

## 实验 4: ALU 设计

### 一、实验目的

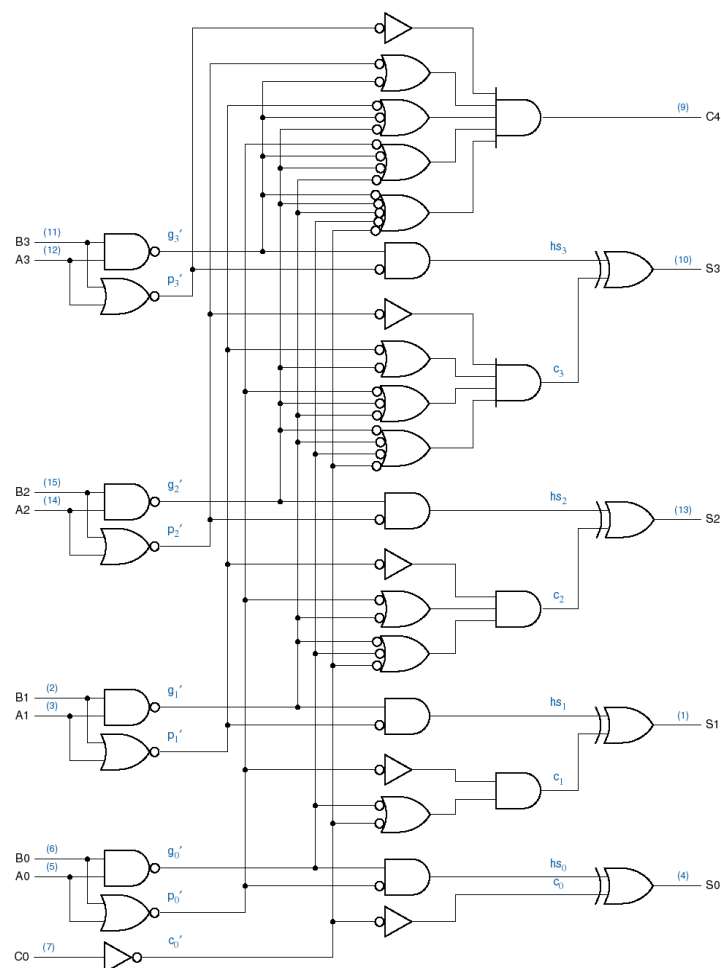
- 1、掌握快速加法器 CLA 和先行进位逻辑 CLU 的设计方法。
- 2、掌握 32 位先行进位加法器及相关标志位的实现方法。
- 3、掌握 ALU 的设计方法, 根据指令要求实现 6 种操作的 ALU 器件。

### 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。



- 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 CLU。

$$C_1 = G_0 + P_0 C_0$$

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

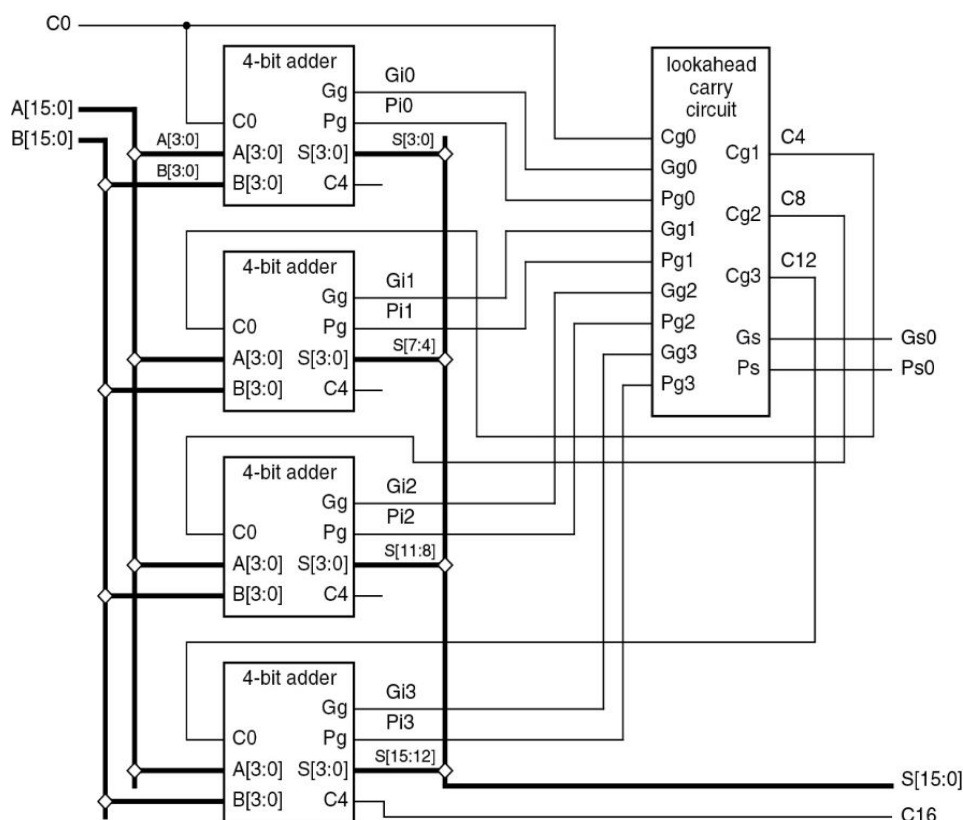
$$C_3 = G_2 + P_2 C_2 = G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_0$$

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

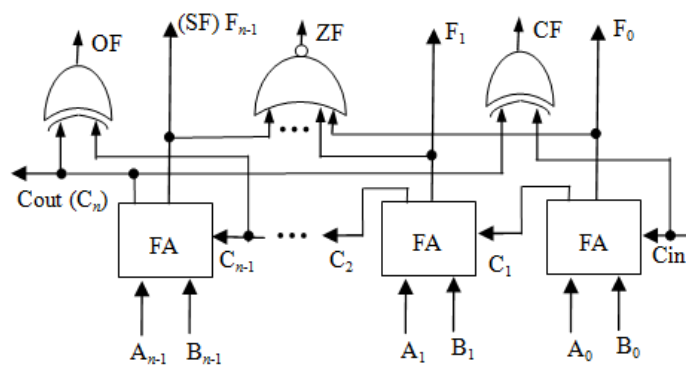
- 3、据给出的逻辑表达式和电路原理图，在 4 位 CLA 中增加支持组间并联的 Gg、Pg 输出端，加上 4 位 CLU，设计并实现 16 位先行进位加法器。

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

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

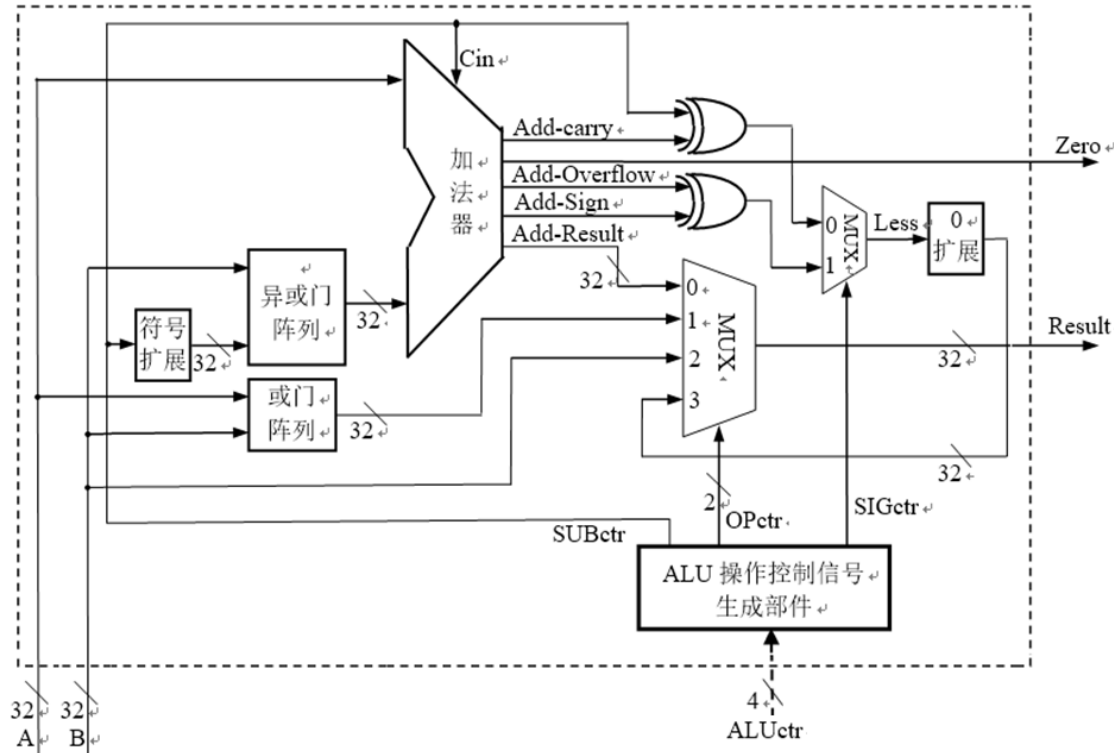


- 4、利用两个 16 位先行进位加法器构建一个 32 位加法器，并根据给出的标志位生成电路原理图，在 32 位加法器中生成 CF、SF、OF、ZF 标志位。



带标志加法器的逻辑电路

5、根据给出的电路原理图和 ALU 引脚定义，设计一个支持 9 条 RV32I 指令所包含的 6 种操作（add、or、slt、sltu、srcB、sub(判 0)）的 32 位 ALU，并对 ALU 的功能进行验证。



ALU 设计原理图

对应的 ALU 操作控制信号取值

指令	功能	操作类型	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	$R[rt] \leftarrow R[rs1] \mid \text{SEXT}(\text{imm12})$	按位或	×	×	01
lui rd, imm20	$R[rt] \leftarrow \text{imm20} \parallel 000H$	选择操作数 B	×	×	10
lw rd, imm12(rs1)	$\text{Addr} \leftarrow R[rs1] + \text{SEXT}(\text{imm12})$ $R[rd] \leftarrow M[\text{Addr}]$	加	0	×	00
sw rs2, imm12(rs1)	$\text{Addr} \leftarrow R[rs1] + \text{SEXT}(\text{imm12})$ $M[\text{Addr}] \leftarrow R[rs2]$	加	0	×	00
beq rs1, rs2, imm12	$\text{Cond} \leftarrow R[rs1] - R[rs2]$	减（判 0）	1	×	×
	if ( $\text{Cond} \text{ eq } 0$ ) $\text{PC} \leftarrow \text{PC} + (\text{SEXT}(\text{imm12}) \times 2)$	加	0	×	00
jal rd, imm20	$R[rd] \leftarrow \text{PC} + 4$ $\text{PC} \leftarrow \text{PC} + (\text{SEXT}(\text{imm20}) \times 2)$	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

## 四、思考题

- 1、若需要增加一条“sub rd,rs1,rs2”指令，则在所设计的 32 位 ALU 中要做哪些修改？
- 2、若需要增加一条“sll rd,rs1,rs2”指令，则在所设计的 32 位 ALU 中要做哪些修改？
- 3、如何验证运算器的结果是否正确？

## 五、实验报告

- 1、根据本次实验内容的要求，写出实验操作步骤，包括：电路原理图、功能表、仿真检测图、输入输出对应表、错误现象及原因分析、思考题等内容。以 word 或 PDF 格式提交
- 2、将实验报告和电路图.circ 文件打包上传到教学支撑平台的网站中。