

《基于 Verilog 和 FPGA/CPLD 的多功能秒表设计》实验报告

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。

实验内容和任务：

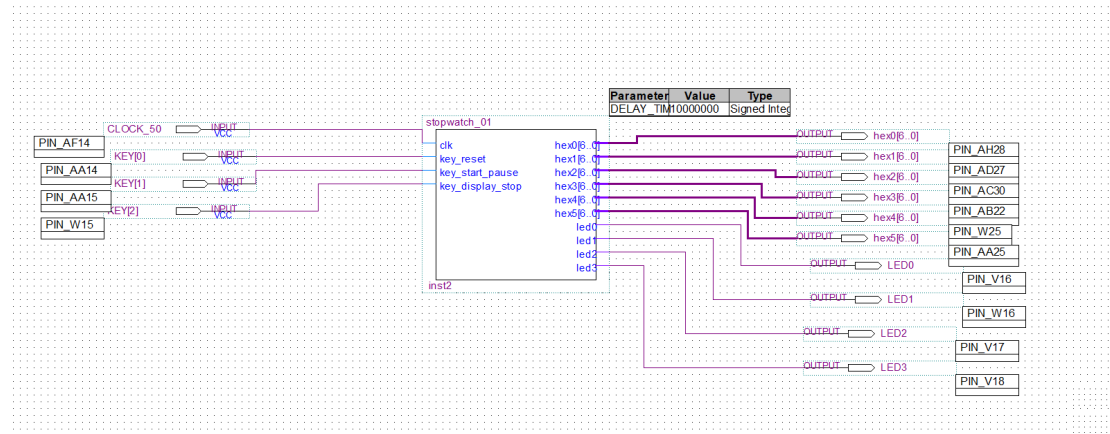
1. 运用 Verilog 硬件描述语言，基于 DE1-SOC 实验板，设计实现一个具有较多功能的计时秒表。
2. 要求将 6 个数码管设计为具有“分：秒：毫秒”显示，按键的控制动作有：“计时复位”、“计数/暂停”、“显示暂停/显示继续”等。功能能够满足马拉松或长跑运动员的计时需要。
3. 利用示波器观察按键的抖动，设计按键电路的消抖方法。
4. 在实验报告中详细报告自己的设计过程、步骤及 Verilog 代码。

实验仪器

- DE1-SOC 实验板
- 软件：Altera Quartus II 13.1

实验内容

电路设计



stopwatch 模块有四个输入：时钟输入 `CLOCK_50`，3 个按键输入 `KEY[0:3]` 分别对应功能按键“计时复位”、“计数/暂停”、“显示暂停/显示继续”。同时，模块将秒表的时间和调试状态输出到 `hex0[6:0]~hex5[6:0]` 六个七段数码管和 4 个 LED 灯中。

硬件设计代码

按键状态及消抖

```
always @ (posedge clk) // 每一个时钟上升沿开始触发下面的逻辑，
// 进行计时后各部分的刷新工作
begin
    //start 按键
    if(key_start_pause && !start_1_time)begin //当未被按下且状态变量不为1
        counter_start <= counter_start+1;
        if (counter_start == 500000)begin//等10ms
            start_1_time <= ~start_1_time;
            counter_start<=0;
            start <= ~start; //在放开后更新状态变量，计时开始
        end
    end
    else if(!key_start_pause && start_1_time)begin//当按钮被按下且状态变量为1
        counter_start <= counter_start+1;
        if(counter_start == 500000)begin
            start_1_time <= ~start_1_time;
            counter_start<=0;
        end
    end
end
```

```

        end
    end
    else begin
        counter_start <=0;
    end

```

计时

```

always @ (posedge clk) // 每一个时钟上升沿开始触发下面的逻辑，
begin
    // ..... 按键部分代码
    if (start)begin
        counter_50M <= counter_50M+1;
        if(counter_50M ==500000)begin // 每500000 次（即10ms）
            counter_50M <=0;
            .....

```

实验总结

实验结果

将设计代码编译后，烧录进入开发板中，秒表计时准确，按键的消抖效果良好，响应及时准确。

实验过程中的问题

- 第一次烧录后发现，按键响应存在问题：

第一次的代码实现未对按键进行消抖，按键按下后状态位转换存在问题。

- 秒表时间略慢于设计预期：

检查后发现，计时部分代码存在问题：

```

if(counter_50M ==500000)begin//10ms
counter_50M =0;

msecond_counter_low = msecond_counter_low + 1;

```

由于采用了阻塞赋值的方式，导致计数更新阻塞，使得秒表计时慢于实际设计时间。改用非阻塞赋值后，计时恢复正常

- 按键按下后秒表计时停止：

同样是由于阻塞赋值的方式，导致 start 变量更新阻塞，进而导致计时阻塞。

感想

通过本次实验我比较熟练的掌握了 Verilog 硬件描述语言和对 FPGA/CPLD 的编程操作，也熟悉了使用 Altera Quartus 和 ModelSim 进行 FPGA 开发和模拟的全过程。