

# 计算机系统结构实验报告 Lab02

## FPGA 基础实验：4-bit adder

徐阳 521021910363

2023 年 5 月 1 日

### 目录

1 简介	2
2 实验目的	2
3 实验原理与功能实现	2
3.1 一位全加器 . . . . .	2
3.2 四位全加器 . . . . .	2
3.3 仿真验证 . . . . .	3
4 上板验证	3
5 总结与思考	4
6 致谢	5

## 1 简介

在本 Lab 中，我在中 Vivado 实现了一个 4 位全加器模块，该模块输入两个 4 位二进制数和上一次进位结果，输出加法结果和本次加法进位结果，实现了带进位的两个 4 位二进制数的加法。最后我通过软件仿真和上板实验验证了程序的正确性。

## 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. 理解与实现一位全加器和四位全加器
4. 掌握激励文件的编写，使用功能仿真验证程序正确性
5. 上板验证

## 3 实验原理与功能实现

### 3.1 一位全加器

一位全加器输入为两个一位二进制数  $a$ ， $b$  和上一部分的进位结果  $ci$ ，输出为加法结果  $s$  和进位位  $co$ ，通过真值表可以写出输入端和输出端的逻辑关系如式 1 所示，其中  $\oplus$  表示异或。

$$\begin{aligned} s &= a \oplus b \oplus ci \\ co &= (a \wedge b) \vee (a \wedge ci) \vee (b \wedge ci) \end{aligned} \quad (1)$$

得出输入和输出断后就可以开始一位全加器模块的代码编写。由于  $s$  和  $co$  的真值逻辑比较复杂，且 Verilog 自带的逻辑单元只有 and,or,xor 这些简单操作，所以我们加入了中间值  $s1, c1, c2, c3$  以辅助计算，最后代码如图 1 所示，完整代码见源文件。

### 3.2 四位全加器

四位全加器即为四个一位全加器的串联，前一个一位全加器的  $co$  端接入下一个一位全加器的  $ci$  端，而第一个全加器的  $ci$  端和最后一个全加器的  $co$  端分别为四位全加器的  $ci$  端和  $co$  端，同时还要设置三个中间变量  $[2:0] ct$  来传递中间结果，代码实现如图 1 所示。

### 3.3 仿真验证

编写激励文件代码如图 2 所示，进行仿真验证。在仿真中检验了多组数据的计算，仿真结果如图 3 所示，可见仿真结果正确，我们完成了四位全加器的视线。

```

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

```

```

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 1: 一位和四位全加器的代码实现

```

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 2: 激励文件代码

## 4 上板验证

在上板验证中用实验板上的 8 个 Switch 对应二组 4 位二进制输入，4 个 LED 发光二极管对应输出，并用 2 个七段数码管显示运行结果。我们先按实验指导书上的代

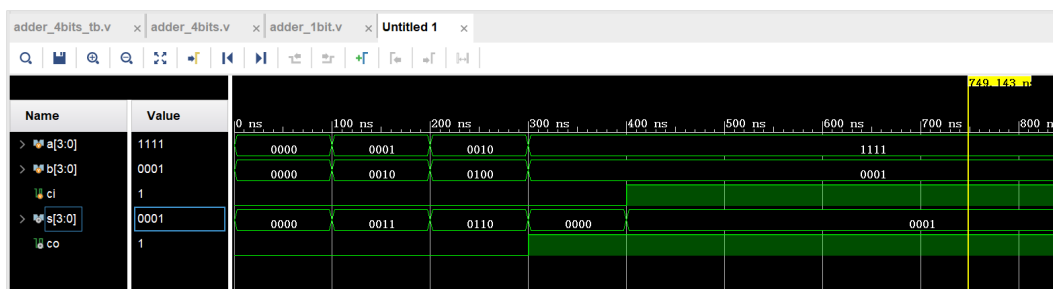


图 3: 仿真验证

码编写 Top.v 文件，再加载 display 核这个七段数码管 SEGMENT 和 LED 发光二极管显示模块，最后添加约束文件，点击 Generate Bitstream，系统将自动完成综合、实现、并生成 FPGA 配置文件，连接好实验板的电源和 JTAG 下载线，然后打开开关，烧写 bit 文件。代码上传完成后，拨动开关，观察到 LED 和数码管如图 4所示，可见上板验证结果正确。

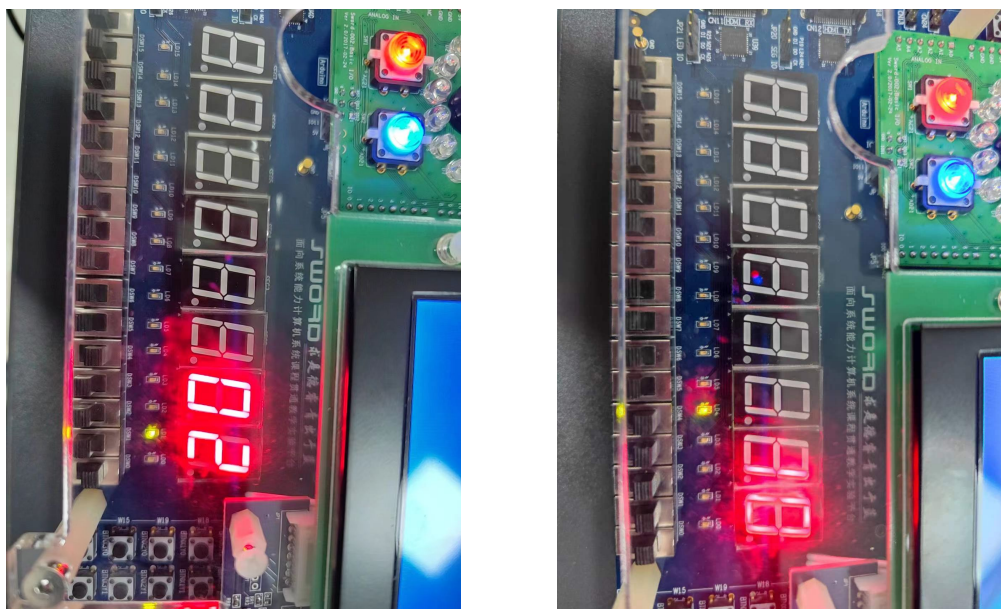


图 4: 上板验证

## 5 总结与思考

本实验的目标是实现一个带进位的四位全加器，如果直接按照四位全加器的输入输出真值表，想要写出其中的逻辑关系是几乎不可能的。我们通过分解任务，将四位全加器化为了四个一位全加器的任务，这样将真值表转化成逻辑关系的任务量是可以接受的，然后再将他们连接，实现了四位全加器。

这样的思路十分的巧妙，也让我学到了许多，将要实现的功能元件划分为更细小的模块，这样每个模块都能更简单的实现，最后将他们连接在一起，相信这样化繁为简的

技巧在以后的实验中还会经常用到。

通过这次实验,我也对 Verilog 和 Vivado 有了更深刻的了解,尤其是其中的 module 的运行逻辑,还有 wire 和 reg 类型的区别,为以后的实验奠定了基础。

## 6 致谢

感谢老师课堂上对于相关内容的教学。

感谢助教课堂上对于我的疑问的解答。

感谢老师与助教提供的实验手册对于实验的指导。