

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：

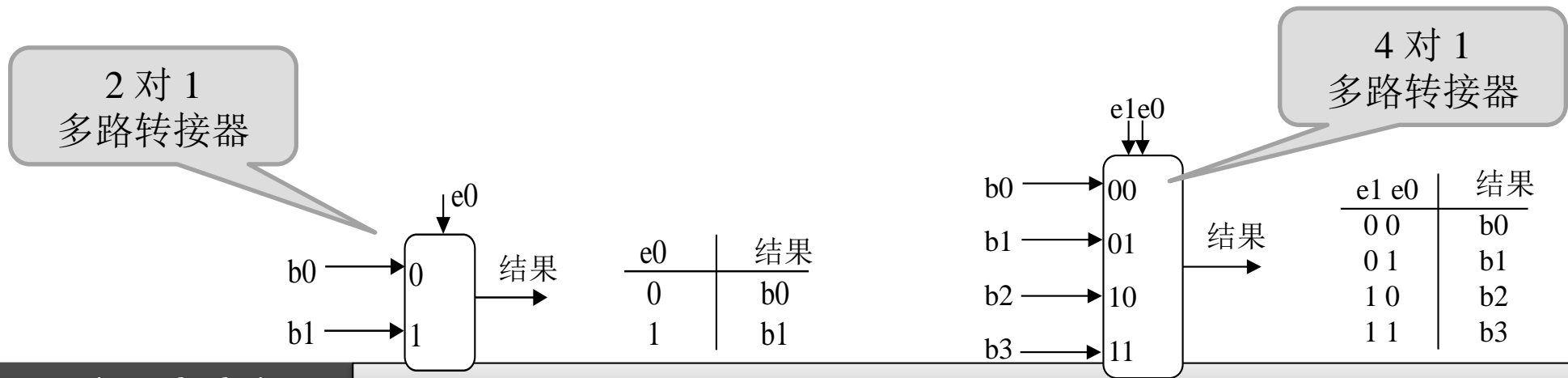
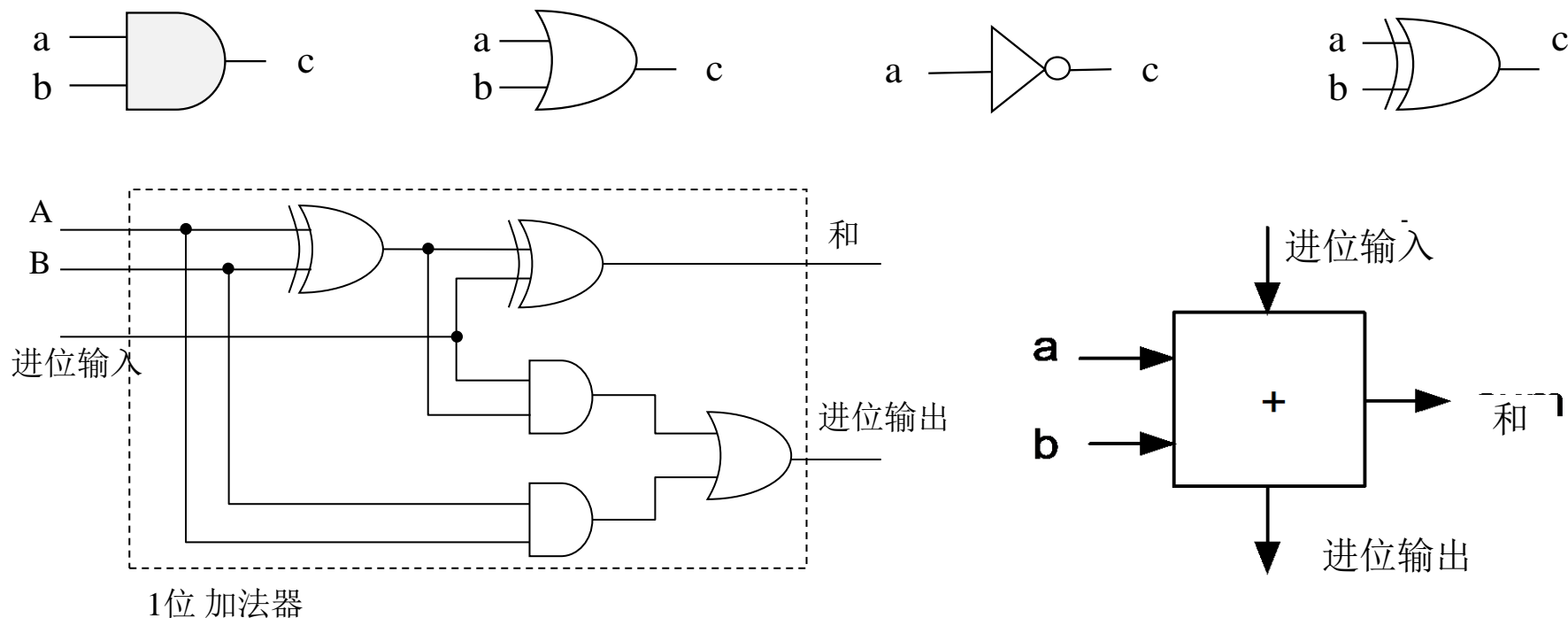
Y. Chen, G. De Luca Service-Oriented Computing and System Integration: Software, IoT, Big Data, and AI as Services, 8th edition, Kendall Hunt Publishing, 2022.

<https://www.public.asu.edu/~ychen10/book/socsi.html>

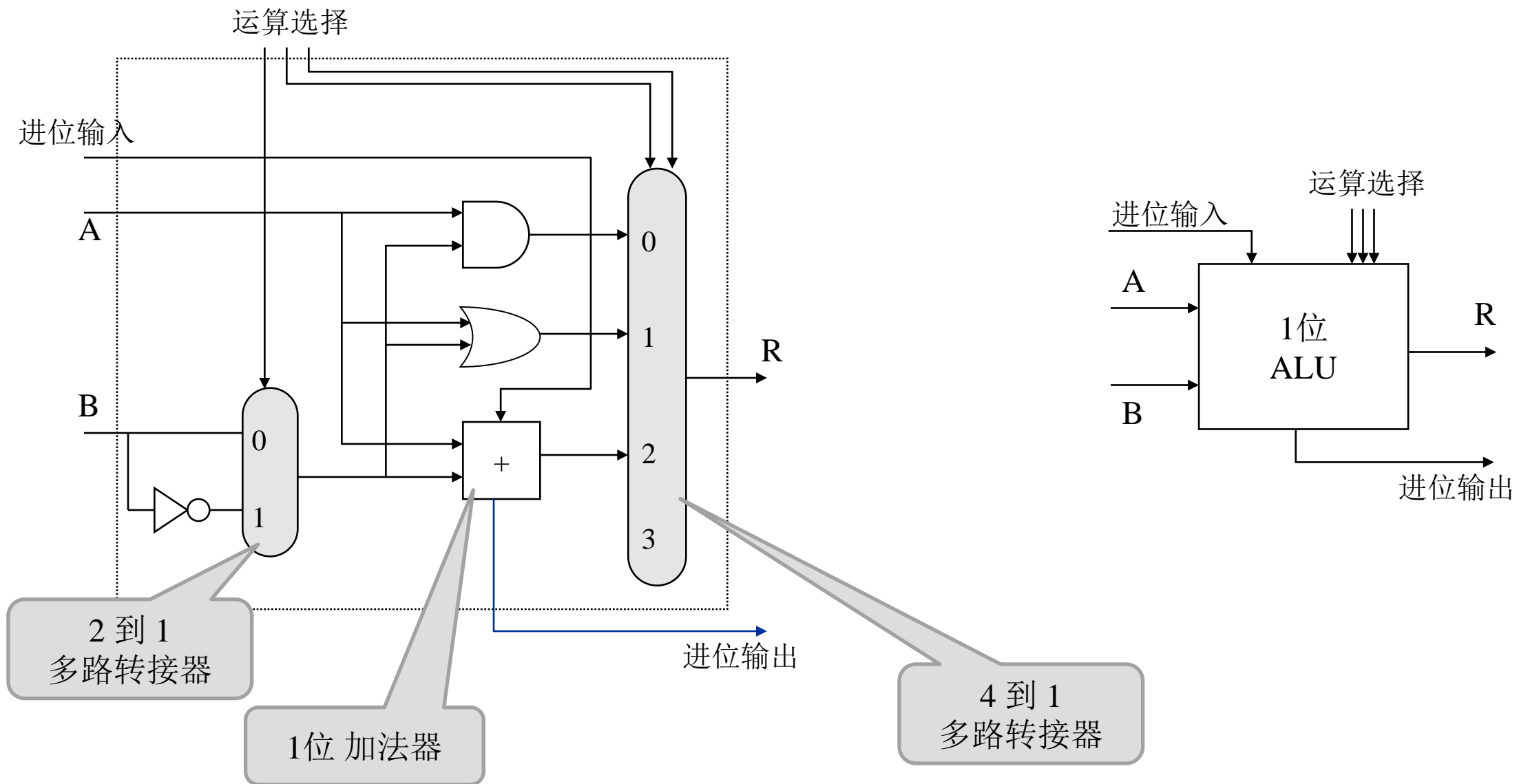
学习

- ❑ 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

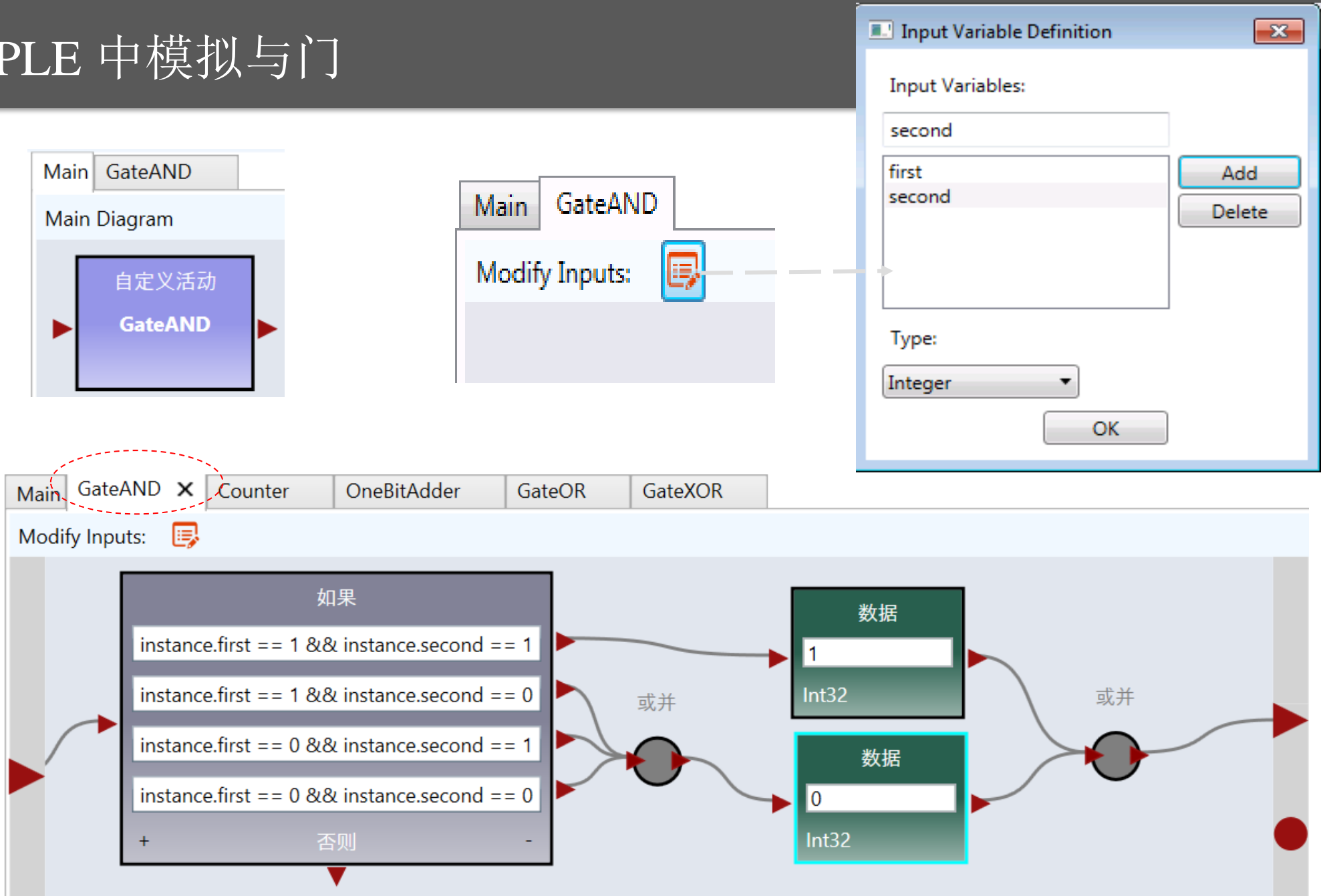
案例研究：模拟 ALU



1位 ALU

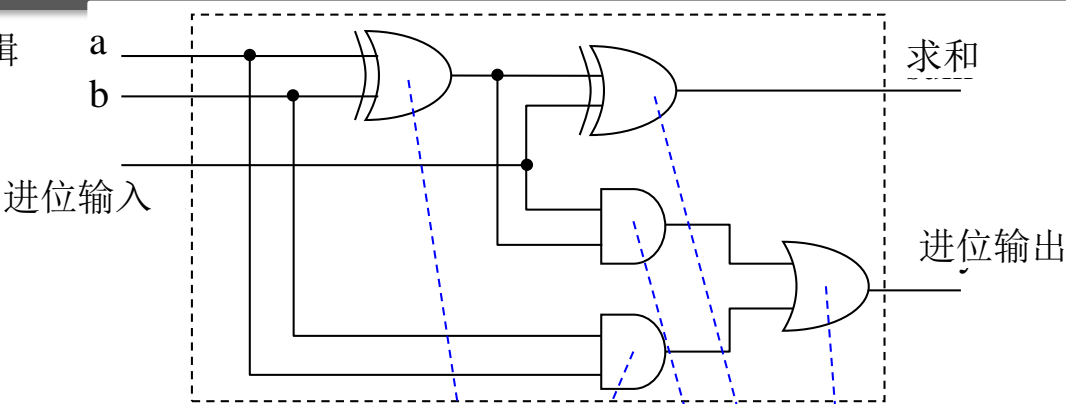


在 VIPLE 中模拟与门

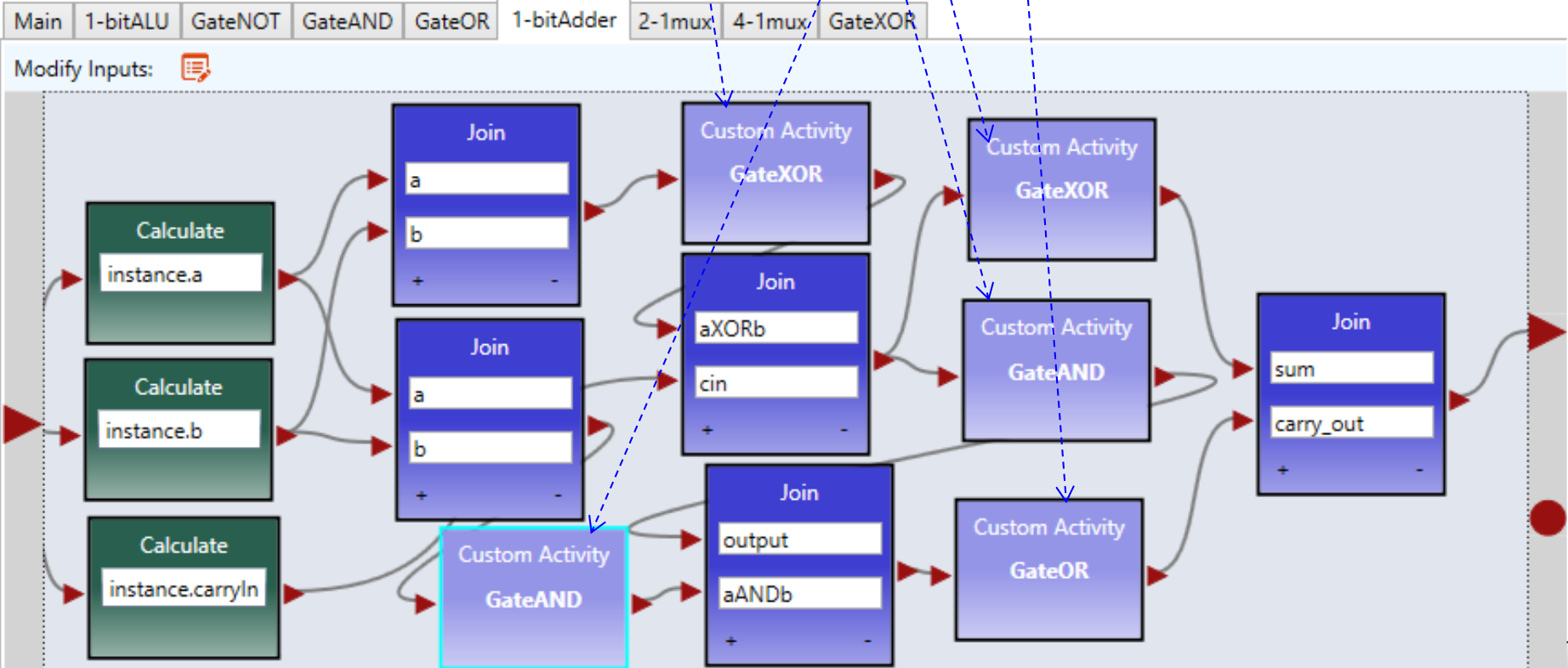


案例研究：构建1位加法器

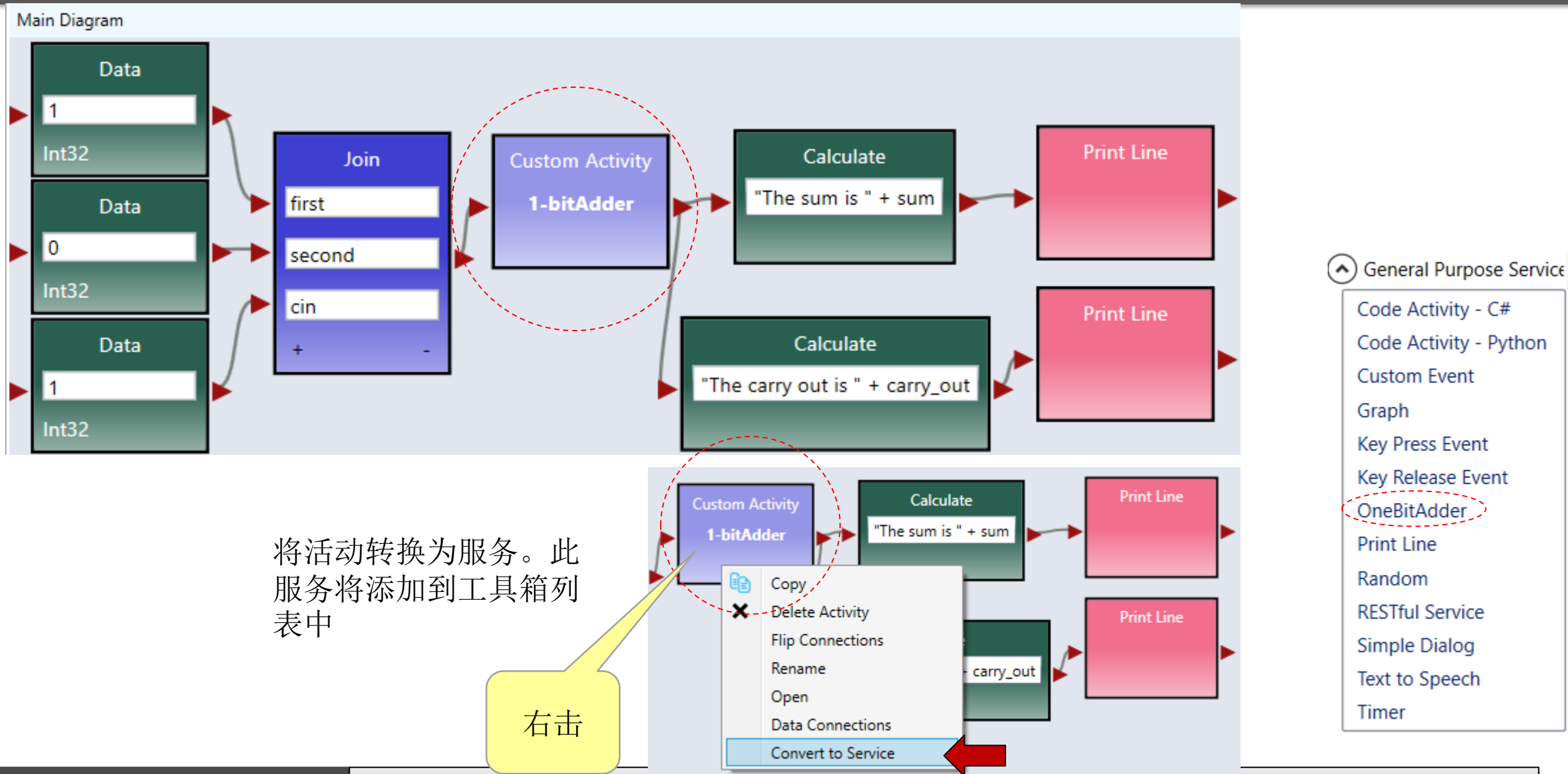
1位加法器逻辑设计



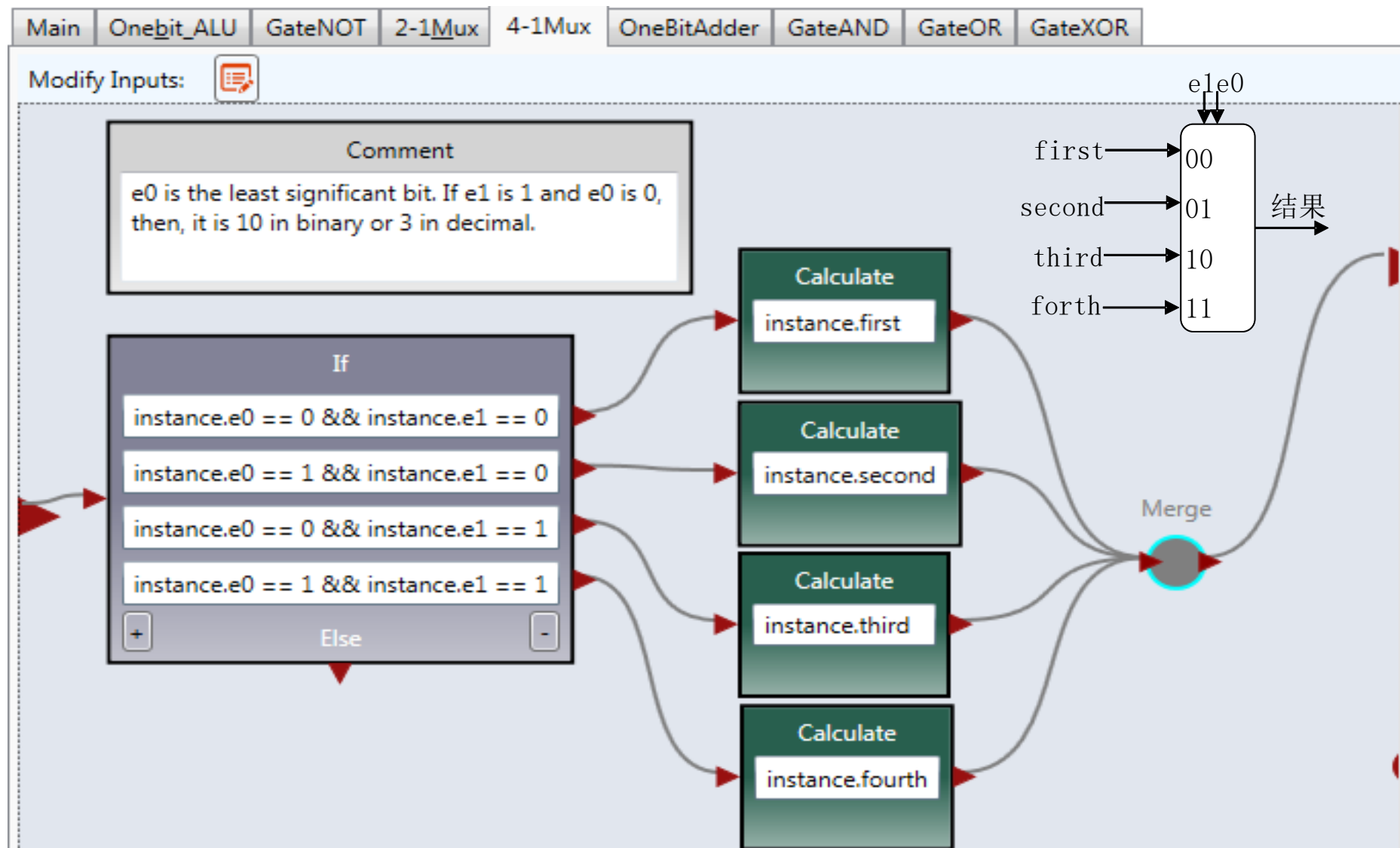
VIPLE 实现



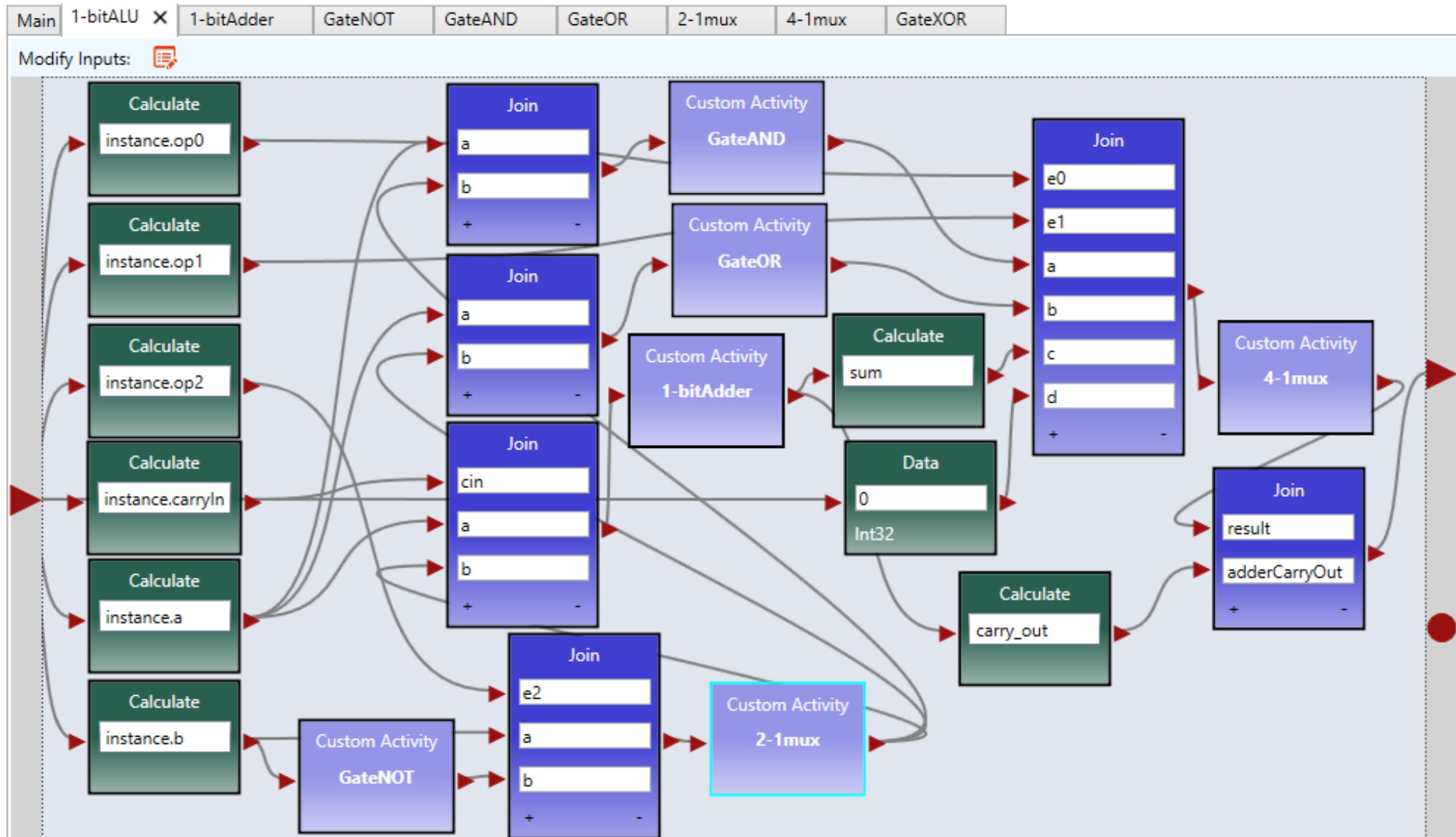
测试1位加法器



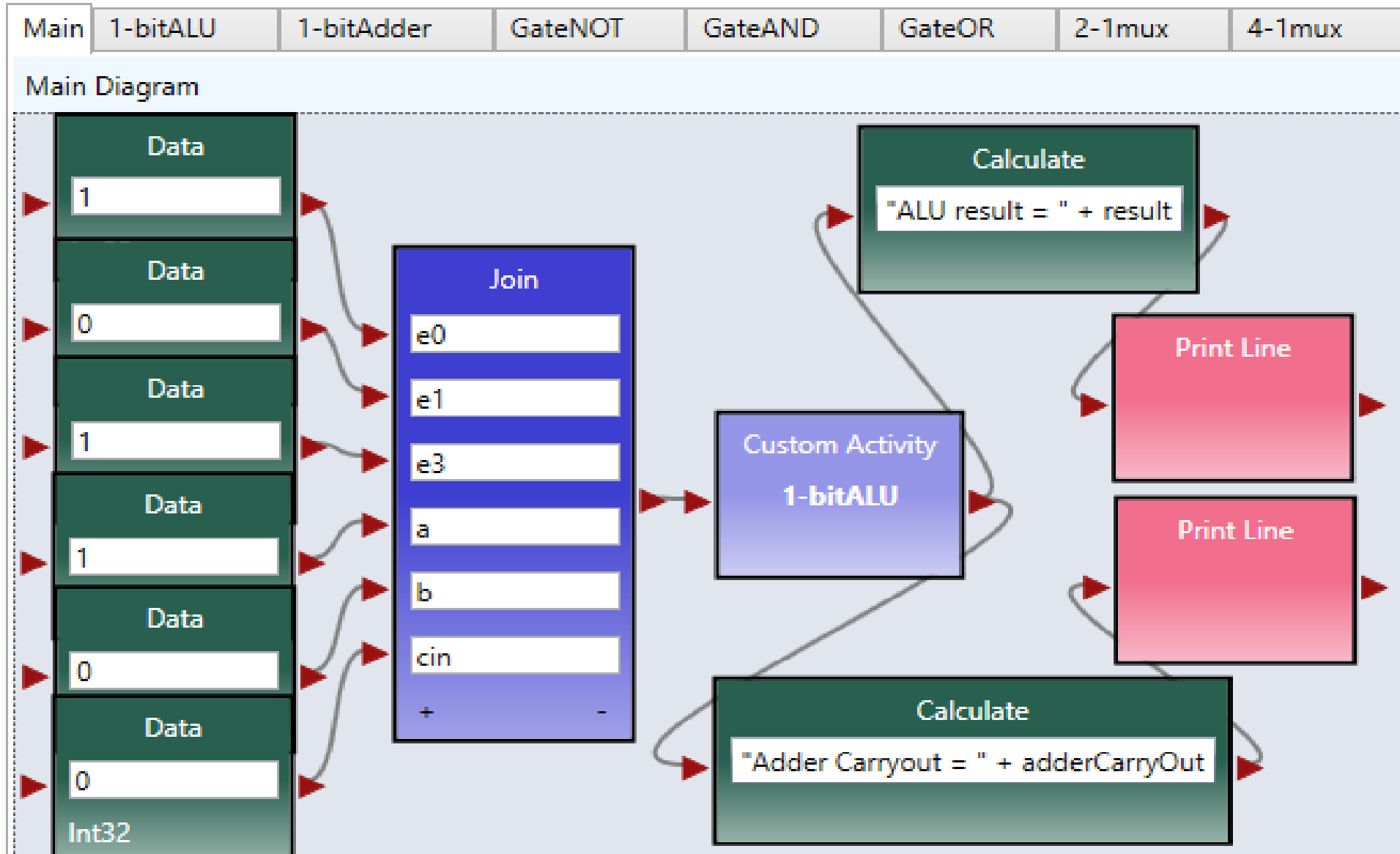
创建 2-1 多路转接器



创建1位ALU



测试1位ALU



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

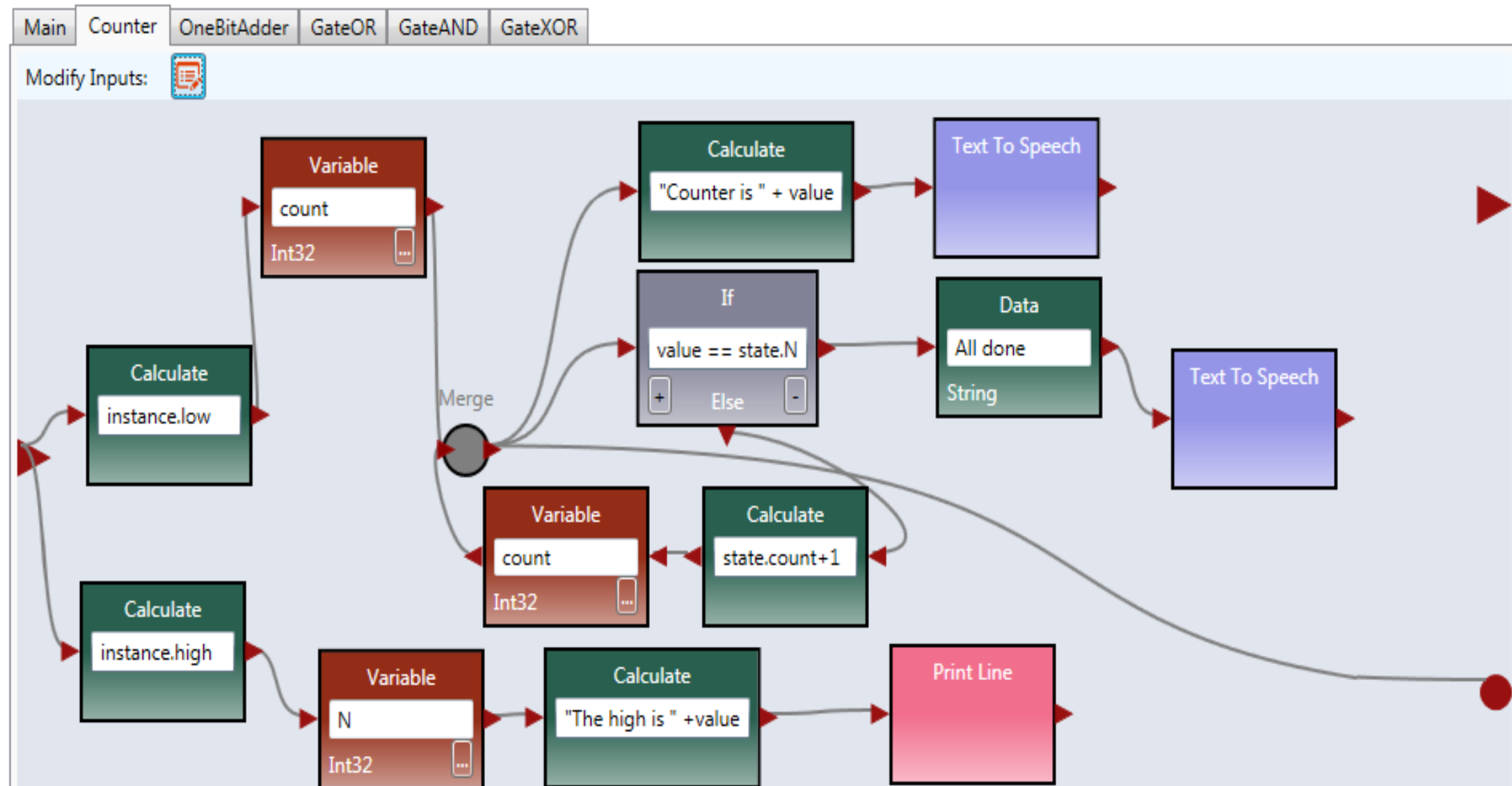
CountTo7	a	b	carryIn
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

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

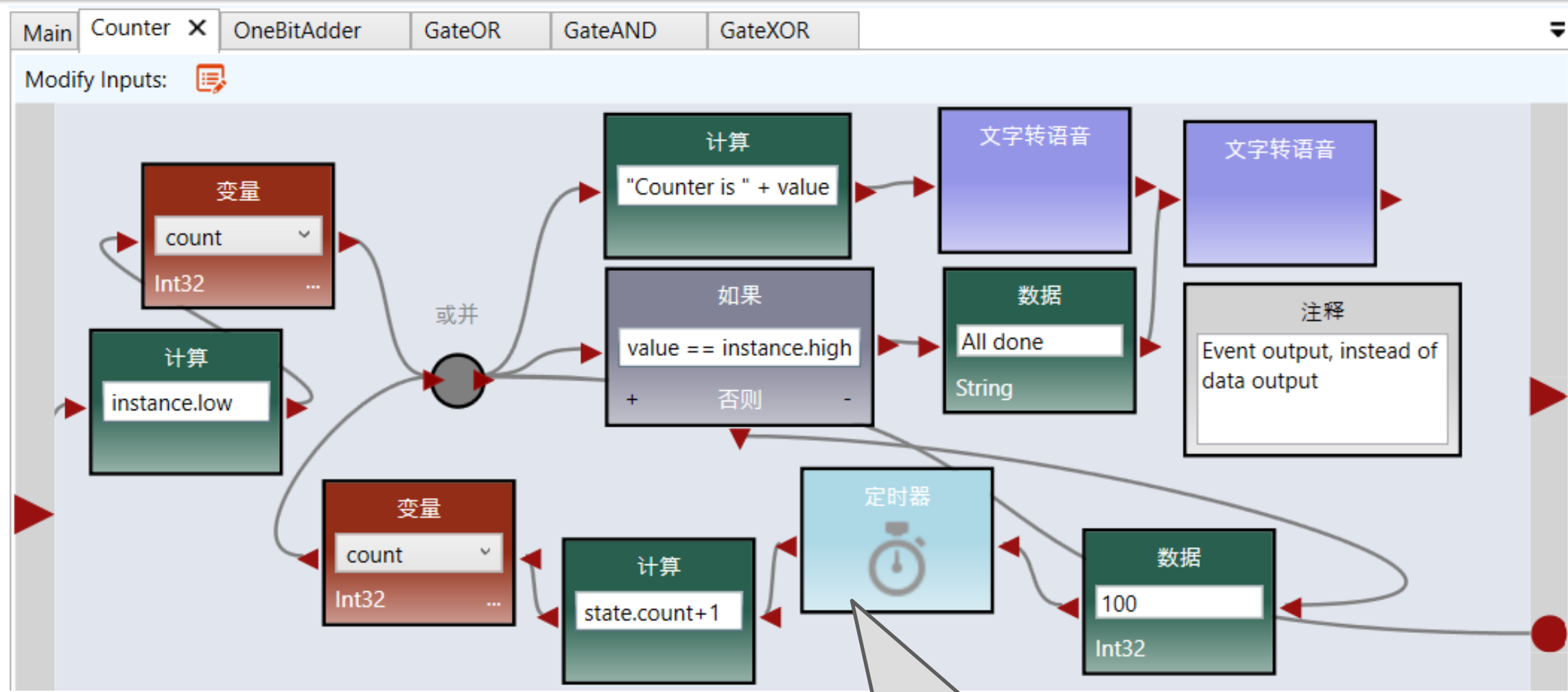
if CountTo7 = 0, 1, 4, 5, then b = 0, else b = 1;

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

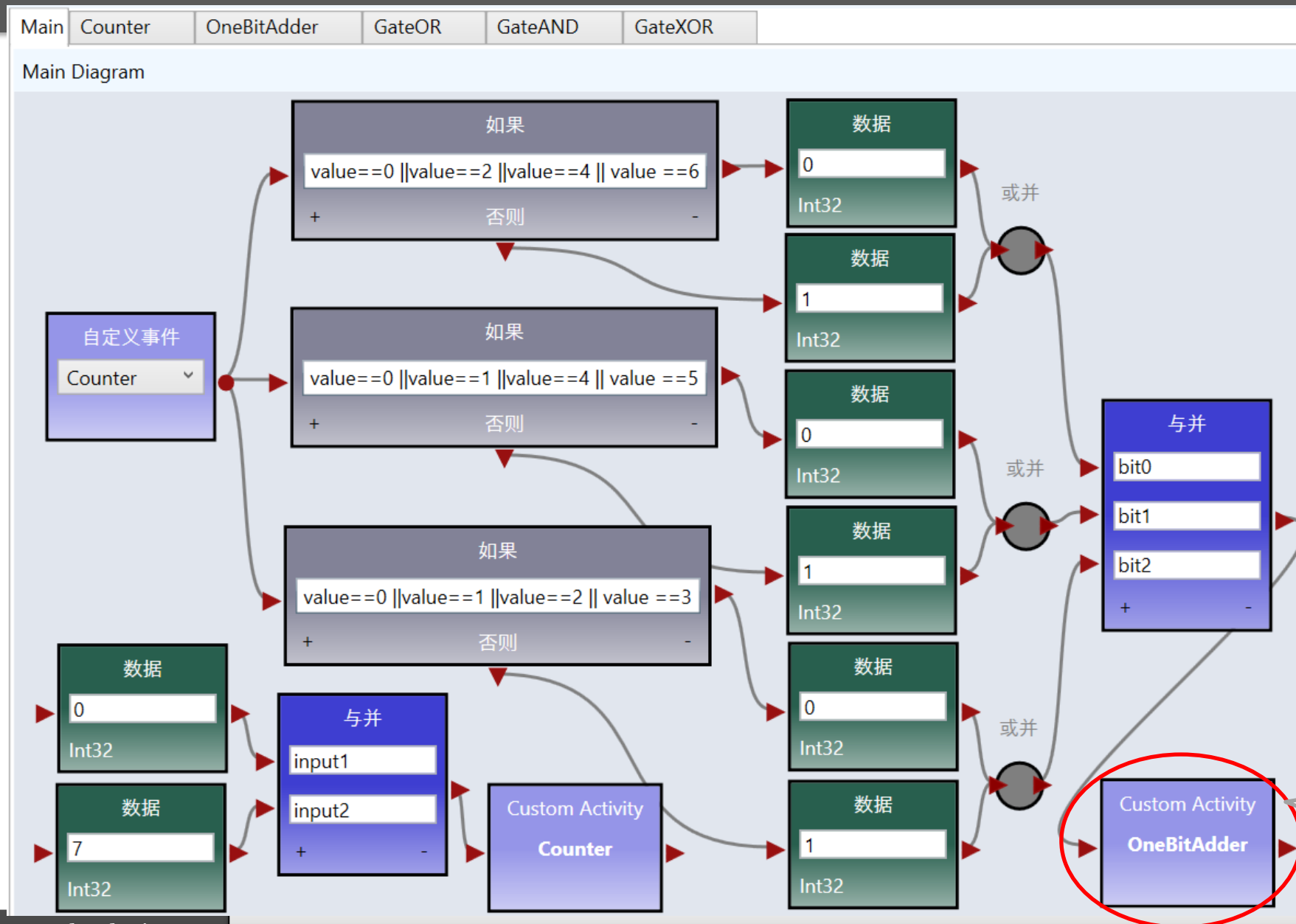


在计数器中添加延迟，以保证计数与加法器同步



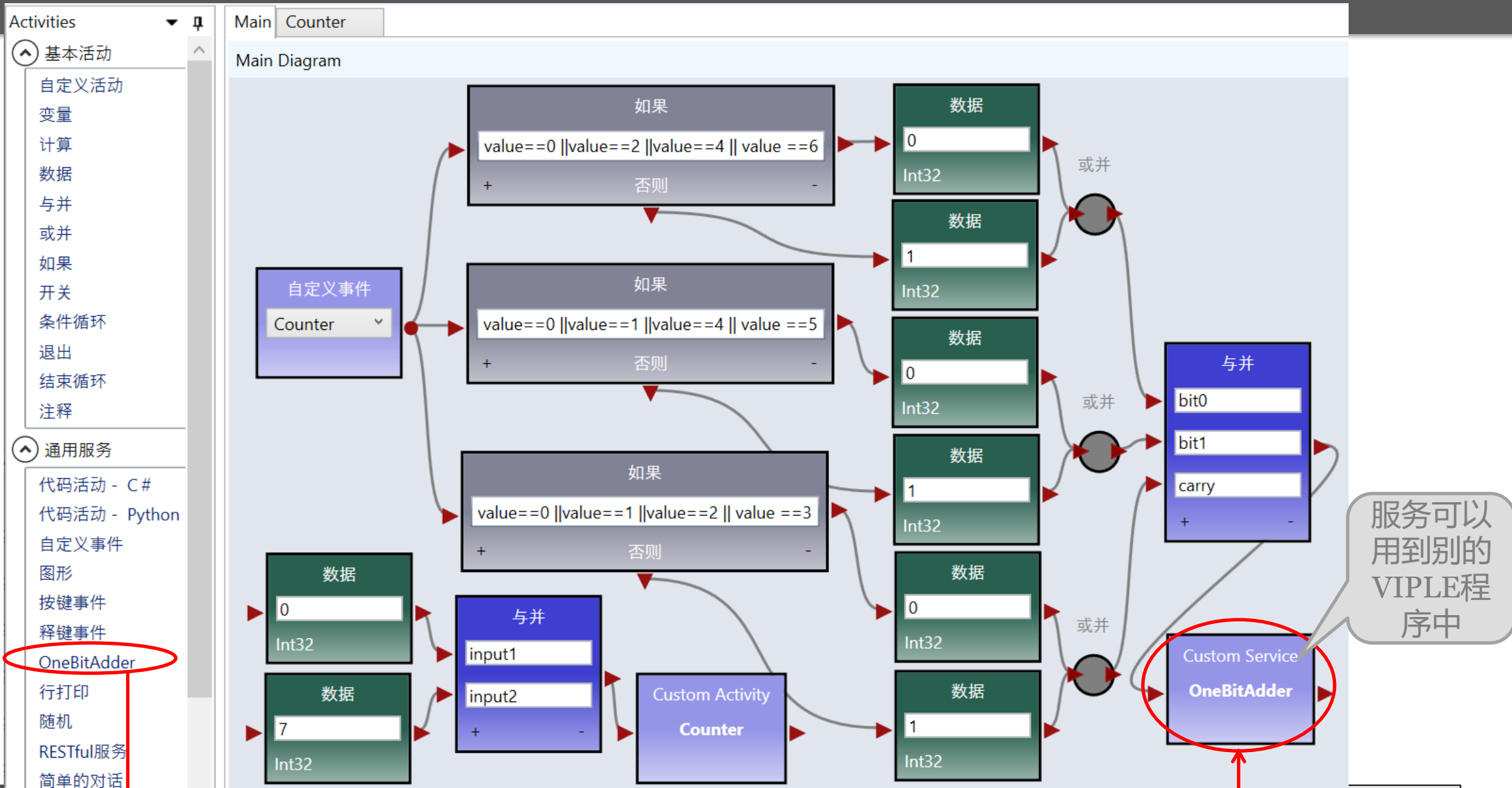
在计数器中添加延迟，以保证计数与加法器同步

1位加法器自动测试（活动）



转换成服务：
对比下一张
幻灯片

1位加法器自动测试（服务）



RISC-V 计算机架构仿真器

Unamed - VIPLE

File Edit Services Run Tools Language Help

Start Start Unity Simulator Start Unity Simulator 2 Start Unity Traffic Simulator Start TORCS Start Web 2D Simulator Start Web 3D Simulator Start RISC-V Simulator

Activities Basic Activities Custom Activity Variable Calculate Data Join

Select an example 127.0.0.1 8222 WebSocket Open Close

Animation Speed 10

Reset Run Step Fetch Decode ALU Compare Mem/Reg PC

Memory C Code

Address	0	1	2	3	Instructions
00000000	93	00	00	02	addi x1, x0, 32
00000004	37	01	00	c0	lui x2, 0xc0000
00000008	83	c1	00	00	lbu x3, 0(x1)
0000000c	63	88	01	00	beq x3, x0, +16
00000010	23	00	31	00	sb x3, 0(x2)
00000014	93	80	10	00	addi x1, x1, 1
00000018	6f	f0	1f	ff	jal x0, -16
0000001c	6f	00	00	00	jal x0, 0
00000020	48	65	6c	6c	-
00000024	6f	00	00	00	jal x0, 0
00000028	00	00	00	00	-
0000002c	00	00	00	00	-
00000030	00	00	00	00	-

Program counter pc 00000000 mepc 00000000 pc+4 00000004

Bus addr - data - irq false

Text I/O 0 1 b0000000 (ctrl, data) 00 00

Instruction reg. instr - fn - rs1 - rs2 - rd - imm -

ALU op - a - b - r -

Comparator op - a - b - taken -

General-purpose regs x0 00000000 x1 00000000 x2 00000000 x3 00000000 x4 00000000 x5 00000000 x6 00000000 x7 00000000 x8 00000000 x9 00000000 x10 00000000 x11 00000000 x12 00000000 x13 00000000 x14 00000000 x15 00000000

VIPLE Program

Main Diagram

Robot/IoT Controller 0 Data false Variable connected Boolean

Robot/IoT Message In Data true Variable connected Boolean

#include "printf.h" void main(void) { printf("Hello Viple\n"); int i; for(i = 0; i <= 5; i++) { printf("%d\n", i); } }

Print Line

Calculate "hex\n" + state.hexFile

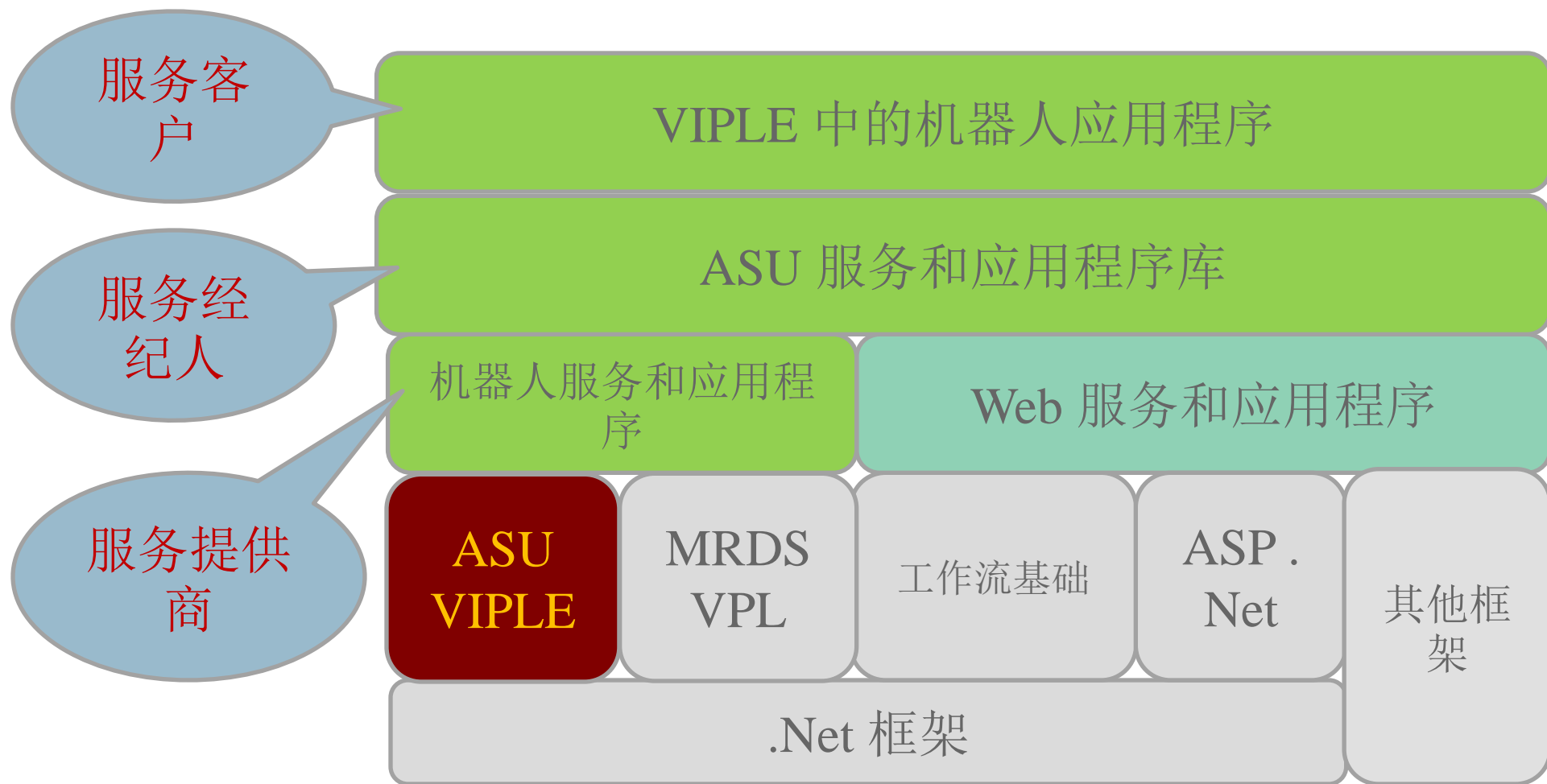
Robot/IoT Message Out

RISC-V 计算机架构仿真器

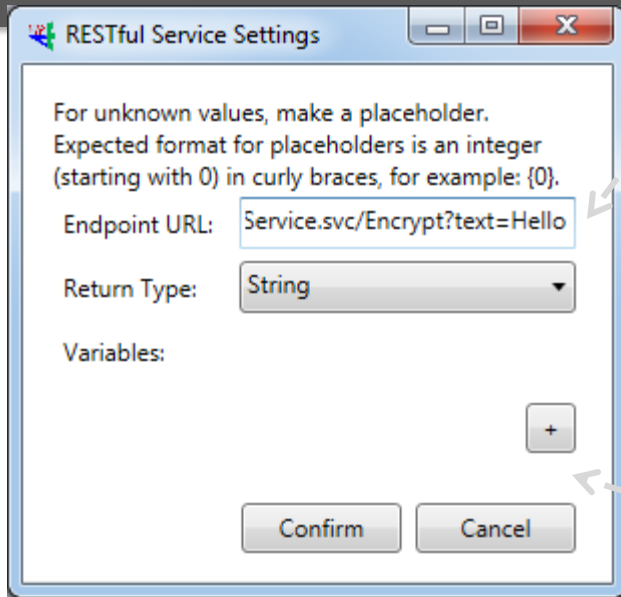
WebSocket

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

VIPLE 面向服务



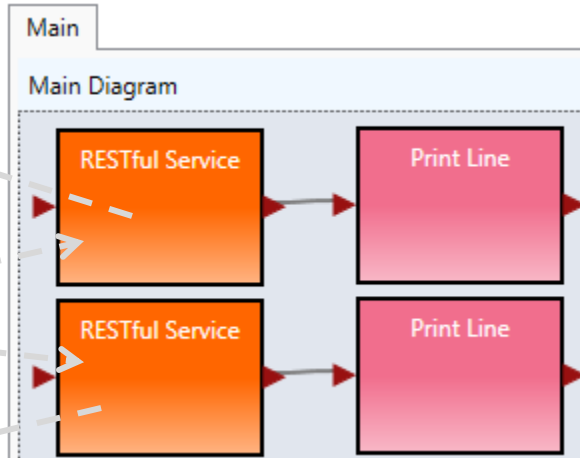
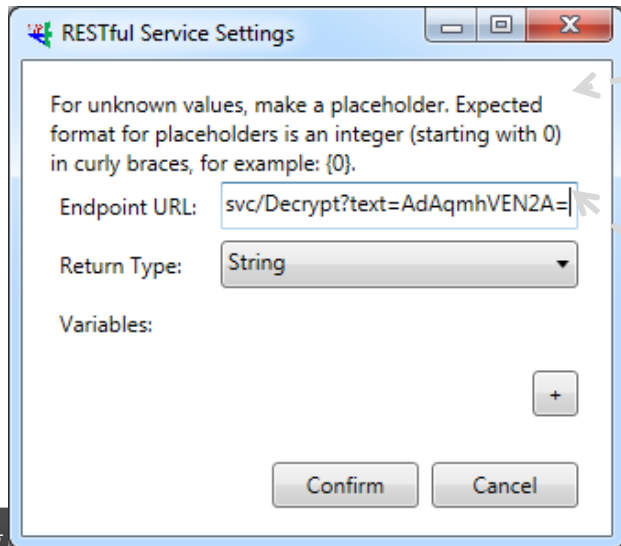
调用 RESTful 服务(Web API)



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

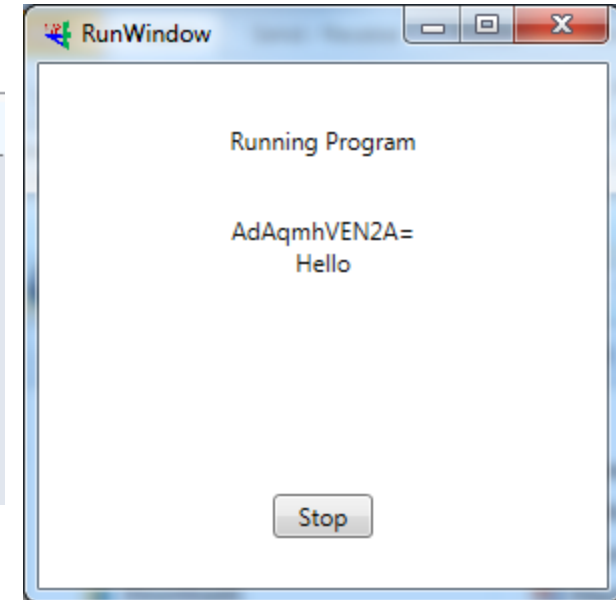
服务地址，参数值 = Hello，被调用

RESTful Service

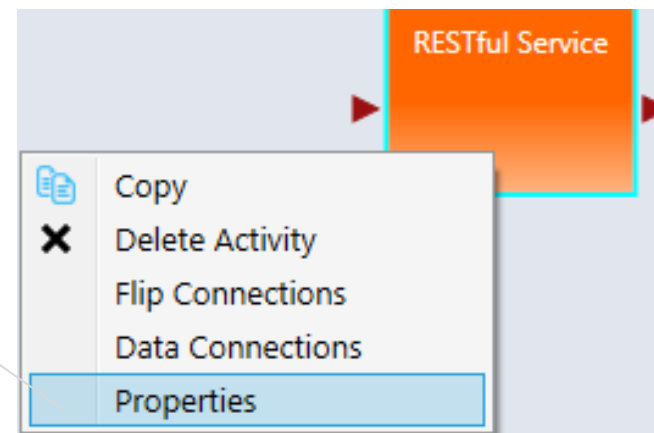
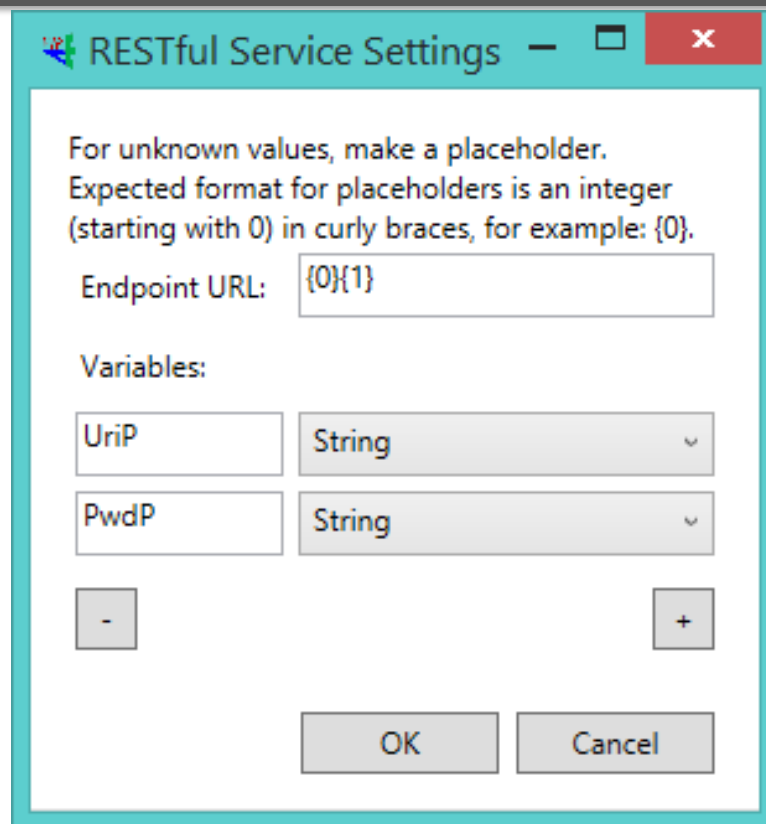


<http://venus.sod.asu.edu/WSRepository/Services/EncryptionRest/Service.svc/Decrypt?text=AdAqmhVEN2A=>

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>

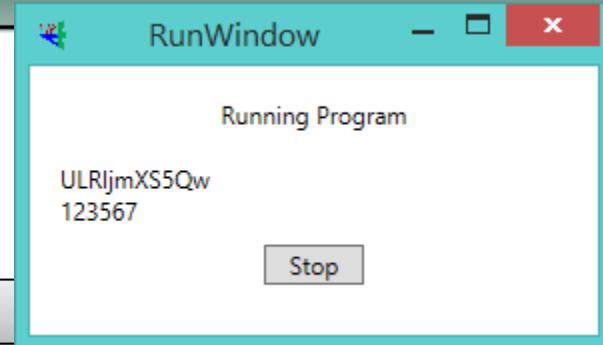
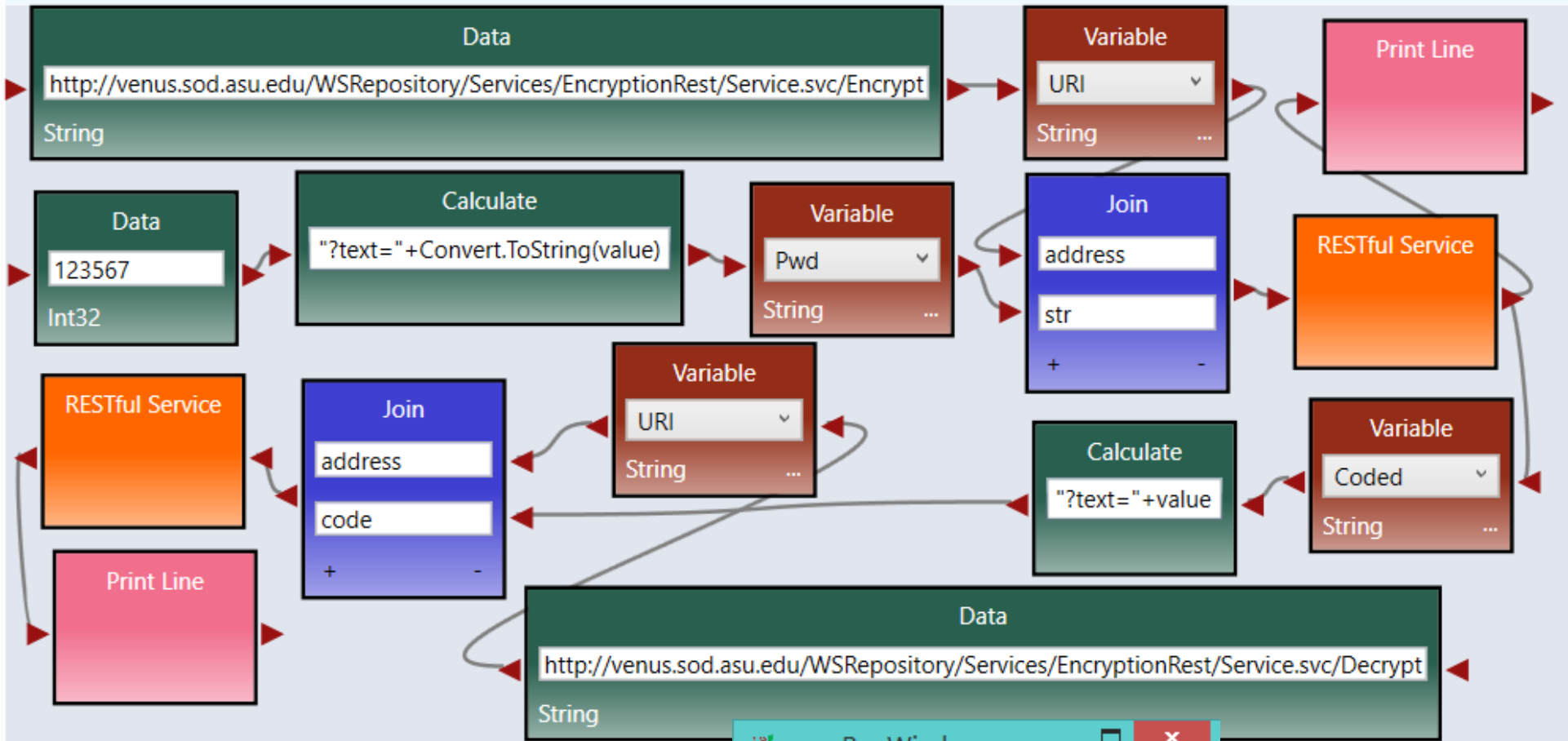
基础地址

{0}

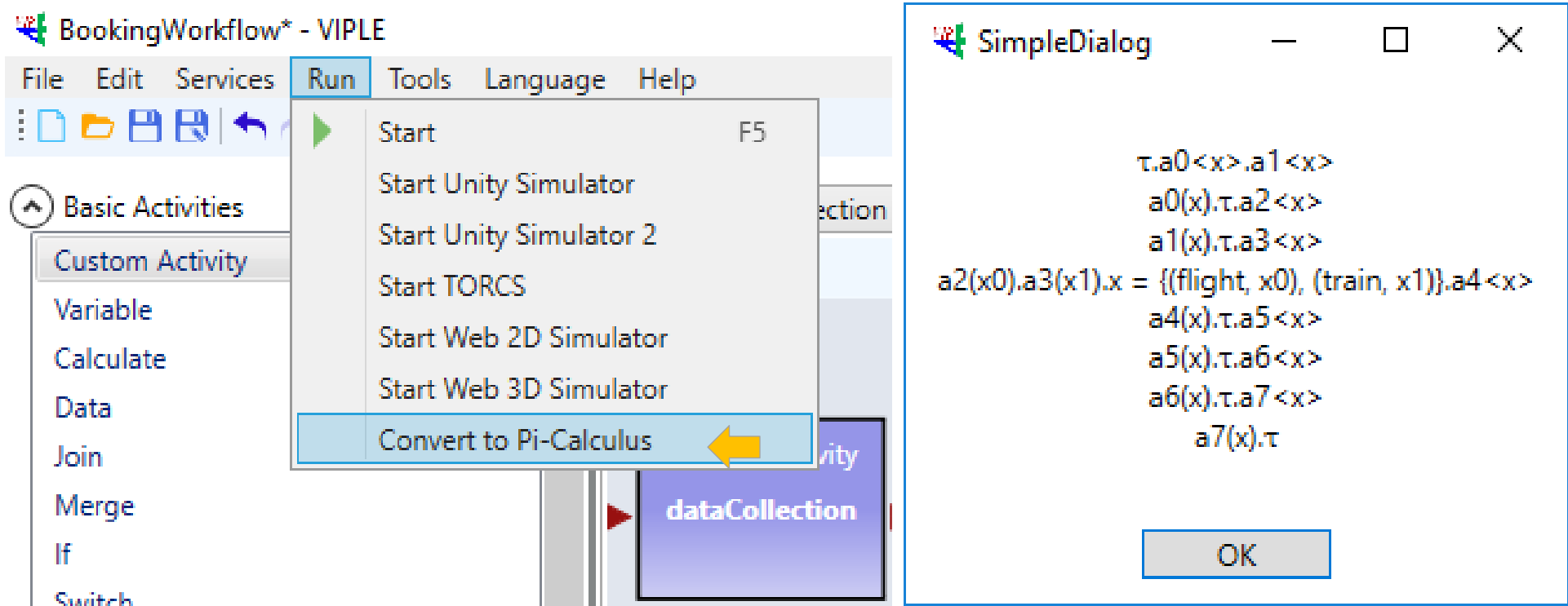
参数值

{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

The image illustrates the steps to add a Code Activity to a workflow in Visual Studio:

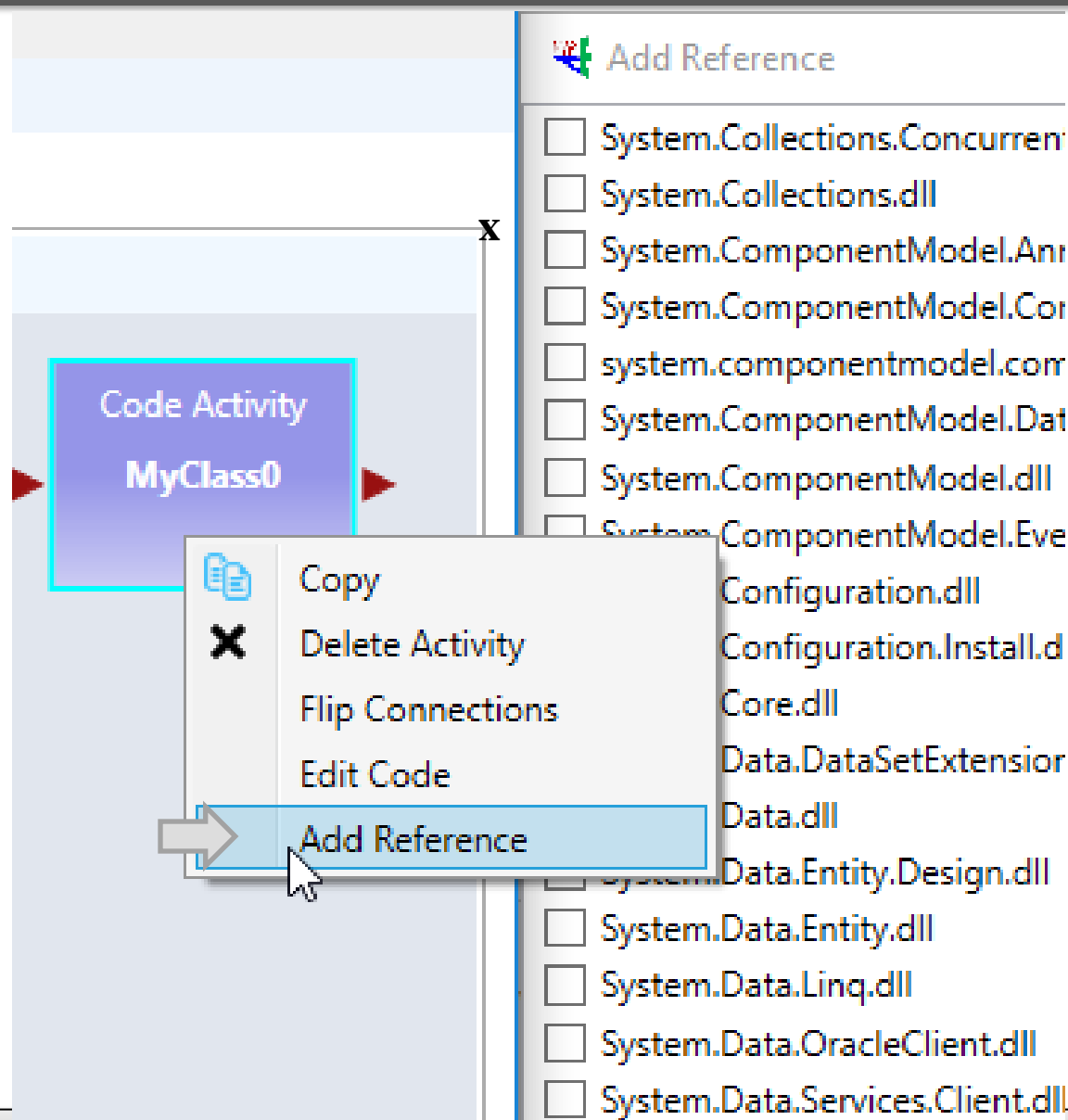
- General Purpose Services**: A list of services is shown on the left. **Code Activity - C#** is selected and circled in red. An arrow labeled ① points from this selection to the next step.
- Code Activity**: A purple block labeled "Code Activity" is added to the workflow. An arrow points from this block to the next step.
- Input Class Name**: A dialog box appears with the text "Please enter a class name." and a text box containing "MyCodeActivity". Buttons for "OK" and "Cancel" are at the bottom.
- Code Editor**: The code editor shows the implementation of the `NOT_Gate` class. The code is as follows:

```
1 using System;
2
3 [Serializable]
4 public class NOT_Gate : CodeUtilities.CodeBase
5 {
6     public NOT_Gate()
7     {
8     }
9 }
10
11 // To execute your code, you must override the Execute method.
12 public override void Execute()
13 {
14     // Obtain the value of the input to this activity.
15     // The type of this value will depend on what input you pass to this activity.
16     int myInput = (int)Input;
17
18     // You can use the PrintLine method to print strings to the run window
19     // during runtime.
20     PrintLine("Hello from NOT_Gate!");
21
22     // You can pass output in a similar way.
23     if (myInput == 0)
24     { Output = 1; }
25     else
26     { Output = 0; }
27 }
28 }
```

Below the workflow, a context menu is shown for the **Code Activity MyCodeActivity** block. The menu options are: Copy, Delete Activity, Flip Connections, Edit Code (highlighted with a grey arrow and labeled ③), and Add Reference.

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.dll** 添加到你的程序中



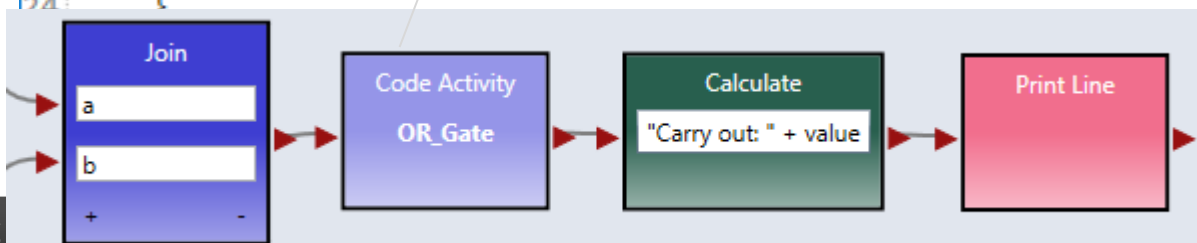
CodeActivity 示例

```
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    }
25 }
```

用于在 CodeActivity 中使用 Join 取多个值

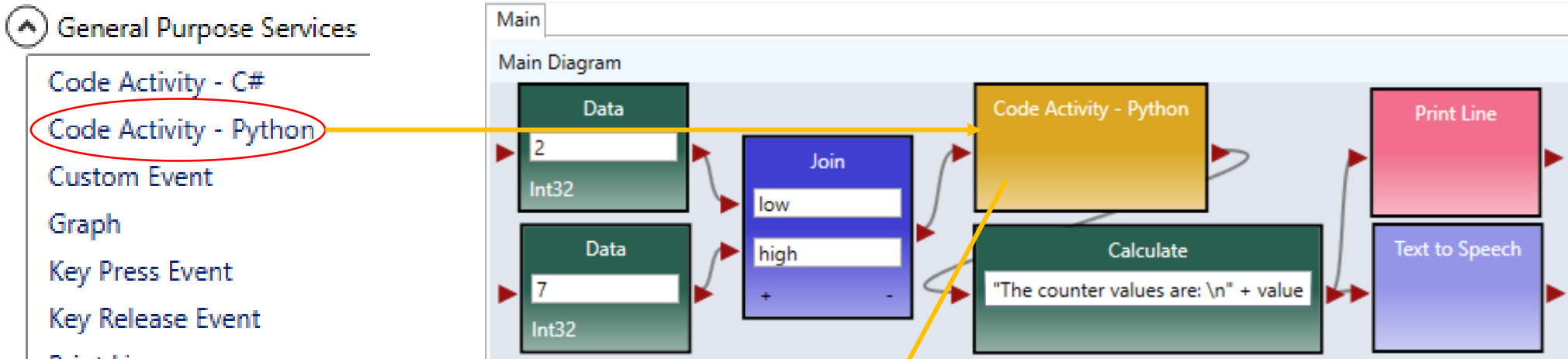
如果只有一个输入，我们不使用 Join。在这一情况下，我们使用：
`int x = (int)Input`

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")
```

