# Resilient Microring Resonator Based Photonic Networks

Christopher J. Nitta[1]
cjnitta@ucdavis.edu

Matthew K. Farrens[1]
farrens@cs.ucdavis.edu

Venkatesh Akella[2]
akella@ucdavis.edu

[1]Department of Computer Science
[2]Department of Electrical & Computer Engineering
University of California, Davis
One Shields Avenue
Davis, CA, USA

## ABSTRACT

Microring resonator-based photonic interconnects are being considered for both on-chip and off-chip communication in order to satisfy the power and bandwidth requirements of future large scale chip multiprocessors. However, microring resonators are prone to malfunction due to fabrication errors, and they are also extremely sensitive to fluctuations in temperature. In this paper we derive a fault model for microring based optical links that can be used by computer architects to make informed design choices. We evaluate different schemes for improving resilience, such as retransmission versus error-correction, using an optical fault simulator based on our fault model. We show how meeting a target mean time between failures (MTBF) affects the choice of resilience scheme - our investigation indicates that until fault rates are in the range of $10^{-21}$ to $10^{-24}$ per cycle, error detection/correction schemes will be needed in order to meet a 1M hour MTBF. We also evaluate how the resilience scheme impacts the performance of the link, which will help an architect choose the appropriate scheme based on the throughput requirements of a particular design.

## Categories and Subject Descriptors

B.4.5 [**INPUT/OUTPUT AND DATA COMMUNICATIONS**]: Reliability, Testing, and Fault-Tolerance—*Hardware reliability*

## General Terms

ReliabilityTheory

## Keywords

photonic interconnect, fault model, microring, resilience

## 1. INTRODUCTION

As the number of cores on a chip continues to climb, architects will need to address both bandwidth and power consumption issues related to the interconnection network (both on- and off-chip). Since electrical interconnects do not scale well (mainly for latency and power consumption reasons), architects are now looking elsewhere for solutions. Optical interconnects have been shown to be a possible alternative, for both off-chip/chip-to-chip communication [3] and more recently for on-chip networks. Researchers have demonstrated the benefits of using optical interconnects between the CPU and DRAM [2, 20] to overcome pin limitations, and many recent studies [27, 8, 14, 25, 13] have documented the advantages of on-chip optical interconnects in terms of energy usage per bit and sustainable bandwidth.

The main enabling technologies for optical interconnects are Wavelength Division Multiplexing (WDM) and CMOS-compatible silicon microring resonators [18] WDM allows multiple data bits to be sent down an optical waveguide simultaneously, while microrings (which can be switched as high as 40GHz) are used to realize wavelength-selective modulators, demodulators, and switches. Microring resonators with a Free Spectral Range (FSR) of 62.5nm for 1550nm lasers have already been demonstrated [28], and this FSR could theoretically provide enough bandwidth for approximately 97 WDM channels at 40GHz - that is close to 3.8Tbps on a single waveguide. Furthermore, the authors of [1] estimate that on-chip photonic links can be highly energy efficient, requiring only tens of femtojoules per bit. This is far below the limit in electrical networks of thousands of femtojoules per bit, estimated by Miller in [22].

However, microring resonators are highly sensitive to fabrication inaccuracies [18, 17, 30] and temperature variations. A change of as little as 1°C can shift the resonance wavelength of a microring by as much as 0.1nm, causing it to respond to a completely different wavelength than intended. This problem can be dealt with using a technique called *trimming*, although trimming requires a significant amount of power [1] which negatively impacts one of the main advantage of optical interconnects (the low energy per bit). In addition, as discussed in [23], it is not clear that large-scale networks with over a hundred thousand rings (as proposed in [27] and [25]) can be continuously trimmed in a stable manner.

In many areas, such as hard disks, flash memory, wireless communication, and small geometry DRAMs, it is common to try to realize a reliable communication link using underlying components that may be unreliable, instead of attempting to make *all* the individual components reliable. In this paper we explore the feasibility of making optical links resilient - in other words, is it possible to tolerate some malfunctioning rings and still be able to communicate reliably using an on-chip optical link? We find this approach particularly appealing because of the surplus bandwidth that exists in a WDM based optical network, which can potentially be leveraged to improve the resilience of the network.

In order to evaluate reliability and resilience, one needs to know the type of faults that are likely to occur in the optical domain. Un-


*MICRO '11* December 3-7, 2011, Porto Alegre, Brazil

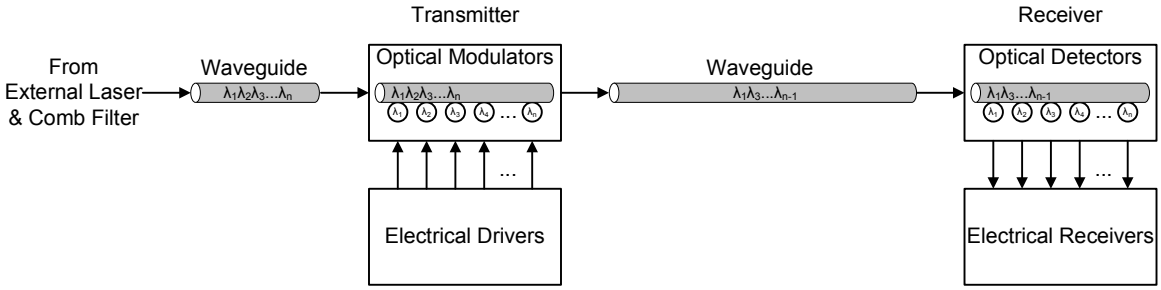
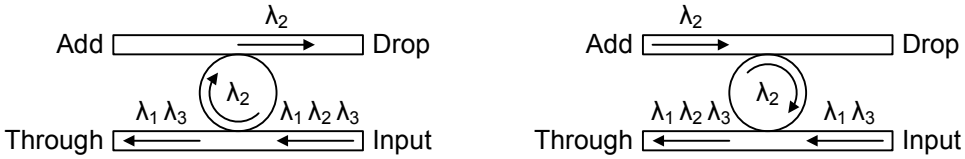



**Figure 1: Example Optical Link**




(a) Filtering $\lambda_2$ from Input to Drop

(b) Injecting $\lambda_2$ from Add to Through

**Figure 2: Microrings being used to filter (a) and inject (b) $\lambda_2$**

fortunately, since the fabrication of nanophotonic components is still in the nascent stage, there is very little in the literature on either the nature of defects or how to model them. Therefore, in this work we take a computer architect's perspective and propose simple abstractions (fault models) for the manifestation of the errors due to malfunctioning microresonator rings. We then use these fault models to evaluate techniques for realizing reliable optical links using error-detection and error-correction schemes. In particular, we identify the unique aspects of the defects and study their impact on modulation schemes and error recovery mechanisms, and use these results to determine the fault rate that must be attained in order to meet a given mean time between failures (MTBF).

The main contributions of this paper are: (1) Demonstrating that an error detection scheme will almost certainly be needed in order for large scale microring resonator-based networks to be realizable. (2) The derivation of simple fault models for microring-based optical links that can be used by a computer architect to explore performance, power, and area trade-offs and make informed design choices, (3) An evaluation of which scheme is the most appropriate in order to meet a target MTBF, provided the microring fault rate is known, and (4) An exploration of the trade-offs of different resilience schemes (such as retransmission versus forward error-correction) using an optical fault simulator.

The rest of the paper is organized as follows: we start by presenting a detailed overview of how optical links are realized in on-chip networks, and then examine the sources and nature of faults. Section 3 presents fault models for optical links, and Section 4 discusses the trade-offs between reliability of links and throughput. In Section 5 we describe an optical link fault simulator used to evaluate different error correction and detection schemes, and in Section 6 we present an MTBF analysis of the various schemes. Finally, in Section 7 we present an overview of related work and then we conclude with a short discussion in Section 8.

# 2. BACKGROUND & LINK COMPONENT DESCRIPTION

In order to address the resilience of optical communication, we first must examine how optical links are constructed. Figure 1 presents a typical on-chip optical link that uses an external laser as a light source. The external laser passes through a comb filter, which creates the necessary set of wavelengths used for communication, and then enters the chip. These wavelengths are delivered to the transmitter section of the source node via an optical waveguide.

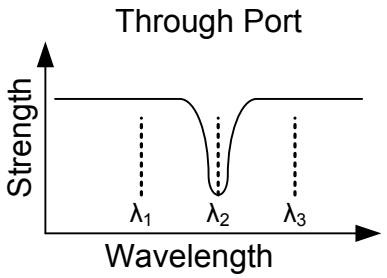
The transmitter, consisting of electrical drivers and optical modulators, uses the modulators to remove certain wavelengths (in this case $\lambda_2$ and $\lambda_n$), creating the desired pattern. (Generally, we assume that the presence of a wavelength represents a logic 1 and the absence represents a logic 0.) This pattern then travels down the waveguide from the source to the destination node. When the transmitted value arrives at the destination, the optical detectors convert the photonic power back to an electrical signal and the transmission is complete. Understanding in more detail how the individual components of the optical link function provides the basis for the creation of our fault models. We will focus our fault discussion on microring resonators, since they are the core component for modulating and filtering specific wavelengths in photonic links.

## 2.1 Microring Resonators

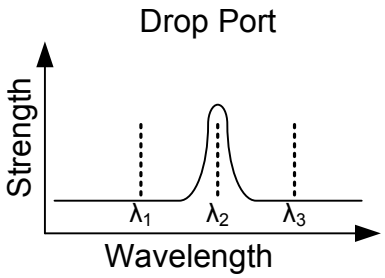
Microring resonators are designed to resonate only when presented with specific individual wavelengths, behaving in essence as band pass/reject filters. They are typically configured to have two input (*input* and *add*) and two output (*through* and *drop*) ports. Figure 2(a) shows a microring resonator filtering (removing) $\lambda_2$ from the *input* port, which carries multiple wavelengths. Figure 2(b) shows a microring resonator injecting $\lambda_2$ from the *add* port onto the *through* port, where it joins other wavelengths.

Figure 3 shows the theoretical *through* and *drop* power as a function of wavelength for a microring tuned to $\lambda_2$. The Y-axis of the graphs in Figure 3 represents the signal strength, which is the percent of power provided at the *input* port that makes it to the *through* port (Figure 3(a)) or the *drop* port (Figure 3(b))[1]. The function shown in Figure 3(a) represents both the percent of power that reaches the *through* port from the *input* port, as well as the per-

[1]The units were purposely left off since the actual values are not important, but ideally the top value would be 100% (0dB attenuation) and the bottom would be 0% ($-\infty$dB attenuation).




(a) Power at Through port, Microring Tuned to $\lambda_2$




(b) Power at Drop port, Microring Tuned to $\lambda_2$

**Figure 3: Power at Through (a) and Drop (b) ports for Microring Tuned to $\lambda_2$**

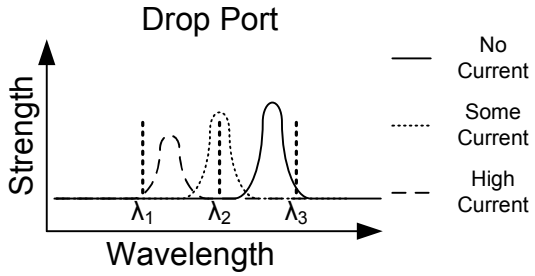



**Figure 4: Degradation in signal quality**

cent power that reaches the *drop* port from the *add* port. The same is true of Figure 3(b), which represents the percent of power for *input* → *drop* and *add* → *through*.

The resonance frequency of a microring can be changed by heating the microring, causing a shift towards the red end of the spectrum, or by electrically injecting current into the microring (which shifts the resonance towards the blue [1]). Microrings respond quickly enough to current injection that it can also be used for modulation; unfortunately, current injection causes a significant degradation in the quality of the modulated signal. Figure 4 shows that as current is injected, the resonance wavelength moves to the left and the strength of the signal decreases. Signal degradation due to current injection is a further complication to the implementation of reliable nanophotonic networks based on microring resonators. The proposed methods of modulation and blue shift trimming increase the likelihood that either insufficient light will transfer from the *input* port to the *drop* port (if filtering wavelengths) or from the *add* port to the *through* port (if injecting wavelengths).

## 2.2 Photonic Waveguides

Unlike electrical wires, photonic waveguides are designed to carry multiple bits of information along a single waveguide. Photonic waveguides have relatively low signal crosstalk [21] and are capable of carrying signals over a longer distance at higher signaling rates with lower losses than their electrical counterpart. Signals do suffer some losses in waveguides, however, due to effects such as scattering and radiation mode coupling. According to [17], waveguide scattering losses are highly dependent upon the fabrication process. Increased waveguide losses (or higher path attenuation) reduces the photonic power that will reach the photodetectors, and thus must be accounted for in the fault models.

## 2.3 Permanent vs. Temporary Faults

Faults can be classified as either permanent or temporary. Permanent faults are primarily due to fabrication errors, while temporary faults may be due to environmental factors such as fluctuations in temperature or EMI. Permanent faults that cannot be overcome using architectural resilience techniques will lower the fabrication yield of on-chip photonic networks, while temporary microring faults due to things like temperature fluctuations can cause higher path attenuation, since the shifting of the resonant wavelength due to temperature changes can increase the ring attenuation (if not perfectly corrected via *trimming*). However, the ultimate impact on the photonic network is whether or not faults manifest themselves as a bit errors, not if the fault is temporary or permanent.

# 3. PHOTONIC LINK FAULT MODELING

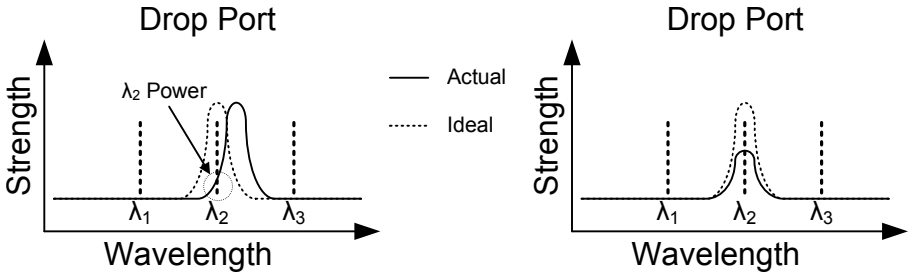
The faults that occur in the photonic components can result in a variety of different bit errors. In this section we will abstract the various low level faults due to defective microring resonators into a set of "fault models" that can be used by the architect when implementing a resilient photonic network.[2]

Microrings that do not resonate at their designed spectral position and waveguides with increased attenuation will be abstracted further. We consider microrings that do not resonate as designed to be faulty, which will happen if it is resonating to the wrong wavelength, the signal attenuation is too great, or both. The two exclusive cases are illustrated in the graphs in Figure 5. In this figure the dashed lines show the desired function, while the solid lines show the actual. Figure 5(a) shows that when the microring is not accurately tuned to the desired wavelength $\lambda_2$, the amount of $\lambda_2$ that appears on the *drop* port is very small - the amount of the $\lambda_2$ line that lies below the solid black line. This misalignment could be the result of thermal drift, improper fabrication, insufficient *trimming*, etc. Figure 5(b) shows the power to the *drop* port of a microring that is excessively attenuating the signal, which results from improper fabrication or too much current injection (as noted earlier, resonance deteriorates under current injection).

Some of the faulty optical components in an on-chip network could potentially be made to work correctly by increasing the amount of system power used. Off resonance microrings that are being trimmed, for example, simply require more signal power to operate correctly. Waveguides that have greater path attenuation may also be compensated for by increasing the photonic power, enough so that sufficient power reaches the photodetectors. We will focus on the cases for which addressing the fault will not be as simple as increasing the power.

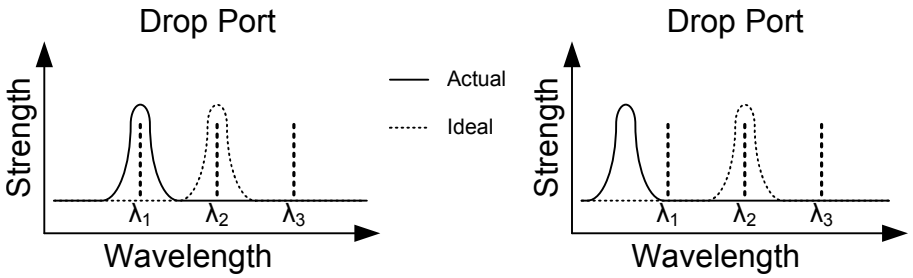
Microrings that do not resonate at their designed spectral position (as in Figure 5(a)) can be put into one of two categories: *inter-*

[2]We will follow the terminology and flow presented in [26]: Defect → Fault → Error → Malfunction → Degradation → Failure




(a) Drop for Off Resonance Microring (b) Drop for Attenuated Microring

**Figure 5: Drop for Off Resonance (a) and Attenuated (b) Microrings Designed to Resonate on $\lambda_2$**




(a) Drop for Interfering Microring (b) Drop for Non-Interfering Microring

**Figure 6: Drop for Interfering (a) and Non-Interfering (b) Microrings Designed to Resonate on $\lambda_2$**

*fering* or *non-interfering*. Interfering microrings are those whose frequency has drifted so far that they are actually resonating at another wavelength channel. Figure 6(a) shows the power to the *drop* port of an interfering microring that is designed to resonate at $\lambda_2$, but is interfering with $\lambda_1$. Non-interfering microring are those that do not resonate at the desired wavelength, but do not interfere with any other wavelength either. Figure 6(b) shows the power to the *drop* port of a non-interfering microring that is designed to resonate at $\lambda_2$, but is resonating below $\lambda_1$. Microrings that have increased attenuation are considered to be non-interfering microrings, since the end result is the same as a slightly off-resonance non-interfering microring (in both cases, a diminished amount of the desired wavelength appears on the output port.) Microrings that are partially interfering could also be viewed as being two non-interfering microrings, since both cases result in neither of the two wavelengths being properly transmitted or received.

## 3.1 Link Component Structure Dependent Errors

The types of errors that will result from faults depends upon the structure of the link components. We will focus on the transmitter and receiver sections of the on-chip optical networks, since the proposed networks all have similar transmitter/receiver structures and differ primarily in the interconnection topology. Transmitting data is done in one of two ways: by actively modulating ones (transitioning wavelengths from the *input* waveguide to *drop* waveguide) or by actively modulating zeros (removing wavelengths from the *through* waveguide). The receiver section for a link will consist of a set of microring resonators that are either always on-resonance (as in [27]), or enabled whenever a message is sent (as in SWMR, proposed in [25]).

### *3.1.1 Non-Interfering Microring Fault Errors*

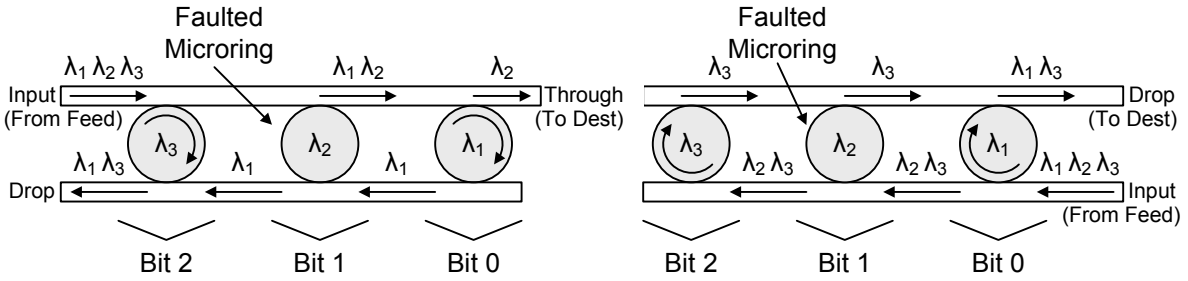
Non-Interfering faults do not move the desired wavelength from the *input* port to the *drop* port (or *add* port to *through*), but do not transition any other wavelength either. Thus, a non-interfering faulty microring that is in the receiver section will result in zeros always being received for that bit, since the proper wavelength will never transition from the *input* port to the *drop* port. This is essentially a "stuck-at-zero" fault, and only results in a bit error when a one is being sent on that bit.

The types of errors generated by a non-interfering faulted microring in the transmitter section will depend upon the method of modulation. In the case where zeros are actively modulated (a wavelength is removed from the *through* waveguide), a faulty microring will result in the wavelength always being present at the destination (a one will always be detected, which corresponds to a "stuck-at-one" fault). This is shown in Figure 7(a), where a three bit transmit section is attempting to send all zeros. The bit 1 modulator is faulty, so $\lambda_2$ is not being removed from the waveguide.

In the case where ones are actively modulated (a wavelength is transitioned from the *input* to *drop* ports), a faulted microring will result in its resonant wavelength never being present at the destination (a zero will always be detected, which corresponds to a "stuck-at-zero" fault). Figure 7(b) illustrates a three bit transmit section that is attempting to send all ones - again, bit 1 has a faulted modulator and therefore is not transitioning $\lambda_2$ from the *input* feed to the *drop*.

### *3.1.2 Interfering Microring Fault Errors*

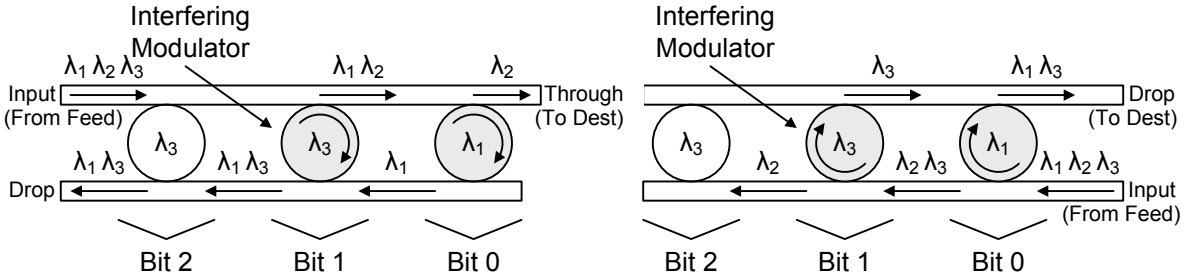
Interfering faults are much more problematic than non-interfering faults. It is possible for double bit errors to occur when an interfering faulted microring is involved, for example. Figure 8 illustrates




(a) Modulating Zero (Removing Wavelengths). Attempting to send 000, actually sending 100

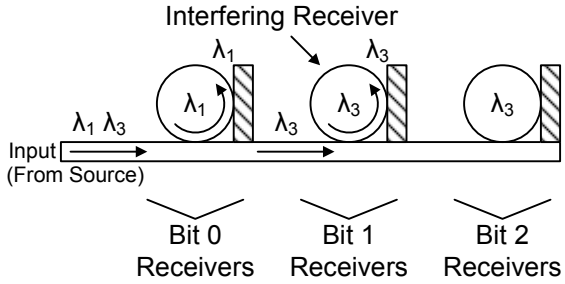
(b) Modulating Ones (Adding Wavelengths). Attempting to send 111, actually sending 101

**Figure 7: Single Errors due to Faulty Microring for Modulation of Zeros (a), and Modulation of Ones (b)**




(a) Modulating Zeros. Attempting to send 100, actually sending 101

(b) Modulating Ones. Attempting to send 011, actually sending 101




(c) Receiver. Sent a 101, reads a 110

**Figure 8: Double Bit Error from Interfering Microring Fault for Modulation of Zeros (a), Modulation of Ones (b), and Reception (c)**

double errors for both forms of modulation and for reception. Figure 8(a) shows a three bit transmit section attempting to transmit the value 100, but bit 1 is interfering with bit 2 ($\lambda_3$ is removed instead of $\lambda_2$), leading to the value 010 being sent. In figure 8(b), the transmit section is attempting to send the value 011, but again bit 1 is interfering with Bit 2 ($\lambda_3$ is transitioned instead of $\lambda_2$) causing a 101 to be sent. Finally, Figure 8(c) shows a three bit receive section that has been sent the value 101, but since bit 1 is interfering with bit 2 ($\lambda_3$ is removed by Bit 1 instead of by Bit 2) it reads a 110.

Interfering modulators will result in the interfer**ing** bit being "stuck-at" (similar to a non-interfering fault), and the interfer**ed** bit being a logical function of the interfering and interfered bits (similar to a *bridged* fault). In the case where zeros are actively modulated, the interfered bit will be a logical AND of the interfering and interfered bits, since either modulator will remove the wavelength in the case of a zero, and only both bits being a one will result in the wavelength passing unperturbed. In the case where ones are actively modulated, the interfered bit will be a logical OR of the interfering and interfered bits. The case where ones are actively modulated is symmetric to that of the case where zeros are actively modulated, as one might expect.

In the receive section, microrings that are resonating at another wavelength may or may not actually be interfering. Figure 8(c) shows that Bit 2 cannot interfere with Bit 0, even if it is resonating at $\lambda_1$. A microring resonating at another wavelength but not interfering behaves like a non-interfering microring ("stuck-at-zero"). However, in the case where one microring *is* interfering with another, the interfered bit will manifest as a "stuck-at-zero", and the interfering bit will receive the interfered bit's information.

## 3.2 Unidirectional Bit Errors

The choice of modulation and reception topology can lead to an asymmetry of errors when certain faults occur. It is clear that interfering faults can lead to double bit errors, but non-interfering faults lead to errors in a single direction. Non-interfering faults for receivers and modulators that actively modulate ones will only cause $1 \rightarrow 0$ bit errors, since they are "stuck-at-zero" faults. Increased path attenuation can also lead to $1 \rightarrow 0$ bit errors (since insufficient

photonic power to switch from $0 \rightarrow 1$ reaches the photodetector). On the other hand, modulators that actively modulate zeros will have faults that yield $0 \rightarrow 1$ bit errors. Non-interfering faults for components result in the following unidirectional bit errors:

**Modulator (Active Zeros)** – Light will not be successfully removed from the *through* waveguide. When zeros are sent, a one will be received ($0 \rightarrow 1$ bit error).

**Modulator (Active Ones)** – Light will not be successfully transitioned to the *drop* waveguide. When ones are sent, a zero will be received ($1 \rightarrow 0$ bit error).

**Receiver** – Light will not be successfully transitioned from the *input* to the photodetector. When ones are sent to it, a zero will be received ($1 \rightarrow 0$ bit error).

**Waveguide** – Increased waveguide attenuation results in insufficient light being received at the end of the waveguide. When ones are sent, a zero will be received ($1 \rightarrow 0$ bit error).

The type of single bit errors that will occur in a photonic link can be designed to be unidirectional if the correct link component structure is chosen. This is important, because unidirectional errors can be dealt with more efficiently, and if we are willing to give up some bandwidth and separate our channels more we may be able to minimize/eliminate interfering faults.

# 4. LINK RELIABILITY/THROUGHPUT TRADE-OFF

The performance/power vs. resilience trade-off is well understood in the electrical domain; unfortunately, given the nature of photonics, this trade-off is not as clear in the optical realm. An analysis of the average and peak throughput of an optical crossbar as a percentage of the entire network bandwidth during the execution of certain SPLASH-2 benchmarks shows that the average network utilization is quite low [24], so it is unlikely that the resilience techniques proposed in this paper will have much of an impact on overall performance. In fact, since much of the power consumed in a photonic link is static (the external laser and microring trimming, for example), the relative power penalty of retransmission or of error detection/correction techniques may be much lower than it would be with electrical links. In photonic systems the high static overhead means the cost of transmitting data is mostly pre-paid - the more you transmit, the lower the average cost/bit becomes.

Improving communication link reliability can be accomplished by increasing the probability that each transmission will be received correctly, by retransmitting until the transmission is received correctly, or both. Increasing the probability of a correct reception can be done using fairly straight-forward techniques, such as reducing the error rate (reducing the device fault rate) and/or adding bits in order to correct for errors. Retransmitting messages until they are properly received is a little more complicated, since it requires a feedback communication link and a communication protocol.

A common method of providing reliable data transmission over an unreliable communication channel is to use an Automatic Repeat reQuest (ARQ) protocol. In order to implement an ARQ protocol, errors must be detectable, and since the communication rate of on-chip networks is very high, the codes used for error detection must enable fast encoding and decoding. The implementation of an ARQ protocol and the additional error detecting bits will reduce the potential bandwidth of a given network, but will increase its resilience.

An ARQ protocol alone will not guarantee reliable communication for all fault sources. If the faults are permanent, for example, ARQ protocols will unsuccessfully repeat transmissions until the maximum retransmission count is reached. To circumvent this problem, a Hybrid ARQ (HARQ) protocol can be employed, which utilizes Forward Error Correction (FEC) in order to correct a small number of errors and only requests retransmission for uncorrectable cases. Thus, a HARQ protocol can make on-chip networks reliable even in the presence of some permanent faults, as long as they are correctable by the FEC. There are two main types of HARQ protocols: Type I, in which the FEC bits are sent with each transmission, and Type II, which sends error detecting bits with the initial transmission and only sends FEC bits if needed (FEC bits are not sent at all if the transmission is received correctly) [16]. Type II HARQ protocols do not map well to a parallel data path, because of mismatches in the sizes of flits - therefore, in this paper we focuses only on Type I HARQ protocols.

**Table 1: N Choose K Code Counts**

| N | K | Codes | Bits/Block | Efficiency |
|---|---|---|---|---|
| 2 | 1 | 2 | 1 | 50% |
| 4 | 2 | 6 | 2 | 50% |
| 6 | 3 | 20 | 4 | 66.7% |
| 8 | 4 | 70 | 6 | 75% |
| 10 | 5 | 252 | 7 | 70% |
| 12 | 6 | 924 | 9 | 75% |

## 4.1 Error Detecting Codes

Cyclic Redundancy Check (CRC) is one of the most widely used error detection codes in digital networks and storage devices. An $n$ bit CRC is capable of detecting any single error burst of up to $n$ bits in length. CRC may not be well suited for this environment, though, since communication is not a serial stream of bits (making burst errors less likely), and the block length is relatively short. Furthermore a CRC code is typically calculated in hardware using a Linear Feedback Shift Register (LFSR), which would have difficulty keeping up with the communication rates of on-chip networks (although parallel implementations exist [6, 19]).

Berger codes can detect any number of unidirectional bit errors with the addition of $k = \lceil log_2(n+1) \rceil$ check bits, where $n$ is the number of data bits [4]. The efficiency of the coding make Berger codes good candidates for use in this setting - unfortunately, Berger codes require the computation of the *weight* of the codeword, which is very expensive.

Extended Hamming codes for single error correction and double error detection (SECDED) have been used in a number of memory systems, including the CRAY-1. The same SECDED codes can be utilized as a triple error detection (TED) code, if no correction is performed. A SECDED or TED code can be implemented for 64 and 32 data bits with the addition of 8 and 7 check bits, respectively.

Another approach to error detection is to use multiple signals to transmit a single bit of information. This approach is commonly used in high speed communications, such as Low Voltage Differential Signaling (LVDS). Multi-Bit Differential Signaling (MBDS) [15] has been proposed to overcome the low code rate efficiency of LVDS, and has been suggested for use in short range (board to board) optical communication [7]. These approaches are essentially an N choose K (NcK) encoding - for example, LVDS is a 2c1 encoding, since only one of the two signals will be a one at any given time. NcK encodings can detect all odd number of bit errors, and may be able to detect some even number of bit errors. Significantly, NcK encoding can detect *any* number of unidirectional errors.

The number of valid codes in an NcK encoding is $\binom{n}{k} = \frac{n!}{k!(n-k)!}$. A single NcK block need not be used for the entire data width -

multiple NcK blocks could be used to implement a larger data path with error detection capability. Table 1 shows the number of codes and bits per block that can be encoded using various values for N and K. Looking at Table 1, it should be clear that a single 4c2 block does not improve the coding efficiency over two 2c1 blocks (although two 4c2 blocks does yield 36 codes, which is sufficient to encode 5 bits). Encoding and decoding of the NcK blocks will need to be efficient in order to work at the speeds necessary in this environment.

## 4.2 Forward Error Correction (FEC)

As discussed previously, HARQ requires Forward Error Correction (FEC). One possible FEC code that could be used in HARQ is the extended Hamming SECDED code - another approach is to combine the NcK encoding with either a parity block or a Reed Solomon code. Since any odd number of bit errors can be detected with an NcK encoding, the detected errors could be treated as block erasures, and an additional parity block could be used to recover from a single erasure (as is commonly done in RAID-5).

The redundancy could be extended to protect against double erasures as in RAID-6, as long as the size of the Galois Field (GF) being used for the Reed Solomon code blocks is large enough. A $\mathbf{GF}(2^n)$ can cover $2^n - 1$ data blocks; therefore, a 2c1 code could only cover a single data block, while a 6c3 could cover 15 data blocks ($2^4 - 1$) or up to 60 bits of data. Equations 1 and 2 show how parity and the Reed-Solomon code is calculated, respectively:

$$\mathbf{P} = D_0 + D_1 + ... + D_{n-1} \tag{1}$$

$$\mathbf{Q} = g^0 \cdot D_0 + g^1 \cdot D_1 + ... + g^{n-1} \cdot D_{n-1} \tag{2}$$

In these equations, "addition" is handled by an XOR, and "multiplication" is done in the Galois Field. At first glance it may seem that calculating the GF multiplication may be too complex, but since we are proposing only a 4-bit code word and the multiplication is being done with a constant value, it can be realized with a simple look-up-table.

# 5. EXPERIMENT

Since this area is so new, there are no measured fault and error rate numbers to work with. Therefore, we developed an optical link simulator, which uses statistical sampling to determine the average rate of error for various proposed detection/correction schemes.

The simulator places each transmission into one of 5 categories: **correct** (no bit errors occur), **incorrect** (an error goes undetected), **detected error** (the encoding detects an uncorrectable error), **corrected** (an error is detected and corrected properly), or **corrected wrong** (an error is detected but corrected improperly).

The simulator takes as input the number of faults $F$ (which can be interfering or non-interfering), the encoding scheme, and the number of samples to perform. The simulator selects the $F$ rings which will have faults at random, and then a (random) bit pattern is created and sent to the receiver. The simulator determines the pattern that is detected at the receive side, and if it is correct, incorrect, has a detected error, is corrected, or is corrected wrong. This process is repeated for each sample until the count reaches the desired number (10M samples per configuration in our case). By doing this statistical sampling we can determine the average rate of corrects, incorrects, etc. for a given number of faults. Using a set probability of a ring faulting, we can also determine the probability of 1 fault, 2 faults, etc. All this information can be combined to enable us to determine the probability of an undetected error given a set probability of a single ring faulting.

The error detection techniques we simulated were 32-bit TED (TED32), 64-bit TED (TED64), 32-bit 2c1 (2c1-32), and 32-bit 6c3 (6c3-32). The error correction techniques we examined were 32-bit SECDED (SECDED32), 64-bit SECDED (SECDED64), 32-bit 2c1 with parity (2c1p-32), 32-bit 6c3 with parity (6c1p-32), and 32-bit 6c3 with Reed-Solomon (6c1rs-32). Table 2 shows the number of non-interfering and interfering microrings that each technique is guaranteed to detect or correct, respectively. The nomenclature of 0's and 1's refers to whether zeros or ones were being actively modulated. Notice that for NcK protocols, error detection and correction capabilities are greater when ones are actively modulated than when zeros are.

In order to evaluate the impact on network throughput of using the ARQ and HARQ protocols, a separate link simulator was developed. The simulator determines the average throughput per cycle given link latency, error rates (for both data and feedback channels), data path width, and packet width. For the sake of brevity the results presented here assume an error-free feedback link. We did do simulations in which the feedback channel was faulty, but our results indicated that the throughput is more dependent upon the error rate of the data channel than it is upon the error rate of the feedback link.

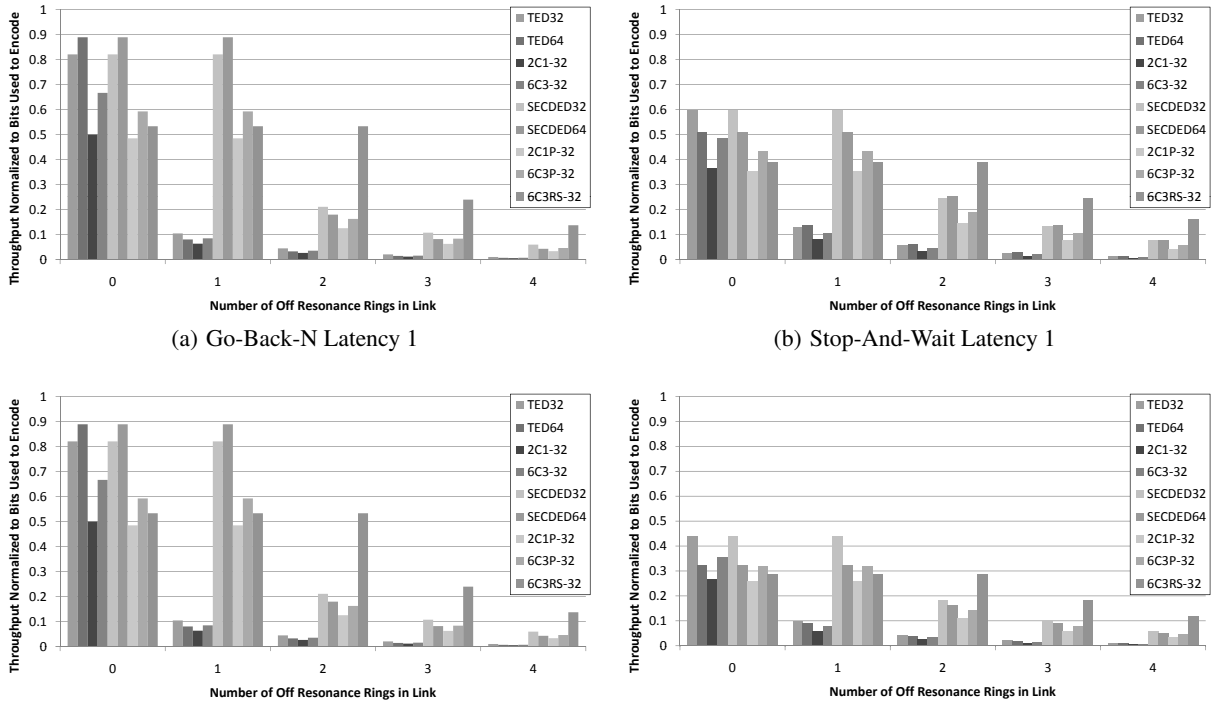
Only two of the three main types of ARQ protocols (Go-Back-N and Stop-And-Wait) were evaluated when calculating the maximum throughput. Selective Repeat was not analyzed since it requires the packet segments or flits each contain a unique segment identification number for the selective retransmission, and this additional information would greatly impact the payload capacity of each segment. The Go-Back-N protocol is relatively simple to implement and has the potential for maximal utilization of the data channel, although it does rely on continual transmission of unacknowledged packets. While this is a problem for some technologies, it is not for photonic systems - the fact that the external laser power is a fixed cost means that the more you transmit the lower the average cost/bit becomes. The use of a Stop-And-Wait protocol in the optical domain is not a completely new concept, either - the authors of [8] suggests the use of a similar protocol for signalling a dropped packet in the Phastlane architecture. A NAK only Stop-And-Wait will not protect against errors in the feedback link, since it is impossible to distinguish between a lost NAK and a NAK never being sent, but this protocol could be extended to provide reliable communication by changing the NAK to an ACK and retransmitting if the ACK is not received.

Figure 9 shows the maximum throughput results for the two ARQ protocols, with the normalized throughput on the Y axis and the number of non-interfering microrings that were faulted on the X axis. The throughput results are normalized to the number of bits required for each encoding. Packets of 256-bits were assumed, and ones were actively modulated. Notice that the Go-Back-N protocol appears immune to link latency - this is because the window size is sufficient for the latencies presented. Eight cycles are required to transmit a single packet with a 32-bit data channel, so a window size of two is enough to receive an acknowledgement before "going back n" even when the link latency is three cycles. It should also be noted that the SECDED codes are only capable of reliably correcting or detecting up to two misbehaving rings - the results for three and four off resonance rings are provided purely for the sake of comparison.

In the absence of faults, TED and SECDED are the most efficient coding schemes analyzed in terms of effective data throughput given the amount of bandwidth used to encode. However, as faults begin to occur (and errors are manifested), there are noticeable differences between the schemes. SECDED in particular performs

**Table 2: Error Detection/Correction**

| Encoding | Max Detect | | | | Max Correct | | | |
|---|---|---|---|---|---|---|---|---|
| | Non 0's | Non 1's | Int 0's | Int 1's | Non 0's | Non 1's | Int 0's | Int 1's |
| TED32 | 3 | 3 | 1 | 1 | N/A | N/A | N/A | N/A |
| TED64 | 3 | 3 | 1 | 1 | N/A | N/A | N/A | N/A |
| 2c1-32 | 1 | ANY | 0 | 0 | N/A | N/A | N/A | N/A |
| 6c3-32 | 1 | ANY | 0 | 0 | N/A | N/A | N/A | N/A |
| SECDED32 | 2 | 2 | 1 | 1 | 1 | 1 | 0 | 0 |
| SECDED64 | 2 | 2 | 1 | 1 | 1 | 1 | 0 | 0 |
| 2c1p-32 | 2 | ANY | 1 | 1 | 1 | 1 | 0 | 0 |
| 6c3p-32 | 2 | ANY | 1 | 1 | 1 | 1 | 0 | 0 |
| 6c3rs-32 | 2 | ANY | 1 | 1 | 1 | 2 | 0 | 0 |




(a) Go-Back-N Latency 1

(b) Stop-And-Wait Latency 1

(c) Go-Back-N Latency 3

(d) Stop-And-Wait Latency 3

**Figure 9: Normalized Throughput vs. Number of Off-Resonance Microrings for GBN (a), (c) and SAW (b), (d)**

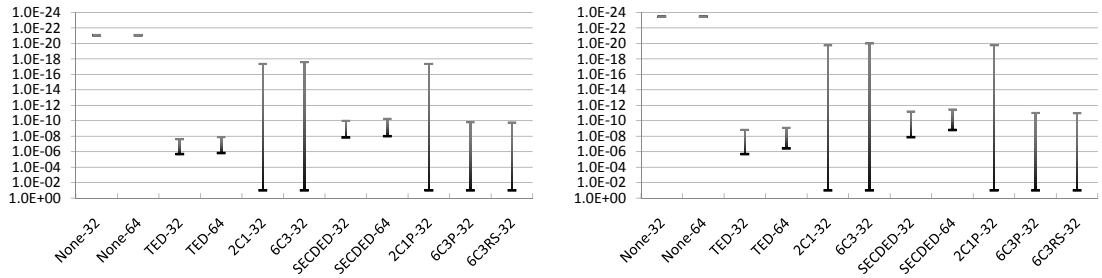
better than TED when there are single and double non-interfering faults, since a double fault does not ensure there is a double error.

What may seem somewhat counter-intuitive is that the 64-bit versions of TED and SECDED are less efficient than their 32-bit counterparts when the Stop-And-Wait protocol is used. This result is a byproduct of the fact that 64-bit versions have more "unused" bandwidth while waiting for the acknowledgement. Since the number of cycles that the link is stopped is the same for all encodings, the 64-bit versions wind up having a lower *utilization* of the link.

These results show that the use of FEC can dramatically improve throughput if microring faults are occurring. The efficiency of the SECDED encoding for single ring faults is evident - it is likely a desirable choice when single ring faults are common, and three or more microring faults rarely occur (since in that case errors could be corrected wrong or go undetected). The 6c1rs-32 encoding, on the other hand, clearly has the most efficient throughput in the case of two or more non-interfering faulted microrings. The choice of which encoding and protocol to use will be driven by the reliability of the underlying nanophotonic devices.

# 6. MEAN TIME BETWEEN FAILURE ANALYSIS

In order to justify choosing one encoding scheme over another one must know both the microring fault rate and the rate of interfering vs. non-interfering faults. Since this information is not yet available, we have taken a different approach; we have determined the fault rate that microrings must attain in order to meet a particular Mean Time Between Failure (MTBF) for a single link, and also for an entire network (such as a photonic torus). These calculations can not only guide architects in the choice of encoding schemes once microring resonators mature, but equally as important these results provide goals and targets for device researchers and manufacturers. The MTBF for a link can be calculated given the fault rate, the probability that a fault is interfering/non-interfering, and




(a) 1M hr Link MTBF

(b) 1M hr Torus MTBF

**Figure 10: Required Microring Fault Rate to Attain 1M hr MTBF for a Link (a) and a 8-ary 2-cube Torus (b) by Encoding Scheme**

the simulation results from Section 5. Since the target MTBF is known, the fault rate is simply varied until the target MTBF is achieved. Figure 10 shows the required microring fault rate given a particular encoding and a desired MTBF of 1M hours, for a single link (10(a)) and an 8-ary 2-cube Torus (10(b)). The torus was assumed to have direct links between the nodes (no microrings were assumed for routing). The Y-axis is the required fault rate that must be attained, with a lower fault rate (meaning higher quality microrings) being higher on the axis.

The spread of the values is due to different rates of interfering faults. The values at the lower fault rates assume the probability of a fault being interfering has a uniformly random distribution – in other words, the probability of an interfering fault is the percent of the Free Spectral Range (FSR) that other channels occupy. The values requiring a higher fault rate, on the other hand, assume the resonance point of a microring will drift from the desired point based on a normal distribution, centered at the desired resonance frequency (which yields a dramatically lower rate of interfering faults.)

The results show that the NcK encodings such as the 2c1-32 or the 6c3rs-32 are the best choice when fault rates are very high but the rate of interfering faults is very low. The Hamming codes are best when the fault rates are moderately high, with TED winning out over SECDED if correcting for fabrication errors is not a concern. In order for nanophotonic links/topologies to meet a 1M hour MTBF without using error detection or correction schemes, microrings will need to be fabricated such that fault rates are in the range of $10^{-21}$ to $10^{-24}$/cycle. The conservative assumption we used in our simulations (that any undetected bit error will result in a failure) means these numbers are probably a little high, but it is unlikely that the actual bit error rate that results in failure will change these results by very much (certainly not orders of magnitude.) Given the current immature state of the technology, it is clear that some type of error detection scheme will be needed if large scale microring resonator-based networks are to become a reality. Microring-based photonic networks that do not implement error detection or correction schemes will be inherently unreliable due to their low MTBF.

# 7. RELATED WORK

In on-chip nanophotonic networks, the use of the ARQ protocol was proposed by the authors of [8]. They suggest the use of a Stop-And-Wait protocol designed to request retransmission in the case of a dropped packet. The Phastlane architecture presented in [8] uses a NAK to signal that the packet was dropped due to insufficient buffer space. If the NAK is not received within a specific time window it is assumed that the packet was successfully buffered. This scheme could be extended to provide reliable communication with the addition of error detection bits and the additional hardware to make the feedback channel reliable as is done in [12].

Recently in [29] the use of CRC was proposed for the Macrochip system. Many of the links in the Macrochip design are serial or pseudo-parallel and have maximum packet payloads of 4KB, making CRC suitable for the application. The focus in [29] is on Bit Error Rate (BER), and is not concerned with the source of the errors.

Nanophotonic interconnects do have the potential for being unreliable, but electrical on-chip interconnects are not expected to be error free either - especially when implemented using very deep submicron technology [5, 9]. Fu and Ampadu in [12] investigates the use of a Type-II HARQ protocol for electrical interconnects. In this study Hamming product codes with Type-II HARQ are compared against Hamming, ARQ with CRC-5, Extended Hamming (SECDED), and Bose-Chaudhuri-Hocquenghem (BCH) for delay, area and power efficiency. (Reed-Solomon codes can be considered a non-binary BCH code). It is interesting to note that Fu chose not to implement the CRC-5 with a LFSR, opting for a more complex implementation. A Go-Back-N protocol was implemented for the ARQ portion of the work, and triple modular redundancy was implemented to protect the ACK/NAK feedback signal.

Fu and Ampadu also investigated the use of a dual-mode HARQ scheme in [11]. The proposed scheme uses a SECDED code for 64 bits, or four 16 bit SECDED codes in the case of a high noise environment. The use of Extended Hamming codes for both encoding techniques allows hardware sharing, which increases the area only slightly. The proposed scheme yields up to a 35% energy improvement compared to previous solutions - however, it is unclear if the signal interleaving that is beneficial in the dual-mode work would be as beneficial in a WDM environment.

The authors in [10] investigates the energy efficiency and performance of ARQ, FEC, and HARQ in on-chip networks. The ARQ scheme utilizes a CRC-8 that was implemented with a parallel bit code generator, while the FEC scheme analyzed uses overlapping parity bits (assumed Hamming). The results showed the trade-offs of performance and energy, and depending upon the environment (voltage swing, noise power, wire length, etc) one scheme may be better than another.

# 8. DISCUSSION AND CONCLUSIONS

In this paper we show show that the enormous (often surplus) bandwidth of an optical link can be leveraged to realize a variety of schemes for improving the reliability of an optical link, and that the computer architect needs to choose the appropriate scheme based on the topology, error rate and other parameters. We also show that the fault rates for photonic microrings must be very low before optical networks can be implemented without using any error

correction or error detection schemes - some sort of error detection scheme will almost certainly be needed by large scale microring resonator-based networks if they want to meet a 1M hour MTBF.

Another possible approach to making networks resilient is to use redundant paths in the network to detect and avoid links that are bad. This is a well studied problem in traditional networking, and certainly the techniques proposed here can be used to *detect* an erroneous link and modify the routing scheme used. However, in optical networks, rings and buses are more widely used due to their ease of layout, and unfortunately, these topologies are not particularly suitable for exploiting redundant links. In this paper we only considered the resilience of a single link. In the future, we plan to extend it to more complex topologies, which would involve modeling the intermediate optical switches.

# 9. REFERENCES


[1] J. Ahn, M. Fiorentino, et al. Devices and architectures for photonic chip-scale integration. *Applied Physics A: Materials Science & Processing*, 95:989–997, June 2009.

[2] S. Beamer, C. Sun, et al. Re-architecting dram memory systems with monolithically integrated silicon photonics. In *Proceedings of the 37th annual international symposium on Computer architecture*, ISCA '10, pages 129–140, New York, NY, USA, 2010. ACM.

[3] A. F. Benner, M. Ignatowski, et al. Exploitation of optical interconnects in future server architectures. *IBM J. Res. Dev.*, 49(4/5):755–775, 2005.

[4] J. Berger. A note on error detection codes for asymmetric channels. *Information and Control*, 4(1):68 – 73, 1961.

[5] F. Caignet, S. Delmas-Bendhia, and E. Sicard. The challenge of signal integrity in deep-submicrometer cmos technology. *Proceedings of the IEEE*, 89(4):556 –573, Apr. 2001.

[6] G. Campobello, G. Patane, and M. Russo. Parallel crc realization. *IEEE Transactions on Computers*, 52:1312–1319, 2003.

[7] D. M. Chiarulli, S. P. Levitan, et al. Efficient optical communications using multibit differential signaling. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 6126 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, pages 140–147, Mar. 2006.

[8] M. J. Cianchetti, J. C. Kerekes, and D. H. Albonesi. Phastlane: a rapid transit optical routing network. *SIGARCH Comput. Archit. News*, 37(3):441–450, 2009.

[9] C. Constantinescu. Trends and challenges in vlsi circuit reliability. *Micro, IEEE*, 23(4):14 – 19, 2003.

[10] A. Ejlali, B. Al-Hashimi, et al. Performability/energy tradeoff in error-control schemes for on-chip networks. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 18(1):1 –14, 2010.

[11] B. Fu and P. Ampadu. A dual-mode hybrid arq scheme for energy efficient on-chip interconnects. In *Nano-Net*, volume 3 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 74–79. Springer Berlin Heidelberg, 2009.

[12] B. Fu and P. Ampadu. On hamming product codes with type-ii hybrid arq for on-chip interconnects. *Trans. Cir. Sys. Part I*, 56:2042–2054, September 2009.

[13] G. Hendry, S. Kamil, et al. Analysis of photonic networks for a chip multiprocessor using scientific applications. *Networks-on-Chip, International Symposium on*, 0:104–113, 2009.

[14] N. Kirman, M. Kirman, et al. Leveraging optical technology in future bus-based chip multiprocessors. In *MICRO 39: Proceedings of the 39th Annual IEEE/ACM International Symposium on Microarchitecture*, pages 492–503, Washington, DC, USA, 2006. IEEE Computer Society.

[15] S. P. Levitan, D. M. Chiarulli, et al. Power efficient communication using multi-bit-differential signaling. In *16th Annual IEEE-LEOS Workshop on Interconnections within High-Speed Digital Systems*. IEEE, May 2005.

[16] S. Lin and D. J. Costello, Jr. *Error Control Coding*. Prentice Hall, Upper Saddle River, NJ, Jun 2004.

[17] M. Lipson. Guiding, modulating, and emitting light on silicon-challenges and opportunities. *Lightwave Technology, Journal of*, 23(12):4222–4238, Dec. 2005.

[18] M. Lipson. Compact electro-optic modulators on a silicon chip. *Selected Topics in Quantum Electronics, IEEE Journal of*, 12(6):1520–1526, Nov.-dec. 2006.

[19] W. Lu and S. Wong. A fast crc update implementation. In *Proceedings of the 14th Annual Workshop on Circuits, Systems and Signal Processing,*, pages 113–120, November 2003.

[20] P. Mejia, R. Amirtharajah, M. Farrens, and V. Akella. Performance evaluation of a multicore system with optically connected memory modules. In *Networks-on-Chip (NOCS), 2010 Fourth ACM/IEEE International Symposium on*, pages 215 –222, May 2010.

[21] D. Miller. Rationale and challenges for optical interconnects to electronic chips. *Proceedings of the IEEE*, 88(6):728–749, Jun 2000.

[22] D. Miller. Device requirements for optical interconnects to silicon chips. *Proceedings of the IEEE*, 97(7):1166 –1185, july 2009.

[23] C. Nitta, M. Farrens, and V. Akella. Addressing system-level trimming issues in on-chip nanophotonic networks. In *High Performance Computer Architecture, 2011. HPCA 2011. IEEE 17th International Symposium on*, Feb. 2011.

[24] C. J. Nitta. *Design and Analysis of Large Scale Nanophotonic On-Chip Networks*. PhD thesis, University of California, Davis, 2011.

[25] Y. Pan, P. Kumar, et al. Firefly: illuminating future network-on-chip with nanophotonics. *SIGARCH Comput. Archit. News*, 37(3):429–440, 2009.

[26] B. Parhami. A multi-level view of dependable computing. *Computers & Electrical Engineering*, 20(4):347 – 368, 1994.

[27] D. Vantrease, R. Schreiber, et al. Corona: System implications of emerging nanophotonic technology. In *ISCA '08: Proceedings of the 35th International Symposium on Computer Architecture*, pages 153–164, Washington, DC, USA, 2008. IEEE Computer Society.

[28] Q. Xu, D. Fattal, and R. G. Beausoleil. Silicon microring resonators with 1.5-$\mu$m radius. *Opt. Express*, 16(6):4309–4315, 2008.

[29] X. Zheng, P. Koka, et al. Energy-efficient error control for tightly coupled systems using silicon photonic interconnects. *J. Opt. Commun. Netw.*, 3(8):A21–A31, Aug 2011.

[30] L. Zhou, K. Okamoto, and S. Yoo. Athermalizing and trimming of slotted silicon microring resonators with uv-sensitive pmma upper-cladding. *Photonics Technology Letters, IEEE*, 21(17):1175–1177, Sept.1, 2009.