**2024计算机体系结构**

**Quiz 5**

**姓名： 常添**

**学号：** 2022111699

**课程的主要内容：第1章 计算机系统量化分析基础、第2章 指令系统、第3章 基本流水线、第4章 指令级并行、第5章 存储层次、第6章 IO系统**

1. **你认为在本门课程的内容中，哪一部分学习起来最困难？**

指令集并行的记分牌，Tomasulo算法，硬件前瞻，只能大致理解原理，具体的实现还是很难掌握。

减小cache命中时间的方法，我认为可以不讲，简单讲几个思路就行。

**2. 在本门课程中，如果增加一部分内容的话，你希望增加一些什么知识？**

增加Intel x86和amd64架构的讲解，并与MIPS比较，这样可以与自己的机器相结合，还可以对比其他专业课所学的内容。

**3. 在本门课程中，如果减少一部分内容的话，你认为那些内容可以简化？**

改进cache的方法不必讲那么多种，主要让同学们理解思想即可。

**4. 针对课程内容的设置（课堂、作业、Quiz、报告、实验），有什么改进意见？**

**实验评价：**实验内容难度较大，我自己并不是很擅长硬件，很多时候我花费了一整个周末的时间都无法完成，而且仿真和上板之间存在一定差异，实际上浪费了很多时间在编译上。我知道实验是由龙芯竞赛的学长设计的，**但这也导致一些不擅长硬件编写和逻辑的同学会遇到很大的困难。**

我认为实验可以效仿操作系统实验，操作系统课程实验并没有让我们从0开始编写操作系统，而是让我们根据环境和具体的例子理解原理和测试系统。计算机体系结构如果要求每个人都从0开始设计CPU的话其实并不是很合适，毕竟在未来只有极少的一部分人会在就业或者科研方面选择这一个领域，更重要的是理解其工作原理，所以**我建议把实现型实验改成验证性或者测试性实验，希望得到采纳。**

我目前没法提出具体的改进方案，但是如果实验的内容能**结合同学们的具体设备**（Intel/AMD CPU，Nvidia GPU等）那肯定会极大增加同学们的兴趣。例如，既然学习了量化分析的方法，**让我们编写一个具体的测试程序从不同角度定量评价咱们课程所有同学的CPU性能，这样的开放型实验或许会十分有趣，**也用到了课堂上的理论知识，同学们也会非常好奇并增加对课程的兴趣。

**课程评价：**

首先这个课程在互联网上的公开资料很少，如果自己对课堂内容难以理解的话只能反复读书和ppt，不过这个课程的深度摆在这里，这一点无可厚非，但是缺少资源总会让人觉得在学习时没有着落。**课程的内容如果自己不能动手验证而只是学习理论，最后就是仅仅背书。**因此，我强烈建议找到Intel/AMD的官方文档，并结合我们同学的计算机使用经历来扩充课程内容，比如这些厂商的不同设备CPU有哪些主要区别，不同代的产品主要的提升是什么，类似这样。**计算机内容的学习需要结合现实中直观的现象，否则就会略显枯燥。**

如果真的想让同学们提升对处理器架构的兴趣，**可以收集这些厂商的发布会和官方手册，结合课程内容，增加一些小组讨论的成分。**这些应该让我们在课堂中讨论和展示，而不是写到报告里作为分数的一部分。具体来说，可以分析Intel最新的i9处理器的大小核有什么特点，苹果的M系列芯片如何兼顾性能与功耗的，华为的鲲鹏架构与国外的主流架构有什么不同，这些内容让我们自己挖掘，然后讨论，最后老师可以给出学术解释或者点评，或许是这门课的一个有意义调整方向。

**5. 关于这门课，对你下一届的同学有哪些建议？**

**· 这门课程实验很难，要很大耐心才能做好。**（基于目前的状态）

* **这门课程学术性和理论性很强，学起来会很抽象。对于擅长写软件程序的大多数同学来说是一大挑战。**
* **这门课程的参考资料不多，**一定要多跟同学和老师交流。
* **这门课程的很多内容无法像计算机网络（自己可以完全配置网络）、操作系统（linux内核有完全注释，也有相关调试软件）那样在自己的设备上验证**，所以你可能会对课程内容有质疑，即我看不见讲的这些内容是如何在自己的机器上工作的，那它到底对不对，它还在用吗？自己也无法直接运行验证本课程的理论算法。但事实就是这样，这个课程很多内容并不是开源的，官方统一标准和文档很少，这一点和常规的计算机学科很不一样，我认为这是最难过的一关。（即使课程组大幅调整课程内容也无济于事，因为那是芯片公司的商业机密，如果OS界没有Linux，也是同样的道理）。
* **这门课程既不是纯编程实现CPU，又不是纯理论课，其实定位蛮奇怪的。**纯编程课，不用管那么多，就是实现一个个模块的实现就好了，就看需求是不是都实现了；可是Verilog太难写了，比C++和Java要求多得多，经常编译几十分钟，网上还不好找文档参考，而且上板又是一大难题，搞不懂原理只会写程序最后都会卡在上板。纯理论课，像算法，数理逻辑，形式语言，讲原理，公式，推导，一步步逻辑十分清晰且严格；但是这个课并不能完全“形式化”，它有实践操作探索的内容在，就不能完全的用数学的逻辑来看待。

**以上五点请自行体会。**

**6. 通过本门课程的学习你是否有收获？如果本门课程改为选修课，你是否会选？**

· 体系结构是来连接软硬件的重要桥梁，本课程主要让我理解了计算机系统中指令集是如何工作并一步步优化的，补充了我对计算机的认识，当然，我也认识到了搞架构并实现的都是大佬，软硬通吃。

* 根据第五部分的结论，如果作为选修课，即使没有实验，我也大概率不会选择该课程，没有任何的其他意思，只是不适合我。喜欢玩理论的信安、网安专业，和喜欢玩实践的大数据、AI、软件工程专业，如果没有深入学习CPU的打算，大概率也不会选。计算机组成原理对于他们来说都非常厌烦，认为毫无意义，奈何它是大学分的考研课，不得不学。比计算机组成原理更深，更难，更抽象的体系结构课自然不会被他们看上，有那些时间早学深度学习、数据挖掘和应用开发了。
* 我很理解各位芯片报国的一腔热血，但是上过了体系结构课才知道实现起来太难了，我这种没有天赋的菜鸟还是老老实实卷CV、NLP、DS吧，system或者arch方向真的不适合大多数人。

**声明：本文档所有内容仅代表我对本课程的意见或建议，与课程组成员没有任何关系。**