IMPLEMENTING LOW-DIAMETER ON-CHIP NETWORKS FOR MANYCORE PROCESSORS USING A TILED PHYSICAL DESIGN METHODOLOGY

Yanghui Ou, Shady Agwa, Christopher Batten

Computer Systems Laboratory
Cornell University
REAL MANYCORE IMPLEMENTATIONS USE SIMPLE MESH OCNs

KiloCore, 1000 cores, 32x32 mesh
UC Davis

Epiphany-V, 1024 cores, 32x32 mesh
Adapteva, Inc

Celerity, 496 cores, 16x31 mesh
University of Washington, University of Michigan, Cornell University, UC San Diego
PLENTY OF NOVEL OCN TOPOLOGIES PROPOSED IN THE ACADEMIC AREA

- **Flattened Butterfly**
  - Kim+, MICRO’07

- **Concentrated Mesh, Fat-Tree**
  - Balfour+, ICS’06

- **Multi-drop Express Channels**
  - Grot+, HPCA’06/ISCA’11

- **Clos Network**
  - Kao+, TCAS’11

- **Slim NoC**
  - Besta+, ASPLOS’18

- **SMART NoC**
  - Chen+, HPCA’13

- **Asymmetric High-Radix**
  - Abeyratne+, HPCA’13
**GAP BETWEEN PRINCIPLE AND PRACTICE**

- Why do manycore processor implementations with 500-1000 cores continue to use simple high-diameter on-chip networks?
  - Manycores require simple, low-area routers
  - Manycores use standard-cell-based design
  - Manycores use a **tiled physical design methodology** with three key constraints:
    1. Design is based on tiling a **homogeneous** hard macro across the chip
    2. All chip top-level routing between hard macros must use **short wires** to neighboring macros
    3. Timing closure for the hard macro must imply timing closure at the chip level

**Hard Macros in Celerity**
Implementing Low-Diameter OCN for Manycore Processors Using A Tiled Physical Design Methodology

Motivation

Manycore OCN Topologies

Manycore OCN Analytical Modeling

Manycore OCN Physical Design

PyOCN Framework
**TARGET CHIP: 16 x 16 MANYCORE**

- 16x16 manycore at 1GHz using 14nm technology
- 3mm x 3mm, 185µm x 185µm per core
- Per-core area roughly corresponds to an in-order RV32IMAF processor with 4KB data cache and 4KB instruction cache

<table>
<thead>
<tr>
<th>Component</th>
<th>Area (µm²)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RV32IMAF-IO</td>
<td>15983</td>
</tr>
<tr>
<td>4KB data cache</td>
<td>9407</td>
</tr>
<tr>
<td>4KB inst. cache</td>
<td>9347</td>
</tr>
<tr>
<td>Total</td>
<td>34737</td>
</tr>
</tbody>
</table>

| Per-core Area | 24,250µm² | 103,500µm² |
| Process       | 16nm      | 16nm       |
| Frequency     | ~1GHz     | 500MHz     |
| ISA           | RV32IM    | RV64G      |
| Issue Width   | Single    | Dual       |
| L1 Memory     | 8KB       | 64KB       |
RUCHE CHANNELS TO REDUCE THE OCN DIAMETER

- Directly skips one or more routers
- Reduces network diameter
- Increases the number of bisection channels
- Increases router radix

Concurrently proposed with T. Jung et al, *Ruche Networks: Wire-Maximal No-Fuss NoCs*
CONCENTRATION TO REDUCE THE OCN DIAMETER

- Groups multiple cores together to share one router
- Reduces network diameter
- Reduces the number of routers
- Reduces the number of bisection channels
- Increases router radix

16x16 manycore

No concentration

Concentration factor of four

Concentration factor of eight
**TILED PHYSICAL DESIGN – NO RUCHE CHANNELS**

- Only has near channel in both dimensions
- Pins are aligned to ensure short global routing

mesh-c1r0 hard macro in 1D

mesh-c1r0 tiled physical design
Near channel, far channel, and one feedthrough channel in one dimension

Short cross-over routing between feedthrough channel and far channel
Near channel, far channel, and two feedthrough channels in one dimension

Short cross-over routing between feedthrough channels and far channel
Tiled Physical Design – Folded Torus

- Only far channel and feedthrough channel in one dimension
- Short cross-over routing between feedthrough channels and far channel
- Short wrap-around routing at the edge

Motivation

- Topologies
- Analytical Modeling
- Physical Design
- PyOCN Framework

torus-c1r0

torus-c4r0

torus-c8r0

Feedthrough Channel
Far Channel

torus-c1r0 hard macro in 1D

Torus-c4r0 tiled physical design
Implementing Low-Diameter OCN for Manycore Processors Using A Tiled Physical Design Methodology

Motivation

Manycore OCN Topologies

**Manycore OCN Analytical Modeling**

Manycore OCN Physical Design

PyOCN Framework
**ANALYTICAL MODELING METHODOLOGY**

- Model the latency, area, and bandwidth analytically before doing physical design to narrow down our focus.
- Router area model and channel latency model are constructed based on physical results and floorplans.
- Zero-load latency is calculated analytically:
  \[ T_\delta = H_R t_R + H_C t_C + \frac{L}{b} \]
- **Observation**
  - Router area **does not scale quadratically** as radix increases.
  - A packet can travel a very long distance on the channel in one cycle.
ANALYTICAL MODELING RESULTS

- Moderate ruche factor improves bandwidth and/or reduces area
- Moderate concentration reduces latency at similar bandwidth and area
- Increasing ruche factor does not necessarily improve latency as it may lead to narrower channels which increases serialization latency
Implementing Low-Diameter OCN for Manycore Processors Using A Tiled Physical Design Methodology

Motivation

Manycore OCN Topologies

Manycore OCN Analytical Modeling

Manycore OCN Physical Design

PyOCN Framework
HARD MACRO DESIGN METHODOLOGY

- Map global timing constraints to local timing constraints
- Use three metal layers for local horizontal routing (M2, M4, M6), three layers for vertical routing (M3, M5, M7)
- Connect “dummy cores” to the injection and ejection ports of the router to prevent ASIC toolflow from optimizing away any logic
- Use routing and placement blockages to prevent the ASIC toolflow from using the routing resources reserved for the real cores
**EXAMPLE HARD MACROS**

- 185µm mesh-c1r0-b32
- 185µm mesh-c1r0-b64
- 185µm mesh-c1r0-b128
- 185µm mesh-c1r0q0-b32
- torus-c1r0-b32

---

**Concentration Factor of Four**

- 375µm mesh-c4r0-b128
- mesh-c4r2-b64

---

**No Concentration & Ruche Channels**

- No Ruche Channels

---

**Ruche Factor of Two**

Motivation • Topologies • Analytical Modeling • Physical Design • PyOCN Framework
1. Design is based on tiling a homogeneous hard macro across the chip
2. All chip top-level routing between hard macros must use short wires to neighboring macros
3. Timing closure for the hard macro must imply timing closure at the chip level
MACRO-LEVEL RESULTS FOR PROMISING TOPOLOGIES

Bandwidth vs Area

Latency vs Area
64b Message

Latency vs Area
256b Message

- Increasing bandwidth leads to increase in area for all topologies

- Increasing concentration and ruche factor leads to lower latency & lower Area
Implementing Low-Diameter OCN for Manycore Processors Using A Tiled Physical Design Methodology

Motivation

Manycore OCN Topologies

Manycore OCN Analytical Modeling

Manycore OCN Physical Design

PyOCN Framework
PyOCN: A Unified Framework for Modeling, Testing, and Evaluating OCN

Motivation

- Topologies
- Analytical Modeling
- Physical Design
- PyOCN Framework
Motivation • Topologies • Analytical Modeling • Physical Design • PyOCN Framework

Open-Sourced on GitHub
https://github.com/cornell-brg/pymtl3-net

Packaged on PyPi
https://pypi.org/project/pymtl3-net

IEEE Int’l Conf. on Computer Design (ICCD-37), November 2019

To create a virtual environment and install pymtl3-net along with all of its dependencies:

```
python3 -m venv $(HOME)/venv/pymtl3
source $(HOME)/venv/pymtl3/bin/activate
pip install pymtl3-net
```

To test a 4-terminal ring with with a single packet:

```
pymtl3-net test ring --terminals 4 \ --dump-vcd
```

To simulate a 2x2 mesh with specific injection rate:

```
pymtl3-net sim mesh --nrows 2 --nrows 2 \ --injection-rate 10 -v
```

To simulate a 2x2 mesh across injection rates:

```
pymtl3-net sim mesh --nrows 2 --nrows 2 \ --sweep -v
```

To generate a 4x4 mesh Verilog RTL:

```
pymtl3-net gen mesh --nrows 4 --nrows 4
```
Implementing Low-Diameter OCN for Manycore Processors Using A Tiled Physical Design Methodology

- We present a tiled physical design methodology to implement low-diameter OCNs for manycore processors
- We analyze the latency, area, and bandwidth tradeoffs of 12 topologies with different concentration and ruche factor
- Long channels are the key to fully exploiting the VLSI wiring capability but must leverage a tiled physical design methodology
- Moderate concentration and ruching can reduce latency at similar area and bisection bandwidth

This work was supported in part by NSF CRI Award #1512937, DARPA POSH Award #FA8650-18-2-7852, and equipment, tool, and/or physical IP donations from Intel, Synopsys, and Cadence.