**CFC(FBD)语言**

IEC61131-3中没有定义CFC(Continuous Function Chart)语言，因此，不属于IEC标准语言，但是定义了FBD(Function Block Diagram)语言，并且和FBD语言很相似。

# M6实现内容

## 元件类型

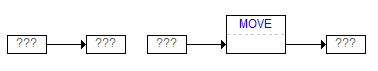
|  |  |  |
| --- | --- | --- |
| **元件名称** | **元件** | **说明** |
| 输入元件 |  |  |
| 网络输入元件 |  |  |
| 输出元件 |  |  |
| 块元件 |  | 包括有效的操作符、函数、功能块、用户程序等 |
| 跳转元件 |  | 跳转条件必须为BIT或BOOL类型（包括可以转换成BOOL的类型）的变量 |
| 标号元件 |  | 与跳转元件配合使用，设置跳转的位置 |
| 返回元件 |  | 用来提供从函数、功能块或程序中提前退出的功能 |
| 注释元件 |  |  |
| 多输入 |  |  |
| 置反 |  | 用来对输入、输出、跳转、返回命令取反，在连接上用一个圆圈表示否定的符号 |
| 置位/复位 |  | 置位的符号是S，复位的符号是R |
| 使能 |  | 用来给选中的块添加一个附加的布尔型使能输入EN端（使能输入端）和一个附加的布尔型使能输出ENO（使能输出端） |

## 组态规则

在CFC语言中逻辑运算是按照元件的执行顺序号的先后顺序进行运算。

* **赋值**

赋值运算可以通过直接将输入元件与输出元件连接在一起，也可以通过在这两个元件之间添加MOVE块来进行组态。



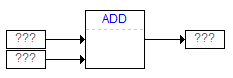
* **功能块调用**

功能块调用是通过添加块元件，并将块类型修改为功能块名，然后关联功能块变量，并对输入、输出引脚连接相应的输入元件、输出元件。



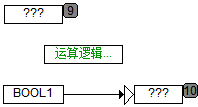
* **函数或操作符调用**

函数或操作符调用是通过添加块元件，并将块元件中类型修改为对应的调用名称，然后对输入引脚、输出引脚分别连接输入元件、输出元件来进行组态。



* **跳转**

跳转是通过添加标号元件，并关联说明，然后添加跳转元件，关联相应的标号说明，并对其输入引脚关联BOOL型变量或常量来进行组态。



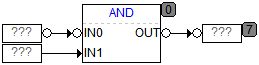
* **返回**

返回是通过添加返回元件来组态，当POU运算到当前返回元件，则后续的逻辑不再运算。



* **置反**

置反是通过选中输入元件的引脚、输出元件的引脚、块元件的输入或输出引脚，然后单击置反菜单来组态。



* **置位/复位**

置位/复位是通过选中输出元件，然后单击置位菜单来组态。

* **注释**

注释是通过添加注释元件，然后输入说明内容来进行组态。



* **块元件级联**

块元件级联是通过将前一个块元件的输出引脚与下一个块元件的输入引脚进行连接来进行组态。

