© 2014 Raghavendra Pradyumna Pothukuchi

SYSTEMATICALLY CONTROLLING THE ERROR RATES IN VARIATION-PRONE NETWORKS-ON-CHIP FOR ENERGY EFFICIENCY

BY

RAGHAVENDRA PRADYUMNA POTHUKUCHI

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2014

Urbana, Illinois

Adviser:

Professor Josep Torrellas

ABSTRACT

Networks-on-Chip (NoCs) are prone to within-die process variation as they span the whole chip. To tolerate variation, their voltages (V_{dd}) carry over-provisioned guardbands. As a result, prior work has proposed to save energy by dynamically managing V_{dd}, operating at reduced V_{dd} while occasionally suffering and fixing errors. Unfortunately, these proposals use ad-hoc controller designs that may not work under other scenarios and do not provide error bounds.

This thesis develops a scheme that dynamically minimizes the V_{dd} of groups of routers in a variation-prone NoC using formal control-theory methods. The scheme, called *Contra*, saves substantial energy while guaranteeing the stability and convergence of error rates. Moreover, the scheme is enhanced with a low-cost secondary network that retransmits erroneous packets for higher energy efficiency. The enhanced scheme is called *Contra+*. Both Contra and Contra+ are evaluated using simulations of NoCs with 64–100 routers. In an NoC with 8 routers per V_{dd} domain, the proposed schemes reduce the average energy consumption of the NoC by 27%; in a futuristic NoC with one router per V_{dd} domain, Contra+ and Contra reduce the average energy consumption by 37% and 32%, respectively. The performance impact is negligible. These savings are significant over the state-of-the-art. The results categorically state that formal control is essential to attain a stable, scalable, and energy-efficient design. Additionally, it is found that while the secondary network helps Contra+ attain higher energy savings, it has a non-negligible hardware cost. Hence, Contra is the most cost-effective design.

Kṛṣṇārpaṇamastu
dedicated to Krishna

ACKNOWLEDGMENTS

At the outset, I express my gratitude to Prof. Torrellas, for giving me the opportunity to work on this project, his insightful advice and constant support. I am also indebted to Amin Ansari for introducing me to the idea and guiding me throughout the project. I would also like to express my thanks to Bhargava Reddy for his help in this work. I am fortunate to have the company of Aditya, Tom, Jiho, Wooil, Yasser, Mengjia, Tanmay and other iacoma members for not only their helpful feedback on this work but also for all the time that we spent together. It has been a great experience and hope that it will continue to be. I also thank other graduate students of the Architecture area at UIUC for the interaction during the reading group and elsewhere, that was instrumental in sharpening my understanding of architecture and keeping my knowledge current.

It is my fortune to have a large group of friends who bore with me through thick and thin, leaving me with plenty of good things to fondly remember each one of them for years to come. Be it advanced architecture or soy or Moore's law, they were always there to... fight. In this limited space here, I cannot list them all, but my heartfelt thanks go to each one of them.

While I want to, I don't think I can express enough thanks to my family - any finite expression would be belittling for all the love, care, patience and support they have unconditionally showered on me all these years.

I am forever in debt to my teachers at school, BITS Pilani, UIUC and elsewhere who transformed me to what I am now. There are many others who touched my life through its course, gently nudging, guiding, helping and channelling it to give it the form it has now, and the momentum for it flow. I am grateful to all of them.

All this effort is attributed to that Queen beyond the Absolute Truth, the *Mahā Māya*, who revolves the wheel of this world -"*Mahāmāyā viśvam bhramayasi parabrahma mahiṣi*" (*Sauṃdaryalahari, Ādi Śaṃkarācārya*)

TABLE OF CONTENTS

LIST OF TABLES vii

LIST OF FIGURES viii

CHAPTER 1 INTRODUCTION 1

CHAPTER 2 BACKGROUND 4
2.1 Process Variations and Resilience-Energy Tradeoff 4
2.2 Modeling Timing Faults 4
2.3 Voltage Regulation for Multiple V_{dd} Domains 5
2.4 Ad hoc Approaches for Energy Savings 5
2.5 Formal Control for Stable Energy Reduction 6
2.6 The Tangle Architecture 6

CHAPTER 3 CONTRA ARCHITECTURE 8
3.1 Controller Design 8
3.2 *Contra+* : Low-Cost Secondary Network 15

CHAPTER 4 DESIGN ISSUES 18
4.1 Error Handling 18
4.2 Multiple Routers per V_{dd} Domain 19
4.3 Controller Characteristics and Aging 19
4.4 Costs of the Contra and Contra+ Designs 20

CHAPTER 5 EXPERIMENTAL METHODOLOGY 21

CHAPTER 6 EVALUATION 25
6.1 Comparing the Different Schemes 25
6.2 Contra/Contra+ Characterization 31

CHAPTER 7 CONTRA/CONTRA+ DESIGN SPACE EXPLORATION 34
7.1 Error-Rate Threshold for Activating Controller: 34
7.2 Reference Error-Rate: 34
7.3 Link Width of the Secondary Network: 35
7.4 Fat-Tree Bandwidth: 36

CHAPTER 8 RELATED WORK 38

CHAPTER 9 CONCLUSION 40

REFERENCES 41

LIST OF TABLES

5.1 NoC architectures compared. 23
5.2 Architecture and variation parameters. For memory hierarchy latencies, we give round-trip latencies from the core. . . 24

LIST OF FIGURES

3.1 Probability of a timing error as a function of V_{dd} for 64 routers. Each curve represents the behavior of a router. 9
3.2 Overview of Contra. 13
3.3 Implementation of the controller using adders and lookup tables. 14
3.4 A labeled AVL tree and the internals of a tree switch. 16
3.5 Simple routing logic for the tree switches. 17

6.1 Comparing the energy savings, average error rate, performance overhead, and maximum error rate of the different schemes. Energy and performance are normalized to the NoC without V_{dd} reduction (Baseline). The figure shows data for 64- and 100-router NoCs. 26
6.2 Variation in the V_{dd} of each router over time (in epochs) in a 64-router NoC running the GemsFDTD application. Note that *both voltage regulators* in the hierarchical VR contribute to the voltage reduction. 27
6.3 Variation of error rate over time (in epochs) in four routers. . 29
6.4 Comparing the energy savings and performance overhead of the different schemes with 8-router V_{dd} domains. 30
6.5 Energy Savings across NoC sizes and available V_{dd} domains in Contra and Contra+ 31
6.6 Impact of minimum V_{dd} step on Contra and Contra+ 33
6.7 Utilization of the highest and the lowest (complete) levels of the secondary network for different NoC sizes. 33

7.1 Avg. error rate for different controller activation thresholds. . 35
7.2 Avg. error rate for different reference error rates. 35
7.3 Utilization of the highest and the lowest (complete) levels of the secondary network for different link widths. 36
7.4 Utilization of the highest and the lowest (complete) levels of the secondary network for different fat-tree bandwidth provisioning schemes 37

CHAPTER 1

INTRODUCTION

Aggressive performance-enhancing techniques combined with the non-ideal scaling of CMOS devices have resulted in an exponential increase in the power densities of processor chips [1], making energy a primary constraint in processor design [2]. Adding to the energy crisis is the issue of variability in process, temperature and supply voltage (V_{dd}) parameters — the result of the reduction in V_{dd} and feature dimensions [3]. To build resilient chips that can tolerate such variations, designers are forced to use conservative guardbands, defeating the goal of energy efficiency.

Networks on Chip (NoCs) are especially prone to such variations. They connect distant parts of the chip which, due to variations, exhibit different characteristics. Hence, the NoC has to be designed to safely work in the slowest and in the leakiest parts of the chip. This leads to conservative designs and energy inefficiencies. Unfortunately, the NoC already consumes a substantial fraction of the on-chip energy — potentially up to 30–40%, according to the literature [4, 5, 6, 7, 8, 9] — and this contribution may increase in communication-dominated future exascale systems [10]. As a result, we need to find novel NoC solutions that balance the opposing goals of low energy consumption and variation tolerance.

A possible approach to achieve energy efficiency in a variation-affected environment is to operate the design at reduced guardbands, occasionally suffering and fixing errors. Prior proposals such as Razor [11] and BlueShift [12] have used circuit techniques to run cores at high clock frequencies for the available timing guardbands. Recently, Tangle [13] has reduced the V_{dd} guardbands of different routers in a variation-affected NoC without changing their frequency.

Nearly all of these proposals, however, use *ad hoc* decisions, relying on empirically-tuned settings to control the frequency or the V_{dd} that leads to improved operation. While such rules or thresholds work for the specific

environments analyzed, we do not know how well they work under other conditions. Moreover, we do not know how close we are to the upper bound of the possible gains. Lastly, re-using the design for future architectures leads to the full search of a large design space, as the current design is highly tuned to the current environment.

To address these limitations, we require *formal* methodologies. In this work, we use formal control techniques to design a V_{dd} controller. Our goal is to dynamically control the V_{dd} of groups of routers in a variation-prone NoC, keeping V_{dd} at its lowest values while sustaining a small but tolerable number of errors. The design, called *Contra*, provides guarantees on the convergence, stability, and maximum resulting error rates for the NoC under control. The result is a robust, scalable and energy-efficient system.

We propose two Contra designs, a basic version and an aggressive variant called *Contra+*. The latter additionally includes a low-cost secondary network for higher energy efficiency. The secondary network operates at nominal V_{dd}, and retransmits a packet when it has suffered an error in the primary NoC. Retransmitting the packet on the primary NoC could result in repeated errors, which would then force the controller to increase the steady-state V_{dd} suboptimally.

We evaluate Contra+ and Contra with simulations of variation-affected NoCs with 64–100 routers running a mix of workloads. With 8 routers per V_{dd} domain, our schemes reduce the average energy consumption of the NoC by 27% with negligible performance overhead. For a futuristic design with one router per V_{dd} domain, Contra+ and Contra reduce the average energy consumption of the NoC by 37% and 32%, respectively, with negligible performance impact. These savings, which already include the penalty of power losses in voltage regulators, are substantially higher than those attained by prior approaches. While the secondary network helps Contra+ attain higher energy savings, its non-negligible hardware cost makes Contra+ less cost-effective than Contra.

Overall, the contributions of this work are:

- The design of Contra, a scheme that dynamically minimizes the V_{dd} of groups of routers in an NoC using formal control-theory approaches.
- Contra's enhancement into Contra+, a scheme that further integrates a low-cost secondary network operating at nominal V_{dd} for higher energy

efficiency.

- An evaluation of Contra and Contra+ that demonstrates large and controllable energy reductions with negligible performance impact over the best existing approaches.

This thesis is organized as follows: Section 2 presents the motivation and background; Sections 3 and 4 describe our designs; Sections 5 and 6 evaluate them; and Section 8 describes related work.

CHAPTER 2

BACKGROUND

2.1 Process Variations and Resilience-Energy Tradeoff

With decreasing feature sizes, process variations have become an important concern for chip manufacturers [3]. In this paper, we are interested in Within-Die (WID) process variations. WID variations have a systematic and a random component [14, 15]. The random component is due to random dopant fluctuations, while the systematic component is typically due to imprecisions in the manufacturing process, and exhibits significant spatial correlation. WID variations may reduce the delay in some paths and increase it in others. This results in higher V_{dd} guardbands and lower operating frequencies at the chip level, as dictated by the slowest paths.

Aggressively reducing V_{dd} or frequency guardbands decreases energy consumption but can create timing violations, as some of the variation-affected paths may be too slow. If such violations can be detected and corrected without significant overheads, overall energy efficiency can improve. This insight has been used by several authors. Specifically, Razor [11] and BlueShift [12] decrease the frequency guardbands in processor pipelines. Similarly, Tangle [13] reduces V_{dd} guardbands in NoCs while keeping the frequency unchanged. A related approach is Hi-ECC [16], which saves energy by reducing DRAM refresh frequency at the expense of increasing the strength of ECC codes.

2.2 Modeling Timing Faults

As V_{dd} decreases, certain paths may start missing timing under certain logic values and cause intermittent faults. This timing fault rate increases as V_{dd}

decreases. VARIUS-NTV [17] is a tool that models process variations and the timing violations that ensue from reduced guardbands. VARIUS-NTV takes a certain logic structure (e.g., the synthesized RTL implementation of an NoC router), and gives the probability of a timing error for each path in the different pipeline stages, for a given V_{dd}. In this work, we assume that, whenever a timing violation occurs, it causes an incorrect execution.

2.3 Voltage Regulation for Multiple V_{dd} Domains

An effective approach to tolerate process variation is to divide the circuit into multiple V_{dd} domains. In this way, a high V_{dd} can be applied to sections of the circuit that are slow due to systematic variations, and a low V_{dd} to sections that are fast. In an NoC, a natural V_{dd} domain is a set of neighboring routers.

Currently, using multiple on- or off-chip Switching Voltage Regulators (SVR) for multiple V_{dd} domains consumes significant area and power [18]. However, upcoming technology is likely to provide better solutions. One approach may involve integrating V_{dd} regulators hierarchically [19, 20]. The first level consists of a few SVRs on a stacked die, while the second level consist of many on-chip low-drop-out (LDO) regulators. Each LDO is fed by one of the SVRs and provides the V_{dd} for a domain. The area overhead of LDOs is negligible, as they reuse the hardware for power-gating a circuit. In addition, LDOs have high efficiencies if the ratio of their output (V_O) to input (V_I) is nearly 1. Also, level converters, required for communication across V_{dd} domains, can be efficiently designed by combining them with latches [21]. In this paper, we use 8 routers per V_{dd} domain for a conservative design, and 1 router per V_{dd} domain for an aggressive, futuristic design. To keep the analysis simple, we assume that providing multiple V_{dd} domains in the NoC wastes 10% of the total power.

2.4 Ad hoc Approaches for Energy Savings

Prior schemes such as Tangle [13] achieve energy savings in the context of process variation by devising rules that are tuned entirely on intuition and

empirical observations. However, they lack the rigor and development of the intuition that would enable architects to place certain guarantees on the cost and performance of the components they design. As a result, they suffer from two limitations: conservativeness and unreliability. In [13], for example, a regulator periodically reduces the voltages of the routers in an NoC, and if an error occurs, immediately raises the voltage of all routers in the path of that flit. This conservative approach cannot even guarantee that a stable operating point exists, let alone achieving it. Thus, the worst possible behavior of the scheme is not quantifiable, and the rest of the system has to deal with this uncertainty somehow. Of course, the energy savings are subopimtal due to the conservative voltage update mechanism.

2.5 Formal Control for Stable Energy Reduction

Proposals such as those described above are typically controlled by ad-hoc techniques that rely on empirically-found rules and thresholds. As a result, these techniques may not work under other scenarios, or need substantial modification.

We take a different approach by adopting a systems perspective and use control-theoretic techniques, which are formally derived and manage the system in an organized manner. Such techniques gracefully adapt to changes, and provide guarantees on stability, convergence and error bounds. PID controllers (Proportional, Integral, Differential) are frequently used due to their simplicity and effectiveness [22].

2.6 The Tangle Architecture

Our Contra architecture builds on Tangle [13]. Tangle uses an ad-hoc hardware controller to dynamically reduce the V_{dd} of groups of routers in an NoC while monitoring the error rate. The goal is to save energy without incurring intolerable error rates. The controller operates in *epochs*, which are fixed time intervals. At the beginning of each epoch, the controller reduces the V_{dd} of all the router groups in the system by ΔV_{dec}. Then, for the rest of the epoch, it monitors the errors that occur in the NoC. If it observes an error,

it increases the V_{dd} of a subset of the routers by ΔV_{inc}. With this approach, the V_{dd} of each group of routers converges to a low but safe value.

Erroneous flits are detected using switch-to-switch CRC and are dropped. When the controller detects an error, it increases the V_{dd} of *all* the routers in the path of the faulty flit. Eventually, the sender node times-out and retransmits the packet containing the dropped flit using the same network.

When an error is detected, Tangle increases the voltages of *all* routers in a path traversed by a faulty flit instead of rectifying the error source alone. This leads to conservative and sub-optimal energy savings.

CHAPTER 3

CONTRA ARCHITECTURE

Contra's goal is to dynamically reduce the V_{dd} of groups of routers in a variation-prone NoC (without changing the frequency) in a way that *guarantees* the convergence and stability of V_{dd}, and the bounds on error rates. For this, Contra uses a controller developed using *formal* control-theoretic approaches. The controller updates the voltages of the routers using a light weight control network similar to that used in [13]. We also present a more advanced design called *Contra+* that in addition, eliminates Contra's control network and adds a low-cost secondary network to save additional energy.

In Contra, the controller changes the V_{dd} only once per epoch — at the end of the epoch, irrespective of the number of errors observed. The length of the epochs is adaptive, in that if there are only a few flits in the network, the controller postpones any V_{dd} tuning on a router until enough flits have been seen by the router to make a decision. When Contra increases the V_{dd}, it does so only on the router (or group of routers) that logged the error.

This section describes the formal controller and the secondary network. The discussion implicitly assumes the availability of a V_{dd} domain per router. However, our evaluation will assess both a conservative NoC design with 8 routers per V_{dd} domain and a futuristic one with one router per V_{dd} domain.

3.1 Controller Design

3.1.1 Modeling the Error Rate:

The design of a formal controller requires a good model for the relationship between the error rate at a router and V_{dd}. Figure 3.1 shows the probability of a timing error occurring in an NoC router as a function of its V_{dd}, as obtained from studies on a 64-router NoC using VARIUS [15] and other

tools (described in Section 5).

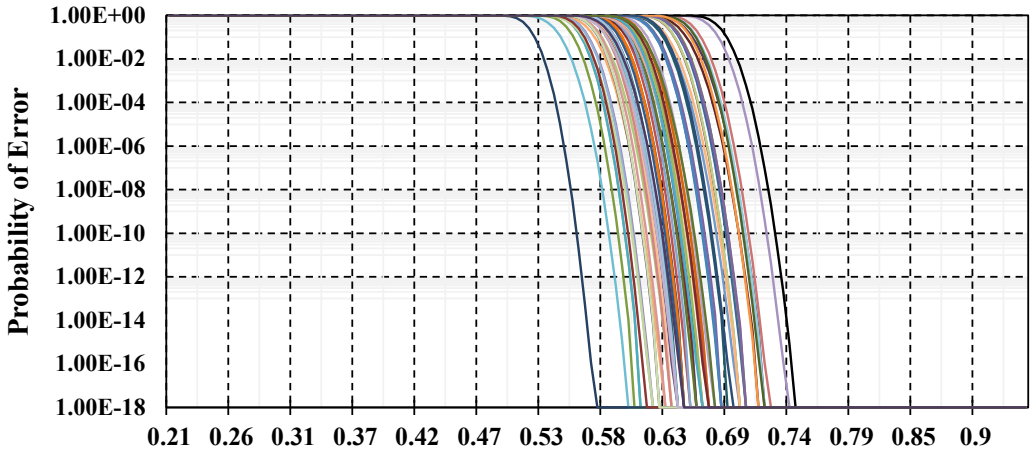


Figure 3.1: Probability of a timing error as a function of V_{dd} for 64 routers. Each curve represents the behavior of a router.

This graph shows an exponential increase in the probability of an error as V_{dd} is reduced. Due to process variations, the minimum error-free V_{dd} for different routers is spread across a wide range. For V_{dd} values above $750mV$, all the 64 routers are largely free of failure; the chance of failure for any stage of any router is less than 10^{-18} failures per cycle. However, some routers can operate at a V_{dd} as low as $570mV$ without failure. At the extremes of each curve, the V_{dd} is either too low that the probability of failure is 1, or it is high enough that the probability is 0. The desired point of operation is in the transition region that sustains a small error rate.

To mathematically analyze the system, we first model the "S" shaped probability distribution function using different sigmoid functions, such as the Error Function (*erf(x)*), Logistic Distribution, and Gompertz functions. More details on the properties of these functions and their applicability for sigmoid distributions can be found in [23, 24, 25, 26]. We find that an Error Function *erf(x)*, shown in Eq. (3.1), fits the probability distribution closely. Eq. (3.2) shows the probability of an error as a function of the router V_{dd}, after normalizing the V_{dd} distribution. Then, we linearly approximate the central region of this function using a first order Taylor polynomial shown in Eq. (3.3). Such linearization techniques are required and widely employed to enable the use of linear controllers. Linear controllers are well-studied and have several analysis tools that facilitate their implementation. Hence, we use a linear controller for our design [22].

$$erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x \mathrm{e}^{-t^2} \mathrm{dt} \tag{3.1}$$

$$P(Error) = 0.5(1 - erf\left(\frac{V_{dd} - 0.595}{0.012}\right) + erf\left(-\frac{V_{dd} - 0.595}{0.012}\right)) \tag{3.2}$$

$$P(\widehat{Error}) = 0.5 - \frac{1}{\sqrt{\pi}}\left(\frac{V_{dd} - 0.595}{0.012}\right) \tag{3.3}$$

To bring the router's operating point within the linear region, its V_{dd} is decreased steadily from the nominal V_{dd} until the error rate crosses an activation threshold. As described in Section 7.1, this threshold can be varied to suit different design choices. Within this region, the controller is activated and begins to generate decisions to change the router's V_{dd}.

3.1.2 Controller Design Aspects:

There are two primary choices for the domain of the controller, namely, continuous-time and discrete-time. A continuous-time controller continuously changes the V_{dd} and is not realistic. A discrete-time controller changes V_{dd} in discrete steps, and is practical. Hence, we design a discrete-time controller.

The controller reduces the V_{dd} of a router in steps that become progressively smaller. As the controller reduces the V_{dd} applied, it is unable to abruptly stop the direction of V_{dd} change even after it reaches the steady state voltage, $\widetilde{V}_{dd}$. Hence, V_{dd} decreases below $\widetilde{V}_{dd}$, increasing the error rate momentarily. Then, it quickly reverses the direction of V_{dd} change and increases the V_{dd} by taking steps in the opposite direction. Now, this might increase the V_{dd} slightly above $\widetilde{V}_{dd}$. The former phenomenon is called overshoot and the latter is called undershoot. This process repeats for a few times and, each time, the magnitudes of the overshoots and undershoots become progressively smaller. Finally, the controller will stabilize the router's V_{dd} to $\widetilde{V}_{dd}$.

It is important to ensure that the overshoots are small, that there is no oscillatory behavior, and that the time to achieve stability (i.e., to reach $\widetilde{V}_{dd}$ eventually) is small. Oscillations occur when the V_{dd} values rise and fall

by the same (or increased) amount around $\widetilde{V}_{dd}$ in subsequent epochs. The controller should avoid such oscillations.

The controller should also be robust to disturbances arising from hardware or model limitations. Examples of hardware limitations are voltage regulator constraints on the minimum allowable step size and finite number of output V_{dd} levels. Model limitations arise form simplifications made during model and controller development. From Figure 3.1, we see that the linear regions of all routers have nearly the same slope, representing similar input-output dynamics. This intuition simplifies our design because we can reuse the same controller constants for all routers (requiring only error histories to be separately maintained). Hence, we use the statistical average behavior of the routers in the linear region to develop the model in Equation 3.3.

It is possible for an individual router to deviate from this model. To account for these factors, we impose uncertainty constraints on the controller and add suitable gain and phase margins to ensure the robustness of the controller subject to such uncertainties.

3.1.3 Closed-Loop PID Controller:

Considering the requirements and constraints of the controller operation mentioned above, we design a discrete-time closed-loop PID (Proportional, Integral, Differential) Controller with output feedback. PID controllers have been widely used in several domains due to their flexibility and wide range of applicability in linear systems [22]. The PID controller takes the error rate of a router and the reference error rate ($E_\circ$) as inputs and generates a change in the V_{dd} of that router. The output $\Delta V_n(i)$, which is the change from the current V_{dd} for the nth router in the ith epoch, is given by (simplified form):

$$\Delta V_n(i) = K_P \Delta E_n(i) + K_I \sum_{k=0}^{i} \Delta E_n(k) + K_D \{\Delta E_n(i) - \Delta E_n(i-1)\} \qquad (3.4)$$

where $\Delta E_n(i) = E_n(i) - E_\circ$, and $E_n(i)$ is the error rate for the nth router at the beginning of ith epoch. The first term in Eq. 3.4 represents the Proportional Gain (G_P), i.e., the change in V_{dd} proportional to the deviation of $E_n(i)$ from $E_\circ$. The second term represents the Integral Gain (G_I), i.e., the change in V_{dd} proportional to the sum of all the deviations that occurred

before. The last term represents the Differential Gain (G_D), i.e., the change in V_{dd} proportional to the rate at which the deviation has changed from the previous epoch to this epoch. K_P, K_I, and K_D are the proportionality constants in these terms.

For a general system, using a simple controller with Proportional Gain alone would result in a steady-state error. Therefore, we add an Integral Gain that can nullify this error. However, the integral component tracks only the history and may not be responsive enough for large deviations (e.g., an increase in error rate). To incorporate a timely reaction to large deviations and use a predictive action, we also add a Differential Gain.

As the controller generates a change in V_{dd} as a function of the deviation from the reference value ($\Delta E_n(i)$), its output may become very small as $\Delta E_n(i)$ approaches zero. This can happen when the router's V_{dd} gets closer to the steady state value. However, such arbitrarily small steps may not be possible due to limitations of V_{dd} regulators. As a result, we restrict the minimum magnitude of the step which the controller produces to the minimum tuning step of the V_{dd} regulator, V_{MS} (for example, $10mV$). However, this might lead to sudden increases in the error-rate when the controller decides to reduce the V_{dd} by a small amount but the V_{dd} regulator pushes it down by V_{MS}. To mitigate this effect, V_{dd} reductions smaller than $2mV$ are ignored. In addition, to avoid arbitrary steps that cannot be generated by the V_{dd} regulator, we quantize the output of the controller to the nearest multiple of V_{MS} and generate $\Delta\hat{V}_n(i)$. The following equation shows the change in V_{dd}, in relation with $\Delta V_n(i)$ after imposing these constraints.

$$V_n(i) = \begin{cases} V_n(i-1); & \text{-}2mV < \Delta V_n(i) < 0 \\ V_n(i-1) + \Delta\hat{V}_n(i); & \text{otherwise} \end{cases}$$

Figure 3.2 illustrates the overall design. The NoC is divided into several V_{dd} domains, each consisting of a group of routers. The controller is time-shared across all routers. It monitors their error rates and generates the change in their V_{dd} values as the output. After all the routers in a V_{dd} domain have a decision, the V_{dd} for the entire domain is generated by changing the value of the V_{dd} regulator (VR) that is connected to that domain. This step is detailed in Section 4.2. The figure shows the controller generating a decision for the nth router in the jth V_{dd} domain at the beginning of the ith epoch,

using $E_n(i)$ and $E_\circ$. This decision is sent to the VR, which finally adjusts the V_{dd} of the whole domain.

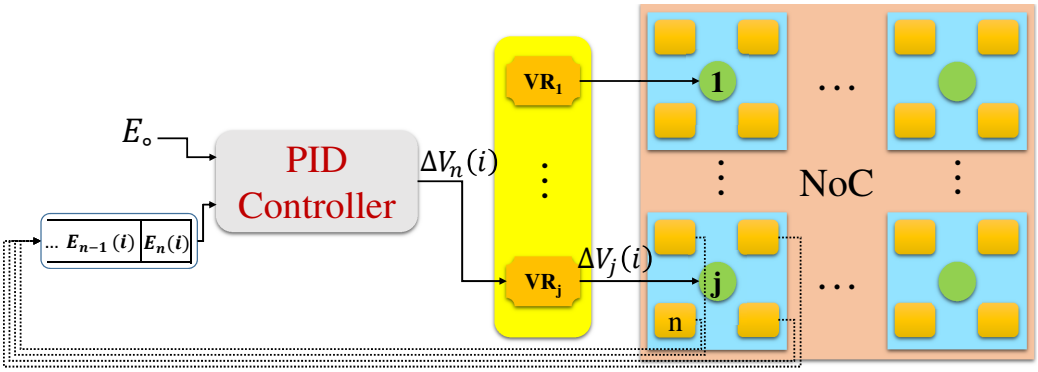


Figure 3.2: Overview of Contra.

To design the controller, we use the z-transform [22] of the difference equation of error rate as a function of V_{dd}, and tune the PID controller using MATLAB. Using the root-locus method to stabilize the controller, we obtain a range of values for K_P, K_I and K_D. We start with a settling time of 35 and target the peak overshoot to be 0.1% error rate. We then tune K_P, K_I and K_D to minimize the deviations around the reference error-rate after the controller enters the stability band (after the settling time). In this band, we define the acceptable error margin to be 5% of the reference error-rate. Using pole-zero analysis to keep the settling time, overshoot and undershoot acceptable, we find the values of K_P, K_I and K_D to be -2.9966×10^{-3}, -4.9766×10^{-6} and -7.1804×10^{-2}.

3.1.4 Hardware Implementation:

The controller performs three steps:

1. It calculates the error rate of the nth router at the beginning of the ith epoch, $E_n(i)$, by dividing the number of errors by the total number of flits. We approximate this division by using bit shifts.

2. It generates the gains in Eq. 3.4 (G_P, G_I, and G_D), using the hardware shown in Figure 3.3. First, the controller generates $\Delta E_n(i)$, which is the deviation of the current error rate $E_n(i)$ from the reference error rate $E_\circ$. At the same time, it obtains the previously accumulated deviation,

$\sigma_n(i-1)$ and the previous deviation, $\Delta E_n(i-1)$ from two error tables indexed by the router number, n. Next, it uses $\Delta E_n(i)$ to obtain the Proportional Gain, G_P, from a 128-entry lookup table of precomputed gains, similar to those used in [27]. Simultaneously, it uses $\sigma_n(i-1)$ and $\Delta E_n(i-1)$ to calculate the current accumulated deviation ($\sigma_n(i)$) and the difference from the previous deviation. Then, these results are used to index into tables of precomputed gains to obtain G_I and G_D respectively. $\sigma_n(i)$ and $\Delta E_n(i)$ are written back to the error tables for use in the next epoch.

3. Finally, it generates the required change in V_{dd} by adding the three gains. A comparator is used to ignore downward changes smaller than $2mV$. This output is quantized based on the V_{dd} regulator output. For example, a controller output of $7.3mV$ is changed to $10mV$ if the smallest step the V_{dd} regulator can take is $10mV$.

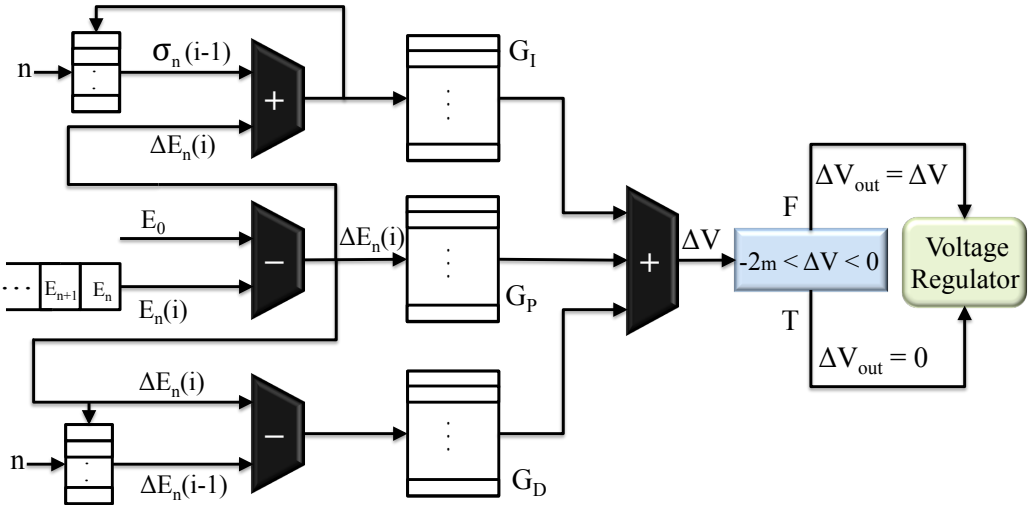


Figure 3.3: Implementation of the controller using adders and lookup tables.

Fixed-point adders are used for all the additions and subtractions. The V_{dd} tuning happens only once in an epoch and the latency of this process does not impact performance. The overheads are further reduced by adapting the epoch lengths.

3.2 *Contra+* : Low-Cost Secondary Network

We use switch-to-switch CRC to detect an erroneous flit, and then retransmit the corresponding packet from the source. This approach is an arguably reasonable energy-efficient solution.

However, Contra does not increase the V_{dd} of a router immediately when an error is detected; the V_{dd} can only be increased at the end of an epoch. This is done to avoid over-reacting after an error. Unfortunately, since the V_{dd} conditions have not changed, it is possible that the retransmitted packet ends up suffering an error as well. This process can potentially repeat multiple times, slowing down message delivery and making the controller believe that the error rate is higher than it really is. As a result, the Contra controller may end up setting the V_{dd} of the routers to a value higher than needed.

To avoid this problem, we also propose a more advanced Contra design called *Contra+* that additionally eliminates Contra's control network and adds a low-cost *Secondary Network* operating at the nominal V_{dd}. When an error occurs, the packet is retransmitted through this secondary network. This guarantees packet delivery and ensures high performance, albeit at the cost of an additional, simple network.

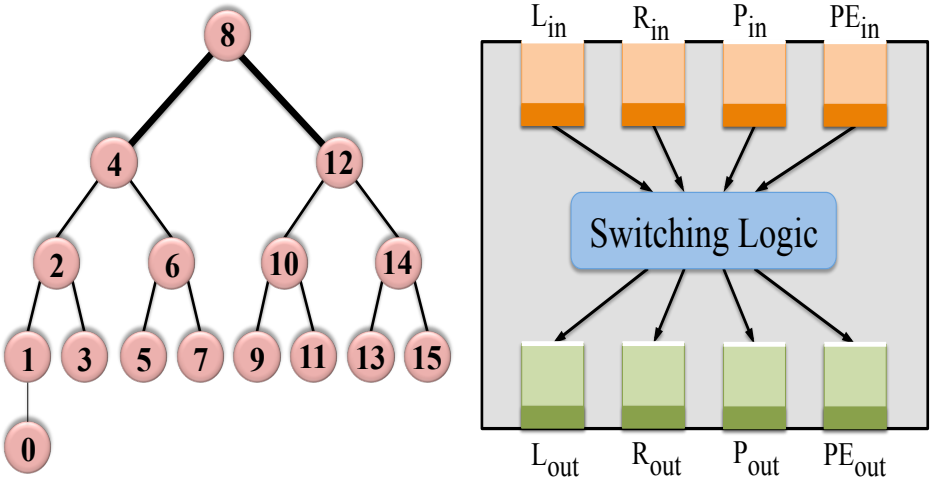
The secondary network needs to ensure that retransmission is energy-efficient and fast. In addition, the network should be scalable and simple. Finally, we note that the secondary network carries very little traffic and, therefore, its links will be lightweight. These characteristics, in particular the combination of low traffic and fast packet delivery, suggest a different topology than the primary network, which carries high traffic and serves general-purpose demands.

Therefore, while the primary network is a mesh, we choose a modified fat-tree network for the secondary network. There are three reasons for this. First, theoretical studies have established a fat-tree's benefits in terms of small average distance and scalability [28]. Secondly, there have been studies that show how to layout a fat H-tree into VLSI structures efficiently [29, 30, 31, 32]. Thirdly, we substantially simplify the routing logic, and make the tree's layout modular by organizing it as a fat AVL tree [33]. An AVL tree has processors at every level of the tree — unlike a conventional tree, which has processors only at the leaves.

An AVL tree network with N nodes has a total number of links equal to N-

1. Its diameter and the average distance grow only as logN. This enables the design of the secondary network to scale to large chips, while maintaining a small diameter and average distance. Moreover, the layout is modular because both tree leaves and intermediate tree nodes have processors. Finally, as we see next, the routing logic is very simple.

Figure 3.4a shows a 16-node AVL tree with labeled nodes. The numbering is such that, for a node labeled i, all the nodes in its right subtree have numbers larger than i, and all nodes in its left subtree have numbers smaller than i.



(a) Fat AVL tree of 16 nodes. (b) Tree switch.

Figure 3.4: A labeled AVL tree and the internals of a tree switch.

Figure 3.4b shows an AVL tree switch. The links flowing inward and outward from the switch are as follows: PE_{in} and PE_{out} connect to the local processing element (PE); L_{in} and L_{out} connect to the left child (L); R_{in} and R_{out} connect to the right child (R); and P_{in} and P_{out} connect to the parent node (P). Each of these links has a queue.

Figure 3.5 shows how the simple routing logic works. Each node has three registers, which contain: the number assigned to the node (*Self*), the largest number in the right subtree (R_{max}), and the smallest number in the left subtree (L_{min}). Given a flit in one of the input buffers, we compare its destination number (*Dest*) to these three numbers. *Dest* can fall into one of the five regions shown in Figure 3.5: lower than L_{min}, equal or higher than L_{min} but lower than *Self*, equal to *Self*, higher than *Self* but lower or equal

to R_{max}, and higher than R_{max}. Based on where it falls, the flit is routed to the parent, the left subtree, the local node, the right subtree, or the parent, respectively.

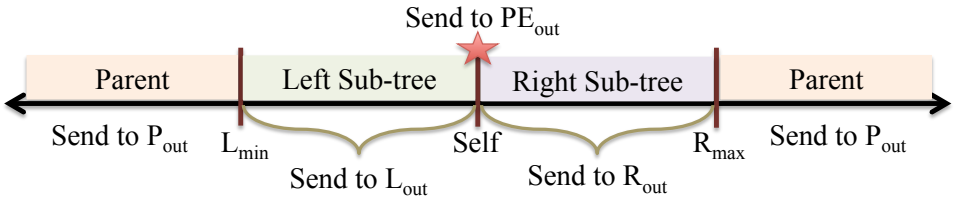


Figure 3.5: Simple routing logic for the tree switches.

The average utilization of the secondary network is low. Hence, we set the bandwidth of the links at the leaves to be 1/8th of that of the primary network links. To support the increasing utilization of the links at the upper tree levels, we double the width of the links every two levels.

CHAPTER 4

DESIGN ISSUES

4.1 Error Handling

To detect errors, Contra uses 8-bit WCDMA-8 CRC code [34]. Whenever a flit is corrupted by a router, it fails the CRC at a neighboring router and is dropped.

To calculate the error rate at every router, the total number of flits routed and the number of flits corrupted by the router since the last V_{dd} tuning decision have to be measured. To measure the former, a counter is placed in each router, which is incremented whenever the router transmits a flit. To measure the latter, counters placed in the neighboring routers are incremented whenever a corrupted flit from that router is detected. At the beginning of every epoch, the controller reads these counters using the control network (Contra) or the secondary network (Contra+).

An error condition is identified using a time-out at the source. A time-out occurs if the source does not receive an acknowledgment (ack) within a certain period. Contra leverages the acks of transactions at the protocol layer (cache coherence transactions in shared-memory systems [35]) and does not add additional acks. If an ack does not exist for a given transaction, we add it. On a time-out, the source assumes the packet is dropped and retransmits the packet on the primary network (Contra) or secondary network (Contra+).

In scenarios of network congestion, even though long-duration message stalls are unlikely in NoCs [36], it is possible that the source might pessimistically assume that a packet is dropped even though its flits are progressing slowly through the network, and re-send the packet. In this case, the destination receives two copies of the same packet. By maintaining a pair-wise sequence number of the last acknowledged packet, duplicates are identified and dropped at the destination.

An error may cause a packet to be misrouted to an incorrect destination node. In this case, when the destination receives the packet, it is dropped. It is also possible that an error causes a packet to enter a routing cycle and never be delivered anywhere. To avoid this case, each packet contains its time of injection. When such a time gets too old for a packet, a router drops it. In both cases, the source times-out and retransmits.

4.2 Multiple Routers per V_{dd} Domain

When we have multiple routers per V_{dd} domain, the operation of the controller proceeds in two steps. First, the controller makes a decision for each router in the domain, after having measured the error rate in the router. Second, the controller sets the V_{dd} of the domain to be the maximum of all the V_{dd} of the routers in that domain. To prevent a lagging router from delaying the decision of a domain indefinitely, Contra places an upper bound on the number of epochs and observed errors before making a decision for a router.

Due to this coarse granularity of V_{dd} control, the energy saved will be reduced compared to single-router V_{dd} domains. In reality, due to systematic variations, physically proximal routers are similarly affected, and their operational V_{dd} will likely be close.

4.3 Controller Characteristics and Aging

As shown in Figure 3.3, the PID controller is composed of lookup tables and adders. Each table has 128 entries, and each entry is 15 bits. Consequently, the controller's area is very modest. It is a simple circuit, with three adders and one table lookup in its longest path. Moreover, its operation is not time-critical. For example, for a 64-domain network, the controller only needs to operate 64 times every 50-microsecond epoch.

Over many months of NoC use, the routers are likely to age and change their speed characteristics slightly. At that point, we may want to re-profile the routers, recompute the K_P, K_I and K_D controller constants, and reprogram the controller. Since aging effects occur at year-level timescale [37], reprogramming on a yearly basis is not a big deal.

4.4 Costs of the Contra and Contra+ Designs

Compared to a standard NoC, a Contra/Contra+ system has four main costs, where three are similar to the state-of-the-art ad hoc approach, Tangle [13], and one is new. The three similar costs are:

1. Cost of voltage regulation (VR): The multiple V_{dd} domains of the NoC are generated by voltage regulators. As discussed in Section 2.3, VR involves a power loss, but there are many different voltage regulator designs and the technology is evolving. To keep the analysis simple, we assume that providing multiple V_{dd} domains in the NoC wastes 10% of the NoC power.

2. Cost of error detection and counting: Our results in the evaluation include the energy and performance overheads of using link-level CRC and error counters in each router. The performance overhead of using CRC is negligible as it adds only one extra cycle for encoding at the source and checking is done in a shadow path.

3. Cost of retransmission buffers: The retransmission buffers that temporarily store transmitted packets to enable retransmission in case of a fault, consume area and power. They may also stall the router if they become full. Our results include the impact of these buffers.

The new cost is the formal controller system. In addition, for Contra+, another cost is the secondary network, which replaces Contra's control network. Some of the formal controller and secondary network parameters are listed in Section 5. These two components introduce only small overheads. On average, the controller introduces less than a 1% energy overhead to the NoC (most of it static), while the secondary network introduces a 1.7% energy overhead (with comparable static and dynamic components). Our results in the evaluation include the energy and performance overheads introduced by these two components.

CHAPTER 5

EXPERIMENTAL METHODOLOGY

To evaluate the energy and performance of Contra, we use several tools. First, we take the Verilog implementation of a wormhole-switched virtual channel router from [38], and develop a 3-stage router similar to that used in [39]. Using the Synopsys Design Compiler, we perform timing analysis on this design to extract the 32 slowest paths and their netlists, for each stage. We use a 3-stage router instead of a more aggressive single-cycle one for two reasons. First, single-cycle routers tend to operate at slower clock rates. Second, they often rely on speculative operation, which makes them less energy efficient [40]. As we explore a futuristic chip design, we perform process variation modeling at $11nm$ using VARIUS-NTV [17], an updated version of VARIUS [15]. The baseline chip comprises 64 nodes with private L1 caches, 64 routers with virtual channels, and 64 banks of a shared L2. To obtain the timing failure rates due to process variation, we first generate the chip floorplan. Next, the logical efforts extracted from the synthesis phase are used to enable a better delay modeling of the different stages of routers.

To obtain performance and energy consumption metrics, we use a cycle-level microarchitectural simulator of a multiprocessor chip with an NoC [39]. The architecture parameters are shown in Table 5.2. The baseline design of the main network has 64 routers in an 8x8 2D-mesh. Each router has 5 physical channels (PCs), including the local port from the corresponding core to the router, and 2 virtual channels (VCs) per PC. A packet has at most five 128-bit flits, and we use a buffer depth of 4 flits per VC. We use a wormhole-switched 2D-mesh network with deterministic X-Y routing that uses credit-based flow control. The secondary network is a fat AVL tree. The router is implemented in structural RTL Verilog and synthesized using the Synopsys Design Compiler with Nangate $45nm$ open cell library. Based on rules given in [41] and technology parameters from the ITRS report [42], we scale it down to $11nm$.

The simulator also models the cores and caches. The cores are 2-issue out-of-order Alpha DEC EV4-like cores. Their frequency is 1 GHz to save power and enable full-system operation under a typical power envelope. Similar low frequencies have been used by researchers for large CMPs [43, 39].

We explore network configurations with 1 to 64 routers per V_{dd} domain. The default system in our evaluation has 1 router per V_{dd} domain. We also vary the network size from 4x4 to 10x10 nodes. For each network, we use VARIUS-NTV to generate chips with representative variation profiles.

To estimate the static and dynamic power in the router buffers, crossbar, and clock distribution, we use our synthesis results from the Synopsys Design Compiler. The results agree with those in DSENT [44] generated by SPICE. For example, for the power in the router buffers, our synthesis experiments estimate about $6.54mW$, while DSENT reports $6.93mW$.

We use MATLAB to obtain the linear approximation of the probability of an error as a function of router V_{dd}, and design the controller. Using pole-zero analysis, we find the values for K_P, K_I, K_D that result in a stable controller with acceptable overshoots and undershoots (Section 3.1.3).

For our experiments, we run 10 multi-programmed workloads. They are a subset of those used in [36]. Of these workloads, 3 are commercial (sap, sjas, and tpcw), 4 are engineering (429.mcf, 437.leslie3d, 459.GemsFDTD, and 483.xalancbmk) and 3 are scientific (art, ocean, and swim). We use SimPoint [45] to select representative windows of instructions for simulation, and we simulate about 10M instructions per thread after warm-up. In the experiments, we use a source time-out threshold of three times the round-trip latency, a reference error rate of 0.05%, a controller activation threshold of 0.001%, and a nominal epoch length of 50,000 processor cycles.

Our evaluation compares the energy consumption, performance, and error rates of the five NoC architectures of Table 5.1. These architectures are, in addition to Contra and Contra+: (1) a plain NoC at nominal V_{dd} (Baseline); (2) our understanding of Tangle [13]; and (3) an enhanced version of Tangle where, on an error, only the erroneous router changes V_{dd}, not all routers in the flit path (see top of Section 3). This version is called Tangle+, and represents an ad-hoc scheme that is more aggressive than Tangle and closer to Contra.

In Tangle and Tangle+, we use a fixed step size of $10mV$. The reason is that the scheme presented in [13] requires prior knowledge of a "convergence

voltage" $V_{AvgTest}$. We consider that such V_{dd} is not known for a variation-affected system, and the V_{dd} changes in an ad-hoc scheme have to be conservative. In any case, this change has minimal impact because it only affects the initial V_{dd} changes.

Table 5.1: NoC architectures compared.

Name	Architecture
Baseline	Plain NoC without the components of Section 4.4
Tangle	Tangle as described in [13]
Tangle+	Tangle except that, on an error, only the erroneous router changes V_{dd}, not all routers in the flit path (see top of Section 3)
Contra	Proposed Contra architecture
Contra+	Proposed Contra architecture plus the secondary network

Table 5.2: Architecture and variation parameters. For memory hierarchy latencies, we give round-trip latencies from the core.

Core Parameters	
Fetch, issue, and commit	2 per cycle
ROB; Ld/St queue	64 entries; 16/16 entries
Issue queue; I-fetch queue	64 entries; 32 entries
Branch (BR) predictor	Tournament (bimodal + 2-level)
BR target buffer; history table	1024 entries, 2-way; 2048 entries
Memory System Parameters	
L1 data cache	32KB, 2-way, 2 cycles latency, 64B line
L1 instr. cache	32KB, 2-way, 2 cycles latency, 64B line
L2 cache	32MB shared, static 64-bank addressing
	Bank: 8-way, 64B line, 6 cycles latency (local)
Main memory	260 cycles latency, 4 memory controllers
Main Network-on-Chip Parameters	
Topology; routing	8x8 2D-mesh; X-Y routing, wormhole
Number virtual channels	2 per physical channel
Buffer depth in router	8
Min. V_{dd} tuning step	$10mV$, 20 cycles latency
Replay buffer depth;Link width	8; 128b
Nominal V_{dd}	$825mV$ (10% guardband)
Length of an epoch	50,000 cycles (min.)
Num. routers per V_{dd} domain	64-1; 1 is default
Penalty to all Contra NoCs	10% power due to V_{dd} regulation
Secondary Network Parameters	
Topology; routing	AVL Tree; Least Common Ancestor routing
Channel width	1 (lowest level), double every two levels
Link width	16b
Number of buffers	One per link
Buffer depth	1
Nominal V_{dd}	$825mV$
Process Variation Parameters	
Tech. node; V_{dd} guardband	$11nm$; 10%
Total V_{th} (σ/μ)	12.5% (equal random & systematic)
Total L_{eff} (σ/μ)	6.25% (equal random & systematic)
Correlation range ϕ	0.1 (for both V_{th} and L_{eff})

CHAPTER 6

EVALUATION

6.1 Comparing the Different Schemes

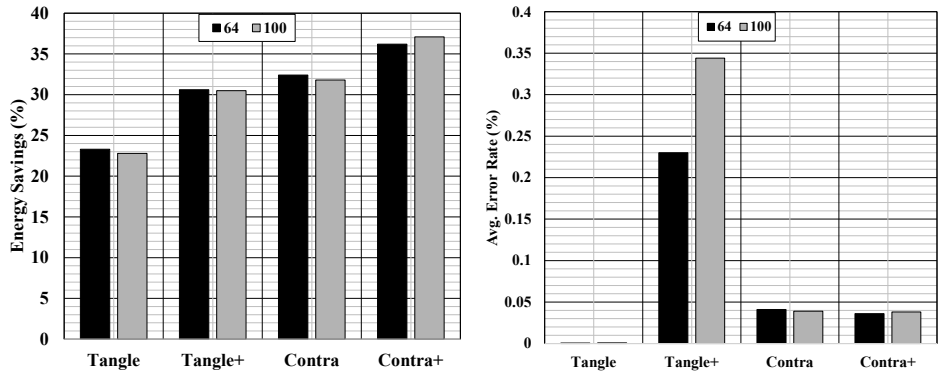
We start the evaluation by comparing the schemes in Table 5.1. Figure 6.1 compares them under four metrics: energy savings (a), average error rate (b), performance overhead (c), and maximum error rate (d). The figure shows data for 64- and 100-router NoCs. Next, we consider each metric in turn.

6.1.1 Comparing Energy Savings:

Figure 6.1(a) shows the NoC energy savings relative to an NoC without V_{dd} reduction (Baseline). The data corresponds to the average of all the applications. We see that Contra+ saves 36-37% of the NoC energy, while Contra saves 31-32%. Tangle+ saves about as much energy (30%), and Tangle only saves 23%.

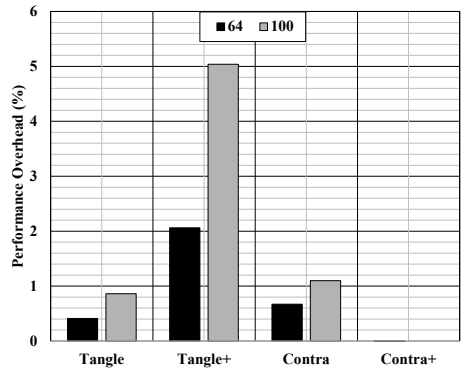
Contra+ saves the most because it uses its advanced hardware (including the secondary network) to control V_{dd} well, and keep it the lowest. Contra also has a formal controller, which helps it reduce the V_{dd}. However, on an error, the message is retransmitted on the same network. Since the V_{dd} of the routers is not changed until the beginning of the next epoch, the retransmitted message may suffer the same error again. As a result, the error rate observed by Contra is slightly higher than that of Contra+, and Contra will converge to a slightly higher V_{dd} — hence, saving less energy.

Tangle follows a different approach. As soon as a router detects an error, Tangle increases the V_{dd} without waiting for the epoch to complete. Moreover, Tangle increases the V_{dd} of all the routers in the path of the flit. As a result, the error rate will be low, but the V_{dd} values are higher than the other schemes. Consequently, Tangle saves the least energy.

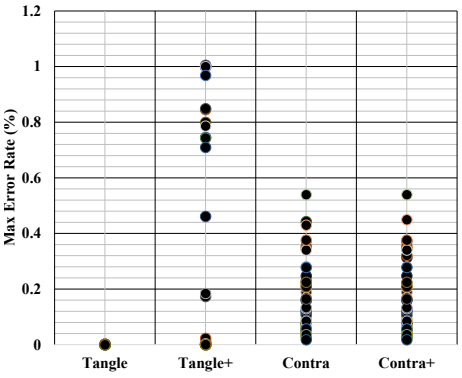


(a) Energy savings.

(b) Average error rate.



(c) Performance overhead.



(d) Maximum error rate.

Figure 6.1: Comparing the energy savings, average error rate, performance overhead, and maximum error rate of the different schemes. Energy and performance are normalized to the NoC without V_{dd} reduction (Baseline). The figure shows data for 64- and 100-router NoCs.

Finally, Tangle+ follows the same eager approach as Tangle to increase the V_{dd} of a router immediately when an error is detected. However, it only increases the V_{dd} of that router. Hence, the energy savings are higher than in Tangle.

The impact of the different schemes on V_{dd} is shown in Figure 6.2. The figure has one plot per scheme, which shows the variation of the V_{dd} of each router over time (in epochs) in a 64-router NoC running the GemsFDTD application. Each plot also shows the average V_{dd} as a thicker line.

From the plots, we see that the steady-state average V_{dd} goes up from Contra+ to Contra, Tangle+, and Tangle. This change largely explains the difference in energy savings in Figure 6.1(a). In addition, the plots show the very different convergence behavior of Contra+ and Contra on the one hand,

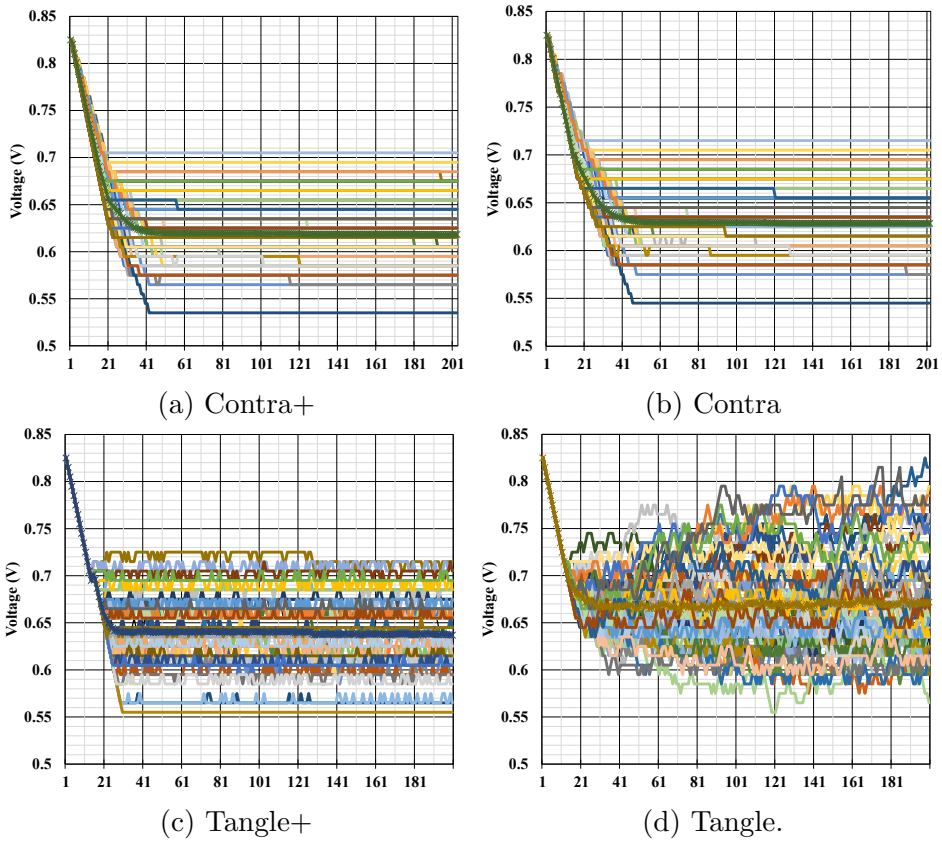


Figure 6.2: Variation in the V_{dd} of each router over time (in epochs) in a 64-router NoC running the GemsFDTD application. Note that *both voltage regulators* in the hierarchical VR contribute to the voltage reduction.

and of Tangle+ and Tangle on the other. Specifically, in the former, the V_{dd} of each router converges smoothly. This is thanks to the formal controller. However, in Tangle+ and Tangle, the V_{dd} of each router keeps oscillating. This produces unoptimal V_{dd} values — sometimes too high and therefore wasteful, and sometimes too low and so error-inducing.

6.1.2 Comparing the Average Error Rate:

Figure 6.1(b) shows the average error rate across all the epochs, routers, and applications. There are bars for 64- and 100-router NoCs. Recall that we set the reference error rate for our PID controller to 0.05%. As expected, we see that Contra+ and Contra have an average error rate lower than the reference one.

Tangle has a very low error rate. The reason is that Tangle is a very conservative scheme. Indeed, as soon as a router detects an error, the V_{dd} of all the routers in the flit path is immediately increased. On the other hand, Tangle+ suffers a much higher average error rate: 0.23% and 0.34% for 64- and 100-router NoCs, respectively. The reason is that, on an error, only one router increases its V_{dd}. Therefore, a flit traversing a low-V_{dd} path may suffer errors in each router. What is worse, given the ad-hoc nature of Tangle+, its average error rate is not guaranteed to be below any particular value, and can be quite high.

6.1.3 Comparing Performance Overhead:

Figure 6.1(c) shows the average overhead of program execution relative to execution on a machine with the Baseline NoC. There are bars for 64- and 100-router NoCs. We can see that Contra+ has no visible performance overhead. It helps that, on an error, the packet is retransmitted on the unloaded secondary network.

Contra has a performance overhead of 1% or less. The overhead is due to the re-transmission — over the main network — of packets following errors. These packets add network congestion and more delay to the sender.

Tangle is a very conservative scheme that suffers very few errors (Figure 6.1(b)). As a result, it only has a small overhead. The main reason for the overhead is that routers are unavailable during V_{dd} changes.

Finally, Tangle+ has a 2-5% performance overhead. This high overhead results from the recovery from its relatively frequent errors (Figure 6.1(b)). These errors were the result of its aggressive yet ad-hoc nature. Consequently, Tangle+ is not attractive.

6.1.4 Comparing the Maximum Error Rate:

Finally, Figure 6.1(d) shows the maximum error rate observed by each of the routers in a 64-router NoC. For each scheme, we have 64 data points. The data corresponds to running the GemsFDTD application.

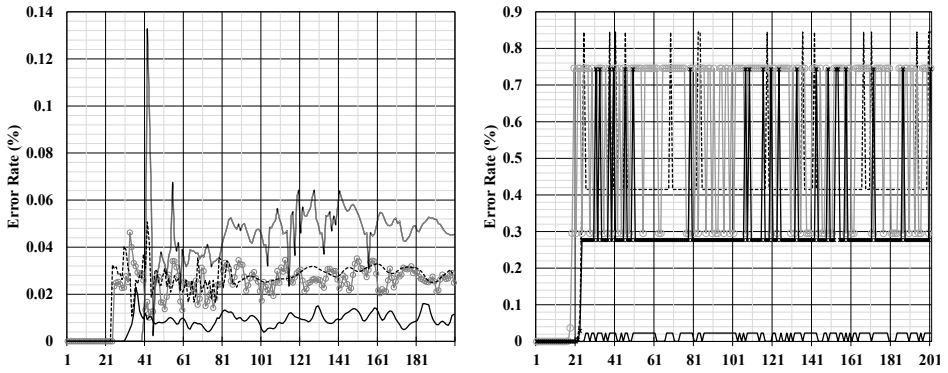
For Contra+ and Contra, these data points correspond to the first error overshoot. After this, the error rate decreases and very soon converges to

the target value. As seen in the figure, Contra+ and Contra have at least one router that reaches a 0.55% error rate.

Recall that, according to Section 5, we designed the Contra controller for a maximum error rate of 0.1%. However, Figure 6.1(d) shows that some routers reach 0.55%. This is because the controller is designed for the average conditions of the 64 routers. Due to variation, routers deviate from average conditions and, therefore, some see higher error rates in their first overshoot. To ensure that we are always below the 0.1% error rate, we can provide a per-router controller. Alternatively, we can reduce the size of the minimum V_{dd} change step, currently set to a conservative 10 mV. We try the latter in Section 6.2.3.

As shown in Figure 6.1(d), Tangle has a very small maximum error rate, but the opposite is true for Tangle+. First, Tangle+ has routers that reach 1% error rate. In addition, for Tangle+, the whole execution of the program sees error-rate peaks of magnitude comparable to 1%; Tangle+'s ad-hoc nature means that there is no error-rate convergence.

To see this effect, consider Figure 6.3, which shows how the error rate varies over time (in epochs) for Contra (Chart (a)) and Tangle+ (Chart (b)). The figure corresponds to the execution of the GemsFDTD application. It shows the error rate of four routers: one with few errors, two with a medium number of errors, and one with many errors. Note that the Y axes of the two plots use numbers that are one order of magnitude apart.



(a) Contra (b) Tangle+.

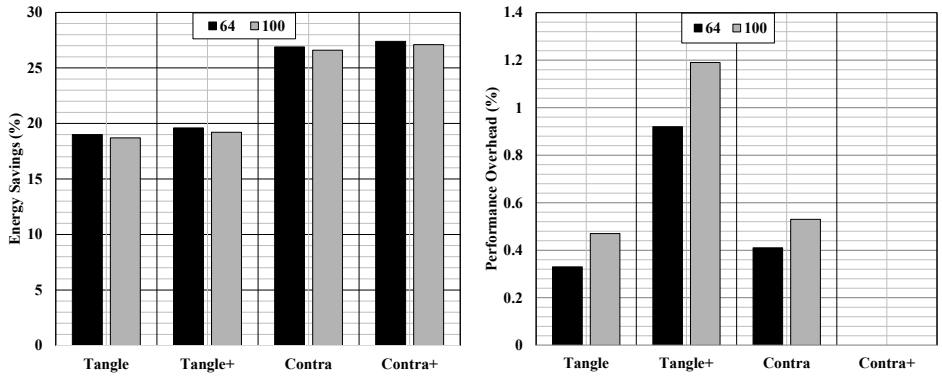
Figure 6.3: Variation of error rate over time (in epochs) in four routers.

From Figure 6.3(b), we see that the routers in Tangle+ keep the same error profile as time goes on. They do not converge. In Figure 6.3(a), however,

we see that the routers in Contra have an initially high error rate and then converge to lower values.

6.1.5 Larger V_{dd} Domains:

Figure 6.4 repeats the energy and performance plots of Figure 6.1 for NoCs with V_{dd} domains that cover 8 routers each. We can see that Contra+ and Contra save more energy than Tangle, but the reductions are smaller than with per-router V_{dd} domains. The reason is that the larger domains become less homogeneous in their variation parameters, and setting a single V_{dd} for the whole domain is less beneficial. Overall, Contra+ and Contra save 9% and 8% more NoC energy, respectively, than Tangle. In addition, their performance overhead is negligible. On the other hand, Tangle+ does not save much energy beyond Tangle and its performance overhead is about 1%.



(a) Energy savings (b) Performance overhead

Figure 6.4: Comparing the energy savings and performance overhead of the different schemes with 8-router V_{dd} domains.

6.1.6 Summary:

Overall, Contra+ has the best scores in all dimensions. It has no performance overhead and, compared to the only previously-proposed scheme (Tangle), it reduces the NoC energy by an additional 13% for single-router V_{dd} domains (or by 9% for 8-router domains). Its effectiveness is in part due to the secondary network, which highly-efficiently retransmits the packets that failed.

A cheaper and, to our opinion, more cost-effective alternative is Contra, which does not have the secondary network. Compared to Tangle, Contra has the same minimal performance overhead and reduces the NoC energy by an additional 9% for single-router V_{dd} domains (or by 8% for 8-router domains). Importantly, both Contra+ and Contra have the same smooth convergence of router V_{dd} and error rates, as enabled by a formal controller.

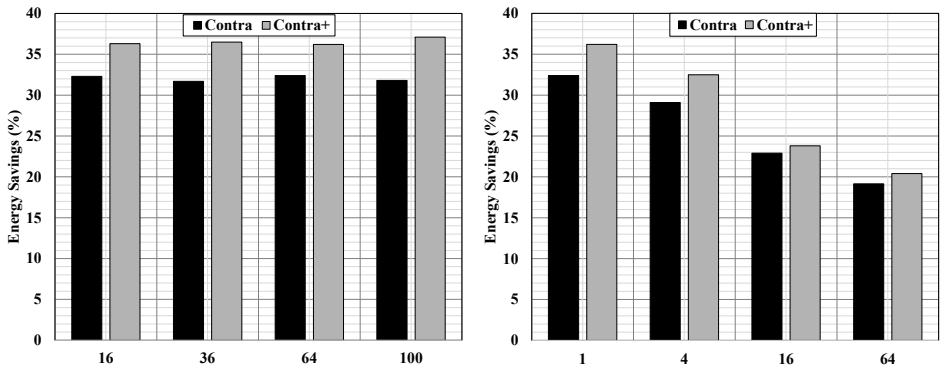
Tangle+ is not an attractive scheme. While, for single-router domains, it saves nearly as much energy as Contra, it slows down execution by 2-5%. In addition, even in steady state, it suffers large oscillations in error rate and, to a lesser degree, V_{dd}. Moreover, it provides no guarantees on error-rate limits.

6.2 Contra/Contra+ Characterization

We now examine some aspects of the Contra/Contra+ designs.

6.2.1 Energy Savings for Different NoC Sizes:

The careful design of the controller in Contra and Contra+ enables these architectures to retain the energy savings (32% and 37%, respectively) across NoC sizes, as shown in Figure 6.5a. Additionally, due to its advanced design, Contra+ is able to save more energy than Contra for all network sizes.



(a) Across NoC sizes (b) Across V_{dd} domain sizes

Figure 6.5: Energy Savings across NoC sizes and available V_{dd} domains in Contra and Contra+

6.2.2 Energy Savings for Different V_{dd} Domain Sizes:

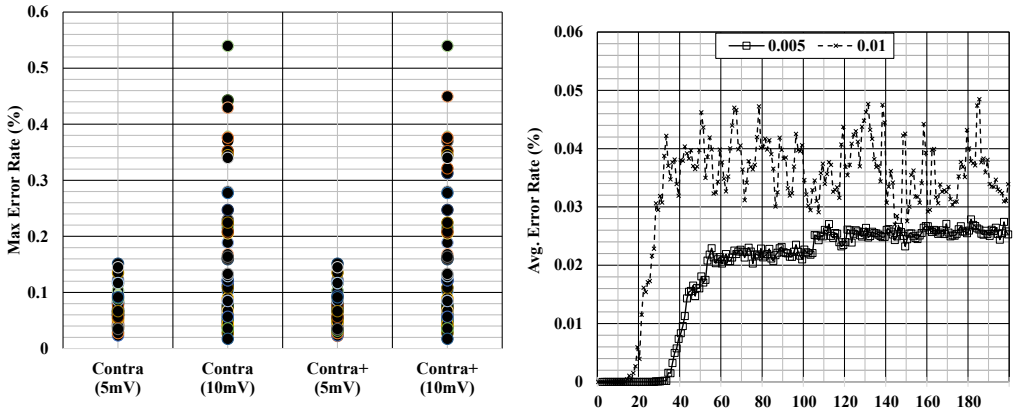
As mentioned earlier, the energy savings is expected to reduce with increasing size of the V_{dd} domains in the chip. With larger V_{dd} domains, the V_{dd} of an entire domain is set conservatively to the maximum of the voltages of all the routers in the domain. Hence, the savings decrease as shown in Figure 6.5b. From this figure it can also be seen that, even in the case of a single V_{dd} domain for the entire chip (i.e. size of 64), Contra and Contra+ are able to save 19-20% of the NoC energy.

6.2.3 Reducing the Minimum V_{dd} Step Size:

The output of the controller in Contra and Contra+ is limited by the minimum V_{dd} step that it is allowed to generate. As the minimum voltage step becomes smaller, two observable metrics improve. First, the maximum error rate (the peak overshoot) is lower with a smaller step, because the controller can finely adjust the voltages. Second, the variation of error rate with time is smoother, nearly eliminating a majority of sharp transients. The first result is plotted in Figure 6.6a which shows the maximum error rate coming down to 0.16% from 0.6% when a minimum step of $5mV$ is used. The second improvement is shown in Figure 6.6b that demonstrates the difference in the behavior of average error rate with time, for $5mV$ and $10mV$ steps. Smaller steps are expected to become available with upcoming circuits and future technologies.

6.2.4 Secondary Network Utilization in Contra+:

Contra+ includes an additional network in its design to achieve higher energy savings and further reduce the performance overheads of Contra. The novel design of this network along with the tight control over error rates keeps the utilization low. Even for a 100-router NoC, the utilization at the root layer does not go beyond 10% for any application, as shown in Figure 6.7. For smaller NoC sizes, the utilization is much smaller.



(a) Max Error Rate

(b)

Figure 6.6: Impact of minimum V_{dd} step on Contra and Contra+

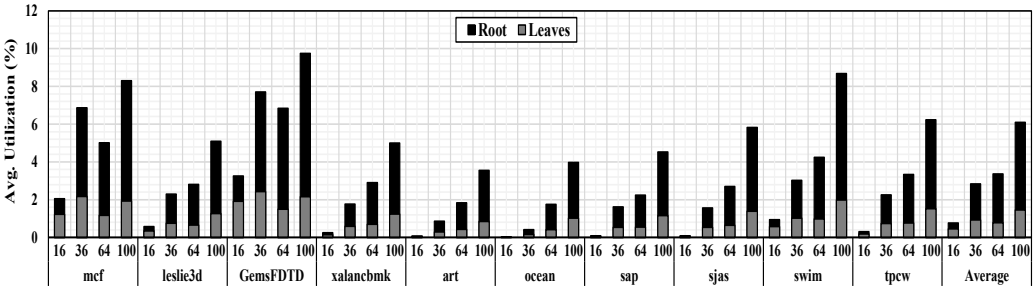


Figure 6.7: Utilization of the highest and the lowest (complete) levels of the secondary network for different NoC sizes.

CHAPTER 7

CONTRA/CONTRA+ DESIGN SPACE EXPLORATION

Contra and Contra+ contain several configurable parameters that can be adjusted by the upper hardware and software layers to suit the requirements of the operating scenario. We present the results of an exploration on the impact of a few such parameters.

7.1 Error-Rate Threshold for Activating Controller:

As described in Section 3.1.1, the controller is activated for the first time only after the error rate reaches a threshold. From that instant, the controller steers the voltages of the routers to their stable values (as dictated by the reference error rate). A very low threshold would begin the controller in the non-linear region, reducing the rate at which the controller changes the voltages. A higher threshold error rate delays the activation of the controller and it may not have sufficient history to build a strong integral gain, G_I. This may deter the system from smoothly converging to the steady state, resulting in higher error rates and longer convergence times. This is shown in Figure 7.1. For an unreasonable threshold of 0.1% (greater than the reference error rate of 0.05%) the magnitude and variation of error rates is high as the controller starts late.

7.2 Reference Error-Rate:

Reference error rate ($E_\circ$) determines the steady state error rate in Contra. A higher value for $E_\circ$would lead to more errors and longer convergence times. Figure 7.2 illustrates this behavior for different values of $E_\circ$. The steady state voltages are only slightly lower (by at most 1%) even for higher steady state error rates, due to the high sensitivity of error rate to V_{dd}. While $E_\circ$

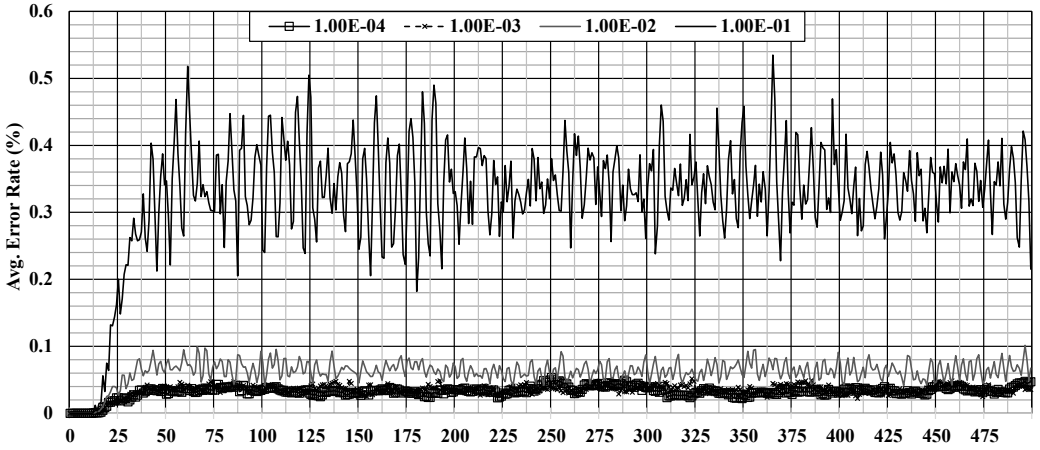


Figure 7.1: Avg. error rate for different controller activation thresholds.

presents an opportunity to accrue more energy savings for an acceptable performance overhead, the additional energy savings due to a slightly higher $E_\circ$ may not be attractive enough due to the marginally lower voltages. As there are diminishing returns for $E_\circ$values $\leq 0.05\%$, we use this value in our evaluations.

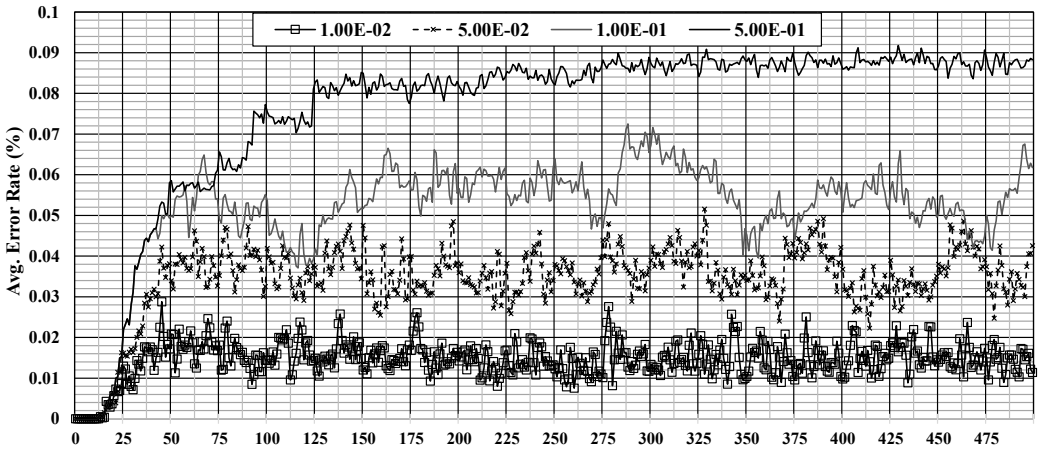


Figure 7.2: Avg. error rate for different reference error rates.

7.3 Link Width of the Secondary Network:

The width of the links in the secondary network for Contra+ plays an important role in determining the network's power consumption and utilization. Wider links increase static power consumption while decreasing the buffer

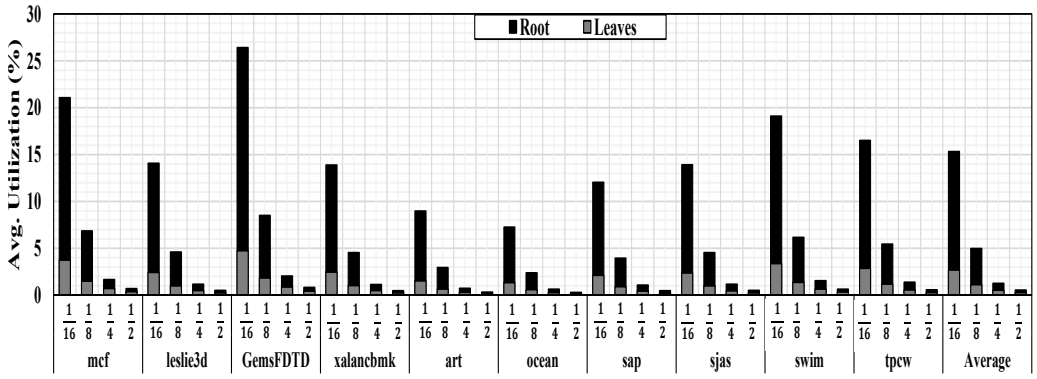


Figure 7.3: Utilization of the highest and the lowest (complete) levels of the secondary network for different link widths.

utilization in the network. The converse is true for thinner links. We found that the average utilization of the secondary network at the root node grows slowly and stays below 10% for link widths up to one-eighth of the original network link width. For thinner links, the average utilization at the root node could go up to 26% for some applications. These results are shown in Figure 7.3

7.4 Fat-Tree Bandwidth:

The amount of traffic seen by a node in a tree network increases, as it moves closer to the root node. A fat-tree alleviates this problem by appropriately widening the channels at the higher nodes. In this regard, we explored three options for the secondary network in Contra+;doubling the link width every level, (A), doubling the link width at every other level (Alt) ,and never change. (N). As evident from Figure 7.4, never changing the link widths could increase the average utilization at the root node up to 35%. Doubling the link widths at every layer leads to an average utilization of less than 1% while wasting area and energy. For our evaluation, We picked the intermediate option (double every two levels) as it is a balanced option.

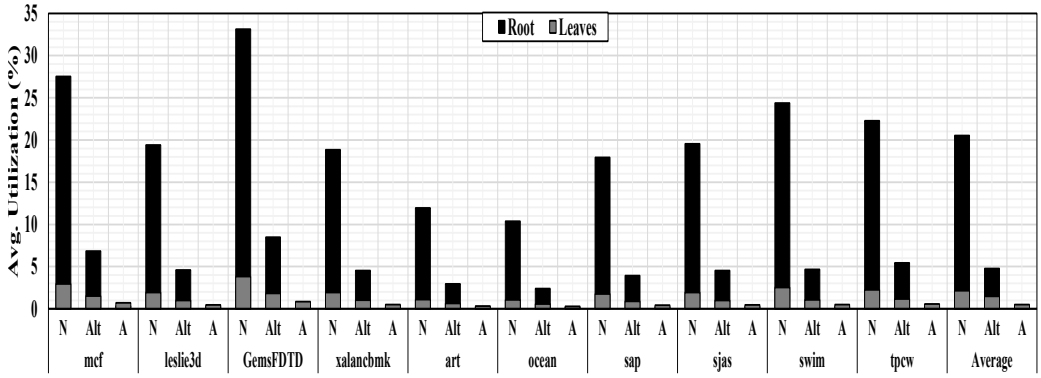


Figure 7.4: Utilization of the highest and the lowest (complete) levels of the secondary network for different fat-tree bandwidth provisioning schemes

CHAPTER 8

RELATED WORK

The impact of process variation on NoCs has been studied by Nicopoulos *et al.* using rigorous circuit analysis [46]. Their analysis does not include a fault model and the impact of faults at the system level. In [9], Li *et al.* show that a high degree of process variation can force major design modifications to the underlying network architecture. Ogras *et al.* explored the effectiveness of multiple V_{dd}-frequency domains in NoCs when dealing with a deep sub-micron process [47]. In [48], Lefurgy *et al.* use a non-formal feedback regulator to reduce the V_{dd} guardbands by continuously monitoring the available timing margin. This approach is conservative but always safe, requires continuous monitoring of the pipeline, and responds immediately on an event as opposed to periodic changes.

Approaches to compensate for the timing variations of links by time borrowing or cycle tuning have been proposed in [49, 50]. Time borrowing or stealing is a technique that has been used to tackle process variations in the processor pipeline [51, 52, 53]. Such techniques are mostly applied statically, at manufacturing time and require circuit-level modification to the underlying router design. This limits these techniques in adapting to the runtime conditions of the system. Additionally, for high degrees of process variation, these proposals require changes to the timing characteristics of the circuit.

Many techniques have been proposed which look at the fault tolerance of both links and routers in the presence of permanent and transient faults [54, 55, 39]. These solutions may incur unnecessarily high overheads as they are proactive, and tackle all types of faults in the same way. Instead, our goal is to adjust the circuit parameters to address timing failures.

The application of control theory to save power was explored by Wu *et al.* [27] in multiple-clock domain processors, and by Raghavendra *et al.* [56] in data centers. We have used several of their insights. However, we focus on a different domain, namely NoC routers. Also, unlike their approaches,

we do not assume multiple frequency domains, since the frequency domains necessitate the use of asynchronous queues to communicate across domains.

CHAPTER 9

CONCLUSION

This thesis presented Contra, a scheme that dynamically minimizes the V_{dd} of groups of routers in a variation-prone NoC at constant frequency using formal control-theory methods. Moreover, we enhanced Contra with a low-cost secondary network that retransmits erroneous packets for higher energy efficiency. The enhanced scheme is called Contra+.

We evaluated Contra and Contra+ with simulations of NoCs with 64–100 routers, and compared them to two instantiations of the best known prior work. In an NoC with 8 routers per V_{dd} domain, our schemes reduced the average energy consumption of the NoC by 27%; in a futuristic NoC with one router per V_{dd} domain, Contra+ and Contra reduced the average energy consumption of the NoC by 37% and 32%, respectively. The performance impact was negligible. These savings, which already include the penalty of power losses in voltage regulators, are 8–13 percentage points higher than the best existing ad hoc approaches. Moreover, we showed that formal control is essential to provide guarantees on the convergence, stability, and maximum resulting error rates. Finally, we concluded that while the secondary network helps Contra+ attain higher energy savings, it has a non-negligible hardware cost. Hence, Contra is the most cost-effective design.

REFERENCES

[1] M. Horowitz, E. Alon, D. Patil, S. Naffziger, R. Kumar, and K. Bernstein, "Scaling, Power, and the Future of CMOS," in *Electron Devices Meeting, 2005. IEDM Technical Digest. IEEE International*, December 2005.

[2] T. N. Mudge, "Power: A First-Class Architectural Design Constraint," *IEEE Computer*, vol. 34, no. 4, pp. 52–58, 2001.

[3] S. Y. Borkar, "Designing Reliable Systems from Unreliable Components: The Challenges of Transistor Variability and Degradation," *IEEE Micro*, 2005.

[4] S. Y. Borkar, "Future of interconnect fabric: a contrarian view," in *International Workshop on System Level Interconnect Prediction*, 2010.

[5] N. Carter, A. Agrawal, S. Borkar, R. Cledat, H. David, D. Dunning, J. Fryman, I. Ganev, R. Golliver, R. Knauerhase, R. Lethin, B. Meister, A. Mishra, W. Pinfold, J. Teller, J. Torrellas, N. Vasilache, G. Venkatesh, and J. Xu, "Runnemede: An Architecture for Ubiquitous High-Performance Computing," in *International Symposium on High Performance Computer Architecture*, Feb. 2013.

[6] S. Dighe, S. Vangal, P. Aseron, S. Kumar, T. Jacob, K. Bowman, J. Howard, J. Tschanz, V. Erraguntla, N. Borkar, V. De, and S. Borkar, "Within-Die Variation-Aware Dynamic-Voltage-Frequency-Scaling With Optimal Core Allocation and Thread Hopping for the 80-Core TeraFLOPS Processor," *J. Solid-State Circuits*, vol. 46, no. 1, pp. 184–193, 2011.

[7] X. Fu, T. Li, and J. A. B. Fortes, "Architecting reliable multi-core network-on-chip for small scale processing technology," in *Dependable Systems and Networks*, 2010, pp. 111–120.

[8] J. Howard, S. Dighe, S. Vangal, G. Ruhl, N. Borkar, S. Jain, V. Erraguntla, M. Konow, M. Riepen, M. Gries, G. Droege, T. Lund-Larsen, S. Steibl, S. Borkar, V. De, and R. Van Der Wijngaart, "A 48-core ia-32 processor in 45 nm CMOS using on-die message-passing and dvfs for performance and power scaling," *Journal of Solid-State Circuits*, vol. 46, no. 1, pp. 173 –183, 2011.

[9] B. Li, L.-S. Peh, and P. Patra, "Impact of Process and Temperature Variations on Network-on-Chip Design Exploration," in *NOCS*, 2008.

[10] S. Hemmert, "From Petascale to Exascale: R & D Challenges for HPC Simulation Environments," March 2011. [Online]. Available: https://asc.llnl.gov/content/assets/docs/exascale-hwaWG.pdf

[11] D. Ernst, N. S. Kim, S. Das, S. Pant, R. Rao, T. Pham, C. Zeisler, D. Blaauw, T. Austin, K. Flautner, and T. Mudge, "Razor: A low-power pipeline based on circuit-level timing speculation," in *International Symposium on Microarchitecture*, Dec. 2003.

[12] B. Greskamp, L. Wan, U. R. Karpuzcu, J. J. Cook, J. Torrellas, D. Chen, and C. B. Zilles, "Blueshift: Designing processors for timing speculation from the ground up," in *International Symposium on High Performance Computer Architecture*, 2009.

[13] A. Ansari, A. Mishra, J. Xu, and J. Torrellas, "Route-oriented dynamic voltage minimization for variation-afflicted, energy-efficient on-chip networks," in *International Symposium on High Performance Computer Architecture*, 2014.

[14] X. Liang, R. Canal, G.-Y. Wei, and D. Brooks, "Replacing 6T SRAMs with 3T1D DRAMs in the L1 Data Cache to Combat Process Variability," *IEEE Micro*, vol. 28, no. 1, pp. 60–68, Jan. 2008.

[15] S. Sarangi, B. Greskamp, R. Teodorescu, J. Nakano, A. Tiwari, and J. Torrellas, "VARIUS: A model of process variation and resulting timing errors for microarchitects," *Transactions on Semiconductor Manufacturing*, no. 1, pp. 3–13, Feb. 2008.

[16] C. Wilkerson, A. R. Alameldeen, Z. Chishti, W. Wu, D. Somasekhar, and S.-L. Lu, "Reducing cache power with low-cost, multi-bit error-correcting codes," in *ISCA*, 2010, pp. 83–93.

[17] U. R. Karpuzcu, u. B. Kollur, N. S. Kim, and J. Torrellas, "VARIUS-NTV: A Microarchitectural Model to Capture the Increased Sensitivity of Manycores to Process Variations at Near-Threshold Voltages," in *International Conference on Dependable Systems and Networks*, 2012.

[18] L. Chang, R. Montoye, B. Ji, A. Weger, K. Stawiasz, and R. Dennard, "A Fully-Integrated Switched-Capacitor 2:1 Voltage Converter with Regulation Capability and 90% Efficiency at 2.3A/mm2," in *IEEE Symposium on VLSI Circuits*, June 2010.

[19] "Intel® Xeon® Processor E3-1200 v3 Product Family Datasheet," June 2013, http://www.intel.com/content/dam/www/public/us/en/documents/datasheets/xeon-e3-1200v3-vol-1-datasheet.pdf.

[20] H. R. Ghasemi, A. Sinkar, M. Schulte, and N. S. Kim, "Cost-Effective Power Delivery to Support Per-Core Voltage Domains for Power-Constrained Processors," in *Design Automation Conference*, June 2012.

[21] F. Ishihara, F. Sheikh, and B. Nikolic, "Level Conversion for Dual-Supply Systems," in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Feb. 2004.

[22] G. C. Goodwin, S. F. Graebe, and M. E. Salgado, *Control System Design*, 1st ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2000.

[23] E. W. Weisstein, ""Erf." From MathWorld–A Wolfram Web Resource." http://mathworld.wolfram.com/Erf.html.

[24] E. W. Weisstein, ""Gompertz Curve." From MathWorld–A Wolfram Web Resource." http://mathworld.wolfram.com/GompertzCurve.html.

[25] E. W. Weisstein, ""Logistic Distribution." From MathWorld–A Wolfram Web Resource." http://mathworld.wolfram.com/LogisticDistribution.html.

[26] E. W. Weisstein, ""Sigmoid Function." From MathWorld–A Wolfram Web Resource." http://mathworld.wolfram.com/SigmoidFunction.html.

[27] Q. Wu, P. Juang, M. Martonosi, and D. W. Clark, "Formal online methods for voltage/frequency control in multiple clock domain microprocessors," in *International conference on Architectural Support for Programming Languages and Operating Systems*, 2004. [Online]. Available: http://doi.acm.org/10.1145/1024393.1024423

[28] C. E. Leiserson, "Fat-trees: universal networks for hardware-efficient supercomputing," *IEEE Transactions on Computers*, vol. 34, no. 10, pp. 892–901, Oct. 1985. [Online]. Available: http://dl.acm.org/citation.cfm?id=4492.4495

[29] D. Ludovici, F. Gilabert, S. Medardoni, C. Gomez, M. Gomez, P. Lopez, G. N. Gaydadjiev, and D. Bertozzi, "Assessing fat-tree topologies for regular network-on-chip design under nanoscale technology constraints," in *Design, Automation Test in Europe Conference Exhibition, 2009. DATE '09.*, 2009, pp. 562–565.

[30] H. Matsutani, M. Koibuchi, and H. Amano, "Performance, Cost, and Energy Evaluation of Fat H-Tree: A Cost-Efficient Tree-Based On-Chip Network," in *International Parallel and Distributed Processing Symposium*, March 2007, pp. 1–10.

[31] H. Matsutani, M. Koibuchi, Y. Yamada, D. Hsu, and H. Amano, "Fat H-Tree: A Cost-Efficient Tree-Based On-Chip Network," *IEEE Transactions on Parallel and Distributed Systems*, vol. 20, no. 8, pp. 1126–1141, Aug 2009.

[32] Z. Wang, J. Xu, X. Wu, Y. Ye, W. Zhang, M. Nikdast, X. Wang, and Z. Wang, "Floorplan optimization of fat-tree based networks-on-chip for chip multiprocessors," *IEEE Transactions on Computers*, vol. 99, no. PrePrints, p. 1, 2012.

[33] G. M. Adel'son-Vel'sky and Y. M. Landis, "An Algorithm for the Organization of Information," *Doklady Akademii Nauk USSR*, vol. 146, no. 2, pp. 263–266, 1962, english translation in Soviet Mathematics Doklady, vol. 3, pp. 1259-1263.

[34] P. Koopman and T. Chakravarty, "Cyclic Redundancy Code (CRC) Polynomial Selection For Embedded Networks." in *Dependable Systems and Networks*, 2004.

[35] D. Lenoski, J. Laudon, K. Gharachorloo, A. Gupta, and J. Hennessy, "The directory-based cache coherence protocol for the DASH multiprocessor," *SIGARCH Comput. Archit. News*, vol. 18, no. 3a, pp. 148–159, May 1990.

[36] A. K. Mishra, N. Vijaykrishnan, and C. R. Das, "A case for heterogeneous on-chip interconnects for CMPs," in *International Symposium on Computer Architecture*, 2011.

[37] K. Bhardwaj, K. Chakraborty, and S. Roy, "Towards graceful Aging Degradation in NoCs Through an Adaptive Routing Algorithm," in *Design Automation Conference*, June 2012.

[38] L.-S. Peh and W. J. Dally, "A Delay Model and Speculative Architecture for Pipelined Routers," in *High Performance Computer Architecture*, 2001.

[39] D. Park, C. Nicopoulos, J. Kim, N. Vijaykrishnan, and C. R. Das, "Exploring Fault-Tolerant Network-on-Chip Architectures," ser. DSN '06, 2006, pp. 93–104.

[40] A. Kumar, P. Kundu, A. P. Singh, L.-S. Peh, and N. K. Jha, "A 4.6Tbits/s 3.6GHz single-cycle NoC router with a novel switch allocator in 65nm CMOS," in *ICCD*, 2007, pp. 63–70.

[41] S. Borkar, "Design Challenges of Technology Scaling," *IEEE Micro*, Jul. 1999.

[42] International Technology Roadmap for Semiconductors (ITRS),, *2012 Update*.

[43] S. Jain, S. Khare, S. Yada, V. Ambili, P. Salihundam, S. Ramani, S. Muthukumar, M. Srinivasan, A. Kumar, S. Kumar, R. Ramanarayanan, V. Erraguntla, J. Howard, S. R. Vangal, S. Dighe, G. Ruhl, P. A. Aseron, H. Wilson, N. Borkar, V. De, and S. Borkar, "A 280mV-to-1.2V wide-operating-range IA-32 processor in 32nm CMOS." in *International Solid-State Circuits Conference*, 2012.

[44] C. Sun, C.-H. Chen, G. Kurian, L. Wei, J. Miller, A. Agarwal, L.-S. Peh, and V. Stojanovic, "DSENT - A Tool Connecting Emerging Photonics with Electronics for Opto-Electronic Networks-on-Chip Modeling," in *International Symposium on Networks on Chip*, 2012.

[45] T. Sherwood, E. Perelman, G. Hamerly, and B. Calder, "Automatically characterizing large scale program behavior," *SIGARCH Comput. Archit. News*, vol. 30, no. 5, pp. 45–57, Oct. 2002.

[46] C. Nicopoulos, S. Srinivasan, A. Yanamandra, D. Park, V. Narayanan, C. R. Das, and M. J. Irwin, "On the Effects of Process Variation in Network-on-Chip Architectures," *IEEE Transactions on Dependable and Secure Computing*, vol. 7, no. 3, pp. 240–254, July 2010.

[47] Ü. Y. Ogras, R. Marculescu, and D. Marculescu, "Variation-adaptive feedback control for networks-on-chip with multiple clock domains," in *Design Autoamtion Conference*, 2008.

[48] C. R. Lefurgy, A. J. Drake, M. S. Floyd, M. S. Allen-Ware, B. Brock, J. A. Tierno, and J. B. Carter, "Active management of timing guardband to save energy in power7," in *International Symposium on Microarchitecture*, 2011.

[49] A. K. Mishra, R. Das, S. Eachempati, R. R. Iyer, N. Vijaykrishnan, and C. R. Das, "A case for dynamic frequency tuning in on-chip networks," in *MICRO*, 2009, pp. 292–303.

[50] M. Simone, M. Lajolo, and D. Bertozzi, "Variation tolerant NoC design by means of self-calibrating links," ser. DATE, 2008.

[51] X. Liang and D. Brooks, "Mitigating the impact of process variations on processor register files and execution units," in *MICRO*, 2006, pp. 504–514.

[52] X. Liang, G.-Y. Wei, and D. Brooks, "Revival: A Variation-Tolerant Architecture Using Voltage Interpolation and Variable Latency," *IEEE Micro*, vol. 29, no. 1, pp. 127–138, Jan. 2009.

[53] A. Tiwari, S. R. Sarangi, and J. Torrellas, "ReCycle: pipeline adaptation to tolerate process variation," ser. ISCA, 2007.

[54] D. Bertozzi, L. Benini, and G. de Micheli, "Low Power Error Resilient Encoding for On-Chip Data Buses," in *Conference on Design, Automation and Test in Europe*, 2002.

[55] T. Dumitras, S. Kerner, and R. Marculescu, "Towards on-chip fault-tolerant communication," in *Asia and South Pacific Design Automation Conference*, 2003.

[56] R. Raghavendra, P. Ranganathan, V. Talwar, Z. Wang, and X. Zhu, "No "power" struggles: coordinated multi-level power management for the data center," in *International Conference on Architectural support for programming languages and operating systems*, 2008.