****

计算机组成原理（甲）

实 验 报 告

|  |  |
| --- | --- |
| **学 院** | 网络空间安全学院 |
| **专 业** | 网络工程 |
| **班 级** | 19272401 |
| **学 号** | 19061440 |
| **学生姓名** | F001 |
| **教师姓名** | 袁理峰 |
| **完成日期** | 2020-12-04 |
| **成 绩** |  |
| 实验四 寄存器堆设计实验 | | |
| 1. **实验目的** 2. 学习使用Verilog HDL进行时序电路的设计方法 3. 掌握灵活运用Verilog HDL进行各种描述与建模的技巧和方法 4. 学习寄存器堆的数据传送与读/写工作原理，掌握寄存器堆的设计方法 | | |
| 1. **实验原理**   寄存器的相关原理，并以堆的形式存放。  CPU内部通常包含若干个通用寄存器，以暂存参加运算的数据和中间结果。寄存器速度快，个数小，但是RISC CPU的设计强调设置大量的寄存器。  所谓寄存器堆，就是一个寄存器集合，为方便访问其中的寄存器，对寄存器堆中的寄存器进行统一编码，成为寄存器号或寄存器地址。每个寄存器均通过指定寄存器号进行访问。  本设宴设计了一个32\*32位的寄存器对，即含有32个寄存器，每个寄存器32位。该寄存器堆有两个读端口、1个写端口，即能够同时读出两个寄存器的值，写入1个寄存器，下图为其功能表：   |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | 输入信号 | | | | | 输出信号 | | 操作 | | R\_Addr\_A | R\_Addr\_B | Write\_Reg | W\_Addr | W\_Data | R\_Data\_A | R\_Data\_B | | 寄存器号 | —— | —— | —— | —— | A口数据 | —— | 读A口 | | —— | 寄存器号 | —— | —— | —— | —— | B口数据 | 读B口 | | —— | —— | 1 | 寄存器号 | 写入数据 | —— | —— | 写操作 | | | |
| 1. **主要操作步骤及实验结果记录**   （对实验过程中的主要操作步骤进行描述，并随时记录实验过程中观察到的结果，必要时可辅助截图）  任务一：在Xilinx ISE中创建工程，编辑程序源代码，然后编译、综合，若编译出错，则修改程序代码，直至正确。          尽管通过了语法检查，但是在查看电路的过程中，出现了相关错误    上网查询后得知，该错误是由于在不同的子程序内调修改了相同变量的值，导致最终结果不知道依据哪个程序进行，导致程序出现错误。  但是奇怪的是，该问题似乎难以解决。相同的代码，在不同的终端上运行，结果不同（老师电脑没问题我的电脑有问题，机房电脑有问题，且都是相同的问题），尽管了解了问题的原因，但是经过无数次修改之后，最后还是有问题，难以观察其电路图。  任务二：编写激励代码，观察仿真波形。如果验证逻辑有误，则修改代码，重新编译、仿真，直至正确。      画出仿真波形后，发现程序内部好像是正常执行的，但是总是难以查看电路图。  与其他同学交流后，我了解到其内部的原理如下图所示    内部具体结构更为复杂： | | |
| 1. **实验分析总结及心得**   （结合所学知识对实验过程中观察到的实验结果进行分析总结，以便加深对知识的理解，并总结通过实验学到的知识或技术）  上次遇到问题的时间已经很久了，宝贵的问题终于久别重逢。  此次实验中，尽管所有同学使用的代码相同，但是一大部分人出现了上述的错误。经查询，该问题是由于“在不同的进程里面同时对同一个[寄存器](http://www.so.com/s?q=%E5%AF%84%E5%AD%98%E5%99%A8&ie=utf-8&src=internal_wenda_recommend_text)进行了赋值当在不同进程里面对同一个寄存器赋值的时候 编译器无法判断寄存器的值到底该被哪个进程赋值”这确实是个问题，但是尽管尝试了更改变量名，调整函数内部语句等多种操作，该错误仍然会出现。但是若将出错行注释掉，内容又相应正常。  更为奇怪的是，该段代码在不同终端运行时的结果不尽相同。有相同错误的，有一切正常的，我进行的猜测是可能是操作系统的环境，以及配置ISE时的相关环境上的区别，导致的这个现象发生。另一种猜测是，由于ISE可能涉及相关硬件的操作，这是由于硬件本身的复杂性导致的，出现的某种“玄学错误”。例如接触不良、引脚松动等，当然，这些只是硬件本身的问题，并不是引用在该问题上，仅是一种启发与思考。  在这个实验中，32\*32位的寄存器堆内部详细结构确实十分复杂，现实中真正使用的CPU结构必然更为复杂，不由得对相关工程师心生敬意。 | | |