第01章　引言

重点部分：

1、 PPT 的翻页过程：软件部分+硬件部分

2、 冯诺依曼体系结构的本质特征：存储程序和指令驱动的执行

3、 性能、价格与功耗

（1）、性能

影响性能的因素：算法影响最大、编译器、指令系统、微结构、主频

（2）、成本

（3）、功耗

- 动态功耗+静态功耗：$p_{total} = P_{switch} + P_{short} + P_{leakage}$
- 低功耗优化层次：结构级、逻辑级、电路级和工艺级

4、 计算机体系结构演变：摩尔定律，登纳德定律

“墙”：存储墙、功耗墙、带宽墙、应用墙、成本墙

未来 CPU 结构：多核+向量处理、众核、带有协处理器的异构多核、多核+向量处理+专用处理器

5、体系结构设计的基本原则：平衡性（Amdahl 定律）、局部性、虚拟化、并行性

局部性种类：指令局部性、访存局部性、转移局部性

并行层次：指令级并行、数据级并行、任务级并行

复习提纲：

计算机体系结构的研究内容：

- 冯诺依曼结构存储程序和指令驱动执行是计算机体系结构的基础
- 计算机系统的四个层次及三个界面
  - 四个层次：应用程序、操作系统、硬件系统和晶体管
  - 三个界面：API、ISA、工艺模型

衡量计算机的指标：

- 衡量性能的指标：性能最本质的定义、IPC、影响 IPC 的因素
  - 性能的最本质定义
    - 完成一个任务（如后天的天气预报）所需的时间
    - 以指令为基本单位
  - IPC（Instructions per cycle）：乱序执行、多发射、存储层次

- 功耗的组成：静态功耗、动态功耗

计算机体系结构的发展趋势

- 摩尔定律、计算机应用、计算机体系结构三者关系
  - 半导体工艺技术和计算机体系结构技术互为动力
  - 应用需求是计算机体系结构发展的持久动力

摩尔定律是指集成电路上可以容纳的晶体管数目在大约每经过 18 个月到 24 个月便会增加一倍. 摩尔定律的提出, 推动了计算机应用和计算机体系结构的发展. 计算机应用是指计算机技术在各个领域的应用. 计算机体系结构是指计算机硬件和软件的结构. 摩尔定律的提出促进了计算机体系结构的发展，使得计算机应用得以广泛应用于各个领域.

计算机体系结构的设计原则

- 平衡性、局部性、并行性、虚拟化

第02章　指令系统

重点部分：

1、 指令系统的设计原则：兼容性、通用性、高效性和安全性

2、 影响因素：工艺技术、系统结构、操作系统、编译技术和程序设计语言和应用程序

3、 指令系统的分类(按指令长度)复杂指令系统、精简指令系统和超长指令字(VLIW)。

- RISC 技术有利于指令流水线的高效实现（X86 处理器内部也把 CISC 翻译成简单操作来优化流水线），VLIW 技术用于指令流水线优化不是很成功
- RISC 的精髓：简化指令间关系，有利于指令流水线高效实现

4、 存储结构的演变：连续实地址、段式、页式虚拟存储和段页式

5、 运行级别的演变：唯一实模式、保护模式、调试模式和客户模式

6、 指令系统的组成

a) 地址空间的组成：寄存器空间+内存空间

b) 指令系统的地址空间演变：

i. 堆栈型：零地址指令

ii. 累加器型：单地址指令

iii. 寄存器型：多地址指令

c) 操作数的表示

d) 寻址方式：Register, Immediate, Displacement, Register indirect

e) 指令操作：算术和逻辑运算指令、访存指令、转移指令和系统管理指令

f) 指令编码：定长、变长和混合

i. 简单操作和简单寻址方式用得最多

ii. 简单指令便于高效实现和使用

iii. 硬件优化应充分考虑兼容性

g) loongArch 自主指令系统

i. 3+3+3 能力：基础软件、动态翻译虚拟机、二进制翻译

ii. 特点：先进性、兼容性、模块化、扩展性

复习提纲：

指令系统简介

- 软硬件界面，结构设计者对应用的深入理解，指令操作和运行时环境

指令系统的设计原则

- 兼容性、通用性、高效性、安全性

指令系统的演变

- 指令集分类：CISC、RISC、VLIW
- 存储管理：连续实地址、段式、页式、段页式
- 运行级别：用户态、核心态

指令系统组成

- 地址空间：寄存器空间、存储空间；堆栈型、累加器型、寄存器型
- 操作数：操作数的存储（寻址方式、大小尾端），操作数的特征
- 指令操作和编码：运算指令、访存指令、转移指令、系统管理指令

RISC 指令系统比较：“亲兄弟”和“表兄弟”

C 语言的机器表示

- 如 C 语言的不同变量映射到地址空间的不同段

第03章　特权指令系统

重点部分：

1、 异常：使处理器从软件的正常执行流中脱离的事件，也叫**例外**

2、 异常的分类（依据来源）：
   a) 外部事件
   b) 指令执行中的错误：不存在的指令、浮点除以 0、地址不对齐、用户态无权访问
   c) 数据完整性问题
   d) 地址转换异常
   e) 系统调用和陷入
   f) 需要软件修正的运算
3、 操作系统与异常
   a) 用户态程序只能通过异常与操作系统交互
   b) 操作系统通过异常机制支撑应用程序运行
4、 异常处理的流程
   a) 异常处理准备
      i. 精确异常
   b) 确定异常来源
   c) 保存执行状态
   d) 执行异常处理
   e) 恢复执行状态
   f) 返回正常执行流
5、 中断
   a) 中断输入与中断优先级
   b) 向量化中断
   c) 传递中断事件
      i. 通过中断线传递
      ii. 通过中断消息传递（MSI）：一个设备申请多个中断，降低中断处理延迟、实现负载均衡
6、 存储管理
   a) 虚拟存储原理
      i. 多进程环境下统一的编程空间
      ii. 多进程环境下的共享与保护
      iii. 支持大于实际物理内存的编程空间
   b) LoongArch 处理器对虚拟存储的支持
      i. 虚拟地址空间是线性平整的
      ii. 内存物理空间范围表示为 0~$2^{plen}-1$
      iii. 两种虚实地址翻译模式：直接地址翻译模式和映射地址翻译模式
   c) 内存中的页表组织
      i. 三级页表，每项 8 个字节
      ii. 页表项内容
      iii. 页表通过直接映射方式访问
      iv. 每个进程的页表基地址存入进程上下文中
7、 异常返回
   a) 异常处理在核心态下进行
   b) 异常返回时使用 **ERTN** 指令，同时置为用户态

第04章　软硬件协同

重要部分：

1、 应用程序二进制 ABI:数据表示与对齐、寄存器使用、函数调用、栈布局、目标文件和可执行程序格式
   a) MIPS ABI
      i. O32
      ii. N64
      iii. N32
   b) LoongArch ABI
      i. LP32
      ii. LPX32
      iii. LP64
   c) X86 ABI
      i. i386
      ii. x86-64
      iii. x32
2、 LoongArch ABI 栈布局
   a) 栈
      i. 保存上下文
      ii. 传递参数
      iii. 保存临时变量，非静态局部变量
   b) 使用分类
      i. 简单叶子函数
      ii. 编译时可确定栈大小的：只有 sp
      iii. 运行时改变栈大小:fp+sp
   c) 栈帧构成
      i. 参数区+临时变量区+子函数参数
      ii. 编译器优化后不一定保存参数区
3、 上下文切换场景
   a) 函数调用
   b) 例外和中断
   c) 系统调用
   d) 进程
   e) 线程
   f) 虚拟机
4、 同步与通讯
   a) 临界区：可能并发访问同一块数据的代码
   b) 锁的类型：自旋锁、互斥锁、读写锁
   c) LL/SC 操作

第05章　计算机组成原理和结构

重要部分：

1、 冯诺依曼结构基本原理：存储程序和指令驱动执行
2、 计算机组成部分
   a) 逻辑上

i. 输入输出设备
ii. 运算器
iii. 控制器
iv. 存储器

b) 物理上
i. CPU
ii. 内存
iii. IO 设备

3、 控制器提升性能技术
a) 指令流水线技术
b) 乱序执行技术：缓解由于指令相关引起的阻塞，使用重命名寄存器
c) 超标量技术
d) 转移预测技术
i. BHT 表

4、 存储
a) 存储介质分类：SRAM、DRAM、闪存和磁性存储介质
b) 存储层次：时间局部性和空间局部性
c) 高速缓存（Cache）
i. 降低失效率
ii. 降低失效延迟
iii. 降低命中延迟
iv. 提高 Cache 访问并行性

5、 输入输出设备
a) 硬盘
b) 闪存（Flash）
c) GPU

6、 计算机硬件结构演进
a) CPU+GPU+北桥+南桥
b) CPU+北桥+南桥
c) CPU+弱北桥+南桥
d) CPU+南桥
e) SOC 单片方案

7、 处理器和 IO 间通信
a) 三大部分关系
i. CPU/内存：CPULoad/Store 访问内存，高带宽、低延迟
ii. IO/内存：IO 通过 DMA 访问内存，较高带宽，高延迟
iii. CPU/IO:CPU 通过 PIO 访问 IO，低带宽、高延迟

b) CPU 与 IO 同步的关系
i. IO 寄存器寻址方式：专用指令、地址空间映射
ii. CPU 与 IO 设备间的同步：查询、中断

c) 存储器与 IO 设备间的通信
i. PIO
ii. DMA

8、 CPU、GPU 与 DC 间的数据传输

a) 模式一：CPU 与 DC 使用共享显存
b) 模式二：CPU 与 DC 使用独立显存
c) 模式三：CPU、GPU/DC 使用共享显存
d) 模式四：CPU、GPU/DC 使用独立显存

9、IO 中断控制器

第06章　计算机总线接口技术

重要部分：

1、总线规范：机械层、电气层、协议层和架构层
2、总线分类
a) 数据传送方向：单向、双向：半双工与全双工
b) 数据组织方式：并行、串行
c) 数据握手方式：无、Vallid-Ready 、Credit
d) 连接方式：共享信号、点对点
e) 时钟实现方式：全局时钟、源同步：时钟随数据一起发送和时钟嵌入到数据中发送
f) 总线实现位置：片上总线、内存总线和 IO 总线
3、片上总线：芯片内部模块互联使用的总线
a) 基本读写操作
i. 支持将读写请求按地址发送到目标模块
ii. 支持将读写响应返回到发起模块
b) 特点（与片外相比）
i. 引线资源丰富
ii. 全局时钟相对容易实现
iii. 不需要复杂的物理层转换
iv. 不使用三态信号
v. 连接单元为 buffer 和 mux
c) 性能
i. 频率
ii. 位宽
iii. 带宽利用率
d) 例子：AMBA 总线、APB 总线（单主设备、共享式、片选）、AHB 总线、
e) AXI 总线
4、内存总线
a) DRAM 内存结构
b) 内存总线信号分类：时钟信号、地址命令信号、数据及数据采样信号
5、系统总线：用于处理器与桥片的连接，同时也作为多处理器间的连接构成多路系统
a) 例子：HT 总线
6、IO 总线：连接处理器与输入输出设备的总线

第07章　计算机系统启动过程分析

重要部分：

1、处理器核初始化
a) 处理器复位
b) 外部调试接口初始化

c) 串口初始化
d) TLB 初始化
e) Cache 初始化

2、 总线接口初始化
a) 内存接口初始化
b) IO 总线初始化

3、 设备探测及驱动加载
a) 设备探测使用配置空间
b) 地址空间分配：基址寄存器（BAR）
c) PCI 设备的探测

4、 多核初始化
a) 核间通信机制：核间中断、信箱寄存器

复习提纲：

第一条指令从哪里取的？什么时候把内核拷贝到内存？

：0x01C00000,上述初始化过程完成后（唤醒其他核之前），BIOS 从硬盘把内核取到内存中（DMA 或者 IO），并跳转到内核地址。

第08章 运算器设计

重要部分：

1、 数的表示
a) 二进制
b) 定浮点表示
c) 原码、补码、移码
d) IEEE754 浮点数标准

2、 MOS 晶体管工作原理
a) NMOS 管、PMOS 管

3、 CMOS 逻辑电路：组合逻辑、时序逻辑、CMOS 晶体管级电路图

4、 定点补码运算器
a) 一位全加器
b) 行波进位加法器
c) 块内并行块间串行
d) 块间并行加法器

5、 减法器、比较器、移位器

6、 ALU 实现

7、 乘法器
a) 迭代式硬件乘法
b) 补码乘法算法
c) Booth 乘法器
d) 华莱士树

第09章 指令流水线

重要部分：

1、 简单 CPU 数据通路

2、 流水线处理器

a) 多周期 CPU：取指、译码、执行、访存 、写回
b) 流水线 CPU 设计逻辑：增加触发器，被处理的数据核指令的控制逻辑都要存入触发器随流水逐级传递下去

3、 指令相关：数据相关、控制相关、结构相关
a) 控制相关：跳转指令与取指 PC 的相关
i. 引入流水线阻塞来解除控制相关
ii. 转移延迟槽技术
b) 数据相关：RAW WAW WAR
i. 阻塞解决数据相关
ii. 前递解决数据相关
iii. 设计新的流水线结构，增加标志位

4、 异常与流水线：
a) 异常分类、异常发送的流水线阶段、异常特征
b) 异常处理

5、 指令调度技术
a) 编译器的静态调度，尽量避免程序执行时由于相关引起的阻塞
i. 循环展开技术
ii. 寄存器重命名
iii. 改变指令执行技术
iv. 增加发射宽度
b) 软件调度与硬件调度

6、 动态调度技术：保留站、寄存器增加域、ROB
流水阶段：取指、译码、发射、执行、写回、提交，有序进入、乱序执行、有序结束

7、 多发射技术

8、 转移猜测技术
a) BHT 表：一位 BHT 表、两位 BHT 表

9、 高速缓存技术
a) Cache 块的位置：全相联、组相联、直接相联
b) Cache 失效的替换机制：随机替换、LRU、FIFO
c) 写策略：
i. 写命中策略：写回、写穿透
ii. 写失效策略：写分配、写不分配
d) Cache 性能分析与优化：
i. 访存性能优化
ii. 降低失效率（MissRate）
iii. 降低失效延迟（MissPenalty）
iv. 降低命中延迟（HitTime）
v. 提高 Cache 访问并行性

$$CPUtime = IC \times \left(\frac{AluOps}{Inst} \times CPI_{AluOps} + \frac{MemAccess}{Inst} \times AMAT \right) \times CycleTime$$

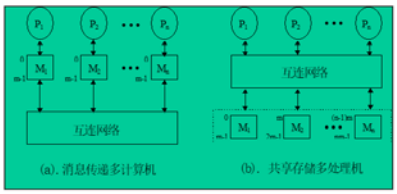
$$AMAT = HitTime + MissRate \times MissPenalty$$

第10章 并行编程基础

重要部分：

1、 计算机体系结构（Computer Architecture）是描述计算机各组成部分及其相互关系的一组规则和方法，是程序员所看到的计算机属性。
2、 计算机体系结构主要研究内容包括指令系统结构(Instruction Set Architecture，简称 ISA）和计算机组织结构（Computer Organization)。
   a) 微体系结构（Micro-architecture）是微处理器的组织结构。
   b) 并行体系结构是并行计算机的组织结构。
   c) 冯诺依曼结构的存储程序和指令驱动执行原理是现代计算机体系结构的基础。
3、 并行层次：指令级并行、数据级并行、任务级并行
4、 并行编程模型
   a) 单任务单数据并行 SIMD
   b) 多任务并行：共享存储、消息传递

消息传递与共享存储



- 多地址空间
- 消息传递通讯
- 编程困难、程序移植困难
- 通用性差
- 可伸缩性好

- 单地址空间
- 共享存储通讯
- 编程容易、程序易移植
- 通用性强
- 可伸缩性一般

5、 常见的并行处理结构：SIMD 结构、SMP 结构、CC-NUMA 结构、MPP 或机群结构
6、 典型的并行编程环境
   a) SIMD 编程
   b) Posix 编程环境
   c) OpenMP 编程环境
   d) MPI 编程环境
7、 通信模式：
   a) 点对点通信
      i. 阻塞与非阻塞方式
      ii. 四种通信模式：标准模式、缓冲模式、同步模式、就绪模式
   b) 集体通信：路障、广播、收集、散播、归约

第11章 多核处理器

重要部分：

1、 多核处理器的动力：工艺技术、功耗墙、并行结构的发展
2、 流行的 CPU 结构：多核+向量处理、众核、带有协处理器的异构多核
3、 共享存储多核处理器的关键问题

- 通用多核处理器一般采用共享存储结构
- 多个处理器核发出的访存指令次序如何约定？
  - 存储一致性模型：如顺序一致性、处理器一致性等
- 多个处理器核间共享片上Cache如何组织及维护一致性？
  - Cache一致性协议：片上Cache结构及Cache一致性协议
- 多个核处理器核间如何实现通信？
  - 片上互连结构
- 多个处理器核间如何实现同步？
  - 多核同步机制：互斥锁操作（lock）、路障操作（barrier）

4、 通用多核处理器的片上 Cache 结构：私有 Cache、片内共享 Cache、片间共享 Cache（典型结构：片内共享 LLC，片间共享内存）

5、 共享 LLC 结构：UCA 集中式共享结构、分布式共享结构

6、 存储一致性模型：顺序一致性模型、处理器一致性模型、弱一致性模型、释放一致性模型

7、 Cache 一致性协议：存储一致性模型对 Cache 一致性协议有制约作用

a) 如何传播新值：Write Invalidate vs. Write Update

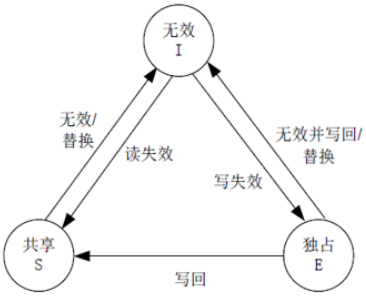
b) 向何处传播新值：Snoopy vs. Directory Protocol

8、 ESI 协议（重点）

a) E（Exclusive，独占）：表明对应 Cache 行被当前处理器核独占，当前处理器核可以随意读写，其他处理器核如果想读写这个 cache 行需要请求占有这个 cache 块的处理器核释放该 Cache 行

b) S（Shared，共享）：表明当前 Cache 行可能被多个处理器核共享，只能读取，不能写入

c) I （Invalid，无效）：表明当前 Cache 块是无效的



9、 多核处理器的互连结构

a) 常见互联结构：总线、交叉开关、片上网络

10、多核处理器的同步机制：

a) 同步机制：锁操作、路障操作、事务内存

b) LL/SC 原子指令对

11、典型多核处理器

第12章 计算机性能分析和评价

重要部分：

1、 各种性能评价指标

a) 执行时间或者响应时间

b) 归一化的执行时间（与一台标准机器相比）
c) 每条指令的时钟周期数（CPI）
d) 每秒钟执行百万条指令（MIPS）
e) 每秒钟执行百万条浮点指令（MFLOPS）
f) 每秒钟执行的事务（TPS）
g) 每秒帧率（FPS）
h) 带宽（MBPS）
i) 主频（MHz）

2、 CPU 时间的组成：

CPU时间的组成

- 指令数：算法、编译系统、ISA
- IPC：微结构、ISA、编译器
- 主频：微结构、工艺技术

$$CPUTime = \frac{Seconds}{Program} = \frac{Instructions}{Program} \times \frac{Cycles}{Instruction} \times \frac{Seconds}{Cycle}$$

3、 CPI 或 IPC

$$\text{CPU time} = \text{Cycle Time} \times \sum_{j=1}^{n} \text{CPI}_j \times \text{I}_j$$
$$= \text{Cycle Time} \times \text{CPI} \times \text{Instruction Count}$$

$$\text{CPI} = \sum_{j=1}^{n} \text{CPI}_j \times \text{F}_j \quad \text{where } \text{F}_j = \frac{\text{I}_j}{\text{Instruction Count}}$$

4、 影响 CPU 频率的因素：工艺技术、微结构、逻辑设计、物理设计
5、 并行系统的评价指标：可扩展性、加速比
a) Amdahl 定律：并行效果受串行部分限制

$$\text{ExTime}_{new} = \text{ExTime}_{old} \times \left[(1 - \text{Fraction}_{enhanced}) + \frac{\text{Fraction}_{enhanced}}{\text{Speedup}_{enhanced}} \right]$$

$$\text{Speedup}_{overall} = \frac{\text{ExTime}_{old}}{\text{ExTime}_{new}} = \frac{1}{(1 - \text{Fraction}_{enhanced}) + \frac{\text{Fraction}_{enhanced}}{\text{Speedup}_{enhanced}}}$$

6、 基准测试程序（黑盒法）

常见的基准测试程序类型

- 微基准测试程序：Sim-Alpha, Coremark, LMBench, Stream
- 串行CPU基准测试程序：SPEC CPU, EEMBC
- 并行CPU基准测试程序：Splash2, PARSEC, NPB, Linpack
- 专项基准测试程序：SPECjvm, SPECjbb, SPECsfs, SPECviewperf, TPC

7、 性能分析方法（白盒法）
a) 性能分析和评估的分类

- 测量平台及微结构和软件及数据结构之间的交互行为
- 基于分析和统计的建模
  - 概率模型、队列模型、马科夫模型、Petri网模型
- 基于模拟的建模
  - 踪迹驱动模拟、执行驱动模拟、全系统模拟、事件驱动模拟、统计方法模拟
- 性能测量
  - 片上的硬件监测、片外的硬件监测、软件监测、微码插桩

8、 模拟建模与模拟器

a) 模拟器类型：系统模拟器和部件模拟器、全系统模拟器和用户模拟器、执行驱动和踪迹驱动、时钟驱动和事件驱动、功能模拟器和时许模拟器

b) 模拟技术：

i. 统计模拟技术

ii. 采样模拟技术

iii. 时序优先技术

c) 模拟器例子：SimOS 模拟器、Simplescalar、GEM5 模拟器

9、 性能测量的方法

a) 片上性能计数器

b) 处理器性能分析工具

i. Perf