《数字电路与数字系统实验》实验报告

**实验4: 触发器与锁存器**

**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:** 2018-10-09

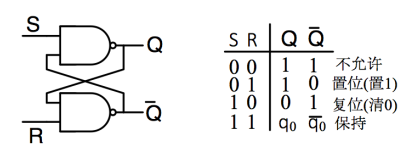
1. **实验目的**
2. 复习锁存器和触发器的工作原理.
3. 复习时序电路中电路时序图的分析和阅读.
4. 学会对时序电路进行仿真.
5. 了解verilog语言中阻塞赋值语句和非阻塞赋值语句的区别.
6. 掌握模块实例化方法, 并在一个工程里完成一个异步清零和一个同步清零的D触发器.
7. **实验原理**

锁存器和触发器是时序电路的基本构件. 锁存器和触发器都是由独⽴的逻辑门电路和反馈电路构成的, 锁存器在时钟信号为有效电平的整个时间段, 不断监测其所有的输⼊端, 此段时间内的任何满⾜输出改变条件的输⼊, 都会改变输出; 触发器只有在时钟信号变化的瞬间才改变输出值.

常见的有RS锁存器、D锁存器、D触发器、JK触发器和T触发器等, 其中最常⽤的是D触发器. 在FPGA器件中, 时序逻辑电路多由D触发器实现.

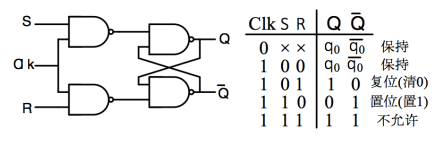
1. RS锁存器

下图是⼀个使⽤与⾮门构成的RS锁存器和其真值表, 根据电路原理图和真值表可以看出, RS锁存器有四种不同的状态. 当R和S同时为0时Q和Q⾮均为1, 这种情况是不允许的, 所以是⽆效状态. 当仅有S为 0时, 输出Q值为1, 当仅有R为0时, 输出Q值为0. 当R和S全为1时, Q值保持原来的值不变. 由电路原理图可以分析出, 只有当R和S全为0时, 此时的锁存器值⽆效, 其他三种状态都是稳定的, 只要输⼊的值不改变, 输出的值也保持不变.



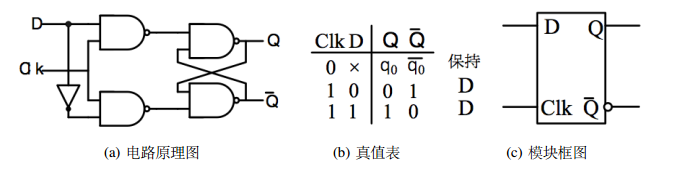
1. 时钟触发的RS锁存器

在RS锁存器的基础上增加两个与⾮门, 就可以构成由时钟信号触发的锁存器. 下图在RS锁存器的基础上增加了两个与⾮门, 可以看出, 当 Clk为0时, 这两个新加的两个与⾮门⼯作在 “关门” 状态输出恒为1. 此时对于RS锁存器⽽⾔处于保持状态, 当Clk为1时, 新加的两个与⾮门⼯作在 “开门” 状态, 此时RS锁存器的⼯作状态同不带时钟使触发端的RS锁存器相同.



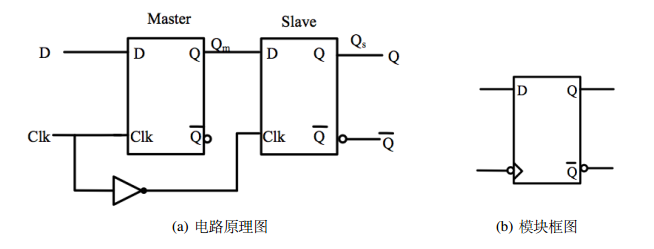
1. D锁存器

RS锁存器中有⼀个Q和Q⾮同时为1的⽆效状态, 这是R和S同时为1的缘故, 如果强制R和S总是相反的逻辑, 就可以避免这⼀现象产⽣. 如下图所示, 这个电路就是D锁存器电路, 当时钟触发信号为0时,输出保持不变, 当时钟触发信号为1时, Q输出D的值, 即Q随着D值的改变⽽改变.



1. 时钟边缘触发的D触发器

前⾯讨论的锁存器都是在时钟为⾼电平(也可以设计为低电平)时触发的, 如果我们希望锁存器只在时钟的特定时刻(如上升沿或者下降沿)触发, 锁存此时刻D的值, 这样的锁存器通常称为时钟边沿触发的触发器. ⽤两个锁存器可以构成触发器, 如下图所⽰. 图中的两个D锁存器,前者为主锁存器, 后者为从锁存器, 当Clk信号为1时, 主锁存器的Qm随着D的变化⽽变化, 从锁存器的状态保持不变. 当时钟信号变为0后,主锁存器的状态不再变化, 从锁存器Qs的状态则跟随Qm状态的变化⽽变化, 由于当Clk=0时, Qm不会发⽣变化, 因此对于外部的观察者⽽⾔, 在⼀个时钟周期内Q只在Clk从1变为0(即时钟的负跳变沿或下降沿)的时候发⽣⼀次变化. 因此, 我们也可以说输出信号Q是在时钟下降沿采集到的输⼊信号D的瞬间值.

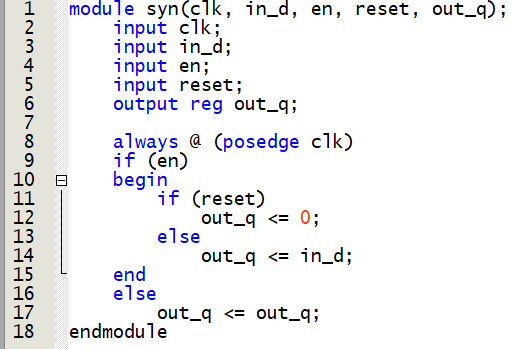


1. **实验环境/器材**

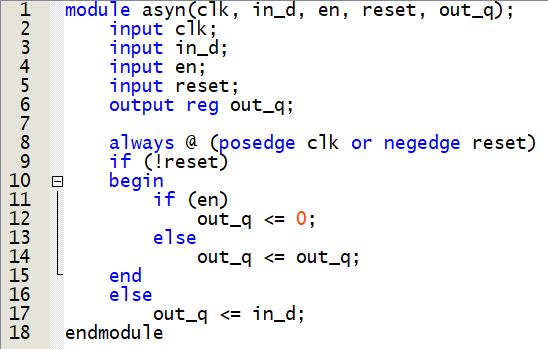
系统环境是window10, 硬件环境是DE10-Standard开发板, 软件环境是Quartus.

1. **程序代码**

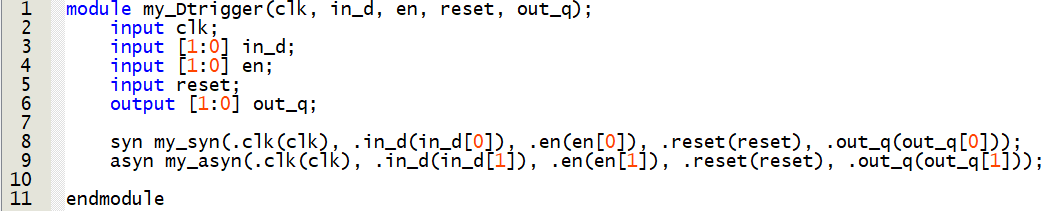
下图是同步清零上升沿触发的触发器代码



下图是异步清零下降沿触发的触发器代码

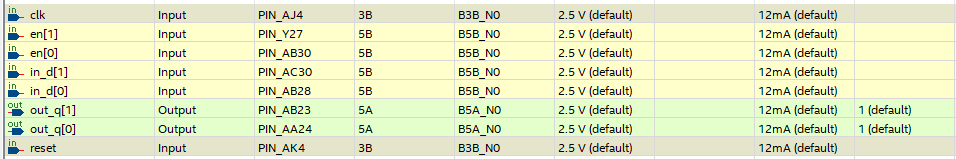


下图是综合两个模块的顶层模块代码



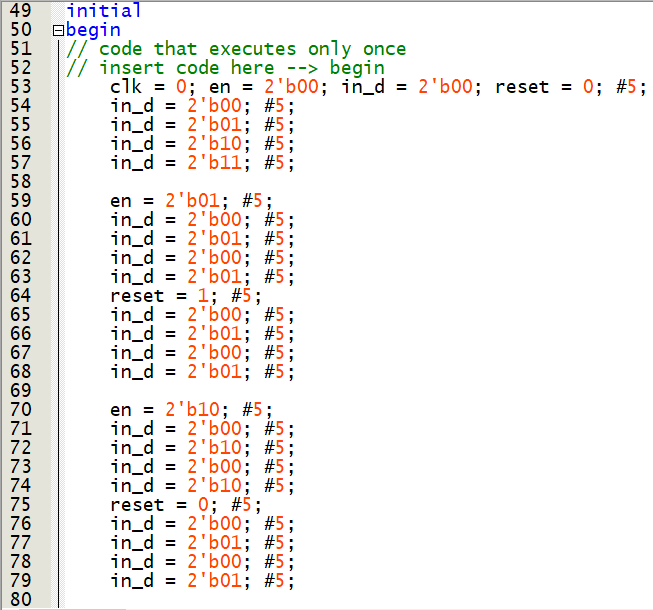
1. **实验步骤**

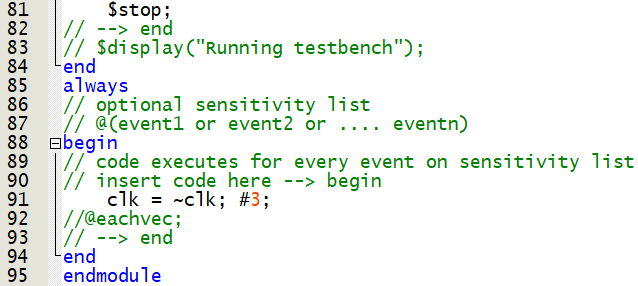
首先分别编写了两个不同触发器的代码, 最后在顶层模块中将它们综合起来(见四), 通过编译后又设计了仿真模拟的代码(见六), 验证通过后分配了引脚(如下图), 最后讲生成的二进制烧写文件导入开发板进行硬件验证, 并通过了助教的查验.



1. **测试方法**

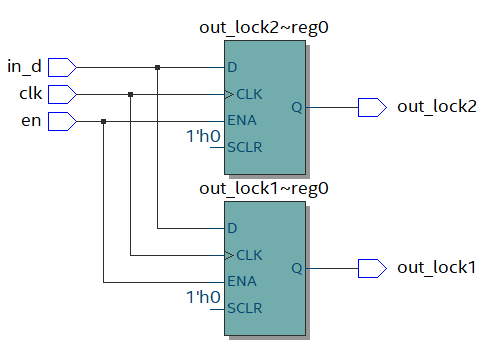
在顶层文件的测试代码中同时测试了两个触发器, 编写了如下的验证代码进行仿真验证(波形图见七的第二部分), 并在开发板上进行了硬件测试(见七的第二部分).

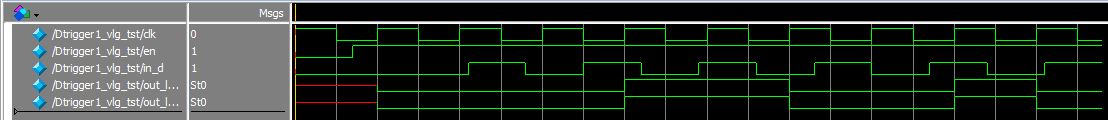




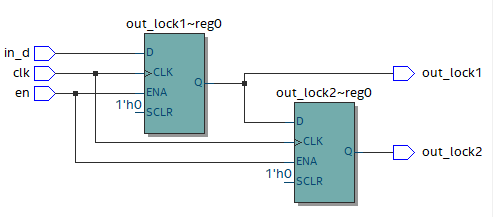
1. **实验结果**
2. 分析阻塞和非阻塞 RTL 视图和仿真结果.

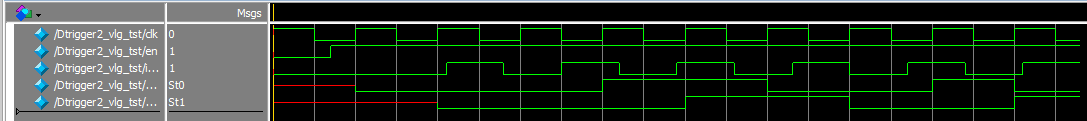
阻塞赋值的RTL视图和仿真波形图如下





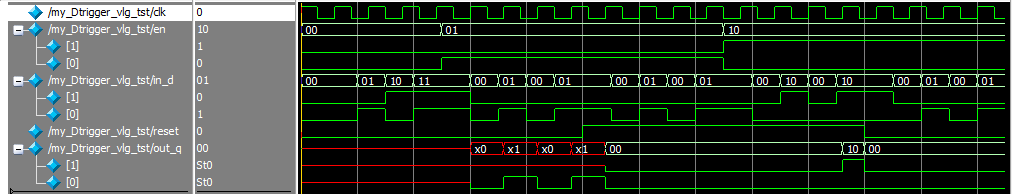
非阻塞赋值的RTL视图和仿真波形图如下





1. 一个同步清零和一个异步清零的 D 触发器

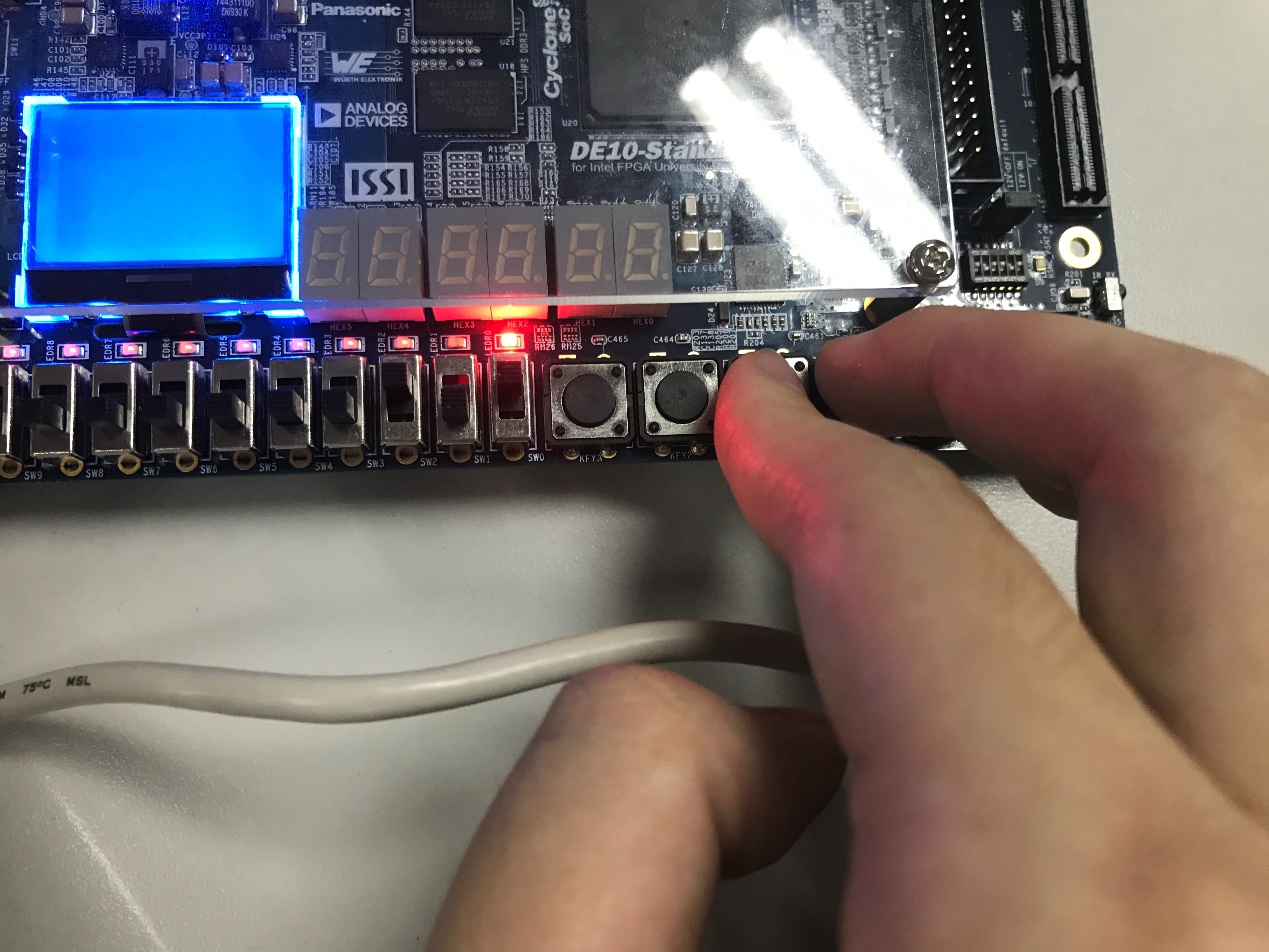
仿真波形图如下



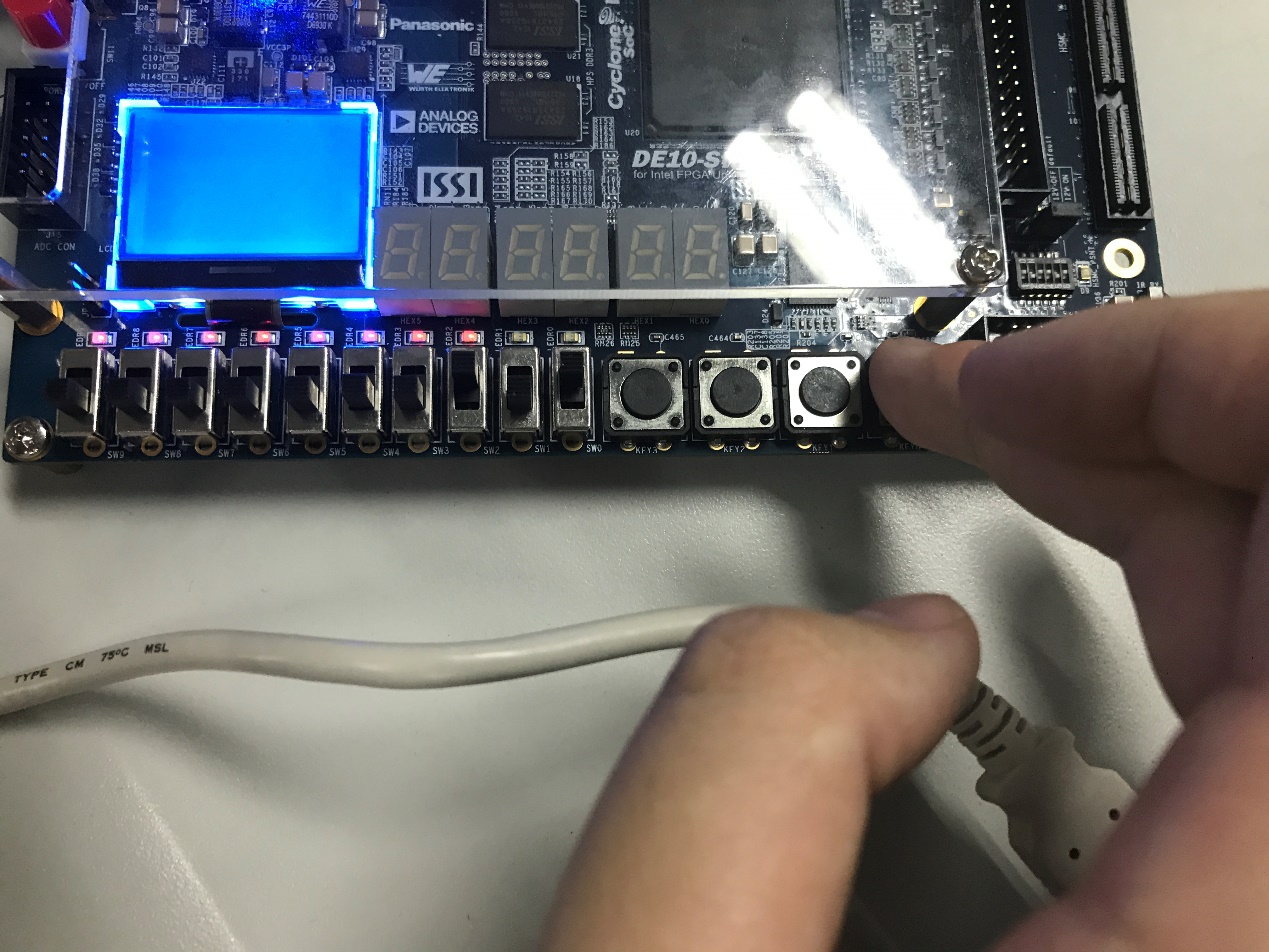
硬件验证结果如下:

button[0]是时钟端, button[1]是清零端. SW0和SW1分别是同步和异步清零触发器的使能端, SW2和SW3分别是上述两个触发器的输入, 输出分别展示在LED0和LED1上.

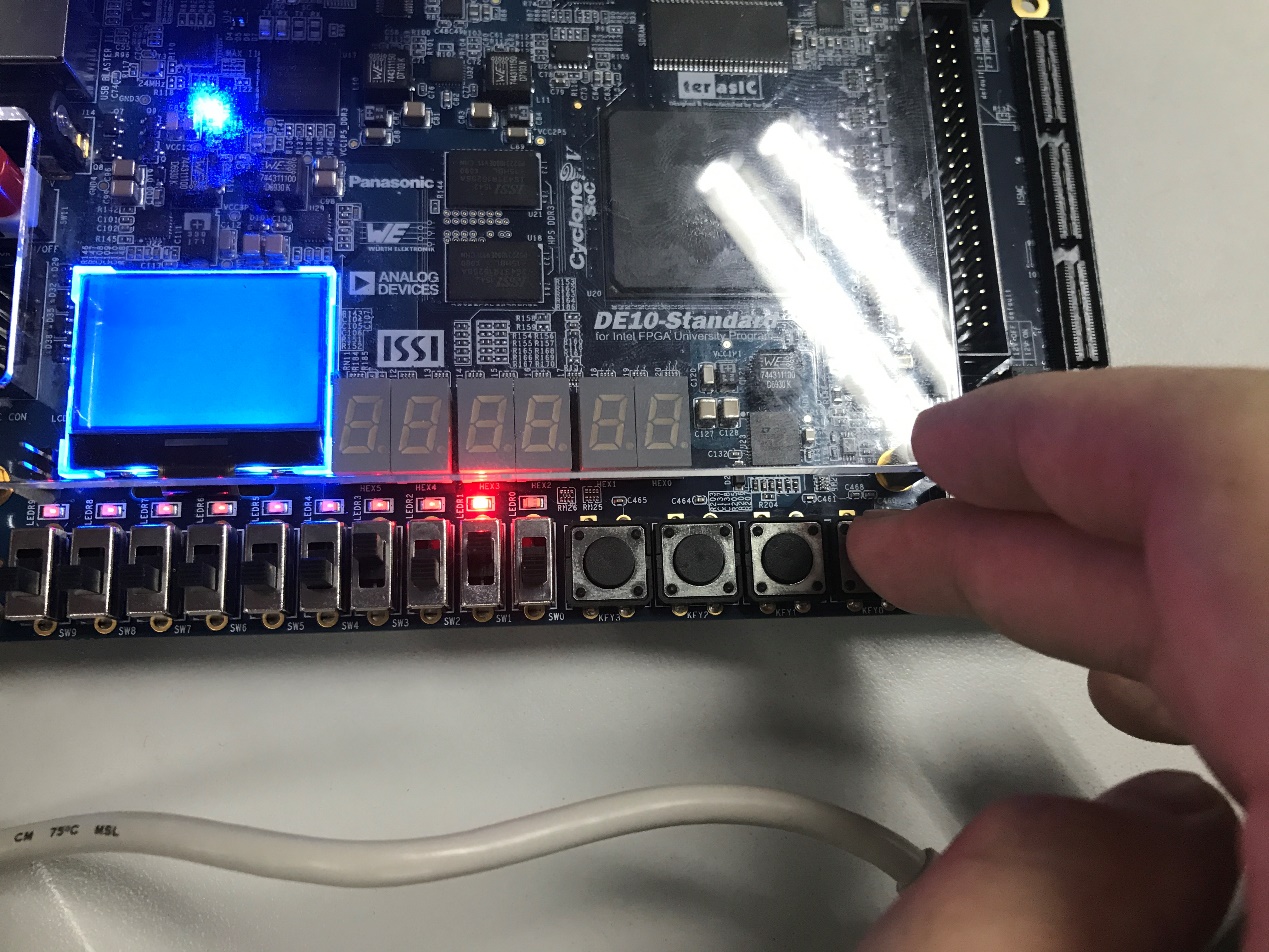
此时同步清零上升沿触发的D触发器生效, 按下清零端使其为0, 再弹起时钟端, 输出置为1.



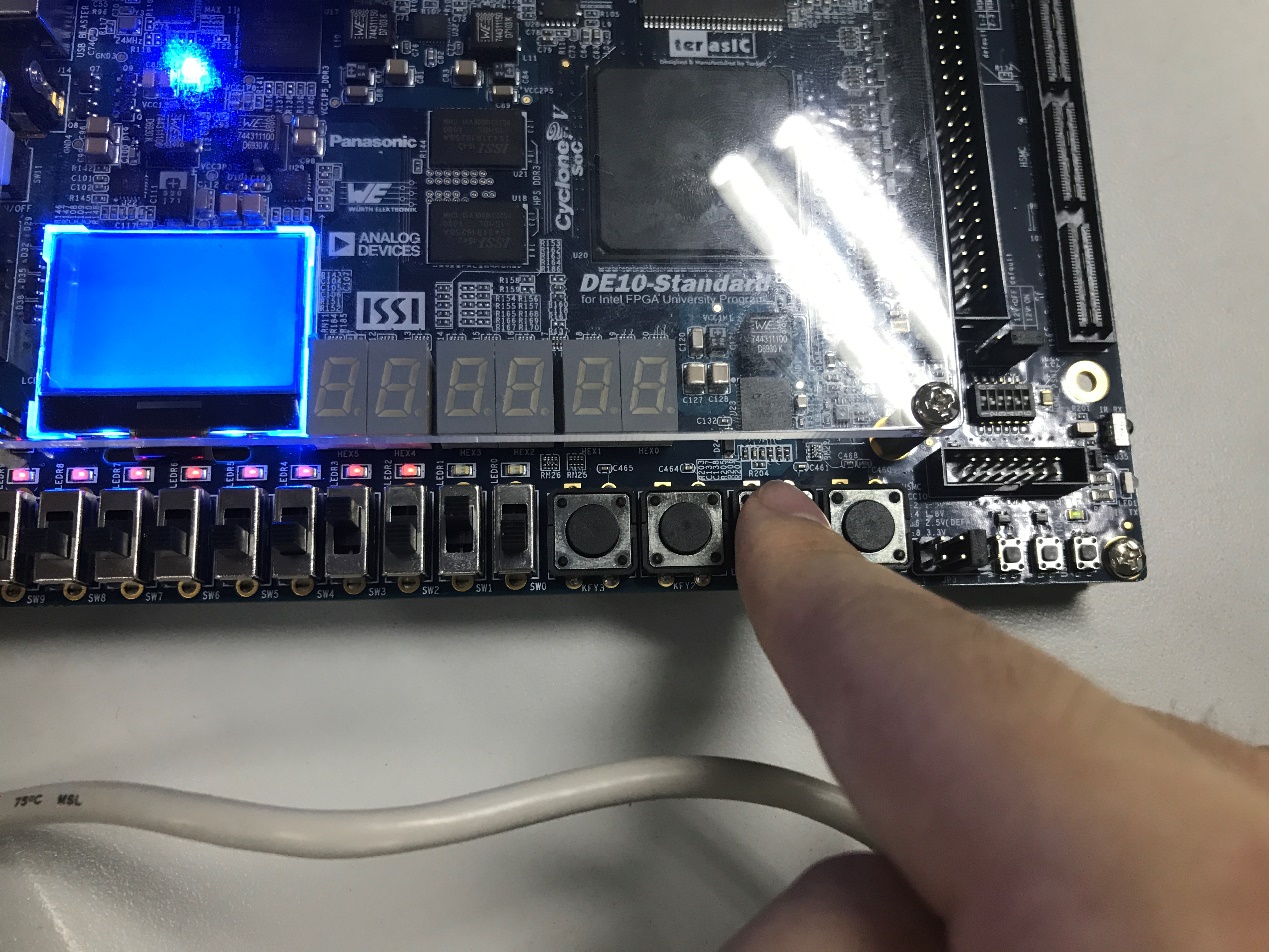
此时松开清零端, 再松开时钟端给出一个上升沿的时钟信号, 方可将输出清零.



此时异步清零下降沿触发的D触发器生效, 清零端保持为1, 再弹起时钟端, 输出置为1.



按下清零端, 无需等待时钟信号, 直接将输出清零.



1. **实验中遇到的问题及解决方案**
2. 不了解同步清零和异步清零的区别, 上网搜索后解决.
3. 不知道怎么调用写好的模块, 上网搜索后解决.
4. 在编写测试文件时, 发现无论将哪一个文件设为顶层, 生成的测试文件都是以第一次创建的.v文件为模版的测试文件, 网上查阅无果后, 经过与同学的讨论, 发现将某一文件设为顶层后, 还需要进行一次Analysis & Synthesis, 生成的测试文件才是新设置的顶层文件的模版.
5. **实验得到的启示**
6. 每个模块分文件设计, 最后在顶层模块里综合是一种良好的编程风格, 提高了工程可读性和模块的可重用性, 今后的设计应该多采用这种方法.
7. **意见和建议**
8. 对于实验内容4.3.1, 讲义前面给出了太多细节, 希望可以只给代码, 让同学自己去探索结果.