

本科实验报告

课程名称： **FPGA应用**

主讲教师： 夏书峰

学院、系： 微电子学院

专 业： 电子科学与技术类

班 级：

学 号：

姓 名：

联系电话：

电子邮箱：

2021～2022学年第3学期

实验成绩记录表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 内容 | 学时 | 满分 | 得分 |
| 1 | Vivado/ModelSim实践 | 4 | 15 |  |
| 2 | 时序逻辑设计1 — RTC | 4 | 20 |  |
| 3 | Xilinx嵌入式逻辑分析仪 | 4 | 15 |  |
| 4 | Xilinx嵌入式处理器实践 | 4 | 20 |  |
| 5 | 时序逻辑设计2 — FSM | 8 | 30 |  |
|  |  |  |  |  |
| 总计 | | 24 | 100 |  |

主讲教师签字：

实验一《Vivado/ModelSim实践》实验报告

【实验内容】

一、EDA平台搭建

1. 安装Mentor ModelSim SE软件，配置License
2. 安装Xilinx Vivado/Vitis软件，2020.2或2022.1版本都可以，配置license

二、熟悉ModelSim软件使用流程

1. 把理论课的Verilog HDL代码放进ModelSim或Vivado仿真，熟悉Verilog HDL代码风格，熟悉软件使用流程；
2. 了解测试向量的编写方法，包括用random产生激励信号的方法，以及用列举法产生激励信号的方法；新建空白工程，将被测的组合逻辑模块mux.v与测试向量tstmux.v加入工程，体会测试向量模块如何产生输入激励和查看输出信号；对模块进行测试，得到仿真波形，熟悉ModelSim各窗口功能和操作流程；
3. 自己在Verilog基础1课件后部找系统函数部分，改造一下测试向量文件，让测试向量在运行一定时间（例如10000个仿真时间单位时）自动停下来。

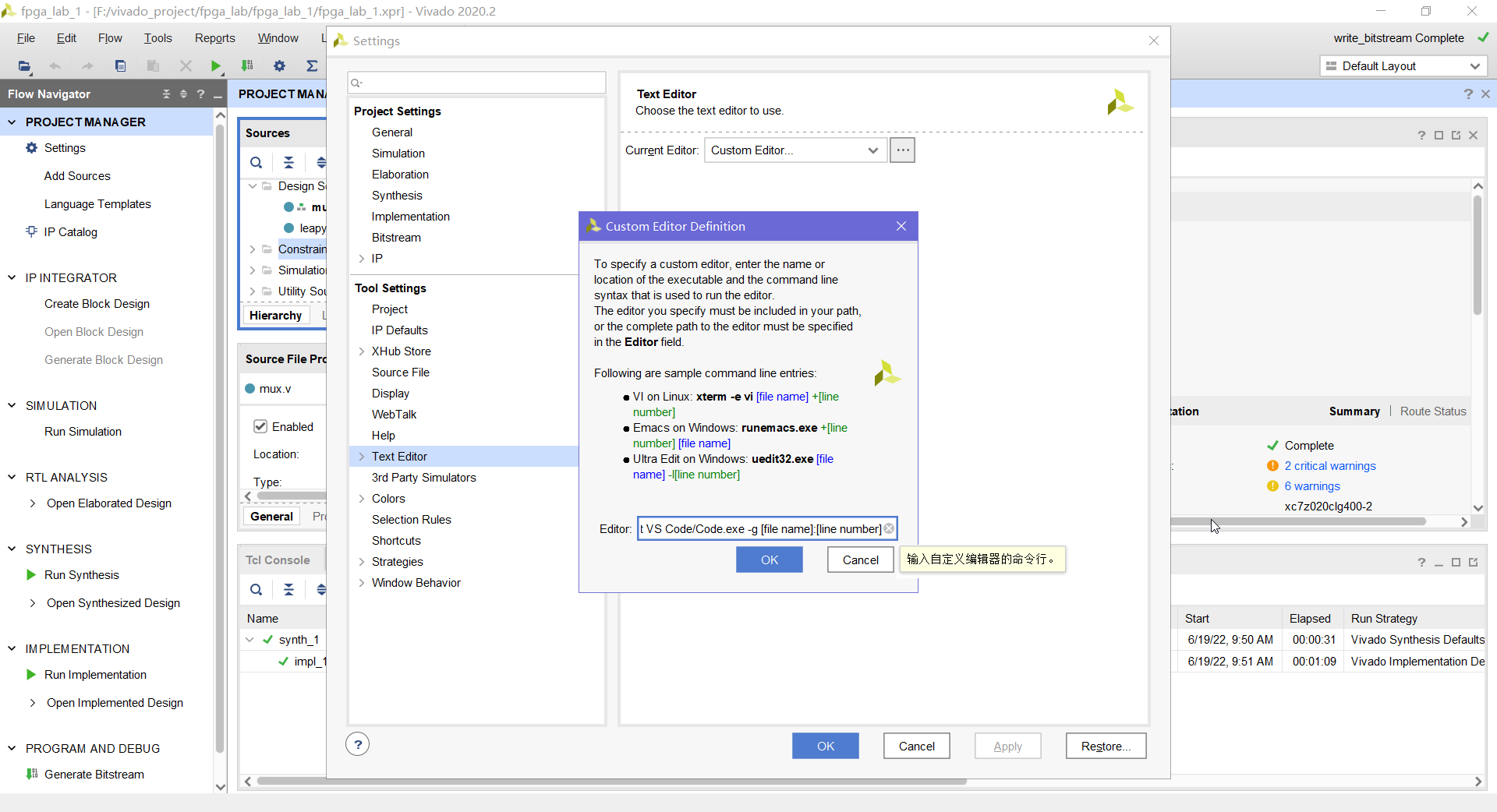
三、熟悉Vivado软件开发流程

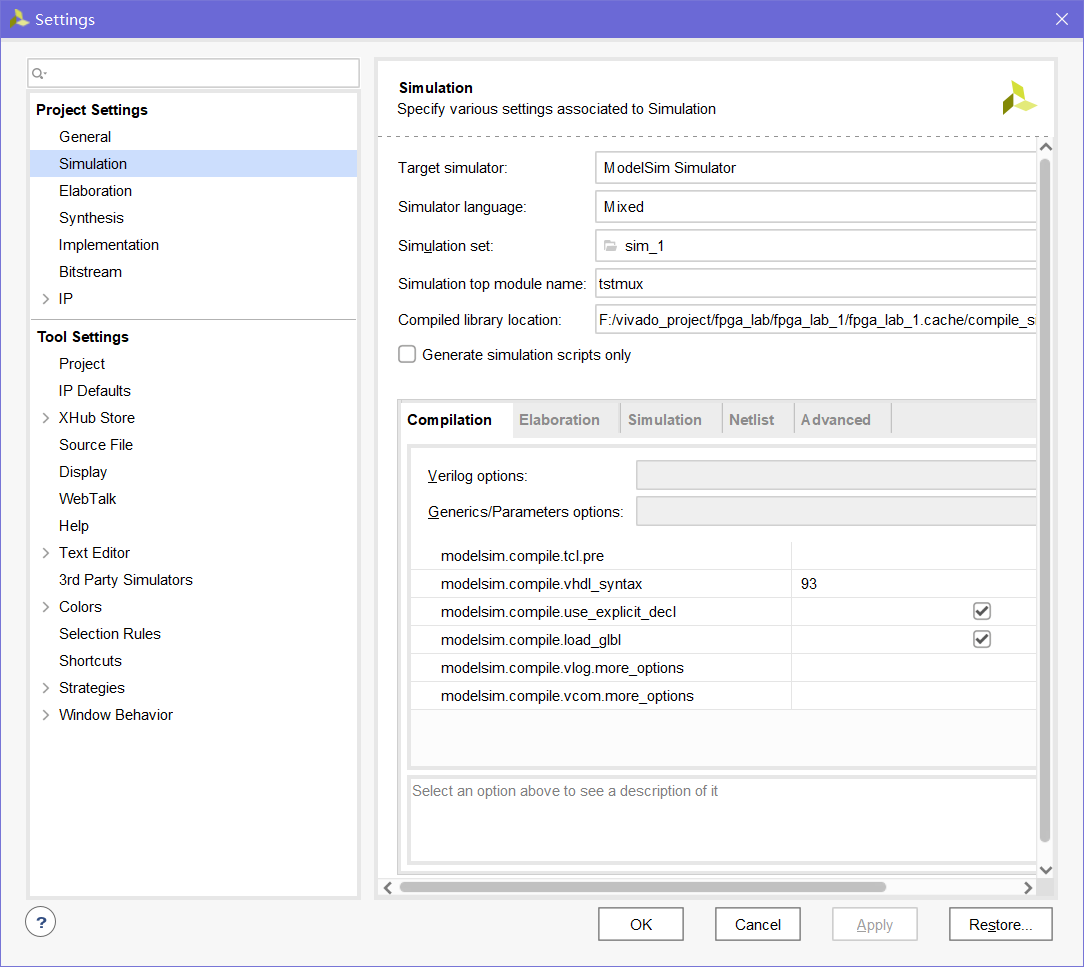
1. 按照视频的流程熟悉在Vivado下建立工程，输入设计文件，设定管脚约束，综合、布局布线，产生编程文件、下载到FPGA，测试逻辑功能等完整过程。

2. 尝试在Vivado里做仿真的流程，Vivado也可以像ModelSim一样，编写TB对所设计的模块做仿真测试。

【设计文件与实验过程】

熟悉软件使用流程部分，将提供的源文件放入工程并配置好了Vivado。使用Visual Studio Code替换Vivado的自带代码编辑器，同时使用Modelsim作为仿真器。如下所示：

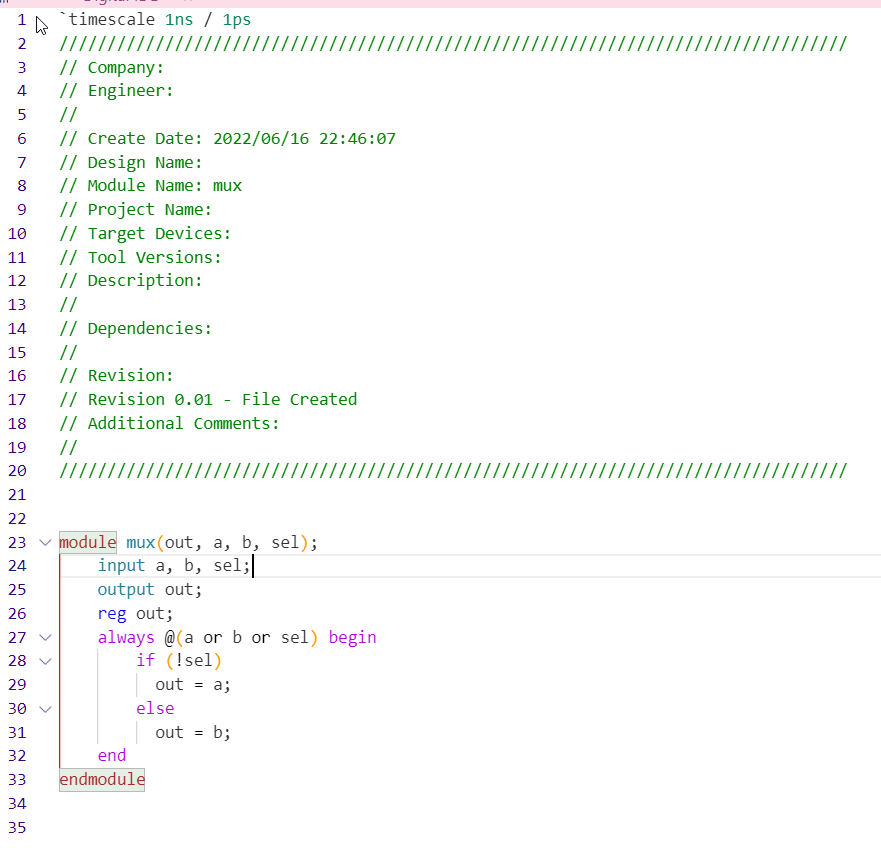




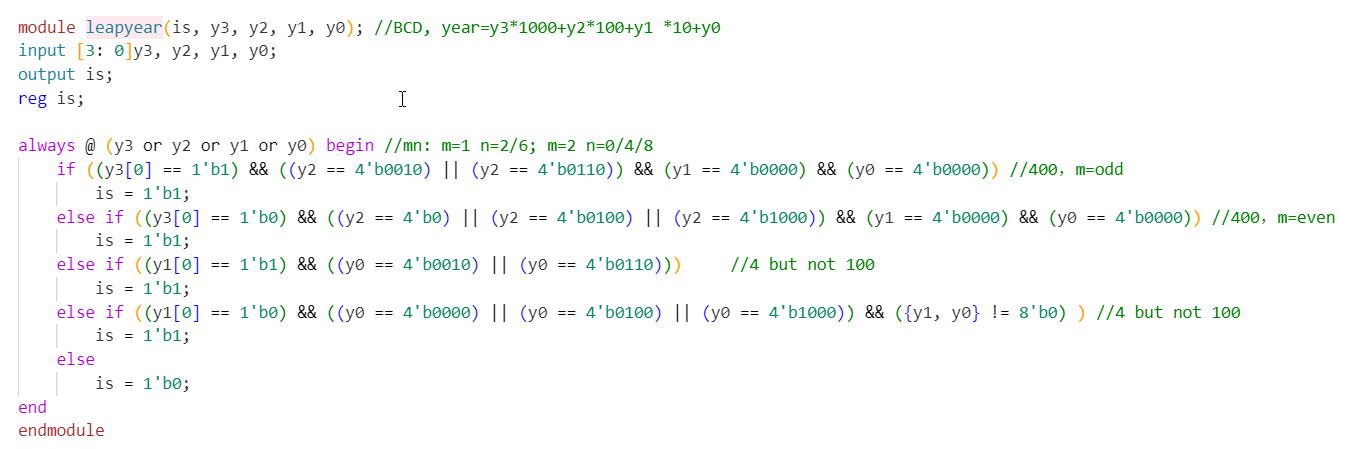


使用到开源Verilog仿真工具Verilator作为代码静态检查工具，使用iStyle工具进行代码格式化

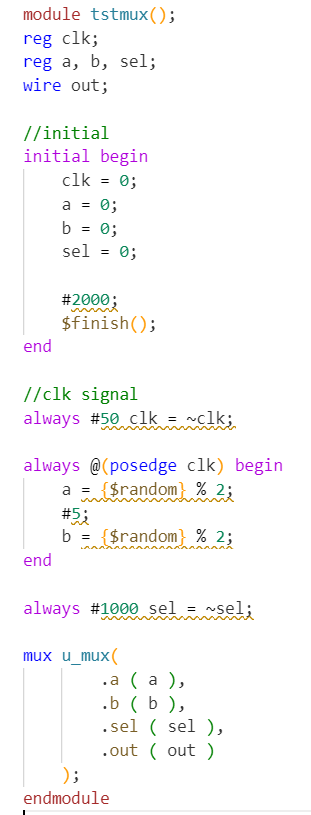
【源文件mux.v】



【源文件leapyear.v】

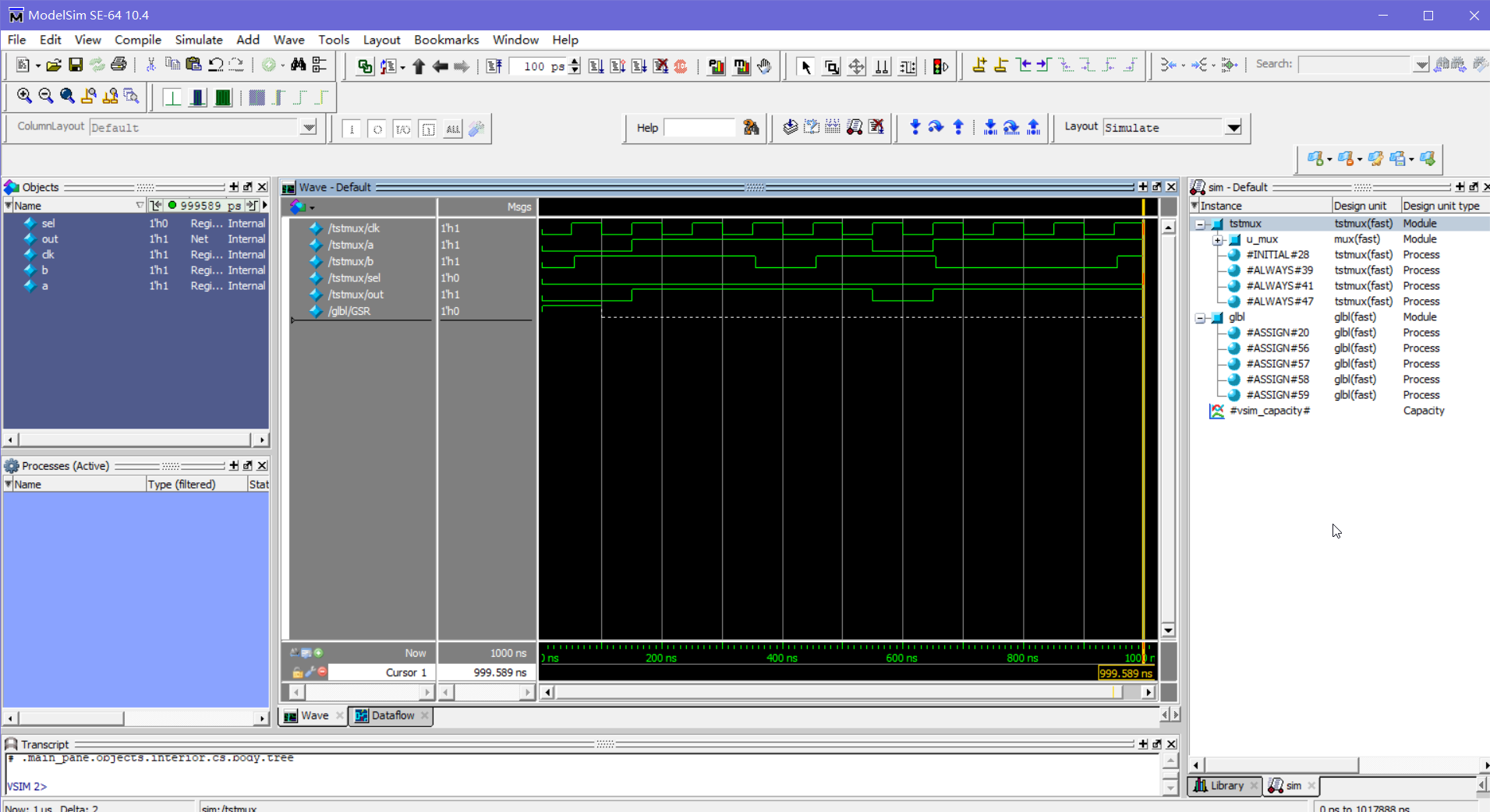


【测试向量tstmux.v】



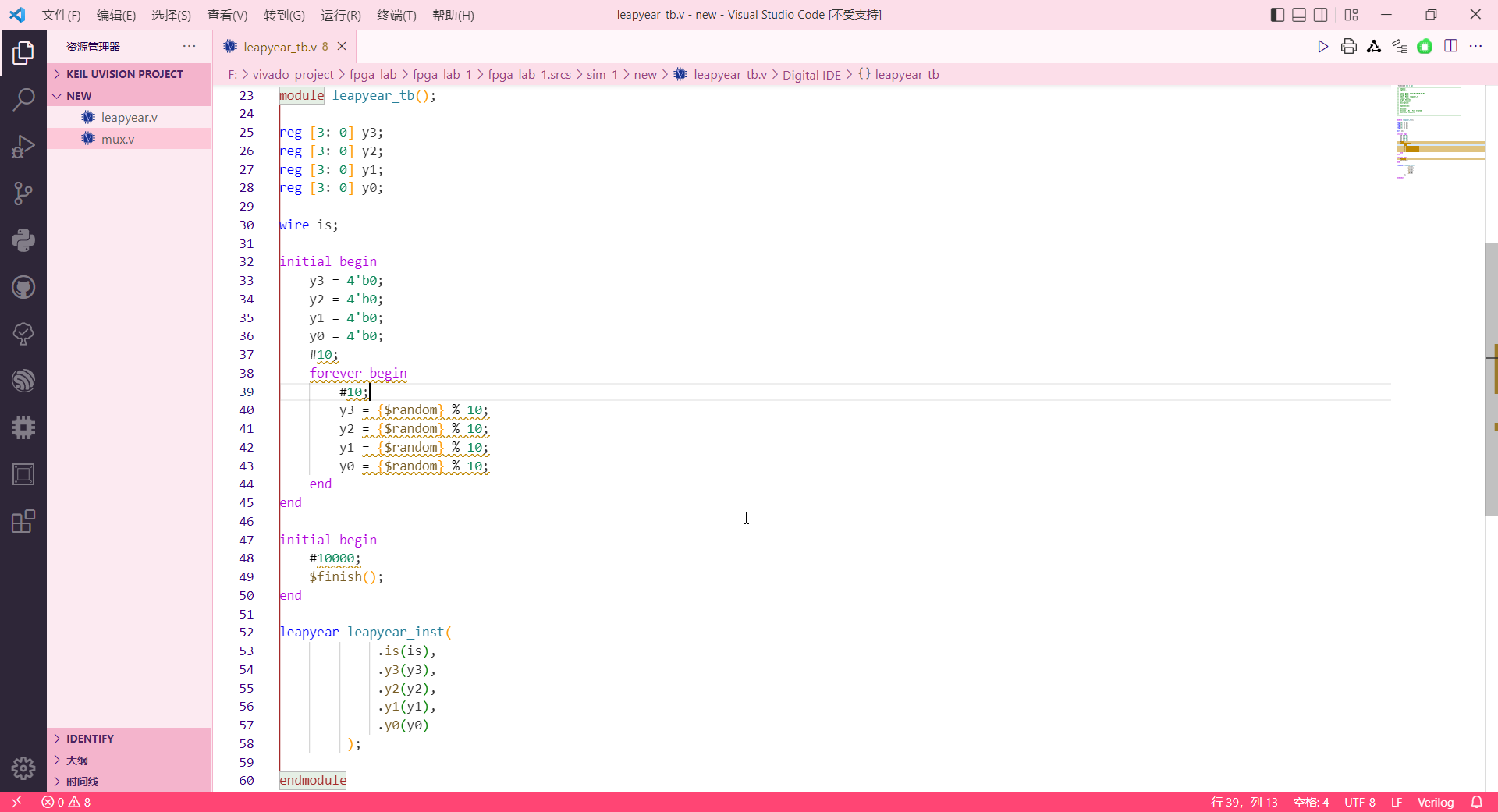
这里让测试程序运行2000ns后停止，即运行两轮sel变换的时间；同时在其中每50ns令输入数据改变一次

对应仿真结果（行为级仿真）如下所示：



实现了两输入多路选择器的效果

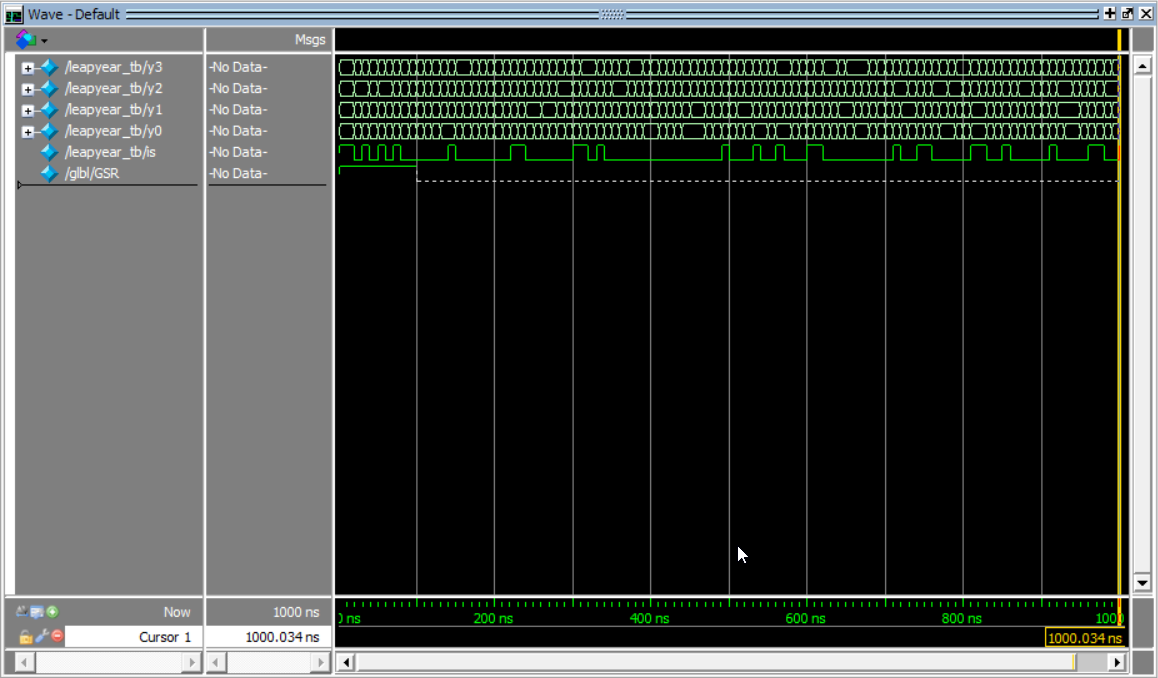
【测试向量leapyear\_tb.v】



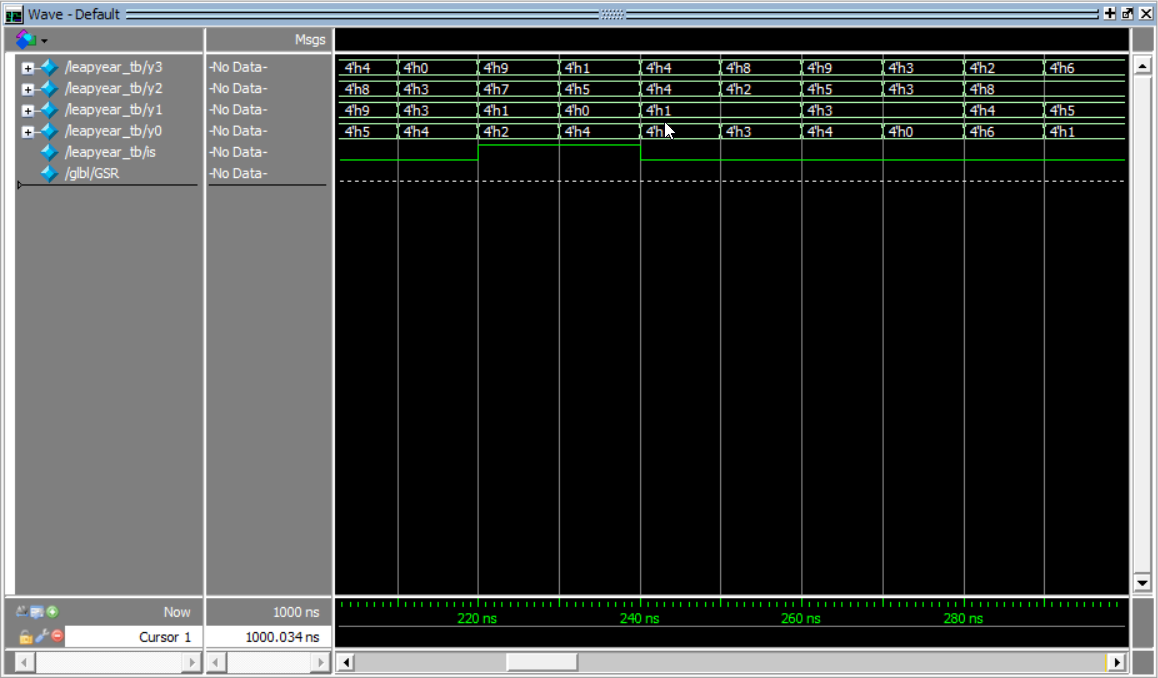
这里让程序运行10\_000ns后结束，仿真之前将leapyear.v设置为工程的顶层文件，分别使用Modelsim和Vivado进行行为级仿真

仿真结果如下：

Modelsim仿真：



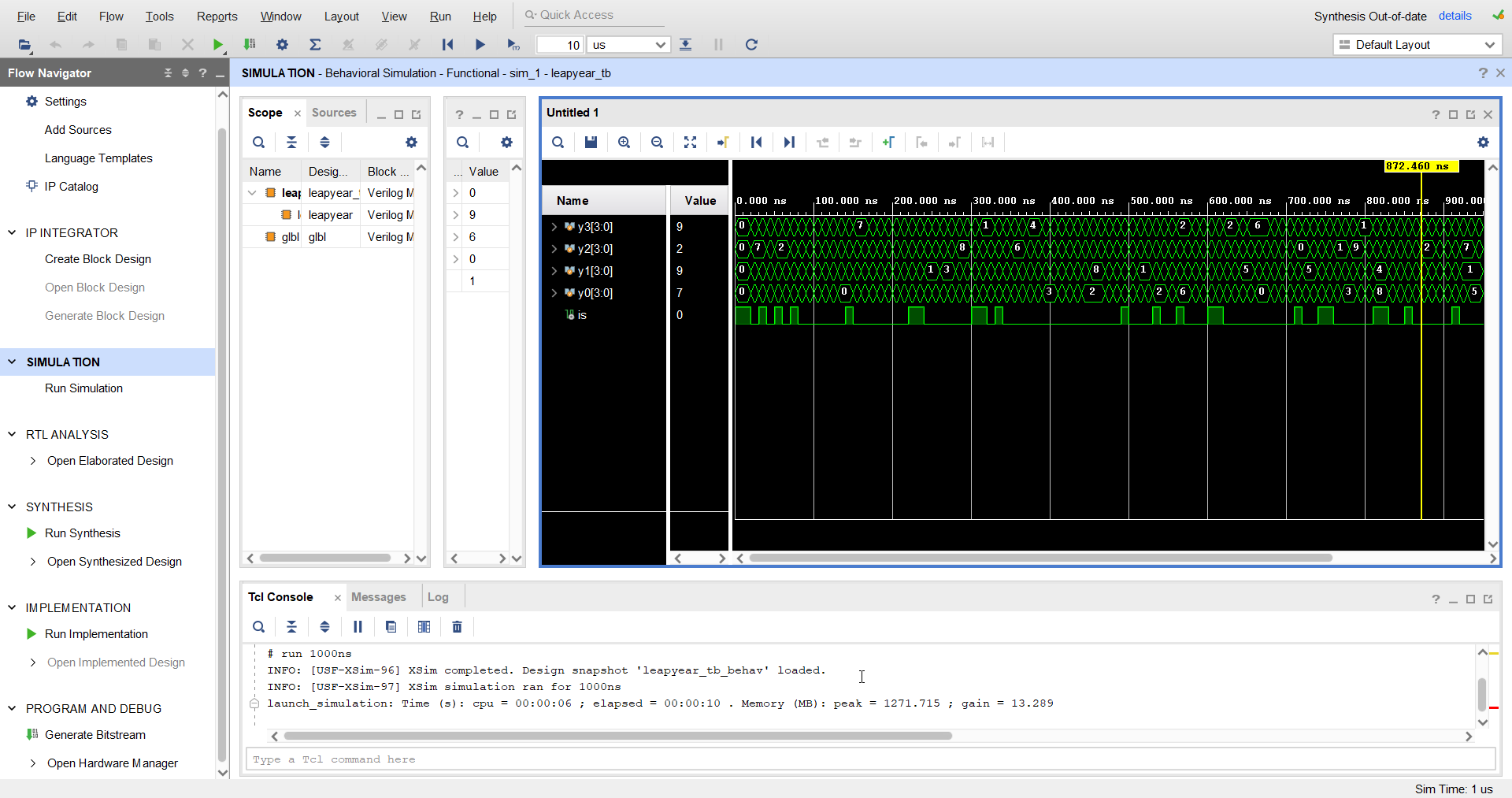
其中一段结果展示：



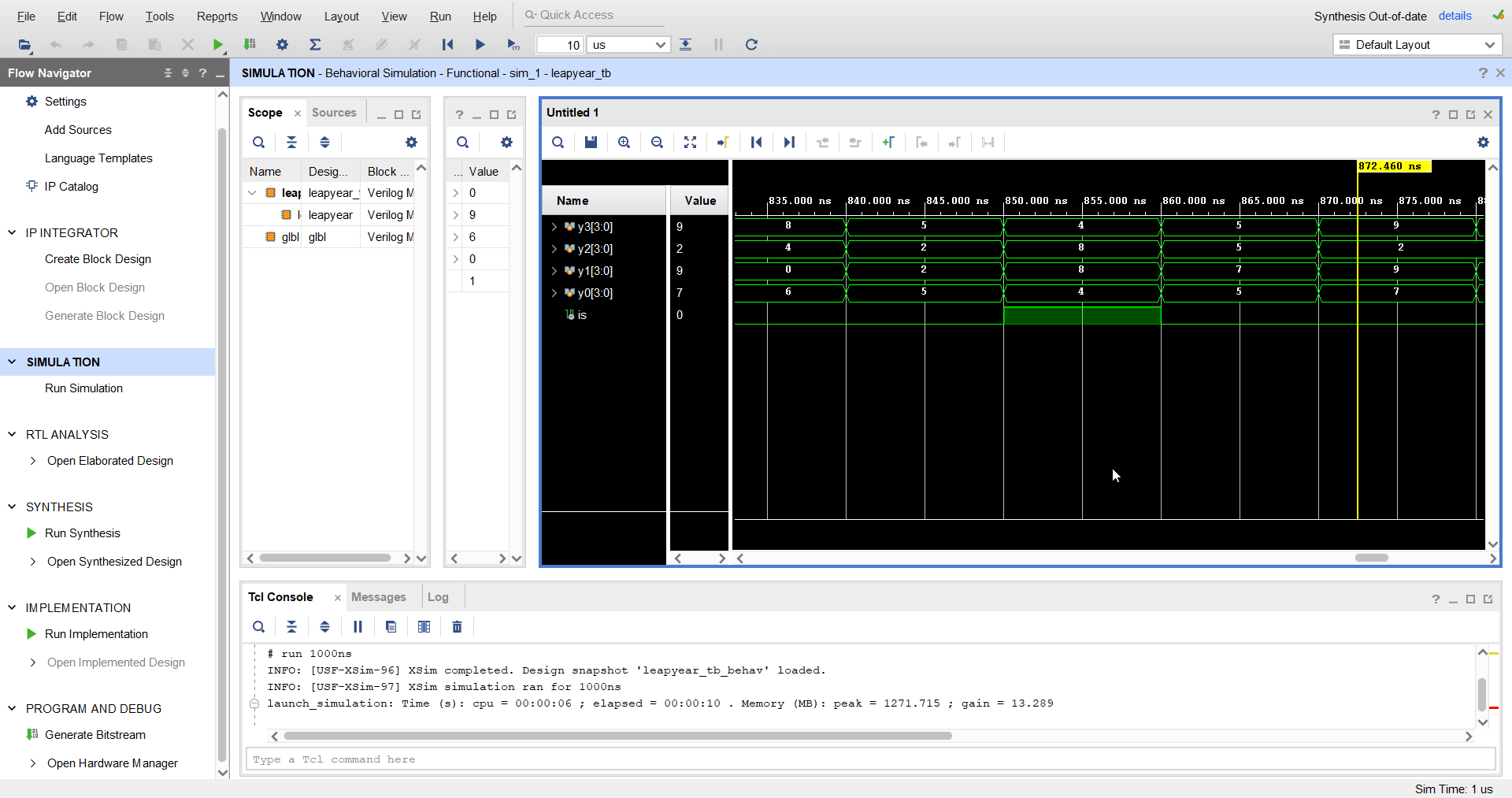
9712年、1504年都符合闰年定义，是闰年

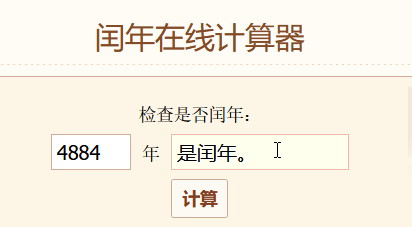


Vivado仿真：



其中部分结果：





可以判断验证正确

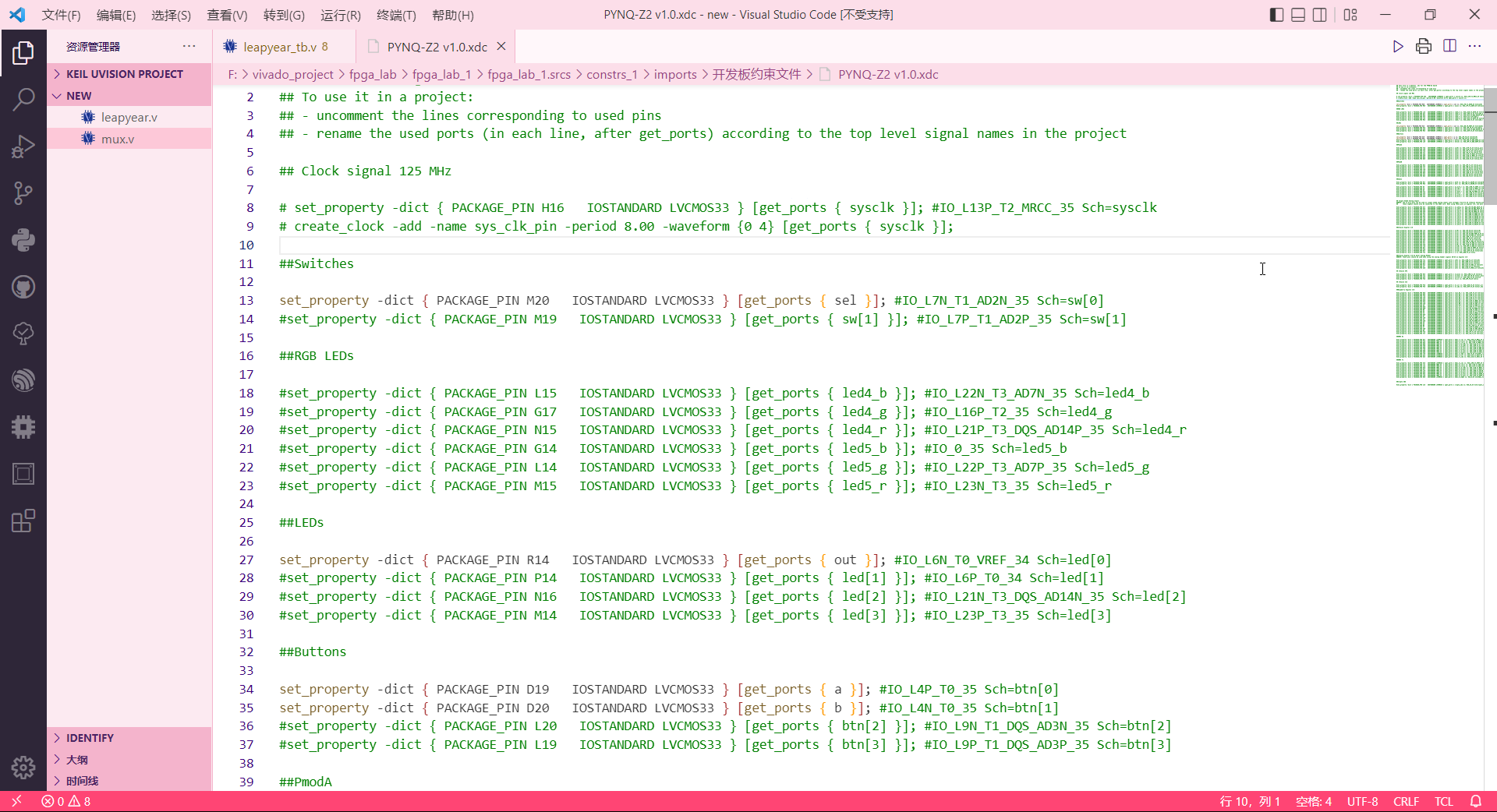
【结果分析】

项目中的DUT和测试向量都能正常完成工作。通过实验熟悉了Vivado和Modelsim的基本使用。

【熟悉Xilinx FPGA开发完整过程】

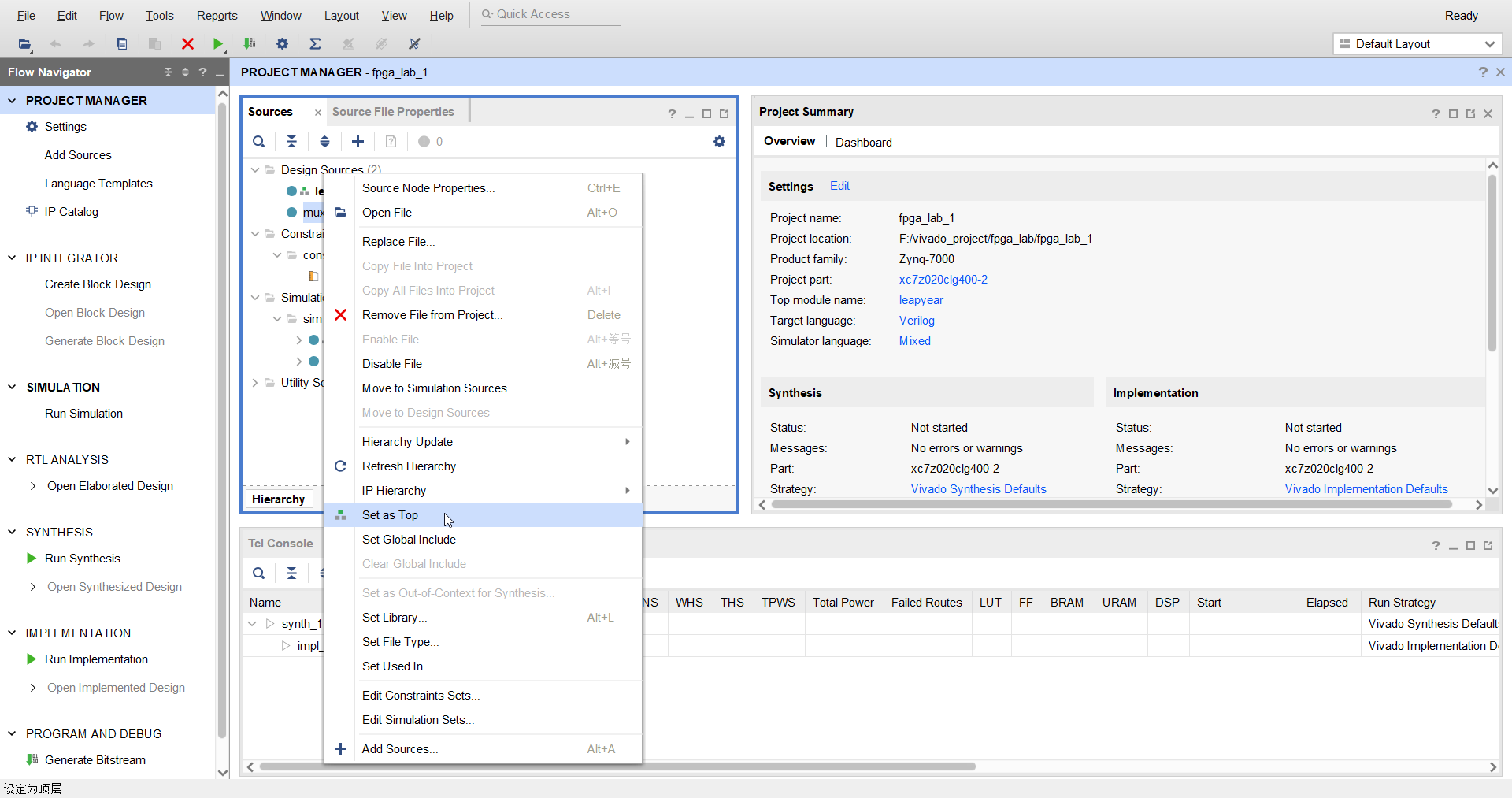
使用pynq-z2开发板官方提供的xdc约束文件，测试mux.v

模块的两个输入引脚直接使用板载按钮；选择引脚使用拨动开关；输出连接到板载LED，修改后的约束文件如下：

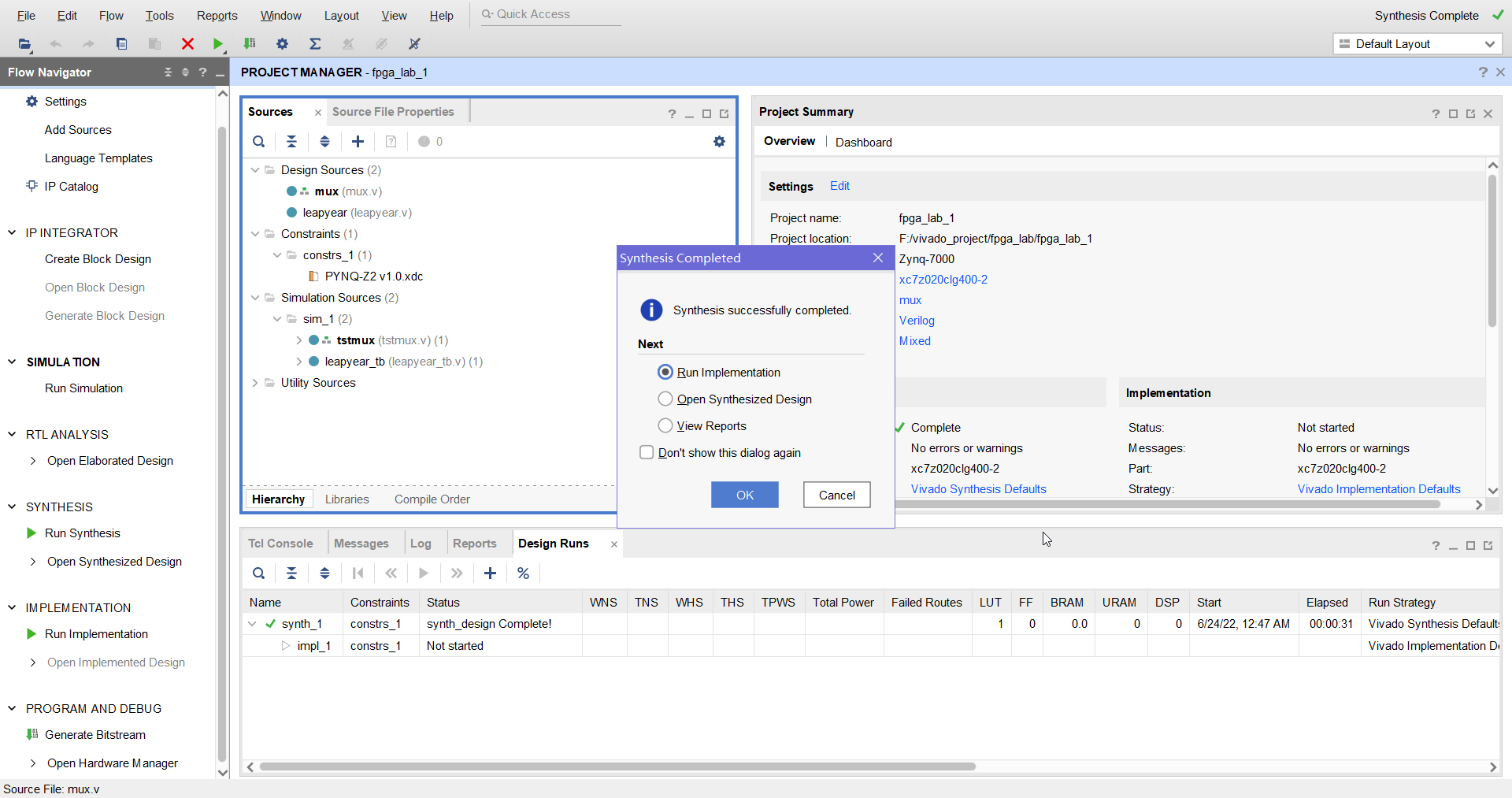
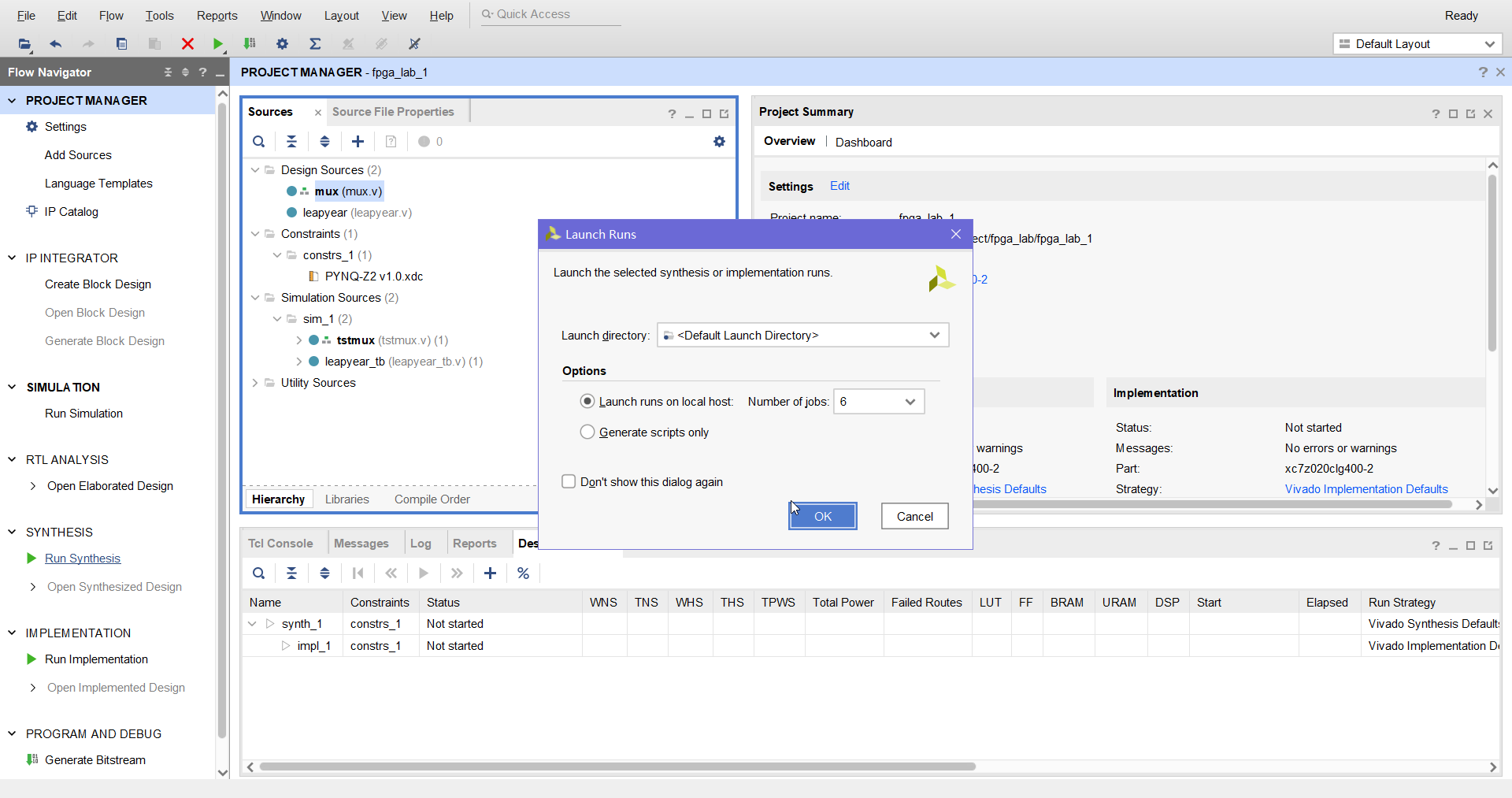


在Vivado中将mux.v设置为顶层模块，依次进行综合、布局布线、生成比特流文件，并通过JTAG烧录到FPGA，过程如下：

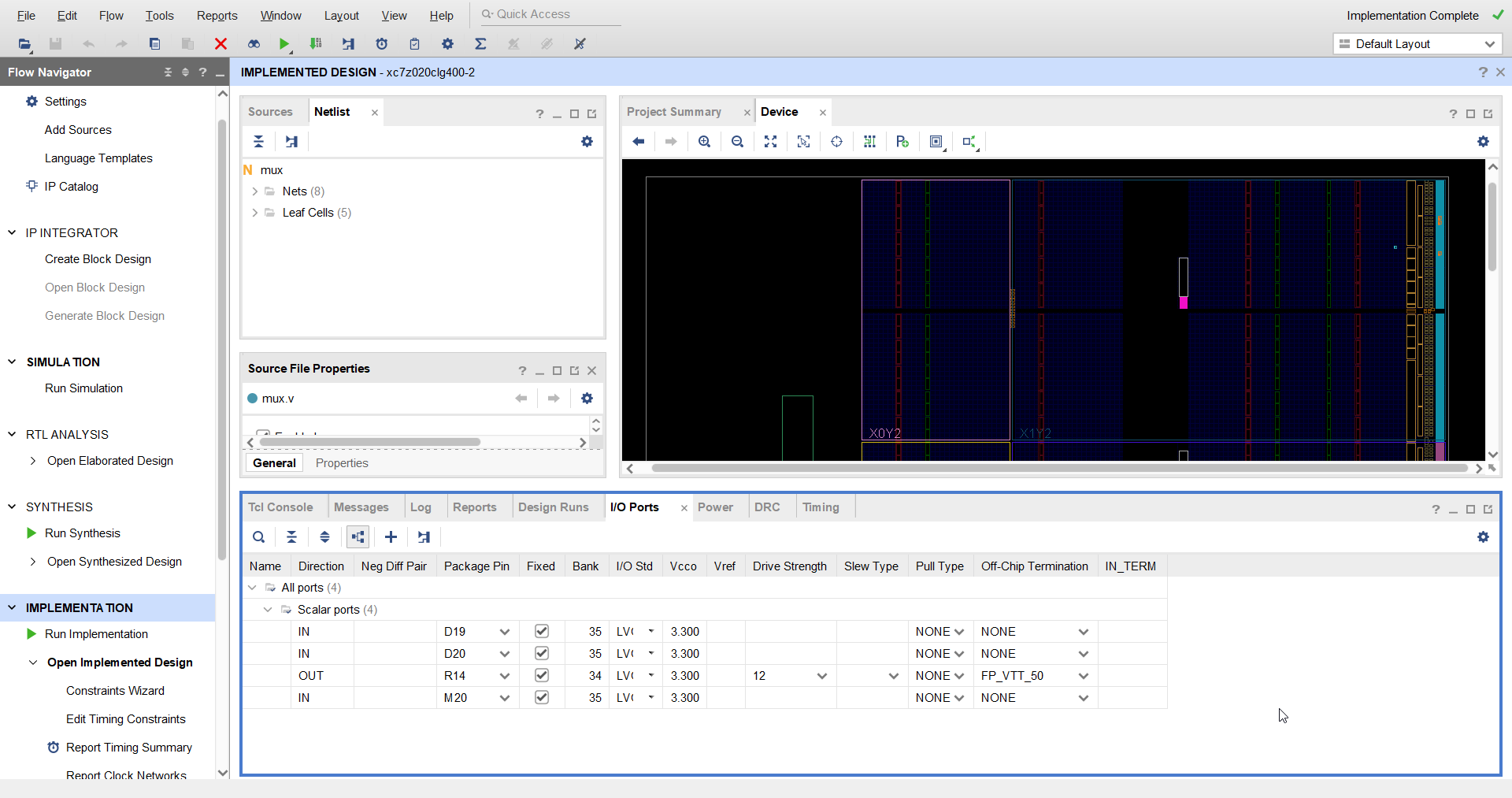
设置顶层文件



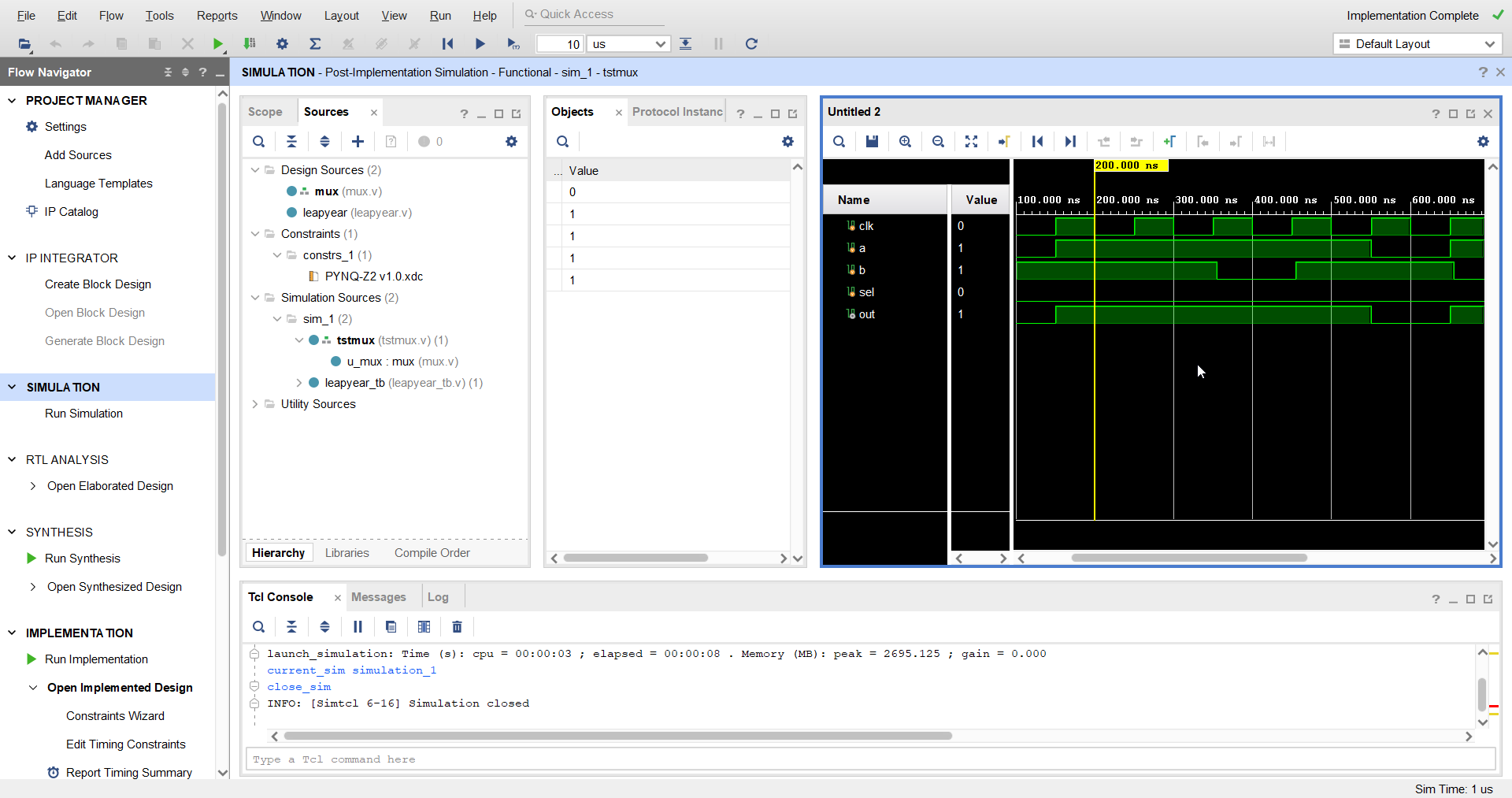
综合、布局布线



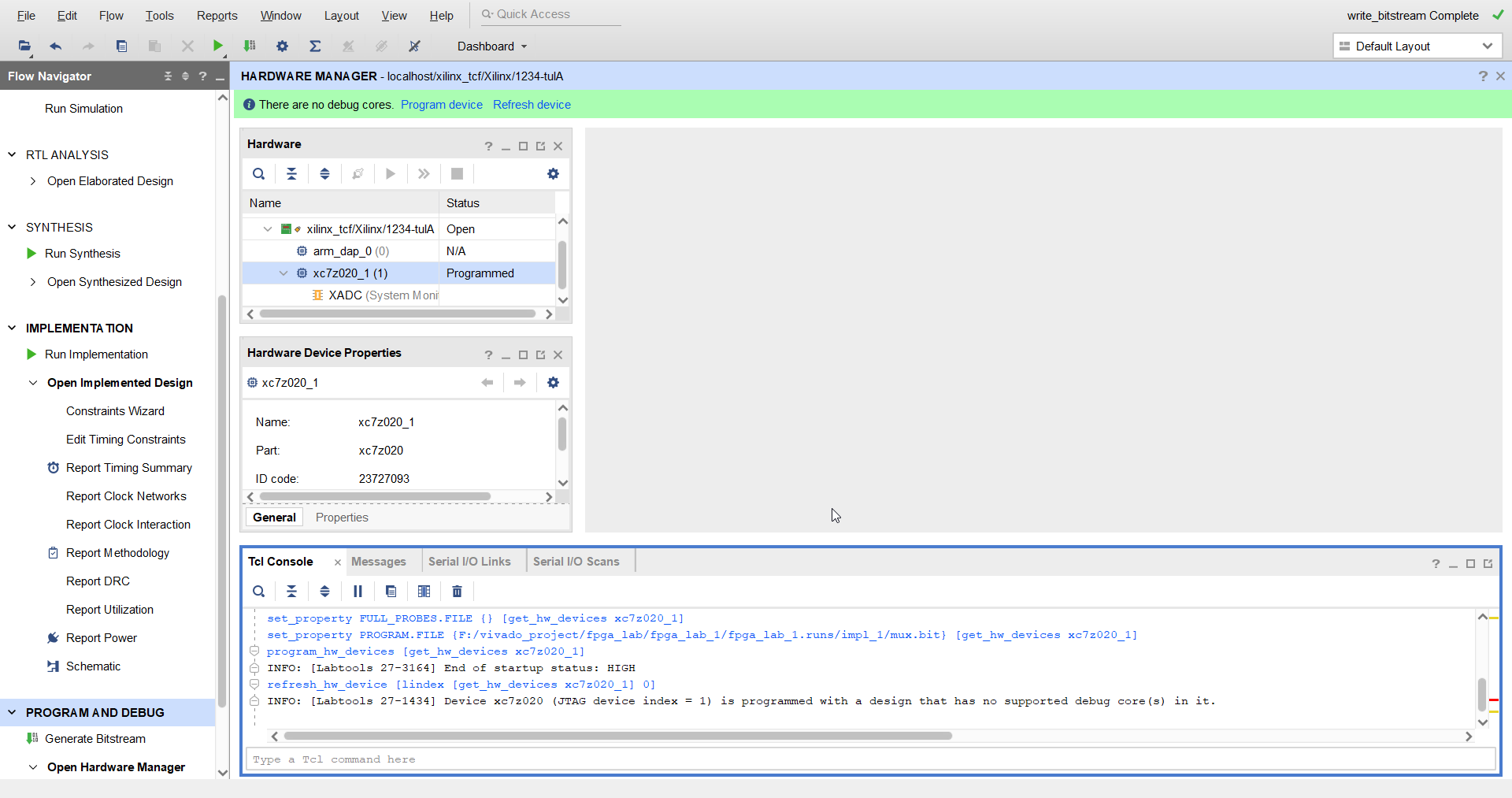
检查IO和布线结果



进行实现后仿真



生成比特流并烧录，上板验证



上板验证效果如下：

