

计算机系统结构实验

实验 1 报告

FPGA 基础实验
LED Flow Water Light

李子龙

518070910095

2021 年 5 月 19 日

目录

1 实验目的

1. 熟悉 Xilinx 逻辑设计工具 Vivado 的基本操作
2. 掌握使用 Verilog HDL 进行简单的逻辑设计
3. 使用功能仿真与 I/O Planing 添加管脚约束
4. 生成 Bitstream 文件并上板验证流水灯

2 实验原理与实现

本实验通过需要实现 8 阶流水灯。首先使用 `cnt_reg` 寄存器记录时钟周期数目：

```

1  reg [23:0] cnt_reg;
2  always @ (posedge clock)
3      begin
4          if (reset)
5              cnt_reg <= 0;
6          else
7              cnt_reg <= cnt_reg + 1;
8      end

```

接着，对于 LED 灯所对应的循环，如果是复位信号，就会改为 01H，即仅点亮最低位的灯。当计数器循环满时，如果 LED 是 80H，即仅最高位灯亮，则改为 01H 仅最低位亮以循环；否则就直接向左移位，以仅点亮下一个高位的灯。

```

1  reg [7:0] light_reg;
2  always @(posedge clock)
3      begin
4          if (reset)
5              light_reg <= 8'h01;
6          else if (cnt_reg == 24'hffffff)
7              begin
8                  if (light_reg==8'h80)
9                      light_reg <= 8'h01;
10                 else
11                     light_reg <= light_reg << 1;
12             end
13      end

```

当然在上板子后，对于该硬件 xc7k325tffg676-2 需要修改 `reset` 为取反信号，最终实现请见附录 ??。

A 流水灯设计代码实现

Listing 1: *flowing_ight.v*

```

1  `timescale 1ns / 1ps
2  ///////////////////////////////////////////////////////////////////
3  // Company: SJTU CS
4  // Engineer: Zilong Li
5  //
6  // Create Date: 2021/05/12 09:10:24
7  // Design Name:
8  // Module Name: flowing_light
9  // Project Name:
10 // Target Devices:
11 // Tool Versions:
12 // Description:
13 //
14 // Dependencies:
15 //
16 // Revision:
17 // Revision 0.01 - File Created
18 // Additional Comments:
19 //
20 ///////////////////////////////////////////////////////////////////
21
22 module flowing_light (
23     input clock_p,
24     input clock_n,
25     input reset,
26     output [7:0] led
27 );
28
29     reg [23:0] cnt_reg;
30     reg [7:0] light_reg;
31
32     IBUFGDS IBUFGDS_inst (
33         .O(CLK_i),
34         .I(clock_p),
35         .IB(clock_n)
36     );
37
38     always @ (posedge CLK_i)
39     begin
40         if (!reset) // 板子电平取反
41             cnt_reg <= 0;
42         else
43             cnt_reg <= cnt_reg + 1;
44     end
45     always @(posedge CLK_i)
46     begin
47         if (!reset)
48             light_reg <= 8'h01;
49         else if (cnt_reg == 24'hffffff)
50             begin
51                 if (light_reg==8'h80)
52                     light_reg <= 8'h01;
53                 else
54                     light_reg <= light_reg << 1;
55             end
56     end
57     assign led = light_reg;
58 endmodule

```

