

研究生精品课程简介

课程名称：高级计算机体系结构

课程代码：0700037

选课人数：30

开课学院：计算机学院

授课教师：计卫星

| 育人要点 | 成效简介 |
|------|---|
| 教师风范 | <p>副教授，信息技术新工科产学研联盟副秘书长，计算机学会体系结构专业委员会委员。2008年毕业于北京理工大学计算机学院获工学博士学位，同年进入北京理工大学计算学院工作。2012年3月至2013年3月赴美国罗切斯特大学计算机系访问1年。目前主要研究方向包括程序语言设计与实现、大规模代码分析、并行与高性能计算等，主持和参与国家自然科学基金项目多项，在国内外著名期刊和国际会议（TACO、PLDI、RTSS和DATE等）上发表论文50余篇；获省部级教学成果奖1项，获得国家级教学成果奖1项，出版教材3部。</p> |
| 价值塑造 | <p>弘扬自力更生、艰苦奋斗的延安精神：从体系结构的发展历史，国产处理器发展历程和现状，以及与其配套的系统软件的研究现状和未来，说明自力更生，艰苦奋斗，实现国产自主研发的重要性。</p> <p>培育高水平跨学科创新人才，促进学科交叉问题的研究：系统是任何计算机的核心，也是整个应用的坚实基础。应用的发展牵引着系统的研发动向。系统安全、软件加速等都涉及到从上层应用到系统底层的问题。本课程以此为切入点，深入探讨体系结构与其他学科方向之间的关联和相关的研究问题，引导学生从不同的层面和视角对问题进行深入的研究，鼓励在学科方向交叉领域形成原始创新。</p> |
| 知识教育 | <p>本课程包括计算机体系结构的重要基础理论、最新研究和技术进展和典型应用实践三部分构成：</p> <p>重要基础理论主要包括：典型体系架构、存储模型、Cache一致协议、指令乱序执行模型、计算机系统性能度量方法等几个方面，通过这些内容讲解，让学生了解经典理论和模型，把握住核心理论和技术。</p> <p>最新研究和技术进展主要包括：非易失性存储技术、硬件加速器技术、虚拟化技术、云计算技术等，通过这些内容的讲解，让学生了解与体系结构相关的技术发展趋势，把握技术的发展趋势。</p> <p>典型应用实践主要包括：GPU并行编程实践、大规模数值求解的超级计算机编程实践、智能应用的硬件加速器编程实践、软件分析的安全实践等。通过这些典型应用实践让学生加深对体系结构的理解，学会在学科专业的研究和学习中灵活运用体系结构相关内容。</p> <p>科教融合：结合正在执行的课题基因匹配硬件加速器设计介绍硬件加速技术，结合正在参与执行的重点研发计划介绍超级计算机上的软件优化技术。</p> <p>产教融合：邀请忆芯科技首席架构师薛立成介绍存储芯片设计方法和最新进展，采用寒武纪和华为的最新加速卡板进行智能应用加速实践。</p> |



实践能力
(创新性、
批判性、
颠覆性
思维培养)

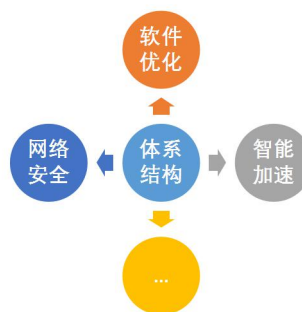
课程旨在引导学生结合自己的研究方向和体系结构从事跨学科专业交叉研究，培养“体系结构+”的研究和工作思维 and 意识，主要体现在以下三个方面：

体系结构+网络安全：以系统安全为切入点，介绍系统层次与体系结构相关的安全问题以及相关的防范方法。例如 Intel 处理器的猜测执行漏洞发生的原因以及最新的防范方案。

体系结构+智能计算：以智能加速器设计为切入点，介绍硬件加速器的设计与使用方法。例如寒武纪设计的神经网络加速器的基本架构和加速原理，及其编程实践。

体系结构+软件优化：以软件并行优化为切入点，介绍 GPU 的基本架构、编程模型和常见算法的优化方法。例如 GPU 上的稀疏矩阵向量乘、稀疏矩阵和稀疏矩阵乘的并行优化。

课程结束时，要求选课学生结合自己的研究方向，选择上述三个方向中一个给出相应的方案和初步实现，并根据问题的典型性、方案的可行性和实现效果进行综合评定。



此外，本课程重点训练学生的思辨能力，就目前与体系结构相关的话题进行调研并讨论，可选的内容包括：

- (1) 中国究竟要不要设计自主指令集架构处理器？
- (2) RISC-V 架构能否让国产 IC 真正“自主研发”？
- (3) 未来处理器架构该如何发展？

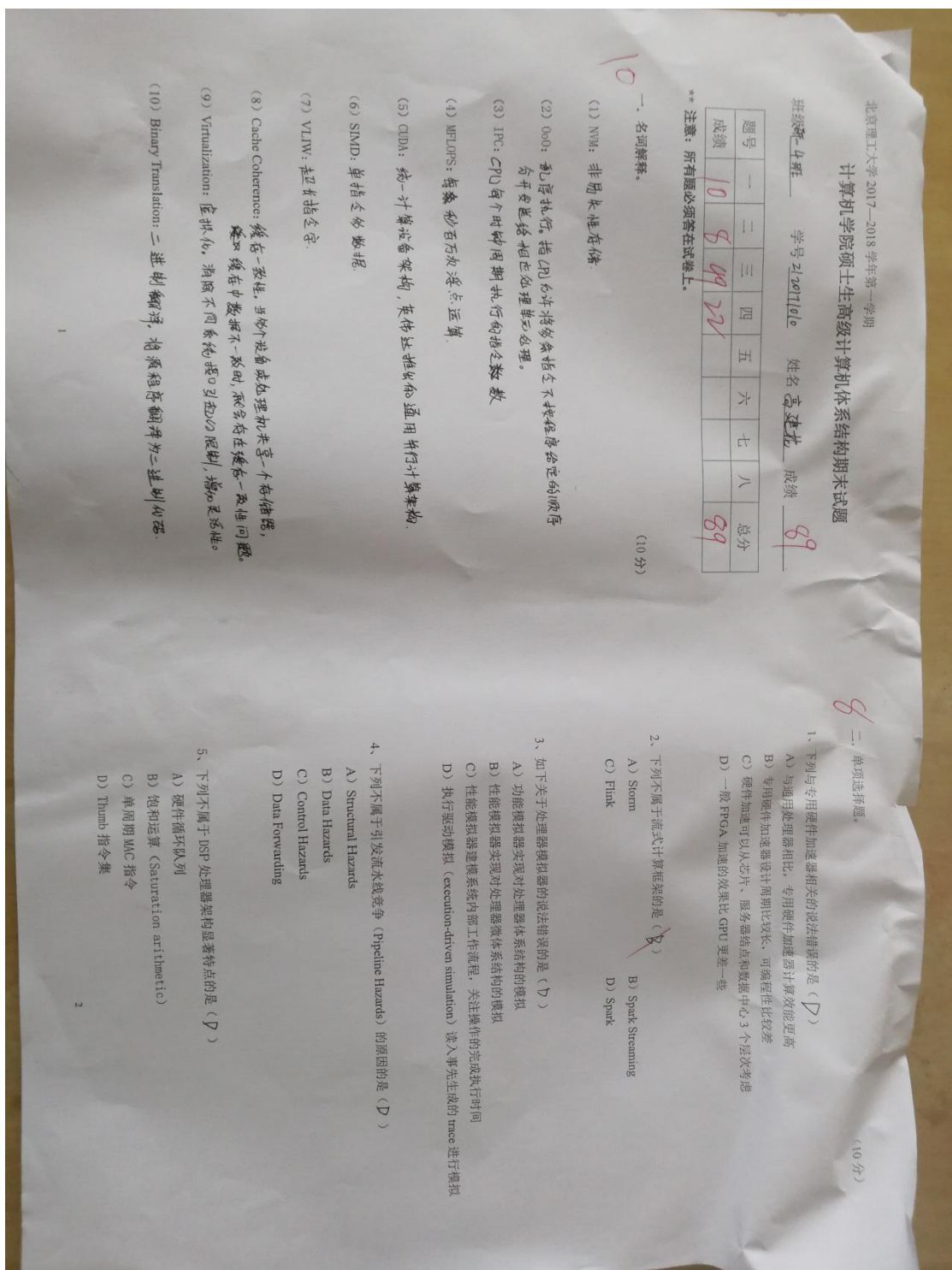
此部分教师需要首先进行问题背景进行阐述，并对学生进行分组调研和辩论，并由学生和教师为每个学生的表现打分。

| | |
|------|---|
| 课程考核 | 成绩评定依据:平时作业和实践成绩占 30%，期末笔试成绩占 70%。 研究生答卷见附件 1. |
| 学院意见 | 学院领导： _____ 年 月 日 |

识别下方二维码可参与课程的互动评价：



对研究生课程建设任何意见建议，请联系研究生院培养办公室：mayc@bit.edu.cn



北京理工大学 2017-2018 学年第一学期
计算机学院硕士生高级计算机系统结构期末考试题

班级 生班 学号 2120110101 姓名 高建花 成绩 89

| | | | | | | | | | |
|----|----|---|----|----|---|---|---|---|----|
| 题号 | 一 | 二 | 三 | 四 | 五 | 六 | 七 | 八 | 总分 |
| 成绩 | 10 | 8 | 11 | 22 | | | | | 89 |

** 注意：所有题必须答在试卷上。

一、名词解释。(10分)

- (1) NVM: 非易失性存储
- (2) OoO: 乱序执行, 指 CPU 允许将多条指令不按程序给定的顺序
分开送经相对处理单元处理。
- (3) IPC: CPU 每个时钟周期执行的指令数
- (4) MFLOPS: 每秒百万次浮点运算
- (5) CUDA: 统一计算设备架构, 使传统软件和通用并行计算架构。
- (6) SIMD: 单指令多数据
- (7) VLIW: 超长指令字
- (8) Cache Coherence: 缓存一致性, 当多个处理器或处理机共享一个存储器,
缓存中数据不一致时, 系统会在缓存一致性问题上。
- (9) Virtualization: 虚拟化, 消除不同系统接口之间的限制, 增加灵活性。
- (10) Binary Translation: 二进制翻译, 将源程序翻译为二进制代码。

二、单项选择题。(10分)

1. 下列与专用硬件加速器相关的说法错误的是 (D)
- A) 与通用处理器相比, 专用硬件加速器计算效能更高
 - B) 专用硬件加速器设计周期比较长, 可编程性比较差
 - C) 硬件加速可以从芯片、服务器结点和数据中心 3 个层次考虑
 - D) 一般 FPGA 加速的效果比 GPU 更差一些
2. 下列不属于流式计算框架的是 (B)
- A) Storm
 - B) Spark Streaming
 - C) Flink
 - D) Spark
3. 如下关于处理器模拟器的说法错误的是 (D)
- A) 功能模拟器实现对处理器体系结构的模拟
 - B) 性能模拟器实现对处理器体系结构的模拟
 - C) 性能模拟器建模系统内部工作流程, 关注操作的完成执行时间
 - D) 执行驱动模拟 (execution-driven simulation) 读入事先生成的 trace 进行模拟
4. 下列不属于引发流水线竞争 (Pipeline Hazards) 的原因是 (D)
- A) Structural Hazards
 - B) Data Hazards
 - C) Control Hazards
 - D) Data Forwarding
5. 下列不属于 DSP 处理器架构显著特点的是 (D)
- A) 硬件循环队列
 - B) 饱和运算 (Saturation arithmetic)
 - C) 单周期 MAC 指令
 - D) Thumb 指令集

三、简答题

1. 假设变量 A 和 B 的初始值均为 0，当表格中的代码在三个处理器上执行时，寄存器 R1 的值可能会是什么，并解释为什么。

| | P1 | P2 | P3 |
|------|-----------|-----------|------|
| A=1 | | | |
| B=1 | $!(A==1)$ | B=1 | |
| R1=A | | $!(B==1)$ | R1=A |

答：① R1=1
 P1 执行 B=1，然后 P2 执行 B=1，得到 B=1。
 P2 执行 B=1，然后 P3 执行 B=1，得到 B=1。
 P3 执行 B=1，然后 P1 执行 B=1，得到 B=1。
 最终 B=1，R1=A。

② R1=0, P1, P2, P3 有写操作，写 P1 的 code 中 A 的值为 1，并稍后如 B 变量和 P2 的 B 为 2 执行 B=1，并稍后 B 的值为 2，P1 和 P3 此时，有写操作 P3 还未读到 P1 的 A 更新，但读到 P2 中 B 的更新，即 A=0, B=1，所以 R1=0。

2. 如下表给出了多种存储介质的参数对比，请简要说明 PRAM（又称为 PCM）的工作原理，并从系统设计的角度来看如何在新的计算机系统使用 PRAM?

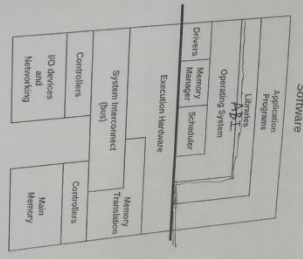
表 1 多种存储介质的参数对比表

| Metrics | Emerging Products | | | | Proprietary | | | | |
|-------------------------|-------------------|------------------|-----------------|------------------|-------------------|-----------------|-----------------|-----------------|-------------------|
| | SRAM | DRAM | Fused/NVR | FEMRAM/NSD | FRAM | MRAM | PRAM | PRAM | STR-DRAM |
| Non-Volatile | No | No | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
| CPI Speed | 50-120 | 6-10 | 10 | 5 | 15-34 | 18-10 | 8-12 | 4-10 | 4-20 |
| Read Throughput | 1-100 | 30 | 10 | 30 | 20-80 | 3-20 | 20-24 | 10-20 | 2-20 |
| Write/Read Throughput | 1-100 | 15 | 10 | 10 | 100/100 | 60-120 | 10-20 | 10-20 | 2-20 |
| Endurance | 10 ¹⁰ | 10 ¹⁴ | 10 ⁵ | 10 ¹² | >10 ¹⁵ | 10 ⁸ | 10 ⁸ | 10 ⁸ | >10 ¹⁵ |
| Write Power | Low | Low | Ver High | Low | High | High | Low | Low | >10 ¹⁵ |
| Other Power | Current | Refresh | Current | None | None | None | None | None | None |
| Consumption | Low | Low | Ver High | Low | High | High | Low | Low | >10 ¹⁵ |
| High Voltage Resiliency | No | 3 | 4-8 | 18-20 | 2-3 | 3 | 1.5-3 | 1.5-3 | <1.5 |

答：PRAM 的工作原理：利用电致阻变在不同温度下所表现出的不同状态的特性，通过电致阻变控制 PRAM 介质的单元电阻值来改变单元中介电和电阻值，利用不同的尺寸特性来电阻值来改变逻辑值。
 (2) 从表中可以看出，与目前较为常用的 DRAM 相比，PRAM 具有非易失性、低功耗、高读写速度等优点，但其性能和功耗、耐久性等还有待提高。从技术上看，PRAM 可以作为一种存储 cache 的补充，但其功耗、耐久性等还有待提高。另外 PRAM 也可以作为一种存储 cache 的补充，但其功耗、耐久性等还有待提高。

3. 请简要说明 Intel Itanium 2、AMD K8 和 VIA C7 三种指令集并行技术各自的特点。

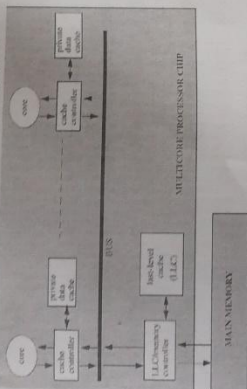
Intel Itanium 2: 指令级并行 (ILP) 和线程级并行 (TLP) 相结合，通过多线程技术提高指令级并行性。
 AMD K8: 多线程并行 (MTP)，通过多线程技术提高指令级并行性。
 VIA C7: 多线程并行 (MTP)，通过多线程技术提高指令级并行性。
 此外，还有超线程 (HT) 技术，通过超线程技术提高指令级并行性。
 指令级并行 (ILP) 是指通过提高指令级并行性来提高处理器性能。超线程 (HT) 是指通过超线程技术提高指令级并行性。多线程并行 (MTP) 是指通过多线程技术提高指令级并行性。
 并行技术的主要实现技术包括：多线程并行、超线程、指令级并行、线程级并行等。



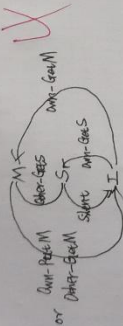
答：超线程技术加在 API (Application Programming Interface, system calls) 层实现。

PRAM 虚拟机的主要实现技术：
 二进制的翻译
 寄存器映射
 指令重排
 控制流优化
 寄存器优化
 指令级并行

6、如下所示为一多核处理器体系结构，假设该处理器采用基于总线监听 Cache 一致性协议，请给出 Cache 控制器（图中 Cache controller）的 MSI 协议状态转换图，并说明如何对该协议进行改进以提高 Cache 系统的效率？



答：Cache 控制器的 MSI 协议状态转换图：



改进提高 Cache 系统的效率

- ① 增加 E (Exclusive) 状态，当 Cache 得到句数据自由持有状态，此时 Cache 即为 E 状态，增加了 write - Cache 数量，提高了 Cache 的效率，降低了延迟，减少了总线通信阻塞。
- ② 增加 O (Owned) 状态，~~...~~

7、请从存储顺序一致性 (Memory Consistency Model) 的角度说明，如下 Java 代码并运行在多个处理器内核上执行时存在什么样的问题，以及如何修复该问题。

```

class Foo {
    private Helper helper = null;
    public Helper getHelper() {
        if (helper == null)
            synchronized (this) {
                if (helper == null)
                    helper = new Helper();
            }
        return helper;
    }
}
    
```

答：调用 `getHelper() = new Helper()`；使用 `new` 初始化对象时，总共会有两个步骤，一是开辟空间，此时 `helper` 已非空，二是调用构造函数初始化。两个步骤是先后顺序，所以会有一些时间和间隔。如果两个处理器在同时调用 `getHelper()`，此时 `helper` 非空，但未完全初始化，导致返回的 `helper` 并非完整数据。
 修复该问题：可在 `getHelper()` 方法中定义局部变量，通过 `new Helper()` 来初始化局部变量，然后将其赋值给 `helper`，从而避免了上述问题。

四. 综合应用.

1. 请列举自己研究领域的一个计算密集型的问题算法, 说明使用 CPU, GPU 和硬件加速器哪种平台实现更合理一些, 并解释为什么.
答: 神经网络训练算法: 通过计算神经网络上的数据梯度, 来得到梯度的形式, 也就是神经网络的信息.

(24分)

计算法的核心部分是互相作用的数据集, 数据和参数. 在很大计算量, 以及大规模作用域, 因为其与层之间有数据依赖, 不能为重叠并行, 而且不同层计算量不同, 负载不均衡. 对于上述算法, 可以采用 GPU 并行计算. 通过设计合适的控制调度算法, 来将任务分配到计算量不同的设备 CPU 和 GPU, 大计算量任务在 GPU 上, 小计算量以及核心控制逻辑在 CPU 上运行, 充分利用控制逻辑.

2. 如下为一指令序列, 请分析这些指令之间的 RAW, WAR 和 RAR 相关性, 并以此为例说明如何通过寄存器重命名实现多条指令并行执行.

| | |
|---------|------------|
| (1) add | r3, r2, r3 |
| (2) sub | r2, r1, r3 |
| (3) mul | r1, r3, r1 |
| (4) add | r2, r3, r1 |
| (5) add | r2, r1, r3 |

答: RAW (Write-After-Write): (2) → (4), (4) → (5)

WAR (Write-After-Read): (1) → (2), (2) → (3), (1) → (4), (1) → (5), (3) → (4), (3) → (5)

RAW/Read-After-Write: (1) → (2), (1) → (3), (1) → (4), (1) → (5)

- 寄存器重命名
- (1) add R₀, R₂, R₃
 - (2) sub R₂, R₁, R₄
 - (3) mul R₀, R₁, R₁
 - (4) add R₂, R₄, R₀
 - (5) add R₀, R₀, R₄

| | | | |
|-----|----------------|----------------|----------------|
| | R ₁ | R ₂ | R ₃ |
| (1) | R ₁ | R ₂ | R ₃ |
| (2) | R ₁ | R ₂ | R ₄ |
| (3) | R ₀ | R ₁ | R ₁ |
| (4) | R ₂ | R ₄ | R ₀ |
| (5) | R ₀ | R ₀ | R ₄ |

指令指令并行执行

