

# 并行计算与多核程序设计

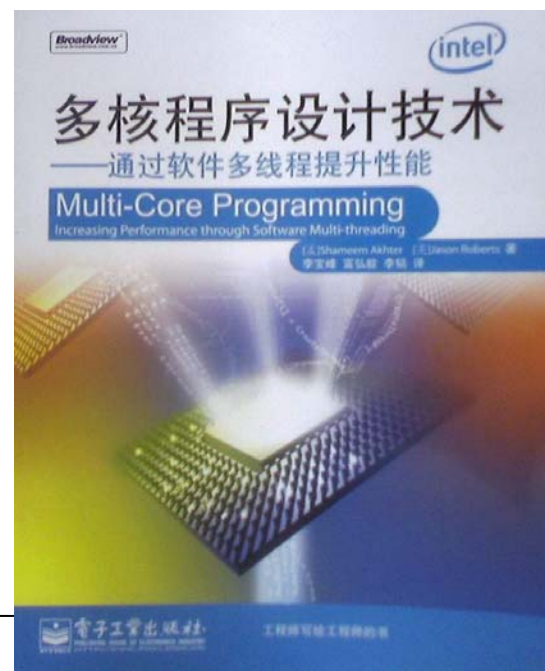
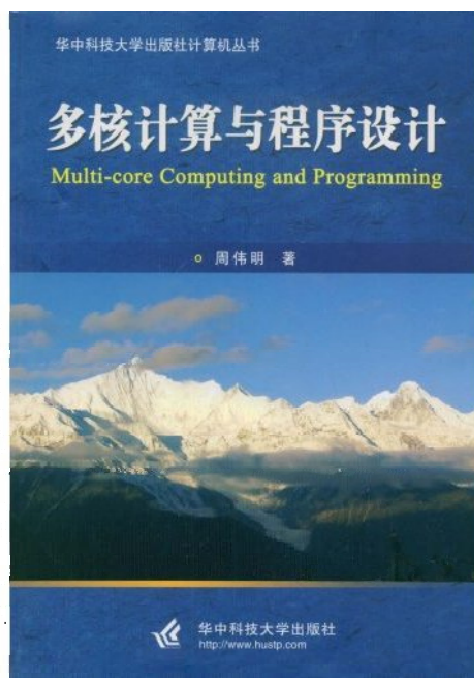
## 理论课0 课程概述

杨剑锋，谢银波

{yjf.whu, xyb.whu}@gmail.com

# 教材

- ❑ 多核计算与程序设计, 周伟明, 华中科技大学出版社
- ❑ 多核程序设计, 陈天洲等, 清华大学出版社, 2007年8月
- ❑ 多核程序设计技术-通过软件多线程提升性能, 电子工业出版社, 2007年3月



理论	内容细节	学时
多核技术导论	并行体系与多核体系结构的起源和特点； 典型多核芯片Cell、Intel、AMD双核芯片体系分析； 多核系统软件对并行编程的支持。	2
并行计算基础	并行计算机体系结构； 并行计算模型； 并行环境、编程语言与并行编译器； 几种常用并行算法与算法评估方法。	2
线程的基本概念	进程、线程的概念； 用户级线程、内核级线程和多线程的映射模型； 多线程的互斥、同步的基本概念。	2
Windows多线程编程及调优	Windows平台下的线程库，包括Win32 线程库、MFC线程库以及.Net Framework线程库； 重点介绍如何使用Win32 API来创建线程、管理线程； 如何实现线程间同步，包括MFC和.Net Framework进行多线程同步的方法； 使用调试工具对多线程程序进行调试。	4
Linux 多线程编程	POSIX 线程操作的一系列的相关的函数，包括基本线程创建、撤销、退出函数； 线程之间的互斥和同步的操作； 使用GDB 调试多线程程序。	2
多线程性能研讨	对比Linux、windows平台，对比各种线程库，对比各种同步机制，与学生探讨多线程程序的性能分析与调优方法 结合具体实例，剖析线程工具的使用方法	2

OpenMP编程优化	OpenMP简介； OpenMP多线程编程方法； 性能分析。	4
OpenMP研讨	对比OMP与多线程，研究多线程适应于哪些应用环境，OMP适应于哪些环境，OMP的各种编程方法适应于哪些场合 结合高性能计算实例，剖析OMP对核的利用率	2
MPI编程及性能优化	MPI简介； MPI程序特点和多核MPI软件包安装配置； MPI程序框架与标准的点对点通信、群集通信和排错； 性能分析优化。	2
MPI研讨	分析MPP上与CMP上MPI的差异性 对比MPI与多线程、OMP的编程思想的差异性 以曙光5000A为例，分析现有超级计算机上的软件设计方法以及CPU利用率	2
多核软件工具	硬件性能评测工具； 多线程程序的性能评测方法； 评测优化工具使用，包括C++编译器、VTune性能分析器、MKL数学核心函数库、Thread Checker线程检查器和Thread Profiler线程档案器。 Parallel studio工具。	4
性能调优研讨	对比SMT（SUN）、CMP（intel&AMD）、CELL（IBM）的性能调试 谈论硬件级支持的性能监控手段、OS级性能检测方法 分析现有工具的不足	2
综合创新实验及经验交流	学生设计与实现综合性的并行计算与多核程序设计实验。鼓励与科研项目想结合，学生将设计过程和结果做汇报	2

# 学完本课程的结果

---

- 学会如何编写多核计算机上的软件
  - 多个处理器运行你的程序
- 学会如何评价一个多核软件
  - 在不同核上充分利用硬件资源
- 学会如何多核优化
  - 对一个单核程序，能编出多核程序，能充分优化
- 学会多核硬件的评价
  - 知道软件是怎么在多核硬件上跑
- 扩大处理器视野
  - 了解各种处理器以及在这些硬件上编程的差异
- 有3学分的成绩
  - 上机考试，以时间定成绩

# 并行计算与多核程序设计

## 理论课1-2 多核技术导论

杨剑锋，谢银波

{yjf.whu, xyb.whu}@gmail.com

# 1 微处理器发展史

---

- 1945年，世界上第一台全自动电子数字计算机ENIAC
- 电子管微处理器：忽略
- 处理器发展
  - 第一代（1971~1973）：4位或8位微处理器。代表：8008
  - 第二代（1974~1977）：集成度提高1-4倍，运算速度提高10-15倍。代表：Z80，8080
  - 第三代（1978~1984）：16位。代表：8086。
  - 第四代(1985~1992)：32位微处理器，代表80386
  - 第五代微处理器(1993—1995年)，64位数据总线，32位地址总线，CPU内部采用超标量流水线设计，代表：奔腾，K5，powerpc
  - 第六代（1993~2002）：350nm以下工艺微处理器，平行并发计算而设计(EPIC)架构，代表：安腾
  - 多核时代（2002~今）：从2002年超线程技术开始的多核时代，代表：酷睿2

# 微处理器

---

- 1971年，第一款微处理器4004
- 1978年，英特尔公司8086
- 1979年，英特尔公司又开发出了8088
- 1981年，美国IBM公司将8088芯片用于其研制的PC机
- 1982年，英特尔公司研制出了80286微处理器
- 1985年10月17日，英特尔80386 DX正式发布
- 1989年，英特尔推出80486芯片
- 1993年，新一代586 CPU问世
- 1996年底，多能奔腾Pentium MMX
- 1998年，奔腾二代Pentium II 、至强Xeon、赛扬Celeron
- 1999年春，英特尔公司Pentium III奔三
- 2000年，英特尔公司Pentium 4，奔四

-----以下为本课程关注的时代-----

- 2002年，英特尔超线程技术（HT，Hyper-Threading），预告多核时代
- 2005年4月18日，英特尔全球同步首发奔腾D处理器（双芯片方案）
- 2005年4月21日，AMD推出了双核速龙™ 64 X2处理器（双核）
- 2006年7月27日，英特尔推出酷睿2：英文Core 2 Duo
- 2009年9月，至强六核
- 2009年10月，酷睿i7



# 汇编指令集变化

## □ 提高电脑在多媒体、3D图形方面的应用能力

### ■ MMX

- 57条多媒体指令

### ■ SSE

- Streaming SIMD
- Extensions

### ■ 3D NOW!

- 3D加速指令集

## □ 怎么利用这些指令

- 用专业编译器

## □ 2009年最新SSE4.2

- 4.1包括47条指令，4.2包括7条指令。
- 4.2增加两类汇编：STTNI(STring & Text New Instructions)和ATA(Application Targeted Accelerators)
- 加速XML文本的字符串操作、存储校验
- XML的解析速度最高是原来的3.8倍。
- 这就是为什么office2007采用XML，运行在新处理器上速度快？



改进了的英特尔® 酷睿™ 微架构 (Penryn) 支持 47 条新的指令  
这是SSE,SSE2,SSE3和SSSE3 趋向的延续

# 多核概念

---

- ❑ 单芯片多处理器（Chip multiprocessors, 简称CMP）
- ❑ CMP是由美国斯坦福大学提出的
- ❑ 将大规模并行处理器中的SMP（对称多处理器）集成到同一芯片内，各个处理器并行执行不同的进程
- ❑ CMP vs SMT
  - SMT处理器结构的灵活性比较突出
  - 当半导体工艺进入0.18微米以后，线延时已经超过了门延迟，要求微处理器的设计通过划分许多规模更小、局部性更好的基本单元结构来进行
  - 由于CMP结构已经被划分成多个处理器核来设计，每个核都比较简单，有利于优化设计，因此更有发展前途

## 2 多核技术是处理器发展的必然

---

- 推动微处理器性能不断提高的因素
  - 半导体工艺技术的飞速进步
  - 体系结构的不断发展
- 这两个因素相互影响，相互促进
  - 工艺和电路技术的发展使得处理器性能提高约20倍
  - 体系结构的发展使得处理器性能提高约4倍
  - 编译技术的发展使得处理器性能提高约1.4倍。
- 如今，这个规律已经很难维持

# 十亿晶体管时代即将到来

---

- 通用微处理器的主频已经突破了4GHz
- 数据宽度也达到64位。
- 65nm工艺的微处理器已经批量生产
  - 大连
- 45nm工艺以下的微处理器也已问世
- 芯片上集成的晶体管数目已经超过10亿个
  - 如何有效地利用数目众多的晶体管？
- 多核
  - 通过在一个芯片上集成多个简单的处理器核
  - 利用这些晶体管资源
  - 发挥其最大的能效

# 门延迟逐渐缩短，全局连线延迟却不断加长

---

- 晶体管特征尺寸不断缩小
  - 使得晶体管门延迟不断减少
  - 互连线延迟却不断变大
- 芯片的制造工艺达到0.18微米甚至更小时
  - 线延迟已经超过门延迟
  - 成为限制电路性能提高的主要因素
- CMP(单芯片多处理器)的分布式结构
  - 全局信号较少，与集中式结构的超标量处理器结构相比，在克服线延迟影响方面更具优势

# 符合Pollack规则

---

## □ Pollack规则

- 处理器性能的提升与其复杂性的平方根成正比
- 如一个处理器的硬件逻辑提高一倍，至多能提高性能40%
- 采用两个简单处理器构成一个相同硬件规模的双核处理器
  - 可获得70%~80%的性能提升
  - 在面积上也同比缩小

# 能耗不断增长

---

- ❑ 芯片的发热现象日益突出
  - 工艺技术的发展 and 芯片复杂性的增加
- ❑ 多核处理器
  - 单核速度较慢
    - ❑ 处理器消耗较少的能量，产生较少的热量
  - 单核处理器里增加的晶体管可用于增加多核处理器的核
  - 多核处理器关闭（或降频）一些处理器等低功耗技术

# 设计成本的考虑

---

- 设计成本随时间呈线性甚至超线性的增长
  - 处理器结构复杂性的不断提高
  - 人力成本的不断攀升
- 多核处理器
  - 处理器IP复用
  - 极大降低设计的成本
  - 模块的验证成本也显著下降



# 微电子技术进步的效果

---

- 1982年以来:
  - CPU性能提高3500倍，内存价格下降45000倍，硬盘价格下降360万倍
- 全球人均拥有晶体管数2004年500个2010年10亿个
- 现在45nm处理器中每个晶体管的价格，仅相当于1968年的百万分之一。
- 2010年平均每个晶体管成本只有十万分之一美分

# 微纳技术正在重构集成电路产业体系

---

- 晶体管尺寸缩小， $\text{SiO}_2$ 栅介质变薄，栅极漏电流增加，功耗上升，传统微电子技术临近极限。
- 集成电路技术十年内仍可能按摩尔定律增长，将沿着两个方向发展：
  - 采用高K栅介质和金属栅极的创新工艺，缩小特征尺寸的技术有重大突破；
  - 发展固态量子器件等：包括单电子器件、自旋器件、磁通量器件等和基于自组装的原子和分子器件与一维结构的纳米线、碳纳米管等低维器件。
- 未来的信息功能器件将充分利用微电子、纳米（甚至生物纳米）传感器MEMS技术等微纳技术正在重构集成电路产业体系
- 集成器件由单纯追求性价比正在转向追求功耗比的技术创新

# 体系结构发展的必然

---

- 超标量结构和超长指令字结构
  - 在目前的高性能微处理器中被广泛采用
  - 超标量结构
    - 使用多个功能部件同时执行多条指令，实现指令级的并行
    - 控制逻辑复杂，实现困难
    - 超标量结构的指令并行程度一般不超过8
  - 超长指令字结构
    - 使用多个相同功能部件执行一条超长的指令
    - 编译技术支持问题
    - 二进制兼容问题。
- 未来的主流应用需要同时执行更多条指令的能力
  - 单一线程不太可能提取更多的并行性
  - 不断增加的芯片面积提高了生产成本
  - 设计和验证所花费的时间变得更长
- 更复杂化的设计也只能得到有限的性能提高

- 
- 单芯片多处理器通过在一个芯片上集成多个微处理器核心来提高程序的并行性
    - 每个微处理器核心是一个相对简单的单线程微处理器或者比较简单的多线程微处理器
    - 多个微处理器核心就可以并行地执行程序代码
    - 具有了较高的线程级并行性
  - 由于CMP采用了相对简单的微处理器作为处理器核心
    - 使得CMP具有高主频、设计和验证周期短、控制逻辑简单、扩展性好、易于实现、功耗低、通信延迟低等优点
  - CMP能充分利用指令级并行和线程级并行
  - 目前CMP已经成为处理器体系结构发展的一个重要趋势

### 3 并行计算机

---

- 由一组处理单元组成，这组处理单元通过相互之间的通信与协作，以更快的速度共同完成一项大规模的计算任务。
- 出现背景：
  - 60年代初期，晶体管以及磁芯存储器的出现，处理单元变得越来越小，存储器也更加小巧和廉价。出现规模不大的共享存储多处理器系统，即大型主机（Mainframe）。
  - 60年代末期，同一个处理器开始设置多个功能相同的功能单元，流水线技术也出现了，在处理器内部的应用大大提高了并行计算机系统的性能。
- 两个最主要的组成部分
  - 计算节点
  - 节点间的通信与协作机制

# 并行计算机的弗林分类

---

- Flynn根据指令流和数据流的不同组织方式，把计算机系统的结构分为以下四类：
  - 单指令流单数据流（Single Instruction stream Single Data stream, SISD）
  - 单指令流多数据流（Single Instruction stream Multiple Data stream, SIMD）
  - 多指令流单数据流（Multiple Instruction stream Single Data stream, MISD）
  - 多指令流多数据流（Multiple Instruction stream Multiple Data stream, MIMD）

# 并行计算机系统结构分类

---

- 分布式存储器的SIMD处理机
  - 含有多个同样结构的处理单元（PE），通过寻径网络以一定方式互相连接。每个PE有各自的本地存储器（LM）。
- 向量超级计算机（共享式存储器SIMD）
  - 集中设置存储器，共享的多个并行存储器通过对准网络与各处理单元PE相连。在处理单元数目不太大的情况下很理想。
- 对称多处理器（SMP）
  - 一个计算机上汇集了一组处理器，各处理器之间共享内存子系统以及总线结构。
- 并行向量处理机（PVP）
- 集群计算机

## 4 片上多核处理器架构

---

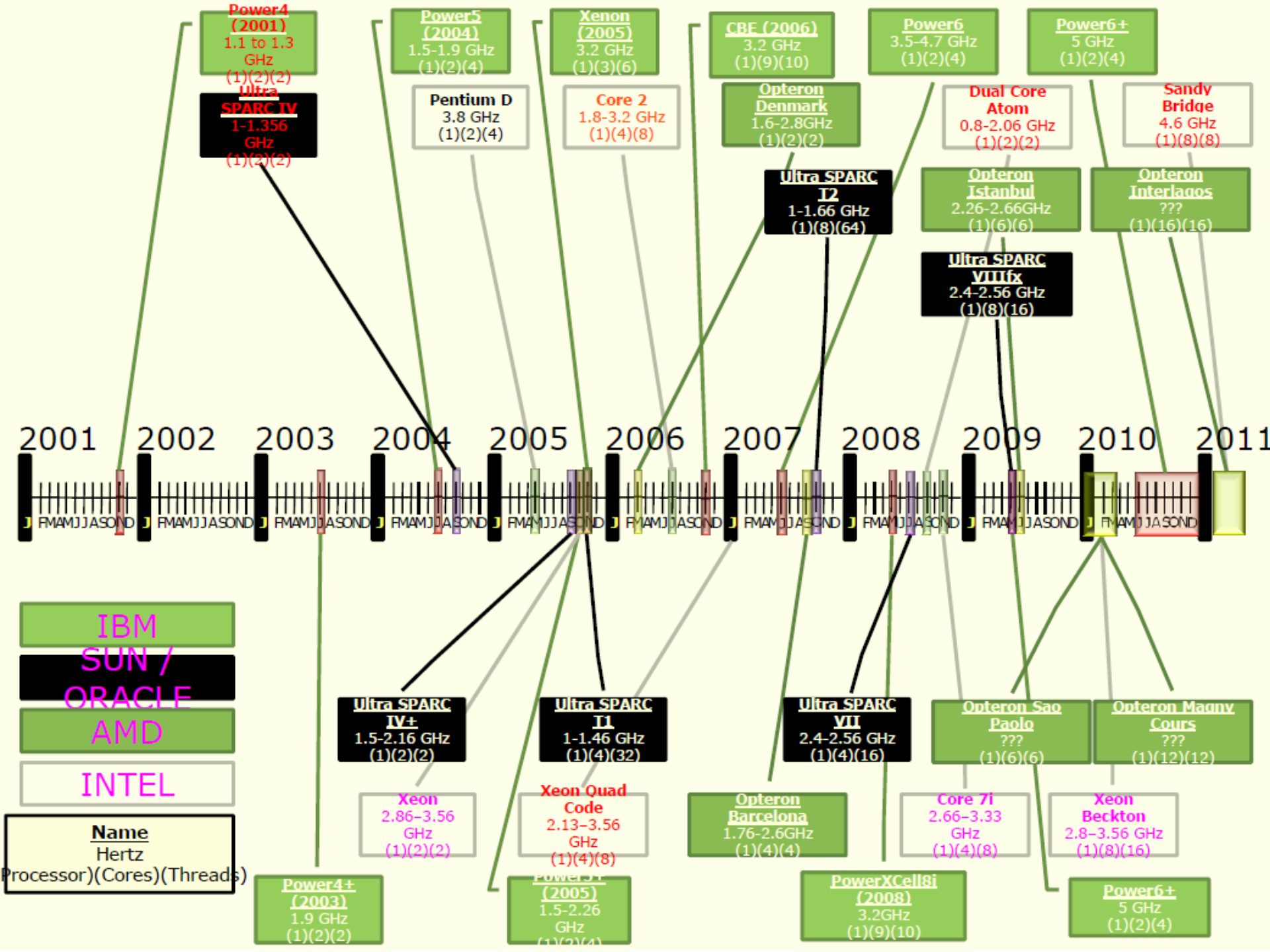
- 片上多核处理器（Chip Multi-Processor, CMP）就是将多个计算内核集成在一个处理器芯片中，从而提高计算能力。
- 按计算内核的对等与否，CMP可分为同构多核和异构多核
- CPU核心数据共享与同步
  - 总线共享Cache结构：每个CPU内核拥有共享的二级或三级Cache，用于保存比较常用的数据，并通过连接核心的总线进行通信。
  - 基于片上互连的结构：每个CPU核心具有独立的处理单元和Cache，各个CPU核心通过交叉开关或片上网络等方式连接在一起。
- 给程序开发者带来的挑战



# 主流多核处理器

---

- 2002年
  - 超线程技术（HT, Hyper-Threading）
- 2005年4月18日
  - 奔腾D处理器，第一个多核
- AMD K8架构
- Intel Core微架构
- Intel 四核
- AMD 四核



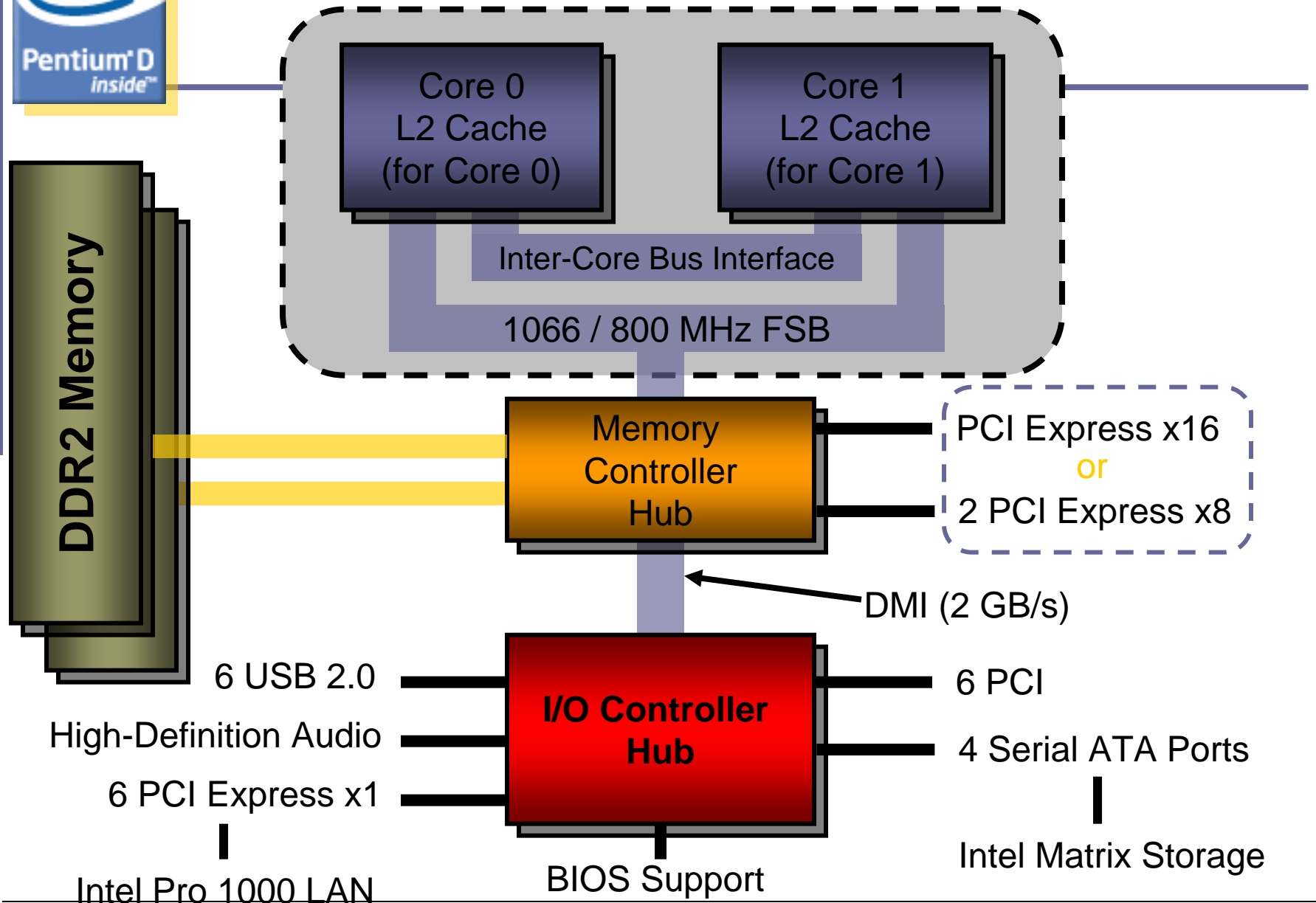
# 多核特点

---

- 优势
  - 功耗低
  - 性能高
- 缺点
  - 编程复杂



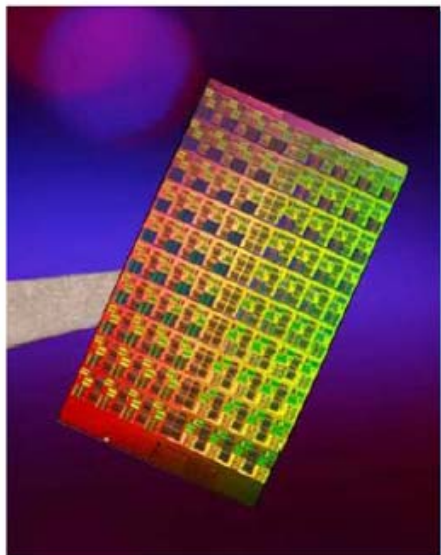
# Pentium D with 975X Chipset



# 80核计算机

## 万亿级芯片原型问世

1亿个晶体管 ● 80 个块面 ● 275mm<sup>2</sup>



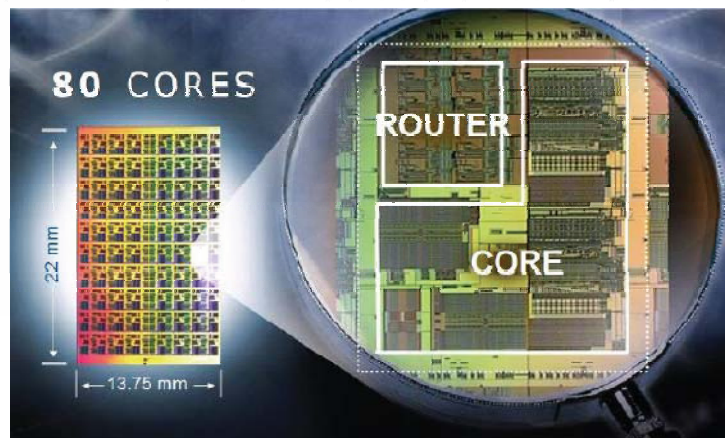
首个万亿级可编程硅片：

- 万亿次浮点运算的性能
- 块面式设计方法
- 片上网状网络
- 新型时钟门控
- 功率监控能力
- 支持3维内存

并非为IA或产品而设计

## 最终结果

在62瓦功率下示范每秒万亿次浮点运算性能



# 80核处理器

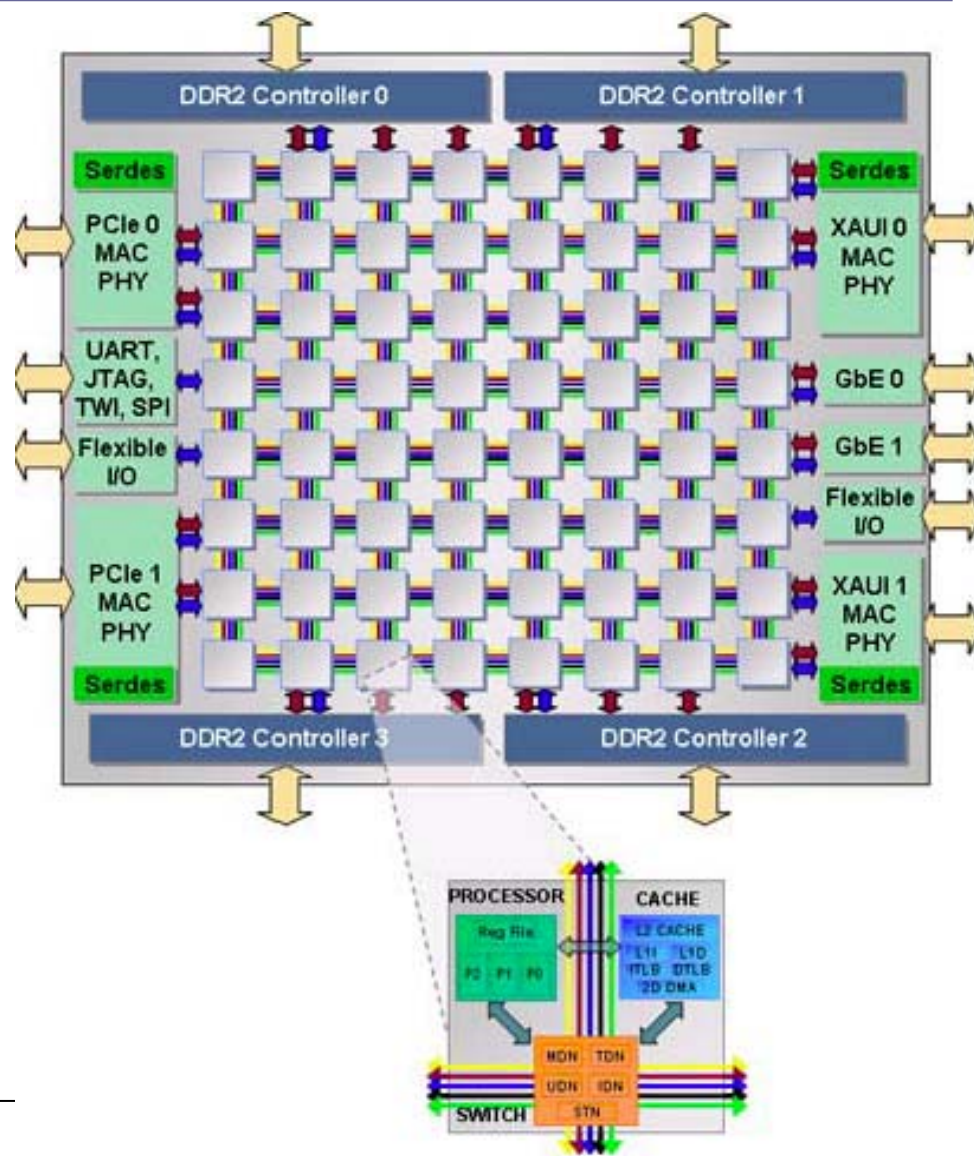
- 2007-02-11
- 核心面积275平方毫米
- 主频3.16GHz
- 电压0.95V
- 数据带宽1.62Tb/s
- 浮点运算能力1.01TFlops
  - 相当1万颗10年前的Pentium Pro
- 功耗不过62W
  - 比core 2 duo还低
  - 是四核心Xeon X5355 2.66GHz的大约一半，浮点运算能力估计为50-60GFlops(每秒十亿次)





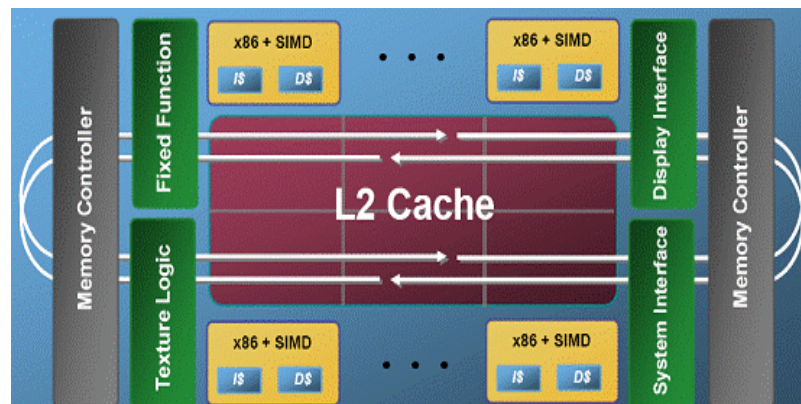
# Tilera 64-core processor

- MIT的Tilera公司
- 90纳米工艺
- 工作频率600~900MHz
- 总体功耗不过19.2W
- 总体性能是英特尔双核Xeon的10倍
- 每瓦特性能是30倍
- 每万颗的批发价435美元
  - 每千颗四核2.33GHz Xeon处理器是455美元
- 2009年10月26日宣布了下一代的100核计划。40nm



# Larrabee众核

- 图形
  - 支持 DirectX 和 OpenGL
- 高吞吐率
- 8到64个处理器核
  - 每个核是早期Pentium
  - 每个处理器核包含一个16路的SIMD处理单元
  - 包含固定功能的纹理处理单元
- 整个芯片支持完整的cache一致性协议
- 计算能力:
  - 1GHz的主频、32个处理器核的情况下，处理器的计算能力能达到1 TeraFLOPS





# Godson-T龙芯众核（64核）

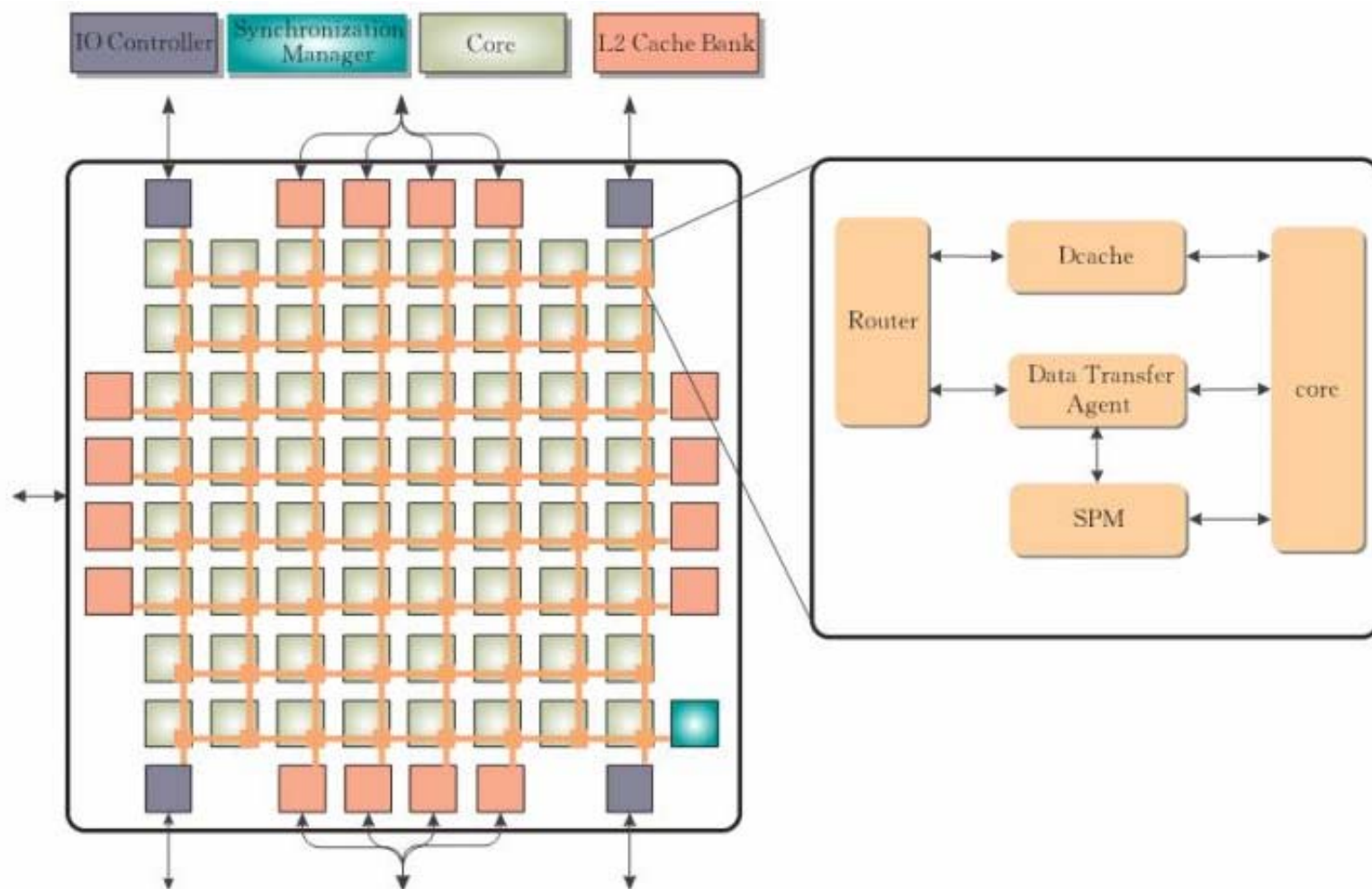


图 2 Godson-T 结构图

# 系统芯片

---

- 或称片上系统（System on Chip, 简称SoC）
- 在一个芯片上实现完整功能
  - 信号采集、转换、存储、处理和I/O 等
- IP核复用
  - IP, Intellectual Property, 知识产权
- 主要用于嵌入式系统

## 5 芯片组对多核的支持——固件

- ❑ 固件：嵌入到硬件设备中的软件
  - 烧写在flash等介质中
  - 可以被当作一个二进制映像文件由用户从硬件设备中调用
- ❑ 在集成电路只读存储器中的计算机程序
- ❑ 是可擦写可编程芯片
- ❑ 程序可以通过专门的外部硬件进行修改
- ❑ 不能被一般的应用程序改动。

操作系统和应用软件

内核

汇编器

固件

硬件

# BIOS (Basic Input/Output System)

---

## □ 功能

- 系统硬件和操作系统之间的抽象层
  - 初始化和配置系统的硬件
  - 启动操作系统
  - 提供对系统设备底层的通讯
- 连接CPU、芯片组和操作系统的固件
- IBM兼容计算机中启动时调用的固件代码

## □ 组成

- 上电自举即POST (Power On Self Test)
- 在线的中断服务 (主要由legacy 操作系统使用)

# BIOS 续

---

## □ 流程:

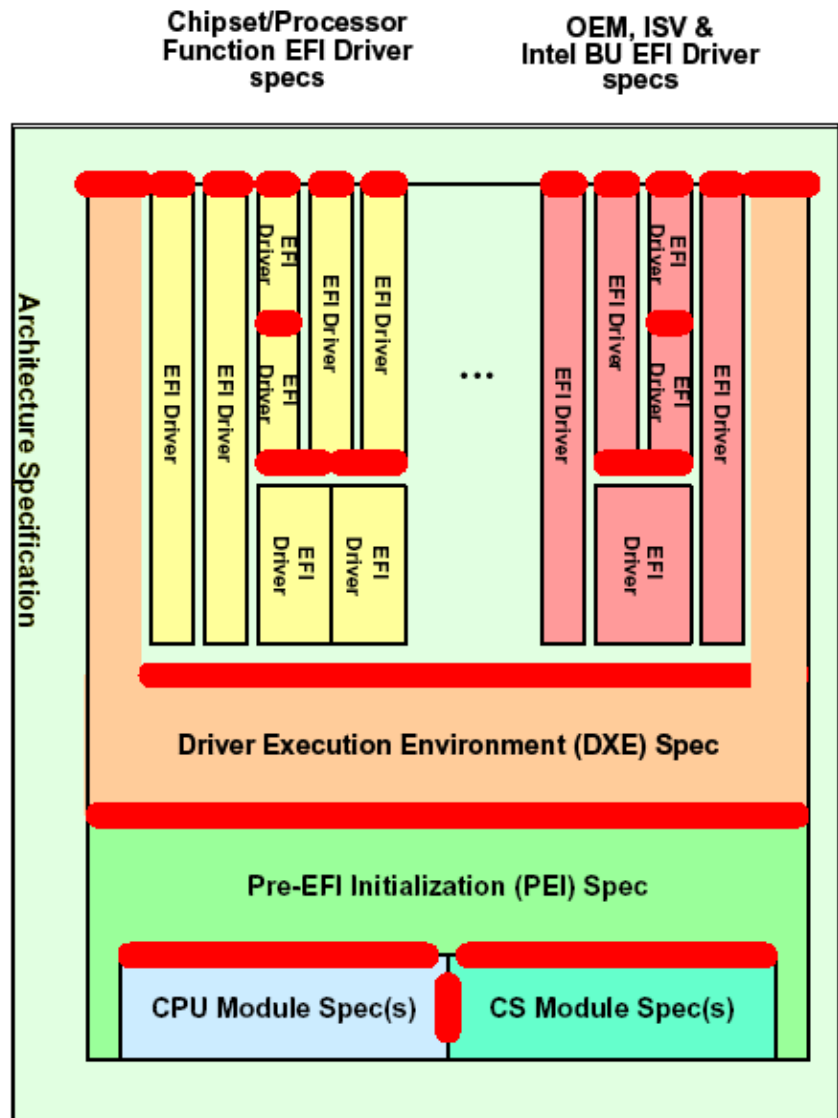
- 加电时BIOS从flash、PROM或是EPROM中启动并完成初始化
- 进行加电自检
- 对硬盘，内存，显卡，主板等硬件进行扫描检查
- 将自己从BIOS内存空间中解压到系统的内存空间中
- 开始从那里运行

## □ 正在被以EFI所取代

# 可扩展固件接口EFI

## □ Extensible Firmware Interface

- 操作系统与平台固件之间的软件接口
- 接口包括包含平台信息的数据表和启动时及启动后的服务
- EFI启动管理器被用来选择装载操作系统，不再需要专门的启动装载机机制辅助
- Framework是一种固件的架构，是EFI固件接口的一种实现



# UEFI

---

## Unified EFI联盟

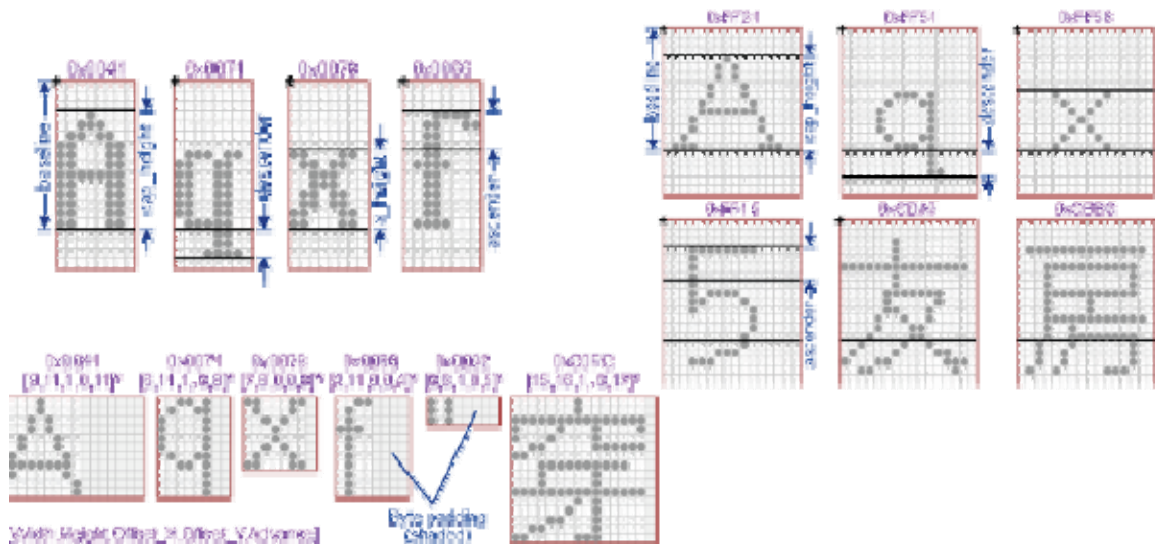
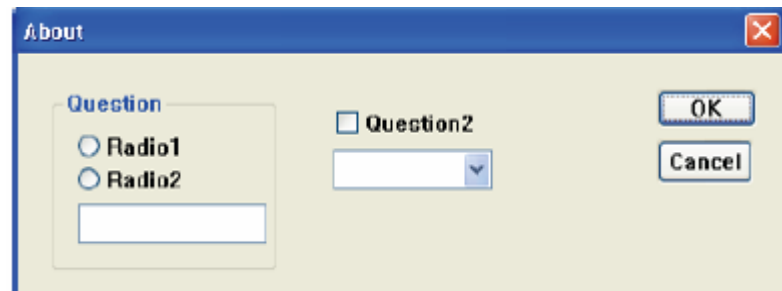
- 在华盛顿州注册的非赢利性公司
  - 定义、推广和管理UEFI规范的演变
  - 继续减少规范被接受的障碍
- 推广者：
  - IBVs: AMI, Insyde, Phoenix
  - OEMs: Dell, HP, IBM, Lenovo
  - AMD, Apple, Intel, Microsoft
- 成员等级：
  - 推广者, 贡献者和接受者
- 更多信息: [www.uefi.org](http://www.uefi.org)

## UEFI 2.1简介

- 约一年的标准制定工作
  - 基于UEFI 2.0
- 董事会于2007年1月23日正式批准
  - 可从[www.uefi.org](http://www.uefi.org)下载
- 新内容提要：
  - 用户接口基础架构
  - 硬件出错纪录支持
  - 已验证变量的支持
  - 扩展简单输入支持
  - 点对点设备协议

# UEFI 2.3发布, 2009年5月

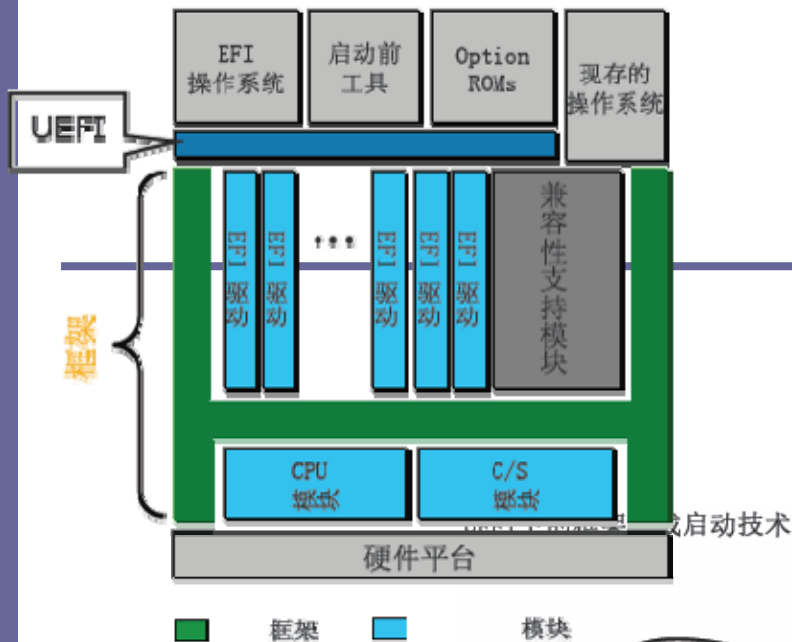
- 主板启动过程中的软件设计
  - 支持网络
  - GUI提供本地字符、表格
  - Eg: 启动过程的网络视频播放
- 增加了对ARM的支持
  - 已经在XScale上实现了UEFI



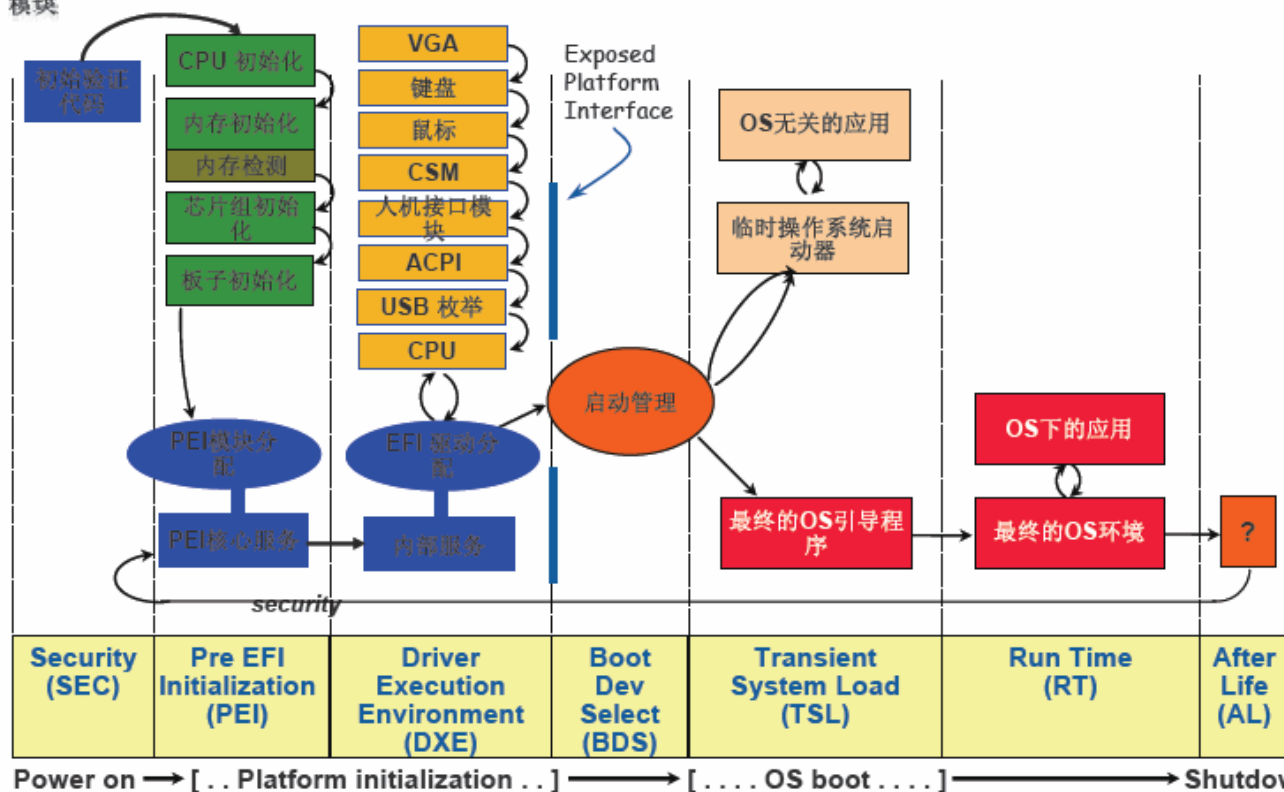


# EFI对多核支持

- 在Framework中定义了两类处理器
  - BSP (boot strap processor)
    - 执行EFI的初始化代码，设置APIC环境，建立系统范围的数据结构，开始并初始化AP。
  - AP (application processor)
    - 在系统上电或重启之后，AP会自己进行一个简单的设置，然后就等待BSP发出Startup信号。
- Framework在多核计算机中初始化过程如下：
  - SEC：从实模式切换到保护模式，处理不同的重启事件、对每个处理器进行缓存设置。
  - PEI：做尽量少的硬件初始化，而把更多的留给DXE。
  - DXE：对所有可用的硬件设备进行初始化，为建立控制台和启动操作系统提供必要的服务。
  - BDS：建立所需的控制台设备，在输出控制台上显示用户界面。
- EFI需要提交操作系统包括处理器在内的有关信息



## 框架启动执行顺序



# 一些案例

- Linux Bios
- 5秒钟启动片内操作系统

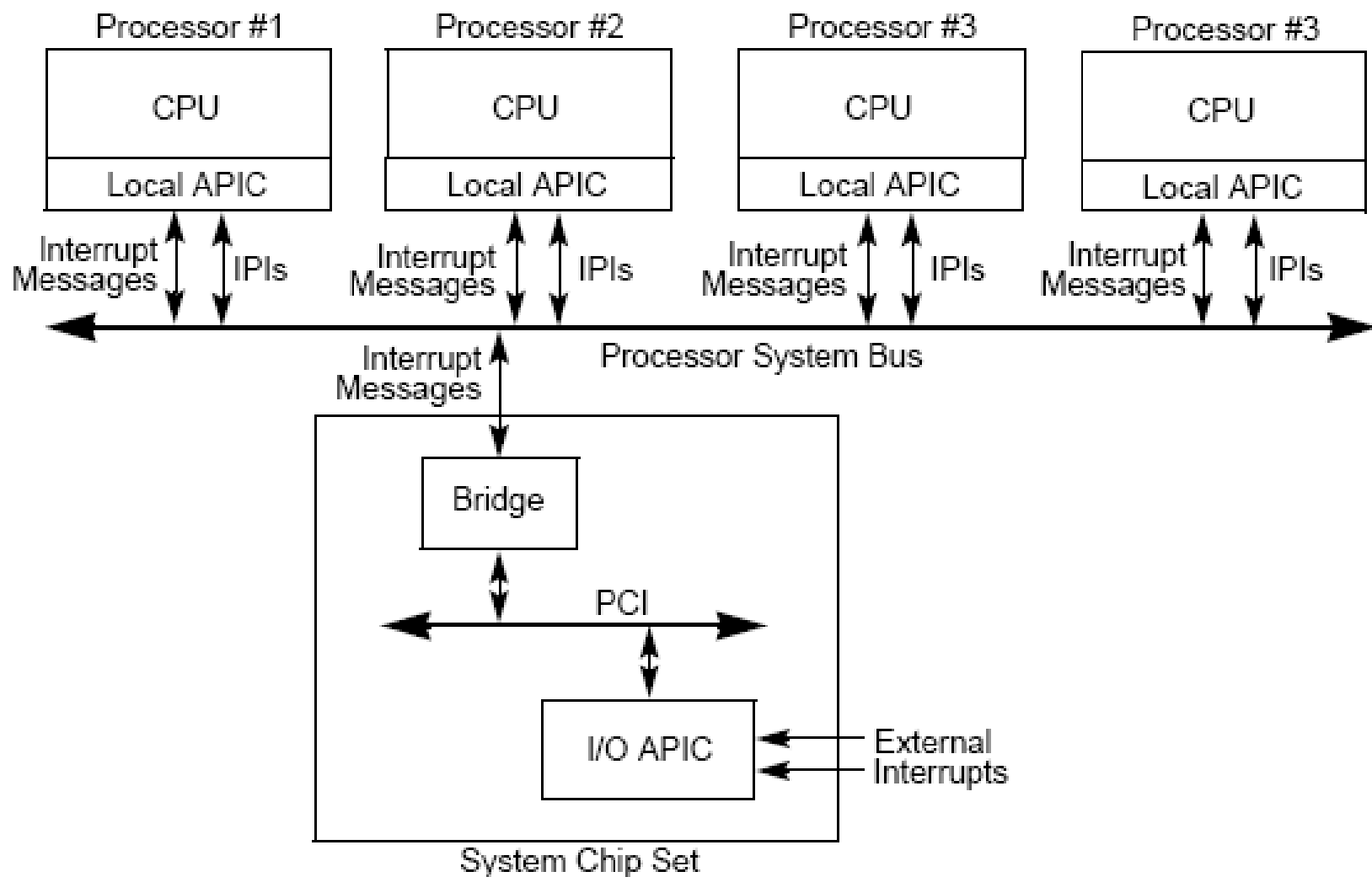
实现需求

- 对当前框架作最小的修改
  - 保护对当前框架的已有投资
  - 降低调试和代码维护难度
  - 现存的符合UEFI标准的驱动必须能在新的集成启动环境内运行
  - 不修改代码就能实现新需求
- 我们需要突破PC思维方式来支持HPC和嵌入式系统的市场需求



当前的固件实现方案存在不足

# 多核高级可编程中断控制器 APIC



# 6 虚拟化技术

---

## □ 虚拟化

- 在同一台物理服务器上运行多个软件应用
- 运行在不同的操作系统上
- 用更少的服务器获得更高的整体性能，提高计算机性能的
- 利用率

## □ 优势

- 高效、节能、节省空间、省电、兼容性

## □ 硬件及软件的虚拟化

## □ 平台虚拟化

## □ 资源虚拟化

# 7 操作系统对多核处理器的支持方法

---

## □ 调度与中断

- 对任务的分配进行优化。使同一应用程序的任务尽量在一个核上执行。
- 对任务的共享数据优化。由于CMP体系结构共享二级缓存，可以考虑改变任务在内存中的数据分布，使任务在执行时尽量增加二级缓存的命中率。
- 对任务的负载均衡优化。当任务在调度时，出现了负载不均衡，考虑将较忙处理器中与其他任务最不相关的任务迁移，以达到数据的冲突量小。

## □ 输入输出系统

## □ 存储管理与文件系统

# 8、GPGPU

---

## □ GPU/GPGPU的编程模型

- OpenCL/CUDA
- 本课程不介绍
- 将来会有一门课程《GPU/GPGPU设计方法》中介绍

## □ Multicore/manycore的编程模型

- 多核、众核（超过八个处理器核的设计就被称为‘众核’处理器）
- 多线程，OpenMP，MPI库，SHMeM 库
- 均属于本课程的内容

## 9、2020—2030年之间可能出现的“技术墙”

---

- 许多信息技术不约而同地将在2020-2030年之间出现难以逾越的障碍。
- 到2020年左右，摩尔定律将不再有效，集成电路正在逐步进入“后摩尔时代”，我们必须更多地从Beyond CMOS中寻找新的出路。
- 计算机正逐步进入“后PC时代”，终端设备将从“高大全”向“低小专”（“专”指个性化）转变，降低功耗是首要目标。
- 2020年以后，超级计算机的“千倍定律”将失效，只在现有的技术基础上做改进，2030年肯定做不出Zettaflops级（ $10^{21}$  flops）水平的计算机。
- 进入“后IP”时代是不可避免的发展过程，可能需要20年时间才能真正突破TCP/IP协议的局限。



谢谢