Journal of Universal Computer Science, vol. 14, no. 22 (2008), 3716-3736
submitted: 23/4/08, accepted: 27/12/08, appeared: 28/12/08 © J.UCS

Online Network-on-Chip Switch Fault Detection and Diagnosis Using Functional Switch Faults

Naghmeh Karimi
(School of Electrical and Computer Engineering, University of Tehran, Tehran, Iran
naghmeh @cad.ece.ut.ac.ir)

Armin Alaghi
(School of Electrical and Computer Engineering, University of Tehran, Tehran, Iran
armin @cad.ece.ut.ac.ir)

Mahshid Sedghi
(School of Electrical and Computer Engineering, University of Tehran, Tehran, Iran
mahshid @cad.ece.ut.ac.ir)

Zainalabedin Navabi
(School of Electrical and Computer Engineering, Microelectronics Centre of Excellence
University of Tehran, Tehran, Iran
navabi @ece.wpi.edu)

Abstract: This paper presents efficient methods for online fault detection and diagnosis of Network-on-Chip (NoC) switches. The fault model considered in this research is a system level fault model based on the generic properties of NoC switch functionality. The proposed method is evaluated by fault simulation in a platform using this system level fault model. The experimental results show that with a relatively low area overhead, a large number of NoC switch faults can be detected and diagnosed.

Key-Words: Network-on-Chip, Switch, Online Testing, Functional Fault Model, Fault Diagnosis.
Categories: B.4.3, B.4.5, B.8.1

1 Introduction

With advancement in VLSI technology and moving towards deep-submicron and nano design, System-On-Chip is to contain several cores on a single die in which each die is composed of thousands of gates [Marculescu, 06]. On the other hand, due to the complexity of the applications that are run on SoCs these architectures should meet power and performance requirements.

One of the main problems in ultra deep submicron technologies is global delays. In fact gate delays scale down with the technology but global wire delays grow exponentially due to the increase of crosstalk and capacitance caused by narrow channel width.

The most frequently used communication architecture in SoC structures is the centralized bus-based approach in which all the components attached to the bus, share the same transmission medium using arbitration logic to serialize bus access requests. Shared-bus communication approaches are cheap and easy to release. However limited bandwidth due to the increasing load on global bus line and limited scalability of the shared-bus communication approaches are the main factors that encourage designers to move towards a shared segmented global communication structure [Bjerregaard, 06]. On the other hand, hierarchical bus architectures that reduce low bandwidth problem of shared-bus approaches lack the necessary scalability [Mello, 05].

According to the above mentioned problems, on-chip interconnect plays an important role in dictating performance, energy and fault tolerance in overall system [Kim, 06] [Benini, 02] [Dally, 01] [Ho, 01]. Thus there is a trend towards designing a scalable, reusable and predictable SoC-based architecture [Orgas, 06] [Jantsch, 03]. One such communication-centric approach is the Network on Chip (NoC) [Hansson, 05] [Bertozzi, 05] [Keutzer, 00] [Murali, 05a] [Liu, 05].

An NoC is a scalable packet switched communication platform composed of a set of structured switches and point to point channels interconnecting the processing cores of a SoC in order to support communication among them [Cota, 03a] [Hosseinabadi, 07]. NoCs are characterized by different tradeoffs regarding throughput, latency, silicon area, power consumption and reliability [Pande, 05].

The failure rate in emerging nano technologies is estimated to be in order of 10^{-2} to 10^{-1} due to the shrunk size and frequency characteristics of these technologies in comparison with 10^{-9} to 10^{-7} failure rate in CMOS technologies [European Commission, 01] [Nikolic, 01]. This unreliability of nano and deep submicron technologies makes online testing techniques essential.

The main focus of this paper is reliability enhancement in NoC switches by applying online error detection and diagnosis. The rest of this paper is organized as follows. Section 2 presents an overview of the related works in NoC testing. Section 3 discusses basic concepts of NoCs. The architecture of the NoC switch used in this research is presented in Section 4. Section 5 discusses a fault model for NoC switches. This is a high level fault model and is suited for our online error detection and diagnosis methods, and perhaps some high level reliability applications. Our online fault detection methods are presented in Section 6. Section 7 presents fault classification and related detection methods. Section 8 deals with our online error diagnosis. Finally the experimental results of applying the proposed scheme to a number of NoC structures is given in Section 9 followed by the conclusion remarks in Section 10.

2 Related Works

In SoC designs many researchers deal with testing the IP cores, giving little emphasis on the communication infrastructure [Grecu, 06b]. Different research groups have proposed the reuse of the communication infrastructures as a Test Access Mechanism (TAM) to deliver the test data to the individual cores of SoCs [Vermeulen, 03] [Cota, 03b].

Nahvi [Nahvi, 04] and Aktouf [Aktouf, 02] propose the use of packet switching to test embedded cores. Nahvi et al. propose the use of a packet switch communication-based TAM, so called NIMA, for a SoC. Since NIMA has been primarily designed for testing, routing and addressing strategies are defined considering only the test requirements of each system [Nahvi, 04]. Aktouf [Aktouf, 02] suggests the use of a boundary scan wrapper to test NoC components. The test includes the routers, the RAM blocks, and the embedded processors of the NoC architectures.

The use of on-chip communication infrastructure as TAM for testing core-based systems has been proposed in [Cota, 03a] to reduce test costs in terms of area and pin number.

In addition to testing cores in NoCs, the communication infrastructure should be tested carefully. According to the similarities of testing wires and switches in NoCs with the testing of RAM-based FPGA interconnections, FPGA testing methods can also be applied to NoCs [Jantsch, 03]

Grecu et al. [Grecu, 07] propose a test strategy for testing switch blocks as well as inter-switch wire segments. This method is based on using already tested NoC components to transport test data to the components under test in a recursive manner. Because of the inherent parallelism of the data transport mechanism, test time is reduced.

A number of methods suggest using a wide variety of Design-For-Test (DFT) techniques for testing NoC components.

A test strategy for NoC routers based on partial scan and on-chip response evaluation has been proposed in references [Amory, 05a] and [Amory, 05b].

In reference [Li, 06] a test data transportation method with multiple data flit formats and a novel scan chain configuration is presented which maximize the utilization of an NoC channel without adding too much hardware overhead.

Ubar [Ubar, 03] deals with BIST strategies to test NoC architectures. Another BIST method for testing inter-switch links in a NoC structure has been proposed in reference [Grecu, 06b]. This method uses a high level fault model [Cuviello, 99] to deal with the crosstalk effects due to inter-wire coupling.

Another test methodology for testing the NoC switches has been presented in references [Hosseinabady, 06] and [Hosseinabady, 07]. This methodology broadcasts same set of test vectors to all switches of an NoC structure and detects existing faults through comparison of switch outputs with each other.

With transient and intermittent faults becoming a dominant failure mode in modern VLSI, widespread deployment of online test approaches has become crucial. Traditionally, error detection and correction mechanisms are used to protect communication subsystems against the effects of transient malfunctions. What follows discusses some of the previous works in online testing.

Raik et al. [Raik, 06] have proposed an external test method for NoCs based on functional fault models, which targets single stuck-at faults in the network switches. The proposed method eliminates the need for scan paths and wrappers.

Murali et al. [Murali, 05b] consider error detection codes to build self checking NoCs. They classify error detection codes for NoC applications to switch-to-switch (s2s) and end-to-end (e2e) categories. In the former category all the switches located in the path of the packet sender node and the packet receiver node check the packet to

detect the probable faults while in the latter case just the sender and receiver nodes deal with the validity of the packet.

For self checking circuits two frequently used error detection codes are parity checking and dual rail codes. Code disjoint switches along with parity checkers can also be used for online fault detection and diagnosis in NoC communication fabrics [Grecu1, 06].

Comparing the method presented in reference [Grecu, 06a] with e2e and s2s approaches [Murali, 05b] shows the effectiveness of using code disjoint schemes over e2e and s2s approaches in terms of latency, power consumption and throughput.

Bhojwani et al. propose using Test Infrastructure-IP in NoC architectures for online testing of NoC structures. This research discusses the potential hazards to online testing and proposes a number of techniques to mitigate them [Bhojwani, 07].

A number of approaches to achieve fault tolerant NoC architectures have also been presented in the literature [Park, 06] [Kim, 05]. Reference [Park, 06] discusses various types of reliability hazards in NoC structures and proposes a number of recovery techniques for reliability enhancement in presence of reliability hazards.

In general, the choice of an error recovery technique for an application requires considering different parameters, i.e., power, performance, and reliability tradeoffs [Murali, 05b].

Kim et al. [Kim, 05] classify soft errors that disturb the correct operation of the NoCs as link and router errors. The former occurs during the traverse of flits from one router to another while the latter occurs within the router architecture. Considering separate error coding techniques for header flit, five types of retransmission techniques are used to remove link errors in this approach. Moreover a number of transient fault protection techniques are used in this method to deal with router errors.

Along with testing techniques for synchronous NoCs, test of asynchronous NoCs that are used for Globally Asynchronous Locally Synchronous (GALS) platforms has also been studied in the literature and a number of test architectures have been proposed [Tran, 06] [Beigne, 05].

3 Preliminaries

As discussed in the previous sections, in designing NoC systems, several issues including topologies, routing mechanism and switching techniques should be considered.

There are different kinds of topologies for NoC architectures, among which the mesh architecture is getting more popular for its modularity.

Routing in an NoC deals with the path that a message passes from a sender to a receiver. In this paper we focus on regular 2-D NoC topologies using XY routing algorithm where a packet is first routed in the X direction and then in the Y direction before reaching the destination [Jantsch, 03].

Another issue in NoC structures to establish an internal path between an input and an output port of the router is the switch mechanism.

In circuit switching, a dedicated path from source to destination is established to send the packets. In this technique, bandwidth reservation increases average delay and decreases throughputs.

In packet switching, each packet is divided into fixed length packets. In this method unlike circuit switching, whenever the source wants to send a packet, it transmits the data on a path, established based on the intermediate routing decisions along the way.

In wormhole switching [Jantsch, 03], each packet consists of multiple fixed length control flow units, called flits. The first flit of a packet, so called header, includes the routing information to establish a path between source and destination. The subsequent flits follow this path. In this method no packet reordering is required in the destination [Amory, 05a]. In this paper switching is based on the wormhole approach.

4 NoC Switch Architecture

The NoC switch architecture used in this paper is illustrated in Figure 1. This switch includes five input/output ports; four of them connect the switch to its neighboring switches in the mesh and one connects the switch to its corresponding processor. In Figure 1, I/O ports are specified by the first letter of their direction, e.g., the I/O port located at the North of the switch is specified by 'N'.

As we mentioned in the previous section, in this research switching is based on the wormhole switching approach where packets are transmitted as units of message flow control, named flits. Therefore, a FIFO buffer is located at each input port of the switch which stores a few of incoming flits to provide flow control in transferring flits between routers and through routers. A four-to-one multiplexer is located at each output port. The inputs of the multiplexers are connected to the data at the head of FIFOs through a crossbar switch. We have not shown the crossbar switch for the sake of clarity and have only named each input of the multiplexer with the name of the input port feeding it. The router in the switch determines the output port to which an incoming flit must be sent based on the routing information contained in the header flit. In addition, the router issues appropriate select signals for the multiplexers.

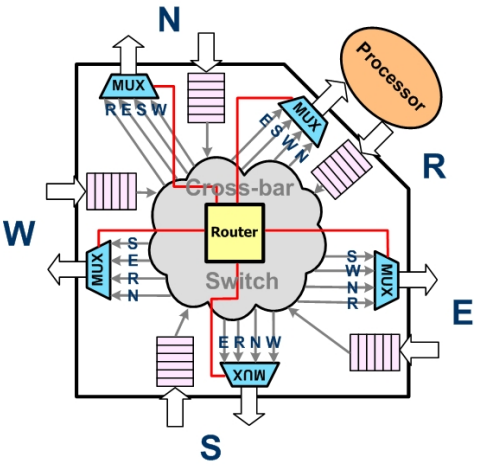

Figure 1: NoC Switch Architecture

5 System Level Fault Model for NoC Switch

Reference [Bengtsson, 06] presents a system level fault model based on the generic properties of a NoC switch functionality. We are using this fault model for online fault detection and diagnosis applications. The proposed model consists of five fault types:

1. **Dropped Data Fault**: The switch receives the data, but never sends it to the intended output port.
2. **Corrupt Data Fault**: The switch corrupts the incoming data and sends it to intended output port.
3. **Direction Fault**: The switch sends the incoming data to an output port other than the intended one.
4. **Multiple Copies in Space Fault**: The switch sends an incoming packet to the correct output port as well as another output port.
5. **Multiple Copies in Time Fault**: The switch sends multiple copies of an incoming packet to the intended output port throughout the time.

In this research, considering the above fault model, we propose a number of fault detection and diagnosis methods for online testing of NoC switches. Note that each of faults discussed above, may originate from more than one component of a switch. For example, we will show that Corrupt Data fault may occur in either of the router, FIFO buffers, crossbar switches, or the multiplexers in a switch. Different types of faults in a NoC switch are described in Sections 5.1 to 5.5.

5.1 Dropped Data Fault

Dropped data faults may occur in the FIFOs, the router, or the multiplexers of a switch. For describing the first case (faults in the FIFO) we need to explain the switch architecture with more details. For each FIFO in the switch, there are Head and Tail counters that point to the first and last locations of the FIFO respectively. Consider the case that the Head counter is faulty and its current value is increased by 2 or more each time a buffered flit is being removed from the head of FIFO to be sent to its appropriate destination. In this situation, some of the flits in the FIFO will be lost and never come out from the switch.

For the second case (faults in the router), assume that a fault has occurred in the router while the router is routing a buffered flit to its destination. Assume that the router removes the flit from the FIFO, but does not issue appropriate select signals for the multiplexers. In this situation, the flit has been removed from the FIFO, but has not been sent to any output port. Hence, the packet has been dropped and will never come out from the switch. For the third case (faults in the multiplexers), assume that the router has decided to send an incoming flit to a specific output port and has removed the flit from its corresponding FIFO. If the output multiplexer is faulty, e.g., one or more bits of its select signal are stuck at one or zero, one of its unintended inputs is selected and consequently that flit will be dropped.

For the second case (faults in the router), assume that a fault has occurred in the router while the router is routing a buffered flit to its destination. Assume that the

router removes the flit from the FIFO, but does not issue appropriate select signals for the multiplexers. In this situation, the flit has been removed from the FIFO, but has not been sent to any output port. Hence, the packet has been dropped and will never come out from the switch. For the third case (faults in the multiplexers), assume that the router has decided to send an incoming flit to a specific output port and has removed the flit from its corresponding FIFO. If the output multiplexer is faulty, e.g., one or more bits of its select signal are stuck at one or zero, one of its unintended inputs is selected and consequently that flit will be dropped.

5.2 Corrupt Data Fault

Corrupt data fault may happen in each of the four components of a switch (FIFO, crossbar switch, multiplexer or router). Faults in one or more bits of the FIFO memory may result in a Corrupt Data fault originating from a FIFO. For the crossbar switch, a similar explanation can be presented. For a multiplexer, its output can be faulty and this fault will result in sending data other than the incoming data to its output. For the router part, consider the case below. Assume that the router has received the header flit of a packet and has targeted an appropriate output port. Thus it sends the header flit as well as one or more of its following flits, but in the middle of its operation, a fault occurs and results in sending rest of the flits to another output port. We can call this fault a Corrupt Data fault since some of the flits of the packet have been distracted from the correct path from the source to the destination and will be lost in the following switches due to the lack of a header flit and appropriate routing information.

5.3 Direction Fault

Misdirection is the result of the faulty behavior of the switch router. Assume that the router has a faulty behavior and makes wrong decisions while routing its incoming data. This faulty behavior may result in issuing inappropriate select signals for the multiplexers and thus sending the data to an output port other than the one implied by the destination address of the packet.

5.4 Multiple Copies in Space Fault

Multiple Copies in Space fault can occur either in the router of a switch or its multiplexers. As discussed above, the faulty behavior of the router may result in issuing inappropriate select signals for the multiplexers. One of the consequences of this inappropriate signaling could be that two or more than one multiplexers select the same input port on their inputs; therefore, the data comes out from the correct output port as well as from one or more other output ports.

If an unintended multiplexer selects the same incoming data because of the faulty behavior, the same data will also come out from another output port. In this situation, Multiple Copies in Space fault has occurred.

5.5 Multiple Copies in Time Fault

Multiple copies in time faults are originated from the FIFOs contained in the switches. Consider a FIFO with a faulty "empty" signal. The false empty signal may cause the FIFO send its old data out of the switches port. This old data is usually an earlier packet's flits, and sending the same sequence of flits out of the switch's ports leads to repetition of a packet, i.e., multiple copies in time.

Table 1 summarizes the above discussion and shows the possible existing faults in different components of a NoC switch.

System Level Fault	Origin of Fault			
	Router	FIFO	Multiplexer	Crossbar Switch
Dropped Data				-
Corrupt Data				
Direction		-	-	-
Multiple Copies in Space		-		-
Multiple Copies in Time	-		-	-

Table 1: Different Types of Faults in NoC Switch Components

Note that fault types discussed above may overlap with each other, i.e., a single stuck-at fault at a wire in the gate-level implementation may appear as more than one system-level fault. For example consider a switch architecture in which each port has a pair of 8-bit data buses for exchanging packets with its neighboring switch. Assume that a packet has an 8-bit header and a data payload whose length is a multiple of 8 (as bits). While transmitting a packet, header byte will be sent on the 8-bit data bus followed by data payload bytes. Assume a stuck-at zero fault occurs on one of the lines of the data bus. If the value of the corresponding bit in the header of a packet crossing that bus is '0', only the payload of the packet will be affected which results in a corrupt data fault. On the contrary, if the value of the corresponding bit of the header is '1', two situations may arise: if the source or destination address value changes to a valid address in the NoC, a direction fault occurs. But, if that stuck-at fault results in an invalid destination address in the NoC, the packet is lost and a dropped data fault occurs.

Another point that is worth mentioning about our high level system faults is the level of defect coverage associated with such fault model. Correspondence with the actual faults can be regarded as a way of validating our high-level fault model. Each RTL component of a switch is targeted by at least one system-level fault model and a system-level fault corresponds to several RTL faults (such as crossbar switch faults, FIFO faults, etc.). We do not claim that the corresponding RTL faults consider all of the possible defects of a switch, but a justification similar to the one used for the stuck-at fault model can be made to show the acceptability of system-level fault models. At the gate level, it is true that all gate-level (stuck-at) faults do not cover

hundred percent of the physical faults. Similarly, hundred percent of RTL faults cover only a part of the gate-level faults (see Figure 2)

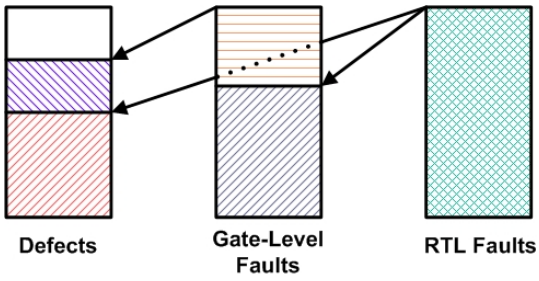

Figure 2: Correspondence of Fault Models at Different Levels of Abstraction

6 Proposed Online Fault Detection

In this section we introduce our different methods for online fault detection. The proposed methods cover a portion of the faults discussed in the previous section. We consider NoCs of different size and assume that the NoCs have two Input/Output Switches (IOS) at two opposite corners. Packets are generated at the IOSs' corresponding Processing Elements (PEs) and are sent to a mid-way PE. Then the packets are sent to the IOS at the opposite corner.

Our fault detection methods are: Distraction Detection, Switch Count [Alaghi, 07], Sequence Number and Cyclic-Redundancy Check.

6.1 Distraction Detection

Our online fault detection, called distraction detection, is applied to mesh-based NoCs with XY routing algorithm. As shown in Figure 3, in this method, first, the switches extract routing information from the receiving packets and following this, based on the XY routing algorithm it is decided whether the packet was supposed to pass through this switch or it has been distracted. This method was developed to detect direction faults. However it also detects some of the other faults as well.

In the 2-D mesh-based architectures the distraction detection method still leaves a few direction faults undetected. These faults cause a packet to be transferred between two neighboring switches forever. This packet might still be in the routing path and so the corresponding fault cannot be detected by the distraction detection method.

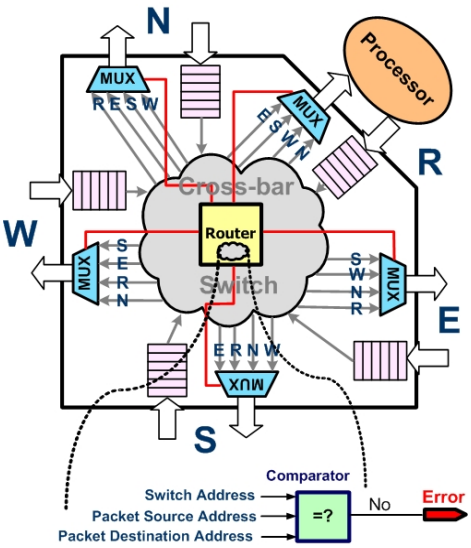

Figure 3: Self Testable Switch

As an example of shortcoming of this method, consider the 3x3 NoC shown in Figure 4. To clarify the proposed algorithms, an ID number has been assigned to each switch. Assume that only the processors related to switches 1 and 9 are externally accessible (IOS PEs). Assume that in this figure, switch 7 is faulty such that it always redirects all the receiving packets to switch 8. In this case, the direction fault of switch 7 cannot be detected by distraction detection method. The same scenario happens if switch 3 always redirects all the input packets to switch 2.

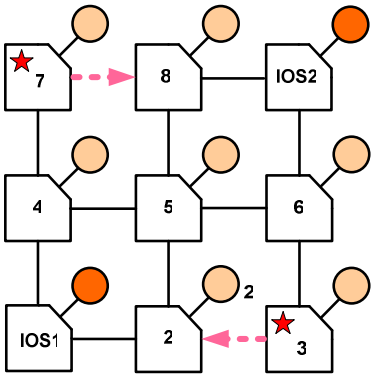

*Figure 4: A sample 3*3 NoC Architecture*

6.2 Switch Count

To enhance online fault detection in mesh-based NoCs, the switch count approach is proposed. In this method each packet includes a switch count field which is incremented automatically by 1 when the packet passes through a switch. The overflow of this field is the evidence of a fault in the NoC structure. For example, assume that a packet from IOS1 in Figure 4 is to go to IOS2. In this case if switch 3 is faulty, the packet is sent to switch 2 after it is received by switch 3, and it goes back and forth between switches 2 and 3 several times.

Since both of the Y locations of switch 2 and 3 are equal to the Y location of the source switch (switch 1), the direction fault of switch 3 is not detected by distraction detection method. However, using the switch count approach, the switch count field of the transmitting packet overflows due to the several rotation of the packet between switches 2 and 3. By applying the distraction detection method along with the switch count method, all the direction faults of all NoC switches can be detected.

6.3 Sequence Number

As mentioned before, the packets are generated in a PE connected to an IOS and are sent to a mid-way processor and then sent to the IOS on the opposite side. For switch fault detection with the Sequence Number method we assign a unique number to each packet generated at IOSs. These numbers are sequential and are sent as a flit of the packets. In this method, the sequence of the received packets is checked at the destination node. If a packet does not arrive at the destination or appears more than once then there exists a fault somewhere in the NoC. This method was mainly developed to detect dropped packets and multiple copies in time faults.

The additional hardware required for implementing this method uses the sequence number flit and has a watchdog counter at the IOSs' PEs to detect out of order appearance of sequence numbers of the received packets.

6.4 Cyclic-Redundancy Check

By adding a CRC flit to each packet while generated in IOSs' PEs, a corruption in the packet flits, including the control and data flits are detected at the destination by a simple CRC checking. To implement this method, the CRC circuit should be added to the IOSs' PEs and CRC flits must be included in the packets.

7 Fault Classification and Detection Methods

In this section, we classify the faults discussed in Section 5 into smaller groups and show how they are detected with different fault detection methods discussed in Section 6.

7.1 Misrouted Packets

The misrouted packets caused by the direction faults may reach the destination. These packets can only be detected by the distraction detection method. The misrouted packets that cannot be detected by distraction detection method and are sent back and

forth between NoC switches are finally detected by the switch count or by the sequence number method. The other misrouted packets that get trapped into the processing elements are only detected by the sequence number method.

7.2 Multiple Copies in Space Packets

The extra packets generated by a faulty switch may reach the destination. In this case the fault can be detected by both distraction detection method and the sequence number method.

If the extra packet wanders around the NoC, i.e., the packet goes back and forth between NoC switches, it can only be detected by the switch count method. However, if the packet gets trapped into a processing element, it may be detected by the sequence number method.

7.3 Multiple Copies in Time and Dropped Packets

The multiple in time packets and the dropped packets can only be detected by the sequence number method.

7.4 Corrupt Packets

If the corruption happens on the data flits, the fault can only be detected by the Cyclic-Redundancy Check at the destination. However, if the control flits get corrupted, they may also be considered as a misrouted packet and detected by the distraction detection method.

Table 2 shows the detection methods and the faults that can be detected by each method. As shown in this table, a detection method may be able to detect more than one type of fault. For example, the switch count method detects the wandering packets which go back and forth between two NoC switches due to a direction fault or a multiple copy in space fault. On the other hand, some faults can be detected by more than one detection method. For example, in the case of multiple copies in space packet, if the extra copy reaches the destination, the fault can be detected by the sequence number method as well as the distraction detection method.

8 Online Fault Diagnosis

The faults detected by the distraction detection method and the switch count method can be immediately diagnosed by the neighboring switch of the faulty switch. Assuming a single switch failure, when a switch catches a misrouted packet, it is obvious that the neighboring switch in the direction of the port from which the faulty packet arrives, is faulty. The faults detected by the sequence number or Cyclic-Redundancy Check, however, cannot be immediately located. In this case a series of switches are marked as suspicious to be faulty. The selected marked switches are the ones that lie in the path of the faulty packet. The list of the suspicious switches is updated every time another packet (faulty or non-faulty) arrives at IOSs. If a faulty packet arrives at the IOSs, the switches in its paths are added to the list of suspicious ones, and if a non-faulty packet arrives at IOSs, the switches in its paths are excluded

from the faulty list. For a single failure, the switch that appears in the list more than the other switches is the faulty switch.

A simple diagnosis scenario is shown in Figure 5. Assume that the switch designated with the star is faulty and it corrupts the data that passes through it. Two test packets are sent from IOS1 towards IOS2; one is to be processed by PE1 and the other to be processed by PE2. One packet travels the path shown by the dotted arrows and the other travels the solid arrow path. Since both paths pass through the faulty switch, IOS2 marks all switches except IOS1 and IOS2 along the dotted and solid paths as suspicious. Since the star switch is the only switch that is marked twice, it is confirmed by IOS2 as the faulty switch.

As another example, in Figure 5 consider the case in which after the packet that is processed by PE1, two other packets that are going to be processed by PE3 and PE4 are coming in. Since these packets do not pass the faulty switch, they are reported as non-faulty and after excluding the switches in their paths from the switches shown in dotted arrows, the star switch is diagnosed.

System Level Faults	Detection Method			
	Distraction Detection	Switch Count	Sequence Number	CRC
Dropped Data	-	-		-
Corrupt Data		-	-	
Direction				-
Multiple Copies in Space				-
Multiple Copies in Time	-	-		-

Table 2: Detection Methods for System Level Faults in Different Parts of a NoC Switch

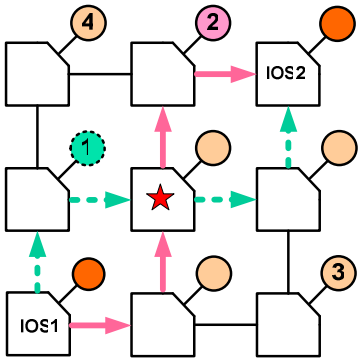

Figure 5: Fault Diagnosis

9 Experimental Results

The proposed online test strategy was evaluated for different NoC sizes, ranging from 3x3 to 10x10 with respectively 9 and 100 processors and switches assuming various packet traffics. Our online testing and diagnosis methods were simulated in a high level VHDL based platform [Sedghi, 07]. In the selected NoCs, random packets were routed through the NoCs and the fault coverage (the number of detected faults over the number of all injected faults) was reported.

The results of applying our first online test method (distraction detection method) to the discussed NoCs are given in Table 3. In this table "addressed switch" is the number of switches that have been addressed as the destination for random generated packets over the total number of switches, i.e., in the 100% addressed switch scenario, all NoC switches become the destination of at least one random generated packet.

Addressed Switch / NoC Architecture	25%	50%	75%	100%
3x3	30	45	67	67
5x5	36	57	64	74
7x7	42	63	71	76
10x10	48	67	72	78

Table 3: Distraction Detection Fault Coverage Results (%)

Table 4 shows the resulting fault coverage after applying the distraction detection method along with the switch count method to the test case NoCs. Since the packets are randomly generated, the fault coverage of each experiment may differ from other experiments. In this research we have generated five different sets of packets and reported the average fault coverage in Tables 3 and 4. As illustrated in these tables, more than 20% of the misrouted-packet faults are hard-to-detect even in the case of 100% addressed switch. These faults are the ones that cause the packets to get trapped into the processors, and as mentioned in Section 7.1, can be considered as dropped packets and are detected by the sequence number method.

Addressed Switch / NoC Architecture	25%	50%	75%	100%
3x3	39	51	73	73
5x5	44	65	69	76
7x7	52	75	77	77
10x10	61	77	78	78

Table 4: Distraction Detection and Switch Count Fault Coverage Results (%)

The Sequence Number method discussed earlier was mainly developed to detect dropped packets in the NoC. The fault coverage simulation results of applying the Sequence Number method to a 5x5 NoC is shown in Table 5. Note that the 500% packet ratio was simulated in order to make sure that each switch gets a packet more than once.

The faults originated from FIFOs and Multiplexers are classified as "Port Faults" and the faults caused by routers and crossbar switches are classified as "Internal Faults".

Addressed Switch / Faults	500%	75%	50%	25%
Port Faults	100	81.14	71.43	55.81
Internal Faults	100	100	100	88.89

Table 5: Sequence Number Fault Coverage Results (%)

As discussed before, the Cyclic-Redundancy Check (CRC) method is mainly used to detect the corruption faults. Table 6 shows the coverage results of applying this method to a 7x7 NoC.

Addressed Switch / Faults	500%	75%	50%	25%
Port Faults	99.9	83.23	75.3	56.09
Internal Faults	100	100	100	98.67

Table 6: CRC Fault Coverage Results (%)

If the Input/Output Switches drop the packets, they cannot be diagnosed with our method, since the entire packets pass through the IOSs. Accordingly, the nearer a switch to the IOS, the harder the diagnosis. Tables 7, 8, 9 and 10 show the diagnosis results for different faults on NoCs. As shown in these tables, if enough packets travel through the NoC, almost all faults can be diagnosed. The undiagnosed faults can be categorized to two groups: the first group includes the direction faults that are not detected by our distraction detection method, but are categorized as dropped faults and are covered by the other discussed methods. The second group includes faulty IOSs. 22% of the faults in a 3x3 NoC relate to IOSs (2 out of 9 switches), similarly in a 10x10 NoC, IOSs are responsible for 2% of faults. Excluding these hard to detect/diagnose faults, our proposed diagnosis method shows a great performance.

Addressed Switch / Faults	500%	75%	50%	25%
Direction Faults	67	67	45	30
Dropped Faults	73.34	38.67	29.87	17.45
Corruption Faults	77.78	70.22	58.89	33

Table 7: Diagnosis Rate for a 3x3 NoC (%)

Addressed Switch / Faults	500%	75%	50%	25%
Direction Faults	74	64	57	36
Dropped Faults	91.73	44.28	31.79	15.15
Corruption Faults	92	91.36	87.68	61.12

Table 8: Diagnosis Rate for a 5x5 NoC (%)

Addressed Switch / Faults	500%	75%	50%	25%
Direction Faults	76	71	63	32
Dropped Faults	95.22	42.84	31.55	16.83
Corruption Faults	95.92	95.43	94.45	80

Table 9: Diagnosis Rate for a 7x7 NoC (%)

Addressed Switch / Faults	500%	75%	50%	25%
Direction Faults	78.26	78.26	78.26	76.87
Dropped Faults	97.40	40.67	28.88	15.88
Corruption Faults	98	97.92	97.76	93.03

Table 10: Diagnosis Rate for a 10x10 NoC (%)

As shown in the above tables, the NoC dimensions affect the results of the fault detection and diagnosis methods. For the larger NoCs, even a small number of packets cover many faults and lead to a higher fault coverage. However, in fault diagnosis, the situation is different. Since the NoCs of larger dimensions have a larger number of switches, it becomes harder to diagnose the faulty switch among them while sending a limited number of packets. Figure 6 shows the effect of the NoC dimensions on the coverage results while detecting and diagnosing the dropped faults in two high-traffic

(75% packet ratio) and low-traffic (25% packet ratio) cases. As shown in this figure in both 25% and 75% addressed switch cases, the diagnosis rate is reduced by increasing the NoC sizes since fault diagnosis gets harder for larger NoCs while sending a limited number of packets. However as shown in Tables 7-10, considering 500% packet ratio the fault diagnosis of larger NoCs are better than smaller ones since in this case the probability that all the switches in the NoC are being targeted via at least one packet is more than the cases with limited packet ratios. In addition as discussed above, in this case all the faulty switches expect faulty IOSs can be diagnosed.

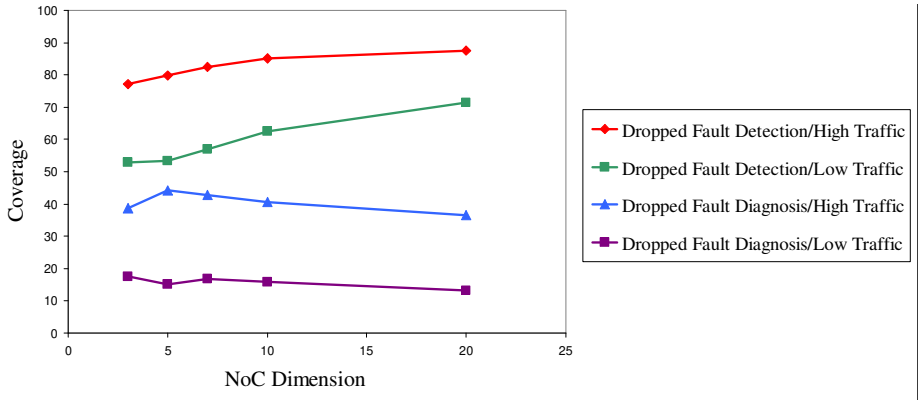

Figure 6: Effect of NoC Dimensions on Fault Coverage and Diagnosis Rate

To evaluate the hardware overhead imposed by applying our methods, we applied the proposed methods to an existing NoC switch [Hosseinabadi, 07] and synthesized it. Table 11 shows the area and communication overhead imposed by applying the proposed methods to this switch. The last column in Table 11 shows the number of the flits added to each packet in our fault detection and diagnosis methods. As discussed in Section 6, no flit is added to the packets in distraction detection method.

Overhead / Detection Method	Switch Area Overhead (%)	Packet Flit Overhead
Distraction Detection Method	1.6	0
Switch Count Method	9.3	1
Sequence Number Method	0	1
CRC Method	0	2

Table 11: Area and Communication Overhead

10 Conclusion

This paper presented an online fault detection and fault location method for NoC switches. Various forms of functional switch faults were considered in this research, including dropped and corrupted data faults, direction faults, and faults resulting in multiple copies of packets in time and space. For each of these faults an error detection and diagnosis method has been proposed. The proposed algorithms have been evaluated in terms of fault coverage and the necessary area overhead. The experimental results show that with a relatively low area overhead, a large number of NoC switch faults can be detected and diagnosed. The steps taken in this work are essential for design of reliable NoC structures.

References

[Aktouf, 02] Aktouf, C.: A Complete Strategy for Testing an on-chip Multiprocessor Architecture, IEEE Trans. on Design and Test of Computers, Vol. 19-1, 2002, 18-28.

[Alaghi, 07] Alaghi, A., Karimi, N., Sedghi, M., and Navabi, Z.: Online NoC Switch Fault Detection and Diagnosis Using a High Level Fault Model, In Proc. IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT), 2007, 21-29.

[Amory, 05a] Amory, A. M., Briao, E., Cota1, E., Lubaszewski, M.S, and Moraes, F. G.: A Scalable Test Strategy for Network-on-Chip Routers, In Proc. International Test Conference (ITC), 2005.

[Amory, 05b] Amory, A. M., Brião1, E. W., Cota1, É. F., Lubaszewski, M. S., and Moraes, F. G.: A Cost-Effective Test Flow for Homogeneous Network-on-Chip, In Proc. European Test Symposium (ETS), 2005.

[Beigne, 05] Beigne, E., Clermidy, F., Vivet, P., et al.: An Asynchronous NoC Architecture Providing Low Latency Service and Its Multi-Level Design Framework, In Proc. 11th International Symposium on Asynchronous Circuits and Systems (ASYNC), 2005, 54–63.

[Bengtsson, 06] Bengtsson, T., Kumar, S., and Peng, Z., Application Area Specific System Level Fault Models: A Case Study with a Simple NoC Switch, In Proc. (electronic) International Design and Test Workshop (IDT), 2006.

[Benini, 02] Benini, L., and De Micheli, G.: Networks on Chips:A Paradigm, IEEE Trans. on Computer, Vol. 35, Jan 2002, 70-78.

[Bertozzi, 05] Bertozzi, D., Jalabert, A., Murali, S., et al.: NoC Synthesis Flow for Customized Domain Specific Multiprocessor Systems-onchip, IEEE Trans. on Parallel and Distributed Systems, Vol. 16, No. 2, 2005, 113–129.

[Bhojwani, 07] Bhojwani, P., Mahapatra, R. N., Mahapatra: A Robust Protocol for Concurrent On-Line Test (COLT) of NoC-based Systems-on-a-Chip, In Proc. Design Automation Conference, 2007, 670-675.

[Bjerregaard, 06] Bjerregaard, T., and Mahadevan, S.: A Survey of Research and Practices of Network-on-Chip, ACM Computing Surveys, Vol. 38, No. 1, 2006, 1-51.

[Cota, 03a] Cota, E., Kreutz, M., et al.: The Impact of NoC Reuse on the Testing of Core-based Systems, In Proc. IEEE VLSI Test Symposium, 2003, 128–133.

[Cota, 03b] Cota, E., et al.: Power aware NoC Reuse on the Testing of Core-Based Systems, In Proc. International Test Conference (ITC), 2003, 612-621.

[Cuviello, 99] Cuviello, M., Dey, S., Bai, X., and Zhao, Y.: Fault Modelling and Simulation for Crosstalk in System-on-Chip Interconnects, In Proc. IEEE/ACM International Conference on Computer-Aided Design, 1999, 297-303.

[Dally, 01] Dally, W. J., and Towles, B.: Route Packets, Not Wires: On-Chip Interconnection Networks, In Proc. Design Automation Conference (DAC), 2001, 683--689.

[European Commission, 01] European Commission, Technology Roadmap for Nanoelectronics, 2001.

[Grecu, 06a] Grecu, C., Ivanov, A., Saleh, R., et al.: On-line Fault Detection and Location for NoC Interconnects, In Proc. 12th IEEE International Symposium on On-Line Testing (IOLTS), 2006, 145-150.

[Grecu, 06b] Grecu, C., Pande, P., Ivanov, A., and Saleh, R.: BIST for Network-on-Chip Interconnect Infrastructures, In Proc. 24th IEEE VLSI Test Symposium (VTS'06), 2006, 30-35.

[Grecu, 07] Grecu, C., Ivanov, A., Saleh, R., and Pande, P. P.: Testing Network on Chip Communication Fabrics, IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems (TCAD), Vol. 26, No. 12, 2201-2214, 2007.

[Hansson, 05] Hansson, A., Goossens, K., and Radulescu, A.: A Unified Approach to Constrained Mapping and Routing on Network-on-Chip Architectures, In Proc. 3rd Hardware/Software Codesign and System Synthesis (CODES+ISSS '05), 2005, 75–80.

[Ho, 01] Ho, R., Mai, K. W., and Horowitz, M. A.: The Future of Wires, In Proc. Proceedings of the IEEE, 2001, 490–504.

[Hosseinabady, 06] Hosseinabady, M., Banaiyan, A., Bojnordi, M. N., and Navabi, Z.: A Concurrent Testing Method for NoC Switches, In Proc. Design Automation and Test in Europe (DATE), 2006, 1171-1176.

[Hosseinabadi, 07] Hosseinabadi, M., Dalirsani, A., and Navabi, Z.: Using the Inter- and Intra-Switch Regularity in NoC Switch Testing, In Proc. Design Automation and Test in Europe(DATE), 2007, 361-366.

[Jantsch, 03] Jantsch, A., and Tenhunen, H., Networks on Chip, Kluwer Academic Publishers, 2003.

[Keutzer, 00] Keutzer, K., Malik, S., Rabaey, J. M., and Sangiovanni-Vincentelli, A.: System-level Design: Orthogonalization of Concerns and Platform-Based Design,

IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, Vol. 19, No. 12, 2000, 1523–1543.

[Kim, 05] Kim, J., Park, D., Nicopoulos, C. et al. : Design and Analysis of an NoC Architecture from Performance, Reliability and Energy Perspective, In Proc. Symposium on Architecture for Networking and Communications Systems (ANCS), 2005, 173-182.

[Kim, 06] Kim, J., Nicopoulos, C. A., Park, D., Vijaykrishnan, N., Yousif, M. S., and Das, C. R.: A Gracefully Degrading and Energy-Efficient Modular Router Architecture for On-Chip Networks, In Proc. 33rd Annual International Symposium on Computer Architecture (ISCA), 2006, 4-15.

[Li, 06] Li, M., Jone, W. B, Zeng, Q. A.: An Efficient Wrapper Scan Chain Configuration Method for Network-on-Chip Testing, In Proc. Emerging VLSI Technologies and Architectures (ISVLSI'06), 2006.

[Liu, 05] Liu, C., Iyengar, V., Shi, J., and Cota, E.: Power-Aware Test Scheduling in Network-on-Chip Using Variable-Rate On-Chip Clocking, In Proc. VTS, 2005, 349-354.

[Marculescu, 06] Marculescu, R., Ogras, U. Y., and Zamora, N. H.: Computation and Communication Refinement for Multiprocessor SoC Design: A System-Level Perspective, ACM Trans. on Design Automation of Electronic Systems, Special Issue on Novel Paradigms in System-Level Design, Vol. 11, No. 3, 2006, 564-592.

[Mello, 05] Mello, A., Tedesco, L., Calazans, N., and Moraes, F.: Virtual Channels in Networks on Chip: Implementation and Evaluation on Hermes NoC, In Proc. Integrated Circuits and Systems Design, 2005, 178-183.

[Murali, 05a] Murali, S., Benini, L., and Demicheli, G.: Mapping and Physical Planning of Networks-on-Chip Architectures with Quality-of Service Guarantees, In Proc. Asia and South Pacific Design Automation Conference (ASP-DAC '05), Vol. 1, 2005, 27–32.

[Murali, 05b] Murali, S., De Micheli, G., Benini, L., et al.: Analysis of Error Recovery Schemes for Networks on Chips, IEEE Trans. on Design and Test of Computers, Vol. 22, No. 5, 2005, 434-442.

[Nahvi, 04] Nahvi, M., and Ivanov, A.: Indirect Test Architecture for SoC Testing, IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, Vol. 23, No. 7, 2004, 1128-1142.

[Nikolic, 01] Nikolic, K., Sadek, A., and Forshaw, M.: Architectures for Reliable Computing with Unreliable Nanodevices, In Proc. IEEE NANO, 2001, 254.259.

[Orgas, 06] Orgas, U. Y., and Marculescu, R.: Communication-Based Design for Nanoscale Socs, VLSI Handbook, Wai-Kai Chen(ed.), 2nd Edition, CRC Book Press, 2006, Chapter 16.

[Pande, 05] Pande, P. P., De Micheli, G., Grecu, C., et al.: Design, Synthesis, and Test of Networks on Chips, IEEE Trans. on Design and Test of Computers, Vol. 22, No. 5, 2005, 404-413.

[Park, 06] Park, D., Nicopoulos, C., Kim, J., et al.: Exploring Fault-Tolerant Network-on-Chip Architectures, In Proc. International Conference on Dependable Systems and Networks (DSN'06), 2006, 93-104.

[Raik, 06] Raik, J., Govind, V., and Ubar, R.: An External Test Approach for Network-on-a-chip Switches, In Proc. 15th Asian Test Symposium (ATS), 2006, 437-442

[Sedghi, 07] Sedghi, M., Alaghi, A., Koopahi, E., and Navabi, Z.: An HDL-Based Platform for High Level NoC Switch Testing, In Proc. Asian Test Symposium (ATS), 2007, 453-458.

[Tran, 06] Tran, X. T., Durupt, J., Bertrand, F., et al.: A DFT Architecture for Asynchronous Networks on-Chip, In Proc. 11th IEEE European Test Symposium (ETS'06), 2006, 219-224.

[Ubar, 03] Ubar, R., and Raik, J.: Testing Strategies for Network on Chip, Networks on Chip, A. Jantsch and H. Tenhunen (ed.), Kluwer Academic Publisher, 2003, 131-152.

[Vermeulen, 03] Vermeulen, B., Dielissen, J., Goossens, K., and Ciordas, C.: Bringing Communication Networks On-Chip: The Test and Verification Implications, IEEE Communications Magazine, Vol. 41, No. 9, 2003, 74-81.