Power and Area Evaluation of a Fault-Tolerant Network-on-Chip

Thawra Kadeed, Eberle A. Rambo, and Rolf Ernst
Institute of Computer and Network Engineering
TU Braunschweig, Germany
{kadeed|rambo|ernst}@ida.ing.tu-bs.de

***Abstract*—As fault-tolerant Networks-on-Chip (NoCs) become prevalent in reliable systems, their overhead must be accurately evaluated. In this paper, we evaluate the overhead of a soft error resilient real-time NoC router for ASICs in terms of area and power. We employ a power analysis framework and load profiles that provide accurate power figures. Furthermore, we analyze the power behavior in normal operation as well as under errors. Experiments show that the employed error detection and retransmission schemes in our NoC contribute low power overhead when compared to previously proposed scheme.**

***Index Terms*—ASIC, Realistic Power Calculation, Resilient NoC, Power Analysis under Errors.**

I. Introduction

Networks-on-Chip (NoCs) are presented as an interconnect solution in Multiprocessor Systems-on-Chip (MPSoCs), for both Application-Specific Integrated Circuits (ASICs) and Field Programmable Gate Arrays (FPGAs) [1]. They offer better scalability, modularity, high data throughput, enhanced performance, and design productivity [2]–[4].

As the NoC plays a critical role in MPSoCs, it must handle noise and errors that might occur in its infrastructure. The errors and corruption may occur anywhere in the network, making fault-tolerant NoCs an important topic in safety critical and high availability applications. [3], [5] give a good overview of existing work in fault tolerant NoCs, addressing both permanent and transient errors, also called hard and soft errors respectively.

In this work, we focus on soft errors since they are more likely to happen. Almost all research has focused on general purpose systems and upon error occurrence, the error recovery is realized by retransmitting lost or corrupt packets. Fault-tolerance approaches based on such error recovery mechanisms cannot be applied in real-time designs, which require that all errors and their effects must be known and properly handled [6]. However, retransmission protocols do not work if soft errors cause static effects and, therefore, lead to the failure of the network [4].

Hence, in this work, we focus on fault-tolerance mechanisms that can be used in real-time systems and can handle static effects caused by soft errors. Several kinds of resilient NoCs have been developed for FPGA [7], [8]. Those implementations give a first good estimate for the overhead of a fault-tolerance mechanism. However, the different structures of FPGA and ASIC make these estimates inaccurate for ASIC designs. And, as future MPSoCs demand for a high circuit efficiency, more accurate, ASIC based overhead estimations are needed. Therefore, in this paper, we are motivated to provide an accurate evaluation of the real-time resilient router overhead in ASIC. Moreover, power is increasingly becoming a very important metric in hardware design [9]. And, as power consumption plays a significant role in battery lifetime and thermal perspective [10], an accurate power calculation is required.

Therefore the **contributions** of this paper are as follows: we provide an accurate evaluation of a resilient real-time router and compare it against a baseline one (no error handling in the network) in terms of area and power, where a realistic, more accurate, power estimation is provided by employing a power analysis framework. Thereafter, power behavior under errors is also investigated, where the power consumption of error recovery and packet retransmission mechanisms is evaluated. We also compare different fault-tolerance mechanisms to provide design guidelines for power efficient NoCs.

II. State of the Art

The NoC design space is large, as there are many design parameters that a NoC designer can vary for a certain architecture [1]. There exists a variety of mechanisms and NoCs that have been compared and investigated in terms of power, area and their resulting reliability [11].

For a baseline router, without special support for fault tolerance, the authors evaluate in [9], [12], [13] the power consumption of different router architectures. The authors in [12] estimate the power consumption on different architectures of a Switch Fabric. The ORION simulator [9] provides a simulation framework for estimating detailed power characteristics of a NoC. In [13], Banerjee explores the breakdown power consumption for various different baseline router designs.

The power overhead of using reliable routing algorithms and additional control logic to harden a NoC was investigated in [7], [8]. However, they only use random traffic to estimate the power. While this is sufficient for a general idea on the power overhead, random traffic can lead to inaccurate and pessimistic results compared to a real system behavior. Moreover, on-line fault detection and location for NoC interconnects is introduced in [14]. They propose a resilient router, which employs Code-Disjoint Detection (CDD) scheme as an error recovery and they compare it with Switch-to-Switch (S2S) and End-to-End (E2E) schemes, introduced in [15].

978-1-5386-4034-0/17/$31.00 ©2017 IEEE

A resilient real-time NoC architecture has been proposed and evaluated for FPGA in [16]. The architecture is resilient and predictable. That is, soft errors present only transient effects (resilient) and those effects are limited in time and in scope (predictable). We employ the resilient router architecture of [16] to develop our design for ASIC and evaluate area and power.

The rest of this paper is organized as follows: the referenced resilient router architecture is described in Section III. In Section IV, we present the resilient router overhead in ASIC, where we employ a power analysis framework to provide an accurate power evaluation. The experimental results of resilient router overhead and realistic power consumption are reported in Section V, where we also analyze the power consumption under errors and compare our resilient router overhead with related work's. Finally, Section VI concludes the paper.

III. The Resilient NoC Architecture

The starting point of this work is the resilient real-time NoC architecture of [16]. The work was developed based on the results of a Failure Mode and Effect Analysis (FMEA) [4] of a real-time NoC [17]. The NoC is wormhole-switched with per virtual channel Stop-and-Wait flow control and implements deterministic XY source routing, where the route and Virtual Channel (VC) are defined at the source node. Variable sized packets are split into fixed-size flow control units (flits), which are transferred through a number of VCs. The packets are composed of a Head Flit (HF), zero or more Body Flits (BFs), and one Tail Flit (TF). A Single Flit (SF) packet is also supported.

The NoC routers implements SLIP based arbitration, consisting of a two stage round robin. The router architecture is shown in Figure 1. The router is input-buffered and the Input Buffer (IB) contains 5 VCs, where each VC is a FIFO queue. The Switch Arbiter (SA), implements the SLIP arbitration and gives the arbitration grants to the crossbar switch that connects the IBs to the corresponding output ports. The Virtual Channel Access Controller (VCAC) manages the access to VCs.

Resilience is achieved in [16] with three mechanisms: Fault Containment (FC) and Resilient Router (RR) at router level, and Reliable Transport (RT) at Network Interface (NI) level. FC and RR responsible for recovering from error in time, thus, ensuring that resilience and predictability are satisfied. RT responsible for guaranteeing the data integrity and delivery. Figure 1 presents an overview of the resilient router, where changes to the baseline architecture are highlighted. FC is represented by Ingress Filters and CRC Generators (red blocks), which are added to the input ports of the router. The filter is responsible for deciding whether the flit is valid and may be safely propagated or not. If not, the error-affected flit is dropped before altering the router's state. The filter depends on Cyclic Redundancy Check (CRC) codes and Last Output Port (LOP) data to detect the error. LOP checking helps the router to know if the flit comes from the correct output port of the upstream router or not. The CRC generator block generates the new CRC code for HFs and SFs because of the route rotation.

To recover from errors that might occur outside or inside the router, the RR mechanism has been introduced [16]. It hardens the router's components to prevent static effects from transient faults and to allow the router to continue responding independently from the success of the previous arbitration cycle. In the figure, we highlight the components with major changes (VCAC and the Control Buffer), implementing the new resilient VC flow control.

Based on the design of the above-presented resilient real-time NoC for FPGAs, we derive our resilient real-time router design for ASIC.

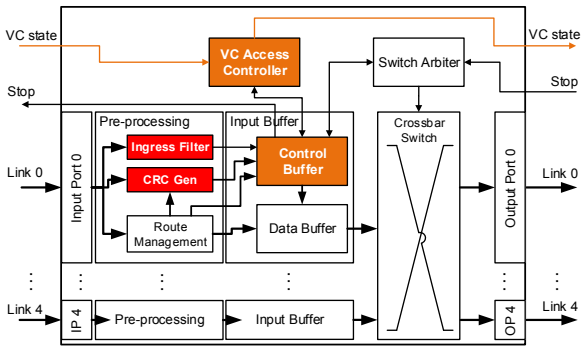

Fig. 1: The resilient router architecture

IV. Resilient Router Overhead in ASIC

In order to develop our resilient router for ASIC, we modified the RTL description as follows: **style modifications**, which are required to make the design synthesizable by ASIC tools; and **design optimizations for ASIC**, where we implemented the Data Buffer as a set of registers instead of RAM. We used only logic cells as opposed to memory cells since the latter do not pay off for implementing small buffers.

After performing the above-mentioned modifications, we synthesized the baseline and resilient fully connected routers in 4×4 mesh NoC by utilizing a standard ASIC design flow. The fully connected router comprises of 5 ports, with 4 ports connecting to neighboring routers and the fifth one connecting to a tile (e.g. processor, memory). A 65nm CMOS process technology (from UMC) with core cell libraries of both high and low threshold voltages in worst case corner (WC, 0.9V, 125°C) was selected in this work. Synthesis for ASIC implementation was carried out by using Synopsys tool chain. All the design units of the router were synthesized using a common compile script. A top-down approach was used for this compilation, while preserving the design hierarchy. After implementing the NoC router design as an ASIC, the overhead metrics regarding area and power of such a router are evaluated. In the sequence, we address the measurement methodology for each of these metrics separately.

A. Area Overhead

The area for the standard-cell implementation is defined as the area of the final core in the netlist. Design Compiler (DC) reports the hierarchical area for each design unit of such a router. In Section V-B, by using the best-effort synthesis method

in DC, we present the area overhead for ASIC routers, and report results of simulations at a clock frequency of 550MHz.

B. Power Analysis Framework

First, we evaluated the power consumed by the resilient router using Synopsys PrimeTime. The reported total power consumed by such a router is 43.6mW. The high figure is due to the clock tree dynamic power. In our design, because of optimizing the router Data Buffer (DB) for ASIC, where the RAM in each DB is implemented as a set of registers, the clock tree has a high impact on power. To minimize that effect, we then enabled Clock Gating (CG) [18] in our design. This means that the synthesis tool automatically inserted low-level Clock Gating cells whenever appropriate enabling signals were detected in the source RTL design.

With CG, the introduced superfluous dynamic power is drastically reduced. This is attributed to the fact that in our router each DB stores 3500 bits of data (i.e. 5 VCs, 5 flits per VC, 4 Physical Units (Phits) per flit). From those, at most 35 bits, i.e. 1 Phit can be modified in a given clock cycle. Thus, we need only 35 register bits to be triggered every clock cycle.

Since the CG design is highly load dependent, we introduce in the following the application-driven workloads generation to inject it at the router inputs and calculate the accurate power.

1) Application-Driven workloads generation: In order to accurately calculate the power consumed by our resilient router, described in Section III, we employ application-driven workloads. Based on that, the switching information is used to gain a reasonably accurate calculation of the power consumption and enable meaningful power optimizations. Therefore, different communication workloads are run and their impact on overall router is investigated.

Benchmarks from the MiBench suite [19] are employed as real-time applications generating traffic in the NoC. In our design, we employed two different simulation scenarios to generate two different workloads, detailed in Table I. Moreover, we consider the technique of replicated execution [20], which provides reliable execution of real-time applications on unreliable hardware. To do this, each application is mapped twice to our platform (cf. Figure 2) in a Dual Modular Redundancy (DMR) configuration [21]. The first scenario is the *Data intensive*, where, on average, each core performs one access (per packet) to the memory every 162 cycles. Since the packet's size is 5 flits and the NoC transmission time per link is 1 flit every 6 clock cycles, then the NoC load with the first scenario is 21.15%. The second scenario is Mixed workload, where, on average, each core performs one access to the memory every 502 cycles. Consequently, the NoC load is 7.14%.

The benchmarks of each workload are run on the Gem5 simulator separately. An ARMv7 operating at 550MHz, with 32kB split L1 cashes and an external DDR3 memory are employed. A trace with memory accesses of each application is captured and fed into OMNeT++, where delay from accesses to shared resources, such as the NoC and the DDR3, are appropriately introduced. The resulting topology and mapping are shown in Figure 2. The traffic at the highlighted router (the closest router to the memory which is fully connected) is then recorded in a trace for each workload scenario, which will be used for the power simulations.

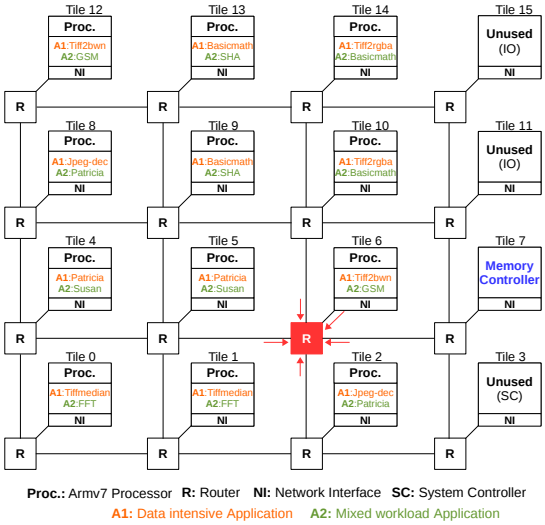

Fig. 2: MiBench applications mapping on 4×4 NoC

After the flit-level traces have been generated for each workload scenario, testbench files were created. The latter read flits traces coming at each input port of the router, subdivide each flit further into four Phits, and inject them into the router.

TABLE I: MiBench applications for *Data intensive* and *Mixed workload* scenarios

Data intensive (run time= 20ms) (avg. 0,00615 access/cycle) 1 access every 162 cycles		**Mixed workload** (run time= 100ms) (avg. 0,00199 access/cycle) 1 access every 502 cycles	
Application	**Program**	**Application**	**Program**
Tiffmedian	[consumer]	FFT	[telecomm]
Jpeg-dec	[consumer]	Patricia	[network]
Patricia	[consumer]	Susan	[automotive]
Tiff2bwn	[network]	GSM	[telecomm]
Basicmath	[consumer]	SHA	[security]
Tiff2rgba	[automotive]	Basicmath	[automotive]

2) Switching Activity and Power Calculation: Upon the generated traffics, Gate-Level simulation of the synthesized router was carried by Mentor Questasim 10.4c simulator to generate Switching Activity Interchange Format (SAIF) files. These files contain the switching activity information, which determines the toggle (switching from 0→1 and from 1→0) rate of all nodes, signals, nets, input and output ports of the circuit. Upon the generated SAIF files, the power was calculated with Synopsys PrimeTime. Here, the accurate averaged power with bit-level accuracy is calculated for each component of the router. This basic framework can easily be generalized from router level to evaluate the power of the complete NoC in different sizes, employing a variety of desirable applications.

V. Experimental Results

In this Section, we evaluate the ASIC resilient router, along with its implementation overhead in terms of area and power

in comparison to the baseline design. For that, we employ the introduced benchmark-based evaluation framework to obtain accurate power figures. Additionally, we investigate the power behavior under errors and evaluate the power overhead of error recovery and of the delivery missing packets.

A. Resilient Router Power Overhead

Let us first present the total quiescent power (i.e. no traffic) consumed by the router. Figure 3 details the leakage and the dynamic power for baseline (3.7mW in total) and resilient (4.4mW in total) routers. As anticipated, the extensions for resilience lead to a higher power consumption. This idle period power is mainly caused by the activity on the clock pins of high-level non-clock gated synchronous elements.

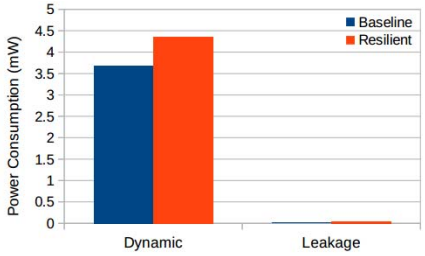

Fig. 3: Routers'quiescent power

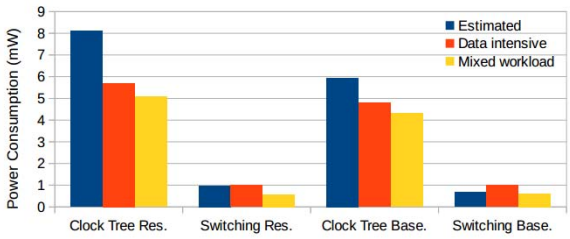

Fig. 4: Estimated and load-dependent power of Resilient (Res.) and Baseline (Base.) Routers

Next, we measured the dynamic power under application-driven trace-based workloads using two different scenarios: *Data intensive* and *Mixed workload* (cf. IV-B1). Figure 4 illustrates the switching activity and the clock tree power per router, where we see now the total power increases with the higher load. Figure 4 also shows that, under *Data intensive* scenario, which provides higher loads, the resilient router consumes 15.63% more power than baseline. The power overhead is quite reasonable under 65nm technology library and higher load scenarios. Moreover, in order to show the inaccuracy between the estimated power (done automatically by the tool) and the load-dependent power, Figure 4 also plots the estimated power per router. One can notice that the inaccuracy (the difference between the estimated and realistic power) increases when the router is more complex (resilient case). On the other hand, we see the estimated switching power is equal or less than the *Data intensive* switching power, where the PrimeTime applies a default toggle rate and static probability on the all nets of the circuit which does not reflect the reality.

B. Resilient Router Area Overhead

Another important aspect of this work is to evaluate the area overhead to achieve resilience. Figure 5 details the area and registers overhead for each component in both baseline and resilient routers. The total resilient router size increases only 14.3%, and the total number of additional registers is 5.14%. These results indicate that the overhead of hardening the NoC using FC and RR mechanisms contribute tolerable cost w.r.t. area. On the other hand, the FPGA implementation of the resilient router in [16] indicates that the overhead to achieve resilience is 93.7% additional LUTs, 70% additional registers (considering DBs as BRAMs), and 37.68% more power under full load with random uniform traffic. From that, we see an inefficient resilience overhead when we optimize the NoC for FPGA, which leads the designers to develop a real NoC component inside FPGAs.

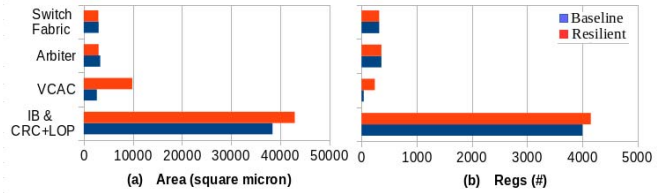

Fig. 5: Baseline and Resilient routers area overhead

C. Power Analysis under Errors

Although resilience is a highly desirable feature in a design, it must not incur vast costs, especially in terms of power. Hence, upon the introduced power analysis framework, we evaluate the power overhead under errors and break it down into error detection, error recovery and packet retransmission.

1) Error-Detection Power Overhead: The error detection overhead corresponds to the additional power consumed by the ingress filter (FC mechanism components: CRC, LOP and dropping-packet logic) and by the resilient components (RR mechanism: hardening VCAC and CB) in error-free case. The error detection overhead is the price paid during the NoC operation, regardless of error occurrences. This overhead is already reported in Section V-A, where it has been analyzed under data intensive and mixed workload scenarios.

2) Error-Recovery Power Overhead: The error-recovery overhead corresponds to the additional power consumed by the router in the error-case to recover from this error and prevent it from propagating. According to the FMEA [4] and upon the functional error model derived in [16], which addressed the impacts of soft error at each component separately, we employ randomly and intended bit-flip error injections inside and outside the router in order to track the power behavior under errors. Next, we evaluate the power consumed by the Fault Containment and Resilient Router mechanisms separately.

a) The FC mechanism: Error-affected routing data could lead to HF, BF or TF corruption. Figure 6 presents the power consumption under different flits corruptions. For reference, the power in error-free case is also plotted.

With Fault Containment, the router under errors consumes less power than in the error-free case. That is due to the

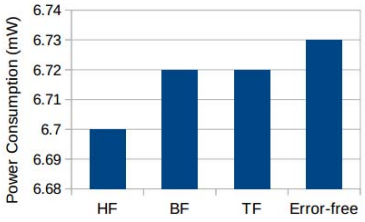

Fig. 6: Power consumption under different corrupt flits

dropping-flit logic, which confines the corrupt flit and drop it preventing the faulty flit to affect the internal state of the router. Figure 6 also shows that the HF corruption contributes the lowest power overhead since the whole packet is dropped. Additionally, the power overhead of handling a detoured flit has the same impact of flit corruption because the derouted flit will also be dropped by the LOP logic.

b) The RR mechanism: Table II illustrates the power overhead of error recovery for each error-affected component separately. An error affecting the VCAC would lead to either improper VC release or reservation, which is in turn generally interpreted as Head-less or Tail-less packet respectively. An error affecting the IB, particularly the CB, which is responsible of memory access management, would lead to either improper read or write, leading to a flit loss; or to an inconsistent buffer state, which results in a reset of CB state. The Switch Fabric recovery mechanism is zero, because its state is reset at each arbitration cycle. The Arbiter recovery mechanism has negligible power overhead and is not shown. Overall, the router power under errors either remains the same or decreases (negative overhead). The former because the error recovery does not trigger any additional logic, as the router automatically recovers from errors at every cycle. The latter because corrupt packets are dropped and are not processed by the router, resulting in less traffic internally.

TABLE II: Recovery power overhead under RR mechanism

Affected Block	Error Impact	Recovery Power Overhead (%)
IB → CB	Improper r/w Inconsistent buffer state	0 0
VCAC	Improper rel. (Head-less) Improper res. (Tail-less)	- 0.45 - 0.15
Arbiter	lost or derouted flit	0 (Automatic Recovery)
Crossbar	corrupt, lost and derouted flit	0 (Automatic Recovery)

3) Packet-Retransmission Power Overhead: The most important factor of power overhead is the power to provide packet delivery guarantee and ensure the correct functionality under errors. To ensure packet delivery, the Reliable Transport (RT) mechanism at transport layer is employed in this work, implementing the Stop-and-Wait (SNW) protocol [22]. Here, the sender NI waits for an acknowledgment (ACK) signal from the receiver NI before sending the next packet or retransmitting after a timeout. Moreover, to calculate the worst case power overhead in case of error, we consider the Negative Acknowledgment (NACK) signal as an additional signal, sending from the receiver NI to request the retransmission of missing packets. We derived our results at router level, where in error-case, the retransmission overhead is restricted to retransmit the NACK signal, which is a single flit packet, and the missing or corrupt packets.

Figure 7 plots the power consumption of the three above-mentioned mechanisms at the router level in both *Data intensive* and *Mixed workload* scenarios, where the effects of different flit-error rates on the power consumption are investigated. We define the flit-error rate as the probability of a flit having a single bit-flip, resulting in higher error rates than expected in practice [23]. For simplicity, we plot the error recovery power consumption for both FC and RR mechanisms in only one line because, in worst case (tail-less packet), both of them contribute the same recovery power consumption (cf. Table II and Figure 6). We also see the entire power consumption (FC+RR+RT) under high error rates contribute acceptable power overhead compared with error-free case.

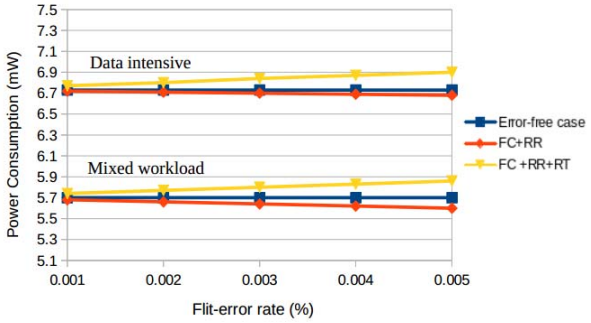

Fig. 7: Power consumption under FC, RR and RT mechanisms

D. Comparison with Related Work

To evaluate our design and to give some design guidelines regarding fault-tolerant NoC w.r.t. power consumption, we compared our design with related work's [14]. They use code-disjoint routers, which implement a so-called CDD scheme to detect and determine the location of data corruption (in the link or inside the router). Hence, data packets are checked at the input and output of the router with parity check codes [14]. Their router also provides retransmission buffers at each input port to retransmit flits in the case of errors. We compare our router power overhead with theirs despite the fact that the approaches are not equivalent. For instance, their approach cannot recover from errors in the control logic (thus resulting in failure) as opposed to our router, which accounts for all possible impacts of errors [4].

They report simulation results for power consumption by flit and analytical power overhead under traffic and errors, where they do not have sufficient analytical information about how the power overhead changes under errors. Therefore, we compared only the detection power overhead. We employed our power analysis framework to calculate the power consumption by a flit of the baseline router and each resilient component. Moreover,

we employed Eq. 1 to obtain the analytical detection power overhead per cycle under the data intensive traffic scenario, where the Router Load RL = 0.22 flit/cycle/router. We used the same RL for both compared routers to get an accurate difference of detection schemes overheads. We analytically describe the power overhead of FC and RR mechanisms as:

$$P_{FC+RR} = RL \times \{P_{FC} + P_{RR}\}$$
$$P_{FC+RR} = RL \times \{P_{CRC+LOP+dp} + P_{h\_cb} + P_{h\_vcac}\} \quad (1)$$

• *RL*: represents the amount of traffic at each router (Router Load), measured in flits/cycle/router;
• $P_{CRC+LOP+dp}$: the power consumption of ingress filter component (CRC and LOP codes and dropping-packet logic);
• P_{h_cb}: the power overhead of hardening control buffer (h_cb);
• P_{h_vcac}: the power overhead of hardening virtual channel access controller (h_vcac).

Table III shows that our error detection scheme contributes significantly lower power overhead than the CDD scheme. That is mainly due to the retransmission buffers needed in CDD scheme to retransmit the flit at router level in case of error. On the other hand, the latency in our design, which employs E2E retransmission, is larger than the CDD scheme's, which employs S2S retransmission. The latency overhead between E2E and S2S is proved in [14] [15]. However, this increase in latency in our design only occurs for the rare E2E retransmission with error while the CDD power overhead affects every individual transmission.

TABLE III: Comparison of power overhead per router

	[14]	Our Router
Tech. lib.	65nm	65nm
Packet size	4 flits	5 flits
Flit size	64 bits	140 bits
Baseline router power (mW)	22	5.14
Error detection power overhead (mW)	2*P_{PAR} + P_{RB} 0.3 + 0.53 0.83	P_{FC} + P_{h_cb} + P_{h_vcac} 0.048 + 0.02 + 0.05 0.12
Power overhead [Eq. (1)] (mW)	0.183	0.03

VI. Conclusion

In this paper, we have evaluated the overhead to achieve soft error resilience in a Network-on-Chip designed for ASICs. For this, we compared baseline and resilient designs in terms of area and power, targeting a 65nm CMOS library and using a benchmark-based evaluation framework. For different resilience mechanisms and benchmark profiles, we investigated the resilience power overhead under regular operation and in the case of errors. Overall, the resilience cost in terms of power is 15.63% and in terms of area is 14.3%. The experiments indicate that the employed FC and RR mechanisms entail far lower overhead than previously proposed link-level mechanisms. For the future, we plan to use the resulting power information for new routing and load distribution methods.

Acknowledgment

The authors are grateful to Sebastian Tobuschat for his valuable comments that contributed to this paper.

This work has been partially funded by the German Research Foundation (DFG) as part of the priority program "Dependable Embedded Systems" (SPP 1500 – spp1500.itec.kit.edu).

References

[1] K. A. Helal and et al., "Comparative review of NoCs in the context of ASICs and FPGAs," in *IEEE International Symposium on Circuits and Systems (ISCAS). IEEE*, 2015.
[2] M. Amoretti, "Modeling and Simulation of Network-on-Chip Systems with DEVS and DEUS," *The Scientific World*, 2014.
[3] M. Radetzki and et al., "Methods for fault tolerance in networks on chip," *ACM Comput. Surv*, vol. 44, pp. 1–35, 2012.
[4] E. A. Rambo and et al, "FMEA-Based analysis of a Network-on-Chip for mixed-critical systems," in *Proceedings of NOCS'14*, 2014.
[5] S. Borkar, "Designing reliable systems from unreliable components:the challenges of transistor variability and degradation," *Micro, IEEE*, vol. 25, no. 6, pp. 10–16, Nov. 2005.
[6] S. Hesham, J. Rettkowski, and et al., "Survey on real-time networks-on-chip," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 5, pp. 1500–1517, May 2017.
[7] A. Patooghy and S. G. Miremadi, "XYX: A Power & Performance Efficient Fault-Tolerant Routing Algorithm for Network on Chip," in *17th Euromicro International Conference on Parallel, Distributed and Network-based Processing. IEEE*, 2009.
[8] C. Killian and et al., "Smart reliable network-on-chip," in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2014, pp. 242–255.
[9] H. Wang and et al., "Orion: a power-performance simulator for interconnection networks," in *Proceedings of the 35th Annual IEEE/ACM International Symposium on Microarchitecture,(MICRO-35)*, 2002.
[10] Y. Wu, S. Thomson, and et al., "Free Razor: A Novel Voltage Scaling Low-Power Technique for Large SoC Designs," *in IEEE Transactions on VLSI Systems*, vol. 23, no. 11, pp. 2431–2437, 2015.
[11] D. Bertozzi, L. Benini, G. Micheli, and D., "Error control schemes for on-chip communication links: the energy-reliability trade-off," *Transactions on CAD, IEEE*, vol. 24, no. 6, 2005.
[12] T. T. Ye, G. Micheli, D., and L. Benini, "Analysis of power consumption on switch fabrics in network routers," in *Proceedings of the 39th annual Design Automation Conference. ACM*, 2002.
[13] A. Banerjee, P. T. Wolkotte, R. D. Mullins, S. W. Moore, and G. J. Smit, "An energy and performance exploration of network-on-chip architectures," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 17, no. 3, pp. 319–329, 2009.
[14] C. Grecu, A. Ivanov, R. Saleh, E. S. Sogomonyan, and P. P. Pande, "On-line fault detection and location for NoC interconnects," in *12th International On-Line Testing Symposium (IOLTS)*, 2006.
[15] S. Murali, T. Theocharides, N. Vijaykrishnan, M. J. Irwin, L. Benini, and D. M. G., "Analysis of error recovery schemes for networks on chips," *IEEE Design & Test of Computers*, vol. 22, no. 5, pp. 434–442, 2005.
[16] E. A. Rambo, C. Seitz, S. Saidi, and R. Ernst, "Designing Networks-on-Chip for High Assurance Real-Time Systems," in *Pacific Rim International Symposium on Dependable Computing (PRDC)*, Christchurch, New Zealand, 2017.
[17] S. Tobuschat, P. Axer, R. Ernst, and J. Diemer, "IDAMC: A NoC for Mixed Criticality Systems," in *Embedded and Real-Time Computing Systems and Applications (RTCSA)*, 2013, pp. 149–156.
[18] Mullins and Robert, "Minimising dynamic power consumption in on-chip networks," in *System-on-Chip, International Symposium on. IEEE*, 2006, pp. 1–4.
[19] M. R. Guthaus and et al., "MiBench: A free, commercially representative embedded benchmark suite," in *IEEE International Workshop on Workload Characterization, WWC-4*, 2001.
[20] E. A. Rambo and R. Ernst, "Replica-Aware Co-Scheduling for Mixed-Criticality," in *Euromicro Conference on Real-Time Systems (ECRTS)*, Dubrovnik, Croatia, June 2017.
[21] P. Axer, M. Sebastian, and R. Ernst, "Reliability analysis for MPSoCs with mixed-critical, hard real-time constraints," in *Proceedings of the international conference on Hardware/software codesign and system synthesis (CODES+ISSS)*, 2011, pp. 149–158.
[22] E. A. Rambo, S. Saidi, and R. Ernst, "Providing Formal Latency Guarantees for ARQ-based Protocols in Networks-on-Chip," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2016.
[23] J. Autran and et al., "Altitude and underground real-time SER characterization of CMOS 65 nm SRAM," *IEEE Transactions on Nuclear Science*, vol. 56, no. 4, pp. 2258–2266, 2009.