2021《FPGA 应用实验》实验报告

实验编号：　实验一　　　　　　　实验时间：　2021.3.16

实验名称：　利用 8 个发光二极管（LED）形成流水灯显示

班级：　F1803603　　学号：　518021910534　姓名：　尹俊同

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。

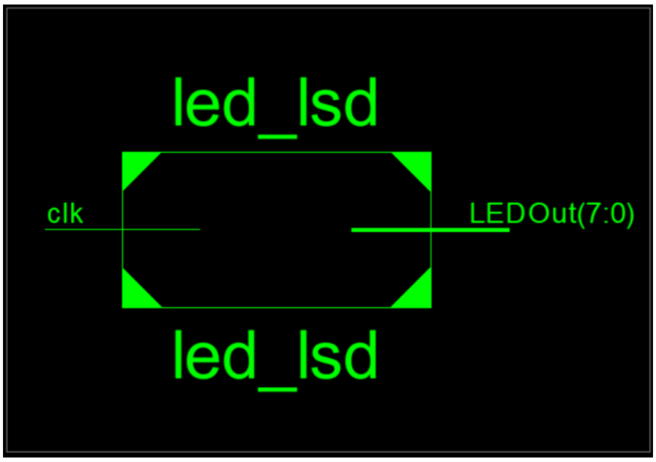
2、实验设计要求：

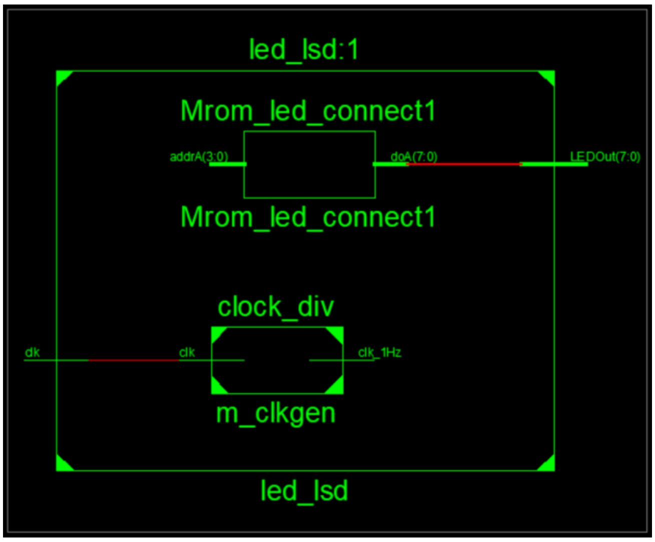
在 Spartan-3E FPGA Starter Kit Board 上有 8 个发光二极管（LED7 ～ LED0）。
使用开发板的全局时钟信号 CLK_50MHz，管脚为 P = C9。产生 1 Hz 的秒脉冲，每秒钟点亮一个 LED。

开始 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;
（6）LED Out = 8’b0001_1111;
（7）LED Out = 8’b0011_1111;
（8）LED Out = 8’b0111_1111;
（9）LED Out = 8’b1111_1111;
（10）不断重复（1）～（9）

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，如图 2-10 所示。led 标记为 LED0 到 LED7。LED7 是最左边的 LED, LED0 是最右边的 LED。

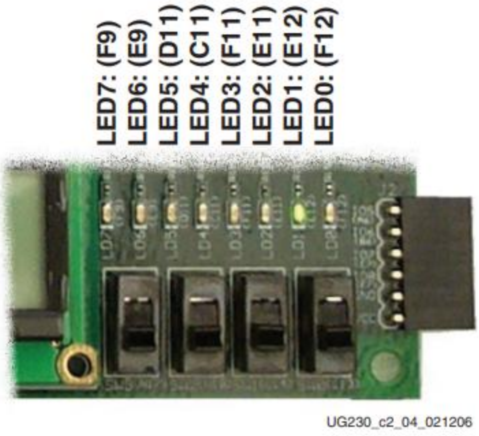
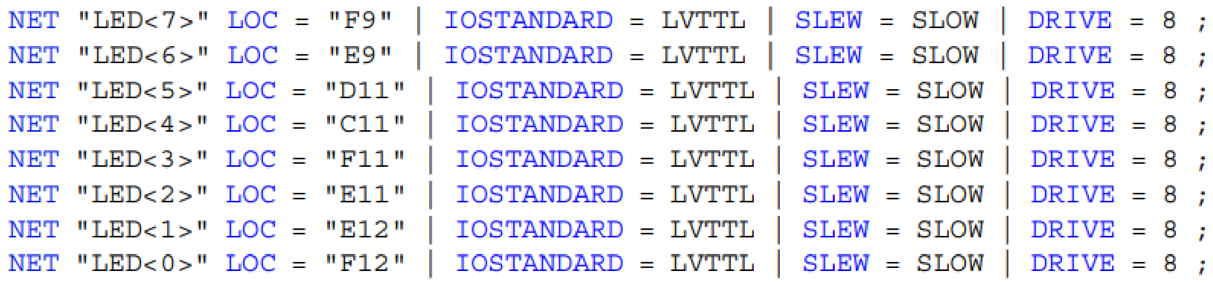


Figure 2-10: **Eight Discrete LEDs**

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：图 2-11 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 "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 ;
```

Figure 2-11: **UCF Constraints for Eight Discrete LEDs**

2、50MHz 时钟

如图 3-1 所示，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。

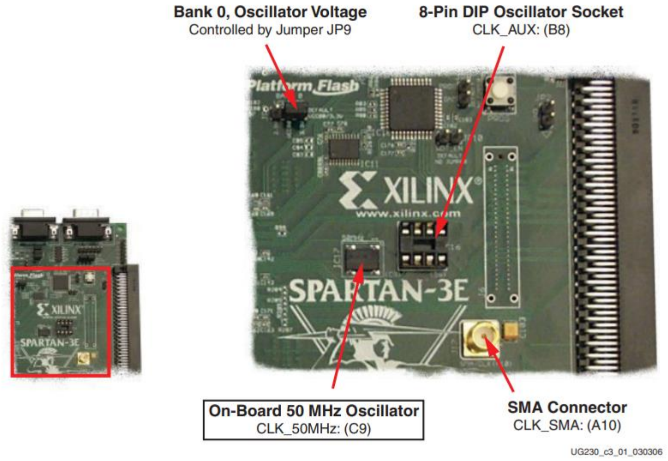
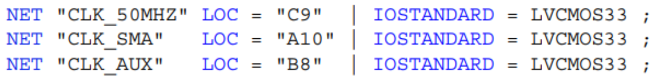


Figure 3-1: **Available Clock Inputs**

50MHz 时钟振荡器的 UCF 约束如下图。



```
NET "CLK_50MHZ" LOC = "C9"  | IOSTANDARD = LVCMOS33 ;
NET "CLK_SMA"   LOC = "A10" | IOSTANDARD = LVCMOS33 ;
NET "CLK_AUX"   LOC = "B8"  | IOSTANDARD = LVCMOS33 ;
```

Figure 3-2: **UCF Location Constraints for Clock Sources**

5、Verilog 模块设计

```
// 顶层模块

//led_lsd.v

`timescale 1ns / 1ps

`include "clock_div.v"


module led_lsd(input clk, output [7:0] LEDOut);
```

```
    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; //下一个状态就是状态1，第一个灯亮
        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


// 时钟分频模块
//clock_div.v
module clock_div(output reg clk_1Hz,
     input clk
     );

    //1Hz的输出信号需要每1/2s翻转一次,1/2s =25,000,000个信号脉冲
     parameter       PULSESCOUNT       =       25'b1_0111_1101_0111_1000_0100_0000,       //
25,000,000=1_0111_1101_0111_1000_0100_0000
            RESETZERO = 25'h0;

     reg [24:0] counter;
```

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

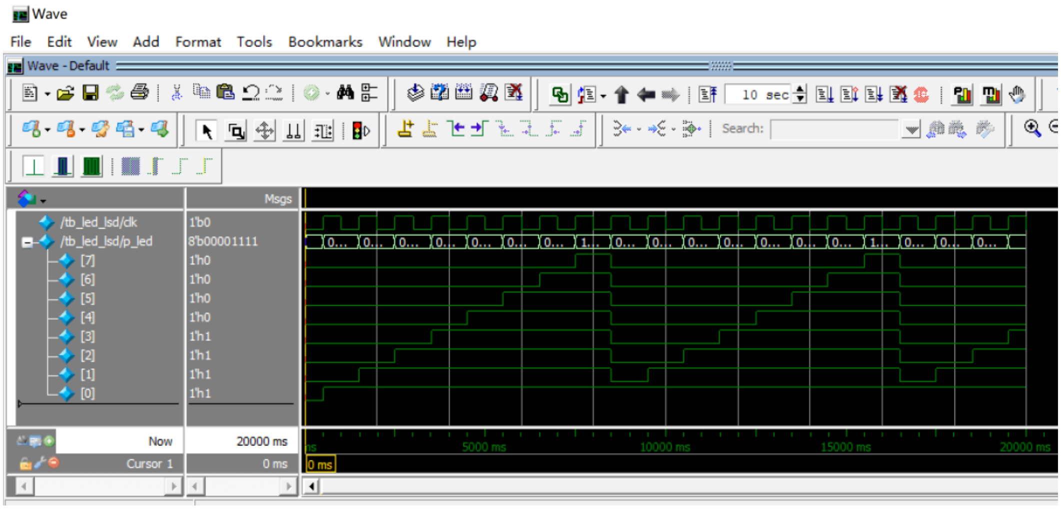
```
// ucf 引脚约束
NET "clk" LOC = "C9" | IOSTANDARD = LVCMOS33 ;
NET "LEDOut<7>" LOC = "F9" | IOSTANDARD = LVTTL | SLEW = SLOW | DRIVE = 8 ;
NET "LEDOut<6>" LOC = "E9" | IOSTANDARD = LVTTL | SLEW = SLOW | DRIVE = 8 ;
NET "LEDOut<5>" LOC = "D11" | IOSTANDARD = LVTTL | SLEW = SLOW | DRIVE = 8 ;
NET "LEDOut<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 ;
```

```
//测试台模块
`include "led_lsd.v"
module tb_led_lsd;
  reg clk;
  wire [7:0]p_led;

  led_lsd    led(.LEDOut(p_led),.clk(clk));

  initial
  begin
    clk=0;
    forever #10 clk=~clk;
  end

  initial
#3000 $stop;

  initial
```

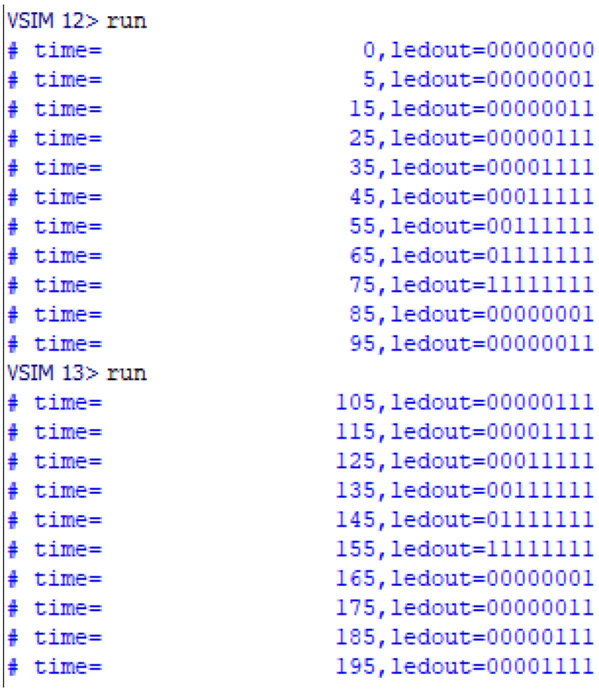
```
$monitor("time=%d,ledout=%b",$time,p_led);

endmodule
```

6、试验仿真结果和分析

如图，随着 50MHz 时钟信号输入，LED 正确显示流水灯。



	Msgs
/tb_led_lsd/clk	1'b0
/tb_led_lsd/p_led	8'b00001111
[7]	1'h0
[6]	1'h0
[5]	1'h0
[4]	1'h0
[3]	1'h1
[2]	1'h1
[1]	1'h1
[0]	1'h1



```
VSIM 12> run
# 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
VSIM 13> run
# time=                  105,ledout=00000111
# time=                  115,ledout=00001111
# time=                  125,ledout=00011111
# time=                  135,ledout=00111111
# time=                  145,ledout=01111111
# time=                  155,ledout=11111111
# time=                  165,ledout=00000001
# time=                  175,ledout=00000011
# time=                  185,ledout=00000111
# time=                  195,ledout=00001111
```