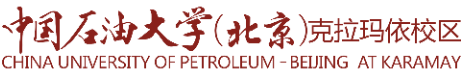
实 验 报 告





课 程 ：计算机组成原理

实验名称：1.可控加减法电路设计

2.4位快速加减法设计（选做）

3.32位ALU设计

学 院 ：石油学院

专 业 ：数据科学与大数据技术

学 号 ：2023015509

姓 名 ：胡林森

2024年秋季学期

目录

[一、实验目的 3](#_Toc181824435)

[ 1.可控加减法电路设计 3](#_Toc181824436)

[ 2.4位快速加减法设计（选做） 3](#_Toc181824437)

[ 3.32位ALU设计实验 3](#_Toc181824438)

[二、实验内容 3](#_Toc181824439)

[ 1.可控加减法电路设计 3](#_Toc181824440)

[ 2.4位快速加减法设计（选做） 4](#_Toc181824441)

[ 3.32位ALU设计实验 5](#_Toc181824442)

[三、实验步骤 6](#_Toc181824443)

[ 1.可控加减法电路设计 6](#_Toc181824444)

[ 2.4位快速加减法设计（选做） 6](#_Toc181824445)

[ 3.32位ALU设计实验 6](#_Toc181824446)

[四、实验结果 6](#_Toc181824447)

[ 1.可控加减法电路设计 6](#_Toc181824448)

[ 2.4位快速加减法设计（选做） 7](#_Toc181824449)

[ 3.32位ALU设计实验 8](#_Toc181824450)

[加操作00 8](#_Toc181824451)

[乘操作01 9](#_Toc181824452)

[比较操作10 9](#_Toc181824453)

[右移操作11 10](#_Toc181824454)

[整体电路 10](#_Toc181824455)

[五、问题和感想 11](#_Toc181824456)

# 一、实验目的

## 1.可控加减法电路设计

- 掌握1位全加器的实现逻辑，掌握多位可控加减法电路的实现逻辑；

- 通过实验尽快熟悉Logisim平台基本功能，能在Logisim中实现多位可控加减法电路。

## 2.4位快速加减法设计（选做）

- 掌握先行进位的原理，能利用相关知识设计4位先行进位电路，并利用设计的4位先行进位电路构造4位快速加法器。

## 3.32位ALU设计实验

- 理解算术逻辑运算单元（ALU）的基本构成，掌握Logisim中各种运算组件的使用方法，熟悉多路选择器的使用，能利用Logisim中的运算组件构造指定规格的ALU。

# 二、实验内容

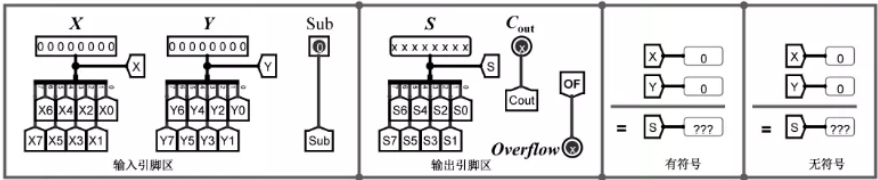
## 1.可控加减法电路设计

- 利用Logisim打开实验资料包中的alu.circ文件，在对应子电路中利用已封装好的全加器设计8位串行可控加减法电路，其电路封装与引脚功能描述如表2.1所示。



- 8位串行可控加减法电路框架如图2.1所示，请在相应电路中使用对应引脚的隧道标签实现相应的加减法控制逻辑。注意这里X、Y、S引脚的每一位均用分线器连接到了对应隧道标签，绘图时可灵活复制使用。



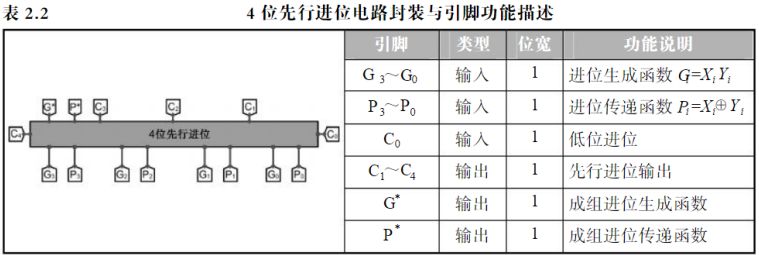


- 8位串行可控加减法电路是组合电路，设计完成后可以利用手形戳工具单击输人引脚，设置输入引脚的值，尝试设置不同的输入组合，观察探测区运算结果是否正确。

## 2.4位快速加减法设计（选做）

1)设计4位先行进位电路

在Logisim中打开alu.circ文件，在对应子电路中利用基本逻辑门实现可级联的4位先行进位电路，其电路封装与引脚功能描述如表2.2所示。

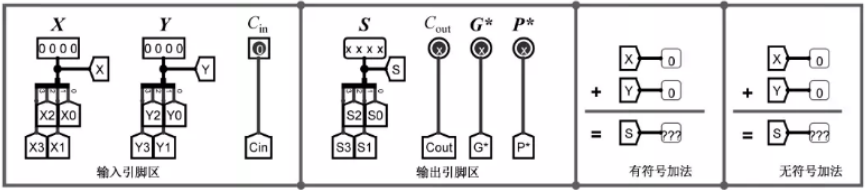


2)设计4位快速加法器

在4位快速加法器子电路中利用已经设计好的4位先行进位电路以及基本逻辑门实现4位快速加法器，其电路封装与引脚功能描述如表2.3所示。



4位快速加法器电路框架如图2.2所示，请在相应电路中使用对应引脚的隧道标签实现相应的逻辑。注意X、Y、S引脚的每一位均用分线器连接到了对应隧道标签，绘图时可灵活使用。图2.2右侧探测区利用探针组件直观显示了有符号加法和无符号加法的运算结果，方便观察测试



3)功能测试

先行进位电路和快速加法器电路都是组合电路。设计完成后可以利用手形戳工具单击输入引脚，设置输入引脚的值，尝试设置不同的输人组合，观察探测区运算结果是否正确。

## 3.32位ALU设计实验

- 打开Logisim，构建一个32位算术逻辑运算单元

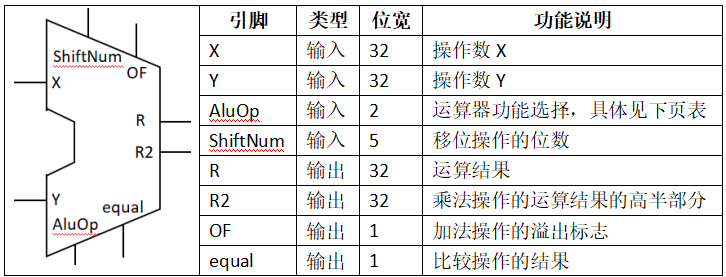
• 支持算术加、乘、相等比较、以及右移等四种操作；

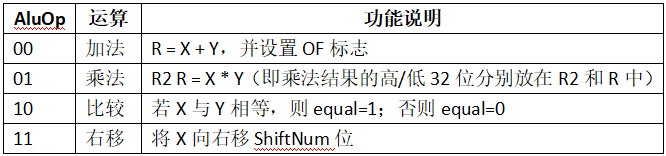
• 可以利用Logisim平台中已有运算部件，如加法器、乘法器、比较器、移位器、多路选择器等；

• 无需封装，但要有所有要求的引脚，这些引脚包括：

♦ 4个输入：X，Y，AluOp，ShiftNum；

♦ 4个输出：R，R2，OF，equal。





**- 功能测试**

• 分别设置AluOp为00、01、10、11，并展示进行相应的运算后，ALU给出的结果是正确的。要求用截图展示结果，并辅以文字说明。

# 三、实验步骤

## 1.可控加减法电路设计

利用已经封装好的全加器设计8位串行可控加减法电路，其电路引脚定义如图所示，用户可以直接使用在电路中使用对应的隧道标签，其中 X，Y 为两输入数，Sub 为加减控制信号，S 为运算结果输出，Cout 为进位输出，OF 为有符号运算溢出位

## 2.4位快速加减法设计（选做）

在当机器数数位是确定的时候，我们可以用一种复制的计算算出每一位的进位，无需等待低位给出进位信号，因此出现了先行进位电路。

把Pi和Gi对应的值通过与运算和异或运算Xi和Xi的值送到CLA74182中，然后就可以通过74182的输出的进位值直接的计算出每一位二进制数对应的值采用异或运算得到

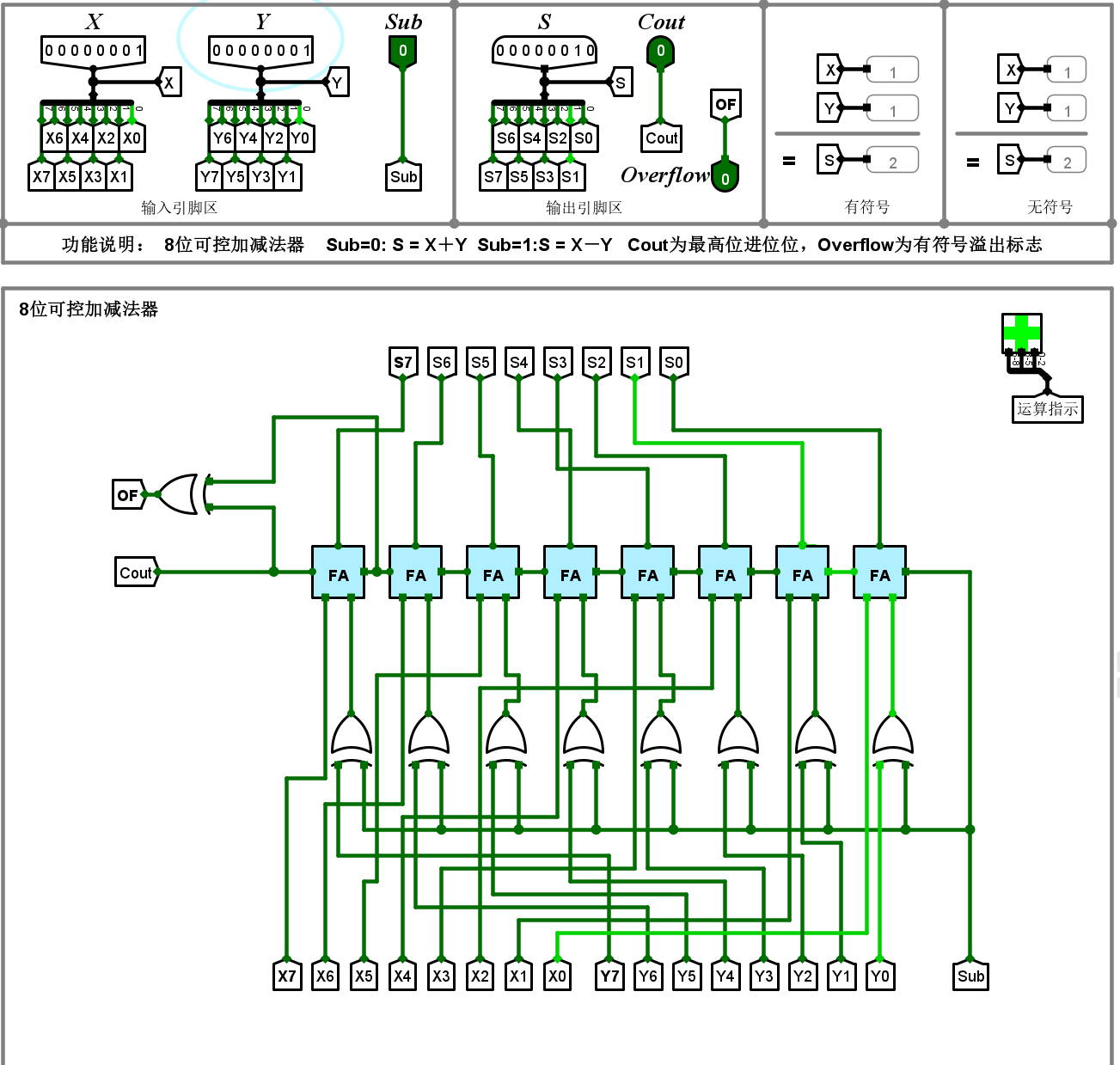
## 3.32位ALU设计实验

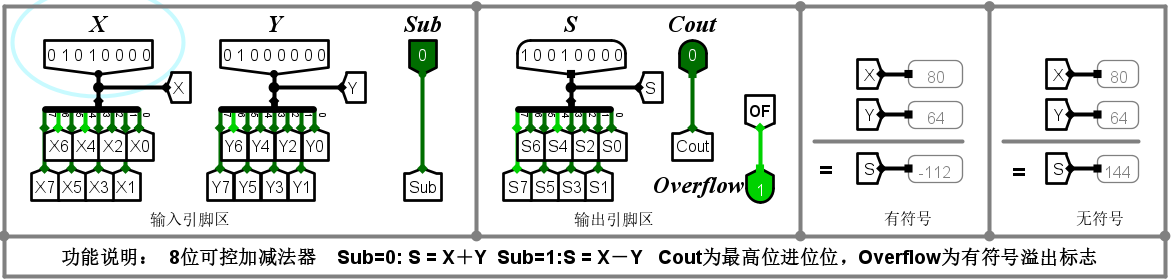
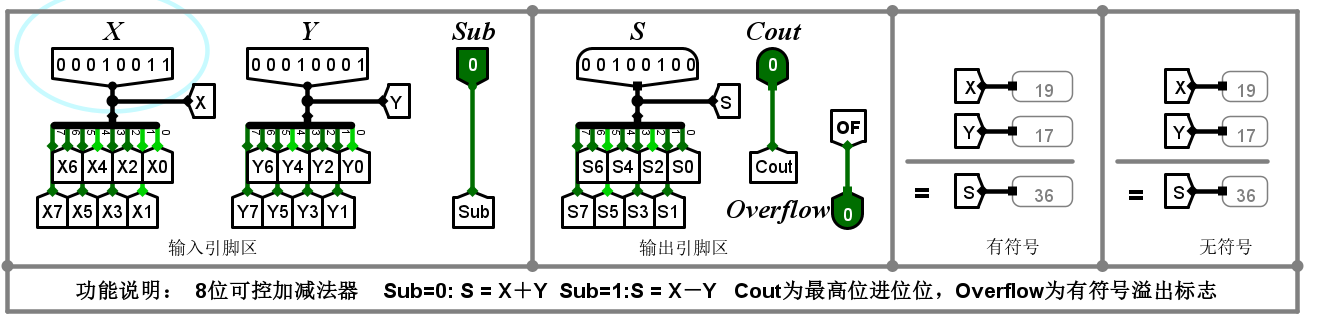
利用封装好的4位快速加减法设计中的4位加法器组建32位加法器；使用分线选择器、加法器、乘法器、移位器等，连接电路，构建好各OP值所对应的操作，最后用选择器来选择操作、连接输出就可。输入数据进行调试。

# 四、实验结果

## 1.可控加减法电路设计

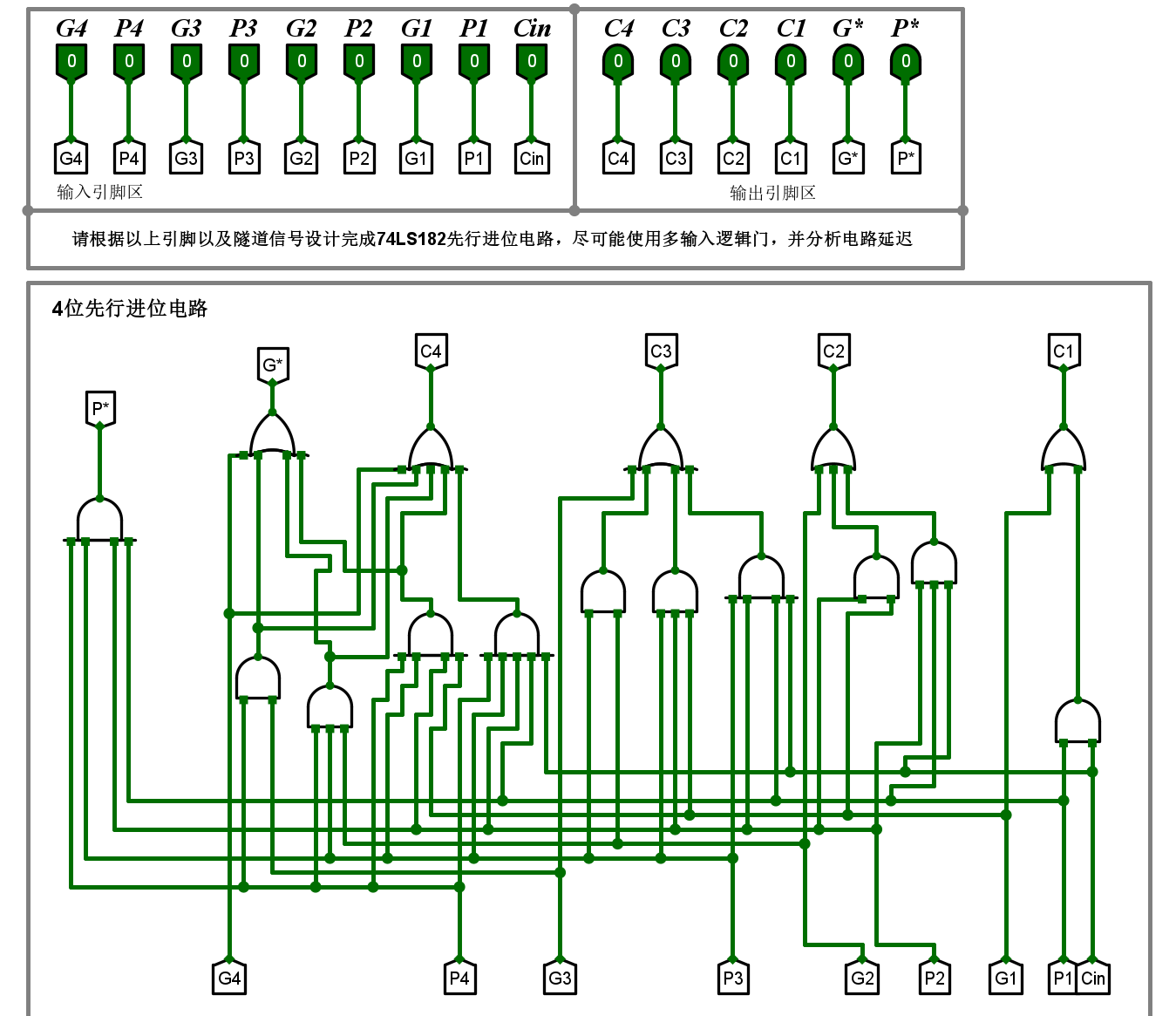
8位可控加减法：输入X=00000001，Y=00000001,输出2，正确；输入X=00010011，Y=00010001,即19+17，输出36，正确；输入X=01010000，Y=01000000，输出-112(溢出)，溢出指示位为1，符合要求，正确。如下图：



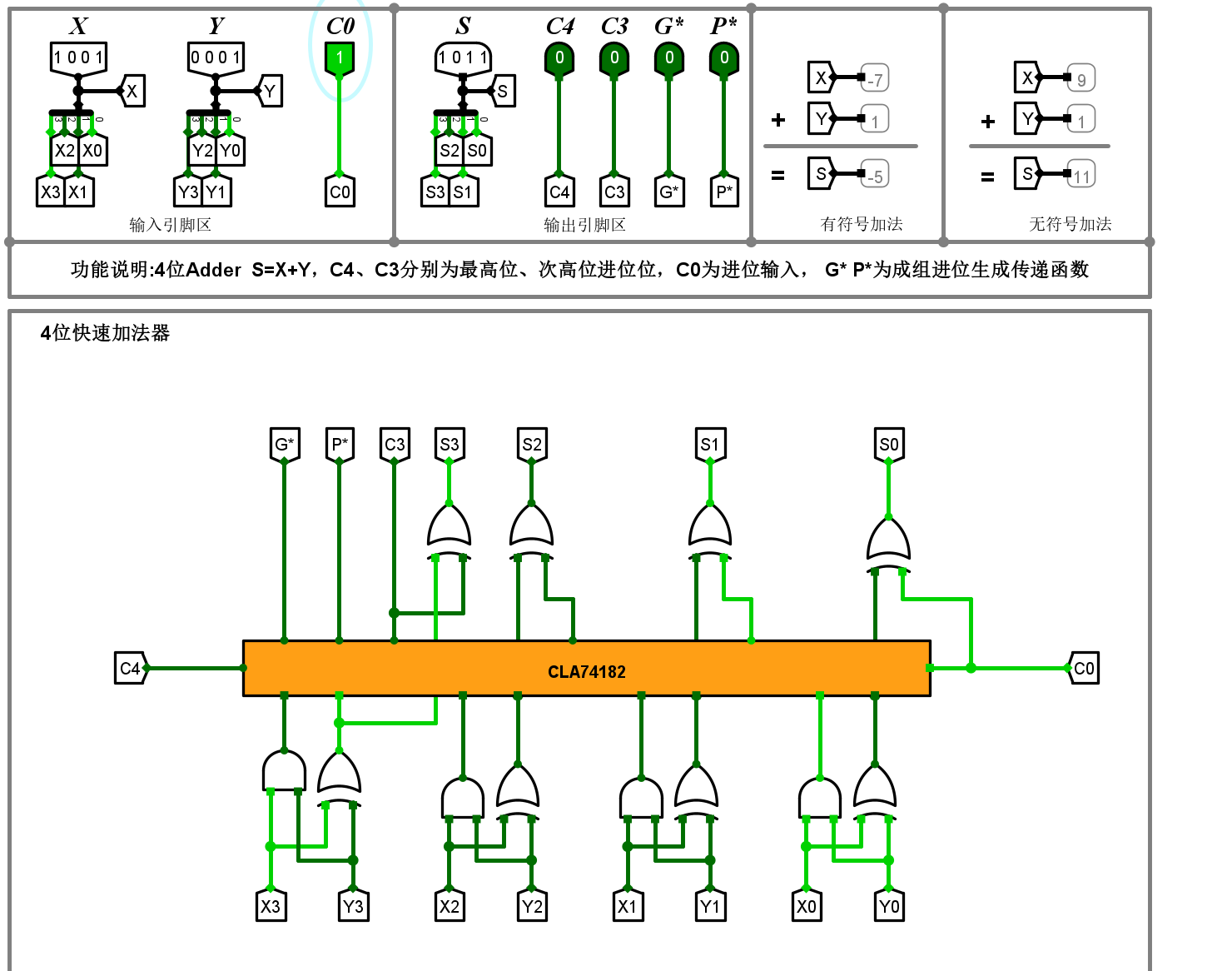


## 2.4位快速加减法设计（选做）

连接好先行进位电路



然后，设X=1001，Y=0001，若为无符号数，这X+Y=1010，即10；若为有符号X+Y是-6，但进位输入为1，所有无符号数时，X+Y=11，有符号时X+Y为-5。正确。



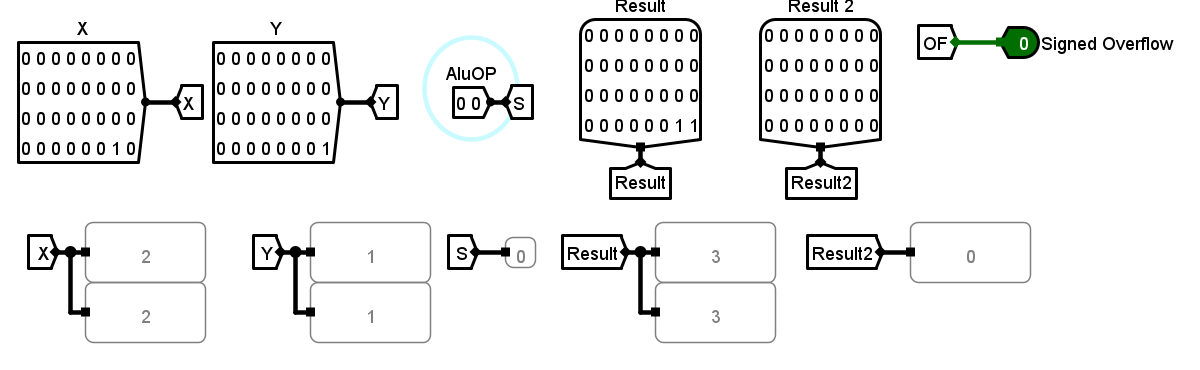
## 3.32位ALU设计实验

### 加操作00

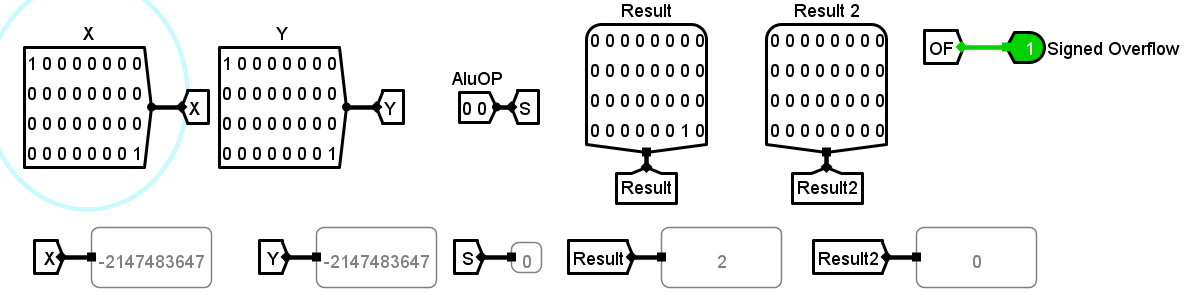
逻辑电路实现了：X+Y；X+Y作为有符号数溢出与否的判断。

可能出现2种情况：无溢出、符号数溢出。

1. 无溢出：如下图，

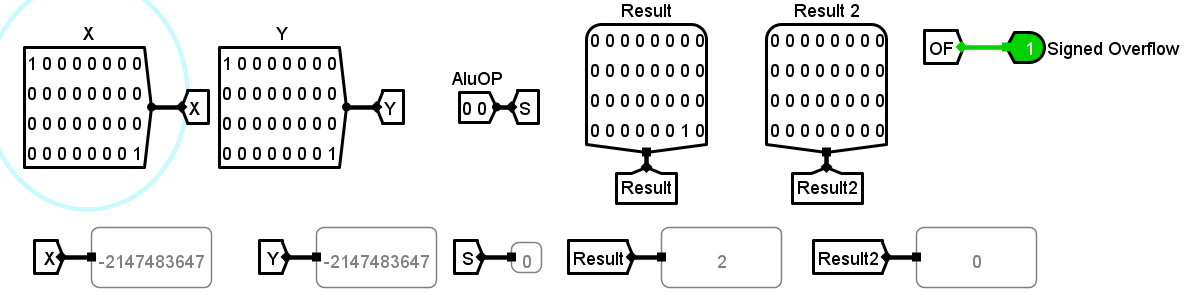


1. 符号数溢出：如下图，输入X=Y=10000….001，OF为1



### 乘操作01

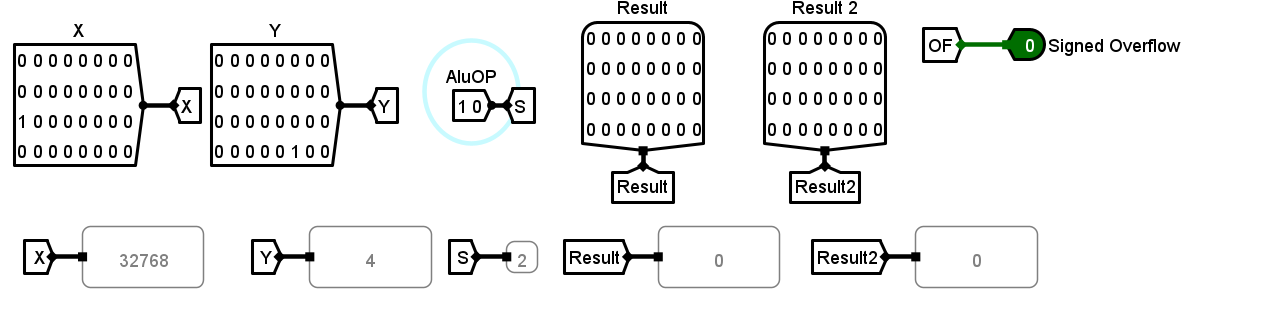
对XY做乘法，输出正确。



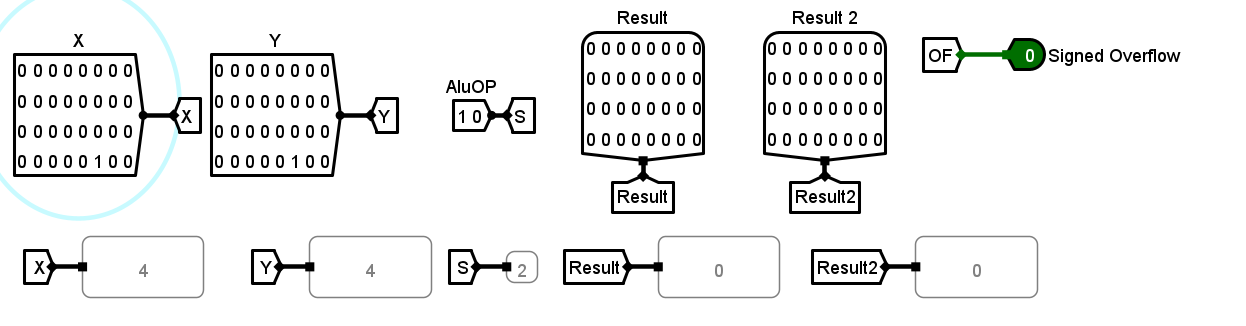
### 比较操作10

当X<Y时输出1，其他情况输出0。

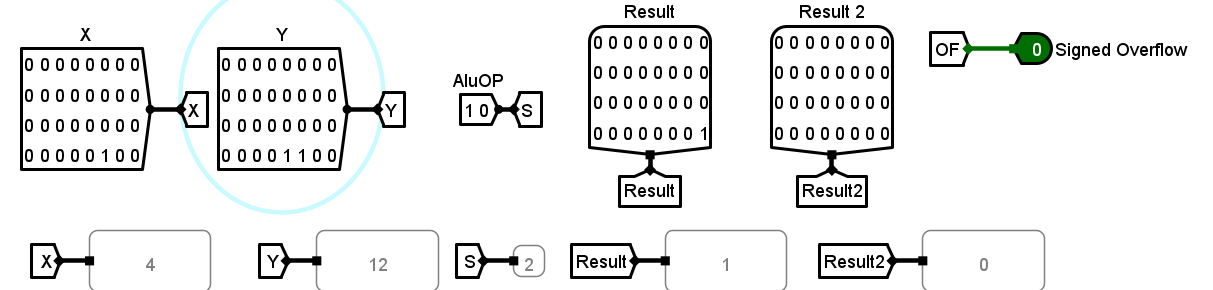
1. X>Y，输出0



1. X=Y，输出0

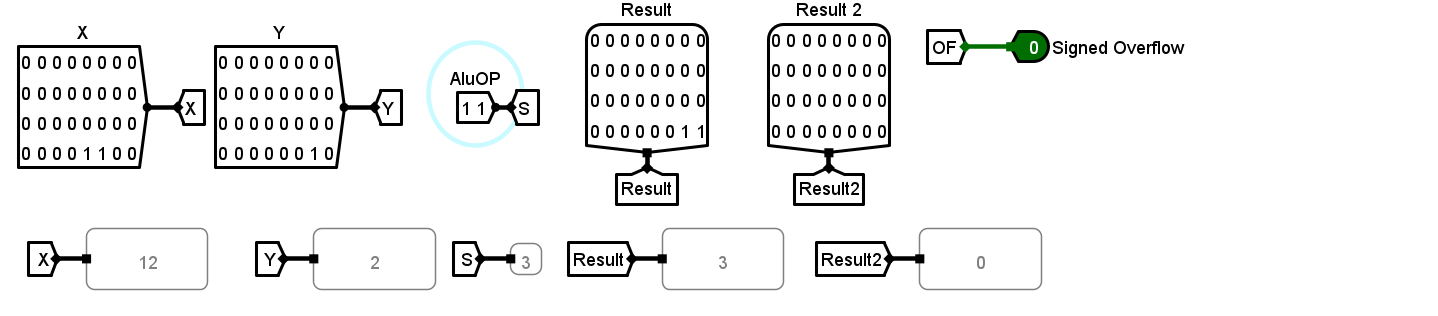


1. X<Y，输出1

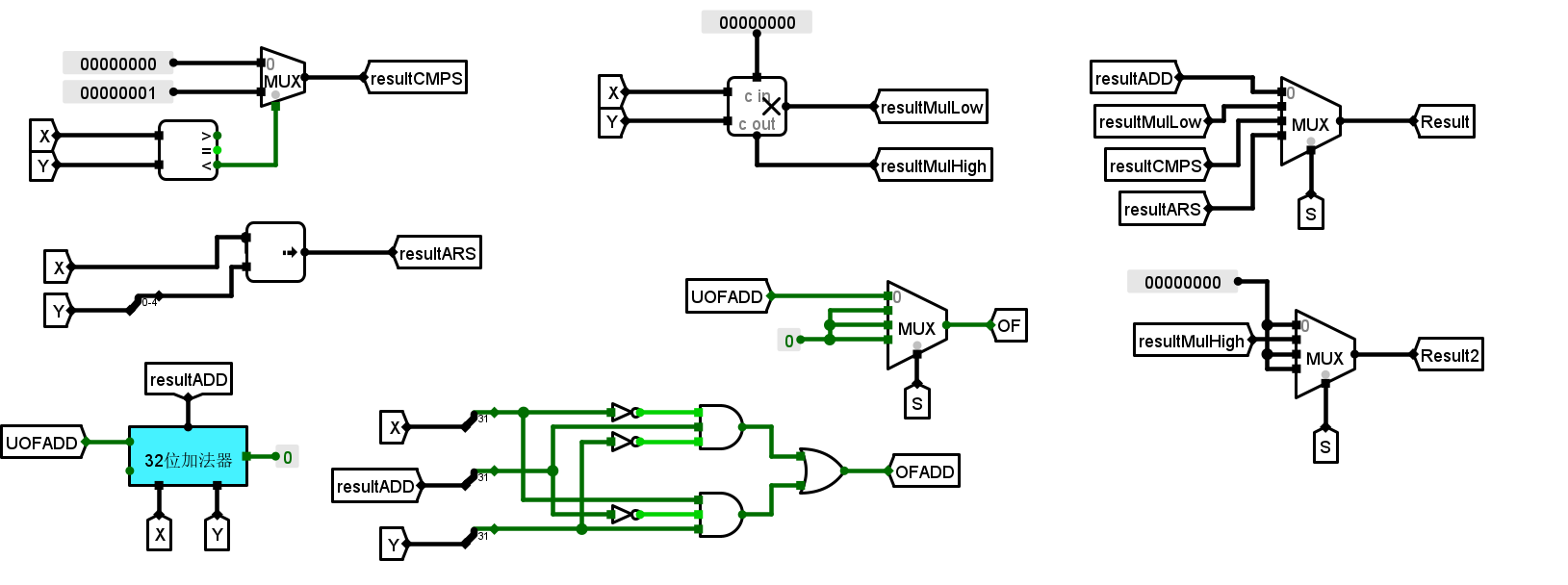


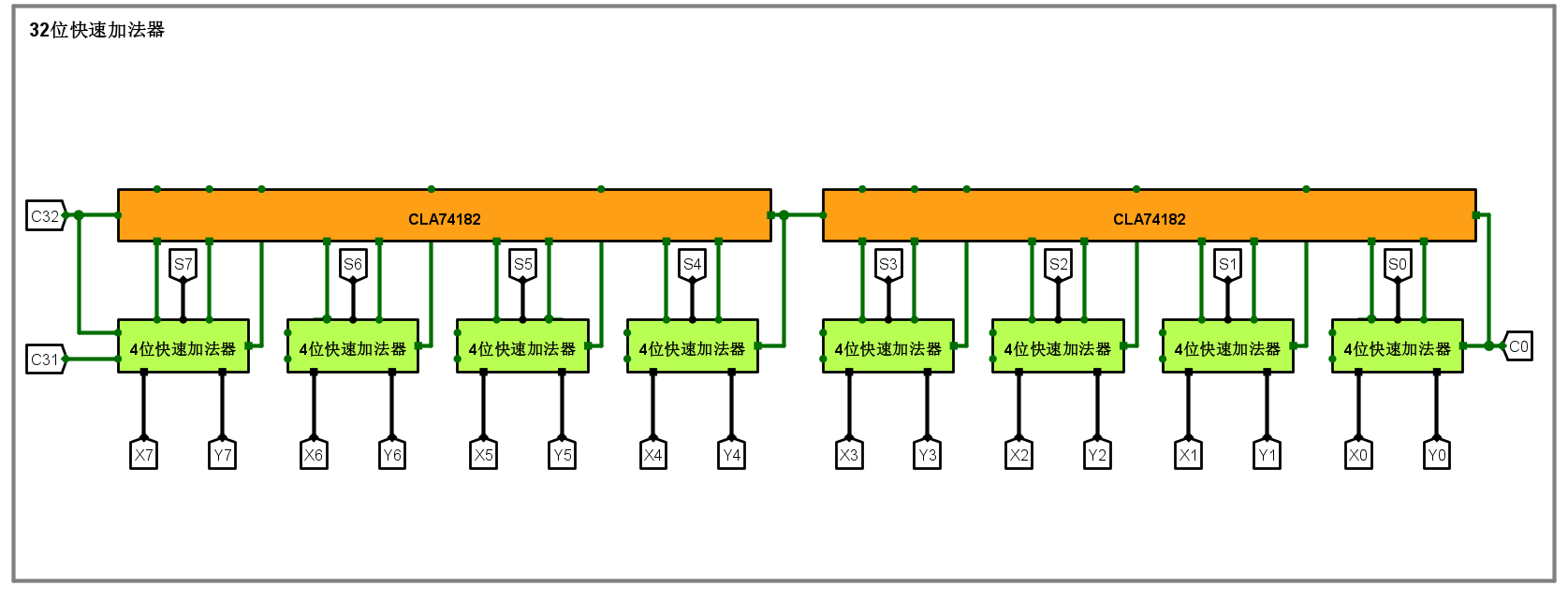
### 右移操作11

输入X=0000……1100,Y=000……10,将X右移3位，输出3正确。

****

### 整体电路





# 五、问题和感想

**问题：**

1. 在进行32位ALU设计时，我最初对Logisim平台中的一些高级功能，如数据隧道（tunnel）的使用不够熟悉。这导致了在设计过程中，我无法有效地利用这些功能来简化电路设计，增加了设计的复杂度。
2. 在构建多路选择器（MUX）时，我未能正确地将所有未用到的输入位编码为0，这导致了输出结果的不准确。这个问题在初步测试时并未立即显现，但在进行更复杂的运算时，错误的影响变得明显。
3. 另一个问题是对先行进位加法器（CLA）的原理理解不够深入。

**感想：**

1. 通过自身学习理解MUX、简单ALU设计, 真正动手设计和实现一个ALU时，理论与实践之间的差距。这种差距促使我更加积极地查阅资料，深入理解每个组件的功能和使用方法。
2. 逐渐掌握了Logisim平台的各种功能，包括数据隧道、多路选择器等。这些工具的使用大大提高了我的设计效率，也让我对计算机硬件设计有了更深入的理解。