实验一 32位ALU

# 支持的运算

由于这里没有用寄存器用于存放64位数据，所以对于乘法、除法这里不支持，而且只支持双目运算符。本ALU支持的运算如下：

* 加
* 减
* 与
* 或
* 异或
* 或非
* 左移
* 右移

# 标志位

1. 符号标志：如果结果为负数，则为1，否则为0
2. 进位标志：对于无符号数来说有用
3. 零标志：结果是否为0，如果结果为0，则为1
4. 溢出标志：对于有符号数来说有用
5. 奇偶标志：是否包含偶数个1，如果是，则为1，否则为0

# 思路

本实验较为简单，这里我直接使用了Verilog的操作符号+、-等。主要的思路就是根据select信号来选择不同的运算操作。对于标志位的计算，思路如下：

* 符号标志：与32位结果的高位相同
* 进位标志：对于加法运算，判断结果是否小于两个操作数，对于减法运算，判断结果是否大于被减数
* 零标志：判断32位结果是否为0
* 溢出标志：用双符号位数据进行运算时，两符号位如果不同代表发生了溢出
* 奇偶标志：对结果进行自身位异或，然后取反

# 仿真结果

测试用例如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 操作数1 | 操作符 | 操作数2 | 预期结果 | 预期标志符 |
| 0x8000\_0000 | + | 0x8000\_0000 | 0x0000\_0000 | SF=0 CF=1 ZF=1 OF=1 PF=1 |
| 0x8000\_0000 | - | 0x0000\_0001 | 0x7FFF\_FFFF | SF=0 CF=0 ZF=0 OF=1 PF=0 |
| 0x0000\_00FF | & | 0x0000\_00F0 | 0x0000\_00F0 | SF=0 CF=0 ZF=0 OF=0 PF=1 |
| 0x0000\_00F0 | | | 0x0000\_000F | 0x0000\_00FF | SF=0 CF=0 ZF=0 OF=0 PF=1 |
| 0x0000\_00FF | ^ | 0x0000\_000F | 0x0000\_00F0 | SF=0 CF=0 ZF=0 OF=0 PF=1 |
| 0x0000\_0000 | 或非 | 0x0000\_000F | 0xFFFF\_FFF0 | SF=1 CF=0 ZF=0 OF=0 PF=1 |
| 0x0000\_0001 | << | 0x0000\_000C | 0x0000\_1000 | SF=0 CF=0 ZF=0 OF=0 PF=0 |
| 0x0000\_1000 | >> | 0x0000\_000C | 0x0000\_0001 | SF=0 CF=0 ZF=0 OF=0 PF=0 |

仿真结果如下：

图形用户界面

描述已自动生成

可以看到，两者的结果相符。