实验三报告

231220088 陈翔宇

实验内容

一、4位先行进位部件CLU

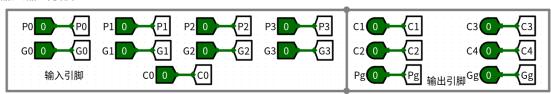
4位先行加法器

根据题目描述可以画出真值表

P0	P1	P2	P3	G0	G1	G2	G3	CO	C1	СЗ	C2	C4	Pg	Gg
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0	1	0	1
0	0	0	0	0	0	0	1	1	0	0	0	1	0	1
0	0	0	0	0	0	1	0	0	0	1	0	0	0	0
0	0	0	0	0	0	1	0	1	0	1	0	0	0	0
0	0	0	0	0	0	1	1	0	0	1	0	1	0	1
0	0	0	0	0	0	1	1	1	0	1	0	1	0	1
0	0	0	0	0	1	0	0	0	0	0	1	0	0	0
0	0	0	0	0	1	0	0	1	0	0	1	0	0	0
0	0	0	0	0	1	0	1	0	0	0	1	1	0	1
0	0	0	0	0	1	0	1	1	0	0	1	1	0	1
0	0	0	0	0	1	1	0	0	0	1	1	0	0	0
0	0	0	0	0	1	1	0	1	0	1	1	0	0	0
0	0	0	0	0	1	1	1	0	0	1	1	1	0	1
0	0	0	0	0	1	1	1	1	0	1	1	1	0	1
0	0	0	0	1	0	0	0	0	1	0	0	0	0	0
0	0	0	0	1	0	0	0	1	1	0	0	0	0	0
0	0	0	0	1	0	0	1	0	1	0	0	1	0	1
0	0	0	0	1	0	0	1	1	1	0	0	1	0	1
0	0	0	0	1	0	1	0	0	1	1	0	0	0	0
0	0	0	0	1	0	1	0	1	1	1	0	0	0	0
0	0	0	0	1	0	1	1	0	1	1	0	1	0	1
0	0	0	0	1	0	1	1	1	1	1	0	1	0	1
0	0	0	0	1	1	0	0	0	1	0	1	0	0	0
0	0	0	0	1	1	0	0	1	1	0	1	0	0	0
0	0	0	0	1	1	0	1	0	1	0	1	1	0	1
0	0	0	0	1	1	0	1	1	1	0	1	1	0	1
0	0	0	0	1	1	1	0	0	1	1	1	0	0	0
0	0	0	0	1	1	1	0	1	1	1	1	0	0	0
0	0	0	0	1	1	1	1	0	1	1	1	1	0	1
0	0	0	0	1	1	1	1	1	1	1	1	1	0	1
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	1	0	0	0	0	0	0
0	0	0	1	0	0	0	1	0	0	0	0	1	0	1
0	0	0	1	0	0	0	1	1	0	0	0	1	0	1
0	0	0	1	0	0	1	0	0	0	1	0	1	0	1
0	0	0	1	0	0	1	0	1	0	1	0	1	0	1
0	0	0	1	0	0	1	1	0	0	1	0	1	0	1
0	0	0	1	0	0	1	0	1	ı	1	0		0	
	0	0	1	0	1	0		0	0	0	1	0	0	0
0	0	0	_1_	0	_1_	0_	0_	_1_	0	0	_1_	0	0	0

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

• CO: 低位进位

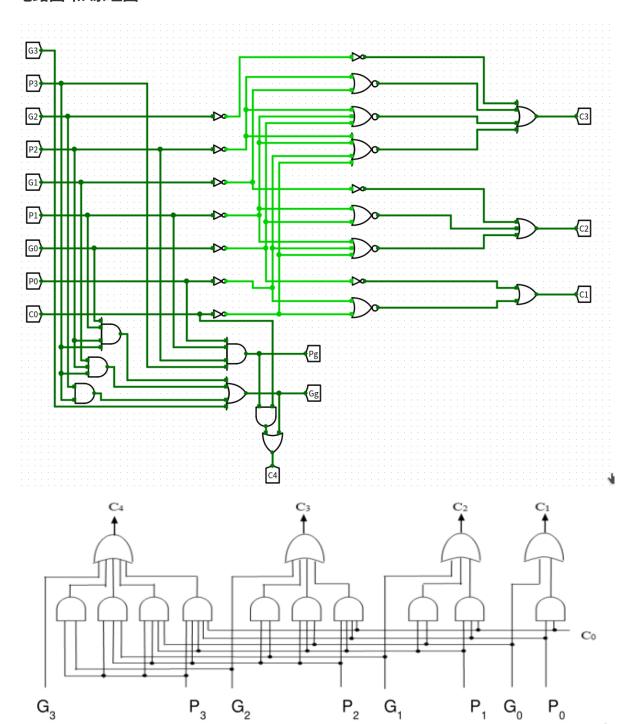
• Pi: 传递函数

• Gi: 生成函数

• Pg: 高位传递函数

• Gg: 高位生成函数

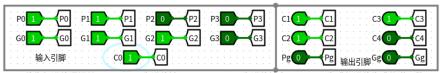
电路图 和 原理图



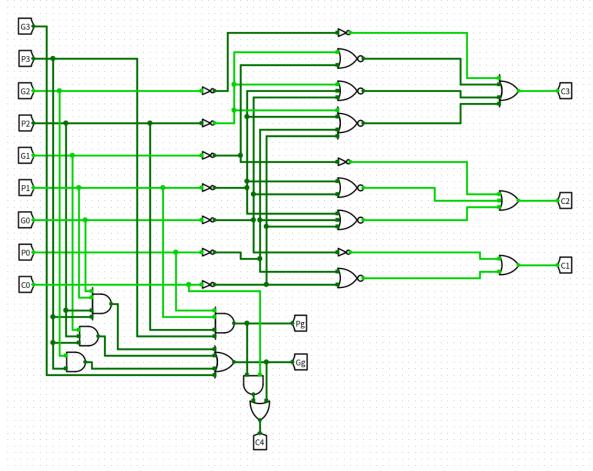
仿真测试

1	Cnt	Х	Υ	Cin	7	Court	Ca	Da
	Cnt				Z	Cout	Gg	Pg
2	00	f	2	0	1	1	1	1
3	01	е	3	0	1	1	1	1
4	02	d	4	0	1	1	1	0
5	03	С	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	Θ
9	07	8	9	0	1	1	1	Θ
10	08	7	а	0	1	1	1	1
11	09	6	b	0	1	1	1	1
12	0a	5	С	0	1	1	1	Θ

0b	4	d	Θ	1	1	1	Θ
0c	3	е	Θ	1	1	1	1
0d	2	f	Θ	1	1	1	1
0e	1	f	Θ	0	1	1	1
0f	0	f	Θ	f	0	0	1
10	f	f	1	f	1	1	1
11	е	f	1	е	1	1	1
12	С	f	1	С	1	1	1
13	a	f	1	a	1	1	1
14	8	7	1	0	1	0	1
15	6	9	1	0	1	0	1
16	4	b	1	0	1	0	1
17	2	d	1	0	1	0	1
18	3	С	1	0	1	0	1
19	5	С	1	2	1	1	0
1a	7	С	1	4	1	1	1
1b	9	С	1	6	1	1	0
1c	b	С	1	8	1	1	1
1d	d	С	1	а	1	1	0
1e	f	С	1	С	1	1	1
1f	0	С	1	d	0	0	0
	0c 0d 0e 0f 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e	0c 3 0d 2 0e 1 0f 0 10 f 11 e 12 c 13 a 14 8 15 6 16 4 17 2 18 3 19 5 1a 7 1b 9 1c b 1d d 1e f	0c 3 e 0d 2 f 0e 1 f 0f 0 f 10 f f 11 e f 12 c f 13 a f 14 8 7 15 6 9 16 4 b 17 2 d 18 3 c 19 5 c 1a 7 c 1b 9 c 1c b c 1d d c 1e f c	0c 3 e 0 0d 2 f 0 0e 1 f 0 0f 0 f 0 10 f f 1 11 e f 1 11 e f 1 12 c f 1 13 a f 1 14 8 7 1 15 6 9 1 16 4 b 1 17 2 d 1 18 3 c 1 18 3 c 1 19 5 c 1 1b 9 c 1 1c b c 1 1d d c 1 1e f c 1	0c 3 e 0 1 0d 2 f 0 1 0e 1 f 0 0 0f 0 f 0 f 10 f 1 f 1 f 11 e f 1 e f 1 e f 11 e f f f f f f f f f f f f f f f f f f	0c 3 e 0 1 1 0d 2 f 0 1 1 0e 1 f 0 0 1 0f 0 f 0 f 0 10 f f 1 f 1 11 e f 1 e 1 11 e f 1 e 1 12 c f 1 c 1 13 a f 1 a 1 13 a f 1 a 1 14 8 7 1 0 1 15 6 9 1 0 1 16 4 b 1 0 1 17 2 d 1 0 1 18 3 c 1 0 1 19 5 c 1 2 1 10 1 1 1	0c 3 e 0 1 1 1 0d 2 f 0 1 1 1 0e 1 f 0 0 1 1 0f 0 f 0 0 0 10 f f 1 f 1 1 11 e f 1 e 1 1 1 11 e f 1 e 1



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



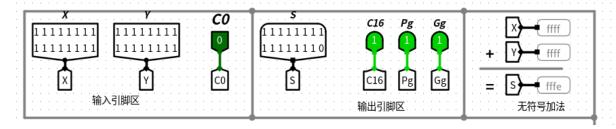
真值表: 真值表

错误现象及分析

在完成实验过程中没有遇到任何错误。

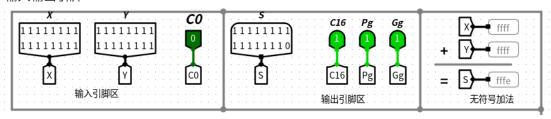
二、16 位两级先行进位加法器实验

基本原理



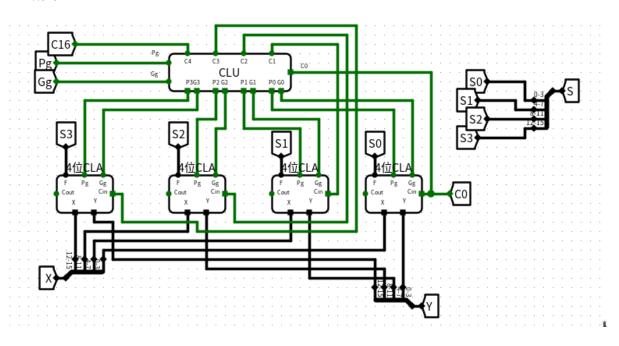
整体方案设计

1. 输入输出引脚

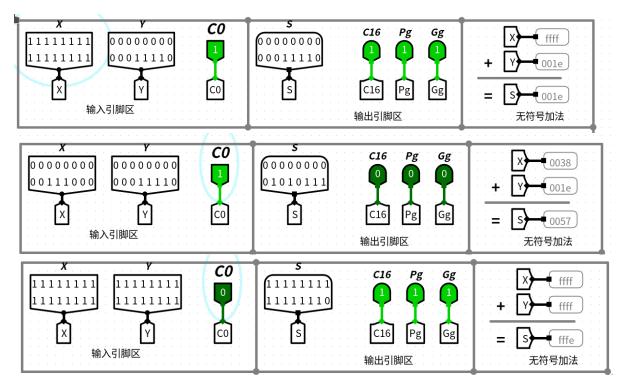


- X: 操作数X
- Y: 操作数Y
- S:和
- CO: 低位进位
- C16: 溢出进位
- Pg: 传递函数
- Gg: 生成函数

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。

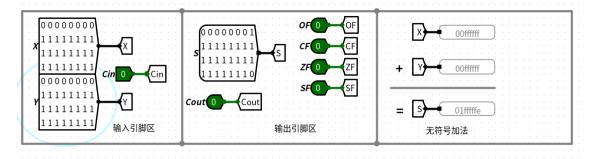
三、32 位快速加法器构建实验

基本原理

- 1 通过先行进位器获得每一位的进位Ci,此时可通过 Si = Xi ⊕ Yi ⊕ Ci直接获取和的第i位。
- 2 由于Ci的计算过程除了Pi、Gi外只依赖低位进位C0,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

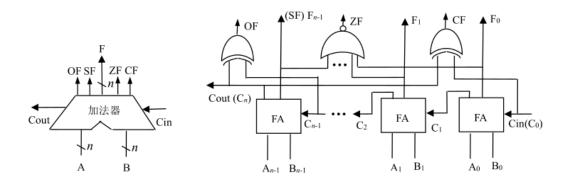
• OF:溢出位

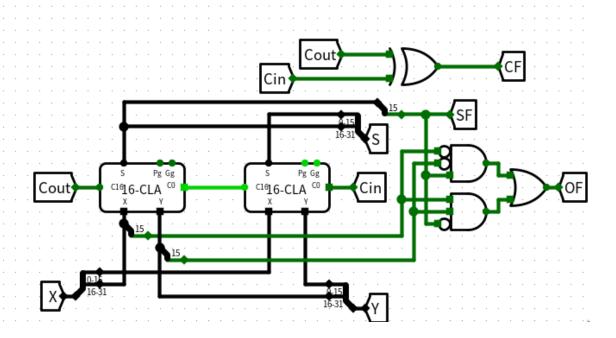
• CF:进位标志

● ZF : O标志位

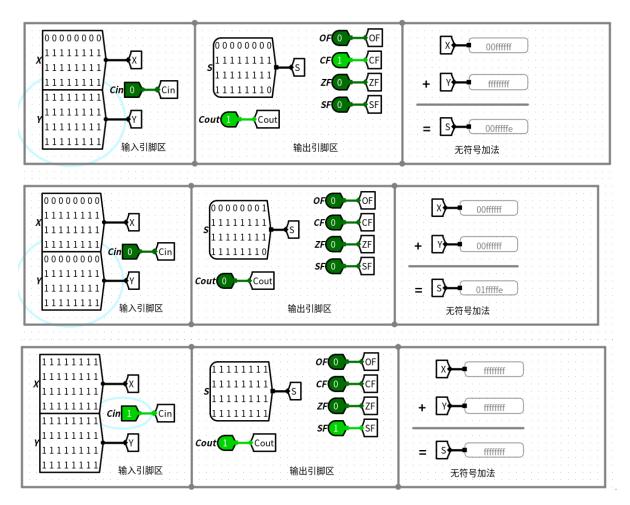
• SF:符号位

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。

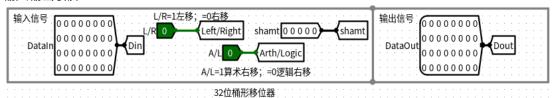
四、32 位桶形移位器设计

基本原理

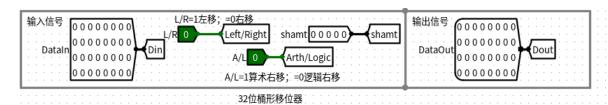
- 在移位操作之前,先将被移位数的最高位(对于左移是左边第一个位,对于右移是右边第一个位)暂存起来。
- 2 将被移位数的其余位按照指定的方向(左移或右移)进行移动指定的位数。
- 3 将暂存的最高位移到移位后的最高位位置,而新产生的最低位则用0或特定的值(如算术移位时用符号位填充)填充。
- 4 将暂存的最高位和移位后的数合并,形成最终的移位结果。

整体方案设计

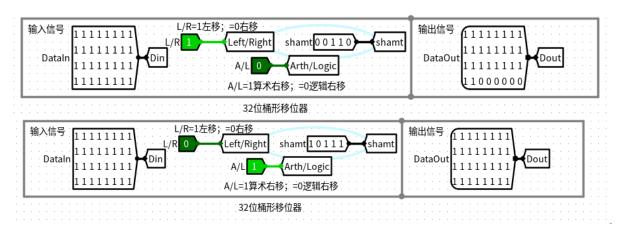
1. 输入输出引脚



电路图



仿真测试



五、ALU 设计实验

基本原理

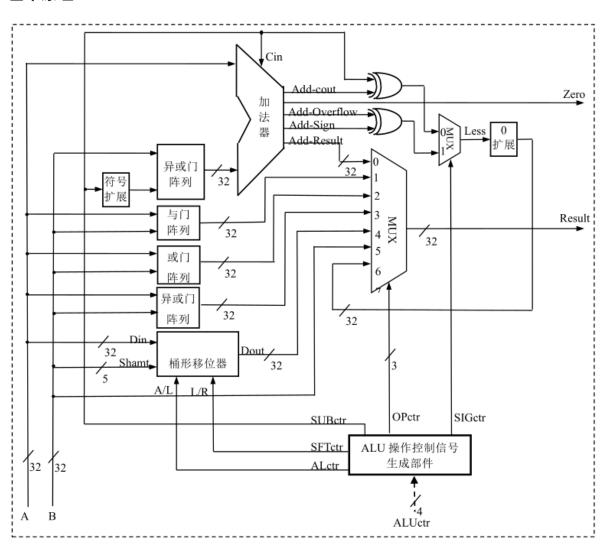


表 4.1 ALUctr 的四位编码及其对应的控制信号

ALUctr <3:0>	指令操作类型	SUBctr	SIGctr	ALctr	SFTctr	OPctr <2:0>	OPctr 的含义
0000	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
0001	sll,slli: 逻辑左移	×	×	0	1	100	选择移位器结果输出
0010	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
0011	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
0100	xor,xori: 异或	×	×			011	选择"按位异或"结果输出

0101	srl,srli: 逻辑右移	×	×	0	0	100	选择移位器结果输出
0110	or,ori: 或运算	×	×	×	×	010	选择"按位或"结果输出
0111	and,andi: 与运算	×	×	×	×	001	选择"按位与"结果输出
1000	sub: 减法	1	×	×	×	000	选择加法器的结果输出
1101	sra,srai: 算术右移	×	×	1	0	100	选择移位器结果输出
其余	(未用)						
1111	lui: 取操作数 B	×	×	×	×	101	选择操作数 B 直接输出

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

 $SIGctr = m_2$

 $ALctr = m_{13}$

 $SFTctr = m_1$

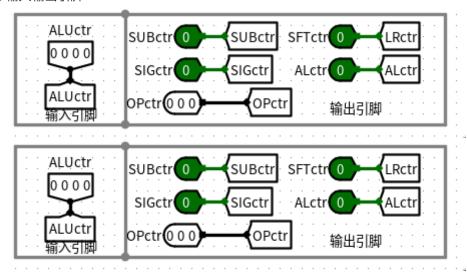
Opctr[2]= $\sum m(1,2,3,5,13,15)$

Opetr[1]= $\sum m(2,3,4,6)$

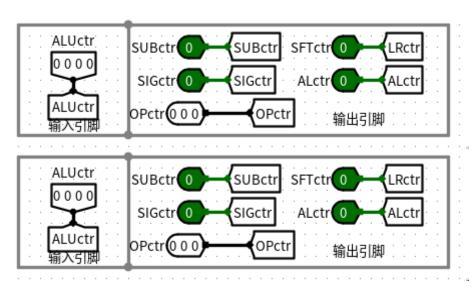
Opctr[0]= $\sum m(4,7,15)$

整体方案设计

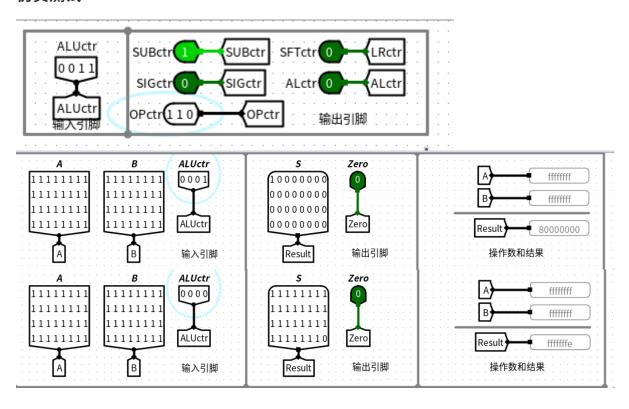
1. 输入输出引脚



电路图

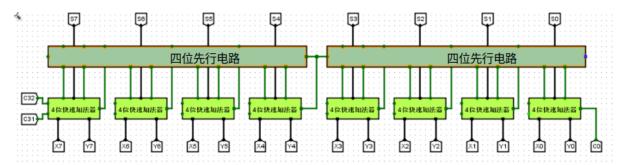


仿真测试



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。