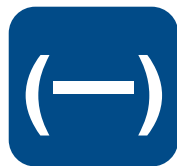




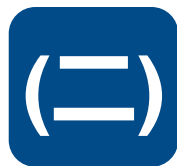
天津大学
Tianjin University

实验三：32位RISC指令集仿真器的设计与实现

实验三



实验简介

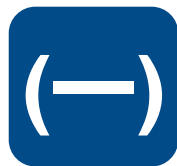


实验内容

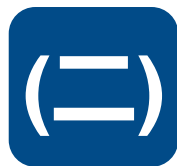


实验评分依据

实验三



实验简介



实验内容



实验评分依据

实验简介

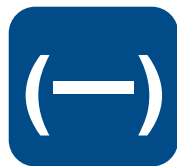
- 采用高级语言（C/C++、Java或Python等）设计面向32位**MIPS**或**LoongArch**指令集子集的指令集仿真器 -- **TEMU**，用于实现对程序执行过程的模拟和调试。该指令集仿真器可模拟常见指令，支持单步执行、断点、显示寄存器信息、显示特定存储单元信息等功能，为后续的处理器的设计实验提供**仿真测试工具**。

实验分组

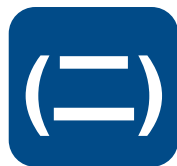
- **最多4人**一组，不同老师的学生不能组队；
- 每组指定一位组长，组长年底实验三、四**加3分**；
- 扫描右侧二维码填写分组，并确定每组所选择的目标指令集
 - MIPS32
 - Loongarch32



实验三



实验简介



实验内容



实验评分依据

平台和开发语言

- 操作系统: Linux, Windows
- 版本控制: gitee、github、gitlab
- 编程语言: C/C++, Java, Python

实验内容

□ TEMU可以是以下两种呈现方式：

- 图形界面（鼓励，不强制）
- 控制台交互命令（**已提供初始框架代码**）

□ TEMU的输入可以选择以下三种形式：

- 汇编程序
- 可执行文件（ELF文件）
- 可执行文件中的代码段和数据段（初始代码框架已提供） **推荐**

指令集要求 – Level1 (MIPS32)

□ MiniMIPS32_Lite指令集

- 32个通用寄存器，小端字节序
- 共计17条指令（14条必做 + 3条随机）
 - ✓ **必做指令：** ori、lui、addu、bne、lw、sw、andi、or、xor、addiu、beq、lb、sb、sll
 - ✓ **随机指令：**（算术运算：add、addi、sub、slt），（移位运算：sllv、srav、sra、srlv），（分支转移：jalr、bgez、blez、bltz），3类指令中各随机1条指令。

指令集要求 – Level2 (Loongarch32)

□ Loongarch32_Lite指令集

- 32个通用寄存器，小端字节序
- 共计17条指令（14条必做 + 3条随机）
 - ✓ **必做指令：** addi.w、add.w、lu12i.w、pcaddu12i、or、ori、andi、xor、beq、bne、st.w、ld.w、st.b、ld.b
 - ✓ **随机指令：**（算术运算：slt、sltu、slti、sltui），（移位运算：sll.w、srl.w、sra.w、srai.w），（分支转移：blt、bge、bltu、bgeu），3类指令中各随机1条指令。

存储管理

- 重启复位后的地址设置为**0x80000000**
- 对于TEMU而言，测试程序的存储空间分配：
 - **.text段（代码段）** 位于0x80000000 ~ 0x8000FFFF
 - **.data段（数据段）** 位于0x80010000 ~ 0x8001FFFF
 - 采用固定地址映射，**虚拟地址最高位清“0”即可**
 - 与最终的功能测试程序保持一致
- TEMU暂可不支持外设

TEMU的命令格式（控制台交互命令）

| 命令 | 格式 | 使用举例 | 说明 |
|--------|-------------|----------------|--|
| 帮助* | help | help | 打印命令的帮助信息 |
| 继续运行* | c | c | 继续运行暂停的程序 |
| 退出* | q | q | 退出TEMU |
| 单步执行 | si [N] | si 10 | 程序单步执行N条指令后暂停，当N没有给出时，缺省为 1。 |
| 打印程序状态 | info SUBCMD | info r | 打印寄存器状态 |
| | | info w | 打印监视点状态 |
| 扫描内存 | x N EXPR | x 10 \$t0 | 求出表达式EXPR的值，将结果作为起始内存地址，以16进制形式输出连续的N个4字节。 |
| 设置监视点 | w EXPR | w \$at==0x1010 | 当表达式EXPR的值发生变化时，暂停程序执行。 |
| 删除监视点 | d N | d 2 | 删除序号为N的监视点 |

*: 已经实现

TEMU指令仿真输出文件 (Golden Trace)

| PC值 | 寄存器编号 | 待写入寄存器的值 |
|----------|-------|----------|
| bfc00000 | 08 | ffffffff |
| bfc00004 | 08 | ffffffff |
| bfc006b8 | 04 | bfaf0000 |
| bfc006bc | 04 | bfaff008 |
| bfc006c0 | 05 | bfaf0000 |
| bfc006c4 | 05 | bfaff004 |
| bfc006c8 | 11 | bfaf0000 |
| | | |
| | | |

记录每条指令的PC和写寄存器的信息

不包含分支指令和Store指令 (折中)

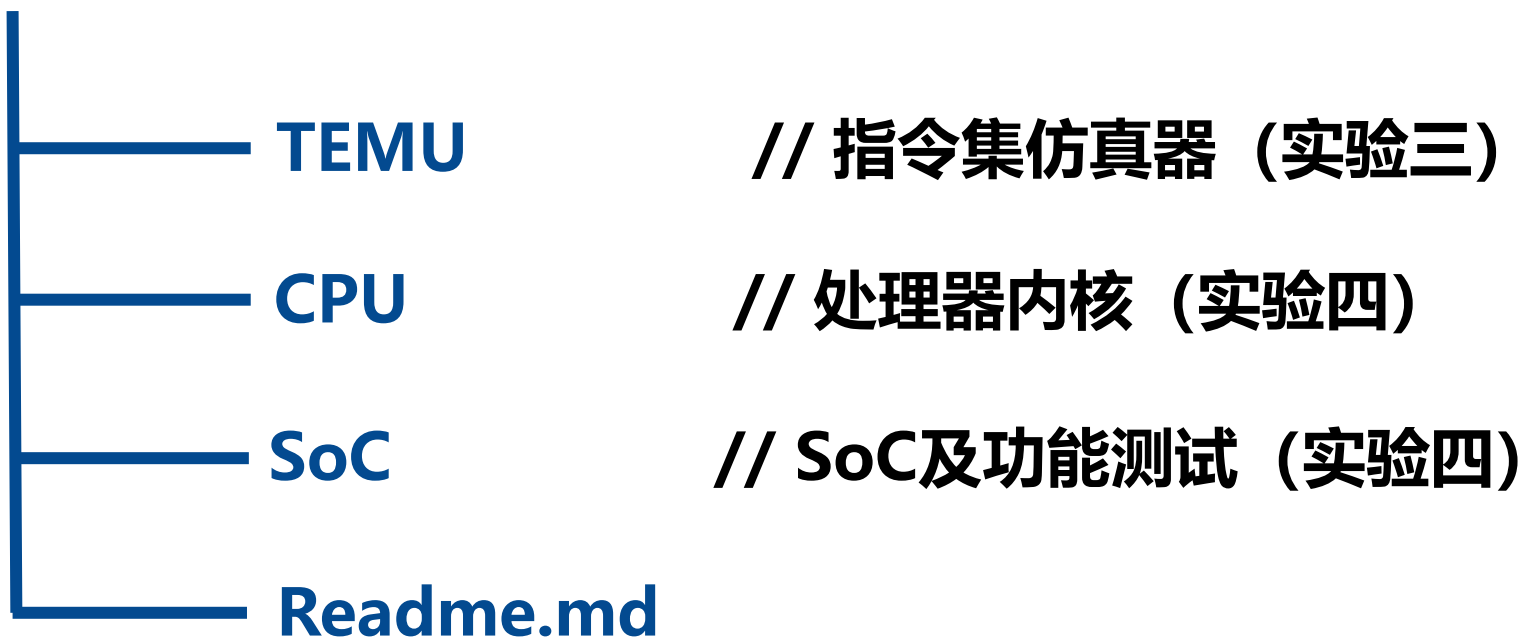
与后续所设计的CPU进行交叉验证

□ TEMU代码框架（控制台交互命令方式）

- 根据各组任务等级，按如下地址进入相应仓库
 - ✓ <https://gitee.com/wjztju/minimip32-lite> (Level 1)
 - ✓ <https://gitee.com/wjztju/loongarch-lite> (Level 2)
- 点击右上角“Fork”，将代码框架导入各组自己账户
- 最后从各自账户git clone整个框架代码，完成后续开发

目录结构

根 目 录



实验源代码和资料^{2/2}

□ 随机指令

智慧树 → 资源管理 → 其他资源 → 实验任务 → 随机指令分配.pdf

□ 交叉编译器和安装配置手册

智慧树 → 资源管理 → 其他资源 → MIPS32编译环境 / Loongarch编译环境

□ TEMU主要功能的实现思路

智慧树 → 资源管理 → 其他资源 → 实验资料 → TEMU仿真器.pdf

□ 指令手册

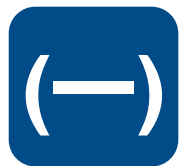
智慧树 → 资源管理 → 其他资源 → MIPS32指令规范 / Loongarch指令规范

□ MIPS Simulator (图形界面仿真器参考)

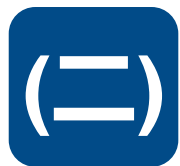
QtSpim (跨平台, <http://pages.cs.wisc.edu/~larus/spim.html>)

Mars (Linux, <http://courses.missouristate.edu/KenVollmar/MARS/>)

实验三



实验简介



实验内容



实验评分依据

评分依据

- Level1/2起评分均为100，但**Level2的最终得分 = 实际分数 * 1.05**
- 支持所要求的指令 (**35分**)
- 单步执行、打印寄存器状态、扫描内存 (**15分**)
- 表达式求值 (**15分**)
- 设置监视点、删除监视点 (**15分**)
- 正确输出Golden Trace文件 (**15分**)
- 支持图形界面 (**5分**)

注意事项

- 框架代码运行环境与NEMU尽量一致。
- 使用git进行版本控制，以组为单位进行远程仓库同步。
- **阶段性更新远程仓库（更新频率 ≤ 3 天），并为工程打标签（git tag），标签注明代码的主要更改，作为评分参考。**
- **本门课程所有工程均使用同一个远程仓库管理！！！！**
- **实验三无需提交实验报告，只需要提交工程代码**