International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 03 Issue: 08 | Aug-2016 www.irjet.net p-ISSN: 2395-0072

Fault Tolerant Network-on-Chip Using Built-in-Self Test

Rajeswari R.P.[1], Ms. Manju M.S.[2],

[1]*M.tech Student, Applied Electronics and Instrumentation, Lourdes Matha College of Science & Technology*
[2]*Asst. Professor, Dept. Of ECE, Lourdes Matha College of Science & Technology*

---***---

Abstract **— Multiprocessor Systems on Chip (MPSoC) technology is growing with great complexity to address the need of highly scalable communication infrastructure. As the intellectual properties (IP modules) in the System-on-Chips (SoCs) increases, conventional bus based communication architecture became less efficient to meet up the real time requirements such as power consumption, latency and bandwidth. In order to cope with this growing complexity, "Network on Chip (NoC)" concept was introduced. NoC interconnects the IP modules in SoCs. NoC consists of nodes which is interconnected with links and has a network adapter. It has an inherent redundancy, which helps to deal with many communication bottlenecks and tolerate faults. The inherent structural redundancy of on-chip interconnection networks (NoC) can be addressed by adaptive routing algorithms for providing the connectivity even though the network components are non-functional due to faults, which will appear at an increasing rate with future chip technology nodes. The faults found in a NoC are categorized to permanent and transient. Thus, the main design constraint in NoC is to make it tolerant to these faults.**

In this paper, the detection of a faulty router is done by implementing built-in-self test for each node or device. The main component is a linear feedback shift register. It is a low cost application which can ensure whether a node is functioning good or bad with in less time. Its power consumption is found to be less. The proposed system is to reduce cost and latency of SoCs.

Index Terms **— MPSoC, Network-on-Chip, Linear feedback Shift Register, Built-in-Self Test.**

I. INTRODUCTION

Recently the trend of embedded systems has been moving toward multiprocessor systems-on-chip (MPSoCs) in order to meet the requirements of real-time applications. The complexity of these SoCs is increasing and the communication medium is becoming a major issue of the MPSoC. Generally, integrating a network-on-chip (NoC) into the SoC provides an effective means to interconnect several processor elements (PEs) or intellectual properties (IP) (processors, memory controllers, etc.). The NoC medium features a high level of modularity, flexibility, and throughput. An NoC comprises routers and interconnections allowing communication between the PEs and/or IPs. The NoC relies on data packet exchange. The path for a data packet between a source and a destination through the routers is defined by the routing algorithm. Therefore, the path that a data packet is allowed to take in the network depends mainly on the adaptiveness permitted by the routing algorithm (partially or fully adaptive routing algorithm), which is applied locally in each router being crossed and to each data packet.

Dynamically reconfigurable 2-D meshes NoCs (DyNoC, CuNoC, QNoC, ConoChi, etc.) are suitable for field-programmable gate array (FPGA)-based systems. Thanks to the partial dynamic reconfiguration of FPGAs with varying position and the number of implemented PEs and IPs, higher adaptiveness is allowed in MPSoCs during runtime. To achieve a reconfigurable NoC, an efficient dynamic routing algorithm is required for the data packets. The goal is to preserve flexibility and reliability while providing high NoC performance in terms of throughput. Furthermore, faulty nodes or even faulty regions make communications within the networks harder and even impossible with some routing algorithms, as shown. Therefore, dynamic component placement and faulty nodes or regions are the main reasons why fault-tolerant or adaptive algorithms have been introduced and used in runtime dynamic NoCs. Regarding adaptive or fault-tolerant routing algorithms, several solutions have been proposed. Generally, these algorithms correspond to a modified XY routing algorithm that allows faulty or unavailable regions to be bypassed. In the case of adaptive routing algorithms based on the turn model, zones are defined corresponding to faulty nodes or unavailable regions already detected in the NoC. The neighboring routers of these zones must not send data packets towards these known faulty routers or unavailable regions. Several solutions have been proposed to achieve this constraint. One solution is to include a routing table containing the output port to use for each destination in the network. These tables are updated by an initialization algorithm. The main drawback of this solution is the requirement to invoke the algorithm at a non-specified time in order to update the routing tables of the NoC routers.

Another solution usually applied is the use of chains and rings formed around the adjacent faulty nodes and regions, in order to delimit rectangular parts in the NoC covering all the faulty nodes or unavailable regions. These chains or rings of switches modify the routing tables, which therefore differ from the standard tables realizing the XY routing algorithm. These specific switches integrate in their tables additional routing rules that allow the faulty zones and regions dedicated to dynamic IP/PE instantiations to be bypassed, while avoiding starvation, deadlock, and live-lock situations. Another reliable routing algorithm solution is the use of the de Bruijn graph. This algorithm is deadlock-free and handles the bypassing of faulty links between two switches by assuming that nodes are aware of the faulty link that is connected to them by the use of a detection mechanism. However, these solutions do not give the mechanism to detect a faulty link or router.

With regard to the increasing complexity and the reliability evolution of SoCs, MPSoCs are becoming more sensitive to phenomena that generate permanent, transient, or intermittent faults. These faults may generate data packet errors, or may affect router behavior leading to data packet losses or permanent routing errors. Indeed, a fault in a routing logic will often lead to packet routing errors and might even crash the router. To detect these errors, specific error detection blocks are required in the network to locate the faulty sources. Moreover, permanent errors must be distinguished from transient errors. Indeed, the precise location of permanent faulty parts of the NoC must be determined, in order for them to be bypassed effectively by the adaptive routing algorithm. To protect data packets against errors, error correcting codes (ECCs) are implemented inside the NoC components. Among the well-known solutions, three are usually applied for the MPSoC communications based NoC. First, the end-to-end solution requires an ECC to be implemented in each input port of the IPs or PEs in the NoC. The main drawback of this solution is its incapacity to locate the faulty components (PE, IP, router, data bus, etc.) in the NoC. Consequently, it is inadequate for dynamic NoCs, where the faulty and unavailable zones must be bypassed. Second, the switch-to-switch detection is based on the implementation of an ECC in each input port of the NoC switches. For instance, in a router of four communication directions (North, South, East, and West), four ECC blocks are implemented. Therefore, when a router receives a data packet from a neighbor, the ECC block analyzes its content to check the correctness of the data.

This process detects and corrects data errors according to the effectiveness of the ECC being used. Third, another proposed solution is the code disjoint. In this approach, routers include one ECC in each input and output data port. This solution localizes the error sources, which can be either in the switches or on the data links between routers. However, if an error source is localized inside a router, this solution mechanism disables the totality of the switch. These online detection mechanisms cannot disconnect just the faulty parts of the NoC, and hence do not give an accurate localization of the source of errors. The result is that the network throughput decreases while the network lo ad and data packet latency increase. Moreover, they are not able to distinguish between permanent and transient errors. For all these techniques, each ECC implemented in the routers of the network adds cost in terms of logic area, latency in data packet transmission, and power consumption.

An analysis of the source and destination addresses, as presented, is among the techniques usually proposed to be able to detect faulty routing decisions. When a router receives a data packet, it compares its own address to the destination and source addresses. Then, the router checks its own position in the deterministic XY path of the NoC for the considered data packet. The router performing this checking is able to decide whether the switch from which the packet was received made a routing error or not according to the correct XY path. However, this technique has a major drawback; it is unable to handle the bypass of faulty nodes and unavailable regions. Consequently, this solution cannot be applied in adaptive or fault-tolerant routing algorithms. Indeed, as specified in a turn model algorithm, the structure of the reconfigurable NoC may contain bypass areas in which the switches take routing decisions differently from the XY-routing algorithm. For handling message routing errors in dynamic networks, a new faulty switch detection mechanism is required for adaptive or fault-tolerant routing algorithms.

In this paper, a method is adapted in order to find out a faulty router accurately. Built-in-self Test is a low cost method which is used for testing circuits or devices for verifying whether it is functioning right or wrong.

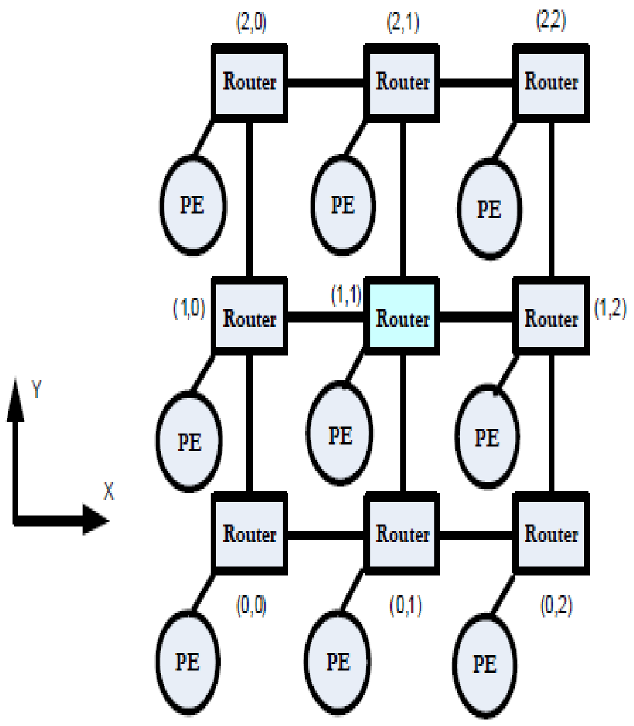


Figure 1.1 Basic Model of NoC

II. BUILT-IN-SELF TEST

Till now we have been looking into VLSI testing, only from the context where the circuit needs to be put to a "test mode" for validating that it is free of faults. Following that, the circuits tested OK are shipped to the customers with the assumption that they would not fail within their expected life time; this is called off-line testing. In other words, in off-line testing, a circuit is tested once and for all, with the hope that once the circuit is verified to be fault free it would not fail during its expected life-time. However, this assumption does not hold for modern day ICs, based on deep sub-micron technology, because they may develop failures even during operation within expected life time. To cater to this problem sometimes redundant circuitry are kept on-chip which replace the faulty parts. To enable replacement of faulty circuitry, the ICs are tested before each time they startup. If a fault is found, a part of the circuit (having the fault) is replaced with a corresponding redundant circuit part (by re-adjusting connections). Testing a circuit every time before they startup, is called Built-In-Self-Test (BIST). Once BIST finds a fault, the readjustment in connections is done by replacing the faulty part with a fault free one.

A. BASIC ARCHITECTURE OF BIST

BIST is basically same as off-line testing using ATE where the test pattern generator and the test response analyzer are on-chip circuitry (instead of equipments). As equipments are replaced by circuitry, so it is obvious that compressed implementations of test pattern generator and response analyzer are to be designed. The basic architecture of BIST is shown in Figure 2. 1.

As shown in Figure 2.1, BIST circuitry comprises the following modules (and the following functionalities)

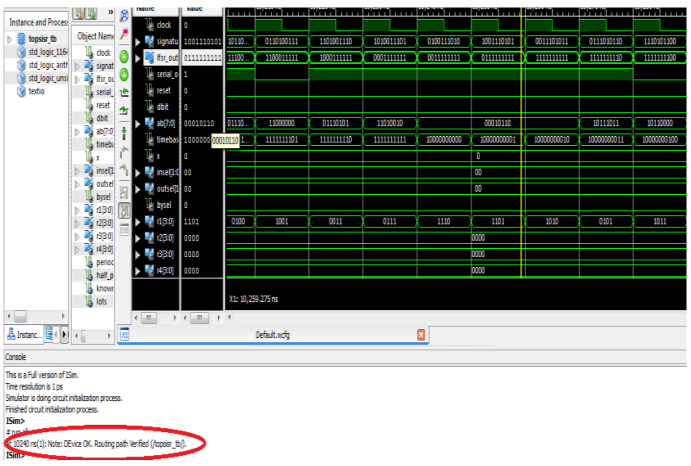
1. *Hardware Test Pattern Generator:* This module generates the test patterns required to sensitize the faults and propagate the effect to the outputs (of the CUT). As the test pattern generator is a circuit (not equipment) its area is limited. So storing and then generating test patterns obtained by ATPG algorithms on the CUT using the hardware test pattern generator is not feasible. In other words, the test pattern generator cannot be a memory where all test patters obtained by running ATPG algorithms (or random pattern generation algorithms) on the CUT are stored and applied during execution of the BIST. Instead, the test pattern generator is basically a type of register which generates random patterns which act as test patterns. The main emphasis of the register design is to have low area yet generate as many different patterns (from 0 to 2n, if there are n flip-flops in the register) as possible.
2. *Input Mux:* This multiplexer is to allow normal inputs to the circuit when it is operational and test inputs from the pattern generator when BIST is executed. The control input of the multiplexer is fed by a central test controller.
3. *Output response compactor:* Output response compacter performs lossy compression of the outputs of the CUT. As in the case of off-line testing, in BIST the output of the CUT is to be compared with the expected response (called golden signature); if CUT output does not match the expected response, fault is detected. Similar to the situation for test pattern generator, expected output responses cannot be stored explicitly in a memory and compared with the responses of the CUT. So CUT response needs to be compacted such that

comparisons with expected responses (golden signatures) become simpler in terms of area of the memory that stores the golden signatures.

4. *ROM:* Stores golden signature that needs to be compared with the compacted CUT response.
5. *Comparator:* Hardware to compare compacted CUT response and golden signature (from ROM).
6. *Test Controller:* Circuit to control the BIST. Whenever an IC is powered up (signal start BIST is made active) the test controller starts the BIST procedure. Once the test is over, the status line is made high if fault is found. Following that, the controller connects normal inputs to the CUT via the multiplexer, thus making it ready for operation.

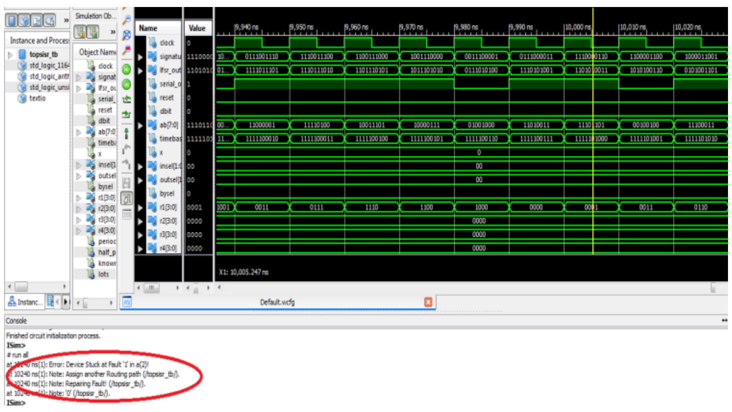
III. SIMULATION RESULTS

The BIST is simulated using Xilinx ISE Design Suite 13.1 and the results shows that the method is abletofind out the error in the device very accurately and time consumption taken for the test is very less. The figure 3.1, 3.2 and 3.3 shows the fault detetction and power consumption respectively.



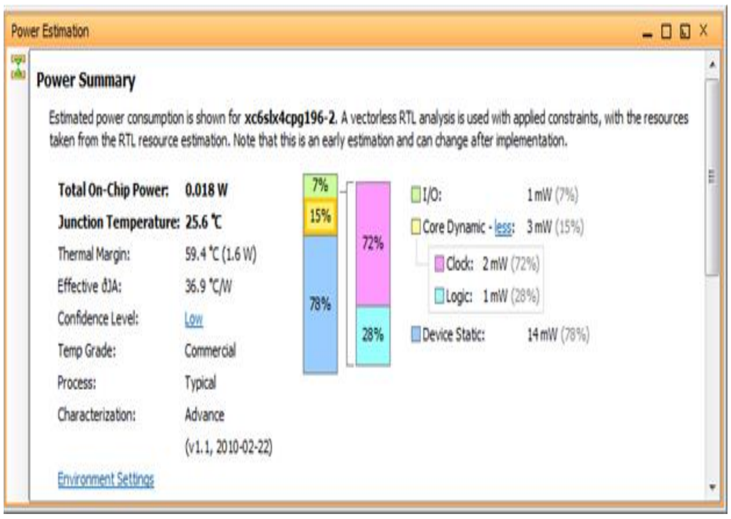
```
Console
This is a Full version of ISim.
Time resolution is 1 ps
Simulator is doing circuit initialization process.
Finished circuit initialization process.
ISim>
# run all
at 10240 ns(1): Note: DEvice OK. Routing path Verified (/topsisr_tb/).
ISim>
```

Figure 3.1 Results when device is OK



```
Console
Finished circuit initialization process.
ISim>
# run all
at 10240 ns(1): Error: Device Stuck at Fault '1' in a(2)!
at 10240 ns(1): Note: Assign another Routing path (/topsisr_tb/).
at 10240 ns(1): Note: Repairing Fault! (/topsisr_tb/).
at 10240 ns(1): Note: '0' (/topsisr_tb/).
ISim>
```

Figure 3.2 Results when device is faulty. It shows a stuck-at-1 fault.



Power Estimation

Power Summary

Estimated power consumption is shown for **xc6slx4cpg196-2**. A vectorless RTL analysis is used with applied constraints, with the resources taken from the RTL resource estimation. Note that this is an early estimation and can change after implementation.

Total On-Chip Power:	**0.018 W**
Junction Temperature:	**25.6 ℃**
Thermal Margin:	59.4 ℃ (1.6 W)
Effective ϑJA:	36.9 ℃/W
Confidence Level:	Low
Temp Grade:	Commercial
Process:	Typical
Characterization:	Advance
	(v1.1, 2010-02-22)

Environment Settings

I/O:	1 mW (7%)
Core Dynamic - less:	3 mW (15%)
Clock:	2 mW (72%)
Logic:	1 mW (28%)
Device Static:	14 mW (78%)

Figure 3.3 Power Summary

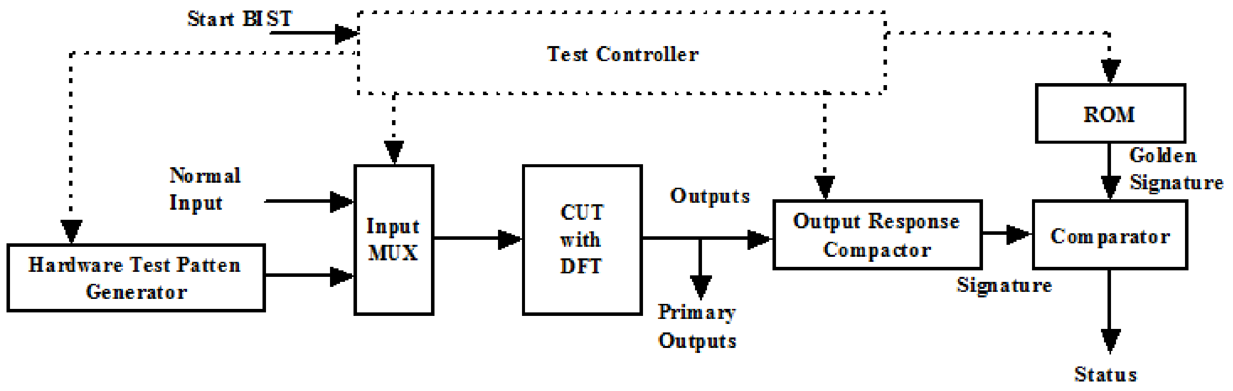


Figure 2.1 Basic Architecture of BIST

IV. CONCLUSIONS AND FUTURE WORK

This paper proposed a NoC with Built-in-Self Test. The aim of this work is to find out the faulty router accurately in real time and in turn to achieve improved performance in terms of latency and time consumption. Through this, the routing nature in a NOC can be made adaptive and fault tolerant. This can be made more adaptive by changing the conventional routing algorithm with shortest path Dijkstra's algorithm Regarding the proposed mechanism, the simulations presented in this paper clearly show the efficiency of the technique, which can aid in data transfer in a NoC faster than the non-reliable one.

REFERENCES

[1] Cedric Killian, Camel Tanougast, Fabrice Monteiro, and Abbas Dandache ,2013 "Smart Reliable Network-on-Chip" IEEE Transactions On Very Large Scale Integration (VlSI) Systems., vol. 22, no. 2, Feb. 2014

[2] K. Sekar, K. Lahiri, A. Raghunathan, and S. Dey, "Dynamically configurable bus topologies for high-performance on-chip communication," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 16, no. 10, pp. 1413-1426, Oct. 2008.

[3] J. Shen and P. Hsiung, Dynamic Reconfigurable Network-on-Chip Design: Innovations for Computational Processing and Communication, J. Shen and P. Hsiung, Eds. Hershey, PA, USA: IGI Global, 2010.

[4] G.-M. Chiu, "The odd-even turn model for adaptive routing," IEEE Trans. Parallel Distrib. Syst., vol. 11, no. 7, pp. 729-738, Jul. 2000.

[5] Y. M. Boura and C. R. Das, "Efficient fully adaptive wormhole routing in n-dimensional meshes," in Proc. 14th Int. Conf. Distrib. Comput. Syst., Jun. 1994, pp. 589-596.

[6] C. Bobda, A. Ahmadinia, M. Majer, J. Teich, S. Fekete, and J. van der Veen, "'DyNoC: A dynamic infrastructure for communication in dynamically reconfigurable devices," in Proc. Int. Conf. Field Program. Logic Appl., Aug. 2005, pp. 153-158.

[7] T. Pionteck, R. Koch, and C. Albrecht, "Applying partial reconfiguration to networks-on-chip," in Proc. Field Program. Logic Appl. Int. Conf., Aug. 2006, pp. 1-6.

[8] S. Jovanovic, C. Tanougast, and S. Weber, "A new high-performance scalable dynamic interconnection for fpga-based reconfigurable systems." in Proc. Int. Conf. Appl.-Specific Syst., Archit. Process., Jul. 2008, pp. 61-66.

[9] S. Jovanovic, C. Tanougast, C. Bobda, and S.Weber, "CuNoC: A dynamic scalable communication structure for dynamically reconfigurable FPGAs," Microprocess. Microsyst., vol. 33, no. 1, pp. 24-36, Feb. 2009.

[10] P. Lysaght and J. Dunlop, "Dynamic reconfiguration of FPGAs," in Proc. Int. Workshop Field Program. Logic Appl. More FPGAs. 1994, pp. 82-94. 32

[11] J.Wu, "A fault-tolerant and deadlock-free routing protocol in 2d meshes based on odd-even turn model," IEEE Trans. Comput., vol. 52, no. 9, pp. 1154-1169, Sep. 2003.

[12] D. Park, C. Nicopoulos, J. Kim, N. Vijaykrishnan, and C. Das, "Exploring fault tolerant network-on-chip architectures," inProc. Int. Conf. Depend. Syst. Netw., Jun. 2006, pp. 93-104.

[13] 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.