

## 6.9 实验九 行为级 ALU

### 1. 实验介绍

在本次实验中，我们将使用 Verilog HDL 语言实现行为级 ALU 的设计和仿真。

### 2. 实验目标

- 深入了解 ALU 的原理
- 学习使用 Verilog HDL 语言进行行为级 ALU 的设计与仿真。

### 3. 实验原理

ALU 是负责运算的电路。ALU 必须实现以下几个运算：加(ADD)、减(SUB)、与(AND)、或(OR)、异或(XOR)、置高位立即数(LUI)、逻辑左移与算数左移(SLL)、逻辑右移(SRL)以及算数右移(SRA)、SLT、SLTU 等操作。输出 32 位计算结果、carry(借位进位标志位)、zero(零标志位)、negative(负数标志位)和 overflow(溢出标志位)。

本实验实现 ALU 的基本思想是：在操作数输入之后将所有可能的结果都计算出来，通过操作符 aluc 的输入来判别需要执行的操作来选择需要的结果进行输出。图 6.9.1 所示为本实验的 ALU 参考原理图。表 6.9.1 所示为 aluc 的值所对应的运算。表 6.9.2 所示为 addsub32 标志位规则（仅供参考）。

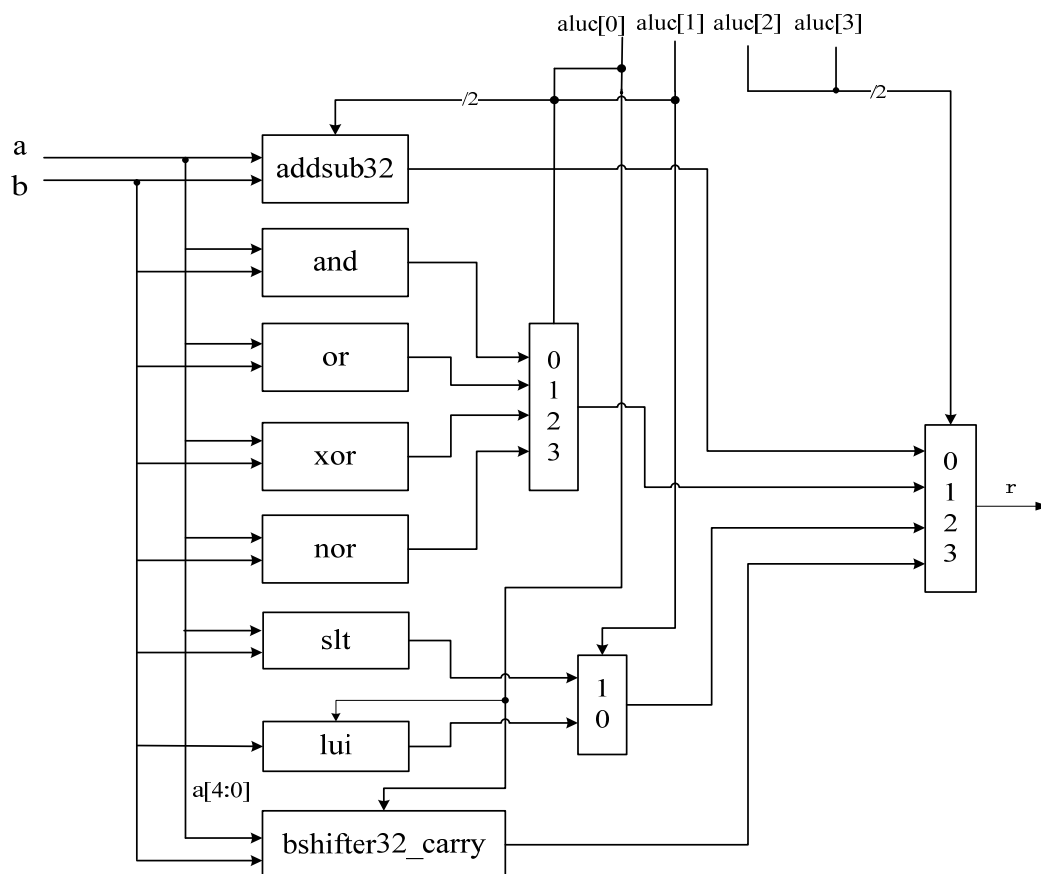


图 6.9.1 ALU 的原理图供参考

表 6.9.1 aluc 的值所对应的运算

	aluc[3]	aluc[2]	aluc[1]	aluc[0]
Addu r=a+b 无符号	0	0	0	0
Add r=a+b 有符号	0	0	1	0
Subu r=a-b 无符号	0	0	0	1
Sub r=a-b 有符号	0	0	1	1
And r=a & b	0	1	0	0
Or r=a   b	0	1	0	1
Xor r=a ^ b	0	1	1	0
Nor r=~ (a   b)	0	1	1	1
Lui r={b[15:0],16'b0}	1	0	0	X
Slt r=(a<b)?1:0 有符号	1	0	1	1
Sltu r=(a<b)?1:0 无符号	1	0	1	0
Sra r=b>>>a	1	1	0	0
Sll/Slr r=b<<a	1	1	1	X
Srl r=b>>a	1	1	0	1

表 6.9.2 ALU 标志位规则

zero 标志位	1. $z=1$ 表示运算结果是零, $z=0$ 表示运算结果不是零。 2. 对于 Slt 和 Sltu 运算, 如 $a-b=0$ , 则 $z=1$ , 表示进行比较的两个数大小相等。 3. 所有运算均影响此标志位。
carry 标志位	1. 无符号数加法运算 (Addu) 发生上溢出, 则该标志位为 1。 2. 无符号数减法运算 (Subu) 发生下溢出, 则该标志位为 1。 3. 无符号数比较运算 (Sltu), 如 $a-b<0$ , 则该标志位为 1。 4. 移位运算, 该标志位为最后一次被移出的位的数值 (在移位模块实现)。 5. 其他运算不影响此标志位。
negative 标志位	1. 有符号数运算 Add 和 Sub, 操作数和运算结果均采用二进制补码的形式表示, $N=1$ 表示运算的结果为负数, $N=0$ 表示结果为正数或零。 2. 有符号数比较运算 (Slt), 如果 $a-b<0$ , 则 $N=1$ 。 3. 其他运算, 运算最终结果的最高位 $r[31]$ 为 1, 则 $N=1$ 。
overflow 标志位	1. 对于有符号加减法运算 (Add 和 Sub), 操作数和运算结果均采用二进制补码的形式表示, 有溢出时该标志位 $o=1$ 。 2. 只有有符号加减法运算影响此标志位。

● 接口定义:

```
module alu(  
    input [31:0] a,      //32 位输入, 操作数 1  
    input [31:0] b,      //32 位输入, 操作数 2  
    input [3:0] aluc,    //4 位输入, 控制 alu 的操作  
    output [31:0] r,     //32 位输出, 由 a、b 经过 aluc 指定的操作生成  
    output zero,         //0 标志位  
    output carry,        // 进位标志位  
    output negative,     // 负数标志位  
    output overflow      // 溢出标志位  
);
```

**提示:** 本次实验允许使用行为级建模方式实现 ALU, 可以使用“+”“-”“<”“>”等运算符号实现 ALU 中的计算模块

#### 4. 实验步骤

1. 新建 Vivado 工程, 用 Verilog 行为描述编写 ALU 各个模块, 将这些模块组成一个 ALU。
2. 使用 ModelSim 进行仿真, 测试 ALU 的每个模块, 验证 ALU 的正确性
3. 按照要求书写实验报告