实验四 32 位 ALU 设计实验

原创

zhou_pig 于 2021-04-15 21:15:36 发布 5677 收藏 76

分类专栏： 计算机组成原理

本文链接：https://blog.csdn.net/m0_46252199/article/details/115682105

一、实验预习要求

1. 预习《计算机组成原理》慕课附录 3.4 节、《计算机硬件系统设计》慕课 4.3 节。

二、实验目的

2. 掌握定点数加减法溢出检测方法。
3. 理解算术逻辑运算单元 ALU 的基本构成。
4. 掌握 Logisim 中各种运算组件的使用方法：

1. 逻辑运算：与、或、非、异或
2. 算术运算：乘法器、除法器、求补器、比较器
3. 移位器

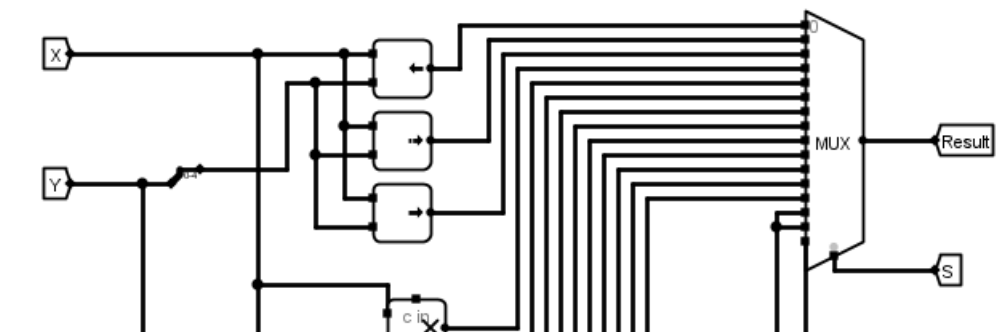
5. 熟练掌握多路选择器的使用方法。
6. 能利用前述实验完成的 32 位加法器（禁止使用 Logisim 自带的加法器/减法器组件）和 Logisim 的运算组件构造指定规格的 ALU 单元。

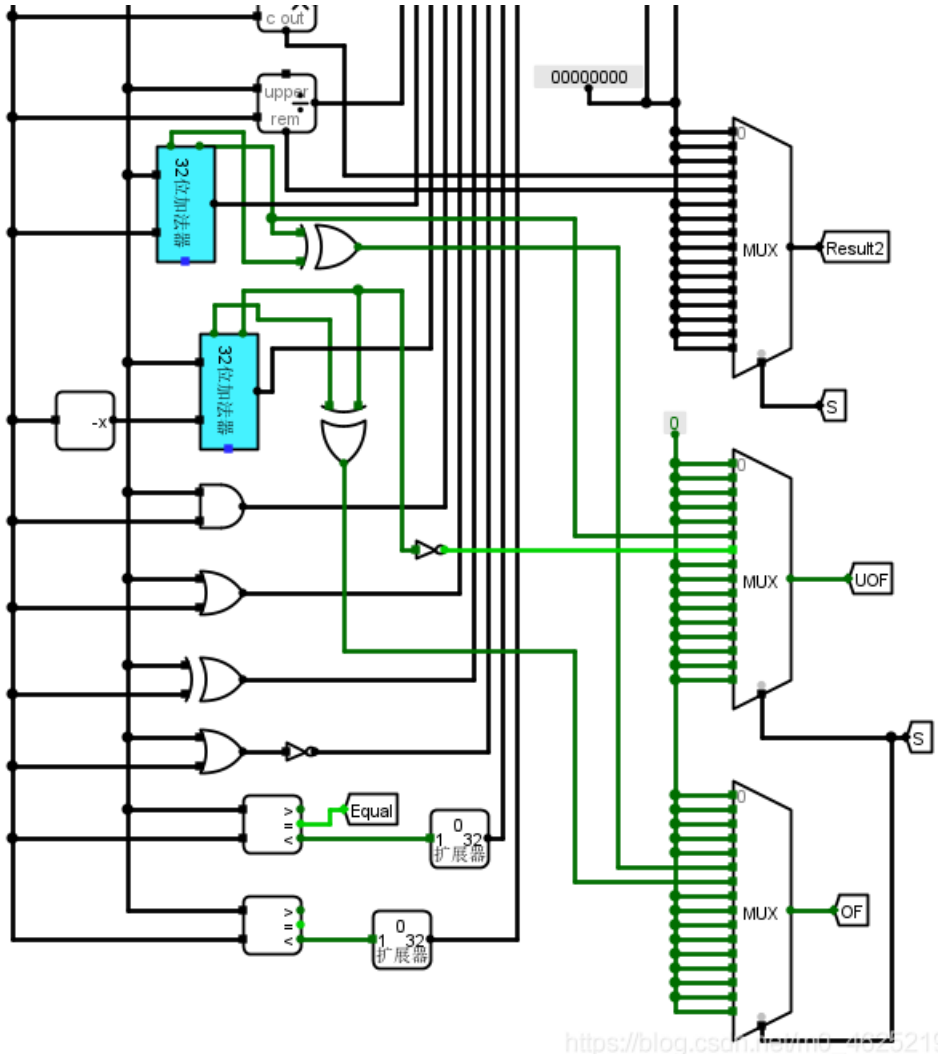
三、实验内容

使用 Logisim 软件打开实验电路图“alu.circ”完成算术逻辑运算单元 ALU 子电路的设计，并在 ALU 自动测试电路中进行测试。

7. 算术逻辑运算单元 ALU 子电路设计。
8. ALU 自动测试。

先上电路图：





实验原理分析：

ALU 运算操作码定义 如下：

ALU OP	十进制	运算功能
0000	0	R = X << Y 逻辑左移 (Y取低五位) R_2=0
0001	1	R = X >> Y 算术右移 (Y取低五位) R_2=0
0010	2	R = X >> Y 逻辑右移 (Y取低五位) R_2=0
0011	3	R = $(X * Y)_{[31:0]}$ R_2 = $(X * Y)_{[63:32]}$ 无符号乘法
0100	4	R = X/Y R_2 = X%Y 无符号除法
0101	5	R = X + Y (Set OF/UOF)
0110	6	R = X - Y (Set OF/UOF)
0111	7	R = X & Y 按位与
1000	8	R = X | Y 按位或
1001	9	R = X⊕Y 按位异或
1010	10	R = ~(X |Y) 按位或非

1011	11	R = (X < Y) ? 1 : 0 有符号比较
1100	12	R = (X < Y) ? 1 : 0 无符号比较

根据OP进行操作的原理

使用**数据选择器**，将op（这里的s）所有取值的对应运算功能实现，最后根据op的值来进行数据选择。

0，1，2操作

这三个操作类似，我们可以一起说。需要实现的运算功能，我们都可以直接在logisim中直接找到。
实验很明显的说到，
Y取低五位，因为x的数据位宽就是32位，所以最多只能被移动5位（2^5=32），取Y低五位我们可以用**分线器**实现。

选区: 分线器(Splitter)	
朝向	右 (东)
分线端口数	1
位宽	32
外观	左手式
第0位对应分线端口	0 (顶部)
第1位对应分线端口	0 (顶部)
第2位对应分线端口	0 (顶部)
第3位对应分线端口	0 (顶部)
第4位对应分线端口	0 (顶部)
第5位对应分线端口	无
第6位对应分线端口	无
第7位对应分线端口	无

3操作

实现乘法使用乘法器即可，由于相乘可能出现溢出，R2对应端口连接高进位。

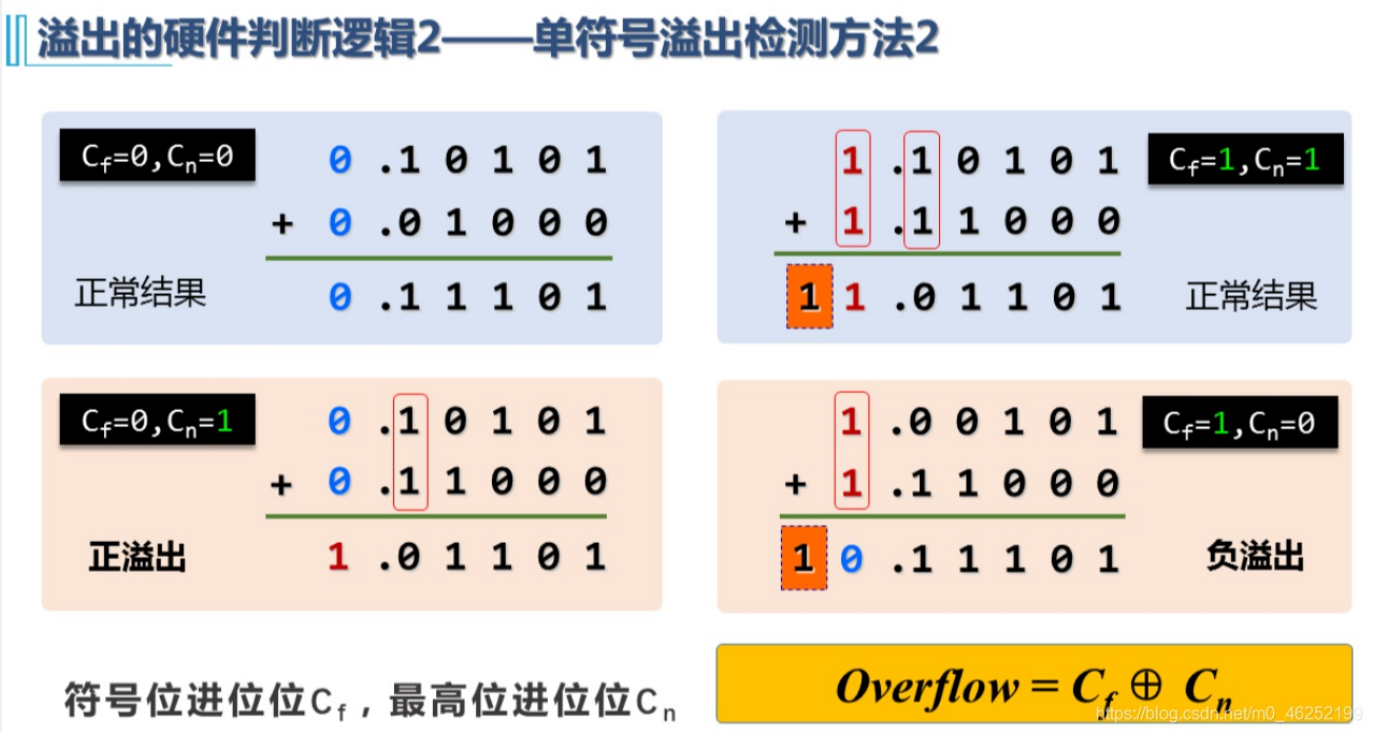
4操作

将x和y连接触发器，将商和余数连接对应端口即可。

5，6操作

加法直接连接加法器输入端即可，减法我们可以看成x+(-y)，使用一个**求补器**将y的补码送进加法器。
有符号加减法溢出，我们将最高位和符号位进位异或（下面有图片）。
无符号加法溢出很好理解，如果**最高的两位相加，产生了进位，那么肯定就是溢出了！**
无符号减法，**因为我们求补码多加了一个模，按理说就应该会产生一个进位，所以当进位为1是，其实并未溢出。进位为0，说明1（应该进的）+1（溢出的）=>0,此时是真正溢出了！所以我们要取反！**

引用一张ppt吧，嘿嘿，挺简明的。



7，8，9，10操作

根据需要实现的逻辑功能，连接对应即可。

11，12操作

我们可以使用现成的**比较器**进行比较，同时注意比较器中有 无符号和有符号的 选项，注意选择。
此外，由于比较结果数据位宽为1位，而选择器数据位宽为32位，这里需要使用**位扩展器**。

选区: 比较器(Comparator)	
数据位宽	32
比较器类型	无符号

选区: 比较器(Comparator)	
数据位宽	32
比较器类型	2的补码

[外链图片转存中…(img-luESZXLn-1618490852793)]

其他细节

1） 由于不能有引脚悬空，所以**不输入的都要接地**
2）**result2只在X和Y进行乘除操作才会有数值**，其他时候都是0,这同样可以使用一个 数据选择器实现
3）同样的，**UOF和OF只在X和Y进行加减操作才会有数值**，其他时候都是0,同样可以使用一个 数据选择器实现
4）Equal可以在比较器的 '=' 那里直接连接