Transient and Permanent Error Control for High-End Multiprocessor Systems-on-Chip

Qiaoyan Yu[1], José Cano[2], José Flich[2], Paul Ampadu[3]

[1]University of New Hampshire, USA

[2]Universitat Politècnica de València , Spain

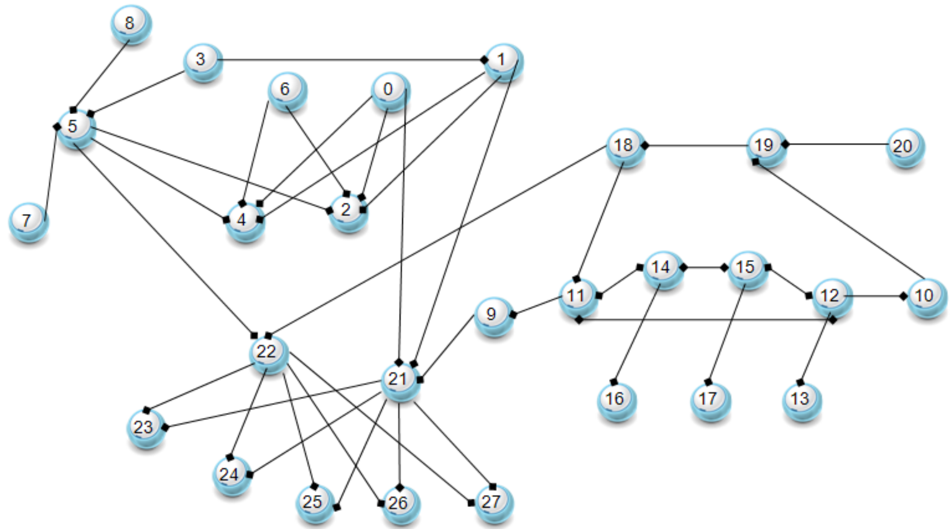
[3]University of Rochester, USA

Outline

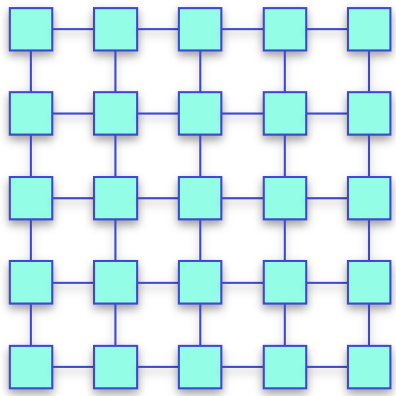
- Introduction & Motivation
  - Impact of permanent and transient errors on NoC routers
  - Advanced topologies
- Proposed method
  - LBDRhr
  - Transient error control in LBDRhr
- Experimental results
- Summary and conclusions

Introduction

- Types of MPSoCs:
  - Application-specific
    - ➢Fully irregular topologies
    - ➢System design totally customized
    - ➢E.g. Spidergon STNoC



  - High-end
    - ➢Regular structures (2D mesh-based topologies)
    - ➢E.g. Tilera



This work focuses here !!!

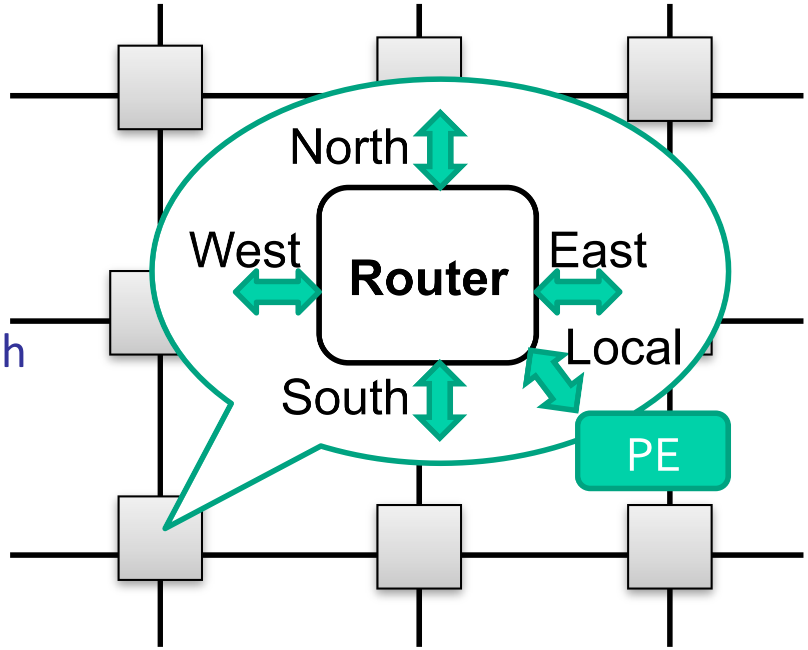
Introduction

- Critical challenge in current NoCs: **RELIABILITY**
  - Permanent errors
    - E.g. due to defective components (links, routers)
    - Solution based on fault-tolerant routing → **Logic-based Distributed Routing (LBDR)**
  - Transient errors
    - E.g. due to particle strike
    - Solution based on error control coding → **Inherent information redundancy (IIR)**

It could be a good solution for addressing both permanent and transient errors in NoCs

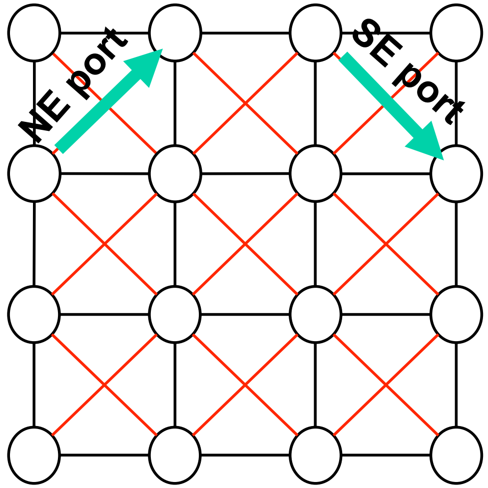
Introduction & Motivation

- Problem: both LBDR and IIR methods cannot be applied to topologies and configurations for advanced NoC topologies
  - LBDR approach
    - ➢ Designed for 2D meshes
    - ➢ Routers connected to 1 router neighbour on each dimension and direction
    - ➢ Not ready for transient errors
  - IIR approach
    - ➢ Designed for XY routing
    - ➢ Not suitable for more advanced routing solutions
    - ➢ Not ready for permanent faults



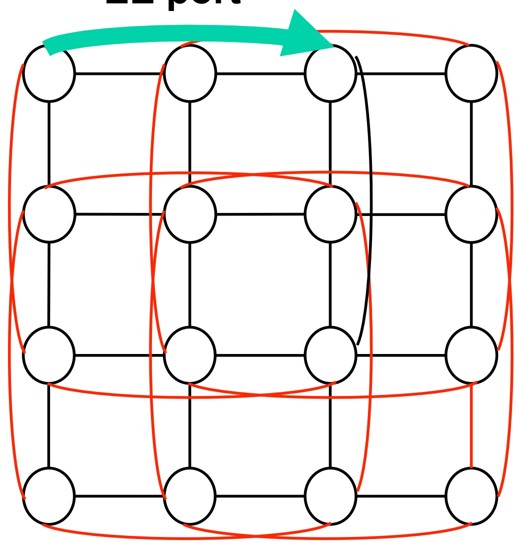
Advanced Topologies

Diagonal mesh



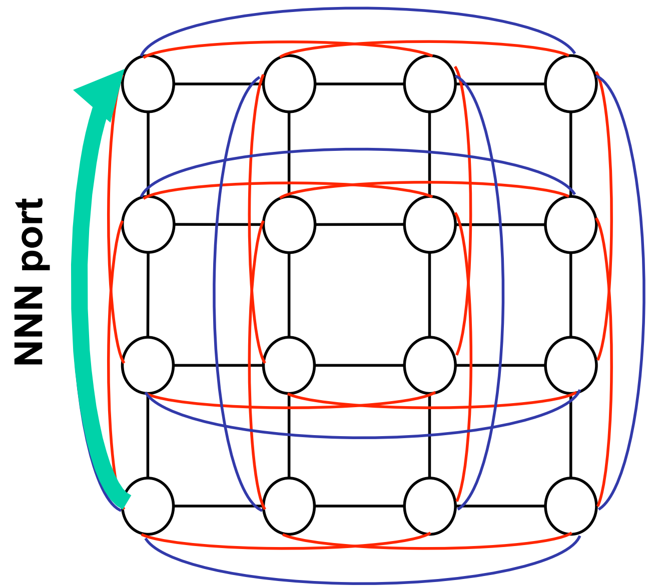
1-hop links
2-hop diagonal links

2D-mesh with express channels



1-hop links
2-hop straight links

Flattened butterfly



1-hop links
2-hop straight links
3-hop links

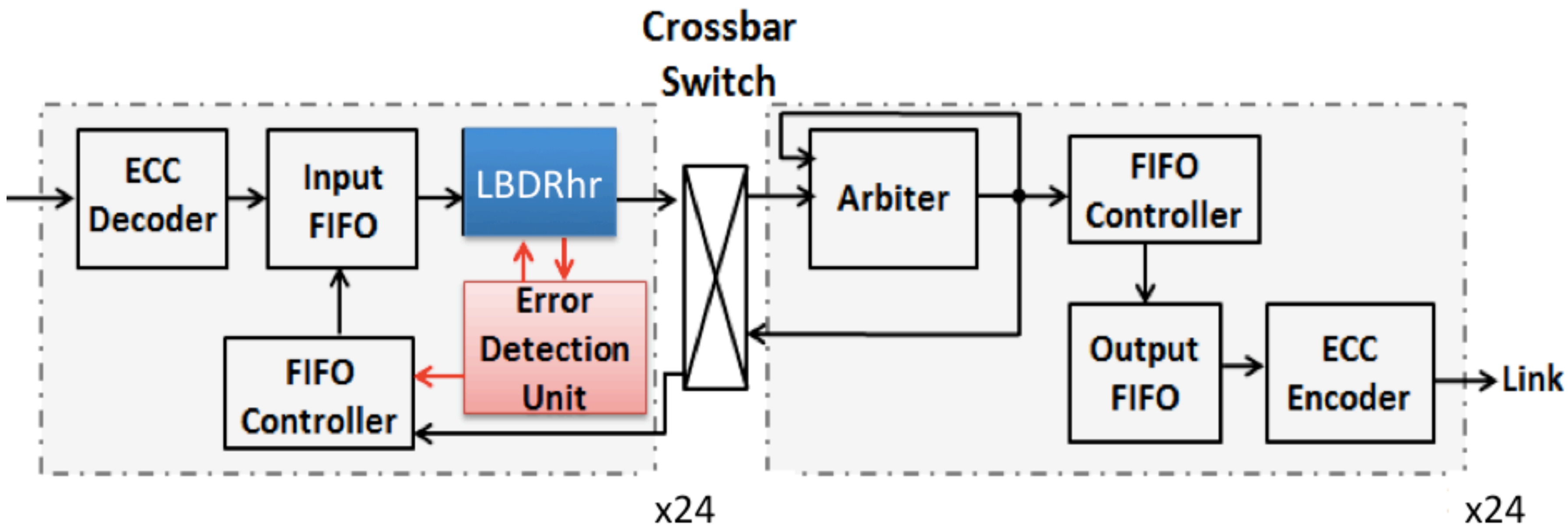
The initial 2D-mesh is the underlying topology!!!

Proposed Ideas

- To address fault tolerance for advanced topologies:
  - Redesign the LBDR mechanism: **LBDRhr** (LBDR for high-radix networks)
    - ➢Adaptive routing algorithm supported
    - ➢2 Virtual Channels
    - ➢Deadlock-free for the high-radix topologies defined
  - Develop a **new method to detect transient errors** in LBDRhr logic
    - ➢Exploits the inherent information redundancy in LBDRhr to significantly reduce the error control overhead

NoC Router Functionality

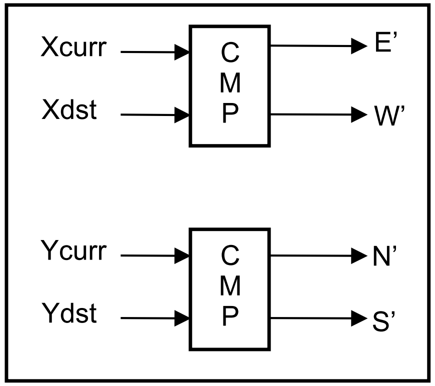
- Compute routing direction for next hop
- Pass the packet to its intended output port

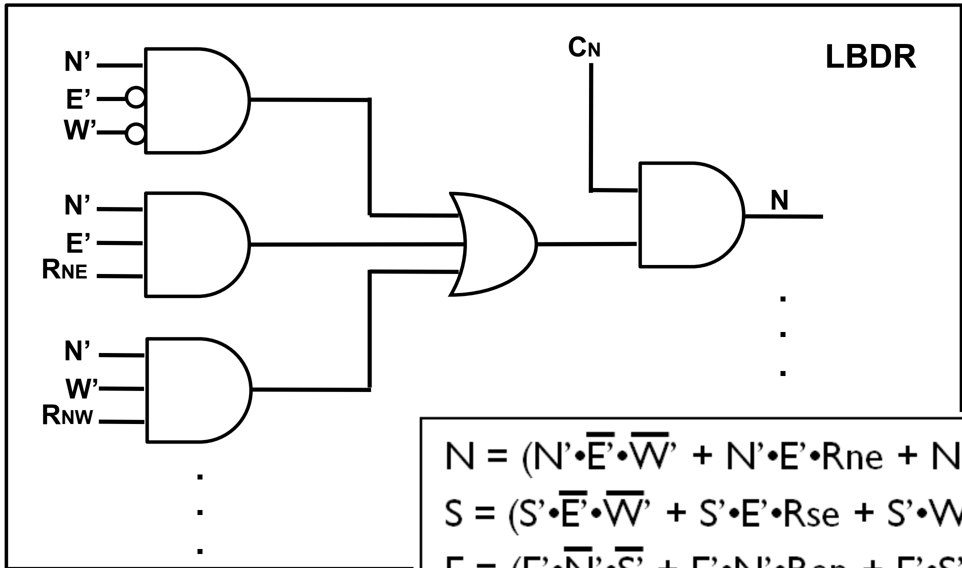


Note: 24 is the maximum number of routing ports for each router, but not all need to be implemented, depends on the topology

Permanent Error Management

- Previous method: Logic-Based Distributed Routing (LBDR)
  - Four routing ports per switch (North, South, East, West)
  - Two sets of bits: Routing bits (Rxy, 2 per output port) and Connectivity bits (Cx, 1 per output port)
  - Minimal path support





$$N = (N' \cdot \overline{E'} \cdot \overline{W'} + N' \cdot E' \cdot Rne + N' \cdot W' \cdot Rnw) \cdot Cn$$

$$S = (S' \cdot \overline{E'} \cdot \overline{W'} + S' \cdot E' \cdot Rse + S' \cdot W' \cdot Rsw) \cdot Cs$$

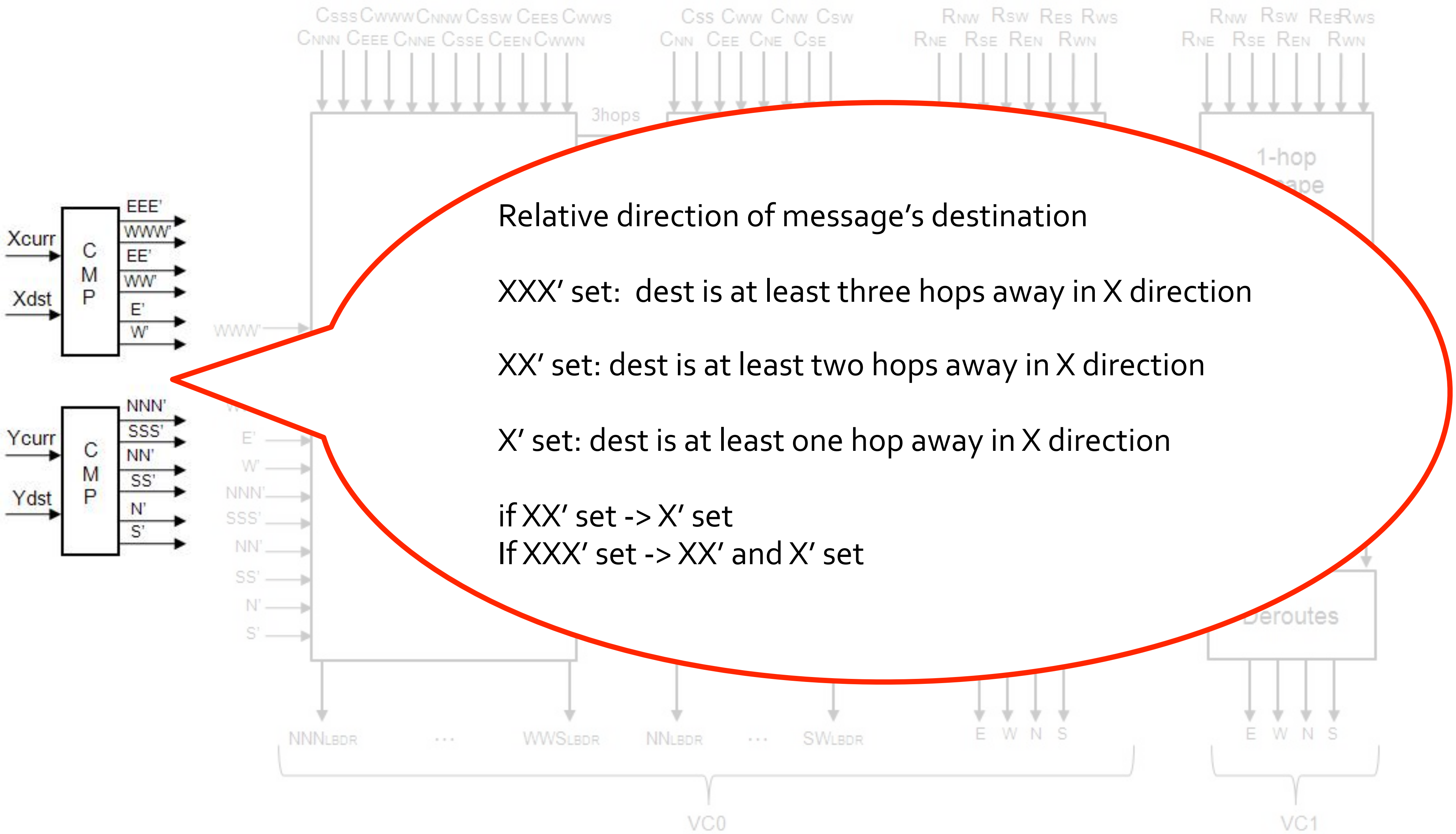
$$E = (E' \cdot \overline{N'} \cdot \overline{S'} + E' \cdot N' \cdot Ren + E' \cdot S' \cdot Res) \cdot Ce$$

$$W = (W' \cdot \overline{N'} \cdot \overline{S'} + W' \cdot N' \cdot Rwn + W' \cdot S' \cdot Rws) \cdot Cw$$

Permanent Error Management

- LBDRhr
  - Tolerates permanent link and router failures
  - Implemented with three basic logic blocks
    - 1-hop, 2-hop and 3-hop ports
  - Uses a few configuration bits to store local information about the neighboring routers
    - 8 configuration bits for routing purposes → Rxy
    - 2 bits for two deroute options (special cases) at every input port → DRx
    - 1 connectivity bit per output port → Cx

LBDRhr logic (common part)



LBDRhr logic (adaptive part)

One gate per output signal

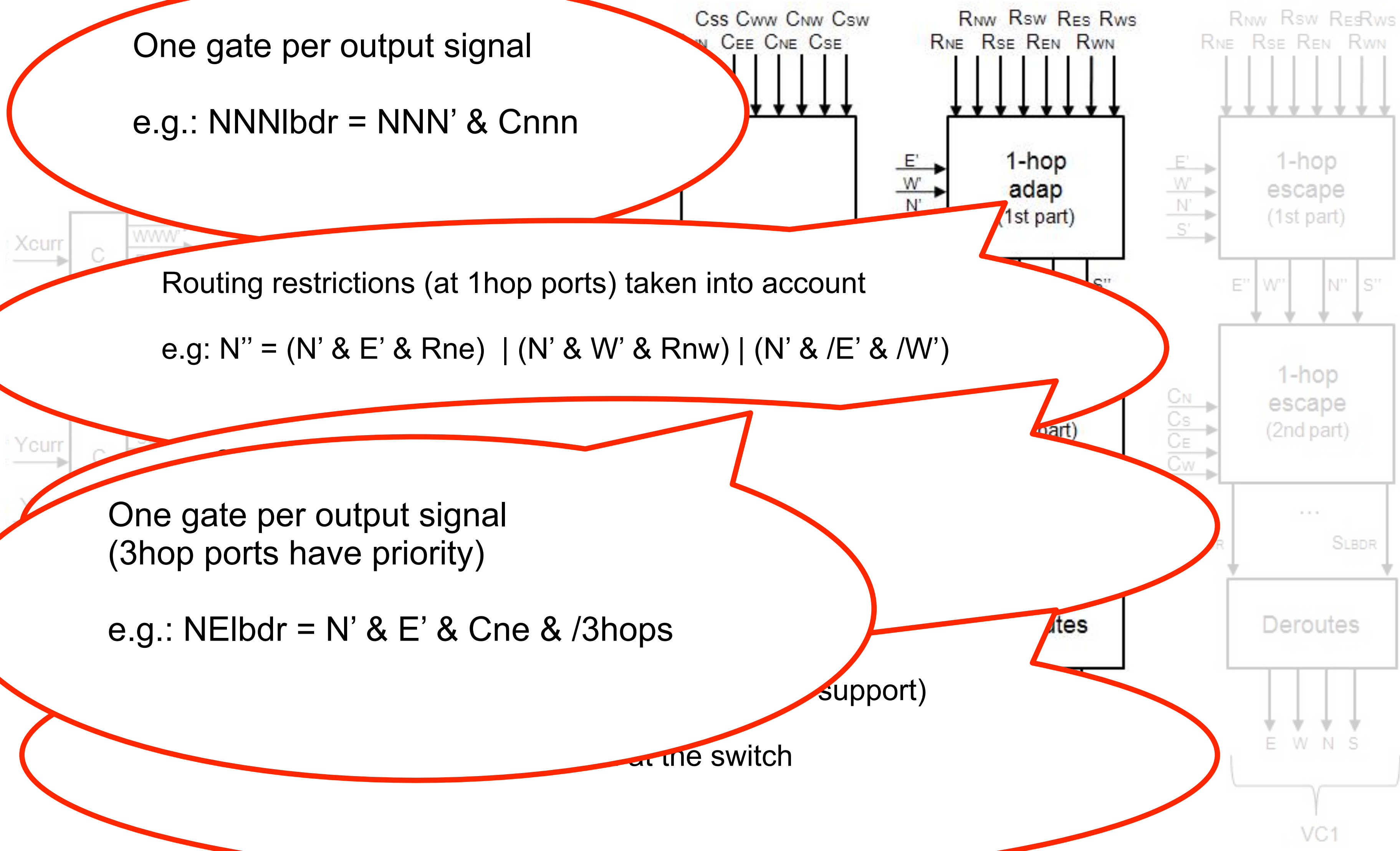
e.g.: NNNlbdr = NNN’ & Cnnn

Routing restrictions (at 1hop ports) taken into account

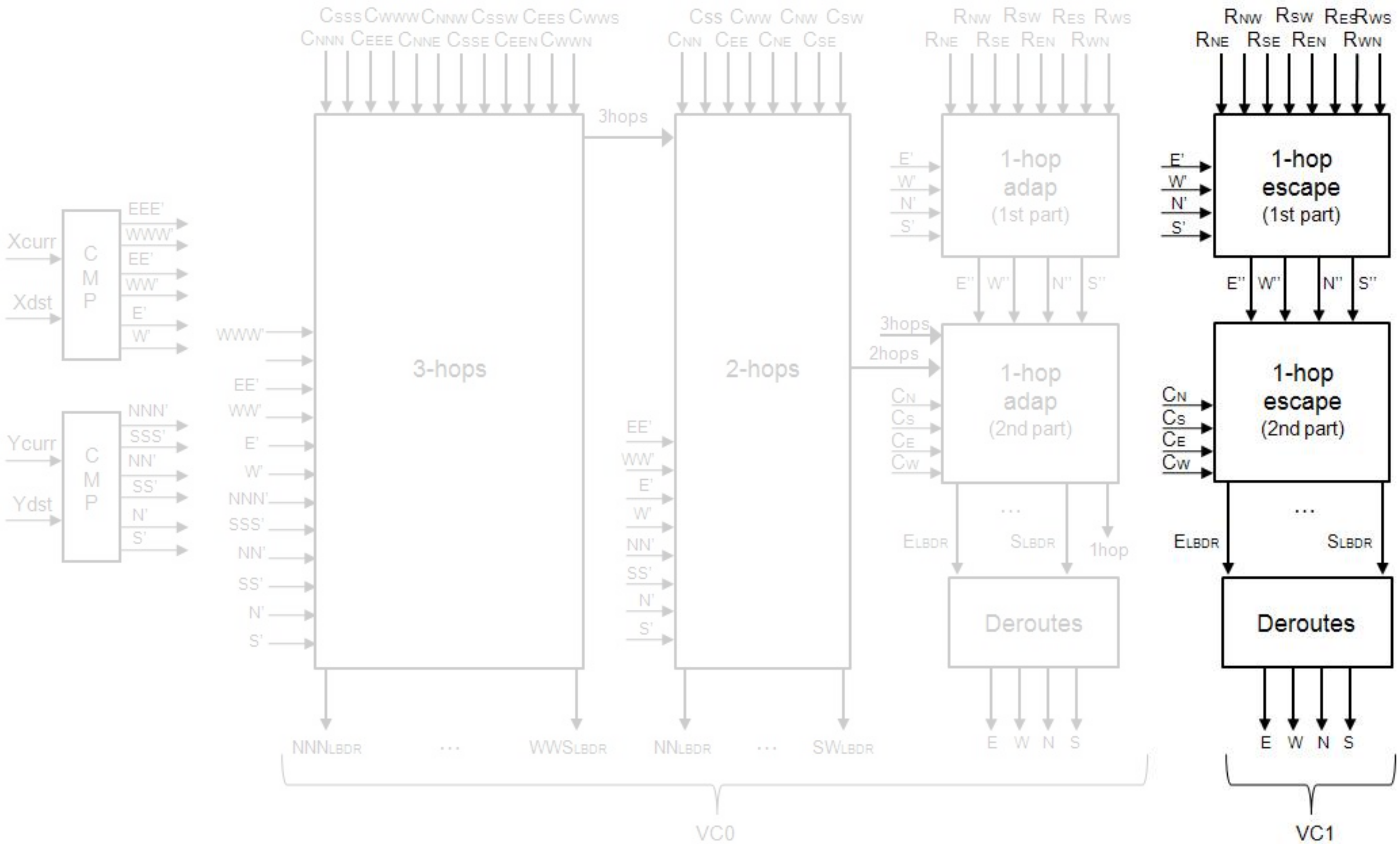
e.g: N’’ = (N’ & E’ & Rne) | (N’ & W’ & Rnw) | (N’ & /E’ & /W’)

One gate per output signal
(3hop ports have priority)

e.g.: NElbdr = N’ & E’ & Cne & /3hops

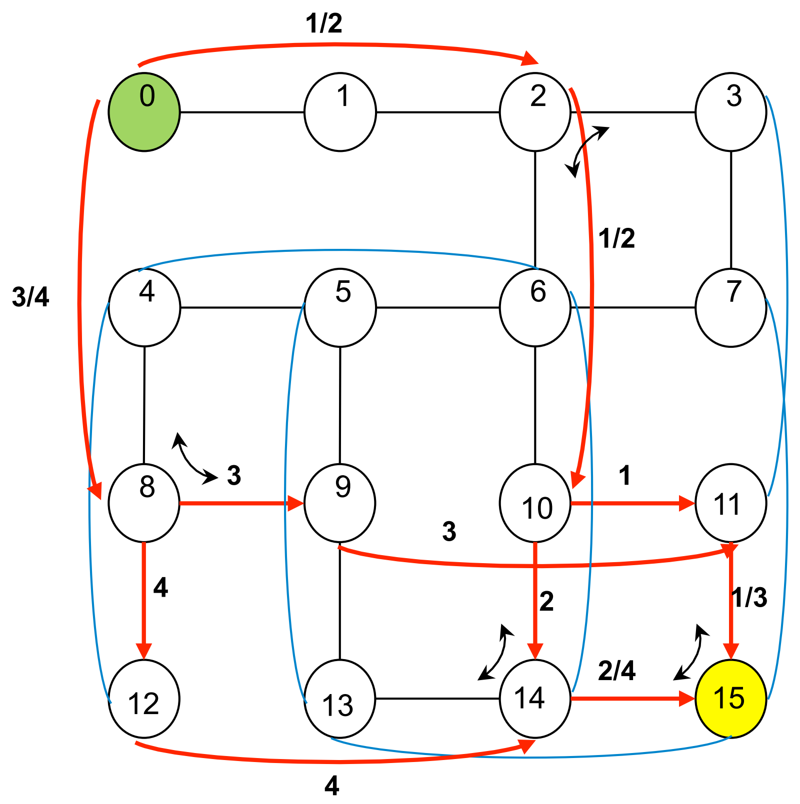


LBDRhr logic (escape part)

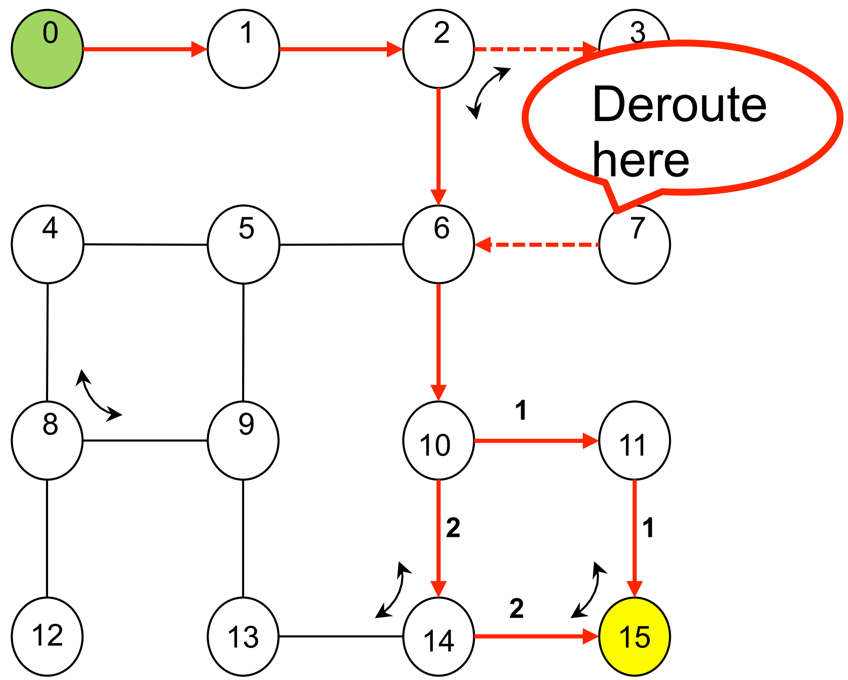


Permanent Error Management

- Deadlock-free routing example



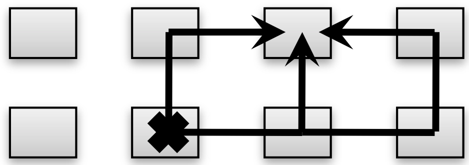
VC0: Faulty 2D-mesh with express channels



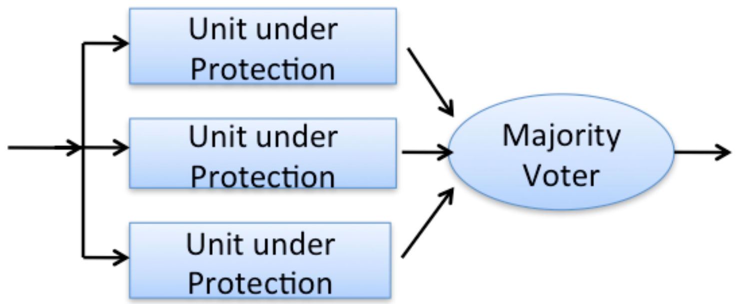
VC1: Faulty 2D-mesh

Prevision Transient Error Control Methods

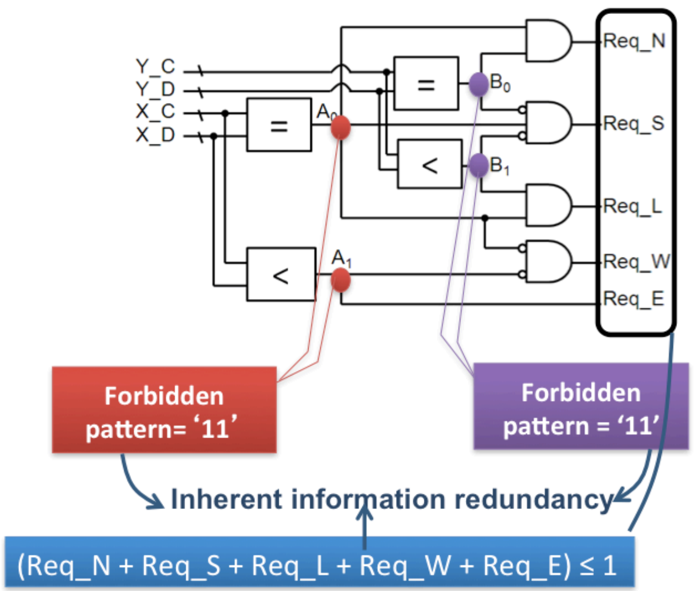
- Limitation of Previous Methods
  - Need knowledge of error locations
  - Consume large link switching power
  - Increase area overhead or
  - Limited to XY routing



Flooding[1]



Triple modular redundancy[2]

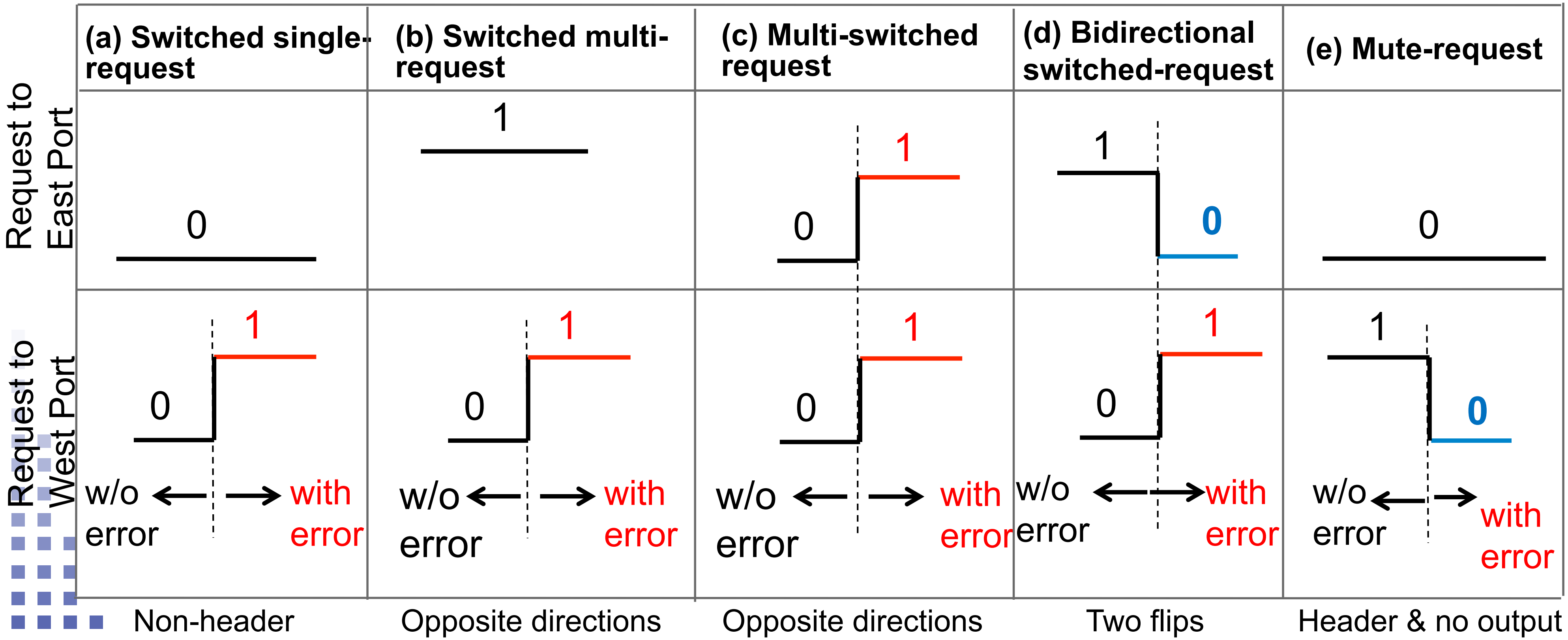


IIR[3]

[1] R. Mărculescu, ISVLSI'03. [2] A. Yanamandra, *et al.* ASP-DAC' 10.
[3] Q. Yu, et al. NOCS'11.

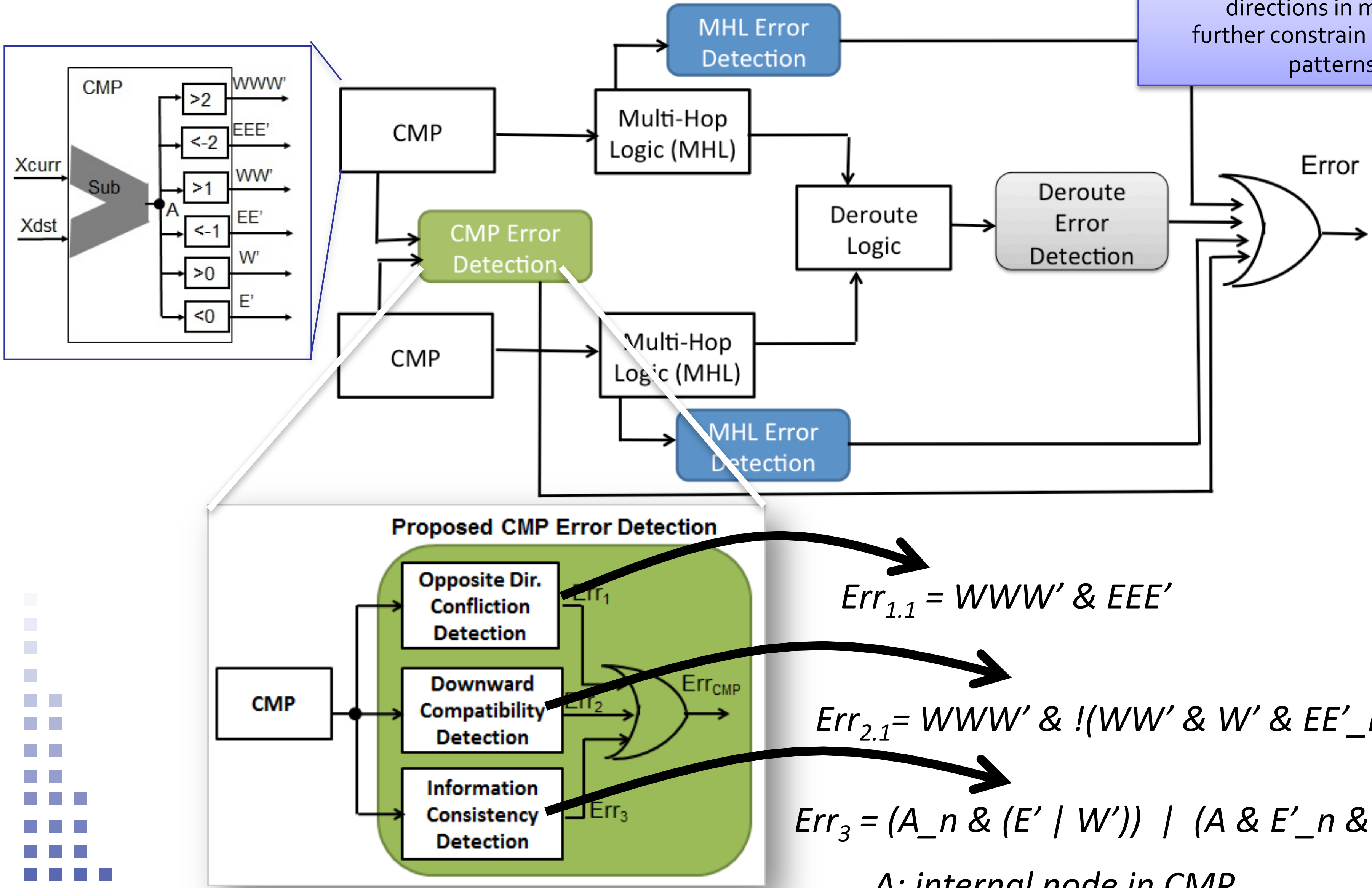
New Inherent Information Redundancy Extraction

- Forbidden signal patterns in routers are regarded as inherent information redundancy (IIR)
- More IIR are found in LBDRhr-based router



Error Detection for CMP in Router

we keep the forbidden signal patterns of the opposite directions in mind to further constrain the signal patterns



$Err_{1.1}$ = WWW' & EEE'

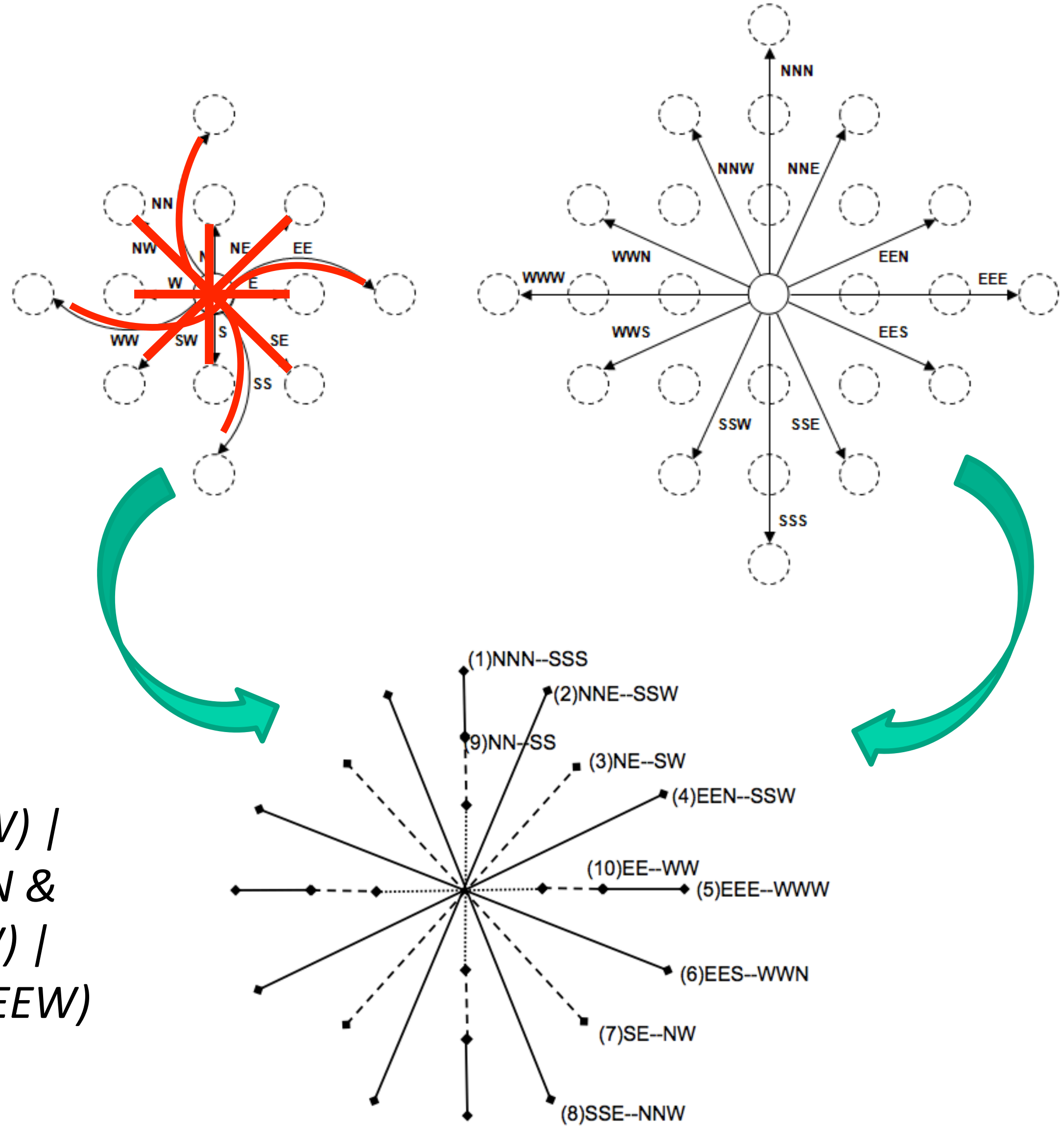
$Err_{2.1}$= WWW' & !(WW' & W' & EE'_n & E'_n)

Err_3 = (A_n & (E' | W')) | (A & E'_n & W'_n)

A: internal node in CMP

Error Detection for Multi-hop Logic

$Err_{2\text{-}hops}$ = (NN & SS) | (EE & WW) | (NE & SW) | (SE & NW)| (NE & SE) | (SW & SE)



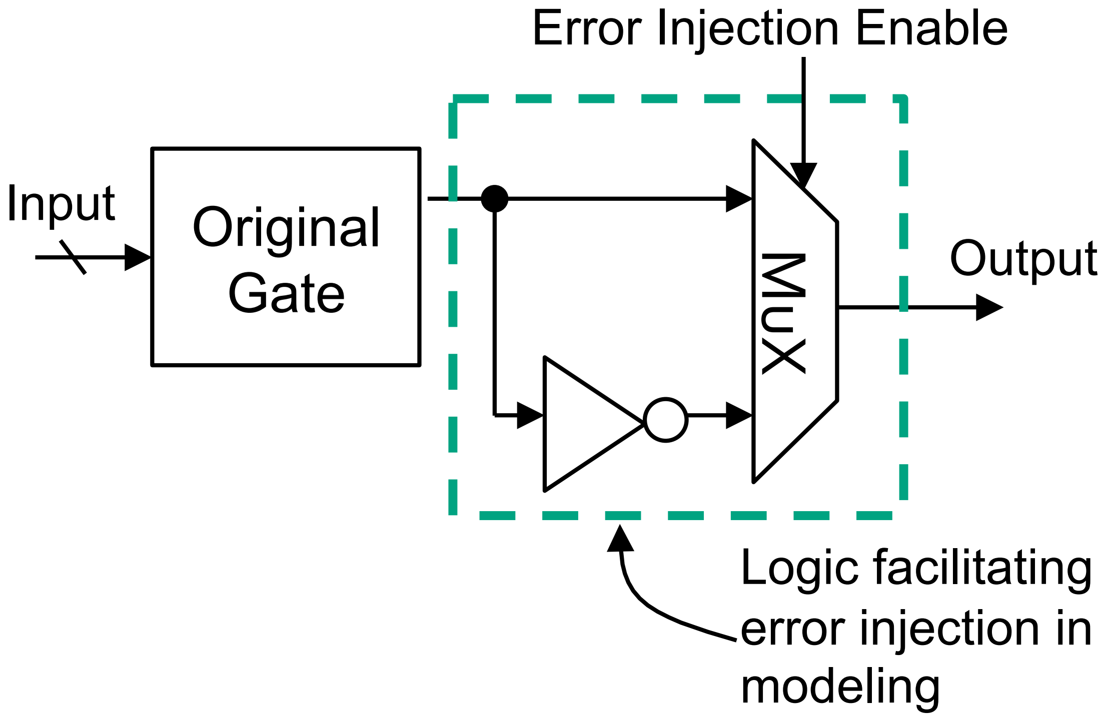
$Err_{3\text{-}hops}$ = (NNE & SSW) | (EEN & SSW) | (EES & WWN) | (SSE & NNW) | (NNN & SSS) | (EEE & WWW) | (NNE & NNW) | (SSW & SSE) | (EEN & EES) | (EES & EEW) | (WWN & WWS)

Experimental Results

- Error Detection Rate
- Reliability
- Flit Throughput and Latency
- Area, Power and Delay

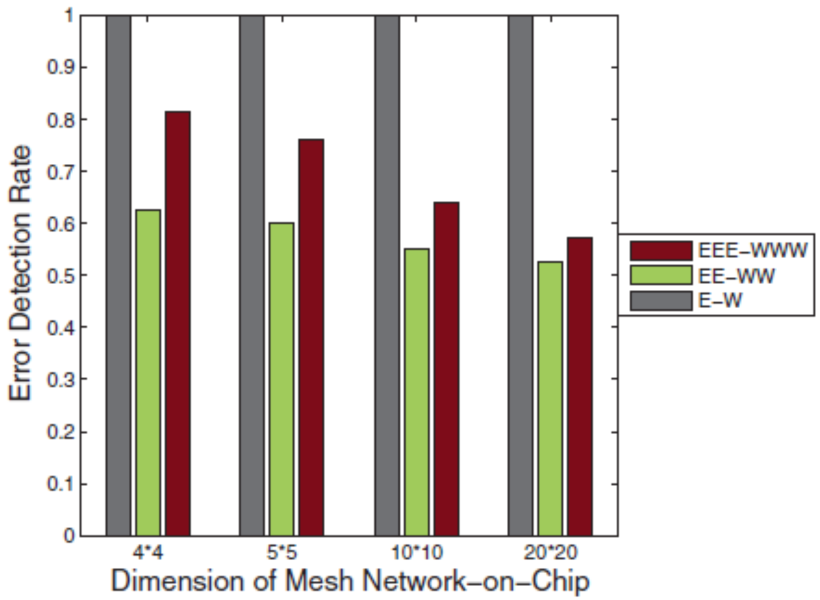
Experimental Results: Setup

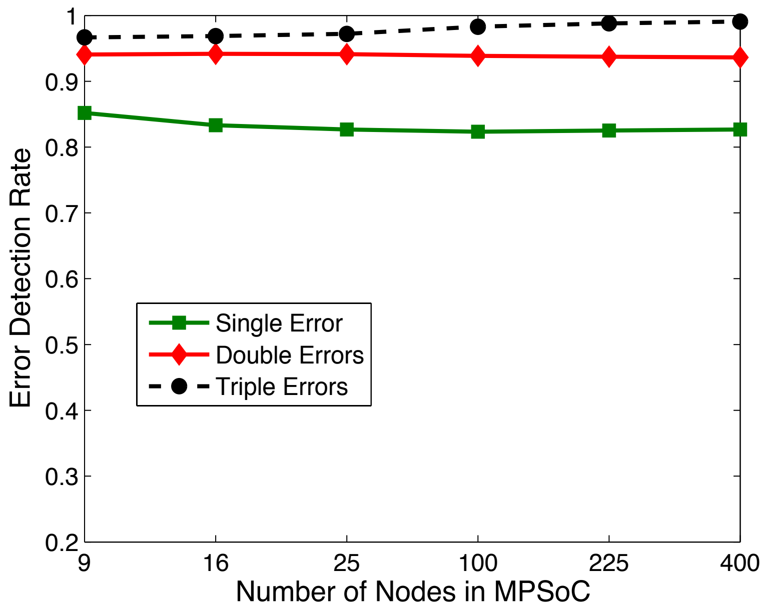
- Multiple NoC topologies
- LBDRhr Routing
- 8-bit address
- Synthesized with a TSMC 65nm technology



Error Detection Rate in CMP

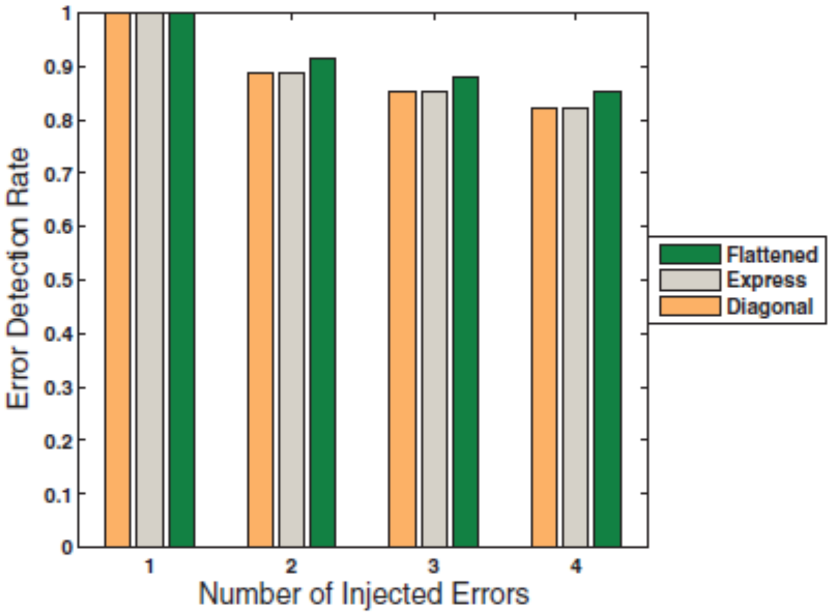
- No matter how the NoC size changes, the error detection rate for E' and W' is 100% because of the use of the internal node.
- The error detection rate for EE', WW', EEE' and WWW' is less than 1.
  - Only the occurrence of opposite direction pairs helps to detect errors in EE', WW', EEE' and WWW' (the non-zero substraction output does not contribute to detect the errors causing wrong EE', WW', EEE', WWW').



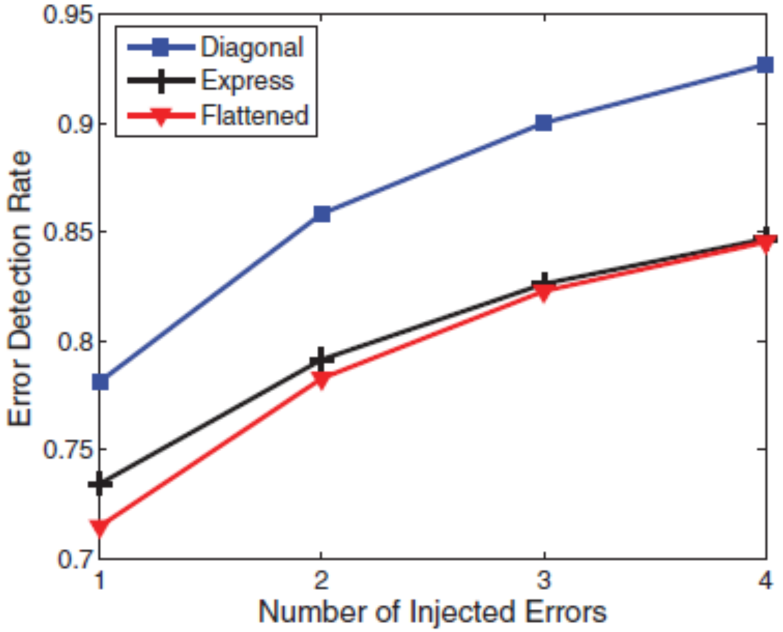


Error Detection Rate in Multi-hop Logic

- 3-hop path



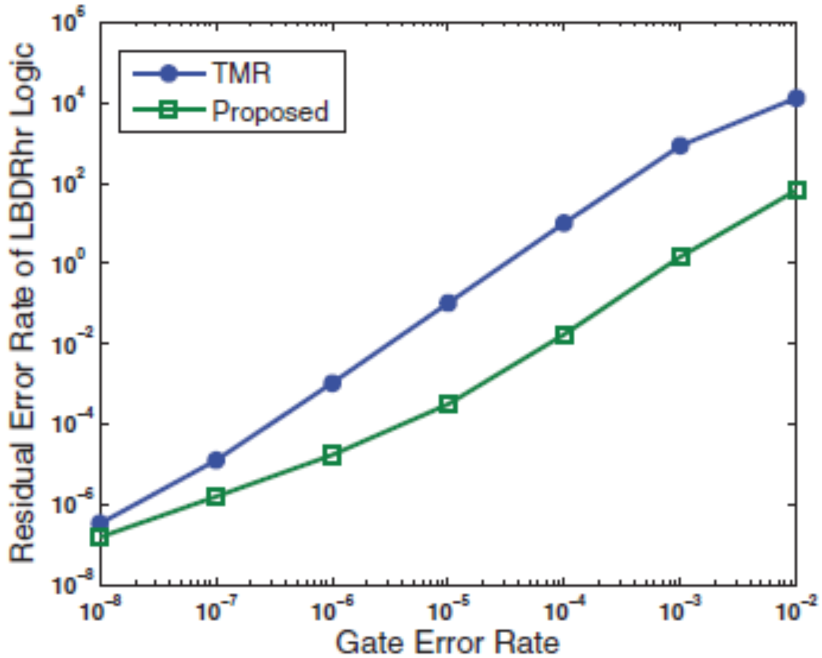
2-hop path

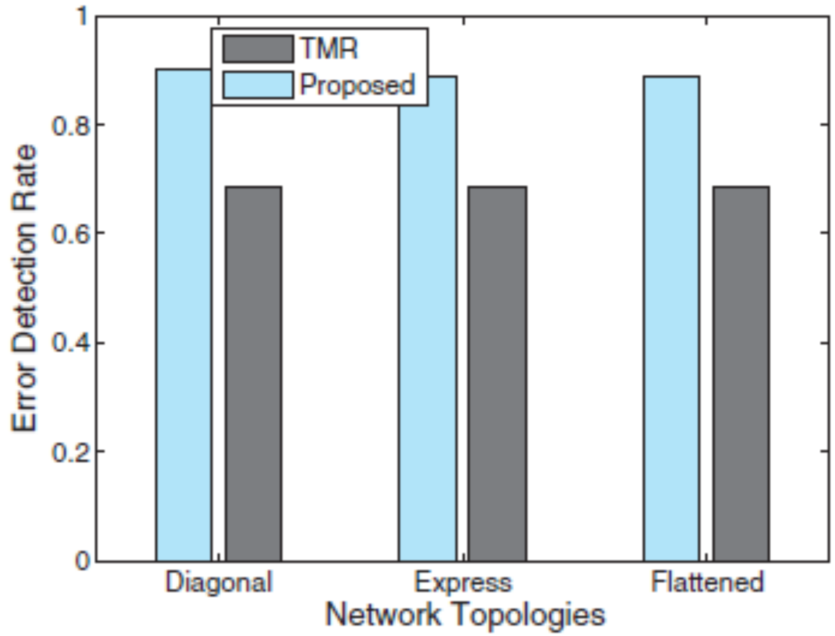


- Achieve minor variation on the error detection rate for different topologies.
- Improve the error detection rate of 2-hops logic as the number of error injected to the logic increases, because of more IIR

Residual Error Rate Comparison

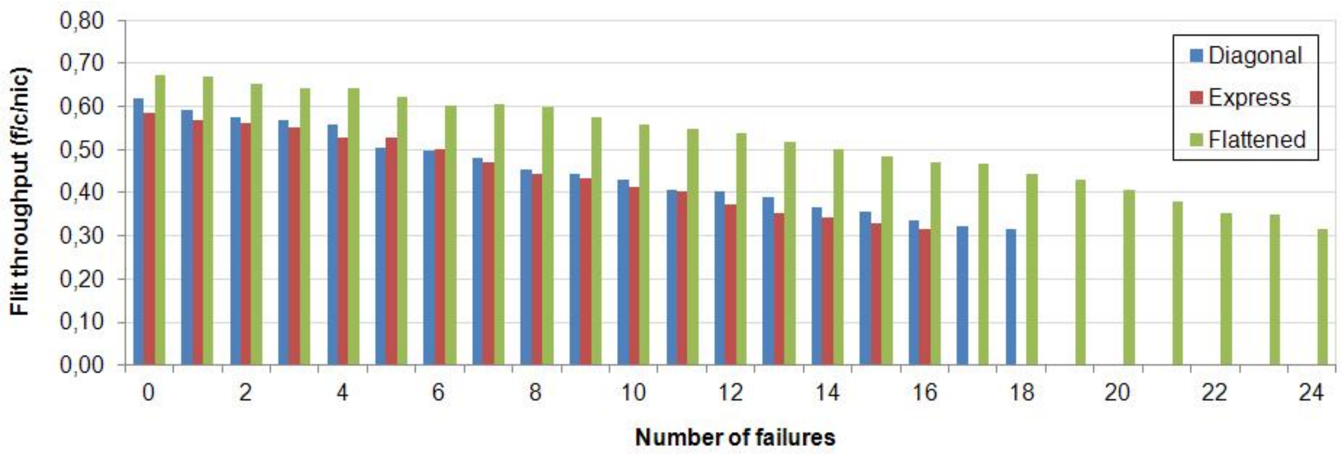
- The proposed method
  - Reduce the residual error rate by two orders of magnitude over TMR
  - Slightly vary the error detection rate





Flit Throughput and Latency

- The number of faulty links in each topology increases up to obtain the underlying 2D-mesh



Area, Power and Delay

		LBDRhr without Error Detection	LBDRhr with Proposed Error Detection	LBDRhr with TMR
Area (μm^2)		342 (100%)	363 **(106.1%)**	806 **(235.7%)**
Delay (ns)		0.495 (100%)	0.54 (109.1%)	0.51 (103%)
Power	Dyn.(µW)	199.97 (100%)	207.27 **(103.7%)**	267.39 **(133.7%)**
	Leak(µW)	1.8084 (100%)	1.8405 (101.8%)	4.0969 (226.5%)

Conclusions

- For **transient errors**, our method reduces the residual error rate and the average power consumption by up to 200x and 30%, respectively, over triple modular redundancy.

- For **permanent errors**, the proposed method is able to cover permanent failures of all the long-range links and 80% of the failure combinations of the 2D-mesh links.

Thank you!

Transient and Permanent Error Control for High-End Multiprocessor Systems-on-Chip

Qiaoyan Yu[1], José Cano[2], José Flich[2], Paul Ampadu[3]

[1]University of New Hampshire, USA

[2]Universitat Politècnica de València , Spain

[3]University of Rochester, USA

Error Detection for Deroute Logic

- Four directions are exclusive is regarded as a new inherent information redundancy

$N_{deroute}$= ~DR[0] &~DR[1]

$E_{deroute}$= ~DR[0]& DR[1]

$W_{deroute}$ =DR[0]&~DR[1]

$S_{deroute}$ =DR[0]&DR[1]