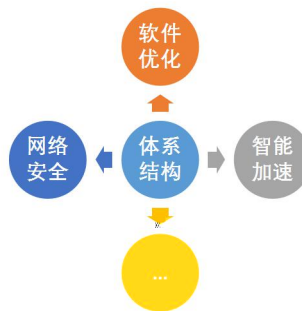


研究生精品课程简介





班级 牛班 学号 20111010 姓名 高越花 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。

(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
分开发送给相对处理器处理。
- (3) IPC: CPU 每个时钟周期执行的指令数
- (4) MEMOPS: 每秒百万次浮点运算
- (5) CUDA: 统一计算设备架构, 使这些异构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器共享一个存储器,
缓存中数据不一致时, 必须存在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

- 1、下列与专用硬件加速器相比较, 通用处理器
 - A) 与通用处理器相比, 寄存器
 - B) 专用硬件加速器保持数据
 - C) 硬件加速器可以以芯片、片
 - D) 一般 FPGA 加速的效果
- 2、下列不属于流式计算机
 - A) Storm
 - C) Flunk
- 3、下列关于处理器
 - A) 功能微处理器
 - B) 性能微处理器
 - C) 性能微处理器
 - D) 执行驱动微处理器
- 4、下列不属于引擎
 - A) Structural Hierarchy
 - B) Data Hierarchy
 - C) Control Hierarchy
 - D) Data Flow Hierarchy
- 5、下列不属于 ISD
 - A) 硬件循环
 - B) 饱和运算
 - C) 单周期 MIP
 - D) Thumbs 指令

三、简答题。

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

P1	P2	P3
A=1	B=1	R1=A
B=1	R1=B	R1=B

答：① R1=1，P1 执行后 A=1，P2 执行后 B=1，P3 执行后 R1=A=1。

② R1=0，P1, P2, P3 有各自的 cache，P1 有 cache，P2 有 cache，P3 有 cache。P1 执行后 A=1，P2 执行后 B=1，P3 执行后 R1=B=1。P1 和 P2 的 cache 未更新，所以 R1=0。

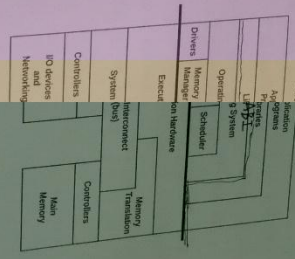
2. 如下表给出了多种存储介质的参数对比，并从系统设计的角度来看如何在新的计算机系统中使用 PRAM?

Medium	SRAM	DRAM	Flash/NonV	FeRAM	MRAM	PRAM	RRAM	STT-RRAM
Non-Volatile	No	No	Yes	Yes	Yes	Yes	Yes	Yes
Cell Size/ μm^2	20-120	6-10	10	5	15-24	18-10	8-12	4-10
Read Throughput	1-100	30	10	30	20-30	3-20	20-30	10-50
Write/Read Throughput	1-100	15	10	10	20-30	3-20	20-30	10-50
Endurance	10 ¹⁶	10 ¹⁴	10 ⁵	10 ¹²	10 ¹²	10 ¹²	10 ¹²	10 ¹²
Write Power	Low	Low	Low	Low	Low	Low	Low	Low
Other Power	Current	Current	Current	Current	Current	Current	Current	Current
Consumption	Low	Low	Low	Low	Low	Low	Low	Low
High Voltage Read/Write	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes

PRAM 的工作原理：利用电势差在特定电压下，通过电势差控制 PRAM 的电阻值，利用不同的电阻特性从上面可以看出，与目前较为常用的 DRAM 相比，PRAM 具有非易失性、低功耗、高速度等优点。但其目前仍存在一些挑战，如写入速度较慢、寿命有限。从发展趋势上看，PRAM 有望成为高功率、低延迟、高可靠性的存储介质。

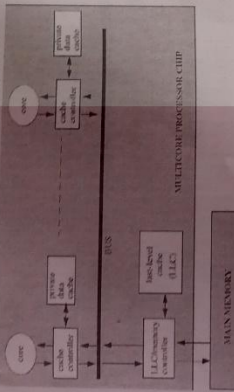
(58 分)

3. 请简要说明 ARMv8-A 的 Cache Coherence 问题。Cache Coherence 是指多处理器系统中，多个处理器共享同一组 Cache，且 Cache 中的数据不一致。Cache Coherence 问题是指多处理器系统中，多个处理器共享同一组 Cache，且 Cache 中的数据不一致。Cache Coherence 问题是指多处理器系统中，多个处理器共享同一组 Cache，且 Cache 中的数据不一致。



ARMv8-A 的 Cache Coherence 问题：在多处理器系统中，多个处理器共享同一组 Cache，且 Cache 中的数据不一致。Cache Coherence 问题是指多处理器系统中，多个处理器共享同一组 Cache，且 Cache 中的数据不一致。Cache Coherence 问题是指多处理器系统中，多个处理器共享同一组 Cache，且 Cache 中的数据不一致。

6、如下所示为多核处理器体系结构，假设该处理器采用基于总线互连的 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，每当 Cache 得到写数据是由互斥态，此时 Cache 即为 E 状态，增加了 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 (Owner) 状态，需要管理数据。

7、请从存储顺序一致性（Memory Consistency Model）的角度说明，如下 Java 代码并如何修复该问题。

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

答：使用双行 helper = new Helper(); 使用 new;

总共有两个步骤，一是开辟空间，此时初始化的对象，二是调用构造函数初始化。两个步骤并非顺序，所以会有一定时间的不同间隔。如果观察是先后顺序，和步骤二之间去读取 helper，此时处理器在步骤一初始化，读取得到的 helper 并非空，但未完全初始化，读取得到的 helper 并非空，但未完全初始化。

修复该问题，可在 getHelper() 方法初始变量，然后将其赋值给 helper，以此后部变量，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问题并平台实现更合理一些。
答: 神经网络训练任务。

计算密集型: 神经网络训练任务。
数据密集型: 图像处理、数据挖掘。

计算密集型: 神经网络训练任务。
数据密集型: 图像处理、数据挖掘。
对于上述算法, 我认为使用 GPU 比 CPU 更合理。因为 GPU 的并行计算能力更强, 且功耗相对较低。在 CPU 上运行, 不仅速度慢, 而且发热量大, 不利于大规模并行计算。

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) mult	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), (3) → (4), (1) → (5)
 RAR (Read-After-Read): (1) → (3), (1) → (5)

	r1	r2	r3
(1)	r1	r2	r3
(2)	r1	r2	r3
(3)	r1	r2	r3
(4)	r1	r2	r3
(5)	r1	r2	r3

