INTERNATIONAL JOURNAL OF CIRCUIT THEORY AND APPLICATIONS
Int. J. Circ. Theor. Appl. 2011; **39**:557–572
Published online 29 April 2010 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/cta.662

Improving Networks-on-Chip performability: A topology-based approach

Haytham Elmiligi[1,*,†], Ahmed A. Morgan[1], M. Watheq El-Kharashi[2] and Fayez Gebali[1]

[1] *Department of Electrical and Computer Engineering, University of Victoria, Victoria BC, Canada V8W 3P6*
[2] *Department of Computer and Systems Engineering, Ain Shams University, Cairo 11517, Egypt*

SUMMARY

The performability metric is commonly used in Networks-on-Chip (NoC)-based systems to represent their abilities to successfully complete specific tasks in finite time intervals. In this paper, we present a novel topology-based performability model for NoC-based systems. The model is used to evaluate the performability of NoC-based systems at early design phases. A comparative study of nine commonly used network architectures is performed using the proposed model. The purpose of the study is to explore the impact of the network topology on the performability of NoC-based systems. Using the output from this study, a new methodology is proposed to improve the performability of a given application at early design phases. In this methodology, a joint consideration of five design parameters (network topology, target application traffic distribution, mapping of processing elements, noise power, and voltage swing) is carried out. Using the proposed methodology, designers can select the optimal topology for a given application that maximizes system performability. The effectiveness of the proposed methodology in determining the optimal topology is verified by experimental work and validated through a case study of a video application. Copyright © 2010 John Wiley & Sons, Ltd.

Received 16 February 2009; Revised 25 August 2009; Accepted 6 December 2009

KEY WORDS: Networks-on-Chip; topology-based design; performability model; systems-on-chip

1. INTRODUCTION

The ever growing complexity of single-die multiprocessor systems posses many challenges to the deep sub-micron fabrication process. One of these challenges is to find a solution for the interconnection complexity problem. In the last few years, the Networks-on-chip (NoC) paradigm appeared as a promising solution for the inter-communication of Systems-on-Chip (SoC) [1] and complex designs such as, cellular nonlinear network (CNN) applications [2]. The migration from a bus-based design to a network-based one opens the door for developing new techniques to improve systems' performance and reliability, and hence, improve their performability. The concept of system performability was primarily introduced by J. F. Meyer in 1978 in his evaluation of an aircraft control system [3]. He defined performability as a composite measure of performance and dependability. Dependability is a marginal term which includes various metrics such as reliability, availability, and security. Since that time, performability analysis is used, in most cases, to measure the performance and reliability in a composite way. This measurement is proven to be essential to properly assess the effectiveness of complex multiprocessor systems [4]. This is

[*] Correspondence to: Haytham Elmiligi, Department of Electrical and Computer Engineering, University of Victoria, Victoria BC, Canada V8W 3P6.
[†] E-mail: haytham@ece.uvic.ca

done by evaluating the ability of a system to successfully complete a specific function in a finite time interval [4]. For NoC-based systems, performability could be modeled at different levels of abstraction. Consequently, improving systems performability could be achieved at different design phases.

1.1. Design for performability

The NoC design approach depends mainly on the target application requirements and design constraints, such as low power consumption, limited silicon area, etc. Shrinking the die size for complex chip designs leads to a complex integration process, which could result in unreliable data transmission over wires. The integration complexity increases the probability of communication errors between processing elements (PEs) due to various noise sources, such as cross-talk, coupling noise, soft errors, etc [5]. Therefore, error-control schemes are used to address the reliability issue. However, using error-control schemes in NoC degrades the overall system performance due to the added hardware (silicon area) and logic operations (extra delay) [6]. Hence, a joint measure of reliability and performance must be considered through a unified measure, *performability*, to optimally choose the most efficient control scheme for a target application [6]. Among many design factors, the performability criterion of a system could be studied at different levels of abstraction. At early design phases, the network architecture must be designed efficiently to achieve a maximum system performability. However, at early design phases, the exact physical layout structure is not yet defined. Hence, designers do not have enough layout information to choose the most efficient network topology for their target application. In this paper, we address this problem by: (1) studying the impact of the network topology on system performability using graph-theoretic concepts, (2) developing a topology-based performability model for NoC-based systems and (3) proposing a new methodology to improve the performability of a given NoC application at early design phases. The proposed methodology aims at selecting the optimal topology that achieves the maximum system performability for a given application, taking into consideration the possible changes in noise power and voltage swing. Finally, the efficiency of the proposed methodology is verified through a case study of a video application.

1.2. Paper organization

This paper is organized as follows. Section 2 highlights the related wok. Section 3 discusses the architecture of various NoC topologies through graph-theoretic concepts. Then, we present our performability model for NoC-based systems in Section 4. The impact of the network topology on system performability is discussed in Section 5. Following that, we present a mathematical problem formulation and propose a methodology to improve the performability of a given application using topology-based design in Section 6. As a proof of the concept, the proposed methodology is verified through a case study in Section 7. Finally, we draw the conclusions and present ideas for possible extensions of this work in Section 8.

2. RELATED WORK

The work related to performability in the NoC literature could be classified into two main categories. The former is reliability and performance issues for NoC applications [5], whereas the latter is topological optimization for NoCs [7].

For the first category, research is focused on modeling and improving NoC-based system reliability [8], or evaluating its performance [9]. Reliability is usually achieved via fault-tolerant designs, which enables the network to survive the failure of one or more components without a disruption of its operation [10]. In NoC-based designs, recovering transient errors that occur in the communication subsystems is one of the main key challenges. As technology scales down, supply voltage decreases and interconnects become more sensitive to noise [11]. Therefore, analysis of error-control schemes for NoCs received increased attention recently [12]. For instance, Ejlali *et al.* analyzed the impact of various error-control schemes on the mutual trade-off between reliability, performance, and energy consumption when voltage swing varies [13].

DOI: 10.1002/cta

As for the second category, NoC topology optimization has been addressed from different perspectives such as reducing the communication cost, power consumption, and performance. In [7], Ahonen *et al.* developed a tool to handle the NoC optimization based on exploiting target application domain-specific features. Their optimization methodology applied network partitioning to minimize the cost of the communication over partition boundaries [7]. Power-aware topology optimization was the focus of many researchers [14, 15]. Chang *et al.* proposed a power-aware topology construction method and compared customized irregular network topologies to regular mesh and torus topologies with respect to packet latency and average energy consumption. A mixed integer linear programming (MILP) formulations for the synthesis of custom NoC architectures are proposed in [16] using an objective function to minimize the power consumption subject to given performance constraints. Also, multi-objective algorithms are used in [17] for the optimization of performance and power dissipation. Another approach proposed in [18] synthesized an architecture which is neither regular nor fully customized. The proposed communication architecture is a superposition of a few long-range links and a standard mesh network [18].

Although some of the above work have addressed the performance and reliability of NoCs, only one group addressed the performability metric in [13]. Even though, this group did not discuss the impact of the network topology design on system performability. In this paper, we fill the gap between the two mentioned research directions by analyzing the impact of the network topology architecture on system performability. We tackle this problem from a totally new, topology-based, perspective aiming at improving the NoC-based system performability at early design phases. We optimize the topological architecture of the interconnection network subject to a joint consideration of performance and reliability, or in another word, *performability*, for a given application.

3. A GRAPH-THEORETIC REPRESENTATION OF NoC TOPOLOGIES

Graph-theoretic concepts have been used in computer networks as a general approach to solve network flow optimization problems [19]. In this section, graph-theoretic concepts are adopted to study the topological architecture of NoCs. Figure 1 shows three types of regular network topologies: Mesh-based, Ring-based, and Tree-based architectures. Mesh-based architecture is represented by Mesh [20], Torus [21], and Folded Torus (Folded) [22] topologies. Ring-based architecture is represented by Ring [23], Octagon (Oct) [24], and Spidergon (Spider) [25, 26] topologies. Finally, Tree-based architecture is represented by Binary Tree (BT), Butterfly Fat Tree (BFT) [27], and SPIN [28] topologies. We address only nine regular topologies in our comparative analysis. However, the same approach could be applied for other regular and even irregular topologies.

Each one of the topologies in Figure 1 could be represented using graph theory as a graph $G=(V,E)$, where each node $v_i \in V$ represents a PE and E is a set of edges that represent the physical communication channels between PEs [29].

At the same time, the communication between PEs in NoC-based system could be represented by a traffic distribution graph (TDG). Figure 2 shows an example of a TDG and its corresponding representation in an $n\times n$ weight matrix (λ) format, n is the number of PEs. In this graph, each vertex $v_i \in V$ represents a PE. Each pair of edges e_{ij}, $e_{ij} \in E$ is associated with two weight factors, λ_{ij}, λ_{ji}, which represent the number of packets per time step being transmitted from v_i to v_j and vise versa, respectively.

Based on the above discussion, the optimization problem could be formulated as follows. Given a TDG $G=(V,E)$, it is required to find the optimum network topology ($\mathscr{H}$) that maximizes the network performability ($\mathscr{P}_{\text{total}}$), taking into consideration the following assumptions.

(1) Data communication between all PEs for a given application could be represented by a TDG.
(2) Shortest path routing is used in all target topologies.
(3) All global links (router-to-router) have the same number of wires (N_{wires}) (i.e. a fixed-bus-width system).
(4) Network interface (NI) units are embedded in the PEs.
(5) All local (router-to-PE) interconnection links have the same length.

DOI: 10.1002/cta

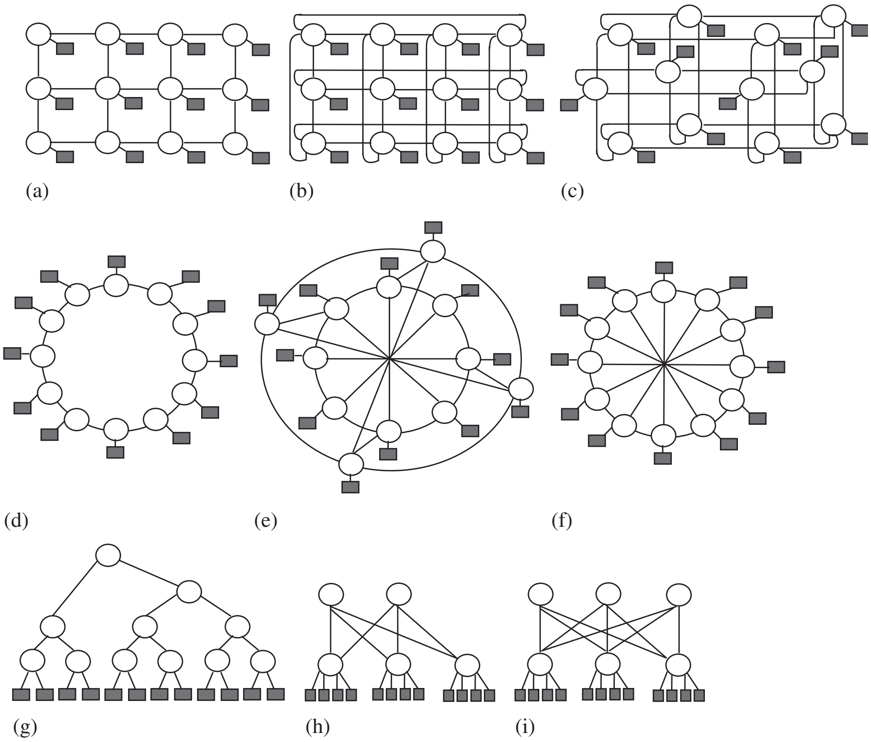

Figure 1. Nine regular NoC topologies connecting 12 PEs: (a) Mesh [20]; (b) Torus [21]; (c) Folded Torus (Folded) [22]; (d) Ring [23]; (e) Octagon (Oct) [24]; (f) Spidergon (Spider) [25, 26]; (g) Binary Tree (BT); (h) Butterfly Fat Tree (BFT) [27]; (i) SPIN [28]. Routers are represented by white circles, whereas Processing Elements (PEs) are represented by dark squares.

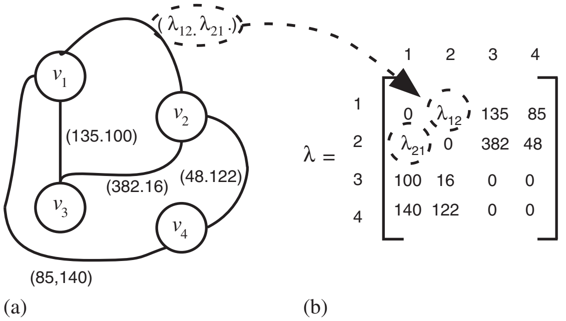

Figure 2. (a) An example of a Traffic Distribution Graph (TDG) and (b) corresponding weight matrix (λ) representation.

To solve this problem, we start with modeling the network performability. Then, a methodology is developed to acquire the optimum topology for a given application.

4. MODELING NoC PERFORMABILITY

In this section, we discuss the sources of errors in NoC-based designs and present the performability definition of a single on-chip interconnect (link), which was proposed in the literature. Then, we extend this performability definition, from a topology-based perspective, to represent the whole network.

DOI: 10.1002/cta

4.1. Sources of Errors in NoC-based Systems

The continuous scaling of technology, shrinking of transistor dimensions, and lowering supply voltage result in higher sensitivity to neutron and alpha particles, leading to significantly higher soft error rates (SER) [30]. In deep sub-micron circuits, since the capacitance associated with circuit nodes is very small, non-negligible disturbance can be originated when energized particles strike a circuit. For instance, for 3.3 V technology, the disturbance noise could reach a level of 21% larger than a normal swing. Thus, to restore the correct value of the struck node, the transistor will take more time to suppress the charge-drift process [31]. This problem becomes more critical with high speed designs, such as advanced SoC-based designs, since the noise voltage pulse may become comparable with the gate propagation delay, which might cause an erroneous transition on the output of combinational logic circuits.

Several techniques have been proposed to mitigate the impact of errors in complex VLSI designs. However, with the recent migration to NoC as an emerging communication infrastructure for complex SoC deigns, new sources of errors are considered. The impact of permanent, transient, and intermittent faults is becoming more significant due to many factors, such as crosstalk, electromagnetic interference, alpha particle hits, and cosmic radiation. These phenomena can alter the synchronization and functionality of NoC-based systems and thus degrade their Quality of Service (QoS) features and, in some cases, lead to failures for the whole system [32].

In NoC-based designs, other problems might occur such as the break-down of links and/or routers. In such cases, fault-tolerant techniques are used to provide substitute routing pathes/services to preserve the network QoS [33].

4.2. Performability of a single on-chip interconnect

The performability of a single on-chip interconnect (link) is defined as the probability to transmit a certain number of useful bits (L), which are put into λ packets through a single link during a specific time interval (T) in the presence of noise [13]. Ejlali *et al.* in [13] developed performability models of a single on-chip interconnect for four communication schemes: simple non-fault-tolerant (SNFT) communication and three error-control schemes, automatic repeat request (ARQ), forward error control (FEC), and hybrid ARQ/FEC (HARQ). In these models, the sum of several uncorrelated noise sources affecting a single on-chip interconnect is modeled as a single gaussian noise source [13, 34]. Given the packet-transmission error rate (PER), the performability (P) for schemes with retransmission capability (ARQ and HARQ) is given by [13]

$$P=\sum_{i=\lambda}^{N(V_{\mathrm{sw}})}\binom{N(V_{\mathrm{sw}})}{i}(1-\mathrm{PER})^{i}\,\mathrm{PER}^{N(V_{\mathrm{sw}})-i} \tag{1}$$

where V_{sw} is the swing voltage used to reduce the energy consumption, $N(V_{\mathrm{sw}})$ denotes the maximum number of packets which could be transmitted during T, and $0\leqslant P\leqslant 1$ [13]. The performability P for schemes that do not have retransmission capability (SNFT and FEC) is given by [13]

$$\begin{aligned} P &= 0,\ \lambda > N(V_{\mathrm{sw}}) \\ &= (1-\mathrm{PER})^{\lambda}, \quad \lambda\leqslant N(V_{\mathrm{sw}}) \end{aligned} \tag{2}$$

Although other noise models are proposed in the literature, the gaussian noise model is chosen by many researchers based on the assumption that all the noise sources collectively induce a noise voltage on the channel that follows a Gaussian distribution with zero mean and variance σ^2 [11, 13, 35].

We use this performability model for the single on-chip interconnect in this paper to represent the global (router-to-router) link performability.

4.3. Performability of a network topology

Based on the above definition for a single link performability, and assuming the shortest path routing, the network performability of a given application depends on the target topology mapping.

DOI: 10.1002/cta

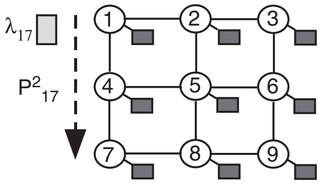

Figure 3. An example of a 3 × 3 mesh topology. Routers are represented by white circles and PEs are represented by dark squares. The gray rectangular represents a number of packets transmitted from node one to node seven (λ_{17}). P is the performability of a single on-chip interconnect and P_{17}^{2} represents the performability when λ_{17} packets are transmitted from node 1 to 7.

Therefore, a unique connectivity matrix (C) is generated for each one of the nine regular network topologies mentioned in Section 3 [36]. This matrix represents the minimum number of links a packet goes through during its transition from the source node to the destination node. Hence, it could be used to represent the architecture and the mapping of PEs for any network topology. For example, the connectivity matrix C of the mesh network topology shown in Figure 3 is written as follows:

$$C=\begin{bmatrix} 0 & 1 & 2 & 1 & 2 & 3 & 2 & 3 & 4\\ 1 & 0 & 1 & 2 & 1 & 2 & 3 & 2 & 3\\ 2 & 1 & 0 & 3 & 2 & 1 & 4 & 3 & 2\\ 1 & 2 & 3 & 0 & 1 & 2 & 1 & 2 & 3\\ 2 & 1 & 2 & 1 & 0 & 1 & 2 & 1 & 2\\ 3 & 2 & 1 & 2 & 1 & 0 & 3 & 2 & 1\\ 2 & 3 & 4 & 1 & 2 & 3 & 0 & 1 & 2\\ 3 & 2 & 3 & 2 & 1 & 2 & 1 & 0 & 1\\ 4 & 3 & 2 & 3 & 2 & 1 & 2 & 1 & 0 \end{bmatrix}$$

For a given TDG, λ_{ij} represents the number of packets being transmitted from node i to node j. The corresponding performability measure over a single link (P) is denoted by P_{ij}. Now, if λ_{ij} packets need to go through c_{ij} hops in a certain topology to reach their destination, and assuming independent packet transmission, the total performability measure is denoted by ($P_{ij}^{c_{ij}}$).

For example, if the number of packets (λ_{17}) is to be transmitted from node one to seven, as shown in Figure 3, these packets have to go through two hops to reach their destination and, hence, the performability in this example could be represented by P_{17}^{2}. $P_{ij}^{c_{ij}}$ in this case is a function of the number of transmitted packets (λ_{ij}).

From the above discussion and using Gaussian noise model to represent the interconnect bit error rate (BER) [11], the total network performability of a given application, normalized to the total number of packets transmitted, when it is mapped to a specific network topology ($\mathscr{H}$) is given by

$$\mathscr{P}_{\text{total}}=\frac{\sum_{i=1}^{n}\sum_{j=1}^{n}P_{ij}^{c_{ij}}}{\sum_{i=1}^{n}\sum_{j=1}^{n}\lambda_{ij}} \tag{3}$$

where P_{ij} is given by [13]

$$P_{ij}=\left(1-\frac{1}{\sqrt{2\pi}}\int_{\frac{V_{\text{sw}}}{2\sigma}}^{\infty}\mathrm{e}^{-\mu^2/2}\,\mathrm{d}\mu\right)^{L\lambda_{ij}} \tag{4}$$

DOI: 10.1002/cta

The normalization to the total number of packets transmitted is important to have a performability representation between 0 and 1.

Because the exact physical layout structure is not usually defined at early design phases, the connectivity matrix (C) is used in (3) based on the assumption that all links have equal BER. However, a modified connectivity matrix (C_{M}) could be used after placement and routing to reflect the exact physical layout information. The modified connectivity matrix (C_{M}) could be generated using the Hadamard product of C and $\mathscr{D}$ matrices as follows:

$$C_{\mathrm{M}} = C \times \mathscr{D} \tag{5}$$

where $\mathscr{D} = [d_{ij}]$ is a matrix that represents the relative physical lengths of the links between routers after placement and routing. The entries of this matrix could be extracted from the ASIC design tool after the placement and routing. Then, C could be replaced by C_{M} in (3), in a back-annotation step, to acquire the exact network performability $\mathscr{P}_{\mathrm{total}}$. Because we target the early design phases in this paper, this step is left for future work.

5. TOPOLOGY-BASED ANALYSIS OF SYSTEM PERFORMABILITY

In this section, we study and analyze the impact of the network topology architecture on system performability using the model proposed in Section 4.3. Taking into consideration the possible changes in noise variance and voltage swing, several experiments are performed to evaluate the performability of the network and explore its characteristics.

5.1. Experimental work

A framework is developed using MATLAB® to generate connectivity matrices for the nine topologies mentioned in Section 3 and calculate their performability using (3). All experiments discussed in this section use SNFT as an example of a network communication scheme. However, other error-control schemes could be easily applied using the equations developed in [13].

To accurately evaluate the performability of different network topology architectures, the following experimental setup were used.

1. Nine connectivity matrices are generated to represent the connectivity of 12 PEs when mapped to the nine network topologies discussed in Section 3. The number of PEs is chosen to be in line with our case study, which is presented in Section 7.
2. A uniform traffic distribution is used to allow omitting the mapping factor (the location of each PE within the network topology). This means, each PE sends the same number of packets to the other 11 PEs. We designed the network to perform that way in order to study only the impact of the architecture without considering the mapping effect.
3. Router architecture is assumed to be based on an output-queue router as an example but other architectures could also be used.
4. Wormhole routing is used as a routing protocol.

Experimental results in Figure 4 show the impact of the network topology on system performability with respect to traffic rate, noise standard deviation (σ), and voltage swing (V_{sw}). The first set of experiments is performed to explore the effect of the traffic rate changes on the network performability. Figure 4(a) shows network performability versus traffic rate in packets per time step for 12 PEs mapped on the nine regular topologies in Figure 1. The voltage swing is set to 0.5 and white gaussian noise standard deviation (σ) is set to 0.06. With the increase in the traffic rate, Torus achieves the maximum performability compared with other topologies, whereas BT gives the minimum performability. The main observation that could be observed from Figure 4(a) is the significant effect of the topology architecture on the performability especially with the increase in the traffic rate. For instance, performability is degraded by 81.20% if BT is chosen instead of Torus to connect 12 PEs transferring 1000 packets/time step per node on a uniform traffic distribution pattern. Knowing this result is very important for designers at early design phases because it allows

DOI: 10.1002/cta

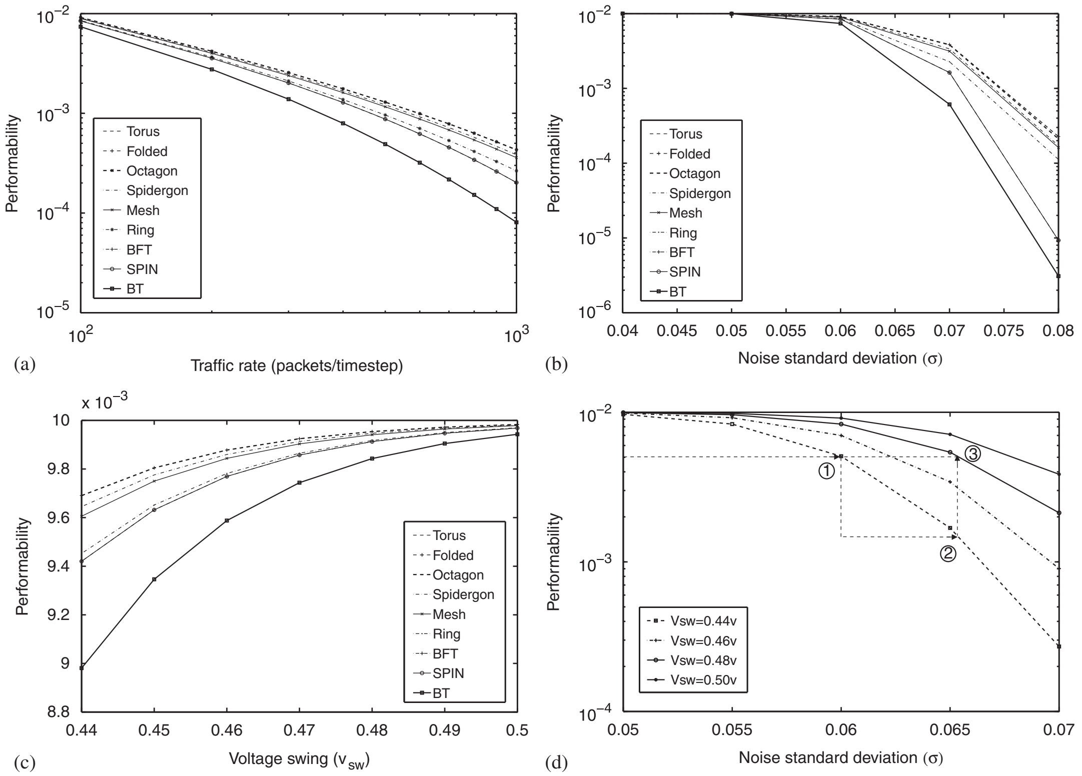

Figure 4. Experimental results that show the impact of the network topology on system performability with respect to traffic rate, noise standard deviation (σ), and voltage swing (V_{sw}). (a) Network performability versus traffic rate (log scale) for 12 PEs mapped on the nine regular topologies in Figure 1, when SNFT scheme is used. $V_{\mathrm{sw}}=0.5$ and $\sigma=0.06$. (b) Network performability versus noise standard deviation (σ) for the nine regular topologies in Figure 1, when SNFT scheme is used. $V_{\mathrm{sw}}=0.5$. (c) Network performability versus voltage swing (V_{sw}) for the nine regular topologies in Figure 1 when SNFT scheme is used. $\sigma=0.05$. (d) Network performability versus standard deviation for different voltage swings when SNFT scheme is used (for Torus topology).

them to choose the appropriate architecture that achieves the maximum performability at the system level design phase. Nevertheless, these results are just examples and cannot be generalized for all applications. In fact, the TDG and PE mapping, which are represented by the λ matrix, are vital factors and will definitely impact system performability.

The second set of experiments is carried out to measure the influence of noise power on system performability when different network architectures are used. Figure 4(b) shows network performability versus noise standard deviation (σ) for the nine regular topologies discussed in Section 3. As shown in the figure, Torus achieves the maximum performability, whereas BT has the minimum value, which is in line with the results of the previous experiment. However, the key observation in Figure 4(b) is that some network architectures are less immune to noise than others. In other words, performability degradation ratio, due to the increase in noise power, is not the same for all network topology architectures. For instance, when the value of σ increases from 0.07 to 0.08, the performability degrades by 94.11% for a Torus topology, whereas it degrades by 99.50% for a BT topology. Consequently, the difference between Torus and BT with respect to performability increases with the increase in noise power. For instance, at $\sigma=0.06$, BT topology achieves a performability of 19.29% less than Torus topology. This percentage increases to 98.64% at $\sigma=0.08$.

The third set of experiments is conducted to measure the impact of changing the voltage swing on the network performability for different network topologies. Voltage swing reduction is often used to address the energy consumption issues in NoC applications [37]. However, reducing the voltage swing increases the BER due to the increase in the signal-to-noise ratio (SNR).

Figure 4(c) shows network performability versus voltage swing for the same nine regular topologies. As shown in the figure, Torus achieves the maximum performability, whereas BT has the minimum value, which is in line with the results of the previous experiments. With the increase in the voltage swing, performability improvement ratio changes according to the used network topology. For instance, increasing the voltage swing from 0.44 to 0.50 leads to an improvement in the performability by 3.01% for Torus topology and by 10.71% for BT topology. Also, the difference between Torus and BT with respect to performability decreases with the increase of voltage swing. For instance, at $V_{\mathrm{sw}} = 0.45$, BT topology achieves a performability of 4.68% less than Torus topology. This percentage decreases to 0.40% at $V_{\mathrm{sw}} = 0.50$. Therefore, a joint consideration of voltage swing and noise power is needed to study the impact of both parameters simultaneously on the network performability.

The fourth set of experiments is performed to address the joint consideration of voltage swing and noise power on the network performability. Figure 4(d) shows network performability versus noise standard deviation (σ) for different swing voltages. Torus topology is taken as an example in this experiment to show how we can improve the performability of a certain topology by choosing the optimum value of V_{sw} for a system noise power level (σ). The main observation which could be made from Figure 4(d) is that increasing the voltage swing could compromise the increase in the noise power to a certain limit. Increasing the voltage swing leads to a corresponding increase in the noise margin, and hence, improve the SNR. To explain this, assume a scenario in which the noise standard deviation increases by 9.17% (by moving from point 1 to point 2 in Figure 4(d)). This increase leads to a corresponding degradation of 66.95% in the network performability. However, increasing the voltage swing by 9.09% (by moving from point 2 to point 3 in Figure 4(d)) compensates the noise increase effect and returns the network performability to its original value. Therefore, for applications that require a certain performability level, such as safety-critical applications, there is an optimum value of V_{sw} that could compensate the expected noise power level (σ) to achieve the design requirements.

5.2. Justifications of the experimental results

In this Subsection, we employ graph-theoretic concepts to study the differences between network architectures. One of the graph-theoretic definitions is the edge cut-set. An edge cut-set is a set of edges whose removal from the graph will disconnect the vertices and partition the network into two or more subnetworks. Table I shows the minimum edge cut-set for the topologies shown in Figure 1, when 12 PEs are connected. As shown in Table I, one broken edge is enough for a topology like BT to disconnect 8 PEs, which is 66.7% of the network, whereas four edges must be broken in a topology like Torus to disconnect only one PE, which is 8.3% of the network in this example.

Table I. Minimum edge cut-set and other statistical analysis for NoC topologies shown in Figure 1, when 12 PEs are connected.

Network topology	No. of edges (m)	Minimum edge cut-set (s)	Percentage s/m (γ) (%)	Total no. of PEs (n)	Max no. of PEs to fail (n_f)	Percentage n_f/n (ϑ) (%)
Mesh	17	2	11.8	12	1	8.3
Torus	24	4	16.7	12	1	8.3
Folded torus	24	4	16.7	12	1	8.3
Ring	12	2	16.7	12	6	50
Octagon	22	3	13.6	12	1	8.3
Spidergon	24	3	12.5	12	1	8.3
Binary Tree	10	1	10	12	8	66.7
BFT	6	2	33.3	12	4	33.3
SPIN	9	3	33.3	12	4	33.3

DOI: 10.1002/cta

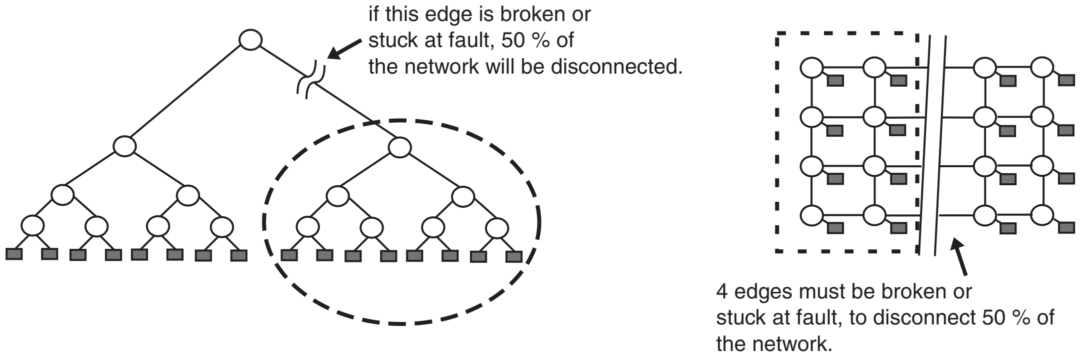
if this edge is broken or
stuck at fault, 50 % of
the network will be disconnected.
4 edges must be broken or
stuck at fault, to disconnect 50 % of
the network.

Figure 5. An example of a Binary Tree network. If one edge of the top level edges is physically broken or always stuck at fault due to high static noise source, 50% of the network will be disconnected.

These statistics reflect several important features, which are unique for each topology. These features could be used as indicators to evaluate the performability of a certain topology relative to others. For instance, maximizing the ratio γ is an essential requirement because it reflects the network ability to preserve its functionality at edge failure. Additionally, minimizing the ratio ϑ is another important objective because it indicates the level of partial system failure at worst case scenarios. The results in Table I are consistent with the results obtained in Section 5.1. As shown in Figure 5, it is crystal clear that the BT architecture is the only one in which 50% of its nodes could be disconnected if only one edge, at the top level, happened to be physically broken or always stuck at fault due to high static noise source. On the other hand, topologies such as mesh need at least four edges to fail in order to disconnect the same percentage of the nodes. This fact remains true no matter what the size of the network is. This discussion justifies the results obtained in Section 5.1.

To conclude the above findings in Section 5.1, there are five important parameters which significantly affect the performability of a NoC-based system: network topology architecture, traffic distribution, PEs' mapping, noise power, and voltage swing. A joint consideration of two or three of these five parameters could lead to a considerable improvement in the performability. However, all of them must be considered simultaneously to acquire an optimum performability, at early system level design phases, for a given NoC-based application. Also, the main reason behind the results found in the previous four-set of experiments is that tree-based topologies, such as BT, have an architecture that forces packets to go through longer paths to reach their distention as the number of levels in the tree increases. On the other hand, mesh-based topologies, especially Torus, have an architecture that allows shorter packet transmission paths.

However, we must emphasise that these results might change if a different traffic pattern (different application) is considered. In our experiment we assumed that each PE sends the same number of packets to all other PEs. We designed the network to perform in that way in order to study only the impact of the architecture without considering the mapping effect. For a different traffic distribution, BT might become the optimum choice and Octagon might be the worse. In other words, the traffic distribution matrix (λ), which is a unique matrix for each application, is a major design parameter that significantly affects the choice of the target network topology. Also, the noise level (σ) and voltage swing (V_{sw}) must be considered based on the design constraint and requirements. We considered all these parameters in our proposed methodology in Section 6.2.

6. PROBLEM FORMULATION AND PROPOSED METHODOLOGY

To acquire a maximum network performability, certain design parameters must optimally be chosen. In this section, we present a mathematical formulation of this optimization problem, then introduce our proposed methodology to solve the optimization problem and, hence, find the network topology architecture that achieves a maximum performability.

6.1. Problem formulation

For a given application represented by a TDG and a traffic distribution matrix (λ)

$$\text{Maximize } \mathscr{P}_{\text{total}} = f(\mathscr{H}, C, V_{\text{sw}}, \sigma) \tag{6}$$

Subject to:

$$c_{ij} = c_{ji}$$
$$V_{\text{sw}a} \leqslant V_{\text{sw}} \leqslant V_{\text{sw}b}$$
$$\sigma_a \leqslant \sigma \leqslant \sigma_b$$

where $C = [c_{ij}]$ represents the connectivity matrix as discussed in Section 4.3. $V_{\text{sw}a}$ and $V_{\text{sw}b}$ are the minimum and maximum permissible swing voltage according to design requirements of a given application respectively. σ_a and σ_b are the margins of the expected noise standard deviation.

6.2. Proposed methodology

The proposed methodology to improve network performability targets the nine regular topologies, discussed in Section 3, aiming at maximizing network performability. Other regular or irregular topologies could be easily considered by adding their C matrices to increase the granularity of the topologies' library. Figure 6 shows a flowchart of the proposed methodology to improve the performability based on topology optimization. This methodology could be explained as follows.

The first step of the proposed methodology considers the TDG as the main design input. Then, from the TDG, a $\lambda = [\lambda_{ij}]$ matrix is generated, with an initial mapping, as a mathematical representation of the graph, where λ_{ij} is the number of packets per time step associated with each

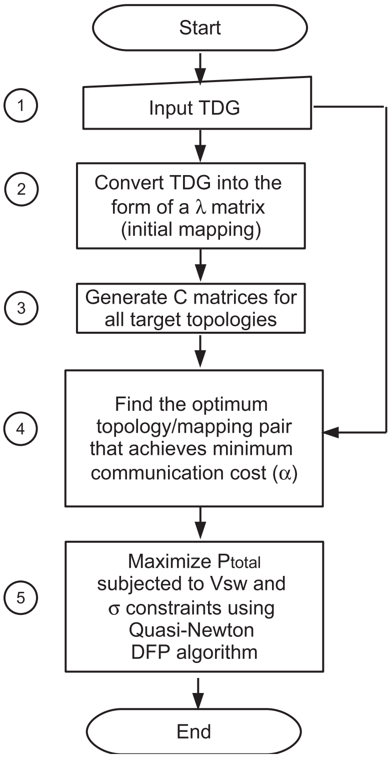
Start
1
Input TDG
2
Convert TDG into the form of a λ matrix (initial mapping)
3
Generate C matrices for all target topologies
4
Find the optimum topology/mapping pair that achieves minimum communication cost (α)
5
Maximize Ptotal subjected to Vsw and σ constraints using Quasi-Newton DFP algorithm
End

Figure 6. Proposed methodology to improve NoC performability using a topology-based approach.

DOI: 10.1002/cta

logical link. The third step is to generate C matrices for all nine topologies based on the number of PEs given in the TDG. Following that, we maximize the performability in two consecutive steps (the fourth and fifth steps). The fourth step finds the optimum topology/mapping pair that achieves minimum communication cost (α) assuming shortest path static routing, where α is given by

$$\alpha = \min\left(\sum_{i=1}^{n}\sum_{j=1}^{n}\lambda_{ij}\times c_{ij}\right), \quad i\neq j \tag{7}$$

From (3) and (4), reducing the minimum communication cost (α) increases the corresponding performability. This is done by re-mapping all PEs such that those who have the highest traffic rates are associated with the shortest distances. A search algorithm is developed to re-allocate PEs to different positions within each one of the nine topologies. The re-mapping process is done by analyzing all numbers in the λ matrix. Then, all possible changes in rows and columns, on a one-by-one base, are done to re-arrange the matrix. This exhaustive search is performed to re-map the PEs such that the highest traffic rates (numbers) are associated with shortest distances (neighboring nodes). Finally, we use the Quasi-Newton David–Fletcher–Powell (DFP) algorithm [38] to find the optimum V_{sw} that could compensate the expected noise level σ and, at the same time, achieves a maximum performability ($\mathscr{P}_{\mathrm{total}}$). Quasi-Newton DFP is used because it is very efficient for more than one parameter varying at the same time, which is the case in our optimization problem. The DFP algorithm is modified to generate a set of optimum points, for V_{sw}, that achieve maximum performability values with a variation of 10^{-5} . Finally, the V_{sw} value that achieves a maximum performability, under given noise level constraints, is selected.

One of the advantages of the proposed methodology is its scalability. The proposed methodology could be used for any application, regardless the number of PEs, as long as its TDG is provided. The generation of λ and C matrices, topology/mapping search, and the mathematical operations of the DFP algorithms are executed using MATLAB®.

7. PERFORMANCE EVALUATION BY EXPERIMENTATION

The proposed methodology is validated through an experimental case study. The video object plane decoder (VOPD) core discussed in [39] is taken as an example to evaluate the performance of the proposed methodology.

Figure 7(a) shows a typical TDG for the video application (VOPD) [39]. The numbers written on the arrows are the number of packets/time step transmitted and the numbers written on the circles

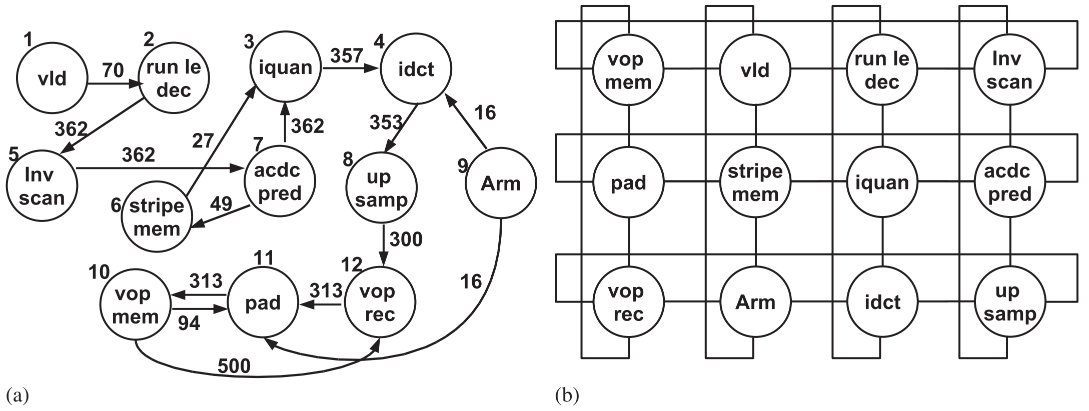

Figure 7. (a) TDG of Video Object Plane Decoder (VOPD) [39] and (b) VOPD mapping onto a Torus topology (an output from the proposed methodology).

represent the PEs' numbers. From this TDG, the traffic distribution matrix (λ), which represents the initial mapping of the VOPD, shown in Figure 7(a), is given by

$$\lambda=\begin{bmatrix}
0 & 70 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 362 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 357 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 353 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 362 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 27 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 362 & 0 & 0 & 49 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 300\\
0 & 0 & 0 & 16 & 0 & 0 & 0 & 0 & 0 & 0 & 16 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 94 & 500\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 313 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 313 & 0
\end{bmatrix}$$

Following that, connectivity matrices are generated for all nine topologies discussed in Section 4.3. A search algorithm developed in MATLAB® is used to find the optimum topology/mapping pair that achieves minimum communication cost (α). Figure 7(b) shows the mapping of VOPD to a Torus topology after applying our search algorithm. The traffic distribution matrix (λ), after mapping, is given by

$$\lambda=\begin{bmatrix}
0 & 94 & 500 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
313 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 313 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 70 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 27 & 0 & 0 & 0 & 0\\
0 & 16 & 0 & 0 & 0 & 0 & 0 & 0 & 16 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 362 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 357 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 353\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 362 & 0\\
0 & 0 & 0 & 0 & 49 & 0 & 0 & 362 & 0 & 0 & 0 & 0\\
0 & 0 & 300 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

The minimum communication cost for all nine topologies are calculated using (7) and compared to show the significant impact of the topology/mapping selection on the network communication cost. As shown in (7), the communication cost is calculated by counting how many hops are needed, for each packet, to reach its destination. Figure 8 shows the minimum communication cost versus topology type after mapping the VOPD core. Results show that for VOPD, a Torus topology achieves the minimum α, whereas BT has the maximum α. The reason behind this result is that the BT architecture forces the packets to go through longer pathes as the number of levels in the tree increases. The communication cost (α) could have been increased by 341.01% if a BT topology is chosen instead of the generated Torus. This example highlights the importance

DOI: 10.1002/cta

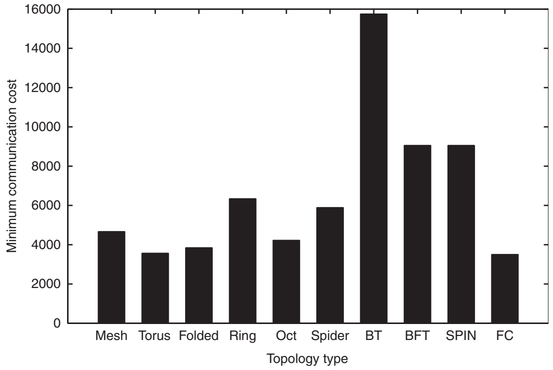

Figure 8. Minimum communication cost (α) versus topology type for the VOPD application. FC represents a fully connected network.

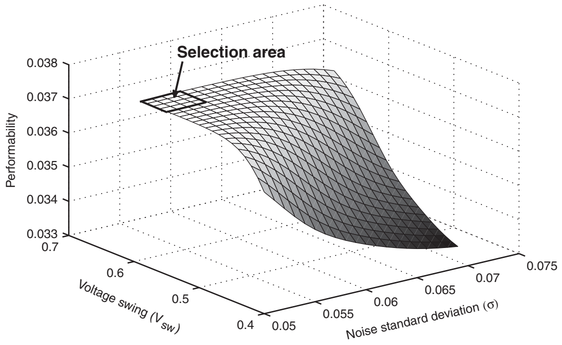

Figure 9. Network performability versus standard noise deviation (σ) and voltage swing (V_{sw}) for an VOPD application mapping to a Torus topology.

of choosing the most appropriate topology/mapping pair for a given application. To verify the efficiency of our search algorithm, we compared the communication cost of the generated Torus to the Torus mapping proposed in [39] and to a fully connected network (crossbar connection) architecture. We found that the communication cost of the generated Torus is 1.35% less than the Torus mapping proposed in [39] and only 1.86% more than a fully connected network.

Finally, the Quasi-Newton DFP algorithm explained in [38] is used to find the set of optimum V_{sw} values that compensates the noise level given in the design constraints. Then, the V_{sw} value that achieves a maximum performability ($\mathscr{P}_{\mathrm{total}}$), under given noise level constraints, is selected. For this case study, we used the following design constraints:

$$0.40 \leqslant V_{\mathrm{sw}} \leqslant 0.65$$

$$0.05 \leqslant \sigma \leqslant 0.08$$

and the initial point (x_0), which is required to start the search in the Quasi-Newton DFP algorithm, is set, randomly, to

$$x_0 = [V_{\mathrm{sw}0}\sigma_0] = [0.40\ 0.05]$$

The resulting output has a performability of 0.3778 at $V_{\mathrm{sw}} = 0.54$ and $\sigma = 0.05$. To have a better understanding of the behavior of performability when noise level and voltage swing change, a 3-D curve is plotted. Figure 9 shows performability versus standard noise deviation and voltage swing for the VOPD application mapping to a Torus topology. The ranges used for both V_{sw} and σ are the ones used as constraints in the optimization problem. Results show that we can find multiple sets of points to choose from to achieve the maximum performability. The selection area shown in the figure contains multiple set of points that achieves a maximum performability of $0.3778xx$. The differences between these points, in terms of performability values, do not exceed 10^{-5}, which is not a vital difference compared with the original value. Results in Figure 9 also highlight the significant effect of the proposed methodology. Choosing inappropriate design value such as $V_{\mathrm{sw}}=0.41$ for a noise level of $\sigma=0.068$ results in a degradation of 10.77% in the network performability compared with the value achieved by the proposed methodology.

8. CONCLUSION AND FUTURE WORK

A new methodology to acquire the optimum topology architecture to achieve maximum performability was presented in this paper. The proposed methodology targets nine regular topologies aiming at maximizing NoC performability at early design phases. Graph-theoretic concepts were adopted to study the topological architecture of NoCs and identify the impact of the network topology architecture on systems' performability. Using these concepts, we studied important design parameters, which significantly affect the performability of a NoC-based system; such as network topology architecture, traffic distribution, noise power, and voltage swing.

As a proof of concept, we validated the proposed methodology through a case study of a VOPD application. Results show that performability could have been degraded by 10.77%, compared with the value achieved by the proposed methodology, if inappropriate parameters were chosen.

We plan to extend this research in three directions. First, we will extend the proposed methodology to more advanced design phases (after placement and routing). The relative physical lengths of the links between routers will be extracted from an ASIC design tool after placement and routing to be used in a back-annotation step in order to acquire the network performability $\mathscr{P}_{\mathrm{total}}$.

Next, we will study the impact of changing the routing protocol on the network performability. Then, we plan to develop a methodology to acquire the most efficient routing protocol that achieves maximum performability for a given application.

REFERENCES

1. Micheli GD, Benini L. *Networks on Chips*: *Technology and Tools*. Morgan Kaufmann Publishers: San Francisco, CA, U.S.A., 2006.
2. Ádm Rák GC, Soós BG. Stochastic bitstream-based CNN and its implementation on FPGA. *Journal of Circuit Theory and Applications* 2009; **37**(4):587–612.
3. Meyer JF. Models and techniques for evaluating the effectiveness of aircraft computing systems. *Semi Annual Status Report NASA-CR-158992 Status Rpt*, University of Michigan, Ann Arbor, MI, 1978.
4. Smith RM, Trivedi KS, Ramesh AV. Performability analysis: measures, an algorithm, and a case study. *IEEE Transactions on Computers* 1988; **37**(4):406–417.
5. Tamhankar RR, Murali S, Micheli GD. Performance driven reliable link design for networks on chips. *Proceedings of the 2005 Conference on Asia South Pacific Design Automation* (*ASP-DAC'05*), Shanghai, China, 18–21 January 2005; 749–754.
6. Ejlali A, Al-Hashimi BM, Rosinger P, Miremadi SG, Benimi L. Performability/energy trade-off in error-control schemes for on-chip networks. *IEEE Transactions on Very Large Scale Integration*. Available: http://eprints.ecs.soton.ac.uk/15948/1/TVLSI_noc.pdf; in press.
7. Ahonen T, Siguenza-Tortosa DA, Bin H, Nurmi J. Topology optimization for application-specific networks-on-chip. *Proceedings of the 2004 International Workshop on System Level Interconnect Prediction* (*SLIP'04*), Paris, France, 14–15 February 2004; 53–60.
8. Dalirsani A, Hosseinabady M, Navabi Z. An analytical model for reliability evaluation of NoC architectures. *Proceedings of the 13th IEEE International On-Line Testing Symposium* (*IOLTS'07*), Crete, Greece, 8–11 July 2007; 49–56.
9. Pande PP, Grecu C, Jones M, Ivanov A, Saleh R. Performance evaluation and design trade-offs for network-on-chip interconnect architectures. *IEEE Transactions on Computers* 2005; **54**(8):1025–1040.
10. Varma A, Eaghavendra C. *Interconnection Networks for Multiprocessors and Multicomputers Theory and Practice*. IEEE Computer Society Press: Los Alamitos, CA, 1991.
11. Bertozzi D, Benini L, De Micheli G. Error control schemes for on-chip communication links: the energy-reliability tradeoff. *The IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 2005; **24**(6):818–831.
12. Murali S, Theocharides T, Vijaykrishnan N, Irwin MJ, Benini L, Micheli GD. Analysis of error recovery schemes for networks on chips. *IEEE Design and Test of Computers* 2005; **22**(5):434–442.

DOI: 10.1002/cta

13. Ejlali A, Al-Hashimi BM, Rosinger P, Miremadi SG. Joint consideration of fault-tolerance, energy-efficiency and performance in on-chip networks. *Proceedings of the Conference on Design*, *Automation and Test in Europe* (*DATE'07*), Nice, France, 16–20 April 2007; 1647–1652.
14. Chang K-C, Chen T-F. Low-power algorithm for automatic topology generation for application-specific networks on chips. *IET Computers and Digital Techniques* 2008; **2**(3):239–249.
15. Elmiligi H, Morgan A, El-Kharashi MW, Gebali F. Power-aware topology optimization for networks-on-chips. *Proceedings of the IEEE International Symposium on Circuits and Systems* (*ISCAS'08*), Seattle, WA, U.S.A., 18–21 May 2008; 360–363.
16. Srinivasan K, Chatha K, Konjevod G. Linear-programming-based techniques for synthesis of network-on-chip architectures. *IEEE Transactions on Very Large Scale Integration Systems* 2006; **14**(4):407–420.
17. Ascia G, Catania V, Di Nuovo AG, Palesi M, Patti D. Efficient design space exploration for application specific systems-on-a-chip. *Journal of System Architecture* 2007; **53**(10):733–750.
18. Ogras U, Marculescu R. It's a small world after all: NoC performance optimization via long-range link insertion. *IEEE Transactions on Very Large Scale Integration* 2006; **14**(7):693–706.
19. Koide T, Kubo H, Watanabe H. A study on the tie-set graph theory and network flow optimization problems. *Journal of Circuit Theory and Applications* 2004; **32**(6):447–470.
20. Kumar S, Jantsch A, Millberg M, Oberg J, Soininen J-P, Forsell M, Tiensyrja K, Hemani A. A network on chip architecture and design methodology. *Proceedings of the IEEE Computer Society Annual Symposium on VLSI* (*ISVLSI'02*), Pittsburgh, PA, U.S.A., 25–26 April 2002; 105–112.
21. Dally W, Towles B. Route packets, not wires: on-chip interconnection networks. *Proceedings of the 38th Design Automation Conference* (*DAC'01*), Las Vegas, NV, U.S.A., 18–22 June 2001; 683–689.
22. Dally W, Seitz C. The torus routing chip. *Distributed Computing* 1986; **1**(4):187–196.
23. Xu J. *Topological Structure and Analysis of Interconnection Networks*. Kluwer Academic Publishers: Norwell, MA, U.S.A., 2001.
24. Karim F, Nguyen A, Dey S. An interconnect architecture for networking systems on chips. *IEEE Micro* 2002; **22**(5):36–45.
25. Coppola M, Locatelli R, Maruccia G, Pieralisi L, Scandurra A. Spidergon: a novel on-chip communication network. *Proceedings of the 2004 International Symposium on System-on-Chip* (*SoC'04*), Tampere, Finland, 16–18 November 2004; 15.
26. Moadeli M, Shahrabi A, Vanderbauwhede W, Ould-Khaoua M. An analytical performance model for the Spidergon NoC. *Proceedings of the 21st International Conference on Advanced Information Networking and Applications* (*AINA'07*), Niagara Falls, ON, Canada, 21–23 May 2007; 1014–1021.
27. Pande P, Grecu C, Ivanov A, Saleh R. Design of a switch for network on chip applications. *Proceedings of the 2003 International Symposium on Circuits and Systems* (*ISCAS'03*), Bangkok, Thailand, 25–28 May 2003; 217–220.
28. Guerrier P, Greiner A. A generic architecture for on-chip packet-switched interconnections. *Proceedings of the Design and Test in Europe Conference* (*DATE'00*), Paris, France, 27–30 March 2000; 250–256.
29. Agnarsson G, Greenlaw R. *Graph Theory*: *Modeling*, *Applications*, *and Algorithms*. Prentice-Hall: Englewood Cliffs, NJ, U.S.A., 2006.
30. Constantinescu C. Impact of deep submicron technology on dependability of VLSI circuits. *Proceedings of the 2002 International Conference on Dependable Systems and Networks* (*DSN'02*), Bethesda, MD, U.S.A., 23–26 June 2002; 205–209.
31. Reorda SM, Violante M. Emulation-based analysis of soft errors in deep sub-micron circuits. *Proceedings of the International Conference on Field-Programmable Logic and Applications* (*FPL'03*), Lisbon, Portugal, 1–3 September 2003; 616–626.
32. Grecu C, Anghel L, Pande PP, Ivanov A, Saleh R. Essential fault-tolerance metrics for NoC infrastructures. *Proceedings of the 13th IEEE International On-Line Testing Symposium* (*IOLTS'07*), Crete, Greece, 9–11 July 2007; 37–42.
33. Pande PP, Grecu C, Ganguly A, Ivanov A, Saleh R. Test and fault tolerance for networks-on-chip infrastructures. *Networks-on-Chips*: *Theory and Practice*, Chapter 7. CRC Press: Taylor & Francis, Florida, U.S.A., 2009.
34. Jantsch A, Tenhunen H. *Networks on Chip*. Kluwer Academic Publishers: Norwell, MA, U.S.A., 2003.
35. Worm F, Ienne P, Thiran P, Micheli GD. A robust self-calibrating transmission scheme for on-chip networks. *IEEE Transactions on Very Large Scale Integration* 2005; **13**(1):126–139.
36. Gross JL, Yellen J. *Graph Theory and its Applications* (2nd edn). CRC Press: Boca Raton, FL, U.S.A., 2005.
37. Svensson C. Optimum voltage swing on on-chip and off-chip interconnect. *IEEE Journal of Solid-State Circuits* 2001; **36**(7):1108–1112.
38. Antoniou A, Lu W-S. *Practical Optimization*: *Algorithms and Engineering Applications*. Springer: Berlin, Germany, 2007.
39. Bertozzi D, Jalabert A. NoC synthesis flow for customized domain specific multiprocessor systems-on-chip. *IEEE Transactions on Parallel and Distributed Systems* 2005; **16**(2):113–129.

DOI: 10.1002/cta