

PLC 梯形图转化为多操作数逻辑指令的设计与实现

邱盛成, 蔡启仲, 李克俭, 潘绍明

(广西科技大学 电气与信息工程学院, 广西 柳州 545006)

摘要: 针对每一个操作数有一个操作码的 PLC 逻辑指令, 设计了一个操作码可带多个操作数, 操作数附加“辅助符号位”的新型 PLC 逻辑指令, 避免在执行逻辑指令过程中有多少个操作数需要译码多少次的状况; 研究这种新指令语句的数据结构和梯形图的数据结构, 应用这两种数据结构研究梯形图转化成新指令表的转化算法; 对梯形图采用按梯级顺序递归扫描方法构造二叉树结构, 然后对二叉树进行后序遍历得到一个中间结果指令表, 对中间结果指令表经过逻辑关系的分析与同类型逻辑关系操作数的合并处理得到最终的指令表; 通过对串并联结构梯形图的转换测试, 其新指令语句的转化结果正确。

关键词: 梯形图; 逻辑指令; 二叉树; 多操作数

Design and Realization of Transformation from PLC Ladder Diagram to Multiple Operands Logic Instruction

Qiu Shengcheng, Cai Qizhong, Li Kejian, Pan Shaoming

(College of Electronic and Information Engineering, Guangxi University of Technology, Liuzhou 545006, China)

Abstract: Each operand in PLC logic instructions contain only an opcode. A new type of PLC logical instruction structure is designed, where its operation code can take multiple operands and its operands additional "auxiliary sign bit". It avoids in the process of execution logic instructions for multiple operands to decode several times. Researched on the data structure of this new instruction statement and the data structure of the ladder diagram, and using these two data structures to achieve the transformation algorithm of the ladder diagram into the new instruction list. Ladder diagram used in cascade order recursive scanning methods constructing binary tree structure, and postorder the tree traversal for an intermediate result instruction list, After the analysis of the logical relationship of intermediate result instruction list and combined process of operation for the same type of logical relations, the final instruction table is obtained. Through the series-parallel structure of ladder diagram transformation test, the new instruction statements transformation result is correct.

Keywords: Ladder; logic instruction; binary tree; multiple operands

0 引言

在可编程控制器 (PLC) 的逻辑指令中, 一般一条逻辑指令由操作码和一个操作数组成, 与母线连接的指令的另一个操作数隐含为 1, 而未与母线连接的指令的另一个操作数来自于上一条逻辑指令执行的结果, 两种情况下的逻辑指令的执行结果都作为下一条指令的操作数^[1]。PLC 执行这类指令序列时, 需要对每条指令进行译码。本文对 PLC 的逻辑指令的操作数结构进行设计, 一个操作码有多个操作数, 每个操作数增加了辅助符号位, 使得逻辑指令序列语句更为简洁, 也避免了在执行 PLC 程序过程中需要对每个操作数的操作码进行译码的处理。

在 PLC 的编程语言中, 指令表是一种字符型描述性语言, 它类似于汇编语言, 与 CPU 可执行的机器码最为接近^[2-3]。梯形图是图像型描述性语言, 形象而直观, 类似于继电器控制原理易于掌握和开发^[1,4]。梯形图翻译成可执行机器码, 则须先将梯形图转化为指令表语句, 然后再转化为机器码。新设计的

指令表语句的梯形图与常用的梯形图的表现形式仍然完全一致, 没有变化, 根据新设计的指令表语句结构, 研究常用 PLC 梯形图与新指令表结构的转化算法^[5-6]。利用二叉树^[7]作为转化过程的中间结构, 经过图形扫描、遍历、逻辑关系分析以及同类型逻辑关系操作数处理等多个步骤完成了整个转化过程。

1 指令表的设计

一般的 PLC 逻辑指令的操作码中还包括有常开、常闭、上升沿脉冲和下降沿脉冲触点的信息, 操作码之后只能带单个操作数, 例如: LDI X01。本文设计逻辑指令的一个操作码有多个操作数, 将常开、常闭、上升沿脉冲和下降沿脉冲触点的信息作为辅助符号位附加在指令的每个操作数后面, 辅助符号位及其功能如表 1 所示。

表 1 辅助符号位的功能关系表

辅助符号	软元件功能
空	常开
I	常闭
P	上升沿触发
F	下降沿触发

设计的操作数的格式为:

操作数 n = 软元件 + 软元件号 + [辅助符号]

按照操作数的通用格式, 新设计的逻辑指令的结构中, 由于每个操作数附加了一个辅助符号位, 而原来的操作码 LD、

收稿日期: 2015-12-16; 修回日期: 2016-01-15。

基金项目: 广西科学基金项目(桂科自 2014GXNSFA118392); 广西教育厅科研项目(2013LX092)。

作者简介: 邱盛成(1989-), 男, 广西北海人, 硕士研究生, 主要从事嵌入式系统与自动化装置方向的研究。

蔡启仲(1956-), 男, 湖南邵阳人, 教授, 硕士研究生导师, 主要从事智能控制方向的研究。

LDI、LDP 和 LDF 只需要一个操作码 LD；同理，对于与运算的 4 个操作码和或运算的 4 个操作码分别用 AND 和 OR 表示；针对和公共母线连接的几个软元件或运算的情况，增加一个操作码 LDR。

设计的逻辑指令格式为：
操作码 操作数 1 [操作数 2] …… [操作数 *n*]
例 1：LDR M10I T20P Y07 X15F
该指令功能表示：与公共母线连接的 M10、T20、Y07、X154 个软元件根据各自的触点信息进行或运算。

例 2：对于一般 PLC 逻辑指令语句
LDI X01
ANP M12
ANF Y03
……
AND M105
应用新设计的逻辑指令为：
LD X01I M02P Y03F …… M105

从例 1 和例 2 可以看出，在一般 PLC 指令语句中，在逻辑关系都为或和与的情况下采用了多行指令语句的形式编写。而在本文设计的指令语句结构中通过添加辅助符号位的形式，使用设计的逻辑指令格式将具有相同逻辑关系与的语句合并为一条语句。其中操作码 LDR 表示带公共母线连接多个连续并联的操作数，LD 表示带公共母线连接多个连续串联的操作数。以此类推，AND 连接多个连续串联的操作数，OR 连接多个连续并联的操作数，其具体关系如表 2 所示。

表 2 基本指令多操作数应用格式表

操作码	操作数
LD	带公共母线与接多个操作数
LDR	带公共母线或接多个操作数
AND	与接多个操作数
OR	或接多个操作数

2 指令的数据结构

由于逻辑指令操作数的数据结构包含软元件，软元件号，和辅助符号 3 个部分，每条逻辑指令可以有一个操作数，或多个操作数，也就是说，每一条逻辑指令的长度是不相同的，设计指令的数据结构必须解决满足这些技术问题。

根据设计的操作数格式，设计的数据结构如下所示。
typedef struct operand//存储指令语句中的操作数
{
char component; //软元件
CString number; //软元件号
CString auxiliary; //辅助符号
}OperandIL;

在设计操作数数据结构中，采用字符或字符串类型进行存储，软元件号虽然是数字类型，但考虑到在界面编辑时，输入的是字符类型，所以采用 CString 类进行存储，这样方便处理，并且可调用 CString 类中的 Format 成员函数将字符串格式化为 10 进制或 16 进制数，方便与机器码的相互转化。

对于一条完整的指令语句，操作数的个数为不确定的值，因此需要采用一种可以动态分配内存空间的数据类型，当操作数个数增加时，可以动态申请内存空间来存放对应的数值，不

浪费内存资源。链表具有插入、删除数据节点方便，且每次增加节点时都向内存申请存储空间的特点^[7]。将操作数数据结构类型 OperandIL 通过链表的形式进行存储，刚好符合本设计要求。由以上思路设计出的单条指令语句数据结构如下所示。

```
typedef struct InstructionList//存储一条指令语句的数据结构  
{  
    CString OpCode; //操作码  
    list<OperandIL> ope; //操作数  
    CString Annotation; //注释  
}IL;  
指令语句存储结构的确定，也决定了梯形图数据结构中的指令结构为 OperandIL 类型。
```

3 梯形图编辑环境

3.1 梯形图编辑界面

在 VS2010 集成开发环境下使用 C++ 高级语言 MFC 框架开发编辑系统。在 MFC 多文档视图中编辑，整个软件系统主要分为三大窗口：梯形图编辑窗口，错误信息提示窗口，文件管理窗口。

在 PLC 程序图中，梯形图由若干阶级构成，自上而下排列，每个阶级起于左母线，经过若干软元件与软元件线圈，止于右母线。因此在梯形图编辑界面的设计中，在梯形图编辑窗口中先确定左右母线的位置，再在左右母线中的区域按均等分为 N 行 M 列的网格来存放元件图片。

3.2 梯形图数据结构

梯形图数据结构的设计主要针对解决两个问题：第一新设计的逻辑指令的梯形图表现形式需要与原来的梯形图完全一致；第二梯形图的数据结构中要隐含含有新指令语句的结构特征，能够为转化算法的实现提供数据特征，完成转化。

基于此考虑将梯形图的数据结构分两层设计，第一层为显示图形层，第二层为存储数据层。显示图形层中用于存放图形元件对应的图片信息、图形元件在编辑界面的坐标，以及上下左右图形元件之间的关系；在存储数据层中，主要存放的是与指令语句结构对应的操作码和操作数结构信息，以及在转化过程中所要保存的中间结果。由以上思路设计出的数据结构如下：

```
class CLadderDiagram  
{  
private:  
    //显示图形层  
    CPoint m_bmpPoint; //图片坐标  
    int m_bmpName; //图片名  
    bool m_behindBranch; //节点后面是否有下划线  
    bool m_frontBranch; //节点前面是否有上划线  
    //存储数据层  
    bool m_Flag; //元件是否被选中  
    bool m_busFlag; //元件是否连接左母线  
    CString m_commandName; //操作码  
    OperandIL m_operandName; //操作数  
    bool m_IsFunc; //是否为功能指令  
    int m_indexFunc; //为功能指令的第几位  
    bool m_mpsFlag; //mps 入栈标识  
    bool m_mppFlag; //mpp 出栈标识  
    ...  
}
```

其中对操作码、操作数数据类型的确定，需要与第 2 节所提到存储指令语句中数据结构一致，以便能够快速的完成转化。

4 梯形图转指令表算法

根据第 2 节、第 3 节中所分别设计的指令语句数据结构和梯形图数据结构。针对两种数据结构研究梯形图转化为指令语句的转化算法。在转化过程中如何将梯形图转化为能够反映其逻辑关系的中间语言，再从中间语言转化到最终的新指令语句结构这两个过程是本设计中所要解决的技术难题。

采用二叉树作为转化的中间语言，因为二叉树中一个父节点可带两个子节点的逻辑结构，刚好符合梯形图实现两软元件间与和或的两个逻辑关系的表现。因此先将梯形图构建出具有逻辑特征的二叉树结构，再根据所得二叉树结构采用遍历、优化、多操作数处理等方法得到最终的指令表结果。如图 1 所示为一个复杂的串并联结构梯形图。按照新设计的指令表结构则该梯形图对应的最终指令表如表 3 所示。

表 3 梯形图的最终指令表

编号	操作码	操作数
0	LDR	X00 T06 X10I X13
1	LD	Y02 M03F M14P
2	LD	X03I C08
3	LDR	Y03 X06I
4	AND	M12F
5	ORB	
6	AND	D12
7	ORB	
8	ANB	
9	AND	X05I
10	OUT	Y01
11	END	

4.1 梯形图转二叉树

梯形图转化为二叉树主要步骤：

1) 初始化一个用于存放二叉树根节点的栈 S，初始化一个空的二叉树 T，从动态二维数组中第一个数组元素开始扫描，按从左往右顺序扫描，若连续扫描到若干元素不含后下划线标识，则按照如下方法构建二叉树 T：新建 AND 节点作为父节点，前两个元素作为子节点，第 3 个元素开始每扫描到一个元素则新建 AND 节点，左子节点为前一元素的根节点，右子节点为当前元素的值。

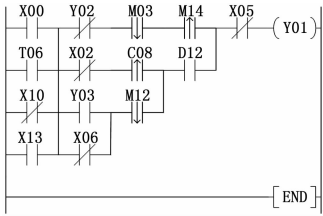


图 1 复杂串并联结构梯形图

2) 在第 1) 步中，若扫描到元素中含有后下划线，并且扫描当前元素对应位置的下一行元素，元素不为空，则将当前元素与前一元素构建好的二叉树 T 根节点压入堆栈 S 中，清空二叉树 T，从当前元素开始按从右往左顺序扫描，直到扫描

到有前上划线标识的元素停止扫描，记下扫描过的元素个数。再从这一元素开始从左往右扫描按照如下方法构建二叉树 T：若扫描过程中所有元素不含后下划线则新建 OR 节点作为父节点，将堆栈中元素弹出作为左子节点，再将扫描过的元素按 (1) 中构造二叉树的方法所得的根节点作为右子节点。返回上一行元素继续扫描。

3) 在第 2) 步中，若扫描当前元素对应位置的下一行元素，该元素为空，则说明该下划线与右边并联块相连，此时新建 AND 节点为父节点，左子节点为前二叉树 T 根节点，右子节点为空，将此 AND 节点压入堆栈 S 中，清空二叉树 T，返回上一行按 (1) 中的方法继续扫描。

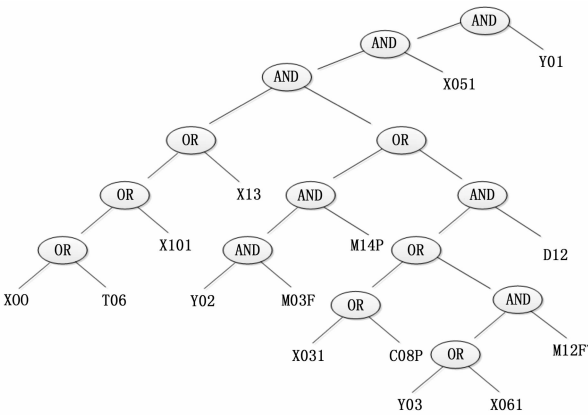


图 2 梯形图二叉树结构

4) 构造二叉树时若仍然遇到元素中含后下划线标识，则递归调用 2) 的方法构造二叉树。直到扫描完整行元素。

5) 整行元素扫描完毕后，该层梯形图的逻辑结构已用二叉树存储起来，后面存在的所有梯形图结构都可用相同的方式用二叉树存储起来。然后创建一个二叉树队列 L，将所有构造好的二叉树存储到队列 L 中，完成整个梯形图程序的存储过程。

如此便将梯形图完整的转化为需要的二叉树结构，图 1 的一个完整的二叉树结构如图 2 所示，从图 2 可以看出，生成的二叉树结构能够完整的将软元件的触点信息反映出来。为转化为指令表后如何判断软元件触点的信息提供了途径。

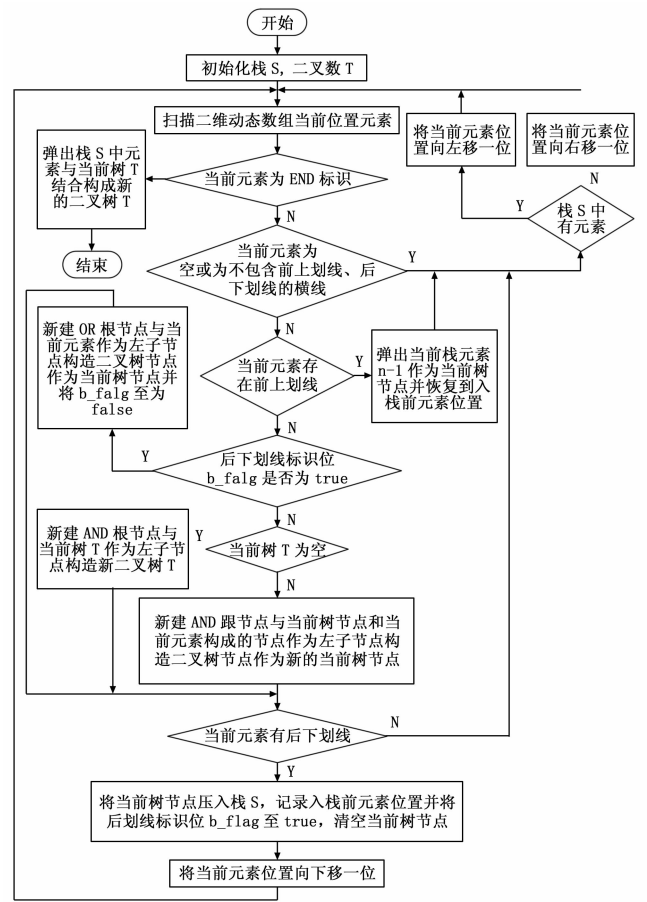
梯形图转化为二叉树的详细程序流程图如图 3 所示，在转化之前，先构造一个二维动态数组将用户编写的梯形图程序按对应的元素位置存储到数组中，然后按从左往右的顺序对数组扫描。在扫描过程中根据元素数据结构的特征按图 3 中的条件进行逐一判断，满足条件则进行对应的操作处理。

其中每次入栈操作表示扫描到的元素节点为并联模块开始节点，需要将之前扫描构造的二叉树压入栈中，再对并联模块构造新的二叉树；而出栈操作则表示扫描到的元素节点为并联模块结束节点，需要弹出栈中二叉树节点与并联模块的二叉树节点合并为新的二叉树。

4.2 二叉树转新指令表

在二叉树构造过程中，已将构造好的所有二叉树存放到队列 L 中，然而二叉树中所反映的仅是梯形图串并联结构关系，却不能得到所需指令表。因此需要对队列 L 中的二叉树进行相应的处理使其能转为对应的指令表。

二叉树转化为指令表主要由三步骤完成：



1) 对队列 L 中的二叉树按先进先出的方式取出, 对其依次进行后续遍历得到指令表如表 4 所示。表 4 的指令表并不是最终指令表仍需做进一步处理。

表 4 对二叉树后续遍历所得指令表

编号	操作码	操作数	编号	操作码	操作数
0	LD	X00	15	LD	Y03
1	LD	T06	16	LD	X061
2	LD	X10I	17	OR	
3	LD	X13	18	LD	M12F
4	OR		19	AND	
5	OR		20	OR	
6	OR		21	LD	D12
7	LD	Y02	22	AND	
8	LD	M03F	23	OR	
9	AND		24	AND	
10	LD	M14P	25	LD	X051
11	AND		26	AND	
12	LD	X031	27	OUT	Y01
13	LD	C08P	28	AND	
14	AND		29	END	

2) 按从上往下的顺序扫描所得指令表, 当扫描到 LD 语句紧跟后面为 OR/AND 指令时, 将 LD 指令中的操作码改为相应的 OR/AND, 删除紧跟其后的 OR/AND 指令。当扫描到连续出现的 LD 指令时, 则记录下连续出现的 LD 指令个数,

并将其先后压入堆栈中, 直到扫描到 OR/AND 指令后再从堆栈中依次弹出 LD 指令, 将 LD 指令中的操作码改为扫描到的 OR/AND 指令。若扫描到连续两个 OR/AND 无操作数指令, 则对第二个指令进行处理若原来是 OR/AND 则改为相应的 ORB/ANB。

3) 按照第 2) 步处理之后所得指令表如表 5 所示, 其中结果与常用 PLC 指令表类似, 但仍有不同的地方, 在其中添加了本文所设计的操作数辅助符号位。但所得指令表所得结果仍不是最终指令表结果。需要再一次重新按从上往下的顺序扫描表 5, 若扫描到 LD 后面连续出现的 OR/AND 指令, 则将扫描到的指令合并成一条操作码为 LDR/LD 的多操作数指令; 若扫描到在 ORB/ANB 之后连续出现 OR/AND 指令, 则将扫描到的指令合并成一条操作码为 OR/AND 的多操作数指令。如此得到最终指令表结果如表 3 所示。

表 5 对图 4 扫描处理后的结果

编号	操作码	操作数	编号	操作码	操作数
0	LD	X00	10	LD	X061
1	OR	X06	11	AND	M12F
2	OR	X10I	12	ORB	
3	OR	X13	13	AND	D12
4	LD	Y02	14	ORB	
5	AND	M03F	15	ANB	
6	AND	M14P	16	AND	X051
7	LD	X031	17	OUT	Y01
8	AND	C08P	18	END	
9	LD	Y03	19		

5 实验结果与分析

根据设计要求, 在 VS2010 集成开发环境下使用 MFC 框架完成软件的编写。梯形图在设计的 PLC 编程软件上的运行结果如图 4 所示, 梯形图转化为指令表的转化结果如图 5 所示。

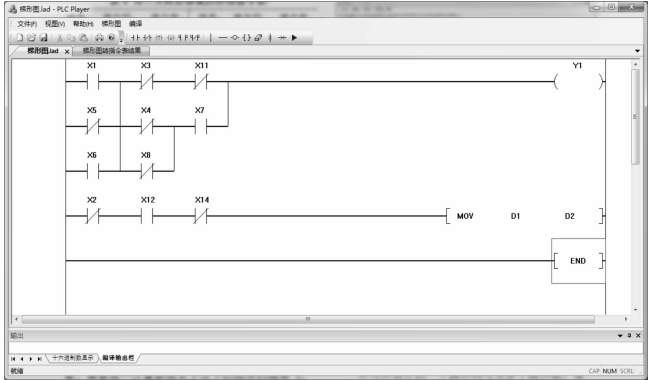


图 4 编辑软件梯形图

由实验结果可以看出, 通过一个具体的转化实例来验证设计算法的准确性和可实现性, 由多路输出的复杂串并联结构的梯形图, 经过编译转化成了具有多操作逻辑指令的指令表, 同时对梯形图功能指令也能完成转化, 基本实现了本设计的功能, 满足设计要求。

6 结束语

针对所设计的一个操作码可带多个操作数, 每个操作数附加一个辅助符号位的 PLC 逻辑指令的结构, 提出了新逻辑指 (下转第 217 页)

值导致图片细节丢失，从图 4 四幅图像的实验结果，可以明显得到本文算法（3IN1-Cuts）的分割效果比文献 [3] 和文献 [5] 的分割结果要好。

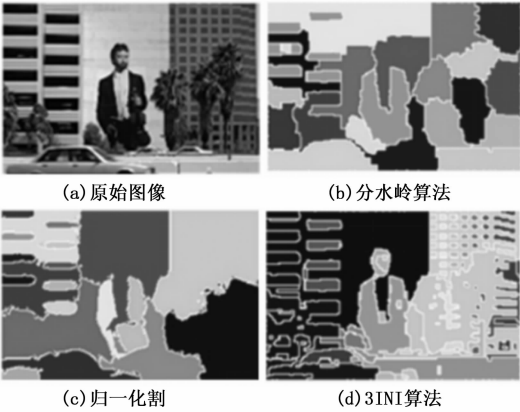


图 4 本文算法与文献算法实验比较

此外，我们还对相应的分割时间进行了对比，以编号为 119082 的图像为例，具体如表 2 所示。

显然，除了无实用价值的传统分水岭算法外，3IN1-Cuts

表 2 五算法耗时统计表

算法类型	图像编号	图像规格	时间/s
传统分水岭	119082	481 * 321	1.29
归一化割法	119082	481 * 321	6.45
文献[3]算法	119082	481 * 321	5.26
文献[5]算法	119082	481 * 321	4.25
本文算法	119082	481 * 321	4.13

算法在分割相同的图像时消耗时间是最小的（图像效果也是最好的，见图 3 与图 4），分别比文献 [3] 和文献 [5] 的改进算法低 1.13 和 0.12 秒。从实验结果可推断，利用改进后的中值过滤算法适度除噪，利用分水岭变换的优点进行初步分割及归一化割的稳定性，能较好地整合了三者的优点，不仅取得较

好的分割效果，还在时间复杂度方面得到优化。

5 结论

本文针对传统分水岭算法过分割现象，提出一种基于分水岭、中值过滤和归一化割算法的综合优化算法，该算法综合 3 种算法的优势，较好地解决分水岭过分割问题，提高了分水岭算法的分割精度，并且降低了归一化割算法的时间复杂度。文章最后通过实验对比证明本文算法与同类改进算法相比在分割效果和时间消耗方面取得了较好的效果。

参考文献：

[1] Nguyen H T, Worrning M, Boomgaard R V D. Watersnakes: energy-driven watershed segmentation [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2003, 25 (3): 330-342.

[2] Wang D. A multiscale gradient algorithm for image segmentation using watersheds [J]. Pattern Recognition, 1997, 30 (12): 2043-2052.

[3] 崔明, 孙守迁, 潘彦鹤. 基于改进快速分水岭变换的图像区域融合 [J]. 计算机辅助设计与图形学学报, 2005, 17 (03): 546-552.

[4] Haris K, Efstratiadis S N. Hybrid image segmentation using watersheds and fast region merging [J]. IEEE Transactions on Image Processing, 1998, 7 (12): 1684-1698

[5] 杨卫莉, 郭雷. 基于分水岭算法和图论的图像分割 [J]. 计算机工程与应用, 2007, 43 (7): 28-30.

[6] 冯林, 孙焘, 吴振宇, 等. 基于分水岭变换和图论的图像分割方法 [J]. 仪器仪表学报, 2008, 29 (3): 649-653.

[7] 陈忠, 赵忠明. 基于分水岭变换的多尺度遥感图像分割算法 [J]. 计算机工程, 2006, 12 (23): 186-187.

[8] 李海滨, 杜益福, 刘彬. 基于图割和分水岭变换的图像分割方法 [J]. 燕山大学学报, 2013, 37 (2): 143-147.

[9] 黄思博. 基于计算机视觉的异常驾驶行为检测方法研究 [D]. 广州: 华南理工大学, 2011.

[10] 高欣. 基于彩色超声图像的血管内膜中层厚度测量算法的研究 [D]. 哈尔滨: 哈尔滨工业大学, 2010.

[11] Vincent L, Pierre S. Watersheds in digital spaces: An efficient algorithm based on immersion simulations [J]. IEEE Trans. on Pattern Analysis and Machine Intelligence, 1991, 13 (6): 583-598.

(上接第 205 页)

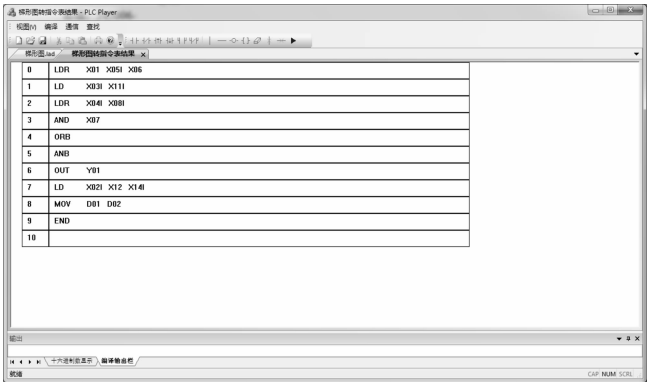


图 5 编辑软件指令表结果

令数据结构和梯形图数据结构，并以二叉树作为中间语言结构，梯形图采用按梯级顺序递归扫描算法扫描得到二叉树，再对二叉树经过后序遍历、逻辑关系分析处理以及同类型逻辑关系操作数合并处理 3 个步骤，完成了从梯形图转化为新指令表。通

过测试，该转化算法能够将用户设计的梯形图正确转化为新逻辑指令结构，为其他控制软件的设计提供了一定的参考价值。

参考文献：

[1] 张振国, 方承远. 工厂电气与 PLC 控制技术 [M]. 北京: 机械工业出版社, 2011.

[2] 邹光华, 刘毅. 基于 XML 的 PLC 数据描述 [J]. 计算机工程, 2006, 01: 113-115.

[3] 蒋宗礼, 姜守旭. 编译原理 [M]. 北京: 高等教育出版社, 2010.

[4] 蒲志新, 熊永超, 熊晓红. PLC 梯形图语言编辑功能的软件实现. 机械与电子, 2003, 3: 54-55.

[5] 苏淑芝. 软 PLC 梯形图编程系统的研究与实现 [D]. 广州: 华南理工大学, 2012.

[6] Bi H, Cheng L D. Research about the conversion of ladder diagram into sentence list in SoftPLC [J]. Control & Automation, 2007, 9 (1): 63-65.

[7] 严蔚敏, 吴伟民. 数据结构 (第 2 版) [M]. 北京: 清华大学出版社. 2010.