

实验三报告

231220088 陈翔宇

实验内容

一、4位先行进位部件CLU

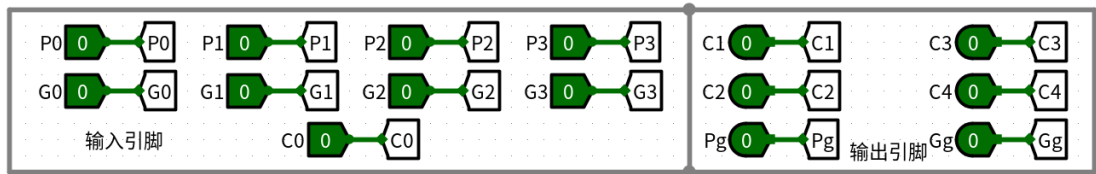
4位先行加法器

根据题目描述可以画出真值表

[illegible]

整体方案设计

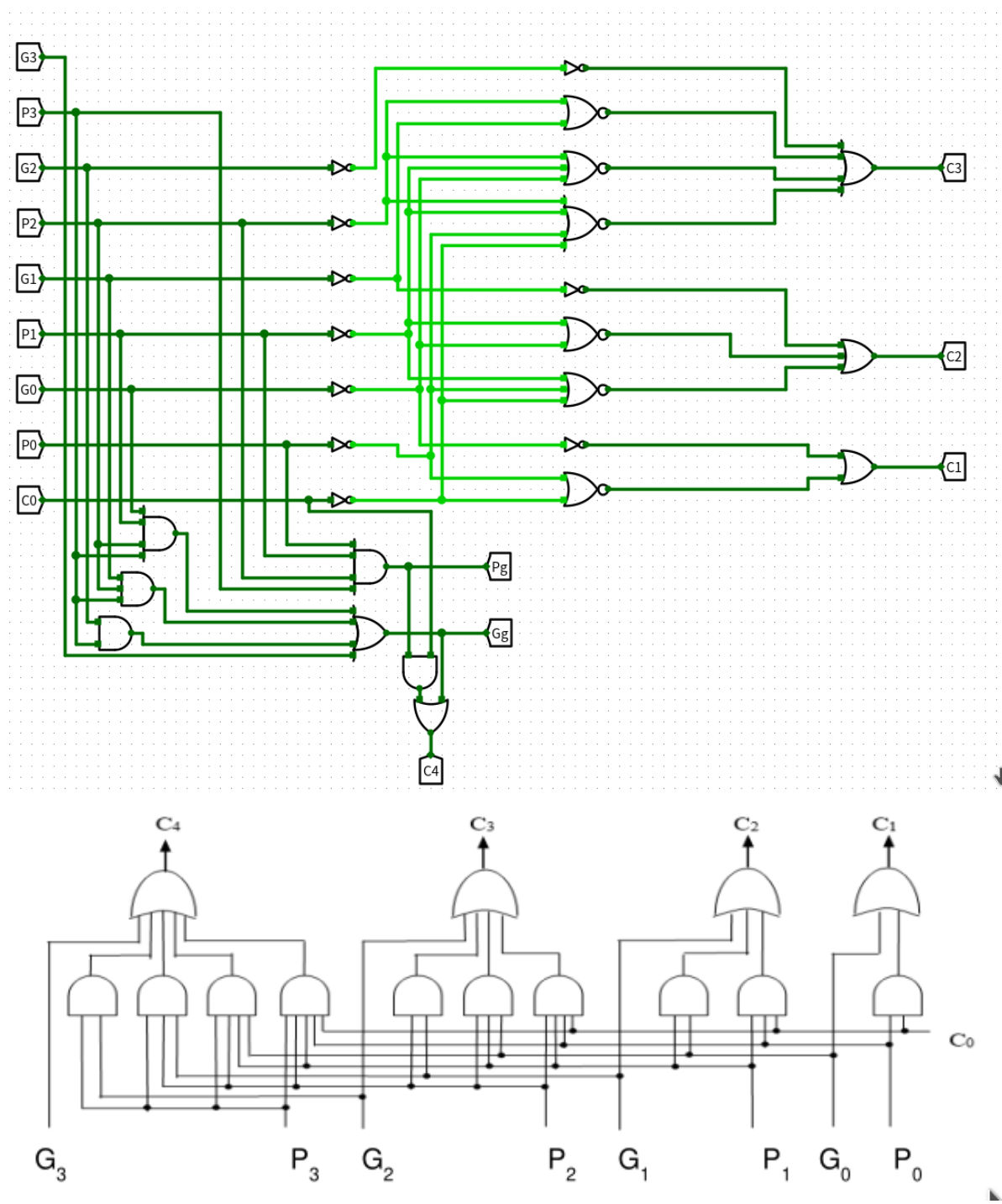
1. 输入输出引脚



四位先行进位逻辑单元CLU，C0为进位输入，C1C2C3C4为进位输出，Pg、Gg为成组进位生成传递函数

- C0: 低位进位
- Pi: 传递函数
- Gi: 生成函数
- Pg: 高位传递函数
- Gg: 高位生成函数

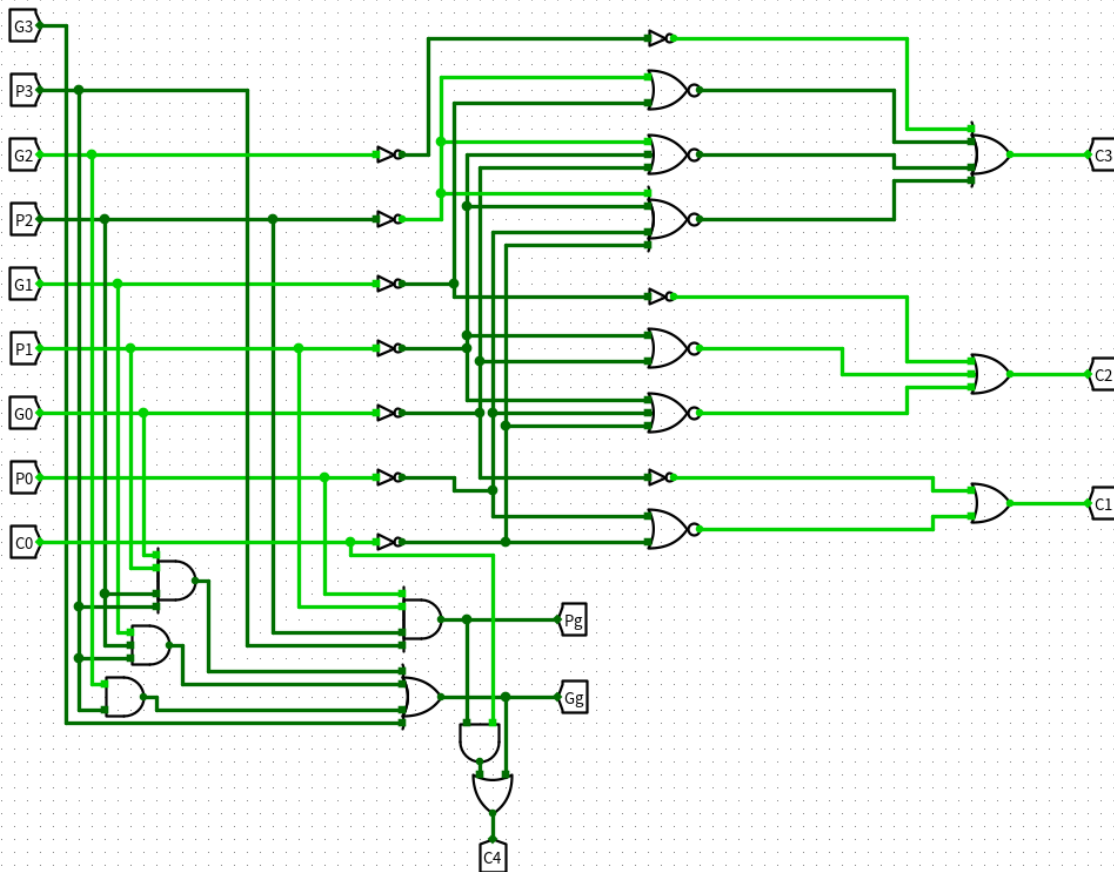
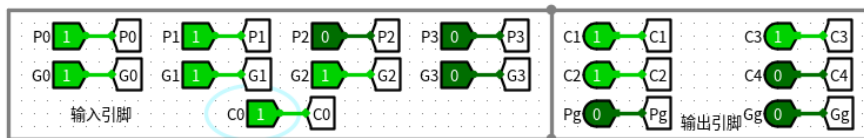
电路图和原理图



仿真测试

1	Cnt	X	Y	Cin	Z	Cout	Gg	Pg
2	00	f	2	0	1	1	1	1
3	01	e	3	0	1	1	1	1
4	02	d	4	0	1	1	1	0
5	03	c	5	0	1	1	1	0
6	04	b	6	0	1	1	1	1
7	05	a	7	0	1	1	1	1
8	06	9	8	0	1	1	1	0
9	07	8	9	0	1	1	1	0
10	08	7	a	0	1	1	1	1
11	09	6	b	0	1	1	1	1
12	0a	5	c	0	1	1	1	0

13	0b	4	d	0	1	1	1	0
14	0c	3	e	0	1	1	1	1
15	0d	2	f	0	1	1	1	1
16	0e	1	f	0	0	1	1	1
17	0f	0	f	0	f	0	0	1
18	10	f	f	1	f	1	1	1
19	11	e	f	1	e	1	1	1
20	12	c	f	1	c	1	1	1
21	13	a	f	1	a	1	1	1
22	14	8	7	1	0	1	0	1
23	15	6	9	1	0	1	0	1
24	16	4	b	1	0	1	0	1
25	17	2	d	1	0	1	0	1
26	18	3	c	1	0	1	0	1
27	19	5	c	1	2	1	1	0
28	1a	7	c	1	4	1	1	1
29	1b	9	c	1	6	1	1	0
30	1c	b	c	1	8	1	1	1
31	1d	d	c	1	a	1	1	0
32	1e	f	c	1	c	1	1	1
33	1f	0	c	1	d	0	0	0



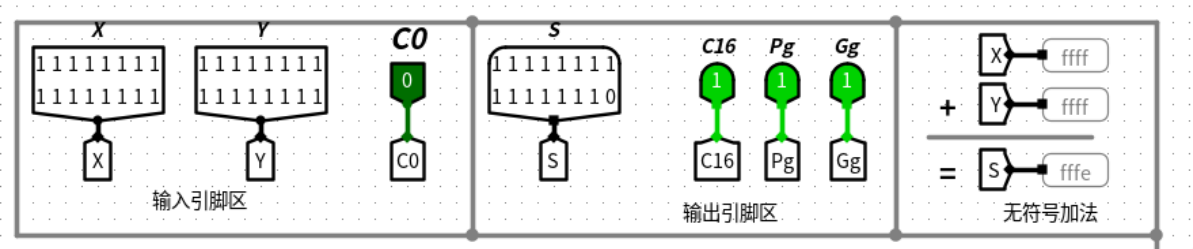
真值表: [真值表](#)

错误现象及分析

在完成实验过程中没有遇到任何错误。

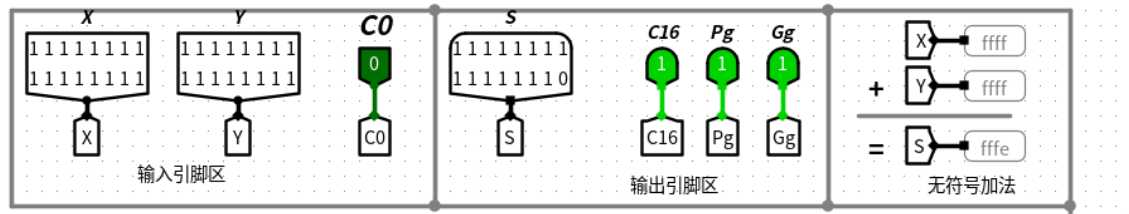
二、16 位两级先行进位加法器实验

基本原理



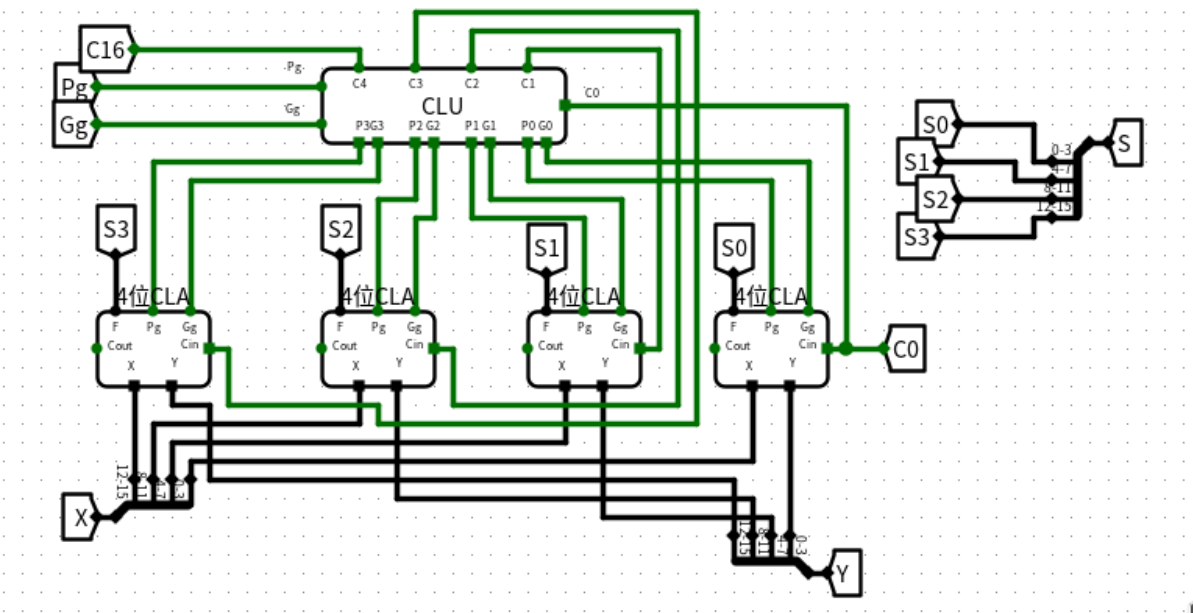
整体方案设计

1. 输入输出引脚

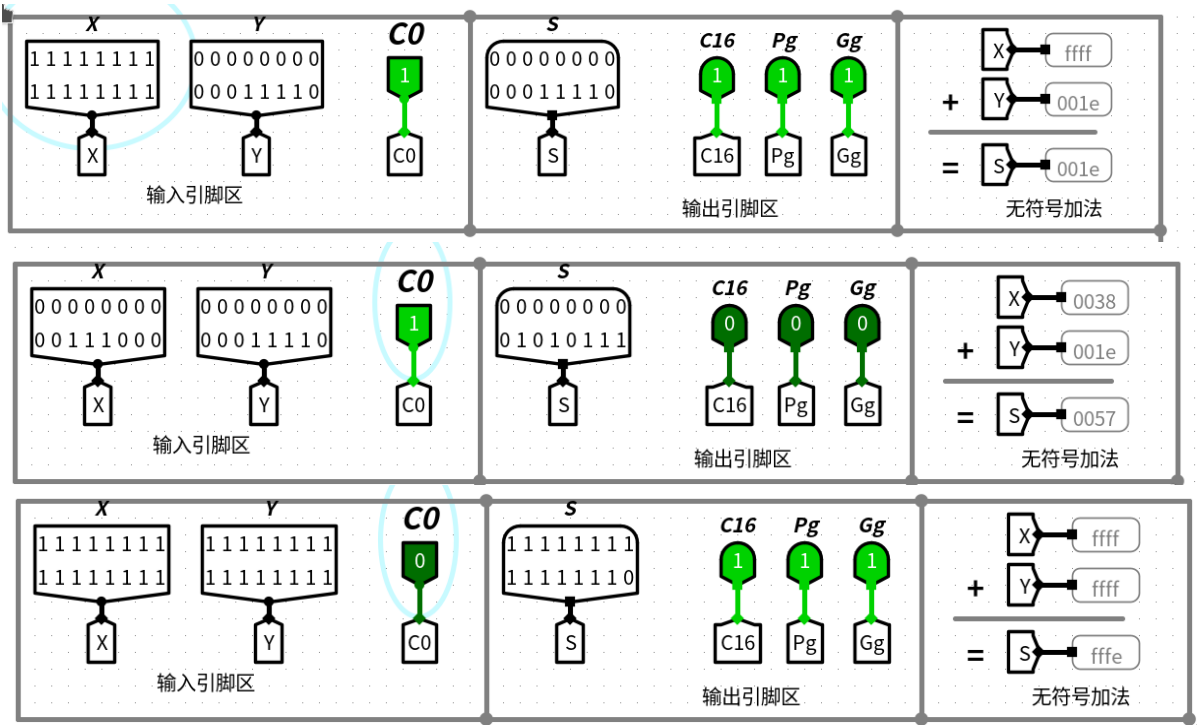


- X: 操作数X
- Y: 操作数Y
- S: 和
- C0: 低位进位
- C16: 溢出进位
- Pg: 传递函数
- Gg: 生成函数

电路图



仿真测试



错误现象及分析

在完成实验过程中没有遇到任何错误。

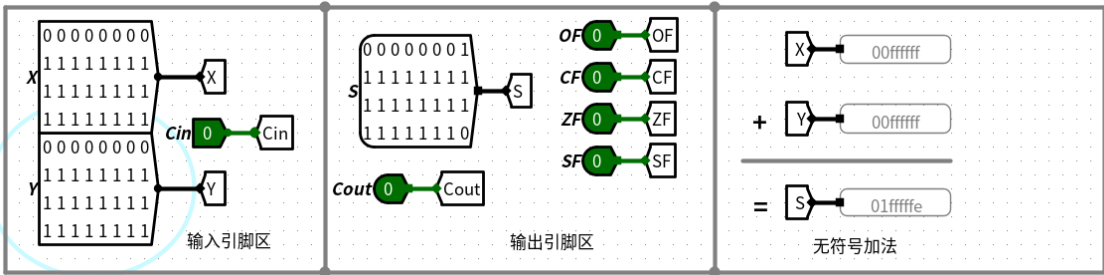
三、32 位快速加法器构建实验

基本原理

- 1 通过先行进位器获得每一位的进位 C_i ，此时可通过 $S_i = X_i \oplus Y_i \oplus C_i$ 直接获取和的第 i 位。
- 2 由于 C_i 的计算过程除了 P_i 、 G_i 外只依赖低位进位 C_0 ，所以可以做到各个位在一个时钟周期内获得结果。

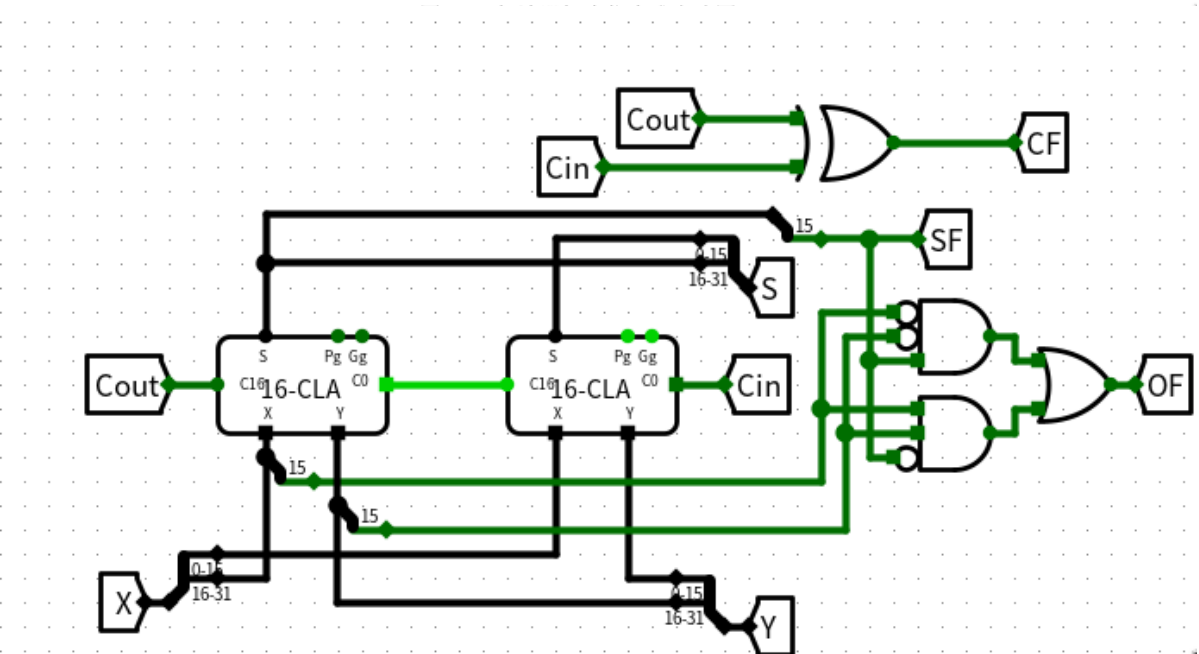
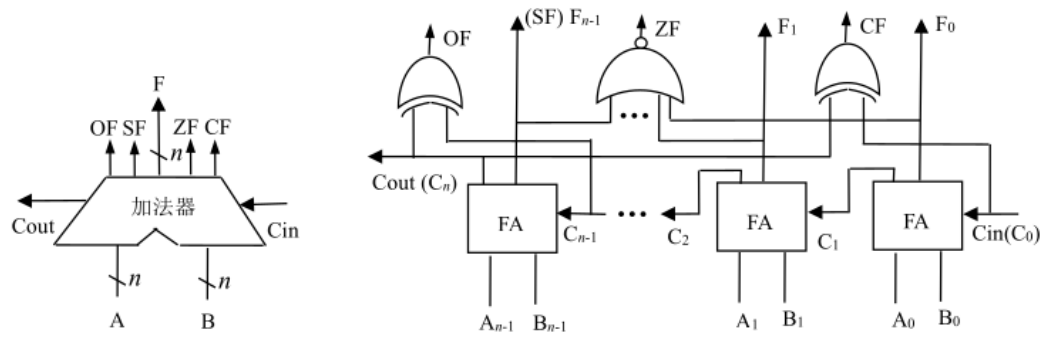
整体方案设计

1. 输入输出引脚

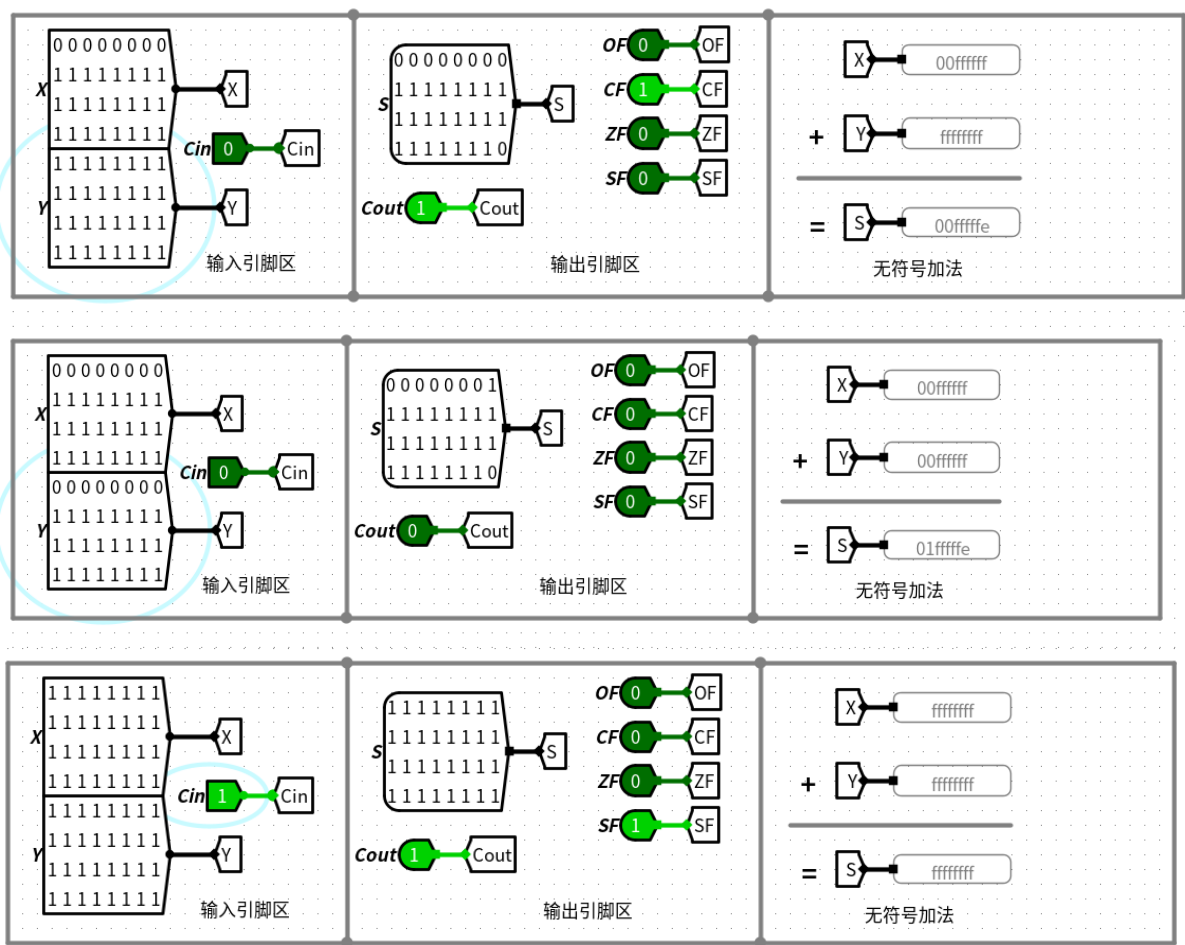


- Y、X：运算数
- Cin：低位进位
- Cout：和
- OF：溢出位
- CF：进位标志
- ZF：0标志位
- SF：符号位

电路图



仿真测试



错误现象及分析

在完成实验过程中没有遇到任何错误。

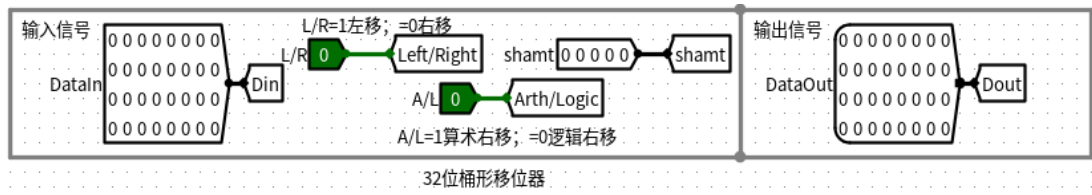
四、32 位桶形移位器设计

基本原理

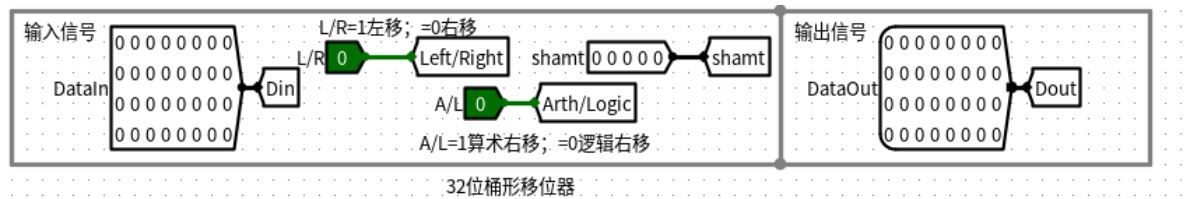
- 1 在移位操作之前，先将被移位数的最高位（对于左移是左边第一个位，对于右移是右边第一个位）暂存起来。
- 2 将被移位数的其余位按照指定的方向（左移或右移）进行移动指定的位数。
- 3 将暂存的最高位移到移位后的最高位位置，而新产生的最低位则用0或特定的值（如算术移位时用符号位填充）填充。
- 4 将暂存的最高位和移位后的数合并，形成最终的移位结果。

整体方案设计

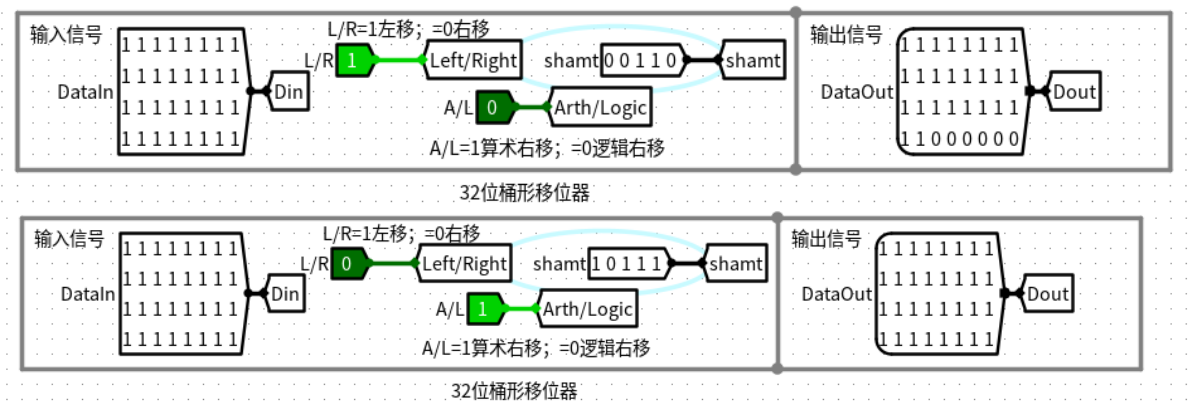
1. 输入输出引脚



电路图



仿真测试



五、ALU 设计实验

基本原理

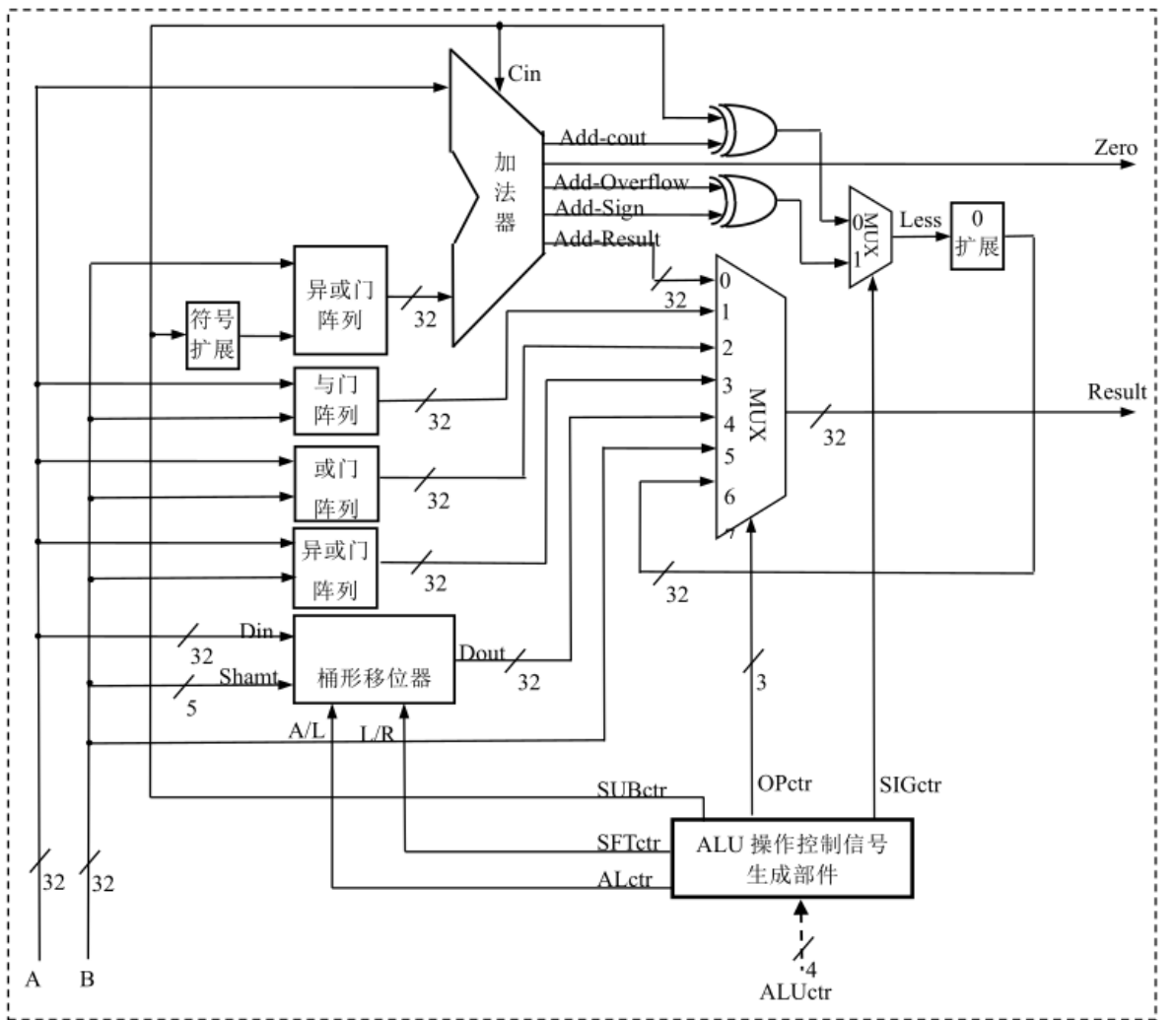


表 4.1 ALUctr 的四位编码及其对应的控制信号

ALUctr <3:0>	指令操作类型	SUBctr	SIGctr	ALctr	SFTctr	OPctr <2:0>	OPctr 的含义
0 0 0 0	auipc,addi,add,jal,jalr, lb,lh,lw,lbu,lhu,sb,sh,s w: 加法	0	×	×	×	000	选择加法器的结果输出
0 0 0 1	sll,slli: 逻辑左移	×	×	0	1	100	选择移位器结果输出
0 0 1 0	slt,slti,beq,bne,blt,bge : 带符号数小于比较	1	1	×	×	110	选择小于置位结果输出
0 0 1 1	sltu,sltui,bltu,bgeu: 无符号数小于比较	1	0	×	×	110	选择小于置位结果输出
0 1 0 0	xor,xori: 异或	×	×			011	选择“按位异或”结果输出



0 1 0 1	srl,srli: 逻辑右移	×	×	0	0	100	选择移位器结果输出
0 1 1 0	or,ori: 或运算	×	×	×	×	010	选择“按位或”结果输出
0 1 1 1	and,andi: 与运算	×	×	×	×	001	选择“按位与”结果输出
1 0 0 0	sub: 减法	1	×	×	×	000	选择加法器的结果输出
1 1 0 1	sra,srai: 算术右移	×	×	1	0	100	选择移位器结果输出
其余	(未用)						
1 1 1 1	lui: 取操作数 B	×	×	×	×	101	选择操作数 B 直接输出

$$\text{SUBctr} = \sum m(2,3,8)$$

$$\text{SIGctr} = m_2$$

$$\text{ALctr} = m_{13}$$

$$\text{SFTctr} = m_1$$

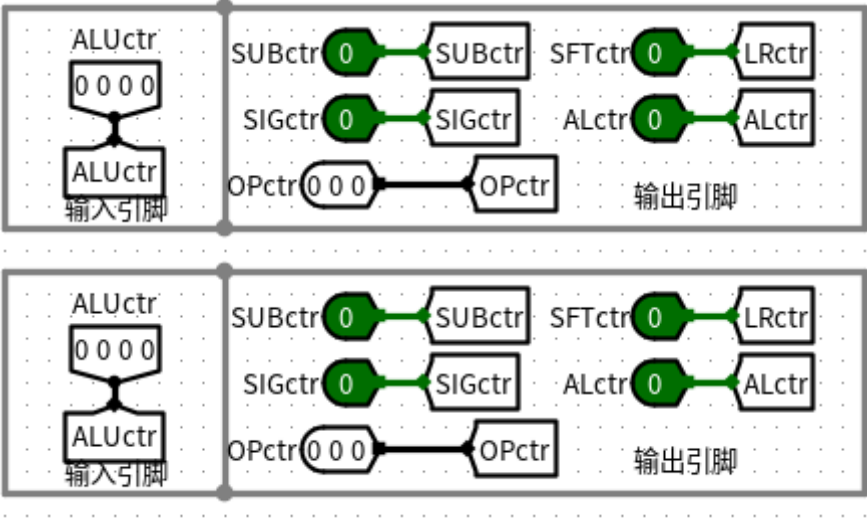
$$\text{Opctr}[2] = \sum m(1,2,3,5,13,15)$$

$$\text{Opctr}[1] = \sum m(2,3,4,6)$$

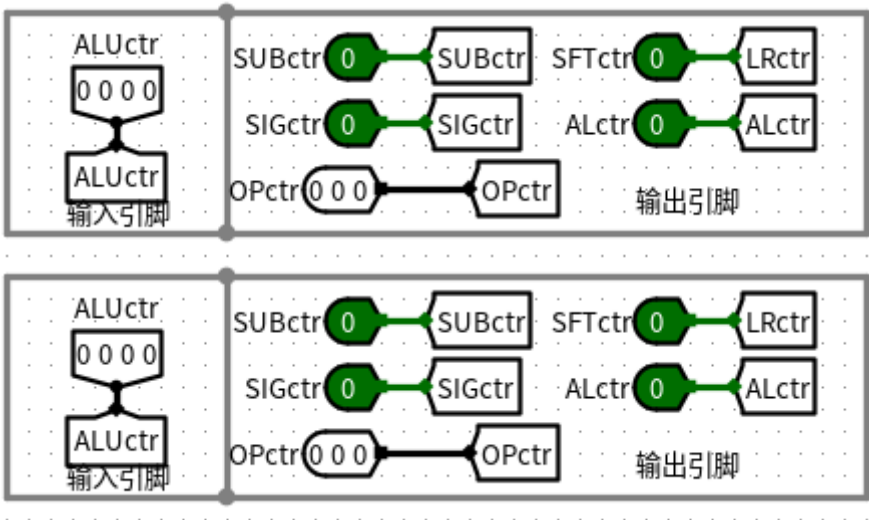
$$\text{Opctr}[0] = \sum m(4,7,15)$$

整体方案设计

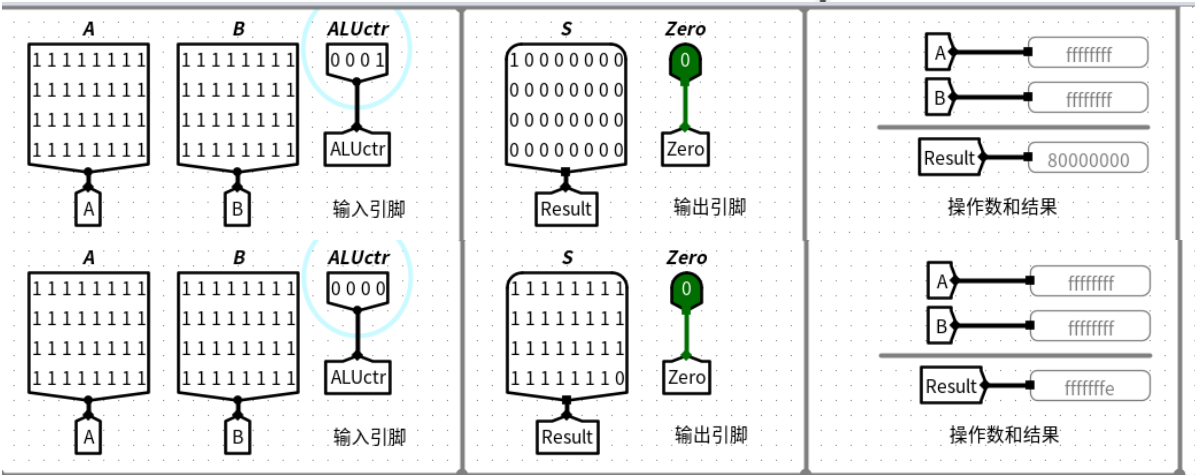
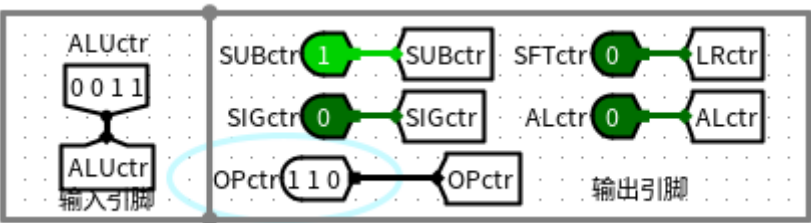
1. 输入输出引脚



电路图

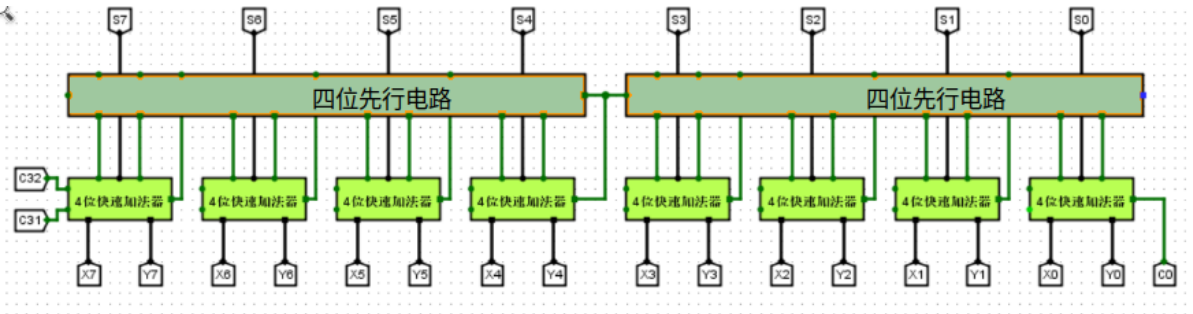


仿真测试



思考题

1. 将实验 3 中的快速乘法器设计电路扩展到 32 位无符号数相乘，并探讨如何将该乘法器融合到实验中的 ALU 电路来实现乘法运算



2.在 RV32I 中新增一条指令，然后在 ALU 中新增一个新运算，并通过测试数据进行验证。

3. 如何实现 32 位无符号数除法器？

- 1

从被除数的最高位开始，逐步向左移动，每次考虑被除数的当前部分（从高位到低位）。
- 2

将当前部分与除数进行比较，如果当前部分大于或等于除数，那么在商中记录一个1，并从当前部分中减去除数的相应倍数。
- 3

如果当前部分小于除数，则在商中记录一个0，并将当前部分保留到下一次迭代中。
- 4

重复以上步骤，直到被除数的所有位都被考虑过。