

0612 【万泉河】浅议 PLC 程序中 SR 和 RS 逻辑的区别

真是印证了一个道理， 越是简单的问题， 争议越大。

这段时间连着写了几篇文章， 关于双线圈， 关于启保停等等， 都是很基础的概念问题。 原本也没觉得有什么亮点。 也只是在一些直播视频讲座时灵机一动想到的， 都是顺便一提， 提完了听众都听懂了。 然而觉得有必要写下来， 给初学者有做启发和帮助。

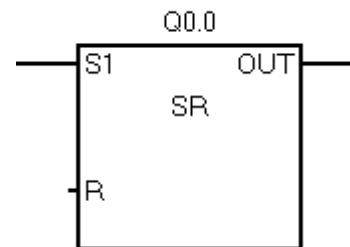
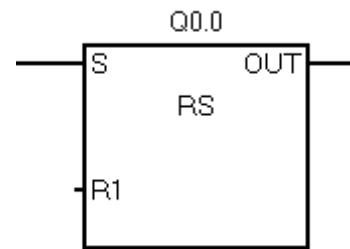
没想到却一波一波的争议， 各种质疑， 非常不可理喻。

既然这样， 那就再向下一点， 下沉到最基本的 SR/RS 指令， 来做个剖析。

PLC 程序中通常有 SR 指令， RS 指令， 以及分开的单独的 R 和 S 指令。 后者不提， 只提前两者。

这两者有什么区别？

估计各位同行在学习 PLC 编程的第一天就遇到了。



编程手册上通常会教大家， RS 指令， S 指令在后面， S 优先。 而 SR 指令， R 指令在后面， R 优先。

然而“优先”这个关键词， 估计那时候大家第一次听说， 还有点烧脑。 烧过之后， 真的后面的编程设计工作反而没什么机会遇到， 然而烧了这一回， 很多人脑子中就形成了深刻的印象， 也错以为是掌握了什么行业核心技能了。

指令的优先级，在编程中随处可见。 比如数学运算符之间， 都是有优先级的。

下表列出了逻辑表达式中可使用的数据类型：

运算	运算符	优先级	第一个操作数	第二个操作数	结果
取反	NOT	3	Bool	-	Bool
求反码			位序列	-	位序列
与	AND、&	8	Bool	Bool	Bool
			位序列	位序列	位序列
异或	XOR	9	Bool	Bool	Bool
			位序列	位序列	位序列
或	OR	10	Bool	Bool	Bool
			位序列	位序列	位序列

—

然而，在实际的工作中，通常很少有人会依赖于优先级进行编程。反而在一些编程规范中，会明确强调程序的易读性，禁止依靠优先级实现程序逻辑。

比如：

知乎

首发于
C语言从入门到精通

...

写文章



赞同 35



分享

七、可读性

1. 算法优先级

注意运算符的优先级，并用括号明确表达式的操作顺序，避免使用默认优先级。防止阅读程序时产生误解，防止因默认优先级与设计思想不符而导致程序错误。

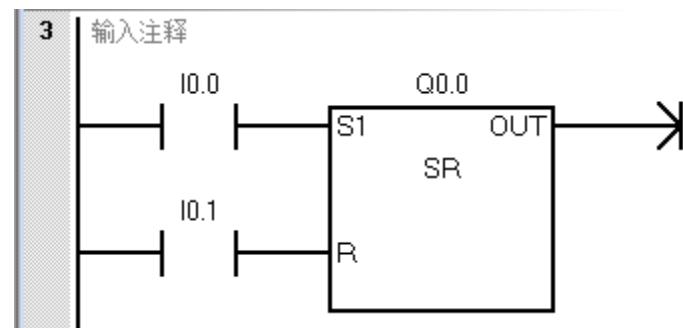
2. 使用有意义的标识代替不易理解的数字

避免使用不易理解的数字，用有意义的标识来替代。

3. 少用技巧性很高的语句

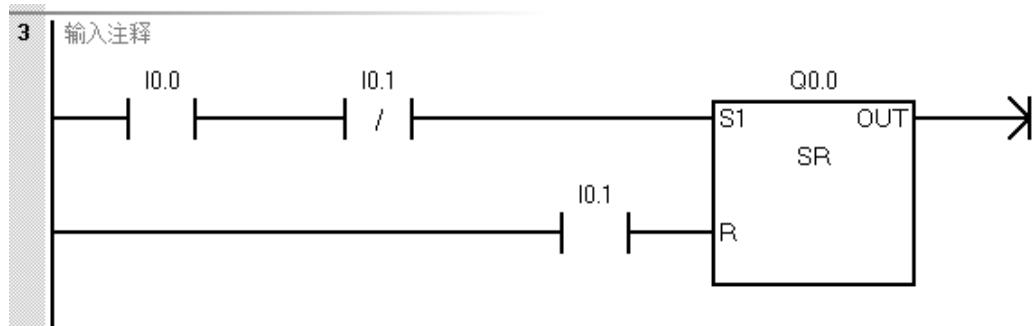
不要使用难懂的技巧性很高的语句，除非很有必要时。高技巧语句不等于高效率的程序，实际上程序的效率关键在于算法。

所以，一段别人写好的逻辑：

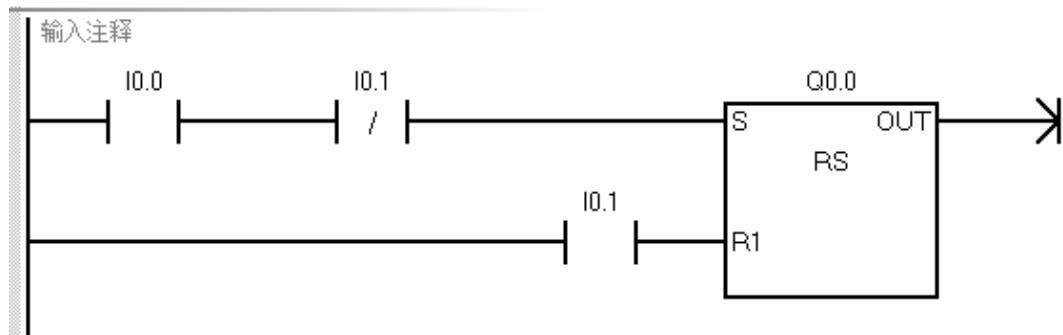


你没有必要从中读出 I0.0 还是 I0.1 优先级更高的潜台词，更没必要抬杠什么如果恰巧 I0.0 和 I0.1 同时发生，会有什么意外结果。并以此标榜自己发现了什么程序漏洞。因为很大的可能性是，实际现场中对这种偶然情形的发生根本不在乎。

因为，如果在乎，就应该直接在逻辑中体现，提醒编程者自己，也提醒后来的阅读者。



这时候，如果 I0.0 和 I0.1 同时发生，而逻辑上又确实有安全需求，不管后面使用的是 SR 还是 RS，运行结果是一样的，都是以 I0.1 优先。



事实上，在 SMART 200 和 TIA PORTAL 中，同样的梯形图符号 SR 和 RS，优先级好像还是颠倒的。然而基于上述的规范，你根本没必要记忆和发现他们。

所以，结论是，对于一个规范程序的作者和阅读者，SR 逻辑和 RS 逻辑没有区别。

茴香豆的四种写法，没有区别。随便用其中的一种即可。

忘了它们吧！

