

本历史

文档更新记录		文档名:	Lab14_TLB MMU 设计（三）	
		版本号	V0.1	
		创建人:	计算机体系结构研讨课教学组	
		创建日期:	2019-11-28	
更新历史				
序号	更新日期	更新人	版本号	更新内容
1	2019/11/28	贾凡	-	草稿。
2	2019/12/03	邢金璋	V0.1	初版。

文档信息反馈: xingjinzhang@loongson.cn

1 实验十四 TLB MMU 设计（三）

在学习并尝试本章节前，你需要具有以下环境和能力：

- (1) 装有 Vivado 的电脑一台。
- (2) 熟悉 Vivado，并能初步使用。
如果对 Vivado 不熟悉，请参考课程讲义中的第一讲内容。
- (3) 初步掌握 Verilog 的简单语法。
- (4) 熟悉龙芯体系结构实验箱（Artix-7）。

通过本章节的学习，你将获得：

- (1) 掌握 TLB MMU 的知识。
- (2) 理解 CPU 中地址翻译的机制。
- (3) 理解 MIPS 架构的 TLB miss 处理过程。

本次实验需要参考的文档包括但不限于：

- (1) Lab14 任务书（本文档）。
- (2) 体系结构研讨课总讲义之第九章“TLB MMU 设计”。
- (3) 《MIPS 指令手册》。

1.1 实验目的

1. 掌握 TLB MMU 的知识。
2. 理解 CPU 中地址翻译的机制。
3. 理解 MIPS 架构的 TLB miss 处理过程。

1.2 实验设备

1. 装有 Xilinx Vivado 的计算机一台。
2. 龙芯体系结构教学实验箱（Artix-7）一套。

1.3 实验任务

本次实验只有一个任务：

1. 在 lab13 的实验环境中，完成：
 - a) CPU 增加 TLB 相关例外：Refill、Invalid、Modified。
 - b) 运行专用功能测试 tlb_func，要求全部通过，共 9 项测试。
2. 本次实验要求以组为单位提交实验报告和 RTL 代码，以报告评分和现场检查评分作为最后的实验得分：
 - (1) 报告评分：描述自己的设计方案，记录调试过程（错误记录应该配截图）。实验报告模板请使用 lab3 的模板。

(2) 现场检查评分：检查包含仿真检查和上板检查。

1.4 实验检查

检查前需提交实验报告（每组一份）和调试好的 RTL 代码。本次实验在 2019 年 12 月 17 日进行检查。

现场检查，分仿真检查和上板检查：

1) 仿真检查：对照波形进行描述 TLB 比较和重填的过程。

2) 上板检查：查看上板行为。

现场检查要求能正确应对检查者的提问，并根据要求进行正确的操作演示

1.5 实验提交

提交的作品包括纸质档和电子档。

(1) 纸质档提交

提交方式：课上现场提交，每组提交一份。

截止时间：2019 年 12 月 17 日 18:10。

提交内容：纸质档 lab14 实验报告。

(2) 电子档提交

提交方式：打包上传到 Sep 课程网站 lab14 作业下，每组提交一份。

截止时间：2019 年 12 月 17 日 18:10。

提交内容：电子档为一压缩包，文件名是“lab14_箱子号.zip”，目录层次如下（请将其中的“箱子号”替换为本组箱子号）。

lab14_箱子号/	目录，Lab14 作品。
--lab14_箱子号.pdf/	Lab14 实验报告，实验报告模板参考“Lab03 实验报告模板_仅供参考.docx”
--myCPU /	目录，myCPU 源码。目录请加一个 readme，简单描述下各文件。

1.6 实验环境

本次实验环境与 lab13 完全相同。

本次实验的步骤是：

1) 准备好 lab13 的实验环境，UCAS_CDE_axi。

2) 在 myCPU 中添加 tlb 例外的支持（需要进行取指、数据访问的地址翻译）。

3) 打开 UCAS_CDE_axi/soft/tlb_func/start.S，将第 5 行的宏定义 TEST_TLB_EXCEPTION 的值从 0 修改为 1。重新编译 tlb_func（先 make reset，再 make）。（如果自己电脑不支持交叉编译，请找其他同学拷贝编译好的 tlb_func 进行后续实验）

4) 打开 myCPU 工程（UCAS_CDE_axi/mycpu_axi_verify/run_vivado/mycpu_prj1/mycpu_prj1.xpr）。

5) 对 myCPU 工程中的 axi_ram 重新定制，此时依然选择加载 tlb_func 的 coe（UCAS_CDE_axi/soft/tlb_func/obj/inst_ram.coe）。

6) 运行 myCPU 工程的仿真（进入仿真界面后，直接点击 run all），开始 debug。

7) 仿真通过后，进行综合、布局布线和生成 bit 流文件，并进行上板验证。