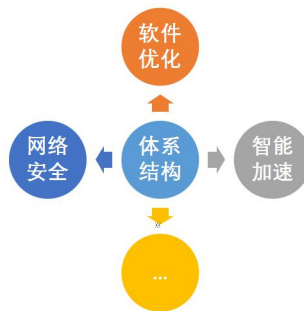


## 研究生精品课程简介

|  |  |
|--|--|
|  |  |
|  |  |
|  |  |
|  |  |



|  |  |
|--|--|
|  |  |
|  |  |



班级 生班 学号 2120111016 姓名 高建北 成绩 89

|    |    |   |    |    |   |   |   |   |    |
|----|----|---|----|----|---|---|---|---|----|
| 题号 | 一  | 二 | 三  | 四  | 五 | 六 | 七 | 八 | 总分 |
| 成绩 | 10 | 8 | 09 | 22 |   |   |   |   | 89 |

\*\* 注意：所有题必须答在试卷上。

一、名词解释。

- (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顺序发出送往相应处理器处理。
- (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 GPU 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单元在缓存一致性问题上, 缓存中数据不一致时, 必须有什么限制, 增加互斥性。
- (9) Virtualization: 虚拟化, 消除不同系统接口引致的问题。
- (10) Binary Translation: 二进制翻译, 将源程序翻译为二进制代码。

(10分)

二、单项选择题

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, 可维护性比较差
  - C) 硬件加速可以从芯片、服务器节点和数据中心 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. 下列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与内部工作流程, 关注操作的完成执行时间
  - D) 执行驱动模拟 (execution-driven simulation) 嵌入事先定义的 trace
4. 下列不属于引发流水线下争 (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 处理器的显著特点的是 (P)
  - 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





(24分)

四、综合应用。  
1. 请列举自己研究领域的一个计算密集型问题的算法，说明使用 CPU、GPU 和硬件加速器哪种平台实现更合理一些。  
答：神经网络训练算法；通过计算密集型在 GPU 上的数据并行，来获得硬件加速。

12  
计算法的核心部分是互相作用即耦合，神经网络三个步骤，在很大计算量，以近似的方式用填充，因为其层与层之间有数据依赖，不能为并行而不同层计算量不同，片断不均匀。  
对于上述算法，我认为使用 GPU 并行更合理一些，通过设计合适的柱状稠度算法来指定不同计算量以任务分发选路 CPU 和 GPU，大计算量任务在众核 GPU 上，小计算量以低核心相度数在 CPU 上运行，充分利用控制逻辑。

10

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

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。

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 R<sub>0</sub>, R<sub>2</sub>, R<sub>2</sub>
  - (2) sub R<sub>2</sub>, R<sub>2</sub>, R<sub>4</sub>
  - (3) mult R<sub>6</sub>, R<sub>4</sub>, R<sub>1</sub>
  - (4) add R<sub>7</sub>, R<sub>4</sub>, R<sub>6</sub>
  - (5) add R<sub>8</sub>, R<sub>6</sub>, R<sub>4</sub>

|     |                |                |                |
|-----|----------------|----------------|----------------|
| X   | Y <sub>1</sub> | Y <sub>2</sub> | Y <sub>3</sub> |
| (1) | R <sub>1</sub> | R <sub>2</sub> | R <sub>3</sub> |
| (2) | R <sub>1</sub> | R <sub>2</sub> | R <sub>4</sub> |
| (3) | R <sub>1</sub> | R <sub>2</sub> | R <sub>4</sub> |
| (4) | R <sub>6</sub> | R <sub>7</sub> | R <sub>4</sub> |
| (5) | R <sub>6</sub> | R <sub>7</sub> | R <sub>4</sub> |

指令重命名并行执行

