

第四次实验—ALU 实验

黄德立—20337200

实验要求

- ①. 了解算术逻辑单元 ALU 的原理;
- ②. 熟悉并运用 Verilog 语言设计 ALU;
- ③. 学习寄存器堆的数据传送与读写工作原理, 掌握寄存器读写的设计方法;
- ④. 熟悉并运用 Verilog 语言设计 ALU;
- ⑤. 学习 Verilog 不同形式的编程方式, 理解 assign 和 always 的区别;

实验设计

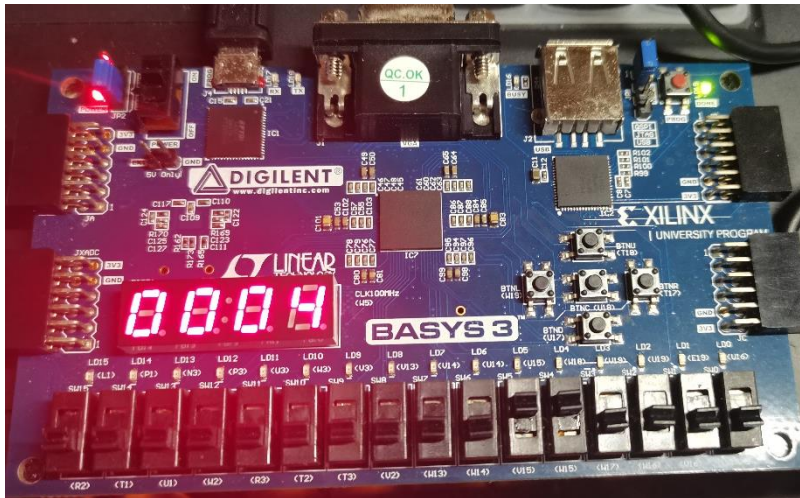
1. 了解 ALU 算术逻辑单元包括加减乘运算和与或操作。
2. 完成 calculate 子模块

```
module calculate(  
    input wire [7:0] num1,  
    input wire [2:0] op,  
    output [31:0] result  
);  
    wire [31:0] num2;  
    wire [31:0] Sign_extend;  
    reg [31:0] result;  
    assign num2 = 32'h01;  
    assign Sign_extend = {24*num1[7], num1};  
  
    always @(*) begin  
        if(op==3'b000)  
            result=Sign_extend+num2;  
        else if(op==3'b001)  
            result=Sign_extend-num2;  
        else if(op==3'b010)  
            result=Sign_extend&num2;  
        else if(op==3'b011)  
            result=Sign_extend|num2;  
        end
```

其中 op 代表选择的算术操作, num2 在模块中固定为 1。为完成 32 位的算术计算, 还需要将 num1 进行符号扩展。扩展的方法是将首位数扩展即可。

实验结果

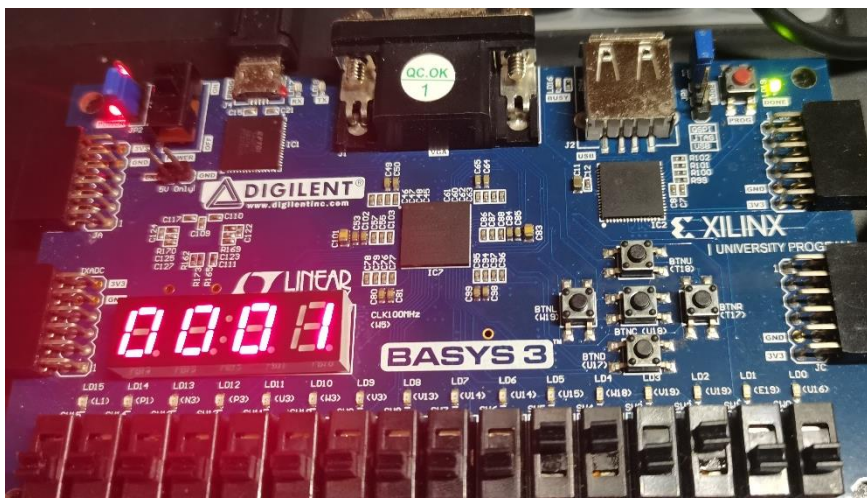
- 选择加法，输入为 3 的时候：



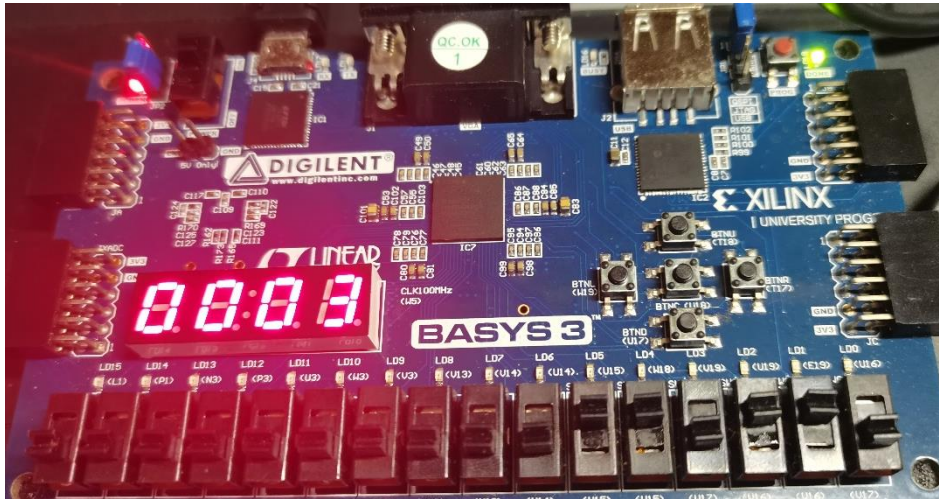
- 选择减法，输入为 3 的时候：



- 选择与运算，输入为 3 的时候：



- 选择或运算，输入为 3 的时候：



实验总结

通过这次实验四, 我掌握了 ALU 的基础原理, 同时也更加熟练掌握了 Vivado 的使用方法。

除此之外, 在实验过程中, 我也收获了自己设计完成实验目标的快乐, 让我对计算机组成原理这门科目更加热爱!