

上海交通大学

SHANGHAI JIAO TONG UNIVERSITY



数字部件实验一

基于 Verilog 和 FPGA/CPLD 的多功能秒表设计

姓 名：王鑫伟

学 院：电子信息与电气工程学院

学 号：516030910041

班 级：F1603701

2018 年 11 月 24 日

一、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、并在设计和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。

二、实验内容

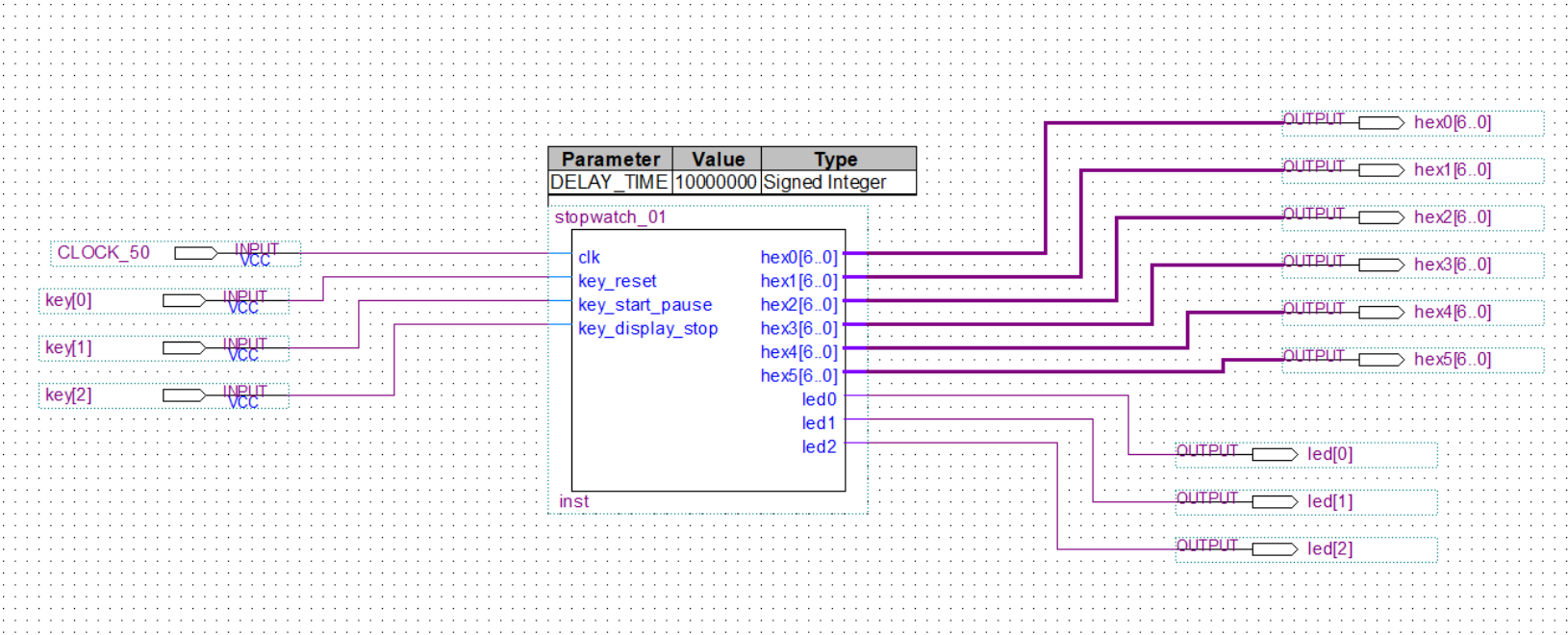
- 1. 运用 Verilog 硬件描述语言，基于 DE1-SOC 实验板，设计实现一个具有较多功能的计时秒表。
- 2. 要求将 6 个数码管设计为具有“分：秒：毫秒”显示，按键的控制动作有：“计时复位”、“计数/暂停”、“显示暂停/显示继续”等。功能能够满足马拉松或长跑运动员的计时需要。
- 3. 利用示波器观察按键的抖动，设计按键电路的消抖方法。
- 4. 在实验报告中详细报告自己的设计过程、步骤及 Verilog 代码。

三、实验设计

3.1 实验仪器与平台

硬件仪器：DE1-SoC 实验板
软件平台：Altera Quartus II 13.1

3.2 顶层设计



[图一] stopwatch.bdf 图示

本次实验的顶层设计图如[图一]所示，由主模块 stopwatch_01、四个 input 接口和九个 output 接口组成。

四个 input 接口中，CLOCK_50 是时钟，会提供一个频率为 50Hz 的时钟信号,key_reset 按键控制“计时复位”动作，key_start_pause 按键控制“计数/暂停”动作，key_display_stop 按键控制“显示暂停/显示继续”动作。

六个 output 接口中，led0、led1、led2 为标记指示灯，在相对应的 key0、key1、key2 按键按下时会亮起；hex5~hex0 为 7 段数码管，具有“分：秒：毫秒”显示的功能。

3.3 Verilog 代码设计

以下仅给出部分代码（实际上关键部分都已经包含）

```
10 parameter DELAY_TIME = 5000000; //消抖动延迟时间
11
12 reg display_work; //显示暂停标记
13 reg counter_work; //计时暂停标记
14
```

新增常量 DELAY_TIME 表示消抖延迟时间

```
48 //当复位键被按下时，led0亮起
49 always @ (key_reset)
50 begin
51     led0 = !key_reset;
52 end
53
54 //当计时键被按下时，led1亮起
55 always @ (key_start_pause)
56 begin
57     led1 = !key_start_pause;
58 end
59
60 //当显示键被按下时，led2亮起
61 always @ (key_display_stop)
62 begin
63     led2 = !key_display_stop;
64 end
65
```

根据按键控制 led 灯的亮起

```
69 if (reset_1_time && !key_reset)
70 begin
71     counter_reset = counter_reset + 1;
72     if (counter_reset == DELAY_TIME)
73     begin
74         counter_reset = 0;
75         reset_1_time = ~reset_1_time;
76
77         counter_work = 0;
78         minute_counter_high = 0;
79         minute_counter_low = 0;
80         second_counter_high = 0;
81         second_counter_low = 0;
82         msecond_counter_high = 0;
83         msecond_counter_low = 0;
84
85         display_work = 1;
86     end
87 end
```

消抖后进行“复位”的相关操作

```
104 if (start_1_time && !key_start_pause)
105 begin
106     counter_start = counter_start + 1;
107     if (counter_start == DELAY_TIME)
108     begin
109         counter_start = 0;
110         start_1_time = ~start_1_time;
111
112         counter_work = !counter_work;
113     end
114 end
```

消抖后进行“计时开始/暂停”的相关操作，并判断 counter_work；“显示暂停/继续”同理

```
158 if (display_work)
159 begin
160     minute_display_high = minute_counter_high;
161     minute_display_low = minute_counter_low;
162     second_display_high = second_counter_high;
163     second_display_low = second_counter_low;
164     msecond_display_high = msecond_counter_high;
165     msecond_display_low = msecond_counter_low;
166 end
167
```

如果 display_work 就将时间写入数码管

```
169 |         if (counter_work)
170 |         begin
171 |             counter_50M = counter_50M + 1;
172 |
173 |             if (counter_50M == 500000)
174 |             begin
175 |                 counter_50M = 0;
176 |                 msecond_counter_low = msecond_counter_low + 1;
177 |
178 |                 if (msecond_counter_low == 10)
179 |                 begin
180 |                     msecond_counter_low = 0;
181 |                     msecond_counter_high = msecond_counter_high + 1;
182 |
183 |                     if (msecond_counter_high == 10)
184 |                     begin
185 |                         msecond_counter_high = 0;
186 |                         second_counter_low = second_counter_low + 1;
187 |
```

如果 counter_work 就继续计时

四、实验步骤

设计了一套实验操作及理想现象

1. 按下 key0, reset: led0 亮，数码管清零
2. 按下 key1, start count: led1 亮，数码管持续变化
3. 按下 key2, stop display: led2 亮，数码管停止变化
4. 按下 key2, start display: led2 亮，数码管有一瞬间突变，然后持续变化
5. 按下 key1, pause count: led1 亮，数码管停止变化
6. 按下 key0, reset: led0 亮，数码管清零

五、实验总结与感悟

实际上大学这两年多的时间里在大一工科创 I 以后就没怎么碰过硬件了，工科创 I 也只是焊了万用表，所以一开始上这门课的时候有很强的陌生感。尽管有教程，但无论是 Verilog 硬件描述语言还是 Quartus II 13.1 软件的操作还是花费了不少精力，实验总体做的有一些吃力，不过相信之后会好一些的。谢谢老师和助教在课堂上以及 QQ 群中对我们实验的帮助。