

第5章 创建子VI

子VI的作用

第5章 创建子VI

LabVIEW中的子**VI**（**SubVI**）类似于文本编程语言中的函数。一般来说，如果在**LabVIEW**中不使用子**VI**如同在编程语言中不使用函数一样，是不可能构建大的程序的。子**VI**是供其他**VI**使用的**VI**，与子程序类似。子**VI**是层次化和模块化**VI**的关键组件，它能使**VI**易于调试和维护。使用子**VI**是一种有效的编程技术，因为它允许在不同的场合重复使用相同的代码。**G**编程语言的分层特性就是在一个子**VI**中能够调用到另一个子**VI**。

第5章 创建子VI

通过构建和使用子VI能方便的实现**LabVIEW** 的层次化和模块化编程，把复杂的编程问题划分为多个简单的任务，使程序结构变得更加清晰、层次更加分明、程序更加易读、调试更加方便。

第5章 创建子VI

我们将常用的功能模块创建成子VI，不仅有效提高代码的使用效率，避免进行频繁的重复操作，也大大节省了编程时间。需要说明的是，一个子VI相当于一个子程序，子VI节点相当于子程序的调用语句，而不是子VI本身。子VI的控件和函数从调用该VI的程序（主程序）框图中接收数据，并将数据返回至该程序框图。

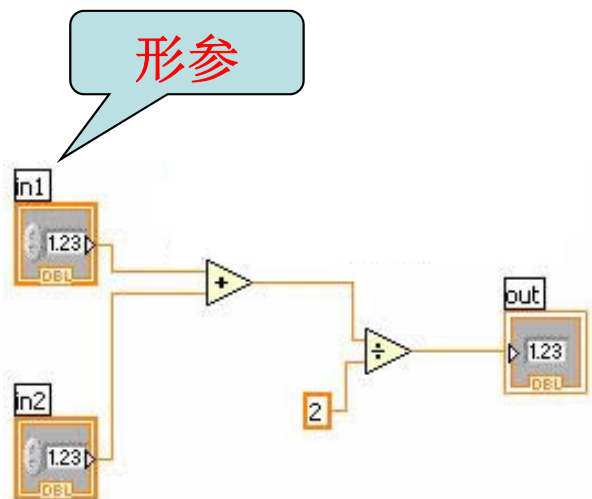
第5章 创建子VI

程序员用**LabVIEW**语言开发程序时，可以和**C**语言一样采用从顶向下的**结构化程序设计方法**，我们每创建一个**VI**程序，都可以将其作为上一级**VI**的子**VI**节点来调用，实现其模块化编程，这是使用**G**语言编程的分层特性。一个子**VI**内可以调用多个子**VI**。本章主要介绍**VI**的两种创建方法以及**VI**的调用、添加至库的方法。

第5章 创建子VI

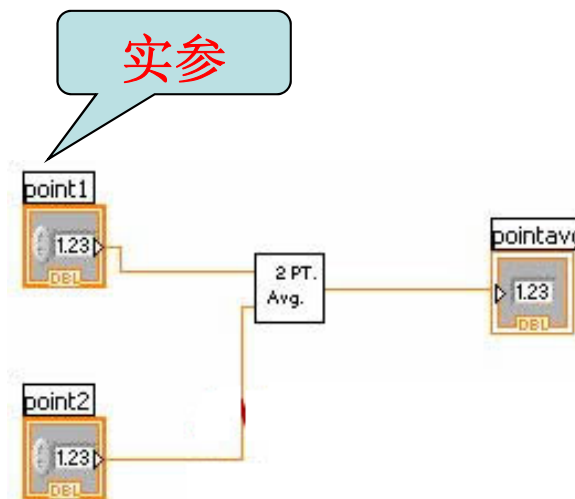
子程序代码

```
function average (in1,in2,out)
{
  out=(in1+in2)/2.0;
}
```



调用子程序

```
main
{
  average (point1,point2,pointavg);
}
```



子VI调用示意图

5.1 图标的创建和编辑

在调用**VI**的主程序框图中，用**图标**来代表子**VI**。另外，子**VI**必须有一个正确连接端子的**连接器**来实现和它上层**VI**的数据交换。

LabVIEW为每个程序创建默认的图标，显示在前面板和程序框图窗口的右上角。原始图标为**NI**的默认程序图标。

对默认图标的编辑可通过图标编辑器来完成。

5.1 图标的创建和编辑

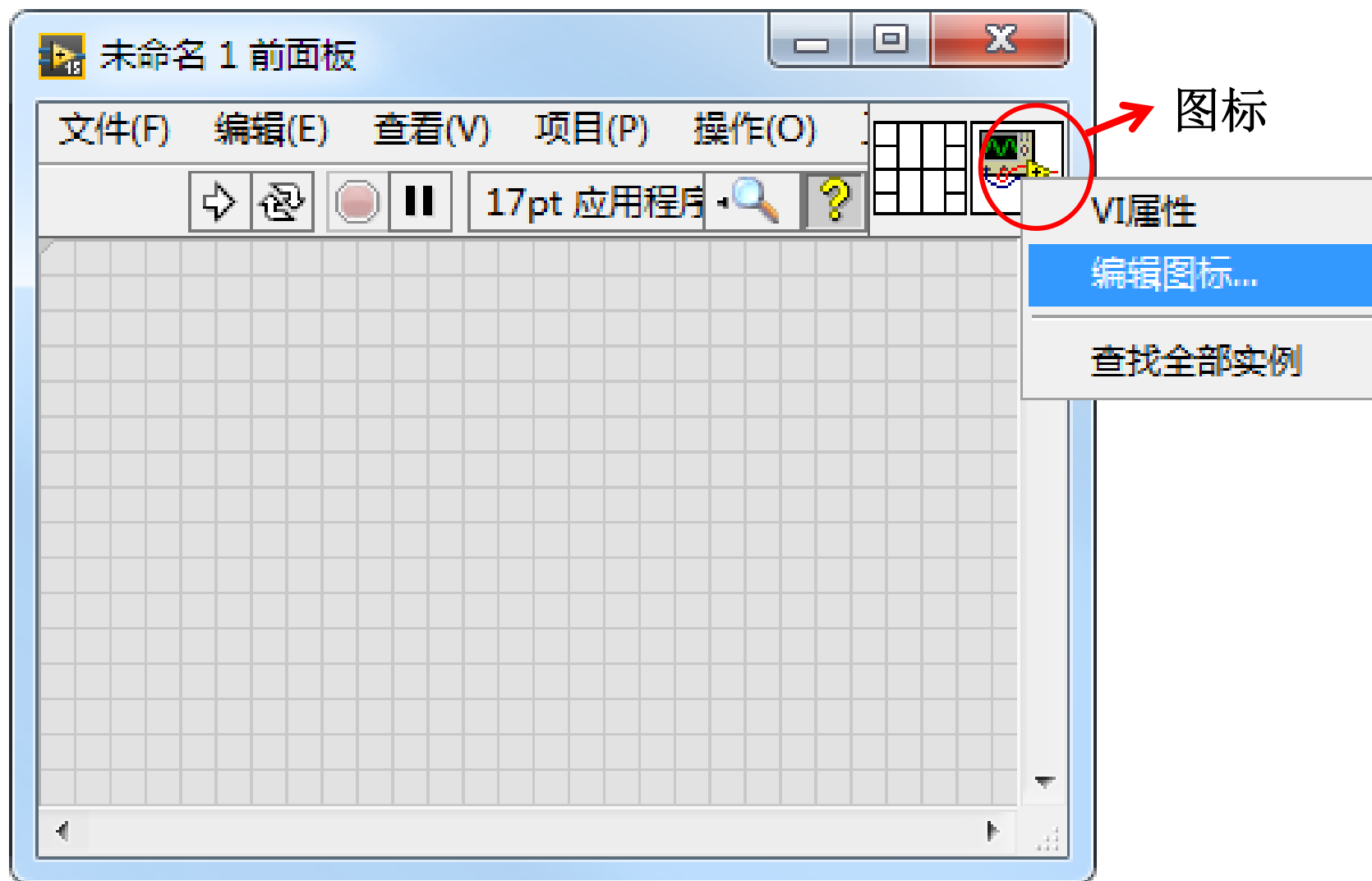


图5-1 默认图标

5.1 图标的创建和编辑

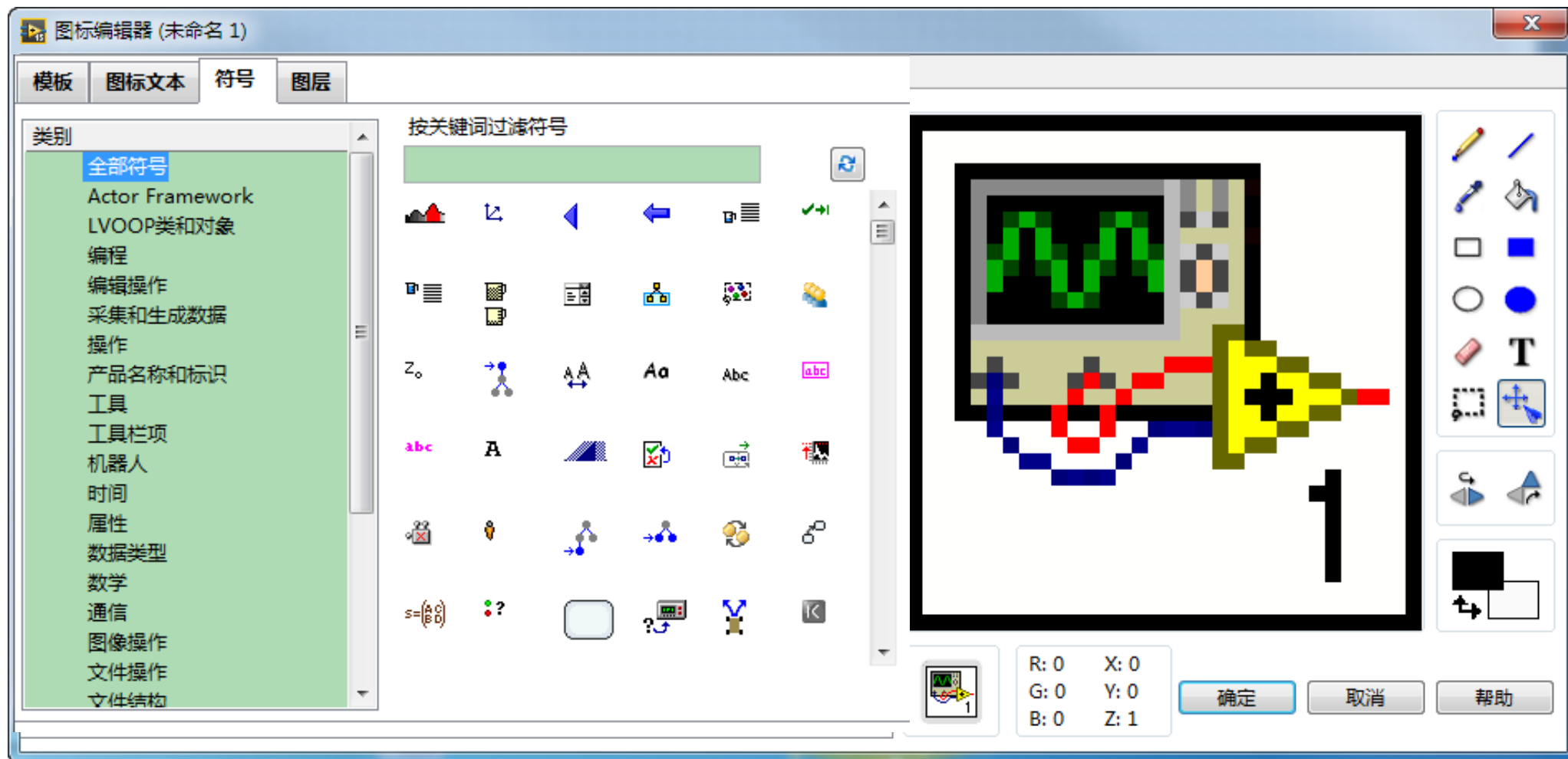
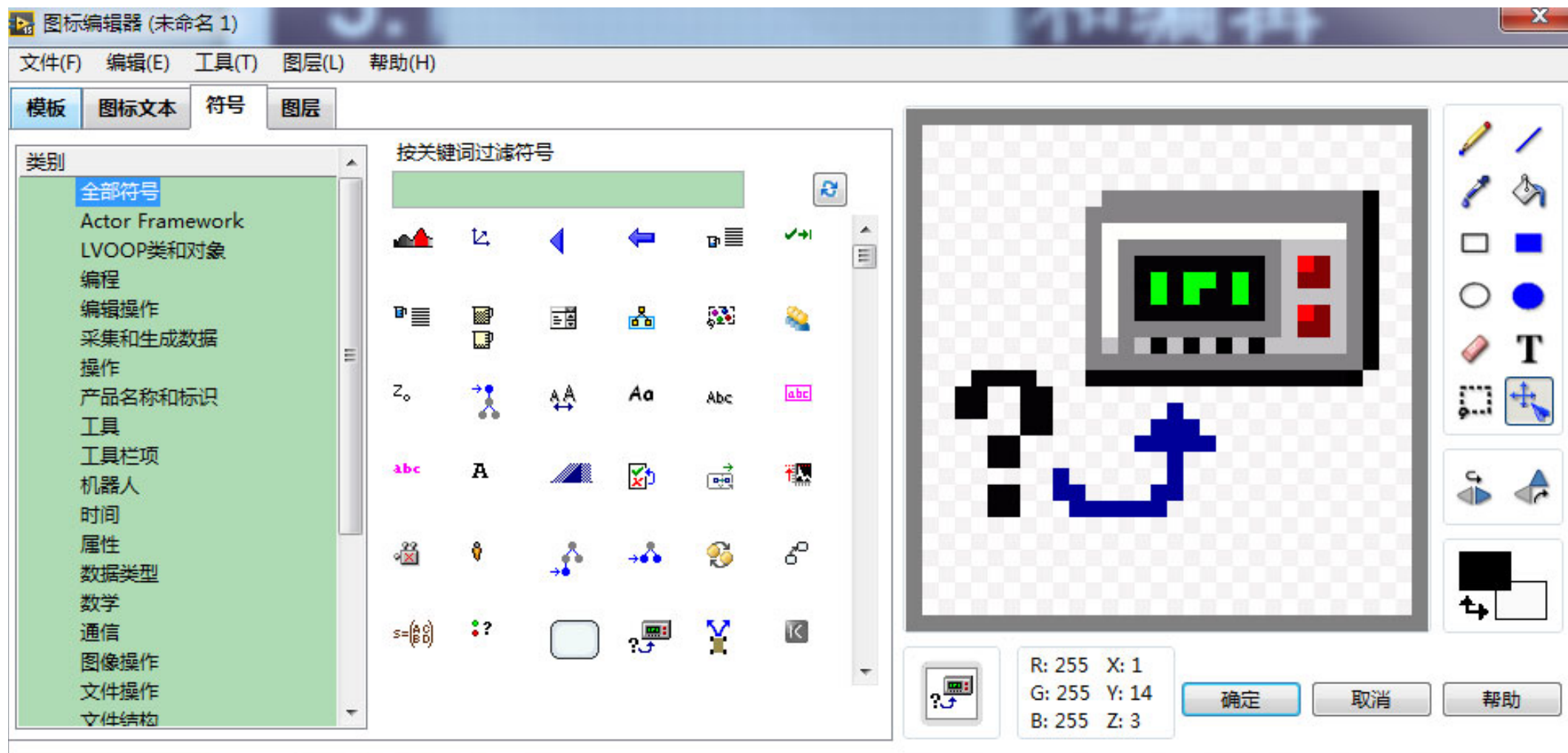


图5-2 图标编辑器窗口

5.1 图标的创建和编辑



如果要在图标上写英文，最适合的字体是“**Small Fonts**”，字号8。

5.2 连接器端口的设置

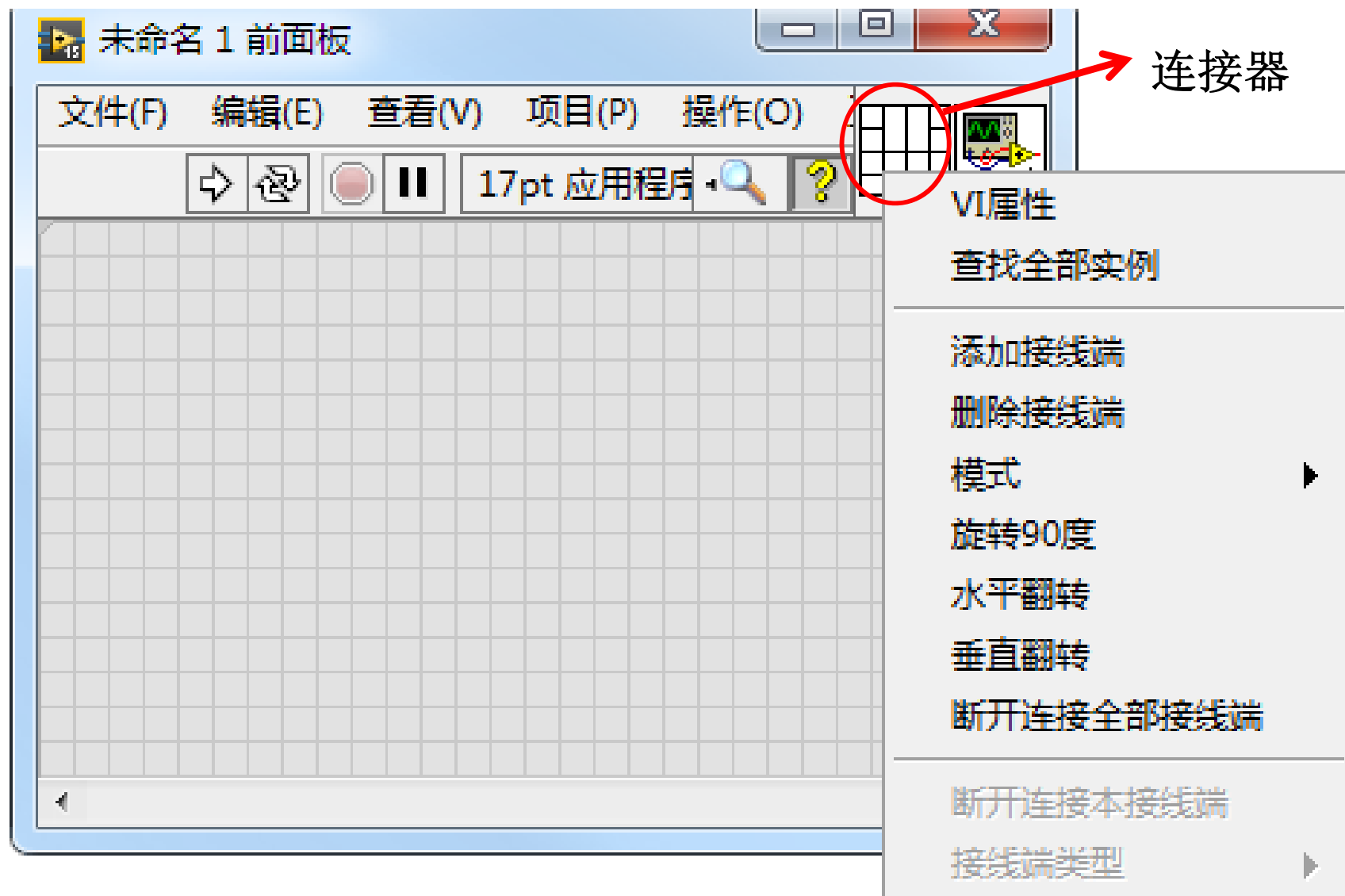


图5-4 从图标窗口到连接器的切换

5.2 连接器端口的设置

端口连接器作为一个**编程接口**，为子VI定义输入、输出端口数和这些端口的接线端**数据类型**。这些输入输出端口相当于编程语言中的**形式参数**和**结果返回变量**语句。

5.2 连接器端口的设置

当调用的子VI程序运行时，子VI 输入端口接收从外部控件或其他对象传输到子VI各端口的数据，经子VI内部处理后再从子VI输出端口向主VI输出结果数据，或传送给子VI外部显示控件。

5.2 连接器端口的设置

一般情况下，**VI**只有设置了连接器端口才能作为子**VI**使用，如果不对其进行设置，则调用的只是一个独立的**VI**程序，不能改变其输入参数也不能显示或传输其运行结果。

（类似于**VB**程序中函数和过程的区别）

5.2 连接器端口的设置

如果需要对子VI节点进行输入输出，那么就需要在连接器面板中有相应的连线端子。我们可以通过选择VI的端子数并为每个端子指定对应的前面板控件或指示器来定义连接器。

5.2 连接器端口的设置

连接器的设置分两个步骤：

- (1) 创建连接器端口，包括定义端口的数目和排列形式；
- (2) 定义连接器端口和控件及指示器的关联关系，包括建立连接和定义接线端数据类型。

5.2 连接器端口的设置

如图5-5所示，快捷菜单的模式选项板提供了**36种预定义的连接端口布局模式**。我们可以选择所需的端口布局模式。



图5-5 选择连接器的模式定义端口

5.2 连接器端口的设置

- 如果定义的端口数超过所需端口数，我们可以不删除这些多余的端口。可以按需要选择端口和相应的控件或指示器建立关联，而对其他端口则不进行关联操作来使多余的端口闲置。
- 一个端口只能关联一个变量，如要关联多个变量，就必须建立一个“簇”变量，再和端口关联。端口关联后，需删除关联关系后，才能更换关联变量。
- 关联时，鼠标处于“连线状态”。

5.2 连接器端口的设置

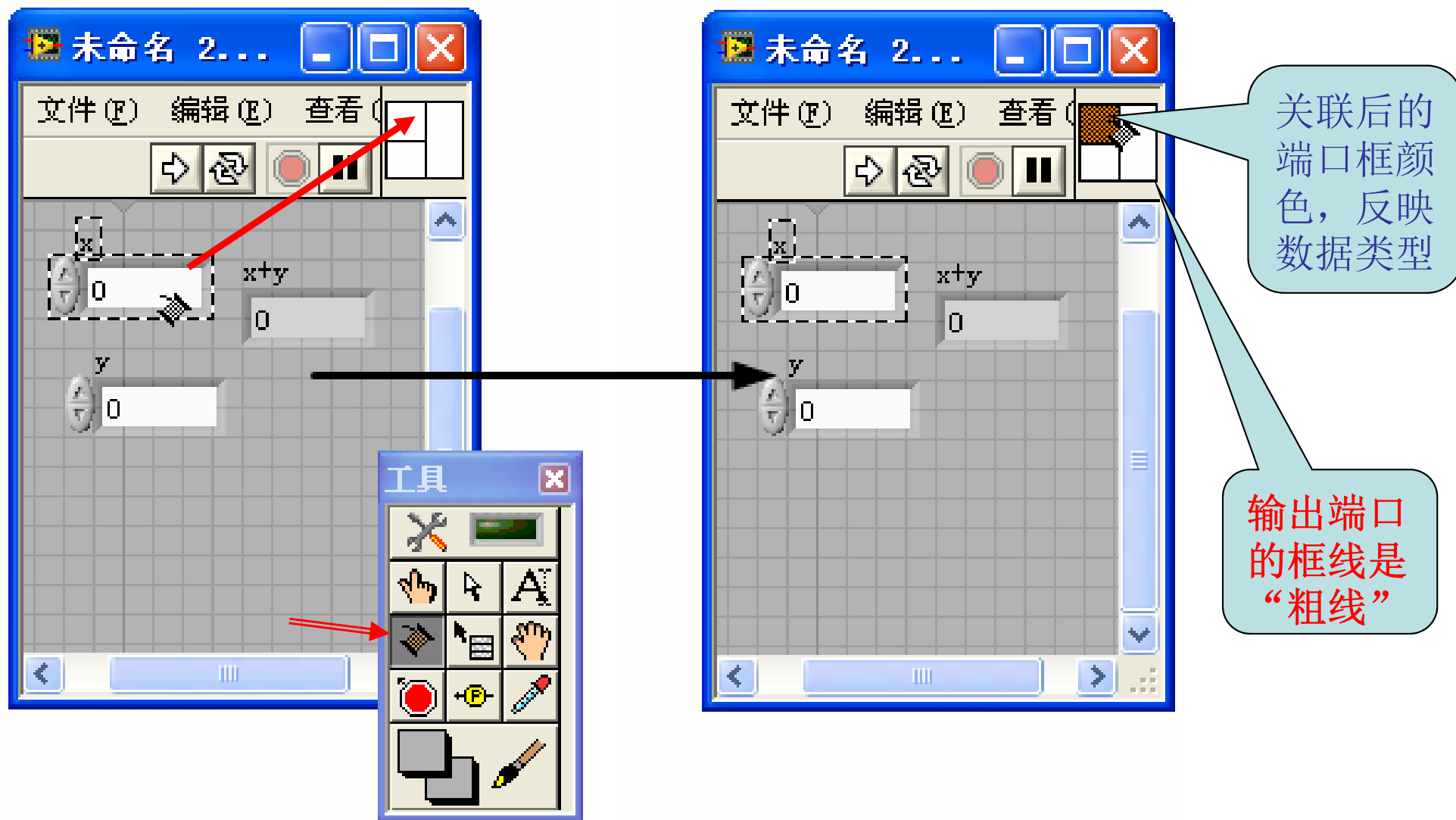
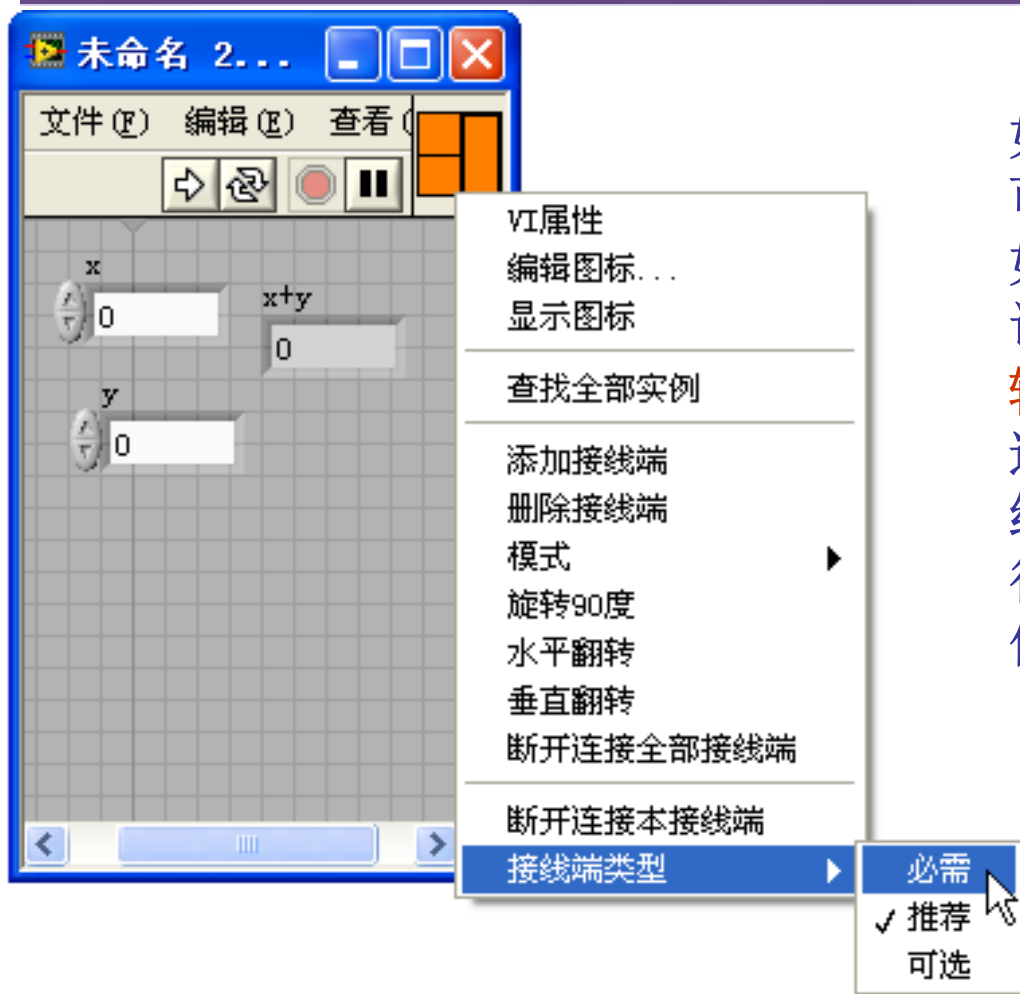


图5-6 建立连接器关联关系

5.2 连接器端口的设置



如某个输入或输出对VI的正常运行是必不可少的，该接线端应设置为“必需”。如在调用子VI时，未连接“必需”输入端，该子VI所在的程序框图将出现断线。

输出接线端不存在“必需”选项。如未连接“推荐”或“可选”输入或输出接线端，该子VI所在的程序框图仍然可以执行。这些接线端未连线时，VI不会产生任何警告。

图5-7 设置端口连接类型

5.3 子VI的两种创建方法

LabVIEW中子VI的创建有两种方法：一种方法是用**现有VI创建子VI**，另一种方法是**选定内容创建成子VI**。前一种方法把整个框图所示的程序创建成子VI是层次化编程的基础，后一种方法选定程序的一部分创建成子VI，相对前一种方法比较灵活机动。选定部分被子VI节点所取代实现程序的模块化编程并增加程序可读性。

5.3.1 现有VI创建成子VI

将VI创建成子VI，关键是连接器的定义。

下面就编写一个求两数较大值的程序，并将此VI创建成子VI。

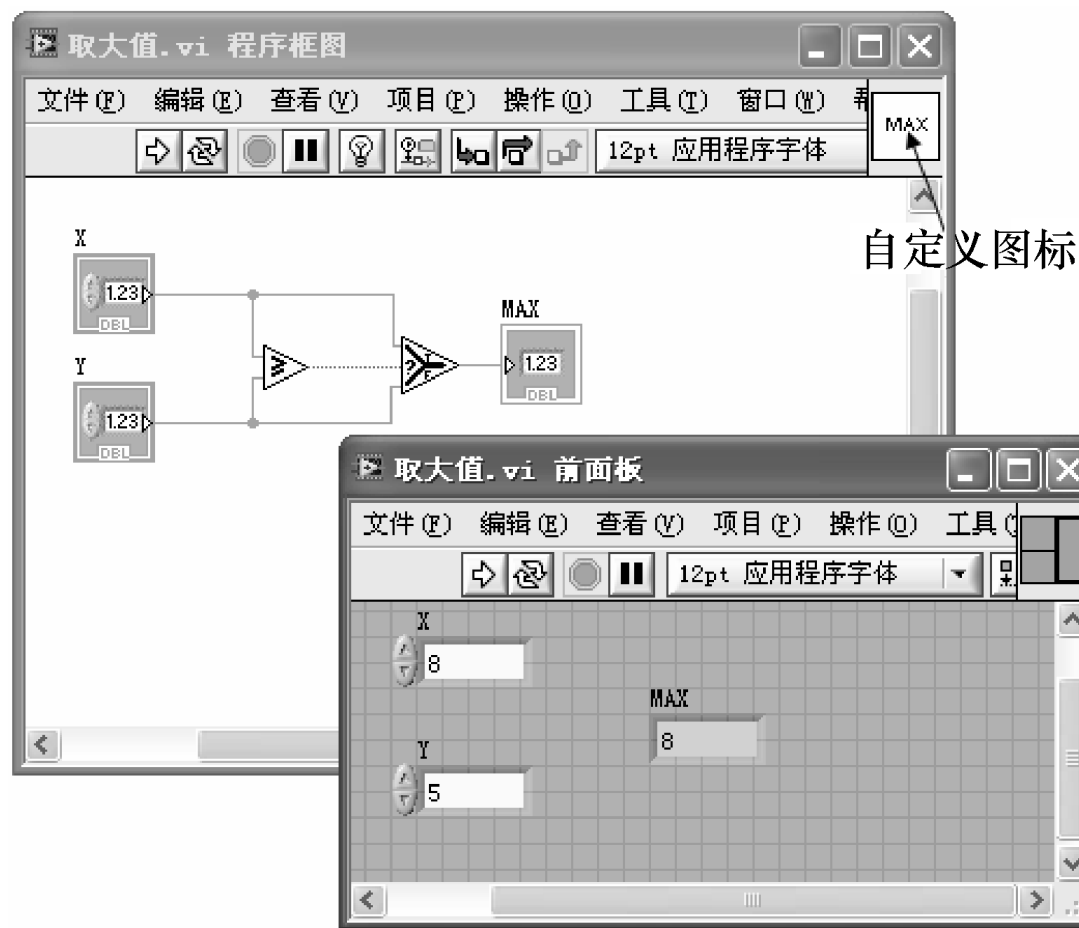


图5-8 创建VI求两数较大值

5.3.1 现有VI创建成子VI

- (1) 创建一个如图5-8所示的VI，此VI用来求两个数中的较大值。
- (2) 在程序框图的空白处单击鼠标右键，打开函数选板，从“比较”子选板中找到“大于等于？”函数和“选择”函数，完成如图5-8所示连接。
- (3) 打开图标编辑器，为了显示此VI的功能，我们可以编辑如图5-8所示的图标。
- (4) 切换到前面板，按前一节所示的方法选择连接器端口。
- (5) 在前面板的文件菜单项中单击保存选项或另存为选项保存此VI。

5.3.1 现有VI创建成子VI

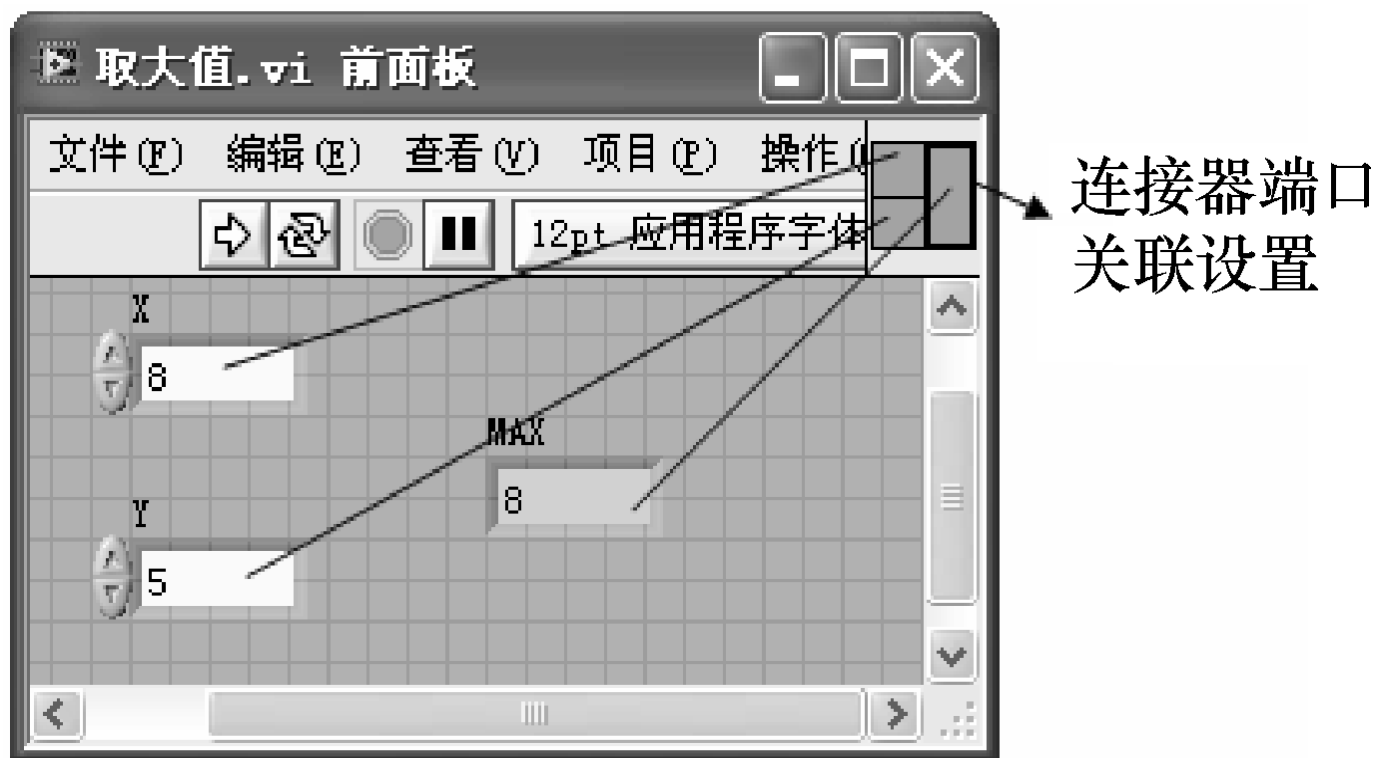


图5-9 选接器的设置

5.3.2 选定内容创建成子VI

当在设计程序的过程中需要**模块化某段程序**以使程序结构清晰或方便以后调用，我们也可以使用选定内容创建成子**VI**的方法。

下面就在**LabVIEW**中**NI**范例的基础上修改一个程序，并选定此**VI**的部分内容创建成子**VI**。步骤如下：

(1) 打开位于启动菜单右下角的“查找范例”选项，在弹出的“**NI**范例查找器”中双击**Basic Amplitude Measurements.vi**。

5.3.2 选定内容创建成子VI

(2) 对如图5-10所示程序进行修改，修改后如图5-11所示。

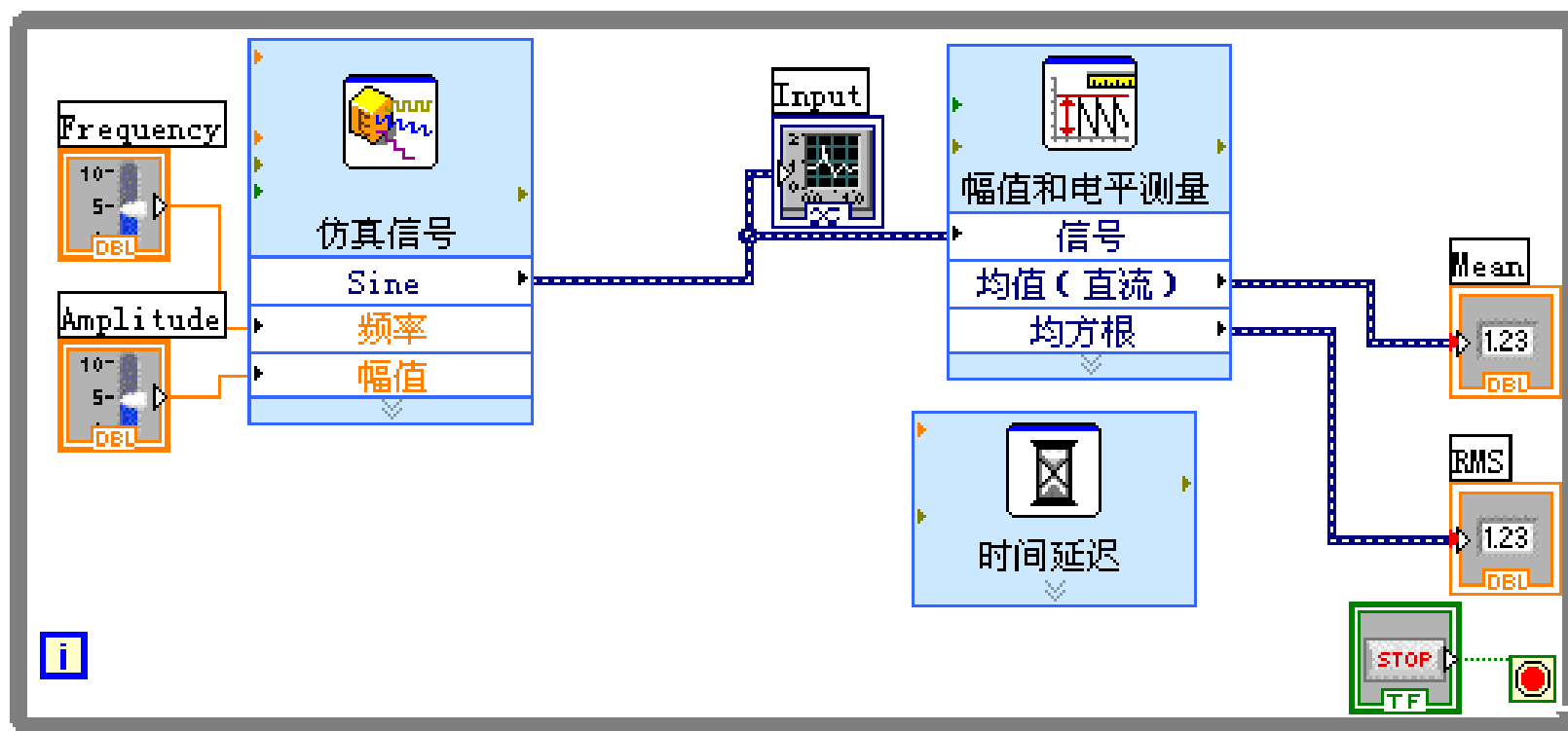


图5-10 正弦信号的测量

5.3.2 选定内容创建成子VI

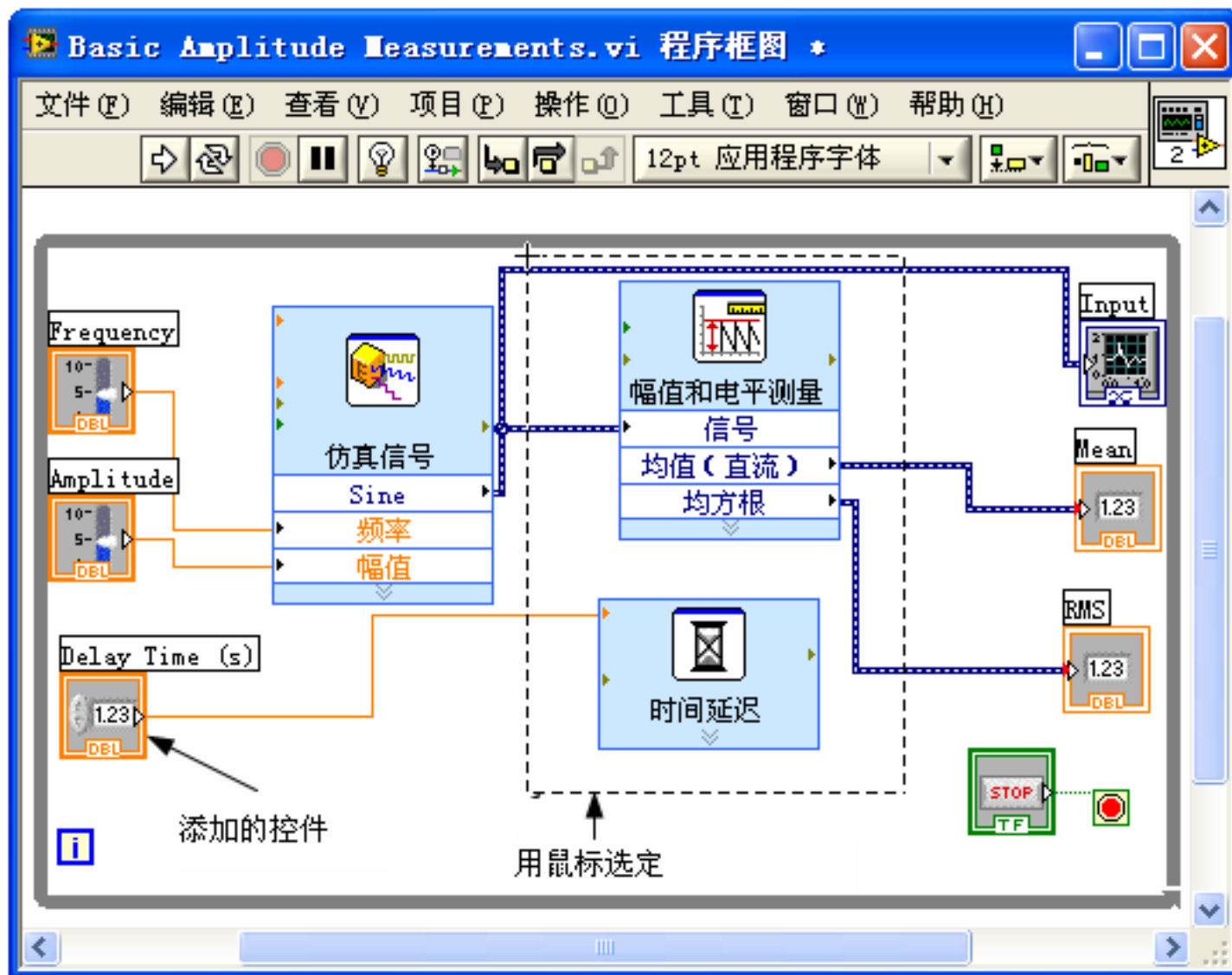


图5-11 框定要创建成子VI的程序内容

5.3.2 选定内容创建成子VI

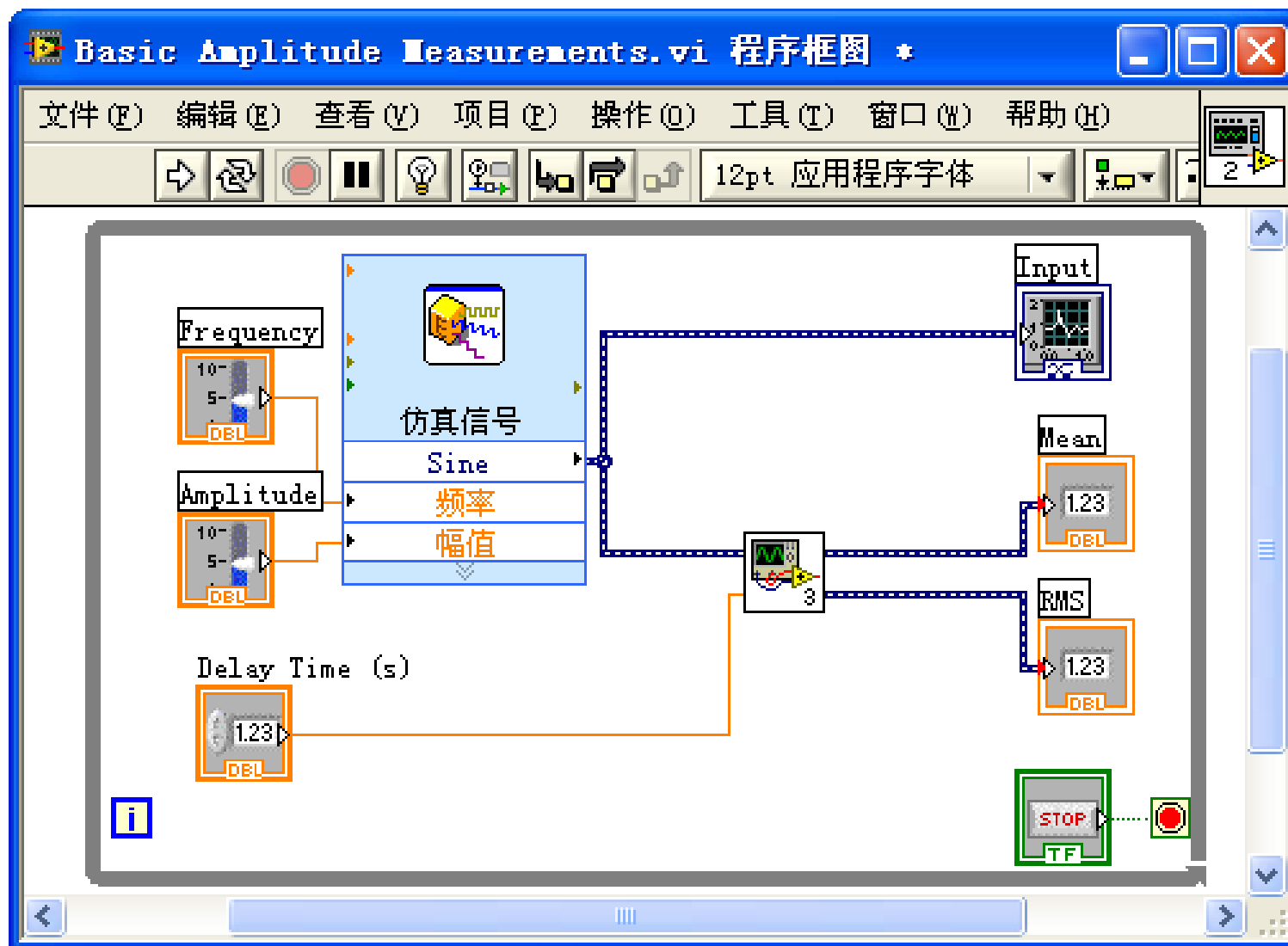
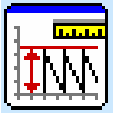


图5-12 框选的程序被子VI图标取代

5.3.2 选定内容创建成子VI

(3) 要完成此子VI，首先要选定程序框图中的相应部分将其创建子VI，然后再对此子VI的图标和连接器进行修改。

(4) 对默认图标进行编辑，首先用截图软件截取幅值和电平测量节点图案, 保存为**BMP**图片格式。

5.3.2 选定内容创建成子VI

- (5) 在图标窗口单击鼠标右键，在快捷菜单选择显示连线板选项，此时子VI默认的连线端类型为如图5-14所示状态。
- (6) 对子VI重命名并保存此子VI。
- (7) 此时，不能对”端口接线板”的形状和关联做修改，否则会导致原Vi程序的数据连线错误。

5.3.2 选定内容创建成子VI

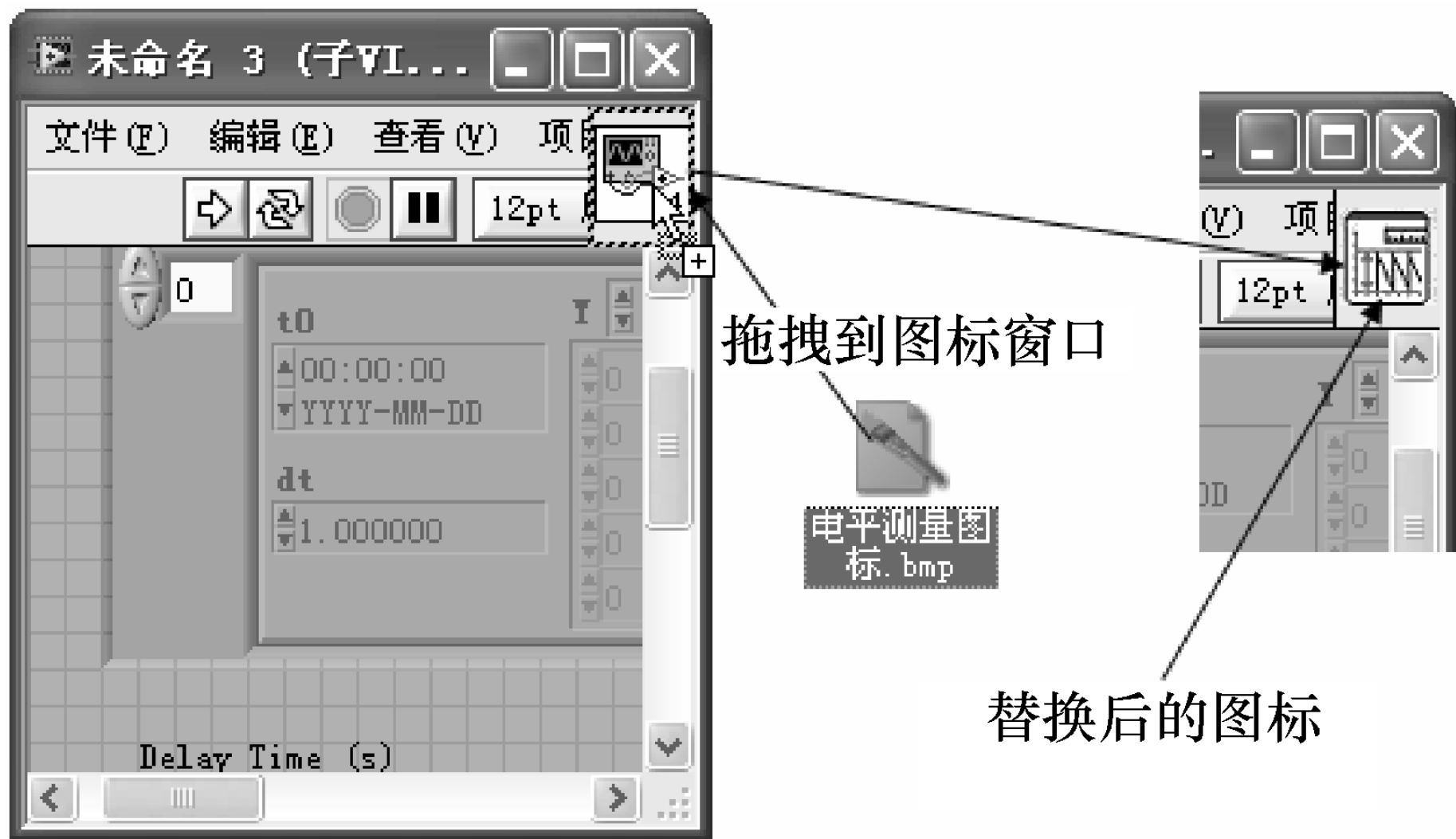


图5-13 对图标进行编辑替换

5.3.2 选定内容创建成子VI

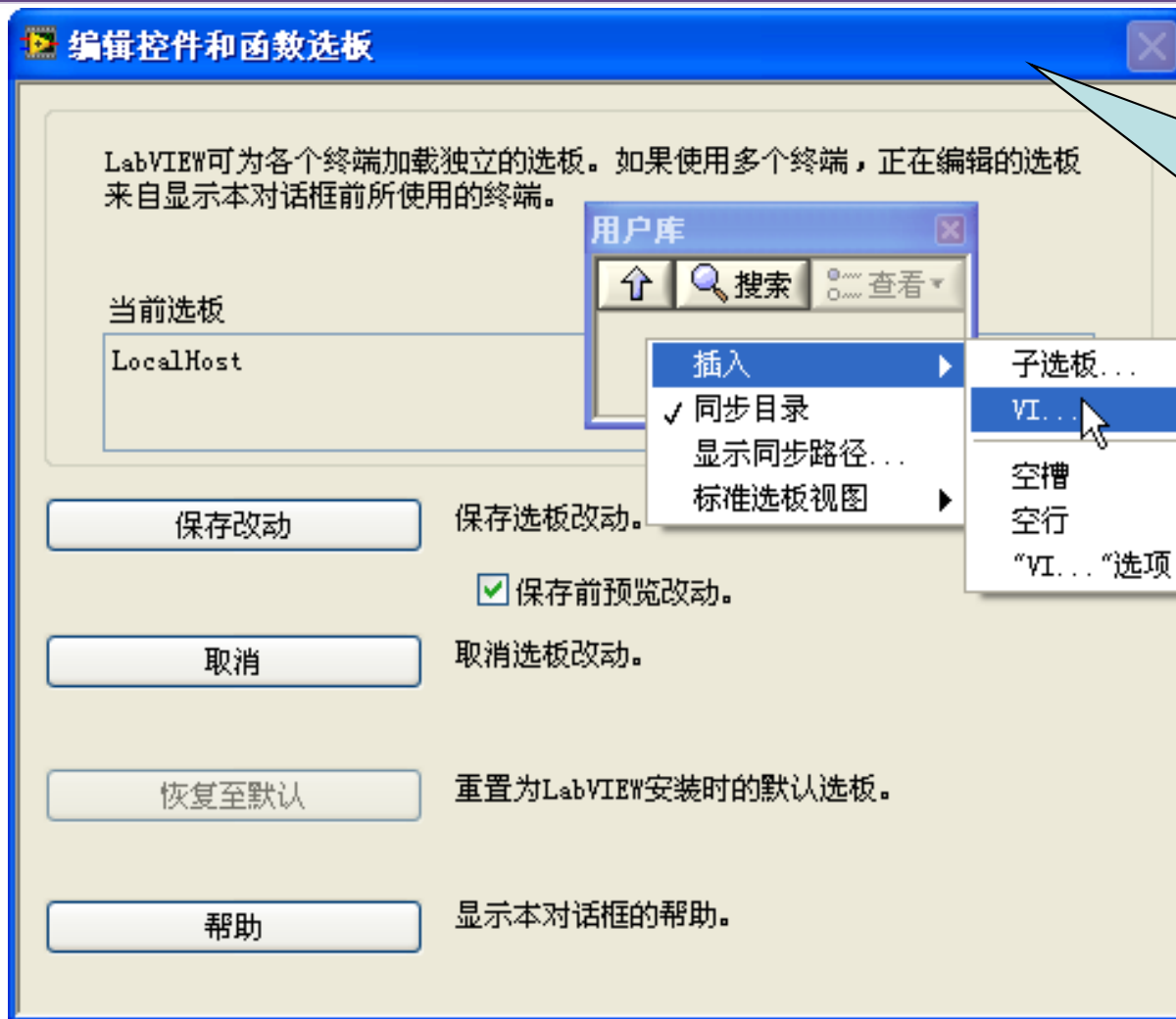


图5-14 对连接器默认值进行修改

5.4 添加子VI至用户库

如果创建的子VI被使用的频率较高，为方便调用，我们可以把子VI添加进函数选板的“用户库”中。调用时只需从函数选板的“用户库”中找到所需要的子VI，拖动此子VI至程序框图即可完成调用。

5.4 添加子VI至用户库



在菜单的“工具”选项下，选“高级”，再选“编辑选板”，打开“选板编辑器”

图5-15 在用户库中插入子VI

5.4 添加子VI至用户库



图5-16 用户库面板子VI图标

5.5 子VI的调用

除了把子VI创建到用户库，从用户库调用子VI，把子VI拖动到程序框图的方法外，我们还可以在主VI程序框图中通过“函数”选板上的“选择VI”子选板来打开子VI，实现调用。选择“函数”选板中的“选择VI”子选板后会弹出一个对话框，在对话框中选择需要调用的子VI，如图5-17所示。

5.5 子VI的调用



图5-17 选择需要调用的子VI

5.5 子VI的调用----实例

“两数取大”

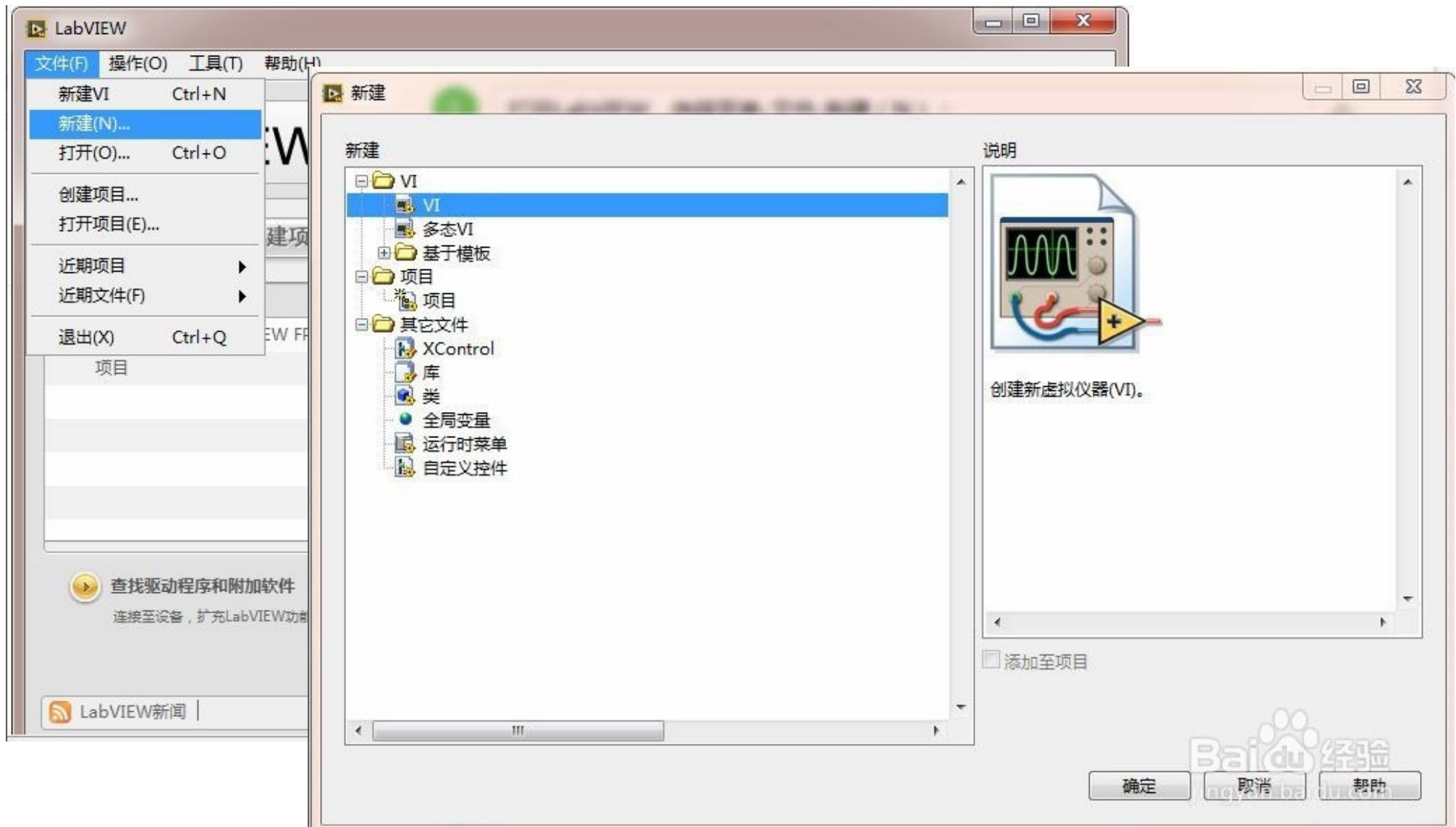
“LABVIEW范例”

“数组处理” 示例

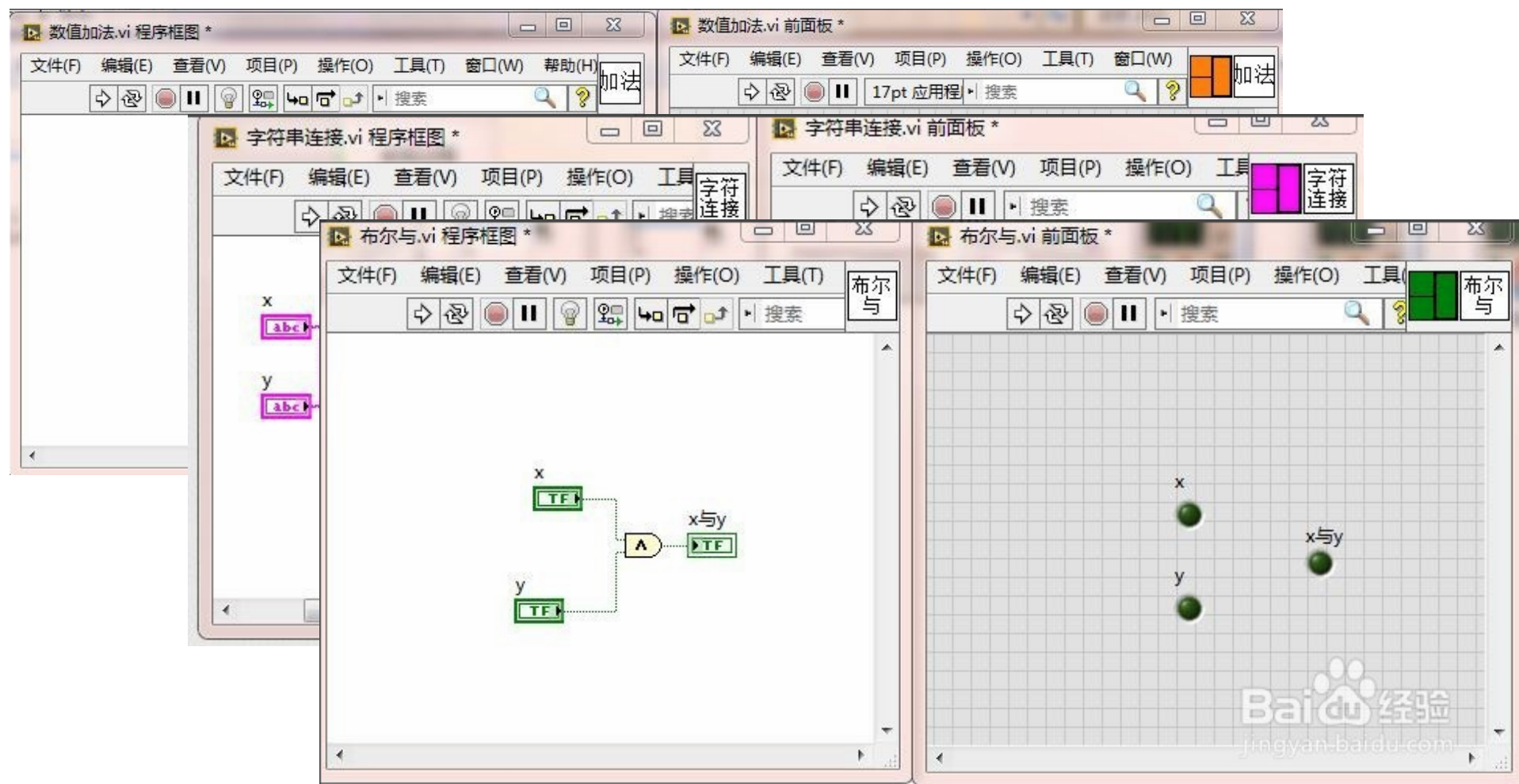
“数组处理” 示例

“显示前面板” 示例

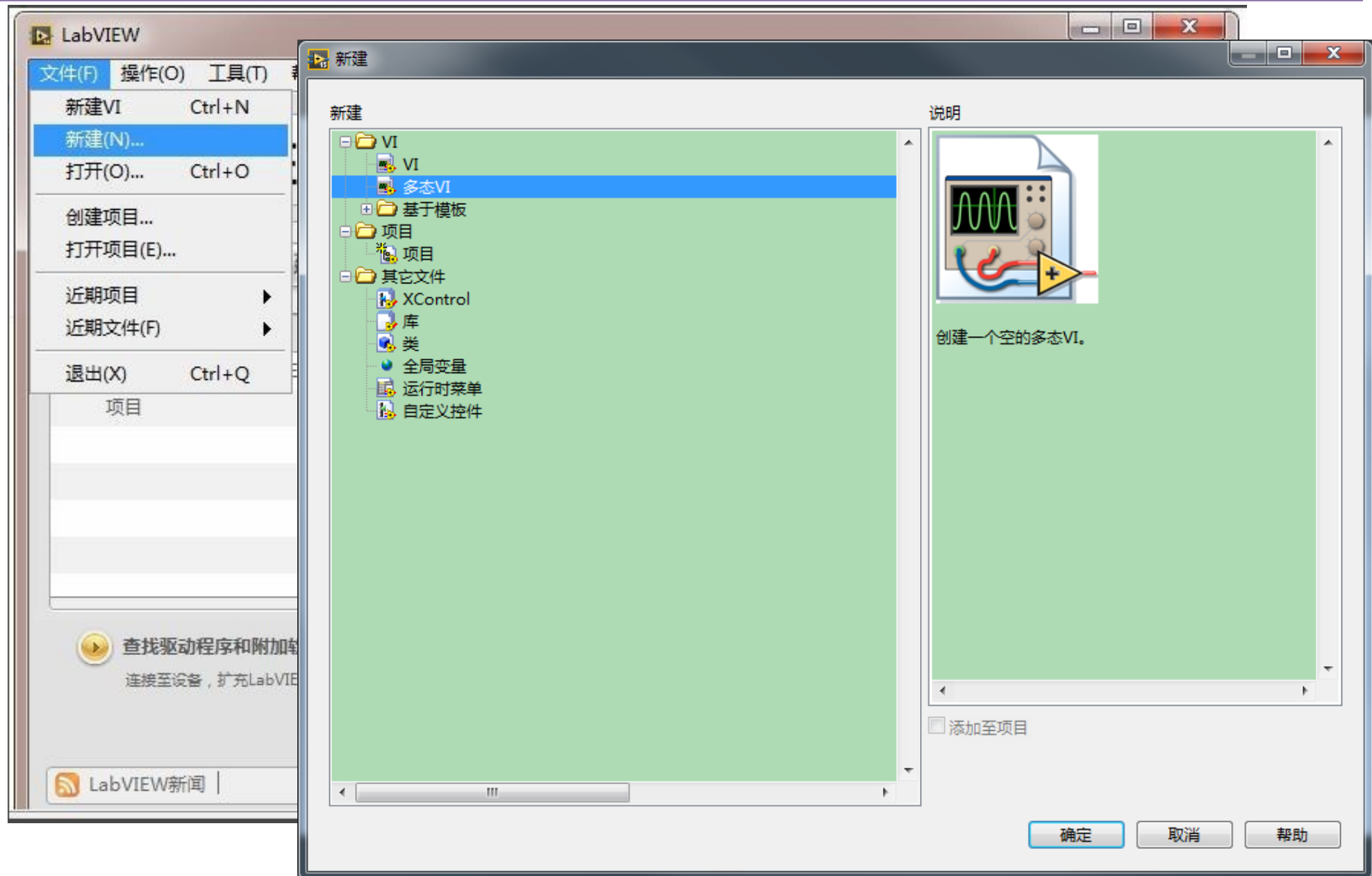
5.6 多态子VI



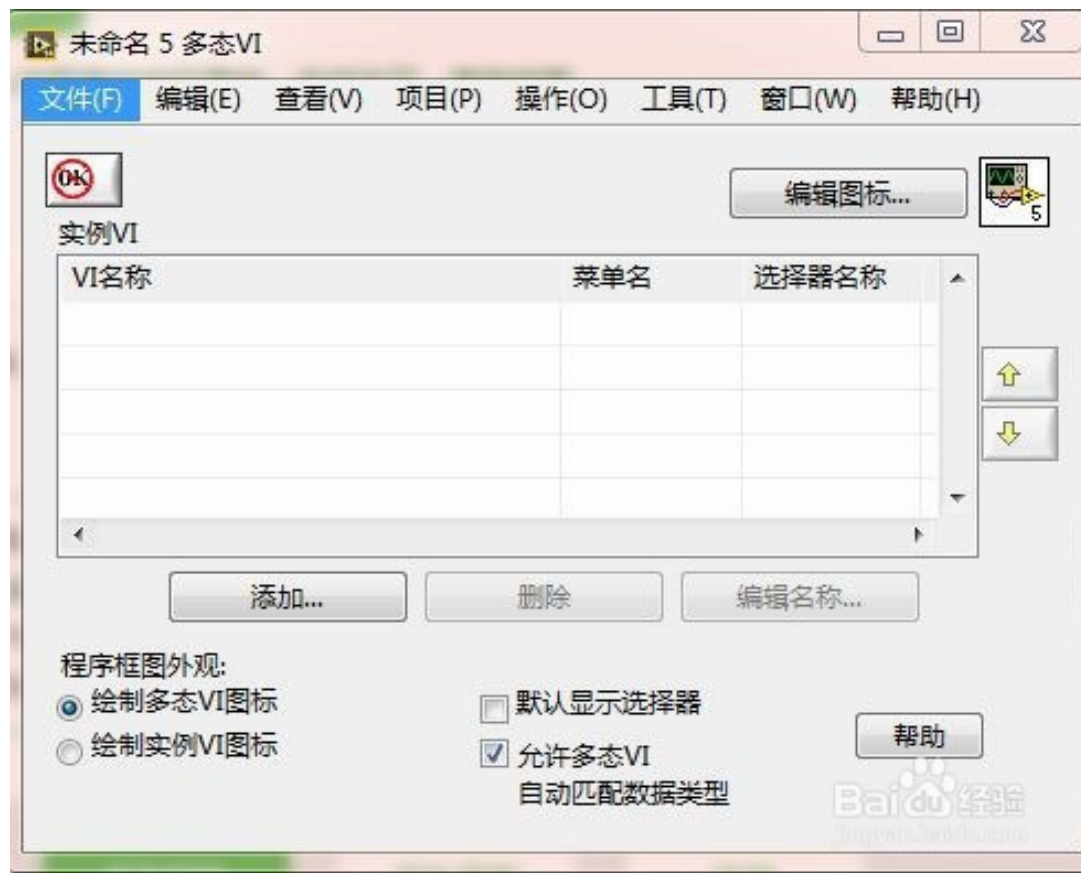
5.6 多态子VI



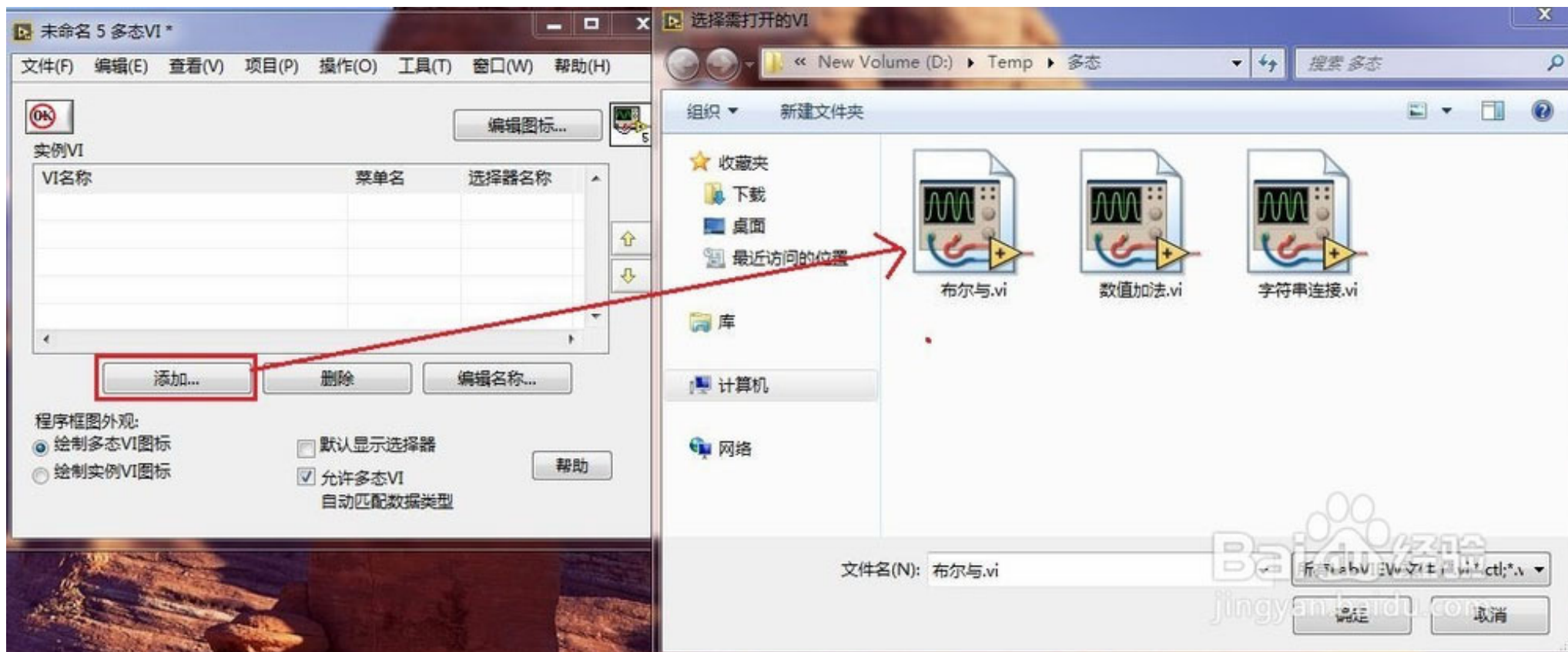
5.6 多态子VI



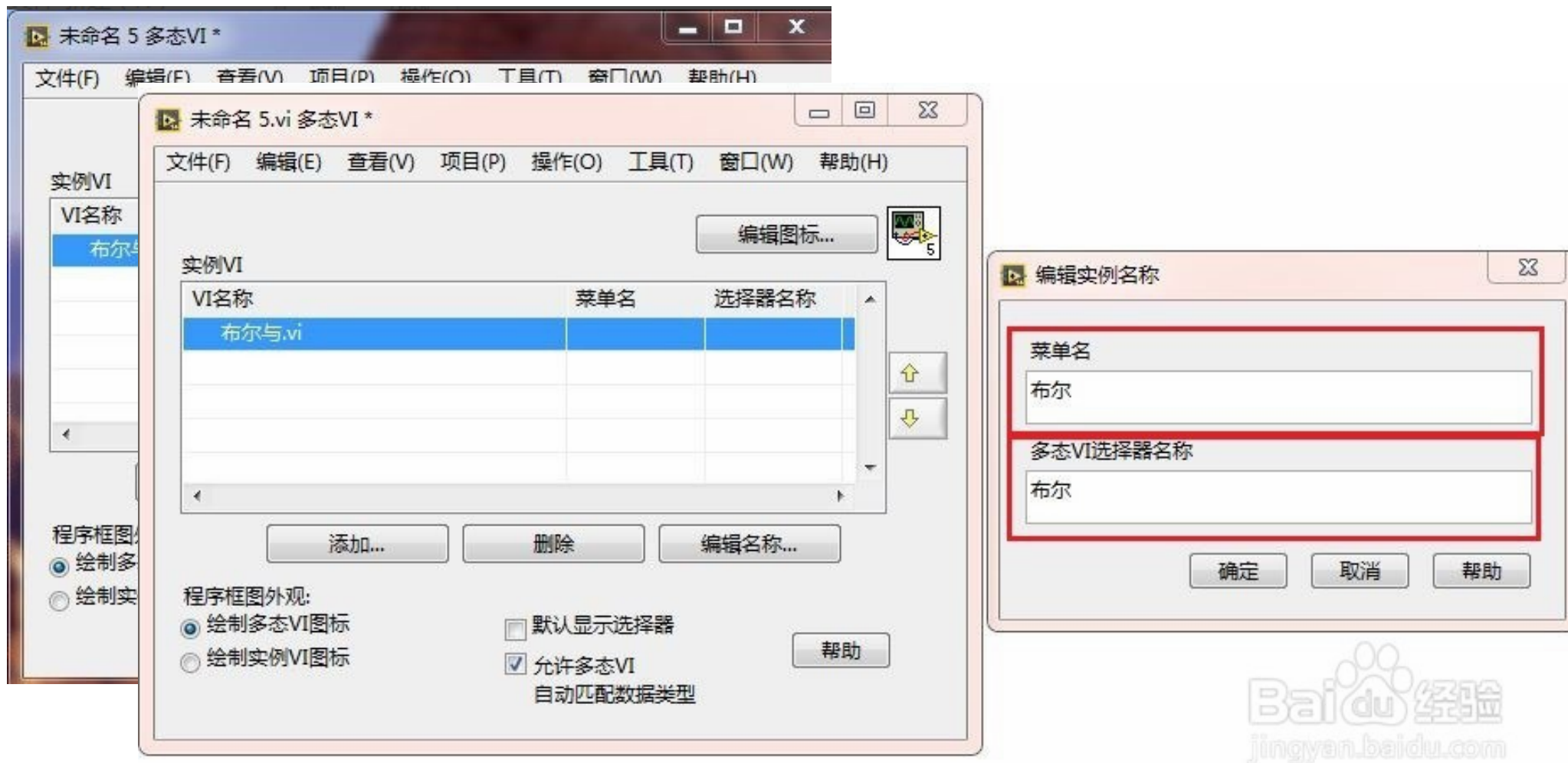
5.6 多态子VI



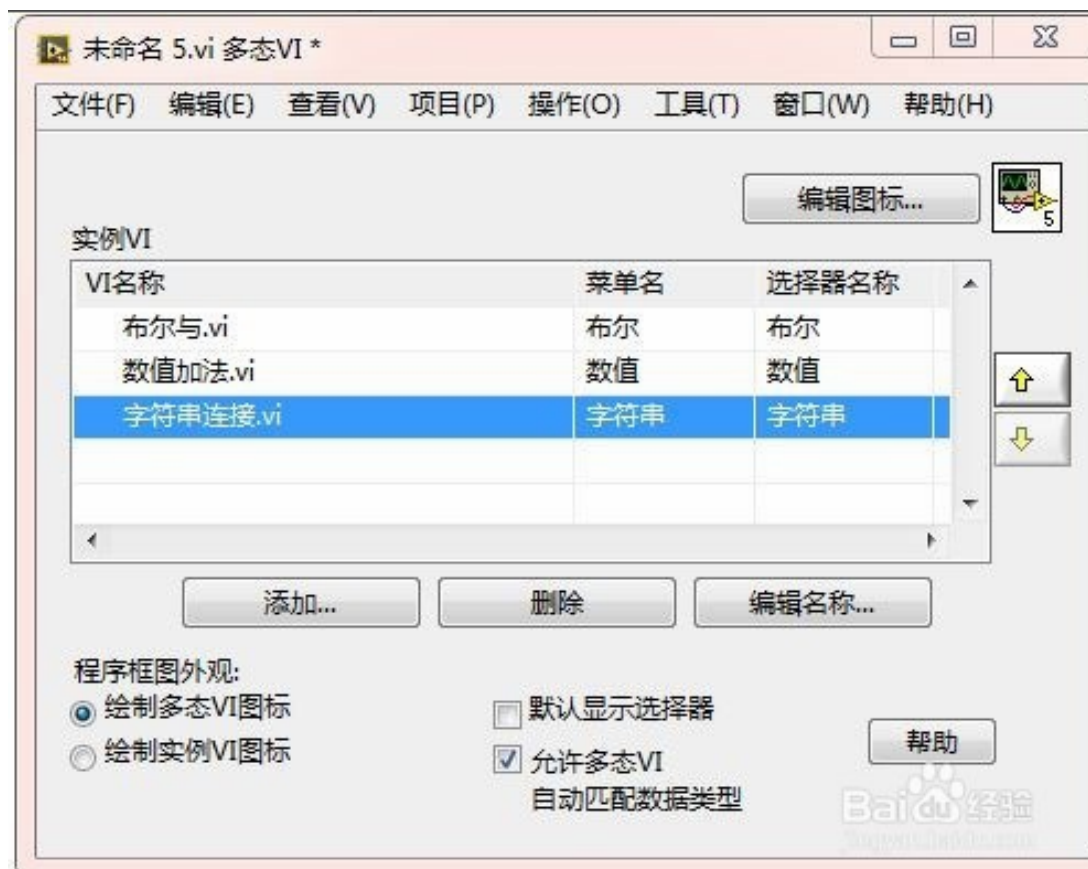
5.6 多态子VI



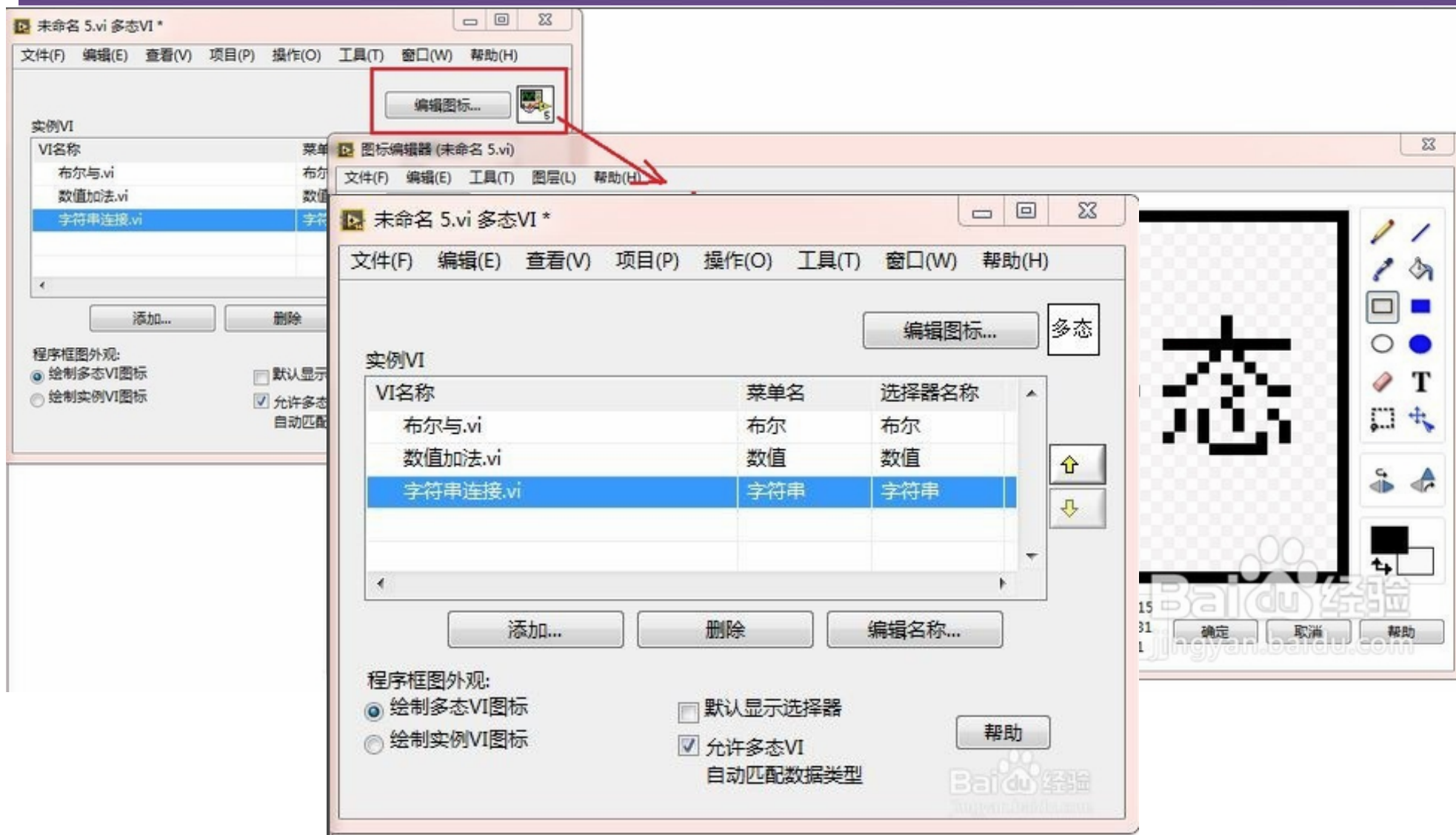
5.5 多态子VI



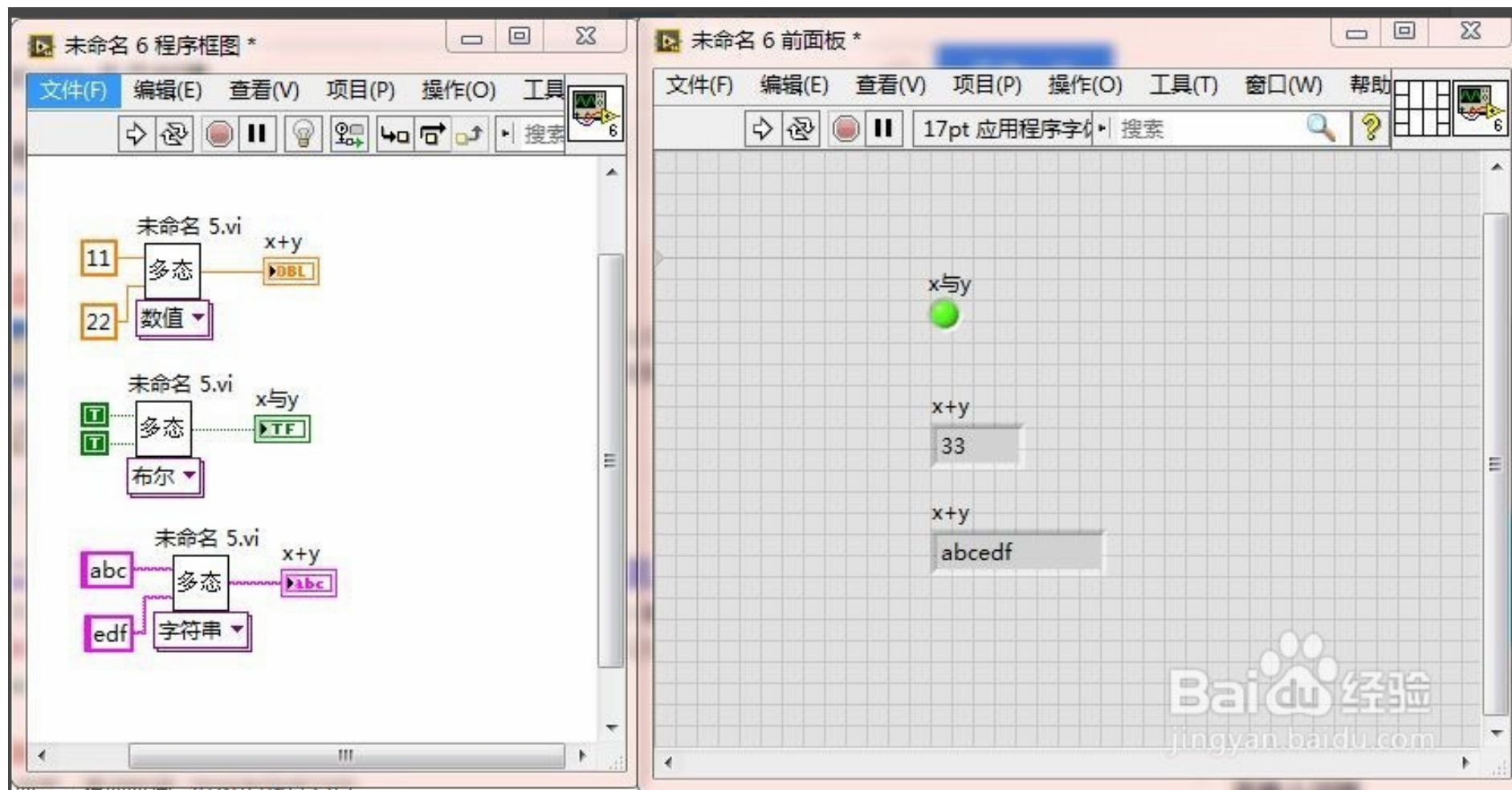
5.6 多态子VI



5.6 多态子VI



5.6 多态子VI



5.7 VI的层次结构

在编写复杂的程序时，不应把所有功能都放在一个框图中。我们应养成习惯，在框图中编写代码前先进行规划，预先确定结构层次以简化框图节省编程时间，并有利于调试代码。通常我们用自顶向下的编程方法，这样的程序结构很像倒立的树。

5.7 VI的层次结构

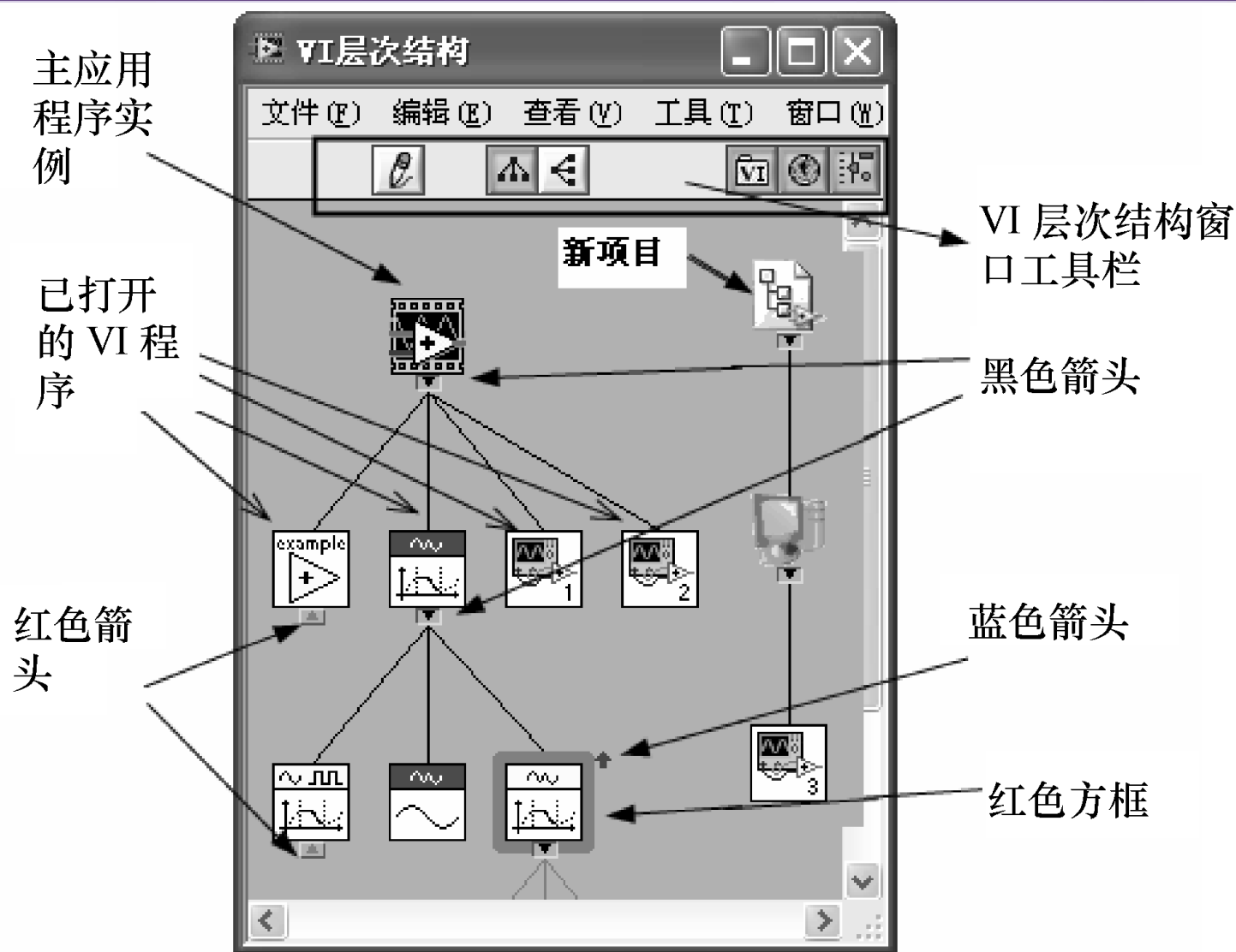








图5-18 VI层次结构窗口

5.7 VI的层次结构

表5-1 VI层次窗口工具条功能说明

图 标	名 称	功 能 说 明
	重做布局	重新安排层次结构节点的位置，使连线交叉情况减少，连线缩短。使层次结构更加清晰
	垂直布局	使层次结构窗口中的节点从上到下排列，根节点置于顶部，各子节点位于上一级节点下方位置
	水平布局	使层次结构窗口中的节点从左到右排列，根节点置于左边，各子节点位于上一级节点右边位置
	包括VI库	在层次结构窗口中显示/隐藏被VI使用的VI库中的程序
	包括全局变量	在层次结构窗口中显示/隐藏被VI使用的全局变量
	包括自定义类型	在层次结构窗口中显示/隐藏被VI使用的自定义类型

5.7 VI的层次结构

所有有子**VI**的节点都有一个小的箭头，用于显示或隐藏其下属的子**VI**。

在**VI**层次结构窗口中打开编辑菜单的查找选项，并在查找对话框中输入要搜索的词，我们就可以搜索指定节点，与输入字符相匹配的节点会在搜索结果框中显示，如图**5-19**所示。

5.7 VI的层次结构

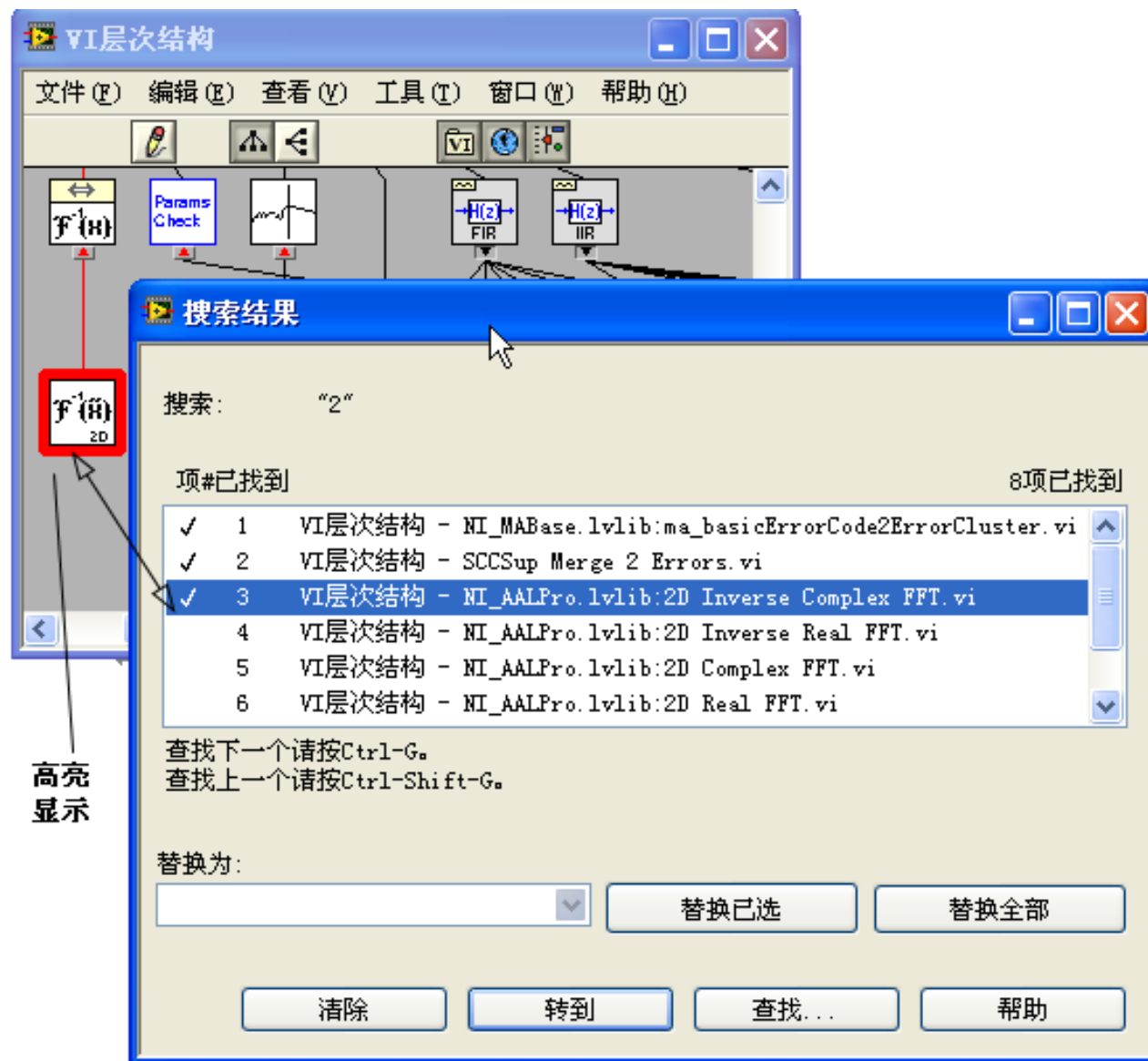


图5-19 搜索层次

5.7 VI的层次结构

在**VI**层次结构窗口直接从键盘输入搜索名也可以找到所需节点，但前提是这些节点在**VI**层次结构窗口内都是可见的。

本章小结

本章主要介绍了如何构建子VI。子VI类似于其他文本编程语言中的子程序，它是**LabVIEW**程序设计的基础。充分利用好子VI，能够使程序框图更加简化，并且程序也易于调试和维护。

掌握如何建立和使用子VI是成功构建**LabVIEW**程序的关键之一。

本章小结

本章首先介绍了如何编辑VI的图标和如何进行连接器端口的设置，为了区别不同的子VI所实现的不同功能，有时需要为不同的子VI创建不同的图标以便于理解；然后详细介绍了两种创建子VI的方法（现有VI创建成子VI和选定内容创建成子VI）并结合具体实例进行说明。最后介绍了如何将一个子VI添加至我们库，如何调用子VI以及VI的层次结构。