

4 Microarchitecture and Performance Analysis of a SPARC-V9

Abstract

- SPARC-V9 的性能模型，包含详细的处理器模型和详细的内存模型
- SPARC64 V 微体系结构和性能分析

Introduction

- SPARC64 V 的工作频率为 1.3 GHz，采用 0.13um CMOS 工艺
- SPARC64 V 开发过程中用软件性能模拟器（性能模型）进行性能评估和性能研究；硬件设计开始之前开发一个性能模型，在硬件开发的同时不断提高其刚性，而且从逻辑模拟器上执行性能测试程序的验证结果来提高模型精度；在整个项目中使用单一的性能模型可以获得评估数据的一致性；最终性能模型的准确性相当高

Strategy

性能模型应该具备的稳定性：

- 包括详细的处理器模型和详细的内存系统模式
- 在早期阶段，通过性能模型的连续刚性来提高精度；一旦逻辑模拟器可用，使用模拟器来验证性能模型，迭代地提高其准确性

策略的实施过程：

- 硬件开发从基本设计阶段开始，进入详细设计阶段，然后进入验证阶段
- 在基本设计阶段开始时可以使用初始性能模型
- 在硬件开发期间，性能架构师进行性能研究，与硬件架构师合作，更新后续研究的性能模型
- 在详细设计阶段结束时，逻辑模拟器基于可用的实际硬件逻辑

Performance model

性能模型是基于实际的硬件逻辑创建的，是用 C 编写的跟踪驱动的软件模拟器；模型的要求是评估系统级性能和多处理器系统性能，改变模型结构的灵活性和高精度的能力；性能模型包括详细的处理器模型和详细的内存系统模型；性能评估的性能模型包括详细的处理器模型和相当粗略的存储器系统模型，例如延迟模型；性能模型的最终版本具有以下功能：

- 可以在寄存器传输级别对处理器进行详细建模；CPU 和内存之间的内存访问资源（包括缓冲区，队列和管道）与物理机中的内存访问资源相同。此外，可以建模用于访问，读取，写入和使内部高速缓存无效的高速缓存协议

- 在寄存器传输级别对处理器进行详细建模
- CPU 和内存之间的内存访问过程：包括缓冲区，队列和管道
- 建模用于访问，读取，写入和使内部高速缓存无效的高速缓存协议
- 连接高速缓存和存储器之间总线网络的详细建模

Evaluation and verification environment

- “逻辑模拟器”使用硬件逻辑创建，是内存系统和处理器内核设计的真实准确表示，可以用于验证内存系统和微体系结构的逻辑
- 开发的最后阶段是研究开发机器上的完整工作负载驱动性能；通过比较物理机器执行输出和性能模型输出来评估性能模型的准确性

Microarchitecture

- SPARC64 V 该处理器是基于 SPARC-V9 架构的 64 位微处理器，工作频率为 1.3 GHz
- 每个周期最多可以发出四条指令；64 条指令窗口和重命名寄存器；重命名寄存器中最多可保留 32 个浮点和 32 个整数结果；存在四种保留站：RSA，RSE，RSF 和 RSBR；每个 RSA 和 RSBR 由 10 个缓冲区条目组成；RSA 用于地址生成操作，RSBR 用于分支操作；每个 RSE 和 RSF 由两组缓冲区组成，每个缓冲区中有八个条目；有两组整数执行单元，两组浮点执行单元和两组地址生成单元；具有两组浮点乘加执行单元；一个周期中分派多达六条指令；特殊的执行单元直接连接到每个 RSE 和 RSF
- 处理器缓存层次结构由两个级别的非阻塞 Cache 组成：L1 指令 Cache 和数据 Cache；L1 Cache 都是 128-KB，2 路组相联，指令获取宽度为 32 个字节，从中可以获取 8 个指令；L2 Cache 是统一的，大小为 2 MB，4 路组相联
- 4 路分支历史表用于分支预测，具有 16K 条目；处理器内核有 16 个加载队列条目和 10 个存储队列条目，以减少阻塞内存访问操作

Non-blocking dual operand access

SPARC64 V 是深度流水线设计：

- 取指到译码需要 5 个时钟周期：1 个周期获得优先级，3 个周期从 L1 缓存中获取指令，1 个周期来验证获取的值
- SPARC64 V 从保留站推测性地发送指令，避免了深度流水线的缺点；此技术命名为“推测性调度”-如果指令的源数据尚未就绪，但在调度指令到达执行阶段之前就已准备就绪，则可以调度指令
- 每个执行单元和其他执行单元之间以及每个执行单元和操作数访问通道之间都有数据路径

- 如果由于请求导致高速缓存未命中而导致预测失败，则必须在执行管道的每个阶段取消所有具有读写后依赖性的指令

Pipelines using speculative dispatch and data forwarding

- L1 数据 Cache 被组织为八个存储单元，每个存储单元为四个字节；每个周期可以接受两个请求，除非存在 bank 冲突；如果冲突，则中止较低优先级请求的执行并在稍后的周期中重试；导致 L1 操作数高速缓存未命中的请求将保留在加载/存储队列中，直到其请求的行在加载到 L1 数据 Cache；当 Cache 的一行重填时，不影响其它行的数据读取

Two level cache hierarchy

- 考虑延迟以及成本，设计采取了 2 级片上缓存的设计

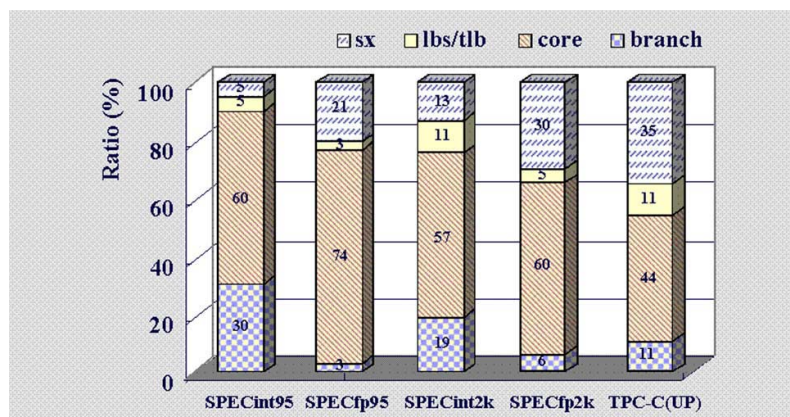
Performance analysis

Workload and trace generation

- SPEC CPU95 和 SPEC CPU2000 基准测试程序来研究处理器
- TPC-C 工作负载研究系统级行为

Benchmark characteristics

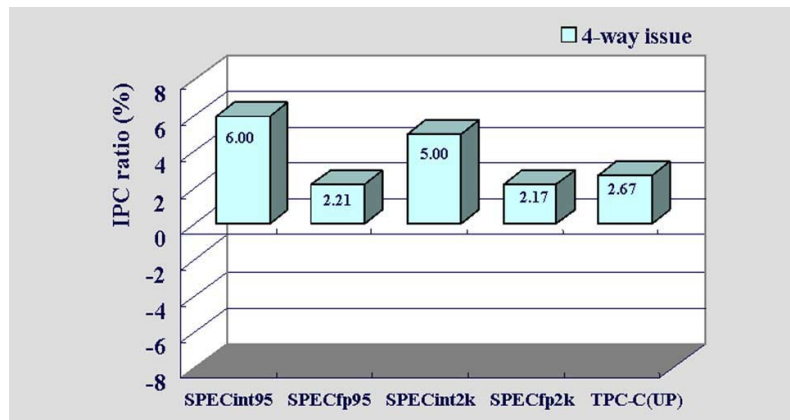
- sx 是由 L2 未命中引起的停顿；lbs/tlb 是由 L1 未命中和 TLB 未命中引起的停顿，core 是 I 单元和 E 单元中的执行时间；branch 是由分支预测失败引起的停顿



- SPECint95 将 30% 的执行时间用于分支未命中预测；百分比高于 SPECfp95 (3%)，意味着 SPECint95 包含比 SPECfp95 更多的分支指令以及更加难以预测的模式
- SPECfp95 在处理器内核中执行时花费了 74% 的执行时间，反映了浮点执行流水线的深度
- 由于 L2 未命中引起的停顿，TPC-C 有很大的惩罚，占执行时间的 35%；L2 缓存结构是提高 TPC-C 性能的关键

Data paths

Superscalar. SPARC64 V 的工程师经过对 SPEC95 和 SPEC2000 基准程序的并行性分析，认为 4 发射要优于 2 发射；



Branch prediction. 分支预测技术应该采取的是两级自适应预测技术，SPARC64 V 的工程师比较了 2 种实现方式：16K-entry, 4-way set associative, 2-cycle access table (“16k-4w.2t”) 以及 4K-entry, 2-way set associative, 1-cycle access table (“4k-2w.1t”)；前者在 SPEC95 和 SPEC2000 上的分支预测率略低，但是 TPC-C 要优于后者；SPARC64 V 的分支预测技术最终为 16K-entry, 4-way set associative

Level one Cache. 设计人员比较了 2 种 L1 Cache 的实现方式：32-KB, direct mapped, 3-cycle access L1 cache (“32k-1w.3c”) and a 128-KB, 2-way set associative, 4-cycle access L1 cache (“128k-2w.4c”)。后者的延迟要略高于前者，但是其 L1 数据 Cache 的未命中率要远远低于前者（两者的未命中率都很低）；设计人员认为 L1 Cache 越大，其命中率越高，一定范围内可以增加其大小，并且直接相联映射在处理大型工作负载时很容易引起 thrashing。

On-chip level-two cache. 设计人员比较了 3 种 L2 Cache 的实现方式：on-chip 2-MB, 4-way set associative cache 与 off-chip 8-MB, 2-way set associative cache 以及 off-chip 8-MB, direct mapped cache；在延迟以及命中率的权衡下，选择了 on-chip 2-MB, 4-way set associative cache 的方式

Hardware prefetching. on-chip L2 Cache 容量太小，设计人员研究了硬件预取算法，发现 FP 程序获得很好的预取优势

Control logic

Reservation stationse. 一个保留站情况下，SPARC64 V 每周期最多发送 2 条指令，两个保留站的 IPC 略低于一个保留站设计，通过权衡一个保留站和两个保留站的 IPC 以及相应的复杂度，SPARC64 V 最终选择了两个保留站设计

Performance model accuracy

性能模型的最终精度. SPECfp2000 为 3.9%, SPECint2000 为 4.2%; 表明性能模型设计思路以及实现是一个很好的策略;

Related work

处理器的性能评估模型对于新处理器设计而言十分重要, 架构师根据性能模型评估修改设计