

实验 4: ALU 设计

一、实验目的

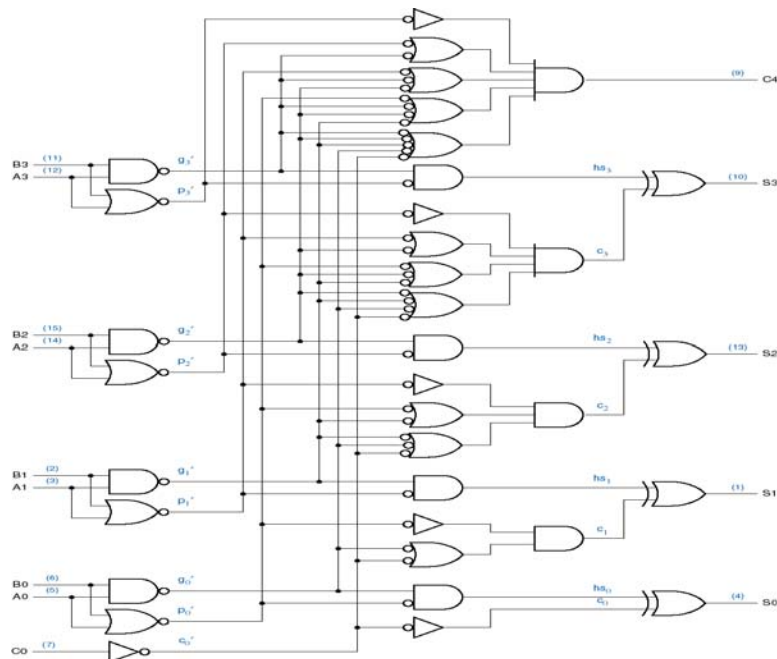
- 1、掌握快速加法器 CLA 和先行进位逻辑 CLU 的设计方法。
- 2、掌握 32 位先行进位加法器及相关标志位的实现方法。
- 3、掌握 ALU 的设计方法, 根据指令要求实现 6 种操作的 ALU 器件。

二、实验环境

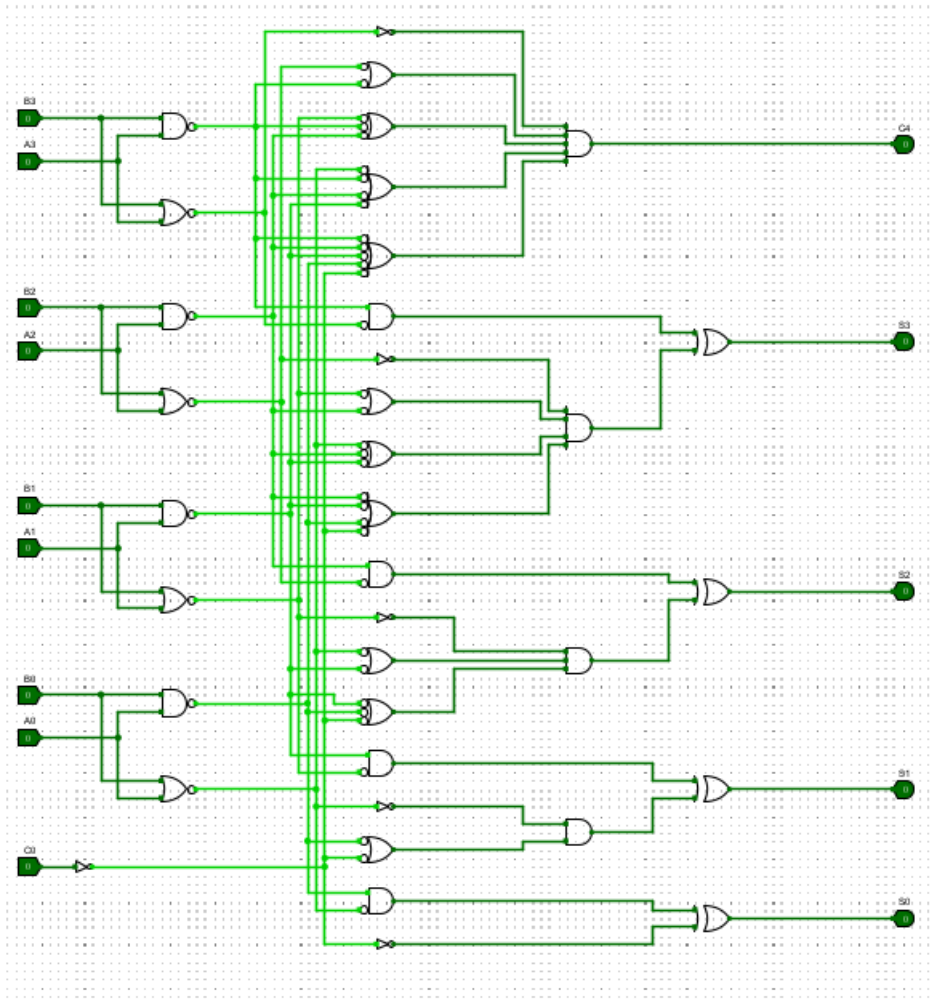
Logisim-ITA V2.16.1.0。

三、实验内容

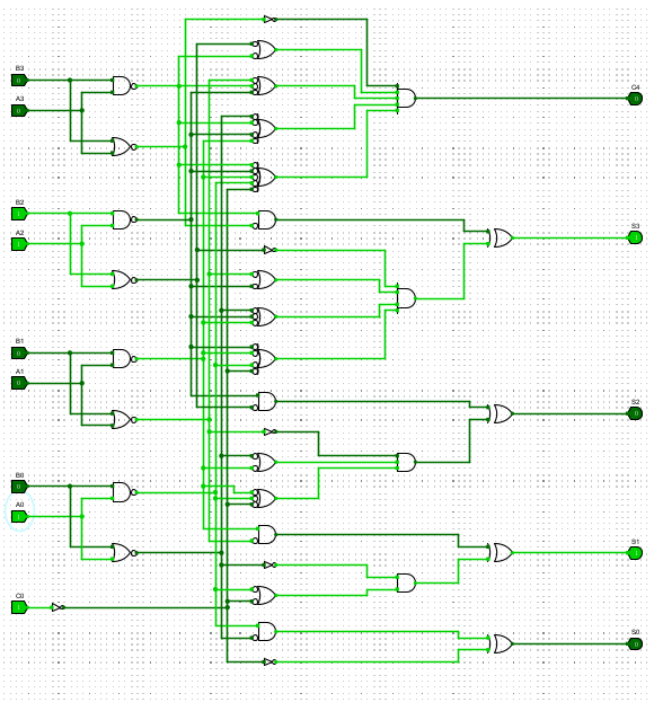
- 1、根据下图给出的电路原理图(参照其他原理图亦可), 实现 并验证 4 位快速加法器 CLA。



- 1) 电路原理图如图
- 2) 实验器材
2 输入与非门*4、2 输入或非门*4、非门*6、或门、2345 输入与门各一个、异或门*4
- 3) 实验步骤
连接电路



4) 仿真检测



2、 根据给出的逻辑表达式，选择合适的逻辑门，实现并验证 4 位先行进位逻辑单元CUU。

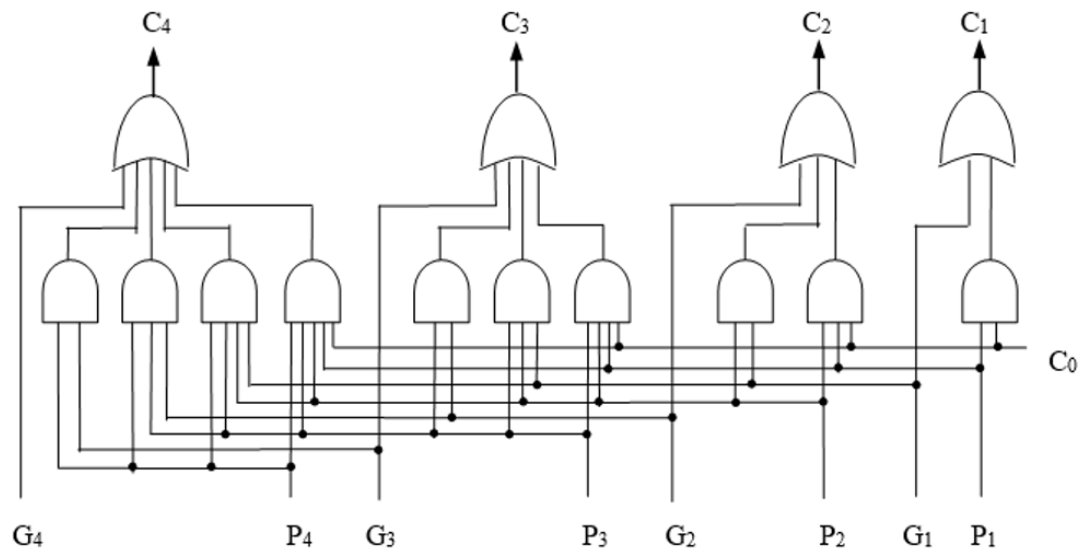
$$C_1 = G_0 + P_0 C_0$$

$$C_2 = G_1 + P_1 C_1 = G_1 + P_1 G_0 + P_1 P_0 C_0$$

$$C_3 = G_2 + P_2 C_2 = G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_0$$

$$C_4 = G_3 + P_3 C_3 = G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0 + P_3 P_2 P_1 P_0 C_0$$

1) 电路原理图

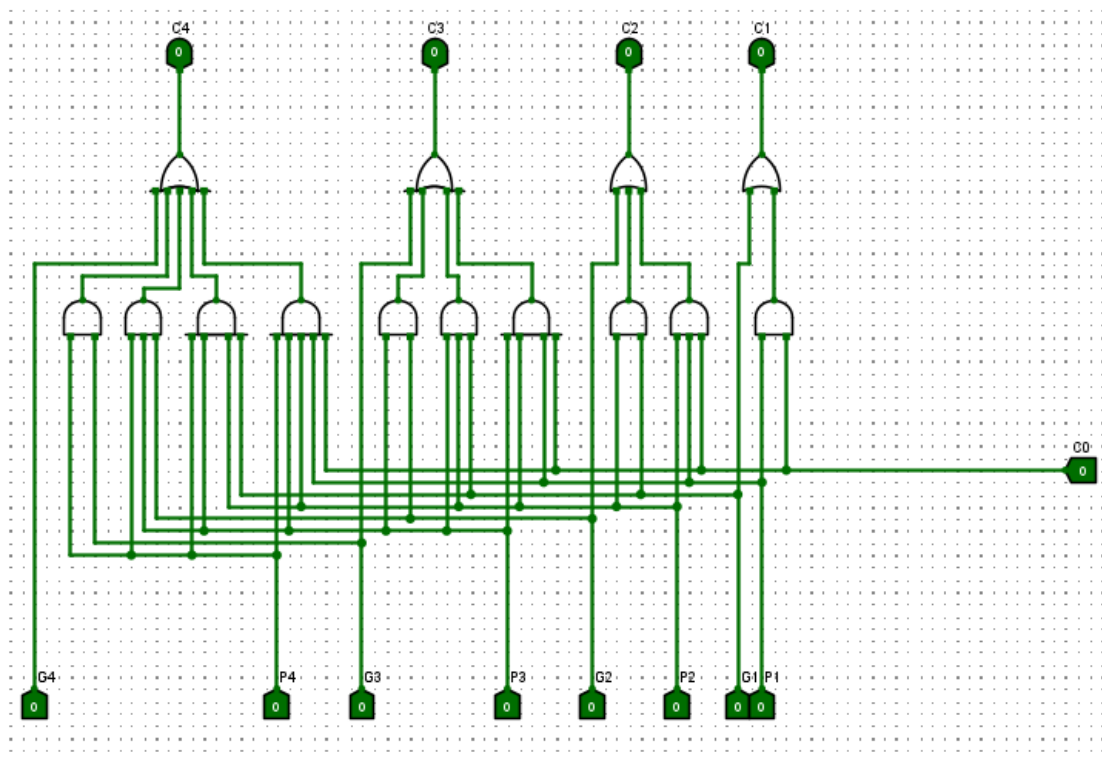


2) 实验器材

2 输入与门*4、3 输入与门*3、4 输入与门*2、5 输入与门*1、2345 输入或门各一个

3) 实验步骤

放置器材并连线

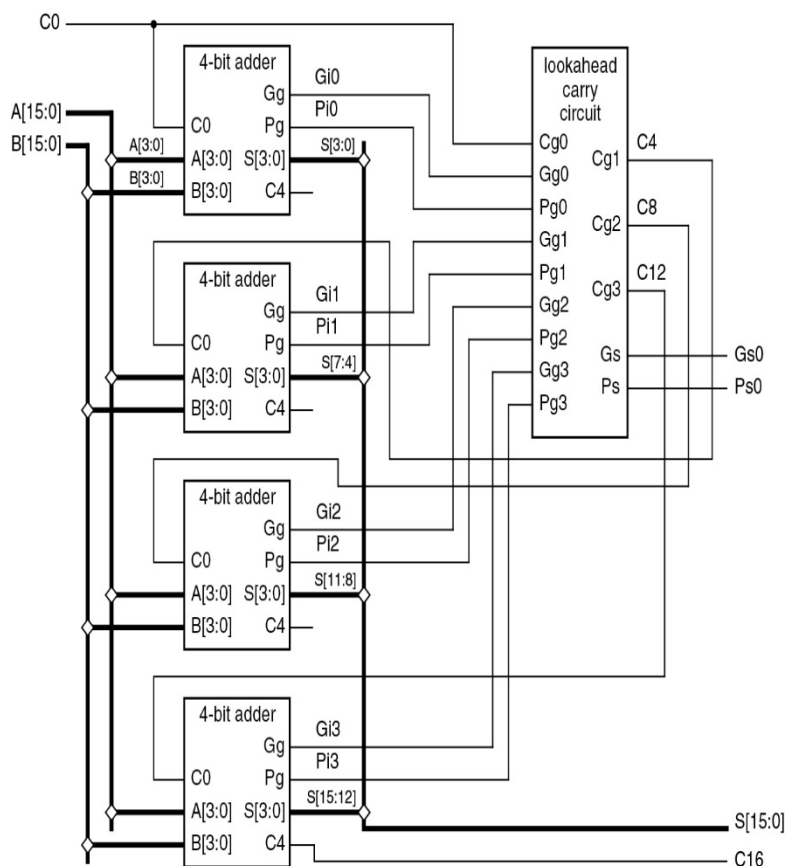


3、 根据给出的逻辑表达式和电路原理图，在 4 位快速加法器中增加支持组件并联的 G_g 、

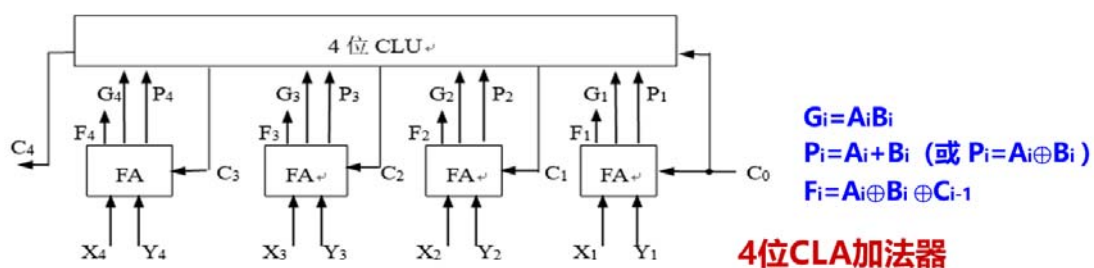
Pg 输出端，加上 4 位先行进位逻辑部件，设计并实现 16 位先行进位加法器。

$$Gg = g_3 + p_3 \cdot g_2 + p_3 \cdot p_2 \cdot g_1 + p_3 \cdot p_2 \cdot p_1 \cdot g_0$$

$$Pg = p_3 \cdot p_2 \cdot p_1 \cdot p_0$$



1) 电路原理图如图以及 4 位 CLA

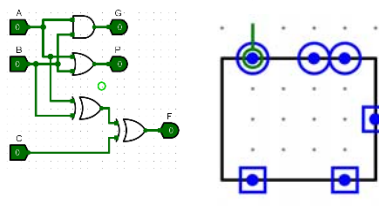


2) 实验器材

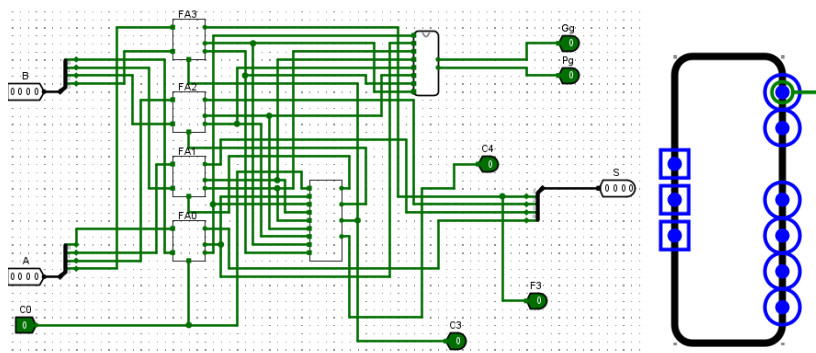
4-bit adder * 4、4-bit CLU、分线器 * 4

3) 实验步骤

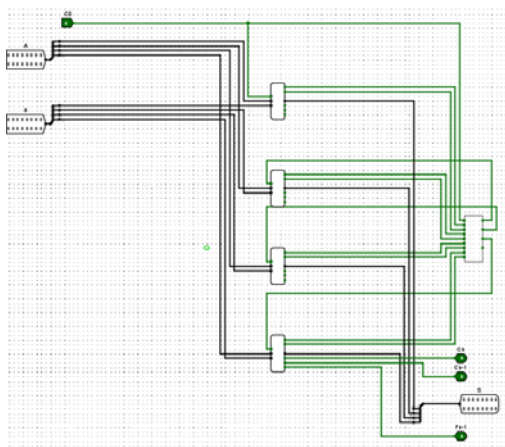
a) 全加器连接并封装



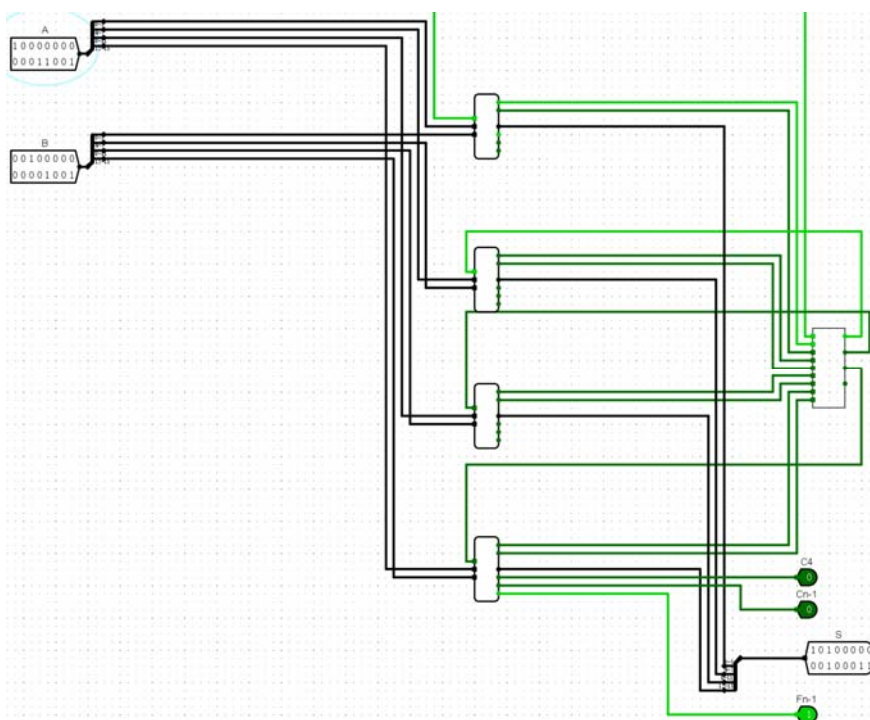
b) 4-bit adder (4-bit CLU * 1、FullAdder * 4)



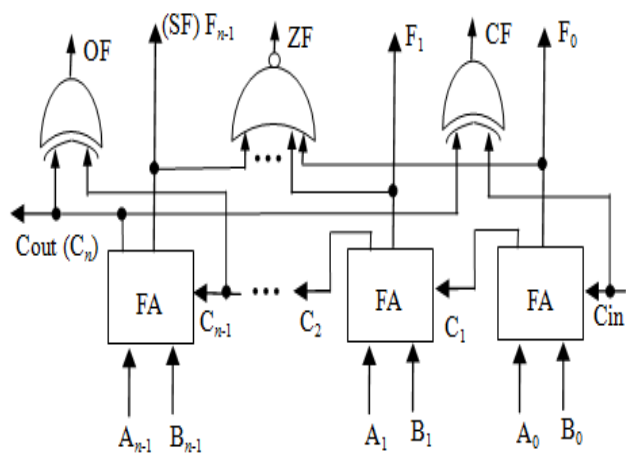
c) 16-bit-CLA



4) 仿真检测



4、根据给出的标志位生成电路原理图，利用两片 16 位先行进位加法器实现 32 位快速加法器，及 CF、SF、OF、ZF 等标志位。



带标志加法器的逻辑电路

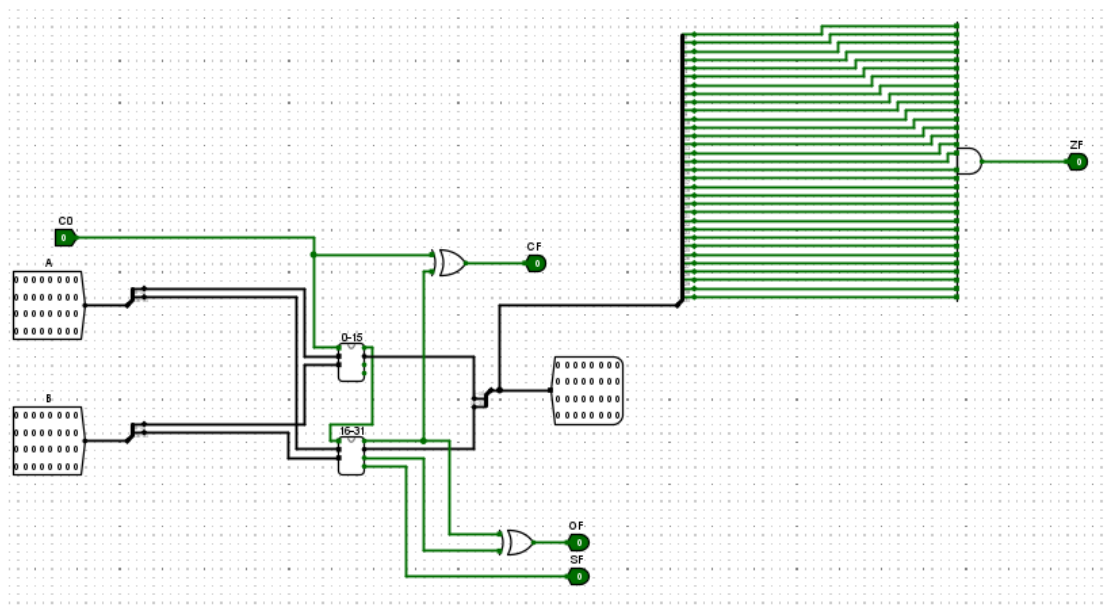
1) 电路原理图如图

2) 实验器材

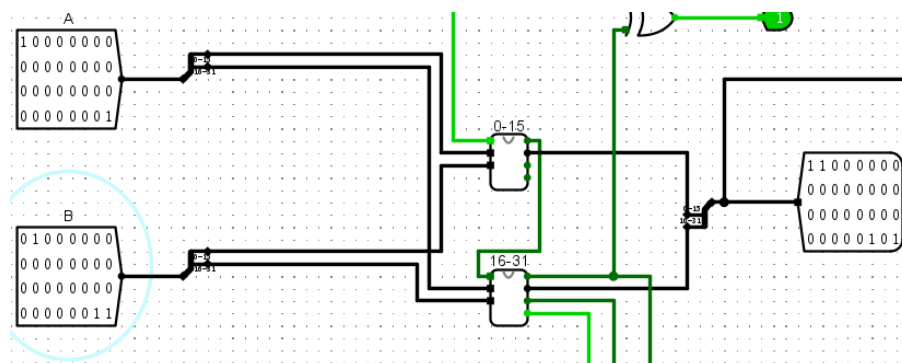
16-bit-CLA * 2、2 输入异或门 * 2、分线器 * 1、32 位与门 * 1

3) 实验步骤

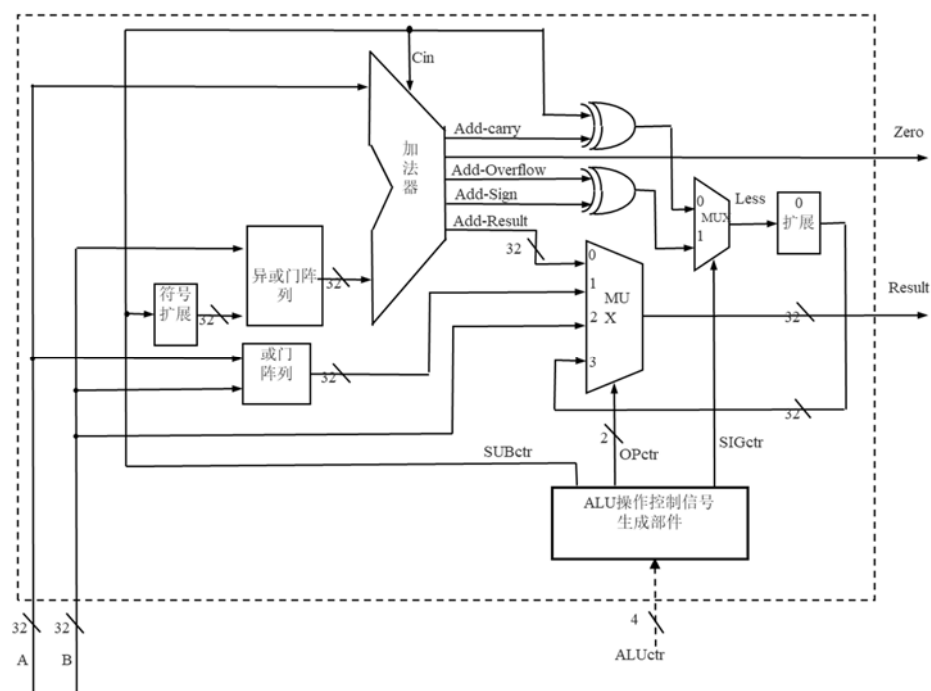
连线



4) 电路仿真检测



- 5、根据给出的电路原理图和 ALU 引脚定义要求，设计并验证支持 9 条指令 6 种操作的 32 位算术逻辑运算单元 ALU，6 种运算包括：add、or、slt、sltu、srcB, 判 0(sub)。



ALU 设计原理图

对应的 ALU 操作控制信号取值

指令	功能	运算类型	SUBctr	SIGctr	OPctr<1:0>
add rd, rs1, rs2	$R[rd] \leftarrow R[rs1] + R[rs2]$	加	0	×	00
slt rd, rs1, rs2	if ($R[rs1] < R[rs2]$) $R[rd] \leftarrow 1$ else $R[rd] \leftarrow 0$	减，带符号整数比较大小	1	1	11
sltu rd, rs1, rs2	if ($R[rs1] < R[rs2]$) $R[rd] \leftarrow 1$ else $R[rd] \leftarrow 0$	减，无符号数比较大小	1	0	11
ori rt, rs1, imm12	$R[rt] \leftarrow R[rs1] \mid \text{SEXT}(imm12)$	按位或	×	×	01
lui rd, imm20	$R[rt] \leftarrow imm20 \mid 000H$	选择操作数 B	×	×	10
lw rd, rs1, imm12	$Addr \leftarrow R[rs1] + \text{SEXT}(imm12)$ $R[rd] \leftarrow M[Addr]$	加	0	×	00
sw rs1, rs2, imm12	$Addr \leftarrow R[rs1] + \text{SEXT}(imm12)$ $M[Addr] \leftarrow R[rs2]$	加	0	×	00
beq rs1, rs2, imm12	$Cond \leftarrow R[rs1] - R[rs2]$	减（判 0）	1	×	×
	if ($Cond \text{ eq } 0$) $PC \leftarrow PC + (\text{SEXT}(imm12) \times 2)$	加	0	×	00
jal rd, imm20	$R[rd] \leftarrow PC + 4$ $PC \leftarrow PC + (\text{SEXT}(imm20) \times 2)$	加	0	×	00

ALUctr 的一种四位编码方案

ALUctr<3:0>	操作类型	SUBctr	SIGctr	OPctr<1:0>	OPctr 的含义
0 0 0 0	add	0	×	0 0	选择加法器的结果输出
0 0 0 1	(未用)				
0 0 1 0	slt	1	1	1 1	选择小于置位结果输出
0 0 1 1	sltu	1	0	1 1	选择小于置位结果输出
0 1 0 0	(未用)				
0 1 0 1	(未用)				
0 1 1 0	or	×	×	0 1	选择“按位或”结果输出
0 1 1 1	(未用)				
1 0 0 0	sub	1	×	0 0	选择加法器的结果输出
其余	(未用)				
1 1 1 1	srcB	×	×	1 0	选择操作数 B 直接输出

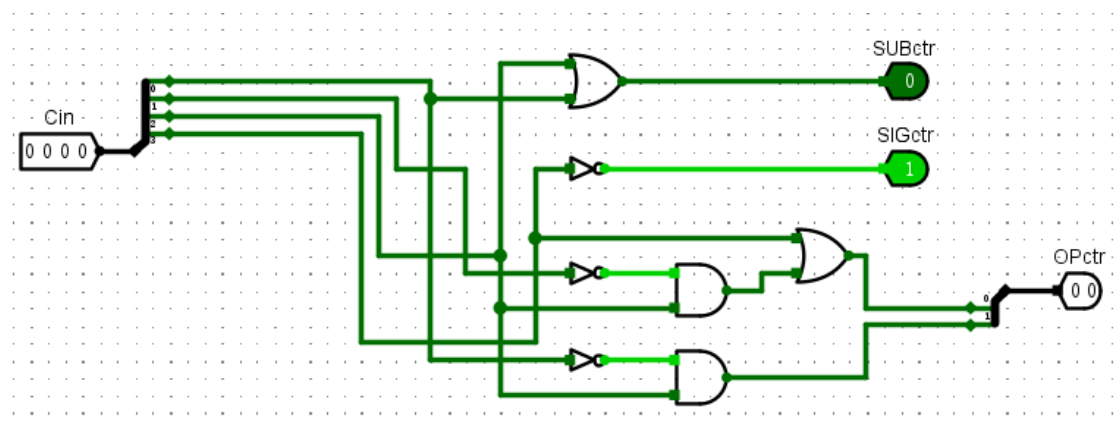
1) 电路原理图如图

2) 实验器材

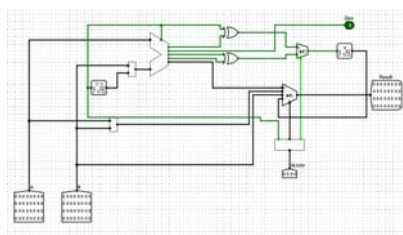
32 位带标志加法器 * 1、或门阵列 * 1、异或门阵列 * 1、位扩展器 * 2、2 输入异或门 * 2、多路选择器 * 2、ALUctr * 1

3) 实验步骤

a) ALUctr



b) ALU 连接



4) 遇到的问题

Aluctr 设计有问题，并且验收时指令对应的数值没有记住