Sun Fireplane System Interconnect and POWER4 System Microarchitecture

Raj Parihar
parihar@ece.rochester.edu

Why System Interconnect?

- Multiprocessor Design Objective
  - Minimize Overall Cost
  - Maximize Overall Performance
  - Higher Reliability
  - Better Scalability

- All objectives are constrained by system Interconnects
  - "Communication is the main bottleneck in high performance computing"

References

- “The Sun Fireplane System Interconnect”
  - Alan Charlesworth

- “POWER4 System Microarchitecture”
  - J. M. Tendler et al.

Multiprocessor System: Sales

- Major share is still with small scale, 8-16 cores, multiprocessors

- Key trend
  - Sales of large and mid scale multiprocessors, with > 8 cores, have doubled ever year
Outline

- Motivation and Overview
- Sun Fireplane Interconnect Generation
- IBM System Interconnect Generation
- Multiprocessor System Architecture
  - Sun Fireplane System Architecture
  - POWER4 System Microarchitecture
- Cache Coherence and Memory Organization
- Large Scale Shared Memory Multiprocessors
- Comparison and Summary

Overview: Cache Coherence

- Mechanism to deal with multiple copy of data in a shared-memory environment

Two basic types of cache coherence protocol

- Broadcast (Snoopy) Coherency
  - All addresses are send everywhere
  - Snoop results are computed and combined
  - Lowest possible latency (i.e. Cache-to-Cache)
  - Suitable for low & mid scale, Hard to scale

- Point-to-point (Directory) Coherency
  - Address sent only to "interested" nodes
  - Directory keeps track of who is "interested"
  - Suitable for generic type of large networks
  - Provides high bandwidth and better scalability

Evolution: Sun System Interconnect

| System interconnect generation | 1.5GHz (T) | 1.5GHz (T) | 1.5Mhz Parallel architecture of 4-Pc (Re) | 2.5GHz Fireplane
|-------------------------------|-----------|-----------|------------------------------------------|-----------------|
| Processor                     | Sunny ISRC | Sunny ISRC | Sunny ISRC-3 | Sunny ISRC-4
| Memory processor              | 16        | 64        | 64            | 64         |
| Memory clock                  | 40 MHz    | 40-50 MHz  | 50-60 MHz     | 50 MHz
| Network clock                 | 10-15 MHz | 10-15 MHz  | 15-20 MHz     | 15 MHz
| Cache coherence mechanism     | Broadcast | Broadcast | Broadcast + point-to-point | Broadcast + point-to-point
| Cache coherence latency       | 250 ns    | 250 ns    | 250 ns        | 250 ns
| System size                   | 32 nodes  | 32 nodes  | 32 nodes      | 32 nodes

Fireplane Coherency Protocol

- Scalable Shared Memory (SSM) Protocol
  - Low latency in local memory accesses (< 24 nodes)
  - Single snooping (Broadcast based) coherence domain
  - High bandwidth across the network (> 24 nodes)
  - Directory based (point-to-point) coherence protocol
- Kind of Hybrid Solution: Best of both world
- Separate address and data interconnects
Fireplane: Address Bus Implementation

- 2-level bidirectional tree-structure of address repeaters
- AR2 is kind of ordering point
  - CPU0 (AR0) -> AR2 -> AR1(I/O1)

Within a Snooping Domain

- Address Request
- Broadcast Address
- Transfer Data
- Read from Memory
- Snoop
- If found, Cache-to-cache transfer

Fireplane: Cache Coherence

- Snoopy Domain
  - MEOSI Protocol
  - Dual Tags
- Snoop-result signals
  - Shared
  - Owned
  - Mapped
- Cache Tags
  - cM, cE, cO, cS, cl
- Dual Tags
  - ds, dO, dl
  - dt (Temporary)

Among the Snooping Domains

- Address Request
- Send Address to Home SSM
- Lock line check cohesency
- Send Response
- Broadcast address on home bus
- Snoop on home bus
- Read data from Memory
- Transfer data to home data agent
- Unlock Line
- Count Responses
- Transfer data to requester
- Move data across center plane
Cache to Cache Transfers

- When data is owned (modified) in a cache
- Inside a Snooping Domain
  - Owning device asserts a snoop result of OWNED
  - Cache sends data directly to requester and memory cycle is ignored
- Between Snooping Domains
  - Three way transfer to supply the data
    - Home SSM -> Owning SSM -> Requesting Data Agent

SYNC @ CS, UofR: Specs

- System Interconnect
  - SunFire V880
- Operating System
  - SunOS 5.8
- CPU Architecture
  - Eight 900 MHz UltraSPARC – III Processors
  - 64 KB L1 D Cache, 32 KB L1 I Cache / Processor
  - 8 MB unified (Data + INS) cache / Processor
  - 16 GB Main Memory

UltraSPARC T1 (Niagara) Processors

- Objective
  - To run as many as concurrent threads possible
  - Maximize the utilization of each core’s pipeline
- UltraSPARC Architecture 2005
  - SPARC V9 ISA (PSO and RMO Memory Model)
- Multicore and Multithreaded
  - 4, 6, 8 CPU cores; each with 4 concurrent threads
  - Optimized for Power: 72 W at 1.4 GHz
- Mainly for server applications
  - i.e. Web servers, smaller database applications

Rock Processors (Shipping: 2009)

- Higher per-thread performance
  - Oppose to Niagara: Maximize the # of threads
  - Greater SMP scalability than Niagara family
  - 1st production processor to support transactional memory
- SPARC V9, 64-bit ISA + VIS 3.0 SIMD MISA
- 16-cores/processors (4 cores/cluster)
  - Each core can run 2 threads simultaneously
  - Chip power consumption: 250 W at 2.3 GHz
- Target Application
  - Back-end database server
  - Floating point intensive HPC workloads
**Fireplane: Conclusion**

- Two level of cache-coherence protocol
- Small and mid scale multiprocessors
  - Use snooping protocol
- Large scale multiprocessors
  - Use directory based point-to-point protocol
  - Use Hybrid solution: Best of both world
- Cache-to-cache transfer to hide latency

---

**POWER4: Overview**

- System Architecture
  - Processor Microarchitecture
  - Interconnect Architecture
- According to IBM
  - Not only a Chip
  - Also refers to System Structure

---

**Evolution: IBM Microprocessor**

<table>
<thead>
<tr>
<th>Name</th>
<th>Year</th>
<th>Op Freq</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>RS/6000</td>
<td>1990</td>
<td>20-30 MHz</td>
<td>RISC based</td>
</tr>
<tr>
<td>POWER2</td>
<td>1993</td>
<td>55-71.5 MHz</td>
<td>8 ins/ cycle</td>
</tr>
<tr>
<td>PowerPC 601</td>
<td>1993</td>
<td>55-72 MHz</td>
<td>32,64-bit</td>
</tr>
<tr>
<td>RS64-II,III,IV</td>
<td>1997</td>
<td>125-750 MHz</td>
<td>Commercial Application</td>
</tr>
<tr>
<td>POWER3,-II</td>
<td>1998</td>
<td>200-450 MHz</td>
<td>2-FP, 3-Fixed function units</td>
</tr>
<tr>
<td>POWER4</td>
<td>2001</td>
<td>1.1-1.3 GHz</td>
<td>Multicore, On-chip L2 cache</td>
</tr>
<tr>
<td>POWER5</td>
<td>2005</td>
<td>1.1-1.9 GHz</td>
<td>SMT, On-die memory controller</td>
</tr>
<tr>
<td>POWER6</td>
<td>2007</td>
<td>3.5-4.7 GHz</td>
<td>65nm, in-order execution</td>
</tr>
<tr>
<td>POWER7</td>
<td>2010</td>
<td>4.0 GHz</td>
<td>Under development, 45 nm</td>
</tr>
</tbody>
</table>

---

**POWER6 (Latest)**

- A dual core design
  - Operating frequency: 3.5, 4.2, 4.7 and 5 GHz
  - 64KB L1 INS and Data cache; 4MB L2 shared cache
  - L3 cache: Off Die, 32 MB, Bus BW – 80 Gbps
  - Capable of Two-way SMT operation
  - Scalability: up to 64 physical processors
- POWER5 (Out-of-order) → POWER6 (In-Order)
- POWER6 based products
  - Blade servers: JS12 and JS22 blade modules: 6 cores
  - POWER 575: Up to 448 cores, up to 256 GB RAM/ frame
POWER4: High Level Features

- Extension of 64-bit PowerPC architecture
- 0.18 um – lithography and SOI technology
- “Speed demons” VS “Braniacs”
  - While UNIX based RS/6000 are of later kind
  - POWER4 clearly falls in former category
- Operating Frequency Range
  - 1.1 GHz – 1.3 GHz
- Up to 32-way SMP using POWER4

POWER4: Processor

- Two-way On-Chip SMP
- Core Microarchitecture
  - Speculative superscalar
  - Out-of-order execution
  - Issue: 8 INS/ Cycle
  - Completion: 5 INS/ Cycle
  - In-flight: > 200 INS
  - 8 Execution units
    - 2 FP Execution units
    - 2 LD/ST units
    - 3 Fix point Execution units
    - 1 BR Ex, 1 CR Ex unit

POWER4: Chip

POWER4: Die Photograph

POWER4: Core

- Branch Prediction Unit
- Instruction Fetch Unit
- Decode, Crack, Group
- Issue Queues
- LD/ ST Queue
- Execution units
  - FP Execution units
  - Fixed Point EX units
  - BR Execution unit
  - CR Execution unit
Conditional Branches

- High performance systems use multi-level branch predictors
- Two aspects of conditional branch prediction
  - Branch outcome: Taken or Not Taken
  - Branch Address: if Taken then to Where?
- What about unconditional branches?
  - Don’t even bother!
  - Compilers are smart enough to deal with them

Decode, Crack, & Group Formation

- INS are split to ensure the high frequency operation
  - Cracking: load with update (index) => Load + Add
- Group: To keep track of program order
  - Also used for imprecise exceptions (group states)
  - Group contains up to 5 IOPs (Internal Operations)
  - Slots are used to preserve the ordering
  - Only one group can be dispatched per cycle
- Dispatch: Into the issue queues (in-order)
- Issue: From issue queue to EX unit (out-of-order)
- Commit: Upon group completion (in-order)

Load/Store Unit Operation

- Crucial to ensure memory consistency in out-of-order machine
- SRQ and SDQ keeps the results till commit
  - Upon group completion SDQ is written to cache
- Hazards in LD/ST Queue (should be avoided)
  - Load hit Store
    - Younger load gets the data from SDQ if an older store is present to the same address
    - In case SDQ doesn’t have data loads are killed and reissued
  - Store hit Load
    - Store checks the LRQ; if younger load found, group is flushed
  - Load hit Load
    - If younger load gets the old data older load shouldn’t get new

Instruction Execution Pipeline

- Instruction flows in groups in program order
- If miss predicted
  - All dependent INS in pipeline are squashed
Storage Hierarchy: Organization

- L1 cache (Sort of directory protocol)
  - Low latency is achieved by low Associativity
- L2 cache (Kind of snoopy protocol)
  - High Associativity reduces the miss rate
- L3 cache
  - Directory is On-chip; Memory is external

<table>
<thead>
<tr>
<th>Component</th>
<th>Organization</th>
<th>Capacity per chip</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1 instruction cache</td>
<td>Direct-mapped, 128-byte line</td>
<td>128 KB (64 KB per processor)</td>
</tr>
<tr>
<td>L1 data cache</td>
<td>2-way, 128-byte line</td>
<td>64 KB (32 KB per processor)</td>
</tr>
<tr>
<td>L2</td>
<td>8-way, 128-byte line</td>
<td>1.5 MB</td>
</tr>
<tr>
<td>L3</td>
<td>8-way, 128-byte line managed as four 128-byte sectors</td>
<td>32 MB</td>
</tr>
<tr>
<td>Memory</td>
<td>—</td>
<td>0-16 GB</td>
</tr>
</tbody>
</table>

L1 Caches

- L1 Instruction Cache
  - Single-ported, Cache line: 32-byte
- L1 Data Cache
  - Triple-ported, Cache line: 32-byte
  - Two 8-byte read and one 8-byte write per cycle
  - Non-blocking data caches
- L1 caches are parity-protected
  - Error causes invalidation and reloading from L2
- L1 and L2 follow “cache-inclusion” property
- Two possible states in L1: Valid or Invalid

L2 Cache

- Shared unified L2 cache
  - Unified = Data + Instruction
  - Shared between 2 cores
  - 128-byte every 4 cycle
- Data is ECC protected
- Directory protocol with CPU
- Coherency processor
  - L2 and CPU data transfer
  - Fabric controller to CPU
  - L2 directory update
- Snoopy protocol with L3 fabric

MESI Protocol: L1-L2

- Enhanced Version of MESI Protocol (Seven states)
  - I (invalid state)
  - SL (Shared state, can be source to local requester)
  - S (Shared state)
  - M (Modified state)
  - Me (Exclusive state)
  - Mu (Unsolicited modified state)
  - T (Tagged state)

<table>
<thead>
<tr>
<th>L2 state</th>
<th>L1 data cache</th>
<th>State in other L2s</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>I</td>
<td>Any</td>
</tr>
<tr>
<td>S</td>
<td>I, S, S' , T</td>
<td></td>
</tr>
<tr>
<td>M, Me, or Mu</td>
<td>I, V</td>
<td>I, S, S'</td>
</tr>
<tr>
<td>T</td>
<td>I, V</td>
<td>I, S, S'</td>
</tr>
</tbody>
</table>
L3 Cache

- L3 Cache
  - L3 Controller (on-chip)
  - L3 Data array (off-chip)
  - 8-way Associativity
  - Block size: 512-byte

- Five coherency states
  - I (invalid state)
  - S (shared state)
  - T (tagged state)
  - Trem (remote tagged)
  - O (pre-fetch data state)

Memory Subsystem: Logical View

- Memory controller
  - Attached to L3 eDRAM
  - Synchronous wave pipeline
  - Bus speed
    - 1/3 of CPU speed
  - Protection
    - 2-bit ECC correction
  - Memory port
    - 4-byte bidirectional
    - Speed: 400 MHz

POWER4 based 8-way SMP

- Chip-to-chip fabric
  - 4, 16B bus/ chip
  - Total BW: 38.4 GBps

POWER4: Multi-chip Module

- MCM-MCM fabric
  - 2, 8B bus/ MCM
  - Total BW: 9.6 GBps
IBM Interconnects: Future Roadmap

- 2+ GHz clock rate for future processors
- Increased parallelism at all levels
- Incorporation of larger caches

Sun Fireplane VS POWER4

**Sun Fireplane**
- TSO, PSO and RMO
- Memory consistency model
- Network BW
  - System Clock: 150 MHz
  - 9.6 GBps/ Address bus
  - 172 GBps/ Max possible
- Peak Memory BW
  - 2.4 GBps

**POWER 4**
- Weakly ordered
- PowerPC consistency model
- Network BW
  - On/off bus: 600 MHz
  - MCM: 9.6 GBps/ MCM
  - Chip-to-chip: 38.4 GBps
  - L2 BW: 100 GBps
  - Memory Port (400 MHz)
  - BW: ~ 11 GBps

**Sun Fireplane**
- Two level of coherence protocol
  - Snoopy protocol
  - Directory based protocol
- Better Scalability
  - Possible to implement system with 8-96 UltraSPARC – III processors
  - Separate network for data and addresses

**POWER 4**
- On-chip two level of cache
  - L1 (sort of point-to-point)
  - L2 (kind of snoopy)
  - L3 is generally directory based
- Good for mid scale
  - Up to 32-way SMP implementation
- High Speed Design

Sun Fireplane VS POWER4

**Sun Fireplane**
- CPU: UltraSPARC-III
  - RISC based
  - In-order execution
  - Multiple independent pipeline
  - Clock: 900 MHz
- Network Latency
  - Addr: 15 sys cy (150 ns)
  - Data: Sys cy (14, 9, 5)
    - 93, 60, 33 ns

**POWER 4**
- CPU: POWER4
  - 64-bit PowerPC based
  - Out-of-order execution
  - Multicore: 2-cores/ chip
  - Clock: 1.1 – 1.3 GHz
  - 2-way SMP for software
- Network Latency
  - MCM: 10% greater than best case memory access latency
### UltraSPARC-III VS POWER4

<table>
<thead>
<tr>
<th>UltraSPARC-III</th>
<th>POWER 4</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CINT2000 base/ peak</strong></td>
<td><strong>CINT2000 base/ peak</strong></td>
</tr>
<tr>
<td>470/533</td>
<td>790/ 814</td>
</tr>
<tr>
<td><strong>CFP2000 base/ peak</strong></td>
<td><strong>CFP2000 base/ peak</strong></td>
</tr>
<tr>
<td>629/ 731</td>
<td>1098/ 1169</td>
</tr>
</tbody>
</table>

Source: IBM