**深 圳 大 学 实 验 报 告**

**课程名称：­ 数字集成电路**

**实验项目名称： ChiselTest 介绍与使用**

**学院： 计算机与软件学院**

**专业： 计算机科学与技术（创新班）**

**指导教师： 蔡晔**

**报告人：何泽锋 学号：2022150221 班级： 高性能特色班**

**实验时间： 2024年 9月 25 日 星期三**

**实验报告提交时间： 2024年 9月 25 日 星期三**

**教务部制**

|  |
| --- |
| **实验目的：**  1. 学习了解 ChiselTest 的相关背景知识，对测试有一个基本的概念与认识；  2. 掌握常用的 ChiselTest 测试 API，能够使用 ChiselTest 测试组合逻辑电路与时序逻辑电路；  3. 能够使用 ChiselTest 生成 VCD 波形。 |
| **实验内容与实验要求:**  1. ChiselTest 相关背景知识介绍；  2. 使用 ChiselTest 测试组合逻辑、时序逻辑电路；  3. 使用 ChiselTest 生成 VCD 波形。 |
| **实验步骤**  一、学习相关的Chisel语法  （1）peek  peek方法用于读取DUT中信号的当前值。通过 dut.io.signal.peek()，可以获取该信号的值。为了方便处理并直观观察信号，可以使用litValue方法将信号转为数值，即将硬件信号（UInt等）转为Int等数据类型。  （2）poke  poke方法用于设置信号的值。通过 dut.io.signal.poke(value)，可以将特定的值写入  DUT的输入信号中，以模拟不同的输入条件.  （3）expect  expect方法用于验证 DUT 输出信号是否符合预期。通过 dut.io.signal.expect(value)，可以将当前信号的值与期望值进行比较。  （4）step  step方法用于控制仿真进程的步进。通过 dut.clock.step(n)，可以逐步执行仿真，观察信号在每个时钟周期内的变化。step方法只能用于时钟信号，因为目的是推进时钟周期  二、实验任务使用 ChiselTest 编写一个测试下面的一个简单 Counter 模块，模块的功能是一个 4bit 的计数器，在 io.enable 有效的时候才会开始计数  （1）Counter模块  Counter模块是一个简单的计数器，只有在io.enable为true时才会计数    图1 Counter代码  （2）编写测试代码，代码提供了对第1个时钟的处理，因为初始时io.enable是false，因此时钟过后计数值不变，为初始值0。对于第2到14个时钟周期，此时io.enable为true，因此每个周期过后计数值都会加1。对于第15到29个周期，此时io.enable再次变为false，因此计数值保持与第14个周期相同。代码通过使用step来模拟时钟的变化。同时也对于每个时钟周期使用了expect来检验输出的信号是否与期望的一致    图2 测试代码  （3）运行结果可以看到计数值与分析结果一致：    图3 命令行运行结果  （4）使用gtkware观察波形图，需要先进入对应路径，再打开扁波形图文件，运行代码如下：    图4 进入波形图所在路径并打开  （5）观察波形图，可以看到计数情况与时钟信号的关系。此处的io.count和counter的波形一致，这是因为io.count是硬件输出，而counter是寄存器结果，二者是相等的。  图5 波形图 |
| 实验结论：  本次实验学习了ChiselTest的测试相关知识，能够完成简单的模块测试。掌握常用的ChiselTest测试 API，能够使用ChiselTest测试组合逻辑电路与时序逻辑电路。能够使用 ChiselTest 生成 VCD 波形，实验中成功使用gtkwave输出计数器模拟步进的波形图。 |
| 指导教师批阅意见：  成绩评定：  指导教师签字：  年 月 日 |
| 备注： |

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。

2、教师批改学生实验报告时间应在学生提交实验报告时间后10日内。