



《计算机组成原理实验》 实验报告

(乘法器实现)

学院名称： 数据科学与计算机学院

专业（班级）： 18 计算机类 6 班

学生姓名： 宋渝杰

学号： 18340146

时间： 2019 年 11 月 28 日

实验一：5 位阵列乘法器设计与实现

一. 实验目的：

1. 掌握 5 位阵列乘法器的原理和设计方法。

二.实验内容：

- 1.用 logisim 实现 5 位阵列乘法器

三.实验原理

实现 5 位阵列乘法器可以通过先实现 1 位乘法器和 1 位全加器，再将 25 个 1 位乘法器整合成 5 位乘法阵列，最后和 20 个 1 位全加器整合成 5 位阵列乘法器即可。

1 位乘法器：根据表达式 $P = X * Y$ 知，一个与门即可实现一位乘法器。

5 位乘法阵列：需要将 X_i 和 Y_j 生成 $X_i * Y_j$ ，只需要 $5 * 5 = 25$ 个与门阵列排序即可实现 5 位乘法阵列。

1 位全加器：根据全加器的表达式 $S_i = X_i \oplus Y_i \oplus C_{in}$ ； $C_{out} = X_i * Y_i + C_{in} * (X_i \oplus Y_i)$ ，使用两个与门，两个异或门，一个或门即可实现一位加法器。

5 位阵列乘法器：根据人工手算乘法的思路，将 X 和 Y_n 一一相乘，得到 5 个部分和，然后将该部分和阵列排序并相加即得到计算结果。5 位乘法阵列已计算出 5 个部分和（每个部分和均为 5 位），因此将 5 个部分和使用 20 个一位全加器进行加法运算，即可得到最后的结果。

四.实验器材

电脑一台，logisim 软件一套

五.实验过程和结果

（一）设计思想

根据上述的实验原理，一步步设计模块，进行连线即可

（二）设计方法

具体步骤如下：

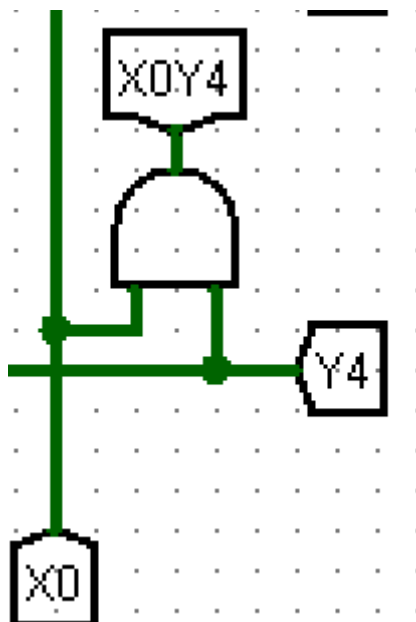
Logisim：

- 1.根据 1 位乘法器的设计思路，设计 1 位乘法器；
- 2.根据 5 位乘法阵列的设计思路，将 25 个 1 位乘法器（与门）阵列排序；
- 3.根据 1 位加法器的设计思路，设计 1 位加法器；
- 4.使用 5 位乘法阵列和 20 个 1 位加法器，形成最终的 5 位阵列乘法器。

（三）设计模块

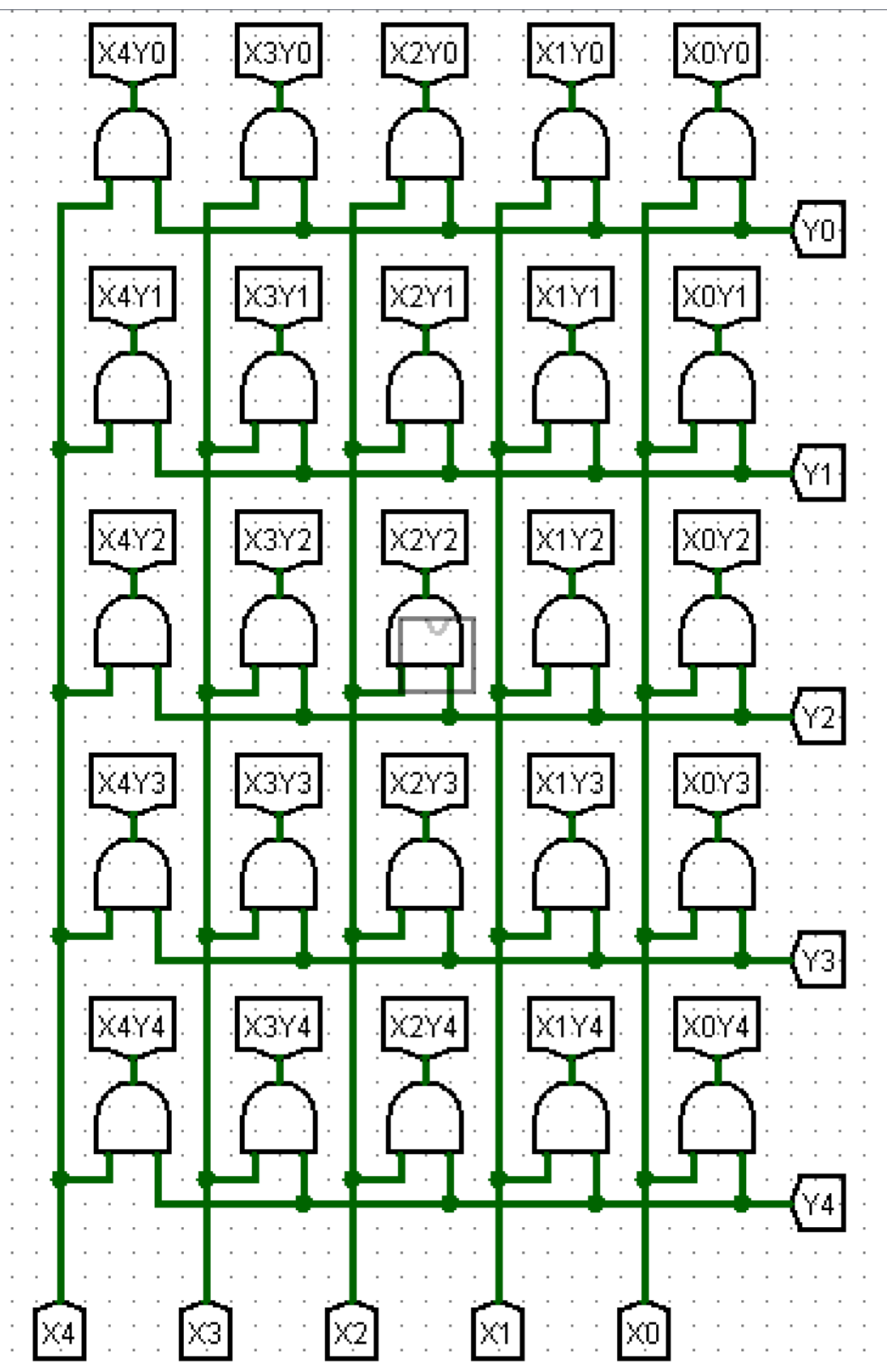
1.设计一位乘法器

根据上述思路得到，一个与门即可实现一位乘法器：



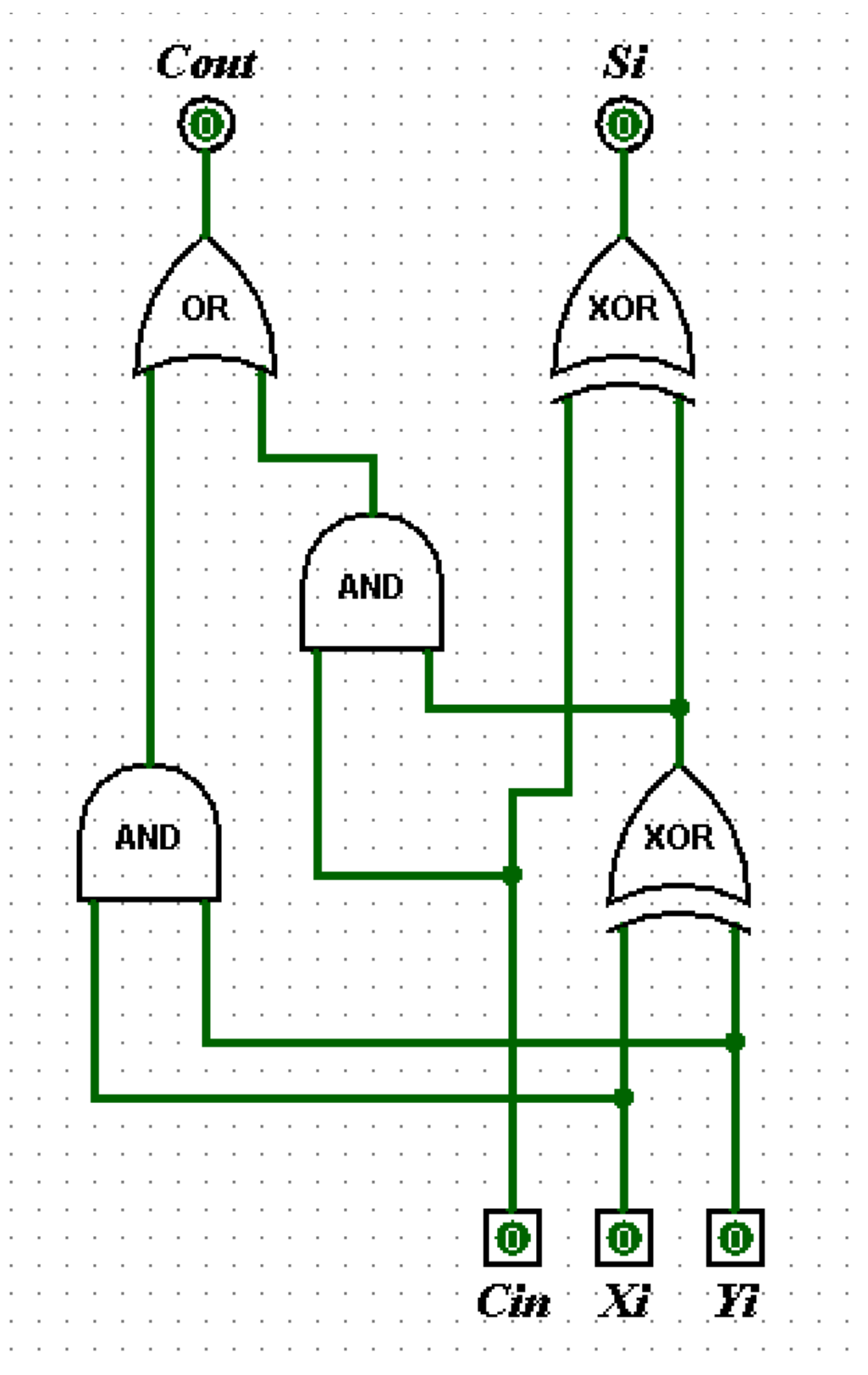
2.设计 5 位乘法阵列

根据上述思路得到，将 25 个与门阵列排序即可实现 5 位乘法阵列：



3.设计 1 位全加器

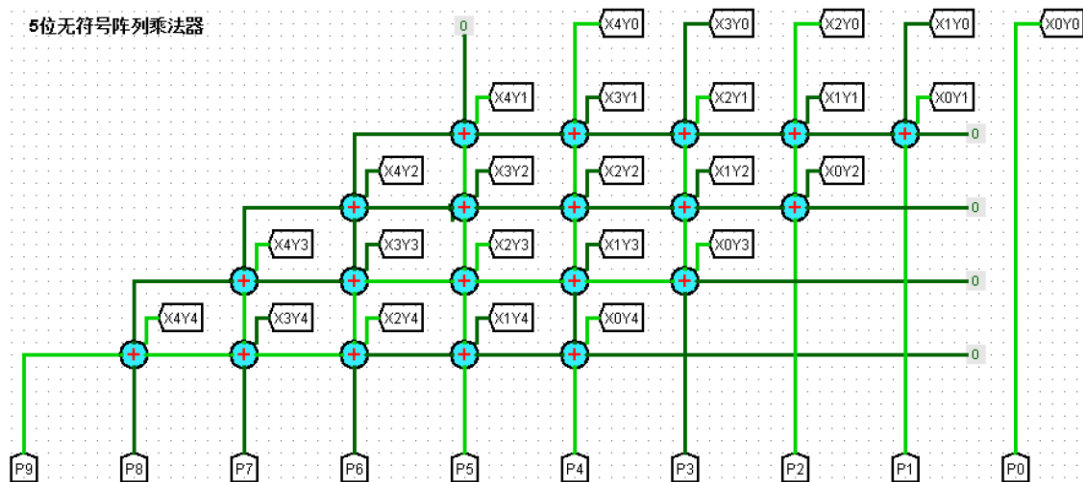
根据 S_i 和 C_{out} 的表达式，设计 1 位全加器电路



4.设计 5 位阵列乘法器

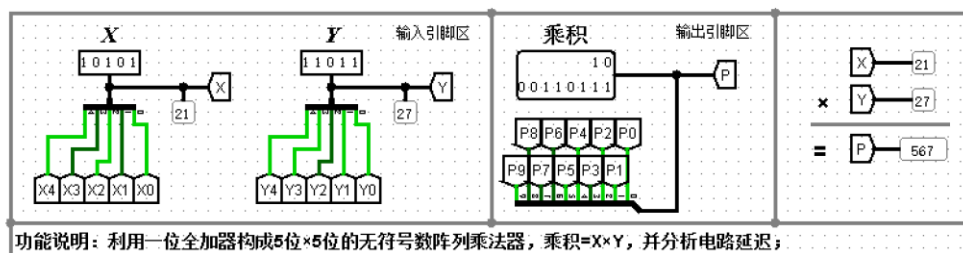
将 5 位乘法阵列和 20 个 1 位全加器阵列连接起来即可

5位无符号阵列乘法器



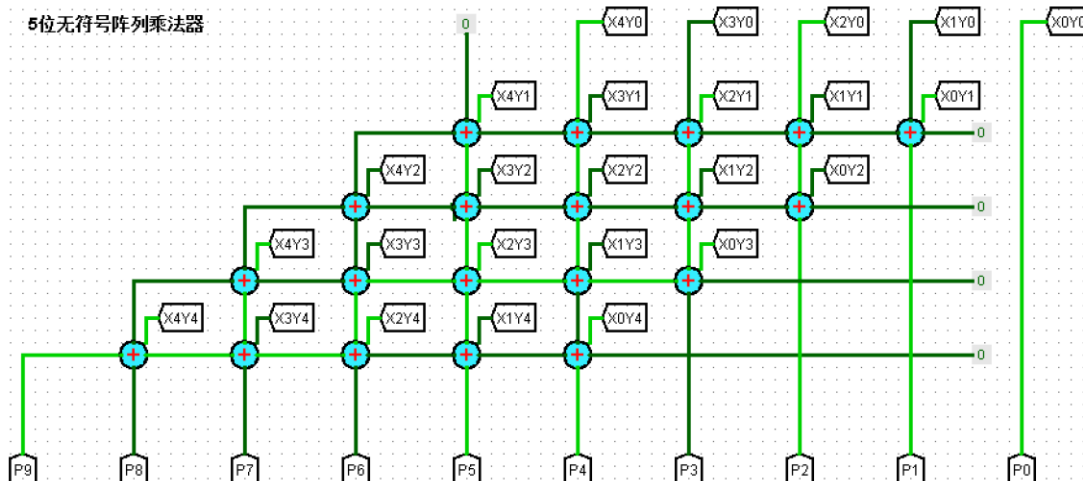
(四) 进行仿真测试

测试样例 1: ($21 * 27 = 567$)

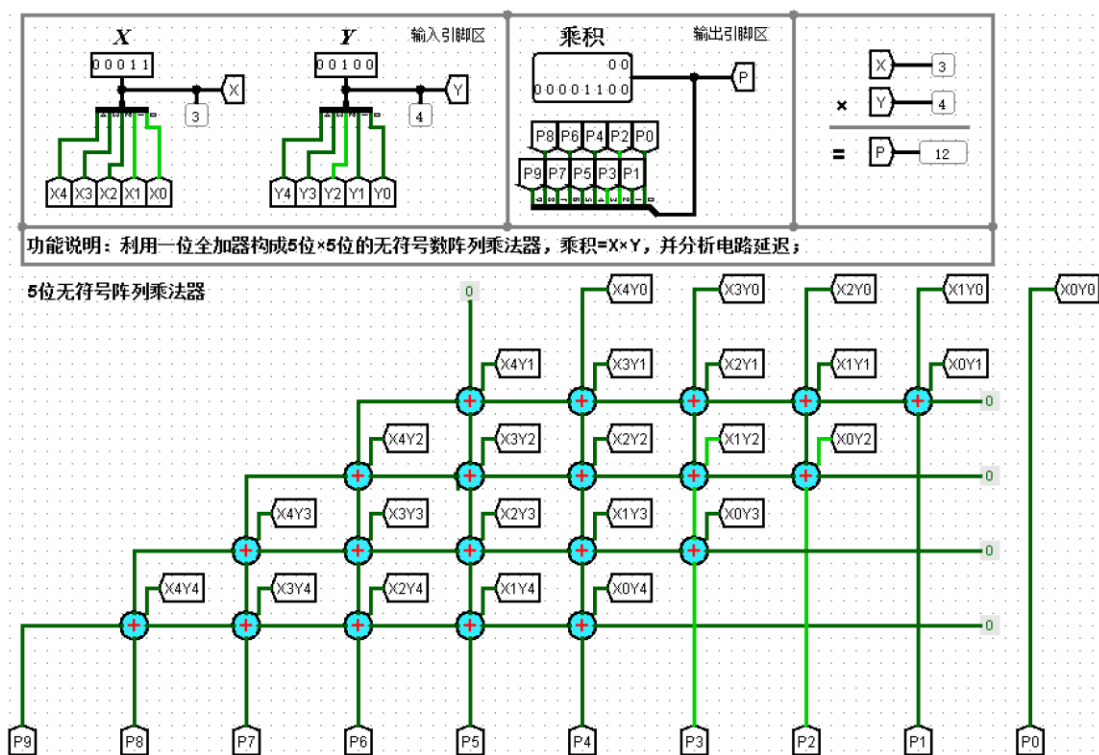


功能说明: 利用一位全加器构成5位×5位的无符号数阵列乘法器, 乘积=X×Y, 并分析电路延迟;

5位无符号阵列乘法器



测试样例 2: (3 * 4 = 12)



实验二：6 位补码阵列乘法器设计与实现

一. 实验目的:

1. 掌握 6 位补码阵列乘法器的原理和设计方法。

二.实验内容:

- 1.用 logisim 在 5 位阵列乘法器的基础上，实现 6 位补码阵列乘法器

三.实验原理

实现 6 位补码阵列乘法器可以通过将两补码数的绝对值先计算出来，使用 5 位阵列乘法器得到积的绝对值（原码），再根据两补码数的符号判断积的符号，最后根据积的符号将绝对值（原码）转换成补码形式即可。

四.实验器材

电脑一台，logisim 软件一套

五.实验过程和结果

（一）设计思想

根据上述的实验原理，一步步设计模块，进行连线即可

（二）设计方法

具体步骤如下：

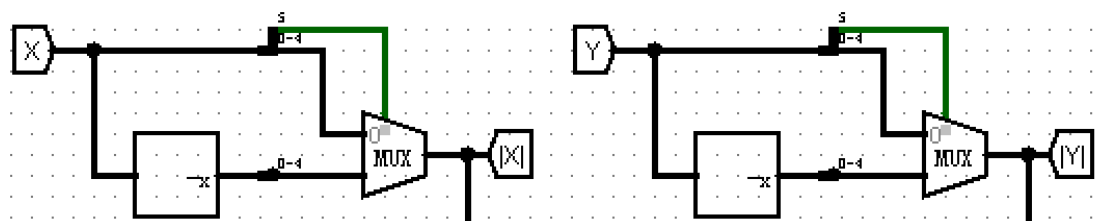
Logisim：

- 1.设计绝对值生成电路；
- 2.使用 5 位阵列乘法器将两个输入的绝对值计算出积的绝对值；
- 3.判断积的符号，并将积的绝对值转化成补码形式。

（三）设计模块

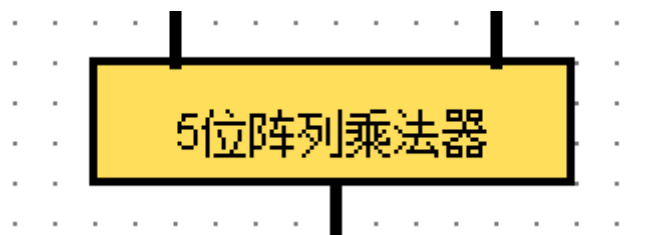
1.设计绝对值生成电路

先判断输入数的符号，如果是正数则绝对值为 0-4 位，如果是负数则绝对值为取反加一后的 0-4 位，使用 logisim 自带的补码器（实现取反加一）和数据选择器即可实现。



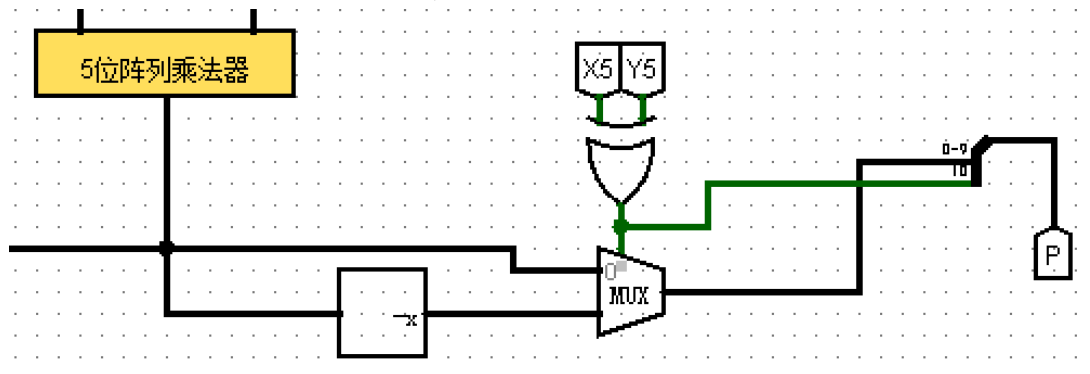
2.使用 5 位阵列乘法器计算积的绝对值

将上述得到的两个输入的绝对值用 5 位阵列乘法器计算即得到积的绝对值：

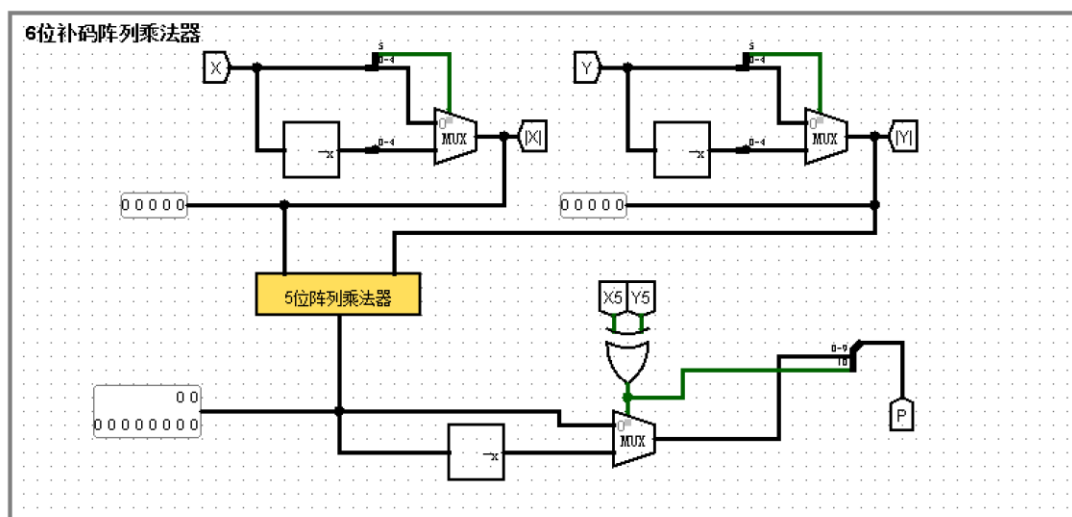


3.生成积的补码形式

根据输入的符号判断积的符号，再根据积的符号判断是否需要进行取反加一操作即可

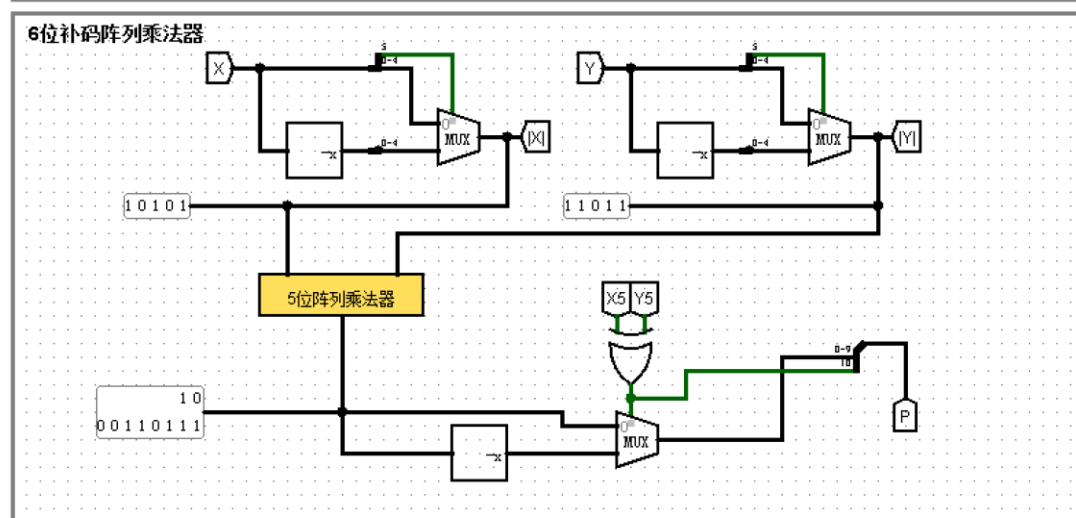
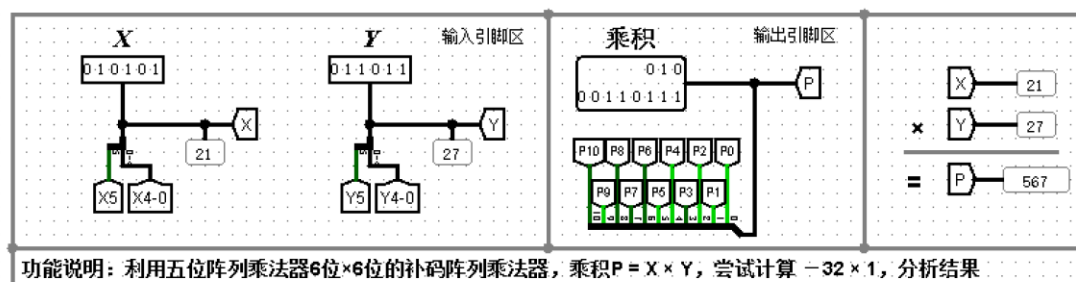


4.最终的电路图

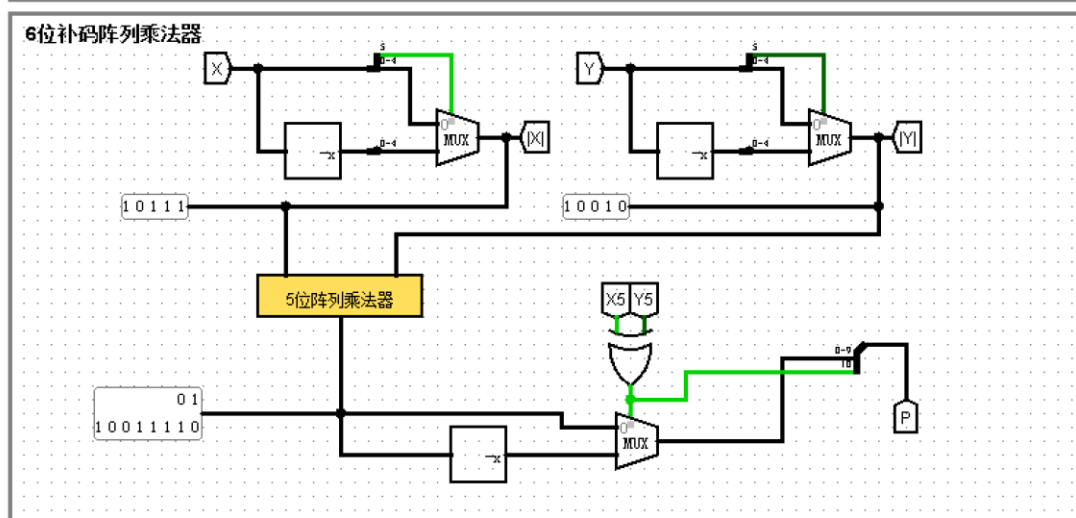
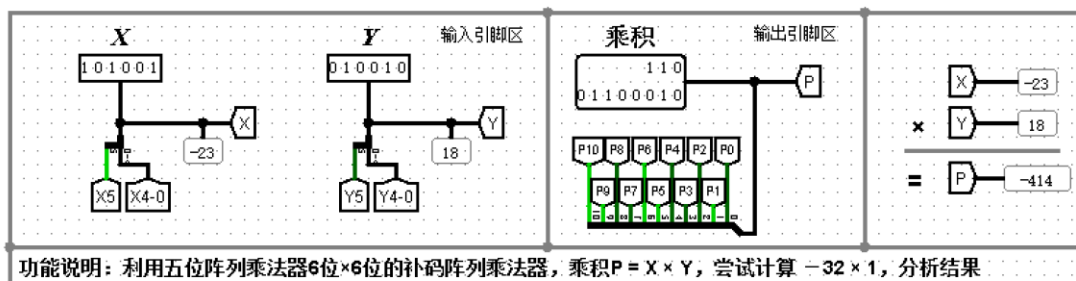


(四) 进行仿真测试

测试样例 1: ($21 \times 27 = 567$)

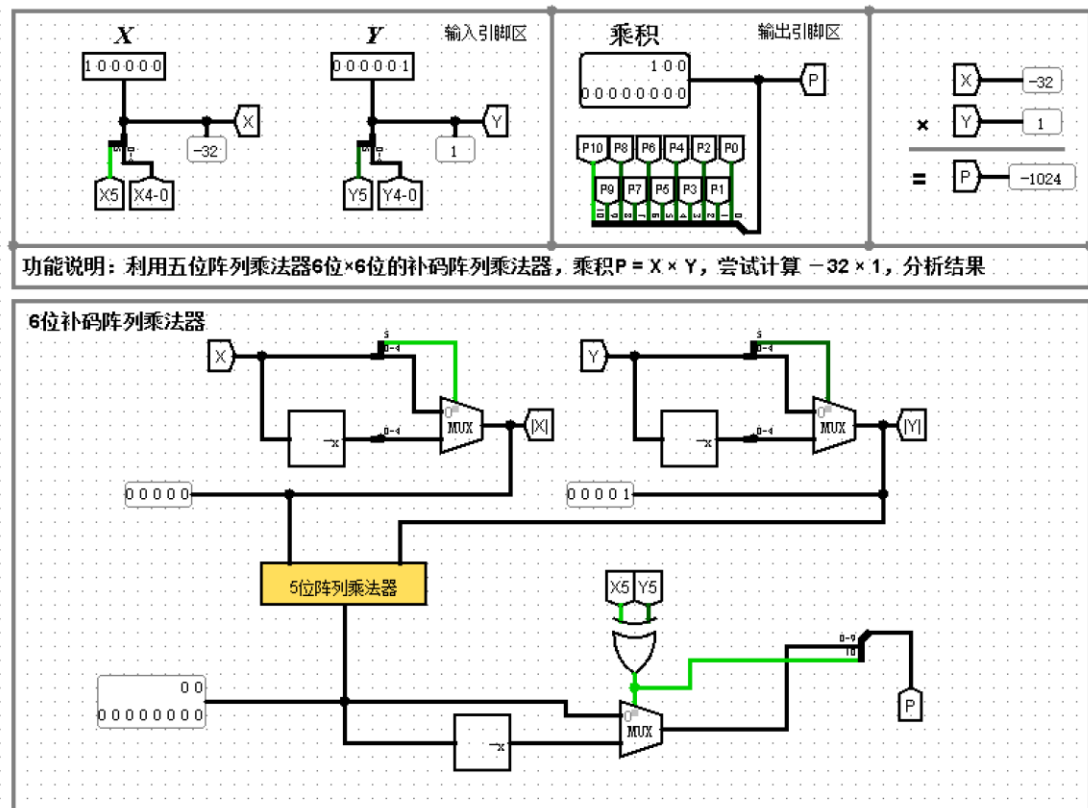


测试样例 2: ($-23 \times 18 = -414$)



测试样例 3: $(-32 * 1)$ (溢出)

由于对 X 取 5 位绝对值的时候出现了误差 (取的值为 00000), 因此在最后的计算中结果也相应出错, 这是 -32 导致数据溢出的问题



实验心得

本次乘法器设计实验中, 具体实现过程比较明确, 遵循步骤设计即可。

遇到的困难:

1. 在 5 位阵列乘法器中, 部分和的累加过程的连线需要一定时间思考分析;
2. 在 6 位补码阵列乘法器中, 也花了一定时间思考出设计过程, 即如何用上 5 位阵列乘法器, 以及怎么判断积的绝对值需要取反加一;
3. 分析 -32×1 的结果错误问题也是一个难点。

解决方式:

1. 翻阅课件 ppt, 根据 5 位阵列乘法器的设计图进行连线, 同时在 logisim 测试正确性;
2. 根据 5 位阵列乘法器是无符号数设计, 以及对输入取绝对值的提示, 设计出了 6 位补码阵列乘法器的实现过程;
3. 在 logisim 使用探针发现 -32 的绝对值 32 超出 5 位表达 (溢出), 导致了计算错误。

实验收获:

1. 完成了 5 位阵列乘法器和 6 位补码阵列乘法器的设计;
2. 对 logisim 软件的运用更加熟练。