FTTDOR: Microring Fault-resilient Optical Router for Reliable Optical Network-on-Chip Systems

Michael Conrad Meyer, Akram Ben Ahmed, Yuichi Okuyama, Abderazek Ben Abdallah
The University of Aizu
Graduate School of Computer Science and Engineering
Adaptive Systems laboratory, Division of Computer Engineering
Aizu-Wakamatsu 965-8580, Japan
E-mail: {d8161104, d8141104, okuyama, benab}@u-aizu.ac.jp

Abstract—Photonic Networks-on-Chip (PNoCs) have been proposed as a disruptive technology solution to the silicon problem showing their great superiority over their electronic counterparts. In these architectures, higher bandwidth can be achieved thanks to the light speed transmissions, and the power required to transmit over a distance is much lower. Despite these advantages, PNoC designs are very complex, and thus are more susceptible to physical defects and short-term malfunctions. Therefore, fault-tolerance has become a primary requirement for these future generation high-performance systems.

In this paper, we present a fault-tolerant optical router, named FTTDOR, with its electrical control module for highly-reliable low-power 3D-Networks-on-Chip (PHENIC). It uses minimal redundancy to assure accuracy of the packet transmission even after faulty Microring Resonators (MRs) are detected. The fault-tolerant optical switch is decomposed non-blocking, with minimal MRs, and requires no MRs for straight transmission (East to West, North to South, and Up to Down, as well as their inverses). Simulation results show that the network can maintain 98% and 94% throughput when considering 3% and 20% fault-rates, respectively. These results come with 35% decrease in the number of MRs when compared to the conventional crossbar switch resulting in 32% power reduction.

Keywords—Fault-tolerant; Optical Router/Switch; 3D-NoCs; Hybrid; Control Router.

I. INTRODUCTION

Even though Integrated Chips (ICs) have been getting smaller throughout the years in accordance with Moore’s Law, this is showing signs of change. Since 2010, the number of cores on a die and chip size has almost completely leveled out [1]. This means that we are rapidly approaching the limit of Silicon chip performance and power where low-power areas are referred to as Dark Silicon. Dark Silicon is one of the biggest fears of silicon chip designers for large-scale computing [2].

There are many ideas as to what the future holds, and one commonly sought after technology is Silicon Photonics. Intel is currently investigating Photonics as a method to connect multiple CPUs, while other companies [3] are focusing their research on Photonics inside the silicon. Combining photonic technology with the Network-on-Chip (NoC) paradigm was just a matter of time. Many Photonic Networks-on-Chip (PNoCs) routers have sprung up and proved the power and throughput benefits of optical technology [4], [5]. The largest drawback is that they have latency issues. Some solutions proposed to use an electrical network for the latency critical packets, and the optical network for the packets which require high throughput [6]. As far as actual fabrication goes, a few places have had success in producing Nanophotonic chips. Cornell has had success with creating buried Microring Resonators (MRs) in polycrystalline silicon [7]. Optical 3D interconnects have been investigated in [8] and [9].

Relying heavily on electrical interconnects and data transmission, ICs have gotten smaller and smaller throughout the years. The smaller ICs are more susceptible to variation in process, and thus are more likely to have faults caused by physical defects. Additionally, the smaller circuits are more vulnerable to transient faults caused by component failures. As the complexity of a circuit increases, there are simply more chances for something to go wrong, and thus many-core processors are highly susceptible to faults. The Optical domain is immune to transient faults caused by radiation [10], but is still susceptible to production variation as well as aging. The aging typically occurs faster in active components as well as elements that have high thermal variation [11]. In the optical domain, the faults can occur in MRs, waveguides, routers, etc. Active components, such as MRs, have higher failure rates than passive components, e.g. waveguides [11].

Fault tolerance is crucial when considering mission critical applications where the system must correctly function even when something goes wrong. One such an application is that of space travel, where repair or replacement is not a possible option, and billions of dollars would be wasted. Some authors have investigated rerouting in the optical domain to provide some fault tolerance [12], [13]. This is a solution that has been widely successful in the electrical domain, and has shown some success in the optical one. The large flaw to this is that it often reroutes traffic in a similar fashion and cause traffic to focus on one area, and essentially ignoring the problematic routers all together.
This paper proposes a fault-tolerant optical router, named FTTDOR, with its electrical control circuit for highly-reliable low-power 3D-NoCs. It uses minimal redundancy to assure accuracy of the packet transmission even after a fault occurs in a MR. Here we only cover MR based faults since the field of node faults and interconnect faults are mainly handled by rerouting by the aforementioned authors. But as far as MR faults, there are no signs of current research, which is the basis of this paper. This allows the switch to maintain all of its capability of routing messages to any direction and avoiding the need to reroute packets. This can significantly save on computation time, and not having to recalculate routes is quite beneficial to the throughput of the system.

The rest of the paper is organized as follows: section 2 talks about the works related to the paper; section 3 covers the PHENIC system, which the switch was designed for and evaluated based on; section 4 goes over the details of the proposed switch; section 5 describes the evaluation methodology and results; finally, section 6 wraps up the paper with some closing remarks and highlights the benefits.

II. RELATED WORK

There are many different forms of NoC architecture out in development [6], [4], [14], [15], [16]. Most of these do not address fault issues within switches. Others do not even address the issue of the throughput limitation of electrical signals. 3D-OASIS [15] and the original mesh [14] 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 [5]. This new technology is not 100% beneficial, as can be seen with the results of certain optical NoCs.

Corona [4] is an example for a pure optical NoC. Upon testing, it shows the latency problems of the Nanophotonic systems. Firefly [16] demonstrated the concept of a hybrid network. It has both optical interconnects and electrical ones to deliver data. The problem is that it did not implement this to the fullest of its potential. Iris [6] was the first design to separate flits into high throughput, and low latency stacks. The high throughput transmissions would be done through the optical network and the low latency signals would be transmitted through a layer of broadcast antennae. This showed great benefits over Corona and the other designs. The proposed PHENIC system is similar in that it separates the data into high-throughput, and low-latency dependent categories. But, instead of using antennae to transmit the data from one router to another, there is a separate electronic mesh network solely for the purpose of these low-latency transmissions.

Wavelength Division Multiplexing (WDM) is a method which employs several channels of data in the form of different wavelengths to occupy a single waveguide. The data is then demultiplexed and the message is reconstructed at the destination. This method requires several MRs at each turn in order to be implemented. This does however provide a sort of redundancy of the whole switch and the optical components of the network interface. This allows for the fault tolerance by replacing a single channel in the case that extra are available. This is detailed by a few authors [17], [18]. If a certain photodetector or modulator has gone bad, then the wavelength can be disabled and replaced with a different one, and no other operations have to be changed. This only helps with modulator and photodetector faults though. Other than this method of completely redundant switch combinations, as far as MRs, not much work has been done to solve switch faults.

Faults in nodes and link faults have been greatly explored by others. This is usually solved in the form of routing algorithms. Ramesh et al. [17] proposes a method of determining and using back up routes. This detection is handled in the form of a cost function based around the load index [17]. Loh et al. uses a default routing algorithm and a backup routing method [12]. The two methods are called Logical Route and Adaptive Route. The Logical Route is a few sets of dimension order routing. This method simply checks for faults and congestion along the way, and if it 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 [12]. However, this requires some knowledge of the faults ahead of time, which can be done by a fault detection algorithm. The last fault tolerant method of routing is proposed by Xiang et al. [19]. This method uses a Minus first routing algorithm as a basis. The authors do not detail how to detect a faulty link, but once a faulty link is discovered, then it 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 type of algorithm is used for a network switch which uses optical and electrical conversions at each node [19]. 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 a few more fault tolerance options which can be detailed in [20]. But, it would deter from the original purpose of converting to the optical domain, namely power and throughput.

As for fault detection, Raik et al. [21] proposed a diagnostic packet method for the electrical mesh NoCs. Their method was to pause the processor, and send diagnostic messages in three steps. The first step was to test the network interfaces of each node, and for this, each node creates a message and sends it to itself, and then checks the message for errors. The second step was to test each individual interconnection link by creating a message to send to the neighbor in each direction. This would test both directions
of each interconnect. The final step was to test the switches within the router. He accomplished this by creating a series of tests which would gradually shift, until each node had managed all four turns (2D). A similar method is needed for photonic chips; but, it does not exist currently.

In this paper, we attempt to tackle the problem of fault tolerance without rerouting in a node-to-node sense. The path within the optical switch changes slightly; but, this should be handled purely by the arbiter and there should be no need for a routing computation to go to waste unless there is some form of blocking or current use.

III. PHENIC SYSTEM

PHENIC is a hybrid optical mesh-based Network-on-Chip (NoC) and a Photonic Communication Network (PCN). Fig. 1 shows the high level view of PHENIC system. In this figure an 8x8x4 configuration example is shown; but, the architecture can be expanded to any desired size, and it was evaluated on a 6x6x6 configuration. This can easily be expanded until we start reaching the signal strength limits of current technology, and the signal to noise ratio becomes a problem. The ECN is highly based off of the 3D-OASIS-NoC [15]. This layer contains all of the electrical components used for processing as well as the electrical networking. The PCN consists of waveguides and MRs which are assembled into optical switches which connect other waveguides, which are used as optical interconnects.

The PCN is a 3D Mesh, which handles communication in an end-to-end fashion. It is controlled by electrical signals which come from the ECN in accordance with the routing requirements of each packet. When the next-port is granted, then the arbiter module controls the individual MRs according to a state table where all MR states are stored. In this fashion, the input and output ports are also reserved. Once the data has successfully been transmitted, then a tear-down signal is sent and the arbiters release the MRs and ports. In section 5, we provide an evaluation of both networks, but we mainly focus on the PCN.

In PHENIC, the tear-down, ACK and Path Setup are all handled by the ECN. This process can be seen in Fig. 2. This network also implements Wavelength Division Multiplexing (WDM), which allows for multiple wavelengths to carry data along the same waveguide. This network is designed to have 16 total wavelengths, but only use 12 at a time, allowing for four whole wavelengths to be used as a sort of modular redundancy to resolve faults, such a method is detailed in [18]. This is purely a form of modular redundancy. Assuming that the path is not currently reserved, then only the Path Setup (PS) packet is sent and it follows the green line illustrated in Fig. 2. But, if any of the links are currently being used for another transmission, then a Path Blocked (PB) packet is sent along the red path, as seen in Fig. 2. In this case, the packet is not transmitted till the requested resources are free or another path is calculated, and the previously reserved MR (state 01) are freed (state 00). Once the path is reserved, the packet is sent into the Network Interface (NI), where it is converted to the optical domain, and transmitted across all of the switches in an end-to-end fashion, riding along the path preset by the MR states. At the other end, the photo-detectors convert the signal back to the electrical domain, where it can be read. The tear-down signal also follows the same red path as the Path Blocked packet would have, and the nodes are released.

IV. FAULT TOLERANT OPTICAL SWITCH

Figure 3 shows the optical switch. The optical waveguides carry the signal, similar 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. 3), special 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 Z direction 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 decomposed non-blocking switch, meaning that the 5 port and the 4 port switches that composed it allow 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 occurs. This means that any port can put data into it and it can output to any other port. The design of the
four ports switch only has redundant MRs on the left side. 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 some redundant MRs, and two extra MR locations. This means that the fault tolerant element of the switch is merely a modification of an already existing switch. 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. As all rerouting in the switch occurs on the core waveguide, traffic certainly increases on this one waveguides 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 FTDDOR switch has been designed to require no MRs from East-West, North-South, and Up-Down traffic. 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 fails. 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).

The labeled resonators can be seen in Fig. 4. Table I 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). In the same table, a "-" denotes a path which does not require any resonators to be turned on. The table includes multiple entries for each of the active paths. The first one mentioned is always the primary path, and the second one mentioned is the backup path. A backup path is only used if there is a detected fault. Fig. 4 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 I, we can see that West to South has a backup route of MRs 8 and 1. This adds an extra resonator to the path (represented dashed green line in Fig. 4) 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.

![Fault-Tolerant 7x7 Optical Switch Example demonstrating an original and backup route for a fault in MR 9 (Redundant MRs drawn as single circle).](image)

The arbiter is in control of the fault recovery. In most networks, the primary purpose is to check for a blocked packet, and output the control signals for the MRs. The arbiter in FTTDOR uses code based around the pseudocode seen in algorithm 1. There are essentially five steps: 1) check the fault status of the MR, 2) check the status of the ports affected by the new path, 3) reserve the path, 4) wait for the ACK then activate the MRs, and 5) wait for the tear-down and release the MRs. It is worth noting that u(Algorithm 1) can refer to one or multiple MRs. At the end of the algorithm, there is a section for resetting the MR to 0 in the case that a *Path Blocked* packet is received. The switch and the arbiter do not handle the fault detection at this point. It is currently assumed that there will be a detection mechanism outside of these two components as well as outside the scope of this paper. The mechanism sends a packet which gets detected by the arbiter similar to the *Path Blocked* packet.

**Algorithm 1** Arbiter Pseudo Code

1: procedure Check and Reserve
2:   Check the Resonators in Play:
3:     u ← MRprimary(In, Out)
4:     if MR[a] == 11 then u ← MRBackup(In, Out)
5:     x ← MRPrimaryRowIn
6:     y ← MRPrimaryColumnOut
7:     z ← null
8:     if ((x = MRprimarIn, Out) || (Out = (Up[Core]Down)) || (In == Core))
9:       then z ← MRBackup(In, Out)
10:      i ← RemoveDuplicatesSum, y, z
11:     for all i > do if ((MR[i] == 01)||(MR[i] == 10)then BLOCK & & MR[a] = 00
12:     Reserve the Resonator:
13:     MR[a] ← 01
14: procedure Wait and Activate
15:   Pauset until ACK = 1.
17:   Pause until Teardown = 1.
18:   MR[a] ← 00.
19: procedure Blocked
20:   MR[a] ← 00
21: procedure FailMR[i]
22:   MR[a] ← 11

---

**V. Evaluation**

**A. Evaluation methodology**

The goal of PHENIC system is to enhance the performance and energy efficiency by taking advantage of photonic 3D-NoC over electrical NoC. The ECN is designed in Verilog-HDL and synthesized with Synopsys Design Compiler CAD tool, using 45nm technology. Table II shows the system configuration and the hardware complexity of the electronic control router is shown in Table V.

An in-house event-driven simulator, based around Memon’s [23], was used to evaluate the throughput of a 3D-Mesh-based NoC system configured with our proposed FTTDOR switch. First, an XYZ routing algorithm was given multiple traffic patterns (Random and Bit-reverse). The data flow from any one port to any other port was measured. Once this data was acquired, it was saved into a profile. This profile was then applied to each switch, and the probability of a route failing or being blocked by another route was calculated for each optical switch. Based on the data flow probability and assuming equal failure rates in all of the MRs, a percentage of the original unblocked data flow is calculated. This number is given as a percentage as normalized to the crossbar without faults. This was able to yield a percentage of data flow change in each of the three switches tested. The results yielded would be similar percentages for wavelength division multiplexing, as well as for a single wavelength, because the traffic percentages would be the same, just overall throughput would increase. Because the single wavelength is included, data patterns of individual wavelengths going to different destinations will also yield similar percentages if wavelength assignment is used. We confirmed throughput calculations in a method similar to that used by Memon [23]. The big difference is that we also account for blocking a path, not just a failed path. The traffic pattern of the whole network is evaluated on a single switch because the 7 port switch is designed for the center nodes of the network and not the outer nodes, and most optical networks will use a wide variety of switches for the corners, and edges.

The first key is determining the probability that a switch would fail (a connection between one input and one output can not be attained). This is referred to as the survival probability, and changes based on the shape of each switch. Assuming equal chances of failure in each MR, and knowing the resonators fault-rate, for example 5%, and that is representative of one resonator. We then take every pair of resonators(10%) that can fail, and match that with the probability that their traffic is required. We also account for the probability that their traffic is already blocked. This gives us equation (1). This equation just says that the switch can achieve maximum throughput percentage minus the traffic with a failed route and minus the traffic with a blocked route, for each failed set of resonators, i.e. the sum of FT(i) the list
of failed paths when a particular set, i, of MRs are faulty, and \( p(FT) \) is the probability of that traffic being used. The same goes for \( BT(i) \) and \( p(BT) \), just for a blocked path, and not a failed one. There is one additional term to the Blocked Throughput Probability equation (3), and that is the sum of \( \sum p(TBBT) \), which represents the sum of probabilities of Traffic that Blocks the Blocked Traffic. \( p(i) \) is the probability of a particular set of resonators failing, this probability will remain constant as long as we are considering one switch and one percentage of failed MRs. We then go through all the sets, and the throughput percentage can be calculated. The probabilities of traffic are taken from the traffic profiles generated. Currently, only random and bit reverse traffic are implemented.

\[
TP(sw) = 1 - \left( \sum p(i) * (FTP(i) + BTP(i)) \right) \\
FTP(sw) = \left( \sum (FT(i) * p(FT)) \right) \\
BTP(sw) = \left( \sum (BT(i) * p(FT) * \sum p(TBBT)) \right)
\]

Finally, we also input the switch, without the redundancies into the CLAP simulator [24], to focus solely on power calculations. This would give power loss for the worst case, and the power loss per tile can be extrapolated with that data and the cost to drive the ring resonators. The number of MRs and waveguide crossings were either taken from a stat sheet or manually counted. Some discrepancy from pictures shown and their data sheets can occur due to counting resonators that would require a figure 8 resonator, which we account as 2 MRs. We determined figure 8 resonators based on the way in which they sit on the design. The average number of MRs per path and crossings per path were calculated based on the likelihood each path would be taken, and multiplying by the total number of the respective units on the path, thus giving a weighted average of each value. The power is based off of data from previous publications about their switch [6], or the results from simulation inside CLAP [24].

### B. Throughput Evaluation

Table III shows what the chances that a random flit will travel from any given input to any given output port when using XYZ routing, a 6x6x6 network and random traffic patterns. The table totals up to 0.999948, which means that .005% of the flits are unaccounted for in rounding error. This data provided a solid base for calculating the fault resiliency of the circuit types. From this evaluation, we can conclude that at without any faults present, the Fault Resilient Tolerant Three Dimensional Optical Router (FTTDOR) can achieve approximately 95.7% (according to equations 1-3 and random traffic, and no faults) of the throughput that a crossbar [25] can. It should be noted that a crossbar represents the maximum number of MRs a switch can use without wasting any. This is because only the dissected component switches are truly non-blocking, which accounts for the 4.3% that is missing. This number is given as a sacrifice to achieve fault tolerance and reduce the number of overall MRs on the chip. The partially blocking PHENIC switch has approximately 85.4% of the throughput of the crossbar switch.

### C. Complexity Evaluation

The next set of data represented in Table IV, is the comparison of MRs and waveguide crossings in each of the four 7x7 optical switches. This data is relevant for
calculating area in the optical layer, optical loss, and power consumption of the switches. The power of the switches is almost completely dependent on the number of active MRs in the path. This does not affect the setup latency all that much due to the way that the control signals are handled. It only gives the extra loss. The crossbar severely lacks in this regard because it always requires one to be on, while the PHENIC, Iris, and FTTDOR all automatically transfer light to the opposite side of the chip if uninterrupted by an active MR. The crossbar also takes up significantly more area because it uses significantly more MRs. As far as loss goes, the FTTDOR has the least amount of active MRs per path and least of amount of crossings per path. This is because the design only costs more area for some of the redundant MRs which allow for fault tolerance. Figure 5 shows the results of the failure resiliency calculations. These results show that at 3% failure rate, all 3 retain a similar value, with the FTTDOR at a slightly higher functionality. While at 20% failure, the FTTDOR can still maintain a little above 90% functionality, while the crossbar is exactly 80% and the original PHENIC switch is even worse than that.

### Table IV

| Microring Resonators and Waveguide crossings comparison results |
|------------------|------------------|-----------------|------------------|
| Waveguide Crossings | 32        | 19     | 49      | 48          |
| Microring Resonators | 29        | 22+10  | 49      | 32          |
| Avg Active MRs/Path | 56        | 52     | 1       | .575        |
| Avg Crossings/Path  | 8.1       | 5.6    | 7       | 8           |
| Power (mW)          | 486       | 473    | 700     | 500         |

### Table V

<table>
<thead>
<tr>
<th>Results of the Electronic Router’s Synthesis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power(uW)</td>
</tr>
<tr>
<td>-----------</td>
</tr>
<tr>
<td>361.064</td>
</tr>
</tbody>
</table>

### VI. Conclusion and Future Work

In this paper, we presented the architecture and evaluation of a fault-tolerant optical router. FTTDOR is targeted for highly-reliable low-power three dimensional Network-on-Chips and is decomposed, non-blocking, and with minimal MRs. In the case of a fault, it becomes a partially blocking switch, where all functionality is preserved assuming that one set of redundant MRs is not compromised. This particular switch was designed for XYZ routing, but can be used with other type of schemes.

Evaluation results show that the network can maintain a 98% throughput after 3% faults, and 95% after 20 % faults. These results come with 35% decrease in the number of MRs when compared to the conventional crossbar switch resulting in 32% power reduction. Compared to PHENIC and the conventional Crossbar switches, our proposed design is much more resilient to ring faults. Additionally, the Crossbar and Iris required more power, area, and rings. The original PHENIC switch required less rings, but more rings were active on a path, resulting in less area but more power.

As a future work, we plan to implement fault detection mechanism so that this can actually be implemented. Then the next step would be to simulate the full network functioning with the fault-tolerance capability. This was the problem mentioned in the related works, which was solved by Raik [21] in the electrical domain. I would also like to run real traffic patterns, to give more relevant results over the current traffic patterns. Currently, it is assumed that the control signals for the resonators has no fault, or a stuck-at-one fault which causes a MR to be permanently on. We will look into adding fault tolerance to the control signals.

### Acknowledgments

This work is supported by VLSI Design and Education Center (VDEC), the University of Tokyo, Japan, in collaboration with Synopsis, INC. and Cadence Design Systems,
REFERENCES


