International Journal of Electrical and Electronics Engineers
Vol. No.8 Issue 01, January-June 2016
www.arresearchpublication.com
IJEEE
ISSN 2321 - 2055

FAULT TOLERANT FIFO BUFFER OF NOC ROUTER

S.Divya[1], R.Logaprabhavathy[2]

[1, 2,] M.E,Department of Electronics and Communication Engineering,

Ultra college of Engineering and Technology for women, Madurai (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 shared buffer in router. It provides alternative way in case of detection of faults otherwise used to improve efficiency. The technique involves repeating tests periodically to prevent accumulation of faults. NoC approach has emerged as a promising solution for on-chip communications. This proposes an 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. The technique involves repeating tests periodically to prevent accumulation of faults. Also design router architecture with shared queues (RoShaQ). 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. xilinx 12.1 tool has been used to demonstrate existing and proposed results.

Keyword- Fault, NOC, Router, Shared queues.

I. INTRODUCTION

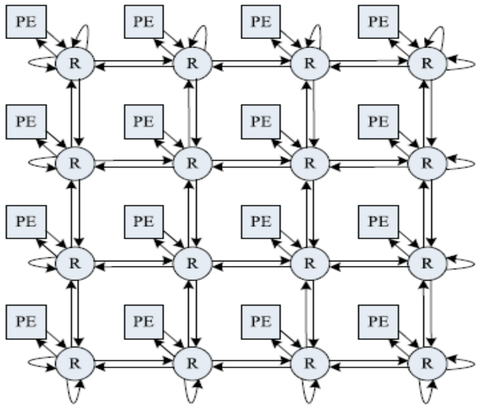
Chip integration has reached a stage where a complete system can be placed in a single chip. When we say complete system, we mean all the required ingredients that make up a specialized kind of application on a single silicon substrate. This integration has been made possible because of the rapid developments 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



PE-processing element

R-router

Fig:1. NoC Architecture

II.EXISTING SYSTEM

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 {↑ *(rx,w¯ x, r ¯ 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.

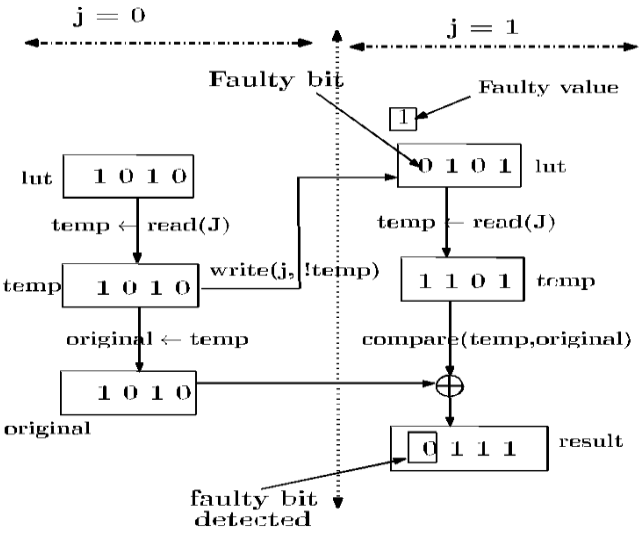


Fig:2. Fault detection during invert phase and restore phase of the transparent SOA-MATS++ test.

where the XOR of 1010 and 1101 yields a 0 at the MSB position of the result indicating a stuck-at-fault at the MSB position. However, for cases where the initial data for a bit position is differentfrom the faulty bit value, the stuck-at-fault cannot be detected for the bit position after the restore phase of the test. It thus requires one more test cycle to excite such faults.

III.PROPOSED SYSTEM

ROSHAQ ARCHITECTURE:

We provide alternate bypassing capacity in the case of detection of faults we move on one more step by allowing input queues to bypass the shared queues as shown in Fig.3. With this design, a packet from an input queue simultaneously arbitrates for both shared queues and an output port; if it wins the output port, it would be forwarded to the downstream router at the next cycle. Otherwise, that means having congestion at the corresponding output port, it can be buffered to the shared queues. Intuitively, at low load, the network would have low latency because packets seem to frequently bypass shared queues. While at heavy load, shared queues are used to temporarily store packets hence reducing their stall times at input ports that would improve the network throughput.

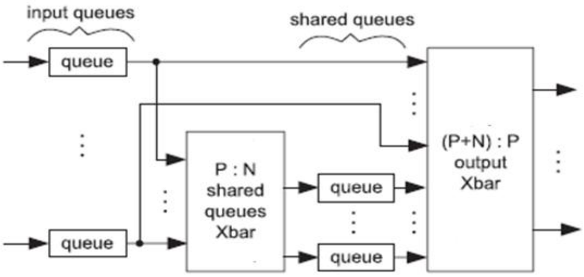


Fig:3. Allows input packets to bypass shared queues. P: the number of router Ports V: the number of VC queues per input port in a VC router. N: the number of shared queues.

RoShaQ, a router architecture with shared queues based on the idea of Fig.3. When an input port receives a packet, it calculates its output port for the next router (look ahead routing), at the same time it arbitrates for both its decided output port and shared queues. If it receives a grant from the output port allocators (OPAs), it will advance to its output port in the next cycle. Otherwise, if it receives a grant to a shared queue, it will be written to that shared queue at the next cycle. In case that it receives both grants, it will prioritize to advance to the output port. Shared-queues allocator (SQA) receives requests from all input queues and grants the permission to their packets for accessing non full shared queues. This shared queue writing policy guarantees deadlock-free for the network The OPA receives requests from both input queues and shared queues. Both SQA and OPA grant these requests in round-robin manner to guarantee fairness and also to avoid starvation and livelock. Input queue, output port, and sharedqueue states maintain the status (idle, wait, or busy) of all queues and output ports, and incorporate with SQA and OPA

IV. RESULT COMPARISONS

The proposed has been simulated and the synthesis report can be obtained by using Xilinx ISE 12.1i.

s.no	Parameter	Existing	Proposed
1	Slice	41	26
2	Lut	90	54

Fig.4. The various parameters used for computing existing and proposed systems with Spartan-3 processor

V. RTL SCHEMATIC

After performing the synthesize process, the RTL schematic has been created automatically based on the functionality. The routing between the different cells can be viewed clearly by this schematic.

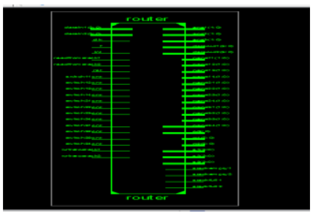


Fig:5. RTL Schematic

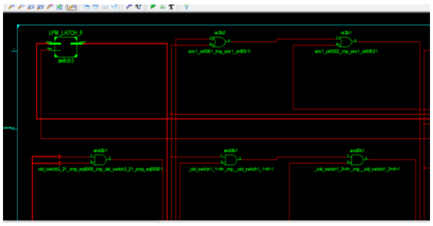


Fig:6. Technology Schematic

VI. 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. A novel router architecture that allowed sharing multiple buffer queues for improving network throughput. Input packets also can bypass the shared queues to achieve low latency in the case that the network load was low

REFERENCES

[1] Bibhas Ghoshal, KanchanManna, Santanu Chattopadhyay, and Indranil Sengupta**"In-Field Test or Permanent Faults in FIFO Buffers of NoC Routers"** IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS

[2] Mohammad Reza Kakoee,, Valeria Bertacco, **"At-Speed Distributed Functional Testing to Detect Logic and Delay Faults in NoCs"** Member, IEEE ,Senior Member, IEEE, and Luca Benini, Fellow, IEEE

[3] MARTIN RADETZKI, University of Stuttgart CHAOCHAO FENG, National University of Defense Technology XUEQIAN ZHAO and AXEL JANTSCH, Royal Institute of Technology" **Methods for Fault Tolerance in Networks-on-Chip"**

[4] Dong Xiang, *and Ye Zhang* **"Cost-Effective Power-Aware Core Testing in NoCsBased on a New Unicast-Based Multicast Scheme"** *Senior Member, IEEE,*

[5] Swaroop Ghosh and Kaushik Roy, **Parameter Variation Tolerance and Error Resiliency: New Design Paradigm for the Nanoscale Era"** Fellow IEEE"

[6] SIMONE BORRI AND MAGALI HAGE-HASSAN**"Analysis of Dynamic Faultin Embedded-SRAMs: Implications for Memory Test"**

[7] William J. Dally and Brian Towles Computer Systems Laboratory Stanford University" **Route Packets, Not Wires: On-Chip Interconnection Networks "**

[8] Bondavalli,, Silvano Chiaradonna,Felicita Di Giandomenico, and Fabrizio Grandoni **"Threshold-Based Mechanisms to DiscriminateTransient from Intermittent Faults"** Member, IEEE

[9] S. Barbagallo, M. Lobetti Bodoni, D. Medina DSRC Italtel Settimo Milanese, Italy G. De Blasio, M. Ferloni, F.Fummi, D. Sciuto Dipartimento di Elettronica e Informmione Politecnico di Milano P.zza L. Da Vinci 32, 20133 Milano – Italy**" A Parametric Design of a Built-in Self-Test FIFO1 Embedded Memory"**

[10] Ad J. van de Goor Delft Univ. of Technology Dept. of Electrical Eng.P.O. Box 5031 2600 GA Delft The Netherlands , Yervant Zorian AT&T Bell Laboratories Engineering Res. Center P.O. Box 900 Princeton, NJ 08542-0900 USA**" Functional Tests for Arbitration SRAM-type FIFOs**

[11] M. Hosseinabady, M. Kakoee, J. Mathew, and D. Pradhan, **"Low latency and energy efficient scalable architecture for massive NoCs using generalized de Bruijn graph,"** *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 19, no. 8, pp. 1469–1480, Aug. 2011.

[12] S. Jovanovic, C. Tanougast, S. Weber, and C. Bobda**, "A new deadlock-free fault-tolerant routing algorithm for NoC interconnections,"** in *Proc. Int. Conf. Field Program. Logic Appl.*, Aug.–Sep. 2009, pp. 326–331.

[13] N. Karimi, A. Alaghi, M. Sedghi, and Z. Navabi, **"Online networkon- chip switch fault detection and diagnosis using functional switch faults,"** *J. Universal Comput. Sci.*, vol. 14, no. 22, pp. 3716–3736, 2008.

[14] C. Grecu, A. Ivanov, R. Saleh, E. Sogomonyan, and P. Pande, "On-line fault detection and location for NoC interconnects," in *Proc. 12th IEEE Int. On-Line Test. Symp.*, Jul. 2006, pp. 145–150.