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。

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。

实验设计：

本质上这个实验是一个简单的多重循环：

clk每50ns为一个周期，所以每200000次周期为10ms，ms位会变动一次，

ms位每变动100次，s位变动1次，

s位每变动60次，min位变动一次。

比较有难度的地方在于需要消抖，为了消抖，额外为一个按钮定义一个状态寄存器，按下按钮之后过一段时间再进入功能，状态寄存器翻转，松开后，再经过一段时间，寄存器再次翻转复位。这就杜绝了抖动的干扰以及快速多次按动按钮的隐患。

以reset key为例：

if (reset\_1\_time && !key\_reset) begin // state is about to change

counter\_reset = counter\_reset + 1;

if (counter\_reset == DELAY\_TIME) begin // signal has been in new state for long enough time

counter\_reset = 0; // clear counter

reset\_1\_time = ~reset\_1\_time; // flip the state

…// reset stopwatch by setting all counters to 0

end

end else if (!reset\_1\_time && key\_reset) begin

counter\_reset = counter\_reset + 1;

if (counter\_reset == DELAY\_TIME) begin

counter\_reset = 0;

reset\_1\_time = ~reset\_1\_time;

end

end

display功能：就是将count寄存器中的值读入display寄存器，

stop功能：就是在display功能基础上，停止count寄存器的变动，

reset功能：就是将count寄存器置零。

实验步骤：

1. 按照教程安装好Quartus II软件以及Cyclone V设备支持。
2. 编写代码。
3. 设置引脚
4. 编译
5. 将电路板接入计算机
6. 在programme中，探测设备，将编译得到的sof文件加入。
7. 烧录

实验心得：

本次实验的代码相对而言逻辑十分简单，主要是让我们熟悉了Verilog的基本语法以及软件安装、环境设置还有完成一次测试的基本操作流程。

在实验中，我曾多次烧录不成功，因为流程的疏漏错误，比如忘记布置管线，或者是细节上的错误，比如两个设备的顺序不同。代码也出现过问题，因为寄存器的使用错误，代码无法实现其预想功能。本次实验让我加深了对于整个操作流程的理解和对于其内部运行逻辑的了解以及Verilog语法的熟悉程度。