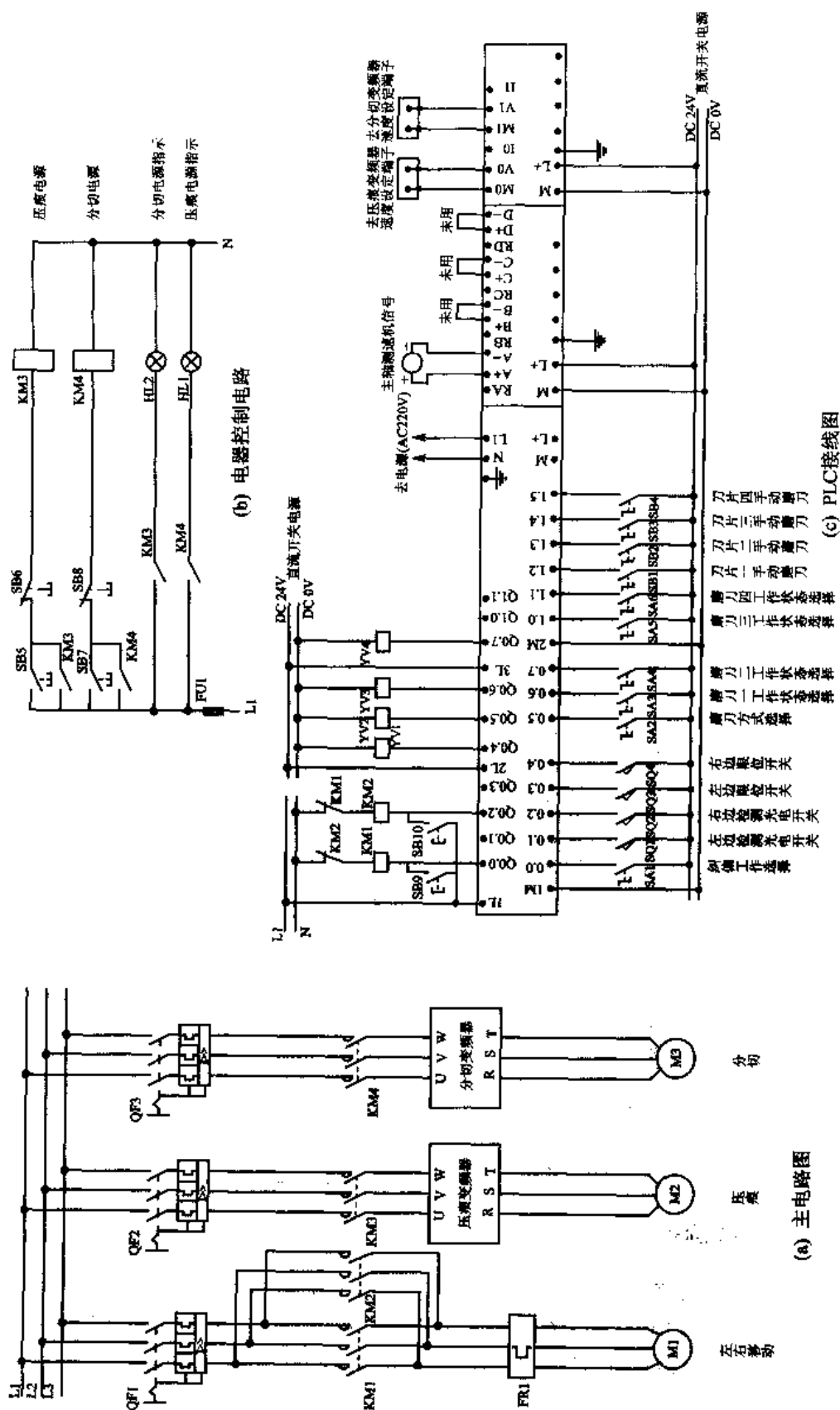
名 称	代 码	地 址 编 号
纠偏工作方式选择	SA1	I0.0
左边检测光电开关	SQ1	I0.1
右边检测光电开关	SQ2	I0.2
左边限位开关	SQ3	I0.3
右边限位开关	SQ4	I0.4
磨刀方式选择	SA2	I0.5
磨刀一工作状态选择	SA3	I0.6
磨刀二工作状态选择	SA4	I0.7
磨刀三工作状态选择	SA5	I1.0
磨刀四工作状态选择	SA6	I1.1
刀片一手动磨刀	SB1	I1.2
刀片二手动磨刀	SB2	I1.3
刀片三手动磨刀	SB3	I1.4
刀片四手动磨刀	SB4	I1.5
主轴速度检测电压输入	—	AIW0
分切装置左移	KM1	Q0.0
分切装置右移	KM2	Q0.2
磨刀一	YV1	Q0.4
磨刀二	YV2	Q0.5
磨刀三	YV3	Q0.6
磨刀四	YV4	Q0.7
压痕变频器速度设定电压	—	AQW0
分切变频器速度设定电压	—	AQW2

### 9.3.4 PLC 系统选型

通过对系统控制要求的分析可知,系统共有开关量输入点 14 点,开关量输出点 8 点,所以选用 CPU 224(14DI/10DO);由于系统需要 1 路模拟量输入和 2 路模拟量输出,所以选用一块 EM231 模拟量输入(四路)扩展模块和一块 EM232(两路)模拟量输出模块。

### 9.3.5 电气控制系统原理图

系统控制原理如图 9-10 所示。分切装置的横向移动由电机 M1 完成,KM1 和 KM2 分



别控制电机的正反转,完成分切装置的左右移动。KM1 和 KM2 既可以由 PLC 自动控制又可以通过手动按钮 SB9 和 SB10 控制, KM3 和 KM4 分别控制压痕变频器与分切变频器的供电和运行,运行的速度由 PLC 的模拟量输出控制。

### 9.3.6 系统程序设计

本系统的控制程序按功能分为四部分,分别为两个变频器的速度控制、自动纠偏、磨刀控制以及磨刀时间和间隔时间的设定。系统的梯形图程序如图 9-11 所示。

压痕变频器的速度根据主轴的速度设定,在本程序中采用定时中断的方式对主轴速度进行采样,采样周期为 100 ms,在中断程序中将采样的主轴的速度存放在 VW10 中,数据范围为 0~32 000,对应的主轴速度为 0~1 500 r/s。详见主程序的网络 1 和中断程序。

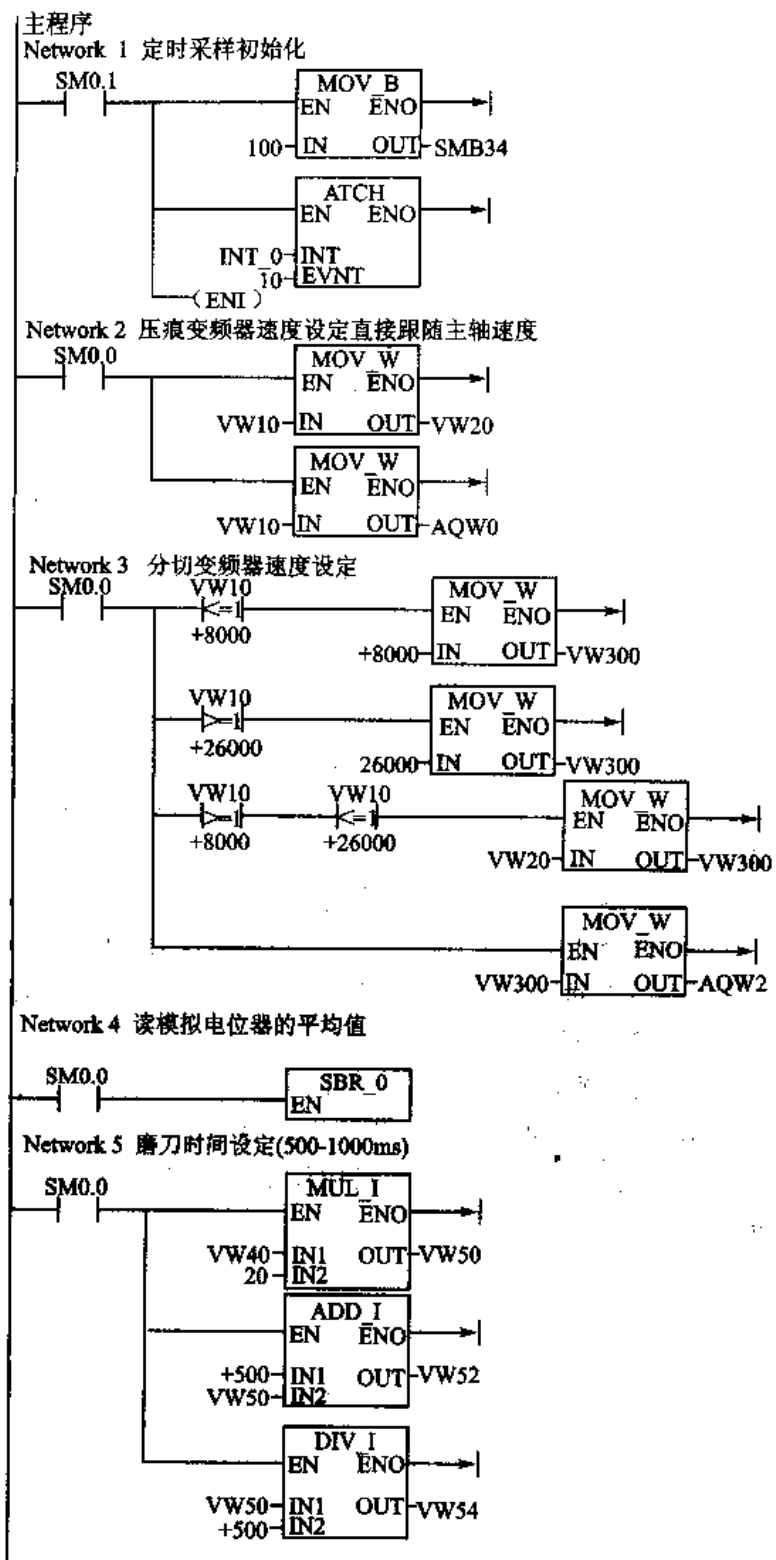
压痕电机要求的速度为 0~1 500 r/s,所以将采到的主轴的速度通过 AQW0 对压痕变频器输出频率进行设定。

分切电机要求的速度为 300~1 200 r/s,所以只要将分切变频器速度设定单元 VW300 的上限和下限分别设定为 26 000(对应速度 1 200 r/s)和 8 000(对应速度 300 r/s),然后通过 AQW2 对分切变频器输出频率进行设定即可。

S7-200 PLC CPU 提供了一个或两个模拟电位器,可以通过调节这些电位器来增加或降低存于特殊存储器(SMB28 和 SMB29)中的值(0~255),这些只读值在程序中可以作为定时器、计数器的设定值等多种功能。在本例中,就是利用这两个模拟电位器来作为磨刀时间和磨刀间隔时间的设定值。

在本例中,对两个特殊存储器中的值进行多次读取并求得平均值,再将此平均值按照要求通过一定的转化,最后得到符合要求的磨刀时间(VW54)和粗设磨刀间隔时间(VW64)的设定值,然后根据主轴的速度将磨刀间隔时间进行微量调整,即主轴速度高则要求磨刀间隔时间短,最后得到磨刀间隔时间(VW66)。

自动磨刀时,停止间隔时间到,则按顺序分别对四片刀进行磨削;手动磨刀时,某一个时刻只能磨一片刀;自动纠偏时,如果边沿检测光电开关检测到纸板走偏,则通过横向移动电机将这个机构做相应的移动,移动的时间为 2 s,移动后应停止一段时间为 2 s,再根据检测结果进行处理,以防止机构抖动。



(a) 主程序①

图 9-11 系统程序图

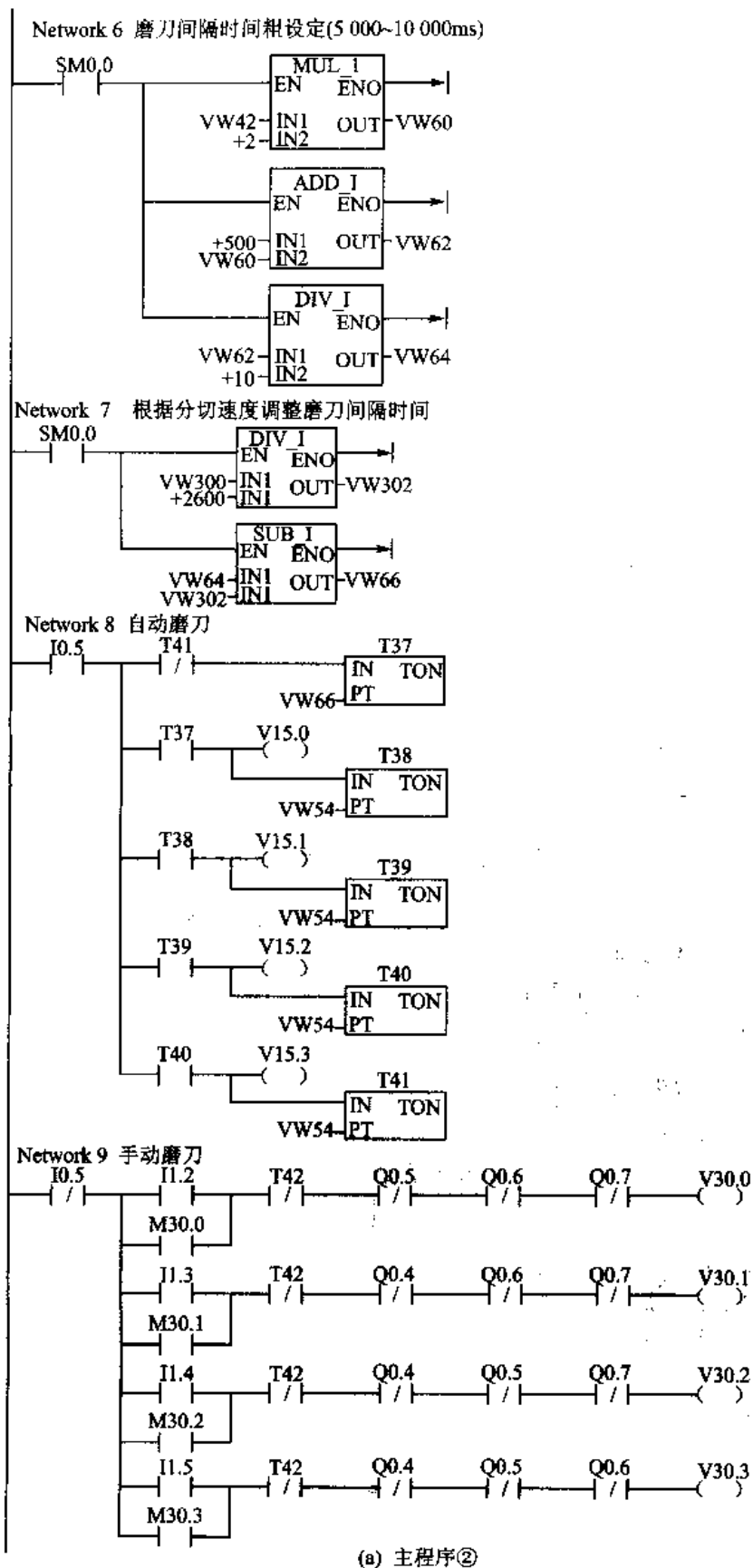


图 9-11 系统程序图(续)

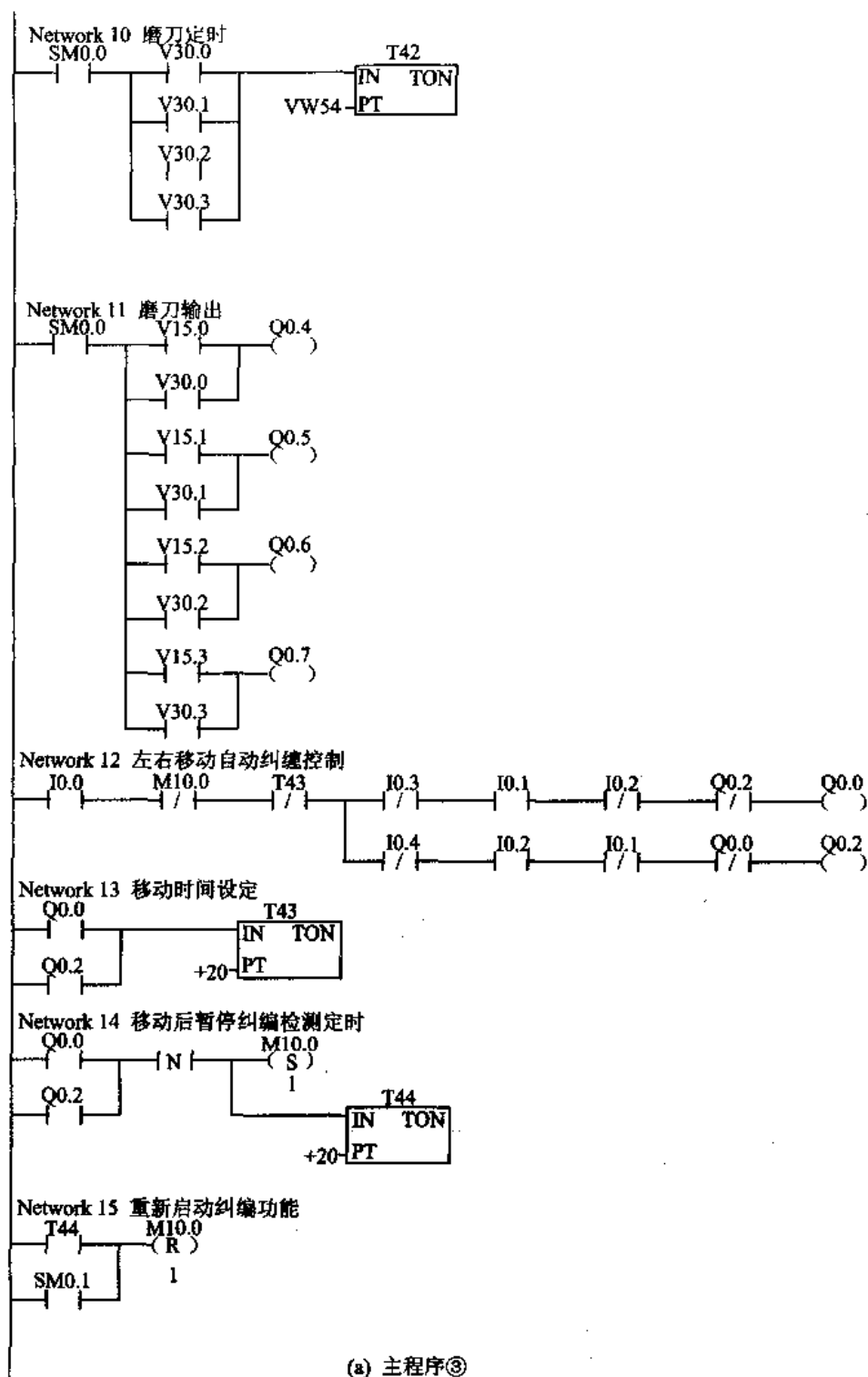
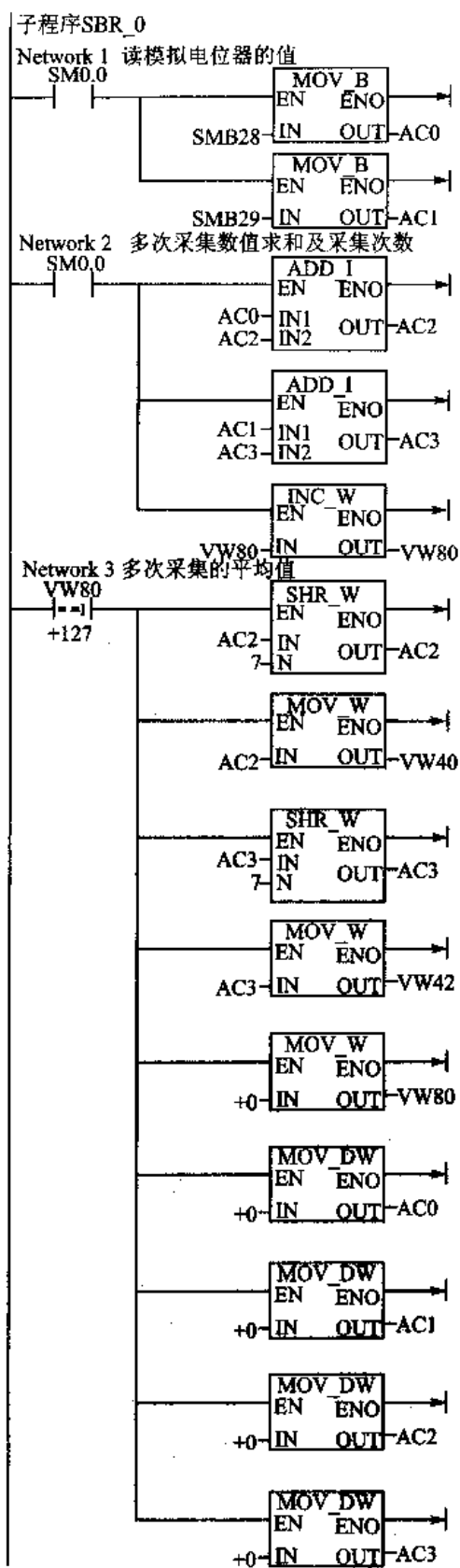
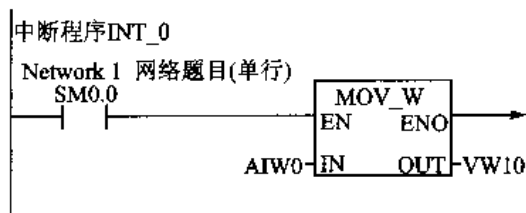


图 9-11 系统程序图(续)



(b)子程序

图 9-11 系统程序图(续)



(c) 中断程序

图 9-11 系统程序图(续)

## 9.4 PLC 在工程应用中要注意的一些实际问题

### 9.4.1 PLC 的安装

可以利用 S7-200 模块上的安装孔把模块固定在控制柜背板上,也可以利用模块上的 DIN 夹子把模块固定在一个标准的 DIN 道轨上。这样既可以水平安装,也可以垂直安装。但安装到控制柜中时,应注意以下两个问题。

(1) 为了防止高电子噪声对模块的干扰,应尽可能将 S7-200 模块与产生高电子噪声的设备(如变频器)分隔开。

(2) S7-200 模块是采用自然对流方式散热的,所以在安装时应尽可能不与产生高热量的设备安装在一起。而且在安装 S7-200 模块时,模块的周围应留出一定的空间,以便于正常散热。一般情况下,模块的上方和下方至少留出 25 mm 的空间,模块前面板与底板之间至少留出 75 mm 的空间。

### 9.4.2 电源的设计

#### 1. 供电电源

可编程控制器一般使用市电(220 V, 50 Hz)。电网的冲击、频率的波动将直接影响到可编程控制器系统实时控制的精度和可靠性;有时电网的冲击,可给系统带来毁灭性的破坏;电网的瞬间变化也是经常不断发生的,由此产生的干扰也会传播到可编程控制器系统中。为了提高系统的可靠性和抗干扰性能,在对可编程控制器的供电系统中一般采用隔离变压器,这样可以隔离掉供电电源中的各种干扰信号,从而提高了系统的抗干扰性能。

如果使用开关电源为可编程控制器提供 24 V 直流电源,一般情况下,对开关电源供电的交流电源也应采用隔离变压器与电源隔离。

另外,在一些实时控制系统中,系统的突然断电会造成严重的后果,此时可以在供电系统中增加 UPS 不间断电源。当市电突然断电后,自动切换到 UPS 电源供电,并且按照工艺要求进行一定的处理,使生产设备处于安全状态。

#### 2. S7-200 CPU 内部直流电源

每个 S7-200 CPU 模块均提供一个 24 V 直流传感器电源和一个 5 V 直流电源。

24 V 直流传感器电源可以作为 CPU 本机和数字量扩展模块的输入、扩展模块(如模拟量模块)的供电电源以及外部传感器电源使用。如果容量不能满足所有需求,则必须增加外部

24 V 直流电源,此时外部电源不能与模块的传感器电源并联使用,以防止两个电源电位的不平衡造成对电源的破坏,但为了加强电子噪音保护,这两个电源的公共端(M)应连接在一起。

当 S7-200 CPU 与扩展模块连接时,CPU 模块为扩展模块提供 5 V 直流电源。如果扩展模块的 5 V 直流电源需求超出 CPU 模块 5 V 直流电源的容量,则必须减少扩展模块的数量。

有关 S7-200 内部直流电源容量的设计与计算请参阅 S7-200 系统手册。

### 9.4.3 系统的接地

在可编程控制器系统中,接地是抑制干扰使系统可靠工作的主要方法。在设计与施工中,如果把接地与屏蔽正确结合起来,可以解决大部分的干扰问题。

接地有两个目的,一是消除各电流流经公共地线阻抗时所产生的噪声电压;二是避免磁场与电位差的影响。正确的接地是一个重要而复杂的问题,理想的情况是一个系统的所有接地点与大地之间的阻抗为零,但这是很难做到的。

在一般的接地过程中要求如下:(1)接地电阻应小于  $4\Omega$ ;(2)具有足够的机械强度;(3)具有耐腐蚀及防腐处理;(4)可编程控制器系统单独接地。

在可编程控制器系统中常见的地线有:

- 数字地 也叫逻辑地,是各种开关量(数字量)信号的零电位;
- 模拟地 是各种模拟量信号的零电位;
- 信号地 通常为传感器的地;
- 交流地 交流供电电源的地线,这种地线是产生噪声的地;
- 直流地 直流供电电源的地;
- 屏蔽地 机壳地,为防止静电感应和磁场感应而设置的地。

不同的地线,处理的方法也是不同的。常用的方法有以下几种。

#### 1. 一点接地和多点接地

一般情况下,高频电路应就近多点接地,低频电路应一点接地。在低频电路中,布线和元件间的电感并不是什么大问题,然而接地形成的环路的干扰影响很大,因此常以一点作为接地点。但一点接地不适合高频,因为高频时地线上具有电感,因而增大了地线阻抗,同时各地线之间又产生电感耦合。一般来说,频率在 1 MHz 以下,可用一点接地;高于 10 MHz 时,采用多点接地;在 1~10 MHz 之间可用一点接地,也可采用多点接地。根据这些原则,可编程控制器组成的系统一般采用一点接地。

#### 2. 交流地与信号地不能共用

由于在一段电源地线的两点之间会有数毫伏,甚至几伏的电压。因此对低电平信号来说,这是一个非常严重的干扰,必须加以隔离和防止,使设备可靠运行。

#### 3. 浮地与接地

全机浮空即系统各个部分与大地浮置起来,这种方法简单,但整个系统与大地绝缘电阻不能小于  $50\text{ M}\Omega$ 。这种方法具有一定的抗干扰能力,但一旦绝缘下降就会带来干扰。

还有一种方法,就是将机壳接地,其余部分浮空。这种方法抗干扰能力强,安全可靠,但实现起来比较复杂。

由此可见,可编程控制器系统还是以接地为好。

#### 4. 模拟地

模拟地的接地方法十分重要,为了提高抗共模干扰能力,对于模拟信号可采用屏蔽浮地技术。

#### 5. 屏蔽地

在控制系统中为了减少信号中电容耦合噪声,准确检测和控制,对信号采用屏蔽措施是十分必要的。根据屏蔽目的不同,屏蔽地的接法也不一样。

### 9.4.4 电缆设计与铺设

一般来说,工业现场的环境都比较恶劣。例如现场的各种动力线会通过电磁耦合产生干扰;电焊机、火焰切割机和电动机会产生高频火花电流造成干扰;高速电子开关的接通和关断将产生高次谐波,从而形成高频干扰;大功率机械设备的启停、负载的变化将引起电网电压的波动,产生低频干扰,这些干扰都会通过与现场设备相连的电缆引入可编程控制器组成的系统中,影响系统的安全可靠工作。所以合理地设计、选择和铺设电缆在可编程控制器系统中十分重要。

对可编程控制器组成的系统而言,电缆包括供电系统的动力电缆及各种开关量、模拟量、高速脉冲、远程通信等信号电缆。一般情况下,对系统供电系统的动力电缆、距离比较近的开关量信号使用的电缆无特殊要求;对模拟量信号、高速脉冲信号以及开关量比较远时,为防止干扰信号,保证系统的控制精度,通常选用双层屏蔽电缆;对通信用的电缆一般采用厂家提供的专用电缆,也可采用带屏蔽的双绞线电缆。

传输线之间的相互干扰是数字控制系统中较难解决的问题。这些干扰主要来自传输导线间分布电容、电感引起的电磁耦合。防止这种干扰的有效方法,使信号线远离动力线或电网;将动力线、控制线和信号线严格分开,分别布线。无论是在可编程控制器控制柜中的接线,还是在控制柜与现场设备之间的接线,都必须注意防止动力线、控制线和信号线之间的干扰。

### 9.4.5 PLC 输出端的保护

当可编程控制器的输出负载为电感性负载时,为了防止负载关断产生的高电压对可编程控制器输出点的损害,应对输出点加以保护电路,保护电路的主要作用是抑制高电压的产生。当负载为交流感性负载时,可在负载两端并联压敏电阻,或者并联阻容吸收电路,如图 9-12(a)所示,阻容吸收电路可选  $0.5\text{ W}$ 、 $100\sim 120\ \Omega$  的电阻和  $0.1\ \mu\text{F}$  的电容;当负载为直流感性负载时,可在负载两端并联续流二极管或齐纳二极管加以抑制,如图 9-12(b)所示,续流二极管可选额定电流为  $1\text{ A}$  左右的二极管。

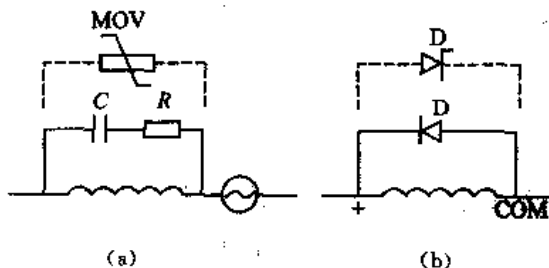


图 9-12 PLC 输出端的保护

## 本章小结

本章主要讲解 PLC 的实际工程应用。

第一节主要讲解在设计一个 PLC 控制系统时应遵循的步骤。从控制系统选择、PLC 选型、软硬件设计到系统调试等诸多方面都进行了详细的说明。大家应掌握其中的精髓,领会 PLC 控制系统设计中实质性的东西。

第二节和第三节是本书作者完成的两个实际项目,通过对它们的详细介绍,大家可以清楚地了解和学习到一个 PLC 实际课题是如何完成的。在这两个大型例子中,我们从题目分析、PLC 选型到系统软硬件设计都进行了详细的讲解,并给出了全部的硬件电路图和梯形图程序。大家可以对其设计过程和程序的细节进行分析,从而掌握 PLC 系统设计的深层次的内容。

最后一节我们介绍了在 PLC 控制系统设计中要注意的实际问题,并给出了解决方案,供大家实际应用时参考。

## 思考题与练习题

1. 一般来说,中小型 PLC 最适合应用在什么类型的控制系统中?
2. 选择 PLC 时,一般要考虑哪两方面的问题?
3. 在恒压供水的例子中,三台泵采取“先开先停”的原则接人和退出,但本例给出的程序设计不适合多泵的情况,请使用移位寄存器的方法设计一个适合多泵(即多控制对象)控制系统的程序,使控制对象满足“先开先停”的原则接人和退出。
4. 任何一个控制系统的接地设计都非常重要,请结合本章的讲解和图 B-1 掌握 PLC 控制系统的接地方法,领会接地在抗干扰和系统保护方面的重要性。
5. 如果是感性负载时,如何完成对 PLC 输出端的保护?

## 第十章 编程软件的使用

S7-200 可编程序控制器使用 STEP7-Micro/WIN 32 编程软件进行编程。

STEP7-Micro/WIN 32 编程软件是基于 Windows 的应用软件,由西门子公司专门为 S7-200 系列可编程控制器设计开发。它功能强大,主要为用户开发控制程序使用,同时也可实时监控用户程序的执行状态。它是西门子 S7-200 用户不可缺少的开发工具。现在加上汉化程序后,可在全汉化的界面下进行操作,使中国的用户使用起来更加方便与实用。

本章主要内容:

- 编程软件的简介及安装;
- 编程软件的功能;
- 如何用编程软件编程;
- 用编程软件进行调试和运行监控。

### 10.1 编程软件安装

#### 10.1.1 系统要求

操作系统:Windows 95、Windows 98、Windows ME 或 Windows 2000。

计算机及配置:IBM 486 以上兼容机,内存 8 MB 以上,VGA 显示器,至少 50 MB 以上硬盘空间,Windows 支持的鼠标。

通信电缆:使用 PC/PPI 电缆将计算机与 PLC 连接。

#### 10.1.2 软件安装

##### 1. 一般英文版本的编程软件安装

STEP7-Micro/WIN 32 编程软件在一张光盘上,用户可按以下步骤安装:

- 将光盘插入光盘驱动器,系统自动进入安装向导。
  - 或在光盘目录里双击 setup,则进入安装向导。
  - 按照安装向导完成软件的安装。软件程序安装路径可以使用默认子目录,也可以用“浏览”按钮弹出的对话框中任意选择或新建一个子目录。
  - 在安装结束时,会出现下面的选项:  
是,我现在要重新启动计算机(默认选项);  
否,我以后再启动计算机。
- 如果出现该选项,建议用户选择默认项,单击“完成”按钮,完成安装。
- 首次安装完成后,会出现一个选项:  
是,我现在浏览 Readme 文件(默认选项);

否,我现在要进入 STEP7 - Micro/WIN 32。

如果出现该选项,建议用户选择默认选项,随后阅读 Readme 文件,并浏览关于 STEP7 - Micro/WIN 32 编程软件的最新信息。该信息也可在光盘中的 Readme X. txt 文件中得到。此处的 X 代表不同国家的语言,通常用 A、B、C、D 和 E 表示,它们分别代表德文、英文、法文、西班牙文和意大利文。

## 2. 汉化版编程软件的安装

必须选择 V3.1.1.6 版本的程序安装,因为汉化补丁程序是附加在该版本的编程软件上的。程序安装如下:

(1) 在光盘目录下,找到“mwin\_service\_pack\_from V3.1 to 3.11”软件包,按照安装向导进行操作,把原来的 V3.1 英文版本编程软件转换为 3.11 版本。

(2) 打开“Chinese 3.1.1”目录;双击 setup,按安装向导操作,完成汉化补丁的安装。

(3) 完成安装。

### 10.1.3 硬件连接

可以采用 PC/PPI 电缆建立个人计算机与 PLC 之间的通信。这是单主机与个人计算机的连接,不需要其他硬件,如调制解调器和编程设备。

典型的单主机连接及 CPU 组态如图 10-1 所示。把 PC/PPI 电缆的 PC 端连接到计算机的 RS-232 通信口(一般是 COM1),把 PC/PPI 电缆的 PPI 端连接到 PLC 的 RS-485 通信口即可。

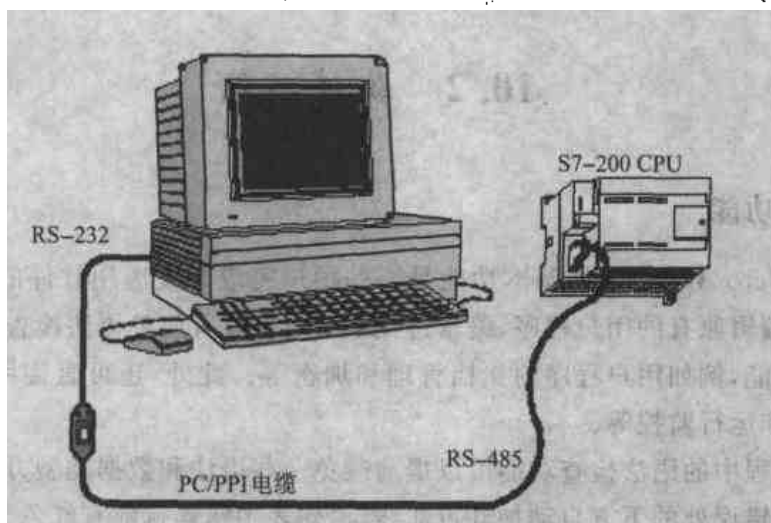


图 10-1 主机与计算机连接

### 10.1.4 参数设置

安装完软件并且设置连接好硬件之后,可以按下面的步骤核实默认的参数:

(1) 在 STEP7 - Micro/WIN 32 运行时单击通信图标,或从“视图(View)”菜单中选择选项“通信(Communications)”,则会出现一个通信对话框。

(2) 在对话框中双击 PC/PPI 电缆的图标,将出现 PG/PC 接口的对话框。

(3) 单击 Properties 按钮,将出现接口属性对话框,检查各参数的属性是否正确,其中通信波特率默认值为 9 600 bps。

### 10.1.5 在线联系

前几步如果都顺利完成,则可以建立与西门子 S7-200 CPU 的在线联系,步骤如下:

(1) 在 STEP7-Micro/WIN 32 下,单击通信图标,或从“视图(View)”菜单中选择“通信(Communications)”选项,则会出现一个通信建立结果对话框,显示是否连接了 CPU 主机。

(2) 双击通信建立对话框中的刷新图标,STEP7-Micro/WIN 32 将检查所连接的所有 S7-200 CPU 站,并为每个站建立一个 CPU 图标。

(3) 双击要进行通信的站,在通信建立对话框中,可以显示所选的通信参数。


此时,可以建立与 S7-200 CPU 主机的在线联系,如主机组态、上装和下装用户程序等。

### 10.1.6 建立、修改 PLC 通信参数

如果建立了计算机和 PLC 的在线联系,就可利用软件检查、设置和修改 PLC 的通信参数。步骤如下:

(1) 单击引导条中的系统块图标,或从“视图(View)”菜单中选择“系统块(System Block)”选项,将出现系统块对话框。

(2) 单击“通信口(Port(s))”选项卡。检查各参数,确认无误后单击确定。如果需要修改某些参数,可以先进行有关的修改,再单击“确认(OK)”键,待确认后退出。

(3) 单击工具条中的下载按钮,即可把修改后的参数下载到 PLC 主机。

## 10.2 软件功能

### 10.2.1 基本功能

STEP7-Micro/WIN 32 的基本功能是协助用户完成开发应用软件的任务,例如创建用户程序、修改和编辑原有的用户程序,编辑过程中编辑器具有简单语法检查功能。同时它还有一些工具性的功能,例如用户程序的文档管理和加密等。此外,还可直接用软件设置 PLC 的工作方式、参数和运行监控等。

程序编辑过程中的语法检查功能可以提前避免一些语法和数据类型方面的错误。

梯形图中的错误处的下方自动加红色曲线,语句表中错误行前有红色叉,且错误处的下方加红色曲线。

软件功能的实现可以在联机工作方式(在线方式)下进行,部分功能的实现也可以在离线工作方式下进行。

联机方式:有编程软件的计算机与 PLC 连接,此时允许两者之间做直接通信。

离线方式:有编程软件的计算机与 PLC 断开连接,此时能完成大部分基本功能。如编程、编译和调试程序系统组态等。

两者的主要区别是:联机方式下可直接针对相连的 PLC 进行操作,如上装和下载用户程序和组态数据等;而离线方式下不直接与 PLC 联系,所有程序和参数都暂时存放在磁盘上,等联机后再下载到 PLC 中。

### 10.2.2 界面

启动 STEP7 - Micro/WIN 32 编程软件,其主界面外观如图 10-2 所示。

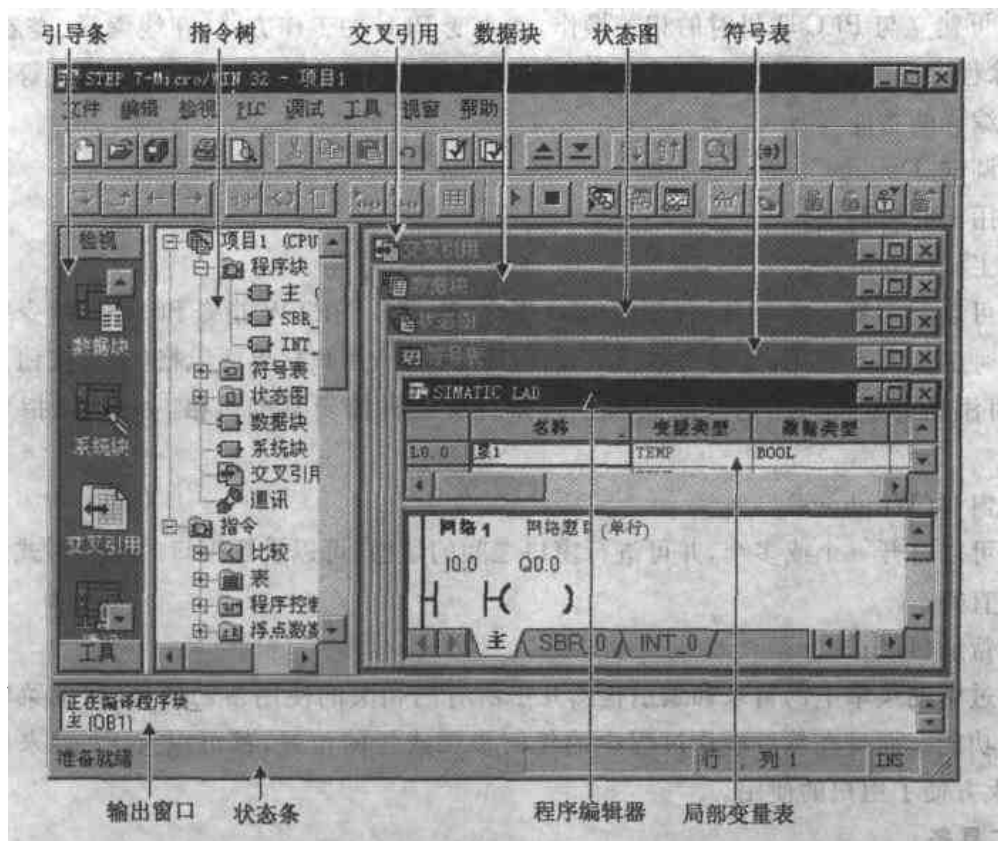


图 10-2 编程软件外观

界面一般可分以下几个区:菜单条(包含 8 个主菜单项)、工具条(快捷按钮)、引导条(快捷操作窗口)、指令树(Instruction Tree)(快捷操作窗口)、输出窗口和用户窗口(可同时或分别打开图中的 5 个用户窗口)。

除菜单条外,用户可根据需要决定其他窗口的取舍和样式的设置。

### 10.2.3 各部分功能

#### 1. 菜单条

允许使用鼠标单击或对应热键的操作,这是必选区。各主菜单项功能如下:

##### (1) 文件(File)

文件操作如新建、打开、关闭、保存文件,上装和下载程序,还有文件的打印预览、设置和操作等。

##### (2) 编辑(Edit)

程序编辑的工具,如选择、复制、剪切、粘贴程序块或数据块,同时提供查找、替换、插入、删除和快速光标定位等功能。

##### (3) 视图(View)

视图可以设置软件开发环境的风格,如决定其他辅助窗口(如引导窗口、指令树窗口、工具

条按钮区)的打开与关闭;包含引导条中所有的操作项目;选择不同语言的编程器(包括 LAD、STL、FBD 三种)。

#### (4) 可编程序控制器(PLC)

PLC 可建立与 PLC 联机时的相关操作,如改变 PLC 的工作方式、在线编译、查看 PLC 的信息、清除程序和数据、时钟、存储器卡操作、程序比较、PLC 类型选择及通信设置等。在此还提供离线编译的功能。

#### (5) 调试(Debug)

调试用于联机调试。

#### (6) 工具(Tools)

工具可以调用复杂指令向导(包括 PID 指令、NETR/NETW 指令和 HSC 指令),使复杂指令编程时工作大大简化;安装文本显示器 TD200 向导;用户化界面风格(设置按钮及按钮样式,在此可添加菜单项);用选项子菜单也可以设置 3 种编辑器的风格,如字体、指令盒的大小等。

#### (7) 窗口(Windows)

窗口可以打开一个或多个,并可进行窗口之间的切换;可以设置窗口的排放形式,如层叠、水平和垂直等。

#### (8) 帮助(Help)

它通过帮助菜单上的目录和索引检阅几乎所有的相关的使用帮助信息,帮助菜单还提供网上查询功能。而且在软件操作过程中的任何步骤或任何位置,都可以按 F1 键来显示在线帮助,大大方便了用户的使用。

### 2. 工具条

工具条提供简便的鼠标操作,将最常用的 STEP7 - Micro/WIN 32 操作以按钮形式设定到工具条。可以用“视图(View)”菜单中的“工具(Toolbars)”选项来显示或隐藏 3 种工具条:标准(Standard)、调试(Debug)和指令(Instructions)工具条。

### 3. 引导条

该条可用“视图(View)”菜单中的“引导条(Navigation Bar)”选项来选择是否打开。

它为编程提供按钮控制的快速窗口切换功能,包括程序块(Program Block)、符号表(Symbol Table)、状态图表(Status Chart)、数据块(Data Block)、系统块(System Block)、交叉索引(Cross Reference)和通信(Communication)。

单击任何一个按钮,则主窗口切换成此按钮对应的窗口。

引导条中的所有操作都可用“指令树(Instruction Tree)”窗口或“视图(View)”菜单来完成,可以根据个人的爱好来选择使用引导条或指令树。

### 4. 指令树

可用“视图(View)”菜单中“指令树(Instruction Tree)”的选项来选择是否打开,并提供编程时用到的所有快捷操作命令和 PLC 指令。

### 5. 交叉索引

它提供 3 个方面的索引信息,即交叉索引信息、字节使用情况信息和位使用情况信息。使编程所用的 PLC 资源一目了然。

### 6. 数据块

该窗口可以设置和修改变量存储区内各种类型存储区的一个或多个变量值,并加注必要的注释说明。

### 7. 状态图表

该图表可在联机调试时监视各变量的值和状态。

### 8. 符号表

实际编程时为了增加程序的可读性,常用带有实际含义的符号作为编程元件代号,而不是直接使用元件在主机中的直接地址。例如编程中 Start 作为编程元件代号,而不用 I0.3。符号表可用来建立自定义符号与直接地址之间的对应关系,并可附加注释,可以使程序清晰易读。

### 9. 输出窗口

该窗口用来显示程序编译的结果信息,如各程序块(主程序、子程序的数量及子程序号、中断程序的数量及中断程序号)及各块的大小、编译结果有无错误、错误编码和位置等。

### 10. 状态条

状态条也称任务栏,与一般的任务栏功能相同。

### 11. 编程器

该编程器可用梯形图、语句表或功能图表编程器编写用户程序,或在联机状态下从 PLC 上装用户程序进行读程序或修改程序。

### 12. 局部变量表

每个程序块都对应一个局部变量表,在带参数的子程序调用中,参数的传递就是通过局部变量表进行的。

## 10.2.4 系统组态


使用 S7-200 编程软件,可以进行许多参数的设置和系统配置,如通信组态、设置数字量输入滤波、设置脉冲捕捉、输出表配置和定义存储器保持范围等。大家在实际工作中用到时可参考编程手册。

## 10.3 编 程

本节介绍如何使用 STEP7-Micro/WIN 32 编程软件进行编程,这是学习编程软件使用的重点。

### 10.3.1 程序文件操作

#### 1. 新建

建立一个程序文件,可用“文件(File)”菜单中的“新建(New)”命令,在主窗口将显示新建的程序文件主程序区;也可用工具条中的按钮来完成。如图 10-3 所示为一个新建程序文件的指令树,系统默认初始设置如下:

新建的程序文件以“项目 1(CPU221)”命名,括号内为系统默认 PLC 的型号。项目包含 7

个相关的块。其中程序块中有 1 个主程序,1 个子程序 SBR\_0 和 1 个中断程序 INT\_0。

用户可以根据实际编程需要做以下操作:

#### (1) 确定主机型号

首先要根据实际应用情况选择 PLC 型号。右击“项目 1(CPU 221)”图标,在弹出的按钮中单击“类型(Type)”,或用“PLC”菜单中的“类型(Type)”命令。然后在弹出的对话框中选择所用的 PLC 型号。

#### (2) 程序更名

项目文件更名:如果新建了一个程序文件,可用“文件(File)”菜单中“另存为(Save as)”命令,然后在弹出的对话框中键入希望的名称。

子程序和中断程序更名:在指令树窗口中,右击要更名的子程序或中断程序名称,在弹出的选择按钮中单击“重命名(Rename)”,然后键入名称。

主程序的名称一般用默认的 MAIN,任何项目文件的主程序只有一个。

#### (3) 添加一个子程序或一个中断程序

方法 1:在指令树窗口中,右击“程序块(Program Block)”图标,在弹出的选择按钮中单击“插入子程序(Insert Subroutine)”或“插入中断程序(Insert Interrupt)”项。

方法 2:用“编辑(Edit)”菜单中的“插入(Insert)”命令。


方法 3:在编辑窗口中单击编辑区,在弹出的菜单选项中选择“插入(Insert)”命令。

新生成的子程序和中断程序根据已有子程序和中断程序的数目,默认名称分别为 SBR\_n 和 INT\_n,用户可以自行更名。


#### (4) 编辑程序

编辑程序块中的任何一个程序,只要在指令树窗口中双击该程序的图标即可。

### 2. 打开已有文件

打开一个磁盘中已有的程序文件,可用“文件(File)”菜单中“打开(Open)”命令,在弹出的对话框中选择打开的程序文件,也可用工具条中的  按钮来完成。

### 3. 上 装

在已经与 PLC 建立通信的前提下,如果要上装 PLC 存储器中的程序文件,可用“文件(File)”菜单中“上装(Upload)”命令,也可用工具条中的  按钮来完成。

## 10.3.2 编辑程序

编辑和修改控制程序是程序员利用 STEP7—Micro/WIN 32 编程软件要做的最基本的工作,本软件有较强的编辑功能,本节只以梯形图编辑器为例介绍一些基本编辑操作。

下面以图 10-4 所示的梯形图程序为例,介绍程序的编辑过程和各種操作。



图 10-3 新建程序的结构

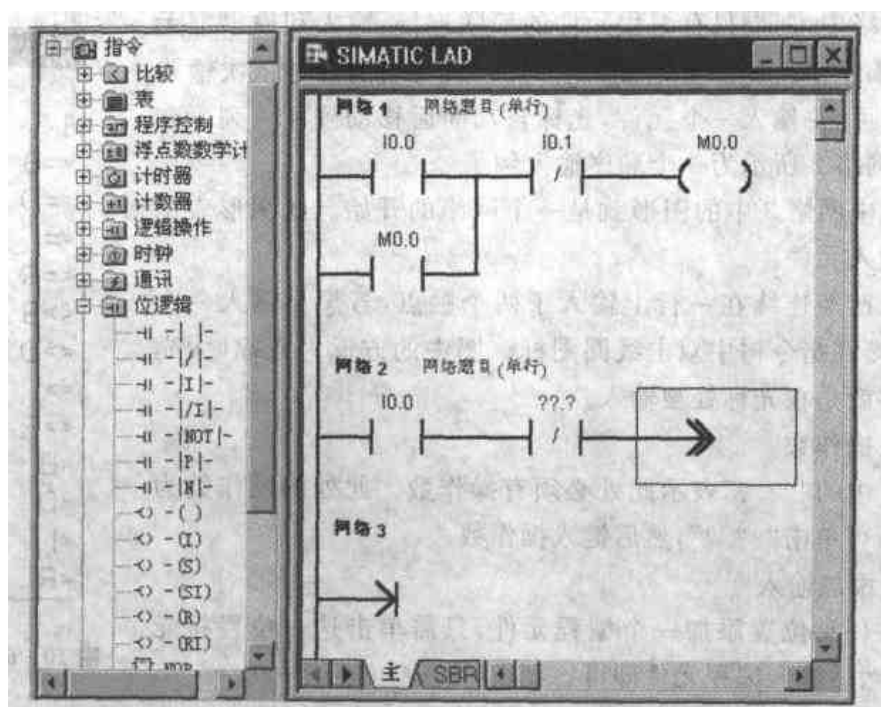


图 10-4 编程举例

### 1. 输入编程元件

梯形图的编程元件(编程元素)主要有线圈、触点、指令盒、标号及连接线。输入方法有以下两种。

方法 1: 用指令树窗口中的“指令(Instructions)”所列的一系列指令按类别分别编排在不同子目录中, 找到要输入的指令并双击, 如图 10-4 所示。

方法 2: 用指令工具条上的一组编程按钮, 单击触点、线圈和指令盒按钮, 从弹出的窗口中下拉菜单所列出的指令中选择要输入的指令单击即可。工具按钮和弹出的窗口下拉菜单如图 10-5 和图 10-6 所示。

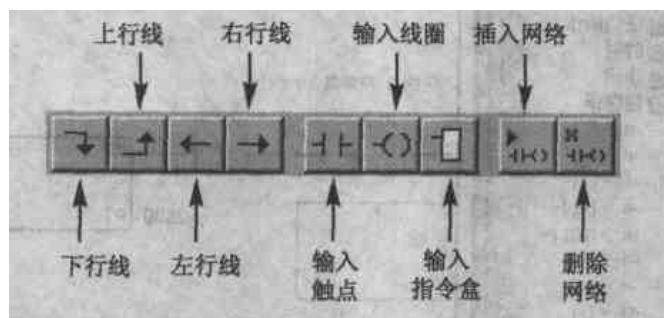


图 10-5 编程按钮

在指令工具条上, 编程元件的输入有 7 个按钮。下行线、上行线、左行线和右行线按钮, 用于输入连接线, 由此可形成复杂梯形图结构。输入触点、输入线圈和输入指令盒按钮用于输入编程元件。图 10-6 为单击输入触点按钮时弹出的下拉菜单。插入网络和删除网络按钮, 在编辑程序时使用。

#### (1) 顺序输入

在一个网络中,如果只有编程元件的串联连接,输入和输出都无分叉,则视作顺序输入。方法非常简单,只需从网络的开始依次输入各编程元件即可,每输入一个元件,光标自动向后移动到下一列。在图 10-4 中,网络 2 所示为一个顺序输入例子。

图 10-4 中网络 3 中的图形就是一个网络的开始。此图形表示可在此继续输入元件。

而网络 2 已经连续在一行上输入了两个触点,若想再输入一个线圈,可以直接在指令树中双击线圈图标。图中的方框为光标(大光标),编程元件就是在光标处被输入。

### (2) 输入操作数

图 10-4 中的“??.”表示此处必须有操作数。此处的操作数为触点的名称。可单击“??.”,然后键入操作数。

### (3) 任意添加输入

如果想在任意位置添加一个编程元件,只需单击这一位置将光标移到此处,然后输入编程元件即可。

## 2. 复杂结构

用指令工具条中的编程按钮(图 10-5 所示),可编辑复杂结构的梯形图,本例中的实现如图 10-7 所示。方法是单击图中第一行下方的编程区域,则在本行下一行的开始处显示光标(图中方框),然后输入触点,生成新的一行。

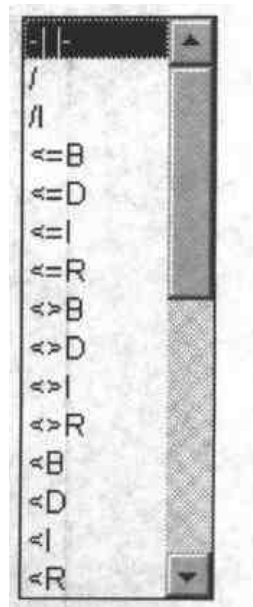


图 10-6 下拉菜单

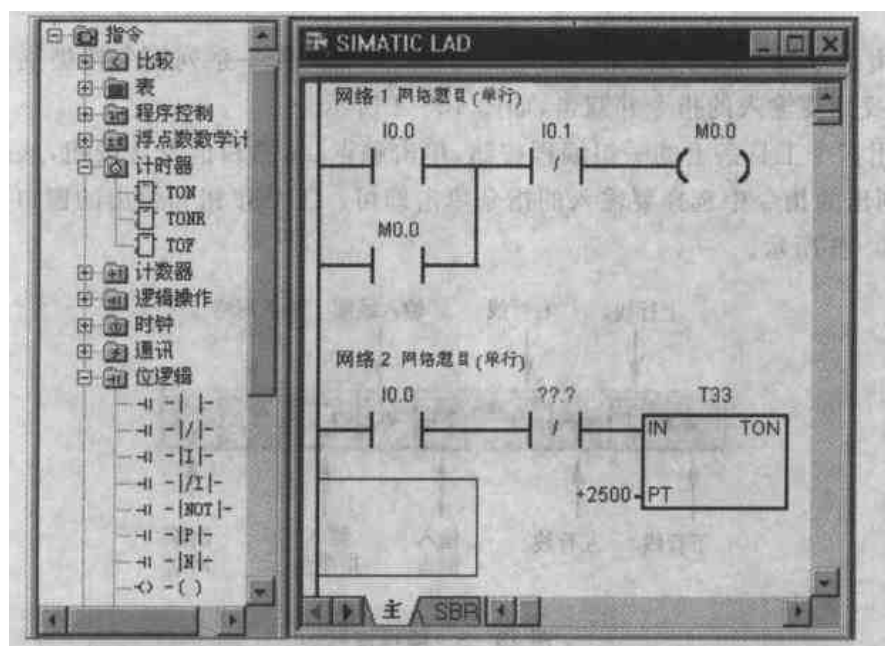



图 10-7 新生成行

输入完成后出现图 10-8 所示界面,将光标移到要合并的触点处,单击按钮  即可。


如果要在某一行的某个元件后向下分支,可将光标移到该元件,单击按钮 。然后便可在生成的分支顺序输入各元件。

图 10-8 向上合并

### 3. 插入和删除

编程中经常用到插入和删除一行、一列、一个网络、一个子程序或中断程序等。方法有两种：在编程区右击要进行操作的位置，弹出下拉菜单，选择“插入(Insert)”或“删除>Delete)”选项，再弹出子菜单，单击要插入或删除的项，然后进行编辑。也可用“编辑(Edit)”菜单中的命令进行上述相同的操作。

对于元件剪切、复制和粘贴等操作方法也与上述类似。

### 4. 块操作

利用块操作对程序大面积删除、移动、复制操作十分方便。块操作包括块选择、块剪切、块删除、块复制和块粘贴。这些操作非常简单，与一般字处理软件中的相应操作方法完全相同。

### 5. 符号表

使用符号表，可将直接地址编号用具有实际含义的符号代替，有利于程序结构清晰易读。具体使用可参考“帮助”中的相关内容。

### 6. 局部变量表

打开局部变量表的方法是，将光标移到编辑器的程序编辑区的上边缘，拖动上边缘向下，则自动显露出局部变量表，此时即可设置局部变量。使用带参数的子程序调用指令时会用到局部变量表，在此不再详述。

### 7. 注 释

梯形图编程器中的“网络 n(Network n)”标志每个梯级，同时又是标题栏，可在此为本梯级加标题或必要的注释说明，使程序清晰易读。方法：双击“网络 n”区域，弹出对话框，此时可以在“题目(Title)”文本框中键入标题，在“注释(Comment)”文本框键入注释。


### 8. 编程语言转换

软件可实现三种编程语言(编辑器)之间的任意切换。选择“视图(View)”菜单，然后单击 STL、LAD 或 FBD 便可进入对应的编程环境。使用最多的是 STL 和 LAD 之间的互相切换，STL 的编程可以按或不按网络块的结构顺序编程，但 STL 只有在严格按照网络块编程的格式下编程才可切换到 LAD，不然无法实现转换。

### 9. 编 译

程序编辑完成，可用“PLC”菜单中“编译(Compile)”命令进行离线编译。编译结束，在输出窗口显示编译结果信息。

## 10. 下 载

如果编译无误,便可单击下载按钮,把用户程序下载到 PLC 中。

## 10.4 调试及运行监控

STEP7 - Micro/WIN32 编程软件提供了一系列工具,可使用户直接在软件环境下调试并监视用户程序的执行。

### 10.4.1 选择扫描次数

选择单次或多次扫描来监视用户程序,可以指定主机以有限的扫描次数执行用户程序。通过选择主机扫描次数,当过程变量改变时,可以监视用户程序的执行。

#### (1) 多次扫描

方法:将 PLC 置于 STOP 模式。

使用“调试(Debug)”菜单中的“多次扫描(Multiple Scans)”命令,来指定执行的扫描次数,然后单击确认(OK)按钮进行监视。

#### (2) 初次扫描

将 PLC 置于 STOP 模式,使用“调试(Debug)”菜单中的“初次扫描(First Scans)”命令。

### 10.4.2 状态图表监控

可使用状态图表来监视用户程序,并可以用强制表操作修改用户程序中的变量。

#### 1. 使用状态图表

在引导条窗口中单击“状态图(Status Chart)”或用“视图(View)”菜单中的“状态图”命令。当程序运行时,可使用状态图来读、写、监视和强制其中的变量,如图 10-9 所示。



图 10-9 状态图表的监视

当用状态图表时,可将光标移到某一个单元格,右击单元格,在弹出的下拉菜单中单击一项,可实现相应的编辑操作。

根据需要,可建立多个状态图表。

状态图表的工具图标在编程软件的工具条区内。单击可激活这些工具图标,如顺序排序、逆序排序、全部写、单字读、读所有强制、强制和解除强制等。

## 2. 强制指定值

用户可以用状态图表来强制用指定值对变量赋值,所有强制改变的值都存到主机固定的 EEPROM 存储器中。

### (1) 强制范围


强制制定一个或所有的 Q 位;

强制改变最多 16 个 V 或 M 存储器的数据,变量可以是字节、字或双字类型;

强制改变模拟量映像存储器 AQ,变量类型为偶字节开始的字类型。


用强制功能取代了一般形式的读和写。同时,采用输出强制时,以某一个指定值输出,当主机变为 STOP 方式后输出将变为强制值,而不是设定值。

### (2) 强制一个值


若强制一个新值,可在状态图表的“新数值(New Value)”栏输入新值,然后单击工具条中的  按钮。

若强制一个已经存在的值,可以在“当前值(Current Value)”栏单击并点亮这个值,然后单击强制按钮。


### (3) 读所有强制操作

打开状态图表窗口,单击工具条中的  按钮,则状态图表中所有被强制的当前值的单元格中会显示强制符号。

### (4) 解除一个强制操作

在当前值栏单击点亮这个值,然后单击工具条中的  按钮。

### (5) 解除所有强制操作

打开状态图表,单击工具条中的  按钮。

## 10.4.3 运行模式下的编辑

在运行模式下编辑,可以在对控制过程影响较小的情况下,对用户程序做少量的修改。修改后的程序下载时,将立即影响系统的控制运行,所以使用时应特别注意。可进行这种操作的 PLC 有 CPU 224、CPU 226 和 CPU226XM 等。


操作步骤:

(1) 选择“调试(Debug)”菜单中的“在运行状态编辑程序(Program Edit in RUN)”命令,因为 RUN 模式下只能编辑主机中的程序,如果主机中的程序与编程软件窗口中的程序不同,系统会提示用户存盘。

### (2) 屏幕弹出警告信息

单击“继续(Continue)”按钮,所连接主机中的程序将被上装到编程主窗口,便可以在运行模式下进行编辑。

### (3) 在运行模式下进行下载

在程序编译成功后,可用“文件(File)”菜单中“下载(Download)”命令,或单击工具条中的下载按钮 ,将程序块下载到 PLC 主机。

### (4) 退出运行模式编辑

使用“调试(Debug)”菜单中的“在运行状态编辑程序(Program Edit in RUN)”命令,然后根据需求选择“选项(Checkmark)”中的内容。

#### 10.4.4 程序监视

利用三种程序编辑器(梯形图、语句表和函数表)都可在 PLC 运行时,监视程序的执行对各元件的执行结果,并可监视操作数的数值(本书只介绍梯形图和语句表的情况)。

##### 1. 梯形图监视

利用梯形图编辑器可以监视在线程序状态,如图 10-10 所示。图中被点亮的元件表示处于接通状态。

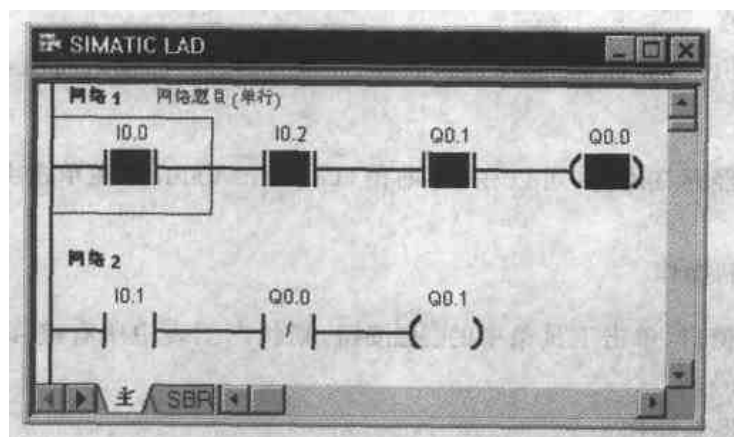




图 10-10 梯形图监视


梯形图中显示所有操作数的值,所有这些操作数状态都是 PLC 在扫描周期完成时的结果。在使用梯形图监控时,STEP7-Micro/WIN 32 编程软件不是在每个扫描周期都采集状态值在屏幕上的梯形图中显示,而是要间隔多个扫描周期采集一次状态值,然后刷新梯形图中各值的状态显示。在通常情况下,梯形图的状态显示不反映程序执行时的每个编程元素的实际状态。但这并不影响使用梯形图来监控程序状态,而且在大多数情况下,使用梯形图也是编程人员的首选。

实现方法是:用“工具(Tools)”菜单中的“选项(Options)”命令,打开选项对话框,选择“LAD 状态(LAD status)”选项卡,然后选择一种梯形图的样式。梯形图可选择的样式有 3 种:指令内部显示地址和外部显示值;指令外部显示地址和外部显示值;只显示状态值。然后打开梯形图窗口,在工具条中单击  程序状态按钮,即可进行梯形图监视。

##### 2. 语句表监视

用户可利用语句表编辑器监视在线程序状态。语句表程序状态按钮连续不断地更新屏幕上的数值,操作数按顺序显示在屏幕上,这个顺序与它们出现在指令中的顺序一致,当指令执行时,这些数值将被捕捉,它可以反映指令的实际运行状态。

实现方法是:单击工具栏上的程序状态按钮 ,出现图 10-11 所示的显示界面。其中,语句表的程序代码出现在左侧的 STL 状态窗口里,包含操作数的状态区显示在右侧。间接寻址的操作数将同时显示存储单元的值和它的指针。

可以用工具栏中的  按钮暂停,则当前的状态数据将保留在屏幕上,直到再次单击这个

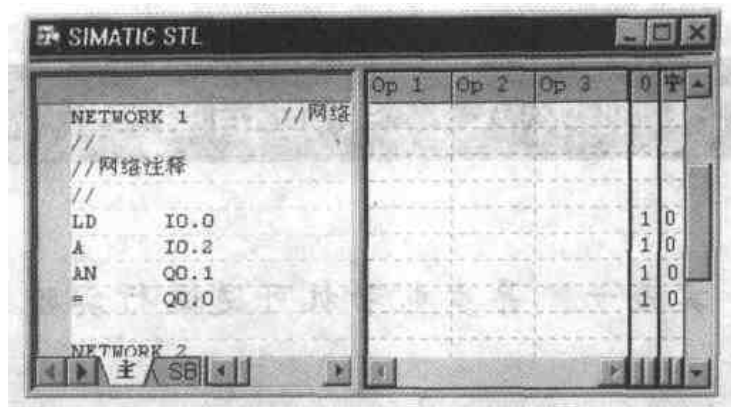


图 10-11 语句表监视

按钮。

图中状态数值的颜色表示指令执行状态:黑色表示指令正确执行;红色表示指令执行有错误;灰色表示指令由于栈顶值为 0 或由跳转指令使之跳过而没有执行;空白表示指令未执行。

可利用初次扫描得到第一个扫描周期的信息。

设置语句表状态窗口的样式:用“工具(Tools)”菜单中的“选项(Options)”命令,打开选项对话框,选择“STL 状态(STL status)”的选项卡,然后进行设置。

说明:目前 S7-200 的编程软件最新版本为 V3.2,它在 V3.1 版本基础上增加了一些新功能,如字符串指令、“复位/置位”和“置位/复位”双稳态锁存指令。在编辑用户程序方面做了改进,使用户使用更加方便。增加了用于位置控制和调制解调器模块的向导。更详细的使用方法请参考新编程软件中的“帮助”信息。

## 本章小结

(1) 用编程软件对 PLC 编程,首先要在计算机上安装 STEP7-Micro/Win32 编程软件,然后建立硬件连接并对通信参数进行设置,最后建立与 PLC 的在线联系与测试。

(2) 编程软件 STEP7-Micro/Win32 功能丰富、界面友好,且有方便的联机帮助功能。应掌握各项常用的功能。

(3) 程序编辑是学习编程软件的重点,可以用打开、新建、或从 PLC 上装程序文件,并对其编辑修改。编辑中应熟练使用菜单、常用按钮及各个功能窗口。符号表的应用可以使程序可读性大大提高,好的程序应加注必要的标题和注释。同一程序可以用梯形图、语句表和功能块图三种编辑器进行显示和编辑,并可直接切换。

(4) 使用状态图表可以强制设置和修改一些变量的值,实现程序调试。如果程序的改变对运行情况影响很小,可以在运行模式下编辑和修改程序及参数值。程序运行监控可用以下三种方法:梯形图法、功能块图法和语句表法,其中语句表监视可以反映指令的实际运行状态。

本章重点应掌握用编程软件 STEP7-Micro/Win 32 进行编程和程序调试的方法。

## 附录 A 实验指导书

### 实验一 异步电动机可逆运行实验

#### 一、实验目的

学习异步电动机控制电路的连接。

#### 二、实验内容

##### (1) 电子式时间继电器的延时整定。

让线圈通电,观察此时各触点的工作情况。调节“时间设定”旋钮,使延时开、闭触点在线圈通电后 10 s 时动作。

##### (2) 异步电动机的正、反转控制。

要求:按启动按钮,电机启动正转,10 s 后自行进入反转状态。按停止按钮,电机停转。

首先按实验图 A-1,分别完成主电路、控制电路的接线。经仔细检查后,先合上控制电路电源,观察接触器、时间继电器工作是否正常,完好时可合上主电路电源。按启动按钮,进行电机正、反转及停转实验。

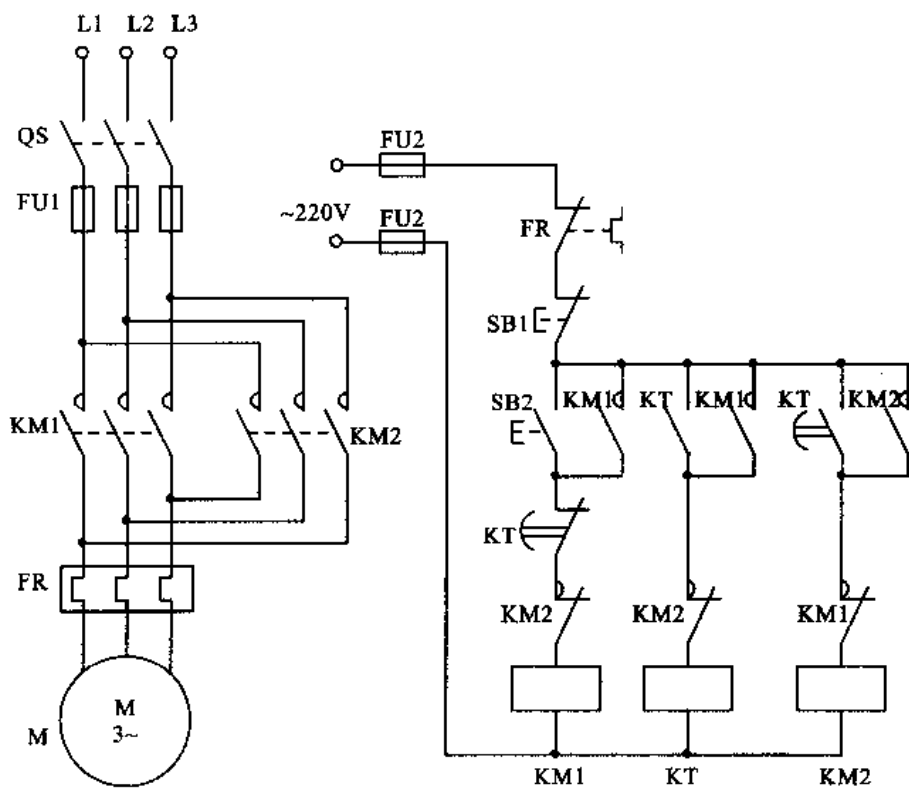


图 A-1 电动机可逆运行控制电路

### 三、预习要求

阅读实验指导书, 根据要求写出实验步骤。

### 四、实验报告要求

- (1) 指出电动机正、反转控制电路中哪些触点起“自锁”作用? 哪些触点起“联锁”作用?
- (2) 写出电气控制电路中各个电器的动作顺序过程。

## 实验二 S7-200 PLC 编程软件使用实验

### 一、实验目的

- (1) 熟悉 STEP 7-Micro/WIN 32 编程软件。
- (2) 上机编制简单的梯形图程序。
- (3) 初步掌握编程软件的使用方法和调试程序的方法。

### 二、实验内容

- (1) 熟悉编程软件的菜单、工具条、指令输入和程序调试。
- (2) 参照第五章的典型电路程序设计的内容, 编写一段简单程序。
- (3) 将程序写入 PLC, 检查无误后运行该程序, 并观察运行结果。

### 三、预习要求

阅读实验指导书, 根据要求设计一段简单程序, 并写出调试步骤。

### 四、实验报告要求

整理出运行调试后的梯形图程序, 写出该程序的调试步骤和观察结果。

## 实验三 抢答器程序设计实验

### 一、实验目的

- (1) 熟悉 S7-200 系列 PLC 的逻辑指令。
- (2) 编制简单的梯形图程序。
- (3) 进一步掌握编程软件的使用方法和调试程序的方法。

### 二、实验内容

- (1) 简单的抢答显示程序的调试。

参加智力竞赛的 A、B、C 三人的桌上各有一只抢答按钮, 分别为 SB1、SB2 和 SB3, 用 3 盏灯 HL1、HL2 和 HL3 显示他们的抢答信号。当主持人接通抢答允许开关 SW 后抢答开始, 最先按下按钮的抢答者对应的灯亮, 与此同时, 应禁止另外两个抢答者的灯亮, 指示灯在主持人断开开关 SW 后熄灭。与各外部输入、输出元件对应的 PLC 输入、输出端子号如表 A-1 所列。

表 A-1 输入、输出地址分配表

输入装置	端子号	输出装置	端子号
按钮 SB1	I0.0	灯 HL1	Q0.0
按钮 SB2	I0.1	灯 HL2	Q0.1
按钮 SB3	I0.2	灯 HL3	Q0.2
开关 SW	I0.3	—	—

将程序写入 PLC, 检查无误后运行该程序。调试程序时应该逐项检查以下要求是否满足:

- SW 没有接通时, 各按钮是否能使对应的灯亮。
- SW 接通时, 按某一个按钮是否能使对应的灯亮。
- 某一盏灯亮后, 另外两个抢答者的灯是否还能被点亮。
- 断开开关 SW, 是否能使已亮的灯熄灭。

如果某一项要求没有达到, 应检查和修改程序, 直到完全满足要求为止。

(2) 复杂的抢答显示程序的设计。

抢答者分为三组: 儿童组 2 人(分开坐), 他们控制按钮 SB11 和 SB12, 其中任何一个被按下, 灯 HL1 都亮; 学生组一人, 用按钮 SB2 控制灯 HL2; 教授组 2 人(分开坐), 当他们同时按下按钮 SB31 和 SB32 时灯 HL3 才亮。主持人按下复位按钮 SB4, 亮的灯熄灭。主持人接通开关 SW 后, 在 10 s 内如果参赛者按下按钮, 电磁开关接通, 使彩球摇动; SW 断开后停止摇动。与输入、输出对应的元器件地址如表 A-2 所列。

表 A-2 输入、输出地址分配表

输入装置	元件号	输出装置	元件号
按钮 SB11	I0.0	灯 HL1	Q0.0
按钮 SB12	I0.1	灯 HL2	Q0.1
按钮 SB2	I0.2	灯 HL3	Q0.2
按钮 SB31	I0.3	电磁开关	Q0.3
按钮 SB32	I0.4	—	—
按钮 SB4	I0.5	—	—
开关 SW	I0.6	—	—

### 三、预习要求

阅读实验指导书, 根据要求设计出抢答程序的梯形图和语句表程序, 并写出调试步骤。

### 四、实验报告要求

整理出运行调试后的抢答显示梯形图程序, 写出该程序的调试步骤和观察结果。

## 实验四 人行道按钮控制交通灯程序设计实验

### 一、实验目的

- (1) 进一步熟悉 PLC 的指令系统, 重点是功能图的编程、定时器和计数器的应用。
- (2) 熟悉时序控制程序的设计和调试方法。

### 二、实验内容

- (1) 只考虑横道线交通灯的控制程序。

某人行横道设有红、绿两盏信号灯, 一般是红灯亮, 路边设有按钮 SB1, 行人横穿街道时需按一下按钮。4 s 后红灯灭, 绿灯亮, 过 5 s 后, 绿灯闪烁 4 次(0.5 s 亮、0.5 s 灭), 然后红灯又亮, 时序如图 A-2 所示。

从按下按钮后到下一次红灯亮之前这一段时间内按钮不起作用。根据时序要求设计出红灯、绿灯的控制电路。将设计的程序写入 PLC, 检查无误后运行程序。用 I0.0 对应的开关模



## 实验五 水位控制程序设计实验

### 一、实验目的

- (1) 熟悉 PLC 的功能指令。
- (2) 熟悉对模拟量的采样方法。
- (3) 熟悉对模拟量处理的常用方法。

### 二、实验内容

用水泵通过一调节阀给一水池供水,水池中用一液位变送器测量水池水位(变送器输出 4~20 mA 电流信号,表示水池中水位的深度)。

(1) 对液位变送器(AIW0)的输出进行采样,要求采样周期为一个扫描周期,多次采样后求得平均值,折算为水池液位。设定一个水池水位,应用 PID 指令控制调节阀,保证水池水位保持在设定值。

(2) 应用定时中断方法,设定采样周期为 100 ms,多次采样后求得平均值,折算为水池液位。

### 三、预习要求

阅读本实验的指导书,编写符合实验内容(1)、(2)项要求的梯形图和语句表程序,并在梯形图上加上必要的注释。

### 四、实验报告要求

整理出调试好的模拟量采集和模拟量输出控制的梯形图程序和语句表程序。

## 附录 B S7-200 PLC 快速参考信息

表 B-1 S7-200 PLC 的 CPU 规范

	CPU 221	CPU 222	CPU 224	CPU 226	CPU 226XM
电 源					
输入电压	20.4~28.8 VDC / 85~264 VAC(47~63 Hz)				
24 VDC 传感器电源容量	180 mA		280 mA	400 mA	
存 储 器					
用户程序空间	2 048 字		4 096 字		8 192 字
用户数据(EEPROM)	1 024 字(永久存储)		2 560 字(永久存储)		5 120 字 (永久存储)
装备(超级电容)	50 小时/典型值(40℃时最少 8 小时)		190 小时/典型值(40℃时最少 120 小时)		
(可选电池)	200 天/典型值		200 天/典型值		
I/O					
本机数字输入/输出	6 输入/4 输出	8 输入/6 输出	14 输入/10 输出	24 输入/16 输出	
数字 I/O 映像区	256(128 入/128 出)				
模拟 I/O 映像区	无	32(16 入/16 出)	64(32 入/32 出)		
允许最大的扩展模块	无	2 模块	7 模块		
允许最大的智能模块	无	2 模块	7 模块		
脉冲捕捉输入	6	8	14		
高速计数	4 个计数器	6 个计数器			
单 相	4 个 30 kHz	6 个 30 kHz			
两 相	2 个 20 kHz	4 个 30 kHz			
脉冲输出	2 个 20 kHz(仅限于 DC 输出)				
常 规					
定时器	256 个定时器;4 个定时器(1 ms);16 个定时器(10 ms);236 定时器(100 ms)				
计数器	256(由超级电容器或电池备份)				
内部存储器位	256(由超级电容器或电池备份)				
掉电保护	112(存储在 EEPROM)				
时间中断	2 个 1 ms 的分辨率				
边沿中断	4 个上升沿和/或 4 个下降沿				
模拟电位器	1 个 8 位分辨率		2 个 8 位分辨率		
布尔量运算执行速度	0.37 $\mu$ s 每条指令				
时 钟	可选卡件		内 置		

续表 B-1

	CPU 221	CPU 222	CPU 224	CPU 226	CPU 226XM
卡件选项	存储卡、电池卡和时钟卡		存储卡和电池卡		
集成的通信功能					
接口	一个 RS-485 口			两个 RS-485 口	
PPI、DP/T 波特率	9.6、19.2、187.5 K 波特				
自由口波特率	1.2 K~115.2 K 波特				
每段最大电缆长度	使用隔离的中继器:187.5 K 波特可达 1 000 米,38.4 K 波特可达 1 200 米 未使用中继器:50 米				
最大站点数	每段 32 个站,每个网络 126 个站				
最大主站数	32				
点到点(PPI 主站模式)	是(NETR/NETW)				
MPI 连接	共 4 个,2 个保留(1 个给 PG,1 个给 OP)				

表 B-2 S7-200 PLC 的 CPU 输入规范

常 规	24 VDC 输入	
类 型	漏型/源型(IEC 类型 1 漏型)	
额定电压	24 VDC,4 mA 典型值	
最大持续允许电压	30 VDC	
浪涌电压	35 VDC,0.5 s	
逻辑 1(最小)	15 VDC,2.5 mA	
逻辑 0(最大)	5 VDC,1 mA	
输入延迟	可选(0.2 至 12.8 ms) CPU226,CPU226XM:输入点 I1.6~I2.7 具有固定延迟(4.5 ms)	
连接 2 线接近开关传感器(Bero)允许漏电流	最大 1 mA	
隔离(现场与逻辑) 光电隔离 隔离组	是 500 VAC,1 分钟 见接线图	
高速输入速率(最大) 逻辑 1=15~30 VDC 逻辑 1=15~26 VDC	单 相 20 kHz 30 kHz	两 相 10 kHz 20 kHz
同时接通的输入	55℃时所有的输入	
电线长度(最大) 屏 蔽 非屏蔽	普通输入 500 米,HSC 输入 50 米 普通输入 300 米	

表 B-3 S7-200 PLC 的 CPU 输出规范

常 规	24 VDC 输出	继电器输出
类 型	固态 - MOSFET	干触点
额定电压	24 VDC	24 VDC 或 250 VAC
电压范围	20.4~28.8 VDC	5~30 VDC 或 5~250 VAC
浪涌电流(最大)	8 A, 100 ms	7A 触点闭合
逻辑 1(最小)	20 VDC, 最大电流	—
逻辑 0(最大)	0.1 VDC, 10 K $\Omega$ 负载	—
每点额定电流(最大)	0.75 A	2.0 A
每个公共端的额定电流(最大)	6 A	10 A
漏电流(最大)	10 $\mu$ A	—
灯负载(最大)	5 W	30 WDC; 200 WAC
感性负载电压	L+ 减 48 VDC, 1W 功耗	—
接通电阻(接点)	0.3 $\Omega$ 最大	0.2 $\Omega$ (新的时候的最大值)
隔 离		
光电隔离(现场到逻辑)	500 VAC, 1 分钟	—
逻辑到接点	—	1 500 VAC, 1 分钟
接点到接点	—	750 VAC, 1 分钟
电阻(逻辑到接点)	—	100 M $\Omega$
隔离组	见接线图	见接线图
延 时	2/10 $\mu$ s(Q0.0 和 Q0.1)	—
断开到接通/接通到断开(最大)	15/100 $\mu$ s(其他)	—
切换(最大)	—	10 ms
脉冲频率(最大)Q0.0 和 Q0.1	20 kHz	1 Hz
机械寿命周期	—	10 000 000(无负载)
触点寿命	—	100 000(额定负载)
同时接通的输出	55℃时, 所有的输出	55℃时, 所有的输出
两个输出并联	是	否
电缆长度(最大)		
屏 蔽	500 m	500 m
非屏蔽	150 m	150 m

表 B-4 S7-200 PLC 的 CPU 存储器范围和特性总汇

描 述	范 围				存取格式			
	CPU 221	CPU 222	CPU 224	CPU 226/226XM	位	字节	字	双字
用户程序区	2 K 字	2 K 字	4 K 字	4/8 K 字				
用户数据区	1 K 字	1 K 字	2.5 K 字	2.5/5K 字				
输入映像寄存器	I0.0~I15.7	I0.0~I15.7	I0.0~I15.7	I0.0~I15.7	Ix.y	IBx	IWx	IDx
输出映像寄存器	Q0.0~Q15.7	Q0.0~Q15.7	Q0.0~Q15.7	Q0.0~Q15.7	Qx.y	QBx	QWx	QDx
模拟输入(只读)	—	AIW0~AIW30	AIW0~AIW30	AIW0~AIW30			AIWx	
模拟输出(只写)	—	AQW0~AQW30	AQW0~AQW30	AQW0~AQW30			AQWx	
变量存储器	VB0.0~VB2047.7	VB0.0~VB2047.7	VB0.0~VB5119.7	VB0.0~VB5119.7	Vx.y	VBx	VWx	VDx
局部存储器 <sup>1</sup>	LB0.0~LB63.7	LB0.0~LB63.7	LB0.0~LB63.7	LB0.0~LB63.7	Lx.y	LBx	LWx	LDx
位存储器	M0.0~M31.7	M0.0~M31.7	M0.0~M31.7	M0.0~M31.7	Mx.y	MBx	MWx	MDx
特殊存储器 只读	SM0.0~SM179.7 SM0.0~SM29.7	SM0.0~SM179.7 SM0.0~SM29.7	SM0.0~SM179.7 SM0.0~SM29.7	SM0.0~SM179.7 SM0.0~SM29.7	SMx.y	SMBx	SMWx	SMDx
定时器	256 (T0~T255)	256 (T0~T255)	256 (T0~T255)	256 (T0~T255)	Tx		Tx	
保持接通延时 1 ms	T0, T64	T0, T64	T0, T64	T0, T64				
保持接通延时 10 ms	T1~T4, T65~T68	T1~T4, T65~T68	T1~T4, T65~T68	T1~T4, T65~T68				
保持接通延时 100 ms	T5~T31, T69~T95	T5~T31, T69~T95	T5~T31, T69~T95	T5~T31, T69~T95				
接通/断开延时 1 ms	T32, T96	T32, T36	T32, T96	T32, T96				
接通/断开延时 10 ms	T33~T36, T97~T100	T33~T36, T97~T100	T33~T36, T97~T100	T33~T36, T97~T100				
接通/断开延时 100 ms	T37~T63, T101~T255	T37~T63, T101~T255	T37~T63, T101~T255	T37~T63, T101~T255				
计数器	C0~C255	C0~C255	C0~C255	C0~C255	Cx		Cx	
高速计数器	HC0, HC3~HC5	HC0, HC3~HC5	HC0~HC5	HC0~HC5				HCx
顺控继电器	S0.0~S31.7	S0.0~S31.7	S0.0~S31.7	S0.0~S31.7	Sx.y	SBx	SWx	SDx
累加器	AC0~AC3	AC0~AC3	AC0~AC3	AC0~AC3		ACx	ACx	ACx
跳转/标号	0~255	0~55	0~255	0~255				
调用/子程序	0~63	0~63	0~63	0~63				
中断程序	0~127	0~127	0~127	0~127				
回 路	0~7	0~7	0~7	0~7				
通信口	0	0	0	0, 1				

注 1: LB60 到 LB63 为 STEP 7 - Micro/WIN 32 V3.0 或更高版本保留

表 B-5 S7-200 PLC[CPU(V1.21)]指令系统速查表

布尔指令			NOT	堆栈取反
LJ Bit	取		EU	上升沿脉冲
LDI Bit	立即取		ED	下降沿脉冲
LDN Bit	取反		— Bit	输出
LDNI Bit	立即取反		=I Bit	立即输出
A Bit	与		S Bit, N	置位一个区域
AI Bit	立即与		R Bit, N	复位一个区域
AN Bit	与反		SI Bit, N	立即置位一个区域
ANI Bit	立即与反		RI Bit, N	立即复位一个区域
O Bit	或		(无 STL 指令形式)	置位优先触发器指令(SR)
OI Bit	立即或		(无 STL 指令形式)	复位优先触发器指令(RS)
ON Bit	或反		实时时钟指令	
ONI Bit	立即或反		TODR T	读实时时钟
LDBx IN1, IN2	装载字节比较的结果 IN1 (x₁<, <=, =, >=, >, <>) IN2		TODW T	写实时时钟
ABx IN1, IN2	与字节比较的结果 IN1 (x₁<, <=, =, >=, >, <>) IN2		字符串指令	
OBx IN1, IN2	或字节比较的结果 IN1 (x₁<, <=, =, >=, >, <>) IN2		SLEN IN, OUT	字符串长度
LDWx IN1, IN2	装载字比较结果 IN1 (x₁<, <=, =, >=, >, <>) IN2		SCAT IN, OUT	连接字符串
AWx IN1, IN2	与字比较结果 IN1 (X₁<, <=, =, >=, >, <>) IN2		SCPY IN, OUT	复制字符串
OWx IN1, IN2	或字比较结果 IN1 (x₁<, <=, =, >=, >, <>) IN2		SSCPY IN, INDX, N, OUT	复制子字符串
LDDx IN1, IN2	装载双字比较结果 IN1 (x₁<, <=, =, >=, >, <>) IN2		CFND IN1, IN2, OUT	在字符串中查找第一个字符
ADx IN1, IN2	与双字比较结果 IN1 (x₁<, <=, =, >=, >, <>) IN2		SFND IN1, IN2, OUT	在字符串中查找字符串
ODx IN1, IN2	或双字比较结果 IN1 (x₁<, <=, =, >=, >, <>) IN2		数学、增减指令	
LDRx IN1, IN2	装载实数比较结果 IN1 (x₁<, <=, =, >=, >, <>) IN2		+I IN1, OUT	整数加法: IN1+OUT=OUT
ARx IN1, IN2	与实数比较结果 IN1 (x₁<, <=, =, >=, >, <>) IN2		+D IN1, OUT	双整数加法: IN1+OUT=OUT
ORx IN1, IN2	或实数比较结果 IN1 (x₁<, <=, =, >=, >, <>) IN2		+R IN1, OUT	实数加法: IN1+OUT=OUT
LDSx IN1, IN2	装载字符串比较结果 IN1 (x₁<, <>) IN2		-I IN1, OUT	整数减法: OUT-IN1=OUT
ASx IN1, IN2	与字符串比较结果 IN1 (x₁=, <>) IN2		-D IN1, OUT	双整数减法: OUT-IN1=OUT
OSx IN1, IN2	或字符串比较结果 IN1 (x₁=, <>) IN2		-R IN1, OUT	实数减法: OUT-IN1=OUT
			MUL IN1, OUT	完全整数乘法: IN1×OUT=OUT
			*I IN1, OUT	整数乘法: IN1×OUT=OUT
			*D IN1, OUT	双整数乘法: IN1×OUT=OUT
			*R IN1, OUT	实数乘法: IN1×OUT=OUT
			DIV IN1, OUT	安全整数除法: OUT/IN1=OUT
			/I IN1, OUT	整数除法: OUT/IN1=OUT
			/D IN1, OUT	双整数除法: OUT/IN1=OUT
			/R IN1, OUT	实数除法: OUT/IN1=OUT
			SQRT IN, OUT	平方根
			LN IN, OUT	自然对数
			EXP IN, OUT	自然指数
			SIN IN, OUT	正弦
			COS IN, OUT	余弦
			TAN IN, OUT	正切

续表 B-6

INCB	OUT	字节增 1	RRB	OUT, N	字节循环右移
INCW	OUT	字增 1	RRW	OUT, N	字循环右移
INCD	OUT	双字增 1	RRD	OUT, N	双字循环右移
DECB	OUT	字节减 1	RLB	OUT, N	字节循环左移
DECW	OUT	字减 1	RLW	OUT, N	字循环左移
DECD	OUT	双字减 1	RLD	OUT, N	双字循环左移
PID	TBL, LOOP	PID 回路	FILL	IN, OUT, N	用指定的元素填充存储器空间
定时器和计数器指令			逻辑操作		
TON	Txxx, PT	接通延时定时器	ALD		与一个组合
TOF	Txxx, PT	关断延时定时器	OLD		或一个组合
TONR	Txxx, PT	带记忆的接通延时定时器	LPS		逻辑堆栈(堆栈控制)
CTU	Cxxx, PV	增计数	LRD		读逻辑栈(堆栈控制)
CTD	Cxxx, PV	减计数	LPP		逻辑出栈(堆栈控制)
CTUD	Cxxx, PV	增/减计数	LDS		装入堆栈(堆栈控制)
程序控制指令			AENO		对 ENO 进行与操作
END		程序的条件结束	ANDB	IN1, OUT	字节逻辑与
STOP		切换到 STOP 模式	ANDW	IN1, OUT	字逻辑与
WDR		看门狗复位(300 ms)	ANDD	IN1, OUT	双字逻辑与
JMP	N	跳到定义的标号	ORB	IN1, OUT	字节逻辑或
LBL	N	定义一个跳转的标号	ORW	IN1, OUT	字逻辑或
CALL	N[N1, ...]	调用子程序[N1, ...可以有 16 个可选参数]	ORD	IN1, OUT	双字逻辑或
CRET		从子程序条件返回	XORB	IN1, OUT	字节逻辑异或
FOR	INDX, INIT, FINAL	For/Next 循环	XORW	IN1, OUT	字逻辑异或
NEXT			XORD	IN1, OUT	双字逻辑异或
LSCR	S_bit	顺控继电器段的启动	INVB	OUT	字节取反
SCRT	S_bit	状态转移	INVW	OUT	字取反
CSCRE		顺控继电器段条件结束	INVD	OUT	双字取反
SCRE		顺控继电器段结束	表指令		
传送、移位、循环和填充指令			ATT	DATA, TBL	把数据加入到表中
MOVB	IN, OUT	字节传送	LIFO	TBL, DATA	从表中取数据 (后进先出)
MOVW	IN, OUT	字传送	FIFO	TBL, DATA	从表中取数据 (先进先出)
MOVD	IN, OUT	双字传送	FND=	TBL, PATRN, INDX	根据比较条件在表中 查找数据
MOVR	IN, OUT	实数传送	FND<>	TBL, PATRN, INDX	
BIR	IN, OUT	字节立即读	FND<	TBL, PATRN, INDX	
BIW	IN, OUT	字节立即写	FND>	TBL, PATRN, INDX	
BMB	IN, OUT, N	字节块传送	转换指令		
BMW	IN, OUT, N	字块传送	BCDI	OUT	BCD 码转换成整数
BMD	IN, OUT, N	双字块传送	IBCD	OUT	整数转换成 BCD 码
SWAP	IN	交换字节	BTI	IN, OUT	字节转换成整数
SHRB	DATA, S_BIT, N	寄存器移位	ITB	IN, OUT	整数转换成字节
SRB	OUT, N	字节右移	ITD	IN, OUT	整数转换成双整数
SRW	OUT, N	字右移	DTI	IN, OUT	双整数转换成整数
SRD	OUT, N	双字右移	DTR	IN, OUT	双字转换成实数
SLB	OUT, N	字节左移	TRUNC	IN, OUT	实数转换成双字(舍去 小数)
SLW	OUT, N	字左移			
SLD	OUT, N	双字左移			

续表 B-6

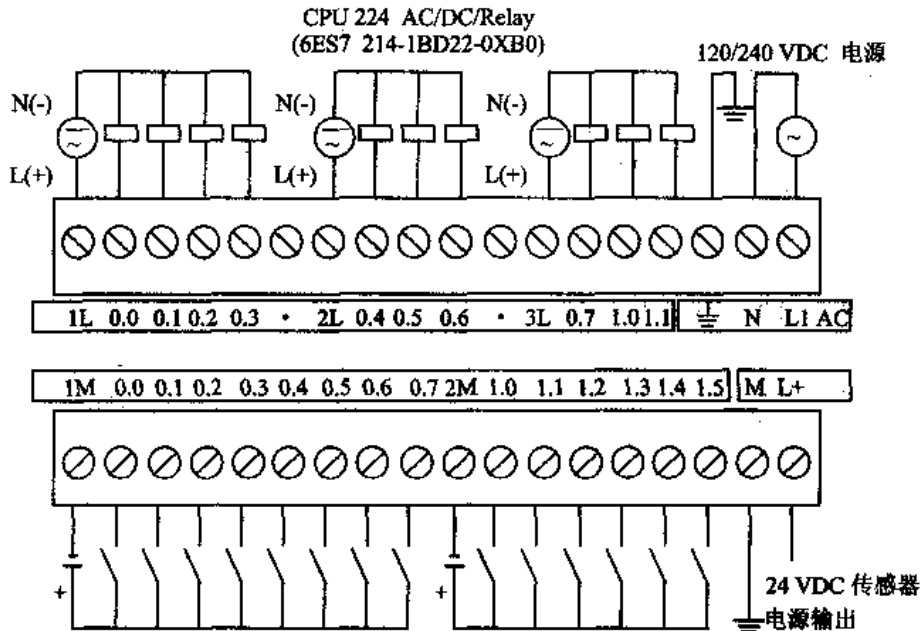
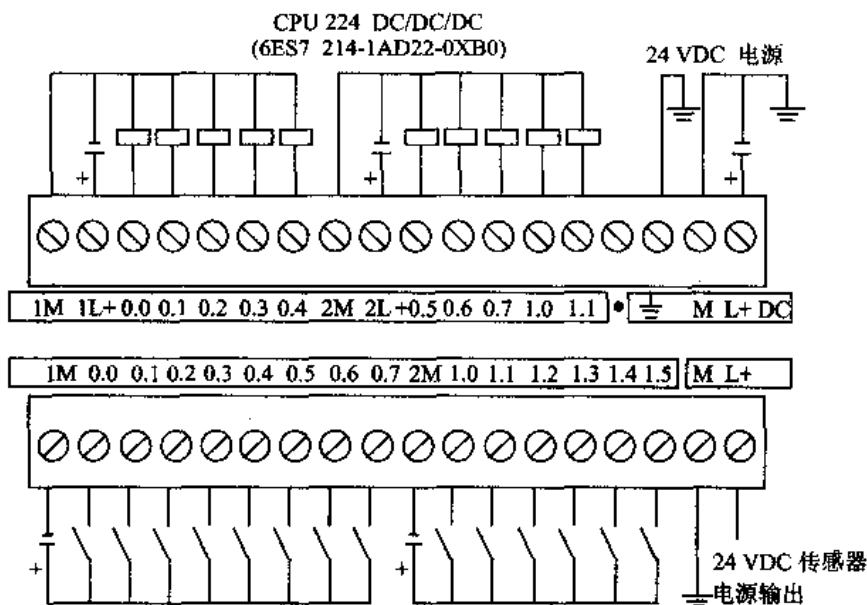
ROUND IN, OUT		实数转换成双整数(保留小数)	中 断	
ATH	IN, OUT, LEN	ASCII 码转换成 16 进制格式	CRETI	从中断条件返回
HTA	IN, OUT, LEN	16 进制格式转换成 ASCII 码	ENI	允许中断
ITA	IN, OUT, FMT	整数转换成 ASCII 码	DISI	禁止中断
DTA	IN, OUT, FMT	双整数转换成 ASCII 码	ATCH	INT, EVNT
RTA	IN, OUT, FMT	实数转换成 ASCII 码	DTCH	EVNT
ITS	IN, FMT, OUT	整数转换为字符串	通 信	
DTS	IN, FMT, OUT	双整数转换为字符串	XMT	TBL, PORT
RTS	IN, FMT, OUT	实数转换为字符串	RCV	TBL, PORT
STI	IN, INDX, OUT	字符串转换为整数	NETR	TBL, PORT
STD	IN, INDX, OUT	字符串转换为双整数	NETW	TBL, PORT
STR	IN, INDX, OUT	字符串转换为实数	GPA	ADDR, PORT
DECO	IN, OUT	解 码	SPA	ADDR, PORT
ENCO	IN, OUT	编 码	高速指令	
SEG	IN, OUT	产生 7 段码显示器格式	HDEF	HSC, MODE
			HSC	N
			PLS	Q

表 B-6 常用特殊继电器 SM0 和 SM1 的位信息

特殊存储器位			
SM0.0	该位始终为 ON	SM1.0	执行某些指令,结果为 0 时置位
SM0.1	首次扫描时为 ON,常用做初始化脉冲	SM1.1	执行某些指令,结果溢出或非法数值时置位
SM0.2	保持数据丢失时为 ON 一个扫描周期,可用做错误存储器位	SM1.2	执行运算指令,结果为负数时置位
SM0.3	开机进入 RUN 时为 ON 一个扫描周期,可在不断电的情况下代替 SM0.1 的功能	SM1.3	试图除以零时置位
SM0.4	时钟脉冲:30 s 闭合/30 s 断开	SM1.4	执行 ATT 指令,超出表范围时置位
SM0.5	时钟脉冲:0.5 s 闭合/0.5 s 断开	SM1.5	从空表中读数时置位
SM0.6	扫描时钟脉冲:闭合 1 个扫描周期/断开 1 个扫描周期	SM1.6	非 BCD 数转换为二进制数时置位
SM0.7	开关放置在 RUN 位置时为 1,在 TERM 位置为 0,常用在自由口通信处理中	SM1.7	ASCII 码到十六进制数转换出错时置位

图 B-1 CPU 224 外围典型接线图

了解 PLC 的外围接线图非常重要,它可以让初学者知道 PLC 和外界是如何联系的。这里我们选取的是 CPU 224 的外围接线图,其他 CPU 的接线图可参考 S7-200 系统手册。CPU 224 外围典型按线图如下所示。



## 参考文献

1. 方承远主编. 工厂电气控制技术. 北京: 机械工业出版社, 2000
2. 陈立定等编. 电气控制与可编程控制器. 广州: 华南理工大学出版社, 2001
3. 台方编. 可编程控制器应用教程. 北京: 中国水利水电出版社, 2001
4. SIEMENS 公司. SIMATIC S7-200 可编程序控制器系统手册. 2002. 10
5. 殷华文主编. 可编程序控制器及工业控制网络. 西安: 西安地图出版社, 2001
6. 王兆义编. 可编程序控制器教程. 北京: 机械工业出版社, 2000
7. 崔亚军. 可编程序控制器及程序设计. 北京: 电子工业出版社, 1993
8. 邱公伟主编. 可编程控制器网络通信及应用. 北京: 清华大学出版社, 2000
9. 廖常初编著. 可编程序控制器应用技术(第三版). 重庆: 重庆大学出版社, 2000
10. SIEMENS 公司. SIMATIC S7-200 应用示例. 1997
11. SIEMENS 公司. SIMATIC S7-200 应用示例(续). 1997
12. 皮壮行等编. 可编程序控制器的系统设计与应用实例. 北京: 机械工业出版社, 2000
13. 施耐德公司编. Altistart 46 User's manual
14. 西门子公司编. MICROMASTER 440 用户手册
15. 廖常初编著. PLC 编程及应用. 北京: 机械工业出版社, 2002. 9
16. 王仁祥编著. 常用低压电器原理及其控制技术. 北京: 机械工业出版社, 2002. 4
17. 刘金琪编著. 机床电气自动控制. 哈尔滨: 哈尔滨工业大学出版社, 1999
18. 刘光文编. 常用电器及电气标准. 成都: 电子科技大学出版社, 1994