西门子6ES7511-1AL03-0AB0|操作指南

产品名称	西门子6ES7511-1AL03-0AB0|操作指南
公司名称	浔之漫智控技术（上海）有限公司
价格	.00/件
规格参数	品牌:西门子 型号:模块 产地:德国
公司地址	上海市松江区广富林路4855弄88号3楼
联系电话	158****1992 158****1992

产品详情

梯形图是PLC控制系统中使用得最多的图形编程语言，被称为PLC的第一编程语言。梯形图与电器控制系统的电路图很相似，具有直观易懂的优点，很容易被工厂电气人员掌握，特别适用于开关量逻辑控制。梯形图常被称为电路或程序，梯形图的设计称为编程。PLC梯形图设计规则（或规范）如下：(1)触点应画在水平线上，不能画在垂直分支上。应根据自左至右、自上而下的原则和对输出线圈的几种可能控制路径来画。(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，应将触头多的那个串联回路放在梯形图的最上面。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。这种安排，所编制的程序简洁明了，语句较少。(4)不能将触点画在线圈的右边，只能在触点的右边接线圈。

摘要：针对目前PIC梯形图编辑软件中梯形图存储结构的复杂和不足，本文基于面向对象的方法，采用二叉树和双向链表相结合的数据结构来描述梯形图功能单元及其拓扑关系，并提出了相应的转换算法，然后简化为只含有功能单元的模型.使后续的指令表转换得到简化。这种模型结构简单、通用性强、易于用C++语言实现，在PLC梯形图编辑软件中应用效果良好。

关键词：PLC;梯形图;二叉树;双向链表;指令表

1 引言

PLC即可编程逻辑控制器，是用来取代用于电机控制的顺序继电器电路的一种器件。梯形图语言是PLC程序设计中最常用的编程语言，它是与继电器线路类似的一种编程语言，梯形图用不同的图符表示不同的指令，用串、并联等概念组织图符的顺序位置表述控制逻辑。梯形图形象直观，与电气控制原理图相呼应。采用梯形图语言设计顺序控制逻辑，具有方便直观的优点，将控制系统的开关趋逻辑与状态表示成梯形图，有利于系统维护与快速故障诊断。由于电气设计人员对继电器控制较为熟悉，因此梯形图编程语言得到了广泛的应用。但是，梯形图不能由计算机直接执行，需要将它转换成计算机能够识别的命令才能够执行。在这个转换的过程中，本文提出了二叉树双向链表的数据结构来表尔梯形图功能元件及其拓扑关系，使后续的指令表序列的生成得到简化。

2 梯形图及数据结构

2.1梯形图基本介绍

在梯形图的图形编辑界面中，用不同的图符表示不同的指令。通常，梯形图的组成中有功能单元、连接单元和空单元。功能单元，如常开指令(-I I-)、脉冲指令(..个卜)、输出指令(Q)等;连接单元为并联连接(下分支(丁)、右分支(卜)、左分支(一)、左转(J)、右转(L))、串联连接(一)和纵向连接(I)。典型的梯形图如图l所示。采用动态增加梯形图的行和列的方法，初始的显示图形的区域为2×n，程序，F始标志(start)和结束标志(End)各占1行，n是一个不超过编辑界面宽度的合适的初始值。梯形图的编辑也有相应的规则和限制，添加这蝗限制和规则是为了简化后续的数据结构和算法设计。

梯形图编辑遵循的规则如下：(1)所有的功能单元都必须画在水平线上，不能l画在乖A分支f：，按照由左向右、由上到下的绘图原则;(2)由几个并联回路组成的串联同路中，包含功能单元最多的并联网路放在最左边，在由几个串联回路组成的并联l川路中。包含功能单元最多的串联回路放在最上边;(3)不能将功能单元画在输fl{指令的右边，即输出指令只能放在一行的最右边。一个简单的梯形图如图1所示。

2.2二叉树及二叉树双向链表

树是一种数据结构，数据元素之间有明显的层次关系。树(Tree)是n(n≥o)个结点的有限集。在任意一棵非宅树巾：(1)有且仅有一个特定的称为根(Root)的结点;(2)当，n>l时，其余结点可分为m(m>o)个互小相交的有限集T1，T2，?，L，j£巾每一个集合本身又是一棵树，并且称为根的子树(Subtree)。

二叉树是一种树型结构，它的特点是每个结点至多只有二棵子树(即二叉树中不存在度大于2的结点)，并且二义树的子树有左右之分，其次序不能任意颠倒。

双向链表可以克服单链表单阿陛的缺点，在舣向链表的结点中有两个指针域，其一指向汽接后继，另一指向直接前趋。二叉树双向链表中以每棵：叉树作为一个链结，将一个二义树森林以一定的顺序连接起来，其中的每个链表结点需要保存对应的二叉树根结点的地址信息。

3转换算法的基本思想

3.1梯形图向二叉树的转换算法

依据二叉树的定义，结合PLC梯形图的特点：以图符表示操作指令，用图符的位置表示串并联的逻辑关系。由于我们采用的梯形同编辑环境是用每一个固定大小的单元格表示一个图符，因而每一个蹦符抽象为二叉树中的每一个结点。

具体的转换思想描述如下：对梯形图巾程序进行从左向右、从上到下的扫描，扫描过程中，识别每个图符所代表的单元类型(功能单元或者连接单兀)，空单元不需处理，用每个起点表示二义树的根结点(Root)，以左子树表示串联连接，右子树表示并联连接。

3.2二叉树转换成二叉树双向链表

梯形图中图元的执行足有同定执行顺序的。通常，一棵二叉树能够表示一个子过程，一个大型的控制系统南多个子过程按一定的先后顺序组织丽成。在梯形图向二叉树转化后得到的足一个二义树森林，它是一个松散的结构，并不能体现一个系统完整的功能，必须采用一种数据结构将这些二叉树按照一定的次序组织起来，这里采用二叉树双向链表。

二叉树双向链表按照程序的执行顺序将一棵棵二叉树连接起来，每个链表结点代表一棵二叉树。通常，我们的链表结点中存放了每个二叉树根结点的信息，这样通过对链表中的二义树按照顺序进行简化和一

次遍历就可以实现梯形图向指令表序列的转化。

3.3二叉树的简化处理过程

由梯形图得到的二叉树双向链表含有大最的连接结点的信息，在由二叉树双向链表向语句表转化的时候，需l要过滤掉这些结点而形成只含有功能单元的.：义树双向链表，并儿能完整地描述梯形图的逻辑功能信息。采用先序递归遍历舣向链表中：叉树结点的方法来完成功能一：义树链表的牛成，在遍历每一棵二义树中图元对象结点的时候，需要进行一系列判断和处理，由此，我们需要设计一个简化算法。具体的简化算法实现见4.2节。

4转换算法的实现

4.1主要的数据结构

4.1.1基本图元数据结构

在整个算法的没汁过程中，采用了面向对象的设计思想，首先将梯形图中的每一个图符抽象为一个图兀对象，对于这些图兀定义了一个基本图元类：

class bascElcment

{public：

int type;//图符单元的类型值

char name[20];//嘲符单，i的变譬名

char dec[20];//图符单元的说明

int row;//图符单元所在的行u|

int f01umn;//I刽符单元所在的列号

bascElcmcnt*lPft;///I：指针

bascElcmcnt*r|ght;//右指针

baseElcment ’parent;//父指针

};

在梯形图设计中涉及到的基本指令单元、计时指令单元、计数指令单元、读写指令单元、操作指令单了亡、比较指令单元、转换指令单元等都由慕奉罔元类baseElement派生出来。

4.1.2二叉树的数据结构

在梯形图中，用每一个图符来表示二叉树巾的结点，以每个起始图元对象作为单棵二叉树的根(Root)。以左子树表爪串联连接，右子树表示并联连接。定义二叉树链结类bTrecbbbb如下：

Class bTrccbbbb

{publc：

base卜1lement root;//根节点图符

bTreebbbb*next;//指向下一二叉树

bTrccbbbb。prior;//指向lii『一二叉树

public：

Insc九l，ef “);//左子树插入

InsertRight();//右了.树插入

DeleteEl咖ent();//删除结点

}

4.1.3二叉树双向链袁的数据结构

梯形图程序的完整信息采用二叉树双向链表来存储，二叉树舣向链表类的数据结构抽象如下：

class treeI.ist

{public：

bTrccbbbb*head;//双向链表头指针

bTreebbbb*currcnt;//当前结点指针

bTrcebbbb*tail;//双向链表尾指针

public：

InertbTrcc(bTreeI。ink*node，bTreebbbb*current);//插入_二义树链结

DIeletebTree(bTreeL.nk*node);//删除二叉树链结}

4.2二叉树的简化算法

二叉树中含有大量的冗余信息，在其向指令表转化的过程中需要对je进行简化处理，采用对每棵二叉树进行一次先序遍历，对每一个图元结点对象进行判断处理。

二叉树的简化主要是过滤掉梯形网中多余的连接图元，这里把主要对九种不同的图元对象做简化处理：(1)功能单元对象;(2)虚结点图元对象;(3)连接单兀包含七种：下分支、右分支、左分支、左转、右转、串联连接、和纵向连接。

对于不同的图元类型进行不同的处理。

这里，简化函数中列出了三种类型的图元的简化处理算法，其他类型处理类似。

```
Predigcst(bTrccbbbb*p，int type)

{swiIch(type)

{case o：//图元对象为功能单元

Break;//功能单元保留

casc l：//图形对象为下分支

p_，pareIlt，left=p—lcft;//去除连接结点

if(p—right!一NUI.I。)//若右了.树存在

p一1eft—right—p—right;//连接右子树

brcak;//下分支连接单元简化处理

case 2：//图元对象为右分支

p—parenl_-right一旷lcft;//去除连接结点

p一1eft—right—p—right;//连接右f树

brcak;//右分支连接单元简化处理

case 8：{//纵向连接单元简化处理)

))
```

5转换实例

图1所示的是一个具有复杂串并联关系的梯形图程序，其中包含的两个逻辑关系式如下所示：

图2为该梯形图程序中的两个逻辑关系式对应的两棵二叉树，包含r梯形图中描述的所有信息，其中扫描中重复的结点我们定义为虚结点。

上面得到的两棵二叉树是一个松散的结构，我们采用了二叉树双向链表将其链接起来，使之完整地描述梯形图的信息。图3给出了一个含有N棵二叉树结点的模型描述。

bTree o～bTree挖一1为梯形图中所包含的二叉树，一般来说，双向链表结点中只需要保存二义树根结点的地址即可。prior和next为双向链表的前驱指针和后驱指针，其中prior指向前一棵二义树的根结点，next指向下一棵二叉树的根结点，head指针指向双向链表的第一个结点，current为当前指针，指向当前结点，tail指针为尾指针，始终指向链表的最后一个结点。

在向指令表转换之前，我们对每一棵=义树结点进行了简化处理，采用4.2节描述的简化算法，得到如下

的精简结构，如图4所示。

对上面得到的简化二叉树，我们只需要经过一次后遍历和一些判断处理，就町以得到相应的指令表序列。

6结束语

本文介绍的这种二叉树双向链表的数据结构简单、清晰、算法易于实现，与项日具体相结合，采用r面向对象的方法并用C++语言来实现，实现了数据和方法的良好封装。同时，由于这种简捷的结构，使后续的由梯形图存储结构到语句表的转换算法的设计变得简单，只需要对二叉树双向链表遍历一次便叮以得到语句表序列。

参考文献：

[1]谭锦洁，程良鸿。殷学鹏.嵌入式PLc中梯形图到Aov图的映射[J].计算机测域与控制，2004，12(10)：993—995.

[2] 严蔚敏，吴伟民.数据结构[M].北京：清华大学出版社，1997：118—158.

[3] 吕俊白.PLc语句表向梯形图自动转换的实现方法[J].华侨大学学报，2005，26(3)：165一167.

[4] 吕俊白，施敏芳.PLc梯形图町视化编辑与语句表的自动生成[J].自动化仪表，2005，26(3)：28—30.

[5]葛芬，吴宁.基于AOV图及二叉树的梯形图与指令表互换算法[J].南京航空航天大学学报，2006，38(6)：754—758.