

**华中科技大学**

人工智能与自动化学院

**计算机组成原理**

**实验报告**

**专业班级： 人工智能2304班**

**学 号： U202315265**

**姓 名： 杜辰宇**

**实验名称： 运算器实验**

**实验时间： 2025年3月28日**

目录

[1、实验目的 3](#_Toc194409869)

[1.1快速加法器实验 3](#_Toc194409870)

[1.1.1掌握串行加法器逻辑实现 3](#_Toc194409871)

[1.1.2掌握快速加法器逻辑实现 3](#_Toc194409872)

[1.1.3 掌握组内先行、组间先行的基本原理 3](#_Toc194409873)

[1.2乘法器实验 3](#_Toc194409874)

[1.2.1理解阵列乘法器的实现原理 3](#_Toc194409875)

[1.2.2掌握原码、补码一位乘法基本原理 3](#_Toc194409876)

[1.3算术逻辑运算单元ALU实验 3](#_Toc194409877)

[1.3.1掌握定点数加减法溢出检测方法 3](#_Toc194409878)

[1.3.2熟悉Logisim中各种运算组件的使用方法 3](#_Toc194409879)

[1.3.3 熟悉多路选择器的使用 3](#_Toc194409880)

[2、实验内容 3](#_Toc194409881)

[2.1、加法器实验： 3](#_Toc194409882)

[2.1.1、八位可控加减法器 3](#_Toc194409883)

[2.1.2、 四位先行进位 4](#_Toc194409884)

[2.1.3、四位快速加法器 5](#_Toc194409885)

[2.1.3、 多位快速加法器 5](#_Toc194409886)

[2.2、阵列乘法器实验 6](#_Toc194409887)

[2.2.1、五位阵列乘法器 6](#_Toc194409888)

[2.2.2、6位补码阵列乘法器 7](#_Toc194409889)

[2.3、一位乘法器实验 8](#_Toc194409890)

[2.3.1、原码一位乘法器 8](#_Toc194409891)

[2.3.2、补码一位乘法器 8](#_Toc194409892)

[2.4、ALU实验 9](#_Toc194409893)

[2.4.1、算术逻辑单元ALU 9](#_Toc194409894)

[2.4.2、算术逻辑单元ALU电路 9](#_Toc194409895)

[2.4.3、测试结果 10](#_Toc194409896)

[3、实验中的bug和解决方案 11](#_Toc194409897)

1、实验目的

1.1快速加法器实验

1.1.1掌握串行加法器逻辑实现

• 能设计8位可控加减法电路

1.1.2掌握快速加法器逻辑实现

• 能设计4位先行进位电路

• 能设计4位快速加法器

1.1.3 掌握组内先行、组间先行的基本原理

• 利用4位快速加法器构建16位、32位快速加法器

• 能分析相关电路延迟

1.2乘法器实验

1.2.1理解阵列乘法器的实现原理

• 能设计无符号阵列乘法器电路

• 能设计有符号补码阵列乘法器电路

1.2.2掌握原码、补码一位乘法基本原理

• 能设计原码、补码一位乘法器

• 重点掌握寄存器、多路选择器的使用

• 能设计简单的状态机进行数据通路控制

1.3算术逻辑运算单元ALU实验

1.3.1掌握定点数加减法溢出检测方法

1.3.2熟悉Logisim中各种运算组件的使用方法

• 逻辑运算部件、乘法器、除法器、移位器

1.3.3 熟悉多路选择器的使用

2、实验内容

2.1、加法器实验：

2.1.1、八位可控加减法器

设计基本思想：多位可控加减法器就是将每一位加减法分配给一个一位全加器，采用串行进位方法，前位的进位信号直接作为输入送给下一位。至于减法的实现需要增加一个控制信号输入sub，通过异或门实现加减法，即加法时，sub=1，各位与1异或，不发生改变，且进位c=0；减法时，sub=1，各位与1异或，相当于取反，且前进位C\_in=1，达到了各位取反加一获得补码的效果。OF是该加减法的溢出检测，判断思路是检验最高位进位（符号位进位）与次高位进位是否一致，进位一致，异或输出为0，判断为没有溢出；进位不一致，异或输出为1，判断为溢出。电路图如2.1.1电路图所示。

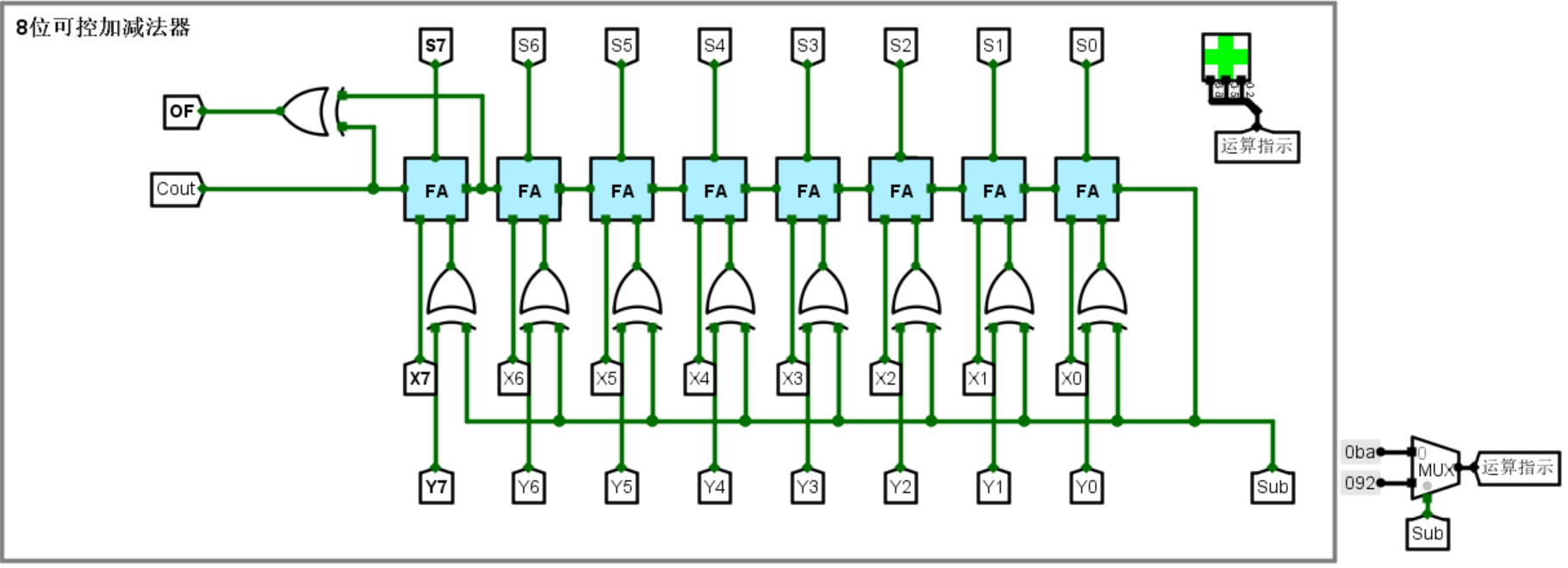


图2.1八位可控加减法器

2.1.2、 四位先行进位

设计思路：根据加法表达式：





4位先行进位逻辑表达式如下：



按照该表达式连接电路得到如图2.2所示电路图。

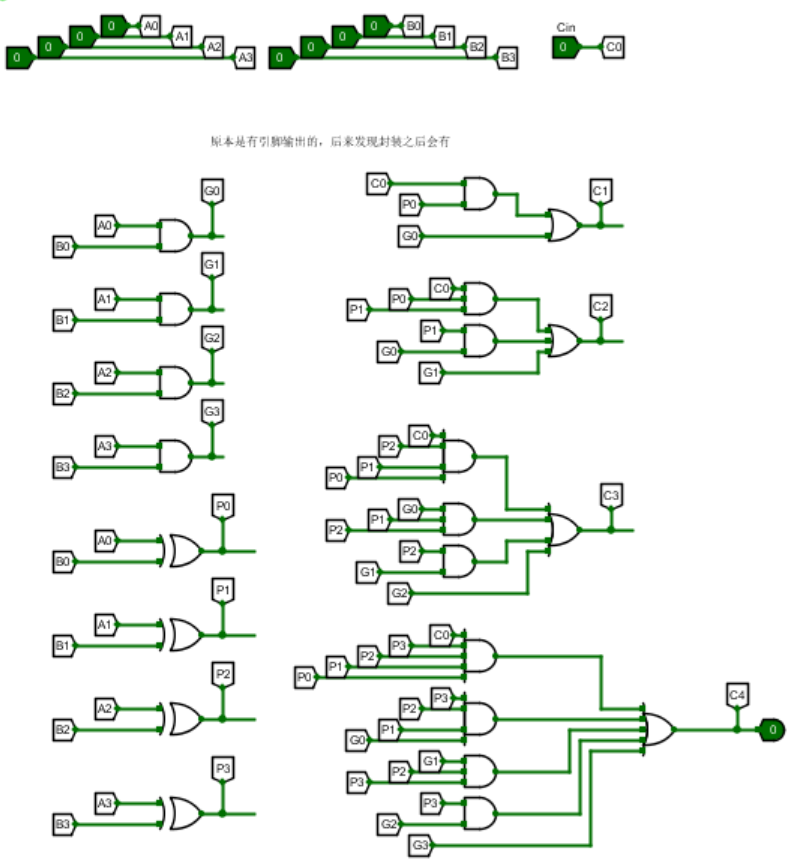


图2.2四位先行进位电路图

2.1.3、四位快速加法器

设计思路：在前面四位先行进位电路的基础上实现四位快速加法器只需要要对应的输入输出以及进位接好。电路图如2.3所示。

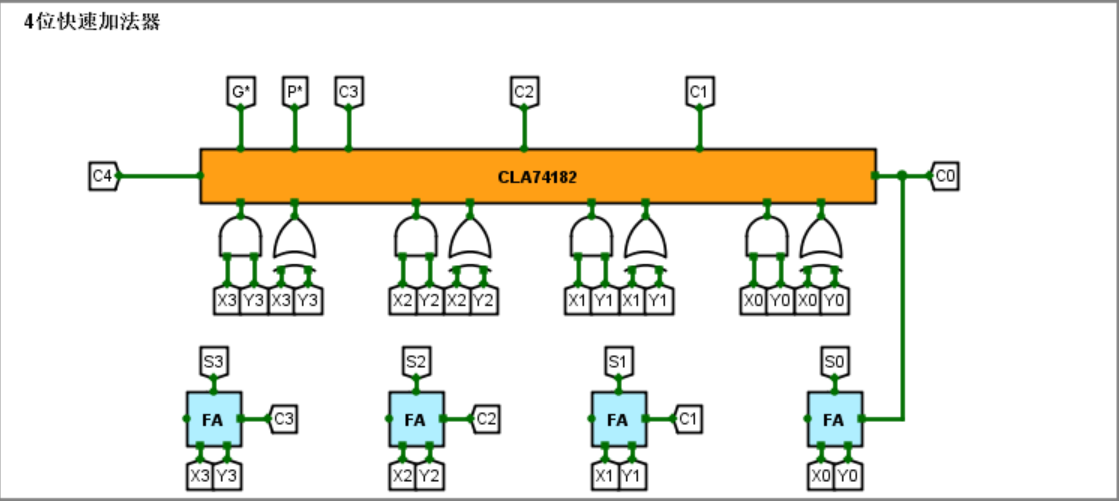


图2.3 四位快速加法器

2.1.3、 多位快速加法器

设计思路：16位和32位快速加法器都是在4位快速加法器基础上实现的，把4为快速加法器封装，16位快速加法器需要用4个4位快速加法器，32位快速加法器需要用2个四位先行进位电路，8个4位快速加法器。

**16位快速加法器：**将4个4位快速加法器输出的成组进位生成、传递函数G\*和P\*及C0连接到先行进位电路的输入端可提前产生C4、C8、C12、C16信号，再将对应信号连接到相应的快速加法器的进位输入端即可构成16位组内并行进位、组间并行进位的快速加法器。实验电路图如图2.4所示。

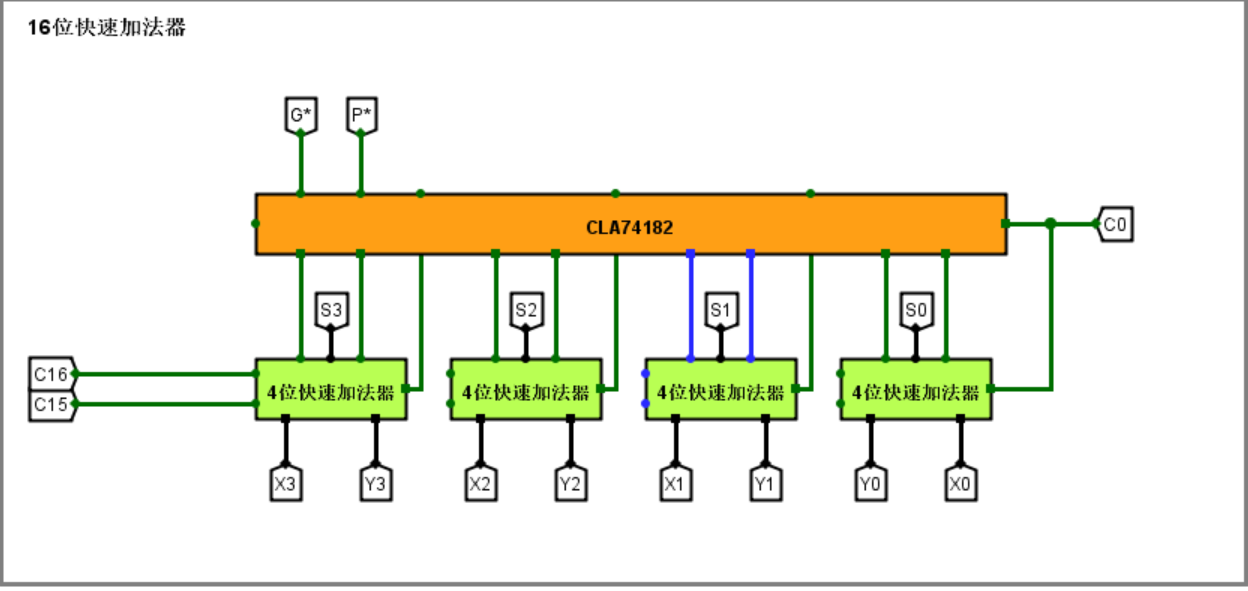


图2.4 十六位快速加法器

**32位快速加法器：**在实现16位快速加法器的基础上，将16位快速加法器进行组间串联连接即可实现32位快速加法器，实验电路图如图2.5所示。

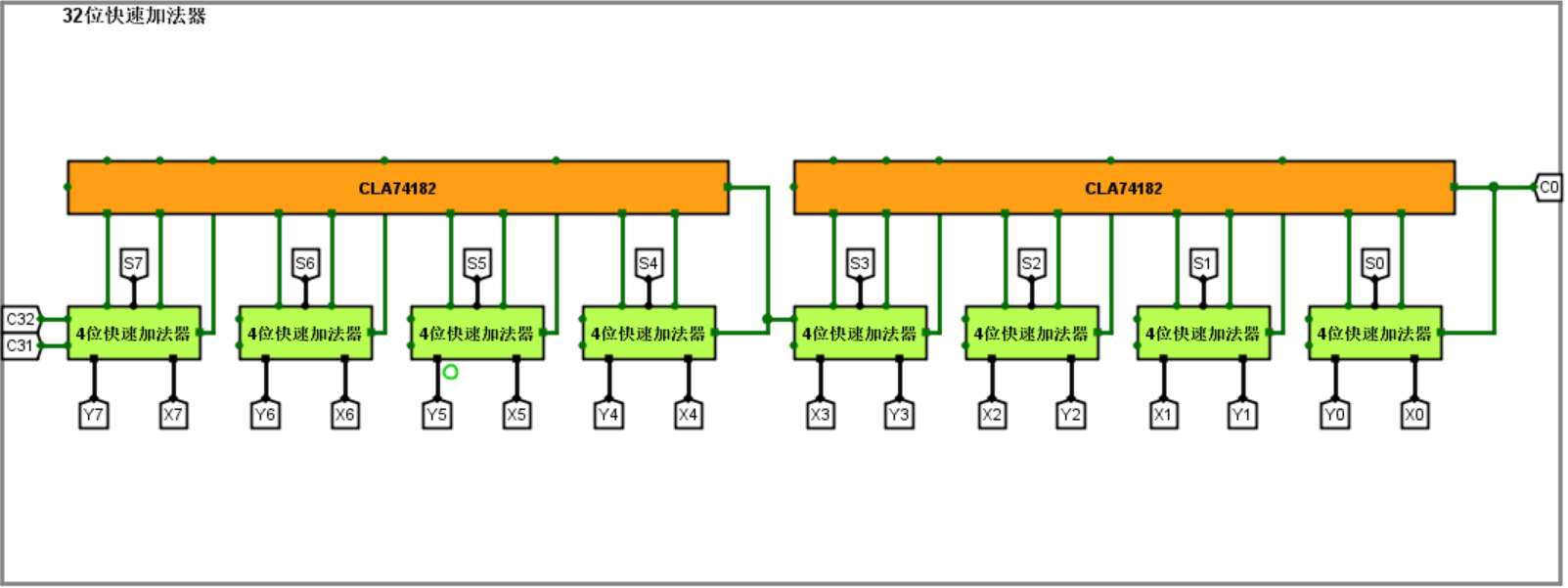


图2.5 三十二位快速加法器

2.2、阵列乘法器实验

2.2.1、五位阵列乘法器

用与门阵列同时产生手动乘法中的各乘积项，同时用[一位全加器](https://so.csdn.net/so/search?q=%E4%B8%80%E4%BD%8D%E5%85%A8%E5%8A%A0%E5%99%A8&spm=1001.2101.3001.7020)按照手动乘法运算的需要将每一列换上的数相加，构成全加器阵列。电路设计如图2.6所示。

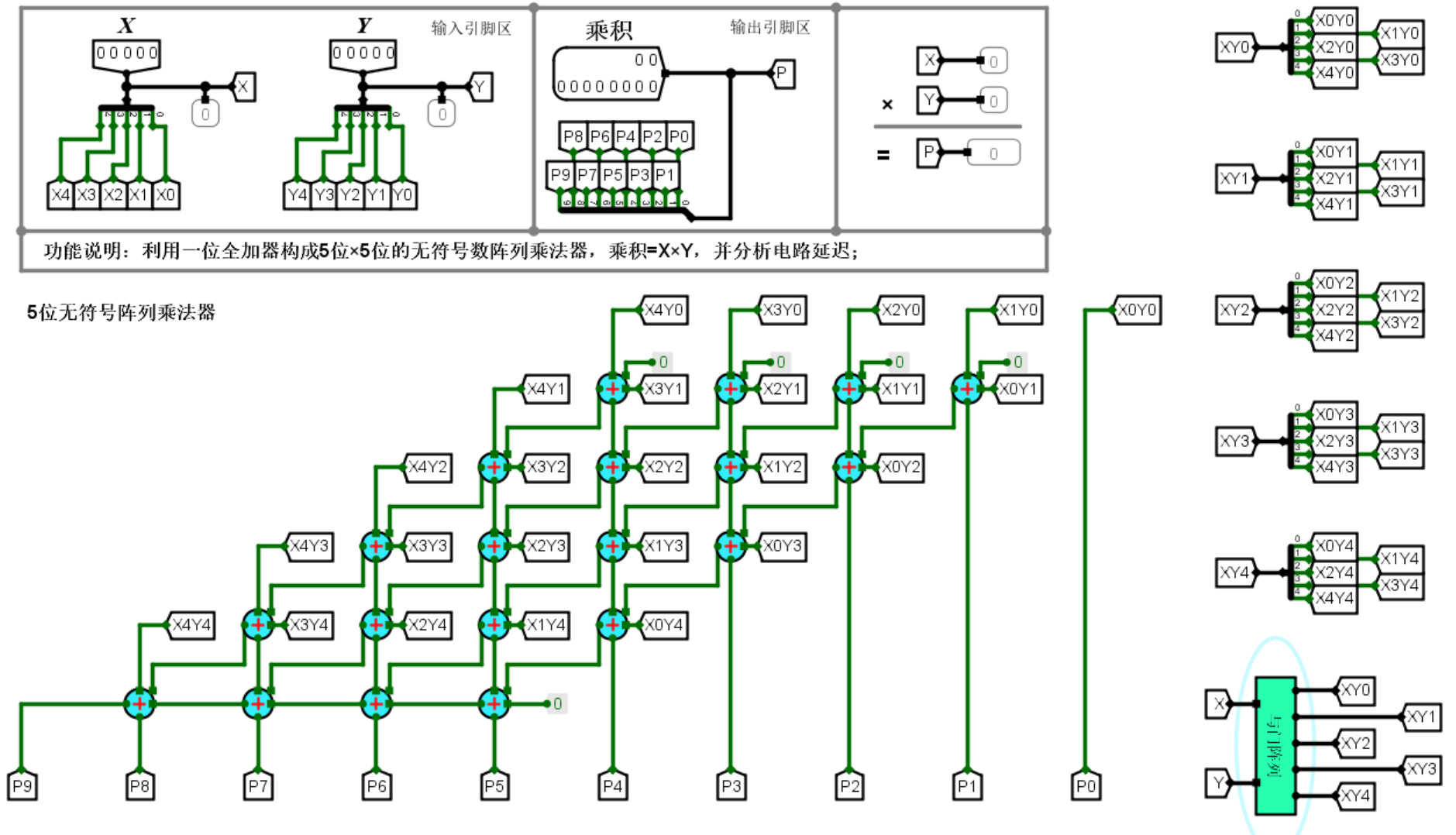


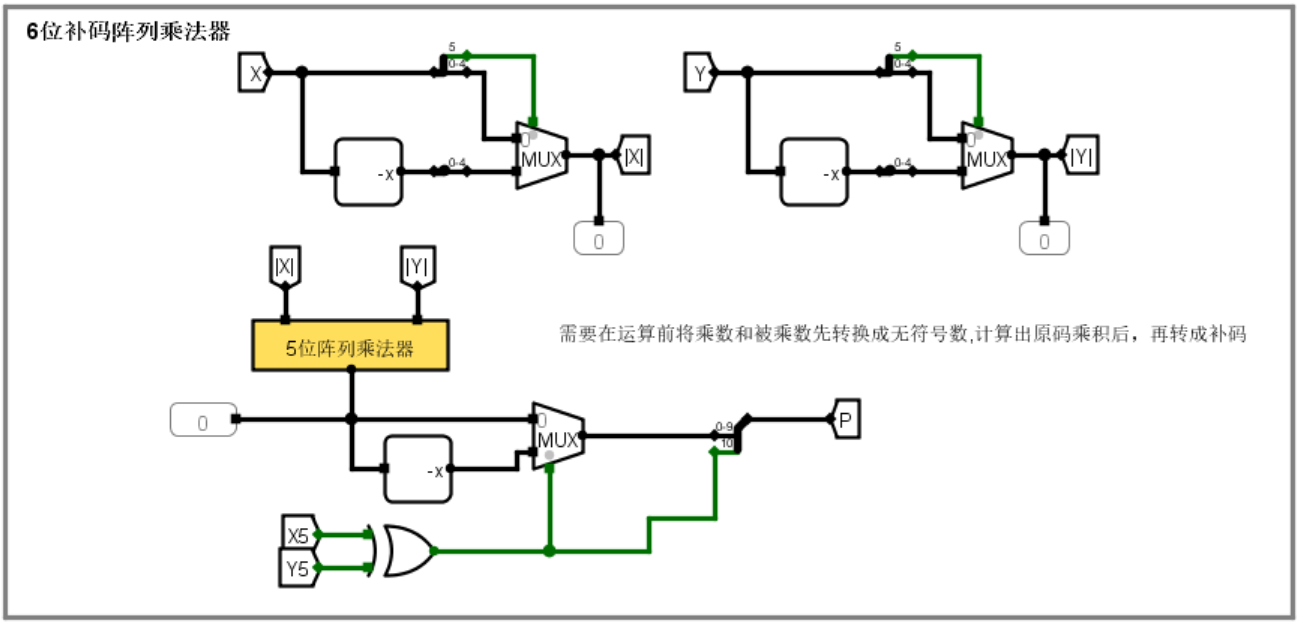
图2.6 无符号斜向进位阵列乘法器电路图

**时间延迟分析：**

延时发生在从左往右的全加器中，故5位无符号斜向进位阵列乘法器的延时：

2.2.2、6位补码阵列乘法器

为了利用上面的无符号斜向进位阵列乘法器，需要在运算前将乘数和被乘数先转换成无符号数,计算出原码乘积后，再转成补码。运算前分别用各自的符号位作为求补控制信号得到|X|,|Y|。将|X|,|Y|输入无符号五位阵列乘法器中，运算出结果后根据两个乘数的符号位判断是否求补。电路设计如图2.7所示。

图2.7 6位补码阵列乘法器

2.3、一位乘法器实验

2.3.1、原码一位乘法器

无符号原码一位乘法的计算： 维护一个部分积，初始为0，每次取乘数b的最低位，如果最低位为1，部分积=部分积+乘数a，如果最低位为0，部分积=部分积+0；然后将部分积右移一位，同时将乘数b右移一位。

寄存器存放部分积，寄存器存放乘数，并且最低位为判断位；计数器中计算时间周期数。，中的部分积和乘数随时钟右移。，计数周期为0时，将y载入寄存器中，进行第一次运算，右移1位。周期数大于8后，触发使能信号，使寄存器中的数不再随时钟信号改变。电路设计如图2.8所示。

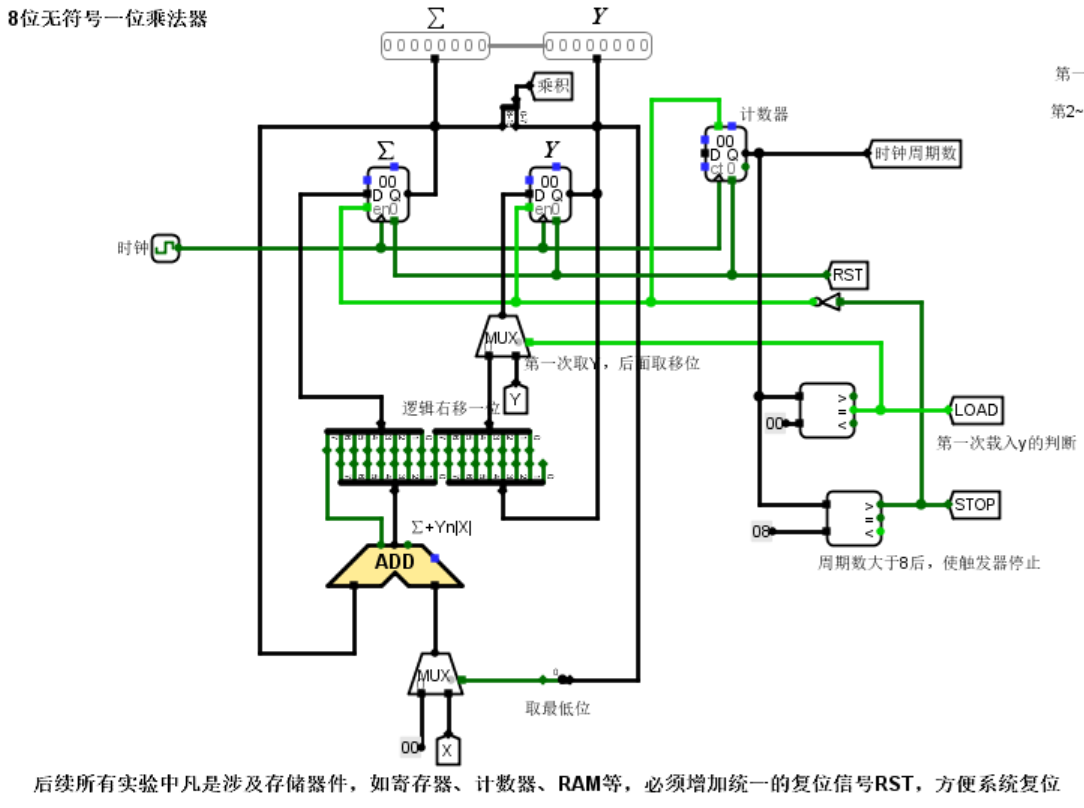


图2.8 8位无符号一位乘法器电路图

2.3.2、补码一位乘法器

与原码1位乘法大致电路设计原理大致相同, 不同的是booth1位乘法根据与的差值决定累加运算的参数是0还是或者是，部分积累加公式为。

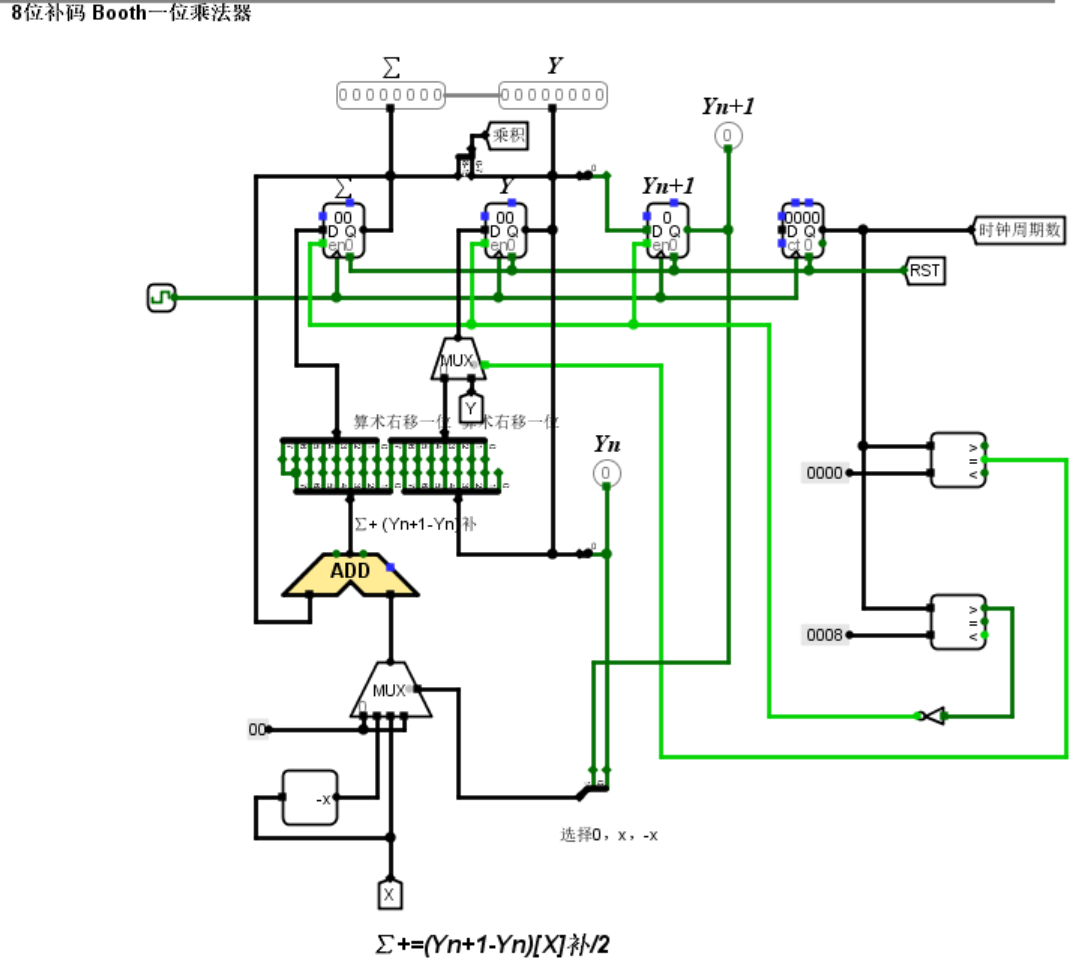


图2.9 8位booth一位乘法器电路图

2.4、ALU实验

2.4.1、算术逻辑单元ALU

ALU是根据OP字段来决定对应的功能，芯片引脚与对应关系如图3.1所示。



表格

描述已自动生成

图3.1 芯片引脚与对应关系

2.4.2、算术逻辑单元ALU电路

ALU算法单元并行进行各种算法单元运算，以算术操作符做为选择信号，选择最终的输出形式，所以，只需分别设计各操作并打好标签，用MUX控制输出即可完成ALU算法单元。电路设计如图

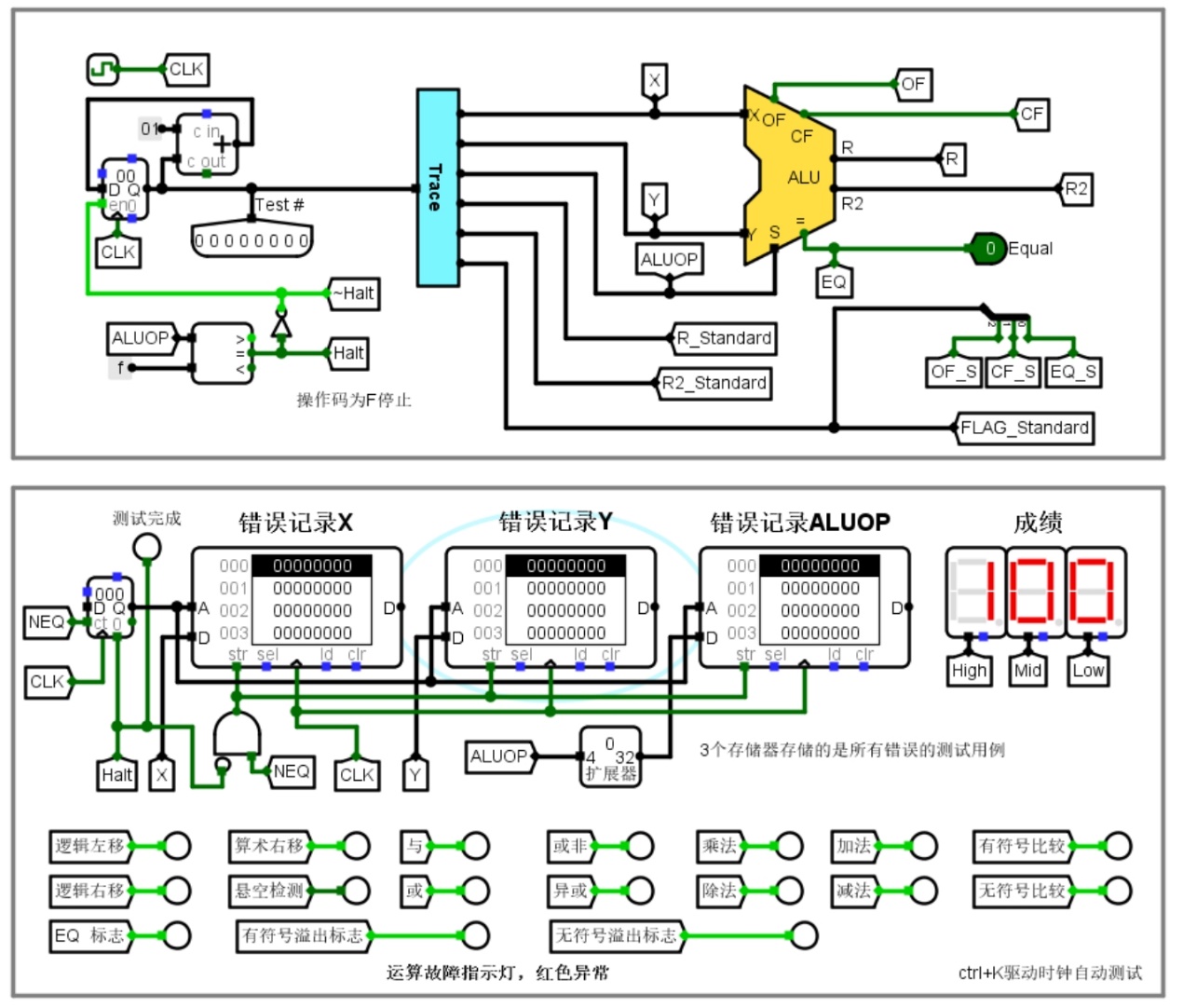
图示, 示意图

描述已自动生成

图3.2 算术逻辑单元ALU电路

2.4.3、测试结果

在ALU自动测试中运行结果如下图所示：

图3.3 ALU测试结果

每个时钟周期显示数字都是100，说明前面电路设计正确。

3、实验中的bug和解决方案

1、在设计原码一位乘法器时，一开始计数器的位数不足，导致计数器溢出后重新计数，使得运算结果发生改变，无法得到正确的结果。将位数多设几位就解决了。

2、在最后ALU自动测试实验发现分数并100，存在几处错误。根据下方的指示灯，到电路中排查电路逻辑并未发现问题，最后发现隧道标签与实际上的不一致。

3、在ALU中，溢出检测标志位OF的输出始终为0，无法正确检测溢出。检查溢出检测逻辑的连接，发现是溢出检测逻辑中的异或门输入信号接反。

4、Logisim仿真软件存在一些问题。试验过程中会出现明明电路正确，出现整块报错的问题。重启软件后才恢复正常。