

Ph.D. Thesis

**PERFORMANCE AND RELIABILITY ENHANCEMENT
TECHNIQUES FOR
TWO DIMENSIONAL NETWORK-ON-CHIPS USING
ADAPTIVE DEFLECTION ROUTERS**

Submitted to

COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY

for the award of the degree of

Doctor of Philosophy

By

SIMI ZERINE SLEEBA

Under the guidance of

PROF. (DR.) MINI M.G.

**RESEARCH GUIDE
DEPARTMENT OF ELECTRONICS ENGINEERING
MODEL ENGINEERING COLLEGE
COCHIN, INDIA -682021**

DECEMBER, 2017

Performance and Reliability Enhancement Techniques for Two Dimensional Network-on-Chips using Adaptive Deflection Routers

Ph.D. Thesis

Author :

Simi Zerine Sreeba
Research Scholar
Department of Electronics
Model Engineering College
Thrikkakara, Kochi
simi.abie@gmail.com

Supervising Guide:

Dr. Mini M.G.
Professor in charge of Principal
College of Engineering, Cherthala, India
&
Research Guide,
Department of Electronics Engineering
Model Engineering College, Cochin, India - 682021
mininair@mec.ac.in

December, 2017

To

My beloved parents

&

My loving husband and children

DECLARATION

I hereby declare that the work presented in this thesis entitled **“Performance and Reliability Enhancement Techniques for Two Dimensional Network-on-Chips using Adaptive Deflection Routers”** is a bonafide work carried out by me, in the Department of Electronics Engineering, Model Engineering College, Kochi under the supervision of Dr. Mini M.G., Professor in charge of Principal, College of Engineering, Cherthala and Research Guide, Model Engineering College, Kochi. The results presented in this thesis or part of it have not been presented for the award of any other degree(s).

Kochi,

04-12-2017

Simi Zerine Sreeba

ACKNOWLEDGEMENTS

I thank God Almighty for his abundant grace that gave me some wonderful years of research life and experience at Model Engineering College since December, 2011. I would like to express my appreciation to many people who have positively influenced my journey in many ways.

I fall short of words when it comes to expressing my gratitude to Prof.(Dr.) Mini M.G., my research supervisor for her excellent guidance and continuous support throughout these years. With her inspiring words and unbeatable courage in times of distress, I am extremely blessed to have her as my supervisor.

I express my sincere thankfulness to Prof.(Dr.) V.P. Devasia, Principal, Model Engineering College for his valuable comments and providing all research facilities during my tenure in the institution.

It is a great privilege that I could associate with Prof. (Dr.) John Jose, Assistant Professor, Department of Computer Science and Engineering, IIT Guwahati who is a passionate teacher and researcher. I am filled with extreme gratitude for the selfless help extended by him through enormous hours spent in manuscript corrections and discussions.

I would like to thank Prof. (Dr.) Jayasree V.K., Former Head of Department of Electronics, Model Engineering College for her valuable help. Special thanks to Shri. Jagadeesh Kumar P., Associate Professor in the Department of Electronics for providing me with lab facilities during the initial years. Working with my friends and co-researchers at Model Engineering

College, especially Shri. Anoop T.R., Mrs. Asha R.S. and Mrs. Neethu M. Sasi, has been the most memorable part of these years. Furthermore, I would like to thank all the faculty members and non-teaching staff members of the department for their wholehearted co-operation.

I would like to acknowledge the financial assistance received from University Grants Commission, Government of India under the Maulana Azad National Fellowship scheme during this time.

I am very fortunate to have a loving family to support me through these tough times. I am indebted to my husband and children for giving me enough time to myself so that I could prepare presentations and keep up with paper deadlines. I thank my parents who were always willing to help so that my work is not interrupted. I thank my parents-in-law and all my dear ones for their moral support during these years.

Simi Zerine Sreeba

CERTIFICATE

This is to certify that this thesis entitled ***“Performance and Reliability Enhancement Techniques for Two Dimensional Network-on-Chips using Adaptive Deflection Routers”*** is a bonafide record of the work carried out by Ms. Simi Zerine Sreeba under my supervision in the Department of Electronics Engineering, Model Engineering College, Kochi. The results presented in this thesis or part of it have not been presented for the award of any other degree(s).

I further certify that the corrections and modifications suggested by the audience during pre-synopsis seminar and recommended by the Doctoral Committee of Mrs. Simi Zerine Sreeba are incorporated in this thesis.

Cochin – 682021.
04-12-2017

Dr. Mini M.G.
Supervising Guide

ABSTRACT

Advancements in the field of silicon process technology unveil the possibility of fabricating billions of transistors into a small chip area. Future Multi-Processor System-on-Chips (MPSoC) make use of this new trend by integrating hundreds and thousands of processing cores into a single silicon substrate. With increase in the number of on-chip processing elements, there is an ever increasing demand for a scalable and modular on-chip communication system. Network on Chip (NoC) emerges as the most suitable form of communication fabric for large MPSoCs. Extensive amount of research is being carried out on various aspects of NoCs with focus on achieving higher performance and reliability of on-chip communication. Data is transmitted in the form of packets from one processing core to another by transiting through various NoC components.

One major component of an NoC system is the router, whose architecture and algorithm play a vital role in improving the performance parameters of the NoC. From the review of recent literature on various routing techniques, it is observed that buffer-less and minimally buffered routers employing deflection routing technique deliver good performance and energy-efficiency simultaneously. This dissertation focuses on the study of architectural and algorithmic enhancement methods in deflection routers to improve the performance of two dimensional mesh NoCs. As NoC designs adhere to stringent power budgets, energy efficiency is also considered as an important performance measure in deflection routers. The shrinking feature size of devices fabricated on an MPSoC may cause component failures including NoC, resulting in malfunctioning of the chip. An analysis of the reliability

issues due to permanent faults in two dimensional mesh NoCs and routing techniques to achieve fault tolerance is investigated in this study. Thermal imbalance due to uneven traffic distribution through the mesh network can lead to irregular wear and tear and chip failure in the long run. Hence deflection routing methods with thermal awareness are also undertaken here due to its importance.

An energy efficient deflection routing algorithm to reduce the packet deflection rate using a novel output port selection strategy is proposed in the thesis. Another approach for achieving high performance is by reducing the router delay and flit latency through single cycle router operation, which is proposed as a novel adaptive deflection router. A unique method for fault tolerant deflection routing that tolerates permanent faults in NoC components and delivers graceful performance at high fault rates is also proposed and evaluated. A new faulty router model that enhances network performance under high fault rates is also put forward. Through various evaluations, it is shown that the newly proposed methods outperform the state-of-the-art techniques in terms of network parameters and reduction of dynamic power consumption. The issue of uneven thermal distribution within a NoC is addressed by suggesting a deflection routing technique that re-routes some packets to regions with lesser traffic density.

Keywords : Network on Chip, Deflection routing, Performance parameters, Permutation Deflection Network, Average latency, Dynamic power dissipation, Fault tolerance, Reliability, Faulty router model, Energy efficient routing, Critical path length, Thermal variance.

CONTENTS

List of Figures.....	vii
List of Tables.....	xiii
List of Algorithms	xv
List of Equations	xvii
Abbreviations.....	xix
1. Introduction	1
1.1 Network-on-Chip.....	4
1.2 Mesh topology.....	5
1.3 Routing.....	5
1.3.1 Deadlock and livelock avoidance.....	6
1.3.2 Buffered router.....	6
1.3.3 Deflection routing.....	7
1.3.4 Flit format.....	9
1.4 Performance parameters.....	9
1.5 Motivation.....	11
1.6 Objectives of the research.....	12
1.7 Contributions of the Thesis.....	12
1.8 Organization of the Thesis.....	13
2. Related Work	15
2.1 Buffered routing methods.....	17
2.2 Buffer-less routing methods.....	18
2.3 Minimally buffered routing methods.....	21
2.4 Fault tolerant routing methods.....	27

2.5 Thermal aware routing methods.....	33
3. Weighted Deflection Routing.....	35
3.1 Weighted Deflection Buffer-less routing.....	39
3.1.1 Weighted Deflection Count.....	39
3.1.2 Directional Weights.....	40
3.1.3 Livelock avoidance.....	43
3.1.4 Enhanced flit header.....	44
3.1.5 Router architecture.....	45
3.1.6 Simulation methodology.....	50
3.1.7 Results and discussions.....	51
3.1.8 Hardware synthesis.....	59
3.2 Weighted Deflection router with Minimal Buffering.....	69
3.2.1 Router pipeline.....	60
3.2.2 Results and discussions.....	62
3.2.3 Router timing, static power and area.....	68
3.3 Chapter summary.....	69
4. High Performance Adaptive Deflection Routing.....	71
4.1 Router architecture.....	77
4.2 Simulation methodology.....	80
4.3 Results and discussions.....	82
4.4 Router delay, static power and area.....	85
4.5 Chapter summary.....	87
5. Fault Tolerant Deflection Routing.....	89
5.1 Coarse grain fault model.....	96

5.2 Router architecture.....	97
5.2.1 Port allocation using PDN	99
5.2.2 Port reallocation using FTLU.....	99
5.2.3 Port reallocation using latches.....	101
5.2.4 Fault Loop Bit.....	107
5.3 Livelock problem.....	107
5.3.1 Proof of livelock freedom.....	109
5.3.2 Disconnected routers.....	111
5.3.3 Fault patterns with gateway routers.....	112
5.4 Experimental evaluation.....	114
5.4.1 Experimental methodology.....	114
5.4.2 Analysis of network level parameters.....	116
5.4.3 Dynamic link power estimation.....	121
5.4.4 Hardware synthesis.....	123
5.5 Limitations of the coarse grain fault model.....	126
5.6 Enhanced fault tolerant router model.....	127
5.6.1 Router architecture for the enhanced model.....	130
5.6.2 Side buffer parameters.....	131
5.6.3 Simulation methodology.....	135
5.6.4 Analysis of simulation results.....	136
5.7 Chapter summary.....	146
 6. Thermal Aware Deflection Routing.....	 147
6.1 Router Architecture.....	152
6.1.1 Port Reallocation Unit.....	153
6.2 Results and Discussions.....	156
6.2.1 Simulation Methodology.....	156

6.2.2 Thermal profile.....	156
6.2.3 Thermal Variance.....	158
6.2.4 Average latency and throughput.....	160
6.2.5 Real applications.....	161
6.2.6 Hardware synthesis.....	162
6.3 Chapter summary.....	163
7. Conclusions.....	165
7.1 Future scope of the research.....	169
References.....	171
Publications.....	183
Curriculum Vitae.....	185

List of Figures

	Page No.
Figure 1.1 Network on Chip in a 3x3 mesh topology	4
Figure 1.2 Microarchitecture of an input buffered virtual channel router	7
Figure 1.3 Flit format in an 8 x 8 mesh NoC	9
Figure 2.1 Sequential output port allocation in BLESS router	19
Figure 2.2 Micro-architecture of CHIPPER	20
Figure 2.3 Micro-architecture of MinBD router	22
Figure 2.4 Micro-architecture of MinBSD router	24
Figure 3.1 Assignment of Directional Weights	41
Figure 3.2 Variation of WDC with flit injection rates in an 8x8 mesh NoC	43
Figure 3.3 Format of enhanced flit header in an 8 x 8 mesh NoC	44
Figure 3.4 Two stage pipeline architecture of WeDBless router	45
Figure 3.5 Average deflection rates for various synthetic traffic patterns using WeDBless and CHIPPER in 8x8 mesh NoC	53
Figure 3.6 Average latency for various synthetic traffic patterns using WeDBless, CHIPPER and BLESS in 8x8 mesh NoC	54
Figure 3.7 Average throughput for various synthetic traffic patterns using WeDBless, CHIPPER and BLESS in 8x8 mesh NoC	56
Figure 3.8 Percentage reduction in (a) deflection rate (b) latency of WeDBless w.r.t to CHIPPER for real workloads.	57
Figure 3.9 Dynamic power dissipation across links for various synthetic traffic patterns using WeDBless , CHIPPER and BLESS in 8x8 mesh NoC.	59

Figure 3.10	Two stage pipeline architecture of MinBWD router.	61
Figure 3.11	Average deflection rate for various synthetic traffic patterns using MinBWD and MinBD in 8x8 mesh NoC.	63
Figure 3.12	(a) Average deflection rate and (b) Average latency for real applications using MinBWD and MinBD in 8x8 mesh NoC	64
Figure 3.13	Average latency for various synthetic traffic patterns using MinBWD, MinBD and CHIPPER in 8x8 mesh NoC.	66
Figure 3.14	Average latency for various synthetic traffic patterns using MinBWD, MinBD and CHIPPER in 4x4 mesh NoC	67
Figure 4.1	PDN structure of MinBSD.	75
Figure 4.2	Architecture of the proposed HiPAD router.	77
Figure 4.3	PDN structure of HiPAD router.	78
Figure 4.4	Average latency for various synthetic traffic patterns using HiPAD, MinBSD and MinBD in 8x8 mesh NoC.	82
Figure 4.5	Traffic Fractioning based on (a) Buffer Occupancy (b) Average Latency for HiPAD and MinBSD in 8x8 mesh NoC.	83
Figure 4.6	Average deflection rate for various synthetic traffic patterns using HiPAD and MinBSD in 8x8 mesh NoC.	84
Figure 4.7	Comparison of (a) Average latency and (b) Average Deflection rate for real applications using HiPAD and MinBSD in 8x8 mesh NoC.	86
Figure 5.1	PDN in FaFNoC router with (a) faulty north port (b) faulty east port.	92

Figure 5.2	Demonstration of path traversed by flit from source (src) to destination (dst) in a 4x4 mesh NoC using Maze routing and proposed work.	94
Figure 5.3.	Coarse grain fault model.	97
Figure 5.4	Output stage of the proposed router.	98
Figure 5.5	Reallocation of output ports in routers with (a) one faulty port (north) using permute and (b) one faulty port (north) using swap circuit and latch (c) two faulty ports (north and west) and (d) three faulty ports (north,south and east)	105 106
Figure 5.6	(a) Livelock free routing in an 8x8 mesh with multiple faults in partially distributed locations (b), (c), (d) fault pattern dividing the network into two regions R1 and R2 connected by a gateway router, G.	113
Figure 5.7	Average throughput Vs. Fault rate under various synthetic traffic patterns for the proposed work, FaFNoC and Maze routing in 8×8 mesh NoC.	118
Figure 5.8	Average hop count Vs. Fault rate under various synthetic traffic patterns for the proposed work, FaFNoC and Maze routing in 8×8 mesh NoC.	119
Figure 5.9	Average latency Vs Injection rate (uniform traffic) for various fault rates for the proposed method, FaFNoC and Maze routing in 8×8 mesh NoC.	120

Figure 5.10	(a) Average Latency and (b) Deflection for 10% fault rate under SPEC CPU 2006 benchmark applications mixes for the proposed method, FaFNoC and Maze routing in 8×8 mesh NoC.	122
Figure 5.11	Link power Vs Fault rate under various synthetic traffic patterns for the proposed method, FaFNoC and Maze routing in 8×8 mesh NoC.	124
Figure 5.12	Representation of a 4x4 mesh NoC using enhanced fault tolerant router model.	128
Figure 5.13	Two stage pipeline architecture of Enhanced Fault Tolerant Router	130
Figure 5.14	Average hop count Vs Fault rate under synthetic traffic patterns for the proposed model, conventional model and Maze routing in 8×8 mesh NoC.	137
Figure 5.15	Average number of deflections per flit Vs Fault rate under (a) Uniform (b) Transpose traffic patterns for the proposed model and conventional model in 8×8 mesh NoC.	139
Figure 5.16	Average flit latency Vs. Fault rate for uniform traffic pattern in 4×4 , 8×8 , 10×10 and 12×12 network sizes.	140
Figure 5.17	Dynamic Link Power (microwatts) Vs. Fault rate under Various synthetic traffic patterns for the proposed model, Conventional model and Maze routing in 8×8 mesh NoC	142

Figure 5.18	Average latency under various SPEC Benchmark application mixes for the proposed model, conventional model and Maze routing in an 8×8 mesh NoC with 10% and 30% fault rates.	143
Figure 5.19	Average deflection rate of flits for SPEC Benchmark applications in an 8×8 mesh NoC with 10% and 30% fault rates.	144
Figure 6.1	Thermal Profile Graph for CHIPPER in an 8×8 mesh NoC using synthetic traffic patterns at pre-saturation injection rate.	151
Figure 6.2	Two stage pipeline diagram of the proposed thermal aware deflection router.	152
Figure 6.3	Structure of Port Reallocation Unit (PRU)	153
Figure 6.4	Port reallocation using PRU	155
Figure 6.5	Thermal Profile Graph for an 8×8 mesh NoC with the proposed router using synthetic traffic patterns at pre-saturation injection rate.	157
Figure 6.6	Normalised variance using synthetic traffic patterns for the proposed router w.r.t. CHIPPER in 8×8 mesh NoC	159
Figure 6.7	Average latency using synthetic traffic patterns for the proposed router and CHIPPER in 8×8 mesh NoC	160
Figure 6.8	Average throughput using synthetic traffic patterns for the proposed router and CHIPPER in 8×8 mesh NoC	161

Figure 6.9 Normalised variance using applications from
SPEC CPU 2006 benchmark suite for the proposed
method w.r.t. CHIPPER in 8x8 mesh NoC

162

List of Tables

		Page No.
Table 2.1	Comparison of on chip routing techniques	24
Table 2.2	Comparison of fault tolerant routing methods	30
Table 3.1	Directional Weights of a router's output ports for various positions of destination router	41
Table 3.2	Current and pre-computed DWs of a flit	49
Table 3.3	Various proportions of SPEC CPU 2006 applications in benchmark mixes M1 to M6.	51
Table 3.4	Comparison of pipeline delay, area and static power of CHIPPER and WeDBLESS	59
Table 3.5	Comparison of pipeline delay, area and power of MinBD and MinBWD	68
Table 4.1	Benchmark mixes (M1 to M7) in various proportions of SPEC CPU 2006 applications	81
Table 4.2	Comparison of router delay, area and static power of MinBSD and HiPAD routers	85
Table 5.1	Various proportions of SPEC CPU 2006 applications in benchmark mixes M1 to M6.	107
Table 5.2	Router delay, static power and area for Maze router, FaFNoC and proposed method normalised w.r.t CHIPPER.	125
Table 5.3	Example of flit movement through major functional blocks of router R10 in 5 consecutive cycles with side buffer size = 4 and BTI = 3 cycles.	133
Table 6.1	Mixes (M1 to M7) of various MPKI applications	

from SPEC CPU 2006 benchmark suite.

148

List of Algorithms

	Page No.
Algorithm I Decision Making in HiPAD router	80
Algorithm II Port Reallocation using FTLU	102
Algorithm III Function of a Permuter Block of PDN	134
Algorithm IV Flit Buffering by Buffer Eject Block	134

List of Equations

	Page No.
Equation 1.1 Total latency of a flit	9
Equation 3.1 Link Activity Factor	58
Equation 4.1 Latency of a flit through a network	76
Equation 5.1 Fault rate	115
Equation 6.1 Thermal Variance	158
Equation 6.2 Average	158

Abbreviations

ARIADNE	Agnostic Reconfiguration In a Disconnected Network Environment
BIST	Built-in Self-Test
BLESS	Buffer-less
BOFAR	Buffer Occupancy Factor Based Adaptive Router
BTI	Blocking Time Interval
CBT	Core Buffer Time
CHIPPER	Cheap Interconnect Partially Permuting Router
DeBAR	Deflection Based Adaptive Router
DW	Directional Weight
ERF	Ejection Ready Flit
ERR	Ejection Ready Register
FaFNoC	Fault Aware Flits Network on Chip
FBP	Free Buffer Priority
FIFO	First-In-First-Out
FLB	Fault Loop Bit
Flit	Flow control unit
FTDR	Fault Tolerant Deflection Router
FTDR-H	Fault Tolerant Deflection Router Hierarchical
FTLU	Fault Tolerant Logic Unit
GCA	Global Congestion Awareness
GLB	Global Load Balancing
HiPAD	High Performance Adaptive Deflection Router

ITRS	International Technology Roadmap of Semiconductors
LAF	Link Activity Factor
LBDR	Logic Based Deflection Routing
MinBD	Minimally Buffered Deflection router
MinBSD	Minimally Buffered Single cycle Deflection router
MinBWD	Minimally Buffered Weighted Deflection router
MD	Minimal and Defect Resilient routing algorithm
MOE	Minimal Odd-Even
MPKI	Misses Per Kilo Instructions
MPSoC	Multi Processor System on Chip
MSHR	Miss Status Handling Registers
NARCO	Neighbour Aware turn-model based fault tolerant routing
NIC	Network Interface Controller
NoC	Network on Chip
PCA	Proximity Congestion Awareness
PDN	Permutation Deflection Network
PE	Processing Element
PEF	Performance Energy Fault-tolerance
RCA	Regional Congestion Awareness
RPU	Route Pre-computation Unit
SBOT	Side Buffer Occupancy Time
SCARAB	Single Cycle Adaptive Routing and Buffer-less Network
SLIDER	Smart Late Injection Deflection Router
TOSR	Tunnelled Overlapped Static Reconfiguration
TPG	Thermal Profile Graph
VC	Virtual Channel
VCR	Virtual Channel Router

WeDBless Weighted Deflection Buffer-less Router
WDC Weighted Deflection Count

Chapter 1

Introduction

Abstract

This chapter commences with an introduction of the evolution of NoC as the communication medium for future multiprocessor MPSoCs. The building blocks of a NoC are described in brief and a broad classification of various routing techniques is explained. With a brief account of the parameters used for performance evaluation of NoCs, the objectives and main contributions of the research are also stated in this chapter.

The recent progress in VLSI technology and shrinking transistor geometries enable billions of such devices to be integrated on to a small silicon area. Each new generation of Multiprocessor System-on-Chip (MPSoC) takes after this trend to incorporate several hundreds of computing cores in a substrate. Homogeneous MPSoCs, also known as chip multiprocessors, consist of multiple numbers of similar processing elements. Each of these cores runs computation-intensive applications which demand huge amount of data transfer through the interconnecting communication medium. The previous generation of MPSoCs used the conventional shared bus for on chip communication because of its low cost and simple characteristics. In a shared bus, only one master can utilize the bus at a time which means that all the bus accesses should be serialized by an arbiter. As the number of processor cores increases, this mode of communication faces scalability issues due to large number of bus requests. The increase in bus length causes additional wire delay. The new trend in MPSoCs demands a paradigm shift from computation centric to communication centric designs [1]. In this context, Networks on Chip (NoC) communication is gaining widespread popularity due to its numerous advantages like scalability and modular topology which interconnects the processing elements [2]. The issue of global on-chip wire delay is tackled in NoCs by replacing them with short wires which transfer data between various processing elements. The biggest advantage of NoC is that it enables communication between multiple pairs of computing cores simultaneously.

1.1 Network on Chip

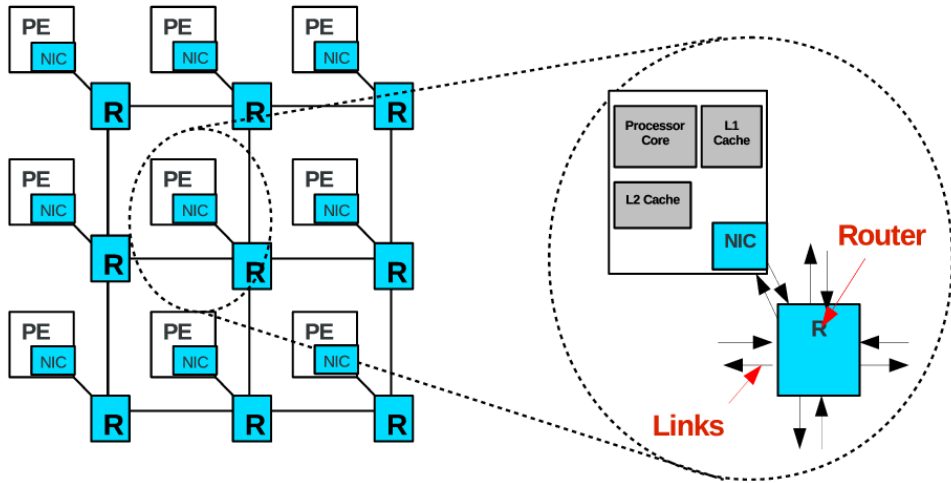


Figure 1.1 Network on Chip in a 3x3 mesh topology

Figure 1.1 shows a sample NoC with 9 Processing Elements (PE) structured as a 3x3 mesh. The main building blocks of an NoC are Network Interface Controllers (NIC), routers (R) and links between the routers. Each processing core is associated with a NIC which connects to a switching element called router. Each PE has a private L1 cache and a shared L2 cache slice. Whenever data is not found in the internal caches, a cache miss is generated and the PE requests for the data from external memory. In this case, data request packets are initiated into the network. Cache coherence transactions may also cause traffic to be injected into the NoC. In NoC, information is exchanged between various nodes in the form of packets. Typical packets are small control messages such as cache block read requests

and larger data packets contain cache block data. The NIC converts the data from the processor's native format to the NoC's packet format and vice versa.

1.2 Mesh topology

A two dimensional mesh is the most widely used topology for connecting PEs in an MPSoC. Each PE is connected to its local router by means of the NIC. In a mesh NoC, routers are interconnected using full duplex bidirectional links with a set of wires in each link direction. As shown in Figure 1.1, a router in a central location of the mesh is connected to its neighbouring routers in north, south, east, west directions and to the local PE using five pairs of input-output ports and bidirectional links. Routers at the edges and corners of the mesh have lesser number of ports and links.

1.3 Routing

Routing is the process of selecting optimal paths for a packet from its source to destination node. Router is considered as the most vital component which is responsible for implementing methods for flow control and congestion control in an NoC. The performance and cost of a router is primarily dependent on its microarchitecture. The routing algorithm is realized using various functional units arranged in a pipelined structure inside the router. The degree of adaptability of a routing algorithm is determined by its capability to forward packets along the network in a distributed manner. A major challenge faced by researchers is to design routing algorithms that deliver high network performance and energy efficiency in compliance with the power and area constraints of the chip. Efficient routing protocols should also ensure reliable

delivery of packets to their destinations when the default routes fail. Uniform traffic distribution is yet another aspect to be considered by the routing function to prevent uneven hotspots in specific chip areas such as the centre of the chip.

1.3.1 Deadlock and livelock avoidance

Deadlock is a state when packets wait for each other in a circular fashion but none of the packets are able to move in the network. Livelock refers to a state in which the packets are able to make progress but they are unable to reach their destination, thereby flowing in the network without getting ejected to their destinations. The routing algorithm should ensure that packet transmission through the network is free from deadlock and livelock.

1.3.2 Buffered router

Conventional routers employ a large number of buffers at the input ports so that packets belonging to different packet transmissions can proceed simultaneously through the same physical channel [3, 4]. In a Virtual Channel Router (VCR), each physical channel is shared by several virtual channels. The microarchitecture of a conventional VCR is shown in Figure 1.2. Packets that cannot be forwarded immediately through desired output ports of a router are temporarily held in FIFO buffers at the input ports till they can proceed further. Usage of these buffers prevents unnecessary wastage of link bandwidth and increases the network saturation point. But these buffers consume significant amount of dynamic power when active and static power when idle [5, 6]. In the Intel Terascale 80-core chip, 28% and for MIT RAW, 36% of chip power is consumed by the NoC [7, 8]. In the TRIPS prototype chip, input buffers of the

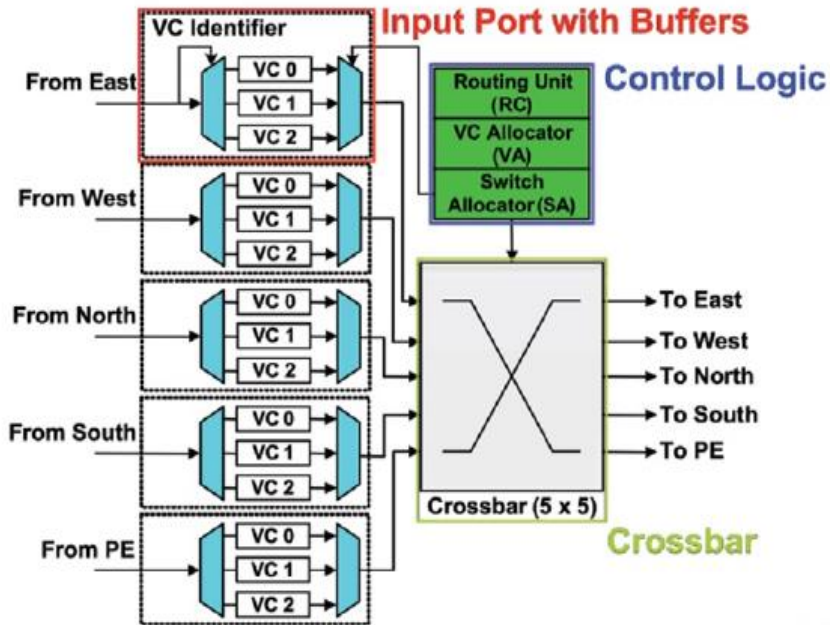


Figure 1.2 Microarchitecture of an input buffered virtual channel router [4].

routers were reported to occupy 75% of the total on-chip network area [9]. Hence buffer size is an important parameter that affects area, power and performance of NoCs.

1.3.3 Deflection routing

VCRs provide an overprovisioned buffer size to accommodate the worst case network traffic. Unlike synthetic traffic patterns used for experimental purposes, packet injection rate is found to be very low for real applications;

hence buffer utilization is very less [10]. In buffer-less deflection routers, the input buffers used in VCRs are completely eliminated and packets are forwarded using deflection routing mechanism [5], [11]. The concept of buffer-less deflection routing has gained wide acceptance as it is capable of delivering performance similar to its buffered counterpart along with significant reduction in power consumption. Packets are split up into flow control units called flits [3]. A router receives flits coming from north, south, east and west directions and also from the processing element connected to it. Flits are introduced into the network from the local processing core through the injection process and flits are removed from the network into its destined processing core by the ejection process. In buffer-less deflection routers, when more than one incoming flit competes for the same output port, one wins and traverses through the desired output port, the others are assigned non-productive ports i.e. such flits are deflected. A necessary condition for buffer-less deflection routing is that the router should have equal number of input and output ports. This ensures that all the flits entering the router move out through the output ports at the end of the router pipeline. Since buffers are not used, this routing mechanism saves power and chip area significantly.

Movement of flits in unproductive directions through the links is termed as deflection. The deflection rate of flits becomes very high at high traffic injection rates and this leads to early saturation of the network compared to buffered routers. As an effort to reduce the flit deflection rate, minimally buffered routers use a small side buffer to store a fraction of the deflected flits [10]. This arrangement helps to achieve performance levels similar to that of NoCs with buffered routers. At the same time, the area and power consumed by the side buffers is negligible compared to that of VCRs. In this dissertation, the

various factors affecting the performance and energy efficiency of deflection router based NoCs is being investigated.

1.3.4 Flit format

The source and destination nodes of a flit are determined by the PE from which it originates. Each flit is routed independently i.e. routing information such as the source and destination addresses are included in the header part of the flit. The typical format of a flit in an 8 x 8 mesh NoC is shown in Figure 1.3. The number of bits that are physically transmitted simultaneously through a NoC link determines the link width. In deflection routers, the link width is equivalent to the number of bits of a flit.



Figure 1.3 Flit format in an 8 x 8 mesh NoC

1.4 Performance parameters

Packet latency is the most frequently used parameter for measuring the performance of an NoC. Average latency is defined as the average number of clock cycles (average time) taken by flits to reach from source to destination. In a deflection router based NoC, total latency of a flit is given by equation 1.1.

$$L_T = L_I + L_R + L_L \quad \text{-- (1.1)}$$

where

- L_T - Total latency of a flit
- L_I - Flit Injection latency
- L_R - Latency at the routers
- L_L - Latency at the NoC links

The flit injection latency, L_I is the time for which flits generated by a PE wait in its core buffer before being admitted into the network. Injection latency increases with network congestion, since flits have to wait for longer time in the core buffer to enter into the network through a vacant flit channel [12]. In NoCs using deflection routing mechanism, latency of a flit through a link is one cycle. Hence, total latency through the links, L_L is equal to the number of hops made by the flit during its traversal.

Each hop through the link either results in a progressive step in the direction of the flit's destination or a deflection in an unwanted direction. The latency experienced by flits in the links can be optimized by reducing these deflections using smart and adaptive routing techniques. Hence, average deflection rate and hop count of flits are also considered as important performance criteria for deflection router based NoCs. The traversal of flits through the links leads to dynamic power dissipation which is directly proportional to the average hop count in the network. An NoC should be designed within specific power limits and minimization of dynamic power wastage across the NoC links due to flit deflections helps to achieve better energy efficiency.

Another commonly used performance parameter is average throughput which is defined as the rate at which packets are delivered by the network per router. For a buffer-less deflection router based NoC, the throughput of the network is equivalent to the packet injection rate. Increase in the injection rate leads to growing congestion in the network and the network subsequently moves into a saturation state. Beyond saturation level, the network congestion is extremely high and flit injection latency rises up. Accordingly, the rise in average throughput slows down and approaches a steady value..

Performance enhancement can also be achieved by optimizing the router latency, L_R . By smart allocation of output ports inside a router, flits can be routed through less congested and minimal paths to their destinations. Router latency can also be minimized by using routers with single cycle delay.

1.5 Motivation

With performance enhancement and increase in complexity of multi core systems, the power consumption also increases linearly [1]. Routing protocols have a significant impact on performance and power of NoC based MPSoCs. Buffer-less routers with deflection routing is used in NoCs to reduce the power consumed by the traditional input buffered VC routers. Though a few efficient deflection routers have been proposed previously, their architectural and algorithmic limitations result in heavy deflection and increased latency of flits in the network. As a result performance is throttled and dynamic power is wasted. Hence there is a high demand for developing high performance and energy efficient deflection routing techniques. Reliability of the chip due to permanent failure of NoC components is another issue which needs to be researched.

Deflection routers need to incorporate fault tolerant logic which minimise power and area overheads while delivering graceful performance. Another area of concern is the non-uniform thermal distribution across the chip arising from uneven network traffic which affects the long term reliability of the chip. Effective solutions to the performance and reliability issues in deflection router based NoCs are proposed in this thesis.

1.6 Objectives of the research

Efficient deflection routing mechanisms that maximize the performance metrics while providing high adaptability to network congestion, high fault resilience and an even thermal distribution are most desirable. The various aspects that limit the performance of two dimensional mesh Networks-on-Chip are investigated in this thesis and architectural and algorithmic techniques are developed which aim at enhancing its

- Performance parameters
- Energy-efficiency
- Fault tolerance
- Uniform thermal profile

1.7 Contributions of the Thesis

The main contributions of this dissertation are listed as follows.

- Proposed an energy efficient routing algorithm based on weighed deflection routing that successfully reduces the deflection rate of flits in buffer-less and minimally buffered deflection router based NoCs.
- Proposed an adaptive deflection router with single cycle delay that achieves higher speed of operation and reduced deflection rates.
- Proposed a fault tolerant technique using deflection routers in two dimensional mesh NoCs which delivers significantly high network performance for high fault rates in NoC components. An enhanced model of a faulty router that improves the availability of fault-free links in the network is also developed.
- Developed an adaptive deflection routing technique that re-routes deflected flits away from the centre of the mesh network to obtain an even thermal distribution throughout the NoC.

1.8 Organization of the Thesis

Chapter 1 presents an introduction to the thesis along with the main objectives, background concepts and author's contributions.

Chapter 2 provides a systematic review of previous research on various routing methods with emphasis on adaptive deflection routing techniques for high performance and energy efficiency. It also discusses literature related to fault tolerance mechanisms adopted in NoC routers. Various load balancing techniques and congestion aware routing methods are also presented here.

Chapter 3 describes a novel output port selection strategy in NoC routers using a weighted deflection routing. The proposed method is evaluated on buffer-less and minimally buffered deflection router based NoCs of various sizes. The impact of the new scheme on NoC performance parameters and dynamic power dissipation are analysed by comparison with state of the art techniques.

Chapter 4 analyses the factors that limit the performance of a recent single cycle routing technique. An new architectural modification is proposed to overcome identified problems like structural isolation. Experimental analysis of the proposed scheme and comparison with earlier work is provided in the latter half of the chapter.

Chapter 5 presents a novel fault tolerant deflection router. The proposed architecture and routing algorithm is explained in detail followed by the experimental evaluation of the proposed scheme at various fault rates. An enhanced fault tolerant router model is also explained with evaluation results..

Chapter 6 presents a thermal aware routing scheme to attain uniform thermal profile across the mesh NoC using deflection re-routing method. Analysis of experimental results of the proposed method is also conducted and compared with the baseline methods at the end.

Chapter 7 concludes the thesis with a summary of the proposed techniques and a brief account of the potential for future research.

Chapter 2

Related Work

Abstract

This chapter presents a detailed review of previous literature related to router architectures and routing algorithms for NoCs. Beginning with buffered routing techniques, various deflection routing methods for achieving high performance and energy efficiency are discussed. Wide range of techniques that incorporate fault tolerance feature in deflection router based NoCs are studied. Various adaptive routing techniques for congestion avoidance and load balancing are also reviewed.

The main attributes of an NoC are topology, routing and flow control. Efficient methods in each of these aspects are desirable in achieving high network performance and energy efficiency. Routers are considered as the most vital components of an NoC as they incorporate various logic blocks for implementing routing algorithms and flow control methods. This chapter presents a comprehensive survey of various state-of-the-art routing techniques for performance and reliability enhancement in two dimensional NoCs.

NoC routers are broadly classified into buffered and buffer-less domains.

2.1 Buffered routing methods

Buffered routers are the conventional routers with buffers at the input side. The concept of multiplexing a single physical channel over several logical channels with independent buffer queues is implemented using the concept of virtual channels. Packets belonging to different transmissions occupy different virtual channels and make progress through the router by sharing the physical channels [4]. Wormhole routing is a flow control method in which only the first flit of each packet contains the header information (head-flit) and all subsequent flits simply follow the preceding flit [3]. This method is largely adopted in VC routers because of minimal buffer requirements and good channel utilization. A flow control scheme which utilises the existing pipelined channels as storage in place of explicit virtual channel buffers is proposed in [13]. A routing approach based on virtual circuits that can be established and broken dynamically according to the network conditions is proposed in [14]. Despite the capability of buffered routers to deliver high network performance, buffers are subject to high energy dissipation both in active and idle states [5].

Buffers also consume significant portion of the router area. Some of the past works mitigate buffer size and maximise its utilization by novel methods like dynamic buffer utilisation and leakage aware buffers with power supply gating for unused buffers [15, 16].

A routing algorithm consists of the logic that selects an output port for a flit arriving at the router's input on the basis of the destination address contained in the flit header. In a deterministic routing algorithm, the routing path between a pair of nodes in the network is fixed. One of the most commonly used deterministic routing algorithms for mesh topology is XY routing, where a packet first moves along the row and then along the column to reach the destination [17]. Adaptive routing algorithms, on the other hand, choose alternate paths for packets if original paths in the network are congested. Adaptive methods like dynamic XY routing are more suitable for attaining improved network performance while tolerating failures in routers and links [18, 19]. Other metrics like free buffer slots or virtual channels are also used by adaptive algorithms for congestion aware routing [20-23]. Another aspect of NoCs with buffered routers is to ensure deadlock and livelock safety of flits. Minimal Odd Even routing (MOE) is a popular deadlock free routing algorithm that prohibits flits from making certain turns during their hop to destination [24].

2.2 Buffer-less routing methods

Buffer-less routers have evolved as an energy efficient solution for on chip networks [5]. SCARAB is one of the first generation routers that handle the port allocation problem without using any buffers [25]. It uses the method of

dropping and re-transmission of packets which are unable to obtain an output port in a productive direction. A separate circuit switched network is maintained to send negative acknowledgements about the discarded packets. As the packets should anticipate retransmission, they are stored in the Miss Status Handling Registers (MSHR) of the processor till a positive acknowledgement is received. Due to high retransmission overheads and requirement of separate networks for data packets and acknowledgements, this approach is not suitable for simple, low power NoC. Deflection routing is the most popular mechanism used in buffer-less routers as it does not incur such overheads.

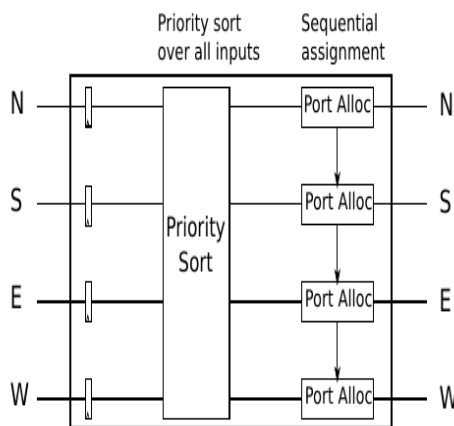


Figure 2.1: Sequential output port allocation in BLESS router [5].

A buffer-less router, BLESS proposes a deflection routing mechanism in which flits arriving at the router's input ports are allocated to one of the output ports after passing through various functional blocks inside the router [5]. Flits are sorted on the basis of age and port allocation unit in the router allocates the output ports in oldest first order. The schematic diagram in Figure 2.1 depicts

the sequential output port allocation in BLESS. As a result of sequential port allocation, the critical path length of the router increases. In order to accommodate the lengthy router operation, a wider clock pulse is required. This results in lowering of the operating frequency of the NoC.

A superior deflection router architecture is that of CHIPPER (Cheap Interconnect Partially Permuting Router) which employs parallel port allocation of flits, thereby reducing its critical path length [11]. The router pipeline of CHIPPER is shown in Figure 2.2. The output port allocator, which is referred to as the Permutation Deflection Network (PDN), consists of four permuter blocks arranged in two stages, each stage having two blocks. P1 maps flits from the north and east inputs to one of its output lines connected to P3 or P4. P2 does the same operation in parallel for flits from south and west input lines. Permuters P3 and P4 connect each of the input flits from P1 and P2 to one of the four output ports. Even though parallel operation makes CHIPPER faster compared to BLESS, it exhibits higher flit deflection rate due to the inefficiency of the flit prioritization scheme.

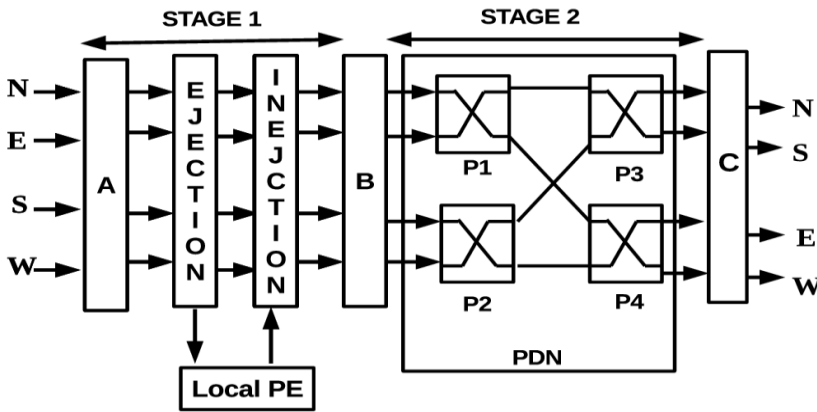


Figure 2.2: Micro-architecture of CHIPPER [11].

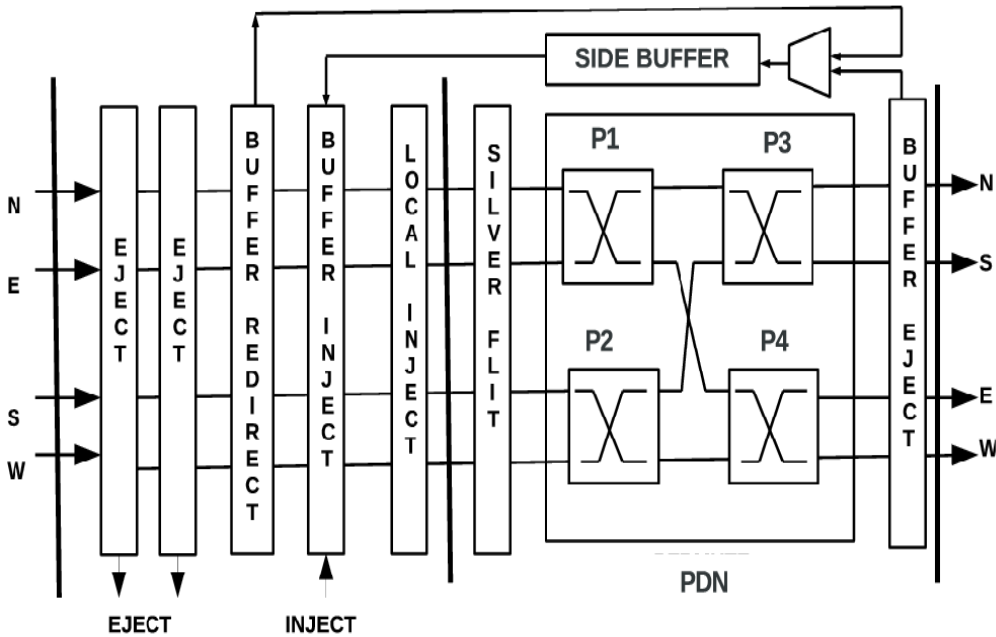
The logic functions in BLESS and CHIPPER are divided into two stages and each stage has a delay of one clock cycle; hence the total delay of flits in each router is two clock cycles. A deflection routing for hierarchical mesh NoCs that uses a high radix crossbar to reduce the number of routers is presented recently [26].

Deflection routing is inherently free from deadlock since the router's resources are not withheld by flits for more than one cycle. However, routing algorithms need to incorporate sufficient mechanisms to guarantee livelock freedom of flits. Sequential allocation of output ports to flits in oldest first order is used for resolving livelock issue in BLESS [5]. This technique ensures that the older flits in the network reach their destinations early. CHIPPER proposes a scheme which marks a flit as the highest priority flit in the entire network [11]. This flit which is termed as the golden flit is guaranteed to occupy productive output ports in each router and reach its destination through the shortest path. Then the golden token is passed to another flit that is next in age order in the network.

2.3 Minimally buffered routing methods

MinBD (Minimally Buffered Deflection router) is the first router to use side buffering technique to overcome the disadvantage of CHIPPER i.e. high deflection rate [10]. The microarchitecture of MinBD router shown in Figure 2.3 has functional units similar to that of CHIPPER arranged as two pipeline stages. The buffer eject unit at the end of the router pipeline picks one of the deflected flits from the output channels of the PDN in each cycle and stores it in a side buffer temporarily. The buffered flits re-enter into the router pipeline

in a subsequent cycle through any of the vacant input channels following first-in-first-out (FIFO) policy. The size of the side buffer is insignificant compared to the virtual channel buffers; hence they cause very less impact on router power and area. For improving the efficiency of the livelock safety mechanism, MinBD uses a local flit priority in addition to the golden prioritisation scheme introduced in CHIPPER. One among the four flits in the input channels of a MinBD router is marked as high priority flit (silver flit), which is given preference to obtain an output port of its choice during port allocation by the



PDN.

Figure 2.3: Micro-architecture of MinBD router [10].

DeBAR uses a minimal central buffer pool to store some of the misrouted flits [27]. It dynamically controls the flit injection from local core and the central buffer pool to ensure fairness and exhibits promising results compared to MinBD. In both MinBD and DeBAR routers, injection of flits from the local processor and side buffer takes place before output port allocation. Here, there is a path for the newly injected flits to move to the side buffer. SLIDER proposes a router architecture in which injection occurs after port allocation [28]. It reduces the problem of channel wastage and intra-router interval occurring in MinBD and DeBAR. Another router with buffer-less wormhole routing which uses a register array to store flits temporarily is proposed in [29]. The study of various NoCs using deflection routing comes out with the opinion that a priority based deflection policy which uses global or history related criteria is most suited for boosting the performance of networks [6]. MinBD, DeBAR and SLIDER incur a pipeline delay of two clock cycles.

As mentioned in equation 1.1, a method for improving the performance of NoC is by reducing the router delay. Some deflection routers achieve high performance by completing the routing operation in a single clock cycle [25],[30]. A recently proposed work in this domain is Minimally Buffered Single Cycle Deflection router (MinBSD) [30]. The microarchitecture of MinBSD router is shown in Figure 2.4. The length of the router pipeline is reduced by parallelisation of independent operations like route computation (RU) and flit prioritisation (PU). MinBSD uses a derivative of the two stage PDN proposed in CHIPPER, each stage having three permuter blocks to interconnect the input ports, two side buffers, SB and CB and four output ports. The two side buffers store a minimum number of misrouted flits and one of them also stores flits generated by the local processing core.

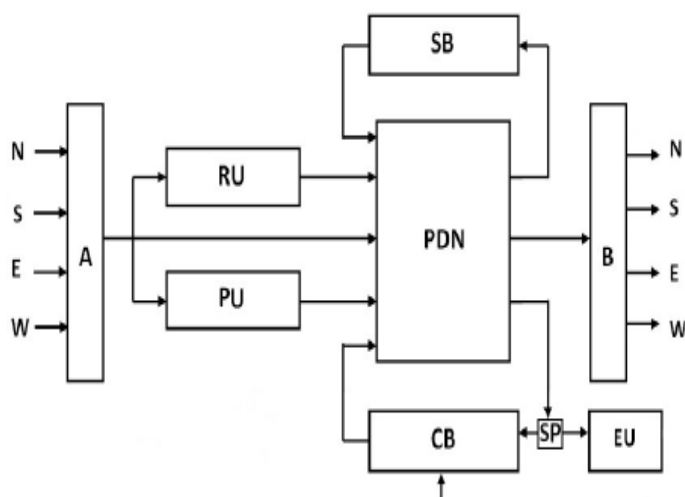


Figure 2.4 Micro-architecture of MinBSD router [30].

Table 2.1 Comparison of on chip routing techniques

Proposed Method	Routing method	Router delay (cycles)	Output port allocation	Merits	Limitations
Wormhole router [3]	Wormhole routing	4	Crossbar switching	Channel bandwidth is flit size, Reduced buffering	Head of line blocking, deadlock occurrence
VCR [4]	Wormhole routing	4	Crossbar switching	Deadlock free channel sharing among various packets	Each virtual channel requires buffer space

BLESS [5]	Buffer-less deflection routing	2	Sequential	No input buffers, low static and dynamic power	Low performance at high injection rate, Low speed of operation
CHIPPER [11]	Buffer-less deflection routing	2	Parallel using PDN	Fast operation	High deflection rate and dynamic power dissipation
MinBD [10]	Minimally buffered deflection routing	2	Parallel using PDN	Low deflection rate and high throughput	Inefficient flit prioritisation
MAS [29]	Buffer-less wormhole routing	2	Crossbar switching	Lower latency and receiver side buffering compared to BLESS-Worm	Buffers required in routers to store flits temporarily.
DeBAR [27]	Minimally buffered deflection routing	2	Parallel using PDN	Minimal central buffer pool, ensures fairness to flits from buffer, higher performance than MinBD	Channel wastage, unnecessary internal flit movement

SLIDER [28]	Minimally buffered deflection routing	2	Parallel using PDN	Reduced channel wastage and intra-router interval, higher performance than DeBAR	Requires two cycles to complete operation
SCARAB [25]	Buffer-less packet dropping	1	Parallel allocator	High frequency of operation	Acknowledgement network and packet retransmission overheads
MinBSD [30]	Minimally buffered deflection routing	1	Parallel	High Speed, no dropping of packets	Poor structural connectivity Penalisation of high priority flits

Table 2.1 summarises the merits and demerits of various on-chip routing techniques. More emphasis is given to the study of deflection routing techniques because of their utility in energy efficient interconnect design. Studies reveal that architectural and algorithmic modifications to the existing bufferless baseline, CHIPPER and minimally buffered MinBD can help in further enhancing the performance parameters of the NoC.

2.4 Fault tolerant routing methods

A wide range of routing techniques for the detection, diagnosis and tolerance of faults in two dimensional NoC systems has been analysed. Two types of faults may occur in NoCs, transient and permanent. Transient faults (soft errors) are caused due to unpredictable reasons like voltage induced delay, radiation induced fluctuations or crosstalk [31]. These types of faults are mostly handled using flow control mechanisms and error control coding schemes like cyclic redundancy check and parity codes [32]. An end to end error correction and retransmission scheme is recently proposed in [31]. Another recent work integrates error detection into the input ports of routers and correction of errors is provided at the network interfaces [33]. On the other hand, device wear out or manufacturing defects may cause permanent failure in chips [34]. Online detection of permanent faults using error syndrome collection is presented in [35]. A Built-in Self-Test (BIST) unit is embedded into each router to diagnose the faults in the Vicis NoC, which makes it a costly solution [36].

In general, permanent faults are tolerated by utilizing the redundant structure of NoC and fault tolerant algorithms that route packets around faulty components. Faulty links and routers in NoC are represented using architectural level fault models. Majority of the prior work use a coarse grained fault model which deactivates a pair of bidirectional links corresponding to a fault in a single direction [37, 38]. Both faulty and fault-free routers in the resultant network consist of only bidirectional links which makes it easy to explore at least one deadlock-free route for a flit. Many of the fault-free unidirectional links cannot be utilized for transmitting flits in this model, this

results in shrinking of available bandwidth with increase in fault rates. A fault tolerant NoC with dynamically configurable bidirectional links is used to tolerate static and dynamic faults [39]. In fault models using high levels of abstraction, routers with faulty logic blocks are modelled by disabling all input-output links to it. Vicis improves upon the coarse-grained model by mapping gate-level failures within the router to a fault in one of the bidirectional links [40]. ImmuneNet reserves an escape virtual channel inside each router for packets that encounter faulty links in their path [38]. In some of the recent papers, hardware faults are included with finer granularity and mapped to a corresponding high level abstraction fault [41, 42].

In uDirec, a fine grained fault model is presented, in which a failure in a router's datapath element is mapped to a fault in one of the unidirectional links [41]. Faults in control circuit elements are fatal to router operation; hence such errors are modelled by marking all input-output ports of the router as faulty. Apart from the commonly used performance parameters, the Performance-Energy-Fault tolerance (PEF) metric is used for evaluating a faulty NoC. In uDirec, one of the nodes is designated as a supervisory node which keeps track of the healthiness of routers and links. On detecting a fault, this node runs a reconfiguration algorithm to update the routing tables according to the newly evolved topology. The reconfiguration phase usually consumes many cycles and normal router operation is stalled during this period. uDirec is reported to incur high reconfiguration overhead and large area overhead whereas TOSR achieves fast reconfiguration [41],[43]. Such centralised approaches involve an additional risk of the supervisory node becoming faulty which might cause the entire chip to fail [44 - 46].

On the contrary, distributed methods achieve fault tolerance by local decision making at each router [37], [47-49]. A virtual channel based routing technique with low fault coverage is presented in [50]. Fault-tolerant routing algorithms that provide deadlock freedom based on the turn model are proposed in [47], [51- 54]. A fault resilient algorithm that works on dynamic reconfiguration of routes is presented in [52]. Hermes [53] adopts dimension order routing or up/down routing based on routing tables whereas NARCO [54] uses a region based fault awareness approach for achieving high throughput. A novel fault tolerant routing approach which avoids virtual channels and utilises acyclic channel dependency graph for deadlock avoidance and connectivity is presented in [55].

The coarse-grained fault model is widely used in deflection routers since all routers have equal number of input and output ports. Most of the deflection routers are equipped with built-in fault tolerant logic. In the presence of faults, routing decisions are made locally at each router; hence reconfiguration is very fast [48, 49]. FTDR-H is a deflection routing method with high fault coverage but consumes large area due to routing tables [49]. Some adaptive routing algorithms make routing decisions based on various cost functions like route length, local fault status or two hop fault information [56, 57]. Recently proposed methods employ the PDN proposed in buffer-less CHIPPER [11] and minimally buffered MinBD [10] for output port allocation of flits. In the Fault Aware Flits based NoC (FaFNoC), fault tolerance is achieved by disallowing some connections from active input lines to faulty output ports of the PDN [58]. Maze routing is an algorithm that uses a variant of the conventional face routing proposed for wireless adhoc networks, which randomly allocates output ports for flits whose productive ports are faulty [59, 60]. Routing is performed

using information contained in the flit header, which is updated as the flit hops out of every router. Although maze routing promises 100% fault coverage and guaranteed packet delivery on various topologies and deflection router architectures, majority of the flits are routed through non-minimal paths. The flits also require wider links to transport the extra bits in the flit header.

Maze routing adopts the golden and silver token mechanisms used in conventional CHIPPER and MinBD to resolve livelock while FaFNoC uses an age based hop counter in the flit header which is incremented at every hop [58, 59]. FTDR ensures livelock safety by prioritising older packets and using a converged routing table which ensures that packets advance towards their destinations deterministically [49]. From the study of previous literature, it is inferred that a precise fault model together with an efficient routing mechanism can improve the performance parameters of an NoC.

Table 2.2 Comparison of fault tolerant routing methods

Proposed Method	Fault tolerant routing method	Fault coverage	Reconfiguration	Merits and Limitations
MD [19]	Dynamic XY	low	Offline	Disconnected network as the faults increase
ARIADNE [37]	Adaptive routing table reconfiguration	full	Distributed	Coarse grained fault model, low area overhead, slow reconfiguration, poor scalability

Vicis [40]	Static routing table reconfiguration	high	Distributed	Improved coarse grained model BIST based , high area overhead, deadlock prone,
uDirec [41]	Scoreboard updation at supervisory node	full	Centralised	Fine grained fault model, high reconfiguration overhead
TOSR [43]	Multiple physical networks for different message types	moderate	Distributed	Fast run-time reconfiguration, additional escape network required
Hermes [53]	Routing table updation	moderate	Distributed	High area overhead
NARCO [54]	OE turn model based routing	High	Distributed	High overhead for packet redundancy
Immunet [38]	Routing table	high	Distributed	Coarse grained fault model, high overhead due to escape VC and three additional routing tables
LBDR, uLBDR [45,46]	Logic Based	high	Centralised	Low area overhead

d2-LBDR [44]	Logic Based	moderate	Centralised	Low area overhead
FTDR, FTDR-H [49]	Deflection routing with routing table updation	High	Fully distributed	Fast reconfiguration, handles transient and permanent faults, high flit prioritisation overhead.
FoN [57]	Deflection routing with neighbor fault information	Full	Fully distributed	Fast reconfiguration, low throughput and high hop count
FaFNoC [58]	Deflection routing by disabling faulty links	Full	Fully distributed	Low router area, Less energy efficient due to high deflection rate, high wiring overhead due to age based hop counter.
Maze routing [59]	Deflection routing with additional header information	Full	Fully distributed	Guaranteed delivery, no reconfiguration overhead, non-minimal routing leads to higher latency, high wiring overhead.

From the comparison of various fault tolerant routing techniques given in Table 2.2, it is observed that majority of the fault tolerant methods for permanent faults have been proposed for buffered VCR based NoCs. Reliability issues related to adaptive deflection router based NoCs needs to be addressed through extensive research. Routing techniques that deliver graceful performance even at high fault rates by exploring minimal fault-free paths are to be devised.

2.5 Thermal aware routing methods

Thermal distribution within a chip is an important factor for the long-term reliability of the chip. Thermal hotspots in certain areas of the chip cause uneven wear and tear leading to reduction in average lifetime of the chip. One reason for hotspot formation is due to high power consuming applications running on processor cores. Efficient application mapping techniques are one way of achieving an even thermal profile in the chip [61,62].

Another important aspect is the uneven distribution of NoC traffic due to poor load balancing capability of the routing methods which deliver high performance. In such NoCs, certain areas tend to be more congested than the rest due to concentration of traffic, creating an uneven thermal profile. Many papers related to thermal aware routing techniques were proposed in recent years. In input buffered routers, Regional Congestion Awareness (RCA) method uses congestion information of a region to improve load balancing capability of the network [63]. Global Congestion Awareness (GCA) and Global Load Balancing (GLB) are also similar approaches where information on global congestion is the metric used for load balancing [64, 65]. In the Free

Buffer Priority (FBP) scheme, the count of free input buffers in downstream routers is taken as a measure for adaptive selection of output ports [4]. In Buffer Occupancy Factor based Adaptive Router (BOFAR), the history of buffer occupancy time of flits decides congestion in a router [66].

Another work introduces an aging aware adaptive routing algorithm that routes packets along the paths which experience least congestion and minimum aging stress [67]. Cool Centers follows an output port selection strategy based on prioritising ports that route packets away from the centre of the mesh [68]. Adaptive deflection routers have an inherent load balancing capability, which can be exploited to route packets so as to achieve uniform thermal distribution throughout the NoC. Proximity Congestion Awareness (PCA) is a deflection routing technique, where routers use the load information of neighbouring routers to avoid congested areas while routing [69].

Chapter 3

Weighted Deflection Routing

Abstract

In this chapter, a novel method for output port selection in buffer-less deflection routers based on weighted deflection routing is described. The proposed method delivers superior performance using an adaptive flit prioritisation scheme and route computation method. Benefits of the proposed method are substantiated by comparison of experimental results with the state-of-the-art techniques. The implementation of weighted deflection routing algorithm on a minimally buffered deflection router and its evaluation are presented in the latter half of the chapter.

The research for energy efficient NoCs led to the migration from buffered to buffer-less router architectures. The first deflection router architecture in the buffer-less category is the BLESS router [5]. The average deflection rate, which is a measure of the unproductive hops made by flits, influences the performance and energy efficiency of deflection router based NoCs. BLESS router adopts sequential output port allocation of flits in its input channels, thereby prioritising older flits to choose their desirable ports; hence average deflections rates are very low. At low and medium values of flit injection rates, throughput of this network is identical to that obtained for buffered NoCs. The age based flit prioritization also helps to guarantee livelock freedom in the NoC. However, the sequential dependency in port allocation leads to a very long critical path in the BLESS router, which limits its speed of operation. The performance bottleneck of BLESS is resolved in CHIPPER with a simple, fast and parallel solution for output port selection method and hence it is a more superior router architecture in the buffer-less domain [11].

The salient feature of CHIPPER architecture is the PDN used for output port allocation of flits. The PDN overcomes the speed limitation of BLESS by eliminating the sequential block and performing parallel port allocation of input flits. The four permuter blocks of the PDN (refer Figure 2.2) map the flits from the four input channels to the four output ports of the router in a parallel way, thereby reducing its critical path length. In spite of the architectural superiority of CHIPPER, the average deflection rate and latency of flits is higher compared to BLESS. The drawbacks of the output port selection mechanism in CHIPPER are explained next.

- CHIPPER uses a token based flit prioritisation mechanism for livelock avoidance. One flit (the oldest flit) is marked as the highest priority flit (golden flit) in the entire network. The golden flit is guaranteed to win a productive output port at each router until it reaches its destination and then the priority is passed on to the next flit in age order. This scheme proves to be inefficient since majority (more than 90%) of the flits are delivered without any priority during their network lifetime. In CHIPPER, the conflict for port allocation of flits other than the golden flit is resolved in a random way. Due to lack of efficient output port selection strategy, flits undergo unnecessary deflections and traverse non-minimal paths to destination. This increases the average latency of the network.
- Another drawback of the CHIPPER architecture is a single ejection port, which can eject only one flit in an operation cycle. For low and medium traffic conditions, more than one flit needs to be ejected at a router once in every three cycles. Due to the lack of sufficient ejection mechanism, the flit with highest priority among them is ejected and others are deflected. Minimisation of such deflections would help in significant reduction of dynamic power dissipation, resulting in better energy efficiency of the NoC.
- The golden flit prioritisation scheme requires a flit ranking based on a global clock which is agreed upon by all the routers in the network. The complexity and area overhead associated with this livelock avoidance mechanism is high compared to its benefits.

3.1 Weighted Deflection Buffer-less routing

The work described in this section combines the merits of BLESS and CHIPPER and overcomes their limitations by proposing a novel Weighted Deflection Buffer-less (WeDBless) routing technique. The aim of the new method is to reduce the average network latency and dynamic power dissipation at the links by minimizing the flit deflection rate. The main contributions of the proposed WeDBless routing technique are listed below.

- The golden flit prioritisation based on a global clock is eliminated and a flit ranking based on the Weighted Deflection Count (WDC) of a flit is introduced. Flits with higher WDC win arbitration for output port in a the PDN of a router.
- For each flit, a router's output ports are prioritized on the basis of Directional Weights (DW). During port allocation, a flit attempts to occupy an output port with lowest value of DW.
- Each router employs a small buffer which can store a single ejection ready flit (ERF) in one cycle. This facilitates dual ejection using a single ejection port.

3.1.1 Weighted Deflection Count (WDC)

The WDC is a value contained in the enhanced header of every flit. During output port allocation, when two flits compete for a single output port, the flit with higher WDC wins and progresses through the port and the other is deflected. Initially, when a flit is injected into the network, its WDC is 0. As the

flit progresses from one router to another, its WDC value changes. WDC is incremented if it is deflected through a non-productive output port and decremented if the flit traverses through a productive output port. Due to this mechanism, frequently deflected flits will have high WDC value, which ensures that they attain higher priority to win productive ports in subsequent hops. Since the proposed algorithm adopts flit prioritisation based on the WDC transmitted in the header of each flit, the global clock transmission is avoided.

3.1.2 Directional Weights (DW)

The flit header also contains four Directional Weights corresponding to the four output ports of a router, which represent the routing choice of a flit. The directional weights can take any one of the three values, -1, +1 or +2 which are coded using 2 bits each i.e. 00, 01 and 10 respectively. A total of 8 bits are used to include the four DW values in the enhanced flit header. An output port that directs the flit to the same row or column as the destination router has DW of -1 (least weight) for the flit. DW of a router's output port has value of +1 or +2 depending on its direction towards destination. Table 3.1 shows DW values of a flit for a router's output port based on the direction of deflection of the flit from the destination. During port allocation, a flit competes to occupy the output port with least DW. Assignment of DWs in WeDBless routing is demonstrated using an example shown in Figure 3.1. In the 4x4 mesh NoC, a flit whose destination is at router (2,2) is assumed to arrive at the south input port of the router (1,1). At (1,1), north and east output ports lead the flit in orthogonal direction towards the destination router. Hence DWs in north and east are +1. Since south and west output ports lead the flit away from its destination, DWs for the flit in those directions will be +2.

Table 3.1: Directional Weights of a router's output ports for various positions of destination router

Direction w.r.t destination	DW of Output port	Two bit value in flit header
Same row or column as destination	-1	00
Orthogonal towards destination	+1	01
Away from destination	+2	10

The 8 bit DW values of this flit at router (1,1) are 00, 10, 00, 10 in north, south, east and west directions respectively. After output port allocation is completed, the WDC of the flit is updated by adding DW of the allotted output port to it. In this example, if north or east output port is allotted to the flit, its WDC value gets incremented by +1 and if south or west output port is allotted, WDC gets incremented by a value of +2. Let us assume that the flit took the east output

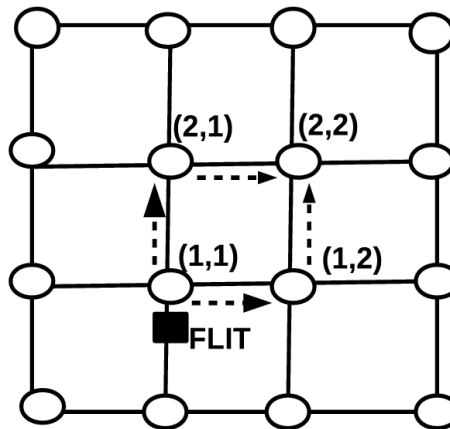


Figure 3.1 Assignment of Directional Weights.

port of (1,1) and moved to router (1,2). Now the new DWs of the flit are computed with respect to the position of the destination router from (1,2). At

(1,2), the north output port is column aligned to the destination (2,2). So DW of north port is -1 for the flit. The DWs of the south, east and west ports are +2, +1 and +1 respectively. During port allocation in router (1,2), if the flit occupies the west port and returns to router (1,1), its WDC is again incremented by +1 due to the addition of DW. In effect, the WDC of the flit is incremented by 2 as it completes a loop from (1,1) to (1,2) and back to (1,1). Higher value of WDC increases the probability of the flit to win its desired output port during the next port allocation.

As a flit gets deflected and moves away from the destination, the WDC value increases due to addition of larger DWs. WDC is decremented by 1 only when a flit is allotted to an output port in the same direction as the destination router. However, the minimum allowable value of WDC is 0 and the maximum value is determined by the number of bits in the WDC field. If a flit having WDC value of 0 traverses in a productive direction, WDC is prohibited from being decremented further. The width of WDC field in the flit header is chosen such that it does not overflow the maximum allowable range. Simulations are performed on an 8x8 mesh NoC with uniform traffic function and the maximum value of WDC attained by a flit for each value of injection rate is observed. Figure 3.2 shows the variation in WDC when flit injection rate is incremented in steps from 0 to saturation level. Since the maximum deflection level is found to be 38, the field width of WDC in the flit header for an 8x8 mesh NoC is taken as 6 bits. The 6 bit WDC can hold a maximum value of 64. As an exceptional case, if the maximum value of WDC is reached, further incrementing of WDC is not allowed and the value is retained. Simulations are repeated using other synthetic traffic patterns and real application traffic for

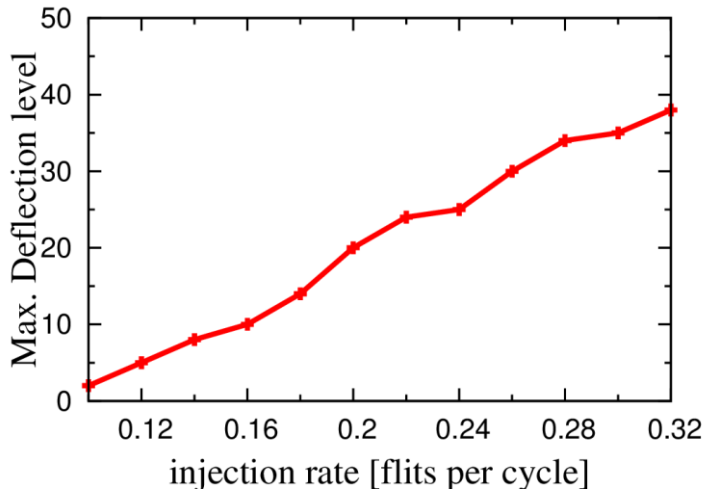


Figure 3.2 Variation of WDC with flit injection rates in an 8x8 mesh NoC.

8x8 network and WDC is not found to overflow in any of the cases. However, for network sizes larger than 8x8, the 6 bit WDC field is found to overflow at high injection rates. Hence, more number of bits are required for WDC field when network dimensions increase.

3.1.3 Livelock avoidance

The proposed WeDBless algorithm adopts a flit prioritisation scheme based on WDC of flits for livelock avoidance. Frequently deflected flits in the network have higher values of WDC due to the addition of DWs. This increases the priority of these flits helps them to win productive ports during output port allocation and progress towards destination. In a router, if all the four inputs flits have same value of WDC, contention for output port is limited by the permutability of the Permutation Deflection Network. After output port allocation, WDC of flits that obtained productive ports is decremented by 1 (if WDC is zero , further decrementing is prohibited) whereas WDC of flits that

deflected through non productive ports is incremented by 1 or 2 depending on the direction of deflection. A situation of possible livelock arises when all these flits return to the current router after getting deflected from the neighboring routers. Now, the updated WDC values of all the flits will be different from each other. So during port allocation in the current router, the flit prioritisation will depend on the new WDC values and the highest priority flit will necessarily proceed along the productive path. A flit will loop back to the current router repeatedly (livelock condition) only if its neighboring routers prioritise the flit in the same way during port allocation. This in turn depends on the WDC of other flits entering the neighboring routers. For a given network dimension, the WDC field is wide enough so that repeated deflections leading to incrementing of WDC doesnot cause overflow and livelock.

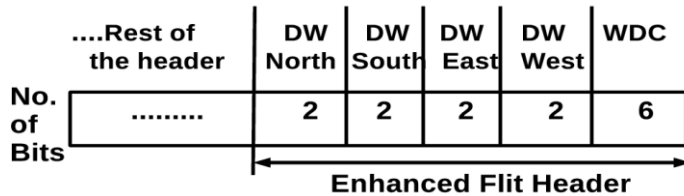


Figure 3.3 Format of enhanced flit header in an 8 x 8 mesh NoC.

3.1.4 Enhanced flit header

In deflection routers, each flit is routed independently. The source and destination addresses of each flit are included in the flit header. In the proposed method, the flit header is enhanced to incorporate the WDC and DW values. Figure 3.3 shows the enhanced flit header format. For an 8 x 8 mesh NoC, the enhanced header consists of a total of 14 bits; 8 bits for DWs and 6 bits for WDC.

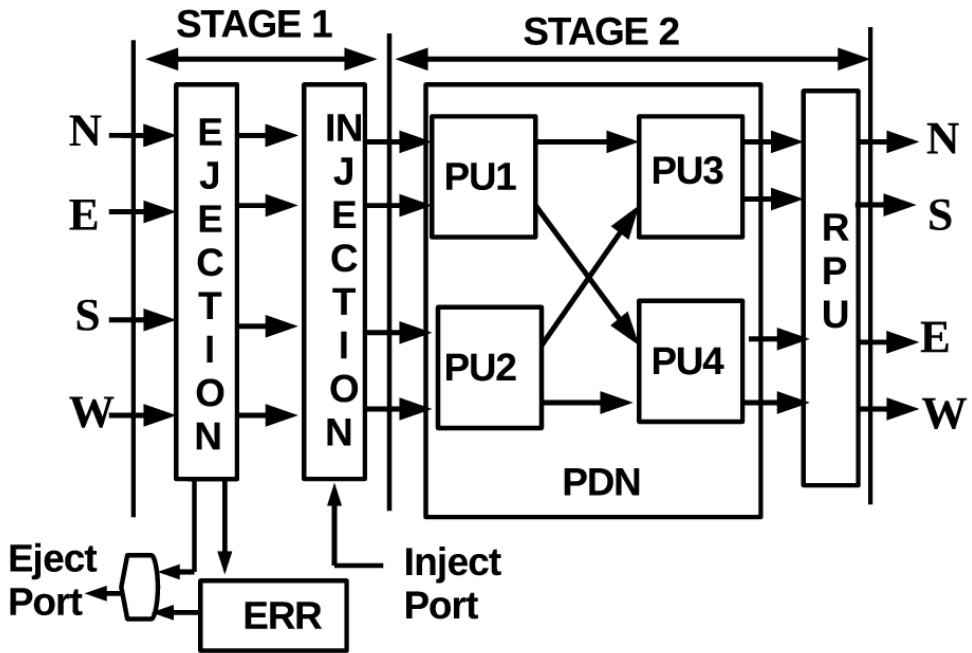


Figure 3.4 Two stage pipeline architecture of WeDBless router.

3.1.5 Router architecture

The pipeline architecture of the proposed WeDBless router is shown in Figure 3.4. The flits from neighbouring routers enter the router pipeline through input ports (shown on left) and move towards output port (shown on right) through various functional blocks. The logic functions are divided into two pipeline stages and the total router latency is two cycles. Ejection and injection constitute the first cycle of the router pipeline. The second cycle comprises of output port allocation by the PDN followed by the Route Pre-computation Unit (RPU) at the end of the router pipeline. After this unit, flits are stored in the output registers and transferred to neighbouring routers through the connecting links. The function of each unit is explained below.

Ejection unit

The function of the ejection unit is to remove flits destined for the local PE from the network. The ejection unit consists of a flit identification circuit, an Ejection Ready Register (ERR) and a eject multiplexer. The flits destined for the local PE are referred to as Ejection Ready Flits (ERF). The flit identification circuit identifies the ERF from among the flits arriving at the input ports of the router. By conducting simulations of an 8 x 8 network using real application traffic, it is observed that for 10% of the operation cycles, there are more than one ERF in a router. For a few cases, more than one ERF is present in two consecutive cycles. In the proposed WeDBless router, when two ERFs are present in one cycle, the ERF with the highest WDC is ejected through the local output port and the other flit is buffered in the Ejection Ready Register (ERR). The buffered ERF is ejected from the router in the next cycle with highest priority. If an ERF is present among the next set of input flits in the following cycle, it gets preference for ejection after the ERR is emptied. If more than two ERF are present in a router in one cycle, one is ejected, one is buffered and the others are deflected. As seen from the simulations, the occurrence of such cases is very rare. Buffering of one ERF in a cycle substantially reduces deflections due to them.

Injection unit

The local PE is allowed to inject a flit into the network through a vacant input channel in the router [5]. This function is carried out by the injection unit. In WeDBless, at most one flit can be injected into the router in an operation cycle. In a given cycle, if flits arrive at all the four input ports of a router, then

injection is possible only if a vacancy is created by ejecting a flit from the network or by buffering a flit in the ERR.

Permutation Deflection Network

WeDBless router uses a simple and fast PDN for output port allocation, which was first proposed in CHIPPER. It efficiently maps each of the input channels to every output port of the router. As shown in Figure 3.4, the PDN consists of four permuter blocks (PU1, PU2, PU3, PU4) arranged in two stages, two units per stage. Each permuter block consists of two inputs and two outputs. In the first stage, the north and east input ports are connected to PU1 and south and west input ports are connected to PU2. In each permuter block, the input flit with higher WDC value has higher priority and is assigned to the output port of its choice. The flit with lower WDC gets the remaining output port. A flit chooses the output port with lesser value of DW from the set of DWs contained in the flit header. Output lines of PU1 and PU2 are connected to the permuters PU3 and PU4 which form the second stage of the PDN. After arbitration in PU3 and PU4, flits are assigned to the four output channels of the PDN.

Route computation on the basis of DWs exploits the path diversity of mesh topology and thereby helps in routing flits through the shortest available path to their destinations. A flit whose destination is equidistant from the current router in the x and y directions of the mesh, will have two equally productive output ports. The DWs of the flit for these ports are -1 and the remaining unproductive ports are +2. Even if this flit failed to obtain any one of the productive ports in the first stage of PDN due to low priority, it can still arbitrate for an equally desirable output port in the second stage. For example,

if a flit at the north input port of a router is destined for a router in the south east direction, both south and east output ports are equally productive for it. If the flit tries to access the desired south output port at permuter PU1 and fails due to lower WDC, it will be deflected to PU4 which is connected to east and west output ports. In permuter PU4, the flit gets a second opportunity to acquire the east output port which is also in a productive direction towards its destination. If the flit loses in this arbitration and gets deflected through the west output port, its WDC value will be incremented with a weight of + 2. The incrementing of WDC of this flit raises its probability of winning a productive port in the successive router. The updation of WDC takes place after output port allocation at permuters PU3 and PU4.

The ERF that move into the PDN for port allocation are deprioritised. This ensures that they are allotted output ports that are not demanded by other flits. This de-prioritisation logic increases the chance of other flits (non ERF flits) to win the desired output ports.

Route Pre-computation Unit

The RPU computes the productive output ports of flits in the succeeding routers. The next hop directions of an input flit in a router is determined by the PDN. RPU computes a new set of DWs for each of these flits to be used for route computation in routers in subsequent hops. RPU performs route pre-computation by incrementing, decrementing or complementing the existing values of DW. For example, consider a flit whose current DWs are North: +2, South: -1, East: +1 and West: +1, which implies that the south output port is desirable for the flit. The different situations that may occur are summarised in Table 3.2 and explained below.

- Case 1: If the flit succeeds in acquiring the south port in the PDN, it will progress towards destination, hence the newly computed set of DWs at the RPU will be same as the existing values (3rd row of Table 3.2).
- Case 2: If the flit is deflected through the north output port in the reverse direction of its destination, the newly computed set of DWs will be same as the existing values. This is because the flit has to traverse backward to reach its destination (4th row of Table 3.2).
- Case 3: If the flit is deflected in an orthogonal direction i.e. east or west port, the DW of the selected port is incremented by 1 whereas DW of the opposite port remains unchanged. For the port in the preferred direction, the new DW is obtained by complementing the current value (5th and 6th row of Table 3.2).

Table 3.2 Current and pre-computed DWs of a flit.

Port allocated to the flit	Direction of destination	Current DW				Pre-computed DW			
		N	S	E	W	N	S	E	W
South	Towards	+2	-1	+1	+1	+2	-1	+1	+1
North	Away	+2	-1	+1	+1	+2	-1	+1	+1
East	Orthogonal	+2	-1	+1	+1	+2	+1	+2	+1
West	Orthogonal	+2	-1	+1	+1	+2	+1	+1	+2

3.1.6 Simulation methodology

The proposed WeDBless router is modelled using a traditional NoC simulator, Booksim [3][70]. Booksim incorporates a buffered virtual channel router having two cycle delay as the default router. This router model is modified to represent the proposed WeDBless architecture as explained in Section 3.2.4. In order to demonstrate the benefits of the proposed routing mechanism, two state-of-the-art router based NoCs viz. BLESS and CHIPPER are also modelled with two cycle latency. In all the three networks, the link latency is taken as one cycle. Packets are assumed to consist of single flits for all simulations. Each flit is prefixed with a header to facilitate independent routing. The flit width for CHIPPER and BLESS is assumed to be 128 bits. For a 8x8 mesh NoC with WeDBless routers, the flit header is enhanced with 14 bits for representing DW (8 bits) and WDC values (6 bits) as mentioned in Section 3.2.3. Synthetic traffic patterns mimic the behaviour of the target application; so they are used in simulations to compare the performance parameters of BLESS, CHIPPER and WeDBless routing algorithms. Synthetic traffic does not allow self-throttling; hence they are best suited to test the network saturation point of an algorithm. Simulations are conducted using uniform-random, transpose, bit complement and tornado traffic patterns. Uniform traffic is used to assess the adaptability and load balancing capability of the routing algorithm whereas transpose, bit-complement and tornado generate network intensive traffic. After providing sufficient warm up time, the average flit latency, deflection rate and throughput values are observed by incrementing the injection rate from low to saturation level.

Another set of simulations is performed using SPEC CPU 2006 benchmark application suite [71] to compare the application level performance

of WeDBless with the other two methods. SPEC consists of 29 applications out of which 12 are integer based and 17 are floating point applications. Six multi-programmed workload mixes are generated by combining applications from the suite with low, medium or high amount of Misses Per Kilo Instructions (MPKI). Applications like calculix, h264ref and gobmk belong to the low MPKI group; bwaves, bzip2, gcc are some of the medium MPKI and hammer, lbm, and mcf are high MPKI applications. The percentage of application from each category used in the workload mixes (M1 to M6) is given in Table 3.3. Each set of workload mix is run on a cycle accurate multicore simulator, Multi2sim with an 8x8 mesh topology and an out-of-order x86 processor and a shared L2 cache at every node [72]. Network traces generated during the execution of these workloads are injected as input packets to the three NoC simulators and various performance parameters are measured and analysed.

Table 3.3 Various proportions of SPEC CPU 2006 applications in benchmark mixes M1 to M6.

Application	Benchmark Mix					
(%)	M1	M2	M3	M4	M5	M6
Low MPKI	100	0	0	50	0	50
Medium MPKI	0	100	0	0	50	50
High MPKI	0	0	100	50	50	0

3.1.7. Results and discussions

Average deflection rate

Average deflection rate is computed as the average number of deflections encountered per flit. The graphs of average deflection rate for four typical

synthetic traffic patterns are shown in Figure 3.5. The weighted deflection routing method aids to minimize the deflection rate and achieve energy efficiency for the NoC. For pre-saturation injection rate, WeDBless reduces the deflection rate by 56% with respect to CHIPPER for uniform traffic profile. The unique routing mechanism in WeDBless reduces deflections in three ways: (1) by computing multiple productive paths for a flit (2) by increasing WDC value for deflected flits so that they win productive output ports in the succeeding router's PDN (3) by providing an Ejection Ready Register to buffer ERF to reduce deflections due to them.

Average latency

Flit latency is measured as the average number of cycles taken by the flit from injection to ejection. Average flit latency for some typical synthetic traffic patterns are shown in Figure 3.6. A lower value of latency means that the flits traverse the network in lesser number of cycles and it results in application speed up. It is obvious that the reduction in deflection rate obtained for WeDBless reflects in the average flit latency also. For uniform traffic, the adaptability and reduced deflections of the proposed algorithm improves the network saturation point by 26% compared to CHIPPER and 8% compared to BLESS. For very low injection rates, the routing decisions of the three algorithms are similar and so the average latency shows the same trend. When injection rate is increased, WeDBless gives higher priority to flits that have undergone more deflections and this reduces the average flit latency compared to CHIPPER. Utilisation of ERR to reduce the deflection of ERF accounts for the reduction in average latency of WeDBless when compared with BLESS. For transpose, bit complement and tornado traffic patterns which

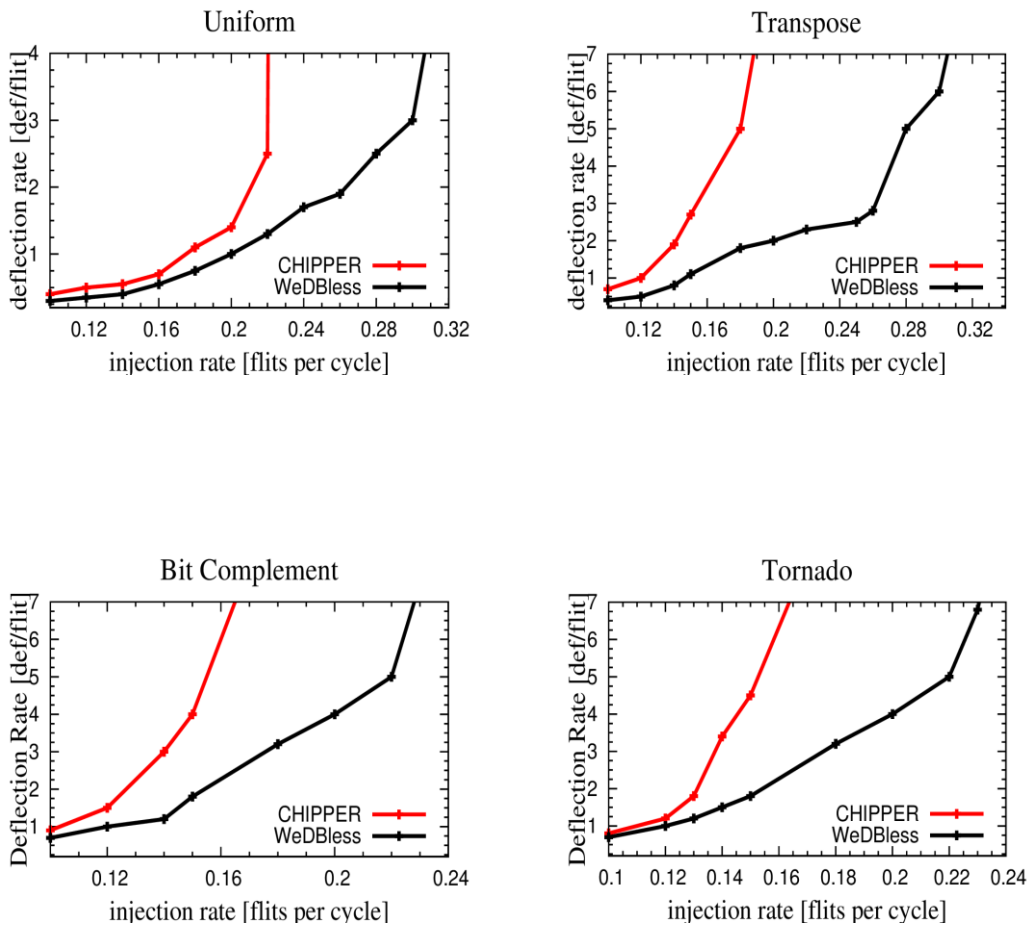


Figure 3.5 Average deflection rates for various synthetic traffic patterns using WeDBless and CHIPPER in 8x8 mesh NoC.

cause significant network congestion, there is 55% improvement in network saturation point for WeDBless compared to CHIPPER.

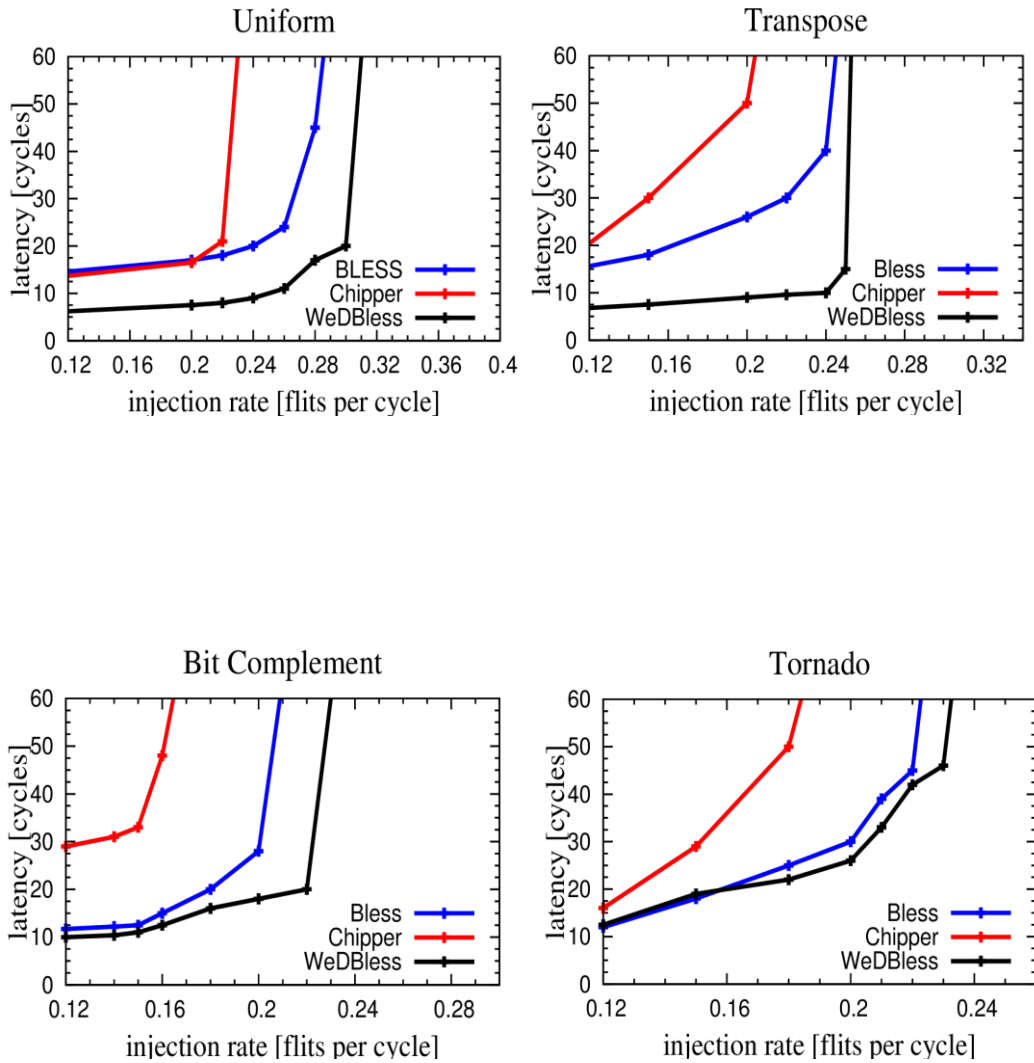


Figure 3.6 Average latency for various synthetic traffic patterns using WeDBless, CHIPPER and BLESS in 8x8 mesh NoC.

Average throughput

The network throughput is computed as the number of flits ejected from the network per router per cycle. In an ideal deflection network, the delivered throughput is equal to the injection rate of flits. At lower injection rates, there is very less congestion in the network. As a result, all the three methods deliver a constant throughput value close to the ideal value. CHIPPER network reaches saturation earlier than the other two methods and average throughput is also low. Comparison of average throughput with synthetic traffic patterns for the three methods is shown in Figure 3.7. For all the synthetic traffic patterns, BLESS and WeDBless deliver similar values of average throughput at low injection rates. Beyond saturation injection rate, WeDBless shows marginally high throughput especially for worst case traffic patterns like tornado.

Real applications

Figure 3.8(a) shows the percentage reduction in deflection rate for WeDBless with respect to CHIPPER for the benchmark mixes from Table 3.3. WeDBless shows a maximum of 55% reduction in deflection rate which is attained for high MPKI application workloads. This is because the proposed routing technique is more effective in reducing deflections under high injection rate application traffic. From latency simulations using real application mixes shown in Figure 3.8 (b), a maximum of 25% reduction in average latency is observed for WeDBless in comparison with CHIPPER.

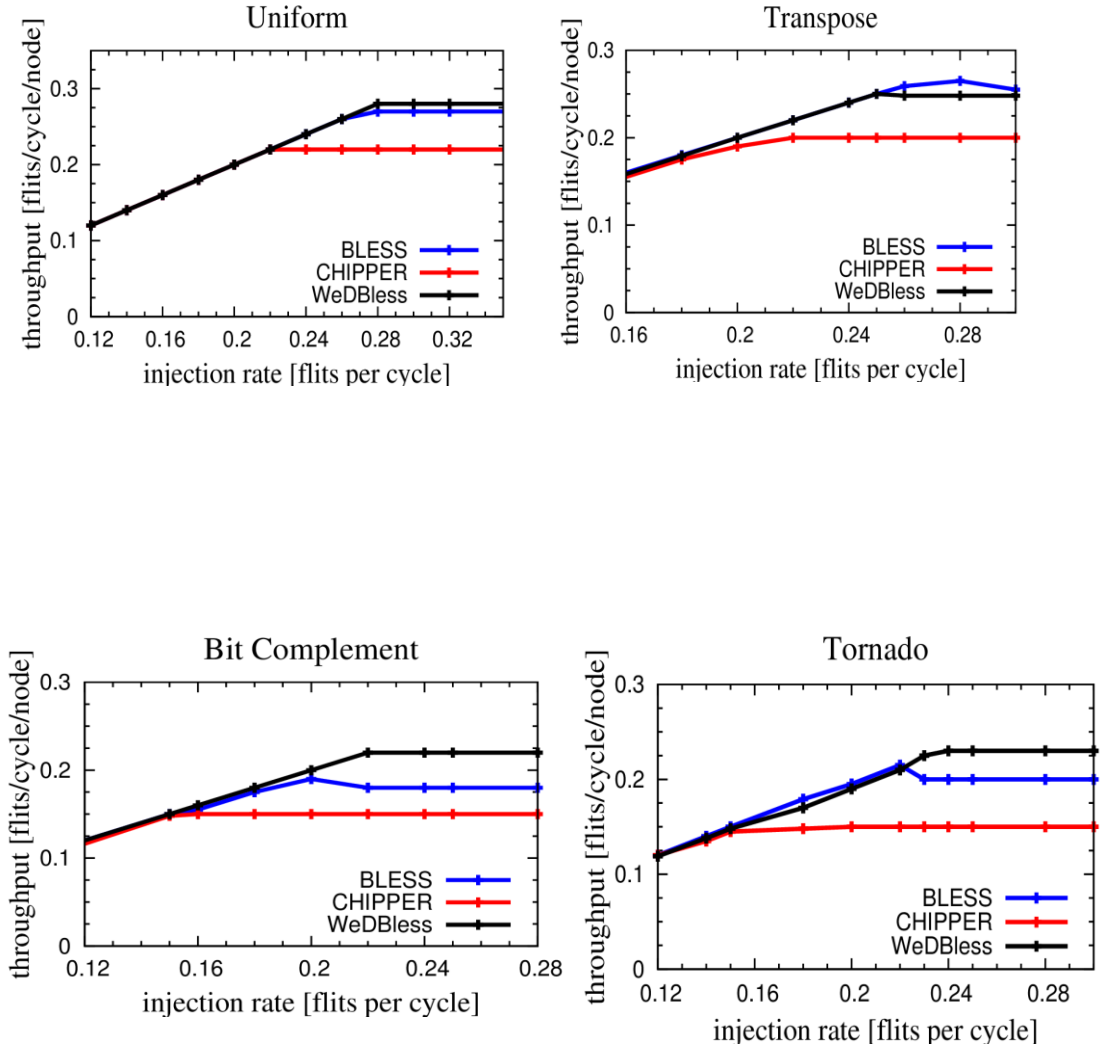
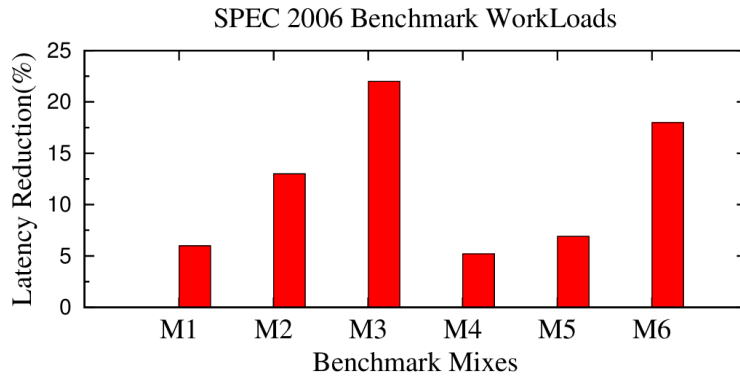


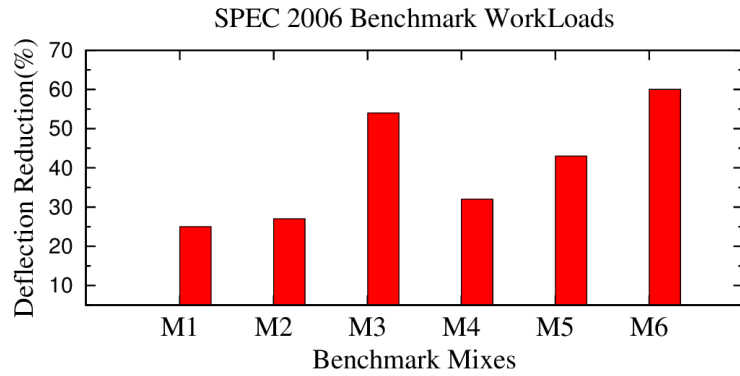
Figure 3.7 Average throughput for various synthetic traffic patterns using WeDBless, CHIPPER and BLESS in 8x8 mesh NoC.

Dynamic power dissipation across NoC links

Increase in the flit deflection rate leads to increased activity and dynamic power dissipation across the NoC links.



(a)



(b)

Figure 3.8 Percentage reduction in (a) deflection rate (b) latency of WeDBless w.r.t CHIPPER for real workloads.

Therefore, energy efficiency of NoCs is proportional to the Link Activity Factor (LAF) which is derived from the average hop count of flits through each of the links at a flit injection rate of 0.1 per cycle per core. LAF is a measure of the number of flits per cycle per link and is calculated using Equation 3.1.

$$LAF = \frac{h}{c * k} \quad - (3.1)$$

where

- h - Sum of hop count of all packets
- c - Total number of simulation cycles (taken as 1,00,000).
- k - Total number of links (112 for 8x8 mesh)

Orion 3.0 tool is used to estimate the dynamic power dissipated in the NoC links [73]. The calculated value of LAF is given as the input load rate in Orion to determine the values of dynamic power for various injection rates. Link length of 2.5 micrometres and 1V Vdd are assumed. A flit width of 128 bits is assumed for CHIPPER. For an 8x8 network, WeDBless router requires 14 additional bits to include the WDC (6 bits) and DWs (8 bits).). Figure 3.9 shows the dynamic link power dissipation for WeDBless, CHIPPER and BLESS for typical synthetic traffic patterns. Compared to CHIPPER and BLESS, reduced flit deflection rate of WeDBless leads to an equivalent reduction in dynamic power across the NoC links.

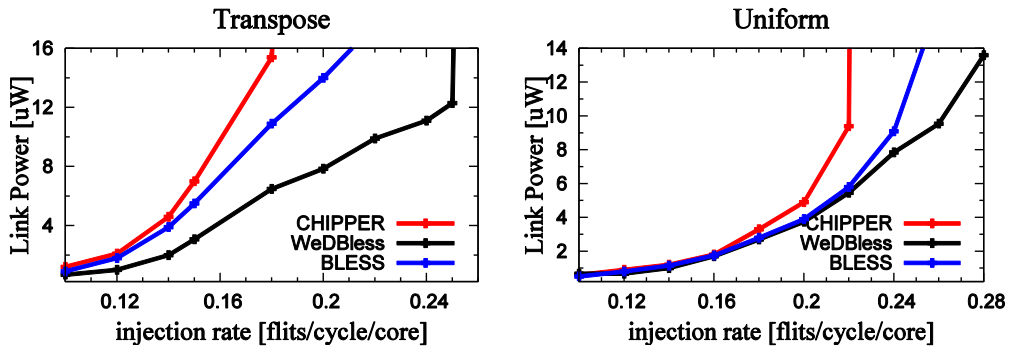


Figure 3.9 Dynamic power dissipation across links for various synthetic traffic patterns using WeDBless, CHIPPER and BLESS in 8x8 mesh NoC.

3.1.8 Hardware synthesis

Table 3.3 Comparison of pipeline delay, area and static power of CHIPPER and WeDBLESS

	Pipeline Delay (nsec)	Area (μm^2)	Static power (μW)
CHIPPER	1.2	2254	182
WeDBLESS	1.3	2467	226

To assess the critical path delay, power and area overhead of the proposed method, Verilog models of CHIPPER and WeDBLESS routers are synthesised by Synopsis Design Compiler using 65 nm CMOS library. The synthesis results are given in Table 3.3. The delay of the second stage of the router pipeline is 1.2 nsec and 1.3 nsec respectively for CHIPPER and WeDBLESS. The RPU in the second stage of WeDBLESS contributes to 8.2% additional delay with respect to CHIPPER. Due to flit header enhancement

with 14 bits for 8x8 mesh NoC (11% increase for a flit size of 128 bits), WeDBLESS incurs an area overhead of 9.4% and static power overhead of 24% compared to CHIPPER.

3.2 Weighted Deflection router with Minimal Buffering

The MinBD router evolved out of the CHIPPER architecture by introducing a small side buffer in the router pipeline [10]. The side buffering technique helps to overcome the inherent drawback of the PDN based output port allocation scheme. In MinBD, a flit that is assigned to an unproductive output port by the PDN is chosen for side buffering. Only one such flit can be buffered in a cycle. The FIFO buffer retains these flits for a fixed number of cycles. These flits re-enter into the router pipeline prior to the port allocation stage through any of the vacant input channels. During port allocation in the PDN, the priority of flits from the side buffer is escalated compared to normal flits so that they can win output ports during arbitration. As in the case of CHIPPER, MinBD also showcases poor performance due to the golden flit prioritisation scheme and deterministic route computation method. To achieve enhanced performance, the newly proposed weighted deflection routing method is incorporated in MinBD by suitably modifying its router architecture.

3.2.1 Router pipeline

Figure 3.10 depicts the two stage pipeline architecture of a Minimally Buffered Weighted Deflection router which is referred to as MinBWD in the following sections of this chapter. The first cycle (Stage I) is constituted by an ejection unit and two injection units placed one after the other. Dual ejection is

enabled using an Eject Buffer along with a single ejection unit as explained in Section 3.1.4.

Of the two injection units, the first one is for injection of flits from the side buffer and the other is for injection from the local PE. In order to prevent starvation of flits in the side buffer, injection from the buffer is given preference over injection from the local PE. Newly generated flits can be introduced into the network only if a vacant flit channel is present in the router after side buffer injection.

The second cycle (Stage II) involves output port allocation by the PDN, side buffering of deflection flits and RPU. The function of PDN and RPU are similar to that explained in Sections 3.3.3 and 3.3.4. In the buffer eject unit, one among the deflected flits from the PDN is chosen at random and

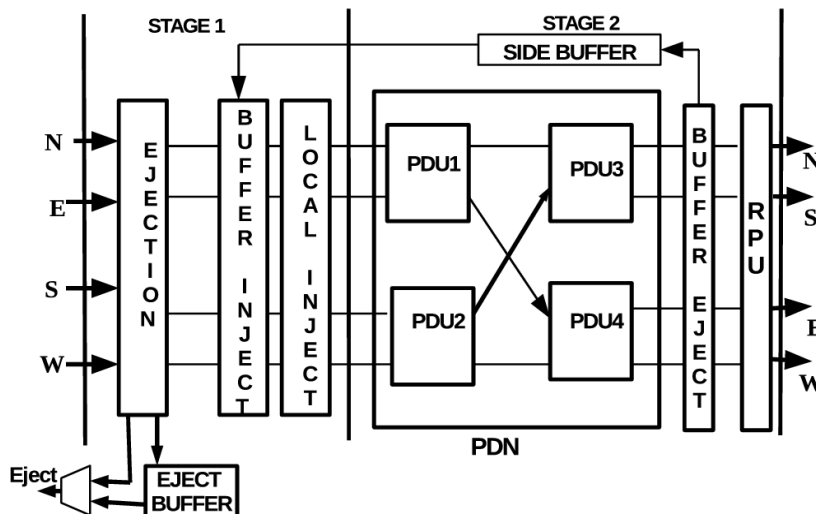


Figure 3.10 Two stage pipeline architecture of MinBWD router.

transferred from the router pipeline to the side buffer. The priority of the buffered flit is raised by incrementing its WDC by 2 before transferring it to the FIFO buffer.

3.2.2 Results and discussions

The experimental setup for the evaluation of the proposed MinBWD routing method is similar to that explained in Section 3.1.5. The simulation results obtained for MinBWD are compared with the baseline architectures, CHIPPER and MinBD. In an 8x8 mesh NoC, the flit header is enhanced by 14 bits for the proposed method.

Average deflection rate and latency

The graphs comparing the average deflection rate of the three methods for three typical synthetic traffic patterns i.e. uniform, transpose and bit complement are shown in Figure 3.11. MinBWD reduces the average deflection rate by 56% compared to MinBD for uniform-random traffic whereas for non-uniform traffic distribution like transpose, the reduction in deflection rate at pre-saturation injection rate is 33%. Unique features of weighted deflection routing mentioned in Section 3.1 account for the reduction in deflection rate. Figure 3.12 (a) shows a comparison of average deflection rates for MinBD and MinBWD from simulations using workloads (M1 to M6) from SPEC CPU benchmark applications given in Table 3.3. The proposed method delivers a maximum of 36% reduction in deflection rate which is obtained for workloads with high MPKI applications. This explains that the benefits of the proposed algorithm are more evident under higher network load.

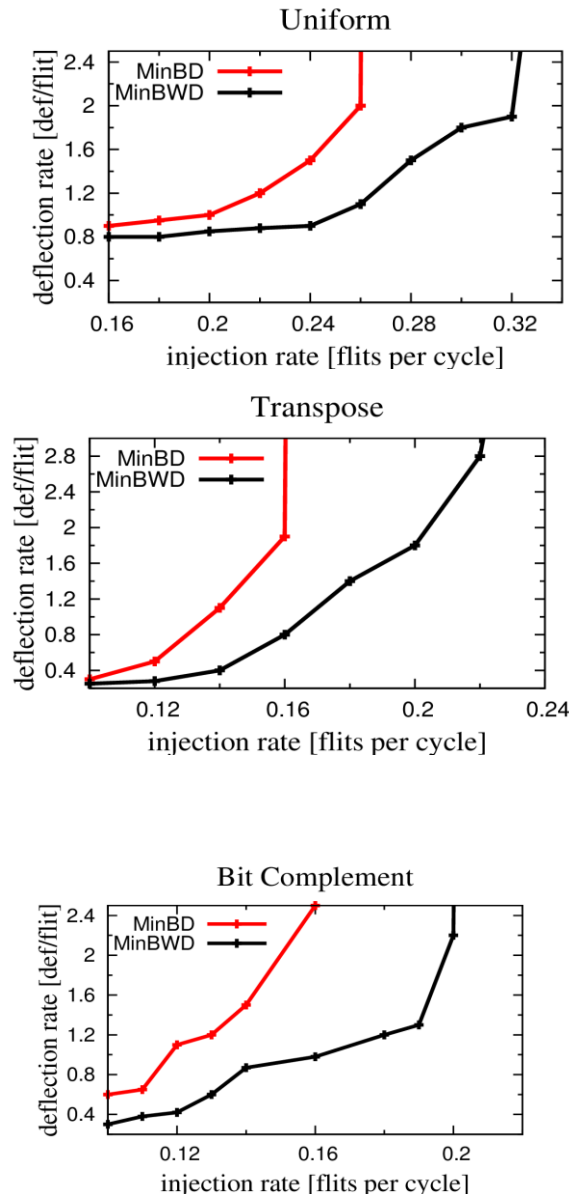
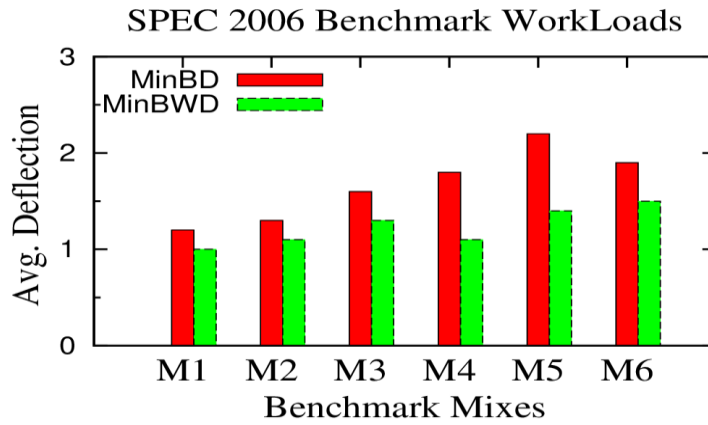
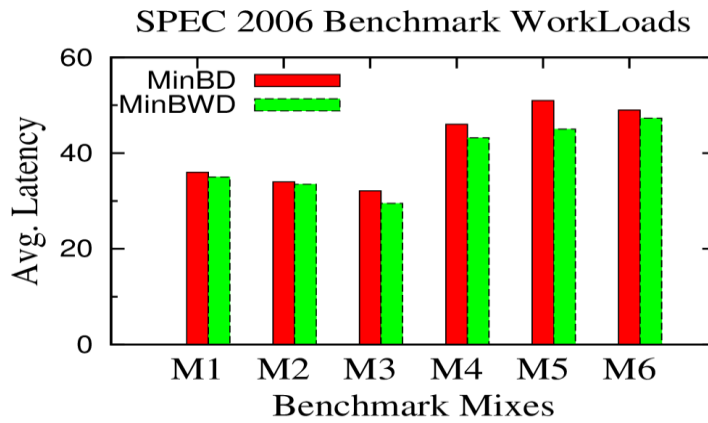


Figure 3.11 Average deflection rate for various synthetic traffic patterns using MinBWD and MinBD in 8x8 mesh NoC.



(a)



(b)

Figure 3.12 (a) Average deflection rate and (a) Average latency for real applications using MinBWD and MinBD in 8x8 mesh NoC.

Figure 3.12(b) shows the comparison of average latency for the same workloads using MinBWD and MinBD. For low MPKI applications, both the methods exhibit equal latency whereas for workloads with larger proportion of high MPKI applications, latency benefits of MinBWD are more significant. Due to the routing efficiency of the proposed method, it can handle higher injection rate than MinBD before the network reaches saturation; hence the throughput delivered by the proposed method is also higher than MinBD in all cases.

Network scaling

The impact of network size on the performance parameters is analysed from Figure 3.13 and Figure 3.14 which represent the average latency graphs for 8x8 and 4x4 mesh sizes respectively. With increasing congestion in the network, the weighted deflection routing algorithm has better capability to explore alternate routes for flits. For larger network sizes, the availability of alternate links increase. Due to this, the proposed method continues to deliver consistent performance for high injection rates. This results in the network saturation point of the proposed method being extended by 26% compared to conventional MinBD. A major drawback of the proposed method is the increase in the number of bits for WDC with increasing network size. For larger network sizes, the enhancement in flit header increases the width of the router's datapath and NoC links.

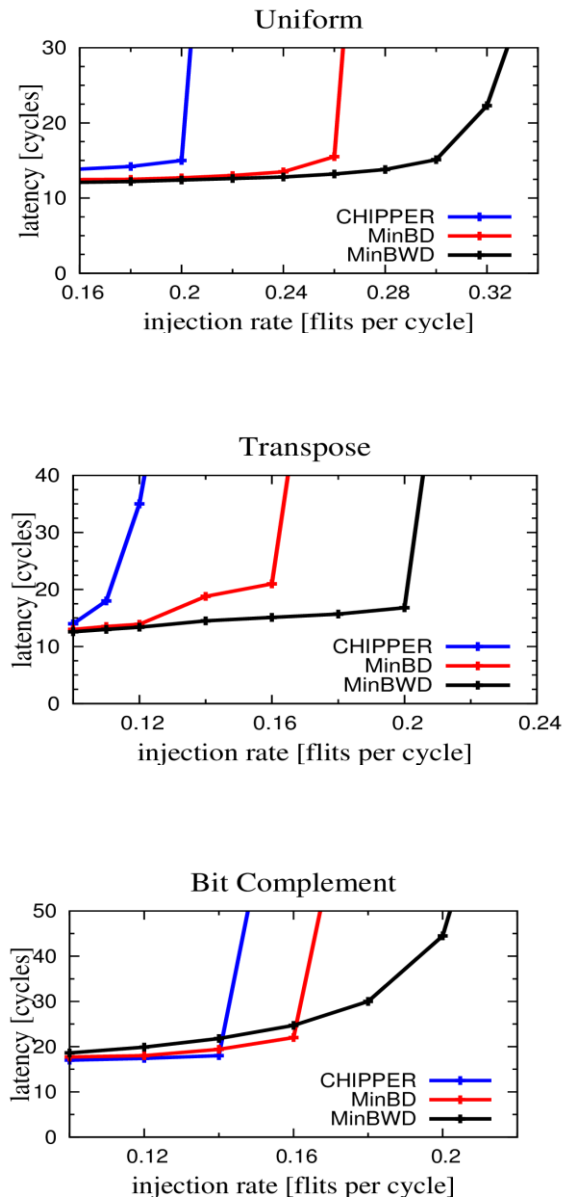


Figure 3.13. Average latency for various synthetic traffic patterns using MinBWD, MinBD and CHIPPER in 8x8 mesh NoC.

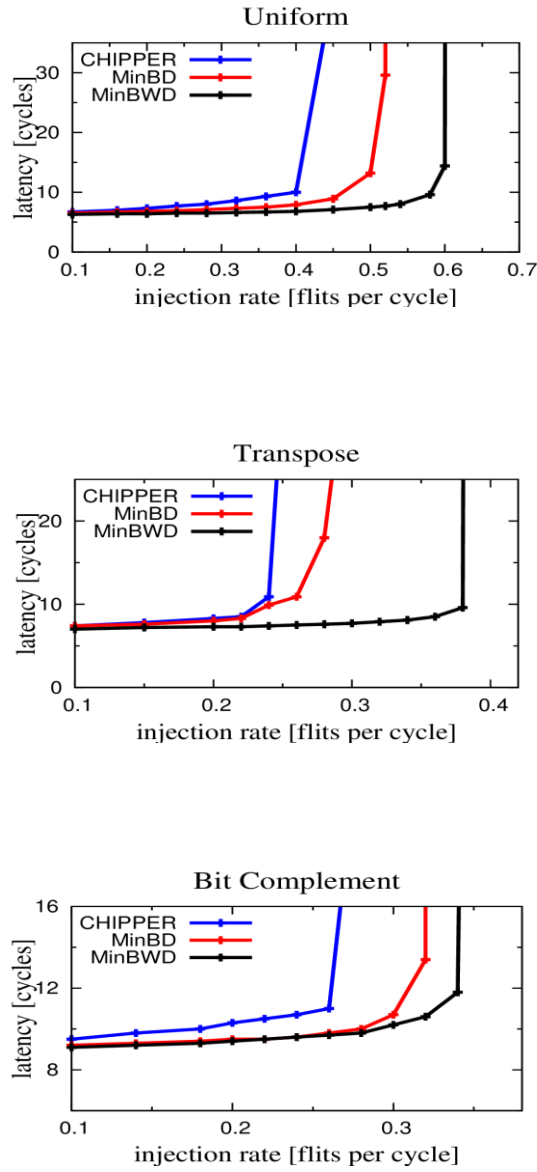


Figure 3.14. Average latency for various synthetic traffic patterns using MinBWD, MinBD and CHIPPER in 4x4 mesh NoC.

3.2.3 Router timing, static power and area

Table 3.4 Comparison of pipeline delay, area and power of MinBD and MinBWD

	Pipeline delay (nsec)	Area (μm^2)	Power (μW)
MinBD	2.1	3242	321
MinBWD	1.9	3079	414

The RTL models of the MinBWD and MinBD routers are synthesised using Synopsys Design Compiler with 65nm technology library. The synthesis results are summarised in Table 3.4. MinBWD enables dual ejection using single ejection block whereas MinBD uses two ejection units in the first stage of the router pipeline. As a result, MinBWD reduces the delay of the first stage (1.4 nsec) by 32% compared to MinBD (2.1 nsec). However, the port allocation stage of MinBWD has longer critical path and it decides the operating frequency of the network. In MinBD, the silver flit prioritisation function together with golden and silver flit priority checking in the permuters accounts for a delay of 2.1 nsec in the output stage. The proposed MinBWD router uses a simpler prioritisation based on WDC, but at the same time adds delay due the RPU which sums up to a datapath delay of 1.9 nsec for the second stage.

By analysing the datapath delay of individual functional blocks in the input/output stages of the two routers, it is concluded that NoCs using the newly proposed MinBWD and MinBD can be operated at similar frequencies. For further improvement in the speed of operation of the proposed router, the computation of DWs can be performed for input flits in the first stage in parallel to the ejection unit, the RPU can be eliminated from the router's critical path and

DWs are eliminated from the enhanced flit header. The benefits of this arrangement are 13% reduction in the critical path delay and 6.3% reduction in link width compared to the original weighted deflection router.

3.3 Chapter Summary

The newly proposed routing method mitigates the problem of high deflection rate and latency due to inefficient output port allocation in deflection routers. The proposed technique is based on weighted deflection routing which uses a flit prioritisation scheme based on Weighted Deflection Count of flits and output port prioritisation based on Directional Weights. This method is implemented on buffer-less and minimally buffered deflection router architectures. The merits of the proposed method are justified by comparison of performance parameters like deflection rate, average latency and dynamic power dissipation with contemporary techniques like BLESS, CHIPPER and MinBD.

Chapter 4

High Performance Adaptive Deflection Routing

Abstract

An adaptive deflection router with single cycle delay is presented in this chapter. Average latency of the NoC is reduced by optimizing the time occupied by flits in side buffers. The router uses a decision making logic unit which intelligently chooses flits for buffering prior to output port allocation. Experimental results confirm that the proposed architecture outperforms previous methods in terms of various network performance parameters.

The operating frequency or clock duration of a deflection router is decided by its longest data path which is otherwise known as the critical path length. In deflection routers like CHIPPER, MinBD and DeBAR, the router delay is two cycles and link traversal consumes one cycle of operation [10, 11], [27]. According to the equation 1.1, the overall performance of the NoC can be boosted by reducing the router delay to a single cycle, which can be achieved by reducing the router's critical path length. However, single cycle deflection routers require wider clock pulse than two cycle routers in order to complete all the routing functions in one cycle. Increasing the clock width leads to higher link latency and slower operating frequency of the network. In such NoCs, performance benefits are obtained in terms of reduced average flit latency such that the total time traversed by flits is lower than that of two cycle router based NoCs.

With the aim of reducing the path length, a minimally buffered single cycle deflection router, MinBSD was proposed [30]. In MinBSD, the functional blocks which constitute a two stage deflection router pipeline are restructured so as to reduce the router delay and complete the entire router operation in a single cycle. Parallelisation of independent operations like route computation, ejection and flit prioritisation help to shorten the router's critical path. MinBSD uses a two stage PDN as depicted in Figure 4.1. Each stage has three permuter blocks that connect the four input ports and two buffer outputs to the four output ports and two buffer inputs. Two buffers viz. the side buffer and core buffer store a small number of misrouted flits. The core buffer also stores flits generated by the local processor. Injection from these two buffers takes place in all cycles. The 3x2 arbiter incorporates port allocation and buffering functions into a single functional block, thereby reducing the PDN delay significantly.

The MinBSD architecture experiences certain limitations like structural isolation, delay due to extra arbiter stage and penalisation of high priority flits. These limitations are discussed in detail below.

- **Problem of structural isolation:** As we can see from Figure 4.1, flits are not always free to move to their desired direction in a single cycle because all input ports are not structurally connected to all output ports. Due to this problem of structural isolation, many of the flits are sent to side buffer or core buffer instead of its desired output port. It occurs in cases when a flit needs to take a left or right turn. Consider a flit coming from the north input link, and wishes to move to the west direction. In the first level of arbiters, the flit chooses the second link of the L1 arbiter and reaches the input of the R2 arbiter. Then it finally moves out of the output links either into the core buffer or side buffer. In the next cycle the flit again gets injected into the same router through the L2 arbiter of the first stage and moves into the R3 arbiter in the second stage. Simulations are conducted on 8x8 mesh NoC using synthetic traffic patterns and it is observed that approximately 24% of the flits destined to an output permuter are sent to a side buffer due to the problem of structural isolation.
- **Delay due to extra arbiter stage:** The flits that are to be buffered need to go through two levels of arbiters. This causes an extra delay in the router, which in turn reduces the running frequency of the network. If it is possible to select the flits to be buffered before they enter the PDN stage, such flits can be moved out into the buffers directly, rather than arbitrating through the permutation network.

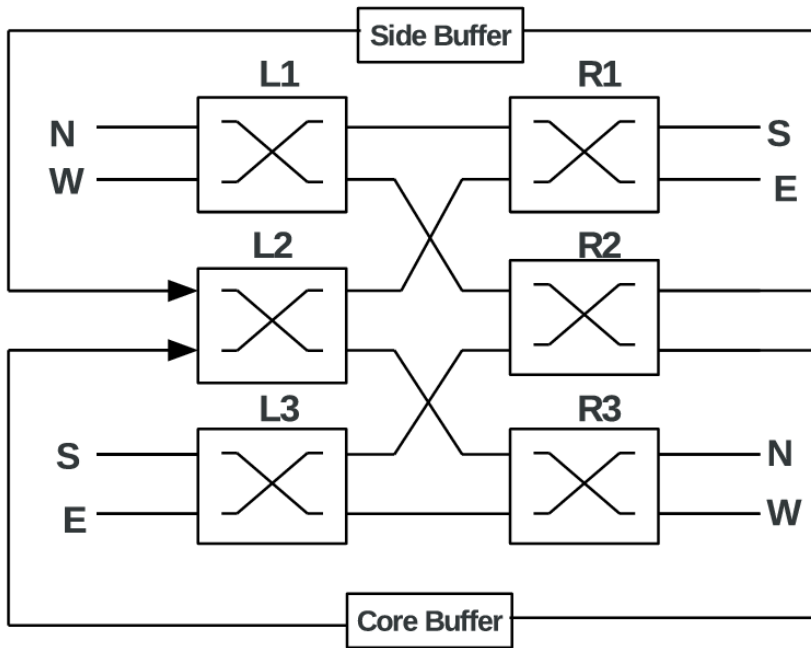


Figure 4.1. PDN structure of MinBSD [30]

- Penalisation of high priority flits:** In each arbiter of the PDN, the flit with higher priority is given preference in the choice of output port. As shown in Figure 4.1, there are no direct paths connecting the L1 and R3 arbiters. Similarly there are no paths between L3 and R1 arbiters. Hence a high priority flit which chooses a desired output in the first stage of the PDN may end up in a side buffer or core buffer. As explained in the previous example, such flits are penalised and they are delayed by an additional cycle before re-entering into the PDN, inspite of having high priority. By experimental analysis, it is found that about 23% of the high priority flits targeting for an output port which is connected to R1 or R3

arbiters gets buffered by moving into the R2 arbiter of the second stage.

For a deflection router with minimal buffering, latency of a flit through the network in cycles is given by the expression,

$$L = CBT + SBOT + HOPS \quad - (4.1)$$

where,

CBT - Wait time in Core Buffer before Injection

SBOT- Side Buffer Occupancy Time

HOPS - Total number of hops from source to destination

Some of the hops undergone by flits will be deflections. By optimising each of these terms, flit latency can be reduced and higher performance can be achieved. By simulating an 8x8 mesh network using MinBSD routers, the flits are fractioned into various bins based on the values of SBOT and overall latency. It is observed that SBOT of 43% of the flits in uniform traffic is 1 or zero cycles , 30% of the flits is 2 to 3 cycles and rest of the flits consume more time in side buffers. Because of this, the total latency of majority of flits is also greater than 10 cycles. This is due to the structural isolation problem discussed earlier. To alleviate these problems and achieve performance enhancement, a High Performance Adaptive Deflection router (HiPAD) for two dimensional mesh NoCs is proposed here.

4.1 Router architecture

The architecture of the proposed High Performance Adaptive Deflection Router (HiPAD) is shown in Figure 4.2. The role of each functional block in the routing process is explained below. Initially the flits undergo routing, ejection and prioritization. These units receive flits arriving at the router through pipeline register A. Routing Unit (RU) does the route computation for incoming flits based on the position of the source and destination router in the mesh. This unit decides the possible directions a flit can move to reach its destination. Ejection unit supports one ejection port per router. In case of multiple flits to be ejected in a router, one among them is selected randomly for ejection. Others continue to move in the router pipeline. Prioritization Unit (PU) prioritizes the flits based on the number of hops remaining to reach the destination. Flit with least number of hops to destination is given highest priority. Next logic block in the router is the injection unit which injects flits from the core buffer into the network through a free input channel. Further, the PDN arbitrates the flits and forwards them to the output links of the router.

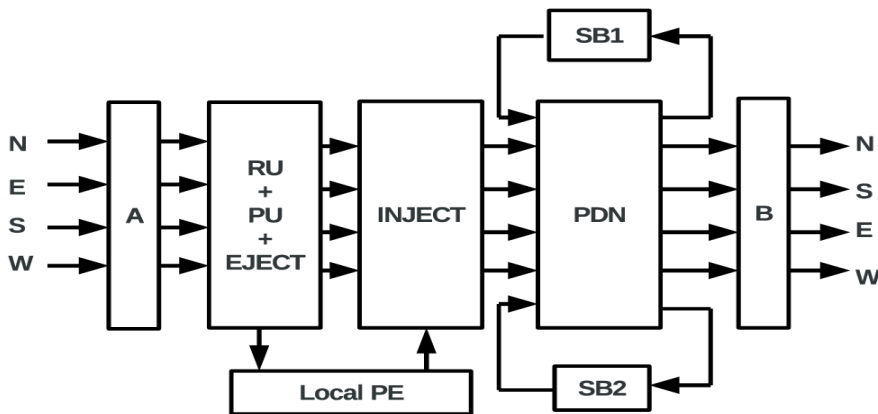


Figure 4.2 Architecture of the proposed HiPAD router.

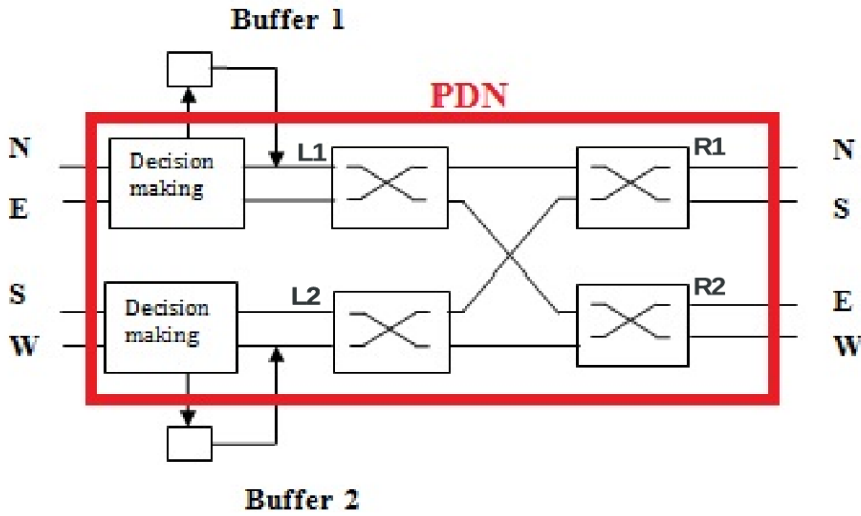


Figure 4.3 PDN structure of HiPAD router

The PDN consists of two stages with each stage having two 2x2 permuter blocks. It allocates the output ports to the flits using the information obtained from routing and prioritization units. At each permuter block of the PDN, the highest priority flit is forwarded to the desired output port whereas other flits get deflected through the remaining port. The PDN mainly has two parts: the decision making units and the port allocator as shown in Figure 4.3. The decision making unit decides which flits are to be injected into the port allocator. The decisions are taken based on the rules given in Algorithm I. The main contributions of the proposed architecture are listed below.

- **No structural isolation:** The 2x2 arbiter stages used in the PDN addresses the structural drawback of MinBSD. In HiPAD, a flit that comes through the input link is free to move in any output direction by passing through the two stages of the permutation network. There are direct paths among all the arbiters as shown in Figure 4.3. Since the

flits with higher priority can reach their desired direction without being buffered; the overall latency of the network reduces significantly.

- **Parallelisation of Independent Operations:** The routing, ejection and prioritization operations are independent of each other. Hence these three operations can be performed in parallel which reduces the critical path delay inside the router.
- **Injection from side buffers in all cycles:** The flits stored in the side buffers are injected in every cycle into the PDN. This prevents the starvation of flits in side buffers. When buffered flits are re-injected into the PDN, they are given higher priority over the flits coming from input links. Thus it is ensured that the flits which have already undergone buffering do not suffer anymore. At a time six flits are taken care of at the input of PDN, 4 flits coming from the input links and 2 flits coming from the buffers. Since injection from buffers occurs in every cycle, only a minimum side buffer size is required. The size of a side buffer is chosen as one. This method makes the proposed architecture more energy and area efficient.
- **Single cycle operation:** The entire operation within the router takes place in a single cycle. This reduces the flit latency in terms of number of cycles. Since all the operations are performed in a single cycle, the router delay increases slightly. Hence the operating frequency of the network using this router will be lesser than its counterparts having two cycle delay.

Algorithm I: Decision Making in HiPAD router

Inputs : Buffer1_out, North_in, East_in

Outputs : Buffer1_input, L1_input1, L1_input2

```

If flit in (North_in) and (East_in)
{
    If no flit in (Buffer1_out)
    {
        If PDNOutputStage(North_in) = PDNOutputStage(East_in)
        {
            Assign L1_input1 = North_in;    //output stages of PDN are R1 and R2
            Assign Buffer1_input = East_in; }
        Else
        {
            Assign L1_input1 = North_in;
            Assign L1_input2 = East_in; }
    }
    Else
    {
        If (PDNOutputStage(North_in) = PDNOutputStage(Buffer1_out)) and
        (PDNOutputStage(North_in) != PDNOutputStage(East_in))
        {
            Assign L1_input1 = Buffer1_out (high priority);
            Assign L1_input2 = East_in (low priority);
            Assign Buffer1_input = North_in; }

        Else { Assign L1_input1=North_in (low priority);
            Assign L1_input2 = Buffer1_out (high priority);
            Assign Buffer1_input = East_in; }
    }
}
Else
{
    If flit in (North_in) and (Buffer1_out)
    {
        Assign L1_input1 = North_in (low priority);
        Assign L1_input2= Buffer1_out (high priority); }
    Else If flit in (East_in) and (Buffer1_out)
    {
        Assign L1_input1= Buffer1_out (high priority);
        Assign L1_input2 = East_in (low priority); }
    Else
    {
        If flit in (Buffer1_out) Assign L1_input1= Buffer1_out ;
        Else If flit in (North_in) Assign L1_input1= North_in ;
        Else If flit in (East_in) Assign L1_input2= East_in ;
        Else Assign L1_inputs =NULL;
    }
}

```

4.2 Simulation methodology

In order to perform simulations, the conventional VCR based Booksim is modified to model the proposed HiPAD router as mentioned in Section 4.2 [3],

[70]. The average performance parameters obtained from simulations are compared with that of MinBSD and MinBD. Experiments are conducted with 1-flit packets. Since deflection routers route each flit independently, every flit contains necessary control information needed for routing. For all the three methods, the flit channel is 140-bit wide: 128-bit data field and 12-bit header field. Necessary reassembly mechanism is used for handling out-of-order delivery of flits. Simulations are conducted with different standard synthetic traffic patterns like uniform, bit-complement, transpose, tornado, shuffle and neighbour and the average values of packet latency, deflection rate and throughput are observed.

Simulations are also performed using multi-programmed workloads from SPEC CPU2006 benchmark suite [71]. Benchmark mixes (M1 to M7) in varying proportions of MPKI given in Table 4.1, are run on a 64 core CMP simulator [70]. The network events generated during these simulations are given as traffic to HiPAD and MinBSD based NoCs and the performance benefits of the proposed method are analysed.

Table 4.1 Benchmark mixes (M1 to M7) in various proportions of SPEC CPU 2006 applications.

Applications	Benchmark mixes						
(%)	M1	M2	M3	M4	M5	M6	M7
Low MPKI	100	0	0	50	0	50	31
Medium MPKI	0	100	0	0	50	50	31
High MPKI	0	0	100	50	50	0	38

4.3 Results and discussions

Average flit latency

Higher latency of flits in the network increases the stall time of applications resulting in their throttling and poor application level performance. Hence for better performance, the average flit latency should be minimal.

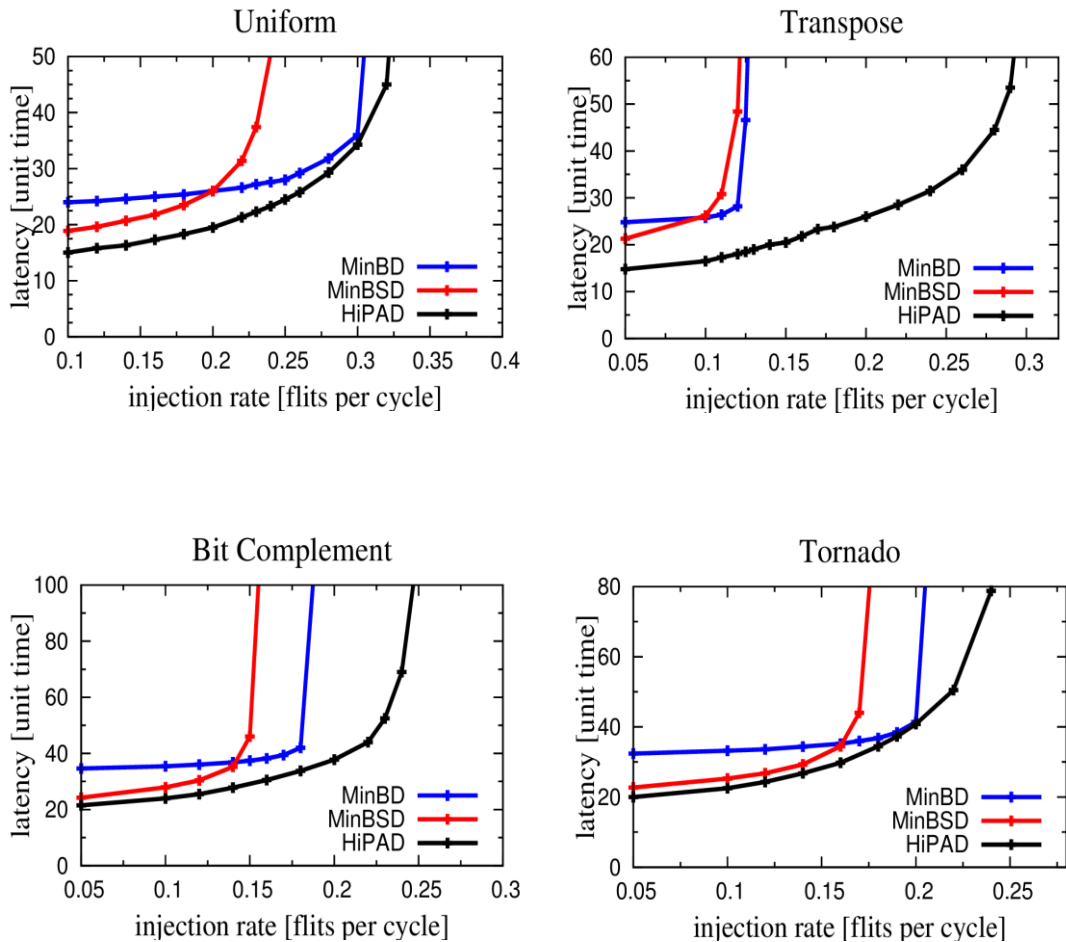
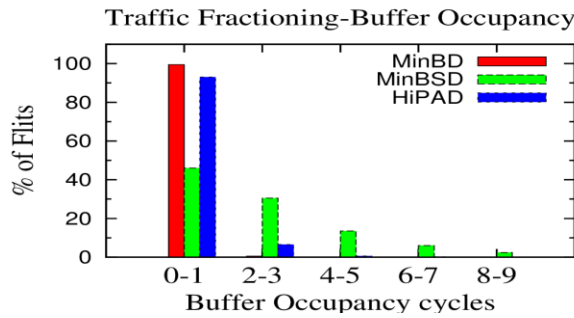
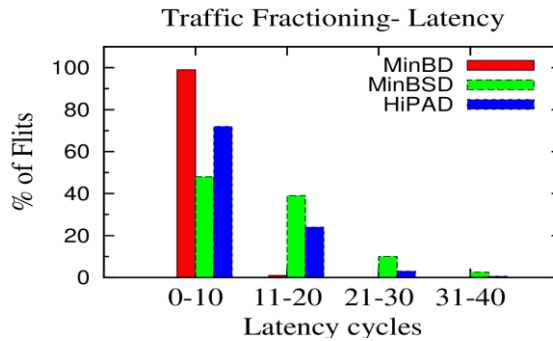


Figure 4.4. Average latency for various synthetic traffic patterns using HiPAD MinBSD and MinBD in 8x8 mesh NoC.

Figure 4.4 shows the average flit latencies of HiPAD, MinBSD and MinBD routers for four typical synthetic traffic patterns on 8x8 mesh network. Flit latency is calculated in unit time by multiplying latency in cycles with time for one cycle (which is different for MinBD, MinBSD and HiPAD). HiPAD exhibits the least average flit latency for all values of injection rates and network saturation point is extended by 33% compared to MinBSD and 6.6% compared to MinBD. Minimization of SBOT using intelligent logic for buffering and deflection accounts for the reduction in average latency. From Figure 4.5 (a), we see that using HiPAD router, SBOT of 93% of the flits is 1 cycle or less and 72% of flits have a latency less than or equal to 10 cycles.



(a)



(b)

Figure 4.5 Traffic Fractioning based on (a) Buffer Occupancy (b) Average Latency for MinBD, MinBSD and HiPAD in 8x8 mesh NoC.

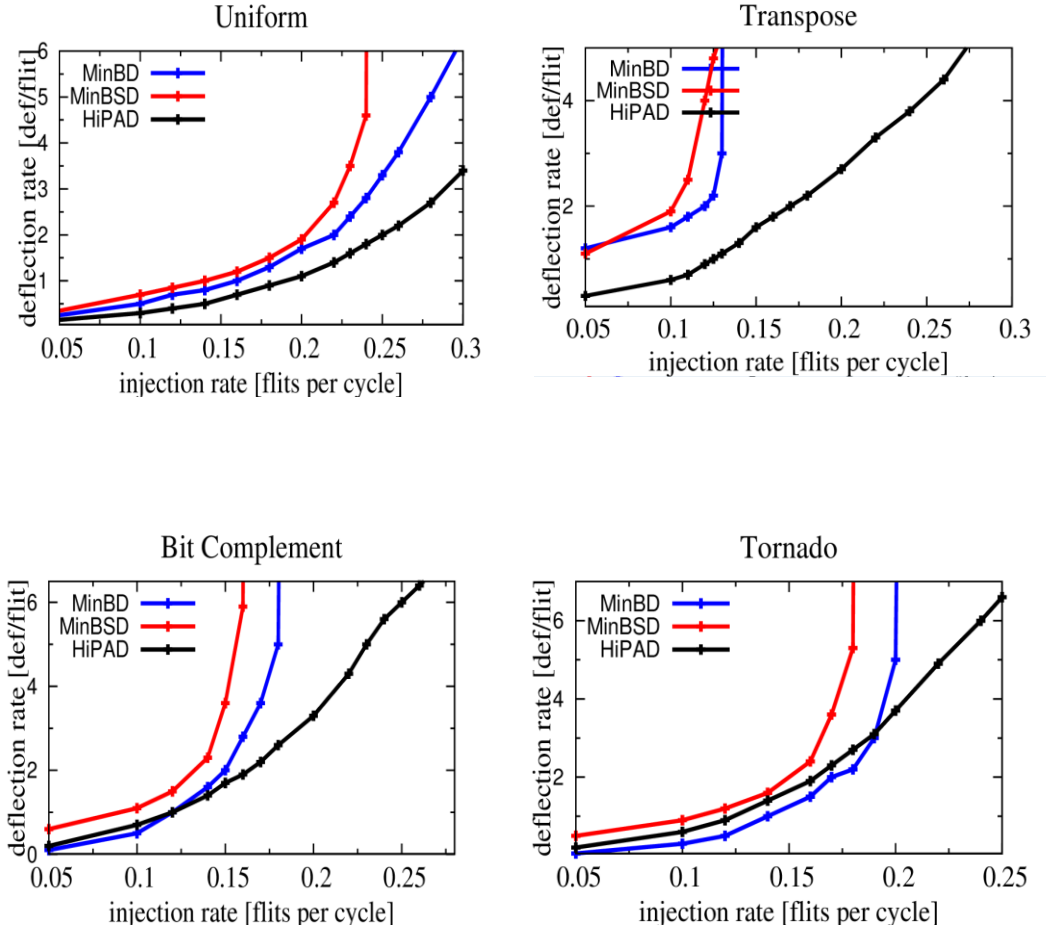


Figure 4.6 Average deflection rate for various synthetic traffic patterns using MinBD, MinBSD and HiPAD in 8x8 mesh NoC.

Average deflection rate

The graph in Figure 4.6 shows that the deflection rate of the proposed work is lower than that of MinBSD for various injection rates. If a flit undergoes lesser deflections, it reaches its destination in fewer number of cycles i.e.

average latency is also low. But sometimes, the waiting time of flits in the buffers may also lead to increased latency. In the proposed work, since one among the flits asking for the same target arbiter is stored in the side buffer due to intelligent decision making, the deflection of flits is reduced.

Real applications

Following the same trend of synthetic traffic functions, HiPAD shows a reduction of 8% in average latency and 34% in average deflection rate compared to MinBSD for real application workloads. This is depicted in Figure 4.7 (a) and (b).

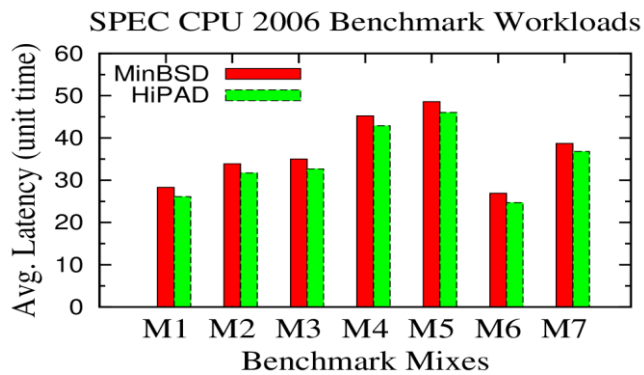
4.4 Router delay, static power and area

Table 4.2 Comparison of router delay, area and static power of MinBSD and HiPAD routers

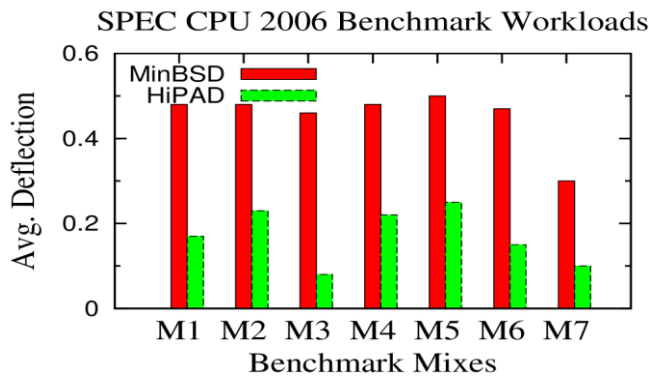
	Router delay (nsec)	Area (μm^2)	Power (μW)
MinBSD	2.25	3300	270
HiPAD	2.56	3999	306

Verilog models of the proposed router as well as MinBD and MinBSD routers are synthesized using Synopsys design compiler with 65nm technology to extract the pipeline latency in each case. Router delay is computed as the total time taken by a flit to move from an input port of a router to an output port through various functional units. The critical path delay of 2.1 nsec for the second stage of MinBD determines its operating frequency. Being single cycle routers, the critical path delay of MinBSD and HiPAD are 2.25 nsec and 2.56

nsec respectively. The critical path of the proposed technique is 14% longer than MinBSD, which is due to the additional delay of 0.38 nsec for the decision making unit introduced into the router pipeline. Since the decision making logic uses conditional check at various levels to choose between the input flits for output port allocation and buffering, it increases the router area and static power by 21% and 13.4% respectively with respect to MinBSD.



(a)



(b)

Figure 4.7 Comparison of (a) Average latency and (b) Average Deflection rate for real applications using HiPAD and MinBSD in 8x8 mesh NoC.

4.5 Chapter summary

This chapter presents a single cycle deflection router architecture for enhanced performance in two dimensional mesh based NoCs. State-of-the-art deflection routers with single cycle operation were experimentally analysed and a few performance limiting factors were identified. Effective solutions like intelligent buffering of flits prior to output port arbitration and reducing the side buffer occupancy time of flits were proposed in the new router. From experimental results, it is seen that the proposed method achieves significant improvement in performance with reduction in average latency and deflection rate compared to single cycle deflection routers proposed earlier.

Chapter 5

Fault Tolerant Deflection Routing

Abstract

This chapter presents a novel, energy efficient deflection routing technique that can tolerate permanent faults in NoC components. The new routing approach is based on reallocation of healthy output ports for flits whose default output ports are faulty. Evaluation of the proposed method on a mesh NoC for various fault rates report reduced flit deflection rate and hop power which brings about significant reduction in dynamic power consumption at the inter-router links compared to state-of-the-art fault tolerance techniques. An enhanced fault tolerant router model that escalates the availability of fault-free paths in the router is also proposed.

The greatest challenge of NoC designs is to meet tough latency and throughput targets, under stringent area and power budgets. Along with shrinking dimensions to deep sub-micron level feature sizes, the reliability of the chip degrades as well. At the chip level, faults may be permanent or transient in nature. Permanent faults that result in chip failure are caused by physical damages such as manufacturing defects and device wear-out. Operational stress induces a phenomenon called electromigration which causes material deformations and loss of connections in a circuit. According to the ITRS 2009 Interconnect Report [72], electromigration is the main cause of on-chip metal interconnect reliability loss in current and future CMOS technologies. Unpredictable causes like power grid fluctuations and particle hits may cause transient faults in chips. As the sole medium of communication in multicore chips, the NoC should be designed to tolerate transient and permanent failure of components and provide reliable communication between healthy nodes inside the chip.

Faults can cause malfunctioning of NoC routers as well as the inter-router links. Discontinuities in the links cause hindrance to the transit of flits through paths obtained during the routing process. In order to tolerate faults and facilitate flit movement, routers use additional hardware and adaptive algorithms to redirect flits through fault-free paths. The inherent path diversity of mesh topology allows most of the routers to be accessible through multiple paths from any other router in the network. In this chapter, an energy efficient deflection routing technique capable of tolerating permanent faults in router logic as well as inter-router links is proposed. A simple logic unit placed next to the output port allocation stage in the deflection router pipeline reallocates flits from faulty ports to healthy ports. This method exploits the path diversity of

mesh topology and thereby ensures that majority of the flits reach destination through minimal paths.

The fault tolerance mechanism used in two recent deflection routers referred to as FaFNoC [58] and Maze routing [59], are studied in detail. These methods use the PDN based deflection router architecture proposed in CHIPPER, which is adopted as the baseline router architecture in this thesis [11]. FaFNoC is a fault tolerant deflection routing technique which uses a modified form of the PDN introduced in CHIPPER for output port allocation. As shown in the examples of Figure 5.1, the interconnections between the four permuter blocks (P1, P2, P3, P4) inside the PDN are adaptively enabled or disabled such that faulty output ports are not allocated to flits. The flit header is extended with additional bits to transmit information about faulty links between routers. Maze routing technique also requires significant extension of flit header with additional bits for carrying information regarding guaranteed flit delivery and unreachable nodes in the network.

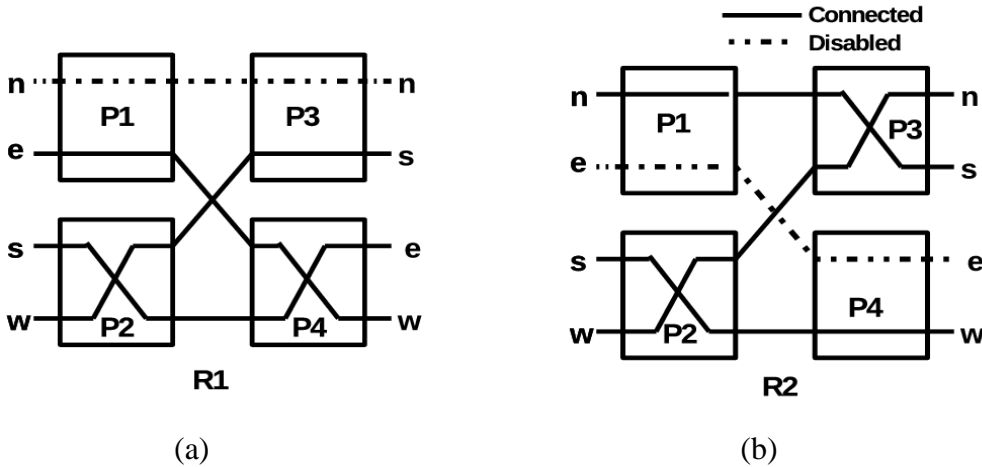


Figure 5.1 PDN in FaFNoC router with (a) faulty north port
(b) faulty east port [58]

Simulations are conducted on 8x8 mesh NoCs using FaFNoC and Maze routing methods using uniform traffic pattern. From the results, it was observed that the structural inefficiency of FaFNoC, algorithmic inefficiency of Maze routing and the area overhead of these two methods limit their performance. A detailed analysis of these limitations is given below.

- **Structural inefficiency:** In the FaFNoC router, a fault handler block disables some of the connections between permuter blocks P1, P2 and P3, P4 in the PDN so that flits do not move to faulty output ports during port allocation. In router R1 of Figure 5.1(a), the north link is faulty and the connection between P1 and P3 (shown as dashed line) in the PDN is disrupted so that north output is not available for port allocation. As a side effect of this technique, a flit from the east input port of P1 will be prohibited from taking up the south output port connected to P3 if it is the flit's preferred output port. Such a flit is compelled to choose either the east or west output port, which results in deflection of the flit to an unproductive direction. Similarly, in router R2 of Figure 5.1(b), where the east link is faulty, the connection between P1 and P4 is disabled. Due to this, north input flits are prohibited from moving to the west output port. In FaFNoC, flits dynamically switch between XY and YX routing in order to choose output ports with lesser traffic congestion. This increases the possibility of flits taking more turns (eg. from north to east or west) compared to static XY algorithm. The partially disabled connections between permuters do not support these turns, hence deflection rate of flits increases. From evaluations on 8x8 mesh, it is found that under low traffic injection rates (less than 0.1 flits/cycle/core)

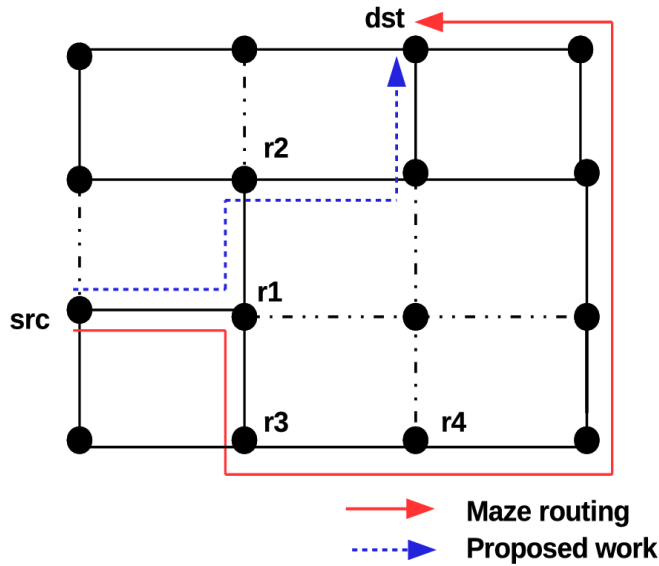


Figure 5.2 Demonstration of path traversed by flit from source (src) to destination (dst) in a 4x4 mesh NoC using Maze routing and proposed work.

and 10% faulty links, 22% of overall flit deflections is due to this structural inefficiency of the PDN explained above. In the proposed method, these issues are addressed by a superior fault tolerant logic unit.

- **Algorithmic inefficiency:** In Maze routing, the routing algorithm computes the preferred output port for each input flit. Maze routing algorithm identifies the flits whose desired output ports are faulty and allots them at random to other healthy output ports either towards right (clockwise direction) or left (anti-clockwise direction) of its desired port. To achieve livelock freedom, the chosen direction is stored in the flit header and this direction rule is followed by the flit in all subsequent routers through which it traverses. Figure 5.2 shows a possible path of a

flit from source (src) to destination (dst) using Maze routing. From src, a flit traverses along the x direction towards dst and reaches router r1. From r1, the preferred output for the flit is in the east which is assumed to be faulty (shown as a black dotted line). So the flit randomly chooses a fault-free output port either to the left (north port) or to the right (south port) of the east port with equal probability. The solid red line shows the path followed by a flit which deflects through the south port of r1 to router r3. In all subsequent hops, this flit chooses the healthy port to the right of the line joining the current router to the destination router. Accordingly, the flit reaches its destination in 8 hops. In our evaluation of Maze routing technique, we find that 34% of flits which deflect due to faulty output ports traverse non-minimal paths to reach their destinations on account of this random selection of direction. These additional deflections increase network activity factor and consume dynamic power. Larger network sizes and higher fault rates result in increased flit latency and throughput degradation. In order to overcome this algorithmic inefficiency, the proposed routing algorithm explores shortest available paths in the presence of faulty links and delivers the flit from src to dst in 4 hops (shown as blue dotted path in Figure 5.2).

- **Area overhead:** Fault tolerant router designs should adhere to strict area and power constraints. FaFNoC and Maze routing techniques do not employ routing tables and hence occupy very less chip area. In Maze routing, information such as Manhattan distance between current router and destination, direction of deflection (left or right) and coordinates of current node are coded in a flit's header using additional bits. In an 8x8 mesh NoC, flit header extension increases the flit channel width by 10%.

In FaFNoC, hop count and fault status of a flit are also transmitted in the header which increases the channel width by 8%. Widening of flit channels leads to an almost quadratic increase in the router area [73]. The proposed method reduces this area and wiring overhead by minimising the fault information transmitted in the flit header.

5.1 Coarse grain fault model

The proposed fault tolerant router is represented using a coarse grain model which effectively represents faults in bidirectional links as well as component failures inside routers. Figure 5.3 shows a 4x4 mesh NoC with some of the unidirectional links as faulty (shown as red line). A fault in either of the input or output channels of a router is represented by disabling both input and output channels in that direction (shown as crossed). This ensures that equal number of active input and output ports are there for a router which is essential for deflection routing. A failure in any of the datapath elements of a router is effectively modelled by converging it to a fault in a specific input-output port. The link corresponding to this port is then disabled by setting appropriate ports of the two routers at its ends as faulty. Failures occurring in crucial functional units of a router are fatal to router operation. Such routers are fully disconnected from the network by disabling all its input-output ports. Four fault flags (1 bit each) are used in each router to represent the functional correctness of output ports in north, south, east and west directions. The fault flag corresponding to a faulty input-output port of a router is set. At the same time, the fault flag of the corresponding input-output port of the adjacent router is also set. Fault flags corresponding to healthy ports of routers are reset. The values of fault flags are set or reset during fault diagnosis phase, which is not part of the thesis. Hence, it

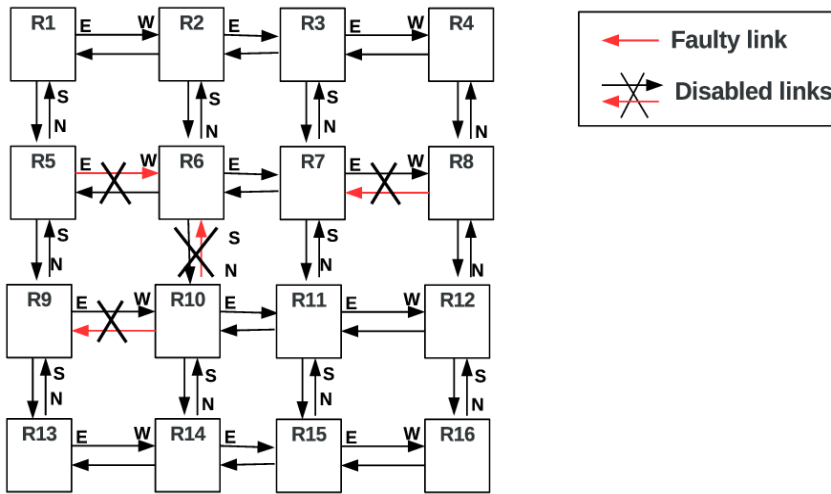


Figure 5.3. Coarse grain fault model

is assumed that the flags are updated using online test methods or during system reboot. The functional blocks used for port re-allocation read the status of the fault flags for activating their circuits as explained later in this chapter.

5.2 Router architecture

A novel fault tolerant deflection router architecture with two stage pipeline is proposed here. The input stage (stage 1) of the router pipeline consists of functional blocks for route computation, ejection and injection of flits. The output stage (stage 2) extends from register B to register C. Figure 5.4 depicts the output stage of a router having four bidirectional ports in a mesh NoC. It consists of a PDN whose four output lines are connected to the Fault Tolerant Logic Unit (FTLU). Latches (L1, L2, L3, L4) are placed between each pair of output lines of the FTLU.

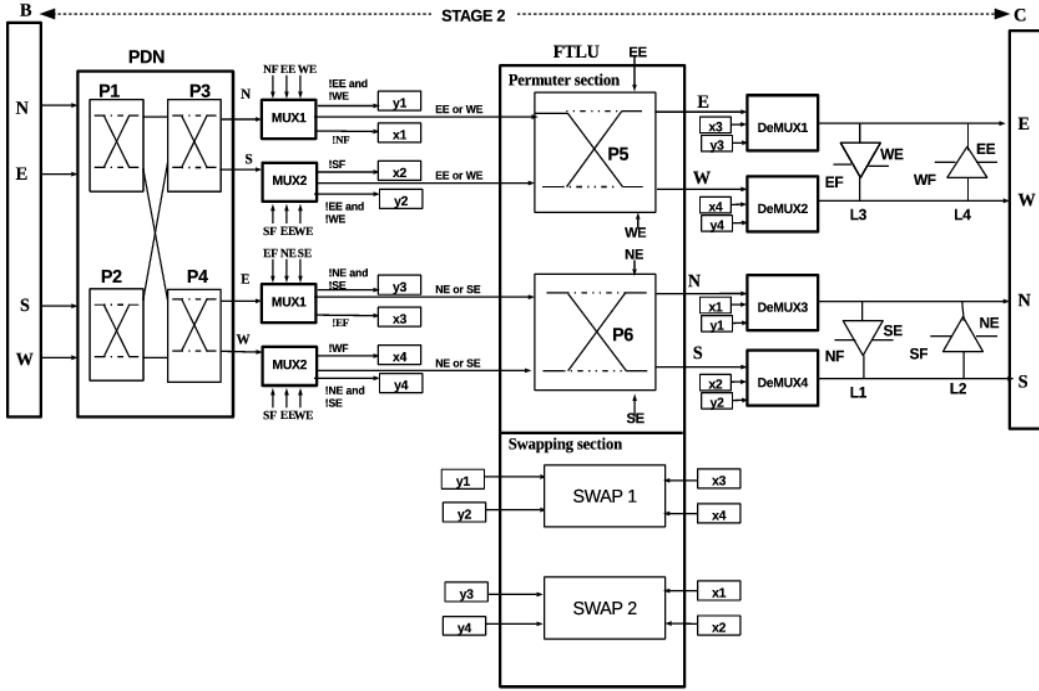


Figure 5.4 Output stage of the proposed router.

In order to facilitate fault tolerance, healthy output ports are reallocated to flits which are allocated to faulty ports by the PDN. The proposed architecture enables two types of displacement of a flit from a faulty to a healthy port:

- (1) in an orthogonal direction using FTLU
- (2) in the geometrically opposite direction using a latch.

Either of these two displacements or a combination of both may be used for reallocation such that all flits reside in healthy ports of register C at the end of the router pipeline. Starting with port allocation by the PDN, the port reallocation mechanisms are explained in detail.

5.2.1 Port allocation using PDN

At the end of the input stage, four flits that are ready for output port allocation are available in a pipeline register B as shown in Figure 5.4. These are either flits in transit through the router or newly injected flits from the local processing core. The PDN reads these four flits and maps them to the four output lines in north, south, east and west directions using four permuter blocks P1, P2, P3 and P4 as per the logic used in CHIPPER [10]. Each permuter block has two inputs and two output ports. At each of these blocks, the flit with higher priority is allotted to an output port of its choice and the flit with lower priority is allotted to the remaining output port. In this work, higher priority is assigned to the flit with lesser number of hops to its destination.

5.2.2 Port reallocation using FTLU

The FTLU consists of two sections which are placed in parallel in the router pipeline. The upper section consists of two permuter blocks viz. P5 and P6. P5 connects flits from north and south output lines of the PDN to east and west output lines of the FTLU. Similarly, P6 maps flits from east and west outputs of PDN to north and south outputs of the FTLU. In the lower section of FTLU, two swapping blocks viz. SWAP1 and SWAP2 are provided. In SWAP1 block, a flit in the faulty north or south port of PDN is swapped with a flit in the healthy east or west port. Similarly, SWAP2 interchanges flits between faulty east or west ports and healthy north or south ports. In each router, four fault flags NF, SF, EF and WF are used to represent the healthiness of the output ports in the north, south, east and west directions respectively. The flag bit corresponding to a faulty output port will be set. Port allocation by the PDN is

done on the basis of computed route of a flit and it does not consider the fault status of the router's output ports.

As a result, some flits may be assigned to the faulty ports at the output of PDN. The FTLU reallocates such flits to healthy ports in an orthogonal direction with respect to the faulty port. In a router with four output ports, the north and south ports are orthogonal to the east and west ports. Algorithm II describes the steps followed in the FTLU. Entry of flits into the FTLU is restricted by multiplexers that are controlled by fault flags as shown in Figure 5.4. If a flit is present at a faulty output of the PDN, the corresponding fault flag is 1 and this gates the flit into one of the two sections of the FTLU (line numbers 4-14 in Algorithm II).

A flit at a faulty port of the PDN either passes through the permuter section or the swapping section for port reallocation. The permuters P5 and P6 in the upper section are activated for port reallocation only if any one of their output ports are empty (line numbers 19-24 in Algorithm II). Four 1 bit flags, NE, SE, EE and WE represent whether the output ports in north, south, east and west directions respectively are empty or not. To avoid livelock problem, it is necessary to reallocate a flit to an output port other than in its input direction. Based on this restriction, a flit may try to reallocate to a orthogonal port in the FTLU, but the desirable port maybe occupied by another flit from the PDN. For energy efficient reallocation in the FTLU, flits from faulty ports are given higher priority to occupy fruitful ports. By activating a swapping block, such flits are swapped with the flits that reside in their productive ports (line numbers 25-26 in Algorithm II). The flits from the healthy ports of the PDN are available to the swapping blocks at registers x1, x2, x3 and x4 and flits from faulty ports are available at y1, y2, y3 and y4. The individual outputs of the permuter section and swapping section of the FTLU are merged into four output lines using

demultiplexers. In a fault-free router whose four input-output ports are healthy, flits from the output of PDN bypass the FTLU and move to their respective output ports through the demultiplexers (line numbers 27-28 in Algorithm II).

In a mesh NoC, the routers at the corners and edges have two and three pairs of input-output ports respectively. In order to maintain simplicity, homogeneous router architecture is used throughout the NoC, i.e. the architectures of corner and edge routers are similar to that of the central routers. Route computation of input flits using XY and YX methods is in such a way that the forbidden output ports of edge and corner routers are not allocated by the PDN. In the FTLU also, the output ports in the forbidden directions are treated as faulty by setting the corresponding fault flag. For example, in the router at the top right corner of a mesh network, north and east output ports are set as faulty ($NF=1$, $EF=1$). Similarly, for a router at the left edge of the mesh, the west output is set as faulty ($WF=1$). The total area and power consumed by the NoC can be reduced by using customized architectures for routers at the edges and corners.

5.2.3 Port reallocation using latches

A latch is used to transfer flits from one port to another when it is enabled. The term 'latch' refers to a circuit that performs this function. Edge triggered flip-flops or level triggered latches could be suitably used for implementing it in hardware. In the proposed method, a latch is used to reallocate a flit from a faulty port to a healthy port in the geometrically opposite direction. In a router with four input-output ports, the two pairs of geometrically opposite ports are east and west, north and south. When a flit is deflected to a

Algorithm II: Port reallocation using FTLU

```

1: Inputs: pdnOutput (N, S, E, W), Faultflags (N, S, E, W), Emptyflags (N, S, E, W);
2: Outputs: ftluOutput (N, S, E, W);
3: for <each port in N, S, E, W> do
4:   if (faultflag (port)) then
5:     if port is (N or S) and EmptyFlag (E or W) then
6:       P5Input(port) ← pdnOutput (port)
7:       // Assigning PDN outputs to permuter P5 or P6 inputs
8:     else
9:       SW AP 1Input(y1 or y2) ← pdnOutput (port)
10:      // Assigning PDN outputs to swap circuit inputs as in Figure 5.4
11:    end if
12:    if port is (E or W) and EmptyFlag (N or S) then
13:      P6Input (port) ← pdnOutput (port)
14:    else
15:      SW AP2Input (y3 or y4) ← pdnOutput (port)
16:    end if
17:  else
18:    (x1, x2, x3, x4) ← pdnOutput (port)
19:    // Assign PDN output to x1 or x2 or x3 or x4 as in Figure 5.4
20:  end if
21: end for
22: Emptyflag (P5Output1 or P 5Output2) ← HighPriority (P5Input1 or P5Input2)
23: Remaining (P5Output1 or P 5Output2) ← LowPriority (P5Input1 or P5Input2)
24: FLB(P5Output1) = 1;          FLB(P5Output2) = 1;    // FLB set to YX routing
25: Emptyflag (P6Output1 or P6Output2) ← HighPriority (P6Input1 or P6Input2)
26: Remaining (P5Output1 or P5Output2) ← LowPriority (P5Input1 or P5Input2)
27: F L B(P6Output1) = 0;        F L B(P6Output2) = 0    // FLB reset to XY routing
28: SWAP1(y1) ↔ (x3 or x4); SWAP1(y2) ↔ (x3 or x4); FLB(x3 or x4) = 1;
29: SWAP2(y3) ↔ (x1 or x2); SW AP2(y4) ↔ (x1 or x2); FLB(x1 or x2) = 0;
30: // Swapping function
31: ftluOutput(N ) ← (P5Output(N) or x1 or y1);
32: ftluOutput(S) ← (P5Output(S) or x2 or y2);
33:
34: ftluOutput(E) ← (P6Output(E) or x3 or y3);
35: ftluOutput(W) ← (P6Output(W) or x4 or y4);
36: // Demux outputs

```

direction which is geometrically opposite to its productive port, the distance to its destination increases. Hence, this kind of port reallocation is only done at the final stage of the router pipeline to reallocate flits that remain in faulty ports at the output of FTLU. As shown in Figure 5.4, latches L1 and L2 are connected between north and south output lines of the router whereas L3 and L4 are connected between east and west output lines. As a result of swapping operation in the FTLU, a flit from a healthy port is transferred to a faulty port in its orthogonal direction. A latch from the faulty port is enabled to transfer this flit to a healthy port in the geometrically opposite direction. The proposed port reallocation mechanism in routers with at most three faulty output ports is explained using examples.

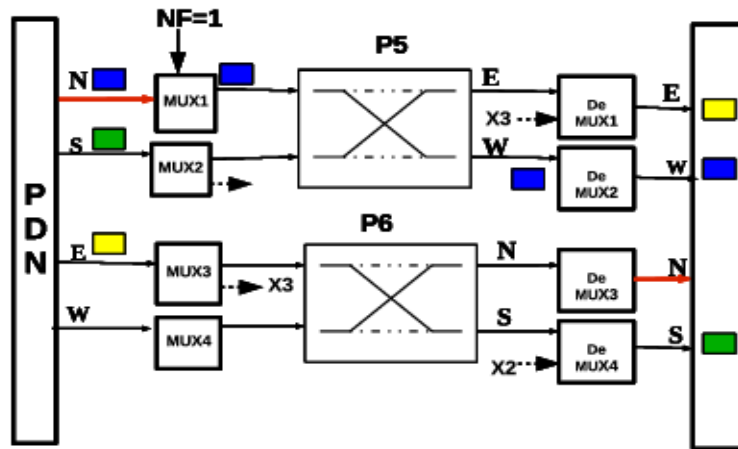
Routers with one faulty port: In Figure 5.5(a), it is assumed that the north port of a router is faulty ($NF=1$). At the output of PDN, the north, south and east ports are occupied by flits. The flit from the north port is directed to the permuter P5 in the FTLU by MUX1. P5 allocates this flit to the west output port which is vacant (line numbers 5-6 in Algorithm II). The flits in the south and east output ports of PDN bypass the FTLU through registers x2 and x3 respectively and move to the pipeline register C via demultiplexers (DeMUX4 and DeMUX1 respectively).

A special case of a router with one faulty port is shown in Figure 5.5(b). Here, flits occupy the north ($NF=1$), east and west output ports of PDN and south port is empty. We assume that the flit in the north port entered the router through the south input port. This flit is restricted from tracing back through the south port due to reasons which are explained in Section 5.3. Therefore, east or west ports are the only suitable locations for reallocating this flit. Assuming that

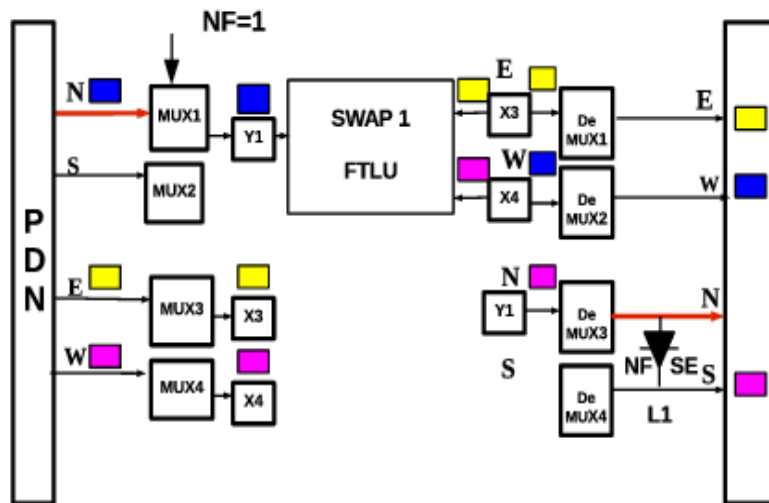
the north flit prefers the west port which is non-empty, the swapping circuit (SWAP1) in the FTLU is activated to interchange flits between north (in register y1) and west ports (in register x3) (line number 25 in Algorithm 2). After swapping, outputs of the FTLU are available in registers y1, x3 and x4. The flit in y1 is joined to the north output line by DeMUX3, and then reallocated to the healthy south port by enabling latch L1.

Routers with two faulty ports: In Figure 5.5(c), the north and west ports of router are faulty ($NF=1$, $WF=1$). Here, a maximum of two flits enter the router through the south and east ports. After port allocation by the PDN, each flit occupies one among the four output ports of the PDN. The different situations that may arise are categorised here.

- **Case 1:** The two flits are at north and south output ports of PDN. Since north port is faulty ($NF=1$), MUX1 is enabled and the flit is passed to permuter P5 in the FTLU. As shown in Figure 5.5(c), P5 reallocates this flit to the east port which is healthy and vacant ($EF=0$, $EE=1$). The flit at the south port bypasses the FTLU through register x2 and moves to its position in pipeline register C.
- **Case 2:** The two flits are at east and west output lines of PDN. Here, MUX4 is enabled since fault flag of west port is high ($WF=1$). The flit passes through MUX4 and the permuter P6 in the FTLU reallocates it to the south port which is healthy and vacant.
- **Case 3:** One flit is at north and the other flit is at west output line of PDN. Permuter P5 reassigns the north flit to the east port which is healthy



(a)



(b)

Figure 5.5: Reallocation of output ports in routers with (a) one faulty port (north) using permuter (P5) (b) one faulty port (north) using swap circuit (SW1) and latch (L1).

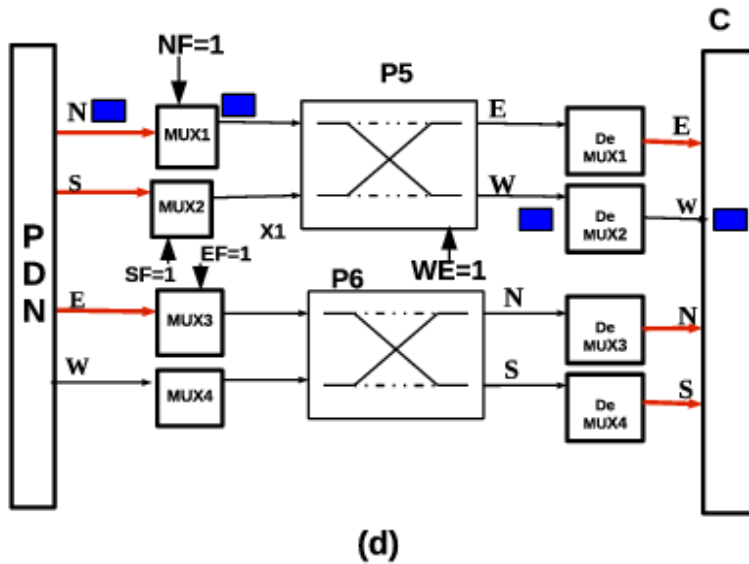
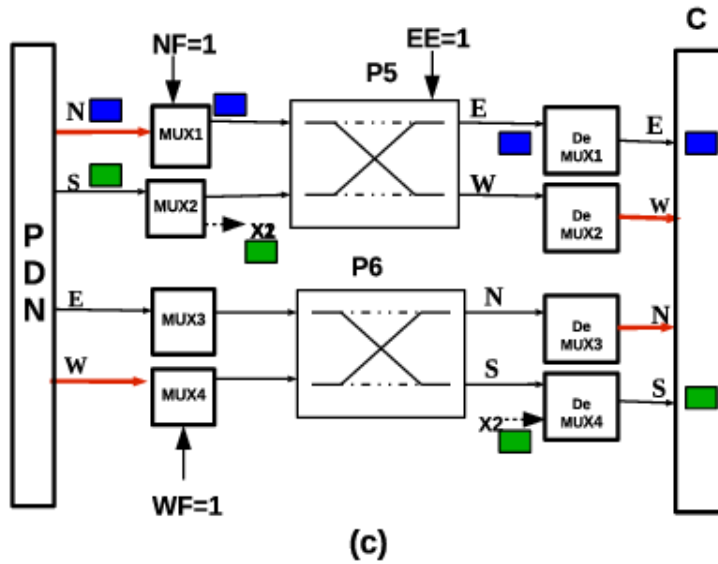


Figure 5.5: Reallocation of output ports in routers with (c) two faulty ports (north and west) (d) three faulty ports (north, south and east) .

and empty. Similarly, the west flit is reassigned to the south port through permuter P6. After reallocation, these flits are moved to the south and east positions in pipeline register C.

Routers with three faulty ports: When three pairs of input-output ports in a router are faulty, one flit needs to be routed through the single healthy port. The constraint placed on port reallocation is overridden in this special case. At the output of PDN, if the flit occupies a faulty port, it is re-allocated to the single healthy port either by the FTLU or latches. The Figure 5.5(d) shows a router where all three ports except the west port are faulty. A flit at the north output of PDN is reassigned to the west port by permuter P5 in the FTLU.

5.2.4 Fault Loop Bit

In the proposed method, the flit header is extended by an additional bit called the Fault Loop Bit (FLB). Productive output ports of a flit in a router are computed by referring to the value of FLB in the flit header. The value 0 and 1 for FLB represents XY and YX route computation methods respectively. The following section explains how FLB is toggled during port reallocation to prevent livelock.

5.3 Livelock problem

In a deflection router, a flit maybe reallocated to an unproductive port due to a fault in its productive direction. If a single deterministic method (e.g. XY routing) is used for route computation in every router, the flit maybe routed back to the same fault location repeatedly. This results in livelock, a problem

which should be completely eliminated by the routing algorithm. The proposed technique ensures livelock freedom for mesh NoCs with all types of fault patterns except those with gateway routers explained in Sections 5.3.2 and 5.3.3. For this, two constraints are imposed during port reallocation.

- Disallowing the reallocation of flits to output ports in the same direction as their input ports.
- Adaptive switching between XY and YX routing methods to change the direction of traversal of the flit after port reallocation.

The first constraint says that a flit is not allowed to back trace to its input direction after port reallocation. For example, a flit which entered the router through the east input port will not be assigned to the east output port during port reallocation by the FTLU or latches. Such flits may choose one of the two remaining ports that are eligible for reallocation. In a router with three faulty ports, a flit entering through the single healthy port is exempted from this restriction. Initial route computation of a flit is based on XY routing i.e. the default value of FLB is 0. On obtaining a faulty port at the output of PDN, the flit passes through FTLU or latch for reallocation to a healthy port. After reallocation, if the flit is assigned to a port in the horizontal direction (east or west) of the mesh, its FLB value is set to 1. Accordingly, productive ports for this flit are computed using YX method in the next router, which implies that the flit makes a vertical hop to a router in the adjacent row. FLB of a flit traversing through a port in the vertical direction (north or south) of the mesh will be reset to 0. Hence, XY method is used for route computation in the next router, which moves the flit to a router in the next column. Whenever a flit bypasses the port

reallocation stage in a router, its FLB is reset to the default value i.e. 0 irrespective of the port to which it is allocated.

The example in Figure 5.6(a) illustrates how toggling of FLB and restricting reverse movement of flits guarantees livelock freedom of flits in a network with multiple faulty links. A flit which is destined to router (6, 6) starts from router (3,0) and initially follows XY routing. At (3, 2), the east output port is faulty. By port reallocation using FTLU, the flit is reassigned to the south output port. Since the flit traverses to (4, 2) through the south port of (3, 2), FLB of flit remains at the value 0. So, next route computation at (4, 2) is based on XY routing and the flit hops to router (4, 3) through the east output port. Again, the flit encounters a fault in the productive east port at router (5, 3). This flit cannot be reallocated by the FTLU since south port of (5, 3) is faulty and the north port is not allowed. So the flit is reallocated to the west port by a latch and its FLB is set to 1. This implies that route computation in router (5, 2) is based on YX routing. Port reallocation for the flit at router (6, 4) is similar to that in router (3, 2). The path followed by the flit upto its destination router (6, 6) is shown in red line in Figure 5.6(a).

5.3.1 Proof of livelock freedom

For establishing livelock freedom of the proposed fault tolerant routing technique, it is assumed that there exists at least one fault-free path between the current router and destination router of the flit. Following the steps given below, we prove that a flit traversing the network from router (x1, y1) reaches its destination at router (x2, y2) in finite number of hops.

STEP 1 : The flit starts from (x_1, y_1) in the horizontal direction ($FLB = 0$, XY routing) and encounters a fault in the productive port (east or west) of an intermediate router (x, y) .

STEP 2 : Since reallocation of the flit to its input direction is forbidden, only ports in the vertical direction (north or south) are eligible for reallocating the flit. Moving along one of the vertical ports, the flit reaches a router $(x \pm 1, y)$ with $FLB = 0$ (line numbers 22-24 or 26 in Algorithm II).

STEP 3 : In router $(x \pm 1, y)$, the flit tries to obtain a port in horizontal direction (XY routing). If the desired port is healthy, the flit succeeds in traversing horizontally. The co-ordinates of the next router are $(x \pm 1, y \pm 1)$ which is nearer to (x_2, y_2) than (x, y) and $FLB = 0$.

STEP 4: If the desired port in the horizontal direction is faulty, the flit is again reallocated to the vertical port other than its input (port reallocation in orthogonal direction by FTLU). This places the flit at router $(x \pm 2, y)$ with $FLB = 0$. Again STEP 3 or 4 are repeated till STEP 3 succeeds and the flit reaches a router which lies in the same column as the destination i.e. $y = y_2$.

STEP 5: Now the flit tries to obtain a port in the vertical direction (north or south). If it encounters a faulty port in an intermediate router (x, y_2) , the FTLU reallocates the flit to the east or west port (in orthogonal direction) and $FLB = 1$. The co-ordinates of the next router is $(x, y_2 \pm 1)$.

STEP 6: The next route for the flit is computed using YX method. If the north or south port of the router is healthy, the flit traverses along the vertical direction

to the next router $(x \pm 1, y2 \pm 1)$ and $FLB = 0$. Again STEP 3 or 4 are repeated till STEP 3 succeeds.

STEP 7: If the desired port in the vertical direction is faulty, the flit is reallocated to a horizontal port other than its input (port reallocation in orthogonal direction by FTLU). This places the flit at router $(x, y2 \pm 2)$ with $FLB = 1$. Again STEP 6 or 7 are repeated till STEP 6 succeeds and $x = x2$ and in STEP 3, $y = y2$. Thus the flit reaches destination $(x2, y2)$.

In general, if a flit is deviated from its productive path due to a faulty port, it is again guided towards its productive direction by adaptive switching between XY and YX routing methods. Since the flit is not allowed to back trace in its input direction after port reallocation, it will progress towards destination and reach there in finite number of hops.

5.3.2 Disconnected routers

When all the four output ports in a router are faulty, no flit enters or leaves the router and injection of new flits from local core is also throttled. Such routers are apparently disconnected from rest of the network. A scheme is proposed to eliminate flits that are destined to disconnected routers in the network. A flit qualifies for elimination if it fails in its attempt to enter the destination router through all the four input ports due to fault. The flit header is extended by a four bit field called Expiry field (EX field). Each bit in the EX field represents an attempt to reach destination router through the north, south, east and west ports respectively. Initially, the four bits are reset to zero. If a flit reaches a router at one hop distance from the destination and gets deflected to an

unproductive direction due to a faulty port, the corresponding bit of EX is set as 1. When the four bits in EX field of a flit are set, it is ready for elimination. A functional block called 'Kill Block' in each router examines the EX field of all input flits and erases the qualified flits from the input buffer. Since Kill block can be placed as a parallel block in the input stage of the router pipeline, it does not introduce additional delay in the router's critical path. However, extension of flit header by four bits increases the width of flit channels which increases the dynamic power dissipation across the links. However, simulation cases consider routers having at most three faulty ports. So the EX field is not appended to the flit header while evaluating the proposed fault tolerant technique.

5.3.3 Fault patterns with gateway routers

The livelock avoidance mechanism using single additional bit (FLB) in the flit header works well with fault locations that are spatially distributed in the mesh network like the one shown in Figure 5.6 (a). However, certain fault patterns partition the mesh NoC into two or more non-overlapped regions. A gateway router is a router that acts as a single common connection point of such regions in the NoC. A flit from one region can reach its destination in another region only by traversing through the gateway router. Figure 5.6 (b), (c) and (d) depict fault patterns that divide the network into two regions, R1 and R2 which are connected through a gateway router, G (shown as red square). In Figure 5.6 (b), a flit from router (5, 5) which is at two hop distance from its destination router (5, 3) has to be routed through the gateway router (3, 7) due to the fault pattern. In router (4, 7), a flit may roam around in an endless loop (shown as blue line) due to its southward path which does not connect to its destination. Figure 5.6(c) shows how a flit moving south towards destination gets stuck in an

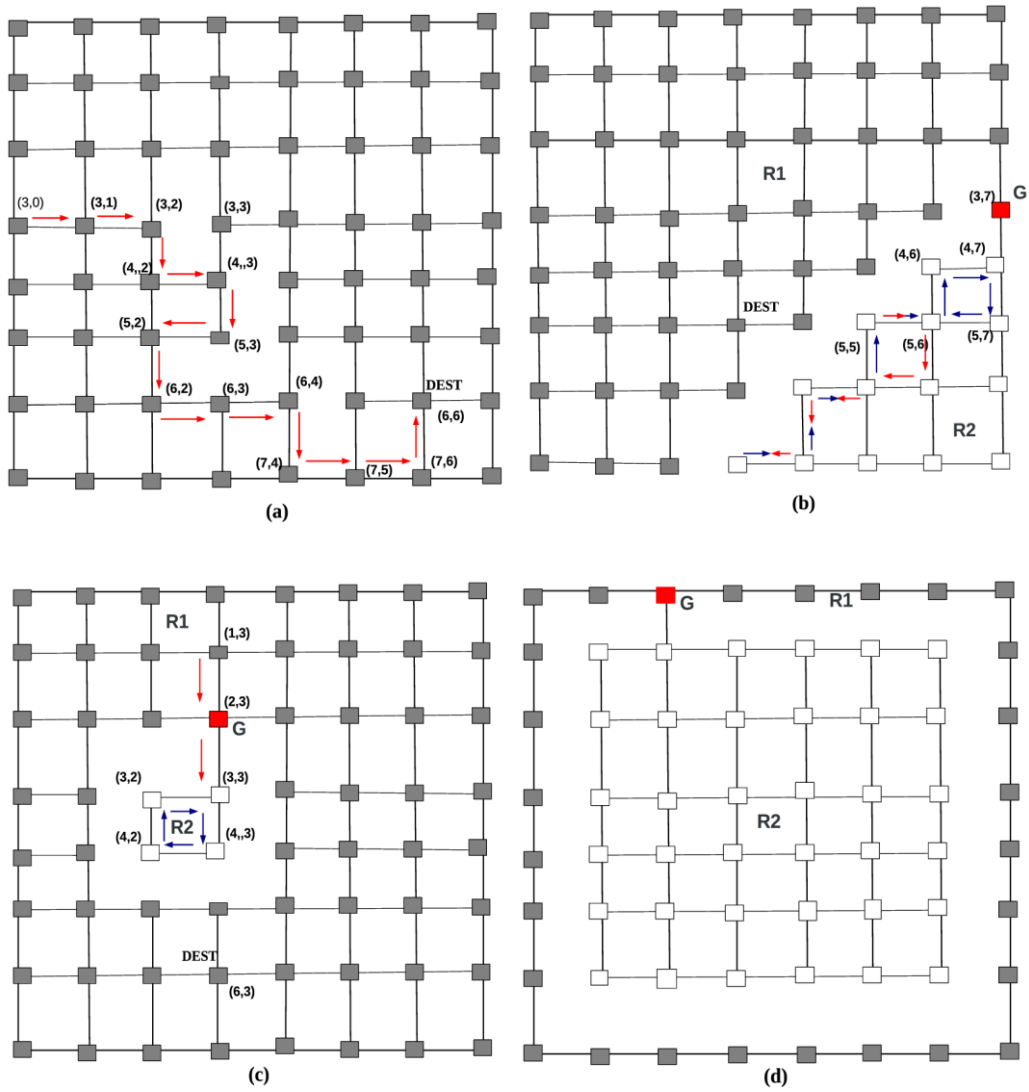


Figure 5.6. (a) Livelock free routing in an 8x8 mesh with multiple faults in partially distributed locations
 (b), (c), (d) fault pattern dividing the network into two regions R1 and R2 connected by a gateway router, G.

infinite loop in region R2 as it cannot trace back through the gateway router to region R1. In Figure 5.6 (d), region R2 is surrounded by region R1 due to the complex fault pattern. To ensure livelock freedom in such cases, additional bits are required in the flit header to trace back from one region to the gateway router so that a flit can reach its destination located in another region. The random fault generator used for this work shows that at 30% fault rate, the probability of formation of fault patterns consisting of a gateway router is only 0.000001. Considering the overhead incurred, the proposed method does not implement livelock safety measures for such fault patterns.

5.4 Experimental evaluation

5.4.1 Experimental methodology

The proposed fault tolerant routing technique is evaluated by comparing various performance parameters with FaFNoC and Maze routing methods explained earlier in this chapter. Experiments are conducted in three phases i.e. simulation, dynamic power analysis and hardware synthesis. The proposed router architecture along with FaFNoC and Maze routing are modelled using a flit level, cycle accurate simulator, Booksim [3][68]. The virtual channel router pipeline of Booksim is modified to accurately model the PDN based deflection routing mechanism and FTLU as mentioned in Section 5.2. All simulations are done for 8x8 mesh NoC. Faults are modelled by disabling a fixed number of bidirectional links in the NoC as mentioned in Section 5.2. Fault rate is the percentage of faulty links out of the total links in the mesh network which is given by Equation 5.1.

$$Fault\ rate = \frac{f * 100}{k} \quad - (5.1)$$

where

f - the number of faulty links

k - the total number of links in the mesh

For an 8x8 mesh NoC, there are a total of 112 bidirectional links and a fault rate of 10% denotes that ports corresponding to 11 links are disabled. In the fault model used here, the disabling of ports does not represent link faults alone. Whenever there is a failure in a datapath element inside a router, it is extrapolated to one of the router's input-output ports. This port is treated as faulty by setting the corresponding fault flag. In order to get a fair comparison, the faulty links are chosen by a random selection process. However, fault patterns which lead to disconnected routers or gateway routers in the network (mentioned in Sections 5.3.2 and 5.3.3) are not considered.

Table 5.1 Various proportions of SPEC CPU 2006 applications in benchmark mixes M1 to M6.

Application	Benchmark Mix					
(%)	M1	M2	M3	M4	M5	M6
Low MPKI	100	0	0	50	0	50
Medium MPKI	0	100	0	0	50	50
High MPKI	0	0	100	50	50	0

The performance of the NoC is evaluated using various synthetic traffic profiles. Traffic is generated by the processor cores at the rate of 0.1 flits/cycle/core for all simulations. In a fault-free NoC, flit injection rate is approximately equal to its generation rate. As the fault rate in the network rises to 30%, there is a proportionate decrease in the number of fault-free channels in the router. As a result, flits must wait in the processor's core buffer for longer time before getting injected into the network through vacant channels. Hence, flit injection rate becomes lesser than the generation rate. A set of 20 simulations are conducted for a particular fault rate by random choosing of faulty channels. Average throughput and average hop count are computed from each set of three distinct fault rates for plotting graphs. Simulations are also conducted using real application traces. Combinations of various low, medium and high MPKI applications from SPEC CPU 2006 benchmark suite [69] are run on a multi-core platform, Multi2Sim[70]. The application mixes (M1 to M6) and the % MPKI in each mix is given in Table 5.1. The network packets generated by these benchmark mixes are injected into the Booksim model and network parameters are analysed.

5.4.2 Analysis of network level parameters

Average throughput, hop count and latency for synthetic traffic patterns

Throughput of a network is the number of flits delivered successfully per cycle per core and has a maximum value of 1. Figure 5.7 shows the average throughput for varying fault rates under uniform, transpose, bit complement and shuffle traffic patterns for the three fault tolerant routing mechanisms. For a fault-free NoC, it is seen that the throughput value is equal to the network

injection rate (which is taken as 0.1 flits/cycle/core). For higher fault rates, the flit injection rate reduces due to congestion in the network. Since deflection routers do not buffer any of the flits in transit, the throughput under various fault rates also decreases in accordance with the injection rate. Figure 5.7 shows that the proposed method meets higher performance for all synthetic traffic patterns compared to the other two. At 30% fault rate, it delivers 38% and 13% higher throughput than FaFNoC and Maze routing respectively for uniform traffic.

Hop count is the total number of productive hops and deflections of a flit from source to destination. The new method efficiently utilizes the FTLU along with an adaptive route computation technique to route flits through the shortest fault-free path. In Maze routing, flits encountering faulty ports are reassigned to ports in random directions. Due to this, a large number of flits traverse through longer paths even when shorter paths exist. This increases the hop count and latency for higher fault rates. Even though FaFNoC switches between XY and YX routing to avoid congestion, the lack of adequate structural connectivity at the PDN results in increased deflections and latency. Hence average hop count is found to be the highest for FaFNoC compared to Maze routing and the proposed method as shown in Figure 5.8. It also shows lowest throughput values for all fault rates. At uniform traffic conditions and 30% fault rate, hop count for the proposed method is lesser by 9% compared to Maze routing and 16% compared to FaFNoC.

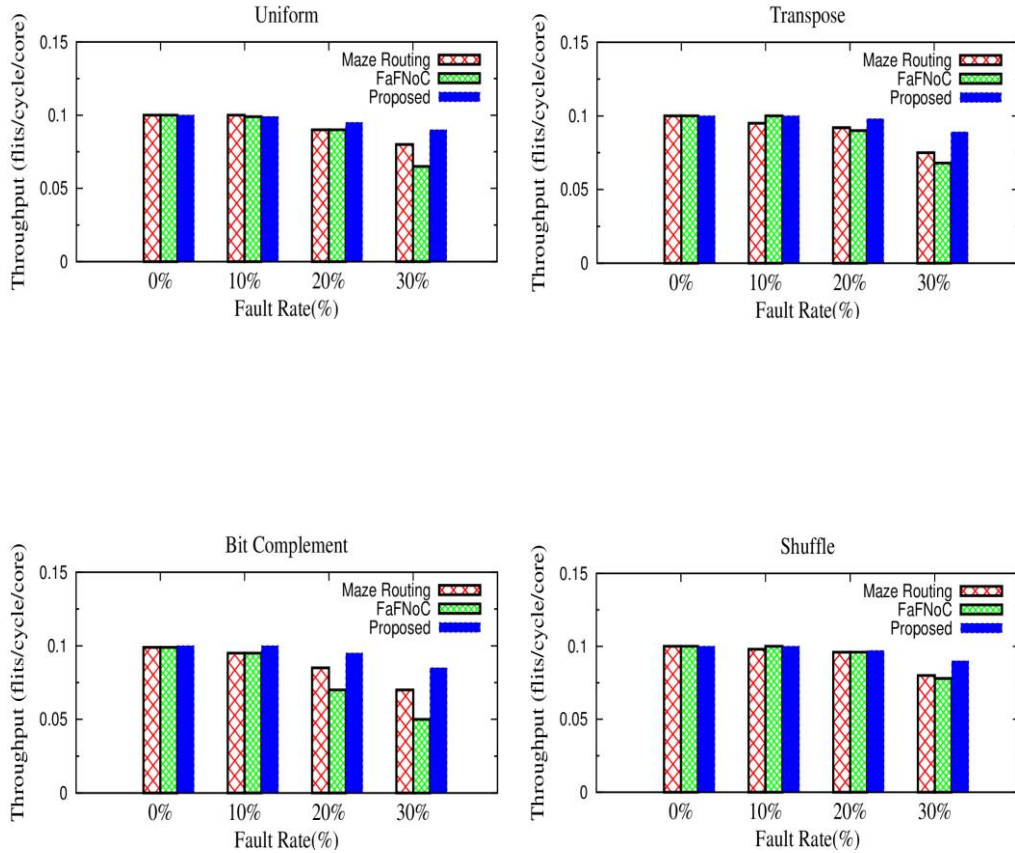


Figure 5.7 Average throughput Vs. Fault rate under various synthetic traffic patterns for the proposed work, FaFNoC and Maze routing in 8×8 mesh NoC.

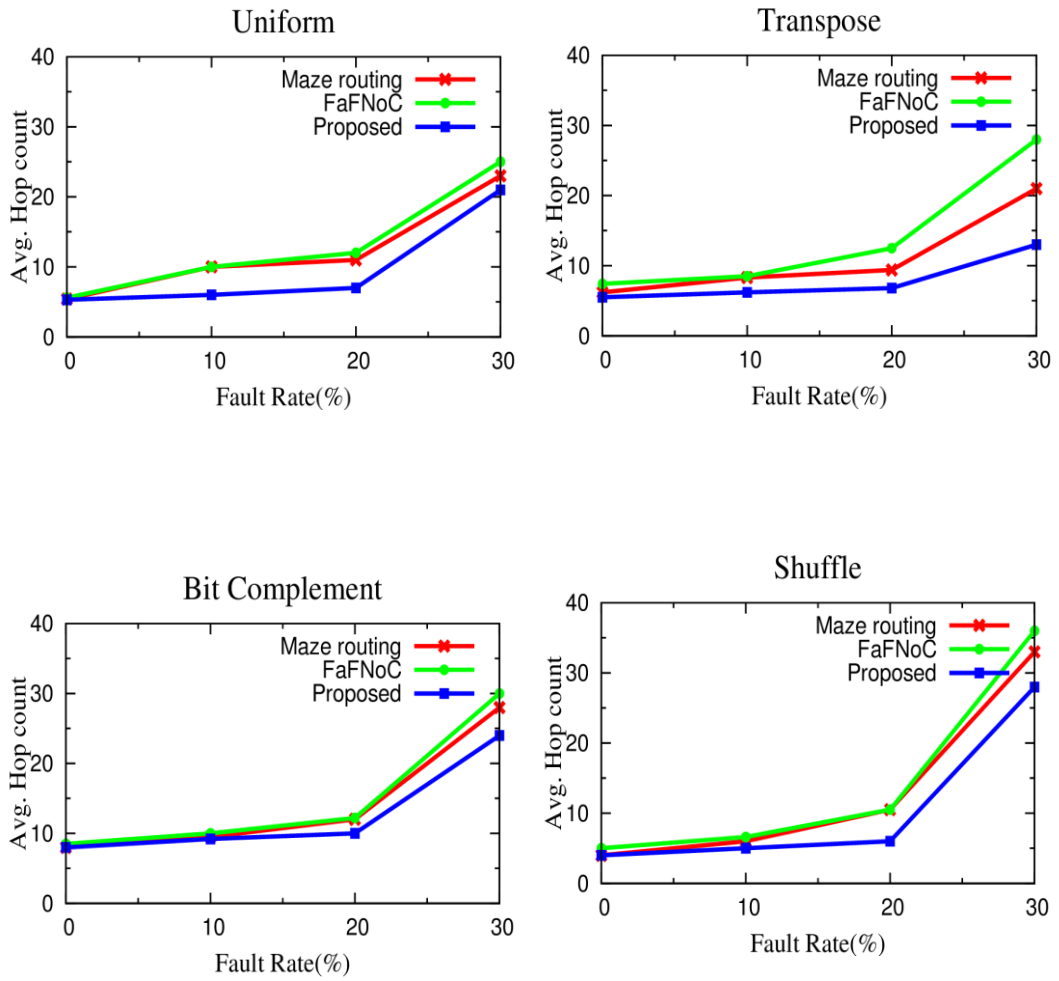


Figure 5.8 Average hop count Vs. Fault rate under various synthetic traffic patterns for the proposed work, FaFNoC and Maze routing in 8×8 mesh NoC.

Latency of a flit is the total number of cycles it takes to traverse from its source to destination. In Figure 5.9, average flit latency for an 8x8 mesh NoC is shown as a function of flit injection rate for uniform traffic for various fault rates. For all the three methods, it is observed that the network tends to saturate early with increasing fault rate. The proposed method shows 3.8% and 8.7% improvement in network saturation point with respect to Maze routing for 10% and 30% fault rates respectively. At 30% fault rates and injection rate of 0.1, Maze routing and FaFNoC show a sharp rise in average latency and both networks saturate. From the graph, it is observed that the proposed routing technique is capable of tolerating higher fault rates with a gradual increase in latency for flit injection rates above 0.1.

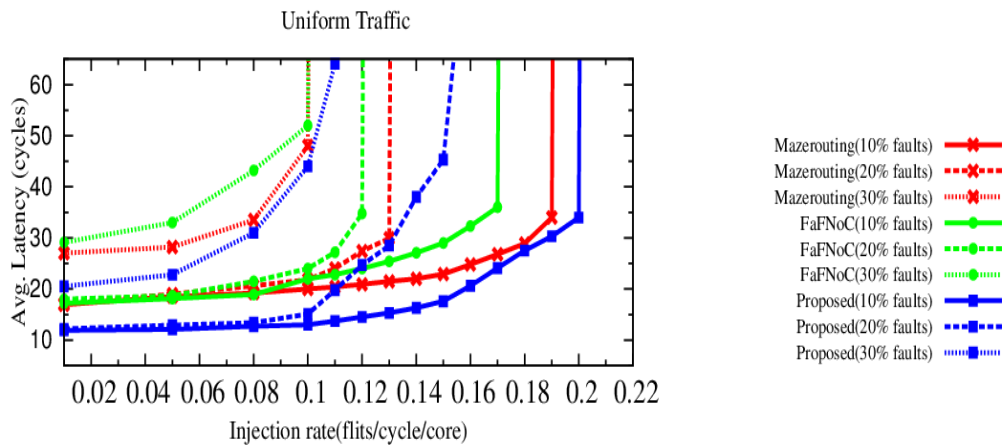


Figure 5.9 Average latency Vs Injection rate (uniform traffic) for various fault rates for the proposed method, FaFNoC and Maze routing in 8×8 mesh NoC.

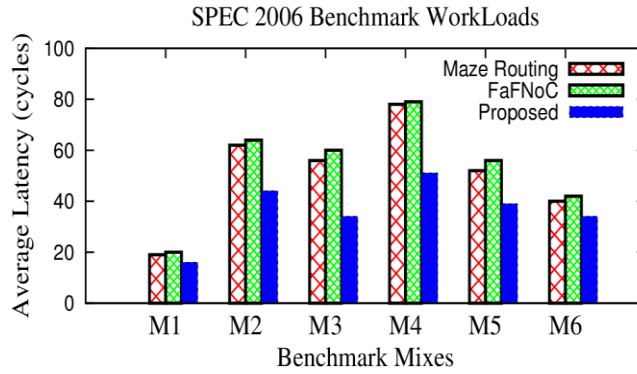
Average latency and deflection rate for real applications

Figure 5.10 (a) and (b) show the graphs of average latency and deflections obtained for benchmark application mixes (M1 to M6) for a fault rate of 10% on flit channels. For real applications, the injection rates are much lower than that chosen for synthetic traffic simulation. For a faulty NoC, at very low injection rates, there is lesser congestion in the network and more than one productive ports may be eligible for reallocation. The merit of the proposed algorithm lies in its capability to reallocate ports that result in shortest possible distance to the flit's destination. An average of 37% and 33% lower latency is obtained for the proposed method across all mixes compared to FaFNoC and Maze routing respectively. Using the new technique, majority of the flit hops are productive, hence deflection rates are considerably lower for all benchmark mixes as seen from the figure.

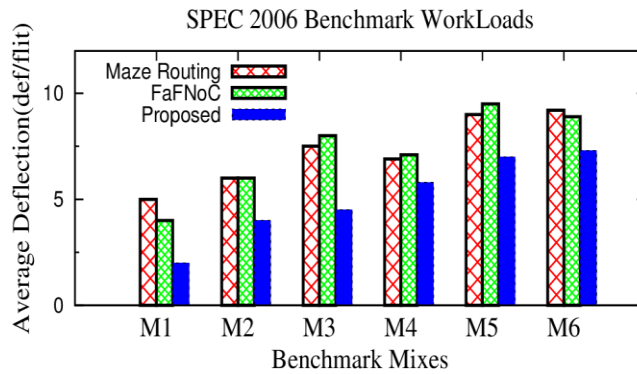
5.4.3 Dynamic link power estimation

Increase in the flit deflection rate leads to increased activity and dynamic power dissipation across the inter-router links. Hence power efficiency of such NoCs is proportional to the Link Activity Factor (LAF) which is derived from the average hop count of flits through these channels during simulations. LAF is a measure of the number of packets per cycle per link and is calculated using Equation 3.1.

Orion 3.0 tool [71] is used to estimate the dynamic power dissipated at the inter-router links. For all the three fault tolerant routing techniques, the



(a)



(b)

Figure 5.10 (a) Average Latency and (b) Deflection for 10% fault rate under SPEC CPU 2006 benchmark applications mixes for the proposed method, FaFNoC and Maze routing in 8×8 mesh NoC.

values of LAF corresponding to various hop count values at various fault rates are computed using the above formula. The number of active links in the network decreases with increasing fault rate. Since LAF is calculated as an average value across all links in the mesh, the total number of links is taken as k (irrespective of the number of active links). The calculated value of LAF is given as the input load rate in Orion to determine the dynamic power. Link length of 2.5 micrometers and a baseline flit width of 128 bits is assumed. For 8x8 network, FaFNoC and Maze routing extend the flit header by 12 and 14 bits respectively whereas the proposed method uses only 1 additional bit for FLB. In Figure 5.11, the link power (in mWatts) obtained from Orion is plotted against varying fault rates for four synthetic traffic functions. From the graph of uniform traffic function, dynamic power due to the proposed method is 20% lower than FaFNoC at 30% fault rate. FaFNoC consumes highest energy among the three methods. The reason for the energy efficiency of the new technique is the reduction in hop count by efficient fault tolerant routing.

5.4.4 Hardware synthesis

Verilog models of the three router architecture are implemented and synthesized using Synopsys design compiler with 65nm CMOS library and the values of router pipeline latency, static power, and area are obtained for the three architectures. Router delay is the time taken by a flit to move from its input to output port through various functional units. This can be divided into two stages.

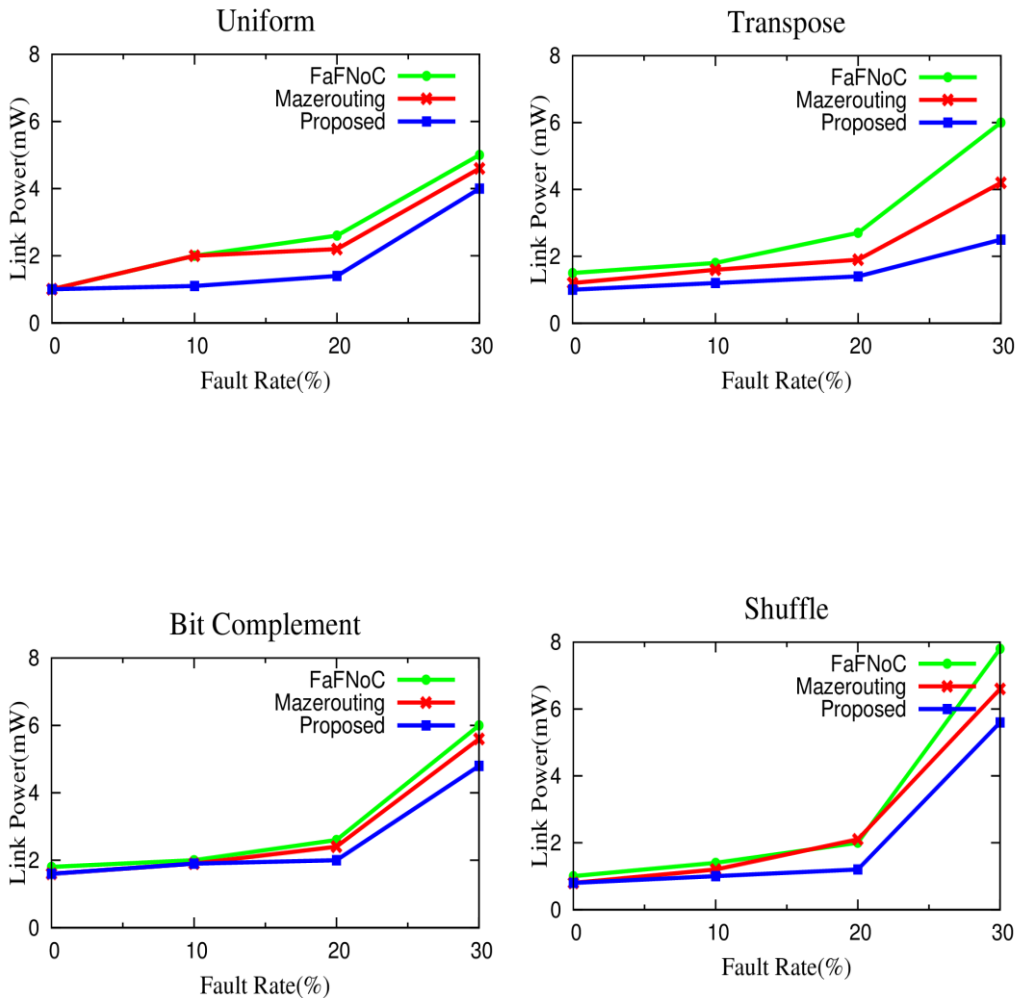


Figure 5.11 Link power Vs Fault rate under various synthetic traffic patterns for the proposed method, FaFNoC and Maze routing in 8×8 mesh NoC.

Table 5.2 Router delay, static power and area for Maze router, FaFNoC and proposed method normalised w.r.t CHIPPER.

	CHIPPER	Maze routing	FaFNoC	Proposed
Pipeline delay	1	1.08	1.15	1.26
Static power	1	1.2	1.26	1.35
Router area	1	1.24	1.3	1.4

The first stage of the three routers has similar functional units, hence all of them have the same delay for the first stage. The second stage consists of the PDN and fault tolerance mechanisms. The critical path length inside each router and operating frequency of the network are determined by the complexity of the fault tolerant logic. The output stage of CHIPPER does not include any fault tolerant logic, so its delay is taken as 1. The FTLU in the proposed router increases the delay of second stage by 26% when compared with CHIPPER. Area and static power consumed by the control logic of the proposed router are 40% and 35% higher than CHIPPER. Maze routing incurs the least area and power among the three architectures as it uses fewer functional units for fault tolerance.

Even though NoCs using the newly proposed router operate at a lesser frequency compared to the other two methods, significant reduction in dynamic power dissipation at the inter-router links is achieved by the novel energy efficient fault tolerant routing mechanism. In an 8x8 mesh NoC, both FaFNoC and Maze routing require 8%-10% wider channels to accommodate the extended

flit headers [58], [59]. The channel width increases with growing network size, resulting in larger area for the router's datapath and inter-router links. The proposed router is advantageous in this aspect, since the flit header uses only one additional bit for FLB irrespective of the network size.

5.5 Limitations of the coarse grain fault model

The conventional fault model used in Section 5.1 has coarse-grain features and is widely used in deflection routers since it ensures that routers have equal number of input and output ports. In this model, a fault in either an input or output link of a router is represented by disabling both input and output links in that direction as shown in Figure 5.3. This essentially means that the input-output ports of the routers connected to the two ends of the link are not allowed to send or receive flits. A break in one of the link wires results in disabling of flit movement through the pair of bidirectional links even when one of them is fault-free. In Figure 5.3, faults exist only in four unidirectional links while router ports corresponding to eight unidirectional links are disabled. The deactivation of the healthy links negatively affects the network performance by a small amount under low fault rates. But when fault rates are significantly high (greater than 30%), these fault-free links can be utilized by flits to reach their destinations in lesser time. Motivated from the limitations of the earlier router model, an enhanced fault tolerant routing model is proposed. The enhanced model helps to improve the path diversity in a faulty NoC that uses deflection routing technique. The new method helps to reduce average flit latency and hop count and improves the energy efficiency of the NoC significantly for large number of link failures.

5.6 Enhanced fault tolerant router model

In the proposed enhanced router model, only the faulty unidirectional links of the bidirectional pairs are disabled while keeping the healthy links active so that they may be utilized for transmitting flits through shorter paths to destination. In Figure 5.12, faults are represented using the enhanced model in a 4x4 mesh NoC. The east input line of R5 connected to the west output line of R6 is faulty. In the proposed model, the east input port of R5 and west output port of R6 alone are disabled. The parallel link connecting east output of R5 to west input of R6 is healthy; hence it is active. Similarly, there are three more unidirectional faulty links which are disabled. Router datapath is considered as an extension of one of its output links. As done for the previous model, faults in the router's datapath elements are propagated to one of the output ports which is then disabled. Errors in control circuit elements like output port allocator and route computation unit may affect the overall routing operation. Since it is difficult to diagnose and isolate functional units with control errors, such a router itself is deactivated by disabling all its input and output ports. As in the coarse grain model, four fault flags (1 bit each) are used to represent the fault status of the north, south, east and west output ports of a router. For a fault-free router, values of the four fault flags will be zero. In case of a link failure or datapath error, the corresponding output flag of the router is set to a value 1. In Figure 5.12, due to the link fault from north output port of R10 to south input port of R6, the NF of R10 is set to 1. The status of the fault flags are taken into account during output port allocation of flits. Since the fault status of an input port is equal to the fault status of the neighbouring router's output port connected to it, separate fault flags are not used for input ports. The values of fault flags are assigned during fault diagnosis phase.

Apart from the fault flags, each router also maintains four block flags, North Block (NB), South Block (SB), East Block (EB) and West Block (WB) in the four directions respectively. Each block flag is 1 bit long and has a default value of 0. Block flags are valid only for the active output ports, they are ignored for the faulty ports. During router operation, the block flags of some of the active output ports are set to a value 1 for a certain number of cycles to temporarily disallow the movement of flits through them. This is explained in detail in the following section.

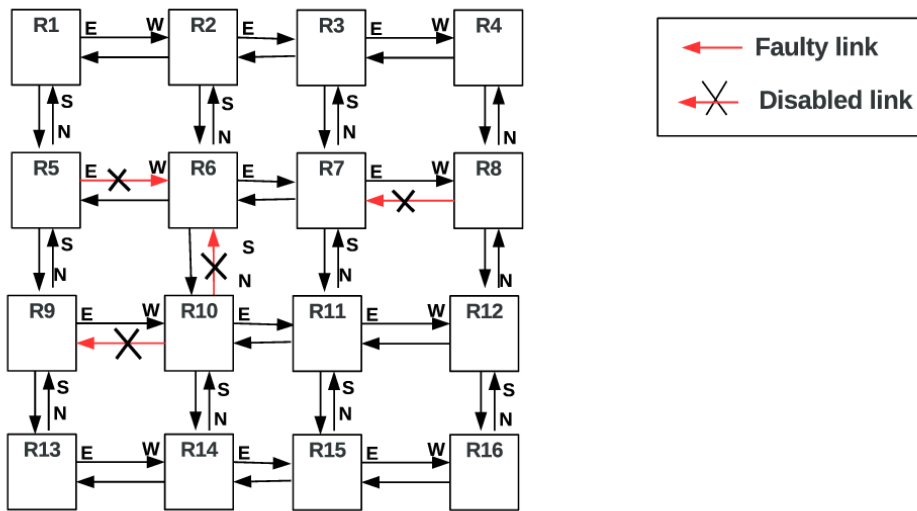


Figure 5.12 Representation of a 4x4 mesh NoC using enhanced fault tolerant router model.

In the enhanced fault tolerant router model, since faulty unidirectional links are disabled, the number of active input and output ports may be unequal.

This violates the basic condition for deflection routing that the number of input and output ports should be equal. Three conditions that may occur are explained below.

- **Case 1: Number of active input ports < Number of active output ports**

In Figure 5.12, router R6 has 4 active output ports and 2 active input ports. In such routers, a maximum of two flits enter into the router pipeline through the active inputs. Since there are more output ports, each flit can be assigned to an output port based on its routing choice and priority.

- **Case 2: Equal number of active input and output ports**

This condition is equivalent to case 1. Since there are sufficient numbers of output ports, all the flits entering through the input ports can hop out of the router at the end of the router pipeline.

- **Case 3: Number of active input ports > Number of active output ports**

In such a situation, sufficient numbers of output ports are not available to accommodate all the input flits. Hence, side buffers are provided inside each router to store the excess flits for a few cycles. In Figure 5.12, router R10 has 4 active input ports whereas it has only 2 active output ports. In certain cycles, flits arrive at all the four active input ports. After output port allocation, flits that are assigned to the faulty output ports i.e. west and north (NF=1, WF=1) are moved

to the side buffer. These flits re-enter into the router pipeline through vacant input lines in a subsequent clock cycle.

5.6.1 Router architecture for the enhanced model

The enhanced fault tolerant router architecture is based on the MinBD router depicted in Figure 2.3 [8]. The routing algorithm computes productive output ports of a flit using XY algorithm. Flit prioritization and livelock avoidance is based on the golden token scheme used in MinBD. The PDN structure used in the previous chapters is used for output port allocation of flits.

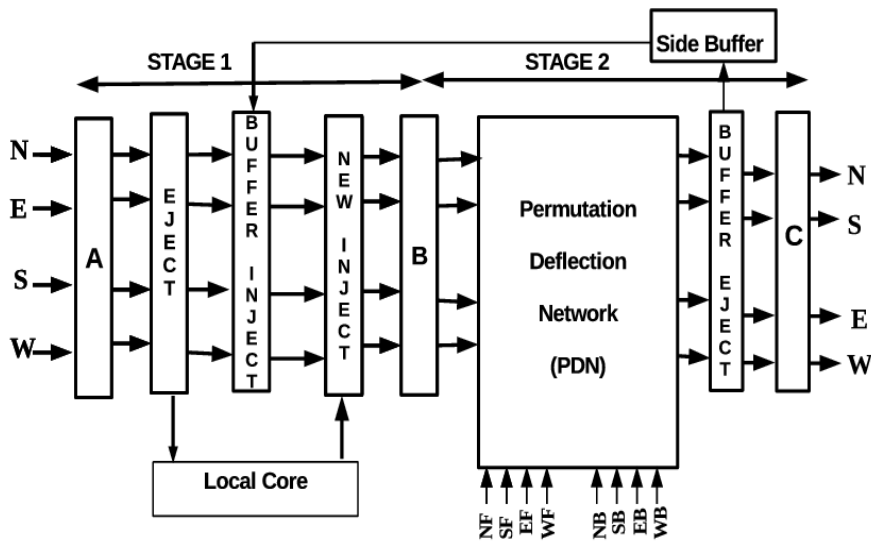


Figure 5.13 Two stage pipeline architecture of enhanced fault tolerant router.

The modified two stage pipeline architecture of the enhanced fault tolerant router is shown in Figure 5.13. Each of the four permeters of the PDN (as in Figure 2.3) map the two input flits to one of the two output lines based on four parameters :

- Computed output port of a flit
- Flit prioritization
- Status of the fault flags (NF, SF,EF, WF)
- Status of block flags (NB, SB, EB, WB)

Among the two input flits in a permuter, the flit with higher priority is assigned to a productive output which is not faulty or blocked. The other flit is assigned to the remaining output. Each of the four permuter blocks function according to Algorithm III. In a router where some of the output ports are faulty, if four flits arbitrate at the PDN, some of them should be necessarily assigned to faulty outputs. Such flits are transferred into a side buffer by the buffer eject unit as per Algorithm IV. Flits that are stored in the side buffer in previous cycles are assigned to vacant input lines in the router pipeline based on first-in-first-out rule.

5.6.2. Side buffer parameters

The size of side buffers should be such that there is enough space to accommodate eligible flits in each cycle. In order to avoid starvation of side buffered flits, they should be injected back into the router's data path. The Blocking Time Interval(BTI) is defined as the number of clock cycles for which a non-faulty output port is temporarily blocked from sending flits. The block flag of an active output port is repeatedly set for BTI cycles and reset in the next

cycle. The value of BTI is constant throughout the NoC. A buffered flit can re-enter into the router pipeline only if an input line is vacant. This vacancy is generated by blocking input flits from neighboring routers by setting their block flags at regular intervals of BTI. This mechanism is explained further using an example from Figure 5.12. Router R10 has four active input ports but only two active output ports (north and west ports are faulty). It is assumed that four flits arrive at the input ports of R10 in a given cycle. After port allocation, two of the flits assigned to the faulty output ports are transferred to the side buffer. In order to inject these flits into the router pipeline, the neighbouring routers, R9 and R6 connected to R10 through the east and south links are blocked from sending flits to R10. This is done by setting the block flags, EB of R9=1 and SB of R6=1 for BTI cycles. During this time, the side buffered flits in R10 can be assigned to the vacant north and west input lines for output port allocation by the PDN. Since the side buffered flits are given higher priority than input flits, they are likely to win fault-free output ports during arbitration in the PDN. Only healthy bidirectional links of a router are active during the BTI. The block flags are reset for one cycle after BTI, during which fault-free unidirectional links are allowed to transmit flits. This enhances the performance of the network at high fault rates.

In order to obtain an optimum size of the side buffer and BTI which can deliver a reasonable trade-off between performance, power and area, simulations are conducted on 8x8 mesh with various synthetic traffic patterns and the following observations are made. In routers with two faulty output links, the probability that flits arrive at each of the four active input ports in two consecutive cycles is less than 15%. In three consecutive cycles, this probability is less than 2%. Accordingly, the side buffer size is chosen as 4 and BTI as 3

cycles. Each router can have a maximum of 4 flits in the side buffer and they are re-entered into the router pipeline within a maximum period of 3 clock cycles after buffering. The flit movement along the major functional blocks of the router R10 in our previous example is shown in Table 5.2. At the end of second cycle, two of the four input flits occupy the side buffer. For the next 3 cycles, input flits through the west and north input lines are blocked. Hence, the buffered flits move into the two vacant input lines in the third cycle. In the next cycle, three flits arbitrate for two fault-free outputs in the PDN, the third flit coming from the neighbouring east or south input port. Two of these flits are assigned to output ports and the third one is buffered. Again, in the next cycle, the buffered flit moves to the PDN along with new flits from neighbouring routers. BTI ends after three cycles and the block flags (SB of R6 and EB of R9) are unblocked.

Table 5.3 Example of flit movement through major functional blocks of router R10(Figure 5.12) in 5 consecutive cycles with side buffer size = 4 and BTI = 3 cycles.

Cycle No	Input flits N,S,E,W	Buffer Inject	PDN flits N,S,E,W	Output flits N,S,E,W	Buffer Eject
1	1, 1, 1, 1	0	0, 0, 0, 0	*, 0, 0, *	0
2	*, 1, 1, *	0	1, 1, 1, 1	*, 1, 1, *	2
3	*, 0, 1, *	2	0, 1, 1, 0	*, 1, 1, *	0
4	*, 1, 0, *	0	1, 0, 1, 1	*, 1, 1, *	1
5	1, 0, 1, 1	1	0, 1, 0, 0	*, 1, 0, *	0

Algorithm III: Function of a Permuter Block of PDN

```

1: Inputs:  in_flit1, in_flit2, fault_flags (NoF, SoF, EoF, WoF), block_flags (NB, SB, EB, WB);
2: Outputs : output1, output2 ;

3:   If Priority(in_flit1)> Priority(in_flit2)
4:   {   highPr_flit = in_flit1;   lowPr_flit= in_flit2; }
5:   Else
6:   {   highPr_flit = in_flit2;   lowPr_flit= in_flit1;}
7:   If route(highPr_flit) == output1
8:   {
9:       If (output1.block_flag==0) and (output1.fault_flag==0)
10:      output1 = highPr_flit; output2 = lowPr_flit;
11:      Else output2 = highPr_flit; output1 = lowPr_flit;
12:  }
13:  Else If route(highPr_flit) == output2
14:  {
15:      If (output2.block_flag==0) and (output2.fault_flag==0)
16:      output2 = highPr_flit; output1 = lowPr_flit;
17:      Else output1 = highPr_flit; output2 = lowPr_flit;
18:  }

```

Algorithm IV: Flit Buffering by Buffer Eject Block

```

1: Inputs: north_flit, south_flit, east_flit, west_flit,
          fault_flags (NoF, SoF, EoF, WoF),
          block_flags (NB, SB, EB, WB);

2: If (north.fault_flag) or (north.block_flag)
3:   { side_buffer.push(north_flit) }

4: If (south.fault_flag) or (south.block_flag)
5:   { side_buffer.push(south_flit) }

6: If (east.fault_flag) or (east.block_flag)
7:   { side_buffer.push(east_flit) }

8: If (west.fault_flag) or (west.block_flag)
9:   { side_buffer.push(west_flit) }

```

5.6.3 Simulation methodology

The virtual channel router architecture of the open source simulator, Booksim [68] is modified to model the minimally buffered deflection router with two cycle latency. Fault flags and block flags are appended as in Figure 5.13 to activate the proposed enhanced fault tolerant model explained in Section 5.6. Simulations are conducted using single flit packets. The width of the flit channel is taken as 140 bits with 128-bit data field and 12-bit header field. The fault rate for a mesh topology is given by Equation 5.1 in Section 5.5. For an 8x8 mesh NoC with 10% fault rate, 11 out of the 112 links are faulty. For each simulation, the faulty links are chosen at random. Each simulation is run for 1 million cycles. Flit injection rate is fixed as 0.1 flits/cycle/node for all simulations. Network parameters like average flit latency, hop count, deflection rate and throughput are calculated by taking average readings of 10 simulations conducted for a fixed fault rate. The graphs are plotted by repeating the simulations for various fault rates (from 0 to 50%) with typical synthetic traffic profiles like uniform, bit complement transpose and neighbour. The performance of the proposed method is also analysed using workloads from SPEC CPU 2006 benchmark application suite given in Table 5.1.

The network performance parameters of the enhanced model are analysed by comparison with two other routing methods which use the MinBD router architecture and the coarse grained faulty router model discussed in Section 5.1. The first method used for comparison differs from the proposed method in the aspect of router model alone (denoted as conventional model in the graphs). The second one is the Maze router [59], which differs in the router model as well as routing algorithm (denoted as Maze routing in the graphs). Due

to the enhancement in the availability of active links, the proposed model exhibits better path diversity compared to the conventional model and Maze routing for the same fault rate. The effects of the three models on the network delay and dynamic power dissipation of flits through the network are analysed.

5.6.4 Analysis of simulation results

Average latency, hop count, throughput, deflection rate

Average flit latency is directly proportional to the average hop count, which is the average number of hops taken by a flit during its traversal through the network to its destination. Since the graphs of average latency and hop count follow the same trend, the graphs for average hop count alone for typical synthetic traffic patterns are shown in Figure 5.14. The proposed model has twice the number of active links compared to the conventional model and Maze routing. These links provide alternate paths to flits whose productive ports are faulty, resulting in reduction of average hop count of flits. At fault rates less than 20%, there is no significant difference in the hop count and latency obtained for the three methods. For 30% fault rate, 68 pairs of input-output ports (34 bidirectional links) are disabled for the conventional model and maze router. In the proposed model, half of the input- output ports among these are active once after every BTI cycles. Hence for uniform traffic simulations and 30% fault rate, the average hop count for the proposed model is 23.5% lesser than conventional model and 46% lesser than Maze routing. Lesser hop count implies that average flit latency is lesser using the proposed model, which accounts for application speed up. At fault rates above 35%, two or more links in each router tend to be faulty. Since input-output ports in the faulty directions are fully disabled in the

conventional model and Maze routing, the flits suffer extremely high latency. The enhanced model exhibits a graceful degradation in performance even at higher fault rates, which is clear from Figure 5.14.

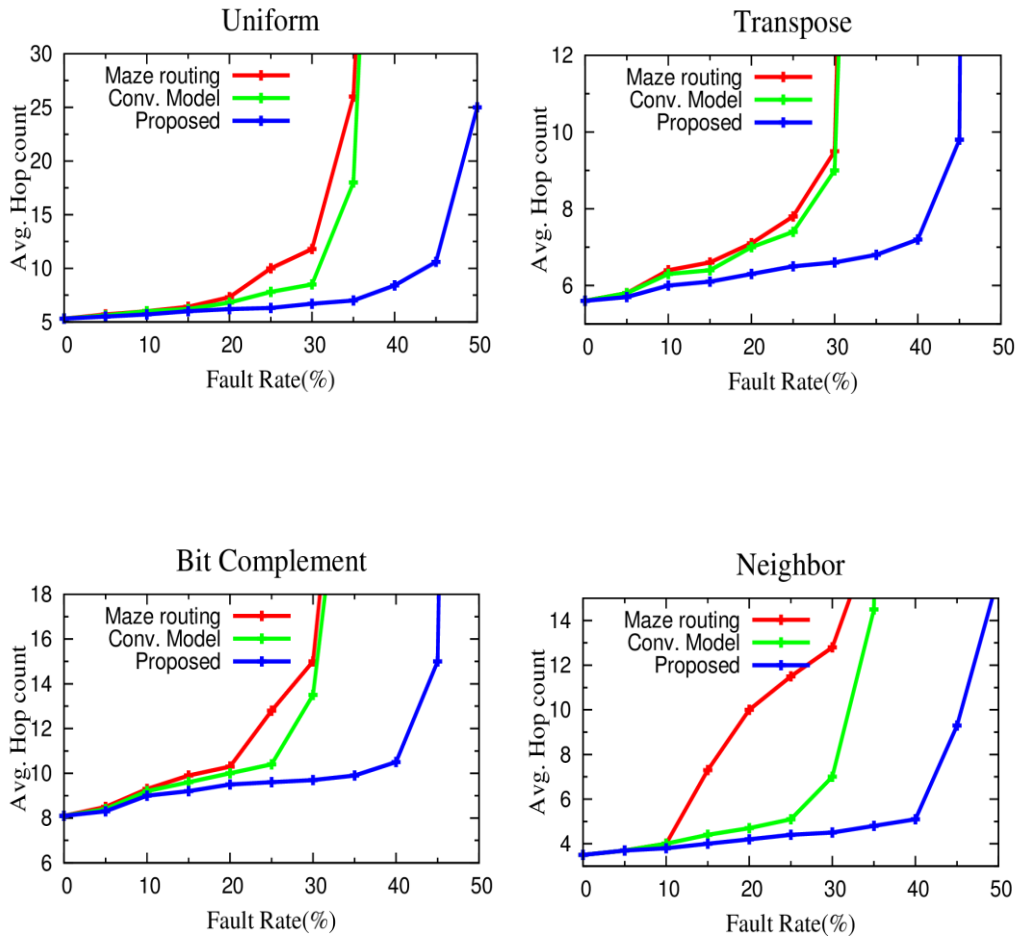


Figure 5.14 Average hop count Vs Fault rate under synthetic traffic patterns for the proposed model, conventional model and Maze routing in 8x8 mesh NoC.

Figure 5.15 shows the average flit deflection rate at various fault rates under uniform and transpose traffic patterns. In a network with faulty links, a flit may be deflected to a non-productive output port due to two reasons:

- (A) productive output port is assigned to a flit with higher priority
- (B) productive output port is faulty.

In Figure 5.15, the deflection rate in the proposed method due to (A) and (B) are compared with that of the conventional model for various fault rates. It is observed for all fault rates that deflection rates are higher for the conventional model compared to the enhanced model. With higher fault rates ($> 20\%$), deflections due to (A) and (B) increases largely for conventional model whereas for the proposed model, deflections due to (A) is almost constant and those due to (B) increases steadily.

Network scaling

Figure 5.16 shows the average flit latency graphs for various network sizes. For all the three methods, it is noted that higher network sizes are capable of withstanding higher fault rates. For 4x4 mesh, 15% to 20% of the faults are tolerated gracefully and the proposed model offers least average latency compared to the other two methods. For 8x8, 10x10 and 12x12 networks, the proposed method tolerates 35%-40% faults, whereas conventional model begins to saturate at fault rates higher than 25% due to limited number of active links. Due to the inefficiency of the algorithm in routing flits through minimal paths, Maze routing exhibits higher latency compared to the other two for all fault rates.

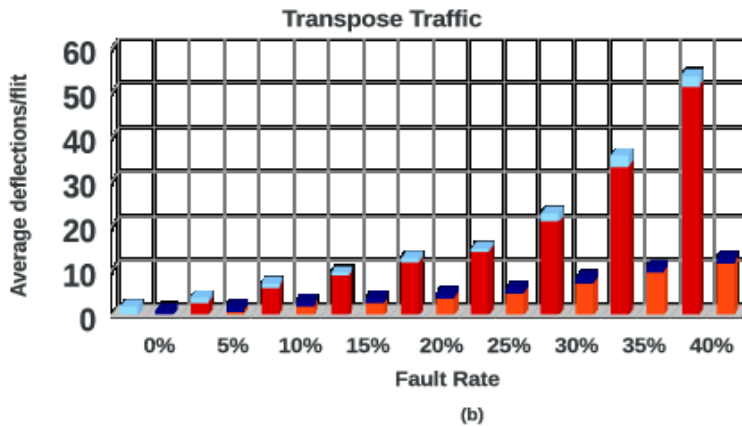
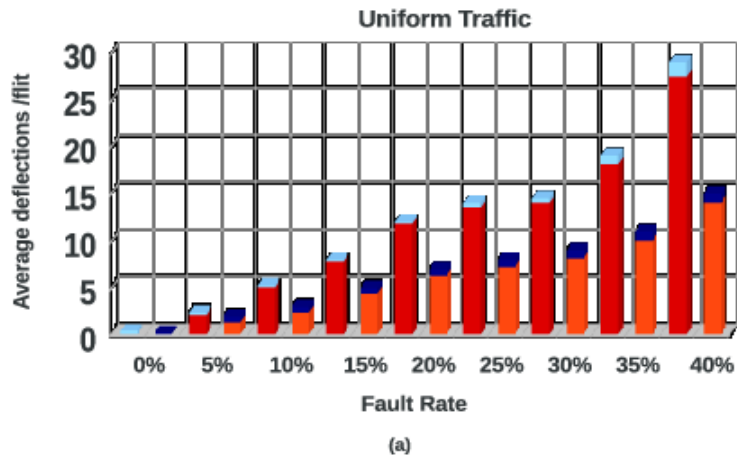


Figure 5.15 Average number of deflections per flit Vs Fault rate under
 (a) Uniform (b) Transpose traffic patterns for the proposed
 model and conventional model in 8×8 mesh NoC.

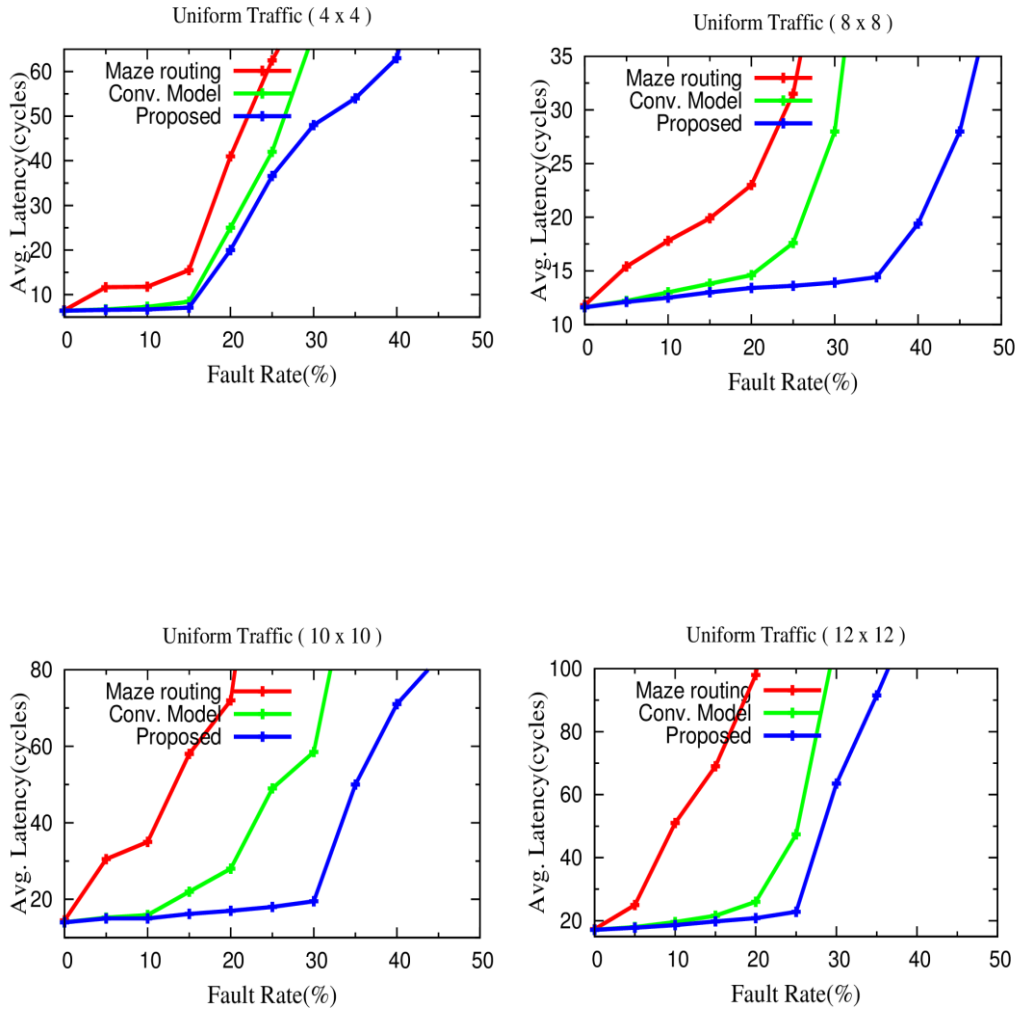


Figure 5.16 Average flit latency Vs. Fault rate for uniform traffic pattern in 4x4, 8x8, 10x10 and 12x12 network sizes.

Dynamic power dissipation

As the flits traverse through the links to reach their destinations, the link activity increases which leads to dynamic power dissipation. In an 8x8 mesh NoC, the LAF is calculated using Equation 3.1 and given as the variable input to Orion tool for computing the dynamic link power. The other parameters are 1V Vdd, 65 nm technology and 1mm link length. The link width for conventional and proposed models is 140 bits. Maze routing requires 152 bits for the flit header where 12 additional bits are for the extended flit header. Link power at various fault rates for the typical synthetic traffic profiles is given in Figure 5.17. Low average hop count for the proposed model at all fault rates gives lower value of LAF, hence dynamic power dissipation is also lower compared to the other two methods.

Real applications

The application level performance of the proposed method is compared with the other two methods using workload mixes M1 to M6 chosen from Table 5.1. Figure 5.18 shows the average latency graphs at 10% and 30% fault rates respectively; Figure 5.19 shows the average deflections per flit at the same fault rates for these workloads. With increase in fault rate, the values of flit latency in Maze routing and conventional model are similar, while average latency is 26% lesser and average flit deflection is 21% lesser for high MPKI workloads (like M3 and M5) using the proposed method.

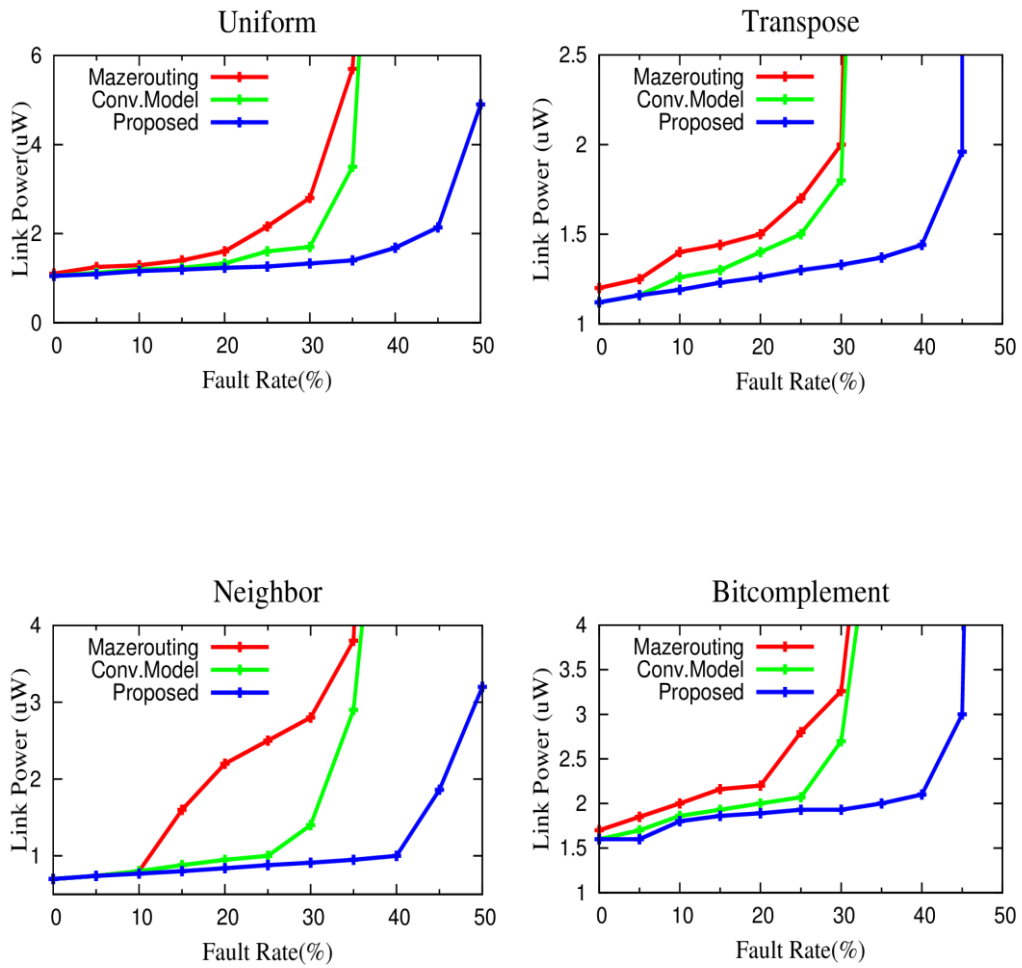
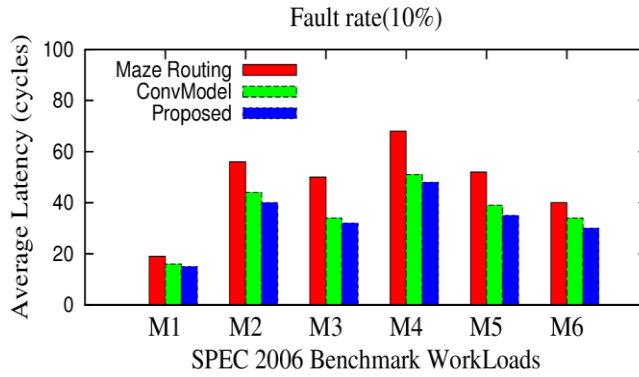
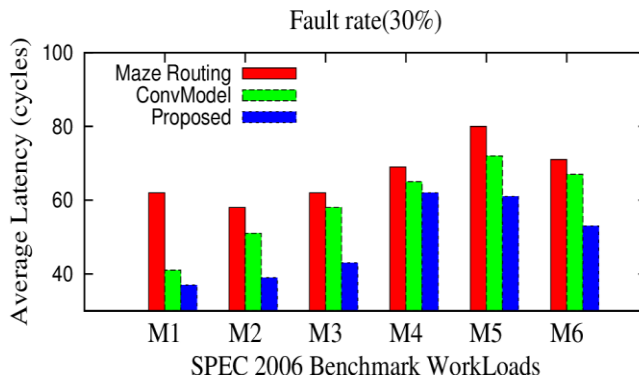


Figure 5.17 Dynamic Link Power (microwatts) Vs. Fault rate under various synthetic traffic patterns for the proposed model, conventional model and Maze routing in 8×8 mesh NoC.

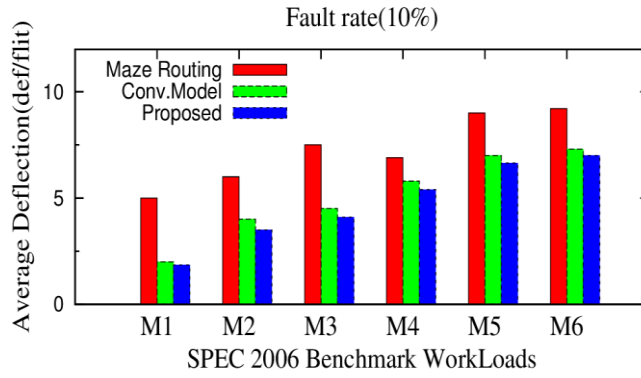


(a)

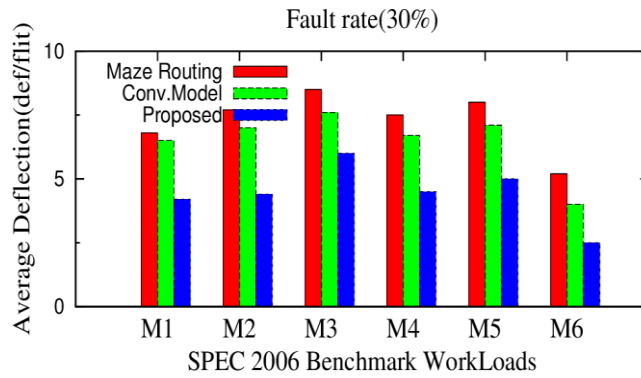


(b)

Figure 5.18 Average latency under various SPEC Benchmark application mixes for the proposed model, conventional model and Maze routing in an 8×8 mesh NoC with (a) 10% and (b) 30% fault rates.



(a)



(b)

Figure 5.19 Average deflection rate of flits for SPEC Benchmark application mixes for the proposed model, conventional model and Maze routing in an 8×8 mesh NoC with (a) 10% and (b) 30% fault rates.

5.7 Chapter summary

In this chapter, a unique fault tolerance technique to improve the performance and energy efficiency of deflection router based NoCs at high fault rates is proposed. This technique consists of a fault tolerant logic unit at the output stage of a deflection router which reallocates flits from faulty ports to healthy output ports. Adaptive switching between XY and YX routing algorithm is adopted to avoid livelock and route flits to destination through the shortest available path. This technique incurs minimum wiring overheads and promises performance gain even when fault rate is high. A novel fault tolerant router model which utilises discriminate disabling of faulty unidirectional links is also discussed. The enhanced model helps to improve the path diversity in a faulty mesh NoC so that flits can be routed through short paths when default routes fail.

Chapter 6

Thermal Aware Deflection Routing

Abstract

Thermal distribution due to the network traffic in a buffer-less deflection router based NoC is analysed in this chapter. The study reveals that thermal activity is not uniform; it is concentrated at the centre of the chip. An adaptive deflection routing technique is proposed in this chapter to obtain an even thermal profile across the mesh network. In order to reduce the load through the central links of the network, deflected flits in each router are reallocated to vacant output ports that are directed towards the edges of the chip. From evaluations, it is seen that the proposed method achieves significant reduction in thermal variance compared to deflection routers without thermal awareness.

In a buffer-less deflection router, flits with higher priority occupy output ports in productive directions during port allocation phase. The remaining flits are randomly assigned to vacant output ports. Deflection is the undesirable traversal of a flit in a direction away from its destination. The architecture of output port allocator inside the router and the routing algorithm together determine the direction of deflection of these flits. In all the methods proposed in this thesis, output port allocation is performed using a PDN structure. The structural connection and port allocation mechanism of the PDN are such that majority of the traffic flows through the central links and routers of the mesh network and very less traffic through the edges. This leads to concentration of dynamic activity and consequent power dissipation at the chip centre, which causes non-uniform usage of network resources and uneven heating up. Formation of thermal hot spots at the centre and cool spots at the edges tends to reduce the average life-time of the chip. In order to warrant long term reliability of the chip, it is desirable to have a uniformly distributed thermal profile within the chip which can be achieved through thermal aware routing.

CHIPPER is a buffer-less deflection router that uses a PDN for parallel allocation of output ports to flits [10]. It adopts a deterministic dimension order routing (DOR) algorithm for route computation. In order to study the traffic distribution within the network using CHIPPER, simulations are conducted on a NoC of size 8x8, connected in a mesh topology. For a simulation period of 50,000 cycles using uniform traffic pattern at pre-saturation injection rate, it is observed that 22% of the total flit deflections are directed towards centrally located routers. At the same time, output ports directed towards the edges of the mesh are found to bear lesser load. The limitations of the router architecture

mainly accounts for the increase in deflections towards the centre of the chip. Unproductive flit movement through the central links of the NoC raises the thermal activity in those areas leading to hot spot formation. Deviation of such traffic towards links at the edges of the mesh network helps to minimize the variation in thermal distribution across the network.

To study the thermal distribution in an 8x8 mesh NoC, simulations are conducted with various synthetic traffic patterns at pre-saturation injection rate. The amount of flits passing through each router is recorded and a Thermal Profile Graph (TPG) is drawn using these readings [66]. Figure 6.1 shows the TPG for uniform, transpose and shuffle traffic patterns. In a TPG, the area occupied by an 8x8 mesh NoC is represented by 49 squares placed like a 7x7 matrix. The corners of each square represent the routers and edges represent unidirectional links towards each router. The colour of each square represents the amount of traffic through the routers at its corners. This value is referred to as the traffic density of a square. During simulations, the number of flits passing through each router is recorded and the traffic density for each square is obtained by summing up the individual values of the corresponding corner routers. These values are coded by choosing an appropriate colour from the colour scale which has a transition from green to red. A square with deep green colour represents an area with least amount of traffic flow. Similarly, a dark red coloured square implies heavy traffic. From Figure 6.1 (a), it is observed that thermal activity is significantly higher at the central locations of the mesh (having more red squares) than at the edges and corners for all the three traffic patterns. The non-uniformity in thermal distribution is more evident for transpose traffic pattern (which has deep red squares from the centre towards right side) than random patterns such as uniform and shuffle.

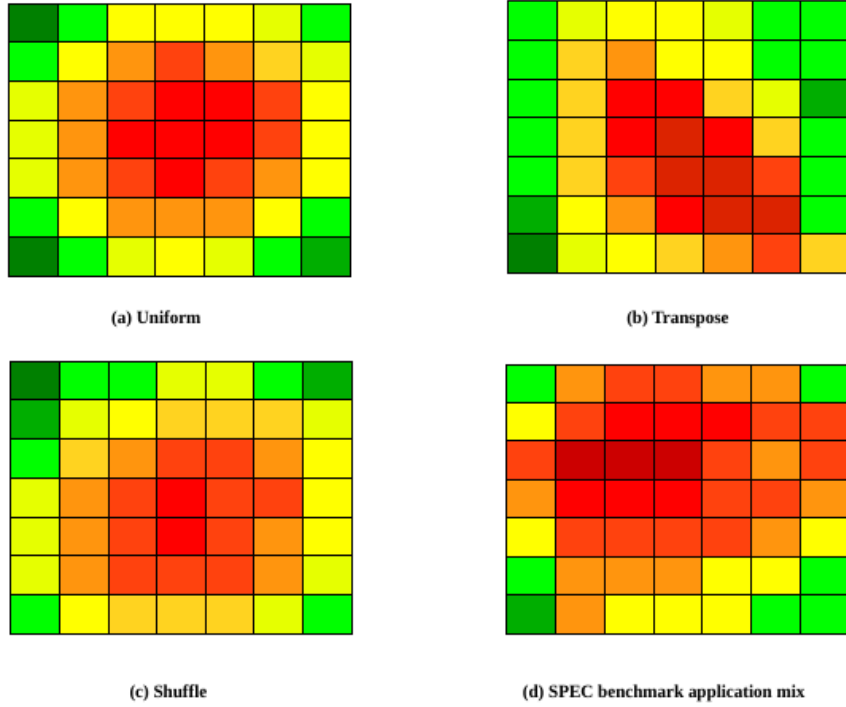


Figure 6.1 Thermal Profile Graph for CHIPPER in an 8x8 mesh NoC using synthetic and SPEC benchmark application traffic patterns

From this analysis, it is observed that thermal activity at the centre of the chip could be reduced by re-routing some flits in each router towards the edges of the mesh. A thermal aware deflection router that balances the load across NoC links by port reallocation of deflected flits is proposed in this chapter. The network performance is not compromised due to this arrangement since flits traversing along productive paths remain unaffected by the additional logic.

6.1 Router architecture

The router architecture consists of two pipeline stages as shown in Figure 6.2. The first stage consists of ejection and injection units which have similar functions as that of CHIPPER. The second stage mainly consists of the PDN which allocates output ports to flits in the four input lines on the basis of the prioritization and routing choice of each flit. The PDN is followed by a Port Reallocation Unit (PRU) which enables thermal aware routing. The golden flit prioritisation scheme introduced in CHIPPER for livelock avoidance is adopted by the proposed thermal aware router. Even though this prioritisation scheme is proven to be ineffective in improving network performance, it gives a fair comparison between the thermal variance in CHIPPER and the proposed routing technique..

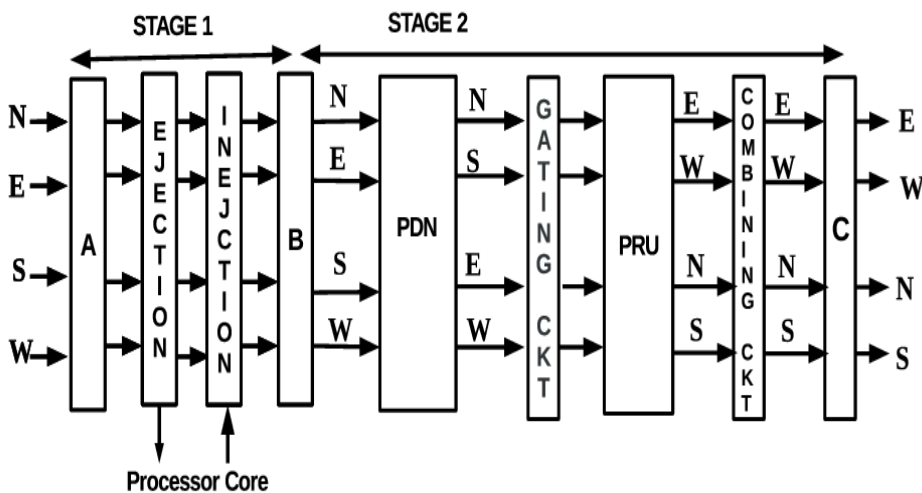


Figure 6.2 Two stage pipeline diagram of the proposed thermal aware deflection router.

6.1.1 Port Reallocation Unit

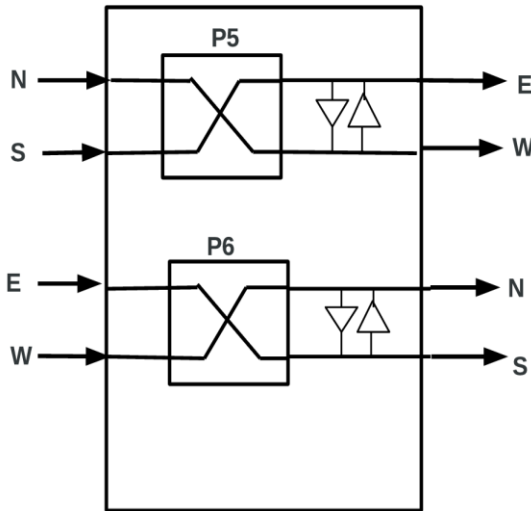


Figure 6.3 Structure of Port Reallocation Unit

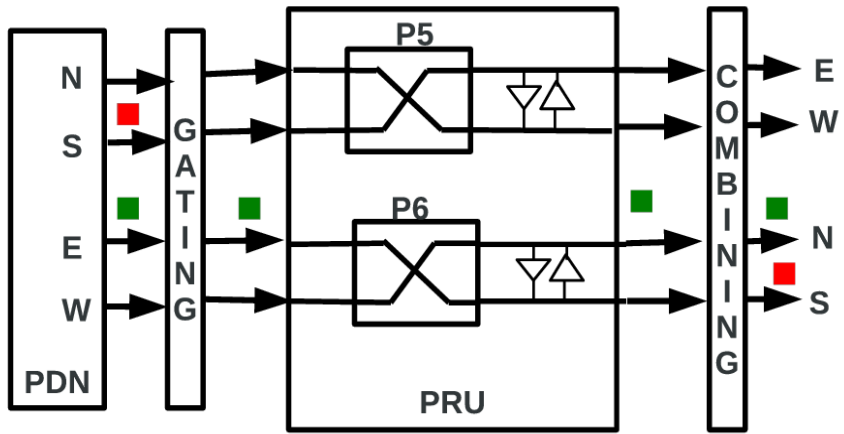
The function of PRU is to reallocate vacant output ports to the deflected flits such that they deflect away from the centre of the chip. A flit from an output line of the PDN is gated into the PRU if it satisfies all the three conditions given below:

- (1) the port assigned by the PDN is not in the productive direction for the flit
- (2) the output port assigned to the flit is directed towards the centre of the chip
- (3) an output port of the router directed away from the centre of the mesh is vacant.

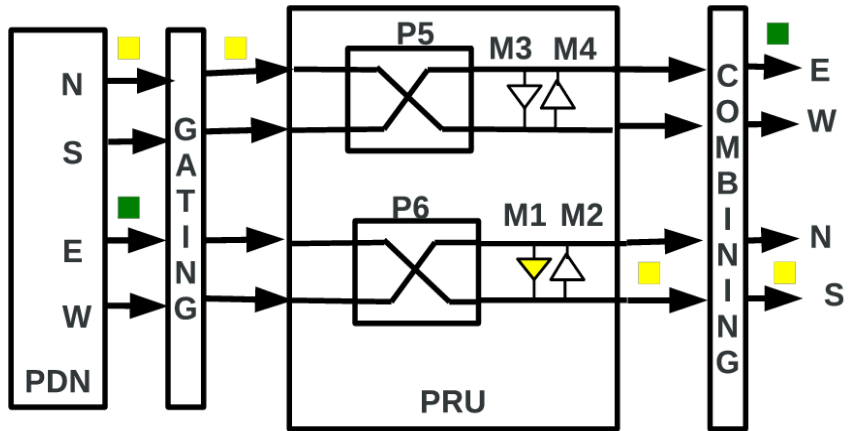
As shown in Figure 6.2, this conditional check and subsequent selective forwarding are done by a gating circuit.

The PRU consists of two permuter blocks (P5, P6), each having two inputs and two outputs. P5 reallocates flits from the north and south output lines of the PDN to the east or west ports of the router if any of them is vacant and directed away the centre of the network. Similarly, P6 connects east and west outputs of the PDN to north and south output ports. Reallocation of flits between north and south or east and west directions is enabled using multiplexers (M1, M2, M3, M4) between output lines of P5 and P6 as shown in Figure 6.3. The combining circuit multiplexes output lines of the PRU with corresponding output lines from the PDN. Flits at the output lines of PDN that do not satisfy any of the conditions explained above bypass the PRU and move into the output register C through the combining circuit.

The functionality of PRU is explained using examples shown in Figure 6.4. For a router in the second row and second column (top left) of an 8x8 mesh NoC, the south and east output ports are directed towards the centre of the mesh whereas the north and west ports are towards the edges. Figure 6.4(a) shows such a router where a deflected flit (green) at the east port is redirected towards the vacant north port by the permuter P6 in the PRU. Since the south port is a productive port for the other flit (red), it bypasses the PRU and moves towards register C. Figure 6.4(b) shows an example of port reallocation from north to south port using multiplexer M1. In this example, permuters P5 and P6 are inactive.



(a)



(b)

Figure 6.4 Port reallocation using PRU

6.2 Results and discussions

6.2.1 Simulation methodology

The thermal aware deflection router is modelled with two cycle latency by modifying NoC simulator, Booksim as explained in Section 6.2. In order to compare the results, the basic CHIPPER based NoC is also modelled in a similar way. Simulations are conducted using typical synthetic traffic patterns and network traces generated from multi-programmed workload mixes (M1 to M7 shown in Table 6.1) of SPEC CPU 2006 benchmark applications. The detailed analysis of the results is given in the following section.

Table 6.1 Mixes (M1 to M7) of various MPKI applications from SPEC CPU 2006 benchmark suite.

Applications	Benchmark Mix						
(%)	M1	M2	M3	M4	M5	M6	M7
Low MPKI	100	0	0	50	0	50	31
Medium MPKI	0	100	0	0	50	50	31
High MPKI	0	0	100	50	50	0	38

6.2.2 Thermal Profile

The TPG for uniform, transpose and shuffle traffic patterns for an 8x8 mesh NoC with the proposed thermal aware routing method is depicted in Figure 6.5. Compared to Figure 6.1, it is seen that the port reallocation strategy

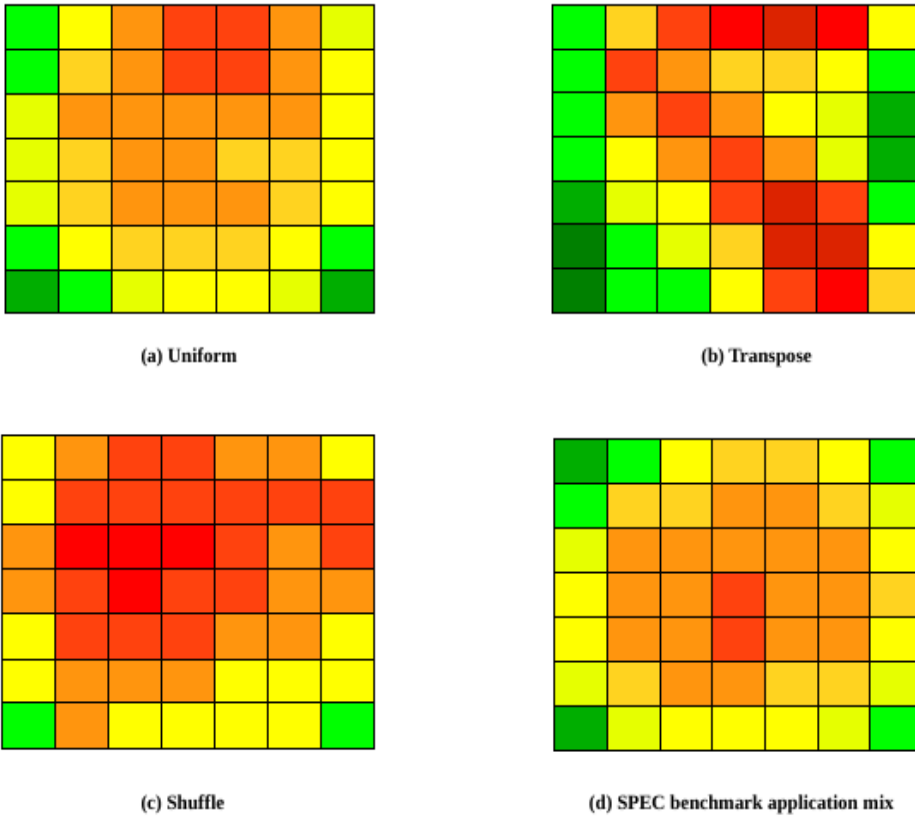


Figure 6.5 Thermal Profile Graph for an 8x8 mesh NoC with the proposed router using synthetic and SPEC benchmark application traffic patterns.

used in the proposed router helps to reduce the traffic density at the centre of the mesh and distribute it towards the edges. In Figure 6.5, the change in colour of squares at the centre from red to orange and yellow is seen for all the three traffic patterns. Similarly, the colours of squares at the edges change from deep green to pale green or yellow. The thermal activity in a square area is

proportional to the traffic density in the region. The similarity of colours in the TPG articulates the increase in uniformity of thermal distribution, which is a result of the proposed method.

6.2.3 Thermal Variance

In order to measure the uniformity in thermal distribution across the NoC, a parameter known as thermal variance [68] is introduced, which is calculated using equation 6.1.

$$ThermalVariance, V_T = \sum_{i=1}^N \frac{((A_v - T_i))}{N} \quad - (6.1)$$

where

$$Average, A_v = \sum_{i=1}^N \frac{T_i}{N} \quad - (6.2)$$

T_i - Traffic density of the i^{th} square in the TPG

N - Total number of squares in the TPG

In a NoC, lower value of thermal variance signifies that the uniformity in thermal distribution is higher. Using Equation 6.1, the thermal variance for CHIPPER and the proposed method are plotted for typical synthetic traffic patterns as given in Figure 6.6. In Figure 6.6, the value of V_T for the proposed method is normalised against that of CHIPPER, for which V_T is taken as 1. Simulations for low, pre-saturation and saturation injection rates show different behaviour in all the graphs. For all traffic patterns, the proposed routing method

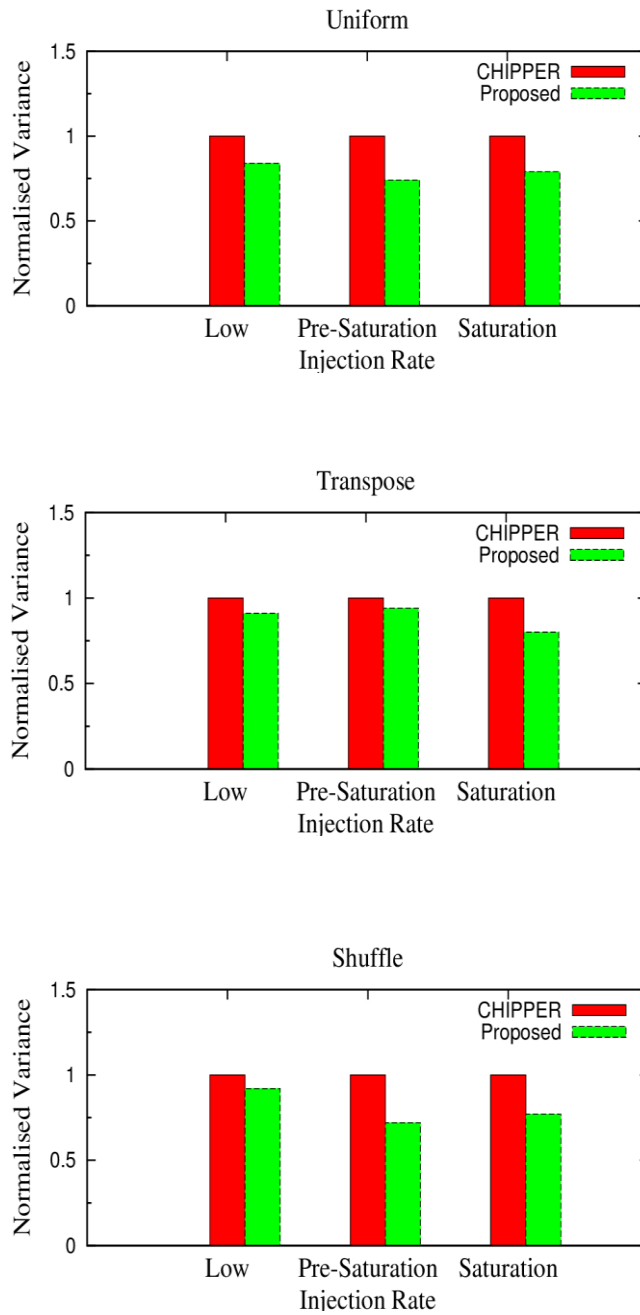


Figure 6.6 Normalised variance using synthetic traffic patterns for the proposed router w.r.t. CHIPPER in 8x8 mesh NoC .

exhibits lower variance compared to CHIPPER. For uniform and shuffle traffic, thermal variance of the proposed method is 26% and 23% lower than CHIPPER at pre-saturation. Transpose pattern represents network intensive traffic with specific packet destinations. Consequently, there is lesser opportunity for reallocating deflected flits to the edges of the NoC, therefore it shows only minor reduction in variance.

6.2.4 Average latency and throughput

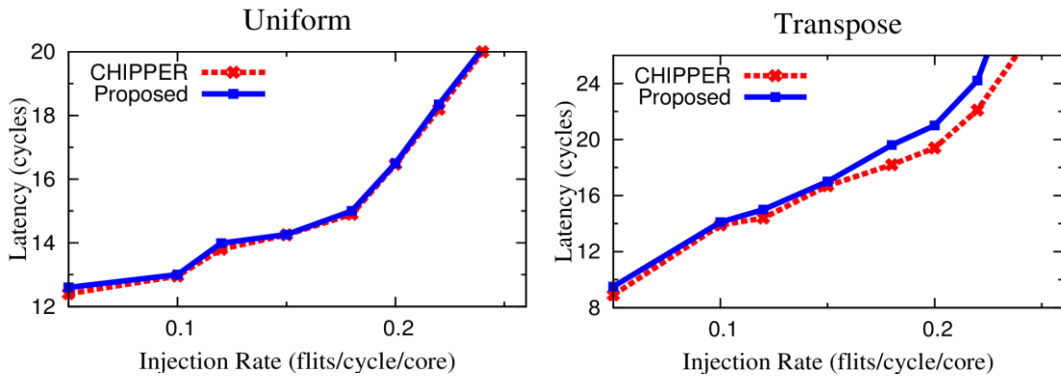


Figure 6.7 Average latency using synthetic traffic patterns for the proposed router and CHIPPER in 8x8 mesh NoC .

Figure 6.7 shows the comparison of average latency for uniform and transpose traffic patterns for 8x8 mesh NoC using CHIPPER and proposed router. As flits progressing in productive paths are unaffected by the introduction of traffic re-routing mechanism in the router pipeline, there is only negligible increase of 0.05% in average latency for the proposed mechanism. It is also found that the deflections per flit is reduced by up to 8% since edge

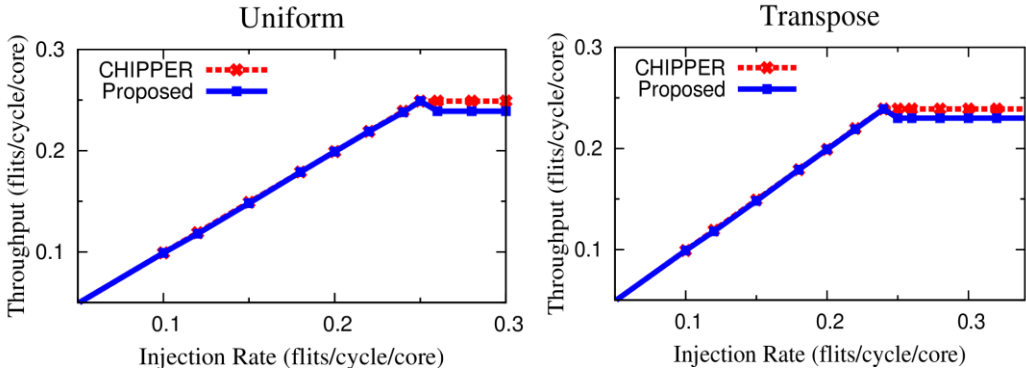


Figure 6.8 Average throughput using synthetic traffic patterns for the proposed router and CHIPPER in 8x8 mesh NoC

routers are lightly loaded and flits encounter lesser port conflicts in them. The average throughput delivered by CHIPPER and the proposed method are similar as shown in Figure 6.8.

6.2.5 Real applications

Figure 6.7 shows the graph of normalised variance for SPEC CPU 2006 workloads. Although the proposed method delivers promising results for all applications, significant reduction in thermal variance is noted for application mixes from M3 to M6. High MPKI applications inject more number of flits into the network, which increases the probability of deflections in the routers. Deflected flits that satisfy the condition for reallocation are forwarded to the edges/ corners of the NoC, due to which thermal variance is low.

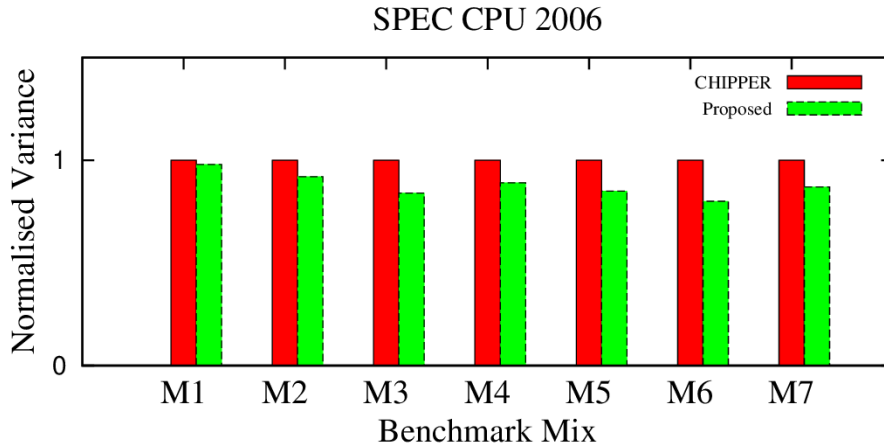


Figure 6.9 Normalised variance using applications from SPEC CPU 2006 benchmark suite for the proposed method w.r.t. CHIPPER in 8x8 mesh NoC.

6.2.6 Hardware synthesis

Verilog models of the proposed router and CHIPPER are synthesized using Synopsys Design Compiler with 65nm CMOS library. The first stage of CHIPPER and the proposed router have the same delay because of similar functional units in both architectures. Since both the architectures use the same port allocation logic, additional delay of 18% due to the PRU occurs in the output stage of the proposed router. Area and static power consumed by the control logic of the proposed router are 14% and 17% higher than CHIPPER. The hardware overhead and the extended critical path of the proposed technique are justified by the significant reduction in thermal variance that it promises.

6.3 Chapter summary

A deflection router which performs adaptive routing helps to achieve a uniform thermal distribution within the mesh network is proposed in this chapter. A logic block is introduced into the router pipeline that performs output port reallocation of flits that are assigned to unproductive directions towards the chip centre. The merit of the proposed thermal aware routing method is quantitatively justified by the reduction in its thermal variance parameter in comparison with a basic deflection router.

Chapter 7

Conclusions

Abstract

This chapter presents a conclusion of this dissertation. The main objectives and contributions of the research work are summarised with the benefits and limitations of the proposed techniques. A few directions for future research on this topic are also suggested.

The objectives of the research were to analyse the factors that curb the performance of two dimensional NoCs and to suggest measures that enhance their performance and reliability. The studies revealed that adaptive deflection routing techniques could be adopted to achieve reliable and energy efficient on chip communication with improved network performance parameters compared to related work.

The thesis started with an overview of NoC building blocks and conventional methods for routing packets. The merits of deflection routers in terms of performance, energy efficiency and fault tolerance were analysed through an extensive review of recent literature. Several adaptive routing techniques using deflection routers that meet the defined objectives have been proposed by the research.

The first proposed method used an novel strategy for output port selection in buffer-less and minimally buffered deflection routers. The proposed technique consists of an algorithm for prioritisation of flits based on the weighted deflection count of flits and a route computation method which helps them to reach their destinations with shortest delay. The method delivers reduced flit deflection rate and consequent reduction in dynamic power dissipation across the NoC links at the cost of nominal increase in static power due to the enhancement of link width.

Another deflection routing technique that reduces router delay by shortening its critical path and enabling single cycle router operation is put forward. The proposed method serves to improve the performance parameters of the NoC in terms of higher operation speed and reduced latency of flits.

A fault tolerant adaptive routing technique for deflection routers is proposed in the latter half of the dissertation. The technique of port reallocation of flits is adopted by this method for ensuring reliable routing at high fault rates with a graceful degradation in performance as the number of faults increases. A new model for representing faulty routers is evaluated and found that it offers improved path diversity for routing flits under high fault rates.

Issues related to non-uniformity in thermal distribution over the chip area are also investigated in this thesis. A thermal aware deflection routing method which reduces the traffic variance and improves the thermal balance by flit re-routing is suggested.

The implementation of routing algorithms based on the proposed techniques requires enhancements in the router hardware. The introduction of new functional blocks into the router's pipeline leads to an increase in the critical path length of the router. The consequent increase in router's area, static power and reduction in operating frequency of the network is indemnified by the benefits from reduced flit latency, deflection rate and dynamic power dissipation obtained for the newly proposed methods. The maximum operating frequency of the network in each proposed method was determined by synthesising Verilog models of the router's control path elements. Due to the non-availability of accurate wire models for the flit datapath in the router and infrastructure for full system implementation, the router delays were approximated as in some of the previous works discussed here.

7.1 Future scope of the research

The work undertaken in this thesis opens up opportunities for research in many directions. The fault tolerant router architecture proposed in chapter 5 provides resilience to permanent faults in NoC routers and links. With shrinking process technology, transient and intermittent faults may also occur frequently. Efficient methods for fault diagnosis in deflection router based NoCs and an integrated approach to tolerate permanent and transient faults may be investigated in future.

In chapter 6, the only factor considered in characterisation of the thermal profile inside the chip is the flit traffic through the NoC which leads to non-uniform thermal concentration. The power consuming applications running on the processor cores also result in thermal hotspots within the chip. The temperature effects of applications can also be considered in future for improving the effectiveness of the proposed thermal aware routing method.

Full system implementation of NoCs with the proposed routing techniques will help in more accurate and realistic computation of system parameters. It will also help in hardware optimisation leading to NoCs with enhanced performance and energy efficiency. This aspect is also undertaken as future work.

Another promising direction is the development of deflection routing techniques that guarantee quality of service for performance critical applications. New prioritisation policies and architectural extensions that improve application-level throughput of applications running on some

processor cores while maintaining fairness and starvation freedom for the others can be investigated.

By the recent success of 3D integration in ICs, NoCs have also expanded its limits to three dimensions of the chip. A 3D NOC uses 3D routers for inter-layer communication. A 3D router extended from a 2D router has two additional, up and down ports giving a total of six directional ports in addition to the local port. The performance of deflection routing NoCs is an interesting research area when scaled up by extending from 2D to 3D architecture using TSVs for inter-die connectivity. The fault tolerant and thermal aware deflection routing schemes proposed in the thesis can be extended in future, to suit the requirements of 3D NoC communication.

REFERENCES

- [1] **S. Borkar** (2007). Thousand core chips: a technology perspective. In Proc. 44th IEEE/ ACM Design Automation Conference. pp. 746-749.
- [2] **D. Wentzlaff, Patrick Griffin, H. Hoffmann, L. Bao, B. Edwards, C.Ramey, M. Mattin, C. Miao, J.F. Brown III and A. Agarwal** (2007). On-chip interconnection architecture of the tile processor. IEEE MICRO. 27(5). pp. 15 – 31.
- [3] **W. Dally, B. Towles** (2003). Principles and Practices of Interconnection Networks, Morgan Kaufmann Publishers.
- [4] **W. Dally** (1992). Virtual-Channel Flow Control. IEEE Transactions on Parallel and Distributed Systems. 3(2), 194–205.
- [5] **T. Moscibroda, O. Mutlu** (2009). A Case for Bufferless Routing in On-Chip Networks In Proc. 36th International Symposium on Computer Architecture, pp. 196–207.
- [6] **Z. Lu, M. Zhong and A. Jantsch** (2006). Evaluation of on-chip networks using deflection routing In Proc. 16th ACM Great Lakes Symposium on VLSI, pp. 296–301.
- [7] **Y. Hoskote, S. Vangal, A. Singh, N. Borkar, S. Borkar** (2007), A 5-GHz Mesh Interconnect for a Teraflops Processor. IEEE Micro 27(5) pp.51–61.

- [8] **M. B. Taylor, W. Lee, J. Miller, D. Wentzlaff, I. Bratt, B. Greenwald, H. Hoffman, P. Johnson, J. Kim, J. Psota, A. Saraf, N. Shnidman, V.Strumpen, M. Frank, S. Amarasinghe, A. Agarwal** (2004). Evaluation of the raw microprocessor: An exposed-wire-delay architecture for ILP and streams In Proc. 31st Annual International Symposium on Computer architecture, pp. 2–13.
- [9] **P. Gratz, C. Kim,R. McDonald, S.W. Keckler, and D.C. Burger** (2006). Implementation and evaluation of on-chip network architectures In Proc. IEEE International Conference on Computer Design. pp. 477–484
- [10] **C. Fallin, G. Nazario, X. Yu, K. Chang, R. Ausavarungnirun, O. Mutlu** (2012). MinBD: Minimally-Buffered Deflection Routing for Energy Efficient Interconnect In Proc. 6th IEEE/ACM International Symposium on Networks on Chip (NoCS). pp. 1–10.
- [11] **C. Fallin, C. Craik, O. Mutlu** (2011). CHIPPER: A Low Complexity Bufferless Deflection Router In Proc. IEEE 17th International Symposium on High Performance Computer Architecture. pp. 144–155.
- [12] **G. Nychis, C. Fallin, T. Moscibroda and O. Mutlu** (2010). Next generation on-chip networks: What kind of congestion control do we need? In Proc. 9th ACM SIGCOMM Workshop on Hot Topics in Networks. pp.1–6.

- [13] **G. Michelogiannakis, J. Balfour and W. Dally** (2009). Elastic-Buffer Flow Control for On-Chip Networks. In Proc. 15th International Symposium on High-Performance Computer Architecture. pp. 151-162
- [14] **Y. Turner and Y. Tamir.** (2007). Deadlock-free connection based adaptive routing with dynamic virtual circuits. Journal of Parallel and Distributed Computing. 67(1). pp. 13-32.
- [15] **C. Nicopoulos, D. Park, J. Kim, V. Narayanan, M. S. Yousif and C. R. Das** (2006). ViChaR : A dynamic virtual channel regulator for Network-on-Chip Routers In Proc. 39th IEEE/ ACM Annual International Symposium on Microarchitecture. pp. 333-346.
- [16] **C. Nicopoulos, A. Yanamandra, S. Sreenivasan, V. Narayanan, and M. J. Irwin** (2007). Variation Aware Low Power Buffer Design In Proc. 43rd ASILOMAR conference on Signals, Systems and Computers. pp. 1402-1406.
- [17] **T. Bjerregaard and S. Mahadevan** (2006). A survey of research and practices of network-on-chip. ACM Computing Survey 38(1). pp. 1-51.
- [18] **M. Li, Q. Zeng, and W. Jone** (2006) DyXY - a proximity congestion-aware deadlock-free dynamic routing method for network on chip. In Proceedings of 43rd ACM/IEEE Design Automation Conference. pp. 849 – 852.
- [19] **M. Ebrahimi, M. Daneshtalab, J. Plosila and F. Mehdipour**(2013). MD: Minimal path-based Fault-Tolerant Routing in On-Chip Networks.

- In Proc. 18Th Asia and South Pacific Design Automation Conference. pp. 35-40.
- [20] **P. Lotfi-Kamran, A. M. Rahmani, M. Daneshtalab, A. Afzali-Kusha and Z. Navabi** (2010). EDXY – A low cost congestion-aware routing algorithm for network-on-chips. *Journal of Systems Architecture* 56 (7). pp. 256–264.
- [21] **G. Ascia, V. Catania, M. Palesi and D. Patti** (2008). Implementation and Analysis of a New Selection Strategy for Adaptive Routing in Networks-on-Chip. *IEEE Transactions on Computers*. 57 (6). pp. 809 – 820.
- [22] **S. Ma, N. E. Jerger and Z. Wang** (2011). DBAR: an efficient routing algorithm to support multiple concurrent applications in networks-on-chip In Proc. 38th Annual International Symposium on Computer Architecture, pp. 413–424.
- [23] **P. Gratz, B. Grot and S. W. Keckler** (2008). Regional congestion awareness for load balance in networks-on-chip In Proc. IEEE 14th International Symposium on High Performance Computer Architecture, pp. 203 –214.
- [24] **G. Chiu** (2000). The Odd-Even Turn Model for Adaptive Routing. *IEEE Transactions on Parallel and Distributed Systems*. 11(7). pp. 729–738.
- [25] **M. Hayenga, N. E. Jerger and M. Lipasti** (2009). SCARAB: A single cycle adaptive routing and bufferless network In Proc. 42nd

- IEEE/ACM International Symposium on Microarchitecture. pp. 244–254.
- [26] **G. Oxman and S. Weiss** (2016). Deflection Routing in Hierarchical Mesh NoCs. *IEEE Embedded Systems Letters*. 8 (2). pp. 45-48.
- [27] **J. Jose, B. Nayak, K. Kumar and M. Mutyam** (2013). DeBAR: Deflection Based Adaptive Router with Minimal Buffering In *Proc. Design, Automation and Test in Europe*. pp. 1583–1588.
- [28] **B. Nayak, J. Jose and M. Mutyam** (2013). SLIDER: Smart Late Injection DEFlection Router for Mesh NoCs In *31st IEEE International Conference on Computer Design*. pp. 377–383.
- [29] **J. Lin, X. Lin and L. Tang** (2012). Making-a-stop: A new bufferless routing algorithm for on-chip network. *Journal of Parallel and Distributed Computing*. 72 (4). pp. 515–524. Elseiver.
- [30] **G. R. Joanna, J. Jose, R. Radhakrishnan and M. Mutyam** (2014). MinBSD: Minimally Buffered Single Cycle Deflection Router In *Proc. Design, Automation and Test in Europe*. pp. 1–4.
- [31] **S. Shamshiri, A. Ghofrani and K. Cheng** (2011). End-to-end error correction and online diagnosis for on-chip networks In *Proc. IEEE International Test Conference*, pp. 1-10
- [32] **S. Murali, T. Theocharides, N. Vijaykrishnan, M.J. Irwin, L. Benini and G. De Micheli** (2005). Analysis of error recovery schemes for

- networks on chips. *IEEE Design and Test of Computers*. 22 (5). pp. 434–442.
- [33] **L. Huang, X. Lin and J. Wang, Q. Li** (2017). A low latency fault tolerant transmission mechanism for Network-on-Chip. In *Proc. IEEE International Symposium on Circuits and Systems*. pp. 1-4.
- [34] **S. Borkar** (2005). Designing reliable systems from unreliable components: the challenges of transistor variability and degradation. *IEEE MICRO*. 25(6). pp. 10-16.
- [35] **A. Ghofrani, R. Parikh, S. Shamshiri, A. DeOrio, K. Cheng and V. Bertacco** (2012). Comprehensive online defect diagnosis in on-chip networks. In *Proc. 30th IEEE VLSI Test Symposium*. pp. 44-49.
- [36] **A. De Orio, D. Fick, V. Bertacco, D. Sylvester, D. Blaauw, J. Hu and G. Chen** (2012). A reliable routing architecture and algorithm for NoCs. *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*. 31(5). pp. 726-739.
- [37] **K. Aisopos, A. DeOrio, L. Peh and V. Bertacco** (2011). ARIADNE: Agnostic Reconfiguration in a Disconnected Network Environment In *Proc. 20th International Conference on Parallel Architectures and Compilation Techniques*. pp. 298–309.
- [38] **V. Puente, J. A. Gregorio, F. Vallejo and R. Beivide** (2004). Immunet: A Cheap and Robust Fault-tolerant Packet Routing Mechanism. *Computer Architecture News*, 32(2). pp. 198–209. ACM SIG-ARCH.

- [39] **W.-C. Tsai, D. Zheng, S. Chen, and Y. Hu** (2011). A Fault-Tolerant NoC Scheme using bidirectional channel. In Proc 48th ACM/EDAC/IEEE Design Automation Conference. pp. 918 –923.
- [40] **D. Fick, A. DeOrio, J. Hu, V. Bertacco, D. Blaauw and D. Sylvester** (2009). Vicis: A Reliable Network for Unreliable Silicon In Proc. 46th ACM/ IEEE Design Automation Conference. pp. 812–817.
- [41] **R. Parikh and V. Bertacco** (2013). uDIREC : unified Diagnosis and Reconfiguration for Frugal Bypass of NoC Faults In Proc. 46th IEEE/ACM International Symposium on Microarchitecture. pp.148–159.
- [42] **Y. He and G. Chen** (2015). An inclusive fault model for Network-on-Chip. In Proc. IEEE 11th International Conference on ASIC, 2015, pp.1-4.
- [43] **M. Balboni, J. Flich and D. Bertozzi** (2015). Synergistic Use of Multiple On-Chip Networks for Ultra-Low Latency and Scalable Distributed Routing Reconfiguration. In Proc. Design, Automation & Test in Europe Conference & Exhibition. pp. 806–811.
- [44] **R. Bishnoy, V. Laxmi, M. S. Gaur and J. Flich** (2015). d²-LBDR: Distance-Driven Routing to Handle Permanent Failures in 2D Mesh NoCs In Design, Automation and Test in Europe. pp. 800–805.

- [45] **S. Rodrigo, S. Medardoni, J. Flich, D. Bertozzi and J. Duato** (2009). Efficient Implementation of Distributed Routing Algorithms for NoCs, *IET Computers and Digital Techniques*. 3 (5). pp. 460–475.
- [46] **S. Rodrigo, J. Flich, A. Roca, S. Medardoni, D. Bertrtozzi, J. Camacho, F. Silla and J. Duato** (2011). Cost-Efficient On-Chip Routing Implementations for CMP and MPSoC Systems. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*. 30(4). pp. 534-547.
- [47] **D. Fick, A. DeOrio, G. Chen, V. Bertacco, D. Sylvester and D. Blaauw** (2009). A Highly Resilient Routing Algorithm for Fault-Tolerant NoCs In *Proc. Design, Automation and Test in Europe*. pp. 21–26.
- [48] **E. Wachter, A. Erichsen, A. Amory and F. Moraes** (2013). Topology-Agnostic Fault-Tolerant NoC Routing Method In *Design, Automation and Test in Europe*. pp. 1595–1600.
- [49] **C. Feng, Z. Lu, A. Jantsch, M. Zhang and Z. Xing** (2013). Addressing Transient and Permanent Faults in NoC With Efficient Fault-Tolerant Deflection Router, *IEEE Transactions on VLSI*. 21 (6). pp. 1053–1066.
- [50] **M. Ebrahimi, M. Daneshtalab, J. Plosila and H. Tenhunen** (2012). MAFA: Adaptive Fault-Tolerant Routing Algorithm for Networks-on-Chip In *Proc. 15Th IEEE Euromicro Conference on Digital System Design (DSD)*. pp. 201-206.

- [51] **J. Wu** (2003). A Fault-Tolerant and Deadlock-Free Routing Protocol in 2D Meshes Based on Odd-Even Turn Model. *IEEE Transaction on Computers*, 52 (9). pp. 1154–1169.
- [52] **Z. Zhang, A. Greiner and S. Taktak** (2008). A Reconfigurable Routing Algorithm for a Fault-Tolerant 2D Mesh Network-on-Chip In Proc. 45th ACM/IEEE Design Automation Conference. pp. 441–446.
- [53] **C. Iordanou, V. Soteriou and K. Aisopos** (2014). Hermes: Architecting a Top-Performing Fault-Tolerant Routing Algorithm for Networks-on-Chips In Proc. IEEE International Conference on Computer Design, pp. 424–431.
- [54] **Y. Zou and S. Pasricha** (2010). NARCO: Neighbor Aware Turn Model-Based Fault Tolerant Routing for NoCs. *IEEE Embedded Systems Letters*. 2(3). pp 85–89.
- [55] **P. Ren, Q. Meng, X. Ren and N. Zheng** (2014). Fault-tolerant routing for on-chip network without using virtual channels. In Proc. 51st IEEE/EDAC/ACM Design Automation Conference. pp. 1-6.
- [56] **A. Kohler, G. Schley and M. Radetzki** (2010). Fault tolerant Network on Chip switching with graceful performance degradation, *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems* 29 (6). pp. 883–896.

- [57] **C. Feng, Z. Lu, A. Jantsch, J. Li, M. Zhang** (2010). FoN: Fault-on-Neighbor Aware Routing Algorithm for Networks-on-Chip In Proc. IEEE International SOC Conference. pp. 441–446.
- [58] **A. Rounge** (2015). Fault Tolerant Network on Chip based on Fault Aware Flits and Deflection Routing In Proc. 9th International Symposium on Networks-on-Chip. pp. 9–16.
- [59] **M. Fattah, A. Airola, R. Ausavarungnirun, N. Mirzaei, P. Liljeberg, J. Plosila, S. Mohammadi, T. Pahikkala, O. Mutlu and H. Tenhunen** (2015). A Low Overhead, Fully Distributed, Guaranteed Delivery Routing Algorithm for Faulty Network-on-Chips In Proc. 9th International Symposium on Networks-on-Chip. pp. 1–8.
- [60] **P. Bose and P. Morin** (2001). Routing with Guaranteed Delivery in Ad Hoc Wireless Networks. Wireless Networks, Vol.7. pp. 609–616.
- [61] **K. Manna, P. Mukherji, S. Chattopadhyay, I. Sengupta** (2017) Thermal-Aware Application Mapping Strategy for Network-on-Chip Based system Design In IEEE Transactions on Computers. 67(4). pp. 528- 542.
- [62] **Yanhua, L., Y. Ruan, Z. Lai, W. Jing** (2011) Energy and thermal aware mapping for mesh-based NoC architectures using multi-objective ant colony algorithm In Proc. 3rd International Conference on Computer Research and Development. pp. 25-29.
- [63] **P. Gratz, B. Grot, S.W. Keckler** (2008). Regional Congestion

- Awareness for Load Balance in Networks-on-Chip In Proc. IEEE 14th International Symposium on High-Performance Computer Architecture, pp. 203–215.
- [64] **M. Ramakrishna, P. V. Gratz, A. Sprintson** (2013). GCA: Global Congestion Awareness for Load Balance in Networks-on-Chip. In Proc. 7th IEEE/ACM International Symposium on Networks on Chip, pp. 21–24.
- [65] **M. Ebrahimi, M. Daneshtalab and J. Plosila** (2012). GLB - Efficient Global Load Balancing Method for Moderating Congestion in On-Chip Networks. In Proc. 7th IEEE International Symposium on Reconfigurable Communication-centric Systems-on-Chip. pp. 1-5.
- [66] **J. Jose, J. S. Shankar, K.V. Mahathi, D. K. Kumar, M. Mutyam** (2011). BOFAR : Buffer Occupancy Factor based Adaptive Router for mesh NoC. In Proc. 4th International workshop on Network on Chip Architectures, 2011, pp. 23–28.
- [67] **K. Bhardwaj, K. Chakraborty, S. Roy** (2012). Towards Graceful Aging Degradation in NoCs through an Adaptive Routing Algorithm In Proc. 49th IEEE/ACM Design Automation Conference. pp. 382–391.
- [68] **J. Jose, B. Jacob and H. M. Kamal** (2014). An Energy Efficient Load Balancing Selection Strategy for Adaptive NoC Routers In Proc. International workshop on Network on Chip Architectures. pp. 31–36.

- [69] **E. Nilsson, M. Millberg, J. Oberg and A. Jantsch** (2003). Load Distribution with the Proximity Congestion Awareness in a Network on Chip. In Proc. Design, Automation and Test in Europe Conference. pp. 11126.
- [70] **N. Jiang, G. Michelogiannakis, D. Becker, B. Towles and W. Dally** (2013). Booksim 2.0 User's Guide. <http://nocs.stanford.edu>.
- [71] **J. Henning.**, SPEC CPU 2006 Benchmark Descriptions, in: SIGARCH Computer Architecture News, 2006.
- [72] **R. Ubal, J. Sahuquillo, S. Petit, P. Lopez** (2007). Multi2sim: A simulation framework to evaluate multicore-multithreaded Processors In Proc. 26th International Symposium on Computer Architecture and High Performance Computing, Munich, Germany. pp. 62–68.
- [73] **A. B. Kahng, B. Li, L. Peh, K. Samadi** (2012). Orion 2.0: A Fast and Accurate NoC Power and Area Model for Early Stage Design Space Exploration. IEEE Transactions on VLSI. 20 (1). pp. 191–196.
- [74] **S.S.I. Association**, International Technology Roadmap for Semiconductors Interconnect, 2009, <http://public.itrs.net/>.
- [75] **Lee, J., Nicopoulos, C., Sung, J.P., M. Swaminathan, J. Kim** (2013). Do We Need Wide Flits in Networks-On-Chip? In Proc. IEEE Computer Society Annual Symposium on VLSI. pp. 2–7.

Publications

Journals :

1. **Simi Zerine Sleetba**, John Jose, Mini M.G. (2018). An Energy Efficient Fault Tolerant Technique for Deflection Routers in 2D Mesh NoCs. IET Computers and Digital Techniques. Vol.12, No. 3, 2018.pp. 69-79.
DOI: 10.1049/iet-cdt.2017.0006.
2. **Simi Zerine Sleetba**, Mini M.G. (2017). An Enhanced Model for Reliable Deflection Routing in Mesh Network On Chip, International Journal of High Performance System Architecture. Inderscience Publishers. Vol.7, No. 2, 2017. 87-97. DOI: 10.1504/IJHPSA.2017.10008113.
3. **Simi Zerine Sleetba**, Mini M.G. (2015). Minimally Buffered Router using Weighted Deflection Routing for Mesh Network on Chip. International Journal of VLSI Design and Communication Systems, AIRCC Publishers. Vol.6, No.3, 2015. pp. 59-71.DOI : 10.5121/vlsic.2015.6306.

Conferences:

1. **Simi Zerine Sleetba**, John Jose, Maurizio Palesi, Rekha James, Mini M.G. (2018). Traffic Aware Deflection Re-routing Mechanism for Mesh Network on Chip in Proceedings of the 26th IFIP/IEEE/AICA International Conference on Very Large Scale Integration, VLSI-SoC, Verona, Italy, October, 2018. (Accepted)
2. **Simi Zerine Sleetba**, John Jose, Mini M.G. (2015). HiPAD: High Performance Adaptive Deflection Router for On Chip Mesh Networks In

Proceeding of the 5th International Conference on Advances in Computing and Communications, Kochi, India, IEEE Xplore, September, 2015, pp. 16-19.

3. **Simi Zerine Sreeba**, John Jose, Mini M.G.(2014). WeDBless: Weighted Deflection Buffer-less Router for Mesh NoCs. In Proceedings of Great Lakes Symposium on VLSI (GLSVLSI) , Houston, USA, ACM Digital Library, May, 2014, pp. 77-78.

PhD Forums:

1. **Simi Zerine Sreeba** and Mini M.G. (2016) High Performance and Reliability Techniques in Deflection Routers for 2D Network on Chip. In Proceedings of the PhD Forum @ 20th International Symposium on Design and Test (VDAT-2016), IIT Guwahati. May, 2016.

Curriculum Vitae

Name : Simi Zerine Sleeba
Gender : Female
Date of Birth : 27-03-1976
Address : Vettikunnel House,
BMC Kollamkudimugal Road,
Thrikkakara P.O., Kochi
Ph: 0484-2425762
Mob: 9496805762
e-mail: simi.abie@gmail.com

Educational Qualifications:

- May, 1997
B.Tech in Electronics and Communication Engineering,
M.A. College of Engineering, Kothamangalam, Kerala, India.
- May, 2010
M.Tech in VLSI and Embedded Systems
Model Engineering College, Kochi, India.
- December, 2011 till Present
Ph.D.
Department of Electronics, Model Engineering College, Kochi, India.

Professional Experience :

- May, 1998 – May, 2003 : Software Engineer,
CMC Ltd., NewDelhi and
Mission Informatics Pvt. Ltd., Kochi.
- June, 2003 –Dec, 2011 : Assistant Professor,
Viswajyothi College of Engineering and
Technology, Muvattupuzha, Kerala, India.
- Dec, 2011 – Present : Research Scholar,
Model Engineering College, Kochi, India.

Areas of Interest : On chip communication architectures,
Chip multiprocessors,
Fault-tolerant systems.