《FPGA 应用实验》实验报告

实验名称： 利用 8 个发光二极管（LED）形成流水灯显示

姓名：肖佳伟 学号：515030910023 班级：F1503001

一. 实验平台：

采用 Xilinx 公司的 FPGA 集成开发环境 Xilinx ISE Design Suite 10.1 sp3，实验开发板为 Xilinx Spartan-3E FPGA Starter Kit。

二. 实验设计要求：

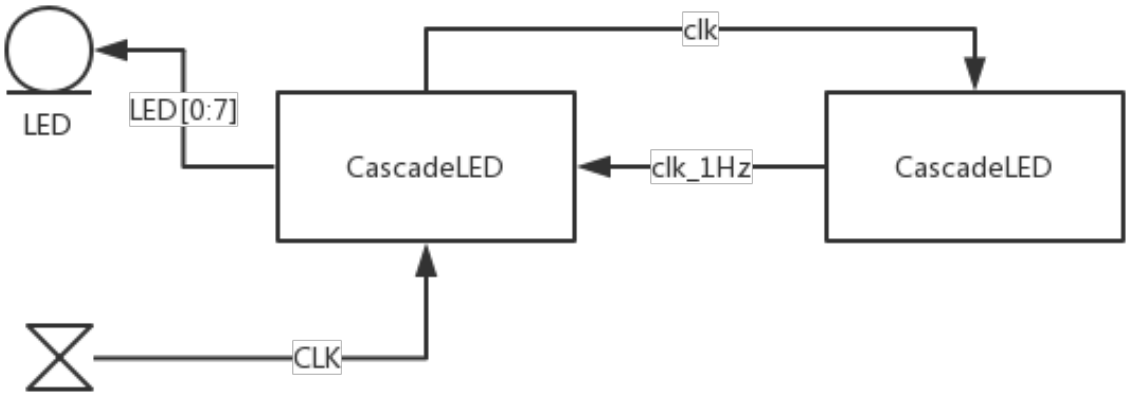
在 Spartan-3E FPGA Starter Kit Board 上有 8 个发光二极管（LED7 ~ LED0）。

使用开发板的全局时钟信号 CLK_50MHz，管脚为 P = C9。产生 1Hz 的秒脉冲，每秒钟点亮一个 LED。

开始 8 个 LED 都为关闭状态（缺省值为：LEDOut = 8'b0000_0000）；即：

i. LEDOut = 8'b0000_0000;
ii. LEDOut = 8'b0000_0001;
iii. LEDOut = 8'b0000_0011;
iv. LEDOut = 8'b0000_0111;
v. LEDOut = 8'b0000_1111;
vi. LEDOut = 8'b0001_1111;
vii. LEDOut = 8'b0011_1111;
viii. LEDOut = 8'b0111_1111;
ix. LEDOut = 8'b1111_1111;
x. 不断重复 i ~ ix。

三. 模块设计框图：



四. 实验原理：

通过有穷自动机控制 LED 二极管，由每一秒由其前一秒的状态决定当前状态，从而使得

LED 二极管的输出为八个状态的有限循环。

五. Verilog 模块设计：

i. CascadeLED.v

```
`timescale 1ns / 1ps
`include "clock_div.v"
module CascadeLED (
    output reg [7:0] LED,
    input CLK
    );

    reg [7:0] state;
    wire p_clk1Hz;

    parameter S0 = 8'b0000_0001;
    parameter S1 = 8'b0000_0010;
    parameter S2 = 8'b0000_0100;
    parameter S3 = 8'b0000_1000;
    parameter S4 = 8'b0001_0000;
    parameter S5 = 8'b0010_0000;
    parameter S6 = 8'b0100_0000;
    parameter S7 = 8'b1000_0000;

    // 1Hz 时钟信号产生模块调用实例
    clock_div m_clkgen( .clk_1Hz(p_clk1Hz), .clk(CLK) );

    always @ ( negedge p_clk1Hz ) begin
        case (state)
            S0: begin state <= S1; LED <= 8'b0000_0001; end
            S1: begin state <= S2; LED <= 8'b0000_0011; end
            S2: begin state <= S3; LED <= 8'b0000_0111; end
            S3: begin state <= S4; LED <= 8'b0000_1111; end
            S4: begin state <= S5; LED <= 8'b0001_1111; end
            S5: begin state <= S6; LED <= 8'b0011_1111; end
            S6: begin state <= S7; LED <= 8'b0111_1111; end
            S7: begin state <= S0; LED <= 8'b1111_1111; end
            default: begin state <= S0; LED <= 8'b0000_0000; end
        endcase
    end

endmodule // CascadeLED
```

ii. clock_div.v

```
`timescale 1ns / 1ps
module clock_div(
    output reg clk_1Hz,      // 1Hz时钟输出信号
    input clk                // 系统时钟输入信号
    );

    // 25'h17D_7840 = 1_0111_1101_0111_1000_0100_0000(bin) (25bits)
    parameter PULSESCOUNT = 25'h17D_7840;
    parameter RESETZERO = 25'h0;

    // 计数器, 25 bits (1_0111_1101_0111_1000_0100_0000(bin))
    // 用于对系统时钟脉冲进行计数, 以产生 1Hz 输出时钟信号
    reg [24:0] counter;

    // 由 clock 信号的上升沿触发
    always @( posedge clk ) begin
        // 25'h17D_7840 个系统时钟脉冲等于 1/2(s)
        if ( counter < PULSESCOUNT )
            counter <= counter + 1'b1;
```

```
        else begin
            clk_1Hz <= ~clk_1Hz;
            counter <= RESETZERO;
        end
    end

endmodule
```

iii. CascadeLED.ucf

```
NET "CLK" LOC = "C9" | IOSTANDARD = LVCMOS33 ;

NET "LED<7>" LOC = "F9" | IOSTANDARD = LVTTL | SLEW = SLOW | DRIVE = 8 ;
NET "LED<6>" LOC = "E9" | IOSTANDARD = LVTTL | SLEW = SLOW | DRIVE = 8 ;
NET "LED<5>" LOC = "D11" | IOSTANDARD = LVTTL | SLEW = SLOW | DRIVE = 8 ;
NET "LED<4>" LOC = "C11" | IOSTANDARD = LVTTL | SLEW = SLOW | DRIVE = 8 ;
NET "LED<3>" LOC = "F11" | IOSTANDARD = LVTTL | SLEW = SLOW | DRIVE = 8 ;
NET "LED<2>" LOC = "E11" | IOSTANDARD = LVTTL | SLEW = SLOW | DRIVE = 8 ;
NET "LED<1>" LOC = "E12" | IOSTANDARD = LVTTL | SLEW = SLOW | DRIVE = 8 ;
NET "LED<0>" LOC = "F12" | IOSTANDARD = LVTTL | SLEW = SLOW | DRIVE = 8 ;
```

iv. tb_cascadeled.v

```
`timescale 1ns / 1ps
module tb_cascadeled;

    // Inputs
    reg CLK;

    // Outputs
    wire [7:0] LED;

    // Instantiate the Unit Under Test (UUT)
    CascadeLED uut (
        .LED(LED),
        .CLK(CLK)
    );

    initial begin
        // Initialize Inputs
        CLK = 0;

        // Wait 100 ns for global reset to finish
        #100;

        // Add stimulus here
        forever begin
            CLK = ~CLK;
            #10 ;
        end

    end

endmodule
```

六. 试验仿真结果和分析：