International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 3, Issue 05, March 2016
Available at http://internationaljournalofresearch.org

Preventing Fault Tolerant in Fifo Buffer of Noc Router

V.NAVEEN KUMAR[1]; Y.L. AJAY KUMAR[2] & S.RAVI KUMAR[3]

[1]M.Tech Dept of ECE, PVKK College, Affiliated to JNTUA, AP, India .
[2]Associate Professor, Dept of ECE, PVKK College, Affiliated to JNTUA, AP, India
[3]Assistant Professor, Dept of ECE, PVKK College, Affiliated to JNTUA, AP, India

Abstract – *The on-line transparent test technique for detection of latent hard faults which develop in first input first output buffers of routers during field operation of NoC and also propose fault tolerant solution by introducing implementation of the proposed test algorithm has been integrated into the router-channel interface and on-line test has been performed with synthetic self-similar data traffic. A prototype implementation of the proposed test algorithm has been integrated into the router-channel interface and on-line test has been performed with synthetic self-similar data traffic. The performance of the NoC after addition of the test circuit has been investigated in terms of throughput while the area overhead has been studied by synthesizing the test hardware. In addition, an on-line test technique for the routing logic has been proposed which considers utilizing the header flits of the data traffic movement in transporting the test patterns.*

Keywords: Fault; NOC; Router; Shared queues.

I. INTRODUCTION

Chip integration has reached a stage where a complete system can be placed in a single chip. When we say complete system, we mean all the required ingredients that make up a specialized kind of application on a single silicon substrate. This integration has been made possible because of the rapid developments in the field of VLSI designs. This is primarily used in embedded systems. Thus, in simple terms a SoC can be defined as "an IC, designed by stitching together multiple stand-alone VLSI designs to provide full functionality for an application." A NoC is perceived as a collection of computational, storage and I/O resources on-chip that are connected with each other via a network of routers or switches instead of being connected with point to point wires. These resources communicate with each other using data packets that are routed through the network in the same manner as is done in traditional networks. It is clear from the definition that we need to employ highly sophisticated and researched methodologies from traditional computer networks and implement them on chip. we have to explore the motivating factors that are compelling the researchers and designers to move toward the adoption of NoC architectures for future SoCs.The area of NoC is still in its infancy, which is one of the reasons why there are various names for the same thing; some call it on-chip networks, some networks on silicon, but the majority agrees upon "Networks on Chips" (NoCs). However, we will be using these terminologies interchangeably throughout our tutorial.NOC is Integrating various processors and on chip memories into a single chip .Faults occur in NOC

- Permanent faults
- Transient fault

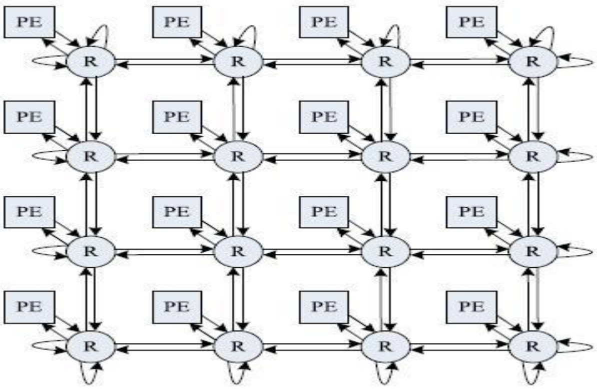

Fig:1. NoC Architecture

II. Related Work

As fault tolerance in NoC design has gained importance among research community, a number of papers have been published covering different aspects of fault tolerance, such as failure mechanisms, fault modeling, diagnosis, and so on. A detailed survey summarizing the research work in these papers has been provided in [3]. Over the years, researchers have proposed a number of Design-For-Testability (DFT) techniques for NoC infrastructure testing (test-ing routers as well as NoC interconnect) [7] and for NoC based core testing [8]. Built-in self test (BIST)-based techniques have also been used for testing routers as well as NoC interconnect, such as [8]. A recent paper on NoC and router testing in [9] provides a summary of the DFT techniques employed for testing NoC interconnects and routers in particular. In addition to novel test architectures, fault tolerant routing algorithms have also been proposed [10].

FIFO buffers in NoC infrastructure are large in number and spread all over the chip. Accordingly, probability of faults is significantly higher for the buffers compared with other components of the router. Both online and offline test techniques have been proposed for test of FIFO buffers in NoC. The proposal in [11] is an offline test technique (suitable for the detection of manufacturing fault in FIFO buffers) that proposes a shared BIST controller for FIFO buffers. Online test techniques for the detection of faults in FIFO buffers of NoC routers have been proposed in [12]. However, the technique considers standard cell-based FIFO buffers, while we consider SRAM-based FIFO designs. Thus, faults considered in this brief are different from those targeted in [12].

III. Proposed Transparent Test Generation

The faults considered in this brief, if applied for SRAMs or DRAMs, can be detected using standard March tests. However, if the same set of faults are considered for SRAM-type FIFOs, March test cannot be used directly due to the address restriction in SRAM-type FIFOs mentioned in and thus we were motivated to choose single-order address MATS++ test (SOA-MATS++) for the detection of faults considered in this brief. The word oriented SOA-MATS++ test is represented as {_ *(wa)*; ↑ *(ra,wb)*; ↓ *(rb,wa)*}; _ *(ra)*} where, *a* is the data background and *b* is the complement of the data background. ↑ and ↓ are increasing and decreasing addressing order of memory, respectively. _ means memory addressing can be increasing or decreasing. Application of SOA-MATS++ test to the FIFO involves writing patterns into the FIFO memory and reading them back. As a result, the memory contents are destroyed. However, online memory test techniques require the restoration of the memory contents after test. Thus, researchers have modified the March tests to transparent March test so that tests can be performed without the requirement of external data background and the memory contents can be restored after test. We have thus transformed the SOA-MATS++ test to transparent SOA MATS++ (TSOA-MATS++) test that can be applied for online test of FIFO buffers. The transparent SOA-MATS++ test generated is represented as $\{\uparrow (rx, w\bar{x}, r\bar{x}, wx, rx)\}$.

The transparent SOA-MATS++ algorithm is intended for test of stuck-at fault, transient fault, and read stuck-at fault, transition fault, and read disturb fault tests developed during field operation of FIFO memories. The fault coverage of the algorithm is shown in Fig. 2. In both the figures, the word size of FIFO memory is assumed to be of 4 bits. As shown in Fig. 2,assume the data word present in *lut* be 1010.The test cycles begin with the invert phase (memory address pointer *j* with 0 value) during which the content of location addressed is read into temp and then backed up in the original. The data written back to SOA-MATS++ test.lut is the complement of content of temp. Thus, at the end of the cycle, the data present in temp and original is 1010, while lut contains 0101.Assume a stuck-at-1 fault at the most significant bit (MSB) position of the word stored in lut.

Thus, instead of storing 0101, it actually stores 1101 and as a result, the stuck-at-fault at the MSB gets excited. During the second iteration of j, when lut is readdressed, the data read into temp is 1101. At this point, the data present in temp and\original are compared (bitwise XOR ed). An all 1’s pattern is expected as result. Any 0 within the pattern would mean a stuck-at fault at that bit position.

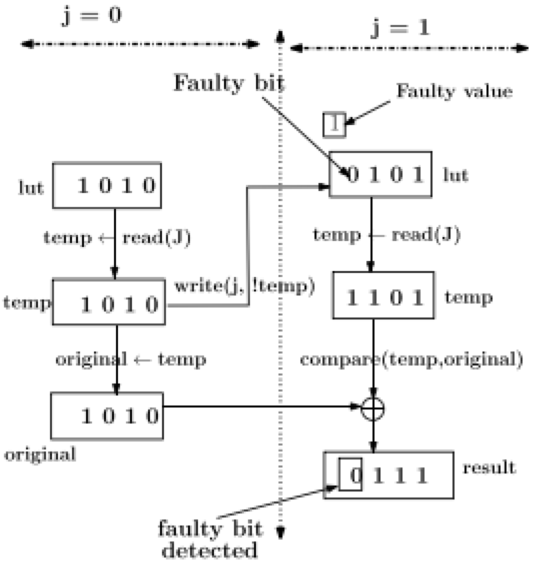

Fig:2. Fault detection during invert phase and restore phase of the transparent SOA-MATS++ test.

IV. Implementation of the Test on FIFO Buffers of NoC Routers

In this section, we present the technique used for implementing the proposed transparent SOA-MATS++ test on a mesh-type NoC. Data packets are divided into flow control units (*flits*) and are transmitted in pipeline manner.

The flit movement in a mesh-type NoC infrastructure considered for this work is assumed to require buffering only at the input channels of routers. Thus, for a data traffic movement from one core to another, the online test is performed only on the input channel FIFO buffers, which lie along the path. The buffers operate in two modes, the *normal* mode and the *test* mode. The FIFO buffers are allowed to be operative in normal mode for sufficient amount of time before initiating their test process. This delay in test initiation provides sufficient time for run-time intermit-tent faults developed in FIFO buffers to transform into permanent faults. The test process of a targeted FIFO buffer is initiated by a counter, which switches the FIFO buffer from normal mode to test mode. The switching of FIFO buffers from normal mode to test mode occurs after a certain period of time without caring about the present state of the FIFO buffer. It may be argued that at the instant of switching, the buffer may not be full, and as a result not all locations would be tested during the test cycle. . However, test initiation after the buffer gets full would cause the following problems. First, wait for the buffer to get full would unnecessarily delay the test initiation process and would allow faults to get accumulated. Second, test of the entire buffer would prolong the test time and would negatively affect the normal mode of operation.

V. RESULTS

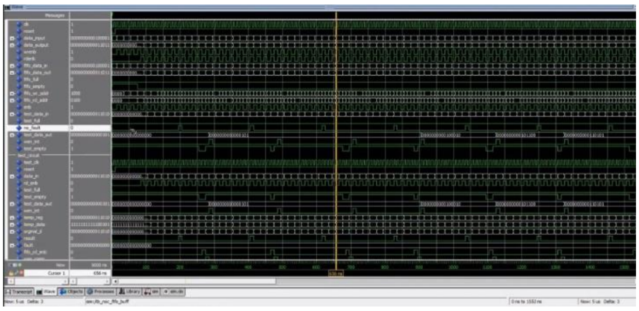

Fig 3; Simulation Results

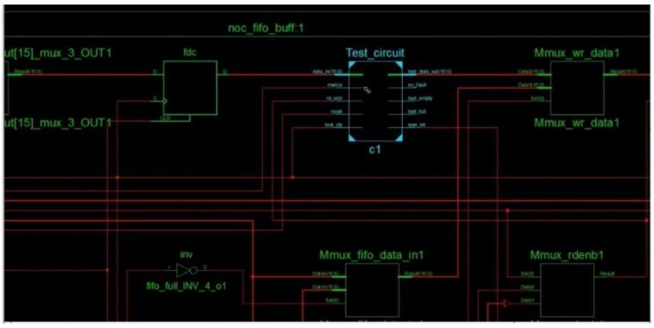

Fig 4: Test diagram for FIFO Buffers of Noc Routers

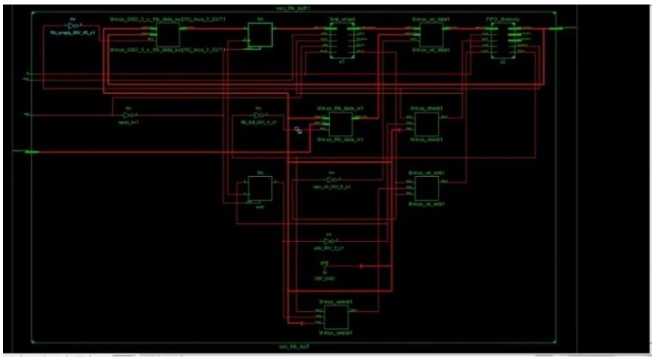

Fig 5: RTL diagram for FIFO Buffers of Noc Routers

IV. CONCLUSION

From this paper we can proposed transparent SOA-MATS++ test generation algorithm that can detect run-time, permanent faults developed in SRAM-based FIFO memories. The proposed transparent test is utilized to perform online and periodic test of FIFO memory present within the routers of the NoC. Periodic testing of buffers prevents accumulation of faults and also allows test of each location of the buffer. Simulation results show that periodic testing of FIFO buffers do not have much effect on the overall throughput of the NoC except when buffers are tested too frequently. We have also proposed an online test technique for the routing logic that is performed simultaneously with the test of buffers and involves utilization of the unused fields of the header flits of the incoming data packets for test pattern encoding. As future work, we would like to modify the proposed FIFO testing technique that will allow incoming data packets to the router under test without interrupting the test.

REFERENCES

[1] W. J. Dally and B. Towles, “Route packets, not wires: On-chip inter-connection networks,” in *Proc. 38th Annu. Design Autom. Conf.*, 2001, pp. 684–689.

[2] A. Bondavalli, S. Chiaradonna, F. Di Giandomenico, and F. Grandoni, “Threshold-based mechanisms to discriminate transient from intermittent faults,” *IEEE Trans. Comput.*, vol. 49, no. 3, pp. 230–245, Mar. 2000.

[3] M. Radetzki, C. Feng, X. Zhao, and A. Jantsch, “Methods for fault tolerance in networks-on-chip,” *ACM Comput. Surv.*, vol. 46, no. 1, pp. 1–38, Jul. 2013, Art. ID 8.

[4] S. Ghosh and K. Roy, “Parameter variation tolerance and error resiliency: New design paradigm for the nanoscale era,” *Proc. IEEE*, vol. 98, no. 10, pp. 1718–1751, Oct. 2010.

[5] S. Borri, M. Hage-Hassan, L. Dilillo, P. Girard, S. Pravossoudovitch, and A. Virazel, “Analysis of dynamic faults in embedded-SRAMs: Implications for memory test,” *J. Electron. Test.*, vol. 21, no. 2, pp. 169–179, Apr. 2005.

[6] M. Bushnell and V. Agrawal, *Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits* (Frontiers in Electronic Testing). New York, NY, USA: Springer-Verlag, 2000.

[7] D. Xiang and Y. Zhang, “Cost-effective power-aware core testing in NoCs based on a new unicast-based multicast scheme,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 30, no. 1, pp. 135–147, Jan. 2011.

[8] K. Petersen and J. Oberg, “Toward a scalable test methodology for 2D-mesh network-on-chips,” in *Proc. Design, Autom., Test Eur. Conf. Exhibit.*, Apr. 2007, pp. 1–6.

[9] D. Xiang, “A cost-effective scheme for network-on-chip router and interconnect testing,” in *Proc. 22nd Asian Test Symp. (ATS)*, Nov. 2013, pp. 207–212.

[10] M. Ebrahimi, M. Daneshtalab, J. Plosila, and H. Tenhunen, “Minimal-path fault-tolerant approach using connection-retaining struc-ture in networks-on-chip,” in *Proc. 7th IEEE/ACM Int. Symp. Netw. Chip (NoCS)*, Apr. 2013, pp. 1–8.

[11] C. Grecu, P. Pande, B. Wang, A. Ivanov, and R. Saleh, “Methodologies and algorithms for testing switch-based NoC interconnects,” in *Proc. 20th IEEE Int. Symp. Defect Fault Tolerance VLSI Syst.*, Oct. 2005, pp. 238–246.

[12] M. R. Kakoee, V. Bertacco, and L. Benini, “A distributed and topology-agnostic approach for on-line NoC testing,” in *Proc. 5th ACM/IEEE Int. Symp. Netw. Chip*, May 2011, pp. 113–120.

[13] S. Barbagallo *et al.*, “A parametric design of a built-in self-test FIFO embedded memory,” in *Proc. IEEE Int. Symp. Defect Fault Tolerance VLSI Syst.*, Nov. 1996, pp. 221–229.

[14] A. J. van de Goor and Y. Zorian, “Functional tests for arbitration SRAM-type FIFOs,” in *Proc. 1st Asian Test Symp. (ATS)*, Nov. 1992, pp. 96–101.

[15] M. Nicolaidis, “Theory of transparent BIST for RAMs,” *IEEE Trans. Comput.*, vol. 45, no. 10, pp. 1141–1156, Oct. 1996. S. Kundu, J. Soumya, and S. Chattopadhyay, “Design and eval-