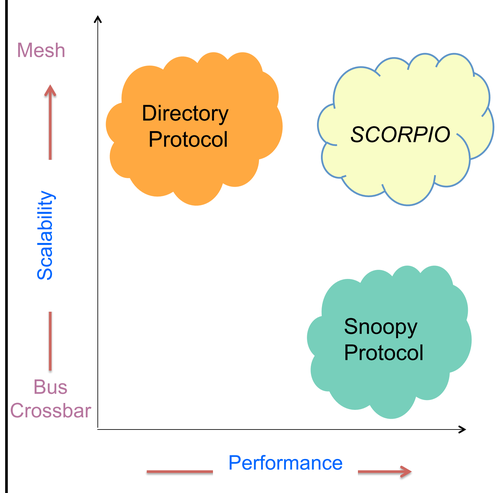
SCORPIO

Scalable COherent Research Processor with Interconnect Ordering

Bhavya Daya, Li-Shiuan Peh

Objectives



1. Transition snoopy protocols to practical globally ordered mesh network
   → **Scalable**

2. Integration with current bus-based single-core processors with minimal design effort
   → **Fast design time, compatible with existing IP**

Contributions:

1. Globally ordered on-chip mesh network that supports snoopy coherence
2. Separate "fast" contention-free network for ordering and request injection notification
3. AMBA interface compatibility between Core-L2, L2-NIC, NIC-Memory Controller

On-Chip Networks

Three Virtual Networks

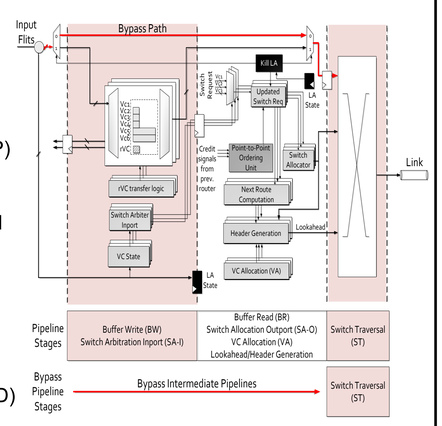
- Globally Ordered Request (GO-REQ)
- Point-to-point Ordered Request (P2P-REQ)
- Unordered Response (UO-RESP)

Virtual Bypassing

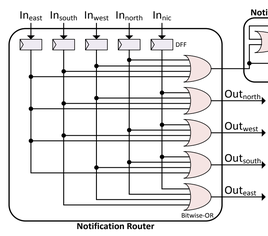
- Reduce per router cycle time to 1 cycle!

Reserved VC (rVC) for Deadlock Avoidance

- Reserved for flit that has highest priority
  src ID==ESID(expected src ID)



Main Network



Latency-Bounded Notification Network

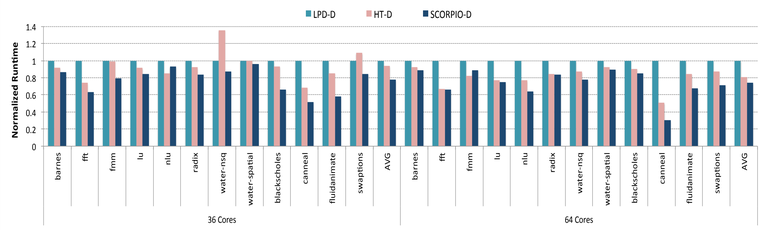
One-hot bit-vector: contention-free merging!

- Sent from source as notification to all cores to expect an ordered request from this source on main network
- Bitwise OR merging of notifications requesting same output port

One cycle per hop: latency-bounded, no contention

- Maximum number of cycles for notification broadcast is 2N-1, where N = sqrt(number of cores)

Architectural Evaluations



LPD-D(Limited-Pointer Directory) : distributed directory with pointers to a few sharers per line.

HT-D(HyperTransport): no sharer information stored at distributed directory (used as ordering point). Will broadcast to all nodes.

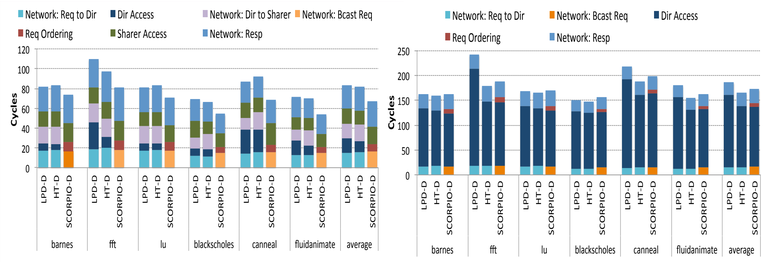
SCORPIO-D(SCORPIO): directory only maintains if the owner is on-chip or not to determine if memory controller should respond.

→Isolate the effects of indirection and directory storage (all conditions equal)

24% better performance over LPD-D and 13% over HT-D overall

Normalized Runtime

Breakdown of Request Delivery Latency

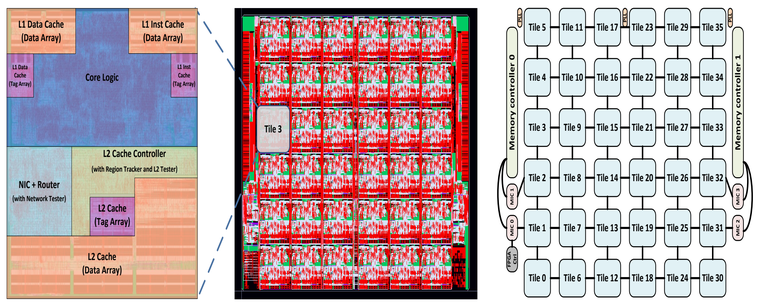


Served by other caches (90% cases)

Served by directory (10% cases)

36-Core Chip Prototype

Fabricated in IBM 45nm SOI Process, 11x13 mm², 600 Million Transistors



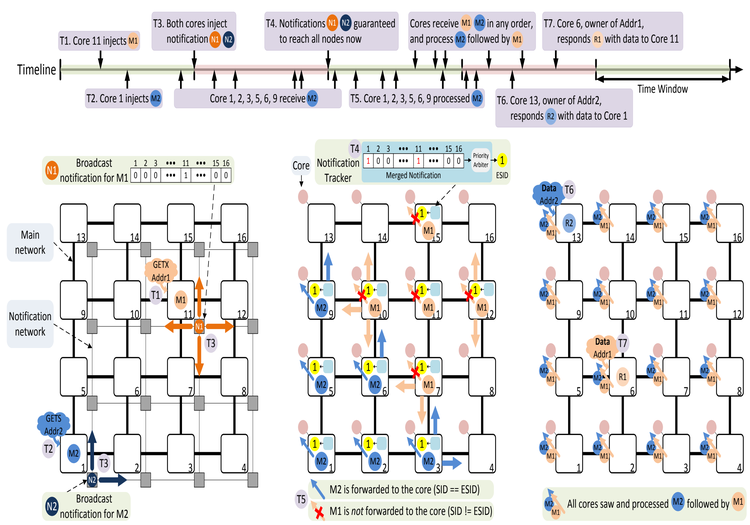
Core	Dual issue, in-order, 10-stage pipeline
ISA	32-bit Power Architecture
L1 Cache	Private split 4-way set associative write-through 16KB I/D
L2 Cache	Private inclusive 4-way set associative 128 KB
Line Size	32 B
Coherence Protocol	MOSI (O: forward state)
Directory Cache	128 KB (1 owner, 1 dirty bit)
Snoop Filter	Region tracker (4KB regions)

6x6 Mesh Network

- 137 bit channel-width
- XY routing, cut-through
- Multicast, lookahead bypassing
- 3-stage router (1-stage with bypassing)

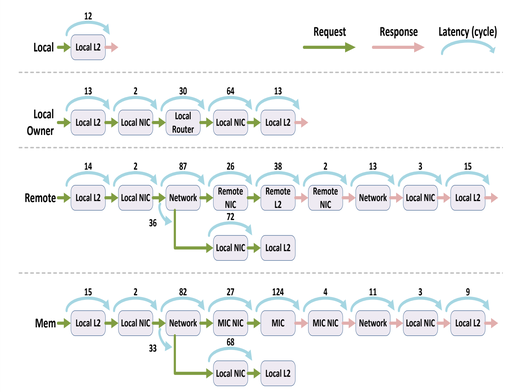
1 GHz (833 MHz post-layout)
28.8 Watts

Walkthrough Example



1. Inject & broadcast L2 misses (M1, M2) on main network: *1-flit pkt with src ID on GO-REQ VNET*
2. Inject & broadcast notifications (N1, N2) on notification network: *send at start of next time window*
3. Notifications reach all cores within 8 cycles: *each core determines message order and sets ESID*
4. Messages M1, M2 arrive, in different orders at different cores: *M1 is buffered until M2 is received*

RTL Evaluations



Barnes Benchmark

In-Network Filtering

⇨ Reduce network traffic and messages at endpoints

⇨ Reduce "network" and "Local NIC" latency

L2 Service Time

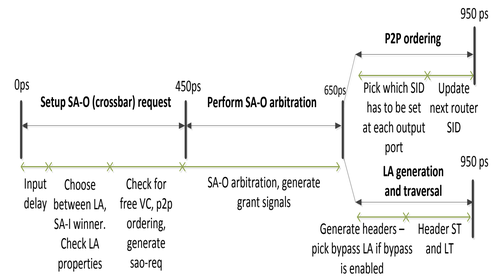
Network Critical Path

Router critical path timing = 950ps

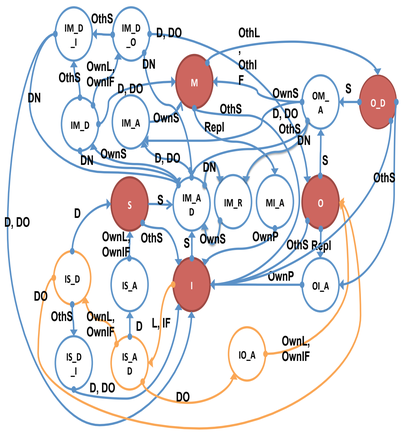
VCs affect SA-I stage

SA-I is off the critical path

⇨ negligible impact on operating frequency as increase number of VCs (required for higher core counts)



In-Network Snoop Coherence



Processor Requests		L Load
		S Store
		IF IFetch
		Repl L2 Replacement
Network Requests	GETS	OwnL Own Load
		OthL Other Load
		OwnIF Own IFetch
		OthIF Other IFetch
	GETX	OwnS Own Store
		OthS Other Store
	PUTX	OwnP Own Writeback
		OthP Other Writeback
Network Data Responses		D Data Response
		DO Data Response – Owner
		DN Data Response - Null

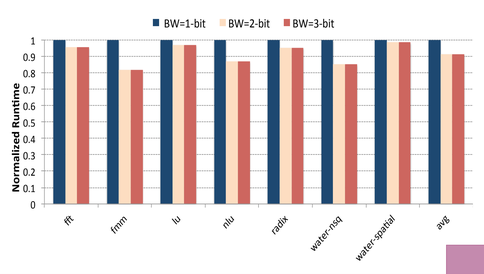
Broadcast MOSI Snoopy Protocol

- O_D stable state indicates dirty line in shared state
- Cacheline owner on chip can be in M, O, or O_D state
- Clean and dirty writeback distinguished for network operation
- Transient states, indicated by 'A' and 'D', identify whether waiting for own request, data response, or both
- Data forwarding tracking to not block incoming network request queue

Writeback race handling:

- Writeback of cacheline
- Another core has a store miss, injects a GETX.
- GETX retry mechanism – send null data if in middle of writeback
- Due to separate request and data VNETs

Network Design Exploration



Goal: Determine if a wider notification network is necessary

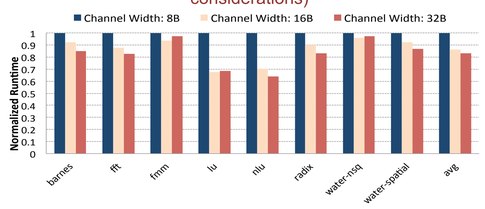
⇨ negligible impact of increasing the notification bit-width because traffic bursts rarely occur.

⇨ Low overhead & scales well

Notification Network Bit-Width

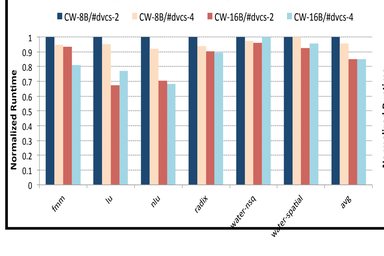
Network Parameter Sweep

Goal: Optimize network parameters (area, timing considerations)

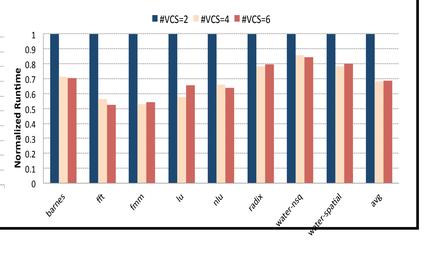


16B channel width

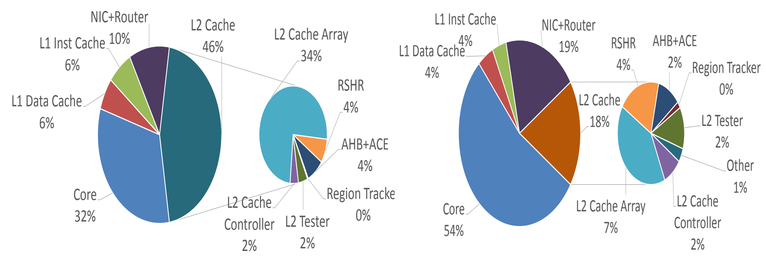
2VCs(3 buffers each) : UO-RESP Vnet



4VCs (1 buffer each) : GO-REQ Vnet



Scalability, Area, Power

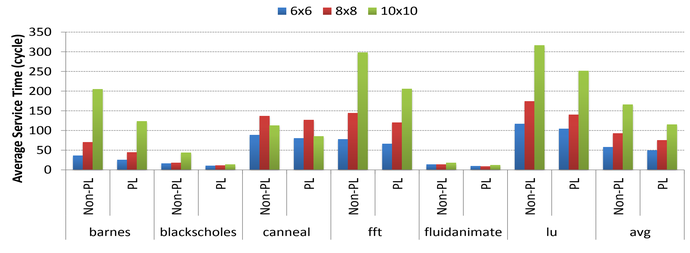


Area

10% of Tile Area

Power

19% of Power Consumption



Average round-trip latency:

Time between a load/store request at the core interface to when the response returns

More VCs for larger network sizes:

16 VCs for 8x8 network

50 VCs for 10x10 network

Effects of Pipelining on the Scalability