FSE598 前沿计算技术

模块 1　计算思维
单元 2　工作流与可视化编程
第 3 讲　用VIPLE 设计 ALU

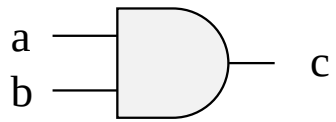
本讲座的英文版内容基于教材：
Y. Chen, G. De Luca Service-Oriented Computing and System Integration: Software, IoT, Big Data, and AI as Services, 8th edition, Kendall Hunt Publishing, 2022.
https://www.public.asu.edu/~ychen10/book/socsi.html

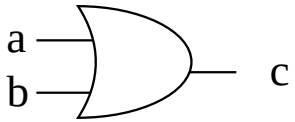
本讲大纲

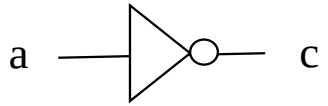
学习

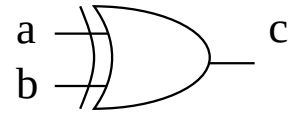
- ALU 设计与实现
- 自动化测试用例生成和测试
- VIPLE 服务生成和应用
- Web 服务调用

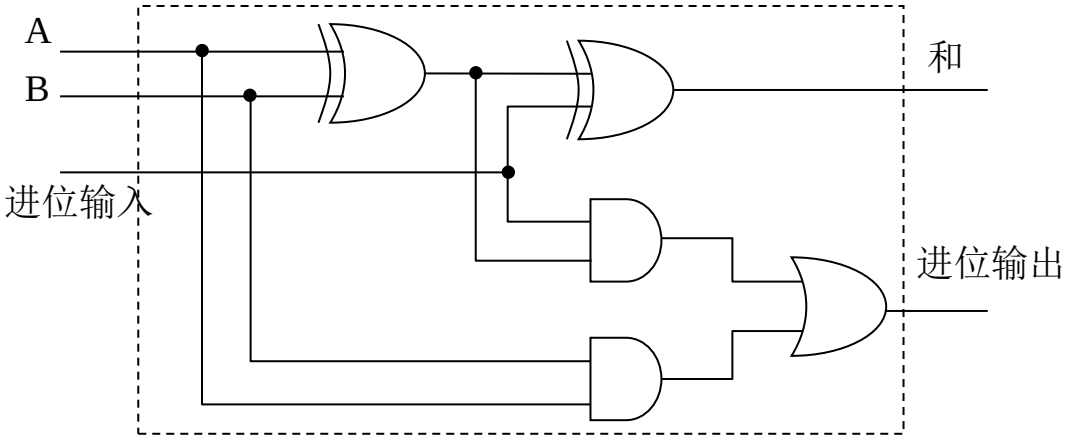
案例研究：模拟 ALU



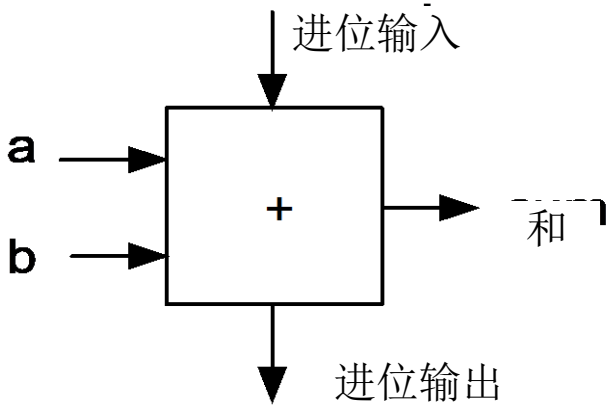




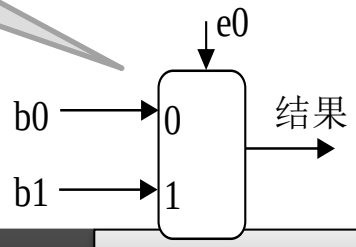




1位 加法器

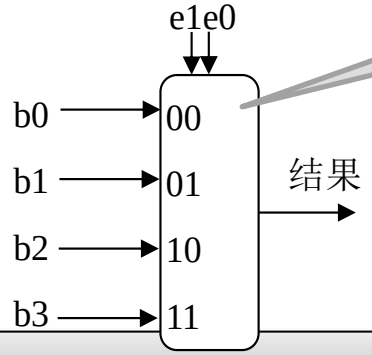


2 对 1
多路转接器



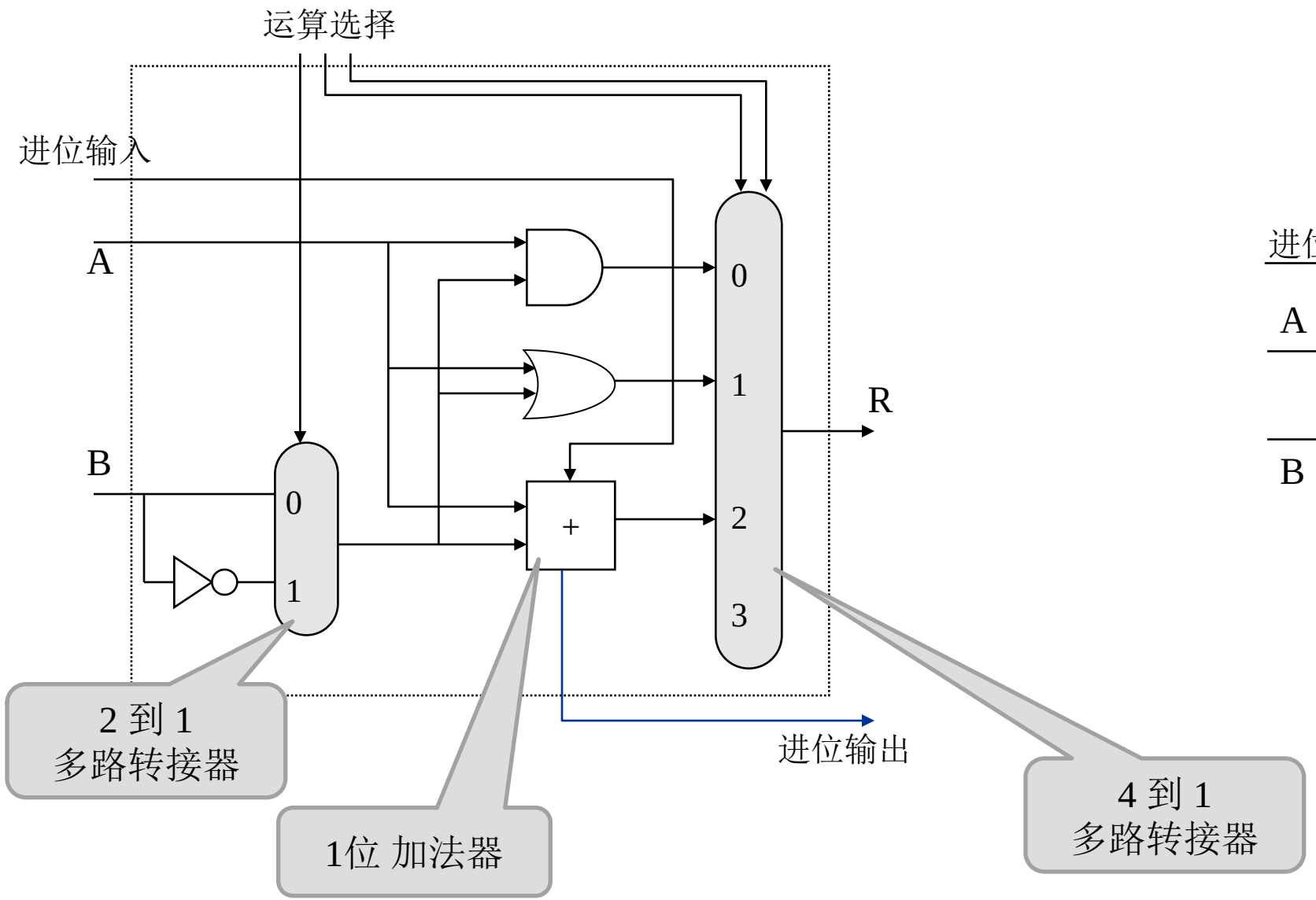
e0	结果
0	b0
1	b1

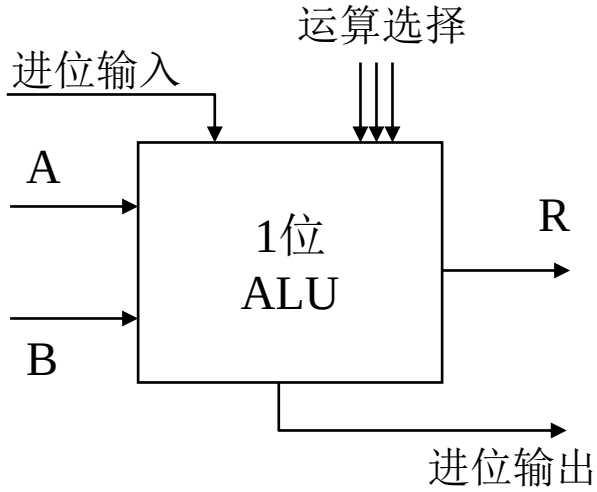
4 对 1
多路转接器



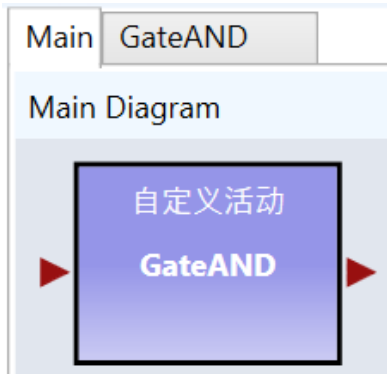
e1 e0	结果
0 0	b0
0 1	b1
1 0	b2
1 1	b3

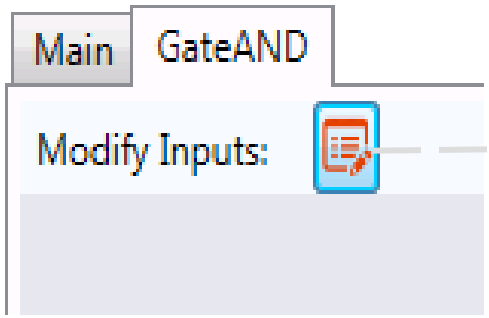
1位 ALU

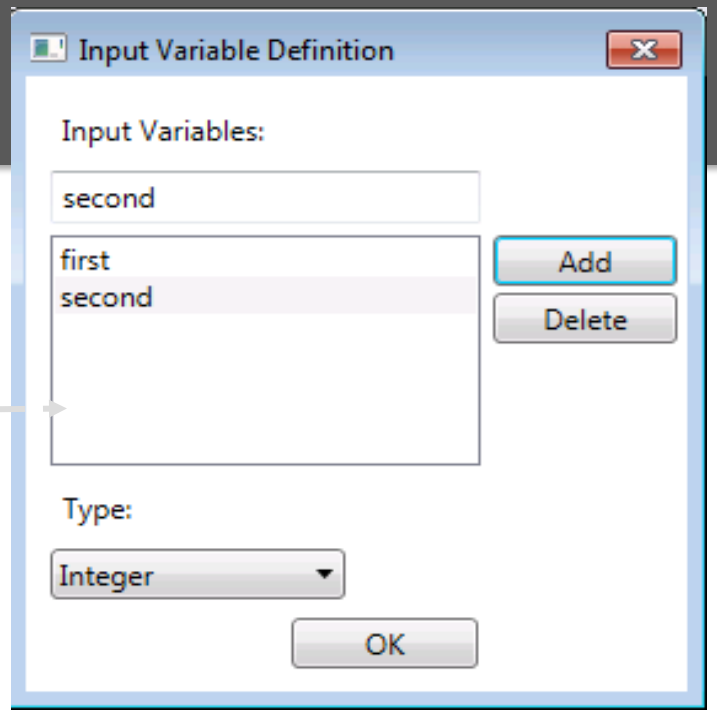

运算选择
进位输入
A
B
0
1
0
1
2
3
+
R
进位输出
2 到 1
多路转接器
1位 加法器
4 到 1
多路转接器

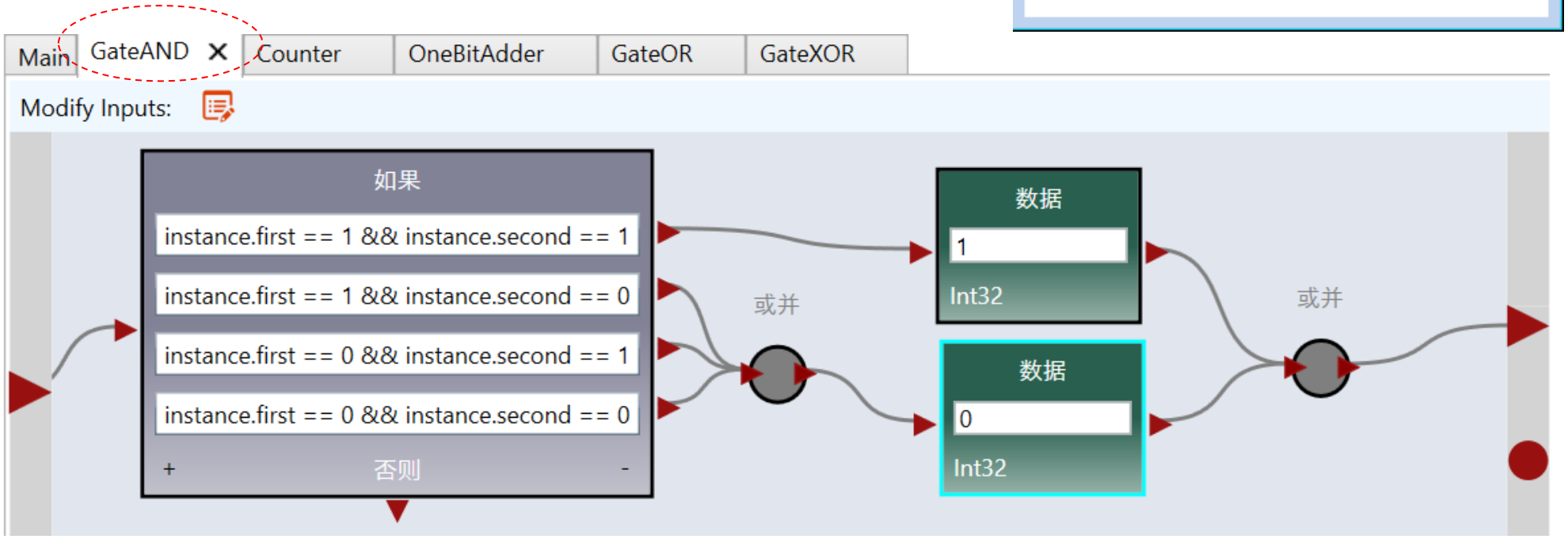

运算选择
进位输入
A
B
1位
ALU
R
进位输出

在 VIPLE 中模拟与门

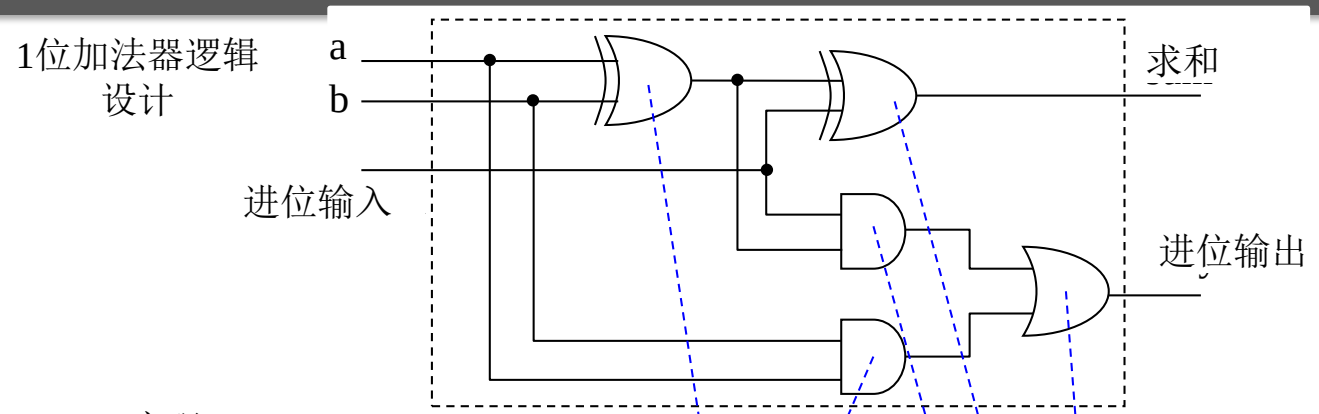




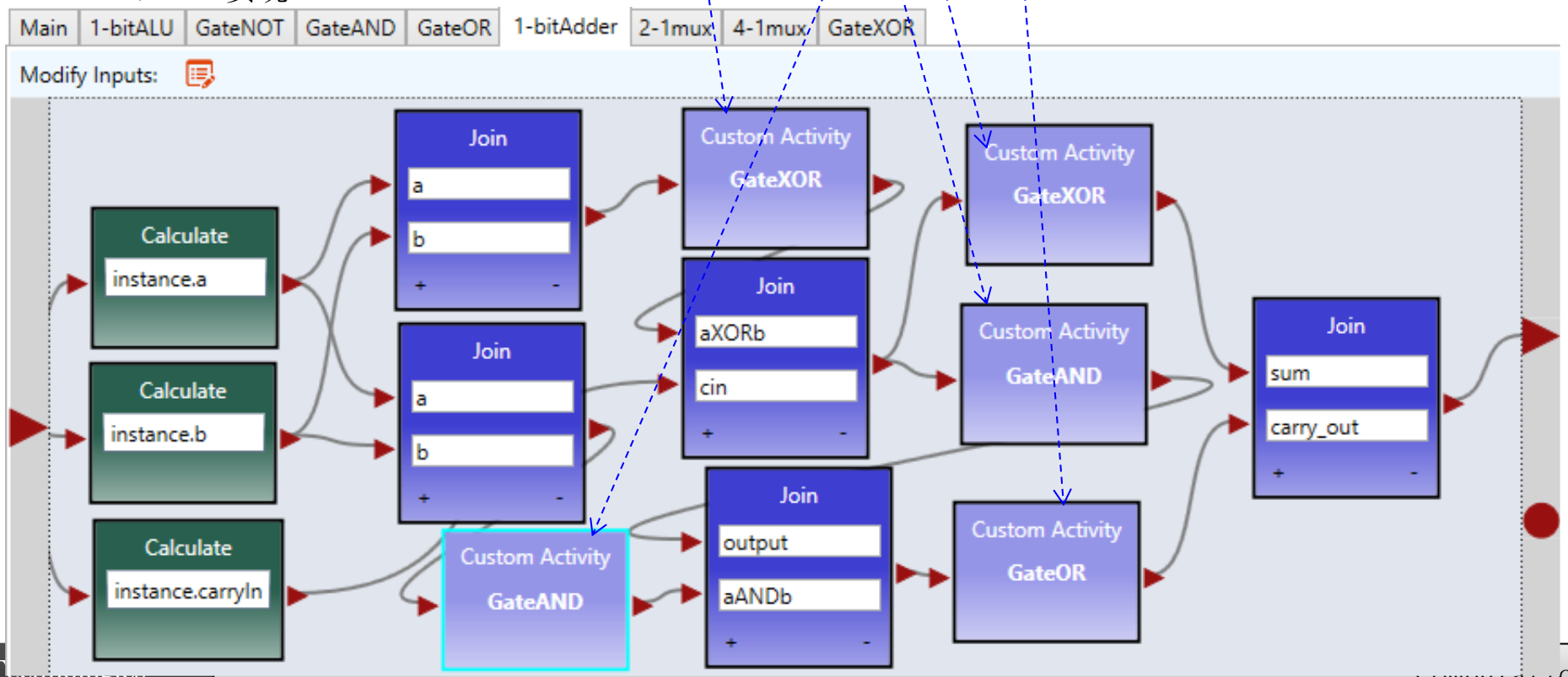




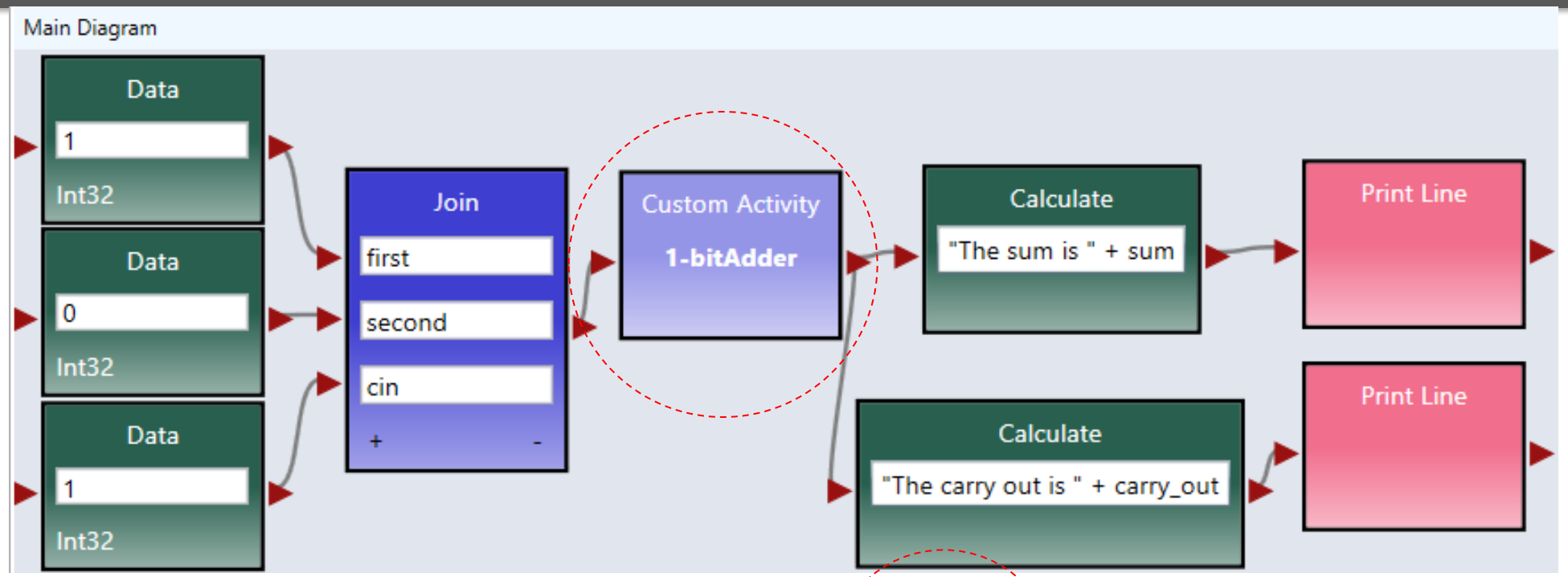
案例研究：构建1位加法器



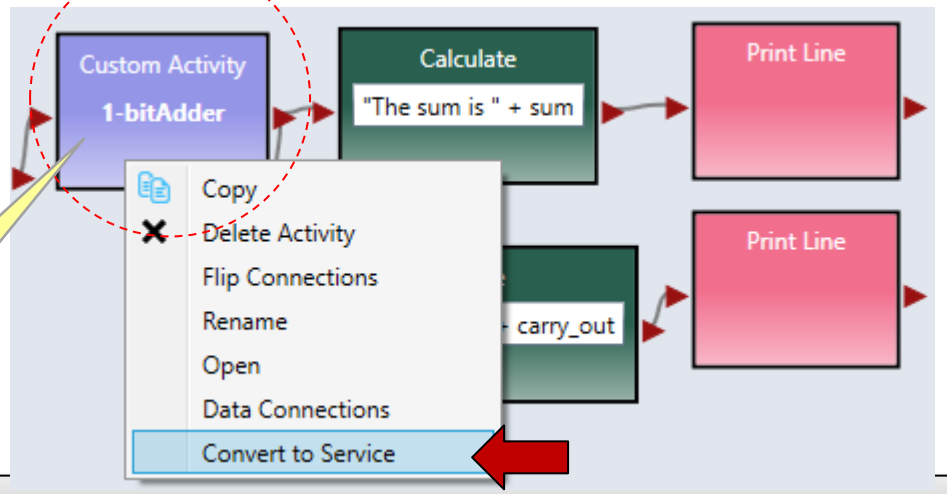
VIPLE 实现



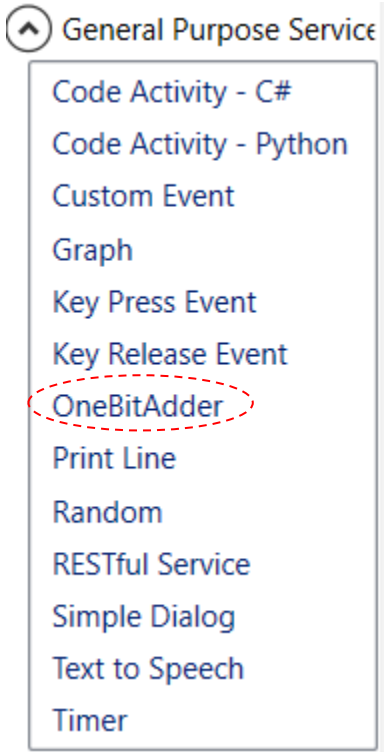
测试1位加法器



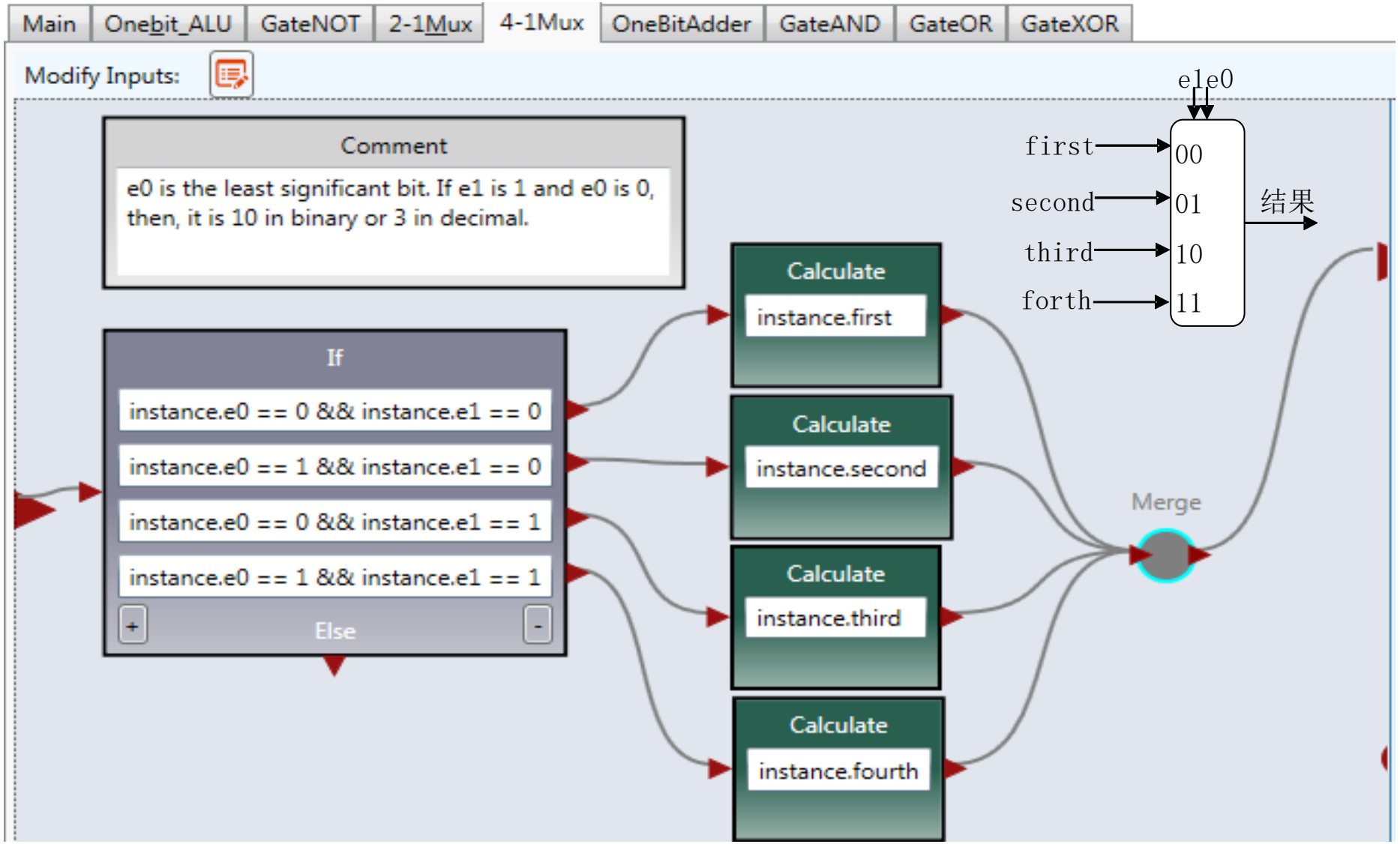
将活动转换为服务。此服务将添加到工具箱列表中



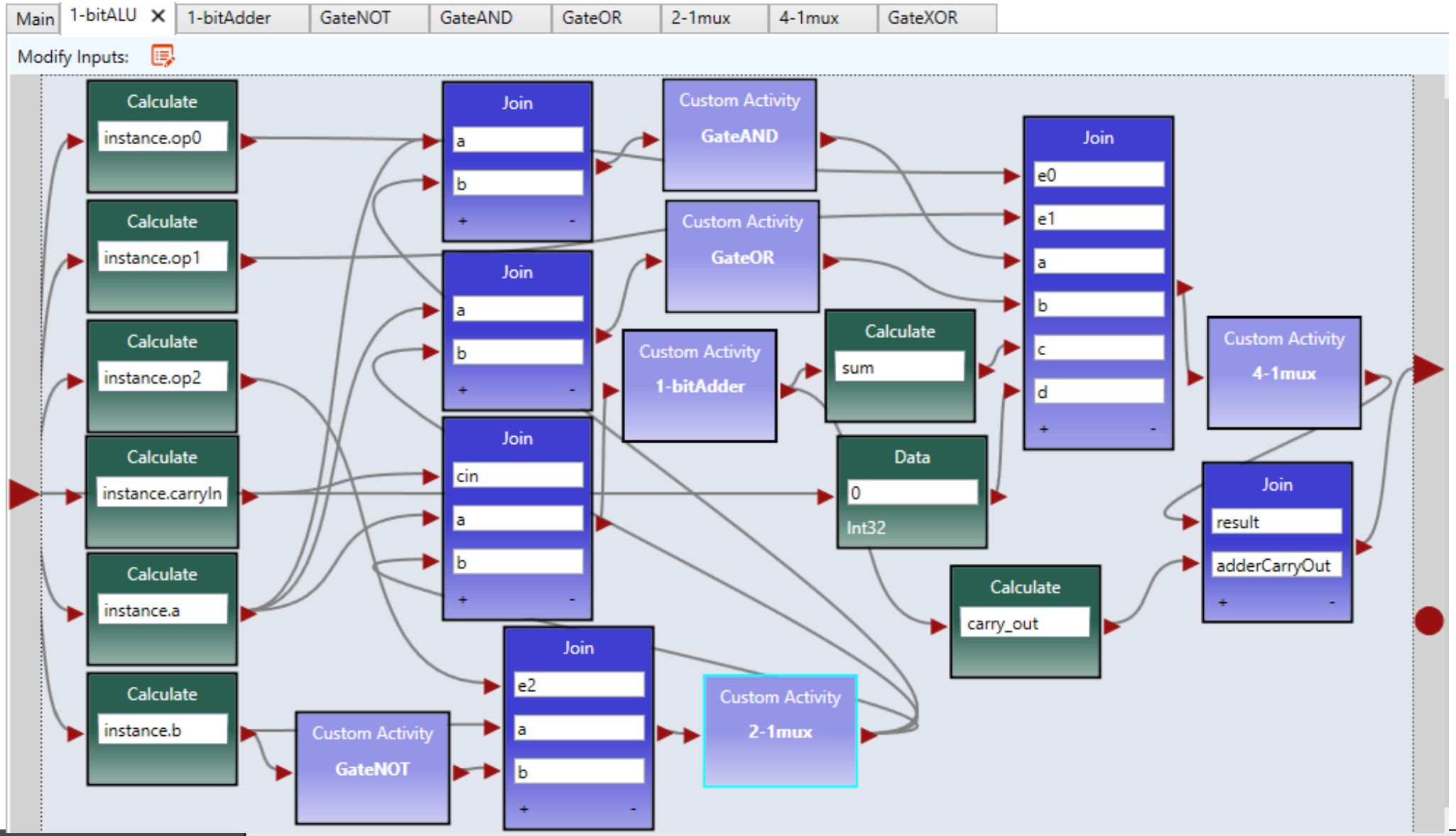
右击



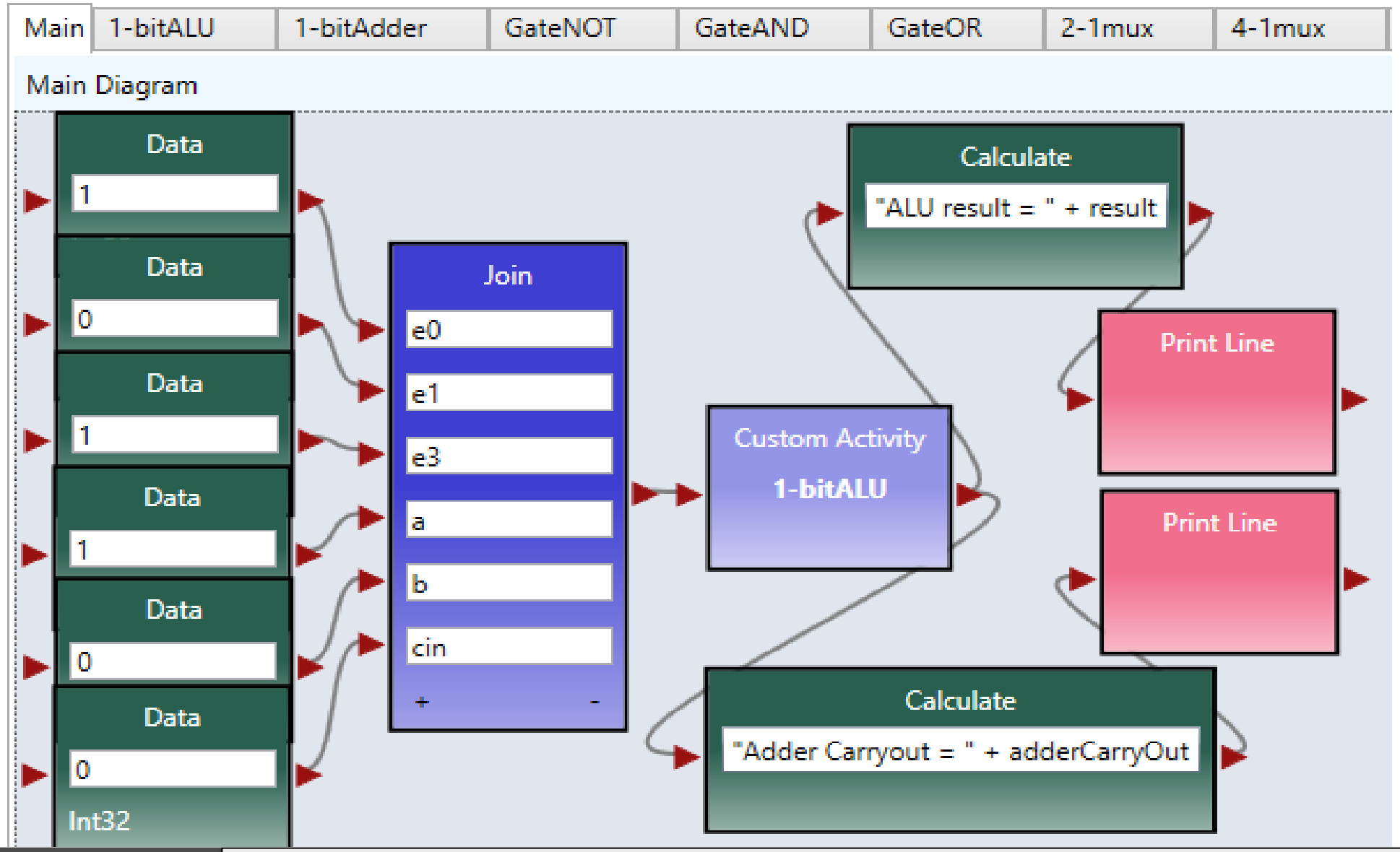
创建 2-1 多路转接器


Main
Onebit_ALU
GateNOT
2-1Mux
4-1Mux
OneBitAdder
GateAND
GateOR
GateXOR
Modify Inputs:
e1e0
Comment
e0 is the least significant bit. If e1 is 1 and e0 is 0, then, it is 10 in binary or 3 in decimal.
first
second
third
forth
00
01
10
11
结果
If
instance.e0 == 0 && instance.e1 == 0
instance.e0 == 1 && instance.e1 == 0
instance.e0 == 0 && instance.e1 == 1
instance.e0 == 1 && instance.e1 == 1
Else
Calculate
instance.first
Calculate
instance.second
Calculate
instance.third
Calculate
instance.fourth
Merge

创建1位ALU


Main
1-bitALU
1-bitAdder
GateNOT
GateAND
GateOR
2-1mux
4-1mux
GateXOR
Modify Inputs:
Calculate
instance.op0
Calculate
instance.op1
Calculate
instance.op2
Calculate
instance.carryIn
Calculate
instance.a
Calculate
instance.b
Join
a
b
Join
a
b
Join
cin
a
b
Join
e2
a
b
Custom Activity
GateAND
Custom Activity
GateOR
Custom Activity
1-bitAdder
Custom Activity
GateNOT
Custom Activity
2-1mux
Calculate
sum
Data
0
Int32
Calculate
carry_out
Join
e0
e1
a
b
c
d
Custom Activity
4-1mux
Join
result
adderCarryOut

测试1位ALU


Main
1-bitALU
1-bitAdder
GateNOT
GateAND
GateOR
2-1mux
4-1mux
Main Diagram
Data
1
Data
0
Data
1
Data
1
Data
0
Data
0
Int32
Join
e0
e1
e3
a
b
cin
+
-
Custom Activity
1-bitALU
Calculate
"ALU result = " + result
Print Line
Print Line
Calculate
"Adder Carryout = " + adderCarryOut

自动化测试用例生成

- 手动测试既耗时又乏味
- 分析1位加法器的测试用例生成

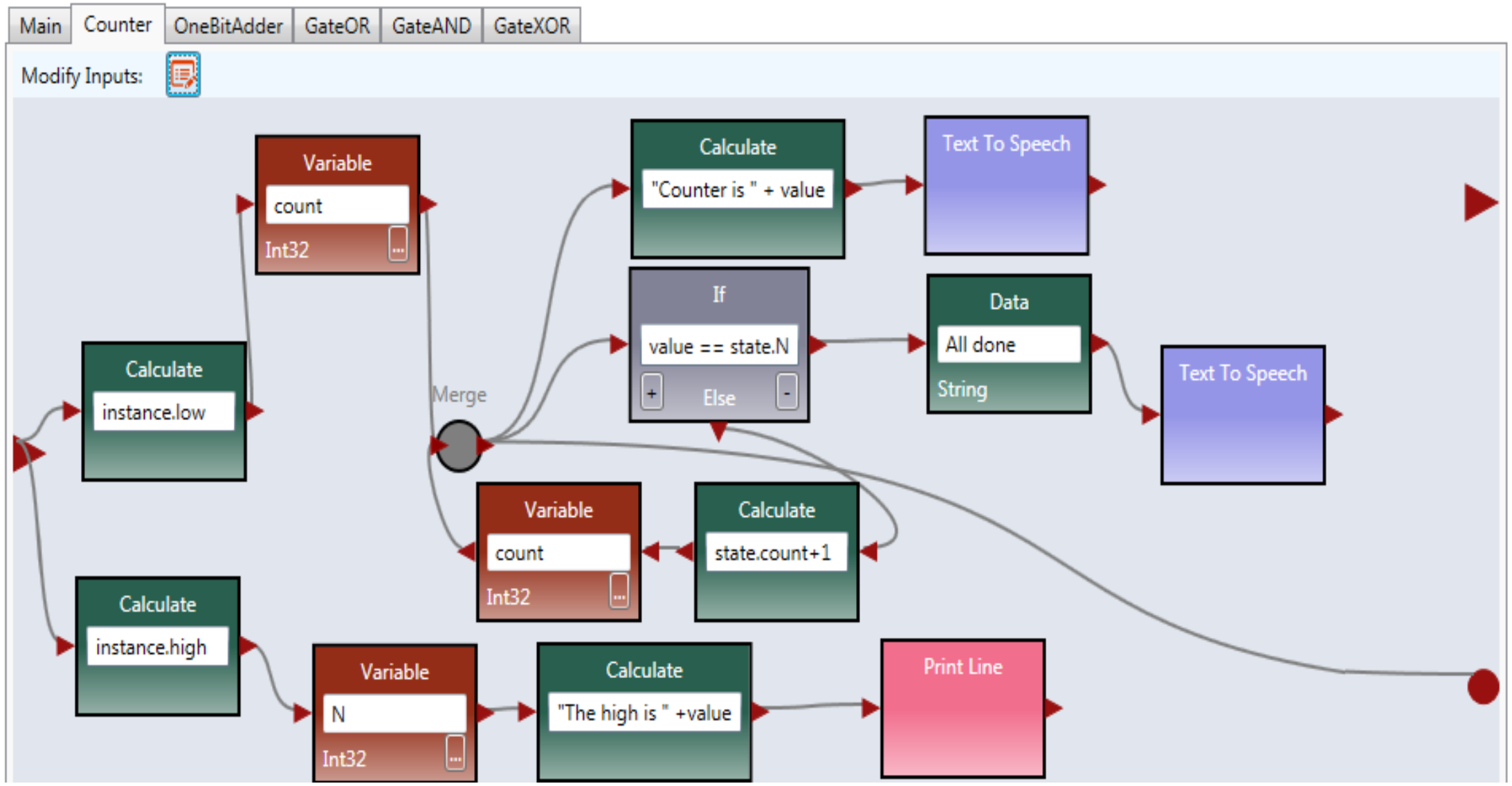
CountTo7	a	b	carryIn
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

if CountTo7 = 0, 1, 2, 3, then a = 0, else a = 1;

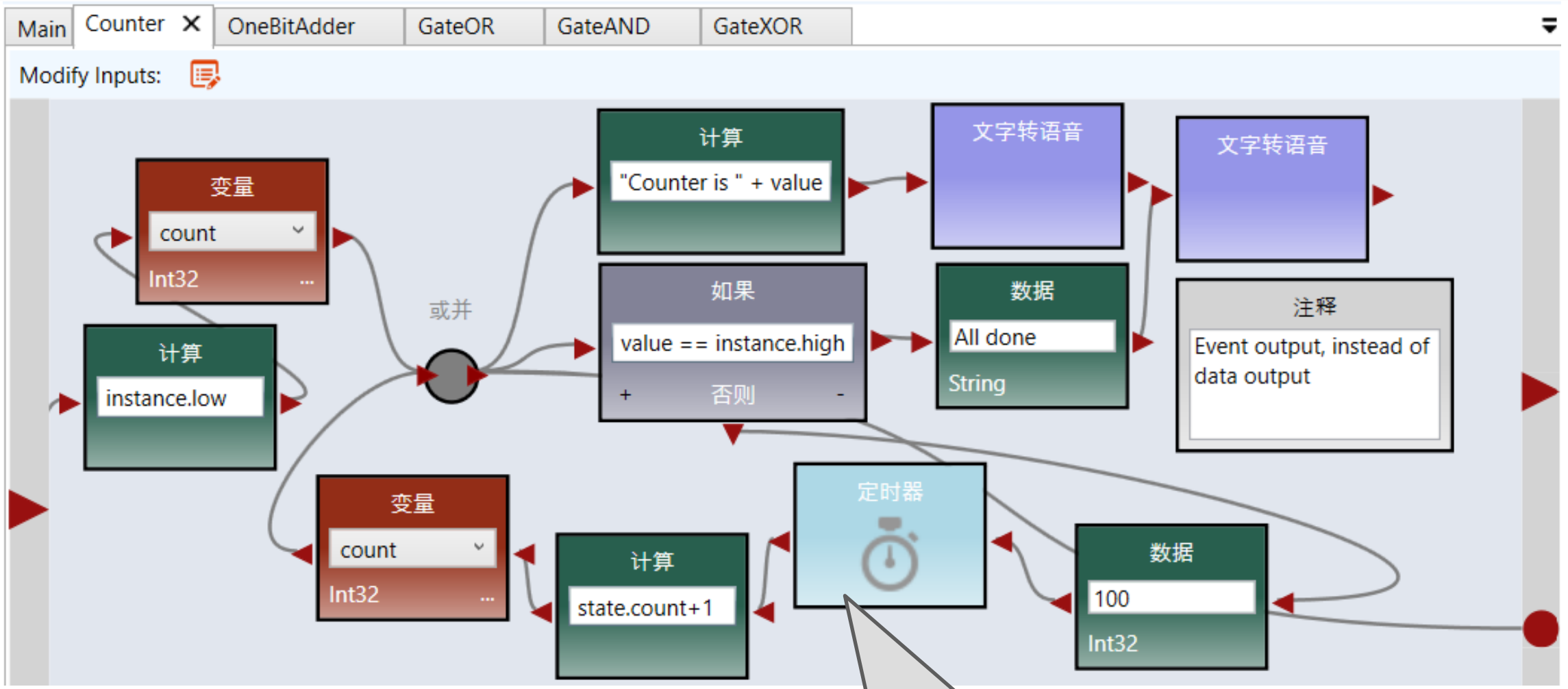
if CountTo7 = 0, 1, 4, 5, then b = 0, else b = 1;

if CountTo7 = 0, 2, 4, 6, then carryIn = 0, else carryIn = 1;

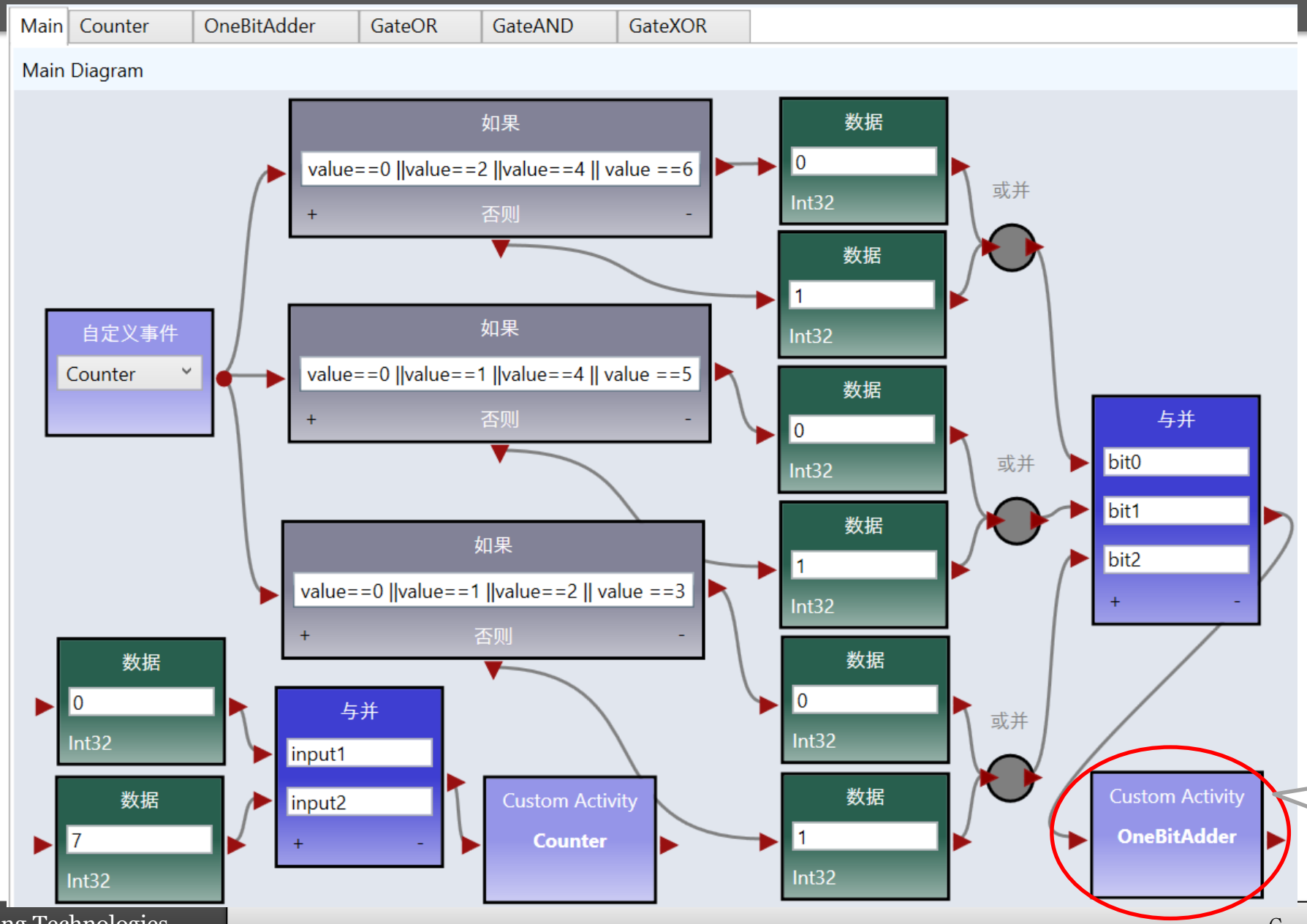
将测试用例作为事件生成的计数器



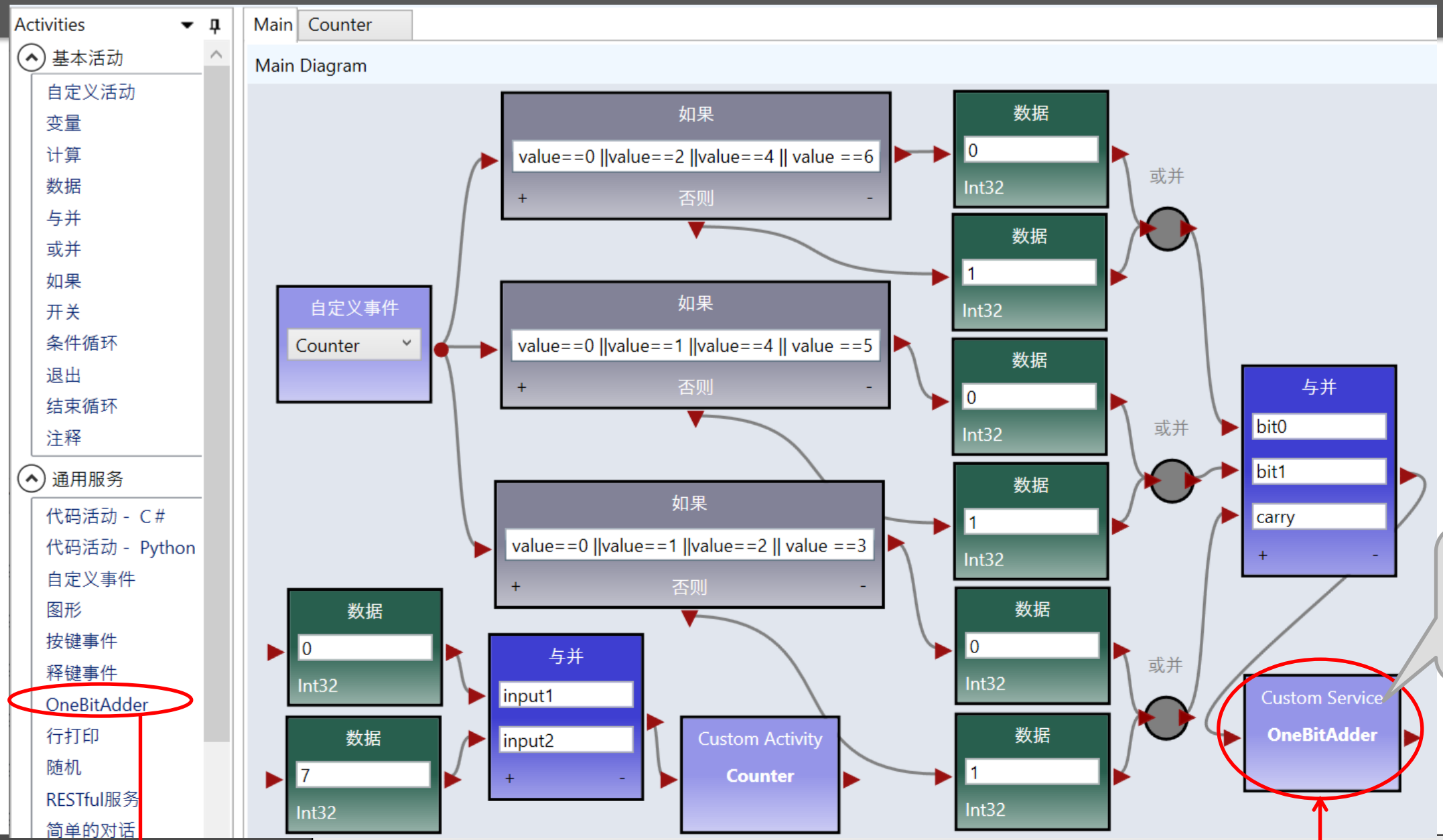
在计数器中添加延迟，以保证计数与加法器同步



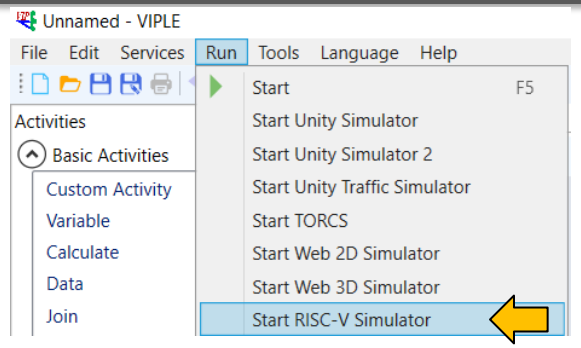
1位加法器自动测试（活动）

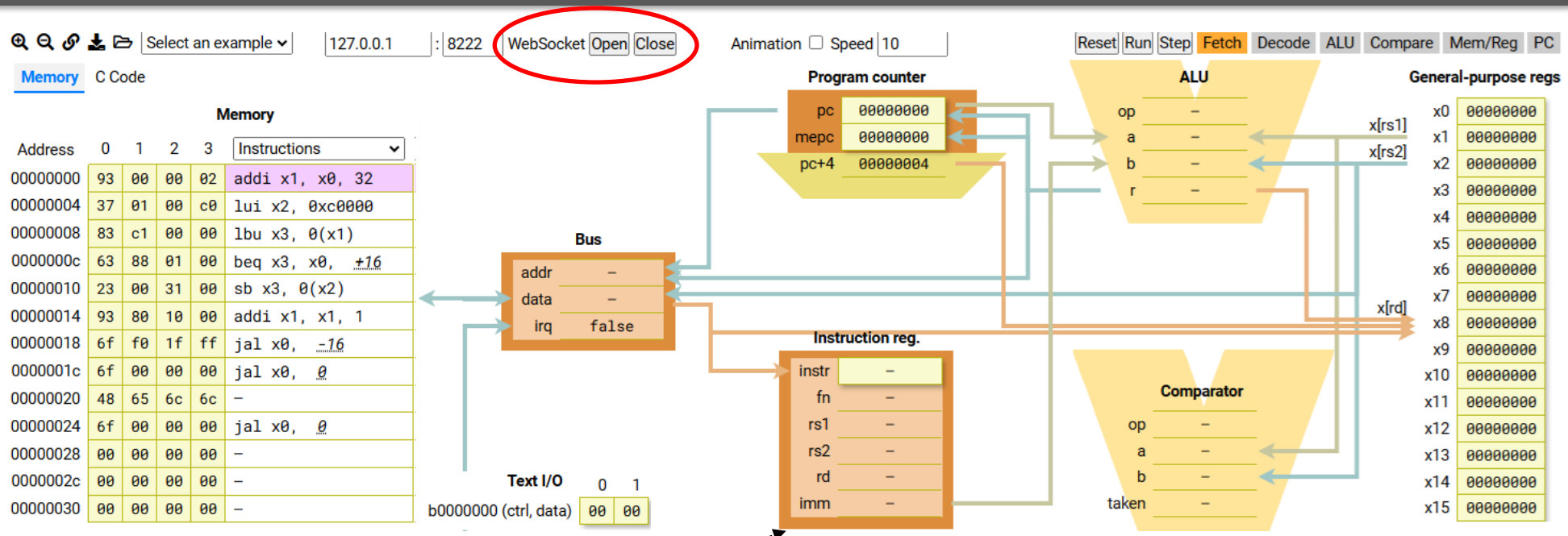


1位加法器自动测试（服务）

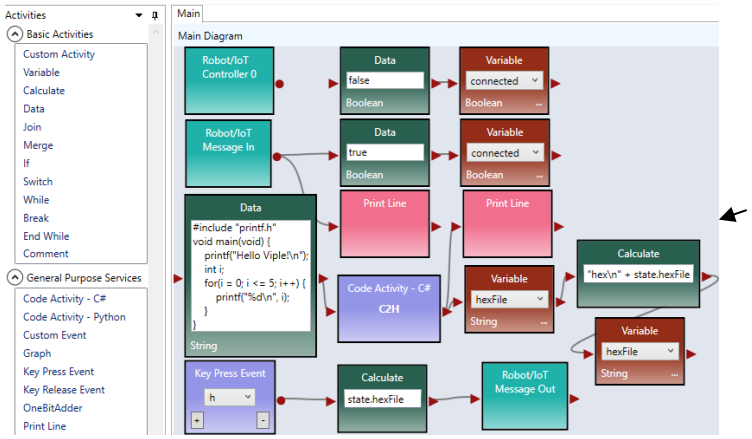


RISC-V 计算机架构仿真器





VIPLE Program

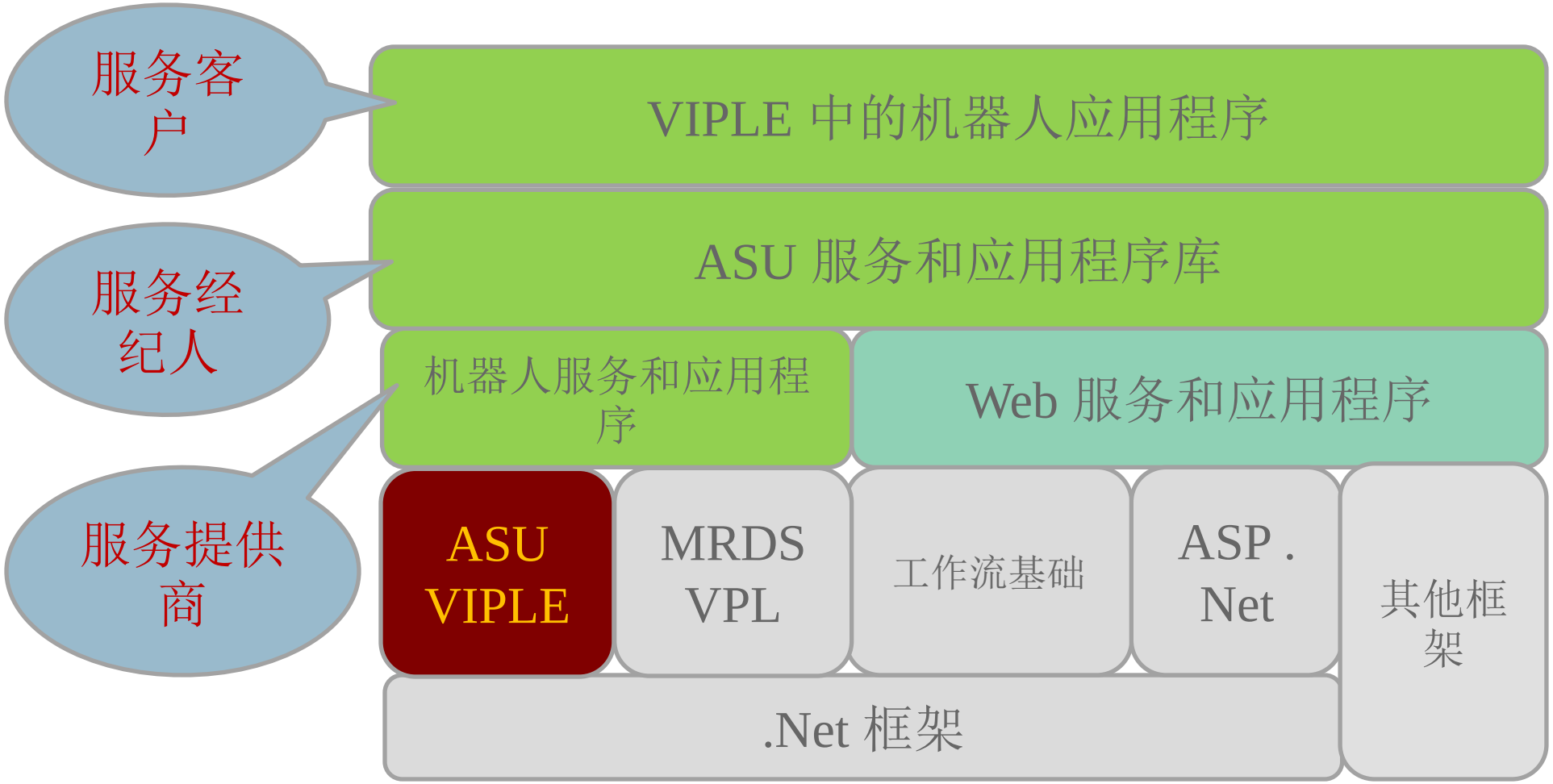


WebSocket

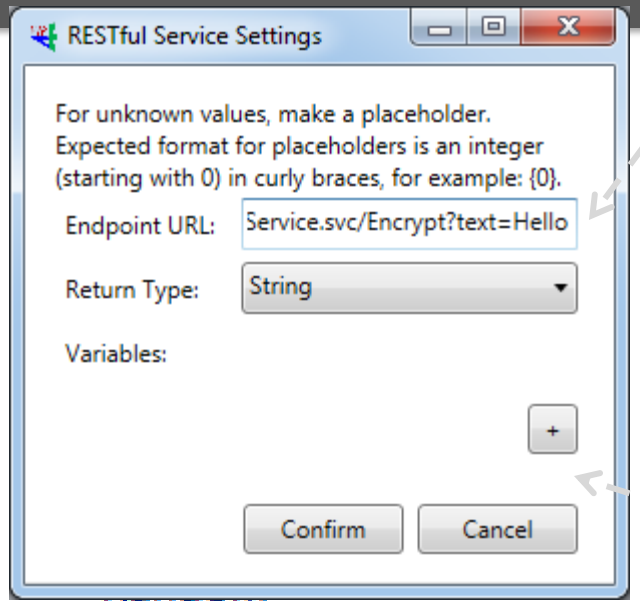
RISC-V 计算机架构仿真器

https://venus.sod.asu.edu/VIPLE/RISCV/

VIPLE 面向服务


服务客户
服务经纪人
服务提供商
VIPLE 中的机器人应用程序
ASU 服务和应用程序库
机器人服务和应用程序
Web 服务和应用程序
ASU VIPLE
MRDS VPL
工作流基础
ASP . Net
其他框架
.Net 框架

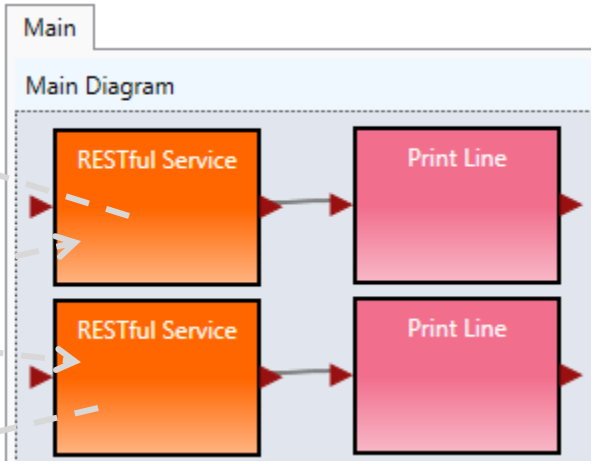
调用 RESTful 服务(Web API)

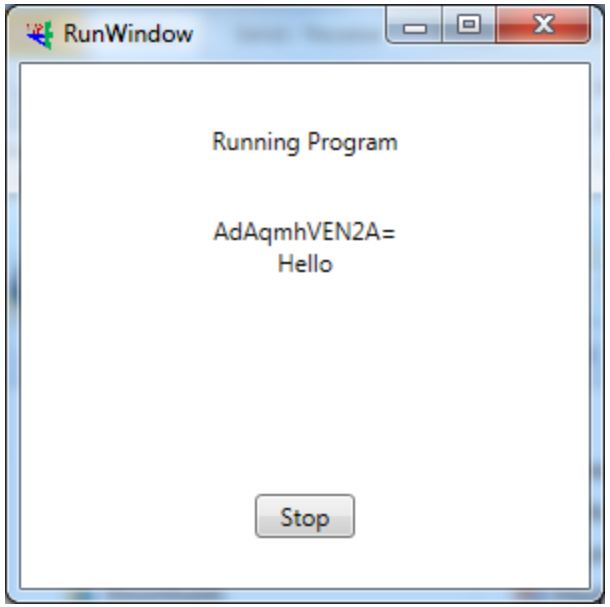


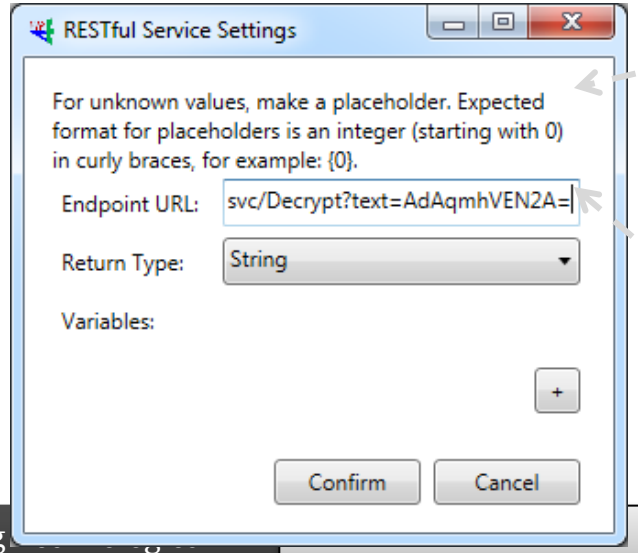
http://venus.sod.asu.edu/WSRepository/Services/EncryptionRest/Service.svc/Encrypt?text=Hello

服务地址，参数值 = Hello，被调用

RESTful Service



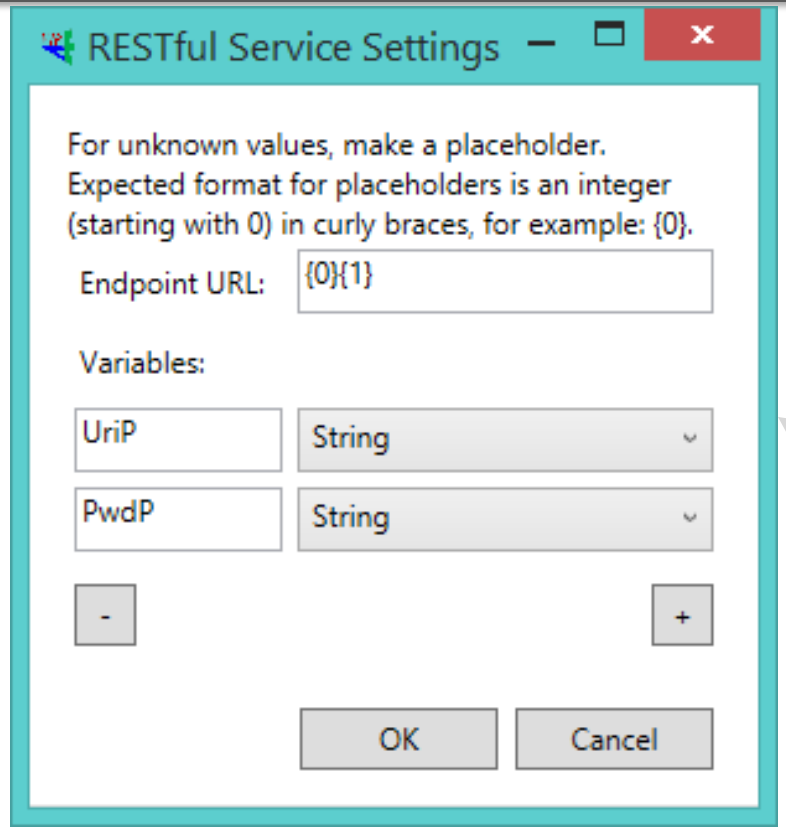


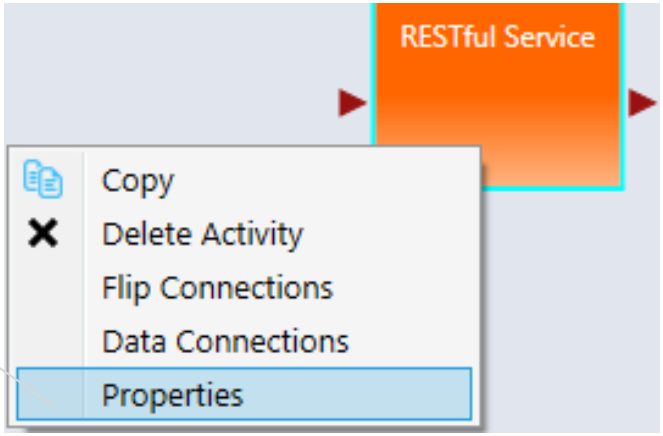


http://venus.sod.asu.edu/WSRepository/Services/EncryptionRest/Service.svc/Decrypt?text=AdAqmhVEN2A=

服务地址，参数值 = AdAqmhVEN2A=，被调用

调用 RESTful 服务 2



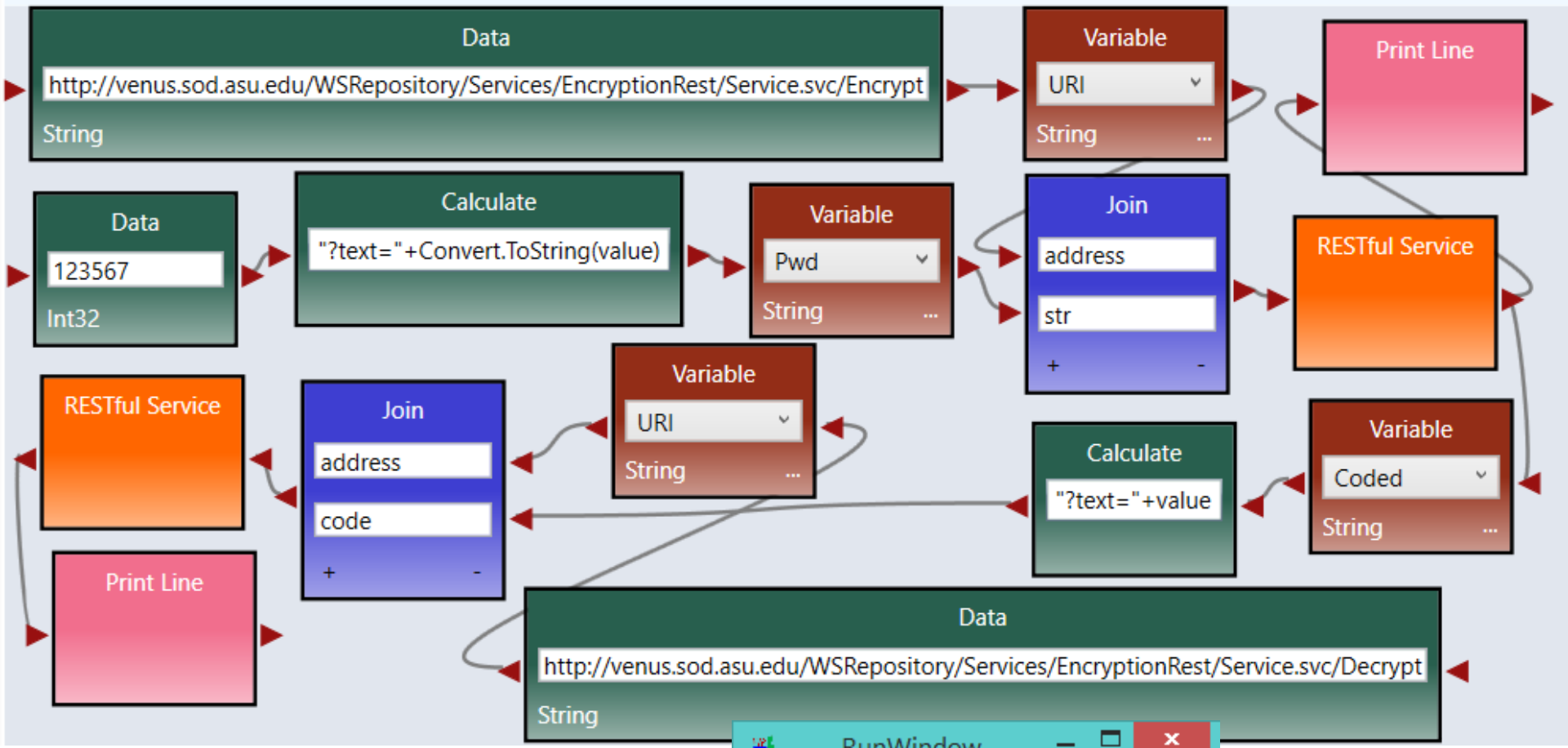


http://venus.sod.asu.edu/WSRepository/Services/EncryptionRest/Service.svc/Encrypt?text=Hello

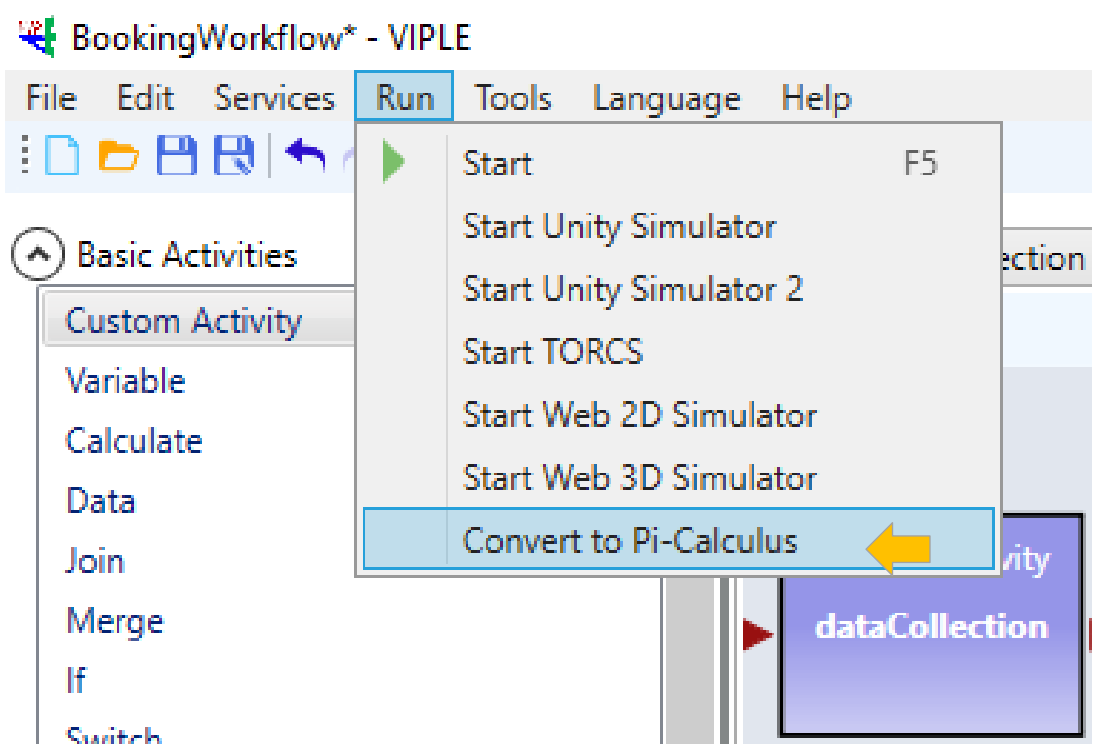
基础地址 {0}

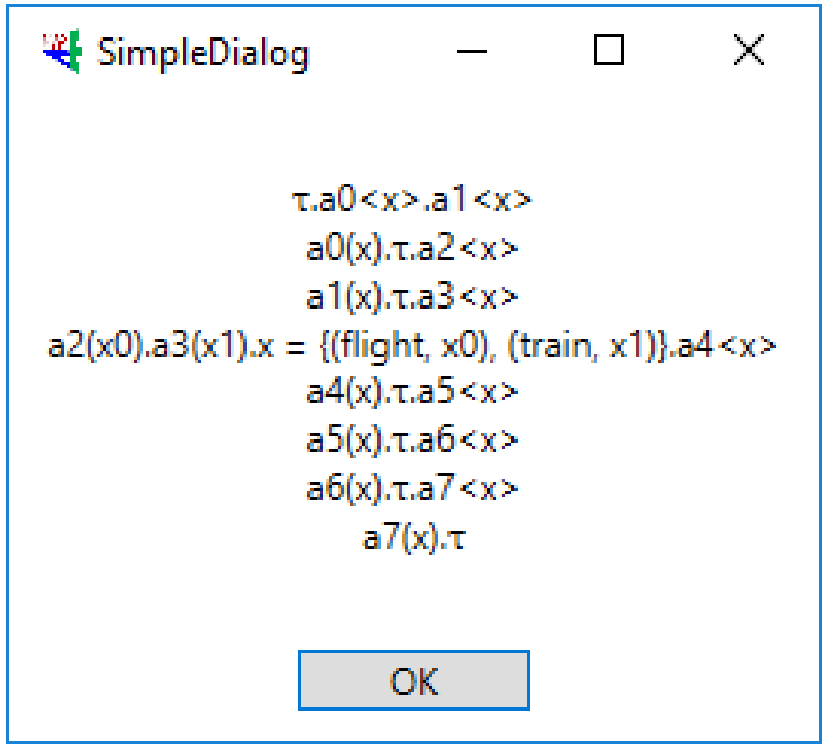
参数值 {1}

调用 RESTful 服务 3

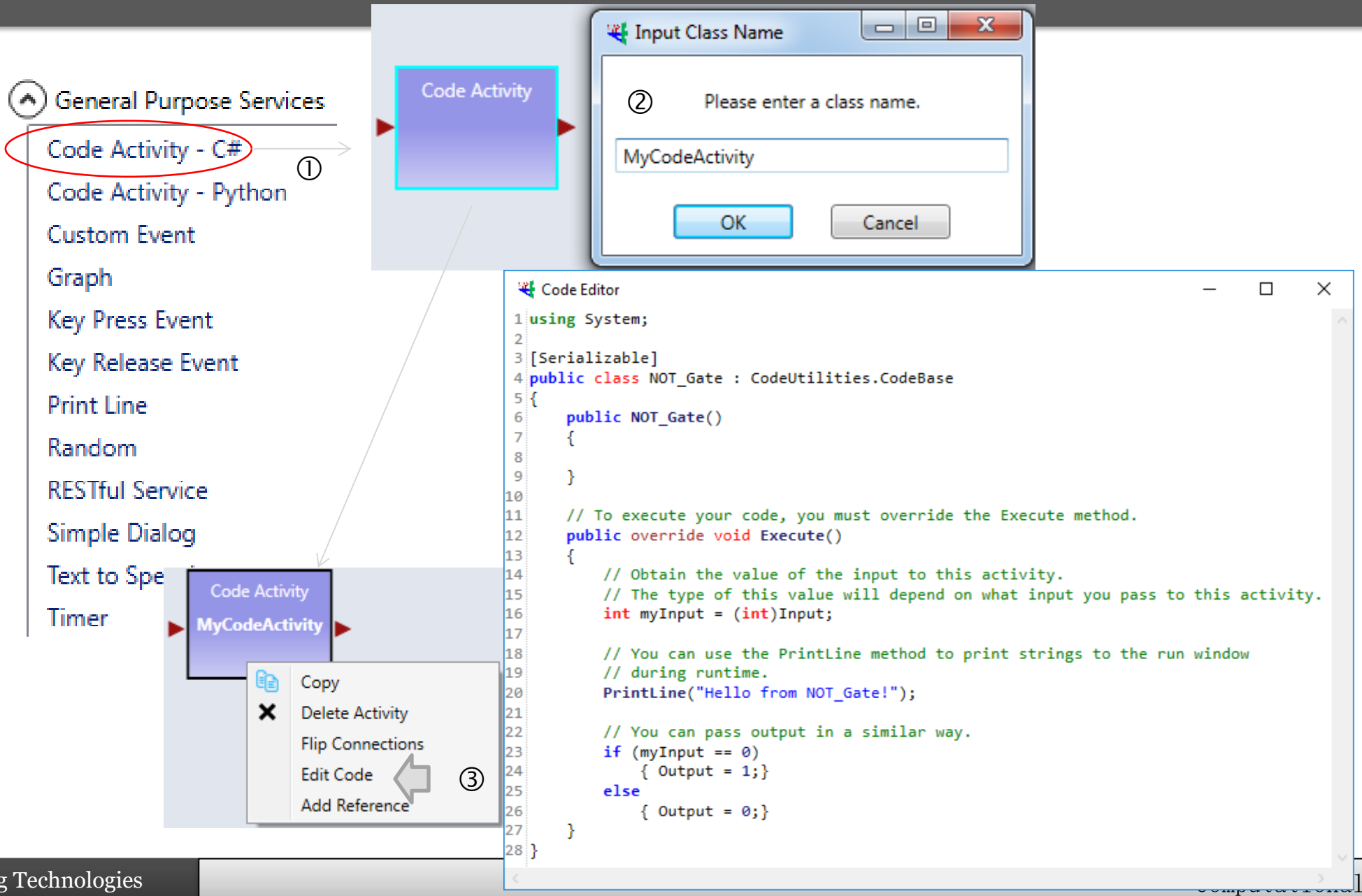


将 VIPLE 代码转换为 π-Calculus 表达式



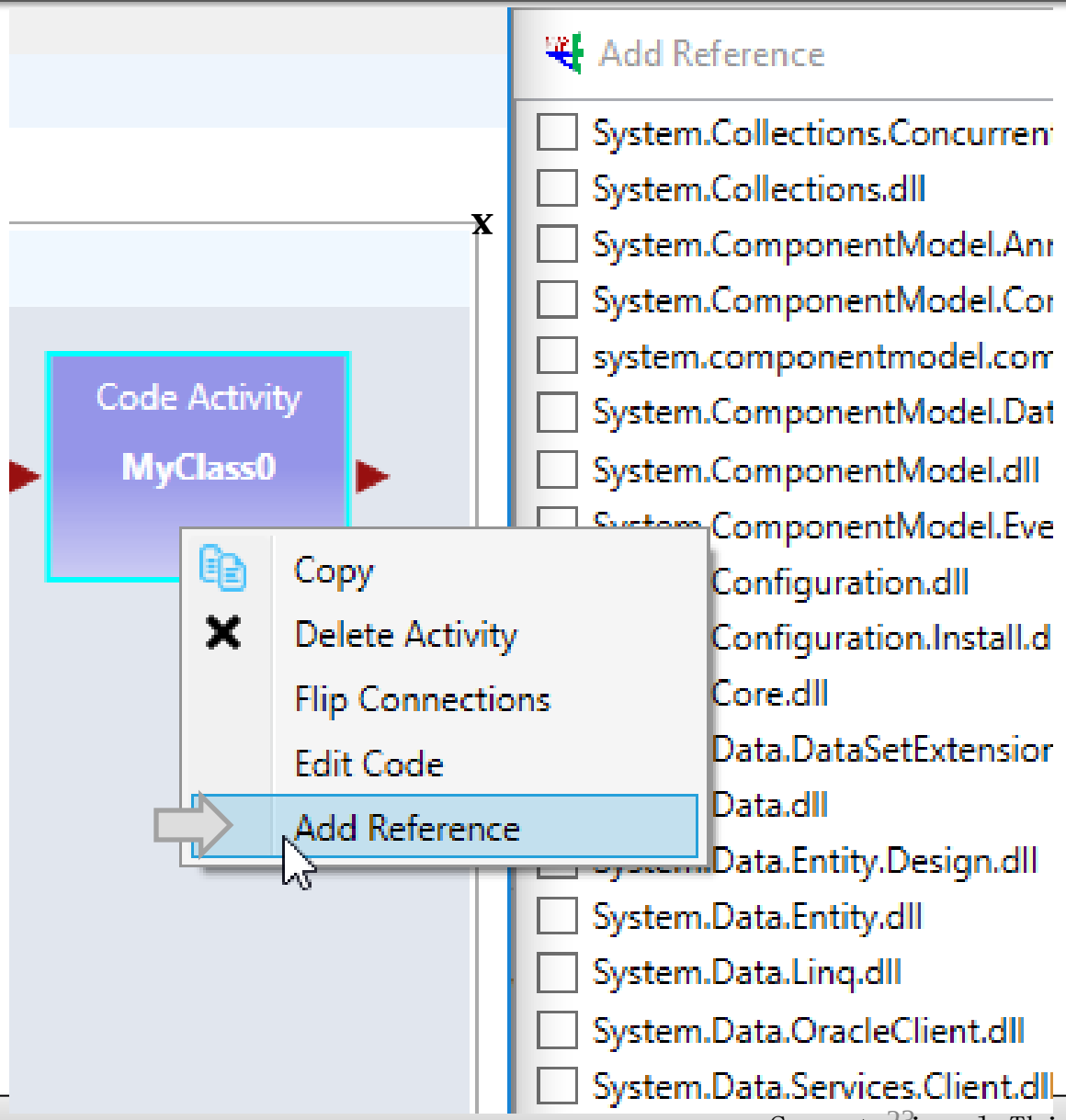


Code Activity：将任何 C# 类装入一个活动中



将 DLL库添加至 CodeActivity

- 使用系统；已经包含在 CodeActivity
- 如果你想使用额外的 DLL 库函数，你可以右击你的 CodeActivity，然后选择 Add Reference。
- 选中你要添加的任何 dll 库。
- 添加库后，你可以将：using System.Collections.dllin System.Collections.dll 添加到你的程序中



CodeActivity 示例

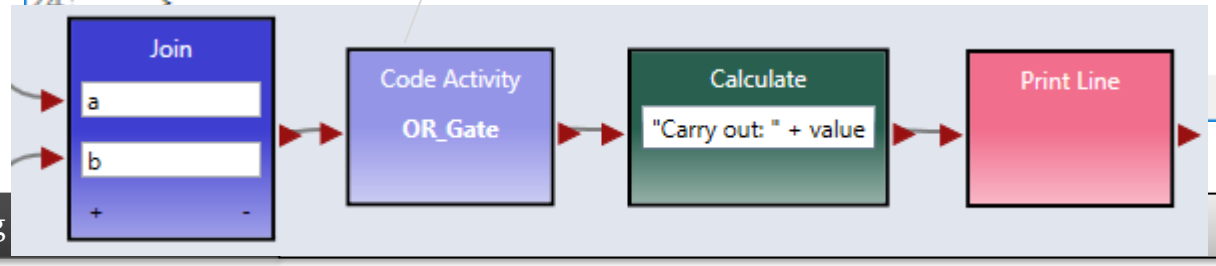
Code Editor

```
1  using System;
2  using System.Collections.Generic;
3  using VisualProgrammingEnvironment;
4
5  public class OR_Gate : CodeUtilities.CodeBase
6  {
7      // Setting the return type up here allows you to use the "value" keyword
8      // correctly in connected activities.
9      public OR_Gate()
10     {
11     }
12
13     // To execute your code, you must override the Execute method.
14     public override void Execute()
15     {
16         Dictionary<string, object> input = (Dictionary<string, object>)Input;
17         int x = (int)input["a"];
18         int y = (int)input["b"];
19
20         if (x == 0 && y == 0)
21             Output = 0;
22         else
23             Output = 1;
24     }
```

用于在 CodeActivity 中使用 Join 取多个值

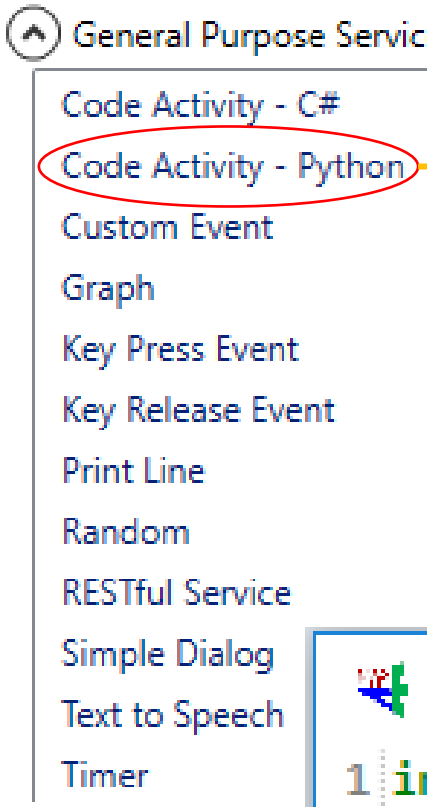
如果只有一个输入，我们不使用 Join。在这一情况下，我们使用：
int x = (int)Input

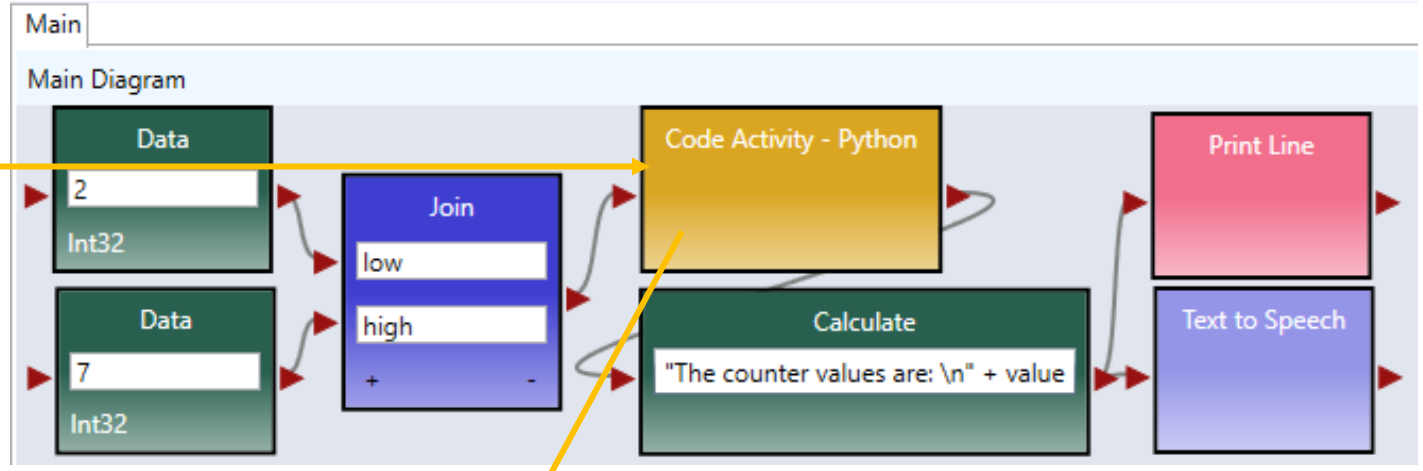
如果需要多个输出，将它们组合到一个结构中



注意：某些计算机/VS 版本可能需要你添加对 System.Runtime 的引用，才能使此代码正常工作。

Code Activity Python：将 Python 代码装入一个活动





Code Editor

```
1 import sys
2 lower = int(sys.argv[1])
3 higher = float(sys.argv[2])
4 counter = lower
5 while counter <= higher:
6     print("The counter value is ", counter)
7     counter = counter + 1
8 print("All done")
```