The CSI Journal on Computer Science & Engineering

The CSI Journal on
Computer Science and Engineering
Vol. 12, No. 2 & 4 (b), 2014
Pages 1-10
Regular Paper

RSFR: A Recursive Self-Testable and Fault-Tolerant Routing Protocol for NoC Routers

Sanaz Alamian Ramin Fallahzadeh Shaahin Hessabi

Department of Computer Engineering, Sharif University of Technology, Tehran, Iran

Abstract

This paper proposes a novel routing algorithm and test strategy which provides deadlock-freedom, low packet dropping and low latency in networks-on-chip (NoCs).We first propose an algorithm, in which functionality of a router is revised in order to enable it to test its own, as well as the preceding router's, functionality. The recognized faults are reported to successive routers along the packet's path to the destination without the need for generating any extra non-data packet. Furthermore, we enrich the initial scheme by adding several features such as ability to backtrack the misled packets, prediction of the safest path to destination, and clearance of transient faults. We show that the enriched approach, namely RSFR, demonstrates a significant performance improvement in terms of latency, drop-rate and fault coverage while retaining deadlock-freedom. The experimental results illustrate that fault coverage for routers can reach up to almost 100% with yet low power consumption.

Keywords: NoC Routers, Fault-Tolerant NoC, Functional Test, Routing Algorithm, Latency, Drop-Rate.

1. Introduction

SYSTEM-ON-CHIP (SoC) consists of large number of embedded processor cores. This fact makes traditional bus structures unsuitable as a communication means for these cores. Network-on-chip (NoC), which has been proposed for communication among cores in SoCs to provide the high performance and power requirements, has created new challenging issues. One of these challenging issues is testing the network structure and its functionality during transmission of the data packets. SoC testing consists of testing cores, as well as the communication fabric (network). Testing cores can be accomplished by using NoC as Test Access Mechanism (TAM), so that the area overhead and test time can be reduced. Therefore, for the purpose of reliability of the system, it is important to ensure that the communication fabric (NoC) is working as expected.

Various test methodologies [1-7] are utilized to test different parts of the network infrastructure. These methods are classified into two major classes; the first class, i.e. structural testing, contains scan chains or built-in self-test (BIST) architectures. These methods detect stuck-at faults by examining the structure of network. In the second class, the test methods aim to examine the functionality of network, to certify the correctness of routing and delivering data packets. The main targets of these methodologies [12, 14] are high level routing and switching faults. Designing a NoC structure requires consideration of various tradeoffs regarding throughput, latency, silicon area, power consumption and reliability [2]. Upon the requirement of testing NoC, various challenges such as fault coverage, ability to detect and locate faults, and designing a fault-tolerant network are arisen.

In this paper, we focus on high level routing and switching faults in functional and online testing method in order to test the combinational parts of switches, and make the network-on-chip fault-tolerant and testable with high fault coverage. We present an adaptive, self-testable, deadlock-free and fault-tolerant routing algorithm. The ability to discriminate the exact location of faulty routers, during the normal operation (without the need of generating

any extra non-data packet) is the main contribution of this paper.

The rest of the paper is organized as follows. Some prior related work will be discussed in Section 2. Section 3 provides some preliminary background. Sections 4 and 5present the initial and enriched proposed algorithms, respectively. The simulation model and metrics, simulation results and corresponding analysis will be illustrated in Section 6. Section 7explains future direction, and finally, Section 8presents conclusions.

2. Related Work

A wide variety of standard Design-for-Test (DFT) and structural techniques have been presented for testing NoC-based designs [3]. Amory et al. [4] have presented a partial scan-based test strategy in order to test the router and its buffers. It utilizes two scan chains per router, and a test wrapper for the whole NoC. Hosseinabady et al. [5] have proposed a scan-based concurrent test for switches. A switch, as a test source, generates test vectors, which are broadcast through the scan chain that contains NoC switches. Furthermore, in [6], they have indicated that this test methodology can be broadened to support different switch structures in a NoC. In [7], a test wrapper is proposed to identify the location of damaged and failed routers. Grecu et al. [8] utilize BIST for NoC interconnect infrastructures. Their method uses a high-level fault model to deal with crosstalk fault in inter-switch links based on Maximal Aggressor Fault (MAF) model.

While these methods test NoC interconnection network structurally, there are other test strategies which aim to functionally test the interconnection network. In [1], a functional-based test method is presented for testing the interconnects and routers in a scalable manner, by breaking the network into smaller 2×2 networks, and implementing various test rounds in order to test all 2×2 networks in the main network. Zheng et al. [9] have proposed a functional method for testing switches. They first test boundary switches, and use fault-free switches as throughway switches to transfer test vectors to the central switches for testing them, as well. In this method, the straight paths and turn paths are tested. Alaghi et al. [2] have presented an online NoC switch fault detection and diagnosis based on high-level fault models in routing.

The main idea in [2] is to transform regular switches into self-testable ones, to test whether they have routed data packets according to the specified routing algorithm. Grecu et al. [10] have proposed a strategy for online fault detection and diagnosis for NoC communication fabric by using parity checkers at input and output ports for data fault models. DeOrio et al. [11] propose a table-based fault-tolerant routing algorithm. Each router selects the direction, based on its updated entry in the table, and applies certain rules to avoid deadlock.

A novel fully-adaptive routing algorithm, consisting of four developed algorithms combined with turn models for creating fault-tolerant deadlock-free routing algorithms, has been presented in [12]. Since XY routing algorithm prohibits lots of turns, and therefore decreases potential adaptability, in [12] each of the proposed routing algorithms prohibits only two turns in order to be deadlock-free and fault-tolerant. Another fault-tolerant routing approach, i.e. an automatic re-routing extension to the packet connected circuit for NoCs, has been introduced in [13]. In [14], another deadlock-free routing algorithm has been presented. No virtual channel has been used, and there is no dropped packet. However, if there are more than two faulty channels in network, deadlock will occur and it will not be fault-tolerant [14]. In this routing algorithm, the data packet will be routed via minimal route if it is fault-free. Otherwise, it will take all the shortest routes around the faulty channels that do not cause deadlock or livelock. A reconfigurable routing method for fault-tolerant Mesh-based NoC has been inaugurated in [15]. In this method, faults are detected and diagnosed by usual structural methods such as BIST. After detection, BIST mechanism updates the configuration 8-bit register of each router by a fault detection process. This method works correctly in the case of two faulty routers; but it will partially work in presence of more than two faulty routers in each contour. Wachter et al. [16] propose a backtrack-based routing algorithm to search the path between source-destination pairs based on the network topology. In [17], a novel routing algorithm is presented. This scheme uses a three-step procedure: seek a new path, backtrack the path, store the new path. This scheme is effective in complex scenarios.

One of the potential problems associated with structural-based testing methods is area overhead caused by the test wrapper and extra pins required to connect the TAM ports to core-under-test ports [18]. Another problem is offline testing which requires obstruction in normal operation, while functional-based testing methods can be applied to NoC concurrently with the normal operation of the network.

A very significant point to achieve high fault coverage is to detect the exact fault location, with low power consumption and low added traffic. However, most of the above methods, either do not locate the fault or diagnose it inaccurately with high latency or power consumption.

3. Preliminaries

In this section, first, the structure of a typical NoC is discussed, then a closer look at the different switch fault models and test methods is taken.

3.1. NoC Structure

NoC consists of routers, links and network interfaces (NIs). Links provide the connection between the routers and also NIs. Network interfaces are placed between a router and an IP core, and are used as mediators between IP cores and the interconnect network. Routers are responsible for routing data units of transmission from source to destination [18].

As is shown in figure 1, router architecture is composed of ports, buffers and combinational circuit (also called Routing Logic Block). Based on the routing scheme, combinational circuit connects the input port to the appropriate output port.

3.2. Switch Fault Models

The system level fault models contributed to the router functionality are summarized in this subsection.

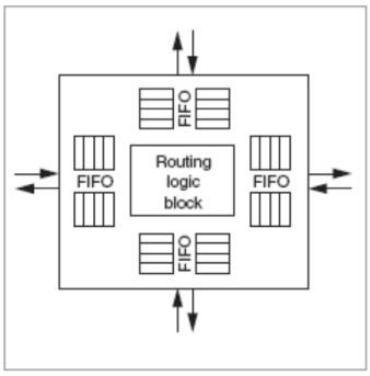


Figure 1. Structure of router

Dropped data fault: The switch erroneously drops the received packet. This type of fault is usually transient, meaning that a router drops the packets occasionally.

Corrupt data fault: Although the routing is correct, data gets corrupted during the routing of packet.

Direction fault: The switch sends the packet through a wrong output and consequently misleads the packet. Stuck-at-Port (SAP) is the most typical direction faults, when a router always sends the packet to a specific port. This output port might be at north (stuck-at-north), east (stuck-at-east), west (stuck-at-west) or south (stuck-at-south).

Multiple copies in space fault: In addition to the desired output port, packet is sent through several other outputs.

3.3. Testing NoC

Test strategies aim to solve various NoC problems caused by flawed design implementation and effect or representation of defect on system. Testing methods are categorized into functional testing and structural testing.

Functional testing: Examines whether or not the device-under-test shows the supposed behavior. This can be done either by applying functional test vectors and examining the output, or by using hardware redundancy in the design.

Structural testing: Structural testing refers to testing the structure of the device-under-test by using lower levels of fault models [18]. For instance, testing logic gates or input/output ports for stuck-at-faults (SAFs) is considered structural testing.

Furthermore, testing methods can be classified into offline and online testing.

Offline testing refers to testing the NoC components while the chip is not operating. Offline testing obstructs the normal operation of NoC.

Online testing refers to testing the NoC without obstructing the normal operation.

In this work, we propose an online and functional test strategy.

4. Initial Approach

This section begins with an explanation of the essential registers included in each router, plus the requisite routing information added in packet header. The initial test method and fault tolerant routing algorithm are described at the end of this section.

4.1. The Proposed Structure

This subsection includes some prerequisite information about the structures considered in this paper.

1) **Requisite Register:** Each router contains a 10-bit register similar to figure 2 in order to maintain the status of its neighbors, whether they are faulty or have more than one faulty neighbor. To this end, the register retains the neighbors' status using four bits, each of which represents the status of one neighbor. '1'indicates that the related neighbor is faulty. The last four bits of this register exhibit the risky status of neighbors. A router is contemplated risky if it is exposed to more than one faulty neighboring router. Likewise, if any of these bits equals '1', it indicates the risky situation of the pertinent neighbor. The two remaining bits belong to the current router status. Bit0 states whether the current router is a transient dropper or not, and the other one reveals its risky situation.

Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
risky status of the north neighbor	risky status of the south neighbor	risky status of the east neighbor	risky status of the west neighbor	Current router' s risky status	Faulty status of the north neighbor	Faulty status of the south neighbor	Faulty status of the east neighbor	Faulty status of the west neighbor	Current router' s dropping status

Figure 2. 10-bit local registerin initial approach

2) **Header Fields:** Routers update their registers after obtaining the most recent information about their neighbors by referring to the information embedded in the header of data packets. Hence, some requisite information need to be transferred within the packet header.

As it is illustrated in figure 3, three fields are added to the header. The faulty router field represents the address of the last faulty router that has been seen by the preceding router.

The force field indicates whether the previous router has erroneously misled the packet out of its supposed path toward its destination or was forced to do so. Note that some packets might have already been distracted from the supposed path; another situation is that the next router along the desired path is faulty, thus the fault-free router is forced to find a path other than the supposed one. Since we intend to minimize the drop-rate, in such situation router sets the force bit to '1' in order to inform the next router of its situation, and then continues routing the packet towards destination. Opting for negative-first [12] deadlock-free routing algorithm as the baseline approach, each router is simply able to determine the supposed path according to the current location of packet and its destination.

The equality of risky field to '1' illustrates that the preceding router has more than one faulty neighbor. On the account of the information inside the register, each router sets this field equal to '1' if it has perceived that there is more than one faulty router in its neighborhood.

SOP	*faulty router*	*Force*	*risky*	Destination address	EOP

Figure 3. Fields of the packet header in initial approach

4.2. The Proposed Test Strategy

In this step, the fault models considered in the initial approach is delineated and then the test method is explained.

1) **Fault Models:** The fault models considered in this approach are transient dropping, multiple copies in space, and SAF.

In this paper, the combinational parts of the routers such as the routing unit, the multiplexers and the crossed bar switch are under the test.

2) **Fault Detection Method:** For achieving deadlock-freedom, we selected the negative-first routing algorithm as our baseline approach. With reference to this routing algorithm, eliminating two turns, as shown in figure 4, will guaranty deadlock-freedom [12]. Negative-first routing algorithm is based on the XY routing algorithm. This assessment discredits the XY routing algorithm on the grounds of omitting four turns, which wastes some of the adaptability.

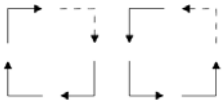


Figure 4. The permitted (solid line) and prohibited (dashed line) turns

In order to meet our goals, output ports are prioritized to be chosen depending on the position of the current router, the position of destination and the routing information offered in local register and packet header. This prioritizing process makes the network fault-tolerant. It remarkably increases the adaptability, which results in reduction of drop-rate. In figure 5, the numbers assigned to output ports declare their priorities in different situations. The first output port with the highest priority that satisfies the requisite conditions defined in routing algorithm will be chosen as the desired output port. If none of the priorities satisfies the essential condition, the packet will be dropped.

Destination X_D	Destination Y_D	Priorities of permitted output ports (according to the proposed work)
$>X_C$	$<Y_C$	1.South 2.West
$>X_C$	$>Y_C$	1.East 2.North 3.South 4.West
$>X_C$	$=Y_C$	1.East 2.South 3.West
$<X_C$	don't care	1.West 2.South
$=X_C$	$<Y_C$	1.South 2.West
$=X_C$	$>Y_C$	1.North 2.West 3.South

Figure 5. Output ports priorities for different positions of the destination and the current router

The whole process in a router after the entrance of a packet comprises four steps:

- Check if the prior router has followed the proposed algorithm described in figure 5; otherwise set the related bit in your register to '1'
- Extract the faulty router field in header. If it is one of your neighbors, set the related bit of your register to '1'
- Choose the output port according to the proposed algorithm
- Set the header's fields if required.

If the chosen output port is not the highest priority, then the current router must inform the successive router of this condition. The current router sets the force bit in the packet header equal to '1'. As a result, each router is able to determine whether the earlier router was faulty or not. This is where the detection and location of SAP and multiple copies in space faults state takes place in this algorithm.

With the purpose of diagnosing the transient dropping fault, each router sets a one-bit register to '1'as a new packet arrives, and when it leaves, the router changes back the bit to '0'. Upon arriving a new packet, the current router checks the aforementioned bit. If it is equal to '1', it conceives that the last packet has been dropped and consequently consider itself as a transient dropper.

As mentioned above, the router sets the faulty router field according to the three steps shown in figure 6.

The risky bit of the header is set to '1' if the current router realizes that more than one of its neighbors are faulty based on the information offered in the current packet header and in its own register.

IF you have dropped the last packet **THEN**
  Fill in the *faulty router* Field of header with your own address;
ELSIF the preceding router has routed the packet supposedly **THEN**
  IF the output port with the highest priority is NOT suitable **THEN**
    Fill in the *faulty router* field of header with its address;
ELSE
  IF the *force* bit of header has NOT been set to '1' **THEN**
    Fill in the *faulty router* field of header with the address of prior router;

Figure 6. The procedure of setting the faulty router field in header

4.3. The Proposed Fault Tolerant Routing Algorithm

The proposed fault-tolerant routing algorithm is described in figure 7 As mentioned before, the chosen output is the one with the highest priority that meets the requisite conditions defined in routing algorithm. These conditions aim to guaranty deadlock-freedom and minimize the drop-rate. The ensuing steps describe the requisite conditions that the desired output port must meet:

- The output port must not be connected to a faulty router.
- The output port must not form the prohibited turns depicted in figure 4.
- The output port cannot be the same as the input port
- The output port should not be connected to a risky router (as much as possible)

The risky routers, will likely drop the packet (since most of the ports will be eliminated according to the aforementioned conditions), thus the risky routers have lowest priority in our proposed routing algorithm.

The input port cannot be selected as the desired output port, since the U-turn may violate the deadlock-freedom of the proposed scheme.

5. Enriched Approach

This section commences with the explanation of the revised requisite register in each router, considered fault models and the routing information added in the packet header. The

section proceeds with the proposal of the enriched test strategy; the novel fault tolerant routing algorithm, namely RSFR, and the employed backtrack technique.

Read the Header;
IF (the *risky* bit in header = '1') **THEN** set the risky-bit related to previous router to '1';
IF (previous router is faulty) **THEN**
IF (it has not been recognized before) **THEN** set the pertinent bit in register to '1';
IF (the faulty router posted in header is a neighbor) **THEN** set the pertinent bit in register to '1';
IF (the neighbor with the highest priority is with NO-PROBLEM) **THEN** choose it as the next router;
ELSIF (the neighbor with the second priority is with NO-PROBLEM) **THEN** choose it as the next router;
ELSIF (the neighbor with the third priority is with NO-PROBLEM) **THEN** choose it as the next router;
ELSIF (the neighbor with the forth priority is with NO-PROBLEM) **THEN** choose it as the next router;
ELSIF (there is any risky neighbor(s)) **THEN** choose the neighbor with the highest priority;
ELSETHEN drop the packet and set the risky bit in your register to '1';
IF (the drop-bit = '1') **THEN** set the *faulty router* field in header with your address;
ELSIF (the previous router is faulty and has not been recognized before) THEN set the *faulty router* field in header with previous router's address;
ELSIF (the neighbor with the highest priority is now with NO-PROBLEM) THEN set the faulty router field in the header with address of the neighbor with the highest priority and set the *force* field of header to '1';
Set the *risky* field of header based on the risky-bit of register;

Figure 7. The whole process of routing

5.1. The Proposed Structure

1) **Requisite Register**: Similar to the initial approach, a local register, as shown in figure 8, is allocated to each router in favor of maintaining the fault tolerance of the network. As mentioned before, each router considers a single bit pertained to each router in its neighborhood in order to sustain its status as a faulty or fault-free neighbor. Bits numbers 2 to 5 are representatives of faulty status of the west, the east, the south and the north neighbors, respectively. Moreover, hence it is highly unlikely for a router to have at least two risky neighbors, we have allotted two bits to address one risky direction for each router, according to table 1.

In comparison with the requisite register in the initial approach, we have halved the necessary registersin employing our risk awareness technique.

Table 1. Addressing the Direction of Risky Neighbor in Requsite Register

Bit number 7	Bit number 6	Risky direction
0	0	West
0	1	East
1	0	South
1	1	North

In addition, another risky neighbor may be retained in the pertinent field in the header described in ensuing subsection. Therefore, at least one risky neighbor is offered to the current router for conducting the routing process.

In this paper, four fault models have been considered: transient dropping, SAF, multiple copies in space, and finally turn/straight fault (having trouble in steering turn paths or straight paths between input port and the supposed output port). Depending on the input port and the desired output port, a packet might pass the router in a straight or turn path. For instance, according to the routing algorithm described in the initial approach, if the destination is above the current router in the same column, the out port will be the north. If the current router has received the packet from the east neighbor, this path is considered as a turn path. A straight/turn faulty router may have trouble in steering turn paths, straight paths or both. In order to discriminate between the three types of straight/turn fault, the requisite register contains two bits to clarify the type of fault in one faulty neighbor, as shown in table 2.

Table 2. Addressing the Type of Fault in Requisite Register

Type of Fault	Bit 0	Bit 1
SAF	0	0
Transient dropper	0	0
Turn and Straight combined	0	0
Turn	0	1
Straight	1	0
Fault-free	1	1

Since the possibility of having more than one turn-faulty router or straight-faulty router among four neighbors is low, the register contains only two bits as an indicator of the type of fault for its faulty router. However, it is possible that more than one faulty router exist in the router immediate proximity. In such case, router must be able to distinguish which of the faulty routers status has been explained by these two bits. Such discrimination is possible by defining priority among the neighbors. Whenever a router, according to the pertinent fields in header or the routing decision of the previous router, discriminates a turn-faulty or straight-faulty neighbor, it will fill these two bits based on this priority order: west, east, south, and north. Thus, if these two bits are already filled by a lower priority neighbor, these two fields will be replaced by the most recent information and afterwards, these two bits represent the type of fault for the recently detected faulty router which has the highest priority. Although the information about the type of fault in the previous faulty neighbor is deleted, the router is still informed about all faulty neighbors in bit 2 to bit 5 of its local register. Thus, even in such occasional cases, router is able to prevent delivering packets to faulty neighbors.

As is shown in figure 8, unlike the initial approach the two bits used to represent the status of current router have been omitted tooptimize the local register size. The current router risky status is merely recognized by faulty status of neighbors in its requisite register. Furthermore, its status, either as a transient dropper or a risky router, will be announced to the successive routers through the header of data packet.

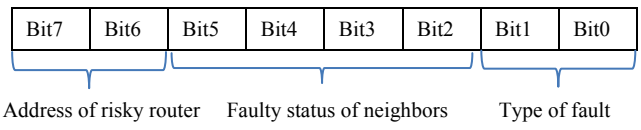


Figure 8. 8-bit local register in RSFR scheme

2) **Header fields:** In RSFR, two new fields have been added for routing information in packet header, as illustrated in figure 9. The two-bit TOF (Type of Fault) indicates the

type of the detected faulty router declared in the faulty router field, whether it is turn-faulty, straight-faulty, SAF or transient dropper. This field is filled alongside the faulty router field. The two-bit PID (Penultimate Input Direction), represents the input direction of the prior router. PID field enables the router to determine whether the occurred fault in previous router is turn or straight fault. Moreover, TTL (Time-to-Live) has been considered for the packet in order to prevent live lock, by dropping the trapped packets. The value of TTL is obtained experimentally, and is equivalent to the longest path taken by a packet through the network in presence of faulty routers.

Figure 9. Fields of the packet header in RSFR scheme

5.2. The Proposed Test Strategy

On the grounds that a new fault model has been added to work compared to the initial approach, it is evident that the test strategy and also the routing algorithm will have some ensuing modifications.

Similar to the initial approach, when a router receives a packet, according to its own position and the packet destination, it opts to distinguish whether it was supposed to get this packet or not. If the preceding router has been recognized as a faulty router, then, the host conducts the following steps:

- Discriminate the type of fault by determining the desired output port in previous router and the input port in previous router which is offered in PID field of packet header.
- Set the TOP and faulty router fields in header.
- Apply the extracted information to its local register.

The supposed output port for the preceding router is determined according to the position of the previous router and the position of the destination. The discrepancy between the chosen output port by the last router and the supposed one, illustrates whether the type of fault is turn or straight.

When applying the recently detected fault, if the related bit of faulty neighbor in register is already '1', it means that the faulty status of this router has been recognized before. Therefore, the host opts to check if the two bits representing the type of fault in register belong to this router or not through the prioritizing process described in previous subsection. If the type of fault in local register is related to this neighbor and the type of fault is not the same as the one discovered now (for example the register suggests that this router is turn-faulty and according to the current detection, its type of fault is straight), the host will set the type of fault in register equal to "00". Hence, the next time the current router is aware that the mentioned router is not a suitable option either for steering turn or straight path. In addition, on the grounds that the mentioned router's situation will be revealed by the TOF in header, other routers which are in close proximity to this router will as well be cognizant of its situation.

According to the mechanism presented in figure 6, if the current router perceives that the faulty router field of the header contains the address belonging to the preceding router, it marks that router as a transient dropper in its own register, and also sets the two bits of register that represent the type of fault to "00". Since the dropping fault is transient, the transient dropper, after successfully transmitting few certain number of packet, will advertise a clearing signal within the packets header to inform the neighbors about the change. The clear router carries out the clearing signal by setting its address in faulty router field and '11' in TOF field of head. In RSFR, the amount of successfully transmitted packets needed for a transient dropper to start advertising clearing signal is called "Clearing Period".

5.3. Fault Tolerant Routing Algorithm

In comparison with the initial approach, RSFR routing algorithm benefits from "path prediction" mechanism. Path prediction mechanism adds these two advantageous conditions to requisite conditions discussed in subsection 4.3:

- Leading the packet to the output port must never result in occurring the prohibited turns depicted in figure 4.
- If the next router is straight/turn faulty, the forthcoming path in the next router must not match its TOP.

Figure 10 illustrate the situation in which the north port may seem safe, but leading the packet through it will eventually result in a prohibited turn at some point. Therefore, in this example, the north port cannot be a desired port.

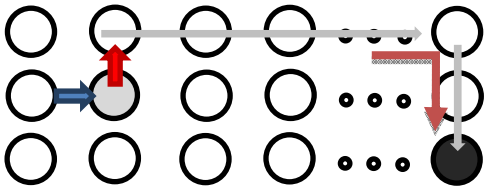


Figure 10. Path prediction mechanism in RSFR scheme

On the other hand, if the output port is straight/turn faulty, according to the location of successive router and the destination, the current router will determine the probable path in the next router and if it does not match the detected type of fault, the packet can be delivered to the next router.

Path prediction mechanism increases the adaptability and fault tolerance in NoC.

5.4. Backtrack Procedure

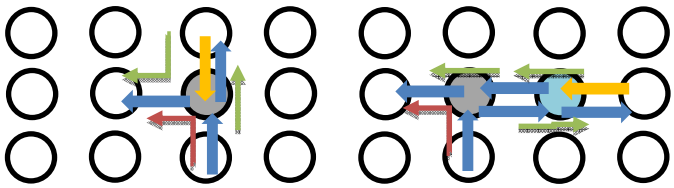
RSFR utilizes backtrack (BT) procedure to avoid dropping the misled packets as much as possible. If there is no path from current router to the packet destination and the previous router is not faulty, the current router backtracks the packet through the input port. While the initial approach prohibits U-turns, in RSFR it is allowed only in case of backtracking.

BT packet is marked using this unique signature: router puts its own address in faulty router field of header and sets the TOFto"01".

Upon receiving a BT packet (packet marked with aforementioned signature), the current router marks the preceding router as faulty, since the prior router has had no path to destination and is no longer useful for the current router. Yet, the prior router may still be useful to other routers in network. Therefore, even though the current router

updates its local register, it will not announce this to other routers. Updating the local register will prevent the BT packet from getting transferred to the previous router again. The BT packet will then be routed to the next priority among the appropriate output ports. This procedure continues until either a router has no appropriate output port or the packet is merely dropped due to TTL expiry.

Deadlock Freedom Proof:
Without devising some certain limitations, making U-turn in BT procedure would violate the prohibited turns and consequently eliminate the deadlock freedomof routing algorithm. Figure 11 illustrates the two possible scenarios that will result in violation of prohibited turns. The yellow arrow marks the backtracked packet and the narrow green arrows show the made turns. Figure 11 demonstrates that although none of the made turns violates the prohibited turns, the overall made turn opposes the basis of Negative-first routing algorithm.



a. Simple U-turn b. Long U-turn

Figure 11. Two possible scenarios for violation of prohibited turns (For each scenario, only one of the possible cases is depicted below. Similar forms are not depicted)

RSFR is able to detect these two possible violations and drop them in order to provide deadlock-freedom. As discussed earlier, before transferring a packet, PID field of header is filled with the second last input port. When backtracking a packet, router refrains from updating this field, therefore the PID in a backtracked packet is technically the third last input port.

In the first scenario, upon receiving a BT packet, the grey-colored router will not consider the output ports that would form a prohibited turn with the third last input port, offered in PID field of header.

In the second scenario, the grey-colored router will not directly receive the backtracked packet, and as a result is not able to detect the occurrence of prohibited turn (marked by red arrow). In order to prevent the second scenario, the blue-colored router which directly receives the BT packet must prevent the shaping of a "long U-turn". To this end, in addition to prohibited turns, blue-colored router must eliminate the output port which is in opposite direction of PID. In the second scenario, the blue-colored router must eliminate the west output port since the PID in BT packet is east.

Figure 12 illustrates two examples of the proposed method. In the first example, shown in figure 12 (a), router (at node) 7 is the source, and router 15 is the destination. Router 11 is supposed to deliver the received packet to router 15 in a straight path. However, this router is straight-faulty, and mistakenly sends the received packet to its north neighbor. Since router 10 has received a misdirected packet, by determining the supposed path in previous router, it sets router 11 as a straight-faulty neighbor in its local register. Although router 11 is straight-faulty, according to current location of packet and destination, router 11 will deliver the packet in a turn path; therefore, it seems safe for forwarding the packet through the destination. The second example, shown in figure 12 (b), is similar to the first example, but the type of fault in router 11 is stuck-at-north. Similarly, router 10 marks router 11 as straight-faulty and sends the packet to router 11. Since router 11 is stuck-at-north, it will send the packet back to router 10.At this point, router 10 will change the previous router's type of fault from straight-faulty to stuck-at-north. According to figure 5, the eligible output ports are south and west. The south neighbor is faulty and the west port will form the prohibited turn, thus the packet will be dropped. The RSFR aim to minimize the possibility of live-lock in NoC.

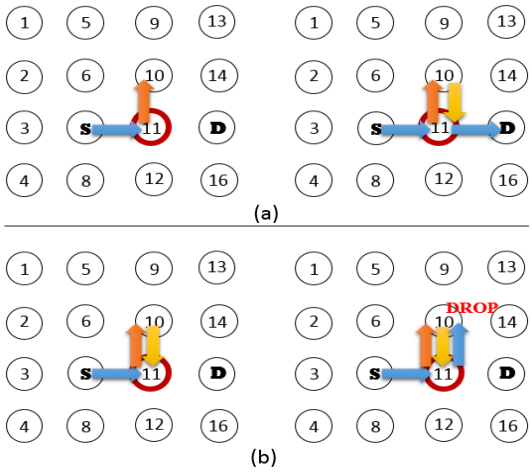


Figure 12. Two examples of the functionality of RSFR

6. Experimental Results

In this section we evaluate the performance of RSFR scheme described in Section 5, the initial approach described in Section 4 and the base-line approach (negative-first). Simulation are conducted using NoCem [19] (a 2-dimensional integrated emulation environment for NoC research, described in VHDL). To this end, the routing algorithms as well as the data packet generator and fault injection unit are implemented in NoCem. Taking advantage of data packet generator, we are able to adjust the percentage of addressed routers in our simulations.

For simulation results, we have chosen average latency, drop-rate and power overhead in order to make a comprehensive evaluation of the performance of different schemes.

Average latency: The average time between the start of a packet transmission at the source until it arrives at the destination.

Drop-rate: The number of data packets not delivered to the destinations per those generated by the data packet generator. This could be due to either routing algorithm or TTL expiration.

Fault coverage: The number of detected control faults per number of all injected control faults.

Power consumption overhead: The ratio of total consumed power throughout the simulation, to that in baseline approach.

For obtaining better accuracy, simulations are carried out for 5 different benchmarks and the results presented in this section are the arithmetic mean. The simulation model is summarized in table 3.

Table 3. Simulation Parameters

NoC size	4×4 to 16×16
total sent PKTs	125 to 2000
Topology	Mesh
addressed routers [%]	25 to 100
PKT injection rate [PKT/cycle]	2.5
injection cycle [ns]	8
Clearing Period	5

Addressed routers are the percentage of routers selected as destinations during the simulation. Note that with increasing NoC size, the number of sent packets increases proportionally. The number of sent packets corresponding to different sizes of the network is depicted in table 4.

Table 4. Number of Injected Packets in Different Sizes of Network

NoC architecture	Total sent PKTs
4×4	125
8×8	500
16×16	2000

Table 5. Average Latency [ns] in Nagative-first and Initial Approach for Different Sizes of Network

NoC Architecture	Negative-first-with faulty routers	Negative-first-with fault-free routers	initial approach-with faulty routers	Initial Approach-with fault-free routers
4×4	35.6	32	32.8	32.4
8×8	196	52.8	54.2	53.2
16×16	433.4	92	152.8	93

Table 6. Average Latency [ns] in RSFR for Different Sizes of Network

NoC Architecture	RSFR-with faulty routers	RSFR-with fault-free routers
4×4	34	32.6
8×8	56.4	54
16×16	161.6	94.2

In table 5 and table 6, the average latency for negative-first, the initial approach and RSFR in three different sizes of network is shown. In Negative-first routing algorithm, unlike the two proposed methods, there is no perception of faultiness or riskiness of adjacent routers, which leads to high live-lock probability, and eventually high drop-rate due to TTL expiry; therefore the figures for Negative-first routing algorithm are excessively higher than those of our proposed methods. Also in the absence of fault in the network, the average latency in RSFR is almost equivalent to initial approach and Negative-first routing algorithm. In table 5 and table 6, the fault concentration in network is maintained at 10%.

Table 7. Drop-ratein Nagative-first and Initial Approach for Different Sizes of Networkwith the Presence of 10% fault

NoC Architecture	Negative-first-with faulty routers	initial approach -with faulty routers	RSFR-with faulty routers
4×4	0.120	0.056	0.025
8×8	0.264	0.042	0.018
16×16	0.236	0.021	0.011

Table 7 exhibits the drop-rate for negative-first, the initial approach and RSFR in three different sizes of network. While the two proposed schemes have substantially lower drop-rates, the increase in network size has decreased the drop-ratein our proposed schemes. This is due to its adaptability feature that takes full advantage of large NoCs (because in such networks the number of possible paths to destination is greater). A similar pattern, however, cannot be observed in negative-first routing algorithm. Among the proposed routing schemes, enriched approach (RSFR) outperforms the initial approach, mostly due to its backtracking capability.

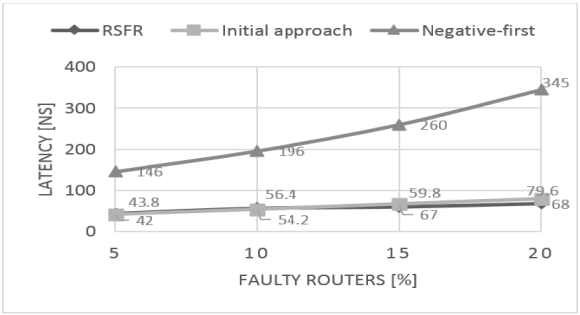


Figure 13. Average latency [ns] vs. faulty routers [%]

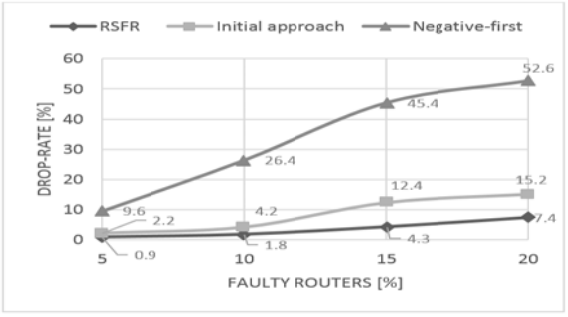


Figure 14. Drop-rate [%] vs. faulty routers [%]

Figure 13 and figure 14 depict the drop-rate and average latency, respectively, in an 8×8 NoC in various concentration of fault. The outcome demonstrates significant improvement in terms of latency and drop-rate from the baseline approach (Negative-first) to the proposed initial approach; when 20% of routers are faulty, the drop-rate descended 37.4% and the average latency is four times better. RSFR, however, has again outperformed the initial approach especially in terms of drop-rate.

Table 8. Fault coverage [%] in a 8×8Network

Addressed Routers [%] / Routing algorithm	25	50	75	100
Initial Approach	66	74	85	100
RSFR	84	92	98	100
ref [20]	≈32	≈63	≈71	≈76
Ref [2]	≈48	≈84	≈93	≈98

Table 8 shows the resulting fault coverage applying three methods discussed in this work in addition to distraction detection method [20] and combined distraction detection method and trapped packet detection method [21]. As it is shown, in terms of fault coverage, RSFR surpasses the other proposed methods.

The power consumption overhead is illustrated in table 9. The results are obtained using Power Complier tool for various packet injection rates in a fault-free 8×8 NoC. It can be seen that compared to the base-line approach, the power overhead caused by applying our proposed schemes is legitimately satisfactory.

Table 9. Power Overhead [%] vs. Injection Rate [PKT/cycle] Applying our Proposed Methods

Injection Rate [PKT/Cycle]		1.5	2.5	3.5	4.5
Power Overhead [%]	**Initial Approach**	2.7	4.4	5.6	6.7
	RSFR	3.1	5.2	5.9	7.1

7. Future Work

As a future work, we intend to extend our proposed scheme to detect and locate several data faults that may occur in NoC links in an on-line manner. These faults include stuck-at-fault, AND/OR-short and crosstalk. Moreover, a more in-depth analysis using more detailed metrics could be carried out.

8. Conclusions

In this work we proposed an adaptive fault-tolerant and deadlock-free routing algorithm namely RSFR, which empowers the routers to accurately locate and detect control faults occurring in network-on-chip. RSFR achieves high fault coverage in various NoC sizes. Experimental results indicate that RSFR is able to achieve low latency and low power consumption and substantially subordinate the drop-rate due to optimum utilization of local register, best-route anticipation by highly prioritizing the output ports and finally taking advantage of its backtrack mechanism.

The thorough performance analysis of negative-first, the initial approach and RSFR (enriched approach) conducted in Section 6, demonstrates great improvement in each step. It also shows that when there is no fault in the network, our proposed method will not tangibly affect the latency or power consumption, while in presence of fault, RSFR remarkably outperforms both baseline and initial schemes.

References

[1] M. Hervé, P. Almeida, F. L. Kastensmidt, É. Cota, and M. Lubaszewski, "Concurrent Test of Network-on-Chip Interconnects and Routers," *Proc. IEEE Intl Latin American Test Workshop*, pp. 1-6, 2010.

[2] A. Alaghi, N. Karimi, M. Sedghi, and Z. Navabi, "Online NoC Switch Fault Detection and Diagnosis using a High-Level Fault Model," *Proc. IEEE Intl Symp. Defect and Fault Tolerance*, pp. 70-74, 2007.

[3] M. Sedghi, E. Koopahi, A. Alaghi, M. Fathy, and Z. Navabi, "A NoC Test Strategy Based on Flooding with Power, Test Time and Coverage Considerations," *Proc. IEEE Intl Conf. Very Large Scale Integration Design*, pp. 28-34, 2008.

[4] A. M. Amory, E. Briao, E. Cota, M. Lubaszewski, and F. G. Moraes, "A Scalable Test Strategy for Network-on-Chip Routers," *Proc. IEEE Intl Conf. Test*, pp. 752-758, 2005.

[5] M. Hosseinabady, A. Banaiyan, M. N. Bojnordi, and Z. Navabi, "A Concurrent Testing Method for NoC Switches," *Proc. IEEE Intl Conf. Design Automation and Test in Europe*, pp. 1171-1176, 2006.

[6] M. Hosseinabady, A. Dalirsani, and Z. Navabi, "Using the Inter and Intra-Switch Regularity in NoC Switch Testing," *Proc. IEEE Intl Conf. Design Automation and Test in Europe*, pp. 361-366, 2007.

[7] S. Babaei, M. Mansouri, B. Aghaei, and A. Khadem-zadeh, "Online-Structural Testing of Routers in Network on Chip," *Journal of World Applied Sciences*, vol. 14, no. 9, pp. 1374-1383, 2011.

[8] C. Grecu, P. Pande, A. Ivanov, and R. Saleh, "BIST for Network-on-Chip Interconnect Infrastructures," *Proc. IEEE Intl Symp. Very Large Scale Integration Test*, pp. 26-30, 2006.

[9] Y. Zheng, H. Wang, S. Yang, C. Jiang, and F. Gao, "Accelerating Strategy for Functional Test of NoC Communication Fabric," *Proc. IEEE Intl Symp. Asian Test*, pp. 89-93, 2010.

[10] C. Grecu, A. Ivanov, R. Saleh, E. S. Sogomonyan, and P. P. Pande, "On-Line Fault Detection and Location for NoC Interconnects," *Proc. IEEE Intl Symp. On-Line Testing*, pp. 145-150, 2006.

[11] A. DeOrio, D. Fick, V. Bertacco, D. Sylvester, D. Blaauw, J. Hu, and G. Chen, "A Reliable Routing Architecture and Algorithm for NoCs," *IEEE Trans. Computer Aided Design of Integrated Circuits and Systems*, vol. 31, no. 3, pp. 726-739, 2012.

[12] T. Lehtonen, P. Liljeberg, and J. Plosila, "Analysis of Fault Tolerant Deadlock-Free Routing Algorithms for Mesh NoCs," *Proc. IEEE Intl Workshop on Diagnostic Services in Network-on-Chips*, pp. 54-57, 2009.

[13] P. Rantala, T. lehtonen, J Isoaho, and J. Plosila, "Fault-Tolerant Routing Approach for Reconfigurable Networks-on-Chip," *Proc. IEEE Intl Symp. System-on-Chip*, pp. 1-4, 2006.

[14] A. Alhussien, N. Bagherzadeh, F. Verbeek, B. Van Gastel, and J. Schmaltz, "A Formally Verified Deadlock-Free Routing Function in a Fault-Tolerant NoC Architecture," *Proc. IEEE Intl Symp. Integrated Circuits and Systems Design*, pp. 12-16, 2012.

[15] Y. Liu, Y. Ruan, Z. Lai, and L. Sun, "A Reconfigurable Routing Method for Fault-Tolerant Mesh-Based Network on Chip," *Journal of Information and Computational Science*, vol. 10, no. 1, pp. 157-165, 2013.

[16] E. Wachter, A. Erichsen, A. Amory, and F. Moraes, "Topology-Agnostic Fault-Tolerant NoC Routing Method," *Proc. IEEE Intl Conf. Design Automation and Test in Europe*, pp. 1595-1600, 2013.

[17] E. W. Wachter, and F. G. Moraes, "MAZENOC: Novel Approach for Fault-Tolerant NoC Routing," *Proc. IEEE Intl Conf. System on Chip*, pp. 364-369, 2012.

[18] G. Nazarian, *On-Line Testing of Routers in Networks-on-Chips*, MSc Thesis, Delft University of Technology, Delft, Netherlands, 2009.

[19] G. Schelle, and D. Grunwald, "On-Chip Interconnect Exploration Formulticore Processors Utilizing FPGAs," *Proc. IEEE Intl Workshop on Architecture Research using FPGA Platforms*, pp. 230-232, 2006.

[20] S. Klotz, and S. Holst, *Online Diagnosis of Network-on-Chip*, Technical Report in Reliable Networks-On-Chip in the Many-Core Era, Haupt-Seminar, University of Stuttgart Institut für Technische Informatik, 2009.

Sanaz Alamian received her bachelor degree in Computer Engineering in 2010 from K.N.Toosi University of Technology in Tehran, Iran. She continued her education in Computer Architecture at Sharif University of Technology toward Master's degree. Currently, she is a PhD student in University of California, Irvine, working on parallel computing and parallel programming in graph-based algorithms used in big data centers and social networks.
E-mail: sani_alamian@alum.sharif.edu

Ramin Fallahzadeh received his B.S. degree in computer engineering from Sharif University of Technology, Tehran, Iran in 2014. Currently, he is a Ph.D. student in computer science at Washington State University. His current research interests include embedded systems, pervasive computing, networks on chip, and wireless sensor networks. The focus of his research is on algorithm design and system level optimization of embedded and pervasive systems with applications in healthcare and wellness. He is a student member of the IEEE.
E-mail: rfallahz@eecs.wsu.edu

Shaahin Hessabi received the B.Sc. and M.Sc. degrees in Electrical Engineering from Sharif University of Technology, Tehran, Iran in 1986 and 1990, respectively. He received his Ph.D. degree in Electrical and Computer Engineering from University of Waterloo, Waterloo, Ontario, Canada in 1995. He joined Sharif University of Technology in 1996, and is currently an associate professor at the Department of Computer Engineering. His current research interests include System-on-Chip and Network-on-Chip, and VLSI design and test. He has published more than 100 refereed papers in the related areas. Dr. Hessabi has served as the program chair, general chair, and program committee member of various conferences.
E-mail: hessabi@sharif.edu

Paper Handling Data:

Submitted: 21.08.2014
Received in revised form: 11.12.2014
Accepted: 23.01.2015
Corresponding author: Dr. Shaahin Hessabi,
Department of Computer Engineering, Sharif University of Technology, Tehran, Iran.