# 2022《FPGA 应用实验》实验报告

实验编号：lab 2　　实验时间：2022.04.01

实验名称：计时模块设计

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：

**使用 8 个发光二极管（LED7~LED0）作为 8 位计数器的输出显示。**

**(1)使用滑动开关 SW3 控制计数器开始/停止:**

a)当 SW3=0 时,停止计数,此时,计数器为 0，8 个 LED 都为关闭状态(缺省值为:LEDOut=8’b00000000);

b)当 SW3=1 时,开始计数;

**(2）使用滑动开关 SW0，管脚 P=L13，作为设置开关，当 SW3=0 时，即，停止计数时，使用 SW0 选择计数方式:**

a)当 SW0=0 时，8 位计数器每秒计数，从 0 到 127，循环反复。

b)当 SW0=1 时，计数器为两个四位计数器;

高四位:0~9，每秒计数，从 0 到 9，循环反复。

低四位:0~9，每 1/10 秒计数，从 0 到 9，循环反复。

要求，计数过程中，不可进行计数方式转换。

**实验要求：**

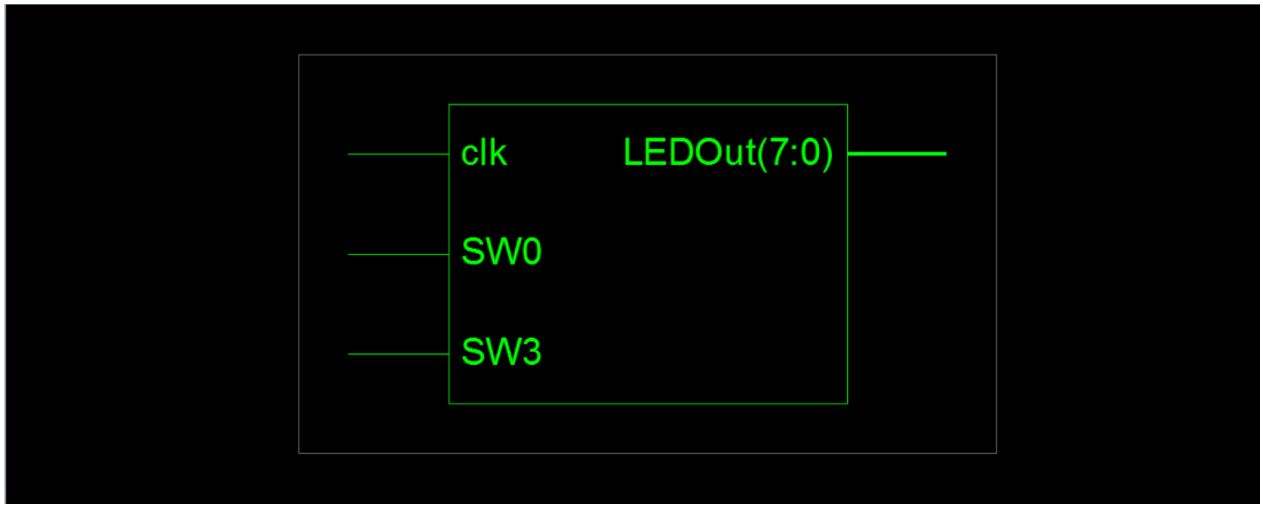
基于 FPGA 集成开发环境 Xilinx lSE 10.1 和仿真工具 ModelSim XE 6.3c:

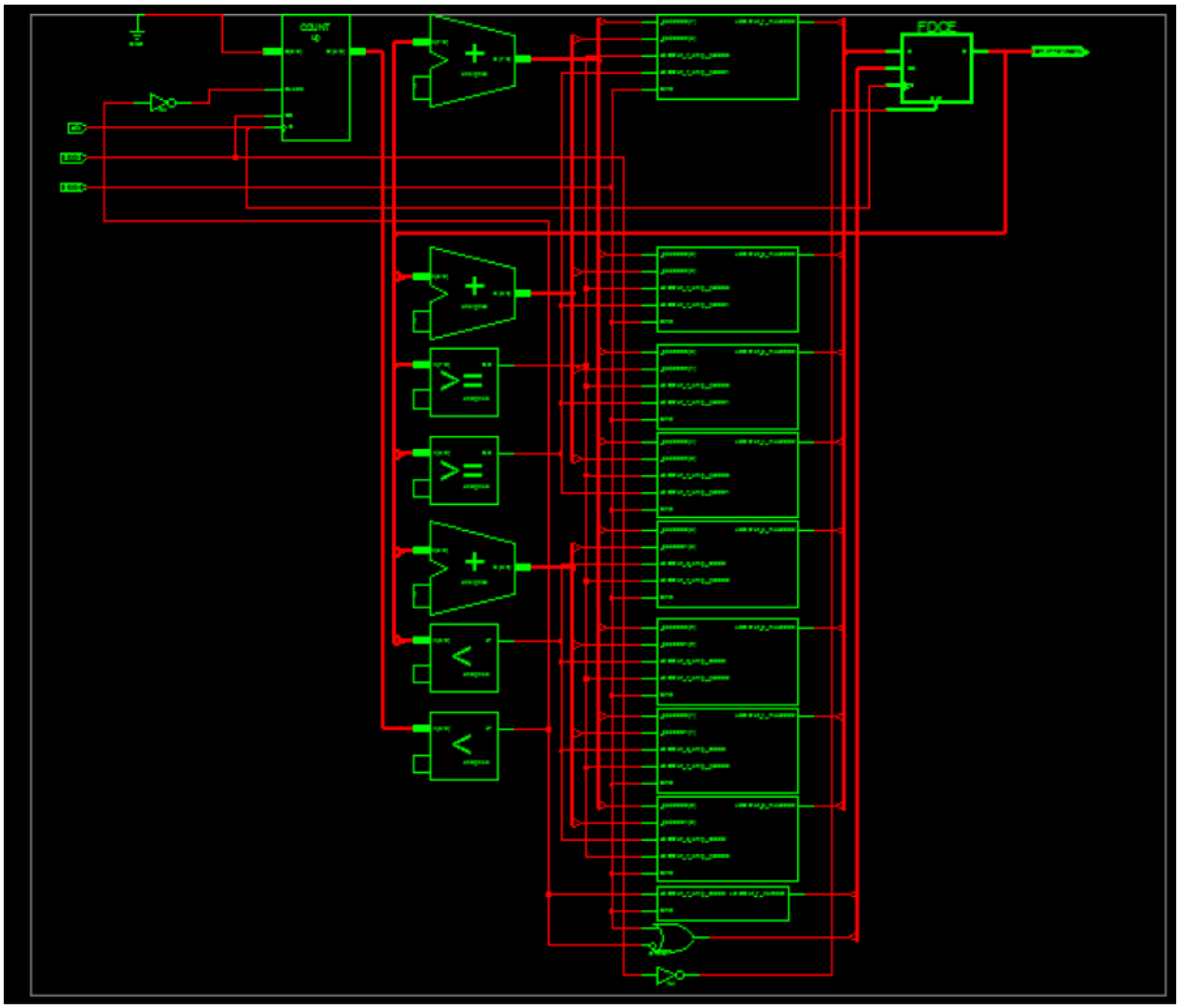
(1) 编写设计模块和测试台模块;

(2) 完成行为(功能）仿真;

(3) 完成设计实现（lmplement）布局布线后仿真。

## 3、模块设计框图



## 4、实验原理：

**计时模块使用异步复位:**

输入/输出

- 3 个输入变量：clk，SW3(rst)，SW0(En)
- 1 个输出变量：LEDOut [7:0]，使用八位 LED 代表输出。
- 功能：可以通过修改 SW0 确定计数模式，关断 SW3 实现开关。

**时钟分频：**

板载电路上有引脚 C9 接入一个 50MHz 的时钟信号，可以通过 clock_div.v 设计 clk_10 信号以表示 100ms 等，这里设计时使用计数变量 one_ten 表示默认 clk=0.1s 与 1s 时间的转换。

**测试台：**

在指定时间后启动 SW3，在此之前无信号输出；在 500 个单位时长前为 SW0=1 计数模式，在之后修改 SW0=1，修改时所有信号置 0，SW3=0。

## 5、Verilog 模块设计

**//顶层模块**

**//counter.v**

```
`timescale 1ns/1ps;
//`include "clock_div.v"

module eight_bit_counter(
  input SW3, // rst
  input SW0, // enable
  input clk,
  output reg [7:0] LEDOut
);



//reg clk_10=1'b0;
reg [3:0] one_ten = 4'b0000;//100ms=1s
//reg[21:0]k;
```

```
//creating new clock
//always @(posedge clk)
//begin
//if (k>=2500000) //è®¡æ□¶
//begin
//clk_10<=~clk_10; //状态转换，从高电平跳到低电平，或从低电平跳到高电平
//k<=0;
//end
//else
//k<=k+1;
//end


always @(posedge clk or negedge SW3)
  begin

    //SW3 == 0, stop
    if(!SW3) begin
        //initial parameter
          LEDOut <= 8'b0;
          //one_ten <= 1'd0;
        end

    //SW3 == 1, run
    else begin

        //SW0 == 0
        if(!SW0) begin
                if (one_ten < 4'b1001) begin one_ten <= one_ten+4'b0001; end
                else begin
                  one_ten <= 4'b0000;
                    if(LEDOut>=8'b0111_1111) begin LEDOut <= 8'b0; end
                    else begin LEDOut <= LEDOut+8'b0000_0001; end
                end
               end

        //SW0 == 1
        else begin
```

```
            if (LEDOut [3:0] < 4'b1001) begin LEDOut [3:0] <= LEDOut [3:0] +4'b0001; end
            else begin LEDOut [3:0] <= 4'b0000; end

            if(one_ten < 4'b1001) begin one_ten <= one_ten+4'b0001; end
            else begin
               one_ten <=4'b0000;
                 if(LEDOut [7:4] >= 4'b1001) begin LEDOut [7:4] <= 4'b0000; end
                 else begin LEDOut [7:4] <= LEDOut [7:4] + 4'b0001; end

               end
            end
        end
  end

endmodule
```

**//引脚约束**

**//counter.ucf**

```
NET "SW0" LOC = "L13" | IOSTANDARD = LVTTL | SLEW = SLOW | DRIVE = 8;
NET "clk" LOC = "C9" | IOSTANDARD = LVCMOS33;
NET "SW3" LOC = "L14" | 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;
```

**//test_bench**

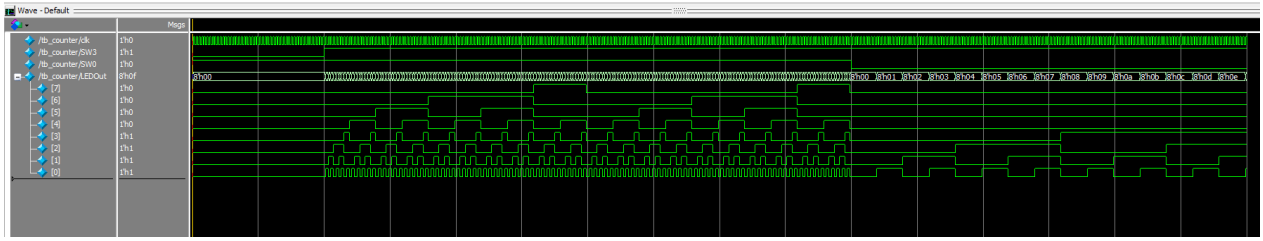
**//tb_counter.v**

```
`timescale 1ns/1ps;
`include "counter.v"

module tb_counter;

// Inputs
```

```
reg clk, SW3, SW0;

// Outputs
wire [7:0] LEDOut;


initial begin SW3 = 1'b0; #100 SW3 = 1'b1; end
//initial begin SW0 = 0; forever #100 SW0 = ~SW0; end


initial begin clk = 0; forever #1 clk=~clk; end


initial begin
  SW0 = 1;
  #500 SW0 = ~SW0;
end

eight_bit_counter uut(
.SW3(SW3),
.SW0(SW0),
.clk(clk),
.LEDOut(LEDOut)
);

initial
#30000 $stop;

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

endmodule
```

## 6、试验仿真结果和分析

如图：功能仿真中，正确输出：



显示输出

解释：起初 SW3=0，不启动，无输出；

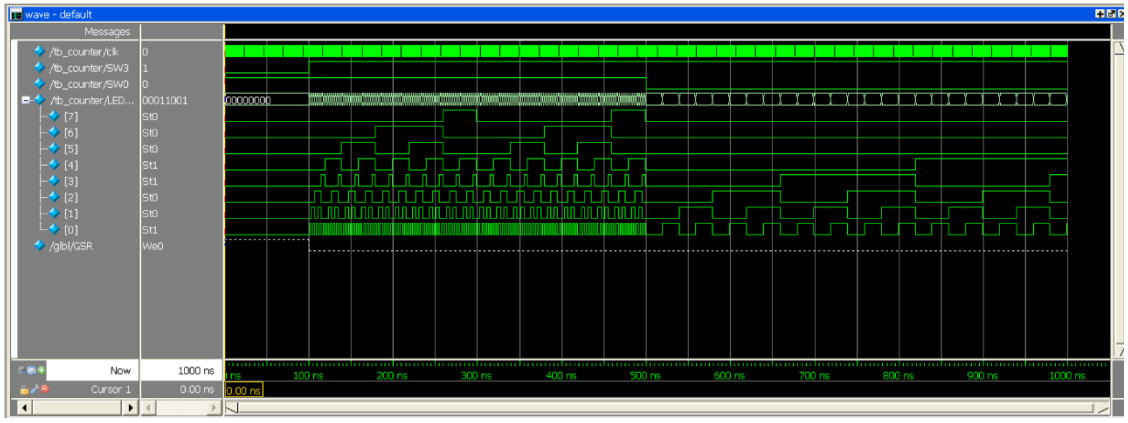
SW3=1 后，启动时 SW0=1，按第二种计数方式执行，前四位与后四位分别计数；

指定时间后，关断 SW0，修改为 SW3=1，继续执行，此时 0-127 计数。

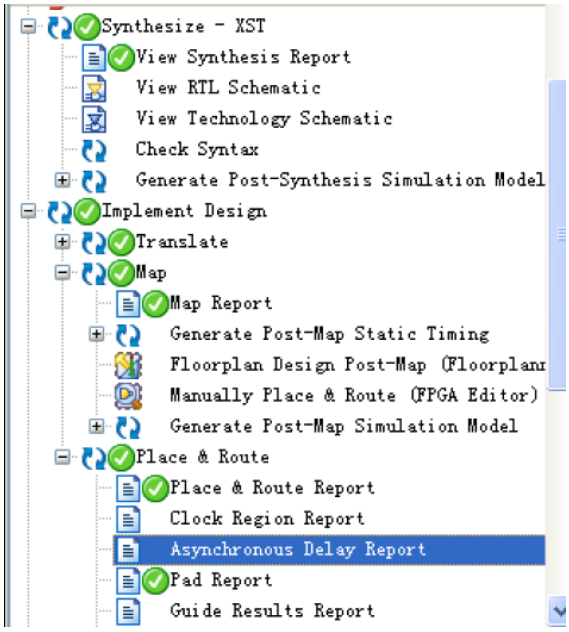
```
VSIM 59> run 800000
# time=                   0,ledout=00000000
# time=                 101,ledout=00000001
# time=                 103,ledout=00000010
# time=                 105,ledout=00000011
# time=                 107,ledout=00000100
# time=                 109,ledout=00000101
# time=                 111,ledout=00000110
# time=                 113,ledout=00000111
# time=                 115,ledout=00001000
# time=                 117,ledout=00001001
# time=                 119,ledout=00010000
# time=                 121,ledout=00010001
# time=                 123,ledout=00010010
# time=                 125,ledout=00010011
# time=                 127,ledout=00010100
# time=                 129,ledout=00010101
# time=                 131,ledout=00010110
# time=                 133,ledout=00010111
# time=                 135,ledout=00011000
# time=                 137,ledout=00011001
# time=                 139,ledout=00100000
# time=                 141,ledout=00100001
# time=                 143,ledout=00100010
# time=                 145,ledout=00100011
# time=                 147,ledout=00100100
# time=                 149,ledout=00100101
# time=                 151,ledout=00100110
# time=                 153,ledout=00100111
# time=                 155,ledout=00101000
# time=                 157,ledout=00101001
# time=                 159,ledout=00110000
# time=                 161,ledout=00110001
# time=                 163,ledout=00110010
# time=                 165,ledout=00110011
# time=                 167,ledout=00110100
# time=                 169,ledout=00110101
# time=                 171,ledout=00110110
# time=                 173,ledout=00110111
# time=                 175,ledout=00111000
```

```
# time=                 499,ledout=00000000
# time=                 519,ledout=00000001
# time=                 539,ledout=00000010
# time=                 559,ledout=00000011
# time=                 579,ledout=00000100
# time=                 599,ledout=00000101
# time=                 619,ledout=00000110
# time=                 639,ledout=00000111
# time=                 659,ledout=00001000
# time=                 679,ledout=00001001
# time=                 699,ledout=00001010
# time=                 719,ledout=00001011
# time=                 739,ledout=00001100
# time=                 759,ledout=00001101
# time=                 779,ledout=00001110
# time=                 799,ledout=00001111
```

在对应时刻显示计时器输出的情况 (左 SW0=1 右 SW0=0)



后仿真，延时输入信号




执行仿真和实现之后的全仿真过程