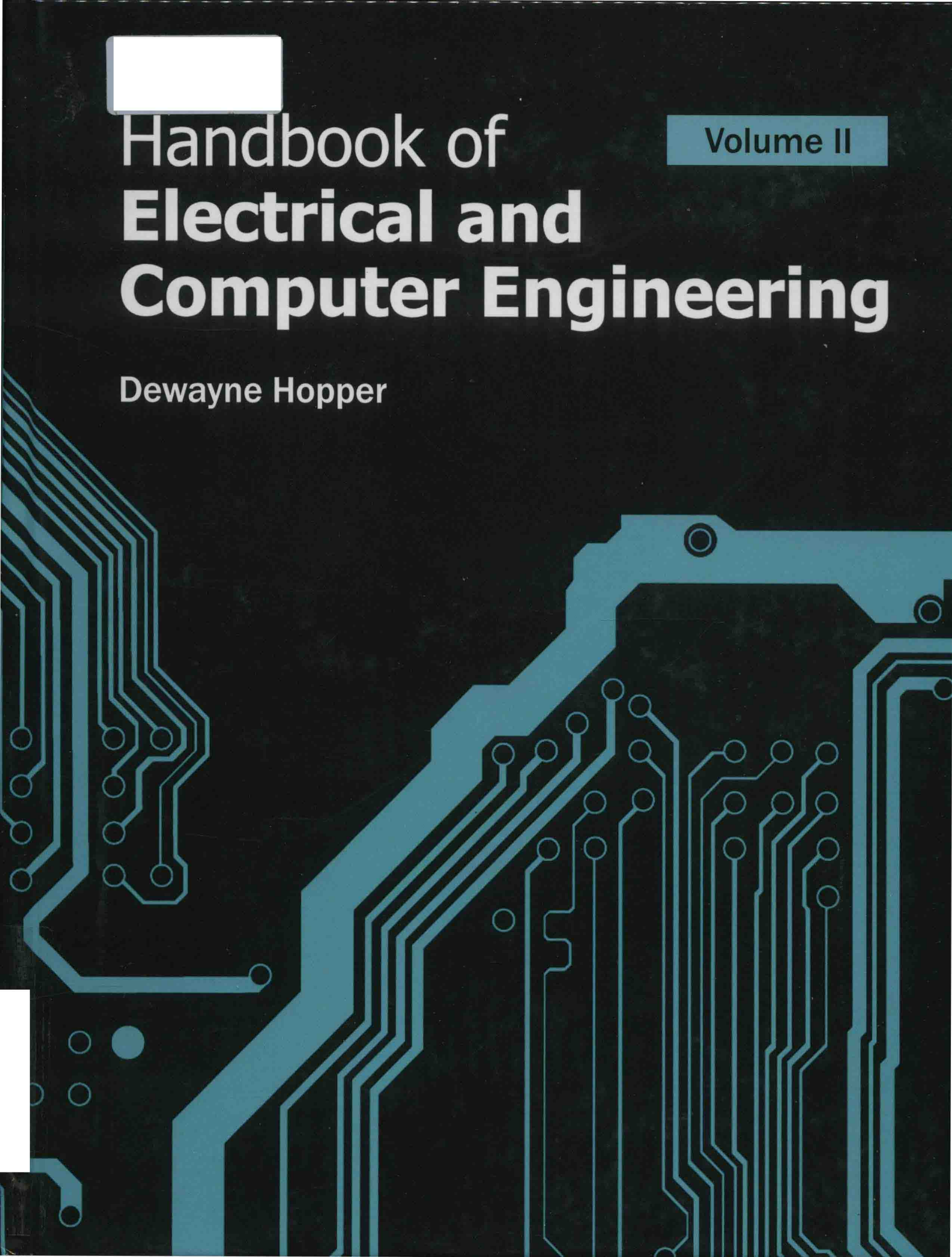
Handbook of Electrical and Computer Engineering

Volume II

Dewayne Hopper



Handbook of Electrical and Computer Engineering

Volume II

Edited by **Dewayne Hopper**

NY RESEARCH PRESS

New York

Published by NY Research Press,
23 West, 55th Street, Suite 816,
New York, NY 10019, USA
www.nyresearchpress.com

Handbook of Electrical and Computer Engineering: Volume II
Edited by Dewayne Hopper

International Standard Book Number: 978-1-63238-235-1 (Hardback)

Printed in China.

Handbook of Electrical and Computer Engineering

Volume II

Preface

The field of engineering that deals with the study and application of electronics and electricity is termed as electrical engineering. This field gained recognition as one of the biggest fields of engineering during the 19th century. Electric power has gained a foothold in the most basic activities of our lives. There is not a part of our daily lives that is not involved with electrical power in some way or the other. There are many subdivisions of electrical engineering like telecommunications, control systems, instrumentation and electronics, though electronics can be termed as a field in its own right. The skills required of an electrical engineer are variable and they work in a huge variety of industries. Computer engineering on the other hand is a discipline that combines various fields of electrical engineering and computer science and specializes in the development of computer hardware and software. Computer engineers can be involved in the many hardware and software facets of computing, like the design of microprocessors and personal computers to circuit design. It can be said that this field of study focuses on both how computer systems work as well as their integration in the larger picture. Both electrical and computer engineering are disciplines that can be counted as the biggest and fastest growing in the range of engineering fields that are there in the industry.

This book is an attempt to collate all current data and research on computer and electrical engineering. I am thankful to all the contributing authors for the hard work and effort put in these researches. I also wish to acknowledge the efforts of the publishing team who provided excellent technical assistance, whenever needed. Lastly, I wish to thank my friends and family who have supported me at every step in my life.

Editor

Contents

On Asymptotic Analysis of Packet and Wormhole Switched Routing Algorithm for Application-Specific Networks-on-Chip

Nitin

Department of CSE and ICT, Jaypee University of Information Technology, Waknaghat, Solan 173234, Himachal Pradesh, India

Correspondence should be addressed to Nitin, delnitin@ieee.org

Academic Editor: Vivek Kumar Sehgal

The application of the multistage interconnection networks (MINs) in systems-on-chip (SoC) and networks-on-chip (NoC) is hottest since year 2002. Nevertheless, nobody used them practically for parallel communication. However, to overcome all the previous problems, a new method is proposed that uses MIN to provide intra-(global) communication among application-specific NoCs in networks-in-package (NiP). For this, four $O(n)^2$ fault-tolerant parallel algorithms are proposed. It allows different NoCs to communicate in parallel using either fault-tolerant irregular Penta multistage interconnection network (PNN) or fault-tolerant regular Hexa multistage interconnection network (HXN). These two are acting as an interconnects-on-chip (IoC) in NiP. Both IoC use packet switching and wormhole switching to route packets from source NoC to destination NoC. The results are compared in terms of packet losses and wormhole switching which comes out to be better than packet switching. The comparison of IoC on cost and MTTR concluded that the HXN has the higher cost than the PNN, but MTTR values of the HXN are low in comparison to the PNN. This signifies that the ability to tolerate faults and online repairing of the HXN is higher and faster than the PNN.

1. Introduction and Motivation

Parallel Processing refers to the concept of speeding-up the execution of a program by dividing the program into multiple fragments that can execute simultaneously, each on its own processor. A program being executed across n processors might execute n times faster than it would use a single processor.

It is known that one way for processors to communicate data is to use a shared memory and shared variables. However, this is unrealistic for large numbers of processors. A more realistic assumption is that each processor has its own private memory and data communication takeing place using message passing via an Interconnection Networks (INs).

INs originated from the design of high-performance parallel computers. INs make a major factor to differentiate modern multiprocessor architectures and are categorized according to a number of criteria such as topology, routing strategy, and switching technique. IN is building up of switching elements; topology is the pattern in which the individual switches are connected to other elements, like processors, memories, and other switches.

1.1. Interconnection Networks.

> *"Interconnection Networks should be designed to transfer the maximum amount of information within the least amount of time (and cost, power constraints) so as not to bottleneck the system."*

INs have a long development history [1–8]. The Circuit switched networks have been used in telephony. In 1950s, the interconnection of computers and cellular automata as few prototypes was developed until 1960 when it awaited full use. Solomon in 1962 developed multicomputer network. Staran with its flip network, C.mmp with a crossbar and Illiac-IV with a wider 2D network received attention in early 1970s. This period also saw several indirect network used in vector and array processors to connect multiple processors to multiple memory banks. This problem was developed in several variants of MINs. The BBN Butterfly in 1982 was one of the first multiprocessors to use as an indirect network. The binary e-cube or hypercube network was proposed in 1978 and implemented in the Caltech Cosmic Cube in 1981. In the early 1980s, the academic focus was on mathematical properties of these networks and

became increasingly separated from the practical problems of interconnecting real systems.

The last decade was the golden period for INs research driven by the demanding communication problems of multicomputer enabled by the ability to construct single-chip Very Large Scale Integration (VLSI) routers, the researchers have made a series of breakthroughs that have revolutionized in digital communication systems. The Torus Routing Chip, in 1985, was one unique achievement. The first of a series of single-chip routing components introduced wormhole routing and virtual channels used for deadlock avoidance. The whole family of chip laid the framework for analysis of routing, flow-control, deadlock, and livelock issues in modern direct networks. A flurry of research followed with new theories of deadlock and livelock, new adaptive routing algorithms, and new methods for performance analysis. The research progressed in collective communication and network architectures on a regular basis. By the early 1990s, low-dimensional direct networks had largely replaced the indirect networks of the 1970s, and the hypercubes of the 1980s could be found in machines from Cray, Intel, Mercury, and some others. The applicability of INs in digital communication systems with the appearance of Myrinet was adopted in 1995. The point-to-point multiple networks technology replaced the use of buses, which were running into a limited performance due to electrical limits and were used in the barrier network in the Cray T3E, as an economical alternative to dedicated wiring. However, the interconnection network technology had certain barriers on design, and the various researchers and engineers have observed analysis of these networks [1, 4–8].

1.1.1. Multistage Interconnection Networks. As the acceptance and subsequent use of multiprocessor systems increased, the reliability, availability, performability, and performance characteristics of the networks that interconnect processors to processors, processors to memories, and memories to memories are receiving increased attention. A brief survey of INs and a survey of the fault-tolerant attributes of MINs are reported in [1–8]. A MIN in particular is an IN that consists of cascade of switching stages, contains switching elements (SEs). MINs are widely used for broadband switching technology and for multiprocessor systems. Besides this, MINs offer an enthusiastic way of implementing switches used in data communication networks. With the performance requirement of the switches exceeding several terabits/sec and teraflops/sec, it becomes imperative to make them dynamic and fault tolerant [9–14].

The typical modern day application of the MINs includes fault-tolerant packet switches, designing multicast, broadcast router fabrics, while SoCs and NoCs are hottest research topics in current trends [9–14]. Normally the following aspects are always considered while deigning the fault-tolerant MINs: the topology chosen, the routing algorithm used, and the flow control mechanism adhered. The topology helps in selecting the characteristics of the present chip technology in order to get the higher bandwidth, throughput, processing power, processor utilization, and probability of acceptance from the MIN-based applications, at an optimum hardware cost. Therefore, it has been decided to work on both irregular and regular fault-tolerant MINs as an application for NoCs.

1.2. Networks-in-Package. Networks-in-package (NiP) designs provide integrated solutions to challenging design problems in the field of multimedia and real-time embedded applications. The main characteristics of NiP platforms are as follows:

(1) networking between chip-to-chip in a single package,

(2) low development cost than NoC approach,

(3) low power consumption,

(4) high performance,

(5) small area.

Along with these characteristics, there are various fields to explore in NiP, which include the following:

(1) communication versus computation,

(2) deep Sub-Micron effect,

(3) power,

(4) global synchronization,

(5) heterogeneity of functions.

This paper focuses on an emerging paradigm that effectively addresses and presumably overcomes the many on-chip interconnection and communication challenges that already exist in today's chips or will likely occur in future chips. This new paradigm is commonly known as the NoC paradigm [15–18]. The NoC paradigm is one, if not the only one, fit for the integration of an exceedingly large number of computational, logic, and storage blocks in a single chip. Notwithstanding this school of thought, the adoption and deployment of NoC face important issues relating to design and test methodologies and automation tools. In many cases, these issues remain unresolved.

1.3. Networks-on-Chip. NoC is an emerging paradigm for communications within VLSI systems implemented on a single silicon chip. In a NoC system, modules such as processor cores, memories, and specialized Intellectual Property (IP) blocks exchange data using a network as a "public transportation" subsystem for the information traffic. A NoC is constructed from multiple point-to-point data links interconnected by switches, such that messages can be relayed from any source module to any destination module over several links, by making routing decisions at the switches. A NoC is similar to a modern telecommunications network, using digital bit-packet switching over multiplexed links. Although packet switching is sometimes claimed as a necessity for NoC, there are several NoC proposals utilizing circuit-switching techniques. This definition based on routers is usually interpreted so that a single shared bus, a single crossbar switch, or a point-to-point network is not NoC, but practically all other topologies are. This is somewhat confusing since all above mentioned are networks

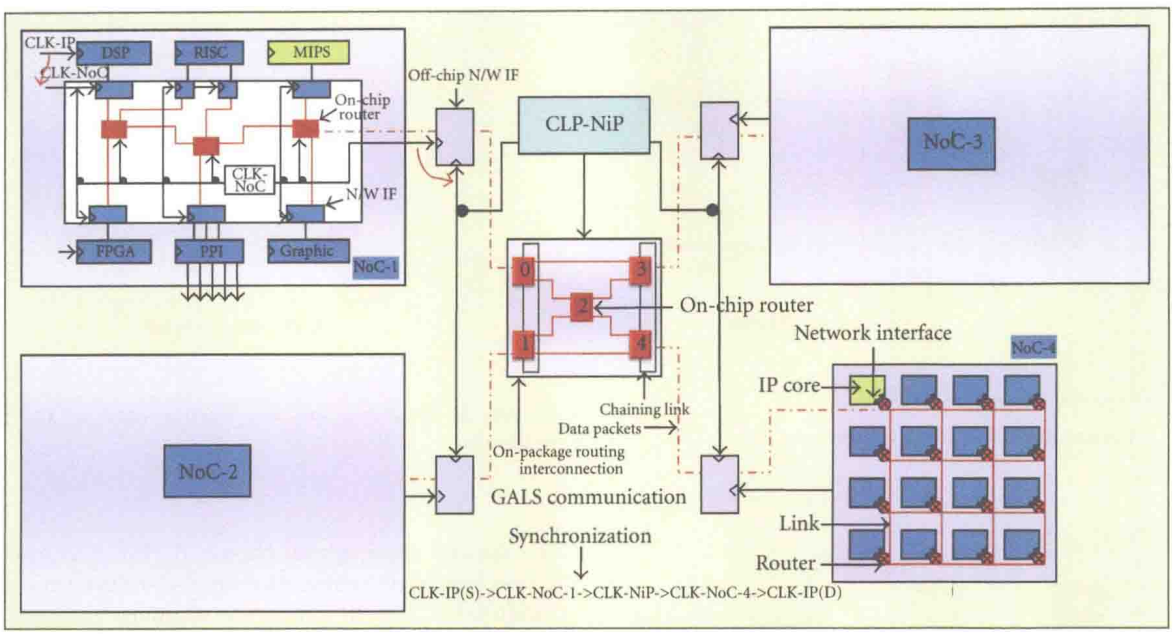
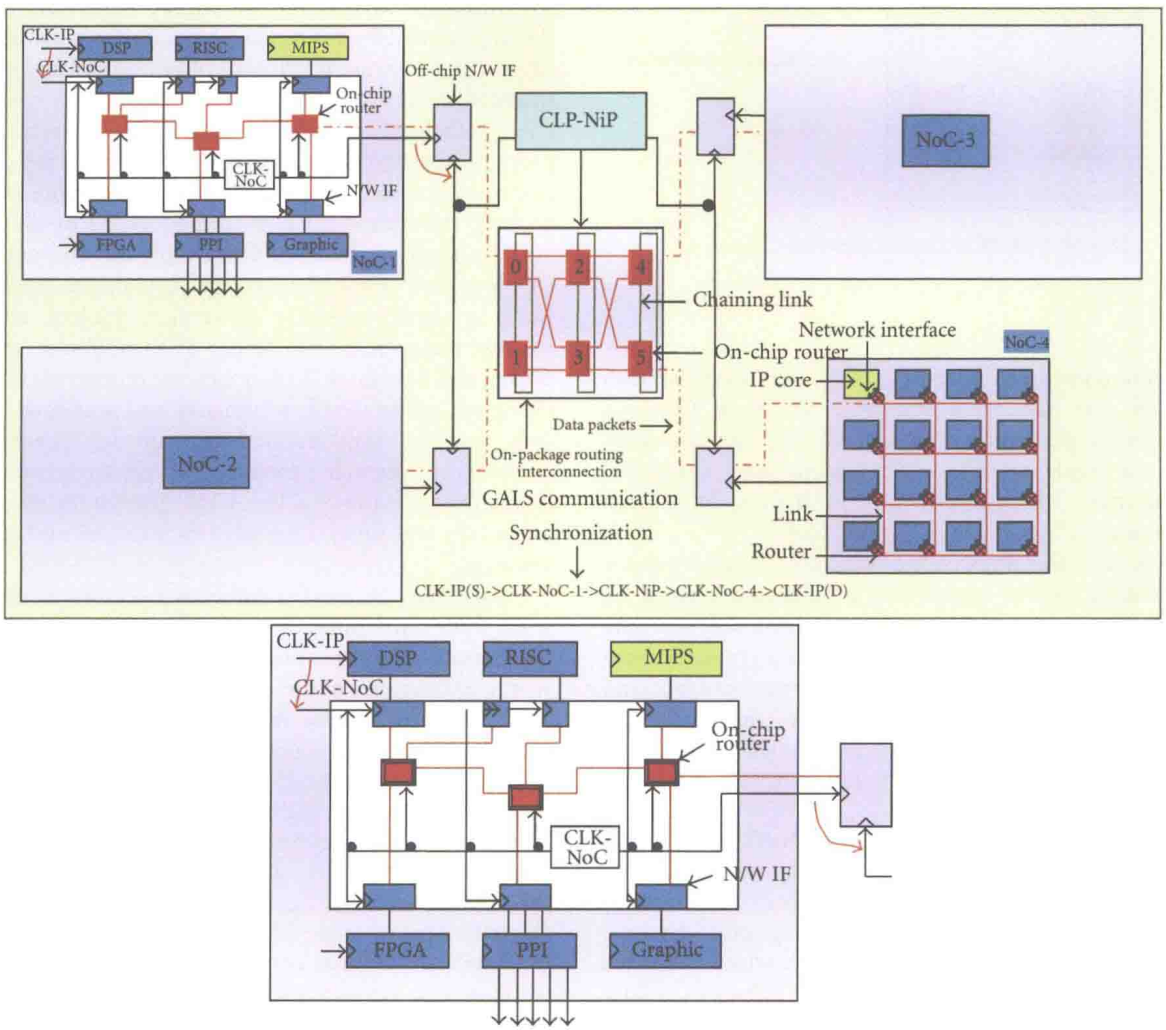


FIGURE 1: Parallel communication among various NoCs using PNN.



(a) Internal architecture of NoC-1 used in NiP.

FIGURE 2: Parallel communication among various NoCs using HXN.

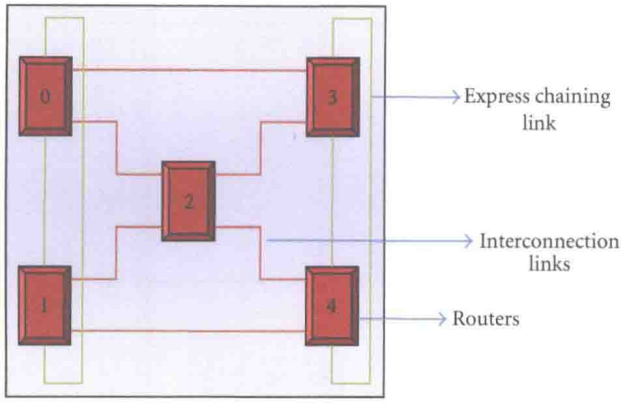


FIGURE 3: PNN as Interconnect-on-Chip.

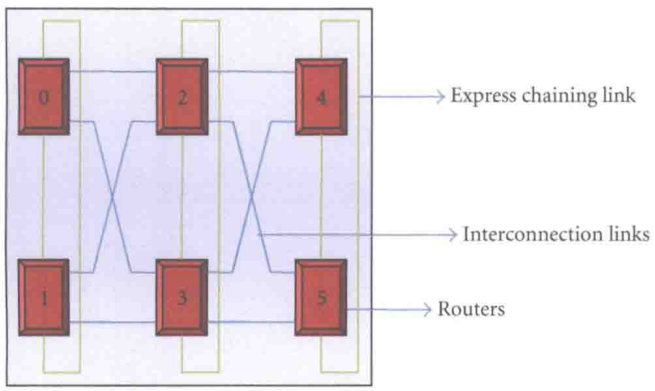


FIGURE 4: HXN as Interconnect-on-chip.

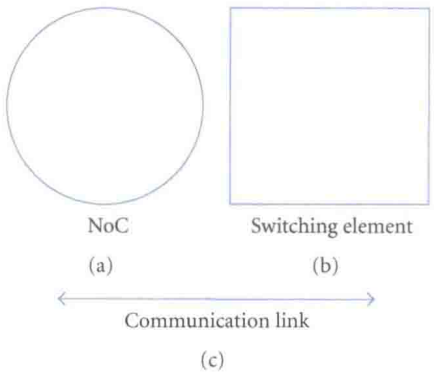


FIGURE 5: Following are the legends used in the simulation technique.

but are not considered as NoC. Note that some articles erroneously use NoC as a synonym for mesh topology although NoC paradigm does not dictate the topology. Likewise, the regularity of topology is sometimes considered as a requirement, which is obviously not the case in research concentrating on "Application-Specific NoC".

The wires in the links of the NoC are shared by many signals. A high level of parallelism is achieved, because all links in the NoC can operate simultaneously on different data packets. Therefore, as the complexity of integrated systems keeps growing, a NoC provides enhanced performance and scalability in comparison with previous communication architectures. Of course, the algorithms must be designed in such a way that it offers large parallelism and can hence utilize the potential of NoC.

Several forces drive the adoption of NoC architecture: from a physical design viewpoint, in nanometer Complementary Metal-Oxide Semiconductor (CMOS) technology interconnects dominate both performance and dynamic power dissipation, as signal propagation in wires across the chip requires multiple clock cycles. NoC links can reduce the complexity of designing wires for predictable speed, power, noise, reliability, and so on, thanks to their regular, well-controlled structure. From a system design viewpoint, with the advent of multicore processor systems, a network is a natural architectural choice. A NoC can provide separation between computation and communication, support modularity and IP reuse via standard interfaces, handle synchronization issues, serve as a platform for system test, and, hence, increase engineering productivity.

Although NoC can borrow concepts and techniques from the well-established domain of computer networking, it is impractical to reuse features of old networks and symmetric multiprocessors. In particular, NoC switches should be small, energy efficient, and fast. Neglecting these aspects along with proper, quantitative comparison was typical for early NoC research, but today all are considered in more detail. The routing algorithms must be implemented by simple logic, and the number of data buffers should be minimal. Network topology and properties may be Application Specific. NoC need to support quality of service, namely, achieve the various requirements in terms of throughput, end-to-end delays, and deadlines. To date, several prototypes of NoCs are designed and analyzed in industry and academia. However, only few are implemented on silicon. However, many challenging research problems remain to be solved at all levels, from the physical link level through the network level and all the way up to the system architecture and application software.

Most NoC are used in embedded systems, which interact with their environment under more or less hard time constraints. The communication in such systems has a strong influence on the global timing behavior. Methods are needed to analyze the timing, as average throughput as well as worst-case response time [17]. However, from a VLSI design perspective, the energy dissipation profile of the interconnect architectures is of prime importance as the latter can represent a significant portion of the overall energy budget. The silicon area overhead due to the interconnect fabric is important too. The common characteristic of these kinds of architectures is such that the processor/storage cores communicate with each other through high-performance links and intelligent switches and such that the communication design is represented at a high abstraction level. The different NoC topologies are already used in [19], and these topologies give different communication structure in NoC [20].

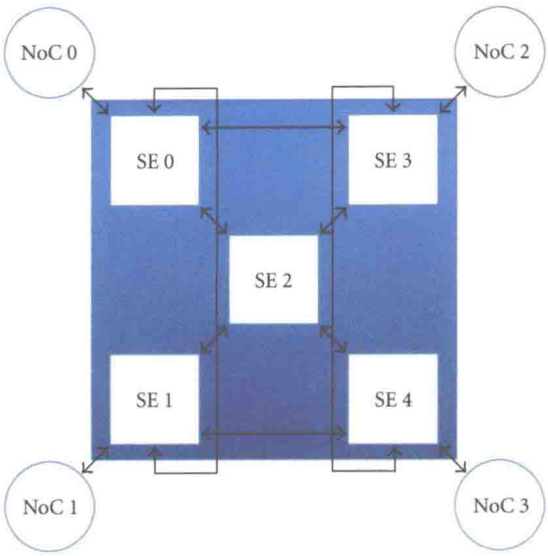


FIGURE 6: The initial architectural NiP model using PNN.

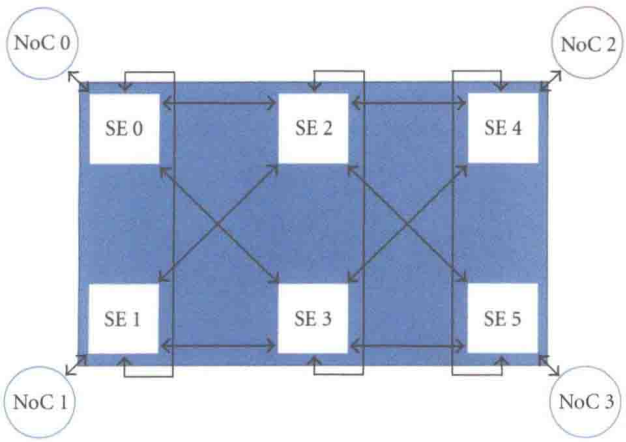


FIGURE 7: The initial architectural NiP model using HXN.

The application of the MIN in SoCs [10, 15–18] and NoCs [10, 16–18] is consistently drawing attention since year 2002. The parallel communication among Application-Specific NoC [9, 10] is a major problem to handle by the researchers. Nevertheless, nobody used them practically for parallel communication. The literature survey reveals that Star, Common Bus, and Ring topologies were used as a medium to set up intra-NoCs communication [21]. However, these communication systems have many tradeoffs and disadvantages as mentioned below:

(1) high latency,

(2) low scalability,

(3) poor performance,

(4) zero fault-tolerance,

(5) no On-chip repairability,

(6) high contention over sharing of channel,

(7) presence of livelock,

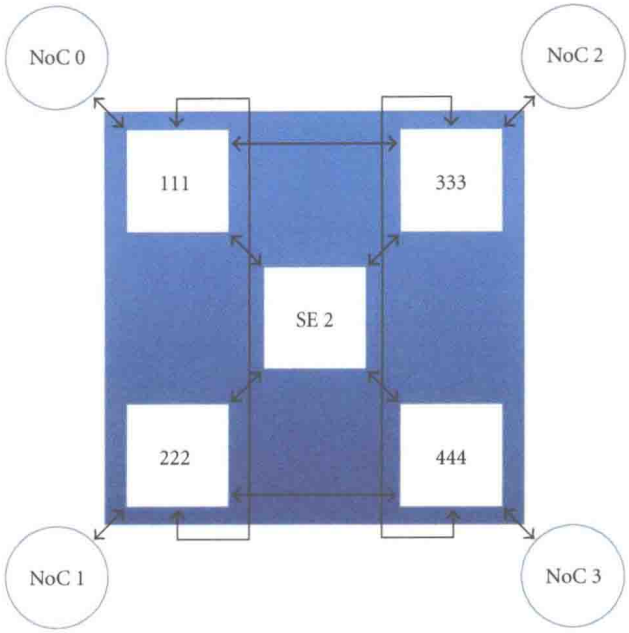


FIGURE 8: The simulation state of NiP-PNN model at first step of best case.

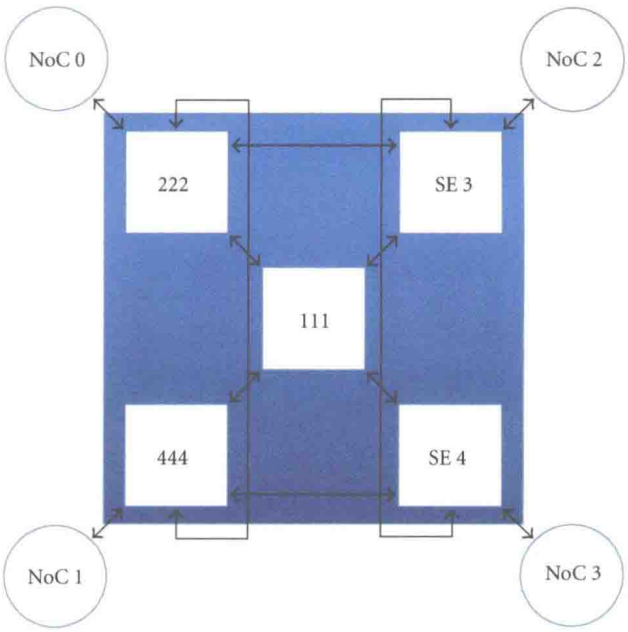


FIGURE 9: The simulation state of NiP-PNN model at second step of best case.

(8) presence of deadlock,

(9) low probability of acceptance of data packets.

However, to overcome all the previous problems, for the first time this paper proposes a new method that sets up intra-(global) communication between Application-Specific (heterogeneous or homogenous) NoCs in NiP. The said architecture uses $O(n)^2$ time fault-tolerant packet and wormhole switching parallel algorithms. These algorithms allow different NoCs to communicate efficiently in parallel with

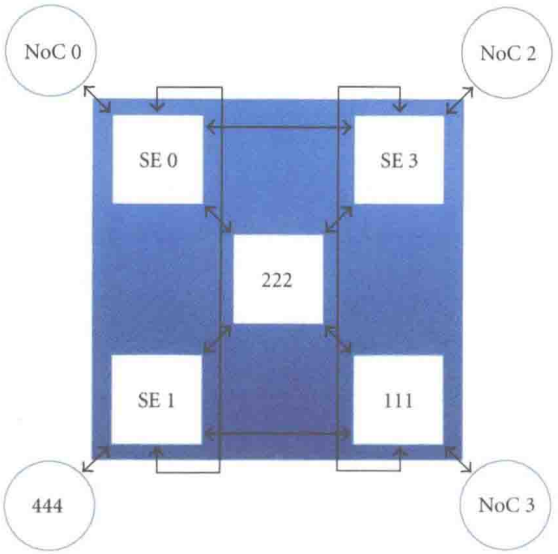


FIGURE 10: The simulation state of NiP-PNN model at third step of best case.

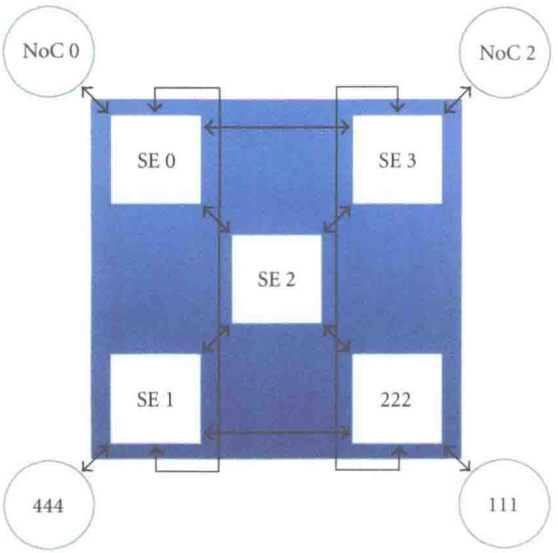


FIGURE 11: The simulation state of NiP-PNN model at fourth step of best case.

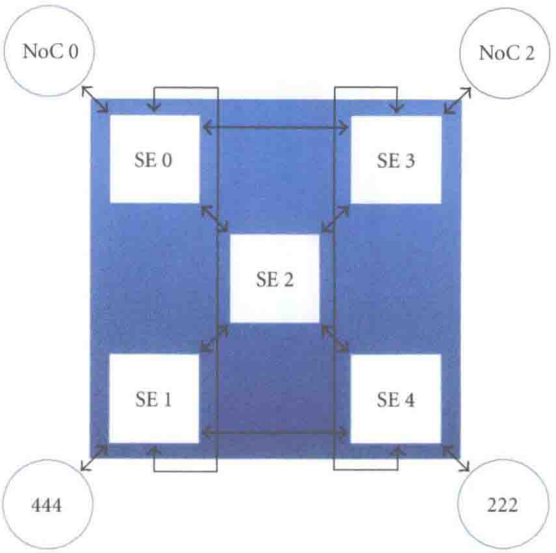


FIGURE 12: The simulation state of NiP-PNN model at fifth step of best case.

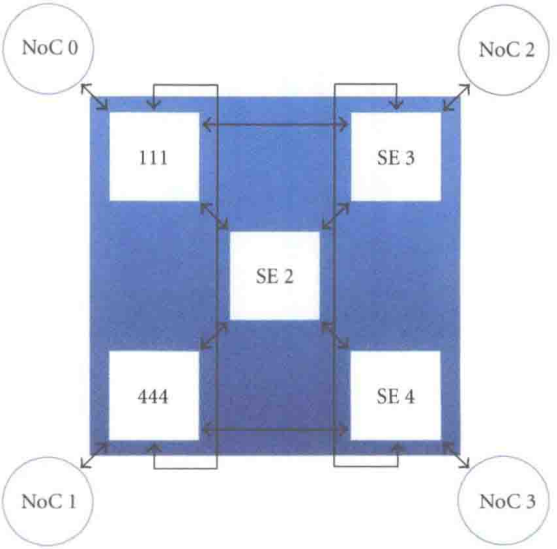


FIGURE 13: The simulation state of NiP-PNN model at first step of worst case.

minimum number of the packet losses. Out of the two IoCs, one is 2 × 2 fault-tolerant irregular Penta multistage interconnection networks (PNNs), with 3 stages and 5 SEs (all stages include chaining or express links, except the middle one) and other is 2 × 2 fault-tolerant regular Hexa multistage interconnection networks (HXNs), with 3 stages and 6 SEs (all stages include chaining or express links, except the middle one).

The rest of the paper is organized as follows: Section 2 describes the general NiP architecture including the fault-tolerant parallel algorithm designed to provide parallel communication among different NoCs using HXN and PNN followed by the their comparisons on Cost and Mean Time to Repair (MTTR). Section 3 provides the conclusion followed by the references.

2. Application-Specific NiP Architecture Using Irregular PNN and Regular HXN

The general architecture of NiP resembles with the Open Systems Interconnection (OSI) Model. The Physical layer refers to all that concerns the electric details of wires, the circuits and techniques to drive information, while the Data Link level ensures a reliable transfer regardless of any unreliability in the physical layer and deals with medium access. At the Network level there are issues related to the topology and the consequent routing scheme, while the Transport layer manages the end-to-end services and the packet segmentation/reassembly. Upper levels can be viewed merged up to the Application as a sort of adaptation layer that implements services in hardware or through part of

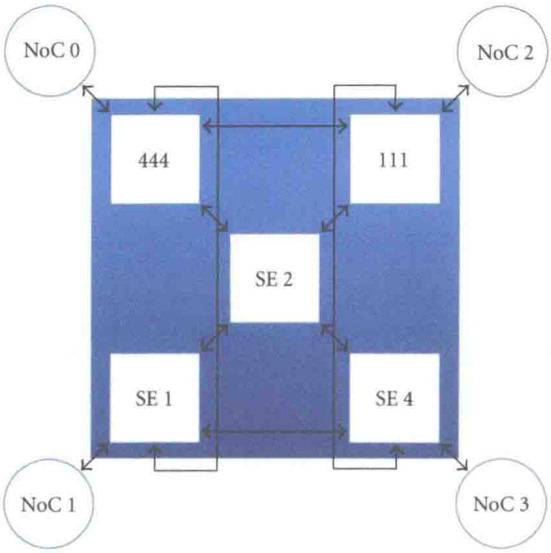


FIGURE 14: The simulation state of NiP-PNN model at second step of worst case.

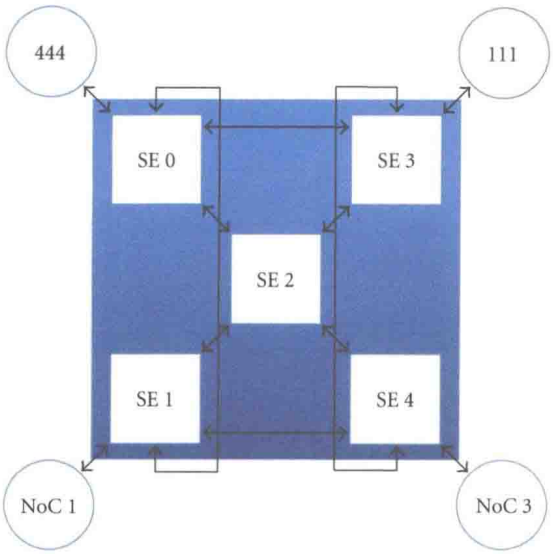


FIGURE 15: The simulation state of NiP-PNN model at third step of worst case.

an operating systems and exposes the NoC infrastructure according to a proper programming model, for example, the Message Passing (MP) paradigm.

NiP is a specific approach, which provides the common interface through which all NoC can communicate together more efficiently and robustly. It contains three different types of building blocks, appropriately interconnected to each other and a patented network topology that promises to deliver the best price/performance trade-off in future NiP applications as follows [20].

Figures 1 and 2 show a general NiP architecture in which four NoC chips are mounted on a single package. These NoC communicate with each other through an intermediate chip, known as IoC [22].

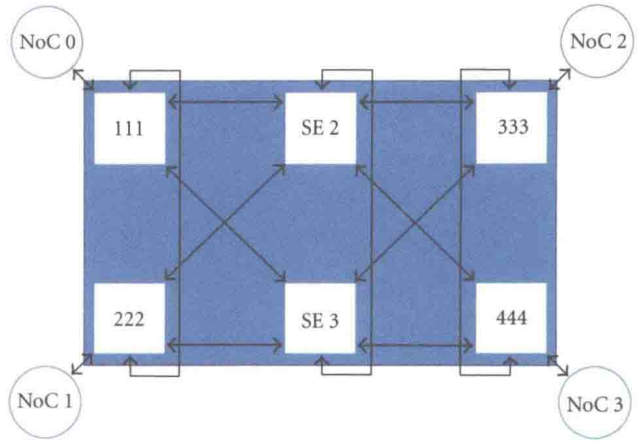


FIGURE 16: The simulation state of NiP-HXN model at first step of best case.

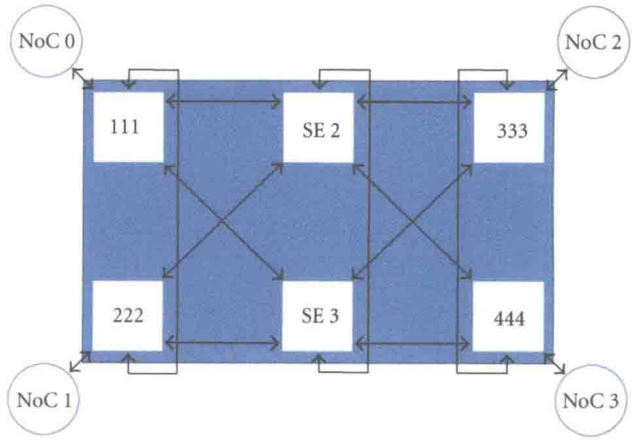


FIGURE 17: The simulation state of NiP-HXN model at second step of best case.

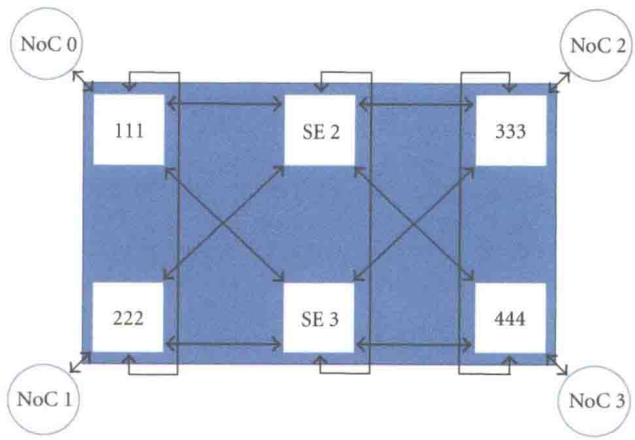


FIGURE 18: The simulation state of NiP-HXN model at third step of best case.

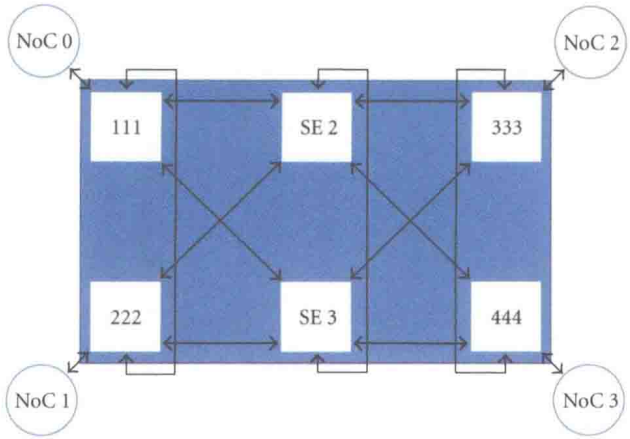


FIGURE 19: The simulation state of NiP-HXN model at fourth step of best case.

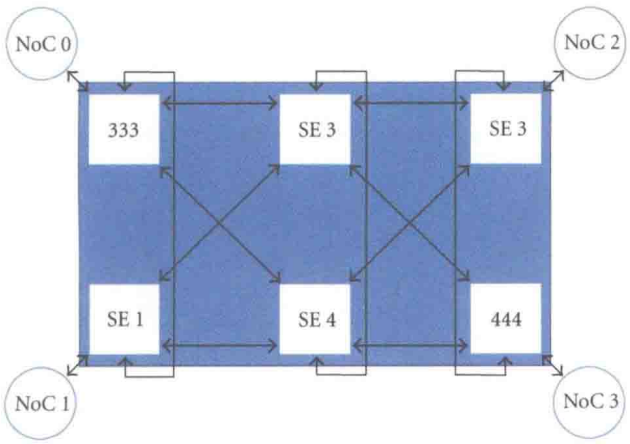


FIGURE 20: The simulation state of NiP-HXN model at first step of worst case.

2.1. Interconnects-on-Chip Architecture. Figures 3 and 4 show the different types of architecture of IoC, that is, one belongs to the class of irregular fault-tolerant MIN and other belongs to the class of regular fault-tolerant MIN. This first chip, shown in Figure 3, consists of five routers; working as Switching Elements (SEs) is known as PNN and Figure 4 shows the architecture of HXN with 6 SEs.

These routers are connected with the main link and chaining or express links, which makes the IoC highly, fault tolerant. Here the architectural design of IoC is similar to a small MIN, widely used for broadband switching technology and for multiprocessor systems. Besides this, it offers an enthusiastic way of implementing switches/routers used in data communication networks. With the performance requirement of the switches/routers exceeding several terabits/sec and teraflops/sec, it becomes imperative to make them dynamic and fault tolerant. The typical modern day application of the MIN includes fault-tolerant packet switches, designing multicast, broadcast router fabrics while SoC and NoC are hottest now days [9–14].

TABLE 1: Initial source and destination information used to set up communication between NoCs.

Source NoC	Destination NoC	Payload
0	3	111
1	3	222
2	3	333
3	1	444

TABLE 2: Intermediate SE information while data moves from source NoC to destination NoC at first step of best case.

Source NoC	Destination NoC	Switching element at first step
0	3	0
1	3	1
2	3	3
3	1	4

TABLE 3: Intermediate SE information while data moves from source NoC to destination NoC at second step of best case.

Source NoC	Destination NoC	Status/switching element at second step
0	3	2
1	3	0
2	3	Destroyed
3	1	1

TABLE 4: Intermediate SE information while data moves from source NoC to destination NoC at third step of best case.

Source NoC	Destination NoC	Status/switching element at third step
0	3	4
1	3	2
3	1	Reached successfully at NoC 1

TABLE 5: Intermediate SE information while data moves from source NoC to destination NoC at fourth step of best case.

Source NoC	Destination NoC	Status/switching element at third step
0	3	Reached successfully at NoC 3
1	3	4
3	1	Reached successfully at NoC 1

2.2. Switching Methodologies, Testbed, and Assumptions. Switching techniques determine when and how internal switches connect their inputs to outputs and the time at which message components may be transferred along these paths. For uniformity, the same approach for all NoC architectures has been used here. There are different types of switching techniques [6–8] as follows.

Definition 1 (*Circuit Switching*). A physical path from source to destination is reserved prior to the transmission of the data. The path is held until all the data has been