

Vivado 下按键检测实验

1 文档简介

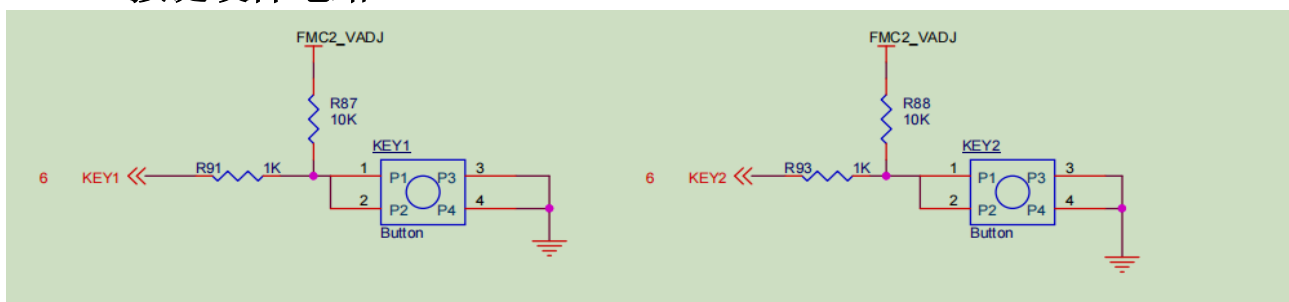
通过按键检测实验，检测开发板的按键功能是否正常，了解硬件描述语言和 FPGA 的具体关系，学习 Vivado RTL ANALYSIS 的使用。

2 实验环境

- Windows 7 SP1 64 位
- Vivado (vivado2017.4)
- 黑金 FPGA 开发板 (AXKU040 开发板)

3 实验原理

3.1 按键硬件电路

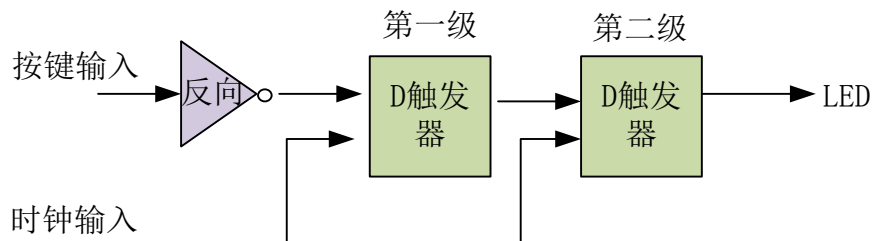


AXKU040 开发板按键部分电路

从图中可以看到，电路的按键松开时是高电平，按下时是低电平。

3.2 程序设计

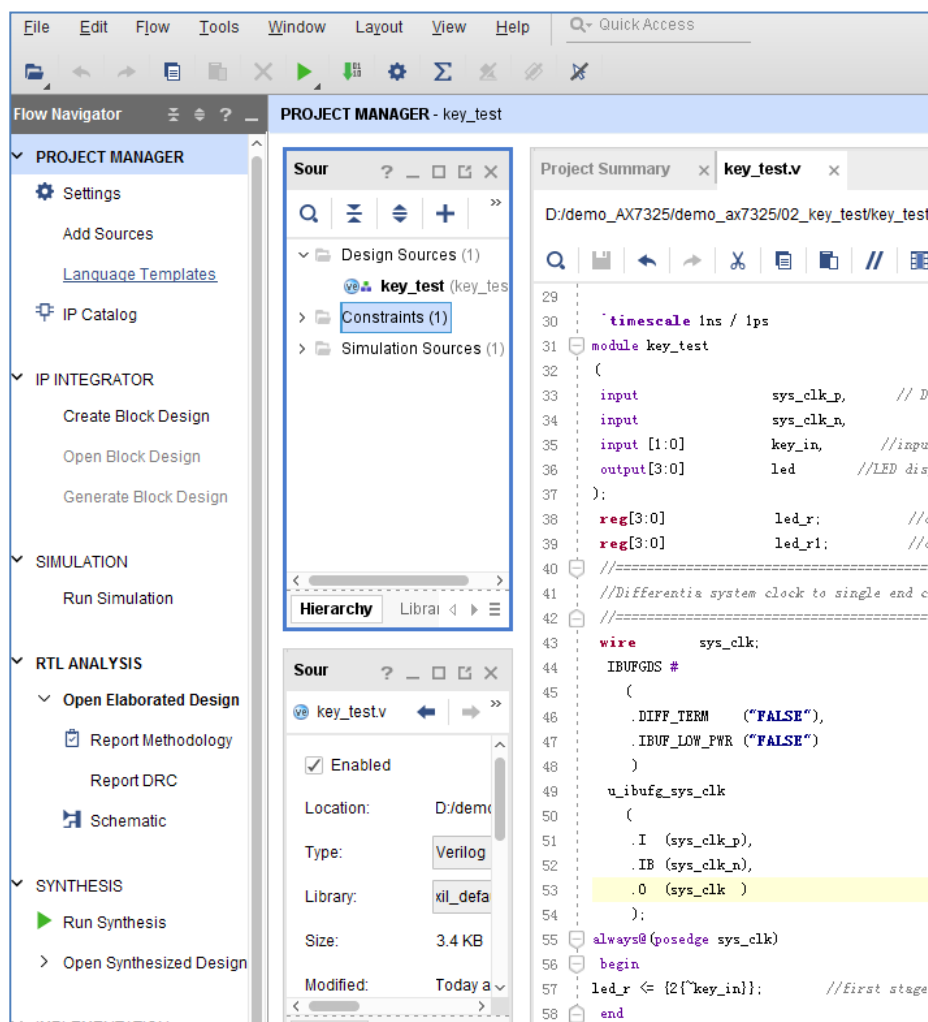
这个程序没有设计的很复杂，通过简单的硬件描述语言看透硬件描述语言和 FPGA 硬件的联系。首先我们将按键输入经过一个非门后再经过 2 组 D 触发器。一个经过 D 触发器的信号，会在 D 触发器时钟上升沿锁存然后再送到输出。



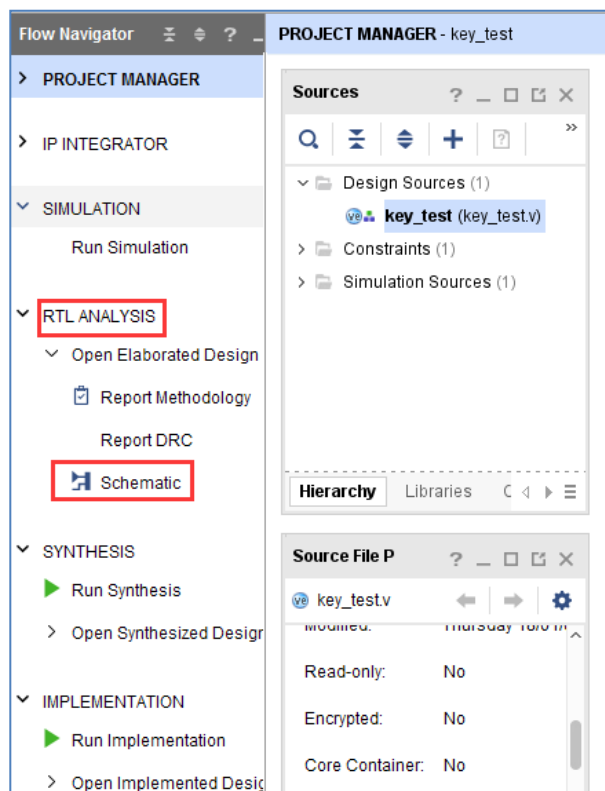
在进行硬件描述语言编码之前，我们已经把硬件构建完成，这是一个正常的开发流程。有了硬件设计思路无论是通过画图还是通过 Verilog HDL、VHDL 都能完成设计，根据设计的复杂程序和对某种语言的熟悉程序来选择工具。

4 工程分析

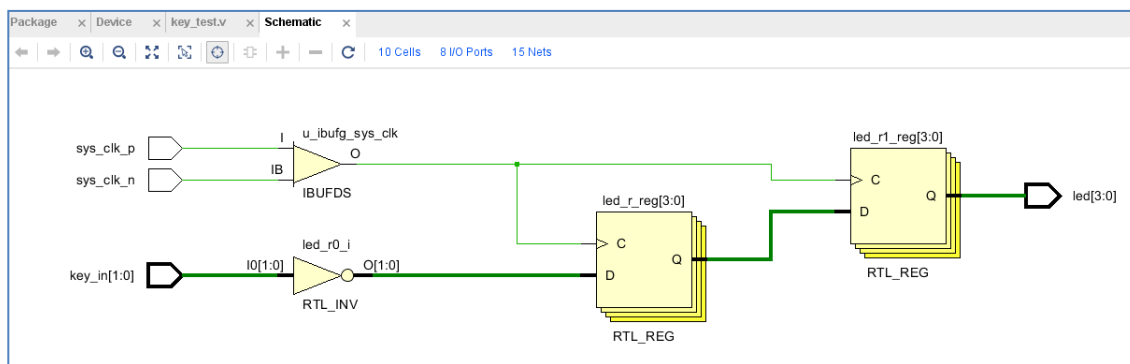
- (1) 首先建立按键的测试工程，添加 verilog 测试代码，完成编译分配管脚等流程。



- (2) 使用 RTL ANALYSIS 工具查看设计



(3) 分析 RTL 图，可以看出第一级 D 触发器经过取反后输入，第二级直接输入，和预期设计一致。



5 实验现象

程序下载到开发板以后，ku040开发板"LED1"、"LED2"、"LED3"、"LED4"都处于熄灭状态，按键“KEY1”按下“LED1和“LED3”亮，LED2和“LED4”灭；按键“KEY2”按下“LED2和“LED4”亮，“LED1和“LED3”灭。

6 附录

key_test.v(verilog 代码)

```
`timescale 1ns / 1ps
```

```

module key_test
(
    input          sys_clk_p,          // Differentia system clock 200Mhz input on board
    input          sys_clk_n,
    input [1:0]    key_in,             //input four key signal,when the keydown,the value is 0
    output [3:0]   led                 //LED display ,when the siganl high,LED lighten
);
    reg [3:0]      led_r;               //define the first stage register ,generate four
    D Flip-flop
    reg [3:0]      led_r1;             //define the second stage register ,generate four
    D Flip-flop
    //=====
    //Differentia system clock to single end clock
    //=====
    wire          sys_clk;
    IBUFGDS #
    (
        .DIFF_TERM    ("FALSE"),
        .IBUF_LOW_PWR ("FALSE")
    )
    u_ibufg_sys_clk
    (
        .I  (sys_clk_p),
        .IB (sys_clk_n),
        .O  (sys_clk)
    );
    always@(posedge sys_clk)
    begin
        led_r <= {2{~key_in}};         //first stage latched data
    end

    always@(posedge sys_clk)
    begin
        led_r1 <= led_r;               //second stage latched data
    end

    assign led = led_r1;
endmodule

```

疑点说明:

led_r <= {2{~key_in}}; //对按键进行位取反是因为按键是低电平有效，led值为高电平被点亮。
 这里输入的按键值决定了led的变化，{}这里是位拼接的用法 2这个数这里
 可以理解为填充次数 举例 key=2'b01 “<=”右边的值就等于 4'b1010