

Sun Rock Microprocessor

Processor Design Methodology

- Set design goals and budgets (die size, transistor count, power, performance)
- Choose target workloads
- Iterate:
 - Simulate the proposed architecture in different configurations using the target workloads
 - Evaluate performance, power consumption and identify bottlenecks
 - Refine the design
- Two obvious dependencies:
 - Target workloads are well chosen
 - Simulator produces accurate results

Design Goals

- Good single-threaded performance
- High throughput
- High floating-point performance

Target Workloads

- Commercial workloads:
 - Online transaction processing (TPCC)
 - Server workloads (SPECjbb)
 - Proprietary in-house workloads
- Scientific workloads:
 - Massively-parallelized numerically-intensive workloads with huge memory footprints

Workload Properties

- Commercial workloads:
 - Large memory footprint
 - Unpredictable memory access patterns (high cache miss rate)
 - Complex conditional logic (high branch prediction miss rate)
 - Often parallelizable highly-threaded workloads
 - Throughput is the primary concern
- Scientific workloads:
 - Huge memory footprint (potentially TB)
 - Often predictable memory access patterns
 - Floating point intensive

Rock Architecture

- 4 "core clusters" each containing:
 - 4 superscalar cores supporting two threads
 - 2 floating point units
 - 2 x 32k 4-way set associative D cache
 - 32k 4-way set associative I cache
- 2M 8-way set associative L2 cache
- 8192 entry 8-way set associative TLB
 - 32 entry uDTLB per two cores
 - 64 entry uTLB per core cluster
- Novel hardware features:
 - Execute ahead
 - Simultaneous speculative threading

Rock vs Niagara Architecture

- Compared to Niagara:
 - More execution cores
 - Fewer supported threads per core (2 vs 8)
 - Far more complex cores
 - Smaller L2 cache (2M vs 4M)
 - Smaller L1 I cache per core (8k vs 16k)
 - Larger L1 D cache per core (16k vs 8k)

Rock Architecture

- Up to 32 simultaneous threads of execution (strands)
- Not much cache:
 - 16k D cache per core
 - 2M shared L2 (250+ cycles per miss)
- Small TLBs:
 - 16 8k uDTLB entries per core
 - 8192 TLB entries chip wide (500 cycles per miss)
- No (conventional) OoO execution

- This implies a massive problem with memory latency
- Where does the performance come from?

Execute Ahead

- A lengthy stall triggers a checkpoint of state
- Subsequent instructions are examined
 - If deps are available, instructions are executed
 - If deps are not available, instructions are "deferred"
- Once the stall completes, deferred instructions are replayed

- Consequences:
 - Independent instructions can be executed out of order
 - Multiple memory references can be outstanding
 - Caches are warmed once execution resumes

Simultaneous Speculative Threading

- Resources of two hardware threads assigned to a single software thread
- Replay of deferred instructions to proceed in parallel with execute ahead
- Consequences:
 - More aggressive instruction level parallelism
 - Reduced execute ahead failure due to resource exhaustion.

Resource Sharing

- Very limited on-chip resources
 - 2Mb cache between up to 32 threads
 - 8192 TLB entries (only 64Mb of mapped address space using 8k pages)
- Running a threaded workload will share resources effectively.
- Running many processes will not share resources effectively
 - Increased memory latency
 - Potential pathological poor behaviour (eg TLB thrashing)

Summary

- Focuses on commercial workload throughput
- Different design trade-offs compared to Niagara
- Fewer threads, but executes each thread faster
- Novel architectural features:
 - Execute ahead
 - Simultaneous speculative threading

S. Chaudry, R. Cypher, M. Ekman, M. Karlsson, A. Landin, S. Yip, H. Zeffner, M. Tremblay. Rock: A high-performance Sparc CMT processor. *IEEE Micro*. 29(2): 6-16, March 2009.