

实验名称：流水灯与数字秒表 姓名：__ 学号：

浙江大学



本科实验报告

装
订
线

姓名：_____

学院： 生物医学工程与仪器科学学院

系： 生物医学工程系

专业： 生物医学工程

学号：_____

指导教师： 余锋

实验名称: 流水灯与数字秒表 姓名: 学号:

专业: 生物医学工程

姓名:

学号:

日期: 2025.3.4

地点: 教 6-204

浙江大学实验报告

课程名称: 硬件描述语言 指导老师: 余锋

实验名称: HDL 实验一

一、实验要求

二、实验代码与注释

三、仿真结果与分析

四、讨论与心得

一、实验要求

1、设计流水灯

提供输入: `clk_i`——时钟信号, 100MHz

`rst_i`——复位信号, 高电平复位

要求输出: `led_o`——初始状态为 `led_o[0]` 亮, 其他灯不亮, 每过 10us 左右亮的灯向左传递, 当传递到最后的 `led_o[7]` 时, 下一次再传递为 `led_o[0]`

2、设计数字秒表

提供输入: `clk_i`——时钟信号, 100MHz

`rst_i`——复位信号, 高电平复位

`trig_i`——触发信号, 在该信号有效时输出一次计时数据

要求输出: `second_o`——“秒”输出

`millisec_o`——“毫秒”输出

`valid_o`——表示输出有效

注意:

(1) 当 `rst_i` 信号有效时秒表的输出全为 0, 无效后开始计时;

(2) 该实验需要 3 个计数器, 一个基本计数器 `base_cnt`, 一个“毫秒”计数器 `millisec`, 一

实验名称: 流水灯与数字秒表 姓名: 学号:

个“秒”计数器 second;

(3) base_cnt 值每达到 8 则更新 millisec 的值加 1, 同时 base_cnt 清零, millisec 值每达到 1024 则更新 second 的值加 1, 同时 millisec 清零, second 值每达到 59 后重新开始计时。

(4) 一旦 trig_i 信号有效则将当前 millisec 输出到 millisec_o, 将当前 second 输出到 second_o, 同时将 valid_o 置为高电平表示输出数据有效

二、实验代码与注释

1、设计流水灯

```
module stream_led(  
    input          clk_i,  
    input          rst_i,  
    output reg [7:0] led_o  
);  
  
    reg [10:0] clk_cnt;  
  
    always @(posedge clk_i) begin  
        if (rst_i) begin  
            clk_cnt <= 0;  
            led_o  <= 8'b00000001;  
        end else begin  
            if (clk_cnt == 11'd999) begin  
                clk_cnt <= 0;  
                led_o <= {led_o[6:0], led_o[7]};  
            end else begin  
                clk_cnt <= clk_cnt + 1;  
            end  
        end  
    end
```

实验名称: 流水灯与数字秒表 姓名: __ 学号: __

```
    end  
end  
  
endmodule
```

```
`timescale 1ns/1ns  
module tb_stream_led;  
  
    reg        clk;  
    reg        rst;  
    wire [7:0] led;  
  
    stream_led inst_stream_led(  
        .clk_i(clk),  
        .rst_i(rst),  
        .led_o(led)  
    );  
  
    initial begin  
        clk <= 1'b0;  
        rst <= 1'b1;  
        #100;  
        rst <= 1'b0;  
    end  
  
    always begin  
        #5 clk = ~clk;  
    end
```

实验名称：流水灯与数字秒表 姓名：__ 学号：

```
endmodule
```

首先要在复位时使计数器清零，同时也是第一个 LED 亮的时候；当计数器 `clk_cnt` 计数 1000 次（10us）的时候就置零，然后让 `led_o` 左移一位。

2、设计数字秒表

```
module stopwatch(  
    input          clk_i,  
    input          rst_i,  
    input          trig_i,  
    output reg [10:0] millisec_o,  
    output reg [5:0]  second_o,  
    output reg      valid_o  
);  
  
    reg [3:0]  base_cnt;  
    reg [10:0] millisec;  
    reg [5:0]  second;  
  
    always @(posedge clk_i or posedge rst_i) begin  
        if(rst_i) begin  
            base_cnt <= 4'd0;  
        end  
        else begin  
            if(base_cnt == 4'd8) begin  
                base_cnt <= 4'd0;  
            end  
            else begin  
                base_cnt <= base_cnt + 4'd1;  
            end  
        end  
    end
```

实验名称： 流水灯与数字秒表 姓名： 学号：

```
        end
    end
end
always @(posedge clk_i or posedge rst_i) begin
    if(rst_i) begin
        millisec <= 11'd0;
    end
    else begin
        if(base_cnt == 4'd8) begin
            if(millisec == 11'd1023) begin
                millisec <= 11'd0;
            end
            else begin
                millisec <= millisec + 11'd1;
            end
        end
    end
end
end
always @(posedge clk_i or posedge rst_i) begin
    if(rst_i) begin
        second <= 6'd0;
    end
    else begin
        if(base_cnt == 4'd8) begin
            if(millisec == 11'd1023) begin
                if(second == 6'd59) begin
                    second <= 6'd0;
                end
            end
            else begin

```

实验名称: 流水灯与数字秒表 姓名: 学号:

```

                second <= second + 6'd1;
            end
        end
    end
end
always @(posedge clk_i or posedge rst_i) begin
    if(rst_i) begin
        millisec_o <= 11'd0;
        second_o <= 6'd0;
        valid_o <= 1'b0;
    end
    else begin
        if(trig_i) begin
            millisec_o <= millisec;
            second_o <= second;
            valid_o <= 1'b1;
        end
        else begin
            valid_o <= 1'b0;
        end
    end
end
endmodule

```

```

module tb_stopwatch;

    reg        clk;

```

实验名称: 流水灯与数字秒表 姓名: __ 学号: __

```
reg        rst;

reg        trig;

wire [5:0]  second;

wire [10:0] millisec;

wire        valid;


stopwatch my_stopwatch(
    .clk_i(clk),
    .rst_i(rst),
    .trig_i(trig),
    .second_o(second),
    .millisec_o(millisec),
    .valid_o(valid)
);


initial begin
    clk <= 1'b0;
    trig <= 1'b0;
    rst <= 1'b1;
    #100;
    rst <= 1'b0;
    #100000;
    trig <= 1'b1;
end

always begin
    #5 clk =~ clk;
end

always @(posedge clk) begin
```



```

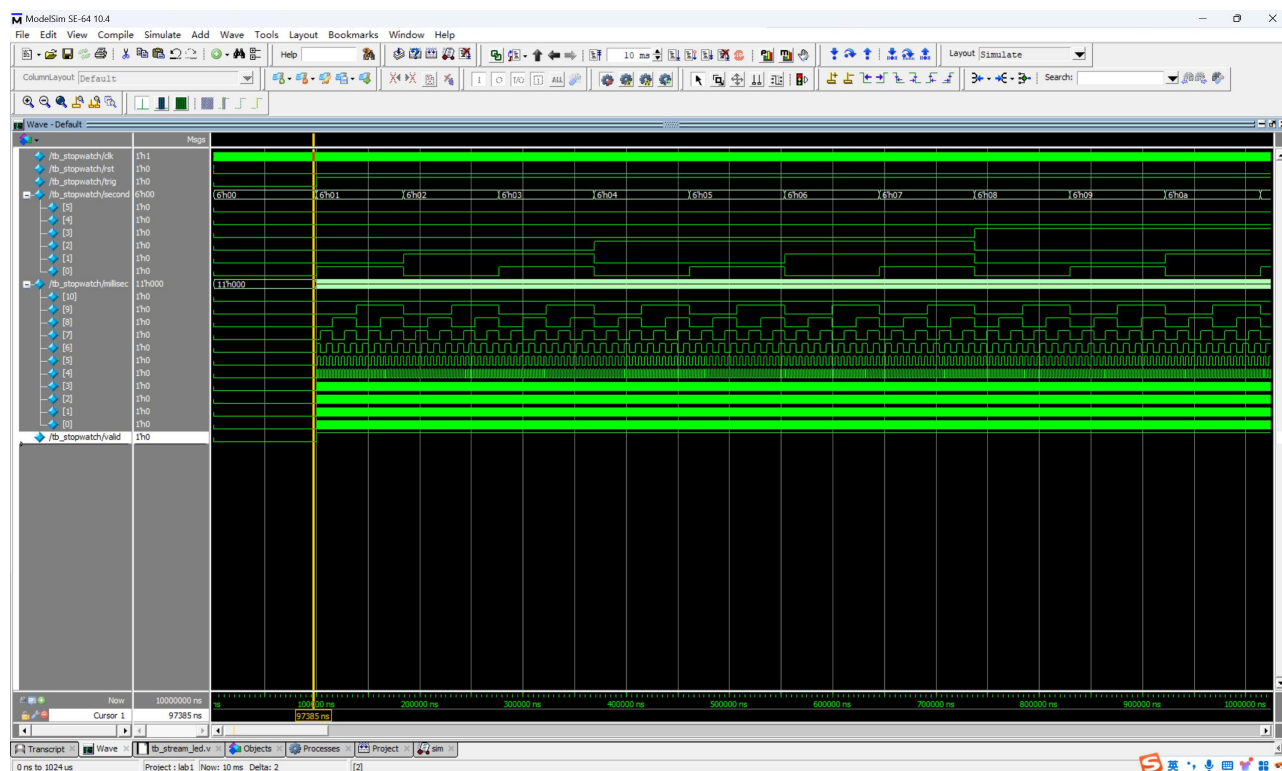
    if(valid)
        $display("%d second %d millisecond", second, millisec);
    end
endmodule

```

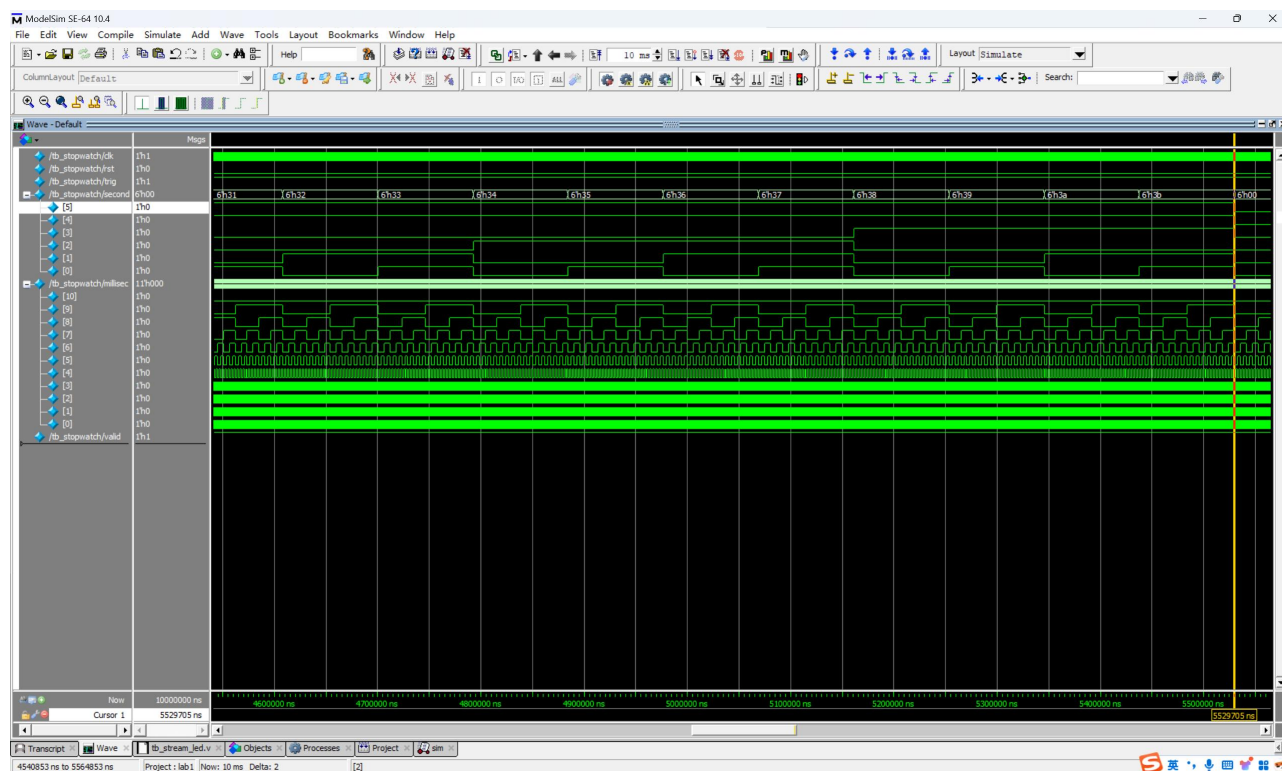
三、仿真结果与分析

2、设计数字秒表

实验名称：流水灯与数字秒表 姓名：__ 学号：__



在初始阶段，由图可见 trig_i 有效时， millicsec_o 和 second_o 立即输出当前值， valid_o 变高 1 个时钟周期。



base_cnt 每 8 个时钟周期归零， millicsec 每 1024 次 base_cnt 归零递增 1 秒， second 每 60 秒循环。上图是 second 归 0 时的情况。

实验名称：流水灯与数字秒表 姓名：__ 学号：

四、讨论与心得

通过本次实验，我不仅完成了流水灯与数字秒表的设计，更在实践中深化了对 Verilog 时序逻辑的理解。在流水灯模块中，通过 100MHz 时钟驱动的循环移位设计，让我掌握了如何通过计数器精确控制时序，尤其是在复位与循环逻辑的处理上，学会了通过寄存器级联实现状态转移。而在秒表模块中，三级计数器的级联设计是最大挑战，从基础的 8 分频到毫秒累加再到秒循环，每一步都需要精准的时序计算和状态同步。在 ModelSim 波形调试过程中，通过观察时钟边沿、复位信号与计数器的响应关系，我逐渐熟悉了如何通过波形验证逻辑正确性，特别是在触发信号的捕捉与输出有效性的控制上。

这些经验不仅提升了我的硬件调试能力，更让我意识到模块化设计的重要性 —— 将复杂系统拆解为可独立验证的子模块，能够显著降低设计复杂度。实验中遇到的挑战，比如秒表进位逻辑的延迟处理，迫使我反复推敲代码的时序约束。总体而言，本次实验为我后续设计更复杂的数字系统奠定了坚实基础，也让我对 Verilog 的硬件描述特性有了更直观的认识。