《数字电路与数字系统实验》实验报告

**实验5: 计数器和时钟**

**姓名:** 尹浚宇

**学号:** 161130118

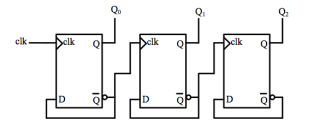
**班级:** 2018-2019第一学期数字电路与数字系统实验2班

**邮箱:** [908664035@qq.com](mailto:908664035@qq.com)

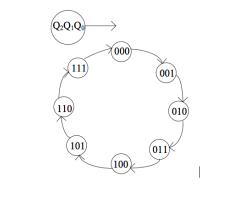
**实验时间:** 2018-10-16

1. **实验目的**
2. 复习计数器的工作原理.
3. 了解几种简单计数器的工作过程和设计方法.
4. 学会使用开发板提供的系统时钟.
5. 掌握计数器的设计和定时器的工作原理.
6. **实验原理**
7. 加法计数器

利⽤触发器可以构成简单的计数器. 下图是由3个上升沿触发的D触发器组成的3位⼆进制异步加法计数器, 即在每个Clock的上升沿, 计数器输出Q2Q1Q0加1.



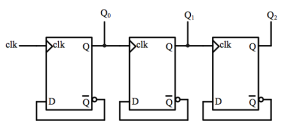
下图是此3位⼆进制异步加法计数器的状态转移图



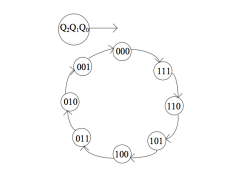
也可以给D触发器加上 “清零” 和 “置数” 端, 构成⼀个可以清零和置数的⼆进制异步加法计数器.

1. 减法计数器

利用D触发器同样可以构成减法计数器. 下图是由3个上升沿触发的D触发器组成的3位⼆进制异步减法计数器.

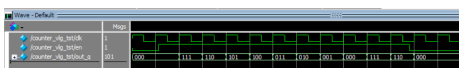


下图是此 3 位⼆进制异步减法计数器的状态转移图



利用Verilog语⾔可以⽅便的构建计数器, 下图就是⼀个3位⼆进制减法计数器的代码和仿真图, 也可以⽤类似的代码构成加法计数器.



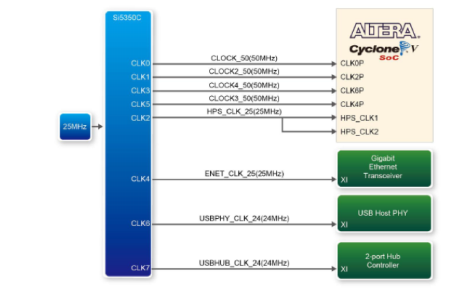


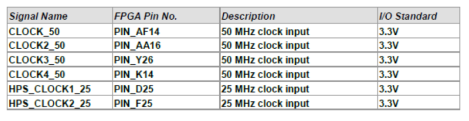
1. 定时器

如果在计数器的时钟输⼊端输⼊⼀个固定周期的时钟, 那么计数器就变成了定时器.

* 1. 开发板上的时钟信号

DE-10 Standard 开发板为 Cyclone V SOC FPGA 提供了四个频率为 50MHz的外部输⼊时钟, 这些时钟均可供⽤户使⽤. 另外还给开放平台上的 HPS 提供了⼀个 25MHz 的时钟, 如下图所示.



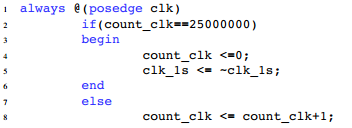


将此时钟信号作为计数器的时钟信号, 即可构成⼀个定时器：

计时时间 = 脉冲个数 × 脉冲周期

* 1. 生成需要的时钟信号

利⽤开发板上提供的频率为50MHz时钟信号和定时器, 我们可以设计任何我们需要的时钟信号. 如下实例是产⽣周期为1秒的时钟信号的参考代码. 其中clk是系统时钟, clk\_1s是产⽣的周期为1秒的时钟.



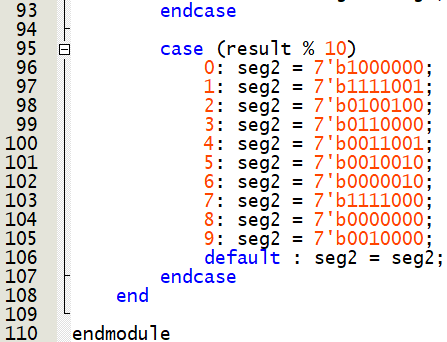
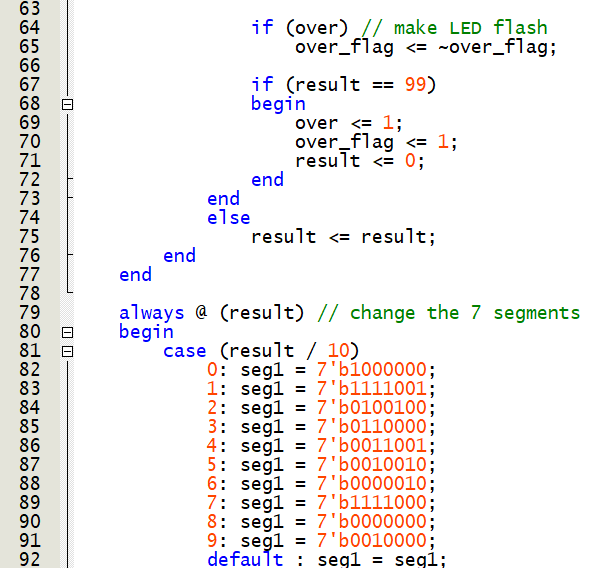
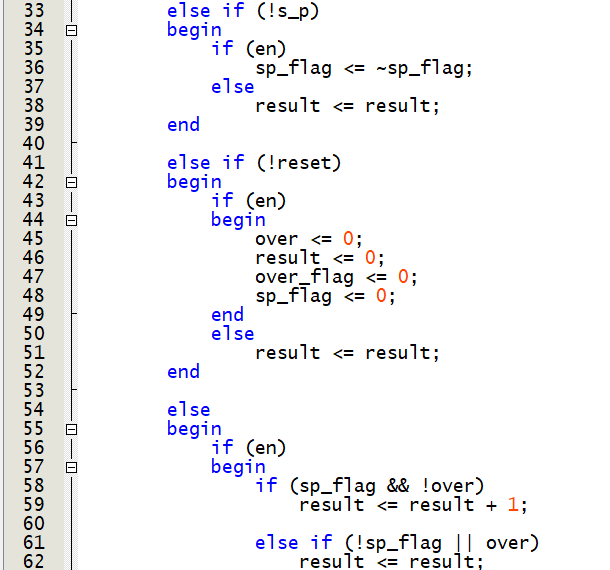
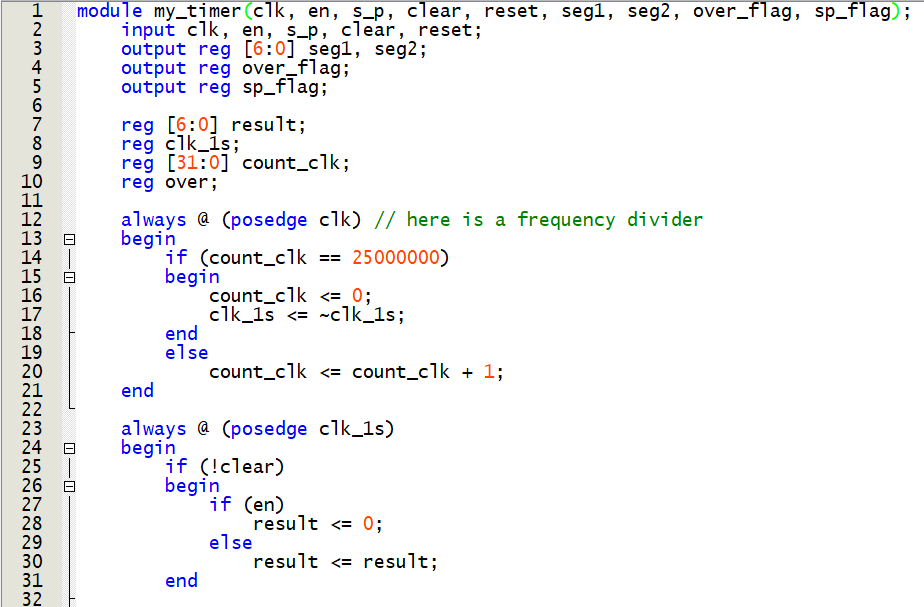
1. **实验环境/器材**

系统环境是window10, 硬件环境是DE10-Standard开发板, 软件环境是Quartus.

1. **程序代码**

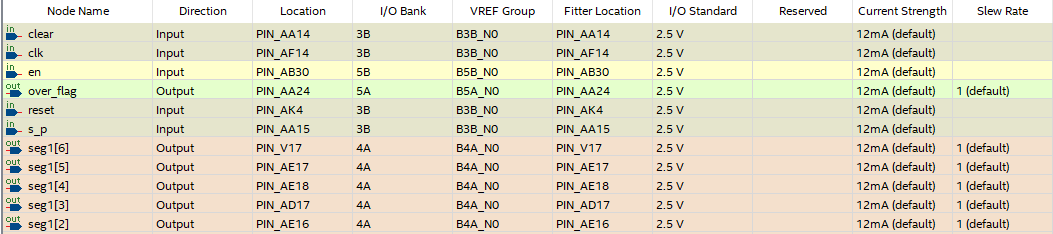
采用同步下降沿触发的方式触发暂停, 清零和复位. 其中clk是系统时钟, 频率为50MHz, en是使能端, s\_p是开始暂停端, clear是清零端, reset是复位端, seg1是

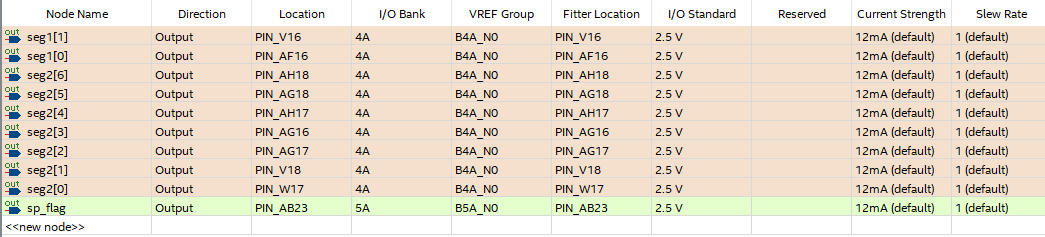
显示十位数字数码管, seg2是显示个位数字的数码管, over\_flag是显示计数结束的二极管, 计数结束后会闪烁, sp\_flag是显示计数状态的二极管, 亮的时候正在计数, 暗的时候为暂停状态. 整个程序分三个always块, 第一个是一个分频器, 用系统时钟产生一个1Hz的时钟信号Clk\_1s, 第二个利用Clk\_1s进行计数, 第三个将结果在数码管上进行显示.



1. **实验步骤**

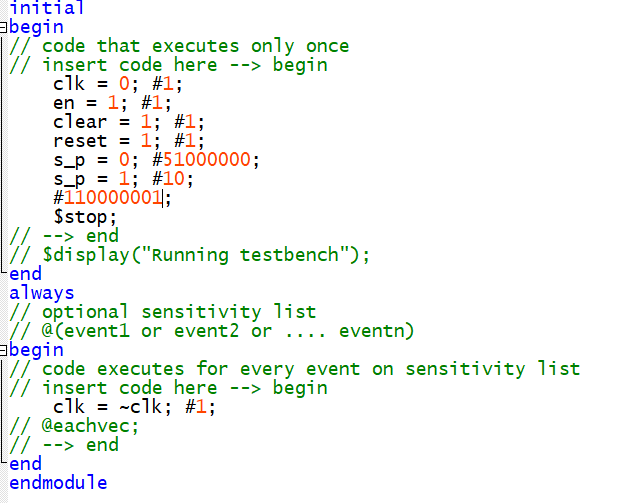
首先设计了一个分频器将开发板的时钟转变为一个频率为1Hz的时钟, 而后利用这个新的时钟编写了计数器(具体代码见四), 通过编译后又设计了仿真模拟的代码(见六), 仿真模拟通过后分配了引脚(如下图), 最后讲生成的二进制烧写文件导入开发板进行硬件验证, 并通过了助教的查验.





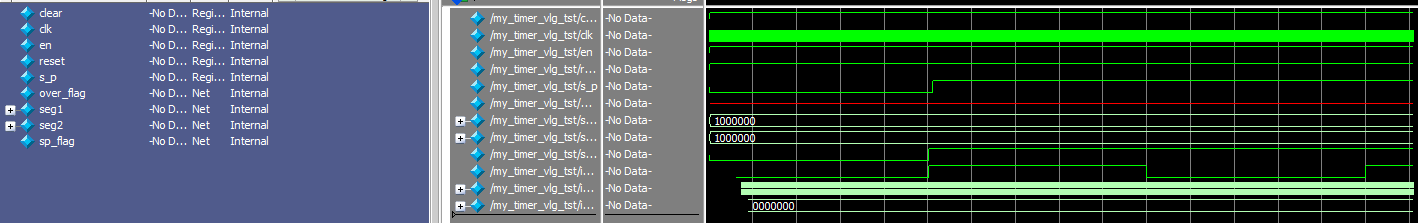
1. **测试方法**

采用仿真验证和硬件验证的测试方法, 首先编写了如下的测试代码(见下图)进行仿真验证(仿真波形图见七), 而后进行了硬件验证(见七的图片和压缩包的视频).

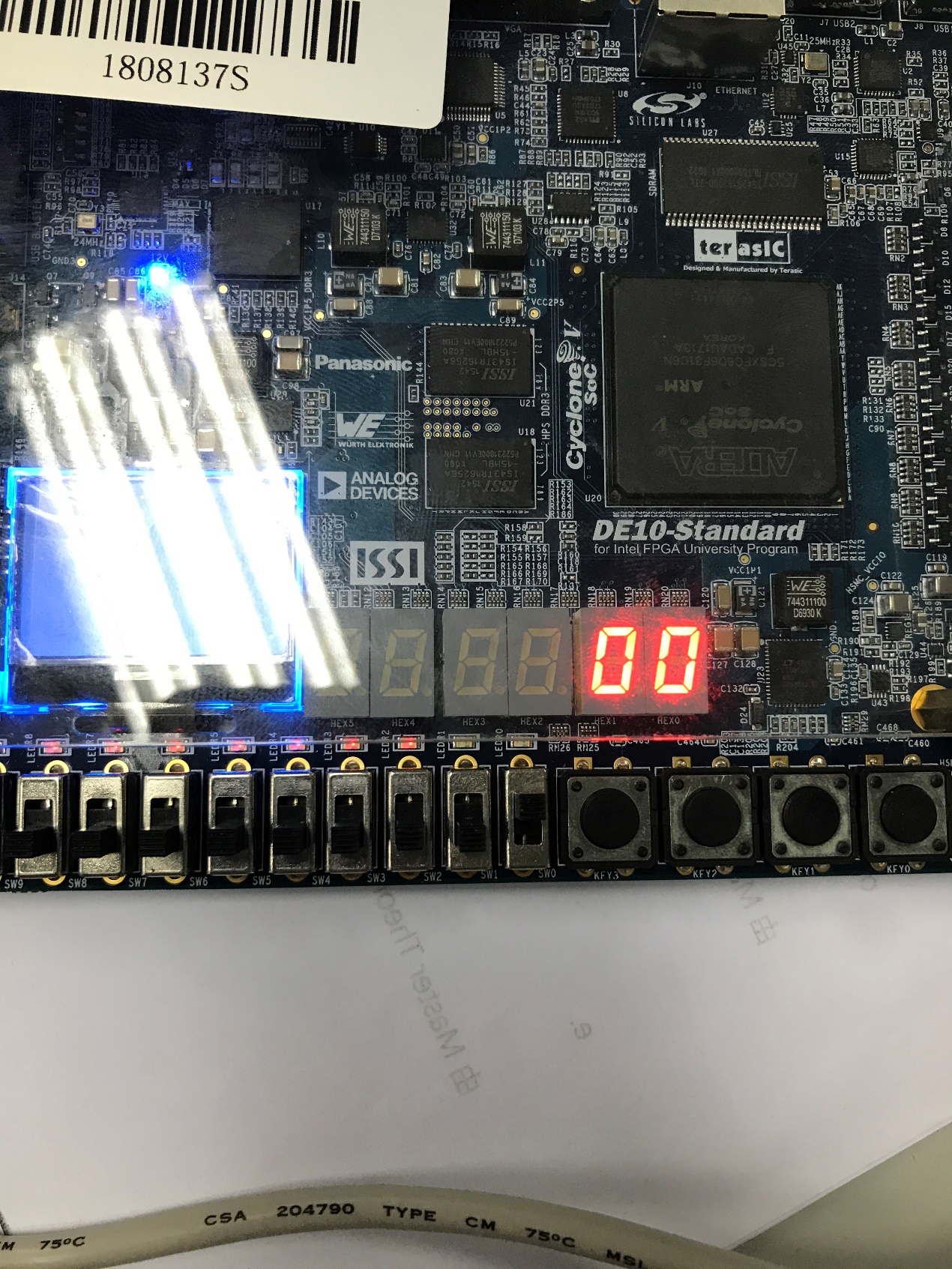
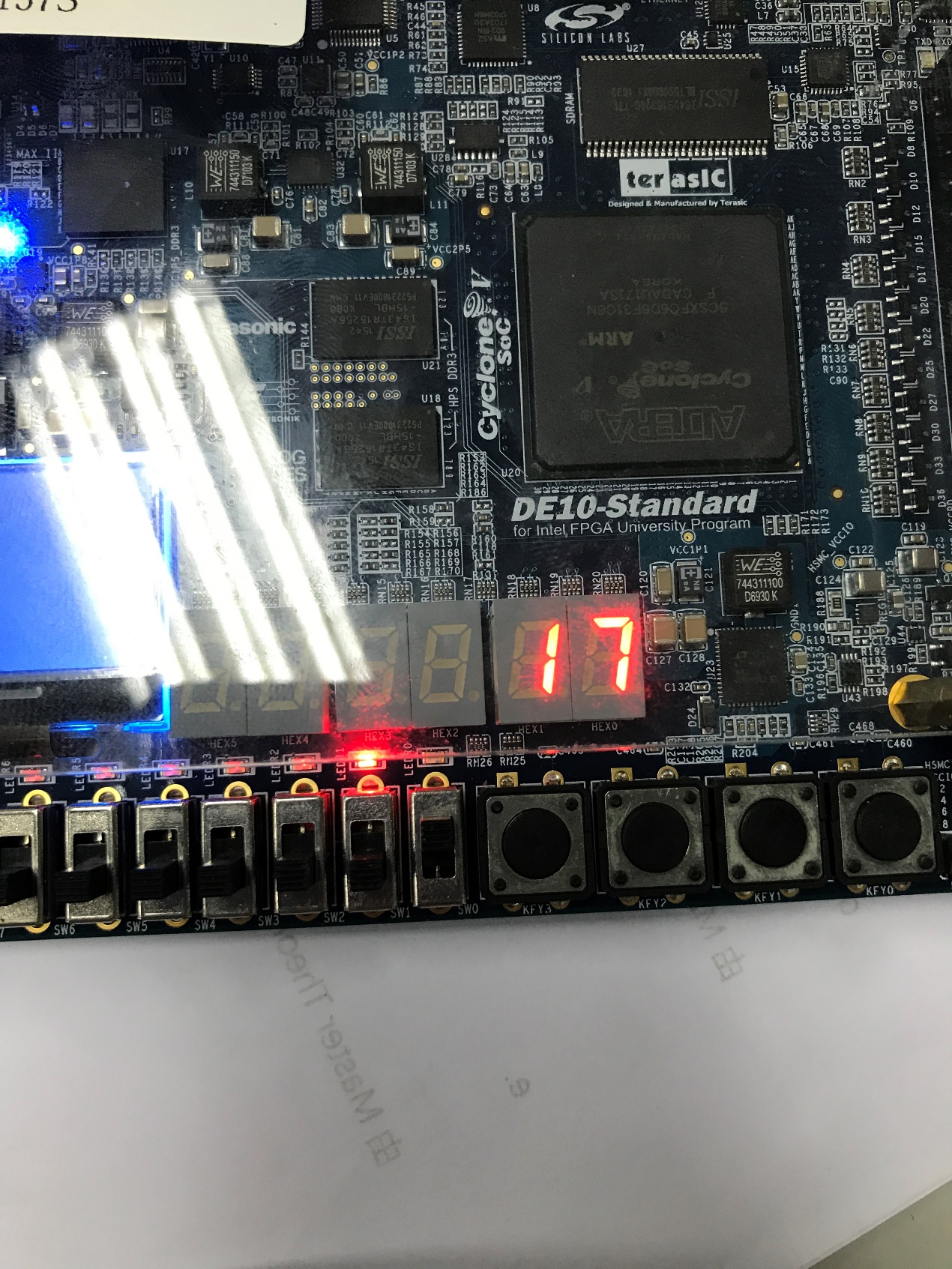
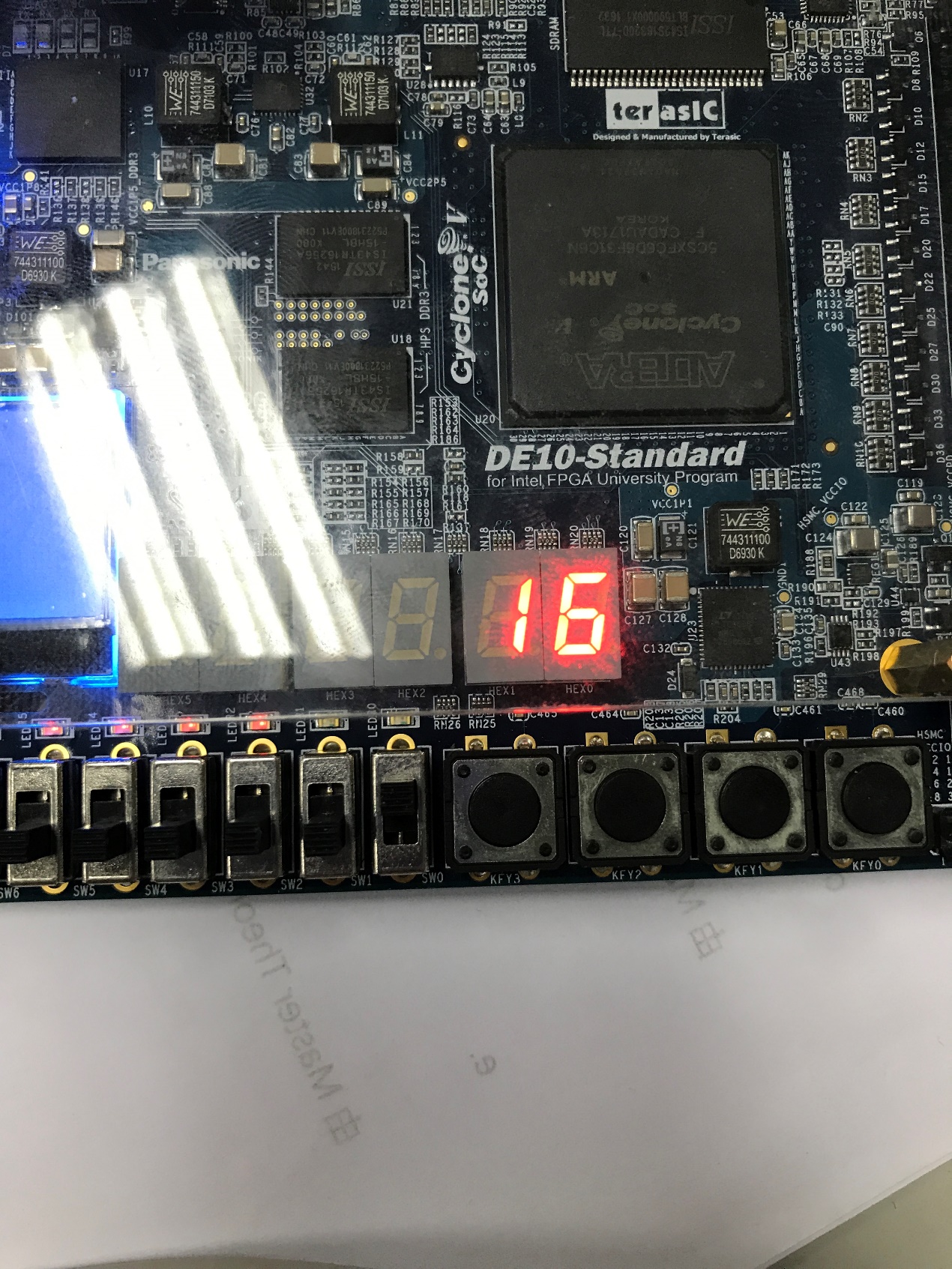
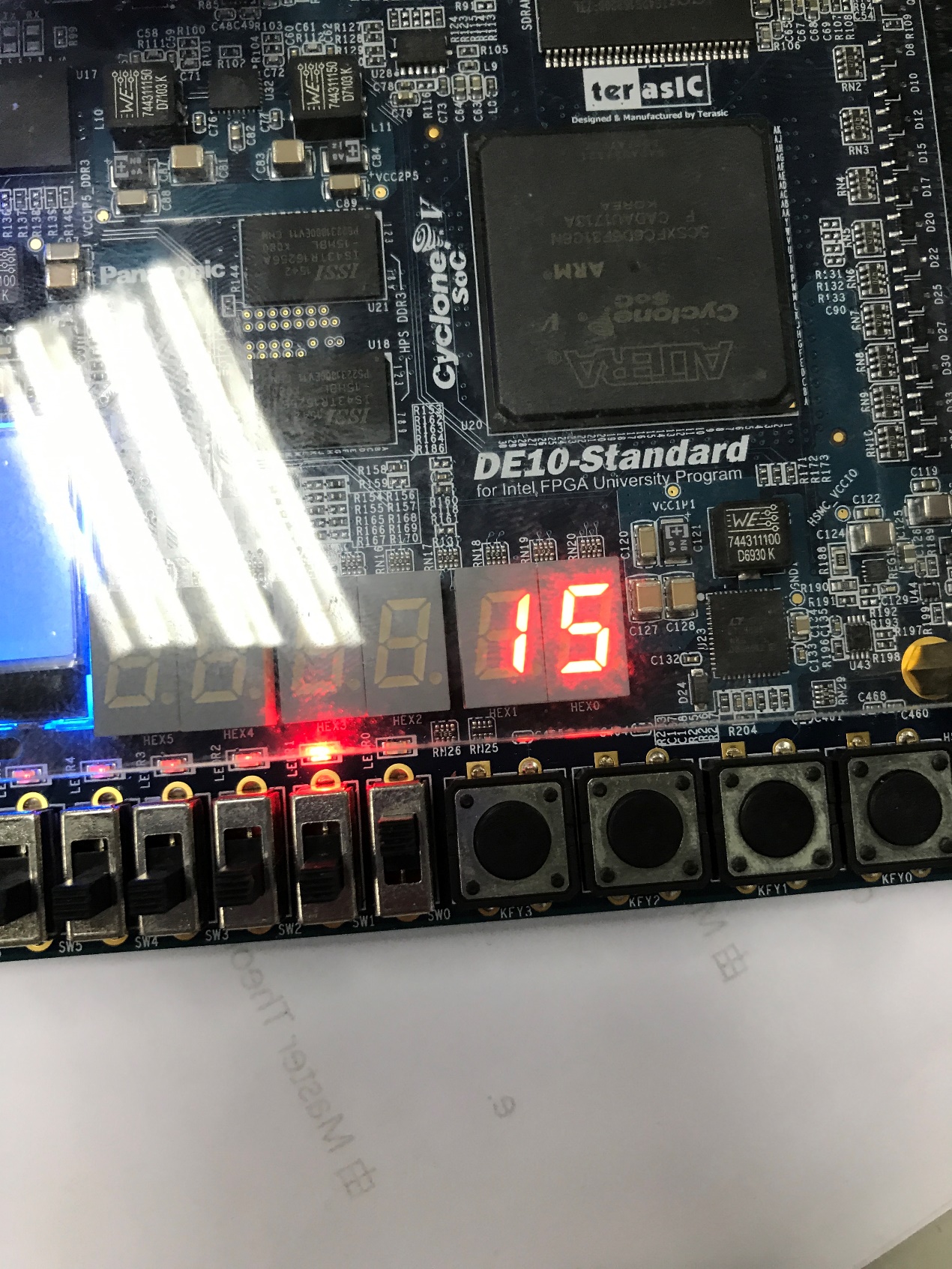
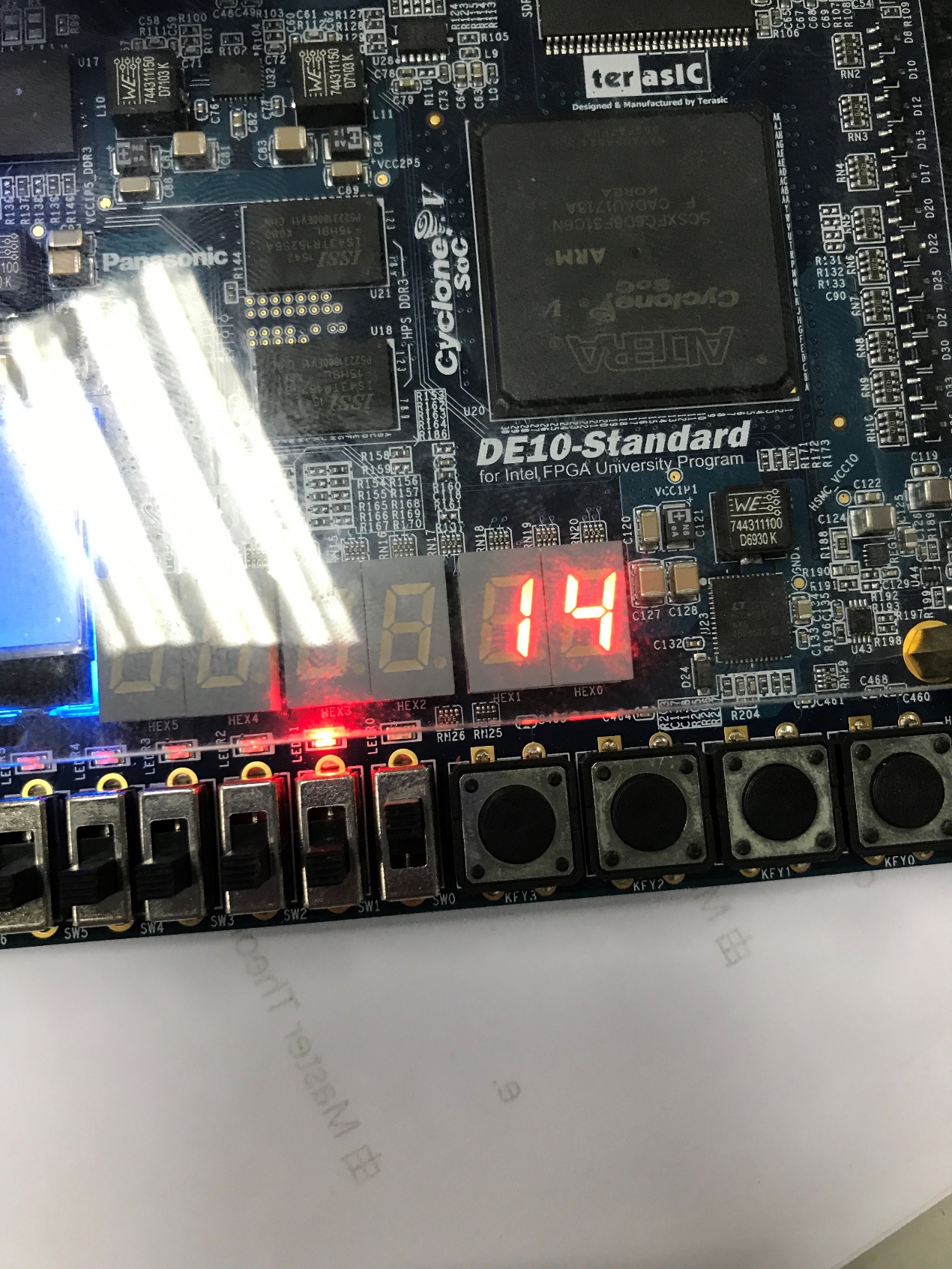
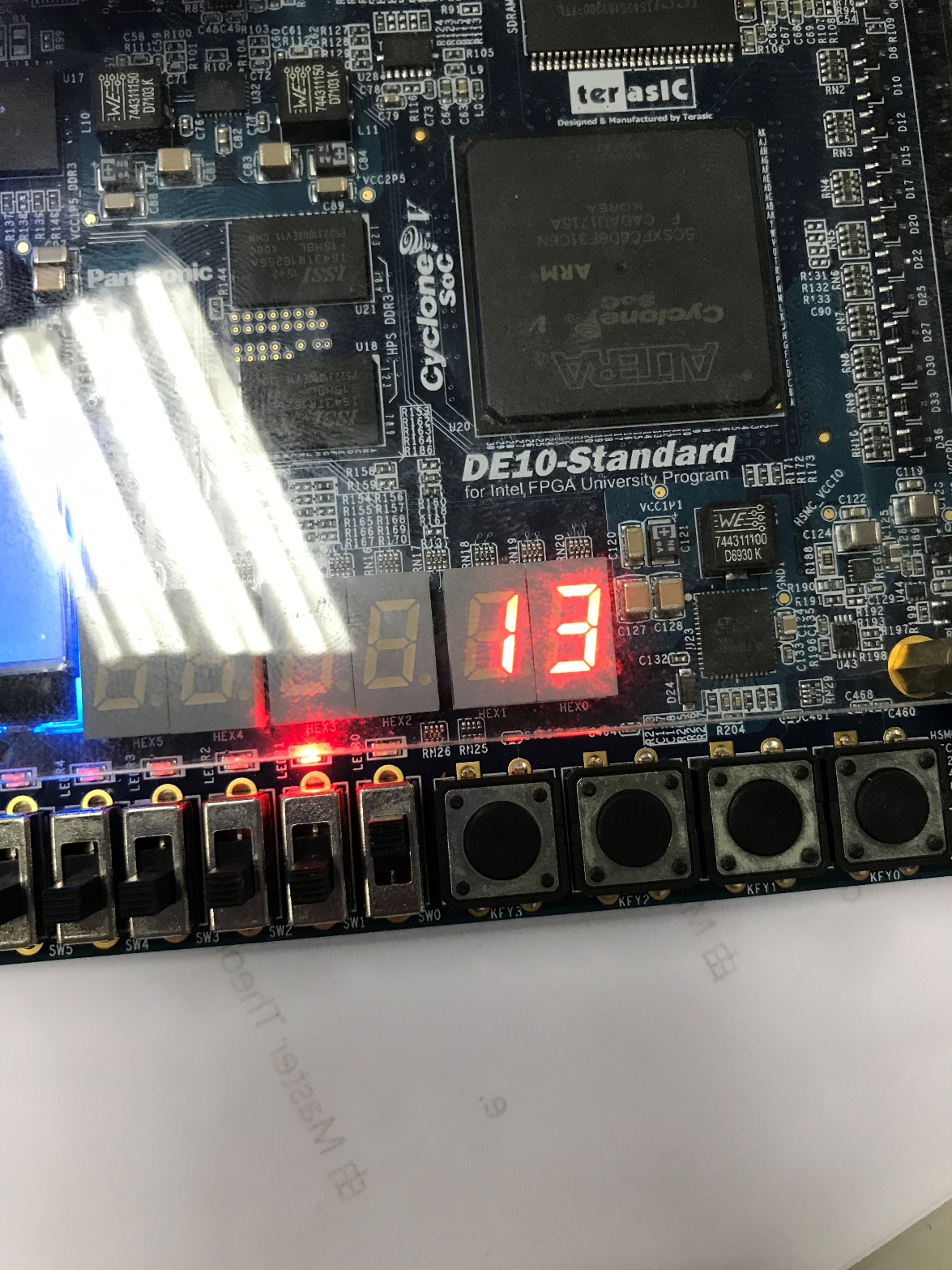


1. **实验结果**

仿真波形图如下:



硬件验证结果如下(更多细节见压缩包内视频):



1. **实验中遇到的问题及解决方案**
2. 开始时分了很多个always模块, 在不同always块里都对一个变量进行了赋值, 结果编译出现了错误, 后面百度后精简了设计, 解决了问题.
3. **实验得到的启示**
4. 不要在多个always块里对同一个变量进行赋值.
5. 利用系统时钟加分频器可以产生满足自己需要的时钟信号.
6. **意见和建议**
7. 可以对拓展实验给出一点指导.