

按键检测实验

2020-12-15

1 文档简介

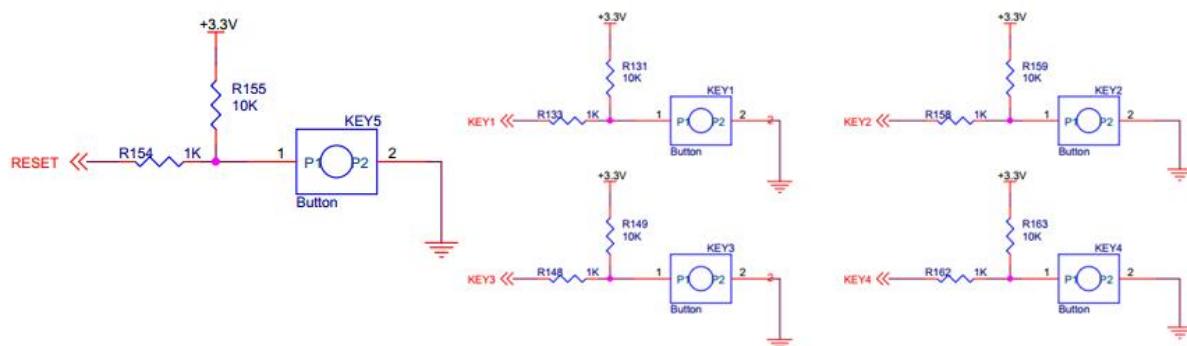
通过按键检测实验，检测开发板的按键功能是否正常，了解硬件描述语言和 FPGA 的具体关系，学习 PDS View RTL Schematic 的使用。

2 实验环境

- Windows 10 64 位
- PDS
- AXP100 开发板

3 实验原理

3.1 按键硬件电路

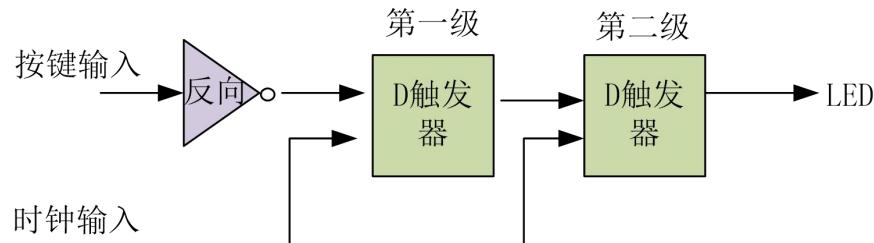


按键部分电路

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

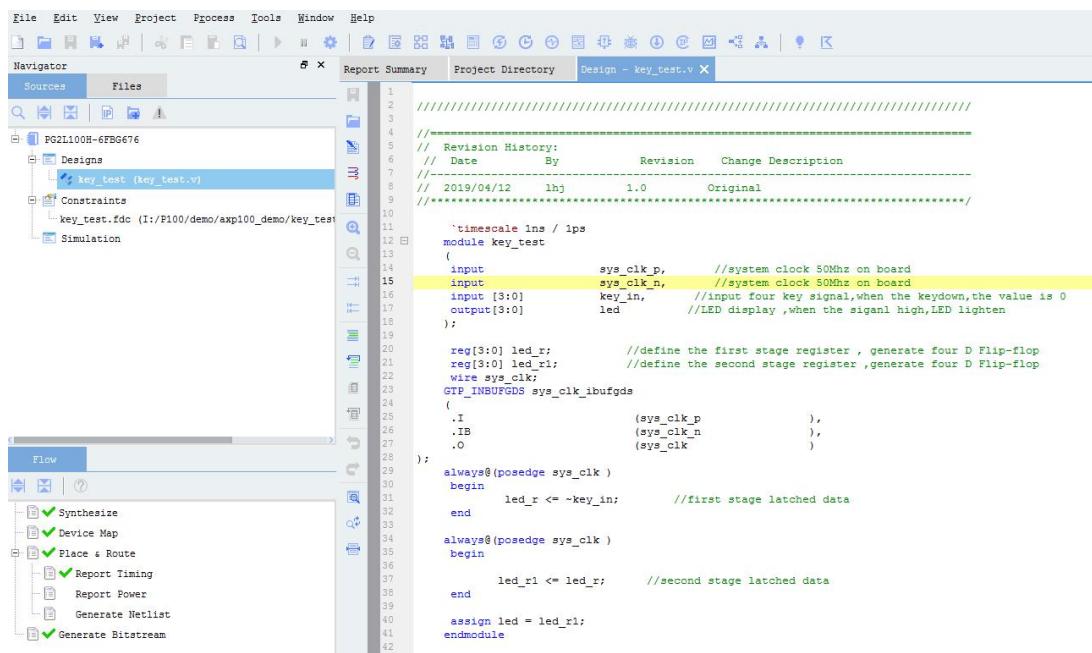
这个程序没有设计的很复杂，通过简单的硬件描述语言看透硬件描述语言和 FPGA 硬件的联系。首先我们将按键输入经过一个非门后再经过 2 组 D 触发器。经过 D 触发器的信号，会在 D 触发器时钟输入的上升沿锁存然后再送到输出。



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，根据设计的复杂程度和对某种语言的熟悉程度来选择工具。

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。

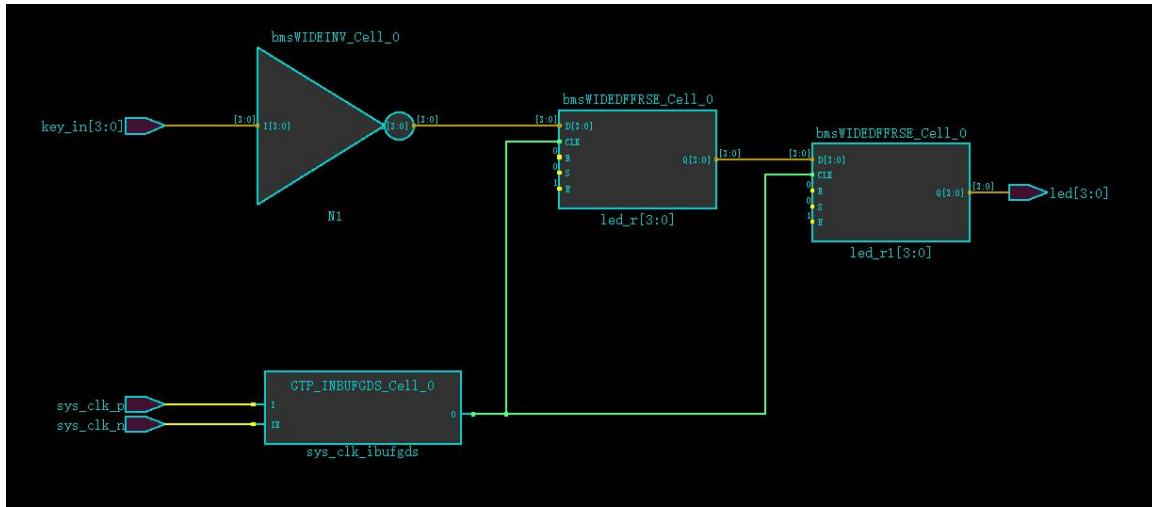


```
1 //////////////////////////////////////////////////////////////////
2 // Revision History:
3 // Date      By      Revision  Change Description
4 // 2019/04/12  lhj    1.0      Original
5 //////////////////////////////////////////////////////////////////
6
7 'timescale 1ns / 1ps
8
9 module key_test
10
11   input  sys_clk_p,      //system clock 50Mhz on board
12   input  sys_clk_n,      //system clock 50Mhz on board
13   input [3:0] key_in,    //input four key signal,when the keydown,the value is 0
14   output[3:0] led;      //LED display ,when the signal high,LED lighten
15
16   reg[3:0] led_r;       //define the first stage register , generate four D Flip-flop
17   reg[3:0] led_r1;      //define the second stage register ,generate four D Flip-flop
18   wire sys_clk;
19   GTP_INBUFGDS sys_clk_ibufgds
20   (
21     .I      (sys_clk_p),
22     .IB     (sys_clk_n),
23     .O      (sys_clk)
24   );
25
26   always@(posedge sys_clk)
27   begin
28     led_r <= ~key_in;      //first stage latched data
29   end
30
31   always@(posedge sys_clk)
32   begin
33     led_r1 <= led_r;      //second stage latched data
34   end
35
36   assign led = led_r1;
37
38
39
40
41
42
```

(2) 使用 View RTL Schematic 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, "LED1"、"LED2"、"LED3"、"LED4"都处于亮状态, 按键 "KEY1" 按下 "LED1" 灭, 按键 "KEY2" 按下 "LED2" 灭, 按键 "KEY3" 按下 "LED3" 灭, 按键 "KEY4" 按下 "LED4" 灭。

6 附录

key_test.v(verilog 代码)

```
//////////  
=====  
// Revision History:  
// Date      By      Revision  Change Description  
//-----  
// 2019/04/12  lhj      1.0      Original  
//*****
```

```
`timescale 1ns / 1ps  
module key_test  
(  
  input  sys_clk_p,  //system clock 50Mhz on board  
  input  sys_clk_n,  //system clock 50Mhz on board
```

```
input [3:0]      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
wire sys_clk;
GTP_INBUFGDS sys_clk_ibufgds
(
    .I          (sys_clk_p      ),
    .IB         (sys_clk_n      ),
    .O          (sys_clk      )
);
always@(posedge sys_clk )
begin
    led_r <= ~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
```