Runtime Techniques to Mitigate Soft Errors in Network-on-Chip (NoC) Architectures

Travis Boraten, *Student Member, IEEE*, and Avinash Karanth Kodi, *Senior Member, IEEE*

***Abstract*—As aggressive scaling continues to push multiprocessor system-on-chips (MPSoCs) to new limits, complex hardware structures combined with stringent area and power constraints will continue to diminish reliability. Waning reliability in integrated circuits will increase the susceptibility of transient and permanent faults. There is an urgent demand for adaptive error correction coding (ECC) schemes in network-on-chips to provide fault tolerance and improve overall resiliency of MPSoC architectures. The goal of adaptive ECC schemes should be to maximize power savings when faults are infrequent and increase application speedup by boosting fault coverage when faults are frequent. In this paper, we propose runtime adaptive scrubbing (RAS), a novel multilayered error correction and detection scheme with three modes of operation enabled by an area-efficient configurable encoder for encoding packets on the switch-to-switch (s2s) layer, thus preventing faults from accumulating up the network stack and onto the end-to-end layer. As fault rates fluctuate we propose a dynamic methodology for improving fault localization and intelligently adapt fault coverage on demand to sustain graceful network degradation. RAS successfully improves network resiliency, fault localization, and fault coverage as compared to traditional static s2s schemes. Simulation results demonstrate that static RAS improves network speedup by 10% for Splash-2/PARSEC benchmarks on a 8 × 8 mesh network while reducing area overhead by 14% and incurring on an average 6.6% power penalty by boosting fault tolerance when fault rates increase. Further, our dynamic RAS scheme maintains 97.88% of network performance for real applications while incurring 20% power penalty.**

***Index Terms*—Error correction, fault tolerance, on-chip communication networks, reconfigurable architectures, reliability.**

I. Introduction

IN THE world of integrated circuits, technological advancements in nano-fabrication are paramount if chip designers are expected to sustain Moore's law and extract maximum performance. Moore's law predicts the number of transistors in future integrated chips will double every two years. To eclipse performance goals in each technology generation, fabrication technology must overcome transistor sizing, speed, and overall chip density limits. In the past, designers leveraged aggressive frequency scaling and parallelism to improve performance by exploiting available chip densities. Today, emerging technologies has expanded research into the age of fully embedded multiprocessor system-on-chip (MPSoC) architectures and driven the integration of hundreds to thousands of cores [1]. With more cores in tandem on the chip, increased simultaneous memory access and parallel execution will allow for future performance gains. To meet the high performance communication requirements needed to support multiple cores, packet-based network-on-chip (NoC) architectures have emerged as the leading strategy to deliver scalable, modular, and adaptive networks in MPSoCs [2], [3]. Traditional bus-based networks can no longer overcome the fundamental challenges associated with wire scaling (due to high capacitance and resistance) and increased wire delay [4], [5]. Today, commercial NoC-based interconnects have already been proposed and implemented in many MPSoCs such as Tilera's 72-core architecture [6], Intel's 80-core TeraFlops [7], and NVIDIA's CUDA 512-core Fermi architecture [8]. As NoCs will continue to be the backbone of future on-chip architectures, exhaustive research has expanded the field of NoCs into several directions: network topologies [3], [9], routing algorithms [10], [11], router micro-architectures [12], [13], security [14], and fault tolerance [15].

According to the International Technology Roadmap for Semiconductors [16], SoC power consumption trends show an alarming increase in overall chip power consumption as technology scales below to the submicron regime. In efforts to meet shrinking power budgets, the power consumption for transistors will continue to worsen as feature sizes decrease with increasing leakage current leading to the rise of dark silicon [17]. For NoCs, this challenge is already apparent as expensive buffers consume 46% of overall router power [18] and total NoC power consumption reaches 28% of total chip power as seen in Intel's teraflops architecture [7].

Stringent area and power constrains also create further reliability challenges for complex hardware structures since scaling transistors are more susceptible to transient and permanent faults. Since NoCs act as the sole transport of data between cores, the rising unpredictability of faults makes ensuring data integrity and graceful network degradation a top concern for further research. To prevent the possibility of isolating the cores, NoCs must employ adaptive schemes or tailor error-tolerant techniques to handle a wide array of irregular effects caused by cross-talk, process variation, and multiple-bit upsets [19]–[21]. The rise in soft errors and dynamic fault rates observed in [22] further provides motivation for NoCs

Manuscript received August 24, 2016; revised November 5, 2016; accepted January 15, 2017. Date of publication February 6, 2017; date of current version February 16, 2018. This work was supported by the NSF under Grant CCF-1054339 (CAREER), Grant CCF-1420718, Grant CCF-1318981, and Grant CCF-1513606. This paper was recommended by Associate Editor L. P. Carloni.

The authors are with the Department of Electrical and Computer Science, Ohio University, Athens, OH 45701 USA (e-mail: kodi@ohio.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCAD.2017.2664066

to incorporate dynamic fault tolerant techniques as a 10 °C increase in router temperature results in a 2% increase in fault rates [22] caused by timing and process variation. For NoCs to effectively improve fault tolerance and prevent systematic failure, multiple approaches must be taken to handle not only the wide variety of faults but also fault localization as both impact performance as well as the proposed fault tolerant technique to handle the fault. With adaptive schemes, failure-prone data paths can be optimized to maximize performance and increase overall chip efficiency by improving resiliency. When faults are infrequent, fault coverage should be reduced to augment power savings and increased on demand to ensure graceful performance degradation.

In this paper, we propose runtime adaptive scrubbing (RAS), a novel multilayered error correction and detection scheme with an area efficient three-mode configurable encoder that encodes packets on the end-to-end (e2e) layer and optimizes fault coverage on the switch-to-switch (s2s) layer in NoCs to sustain application speedup and power savings. With RAS, higher fault granularity and graceful network degradation can also be achieved by tailoring error correction codes (ECCs) to extend the life (usage) of error-prone links affected by both transient and permanent faults in the datapath. We extend the proposed work in [23] by fully utilizing the RAS framework and propose a novel a dynamic transition model for hybrid encoding schemes with variable strength encoders. Our analysis was performed by evaluating threshold selections and the bit error rate (BER) as a function of network parameters. With our proposed dynamic model, dynamic ECC schemes can maintain 97.88% of performance, while incurring an average of 20% power penalty as an upper bound when compared to fault free networks.

In summary, the major contributions of this paper are as follows.

1) We propose a heuristic dynamic transition model for hybrid encoding schemes utilizing RAS variable strength encoders.
2) We minimize performance impact while providing an upper bound on the energy-delay-product (EDP) penalty.

This paper is organized as follows. In Section II, we discuss prior work; in Section III, we first summarize our design of static RAS and then our proposed dynamic RAS technique; in Section IV, we discuss the performance evaluation of our proposed static and dynamic RAS techniques and in Section V, we conclude this paper.

II. Prior Work

As electrical interconnects scale with increasing MPSoCs, challenges to support rising link widths and complex router micro-architectures, will increase with each technology generation. For faulty control paths, light-weight invariance checkers [15] and built-in-self-test (BIST) [24] have been proposed to diagnose faults. Fortunately, in the event of transient upsets within datapaths, instantaneous fault detection can be provided by the combination of forward error correction (FEC) and error detection codes (EDCs). Murali *et al.* [25], a survey of error recovery schemes, evaluated the impact of encoding methods on network performance and power-efficiency. In their experiments they showed e2e and s2s performed better under certain scenarios and hybrid schemes improved the most performance when experiencing higher fault-rates. When faults rates are low, robust s2s encoding schemes consume excess power which is unnecessary and can be avoided by e2e schemes. At the same time, s2s schemes can provide better fault coverage than e2e schemes when fault rates are high, especially in large hop networks. The choice of either e2e or s2s fault detection scheme in NoCs has power, area and performance tradeoffs that vary with the number of CMPs and fault rate in the system. As EDCs scale better than FECs, EDCs are typically adopted in e2e schemes. But EDCs take more cycles to compute, and force costly retransmissions when faults are detected. Park *et al.* [26] evaluated buffer size and placement to improve latency and power efficiency for hop-by-hop retransmission. To alleviate the cost of retransmissions, light weight FECs have been adopted in s2s configurations. However, error detection capability is typically limited to a few bits so that the encoders neither disrupt router pipeline clock cycles nor consume significant area. A strong EDC adds sufficient redundancy, but is wasted when fault rates are low; whereas, a weak EDC allows faults to go undetected and cause retransmissions. Therefore, an adaptable solution that can account for variations in transient and permanent faults in an efficient manner is required.

Yu and Ampadu [27] proposed a dual-layer method for handling transient and permanent faults. Instead of rerouting around links causing faults, a configurable error correction coder offers adjustable fault tolerance for transient faults. To account for permanent faults, each link has a set of spare redundant wires unused until a dedicated history table diagnoses permanent wire damage in the network. If transient fault coverage needs to be boosted, the link encoders can switch from using a two-way interleaved single error correction (SEC) Hamming code to a four-way SEC Hamming code. When encoders switch to the second mode, the serialization of data must split flits in half to account for the additional bits needed for the extra error correction capability. The alternative is shutting down the link. When a link is shutdown routers must reroute packets around them, extending the average hop count which increases the average packet latency and overall power consumption.

Reconfiguration, however, comes at the cost of complexity. Changes in the topology may also create paths that violate existing rules to prevent dead-locks. Fick *et al.* [24] proposed a reliable network for unreliable silicon that uses BIST to diagnose faults and port reconfiguration for the mitigation of faults with turn restrictions to prevent dead-locks. Aisopos *et al.* [28] proposed agnostic reconfiguration in a disconnected network environment (ARIADNE), a reconfigurable algorithm aimed at supporting aggressive scaling, unbounded faults and without pattern constraints in path creation. ARIADNE offers a desired solution based off of up*/down* routing and can execute on any topology. In [29], reversible quad-functional channel buffers were proposed to dynamically adjust the propagation direction of traffic and was shown to improve throughput 2.3× over previous state-of-the-art fault tolerant architectures.

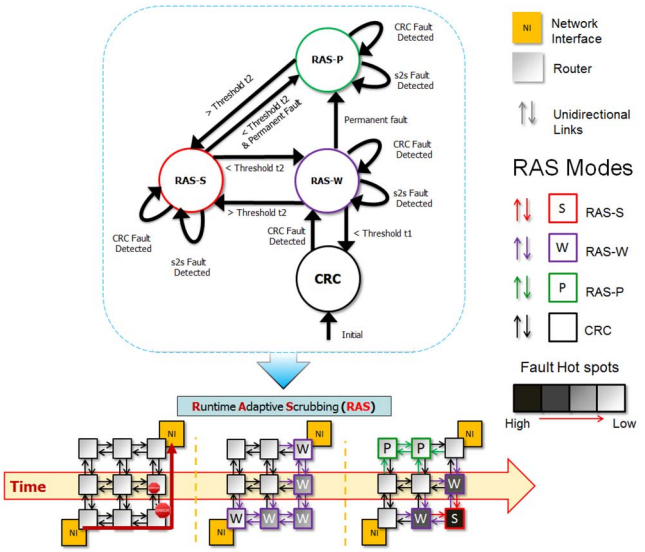


Fig. 1. Proposed state diagram for routers operating in RAS with e2e and configurable s2s encoding.

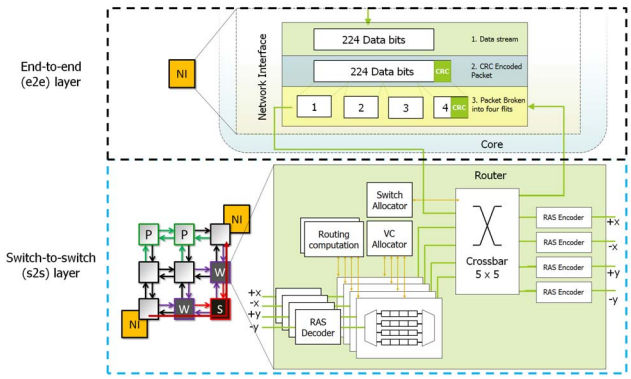


Fig. 2. RAS e2e and s2s micro-architecture.

In our prior work published in [23], we proposed a hybrid e2e and s2s scheme to maximize resiliency and network performance, but also used our proposed adaptive encoder to improve power-efficiency over static schemes by fine-tuning fault tolerance on-demand between routers. As a first-order analysis, we assumed ideal switching and triggering mechanisms between modes to evaluate the best case scenario. By switching to stronger ECCs instead of reconfiguring around faulty links, we found boosting fault coverage was more favorable to improve network degradation and power-efficiency. Since challenges still exist in monitoring the health of fault prone links, we propose a heuristic dynamic transition model in an effort to efficiently utilize our RAS variable strength encoders. Improper utilization could hinder performance and waste overhead cost.

III. Runtime Adaptive Scrubbing

A. RAS Framework

In this section, we will cover the framework for our proposed hybrid encoding scheme, the RAS micro-architecture, configurable s2s encoder design, and finally our proposed method for dynamically selecting modes as links and routers experience various fault rates.

In order to improve resiliency for NoCs in the presence varying fault rates, architectures need to employ adaptable designs such that mitigation is energy efficient. When faults are infrequent, fault coverage should be reduced to augment power savings and increased on demand to ensure graceful performance degradation. To provide graceful network degradation using our proposed RAS technique (extended from [23]), we model our router transition stages as shown in Fig. 1. Initially all routers start in CRC mode using the CRC-32 IEEE polynomial standard without s2s encoding of packets [30]. CRC-mode is implemented to ensure that regardless of the ECC codes employed on the s2s layer, maximum fault coverage is maintained in e2e layer. When faults are detected in links, counters are used to log the frequency of faults and also pinpoint the location of faults. When the number of faults "*e*" in Fig. 1 surpasses threshold *t*1, the router transitions to RAS-W. Routers in RAS-W employ a weak (W) s2s ECC for all input traffic and this information is conveyed to the upstream router. All output traffic is encoded with the s2s ECC defined by the downstream router. Should the error rate *e* exceed threshold *t*2, downstream ECC state selection process checks the error rate and the fault coverage is upgraded to RAS-S (Strong ECC). For routers in RAS-S, all incoming traffic is split from 64 to 32 data bits per cycle. The remaining free bits are then dedicated to encode flits with an aggressive ECC and improve fault detection. In the event that the error rate should decline and diminish below threshold *t*2, the affected routers transition back to RAS-W. If routers are underutilized (low traffic) but minimal s2s encoding is still required, a router can switch to RAS-P (power-efficient ECC). RAS-P splits transmission similar to RAS-S, but provides less fault coverage and lowers power consumption as it employs the weakest ECC required to operate efficiently. With each RAS stage, routers have the flexibility to adapt fault tolerance on demand. If s2s fault coverage is gradually increased from CRC to RAS-S and RAS-W, fewer faults go undetected, but at the cost of additional power and throughput because CRC cannot correct faults on the e2e layer. Improper or early mode selection can, however, have adverse effects on the network. In the RAS encoder design, each mode affects the network speedup, energy cost and fault localization differently. For example in our implementation of RAS-S, we improve fault coverage at the cost of split transmission [27], where flits are serialized into additional segments of to accommodate for stronger ECC. Additional energy and latency is further required to encode each 32-bit segment of the original 64-bit flit. The cost and benefit of each mode will be further discussed in the encoder and decoder section. After which, the proposed mode selection will then be introduced to try and fully maximize performance tradeoffs of each mode, and sustain performance with proper modes selection.

1) RAS Micro-Architecture: In this section, we provide an overview of the proposed RAS micro-architecture for fault tolerant designs in NoCs.

In Fig. 2, a top level view of the proposed RAS micro-architecture is shown. In RAS, each processing core has a

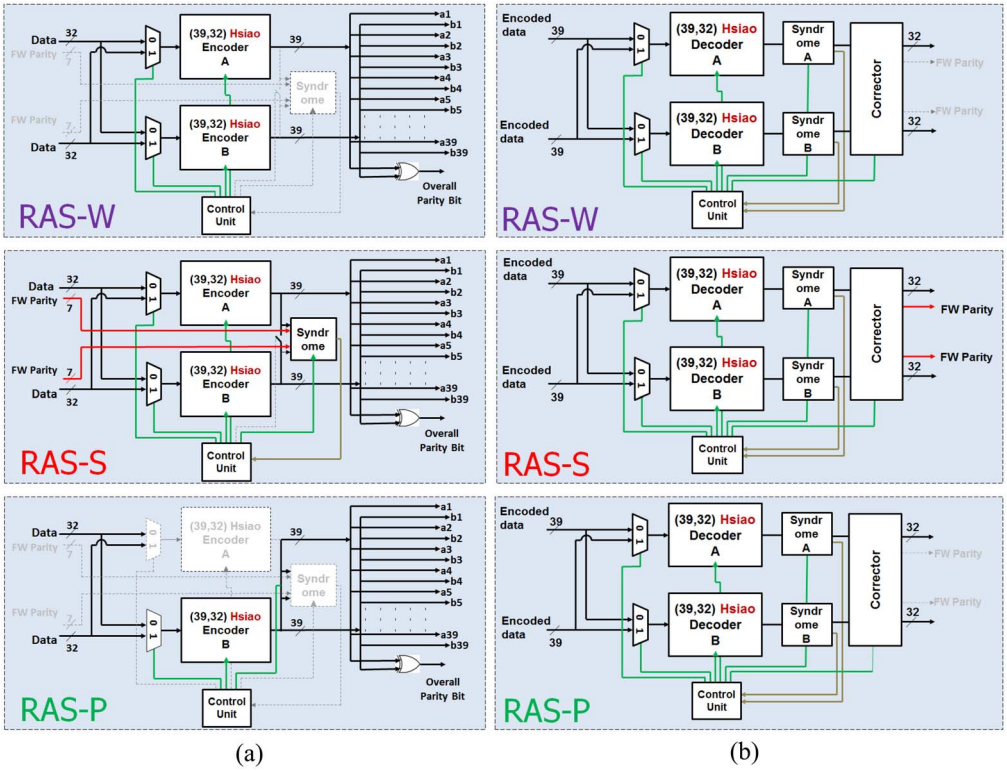


Fig. 3. Proposed RAS three-mode configurable s2s (a) encoder and (b) decoder where all nonactive components and wires are shown in gray for each mode.

network interface connecting the processing element to the network. In a typical mesh network topology, each core is attached to a single router. Each router is then connected in a 2-D grid as shown in Fig. 2(left) with two unidirectional links connecting each router to its adjacent routers. In our proposed design, all data must first be encoded with CRC-32 (256, 224, 32) before it enters the network to form a 256-bit packet. The 256-bit packet consists of 224 data bits which is appended with a 32-bit check code from our CRC e2e encoder.

In RAS, routers consist of a five-stage pipeline with the following stages: 1) buffer write; 2) routing computation; 3) virtual channel allocation; 4) switch allocation; and 5) switch traversal. Each stage of the router is shown in Fig. 2.

While we originally designed RAS with a five-stage pipeline, applying RAS to a shallow (1–3 cycles) pipeline is possible and would require minimal design effort as encoding and decoding executed in parallel to any pipeline stage. With a shorter pipeline, split transmission and s2s retransmissions add stall and delay cycles which will make a larger percentage of the overall packet latency. Therefore, at low network load when speculating will result in the VC and switch being allocated, RAS-S and RAS-P will add latency penalty. However, at medium to high network utilization, queuing latency of the packets at the buffers will dominate and RAS will have marginal impact on the overall performance. However, in CRC and RAS-W modes, there will be no penalty and therefore, some design effort needs to be spent to optimally pick the RAS mode for shorter pipeline.

In RAS, we use triple modular redundancy (TMR) control lines directly connected between routers to transmit router mode information in parallel with flit transmission. We incur a one cycle lag between router mode switching to ensure that flits transmitted in RAS-S and RAS-P finish transmitting the current flit; however, packet latency is unaffected. Once, the mode switch information is received, the router becomes ready to accept data on the new mode in the next cycle.

2) Encoder and Decoder Design: In this section, we will discuss the motivation and design choices for our configurable s2s RAS encoder and decoder. The major challenge in implementing configurable encoders is the complexity required to implement them. The uniqueness and complexity of ECCs typically rules out the possibility of reusing the hardware. Therefore taking advantage of any modularity is a design goal when building configurable encoders. Performance and power tradeoffs should also be minimized to reduce the impact of switching between modes, but it is important to realize the additional cost may be justified if the cost of boosting coverage by using a stronger ECC is less than the cost of rerouting flits and increasing hop counts to bypass it.

To implement an area and power efficient configurable s2s encoder we examine the previous work in multimode encoders such as the ones proposed in dual-layer [27] and inherently modular encoders such as the ones proposed in joint crosstalk avoidance triple-error correction and quadruple-error detection (JTEC-QED) [31] and Hsiao [32] encoders. Yu and Ampadu [27] proposed a dual-layer method for transient and permanent faults with a two mode ECC encoder for transient faults and spare wires for permanent faults. Their ECC encoder, however, requires separate hardware for each mode and relies on interleaving to increase fault coverage. When operating in one mode, the hardware for the other mode is not being used.

TABLE I
DIFFERENCES BETWEEN EACH RAS MODE

			Coverage			
Mode	Codeword	LT	Crosstalk	Burst	Encoder	Router
CRC	CRC-32	n/a	✓	✓	✓	✓
RAS-W	2x Hsiao	1		✓		
RAS-S	JTEC-QED	2	✓		✓	✓
RAS-P	JTEC-QED	2	✓			

In Fig. 3, the block diagrams on the left shows implementation of our three-mode configurable s2s encoder broken down for each mode. Active components for each mode are shown in solid color, while turned off components and control signals are in dashed gray. The foundation of our configurable builds from JTEC-QED encoder proposed in [31] which is implemented using a single Hsiao code encoder, where the output is duplicated and appended with an additional overall parity bit to avoid crosstalk by placing similar signals in pairs. In our design, instead of duplicating the output from a single Hsiao encoder, we include two Hsiao $H(39,32)$ SEC double-error detection (SECDED) encoders and a control block to configure the output and operate the encoders in three RAS modes.

To compliment the descriptions below, we have summarized the differences of each mode in Table I.

a) RAS-W (weak): In RAS-W, shown in the top left inlet of Fig. 3, the 64-bit flits are broken into two 32-bit segments and encoded separately through each $H(39,32)$ Hsiao encoder. The switch that ties the output from the two encoders is turned off and the output from both encoded segments is interleaved to increase burst error protection. If we want to boost coverage and upgrade routers to a stronger ECC, we upgrade to RAS-S.

b) RAS-S (strong): In RAS-S, shown in the middle left inlet of Fig. 3, the data is serialized in what we call *split-transmission* and only 32 bits of the 64-bit flit is encoded per cycle (LT = two cycles) as the Hsiao encoders are run in dual modular redundancy (DMR) to result in a JTEC-QED [31] codeword. The syndrome generator then checks the output from each Hsiao encoder to scan for faults incurred during the encoding process. Additionally, the syndrome generator is also used to compare forwarded (old) parity bits to newly computed parity bits. Comparing the forwarded parity bits (+FW) to the new parity bits allows us to detect faults that occur from the time a flit entered a router (old bits) to the time it exited the router (new). We denote this mode as JTEC+FW throughout the text and use JTEC, JTEC-QED interchangeably.

c) RAS-P (power): Lastly, for RAS-P, split transmission occurs just like RAS-S, but instead of both encoders executing in DMR, only one of the encoders is enabled. All remaining hardware in this mode is turned off to save power, except for the switch that allows the output from the active encoder to be duplicated. Duplication results in a JTEC-QED encoded codeword.

d) Retransmission: In all modes, if faults are detected but cannot be corrected, the flit is dropped and s2s retransmission is immediately requested. For s2s retransmission, routers temporarily store flits in retransmission buffers of $2N_l + 1$ flits, where N_l is the number of cycles needed to cross a link. However, if a fault occurs within a router and is detected by the syndrome generator in RAS-S between the old and new parity bits, the fault cannot be corrected because the correcting hardware does not exist within the encoder. In this scenario e2e retransmission is requested from the appropriate network interface. s2s retransmission can still be used for all other fault scenarios in RAS-S.

e) Decoder: TMR control lines are used to signal which ECC is being used for each port. The decoders in the receiving router tell the sending routers how to encode on their output port. Fault detection in RAS takes place in the following locations.

1) Input port s2s decoders.
2) Output port encoders when in RAS-S.
3) Network interface e2e decoders.

In order to decode each type of ECC, we had to design a configurable decoder shown in Fig. 3(b). In all three modes both decoders are required to decode flits in JTEC-QED from RAS-S and RAS-P and $2 \times H(39, 32)$ from RAS-W. While RAS-P and RAS-S use similar hardware, the syndrome corrector for decoding a flit in JTEC-QED uses a different algorithm that relies on the syndrome from both decoded segments. When in RAS-W, the syndrome and corrector is notified to correct the output from both Hsiao decoders separately. Lastly, to support the internal fault detection mechanism within routers in RAS-S, the existing parity bits are forwarded with the rest of the flit so they can be compared at the encoder with the new parity bits before transmission.

B. Dynamic RAS Architecture

In this section, we propose a method for dynamically selecting the mode of each router in an attempt to localize faults as they manifest and adjust fault coverage on demand using information tracked from our hybrid encoding scheme and s2s encoders. The approach we chose to select thresholds is a heuristic process that can be easily adjusted to accommodate error rates in future generations and emerging technologies.

Recall the RAS state diagram shown in Fig. 1 outlining the dynamic theme of our micro-architecture. Initially, all routers begin in the CRC-mode and do not encode flits with s2s ECC. Once a fault is detected within a packet on the e2e layer, a request for the retransmission is sent back to the source router. As the request travels back to the source, each router the original packet traversed is immediately upgraded from CRC-mode to RAS-W. When faults are detected on the e2e layer the origin of the fault in NoC is unknown. However, the fault can be localized to the path the original packet traversed using a deterministic routing algorithm [28]. We also do not know if the fault was transient or the beginning of a permanent fault. By upgrading routers immediately to boost fault coverage on the s2s layer, we aim to improve the localization of any future faults and prevent costly e2e retransmissions.

To track faults as they manifest each router has a counter used to measure the fault rate. Every fault detected in a router is logged in the counter. To account for faults detected on the e2e layer, we also update the counters for every router in original path of the faulty packet. However, when faults are detected in the e2e layer, CRC encoding only informs us if a

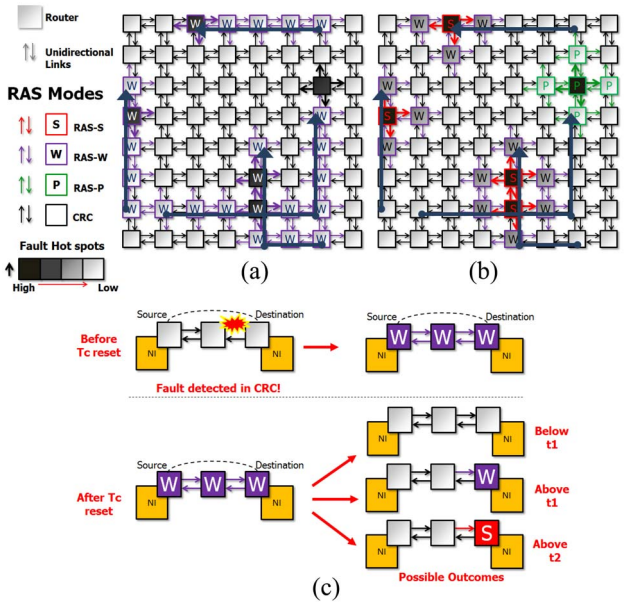


Fig. 4. (a) Routers begin to upgrade around higher error rates and faulty components. Followed by the resulting network in (b) with the ECC boosted on-demand. The decision process is illustrated in (c) where RAS dynamically switching between CRC (white), RAS-W (purple), and RAS-S (red). RAS-P (green) is reserved for permanent faults.

fault exists while the percentage of faults is unknown. Instead of incrementing counters by the number of faults in this scenario, we increment by a base value of three because it is the minimal number of faults required to surpass the detection capability of the s2s layer. After a router upgrades to RAS-W, it will stay in that mode for the time interval *Tc*. *Tc* is the number of decodes chosen before the counter is compared to the thresholds $t1$ and $t2$. If the counter value is less than $t1$, the router downgrades to CRC-mode. If the counter is above $t1$ but below $t2$ it will stay in RAS-W. However, if the counter exceeds $t2$, the router boosts coverage and transitions to RAS-S. In the event of a permanent fault the router transitions to RAS-P to continue using its links instead of rerouting around the fault. An illustrated example of these transitions can be seen in Fig. 4, where Fig. 4(a) and (b) shows the entire network and Fig. 4(c) shows the transition for a specific case.

To quantify the counter thresholds $t1$ and $t2$ we use the following equation from [33] but with parameters values from our architecture:

$$\text{Threshold} = \frac{\lambda_{\text{avg}}\left(1-(1-\epsilon)^{w_f * h_{\text{avg}}}\right)}{T_c} \tag{1}$$

where λ_{avg} is the average injection rate, ϵ is the BER, w_f is the number of bits in a flit, and h_{avg} is the average hop count. The numerator in (1) calculates the number of errors we expect to see based off network parameters and a BER. The resulting threshold is the size (bits) of the counter needed to measure the BER within the T_c interval. Using this equation network parameters can be used to set threshold values for comparison with the estimated BER tracked in each router. To solve for the threshold value in (1) we chose a T_c of 1024 cycles, set w_f to 64 bits and measured the average hop count from Splash-2/PARSEC benchmarks in a 8×8 mesh network

TABLE II
SIMULATION PARAMETERS [29] FOR OBTAINING SPLASH-2/PARSEC TRACES USING THE SIMICS APPLICATION SUITE

Parameter	Value
L1/L2 coherence	MOESI
L2 cache size/assoc	4MB/16-way
L2 cache line size	64
L2 access latency (cycles)	4
L1 cache/assoc	64KB/4-way
L1 cache line size	64
L1 access latency (cycles)	2
Core Frequency (GHz)	5
Threads (core)	2
Issue policy	In-order
Memory Size (GB)	4
Memory Controllers	16
Memory Latency (cycle)	160
Directory latency (cycle)	80

to be 5.5 for each application. As T_c increases in (1), the accuracy of the measured fault rates increases and so does the reconfiguration window. To measure fault rates that are described in the next section we chose a value of 1024. For our analysis we chose to keep this fixed at 1024 in order to evaluate the effect of other parameters in the threshold equation. T_c could be optimized on a case-by-case basis for chips with different reliability constraints. Due to multiple variables that exist in (1) and the adverse performance and power penalties incurred by boosting coverage too early, we will evaluate our proposed dynamic design in the next section across a series of thresholds in effort to adequately adjust fault coverage on demand and analyze the tradeoffs.

IV. PERFORMANCE EVALUATION

In this section, we evaluate our proposed multilayered encoding scheme, the effectiveness of scrubbing packets and our s2s configurable encoder design in terms of power, area overhead, network speedup, and reliability. In order to compare and assess the effectiveness, we compare RAS with several static approaches and evaluate RAS with different thresholds to dynamically localize faults and boost fault coverage on demand.

A. Static Analysis

For the static mode analysis of RAS, we compare our proposed RAS techniques to baseline networks with equivalent single layer s2s ECC encoding schemes. Each scheme is evaluated on a 8×8 64-core mesh topology with a pair of unidirectional links connecting adjacent routers with a supply voltage of 1.0 V and 2-GHz clock. Each router has four VCs for each input port and four 79 bit (64 data, 15 ECC) wide buffer slots per VC. For e2e communication, all packets are assumed to be 256 bits. Before injection into the NoC, each packet is split into four equal flits consisting of 64 bits. Traffic in each network is routed using fault tolerant up/down [28] routing to determine the location of fault and adapt routes when links fail. A failed link is a link that is unreliable for communication because BER in that link exceeds the fault tolerance of the ECC protecting it. In the event of a high BER or permanent faults (within ECC capability) causing a link failure, NoCs must upgrade to a stronger ECC or reroute packets

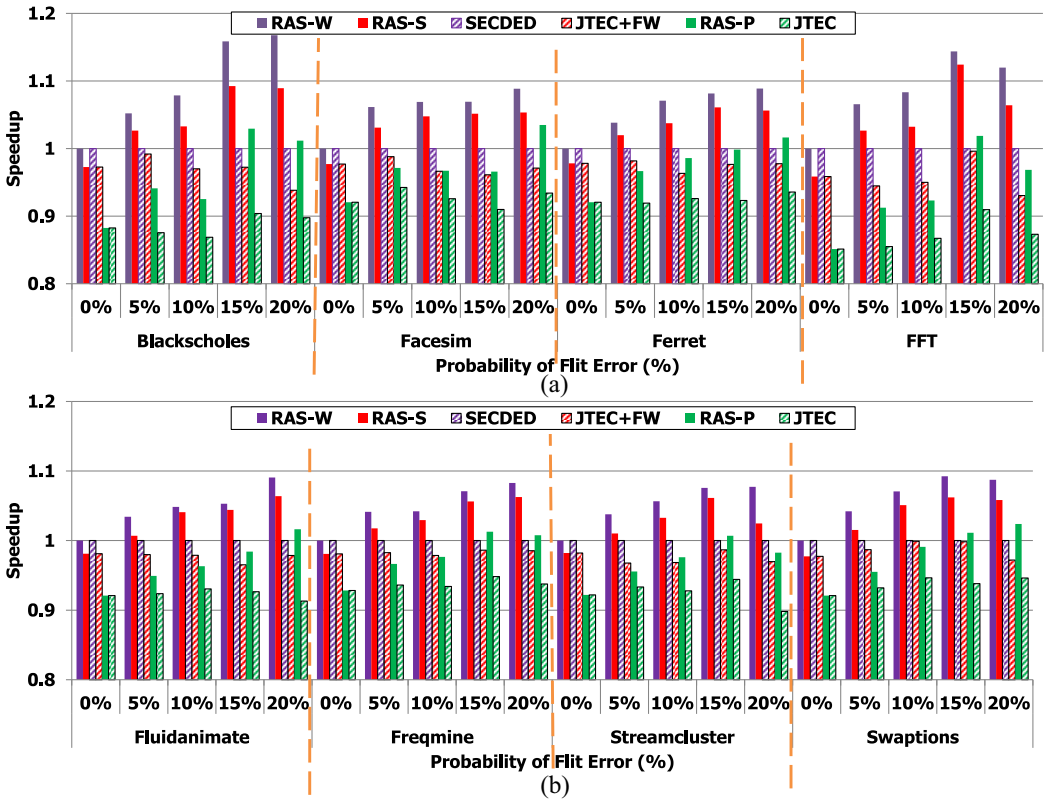


Fig. 5. Network speedup as the flit error injection rate increases when all routers are using the given ECC. Solid color series represent ECC modes with e2e and s2s, while the corresponding patterned series represents the same s2s ECC without CRC (e2e).

around the link. In our evaluation ECCs in the reliability section, we will discuss the required BER to constitute a link failure.

To highlight the effectiveness of hybrid encoding and the benefit of scrubbing flits as they traverse the network, we first compare RAS to static s2s encoding schemes. Second, we will compare our proposed s2s configurable encoder to dual-layers' [27] two mode ECC encoder and ARIADNE [28] as the baseline with one ECC mode. For fair comparison with dual-layer we assume an equivalent split transmission controller proposed in [27] to manage flit serialization in lower bandwidth modes and expanded links to match flit widths. The configurable encoder in dual-layer has two modes but requires separate hardware for each mode. In mode one (ECC1) flits are encoded with a single SEC ECC. In mode two (ECC2) flits are encoded in four interleaved SEC codes. After expanding flits widths for fair comparison, ECC1 became a $H(71, 63)$ SEC hamming encoder, and ECC2 expanded to a four way interleaved $H(19, 14)$ SEC hamming encoder.

For power and area analysis of each compared design, we used the Synopsys Design Compile tool using the TSMC 45-nm technology libraries with a 1.0-V supply voltage and 2-GHz clock. To simulate real network traffic, benchmarks were collected from PARSEC and Splash-2 traces using the SIMICS simulator using the simulation parameters shown in [29, Table II] for the following applications: PARSEC (Blackscholes, Facesim, Ferret, Fluidanimate, Freqmine, Streamcluster, and Swamptions) and Splash-2 (FFT). To model behavior-level faults in the network, we used the fault model proposed in [22] to injects temperature induced timing and process variation faults explicitly for transient faults on the links and datapath within the router pipeline. Dynamic fault rates are modeled through changes in temperature. As we are strictly concerned with protection of datapaths, we do not consider faults in pipeline control logic in our evaluation as it is out of scope for this paper and requires separate techniques. If faults were to occur within the RAS s2s encoder, they would be detected by the inherent DMR of RAS-S and RAS-P, or in CRC. As the fault rates are low in [22] model, we further extrapolated fault rates to observe a worst case scenario. For each compared s2s ECC, if correction fails the flit is corrupt and TMR control lines request retransmission from the retransmission buffers from the upstream router. If a fault is not detected on the s2s layer when it is injected, correcting the fault is no longer possible. If e2e encoding is in place such as CRC-32 in RAS, the fault may still be detected, but will require retransmission. Retransmission buffers of size $2N_l + 1$ flits, where N_l is the number of cycles needed to cross a link (1) were accounted for in power and area analysis for each s2s scheme. When a fault completely evades s2s and e2e encoding, a 200 cycle penalty was assumed as hardware higher up in the network stack can signal for retransmission.

1) Network Speedup: With each technology generation, interconnects are becoming more susceptible to crosstalk, process variation and soft errors. The consequence of providing inadequate fault tolerance will significantly impact overall network performance more as core counts increase,resulting in additional retransmissions and increased chances of silent data corruption (SDC). In s2s schemes, since encoders and

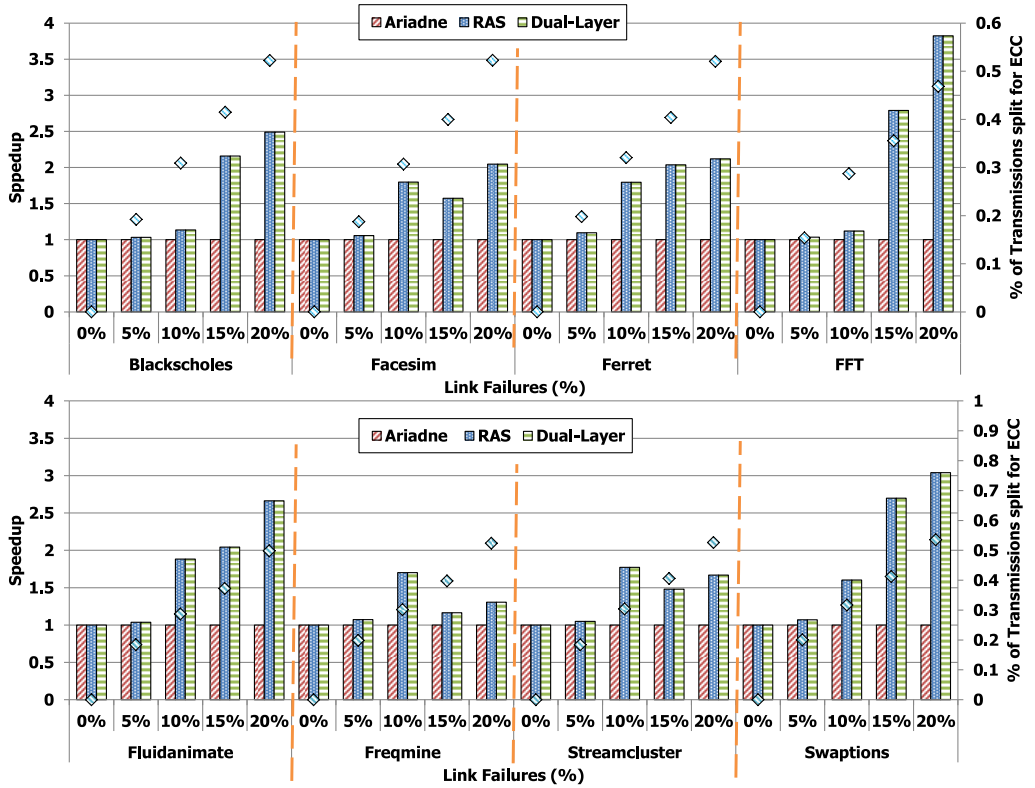


Fig. 6. Network speedup as link failures increase. As links fail, dual-layer and RAS boost fault coverage to continue using the unreliable links while ARIADNE must reroute packets around them. On the secondary vertical axis we show the percentage of link traversals that underwent split retransmission.

decoders are placed in every router, the typical ECC implemented in this type of scheme is relatively weak, in the effort to keep area overhead cost down. The risk as the BER climbs is that purely static s2s schemes will not provide sufficient coverage. To highlight the advantage of a hybrid scheme and the relative network speedup (execution time) gained by scrubbing flits on the s2s layer, we test each s2s encoding mode of RAS with and without e2e encoding as the flit error rate was increased in Fig. 5. In Fig. 5, each RAS mode with e2e encoding is shown in solid colors while each corresponding mode without e2e encoding is shown as the same color with stripes. RAS-S, RAS-W and RAS-P are equivalent to SECDED, JTEC+FW, and JTEC, respectively, without CRC-32 e2e encoding. In every case, each networks with e2e and s2s encoding outperformed its counter part without e2e encoding. As the flit error rate was increased in each mode, with e2e scrubbing the improved speedup over techniques that were implemented without the encoding. The baseline network was chosen as the SECDED network because it is a very common ECC implemented in NoCs. In low flit error rates the cost of splitting retransmission in RAS-S, RAS-P, JTEC, and JTEC+FW can be observed as they all perform below the baseline. The extra cycle penalty on all transmissions, however, is overcome as the flit error rate increases beyond 5% for most cases because each network with split transmission encodes with a stronger ECC. RAS-S which encodes flits in JTEC+FW outperforms RAS-P in every instance because RAS-S has no mechanisms in place to detect intrarouter faults, therefore causing costly e2e retransmissions penalties for every intrarouter fault. On average the significance of combining e2e and s2s encoding to prevent faults from accumulating on the e2e layer improves network speedup by 10%.

Now that we have shown scrubbing flits on the s2s layer can improve network speedup in a hybrid encoding scheme, we also wanted to highlight the advantage of networks employing configurable encoders versus static schemes and show the tradeoffs of rerouting traffic with a fault tolerant routing algorithm versus boosting fault tolerance. In Fig. 6, we compare RAS, dual-layer [27], and ARIADNE [28] as the baseline with a single static s2s-SECDED scheme as randomly selected links fail. For the evaluation setup, we assumed each network initially started with s2s encoding enabled and instantaneous detection of failed links to evaluate the speedup advantage when we can ideally determine when to switch ECCs. In RAS, all routers were in RAS-W, ECC1 for dual-layer, and a standard SECDED ECC for ARIADNE. When a unidirectional link fails and s2s encoding cannot be increased, the adjacent link in the opposite direction is also considered unavailable and both must be rerouted around in Ariande. To reroute traffic around failed links, fault tolerant up/down routing is used in each network to adapt routing tables and prevent deadlocks scenarios in an evolving network topology. When a link failure occurs, instead of rerouting in RAS or dual-layer, both networks boost coverage by upgrading to RAS-S and ECC2, respectively. The results in Fig. 6 indicate that when as little as 10% of the 224 unidirectional links fail in an 8×8 mesh NoC, significant improvements in speedup is shown. This is due to the high latency cost incurred on each packet for each additional hop in Ariande. Split transmission on the other hand only incurs a single cycle penalty per link traversal, and

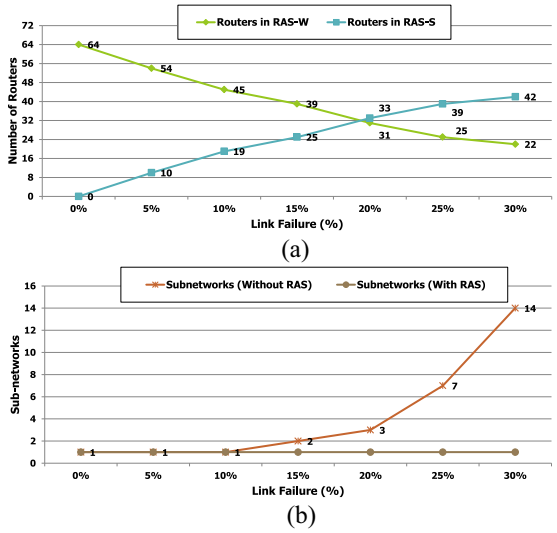


Fig. 7. (a) Number of routers in each mode. (b) Amount of subnetworks formed as link failure percentage increases.

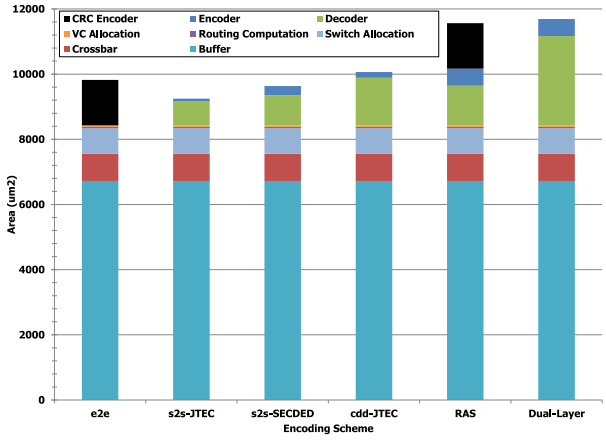


Fig. 8. Single router area overhead for each compared architecture broken down by router stage. Showing CRC encoder cost for comparison.

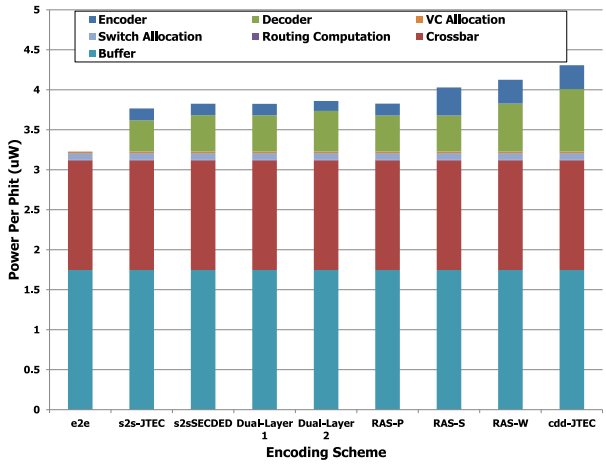


Fig. 9. Router power consumption for each compared architecture.

allows the original topology of the network to be utilized for a small penalty. On the right vertical axis of Fig. 6, we show the percentage of link traversals the split. In this controlled circumstance, RAS and dual-layer provide the same network speedup as they both utilize split transmission, but as we will show later in this section, vary in power consumption, and reliability.

As every unidirectional link failure removes two links from use, a disadvantage to up/down routing in ARIADNE [28], at 10%–15% link failures rates, portions of the network begins to separate themselves. Subnetworks are formed when routers can no longer communicate with every router. As the number of link failures increases, the amount of subnetworks created is shown in Fig. 7(b). It is also important to note how many routers are in each mode. In Fig. 7(a), we show how many routers are in RAS-S and RAS-W. Although we only show modes for the RAS network, the same number of modes in RAS-S and RAS-W would be equivalent to dual-layer's ECC1 and ECC2, respectively. When link failures reach 10% RAS and dual-layer begin to show network speedup improvements over Ariande which must reroute packets. The performance fluctuations between each application can be explained by the differences in traffic patterns. The traffic pattern for each application is different, but the link failures affect the same links in the network for each case.

2) Area: When comparing fault tolerant networks, evaluating the area overhead is an important metric to examine because additional fault coverage typically implies more hardware. By designing area efficient fault tolerant techniques we can minimize the total area footprint of the NoC. One of the primary contributions in this paper is our three-mode configurable s2s encoder. In Fig. 8, we compare total router area cost with the following encoders: RAS, dual-layer, s2s-SECDED, s2s-JTEC, s2s-SECDED, cross-disjoint (cdd) JTEC, and e2e. Even though, we assume the CRC-32 (256, 224, 32) encoder for e2e encoding is implemented within the network interface of each core, for fair comparison we show the cost as if the e2e encoder was in each router. Fig. 8 shows that even though dual-layer has two ECC modes it requires the most router area of the six and 14% more than RAS, followed by cdd-JTEC, s2s-SECDED, s2s-JTEC, and e2e. CRC encoding is conducted in the network interface of each core. If the area cost of the CRC e2e encoder was included with RAS, RAS would still be 1% more area efficient and offer more reliability than dual-layer which will be discussed later in this section.

3) Power: Thus far we have shown RAS improves overall network speedup and reduces area overhead over traditional s2s schemes. In this section, we evaluate how our design stacks up when it comes to overall power consumption. Since additional hardware is generally required to increase fault coverage and improve network resiliency, additional power consumption is not uncommon, but as NoCs consume more overall chip power, researchers should continue to strive for power efficient designs. In Fig. 9, the total power (mW) consumed to send a single flit through each stage of the router pipeline and the next adjacent router is shown for each s2s encoder. For the RAS and dual-layer configurable encoders, the power consumption was broken down further to show the differences of each mode. In Fig. 9, cdd-JTEC consumed the most power even though cdd-JTEC is a single mode encoder; however, in

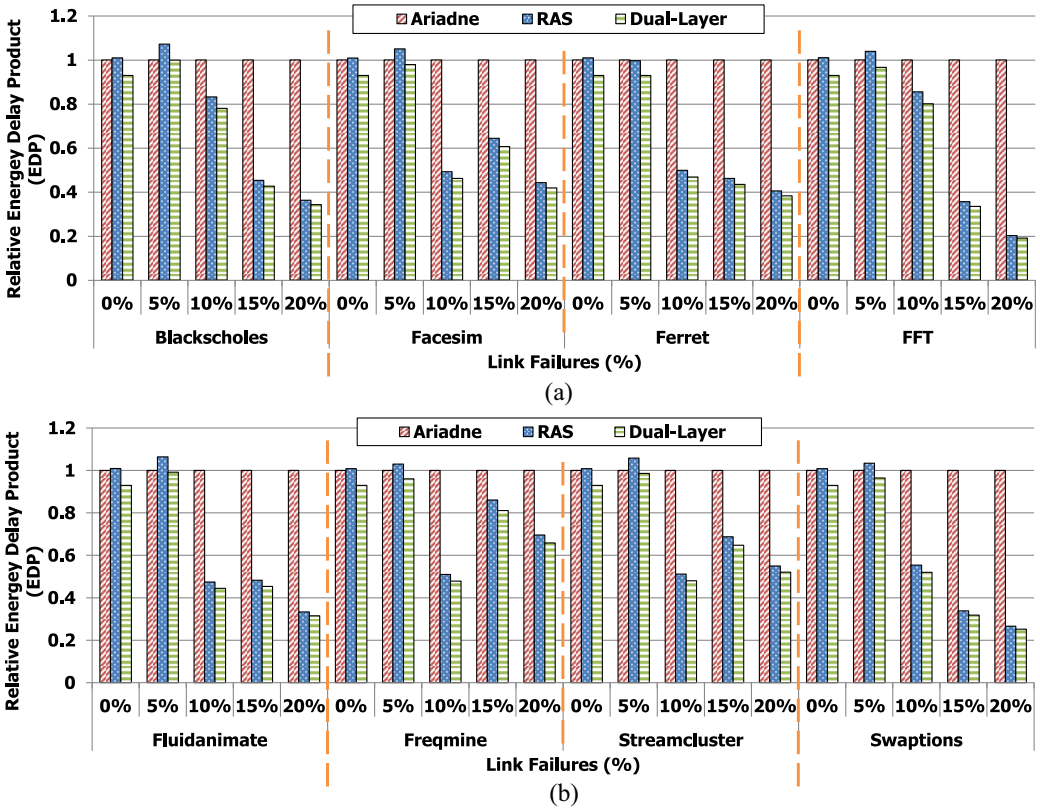


Fig. 10. Network speedup as link failures increase. When link failures occur, DUAL-layer and RAS enable stronger ECC modes to continue using the fault prone links.

cdd schemes, twice the number of encoders and decoders are used in each router. The reasons for additional encoders and decoders in cdd is to provide intrarouter fault coverage. After cdd-JTEC, RAS-W and RAS-S consume the most power. Of the three RAS modes, RAS-W and RAS-S consume the most because nearly all components in each encoder and decoder are active. Compared to single s2s schemes such as s2s-JTEC, each RAS mode consumes more power due to the additional hardware in our design. Despite RAS-P consuming less power than dual-layer's ECC1 and ECC2, on average RAS consumes 6.6% more power than dual-layer, but we save 14% in area overhead and employ stronger ECCs. Now that we are aware of the power penalty induced by each type of s2s encoding, we can evaluate how those penalties translate into overall network power consumption.

To measure energy efficiency of static schemes versus configurable schemes we measured EDP, shown in Fig. 10 as the number of useable links is decreased with the same setup as described in the previous speedup section. As link failures are increased, networks with configurable encoders boost coverage to continue using the faulty links, while networks such as ARIADNE must reroute packets around them. Results displayed in Fig. 10 allow us to measure the tradeoffs of each approach. Every packet that does not take minimal route in ARIADNE, increases the average packet latency and overall power consumption with each additional hop. In RAS and dual-layer, split transmission is traded for additional fault coverage to sustain the usage of those links. When routers are upgraded, every flit incurs an additional cycle delay for each flit, and the power penalty to do so. For low link failures the configurable schemes such as RAS and dual-layer produce similar EDP to ARIADNE. This same pattern was also seen in Fig. 6 when comparing network speedup. In some cases the configurable schemes produce a higher EDP because when routers upgrade to modes with split transmission, all packets traversing the router are affected. In ARIADNE, only the packets forced to reroute cause additional latency and power consumption. However, similar to Fig. 6, as the percentage of link failures increases beyond 10%, the advantage of boosting coverage exceeds the cost of rerouting. This can be explained by the formations of subnetworks and increased dissection of the network where there is as little as 10% link failures. As NoCs continue to scale to support increasing core counts, the more beneficial fault tolerant NoCs will be with adaptive schemes over static schemes. Although RAS on average produces a 6.6% higher EDP than dual-layer, our s2s configurable encoder requires less area overhead and offers higher network resiliency.

4) Reliability: In Fig. 11, we provide area and power overhead cost per fault coverage for each ECC mode. In the table above, the error correction and detection capabilities for each mode is detailed. Since some ECC modes split and interleave segments of the encoded flit, we show the number of bits that can be corrected and detected by segment(s) and the entire flit. In the last column, we calculate a coverage metric that takes each variable into account, and use that coverage value to calculate the area overhead and power consumption per mode (lower is better). While the graph in Fig. 11, does highlight some of the cost/advantage for each mode, it does not take into effect DMR, crosstalk avoidance, intrarouter coverage,

Encoding Mode	Segments Per Flit (A)	Error Correction: Segment (B)	Error Correction: Flit (C)	Error Detection: Segment (D)	Error Detection: Flit (E)	Coverage ((A*B+C)+(A*D+E))
e2e*	1	0	0	3	3	6
s2s-JTEC	2	3	3	3	3	18
s2s-SECDED	1	1	1	2	2	6
Dual-Layer 1	2	1	2	1	2	8
Dual-Layer 2	4	1	4	1	4	16
RAS-P	2	3	3	3	4	19
RAS-S	2	3	3	3	4	19
RAS-W	2	1	2	2	4	12
cdd-JTEC	2	3	3	3	3	18

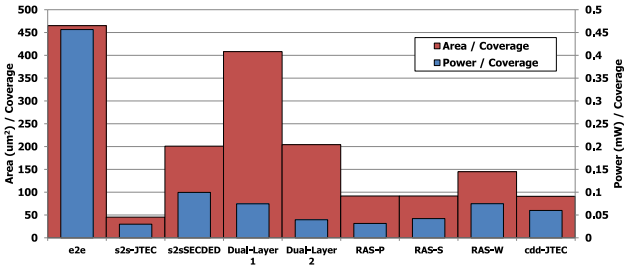


Fig. 11. Reliability comparison of each ECC. In the above table, the error correction and detection capabilities of each code. In the graph, the area and power consumption per those capabilities.

burst error, and odd-bit detection (CRC) that RAS provides outside of s2s ECC.

To further compare the reliability of RAS to dual-layer and commonly used ECCs, we calculate the residual flit error rate (RFER) of each ECC implemented in the compared networks. The RFER as shown in Fig. 12(a) is the probability a flit will contain an undetected fault and nullify the additional fault tolerance supplied by the chosen ECC. Measuring the RFER of each ECC allows us to compare the strength of the code used and delineate the overall reliability of each fault tolerant architecture. As the bit error rate (BER) increases, the RFER of every ECC eventually saturates, except for CRC-32. CRC-32 does not saturate like the rest because beyond the small bit and burst errors detectable in CRC-32, it can also detect all odd bit errors, which places an upper bound on the RFER from exceeding 50%. In Fig. 12, ECC1 and ECC2 of the dual-layer architecture are the first two modes to completely saturate after a flit with no ECC. After which, RAS-W and RAS-S follows, respectively. The RFER of ECC1 and ECC2 increases before each mode of RAS because dual-layer employs the weakest ECCs with only SEC and four-way interleaved SEC. RAS-W on the other hand encodes flits with two-way interleaved SECDED. The RFER of RAS-S saturates last among each s2s flit encoding ECC because it employs the strongest ECC with JTEC-QED. As mentioned previously in the microarchitecture section, the RFER offers insight on when to dynamically switch modes from one ECC to another. It would be desirable for configurable architectures to switch modes as the BER increases to reduce the increase of the RFER in the network. In summary, each s2s ECC employed in RAS outperforms dual-layer. With the CRC-32 combined on the e2e layer, RAS provides significantly higher fault tolerance.

B. Dynamic Analysis

In this section, we evaluate our proposed model for dynamically adjusting ECC coverage in RAS on demand using our s2s configurable encoder. In our prior work [23], only

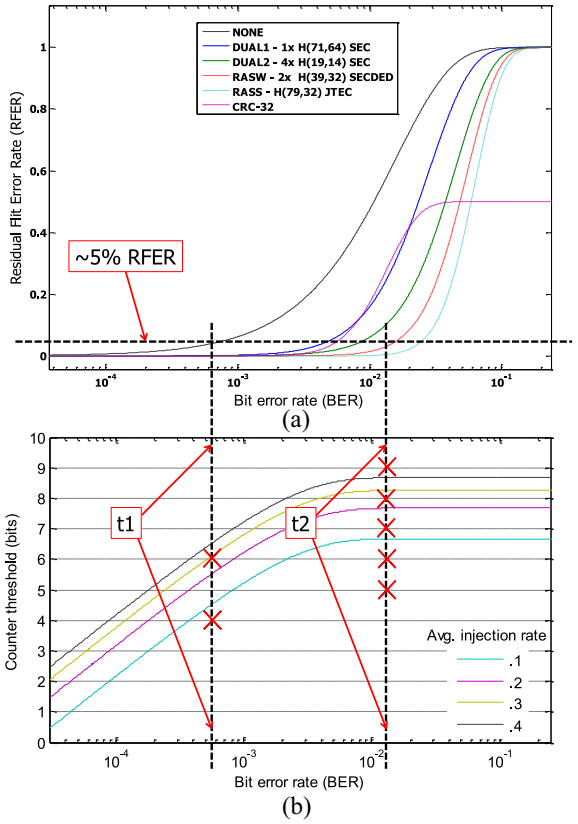


Fig. 12. (a) RFER of the compared ECC networks as the BER is increased. (b) Counter threshold needed to measure the BER for each injection rate. In (b), we show the selection $t1$ and $t2$ for each test case in Table III. The evaluated BER used fall between and after each vertical line.

TABLE III
TEST CASES FOR EVALUATING OUR PROPOSED DYNAMIC SWITCHING MODEL. THE THRESHOLD VALUE REPRESENTS THE NUMBER OF BITS USED TO MEASURE A DESIRED BER THRESHOLD

Case	Threshold 1 (t1)	Threshold 2 (t2)
1	4	5
2	4	6
3	4	7
4	4	8
5	6	8
6	4	9

static analysis was evaluated. A major challenge with adaptive ECC schemes is predicting when to upgrade coverage. Boosting fault tolerance too early can consume excess power and cause unneeded performance penalties when fault rates are low. Failing to adequately upgrade coverage when fault rates increase, raises the risk of full system failure and the chance of SDC. Additionally, while one prediction model may work for one adaptive scheme, there is no guarantee the same model will work for another because each scheme has the possibility to exhibit different tradeoffs for each mode. For example, in RAS-W, while additional power will be consumed after switching from CRC-mode, upgrading will have no negative effects on the speedup. Transitioning from RAS-W to RAS-S, incurs both a power and performance penalty because RAS-S splits flit transmissions.

The approach we chose to select thresholds is a heuristic process that can be easily adjusted to accommodate error

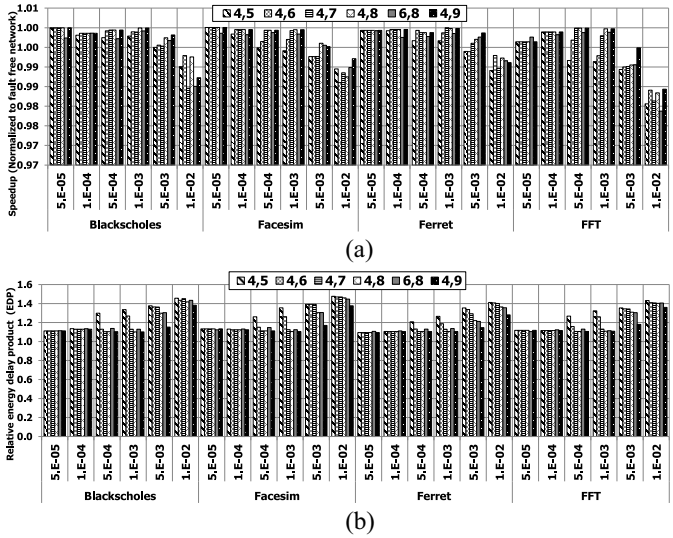


Fig. 13. (a) Speedup for each case as the BER increases. (b) EDP for each case as the BER increases.

rates in future generations and emerging technologies. For the experimental setup of our dynamic model, five routers in an 8×8 mesh were randomly selected to inject faults on every outgoing link (20/224) as the BER was increased. In Fig. 12, we associate both threshold value and RFER with the BER and hypothesized that an RFER of 5% is unacceptable. The 5% error rate is shown as a horizontal line in Fig. 12(a). In RAS there are two transitions of concern, the switch from no s2s ECC (NONE) to RAS-W, and RAS-W to RAS-S. Two vertical lines are shown in Fig. 12(a) that extend to Fig. 12(b), where the s2s ECC in NONE and RAS-W exceeds the 5% RFER. Depending on the technology node and economies of scale, the acceptable RFER and BER may vary but the same approach can be taken. An unacceptable RFER is chosen and then (1) which was described in the previous section, can be used to determine how big counters need to be to measure the undesired fault rates. Every undetected fault that evades detection increases the probability of SDC and full system failure. Therefore the goal should be to boost coverage and switch to stronger ECC modes when the BER increases and causes the RFER in weaker ECCs to climb. To evaluate threshold selection and boost coverage on demand, we then plotted the threshold results of (1) expressed in bits on the y-axis as ϵ (BER) was increased in Fig. 12(b) with different injection rates, λ_{avg}, the only remaining variable of (1) not statically chosen in the previous sections. The plot of (1) shows that as the BER is relatively low, only a few bits are needed to measure a BER of that magnitude. As the BER increases, we need more bits to detect the BER as expected. The challenge is determining what BER is acceptable for each mode. To offer some conjecture on this problem, we can look at the RFER of each ECC and observe when the RFER begins to climb. The RFER as shown in Fig. 12(a) is the probability a flit will contain an undetected fault and nullify the additional fault tolerance supplied in ECC scheme. By correlating the two graphs in Fig. 12 where RAS transitions should take place (vertical lines), we chose a series of test cases shown in Table III. Multiple test cases were chosen to accommodate different injection rates and to evaluate the tradeoffs of threshold accuracy to performance impact. Each test case shown in Table III is marked in Fig. 12(b) with a red *X*. As the transition from RAS-S to RAS-W has a larger performance impact than NONE to RAS-W, we evaluate more values for $t2$ than $t1$. To simulate real network traffic, benchmarks were collected from PARSEC traces using the SIMICS simulator with the simulation parameters previously shown in Table II for the following applications: Blackscholes, Facesim, Ferret, FFT, Fluidanimate, Freqmine, Streamcluster, and Swaptions. When faults are injected, if they are successfully detected on the s2s layer, decoders handle fault mitigation and log each occurrence. If a packet reaches its destination containing faults, e2e retransmission is requested if the fault is detected during the CRC decoding stage. If CRC is unable to detect, the fault we assume a 200 cycle delay before retransmission is requested higher up in the network stack. In the following sections, we will evaluate how thresholds affect network speedup, EDP, mode transitions, decoding, and retransmissions.

1) Network Speedup and EDP: In Fig. 13, we show the network speedup and EDP for each test case as the BER is increased. In order to evaluate an upper bound on the first order analysis of our dynamic model, all test cases for network speedup and EDP are normalized to a fault free network. As the BER is increased, a gradual decline in network speedup is observed for all test cases but does not exceed a decline larger than 2.12%. As the BER does increase test cases with higher $t2$ thresholds perform slightly better than those with lower $t2$ thresholds. For low BER, test case 5 performs significantly worse for some applications because it is the only case where $t1$ is 6 bits instead of 4. With a high $t1$ threshold, routers will transition back to CRC mode more often and fail to detect injected faults. Although each fault missed will more than likely be detected on the e2e layer, the amount of retransmissions will degrade the speedup. This also causes a cascading effect because as routers turn off s2s encoding, fewer faults will be logged in the counters, resulting in inaccurate measurements of the BER. Inaccurately low measurements of the BER will prevent routers from switching to the most efficient mode.

For EDP in Fig. 13(b), a much larger penalty is observed as the BER increases. In scenarios with the highest BER, EDP increases to a maximum of nearly 50% more compared to the baseline of a fault free network but on average 20%. As mentioned above, for test case 5, with the higher $t1$ threshold, EDP increases much faster than cases where $t1$ is lower. As the BER is increased, test cases with lower $t2$ thresholds increase the EDP faster than those with larger $t2$ thresholds. This can be explained by the power consumption tradeoff between RAS-W and RAS-S. When routers are in RAS-S, all incoming traffic is split to augment the stronger ECC used in this mode. Split transmission not only creates an extra cycle delay for each flit, but doubles the link traversal cost. For large traffic traces, the additional cycle delay caused by split transmission can be hidden in overall speedup, but the power penalty as shown in Fig. 13 will be much more pronounced when EDP is measured.

2) Mode Transitions: To try to gain further insight on how each test case affects the architecture in the background, we

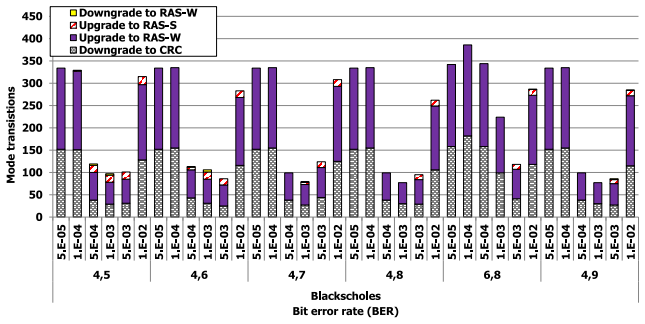


Fig. 14. As the BER was increased we tracked the total number of mode transitions for each case.

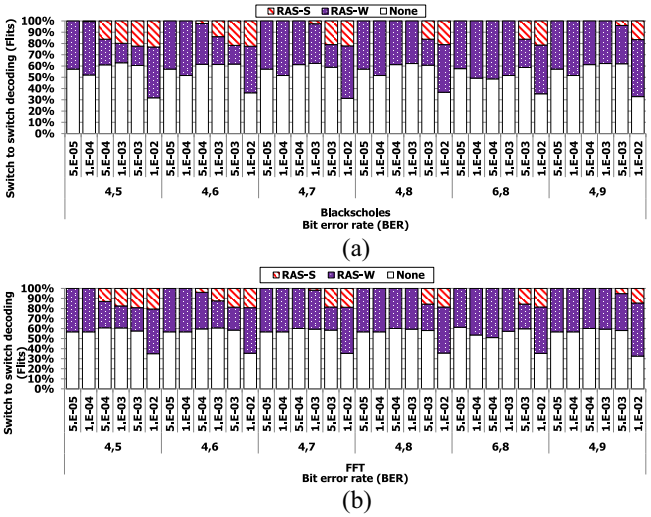


Fig. 15. Percentage of flits decoded in each s2s ECC mode as the BER was increased and RAS dynamically changed modes.

tracked the number of times each router transitioned from one mode to the next. In Fig. 14, we show the total number of transitions that occurred in the network for the Blackscholes application and break it down by mode. In all test cases, except for test case 5, a low BER caused much more mode transitions than when the BER is increased. In test case 5, $t1$ is larger, therefore routers are more likely to transition back to CRC-mode from RAS-W because it is less than likely $t1$ is reached. By observing the results in Fig. 14, and the difference shown between test case 5 from the rest, it may be more beneficial to use a $t1$ lower than 4 bits, which was the lowest $t1$ evaluated. Doing so, could minimize the amount of transitions for lower BERs. Lastly, when the BER reaches 10^{-02}, we see the number of transitions rapidly increase. The increase in transitions in this scenario is because the BER of this magnitude is so high, the ECC in RAS-S is failing to detect faults on the s2s layer. When they are detected on the e2e layer, every router in the path the packet took is then upgraded to RAS-W if it is in CRC-mode, causing a lot of routers to toggle back and forth. If we recall from Fig. 12(a), a BER of 10^{-02}, causes the JTEC-QED ECC employed by RAS-S, to fail at detecting faults, which supports this claim.

3) Decoding and Retransmissions: Lastly, to observe any remaining differences between test cases evaluated in our

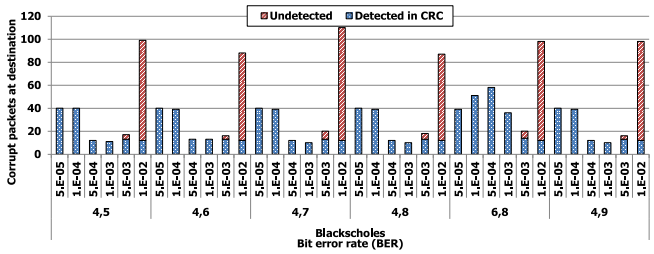


Fig. 16. As the BER was increased the number of retransmissions in each case was logged.

dynamic switching model, Fig. 15 shows the percentage of flits encoded and decoded in each mode. As expected in Fig. 15, the majority of flits were encoded by routers in RAS-W. As the BER was increased, more routers upgraded to RAS-S and the percentage of flits encoded with RAS-S rose. For test cases with larger $t2$ thresholds, the percentage of flits encoded in RAS-S increased at a slower rate.

Fig. 16 shows the total number of packets with errors at the destination (corrupt packets). If routers are upgraded to the appropriate mode, providing adequate fault coverage, we should see the total number corrupt packets decrease. As mentioned above, a $t1$ of 4 bits is potentially too high for low BERs. A smaller $t1$ threshold should capture the BERs and reduce the amount of e2e retransmissions. In the previous section, we claimed that the BER of 10^{-02} was also the cause for the spike observed in mode transitions shown in Fig. 14. The same spike in the number of corrupted packets is seen and further supports that claim.

V. Conclusion

In this paper, we proposed RAS an adaptive hybrid (e2e + s2s) ECC encoding scheme that tunes fault coverage on demand and permits graceful network degradation as fault rates fluctuate in links. Simulation results showed that by scrubbing flits on the s2s layer with our configurable encoder and encoding packets on the e2e layer, network speedup improved on average by 10% over single layer schemes. We then evaluated whether boosting coverage in faulty links with additional hardware would improve network speedup and EDP instead of rerouting packets around them. We found that as 10% of links failed in a 8×8 mesh NoC, adaptive ECC schemes significantly reduced EDP and improved overall network speedup over static schemes. When link failures were below 10%, the cost of boosting coverage and rerouting packets roughly matched in terms of speedup and EDP for most benchmark applications. Since both RAS and dual-layer augmented the cost of boosted fault tolerance by splitting transmission, network speedup for both architectures were equivalent. Although network speedup gains were measured the same, RAS offers three modes RAS-W, RAS-S, and RAS-P compared to ECC1 and ECC2 of dual-layer. When we compared the area overhead of our hybrid scheme to dual-layer and popular s2s encoding methods we showed, RAS was found to cost more than single mode s2s architectures but reduced area overhead by 14% to that of dual-layer. In terms of power consumption, RAS did incur on average a 6.6%

power penalty over dual-layer because most components in our configurable encoder are active in each mode. RAS does, however, implement stronger ECCs than dual-layer. To summarize, while our hybrid encoding scheme does consume 6.6% more overall power than dual-layer, RAS reduces area overhead by 14%, and RAS employs stronger ECCs in each of its three modes, improving the RFER and improving overall resiliency.

REFERENCES

[1] S. Borkar, "Thousand core chips: A technology perspective," in *Proc. 44th Annu. Design Autom. Conf. (DAC)*, San Diego, CA, USA, 2007, pp. 746–749.

[2] L. Benini and G. De Micheli, "Networks on chips: A new SoC paradigm," *Computer*, vol. 35, no. 1, pp. 70–78, 2002.

[3] W. J. Dally and B. Towles, *Principles and Practices of Interconnection Networks*. San Francisco, CA, USA: Morgan Kaufmann, 2003.

[4] D. Sylvester and K. Keutzer, "A global wiring paradigm for deep submicron design," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 19, no. 2, pp. 242–252, Feb. 2000.

[5] R. Ho, K. W. Mai, and M. A. Horowitz, "The future of wires," *Proc. IEEE*, vol. 89, no. 4, pp. 490–504, Apr. 2001.

[6] D. Gillick. (Feb. 2013). *Tilera Announces TILE-Gx72, the Worlds Highest-Performance and Highest-Efficiency Manycore Processor*. [Online]. Available: http://www.tilera.com/about_tilera/press-releases/tilera-announces-tile-gx72-world's-highest-performance-and-highest-effic

[7] Y. Hoskote, S. Vangal, A. Singh, N. Borkar, and S. Borkar, "A 5-ghz mesh interconnect for a teraflops processor," *IEEE Micro*, vol. 27, no. 5, pp. 51–61, Sep./Oct. 2007.

[8] P. Glaskowsky. (2009). *Nvidia's Fermi: The First Complete GPU Computing Architecture*. [Online]. Available: http://www.nvidia.com/content/PDF/fermi_white_papers/P.Glaskowsky_NVIDIA's_Fermi-The_First_Complete_GPU_Architecture.pdf

[9] M. Koibuchi, H. Matsutani, H. Amano, D. F. Hsu, and H. Casanova, "A case for random shortcut topologies for HPC interconnects," *SIGARCH Comput. Architect. News*, vol. 40, no. 3, pp. 177–188, Jun. 2012.

[10] A. Mejia, J. Flich, J. Duato, S.-A. Reinemo, and T. Skeie, "Segment-based routing: An efficient fault-tolerant routing algorithm for meshes and tori," in *Proc. 20th Int. Conf. Parallel Distrib. Process. (IPDPS)*, 2006, p. 105.

[11] M. H. Cho *et al.*, "Oblivious routing in on-chip bandwidth-adaptive networks," in *Proc. 18th Int. Conf. Parallel Architect. Compilation Tech. (PACT)*, Raleigh, NC, USA, 2009, pp. 181–190.

[12] P. Lotfi-Kamran, B. Grot, and B. Falsafi, "Noc-out: Microarchitecting a scale-out processor," in *Proc. 45th Annu. IEEE/ACM Int. Symp. Microarchitect. (MICRO)*, Vancouver, BC, Canada, Dec. 2012, pp. 177–187.

[13] R. Marculescu, U. Y. Ogras, L.-S. Peh, N. E. Jerger, and Y. Hoskote, "Outstanding research problems in NoC design: System, microarchitecture, and circuit perspectives," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 28, no. 1, pp. 3–21, Jan. 2009.

[14] H. M. G. Wassel *et al.*, "SurfNoC: A low latency and provably non-interfering approach to secure networks-on-chip," in *Proc. 40th Annu. Int. Symp. Comput. Architect. (ISCA)*, Tel Aviv, Israel, 2013, pp. 583–594.

[15] A. Prodromou, A. Panteli, C. Nicopoulos, and Y. Sazeides, "NoCAlert: An on-line and real-time fault detection mechanism for network-on-chip architectures," in *Proc. 45th Annu. IEEE/ACM Int. Symp. Microarchitect. (MICRO)*, Vancouver, BC, Canada, 2012, pp. 60–71.

[16] (2011). *International Technology Roadmap for Semiconductors 2011 Edition System Drivers*. [Online]. Available: http://www.itrs.net/Links/2011itrs/2011Chapters/

[17] H. Esmaeilzadeh, E. Blem, R. St. Amant, K. Sankaralingam, and D. Burger, "Dark silicon and the end of multicore scaling," in *Proc. 38th Annu. Int. Symp. Comput. Architect. (ISCA)*, San Jose, CA, USA, 2011, pp. 365–376.

[18] P. Kundu, "On-die interconnects for next generation CMPs," presented at the Intel Labs Single-chip Cloud Comput. Symp., Santa Clara, CA, USA, 2006, p. 12.

[19] H.-K. Peng, C. H.-P. Wen, and J. Bhadra, "On soft error rate analysis of scaled CMOS designs—A statistical perspective," in *Proc. Int. Conf. Comput.-Aided Design (ICCAD)*, San Jose, CA, USA, 2009, pp. 157–163.

[20] A. Dixit and A. Wood, "The impact of new technology on soft error rates," in *Proc. IEEE Int. Rel. Phys. Symp. (IRPS)*, Monterey, CA, USA, 2011, pp. 5B.4.1–5B.4.7.

[21] P. Shivakumar, M. Kistler, S. W. Keckler, D. Burger, and L. Alvisi, "Modeling the effect of technology trends on the soft error rate of combinational logic," in *Proc. Int. Conf. Depend. Syst. Netw. (DSN)*, Washington, DC, USA, 2002, pp. 389–398.

[22] K. Aisopos, C.-H. O. Chen, and L.-S. Peh, "Enabling system-level modeling of variation-induced faults in networks-on-chips," in *Proc. 48th ACM/EDAC/IEEE Design Autom. Conf. (DAC)*, San Diego, CA, USA, 2011, pp. 930–935.

[23] T. Boraten and A. Kodi, "Energy-efficient runtime adaptive scrubbing in fault-tolerant network-on-chips (NOCS) architectures," in *Proc. IEEE 31st Int. Conf. Comput. Design (ICCD)*, Asheville, NC, USA, Oct. 2013, pp. 264–271.

[24] D. Fick *et al.*, "Vicis: A reliable network for unreliable silicon," in *Proc. 46th ACM/IEEE Design Autom. Conf. (DAC)*, San Francisco, CA, USA, 2009, pp. 812–817.

[25] S. Murali *et al.*, "Analysis of error recovery schemes for networks on chips," *IEEE Des. Test Comput.*, vol. 22, no. 5, pp. 434–442, Sep./Oct. 2005.

[26] D. Park, C. Nicopoulos, J. Kim, N. Vijaykrishnan, and C. R. Das, "Exploring fault-tolerant network-on-chip architectures," in *Proc. Int. Conf. Depend. Syst. Netw. (DSN)*, Philadelphia, PA, USA, 2006, pp. 93–104.

[27] Q. Yu and P. Ampadu, "A dual-layer method for transient and permanent error co-management in NoC links," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 58, no. 1, pp. 36–40, Jan. 2011.

[28] K. Aisopos, A. DeOrio, L.-S. Peh, and V. Bertacco, "ARIADNE: Agnostic reconfiguration in a disconnected network environment," in *Proc. Int. Conf. Parallel Architect. Compilation Tech. (PACT)*, Galveston, TX, USA, 2011, pp. 298–309.

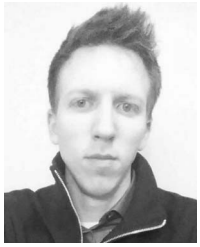
[29] D. DiTomaso, A. Kodi, and A. Louri, "QORE: A fault tolerant network-on-chip architecture with power-efficient quad-function channel (QFC) buffers," in *Proc. IEEE 20th Int. Symp. High Perform. Comput. Architect. (HPCA)*, Orlando, FL, USA, 2014, pp. 320–331.

[30] K. Brayer and J. L. Hammond, Jr., "Evaluation of error detection polynomial performance on the AUTOVON channel," in *Proc. Nat. Telecommun. Conf.*, vol. 1. New Orleans, LA, USA, 1975, pp. 8–21.

[31] A. Ganguly, P. P. Pande, and B. Belzer, "Crosstalk-aware channel coding schemes for energy efficient and reliable NoC interconnects," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 17, no. 11, pp. 1626–1639, Nov. 2009.

[32] M. Y. Hsiao, "A class of optimal minimum odd-weight-column SEC-DED codes," *IBM J. Res. Develop.*, vol. 14, no. 4, pp. 395–401, 1970.

[33] Q. Yu and P. Ampadu, "Dual-layer adaptive error control for network-on-chip links," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 20, no. 7, pp. 1304–1317, Jul. 2012.



Travis Boraten (S'09) received the B.S. and M.S. degrees in electrical engineering and computer science from Ohio University, Athens, OH, USA, in 2011 and 2014, respectively, where he is currently pursuing the Ph.D. degree.

His current research interests include fault tolerance, security, network-on-chips, and computer architecture.

Avinash Karanth Kodi (SM'06) received the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Arizona, Tucson, AZ, USA, in 2003 and 2006, respectively.

He is currently an Associate Professor of Electrical Engineering and Computer Science with Ohio University, Athens, OH, USA. His current research interests include computer architecture, optical interconnects, chip multiprocessors, and network-on-chips.

Dr. Kodi was a recipient of the National Science Foundation CAREER Award in 2011.