Technical Journal, University of Engineering and Technology (UET) Taxila, Pakistan Vol. 22 No. 3-2017
ISSN:1813-1786 (Print) 2313-7770 (Online)

Adaptive Error Control Mechanism for Near Threshold Computing based on Network-on-Chip

F. Habib[1], N. K. Baloch[2], A. Hussain[3], H. Jamal[4]

[1,2,3]Department of Computer Engineering, University of Engineering and Technology Taxila, Pakistan
[4]Faculty of Engineering Sciences, GIK Institute, Topi
[2]naveed.khan@uettaxila.edu.pk

Abstract-In this paper, we present a switching model to increase reliability of Network on Chip (NoC)which is compromised due to Near-threshold computing (NTC) faults. The Proposed method provides three modes of switching to tolerate some diverse faults occurring in the network. In low noise conditions, our model operates on End to End mode to achieve better reliability and low latency. In more significant noise conditions it is shifted towards slope and Hoe to Hop mode to tolerate accmulated faults in the network. The proposed model achieves a better trade-off conditions in term of reliability and latency as compared to BCH and CADEC codes and attain energy efficiency with the help of NTC model and provide reliability by switching between different modes to realizes a better fault correction capability.

Keywords-NoC, NTC, Fault Tolerance

I. INTRODUCTION

The decreasing size of a transistor has [i-ii] enables the designers to integrate billions of transistors on a single chip. This abundant availability of the transistors has led to the concepts of chip multiple processors(CMP). For handling the communication needs between these CMPs has led towards the concepts of the network on chip(NoC) [iii-iv].The communication in the network on chip required minimum latency and high reliability [v]. This decreasing size of the transistor has also give rise to increased power consumptions for the NoC. It revels in research conducted in the US that in 2006, datacenters consumed about 1.5 percent of the total electricity [vi]. This dangerous condition has focused towards optimizing the power consumptions for future computer systems based on NoC. Different alternatives are available to reduce power consumption caused by cores. One possible solution is dark silicon where transistors are underutilized due to low power budget [vii]. One of the best alternatives to the dark silicon is Near-Threshold Computing (NTC) [viii] which operates the transistor at a low voltage level and lead to in a better tradeoff conditions between power and latency in comparison to subthreshold circuits [ix]. NTC has now been in used in NoC to overcome the energy and power constraints of the NoC. However, due to process variations, this power efficiency costs performance loss [x]. The benefit of NTC is that it consequences in increased power efficiency for traditional architectures which were designed for operating on specific voltage [xi].

Due to increase of transistors on a single chip according to Moor's law, NoC architecture tends to achieve better performance and reliability. According to the concept of the NTC, as the operating voltage of the transistor is reduced it results in better energy efficiency [xii] at the cost of performance loss. This performance loss occurred due to increase in the crosstalk, Single-event upsets(SEU) and aging problems which may lead to transient, intermittent or permanent faults [xiii-xiv] which may fail the system [xi].Thus, there is a high need to tolerate the faults occurring due to NTC. The Error Control Coding (ECC) protection can be used to provide desired reliability for NoC.

The previous research has adopted ECC at the datalink layer [xv-xvii]. Hop-to-Hop (H2H) ECC can tackle higher noise but resulted in higher energy consumption if number of faults in the network are lower. Another alternative to the H2H coding scheme is to adopt network-layer ECC protection scheme named the End-to-end (H2H) which correct the error in a packet only when it reaches the destination router [xvii-xx].

These techniques have pros and cons. We propose a framework that can combine the merits ofH2H, E2E and Slope technique to improves the reliability and energy efficiency of the system while maintaining performance.

The proposed framework is designed to address both Logic Voltage Induced (LVI), and Timing Voltage Induced (TVI) based Single Event Upset (SEU) faults. The rest of the paper includes literature review, proposed three-layer model description and performance analysis of the proposed method with state-of-art techniques available.

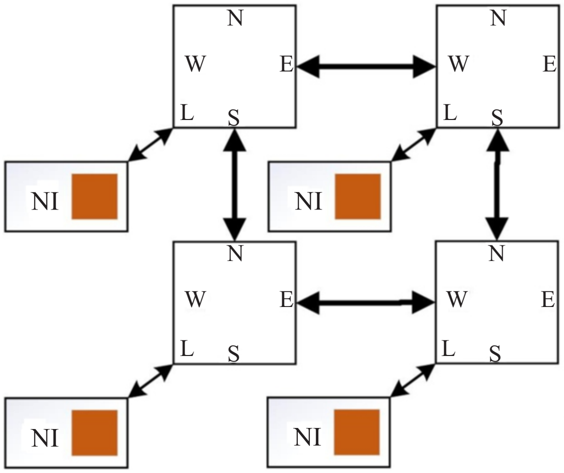
II. LITERATURE REVIEW

E2E protection schemes are operated on the Network Interface. The encoding and decoding process is only performed when it reached the destination router as shown in the Fig.1. H2H protections schemes are performed at the datalink layer and operated on each hop, as shown in Fig.2. Every router input port has its own encoder and decoder. For low noise regions Hybrid Automatic Repeat request [xxi] recovers the faulty flits without incurring latency overhead. In case of low noise, the additional encoding and decoding waste energy. Rossi et al. [xvii] propose a model that use different ECC schemes. To reduce the energy waste, Li et al. [xv] have utilized the error detection capability to the solve the transient faults. Wang, J., et al.[xxii] has conducted a comprehensive study to tolerate the faults caused by NTC and dark silicon. Different coding schemes Hamming, boundary shift code (BSC) and modified dual rail code (MDR) has solved the single error. E2E coding schemes resolve the transient faults only at the NI. The receiver sends an acknowledge signal to the sender if packet is correct. In this work, we take advantage from H2H, E2E and Slope methods to covers the faults caused by the NTC in NoC. E2E is performed only when the number of corrupted packets are less than to a pre-defined threshold value T1. If the number of corrupted bits increased from that threshold value T1, the ECC mode is switched to the second layer which is Slope as shown in Fig.3. In case of more corrupted packets exceeding threshold value T2, the ECC mode is switched to the third layer which is H2H protection. The flit corruption record is saved in history flit which keeps tracks of corrupted flits and helps the control unit to accommodate that switching.

The detailed descriptions of this model are described in the next sections.

III. PROPOSED SWITCHING SCHEME

The Switching protocol work by the amount transient faults occurring in the network. In case of a variable number of transient faults, switching between multiple layers results in improved performance and reliability of the system. We propose an ECC mode switching protocol which shifts between different layers during runtime operation of the system. In case of a small number of transient faults, E2E ECC protection is performed to maintain the integrity of the packet. In case number of faults occurring in the network exceeds a specific threshold value T1, Slope approach is utilized to tackle the faults. Slope utilized optimize locations to place the ECC protection. When the number of faults exceeds the threshold value T2, then H2H ECC protection is used to solve the faults. The state machine diagram is shown in the Fig. 4. There are three state transition along with some intermediate states which are used for synchronization purpose. The intermediate states named as Pre-E2E,Pre-Slope,andPre-H2Hare used for sending the instruction to the network for the mode switching to maintain synchronization. The E2E protection is used in the network if NoC is in the E2E state or Pre-E2E, Slope is used to when the network is in Pre-Slope and Slope state, and H2H protection is used when the network is in the Pre-H2H and theH2H state. When the network starts, it goes in E2E state to reduce the energy consumption and providing better reliability. The network monitors some faults in the network after every T count cycles and compares it with threshold values T1 and T2. If the number of faults exceeds from value T1 then the associated node request to switch the mode to Pre-Slope and if it exceeds from value T2, it sends a request to shift mode to Pre-H2H. This switching information is also delivered to others nodes in the network to inform them there is a mode switch in the network. This information is sent to all nodes of the network to maintain synchronization in the network. The switching request is transmitted to the network in first T propcycles of each T period. During that cycle.



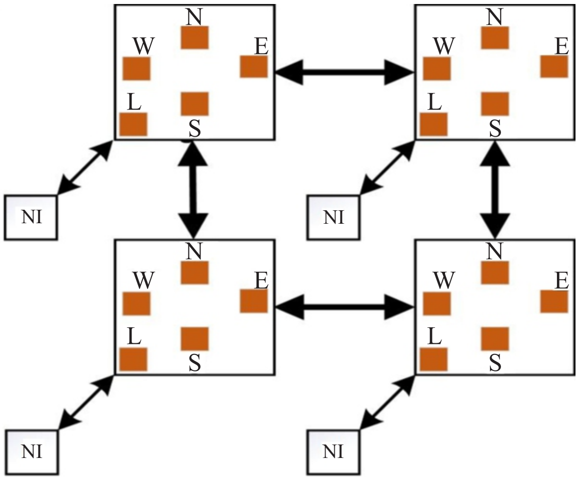


Fig. 1. E2E Protection in Baseline Fig. 2H2H Protection in the Baseline

After that time the nodes switch to operate on E2E, Slope and H2H protection modes. The maximum propagation time for delivering this message to all the node is given by (1).

$$T_{prop} = 3(\sqrt{n} - 1) \quad (1)$$

Where, n is the number of nodes in the network. To prevent mode oscillation in the network, it monitors the requests coming from the network. The only request coming from the local node and neighbor nodes resulted in a state transition. This mode switching is activated only when the number of errors during T count exceeds the specific threshold value.

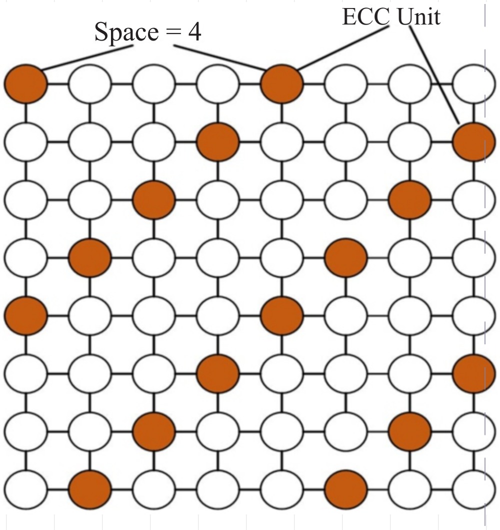


Fig. 3. ECC Protection using Slope

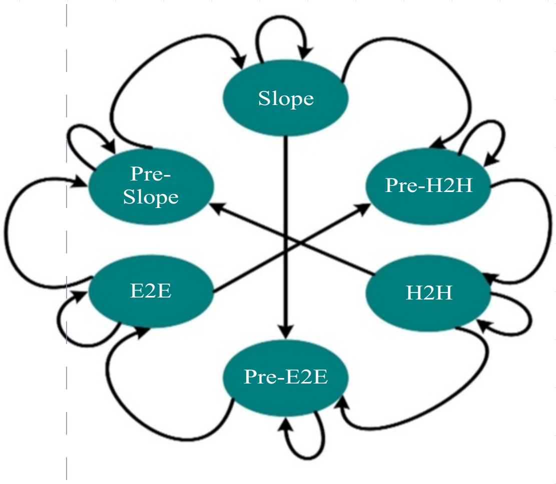


Fig. 4. ECC Mode Switching Protocol

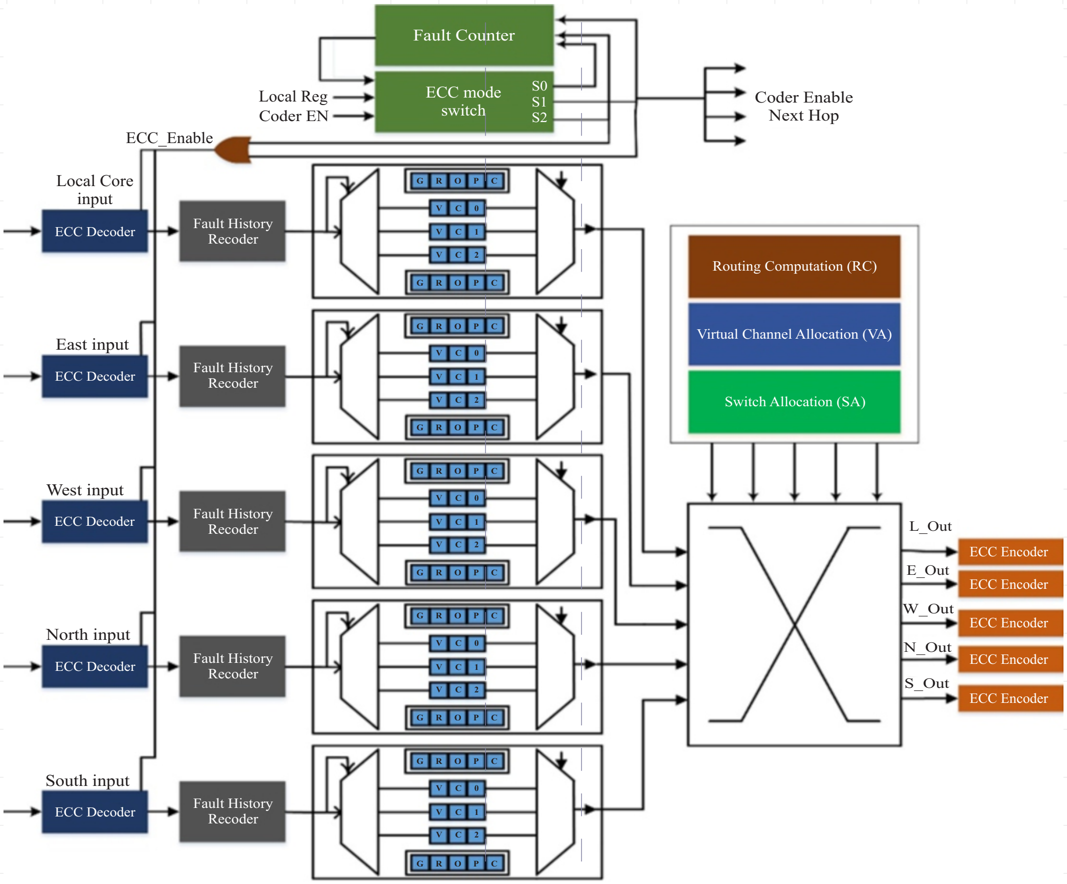


Fig. 5. Router Architecture

For a given traffic load α and error rate p_e after leaving the h hops the packet containing errors are expressed by (2).

$$Total\ corrupted packet = T_{count} \text{ x } \alpha \text{ x} p_e \qquad (2)$$

In which

$$p_e = 1-(1-\varepsilon)^{\omega\rho} \text{ x } h \qquad (3)$$

The term p_e is used to represent the fault model, ωρ is the number of corrupted bits and ε represent total error rate. T_{count} can be obtained by rearranging (2) and by putting (3).

$$T_{count} = \frac{Total\ corrupted\ packets}{\alpha \times (1-(1-\varepsilon)^{\omega\rho} \times h)} \qquad (4)$$

A modular counting timing $T_{count}$is attached to each router. The T_{count} can also be set by (5)

$$T_{count} = \frac{Total\ corrupted\ packets}{\alpha_{avg} \times (1-(1-\varepsilon)^{\omega\rho} \times h_{avg})} \qquad (5)$$

In this equation avg α_{avg} is used to represent average load $h_{avg}$is the average hop counts.

The network interface and router architecture are modifiedto implement the proposed switching methodology in NoC. The network interface is responsible for measuring the number of corrupted flits in the network. If the node is in the E2E state and number of corrupted packets exceeds from threshold value T1 then it will make a transition to Pre-Slope. In case of value exceeding the threshold value T2, it will request a transition to the Pre-H2H state. This is accomplished by keeping track of error history flit maintaining at each router.

The flit starts with a unique Id which determines the router Id. The rest of the flit contains information of error detection or not. If an error is detected at a hop, it is written as "1" otherwise "0" is written in the flit. The counter is incremented if this field contains "1" and counter resets after every T_{count}.

IV. Fault Tolerant Router

The router architecture also needs some modifications to handle this three-layer switching protocol. The proposed router architecture is shown in the Fig.5. A router in NoC consists of five input ports and five output ports named as East, West, North, South and Local. The main addition to the conventional router architecture design is Fault History Recorder, Fault Counter, and ECC mode Switch. It is the responsibility of the Routing Computation Unit (RC) to obtain the destination information. The switch allocator controls the connection between input and an output port with the help of crossbar unit. For facilitating the error control mechanism NACK signal is used. The switching mechanism is represented with the state diagram in the Fig.4. There are six state transitions in the network. For E2E switching in the network the S1=0 to maintain E2E switching overall in the network. For Slope its value is S=1 and for H2H switching its value is S=2.The Fault counter sends the propagation finish signal which indicates mode switching is completed in the network.

If the network is operated on the H2H switching, then the output of the EEC decoder is saved. This flit contains unique bit pattern which is sent to Fault History Recorder, and last bit is used with hop EEC decoder to represent the number of the errors. The other error history bits of the different hops are pushed forward to the next hop. For a 32-bitflit, it can hold records of 24 hops error history.

The switching between different modes depends upon the exchange of the information between different protocol. When the network is operated on E2E switching protocol, the global history is used for switching. When the network is operated on the Slope, and H2H switching protocol, both local and global history are used to determine the total errors in the network and determine if there is a need to mode switch. The total number of errors in the network plays an essential role in switching between these modes. Different switching modes are activated based on the number of faults in the network.

V. Results and Discussions

In this section, we analyzed our proposed scheme by simulating 8x8 mesh using Gem5 [xxiii] and Garnet 2.0[xxiv]. We make use of synthetic and benchmark traffic to evaluate the proposed design. We evaluated our proposed model with Fixed EEC decoder applied in the [xxv] to evaluate the efficiency of the proposed model. We have evaluated different error control mechanism to observe and compare the reliability and performance evaluation of our proposed model on benchmark and synthetic traffic patterns.

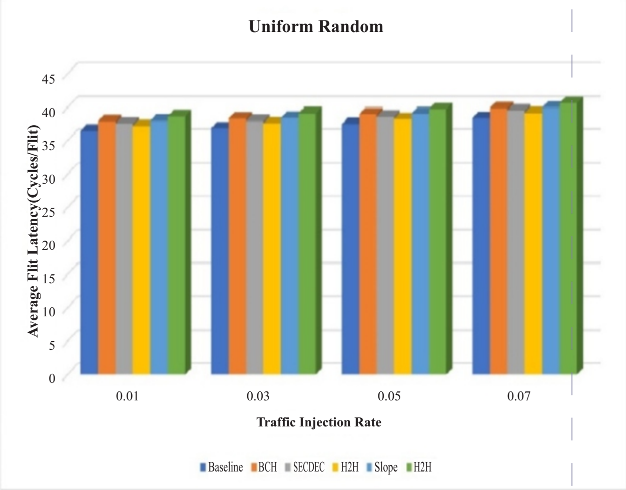


Fig. 6. Comparison on the basis of Uniform Random Traffic

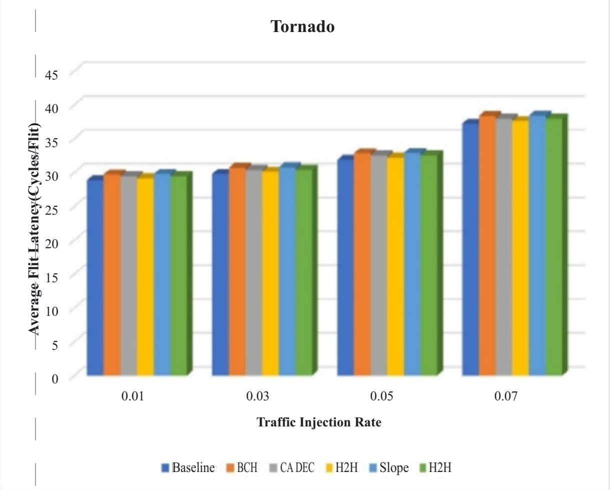


Fig. 7. Comparison on the basis of Tornado Traffic

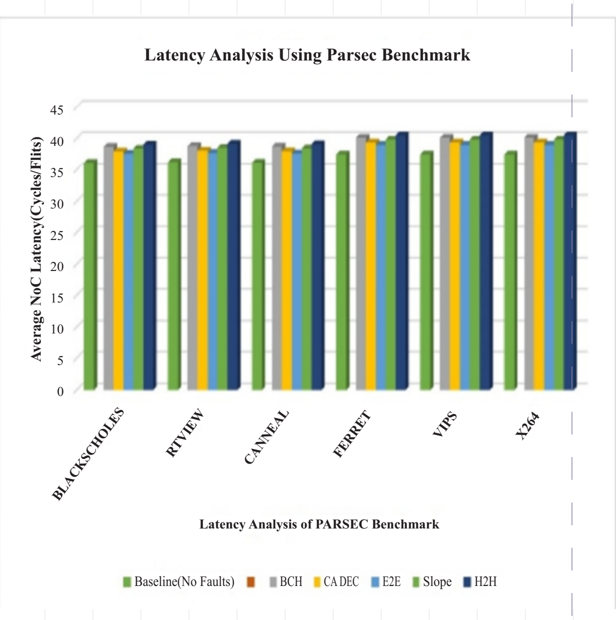


Fig. 8. Comparison on the basis of PARSEC Benchmark

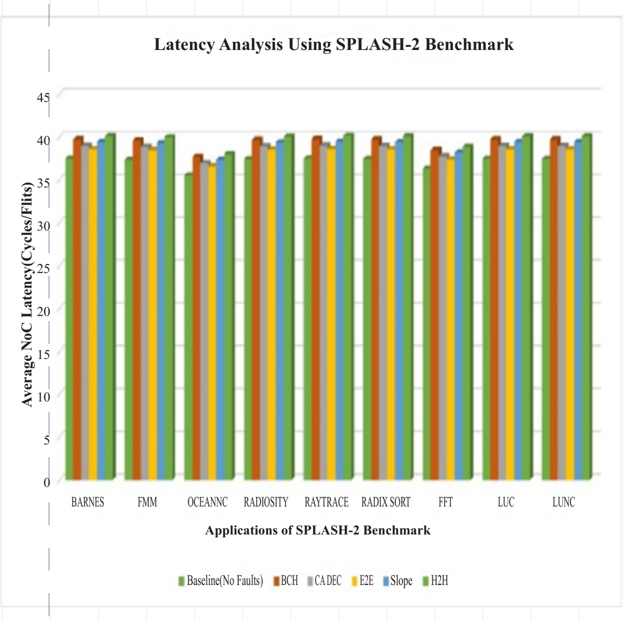


Fig. 9. Comparison on the basis of SPLASH-2 Benchmark

We simulated the desired configuration for the synthetic traffic patterns and observed the effect on latency with increase injected packets. We injected increasing number of randomly generated faults in at the varying inject rate and compared their correction capability and impact on the latency. For uniform random and Tornado traffic pattern, it is observed that as the number of injected faults increased the fault correction capability of the proposed slope is equal to BCH with a minor overhead of latency. The fault correction capability of CADEC is higher as compared to the proposed E2E mode, and E2E is not capable of correcting accumulated faults. To overcome this situation our proposed switching model, change a transition to H2H state which corrects the more considerable number faults with some minor increased overhead of latency as shown in Fig.6 and Fig7.The proposed model is also evaluated for real-time benchmark traffic pattern. The impact on latency with an increased number of injected faults is shown in Fig.8 and Fig.9. Our proposed model has the advantage of shifting different modes to tackle a different number of faults.

The Error correction scheme E2E BCH has error correction capability 3 errors at the destination router. If more than 3 number of errors occurred, then this technique failed to solve that problem. For CADEC, it can solve maximum 2 errors occurred during its path traversal.

In our proposed scheme there are three types of switching modes for tackling a different number of faults. We have utilized adaptive routing in the network to increase the error resiliency of the proposed method. For a fair comparison, we have utilized both schemes for adaptive routing and observed the number of uncorrected flits by these schemes. Our method shows better resiliency to tolerate more faults in the network. For less than 3 number of faults in the network can be tolerated by E2E switching protocol. The correction schemes BCH and CADEC tolerate these faults. The number of uncorrected flits is being monitored if the flits remain uncorrected with BCH and CADEC it means they fail to correct these faults. The proposed model request for transition to a Pre-Slope state where ECC protection is applied to the optimized locations. The slope handles a more substantial number of faults by utilizing the adaptive routing with the help of optimized ECC locations. If the number of uncorrected flits exceeds then, it will switch to H2H switching mode which now can tolerate single fault at each hop. In this way the error resiliency of the proposed model's increases.

VI. CONCLUSION

In this work, adaptive error control mechanism is efficiently utilized and extended to operate in three mode including End to End, slope, and Hop to Hop protection. In this work, we combined NTC with NoC and used switching model to tolerate the faults occurring in the network due to NTC. The error detection outcome and analysis show that this switching results in better reliability and improved performance in term of energy. Simulation results show that the proposed model achieves better error correction capability and improved network performance.

REFERENCES

[i] S. Borkar, “Design challenges of technology scaling,” IEEE Micro, vol. 19, no. 4, pp. 23–29, Jul./Aug. 1999.

[ii] S. Borkar, “Thousand core chips: a technology perspective,” in Proc. IEEE Design Autom. Conf., 2007 pp. 746–749.

[iii] L. Benini and G. De Micheli, “Networks on chips: A new SoC paradigm,” IEEE Comput., vol. 35, no. 1, p. 70–78, Jan. 2002.

[iv] S. Kumar, A. Jantsch, J. P. Soininen, M. Forsell, M. Millberg, J. Oberg, K. Tiensyrja, and A. Hemani, “A network on chip architecture and design methodology,” in Proc. IEEE Comput. Soc. Annu. Symp. VLSI, 2002, pp. 105–112.

[v] H. Zimmer and A. Jantsch, "A fault model notation and error-control scheme for switch-to-switch buses in a network-on-chip," in Hardware/Software Codesign and System Synthesis, 2003. First IEEE/ACM/IFIP International Conference on, 2003, pp. 188-193.

[vi] Report to Congress on server and data center energy efficiency, [U.S. Environmental Protection Agency. [Online]. Available: http://www.energystar.gov/ia/partners/prod_development/downloads/EPA_Datacenter_Report_Congress_Final1.pdf

[vii] H. Esmaeilzadeh, E. Blem, R. St Amant, K. Sankaralingam, and D. Burger, "Dark silicon and the end of multicore scaling," in ACM SIGARCH Computer Architecture News, 2011, pp. 365-376.

[viii] U. R. Karpuzcu, A. Sinkar, K. Nam Sungand, and J. Torrellas, “EnergySmart: Toward energy-efficient manycores for Near-Threshold Computing,” in Proc. IEEE 19th Int. Symp. High-PerformanceComput. Archit., 2013, pp. 542–553.

[ix] R. G. Dreslinski, M. Wieckowski, D. Blaauw, D. Sylvester, and T. Mudge, "Near-threshold computing: Reclaiming moore's law through energy efficient integrated circuits," Proceedings of the IEEE, vol. 98, pp. 253-266, 2010.

[x] R. G. Dreslinski, M. Wieckowski, D. Blaauw, D. Sylvester, and T. Mudge, "Near-threshold computing: Reclaiming moore's law through energy efficient integrated circuits," Proceedings of the IEEE, vol. 98, pp. 253-266, 2010.

[xi] C. Rajamanikkam, J. Rajesh, K. Chakraborty, and S. Roy, "BoostNoC: Power efficient network-on-chip architecture for near threshold computing," in Computer-Aided Design (ICCAD), 2016 IEEE/ACM International Conference on, 2016, pp. 1-8.

[xii] S. Mittal, "A survey of architectural techniques for near-threshold computing," ACM Journal on Emerging Technologies in Computing Systems (JETC), vol. 12, p. 46, 2016.

[xiii] R. Marculescu, U. Y. Ograst, L. Peh, N. E. Jergere, and Y. Hoskote. Outstanding research problems in noc design: system, micro architecture, and circuit perspectives. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 28(1):3{21, 2009.

[xiv] M. Radetzki, C. Feng, X. Zhao, and A. Jantsch. Methods for fault tolerance in networks-on-chip. ACM Computing Surveys (CSUR), 46(1):8, 2013.

[xv] L. Li, N. Vijaykrishnan, M. Kandemir, and M. J. Jrwin, “Adaptive error protection for energy efficiency,” in Proc. ICCAD, 2003, pp. 2–7.

[xvi] Q.Yu and P. Ampadu, “Adaptive error control for nanometer scale NoC links,” IET Comput. Digit. Tech., vol. 3, no. 6, pp. 643–659, Nov. 2009.

[xvii] D. Rossi, P. Angelini, and C. Metra, “Configurable error control scheme for NoC signal integrity,” in Proc. IOLTS, 2007, pp. 43–48.

[xviii] M. Ali, M. Welzl, S. Hessler, and S. Hellebrand, “An efficient fault tolerant mechanism to deal with permanent and transient failures in a network on chip,” Int. J. High Perform. Syst. Arch., vol. 1, no. 2, pp. 113–123, Jan. 2007.

[xix] A. Sanusi and M. A. Bayoumi, “Smart-flooding: A novel scheme for fault-tolerant NoCs,” in Proc. IEEE SoC Conf., 2009, pp. 259–262.

[xx] Y.-C. Lan, M. C. Chen, W.-D. Chen, S.-J. Chen, and Y.-H. Hu, “Performance-energy tradeoffs in reliable NoCs,” in Proc. ISQED, 2009, pp. 141–146.

[xxi] B. Fu and P. Ampadu, “On Hamming product codes with type-II hybrid ARQ for on-chipinterconnects,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 56, no. 9, pp. 2042–2054, Sep. 2009.

[xxii] J. Wang, W. Zhang, Z. Junwei, K. Qiu, and T. Li, "On the Implication of NTC vs. Dark Silicon on Emerging Scale-out Workloads: The Multi-core Architecture Perspective," IEEE Transactions on Parallel and Distributed Systems, 2017.

[xxiii] N. Agarwal, T. Krishna, L.-S. Peh, and N. K. Jha, "GARNET: A detailed on-chip network model inside a full-system simulator," in Performance Analysis of Systems and Software, 2009. ISPASS 2009. IEEE International Symposium on, 2009, pp. 33-42.

[xxiv] S. C. Woo, M. Ohara, E. Torrie, J. P. Singh, and A. Gupta, “The SPLASH-2 programs: Characterization and methodological considerations,” in Proc. Int. Symp. Comput. Archit., 1995, pp. 24–36.

[xxv] Q. Yu and P. Ampadu, "Adaptive error control for NoC switch-to-switch links in a variable noise environment," in Defect and Fault Tolerance of VLSI Systems, 2008. DFTVS'08. IEEE International Symposium on, 2008, pp. 352-360.