

# A New Efficient Smart Reliable Network-on-Chip

Manoj.R

P.G.scholar, ME VLSI design, Sree Sakthi Engineering College, Coimbatore -641104

*manojmano.inc@gmail.com*

**Abstract**— In this paper, we present a new network-on-chip (NoC) that handles accurate localizations of the faulty parts of the NoC. The proposed NoC is based on new error detection mechanisms suitable for dynamic NoCs, where the number and position of processor elements or faulty blocks vary during runtime. Indeed, we propose online detection of data packet and adaptive routing algorithm errors. Both presented mechanisms are able to distinguish permanent and transient errors and localize accurately the position of the faulty blocks (data bus, input port, output port) in the NoC routers. The propose system is Decimal Matrix Code (DMC) to enhance the data reliability. The mechanism accurately identifies the location of faulty block in NoC router and rectifies it. The performance of system is analysed and results show that error correction is feasible without compromising speed and resources.

**Index Terms**— Adaptive algorithm, dynamic reconfiguration, network-on-chip (NoC), reliability, Decimal Matrix Code (DMC).

## I.INTRODUCTION

RECENTLY the trend of embedded systems has been moving toward multiprocessor systems-on-chip (MPSoCs) in order to meet the requirements of real-time applications. The complexity of these SoCs is increasing and the communication medium is becoming a major issue of the MPSoC. Generally, integrating a network-on-chip (NoC) into the SoC provides an effective means to interconnect several processor elements (PEs) or intellectual properties (IP) (processors, memory controllers, etc.]. The NoC medium features a high level of modularity, flexibility, and throughput. An NoC comprises routers and interconnections allowing communication between the PEs and/or IPs. The NoC relies on data packet exchange. The path for a data packet between a source and a destination through the routers is defined by the routing algorithm. Therefore, the path that a data packet is allowed to take in the network depends mainly on the adaptiveness permitted by the routing algorithm (partially or fully adaptive routing algorithm), which is applied locally in each router being crossed and to each data packet.

Dynamically reconfigurable 2-D mesh NoCs (DyNoC, CuNoC, QNoC, ConoChi, etc.) are suitable for field-programmable gate array (FPGA)-based systems. Thanks to the partial dynamic reconfiguration of FPGAs with varying position and the number of implemented PEs and IPs, higher adaptiveness is allowed in MPSoCs during runtime. To achieve a reconfigurable NoC, an efficient dynamic routing algorithm is required for the data packets. The goal is to preserve flexibility and reliability while providing high NoC performance in terms of throughput. Fig. 1 illustrates a dynamic reliable NoC.

Fig. 1(a) shows the communications between several IPs and Fig. 1(b) and (c) depicts the dynamic placement of an IP and the occurrence of a faulty node, respectively, both cases where bypasses determined by the dynamic routing algorithm are required. Furthermore, faulty nodes or even faulty regions make communications within the networks harder and even impossible with some routing algorithms, as shown in Fig. 1(c). Therefore, dynamic component placement and faulty nodes or regions are the main reasons why fault-tolerant or adaptive algorithms have been introduced and used in runtime dynamic NoCs. Regarding adaptive or fault-tolerant routing algorithms, several solutions have been proposed. Generally, these algorithms correspond to a modified XY routing algorithm that allows faulty or unavailable regions to be bypassed. In the case of adaptive routing algorithms based on the turn model, zones are defined corresponding to faulty nodes or unavailable regions already detected in the NoC.

The neighboring routers of these zones must not send data packets towards these known faulty routers or unavailable regions. Several solutions have been proposed to achieve this constraint. One solution is to include a routing table containing the output port to use for each destination in the network. These tables are updated by an initialization algorithm. The main drawback of this solution is the requirement to invoke the algorithm at a non specified time in order to update the routing tables of the NoC routers. Another solution usually applied is the use of chains and rings formed

around the adjacent faulty nodes and regions, in order to delimit rectangular parts in the NoC covering all the faulty nodes or unavailable regions. These chains or rings of switches modify the routing tables, which therefore differ from the standard tables realizing the XY routing algorithm. These specific switches integrate in their tables additional routing rules that allow the faulty zones and regions dedicated to dynamic IP/PE instantiations to be bypassed, while avoiding starvation, deadlock, and livelock situations. Another reliable routing algorithm solution is the use of the de Bruijn graph. This algorithm is deadlock-free and handles the bypassing of faulty links between two switches by assuming that nodes are aware of the faulty link that is connected to them by the use of a detection mechanism. However, these solutions do not give the mechanism to detect a faulty link or router.

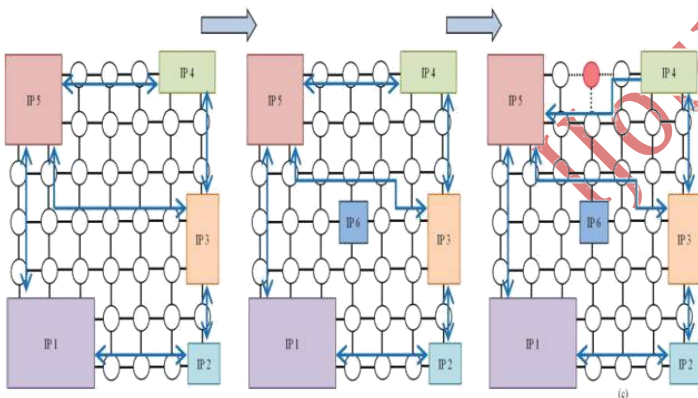
With regard to the increasing complexity and the reliability evolution of SoCs, MPSoCs are becoming more sensitive to phenomena that generate permanent, transient, or intermittent faults. These faults may generate data packet errors, or may affect router behavior leading to data packet losses or permanent routing errors. Indeed, a fault in a routing logic will often lead to a packet routing

NoC. The main drawback of this solution is its incapacity to locate the faulty components (PE, IP, router, data bus, etc.) in the NoC. Consequently, it is inadequate for dynamic NoCs, where the faulty and unavailable zones must be bypassed. Second, the switch-to-switch detection is based on the implementation of an ECC in each input port of the NoC switches. For instance, in a router of four communication directions (North, South, East, and West), four ECC blocks are implemented. Therefore, when a router receives a data packet from a neighbor, the ECC block analyzes its content to check the correctness of the data. This process detects and corrects data errors according to the effectiveness of the ECC being used. Third, another proposed solution is the code disjoint. In this approach, routers include one ECC in each input and output data port. This solution localizes the error sources, which can be either in the switches or on the data links between routers. However, if an error source is localized inside a router, this solution mechanism disables the totality of the switch. These online detection mechanisms cannot disconnect just the faulty parts of the NoC, and hence do not give an accurate localization of the source of errors. The result is that the network throughput decreases while the network load and data packet latency increase. Moreover, they are not able to distinguish between permanent and transient errors. For all these techniques, each ECC implemented in the routers of the network adds cost in terms of logic area, latency in data packet transmission, and power consumption.

An analysis of the source and destination addresses, as presented in this paper and equally is among the techniques usually proposed to be able to detect faulty routing decisions.

When a router receives a data packet, it compares its own address to the destination and source addresses. Then, the router checks its own position in the deterministic XY path of the NoC for the considered data packet. The router performing this checking is able to decide whether the switch from which the packet was received made a routing error or not according to the correct XY path. However, this technique has a major drawback; it is unable to handle the bypass of faulty nodes and unavailable regions. Consequently, this solution cannot be applied in adaptive or fault-tolerant routing algorithms. Indeed, as specified in a turn model algorithm, the structure of the reconfigurable NoC may contain bypass areas in which the switches take routing decisions differently from the XY routing algorithm. For handling message routing errors in dynamic networks, a new faulty switch detection mechanism is required for adaptive or fault-tolerant routing algorithms.

In this paper, we present a new reliable dynamic NoC. The proposed NoC is a mesh structure of routers able to detect routing errors for adaptive routing based on the XY algorithm. Our approach includes data packet error detection and correction. The originality of the proposed architecture is its ability to localize accurately error sources, allowing the throughput and network load of the NoC to be maintained. In our case study, we consider a reliable approach based on the adaptive routing module proximity algorithm. The considered routing algorithm is based on the adaptive turn model routing scheme



**Fig. 1. Illustration of a dynamic reliable NoC. (a) Normal operation. (b) Dynamic implementation of an IP. (c) Online detection of a faulty router**

errors and might even crash the router. To detect these errors, specific error detection blocks are required in the network to locate the faulty sources. Moreover, permanent errors must be distinguished from transient errors. Indeed, the precise location of permanent faulty parts of the NoC must be determined, in order for them to be bypassed effectively by the adaptive routing algorithm.

To protect data packets against errors, error correcting codes (ECCs) are implemented inside the NoC components. Among the well known solutions, three are usually applied for the MPSoC communications based NoC. First, the end-to-end solution requires an ECC to be implemented in each input port of the IPs or PEs in the

and the well-known XY algorithm. This adaptive algorithm is livelock- and deadlock-free and allows data packets to pass around faulty regions.

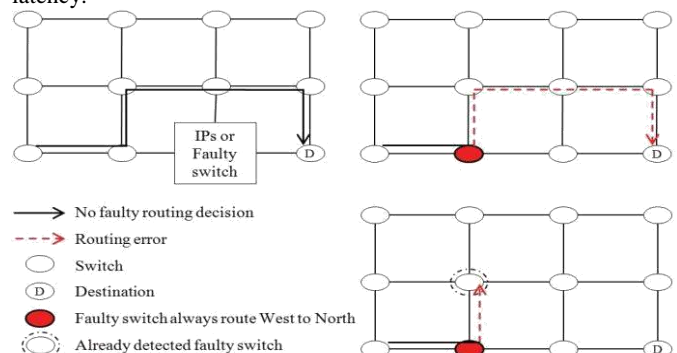
The remainder of this paper is organized as follows. Section II describes the architecture of the proposed reliable switch. Section III details the proposed routing error detection suitable for adaptive routing algorithms. Section IV presents a specific self-loopback mechanism allowing the avoidance of data packet loss, the maintaining of the performance of the NoC, and the effectively localizing of permanent sources of data packet errors. Section V presents FPGA synthesis and NoC performance evaluations, while

Section VI validates the proposed techniques by giving NoC's robustness and localization capacity of the error detection mechanisms.

network by being connected to several routers, or can be dynamically moved on the chip if this only access point becomes faulty. Each port direction is composed of two unidirectional data buses (input and output ports). Each input port is associated to a first-input, first-output (FIFO) (buffers) and a routing logic block. The RKT-switch operation is based on the store-and-forward switching technique. This technique is suitable for dynamically reconfigurable NoCs. Indeed, in our NoC, PEs and IPs can be implemented in place of one or several routers. At any instant with the store-and-forward technique, each data packet is stored only in a single router. Hence, when a router needs to be reconfigured, the router is only required to empty its buffers. On the contrary, with the wormhole switching technique, a single data packet can be spread over several routers. Consequently, the time required to clear all the routers containing partial packet data (flits) and to reconstruct these packets before performing a reconfiguration is more significant. Indeed, if a data packet is sent to a router surrounded by three unavailable nodes, the packet cannot be routed.

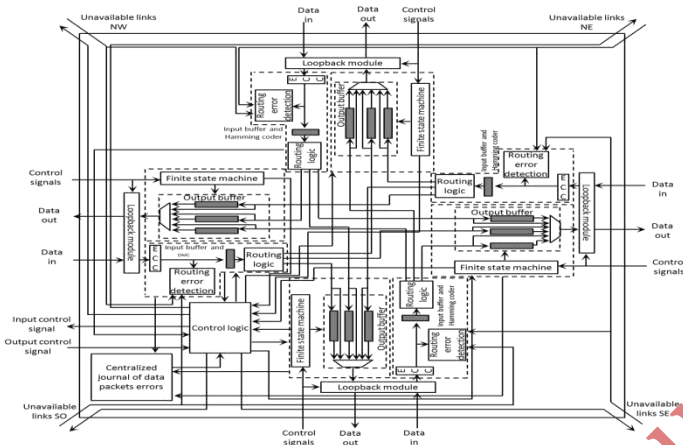
**III. ROUTING ERROR DETECTION**

The reliable switch being proposed incorporates an online routing fault detection mechanism. This approach can operate with adaptive algorithms based on the well-known XY routing algorithm. The main difficulty in routing error detection is to distinguish a bypass of an unavailable component in the NoC (due to the use of the adaptive algorithm) from a real routing error (due to a faulty component in the NoC). Fig. 3 illustrates the challenge for such error detection. Apart from an increase of the data packet latency, the consequence of the nondetection of routing errors is the possible loss of data packets being sent either to an already detected faulty router or to an area performing a dynamic reconfiguration. In order to achieve routing error detection, the proposed reliable router relies on diagonal state indications, on additional routing information in the header flits, and on the routing error detection blocks in each port (see Fig. 2). The basic concept of our approach is the following: Each router receiving a data packet checks the correctness of the routing decision made by the previous crossed switch. This routing error detection is performed in parallel after the DMC to ECC, as shown in Fig. 2. Consequently, this detection does not increase the data packet latency.



**Fig. 3. Illustration of the routing error detection problem (a) to distinguish a dynamic bypass (b) from a routing error and (c) to avoid a loss of data packets**

**IV. LOOPBACK MODULE**



A discussion on the limitations and the extensibility of the presented work is given in Section VII. Finally, conclusions and opinions on future works are given in Section VIII.

**II. BASIC CONCEPT OF THE RKT-SWITCH**

We propose a new reliable NoC-based communication approach called RKT-NoC. The RKT-NoC is a packet-switched network based on intelligent independent reliable routers called RKT-switches. The architecture of the RKT-switch is depicted in Fig. 2. The RKT-switch is characterized by its architecture having four directions (North, South, East, West) suitable for a 2-D mesh NoC. The PEs and IPs can be connected directly to any side of a router. Therefore, there is no specific connection port for a PE or IP. The proposed detection mechanisms can also be applied to NoCs using five-port routers with a local port dedicated to an IP. However, the major drawback of these architectures is when the local port has a permanent error and the IP connected to it is lost or needs to be dynamically moved in the chip because of the dynamic partial reconfiguration. On the contrary, for the four-port RKT-NoC, an IP can replace several routers by having several input ports and hence be strongly connected in the network [5]. Moreover, by using dynamic partial reconfiguration and IPs strongly connected in the NoC, no one fault location is more catastrophic than another. Indeed, an IP may have access to the

### A. Basic Principles

In a dynamic reconfigurable NoC, the position and the number of components in the network can change during operation, as illustrated in Fig. 1. Actually, the number and position of the PE and IP in the NoC can be dynamically modified in order to meet the requirements of the application. Partial reconfigurable regions (PRRs) must be defined inside the FPGA in order to achieve dynamic reconfiguration of the 2-D mesh NoC. These PRRs are the regions where partial reconfigurable modules (PRMs) can be implemented. PRMs represent electronic instantiations of functional units. They are defined by specific partial bitstreams and can be placed according to the application needs. In practice, these PRMs correspond to the PEs and IPs being implemented and placed inside the dynamic NoC, as illustrated in Fig. 1. In a reliable NoC, faulty routers are isolated at runtime during the network operations. Let us consider a permanent faulty router that cannot be corrected. This router is permanently disabled. Similarly, during the reconfiguration of a PRR, no packet can be sent inside the area being reconfigured. Thus, these PRRs are dynamically isolated. However, these isolations can lead to data packet losses or increase packet transmission latency. More precisely, these drawbacks occur when routers containing data packets in their output buffers have their neighboring nodes unavailable due to a dynamic reconfiguration or permanent fault detection. Thereby, these data packets remain stored in the output routers until the end of the reconfiguration (dynamic implementation case) or are lost, in the case of detection of a permanent faulty node. To overcome these drawbacks, the proposed RKT-switch contains output buffer blocks associated with loopback modules, as described in Fig. 2. The role of each loopback module is to empty the buffers of each output port by looping back the data packets in the input port of the router. The result is that the looped back packets are rerouted towards another output port of the router. This avoids data packets becoming trapped when a neighboring switch is detected as permanently faulty, and reduces latency when a neighbor has suffered a dynamic reconfiguration. Fig. 6 illustrates the role of a loopback module. A PE or IP emitter sends data packets towards a destination IP according to the XY routing algorithm. If suddenly router(1, 3) becomes unavailable, the data packets remaining in the West output of router(2, 3) are looped back and rerouted towards its South output. This mechanism allows the stored data packets to be routed to the destination. Therefore, with router(1, 3) being marked as unavailable, the subsequent data packets coming from the East input port are routed directly towards the South port by the dynamic routing algorithm. Furthermore, the main advantage of the combined use of the proposed loopback module, the local historic of data errors, and the switch-to-switch data error detection mechanism (see Fig. 2) is the precise localization and distinction of the sources of data errors. Therefore, we can accurately locate whether the data errors are on the data bus, the input port, or output port, and whether the faults are permanent or transient.

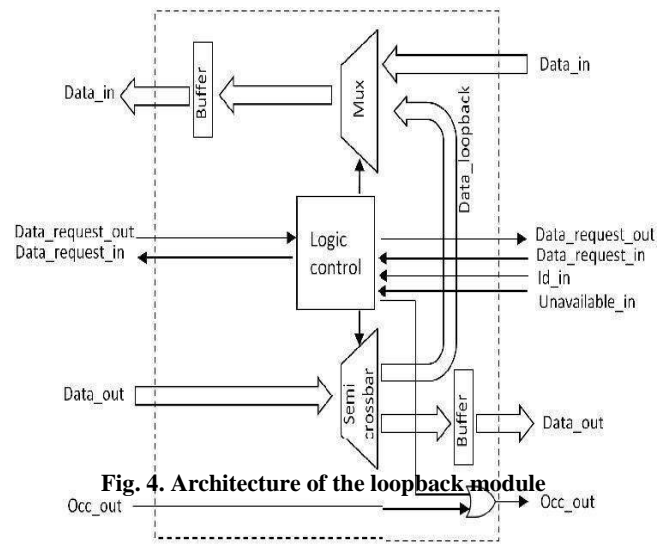


Fig. 4. Architecture of the loopback module

## V. PROPOSED SYSTEM

NoC using ECCs such as Hamming codes, Reed-Solomon codes and punctured difference set (PDS). However these codes needed more area, power and delay. Thus, we propose novel Decimal Matrix Code that ensures maximal error detection and correction. Rather than the conventional approach of detecting a faulty node and bypassing data through neighbouring routers, in DMC the fault is precisely detected and rectified, such that it enables router to be reused. DMC is based on decimal algorithms combining decimal integer addition and subtraction to identify and correct errors. The error detection capacity is enhanced and data reliability improved by using the decimal algorithm. To reduce the area overhead of additional circuits such as encoder and decoder, the encoder-reuse technique is used. This does not disturb the processing, however complicated computations may arise. The schematic of DMC Initially, for encoding the Data bits  $D$  are fed into the encoder followed by deriving the horizontal redundant bits  $H$  and vertical redundant bits  $V$ . After encoding process, the derived code is decoder. Initial ideas performed include divide symbol and arrange matrix,  $N$ -bit word is split into  $k$  symbols each  $m$  bits, and these symbols are organised into  $k1$  and  $k2$  ( $k = k1 \times k2$ ),  $k1$  and  $k2$  denotes numbers of rows and columns in the matrix. The horizontal redundant bits  $H$  are obtained through decimal integer addition, each symbol during the process is considered as decimal integer[4]. Binary operation is performed among bits per column to obtain vertical redundant bits  $V$ .

Consider 32-bit word scheme,  $D0$  to  $D31$  are information bits. The divide symbol concept is used and word is split into eight symbols of 4-bit.  $H0$ –  $H19$  are horizontal crosscheck bits and  $V0$  to  $V15$  are vertical crosscheck bits. It should be noted that depending upon the values chosen for  $k$  and  $m$ , the number of redundant bits and maximum error in memory. If a fault occurs, it can be detected using decoder. Initial ideas performed include divide symbol and arrange

matrix,  $N$ -bit word is split into  $k$  symbols each  $m$  bits, and these symbols are organised into  $k_1$  and  $k_2$  ( $k = k_1 \times k_2$ ),  $k_1$  and  $k_2$  denotes numbers of rows and columns in the matrix. The horizontal redundant bits  $H$  are obtained through decimal integer addition, each symbol during the process is considered as decimal integer. Binary operation is performed among bits per column to obtain vertical redundant bits  $V$ . Thus, to obtain refined results the values of  $k$  and  $m$  should be carefully chosen. By using decimal integer addition the horizontal redundant bits are derived as follows.

$$4\ 3\ 2\ 1\ 0 = 3\ 2\ 1\ 0 + 11\ 10\ 9\ 8\ (1)$$

$$9\ 8\ 7\ 6\ 5 = 7\ 6\ 5\ 4 + 15\ 14\ 13\ 12\ (2)$$

Correspondingly horizontal redundant bits  $H14H13H12H11H10$  and  $H19H18H17H16H15$  are also derived. The “+” indicates decimal integer addition.

The vertical redundant bits are represented as follows.

$$0 = 0 \oplus 16\ (3)$$

$$1 = 1 \oplus 17\ (4)$$

Both binary and decimal addition operations are performed from (1) to (4). The encoder comprising of multiple bit adders and XOR gates are as shown in .The bits  $U31 - U0$  are information bits extracted directly from  $D31$  to  $D0$ , the horizontal and vertical redundant bits are respectively represented by  $H0 - H19$  and  $V0 - V15$ .

## VI. SYNTHESIS RESULTS AND PERFORMANCE EVALUATIONS

### A. FPGA Synthesis Results

The results presented are obtained considering RKT-switches configured to process data packets of four flits and able to hold two data packets in each input buffer. Table III shows the synthesis results in terms of slices registers, slices LUTs, and maximal working frequency for different sizes of data bus and several FPGA technologies (Virtex V-VII Xilinx FPGA). It can be seen that the 32-b RKT-switch requires 4340 registers and 6542 LUTs and can operate up to 459.6 MHz on the Virtex VII FPGA technology. We have also synthesized RKT-NoC for several sizes on the Xilinx Virtex VI technology. These results are given in Table IV. The synthesis results clearly show that our architecture can be efficiently implemented in FPGA technology. It can be stated that an attractive tradeoff between high speed and logic resources has been achieved.

### B. Performance Evaluation

1) *Flit Injection Rate*: The packet injection rate (PIR) is the number of data packets that can be sent in a single clock cycle. For instance, an IP having a PIR of 0.5 means that it can send 50 data packets in 100 clock cycles. The flit injection rate (FIR) is the result of multiplying the PIR value by the number of flits in each data packet.

We have evaluated the  $FIR_{max}$  by simulating different NoC sizes 1

$\times 1$ ,  $3 \times 3$ , and  $4 \times 4$ . Each RKT-NoC is surrounded by the maximum number of communication modules: 4 mod-ules for the  $1 \times 1$ , 12 modules for the  $3 \times 3$ , and 16 modules.

for the  $4 \times 4$  NoC.  $FIR_{max}$  is obtained when the network is working without unavailable components, and when the modules are sending and receiving data packets only from and to the neighbor located at the opposite side of the network. Indeed, no case of router congestion can occur by using this traffic pattern.  $FIR_{max}$  has been estimated to be 0.369 for any RKT-NoC size and any number of IPs.

2) *Latency*: The latency of one RKT-switch ( $Latency_{RTRmin}$ ) is defined by 1. By using store-and-forward switching techniques, a data packet can be sent only when the entire packet has been received. Consequently, this technique adds a latency equivalent to the number of flits ( $N_{flit}$ ) in the data packet. The ECC, which is carried out in series, increases the latency by the number of clock cycles required. It is defined by  $Latency_{ECC}$  in 1. The RKT-switch has at least three clock cycles, one for the routing logic block and two to cross one loopback module when arriving and one when leaving the router. In the rest of this paper, we use data packets of four flits and the Hamming ECC takes two clock cycles to be performed. The  $Latency_{RTRmin}$  is then nine clock cycles.

$$Latency_{RTRmin} = N_{flit} + Latency_{ECC} + 3 \quad (1)$$

In an  $n \times n$  RKT-NoC, the minimal latency ( $latency_{min}$ ) to cross the network from source to destination is defined by 2. The number of switches crossed is  $N_{RKT}$ . Equation 2 takes into account the additional clock cycles for the Ack/Nack data flow control used in our reliable switches. This data flow control technique requires a latency of two clock cycles per router crossed.

$$Latency_{min} = N_{RKT} * latency_{RTRmin} + N_{RKT} * 2$$

The data packet latency depends on the traffic network. We have evaluated the average latency for  $1 \times 1$ ,  $3 \times 3$ , and  $4 \times 4$  RKT-NoC sizes. To evaluate the latency, we have simulated these NoCs surrounded by the maximum number of communication modules. Each module sends and receives data packets. The destinations for the data packets are generated randomly. The data packets are emitted at the maximum PIR. Table V gives the minimal and maximal average latencies for each NoC size in terms of clock cycles and time (ns).

## VII. CONCLUSION

In this paper, we proposed new error detection mechanisms for dynamic NoCs. The proposed routing error detection mechanisms allow the accurate localization of permanent faulty routing blocks in the network. They are suitable for adaptive routing algorithms based on  $XY$  where the main difficulty is to distinguish the bypasses of an unavailable component in the NoC (due to the use of the adaptive algorithm) from real routing errors (due to faulty components in the NoC). Validation simulations of our proposed routing error detection showed a routing error localization close to 96% for routing errors on an adaptive algorithm based on  $XY$  in a  $6 \times 6$  NoC. Regarding the proposed data packet error localization mechanisms, the simulations

presented in this paper clearly show the efficiency of our techniques, which can localize permanent sources of errors more accurately than the switch-to-switch or code-disjoint mechanisms. Moreover, both presented techniques can distinguish permanent and transient errors, and show attractive performance as presented in the FPGA synthesis comparisons with a nonreliable NoC. Our ongoing work focuses on evaluating accurately the impact of faulty detection blocks and improving the routing error detection mechanisms, by protecting the DAI links and routing detection blocks against errors.

#### REFERENCES

- [1] Cédric Killian, Camel Tanougast, Fabrice Monteiro, and Abbas Dandache, "Smart Reliable Network-on-Chip" *IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS*, vol. 22, no. 2, february 2014
- [2] J. Shen and P. Hsiung, *Dynamic Reconfigurable Network-on-Chip Design: Innovations for Computational Processing and Communication*, J. Shen and P. Hsiung, Eds. Hershey, PA, USA: IGI Global, 2010.
- [3] G.-M. Chiu, "The odd-even turn model for adaptive routing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 11, no. 7, pp. 729–738, Jul. 2000.
- [4] Y. M. Boura and C. R. Das, "Efficient fully adaptive wormhole routing in n-dimensional meshes," in *Proc. 14th Int. Conf. Distrib. Comput. Syst.*, Jun. 1994, pp. 589–596.
- [5] C. Bobda, A. Ahmadinia, M. Majer, J. Teich, S. Fekete, and J. van der Veen, "DyNoC: A dynamic infrastructure for communication in dynamically reconfigurable devices," in *Proc. Int. Conf. Field Program. Logic Appl.*, Aug. 2005, pp. 153–158.
- [6] T. Pionteck, R. Koch, and C. Albrecht, "Applying partial reconfiguration to networks-on-chip," in *Proc. Field Program. Logic Appl. Int. Conf.*, Aug. 2006, pp. 1–6.
- [7] S. Jovanovic, C. Tanougast, and S. Weber, "A new high-performance scalable dynamic interconnection for fpga-based reconfigurable systems," in *Proc. Int. Conf. Appl.-Specific Syst., Archit. Process.*, Jul. 2008, PP.61-66.
- [8] S. Jovanovic, C. Tanougast, C. Bobda, and S. Weber, "CuNoC: A dynamic scalable communication structure for dynamically reconfigurable FPGAs," *Microprocess. Microsyst.*, vol. 33, no. 1, pp. 24–36, Feb. 2009.
- [9] P. Lysaght and J. Dunlop, "Dynamic reconfiguration of FPGAs," in *Proc. Int. Workshop Field Program. Logic Appl. More FPGAs*. 1994, PP.82–94.
- [10] J. Wu, "A fault-tolerant and deadlock-free routing protocol in 2d meshes based on odd-even turn model," *IEEE Trans. Comput.*, vol. 52, no. 9, PP.1154–1169, Sep. 2003.
- [11] D. Park, C. Nicopoulos, J. Kim, N. Vijaykrishnan, and C. Das, "Exploring fault-tolerant network-on-chip architectures," in *Proc. Int. Conf. Depend. Syst. Netw.*, Jun. 2006, pp. 93–104.
- [12] S. Jovanovic, C. Tanougast, S. Weber, and C. Bobda, "A new deadlock-free fault-tolerant routing algorithm for NoC interconnections," in *Proc. Int. Conf. Field Program. Logic Appl.*, Aug.–Sep. 2009, PP.326–331.
- [13] D. Fick, A. DeOrio, G. Chen, V. Bertacco, D. Sylvester, and D. Blaauw, "A highly resilient routing algorithm for fault-tolerant NoCs," in *Proc. Design, Autom. Test. Eur. Conf. Exhibit.*, Apr. 2009, pp. 21–26.
- [14] W. Dally and C. Seitz, "Deadlock-free message routing in multiprocessor interconnection networks," *IEEE Trans. Comput.*, vol. C-36, no. 5, pp.547–553, May 1987.
- [15] M. Hosseinabady, M. Kakoei, J. Mathew, and D. Pradhan, "Low latency and energy efficient scalable architecture for massive NoCs using generalized de Bruijn graph," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 19, no. 8, pp. 1469–1480, Aug. 2011.
- [16] C. Grecu, L. Anghel, P. Pande, A. Ivanov, and R. Saleh, "Essential fault-tolerance metrics for NoC infrastructures," in *Proc. Int. On-Line Test. Symp.*, 2007, pp. 37–42.
- [17] A. P. Frantz, L. Carro, E. Cota, and F. L. Kastensmidt, "Evaluating SEU and crosstalk effects in network-on-chip routers," in *Proc. 12th IEEE Int. Symp. On-Line Test.*, Jul. 2006, pp. 191–192.
- [18] C. Grecu, A. Ivanov, R. Saleh, E. Sogomonyan, and P. Pande, "On-line fault detection and location for NoC interconnects," in *Proc. 12th IEEE Int. On-Line Test. Symp.*, Jul. 2006, pp. 145–150.
- [19] N. Karimi, A. Alaghi, M. Sedghi, and Z. Navabi, "Online network-on-chip switch fault detection and diagnosis using functional switch faults," *J. Universal Comput. Sci.*, vol. 14, no. 22, pp. 3716–3736, 2008.
- [20] M. Majer, C. Bobda, A. Ahmadinia, and J. Teich, "Packet routing in dynamically changing networks on chip," in *Proc. 19th IEEE Int. Parallel Distrib. Process. Symp.*, Apr. 2005, p.p. 154.