

堆栈在 PLC 程序设计中的应用

本文介绍了PLC程序设计中堆栈的基本概念 堆栈的工作原理 与堆栈有关的指令及其使用方法 , 以及不同的PLC的堆栈操作的差异。

问: 什么是堆栈?

答: 堆栈由PLC中顺序相连的若干个位存储单元组成, 采用先进后出的数据存取方式。三菱公司的FX系列PLC的堆栈有11层(见图1), 堆栈中的每一层用于存放一个二进制数, 相当于一个辅助继电器的映象寄存器。

问: 堆栈有什么作用?

堆栈主要用于存放用户程序执行过程中的中间数据:

(1)处理电路块的串联和电路块的并联时, 用来储存一个或多个电路块内部的运算结果。

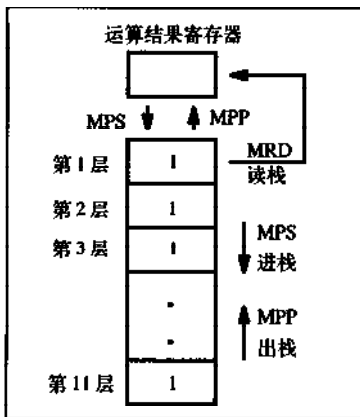


图1 FX系列PLC的堆栈

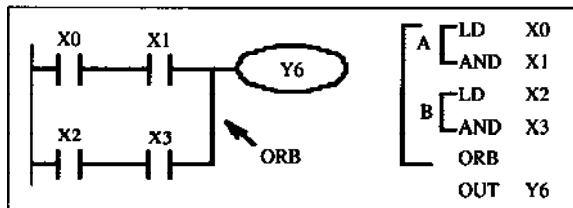
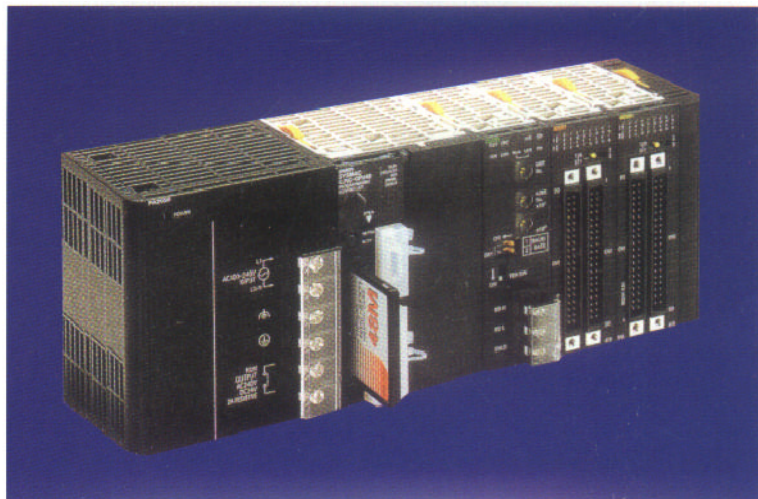


图2 ORB指令



(2)处理由触点与线圈(或触点与其他输出类指令)组成的多分支电路时, 用堆栈来保存分支点的运算结果。

问: 处理电路块的串联或电路块的并联时, 为什么要使用堆栈?

以图2中两个串联电路块的并联为例, 为了将它们并联起来, 首先需要完成两个串联电路块内部的“与”逻辑运算。执行第一条指令时, 取出输入映象寄存器X0中的二进制数, 存放在图1的运算结果寄存器中。执行第二条指令时, 取出X1的值并与运算结果寄存器中的数相“与”, 运算结果存入运算结果寄存器。执行第3条指令时应取出X2的值并放入运算结果寄存器, 但是这一操作将会破坏前两条指令的运算结果。为了解决这一问题, 系统程序自动地将前两条指令的运算结

果存入堆栈, 即将堆栈中的数据依次向下移动一格(最低位的数据丢失), 然后将运算结果寄存器中的数据写入堆栈的第一层(称为栈顶), 这样前两条指令的运算结果便被保存在堆栈的栈顶。

完成上述操作后, 再将X2的值传送到运算结果寄存器。执行完第4条指令后, 两块串联电路的“与”运算结果分别存放在堆栈的栈顶和运算结果寄存器中。执行电路块并联指令ORB时, 将栈顶和运算结果寄存器中的二进制数相“或”, 运算结果存入运算结果寄存器, 堆栈中的数据依次上移一格。经过一上一下的移位, 堆栈中原有的数据复原。

问: 将电路块的串联或并联电路改写为指令表时应注意什么问题?

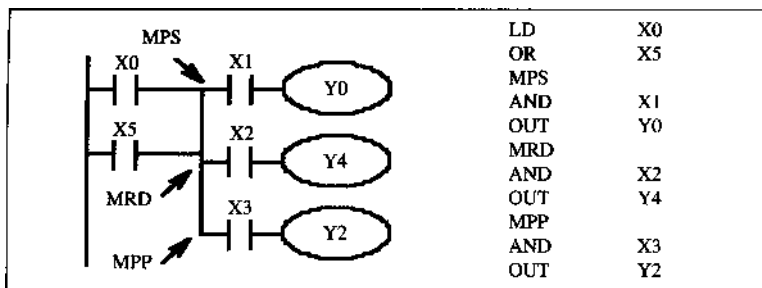


图3 多分支输出电路

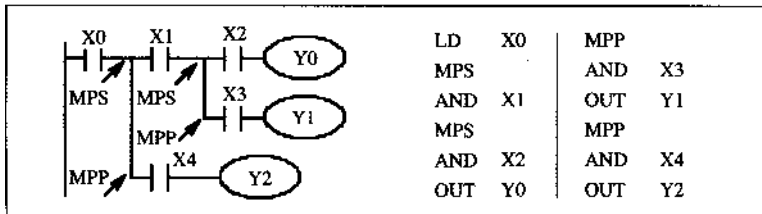


图4 二层栈

电路的起始触点应使用LD(常开触点)或LDI(常闭触点)指令。设一块独立电路中需串联或并联的电路块的个数(或LD和LDI指令的条数)为n,应使用n-1条电路块串联指令(ANB)和电路块并联指令(ORB)将它们串联或并联。在一块独立电路中同时保存在堆栈中的数据个数不能大于堆栈的层数。

在指令表中,ORB或ANB连接它上面靠近它的两块已经连接好的电路。

单个常开触点并联时应使用OR指令,有的初学者对单个触点的并联使用LD和ORB这两条指令,虽然不算错误,但是要多用一条指令。单个触点的串联也应按同样的原则处理。

问:怎样用堆栈操作指令来处理含有输出类指令的分支电路?

MPS(Push),MRD(Read),MPP(Pop)指令分别是进栈、读栈和出栈指令,它们用于多重输出电路(见图3)。

MPS指令用于储存多重输出电路中分支处的逻辑运算结果,以便以后处理有线圈或输出指令的支

路时调用该分支处的运算结果。以图3为例,三个线圈对应的输出指令都要使用X0和X5的并联电路的“或”运算结果。如果不用堆栈来保存该数据,在执行与第一个线圈串联的X1的触点对应的AND X1后,前两条指令的运算结果将会丢失。所以在第二条指令之后,应使用进栈指令(MPS)将“或”运算的结果保存在堆栈的栈顶。堆栈中原来的数据依次向下一层推移。

因为存放在栈顶的数据要多次使用,在执行完第一个线圈对应的OUT指令之后,应使用读栈指令MRD,读取存储在栈顶的电路中分支点处的运算结果,将X2的触点强制性地连接在该点,执行读栈指令后堆栈内的数据不会上移或下移。

在处理最后一条输出分支电路时,应使用出栈指令MPP。该指令弹出(调用并去掉)存储在栈顶的电路中分支点处的运算结果。首先将X3的触点连接在该点,然后从堆栈中去掉该点的运算结果。使用MPP指令时,堆栈中各层的数据依次向上移动一层。栈顶的数据在送入运

算结果寄存器后从栈内消失。

综上所述,对于图3这样的单层分支电路,处理完分支点左边的电路后,应使用进栈指令,保存分支点的运算结果。在处理中间的分支电路时,应使用读栈指令,读取存放在栈顶中的数据。在处理最后一条分支电路时,一定要使用出栈指令,使执行进栈指令之前堆栈中原有的数据复原。

问:什么是堆栈的先进后出的数据存取方式?

答:因为堆栈中的数据是从栈顶“压入”和从栈顶“弹出”的,进栈时栈内原有的数据依次向下移动一层,出栈时栈内的数据依次向上移动一层,所以先压入的数据后弹出堆栈。

这种数据存取方式有些类似于枪支的弹夹中子弹的进出方式。弹夹底部的弹簧将子弹往上顶。从弹夹的入口(顶部)压入一颗子弹,原来的子弹依次向下移动。取出(或射出)弹夹中最上面的子弹后,下面的子弹依次上移。显然,子弹的进出方式也是先进后出的。

问:堆栈为什么要采用先进后出的数据存取方式?

在图4的两级分支电路中,第一条进栈指令MPS将分支点A点处的运算结果保存在堆栈的栈顶,第二条MPS指令将原来在栈顶的A点处的运算结果推移到堆栈的第2层,然后将B点处的运算结果保存在栈顶。执行第一条出栈指令MPP

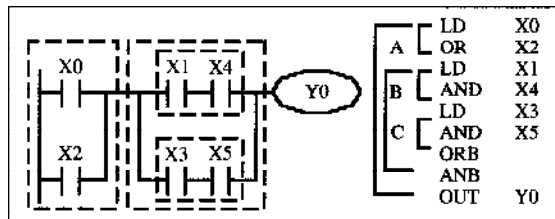


图5 电路块的串并联

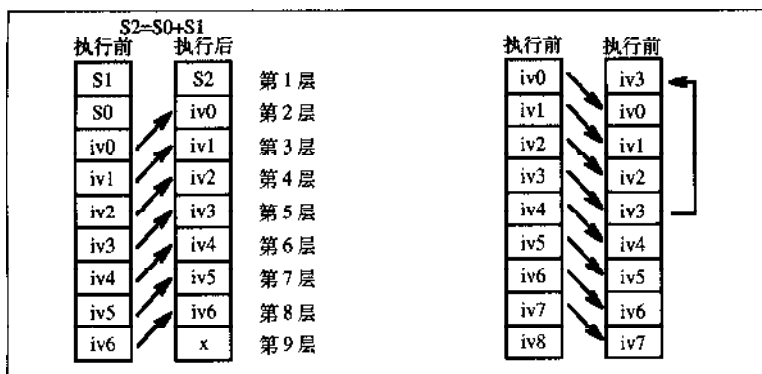


图6 OLD指令的堆栈操作

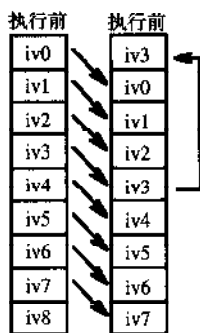


图7 堆栈装载指令

时,将栈顶保存的B点处的运算结果传送到运算结果寄存器,供Y1的线圈对应的输出电路的逻辑运算使用,该输出电路被连接到第二个分支点B点处。同时堆栈中各层的数据依次向上移动一层,原来在第二层的A点的运算结果被移到栈顶。在执行第二条出栈指令MPP时,将栈顶中A点的运算结果传送到运算结果寄存器,供Y2的线圈对应的输出电路的逻辑运算使用,该输出电路被连接到第一个分支点A点处。

在图5所示的电路中,有3块电路需要串并联。执行完前6条指令后,这3块电路内部的运算结果A、B和C分别被存储在堆栈的第2层、第1层和运算结果寄存器中。执行电路块并联指令ORB后,“或”运算B+C的运算结果D存入运算结果寄存器,堆栈中的数据依次上移一层,原来在第2层的数据A移到第一层。执行电路块串联指令ANB后,“与”运算D+A的运算结果存入运算结果寄存器,堆栈中的数据依次上移一层后恢复原状。

从上面两个例子可以看出,堆栈的“先进后出”的数据存取方式刚好可以满足多重分支电路和电路块串并联处理中数据的保存和使用的顺序,因此堆栈在PLC的逻辑运

算中得到了广泛的应用。

问:使用与堆栈有关的指令时应注意什么问题?

答:每一条MPS指令应对应一条MPP指令,在一块独立电路中,MPS指令和MPP指令的条数应相等。处理一个分支点的最后一条支路时,必须使用MPP指令,而不是MRD指令。在一块独立电路中,用进栈指令同时保存在堆栈中的运算结果不能超过11个。

问:在编程软件中用梯形图语言编程时,是否需要使用堆栈指令?

答:在编程软件中用梯形图语言编程时,堆栈的处理是编程软件和PLC自动完成的,用户只需要根据自己的要求画出梯形图就行了。在用编程软件将梯形图转换成指令表时,编程软件会根据电路的结构自动地在程序中加入MPS、MRD和MPP指令。

在用指令表语言编程时,程序设计员需要用堆栈指令来实现输出电路分支点处的数据保存、使用和堆栈的复原,必须由程序设计人员来写入MPS、MRD和MPP指令。

问:西门子的S7—200系列PLC的堆栈指令有什么特点?

答:S7—200的堆栈结构、堆栈操作和堆栈指令与三菱的FX系

列PLC的基本上相同,二者的区别如下:

S7—200有1个9位的堆栈,栈顶用来存储逻辑运算的结果,相当于FX系列的运算结果寄存器。第2~9层用来存储中间运算结果。堆栈中的数据一般按“先进后出”的原则存取。

以执行电路块并联指令OLD的堆栈操作为例,两个电路块的运算结果S0和S1分别存放在堆栈的最上面两层(见图6),执行OLD指令时,将两个二进制数S0和S1相“或”,运算结果S2存放在栈顶,因为空出了一位,第3~9层中的数据依次向上移动一格(见图5)。

入栈(LPS)指令复制栈顶的值并将其压入堆栈的第2层,栈中原来的数据依次向下一层推移,栈底值被推出丢失。

读栈(LRD)指令将堆栈中第2层的数据复制到栈顶。第2~9层的数据不变,但原栈顶值消失。

出栈(LPP)指令使栈中各层的数据向上移动一层,第2层的数据成为堆栈新的栈顶值,原来栈顶的数据从栈内消失。

装载堆栈(LDS n)指令复制堆栈内第n层(n=1~8)的值到栈顶(见图6)。栈中原来的数据依次向下一层推移,栈底值被推出丢失,这条执行使用的可能性极少。

问:用户是否能监视堆栈中的数据?

答:在S7—200的编程软件中,使用菜单命令“工具→选项”选择“STL”标签后,可选择在使用语句表状态监视功能时,是否监视逻辑堆栈,和监视逻辑堆栈的位数,最多可监视4位,即监视堆栈上面的4层。**ET**