

BIST for Network-on-Chip Interconnect Infrastructures

Cristian Grecu¹, Partha Pande², André Ivanov¹, Res Saleh¹

¹ SoC Research Laboratory
Department of Electrical and Computer Engineering
University of British Columbia
2356 Main Mall Vancouver, BC, V6T 1Z4 Canada
{grecuc, ivanov, res}@ece.ubc.ca

² School of Electrical Engineering & Computer Science
Washington State University PO Box 642752
Pullman, WA 99164-2752, USA
pande@eecs.wsu.edu

Abstract

In this paper, we present a novel built-in self-test methodology for testing the inter-switch links of network-on-chip (NoC) based chips. This methodology uses a high-level fault model that accounts for crosstalk effects due to inter-wire coupling. The novelty of our approach lies in the progressive reuse of the NoC infrastructure to transport test data to its own components under test in a bootstrap manner, and in extensively exploiting the inherent parallelism of the data transport mechanism to reduce the test time and implicitly the test cost.

Keywords: *built-in self-test, network-on-chip, interconnect infrastructure, unicast test, multicast test.*

1. Introduction and background

Networks on Chip (NoC) [1] [2] interconnect infrastructures are emerging as a new paradigm that characterizes the on-chip data communication architecture of large Systems-on-Chip (SoCs). In general, two types of NoC architectures have been proposed to date for implementation: (i) regular interconnection structures derived from parallel computing, and (ii) irregular, custom-built NoC fabrics. Kumar [3] proposed a mesh-based interconnect architecture named CLICHÉ. Dally et al. [4] suggested a torus-based architecture. Both of these architectures consist of an $m \times n$ mesh of switches interconnecting computational resources (IPs) placed along with the switches. Guerrier and Greiner [5] proposed the use of a tree-based interconnect (SPIN) and addressed system level design issues. In [6] and [7], the authors describe an interconnect architecture based on the Butterfly-Fat-Tree (BFT) topology for a networked SoC, as well as the associated design of the required switches and addressing mechanisms.

□ - Functional core
■ - Switch

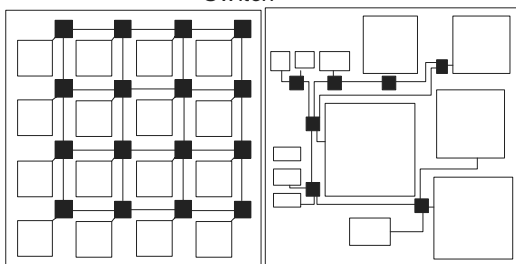


Figure 1: a) Regular, mesh-based NoC [3]; b) Irregular NoC [8]

Benini et al. [8] introduced application specific irregular architectures to build networked SoCs. The common feature of all these interconnect fabrics is that the functional IP blocks communicate with each other through a highly

structured switched medium, consisting of intelligent switches and inter-switch links.

Any new design methodology will only be widely adopted if it is complemented by efficient test mechanisms. To date, the NoC test methodologies are mainly concerned with the testing of the functional IP cores, using the communication infrastructure as a Test Access Mechanism (TAM) [9]. Our work complements this approach by developing the test strategy for the interconnect infrastructure itself. The test strategy of NoC-based interconnect infrastructures needs to address two problems [10]: (i) testing of the switch blocks; (ii) testing of the inter-switch interconnect segments. In [11] we proposed two schemes for testing the combinational blocks of the switches, based on uni- and multi-cast transmission of test-data packets through the switching fabric. Here, we specifically propose and evaluate a test methodology suitable for the NoC inter-switch links.

The NoC interconnects are characterized by poor controllability and observability, due to the fact that they are deeply embedded and spread across the chip. Pin-count limitations restrict the use of I/O pins dedicated for the test of the different components of the data-transport medium; therefore, we propose a built-in self-test (BIST) scheme for the test of the inter-switch links that takes into account fault models specific for the deep-submicron (DSM) technologies.

2. Related work

Since the inception of SoC designs, the research community has targeted principally the testing of the IP cores, giving little emphasis to the testing of their communication infrastructures. Among the different test access mechanisms, TestRail [12] can be mentioned as one of the first to address core-based test of SoCs. After the emergence of the NoC design paradigm, different research groups proposed the reuse of the communication infrastructure as a TAM [9] [12]. In [13] the authors assumed the NoC fabric as fault-free and subsequently used it to transport test data to the functional blocks; however, for large systems, this assumption can be unrealistic, considering the complexity of the design and communication protocols. In [14], the authors proposed a dedicated TAM based on an on-chip network, where network oriented mechanisms were used to deliver test data to the functional cores of the SoC. In [15], Cuvillo et al. proposed a novel fault model for the global interconnects of DSM SoCs that accounts for cross-talk effects between a set of aggressor lines and a victim line. This fault model is referred to as Maximum Aggressor Fault (MAF) and it occurs when the signal transition on a *single* interconnect line (called the *victim line*) is affected through cross-talk by transitions on *all* the

other interconnect lines (called the *aggressors*) due to the presence of the crosstalk effect. In this model, all the aggressor lines switch in the same direction simultaneously.

The objective of this work is to develop an efficient, low overhead BIST methodology for testing the NoC interconnects based on DSM fault models.

2.1 MAF model and tests

The *Maximal Aggressor Fault* [15] model is an abstract representation of the set of all defects that can lead to one of the six crosstalk errors: rising/falling delay, positive/negative glitch, and rising/falling speed-up. The possible errors corresponding to the MAF fault model are presented in Fig. 2 for a link consisting of 3 wires. The signals on lines Y_1 and Y_3 act as aggressors, while Y_2 is the victim line. The aggressors act collectively to produce a delay, glitch or speed-up on the victim.

This abstraction covers a wide range of defects including design errors, design rules violations, process variations and physical defects. For a link consisting of N wires the MAF model assumes the worst-case situation with one victim line and $(N-1)$ aggressors. For links consisting of a large number of wires, considering all such variations is prohibitive from a test coverage point of view [16].

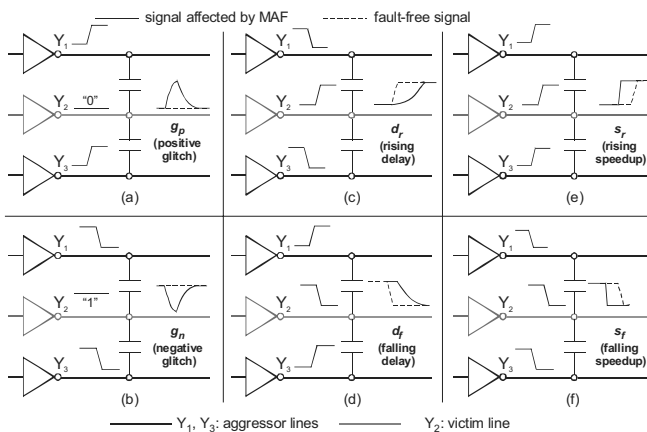


Figure 2: Possible crosstalk errors under the MAF model [15]

The transitions needed to sensitize the MAF faults can be easily derived from Fig. 2. For an inter-switch link consisting of N wires, a total of $6N$ faults need to be tested, requiring $6N$ 2-vector tests. These $6N$ MAF faults cover all the possible process variations and physical defects that can cause any crosstalk effect on any of the N interconnects. They also cover more traditional faults such as stuck-at, stuck-open and bridging faults. Since the corresponding $6N$ tests are known a priori, it is reasonable to use self-test structures that generate the MAF-required tests, with the additional advantage of being able to perform at-speed testing of NoC links.

We validated the accuracy of the MAF models at the link level through SPICE-level simulation, using typical inter-wire coupling capacitances and wire resistances as provided by [21]. However, SPICE-level simulation for a realistic NoC fabric is not feasible; hence, we developed a

high-level simulation method, using combined behavioural/structural HDL models of the inter-switch links and test circuitry. The simulation framework we developed to integrate the MAF fault model with our built-in self test methodology is illustrated in Fig. 3.

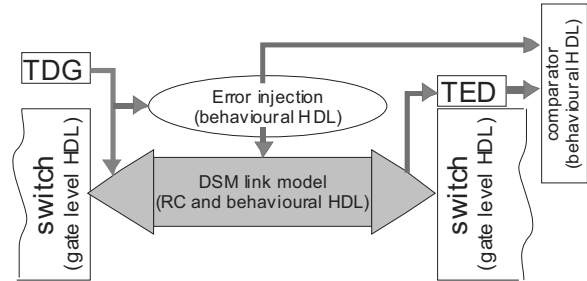


Figure 3: NoC BIST simulation environment with MAF error injection, test data generator/error detector and link under test

In our experiments crosstalk defects were injected into the NoC links under test by introducing behavioural-level interconnect errors that model the DSM crosstalk effects. Different behavioural errors were generated, corresponding to each of the MAF faults shown in Fig. 2. We refer to these as *MAF errors*. The effects of process variations beyond admissible limits on the characteristics of the MAF faults were modeled by varying the temporal properties of the MAF errors. For example, the width of the positive/negative glitches and the duration of rising/falling delays can be varied to simulate the variation of inter-wire mutual coupling capacitances due to process variations. Depending on the test vectors and process parameters variation, the error generator injects crosstalk errors, which are subsequently captured by the test error detector (TED) block.

3. Test requirements

The inter-switch links of a NoC system can be broadly classified as either unidirectional or bidirectional. Their test requirements and test implementation differ accordingly.

Unidirectional transactions

In most of the examples found in practice and literature, the NoC links are unidirectional, in which case the implementation of the self-test structures is simple, a test-generator/test-detector pair sufficing for each link.

Bidirectional transactions

The crosstalk effects may vary when the interconnect lines are driven in different directions, the main reasons being transmission line effects and variations in drivers and loads. Consequently, if in normal operation the inter-switch links are bidirectional, the test has to be conducted in both directions.

Switch-to-switch transactions

The switch-to-switch links are exposed to crosstalk errors due to the fact that they are the longest global wires on the chip (with the exception of clock and power lines). Therefore, it is paramount that the inter-switch links be tested against the crosstalk effects that can occur during the transport of data from one switch to another.

4. Self-test structures

The basic building blocks of our BIST methodology are the test data generator (TDG) and the test error detector

(TED) circuits. The TDG produces the test sequence consisting of 2-bit patterns as explained in Section 2.1. The test vectors are launched on the link under test at the normal operating clock speed from the transmitter side of the link, and then sampled and compared for logical consistency at the receiver side of the link by the TED circuit. In our implementation, the TED samples the values on the inter-switch links on the next active clock edge, thus allowing at most one clock cycle for the test data to travel from the transmitter to the receiver end [23]. Due to the fact that the crosstalk effects are highly dependent on the size of drivers and loads, the test data must be launched before the link drivers, and sampled after the buffers on the receiver side.

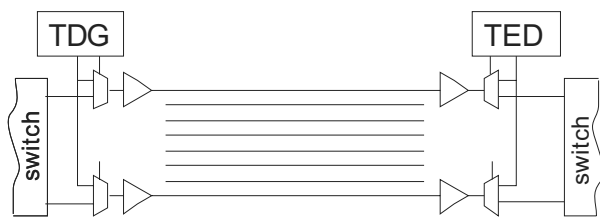


Figure 4: Test data generator (TDG) and error detector (TED)

In the more general case when, due to clock cycle constraints, the inter-switch links are pipelined using techniques such as FIFO insertion or flop-repeated wires [24], the data may require more than one clock cycle to travel from the transmitter switch to the destination switch. Even in these cases, the occurrence of crosstalk effects can still negatively affect the correct functionality and timing characteristics of the NoC fabrics, making crosstalk-aware testing a must-have feature.

4.1 Test data generator

Each possible MAF on a victim line of a link requires a two-vector test sequence to be sensitized. The test sequence exhibits some useful properties which allow for a compact and efficient design of the TDG/TED circuits:

- For each test vector, the logic values on the aggressor lines are the opposite of that on the victim line;
- After having applied the exhaustive set of test sequences for a particular victim line, the test sequence of the adjacent victim line can be obtained by shifting (rotating) the test data by exactly one bit.

The transition from one test vector to another can be concatenated such that the number of test vectors needed to sensitize the MAF faults can be reduced from twelve vectors per wire to eight, as shown in Fig. 5.

The test data analyzer is designed based on these two properties, and consists of a finite state machine and a barrel-shifter controlled by a victim line counter. The state machine generates the logical values corresponding to the MAF tests in eight distinct states s1 to s8. In an s1-to-s8 cycle, the state machine produces eight vectors. During each cycle, one line is tested and is assigned the victim logic values, while the rest of the lines get the aggressor values. The selection of the victim wire is achieved by the victim line counter which controls the barrel shifter such that for the first eight test cycles, the first wire of the link is configured as the victim; during the second set of eight test cycles, the second wire is configured as the victim, and so on and so forth. After each eight-vector

sequence, the barrel-shifter shifts by one bit position, and an identical eight-vector sequence is applied now with a new corresponding wire acting as the victim. This procedure continues until all the lines of the link are exhausted. The state machine starts from the idle state s0 and executes a transition s0-to-s1 as soon as it gets an active T_start signal. For each individual line, the state machine cycles through states s1 – s8, and returns to the idle state s0 upon completion of N eight-vector sequences.

According to the specific design style of the inter-switch links, only a subset of the MAF crosstalk effects may be of interest. In these cases, the state machine that generates the MAF tests will only possess the appropriate subset of states.

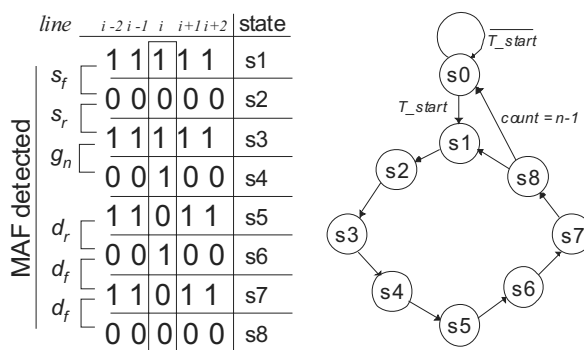


Figure 5: a) Test sequence for wire i ; b) State machine for MAF patterns generation

4.2 Test error detector

The task of the error detector is to verify the correctness of each transition generated by the two-vector pairs for each victim line, for all the MAF cases (i.e., d_f , d_r , g_p , g_m , s_r , s_f). In order to assess the correctness of the transitions, the error detector block generates locally a set of N eight-vector patterns identical to the one generated by the TDG block, but delayed by one clock cycle. This ensures that, for the d_r and d_f MAFs, the maximum allowable delay of a signal on any of the victim wires is less than one clock cycle. The logical comparison is carried out by an XOR network that compares the data sampled at the receiver with the data generated locally. If the two sets do not coincide, an error flag is raised marking that the link failed the MAF test.

In the more general case where the inter-switch lines are multi-cycle paths, the TED circuits must generate the MAF tests with a delay equal to the pipeline length of the inter-switch link. The TED circuits may differ with respect to the amount of delay they introduce (in multiples of the clock cycle); however, the implementation can remain essentially identical to the one in Fig. 5.

We developed and simulated parameterized VHDL models of the TDG and TED blocks explained above. These models are parameterized in terms of the number of lines per inter-switch link, and can be easily instantiated to fit different width requirements of a NoC chip. We synthesized and verified our designs using an ASIC standard cell design flow based on a TSMC 0.13 μm CMOS technology. Table 1 reports the area overhead of these blocks in terms of minimum-size 2-input NAND gates, for different link widths.

Table 1: Test area overhead of basic BIST blocks

Link width (bits)	TDG (2-input NAND gates)	TED (2-input NAND gates)
16	287	314
32	471	506
64	812	886

For comparison, the typical size of a NoC switch including the FIFO buffers is around 30K gates [6], resulting in a very low extra silicon area for MAF self-test implementation.

5. Distributed/concurrent test of NoC inter-switch links

So far in this paper, we have presented the fault model and the basic hardware blocks required for a self-test implementation of the NoC inter-switch links. A system-wide test implementation has to satisfy the specific requirements of the NoC fabric and exploit its highly parallel and distributed nature for an efficient realization. In fact, it is advantageous to combine the testing of the NoC inter-switch links with that of the other NoC components (i.e., the switch blocks) in order to reduce the total silicon area overhead. In this view, we suggest that the MAF test vectors be organized in test packets consisting of the actual patterns (derived according to Fig. 2), control signals, and a header containing the routing information, as indicated in Fig. 6.

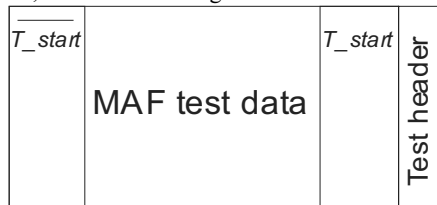


Figure 6: MAF test packet

Next, we propose three different strategies for implementing the self-test methodology based on the MAF fault model.

5.1 Point-to-point MAF self test

This is a simple, straightforward realization of the MAF BIST in which each link is tested with the help of a pair of TDG/TED blocks placed at the transmitter and receiver ends, respectively. For each link, the test configuration is similar to the case shown in Fig. 3.

The advantages of this method are ease of implementation (no other control circuitry is required) and short test time, since all the links can be tested simultaneously. One disadvantage of this approach is the amount of silicon real-estate that it consumes, since it requires one TDG/TED pair per link.

5.2 Interleaved unicast MAF test

The use of one TDG for each link can be eliminated by interleaving the test of the NoC links and the test of the switches. In [11] we presented a test data transport mechanism for delivering test patterns from an on-chip or off-chip test source to the NoC switches. The test data was generated off-line and transported in a sequential-recursive manner, using the upstream tested switches to transport test data to the downstream switches-under-test. The switches have two modes of operation: the *test mode*, in which test data is

scanned-in/captured/scanned-out, and a *normal mode* in which test patterns are transported towards the next switch to be tested. The idea behind this approach is to use the NoC infrastructure progressively for testing its own elements in a recursive manner. This strategy eliminates the need for additional test access mechanisms for delivering test data to the NoC components under test. This principle can be extended by combining the MAF test sequences with the test packets directed to the switches. As shown in Fig. 7, a *Global Test Controller (GTC)* block injects the test packets intended to the NoC switches, and interleaves those with the MAF test packets. Thus, a single TDG block is required to generate the MAF test patterns for all inter-switch links. The MAF test data is organized in test packets and a header is appended that identifies the link to be tested. The header also contains a field for carrying the T_start signal, and the routing information, which describes a path along previously tested, fault-free switches and links.

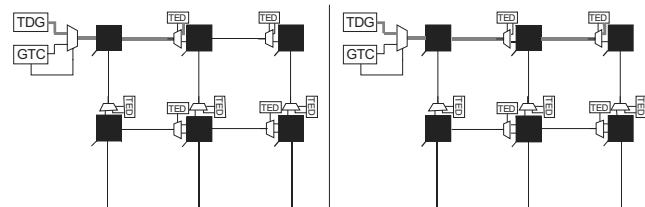


Figure 7: Interleaved unicast MAF test (for simplicity, only a subset of uni-directional links is shown)

As soon as the test packet reaches the link-under-test, the header is discarded, the TED circuit is enabled (T_start signal is active) and starts comparing the MAF data generated locally with the incoming test vectors. Whenever a mismatch is encountered, the error flag is raised and the test procedure is halted.

5.3 Interleaved multicast MAF test

The inherent parallelism of the NoC communication fabrics can be exploited with the goal of achieving a shorter test time while maintaining a reduced test area overhead. Instead of delivering the MAF test packets in a serial fashion, multiple packets can be sent over non-overlapping links, such that more than one link at a time is subjected to the test procedure.

We will refer to this type of MAF test data transport as *multicast MAF test*, since a single TDG circuit is used, multiple links are tested at the same time, and MAF test data travels toward the links under test on multiple parallel paths. In order to support multicast MAF test, the NoC must have built-in multicast capabilities. This feature is relatively easy to implement and it is discussed in more detail in [19] and [20]. For the sake of completeness, here we briefly present the main features and requirements of a NoC multicast implementation.

We consider the case of a wormhole routed NoC, where the data packets are divided into smaller fixed length units, called flits (flow control units). The header flit of a packet contains the routing information. Upon arrival at an intermediate switch, the header flit is decoded, the next switch on the path is determined and the subsequent flits simply follow on this reserved path in a pipelined fashion. To incorporate multicasting, the addressing mechanism has to be

modified, as incoming flits will be directed to more than one switch at a time.

The header of a multi-destination message must carry the destination nodes addresses. To route a multi-destination message, a switch must be equipped with a method for determining the output ports to which a multicast message must be simultaneously forwarded.

One way to implement multicasting is to simply unicast multiple times, at the cost of significantly increasing the latency. The all-destination encoding is another simple scheme in which all destination addresses are carried by the header. This encoding scheme has two main advantages. First, the same routing hardware can be used for both single- and multi-destination messages. Second, the message header can be processed on the fly as the address flits arrive. However, the drawback of this scheme is that as the number of switch blocks of the NoC increases, the header length increases accordingly and thereby may result in significant overhead.

A form of header encoding that can accomplish multicast to an arbitrary set of destinations and also limits the size of the header is known as bit-string encoding [20]. A multi-destination worm with a bit-string encoded header carries an encoding of the destinations, which consists of an N -bit string, with a '1' bit in the i^{th} position indicating that switch i is a multicast destination. To decode a bit-string encoded header, a switch must possess knowledge of the switches reachable through each of its output ports [19]. The reachability information can be encoded using a similar N bit string for each output port with 1's denoting switches reachable via the respective output port. When a multi-destination worm with a bit-string encoded header arrives at a switch, the switch compares the bit-string in the header with the reachability information associated with each of its output ports.

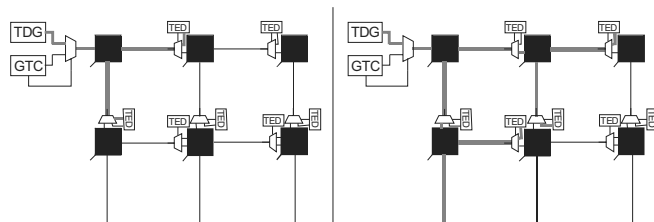


Figure 8: Interleaved multicast MAF test (only a subset of unidirectional links shown)

When a multicast MAF test packet reaches a switch that possesses links to be tested, it is forwarded simultaneously to *all* the untested adjacent links. Hence, all these links are tested in parallel, reducing the test time significantly. Fig. 8 illustrates the succession of two multicast test sequences for a subset of inter-switch links of a mesh-based NoC topology.

6. Experimental results

We implemented the three BIST schemes (point-to-point, unicast, multicast) on an 8-by-8 mesh NoC architecture, similar to the one in Fig. 1a. In each case, we designed and synthesized the hardware blocks used by the corresponding BIST scheme using a standard cell library in a CMOS 130 nm TSMC technology and Synopsys's synthesis tools. The efficiency of each approach was evaluated with respect to the test time and silicon area required.

For the unicast and multicast MAF tests, we used source routing [22] to establish the path of the test packets, due to its simple and area-efficient implementation.

The correct functionality of the BIST mechanism was verified by comparing the individual outputs of the TED blocks with the outputs of the error generator. Several sets of randomly perturbed coupling capacitances were considered and simulations were run with each of these sets. For all simulation runs, the BIST mechanism consistently reported the detection of all the injected errors correctly, confirming the correct functionality of the BIST mechanism.

Table 2: Test time and test area overhead

BIST type	Test time (clock cycles)	Test area overhead (2-input NAND gates)
Point-to-point	50×10^3	166404
Interleaved unicast	124×10^6	89112
Interleaved multicast	42×10^4	90026

Table 2 shows the test time and area overhead for a mesh-based NoC with 64 switches, interconnected with unidirectional pairs of 64-bit wide links. Note that the point-to-point BIST is the fastest one while the interleaved unicast is the slowest (approx. 3 orders of magnitude slower than the point-to-point BIST). The multicast BIST offers a good compromise between the test time and test area overhead (approx. one order of magnitude test time reduction and half the area). As an example, the MAF test time of a NoC chip with a rated clock of 1 GHz, would be 50 ms for the point-to-point method, 124 s for the unicast BIST, and 420 ms for the multicast method. The area numbers in Table 2 take into account the Global Test Controller (GTC) blocks in Figs. 7 and 8.

6.1 Limitations

Crosstalk between global wires routed closely is one of the main causes that affect signal integrity in DSM chips. Designers can employ crosstalk avoidance codes (CAC) in order to mitigate crosstalk effects [17] [18]. CACs are implemented by dedicated coder-decoder circuits that reduce the amount of self- and mutual switching occurring on the data lines. Therefore, the MAF tests are irrelevant when applied directly to links that transport CAC-coded data in normal operation, since the worst-case 2-vector patterns of the MAF tests will never appear as valid functional data.

On the other hand, applying MAF tests at the inputs of CAC coders can cause coverage loss. Consider the case of a simple CAC, i.e., the forbidden pattern code (FPC) [17] for a 4-bit data link $d_3d_2d_1d_0$ coded to a 5-bit codeword $c_4c_3c_2c_1c_0$, referred to as FPC(4,5). According to this code, the MAF 2-pattern test sequence (0000, 1101) is translated by the CAC coder into (00000, 11001). The original MAF pattern tests for a positive glitch on d_1 and stuck-at-0 faults on d_3 , d_2 and d_0 . However, because of the use of FPC, the MAF fault and one of the stuck-at-0 faults are not sensitized. This example illustrates that MAF tests may have to take into account the potential existence of special coding techniques such as crosstalk avoidance codes.

7. Conclusions

We have developed and demonstrated a self-test methodology for at-speed testing of inter-switch links in NoC communication fabrics based on the MAF fault models. The self-test methodology involves the insertion of the test structures, such as test data generators, and error detectors, into the NoC medium. We have also proposed and investigated different options for reducing the test area overhead by reusing the NoC elements and by exploiting the inherent parallelism of the on-chip networks. The parameterized test structures have been synthesized and simulated. We validated our approach by applying it to a 64-switch NoC and simulating the BIST at HDL level. We have also presented potential limitations of the MAF tests.

8. Acknowledgements

The authors wish to acknowledge support from Micronet, PMC-Sierra, Gennum, NSERC and CMC. The authors also thank S. Chemla of L'Ecole Polytechnique Universitaire de Marseille for help with simulations.

9. References

- [1] L. Benini, G. De Micheli, "Networks on Chips: A New SoC Paradigm" *Computer*, Volume: 35 Issue: 1, Jan. 2002, pp.: 70-78.
- [2] P. Magarshack, P.G. Paulin, "System-on-Chip beyond the Nanometer Wall", *Proceedings of DAC*, June 2-6, 2003, Anaheim, USA, pp. 419-424.
- [3] S. Kumar et al, "A Network on Chip Architecture and Design Methodology," *Proceedings of ISVLSI*, Pittsburgh, USA, 2002, pp. 117-124.
- [4] W. J. Dally, B. Towles, "Route Packets, Not Wires: On-Chip Interconnection Networks", *Proceedings of DAC*, Las Vegas, Nevada, USA, June 18-22, 2001, pp. 683-689.
- [5] P. Guerrier, A. Greiner, "A Generic Architecture for on-chip Packet-switched Interconnections", *Proceedings of DATE*, Paris, France, March 27-30, 2000, pp. 250-256.
- [6] P. P. Pande, C. Grecu, A. Ivanov, R. Saleh, "Design of a Switch for Network on Chip Applications", *Proceedings of ISCAS*, Bangkok, May 2003, Volume 5, pp. 217-220.
- [7] C. Grecu, P. P. Pande, A. Ivanov, R. Saleh "Structured Interconnect Architecture: A Solution for the Non-Scalability of Bus-Based SoCs", *Great Lakes Symposium on VLSI*, Boston, USA 26-28 April 2004, pp 192-195.
- [8] L. Benini, D. Bertozzi, "Xpipes: A Network-on-chip Architecture for Gigascale Systems-on-chip", *IEEE Circuits and Systems Magazine*, Volume 4, Issue 2, 2004, pp. 18-31.
- [9] E. Cota, Luigi Caro, Flavio Wagner, Marcelo Lubaszewski, "Power aware NoC Reuse on the Testing of Core-Based Systems", *Proceedings of ITC 2003*, pp. 612-621.
- [10] P. Pande, C. Grecu, A. Ivanov, R. Saleh, G. De Micheli, "Design, Synthesis and Test of Networks on Chip", *IEEE Design and Test of Computers*, Vol. 22, No. 5, 2005, pp. 404-413.
- [11] C. Grecu, P. Pande, B. Wang, A. Ivanov, R. Saleh, "Methodologies and Algorithms for Testing Switch-Based NoC Interconnects", *Proceedings of IEEE DFT 2005*.
- [12] E. J. Marinissen, R. Arendsen, G. Bos, H. Dingemanse, M. Lousberg, C. Wouters, "A Structured and Scalable Mechanism for Test Access to Embedded Reusable Cores", *Proceedings of ITC 1998*, pp. 284-293.
- [13] B. Vermeulen, J. Dielissen, K. Goossens, C. Ciordas, "Bringing Communications Networks on a Chip: Test and Verification Implications", *IEEE Communications Magazine*, Volume 41, Issue 9, sept. 2003, pp. 74-81.
- [14] M. Nahvi, A. Ivanov, "Indirect Test Architecture for SoC Testing", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Volume 23, Issue 7, July 2004, pp. 1128-1142.
- [15] M. Cuvillo, S. Dey, X. Bai, Y. Zhao, "Fault Modeling and Simulation for Crosstalk in System-on-Chip Interconnects", *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, San Jose, CA, Nov. 1999, pp. 297-303
- [16] X. Bai, S. Dey, "High-level Crosstalk Defect Simulation for System-on-Chip Interconnects", *Proceedings of the 19th IEEE VLSI Test Symposium*, Marina del Rey, CA, April 2001, pp. 169-175
- [17] S. Sridhara, N. Shanbhag, "Coding for Reliable On-Chip Busses: Fundamental Limits and Practical Codes", *Proceedings of the IEEE Conference on VLSI Design 2005*, pp. 417-422
- [18] D. Rossi, A. K. Nieuwland, A. Katoch, C. Metra, "Exploiting ECC Redundancy to Minimize Crosstalk Impact", *IEEE Design and Test of Computers*, Jan/Feb 2005, Vol. 22, No. 1, pp. 59-70.
- [19] P. P. Pande, C. Grecu, A. Ivanov, R. Saleh, "Switch-Based Interconnect Architecture for Future Systems on Chip", *Proceedings of SPIE, VLSI Circuits and Systems*, Vol. 5117, pp. 228-237, 2003, Maspalomas, Spain.
- [20] R. Sivaram, C. Stunkel, D. K. Panda, "Implementing Multidestination Worms in Switch-Based Parallel Systems: Architectural Alternatives and their Impact." *Proceedings of the 24th ACM Annual International Symposium on Computer Architecture (ISCA-24)*, June 1997.
- [21] P. Kapur, G. Chandra, J. McVittie, K. C. Saraswat, "Technology and Reliability Constrained Future Copper Interconnects - Part II: Performance Implications", *IEEE Transactions on Electron Devices*, Vol. 49, No. 4, April 2002 pp. 598-604.
- [22] J. Duato, S. Yalamanchili, L. Ni, *Interconnection Networks - An Engineering Approach*, Morgan Kaufmann, 2002.
- [23] C. Grecu, P. Pande, A. Ivanov, R. Saleh, "Timing Analysis of Network on Chip Architectures for MP-SoC Platforms", *Microelectronics Journal*, Elsevier, Vol. 36(9), pp. 833-845.
- [24] Pasquale Cocchini, "Concurrent Flip-Flop and Repeater Insertion for High Performance Integrated Circuits", *Proceedings of ICCAD 2002*, pp. 268-273.