

Lab02实验报告

FPGA基础实验：4-bitAdder

- 一、实验目的
- 二、实现步骤
 - 1.adder_1bit.v(部分)
 - 2.adder_4bits.v(部分)
 - 3.仿真激励文件adder_4bits_tb.v(部分)
 - 4.仿真结果：
- 三、总结与心得体会
 - 1.总结
 - 2.心得体会

Lab02实验报告

FPGA基础实验：4-bitAdder

518021910489 陈沛宇

一、实验目的

1. 掌握Xilinx逻辑设计工具 Vivado的基本操作。
2. 掌握 VerilogHDL进行简单的逻辑设计。
3. 使用功能仿真。

二、实现步骤

1.adder_1bit.v(部分)

```
wire s1, c1, c2, c3;  
    and (c1, a, b),  
        (c2, b, ci),  
        (c3, a, ci);  
xor (s1, a, b),  
    (s, s1, ci);  
  
or (co, c1, c2, c3);
```

通过逻辑运算实现一位加法器。

2.adder_4bits.v(部分)

```
wire [2:0] ct;  
  
adder_1bit a1(.a(a[0]), .b(b[0]), .ci(ci), .s(s[0]), .co(ct[0])),  
    a2(.a(a[1]), .b(b[1]), .ci(ct[0]), .s(s[1]), .co(ct[1])),  
    a3(.a(a[2]), .b(b[2]), .ci(ct[1]), .s(s[2]), .co(ct[2])),  
    a4(.a(a[3]), .b(b[3]), .ci(ct[2]), .s(s[3]), .co(co));
```

调用了adder_1bit实现了adder_4bits。

Source一栏出现了如下的层次结构:



3.仿真激励文件adder_4bits_tb.v(部分)

```
reg[3:0] a;
reg[3:0] b;
reg ci;

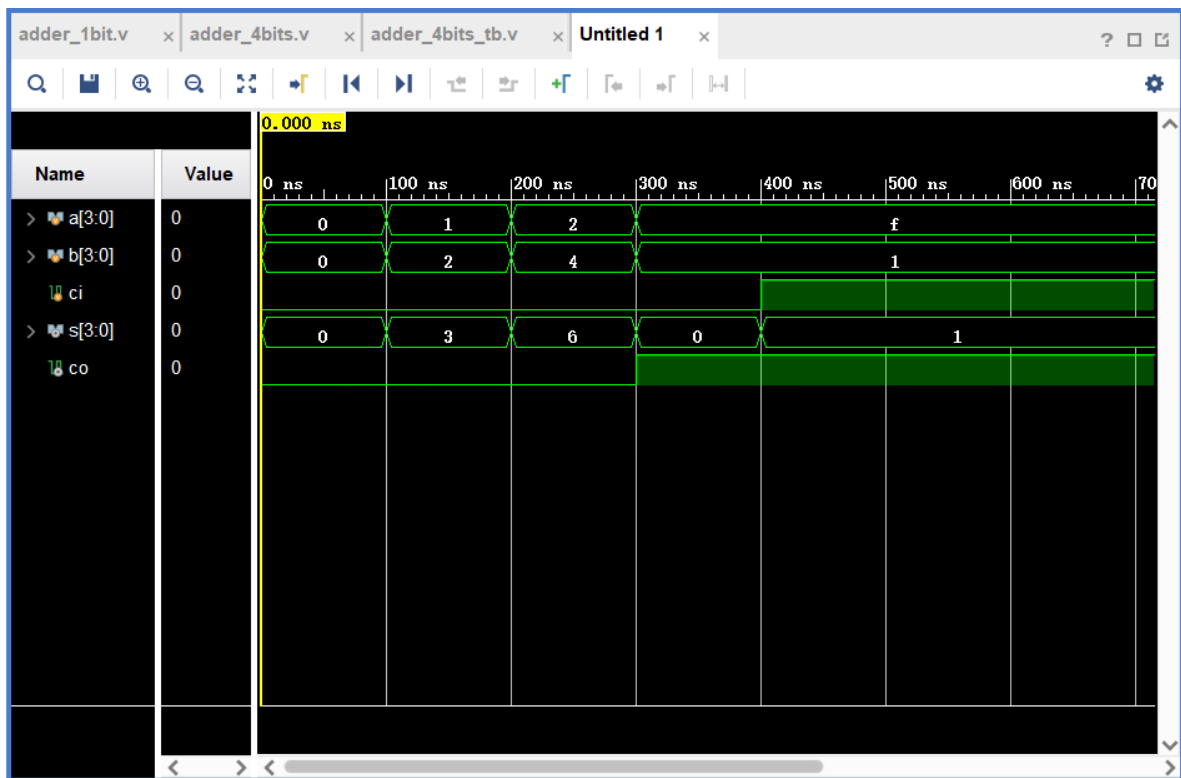
wire[3:0] s;
wire co;

adder_4bits u0(
    .a(a),
    .b(b),
    .ci(ci),
    .s(s),
    .co(co)
);

initial begin
    a = 0;
    b = 0;
    ci = 0;

    #100;
    a = 4'b0001;
    b = 4'b0010;
    #100;
    a = 4'b0010;;
    b = 4'b0100;
    #100;
    a = 4'b1111;
    b = 4'b0001;
    #100;
    ci = 1'b1;
end
```

4.仿真结果:



对仿真结果的解释：观察仿真激励文件`adder_4bits_tb.v`，作为input的a、b、ci初值为0，#100表示经过100ns后，修改a、b的值，此时可以看到作为output的s的值发生了改变，恰好为假发的结果，后续代码原理相同，在ci加入后结果s和co也是正确的。

三、总结与心得体会

1.总结

对源代码一些操作的总结：

```
module, endmodule //定义一个基本模块，两个关键字间的语句是对该模块的描述。
output, input     //关键字指定输入输出
reg               //变量，用<=赋值
wire              //变量，常用于表述以assign关键字指定的组合逻辑信号
reg/wire[n:0]     //代表数据的位宽
assign            //描述组合逻辑的关键字
#100              //指令指明延时，单位可指定
```

2.心得体会

Lab02相比Lab01在源代码上更加复杂，但教会了我如何在Verilog的语法下设计逻辑进行运算，也让我学习到了数据的声明和指令的延时等操作，让我对Verilog的理解更深了一步。但现在阅读Verilog代码仍然觉得很晦涩，还需要多阅读参考资料熟悉语法和设计方法。