2022《FPGA 应用实验》实验报告

实验编号：lab 1　　实验时间：2022.03.12

实验名称：利用 8 个发光二极管（LED）形成流水灯显示

班级：F1903604　学号：519021910917　姓名：费扬

1、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。

2、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 1 Hz 的秒脉冲，每秒钟点亮一个 LED。

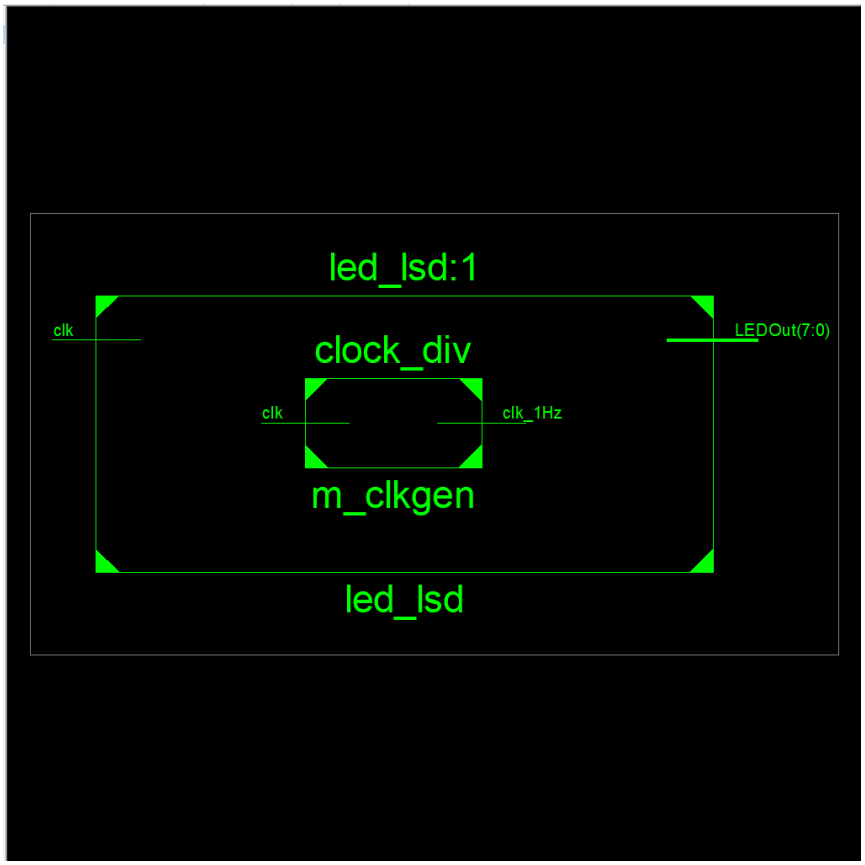
使用滑动开关 SW0，管脚 P=L13，作为复位键。

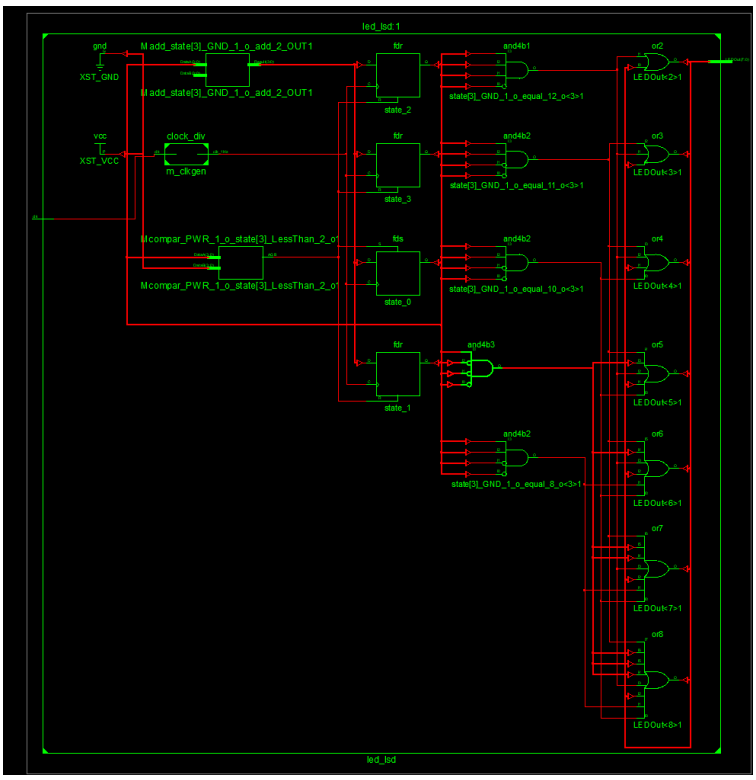
复位后，8 个 LED 都为关闭状态（缺省值为：LEDOut = 8’b0000_0000）；

即：

（0）LEDOut = 8’b0000_0000;
（1）LED Out = 8’b0000_0001;
（2）LED Out = 8’b0000_0011;
（3）LED Out = 8’b0000_0111;
（4）LED Out = 8’b0000_1111;
（5）LED Out = 8’b0001_1111;
（6）LED Out = 8’b0011_1111;
（7）LED Out = 8’b0111_1111;
（8）LED Out = 8’b1111_1111;
（9）不断重复（0）～（8）

3、模块设计框图





4、实验原理：

1. LED 原理

Spartan-3e FPGA Starter Kit 板有 8 个独立的表面安装 led 位于滑动开关上方，如下图所

示。led 标记为 LED0 到 LED7。LED7 是最左边的 LED, LED0 是最右边的 LED。

Figure 2-10: **Eight Discrete LEDs**

操作：每个 LED 的一端连接到地，另一端通过 390 欧限流电阻连接到 Spartan-3e 设备上的引脚。为了点亮单个 LED，需驱动相关的 FPGA 控制信号为高。

UCF 位置约束：下图提供了四个按钮开关的 UCF 约束，包括 I/O 引脚分配，使用的 I/O 标准，输出回转率和输出驱动电流。

```
NET "clk" LOC = "C9" | IOSTANDARD = LVTTL | SLEW = SLOW | DRIVE = 8 ;
NET "LEDOut<7>" LOC = "F9" | IOSTANDARD = LVTTL | SLEW = SLOW | DRIVE = 8;
NET "LEDOut<6>" LOC = "E9" | IOSTANDARD = LVTTL | SLEW = SLOW | DRIVE = 8;
NET "LEDOut<5>" LOC = "D11" | IOSTANDARD = LVTTL | SLEW = SLOW | DRIVE = 8;
NET "LEDOut<4>" LOC = "C11" | IOSTANDARD = LVTTL | SLEW = SLOW | DRIVE = 8;
NET "LEDOut<3>" LOC = "F11" | IOSTANDARD = LVTTL | SLEW = SLOW | DRIVE = 8;
NET "LEDOut<2>" LOC = "E11" | IOSTANDARD = LVTTL | SLEW = SLOW | DRIVE = 8;
NET "LEDOut<1>" LOC = "E12" | IOSTANDARD = LVTTL | SLEW = SLOW | DRIVE = 8;
NET "LEDOut<0>" LOC = "F12" | IOSTANDARD = LVTTL | SLEW = SLOW | DRIVE = 8;
```

2. 50MHz 时钟

Spartan-3E FPGA Starter Kit 单板支持三个主时钟输入源，它们都位于 Xilinx 标识的下方，靠近 spartan-3e 标识。该板包括一个板载 50 MHz 时钟振荡器。时钟可以通过 SMA 风格的连接器提供。FPGA 也可以通过 SMA 风格的连接器产生时钟信号或其他高速信号。可选安装一个单独的 8 针 DIP 风格时钟振荡器在提供的插座。

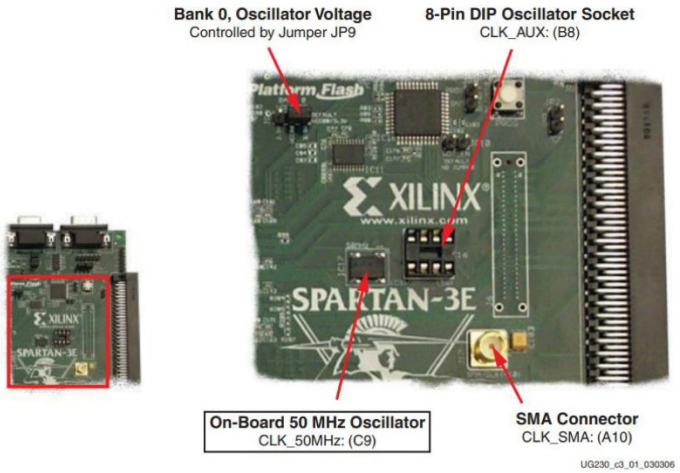


Figure 3-1: **Available Clock Inputs**

5、Verilog 模块设计

//顶层模块

//led_lsd.v

```
module led_lsd(
    output [7:0] LEDOut,
    input clk
    );
 reg [7:0] led_connect;
 reg [3:0] state;
 wire clk_1;

 clock_div m_clkgen(.clk_1Hz(clk_1),.clk(clk));

 always @(posedge clk_1) begin
if (state > 8) state <= 1; //下一个状态为1，1号灯亮
else state <= state + 1;
 end

 always@(*)
case (state)
1: led_connect<=8'b0000_0001;
2: led_connect<=8'b0000_0011;
3: led_connect<=8'b0000_0111;
4: led_connect<=8'b0000_1111;
5: led_connect<=8'b0001_1111;
6: led_connect<=8'b0011_1111;
7: led_connect<=8'b0111_1111;
8: led_connect<=8'b1111_1111;
default: led_connect<=8'b0000_0000;
endcase

assign LEDOut = led_connect;

endmodule
```

//时钟分频

//clock_div.v

```
module clock_div( output reg clk_1Hz,
                      input  clk
                    );
parameter PULSESCOUNT = 18'h3_D090, // = 11_1101_0000_1001_0000(bin) (16bits)
                  RESETZERO = 18'h0;

    reg [17:0] counter;


 always @(posedge clk) begin
  if ( counter < PULSESCOUNT )
   counter <= counter + 1'b1;
  else begin
   clk_1Hz <= ~clk_1Hz;
   counter <= RESETZERO;
  end
 end
endmodule
```

//引脚约束

//led_lsd.ucf

```
NET "clock" LOC="C9" | IOSTANDARD = LVCMOS33;
NET "LEDOut<7>" LOC = "F9" | IOSTANDARD = LVTTL | SLEW = SLOW | DRIVE = 8;
NET "LEDOut<6>" LOC = "E9" | IOSTANDARD = LVTTL | SLEW = SLOW | DRIVE = 8;
NET "LEDOut<5>" LOC = "D11" | IOSTANDARD = LVTTL | SLEW = SLOW | DRIVE = 8;
NET "LEDOut<4>" LOC = "C11" | IOSTANDARD = LVTTL | SLEW = SLOW | DRIVE = 8;
NET "LEDOut<3>" LOC = "F11" | IOSTANDARD = LVTTL | SLEW = SLOW | DRIVE = 8;
NET "LEDOut<2>" LOC = "E11" | IOSTANDARD = LVTTL | SLEW = SLOW | DRIVE = 8;
NET "LEDOut<1>" LOC = "E12" | IOSTANDARD = LVTTL | SLEW = SLOW | DRIVE = 8;
NET "LEDOut<0>" LOC = "F12" | IOSTANDARD = LVTTL | SLEW = SLOW | DRIVE = 8;
```

//test_bench

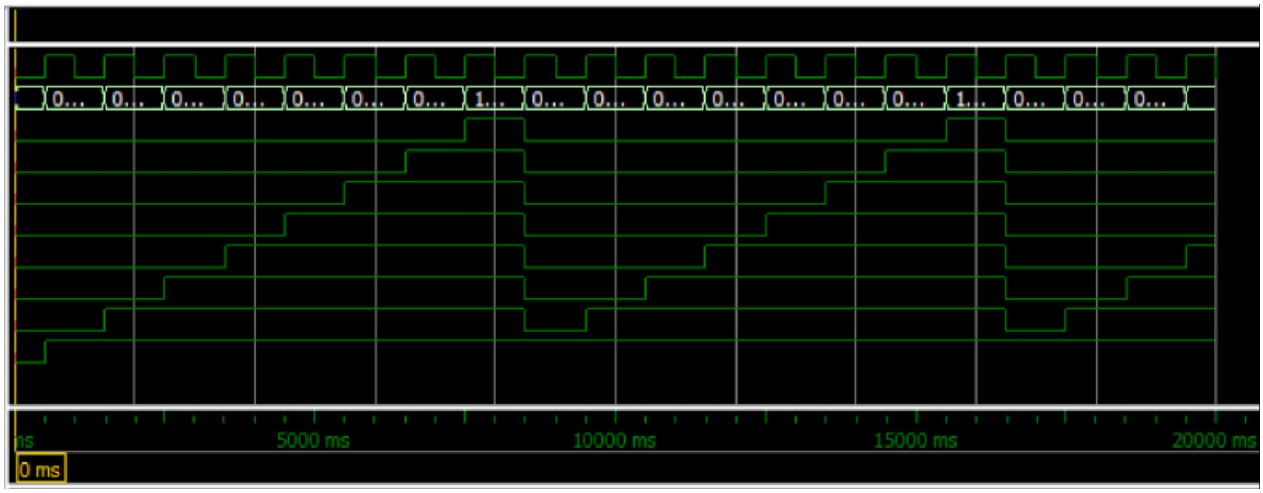
//led_lsd_tb.v

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

```
// Inputs
reg clk;

// Outputs
wire [7:0] p_led;

// Instantiate the Unit Under Test (UUT)
led_lsd uut(
.LEDOut(p_led),
.clk(clk)
);

initial begin
// Initialize Inputs
clk = 0;
forever #10 clk=~clk;
end

initial
#3000 $stop;

initial
$monitor("time=%d,ledout=%b",$time,p_led);

endmodule
```

6、试验仿真结果和分析

如图：仿真中，LED 正确显示流水灯：




```
# time=                    0,ledout=00000000
# time=                    5,ledout=00000001
# time=                   15,ledout=00000011
# time=                   25,ledout=00000111
# time=                   35,ledout=00001111
# time=                   45,ledout=00011111
# time=                   55,ledout=00111111
# time=                   65,ledout=01111111
# time=                   75,ledout=11111111
# time=                   85,ledout=00000001
# time=                   95,ledout=00000011
```