ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

# Dynamic Router Design For Reliable Communication In Noc

Mr. G.Kumaran[1], Ms. S.Gokila, M.E., [2]

VLSI Design, Electronics and Comm. Department, Pavai College of Technology, Pachal, Namakkal District, India[1]

Assistant Professor, VLSI Design, Pavai College of Technology, Pachal, Namakkal District, India[2]

**ABSTRACT:** Technological evolution enables the integration of billions of transistors on a chip. As VLSI technology scales, and processing power continues to improve, inter-processor communication becomes a performance bottleneck. On-chip networks have been widely proposed as the interconnect fabric for high performance SoCs. Recently, NoC architectures are emerging as the candidate for highly scalable, reliable, and modular on-chip communication infrastructure platform. New network-on-chip (NoC) that handles accurate localizations of the faulty parts of the NoC. The proposed NoC is based on store and forward technique, loop back mechanism. 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 mechanism 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, while preserving the throughput, the network load, and the data packet latency. To provide localization capacity analysis of the presented mechanisms, NoC performance evaluations, and field-programmable gate array synthesis.

**KEYWORDS:** Error Correction Code, Virtual Channel, LoopBack.

## 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.. 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.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. Furthermore, faulty nodes or even faulty regions make communications within the networks harder and even impossible with some routing algorithms. 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.

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 nonspecified 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. This adaptive algorithm is livelock- and deadlock-free and allows data packets to pass around faulty regions. 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.

NOC Framework Flow

This framework explains the control and data flow sequences followed in the transfer of messages between PEs. It first explains the sequence of operations followed when two PEs communicate with each other. Chapter also clearly explains how the underlying architecture modifies itself for different switching mechanism, Routing algorithms and modes of Operation (with and without Handshaking). A packet flow can be initiated either by Traffic Generator or by the master PE. TG can be used to override/mask network conditions or signals from PE. There are mainly three types of message transfers in a network. i) Master PE sending a message/data to slave PE ii) Master PE requesting data from slave PE and iii) Master to Master communication. Figure 1 shows the Framework and Routing Node with its associated components.

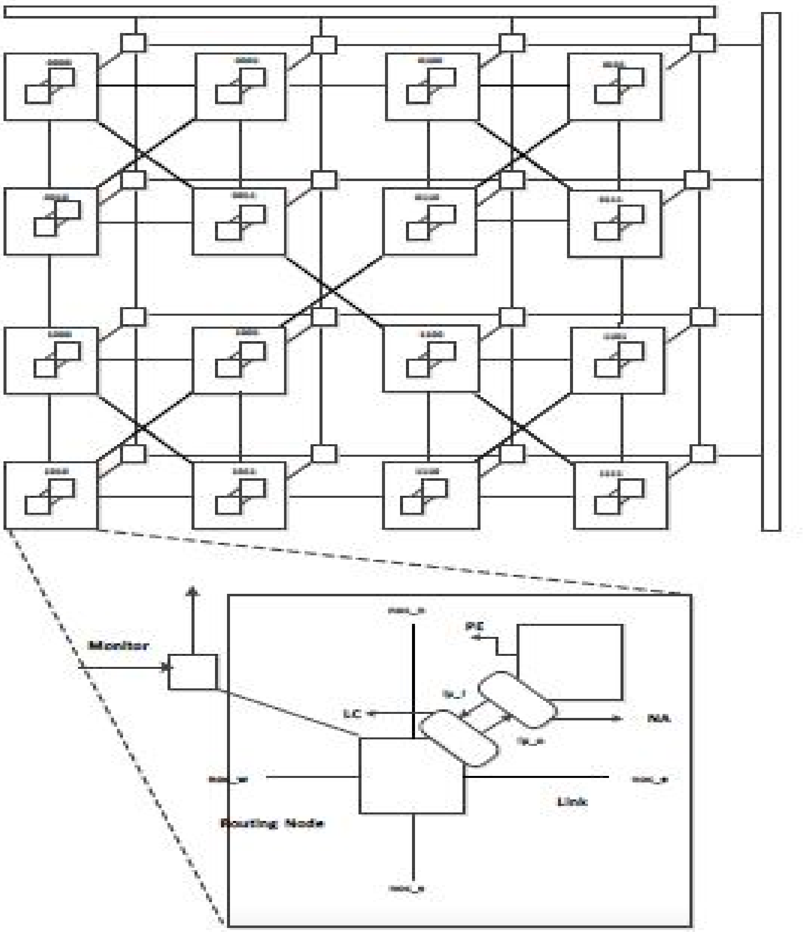

Figure.1. NoC Framework and Routing Node

## II. THE RKT-SWITCH

The proposed 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.1. 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 RKTNoC an IP can replace several routers by having several input ports and hence be strongly connected in the network. 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 network by being connected to several routers, or can be dynamically moved on the chip if this only access point becomes faulty.

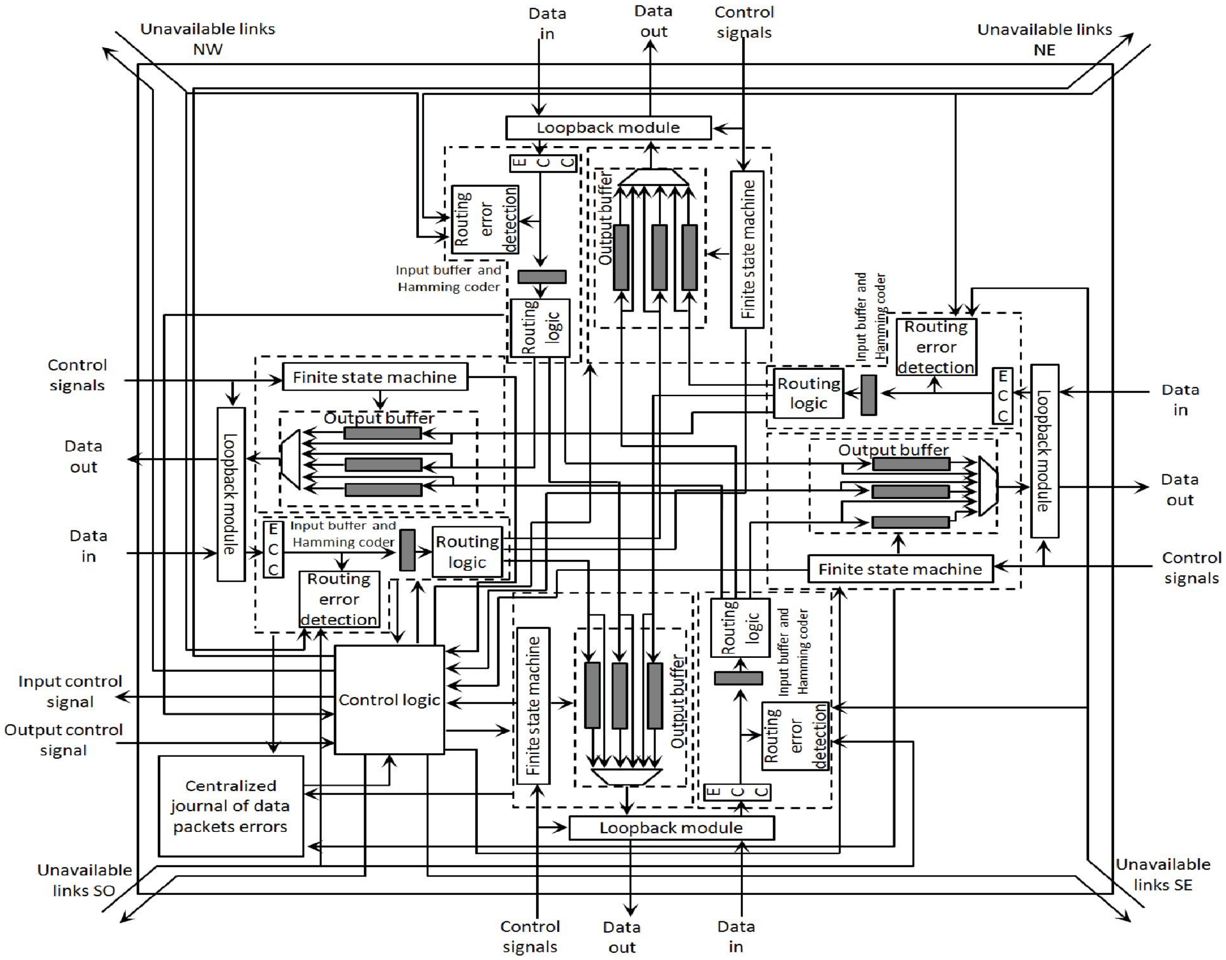

Fig.2. Architecture of the reliable router RKT-switch.

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. The RKTNoC uses nonbouncing routers, so that if a router is surrounded by three unavailable neighbors, it also becomes unavailable. Indeed, if a data packet is sent to a router surrounded by three unavailable nodes, the packet cannot be routed.

The data flow control used in our architecture is the Ack/Nack solution, which can handle fault-tolerant transmissions, although this does increase the energy consumption. This solution relies on the retransmission of packets being received as faulty by a neighboring node. Being able to perform a packet retransmission after it has been sent to a node requires that a copy of the packet be locally saved until an Ack or Nack is received. If a neighboring router receives a flit containing an error that cannot be corrected by the ECC, a Nack is sent back and the whole packet is retransmitted. Otherwise, an Ack is generated at full packet reception. More precisely, an Ack is generated only when all the flits of the data packet have been received and checked by the router, which reduces latency.

## 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:

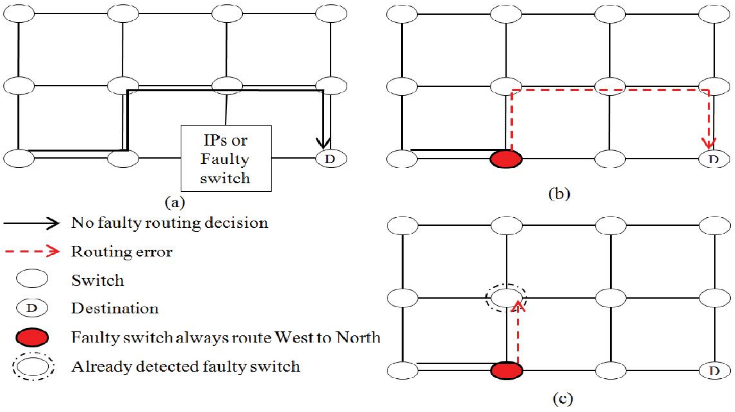
IPs or Faulty switch
D
(a)
(b)
(c)
No faulty routing decision
Routing error
Switch
Destination
Faulty switch always route West to North
Already detected faulty switch

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

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 Hamming ECC. Consequently, this detection does not increase the data packet latency.

A. Elements Required for Routing Error Detection

1) Diagonal Availability Indications: The RKT-switch uses information links to indicate to its neighbors its availability status. We define as unavailable an input port that cannot receive data packets. To preserve the highest throughput of the

NoC, our strategy is to disconnect only the faulty parts of the routers. Thereby, if a router input port is permanently faulty, it is disabled while maintaining the other input ports as active, in order to obtain a partially operating switch. On the contrary, if all input ports are faulty, the router is considered as unavailable. Similarly, we define as unavailable NoC components that cannot receive data packets due to permanent faults or a partial dynamic reconfiguration.

2) Routing Error Localizations: Each routing error detection block of the router inputs owns three journals to keep the routing error detection results. These journals are related to the routing logic blocks of the neighboring router connected to the considered input port. For example, in the West routing error detection block of router (i, j) has three journals corresponding to the West, North, and South routing blocks of the router (i−1, j). In addition, the location of the faulty routing algorithm blocks in the neighboring routers can be deduced from these journals. A permanent error is considered when three successive routing errors are detected for a specific routing logic block.

3) Structure of Information Fields in the Data Packets: A sliding gather data (SGD) field is added to each header flit of the data packets being transmitted. Table I details the structure of a data packet. A flit-type bit is used to distinguish the header from the data flits. The SGD field contains the addresses of the previous and penultimate crossed routers. Each router receiving a data packet checks this SGD field and validates the routing choice made by the previous router. To achieve the routing validation, the SGD field is updated by each router crossed along the transmission path. This update is done by each input buffer block. This requires an update of the Hamming code because of the modification of the header flit of the data packets. A unique routing path indication (URPI) bit is added to the header of the data packets. This bit is set if a router has only a single routing output path available.

## IV. LOOPBACK VIRTUAL CHANNEL MODULE

### A. Basic Principles

In a dynamic reconfigurable NoC, the position and the number of components in the network can change during operation. 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 bit streams 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.

(i)Virtual Channel with Loopback

The same issue can happen in SF routing where only 1 buffer is available at each input port. To avoid head-on-blocking in SF routing, we use loop backed VCs. In presence of congestion, the packet from VC is routed back to the VC identifier logic to route it in a later point of time. The implementation consists of 2 Virtual Channel.Whenever a packet reaches the VC identifier, it checks for the rate of occupancy in both Channels and allocate the packet to the least used. The switch allocator keeps track of previous pop and sends the packet to arbiter in a round robin fashion. When the receiving channel is busy, indicating congestion, then the packet routed from SA, is routed back to the VC identifier. In cases where every outgoing channel is busy, the Switch Allocator will wait for VC ready signal to toggle. When any of the channels becomes available, SA pops every available packet once. This is the worst case scenario.

This helps in reducing power due to unnecessary loopback and also in routing packets whenever the resource is available, increasing throughput.

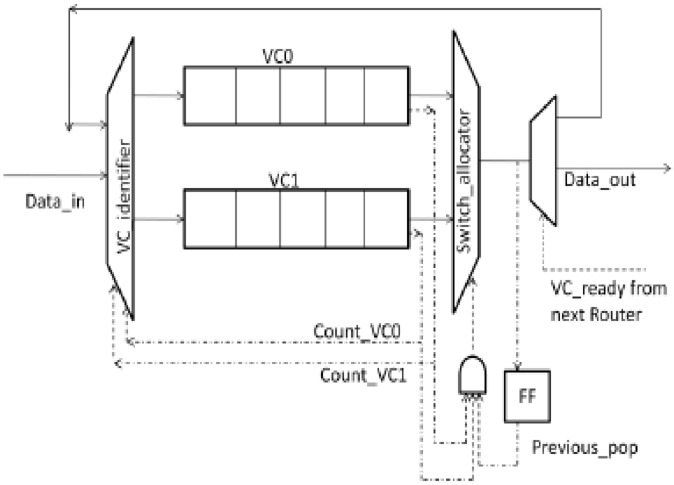

Fig. 5. Figure 25. Loopback VC Top – Implementation details

(ii) Modifications to SF Routing - VC AND LOOPBACK VC

This section provides an overview of the architecture of a generic Virtual Channel (VC) router, the implemented for Wormhole switching. It also detail the implementation of loopback VC, a novel architecture implemented in our NoC system for Store and Forward Switching.

(iii) Overview of a Virtual Channel Router

Wormhole routing is prone to congestion and deadlocks. In 1987, Dally and Seitz introduced the idea of Virtual Channel to develop deadlock free routing algorithms of networks that use wormhole routing [52]. In his paper [51] he has described a theoretical model for networks using virtual channel flow control and has presented the results of his simulation studies. These results indicate a dramatic increase in network throughput due to the use of Virtual Channels. Consider a scenario where in the packet at the top of FIFO has to be routed to east, but the east port is blocked due to congestion. This is head on blocking. To reduce congestion and avoid head on blocking every physical channel is divided into Virtual Channels 55 (VC). Every input port except the IP port has 4 virtual channels. The IP port has a single FIFO. However too many VCs could cause a packet to spread across many routers, which increases its latency in the network. When a new packet reaches an input port of a router, an attempt is made to allocate an unused VC for the new packet. In cases where all the VCs are occupied, the VC identifier module checks the buffer count of each VC and directs the incoming packet to the lowest occupied FIFO. When more than one VC in a input port is occupied, the Switch Identifier module determines the next packet to be routed in a round robin manner. The packet is then routed to the output port based on the routing signals obtained from NoC Arbiter. In our design, once a portion of a packet occupies a VC queue, other flits of same packet are not allowed to use only the same VC flit slots. In other words the VC Identifier looks for header and determines the VC. Once a VC is identified for a particular packet, the entire packet uses the same VC. This avoids wastage of VC resources if a packet gets blocked before reaching destination. Assume there are six incoming packets P0, P1, P2, P3, P4, P5, P6 and P7 with designated output ports East, East, East, East, West, West and South respectively in current router. Consider the condition when East port is busy now. The Figure 5 shows the scenario where in P0, P1, P2, P3, occupying the head of

VC0 to VC3 respectively cannot move forward hence blocking packets P5 and P6 to the West port. This is a head on blocking scenario.

B. Localization of Data Packet Errors

To locate and distinguish permanent and transient errors, a local historic of data packet errors is implemented locally in each router, as described in Fig. 2. This block is composed of journals related to the input and output ports. These journals are 3-bit-deep shift registers. The RKT-switch uses the Ack/Nack data flow control. When a data packet is transmitted to a neighboring node, a copy of the data is stored locally until the Ack is received. If no error occurred during the transmission (reception of an Ack), a set to "0" is added to the journal related to the input port of the get-in direction and the output port. If an uncorrectable error is detected by the neighbor, a retransmission is performed in response to a Nack. If three Nacks are received, the packet is looped back and a set to "1" is added into the journal related to the input and output ports taken by the data packet. Indeed, the error source can be located on the bus, in the input port, or in the output port. After going through the loopback, the data packet is checked by the input ECC. If an uncorrectable error is detected by the ECC, the data packet is destroyed. If no error is detected, we can conclude the errors detected by the neighbor occurred on the data bus. If the error occurred consecutively three times on the bus (i.e., three Nacks), we can conclude that there is a permanent error on the data bus. The local historic has a threshold before disconnecting a part of a router. This threshold is the number of consecutive errors required to flag an error source as permanent. Here, To set a threshold of 3 (see Section VI for more details on the impact of the threshold). When three consecutive errors occur on the same journal related to an input or output, the local historic of data packet errors concludes that a permanent error exists in the related direction.

Physical Channel Selection Rule

Suppose S and T are the source node and destination node in WKd,L, respectively. A routing path between them can be constructed as follows.

Step 1. Compare the first two bits of the node addresses of S and T. If they are same, the destination node is in same sub-network. If they are different,

Step 2. Determine the flipping edge. The flipping edge (X, Y) is the bridge between the two sub-networks.

Step 3. Determine the routing path from S to X, and the routing path from X to T. The routing path from S to T is the concatenation of the routing path from S to X, the flipping edge (X, Y), and the routing path from Y to T.

Virtual channel selection rule

After selection of next physical channel according to the algorithm in previous section, suitable virtual channel must be selected in this physical channel by considering deadlock avoidance conditions. Each router node contains 4 channels at each output ports. Every packet arriving the port is routed to corresponding channels based on its next route. North/Over going packets to VC0, South/Across going packets to VC1, East going packets to VC2, and West going packets to VC3.

The data packets being looped back, after being checked by the ECC, are checked by the routing error detection block. However, the routing error detection block finds in the SGD field that the previous router address is its own address and deduces a loopback. Consequently, it does not apply the routing error detection algorithm. When a permanent fault is detected in a router, the faulty part of the NoC has to be isolated. The part to be isolated has been located accurately by using the local historic and the loopback with the switch-to-switch error detection mechanism. It can be located in the input port, the output port, or the data bus. If the error is in the input port, the router locally activates the horizontal availability link of the faulty input port, and the two associated DAI links. In this way, the neighboring component connected to the faulty port cannot send new data packets in this direction, and the DAI flags indicate to the diagonal neighbors the possibility to bypass its position. If the error is on the data bus or in the output port, the router detecting the permanent error must indicate to the neighbor to activate its availability indications links.

## V. CONCLUSION AND FUTURE WORK

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 × 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 bor 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] K. Sekar, K. Lahiri, A. Raghunathan, and S. Dey, "Dynamically configurable bus topologies for high-performance on-chip communication," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 16, no. 10, pp. 1413–1426, Oct. 2008.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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

[7] P. Lysaght and J. Dunlop, "Dynamic reconfiguration of FPGAs," in Proc. Int. Workshop Field Program. Logic Appl. More FPGAs. 1994, pp. 82–94..

[8] 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.

BOOKS:

1. Innovations for computational processing and communication
2. Applying partial reconfiguration to networks-on-chip,
3. Essential fault tolerance metrics for noc infrastructures
4. Highly resilient routing algorithm for fault-tolerant
5. A new deadlock-free fault-tolerant routing algorithm

WEBSITES:

http://www.nocarc.com
http://www.nocsymposium.org
http://www.ocpip.org/uploads/documents/NoC-Benchmarks-WhitePaper-15.pdf
http://www.sigda.org/newsletter/2006/060415.txt