ISSN 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering
Vol. 1, Issue 2, August 2012

NoC Self-Test

E.Prasad[1], B.Chandra Kala[2], A.Subramanyam[3]

Assistant Professor, Department of ECE, S V College of Engineering, Tirupati, A.P, India[1,2,3]

ABSTRACT: Nowadays interconnect switch network has been used to connect cores in SoC to scale up the number of cores and to reduce the latency. These network communication platforms are called as Network on Chip (NoC) which supports scalability and reusability. NoC switch includes a complex logic structures such as memories, registers and interfaces which makes the testing of a switch more challenging and complex. Often the test time, test cost and fault coverage of NoC switch structures are very less and without testing the switch one cannot test the embedded cores. In this paper a NoC self-test architecture is proposed to test NoC switch by employing BIST based testing methodology.

KEYWORDS: Built in Self-Test (BIST), Network On chip (NoC), System on Chip (SoC), Output Response Analyzer (ORA), First in First OUT (FIFO).

I. INTRODUCTION

Interconnect delays in SoC cores are very large, conventionally Bus-based and Point-to-point communication architectures are used [9] [10], but these methods do not support system requirements like Bandwidth, scalability and reusability. Alternative architecture to conventional architectures is integrated switch network which supports reusability and scalability. The 2-D mesh NoC architecture is as shown in figure 1. The cores are connected to the switches to form a network. Data is organized in packets and transferred through interconnection links. Different network topologies, switching strategies and routing algorithms can be used to meet requirements of performance, hardware overhead, and power consumption. Based on arbitration and routing algorithm the respective data injected is transferred through the respective direction. Testing of NoC Switch is quite challenging and complex in nature for designing the test mechanism [1] [2]. Performance should be compromised for complex NoC structure. The test results of NoC switch structure often results with low fault coverage in test mode. In this project new self-test mechanism is proposed which addresses the issue of testing time, test cost and fault coverage with fault grading based on functional patterns. The STUMPS BIST is most widely BIST architecture compared to other BIST architectures [10], ensuring the lower area overhead based self-test environment. Here in this paper, BIST based self-test mechanism is introduced for testing the NoC switches. The implementation, working procedure and analysis of NoC self-test with BIST is discussed in further sections. The remaining paper is organised as follows, in section II NoC environment is discussed, section III explains the detailed.

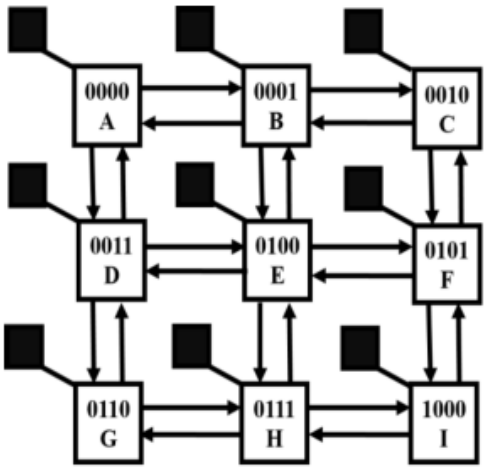

Figure 1: 2-D Mesh NoC

testing process of STUMPS BIST. The proposed NoC self-test structure is explained in Section IV. The experimental setup and analysis of the NoC self-test BIST architecture are given in section V, Section VI concludes the paper with its future scope.

ISSN 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering
Vol. 1, Issue 2, August 2012

II. NOC ENVIRONMENT

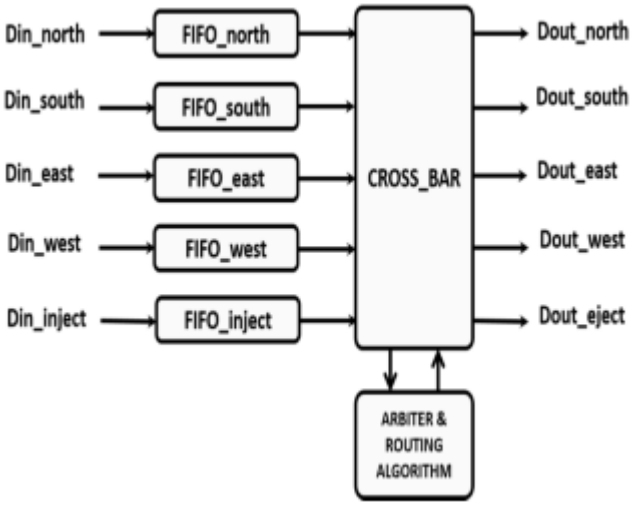

Figure 2.: Cross Bar Switch

Generally the discussed NoC from section I. consists of processing elements, crossbar switch, buffers, and arbiter. The performance of NoC depends on switching strategy, Network topology and Routing algorithms. Network topology gives the physical organization of processing elements and nodes. General topology used is direct 3x3 mesh network topology which has direct connection to neighbouring nodes. Switching techniques defines the granularity of data transfer from source to destination with packet switching technique. The routing algorithms provide path to route data from source to destination with static distributed routing algorithm. Arbitration schemes are used in NoC to avoid congestion in routing path with static priority arbiter. The simple NoC switch block diagram is as shown in figure 2 each switch consists of a crossbar and internal memory blocks. The processing elements generate the message and it is transferred in form of packets which is as shown in figure 2.1, these packets are divided into FLIT‘s (flow control unit) [12][13]. FLIT‘s consists of head flit, body flit and tail flit. Head flit ensures the arrival of data, body flit consists of destination address and tail flit gives the acknowledgement, once the data reaches destination. FLIT is classified into physical units known PHIT. The packet structure consists of four FLITs and each PHIT size is of 8 bit. FIFO buffer size is same as the FLIT size. The NoC Environment Block diagram is shown in figure 2.2. The working procedure of NoC environment is as discussed in below steps.

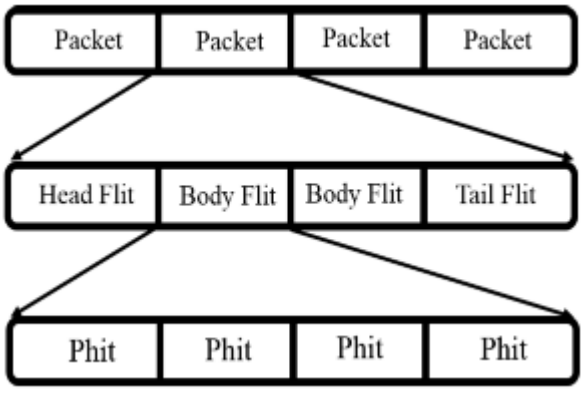

Figure 2.1: Packet Structure

i. Processing elements generates data that to be routed, the data is transferred in form of packets which is given as input to the FIFO buffers.
ii. Based on the head flit the address is decoded from the FLIT. By consulting the routing table, next switch where data to be sent is identified. For example the current source is switch A and the destination is switch I, by consulting the routing table the next switch to be routed will be west (Switch B).
iii. Based on the decoded address, request to o the output link is posted for arbiter. This posted request is used to generate the control signals for crossbar which route input data the respective output link.
iv. Once the destination address matches with the current switch address, data is ejected via eject port. Otherwise the process from step 1 for the current switch.

ISSN 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering
Vol. 1, Issue 2, August 2012

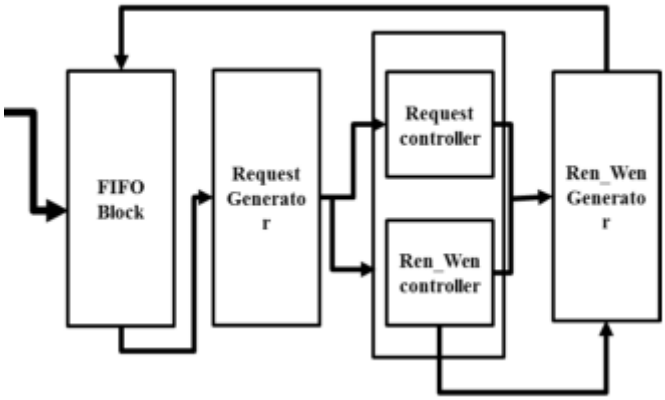

Figure 2.2: NoC Switch Environment

As mentioned in above procedure a NoC Switch must be tested against all possible data transmissions from one switch to the other along with routing paths. The above NoC environment consists of buffers, switches, Arbiters and links. Which makes the design complex and also NoC uses packet structure to route the data. Testing NoC using randomly generated test patterns will not give assurance of complete data routing coverage. Hence functional test patterns are generated to test the NoC environment; these functional test patterns are designed in such a way that all the possible routing paths are triggered. Functional patterns are designed giving respective destination address at the source location. The routing paths between the source and destination addresses are also covered by loading these functional patterns, ensuring the greater test coverage compared to that of random test patterns. In conventionally existing BIST environment an additional block namely Custom TPG is used for generating the functional patterns, which is discussed in the further section III and IV.

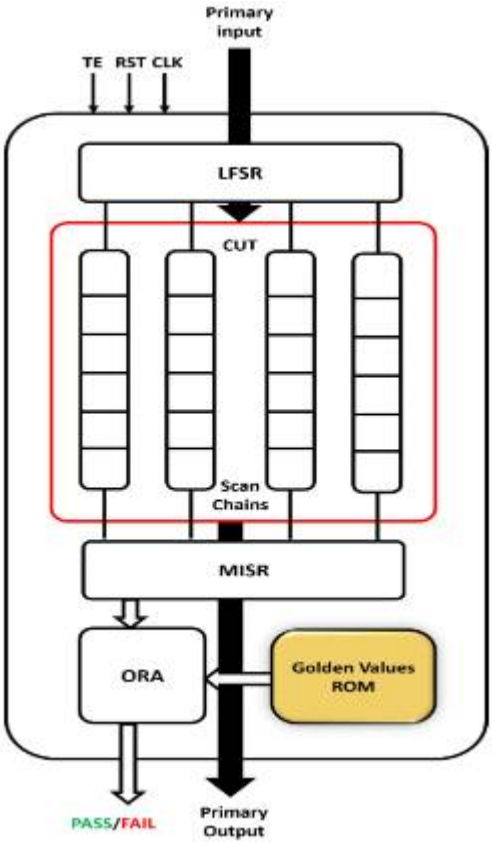

Figure 3: STUMPS BIST Environment

III. STUMPS BIST ENVIRONMENT

In order to provide a NoC Switch with Self-test mechanism a BIST based environment with additional TPG block is required. In this section, the basic working procedure of conventionally existing STUMPS BIST architecture is discussed. The typical architecture of the STUMPS BIST [6] [7] is shown in figure 3, the whole works in functional

ISSN 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering
Vol. 1, Issue 2, August 2012

and BIST mode which is controlled by test mode signal. The basic components of STUMPS BIST includes scan-chain inserted CUT (NoC Environment as CUT), TPG, Compactor, ROM and ORA. The external LFSR is basically used as TPG, which is used to generate pseudo random test patterns, which determines the fault coverage, testing power, testing time and testing energy. Test response after the application of test vectors are compared with the golden response and the test output ―pass/fail‖ is generated which is used to represent whether the chip is subjected to defects or not, and also for diagnosis purposes. The STUMPS BIST is much efficient in terms of test time, test cost and fault coverage when compared to other BIST architectures [2] [3], because of which the test architecture proposed for NoC is of STUMPS BIST.

The flow of testing process in STUMPS BIST is given below,

i. The chip is put into test mode, LFSR generates the test patterns and it is applied to the CUT.

ii. The data from LFSR is loaded into scan chains by making scan enable signal high. The scan enable is made low for one clock cycle for capturing the combo response which is shown in figure 3.1.

iii. The output from the CUT is taken and given as input to compactor [4], where the compactor compact‘s the test response.

iv. The test result is generated as logic =1‘ if compacted test response matches with the golden response stored in ROM else logic =0‘ is generated, which indicates the chip is bad. The Environment is modified by adding up a custom TPG block to provide the patterns covering all the possible routing directions and addresses. The working procedure and implementation of proposed architecture is discussed in the next section IV.

IV. PROPOSED NOC SELF-TEST ARCHITECTURE

In the proposed NoC switch test architecture a self-test mechanism is introduced. Functional pattern block is additional to the basic BIST environment. which is shown in the proposed NoC test Structure figure 4. NoC switch architecture is generic and complex, the functional test patterns will give the better fault coverage compared to test patterns generated by ATPG. During capture mode the inputs to the CUT is driven from the custom TPG logic block to generate the functional patterns and during shift mode patterns from the LFSR block is applied. During BIST mode the patterns are generated as shift mode patterns and capture mode patterns which are applied to the CUT resulting in a fault grading [5]. Due to the application of functional patterns from Custom TPG during shift and LFSR patterns during capture increases the fault coverage. The custom patterns are generated from FSM controller the patterns ensure generation of all the possible combinations. During both shift mode and capture mode patterns only from custom TPG is applied to the CUT.

Table 1: Results for proposed NoC self-test Architecture

Pattern Type	Number of Test cycles	Number of Sequences	Number of Patterns	Fault Coverage
ATPG	2574	64	93	97.03%
LFSR	800	14	30	69.39%
Functional	800	14	30	63.08%
Functional and LFSR	800	14	30	90%

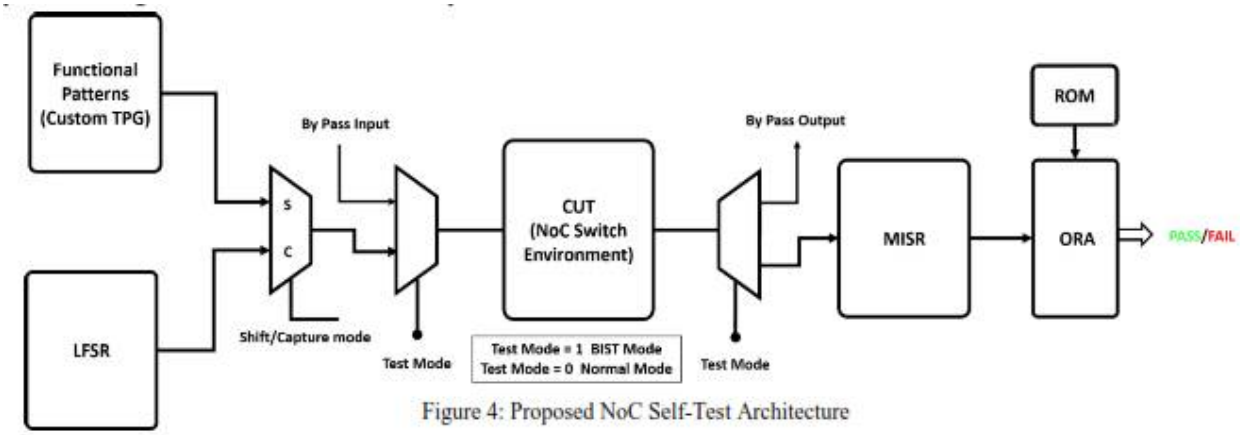

Figure 4: Proposed NoC Self-Test Architecture

ISSN 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering
Vol. 1, Issue 2, August 2012

V. RESULTS AND ANALYSIS

A. EXPERIMENTAL SETUP

To evaluate the NoC self-test architecture proposed in section III, the proposed self-test architecture is built for dummy processing components 3x3 mesh NoC switch which internally consists of typical elements such as crossbar, static priority arbiter, distributed static Routing algorithm, FIFO buffers of 8x2 bits and controllers. For overall environment 8 scan chains are created with the available flops, 8-bit external TPG is constructed for proposed BIST. One 8bit Function Pattern TPG is constructed to grade the patterns with LFSR during capture mode. For calculating fault coverage, thirty patterns are generated with both functional and LFSR patterns. Cadence RC is used for scan insertion, power and area estimation with 90nm TSMC technological library, Cadence encounter test is used for calculating the fault coverage where only static faultsare considered.

B. ANALYSIS

The experimental results for the proposed NoC Self-Test architecture is shown in table 2. From the table 2, the proposed NoC self–test architecture with fault grading of both functional and LFSR patterns is capable of achieving better fault coverage around 90% when compared to the individual patterns of either Function and LFSR with a coverage around 69% and 63%. The number of test cycles used for functional, LFSR and Fault grading architectures is 800 clock cycles. Compared to test patterns from ATPG, the Fault grading architecture in NoC Self-test can overcome the fault coverage of 60% to 90%, with 30% rise in the coverage. So from the experiments it is evident that the proposed NoC self-test architectures achieve fault coverage above 90%.

VI. CONCLUSION

In this paper a new NoC self-test architecture is proposed which increases the fault coverage compared to conventional methods by employing Fault grading technique. The experimental results shows that the proposed self-test architecture with fault grading increases fault coverage by 30% compared to NoC test architecture with only LFSR pattern based and only Custom TPG based. The proposed architecture ensures the coverage beyond 90% compared to conventionally existing test mechanisms for NoC switchbased Networks. The remaining fault coverage can be
achieved by generating the deterministic patterns in CustomTPG. Since NoC self-test architecture is reusable, the test power of NoC structure can be reduced based on the scheduling. i.e. one switch can be actively tested at a time ensuring that other switches are inactive, this reduces the overall dynamic power.

REFERENCES

[1]. Kurt shuler, ―Prioritize Fault tolerance in Design for automotive, Industrial, Military and Medical SoCs‖, Arteris, December 2014.
[2]. Nayana M, G Harish and Siva Yellampalli,―Comparative analysis of BIST architectures‖, Proc. 2 International Conference on Advanced Trends in VLSI and Signal Processing-2014 KSIT,Bangalore.
[3]. Nayana M, G Harish and Siva Yellampalli, ―Modified Low power STUMPS Architecture‖, IEEE International conference in electronics, communication and computational engineering, Adhiyaman Engineering Collage, Hosur, 2014.
[4]. S Praveen, Siva Yellampalli, Ashish Kothari and G Harish, ―Implementation of ATPG and Fault simulation Flow to Fault Grade the Functional Test vectors‖, IEEE International conference in electronics, communication and computational engineering, Adhiyaman Engineering Collage, Hosur, 2014.
[5]. Stroud, E.C., ―A Designer's Guide to Built -In SelfTest‖. ISBN 21-4020-7050-0., Kluwer Academic Publishers, Boston, MA, 313, 2002.
[6]. Ravi Kiran N and Siva Yellampalli, ―Low hardware cost STUMPS BIST‖, IEEE International conference on circuits, communication, control and computing, I4C2014, MSRIT,
Bangalore.
[7]. JoseFlich, DavidBertozzi,―Designing NoC Architecture ‖ *In NOC* ,Taylor And Francis. Ed., 2nded.USA:CRC group,2011,pp.33-66.
[8]. Nikhil Dutt, ―On Chip Communication Architecture‖ *In System On Chip Interconnect*, Morgan and kaufmann.Ed. 3rd ed.USA:Elsevier,2008,pp.439-466.
[9]. Radu Marculescu, Umit Y.Kgras, Natalie Enright Jerger‖ Outstanding Research Problems in NoC Design: system, microarchitecture, and Circuit perspectives‖ ComputerAided Design of Integrated Circuits and Systems, IEEE,vol.28, No.1, Jan 2009. .
[10]. Saad Mubeen, Shashi Kumar‖Desigining efficient source routing for Mesh Topology NOC Platforms‖Computer Aided design Of Integrated circuits and systems,IEEE,vol.12,No.4,june 2013.