Scalability Port
A Coherent Interface for Shared Memory Multiprocessors

Mani Azimi, Faye Briggs, Michel Ceklev*, Manoj Khare*, Akhilesh Kumar, Lily P. Looi

Enterprise Platforms Group
Intel Corporation

* Author made contribution while at Intel

Outline

• Design Goals
• Architecture
• Implementation
• Summary
Scalability Port (SP) Goals

• Designed for mid-range shared memory multiprocessors
  – Point-to-point interface with
    – Good scalability for mid-range systems and ability to extend to high-end systems
    – Enable cost-effective system architecture
  – Shared buses not cost-effective beyond limited number of processors
    – Limited speed due to signaling challenges
    – Proximity of devices on a bus causes thermal and mechanical challenges
    – Hierarchical approaches limit scalability

SP Goals

• Allow flexible system architecture
  – Enable cost-optimal small systems to scalable high-end systems
  – Enable system vendors with proprietary system interconnects and components to use Intel building blocks

• Modular architecture
  – Enhancements in orthogonal increments
  – Reduced verification complexity
SP Architecture Overview

- Layered Architecture
  - Physical, Link and Protocol Layers
- Parallel interface with cut-through routing to minimize latency
- Robust error detection and recovery
  - Link layer retry + end-to-end ECC
- Protocol tolerant of unordered network and resource constraints at targets
  - Allows flexibility in implementation choices

Physical Layer

- Simultaneous Bi-Directional (SBD) signaling
  - Pin efficient, full-duplex interface
  - 800 Million Transfers per Sec per Direction
- Clocking
  - Single clock source
  - Source synchronous interface
- 3.2 GB/Sec/Direction per SP
  - 42 bit wide interface with 32 data, 2 link layer control, 2 SSO and 6 error control bits
  - Additional signals for strobe, reference voltage, etc.
- 20”-25” trace on FR4 with up to 2 connectors
- Support for hot-plug
Physical Layer

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data Rate</strong></td>
<td>800Mb/sec/direction SBD</td>
</tr>
<tr>
<td><strong>Process</strong></td>
<td>0.18um CMOS</td>
</tr>
<tr>
<td><strong>Supply Voltage</strong></td>
<td>1.3V</td>
</tr>
<tr>
<td><strong>Voltage and Timing Margins</strong></td>
<td>&gt;10%</td>
</tr>
</tbody>
</table>

Link Layer

- Request and response virtual interconnect
  - Two virtual channels with VC identifier per flit (168 bits)
  - Network routing restricted to avoid deadlock
- Flow Control
  - Credit based flow control at flit granularity
- Reliable Transmission
  - Parity for error detection, link level retry for recovery
  - Sliding window protocol without sequence number
Link Layer

Flit Format

<table>
<thead>
<tr>
<th></th>
<th>32</th>
<th>4</th>
<th>2</th>
<th>2</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Payload</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ECC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Link Layer Control</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SSO</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parity</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Link layer control
  - Flit type, VC identifier, Credit, Packet delimiter, Ack

Protocol Layer

- Packetized interface with multiplexed request, response and data
  - Packet header allows up to 32 nodes, 50 bit address space and 64 outstanding transactions per node
- No reliance on ordered fabric
  - Can work with other unordered fabric
  - Improved performance due to available concurrency
- Event driven protocol without fixed timing dependencies
- Transaction retry for flow control and conflict resolution
Coherency Protocol

- Invalidation protocol with MESI states
- Allows separation between directory agent and home memory
  - Speculative memory accesses to hide latency
  - Building blocks designed for cost-sensitive systems can be reused in scalable systems
- Optimization suitable for commercial workloads and for accesses from I/O devices
  - Cache to cache transfers without memory update
  - Cache line ownership without data fetch
  - Coherent data fetch without altering cache states

Coherency Example

Read to a Remote Clean Line

1. Requesting Node
2a. Snoop Filter Lookup
3a. Snoop Response
3c. Snoop Filter Update
2b. Speculative Read
3b. Read Confirm
4. Data Response

Home Node

Coherency Switch

SNC = Syst Node Ctrl
IOH = IO Hub
SPS = SP Switch
SF = Snoop Filter
Coherency Example

Read to a Dirty Line

Coherency Protocol

- Distributed conflict resolution
  - Directory controller determines the access order between concurrent accesses
    - Retry response on conflict at directory agent
  - Caching agents resolve races due to network reordering
    - Addresses of outgoing requests and incoming probes are compared
    - Incoming probes blocked on conflict and released on completion or retry of outgoing requests
Conflict Example

Coherency Protocol

- Deadlock Prevention
  - Separate request and response virtual network
  - Request retry on resource unavailability
    - Not applicable to forwarded requests
    - Nodes provide buffering for blocked forwarded requests
- Starvation Avoidance
  - Source detects starving transactions
    - Starving transactions completed before accepting new transactions
  - Target must be fair to all sources
    - Resource reservation per source or prioritization of sources with rejected requests
Other Protocol Features

- Message based interrupt delivery
- TLB consistency protocol
- Support for pipelined writes to MMIO
- Error Handling
  - End-to-end data protection with ECC
  - Data poisoning for error containment
  - Error isolation at component and link level
  - Response status to indicate transaction failure

SP Implementation

- E8870/E8870SP chipset for Itanium®2 processor
  - 1-4P without coherency switch
  - 8P and above using coherency switch

Two Node Direct Connect

Multi-Node with SPS

SNC = Scalable Node Ctr
SIOH = Server I/O Hub
### Comparison

<table>
<thead>
<tr>
<th></th>
<th>Point-to-point (E8870SP System)</th>
<th>Hierarchical bus (Sun Fire 6800)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Processors</strong></td>
<td>16 Itanium®²</td>
<td>24 UltraSPARC-III</td>
</tr>
<tr>
<td><strong>Max Data BW</strong></td>
<td>25.6 GB/sec</td>
<td>9.6 GB/sec</td>
</tr>
<tr>
<td><strong>Max Snoop Rate</strong></td>
<td>532 million/sec with 2 switches</td>
<td>150 million/sec</td>
</tr>
<tr>
<td><strong>Complexity</strong></td>
<td>4 SNC, 2 IOH and 2 switch components, 2 SP links per CPU/memory board</td>
<td>&gt;32 Data Switches, &gt;8 Address Repeaters, 288 bit data + address/control per CPU/memory board</td>
</tr>
</tbody>
</table>

*Other names and brands may be claimed as the property of others.*

---

### SP Summary

- **Scalable System Interface**
  - Point-to-point interface with simultaneous bi-directional signaling
  - Modular layered architecture
  - Interface for scalable building blocks
  - Enhanced RAS support
- **E8870/E8870SP Chipset Implementation**
  - Cost-effective and scalable architecture
  - Building block with other architectures