

版本历史

文档更新记录		文档名:	Lab06_在流水线 CPU 中添加运算类指令	
		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:	2019-09-25	
更新历史				
序号	更新日期	更新人	版本号	更新内容
1	2019/09/25	贾凡	-	草稿。
2	2019/09/29	邢金璋	V0.1	初版。

文档信息反馈: xingjinzhang@loongson.cn

1 实验六 在流水线 CPU 中添加运算类指令

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）。
- (5) 了解 CPU 流水线结构。

通过本章节的学习，你将获得：

- (1) 深入理解译码和执行两个流水级的工作过程。
- (2) 掌握在流水线 CPU 中添加运算类指令的方法。

本次实验需要参考的文档包括但不限于：

- (1) Lab06 任务书（本文档）。
- (2) 体系结构研讨课总讲义之第五章“在流水线 CPU 中添加运算类指令”。
- (3) 体系结构研讨课总讲义之附录 D “‘体系结构研讨课’MIPS 指令系统规范”。

1.1 实验目的

1. 加深对流水线结构的理解。
2. 学会在流水线 CPU 中添加运算类指令的方法。

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

本次实验只有一个子任务：

1. 在实验五的 CPU 代码基础上添加更多的指令，具体包括算术逻辑运算类指令 ADD、ADDI、SUB、SLTI、SLTIU、ANDI、ORI、XORI、SLLV、SRAV、SRLV，乘除运算类指令 MULT、MULTU、DIV、DIVU，以及乘除法配套的数据搬运指令 MFHI、MFLO、MTHI、MTLO。运行 func_lab6，要求成功通过仿真和上板验证。
2. 本次实验要求每个人独立提交实验报告和调试好的 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（每人一份）和调试好的 RTL 代码。本次实验在 2019 年 10 月 15 日进行检查。

现场检查，分仿真检查和上板检查：

- 1) 仿真检查：对照波形进行描述新加入的指令的执行过程。
- 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，每人都必须要有。

截止时间：2019 年 10 月 15 日 18:10。

提交内容：纸质档 lab6 实验报告。

(2) 电子档提交

提交方式：打包上传到 Sep 课程网站 lab6 作业下，每人都必须要有。

截止时间：2019 年 10 月 15 日 18:10。

提交内容：电子档为一压缩包，文件名是“lab6_箱子号_学号.zip”，目录层次如下（请将其中的“箱子号”替换为本组箱子号，“学号”替换为自己的学号）。

lab6_箱子号_学号/	目录，lab6 作品。
--lab6_箱子号_学号.pdf/	Lab6 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

本次实验的硬件环境沿用 lab3 发布的 UCAS_CDE，软件环境使用新发布的软件环境 func_lab6.zip，和 func_lab4 相比，本次软件加入了对前述的新加入指令的测试内容。

本次实验的步骤是：

- 1) 准备好 lab5 的实验环境，UCAS_CDE，该环境也会作为 lab6 的实验环境。
- 2) 将 lab6 发布的软件程序 func_lab6.zip，解压后，将 func_lab6/拷贝到 UCAS_CDE/soft/目录里，与 func_lab4 同层次。
- 3) 打开 cpu132_gettrace 工程（UCAS_CDE/cpu132_gettrace/run_vivado/cpu132_gettrace/cpu132_gettrace.xpr）。
- 4) 对 cpu132_gettrace 工程中的 inst_ram 重新定制，此时选择加载 func_lab6 的 coe（UCAS_CDE/soft/func_lab6/obj/inst_ram.coe）。
- 5) 运行 cpu132_gettrace 工程的仿真（进入仿真界面后，直接点击 run all 等待仿真运行完成），生成新的参考 trace 文件 golden_trace.txt（UCAS_CDE/cpu132_gettrace/golden_trace.txt）。要等仿真运行完成，golden_trace.txt 才有完整的内容。
- 6) 打开 myCPU 工程（UCAS_CDE/mycpu_verify/run_vivado/mycpu_prj1/mycpu_prj1.xpr）。

-
- 7) 对 myCPU 工程中的 inst_ram 重新定制，此时选择加载 func_lab6 的 coe (UCAS_CDE/soft/func_lab6/obj/inst_ram.coe)。
 - 8) 运行 myCPU 工程的仿真（进入仿真界面后，直接点击 run all），开始 debug。
 - 9) 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。