

版本历史

文档更新记录		文档名:	Lab12_TLB MMU 设计（一）	
		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:	2019-11-21	
更新历史				
序号	更新日期	更新人	版本号	更新内容
1	2019/11/21	贾凡	-	草稿。
2	2019/11/26	邢金璋	-	初版。

文档信息反馈: [xingjinzhang@loongson.cn](mailto:xingjinzhang@loongson.cn)

# 1 实验十二 TLB MMU 设计（一）

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) 学会设计 TLB 模块。

本次实验需要参考的文档包括但不限于：

- (1) Lab12 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. 学会设计 TLB 模块。

## 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. 完成：
  - a) 按照课程讲义规定的接口设计 TLB 模块。
  - b) 通过简单的读写和查找测试。
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和调试好的 RTL 代码。本次实验在 2019 年 12 月 3 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 查找的过程。

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 3 日 18:10。

提交内容：纸质档 lab12 实验报告。

### (2) 电子档提交

提交方式：打包上传到 Sep 课程网站 lab12 作业下，每组提交一份。

截止时间：2019 年 12 月 3 日 18:10。

提交内容：电子档为一压缩包，文件名是“lab12\_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）。

lab12_箱子号/	目录，lab12 作品。
lab12_箱子号.pdf/	Lab12 实验报告，实验报告模板参考“Lab03 实验报告模板_仅供参考.docx”
--tlb/	目录，自实现 TLB 模块
--tlb_test.bit	功能测试 bit 文件

## 1.6 实验环境

本次实验的环境与之前的环境不相同。本次实验的环境是针对 TLB 模块的验证，验证环境中，有 tlb\_test 模块向 tlb 发出读写和查找请求，默认 tlb 是 16 项，共有 16 次写，16 次读和 26 次查找操作。

本次实验环境的目录如下表所示：

tlb_test/	实验环境目录。
--rtl/	包含 TLB 模块以及测试模块的设计代码目录。
--tlb	目录，其中应当包含 TLB 模块，需要同学们自己完成。
--tlb_test.v	tlb_test 的顶层文件。
--testbench/	功能仿真验证目录。
--testbench.v	仿真顶层。
--run_vivado/	Vivado 工程的运行目录。
--tlb_test.xdc	Vivado 工程设计的约束文件。
--tlb_test/	创建的 Vivado 工程，名字就叫 tlb_test。
--tlb_test.xpr	Vivado 创建的工程文件。

本次实验的步骤是：

- 1) 准备好 lab12 的实验环境，tlb\_test。
- 2) 阅读讲义和任务书，开始编写 TLB 模块的代码，该模块名需要命名为“tlb”，input/output 端口参见讲义第九章第 2 小节“TLB 模块设计指导”。
- 3) 将写好的代码 tlb.v 拷贝到 tlb\_test/rtl/tlb/目录中。
- 4) 打开 tlb\_test 工程（tlb\_test/run\_vivado/tlb\_test/tlb\_test.xpr）。
- 5) 通过“Add Sources”将编写好的 tlb.v 添加到工程中。
- 6) 运行 tlb\_teset 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。

## 1.7 实验说明

### (1) 模块接口

本实验中的 TLB 接口定义参见课程讲义第九章第 2 节。

在实验中使用 generate 语法可以显著减少代码量。

### (2) 仿真效果

在仿真时，会有 16 次写，16 次读以及 26 次查操作，所有操作都完成后会打印 PASS。

```
[ 2705 ns] OK!!!write
.....
```

```
Test end!
```

```
----PASS!!!
```

如果仿真中发现错误，请进行 debug，需要观察 TLB 接口的访问，明了该次请求的效果，然后查看 TLB 的读出数据是否与预期效果相同。

### (3) 上板效果

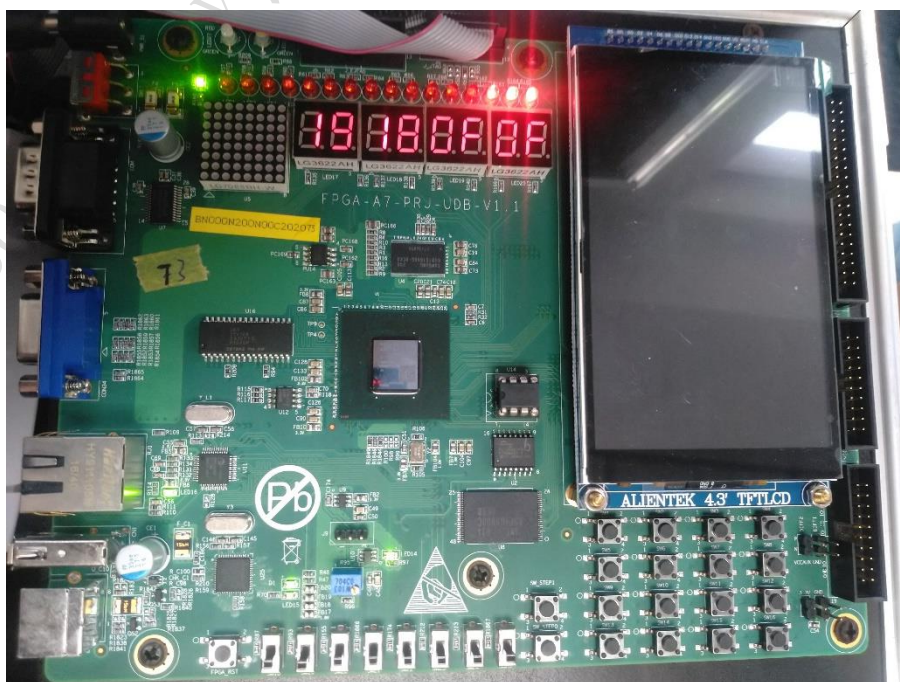


图 1:测试成功时显示的画面

---

第一阶段上板运行时，需要看到数码管如下变化：

- (1) 首先是写操作（W），最右侧的数码管会从 0x00 累加到 0x0f，此后 LED 等的最右边一个亮起，表示写操作完成；
- (2) 之后是读和查同时进行，相应的数码管也会开始累加：
  - a) 对于读操作（R），会进行 16 次读，次右侧的数码管会从 0x00 累加到 0x0f
  - b) 对于 0 号查操作（S0），会进行 13 次查（查偶数次请求），次左侧的数码管会按步长为 2 从 0x00 加到 0x18，也就是 0、2、4……
  - c) 对于 1 号查操作（S1），会进行 13 次查（查奇数次请求），最左侧的数码管会按步长为 2 从 0x01 加到 0x19，也就是 1、3、5……
- (3) 到 LED 的右侧三个灯全部亮起的时候表明测试完成。此时正确的数码管显示是 0x19180f0f。如果数码管停在其他数值上，表示上板失败。