## 第1章 Process Simulate 循环仿真

在 Process Simulate 软件中提供了两种可以加载制造工艺过程仿真研究文件的模式。

1. "标准模式" · : 基于时间的仿真模式。将定义好的操作序列从开始到结束进行 仿真模拟,通过标准模式只能仿真模拟一个生产周期。

2. "生产线仿真模式" · : 基于事件的循环仿真模式。生产线仿真模式可以基于事件和触发器来仿真模拟多个动态生产周期,而不是预定义的序列。

在生产线仿真模式下, Process Simulate 中的操作(除复合操作之外)都会自动生成 Operation\_end 信号。当仿真过程中需要评估执行哪些操作时,这些信号会被用作默认转 换条件。当运行模拟仿真时,每次某个操作执行结束时,这个特定的运算操作结束信号 被设置为 true,信号持续1个计算周期(即时间间隔),然后重置为 false,如图 1-1 所示。 每次执行操作时,都会重复此操作。



1.1 创建操作序列的循环仿真

要实现操作序列的循环仿真,需要进入"生产线仿真模式"中完成创建。在这个过程中,需要创建循环仿真结构,并创建需要的"信号""过渡条件""信号事件"等要素, 从而实现整个操作序列的循环仿真。具体操作步骤如下。

 ●1 单击"打开研究"按钮≥,如图 1-2 所示,在弹出的"打开"对话框中,选择"Session
 1-Cyclic Simulation"文件夹中的"S01-E01.psz"研究文件,然后单击对话框中的"打开" 按钮,如图 1-3 所示。系统将以标准模式打开该研究文件,如图 1-4 所示。

——Process Simulate 基于事件的循环仿真应用



图 1-2

🌺 打开						>	<
$\leftarrow$ $\rightarrow$ $\checkmark$ $\uparrow$	Sector Se	ession 1 - Cyclic Simulati	, ~ C		Session 1 - (	Cyclic Si	
组织▼ 新建文	件夹				≣ •		)
> 🟡 Aspera	~ 名称	修改日期	类型	大小			
> 🔷 同步空间	Exercise	2020/9/4 10:43	文件夹				
~ 📮 此电脑	📒 S01 SysRoot	2022/2/28 11:31	文件夹				
> ▶ 视频	脧 S01-E01.psz	2020/9/4 12:14	Process Simulat	347 KB			
<ul> <li>&gt; 图片</li> <li>○ 文档</li> <li>&gt; 下载</li> <li>○ 音乐</li> <li>&gt; ● 桌面</li> <li>&gt; ○ DATA</li> </ul>	🏠 S01-E02.psz	2017/7/10 18:42	Process Simulat	324 KB			
	文件名(N): S01-E01	.psz	~	· 研究文件 (*.	psz)	~ 取消	

图 1-3



图 1-4

●2 右击"操作树"查看器中的"STATION"操作,如图 1-5 所示,在弹出的快捷 菜单中选择"设置当前操作"选项,如图 1-6 所示。在"序列编辑器"查看器中,可以 看到已将"STATION"操作设为当前操作,如图 1-7 所示。在"序列编辑器"查看器中, 单击"正向播放仿真"按钮 ▶,运行该操作仿真。





图 1-7

**13** 可以看到仿真模拟根据"序列编辑器"中定义的操作序列只运行了一个周期, 这是基于时间的仿真的标准行为。接下来实现操作序列多周期的循环仿真。

☑ 单击"主页"菜单下的"生产线仿真模式"按钮(图 1-8),在弹出的"切换研究模式"对话框中单击"否"按钮(图 1-9),并在弹出的警告对话框中单击"关闭" 按钮(图 1-10),关闭该警告对话框。



----Process Simulate 基于事件的循环仿真应用



图 1-10

**05** 此时已将操作序列切换到了"生产线仿真模式",在"操作树"查看器中有了 一个"LineOperation"根节点(图 1-11)。





**D6** 将"LineOperation"设置为当前操作,并在序列编辑器中播放仿真。可以看到, 此时仿真的运动动作不受控制。原因就是基于事件的仿真行为与基于时间的仿真是不同的。

07 接下来讲解生产线仿真模式是如何工作运行的。在"主页"菜单下,执行"查看器"
 (图 1-12) → "信号查看器"命令(图 1-13),打开"信号查看器"面板(图 1-14)。



信号查看器								
🧳 若   🍸 🍢 🚰 💱 隊 🕅	2							
信号名称	内存	类型	Robot Signal Nar	地址	IEC 格式	PLC 连接	外部连接	资源
SIFT DRIVE DOWN end		BOOL		No Address	1			LIFT
R1 LOAD PART end		BOOL		No Address	1			R001
M LIFT DRIVE UP end		BOOL		No Address	1	(m)		LIFT
R2 WELD end		BOOL		No Address	I. Contraction			R002
R3 WELD end		BOOL		No Address	1			R003
No R2 TDR end		BOOL		No Address	1			R002
R3 TDR end		BOOL		No Address	1			R003
R1 REMOVE PART end		BOOL		No Address	1			R001
序列编辑器 路径编辑器	涉查看器	信号查	看器					

图 1-14

**DB** 在"信号查看器"左边的"信号名称"栏中,可以看到各种操作的"结束信号", 目前也是唯一的信号。当 Process Simulate 的每一个操作被创建时,该信号会被自动生成。 "结束信号"为布尔信号,并且具有简单的行为:当操作结束时,其结束信号从 false(0) 变为 ture(1),持续1个时间间隔,然后变回 false,从而有效发出1个短的 true 脉冲。

● 单击"序列编辑器"查看器中的"定制列"按钮 ■,如图 1-15 所示。在弹出的"定制列"对话框中的"可用字段"列表中,依次选择"过渡""正在运行"选项,并通过单击,按钮,将"过渡""正在运行"选项放入右边"按以下顺序显示字段"列表中,单击"确定"按钮,如图 1-16 所示,退出"定制列"对话框。可以看到"序列编辑器"查看器中新增了"过渡""正在运行"列,如图 1-17 所示。



序列编辑器		
📾 👾 Q Q Q 💷 🟥	• • • •	
序列编辑器	过渡	正在运行
🖃 晶 LineOperation		
E STATION	\$	
LIFT_DR	<b>‡</b>	
R1 LOAD	<b>‡</b>	
LIFT_DR	<b>‡</b>	
💷 📲 WELDING	\$	
R1 REMO	+	
4		Þ
序列编辑器 路径编辑器	干涉查看器	信号查看器

图 1-17

——Process Simulate 基于事件的循环仿真应用

● "过渡"列包含了链接的操作之间指定的转换条件。与"标准模式"不同,在"生产线仿真模式"中,一个操作会在其前一个操作任务执行完成的情况下被执行,并且,还要满足操作间指定的转换条件。如果我们要检查操作间的过渡条件,可以双击"序列编辑器"查看器中"过渡"栏下面的 \$ 按钮。

例如,检查开始进行"LIFT\_DRIVE\_UP"操作的过渡条件,首先双击"R1 LOAD PART"操作右边"过渡"栏中的 ≵ 按钮,如图 1-18 所示,弹出"过渡编辑器"对话框,如图 1-19 所示,单击"过渡编辑器"对话框中的"编辑条件"按钮,在弹出的对话框中可以看到"LIFT\_DRIVE\_UP"操作执行的过渡条件是 "R1 LOAD PART\_end"(图 1-20),这表示每当 "R1 LOAD PART\_end" 信号被设置为 true, "LIFT\_DRIVE\_UP"操作就将会启动。

序列编辑器		
● ﷺ <b>④ Q Q @ =</b> ∰   ₩ ₩ 세	· (    ) •	• H M 🕨
序列编辑器	过渡	正在运行
🖻 晶 LineOperation		
🖻 🌄 STATION	\$	
	<b>‡</b>	
R1 LOAD PART	<b>₹</b>	
LIFT_DRIVE_UP		AD PART
💷 📲 WELDING	\$ <u></u>	
R1 REMOVE PART	<b>‡</b>	

图 1-18

过渡编辑器 - R1 LOAE	PART			
公共条件: [""R	1 LOAD PART_en	d""		编辑条件
操作 LIFT_DRIVE_UP	分支类型 	条件		
•				

图 1-19



图 1-20

11 接下来检查仿真是如何运行的。

通常情况下,在"序列编辑器"查看器中单击"正向播放仿真"按钮 ▶ 后如图 1-21 所示,仿真将立即运行。在本例中, "LIFT DRIVE DOWN"操作开始运行。



图 1-21

[2] 在"序列编辑器"查看器中,单击"正向播放仿真"按钮 ▶,然后立刻单击"暂 停仿真"按钮 ■,可以看到"LIFT\_DRIVE\_DOWN"操作在运行并在右侧"正在运 行"列中显示了一个运行图标 ▶(图 1-22)。在仿真运行过程中,每次"LIFT\_DRIVE\_ DOWN"操作结束时,"LIFT\_DRIVE\_DOWN \_end"信号就在一个时间间隔内变为 true,并立即再次开始启动操作。此循环行为直到仿真停止,并且与仿真运行时的所有 操作相关。



图 1-22

[3] 在"序列编辑器"查看器中,再次单击"正向播放仿真"按钮 ▶,如图 1-23 所示, 让仿真运行一段时间。此时观察升降装置,可以看到该装置向下移动到底部位置,然后 再次回到顶部,接着开始又一次向下移动,如此循环往复。这是因为"LIFT\_DRIVE\_ DOWN"是序列中的第一个操作,其会运行直到完成,然后再次设置"LIFT\_DRIVE\_ DOWN end"信号为 true,并且重新启动运行。

——Process Simulate 基于事件的循环仿真应用

序列编辑器					
ee ∰ € ⊖ Q ■ 🛤 H H 4	- II 🕨	▶ н н   №	4.	20 – –	
序列编辑器	过渡	正在法行	) 5	10 15	20
🖻 🔒 LineOperation	- <u>1</u>	向播放仿具			
E STATION	\$				
🔁 LIFT_DRIVE_DOWN	<b>‡</b>	•	1		
🔤 🕞 R1 LOAD PART	<b>‡</b>		· ·		
LIFT_DRIVE_UP	<b>‡</b>				
🖭 晶 WELDING	\$				
R1 REMOVE PART	<b>‡</b>				
<		Þ	•		
序列编辑器 路径编辑器 干涉查看	諸器 信号查	音器			

图 1-23

**14** 如果希望以循环仿真的方式运行整个操作,以便每个循环在前一个循环结束时 开始,操作步骤如下。

在 "LIFT\_DRIVE\_DOWN" 之前建立一个新的操作,以控制整个序列操作执行的开始。使用 "非仿真操作" 类型的操作(这样的操作在仿真过程中不执行任何实际操作) 来完成此设定。操作步骤如下。

(1)在"操作树"查看器中,将"STATION"操作设置为当前操作。

(2)在"操作"菜单下,执行"新建操作"(图 1-24)→"新建非仿真操作"命令
 (图 1-25),在弹出的"新建非仿真操作"对话框中输入操作名称 INITIALIZATION,持续时间为默认值 0,如图 1-26 所示。

文件 主页 视图	建模  机器人  操	作工艺	●米 新建复合操作 ■■
		<b>米</b> 医 汤加当前位置	業 新建非仿真操作
创建操作	בן אַנגאָשן אָשע אַרָּדָרָי - דָּבָּיָצָין אַנגאָשָאָן אָש		■米 新建对象流操作
P	<b>될 1-24</b>		图 1-25
	新建非仿真操作	×	
	名称: INITIALIZATION		
	范围: STATION	•	
	描述:		
	起始操作	*	
	持续时间: 0.00	<b>A</b>	
	确定	取消	

图 1-26

(3)在"序列编辑器"查看器中,将新建的"INITIALIZATION"操作拖曳到"LIFT\_ DRIVE\_DOWN"操作之前,如图 1-27 所示;此时,新创建的"INITIALIZATION"操 作成为序列中的第一个操作,如图 1-28 所示。

序列编辑器					
©	- II ►	<b>H H H</b>	i i	0.	00
序列编辑器	过渡	正在运行	2.	5.	10
STATION	兺				
LINT_DRIVE_DOWN	<b>‡</b>			1	
AD PART	<b>‡</b>				
LIFT_DRIVE_UP	<b>‡</b>				
🖭 晶 WELDING	¢				
R1 REMOVE PART	<b>‡</b>				
INITIALIZATION	<b>‡</b>		Þ		

图 1-27

序列编辑器		
∞ ∰ Q Q Q @ ∰   W H 4	$\mathbf{A} = \mathbf{A} \mathbf{A} \mathbf{A}$	▶ N M 🌬
序列编辑器	过渡	正在运行
E C STATION	ŝ	
INITIALIZATION	<b>‡</b>	
- 🔁 LIFT_DRIVE_DOWN	<b>‡</b>	
R1 LOAD PART	<b>‡</b>	
LIFT_DRIVE_UP	<b>‡</b>	
🛨 📲 WELDING	ŝ	
R1 REMOVE PART	<b>‡</b>	

图 1-28

(4) 再单击"链接"按钮,将新创建的"INITIALIZATION"操作关联到"LIFT\_ DRIVE\_DOWN"操作,如图 1-29 所示,结果如图 1-30 所示。相关操作方法在《西门子 数字化制造工艺过程仿真——Process Simulate 基础应用》一书中有详细描述,本书不再 赘述。

序列编辑器			
	₩ <b>4 4 1</b>	▶ ▶ ▶	₩   Þ <sub>©</sub>
		шидп	
INITIALIZATION	<b>‡</b>		
R1 LOAD PART	‡ ‡		
🕀 📲 WELDING	\$ \$		

图 1-29

——Process Simulate 基于事件的循环仿真应用

73 7 3 W 3 H 4 H H			
© \$``O Q Q <b>©</b> \$\$\$	II	• • H	H 🖻
序列编辑器	过渡	正在运行	2 5 10
E STATION	\$		
INITIALIZATION	<b>‡</b>	(	
	<b>‡</b>		
R1 LOAD PART	<b>‡</b>		
LIFT_DRIVE_UP	<b>‡</b>		
🗄 📲 WELDING	\$		
R1 REMOVE PART	<b>‡</b>		

图 1-30

(5)为了控制整个操作序列的执行,将创建一个信号。在"信号查看器"面板中, 单击"新建信号"按钮 ≥,如图 1-31 所示,在弹出的"新建"信号对话框中,选择信 号类型为"显示信号",名称为 FIRST,单击"确定"按钮退出对话框(图 1-32)。在"信 号查看器"面板中,可以看到新增加了一个 FIRST 信号,如图 1-33 所示。

信号查看器							
🤾 🗡   🍸 🍢   🕵 💱 💖							
信之名称	内存	类型	Robot Signal Nar				
→新建信号 IVE DOWN end		BOOL					
R1 LOAD PART end		BOOL					
🐳 LIFT DRIVE UP end		BOOL					
🐳 R2 WELD end		BOOL					
N R3 WELD end		BOOL					
🐳 R2 TDR end		BOOL					
🐳 R3 TDR end		BOOL					
R1 REMOVE PART end		BOOL					
M INITIALIZATION end		BOOL					
序列编辑器 路径编辑器 3	F涉查看器	信号查	看器				

图 1-31

新建			×
类型	数量	名称   关键信号	确定
	1	入硬旧与 FIRST 次派绘山住中	重置
□ ~ 资源输工信号	0	资源输入信号	取消

图 1-32

信号查看器			
🏄 🏹   🍸 🍢   🗗 💱 🐼 🖗	¢;		
信号名称	内存	类型	Robot Signal Nar
LIFT DRIVE DOWN end		BOOL	
R1 LOAD PART end		BOOL	
LIFT DRIVE UP end		BOOL	
R2 WELD end		BOOL	
R3 WELD end		BOOL	
R2 TDR end		BOOL	
R3 TDR end		BOOL	
R1 REMOVE PART end		BOOL	
MINITIALIZATION end		BOOL	
🔏 FIRST		BOOL	
序列编辑器 路径编辑器 3	F涉查看器	信号查	
	图 1-33		

(6)在"序列编辑器"查看器中,双击"INITIALIZATION"操作右边"过渡"列中的 按钮,如图 1-34 所示,在弹出的如图 1-35 所示的"过渡编辑器"对话框中单击"编辑条件"按钮,在弹出的如图 1-36 所示的对话框中将默认的信号 INITIALIZATION\_end 改为 NOT FIRST。单击两次"确定"按钮,退回 Process Simulate 软件应用界面。



图 1-34

			 	The stars have been
公共条件: │INI	IALIZATION_end			编辑杀件
操作	分支类型 条	件		
LIFT_DRIVE_DOWN				
(				
(				

图 1-35

——Process Simulate 基于事件的循环仿真应用

过渡编辑器 - INITIALIZA	TION	-	
NOT FIRST			
	确定	F	辺治
	WHILE		K/H

图 1-36

(7)在"序列编辑器"查看器中,右击"LIFT\_DRIVE\_DOWN"操作(图1-37), 在弹出的快捷菜单中选择"信号事件"选项(图1-38),弹出"信号事件"对话框,在"要 生成/连接的对象"栏中选择"first"信号,并将信号设为true,在"开始时"栏中 选择"任务开始后"选项(图1-39),单击"确定"按钮,退出对话框,结果如图1-40 所示。



图 1-39

确定



序列编辑器 路径编辑器 干涉查看器 信号查看器

取消

(8)在"序列编辑器"查看器中单击"正向播放仿真"按钮 ▶,让仿真运行。可以看到还是一个单循环的完整操作,出现这种情况的原因是"INITIALIZATION"操作位于序列之首,会一直被执行,因为在它前面没有过渡条件,所以会立即开始,立即完成(持续时间=0)。"INITIALIZATION"操作到"LIFT\_DRIVE\_DOWN"操作的过渡条件被改为 NOT FIRST,值设为 true(即 NOT FIRST = true,那么 FIRST = false)。当"INITIALIZATION"操作完成后,立刻会有两个事情发生:"LIFT\_DRIVE\_DOWN"操作开始,并通过使用"信号事件"将 FIRST 信号设置为 true。

"INITIALIZATION"操作再次启动之后,当操作完成时,过渡条件被评估为 false(NOT true = false),这意味着在第一个循环之后直到新的仿真开始(因为 NOT true 是恒定的 false),将不再继续执行"LIFT DRIVE DOWN"操作。

(9)为了实现不断循环的仿真操作,更改从"INITIALIZATION"操作到"LIFT\_ DRIVE\_DOWN"操作的过渡条件,以便在序列的最后一个操作"R1 REMOVE PART" 结束时,其过渡条件值依然为 true。

(10) 在"序列编辑器"查看器中,双击"INITIALIZATION"操作右边"过渡" 列中的 ◆ 按钮,在弹出的"过渡编辑器"对话框中单击"编辑条件"按钮,在弹出的对 话框中将信号 NOT FIRST 改为 NOT FIRST OR "R1 REMOVE PART\_end"(图 1-41)。 单击两次"确定"按钮,退回 Process Simulate 软件应用界面。



图 1-41

(11)将"Line Operation"设置为当前操作,并在序列编辑器中播放仿真。可以看 到仿真会一直持续循环进行,直到停止仿真。

(12)停止仿真,在"文件"菜单下,执行"断开研究"→"另存为"命令,如图1-42
 所示,弹出"另存为"对话框,输入另存为文件名: S01-E01\_OK,如图1-43 所示,单击"保存"按钮。

— Process Simulate 基于事件的循环仿真应用

文件	视图 建模 机器人	🚵 另存为	×
□ 最近的文件 ▶		保存在(I): Texercise	▼ ← 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
		名称 中读访问	修改日期 类型 2022/3/1 14:42 Process Simulate P
🔅 断开研究 🕨	新建研究		
→ 导入/导出 ▶		桌面	
		库	
关于( <u>A</u> )	🔑 以生产线仿真模式打开	此电脑	
(?) 帮助(日)	 		
		KA)÷û	
还坝	另存为	文件名(N): S01-E01_OK.ps	sz. ▼ 保存(S)
🔀 退出(E)		保存类型(T): 研究(*.psz)	▼ 取消
		☑ 包含宏和 SPSS 文件夹(1)	
Ĕ	图 1-42	冬	1-43

## 1.2 创建操作间过渡条件

"过渡条件"就是链接的操作之间指定的转换条件。与"标准模式"不同,在"生 产线仿真模式"中,一个操作要被执行,除了其前一个链接操作任务已执行完成外,还 要满足操作间指定的转换条件,即"过渡条件",否则操作也不能正常运行。下面通过 创建或者编辑操作间的"过渡条件",进一步理解"过渡条件"在仿真过程中带来的影响。

☑ 单击"以生产线仿真模式打开研究"按钮查,弹出"打开"对话框,选择"S01-E01\_OK.psz"研究文件(图 1-44),然后单击"打开"按钮。系统将以生产线仿真模式打开 该研究文件。





02) 在"序列编辑器"查看器中,单击"正向播放仿真"按钮 ▶,如图 1-45 所示, 运行仿真。可以看到操作仿真将会无限循环运行,直到被停止。

序列编辑器		
ee ∰ € € Q Q @ ∰   H4 H 4 4	· III 🕨 🕨 🕨 🕨	0.00 – –
序列编辑器	过,正在运行	) 5 10 15
🖻 晶 LineOperation	论问播放仿具	
🖻 🖧 STATION	\$	
──■ 初始操作	<b>‡</b>	2
🔁 LIFT_DRIVE_DOWN	<b>‡</b>	1
R1 LOAD PART	<b>‡</b>	
LIFT_DRIVE_UP	<b>‡</b>	
🗄 📲 WELDING	\$	
R1 REMOVE PART	<b>‡</b>	
4		•
序列编辑器 路径编辑器 干涉查看器	信号查看器	

图 1-45

3 为了更好地理解"过渡条件",将"R2 WELD"和"R3 WELD"之间的过渡链接移除。在"序列编辑器"查看器中,选择"R2 WELD"和"R3 WELD"操作,然后单击"断开链接"按钮 轢(图 1-46),结果如图 1-47 所示。









砬 在"序列编辑器"查看器中,再次单击"正向播放仿真"按钮 ▶,运行仿真。
 可以看到操作仿真并没有受到影响,依然是无限循环运行直到被停止。原因就是各操作
 间的过渡条件并没有出现冲突或者无法满足的问题。接下来,对操作的过渡条件做一些

——Process Simulate 基于事件的循环仿真应用

编辑修改,看看对仿真结果会带来什么影响并找出问题的原因加以解决。

□5 在"序列编辑器"查看器中,双击"WELDING"操作右边"过渡"列中的 ♥ 按钮(图1-48),在弹出的"过渡编辑器"对话框中(图1-49)单击"编辑条件"按钮, 在弹出的对话框中,将信号改为 "R2 WELD\_end" AND "R3 WELD\_end"(图1-50)。单 击两次"确定"按钮,退回 Process Simulate 软件应用界面。

₩ 注意

"R2 WELD\_end" AND "R3 WELD\_end" 完成信号是 "R1 REMOVE PART" 操作开始的过渡条件。

序列编辑器		
© ; • • • • • • • • • • • • • • • • • •		) 🕅 🌬
序列编辑器	过渡	正在运行
🗆 📲 LineOperation		
🖻 🦥 STATION	\$	
──■ 初始操作	<b>‡</b>	
- 🔁 LIFT_DRIVE_DOWN	<b>‡</b>	
R1 LOAD PART	<b>‡</b>	
LIFT_DRIVE_UP	<b>‡</b>	
	ŧ	
🗄 📲 WELD		NOUE DADT
🗄 品 TIP DRESS		LMOVE FARI
R1 REMOVE PART	<b>‡</b>	
<		►
序列编辑器 路径编辑器 干涉查看器	信号查看	<u>88</u> 88

图 1-48

过渡编辑器 - WELDING	× 过渡编辑器 - WELDING	- 🗆 ×
公共条件:	编辑条件	LD_end"
操作 分支类型 条件 R1 REMOV		
	•	
确定	取消 确定	取消

图 1-49

图 1-50

● 在 "序列编辑器" 查看器中,再次单击 "正向播放仿真" 按钮 ●,运行仿真。
 可以看到操作仿真在 "R2 WELD" 和 "R3 WELD" 操作完成焊接后停止了。下面分析
 一下原因。

(1)从"WELDING"复合操作到下一个操作"R1 REMOVE PART"的过渡条

件是两个焊接操作"R2 WELD"和"R3 WELD"过渡条件的组合,即"R2 WELD\_ end"和"R3 WELD\_end"。也就是"R1 REMOVE PART"操作要开始执行,需要"R2 WELD\_end"和"R3 WELD\_end"同时发出 true 的脉冲信号才可以。

(2) 接下来分别查看 "R2 WELD"和 "R3 WELD"操作的工作时间。在"操作树" 查看器中,右击 "R2 WELD"操作,在弹出的快捷菜单中,选择"操作属性"选项 (图 1-51);在弹出的属性对话框中,单击"时间"折页项,可以看到 "R2 WELD"操 作的工作时间为 19.93 秒(图 1-52)。同理,可以看到 "R3 WELD"操作的工作时间为 19.65 秒(图 1-53)。

		属性 - R2 WELD	×
操作树		常规 时间 工艺 资源	(一产品)
ф <b>4</b>	复制焊缝操作	时间参数:	0.00
□□ □ ❷ 操作	< 删除	开始时间:	0.00
	在前面添加 TSB	经验证的时间:	19.93
	在后面添加 TSB		
🕂 🗆 😋 R1 LOAD PA	■ 操作开始条件		
E C & WELDING	- ■ 操作属性		
⊕ □ ■ <mark>→</mark> R2 VELD ⊕ □ ■→ R3 ↓SLD ⊟ □ ₽ TIP DRESS			确定取消
图 1-51		冬	1-52

属性 - R3 WELD		×
常规时间工艺	资源   产品	
时间参数: 分配时间: 开始时间: 经验证的时间:	0.00 <b>•</b> 0.00 •	
	确定    取消	

图 1-53

(3) "R2 WELD"和 "R3 WELD"操作工作时间不同,也就意味着这两个操作 永远不会同时结束,因此也就不能在同一时刻发出 "R2 WELD\_end"和 "R3 WELD\_ end"为 true 的脉冲信号。其结果就是 "R2 WELD"和 "R3 WELD"过渡条件永远不成立,

——Process Simulate 基于事件的循环仿真应用

因此造成"R1 REMOVE PART"操作无法启动。

 "R1 REMOVE PART"操作不运行的原因找到了,现在来解决这个问题。已经 知道"R2 WELD"操作的工作时间比"R3 WELD"操作的工作时间长,因此指定"R1 REMOVE PART"操作的过渡条件时,只考虑工作时间较长的"R2 WELD"操作结束的 脉冲信号。

在 "序列编辑器" 查看器中,双击 "WELDING" 操作右边 "过渡"列中的 ♥ 按钮(图 1-54),在弹出的"过渡编辑器"对话框中(图 1-55),单击对话框中的"编辑条件"按钮,在弹出的对话框中,将信号改为: "R2 WELD\_end"(图 1-56)。单击两次"确定"按钮,退回 Process Simulate 软件应用界面。

序列编辑器		
∞ ; ⊙ ⊙ Q @ ; ; K k 4 4		- H 🗰 🌬
序列编辑器	过渡	正在运行
🗆 品 LineOperation		
🖻 🐝 STATION	\$	
■ 初始操作	ŧ	
LIFT_DRIVE_DOWN	<b>‡</b>	
R1 LOAD PART	<b>‡</b>	
LIFT_DRIVE_UP	<b>‡</b>	
🖻 品 WELDING	- ŧ	
🗉 📲 WELD		NOUR DADT
🗄 品 TIP DRESS		LMUVE FARI
R1 REMOVE PART	<b>‡</b>	
4		Þ
序列编辑器 路径编辑器 干涉查看器	信号查看	器

图 1-54

过渡编辑器 - WELDING	×	过渡编辑器 - WELDING	- 🗆 ×
公共条件:	编辑条件	"R2 WELD_end"	
操作 分支类型 条件 R1 REMOV			
		确定	取消
团 1 55		团 1.50	-build
图 1-33		图 1-30	

 四 在 "序列编辑器" 查看器中,单击 "正向播放仿真" 按钮 ▶,运行仿真,验证 所做的更改。可以看到整个操作又恢复到循环仿真的状态运行。

09 将完成的研究文件另外保存。



注意,如果问题依然没有解决,还可以进一步查找原因,步骤如下。

(1)在"序列编辑器"查看器中,单击"正向播放仿真"按钮 ▶,然后立刻单击"暂 停仿真"按钮 ■。然后在"操作树"查看器中,右击"R1 REMOVE PART"操作,在 弹出的快捷菜单中选择"操作开始条件"选项(图1-57),弹出"操作开始条件"对话框, 可以看到"R1 REMOVE PART"操作的开始条件是 "R2 TDR\_end" AND "R2 WELD\_ end"(图1-58)。多了一个"R2 TDR\_end"过渡条件,接下来把此过渡条件删除。



图 1-57



(2)在"序列编辑器"查看器中,双击"R2 TDR"操作右边"过渡"列中的 按钮(图 1-59),弹出"过渡编辑器"对话框,单击"编辑条件"按钮,在弹出的对话 框中,将信号"R2 TDR\_end"删除(图 1-60)。单击两次"确定"按钮,退回 Process Simulate 软件应用界面。



(3)在"序列编辑器"查看器中,单击"正向播放仿真"按钮 ▶,运行仿真,验 证所做的更改,可以看到问题已经解决。