

1 课程介绍

亲爱的同学们，你们现在阅读的是“计算机体系结构研讨课”的课程讲义。是的，我将以这种第一人称的方式来写这门课的课程讲义。这主要是受到我的导师胡伟武老师的影响。他在《计算机体系结构》一书中采用了第一人称方式的写作风格，在国内的计算机专业教材中确实不多见。但我在学生时代阅读胡老师的这本教材时，就觉得很亲切，感觉像是在课堂上听他当面对课一般。所以，我也打算在本课程的讲义编写中学习这种行文风格，若有东施效颦之俗，欢迎批评指正。

1.1 本门课程的学习意义

我们这门课名字叫做“计算机体系结构研讨课”，也就是同学们俗称的“体系结构实验课”。实验确实是我们这门课的主要组成。为什么？因为计算机系统结构专业方向本身的工程导向性很强，传统理论课堂教学侧重于原理性知识的传授，课后练习也多针对单个知识点设计，学生学完之后虽然清楚知道每个基本概念，但往往并不清楚各概念之间的联系，以及每个概念与计算机系统整体之间的关系。国科大能够为数字逻辑、组成原理、体系结构、操作系统和编译原理课程都设置了配套的研讨课，并给予了充分的学时，就是为了解决这方面问题的。你们刚从大二开始接触一些计算机系统结构方面的专业课，这种体会可能还不强烈。我带过的一些研究生在这方面就深有体会，他/她们说最初上胡老师的体系结构课时，很多东西都学得“好像是明白了”，虽然作业会做考试能过，但要给别人讲就感到有些吃力了，至于怎么做个 CPU 出来就更是一头雾水，后来回到课题组，深入参与到龙芯处理器核的研制工作中后，就都明白了，觉得“其实挺简单的”。我们没有条件让所有的同学都去龙芯实习个一年半载，但是我们可以通过一定工程规模的课程实验让同学们得到同样效果的训练。具体来说，我们通过安排系列化的实验任务，让学生自己动手构建起一个简单的计算机系统，学生们在这个过程中自然会厘清系统中各元素之间的相互关系，学会以整体的而非分割的观点来看待各个设计要素。

我们之所以强调系统实践必须达到“工程规模”，是因为只有在实验对象达到一定的规模和复杂度后，才能暴露出复杂系统中存在的较为深刻的问题，才能让学生们对系统有较为深刻的认识。为了保证大多数学生都能开发出具有“工程规模”的系统，我们将整个系统的开发分解成若干子系统的设计开发和将子系统综合成最终系统的过程，从而将全系统的复杂度控制在子系统复杂度上。

这种系统实践过程，用形象的话来说，就是在本科阶段，希望同学们能够通过实践，设计一个自己的 CPU，运行一个自己写的 OS，用自己写的编译器编译出一个应用在上面执行。

我们为什么要花这么大的力气去学好计算机系统结构的这些专业课程呢？有的同学可能会说，我将来想搞人工智能、想搞网络，我又不想去做计算机，我干嘛要学好这些课呢？对于这种观点，我先来问同学们几个问题。这几年，人工智能火到不行，但是每个产品背后的算法都是多年之前甚至很多年之前的研究成果，为什么现在才这么火？其次，这一轮人工智能火热的潮流中，是哪些大公司在主导。是谷歌（Google）、英特尔（Intel）、英伟达（NVIDIA）、赛灵思（Xilinx）、等等。为什么？最后，寒武纪为什么诞生在中科院计算所？

随着半导体工艺日益接近其物理极限，传统通用处理器设计提升性能或者面积效率的方式也将很快碰到其天花板，但是应用需求与现有硬件性能仍存在巨大的差异，于是专用化硬件设计作为一种新的性能优化手段策略得到越来越多的重视。从某种意义上讲，专用化硬件模糊了原有的软件和硬件的界面。像我们现在所熟知的基于通用处理器的计算机系统，其软硬件界面就是指令集（ISA），向上向下又有各个不同的抽象层次，各个层次的界面很清楚。界面清楚的好处是什么？就是大家可以各干各的，拼起来还能跑。但是现在不一样了，软硬件之间的界面被专用化硬件方法模糊了，同样是基于卷积神经网络的深度学习算法的应用实现，各家的硬件加速器的做法都不一样，所以各家的软件自然也就不一样了。以前上层应用软件开发人员主要考虑满足客户需求把功能做对，余下的性能问题都交给底层库、操作系统、编译系统和硬件去了，也就是说交给“造计算机”的人去管了。但现在情况变了，一个有别于普通“码农”的应用软件开发人员，他/她可能面对一个需求，首先要规划哪些交给硬件做、哪些交给软件做，并基于此开展后续的软件开发，调试的时候出了功能或者性能的问题，也知道定位到硬件还是软件。所以，即便是应用软件开发人员，也都要开始具备系统观，更不用说应用层之下各层次的研发人员了。怎么具备系统观呢？别着急，学好我们在本科阶段开设数字逻辑、组成原理、体系结构、操作系统和编译原理这一系列课程，并完成配套的系统实践课程，你自然就有系统观了。

啰嗦了这么多。动机很简单。希望同学们不要一上来就从心底抵触这门课。只要你愿意来上课，我还是有信心不让你失望的。

1.2 本课程与系统结构专业课程的关系

同学们可能已经发现，我总是喜欢把“数字逻辑、组成原理、体系结构、操作系统和编译原理”这五门课放在一起说，这是因为这些专业课程的知识都是围绕“如何造计算机”来组织的，相互之间彼此关联呼应，构成一个有机的整体。计算机系统结构专业涉及的各门课程之间的关系大致如图 1-1 的示意，可以看到课程之间存在的知识概念的相互关联，表 1-1 列出这些相互关联的重要知识概念。

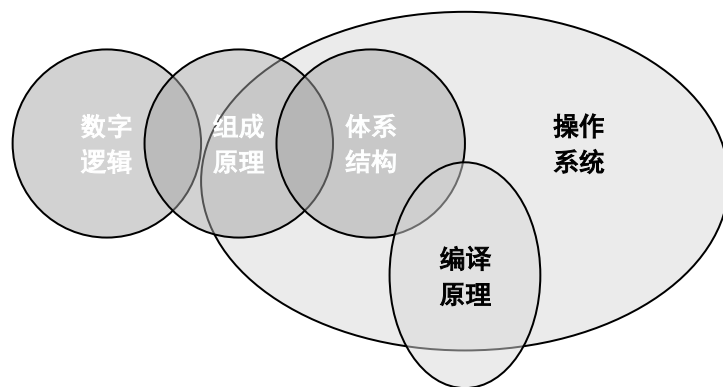


图 1-1 计算机系统专业主要课程相互关系示意

表 1-1 计算机系统专业主要课程重要概念关联

	数字逻辑	组成原理	体系结构	操作系统	编译原理
数字逻辑		数值、数制及编码，算术运算函数（ALU），寄存器传输与控制（Regs, RTL），存储器，状态机设计	数值、数制及编码，算术运算函数（ALU），寄存器及传输与控制（Regs, RTL），存储器，状态机设计	计数器	
组成原理	Datapath（ALU/乘法器），控制器（寄存器传输状态机设计方法），Memory		指令集，Datapath，控制器，算术运算函数（ALU/乘法器），单/多周期/流水 CPU	指令集，单/多周期/流水 CPU	指令集
体系结构				指令集，单/多周期/流水 CPU，中断、DMA，Cache，Memory，虚拟存储，总线	指令集
操作系统		内存管理，应用程序加载，虚拟存储器	内存管理，应用程序加载，虚拟存储器		内存管理，应用程序加载，进程调试
编译原理		程序链接	系统优化	程序加载运行	

上述课程中与本课程实验关系最紧密的是三门课程：“数字逻辑”、“计算机组成原理”（本文有的地方简称为“组成原理”）和“体系结构”，需要同学们较为系统的学习掌握。此外希望同学们有一定的“C 语言”和“汇编语言”的程序设计基础，至少要达到看到代码有点头绪的程度，当然会写会调就更好了。实验有部分内容与“操作系统”和“编译原理”两门课有关，但涉及的知识点不多且以用为主，所以不需要同学们提前系统学习这两门课，相关知识点我们会在教材中必要的地方予以讲解。

围绕本实验课程，“数字逻辑”、“计算机组成”和“体系结构”三门理论课的大体呈现一种递进关系：“数字逻辑”课教会大家设计逻辑电路的基本知识和技能，能设计加法器、移位器这类简单部件；“计算机组成”介绍 MIPS 指令集和简单 CPU 设计；“体系结构”从支撑系统运行和性能分析优化两个角度出发介绍 CPU 的扩展和优化。下面我们把与实验紧密相关的主要知识点都一一罗列出来，算是给同学们提供一个索引。作为实验课，每个知识点的具体内容我就不在讲义中再展开来讲了，同学们应该都可以在自己的理论课教材中查阅到相关内容。不过请同学们注意的是，此处列出的知识点仅是从本课程实验所需来考虑的，没有列出来的并不是不重要。譬如我们没有实验中没有实现浮点运算部件，但是浮点数相关的知识点还是希望大家掌握的。

“数字逻辑”方面

- 1) 数值表示和数制，重点是二进制数、十六进制数。
- 2) 数值的原码表示和补码表示，有符号数、无符号数，溢出。
- 3) 基本逻辑门：与、或、非。
- 4) 布尔代数：逻辑表达式，真值表，逻辑运算的常用运算律。
- 5) 组合逻辑电路和时序逻辑电路的原理。
- 6) $n-2^n$ 变量译码器，如“3-8 译码器”等。
- 7) 不同位宽的数据选择器，如“1 位宽 4 选 1”、“8 位宽 32 选 1”等。
- 8) 一位全加器，串行进位多位全加器。
- 9) 锁存器与触发器。
- 10) 触发器时序分析，触发器的 Setup 延迟、Hold 延迟和 Clock-to-Q 延迟的含义及原理
- 11) 典型状态机设计。
- 12) 计数器设计，如二进制计数器、模- n 计数器。
- 13) 移位寄存器设计。
- 14) 只读存储器 ROM 基本原理。
- 15) 随机存储器 RAM 基本原理。
- 16) 动态存储器 DRAM 基本原理。
- 17) 现场可编程门阵列 FPGA 基本原理。

“计算机组成”方面

- 1) “冯·诺依曼”体系结构。
- 2) RISC 指令集的基本特点。
- 3) 以 MIPS 指令集为例的指令格式：操作码、操作数、寻址。
- 4) 并行进位多位全加器。
- 5) 桶形移位器。
- 6) 补码乘法器。
- 7) 补码除法器。
- 8) 基于多路选择器的数据通路。
- 9) 算术逻辑单元 ALU。
- 10) 寄存器堆，两读一写的 32×32 寄存器堆。
- 11) 指令的取指数据通路、读操作数数据通路、执行数据通路、写回数据通路。
- 12) 指令译码与 CPU 控制逻辑。
- 13) 取指、译码、执行、访存、写回五周期的多周期 CPU 数据通路与控制逻辑设计。
- 14) 取指、译码、执行、访存、写回五级流水 CPU 数据通路与控制逻辑设计。

“体系结构”方面

- 1) 异常：异常事件与异常事件处理。
- 2) 中断：中断识别、中断优先级、中断响应和中断返回。
- 3) 基于页的虚拟存储管理原理：页、页缺失、页分配、页地址映射。
- 4) TLB：TLB 结构、TLB 缺失维护。
- 5) 存储器层次结构：多级 Cache、Cache 基本结构、映射机制、查找过程、替换算法、写策略。
- 6) 总线：基本概念、总线结构、总线的仲裁和通信控制。
- 7) I/O：I/O 寻址方式（独立编址、统一编址），I/O 访问模式（轮询、中断），I/O 数据传输方式（PIO、DMA）。
- 8) 计算机系统性能分析：性能评价指标、常见性能分析建模方法

1.3 课程总体开设思路

基于我到目前为止的教学经验，任何一个班级，上任何一门课，都会同时出现“吃不饱”、“中不溜”和“跟不上”的同学，区别在于这三者的比例，决定这一比例的是课程内容的难易度。想兼顾两头不是个容易的事情，但我们还是想努力去做。

在实验内容上，这学期将延续你们在“计算机组成原理研讨课”所完成的多周期 MIPS CPU 设计，让你们完成 CPU 的流水化设计、增加例外和中断的支持、添加总线接口、增加 TLB 和 Cache。不要害怕，任务确实不少而且越往后越难，所以我们的分值设定是前面的实验分值高后面的实验分值低，也就是说希望 GPA 拿高分的同学确实要多花点精力，这应该也是合理的。另外，在实验安排的顺序上，我并不严格按照“主课”（计算机体系结构）的讲授内容进行，主要还是尽可能地与组成原理的实验课有机地串联起来，这主要就体现在我上来先做简单五级流水 MIPS CPU 的实验。不过在我心目中的课程体系建设中，组成原理实验课就应该让所有的同学都完成简单的五级流水 MIPS CPU，这样我这门课就重点教会大家做一个能跑操作系统的 MIPS CPU，在系统的层面上达到融会贯通。由于你们这一届数字逻辑课程的实验完成度不太够，从而导致了组成原理实验进度受影响，进而影响到我这门课。作为系统结构方向硬件 3 门课的收官课程，我们只得忍痛放弃了后面的部分进阶实验，尽量让更多的同学完成基础的实验训练。

在授课形式上，采用“课堂讲授+现场检查+在线答疑”相结合的方式。除了第一次课程只有课堂讲授，最后一次课程只有现场检查外，几乎每一周的课程都会包含前一周任务的现场检查和后续实验相关内容的课堂讲授。是的，我们每周都会有实验任务要检查，这也是为了让尽可能多的同学能够跟得上。在时间分配上，现场检查占的时间会多一些，这也是希望能多一些和同学们面对面交流、手把手传授的时间。课堂讲授的时候主要是强调重点，细节的东西都会写到讲义当中供同学查阅，同时我们还会把现场检查时发现的普遍问题进行集中的讲解答疑。在线答疑主要通过 piazza 平台开展，这个平台除了由于服务器不在境内而导致访问有点卡这个缺点之外，几乎是个理想的课程在线答疑平台。你们可以在此平台上提问，除了我们授课团队的老师和助教进行解答外，同学们也可以尝试着回答其他同学的提问，如果回答正确将被教师和助教确认，被多次确认回答的学生期末将获得额外的分数奖励。所有提问按照实验归类，大家都能看得到，比起发邮件问问题效果要好的多，因为你的问题很有可能别人已经问过了并且被解答了，就不用再发邮件问了。我们一周只有一次课 2 个课时的课堂时间，面对面交流的机会太少，所以希望大家多用用好 piazza 这个平台。

在课堂讲授方面，开始的实验我可能会教的多一些，希望把一些我们在工程实践中形成的经验以一种近似“手把手”的方式交给大家，让大家快速地达到一个统一的基础水平；后面的实验我可能会略微放手

一些，譬如布置任务的时候我可能只有一些提示，而我会规定的提交时间之后发布一个较为详细的指导说明，实在无法按时完成的同学可以参考这个说明完成延期提交，但是分数肯定会被打折。所以，在学期开始阶段，请各位“学神”、“学霸”稍微忍耐一下，后面会有你们发挥的机会的。此外，实验任务我们有可能提前发布，也就是说不一定在前一个实验检查之后才发布，而提前提交并完成检查是有额外奖励的。所有额外奖励加上之后总分超过 100 分的按 100 分计。

这里特别说明一下我们对于所谓“跟不上”的同学的考虑。我不知道你们这一届有没有这样的同学，之前的两年中由于某些原因在学业上完成得不太顺利，对于能否继续学下去内心产生了动摇，我真诚地邀请你们在学期开始就与我们私下联系一下，我们会在充分调研你们的过往学习情况之后，给予更多一些关照和帮助。事在人为，只要让我们感受到你在努力，那么至少在这门课上，我相信能够让一些糟糕的事情不再发生。

1.4 课程实验安排

实验 0 实验环境入门

实验任务：

1. 同步 RAM、ROM 的生成。
2. 多周期 MIPS CPU 移植到新的平台中进行评测

实验目的：

1. 熟悉 FPGA 实验环境

时间安排：

第 1、2 周

实验 1 静态 5 级流水简单 MIPS CPU 实现

实验任务：

1. 将《计算机组成原理研讨课》上完成的多周期 MIPS CPU 改进为 5 级流水设计。
2. 将《计算机组成原理研讨课》上完成的 CPU 所实现的指令增加至 51 条。
3. 寄存器相关解决实现全阻塞和激进前递两个版本，比较两者性能差异。

实验目的：

1. 理解流水线数据通路和控制逻辑的基本设计原理。
2. 理解 CPU 流水线设计中将指令信息逐级传递这一设计关键。
3. 理解流水线 CPU 中各类相关的产生机理和解决方式。
4. 理解流水线 CPU 中各类相关的产生机理和解决方式

时间安排：

第 3、4、6、7 周

实验 2 软件编程电子表

实验任务：

1. 在龙芯 SoC 系统上编写软件，通过驱动 GPIO 的方式实现一个含有时、分、秒、百分秒的电子表。
2. 电子表具有复位、暂停、启动、时分秒设置的功能。
3. 按键控制逻辑与 CPU 之间采用中断通信机制，并完成相应的中断处理程序设计。

实验目的：

1. 理解 memory mapped I/O 的访问机制。
2. 知晓控制程序执行延迟的原理。
3. 深入理解 MIPS CPU 处理外部中断这一过程中软硬件交互的过程。
4. 掌握 MIPS 汇编编程。

时间安排：

第 8 周

实验 3 CPU 增加例外和中断的支持

实验任务：

1. 在实验 1 所实现 CPU 基础之上实现对于例外和中断的支持。
2. 例外包括保留指令、溢出、断点、系统调用、地址错例外。
3. 中断包括软件中断、硬件中断（含时钟中断）。

实验目的：

1. 掌握流水线 CPU 中例外处理的设计思路。
2. 熟悉 MIPS 架构下例外处理的软硬件交互界面和交互流程。

时间安排：

第 9、10 周

实验 4 CPU 增加总线接口

实验任务：

1. 在实验 4 所实现 CPU 基础之上实现 AHB-Lite 总线接口。

实验目的：

1. 掌握片上总线的一般性原理。
2. 掌握总线接口与 CPU 内部流水线之间的相互关系。

时间安排：

第 11、12 周

实验 5 CPU 增加 TLB MMU

实验任务：

1. 将实验 5 所实现 CPU 的存储管理机制由直接地址映射改为 TLB 映射方式。
2. 软件可见 TLB 共 16 项，支持 4KB 页大小。
3. 实现指令 TLB Cache 有额外奖励。

实验目的：

1. 深入理解基于 TLB 的存储管理机制，掌握其中软硬件交互的界面和交互流程。

时间安排：

第 13、14、15 周

实验 6 用 CPU 搭建 SoC 跑小型操作系统

实验任务：

1. 完善自己设计的 CPU，用其替换实验 2 中 SoC 系统中的龙芯 CPU，在 FPGA 上正常启动操作系统内

核。

2. 在采用自己 CPU 核的 SoC 上运行 Dhrystone、Coremark 等性能测试，各组进行性能比赛。

实验目的：

1. 在真实系统环境下理解 CPU 各设计要素的互动关系。
2. 初步培养性能评估、优化的能力。

时间安排：

第 16、17、18、19 周

1.5 其它课程注意事项

国科大B62009H计算机体系结构研讨课17-18秋季

国科大B62009H计算机体系结构研讨课17-18秋季

-
1. 课程讲义、实验任务书、实验环境，都通过 sep.ucas.ac.cn 平台发布。
 2. 同学们的实验设计和实验报告，都通过 sep.ucas.ac.cn 平台提交。
 3. 每个实验任务提交的具体要求会在实验任务书详细指定，不做统一指定。
 4. 每次实验的提交情况均会及时反馈，但是每次实验的成绩在期末之前不会发布。
 5. 在 piazza 平台正式启用后，我们不保证邮件、微信、QQ 这些渠道上问题答复的及时性。
 6. 不要把我们的讲义上传到各类共享讲义平台上。这些讲义大多初次成稿，不足之处很多，我们经过多轮磨合后，在合适的时机主动开放出去，在此之前，“家丑不可外扬”。
 7. 除非在实验报告中明确写明所承担任务比例，否则，以组为单位检查的实验，组内各成员打分一致。但是需要说明的是，只有双方签字的纸质版实验报告才能作为调整评分的依据，电子版中写到的一律无效。
 8. 希望你们有能力处理好一个小组内部的分工合作和人际关系，但是如果中途实在觉得无法维系下去，可以找我们调整分组情况，但是，调整之后每组人数仍然不得超过 2 人，且调整之后自行解决实验箱问题。