ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Monthly Peer Reviewed Open Access International e-Journal

Eliminating Error by Designing a Reliable Router by Effective Routing Algorithm for Dynamic NOCS

Palle Prasada Rao
M.Tech (VLSI System Design),
Dept. of E.C.E,
Sits,Kadapa.

P. Harinath Reddy
Asst Prof,
Dept. of E.C.E,
Sits,Kadapa.

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, while preserving the throughput, the network load, and the data packet latency. We provide localization capacity analysis of the presented mechanisms, NoC performance evaluations, and field programmable gate array synthesis.

Index Terms:

Adaptive algorithm, dynamic reconfiguration, network-on-chip (NoC), reliability.

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 [1].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.) [2].

The NoC medium features a high level of modularity, flexibility, and throughput. AnNoC 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 orfully adaptive routing algorithm), which is applied locally in each router being crossed and to each data packet [3], [4]. Dynamically reconfigurable 2-D mesh NoCs (DyNoC, CuNoC, QNoC, ConoChi, etc.) are suitable for fieldprogrammablegate array (FPGA)-based systems [2], [5]–[8].

Thanks to the partial dynamic reconfiguration of FPGAs [9] 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 goalis 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 [5].

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Monthly Peer Reviewed Open Access International e-Journal

Regarding adaptive or fault-tolerant routing algorithms, several solutions have been proposed [10], [11]. 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 [12], 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 [13].

These tables are updated by an initialization algorithm. The main drawback of this solution is the requirement to invoke the algorithm at a nonspecifiedtime 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 thefaulty 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 [3], [12], [14].

Another reliable routing algorithm solution is the use of the de Bruijn graph [15]. 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 [16]. These faults may generate data packet errors, or may affect router behavior leading to data packet losses or permanent routing errors [17]. Indeed, a fault in a routing

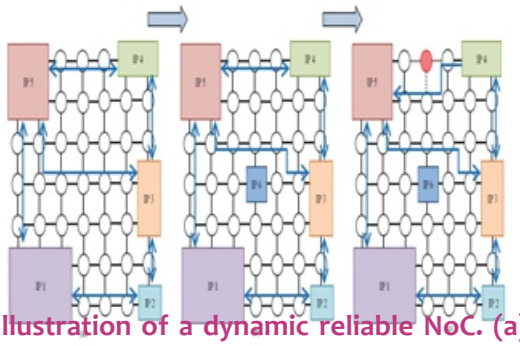

Fig. 1.Illustration of a dynamic reliable NoC. (a) Normal operation. (b) Dynamic implementation of an IP. (c) Online detection of a faulty router.

logic will often lead to packet routing errors and might-even crash the router. To detect these errors, specific errordetection blocks are required in the network to locate the faultysources. Moreover, permanent errors must be distinguishedfrom transient errors. Indeed, the precise location of permanentfaulty 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 theNoC [18]. 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-toswitchdetection 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, andWest), 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[18]. 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.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Monthly Peer Reviewed Open Access International e-Journal

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 [19], 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 thischecking 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 [12], 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 [3], [12], [14], [20]. Our approach includes datapacket error detection and correction. The originality of theproposed architecture is its ability to localize accurately errorsources, allowing the throughput and network load of the NoCto be maintained. In our case study, we consider a reliableapproach based on the adaptive routing module proximity algorithm [12]. The considered routing algorithm is based on the adaptive turn model routing scheme and the wellknownXY algorithm. This adaptive algorithm is livelock- and deadlock-free and allows data packets to pass around faultyregions.

II.BASIC CONCEPT OF THE RKT-SWITCH:

We propose a new reliable NoC-based communication approach called RKT-NoC. The RKT-NoC is a packetswitchednetwork based on intelligent independent reliable routers called RKT-switches. The architecture of the RKTswitchis 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 fiveportrouters with a local port dedicated to an IP. However, themajor 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 [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 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 [7]. At any instant with the storeand- 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 [14], 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.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Monthly Peer Reviewed Open Access International e-Journal

The RKTNoCuses nonbouncing routers [21], 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 [22], although this does increase the energy consumption [23].

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. The Hamming ECC is considered for our RKT-switch, in order to provide a convenient tradeoff between area overhead and error correction capacity.

This choice permits the correction of single event upset (SEU) errors (one bit flip in a flit) and the detection of multiple event upset (MEU) errors (two bit flips in a flit). Moreover, the Hamming code is more suitable for NoCs based on Ack/Nackflow control than the parity bit check. Indeed, on a single bit-flip error occurrence, error correction is possible with the Hamming ECC, whereas the single parity check would require packet retransmission and hence an increased transmission latency.

The distinction between permanent and transient errors is granted thanks to a local historic, which saves the transmission results, and a loopback output mechanism (see Section IV). Furthermore, our solution combined with the loopback mechanism and the novel local historic allows the localization of errors, either on the bus connections or inside the switches, by localizing the faulty port (more details in Section IV). In addition, our reliable structure is based on switch-to-switch detection, offering robustness against SEU and two-bit MEU errors, while maintaining a good tradeoff between area overhead and the capacity to locate errors.

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 [3], [12], [14], [20]. 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 alreadydetected 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.

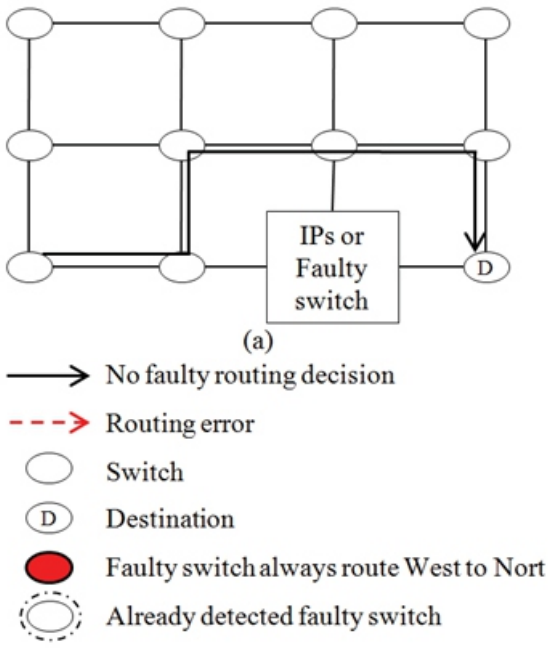

Fig. 3. Illustration of the routing error detection problem (a) to distinguish a dynamic bypass (b) from a routing error and (c) to avoid a loss of data packets.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Monthly Peer Reviewed Open Access International e-Journal

decision made by the previous crossed switch. This routing error detection is performed in parallel after the Hamming ECC, as shown in Fig. 2. 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 availabilitystatus. We define as unavailable an input port that cannotreceive 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 nas 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. The RKT-switch indicates its availability status to the eight direct neighboring routers through the diagonal availability indication (DAI) links. The network structure based on DAI links is shown in Fig. 4. These DAI links allow the checking of the correctness of the routing algorithm. Indeed, each router is able to control the availability status of the neighboring routers and components. For instance, in Fig. 4(a), router(i, j)can check the availability of router(i +1, j –1). The network components (PEs or IPs) are not allowed to route data packets and are restrained to accept only data packets intended for them; hence, their DAI interconnections are set.

2) Journal of 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 Fig. 4(a), 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). Thanks to these journals, the distinction between permanent and transient errors can be ensured.

In addition, thelocation 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:

Asliding gather data (SGD) field is added to each header flit ofthe data packets being transmitted. Table I details the structureof a data packet. A flit-type bit is used to distinguish the headerfrom the data flits. The SGD field contains the addresses of theprevious and penultimate crossed routers. Each router receivinga data packet checks this SGD field and validates the routingchoice made by the previous router. To achieve the routingvalidation, the SGD field is updated by each router crossedalong the transmission path. This update is done by eachinput buffer block. This requires an update of the Hammingcode because of the modification of the header flit of the datapackets. A unique routing path indication (URPI) bit is addedto the header of the data packets. This bit is set if a router hasonly a single routing output path available. This bit allows theavoidance of false detections. Our approach is then suitablefor a dynamic NoC-based fault-tolerant routing algorithm.

B.Illustration of the Routing Error Detection:

The RKT-switch allows online detection of routing faults and the distinguishing of routing errors from dynamic bypasses related to the adaptive routing algorithm. Several examples illustrate the efficiency of the proposed routing error detections. Fig. 4 shows two examples of data packets being routed from router(i – 1, j – 1) to router(i, j – 1). In Fig. 4(a), router(i, j – 1) is receiving a data packet from its West neighbor router. It checks the correctness of the routing decision made by router(i –1, j – 1). As router(i –1, j – 1)obeys the XY algorithm, no error is detected. In Fig. 4(b), router(i, j) is receiving a data packet from the South direction. The XY routing path between router(i, j – 1) and router(i + 1, j – 1) is deduced from the analysis of the SGD field of the data packet and of the penultimate and destination router addresses. Hence, router(i, j) detects that router(i, j–1)

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Monthly Peer Reviewed Open Access International e-Journal

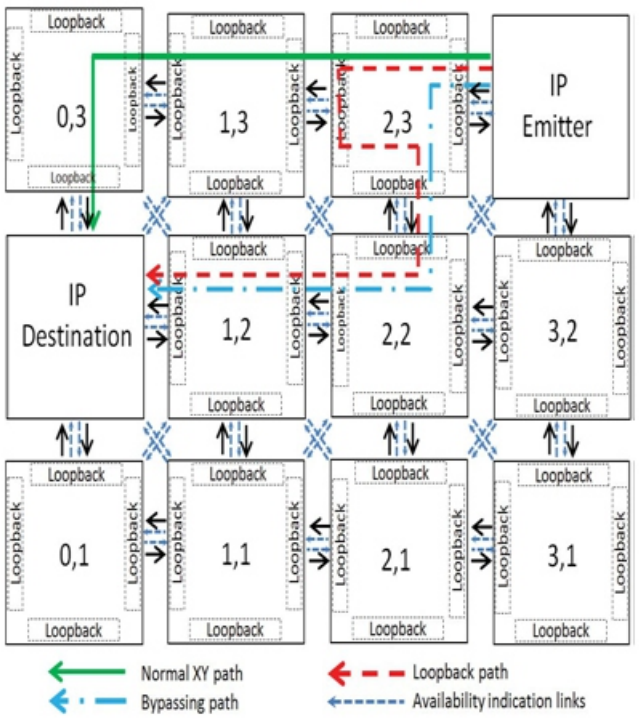

Fig. 5. Illustrations of a data packet loopback and a dynamic bypass decisionis not obeying the XY routing algorithm.

Consequently, router(i, j) checks the DAI links of router(i + 1, j – 1). The router(i + 1, j – 1) being unavailable means that router(i, j – 1) made a bypass decision and hence is not faulty [see Fig. 4(a)]. Otherwise, the router(i + 1, j – 1)is available, which means that router(i, j – 1)made a faulty routing decision [see Fig. 4(b)]. In Fig. 4(b), router(i, j) sets the error journal associated with the position of the routing logic block causing this routing error decision. This routing logic block is identified from the address of the penultimate node.

More precisely, the West routing logic block of the router(i – 1, j – 1) is identified by checking of the SGD field and the identification of the penultimate router router(i – 2, j – 2). Fig. 4(c) illustrates the role of the URPI bit. Here, router(i, j) is receiving a data packet from the North directionaccording to the XY routing algorithm. By checking the DAI of router(i + 1, j + 1), a bypass decision is deduced. However, the destination is located on the North direction compared with the previous router. According to the routing algorithm being used, the bypass should have been done by the North direction. The availability of router(i, j +2) is checked in the URPI in the SGD field.

Indeed, if the URPI is set, then router(i, j + 1) has only one remaining path, and hence did not made a routing error. On the contrary, if the URPI is not set, router(i, j +1) has two remaining available paths to route the data packets, and hence made a routing error.

C.Principle of the Routing Error Detection:

An XY-based adaptive routing algorithm primarily uses the rules of the XY algorithm [12] to route data packets into the network when the required components are available. In thecase of an unavailable component, a specific routing path is locally chosen to bypass its position. When a router receives a data packet, it checks the correctness made by the routingdecision of the previous node, using the routing error detection algorithm. From address comparisons, the router checks if the previous routing decision obeyed the XY routing algorithm. If

it is the case, then the previous decision is correct. Otherwise, the routerdecides whether the previous decision is a bypass decision or a routing error. The detection algorithm is required to check the availability of the router through which the data packets should have passed according to the XY algorithm. This verification is performed thanks to the DAI links. If the router in the XY path is unavailable, the previous router decision was a correct bypass. If it is available, the previous router decision is a routing error.

In the latter case, the router adds one “1” to the error journal associated with the faulty routing logic block. The position of the faulty block is deduced from the address of the penultimate router in the SGD field. If three consecutive errors are performed by the same faulty routing logic block, a permanent error is considered. In this situation, a specific data packet is generated towards the switch generating the routing errors.

This specific one-flit data packet indicates the faulty input port of the considered router that must be disconnected. For NoCs based on multiflit datapackets, it may happen that a flit is received without being preceded by a header flit, which is an erroneous situation. In the proposed RKT-NoC, there is a bit in each flit indicating whether the flit is a header flit or a data flit, as depicted in Table I. When a router receives the first flit of a data packet, it checks after the hamming decoding whether it is a header flit.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Monthly Peer Reviewed Open Access International e-Journal

If not, the flit is destroyed. Therefore, when receiving a data packet, the destination IP or PE counts the number of received flits. If this number does not match the number indicated in the header flit, the packet is destroyed and a retransmission request is sent back to the emitter IP or PE.

VALIDATION OF THE PROPOSED CONCEPTS:

A. Network Load:

We simulated the network load for a 4 × 4 RKT-NoCsurrounded by the maximum number of communication modules. Each communication module sent 10 000 data packets by using a random traffic pattern. Fig. 9(a) shows the network load for the NoC without any errors.

This result shows nonuniformly distributed traffic in the network. Most of the traffic is located on the edge of the NoC where the source and destination modules are connected. In Fig. 9(b), one input of a router has been disconnected from the network by simulating a permanent error. The faulty router is router(3, 2). The simulation result shows a reduction of the network load for the faulty routers. The load of router(3, 2) decreases from 4.39% to 3.53%.

The maximum increase of network load is obtained for router(2, 2), increasing from 4.78% to 7.09%. Fig. 9(c) shows the network load for a network containing one entirely faulty router. More precisely, the four input ports of router(3, 2) are disconnected from the network. The network load clearly increases, especially around the faulty switches. The largest network load increase is for router(2, 2), from 4.33% to 6.1%, which represents an increase of 40%.

These simulation results clearly show the interest of our error detection approach. By disconnecting accurately only the faulty parts of the NoC, we maintain the network load at a level similar to that of a network without fault. We clearly show that by disconnecting one entire router, when using onlythe switch-to-switch error detection mechanism, or using the code-disjoint mechanism in the case of an error inside the router, the network load is much altered, which can lead to the generation of network congestion.

B. Fault Injection Method:

The following simulations have been realized in the Model- Sim\ environment through a C-VHDL co-simulation [26]. We assume nonfaulty detection blocks. The RTL design of the NoC is modified to generate the errors. More precisely, the input buffers, output buffers, data bus, and routing logic of all the routers have a special input to activate errors.

Furthermore, the position of the errors in the data is specified by using a mask input. Similarly, the routing blocks have an input to force a routing decision. For instance, we can force a routing block to always route to the North direction. These inputs are activated by a specific IP modeled in C language, which generates randomly the error positions. Each NoC router has the same probability to get an error, and the positions of the faulty bits are random.

D.Network Robustness Against Transient Data Packet Errors:

We have performed an analysis of the NoC against several SEU (bit flips) frequencies. To evaluate the robustness of the network, we simulated a 6 × 6 RKT-NoC connected to 24 IPs. Each IP injects 100 000 data packets at the maximum PIR by using a random traffic pattern. Each packet contains four Fig. 10. Throughput of a 6 × 6 RKT-NoC for several SEU (bit flips) frequencies.flits of 64 b. The SEU locations in the network are random. The throughput and the data packets latency of the network are shown in Figs. 10 and 11, respectively.

The throughput is almost constant at 116.5 Gbit/s until an SEU rate of 0.1 (one bit flip every 10 clock cycles). After this value, the throughput decreases as the SEU rate increases. The latency of the data packets is shown in Fig. 11, which increases linearly until an SEU rate of 0.1. After this value, the latency increases more rapidly. These results can be correlated with Fig. 12, which shows the number of data packets lost for several SEU injection rates. We can clearly see that the number of packets lost increases from the SEU rate of 0.1. Thus, thedegradation of the NoCperformance is linked to the number of packets lost. Indeed, with uncorrectable errors, the number of retransmissions due to the Ack/ Nack data flow control.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Monthly Peer Reviewed Open Access International e-Journal

increases the data packet latency and decreases throughput. It can be seen that for an SEU rate of 0.2, only 81 data packets have been lost during the transmission of 2 400 000 data packets.

D. Evaluation of the Data Packets Error Localization Capacity:

We propose an analysis of the capacity to locate the error sources regarding permanent data packet errors. This localization capacity is given for a 6 × 6 RKT-NoC using random traffic. For this analysis, 3000 simulation cases have been performed. In each simulation, the position of the permanent errors is random and the errors simulated are 2 b stuck at "1." The results are given in Table VIII.

In this table, we can clearly see all the permanent errors are accurately localized. Regarding the case of permanent errors located on the data bus, the faulty data bus is always localized. Moreover, when localizing the data error sources on the data bus, no data packets are lost by using the loopback module. Regarding the cases where error sources are in the input or output buffer; in 100% of these cases, the faulty port is accurately disconnected. The impact of the threshold (number of uncorrectable errors detected before activating the flags of permanent errors) of the journals in the local historic is shown in these results.

LIMITATIONS AND EXTENSIBILITY:

We assumed nonfaulty correction/detection blocks and DAI through the presented simulations and estimations. If this hypothesis is false, the faults can generate two error cases. The first disables the detection capacity of the block, which cannot detect any errors. The second is the generation of false detection. More precisely, a detection block generating false detection finds an error even when there is none.

For both cases, if the errors are transitory, their effects will not affect the NoC. Indeed, the routers check journals where an error needs to occur consecutively three times before a faulty part is disabled. Regarding the permanent faults generating the errordetection incapacity, the data packets error detection relies on the subsidiarity of the NoCelements.

More precisely, if an ECC does not detect an error in a data packet, this faulty packet is sent to a neighbor. This neighbor will detect the error and generate a Nack. After three Nacks, the data packet is looped back to the router that did not detect the error.

After the loopback, the ECC and the local historic will locate the port in which the error was not detected and the router will isolate the input port including this faulty block. The permanent faults generating error detection incapacity (fault in the detectionblock or the DAI) in routing error detection cannot be detected in the NoC. However, standard solutions of fault tolerance can be applied to the routing error detection blocks and the DAI, like the duplication of the blocks/links [27].

Regarding the permanent false detections, because the presented mechanism can localize accurately the error sources, these error cases will only disconnect small parts of healthy routers, which does not critically affect the NoC, as shown in thesimulations in Section VI-A. The presented error detection mechanisms have been detailed for a router based on the storeand- forward switching technique. However, our mechanisms are also suitable for virtual cut-through and wormhole routing. Indeed, the routing error detection as presented in the RKT-NoC can be used with these switching policies without any modification.

Regarding the data packet errors based on the use of the loopback module, routers using wormhole routing need some modification. More precisely, with wormhole techniques, a data packet can be spread over several routers. Each output block receiving a header flit needs to store locally its routing information.

When a flit fails to be transmitted to a neighbor (i.e., the router receives three Nacks), the packet needs to be looped back. A header flit is generated locallywith the information saved in the output buffer, and the flits that were not transmitted to the neighbor are looped back and analyzed by the ECC located in the input port, as detailed in Section IV-B. The flit already transmitted to the neighbor continues to be routed towards the destination. Thus, this destinationwill receive the data packet in two parts, each having the same header flit, which will permit the reconstitution of the data packet.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Monthly Peer Reviewed Open Access International e-Journal

VIII. 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 × 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-toswitch or code-disjoint mechanisms. Moreover, both presented techniques can distinguish permanent and transient errors, and show attractive performance as presented in the FPGAsynthesis comparisons with a nonreliableNoC.

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] 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.