计算机系统结构实验 Lab2

David Wang

2020 年 5 月

1 概述

1.1 实验名称

FPGA 基础实验：四位加法器

1.2 实验目的

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

2 实验步骤

2.1 新建工程

1. 启动桌面 Vivado 2018.3 开发工具
2. 点击 Create Project
3. 弹出 New Project, 建立一个新工程，点击 Next
4. 输入工程名称，确认勾选中 Create project subdirectory 后点击 Next
5. 选择 RTL Project 工程类型，勾选 Do not specify at this time, 点击 Next
6. 选择 FPGA 参数，点击 Next, 再点击 Finish 结束工程构建

2.2 添加文件

选择 Add or Create Design Sources, 为模块命名，定义好设计模块所需端口。

2.3 输入代码

一位加法器代码如下。

```
    module adder_1bit(
    input a,
    input b,
    input ci,
    output s,
    output co
    );
    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);
endmodule
```

接下来是四位加法器

```
    module adder_4bits(
    input [3:0] a,
    input [3:0] b,
    input ci,
    output [3:0] s,
    output co
    );

    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));
endmodule
```

2.4 仿真程序

```
`timescale 1ns / 1ps


module adder_4bits_tb(    );
    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
endmodule
```

2.5 仿真波形图

从下图可以看出，当给定操作数时，加法器总能给出正确结果。实验成功。

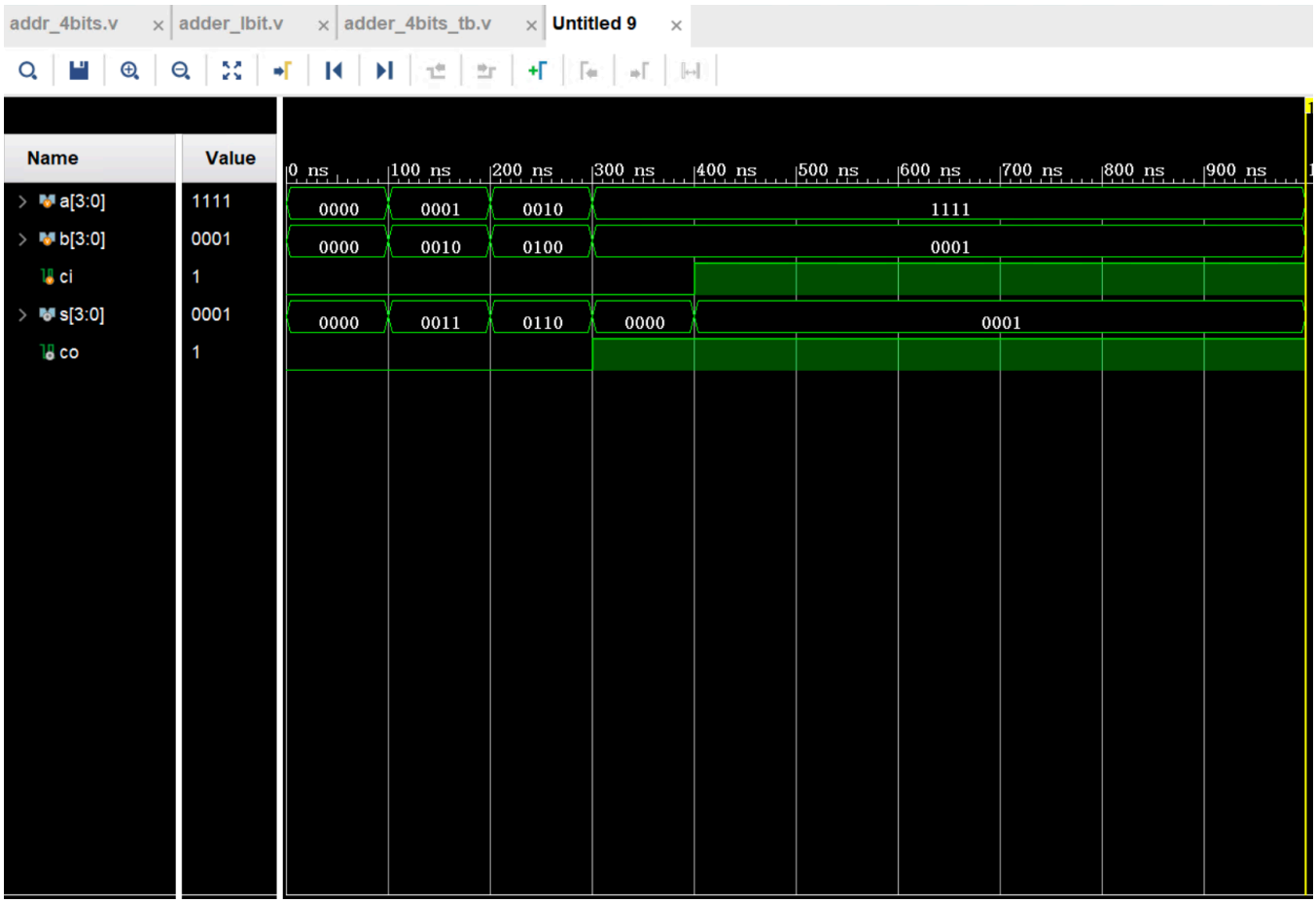
addr_4bits.v
adder_1bit.v
adder_4bits_tb.v
Untitled 9
Name
Value
a[3:0]
1111
b[3:0]
0001
ci
1
s[3:0]
0001
co
1
0 ns
100 ns
200 ns
300 ns
400 ns
500 ns
600 ns
700 ns
800 ns
900 ns
0000
0001
0010
1111
0000
0010
0100
0001
0000
0011
0110
0000
0001

图 1:

3 实验感想

这是我第一次接触逻辑设计工具，操作还不够熟练。代码中因为变量命名不统一的问题发生了很多次波形出现未定值或高阻抗的情况。这启示我应该明确变量名的命名规范，这样才能少犯错误，从而节省实验时间。

Verilog 中的一些关键词如 reg、wire，以及 always @ 语句块的含义在一开始时不是那么容易就能弄懂。我经过查找资料，对于 reg 与 wire 的区别有了一些感性的认识。对于 always 语句块的含义也有了一定程度的了解。我期待将我所学到的知识应用于实验中。