北京工业大学软件学院

2021 - 2022学年 第一学期

数字逻辑实验报告

姓名： 黄嘉祺

学号： 20080216

班级： 200802

指导老师： 贺国平

|  |  |  |  |
| --- | --- | --- | --- |
| 课程名称： | 数字逻辑 | | |
| 报告性质： | □作业报告 实验报告 | | |
| 学号： | 20080216 | 姓名： | 黄嘉祺 |
| 任课教师： | 贺国平 | 课程性质： | 学科基础必修课 |
| 学分： | 3 | 学时： | 48 |
| 班级： | 200802 | 成绩： |  |
| 小组成员： | 李可馨 | | |
| 教师评语： |  | | |

2021年 12 月 13 日

**实验一：HelloWorld**

1. **实验目的：**

1、在实验室做实验的同学、熟悉实验环境，测试一个简单例子, 将 程序烧写到开发板，使用 led 灯来显示例子是否正常运行。

2、由于疫情原因无法到校的，熟悉实验环境，测试一个简单例子，跑仿真 程序，观测波形是否正确。

1. **实验内容：**

（一）、在开发板进行实验

下载并安装vivado 2018.2软件，并熟悉它的使用，打开project 选择lab0，点击 Run Synthesis 弹出对话框, 在对话框中点击 OK。等待，直到出现 Synthesis successfully completed

点击 Run Implementation 弹出对话框, 在对话框中点击 OK. 等待，直到出现 Implementation successfully completed

点击 Generate Bitstream 弹出对话框, 在对话框中点击 OK. 等 待，直到出现 Bitstream Generation successfully completed.

连接开发板，先将开关置于OFF状态

连接好开发板之后。点击 Open Target，选择 Auto Connect。

连接成功之后，点击 Program device，在对话框中点击 Program

观察板子红色亮灯情况。将 Switch0 和 Switch1 推到上方, 观察 Led0 与 led1 是否亮。

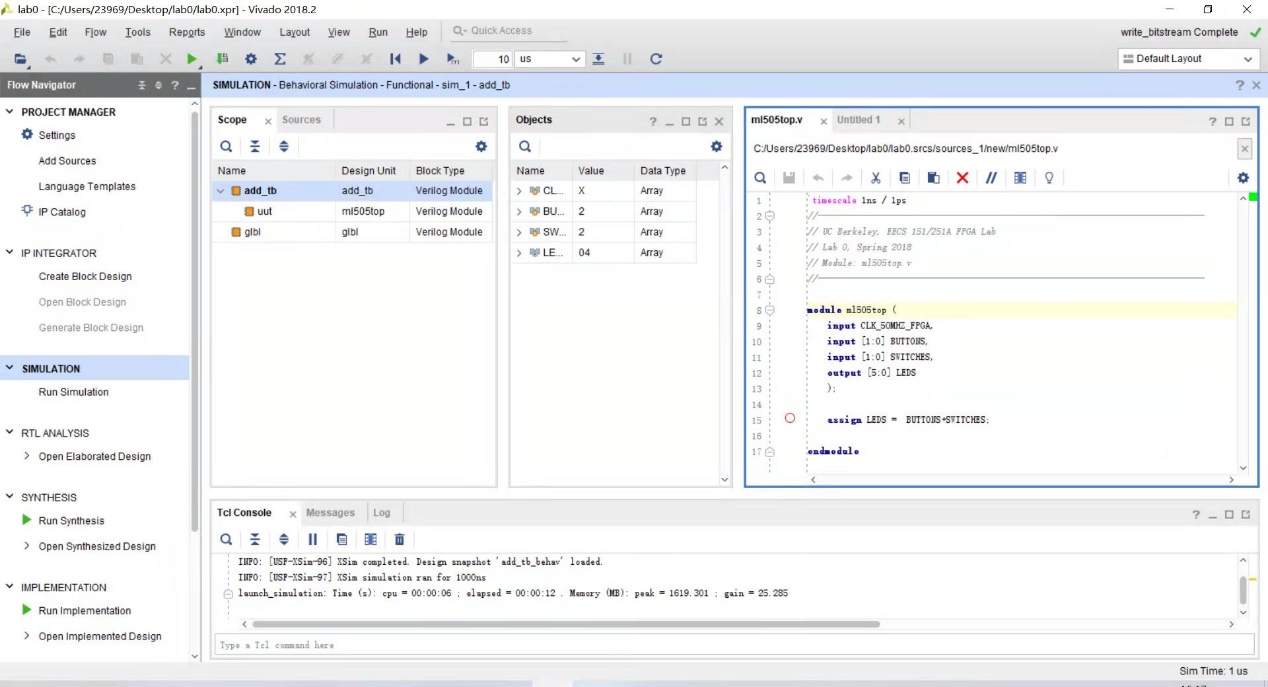
按下 BTNU(R18), 观察 Led0 不亮 led1 不亮 led2 亮。

（二）、跑仿真程序

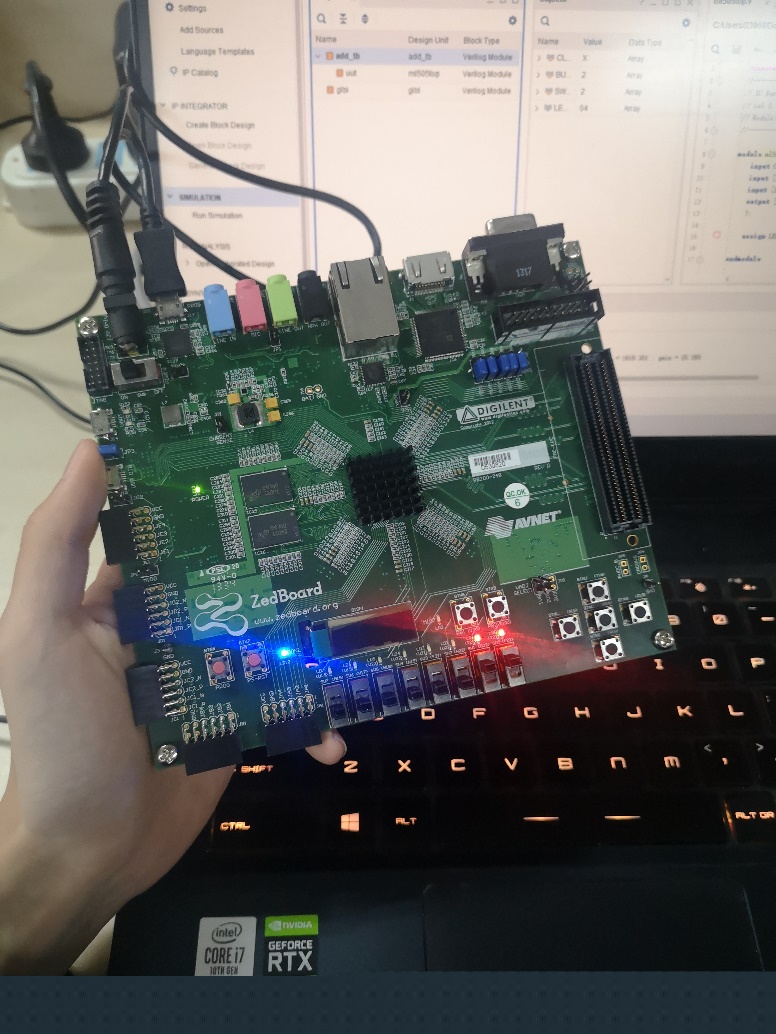
点击 Run Simulation 下的 Run Behavioral Simulation

点击 YES

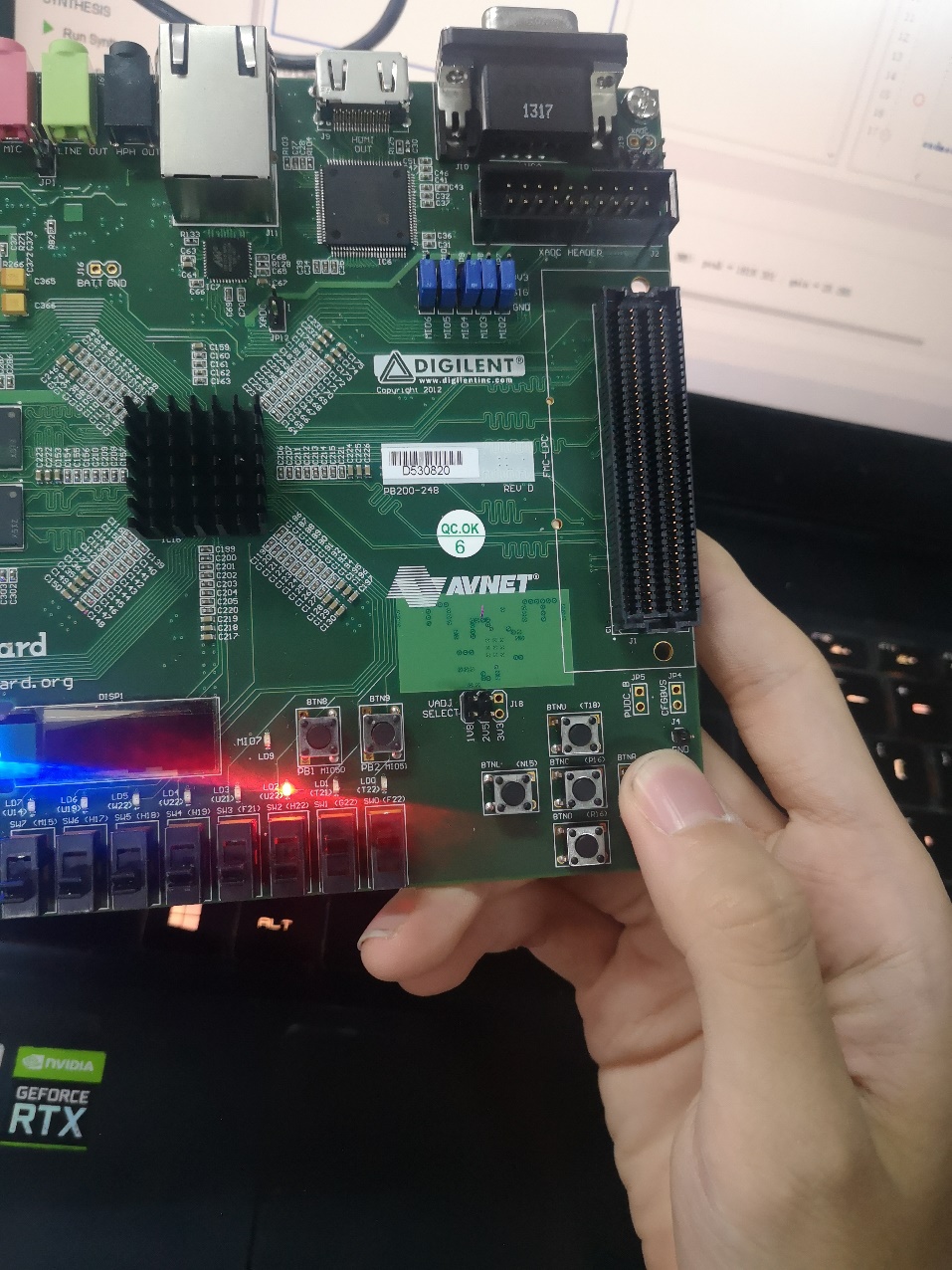
可以看到仿真结果

1. **实验代码（除框架）：**
2. **实验结果：**

Switch0 和 Switch1 推到上方, 观察 Led0 与 led1 是否亮。

****

按下 BTNU(R18), 观察 Led0 不亮 led1 不亮 led2 亮

****

1. **实验总结：**

这个实验实现了代码和电路板的连接，通过编写代码，程序烧写到开发板，将电脑和电路板相连，switch和led组成电路，控制不同的开关，led亮灯情况不同，使用 led 灯来显示例子是否正常运行

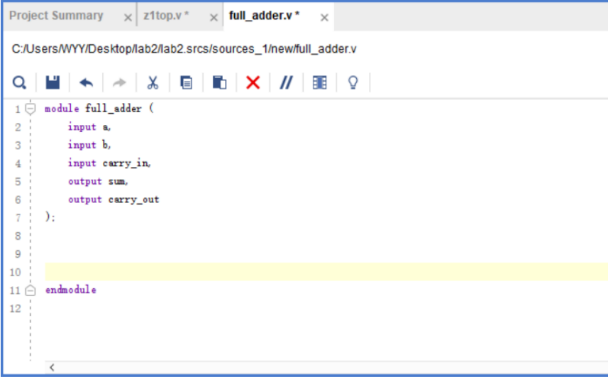
**实验二：全加器实验**

1. **实验目的：**

(1) 设计一个全加器;(2) 基于全加器设计 14 位行波进位加法器

1. **实验内容：**

（一）、构建全加器

1.框架文件已经写好，只需将 full adder.v 文件，利用已有的输入 与输出，将逻辑实现完成。

2.开发板实验结果显示如下:

BTNU(T18)、BTNU(N15)、BTNU(P16) 按任意一个，LED0（T22）会亮

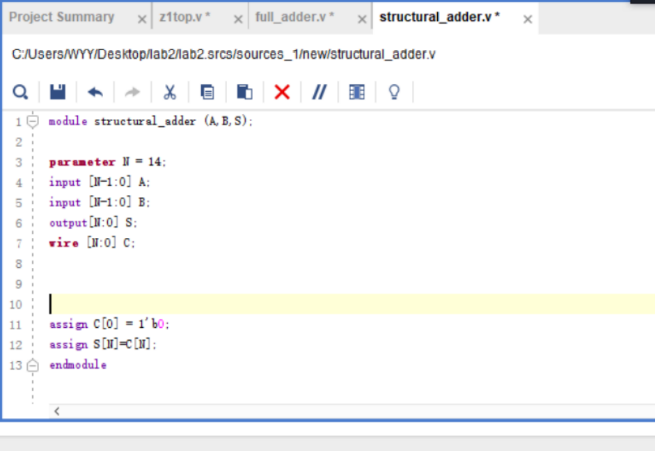
TNU(T18)、BTNU(N15)、BTNU(P16) 按任意两个，LED1（T21）会亮

BTNU(T18)、BTNU(N15)、BTNU(P16) 全按，LED0（T22）和 LED1（T21） 都会亮

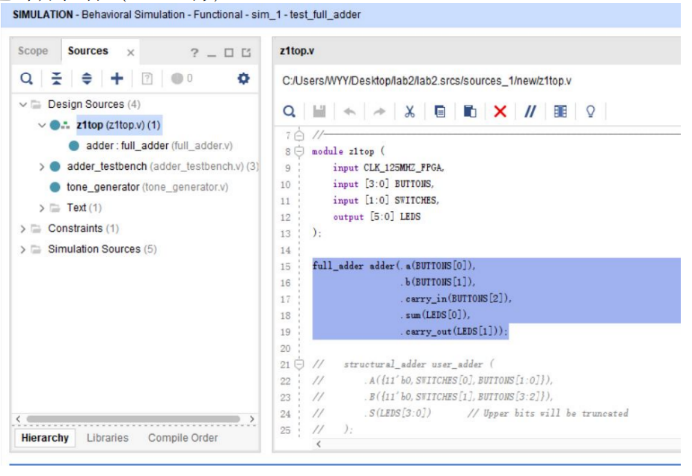
3.仿真实验结果

（二）、构建结构化 14 位加法器

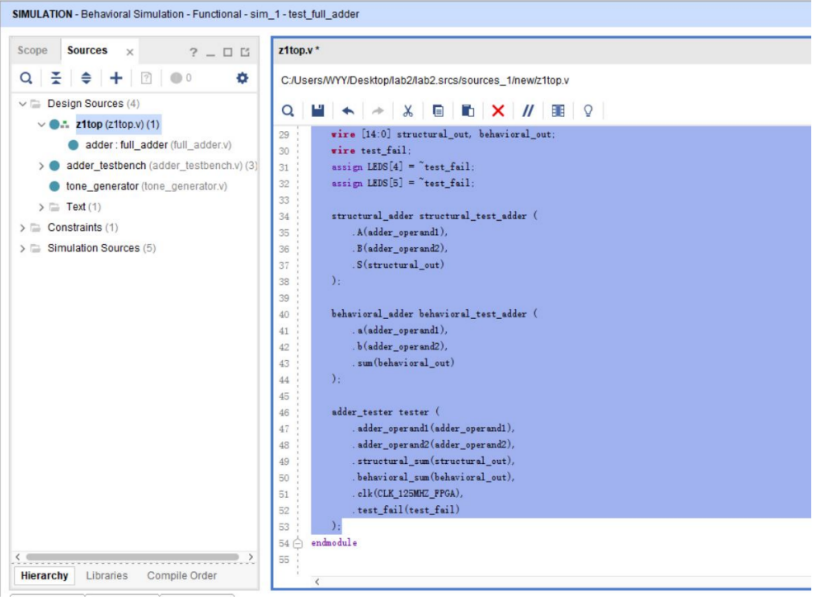
1.利用已经实现的全加器，设计 14 位行波进位加法器。 structural adder.v 文件为实现 14 位进位加法器的文件，利用已有的输入与输出，将逻辑实现完成



2. 开发板实验结果显示如下: 将蓝色部分注释（15-19 行）



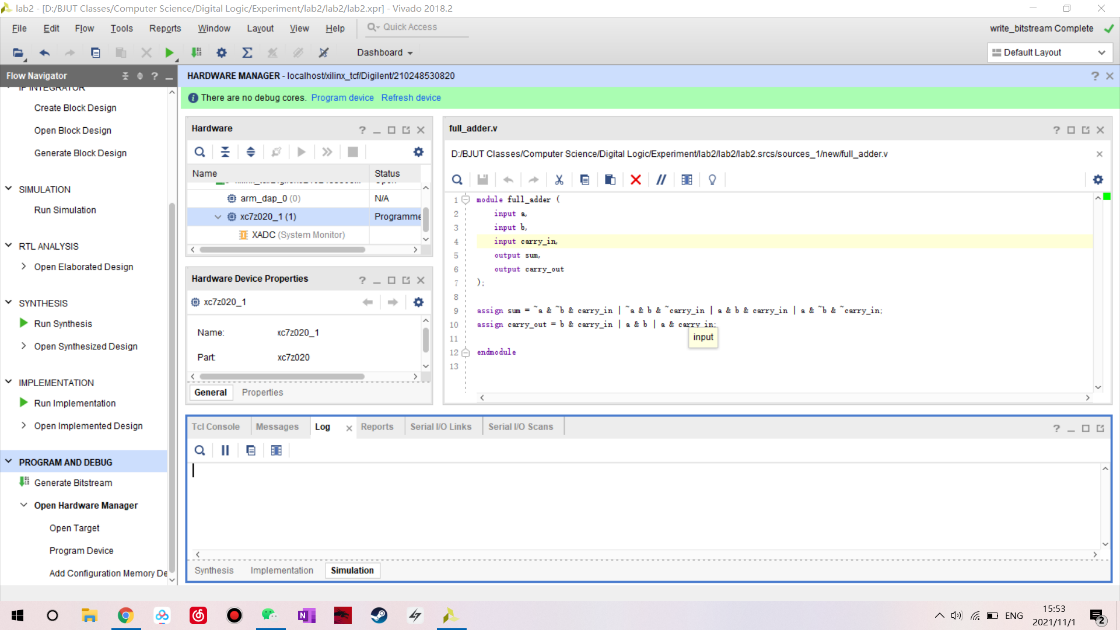
3.将蓝色部分解注（21-52 行）

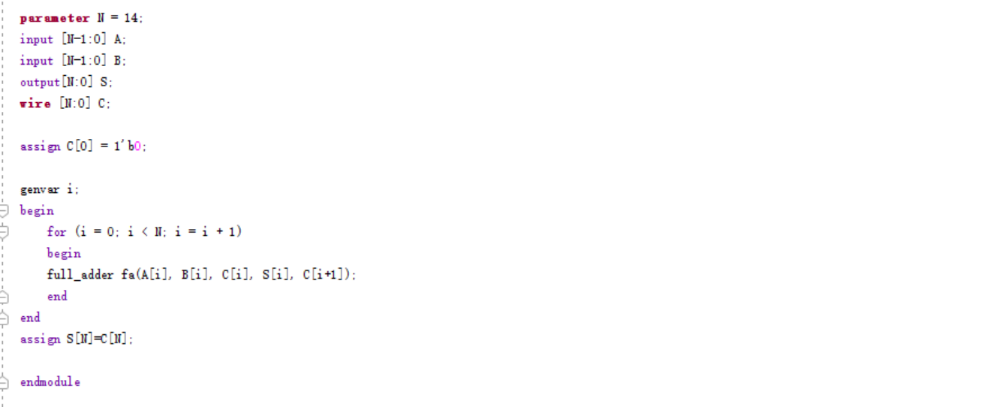


4.程序烧写进开发板，后 LED4（V22）和 LED5（W22) 直接亮起。说明程序写的没有错误

5.把 Switch0 和 Switch1 推到上方，LED3(U21)、LED4（V22）和 LED5 （W22) 会同时亮

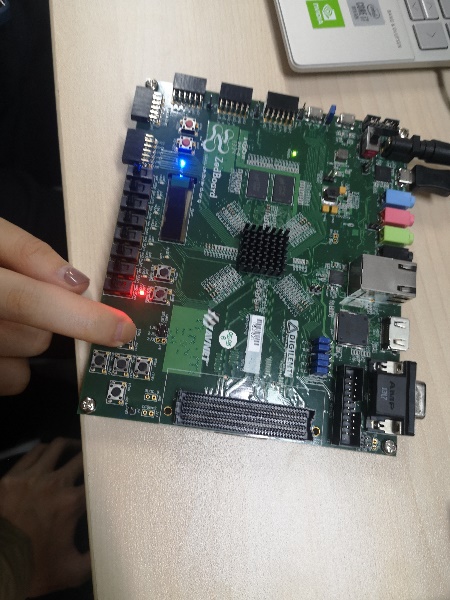
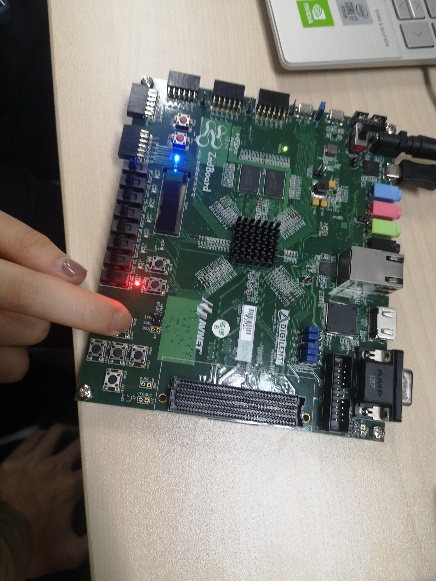
6.把 Switch0 推到上方, 同时按 BTNU（P16）和 BTNU（R18），LED0 （T22）、LED1（T21）、LED2(U22)、LED4（V22）和 LED5（W22) 会同时亮

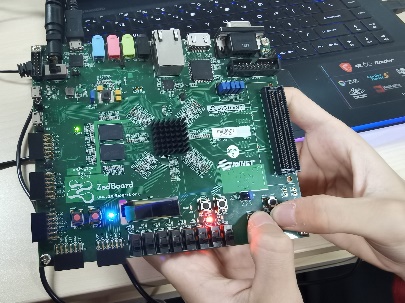
1. **实验代码（除框架）：**

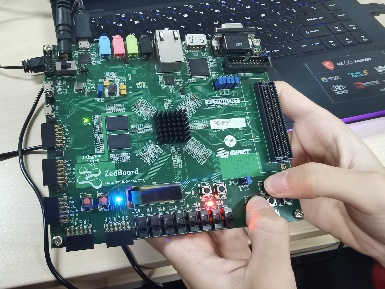
****

1. **实验结果：**

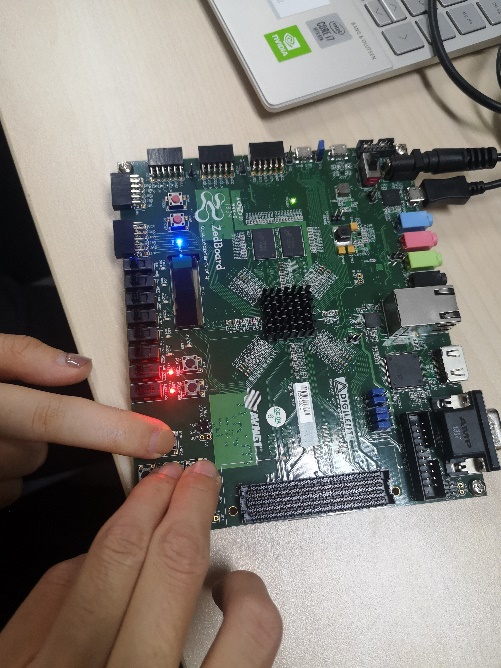
BTNU(T18)、BTNU(N15)、BTNU(P16) 按任意一个，LED0（T22）会亮

****

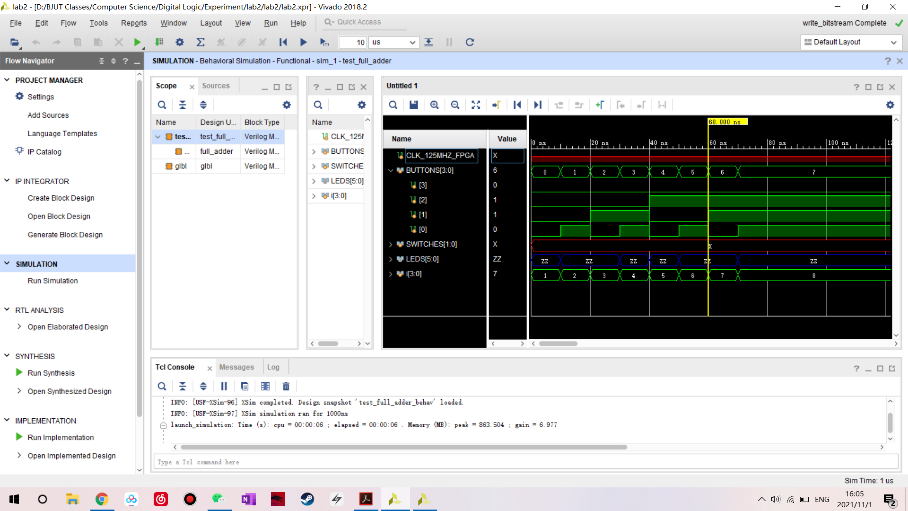
TNU(T18)、BTNU(N15)、BTNU(P16) 按任意两个，LED1（T21）会亮



BTNU(T18)、BTNU(N15)、BTNU(P16) 全按，LED0（T22）和 LED1（T21） 都会亮

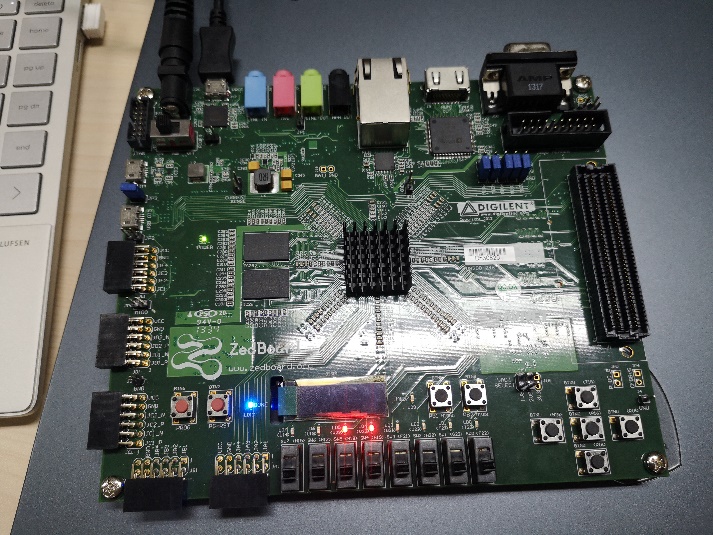
****

仿真实验结果

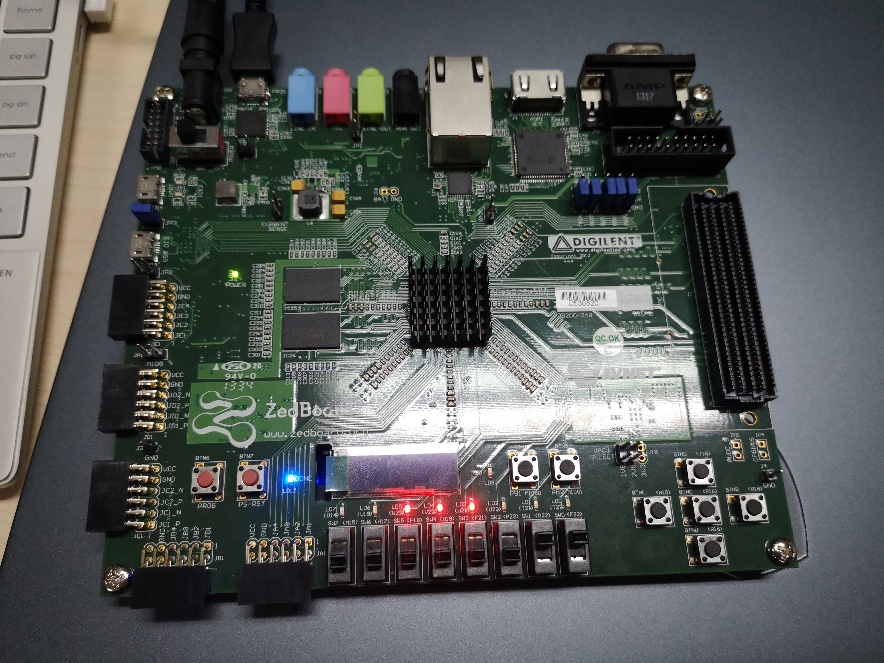
****

14 位行波进位加法器

程序烧写进开发板，后 LED4（V22）和 LED5（W22) 直接亮起

****

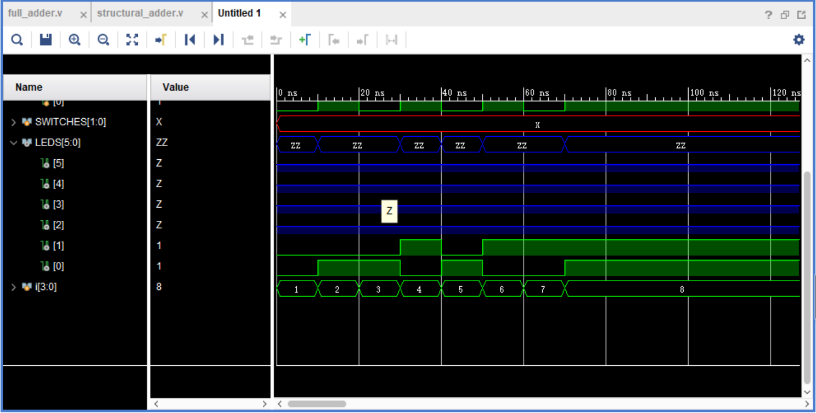
把 Switch0 和 Switch1 推到上方，LED3(U21)、LED4（V22）和 LED5 （W22) 会同时亮

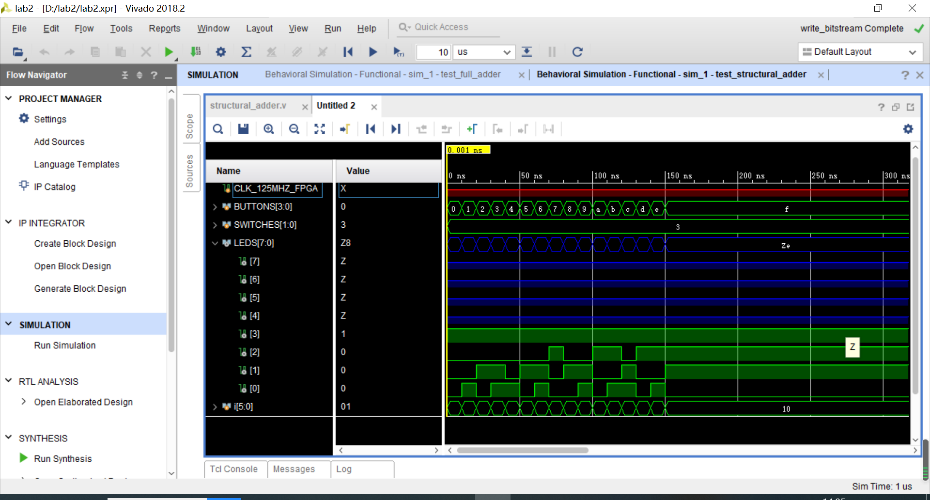
****

把 Switch0 推到上方, 同时按 BTNU（P16）和 BTNU（R18），LED0 （T22）、LED1（T21）、LED2(U22)、LED4（V22）和 LED5（W22) 会同时亮

****

仿真

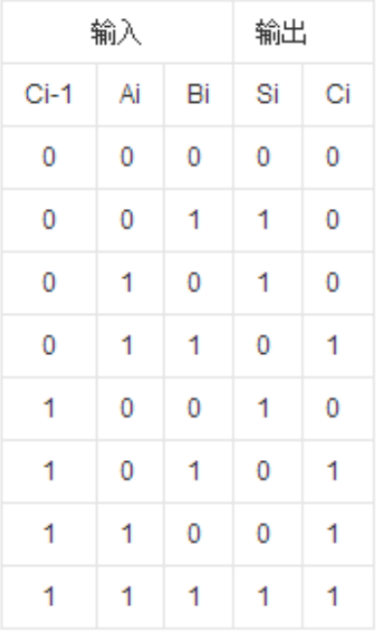


****

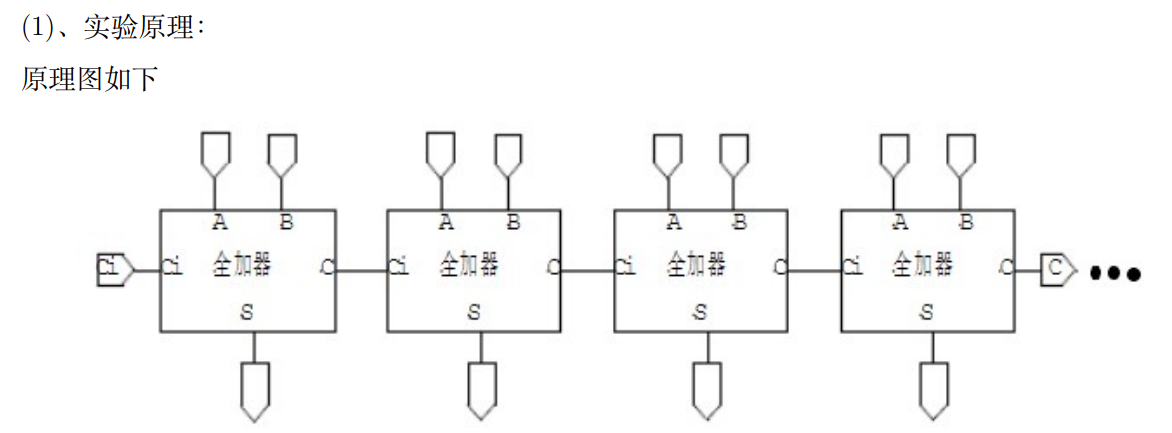
1. **实验总结：**

第一个实验由三个输入两个输出构成了一个电路

用门电路实现两个二进制数相 加并求出和的组合线路，称为一位全加器。一位全加器可以处理低位进位，并 输出本位加法进位。 一位全加器的真值表如下图，其中 Ai 为被加数，Bi 为加数，相邻低位来的进 位数为 Ci-1，输出本位和为 Si。向相邻高位进位数为 Ci



**14位全加器实验原理：**

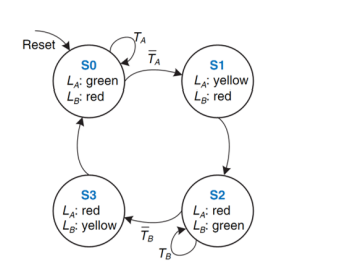


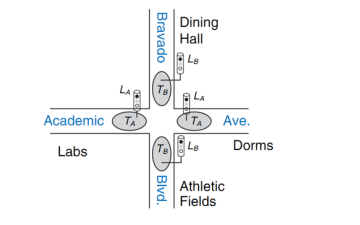
**实验三：交通灯与计数器**

1. **实验目的：**

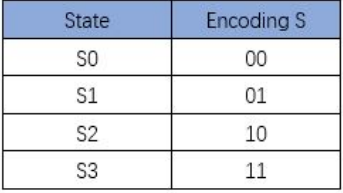
1、设计路口信号灯控制电路；2、设计构成一个 N 进制计数器；

1. **实验内容：**
   1. 、设计路口信号灯控制电路：

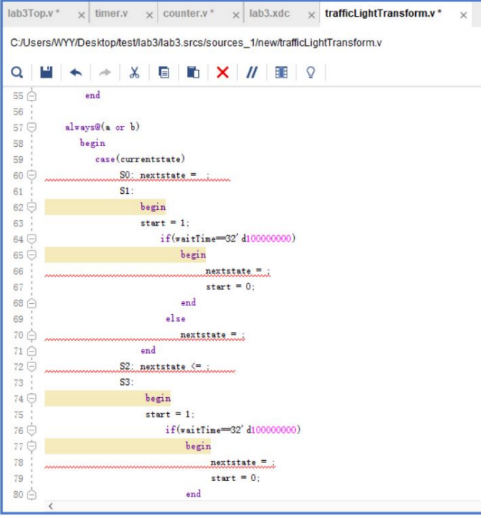






输出编码如下：

框架文件已经写好，只需将 trafficLightTransform 文件中，第 二个 always 中 nextstate 进行赋值。



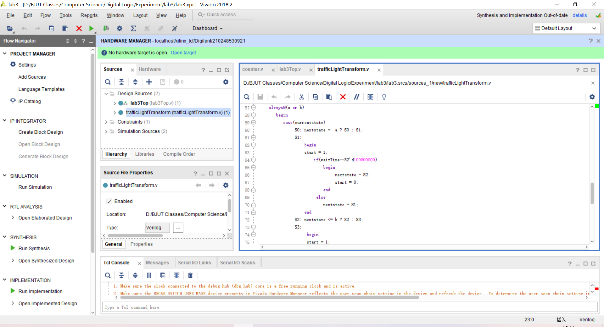
**二、设计构成一个N 进制计数器：**

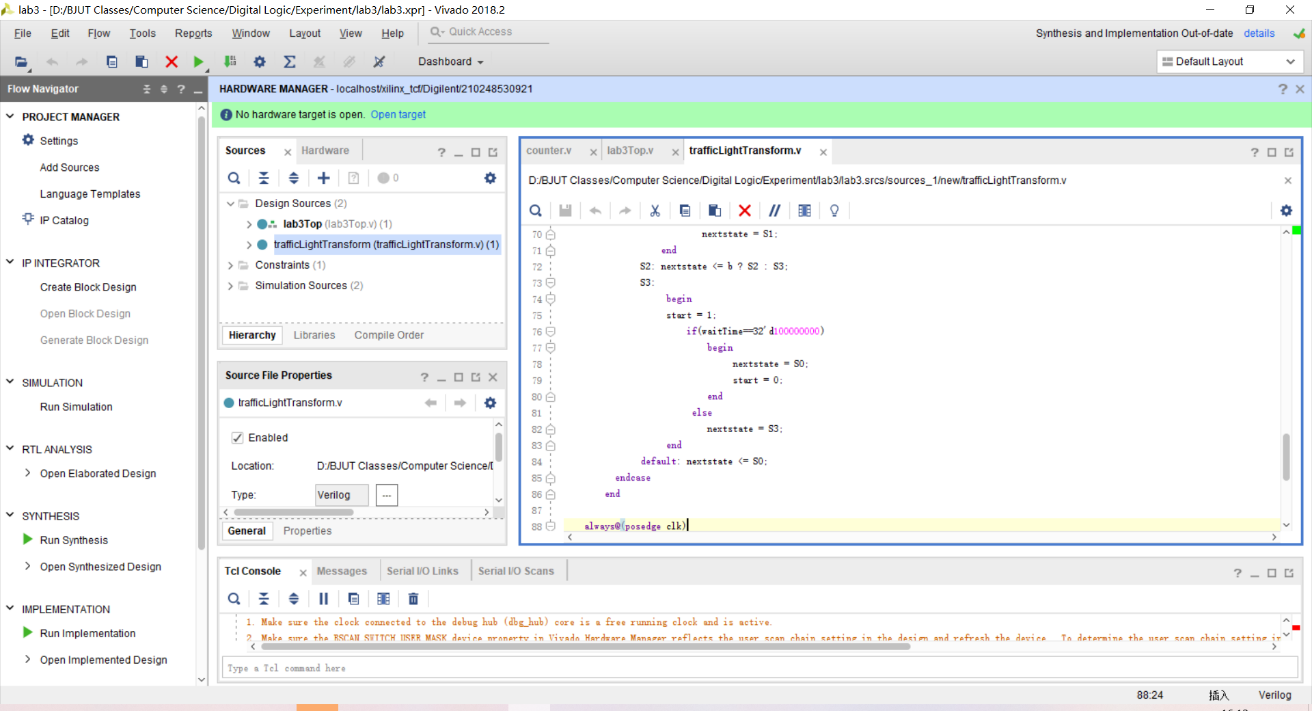
**(1)、实验原理：**

**按SWITCHE[2] 进行计数重置，按SWITCHE[0] 开始计时。**

**（2）、实验内容：框架文件已经写好，只需counter 文件中，always 补充完整。首先将lab3Top 文件，注释与解注。将counter 文件中，always 补充完整。**

****

1. **实验代码（除框架）：**

****

**计数器：**

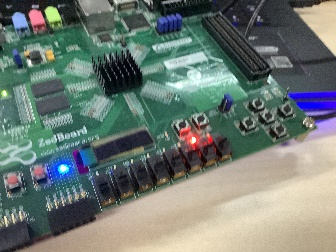
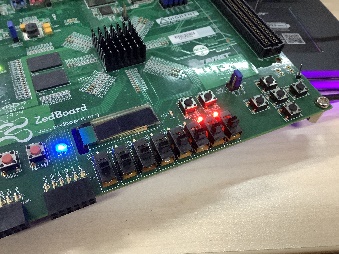
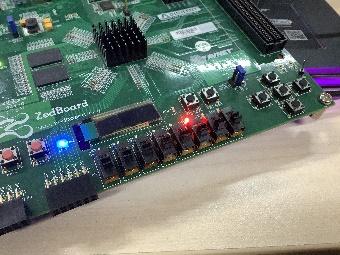


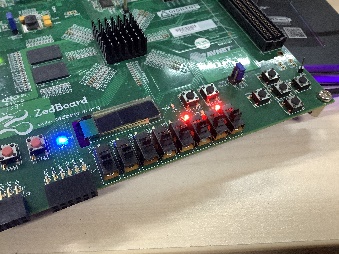
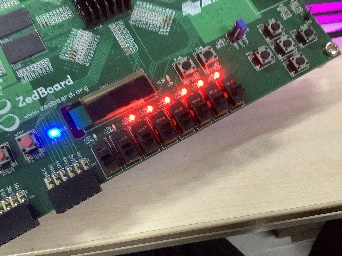
1. **实验结果：**

****

****

**计数器：**

****

****

1. **实验总结：**

**掌握了时序逻辑相关Verilog代码的写法，对时序逻辑的概念以及其在数字电路上发挥的作用有了进一步、更深一层的认识。**

**实验四：蜂鸣器实验**

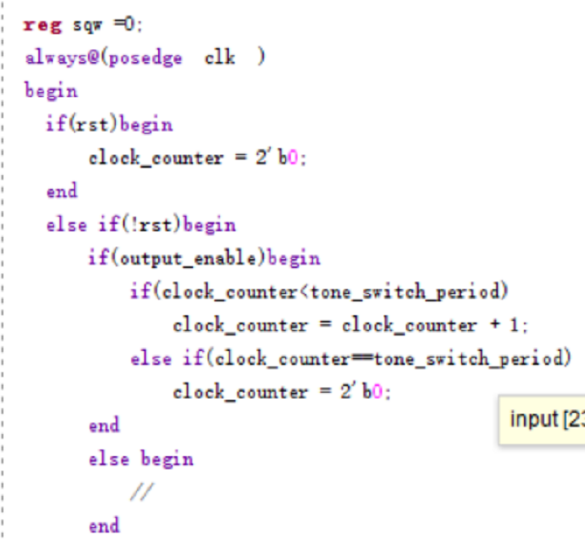
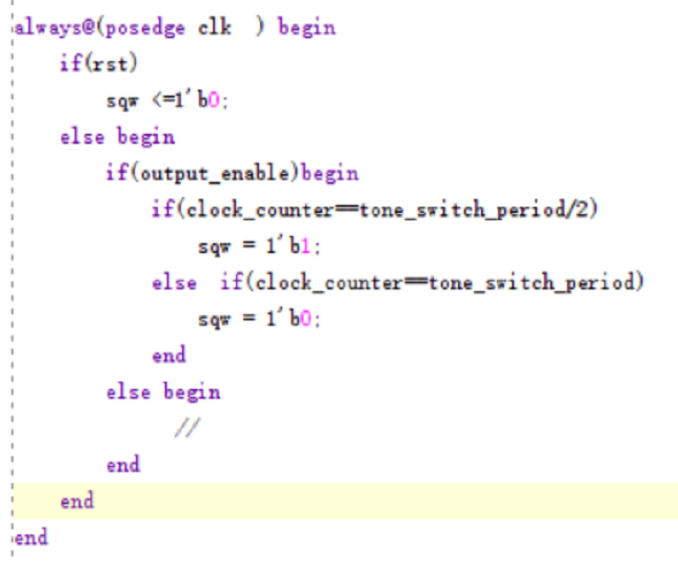
1. **实验目的：**

**设计一个可更改频率的音调发生器；**

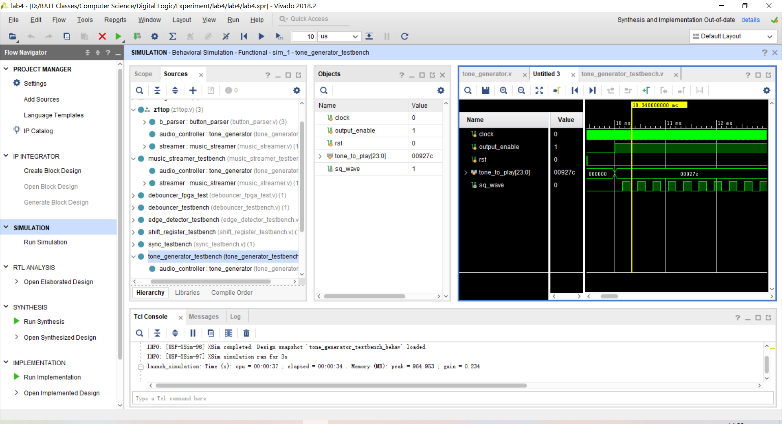
1. **实验内容：**

**需要参考实验一如何进行vivado 工程编译及烧写到开发板，lab4 的工程文件中给出了部分代码。设计一个可更改频率的音调发生器，音调的的不同，是由于振动的频率不同导致的。频率即周期的倒数。框架文件已经写好，只需tone generator.v 文件中，always 补充完整即可。**

1. **实验代码（除框架）：**



1. **实验结果：**

****

1. **实验总结：**

**进一步加深了对于时序逻辑电路的理解，掌握了产生固定频率波形的产生方法。**

**实验五：同步器、抖动器、边沿检测器**

1. **实验目的：**

**1、设计一个同步器；**

**2、设计一个去抖动器；**

**3、设计一个边沿检测器；**

**参考：需要参考实验一如何进行vivado 工程编译及仿真，lab4 的工程文件中给出了部分代码。**

1. **实验内容：**

**一、设计一个同步器：**

**(1)、实验原理：**

**在Verilog 中，数字信号是0 或1。在数字电路中，0 或1 对应于低电压或高电压。如果电路设计良好完全同步，我们只需担心低电压和高电压状态，但在本实验中，我们将处理异步信号。**

**板上按钮和滑动开关的信号没有相关的时钟信号。对于特别是按钮，当这些信号通过寄存器时，可能会违反该寄存器的保持或建立时间限制。这可能会使寄存器进入亚稳态。**

**图1: 亚稳态的“山上的球”比喻。如果违反了寄存器的时序约束，其输出电压会振荡。过一段时间后会不可预测地稳定下来。引入相对于时钟信号不变的异步信号会导致寄存器进入亚稳态。这是不可取的，因为这将导致“中间轨”电压传播到其他逻辑元件，并可能导致工具从未预见到的灾难性时序违规。**

**我们将实现一个同步器电路，它将安全地把异步信号带入同步电路。同步器需要有一个非常小的概率允许亚稳态传播到我们的同步电路。**

**我们希望您在本实验中实现的同步电路相对简单。为了同步一个位，它是一对串联的触发器。该电路同步进入现场可编程门阵列的异步信号(与任何时钟无关)。我们将使用我们的同步器电路将任何异步的off-FPGA 带入我们的FPGA设计的时钟域。**

**二、设计一个去抖动器：**

**(1)、实验原理：**

**在本实验中，去抖电路将接受一个按钮的毛刺数字输入，并输出一个干净的信号，指示一次按钮按压。我们需要一个去抖动器的原因可以在下图中看到。当我们按下按钮时，信号表现得不像一个完美的阶跃函数。相反，按钮信号由于机械“反弹”而出现故障。去抖器将显示单次按键的波形转换为具有单次电压转换的干净信号。**

**去抖器包括:**

**1、采样脉冲发生器：饱和计数器何时采样输入信号。它应该在每个samplecount max 时钟周期，SamplePulse 输出1。默认情况下，sample count max 设置为25000。**

**2、饱和计数器：这是一个计数到pulse count max 的计数器。如果SamplePulse 和input 信号在时钟沿都为高电平，则saturating counter 应增加1。在任一时钟沿，如果输入信号为0，饱和计数器应复位至0。一旦饱和计数器达到pulse count max，它应无限期地保持该值，直到输入信号降至0，此时饱和计数器应复位至0。**

**三、设计一个边沿检测器：**

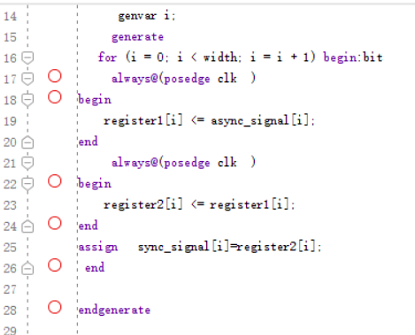
**(1)、实验原理：**

**沿检测器，它可以获取去抖器输出的低至高跃迁，并利用它产生一个1 时钟周期宽的脉冲，供我们的数字设计的其余部分使用。**

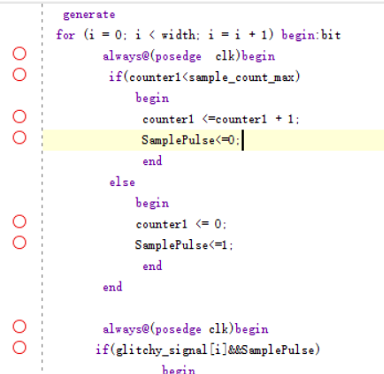
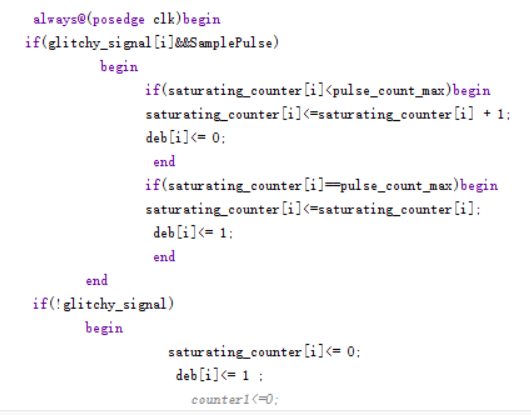
**（2）、实验内容：框架文件已经写好，只需edge detector.v 文件中，always 补充完整即可。**

1. **实验代码（除框架）：**

**同步器:**

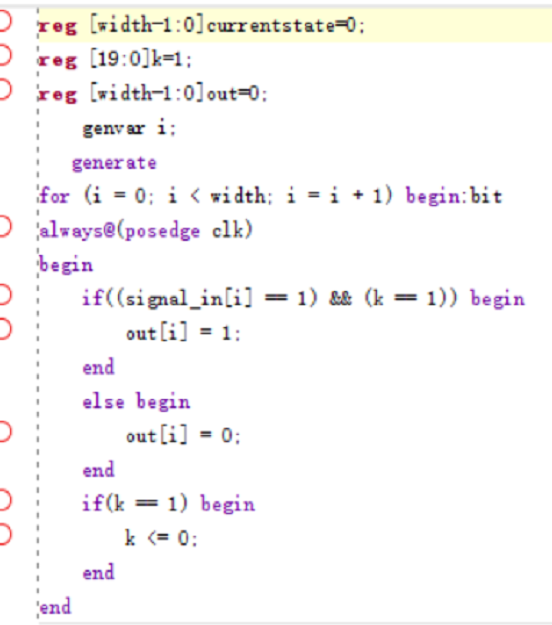


**去抖动器：**

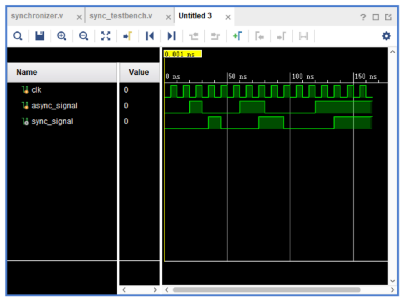


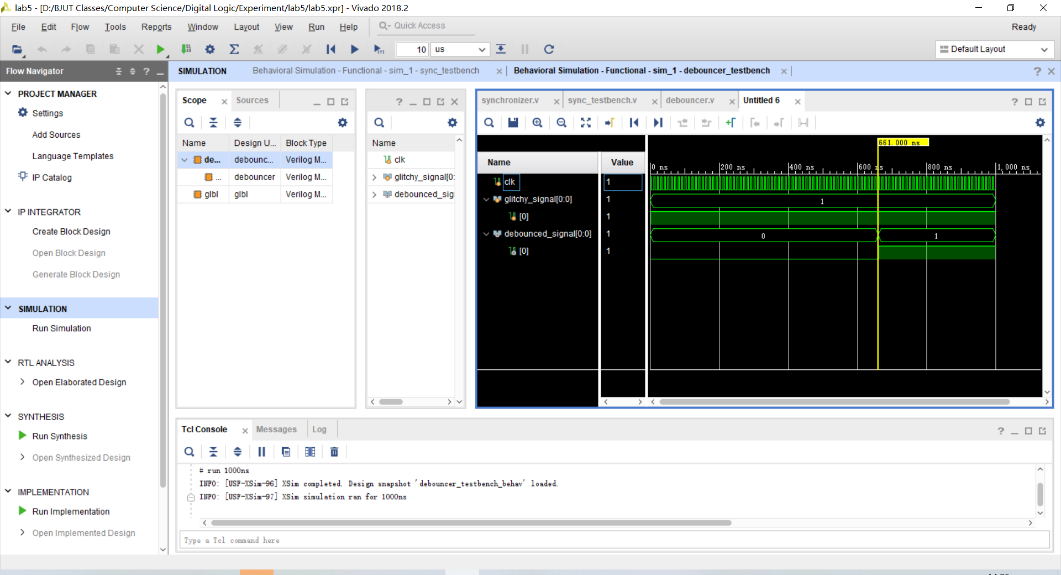


**边沿检测器：**

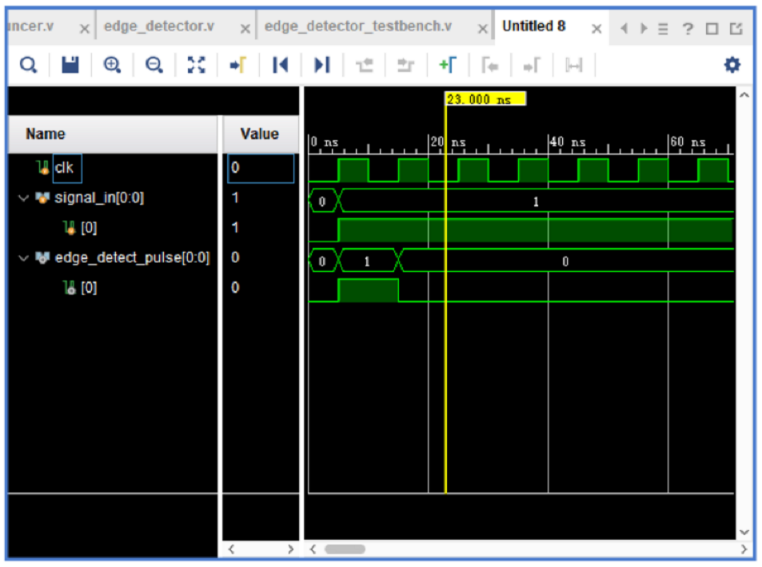


1. **实验结果：**

**同步器： 去抖动器：**

****

**边沿检测器：**



1. **实验总结：**

**通过同步器和去抖动器，俺进一步加深了对于亚稳态，毛刺的理解，掌握了避免产生亚稳态和毛刺的方法。通过边沿检测器，我加深了对于时序逻辑元件的认识。**