

计算机系统结构实验 Lab01

朱凡悦 522031910547

May 2024

1 摘要

在 Lab02 中，通过 Verilog 语言实现了 4 位加法器功能，学习了逻辑的实现，约束文件的使用和直接写法，收获颇丰。

目录

1 摘要	1
2 实验目的	2
3 原理分析	2
4 功能实现	3
5 结果验证	3
5.1 测试用激励文件	3
5.2 加法逻辑的测试	4
6 管脚约束	4
7 总结与反思	4

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
6. 上板验证 (暂不做)

3 原理分析

根据全加器的真值表，容易推出其表达式如下。

$$sum = a \oplus b \oplus cin + a \& b \& \overline{cin} + b \& a \& \overline{cin} + a \& b \& cin = abin$$

$$cout = a \& b \& c + b \& a \& c + c \& a \& b + a \& b \& c = (a \& b) \vee (a \& cin) \vee (b \& cin)$$

根据表达式易得其实现代码。

```
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);
```

图 1: 全加器实现代码

4 位加法器是在全加器的基础上组合实现。

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

图 2: 4 位加法器实现代码

4 功能实现

基于上述原理，易于实现 `adder_1bit.v` 和 `adder_4bits.v`。实现后，生成 `adder_4bits_tb.v` 的激励文件用以仿真测试，生成 `lab02_xdc.xdc` 的管脚约束用以练习。

5 结果验证

5.1 测试用激励文件

实现代码如下。

```
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: 测试用激励文件实现代码

5.2 加法逻辑的测试

运行行为仿真，得到结果如下。可见加法逻辑运行正常。

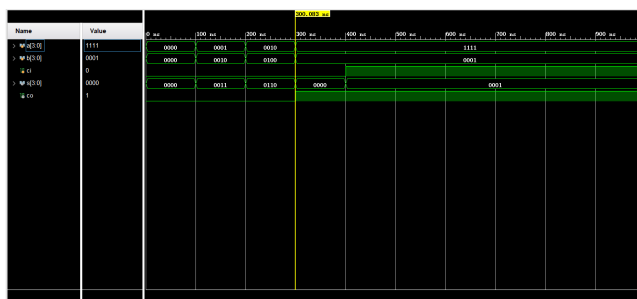


图 4: 仿真结果

6 管脚约束

见 lab02_xdc.xdc 文件

7 总结与反思

本次实验使我上手了 Verilog 逻辑的实现，并且学习了约束文件的直接书写，感谢助教和老师的课上指导和课程组提供的指导书。