A 32-Way Multithreaded SPARC Processor

Politecnico di Milano sede di Como
Architetture per Sistemi Multimediali

Author: Aldo Puglisi 674641

Agenda

- Introduction
  - Complex Processor
  - Multithreaded Processor
- Niagara Approach
  - 32 Threaded Niagara CPU
  - SPARC Pipe
  - Thread Selection Policy
- Niagara Features
  - Caches
  - Coherence
  - Register File
  - Crossbar
- Conclusions
Introduction

- **Server Workload Characteristics**
  - High degree of *Thread Level Parallelism* (TLP)
  - Large numbers of parallel client requests
  - Large working sets with *poor locality* of reference leading to *high cache miss rates*
  - Significant *data sharing among threads* resulting in *coherence misses*
  - *Low Instruction Level Parallelism* (ILP) due to high cache miss rates, load-load dependencies, difficult to predict branches
  - Performance is bottlenecked by *stalls on memory access*

*Difficult to achieve higher performance* in these applications efficiently with *superscalar and superpipelined methods!*

Complex Processor

- *Processor optimized for ILP*
- *Single scalar processor*

*ILP processors mostly reduce CPU time, but memory stall time dominates overall performance!*
Multithreaded Processor

- Multiple threads share a pipeline, and overlapped executions result in higher performance for a multithreaded application.

Niagara Approach

- CPU with **32 threads exploits TLP**
  - Memory and Pipeline stall time hidden by multiple threads
  - Shared L2 cache allows efficient data sharing among threads
  - Shared pipelines allow reuse of resources by overlapping threads, thereby saving area
- **Memory** system designed for higher throughput
  - **High bandwidth** interface to L2 cache for L1 misses
  - Banked and **highly associative** L2 cache
  - **High bandwidth** interface to DRAM
- **Performance/Watt as a design metric**
32 Threaded Niagara CPU

- An implementation of **SPARC V9** architecture
- **Eight 64-bit 4 way multithreaded pipelines**
- 4 way banked **3MB secondary cache**
- **High bandwidth crossbar** interconnect for on chip communication
- **High bandwidth DRAM interface**

**SPARC Pipe**

- Single issue pipeline with **6 stages**
- **4 threads** (group) supported per pipe
- **Each thread has unique:**
  - Register state
  - Instruction buffers
  - Store buffers
- **The thread group shares:**
  - L1 caches
  - TLBs
  - Execution units
  - Pipeline resources
**Thread Selection Policy**

- Switch between available threads every cycle giving **priority to least recently executed thread**
- Threads become **unavailable due to**:
  - **Long latency ops**: loads, branch, mul, div.
  - **Pipeline stalls** such as cache misses, traps, and resource conflicts

**Caches**

- **L1 Caches**
  - Level 1 caches are **shared among 4 threads**
  - **I Cache**: 16kB, 4-way set associative, 32B line size
    - Fetches 2 successive instructions each cycle
  - **D Cache**: 8kB, 4-way set associative, 16B line size
    - Implements **Write-through** policy

- **L2 Cache**
  - Level 2 cache is **shared among 32 thread**
  - 4 Banks, each **12-way set associative**, total 3 MB
  - **64B blocksize**, 64B interleaved across banks
  - Implements **Write-back** policy (L2 to Memory)
### Cache Coherence

- **L2 cache is center of coherency**
- **Write-through**  **L1 has 2 states:**
  - valid/invalid (no Modified state)
- L1 caches kept **coherent by tracking L1 lines** in directories kept in L2
- When a memory location is modified, invalidates are issued by the L2 based on list in the directory
- Thread-to-thread communication is through L2

---

### Register File

- **3R/2W ports**, supports **8 register windows** and **4 threads**
- **SPARC register window** consists of 8 in, 8 local and 8 out registers
  - The **out registers** of a window are **addressed as the in registers** of the next sequential window, but are the same physical registers
- **Architectural set**
  - Complete set of registers
  - Implemented using compact six transistors SRAM cells
- **Working set**
  - Set of registers visible to a thread
  - Implemented using fast register file cells
  - Pipeline accesses occur in the working set
- A **transfer port** links the two register sets
Crossbar

- **High b/w interconnect** between SPARC pipes, cache banks and other shared resources
- **Non blocking design** – allows multiple transactions to be queued up from a source
  - A two-entry queue is available for each source-destination pair
- **Age order dispatch** to destinations ensures fair scheduling across all requestors

Traditional SMP: 32 Threads

Niagara: 32 Threads

Direct Crossbar Interconnect
Lower cost, better RAS, lower and uniform latency, greater and uniform bandwidth...

Conclusions

- Commercial Server Workloads exhibit **high degrees of Thread Level Parallelism**, but poor locality of reference
- **32 threads** are implemented on a **single CPU** to hide memory and pipeline stalls and maximize parallel memory accesses
- A **high bandwidth memory** subsystem with shared cache, services memory references
- **Sharing of resources** at all levels leads to a very area and power efficient design
- **Niagara is opensource**: Chip Design source code in Verilog language can be downloaded from opensparc.sunsource.net
References

- **Niagara: A 32-way multithreaded Sparc processor**
  IEEE MICRO Magazine, March-April 2005
  P. Kongetira, K. Aingaran, and K. Olukotun

- **Sun Microsystems web site**
  www.sun.com/processors

- **SPARC Architecture Manual Version 9**

- **SPARC Architecture Manual Version 8**

- **Improving Application Efficiency Through CMT**
  http://developers.sun.com/solaris/articles/chip_multi_thread.html