Run-Time Monitoring Mechanism for Efficient Design of Application-specific NoC Architectures in Multi/Manycore Era

Akram Ben Ahmed, Takayuki Ochi, Shohei Miura, Abderezek Ben Abdallah
The University of Aizu
Graduate School of Computer Science and Engineering,
Adaptive Systems Laboratory, Aizu-Wakamatsu, Japan.

Email: d8141104@u-aizu.ac.jp
Outline

• Background

• Run-time Monitoring Mechanism (RMM)
  – Workflow
  – PNoC Architecture
  – Updating algorithm

• Evaluation

• Conclusion
Outline

• Background
  • Run-time Monitoring Mechanism (RMM)
    – Workflow
    – PNoC Architecture
    – Updating algorithm

• Evaluation

• Conclusion
Background: Bus-based system Vs. NoC

Bus-based system

- Parallelism problem
- High latency

NoC diagram:
Background: Bus-based system Vs. NoC

NoC based system [Carloni2009, Ben2006]
Background: Bus-based system Vs. NoC

NoC based system [Carloni2009, Ben2006]
Background: NoC Challenges

- Routing [Sullivan1977, Seo2005]

Path selection has an impact on the system performance
Background: NoC Challenges

- Routing [Sullivan1977, Seo2005]
- Flow control [Agarwal2009, Pullini2005]

Efficient flow control is crucial
Background: NoC Challenges

- **Routing** [Sullivan1977, Seo2005]

- **Flow control** [Agarwal2009, Pullini2005]

- **Topology**
  - **Mesh** [Zhang2011]
    - Uniform connection
    - Large hop count

The long distance affects the latency, throughput and power
Background: NoC Challenges

- **Routing** [Sullivan1977, Seo2005]

- **Flow control** [Agarwal2009, Pullini2005]

- **Topology**
  - **Mesh** [Zhang2011]
  - **Torus** [Dally1986]
    - Connects the network extremities to reduce the inter-node distance

- Increasing complexity
- Different wire lengths
- Clock skew
Background: NoC Challenges

- Routing [Sullivan1977, Seo2005]
- Flow control [Agarwal2009, Pullini2005]
- Topology
  - Mesh [Zhang2011]
  - Torus [Dally1986]
  - Customized [Bolotin2004]
    - Especially designed for specific application
- Long design time
- Difficult to implement
Background: Motivation

• One of the important design challenges: **Buffer depth**
  
  – Efficiency of hardware resources
  – Occupies large area of NoC’s router
  – Consumes large power
  – Impact on throughput of NoC

  ✓ Enough buffer depth can help in reducing the average packet delivery latency and the possibility of dropped packets

  – Depends on traffic feature through routers where buffers are located
The type of application requires different features and traffic features

- The type of application example
  - Signal processing, Media processing, Multimedia, and etc
- Traffic feature example
  - Concentrated or distributed, large or small amount, etc
Background: OASIS2-NoC

- 4x4 Mesh topology
- Wormhole-like switching
- Stall-and-Go flow control
- 20 bits flit

OASIS2-NoC 4x4 network system [*]

Background: Approach and Requirements

• Approach:

  **Efficient assignment of buffer resources**
  – Large buffers should be located in heavy traffic channels
  – The buffer overheads in light traffic channels should be reduced

• Requirements:

  **Profiling traffic features of target applications**
  – Monitoring traffic load and congestion on NoC-based SoC
  – Computing optimal buffer depth for traffic load
Outline

• Background

• Run-time Monitoring Mechanism (RMM)
  – Workflow
  – PNoC Architecture
  – Updating algorithm

• Evaluation

• Conclusion
RMM: Workflow

(a)Resetting PNoC systems
   Default buffer depth reconfiguration
   Connecting applications to PNoC

(b)Simulating these applications on PNoC
    Monitoring traffic load and congestion
    Dumping traffic snapshots

(c)Generating traffic reports
    Executing buffer update algorithm
    Re-simulating the applications with updated buffers
    Decision of optimal buffer depths
Outline

• Background

• Run-time Monitoring Mechanism (RMM)
  – Workflow
  – PNoC Architecture
  – Updating algorithm

• Evaluation

• Conclusion
RMM:
PNoC Architecture

- PNoC hardware components
  - NoC with monitoring probes
    - Where applications run
  - Monitoring controller
    - Controls PNoC behavior
  - Monitoring interface
    - Interface between NoC, Controller, and snapshot MEM
  - Snapshot MEM

PE = Processing Element, I/F = Interface, R = Router, NI = Network Interface, P = Monitoring Probe
RMM:
PNoC Router architecture
RMM: PNoC Router architecture

Input module
Input data enter to these modules
- Input buffer (BW)
- Look-Ahead-XY routing (RC)

Crossbar

Monitoring Probe

Stop_in
Config_info
Flit_from_L
Flit_from_N
Flit_from_E
Flit_from_S
Flit_from_W
Timeout

Local input port
North input port
East input port
South input port
West input port

Grant[4:0]
port_req[24:0]
sw_req[4:0]
Flit_sent[4:0]
enqueues

Flit_to_L
Flit_to_N
Flit_to_E
Flit_to_S
Flit_to_W
Total_flits
Congestion
Stop_out
RMM: PNoC Router architecture

- **Arbiter**: Handles the arbitration between the different input port request (SA)
- **Stall/Go**: Includes the flow control module
RMM: PNoC Router architecture

Crossbar
Handles the transfer of flits to their appropriate channels depending on the information received from the arbiter (CT)
Monitoring Probe: Monitors the incoming flits at each input buffer and sends the results to the controller.
RMM:
Monitoring probe

Recording traffic load and congestions

Receiving timeout pulse

Probe Reg: North

Traffic Reg

Dump Reg

Congestion Reg

Dump Reg

Probe Reg: East

Snapshot data traversal

Probe Reg: South

Probe Reg: West

Traffic snapshot

Congestion snapshot

Buffer enqueue

Arbitration block

The University of Aizu
Adaptive systems lab
Outline

• Background

• Run-time Monitoring Mechanism (RMM)
  – Workflow
  – PNoC architecture
  – Updating algorithm

• Evaluation

• Conclusion
RMM: Updating Algorithm

Assignment of buffer resources on heavy traffic channels

Reduction of extra buffer overhead

Buffer depths for applications with low overhead

Traffic Snapshots

Generating traffic reduction
- Total traffic
- Largest congestion

Any congestion?
- No
  - Buffer depth increment
- Yes
  - Results from decreased buffers?
    - No
      - Reverting decreased buffer
    - Yes
      - Grouping routers and buffer depth decrement

Re-simulation with new buffer depth

Any congestion?
- Yes
- No
Outline

• Background
• Run-time Monitoring Mechanism (RMM)
  – Flowchart
  – PNoC architecture
  – Updating algorithm
• Evaluation
• Conclusion
## Evaluation: Methodology

<table>
<thead>
<tr>
<th>Design parameter</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Topology</td>
<td>4x2 and 4x4 in Mesh</td>
</tr>
<tr>
<td>Routing</td>
<td>XY-deterministic routing</td>
</tr>
<tr>
<td>Flow Control</td>
<td>Stall/Go</td>
</tr>
<tr>
<td>Buffer Depth</td>
<td>2 slots, 4 slots, optimal slots</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Evaluation parameters</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Design tools</td>
<td>Quartus II, NiosII IDE</td>
</tr>
<tr>
<td>FPGA</td>
<td>Altera Stratix III EP3SL150</td>
</tr>
<tr>
<td>The number of flits in a packet</td>
<td>4 flits</td>
</tr>
<tr>
<td>The number of packets</td>
<td>1000 packets / PE</td>
</tr>
<tr>
<td>Traffic patterns</td>
<td>Shuffle and Bit complement</td>
</tr>
<tr>
<td>Target applications</td>
<td>Nios II processors and memory units</td>
</tr>
</tbody>
</table>
Evaluation goals:

1. Traffic load monitored by RMM
2. Hardware complexity and performance of optimal buffer depth
   - Better performance with similar or a little extra overhead
   - Lower overhead without performance degradation or with a little degradation

<table>
<thead>
<tr>
<th>Related Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buffer Depth</td>
<td>2 slots (WH), 4 slots (VCT), optimal slots</td>
</tr>
<tr>
<td>The number of flits in a packet</td>
<td>4 flits</td>
</tr>
</tbody>
</table>
Evaluation:
Traffic load in 4x2 network

- Traffic concentrated at the network center
  - Combination of source and destination, and deterministic routing algorithm
  - Several communications share those channels
- A little too small network to distribute traffic load to network
Evaluation:
Traffic load in 4x4 network

- Traffic concentrated at the network center with bit-complement
  - Congestions between x1 line and x2 line
- Distributed traffic on shuffle pattern
- Bit complement pattern caused larger traffic than shuffle pattern
  - Long hop communications on bit complement pattern
# Evaluation: Hardware complexity

## 4x2 Network

<table>
<thead>
<tr>
<th>Buffer depth</th>
<th>WH</th>
<th>VCT</th>
<th>Optimal(bit)</th>
<th>Optimal(shuffle)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buffer depth</td>
<td>2</td>
<td>4</td>
<td>2-4</td>
<td>2-4</td>
</tr>
<tr>
<td>Area</td>
<td>3,580</td>
<td>4,206</td>
<td>3,722</td>
<td>3,632</td>
</tr>
<tr>
<td>Combinational ALUTs</td>
<td>2,276</td>
<td>3,680</td>
<td>2,643</td>
<td>2,504</td>
</tr>
</tbody>
</table>

## 4x4 Network

<table>
<thead>
<tr>
<th>Buffer depth</th>
<th>WH</th>
<th>VCT</th>
<th>Optimal(bit)</th>
<th>Optimal(shuffle)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buffer depth</td>
<td>2</td>
<td>4</td>
<td>2-4</td>
<td>2-4</td>
</tr>
<tr>
<td>Area</td>
<td>9,031</td>
<td>10,236</td>
<td>9,584</td>
<td>9,330</td>
</tr>
<tr>
<td>Combinational ALUTs</td>
<td>5,448</td>
<td>8,570</td>
<td>6,304</td>
<td>5,891</td>
</tr>
</tbody>
</table>

**Notes:**
- WH: + 4 %
- VCT: + 8 %
- Optimal(bit): + 6 %
- Optimal(shuffle): + 2 %
## Evaluation: Hardware complexity

### 4x2 Network

<table>
<thead>
<tr>
<th>Buffer Depth</th>
<th>WH</th>
<th>VCT</th>
<th>Optimal(bit)</th>
<th>Optimal(shuffle)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buffer depth</td>
<td>2</td>
<td>4</td>
<td>2-4</td>
<td>2-4</td>
</tr>
<tr>
<td>Area</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Combinational ALUTs</td>
<td>3,580</td>
<td>4,206</td>
<td>3,722</td>
<td>3,632</td>
</tr>
<tr>
<td>Total registers</td>
<td>2,276</td>
<td>3,680</td>
<td>2,643</td>
<td>2,504</td>
</tr>
</tbody>
</table>

### 4x4 Network

<table>
<thead>
<tr>
<th>Buffer Depth</th>
<th>WH</th>
<th>VCT</th>
<th>Optimal(bit)</th>
<th>Optimal(shuffle)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buffer depth</td>
<td>2</td>
<td>4</td>
<td>2-4</td>
<td>2-4</td>
</tr>
<tr>
<td>Area</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Combinational ALUTs</td>
<td>9,031</td>
<td>10,236</td>
<td>9,584</td>
<td>9,330</td>
</tr>
<tr>
<td>Total registers</td>
<td>5,448</td>
<td>8,570</td>
<td>6,304</td>
<td>5,891</td>
</tr>
</tbody>
</table>

- **22%**
- **20%**
- **19%**
## Evaluation: Execution Time

### 4x2 Network

<table>
<thead>
<tr>
<th>Traffic Pattern</th>
<th>WH</th>
<th>VCT</th>
<th>optimal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shuffle</td>
<td>6.44 ms</td>
<td>5.63 ms</td>
<td>5.62 ms</td>
</tr>
<tr>
<td>Bit complement</td>
<td>5.78 ms</td>
<td>4.26 ms</td>
<td>4.30 ms</td>
</tr>
</tbody>
</table>

- **18% improvement**

### 4x4 Network

<table>
<thead>
<tr>
<th>Traffic Pattern</th>
<th>WH</th>
<th>VCT</th>
<th>optimal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shuffle</td>
<td>14.7 ms</td>
<td>11.5 ms</td>
<td>11.3 ms</td>
</tr>
<tr>
<td>Bit complement</td>
<td>13.0 ms</td>
<td>9.81 ms</td>
<td>9.87 ms</td>
</tr>
</tbody>
</table>

- **23% improvement**

---

The University of Aizu  
Adaptive systems lab  
35
Evaluation: Execution Time

### 4x2 Network

<table>
<thead>
<tr>
<th>Traffic Pattern</th>
<th>WH</th>
<th>VCT</th>
<th>optimal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shuffle</td>
<td>6.44 ms</td>
<td>5.63 ms</td>
<td>5.62 ms</td>
</tr>
<tr>
<td>Bit complement</td>
<td>5.78 ms</td>
<td>4.26 ms</td>
<td>4.30 ms</td>
</tr>
</tbody>
</table>

Negligible

### 4x4 Network

<table>
<thead>
<tr>
<th>Traffic Pattern</th>
<th>WH</th>
<th>VCT</th>
<th>optimal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shuffle</td>
<td>14.7 ms</td>
<td>11.5 ms</td>
<td>11.3 ms</td>
</tr>
<tr>
<td>Bit complement</td>
<td>13.0 ms</td>
<td>9.81 ms</td>
<td>9.87 ms</td>
</tr>
</tbody>
</table>

The University of Aizu

Adaptive systems lab
Outline

• Background
• Run-time Monitoring Mechanism (RMM)
  – flowchart
  – Architecture
  – Updating algorithm
• Evaluation
• Conclusion
• Proposal of Run-time Monitoring Mechanism (RMM) which monitors the traffic load and effectively computes optimal buffer depth.

• From the evaluation results, we found that the system performance in terms of execution time was about 23% better when compared with traditional design methods.
Future work

• Our future work is to use large network size to explore further parameters optimizations.
• Use real and larger benchmarks (such as JPEG encoder) to illustrate proposed architecture under real and large traffic loads.
References


Thank you

for your attention