计算机组成原理实验改革——CPU 设计与汇编程序设计的结合

刘芳　金航　梁丰洲　陈志广

(中山大学 数据科学与计算机学院，广州 510006)

摘　要：针对计算机专业学生“喜软怕硬”的普遍情势，立足于提高学生对计算机硬件底层的兴趣和系统能力，通过将基于硬件描述语言的 CPU 设计，与 X86 反汇编和 MIPS 汇编程序设计相结合，对“计算机组成原理实验”的教学内容体系进行了改革。从教学内容、教学组织、教学方式、代码查重和成绩评价方式等方面，介绍了我们在该课程中的教学改革实践与探索。

关键词： 计算机组成原理实验；汇编程序设计；CPU 设计；教学改革实践

1 引言

在错综复杂的新时代背景下，面对信息领域的“卡脖子”难题，急需加强我国计算机系统方向优秀人才储备。如何培养计算机专业学生“造”计算机、而不仅是“用”计算机的能力，需要加强学生以 CPU 和操作系统为代表的基础能力的培养。“计算机组成原理”课程的实验教学，需要探索如何更好地与“计算机组成原理”理论课内容相呼应，加深学生对计算机工作原理的理解。

同时，针对学时数有限、无法单独开设“汇编程序设计”必修课的情况，如何在“计算机组成原理实验”中注重与 CPU 设计相结合，增强学生对 X86 和 MIPS 两种汇编程序设计的兴趣引导和能力培养，进一步提高学生对计算机硬件和系统结构的兴趣、能力与学习积极性，学会软硬件两条腿走路，需要对现有的实验教学内容体系进行改革、整合和优化。

我们对国内外一些在计算机组成原理实验课程建设上颇有成果的高校进行了调研，可以总结出以下几个特点：1.教学内容经典中与时俱进，实验内容在经典 CPU 设计的基础上，注重反映一些新的技术进展和趋势，如流水线处理器、ARM 处理器设计、分布式系统等；2.课堂以启发式教学为主，主讲教师注重引导，层层推进实验，发挥学生的主动学习能力，培养学生的工程实践能力和创新能力；3.重视学生系统能力的培养。体现在实验教学内容既能够承接前导课程（如数字逻辑电路）的所学知识，又能为后续的课程（如操作系统）打下一定的知识基础，强化学生对计算机组成和设计的本质有更整体性的理解。

2 “计算机组成原理实验”课程的教学实践

中山大学数据科学与计算机学院的“计算机组成原理实验”是面向计算机科学与技术专业的二年级本科生，单独设置的一门课程，共 36 个学时。为了给学生提供一个连续的理论与实践环节相结合的学习环境，增强“计算机组成原理”理论课和实验课的内容和进度匹配、避免实验滞后或超前甚至部分脱离于理论内容、学院核心专业课的理论课和实验课均由同一个老师任教。理论课采用计算机经典教材《计算机组成与设计：硬件/软件接口》，该教材以 MIPS 处理器为例，讲解计算机硬件技术、汇编语言、计算机算术、流水线、存储器层次结构以及 I/O 等基本功能。MIPS 作为代表性的高效精简指令集计算机(RISC)，简单优雅，非常适合大学来介绍计算体系结构相关课程。

汇编语言是计算机系统软硬件的分界与桥梁。通过汇编语言课程的学习，可以对计算机组成原理理论课中包括 CPU 体系结构、指令调度方式、存储器管理、基本输入输出接口的理解都会有一个比较本质而且直观的认识。在对汇编语言实际运用的基础上，还能增加对高级程序设计语言的体会，包括变量的组织，地址的访问，循环与分支在机器码中的处理，调

用函数时参数的传递等。对汇编语言的学习，也是了解计算机操作系统和体系结构的最佳切入点。

但由于总学时数有限, 无法单独开设“汇编语言程序设计”课程，因此我们在本实验课程中对学生的 X86 和 MIPS 两种汇编程序设计能力进行初步的培养，为后续学习“操作系统”等计算机核心系统课程打下基础。下面介绍我们在实验课程的一些教学改革和实践工作。

2.1 课程安排

实验分为三个项目：1、二进制拆弹实验（x86 反汇编）：考虑到学生们具有良好的高级语言程序设计基础，首先引入卡耐基梅隆大学的 ICS 课程[1]实验中的二进制拆弹实验——“binary bombs”，增强学生对程序的机器级表示、汇编语言、调试器和逆向工程等方面原理与技能的掌握；2、MIPS 汇编程序设计：由于学生们在理论课中已对 MIPS 汇编程序设计的基本语法和简单程序设计有所了解，学生们可以迅速开展 MIPS 汇编语言程序设计；3、基于 FPGA 的 CPU 设计：基于理论课中对数据通路和微处理器的讲授，在实验课中向同学们介绍数字系统设计的基本原理、方法和流程，学生们自主设计兼容 MIPS 指令集微处理器。

前两个实验的目的是让学生对程序的本质有一定了解，加深对汇编指令和机器指令两个概念的认识，同时掌握程序的（交叉）编译过程和执行过程，弥补了学生在系统知识方面的空白。经过前两个实验的铺垫后，学生亲手设计一个 CPU，进一步加深对计算机组成原理的理解，最后尝试在自己设计的 CPU 上运行实验二中编写的 MIPS 汇编程序，使整个课程形成一个紧密且完整的体系。

2.2 二进制拆弹实验（x86 反汇编）

本实验要求学生掌握x86指令的行为和执行方式，并能够通过读懂x86反汇编代码，了解程序构成的基本结构和运行过程。CMU的“二进制炸弹”中共有六个常规关卡和一个隐藏关卡，每个关卡需输入一个特定字符串，输入正确时则过关，当所有关卡均通过时，则“拆弹成功”，当字符串输入错误时，“炸弹”将“爆炸”。炸弹的每一关均由C语言编写的，覆盖了所有常见的程序结构（如顺序、分支、循环、递归、函数调用等）

本实验采用安装了 Debian 10 操作系统的 Linux 虚拟机环境进行。学生需要在网页上输入自己的学号生成一个属于自己唯一的差异化炸弹，由此避免了学生之间存在的抄袭问题。“炸弹”程序的差异化体现在两方面：一方面是每个关卡均有三个随机的题目；另一方面是每个题目中出现的整数、字符串等需要学生通过阅读反汇编探索的信息均是随机产生的。生成炸弹时，系统对每一关都将从题库中随机抽取一道题目作为当前关卡的题目，并随机生成题目中涉及的数据，编译出一个唯一的二进制炸弹程序。

学生生成自己的炸弹后，在虚拟机环境中解压、运行和调试。调试器采用 GDB[2]调试工具，反汇编则采用 objdump 反汇编器。通过结合阅读反汇编代码和使用 GDB 查看内存及寄存器值，从反汇编代码中读取程序逻辑，从内存和寄存器中获取数据信息，联合二者推断出预期的答案。二进制炸弹服务器搭建在校园网内，每一次学生提交的答案，都会提交到服务器端，并记录下提交的时间和内容。服务器端还提供了一个可视化的页面显示所有，便于教师跟踪学生实验情况。

在完成实验的过程中，学生通过对 x86 汇编代码的理解，知道了程序在汇编层面上执行的机制，用汇编的视角初步深入到计算机内部，对程序的执行和计算机的本质有了更加深入的认识。

2.3 MIPS 汇编程序设计

传统的 MIPS 汇编程序设计大多采用 PCSpim 或 MARS 等模拟器进行的，这些模拟器比较精简，利用其能够实现的程序功能有限，扩展性也不尽人意。在使用模拟器进行教学的模

式下，一个程序的生命周期大致如图 1 中的路线 C 所示，即学生编写好汇编程序后，直接由模拟器对指令序列进行解释执行。该过程并不涉及真正机器码的执行，因此模拟程度有限，难以真实地还原程序的运行情况。

为此，我们尝试使用开源的 QEMU 模拟器[3]，在其提供的类 Linux 环境中运行 MIPS 程序，这也和学生最常接触到的编程环境类似。这样做还有利于和软件层面相结合，让学生更好地理解程序生成和执行的机制和原理。如图 1 中的路线 B 所示，学生使用 MIPS 汇编语言编写完代码后，依次通过交叉汇编器 mips-linux-gnu-as 和交叉链接器 mips-linux-gnu-ld 生成 MIPS 的 ELF 可执行文件，然后 qemu-mips 执行文件，完成一次程序的编译和运行。调试则使用 qemu-mips 的调试模式(-g 参数)开启远程调试服务器，通过跨平台调试器 gdb-multiarch 进行调试。程序的编译和链接使用 GCC 的交叉编译工具链[4]。在最新的软件包仓库中，这些软件很容易被获取到，亦可以将制作好的运行环境虚拟机通过 FTP 等途径发放给学生，免除学生自行搭建实验环境的困难。

图 1 对比了三种实验路线，程序在汇编、链接和执行阶段的不同：路线 A 是最基本也是最常用到的编译方案，即使用 GCC 编译工具链在 x86 平台上编译本平台运行的 x86 程序，其中 as 和 ld 分别是 GNU 的开源汇编器和链接器。路线 B 仿照 x86 下程序的生成流程，引入了交叉编译，基于学生常使用的 x86 平台计算机，编译并模拟运行 MIPS 程序。与路线 A 的不同之处在于利用 GCC 的交叉编译工具链代替 GCC，以在 x86 平台上编译 MIPS 指令集程序，采用模拟器 QEMU 执行。路线 C 是传统教学模式下，使用 MIPS 指令解释器运行，其最大的特点是未从机器角度执行机器码，而是将汇编指令作为文本解释执行，这和机器执行程序的机制不同。

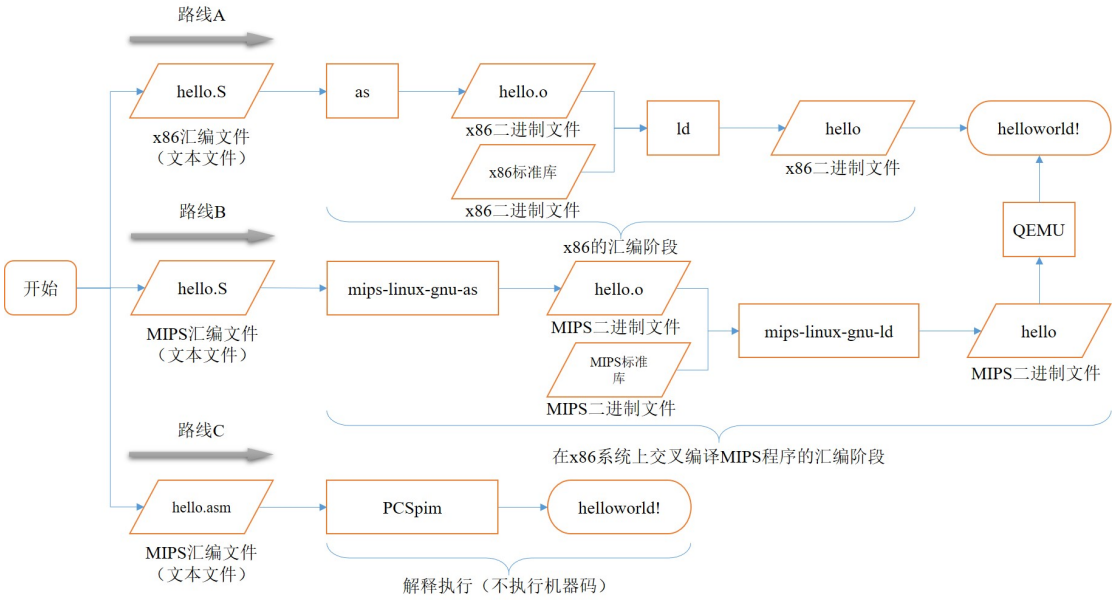

图 1 MIPS 汇编语言实验编译流程

由于交叉编译和远程汇编调试在实际工程项目中会经常用到，因此学生在这个过程中，了解了交叉编译的实现原理和操作流程，开阔了眼界，工程能力也得了训练和培养。

2.4 基于 FPGA 的 CPU 设计（单周期、多周期、流水线）

FPGA 由于具有灵活性和方便性，在云计算、人工智能等领域也正在受到越来越多的关注和应用。实验中采用仿真和上板相结合的方式，实现的指令集为标准 MIPS 指令集，学生将完成仿真实验，并将设计代码烧写到 FPGA 开发板进行测试。

实验环境方面，FPGA 开发板选用 Basys3 开发板，仿真选用 Vivado 或 Modelsim 平台，

使用 Verilog 语言进行开发。开发完成后，首先在仿真平台上运行，随后将程序下载烧写到 FPGA 开发板上。

测试程序方面，我们设计了若干组测试代码段，涵盖了所有类型的标准 MIPS 指令，并针对一些指令的细节（如：溢出、符号扩展等）设计了专门的测试代码。除了需要在开发板上运行测试代码，还要求同学在开发板上调用数码管模块显示数据通路的数据，通过观察内存数据和关键电路时序图判断数据通路的正确性。实现标准的 MIPS 指令集，可以最大程度上还原一台计算机的真正面貌，学生在完成后也会获得一定的成就感，破除学生们对计算机硬件底层的陌生感和畏惧心理，从而提升学生的系统能力。

单周期 CPU 设计实验中，我们要求学生自己根据单周期 CPU 数据通路示意图（图 2）完成各个器件电路的 Verilog 实现，并实现控制信号的产生。为了让学生对控制信号的底层原理有更好的掌握，我们鼓励学生尝试用控制信号真值表方法分析问题并写出逻辑表达式。

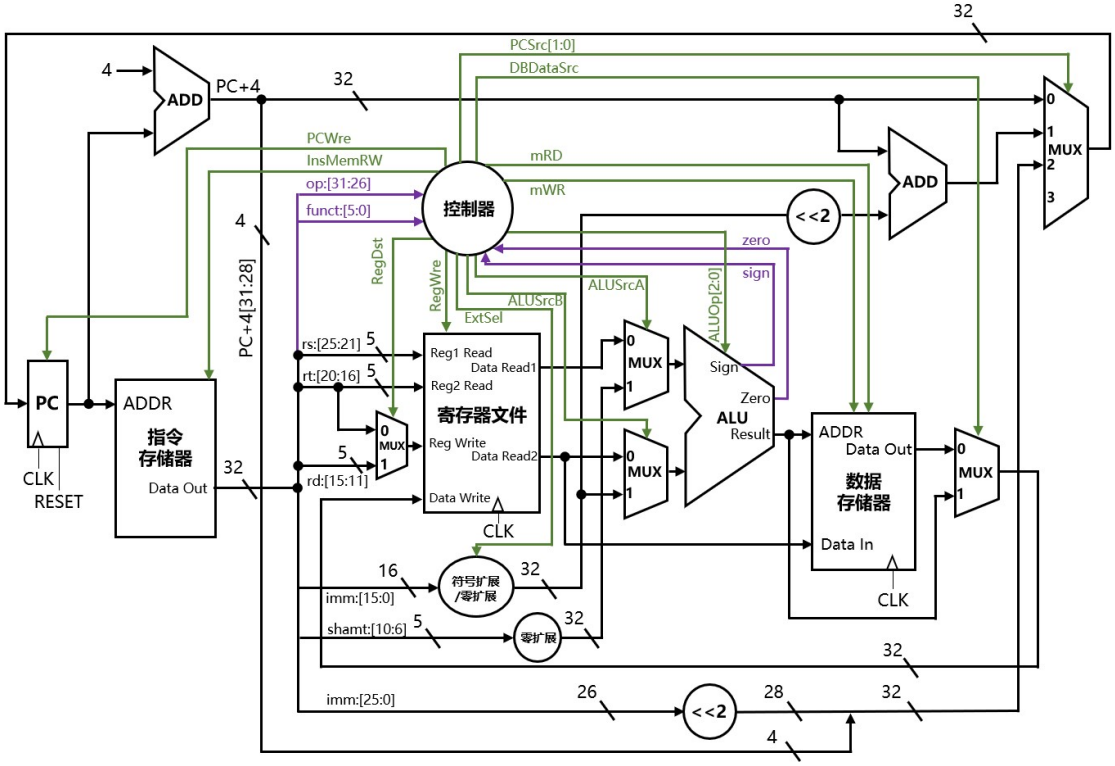

图 2 单周期 CPU 数据通路和控制线路图

完成单周期 CPU 设计后，将进行阶段检查。阶段检查采用现场运行给定测试程序段并观测开发板上的显示和程序寄存器和关键电路波形的方法进行判定。此外，检查的学生还应回答教师或助教针对电路模块或代码提出的疑问，以判定是否为学生独立完成。

在完成单周期 CPU 设计实验的基础上，学生还需要实现多周期 CPU 设计和流水线 CPU 设计。多周期 CPU 设计的核心在于实现状态机的状态转移，如图 3 所示。据学生完成情况反映，极大多数完成了单周期 CPU 设计的学生都能够较为容易地实现多周期 CPU 和流水线 CPU 的设计。在此阶段还加入了部分新的指令（如 SLT、LHU 等指令），进一步完善了学生设计的 CPU，便于学生在自己设计的 CPU 上运行自己设计的 MIPS 汇编程序。

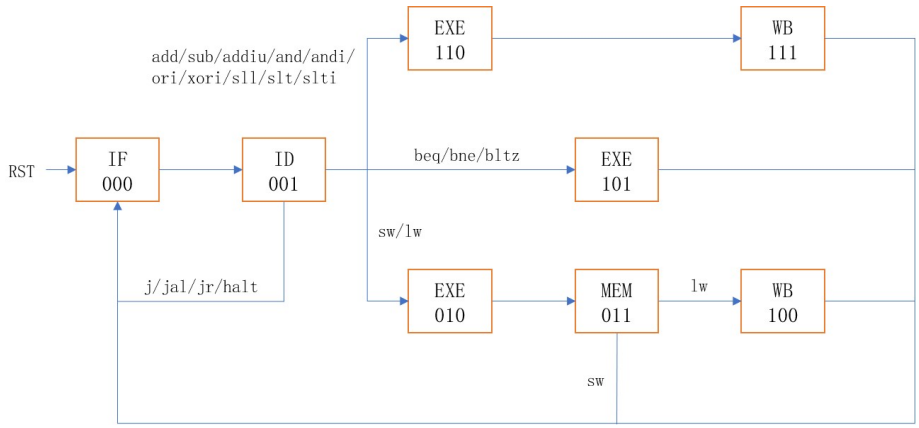

图 3 多周期 CPU 状态转移图

2.5 在自己设计的 CPU 上运行 MIPS 汇编程序

我们在 MIPS 程序设计实验中，要求学生使用汇编语言完成一些具有实际意义的题目。在实现自己的 CPU 后，我们以加分的形式鼓励学生将之前所实现的 MIPS 汇编程序移植到 FPGA 平台上运行。从而将软件和硬件相结合，既在以汇编为代表的软件层面上对系统的上层有所感悟，又深入到硬件层面对以 CPU 为代表的系统底层有所理解，实验内容的结合，有助于实现培养计算机系统能力的教学目标。

2.6 考核评价与代码查重

合理设计课程考核方式和评分规则，可以适当鼓励学生更好地完成实验，我们在课程中采用对每一位同学现场验收、对实验代码进行查重、提交实验报告相结合的方式综合考核。

实验考核频率为每个实验验收一次，考核形式包括演示实验过程、实验代码细节问答、学生自主讲述实验过程等。实验评分方面，二进制拆弹、MIPS 汇编程序设计、单周期 CPU 设计、多周期/流水线 CPU 设计四个部分各占总分的 25%，每个部分的分数又是由现场验收和实验报告得分一起构成。

为督促学生独立自主的完成实验，杜绝学生抄袭、小组实验分工不均甚至有的同学“抱大腿打酱油”的现象，本实验采用一人一组，并对学生所提交的代码进行查重。我们采用了斯坦福大学开发的 Moss 系统，这是一套基于文档指纹技术的软件相似性检测系统，对全球所有高校免费开放，具有非常强大的抄袭检测能力，支持对于 Java 语言、8086 汇编、MIPS 汇编、Verilog 语言等 20 余种编程语言的批量雷同检测[5]。该系统还可以指定对比库，和当届甚至往届学生作业进行对比，从多角度防止出现大规模的抄袭现象，为教师方便快捷地把握学生的真实情况以及维护学术诚信提供了条件。

3 结束语

“计算机组成原理实验”是计算机专业的核心实验课程，我们结合本校专业课程开设的实际情况和需求，将基于硬件描述语言的 CPU 设计，与 X86 反汇编和 MIPS 汇编程序设计相结合。通过有益的探索和实践，提升了学生对计算机系统底层的兴趣和理解。我们的经验，对国内同行有一定的借鉴和参考价值。同时，我们也充分认识到，中国芯片、计算机系统设计等领域面临的“卡脖子”问题根源在于优秀人才储备严重不足，我们在教学中要注重用使命感和责任感引导学生，帮忙学生获得一定的成就感，消除学生对计算机硬件底层的陌生感和畏惧心理，培养对计算机硬件知识的兴趣。同时，还需要我们勇于尝试和实践不同的教学方法，对教学内容、环节、过程等进行精心设计，积极开展教学改革和实践探索。

参考文献

[1] Randal E.Bryant, David O'Hallaron. Computer Systems: A Programmer's Perspective (3rd Edition). 2016.

[2] GDB: The GNU Project Debugger. http://www.gnu.org/software/gdb/documentation/.

[3] QEMU documentation. https://www.qemu.org/documentation/.

[4] GCC online documentation. https://gcc.gnu.org/onlinedocs/.

[5] A System for Detecting Software Similarity. http://theory.stanford.edu/~aiken/moss/.

基金项目：
国家自然科学基金重点项目“大规模固态存储系统结构与技术”(项目编号：NSFC 61832020)

文章名称	作者	通讯地址(含邮编)	联系电话	email	已出书或计划出书	主讲课程、教材及出版社（必填）
计算机组成原理实验改革——CPU设计与汇编程序设计的结合	刘芳	广州市中山大学数据科学与计算机学院	13975192203	liufang25@mail.sysu.edu.cn liufang@nudt.edu.cn		计算机组成原理、信息存储技术