A Scalable Built-in Self-Test/Self-Diagnosis Architecture for 2D-mesh Based Chip Multiprocessor Systems

Shu-Yen Lin, Chan-Cheng Hsu, and An-Yeu (Andy) Wu
Graduate Institute of Electronics Engineering
National Taiwan University
Taipei, Taiwan
linyan@access.ee.ntu.edu.tw

Abstract—In this paper, we proposed a scalable built-in self-test/self-diagnosis architecture, Surrounding Test Ring (STR), to detect and locate faulty FIFOs and faulty MUXs for 2D-mesh based CMP systems. Proposed STR supports 97.79% fault coverage in FIFOs and MUXs and tests with 388 ~ 2886 test cycles in different testing methods and mesh sizes.

I. INTRODUCTION

Chip multiprocessor (CMP) system is a popular design in recent years. For the communication of CMP systems, On-Chip Networks (OCNs) have been proposed to overcome area complexity and serious crosstalk problems of traditional wire/bus-based interconnection [2]. Furthermore, as CMOS technology scales down to very deep-submicron (VDSM), problems of breakdown and failure in devices and interconnections are more serious [1]. Hence, fault-tolerant approaches must be considered in CMP systems.

Since every CMP system contains many replicated processors, a common fault-tolerant approach is to deactivate the faulty processors and remap tasks on remaining ones in software application. However, this approach can not handle faulty routers in the OCNs. Faulty routers cause CMP systems unusable unless the OCNs can be reconfigured to work correctly. In order to support this approach, a built-in self-test and self-diagnosis mechanism to detect and locate faults in OCNs is necessary. In previous works, many researches focus on designing testing mechanisms for OCNs [4][5][6][10]. However, faults may only influence certain functions of a faulty router, and undamaged parts can still work correctly if faulty parts are detected and located. In order to support aforementioned features, each OCN must embed built-in self-test and self-diagnosis circuits.

In this paper, we design a scalable Built-in Self-Test/Self-Diagnosis (BIST/SD) architecture, Surrounding Test Ring (STR), to detect and locate faulty FIFOs and faulty MUXs for 2D-mesh based CMP systems. Only FIFOs and MUXs are considered because these components occupy most area in router architectures.

Fig. 1 shows the block diagram of proposed STR on a 3 × 3 CMP system. STR contains two kinds of components: 1) Test modules and 2) CTRL. The test modules and the CTRL form a ring-like connection surrounding the CMP system. Each test module is connected to one router at the boundary of 2D mesh. According to the connections between test modules and the routers, STR can use the regularity of 2D-mesh topology to test and diagnose faulty FIFOs and faulty MUXs in CMP systems. Besides, the architecture of STR is scalable because STR can support any size of 2D meshes.

The advantages of proposed STR are summarized as follows:
- Proposed scalable BIST/SD architecture (STR) to detect and locate faulty FIFOs and faulty MUXs for 2D-mesh based CMP systems.
- Proposed STR supports 97.79% fault coverage in FIFOs and MUXs and tests with 388 ~ 2886 test cycles in different testing methods and mesh sizes.
- Implemented hardware overheads of STR are less than 4.55% ~ 3.08% in different mesh sizes.
- Less than 2.1% false alarms are achieved in two different testing methods.

The rest of this paper is organized as follows: In Section II, we briefly review the related test mechanisms for OCNs. In Section III, we introduce the STR. In Section IV, the implementations and experiments are discussed. Finally, the conclusion is summarized in Section V.

II. RELATED WORKS

Because of the regularity of OCNs, many previous works proposed methods to reuse OCNs as test access mechanisms for testing IP cores [7][8][9]. However, as CMOS technology scales down, problems of failure and breakdown also affect routers and wires on OCNs. Therefore, many researches focus on methods for testing the OCNs [4][5][6][10].

In [4], Hosseinalipour et al. proposed a wrapper with scan-chains attached to each router. One of the routers is defined as a test access switch to receive test patterns from the external test source and broadcast these patterns to other routers. The wrappers compare the output response in each router to detect faults.

In [5], Petersen et al. proposed a BIST engine embedded in the Network Interfaces (NIs) of routers. Each router is divided into two parts: 1) datapaths and 2) controllers. To test the datapaths, some deflection components are implemented in the router’s datapaths and directly controlled by the NI. Test patterns are sent by the NI to test all datapaths attached to the same router.

In [6], Amory et al. proposed a partial scan method on an IEEE 1500-compliant test wrapper, which applied the regularity of OCNs. The test strategy is scalable and independent of the functions of OCNs.

In [10], Grecu et al. proposed another BIST method for inter-switch links between routers. Test Error Detectors (TEDs) are implemented in two ends of each inter-switch link. A Global Test Controller (GTC) injects test packets into OCNs. Then, the TED can analyze responds on the links in OCNs.

Aforementioned test mechanisms can provide abilities of fault detection for OCNs. However, these designs do not support the feature of fault diagnosis. Instead, we proposed a scalable BIST/SD architecture, STR, which is discussed in Section III.

III. PROPOSED STR

In this section, we introduce proposed STR. In part A, we model the behaviors of a typical router and define the impacts of faulty FIFOs and faulty MUXs. In part B, the architecture of STR is introduced. In part C, two different testing methods, Through-Test (ThruT) and Turn-Test (TurnT), are introduced.
B. Architecture of STR

STR contains two kinds of components: 1) test modules and 2) CTRL, as shown in Fig. 1. Test modules generate and receive test packets through the connections between routers and test modules. The CTRL controls the operations of test modules and collects the test results by the connections between test modules and CTRL. The architecture of the test module is illustrated in Fig. 3. Each test module contains three blocks: 1) Test Packet Generator (TPG), 2) Output response analyzer (ORA), and 3) Shift Register (SR). These components are described as follows:

1. TPG: TPG generates test packets and sends these packets to the router it attached to.
2. ORA: ORA receives test packets from other test module and analyzes correctness of the packets.
3. SR: test results of test packets are stored in SRs.

CTRL is connected to the head and tail of the chain formed by test modules. It controls the test flow of STR and analyzes the test results to identify faulty FIFOs and faulty MUXs.

The test flow of STR can be shown as follows:

1. Test faulty routers: test modules can generate and receive test packets through the connections between test modules and routers. Hence, packets transmitted between two test modules can detect faults in the routers which are passed through. The methods to select test packets are important in this scheme. Two methods, ThruT and TurnT, are discussed in Section. III.C.
2. Collect test results: CTRL collects the test results in test modules. The arrows with solid lines in Fig. 4 show the paths to shift the test results from test modules to CTRL.
3. Diagnose faulty FIFOs and faulty MUXs: CTRL analyzes the test results and diagnoses faulty FIFOs and faulty MUXs in OCNs following TABLE II.

C. Testing methods for STR

Two testing methods of the STR, ThruT and TurnT, are discussed in this section.

1) ThruT

In ThruT, two test modules are selected in vertical or horizontal direction. For instance, \( T_0 \) only transmits test packets to \( T_8 \) in Fig. 4. The testing method of ThruT is discussed as follows:

For testing through datapaths (NS, SN, EW, and WE datapaths) each test module transmits test packets once to another test module. Test packets can detect the through datapaths of the routers which are passed through. For instance, faults in SN datapaths of router \((1,0), (1,1), \) and \((1,2)\) are detected between the test packets from \( T_0 \) to \( T_8 \) in Fig. 4.

For testing source/sink datapaths, each test module transmits test packets \( n \) times to another test module in an \( n \times n \) mesh. For testing the source/sink datapaths (LN, LE, LS, LW, NL, EL, SL, and WL), each NI must add a simple circuit to change destination addresses and source addresses in test packets. For example, faults in EL and LW datapaths of router \((2,1)\) in Fig. 4 can be detected between \( T_3 \) and \( T_5 \).

Although the testing method in ThruT is simple, the turning datapaths (NE, NW, EN, SE, SW, WN, and WS) are untestable. In addition, false alarms in the diagnoses may happen and some nonfaulty datapaths must be disabled to guarantee correct system functions. The reason is that faulty datapaths can be detected but not be located.

2) TurnT

The testing method of TurnT is discussed as follows:

For testing through and turning datapaths, each test module transmits test packets \( 2n \) times to other test modules in an \( n \times n \) mesh. For test modules at north and south (east and west) sides, test packets are transmitted to test modules in east and west (north and south) sides. For example, \( T_0 \) transmits test packets to \( T_3, T_{10}, T_7, T_{16}, T_{10}, \) and \( T_{11} \) in Fig. 4. Test packets in TurnT can detect faults in only turning datapaths but also through datapaths. For instance, faults in the WN datapath of router \((0,0)\) and faults in the SN datapaths of router \((0,1)\) and \((0,2)\) are detected between \( T_0 \) and \( T_3 \) in Fig. 4. For testing source/sink datapaths, the case is the same as the method in ThruT.

The testing method in TurnT is more complex than the method in ThruT, but 20 datapaths (through, turning and source/sink datapaths) in each router can be tested. Besides, false alarms can be reduced because more test results can help the diagnoses of faulty FIFOs and faulty MUXs.
IV. IMPLEMENTATIONS AND EXPERIMENTS

In this section, we evaluate the performance of proposed STR. In part A, testability and hardware overhead are described. In part B, false alarms caused by STR is simulated.

A. Testability and Hardware Overhead of STR

The test packets contain custom patterns to test stuck-at faults in FIFOs and MUXs. The testability of the test packets is calculated by a fault simulator, the Turbo Tester system [3]. In Turbo Tester, a fault-simulation tool (TurboFault) can evaluate the fault coverage of custom test patterns from primary inputs. TurboFault can help us to evaluate testability of the test packet. The test packet can achieve about 97.79% fault coverage in FIFOs and MUXs.

The test cycle of two types of STR are listed in TABLE III. ThruT generates less test packets than TurnT and hence ThruT costs fewer test cycles.

The architecture of STR is implemented in Verilog HDL and synthesized with TSMC 0.13μm process. The hardware overheads of STR are less than 4.55%, 3.67%, and 3.08% in 4 × 4, 5 × 5, and 6 × 6 meshes, respectively.

The comparisons between BIST/SD in the STR and other testing methods in [4],[5],[6],[10] are listed in TABLE IV. The STR contains not only self-test features but also self-diagnosis features that differ from previous works. Comparing to other testing methods, the 20PR also performs lowest hardware overhead and shorter test cycles than testing methods in [4],[5],[10]. In the aspect of coverage, the test coverage of 20PR is a bit lower than testing methods in [6] and [10], over 97% fault coverage is still obtainable.

| TABLE III |
| TEST CYCLES OF STR IN DIFFERENT MESH SIZE. |
| Mesh size | 4 × 4 | 5 × 5 | 6 × 6 |
| ThruT | 388 | 545 | 726 |
| TurnT | 1348 | 2045 | 2886 |

| B. False Alarms of STR |

The following simulation focuses on evaluating the ratio of false alarms for ThruT and TurnT.

A static simulator of STR is developed for evaluating ratio of false alarms among the diagnoses of test results. 300,000 cases in 4 × 4, 5 × 5, and 6 × 6 meshes are randomly generated with 2%~10% router fault rates.

The simulation result is shown in Fig.5. The simulation result shows that TurnT has lower ratio of false alarms than ThruT. However, less than 2.1% false alarms are achieved in ThruT and TurnT.

V. CONCLUSION

In this paper, a scalable BIST/SD architecture, STR, is proposed. STR can detect and locate faulty FIFOs and faulty MUXs for 2D-mesh based CMP systems. In our experiments, STR can support 97.79% fault coverage with 388 ~ 2886 test cycles in different testing methods and mesh sizes. The hardware overheads of STR is less than 4.55% in 4 × 4, 5 × 5, and 6 × 6 meshes. Besides, less than 2.1% false alarms are achieved in different test methods.

FIG. 5 RATIO OF FALSE ALARMS IN DIFFERENT MESH SIZE.

REFERENCES