

运算器设计实验

实验整体框架

快速加法器

- 4位可级联先行进位电路设计
- 4位快速加法器设计
- 16位快速加法器
- 32位快速加法器

乘法器

- 5位阵列乘法器
- 5位乘法流水线

32位算数逻辑运算单元

alu*	
	☆8位可控加减法器
	☆4位先行进位74182
	☆4位快速加法器
	☆16位快速加法器
	☆32位快速加法器
	☆5位阵列乘法器
	☆6位补码阵列乘法器
	☆5位无符号乘法流水线
	☆原码一位乘法器
	☆补码一位乘法器
	☆算术逻辑运算单元ALU
	★ALU自动测试
	★5位乘法与门阵列
	★乘法线流水接口
	★一位全加器FA封装1
	★一位全加器FA封装2
	★一位全加器FA封装3
	★8位串行加法器

打开资源包中的“alu.circ”，可以看到已经建立的电路（白色星为待编辑，黑色星为已完成待调用）

一、快速加法器实验

实验目标

- 掌握并验证串行加法器的逻辑实现
- 理解全并行进位计算的基本原理，能设计快速加法器
- 在Logisim中实现快速加法器

组内先行、组间先行

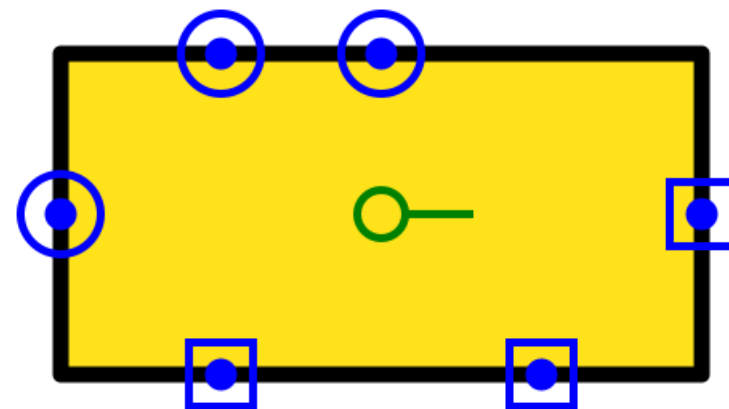
实验任务

- 8位可控加减法电路（已设计，测试验证）
- 4位先行进位电路、4位快速加法器
- 16位、32位快速加法器

(一)：8位可控加减法电路

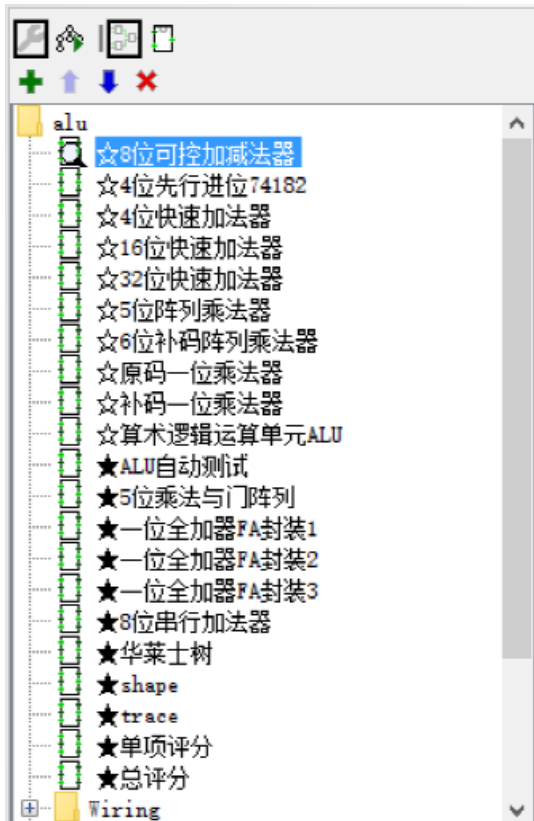
8位可控加减法电路的外部特性

- 输入：8位的X，Y；
1位的功能选择Sub
- 输出：8位的计算结果S；
1位的进位Cout；
1位的溢出判断Overflow (OF)
- 功能： $S = (Sub == 0) ? (X + Y) : (X - Y)$
当Sub为0，进行加法；否则，进行减法运算

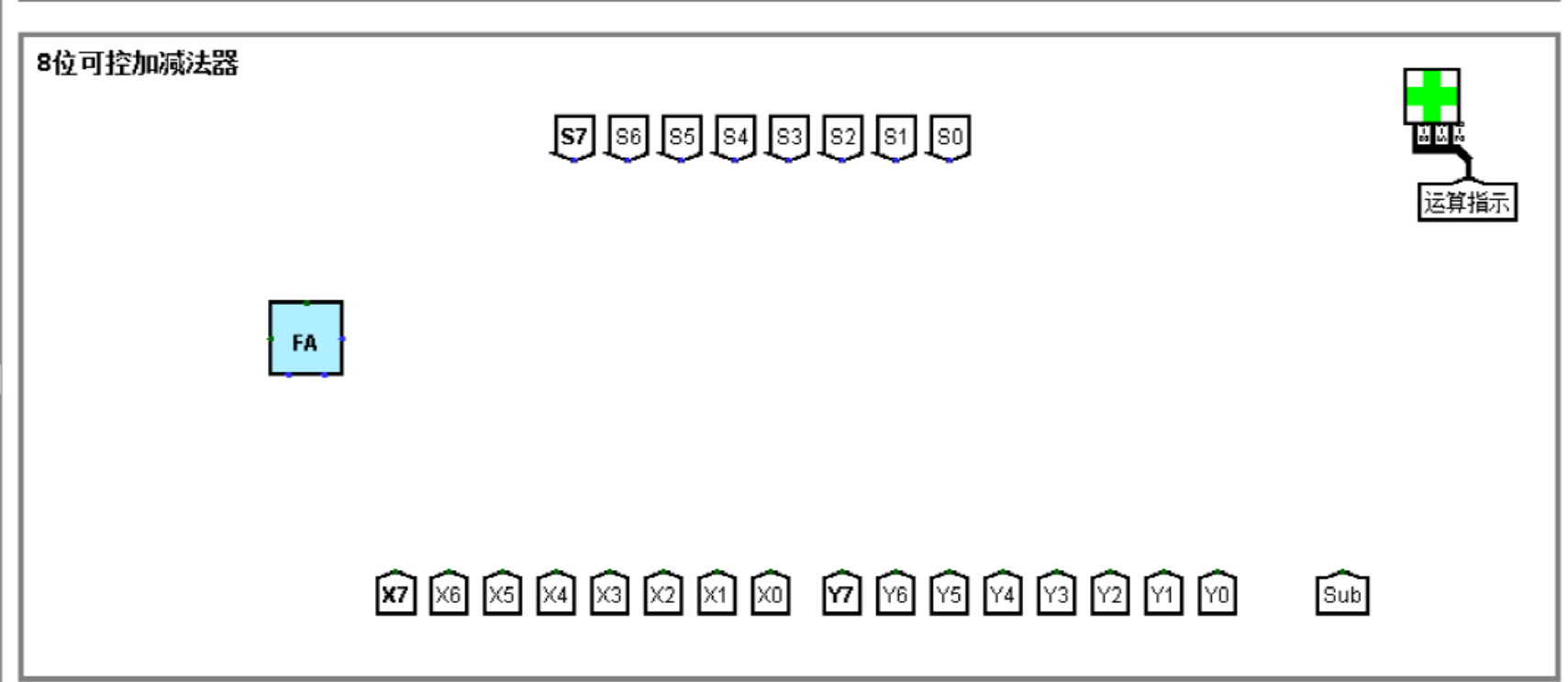
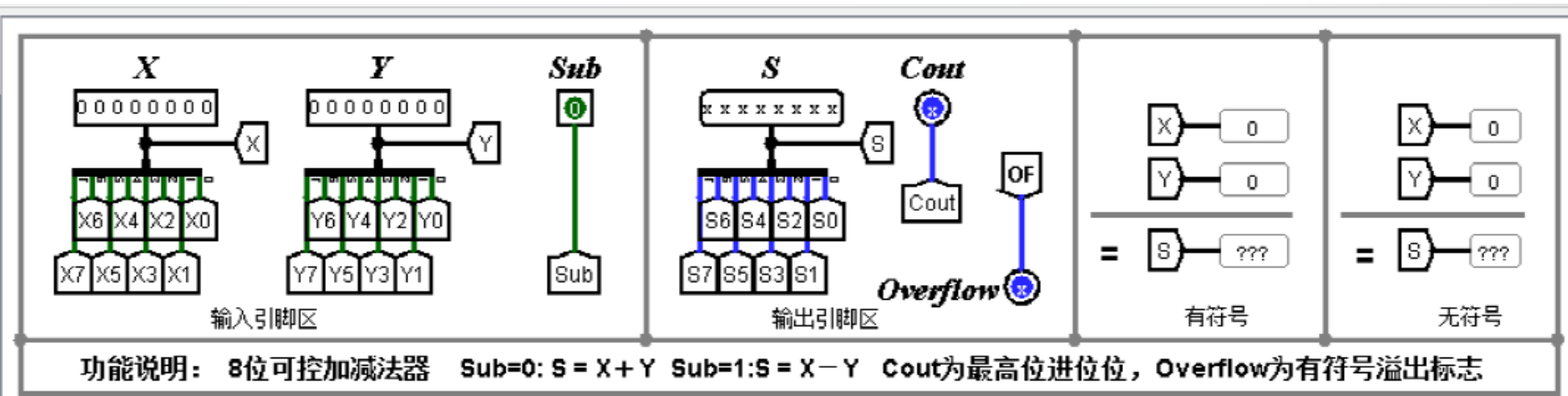


加、减如何统一起来？

利用已经定义好的隧道标签，选择适当的逻辑门、调用已经封装好的1位全加器FA完成电路的设计与连接



Circuit: ☆8位可控加减法器	
Circuit Name	☆8位可控加减法器
Shared Label	CAS
Shared Label Facing	North
Shared Label Font	SansSerif Bold 12



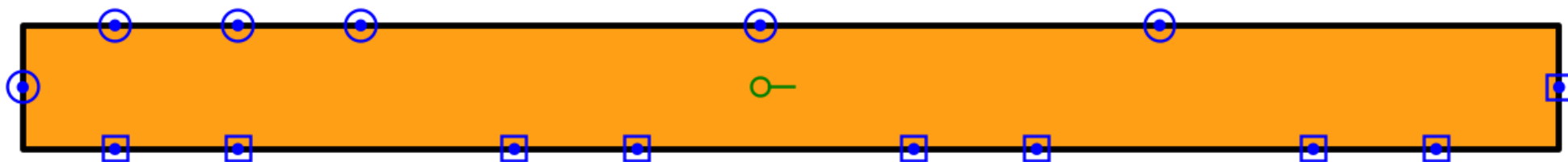
(二)：4位先行进位74182 设计

74182的外部特性

- 输入：4组的P、G信号（生成函数与传递函数）；
- 输出：4位的先行进位信号C4-C1
成组的生成函数与传递函数信号——P*、G*
- 功能：实现先行进位电路

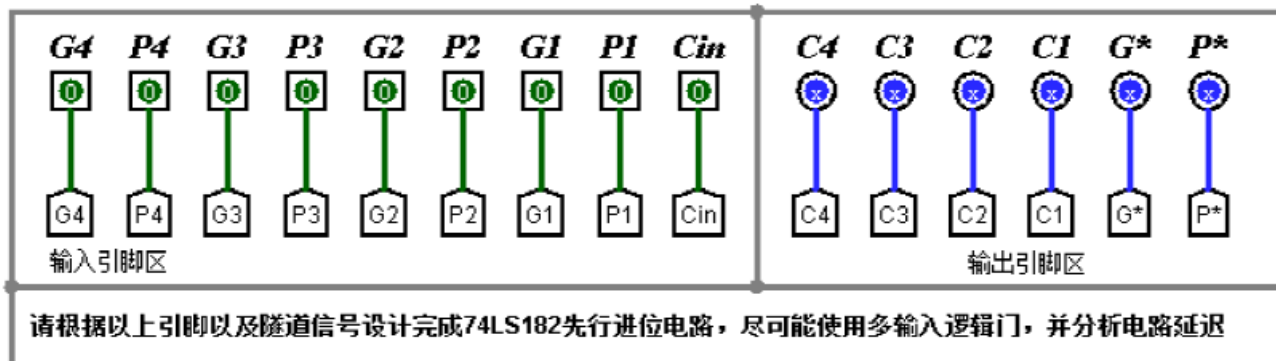
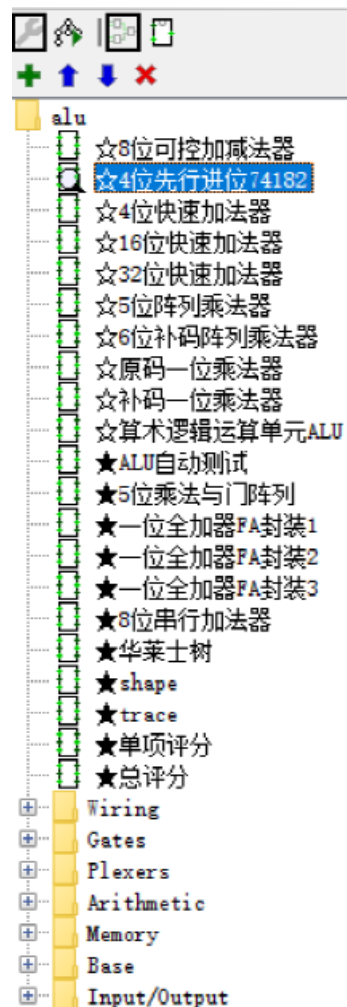
Tip:

可借助Logisim的电路分析功能，填入输出信号的逻辑表达式

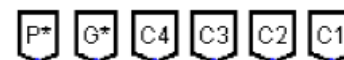


可根据“加减法器设计原理.pdf”，回顾相关知识，完成逻辑表达式的设计

利用已经定义好的隧道标签，选择适当的逻辑门完成电路的设计与连接



4位先行进位电路



(三)：4位的快速加法器

4位快速加法器的外部特性

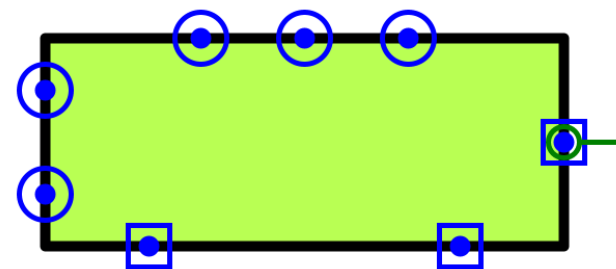
- 输入：4位的输入 X , Y ；进位输入 C_0
- 输出：4位运算结果 S ；最高位进位输出 C_4 （1位），次高位进位输出 C_3 （1位）

成组的生成函数与传递函数信号—— P^* 、 G^*

- 功能：实现2个4位数的加法 $S = X + Y$ 。

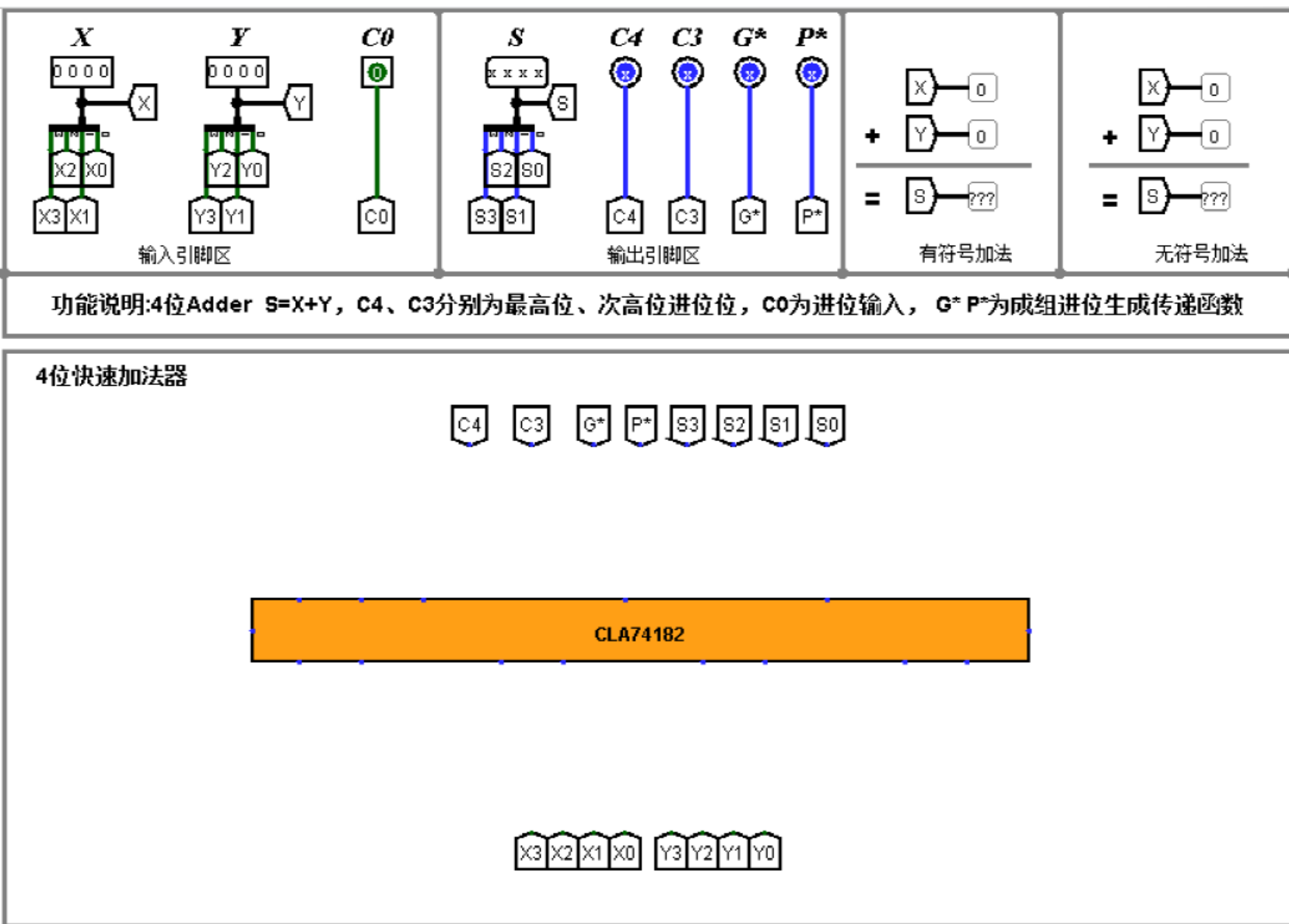
$$\square \quad S_i = A_i \oplus B_i \oplus C_i$$

□ 进位部分利用已完成的74182实现



利用已经定义好的隧道标签，选择适当的逻辑门、已设计的74182子电路，完成电路的设计与连接

- ☆8位可控加减法器
- ☆4位先行进位74182
- ☆4位快速加法器
- ☆16位快速加法器
- ☆32位快速加法器
- ☆5位阵列乘法器
- ☆6位补码阵列乘法器
- ☆原码一位乘法器
- ☆补码一位乘法器
- ☆算术逻辑运算单元ALU
- ★ALU自动测试
- ★5位乘法与门阵列
- ★一位全加器FA封装1
- ★一位全加器FA封装2
- ★一位全加器FA封装3
- ★8位串行加法器
- ★华莱士树
- ★shape
- ★trace
- ★单项评分
- ★总评分
- Wiring
- Gates
- Plexers
- Arithmetic
- Memory
- Base
- Input/Output



（四）：16位的快速加法器

16位快速加法器的外部特性

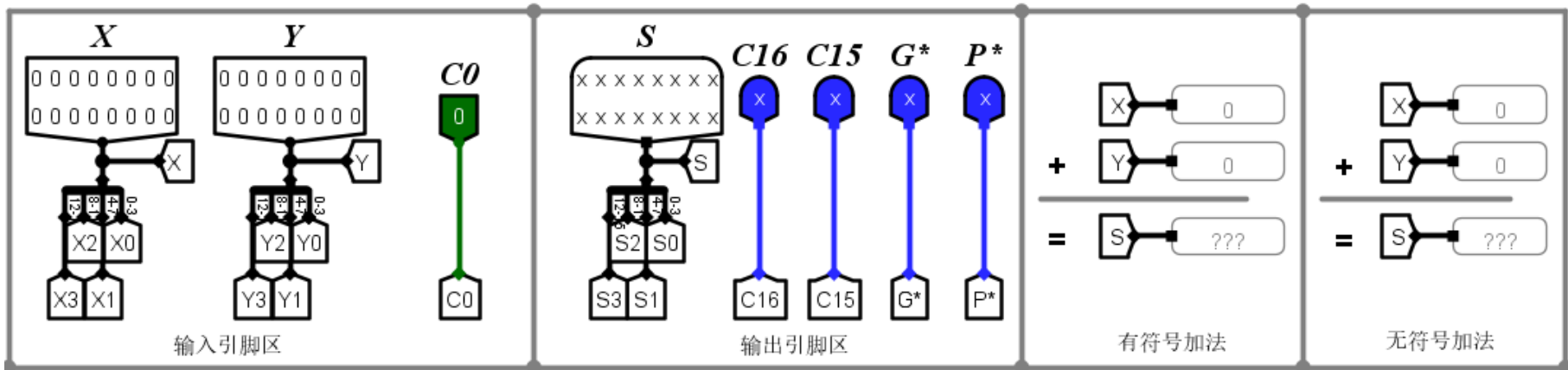
- 输入：16位的输入 X ， Y ；进位输入 C_0
- 输出：16位运算结果 S ；最高位进位输出 C_{16} ，次高位进位输出 C_{15}

成组的生成函数与传递函数信号—— P^* 、 G^*

- 功能：实现2个4位数的加法 $S = X + Y$ 。

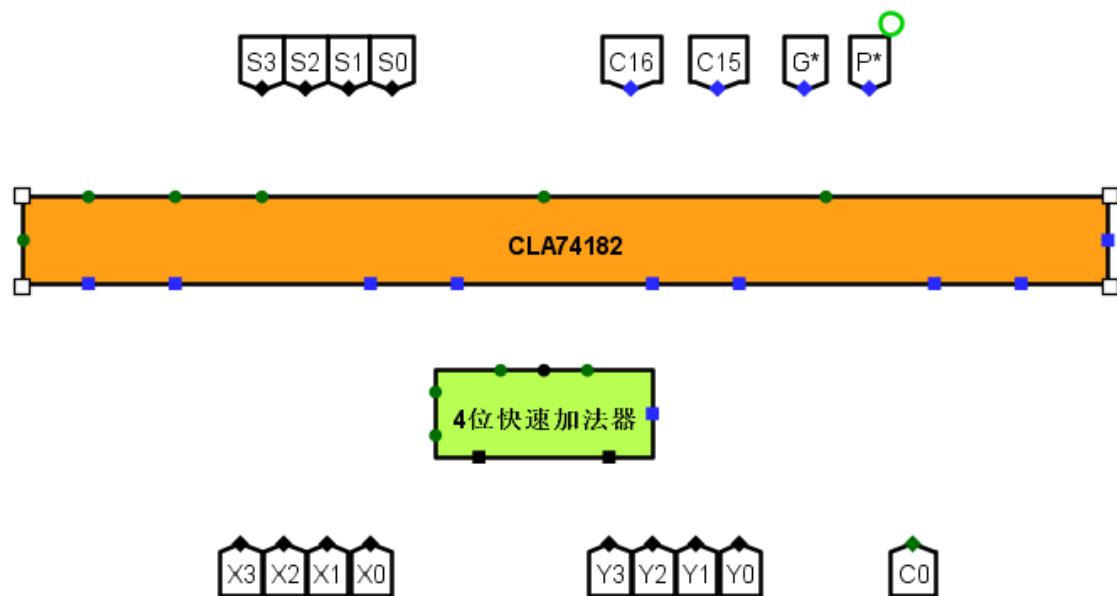
设计思路：利用4组4位快速加法器构成，采用组内先行进位、组间先行进位方式

利用已经定义好的隧道标签，选择4个4位快速加法器、74182、适当的逻辑门和线路，完成电路的设计与连接



功能说明：16位Adder $S=X+Y$ ， $C16$ 、 $C15$ 分别为最高位、次高位进位位， $C0$ 为进位输入， $G^* P^*$ 为成组进位生成传递函数

16位快速加法器

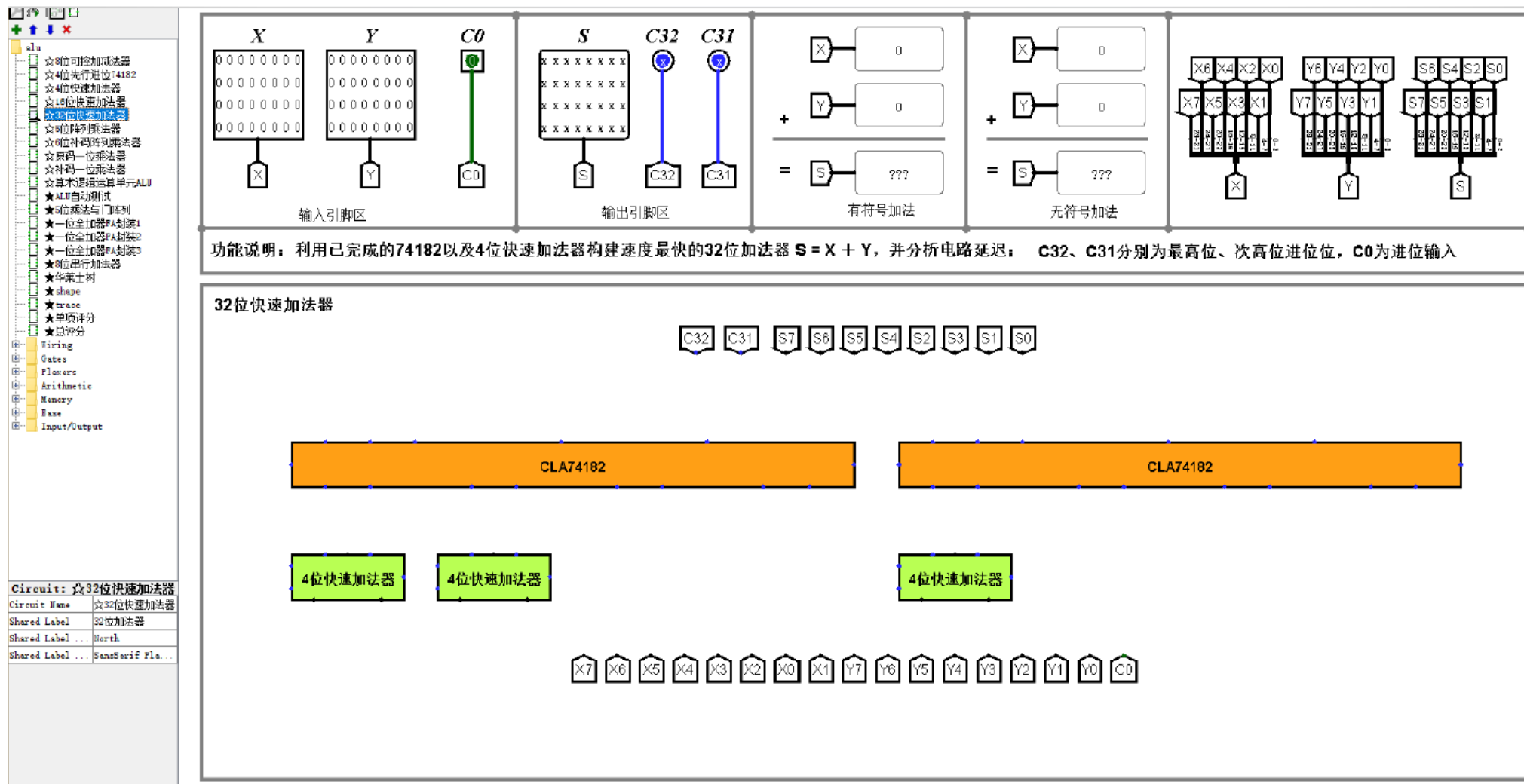


(五)：32位的快速加法器

将2个16位
加法器级联

组间进位如
何处理？串
行 or 先行？

请选择一种
方案并进行
实现



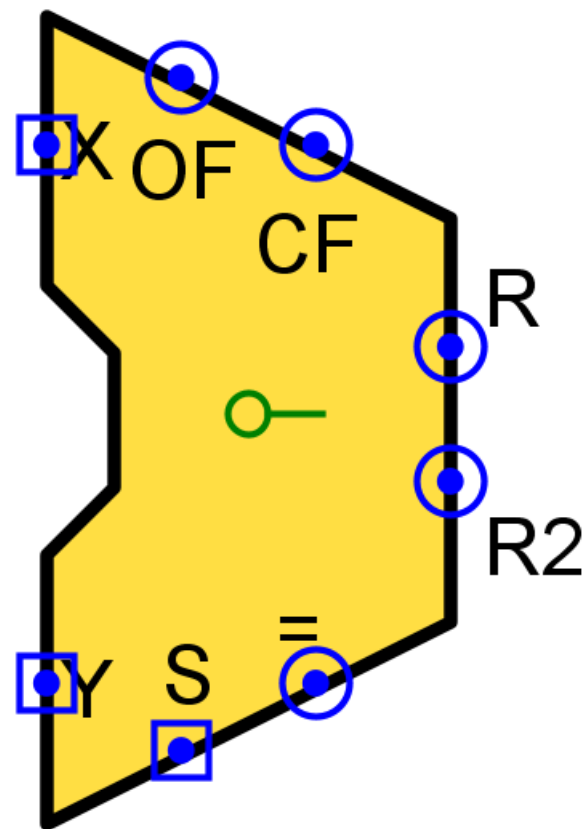
二、32位算数逻辑运算单元实验

实验目标

- 掌握定点数加、减法溢出检测方法
- 理解算数逻辑运算单元ALU的基本构成
- 熟悉Logisim中的各种运算组件
- 熟悉并掌握多路选择器的使用

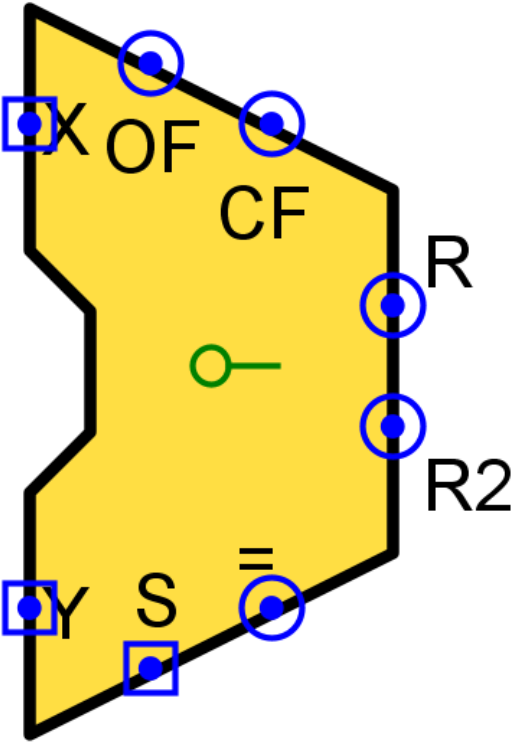
实验任务

- 用简单方式，设计32位ALU
 - ❑ 运用已完成的32位加法器、logisim的其他运算组件进行集成
 - ❑ 禁止使用logisim内置加法器、减法器



32位ALU的外部特性

ALU进行什么运算，由 引脚S（ALU_OP） 决定



引脚	输入/输出	位宽	功能描述
X	输入	32	操作数X
Y	输入	32	操作数Y
S	输入	4	运算操作码 ALU_OP
R	输出	32	ALU运算结果
R2	输出	32	ALU结果第二部分，用于乘法运算结果高位或除法运算的余数位，其它运算时值为零
OF	输出	1	有符号加减运算溢出标记，其他运算为0
UOF	输出	1	无符号加减运算溢出标记，其它运算为0
Equal	输出	1	Equal=(x==y)?1:0, 对所有运算均有效

32位ALU的功能表

ALU_OP	十进制	运算功能
0000	0	$R = X \ll Y$ 逻辑左移 (Y取低五位) $R_2=0$
0001	1	$R = X \gg Y$ 算术右移 (Y取低五位) $R_2=0$
0010	2	$R = X \gg Y$ 逻辑右移 (Y取低五位) $R_2=0$
0011	3	$R = (X * Y)_{[31:0]}$ $R_2 = (X * Y)_{[63:32]}$ 无符号乘法
0100	4	$R = X/Y$ $R_2 = X \% Y$ 无符号除法
0101	5	$R = X + Y$ (Set OF/UOF)
0110	6	$R = X - Y$ (Set OF/UOF)
0111	7	$R = X \& Y$ 按位与
1000	8	$R = X Y$ 按位或
1001	9	$R = X \oplus Y$ 按位异或
1010	10	$R = \sim(X Y)$ 按位或非
1011	11	$R = (X < Y) ? 1 : 0$ 有符号比较
1100	12	$R = (X < Y) ? 1 : 0$ 无符号比较

Y控制X移位的位数。X是32位，所以

Y只需低5位

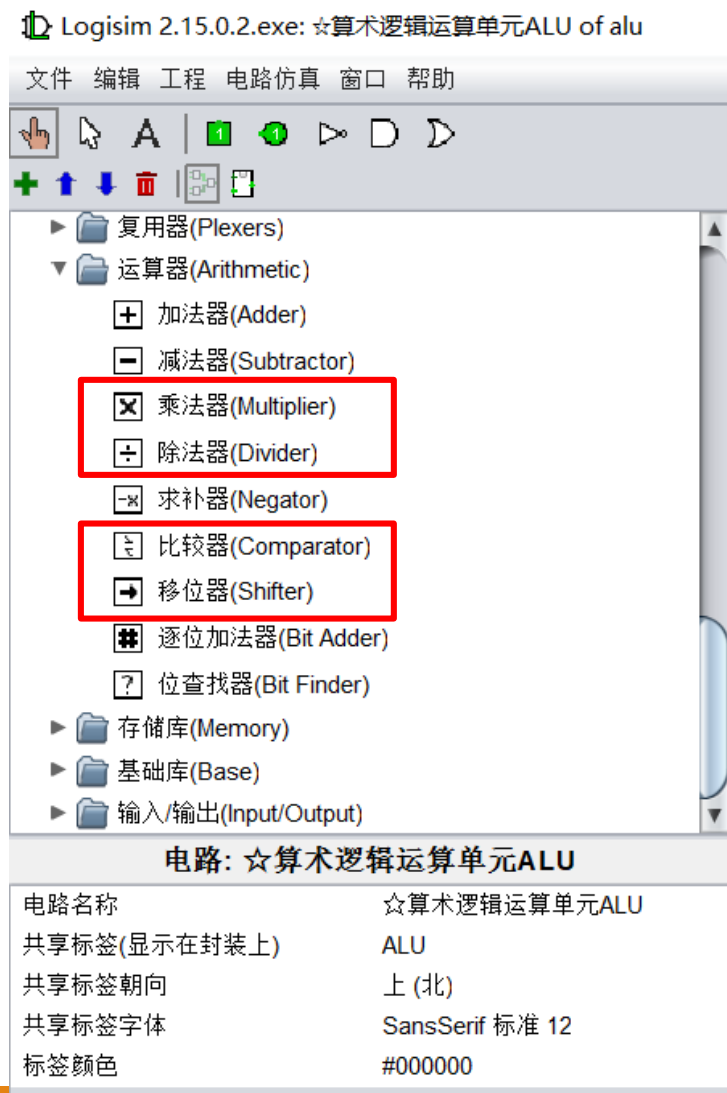
OF: 有符号数运算的溢出标志；如何产生？

UOF: 无符号数运算的溢出标志，即最高位的进位输出

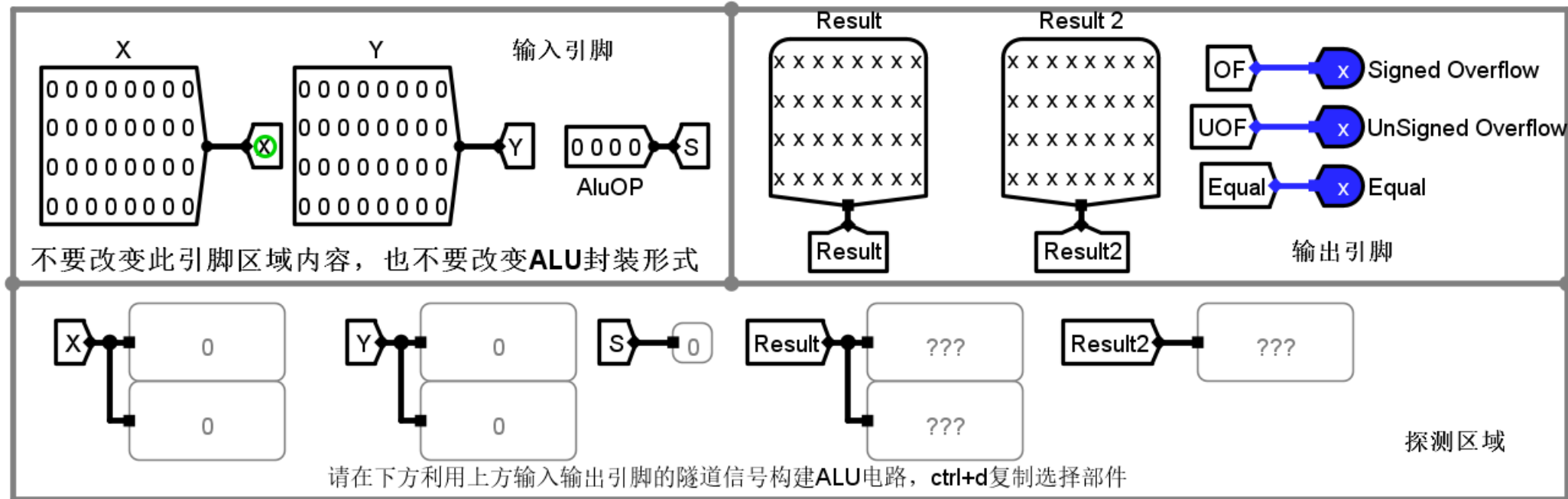
当X<Y，则R=1（00...001）；否则R=0（00...000）

设计思路： 分别实现各个功能后， 再进行集成

1. 对输入的ALU_OP进行译码， 产生不同运算的控制信号
2. 对照功能表， 分别实现X和Y的不同运算功能电路
 - 利用Logisim的运算组件
 - 利用已实现的32位快速加法器
 - 利用逻辑门和线路库
3. 使用多路选择器、三态门或者触发器等， 集成各个功能
利用 1中译码得到的控制信号， 选择 2中的运算结果，
作为最终ALU的输出



32位ALU电路 编辑界面

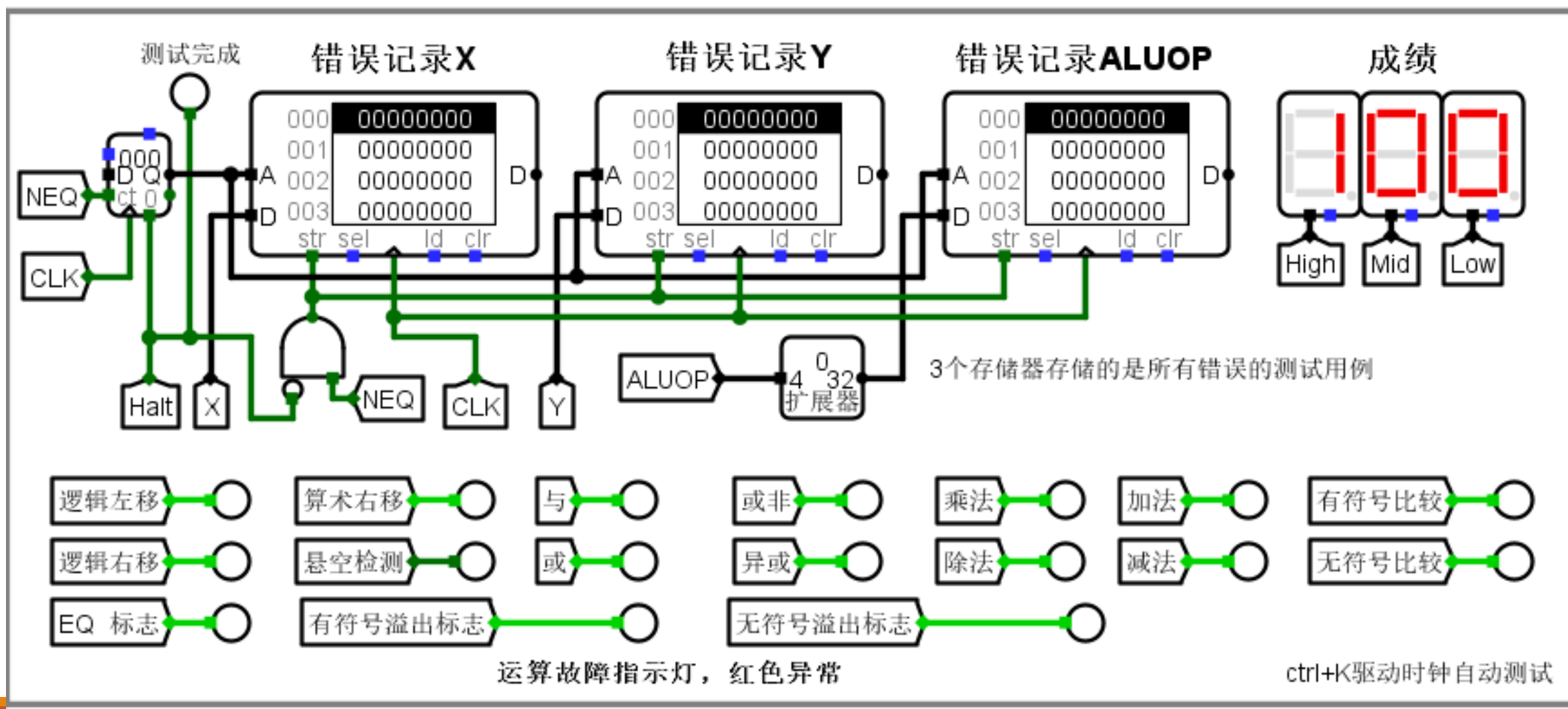


华中科技大学计算机学院 计算机组成原理，鸣谢加州大学伯克利CS61C课程

32位MIPS运算器

- 注意：
- 1、Result2 只有在除法、乘法时产生对应值，其余默认为0
 - 2、OF、UOF只有在加法、减法时产生对应值，其余默认为0
 - 3、Equal 是所有运算中都要计算
 - 4、避免引脚悬空不作设置

ALU电路的自动测试



三、阵列乘法器设计实验

实验目标

- 理解阵列乘法器的实现原理
- 熟悉流水线传输机制、体会流水线设计思路，为流水cpu设计做准备

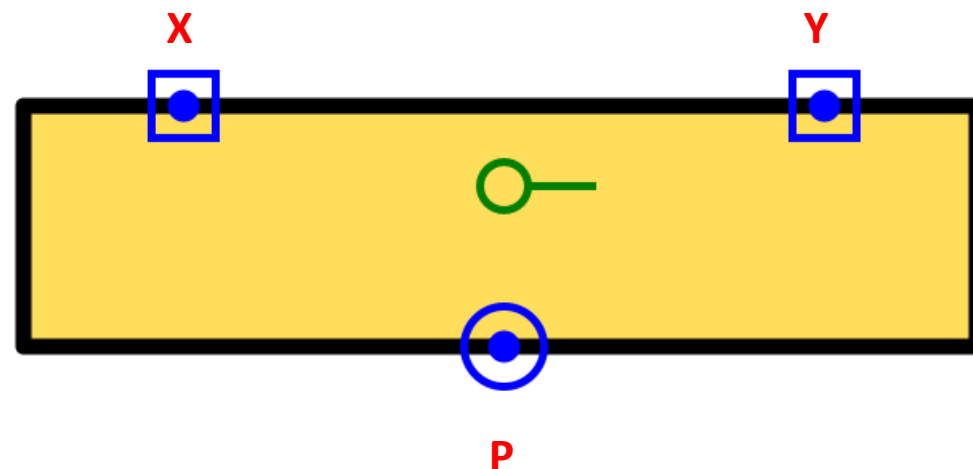
实验任务

- 设计5位无符号阵列乘法器
- 设计6位补码阵列乘法器
- 设计实现5位无符号乘法流水线 进阶任务

(一)：5位无符号阵列乘法器

5位无符号阵列乘法器的外部特性

- 输入：5位的无符号数 X ， Y ；
- 输出：10位运算结果 P ；
- 功能： $P = X * Y$



设计思路：

- 1、构造封装好的“5位乘法与门阵列”已产生 X 、 Y 各位组合后的所有求和项
- 2、利用构造封装好的“1位全加器FA封装3”来构造阵列乘法器

利用已经定义好的隧道标签，添加1位FA，完成电路的设计与连接

Logisim 2.15.0.2.exe: ☆5位阵列乘法器 of alu

文件 编辑 工程 电路仿真 窗口 帮助

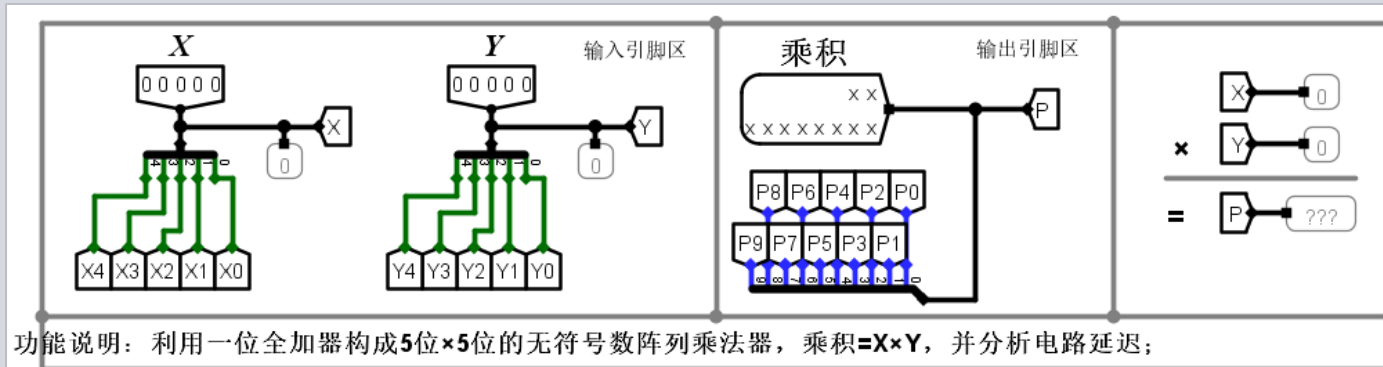


- ☆32位快速加法器
- ☆5位阵列乘法器
- ☆6位补码阵列乘法器
- ☆5位无符号乘法流水线
- ☆原码一位乘法器
- ☆补码一位乘法器
- ☆算术逻辑运算单元ALU
- ★ALU自动测试
- ★5位乘法与门阵列
- ★乘法线流水接口
- ★一位全加器FA封装1
- ★一位全加器FA封装2
- ★一位全加器FA封装3

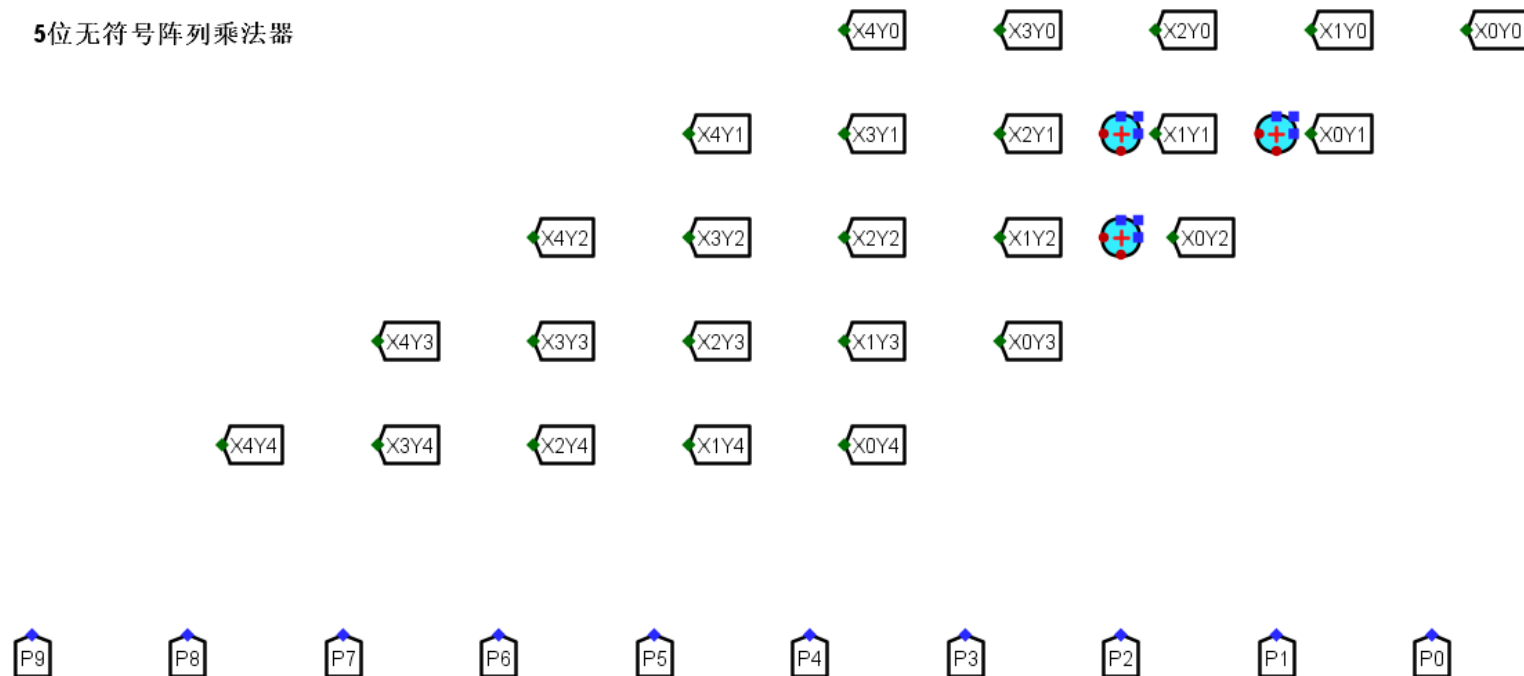
电路: ☆5位阵列乘法器

电路名称: ☆5位阵列乘法器
共享标签(显示在封装上): 5位阵列乘法器
共享标签朝向: 上(北)
共享标签字体: SansSerif 标准 12
标签颜色: #000000

Zoom: 100%



5位无符号阵列乘法器



进位横向传递或
主要斜向传递

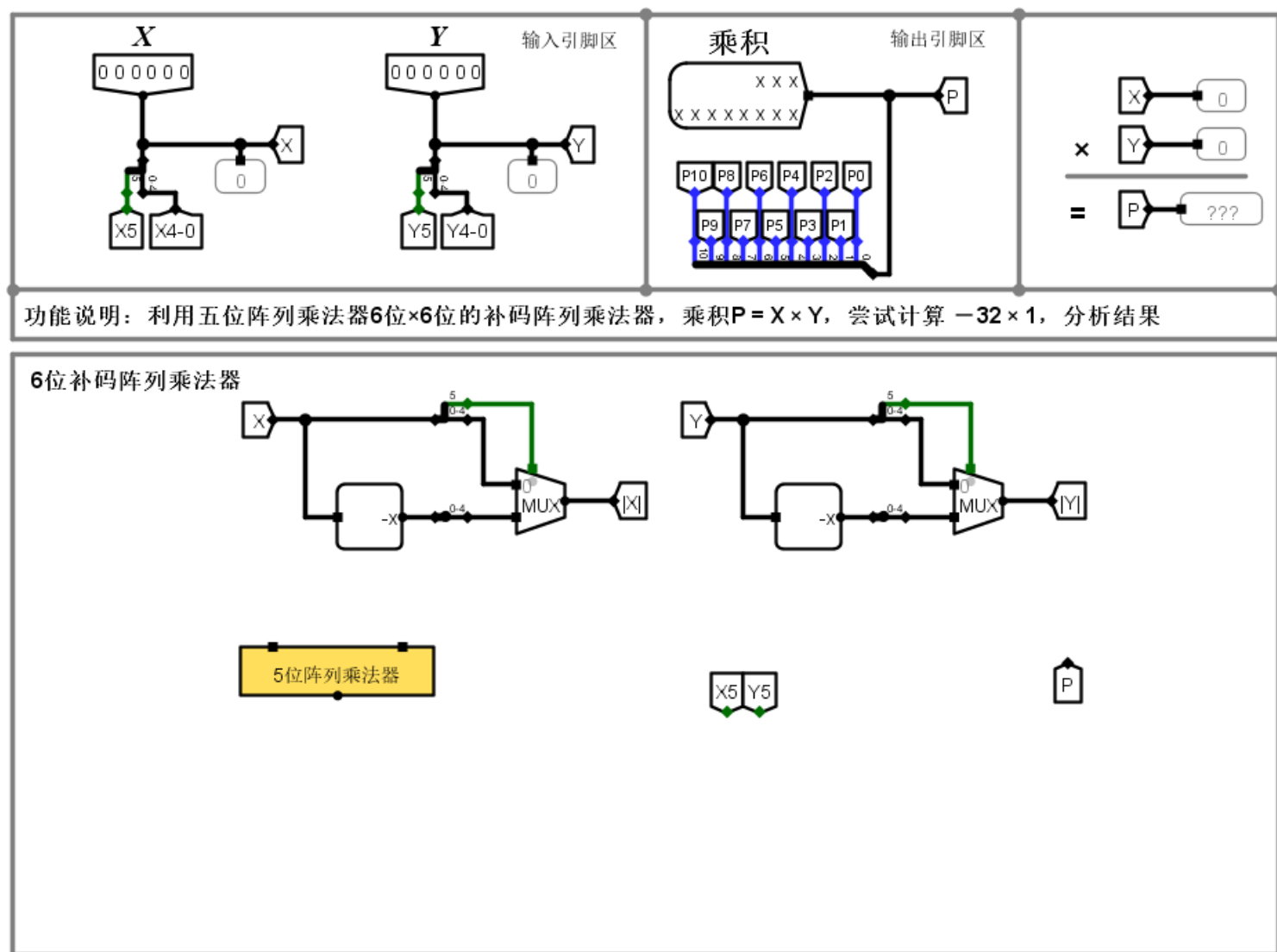
(二)：6位补码阵列乘法器设计

利用：

- 已经定义好隧道标签、求补器电路、5位阵列乘法器完成电路的设计与连接
- 乘积p的数值位与符号位要如何产生？

需测试： $-32 * 1$

记录现象，分析原因？



可根据“阵列乘法器设计原理.pdf”回顾相关原理知识

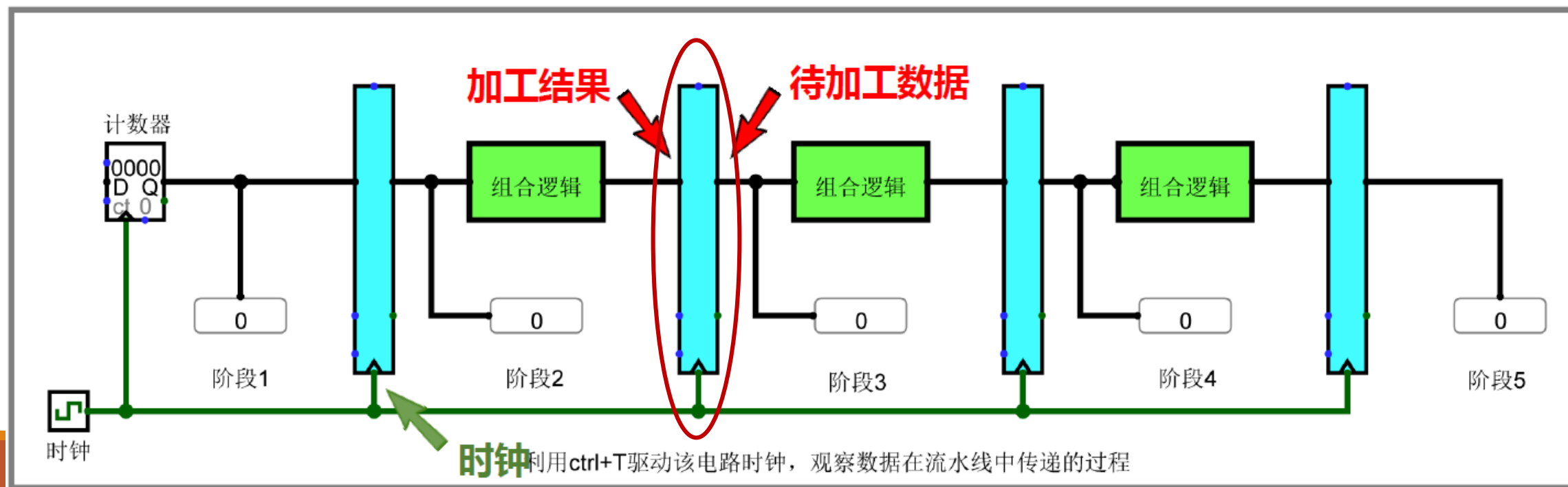
(三)：5位无符号乘法流水线

流水线：复杂问题分解成细粒度任务并发

流水线的基本框架

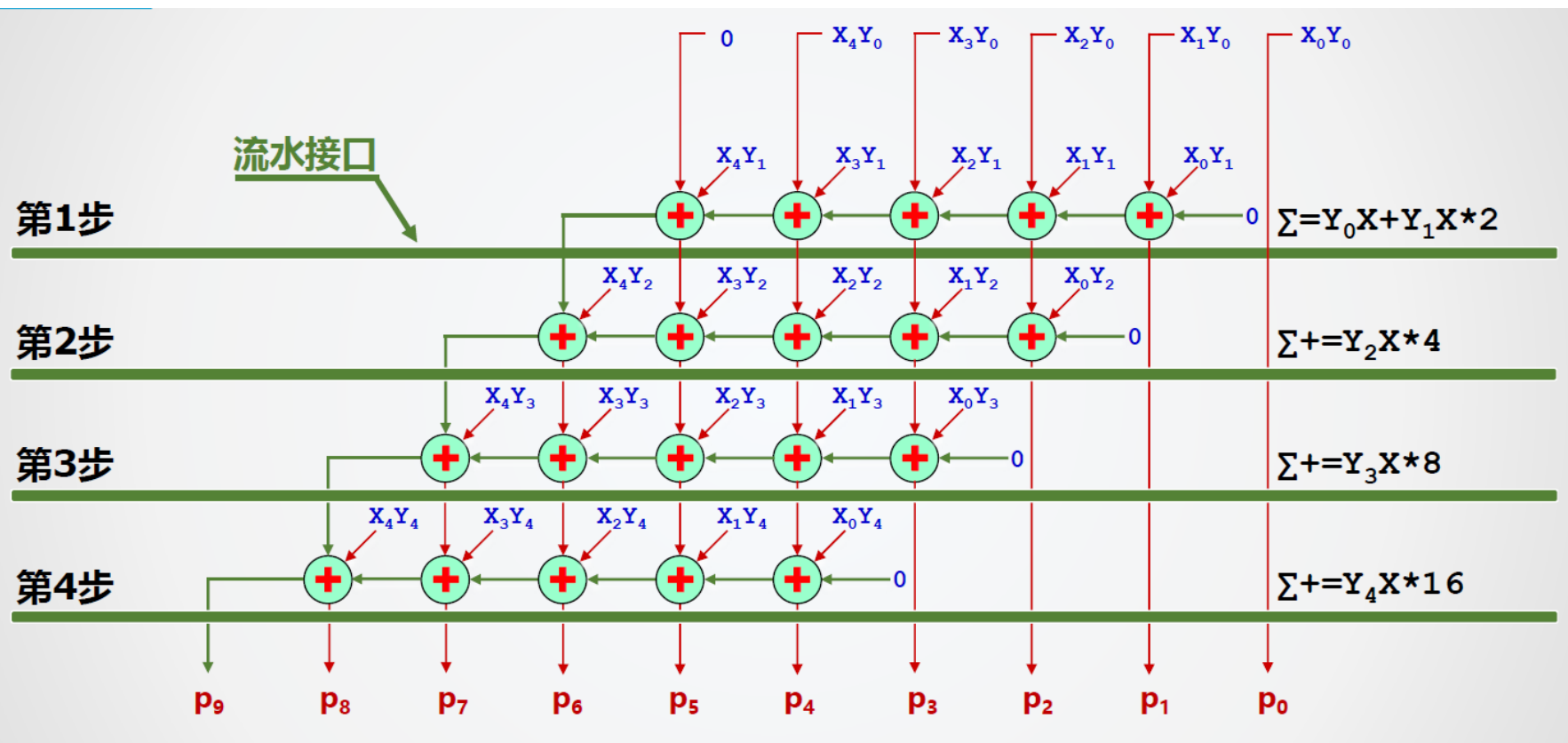
流水线 = 寄存器 + 组合逻辑 + 寄存器 + 组合逻辑 + 寄存器 ...

数据通路串联



乘法流水线设计思路

5位无符号阵列乘法器的优化



1. 分为4个步骤

2. 每1步运算时，

➤ 求和数需进行位扩充

➤ Y_iX 的*2、*4...靠移位器

实现（移动几位？）

➤ 传递给流水接口的

是什么？

乘法流水接口作用：寄存器，锁存当前累加结果和后续步骤需要的 Y_iX

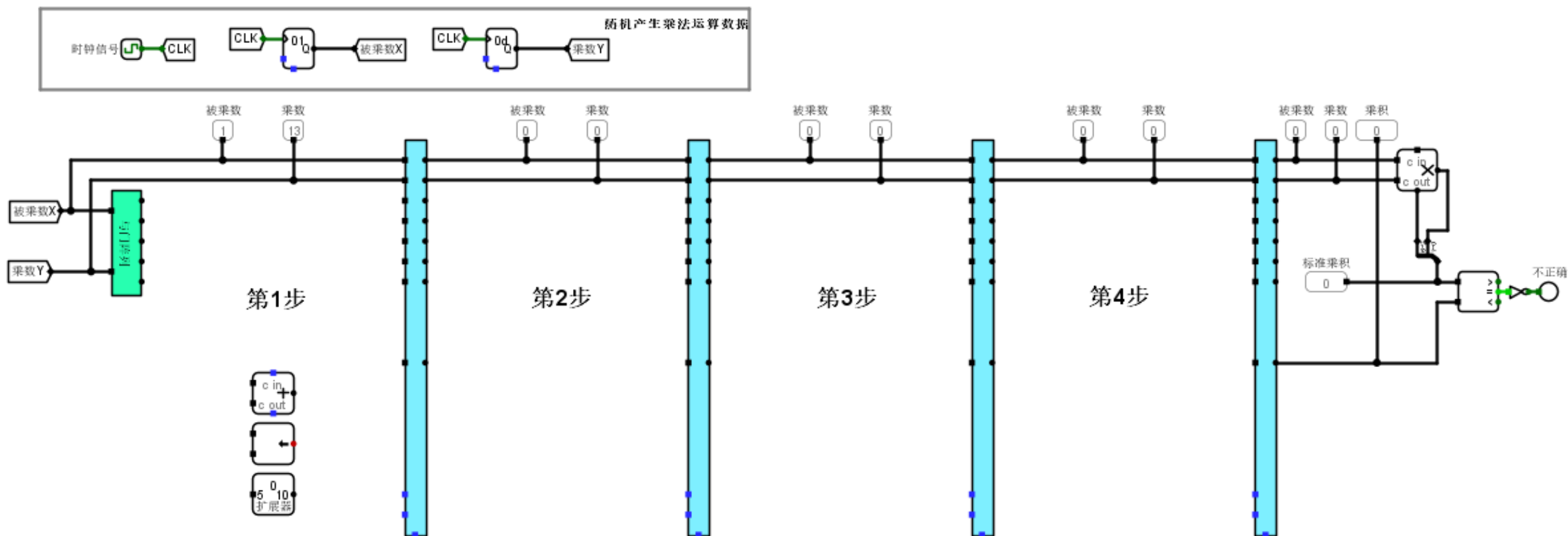


在给定的乘法流水线框架下，利用

- 已经定义好的隧道标签、5位与门乘法阵列、乘法流水线接口
- 添加十位加法器、移位器和位扩展器

完成电路的设计与连接

在流水线框架下，主要完成各阶段组合逻辑的设计实现



乘法流水线接口



乘法流水线接口电路 (已实现、封装)

一组寄存器

- 同步时钟
- 使能端：低电平有效
- 清0端：高电平有效

结合二选一多路选择器实现同步清0

