

Online Fault Detection Method within SRAM Based FIFO Buffer in NOC Router

Sandhiya.J.A¹, Vijay Murugan.S²

PG Student, Dept. of ECE, Adhiyamaan College of Engineering, Hosur, Tamilnadu, India¹

Associate Professor, Dept. of ECE, Adhiyamaan College of Engineering, Hosur, Tamilnadu, India²

ABSTRACT: This project present an online transparent test technique to identify the latent hard faults which enlarge in first in first out (FIFO) buffer of router for the period of field operation of network on chip (NOC), proposed transparent test is utilized to perform on-line and periodic test of FIFO_memory present contained by the router of the NOC. This technique is involved periodically to prevent build-up of fault which developed in buffer. NOC has emerged as a promising role for on-line communication. An on-line test technique for the routing logic has been proposed which consider utilizing the header flits of the data traffic faction in transporting the test pattern. The prototype execution of the proposed test algorithm has been integrated interested in the router channel boundary and on-line tests have been performed with imitation self-similar data traffic. The performance of the NOC after addition of the test circuit have been investigated within terms of throughput while the area overhead has been studied through synthesize the test hardware. This paper study the logic size, area and power consumption using Xilinx 14.2 and modelsim is used to demonstrate proposed result.

KEYWORDS: Fault, FIFO Buffer, NOC, Router, Permanent Test, Transparent Test.

I. INTRODUCTION

By means of increasing the large number of cores in a single chip, on-chip communication efficiency has become one of the important key factors for calculating overall system presentation and cost. System on a chip (SoC) is the design technique at present used by VLSI designers, conventional on extensive IP core reuse. Dedicated wires are efficient only for systems by means of a limited number of cores, as the number of wires in the system upgrades significantly as the number of cores grows. Therefore, dedicated wires have reduced reusability and flexibility. The important amount of area in NoC is occupied by router, which is mainly employed by FIFO buffers and routing logic. therefore the probabilities of run-time faults taking place in buffers and logic are significantly higher compared with the other components of NoC. as a result, NoC communications of test procedure necessity begin using test of buffers and routing logic of routers. In addition, the test should be implemented periodically to ensure that no fault should gets accumulated.[1]

Networks-on-Chip (NoC) is a promising solution for on-chip interconnection in MCSocS due to its scalability, reusability, flexibility, and parallelism. Transient as well as permanent faults are the two different types of faults that can be capable of occur in on-chip networks. Transient faults be temporary and random. They are frequently difficult to be detected and corrected. Permanent faults be caused by physical damages such as developed faults and device wear-out. These faults must be recovered or else tolerated in a way that the network continues performance. Routing techniques afford some steps of fault tolerance in NoCs. They can be categorized into deterministic as well as adaptive approaches. A deterministic routing algorithm uses a permanent path designed for each pair of nodes resulting in increased packet latency particular in packed full networks. In adaptive routing algorithms, packets are not limited to a single path when crossing over from a source to a destination router. Therefore, they can be able to decrease the probability of routing packets in the course of packed areas and thus recover the presentation. March C algorithm, Symmetric MATS+ algorithm have be proposed target in-field permanent faults was developed in SRAM based FIFO memories and it have been utilized to perform online along with periodic test of FIFO memory present restricted through the routers of the NoC. In addition,

we have also included an proposed an online test technique for the routing logic that is performed simultaneously with the test of buffers.

In minimal adaptive routing algorithms, the shortest paths be used for transmitting messages involving routers. In NoCs, faults may arise in cores (such as processing elements and memory modules), links or routers. at the same time as a core is faulty, the connected router by means of links can carry on functioning. When a link is faulty, the approaches related to BFT-NoC preserve the connectivity by a energetic sharing of surviving channels. Thereby, cores and routers perform functioning normally. [2]The majority severe case causes through a faulty router. In this folder, not only the connected core cannot send or receive packets, but also the packets from the other cores cannot be transmitted during this router. The most common solution is to reroute packets around the faulty router or else faulty region. Considering a NoC as a network of routers and NIs, a hardware fault protect occur in different places including: routers, links, and NIs. Faults in the links be able to be detected using a small number of test patterns. Faults in router be able toward occur in two main parts: data-path and control-path. Data-path contains flip-flops in FIFOs, router wires, and output MUXes; control-path include routing logics, FIFO's control part, and arbiters. once a router is in test mode, it must be tested by its neighbours. Each router contains three main test blocks: Test Pattern Generator (TPG), Test Response Analyzer (TRA), as fine as Fault Diagnosis Module (FDM). TPG is in charge for generating test patterns and sending them to the neighbour router which is under test (RUT). TPG of the local port which is inside the router under test's NI sends test packets toward the neighbours. Test response analyzer receives test patterns on or after the neighbour that is under test and analyzes them to detect any fault in the consequent router and links.

An online transparent test technique for first-input first-output (FIFO) buffers and routing logic present inside the routers of the NoC infrastructure. Our contributions are in this way. A transparent SOA-MATS++ test.

II. MODIFIED TRANSPARENT SOA MATS++ ALGORITHM

The faults considered in this brief, if applied for SRAMs otherwise DRAMs, can be detected by means of standard March tests. However, if the same set of the faults are considered for SRAM-type FIFOs, March test cannot be used directly due to the address limit in SRAM-type FIFOs and so to choose single-order address MATS++ test (SOA-MATS++) for the detection of faults considered in this brief. The phrase oriented SOA-MATS++ test is represented as $\{\downarrow(wa); \uparrow(ra,wb); \downarrow(rb,wa)\}; \downarrow(ra)\}$ where, a known as data background and b known as complement of the data background. \uparrow and \downarrow be increasing and decreasing addressing order of memory, respectively. \uparrow it means memory addressing can be increasing or decreasing. Application of SOA-MATS++ test toward the FIFO involves writing patterns into the FIFO memory as well as reading them back. As a result, the memory contents are destroyed. However, online memory test techniques is require the restoration of the memory contents after test. Thus, researchers have been modified the March tests to transparent March test so as to tests be capable of performed without the requirement of external data background and the memory contents can be restored after the test. So that transformed the SOA-MATS++ test towards transparent SOA-MATS++ (TSOA-MATS++) test so as to be applied for online test of FIFO buffers. Here the transparent SOA-MATS++ test generated is represented as $\{\uparrow(rx,wx^-,rx^-,wx,rx)\}$.

Algorithm 1 Transparent SOA-MATS++ Test Algorithm

```

Require: N = number of rows of the FIFO memory
1:  $i \leftarrow 0$ ; /* memory address pointer */
2: while ( $i \leq N - 1$ ) do
3:    $j \leftarrow 0$ ; /* test cycle counter */
4:   while ( $j \leq 2$ ) do
5:      $temp \leftarrow read(i)$ ;
6:     if ( $j = 0$ ) then
7:        $original \leftarrow temp$ ;
8:        $write(i, !temp)$ ;
9:     else
10:      if ( $j = 1$ ) then
11:         $result \leftarrow compare(temp, original)$ ;
12:         $write(i, !temp)$ ;
13:      end if
14:    else
15:       $result \leftarrow compare(temp, original)$ ;
16:    end if
17:     $j \leftarrow j + 1$ ;
18:  end while
19:   $i \leftarrow i + 1$ ;
20: end while

```

The operations is performed during the test represent three phases of the test, namely, invert phase, restore phase, and read phase. The first two operations form a read write pair (rx , wx) representing the invert phase wherever the initial content (content before start of test) of the FIFO buffer location under test (lut) been read and its complement is written back to the same location. The invert phase is follow by restore phase involving the operations (rx , wx), where the content of lut are read and reinverted. At the point of test the lut have been flipped twice to get back the original content.

III. FAULT COVERAGE OF THE PROPOSED ALGORITHM

The on-line transparent test generation of SOA-MATS++ algorithm is intended for test at fault such as transition fault, stuck-at fault, and read disturb fault tests developed during field operation of on-line transparent FIFO memories. The fault coverage of algorithm is shown in Fig. 1. In both the figures, the word size of FIFO memory is assumed to be of 16 bits. The figure is represents based upon the Algorithm 1.

As shown in Fig. 1, assume the data word present in LUT be 1010. The test cycles is begin with the invert phase (memory address pointer j with 0th value) during the content of data is read and written backed to location of original data. The data written back to LUT is the complement of content of $temp$. Thus, at the end of the cycle, the data present in $temp$ plus $original$ is 1010, while LUT contains 0101. Assume stuck-at-1 fault on the most significant bit (MSB) position of the word stored in LUT. Therefore, instead of storing 0101, it in fact stores 1101 and as a result, the stuck-at-fault at the MSB gets excited.

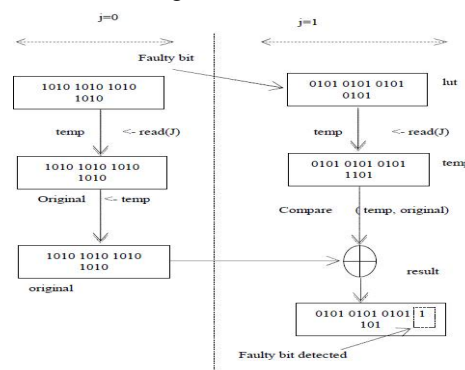


Fig.1 Fault detection during invert phase and restore phase of the transparent SOA-MATS++ test.

The FIFO buffer present in every input channel of a NoC router consists of a SRAM-based FIFO memory of certain depth. During normal operation, data flits arrive through a data-in line of the buffer and are subsequently stored in different locations of the FIFO memory. On request by the neighbouring router, the data flits stored are passed on to the output port through the data out line. Fig. 2(a) shows the FIFO memory with data_in and data_out line. To perform the transparent SOA-MATS++ test on the FIFO buffer, we have added a test circuit, few multiplexers and logic gates to the existing hardware, as shown in Fig.2(a). The read and write operations on the FIFO buffer are controlled by the read enable and write enable lines, respectively. Here the multiplexer's mu6 and mu7 select the read as well as write enable during the normal and test process. During normal operation while the test_ctrl is asserted low, the internal write as well as read enable lines, wen_int as well as ren_int, synchronized with the router clock, provide the write and the read enable, respectively. However, during test process, the write enable and read enable are synchronized with the test clock. since mentioned earlier, the read along with write operations during test are performed at alternate edges of a test clock. The read operations be synchronized with the positive edges, while the write_clk is obtained by inverting the test clock. In test mode (test_ctrl high), the test read as well as write addresses are generated by the test address generators implemented using gray code counters similar to the normal address generation. Muxes m4 in addition to m5 are used to select between normal addresses and test addresses.

A. MARCH-C ALGORITHM

March algorithm is the most commonly used algorithm in memory test. Finite state machine is used to perform read and write operation one via one until all addresses are covered, ensure to each test vector between any two bytes occur at least once "00", "01", "10" and "11". For the sake of checking the influence of read/write order, address increasing as well as decreasing operations are respectively used. March C algorithm is a byte oriented algorithm, a series of operations be carried out at a storage element then move to the next one. The works include the inside test of each unit and also faults among them. Through continuous read and write operations, the system could detect almost all the memory faults

IV. TEST PROCESS ON FIFO BUFFER OF NOC ROUTER

The buffers operate in two different modes, the *normal* mode and the *test* mode. The normal mode and test mode of operation of a FIFO buffer are synchronize with two different clocks. The clock used for test purpose (referred as *test_clk* in this brief) is a faster clock compared with the clock required for normal mode (*routerclock*).

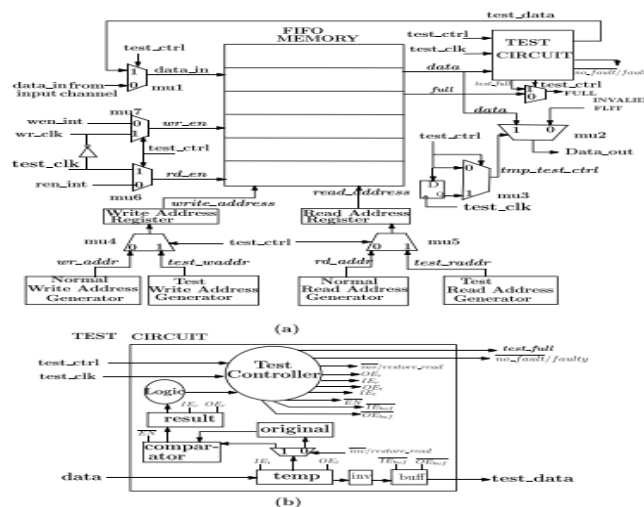


Fig.2 (a) Hardware implementation of the test process for the FIFO buffers. (b) Implementation of test circuit. The FIFO buffers are allowed to be operative in normal mode for sufficient amount of time before initiating their test process. This delay in the test initiation provides enough time for run-time intermittent 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.

A test burst involves series of test read and write cycles. It requires three read and two write cycles, or in other words three cycles of the faster test clock to perform a test on a single location of a FIFO buffer. Thus, repeating the test bursts for number times on a FIFO buffer would cover the test of each location as the number of locations in a FIFO buffer is few. Moreover, periodic testing prevents BUILD UP of fault in the buffer

V. SIMULATION RESULTS

This paper presents a testing based on online BIST of SRAMs. The code is written in verilog language Xilinx ISE 14.2. Whenever test done the respective memory fault signal is indicated. If register output value is all 1's when test done is completed, i.e. fault free memory, otherwise the memory is fault memory. From results we concluded as based on register output memory is faulty or not.

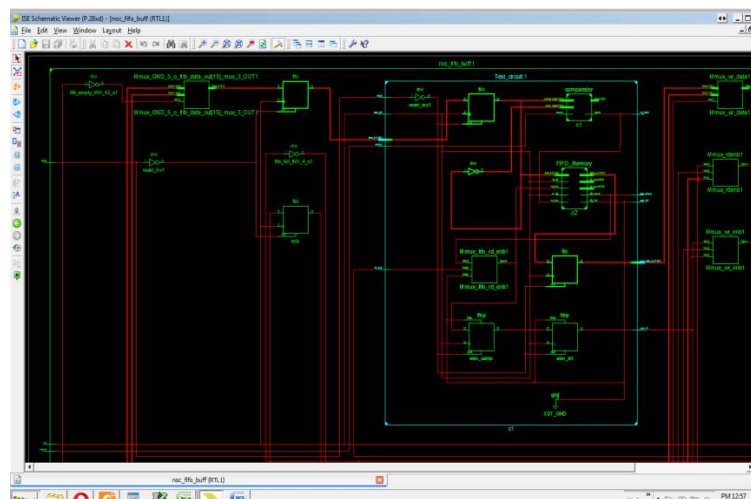


Fig.3 overall architectural view

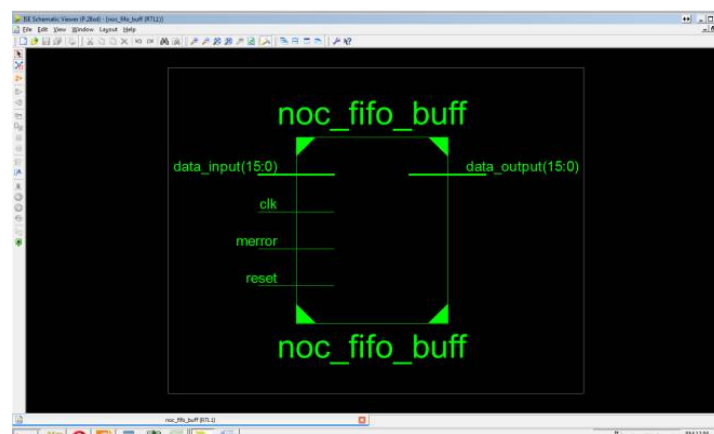


Fig.4 RTL View

Organized by

Department of ECE, Adhiyamaan College of Engineering, Hosur, Tamilnadu, India.

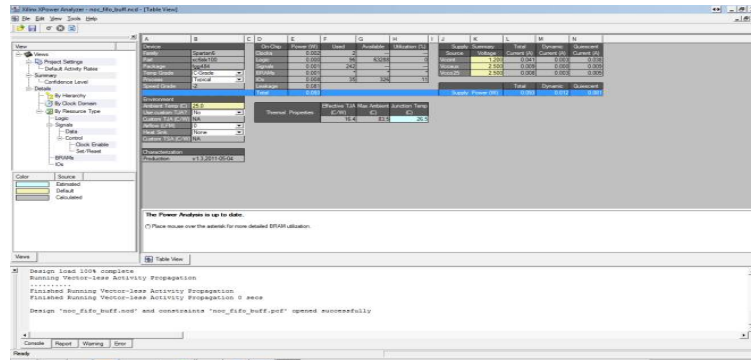


Fig.5 power report

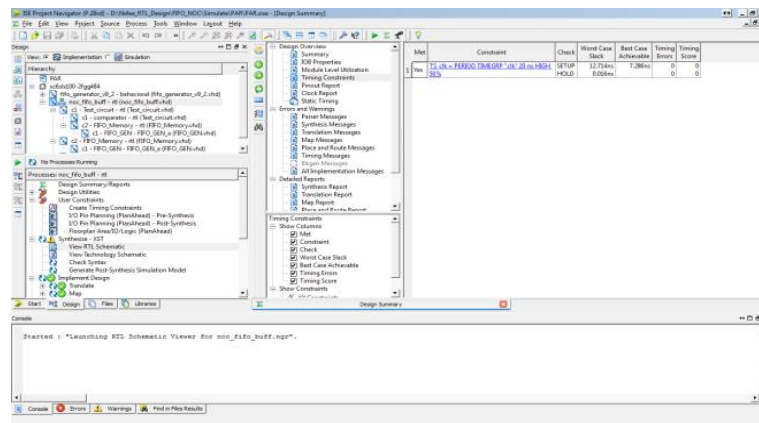


Fig.6 Timing report

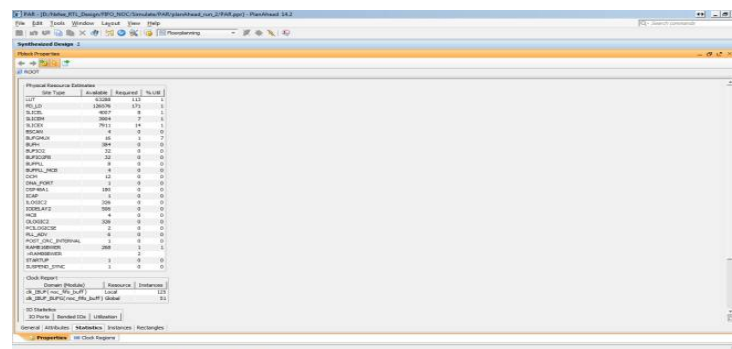


Fig.7 Area report

VI. CONCLUSION

In this paper we have given the explain about the proposed method, in this method a transparent SOA MAT++ test algorithm is used to detect the run time permanent fault which was developed in SRAM based FIFO memories. Here the transparent test is utilized to perform online periodic test in FIFO memories which were present in NOC router. Simulation result shows that periodic test of FIFO buffer is where not having much effect on the overall throughput of NoC apart from when the buffer is tested too recurrently. More over we have also proposed an online test for routing logic and it is performed simultaneously with the test of buffer and using of unused portion of header flits of the incoming data packets for test pattern. In proposed we also included a

FIFO testing technique that will allow incoming data packets to the router under test without interrupting the test.

REFERENCES

- [1] Mohammad Ayoub Khan, Abdul Quaiyum Ansari, n-Bit Multiple read and write FIFO memory model for Network on chip, 978-1-4673-0126-8/11/\$26.00c 2011 IEEE.
- [2] Bibhas Ghoshal, Kanchan Manna, Santanu Chattopadhyay, and Indranil Sengupta, In-Field Test for Permanent Faults in FIFO Buffers of NoC Routers, 1063-8210 © 2015 IEEE.
- [3] L. Benini, G. De Micheli, Networks on chips: a new SoC paradigm, IEEE Computer magazine, 35 (1) (2002) 70–78.
- [4] H. Van Gageldonk, K. Van Berkel, A. Peeters, D. Baumann, D. Gloor, G. Stegmann, An asynchronous low-power 80C51 microcontroller, in: Proceedings of ASYNC'98, 1998, pp. 96–107.
- [5] 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.
- [6] 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.
- [7] M. Ebrahimi, M. Daneshmand, J. Plosila, and H. Tenhunen, "Minimal-path fault-tolerant approach using connection-retaining structure in networks-on-chip," in Proc. 7th IEEE/ACM Int. Symp. Netw. Chip (NoCS), Apr. 2013, pp. 1–8.
- [8] 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.
- [9] M. R. Kakoei, 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.
- [10] 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.
- [11] 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.
- [12] M. Nicolaidis, "Theory of transparent BIST for RAMs," IEEE Trans. Comput., vol. 45, no. 10, pp. 1141–1156, Oct. 1996.