**运**

**算**

**器**

**实**

**验**

**报**

**告**

**书**

**学 院： 人工智能与自动化学院**

**班 级： 人工智能2004班**

**姓 名： 陈乃睿**

**学 号： U202012593**

**实验时间： 2022年5月27日**

目录

[一、 实验名称 3](#_Toc9536)

[二、 实验目的 3](#_Toc1382)

[1. 加法器实验: 3](#_Toc13478)

[2. 阵列乘法器实验： 3](#_Toc3323)

[3. 一位乘法器实验： 3](#_Toc4654)

[4. ALU实验： 3](#_Toc22195)

[三、 实验内容 3](#_Toc20659)

[1. 加法器实验： 3](#_Toc54)

[2. 阵列乘法器实验： 4](#_Toc31896)

[3. 一位乘法器实验： 4](#_Toc31844)

[4. ALU实验： 4](#_Toc7447)

[四、 实验电路与设计思路 4](#_Toc7658)

[1. 加法器实验 4](#_Toc26912)

[(1) 八位可控加减法器 4](#_Toc31518)

[(2) 四位先行进位74182： 5](#_Toc5168)

[(3) 四位快速加法器： 5](#_Toc7691)

[(4) 16位快速加法器： 5](#_Toc31418)

[(5) 32位快速加法器： 6](#_Toc18841)

[2. 阵列乘法器实验 6](#_Toc29477)

[(1) 5位阵列乘法器： 6](#_Toc9522)

[(2) 6位补码阵列乘法器： 7](#_Toc28523)

[(3) 5位无符号乘法流水线： 7](#_Toc27220)

[3. 一位乘法器实验 8](#_Toc15565)

[(1) 原码一位乘法器： 8](#_Toc19440)

[(2) 补码一位乘法器： 9](#_Toc23120)

[4. ALU实验 9](#_Toc19848)

[五、 实验结果 10](#_Toc25549)

[1. 加法器实验 10](#_Toc12387)

[(1) 八位可控加减器： 10](#_Toc16097)

[(2) 4位先行进位74182： 11](#_Toc3930)

[(3) 4位快速加法器： 11](#_Toc24577)

[(4) 16位快速加法器： 11](#_Toc17926)

[(5) 32位快速加法器： 12](#_Toc29047)

[2. 阵列乘法器实验 12](#_Toc26196)

[(1) 5位阵列乘法器： 12](#_Toc13086)

[(2) 6位补码阵列乘法器： 13](#_Toc22477)

[(3) 5位无符号乘法流水线： 14](#_Toc24844)

[3. 一位乘法器实验 15](#_Toc22038)

[(1) 原码一位乘法器： 15](#_Toc8341)

[(2) 补码一位乘法器： 15](#_Toc28786)

[4. ALU实验 16](#_Toc11842)

[六、 实验总结 17](#_Toc31305)

# 实验名称

运算器实验

1. 加法器实验
2. 阵列乘法器实验
3. 一位乘法器实验
4. ALU实验

# 实验目的

1. 加法器实验:
2. 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，熟悉Logisim平台基本功能，能在Logisim中实现多位可控加减法电路。
3. 掌握快速加法器中先行进位的原理，能利用相关只是设计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，能分析对应电路的时间延迟。
4. 理解成组进位产生函数、成组进位传递函数的概念，熟悉Logisim平台子电路的概念，能利用前述实验封装好的4位先行进位电路以及4位快速加法器子电路构建16位、32位、64位快速加法器，并能利用相关知识分析对应电路的时间延迟，理解电路并行的概念。
5. 阵列乘法器实验：

掌握阵列乘法器的实现原理，能够分析阵列乘法器的性能，能在Logisim中绘制阵列乘法器电路。

1. 一位乘法器实验：
2. 掌握原码1位乘法运算的基本原理，熟练掌握Logisim中寄存器组件的使用，能在Logisim平台中设计实现一个8位×8位的无符号数乘法器。
3. 掌握补码1位乘法运算的基本原理，熟练掌握Logisim寄存器组件的使用，能在Logisim平台中设计实现一个8位×8位的定点数补码1位乘法器。
4. ALU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，能利用前述实验完成的32位加法器和Logisim中的运算组件构造指定规格的ALU单元。

# **实验内容**

1. 加法器实验：
   1. 在Logisim模拟器中打开alu.circ文件，在对应的子电路中利用已经封装好的全加器设计8位串行可控加减法电路。
   2. 在Logisim中打开alu.circ文件，在对应子电路中实现可级联的4位先行进位电路，再利用4位先行进位电路构造4位快速加法器。
   3. 在Logisim中打开alu.circ文件，在对应的子电路中利用4位先行进位电路和4位快速加法器构造16位组间先行进位、组内先行进位快速加法器，并验证其功能是否正常。
   4. 用两个16位加法器直接串联，成为32位快速加法器。
2. 阵列乘法器实验：

在Logisim中打开alu.circ文件，在5位阵列乘法器中实现斜向进位的阵列乘法器。

1. 一位乘法器实验：
   1. 在给出的文件中增加控制电路和数据通路使得该电路能自动完成8位无符号的1位乘法运算，首先设置引脚初始值，然后驱动时钟自动仿真，电路可自动完成运算，运算结束结果传输到输出引脚，运算结束时电路应该自动停止。
   2. 在给出的文件中增加控制电路和数据通路，使得该电路能自动完成8位补码1位乘法运算。首先设置引脚初始值，然后驱动时钟仿真，电路自动完成运算，运算结束则结果传输到输出引脚，运算结束时电路应该自动停止。
2. ALU实验：

利用前面实验封装好的32位加法器以及Logisim平台中现有运算组件构建一个32位算术逻辑运算单元（禁用Logisim系统自带的加法器，减法器），可支持算术加、减、乘、除、逻辑与、或、非、异或、逻辑左移、逻辑右移，支持常用程序状态标志（有符号溢出OF、无符号溢出UOF、结果相等Equal）

# **实验电路与设计思路**

1. 加法器实验
   1. 八位可控加减法器

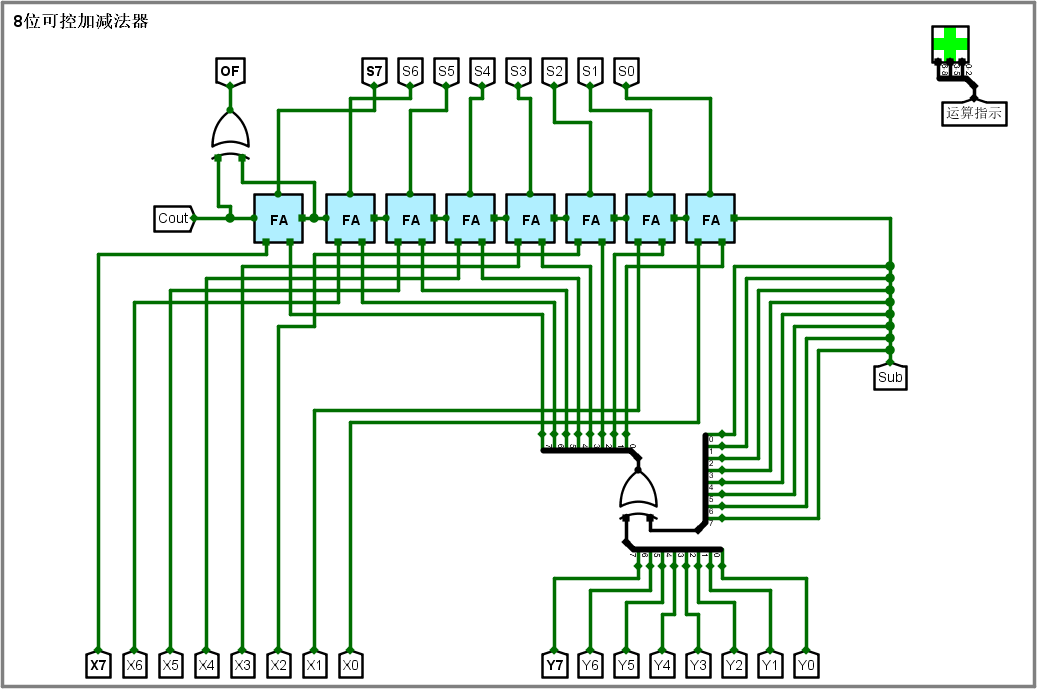


图 1

sub标志着该运算器是运算加还是减。若sub=0，则为加法，将x(i)与y(i)分别相加，输出加完后的数据，并将进位位传入x(i+1)和y(i+1)的加法中；若sub=1，则为减法，此时Y应变为相反数，所有位数应当取反，并在最后加上1，该“1”由sub数据直接传入x(0)与y(0)加法器的进位位中即可。

* 1. 四位先行进位74182：

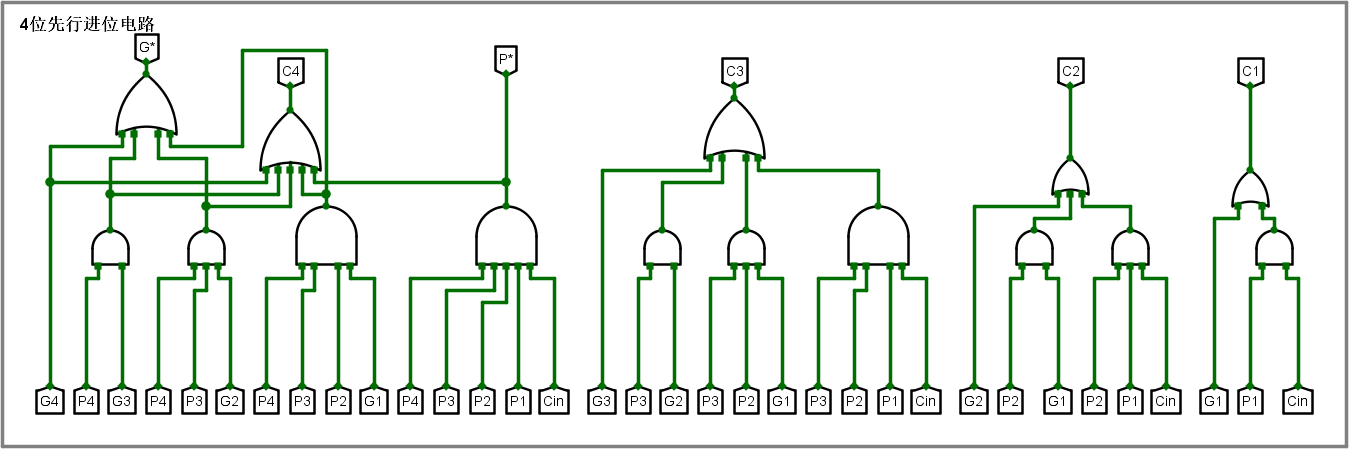


图 2

按照书上推导的公式，用P、Q来计算每个位数的进位，以方便后续四位数加法的直接计算。

* 1. 四位快速加法器：

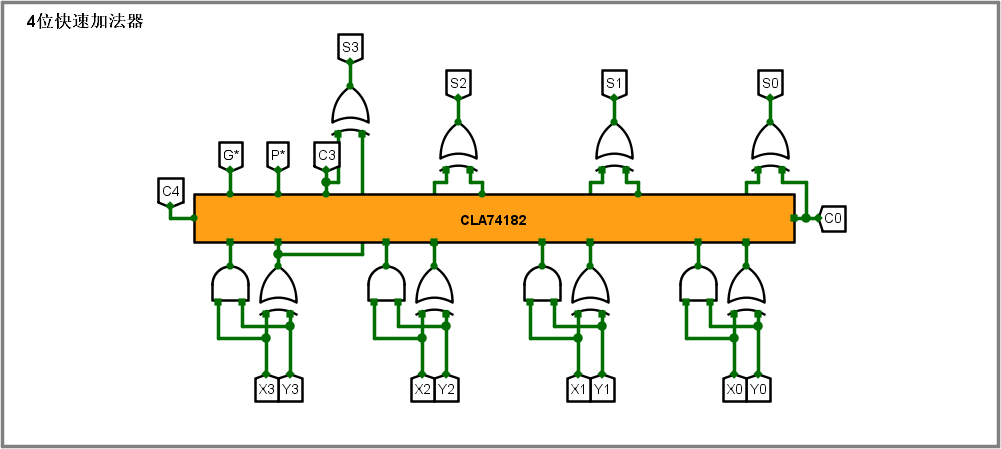


图 3

使用公式计算出加法运算后的每一位，并传出以方便后续16位加法运算的进位计算。

* 1. 16位快速加法器：

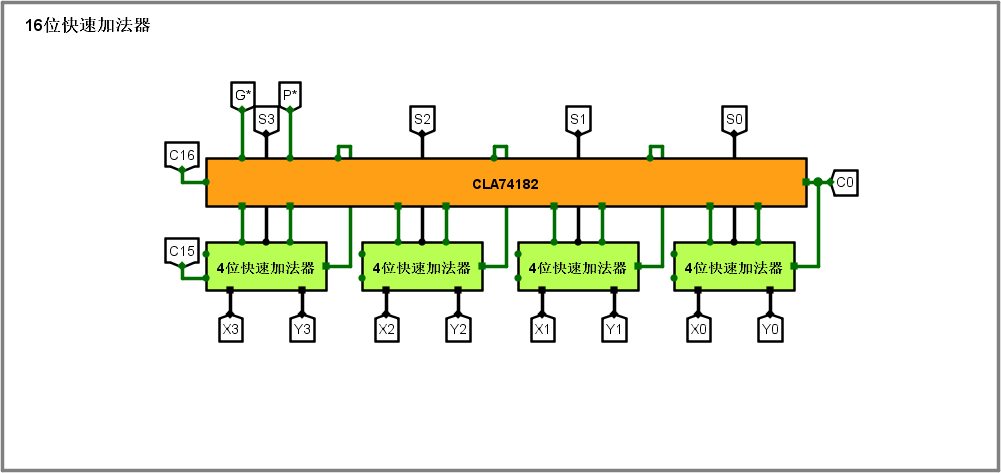


图 4

仿照4位快速加法器，不同的是加法运算后的每一个单位（每4位数为一个单位）由4位快速加法器得出，且进位位由计算得出。

* 1. 32位快速加法器：

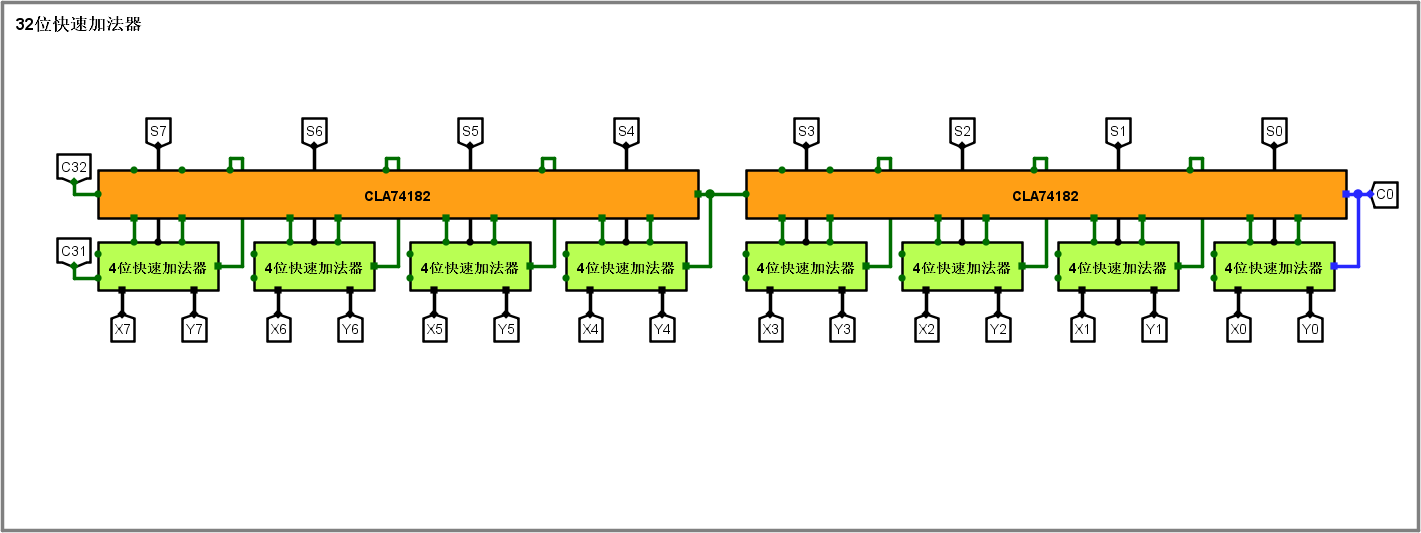


图 5

用两个16位快速加法器相连，其中要将低位的16位加法器的溢出（进位）信息传递到高位的16位加法器中，应传入到快速进位器的初始进位位以及第一个4位快速进位器的进位位。最后引出引脚C31与C32，用于后续实验电路中溢出的判断。

1. 阵列乘法器实验
   1. 5位阵列乘法器：

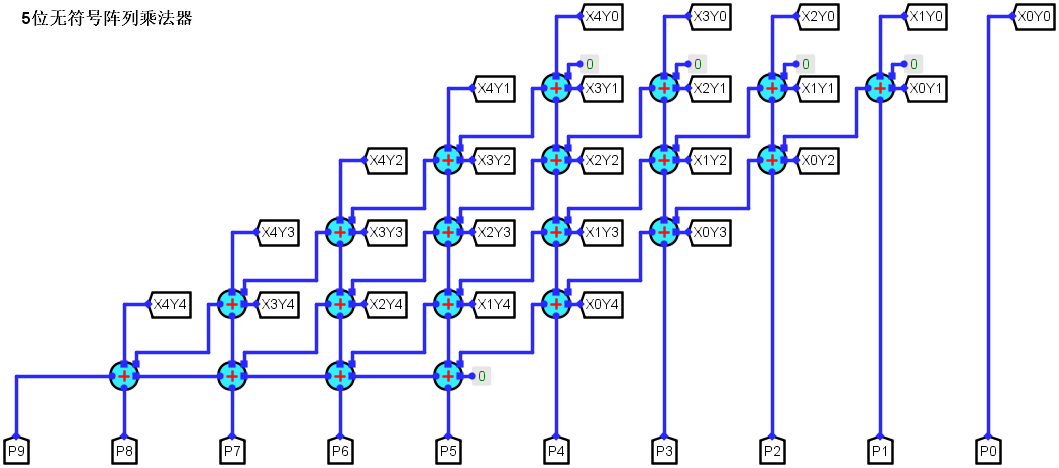


图 6

该乘法器的原理和列竖式计算乘法的原理一模一样，将各行错开相加，将进位位传入更高位的列进行计算，最后得出结果。

* 1. 6位补码阵列乘法器：

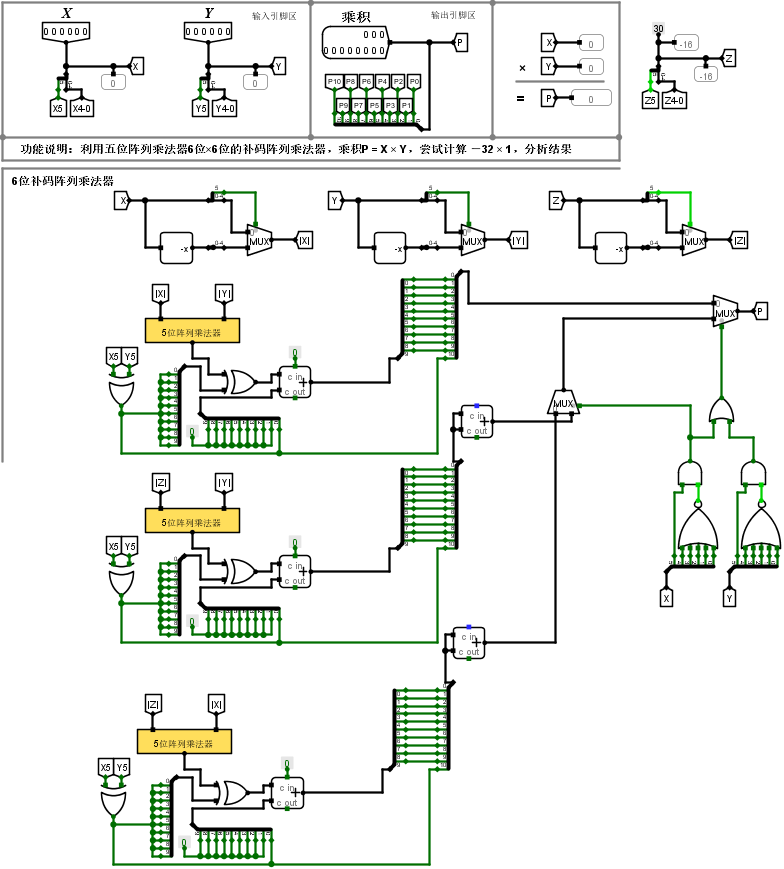


图 7

将两乘数X、Y分为符号位与数据位，符号位（X5、Y5）用于确认结果的符号，数据位用于计算结果的数值。若结果符号为正，则X5X6的符号相同，异或值为0，直接传入10位结果与1位符号即可；若结果符号为负，则X5X6符号不同，异或值为1，数据位应当取反后+1，即：乘法器出来后的数据位每一位与1异或后总体数值+1得到最终的数据位，1传入到符号位。

其中，当X、Y其中有一个补码为“100000”（-32）时，结果会发生错误。此时，可将-32拆成-16-16来进行计算，结果乘2后传出即可。

* 1. 5位无符号乘法流水线：

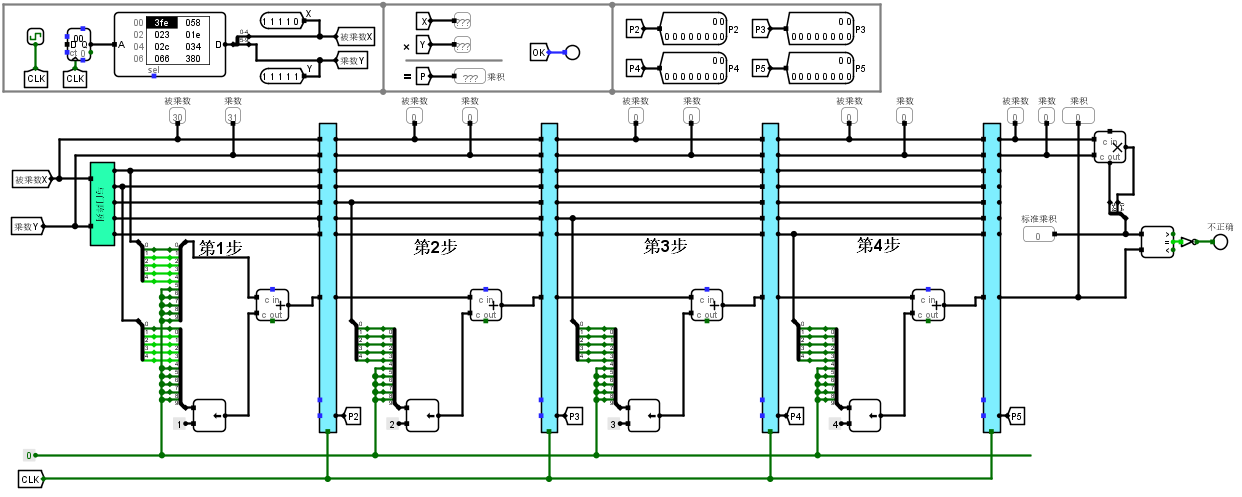


图 8

该电路分为上下两部分，上半部分传输正确的乘法计算式，下半部分用5位无符号乘法器的方法自己连接乘法电路，最终在最右边进行判断。

1. 一位乘法器实验
   1. 原码一位乘法器：

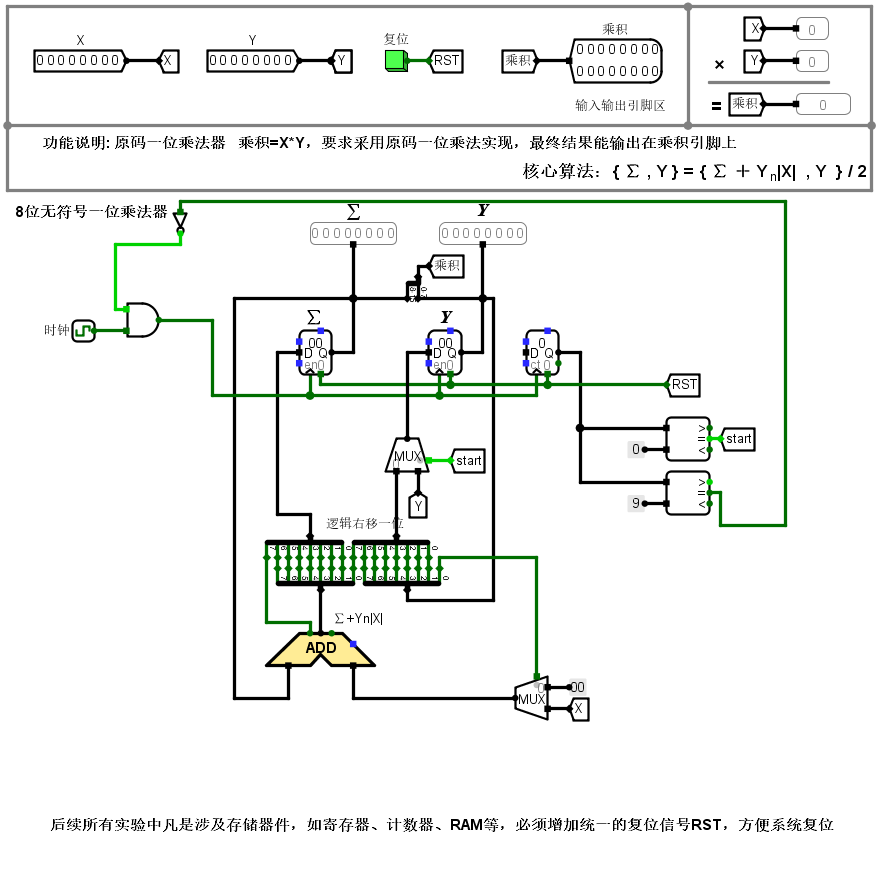


图 9

原码一位乘法器的运行原理和5位数阵列乘法器相似，不同的是该乘法器是一行一行按顺序加出结果。初始累加和为0，每次循环进行一次判断：该次计算时Y的最后一位为0还是1？若为0，则累加和加上0；若为1，则累加和加上X。此后，累加和与Y均右移一位，累加和右移是等价于列竖式时每一行左移；Y右移是为了方便取出Y的下一位（变为最后一位）。其中，累加和右移后，最高位应补齐加法器的溢出位。第一次时钟周期时，应将Y的初始值送入循环；当再进行八次时钟周期后，即9次时钟周期后，加法已经完成，此时应将时钟信号屏蔽，防止乘法器继续变化。

* 1. 补码一位乘法器：

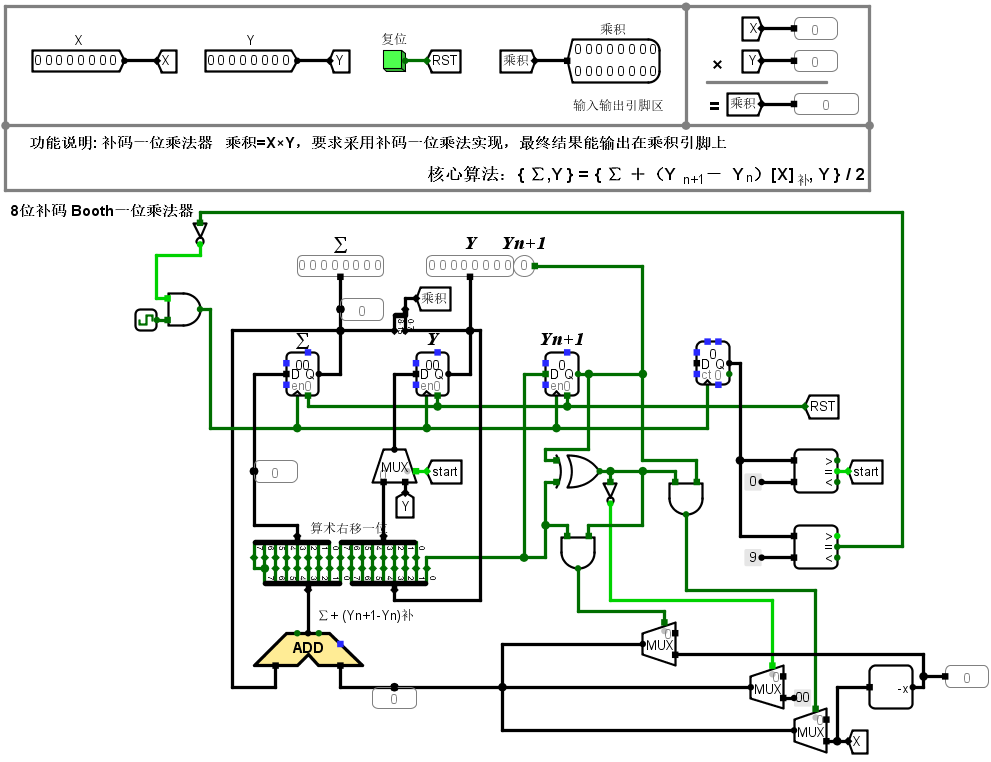


图 10

补码一位乘法器的思想和原码一位乘法器的思想大同小异，主要不同是补位以及加数的不同。当Yn+1与Yn相同时，加数为0；当不同时，若Yn+1为1，则加数应为X的补码；若Yn为1，则加数应为-X的补码。当累积和右移时，最高位补充的不再是加法器的溢出位，而是补平移前的符号位。其余线路与原码一位乘法器相同。

1. ALU实验

算术逻辑运算单元ALU：

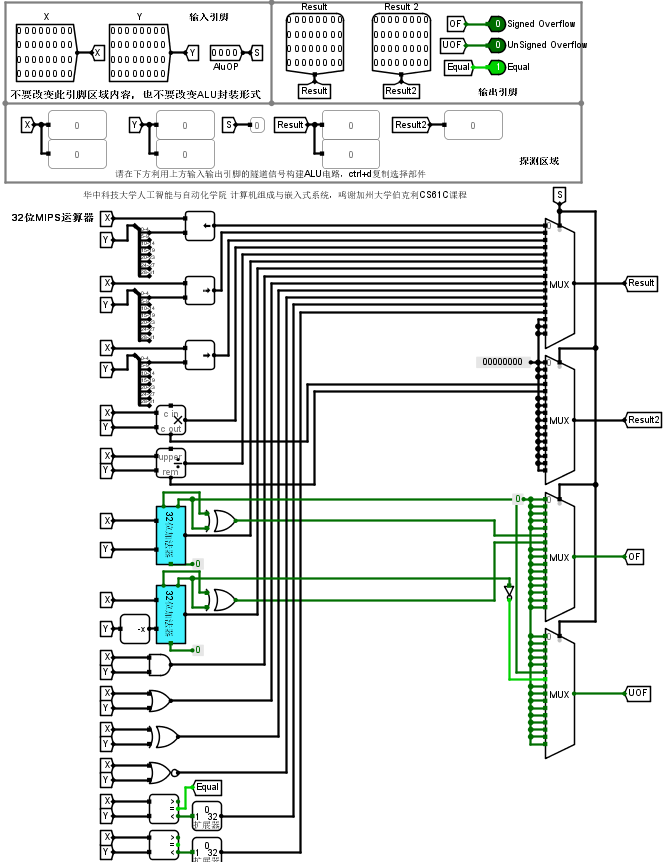


图 11

该实验要求连成一个能实现多个功能的运算器，只需分别完成各个功能的电路后，用译码器选择正确的通道输出即可。

## **实验结果**

1. 加法器实验
   1. 八位可控加减器：

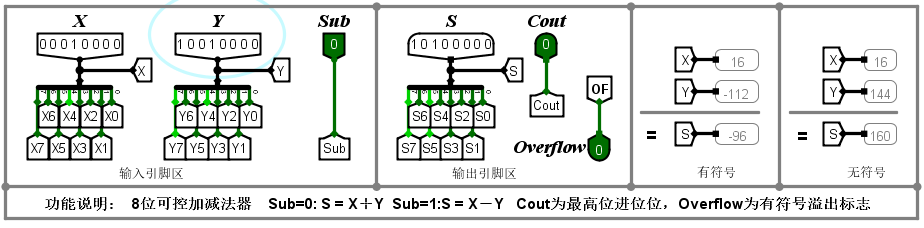


图 12

* 1. 4位先行进位74182：

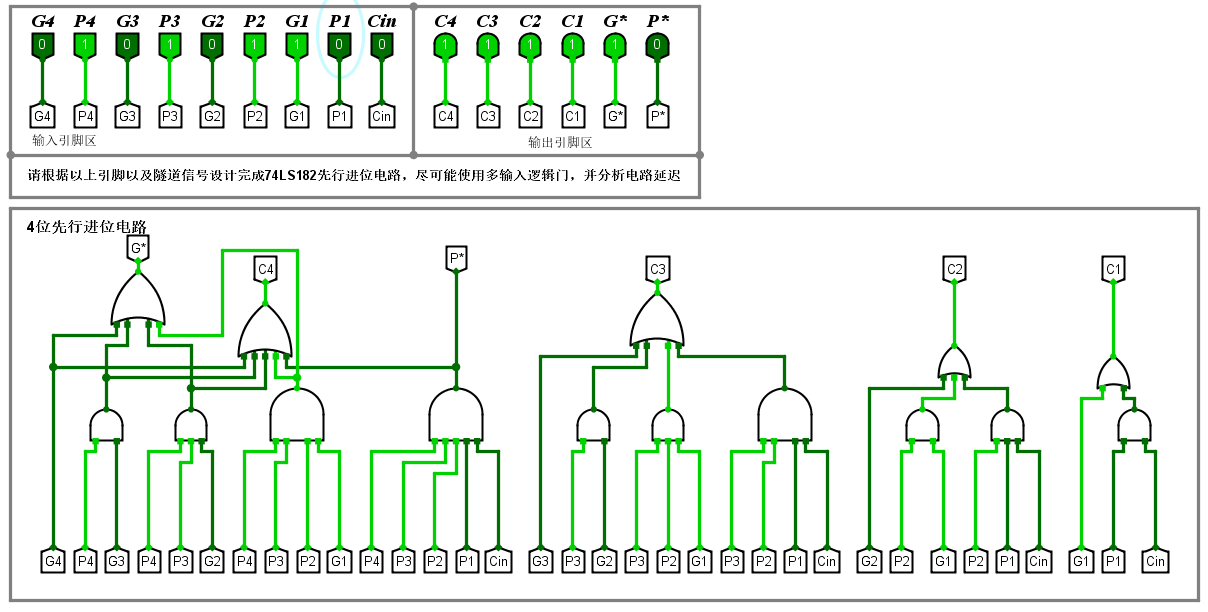


图 13

* 1. 4位快速加法器：

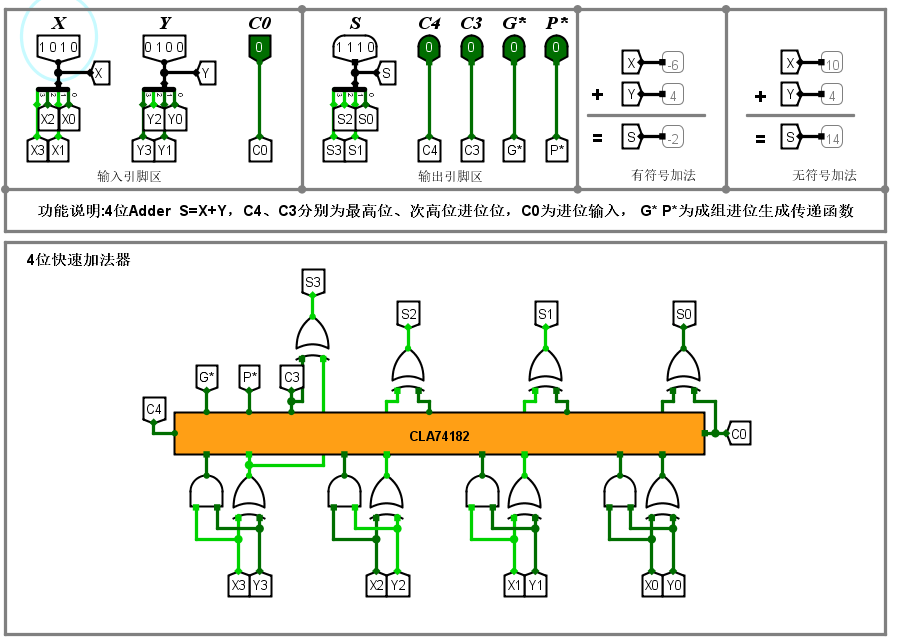


图 14

* 1. 16位快速加法器：

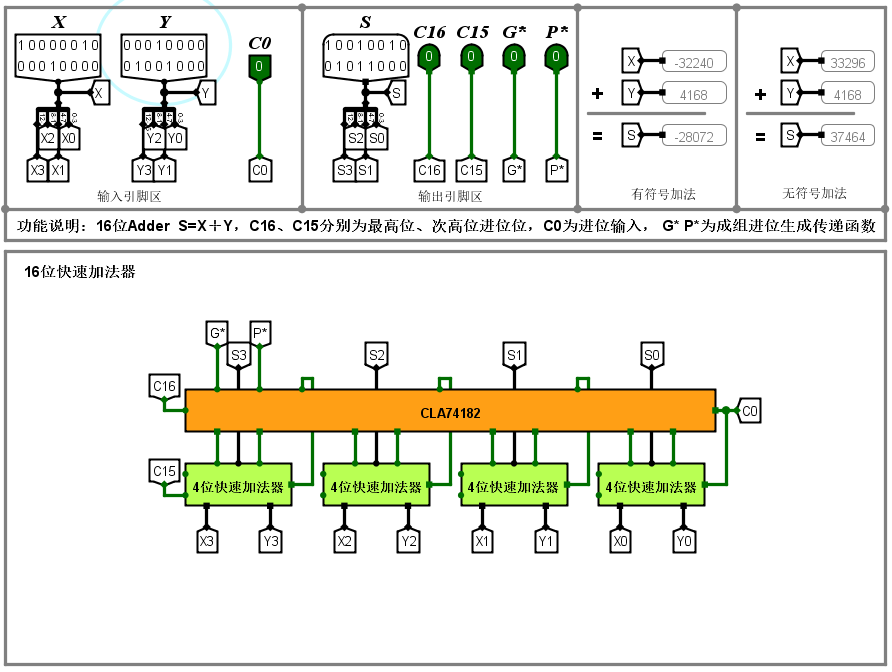


图 15

* 1. 32位快速加法器：

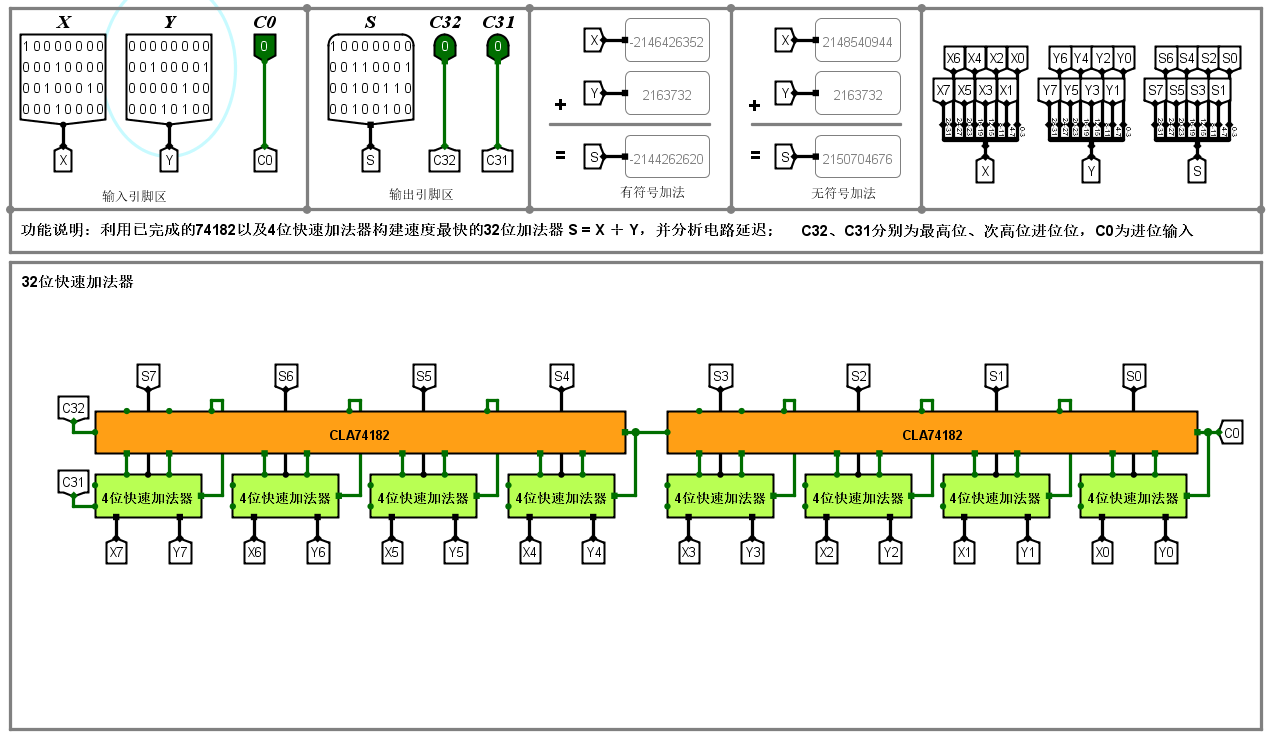


图 16

1. 阵列乘法器实验
   1. 5位阵列乘法器：

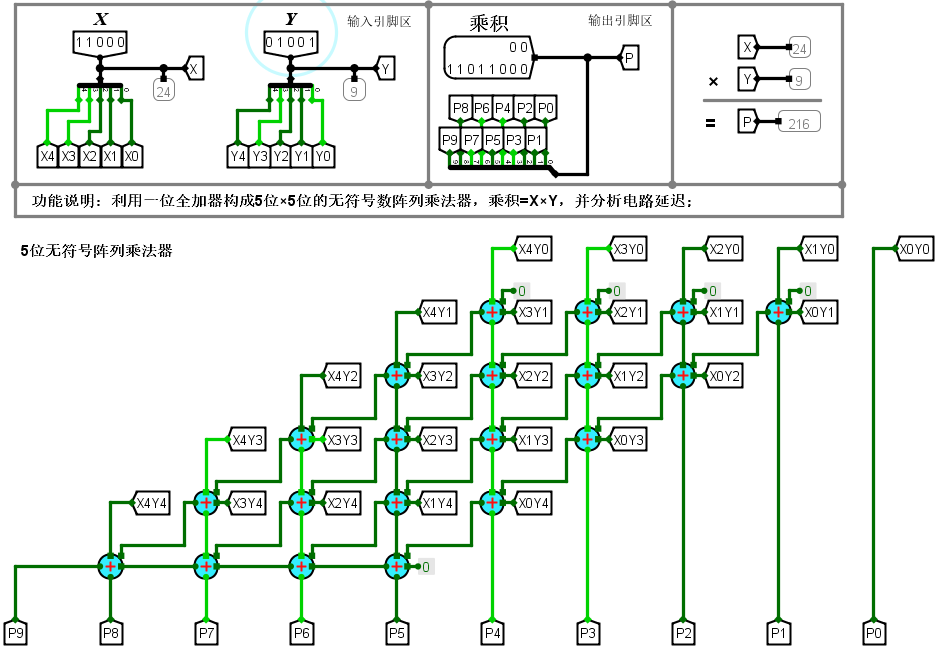


图 17

* 1. 6位补码阵列乘法器：

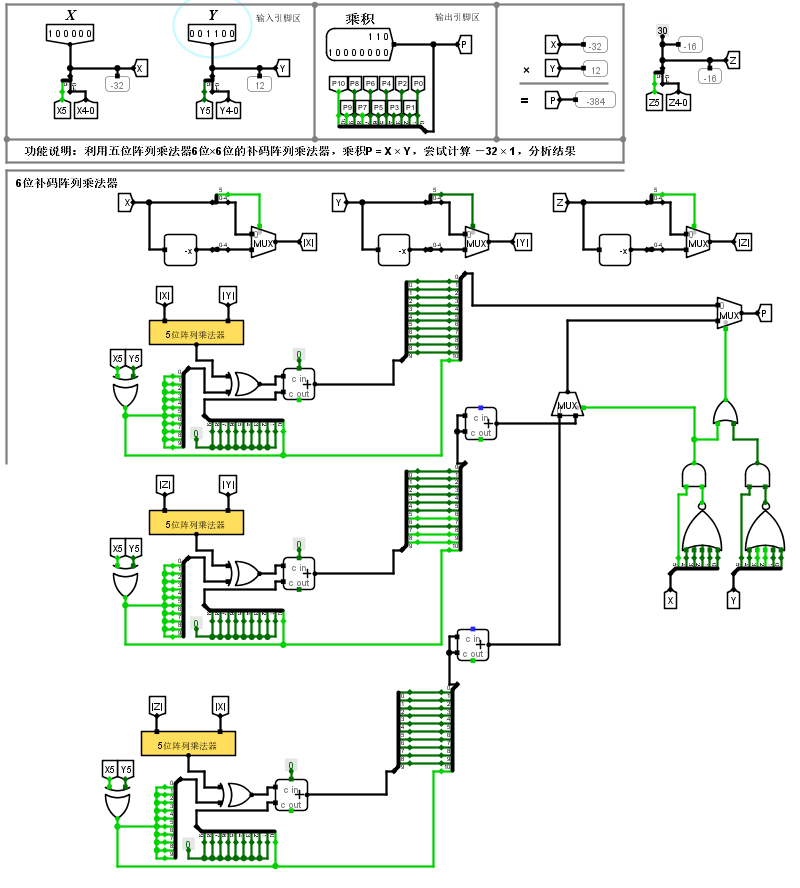


图 18

* 1. 5位无符号乘法流水线：

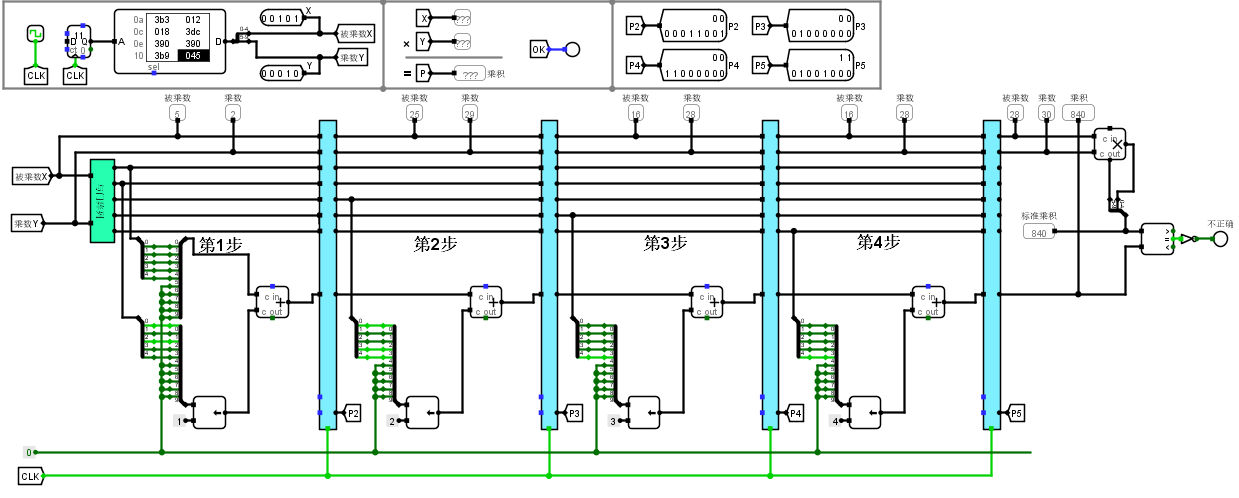


图 19

1. 一位乘法器实验
   1. 原码一位乘法器：

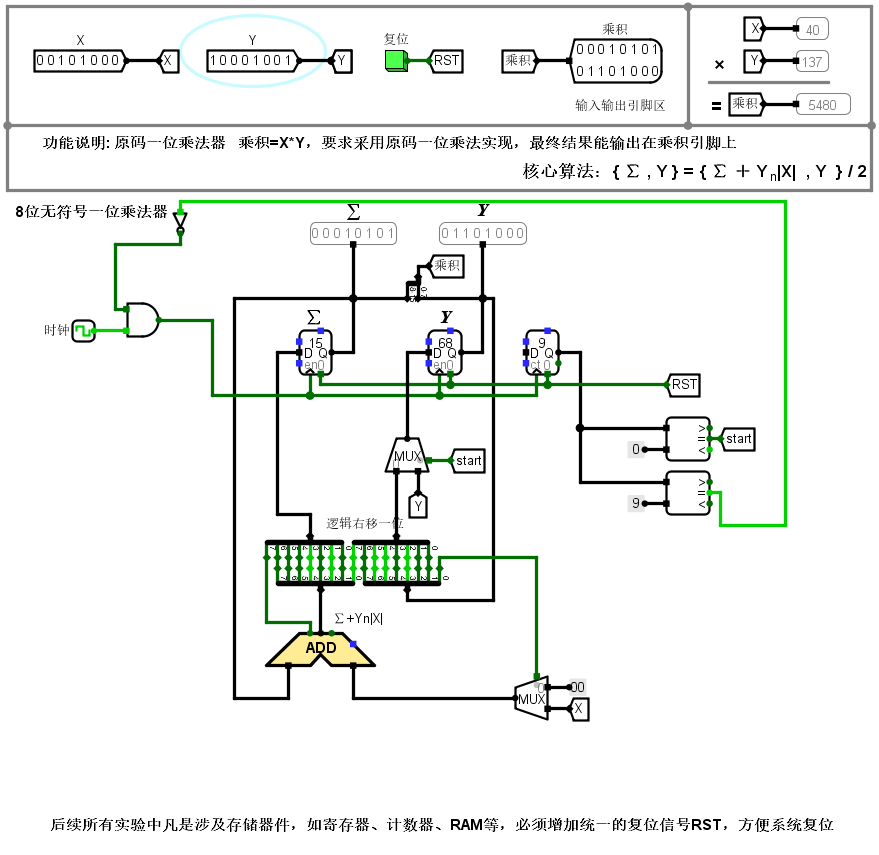


图 20

* 1. 补码一位乘法器：

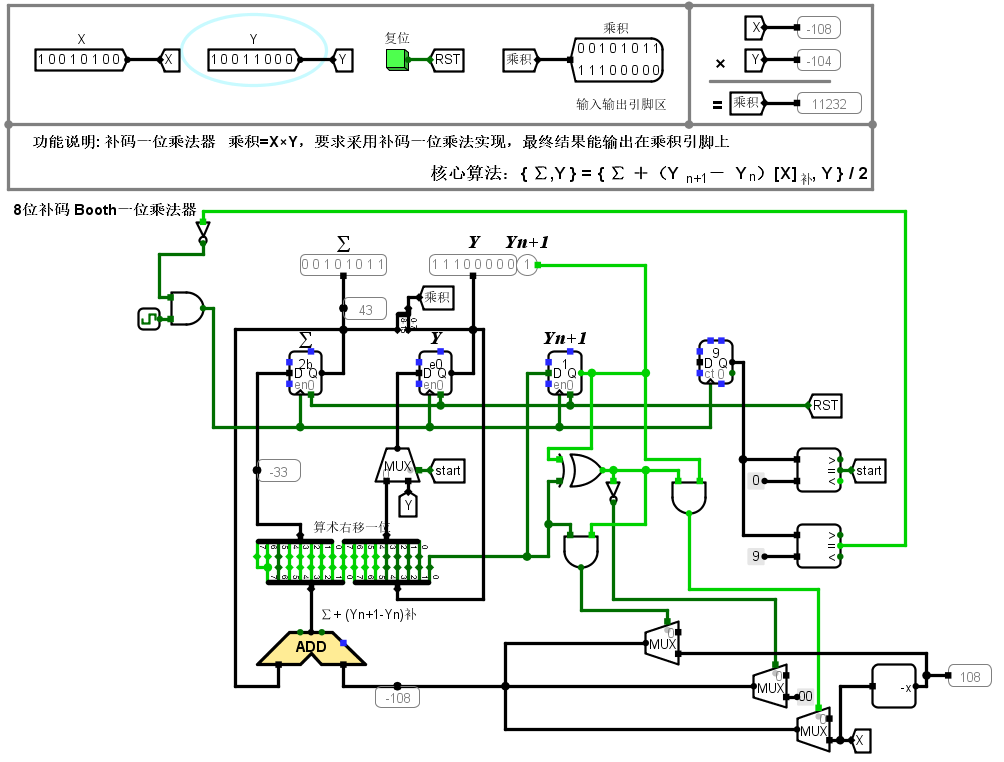


图 21

1. ALU实验

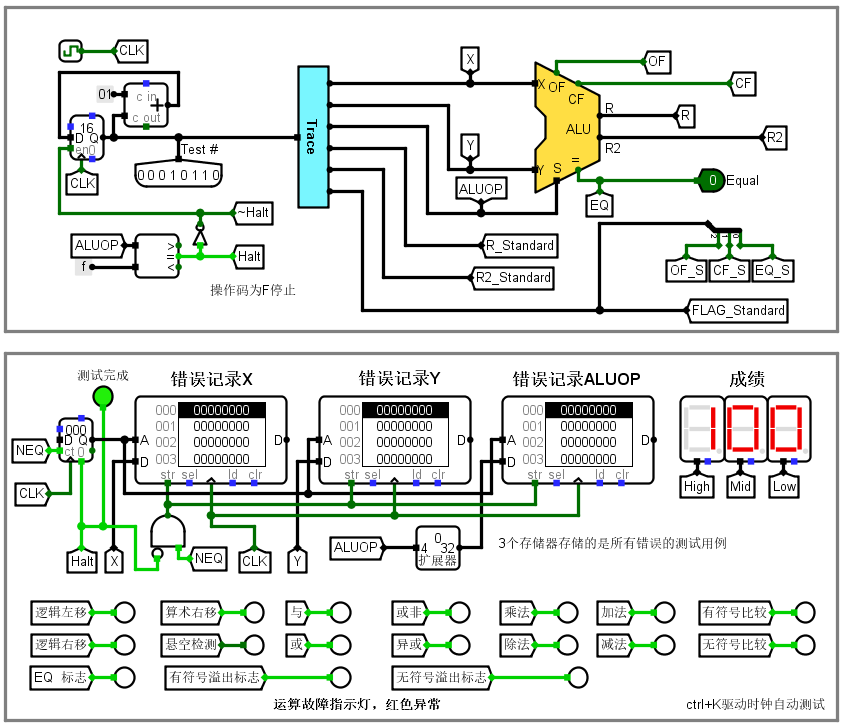


图 22

## **实验总结**

这次实验所连电路数量较多，但较为简单，在连接过程中一共出现了三次错误：一次是在32位快速加法运算器中，低位的16位加法器的溢出位仅仅只传入了高位的16位加法器的先行进位中，未传入4位加法器的进位位中；第二个错误是补码一位乘法器中，累积和右移时将加法器的进位位补充至最高位，而不是将符号位补充至最高位；第三个错误是理解错误题意，在ALU电路中隧道“UOF”的意思是无符号的溢出位判断，而我理解成了无溢出的判断。综上，连接线路不仅是一门技术活，还是一门耐心活。必须要有足够的耐心与一定的头脑，才能成功连接线路。期待自己下一次能有更好的表现。