

实验：基于 Verilog 和 FPGA/CPLD 的多功能秒表设计实验报告

姓名：Meng Yit Koh 学号：517030990022

实验目的

1. 初步掌握利用 Verilog 硬件描述语言进行逻辑功能设计的原理和方法。
2. 理解和掌握运用大规模可编程逻辑器件进行逻辑设计的原理和方法。
3. 理解硬件实现方法中的并行性，联系软件实现方法中的并发性。
4. 理解硬件和软件是相辅相成、并在设计 and 应用方法上的优势互补的特点。
5. 本实验学习积累的 Verilog 硬件描述语言和对 FPGA/CPLD 的编程操作，是进行后续《计算机组成原理》部分课程实验，设计实现计算机逻辑的基础。

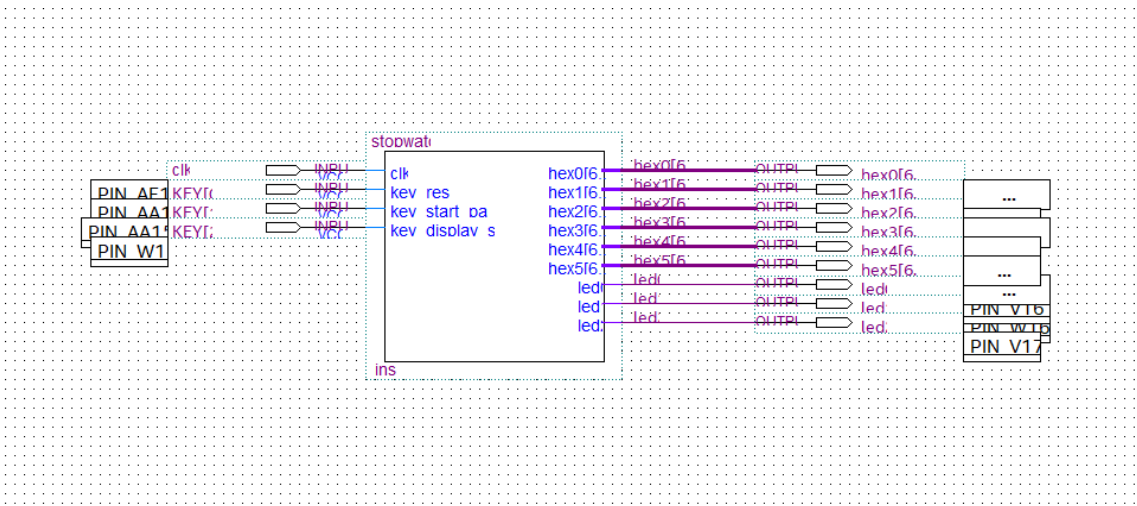
本次实验所用的仪器以及软件

- DE1-SOC 实验板
- Quartus Prime Lite 18.1.0

实验任务

1. 运用 Verilog 硬件描述语言，基于 DE1-SoC 实验板，设计实现一个具有较多功能的计时秒表。
2. 要求将 6 个数码管设计为具有“分：秒：毫秒”显示，按键的控制动作有：“计时复位”、“计数/暂停”、“显示暂停/显示继续”等。功能能够满足马拉松或长跑运动员的计时需要。
3. 利用示波器观察按键的抖动，设计按键电路的消抖方法。
4. 在实验报告中详细报告自己的设计过程、步骤及 Verilog 代码。

Block Diagram



消抖处理

代码如下：

```
// debouncer for key_reset
if (key_reset_state && !key_reset) begin
    counter_reset = counter_reset + 1;
```

```

    if (counter_reset == 8'b11111111) begin
        counter_reset = 0;
        key_reset_state = ~key_reset_state;
    end
end else if (!key_reset_state && key_reset) begin
    counter_reset = counter_reset + 1;

    if (counter_reset == 8'b11111111) begin
        counter_reset = 0;
        key_reset_state = ~key_reset_state;

        msecond_counter_low = 0;
        msecond_counter_high = 0;
        second_counter_low = 0;
        second_counter_high = 0;
        minute_counter_low = 0;
        minute_counter_high = 0;
    end
end else begin
    counter_reset = 0; // for continuing state
end

// debouncer for key_start_pause
if (key_start_pause_state && !key_start_pause) begin
    counter_start_pause = counter_start_pause + 1;

    if (counter_start_pause == 8'b11111111) begin
        counter_start_pause = 0;
        key_start_pause_state = ~key_start_pause_state;
        timess = ~timess;
    end
end else if (!key_start_pause_state && key_start_pause) begin
    counter_start_pause = counter_start_pause + 1;

    if (counter_start_pause == 8'b11111111) begin
        counter_start_pause = 0;
        key_start_pause_state = ~key_start_pause_state;
    end
end else begin
    counter_start_pause = 0;
end

// debouncer for key_display_stop
if (key_display_stop_state && !key_display_stop) begin
    counter_display_stop = counter_display_stop + 1;

    if (counter_display_stop == 8'b11111111) begin
        counter_display_stop = 0;
        key_display_stop_state = ~key_display_stop_state;
        freeze = ~freeze;
    end
end

```

```
end else if (!key_display_stop_state && key_display_stop) begin
    counter_display_stop = counter_display_stop + 1;

    if (counter_display_stop == 8'b11111111) begin
        counter_display_stop = 0;
        key_display_stop_state = ~key_display_stop_state;
    end
end else begin
    counter_display_stop = 0;
end
```

注解：使用counter等待一段时间在记录按下，避免因为抖动而引起的按钮过灵事件。

时间信息计算

代码如下:

```
// update time
if (times) begin
    stopwatch_clk = stopwatch_clk + 1;

if (stopwatch_clk == 500000) begin
    stopwatch_clk = 0;
    msecond_counter_low = msecond_counter_low + 1;

if (msecond_counter_low == 10) begin
        msecond_counter_low = 0;
        msecond_counter_high = msecond_counter_high + 1;

if (msecond_counter_high == 10) begin
            msecond_counter_high = 0;
            second_counter_low = second_counter_low + 1;

if (second_counter_low == 10) begin
                second_counter_low = 0;
                second_counter_high = second_counter_high + 1;

if (second_counter_high == 6) begin
                    second_counter_high = 0;
                    minute_counter_low = minute_counter_low + 1;

if (minute_counter_low == 10) begin
                        minute_counter_low = 0;
                        minute_counter_high = minute_counter_high + 1;

if (minute_counter_high == 10) begin
                            minute_counter_high = 0;
                        end
                    end
                end
            end
        end
    end
end
end
```

```
    end
end
```

注解：低位达到一定数值，高位+1，低位赋0，一直循环。

DE1 秒表显示

代码如下：

```
// 4bit 的 BCD 码至 7 段 LED 数码管译码器模块
// 可供实例化共 6 个显示译码模块
module sevenseg(data, ledsegments);
input [3:0] data;
output [6:0] ledsegments;
reg [6:0] ledsegments;
always @ ( * )
    // gfe_dcba, 7 段 LED 数码管的位段编号
    // 654_3210, DE1-SOC 板上的信号位编号, DE1-SOC 板上的数码管为共阳极接法。
    case(data)
        0: ledsegments = 7'b100_0000;
        1: ledsegments = 7'b111_1001;
        2: ledsegments = 7'b010_0100;
        3: ledsegments = 7'b011_0000;
        4: ledsegments = 7'b001_1001;
        5: ledsegments = 7'b001_0010;
        6: ledsegments = 7'b000_0010;
        7: ledsegments = 7'b111_1000;
        8: ledsegments = 7'b000_0000;
        9: ledsegments = 7'b001_0000;
        default: ledsegments = 7'b111_1111; // 其它值时全灭。
    endcase
endmodule // stopwatch
```

注解：0表示亮，1表示暗。（代码由老师提供）

按键功能

- key2: 暂停时间显示更新。重按跳到应该值并继续每毫秒更新。
- key1: 开始/暂停/继续计时。
- key0: 重置所有状态。

实验总结

实验代码经过编译综合，载入到开发板后，能正常完成预期的秒表功能。按键消抖效果良好，未出现按键不响应或响应多次的现象。

感谢上海交通大学软件学院开设“数字电路设计这门课”，让我学习如何使用 FPGA 实现我想要的数字设计。感谢王老师的教导。感谢热心的助教。