

# PLC 语句表向梯形图自动转换的实现方法

吕 俊 白

( 华侨大学信息科学与工程学院, 福建 泉州 362021 )

**摘要** 实现可编程控制器语句表向梯形图的自动转换,是提高可编程控制器应用系统的维护和改造效率亟等解决的问题.文中通过对 PLC 指令及其对应的梯形图图符进行分析,总结出由 PLC 语句表生成梯形图的规则,并分析软件实现的主要数据结构及其关键技术.实验结果表明,软件能正确实现 PLC 语句表向梯形图的自动转换,可广泛应用于 PLC 应用系统的设计,改造和维护.

**关键词** 可编程控制器, 梯形图, 语句表, 有向图

**中图分类号** TP 311.52; TM 571.6<sup>+</sup> 1; TP 313

**文献标识码** A

可编程控制器 (Programmable Logic Controller, 简称 PLC 或 PC) 是以微处理器为基础,综合计算机技术、自动控制技术,以及通讯技术发展起来的新一代工业自动控制装置.它具有可靠性高、编程简单、使用方便、可在线修改、组合灵活和体积小等优点,在现代工业控制中占有重要地位.在 PLC 应用日益广泛的同时,系统的维护和改造成了人们关注的热点.系统的有效维护和改造,在很大程度上就取决于电气技术人员对其应用软件系统的熟悉程度.因此,在 PLC 应用日益广泛的同时,系统的维护和改造成了人们关注的热点.如何实现 PLC 语句表向电气技术人员所熟悉的梯形图的转换,就成为研究任务的核心.为便于 PLC 应用系统的设计、改造与维护,我们开发了一个软件系统,并将其应用于实现 PLC 语句表向梯形的自动转换.

## 1 语句表与梯形图

### 1.1 语句表

语句是应用程序的基础单元.应用系统的每个控制功能,均由一条或多条语句组成的应用程序来完成. PLC 语句表是一种与汇编语言类似的助记符编程表达式. PLC 的语句由操作码和操作数组成,其一般形式为操作码+ 操作数.操作码( 又称编程指令或编程命令)用助记符表示,它表明 PLC 要完成的某种操作功能.操作数包含为执行某种操作所必须的信息,告诉 PLC 用什么地方的数据来执行此操作.语句表编程有键入方便、编程灵活、能直接被 PLC 所识别等优点,但不如梯形图形象、直观.

### 1.2 梯形图

梯形图起源于继电器逻辑和执行线图,它与电气控制原理图相呼应,是 PLC 的主要编程手段.在梯形图中用不同的图符来表示不同的指令,用串、并联等概念组织图符的顺序位置来表述逻辑.梯形图的设计必须遵循以下 4 点规则<sup>[1]</sup>. (1) 触点应画在水平线上,不能画在垂直分支上,且应遵循自左至右、自上而下的原则. (2) 不包含触点的分支应放在垂直方向,不可放在水平位置,以便于识别触点的组合和对输出线圈的控制路径. (3) 在有几个串联回路相并联时,应将触点最多的那个串联回路放在梯形图的最上面;而在有几个并联回路相串联时,应将触点最多的并联回路放在梯形图的最左面. (4) 不能将触点画在线圈的右边,线圈仅能画在同一行中所有触点的最右边.

收稿日期 2004-10-15

**作者简介** 吕俊白(1969),女,副教授,硕士,主要从事数字图像处理与多媒体技术及过程控制的研究. E-mail: ltgs@hqu.edu.cn

## 2 系统设计与实现

### 2.1 梯形图图符库和 PLC 指令索引表的建立

本系统是基于 OMRON CPM1A 可编程控制器开发的. 该型号的可编程控制器共有 88 条指令(基本指令 15 条, 功能指令 73 条), 其中多数指令有图符与其对应<sup>[2]</sup>. 在设计中, 为使用户界面更加友好, 我们为每个图符的绘制定义了相应的自定义函数, 形成一个梯形图图符函数库. 为便于编辑及编译功能的实现, 以基本指令优先, 并考虑指令的使用频度, 建立了一张 PLC 指令索引表. 索引表包含索引号、指令助记符、参数个数、输入端个数、输出端个数、图符函数入口.

### 2.2 系统的主要数据结构

为充分利用 PLC 的在线修改功能, 方便下一步的系统改造, 在实现 PLC 语句表向梯形图的转换时所采用的数据结构应便于作必要的修改, 同时应易于进一步转换为 PLC 所能识别的语句表. 通过对梯形图及其设计规则的分析可以发现, 用梯形图编制程序的过程实质是一个用梯形图图符来表示操作指令, 用图符的串并联来表示操作指令之间的逻辑关系的过程. 由此, 一个梯形图可以看作是一个由梯形图图符和连接符组合而成的有向图, 其中的梯形图图符可抽象为有向图的顶点, 连接符可抽象为有向图的弧. 为了在程序设计中能更合理地描述梯形图, 我们综合考虑了提高程序执行效率、节省存储空间和便于操作等因素, 采用十字链表数据结构来存储转换后得到的梯形图.

2.2.1 有向图 图是一种数据结构, 它的形式化定义为<sup>[3]</sup>

$$\text{Graph} = (V, R)$$

其中  $V = \{x | x \in \text{Dataobject}\}$ ,  $R = \{VR\}$ ,  $VR = \{ \langle x, y \rangle | P(x, y) \wedge (x, y \in V) \}$ . 图中的数据元素通常称作顶点(Vertex),  $V$  是顶点的有穷非空集合;  $VR$  是两个顶点之间的关系的集合. 若  $\langle x, y \rangle \in VR$ , 则  $\langle x, y \rangle$  表示从  $x$  到  $y$  的一条弧(Arc), 且称  $x$  为弧尾(Tail), 称  $y$  为弧头(Head), 此时的图称为有向图(Directed Graph). 在有向图中, 以顶点  $v$  为头的弧的数目称为顶点  $v$  的入度(InDegree), 记为  $ID(v)$ ; 以顶点  $v$  为尾的弧的数目称为顶点  $v$  的出度(OutDegree), 记为  $OD(v)$ .

2.2.2 主要数据结构 设计中, 我们主要使用了以下数据结构. (1) 图符结构. 用于保存梯形图中各个图符的基本信息.

```
type//图符结构
nodeptr= ^node;
node= packed record
inst: string; //与该图符对应的指令
x, y: integer; //图符所在区域的左上角坐标
index: integer; //图符的索引号, 它是图符的唯一标识
par1, par2, par3: string; //3 个参数(梯形图的图符最多含 3 个参数)
indegree: integer; //入度
outdegree: integer; //出度
firstin: arclink; //指向以该顶点(图符)为弧头的第一个弧结点
firstout: arclink; //指向以该顶点(图符)为弧尾的第一个弧结点
visited: integer; //访问标志
end;
```

(2) 连接符结构. 用于描述相邻两个图符间的关系.

```
type//连接符结构
arclink= ^arctype;
arctype= packed record
tailvex: nodeptr; //指向弧尾顶点(图符)在图中的位置
headvex: nodeptr; //指向弧头顶点(图符)在图中的位置
hlink: arclink; //指向弧头相同的下一条弧
```

tlink: arclink; // 指向弧尾相同的下一条弧

end;

2.3 实现的技术要点

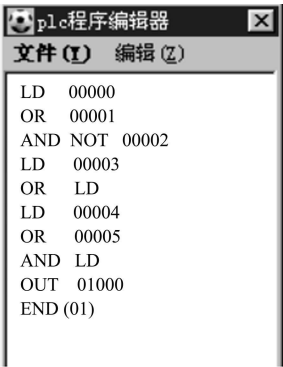
为使语句表转换而成的梯形图便于修改,我们将绘图区域分隔成若干正方形编辑格.同时,在对语句表程序进行识别并将它转换为梯形图时,在内存建立与梯形图相对应的十字链表,用于存放转换所得的梯形图<sup>[4,5]</sup>.

2.3.1 梯形图的生成 在设计中,对指令及其对应的梯形图符进行了详细的分析,总结出梯形图的生成规则.(1) 当前指令的输出端个数为零.指令队列中的所有元素出列,按规则绘制相应的图符.(2) 当前指令的输出端个数不为零.进一步判断是否为 OR, OR NOT 或 OR LD 指令.若是,则指令队列中的所有元素出列,按规则绘制相应的图符;否则,当前指令进指令队列.(3) 指令的输入端个数为零.绘制对应的梯形图符时应与左侧母线连接.(4) 指令的输入端个数为 1.绘制对应的梯形图符时应按程序的逻辑关系,建立正确的连接.(5) 指令的输入端个数为 2.绘制对应的梯形图符时,将占用同一列上的上一行和当前行 2 个编辑区,并把前一列的上一行和当前行中的两个梯形图图符相连接.(6) 指令的输入端个数为 3.绘制对应的梯形图符时,将占用同一列上的当前行及上两行 3 个编辑区;并把前一列的上两行及当前行中的 3 个梯形图符相连接;

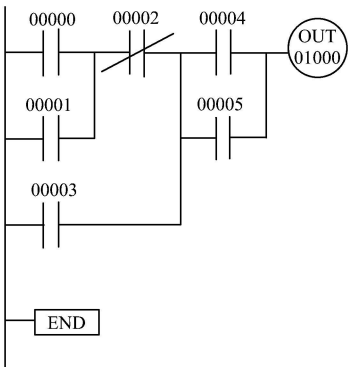
2.3.2 几个关键的技术问题 (1) LD 和 LD NOT 指令的处理.对大量 PLC 语句表进行分析,总结出程序中有如下几个使用 LD 和 LD NOT 指令的场合.(a) 图符直接与左侧母线连接.(b) 逻辑块的串联连接.其特征为前一指令是 AND LD.(c) 逻辑块的并联连接,其特征为前一指令是 OR LD.如果其前一指令是 AND LD 或 OR LD,则在当前行的下一编辑区绘制相应的梯形图图符;否则,该图符与母线上的下一有效点连接.此外,为便于并联逻辑块的处理,应保存与 LD 或 LD NOT 指令图符对应的编辑区的左上角坐标.(2) OUT, OUT NOT 指令的处理.一般情况下,OUT, OUT NOT 指令用于输出逻辑运算的结果.但如果 OUT 指令的操作数包含“TR”则表示逻辑运算结果为中间结果.此中间结果将被存放到暂存继电器 TR<sub>k</sub> 中,而不输出到 PLC 的外部触点.此时,应记下当前编辑区的坐标( $X_{out}, Y_{out}$ ),继续顺序处理后继指令.当遇到 LD 指令,且其操作数中包含“TR<sub>k</sub>”时,再回笔至坐标( $X_{out}, Y_{out}$ )处并画一条竖连接线,之后再继续后继指令的处理.(3) 并联的处理.程序中存在的并联可能有两种形式.(a) 用 OR, OR NOT 实现.取出其前一个 LD 或 LD NOT 指令的梯形图图符所在编辑区的坐标( $X_{or}, Y_{or}$ ),绘制左连接线.将当前 OR 或 OR NOT 指令所对应的图符在同一列的下一编辑区中绘出.绘制右连接线,回笔至( $X_{or+1}, Y_{or}$ )(1 表示一个编辑单位).(b) 用 OR LD 实现.当遇到第 2 个 LD 或 LD NOT 指令时,取出其前一个 LD 或 LD NOT 指令图符所在编辑区的坐标( $X_{or}, Y_{or}$ ),绘制并联块的左连接线.然后,按规则逐一绘制各梯形图图符,并绘制并联块的右连接线.

3 实验结果

系统设计完成后,我们对其进行测试,图 1 为复杂的串并联关系,图 2 为多个输出的情况.实验结果



(a) 源代码



(b) 自动转换生成的梯形图

图 1 复杂的串并联关系

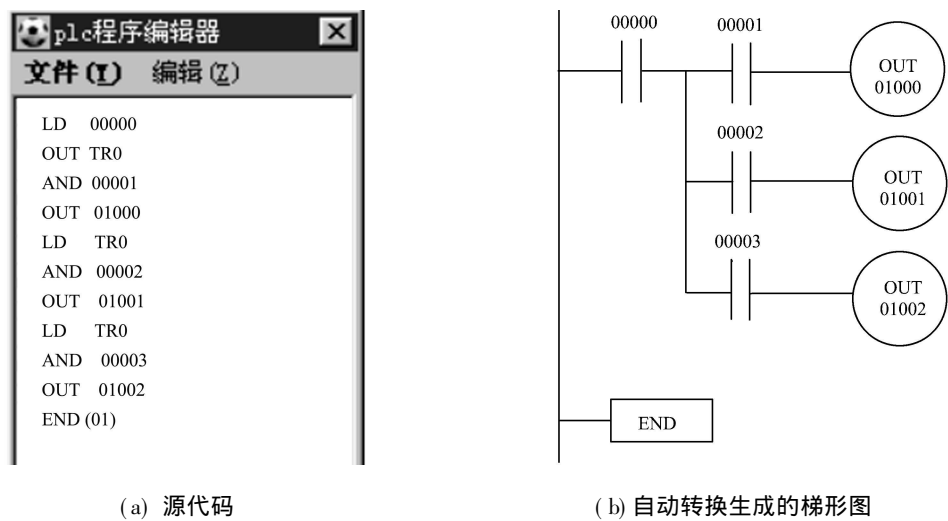


图 2 多个输出的情况

表明, 本系统能正确处理串并联关系及有多个输出等复杂的情况, 自动转换生成的梯形图正确无误.

## 4 结束语

随着工业自动化进程的加快, 可编程控制器的应用领域不断扩大, 可编程控制器应用系统的维护与改造问题越来越受到人们的关注. 本系统可完成应用系统程序向梯形图的自动转换, 并为技术人员对系统作进一步的维护与改造提供了友好的用户界面. 本系统具有形象直观、操作方便、运行速度快等优点.

### 参 考 文 献

- 1 王兆义. 可编程控制器教程[M]. 北京: 机械工业出版社, 1993. 45~ 56
- 2 徐世许. 可编程序控制器原理·应用·网络[M]. 合肥: 中国科学技术大学出版社, 2002. 49~ 119
- 3 严蔚敏, 吴伟民. 数据结构[M]. 北京: 清华大学出版社, 1997. 156~ 170
- 4 张 威, 卢庆龄. Delphi 编程与技巧[M]. 北京: 科学出版社, 1999. 207~ 235
- 5 赵建东, 王广炎, 王小椿. 梯形图编辑系统的设计[J]. 机械与电子, 2000, (4): 49~ 51

## Implementation Method for Automatically Translation of PLC Statement List into Ladder Chart

Lu Junbai

(College of Information Science and Engineering, Huaqiao University, 362021, Quanzhou, China)

**Abstract** For automatically translating statement list of programmable logic controller (PLC) into ladder chart, the author presents here an implementation method; and discusses in detail the principal data structure and its key technology of software implementation. This software can be widely applied to design, remould and maintain the system, It has a fairly good value of practical use and prospect of application.

**Keywords** programmable logic controller , ladder chart, statement list, directed graph