An Efficient Implementation of Distributed Routing Algorithms in NoCs

Authors: J. Flich, S. Rodrigo, and J. Duato

Parallel Architectures Group
Technical University of Valencia, Spain
Agenda

Introduction
System environment
Description
Evaluation
[Further evaluations]
Conclusions
Introduction

- Multi-core architectures are becoming mainstream for designing high performance processors
- Performance on single-core solutions is limited by power
- The trend is to integrate a large number of cores inside a chip
- Need for a high-performance on-chip interconnect (NoC) to communicate efficiently between all chip devices
Introduction (2)

- Area, power and delay are the main constraints when designing a NoC

- Some problems arise:
  - High integration scale -> communication reliability issues
  - Fabrication faults

- Those problems lead to an irregular topology still functional
Introduction (3)

- Virtualization of the chip is also possible thanks to the increasing number of cores
  - Efficient use of resources
  - Distributing system resources among different tasks
- So, the original 2D mesh is partitioned into different irregular topologies.
Introduction (4)

(a) 2D mesh
(b) "+" topology
(c) p topology
(d) q topology

(e) d topology
(f) b topology
(g) non-minimal topology
(h) different regions
Introduction (5)

- To deal with irregular topologies, switches based on forwarding tables are preferred off-chip.
- However, on-chip, area, power and delay constraints are critical as memories do not scale in those terms.
- PROPOSAL: LBDR (Logic-Based Distributed Routing) is implemented to get rid of tables with a minimum logic to allow the use of any distributed routing algorithm.
Agenda

Introduction

System environment

Description

Evaluation

[Further evaluations]

Conclusions
System environment

- For LBDR to be applied, some conditions must be fulfilled:
  - Messages routed with X and Y offsets, every switch must know its own coordinates
  - Every end node can communicate with other node through a minimal path

- LBDR, on the other hand:
  - There is no restriction to be applied in systems with or without virtual channel requirements.
  - Supports both wormhole and virtual cut-through switching
System environment (2)

- LBDR is applicable to any routing algorithm that enforces minimal paths for every source-destination pair:
  - A deterministic routing algorithm without cyclic dependencies can be represented by routing restrictions
  - A routing restriction forbids a packet to use two consecutive channels
System environment (3)

(a) 2D mesh  (b) "+" topology  (c) p topology  (d) q topology

(e) d topology  (f) b topology  (g) non-minimal topology  (h) different regions
System environment (4)

(a) XY in 2D mesh
(b) SRh in 2D mesh
(c) SRv in 2D mesh
(d) UD in 2D mesh

(e) XY in p topology
(f) SRh in p topology
(g) SRv in p topology
(h) UD in p topology
Agenda

Introduction

System environment

Description

Evaluation

[Further evaluations]

Conclusions
Description

- LBDR uses two sets of bits:
  - Routing bits (Rxy), 2 per each output port
  - Connectivity bits (Cx), 1 per each output port
- The four output ports are labeled as N, E, W and S
Description (2)

Routing and connectivity bits required per switch (12 bits, 3 per output port)

First part of the routing logic:
- \(X_{\text{curr}}\)\(\rightarrow\) \(E'\)
- \(X_{\text{dst}}\)\(\rightarrow\) \(W'\)
- \(Y_{\text{curr}}\)\(\rightarrow\) \(N'\)
- \(Y_{\text{dst}}\)\(\rightarrow\) \(S'\)

Second part of the routing logic:
- \(N'' = N'. E'. W' + N'. E'. Rne + N'. W'. Rnw\)
- \(E'' = E'. N'. S' + E'. N'. Ren + E'. S'. Res\)
- \(W'' = W'. N'. S' + W'. N'. Rwn + W'. S'. Rws\)
- \(S'' = S'. E'. W' + S'. E'. Rse + S'. W'. Rsw\)

Logic block for \(N\) port:
- \(N'\)
- \(E'\)
- \(W'\)
- \(Rne\)
- \(N'\)
- \(E'\)
- \(W'\)
- \(Rnw\)

Logic block for \(Cn\) output:
- \(N\)
- \(X_1\)
- \(X_2\)
- \(X_3\)
Description (3)
Description (4)

- **1\textsuperscript{st} part of logic:**
  - $S'$=1, $W'$=1
  - $N'$=0, $E'$=0

- **2\textsuperscript{nd} part of logic**
  - $S''$=0 ($Rsw$=0)
  - $W''$=1 ($Rws$=1, $W'$=1, $S'$=1)

- **Final**
  - $W$=1 ($Cw$=1) -> TO ARBITER
Description (5)

- 1\textsuperscript{st} part of logic:
  - $S'=1$
  - $W'=0$, $N'=0$, $E'=0$
- 2\textsuperscript{nd} part of logic
  - $S''=1$ ($S'=1$, $E'=0$, $W'=0$)
- Final
  - $S=1$ ($Cs=1$) -> TO ARBITER
Description (6)

- **1\textsuperscript{st} part of logic:**
  - $S' = 1$
  - $W' = 0$, $N' = 0$, $E' = 0$

- **2\textsuperscript{nd} part of logic**
  - $S'' = 1$ ($S' = 1$, $E' = 0$, $W' = 0$)

- **Final**
  - $S = 1$ ($C_s = 1$) -> TO ARBITER
Description (7)

• LBDR has visibility of one hop away -> LBDRe expands visibility to two hops away

• LBDRe adds four more bits per output port. It is a second set of routing bits (R2xy), meaning that y direction can be taken two hops away through the x direction
Description (8)

For further details of the full logic, please refer to the paper.
Description (9)

- Why LBDRe?
Description (10)
Description (11)
Agenda

Introduction
System environment
Description
Evaluation
[Further evaluations]
Conclusions
Evaluation

- NOXIM Simulator
  - Wormhole switching
  - Input port buffer 4-flit long
  - Packets 32-flit long
  - 8x8 mesh with different irregular topologies
  - XY, UD and SRh routing algorithms
Evaluation (2)

- Performance achieved for different routing algorithms on a 2D mesh
Evaluation (3)

- Comparison of performance for LBDR and LBDRe
Agenda

Introduction
System environment
Description
Evaluation

[Further evaluations]
Conclusions
Further evaluations

- Study on impact on area, power and delay constraints
- Evaluations achieved with much more detail using Synopsys Design Compiler and 90nm technology library from TSMC
- Good expectations. Region-Based Routing(*), with much more logic implied than LBDR, gets better results than implemented tables

(*) Region-Based Routing: An Efficient Routing Mechanism to Tackle Unreliable Hardware in Network on Chips, NoCs 2007
Further evaluations (2)

- Minimum logic ($n \times n$ 2D mesh, d ports):
  - Table-based: $n \times n \times d \times d$ bits
  - RBR: 4 comparators, 4 registers $\log_2(N)/2$ bits, 1 register $d+1$ bits, 1 register $d$ bits
  - LBDR: 12 bits per switch (3 per output port), 2 comparators, 2 inverters and 5 gates
Agenda

Introduction
System environment
Description
Evaluation
[Further evaluations]

Conclusions
Conclusions

- LBDR (and LBDRe) allows for implementing most of the distributed routing algorithms in suitable topologies for NoCs.

- Future work:
  - Applicability on system/chip virtualization
  - Support non-minimal paths
  - Broadcast
Thank you.