

Micro-ring Fault-resilient Photonic On-chip Network for Reliable High-performance Many-core Systems-on-Chip

Michael Conrad Meyer

A DISSERTATION

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
IN COMPUTER SCIENCE AND ENGINEERING



Adaptive Systems Laboratory
Graduate Department of Computer and Information Systems
University of Aizu, Japan

March 2017

The thesis titled

Micro-ring Fault-resilient Photonic On-chip
Network for Reliable High-performance Many-core
Systems-on-Chip

by


Michael Conrad Meyer

is reviewed and approved by:

Chief referee

Professor

Abderazek Ben Abdallah

Ben A. Abdallah 


Professor

Toshiyuki Miyazaki

Toshiyuki Miyazaki 

Professor

Tsuneo Tsukahara

Tsuneo Tsukahara 


Professor

Junji Kitamichi

Junji Kitamichi 

Senior Associate Professor

Yukihide Kohira

Yukihide Kohira 

University of Aizu

March 2017

*Dedicated to
my family*

Micro-ring Fault-resilient Photonic On-chip Network for Reliable High-performance Many-core Systems-on-Chip

Michael Conrad Meyer

Submitted for the Degree of Doctor of Philosophy

March 2017

Abstract

Humans continue to demand higher performance from their computing systems, and as a result we have had aggressive increases in the scaling of technology, but this is showing signs of change. The power consumed by a chip is ever increasing, and recently the power efficiency of communications has become as important as the computational power of the cores. Typical electronic Networks-on-Chip (NoCs) are reaching their performance limitations thanks to various factors.

One highly sought after technology is Photonic Networks-on-Chip (PNoCs). PNoCs offer several benefits over conventional electrical NoCs, such as high-bandwidth support, distance independent power consumption, lower latency, and improved performance-per-watt. Wavelength Division Multiplexing allows for multiple parallel optical streams of data to concurrently transfer through a single waveguide and MRs can be switched at speeds as high as 40 GHz to realize wavelength-selective modulators or switches. These technologies allow for multiple bits of data to travel concurrently through the same waveguide, which contradicts the one bit per wire limitation of electronic circuits. Another benefit is that data is transferred in an end-to-end fashion once a path is configured, meaning that the data does not need to be buffered multiple times, and thus saving power.

The photonic domain is immune to transient faults caused by radiation, but is still susceptible to process variation (PV) and thermal variation (TV) as well as aging. The aging typically occurs faster in active components as well as elements

that have high thermal variation. In the optical domain, faults can occur in MRs, waveguides, routers, etc. Active components, such as photodetectors, have higher failure rates than passive components, e.g. waveguides. Moreover, when paired with the fact that a PNoC is highly vulnerable, as a fault may expose the single-point failure, a faulty MR can cause a message to be misdelivered or lost. In this dissertation, a set of novel photonic routing algorithms and architectures are proposed for future on-chip optical networks.

First, a new fault tolerant photonic switch, capable of handling multiple faulty MRs. The proposed switch is based on a non-blocking 5-port optical router. It requires no MRs to travel in the opposite direction (e.g. East to West or North to South). The switch is also able to handle the previous hybrid spatial switching used in PHENIC.

Second, a fault tolerant Path configuration algorithm, which checks for MR faults and allocates the proper MRs to be used. This means that our previous 2 state MRST must also have a faulty state. Additionally, the algorithm must use two MR Configuration Tables, one for standard use and another to be used for the backup paths. This makes all of the routing decisions within a single optical switch.

Third, a power estimation scheme for the optical layer, which is fast enough to be used for routing decisions. Because of the speed that the calculation must be done, the calculation itself must be simple.

Finally, I propose an architecture and routing algorithm pair, which allow for the network to make “strain” based decisions for the routing. This strain value is based on the number of faulty MRs and the optical power of a node. This should improve the network's reliability and performance by avoiding nodes with high temperature, a high number of faulty nodes, or a lot of traffic.

The proposed architectures and algorithms were evaluated with a discrete-event simulator, which incorporates detailed physical models of the photonic components. Results show that the proposed system was able to achieve a higher reliability with minimal sacrifices in the overall system performance and energy. The resulting system is able to address the problems of process variation as well as temperature variation in optical components, and is more reliable than previous existing systems.

高性能なメニーコアシステムオンチップの為の マイクロリングの障害耐性を持つ 光通信オンチップネットワーク

マイヤー マイケル コンラッド

博士号学位のために 2017 年 3 月に提出

概要

人々はコンピューティング・システムにおいて、常により高度な機能を求め続け、その結果、技術の規模を飛躍的に拡大してきました。しかし、この状況にも変化の兆しが見えてきました。一つの IC チップ上で消費される電力は増加の一途をたどり、近年では CPU のコアの能力そのものと同等に、そこで消費される電力の効率性も重要視されてきました。現在、電子ネットワークオンチップ (NoCs) は様々な要因によって性能の限界に近づいてきています。

オンチップ光ネットワーク (PNoCs) は現在、研究対象として最も注目されているものの一つです。PNoCs はこれまでの電子 NoCs に比べ幾つかの点で優位性を示しております。それらは、高帯域幅でのサポート、距離に依存しない消費電力、レイテンシや 1 ワットあたりの性能などです。波長分割多重通信は複数の並列での光の流れを一つの導波管で行うことができ、MR は 40GHz の速度での切り替えを可能で、波長選択変調器またはスイッチとして使用することができます。このことは、複数ビットのデータを同じ導波管で同時に伝達することが可能ということで、一つの導線に 1 ビットという制限のある電子回路とは異なるものです。また、他の利点としては、一度経路が形成されてしまえば、データは起点から終点まで一気に到達することが挙げられます。つまり、データのバッファを何度も行う必要がないということであり、消費電力を抑えることにもつながります。

フォトリック領域は放熱によって瞬間的な故障を引き起こすことはありませんが、経年劣化はもちろん工程変動 (PV) や熱変動 (TV) には影響を受けます。劣化は動的な構成品及び温度変化の高い部分で発生します。オプティカル領域では、MR や導波管、ルータなどで故障が発生します。動的機器の光検出器などは、導波管などの受動的機器に比べ故障発生割合が高くなります。また、PNoC に高い脆弱性が

あるとされた場合、故障により単一障害が発生する可能性や、故障した MR がメッセージの誤伝達や喪失を起こす可能性があります。

本論文では、将来のオンチップ光ネットワークのための、新たな光回路アルゴリズムとその構造を提案します。

第一に、複数の故障した MR を処理する能力を持つ、新たな障害耐性のある光スイッチを提案します。このスイッチは5つのノンブロッキングポートを持つ光ルータをベースにしています。これは、MR が反対方向に進まないように（東から西へまたは北から南へ）する必要があります。またこのスイッチは PHENIC で使用されたハイブリッド空間スイッチングも取り扱うことが可能です。

第二に、MR の故障をチェックし、適切な MR を割り当てる障害耐性パス設定アルゴリズムを提案します。これは以前の二つの MRST の一つが異常状態にあるということの意味しています。さらに、これには2つの MR 設定テーブルを使用しなければなりません。一つは通常使用のため、もう一つはバックアップパスのためとなります。これにより、全てのルーティングの決定が一つの光スイッチの中で行われます。

第三に、オプティカル層での電力見積りのスキームを提案します。これにはルート決定に使用できるほどの速さが必要です。計算のスピードが要求されるため、計算そのものが単純でなければなりません。

最後に、ネットワークが「歪み」に基づいたルーティングの決定を可能とする回路アルゴリズムとその構造を提案します。この歪み値は故障した MR の数とノードの光力に基づきます。高温のノードや故障数の多いノード、高通信量を避けることにより、ネットワークの信頼性と性能が進展することになります。

提案したアルゴリズムと構造を、複数のフォトニック部品を組み込んだ詳細な物理モデルの分散型シミュレーターで評価を行い、その結果、性能と電力において最小の犠牲で、高い信頼性を得ることができました。完成したシステムはオプティカル構成における工程変動および熱変動の問題にも対処でき、これまでに存在したものよりも信頼できるシステムとなりました。

Declaration

I, Michael Meyer, D8161104: hereby declare that this dissertation entitled “Micro-ring Fault-resilient Photonic On-chip Network for Reliable High-performance Many-core Systems-on-Chip” represents my original work carried out as a doctoral student of the UoA and to the best of my knowledge, it contains no material previously published nor any material presented for the award of any degree or diploma of any other institution. Any contribution made to this research by others with whom I have worked at the UoA or elsewhere is explicitly acknowledged in the dissertation. Works of other authors cited in this dissertation have been duly acknowledged under the section “Bibliography”.

Date: February 22 2017

Copyright © 2017 by Michael Conrad Meyer.

“The copyright of this thesis rests with the author. No quotations from it should be published without the author’s prior written consent and information derived from it should be acknowledged”.

Acknowledgements

I would like to express my sincere gratitude to thank my advisor, Professor Abderazek Ben Abdallah, for his guidance over the past three years. He has guided me on not only just writing and research, but also on professional expectations of doctors.

I would also like to thank Professor Toshiyaki Miyazai, Professor Tsuneo Tsukahara, Professor Junji Kitamichi, and Professor Yukihide Kohira of the University of Aizu for taking the time to review my thesis. I would also like to thank Prof. Yuichi Okuyama for his help over the past three years.

I would like to thank all my friends, at home and in Japan. Current and previous members of the Adaptive Systems Laboratory at the University of Aizu have helped me with my life in Japan, and have made the whole experience more enjoyable.

Lastly, I would like to thank my family for all their love and encouragement. For my parents who raised me with a love of science and supported me in all my pursuits.

Contents

Abstract	iv
Declaration	viii
Acknowledgments	ix
1 Introduction	1
1.1 Background	1
1.2 Current System Design	1
1.3 Network On Chips	4
1.4 Photonic Interconnects	6
1.5 Reliability Issues in Photonic Networks-on-Chip	7
1.6 Thesis Objectives and Contributions	8
1.7 Thesis Outline	10
2 Background	11
2.1 Photonic NoCs	11
2.1.1 Circuit Switching	11
2.1.2 Wavelength-Routed	13
2.1.3 Photonic Communication	14
2.1.4 Photonic NoC Components	16
2.1.4.1 Laser	16
2.1.4.2 Waveguide	17
2.1.4.3 Modulator	17
2.1.4.4 Photodetector	18

2.1.4.5	Micro-Ring Resonator	18
2.2	Fault Models	20
2.2.1	Photonic NoC Signal Strength	20
2.2.2	Electrostatic Discharge	20
2.2.3	Noise	21
2.2.4	Aging	21
2.2.5	Process Variability	22
2.2.6	Temperature Variation	23
2.3	Chapter Summary	23
3	Related Works	25
3.1	Conventional PNoCs	25
3.2	PNoC Fault-Tolerance	27
3.2.1	Rerouting	27
3.2.1.0.1	Fault Regions	28
3.2.1.0.2	Look Ahead Routing	28
3.2.1.0.3	Buffering and Checking	29
3.2.2	Hardware Redundancy	29
3.2.3	Tuning	30
3.3	Other Usable Fault-Tolerance Schemes	33
3.3.1	Examples of Coding	34
3.3.1.0.4	Single Error Correcting Code(SEC)	34
3.3.1.0.5	Forward Error Correction	34
3.3.1.0.6	Combination	34
3.3.1.0.7	Power Efficiency of Coding	34
3.3.2	Other Options From Electrical NoC	35
3.4	Chapter Summary	35
4	Fault-Tolerant Photonic On-chip Network Architecture	36
4.1	Introduction	36
4.2	System Architecture	36
4.2.1	Network Architecture	37

4.2.2	Node Architecture	40
4.2.3	Electronic Router Architecture	41
4.2.4	Arbiter Architecture	42
4.2.5	FT-PHENIC Routing Algorithm	43
4.3	FTTDOR: Fault-tolerant Non-Blocking Photonic Switch	46
4.3.1	Building Blocks	48
4.3.1.1	Waveguides	48
4.3.1.2	Micro Rings Resonators	49
4.3.2	Micro-Ring Configuration	49
4.3.3	Optical Power Loss Evaluation	50
4.4	Light-Weight Electronic Controller Architecture	54
4.5	FTTDOR Evaluation	56
4.5.1	Area Evaluation	56
4.5.2	Loss and Bit Error Rate	57
4.6	Chapter Summary	59
5	Fault-Tolerant Path Configuration Algorithm	61
5.1	Introduction	61
5.2	Fault-Tolerant Path Configuration Algorithm	61
5.2.1	Path Configuration	62
5.2.1.1	Path Setup	62
5.2.1.2	Blocked Paths	64
5.2.1.3	Faulty Switch	65
5.2.1.4	ACK	66
5.2.1.5	Payload Transmission	66
5.2.1.6	Teardown	67
5.2.2	Advantages of the Proposed Path Configuration Algorithm	68
5.3	Evaluation	69
5.3.1	Methodology and Assumptions	69
5.3.2	Complexity Evaluation	70
5.3.3	Latency Evaluation	73
5.3.3.1	Latency at Different Packet Injection Rates	73

5.3.3.2	Latency at Different Fault Injection Rates	73
5.3.4	Bandwidth Evaluation	75
5.3.4.1	Bandwidth at Different Packet Injection Rates	75
5.3.4.2	Bandwidth at Different Fault Injection Rates	75
5.3.5	Energy Evaluation	76
5.3.5.1	Energy Breakdown	76
5.3.5.2	Total Energy and Energy Efficiency	78
5.4	Chapter Summary	80
6	Strain-Aware Routing Algorithm	81
6.1	Introduction	81
6.2	Power Estimation	81
6.2.1	Power Estimation Calculation	83
6.3	LASA Algorithm	86
6.3.1	Routing	87
6.3.2	Strain	92
6.4	Evaluation	94
6.4.1	Methodology	94
6.4.1.1	Power Estimate Accuracy Methodology	94
6.4.1.2	Algorithm Evaluation Methodology	96
6.4.2	Power Estimate Evaluation	97
6.4.3	LASA Routing Algorithm Evaluation	98
6.4.3.1	Performance Evaluation	98
6.4.3.2	Energy Evaluation	99
6.4.3.3	SAFT-PHENIC Fault-Tolerance Evaluation	100
6.4.4	Chapter Summary	102
7	Conclusion and Discussion	105
7.1	Contributions	105
7.2	Results Summary	106
7.3	Discussion	107

List of Figures

1.1	SoC design complexity trends.	2
1.2	Power consumption trends for communication-centric SoC design.	3
1.3	Power consumption trends for computation-centric SoC design.	4
1.4	SoC architecture: (a) Shared-bus (b) Point-2-Point (c) NoC	5
2.1	Anatomy of EA-PNoC architecture.	12
2.2	Anatomy of WR-PNoC architecture.	15
2.3	Functional diagram of an optical communication.	16
2.4	Cross-section of a waveguide.	17
2.5	Micro-ring modulator	18
2.6	Circuit model of germanium detector with inductive gain peak.	19
2.7	Micrographs of a fabricated microring resonator.	20
3.1	WDM fault tolerance example.	30
3.2	Example of a thermally tuned MR	31
3.3	Example of thermal effects, voltage effects and athermal rings	32
4.1	FT-PHENIC system architecture.	38
4.2	FT-PHENIC architecture. (a) Network (b) Optical switch (c)Node architecture	39
4.3	Architecture of a single node.	40
4.4	Control router architecture.	41
4.5	Fault-tolerant arbiter architecture	42
4.6	Comparison of 3 different 5-port switches	45
4.7	Fault-tolerant 5x5 optical router.	48

4.8	Fault-tolerant 4x4 optical router.	49
4.9	Fault-tolerant 3x3 optical router.	50
4.10	Showing an example of rerouting within a router with a fault at MR 9.	51
4.11	Photonic switch components	51
4.12	Photonic switch building blocks instantiation.	52
4.13	PHENIC's light-weight electronic router.	56
4.14	Signal, noise, and SNR average values for FFT simulation	59
4.15	Worst case SNRs for each network for FFT	60
4.16	Worst case BERs for each network for FFT	60
5.1	Successful path-setup.	62
5.2	Failed path-setup.	64
5.3	Faulty path-setup.	65
5.4	ACK phase.	66
5.5	Payload transmission.	67
5.6	Tear-down phase.	68
5.7	Overall latency comparison results of all systems under random uniform traffic for various packet injection rates.	72
5.8	Latency results of each system as faults are introduced.	74
5.9	Bandwidth comparison results under random uniform traffic.	75
5.10	Bandwidth comparison results as faults are introduced.	76
5.11	Total energy breakdown comparison under random uniform traffic near-saturation:(a) 16-core systems, (b) 64-core systems, (c) 256-core systems.	77
5.12	Total energy and energy efficiency comparison results under random uniform traffic near-saturation with (a) 0% and (b) 4% faulty MRs.	78
5.13	Total energy and energy efficiency comparison results under random uniform traffic near-saturation with (a) 10% and (b) 30% faulty MRs.	79
6.1	FT-PHENIC system architecture. (a) 4x4 mesh-based system, (b) 5x5 non-blocking photonic switch, (c) Unified tile including PE, NI and control modules.	82

6.2	Arbiter architecture.	84
6.3	SAFT-PHENIC node architecture.	86
6.4	Flowchart of the LASA algorithm	89
6.5	Example cases for strain values and how the two different routing algorithms react	90
6.6	Overall latency results with various packet injection rates	98
6.7	Bandwidth comparison results.	99
6.8	Total energy and energy efficiency comparison results near-saturation.	100
6.9	Peak (a)Optical and (b) Electrical energy of the most active node in the different networks.	101
6.10	Affect on bandwidth as faulty MRs are introduced to (a)4x4 (b)8x8 and (c)16x16 Networks.	103

List of Tables

2.1	Overview of fault causes and effects	22
4.1	Microring configuration for normal data transmission.	50
4.2	Microring backup configuration for data transmission.	53
4.3	Insertion loss parameters.	53
4.4	Comparison between 5×5 optical routers.	54
4.5	Power loss comparison.	55
4.6	Area parameters [1]	57
5.1	Configuration parameters.	70
5.2	Photonic communication network energy parameters [2]	70
5.3	Ring requirement and static power consumption results for 64-core systems.	71
5.4	Ring requirement and static power consumption comparison results for 256-core systems.	71
6.1	Example simulation energy values for a 4x4 network(J)	95
6.2	Example estimated energy values for a 4x4 network(J)	95
6.3	Example of error calculation(%)	95
6.4	Configuration parameters.	96
6.5	Photonic communication network energy parameters [2]	96
6.6	4x4 mesh accuracy results	97
6.7	8x8 mesh accuracy results	97
6.8	16x16 mesh accuracy results	97
6.9	Evaluation results summary under uniform random traffic.	104

List of Abbreviation

3D-IC:	Three dimensional Integrated Circuit
3D-NoC:	Three dimensional Network-on-Chip
ACK:	Acknowledgment
BER:	Bit Error Rate
DB:	Detector Bank
DPE:	Data Processing Element
DOR:	Dimension Order Routing
DWDM:	Dense Wavelength Division Multiplexing
ESD	Electrostatic Discharge
EA-PNoC:	Electro Assisted PNoC
E-NoC:	Electronic Network-on-Chip
ECN:	Electronic Control Network
EOR:	Electro-Optic Router
FCA:	Free Carrier Absorption
FFT:	Fast Fourier Transform
ITRS:	International Technology Road-map for Semiconductors
MB:	Modulator Bank
MR:	Micro-Ring Resonator
MRCT:	Micro-Ring Configuration Table
BUMRCT:	Backup Micro-ring Configuration Table
MRs:	Micro-Ring Resonators
MRST:	Micro-Ring State Table
MPSoC:	Multiprocessor Systems-on-Chip
MWSR:	Multiple Write Single Read

NBPS:	Non-Blocking Photonic Switch
NI:	Network Interface
P2P:	Point-to-Point
PBP:	Path Blocked Packet
PCN:	Photonic Communication Network
PE:	Processing Element
PIC:	Photonic Integrated Circuit
P-NoC:	Photonic Network-on-Chip
PSCP:	Path Setup Control Packet
PSC:	Photonic Switch Controller
PV	Process Variation
SNR:	Signal to Noise Ratio
SoC:	System-on-Chip
SRMW:	Single Write Multiple Read
TIA	Trans-Impedance Amplifier
TV	Thermal Variation
WDM:	Wavelength Division Multiplexing
WR-PNoC:	Wavelength-Routed Photonic Network-on-Chip

Chapter 1

Introduction

1.1 Background

In the newest era of life, technology is ever present. It is no longer something used strictly by researchers, designers or gamers. It is in everyone's pocket everyday for personal activities and almost everyone uses technology at work. Technology has become a way of making life easier, and almost everyone enjoys that benefit. With the increase in demand for technology, there is an increase in demand for the quality of the technology. Consumers usually determine quality based on speed and reliability. This has led to reliable embedded systems becoming a more popular option for hardware developers. From such, the majority of applications marketed apply to: audio/visual, medical, robotics, aeronautics, defense, refrigerators, watches, and many others. This field is limitless in what other fields it can affect, and so improving this field will result in an improvement and the lives of all humans.

1.2 Current System Design

Current embedded chip design is shifting into the System-on-Chip (SoC) paradigm. SoCs are single chips which include several modules such as processors, memory, sensors or I/O peripherals [3, 4]. By integrating these components into a single chip, the SoC is able to complete tasks at faster rates, and consume less power. This becomes more true as transistor sizes continue to shrink and the interconnection

between chips becomes the largest distance that data has to travel. According to Moore's law [5], the number of gates that can fit on a chip doubles every 18 months. The most recent research has successfully created a gate with 1nm technology [6]. The shrinking of gate sizes increases the efficiency of each gate, but the shrinking can also allow additional gates to fit on a single die, thus allowing for more advanced chips to be made.

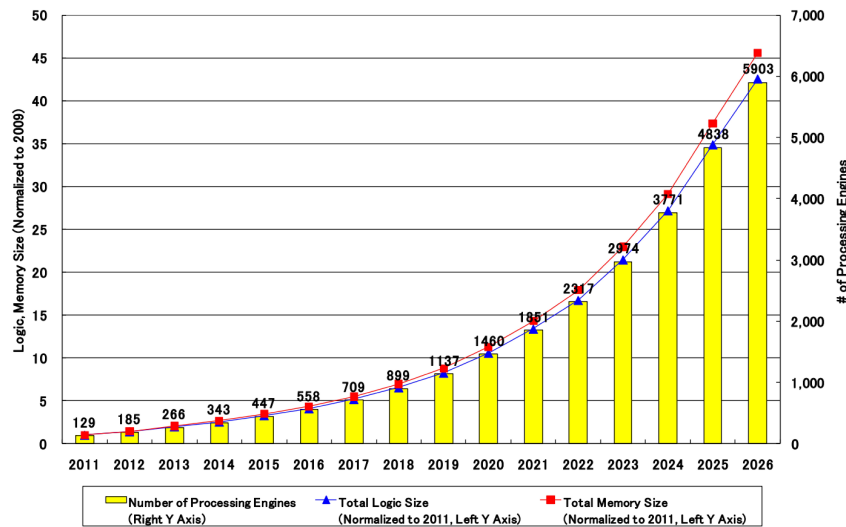


Figure 1.1: SOC design complexity trends [7].

Figure 1.1 shows the design complexity trends of SoCs according to the International Technology Road-Map for Semiconductors 2011 (ITRS) [7]. The ITRS predicts that the number of Processing Elements (PEs) that will fit on a single chip will be just shy of 6000 by the year 2026. This will lead to SoCs with more than 70 TFlops of processing performance [7].

Because the number of cores is increasing, there is a heightened demand for designing a more efficient communication system. There are several traits to consider when designing the new technology. Some of the main constraints are: power consumption; silicon area; design complexity; manufacturing yield; and scalability. The communication methods have started to become a large influence on the chip's overall performance and power consumption [8]. Interconnection networks can now account for more than 50% of the chip's dynamic power consumption; and, this is expected to increase [9]. Most experts expect the power consumption per DPE to

decrease, but this is going to be offset by the increase in the number of DPEs per chip, which increases the demand for effective results in critical chip packaging and cooling issues. The power trends are also assessed by the ITRS in Figs. 1.2 and 1.3

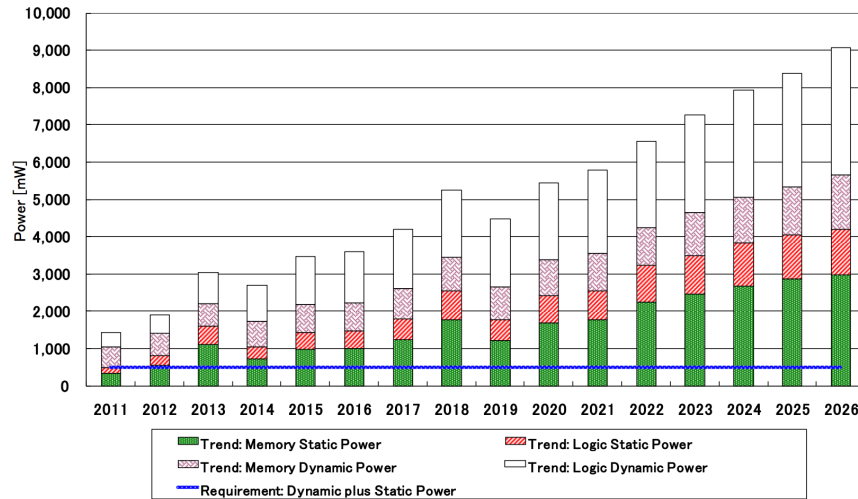


Figure 1.2: Power consumption trends for communication-centric SoC design [7].

The traditional, bus-based networks are weak when it comes to scalability, and parallelism, resulting in high latency and power consumption. Because of these negative factors, bus-based and point-to-point (P2P) networks have become unsuitable for future designs. Fortunately, a new system, called Network-on-Chip (NoC) has already been designed and tested to address these issues. Figure 1.4 shows the main differences between the three types of SoCs. The traditional bus is shown in Fig 1.4 (a). All devices are connected to a single bus device, which can then connect various devices together. A P2P network is shown in Fig. 1.4 (b). Figure 1.4 (b) clearly illustrates how each device is directly connected with the other devices that they could want to communicate with. This system takes a lot of hardware to connect every device to every other device, so links which are rarely used, or not used at all are omitted. This means that even if two components that previously did not communicate wanted to communicate for a future application, they are unable to do so. A mesh-based NoC is shown in Fig. 1.4 (c). Routers are placed and connected to each other in a grid-like pattern. Each router is also connected to a PE. Some

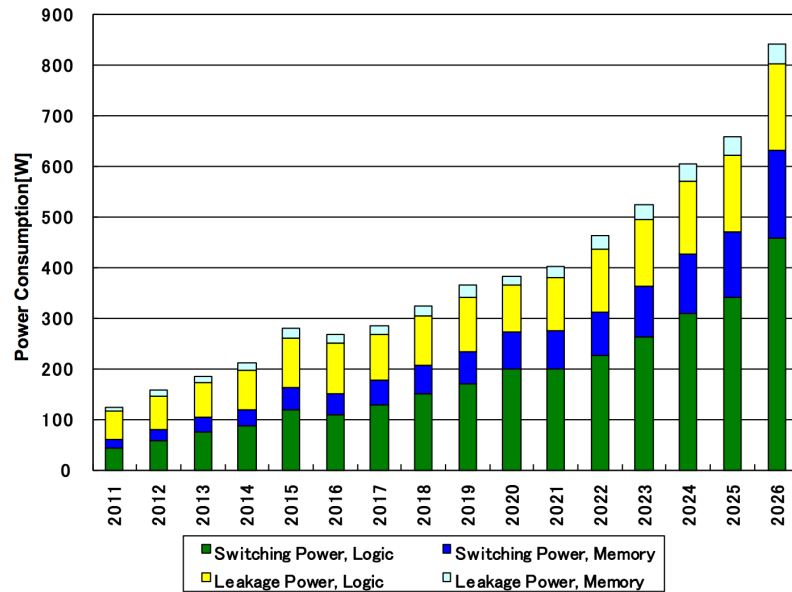


Figure 1.3: Power consumption trends for computation-centric SoC design [7].

NoC architectures have multiple PEs per router.

1.3 Network On Chips

Network-on-Chip [10–14] has arisen as a solution to the problems that face bus-based and P2P SoCs. It is based on a scalable network that can expand as much as you can fit onto a chip. At the simplest level, it is just a grid of routers that are connected together.

Traditional bus-based SoCs functioned well for a while, but they had limitations. They had very little scalability, and their bandwidth was quite limited [15]. Kumar [15] proposed a system which would reuse components and architectures, in order to simplify the design, and make each system, much more scalable. This would be effective on billion-transistor chips. Thus, the Network-on-Chip was born. Kumar also justified his design by stating “The design costs can be justified by increasing the implementation volumes and it is likely that the billion-transistor chips are not designed for single product instances or single applications. The design methodology must therefore support product family management. Tolerance of incomplete specifications, management of configurations and modifications, support

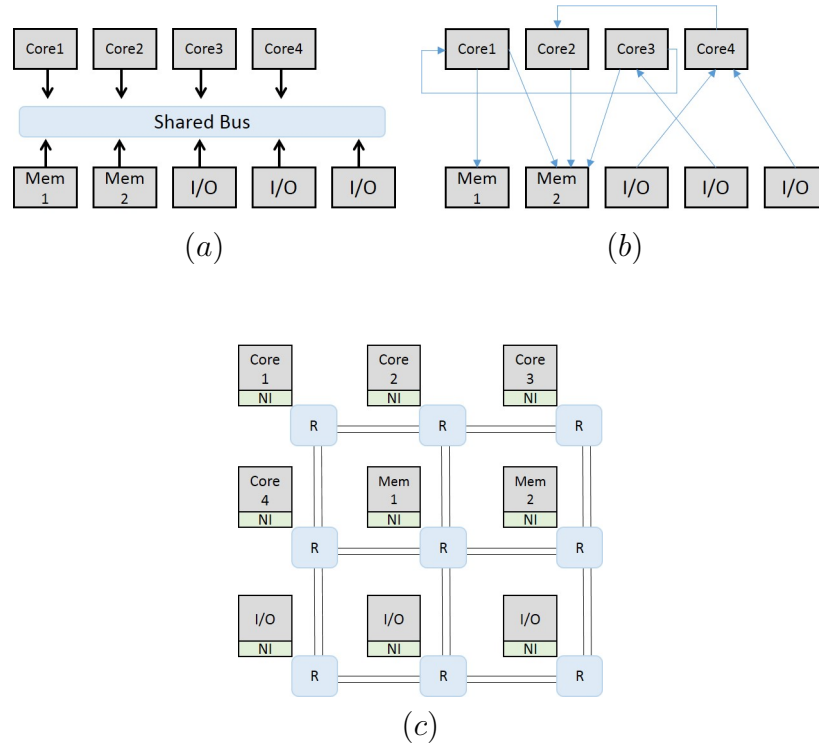


Figure 1.4: SoC architecture: (a) Shared-bus (b) Point-2-Point (c) NoC

for multiple languages and methods, and capability to handle different abstraction levels simultaneously are desirable characteristics.” This new NoC technology would soon be crucial for the many-core era. Communications occur in a hop-by-hop basis that allows for any router to communicate with any other router. Some authors have created different shapes, such as trees, rings, or toruses. But, the main concept remains the same: connect routers to other routers to allow for scalability and parallelism.

This solved the interconnect bottleneck and bandwidth problems of traditional bus-based multi-core Systems-on-Chip [3, 4]. At around the same time as the development of NoCs, three-dimensional integrated circuits (3D-ICs) were also being adopted [16, 17]. These 3D-ICs have gotten the focus of many designers as a possible solution for eliminating the current bottleneck issue. Combining these two technologies was inevitable, and promised large throughput benefits. Many different three-dimensional network-on-chips (3D-NoCs) have been designed and developed [18–20].

Most NoCs in development do not address fault issues within switches. Others do not even address the issue of the throughput limitation of electrical signals. 3D-

OASIS [21] and the original mesh [15] architecture both resort to using electrical interconnects, which have been proven to have lower throughput than their optical alternative. Additionally, optical data requires less power per bit [22]. This new technology is not 100% beneficial, as can be seen with the results of certain optical NoCs.

1.4 Photonic Interconnects

With recent demonstrations of feasibility in fabricating photonic components for on-chip communication, research focus is now on on-chip photonic communication for future high-performance many-core processors. Photonic Networks-on-Chip (PNoCs) [23–27] offer several benefits over conventional electrical NoCs, such as high-bandwidth support, distance independent power consumption, lower latency, and improved performance-per-watt. When combined with Wavelength Division Multiplexing (WDM), multiple parallel optical streams of data can concurrently transfer through a single waveguide, while micro-ring resonators (MRs), which can be switched at speeds as high as 40 GHz, and be used to realize wavelength-selective modulators and switches [28]. While a single-layer configuration can provide low-loss waveguides and high-performance photonic devices, it suffers from limited integration density due to waveguide crossings and limited real estate. A way to go beyond this limitation is to monolithically stack multiple photonic layers above Si as multi-layered electrical interconnections are realized in modern electronic circuits [29–31].

The key to the power and performance benefits of a PNoC come from the fact that once a path is active, the data can be transmitted end-to-end without any additional hops being required. This means that the energy used for the transmission of the payload is independent of the number of hops because it does not require buffering, repeating, or regenerating. In a typical NoC, each message gets buffered, read, and transmitted at each node, which requires a lot of energy because the payload is quite large. Optical routers need to be switched on when the transmission begins, and then switched off when it ends, so they have very little power wasted due to switching. This also allows for the transmission of ultra-high bandwidth

messages. The combined advantages of bandwidth density and power efficiency make photonic interconnects one of the most likely options as a replacement for electronic interconnects.

Another benefit of using photonic interconnects is that they allow for voltage isolation between different sections of a chip. This is because photodetectors count photons, without measuring a potential difference, and isolate the voltage from the modulator [32].

Photonic interconnects are not the only solution that has been proposed to solve the problems set forth by electronic interconnects. On-Chip Wireless Interconnects [33,34] were proposed especially for the use of broadcast messaging. They are most commonly seen in hybrid networks, and would not replace electronic interconnects completely, but rather alleviate some of the traffic for them. This technology provides higher bandwidth, lower latency, less area overhead and reduced energy dissipation for communications; however, the NoC based wireless is unreliable because the probability of error is eminent due to synchronization delays at the receiver.

1.5 Reliability Issues in Photonic Networks-on-Chip

The main components of a PNoC include a laser source, which generates phase-coherent and equally spaced wavelengths, waveguides, which are used as a transmission medium, and modulators and photodetectors, which convert electrical digital data to and from photonic signals [35]. A typical on-chip optical link uses an external laser as a light source. It is expected that the laser source could produce up to 64 wavelengths per waveguide for a WDM network [36].

The photonic domain is immune to transient faults caused by radiation [37], but is still susceptible to process variation (PV) and thermal variations (TV) as well as aging. The aging typically occurs faster in active components as well as with elements that have high TV [38]. In the optical domain, faults can occur in MRs, waveguides, routers, etc. Active components, such as photodetectors, have higher failure rates than passive components, e.g. waveguides [38]. A single MR failure can

cause messages to be misdelivered or lost, which results in bandwidth loss or even complete system failure. Together, fabrication-induced PV and TV effects present enormous performance and reliability concerns. TV causes a microring to respond to a wavelength that is different from that intended. This can take the form of a pass-band shift in the MRs. When an MR heats up, it expands, changing its radius, and therefore shifting the wavelengths which it uses to the right (a larger wavelength) [39]. As reported in [28], a change of as little as 1°C can shift the resonance wavelength of a microring by as much as 0.1nm. This is not permanent and will return when the temperature returns to normal; therefore, a system's temperature must be kept at a reasonable level in order for the MRs to resonate correctly. This is challenging, especially in large and complex computing systems, which use thousands of these components. The trimming technique [40] is generally used to dynamically modify the resonance frequency of a microring to overcome both thermal drift and fabrication inaccuracy. This technique can be accomplished by dynamically increasing the current in the $n+$ region or by heating the ring [40–42].

PV is the variation of critical physical dimensions, e.g. thickness of wafer, width of waveguides, which can also affect the resonant wavelengths of MRs. This means that not all fabricated MRs can be used due to PV. As a result, network nodes that do not have all MRs working would lose some or all of wavelengths/bandwidth in communication [43]. To solve this problem, Xu et al. [44] proposed a method of flexible wavelength assignment for Single-write Multiple-read or Multiple-write Single-read networks. Because the networks are already built with excess detectors or modulators for each message, the node with the excess components can compensate and rematch to the components which have been affected by PV.

1.6 Thesis Objectives and Contributions

The Adaptive Systems Laboratory at the University of Aizu has already made great strides in the performance aspect of EA-PNoCs. We have created a novel energy-efficient and high-throughput many-core hybrid Silicon-Photonic Network-on-Chip architecture (PHENIC). The PHENIC system has a Non-Blocking Pho-

tonic Switch (NBPS) and equipped with contention-aware path configuration algorithm [45]. This system mainly focused on the performance and power aspects, but was still vulnerable to many kinds of faults.

To solve issues created by process variation, we propose a fault-tolerant switch and path configuration algorithm for the PHENIC system. Changing the path configuration algorithm requires a few significant changes to the architecture, so we call the new system Fault-tolerant PHENIC (FT-PHENIC).

To further improve the FT-PHENIC system, the system needed to account for thermal variation. To help alleviate thermal variations in the chip, and avoid the hotspots, we created a Strain-Aware routing algorithm for the FT-PHENIC system. To help divide the non-strain-aware and the strain-aware systems during testing, we refer to it as SAFT-PHENIC. The overall architecture is the same.

The main contributions of this research are as follows:

- A new Fault-tolerant Photonic Switch (FTTDOR) [30], which is capable of handling faulty MRs. It will have redundant MRs at key locations, which are critical for creating the backup paths. This will allow for some fault tolerance that is independent from the routing algorithm.
- A fault-aware path configuration algorithm [46] that aims to read faulty statuses of MRs, and adjust the control signals. This algorithm utilizes an additional MR configuration table to keep track of backup paths within the optical router. The proposed algorithm would allow for less packets to require rerouting, and increase the overall reliability of a network, contrary to the standard path configuration algorithms.
- A simple scheme for estimating the power consumed by the optical layer. This technique is fast enough to be used for routing algorithms, and is not as accurate as simulation or advanced calculation techniques. This power estimation technique is based off of the types of traffic that occur at each node.
- A strain aware routing algorithm (LASA). We create a parameter called strain, which is based off of the faulty MRs in a router, and the power that the router

consumes. The routing algorithm uses the power estimates to give the network a comparative index for the temperature of each node.

- A detailed performance evaluation where we highlight the efficiency of the proposed system and the reliability gain when compared to previously proposed PNoC systems [47]. We also assure that the costs of such systems does not outweigh the benefits, and that the systems can still be considered to have high-performance.

1.7 Thesis Outline

The rest of the thesis is organized as follows:

- In Chapter 2, we will cover some progress that PNoCs have made up till now, and the causes of failures inside PNoCs.
- Chapter 3 presents some of the related works that deal with PNoC Faults.
- Chapter 4 introduces our proposed fault-tolerant FT-PHENIC architecture and optical router (FTTDOR).
- Chapter 5 covers the fault-tolerant path configuration algorithm (FTPP). This algorithm handles two MR configuration tables.
- Chapter 6 Covers our new strain aware routing algorithm (LASA).
- Finally in Chapter 7, we finish this thesis with the conclusions, as well as some future steps to optimize it even more.

Chapter 2

Background

In this chapter, photonic on-chip networks are introduced together with their principal components, and some of the fault models. The goal of this chapter is to explain how PNoCs work. The benefits and drawbacks of each point will be highlighted.

2.1 Photonic NoCs

In this section, we will describe the two main approaches for PNoCs. The first is circuit switching, which is used in EA-PNoCs. The second is wavelength-selective which is commonly used in WR-PNoCs. Some other unique schemes have been proposed, but are not nearly as heavily researched as the other two at this time. One example is Time- Division-Multiplexing [48].

2.1.1 Circuit Switching

EA-PNoCs utilize path setup techniques to reserve a path, and then send a message across the path. This means that once a path is secured, a message of any size can be sent and large ones would require almost no extra time to reach its destination when compared to a small one. This means that the main problem with this type of network is setting up the path in an efficient manner. Currently, the path setup process has additional power and latency overheads. The standard EA-PNoC message process is to first have the source node send a path-setup packet

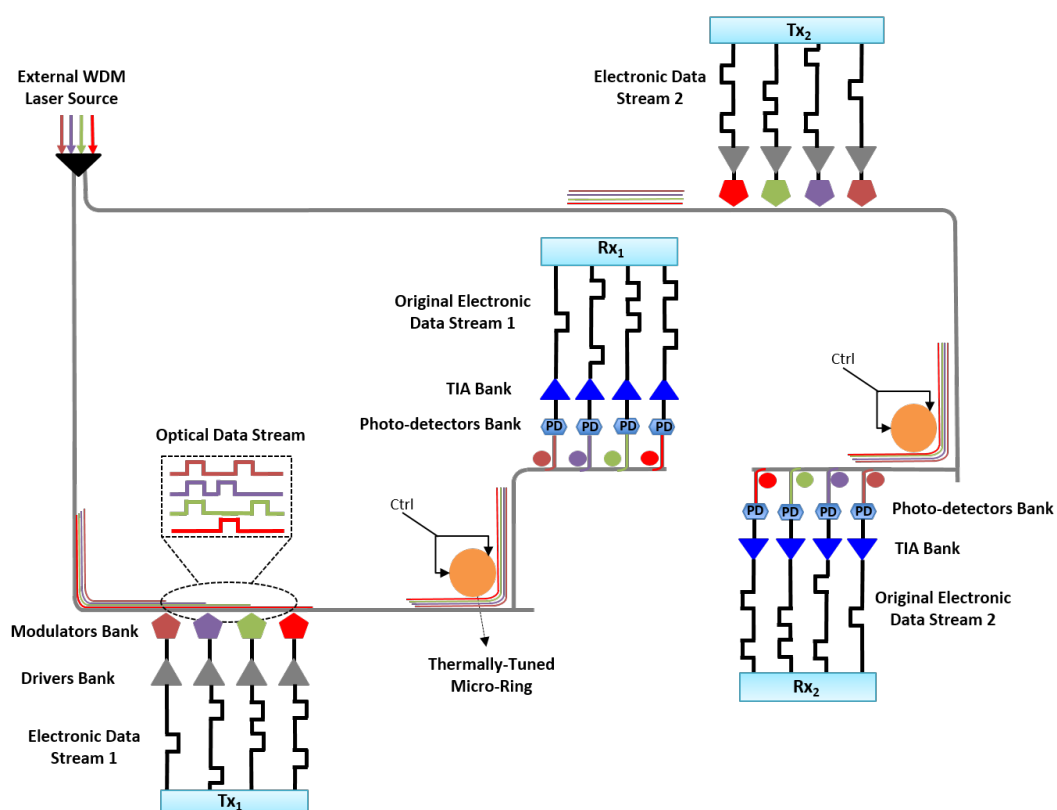


Figure 2.1: Anatomy of EA-PNoC architecture. The two communications are using the same set of wavelengths in a circuit switching scheme [45].

to the destination node via the electrical layer. This packet tells each node that it visits to reserve the corresponding MRs, setting up the path. This packet contains information about the source and destination node, as well as some control information, like packet type. When the destination receives a path-setup packet with a matching destination address, the destination node will send out an ACK packet to the source. This packet will switch all of the reserved MRs on and let the source node know that the network is ready. When the source receives the ACK packet, it begins sending the payload via the optical network. When the transmission is done, a tear-down packet is sent from the source node to release the MRs, and let them be available for use for other transmissions.

Figure 2.1 shows two communications between two source-destination pairs. Both communications can use all of the available wavelengths because the path is completely reserved. This means that EA-PNoCs favor large message sizes because of the latency overhead, which contrasts with WR-PNoCs which have low latency and favor small message sizes. This means that EA-PNoCs are aimed at bandwidth-intensive applications.

2.1.2 Wavelength-Routed

Wavelength-routed (WR) architectures use set wavelengths for communications for each source-destination pair. This is achieved by using a combination of modulators, filters, waveguides and photodetectors. This type of network generally does not have switching MRs. Wavelength-routed networks start by selecting the appropriate wavelengths for the source-destination pair, and then sends it on to an optical bus. This bus then gets filtered at the receiving end, so that each node is responsible for its own wavelengths. This means that if a WR architecture has 128 wavelengths, and 64 nodes, then each node could use a maximum of 2 wavelengths. This allows the architectures to have a lower latency than electro-assisted architectures because they do not require any path setup. The optical transmission speeds are generally slower compared to EA-PNoCs which utilize a similar number of optical devices. To compensate for that, these networks tend to have very large optical areas, which are mostly consumed by incredibly large modulator banks, or they reduce the number

of nodes by assigning multiple PEs to a single node.

Figure 2.2 shows the architecture of a wavelength-routed network's router. As we said earlier, each source-destination pair has a specific wavelength or set of wavelengths. A majority of literatures in the PNoC field investigate this routing technique. The key to making this type of architecture successful is minimizing the number of photonic devices, while ensuring a contention-free network. There are many different forms of WR-PNoCs, which utilize different numbers of writers and receivers at each node. The five main types are: Source-based, Destination-Based, Single-write Multiple-Read, Multiple-Write Single Read, and Fully-Connected.

Source-based is when each node can read a single wavelength channel. Every other node has the ability to write to this channel. This means that for N nodes, the network has $N \times (N-1)$ modulators, N photodetectors, and N channels. Contention can only occur if two nodes want to send a message to the same node.

Destination based is when each node can write to a single wavelength. Every other node has the ability to read this channel. This means that the destinations have to selectively read the wavelength. This also results in a network with N nodes using $N \times (N-1)$ photodetectors, N modulators, and N channels.

Single-Write Multiple-read networks (SWMR) requires a single snake-like waveguide for each node. A single node can write to that waveguide, but every node can read from it. This means that it requires N waveguides, but unlike the destination-based routing, it can reuse wavelengths for different nodes, because they don't share the same waveguide.

Multiple Write Single-Read networks (MWSR) also requires a single snake-like waveguide for each node. Every node can write to that waveguide, but a single node can read from it. This means that it requires N waveguides, but unlike the source-based routing, it can reuse wavelengths for different nodes, because they don't share the same waveguide.

2.1.3 Photonic Communication

Figure 2.3 shows the flow of data through a typical optical communication. The electrical data is encoded into a message. The second step is to serialize the message.

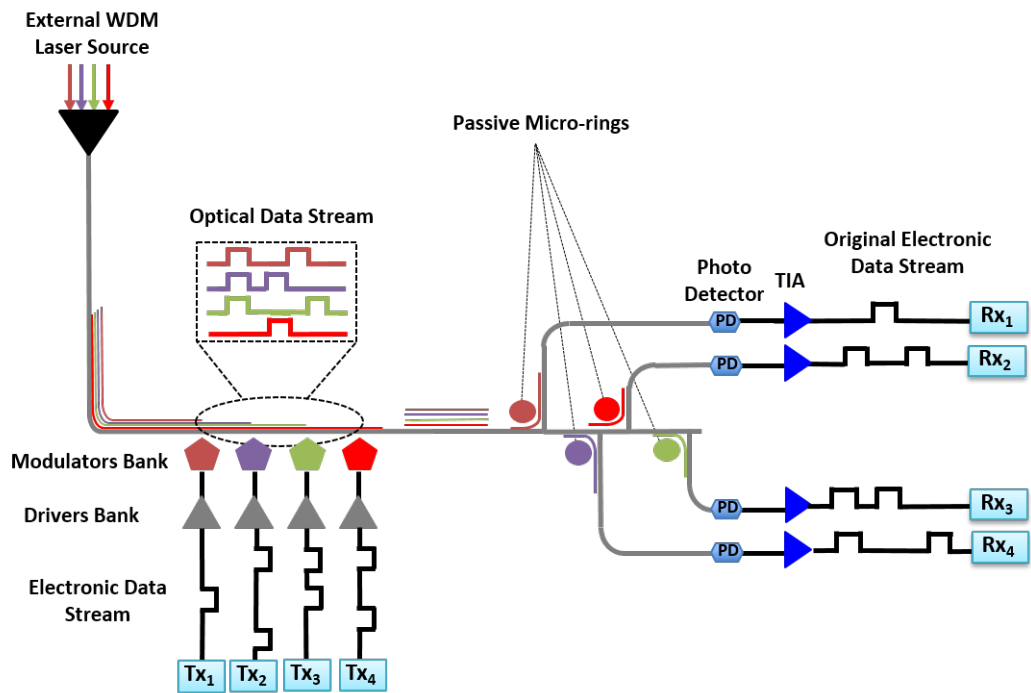


Figure 2.2: Anatomy of WR-PNoC architecture. The four communications are using a different set of wavelengths [45].

The serialization step's goal is to reduce the number of wires required at the output by combining multiple incoming data streams (i.e., wires). The overall data rate remains constant across the serialization step. Third, a driver circuit is required for each transmission wire to condition each signal with the appropriate peak-to-peak voltage levels and to supply an adequate amount of current to drive the optical modulator. Then, modulators translate the electrical signal into an optical signal, so it can be transmitted in the photonic network. The data then follows the preset route of the optical network, until it reaches the detectors at the destination node.

On the receiver side, similar steps are taken in reverse to undo the encodings. First, the photodetectors receive the incoming light signal. Each photodetector then converts a single stream of data into an electrical current. After the detection, the resulting current goes through a Trans-Impedance amplification step to convert the output of the photodetectors from a current-based signal to a voltage-based one. The deserialization occurs, which undoes the serialization, thus restoring the original data rate. Finally, the decoding step restores the original data signal from

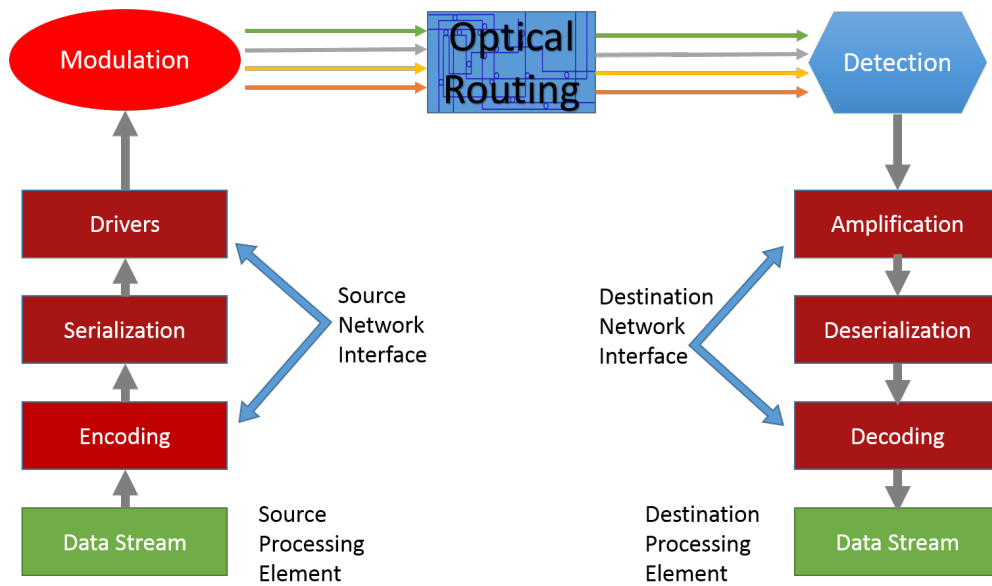


Figure 2.3: Functional diagram of an optical communication.

the transmitting node.

2.1.4 Photonic NoC Components

This section details each of the components that are used in on-chip photonic networks.

2.1.4.1 Laser

The first component in the life cycle of an optical message is the laser source. They emit light by stimulating a photon emitter and focusing it in such a way that it can give consistent light. The key parts are a pump, a gain medium, and in some cases a feedback mechanism [49]. Specifically for optical communication purposes, the main parameters for ascertaining a laser's viability are operating wavelengths, power efficiency, output power, signal stability, footprint area, and CMOS compatibility. Lasers have on-chip and off-chip varieties. Generally, the off-chip lasers are stronger and more efficient, but take up more area. Using an off-chip laser with quantum dot Si optical amplifiers can produce a wide band of accessible wavelengths with minimal source noise. This light stream can easily be modulated into an appro-

appropriate optical signal which is suitable for optical messages. By moving the source off-chip, the chip itself can have less complexity and because of the efficiency of off-chip sources, the whole combination can use minimal amounts of power.

2.1.4.2 Waveguide

The optical equivalent to a wire is the waveguide. Waveguides are used to carry the high-speed optical data streams across distances. Most nanophotonic chips use a large difference in the optical coefficients of the silicon waveguide and the silicon dioxide cladding to create a small optical mode count. This allows for on-chip photonic devices to be integrated very closely together.

Crystalline silicon photonic waveguides can transport optical data at terabits-per-second data rates across the entire chip using WDM technology [50, 51]. Like wires in the electrical domain, optical waveguides can have non-linear trajectories, requiring them to bend [50, 52]. Some designs require waveguides to cross across each other's paths [50, 53]. The cross section of a waveguide can be seen in figure 2.4.

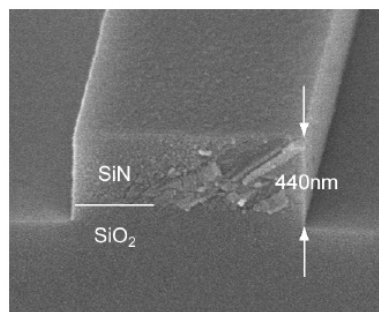


Figure 2.4: Cross-section of a waveguide [54].

2.1.4.3 Modulator

Modulators are the components that are responsible for taking an electrical signal and the output of a laser source, and creating an optical signal [50, 51, 55]. They are typically used in an array (the modulator array can also be called a modulator bank) form, with each one in the array being responsible for a specific wavelength.

The wavelength that a modulator uses is referred to as its resonant wavelength, and is determined by the round-trip phase difference that is caused by the MR. Optical modulators can be operated at frequencies as high as 110 GHz [56]. Figure.2.5 shows a modulator with its embedded heater needed to tune to a specific wavelength.

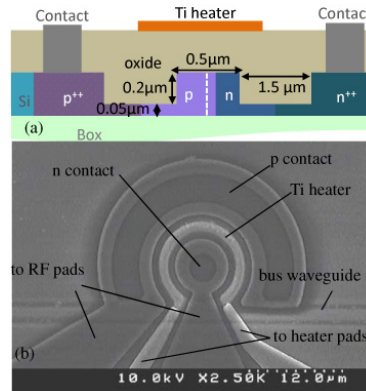


Figure 2.5: Micro-ring modulator. (a) Cross-section of the designed modulator with a local heater and (b) Top-view SEM image of the fully fabricated microring modulator. The ring radius is 5 m [57].

2.1.4.4 Photodetector

At the other end of a standard communication is the photodetector. This device absorbs the photons and converts the optical data stream into its electrical equivalent at the receiving end of a transmission. Similar to the modulators, photodetectors are responsible for a single wavelength, and thus are also seen as an array (also called a photodetector bank). To separate the signal, passive devices called filters are attached immediately before each photodetector. These filters only extract the specific wavelength that correlates to that photodetector, so that the other wavelengths can remain in the waveguide and be picked out by other filters down the line. A few different materials have been used to make these photodetectors [38, 58, 59]. A circuit model of a photodetector can be seen in figure 2.6.

2.1.4.5 Micro-Ring Resonator

The final component in optical networks are the Microring Resonators, commonly referred to as MRs. We have already mentioned how they are used as part of modu-

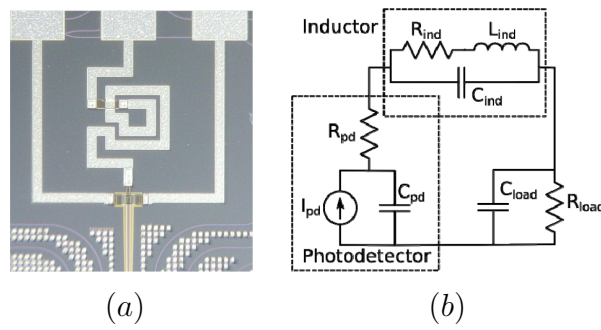


Figure 2.6: Circuit model of germanium detector with inductive gain peak.(a) Optical micrograph of the gain peaked photodetector using the 360 pH inductor. The inductor is approximately $100 \mu\text{m} \times 100 \mu\text{m}$ in size [60] and (b) Inductor used to peak the frequency response of the photodetector [60].

lators, or can be used as filters for detectors, but they are also used as the switches of optical circuits. MRs can be densely integrated thanks to their small size. If they are set up as passive components, we refer to them as filters. The wavelengths that correlate to the MRs are based on the MRs' Free Spectral Ranges (FSR) [50]. As the MR's radius gets smaller, the FSR will increase, and the amount of wavelength that will pass through the MR is reduced, and thus filters are made of very small MRs, which can only pass a single wavelength. The larger MRs will have a smaller FSR, and can act as broadband switches which can have many different wavelengths pass through it. These broadband switches are a vital component to WDM routing [36], and the smaller MRs are critical for wavelength selective routing [61]. MRs often require tuning because they are sensitive to small changes in temperature. This will be covered in detail in Chapter 6. Switching MRs can be placed at different positions. The most commonly seen position for a switching MR is between two parallel waveguides, which have data flowing in opposite directions. A micrograph of one such MR can be seen in figure 2.7. The other position where switching MRs can be seen are at waveguide crossings. This allows for a reduction of the necessary bends in optical switch architectures, and can often reduce the overall loss of the switch.

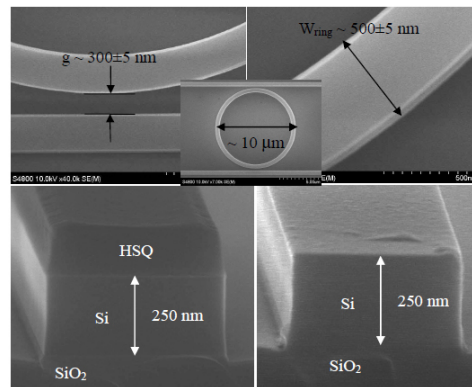


Figure 2.7: Scanning electron micrographs of a fabricated microring resonator and waveguide cross-section at two cleaved facets [62].

2.2 Fault Models

It is worth noting that the light is not sensitive to radiation or electromagnetic fields, the signals which control the optical network can be sensitive to them. The following is a list of actual possible causes that can contribute to the failure of an optical device.

2.2.1 Photonic NoC Signal Strength

Typical NoCs are defined by their power consumption, delay and throughput. PNoCs also have to consider the Signal-to-Noise Ratio at the receiving end. Because they do not buffer and retransmit, the signal gets weaker based off of how many hops it jumps. This does not significantly affect the power the network consumes, but it can lead to a higher sensitivity to noise.

2.2.2 Electrostatic Discharge

While the waveguides are not electrically conductive, the switches and photo-detectors are. This means that they are sensitive to high currents. One thing which can ruin an IC is electrostatic discharge(ESD). This is when a current enters in through the I/O pins of the control circuit, or it can be caused by an extremely strong magnetic field. This results in the aforementioned extreme current, and this extreme current causes severe damage to the silicon in the components. Possible

points of damage are the dielectric, the PN junctions, and any wiring connected to the controllers. Because of the scaling, the causing phenomena have become harder to control [63]. This can be prevented by proper packaging to the IC providing ESD protection at the pins.

2.2.3 Noise

This is one of the unique things that we categorize as a cause for a fault. The reason is because the noise can be caused simply by poorly matched wavelengths. Noise can also be caused by creating a path that is too long, or a path that crosses too many intersections. These paths tend to be caused by rerouting or non-minimalistic routing, but other factors can contribute and cause more noise. The most common factors are listed in the following subsections.

2.2.4 Aging

Over time, all silicon based ICs wear down. Some of the aging affects only the active components, because of their electrical subcomponents, while other aging affects the optical properties of the components.

Electromigration- This mainly affects the wires which control the ring resonators. It does not affect the waveguides in any way. It originally causes a delay in the wire, and can eventually lead to an open, or to a short to a nearby wire. It achieves this by thinning out the narrowest portion of the wire due to higher current density at the bottleneck. [64]

Laser Degradation- After the lasers have been on for several hundred hours, they start to show signs of degradation. This shows in the form of either missing wavelengths, which can cause a channel fault, or general weakening of the original laser signal. In each of these cases, it does not become a true problem until the signal falls to a level where the worst case scenario's Signal-to-Noise ratio is too weak to receive an understandable signal. [65]

Photodetector Degradation- Various studies have been done for different types of photodetectors showing that they degrade overtime, particularly from being ex-

Table 2.1: Overview of fault causes and effects

	Physical cause	Accel. By Process variation	Fault Class	Burstiness	Optical or Hybrid components
ESD	Build up of static energy which is quickly released	y	permanent logic fault	n	hybrid
Noise	Internal reflectance in waveguide	y	Intermittent logic fault	n	Optical
	Waveguide crossing	n	Intermittent logic fault	y	Optical
	Intrachannel Noise	n	Intermittent logic fault	y	Optical
Aging	Electron Migration	y	Intermittent --> Permanent delay and Logic Fault	n	hybrid
	Laser Degradation	y	Intermittent --> Permanent Logic Fault	n	hybrid
	Photodetector degradation	y	Intermittent --> Permanent delay and Logic Fault	n	hybrid
	MRR Degradation	y	Intermittent --> Permanent delay and Logic Fault	n	hybrid
	Waveguide Degradation	y	Intermittent --> Permanent Logic Fault	n	Optical
Temp. Variation	Variation in wear-out effects due to temp. differences	y	Intermittent and Permanent delay, delay, stuck-at, and Missrouting	Can be	Optical and Hybrid
	Performance variation due to temp. Differences	y	Intermittent delay & Missrouting	y	hybrid

posed to thermal conditions or UV light. It is reasonable to assume that no matter what material photo detectors are made out of, they all seem to be vulnerable to degradation due to thermal variation, which is present in all networks. [38] [59]

A lot of work has been done to combat the effects of aging. Some examples are Agarwal [66], Keane [64], and Kim [67]. These are mainly focused on the electrical side, but the fact that these do exist show the hope for a future where optical aging can be researched and prevented. Many parameters such as the wavelengths and laser strength can possibly be modified throughout the life of a chip to counteract the aging effects in a similar manner to what Mintarno does for Electrical networks [68].

2.2.5 Process Variability

This can affect both the active and inactive components of the optical network. The variability accounts for material impurities, doping concentrations, and size and geometries of structures [69]. One single dimple in a particular point in the coupling region of a ring resonator can greatly affect the coupling properties and thus cause problems for the switch, or maybe just the channel. A poor geometry can also cause a certain component to be more sensitive to aging or ESD. Obviously if a variation

gets bad enough, an entire link can be rendered useless. This would be considered an early permanent fault, and should be detected before a device is released. The impurities in a waveguide can cause such a block, or cause there to be a change in the reflectivity of the material, and that causes a higher amount of insertion loss, resulting in a lower signal-to-noise ratio. Other similar chains-of-events can occur from bad doping of the photodetectors. Minimizing this process variability can greatly increase the reliability of the system, even without implementing fancier and area or energy heavy redundancies. The unfortunate truth is that with recent advances in scaling, the variability continues to increase [70] [71].

2.2.6 Temperature Variation

For electrical components, temperature variation can cause changes in properties such as resistivity and cause more power consumption or delay, but in the optical domain, it is quite different. Ring resonators are tuned by heating up the ring, causing them to expand, which changes their passband wavelength. If the chip heats up to a point beyond the tuning, then certain channels just disappear as a whole. The increase in temperature also causes the photodetectors to degrade as mentioned in the previous section. These temperature variations also tend to speed up other forms of aging as well.

Table 2.1 summarizes the physical causes and their effects. Many of these will need to be researched further, and only time will tell exactly how reliable optical is with some other phenomena, but for now, this is a comprehensive list of all physical sources of failures within an optical network. We separated the pure optical from the hybrid components so that it can show exactly how resilient the photons and waveguides really are, when compared with wires, but no Optical Network-on-Chip is completely free of wires.

2.3 Chapter Summary

In this chapter, PNoCs were explained. This included the components as well as how they can be used together. We also reviewed how faults can occur in the

different components of the PNoC. We also discuss the typical metrics for measuring any On-chip network. The next chapter will cover some of the important works dealing with PNoC architectures and fault tolerance.

Chapter 3

Related Works

In this chapter, we will first discuss some famous PNoC networks. We will then have a section detailing some of the fault tolerance schemes which have been used in photonic on-chip networks. Finally, we will talk about some schemes that have only been implemented in the electrical domain, but should have no problems being directly ported to the optical domain.

3.1 Conventional PNoCs

With current NoCs, many researchers focus on the power, area and bandwidth. The problem of area changes from one design to the next, but seems to have some effective solutions, such as reducing the transistor size, or implementing 3D-ICs [17]. One solution for both the power and the bandwidth is to use optical technology. This is implemented in the form of Nanophotonic circuitry. A few people have designed their own Photonic NoCs(PNoCs), but only a few really define the PNoC category: Firefly [72], Corona [73], and Optical Mesh(OMesh) [74].

The first one we will mention is OMesh. This is a typical conversion of the original mesh, and it has been converted to use optical interconnects. Another common type of 2D-NoC is Torus, which is a mesh where the edges connect. In the network designed by Columbia University [74], a path is first set up in the electrical domain, and then the entire message is transmitted. The packet goes through Optical routers, much like an electrical NoC. The basic switching element

is called a Microring Resonator (MR), and they can be set up in special patterns to achieve multi-port optical routers. The wires are replaced by waveguides. The greatest difference involves the network interfaces, where the chip must include a laser with modulators to generate a signal, and photodetectors to read the signal. Because of the nature of photonic routing, messages can't be buffered in the optical domain, and thus can only be converted into electronic data once. This limitation is eclipsed by the great improvements in both power and bandwidth.

The next major PNoC is Corona [73]. Corona is a network which involves the snaking of a waveguide around to all of the tiles. Each tile has optical switches, detectors, and modulators. Each tile also has either multiple cores or some cache. One major benefit of Corona is that it is completely optical, and involves no electrical network behind the scenes. This does more closely resemble a bus than a NoC, but it is still important nevertheless.

The final related network is really a Hybrid NoC. Firefly [72] has an optical network which communicates between clusters, making it similar to Corona. Each cluster has multiple dies, which communicate with each other in the electrical domain. If a message was only intended to travel a short distance, then it would only use the electrical network. However, for inter-cluster communication, the network uses the optical domain. This gives the optical benefits of speed and distance for the nodes which are far away, but keeps the latency benefits of the electrical domain for the flits which only need to travel a short distance.

Each one of these can utilize a technique called Wavelength Division Multiplexing (WDM) [36]. This is a technique which uses only one waveguide to transmit several bits of data simultaneously through the use of multiple wavelengths. Because of the properties of light, these wavelengths only minimally interfere with each other, and can be filtered out one at a time when the message is converted back to the electrical domain. This helps with improving the performance of the network.

3.2 PNoC Fault-Tolerance

We found three main types of optical fault tolerance. The first is various methods of rerouting. The second involves techniques utilizing hardware redundancy. The final is tuning, which is specific to faults caused by thermal variation. Some additional methods mix these.

3.2.1 Rerouting

This comes as an option for mesh-based architectures because of the large amount of possible minimal paths. It requires some extra logic in the routing decision, but this is minimal compared to an extra interconnect at each location. One requirement is that the routing algorithm can not be deterministic. For it to truly support multiple faults, it must also support non-minimal routing, to avoid a non-reserved deadlock situation. It should also be noted that implementing fault tolerance on a deadlock free algorithm can negate that feature. This is not troublesome to optical networks as deadlock is a non-issue due to the fact that the E2E path is reserved before the transmission can start.

Ramesh proposed a method [75] of determining and using back up routes. Initially, a primary route is determined by the following algorithm an algorithm which determines the least cost path. This path is used unless there is a fault detected. This detection is handled by a cost function based around the load index.

The key to the fault tolerance is in the backup path. Ramesh proposes to use a set of probe packets. When the destination receives one of the probe packets, it then sends a PACK signal for each probe packet. If a packet is dropped due to faults, then a NACK signal is sent. This is determined by not receiving one of the sequential packets. The blocking probability of a path is the percentage of probe packets along that path that resulted in NACKs. Then the paths are sorted by blocking probability, and they are set up as backup paths in ascending BP order. Because this algorithm only activates when a problem is detected, it doesn't add any delay until faults occur, which may cause a standstill in a non-fault-tolerant system.

Loh breaks his algorithm [76] into a similar fashion to Ramesh. It has a Default Routing algorithm and a backup routing method. His two methods are called Logical Route and Adaptive Route. The Logical Route involves a few sets of dimension-order routing. The adaptive algorithm determines which of the deterministic routings to use.

This method checks for faults and congestion along the way, and if they can be detected, then it tries to switch to the other form of dimension-order routing. This is an attempt to shift from x to y when a problem is found in the x direction. This results in a routing algorithm which is minimal and adaptive, deadlock-free, and livelock-free.

3.2.1.0.1 Fault Regions This method of has each node keep track of the permanent faults of its neighbors. This then allows for the path-making decision to be educated based on a certain distance away. It can guarantee that no old permanent faults are going to cause problems with the transmission. One such an algorithm is proposed by Xingyun [77]. He proposed a quite interesting optical network. It is a form of torus which only allows data in two directions.

This allows for some unique fault tolerance ideas. While they may not be minimalistic routing, it will switch directions, go under the chip and come back from the top and reroute to avoid a bad crossing. This could possibly cause large amounts of insertion loss from routing around the network's length multiple times. This loss would translate to high power cost, and not yield any true benefits from converting to optical. This is still only monitoring its own outputs though.

3.2.1.0.2 Look Ahead Routing This type of routing is interesting to think about implementing in a nanophotonic setting. Look ahead routing is where a node has knowledge of its neighbors' faulty links, and possibly its neighbors' neighbors' links. With this data at hand, the routing can protect a path and guarantee its success. The only issue would be implementing one of the detection algorithms mentioned at the beginning of this section. Although it hasn't yet been implemented in a photonic chip, there is no known reason to prevent it from being translated over.

3.2.1.0.3 Buffering and Checking Dong Xiang's method [43] uses a Minus first routing algorithm as a basis. The author does not detail how to detect a faulty link, but once a faulty link is discovered, the algorithm then runs a misroute algorithm.

This method attempts to find all paths from the source to the destination from the problematic node, and then determines which one takes the least amount of time. This switch shows that only the links are optical, and the switches themselves are electrical. This also allows for the implementation of buffers, which allow for additional fault tolerance options which are detailed in Radetzki's paper [69].

3.2.2 Hardware Redundancy

The main concept of hardware redundancy is the use of additional hardware to compensate for a faulty piece of hardware. This can be commonly seen in ONoCs in the detector and modulator banks of WDM-Networks.

Various authors have detailed how WDM can be used as a fault tolerance tool [51, 78, 79]. The basic idea is that if a certain wavelength causes problems, either through noise or a manufacturing defect, and this problem can be detected, then certain wavelengths can be disabled and enabled. This is highly effective for modulator and photo-detector based faults. They focus on permanent and intermittent faults, because a transient fault would occur far too fast for a wavelength to be switched. The key idea is that if a network requires 60 wavelengths, then you should design it with a couple extra channels, say 4, then it could tolerate up to 4 faulty photodetectors or modulators. Supposing that a network uses wavelengths 1-60, and the photodetector associated with wavelength 44 is found to be faulty. Once it is disabled and replaced, then wavelengths 1-43 and 45-61 will be used. Thus, avoiding the faulty photodetector, and still maintaining 60 wavelengths (channels) being used. This is demonstrated on a small scale 4-wavelength system in figure 3.1. You can see that by disabling the faulty modulator and corresponding photodetector, and using ones that correlate to a different wavelength, it can still maintain the bandwidth of three channels.

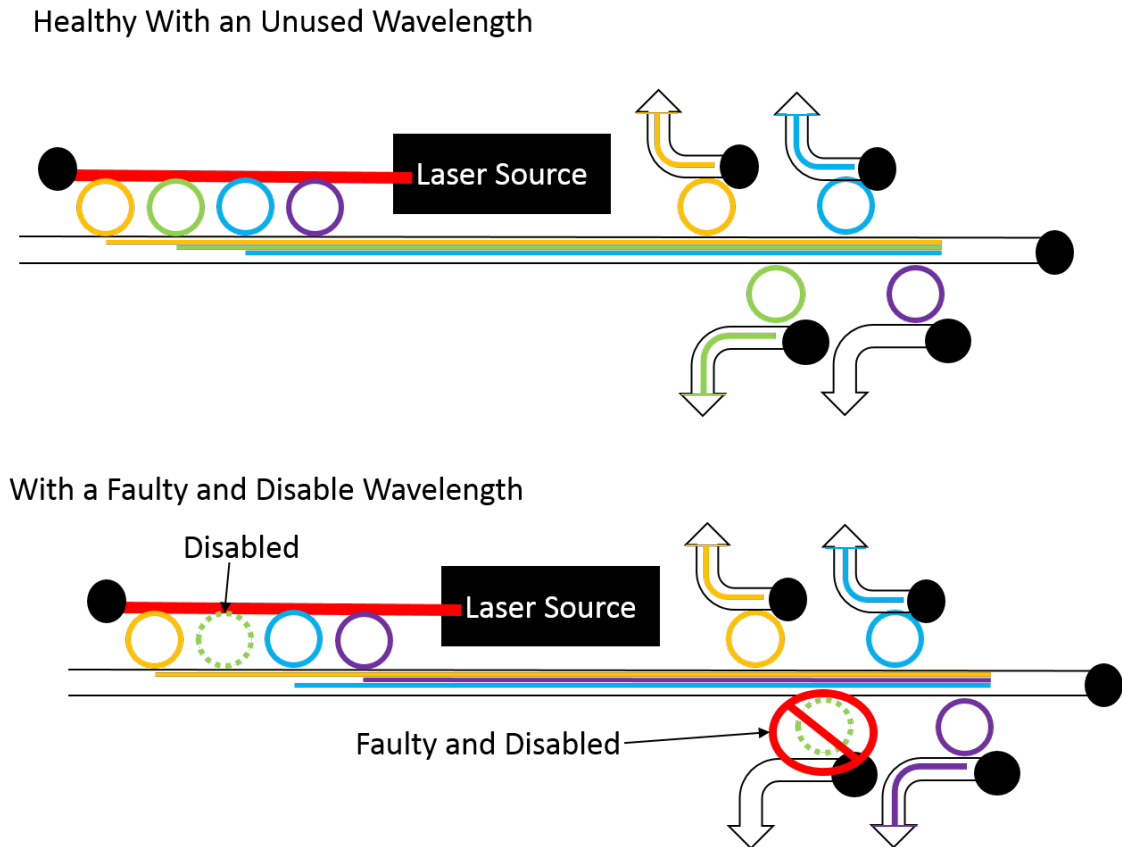


Figure 3.1: WDM fault tolerance example.

3.2.3 Tuning

Thermal variation (TV) effects present enormous performance and reliability concerns. TV causes a microring to respond to a different wavelength than intended. This can take the form of a passband shift in the MRs. When an MR heats up, the wavelengths which it uses shift to the right (a larger wavelength, or a red shift) [39]. As reported in [28], a change of as little as 1°C can shift the resonance wavelength of a microring by as much as 0.1nm . This effect can be seen in Fig. 3.3 (A), because this figure demonstrates intentionally heating up a ring. This may have serious performance or power costs on the chip [80]. The effect is not permanent and will return when the temperature returns to normal; therefore, a system's temperature must be kept at a reasonable level in order for the MRs to resonate correctly. This is challenging, especially in large and complex computing systems, which use thousands of these components. The trimming technique [40]

is generally used to dynamically modify the resonance frequency of a microring to overcome both thermal drift and fabrication inaccuracy. This technique can be accomplished by dynamically increasing the current in the $n+$ region, this effect can be seen in Fig. 3.3 (B), or by heating the ring [40–42], which can be seen in Fig. 3.3 (A). Additionally, some architectural approaches are effective, and can be combined with the aforementioned technological improvements.

Tuning [40, 81] was a solution which was mentioned in the introduction, and appears to be a promising answer to the problem. This is when a MR's temperature is controlled by outside means. This can be done by heating it up externally, or by running excess current through the MR, via a device similar to one seen in Fig. 3.2. In either method, this is only useful to heat up the device, and cooling it down requires other external techniques.

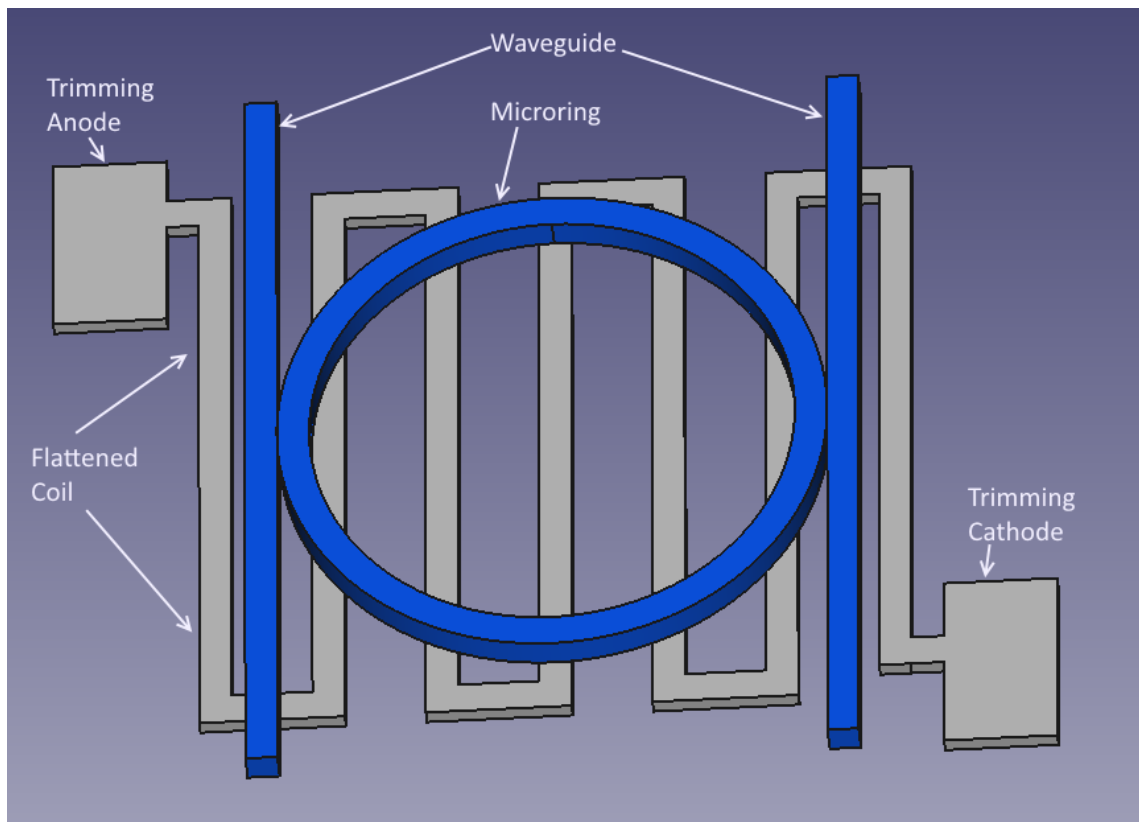


Figure 3.2: Example of a thermally tuned MR

The three main solutions for tuning can be seen in Fig. 3.3. (A) shows the standard thermal tuning. This has the same effect as a ring heating up because it

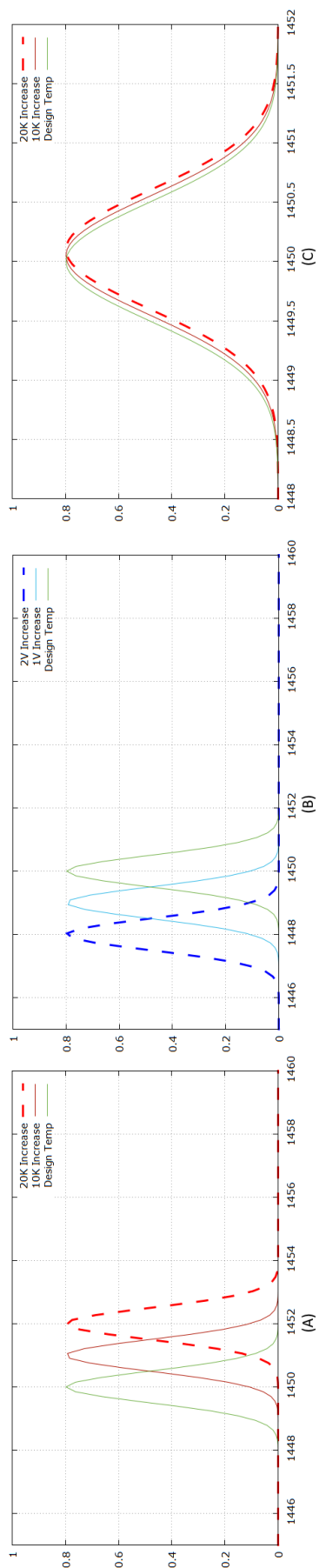


Figure 3.3: Example of thermal effects, voltage effects and athermal rings

is a method that heats up the ring. The initial design parameter is given in green. As the rings heat up, their passband shifts to the right. This means that the ring that was designed for a wavelength of 1450, will pass less than 10% of the signal when heated up 10° C, which is a large amount of insertion loss. (B) is Voltage Tuning [82]. An increase in voltage actually shifts the wavelengths to the left. This works against the heat. The downside to this is that it costs more power, and can heat up the ring because of the higher voltage used. This makes it require even more voltage, thus heating the ring up even more, and the cycle goes on and on. (C) shows the case of Athermal MRs [83]. The graph has been zoomed in to show the significance. The rings are made with a special material that is not affected by thermal variation (or at least the effect is greatly reduced). This requires special materials and is currently difficult to fabricate. Additionally, if the heat does get out of control enough to affect the MR, then it will take more effort to bring it back to the anticipated wavelength.

3.3 Other Usable Fault-Tolerance Schemes

A lot of other schemes can be used with PNoCs. Two common categories are routing algorithms and Error Correcting Codes(ECCs).

For optical signals, it seems like the only valid in operation testing for transient faults passed on from the electrical signals is information redundancy. Information redundancy can take the form of parity bits or error correcting code(ECC). These have been used commonly in NoCs, and can be adapted for PNoCs. This will be detailed later on. This is also effective in preventing some intermittent faults. In [84], the authors compare various ECCs schemes on a single optical link. After testing various schemes with different amounts of coding bits, the author found that the most efficient scheme was using a 64 bit SECDED code. It gave significant fault tolerance while only having small drawbacks to the system performance.

3.3.1 Examples of Coding

Information redundancy at the link level exists almost exclusively as coding techniques. They can either be codes to detect errors, or codes that fix the errors. We will detail a few coding techniques which have been selectively approved to be used for optical. One reason that this is a good option for optical is that usually the biggest drawback for this is that it increase the message width, but a typical optical network performs better than an electrical network as the message size increases. This is because most of the overhead is based around path setup, and not the transmission itself.

3.3.1.0.4 Single Error Correcting Code(SEC) The most well-known SEC code is Hamming Code [85]. A set of SEC codes can be combined to protect sections of the Code. This achieves better fault tolerances, but still only protects the message from one error per section. This helps protect both the links, and the switches with one code.

3.3.1.0.5 Forward Error Correction This is essentially a modified form of Hamming code, which is capable of detecting and correcting a certain number of errors. This mainly geared towards noisy signals, which is the largest problem when referring to optical signals across multiple hops.

3.3.1.0.6 Combination Multiple kinds of codes can be combined in order to achieve a higher level of accuracy and error correction. This can be used to avoid crosstalk in electrical signals since the control signals of the ring resonators are usually parallel. For example, SECDED can correct only one bit, while detecting up to two faults [86].

3.3.1.0.7 Power Efficiency of Coding Power cost is a large concern when using coding schemes. This is because the coding and decoding require specific modules at the network interface. It also lengthens the transmitted message. In electrical, the interconnect is a much larger concern, but since the data is transmitted

optically, which is much more efficient, the only significant concern is the power cost of the interface modules.

Higher efficiency can be attained by the aforementioned joint coding schemes (combinations). Reducing the number of bits of the code actually reduces the size of the module to translate it. Yu [87–89] proposes to use a simple code and strengthen it to cope with more errors, such as using the codes for sections, that way the same module can be used section by section. One major benefit that optical networks enjoy is that E2E is much more efficient than Hop-to-Hop coding.

3.3.2 Other Options From Electrical NoC

The other category of fault tolerance is various importable fault-tolerance schemes. If a fault tolerance scheme from an ENoC does not require the network to buffer and check at each node, or it can check during path setup, then the algorithm has a good chance of being successfully adapted to optical networks. We have already detailed some of these schemes in the previous section of this chapter, but many other schemes are available for porting. A lot of these schemes are listed in a survey written by Radetzki et al. [69].

3.4 Chapter Summary

In this chapter, we discussed some of the important related works that deal with PNoC architectures. More specifically, we covered the conventional network architectures, and fault tolerance mechanisms. In previous works, there hasn't been a method to avoid creating thermal variance across a chip, or react to process variation inside an optical switch. In the coming chapters, the proposed fault-tolerant router and path configuration schemes will be introduced as a solution to the process variation of optical switches, as well as a new routing algorithm which can prevent the traffic from creating hotspots in the network. These proposals combine to make a more reliable PNoC for future systems.

Chapter 4

Fault-Tolerant Photonic On-chip Network Architecture

4.1 Introduction

In this chapter we explain the proposed photonic network architecture and switch. The most important component of the network is the fault-tolerant electro-optic router. The main focus of this chapter will be on the fault-tolerant optical switch (FTTDOR), which can come in many forms. First, we will explain how the PNoC architecture is implemented. In this chapter, the trade-offs between blocking and non-blocking switches are briefly discussed. In addition, the challenges of designing a fault-tolerant optical switch are discussed. This chapter is related to the work published in [46].

4.2 System Architecture

The simplified block diagram of the FT-PHENIC system is shown in Fig. 4.1. The system consists of two networks: the top one is the photonic communication network (PCN), which consists of photonic switches that are interconnected by waveguides. The other network is the electronic control network (ECN), which is used for path reservation and configuration of the optical switches in the PCN. This is done by powering the MRs either *ON* or *OFF*. Each processing element (PE)

is connected to a local electronic router and also connected to the corresponding modulator and detector bank in the PCN. There are two types of messages that are used in FT-PHENIC: Control signals, which are routed in the ECN and used for path configuration; and payload signals, which are converted to optical data and sent through the PCN.

4.2.1 Network Architecture

The network is broken into two layers. They can clearly be seen in figure 4.2, which shows the whole network's architecture. The top layer is the PCN, which is responsible for transmitting the large payloads of the network. This is handled by connecting several optical switches together via optical interconnects. These interconnects can transmit data bidirectionally, but the design has one for each direction. The reason we state that they can handle bidirectional data is that the teardown signal in the network travels backwards through the line. It goes in the output ports and out the input ports.

The bottom layer is the electronic layer, which contains the electronic control network, the network interface, and processing elements. The processing elements (PEs) are responsible for making the actual calculations of the program. These PEs also determine where messages need to be sent, and what the message's payload consists of. The network interface (NI) is the primer between the PE and the two networks. It handles converting the payload to an optical signal, as well as creating the path-setup packet to initiate path reservation. It is directly connected to the modulator bank and detector bank on the optical layer via electrical TSVs. These TSVs control the modulators, and make them generate an optical data stream. The TSVs that are connected to the photodetectors bring the payload signal to the NI, so that it can send the payload to the PE. The electronic control network consists of electronic control routers that are interconnected electronically, similar to a standard ENoC.

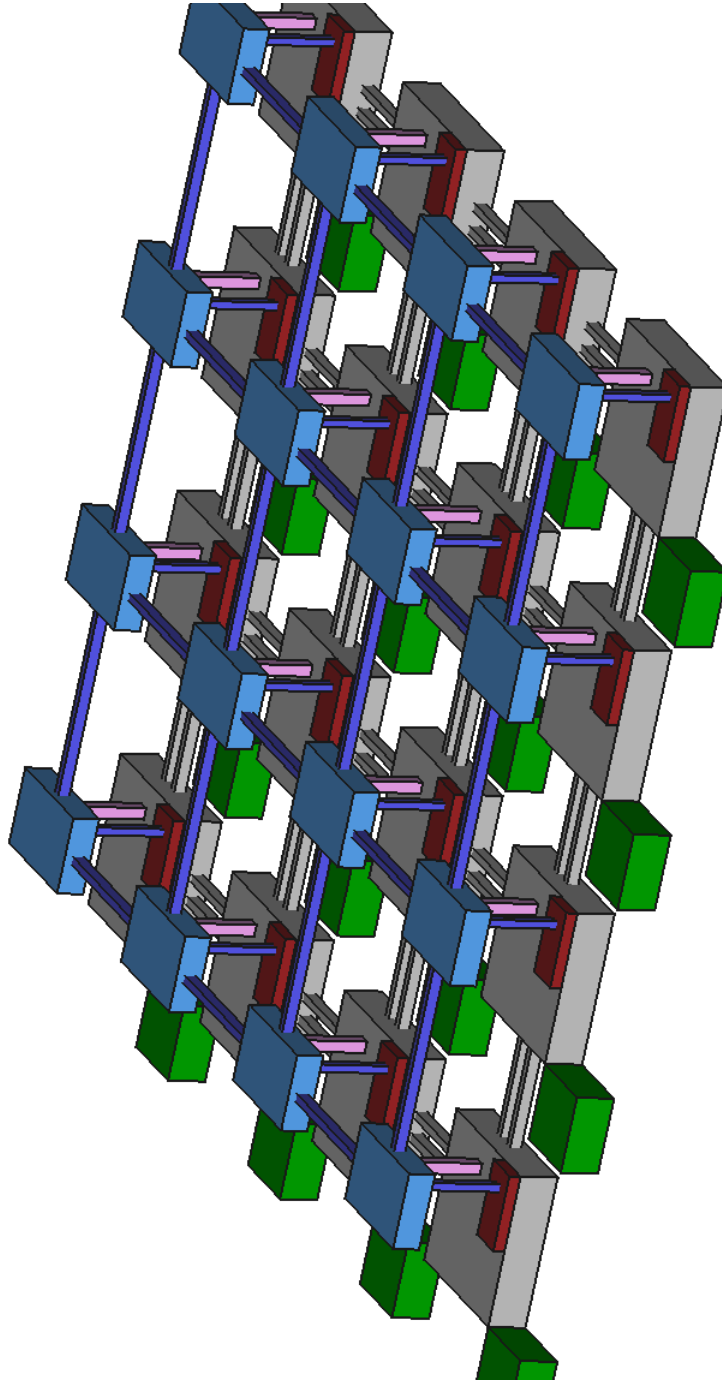


Figure 4.1: FT-PHENIC system architecture. Green boxes represent the PEs, grey boxes are the routers, blue boxes are the optical switches, red boxes are the network interfaces, pink wires are the control signals, grey wires are electrical interconnects, and purple wires are the optical interconnects

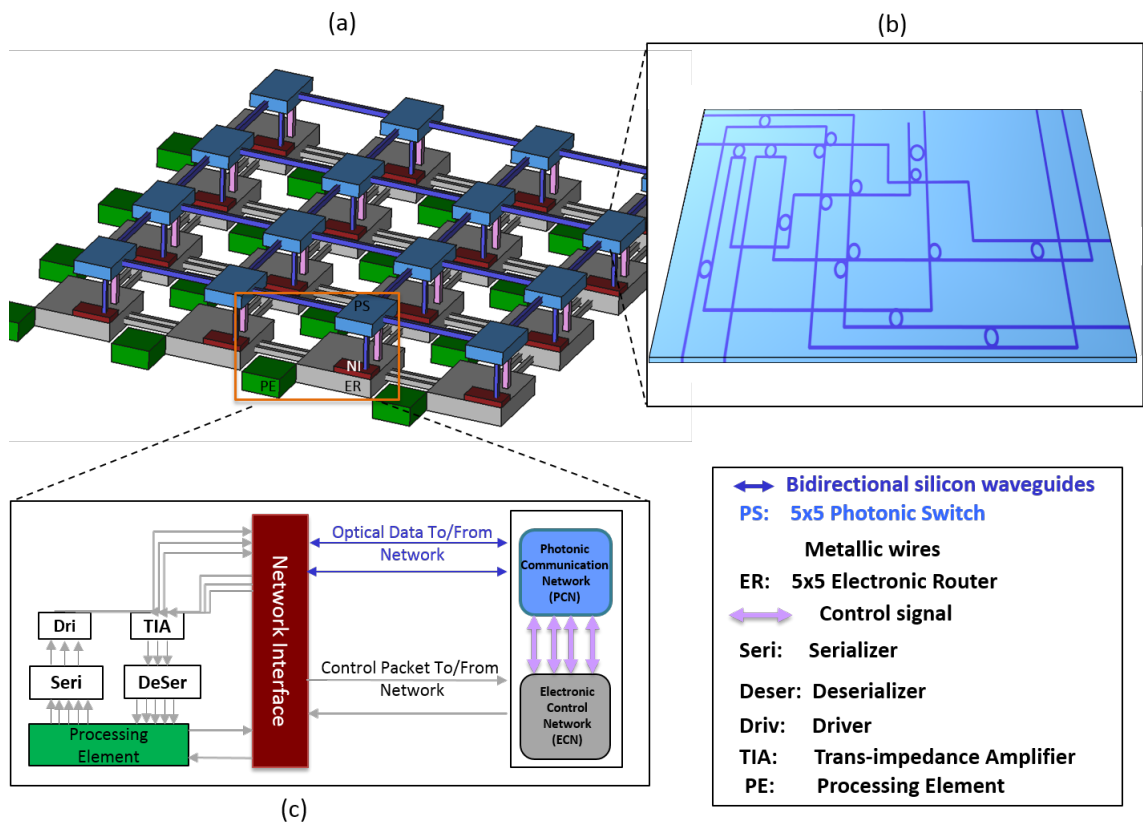


Figure 4.2: FT-PHENIC architecture. (a) Network (b) Optical switch (c) Node architecture

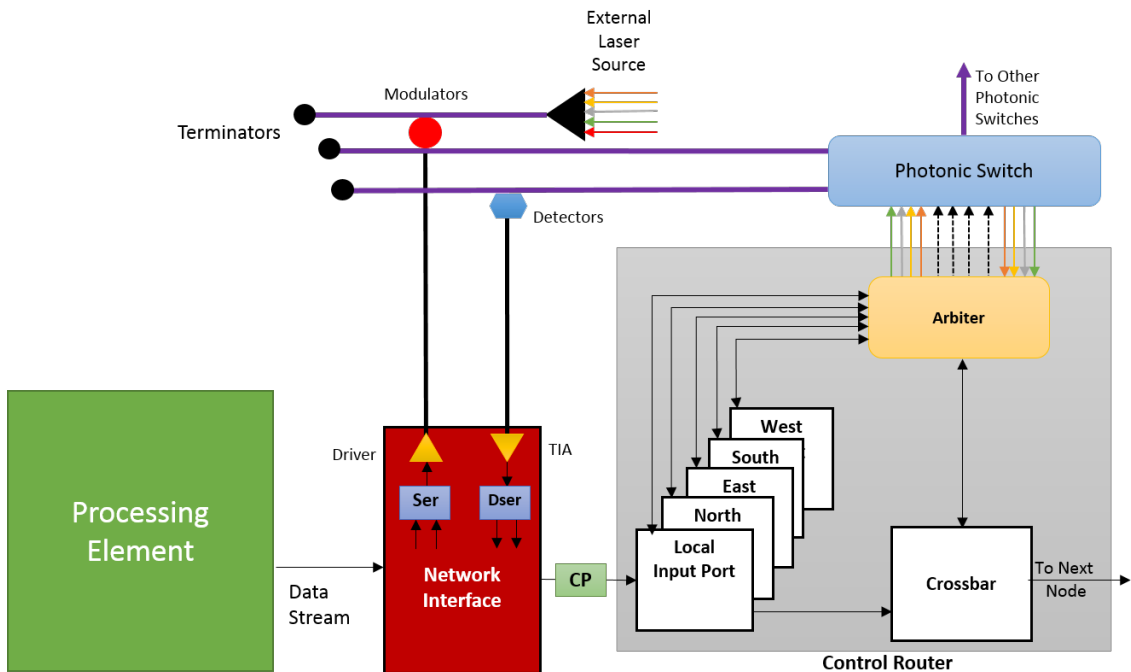


Figure 4.3: Architecture of a single node.

4.2.2 Node Architecture

Each node has components in both of the layers. The node architecture can be seen in figure 4.3. Each node has a modulator bank, detector bank, network interface, photonic switch and control router. Most of the details about these components have been described under the network architecture. To reiterate, the network interface connects the networks to the PE. One important detail that did not make it into the network architecture's description is how the PS is controlled. In the figure, you can see that the electronic router's arbiter is connected to the photonic switch with black lines as well as colored lines. The colored lines represent the signals necessary for the optical teardown, which is handled from node to node without the use of the network interface or the ECN. The arbiter receives the Teardown signal, disables the MRs that it was previously using for that transmission and then sends the signal to the next node. This is done only using optical communications. The other black lines represent the control signals for the photonic switch. These signals are what turn the MRs on or off based off of the decisions that the arbiter makes.

4.2.3 Electronic Router Architecture

The electronic routers have 3 main components, the Arbiter, crossbar, and input ports units.

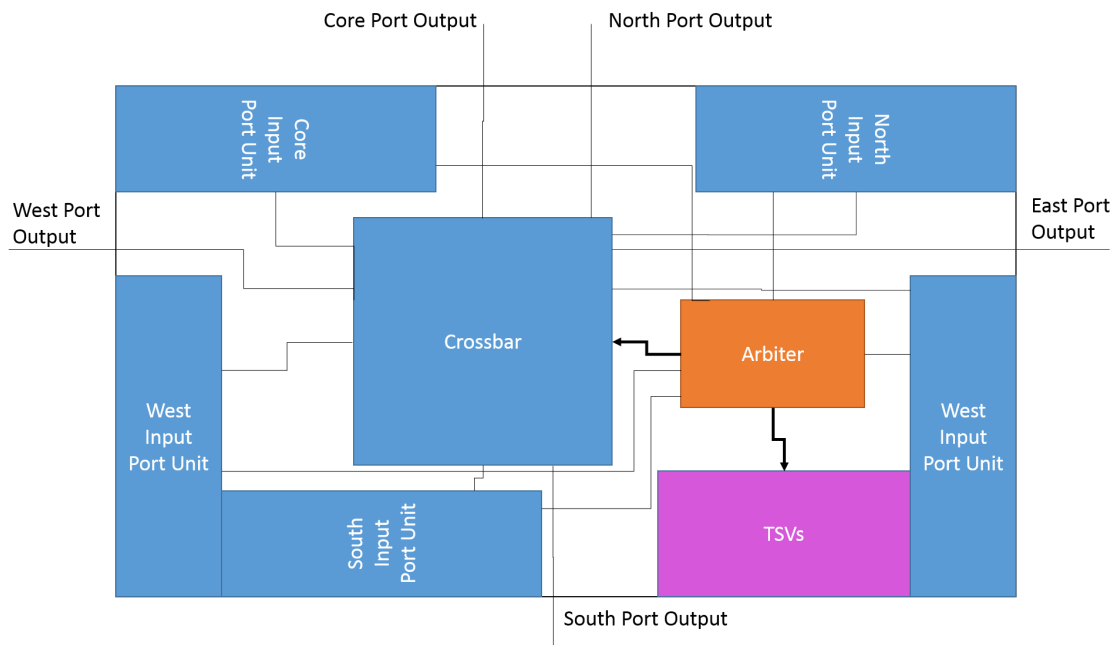


Figure 4.4: Control router architecture.

The input ports themselves consist of an input port buffer, and a router module. The input port buffers take the incoming data and stores it into a FIFO buffer. This is the first pipeline stage of a path setup packet. The router module then takes the bits of data that correlate to the messages destination and send it to the route computation unit in the arbiter. The arbiter determines which port the packet needs to be sent to next, thus accounting for the second pipeline stage. The arbiter then sends a signal to the 5x5 crossbar to prepare it for the input-to-output connection. The crossbar is a unit that can connect any input port to any output port. Once the crossbar has the proper switches allocated, the buffer sends the packet out to traverse the crossbar, thus accounting for the final pipeline stage (switch allocation is done simultaneously). The packet then goes on to the next node.

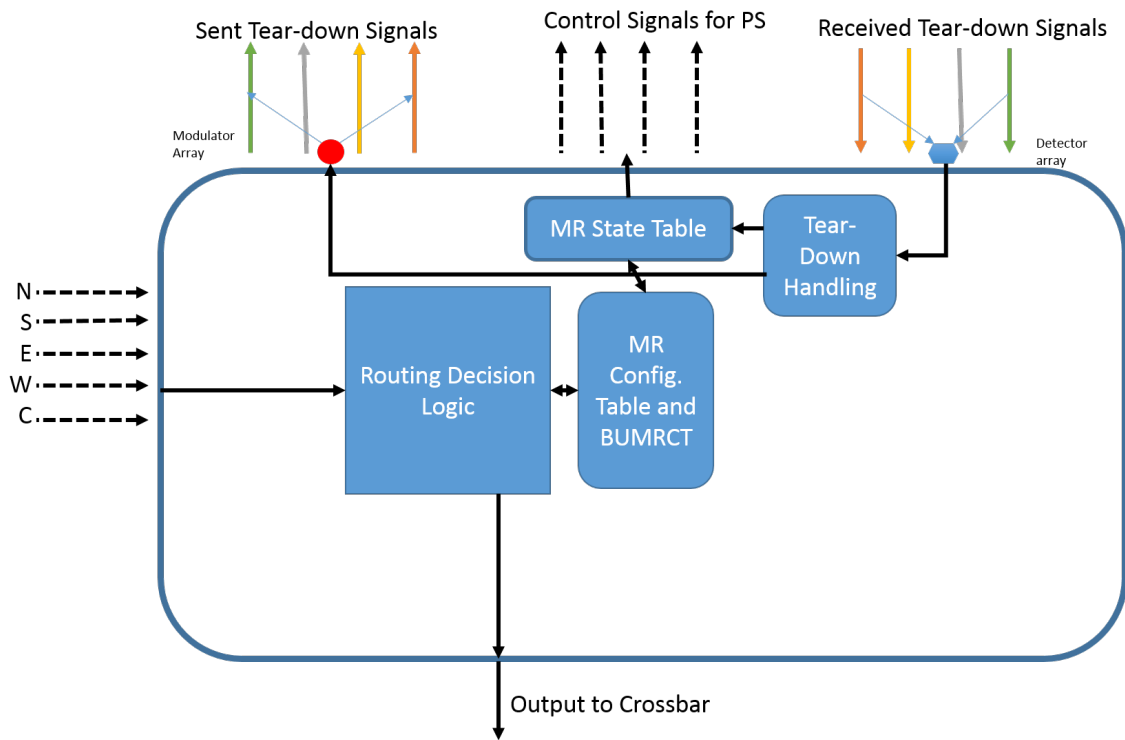


Figure 4.5: Fault-tolerant arbiter architecture

4.2.4 Arbiter Architecture

The arbiter has its own unique architecture, which can be seen in figure 4.5. It has a routing calculator unit, two MR configuration tables, an MR state table, and a teardown handling unit. The routing calculator unit will take the input port, and output port of a packet, the fault data, and determine the best option for the next node for the packet, and the corresponding port that leads to that node. It does this by using the information from the MR-state table, correlating that to certain port communications in the configuration table, and checking for their availability. If they are available, the MR state table is updated, and the MRs are reserved. If the port is unavailable, then it sends the path blocked packet. This process is more thoroughly described in the section titled “Fault-Aware Path Configuration Algorithm.” The teardown process is also described, but its important to note how even though the process involves the arbiter, the only other element that interacts with the teardown unit is the MR state table. This means that teardown can be done simultaneously with a routing computation.

4.2.5 FT-PHENIC Routing Algorithm

The routing algorithm is heavily based around Dimension Order XY. We call this new algorithm the Optical Hybrid Fault-tolerant Routing Algorithm (OHFT). OHFT, shown in Algorithm 1, starts off as a XY-DOR algorithm, with a toggle, and fault information of neighboring nodes. When a fault is encountered, OHFT attempts to keep minimalistic routing by simply switching the algorithm to DOR YX. If both of the minimalistic paths are unavailable, then the algorithm checks the remaining two ports, and utilizes the port which is less likely to return it to the same node.

The inputs are quite simple, it requires knowledge of the Dimension-Order-Routing Flag, the destination node, current node, and the fault status of neighboring nodes. The *Next-node* calculation, which is first called in line 1, is simply XY routing when DORF is true, and YX routing when DORF is false. Line 2 checks for the fault status of the node which is returned by the DORF calculation. In lines 3-5 the algorithm checks this fault status and either passes the original Next node, or recalculates the next node for the opposite DOR, by toggling the flag, and recalculating the *Next-node*. Line 6 and 7 check for the fault status of the node which is returned by the second *Next-node* calculation, which uses the new DORF. OHFT is selected for the path-setup routing selection in the ECN because of the benefits in fault tolerance that it provides. As with all adaptive routing, this can encourage extra turning, which may lead to loss issues in the long run, even though it still attempts minimalistic routing in terms of hops. It should be noted, that after a significant amount of node failures, the algorithm can have livelock, but this has minimal effect on the system as the MR failure rate would already be so high that the whole system is degrading by that point.

Algorithm 1: Fault-tolerant routing algorithm.

```

// Destination address
Input:  $X_{dest}, Y_{dest}$ 
// Current node address
Input:  $X_{cur}, Y_{cur}$ 
// Optical Fault status information
Input: PCN-in
// Current Direction Flag
Input: DORF-In
// Current Direction Flag
Output: DORF-Out
// Next Node
Output: Next
// Calculate the next node according to the Dimension Order Flag
1  $Next \leftarrow$  Next-node ( $X_{cur}, Y_{cur}, X_{dest}, Y_{dest}, DORF$ ); // Read fault
   information for the next-node
2  $Next\text{-}fault \leftarrow$  Fault-status ( $Next$ ); // Check the Other Minimal Direction
3 if ( $Next\text{-}fault == 0$ ) then
4   | return Next;
5 else Toggle DORF;  $Next \leftarrow$  Next-node ( $X_{cur}, Y_{cur}, X_{dest}, Y_{dest}, DORF$ );
   // Read fault information for the next-node
6  $Next\text{-}fault \leftarrow$  Fault-status ( $Next$ ); // Check the Other Minimal Direction
7 if ( $Next\text{-}fault == 0$ ) then
8   | return Next;
9 else Check both remaining directions;
   // A is against the current DOR priority, B is the node with it
   // Check the Other Minimal Direction
10 if ( $Next\text{-}faultA == 0$ ) then
11   | return A;
12 else Toggle DORF; return B;

```

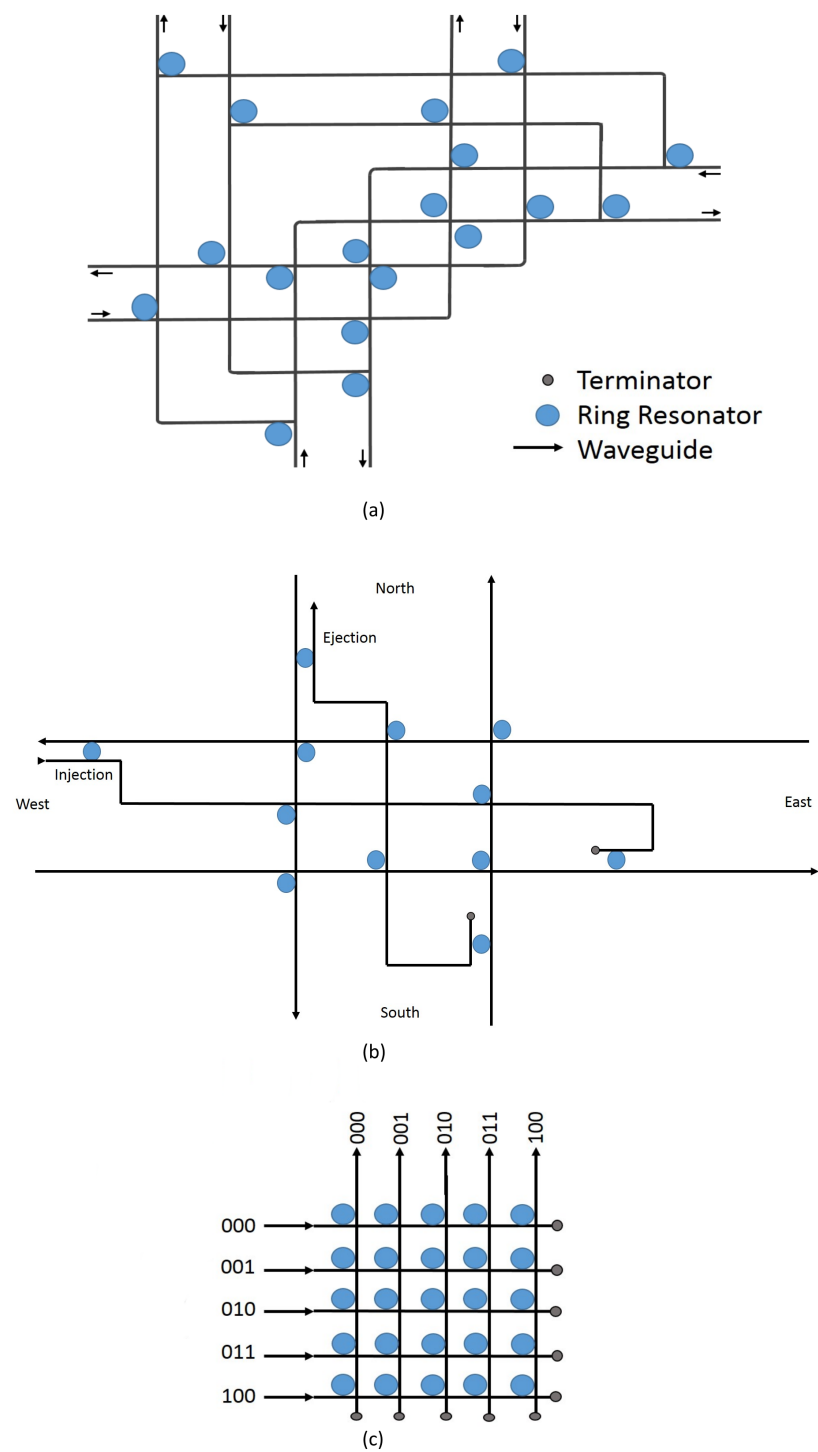


Figure 4.6: Comparison of 3 different 5-port switches (a)PHENIC, (b)Crossbar, (c) Crux [90])

4.3 FTTDOR: Fault-tolerant Non-Blocking Photonic Switch

Figures 4.7-4.9 shows the 5, 4, and 3 port versions of the FTTDOR optical switch. A node inside the network will use the 5-port variation, any nodes on an edge will use the 4-port variation, and each corner will have one of the 3-port variations. This is to cut down on power, and area. Another thing worth noting is that the ports may be labeled North, South, East, or West, but these labels will change, because no corner will actually have an East and a West port, but it will have 2 directional ports and a connection to the network interface. The optical waveguides carry the signal, similarly to how a wire carries an electrical signal.

Each of the circles shows the location of a MR. At special locations on the switch (Fig. 4.7), at specific critical locations, redundant MRs were placed to assure fault tolerance even if one of the MRs on the backup path has a fault. A fault at these locations would not change the route, just simply use a backup MR. The backup route for the NSEW directions is to actually use the waveguide connected to the core ports as a master backup; therefore, the redundant MRs are all chosen at the locations which connect the NSEW ports to the Core.

For a majority of faults, the design of the switch allows for an alternate, slightly less power efficient route. In fact, the backup route is less power-efficient because the packets travel across more waveguide distance, go through more active MRs, and cross more waveguides; however, the switch still maintains all of its functionality. Backup routes are only intended for use in the switches in which faults have occurred, the extra loss will have minimal effect on the message's signal strength across the whole network.

The original form is a 5-port non-blocking switch, meaning that it allows for routing from any available port to any other available port. Once a fault is detected, the switch becomes blocking; but, it should be able to maintain all functionality as long as none of the redundant MRs fail. This means that both of the MRs at any one of the critical locations are faulty. The design of the 7-port switch only has redundant MRs on the left side of the z-axis area. This is due to the fact that in XYZ routing,

the Up and Down should not ever travel into the X and Y directions. This means that their existence is not even necessary, but it is a nice feature for use in other routing algorithms. The fault-tolerant element comes in from the redundant MRs. This means that the fault-tolerant element of the switch is merely a modification of an already existing switch, as seen by the plethora of switches made in figures 4.7-4.9. This is an elegant solution to the problem of fault tolerance, which can be implemented on almost any switch that already exists. Because the redundant MRs lie dormant, they do not require much power other than the boost in signal strength required to compensate for the signal loss, caused by passing by an inactive MR, which is minimal. As all rerouting in the switch occurs on the core waveguide, traffic certainly increases on this one waveguide as too many faults occur, which is why it should be treated as a node failure after a threshold of failed MRs is reached.

The FTTDOR switch has been designed to require no MRs for inverse traffic (i.e. East-West or North-South). Since this kind of traffic accounts for a majority of the traffic of the PCN, such design will save on power and continue to function in the case of any MR failures. Assuming that a single location of redundant MRs does not fail all together, the switch is able to maintain all functionality at slowed speeds. Additionally, the MRs which connect parallel waveguides are replaced with racetracks. This allows for a wider pass-band of light frequencies, makes them less sensitive to physical faults, and have a larger Mean Time Between Failures (MTBF) [91]. Racetracks are oval shaped MRs, which have longer coupling regions.

The labeled resonators can be seen in Fig. 4.7 for the 5-port switch. Table 4.1 shows the corresponding resonators which need to be switched on for traffic going from any one particular port (left column) to any other particular port (top row), for the 5-port switch. In the same table, a “-” denotes a path which does not require any resonators to be turned on. Table 4.1 shows the primary paths for each of the active paths. It lists the MRs used to make the path. Table 4.2 shows the backup path for each communication, in case an MR fails. A backup path is only used if there is a detected fault. Fig. 4.10 is actually a demonstration of when a fault occurs in MR 9, and a packet is trying to travel from West to South. At the absence of a fault, only MR 9 would be used, and the signal would follow the red line. But

referring to Table 4.1, we can see that West to South has a backup route of MRs E, 15, and A. This adds 2 extra resonators to the path (represented dashed green line in Fig. 4.10) and some extra bends. But, the switch can still be used for West to South transmission. This is the key feature of the FTTDOR switch. These benefits can even be realized when comparing to the other 5-port routers that can be seen in Fig. 4.6.

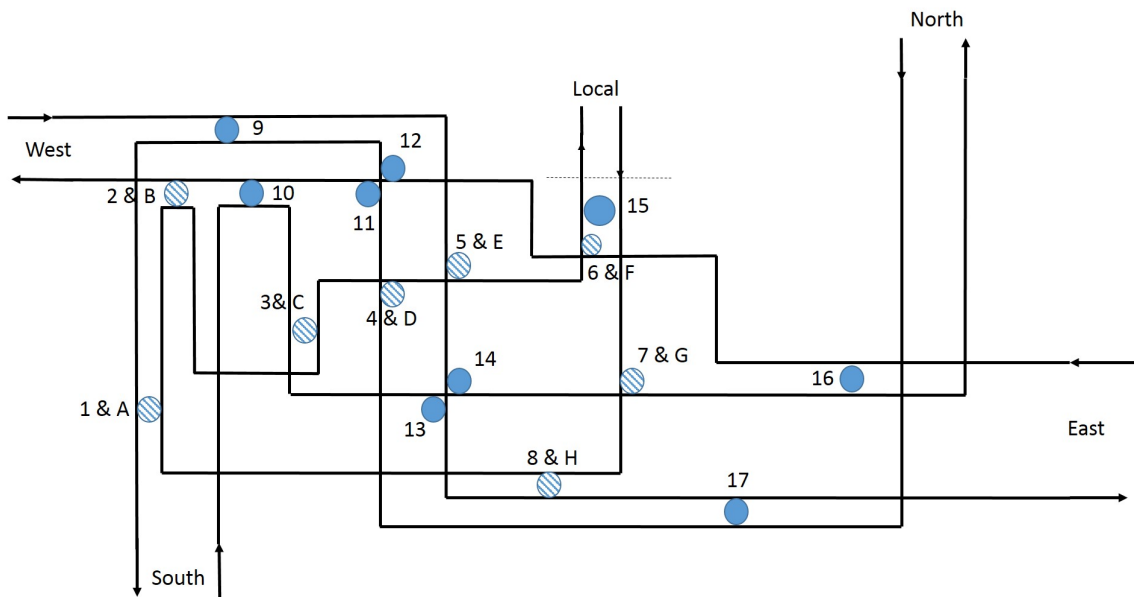


Figure 4.7: Fault-tolerant 5x5 optical router.

4.3.1 Building Blocks

4.3.1.1 Waveguides

The core of the proposed switch is a 5x5 non-blocking switch. As shown in Fig. 4.7, two connections are used for the ejection and injection from the core's output and input, respectively. This is one waveguide, which snakes around, and connects to each of the other waveguides via a single MR. Additionally, there is one pair of waveguides that go from east to west, and another pair that go north to south.

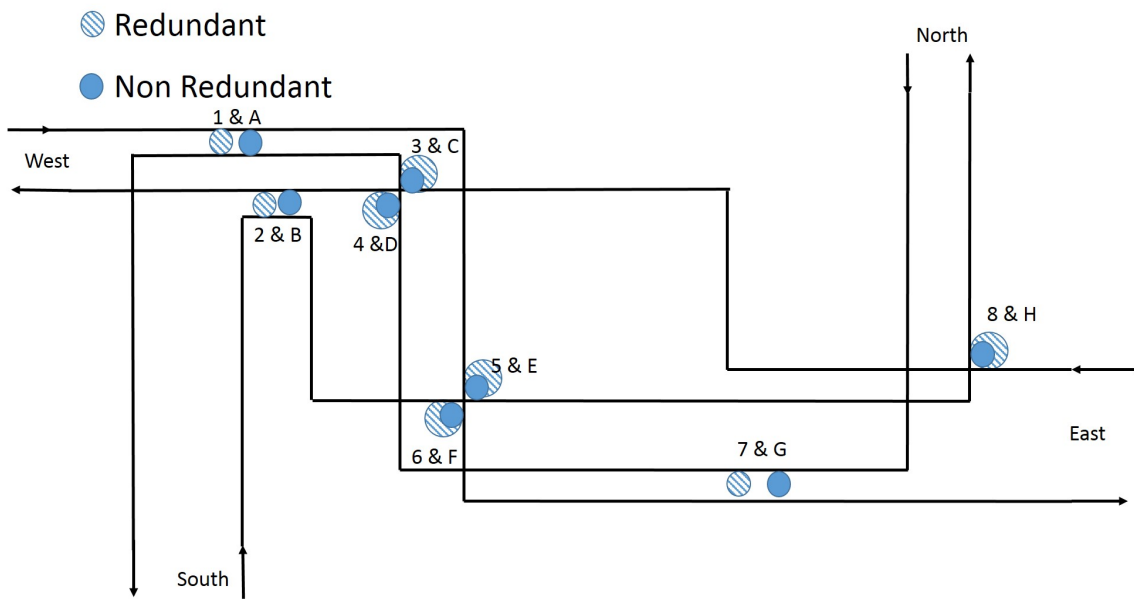


Figure 4.8: Fault-tolerant 4x4 optical router.

4.3.1.2 Micro Rings Resonators

FT-PHENIC's photonic switch uses two types of MRs. First, it has standard MRs, which are depicted with a solid dot. Second, at key locations, it uses redundant MRs. These locations are all chosen at the points where the core waveguide meets the other waveguides. This is because the functionality of these locations are critical for the fault tolerance to occur, so we reinforce it by adding MRs which connect the same 2 waveguides together. These locations are marked by a dashed circle.

4.3.2 Micro-Ring Configuration

Table 4.1 shows the MR configuration for data transmission, where 16 MRs are used in a non-blocking fashion. The proposed network also has a Backup MR Configuration Table (BUMRCT). This uses an additional 9 MRs to ensure that every path inside the switch is attainable with a completely different set of switches, and can be seen in Table 4.2.

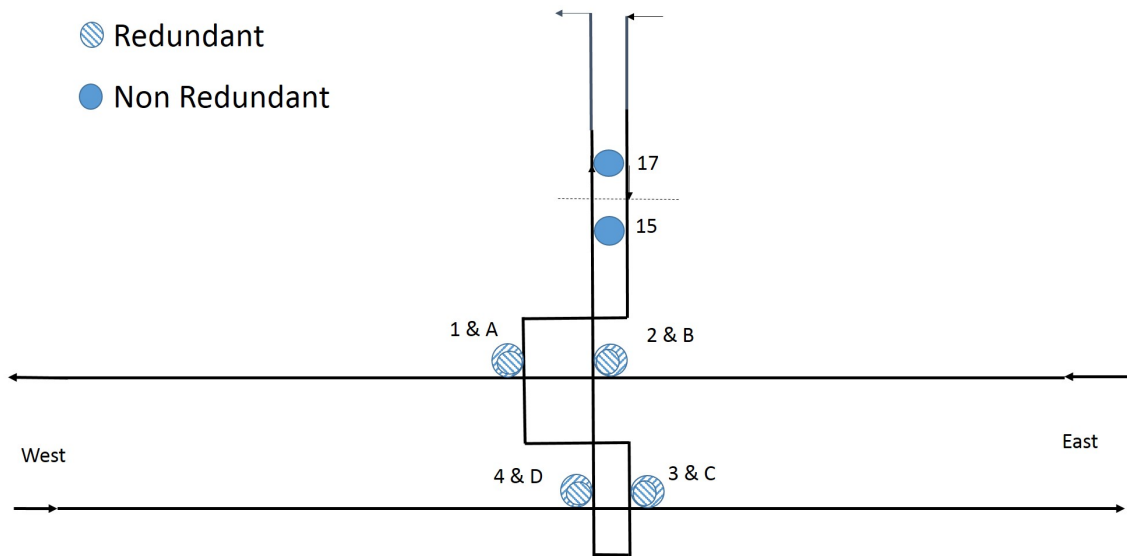


Figure 4.9: Fault-tolerant 3x3 optical router.

Table 4.1: Microring configuration for normal data transmission.

Output/Input	Core	North	East	South	West
Core	-	4	6	3	5
North	7	-	16	None	14
East	8	17	-	13	None
South	1	None	12	-	9
West	2	11	None	10	-

4.3.3 Optical Power Loss Evaluation

The trade-off between the blocking and the non blocking switches is essentially about performance vs power. The total optical laser power delivered to the chip is given by equation 4.3.1, where P , S , IL_{max} , and n are the power threshold, the detector sensitivity, the worst case insertion loss and the number of wavelengths, respectively.

$$P_{threshold} - D_{sensitivity} \geq IL_{max} + 10 \log_{10} n \quad (4.3.1)$$

The power threshold is the maximum amount of injected power that will prevent photonic component from having non-linear behavior. For example, waveguides and

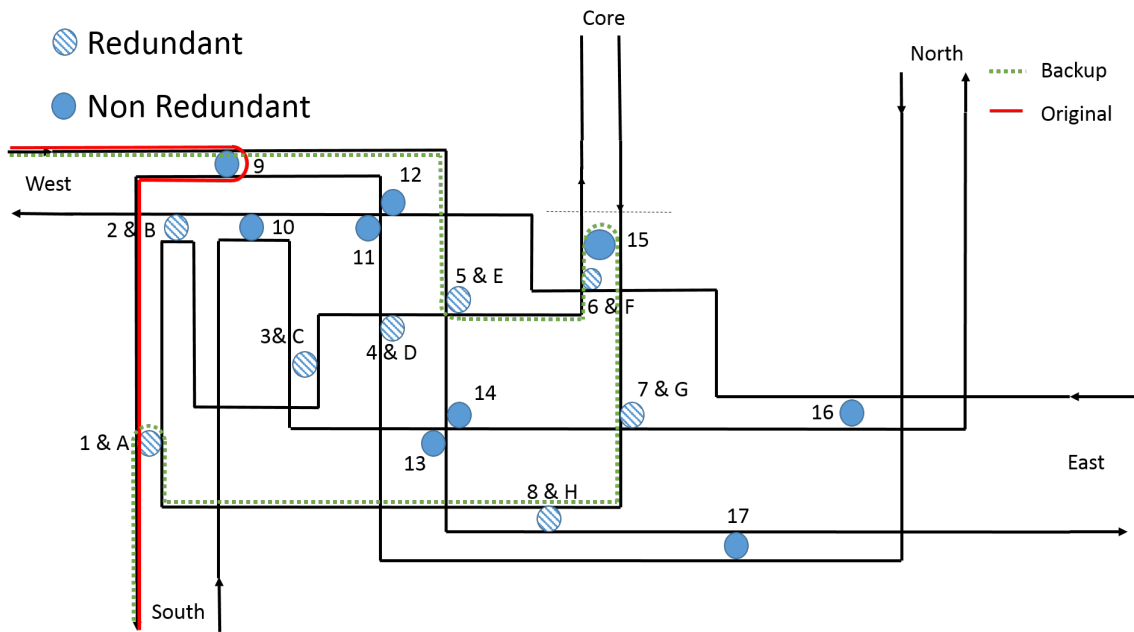


Figure 4.10: Showing an example of rerouting within a router with a fault at MR 9.

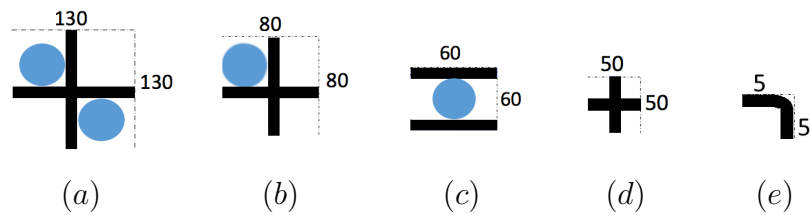


Figure 4.11: Optical router components. (a) crossing element with two complementary MRS, (b) crossing element with a single MR, (c) parallel switching MR, (d) Waveguide crossing, and (e) Waveguide bend. Numbers correlate to the marked dimension in μm [1, 92].

modulators have a power threshold of 15 dBm [93] and -2 dBm [50], respectively. The detector sensitivity is the amount of signal power required to excite the photodiode and generate a signal. A sensitivity of 7.3 dBm has been demonstrated in [94] with a bit-error-rate of 10^{-12} . IL_{max} represents the worst case insertion loss from source to destination.

$$Total_{(loss)} = PassBy_{(loss)} + PassThr_{(loss)} + Cross_{(loss)} + Bend_{(loss)} + Prop_{(loss)} \quad (4.3.2)$$

$PassBy_{(loss)}$ is when the signal passes by a MR without entering into it. The

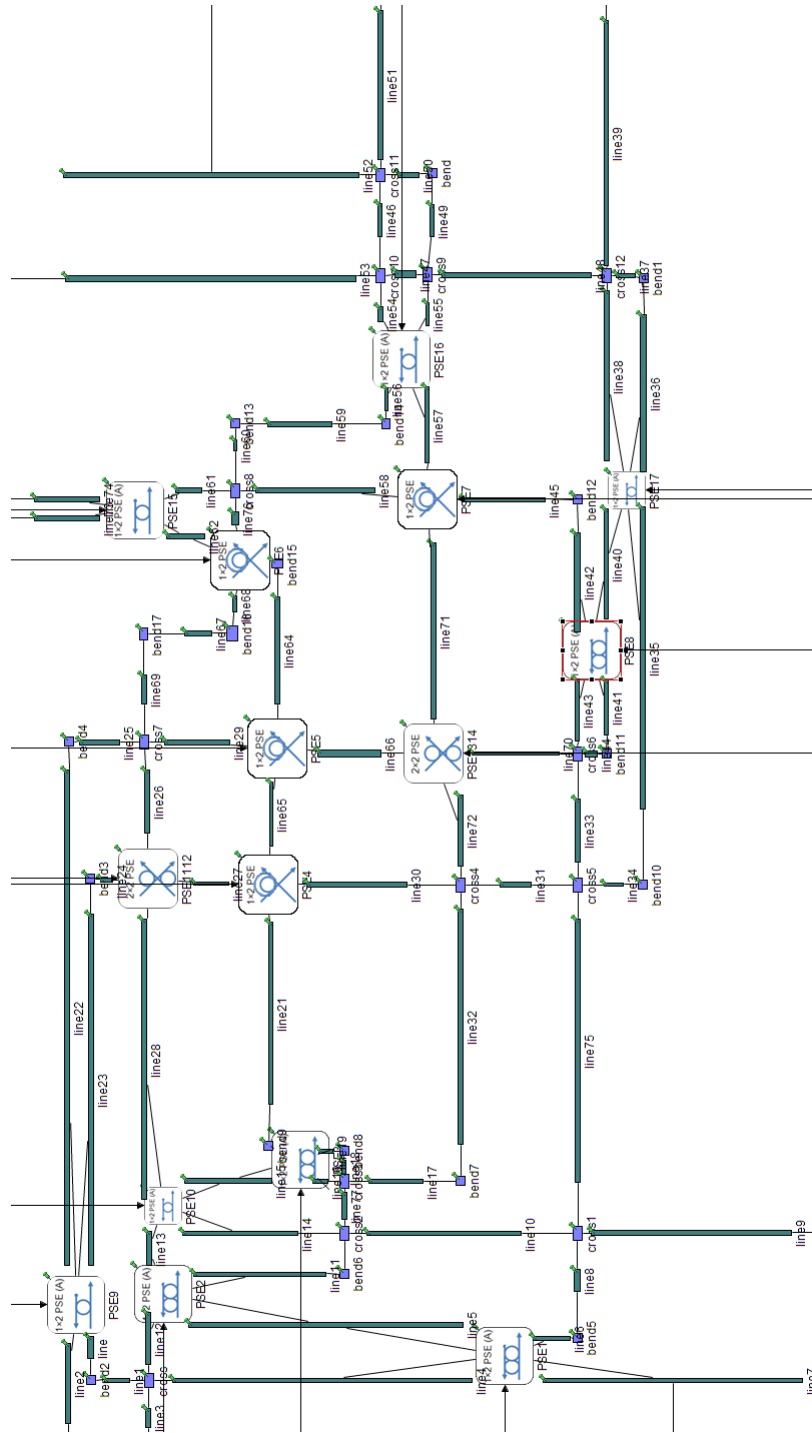


Figure 4.12: Photonic switch building blocks instantiation.

Table 4.2: Microring backup configuration for data transmission.

Output/Input	Core	North	East	South	West
Core	15	D	F	C	E
North	G	-	F,15,G	None	E,15,G
East	H	D,15,H	-	C,15,H	None
South	A	None	F,15,A	-	E,15,A
West	B	D,15,B	None	C,15,B	-

$PassThr_{(loss)}$ indicates when the signal enters a MR. The $Cross_{(loss)}$, $Bend_{(loss)}$ and $Prop_{(loss)}$ are the losses caused by the crossing element, bending element and the total propagation loss between the source and the destination, respectively.

Because the path configuration scheme uses some optical transmissions, the losses also had to be considered. Because the teardown is handled on a hop by hop basis, it only traverses a single interconnect and has a significantly lower loss than the payload transmission.

The power overhead is created by the fact that modulators lose an amount of power, and the photo detectors require a certain amount of power. This is consistent for any transmission, independent of its hop count, so it is considered to be an overhead. This is increased because of the use of optical components for the ACK and teardown, but still result in overall power savings for the chip, because when they are done on the ECN, they require a lot of power to buffer and retransmit at each hop. There are a few significant differences between optical switches. One of

Table 4.3: Insertion loss parameters [95–97].

Parameter	Value
Propagation loss (silicon)	1.2 dB/cm
Waveguide crossing	0.12 dB
Waveguide bending	0.005 dB/90°
Drop into a ring	0.5 dB
Passing by a ring	0.005 dB

the most major concerns is transmission capabilities, and then the second biggest concern is insertion loss. To show the difference between some switches, the insertion loss of the FTTDOR photonic switch is evaluated against a blocking switch [92], and PHENIC's previous switch. Figure 4.11 shows the different components used to

model the photonic switches. The numbers in the figures represent the dimension of the component [1]. The two basic switching elements are the crossing MR and the parallel MR. Table 4.3 shows the value of various sources of loss that an optical signal can experience as reported in [95–97]. These values can be used to calculate the loss for every case of a switch.

Table 4.4: Comparison between 5×5 optical routers.

	PHENIC	Bl-Switch [92]	FTTDOR
Non-blocking	Yes	No	Yes
Number of Ring	18	12	16+9
Number of Crossing	27	10	19
Passive routing	Yes (x4)	Yes (x4)	Yes (x4)

Table 4.4 compares the three evaluated switches on a basic level. It is clear that the blocking switch has better physical characteristics (i.e., less waveguide crossing and fewer MRs) than the FTTDOR. This kind of switch is used for light traffic loads, because if packets do not have to compete for the same node, then using this switch gives better power efficiency, and will not affect the performance. On the other hand, if the network does have a significant enough amount of traffic, then the non-blocking switches have much higher performance capabilities. This is because a network that uses a blocking switch shows higher energy and the number of blocked requests increases [45].

Table 4.5 shows the optical power loss of each of the twenty possible communication pairs inside the switch, (e.g. $N \mapsto L$ is the optical power loss from the North port to the Local port). It shows that the blocking switch has the best average loss. This is due to the reduced number of crossings and MRs inside the router.

4.4 Light-Weight Electronic Controller Architecture

In an Electronically Assisted PNoC, the Electronic Control Network is considered to be the main source of latency and power consumption. This overhead might be caused by the use of an inappropriate message size, a non-optimized physical channel

Table 4.5: Power loss comparison.

Direction	PHENIC	Blocking [92]	FTTDOR	Direction	PHENIC	Blocking [92]	FTTDOR
E \rightarrow L	1.36	0.5	0.99	N \rightarrow S	1.36	1	1
E \rightarrow N	1.11	0.62	0.98	N \rightarrow W	1.11	0.62	1.35
E \rightarrow S	0.99	0.63	1.35	S \rightarrow E	0.99	0.63	1.12
E \rightarrow W	1.48	1	0.62	S \rightarrow L	0.87	1.13	1.26
L \rightarrow E	1.48	1.13	0.76	S \rightarrow N	1.48	1	0.64
L \rightarrow N	1.24	1.5	1	S \rightarrow W	1.11	0.62	0.87
L \rightarrow S	1.11	0.5	1.12	W \rightarrow E	1.36	1	0.64
L \rightarrow W	0.86	1.12	1.25	W \rightarrow L	0.74	1.5	0.76
N \rightarrow E	0.99	0.62	0.99	W \rightarrow N	1.11	0.62	1.13
N \rightarrow L	1.24	1.12	1.38	W \rightarrow S	0.99	0.62	0.63
Average Loss							
PHENIC	1.15						
Blocking PNoC	0.87						
FTTDOR	0.99						

width, or especially the used a non-optimized path setup protocol, which is a source of both power and latency overhead. To solve this problem, in [25] we proposed a path configuration scheme that reduces the load of the electronic router. It does this by reducing the overall amount of messages that go through the electronic router.

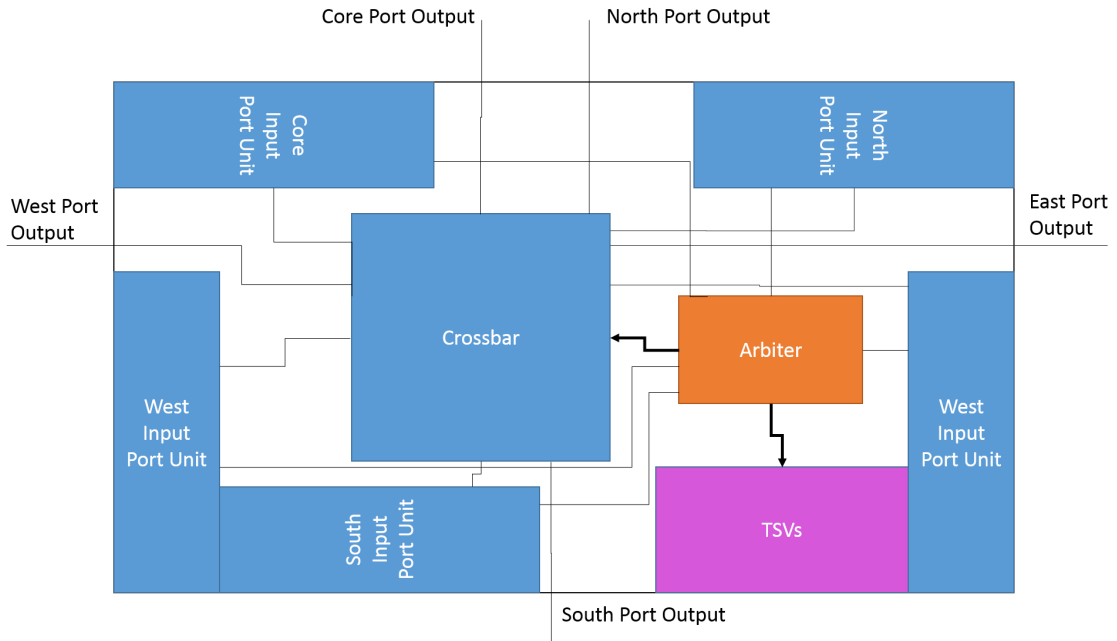


Figure 4.13: PHENIC's light-weight electronic router.

Figure 4.13 shows PHENIC's light-Weight Electronic Controller. This will be modified in the chapter about the path configuration algorithm, so we will discuss it in detail there. As an overview, this unit controls the optical switches via electrical TSVs. It is called it lightweight because it is able to handle teardown signals without using the electrical network, and it does not require a reserved state.

4.5 FTTDOR Evaluation

4.5.1 Area Evaluation

The area evaluation of the photonic circuit was done as a sum of products. Equations 4.5.3 - 4.5.5 show the process used.

The parameters used to get to equation 3 can be seen in Table 4.6.

Table 4.6: Area parameters [1]

Component	Area(mm^2)
Modulator	0.13
Ring Resonator	.36
Waveguide 1mm	.02
Waveguide Bend	.025
Terminator	.02
Photodetector	0.22

$$OpticalArea = \sum_0^n count(comp) * A(comp) \quad (4.5.3)$$

$$\begin{aligned}
OpticalArea = & count(MR) * Area(MR) + \\
& count(terminator) * Area(terminator) + \\
& count(waveguide) * Area(waveguide) + \\
& count(modulator) * Area(modulator) + \\
& count(photodetector) * Area(photodetector) + \\
& count(bend) * Area(bend) \quad (4.5.4)
\end{aligned}$$

$$\begin{aligned}
OpticalArea = & count(MR) * .36 + count(terminator) * .02 + \\
& count(waveguide) * .02 + count(modulator) * .13 + \\
& count(photodetector) * .22 + count(bend) * .025 \quad (4.5.5)
\end{aligned}$$

4.5.2 Loss and Bit Error Rate

The final section is the reliability evaluation. We start with the signal to noise ratio, which is taken from Phoenix Sim. We are able to get detailed information on the signal and the noise levels in the network for each traffic pattern. To start off,

the bit error rate is defined as seen in equation 4.5.6.

$$BER = p(1)P(0/1) + p(0)P(1/0) \quad (4.5.6)$$

This is to say that the probability of a 0 bit being read as a 1 times the probability of a 0 bit, plus the probability of a 1 bit being read as a 0 times the probability of a 1 bit. The probability that a bit is either a 1 or a 0 is assumed to be 50%, thus simplifying it to equation 4.5.7.

$$BER = \frac{1}{2}[P(0/1) + P(1/0)] \quad (4.5.7)$$

The next step is determining the probability of the misread bits, which is given by Agrawal in [98]. This creates our final definition of the BER, which is shown in equation 4.5.8.

$$BER = \frac{1}{2}erfc(\sqrt{SNR}) \quad (4.5.8)$$

Where erfc is the complementary error function. Agarwal also goes on to state that the critical BER goal should be 1E-9, or that only one in a billion bits has an error. He also states that this happens around a SNR value of 15.6.

To find the BER, we have to first start with the SNR. Figure 4.14 shows the average received signal strength, average noise and average SNR. This was evaluated on multiple benchmarks, but because of the long distance communications, only FFT is shown. The Data flow's hop-by-hop nature renders its data useless for evaluating a network, and simply evaluates the network interfaces.

For a true evaluation of the Error rate, we use the worst case SNR, which can be seen in Figure 4.15. It is important to remember that the goal SNR was 15.6 dB. Based on this, we can see that none of the networks were able to handle a 256 core simulation. This is because as the network grows, the corners get farther and farther from each other. As the propagation distance grows, the loss also increases, and the signal becomes too weak to create a current at the photodetector.

We then calculated the BERs according to the values from the SNR, and these results can be seen in Figure 4.16.

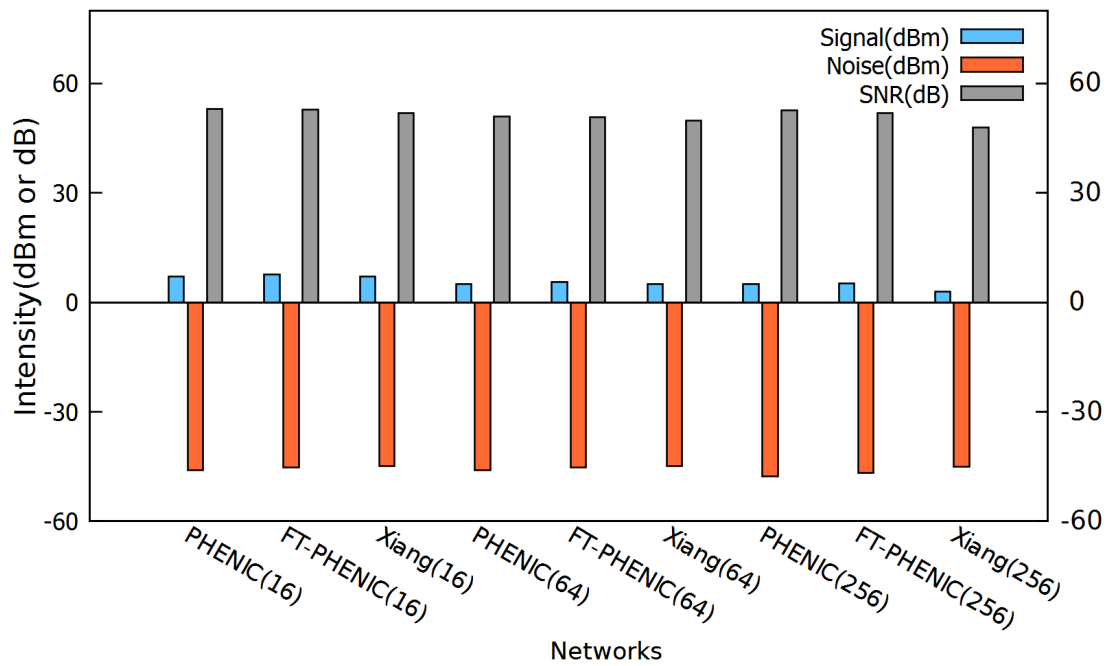


Figure 4.14: Signal, noise, and SNR average values for FFT simulation

4.6 Chapter Summary

In this chapter, we introduced the proposed FTTDOR router, which is used in the FT-PHENIC architecture. The proposed system uses a fault-tolerant non-blocking photonic switch and a light-weight electronic controller. We discussed the different challenges when designing the electronic controller and how the blocking occurrence degrades the system performance considerably, especially if a blocking photonic switch is used. In the next chapter, the path configuration algorithm is introduced, delivering a few more details on the architecture as a whole. A contention-aware path configuration algorithm was imbued with fault-tolerance.

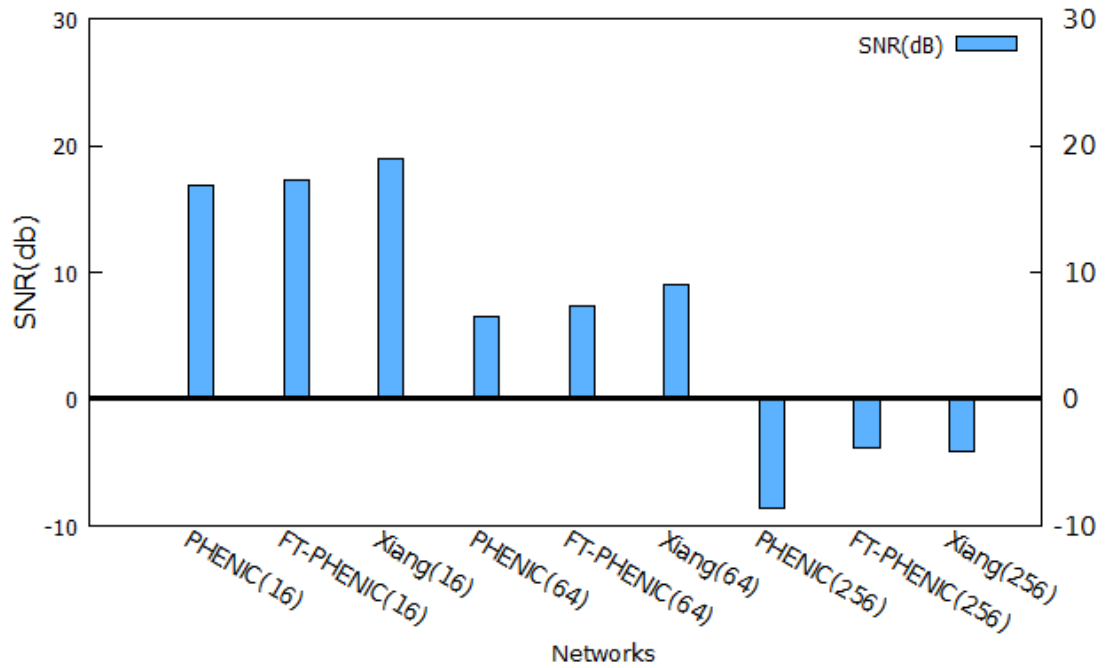


Figure 4.15: Worst case SNRs for each network for FFT

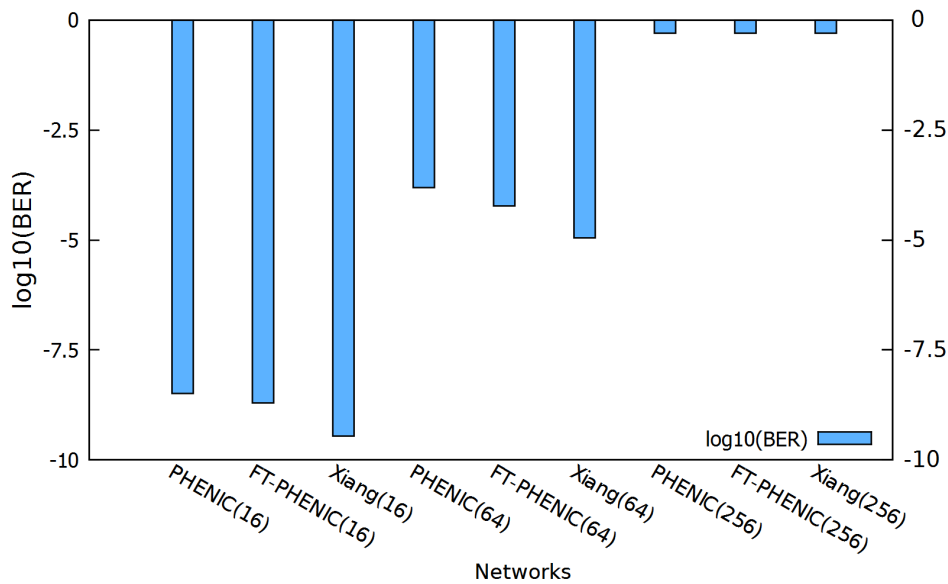


Figure 4.16: Worst case BERs for each network for FFT

Chapter 5

Fault-Tolerant Path Configuration Algorithm

5.1 Introduction

After introducing the FT-PHENIC architecture, this chapter covers the proposed fault-tolerant path configuration algorithm. First, the algorithm is described, and then the system is evaluated. We will inject several faulty MRs into the network and see their affect on the system. Additionally, We will compare the new FT-PHENIC to some other systems without faults, so we can see the cost of implementing the new Fault-tolerant Path Configuration Algorithm.

In chapter 4, we covered the architecture to explain how the switch is used. This chapter is about path setup, and involves interacting with the whole network. It involves interacting with the network architecture, the arbiter architecture, The electronic router's architecture, the FT-routing algorithm, and then the path configuration algorithm. This chapter is relates to the work published in [99].

5.2 Fault-Tolerant Path Configuration Algorithm

One major component of the system that requires change for fault tolerance is the path configuration algorithm. This is where the router can handle fault awareness and manage a fault tolerant switch. It does need to be made aware of different

statuses of the MRs, and make a path choice based on that. Additionally, the path configuration must handle all of the requirements set forth by the original PHENIC's path configuration algorithm.

5.2.1 Path Configuration

5.2.1.1 Path Setup

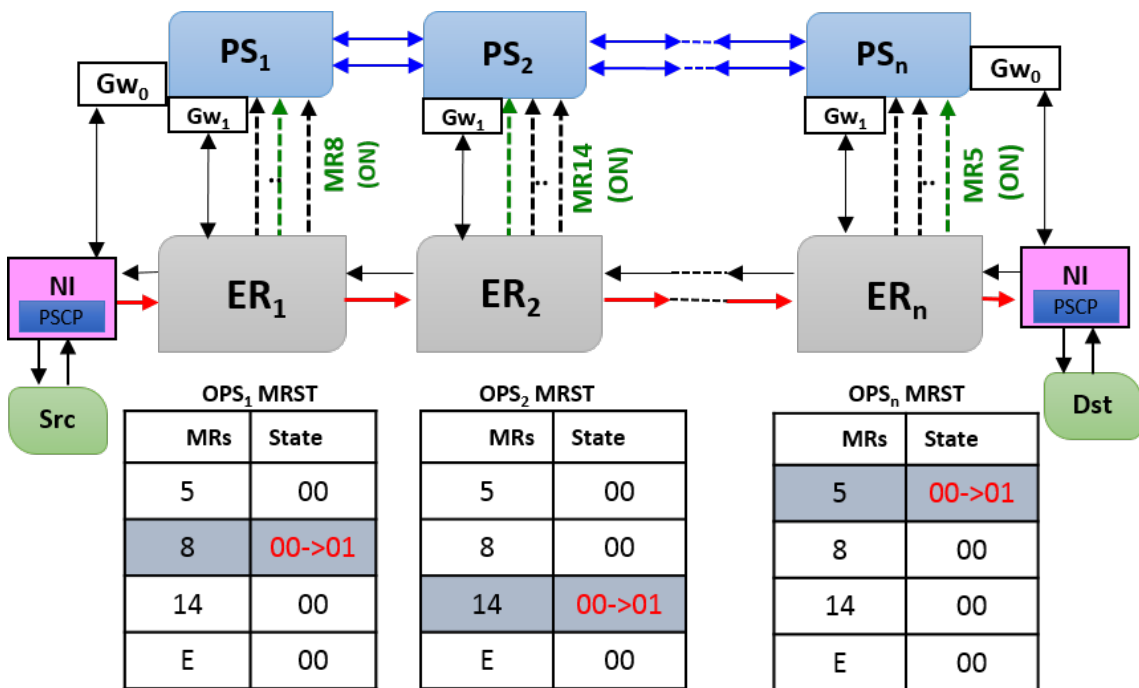


Figure 5.1: Successful path-setup.

Every optical path must be configured in order for the transmission to end up at the proper place. The way that this is done is by first sending a packet through the ECN. This packet is called the Path-Setup-Control-Packet (PSCP). This contains information of the source and destination addresses, and other information. Some information is as basic as message ID or error-correcting code to ensure message integrity, but some of this information is used to determine the packet type. Some PSCPs are for Path Setup (PS), and others are Path-Blocked (PB). The field that determines the packet type is a single bit, and is “0” for a PS and “1” for a PB. The Path-Blocked process will be described in the next subsection.

Algorithm 2: Fault-tolerant path-configuration algorithm.

```

// Path Setup Control Packet for communication  $i$ , PSCP $_i$ 
// Path Blocked Packet for communication  $i$ , PB $_i$ 
Input:  $S_i, D_i$ 
// From ACK detector
Input:  $Detc_{ACK_s}$ 
// To ACK modulator
Output:  $Mod_{ACK_s}$ 
// From Teardown detector
Input:  $TeardMod_i$ 
// To Teardown modulator
Output:  $TeardMod_i$ 
// To Microring resonator
Output:  $MRs_{j=0\dots n}$ 
// Buffer writing and routing computation stages
1 initialization;
2 while (Path-Setup-Control-Packet (PSCP) !=0) do
3   DestAdd  $\leftarrow$  PSCP $_i$ ;
4   PortIn  $\leftarrow$  PSCP $_i$ ;
5   if (resource are available) then                                     /* check MRs state */
6     if (MR is not faulty) then
7       Grant $_i$   $\leftarrow$  Arbiter;
8     else if (Backup MR is Not Faulty) then                             /* check backup MRs state */
9       GrantBackup $_i$   $\leftarrow$  Arbiter;
10    else                                                                 /* no possible path */
11      Blocked $_i$   $\leftarrow$  Arbiter;
12      FaultyNodeList  $\leftarrow$  Node;
13    end
14  else                                                                 /* generate path blocked */
15    Blocked $_i$   $\leftarrow$  Arbiter;
16  end
17 end
// Path blocked
18 initialization;
19 while ( $PB \neq 0$ ) do                                                 /* Path blocked arrives */
20   if ( $MRs_i$  state is reserved) then                                 /* release reserved MRs */
21     release  $\leftarrow$  MR $s_i$ ;
22 end
// Generate ACK
23 initialization;
24 while ( $NI$  receiver  $\leftarrow$  PSCP $_i$ ) do                               /* PSCP arrives to Dest */
25   if ( $PSCP$  arrives to  $NI$ ) then                                     /* generate ACK to Src */
26     ACK $_i$   $\leftarrow$  To modulator ACK ( $\lambda_0$ );
27 end
// Receives ACK
28 initialization;
29 while ( $NI$  receiver  $\leftarrow$  ACK $_i$  ( $\lambda_0$ )) do                       /* ACK arrives to Src  $\lambda_0$  */
30   if ( $ACK$  arrives to the  $N_i$  sender) then                         /* modulate the data */
31     Data $_i$   $\leftarrow$  To Data's Modulator;
32 end
// Identify and Generate Teardown $_i$ 
33 initialization;
34 while (From detector signal =Teardown $_i$  with  $\lambda_i$ ) do
35   findInport  $\leftarrow$   $\lambda_i$ ;                                           /* find In-port according to the wavelength */
36   free  $\leftarrow$  MR $s_i$ ;                                                 /* Free involved MRs */
37   Teardown $_i$   $\leftarrow$  To modulator  $\lambda_i$ ;                             /* generate new Tear-down according to  $\lambda_i$  */
38 end

```

In Fig. 5.1, the PS packet starts at the source node. It is then sent to the next node according to the routing algorithm that is used. At each node along the way, the path setup packet reserves the interconnects and MRs that are required for that node’s role in routing the optical message. The way that this is done is that the corresponding input and output ports for that message at that node have a corresponding set of MRs in the MR Configuration Table (MRCT). The MRCT is specific to the switch that is used for the network.

Once the proper MRs are determined based on the ports, the ports are checked. Assuming that the ports are free, then the state for the corresponding MRs in the MR State Table (MRST) is changed from a “00” (Free) to a “01” (Active). The third state, “10”, is the faulty state. This will be described later on in this section. Once each of the MRs in the whole path from source to destination have been set to “01”, the hop-based reservation process is complete, and then the ACK signal is sent.

5.2.1.2 Blocked Paths

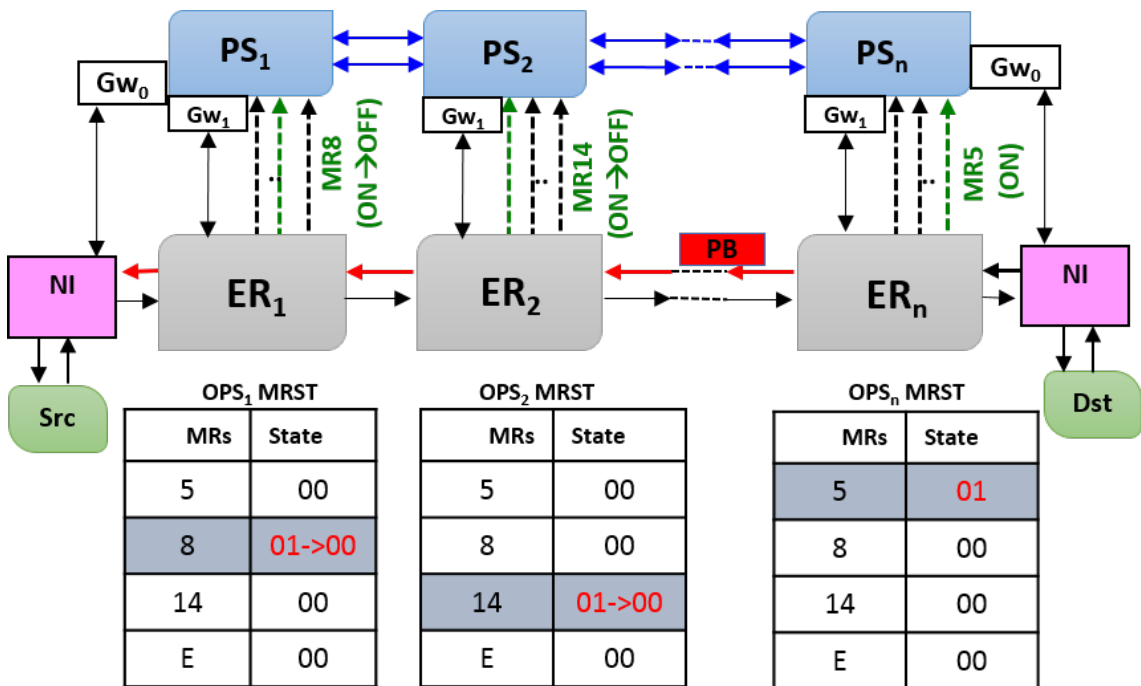


Figure 5.2: Failed path-setup.

In some cases, the requested MRs or paths are already reserved. In such a case, the blocked path process initiates. This can be seen in figure 5.2. The last node of the process wants to reserve MR 5, but it is already active. So the electrical router creates a path blocked packet. This packet will travel the reverse direction of the original set-up packet and frees the MRs in the corresponding MRSTs. Once it reaches the source node, all MRs that were set up via the PS packet should have been freed by the PB packet.

5.2.1.3 Faulty Switch

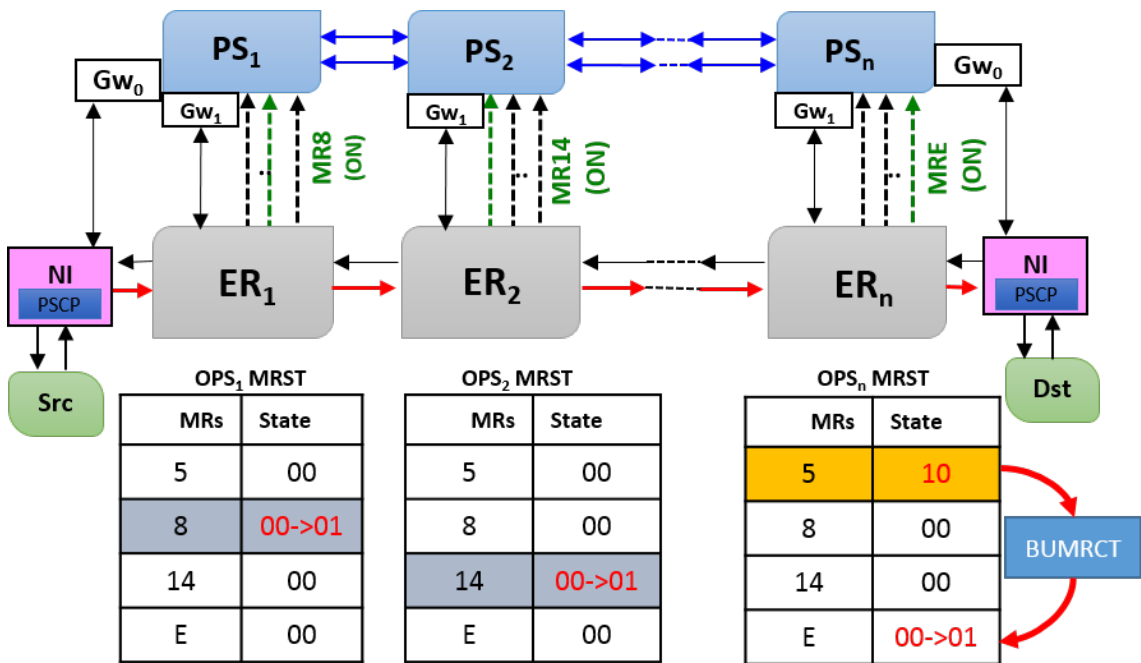


Figure 5.3: Faulty path-setup.

In other cases, the requested MRs are marked as faulty. In such cases, the switch attempts to use a different path inside the switch to achieve the same hops. This can be seen in figure 5.3. The last node of the process wants to reserve MR 5, but it is already marked as faulty. The arbiter then checks the backup path according to the Backup MR State Table. If this path is available and not faulty, then everything is done. If the backup path also contains a faulty MR, then 2 things must occur. First, the Node is marked as faulty, and not just a single MR fault. Second, the path-blocked process must be initiated, so that the previous MRs can be freed up,

and the packet can attempt to take a different path.

5.2.1.4 ACK

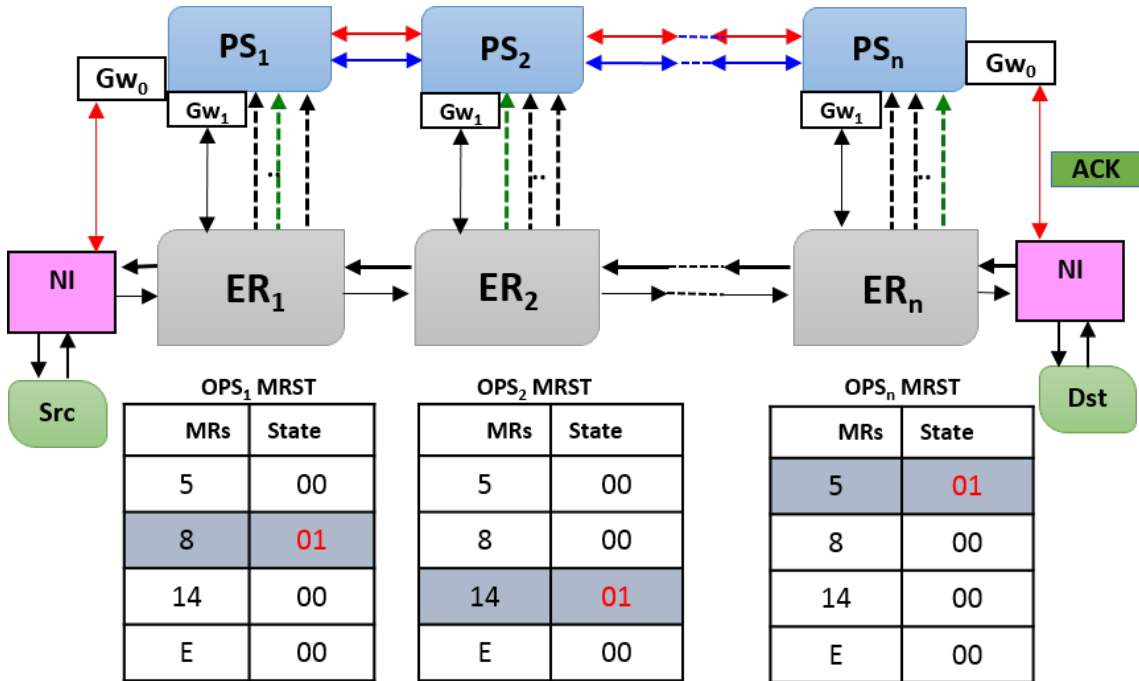


Figure 5.4: ACK phase.

As previously stated, when the path setup packet reaches the destination node successfully, the ACK process begins. Most works send the ACK backwards through the ECN, but in accordance with [25], our research group sends it through the optical domain from the destination node to the source node. This requires some additional hardware, but removes the electrical transmission completely, and thus a packet is able to complete its process faster. This process can be seen in figure 5.4. Once the source node receives the ACK signal, the payload transmission can begin.

5.2.1.5 Payload Transmission

Once the ACK signal arrives at the source node, then the payload gets prepped for transmission and is sent directly from the network interface into the optical layer via some encoding steps and the data modulators. The light follows the pre-made path according to the reserved MRs, and arrives at the photodetectors in the

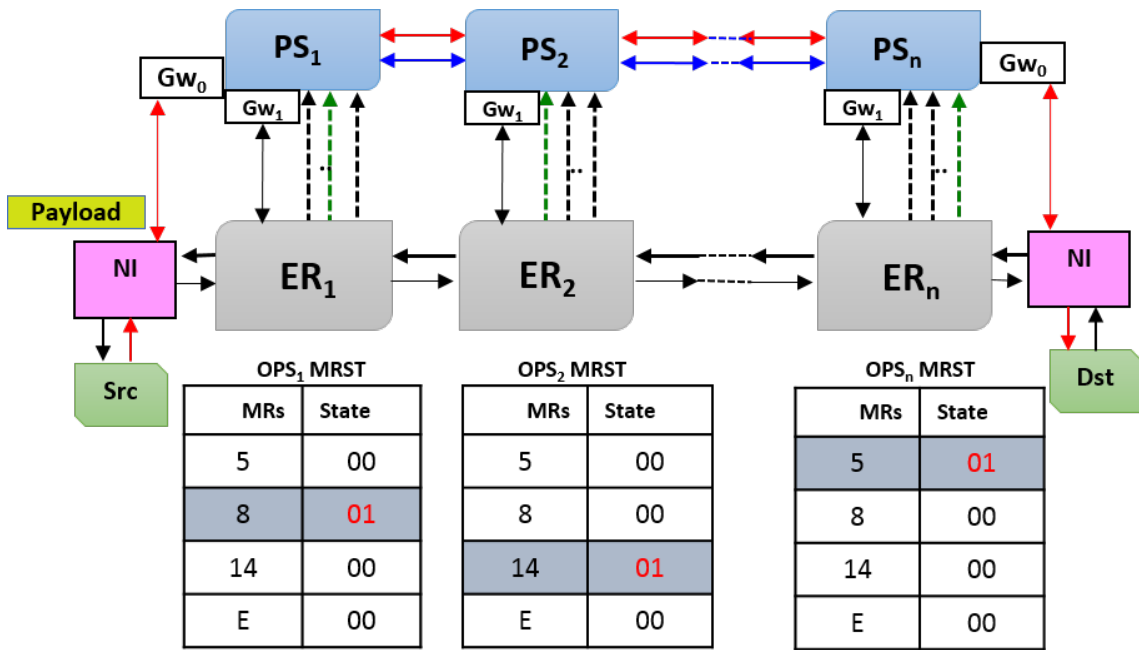


Figure 5.5: Payload transmission.

destination node, as can be seen in figure 5.5. The destination node converts it to electrical data, and then issues a Teardown Packet.

5.2.1.6 Teardown

The final process is the Teardown step, as shown in lines 26 – 31 of Algorithm 2. When the entire payload is completely received, then the MRs need to be released so that future transmissions can use the ports. The source node sends a Teardown packet to the destination after a predetermined number of cycles. Figure 5.6 shows the process. First, the source’s NI sends the electrical Teardown packet (TD) to the first electronic router ER_1 . The electronic router determines the MR that is used in the MRST. The corresponding MR (8 in this example) is then released by being set to “00” (“free”). Once the MRST is set to free, then the electrical “OFF” signal is sent to the MR itself. The electrical router then communicates to the NI, and the NI sends the tear-down in the optical domain to the next relevant node. This node performs a similar action, and the tear-down propagates through the path one node at a time. Eventually, all of the nodes have their resources released, and the message’s life cycle is complete.

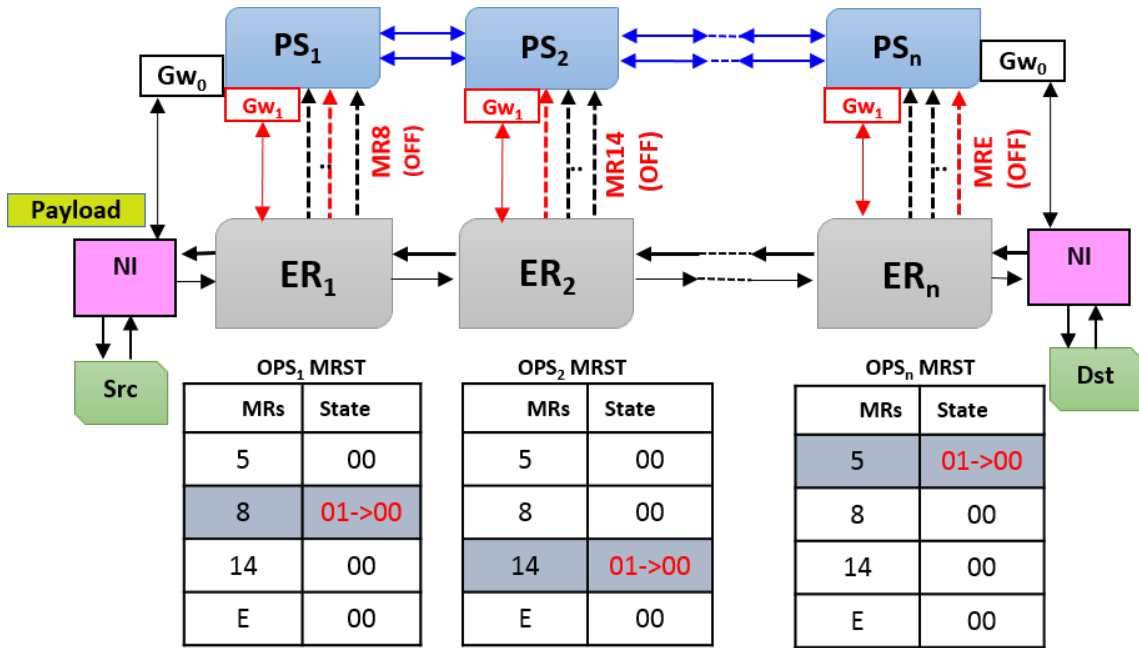


Figure 5.6: Tear-down phase.

5.2.2 Advantages of the Proposed Path Configuration Algorithm

The proposed path configuration is based off of previous works [23, 24, 30, 46, 92, 100–102]. One major difference is the use of a faulty state. We also utilize switches with two MRCTs, so that it can be used with fault tolerant switches, and provide the fault check. One thing that is unique to the research groups version is the use of optical control signals.

Conventional path-setup algorithms use the ECN for the ACK and teardown signals. This means that at each hop, each message must be buffered, the next-hop must be computed and arbitration has to occur. Additionally, the conventional algorithms are not able to notice faulty MRs or utilize fault tolerant switches. With the proposed algorithm, the ACK, Teardown, and Path Blocked packets are carried via the PCN. As a consequence, the ETE latency can be significantly reduced in addition to the dynamic energy saving that can be achieved. In addition, we considerably decrease the latency caused by the path blocking that requires several cycles for the path dropping and the new PSCP generation. Another key feature of the proposed path setup algorithm is the efficiency of the ECN resources' utilization.

By moving the acknowledgment signals to the upper layer, we can reduce the buffer depth to only 2 slots, since half of the network traffic is eliminated. This reduction is a key factor to design a light-weight router, highly optimized for latency and energy.

5.3 Evaluation

5.3.1 Methodology and Assumptions

We evaluated the proposed FT-PHENIC system using a modified version of PhoenixSim Simulator, which is developed in the OMNeT++ simulation environment [1]. The simulator incorporates detailed physical models of basic photonic building blocks such as waveguides, modulators, photodetectors, and switches. Electronic energy performance is based on the ORION simulator [2]. We evaluate the bandwidth performance and energy consumption for 16, 64 and 256 core systems. We compare the performance of the proposed FT-PHENIC systems with the baseline PHENIC [29], and the system using the algorithm proposed by Xiang et al. [43]. Xiang’s network was chosen over other typical systems [61, 72, 73, 103], because it uses some form of fault tolerance, and most of the conventional systems’ fault results would mimic the baseline PHENIC.

For benchmarks, we used *Random Uniform* and *Bitreverse* traffic patterns. *Random Uniform* traffic is a communication pattern where the destinations are randomly and uniformly selected each time a new communication occurs. In *Bitreverse*, each node sends messages to the complement node of its ID; thus, resulting in very long communications to observe the scalability of the proposed system. Tables 5.1 and 5.2 show the system and energy configuration parameters, respectively. We also used Fast Fourier Transform (FFT) and DataFlow as two realistic benchmarks. For the fault related data, we disabled a certain number of MRs at random, and recorded the data. To get better results, we would run each system at each fault rate several times, and then averaged each test’s total energy, average bandwidth, and average latency. Currently, the MR is disabled for the whole test, and thus models either a permanent or intermittent fault. Dealing with passband shift or temporary overheating of an MR is outside of the scope of this chapter, and thermal variation

Table 5.1: Configuration parameters.

Network Configuration	Value
Process technology	32 nm
Number of tiles	256, 64, or 16
Chip area (equally divided amongst tiles)	400 mm^2
Core frequency	2.5 GHz
Electronic Control frequency	1 GHz
Power Model	Orion 2.0
Buffer Depth	2
Message size	2 kb
Simulation time	10 ms (25×10^8 cycles)

Table 5.2: Photonic communication network energy parameters [2]

Network Configuration	Value
Datarate (per wavelength)	2.5 GB/s
MRs dynamic energy	375 fJ/bit
MRs static energy	400 μ W
Modulators dynamic energy	25 fJ/bit
Modulators static energy	30 μ W
Photodetector energy	50 fJ/bit
MRs static thermal tuning	1 μ W/ring

will be targeted in the next chapter. The fault rates were chosen to span from 0 to 30% due to the fact that at this point, all of the tested networks were in deadlock or totally crashed. Fault detection was not handled in the scope of this thesis, and the network just knew the locations upon injection.

5.3.2 Complexity Evaluation

In this section we evaluate the complexity of the proposed system against the two other architectures. The evaluation considers the number of used rings and the resulting static thermal tuning. The number of used MRs is given by equation 5.3.1, where $Mod/Detc_{(ring)}$ is the number of rings required to modulate/detect the payload signal. $Switch_{(ring)}$ is the number of rings required for the photonic switch to route the optical data. Finally, the $ACKs$ is the number of rings required to handle

the acknowledgment signal.

$$Total_{(ring)} = Mod/Detc_{(ring)} + Switch_{(ring)} + ACKs_{(ring)} \quad (5.3.1)$$

Tables 5.3 and 5.4 show the comparison results for 64 and 256 core system, respectively.

Table 5.3: Ring requirement and static power consumption results for 64-core systems.

	FT-PHENIC	PHENIC	Xiang
Mod/Detc	64	64	64
Switch	1152	1152	1600
ACKs	640	640	-
Redundant MRs	384	-	-
Total Rings	2240	1856	1664
Static Power(mW)	44	37	33

tively. We can see that the optimized networks have the lowest number of rings. In fact, this type of network is even more sensitive to MR faults as each MR is critical for the functionality of the node. In addition, with minimal number of rings, the resulting insertion loss is lower than the fault tolerant design. For the proposed FT-PHENIC system, the additional rings are used for acknowledgment signals and for fault-tolerance, which are not considered by the other networks, . This increase in rings can reach 33% when compared to the optimized crossbar and the baseline PHENIC system.

Table 5.4: Ring requirement and static power consumption comparison results for 256-core systems.

	FT-PHENIC	PHENIC	Xiang
Mod/Detc	256	256	256
Switch	4608	4608	6400
ACKs	2560	2560	-
Redundant MRs	1536	-	-
Total Rings	8960	7424	6656
Static Power(mW)	179	149	133

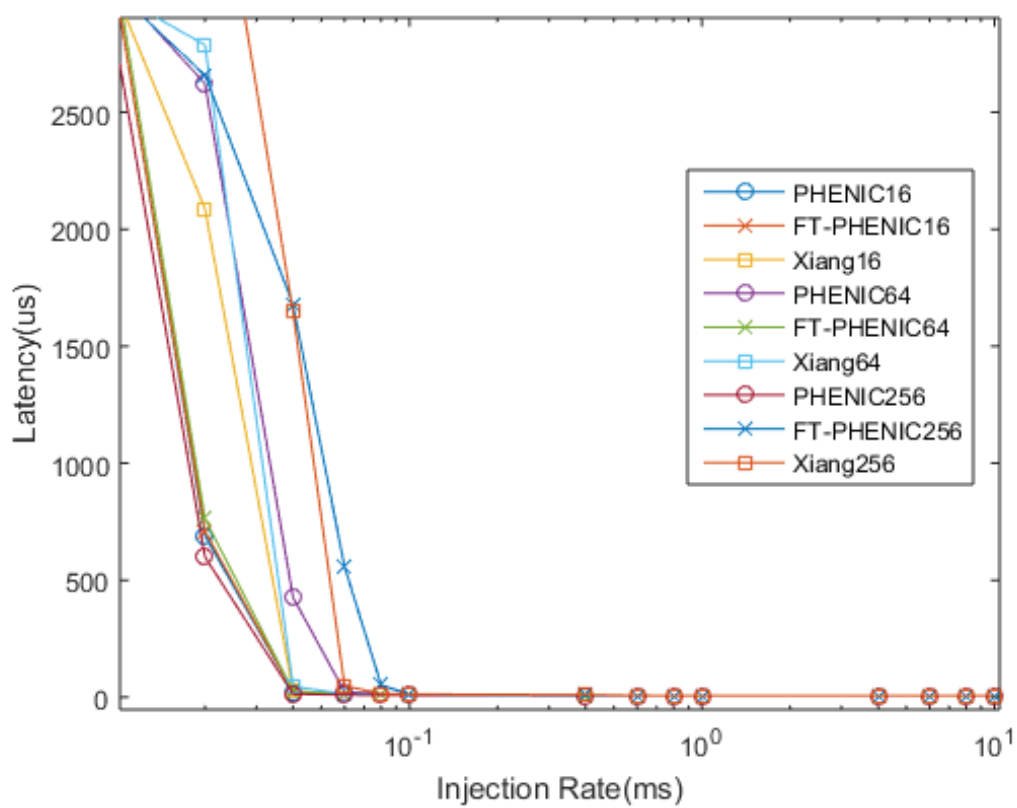


Figure 5.7: Overall latency comparison results of all systems under random uniform traffic for various packet injection rates.

5.3.3 Latency Evaluation

5.3.3.1 Latency at Different Packet Injection Rates

Understanding the cost of implementing the fault tolerance mechanisms is important, so we will first compare the networks' fault-free performance. Figure 5.7 shows the overall average latency. We can see that for zero-load latency, all networks behave in the same way. Near saturation, PHENIC shows more flexibility and scalability in 256 cores when compared to the other networks. This is because it was optimized for performance, and the other networks implement fault-avoidance. These fault-avoidance mechanisms require extra time for checking, and thus the networks that include them have a higher initial latency. For the 64-core configuration, the crossbar-based system slightly outperforms both PHENIC systems in terms of latency. This can be explained by the use of Optical-to-Electronic conversion of the *Teardown* which affects the overall latency of small networks. Performing simulation at the saturation region shows us which networks can better handle small increases in the injection rate.

5.3.3.2 Latency at Different Fault Injection Rates

The latency is heavily affected by the failure rate of MRs, and as more faults are injected into the system, the latency increases until the whole system fails. This is primarily caused by failed path setup. Figure 5.8 shows the results of the latency test when adding in varying amounts of MR failures. The FT-PHENIC demonstrates its ability to withstand MR failures over all other systems. As more faults are injected, some systems fail to complete the simulation due to crashing. We can also see that the non-fault-tolerant networks seem to tolerate 1% faults because of each node having some non-critical MRs for the ACK and tear down, which would simply make the node look reserved. Additionally, this figure shows average latency, so some of the packets may not have been delivered when the simulation ended and resulted in latencies in the hundreds of milliseconds. At 30% faults, all systems have in crashed.

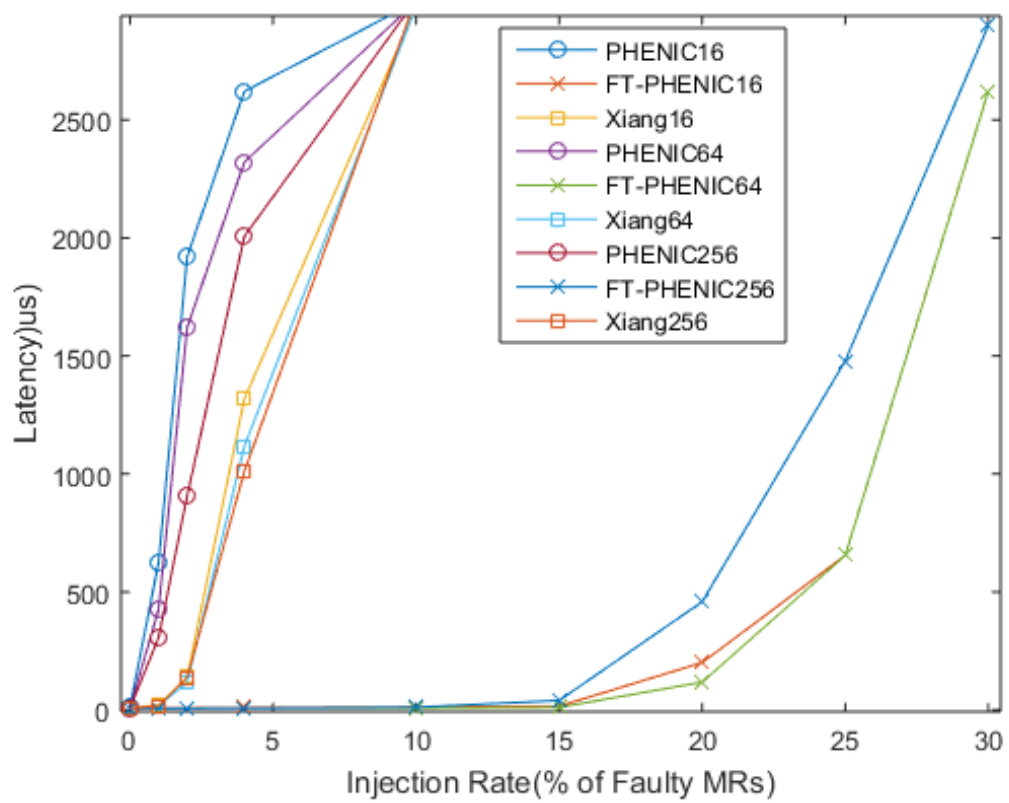


Figure 5.8: Latency results of each system as faults are introduced.

5.3.4 Bandwidth Evaluation

5.3.4.1 Bandwidth at Different Packet Injection Rates

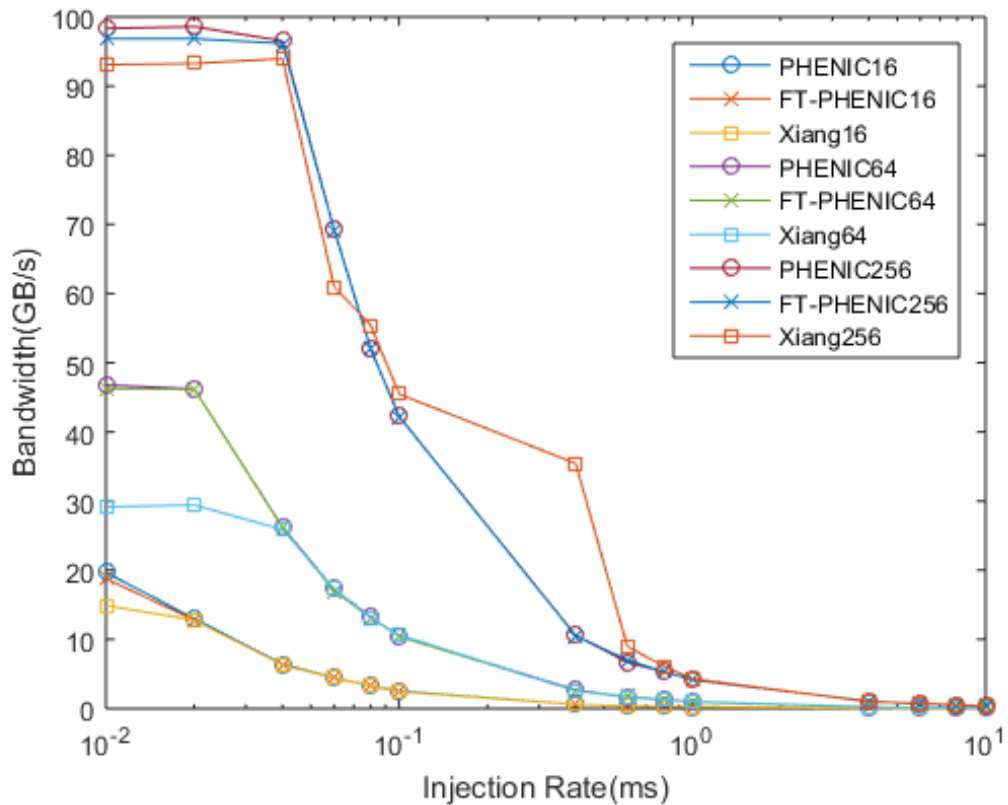


Figure 5.9: Bandwidth comparison results under random uniform traffic.

For the achieved bandwidth, Fig. 5.9 shows that the bandwidth of FT-PHENIC is similar to the baseline, which is slightly improved over the Xiang System. This behavior is observed for 16, 64 and 256 core systems. The latency increase caused by failed MRs will in turn cause the bandwidth to decrease.

5.3.4.2 Bandwidth at Different Fault Injection Rates

The effects of the failures on the bandwidth can be seen in Fig. 5.10. As with the latency, only FT-PHENIC and Xiang's algorithm show any tolerance to faults, with FT-PHENIC outperforming Xiang's algorithm.

From these results we can see that FT-PHENIC sacrifices a little performance for a large benefit in fault tolerance, when compared to its predecessor. The OHFT

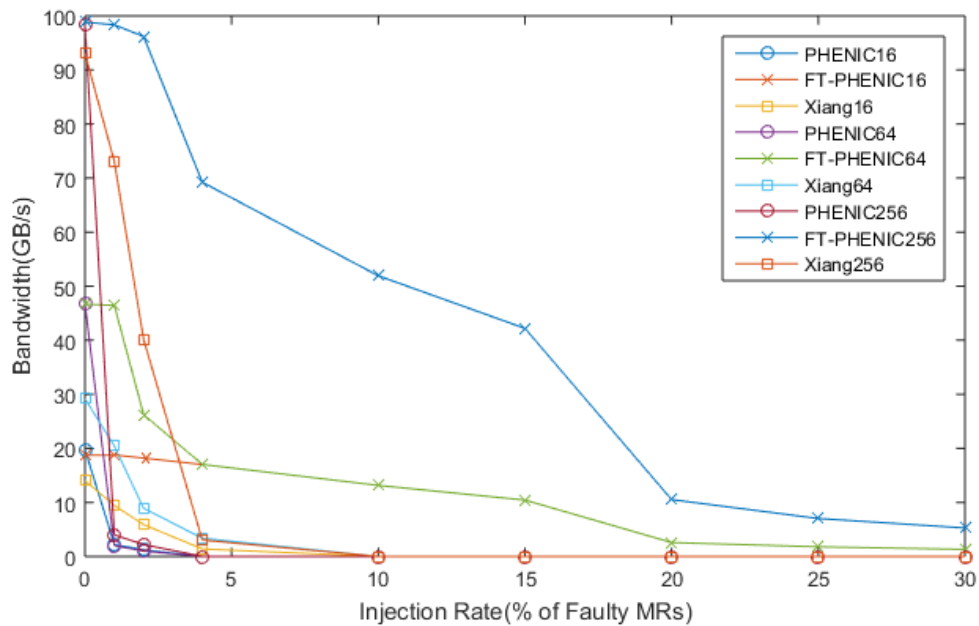


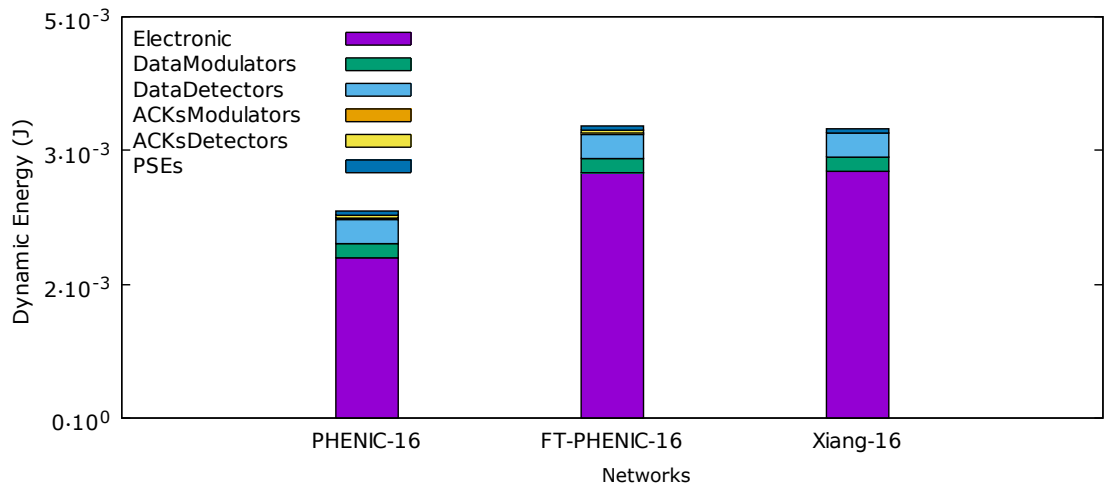
Figure 5.10: Bandwidth comparison results as faults are introduced.

and FT-PHENIC combination still had more scalability than Xiang's algorithm in the end, and had a higher starting bandwidth, and was able to tolerate a higher amount of MR failures, and maintains functionality until around 20%.

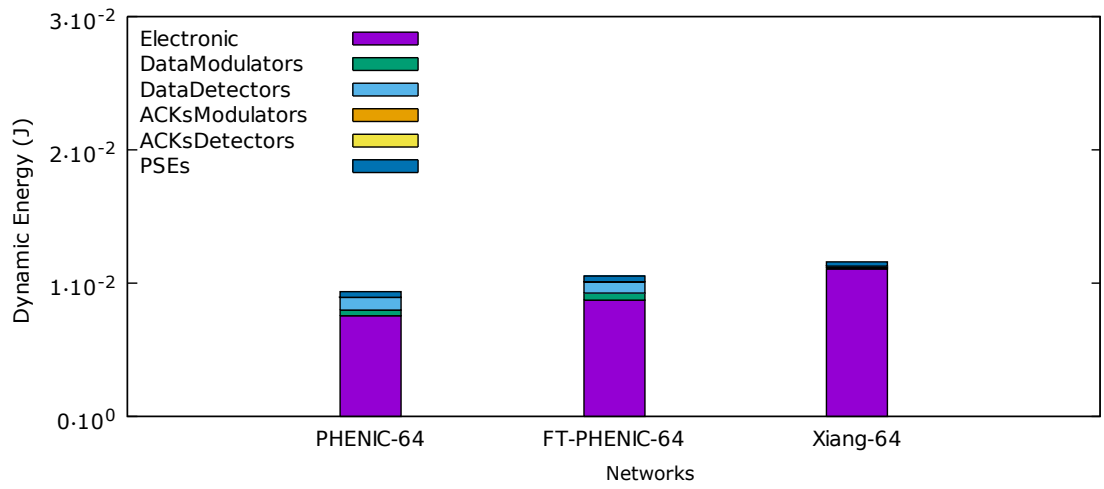
5.3.5 Energy Evaluation

5.3.5.1 Energy Breakdown

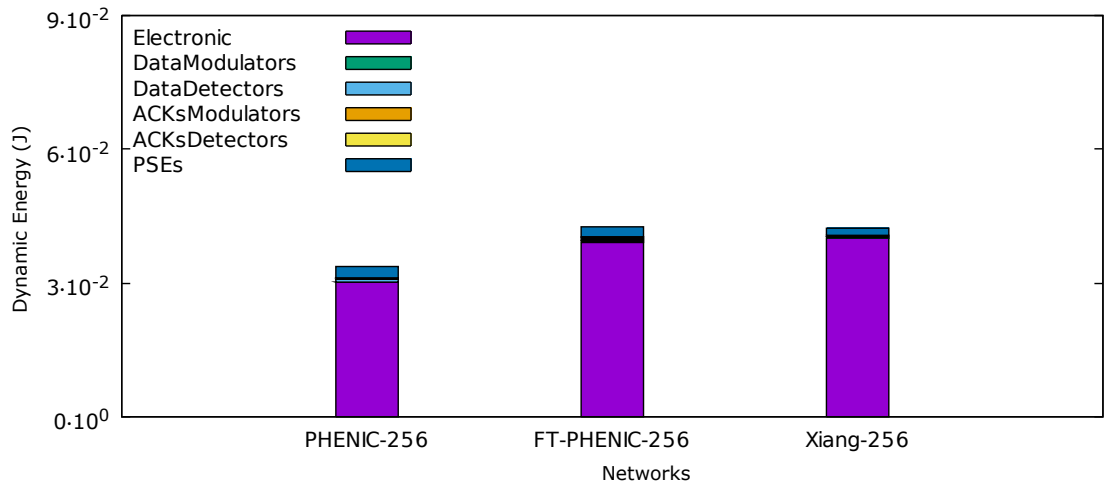
Figure 5.11 (a) - (c) shows the energy breakdown for 16, 64, and 256 core systems. Compared to other networks where the electronic energy reaches 90% of the total energy, PHENIC shows a more balanced energy distribution between the photonic and electronic networks, especially in the 64-core system case. This is despite the fact that the electronic power is still high with 70% of the total system energy. FT-PHENIC does consume more energy, which is one of the sacrifices made for the fault tolerance. From these results, we can see that FT-PHENIC is more reliable than the other systems. We conclude that the obtained improvement by FT-PHENIC is the result of the association of three main factors together: (1) the non-blocking switch supporting optical acknowledgment signals, (2) the light-weight router with reduced



(a)



(b)



(c)

Figure 5.11: Total energy breakdown comparison under random uniform traffic near-saturation:(a) 16-core systems, (b) 64-core systems, (c) 256-core systems.

buffer size, (3) and the fault-tolerant path setup algorithm that adopts hybrid-switching inside the photonic switch. Its drawbacks, when compared to PHENIC, are due to the added redundant MRs, and the FTTP algorithm.

5.3.5.2 Total Energy and Energy Efficiency

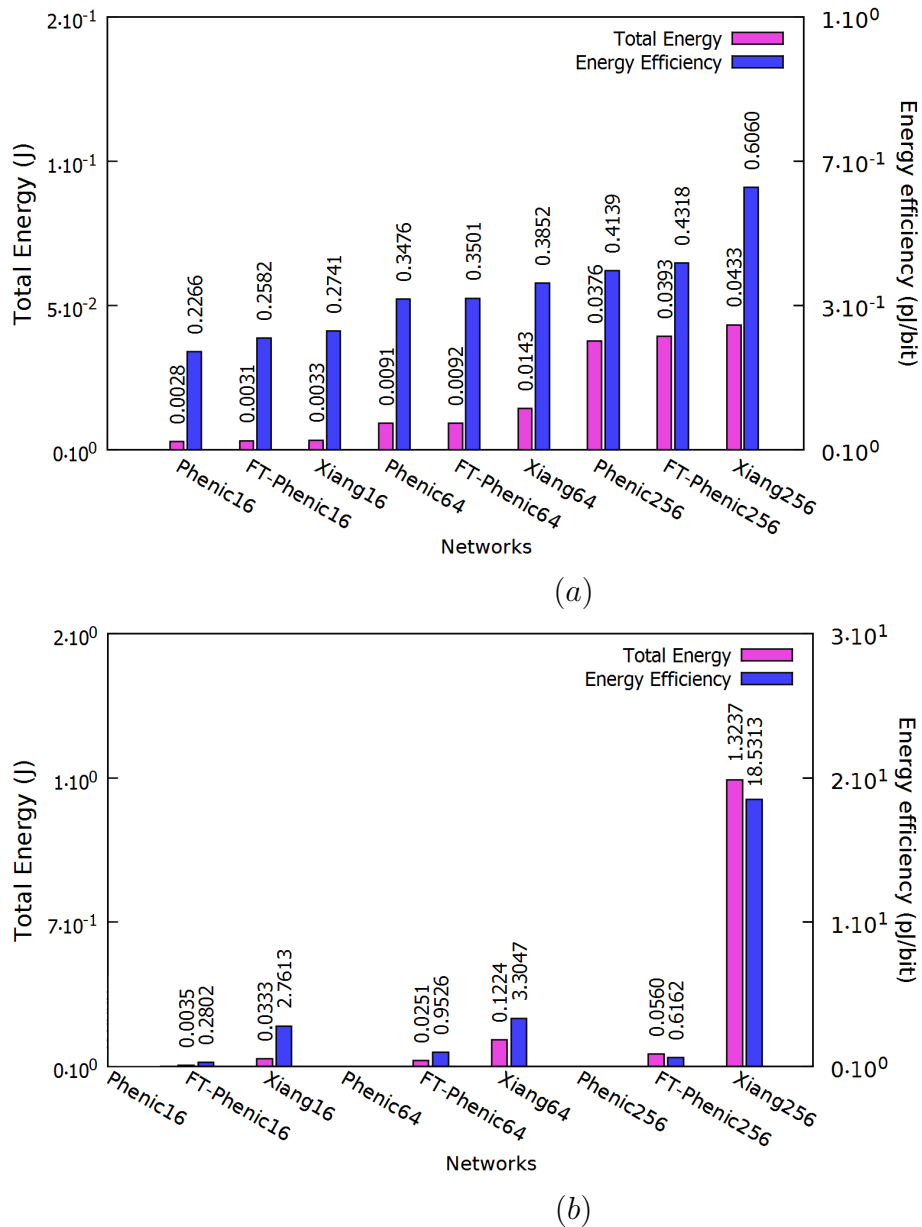
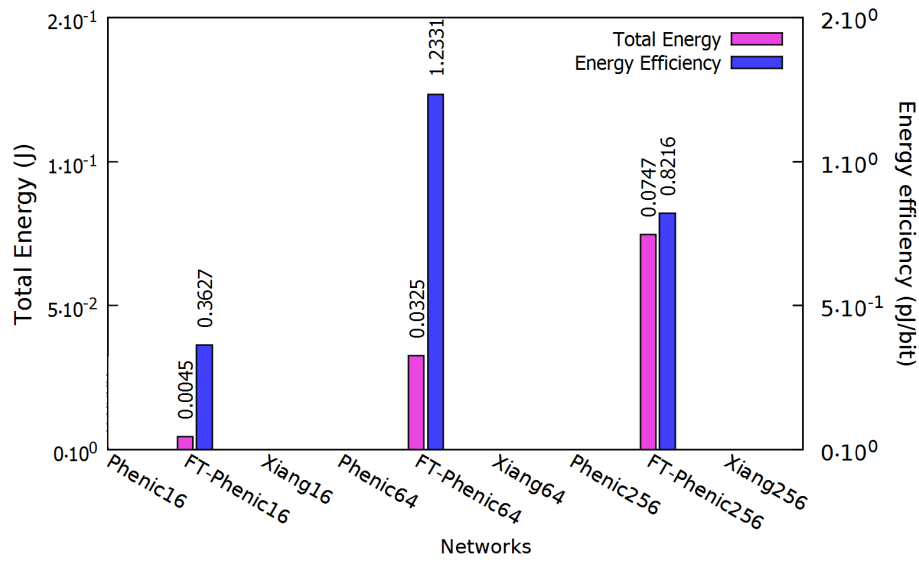
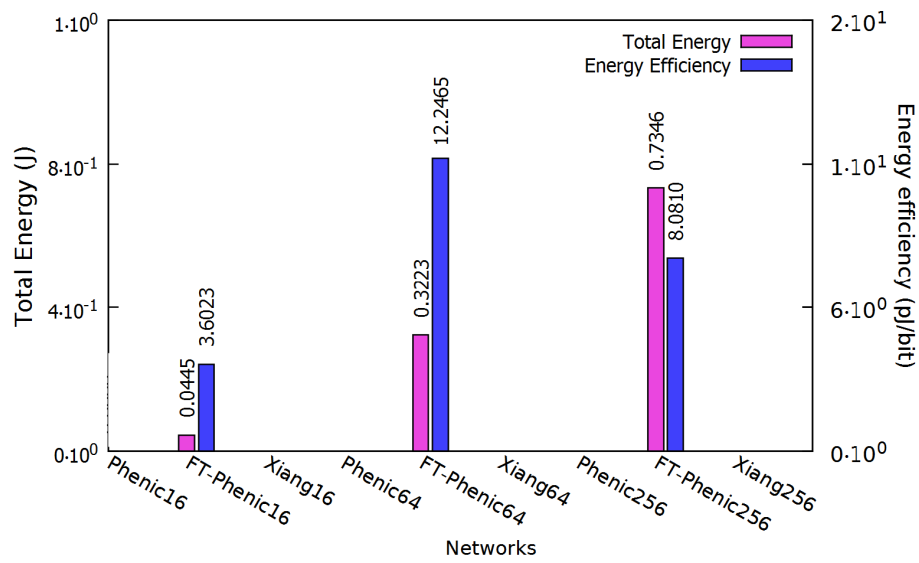


Figure 5.12: Total energy and energy efficiency comparison results under random uniform traffic near-saturation with (a) 0% and (b) 4% faulty MRs.

Figures 5.12 and 5.13 show the energy and energy efficiency as various amounts



(a)



(b)

Figure 5.13: Total energy and energy efficiency comparison results under random uniform traffic near-saturation with (a) 10% and (b) 30% faulty MRs.

of faults are injected into the 16, 64, and 256 core systems. As can be seen in the figures, some systems were not able to complete simulation, and so their energy is marked as 0. The energy efficiency is the ratio of the total energy to the total number of bits that were transmitted. As faults are injected, all systems require more energy. The extra energy comes from the extra run time, and incapability to successfully transmit messages.

We can see that even at 4% faults, FT-PHENIC has the lowest amount of energy consumption for all network sizes. This is because its runtime is only minimally affected when only 4% of MRs are faulty. If we look at the efficiency, at 0% faults, the most efficient is the baseline, which has no adaptive algorithm, and no fault tolerance mechanisms. However, as faults are injected, the efficiency quickly skyrockets towards infinity. We can also see that the one that maintained the best efficiency as faults are introduced was FT-PHENIC. At 10% faults, FT-PHENIC still had an approximate efficiency of 1 for the 256 core system, when all other systems had hit infinity.

5.4 Chapter Summary

This chapter focused on the fault-tolerant path configuration algorithm, named FTTP. It contains details about the proposed algorithm and the evaluation results. The main idea of the FTTP can be applied to any EA-PNoC path setup algorithm, but we applied it to the advanced contention-aware path configuration algorithm that was previously proposed by the Adaptive Systems Laboratory. From the obtained results, we conclude that the proposed FT-PHENIC architecture is well-equipped to handle faulty MRs. These capabilities arise from a combination of the FTTP algorithm and the FTTDOR switch. It gains this capability with minimal performance and power drawbacks. The next chapter attempts to solve the problems associated with temperature variation by routing around nodes that are too hot.

Chapter 6

Strain-Aware Routing Algorithm

6.1 Introduction

PNoCs are highly sensitive to temperature variation. As the MRs and other optical devices heat up, they may have their effective wavelength shifted, and the sender and receiver may not match up. As we have stated in Chapter 3, many different authors have come up with different solutions for this problem. To the best of our knowledge, most of the existing solutions proposed so far have only been applied to electrical circuits. Others have not been applied to fault-tolerant NoCs. In this chapter, we will provide a power estimation scheme, and a routing algorithm that makes use of said scheme to avoid the nodes which have heated up from being overused. This chapter relates to the work published in [104].

6.2 Power Estimation

We intend to create a fault-tolerant power-aware Hybrid Network-on-Chip. To do this, we use the previously proposed FT-PHENIC architecture [46], which tolerates MR faults, and then implement traffic-aware modules into it, so that the data can be used for routing decisions in the future. The FT-PHENIC system, shown in Fig. 6.1, is a mesh-based topology and uses minimal redundancy to assure accuracy of the packet transmission even after faulty MRs are detected. The system uses a Stall-Go mechanism for flow-control, and a Matrix-arbiter as a scheduling tech-

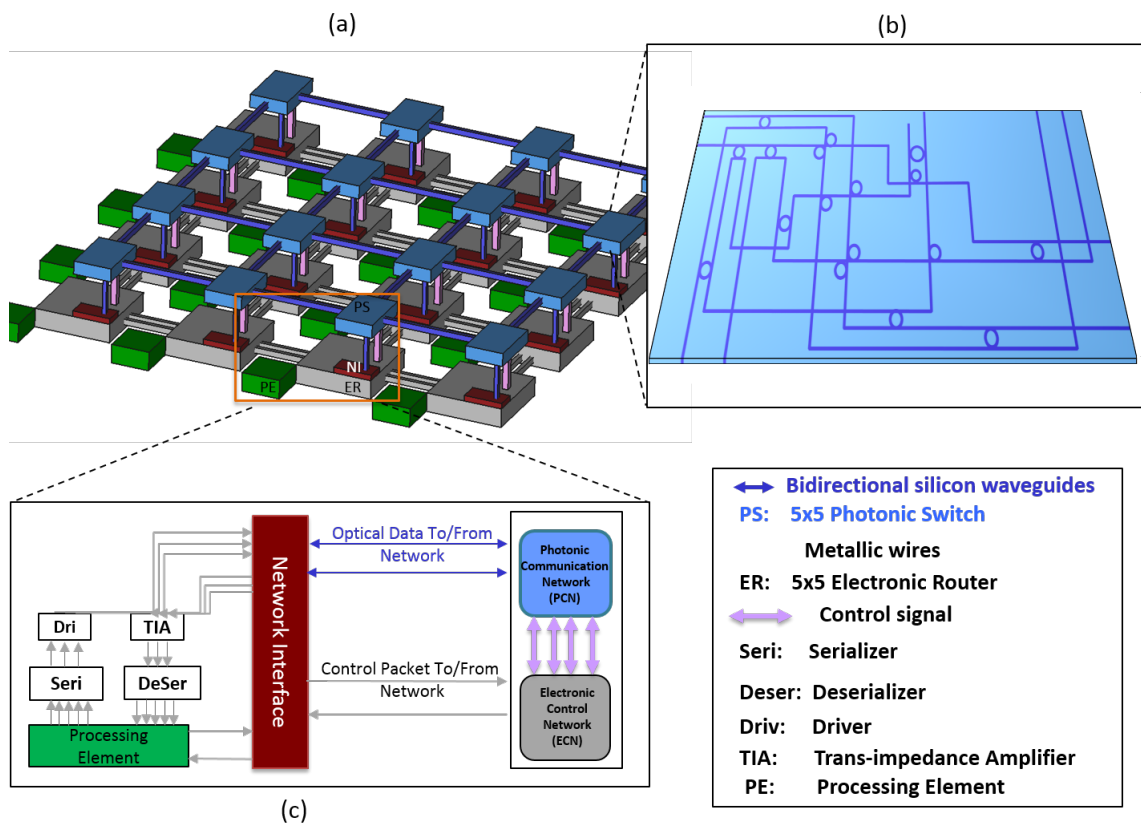


Figure 6.1: FT-PHENIC system architecture. (a) 4x4 mesh-based system, (b) 5x5 non-blocking photonic switch, (c) Unified tile including PE, NI and control modules.

nique [46]. FT-PHENIC is also based on a microring fault-resilient photonic router (FTTDOR) [30] and an adaptive path-configuration and routing algorithm. As illustrated in Fig. 6.1, the system consists of a Photonic Communication Network (PCN), used for data communication, and an Electronic Control Network (ECN), used for path configuration and routing. Each Processing Element (PE) is connected to a local electrical router and also connected to the corresponding gateway (modulator/detector) in the PCN [29]. Messages generated by the PEs are separated into control signals and payload signals. Control signals are routed in the ECN and used for path configuration and routing. The payloads are converted to optical data and transmitted on the PCN.

For a majority of faults, the design of the FTTDOR switch allows for an alternate, slightly less power efficient route. In fact, the backup route is less power-efficient because the packets travel across more waveguide distance, go through more active MRs, and cross more waveguides. However, the switch still maintains all of its functionality. Because backup routes are only intended for use in the switches in which faults have occurred, the additional loss will have minimal effect on the messages' signal strength across the whole network. The overall Architecture can be seen in Fig. 6.3.

The new additions to the FT-PHENIC architecture are the power estimation module inside the electronic router and changes to the arbiter module. The power estimation module takes the traffic information, which is monitored in the arbiter, and performs some simple calculations. In the next section of this chapter, this module will output to the nodes up to 2 hops away. As mentioned at the start of this paragraph, the arbiter has to be changed to monitor the traffic. The new arbiter architecture can be seen in Fig. 6.2.

6.2.1 Power Estimation Calculation

We use the power model from [63], but modified it to be focused on the photonic layer. The power model is given in equation 6.2.1. It is the sum of the power consumed by the modulators, detectors, and photonic switching elements. The static powers of the optical components have all been lumped into one term, and

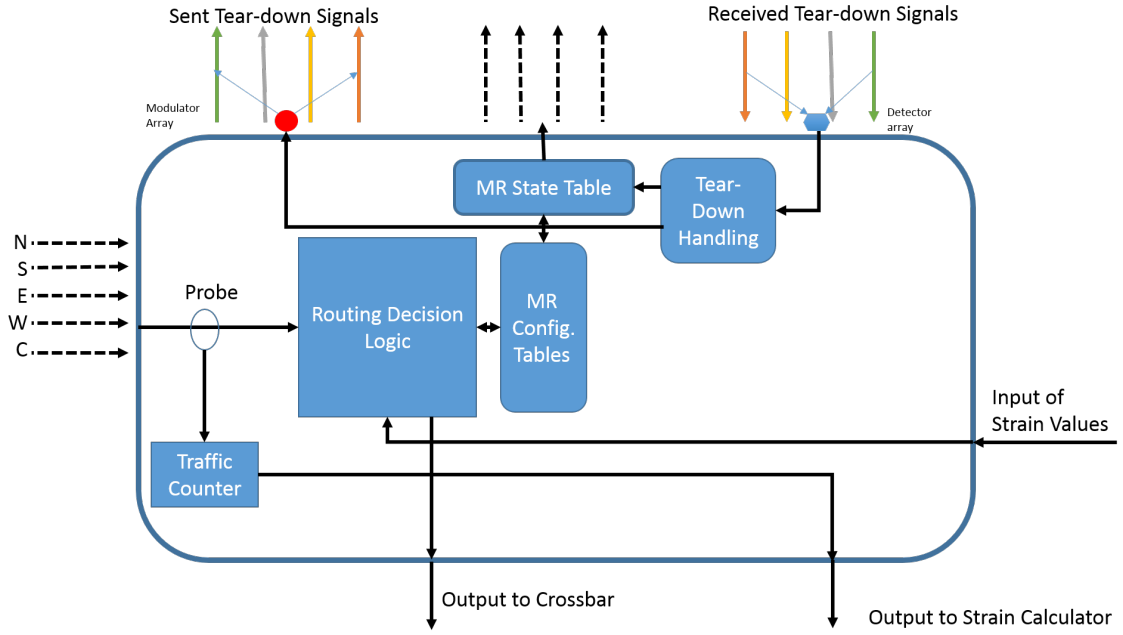


Figure 6.2: Arbiter architecture.

each component's dynamic power has to be calculated.

$$Power_{Tot}(i) = DPower_{Mod}(i) + DPower_{Det}(i) + DPower_{PSE}(i) + Power_{Static} \quad (6.2.1)$$

The power consumption of the modulators and detectors can be estimated based on the number of flits that a node sends or receives, in a similar fashion to how Yang et al. [63] estimated their cross bar energy based around messages received. It is modeled as a power coefficient, C , multiplied by the sum of the switching activity, S or D or G , of the component, for each flit that interacts with that component, all divided by the time frame T . The resulting equations are equations 6.2.2 - 6.2.4.

$$DPower_{Mod}(i) = \left(C_{Mod} \times \sum_{j=1}^{N_{Mod}} S(j) \right) / T \quad (6.2.2)$$

$$DPower_{Det}(i) = \left(C_{Det} \times \sum_{j=1}^{N_{Det}} D(j) \right) / T \quad (6.2.3)$$

$$DPower_{PSE}(i) = \left(C_{PSE} \times \sum_{j=1}^{N_{PSE}} G(j) \right) / T \quad (6.2.4)$$

The activity of the detectors and the modulators are heavily based around the number of flits which are respectively sent from and consumed at the corresponding node, respectively. The PSE's power is more static, but still has an element based on the traffic that goes through the node. All of this results in the total power equation being modified into equation 6.2.5.

$$Power_{Tot}(i) = \left(C_{Mod} \times \sum_{j=1}^{N_{Mod}} S(j) + C_{Det} \times \sum_{j=1}^{N_{Det}} D(j) + C_{PSE} \times \sum_{j=1}^{N_{PSE}} G(j) \right) / T + Power_{Static} \quad (6.2.5)$$

The calculation should be simple enough that it allows for a fast calculation, but sacrifices some of the accuracy, which should not be critical for routing decisions, especially if the two directions have such similar power values, then making the wrong decision shouldn't be a critical difference. Each node is responsible for calculating their own power estimate. The key to making the power estimate work lies in the accuracy of the power coefficients. We get some initial values from a simple simulation. The static power is taken for each network size. The C_{Mod} , C_{Det} , and C_{PSE} values are calculated based on the dynamic power of each component and the number of packets transmitted. For example, if the total dynamic energy of all the modulators is 30, and it has 15 packets, then the C_{Mod} value will be 2. A similar calculation will take place for C_{Det} , using the detector's dynamic energy. To calculate C_{PSE} , we also have to factor in the number of hops, because the data does

not only pass through one node, whereas the data only gets modulated and detected at one node for each process. For C_{PSE} , we take the total dynamic energy of the system, divide it by the number of packets times the average number of hops. With this strategy, we hope to get accurate enough results that can be used for routing decisions.

6.3 LASA Algorithm

We intend to create a fault-tolerant thermal-aware Hybrid Network-on-Chip. To do this, we use the previously presented FT-PHENIC architecture [46], which tolerates MR faults, and implement a thermal-aware routing algorithm into it.

The routing algorithm for the new SAFT-PHENIC network will be the newly proposed Look Ahead Strain Aware (LASA) routing algorithm. The overall architecture will stay almost completely the same, but We have to include an extra module to calculate the strain values, and extra connections to transmit them to the necessary nodes, as can be seen in Fig. 6.3.

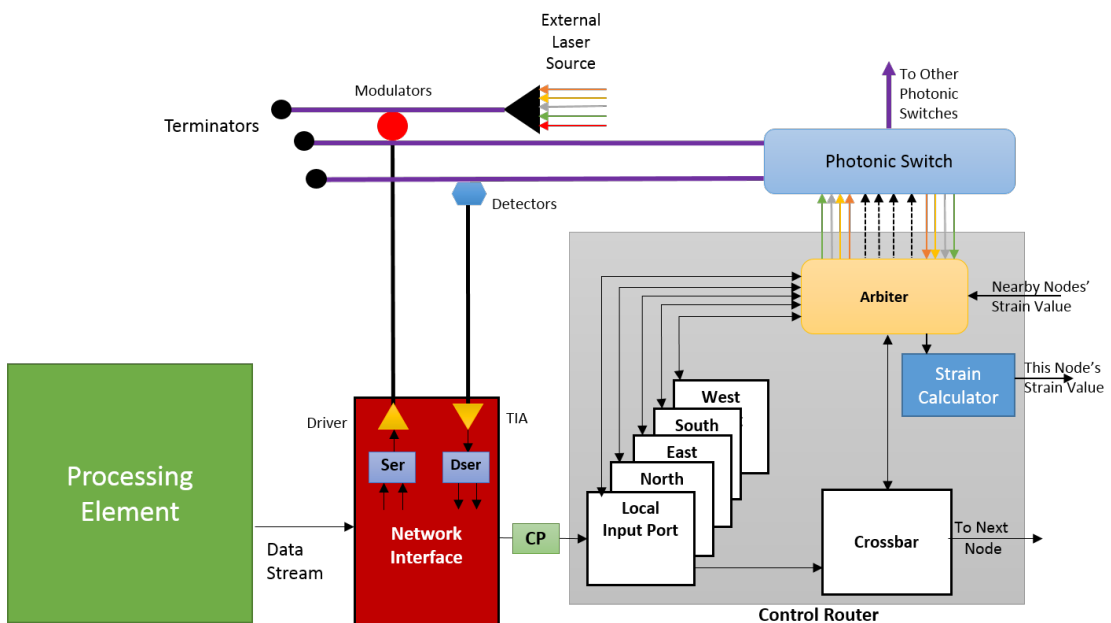


Figure 6.3: SAFT-PHENIC node architecture.

6.3.1 Routing

The Look Ahead Stress Aware algorithm will use information from neighboring nodes in order to avoid hot areas, and areas with congestion. The idea is to use look ahead routing data to avoid making hot spots even hotter. The way in which we can achieve this is with an algorithm inspired by [63]. The algorithm is presented in Algorithm 3.

Breaking down Algorithm 3 can be quite simple. A packet starts at lines 13-24, where the algorithm tries to determine which direction to use by first determining the 2 possible minimal directions, and then determining which direction has the lower strain value. The strain calculation will be defined later, in section 6.3.2. The packet will continue determining the lower strain node until either the X or the Y value matches the destination, meaning that there is only one minimal path. When there is only one minimal path, lines 3-10, the packet simply travels the minimal path. Lines 1 and 2 handle the case where the current node is the destination, and the packet must be sent to the core. The algorithm's flowchart can also be seen in Fig. 6.4. The majority of nodes will be handled by the rightmost diamond, where the algorithm checks the stress values. The other blocks handle the deterministic routing for when the current X or Y values equal the destination X or Y. This algorithm should avoid hotspots as much as it can, while maintaining a minimal path, which is critical to keeping the insertion loss to a reasonable level.

The conventional dimension order XY routing (Algorithm 4) is used for the conventional system and the original PHENIC system, but is ignorant of many aspects of the NoC. The proposed algorithm adds some complexity to the calculation, but should pay off with reduced peak power, and fault avoidance.

Some example routing cases can be seen in Fig. 6.5. Figure 6.5 (a) was composed by setting the source and destination in opposite corners and assigning all of the middle values randomly. The solid line shows the route taken by the LASA algorithm, and the dotted line is the dimension order XY routing (DOR-XY). The LASA algorithm avoids all of the high strain nodes, which are the nodes which have consumed more power. The average strain value of the nodes that LASA passes through is 0.23, while the DOR-XY path has an average of 0.34. Case (b) is the

Algorithm 3: Strain-aware routing algorithm.

```

// Destination address
Input:  $X_{dest}, Y_{dest}$ 
// Current node address
Input:  $X_{cur}, Y_{cur}$ 
// Strain status information
Input: STR-in
// Output Port
Output: Outport
// Compare Current Node to Destination Node
1 if  $(X_{dest} == X_{cur}) \&\& (Y_{dest} == Y_{cur})$  then
2 |   return Local;
3 else if  $(X_{dest} == X_{cur}) \&\& (Y_{dest} > Y_{cur})$  then
4 |   return North;
5 else if  $(X_{dest} == X_{cur}) \&\& (Y_{dest} < Y_{cur})$  then
6 |   return South;
7 else if  $(X_{dest} > X_{cur}) \&\& (Y_{dest} == Y_{cur})$  then
8 |   return East;
9 else if  $(X_{dest} < X_{cur}) \&\& (Y_{dest} == Y_{cur})$  then
10 |  return West;
11 else
12 |  return(StrainDecision());
    // StrainDecision()
    Data: XDir, YDir
13 if  $(X_{dest} > X_{cur})$  then
14 |   $XDir \leftarrow \text{East};$ 
15 else
16 |   $XDir \leftarrow \text{West};$ 
17 if  $(Y_{dest} > Y_{cur})$  then
18 |   $YDir \leftarrow \text{North};$ 
19 else
20 |   $YDir \leftarrow \text{South};$ 
21 if  $(STR - in[XDir] < STR - in[YDir])$  then
22 |  return XDir;
23 else
24 |  return YDir;

```

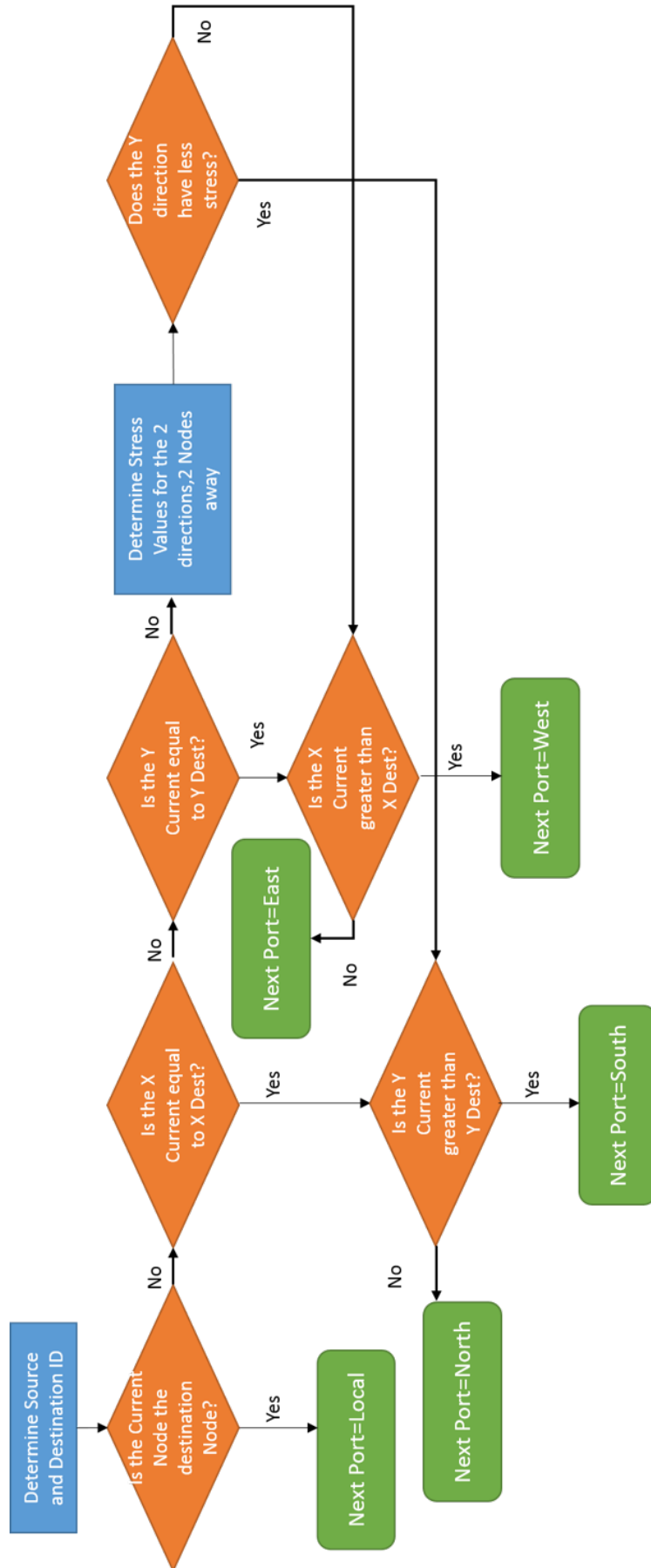


Figure 6.4: Flowchart of the LASA algorithm

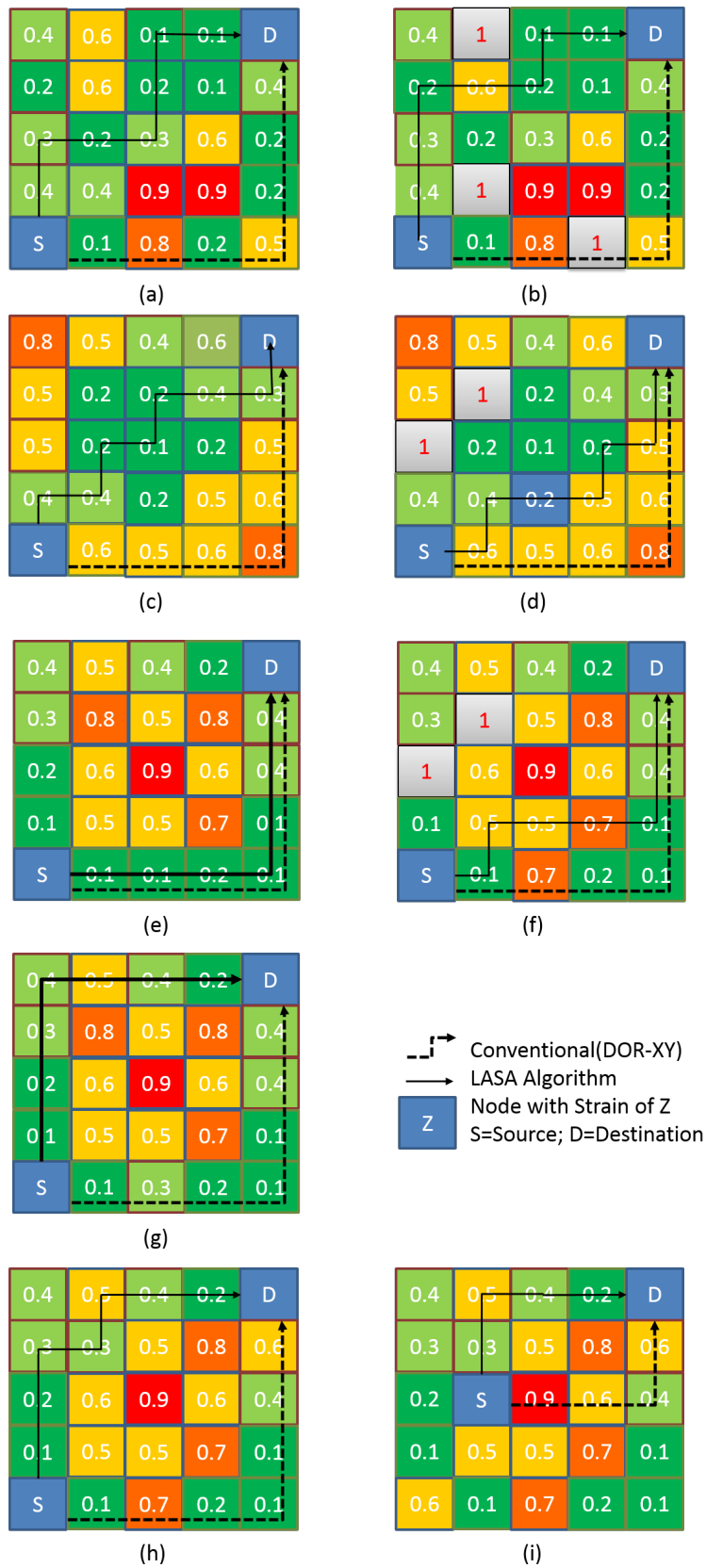


Figure 6.5: Example cases for strain values and how the two different routing algorithms react

Algorithm 4: DOR XY routing algorithm.

```

// Destination address
Input:  $X_{dest}, Y_{dest}$ 
// Current node address
Input:  $X_{cur}, Y_{cur}$ 
// Output Port
Output: Output
// Compare Current Node to Destination Node
1 if ( $X_{dest} == X_{cur}$ ) $\&\&$ ( $Y_{dest} == Y_{cur}$ ) then
2 |   return Local;
3 else if ( $X_{dest} == X_{cur}$ ) $\&\&$ ( $Y_{dest} > Y_{cur}$ ) then
4 |   return North;
5 else if ( $X_{dest} == X_{cur}$ ) $\&\&$ ( $Y_{dest} < Y_{cur}$ ) then
6 |   return South;
7 else if ( $X_{dest} > X_{cur}$ ) then
8 |   return East;
9 else
10 |  return West;

```

same as (a), but with each node having 10% chance of being faulty. Again, we can see that it avoids the faults and the high power nodes. As we can see, the DOR-XY still sends the packet through a faulty node and a high power node. The average for DOR-XY this time was 0.46 as opposed to LASA's 0.27. These averages symbolize the average power of the nodes that the message travels through, and thus their temperatures.

Figure 6.5 (c) uses a new set of values, with the higher strains being allocated to the outside. This creates a case which really highlights the LASA algorithm. This time, LASA has an average strain of 0.29, while the DOR-XY went through all of the hotspots, and had an average of 0.56. This is almost double, and yet again it travels through one of the 2 nodes with the highest strain. Again, 10% faults were applied to (c) and generated (d). The LASA algorithm was forced to avoid the faults, and again went through the center.

Figure 6.5 (e)-(i) are an example of a realistic strain map. The center values are greater than the outside ones. This is doubly true for temperatures, as traffic will be more likely to go through the middle, and the middle has less of a chance to remove the heat. We want to be clear that the strain will not really be double or triple from one node to the next, but we want to show that it is greater or less than nearby nodes, so one digit differences is good enough for demonstration. If we look at (e), we can see that both LASA and DOR-XY take the same route. If we change a single value, we get (g), in which LASA takes a completely different route.

One more small change gives us (h), which lets the packet route around the top corner. In (f), we get the result of applying some faulty nodes to (h). We can see that LASA successfully avoids the faults. In (i), we can see the result of moving the source node from (h). This is where we can see some large benefits from a realistic strain map. The DOR-XY ignores the strain values and goes straight through the 0.9 strain node in the middle. The LASA does as it was designed to do and routes to the edge, where the lower strain values are.

6.3.2 Strain

We calculate Strain based on the power consumption and number of faulty MRs in a node. We use the power model from the previous section, but have modified it to be focused on the dynamic power, because all nodes will have similar static power. We also use the max value from the next 2 nodes in that same direction. If the next node is the edge node, then the second node in that direction will automatically have a value of 0, and the first node's Strain value is prioritized.

$$Strain(i) = \begin{cases} Power_{Dyn}(i), & \text{if } N_{FaultyMRs} \leq 6 \\ 1, & \text{otherwise} \end{cases} \quad (6.3.6)$$

Equation 6.3.6 shows how we calculate the strain of node i . The first term, is the dynamic power of the node, which allows us to estimate how much thermal energy is being put into the node. The conditional statement is there to let the strain account for the faulty MRs. These MRs are ones with permanent or intermittent faults. In the case that the condition isn't met, then we assume that the node has a maximum strain value. This means that the node will likely be avoided.

The dynamic power model is given by equation 6.3.7. It is the sum of the dynamic power consumed by the modulators, detectors, and photonic switching elements.

$$Power_{Dyn}(i) = Power_{Mod}(i) + Power_{Det}(i) + Power_{PSE}(i) \quad (6.3.7)$$

The power consumption of the modulators and detectors can be estimated based on the number of flits that a node sends or receives, in a similar fashion to how [63] estimated their cross bar energy based around messages received. It is modeled as a power coefficient, C , multiplied by the sum of the switching activity, S or D or G , of the component, for each flit that interacts with that component, all divided by the time frame T . The resulting equations are labeled as equations 6.2.2 - 6.2.4.

The dynamic power of the detectors is based off of the amount of flits that are received at that node and the dynamic power of the modulators is heavily based off of the amount of packets generated at that node, but the PSE's dynamic power is a little more tricky. This involves sectioning off a portion to each node that the message passes through, or turns at. All of this results in the total power equation being modified into equation 6.3.8.

$$Power_{Dyn}(i) = \left(C_{Mod} \times \sum_{j=1}^{N_{Mod}} S(j) + C_{Det} \times \sum_{j=1}^{N_{Det}} D(j) + C_{PSE} \times \sum_{j=1}^{N_{PSE}} G(j) \right) / T \quad (6.3.8)$$

Each node will be responsible for calculating their own Power estimate, and strain value, and will send it to the nodes which require the information for routing decisions. This way, the routing decision can be made much more quickly. The key to making the strain value work is the power estimate. The key to making the power estimate work lies in the accuracy of the power coefficients.

$$Strain(i) = \begin{cases} \frac{1}{T} \left(C_{Mod} \times \sum_{j=1}^{N_{Mod}} S(j) + C_{Det} \times \sum_{j=1}^{N_{Det}} D(j) + C_{PSE} \times \sum_{j=1}^{N_{PSE}} G(j) \right), & \text{if } N_{FMR} \leq 6 \\ 1, & \text{otherwise} \end{cases} \quad (6.3.9)$$

As a final result, equation 6.3.9 comes out. The calculation should be simple enough that it allows for a fast calculation, but sacrifices some of the accuracy,

which should not be critical for routing decisions, especially if the two directions have such similar strain values, then making the wrong decision shouldn't be a critical difference.

6.4 Evaluation

6.4.1 Methodology

6.4.1.1 Power Estimate Accuracy Methodology

We simulated the proposed estimation scheme on FT-PHENIC using a modified version of PhoenixSim which is developed in the OMNeT++ simulation environment [1]. The simulator incorporates detailed physical models of basic photonic building blocks such as waveguides, modulators, photodetectors, and switches. Electronic energy performance is based on the ORION simulator, which is integrated with PhoenixSim. We evaluate the performance and energy consumption for 16, 64 and 256 core systems. The systems used thermal-ignorant task mapping.

For benchmarks, we used *Random Uniform* and *Bit-reverse* traffic patterns. *Random Uniform* traffic is a communication pattern where the destinations are randomly and uniformly selected each time a new communication occurs. Tables 6.4 and 6.5 show the system and energy configuration parameters, respectively. We also used FFT and DataFlow for some realistic benchmarks. These benchmarks were chosen to evaluate the estimation scheme and routing algorithm, and so a fully realistic benchmark was not necessary, and would be more relevant for testing a mapping scheme or the design of the whole chip including the PEs.

The evaluation was done by comparing the estimated power of each node to the simulation's energy results. The "T" value was set to match up with the simulation's run time, so that the power for "T" would be equivalent to the total energy of the simulation. We define the error of the estimation for node "i" in equation 6.4.10, where E is the Energy.

$$Error(i) = \left| \frac{E_{Simulation}(i) - E_{Estimation}(i)}{E_{Simulation}(i)} \right| \quad (6.4.10)$$

Using the given examples for estimated energy and the simulation's value of energy for each node, we provide an example error value. Table 6.1 represents the optical energy consumption for each node in a 4x4 photonic mesh network.

Table 6.1: Example s imulation energy values for a 4x4 network(J)

0.000196	0.000254	0.000246	0.000192
0.000221	0.000313	0.000299	0.000235
0.000212	0.000292	0.000302	0.000223
0.000183	0.000237	0.000238	0.000173

Table 6.2: Example estimated energy values for a 4x4 network(J)

0.000189	0.000246	0.000249	0.00019
0.000222	0.000318	0.000309	0.000205
0.000222	0.000283	0.000285	0.000224
0.000183	0.000244	0.000225	0.000160

Table 6.3: Example of error calculation(%)

3.549443	3.538508	1.55323	0.9722
0.40084	1.26193	3.31643	12.74263
4.62852	3.027572	5.609775	0.36122
0.175703	2.65302	5.367033	7.542093

Table 6.2 represents the estimated optical power consumption for each node in a 4x4 photonic mesh network for the run time, so that the total energy should be equivalent to the power. The resulting Error (displayed as a percentage), is given in table 6.3. As you can see, even the nodes that have a higher estimated value have a positive error value. The worst case error in this example is 12.7% (node (4,3)). If we average all of the values together, we get a value of 3.54%. We will use the average error, and the worst case error to evaluate the estimation technique's accuracy.

6.4.1.2 Algorithm Evaluation Methodology

We simulated the proposed SAFT-PHENIC system using a modified version of PhoenixSim which is developed in the OMNeT++ simulation environment [1]. The simulator incorporates detailed physical models of basic photonic building blocks such as waveguides, modulators, photodetectors, and switches. Electronic energy performance is based on the ORION simulator [2]. We evaluate the performance and energy consumption for 16, 64 and 256 core systems. We compare the performance of the proposed SAFT-PHENIC system with the baseline PHENIC [29], the previous version FT-PHENIC [30], and the conventional system [61]. All of the systems will use thermal-ignorant task mapping and we used the same benchmarks as we did for evaluating the accuracy of the estimation scheme.

Table 6.4: Configuration parameters.

Network Configuration	Value
Process technology	32 nm
Number of tiles	256, 64, or 16
Chip area (equally divided amongst tiles)	400 mm^2
Core frequency	2.5 GHz
Electronic Control frequency	1 GHz
Power Model	Orion 2.0
Buffer Depth	2
Message size	2 kb
Simulation time	10 ms (25×10^8 cycles)

Table 6.5: Photonic communication network energy parameters [2]

Network Configuration	Value
Datarate (per wavelength)	2.5 GB/s
MRs dynamic energy	375 fJ/bit
MRs static energy	400 μ W
Modulators dynamic energy	25 fJ/bit
Modulators static energy	30 μ W
Photodetector energy	50 fJ/bit
MRs static thermal tuning	1 μ W/ring

6.4.2 Power Estimate Evaluation

This section summarizes the results of our tests into a simple format. All results shown are given as percentages, and show either the average or the worst case error rate.

Table 6.6: 4x4 mesh accuracy results

4x4	Random	Bit-Rev.	DataFlow	FFT
Average % Error	3.62	3.83	1.79	2.56
Worst % Error	11.97	12.21	6.32	8.65

Table 6.7: 8x8 mesh accuracy results

8x8	Random	Bit-Rev.	DataFlow	FFT
Average % Error	3.60	3.89	1.82	2.53
Worst % Error	12.48	10.31	6.30	8.92

Table 6.8: 16x16 mesh accuracy results

4x4	Random	Bit-Rev.	DataFlow	FFT
Average % Error	3.61	3.97	1.80	2.62
Worst % Error	10.42	10.89	6.42	9.21

For Random Traffic, the average error rate is quite consistently 3.6%. This is a good sign, and is what we had hoped for from the simple calculation.

Bit reverse results seems to remain close to 3.8%, but also seems to increase as the network size increases. This is not a good sign, and is possibly due to the fact that the transmissions get longer, and the dynamic energy model may need to be improved, or use different constants for each network size. Regardless of that fact, the worst case happened on the smallest network, showing how this metric may not be the most valuable for determining the accuracy of the algorithm. The only network where the worst case seemed to be consistent was the Data Flow.

The Data Flow results seemed consistent regardless of network size because each transmission only travels one hop. This low value of 1.8% means that the used modulation and detection constants were working well. Because it only travels one hop, it never just passes through a node. Again, the worst case occurred in a corner, which had less traffic, because it had less nodes next to it.

The final benchmark is FFT. This has a fairly consistent error rate of 2.6% across the different network sizes. This is due to the fact that the transmissions travel less hops than random or bit reverse. The worst case error rate definitely increased with network size, but always seemed to occur towards the center of the chip. We believe that this is due to the central nodes having more traffic pass through them, when compared to the corners.

All in all, the results seem to be promising for a simple calculation, which can easily be done inside the chip. It does sacrifice accuracy for its simplicity. Also, if the technology were to change, we would simply need to change the coefficients.

6.4.3 LASA Routing Algorithm Evaluation

6.4.3.1 Performance Evaluation

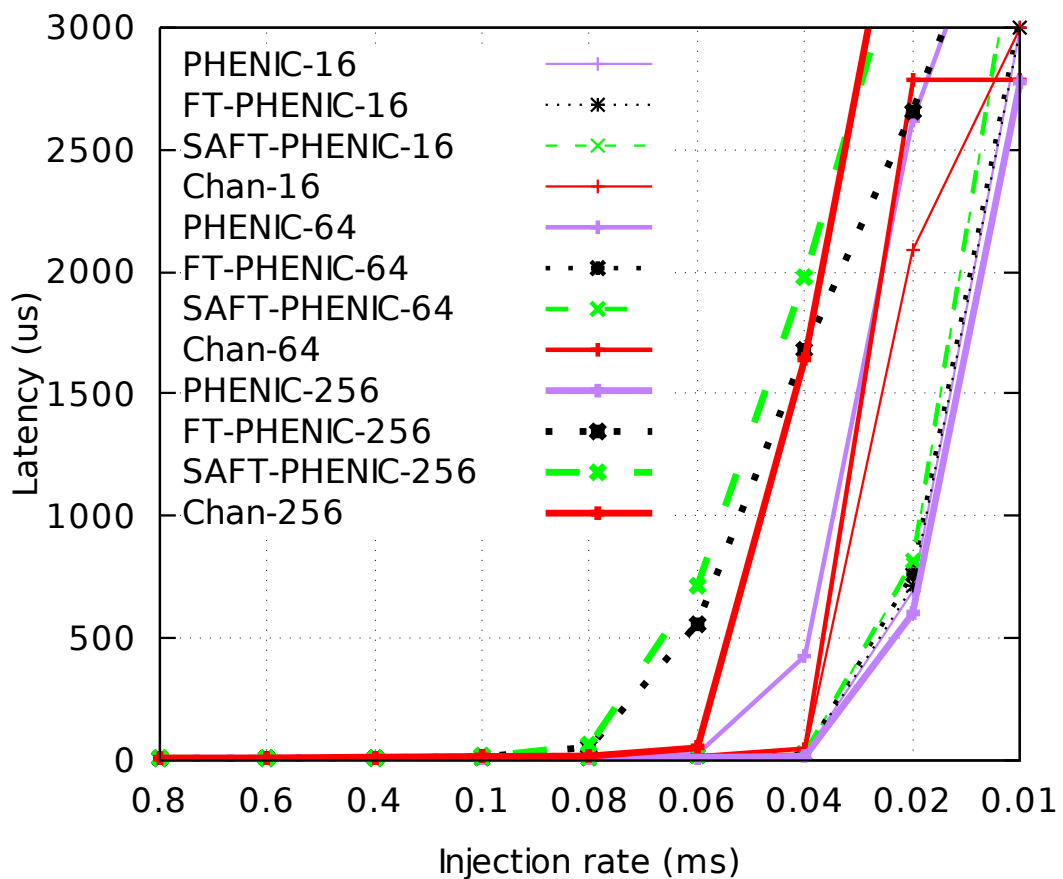


Figure 6.6: Overall latency results with various packet injection rates

Figure 6.6 shows the overall average latency. We can see that for zero-load latency, all networks behave in the same way, with a slight benefit towards the SAFT-PHENIC system because it will avoid the traffic thanks to the strain algorithm. Near saturation, PHENIC shows more flexibility and scalability, when using 256 cores, compared to the other networks. After saturation, we can see a performance sacrifice for using the SAFT-PHENIC vs the original PHENIC architecture. This is one of the prices of the added Fault-Tolerance and Peak Power mitigation.

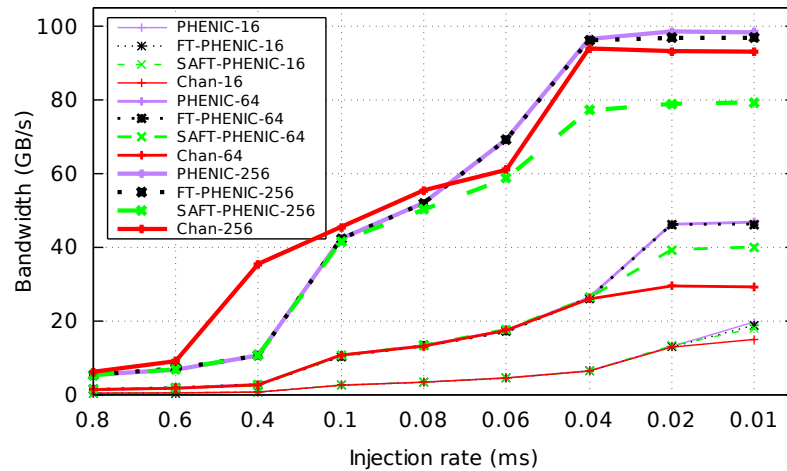


Figure 6.7: Bandwidth comparison results.

Figure 6.7 shows that the bandwidth of SAFT-PHENIC is smaller than the baseline, which is slightly better than the Chan Mesh for 16 and 64 core systems. When the system becomes very large, we can see that the additional calculations start to take a toll on the saturation bandwidth.

6.4.3.2 Energy Evaluation

Figure 6.8 shows the total energy and the energy efficiency comparison results for 16, 64 and 256 core systems. The data was taken at the point before saturation. The definition of energy efficiency is the total energy divided by the total number of bits transmitted. The most efficient is the baseline, which has no adaptive algorithm, and no fault tolerance mechanisms. The second most efficient is the SAFT-PHENIC network. This is because It has good bandwidth results and lower power than the crossbar system and the FT-PHENIC system. This reduction in power is due to

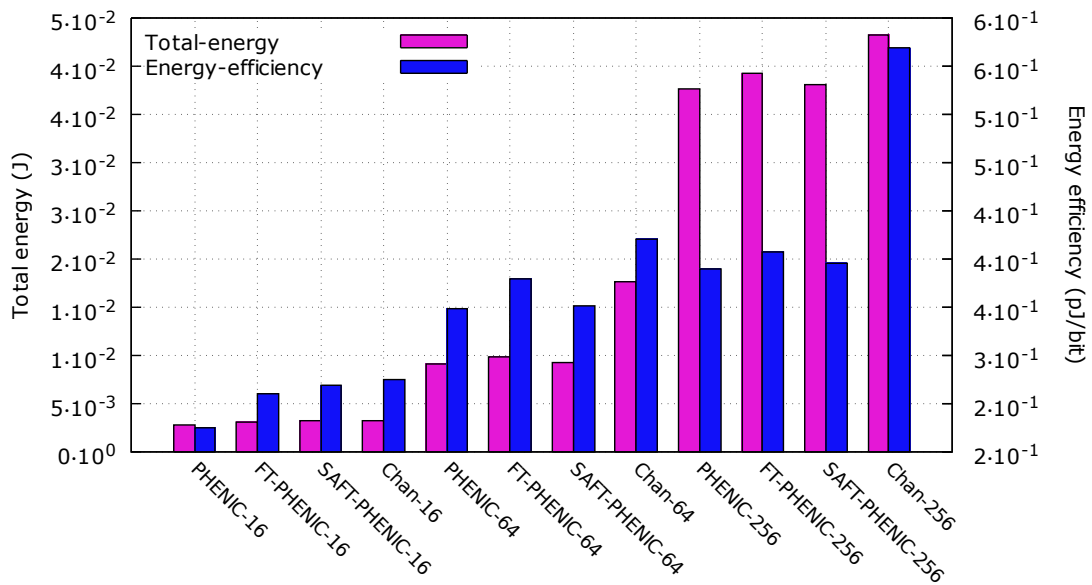
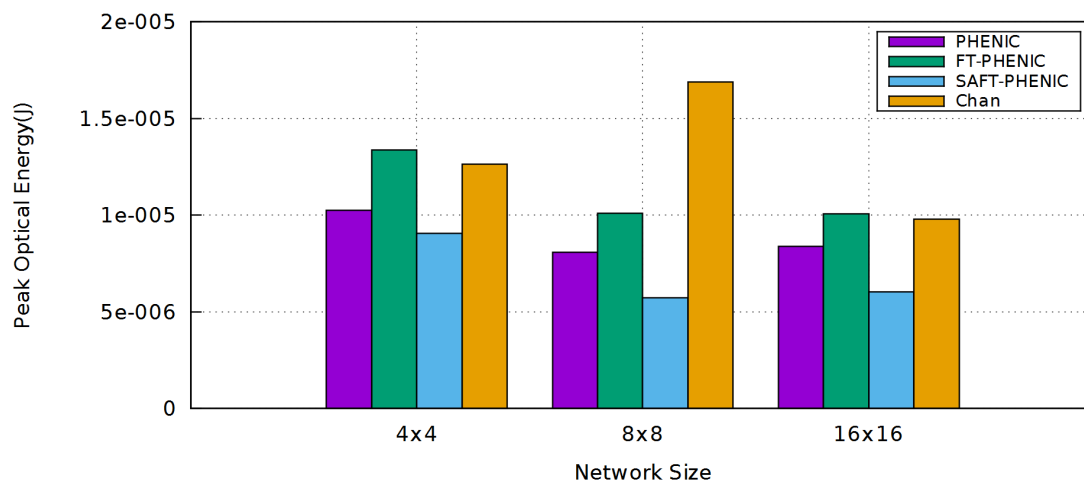


Figure 6.8: Total energy and energy efficiency comparison results near-saturation.

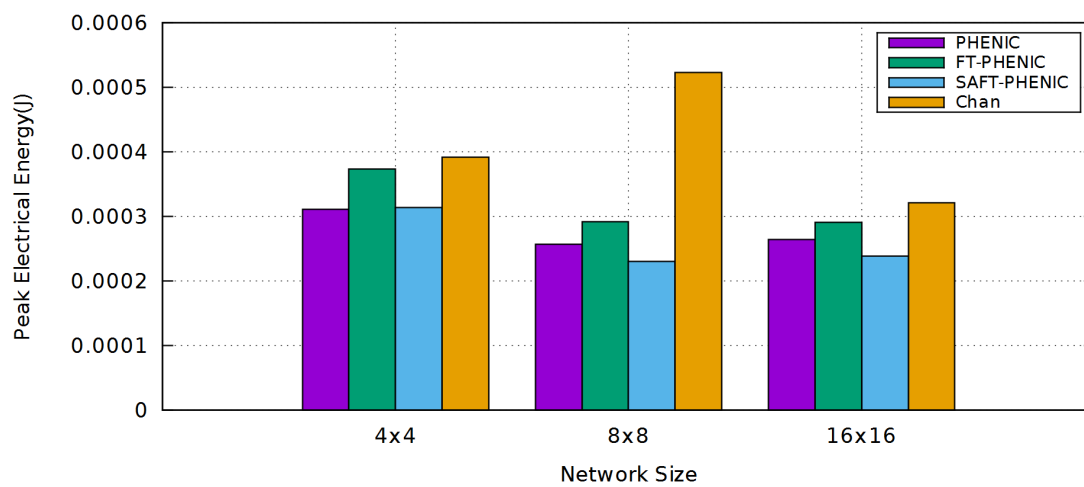
the reduction in blocked packets, which means that less transmissions must occur. The reduction in blocked packets is caused by nodes attempting to use the path less traveled. In Fig. 6.9 (a), we can see that the SAFT system successfully reduced the peak energy of the network in every case compared to all of the other networks. Compared to the previous FT-PHENIC (the most similar network), there is an approximate 38% reduction in peak optical energy. Figure 6.9 (b) shows that we were also able to successfully reduce the peak electrical energy. This is due to the traffic of the electrical network following the traffic of the optical network. For small network sizes, we saw that the PHENIC system provided the lowest energy, because the algorithm would likely cost more energy to compute the algorithm than it would save by avoiding the traffic-heavy nodes. At 8x8, more routing options are available, and we can see the benefits even in the peak electrical energy.

6.4.3.3 SAFT-PHENIC Fault-Tolerance Evaluation

The final section is used to evaluate the Permanent-Fault-Tolerance of the SAFT-PHENIC network. We were just building upon the already tried and proven FT-PHENIC, so we just want to see how well they can hold up against each other. To



(a)



(b)

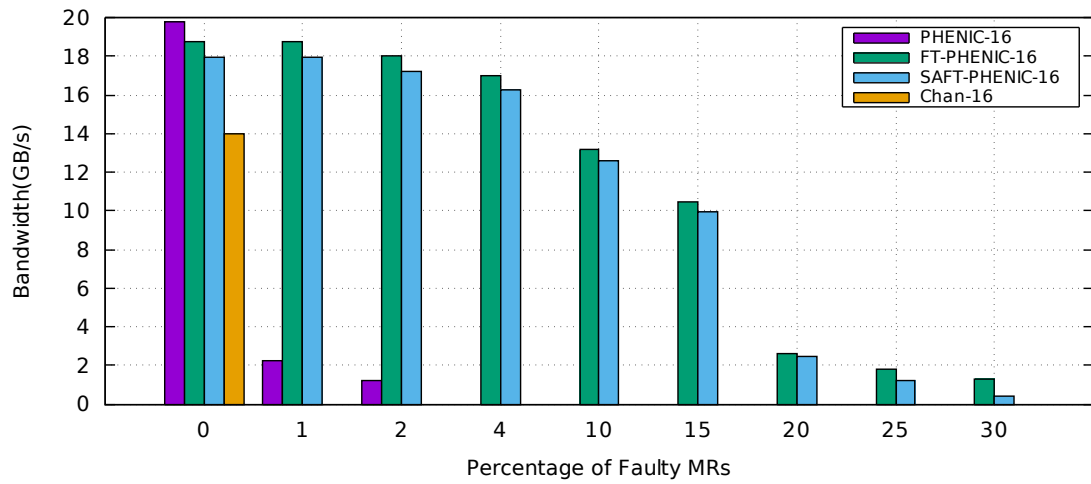
Figure 6.9: Peak (a)Optical and (b) Electrical energy of the most active node in the different networks.

this end, we evaluate the bandwidth as faults are injected into the network. As faults were introduced at different rates, we recorded the effect on the bandwidth of the systems.

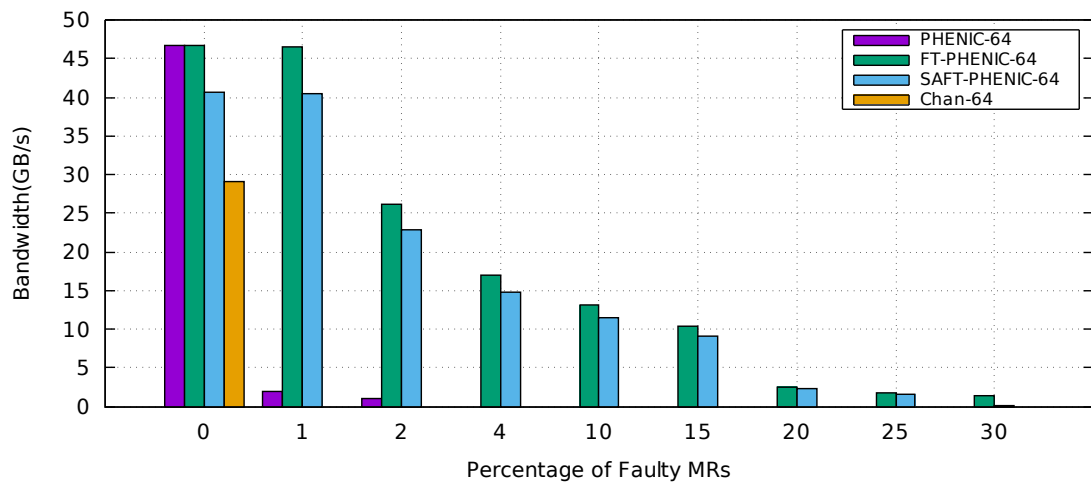
As we can see in Fig. 6.10, because we used results from after saturation, the SAFT does not have the highest starting value. As we inject some errors, we can see that it is as resilient as FT-PHENIC until 30% of MRs are faulty. Neither algorithm is usable beyond 20% because each of them has a huge bandwidth drop at that point, but we can see that at 30% the FT-PHENIC algorithm handles faults better. This is expected as the LASA algorithm limits the number of faults to a point that is slightly lower than the FT-PHENIC's algorithm, and thus handles a lower percentage of faults.

6.4.4 Chapter Summary

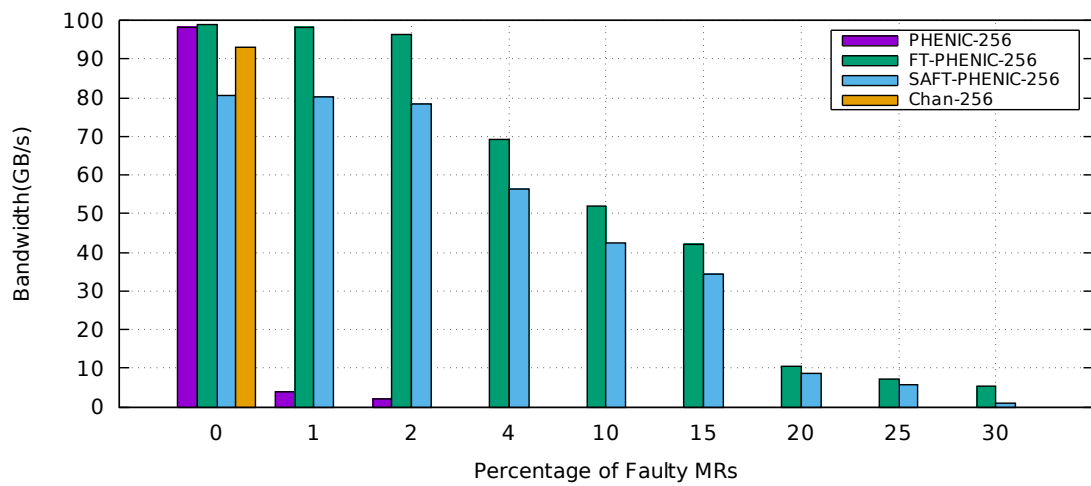
We presented a new control network architecture and its corresponding wavelength-shifting routing algorithm. We showed the performance benefits of this new scheme compared to the previously discussed PHENIC system, and especially compared to the conventional EA-PNoC architectures. The results show a significant reduction in the path configuration delay and the needed time to release the micro-rings. While the dynamic energy with the wavelength-shifting schemes increases, the energy efficiency considerably decreases and becomes independent of the communication's distance.



(a)



(b)



(c)

Figure 6.10: Affect on bandwidth as faulty MRs are introduced to (a)4x4 (b)8x8 and (c)16x16 Networks.

Table 6.9: Evaluation results summary under uniform random traffic.

	Tolerated MR Faults (%)		#Blocked Request		Band. (Gbps)		Path Conf. Energy (μ J)		Buff. Ener. (μ J)		Ener. Eff. (J/b)	
	64	256	64	256	64	256	64	256	64	256	64	256
PHENIC	1	1	42	85	46.7	98.33	1.84	14.56	0.13	2.01	0.34	0.38
PHENIC.BL	0	0	65	152	29.1	67.09	5.84	53.74	0.45	7.21	0.42	0.74
Chan_Mesh	0	0	65	152	29.1	67.09	5.80	53.74	0.45	7.21	0.42	0.74
Chan_Xb	1	1	50	120	47.12	95.563	2.13	31.59	1.11	1.71	0.24	0.51
Shasham	0	0	55	130	46.53	100.48	3.56	50.76	2.28	3.29	0.70	1.31
Xiang_Xb	4	4	51	122	29.17	93.09	2.19	31.82	1.14	1.79	0.54	0.77
FT-PHENIC	25	25	58	140	46.15	96.84	1.98	14.92	0.18	2.09	0.40	0.43
SAFT-PHENIC	20	35	52	122	40.03	79.33	1.89	14.70	0.15	2.02	0.35	0.40

Chapter 7

Conclusion and Discussion

This dissertation concludes with a summary of the main contributions and a discussion of future research. We create an overview of the results of each contribution. The future works will mostly consist of design considerations that were not implemented.

7.1 Contributions

This thesis presents three main contributions: (1) a reliable nanophotonic switch which we call FTTDOR, which has different versions with a different number of ports; (2) a proposed architecture called FT-PHENIC which utilizes a new fault-tolerant path configuration algorithm called FTTP; and (3) a stress-aware fault-tolerant routing algorithm for the optical on-chip networks.

Starting with a non-blocking 5x5 optical switch, we added redundancy at key locations in order to create a switch which can be utilized to provide some fault tolerance before avoiding the switch completely. This results in less blocked packets when faults are introduced, thus maintaining a more graceful decline in performance as a higher percentage of faulty MRs are introduced into the system. This switch can be utilized by any network which uses an optical switch with the same number of ports as one of the variants.

To control the fault-tolerant switch, we proposed a fault-tolerant path configuration algorithm. The proposed algorithm was very specific because it was an adap-

tation of the previous PHENIC's complex optical path configuration algorithm, but we believe that any path configuration algorithm can be modified in a similar way to allow for the use of two MRCTs. This resulted in a new network architecture, which we called FT-PHENIC, which allowed for various fault-tolerance techniques to be implemented.

Finally, we proposed a new and improved routing algorithm. The routing algorithm used traffic and fault information to create a 'Strain' value. This strain value is mostly based off of the power estimate with the assumption that the temperature is based off of the power use. The other factor was an upper limit on the number of MRs that could fail in each switch, which improved the message's ability to avoid possible system failure. This also helped us reduce the number of blocked packets, because the algorithm automatically avoided nodes that were being used more than other neighboring nodes.

7.2 Results Summary

This research mainly focused on improving the fault-tolerance of the network. We attempted to address both PV and TV of optical switches, which are the two biggest concerns of an optical switch. We then made a fault-tolerant optical network based on the improvements.

The FT-PHENIC and SAFT-PHENIC were compared to any relevant networks, such as the old PHENIC that they were based off of or another fault-tolerant network. We simulated it on networks with 16, 64 or 256 cores. This showed the modifications as the network increases in scale. The results showed that FT-PHENIC was much more graceful in terms of performance degradation as faults were injected, tolerating MR faults up to a point where 30% of them had failed. This was much better than the competitor's algorithm which gave out after about 10% of MRs were faulty. This was true across all network sizes.

The newly proposed networks showed some performance drawbacks when none of the MRs were faulty, when compared to the original PHENIC system. The FT-PHENIC was still able to keep similar performance to the PHENIC system because

its algorithm was based off of deterministic XY, which is what PHENIC used. The SAFT-PHENIC system had a significantly different algorithm, and thus had more of a drawback, but still performed better than the competitor.

In terms of the SAFT-PHENIC evaluation, we were able to show that using the algorithm aided in evenly spreading out the power used at each node. It is assumed that this power correlates to the temperature at each node, and by reducing the difference in power consumed by each node, we reduced the difference in temperature of different nodes.

7.3 Discussion

Even though FT-PHENIC and the LASA algorithm showed promising results, there are still some points that we would like to consider in future research.

We tried one method of addressing the thermal sensitivity of optical components, but there are certain elements that we would like to see changed. As stated before, thermally-resilient architectures are critical for the success of PNoC design. Our network attempts to reduce the variation from one node to the next by making their power consumption as equal as possible. This has a major flaw: currently producing the insulation layer between the optical and electrical layers is costly, wastes some space, would yields less chips, and so many designs do not include such a layer. This technique can easily be modified to include energy from the electronic router, with some additional terms, but a router rarely has knowledge of how much power a PE or memory unit is consuming. This means that the most effective way may be some form of temperature sensing. Thermometers are quite large when talking about this scale, and so they have some problems of their own, but as technology improves, better options may become available, and simply changing the power term to a temperature term would work if we had an efficient way to get temperature.

One key thing holding back the reliability of optical switches is the reliability of the basic MR unit. This reliability is based off of the physical parameters that are used when designing each unit. We would like to explore the physical properties of the MRs themselves to improve the reliability. As we have previously said in the

paper, making small changes to the shape, such as using racetracks [91] has led to an improvement in the reliability of MRs, and a reduction in the sensitivity to thermal variation. This means without changing the bending radius or waveguide thickness or material, they were able to improve reliability, and we would like to continue with such research.

Another item that would greatly aid the development of optical routers is the ability to buffer. Even more so the ability to read the data in multiple locations. Currently, splitting the data to be read will cause a large amount of insertion loss. Buffering is currently limited to causing a delay by creating optical coils, and can only delay the signal a very minimal amount of time, and causes a large amount of propagation loss [105].

One other topic to consider for future research is the design of photonic devices being integrated into electronic CMOS design. Currently the photonic design flow is significantly lacks design tools. The ability to use old CMOS tools with new optical libraries is critical for the photonics industry. A large problem is the difference in scale of photonic components and current electronic CMOS processing.

Bibliography

- [1] J. Chan, G. Hendry, A. Biberman, K. Bergman, and L. Carloni, “Phoenixsim: A simulator for physical-layer analysis of chip-scale photonic interconnection networks,” in *Design, Automation Test in Europe Conference Exhibition (DATE), 2010*, March 2010, pp. 691–696.
- [2] A. Kahng, B. Li, L.-S. Peh, and K. Samadi, “Orion 2.0: A power-area simulator for interconnection networks,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 20, no. 1, pp. 191–196, Jan 2012.
- [3] G. Leary and K. S. Chatha, “Design of noc for soc with multiple use cases requiring guaranteed performance,” in *VLSI Design, 2010. VLSID’10. 23rd International Conference on*. IEEE, 2010, pp. 200–205.
- [4] L. Benini and G. De Micheli, “Networks on chips: a new soc paradigm,” *Computer*, vol. 35, no. 1, pp. 70–78, 2002.
- [5] “The law that’s not a law,” *Spectrum, IEEE*, vol. 52, no. 4, pp. 38–57, April 2015.
- [6] S. B. Desai, S. R. Madhvapathy, A. B. Sachid, J. P. Llinas, Q. Wang, G. H. Ahn, G. Pitner, M. J. Kim, J. Bokor, C. Hu, H.-S. P. Wong, and A. Javey, “Mos2 transistors with 1-nanometer gate lengths,” *Science*, vol. 354, no. 6308, pp. 99–102, 2016. [Online]. Available: <http://science.sciencemag.org/content/354/6308/99>
- [7] *International Technology Roadmap For Semiconductors: Chapter: Interconnect*, 2011. [Online]. Available: www.itrs.net

- [8] R. Kumar, V. Zyuban, and D. M. Tullsen, “Interconnections in multi-core architectures: Understanding mechanisms, overheads and scaling,” in *Proceedings of the 32Nd Annual International Symposium on Computer Architecture*, ser. ISCA '05, 2005, pp. 408–419.
- [9] N. Magen, A. Kolodny, U. Weiser, and N. Shamir, “Interconnect-power dissipation in a microprocessor,” in *Proceedings of the 2004 International Workshop on System Level Interconnect Prediction*, ser. SLIP '04, 2004, pp. 7–13.
- [10] A. Ben Ahmed and A. Ben Abdallah, “Graceful deadlock-free fault-tolerant routing algorithm for 3D Network-on-Chip architectures,” *Journal of Parallel and Distributed Computing*, vol. 74, no. 4, pp. 2229–2240, 2014. [Online]. Available: <http://web-ext.u-aizu.ac.jp/~benab/publications/journals/JPDC14/JPDC-preprint.pdf>
- [11] A. Habibi, M. Arjomand, and H. Sarbazi-Azad, “Multicast-aware mapping algorithm for on-chip networks,” in *2011 19th International Euromicro Conference on Parallel, Distributed and Network-Based Processing*. IEEE, 2011, pp. 455–462.
- [12] B. Feero and P. Pande, “Performance evaluation for three-dimensional networks-on-chip,” in *IEEE Computer Society Annual Symposium on VLSI*, March 2007, pp. 305–310.
- [13] F. N. Sibai, “A two-dimensional low-diameter scalable on-chip network for interconnecting thousands of cores,” *Parallel and Distributed Systems, IEEE Transactions on*, vol. 23, no. 2, pp. 193–201, 2012.
- [14] A. B. Ahmed and A. B. Abdallah, “Onoc-spl customized network-on-chip (noc) architecture and prototyping for data-intensive computation applications,” in *Proceedings of the 4th International Conference on Awareness Science and Technology, Seoul, Korea*, vol. 2124, 2012, p. 257262.
- [15] S. Kumar, A. Jantsch, J.-P. Soininen, M. Forsell, M. Millberg, J. Öberg, K. Tiensyrjä, and A. Hemani, “A network on chip architecture and design

- methodology,” in *VLSI, 2002. Proceedings. IEEE Computer Society Annual Symposium on*. IEEE, 2002, pp. 105–112.
- [16] “ITRS Report Interconnect,” <http://www.itrs.net/ITRS%201999-2014%20Mtgs,%20Presentations%20&%20Links/2011ITRS/2011Chapters/2011Interconnect.pdf/>, 2011, [Online; accessed 1-July-2015].
- [17] C.-W. Chou, J.-F. Li, Y.-C. Yu, C.-Y. Lo *et al.*, “Hierarchical test integration methodology for 3-D ICs,” *IEEE Design Test*, vol. 32, no. 4, pp. 59–70, Aug 2015.
- [18] A. Ben Ahmed, A. Ben Abdallah, and K. Kuroda, “Architecture and design of efficient 3D network-on-chip (3D NoC) for custom multicore SoC,” in *Broadband, Wireless Computing, Communication and Applications (BWCCA), 2010 International Conference on*. IEEE, 2010, pp. 67–73.
- [19] S. Fujita, K. Nomura, K. Abe, and T. H. Lee, “3d on-chip networking technology based on post-silicon devices for future networks-on-chip,” in *2006 1st International Conference on Nano-Networks and Workshops*, 2006.
- [20] Y.-L. Jeang, T.-s. Wey, H.-Y. Wang, and C.-W. Hung, “Mesh-tree architecture for network-on-chip design,” in *Innovative Computing, Information and Control, 2007. ICICIC'07. Second International Conference on*. IEEE, 2007, pp. 262–262.
- [21] A. Ben Ahmed and A. Ben Abdallah, “Architecture and design of high-throughput, low-latency, and fault-tolerant routing algorithm for 3D-network-on-chip (3D-NoC),” *The Journal of Supercomputing*, vol. 66, no. 3, pp. 1507–1532, 2013.
- [22] R. Kumar, V. Zyuban, and D. M. Tullsen, “Interconnections in multi-core architectures: Understanding mechanisms, overheads and scaling,” in *Computer Architecture, 2005. ISCA'05. Proceedings. 32nd International Symposium on*. IEEE, 2005, pp. 408–419.

- [23] A. Ben Ahmed and A. Ben Abdallah, "PHENIC: Silicon photonic 3d-network-on-chip architecture for high-performance heterogeneous many-core system-on-chip," in *Sciences and Techniques of Automatic Control and Computer Engineering (STA), 2013 14th International Conference on*, Dec 2013, pp. 1–9.
- [24] A. Ben Ahmed, M. Meyer, Y. Okuyama, and A. Ben Abdallah, "Efficient router architecture, design and performance exploration for many-core hybrid photonic network-on-chip (2D-PHENIC)," in *2nd International Conference on Information Science and Control Engineering (ICISCE)*, April 2015, pp. 202–206.
- [25] A. Ben Ahmed, M. Meyer, Y. Okuyama, and A. Ben Abdallah, "Hybrid photonic noc based on non-blocking photonic switch and light-weight electronic router," in *The 2015 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, October 2015.
- [26] A. Ben Ahmed, Y. Okuyama, and A. Ben Abdallah, "Contention-free routing for hybrid photonic mesh-based network-on-chip systems," in *The 9th IEEE International Symposium on Embedded Multicore/Manycore SoCs (MCSoc)*, Sep. 2015, pp. 235–242.
- [27] A. Ben Ahmed, Y. Okuyama, and A. Ben Abdallah, "Non-blocking electro-optic network-on-chip router for high-throughput and low-power many-core systems," in *The World Congress on Information Technology and Computer Applications*, June 2015.
- [28] C. J. Nitta, M. K. Farrens, and V. Akella, "Resilient microring resonator based photonic networks," in *Proceedings of the 44th Annual IEEE/ACM International Symposium on Microarchitecture*, ser. MICRO-44. New York, NY, USA: ACM, 2011, pp. 95–104. [Online]. Available: <http://doi.acm.org/10.1145/2155620.2155632>

- [29] A. Ben Ahmed and A. Ben Abdallah, "Hybrid silicon-photonics network-on-chip for future generations of high-performance many-core systems," *The Journal of Supercomputing*, vol. DOI: 10.1007/s11227-015-1539-0, 2015.
- [30] M. Meyer, A. Ben Ahmed, Y. Tanaka, and A. Ben Abdallah, "FTTDOR: Microring fault-resilient optical router for reliable network-on-chip systems," in *The 9th IEEE International Symposium on Embedded Multicore/Manycore SoCs (MCSoc)*, September 2015, pp. 227–234.
- [31] S. Zhu and G.-Q. Lo, "Vertically-stacked multilayer photonics on bulk silicon toward three-dimensional integration," *Lightwave Technology, Journal of*, vol. PP, no. 99, pp. 1–1, 2015.
- [32] D. Miller, "Rationale and challenges for optical interconnects to electronic chips," *Proceedings of the IEEE*, vol. 88, no. 6, pp. 728–749, June 2000.
- [33] S. Koenig, J. Antes, D. M. Lopez-Diaz *et al.*, "20 gbit/s wireless bridge at 220 ghz connecting two fiber-optic links," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 6, no. 1, pp. 54–61, Jan 2014.
- [34] D. DiTomaso, A. Kodi, D. Matolak, S. Kaya *et al.*, "A-winoc: Adaptive wireless network-on-chip architecture for chip multiprocessors," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 12, pp. 3289–3302, Dec 2015.
- [35] B. R. Koch, A. Fang, and O. C. J. E. Bowers, "Mode-locked silicon evanescent lasers," *Optics Express*, vol. 18, no. 15, 2007.
- [36] X. Zheng, S. Lin, Y. Luo, J. Yao, G. Li *et al.*, "Efficient wdm laser sources towards terabyte/s silicon photonic interconnects," *Journal of Lightwave Technology*, vol. 31, no. 24, pp. 4142–4154, Dec 2013.
- [37] R. Kappeler, "Radiation testing of micro photonic components," stagiaire Project Report. ESA/ESTEC. September 29, 2004. Ref. No.: EWP 2263.
- [38] Z.-S. Hu, F.-Y. Hung, K.-J. Chen, S.-J. Chang, W.-K. Hsieh, and T.-Y. Liao, "Improvement in thermal degradation of zno photodetector by embedding

- silver oxide nanoparticles,” *Functional Materials Letters*, vol. 6, no. 01, p. 1350001, 2013.
- [39] W. Bogaerts, P. De Heyn, T. Van Vaerenbergh, K. De Vos, S. Kumar Selvaraja, T. Claes, P. Dumon, P. Bienstman, D. Van Thourhout, and R. Baets, “Silicon microring resonators,” *Laser & Photonics Reviews*, vol. 6, no. 1, pp. 47–73, 2012.
- [40] J. Ahn, M. Fiorentino, R. Beausoleil, N. Binkert, A. Davis, D. Fattal, N. Jouppi, M. McLaren, C. Santori, R. Schreiber, S. Spillane, D. Vantrease, and Q. Xu, “Devices and architectures for photonic chip-scale integration,” *Applied Physics A*, vol. 95, no. 4, pp. 989–997. [Online]. Available: <http://dx.doi.org/10.1007/s00339-009-5109-2>
- [41] S. Chu, W. Pan, S. Sato, T. Kaneko, B. Little, and Y. Kokubun, “Wavelength trimming of a microring resonator filter by means of a uv sensitive polymer overlay,” *Photonics Technology Letters, IEEE*, vol. 11, no. 6, pp. 688–690, June 1999.
- [42] D. Rafizadeh, J. Zhang, S. Hagness, A. Taflove, K. Stair, S. Ho, and R. Tiberio, “Temperature tuning of microcavity ring and disk resonators at 1.5- μm ,” in *Lasers and Electro-Optics Society Annual Meeting, 1997. LEOS '97 10th Annual Meeting. Conference Proceedings., IEEE*, vol. 2, Nov 1997, pp. 162–163 vol.2.
- [43] D. Xiang, Y. Zhang, S. Shan, and Y. Xu, “A fault-tolerant routing algorithm design for on-chip optical networks,” in *Reliable Distributed Systems (SRDS), 2013 IEEE 32nd International Symposium on*, Sept 2013, pp. 1–9.
- [44] Y. Xu, J. Yang, and R. Melhem, “Tolerating process variations in nanophotonic on-chip networks,” in *ACM SIGARCH Computer Architecture News*, vol. 40, no. 3. IEEE Computer Society, 2012, pp. 142–152.
- [45] A. B. Ahmed, “High-performance scalable photonics on-chip network for many-core systems-on-chip,” Ph.D. dissertation, University of Aizu, Japan, 2016.

- [46] M. Meyer, Y. Okuyama, and A. Ben Abdallah, “On the design of a fault-tolerant photonic network,” in *The 2015 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, October 2015, pp. 821–826.
- [47] A. Ben Ahmed and A. Ben Abdallah, “Hybrid silicon-photonic network-on-chip for future generations of high-performance many-core systems,” *The Journal of Supercomputing*, vol. 71, no. 12, pp. 4446–4475, 2015.
- [48] G. Hendry, J. Chan, S. Kamil, L. Oliner *et al.*, “Silicon nanophotonic network-on-chip using tdm arbitration,” in *IEEE 18th Annual Symposium on High Performance Interconnects (HOTI)*, Aug 2010, pp. 88–95.
- [49] B. G. Lee, M. A. Belkin, R. Audet, J. MacArthur, L. Diehl, C. Pflugl, F. Capasso, D. C. Oakley, D. Chapman, A. Napoleone *et al.*, “Widely tunable single-mode quantum cascade laser source for mid-infrared spectroscopy,” *Applied Physics Letters*, vol. 91, no. 23, pp. 231 101–231 101, 2007.
- [50] K. Bergman, L. P. Carloni, A. Biberman, J. Chan, and G. Hendry, *Photonic Network-on-Chip Design*. Springer-Verlag New York, 2014, ISBN: 978-1-4419-9334-2.
- [51] M. McLaren, N. L. Binkert, A. L. Davis, and M. Florentino, “Energy-efficient and fault-tolerant resonator-based modulation and wavelength division multiplexing systems,” 22 2014, uS Patent 8,705,972.
- [52] S. K. Selvaraja, W. Bogaerts, and D. Van Thourhout, “Loss reduction in silicon nanophotonic waveguide micro-bends through etch profile improvement,” *Optics Communications*, vol. 284, no. 8, pp. 2141–2144, 2011.
- [53] W. Bogaerts, P. Dumon, D. Van Thourhout, and R. Baets, “Low-loss, low-cross-talk crossings for silicon-on-insulator nanophotonic waveguides,” *Optics letters*, vol. 32, no. 19, pp. 2801–2803, 2007.
- [54] T. Barwicz, M. A. Popovic, P. T. Rakich, M. R. Watts *et al.*, “Microring-resonator-based add-drop filters in sin: fabrication and analysis,” *Opt. Express*, vol. 12, no. 7, pp. 1437–1442, Apr 2004.

- [55] T. Baba, S. Akiyama, M. Imai, N. Hirayama *et al.*, “50-gb/s ring-resonator-based silicon modulator,” *Opt. Express*, vol. 21, no. 10, pp. 11 869–11 876, May 2013.
- [56] D. Chen, H. R. Fetterman, A. Chen, W. H. Steier, L. R. Dalton, W. Wang, and Y. Shi, “Demonstration of 110 ghz electro-optic polymer modulators,” *Applied Physics Letters*, vol. 70, no. 25, pp. 3335–3337, 1997.
- [57] P. Dong, R. Shafiq, S. Liao, H. Liang *et al.*, “Wavelength-tunable silicon microring modulator,” *Opt. Express*, vol. 18, no. 11, pp. 10 941–10 946, May 2010.
- [58] L. Vivien, J. Osmond, J.-M. Fédéli, D. Marris-Morini *et al.*, “42 ghz p.i.n germanium photodetector integrated in a silicon-on-insulator waveguide,” *Opt. Express*, vol. 17, no. 8, pp. 6252–6257, Apr 2009.
- [59] Z. Tu, Z. Zhou, and X. Wang, “Reliability considerations of high speed germanium waveguide photodetectors,” in *SPIE OPTO*. International Society for Optics and Photonics, 2014, pp. 89 820W–89 820W.
- [60] A. Novack, M. Gould, Y. Yang, Z. Xuan *et al.*, “Germanium photodetector with 60 ghz bandwidth using inductive gain peaking,” *Opt. Express*, vol. 21, no. 23, pp. 28 387–28 393, Nov 2013.
- [61] J. Chan and K. Bergman, “Photonic interconnection network architectures using wavelength-selective spatial routing for chip-scale communications,” *IEEE/OSA Journal of Optical Communications and Networking*, vol. 4, no. 3, March 2012.
- [62] S. Xiao, M. H. Khan, H. Shen, and M. Qi, “Compact silicon microring resonators with ultra-low propagation loss in the C band,” *Opt. Express*, vol. 15, no. 22, pp. 14 467–14 475, Oct 2007.
- [63] S. guang Yang, L. Li, Y. ang Zhang, B. Zhang, and Y. Xu, “A power-aware adaptive routing scheme for network on a chip,” in *ASIC, 2007. ASICON '07. 7th International Conference on*, Oct 2007, pp. 1301–1304.

- [64] J. Keane and C. H. Kim, “An odometer for cpus: Microprocessors don’t normally show wear and tear, but wear they do,” *IEEE SPECTRUM*, vol. 48, no. 5, pp. 26–31, 2011.
- [65] S. Luryi, J. Xu, and A. Zaslavsky, *Future trends in microelectronics: up the nano creek*. John Wiley & Sons, 2007.
- [66] M. Agarwal, B. C. Paul, M. Zhang, and S. Mitra, “Circuit failure prediction and its application to transistor aging,” in *25th IEEE VLSI Test Symposium (VTS’07)*. IEEE, 2007, pp. 277–286.
- [67] J. Keane, T.-H. Kim, and C. H. Kim, “An on-chip nbtI sensor for measuring pmos threshold voltage degradation,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 18, no. 6, pp. 947–956, 2010.
- [68] E. Mintarno, J. Skaf, R. Zheng, J. B. Velamala, Y. Cao, S. Boyd, R. W. Dutton, and S. Mitra, “Self-tuning for maximized lifetime energy-efficiency in the presence of circuit aging,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, no. 5, pp. 760–773, 2011.
- [69] M. Radetzki, C. Feng, X. Zhao, and A. Jantsch, “Methods for fault tolerance in networks-on-chip,” *ACM Computing Surveys (CSUR)*, vol. 46, no. 1, p. 8, 2013.
- [70] K. Kuhn, C. Kenyon, A. Kornfeld, M. Liu, A. Maheshwari, W.-k. Shih, S. Sivakumar, G. Taylor, P. VanDerVoorn, and K. Zawadzki, “Managing process variation in intel’s 45nm cmos technology.” *Intel Technology Journal*, vol. 12, no. 2, 2008.
- [71] S. K. Saha, “Modeling process variability in scaled cmos technology,” *IEEE Design & Test of Computers*, vol. 27, no. 2, pp. 0008–16, 2010.
- [72] Y. Pan, P. Kumar, J. Kim, G. Memik, Y. Zhang, and A. Choudhary, “Firefly: Illuminating future network-on-chip with nanophotonics,” in *Proceedings of the 36th Annual International Symposium on Computer Architecture*, ser. ISCA ’09. ACM, 2009, pp. 429–440.

- [73] D. Vantrease, R. Schreiber, M. Monchiero, M. McLaren, N. Jouppi *et al.*, “Corona: System implications of emerging nanophotonic technology,” in *Computer Architecture, 2008. ISCA '08. 35th International Symposium on*, June 2008, pp. 153–164.
- [74] A. Shacham, K. Bergman, and L. P. Carloni, “Photonic networks-on-chip for future generations of chip multiprocessors,” *Computers, IEEE Transactions on*, vol. 57, no. 9, pp. 1246–1260, 2008.
- [75] G. Ramesh and S. SundaraVadivelu, “A reliable and fault tolerant routing for optical wdm networks,” *arXiv preprint arXiv:0912.0602*, 2009.
- [76] P. K. Loh and W.-J. Hsu, “Design of a viable fault-tolerant routing strategy for optical-based grids,” in *Parallel and Distributed Processing and Applications*. Springer, 2003, pp. 112–126.
- [77] Q. Xingyun, F. Quanyou, C. Yongran, D. Qiang, and D. Wenhua, “A fault tolerant bufferless optical interconnection network,” in *Computer and Information Science, 2009. ICIS 2009. Eighth IEEE/ACIS International Conference on*. IEEE, 2009, pp. 249–254.
- [78] L. Sahasrabudhe, S. Ramamurthy, and B. Mukherjee, “Fault management in ip-over-wdm networks: Wdm protection versus ip restoration,” *Selected Areas in Communications, IEEE Journal on*, vol. 20, no. 1, pp. 21–33, 2002.
- [79] J. Zhang and B. Mukherjee, “A review of fault management in wdm mesh networks: basic concepts and research challenges,” *Network, IEEE*, vol. 18, no. 2, pp. 41–48, 2004.
- [80] Y. Ye, J. Xu, X. Wu, W. Zhang, X. Wang, M. Nikdast, Z. Wang, and W. Liu, “Modeling and analysis of thermal effects in optical networks-on-chip,” in *2011 IEEE Computer Society Annual Symposium on VLSI*, July 2011, pp. 254–259.
- [81] C. Nitta, M. Farrens, and V. Akella, “Addressing system-level trimming issues in on-chip nanophotonic networks,” in *High Performance Computer Architec-*

- ture (HPCA), 2011 IEEE 17th International Symposium on, Feb 2011, pp. 122–131.
- [82] A. Guarino, G. Poberaj, D. Rezzonico, R. Degl’Innocenti, and P. Günter, “Electro–optically tunable microring resonators in lithium niobate,” *Nature Photonics*, vol. 1, no. 7, pp. 407–410, 2007.
- [83] B. Guha, B. B. C. Kyotoku, and M. Lipson, “Cmos-compatible athermal silicon microring resonators,” *Opt. Express*, vol. 18, no. 4, pp. 3487–3493, Feb 2010. [Online]. Available: <http://www.opticsexpress.org/abstract.cfm?URI=oe-18-4-3487>
- [84] C. J. Nitta, M. K. Farrens, and V. Akella, “Resilient microring resonator based photonic networks,” in *Proceedings of the 44th Annual IEEE/ACM International Symposium on Microarchitecture*, ser. MICRO-44. New York, NY, USA: ACM, 2011, pp. 95–104. [Online]. Available: <http://doi.acm.org/10.1145/2155620.2155632>
- [85] R. W. Hamming, “Error detecting and error correcting codes,” *Bell System technical journal*, vol. 29, no. 2, pp. 147–160, 1950.
- [86] M.-Y. Hsiao, “A class of optimal minimum odd-weight-column sec-ded codes,” *IBM Journal of Research and Development*, vol. 14, no. 4, pp. 395–401, 1970.
- [87] Q. Yu and P. Ampadu, “Adaptive error control for noc switch-to-switch links in a variable noise environment,” in *2008 IEEE International Symposium on Defect and Fault Tolerance of VLSI Systems*. IEEE, 2008, pp. 352–360.
- [88] Q. Yu and P. Ampadu, “Transient and permanent error co-management method for reliable networks-on-chip,” in *Networks-on-Chip (NOCS), 2010 Fourth ACM/IEEE International Symposium on*. IEEE, 2010, pp. 145–154.
- [89] Q. Yu and P. Ampadu, “Dual-layer adaptive error control for network-on-chip links,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 20, no. 7, pp. 1304–1317, 2012.

- [90] Y. Ye, X. Wu, J. Xu, W. Zhang, M. Nikdast, and X. Wang, "Holistic comparison of optical routers for chip multiprocessors," in *Anti-Counterfeiting, Security and Identification (ASID), 2012 International Conference on*. IEEE, 2012, pp. 1–5.
- [91] M. Mohamed, "Silicon nanophotonics for many-core on-chip networks," Ph.D. dissertation, University of Colorado, 2013.
- [92] J. Chan, G. Hendry, K. Bergman, and L. Carloni, "Physical-layer modeling and system-level design of chip-scale photonic interconnection networks," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, no. 10, pp. 1507–1520, Oct 2011.
- [93] B. Lee, C. Xiaogang, A. Biberman, L. Xiaoping *et al.*, "Ultrahigh-bandwidth silicon photonic nanowire waveguides for on-chip networks," *IEEE Photonics Technology Letters*, vol. 20, no. 6, pp. 398–400, March 2008.
- [94] H. Pan, S. Assefa, W. M. J. Green, D. M. Kuchta *et al.*, "High-speed receiver based on waveguide germanium photodetector wire-bonded to 90nm soi cmos amplifier," *Opt. Express*, vol. 20, no. 16, pp. 18 145–18 155, Jul 2012.
- [95] F. Xia, L. Sekaric, and Y. Vlasov, "Ultracompact optical buffers on a silicon chip," *Nat Photon.*, vol. 1:6571,2006, pp. 2801–2803, 2007.
- [96] W. Bogaerts, P. Dumon, D. V. Thourhout, and R. Baets, "Low-loss, low-cross-talk crossings for silicon-on-insulator nanophotonic waveguides," *Opt. Lett.*, vol. 32, pp. 2801–2803, Oct 2007.
- [97] B. Lee, A. Biberman, P. Dong, M. Lipson, and K. Bergman, "All-optical comb switch for multiwavelength message routing in silicon photonic networks," *IEEE Photonics Technology Letters*, vol. 20, no. 10, pp. 767–769, May 2008.
- [98] G. P. Agrawal, *Fiber-optic communication systems*. John Wiley & Sons, 2012, vol. 222.
- [99] M. "Meyer, Y. Okuyama, and A. B. Abdallah, ""microring fault-resilient photonic network-on-chip for reliable high-performance many-core systems", "

- "*The Journal of Supercomputing*", pp. "1–33", "2016". [Online]. Available: "http://dx.doi.org/10.1007/s11227-016-1846-0"
- [100] C. Adi, H. Matsutani, M. Koibuchi, H. Irie, T. Miyoshi, and T. Yoshinaga, "An efficient path setup for a photonic network-on-chip," in *2010 First International Conference on Networking and Computing (ICNC)*, Nov 2010, pp. 156–161.
- [101] G. Hendry, E. Robinson, V. Gleyzer, J. Chan, L. Carloni *et al.*, "Circuit-switched memory access in photonic interconnection networks for high-performance embedded computing," in *High Performance Computing, Networking, Storage and Analysis (SC), 2010 International Conference for*, Nov 2010, pp. 1–12.
- [102] A. Shacham, K. Bergman, and L. Carloni, "On the design of a photonic network-on-chip," in *First International Symposium on Networks-on-Chip, NOCS 2007*, May 2007, pp. 53–64.
- [103] A. Shacham, K. Bergman, and L. Carloni, "Photonic networks-on-chip for future generations of chip multiprocessors," *IEEE Transactions on Computers*, vol. 57, no. 9, pp. 1246–1260, Sept 2008.
- [104] M. Meyer, Y. Okuyama, and A. Ben Abdallah, "A power estimation method for mesh-based photonic noc routing algorithms," in *Proc. of the Fourth International Symposium on Computing and Networking Hiroshima, Japan*, November 2016, pp. 80–83.
- [105] S. Fathpour and N. A. Riza, "Silicon-photonics-based wideband radar beamforming: basic design," *Optical Engineering*, vol. 49, no. 1, pp. 018 201–018 201–7, 2010. [Online]. Available: <http://dx.doi.org/10.1117/1.3280286>

List of Publications

Refereed Journals

1. **Michael Meyer**, Yuichi Okuyama, Abderazek Ben Abdallah, “Microring Fault-resilient Photonic Network-on-Chip for Reliable High-performance Many-core Systems”, The Journal of Supercomputing, (2016). doi:10.1007/s11227-016-1846-0(Major)
2. Khanh N. Dang, **Michael Meyer**, Yuichi Okuyama, Abderazek Ben Abdallah, “A Low-overhead Soft-Hard Fault Tolerant Architecture, Design and Management Scheme for Reliable High-performance Many-core 3D-NoC Systems”, The Journal of Supercomputing, (2017). doi:10.1007/s11227-016-1951-0. (Major)

Refereed International conferences

1. **Michael Meyer**, Akram Ben Ahmed, Yuichi Okuyama, Abderazek Ben Abdallah, “FTTDOR: Microring Fault-resilient Optical Router for Reliable Network-on-Chip Systems”, Proc. of IEEE 9th International Symposium on Embedded Multicore/Many-core SoCs (MCSoc-15), pp. 227 - 234, Sep 23-25, 2015. (Major)
2. **Michael Meyer**, Akram Ben Ahmed, Yuki Tanaka, Abderazek Ben Abdallah, “On the Design of a Fault-tolerant Photonic Network-on-Chip”, Proc. of IEEE International Conference on Systems, Man, and Cybernetics (SMC2015), Oct. 9-12, 2015, pp. 821 - 826. (Major)

3. **Michael Meyer**, Yuichi Okuyama, Abderazek Ben Abdallah, “A Power Estimation Method for Mesh-based Photonic NoC Routing Algorithms, Proc. of the Fourth International Symposium on Computing and Networking Hiroshima, Japan, November 22-25, 2016. (Non-Major)
4. Khanh Dang, **Michael MEYER**, Yuichi OKUYAMA, Abderazek BEN ABDALLAH, “Reliability Assessment and Quantitative Evaluation of Soft-Error Resilient 3D NoC System” Proc. of the 25th-IEEE Asian Test Symposium (ATS16), November 21-24, 2016. (Major)
5. Achraf Ben Ahmed, **Michael Meyer**, Yuichi Okuyama and Abderazek Ben Abdallah, “Hybrid Photonic NoC Based On Non-blocking Photonic Switch and Light-weight Electronic Router”, in the IEEE Proceeding of the 2015 International Conference on Systems, Man and Cybernetics (SMC), pp. 56-61, October 2015. (Major)
6. Khanh N. Dang, **Michael Meyer**, Yuichi Okuyama, Abderazek Ben Abdallah, Xuan-Tu Tran, “Soft-Error Resilient 3D Network-on-Chip Router”, Proc. of IEEE 7th International Conference on Awareness Science and Technology (iCAST 2015), pp. 84 - 90, Sep. 22-24, 2015. (Major)
7. Achraf Ben Ahmed, **Michael Meyer**, Yuichi Okuyama and Abderazek Ben Abdallah, “Efficient Router Architecture, Design and Performance Exploration for Many-core Hybrid Photonic Network-on-Chip (2D-PHENIC)”, in the IEEE Proceedings of the International Conference on Information Science and Control Engineering (ICISC), pp. 202-206, April 2015. (Non-Major)
8. Abderazek Ben Abdallah, Mitsuhiro Nakamura, Akram Ben Ahmed, **Michael Meyer**, Yuichi Okuyama, “Fault-tolerant Router for Highly-reliable Many-core 3D-NoC Systems”, Proc. of the 3rd International Scientific Conference on Engineering and Applied Sciences (ISCEAS 2015), July 29-31, 2015, Okinawa, Japan. (Non-Major)
9. A. Ben Ahmed, **M. Meyer**, Y. Okuyama, and A. Ben Abdallah, “Adaptive Error- and Traffic Aware Router Architecture for 3D Network-on-Chip Sys-

tems”, IEEE Proceedings of the 8th International Symposium on Embedded Multicore/Many-core SoCs (MCSoc-14), pp. 197-204, Sept. 2014. (Major)