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。

1. 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SWO=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О到9，循环反复。

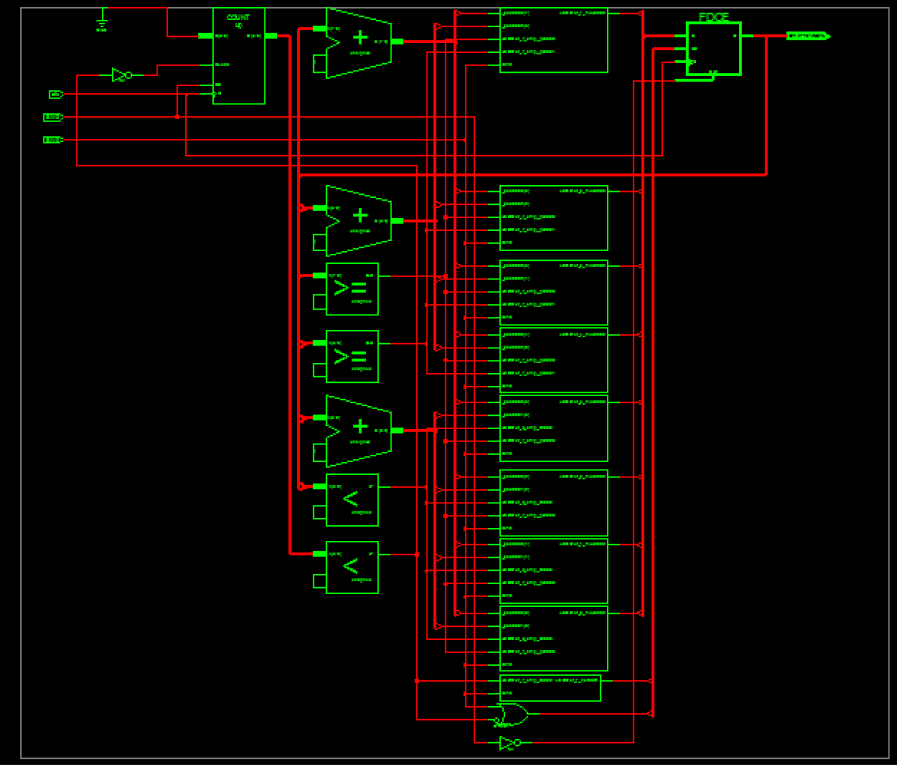
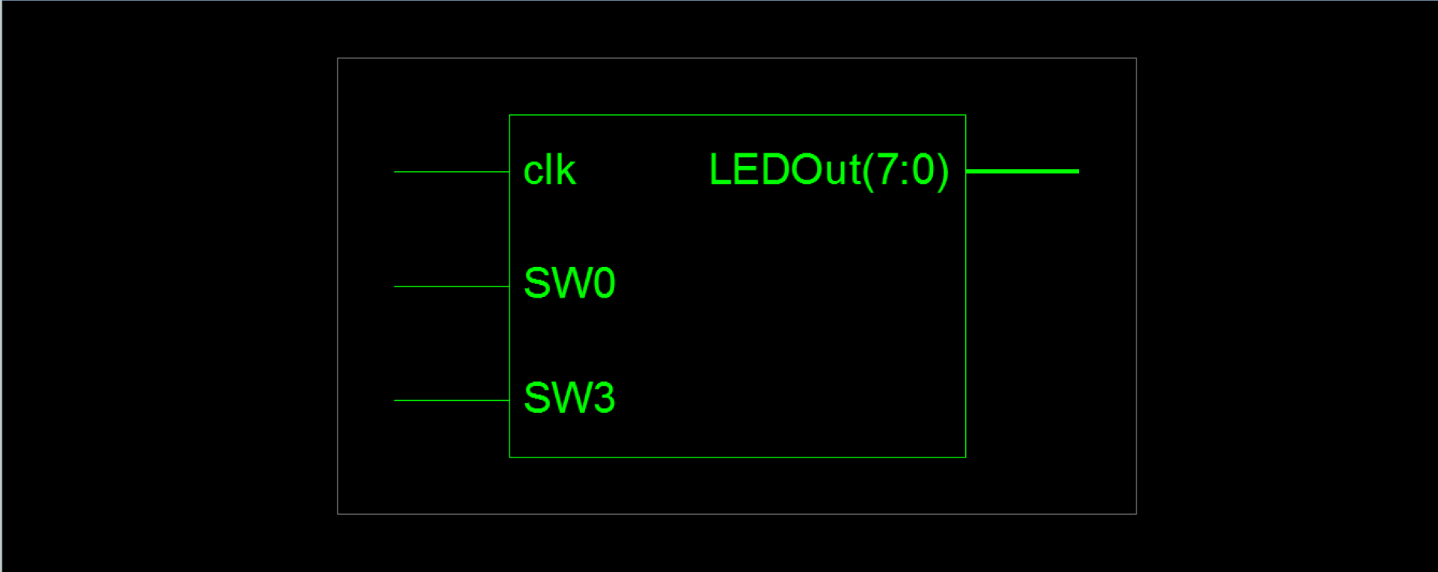
低四位:0~9，每1/10秒计数，从0到9，循环反复。

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。
4. 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;

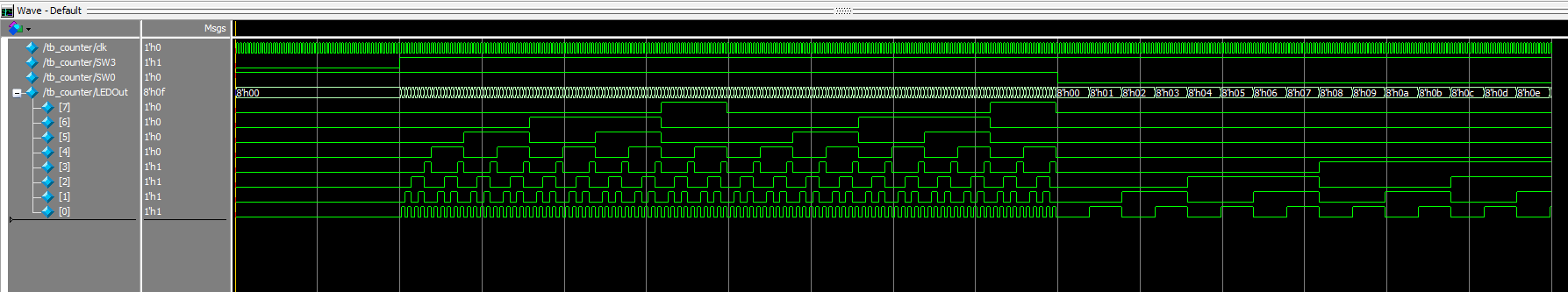
initial

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

endmodule

6、试验仿真结果和分析

如图：功能仿真中，正确输出：

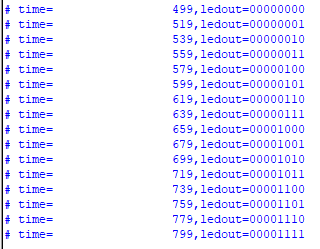
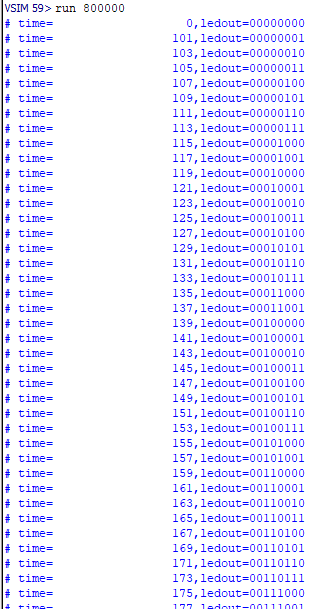


显示输出

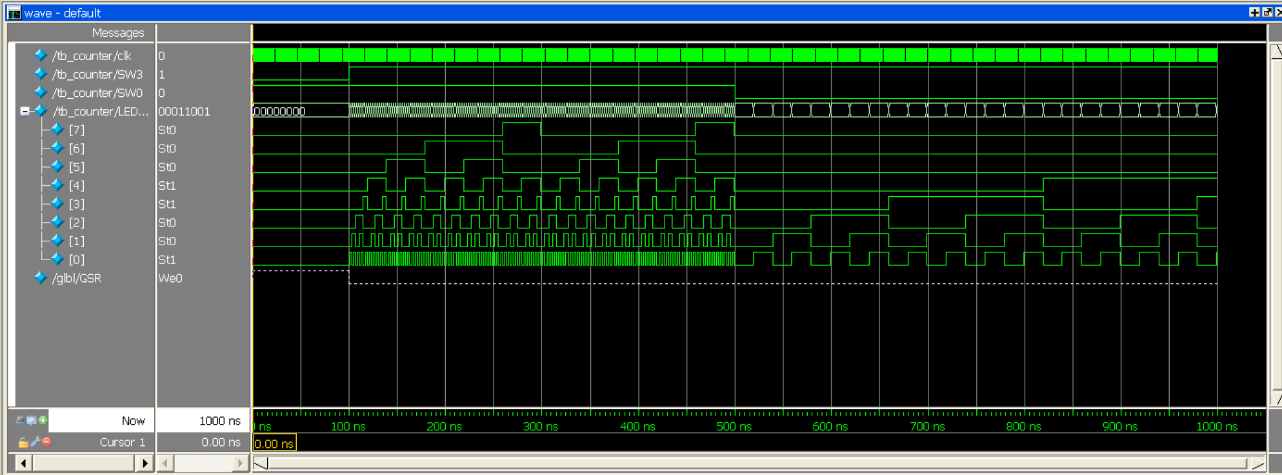
解释：起初SW3=0，不启动，无输出；

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；

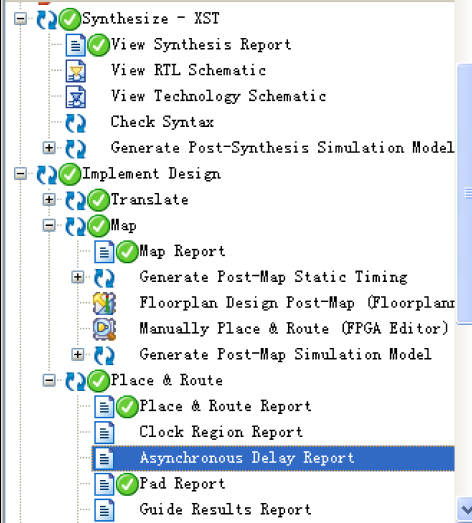
指定时间后，关断SW0，修改为SW3=1，继续执行，此时0-127计数。



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