

2 POWER7 Server

基本介绍

RISC 指令架构, 8 核 4 线程, 45nm 工艺

分离的指令发射队列

追踪指令完成, GCT

负责冲刷流水线

主要提升

- 1) 更多的核心, 从双核到八核
- 2) 更大的 eDRAM 作为缓存
- 3) 高性能的 4 线程核心

指导目标

- 1) 降低功耗和面积
- 2) 保持和第六代的兼容性
- 3) 放弃额外的 L3 缓存
- 4) 加倍的浮点运算资源

POWER7 核心

- 1) 动态多线程: ST, SMT2, SMT4
- 2) 隔离的 SMT4, 每两个线程共享一组资源
- 3) 全局统一的寄存器重命名结构
- 4) 每个模块具有独立的电压和频率管理
- 5) IFU、ISU、LSU、FXU、VSU、DFU
- 6) 更高的带宽满足高性能计算
- 7) 提供 POWER6 模式
- 8) RAS 特性, 检错和纠错能力

1. 指令获取 IFU

32KB4 组相连指令缓存 I-cache, 16 个 bank
64 项 IEADIR 提供组预测
64 项 IERAT 提供地址转换
预取引擎从 L2 获取指令数据到 L1 并附加预译

码

分支预测, link 和 count 寄存器
分支预测表, 8k 项 LBHT 和 16k 项 GBHT
转移运算指令减少分支
指令分组和线程优先级

2. 指令发射 ISU

按组发射指令
寄存器重命名技术
LTAG 和 STAG 管理 load/store 指令

3. 数据获取 LSU

两个对称的 LS 单元, LS0 和 LS1
每时钟周期执行一条 load 或 store
乱序执行, 顺序假象
写操作队列 SRQ, 读操作队列 LRQ
两级地址翻译, $EA[0:67] \rightarrow VA[0:67] \rightarrow RA[0:48]$
分段, 分页, 段快表, 页快表
段页转换一步到位 D-ERAT

L1 D-cache, 2 读端口 1 写端口, 多 bank 并行, 强写穿模式, EA 索引 RA 标签, 组预测技术, 多线程共享

4. 定点单元 FXU

两个对称的 FX 单元, FX0 和 FX1
GPR、ALU、CNT、BSU、DIV、MULT、MXU
常用的做得快
GPR 寄存器组, 4 读端口 4 写端口
乘法 4 周期, 流水线吞吐每周期一条
乘法的运算转移指令, 减少分支
专用的执行单元

5. 向量单元 VSU

六种指令类型, 双指令队列
只使用内部寄存器 VSR, 128 位宽
完全流水线
多种指令使用同一条流水线实现

缓存层次结构

减小的一级缓存和二级缓存
局部的三级缓存和共享的三级缓存
十三缓存状态协议
BSR, 栅栏同步寄存器
二级缓存
256KB, 128bit 缓存行, 8 路组相连
不同的工作频率

三级缓存

紧靠 L2 的局部三级缓存, 4MB
共享的三级缓存, 包含 8 个 4MB 的区域
使用 eDRAM 技术, 降低功耗
4MB 的局部 L3 被视为 victim cache

内存子系统

每个 POWER7 芯片包含两个集成的内存控制器
每个内存控制吃支持 4 通道
更多的信号资源、更多的缓存资源
CRC 纠错、DDR3、层次缓存结构
不同模块具有不同的工作频率
更优的调度算法, 乱序执行能力

IO 子系统

DMA 直接写 cache 技术

片上互联

每个芯片包含 8 个核心, 每个核心有 L2 和局部

L3

每个芯片还有两个内存控制器、两个 IO 控制器
一个多芯片互联的 SMP 接口, 可实现 32 芯片互

联

宽松的一致性协议
奇偶缓存行的轮换服务
不同工作频率的同步问题
内存控制器与片上互联总线相连
使用中央仲裁逻辑

片间互联

标准 SMP 接口实现 32 芯片互联, 共计 256 核心
一级互联: 四芯片互联, 共 32 核
二级互联: 八个一级节点, 共 256 核
重新构建一致性协议

群集互连

额外的 IO 芯片, cluster interconnect
四个节点内 SMP 接口, 实现一级互联
七个节点外 SMP 接口, 实现二级互联
与芯片配合, 实现三级互联
超级节点, 32×32 共 1024 核
支持最多 512 个超级节点互联, 共 512K 个核心

RAS 特性

Reliability, 可靠性

Availability, 有效性

Serviceability, 可用性

不再需要额外的恢复单元

Step 1. 暂停执行并修复

Step 2. 修复失败则冲刷流水线并恢复到错误之前的某个状态

Step 3. 一些其他必要的操作: self-test、reset、refresh

Step 4. 重新开始执行

多个模块具备纠错检错能力