## 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。

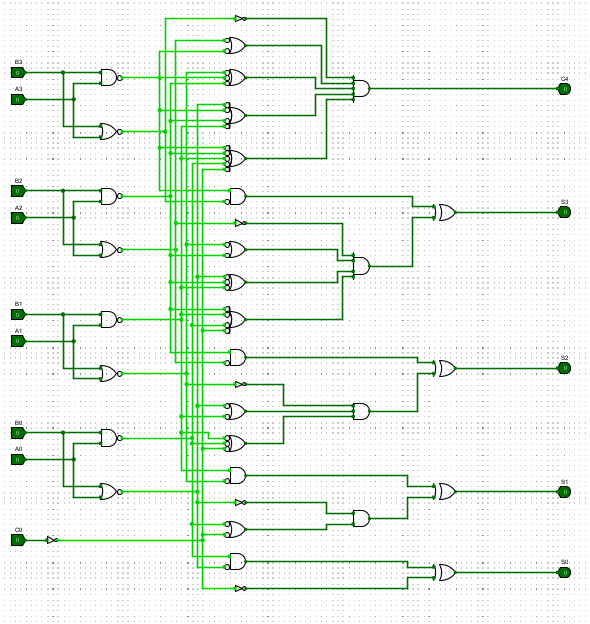


1. 电路原理图如图
2. 实验器材

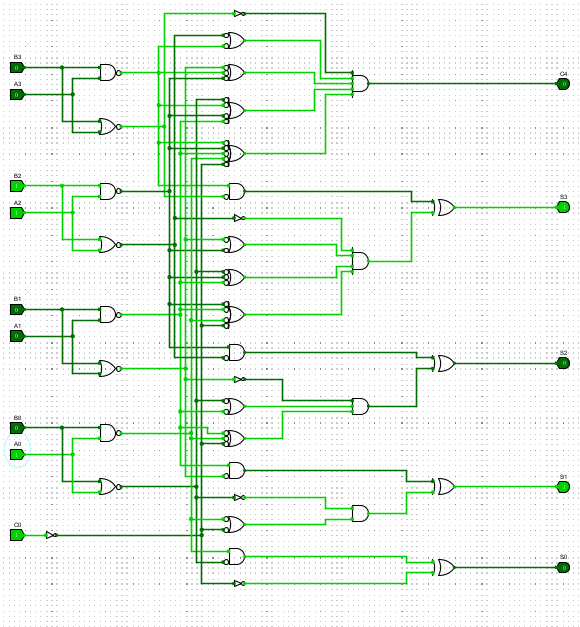
2输入与非门\*4、2输入或非门\*4、非门\*6、或门、2345输入与门各一个、异或门\*4

1. 实验步骤

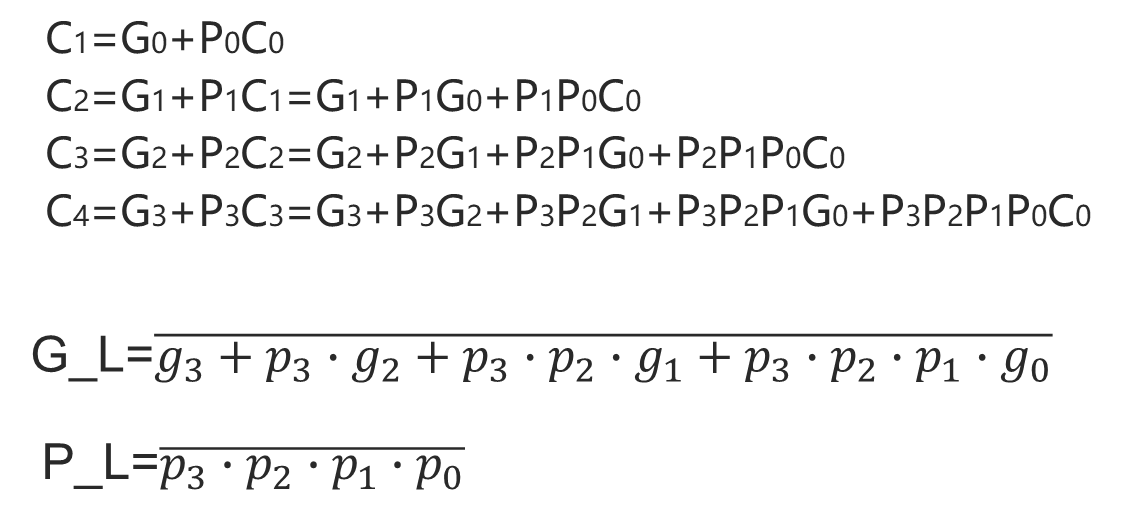
连接电路



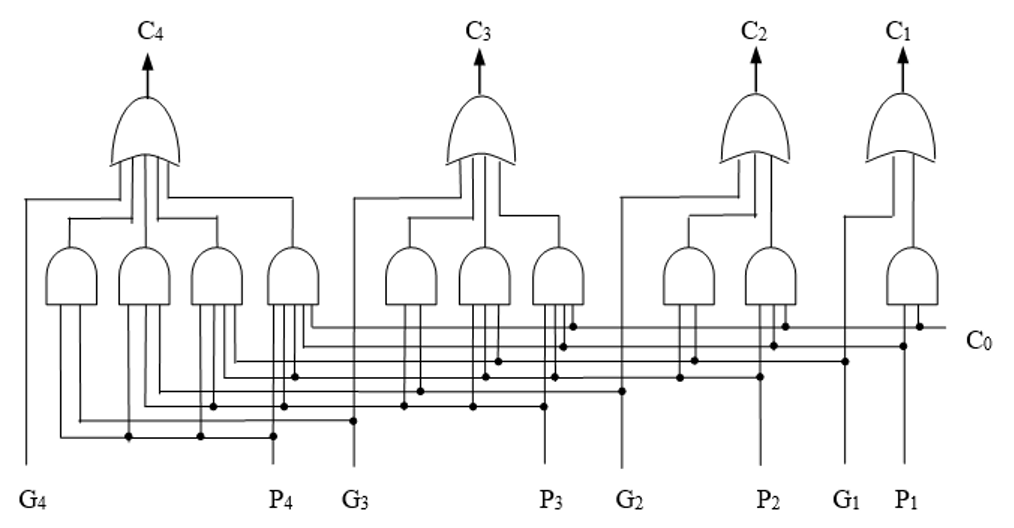
1. 仿真检测



1. 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 CLU。



1. 电路原理图

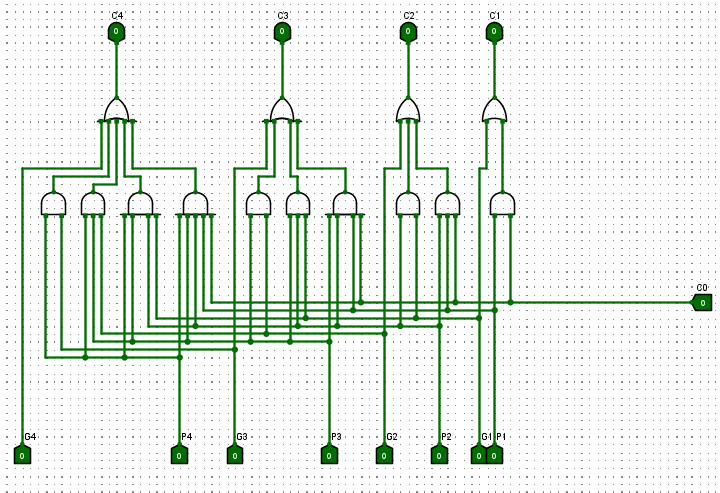


1. 实验器材

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

1. 实验步骤

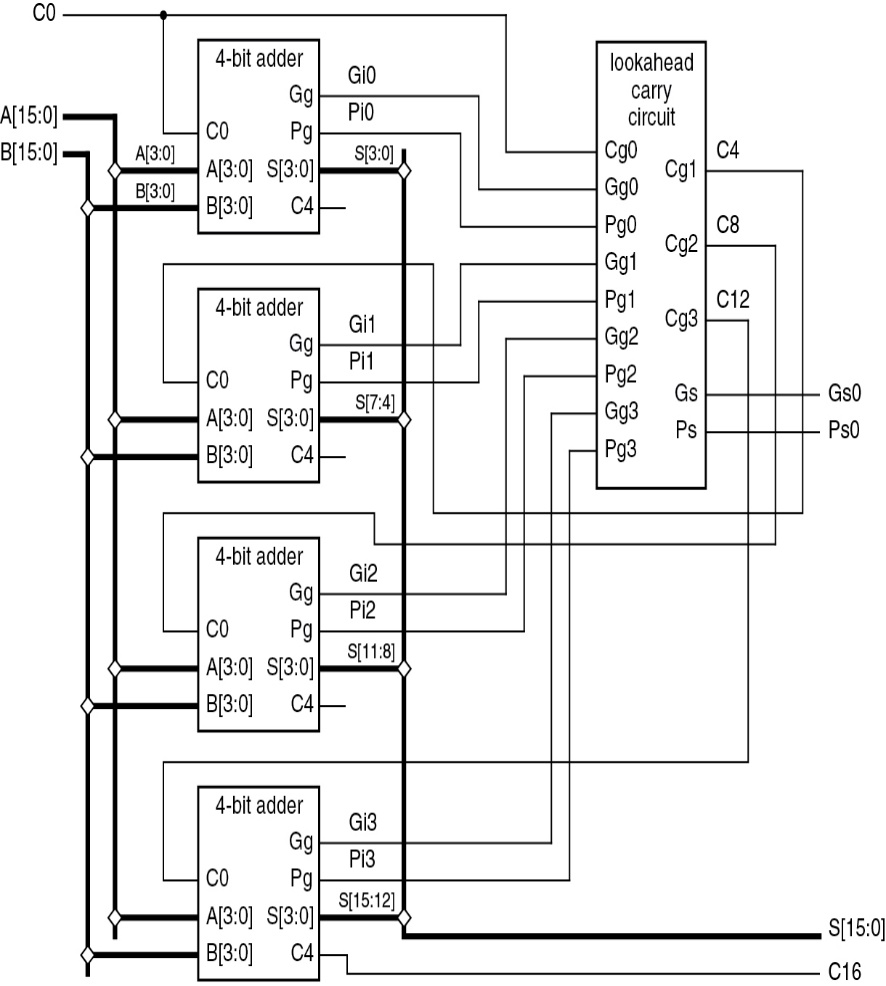
放置器材并连线



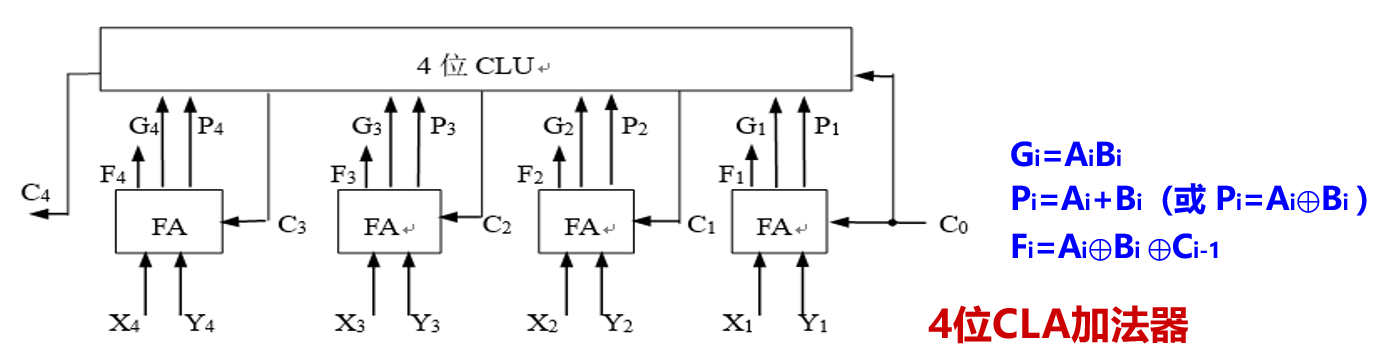
1. 根据给出的逻辑表达式和电路原理图，在4位快速加法器中增加支持组件并联的Gg、Pg输出端，加上4位先行进位逻辑部件，设计并实现16位先行进位加法器。

Gg=

Pg=



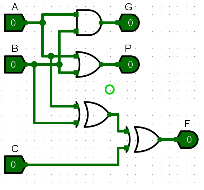
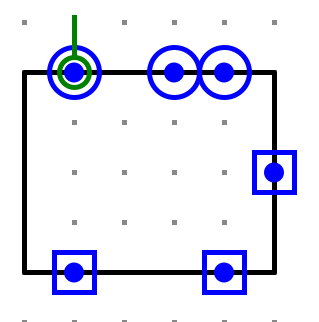
1. 电路原理图如图以及4位CLA



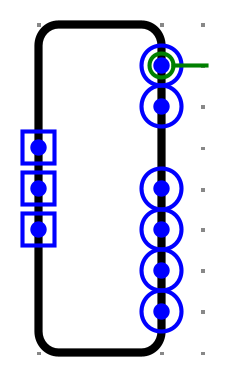
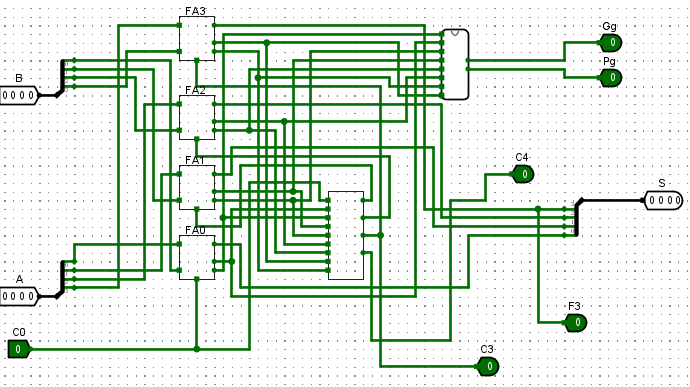
1. 实验器材

4-bit adder \* 4、4-bit CLU、分线器 \* 4

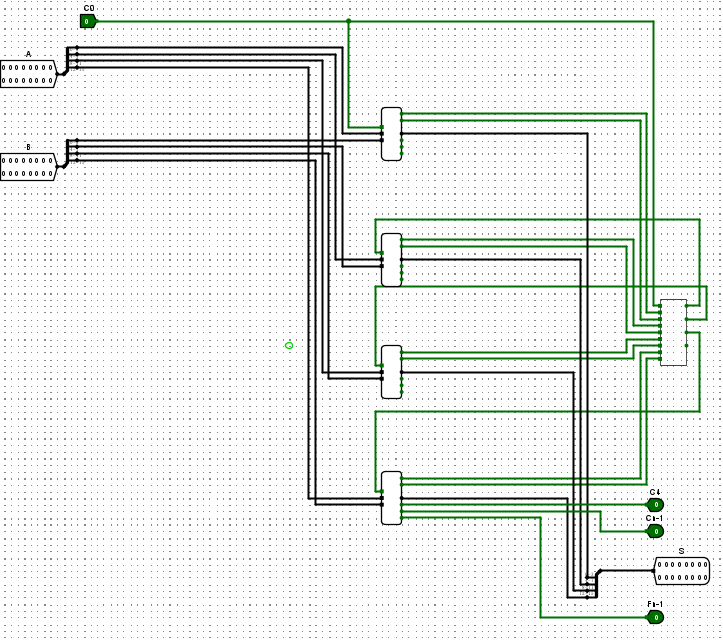
1. 实验步骤
2. 全加器连接并封装



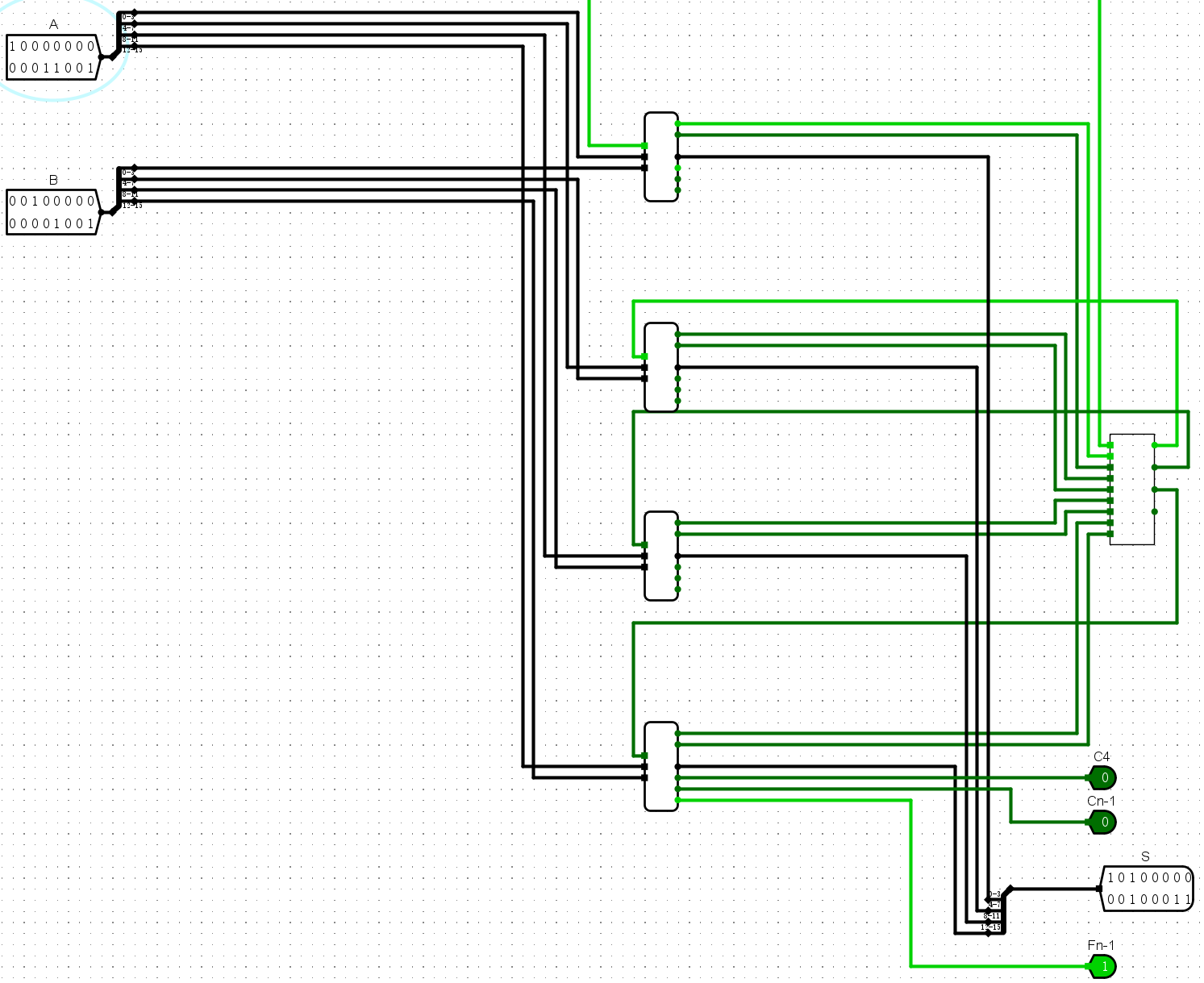
1. 4-bit adder（4-bit CLU \* 1、FullAdder \* 4）



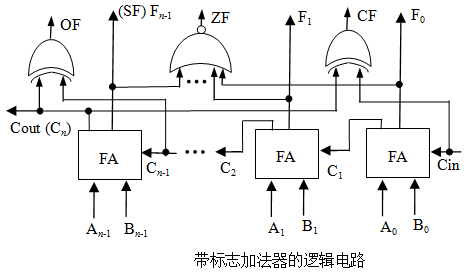
1. 16-bit-CLA



1. 仿真检测



1. 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。

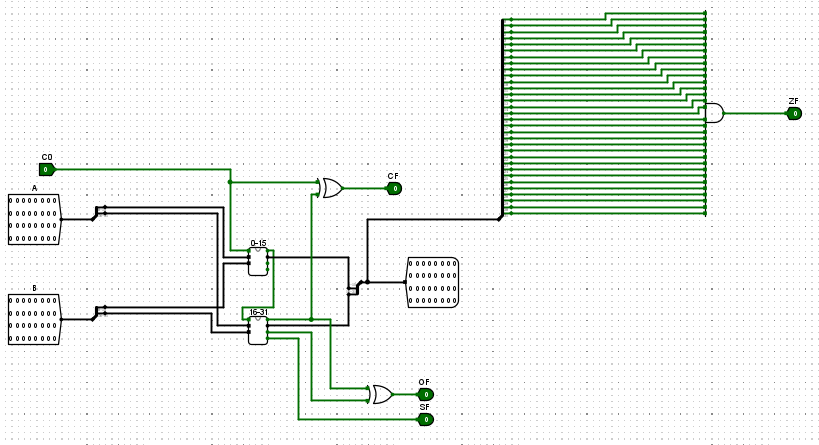


1. 电路原理图如图
2. 实验器材

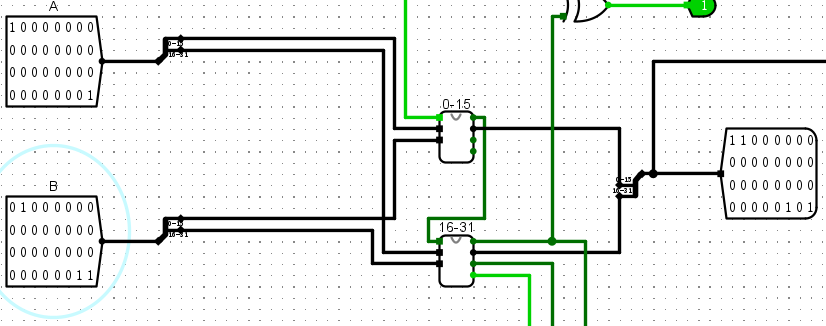
16-bit-CLA \* 2、2输入异或门 \* 2、分线器 \* 1、32位与门 \* 1

1. 实验步骤

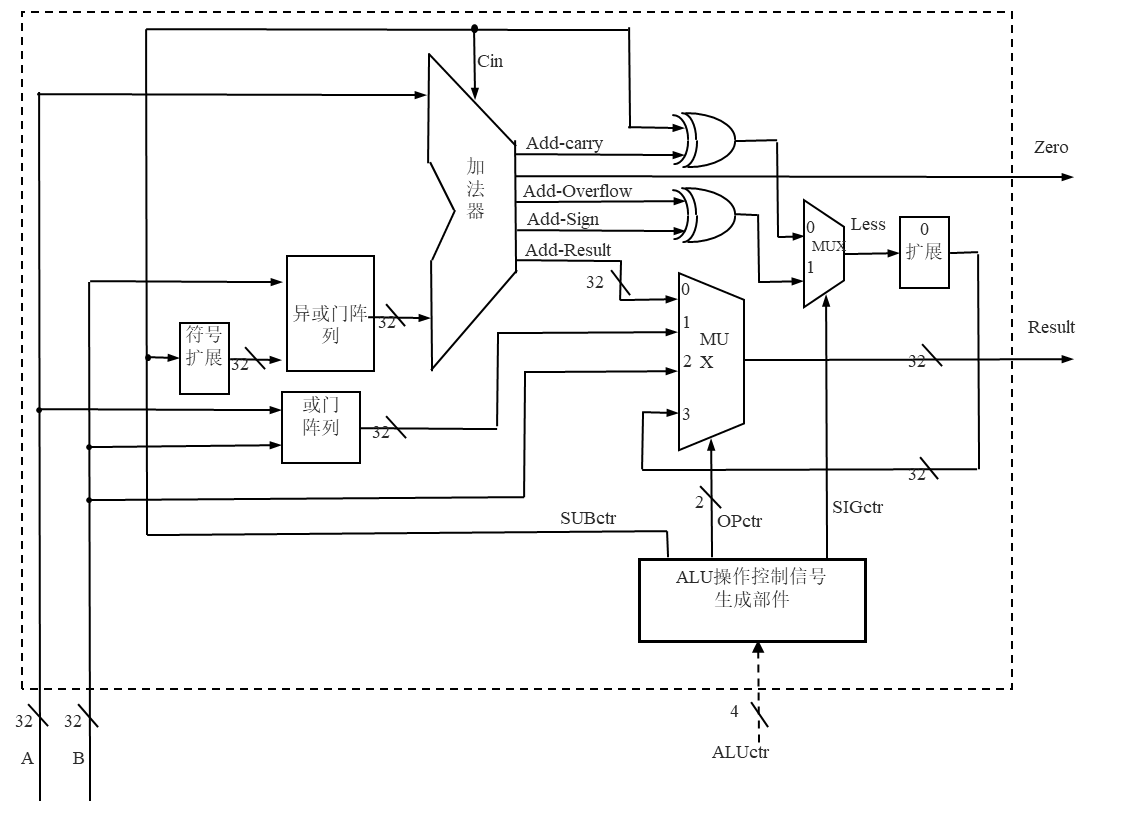
连线



1. 电路仿真检测



1. 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]←R[rs1] + R[rs2] | 加 | 0 | × | 00 |
| slt rd, rs1, rs2 | if (R[rs1] < R[rs2]) R[rd]←1  else R[rd]←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]←1  else R[rd]←0 | 减，无符号数比较大小 | 1 | 0 | 11 |
| ori rt, rs1, imm12 | R[rt]←R[rs1] | SEXT(imm12) | 按位或 | × | × | 01 |
| lui rd, imm20 | R[rt]←imm20||000H | 选择操作数B | × | × | 10 |
| lw rd, rs1, imm12 | Addr←R[rs1] + SEXT(imm12)  R[rd]←M[Addr] | 加 | 0 | × | 00 |
| sw rs1, rs2, imm12 | Addr←R[rs1] + SEXT(imm12)  M[Addr]←R[rs2] | 加 | 0 | × | 00 |
| beq rs1, rs2, imm12 | Cond←R[rs1] – R[rs2] | 减（判0） | 1 | × | ×× |
| if (Cond eq 0)  PC←PC+(SEXT(imm12)×2) | 加 | 0 | × | 00 |
| jal rd, imm20 | R[rd]←PC + 4  PC←PC + (SEXT(imm20)×2) | 加 | 0 | × | 00 |

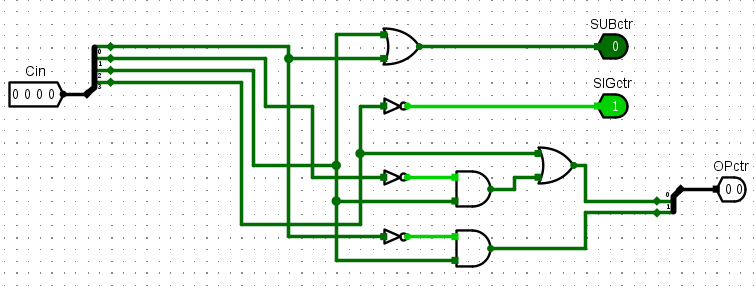
ALUctr的一种四位编码方案

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 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 |

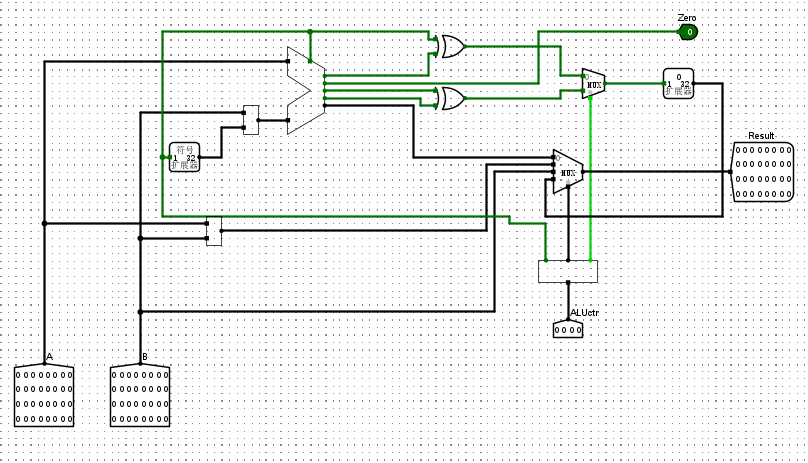
1. 电路原理图如图
2. 实验器材

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

1. 实验步骤
2. ALUCtr



1. ALU连接



1. 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