Micronetwork-based Processor Microarchitectures

Steve Keckler

Department of Computer Sciences
The University of Texas at Austin
Motivations

- Limitations of monolithic processors and memories
  - Wires, design complexity, port limits

- Goal: scalable processor and memories
  - Design complexity scalability
  - Ability for more resources to work together

- Approach
  - Recast as distributed systems
  - Tiles connected via a collection of networks

- Micronet = microarchitectural network
  - “Just” a network tightly integrated into processor/memory
TRIPS Tiled and Networked Processor

- **SOC-like design style**
  - Individually designed tiles
  - 3-8 mm² each
  - 170M transistors

- **Networks**
  - Memory
  - Operands
  - Control

- **Networks enable**
  - Distributed and scalable design
  - Fast design cycle
  - Configurability
Outline

- TRIPS architecture overview

- Integrated processor network (OPN)
  - Replaces bypass and processor/cache bus
  - Reflection on design

- Memory network (OCN)
  - NUCA cache
  - System interconnect
  - Extendable to multiple chips (C2C)
TRIPS Prototype Chip

- 2 TRIPS Processors
  - 16 FPUs each
  - Explicit Data Graph Execution (EDGE)

- NUCA L2 Cache
  - 1 MB, 16 banks

- On-Chip Network (OCN)
  - 2D mesh network
  - Replaces on-chip bus

- Controllers
  - 2 DDR SDRAM controllers
  - 2 DMA controllers
  - External bus controller
  - C2C network controller

- Fabricated in 130nm ASIC
TRIPS Tile-level Microarchitecture

TRIPS Tiles

G: Processor control - TLB w/ variable size pages, dispatch, next block predict, commit
R: Register file - 32 registers x 4 threads, register forwarding
I: Instruction cache - 16KB storage per tile
D: Data cache - 8KB per tile, 256-entry load/store queue, TLB
E: Execution unit - Int/FP ALUs, 64 reservation stations
M: Memory - 64KB, configurable as L2 cache or scratchpad
N: OCN network interface - router, translation tables

DMA: Direct memory access controller
SDC: DDR SDRAM controller
EBC: External bus controller - interface to external PowerPC
C2C: Chip-to-chip network controller - 4 links to XY neighbors
**TRIPS Execution Model**

**Program CFG**
- Block A
  - Basic block
- Block B

**Architecture of a Block**
- Registers
  - r2
  - r3
- Memory
  - i1
  - i2
  - i3
  - i4
  - i5
  - i6
  - r5
- PC

**Distributed Execution**
- PC
- i6
- i5
- i4
- i3
- i2
- i1
- r2
- r3
- r5

**Coarse-grained program sequencing using blocks**
- Dataflow execution within one block, instructions encode communication
- Spatial distribution exposed to the compiler
TRIPS Processor Tiles and Networks

- **Control Networks**
  - Instruction fetch/dispatch (GDN)
  - Completion/commit/flush network (GCN)

- **Operand network**
  - Bypass network among ALUs
  - Register file inputs
  - Load/store access

- **Memory network (OCN)**
  - I/D cache misses to L2/memory
  - Read/write to remote memory
TRIPS Operand Network (OPN)

- **Topology**
  - 5x5 mesh network
  - 1 cycle per hop
  - 140 bit channels

- **Routing**
  - Y-X dimension order
  - 4 entry input FIFOs
  - Destination from instruction targets

- **Flow control**
  - 1 physical channel (no VCs)
  - On-off link control
  - Deadlock free as storage at target is pre-allocated

- **Lightweight and tightly coupled to processor core**
  - Takes place of bypass bus
  - Bisection BW 80GB/sec at 500MHz
Obligatory Router Diagram
Processor Architecture Influences NW

- Latency critical to performance (1 cycle per hop)
  - Simple routers, no VCs

- Deadlock avoidance
  - Easy because destination buffers pre-allocated

- Y-X routing
  - Avoid bottlenecks from RTs and to DTs

- 2-flit messages (sort of)
  - Control header leads data payload by 1 cycle
  - 110 bit payload (64-bit datum plus 40-bit address)
  - But - separate control/data wires

- Speculative header injection
  - Can be canceled by null data flit

- Network selectively flushed when block flushed
OPN Integration to Processor Core

- 2 parallel networks
  - Control (30 bits) for routing and wakeup
  - Data (110 bits)
    - Includes 40 bit address and 64 bit operand for store
    - Bypassed directly into ALU at target

- Speculative injection of control packet
  - For early wakeup at target
  - May require cancel on next cycle
  - Control/data interleaved across operand messages

- Block flush includes flushing block's state in router
Design Experience

Area - remember ASIC standard cell design
- 1 OPN router = 0.25mm² in 130nm
  - A little larger than a 64-bit integer multiplier
- 25 OPN routers = 14% of processor area
- Area breakdown of OPN router

<table>
<thead>
<tr>
<th>Component</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIFOs</td>
<td>75%</td>
</tr>
<tr>
<td>Crossbar</td>
<td>20%</td>
</tr>
<tr>
<td>Arbitration/routing logic</td>
<td>5%</td>
</tr>
</tbody>
</table>

Static timing estimates - nominal corner

<table>
<thead>
<tr>
<th>Component</th>
<th>Time (ps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIFO read</td>
<td>386ps</td>
</tr>
<tr>
<td>Arbitration</td>
<td>253ps</td>
</tr>
<tr>
<td>Crossbar</td>
<td>187ps</td>
</tr>
<tr>
<td>FIFO muxing</td>
<td>473ps</td>
</tr>
<tr>
<td>Latch setup, clock uncertainty</td>
<td>367ps</td>
</tr>
<tr>
<td>Total</td>
<td>1.7ns</td>
</tr>
</tbody>
</table>
Performance Observations

- Average number of hops in network = 2
  - Compiler controls instruction placement

- Average network latency = ~2
  - But can have high variance
  - Small number of critical messages can degrade performance

- Load varies across network node and across applications
  - Depends on concurrency profile

- Standard NW loads are not representative
Highly Non-uniform Injection

- Uniform in RT/DT due to interleaving or registers and data cache
- High injection rates in ETs near registers and data
  - Injection rate reflects instruction placement

![Graph showing non-uniform injection rates](image)
# Network Protocol Overheads

Columns show percentage of critical path (methodology adapted from Fields, et al.)

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>IFetch</th>
<th>OPN Hops</th>
<th>OPN Contention</th>
<th>Fanout</th>
<th>Block Complete</th>
<th>Block Commit</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>a2time</td>
<td>4.9%</td>
<td>13.6%</td>
<td>6.5%</td>
<td>9.5%</td>
<td>2.1%</td>
<td>4%</td>
<td>59.4%</td>
</tr>
<tr>
<td>bezier</td>
<td>2.6%</td>
<td>16.9%</td>
<td>5.2%</td>
<td>12.5%</td>
<td>0.2%</td>
<td>2.6%</td>
<td>59.9%</td>
</tr>
<tr>
<td>dct8x8</td>
<td>5.4%</td>
<td>30.6%</td>
<td>10.0%</td>
<td>3.8%</td>
<td>3.2%</td>
<td>2.1%</td>
<td>44.9%</td>
</tr>
<tr>
<td>matrix</td>
<td>8.0%</td>
<td>20.3%</td>
<td>17.2%</td>
<td>4.9%</td>
<td>4.1%</td>
<td>3.2%</td>
<td>42.4%</td>
</tr>
<tr>
<td>sha</td>
<td>0.6%</td>
<td>17.9%</td>
<td>6.3%</td>
<td>11.7%</td>
<td>0.1%</td>
<td>0.7%</td>
<td>62.7%</td>
</tr>
<tr>
<td>vadd</td>
<td>7.4%</td>
<td>17.7%</td>
<td>13.8%</td>
<td>5.6%</td>
<td>6.0%</td>
<td>7.5%</td>
<td>42.1%</td>
</tr>
<tr>
<td>mcf</td>
<td>1.6%</td>
<td>28.5%</td>
<td>6.1%</td>
<td>0.0%</td>
<td>0.1%</td>
<td>0.2%</td>
<td>63.5%</td>
</tr>
<tr>
<td>twolf</td>
<td>3.0%</td>
<td>18.1%</td>
<td>3.1%</td>
<td>0.8%</td>
<td>0.3%</td>
<td>0.8%</td>
<td>74.0%</td>
</tr>
</tbody>
</table>

- mcf and twolf compiled but not hand-optimized
Operand Network Enhancements

- **Operand multicast**
  - Instructions have limited number of targets (2, 3, or 4)
  - Network injects one copy per cycle
  - Tree of instructions required for high-fanout operands
  - Optimization we are studying
    - Instruction specifies bit-mask of targets
    - Operand network replicates copies

- **Bulk operand movement (i.e. L/S multiple)**
  - Current architecture transmits one operand per message
    - Streaming data into arithmetic array is difficult
  - Optimizations we are studying
    - Single load request fetches multiple operands into successive reservation stations
    - Saves headers and streamlines return of data

- **Replicating network to provide more link BW**
TRIPS Memory Network (OCN)

- **Topology**
  - 4x10 mesh network
  - 1 cycle per hop
  - 128-bit x 2 links

- **Routing**
  - Y-X dimension order
  - 2 entry input FIFOs
  - Destination memory address

- **Flow control**
  - 1-5 128-bit flits/msg
  - 4 VCs for 4 priorities
  - Wormhole routed
  - Credit-based flow control
  - Pipelined credit return

- **Replaces memory bus**
  - Bisection BW 64 GB/sec at 500MHz
Non-Uniform L2 Cache (NUCA)

- Exploit physical locality in cached data

- **N-tile**
  - Resolves address to coordinate
    - M-tile or SDC if on this chip
    - C2C controller if on another chip
  - Injects ld/st request on VC0
  - 1-byte up to full cache line

- **M-tile** performs lookup and returns response on VC3
  - 64KB per M-tile

- **Hop count** depends on destination
  - Static NUCA
  - Total Unloaded latency 7-22 cycles
Network Based Memory Configuration

N-tile mechanisms

- Split mode to adjust cache line address interleaving
  1. Interleaved across 16 tiles
  2. Interleaved across 8 tiles (split cache)

- 16-entry translation table
  - Indexed w/ 4 bits of PA
  - Produces X/Y coordinate of MT

- Convert cache banks to scratchpad
  - Remap address range from one MT to another
  - Create new TLB entry to map new physical region into VA space
**OCN Design Observations**

- **Bandwidth and Latency**
  - Peak injection BW: 74GB/sec, but load is much less
  - Unloaded hit latency: 7-22 cycles

- **Area**
  - FIFO buffers: 75% of router area
  - OCN routers/wires: 32% of L2 area, 10% of die area
  - Opportunity to economize design

- **Timing**
  - Control was the critical path for the router
  - Timing path: 1.5ns (nominal case)
    - 400ps: VC arbitration
    - 427ps: crossbar arbitration
    - 393ps: FIFO control
    - 247ps: latch setup, skew
Chip-to-Chip Network

- On-chip 4-port router for C2C mesh network
- 32-bit x 2 links at 1/2 core clock speed
- Protocol is direct extension of OCN
- Global memory addressing identifies target
Summary

- Fast dynamic networks enable:
  - Distributed processor and memory architectures
  - Configurability

- Design experience
  - Networks were easy to build and verify
  - Larger than expected, but optimization possible

- Future challenges
  - Better traffic management w/out increasing latency
  - Drive router power down to beat other network topologies
  - How many different NWs and types of NWs are needed
    - TRIPS has 3 routed data networks
    - Multiple control networks
  - Does it make sense to design for worst case?
    - Better workloads for network analysis
  - Network interface primitives to the programmer
Acknowledgements

- Co-PIs: Doug Burger and Kathryn McKinley

- TRIPS Hardware Team
  - Raj Desikan, Saurabh Drolia, Madhu Sibi Govindan, Divya Gulati, Paul Gratz, Heather Hanson, Changkyu Kim, Haiming Liu, Ramdas Nagarajan, Nitya Ranganathan, Karu Sankaralingam, Simha Sethumadhavan, Premkishore Shivakumar

- TRIPS Software Team
  - Kathryn McKinley, Jim Burrill, Katie Coons, Mark Gebhart, Sundeep Kushwaha, Bert Maher, Nick Nethercote, Sadia Sharif, Aaron Smith, Bill Yoder

- IBM Microelectronics Austin ASIC Group

- TRIPS Sponsors
  - DARPA Polymorphous Computing Architectures
  - Air Force Research Laboratories
  - National Science Foundation
  - IBM, Intel, Sun Microsystems