Using the Inter- and Intra-Switch Regularity in NoC Switch Testing

Mohammad Hosseinabady, Atefe Dalirsani, and Zainalabedin Navabi
Nanoelectronics-Center of Excellence, University of Tehran 14399 Tehran, Iran
{mohammad, atefeh}@cad.ece.ut.ac.ir, navabi@ece.neu.edu

Abstract

This paper proposes an efficient test methodology to test switches in a Network-on-Chip (NoC) architecture. A switch in an NoC consists of a number of ports and a router. Using the intra-switch regularity among ports of a switch and inter-switch regularity among routers of switches, the proposed method decreases the test application time and test data volume of NoC testing. Using a test source to generate test vectors and scan-based testing, this methodology broadcasts test vectors through the minimum spanning tree of the NoC and concurrently tests its switches. In addition, a possible fault is detected by comparing test results using the inter- or intra- switch comparisons. The logic and memory parts of a switch are tested by appropriate memory and logic testing methods. Experimental results show less test application time and test power consumption, as compared with other methods in the literature.

1 Introduction

The emergence of system-on-chip (SoC) platforms consisting of a large number of embedded processors is imminent. The communication among the cores of these SoC cannot be handled with traditional bus structures. Thus, network-on-chip (NoC) paradigm is emerging as a viable and attractive alternative to address the communication among cores in an SoC [1].

Using the packet concepts, an NoC transfers data between cores. A typical NoC consists of three main parts: switches to route the data packets, interfaces that connects each core to a switch in the NoC, and interconnections among the switches. Switches are connected together in a regular (*e.g.,* mesh) or irregular topology. Figure 1 shows an irregular NoC architecture that implements the MPEG decoder [4]. In this figure, the NoC architecture has seven switches (from s1 to s7), and twelve interfaces between cores and switches.
A switch of an NoC consists of ports that get/send data packets from/to the other switches or cores, a crossbar switch that transfers data packets from the input to the output port, and a router that determines the output port for an incoming packet. In addition, a port has a buffer to store the receiving packets. The router has a routing table to save routing information.

Because NoCs integrate more and more Intellectual Property (IP) blocks into a single chip, testing this class of SoCs becomes an important topic.

1.1 Previous works

Several researchers have presented different aspects regarding the design and implementation of on-chip networks [1], [2], and [3]. Furthermore, the reuse of the NoC as Test Access Mechanism (TAM) has been presented as an efficient method for the test of embedded IP cores, with reduced area, pin count, and test time costs [5]. In addition, test of NoC elements and architecture is one of the active research areas.

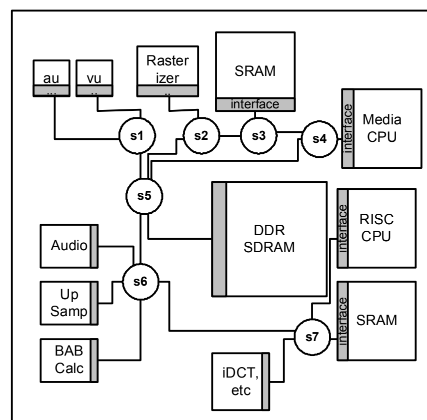
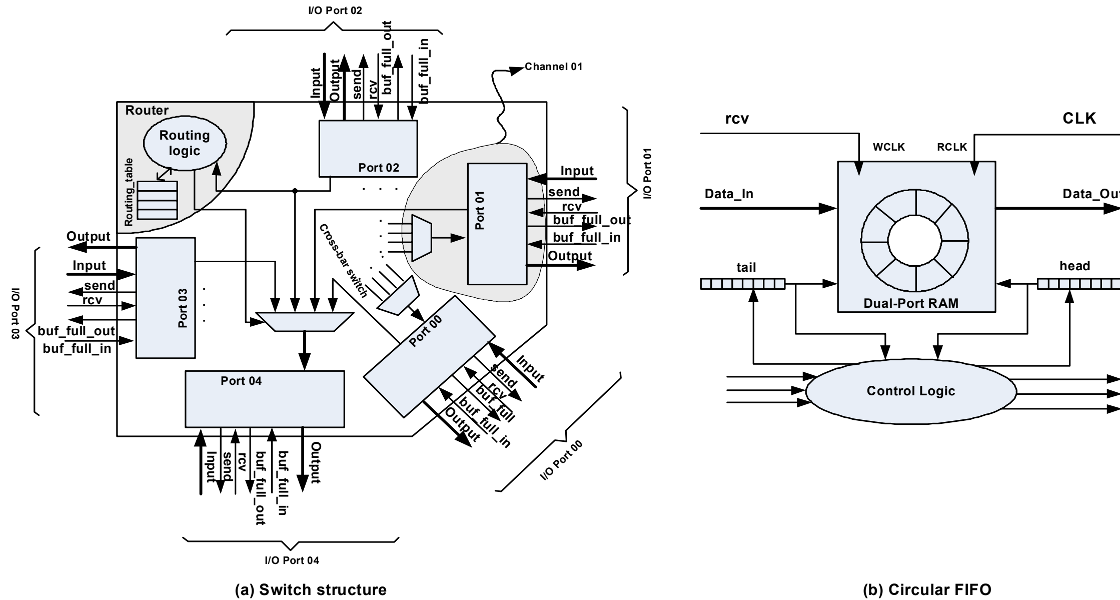


Figure 1 MPEG Decoder [4]

Grecu, *et al.* [8] present a built-in self-test methodology for testing the inter-switch links of network-on-chip (NoC) based chips. This methodology uses a high-level fault model that accounts for crosstalk effects due to inter-wire coupling. Tran, *et al.* [7] develop a methodology to test the asynchronous NoC architectures. This method introduces a modular, scalable and configurable architecture to test asynchronous NoC architectures. Amory, *et al.* [6] propose a scalable test strategy for the switches in an NoC, based on partial scan and on an IEEE 1500-compliant test wrapper. The proposed test strategy takes advantage of the regular design of the NoC to reduce both test area overhead and test time. Hosseinabady, *et al.* [9] propose a concurrent test methodology for testing the switches of an NoC. This algorithm broadcasts the test vectors of switches through the on-chip networks and detects faults by comparing output responses of switches with each other.

978-3-9810801-2-4/DATE07 © 2007 EDAA



(a) Switch structure

(b) Circular FIFO

Figure 2 Switch structure

1.2 Our contribution

This paper proposes a scan-based test mechanism to test switches in a regular or irregular NoC architecture. Taking advantage of intra-switch and inter-switch regularity the proposed mechanism tests switches efficiently. Furthermore, the proposed mechanism tests the logic and memory parts of a switch by using different test schemes to decrease the total test application time and increase test efficiency.

To propose our test mechanism, first, we assume all switches in the NoC have the same structure. However, in Section 5, we extend this proposed mechanism to cover different switch structures in the NoC. Because structures of ports in a switch are the same - called *intra-switch* regularity - the proposed mechanism simultaneously sends identical test vectors to ports of a switch, and then compares the test results together to detect a possible fault in a port. In addition, because structures of routers in switches are the same – called *inter-switch* regularity - the proposed mechanism simultaneously sends identical test vectors to router of switches, and then compares the test results together to detect a possible fault in a port. In this mechanism, one (or a few) test source broadcasts test vectors to switches through the minimum spanning tree of the NoC architecture. Our contributions in this paper are as follows: 1) proposing different test schemes for logic and memory parts of a switch, 2) using the intra- and inter-switch regularity to reduce test application time.

The rest of this paper is organized as follows. The Next section describes the switch structure that is used through this paper as preliminaries. Section 3 proposes our test strategy for testing switches in an NoC based SoC. The test time due to the proposed test mechanism is computed in Section 4. Section 5 extends the proposed mechanism for more general NoC architectures. Experimental results appear in Section 6. Finally, Section 7 concludes the paper.

2 Preliminaries

NoCs can be defined as a set of structured switches and point-to-point channels interconnecting the processing cores of an SoC, in order to support communication among them. As mentioned, switches transfer data from a source core to a destination core by using packets. We consider that each packet in the NoC consists of several consecutive 8-bit flits starting with a header flit, following by data flits, and ending with a tail flit.

Switch structure

In order to implement our test methodology, we have designed a simple switch. Figure 2-a shows the structure of our proposed switch for an NoC. The same as switches in the wide-area networks, this switch consists of three parts: I/O ports, a router and its routing table, and a crossbar switch.

- **Ports**: A port in the switch has two 8-bit data lines to receive and send flits (*Input* and *Output* signals in Figure 2-a) and several controlling signals to handshake the data transfer between two adjacent switches. In addition, the designed port has a circular FIFO buffer with the capacity of 8 flits at its input to save received flits. As shown in Figure 2-b, the circular FIFO buffer has four parts: a dual-port RAM memory, a logic controller, and two counters (i.e., *head* and *tail* of Figure 2-b) to determine read and write addresses.
- **Router:** the router part consists of a routing table which saves routing information and a routing logic that determines the port to which the receiving flits should be delivered. The designed router uses a static routing

algorithm with the wormhole switching scheme to route packets through the NoC. Using the round-robin scheduling scheme, the router arbitrates among ports.

- **Crossbar switch**: using multiplexers the proposed switch implements the crossbar switch structure.

In the rest of the paper, a *channel* refers to the combination of a port and the multiplexer of crossbar switch that feeds that port (Figure 2-a).

3 Proposed Switch Testing

Our test mechanism separately deals with logic and memory parts of a switch. It proposes two different test schemes for channel logics, router logics, and two different test schemes for FIFO and routing table.

Because, there is an intra-switch regularity among channels of a switch, the proposed test scheme applies test vectors to the ports of a switch and compares their outputs together. This scheme is called *intra-switch testing*. This scheme concurrently tests ports of all switches of the NoC.

On the other hand, there is an inter-switch regularity among routers of various switches in the NoC; hence the proposed test scheme applies test vectors to the routers of all switches in the NoC, concurrently. This scheme is called *inter-switch testing*. The details of these two schemes are explained in Sub-section 3.1.

Similar inter- and intra- switch testing schemes are used to test memories in the NoC, which are explained in Sub-section 3.2.

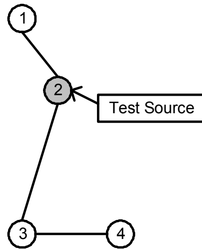


Figure 3 A simple NoC architecture

Note that, a Test Access Mechanism (TAM) is needed to deliver test vectors to a switch in the NoC architecture. To decrease the test application time and test traffic due to broadcasting test vectors among the switches, the proposed TAM takes advantage of the minimum spanning tree in the NoC graph to broadcast test vectors. Moreover, a test source is connected to the root switch – that is called *Test Access Switch (TAS)* – of this spanning tree.

To elaborate our test mechanism, we assume the simple NoC architecture of Figure 3 as a running example.

3.1 Testing the logic parts

Figure 5 shows the proposed test structure (wrapper and scan chains) to implement the two proposed logic testing schemes (*i.e.,* intra- and inter-switch testing). To make channels identical, we add an extra input (the dotted parts of Figure 5) to each multiplexer of the crossbar. This extra input establishes a loop-back path from a port to itself. The switch wrapper consists of a controller and two comparators (one for inter-switch testing and the other for intra-switch testing). A dedicated line carries the test vectors between two adjacent switches. This line is indicated by the *SI* in Figure 5. Furthermore, each switch wrapper sends the test data to other switches by the *SO* signal. Each switch in the spanning tree receives the faulty status of its downstream switches through the *Error-In* signal and delivers the faulty status of itself and downstream switches to its upstream switch by the *Error-Out* signal. In addition to their own scans chain (e.g., *Chain 1* of Figure 5), channels have a shared scan chain that is constructed by concatenation of crossbar switch inputs (*Chain 0* of Figure 5). The routing logic of each switch has its own scan chain. The incoming test vectors in each switch shift into the scan chains of that switch.

Figure 6 shows the timing diagram of broadcasting the logic channel test vectors through the spanning tree of Figure 3. *S1*, *S2*, *S3* and *S4* notations used in Figure 6 refer to the nodes shown in Figure 3. In the diagram of Figure 6, the test vectors of the two scan chains *Chain 1* and *Chain 0* are shifted in consecutively. Because *Chain 0* provides only the input of the circuit under test, during the *Chain 1* shift-in period the responses of the channels to the previous test vector can be compared together. In this manner, the test application time will be decreased, because the test source does not need to broadcast the test responses.

Figure 7 shows the timing diagram of broadcasting the logic router test vectors through the spanning tree of Figure 3. In this diagram, the test source sends a test vector and its corresponding test response to the switches.

3.2 Testing buffers

A critical step in the test methodology of the NoC architecture is test of the FIFO memories that buffer data at the inputs of the switches. Because FIFOs occupy a considerable amount of silicon area relative to the total area of the NoC data transport medium, potential defects affecting these FIFOs can significantly compromises the yield of the NoC based chips [12].

Because of their hardware overhead existing BIST approaches for testing the FIFO blocks in the NoC switches are not suitable [12]. Therefore in our proposed switch, using a methodology like the method proposed in [11], we applied the extended march test of Figure 4 for single and dual port RAMs by using two controllers. One controller handles the test process for all dual-port RAMs in the inputs of channels and the other controller handles the test process for the single port RAM of the routing table.

$$\Uparrow(w0);\Uparrow(r0,w1);\Uparrow(r1,w0);\Uparrow(r0);$$
$$\Uparrow(w1,r1);\Uparrow(w0,r0);$$
$$\Uparrow(w0\mid w1);\Uparrow(r0\mid r1,w1\mid w0);\Uparrow(r1\mid r0)$$

Figure 4 The march algorithm for buffers [11]

Memory test scheme is composed of two steps: shifting appropriate data into the memory data-line scan-chain and applying memory addresses in the ascending order such that the march algorithm of Figure 4 is implemented.

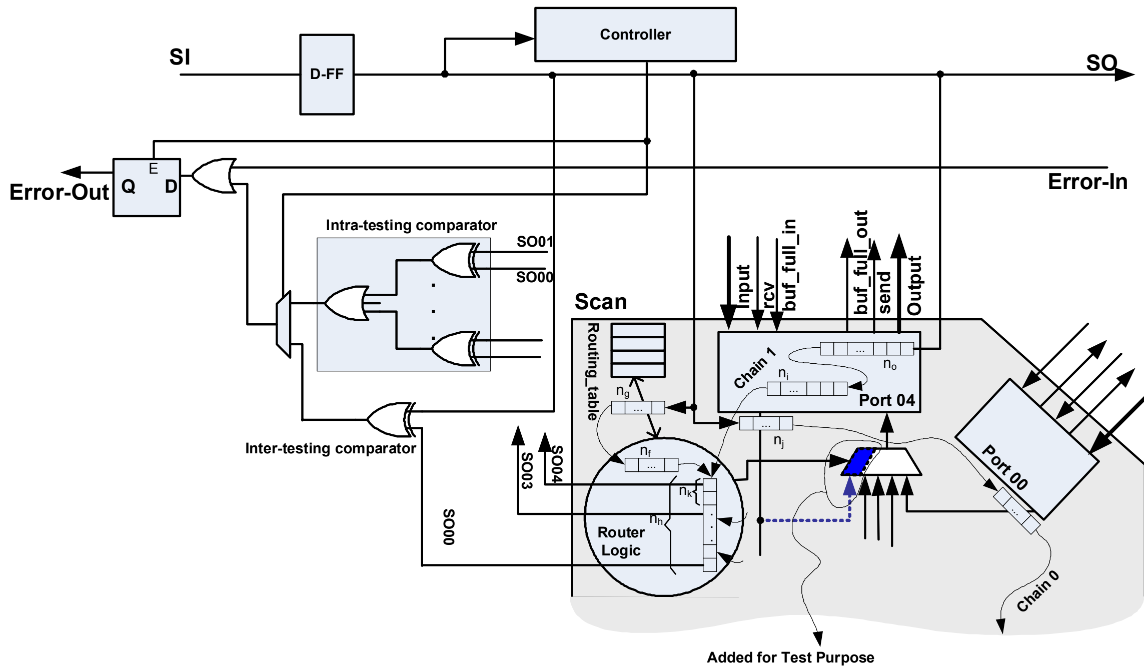


Figure 5 Test structure for a switch

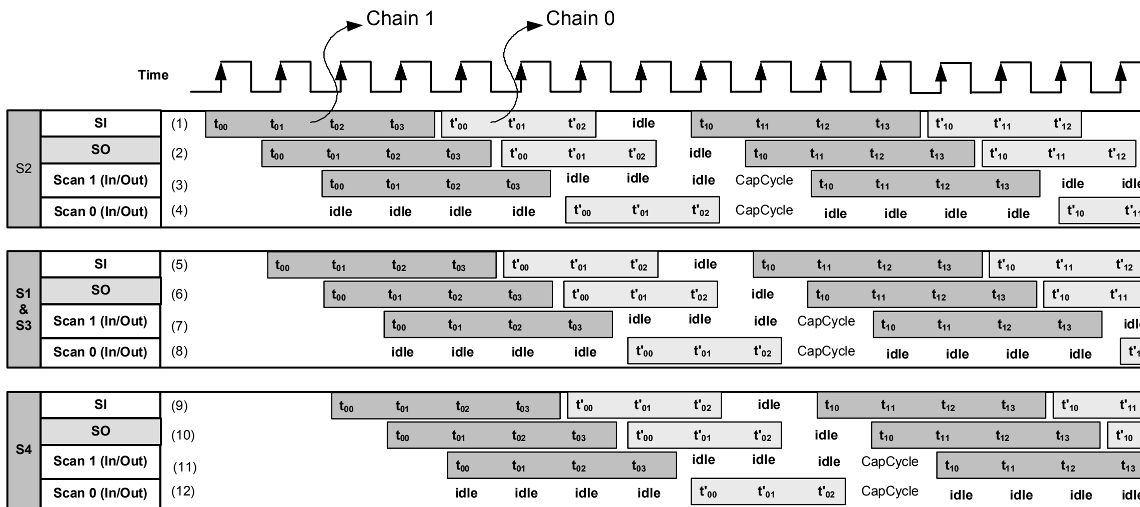


Figure 6 Timing diagram of broadcasting test vectors through channels of Switches in the topology of Figure 5

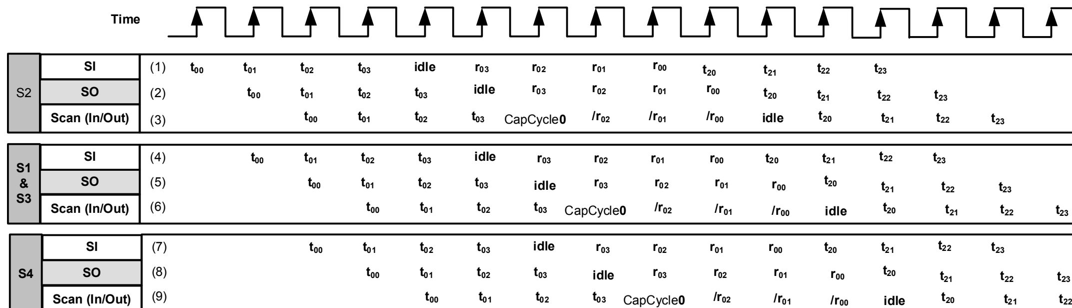


Figure 7 Timing diagram of broadcasting test vectors through routers of Switches in the topology of Figure 5

4 Test Time Calculation

Let N_c be the number of test vectors for the channel of a switch and $|Chain\ 0|$ and $|Chain\ 1|$ be the length of Chain0 and Chain 1 of Figure 5, respectively. Furthermore, let d be the depth of a minimum broadcast tree. Since, switches at the same distance to the test-access switch are tested simultaneously, each switch under test gives the test packet to its successor neighbor switches in the minimum broadcasting tree by one clock delay. Thus, the test application time of channels of switches is equal to:

$$Channels_Test_Time = N_c(|Chain0|+|Chain1|+1)+d \quad \text{Equ. 1}$$

Let N_r be the number of test vectors for the router of a switch and $|TV|$ and $|TR|$ be the length of a test vector and a test response, respectively. Furthermore, let d be the depth of a minimum broadcast tree. Thus, the test application time of routers in an NoC is equal to

$$Router_Test_Time = N_r(|TV|+|TR|+1)+d \quad \text{Equ. 2}$$

Let N_{FIFO} and n_{FIFO} be the length of FIFO buffers and the data length of each FIFO cell, respectively. Equ. **3** computes the test time of the FIFOs of switches.

$$FIFO_Test_Time = 6n_{FIFO}+8N_{FIFO}+d \quad \text{Equ. 3}$$

Let N_{Table} and n_g be the length of routing table buffer and the width of cells in routing table, respectively. Equ. **4** computes the test time of the routing tables of switches.

$$RoutingTable_Test_Time = 6n_g+12N_{Table}+d \quad \text{Equ. 4}$$

5 Extending the Proposed Method

Different switch structures: For NoC architectures with different switch structures or switches with different number of ports, we can group the identical switches and apply the proposed method to each group, separately.

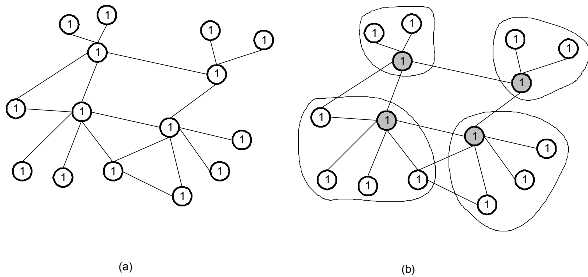


Figure 8 Multiple core trees in a large NoC

Large NoCs: One drawback of one access point for broadcasting the test vector is high traffic overhead in switches, especially the traffic concentration around the access point. To reduce the traffic due to broadcasting the test vectors in NoC, we can increase the number of access points and convert the Network graph to multiple core trees. As an example of this partitioning see the NoC in Figure 8a. This partition include four core trees as shown in Figure 8b. gray nodes in Figure 8b show the access points for the individual core trees.

6 Experimental Results

To evaluate the proposed test mechanism, we have implemented a 5-port switch in VHDL. Each port has eight bit data input and output and a number of control signals (Figure 2). This VHDL description has been synthesized. Table 1 shows gate types and the number of gates of each type in the synthesized switch.

Table 1 Switch statistics

	AND	OR	INV	NAND	DFF
Router	162	150	24	285	52
Channels	280	115	50	445	80
Dual Port RAMs	180	145	20	1595	320
Single Port RAM	20	0	3	143	24

After adding the wrapper (consists of logic and memory test controllers and comparators and other logic of Figure 5), Table 2 shows the test wrapper overhead. The area overheads are estimated by computing the transistor counts in the circuits. For this purpose, we have considered the standard CMOS technology that uses four transistors for NAND and NOR gates, and uses six transistors for AND and OR gates. An inverter uses two transistors, and a flip-flop is implemented using 32 transistors.

Table 2 Test Wrapper Overhead

	AND	OR	INV	NAND	DFF
Test Wrapper	416	376	57	837	184

Wrapper Overhead	%5.74

Test vectors of channels and routers are generated using ATALANTA[10] (a public domain combinational test generator tool) and the test information is shown in Table 3.

Table 3 Test information

	#TV	FC %	#RF
Channel	109	93.31	110
Router	25	98.47	7

#TV = Number of Test Vectors
FC = Fault Coverage
#RF = Number of Redundant Faults

To evaluate our proposed test mechanism in different NoC architectures, we have applied our 5-port switch to four different NoC architectures that have been used in [13] to implement Video Object Plane Decoder and MPEG4 Decoder circuits.

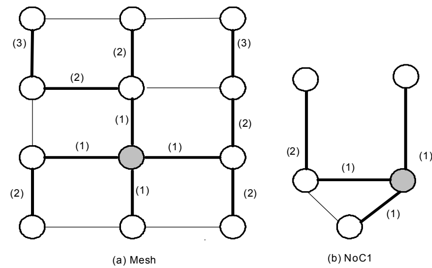


Figure 9 Different topologies for Video Object Plane Decoder

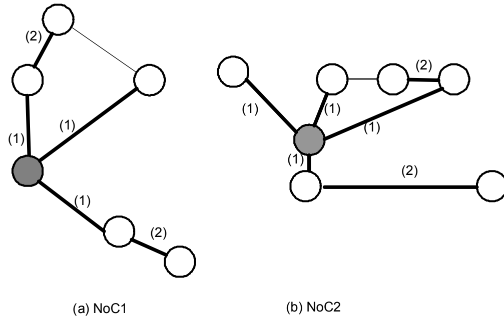


Figure 10 Different topologies for MPEG4 Decoder

Table 4 compares our proposed method with method of [9]. The proposed mechanism of [9] puts the whole switch (containing buffers and FIFOs) in one scan chain. Because, in our mechanism, channels of a switch are tested concurrently, and buffer and FIFOs of a switch are not in the scan chain, our mechanisms is more efficient than the method of [9].

Table 4 Test application time of our method and the method of [9]

		Our approach	Ref [9]	Performance
Video Object Plane Decoder	Mesh	9926	147671	93.27%
	NoC1	9922	147670	93.28%
MPEG4 Decoder	Mesh	9926	147671	93.27%
	NoC1	9922	147670	93.28%
	NoC2	9922	147670	93.28%

Table 5 excludes the memory test of our mechanism and only includes the logic testing as discussed in Section 3.1. We are comparing our logic test scheme with standard full-scan logic testing method. As shown we are seeing an average improvement of 82% with our mechanism. Table 5 shows that we are getting beter than 5x improvement in test time using our approach over full-scan. The 5x factor is due to five concurrent channel testings. The better than 5x improvement is due to the fact that in our approach comparing channel responses to the test vectors does not require sending test responses from test source.

Table 5 Logic test application time of our method and full scan

		Our approach	Standard full scan	Performance
Video Object Plane Decoder	Mesh	9687	54943	82.36%
	NoC1	9685	54942	82.37%
MPEG4 Decoder	Mesh	9687	54943	82.36%
	NoC1	9685	54942	82.37%
	NoC2	9685	54942	82.37%

7 Conclusions

To meet the requirements of an efficient testing for network on chip components, a new mechanism for testing NoC switches has been developed. Using the intra-switch, inter-switch regularity, and regular architecture of an NoC, the proposed testing mechanism broadcasts test vectors to switches to be tested. To increase the efficiency of the test method, logic and memory parts has been handled separately. We have implemented the proposed methodology on a 5-port switch, and the area overhead of the wrapper is 5.74%.

References

[1] P. P. Pande, Cristian Grecu, André Ivanov, and Resve Saleh, Giovanni De Micheli, "Design, Synthesis, and Test of Networks on Chips," in *IEEE Design & Test of Computers*, Vol. 22, No. 5, pp. 404-413, 2005.

[2] T. Bjerregaard, J. Sparso, "Implementation of guaranteed services in the MANGO clockless network-on-chip," in *IEE Proceedings of Computers and Digital Techniques,* Vol. 153, No. 4, pp. 217-229, 2006.

[3] S. J. Lee, K. Lee, H. J. Yoo, "Analysis and-Implementation of Practical, Cost-Effective Networks on Chips," in *IEEE Design & Test of Computers*, Vol. 22, No. 5, pp. 422-433, 2005.

[4] D. Bertozzi and L. Benini, "Xpipes: a network-on-chip architecture for gigascale systems-on-chip," in *IEEE Circuits and Systems Magazine*, vol. 4, no. 2, pp. 18–31, 2004.

[5] A. M., Amory, K. Goossens, E. J. Marinissen, M. Lubaszewski, and F. Moraes, "Wrapper Design for the Reuse of Networks-on-Chip as Test Access Mechanism," in *Proceedings of 11th IEEE European Test Symposium* (ETS'06), pp. 213-218, 2006.

[6] A. M. Amory, E. Briao, E. Cota1, M. Lubaszewski, and F. G. Moraes, "A scalable test strategy for network-on-chip routers," in *International Test Conference* (ITC'05), 2005.

[7] X. T. Tran, J. Durupt, F. Bertrand, V. Beroulle, and C. Robach, "A DFT Architecture for Asynchronous Networks-on-Chip," in *Proceedings of the Eleventh IEEE European Test Symposium* (ETS'06), pp. 219-224, 2006.

[8] C. Grecu, P. Pande, A. Ivanov, and R. Saleh, "BIST for Network-on-Chip interconnect infrastructures," in *Proceedings of the 24th IEEE VLSI Test Symposium* (VTS'06), 2006.

[9] M. Hosseinabady, A. Banaiyan, M. Nazm-Bojnordi, and Z. Navabi "A Concurrent Testing Method for NoC Switches", in *Proceedings of Design, Automation, and Test in Europe* (DATE'06), vol. 1, pp. 1-6, 2006.

[10] H. K. Lee, D. S. Ha, "On the generation of test patterns for combinational circuits," *Tech. Rep.* 12-93, Dept. of Electrical Eng., Virginia Polytechnic Institute and State University, 1993.

[11] S. Barbagallo, M. Lobetti Bodoni, D. Medina, G. De Blasio, M. Ferloni, F. Fummi, and D. Sciuto, "A parametric design of a built-in self-test FIFO embedded memory," in *Proceedings of IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*, pp. 221-229, 1996.

[12] C. Grecu, P. Pande, B. Wang, A. Ivanov and R. Saleh, "Methodologies and algorithms for testing switch-based NoC interconnects," in *Proceedings of IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*, pp. 238-246, 2005.

[13] A. Jalabert, S. Murali, L. Benini. G. De Micheli, "xpipes compiler: A tool for instantiating application specific Networks on Chip," in *Proceedings of Design Automation and Test in Europe Conference*, pp. 884-889, 2004.