

逻辑门

目的:

为了解决数字电路中更复杂的问题,必须使用逻辑。这个过程包括应用真值表和公式来 创建手头问题的解决方案。逻辑使用两种基本状态。人们对这两个状态使用不同的术语:"是 (Yes)"和"不是(No)"、"高(High)"和"低(Low)"或者"1"和"0"。

在数字电路中,能实现基本逻辑运算的电子电路,叫逻辑"门"电路。通常有几种基本 类型的"门"电路,每个"门"在输入和输出之间具有数学关系,可以写成布尔表达式或真 值表。

通过本节的内容,你将学习与逻辑门有关的知识。一旦你掌握了逻辑门的基本知识,解决一个复杂的问题就容易多了。

设备:

RoBo Pro 软件

过程:

打开 RoBo Pro 软件并开始一个新的文件。将使用级别(level)设置为"Level 5: Objects (等级 5:对象)"



点击工具栏中的 "COM / USB" 按钮, 在设置面板中选择 "仿真 (Simulation)" 模式, 然后选择 OK。



剕门

非门是仅有一个输入和一个输出的最简单的门。首先在编程窗口中放置一个按钮控件。 你可以在"操作模块(operating elements)"->"控制模块(control elements)"子分 类中找到"按钮(button)"模块。将其拖拽到编程窗口中。

- Operators	
Operating elements	
Displays	
Control elements	
⊕ Drawing	
-Library	
Program elements	
Button 9.2.1: Button	
Button	≡ .

右键单击"按钮"模块,在其属性面板中,将其命名为 ButtonA。你还可以通过点击 Button color 旁边的"编辑(Edit)"键为按钮设置不同的颜色。注意:"Pushbutton switch (按钮开关)"这个选项的复选框要确认为"√"的状态。 Cedutech[®]北京中教仪人工智能科技有限公司

	Button
· · · · · · · ·	Button text: ButtonA
	Button color: Edit
ButtonA	Text color: Edit
	Pushbutton switch: 🔽
	Initially pressed:
	Note: The "Button text" field is used to
	OK Cancel

现在需要一个虚拟的显示灯控件用来显示输出状态。你可以在"操作模块(operating elements)" -> "显示(displays)"子分类中找到"显示灯(display lamp)"模块。将其 拖拽到编程窗口中。

operators	
Operating elements	
Displays	
Control elements	
Drawing	
Library	
User library	_
Program elements	
Var= 0	
Var= 0	
Var= 0	- · ·
Var 2	

右键单击"显示灯 (display lamp)"模块,将其重新命名为 YLamp。你也可以通过设置改变灯的颜色。注意:"Initially on (初始状态)"这个选项的复选框要确认为"√"的状态。



现在我们有了构建"非门"逻辑所需的两种虚拟控件。

从"编程模块 (program elements)"->"输入、输出 (inputs, outputs)"子分类中 找到"面板输入 (panel input)"模块。将其拖拽到编程窗口中。

···· Variables, timers	
Commands	_
Branch, wait	=
Inputs, outputs	
Operators	
Operating elements	
Displays	
Control elements	
Drawing	
Library	
User library	
	Ŧ
Program elements	
	^
- M1	Ξ
- <mark>01</mark>	
Panel Input 8.7.6: Panel input	
Panel	

右键单击"面板输入(panel input)"模块,在对话框里"Panel(面板)"标签页中, 将其配置给前面建立好的名为"ButtonA"的按钮控件。

																•	Input 📑 🖬 👘
	•				•		•			•			•				
		·		•	•	•	•	•	•	·	•	•	·	•	·	•	Universal Counter Motor Panel TX • •
					•		•		•	•			•				
																•	Select a button or slider from panel:
-	•	•	•		•	•	•	•	•	•	•	•	•				Main ann ann an
																	Main program
																	ButtonA
																1	
	•	•					•	•		•		-	•				
		_	_	_	_	_	1									-	
		B	uff	O	nΑ												
		-															
										•							1
																	Interface / Extension
		_															
				M	ain	pro											IF1 V
-		F	ABC	B	itto	nA	F		•	•			•				
		-	-	0									•				
																	Connection
																	Local: only when function is entered
											•						Static: always hound
	•	•					•	•	•	•			•				Static. always bound
																	Object: when object is created
														11			Let KOBO Pro decide
						-	-										

从 "编程模块 (program elements)" -> "输入、输出 (inputs,outputs)" 子分类中 找到 "面板显示 (panel display)" 模块。将其拖拽到编程窗口中。

右键单击 "面板显示(panel display)" 模块,将其分配给前面建立好的名为 "YLamp" 的显示灯控件,然后选择 "OK" 退出。



现在我们要将"非门"的逻辑添加到程序中。

从"编程模块 (program elements)"->"运算符 (Operators)"子分类中,将下面列出的第一个模块拖拽到编程窗口中。



右键单击这个模块,将运算符设置为 "not (逻辑非)",点击 "OK" 键后,你将看到该 模块的形状和标识都发生了改变。



参照下图,用程序连接相应的模块。



"非门"真值表:

在线运行上面的程序,在下面的表格中记录你观察到的结果。

"ButtonA"按钮被按下时的数字信号为 1, 抬起时是 0。观察"YLamp"灯的亮灭, 如果灯不亮则在列表相应位置中放置一个 0, 如果灯亮则放置一个 1。

NOT	
(Button)A	Y (Lamp)
0	
1	

"非门"布尔表达式:

布尔表达式是详细说明输入和输出之间关系的数学表达式。在我们的例子中的输入被称为 A。因为 A 有两个状态,当输入是 High 时,我们称它为 A,当输入为 Low 时,称为非A。我们通过^(Ā)来表示 NOT A。

在我们的示例中,输出被称为 Y。通过上面的真值表你可以确定"非门"布尔表达公式:

Y = A

与门:

我们在编程窗口中放置两个"按钮(Button)"控件,分别命名为"Button A"和"Button B"。建立两个"面板输入 (panel input)"模块,分别分配给这两个按钮。

再放置一个命名为 "Y Lamp" 的显示灯控件。建立一个"面板显示 (panel display)" 模块,分配给这个灯。

右键点击"运算符 (operator)"模块,将其设置为"and (逻辑与)"。



参照下图,将各模块连接起来。



"与门"真值表:

在线运行程序,在下面的表格中记录观察到的结果。点击按钮将使其从0切换到1。观察不同情况下灯的亮灭,如果灯不亮,则在Y列中放置一个0。如果灯亮了,则放置1。

AND		
А	В	Y
0	0	
1	0	
0	1	
1	1	

"与门"布尔表达式:Y=AB

或门

停止程序,然后右键单击"运算符(Operator)"模块,将其更改为"or(逻辑或)"。



运行程序,填写下面的真值表,并试着写出 OR 门的布尔表达式。

OR		
А	В	Y
0	0	
1	0	
0	1	
1	1	

或门布尔表达式:Y=A+B

或非门

参考下图修改程序。



这个逻辑简称为"或非"门。运行程序进行测试,记录真值表。

NOR		
А	В	Y
0	0	
1	0	
0	1	
1	1	

"或非"门布尔表达式:

与非门

将上面程序中的"或"门改为"与"门。测试程序,记录真值表。

NAND		
А	В	Υ
0	0	
1	0	
0	1	
1	1	

Y = AB

"与非"门布尔表达式: