

版本历史

文档更新记录		文档名:	Lab03_静态 5 级流水 MIPS CPU 实现	
		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:	2017-09-20	
更新历史				
序号	更新日期	更新人	版本号	更新内容
1	2017/09/20	邢金璋	V0.1	初版。

文档信息反馈: xingjinzhang@loongson.cn

1 实验三 静态 5 级流水 MIPS CPU 实现

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 Vivado2017.1 或 Vivado2017.2 的电脑一台。
- (2) 较为熟练使用 Vivado 工具。
- (3) 熟悉龙芯体系结构实验箱（Artix-7）。
- (4) 已完成实验一和实验二。

通过本章节的学习，你将获得：

- (1) CPU 设计中流水线切分的原理和实现。
- (2) CPU 设计中三类相关（数据、控制和结构）的产生原因、影响和处理。
- (3) 一个小型较完备指令集的理解。
- (4) 嵌入式处理器中性能测试程序和原理。

在本章节的学习过程中，你可能需要查阅：

- (1) 文档“A05_“体系结构研讨课”MIPS 指令系统规范”。
- (2) 文档“LEC03_CPU 实验开发环境使用说明”。
- (3) 文档“LEC04_仿真调试说明”。
- (4) 文档“A07_交叉编译工具链安装”。

1.1 实验目的

1. 进一步熟悉实验平台。
2. 迁移组成原理的多周期实验到本学期实验平台上。
3. 熟悉并运用 verilog 语言进行电路设计。
4. 为后续实验打好基础。

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

1. 设计一款静态 5 级流水简单 MIPS CPU。
2. 本次实验分四个阶段完成，时间跨度四周课时（9 月 26 号—10 月 31 号，中间一周国庆放假）。
3. 本次实验要求延续 lab2 实验中的以下要求：
 - (1) CPU 复位从虚拟地址 0xbfc00000 处取指。
 - (2) CPU 虚实地址转换采用：虚即是实。
 - (3) CPU 对外访存接口为取指、数据访问分开的同步 SRAM 接口。
 - (4) CPU 只实现一个操作模式：核心模式，不要求实现其他操作模式。
 - (5) 不要求支持例外和中断。

-
- (6) CPU 顶层连出写回级的 debug 信号，以供验证平台使用。
4. 整个实验中，最后要求实现 MIPS I 指令集，参考文档“A05_“体系结构研讨课” MIPS 指令系统规范”，除了 ERET（非 MIPS I）、MTC0、MFC0、BREAK、SYSCALL 指令，其余指令均要求实现，共 56 条指令。
 5. 本次实验四个阶段，每个分阶段设有底线任务，并对应一个功能测试程序，要求在验证平台上运行功能测试通过。四个阶段的功能测试程序为：lab3_func_1、lab3_func_2、lab3_func_3、lab3_func_4，将会在实验过程中陆续发布。
 6. 本次实验，实验报告要求在四个阶段各提交一份。但报告模块会大大简化，“A06_实验报告模板_v0.2”将在后续发布。
 7. 第一阶段（10 月 10 号检查），要求至少完成如下设计：
 - (1) 至少支持 lab2 要求的 15 条指令：LUI、ADDU、ADDIU、BEQ、BNE、LW、OR、SLT、SLTI、SLTIU、SLL、SW、J、JAL、JR。
 - (2) CPU 微结构为静态 5 级流水。
 - (3) 要求实现 MIPS 架构的延迟槽技术，延迟槽不再设定为 NOP 指令，可能是任意指令。
 - (4) 可以不用考虑数据相关，第一阶段提供的测试程序 func_1 不存在指令间的数据相关。
 - (5) 控制相关由分支指令造成，通过延迟槽技术可以完美解决。
 - (6) 结构相关即某一级流水停顿了，会阻塞上游的流水级。
 - (7) 要求仿真和上板运行 lab3_func_1 通过。
 - (8) 提交第一阶段的实验报告和 lab3-1 作品。
 8. 第二阶段（10 月 17 号检查），在第一阶段的基础上，要求至少完成如下设计：
 - (1) 至少新增如下 16 条指令：ADD、ADDI、SUB、SUBU、SLTU、AND、ANDI、NOR、ORI、XOR、XORI、SLLV、SRA、SRAV、SRL、SRLV。
 - (2) 要求考虑数据相关，func_2 存在指令间的数据相关。
 - (3) 请自行选择数据相关的处理方式。
 - (4) 要求仿真和上板运行 lab3_func_2 通过。
 - (5) 提交第二阶段的实验报告和 lab3-2 作品。
 9. 第三阶段（10 月 24 号检查），在第二阶段的基础上，要求至少完成如下设计：
 - (1) 至少新增如下 15 条指令：DIV、DIVU、MULT、MULTU、MFHI、MFLO、MTHI、MTLO、BGEZ、BGTZ、BLEZ、BLTZ、BLTZAL、BGEZAL、JALR。
 - (2) 要求数据相关采用前递处理。
 - (3) 乘除法指令实现可以调用 Xilinx 的乘除法 IP，后续会发布调用乘除法 IP 的方法。
 - (4) 推荐能力有余的同学自行编写乘除法器，乘法采用 booth 算法+华莱士、除法采用迭代算法，依据实现情况在本课程最后总分上额外加 1-5 分。
 - (5) 要求仿真和上板运行 lab3_func_3 通过。
 - (6) 提交第三阶段的实验报告和 lab3-3 作品。
 10. 第四阶段（10 月 31 号检查），在第三阶段的基础上，要求至少完成如下设计：
 - (1) 至少新增如下 10 条指令：LB、LBU、LH、LHU、LWL、LWR、SB、SH、SWL、SWR。
 - (2) 要求仿真和上板运行 lab3_func_4 通过。
 - (3) 要求仿真和上板运行性能测试程序 Coremark 和 dhrystone 通过，后续会发布性能测试程序。
 - (4) 要求计算自实现 myCPU 的性能。
 - (5) 推荐尽量优化 myCPU 的性能，性能分数会影响本实验的评分，甚至考虑全班整体实现情况，可能会采取总分上额外加分的奖励。

(6) 提交第四阶段的实验报告和 lab3-4 作品。

1.4 实验环境

本次实验实验环境与 lab2 基本一致，只是 CPU 实验开发环境针对新发现的问题进行了优化，目前是 ucas_CDE_v0.2。实验环境介绍请参考“LEC03_CPU 实验开发环境使用说明”。

ucas_CDE_v0.2 中 func 放置在 soft 目录下，整个实验过程中陆续发布 lab3_func_1、lab3_func_2、lab3_func_3、lab3_func_4。

1.5 实验说明

1.5.1 myCPU 实现指令集

本次实验四个阶段，总共要求实现 MIPS I 指令集中的 56 条指令，具体划分见 1.3 节实验任务。

关于指令的详细定义可以参考文档“A05_“体系结构研讨课”MIPS 指令系统规范”。

1.5.2 myCPU 顶层接口

本实验要求 myCPU 同 lab2，按以下接口封装。

表 1-1 myCPU 顶层接口信号描述

名称	宽度	方向	描述
时钟与复位			
clk	1	input	时钟信号，来自 clk_pll 的输出时钟
resetn	1	input	复位信号，低电平同步复位
取指端访存接口			
inst_sram_en	1	output	ram 使能信号，高电平有效
inst_sram_wen	4	output	ram 字节写使能信号，高电平有效
inst_sram_addr	32	output	ram 读写地址，字节寻址
inst_sram_wdata	32	output	ram 写数据
inst_sram_rdata	32	input	ram 读数据
数据端访存接口			
data_sram_en	1	output	ram 使能信号，高电平有效
data_sram_wen	4	output	ram 字节写使能信号，高电平有效
data_sram_addr	32	output	ram 读写地址，字节寻址
data_sram_wdata	32	output	ram 写数据
data_sram_rdata	32	input	ram 读数据
debug 信号，供验证平台使用			
debug_wb_pc	32	output	写回级（多周期最后一级）的 PC，因而需要 mycpu 里将 PC 一路带到写回级
debug_wb_rf_wen	4	output	写回级写寄存器堆(regfiles)的写使能，为字节写使能，如果 mycpu 写 regfiles 为单字节写使能，则将写使能扩展成 4 位即可。
debug_wb_rf_wnum	5	output	写回级写 regfiles 的目的寄存器号
debug_wb_rf_wdata	32	output	写回级写 regfiles 的写数据

1.5.3 myCPU 复位 PC

要求多周期 CPU 复位从 0xbfc00000 处取指（MIPS 架构规范要求如此，以后解释）。由于本实验 inst 和 data ram 地址宽度有限，并没有使用到 32 位地址的高 12 位，因而 CPU 复位 PC 为 0xbfc00000 和为 0x00000000 效果一样，但我们还是要求尽量符合规范。

1.5.4 myCPU 虚实转换

MIPS 常用虚实转换机制包括：固定地址映射（FMT）和 TLB 机制。但本次实验均不要求实现这两种机制，我们默认虚地址即是实地址。

1.5.5 myCPU 延迟槽实现要求

本次实验要求实现延迟槽技术，延迟槽里可存放任何指令。（请思考延迟槽里放分支指令会如何？可以查阅 MIPS 官方手册。）

1.5.6 myCPU 操作模式

MIPS 架构 CPU 操作模式有核心模式、监管模式和用户模式。但本次实验只要求实现核心模式，即认为程序用于运行与核心模式。

1.5.7 myCPU 不要求实现项

本次实验不要求是实现中断和例外，不要求实现任何系统控制寄存器。

1.5.8 myCPU 验证步骤

参考 lab2 和开发环境的使用。

1.5.9 myCPU 验证结果

本次实验验证结果的检查类似 lab2，详细请阅读 lab2 的文档或者验证平台的文档。

(1) 仿真验证结果

仿真结果正确判断有两种方法。

第一种方法，也是最简单的，就是看 Vivado 控制台打印 Error 还是 PASS。正确的控制台打印信息如下图。

```

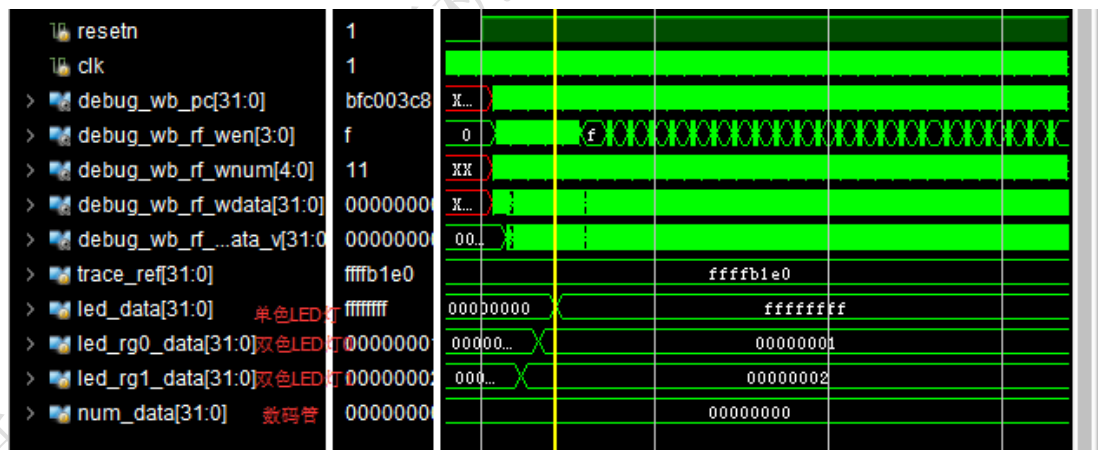
[2112000 ns] Test is running, debug_wb_pc = 0xbfc17530
[2122000 ns] Test is running, debug_wb_pc = 0xbfc17d78
[2132000 ns] Test is running, debug_wb_pc = 0xbfc185c0 每隔10000ns, 打印一次debug_wb_pc
[2142000 ns] Test is running, debug_wb_pc = 0xbfc18e08
[2152000 ns] Test is running, debug_wb_pc = 0xbfc19634
[2162000 ns] Test is running, debug_wb_pc = 0xbfc19e8c
[2172000 ns] Test is running, debug_wb_pc = 0xbfc1a6d4
[2182000 ns] Test is running, debug_wb_pc = 0xbfc1af1c
----[2184405 ns] Number 8'd15 Functional Test Point PASS!!! 第15个测试功能点PASS!!!
----[2189465 ns] Number 8'd16 Functional Test Point PASS!!!
[2192000 ns] Test is running, debug_wb_pc = 0xbfc105fc
----[2194525 ns] Number 8'd17 Functional Test Point PASS!!!
----[2197365 ns] Number 8'd18 Functional Test Point PASS!!!
----[2200405 ns] Number 8'd19 Functional Test Point PASS!!!
[2202000 ns] Test is running, debug_wb_pc = 0xbfc1b1d4
----[2203645 ns] Number 8'd20 Functional Test Point PASS!!! 第20个测试功能点PASS!!!
=====
Test end! 测试程序结束, 没有错误, 打印PASS!!!
---PASS!!!
$finish called at time : 2205425 ns : File "E:/loongson/arch_ucas/17-18/lab3/ucas_CDE_v0.2/mycpu_v
run: Time (s): cpu = 00:00:31 ; elapsed = 00:02:02 . Memory (MB): peak = 1217.645 ; gain = 0.000

```

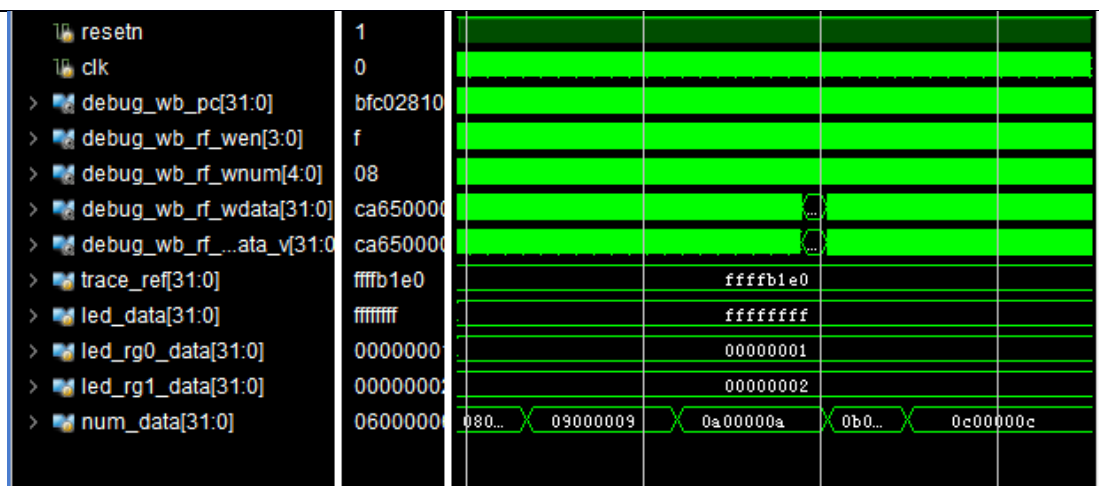
图 1-1 测试通过的控制台打印信息

第二种方法，是通过波形窗口观察程序执行结果 func 正确的执行行为，抓取 confreg 模块的信号 led_data、led_rg0_data、led_rg1_data、num_data：

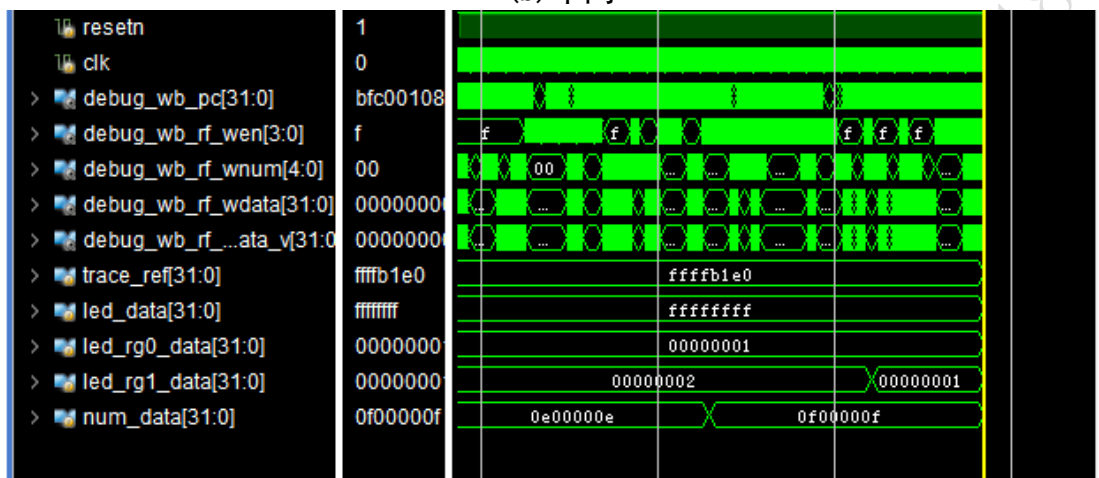
- (1) 开始，单色 LED 写全 1 表示全灭，双色 LED 写 0x1 和 0x2 表示一红一绿，数码写全 0；
- (2) 执行过程中，单色 LED 全灭，双色 LED 灯一红一绿，数码管高 8 位和低 8 位同步累加；
- (3) 结束时，单色 LED 写全 1 表示全灭，双色 LED 均写 0x1 表示亮两绿，数码管高 8 位和低 8 位数值相同，对应测试功能点数目，lab2 中为 0xf，lab3 中各阶段测试功能点数不同。



(a) 开始



(b) 中间



(c) 结束

图 1-2 正确的仿真波形图

(2) FPGA 验证结果

在 FPGA 上验证时其结果正确与否的判断也只有一种方法，func 正确的执行行为是：

- (1) 开始，单色 LED 全灭，双色 LED 灯一红一绿，数码管显示全 0；
- (2) 执行过程中，单色 LED 全灭，双色 LED 灯一红一绿，数码管高 8 位和低 8 位同步累加；
- (3) 结束时，单色 LED 全灭，双色 LED 灯亮两绿，数码管高 8 位和低 8 位数值相同，对应测试功能点数目。

如果 func 执行过程中出错了，则数码管高 8 位和低 8 位第一次不同处即为测试出错的功能点编号，且最后的结果是单色 LED 全亮，双色 LED 灯亮两红，数码管高 8 位和低 8 位数值不同。

最后 FPGA 验证通过的效果图，类似下图，对于 lab2 而言，数码管高 8 位和低 8 位都是显示 0xf。

如果大家想看动态的 FPGA 运行过程，可以对 cpu132_gettrace 下的 Vivado 工程进行综合实现并上板运行，其将显示正确的 FPGA 验证运行过程。

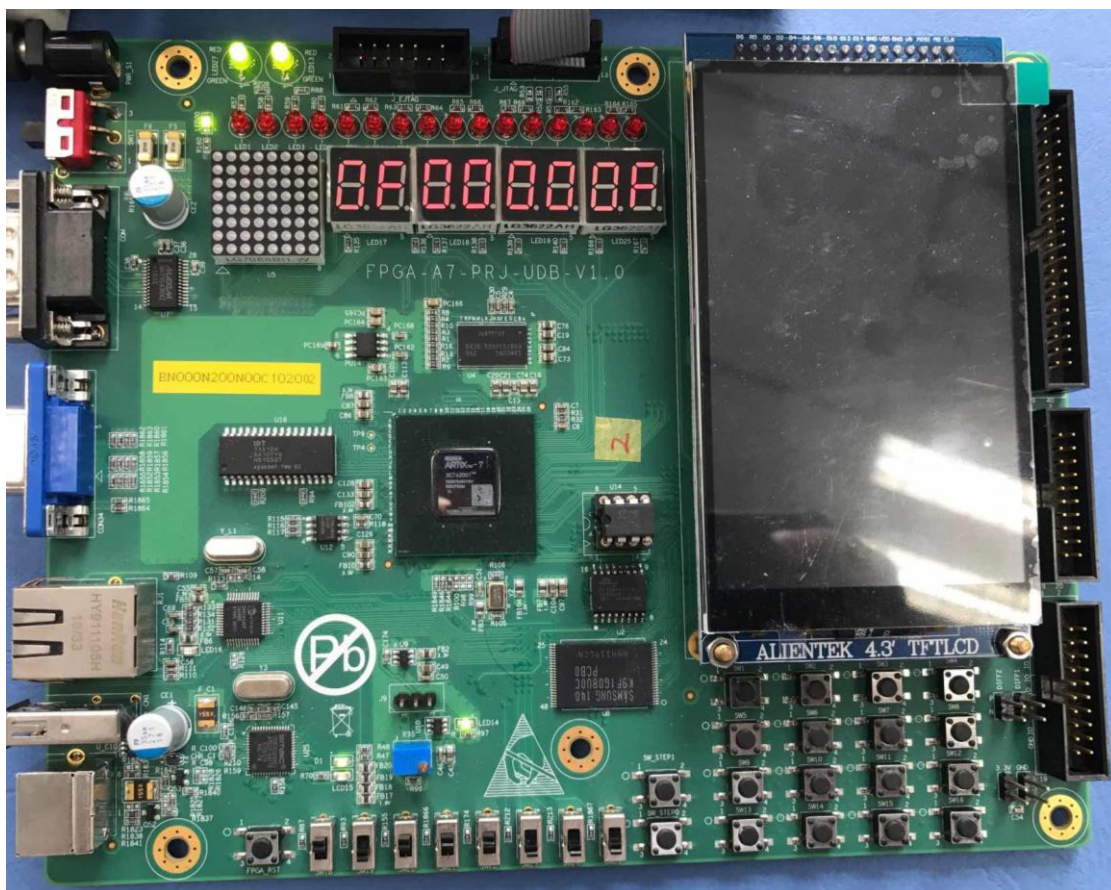


图 1-3 FPGA 验证正确效果图

1.6 实验提交

本次实验要求大家按组完成，小组总共提交一份作品。四个阶段分别提交一次，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，每组都必须要有。

截止时间：2017 年 10 月 10 日、10 月 17 日、10 月 24 日、10 月 31 日 18:00。

提交内容：纸质档 lab3 各阶段实验报告，分别记为 lab3-1、lab3-2、lab3-3、lab3-4。

实验报告模板参考“A06_实验报告模板_v0.2”。

(2) 电子档提交

提交方式：打包上传到 Sep 课程网站 lab2 作业下，每组都必须要有。

截止时间：2017 年 10 月 10 日、10 月 17 日、10 月 24 日、10 月 31 日 18:00。

提交内容：电子档为一压缩包，以第一阶段提交为例，目录层次如下（请将其中的“**组号**”，替换为本组组号）。

-lab3-1_组号/	目录，lab3 作品。
--lab3-1_组号.pdf/	Lab3 验报告，实验报告模板参考“A06_实验报告模板 v0.2
--myCPU /	目录，自实现 CPU 源码。目录请加一个 readme，简单描述下各文件。