

计算机结构实验 lab02

张露伊

2022 年 3 月 15 日

摘要

本实验实现了 4 位全加器。通过实现一位全加器（包括与、或、异或计算）。再通过 top 模块将 4 个一位全加器连线从而构成 4 位全加器。

目录

1	实验介绍	2
1.1	实验名称	2
1.2	实验目的	2
2	原理分析	2
3	功能验证	3
4	心得体会	3

1 实验介绍

1.1 实验名称

FPGA 基础实验：4-bit Adder

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 VerilogHDL 进行简单的逻辑设计
3. 使用功能仿真
4. 约束文件的使用和直接写法
5. 生成 Bitstream 文件

2 原理分析

4 位加法器是计算机系统的基本逻辑器件。被加数 a、加数 b 从低位向高位进位 C_i 作为电路的输入，全加和 S 与向高位的进位 C_o 作为电路的输出。本实验主要实现 4 位加法器，即对两个输入的 4 位的数进行求和并输出 4 位的结果。而 4 位全加器的实现是基于一位全加器的整合。故本实验手写实现 1 位全加器进而实现 4 位全加器。

下面是 1 位全加器的实现代码：4 位全加器的实现代码：

```

23 module adder_1bit(
24     input a,
25     input b,
26     input ci,
27     output s,
28     output co
29 );
30 wire s1,c1,c2,c3;
31 and (c1,a,b),
32     (c2,b,ci),
33     (c3,a,ci);
34
35 xor (s1,a,b),
36     (s,s1,ci);
37
38 or  (co,c1,c2,c3);
39
40 endmodule

```

图 1: adder_1bit.v

```

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

```

图 2: adder_4bit.v

3 功能验证

本实验采用软件仿真的方法进行测试。用 Verilog 编写测试激励文件进行仿真。观察各信号波形对比是否符合预期。

激励测试文件主要代码如下：

```
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
```

图 3: adder_4bit_tb

仿真运行后得到的波形图如下：

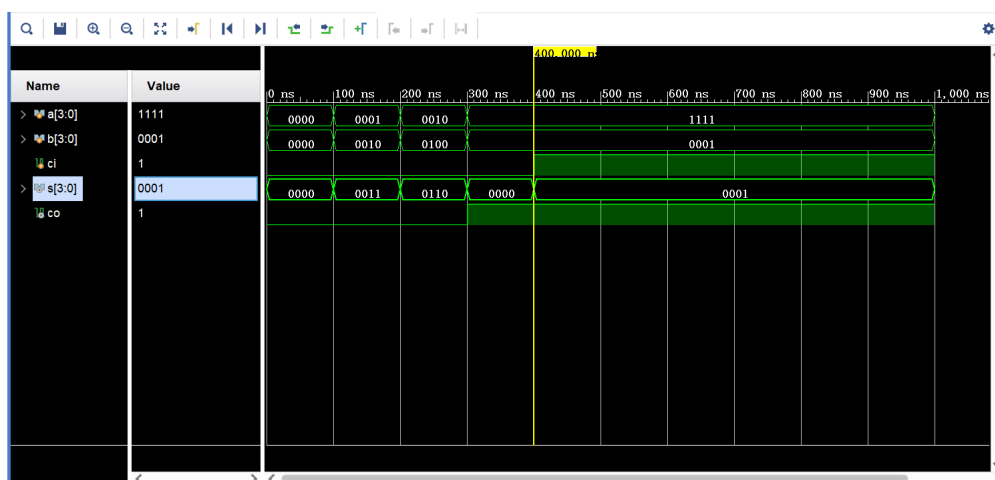


图 4: 仿真波形图

观察分析可知，仿真结果与逻辑功能一致，电路能正常工作。

4 心得体会

在本实验中我学习到 4 位全加器是基于 1 位全加器而实现的，因此如果想实现 4 位全加器首先要实现 1 位全加器。起初自己并不理解代码的含义，比如 ci 和 co 的含义，在了解了全加器的实现原理之后才知道 ci 和 co 表示的是进位。而在理解组件的实现原理之后再看代码就很容易理解了。因此我明白了想要将组件实现好的首要前提是要理解器件实现功能的原理，在充分理解原理的情况下书写代码才能实现更加容易成功。