Online Path-based Test Method for Network-on-Chip

Junkai Zhan*, Letian Huang*, Junshi Wang*, Masoumeh Ebrahimi†, Qiang Li*
*Institution of Integrated Circuits and Systems, University of Electronic Science and Technology of China, Chengdu, China
†Department of Electronic System, Royal Institute of Technology (KTH), Sweden
E-mail address: huanglt@uestc.edu.cn(Letian Huang)

***Abstract*—A considerable amount of routers and links remains idle after each mapping application onto the Network-on-Chip based many-core systems. Online path-based test method is a kind of self-test for these idle components. In this paper, a path-based fabric for NoC is firstly proposed. A path serves as the basic component, covering one link and its associated control logic in the routers. One possibility is to apply fault detection on the idle paths, while the other paths continue to operate normally. Moreover, this paper details the hardware implementation, targeting the stuck-at and bridging faults. It suggests a good trade-off between fault coverage, hardware overhead and test time. Experimental results show that the approach achieves 93% of the stuck-at faults in control unit and cover 100% of the stuck-at and bridging faults on the global link within 256 clock cycles.**

I. Introduction

With the evolution of semiconductor manufacturing process, Multi-Processor Systems-on-Chips (MPSoCs) are likely to integrate many cores into a single chip to deal with high-performance parallel computing [1], [2]. The need to handle an extremely growing communication traffic emerges Network-on-Chip (NoC) as a major research concern [3], [4]. On the other hand, the growing device defects and shrinking feature size have a negative impact on the stability of the chips [5], [6]. Physical failures can occur on NoCs for different reasons such as process variation, particle impact, and aging during the lifetime. Early detection of these faults and curing the chip can prevent the network from events such as corrupted packets, dropped packets, and even deadlock.

Various test methods are proposed for physical fault detection in NoCs. Most of them are based on routers. In [7], the authors proposed an online test wrapper by adding two monitor modules (MM) to both ends of the link in order to test the link. This work does not test the control logic inside routers and its hardware overhead is 43.96%. In another work [8], a test method is presented aiming at detecting and locating faults in links. The work engages four routers in the test process. This test method requires four test rounds to complete the fault diagnosis, which imposes a large overhead. DeOrio *et al.* [9] proposed a Vicis-enhanced router including BIST units and test wrapper to diagnose faulty components in a router, and they used the error correcting code (ECC) unit to tolerate the faults in a data path. Nevertheless, the area overhead of this work is more than 50% and it takes 150,000 clock cycles to complete the test. Kakoee *et al.* [10] proposed a functional online test technique that is capable of detecting faults in routers and links. However, in this method, each test task needs a working router and the auxiliary components. The fault coverage of this method is only 85%. These related works cannot perform the NoC test under the condition that weighs the hardware redundancy and test time and fault coverage well. These methods degrade the system performance by disabling the function of one or more routers.

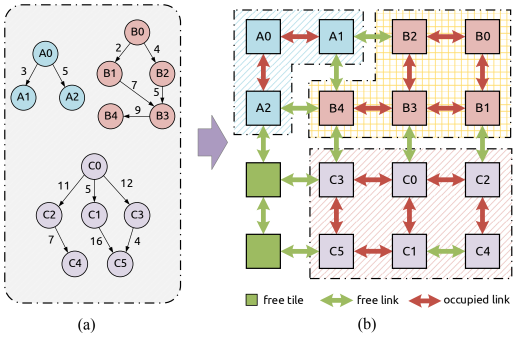

Fig. 1. Application mapping process. (a) Three applications with different task sets. (b) Applications are mapped onto a 4×4 mesh router-based NoC.

Jiang *et al.* [11] showed that there are many free links and some free routers in the system after mapping. As shown in Fig. 1, three applications with three, five, and six tasks are mapped onto a 4×4 mesh NoC where each tile consists of a router and a processing unit [12]. As can be seen, there are two free routers and there is no communication load on some links that connect the routers with different applications, such as the link between A1 and B4. In addition, some links are idle while employed by the same application, such as the link between C1 and C4. This condition motivates the online test method for the free components after each mapping process. The motivation of our test method is to enable NoC-based system with online self-test capabilities in the field, and it conforms to the general goals of online test techniques, such as high test coverage, low hardware cost and minimal system-level performance impact [13].

This paper is organized as follows. Section II describes the proposed test method. In Section III, an implementation of this online test method is described. We present the experimental results and analysis in Section IV, followed by the conclusion in Section V.

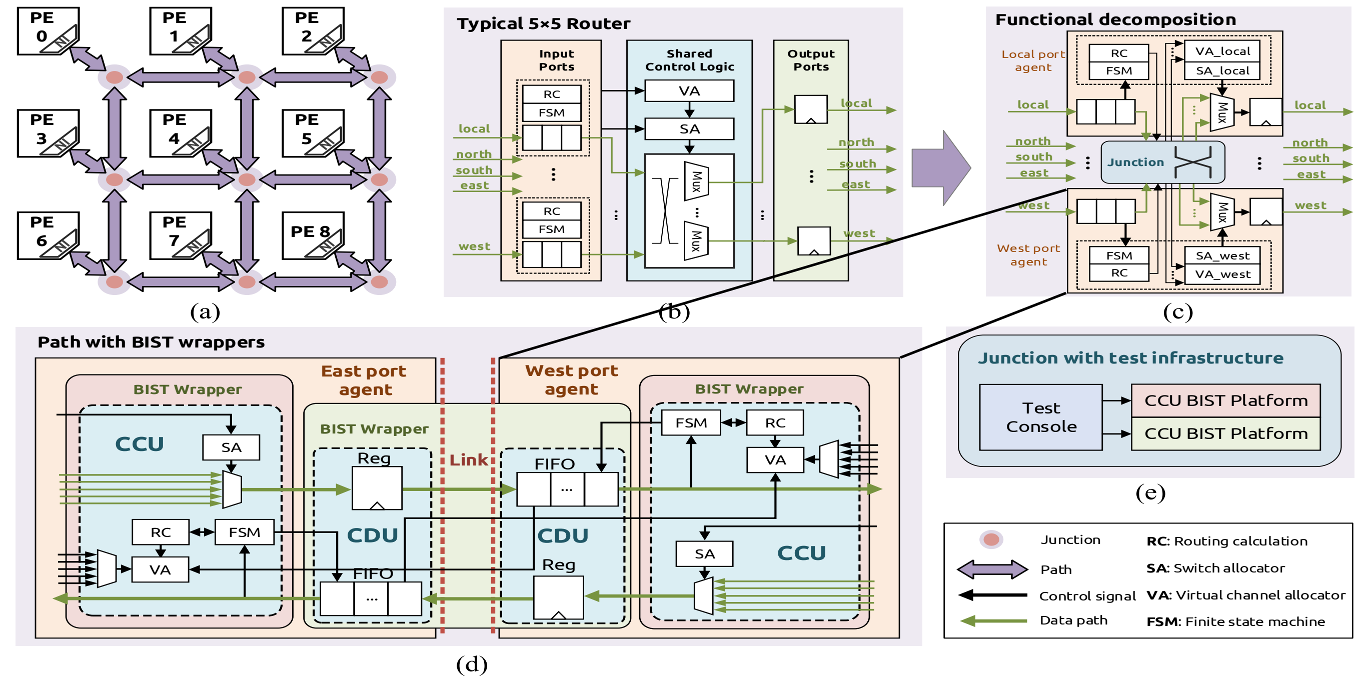

Fig. 2. The path-based NoC fabric. (a) 3×3 mesh path-based NoC. (b) A 5×5 typical router structure. (c) Functional decomposition of the typical router. (d) One path covers two adjacent port agents and the intermediate global link. (e) The junction with the test infrastructure.

II. Path-based Testing for NoC

In this section, we first propose a path-based NoC fabric with finer-grain testability and observability. Then, we describe a test method which is structurally different from the router-based test methods. Finally, the test process is described.

A. Path-based NoC Fabric

The NoC design is commonly divided into routers, links, and network interfaces, which are the main network elements [14]. Fig. 2(b) shows a typical router structure with the shared switch allocator (SA), virtual channel allocator (VA) and crossbar. Guided by this fabric, the test methods for NoC often take a router, a link or a region with multiple routers and links as the test object. In this way, if the test is only performed on the link, the control logic inside the router cannot be tested. Moreover, if the test object contains the control logic, the shared modules should be disabled and thus the entire router can not function. As a result, the current test methods are unable to test all the free components in NoC after mapping without bringing down the working applications.

By considering these constraints, we decompose the NoC into paths and junctions, and define the path-based NoC fabric shown in Fig. 2(a). In the path-based NoC fabric, paths are the basic components instead of routers and links. Junctions connect the paths together and form a mesh network. In a word, we differently form the basic components (i.e. paths) than that in typical designs (i.e. routers and links) to provide a reliable hardware foundation for our online test method.

As shown in Fig. 2(b), in a typical router architecture, the control modules are logically shared among different ports, but in fact, they contain repetitive hardware structures physically. In our design, shown in Fig. 2(c), we divide the shared functional modules into independent port agents plus some structural wires (i.e. junction). A port agent consists of a set of closely coupled modules, including input buffer, output registers, routing calculation unit (RC), FSM, VA, SA and some multiplexers. Based on this structure, a path is defined as a group of two port agents in the adjacent routers and the intermediate global link, as shown in Fig. 2(d). In other words, a path covers a link and its associated control logic in routers, and it becomes a communication structure between adjacent junctions. In this figure, assuming a path between two neighboring routers, the west port agent is in the current router while the east port agent is located in the neighboring router. Different paths are functionally independent and there is no functional coupling between them.

B. Distributed Single Path Test Method

The implementation of the test method requires some test components including necessary test wrappers and test infrastructures. The circuit structure with these test components is shown in Fig. 2(d) and Fig. 2(e). Firstly, we define two units as channel control unit (CCU) and channel data unit (CDU). The CCU are composed of the multiplexer, FSM, RC, VA and SA, while the CDU bundles the input buffer with the output registers to play the role of payload storage and driving. This division is conducive to suit efficient test means to the case, that is, a functional test for CDU and a structural test for CCU. In addition, the shared test infrastructures including the test console and BIST platforms are placed at the junctions as

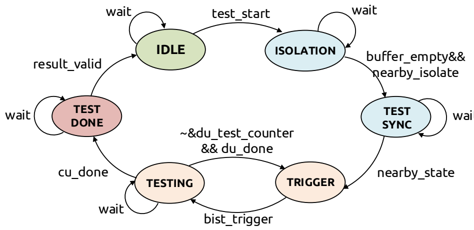

Fig. 3. Finite state machine of the test console for test task management.

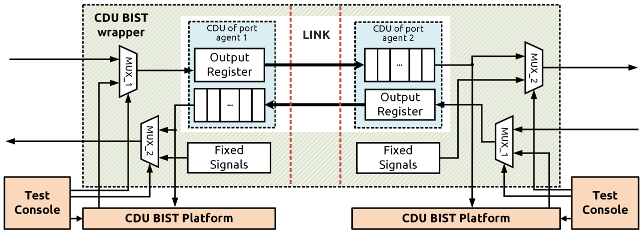

Fig. 4. Test wrappers which encircled the link and the CDUs at two ends of the path.

shown in Fig. 2(e). The paths connected to the same junction share a group of test infrastructure. The test console is a test task manager and the BIST platforms provide test stimuli and compare the test responses.

Using the distributed single path test method, the distributed free paths in the network after the mapping process will be tested in sequence. When the test is running, the wrapper isolates the path under the test from the rest of the network, and the input and output signals are observed and controlled by the test platforms without affecting the normal operation of other paths.

Path-based test method has two features. The existing test methods for NoC generally disable a router with its communication link. These methods block packets in the upstream buffers until the test process is finished. However, in the proposed test method, there is no load on the idle path, it does not affect the network performance. Another feature of this test approach is the finer granularity of the test object that is a single path rather than an entire router. It facilitates locating the fault directly in one transmission direction of the path. This capacity can help in dealing with a smaller faulty object during the subsequent fault recovery and cure process. This information can provide a more detailed basis for the next application mapping.

C. Test Process

In this work, we consider two types of hardware faults that may result in NoC malfunction at the gate level: stuck-at faults and bridging faults. The permanent stuck-at faults can be detected in both CCU and CDU test while the permanent bridging fault can be detected in the CDU test. Moreover, the intermittent faults in CDU are detected by using the redundancy time of CDU test as the CCU test time is much longer than that of CDU. Within a single test task, the CDU test takes place predetermined times, and the results can indicate the time characteristics of the fault.

The test console is responsible for managing the entire operation of the path-based test. Fig. 3 shows the finite state machine of the test console. When the state machine is in the IDLE mode, the path is operational. The state jumps to the ISOLATION state when the start signal is valid. In the ISOLATION state, the state machine blocks the output register and checks the buffer to ensure that the current port agent does not have valid data. In the TEST_SYN state, the test console on one side of the path handshakes with the test console on the other side to ensure that the port agents on both sides are ready for the test. After synchronization, the test console triggers the shared BIST platforms in the TRIGGER state to start the test of CCU and CDU. Then the test task is executed by the underlying hardware in the TESTING state. When the test task is finished, the state machine enters the data collection and output stage, which is the TEST_DONE state. The test results from the CCU and CDU BIST platforms are collected by the test console for further diagnosis and cure.

III. Hardware Implementation

In this section, we describe the hardware implementation of the BIST platforms and the wrappers in detail.

A. Test Platforms

A typical BIST platform with the STUMPS (self-testing using MISR and parallel SRSG) architecture is applied to the CCU test. It uses a linear feedback shift register (LFSR) and a phase shifter to generate a number of pseudo-random test sequences with preset seeds. In this work, we insert multiple scan chains into the CCU to reduce the test time considerably. Then an X-tolerant test response compactor is leveraged to compress the responses which are captured at functional system speed [15]. Finally, a distinct signature is generated by a multiple input signature register (MISR) and compared with the desired response in the test response analyzer (TRA). The BIST controller is responsible for controlling all the test components, it enables the components in a given order and controls the timing. The external BIST interface includes a *start* signal for triggering, a *done* signal for test completion, and a *pass_fail* signal for the test result.

The CDU is tested by predefined packets from the CDU BIST platform on another side of the path. The CDUs of the port agents at both ends and the global link between them are the test objects. In this group of test objects, the data transmission is parallel and there is no combinational logic relation. Therefore, we use all-zero and all-ones and walking-one data flits as stimulus to cover all stuck-at and bridging faults. We use an internal finite state machine to generate the stimulus for the CDU and the one-hot shift operation to decrease the hardware overhead required for generating the walking-one flits. In order to cope with the faulty situation, we use a watchdog to introduce the timeout mechanism. If the test data flit is not received within a limited time, it indicates that the CDU or the link is faulty.

TABLE I
AREA AND POWER CONSUMPTION OF THE TEST COMPONENTS

Module	Area (μm^2)	Percentage (%)	Power (μW)	Percentage (%)
Test Console	269.54	0.89	56.9	1.00
CCU BIST platform	2515.99	8.33	219.0	3.86
CDU BIST platform	1326.18	4.39	194.0	3.42
CCU test wrapper	2758.01	9.13	83.5	1.47
CDU test wrapper	1014.48	3.36	37.6	0.66
Channel Control Unit	3002.33	9.94	306.7	5.41
Channel Data Unit	19306.10	63.94	4773.0	84.17
All test components	7884.20	26.11	591.0	10.42
CCU & CDU	22308.43	73.89	5079.7	89.58
SUMMARY	30192.63	100.00	5670.7	100.00

B. Wrapper

Fig. 4 shows the CDU wrappers at both ends of the path. Because the paths in NoC are synchronous, using multiplexers is a simple and effective method for wrapping. The wrapping process is initiated by the test console to ensure that the buffer in the port agent does not have valid data. This process is divided into two phases: the ISOLATE_IN and ISOLATE_ALL phase. In the former phase, the test console enables the MUX_1 in the wrapper infrastructure to block the input data after current transmission. In the latter phase, the test console enables MUX_2 to mask the request signals from other directions. When the process completed, the test console observes the input signals and controls the output signals of the path. Then, the local test console handshakes with the neighbor test console to ensure that the port agent on the other side is also ready. In addition, The CCU wrapper is similar to the CDU wrapper.

IV. EXPERIMENTAL RESULTS

We implemented an 8×8 NoC with the BIST capacity using Verilog HDL with TSMC 45nm technology to evaluate our proposal. We used Synopsys VCS for the functional simulation and Design Complier for synthesis and TetraMAX for fault coverage evaluation.

A. Hardware Overhead and Power Consumption

Each junction of this NoC supports five physical inputs, and each port agent is equipped with 12-flit FIFO buffer. In Table I, we use the percentage over the whole router structure to present the experimental results. The area overhead of all test components including the test infrastructure and all wrappers is about 35%. The power consumption of the components under the constraint of 500 MHz clock frequency are shown in 4th and 5th columns. The power consumption of all test components is about 10% of the whole equivalent structure.

B. Fault Coverage and Test Time

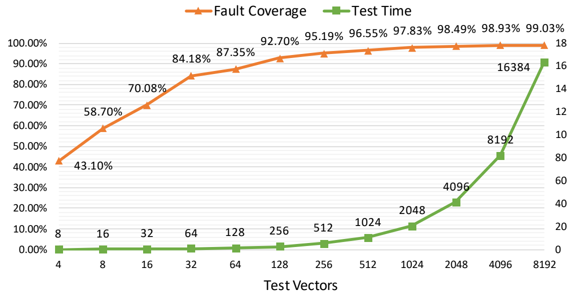

Fig. 5. Stuck-at fault coverage of CCU for different test vector numbers.

TABLE II
COMPARING OUR PROPOSAL BY PREVIOUS WORKS

Methods	Year	Test Object	HR (%)	FC (%)	TT (*cycle*)
DeOrio *et al.* [9]	2012	Router&Link	51	-	150,000
Liu *et al.*[7]	2014	Link	18.4	100	66
Zhang *et al.* [16]	2014	Router&Link	39.1	91.2	400
Kakoee *et al.* [10]	2014	Router&Link	58	85.3	>1500
Bhowmik *et al.* [17]	2016	Link	-	100	416
Aghaei *et al.* [8]	2017	Link	5.1	82.3	70
Proposed method	2018	Router&Link	35.3	92.7	256

Fig. 5 depicts the fault coverage and the test time of CCU at different vector numbers ranged from 4 to 8192. It can be observed that there is a direct linear relationship between the test time and the vector number. Moreover, the fault coverage grows rapidly with the first 256 test vectors and exceeds the value of 90% with the remaining test vectors. Applying 128 test vectors with 93% of fault coverage is considered as a reasonable trade-off in favor. In the CDU test, the fault coverage can reach 100 percent for stuck-at and bridging faults by applying the entire test packet.

In Table II, a comparison is done between the previous works and our proposal in terms of the test object, hardware redundancy (HR) relative to the router area, fault coverage (FC) and the total test time (TT). From the table, it can be seen that our proposal shows advantages in most comparison criteria and achieves a balanced trade-off between them. A better fault coverage and a lower test time make the proposed method more efficient.

V. CONCLUSION

This paper presents a path-based NoC fabric with a finer-grain test capability and observability. A path comprises of an independent global link and its related control logic expanding over two adjacent routers, while a junction solely glues different paths together. The path serves as the basic test object of NoC. Based on this fabric, we introduced an online test method where a path can be tested independently without influencing other paths in the network. The results show that the path-based test method can cover 100% of the stuck-at and bridging faults on the global link, and 93% of the stuck-at faults in the control unit within 256 clock cycles.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China under Grant 61534002, Grant 61761136015, Grant 61701095.

References

[1] P. Ou, J. Zhang, and H. Quan, "A 65nm 39gops/w 24-core processor with 11tb/s/w packet-controlled circuit-switched double-layer network-on-chip and heterogeneous execution array," in *Solid-State Circuits Conference (ISSCC), 2013 IEEE International. IEEE*, Feb. 2013, pp. 56–57.

[2] G. Desoli, N. Chawla, and T. Boesch, "A 2.9tops/w deep convolutional neural network soc in fd-soi 28nm for intelligent embedded systems," in *Solid-State Circuits Conference (ISSCC), 2017 IEEE International. IEEE*, Feb. 2017, pp. 238–239.

[3] P. Vivet, Y. Thonnart, and R. Lemaire, "A 442 homogeneous scalable 3d network-on-chip circuit with 326mflit/s 0.66pj/b robust and fault-tolerant asynchronous 3d links," in *Solid-State Circuits Conference (ISSCC), 2016 IEEE International. IEEE*, Feb. 2016, pp. 146–147.

[4] T. Bjerreraard and S. Mahadevan, "A survey of research and practices of network-on-chip," *ACM Computing Survey*, vol. 38, no. 1, p. article 1, 2006.

[5] C. Constantinescu, "Trends and challenges in vlsi circuit reliability," *IEEE Micro*, vol. 23, no. 4, pp. 14–19, 2003.

[6] S. Borkar, "Thousand core chips: a technology perspective," in *Proceedings of the 44th annual Design Automation Conference*. ACM, 2007, pp. 746–749.

[7] J. X. Liu, J. Harkin, Y. H. Li, and L. Maguire, "Online fault detection for networks-on-chip interconnect," in *2014 NASA/ESA Conference on Adaptive Hardware and Systems (AHS)*, Jul. 2014, pp. 31–38.

[8] B. Aghaei, A. Khademzadeh, M. Reshadi, and K. Badie, "A new bist-based test approach with the fault location capability for communication channels in network-on-chip," *Journal of Electronic Testing: Theory and Applications (JETTA)*, vol. 33, no. 4, pp. 501–513, Aug. 2017.

[9] A. DeOrio, D. Fick, and V. Bertacco, "A reliable routing architecture and algorithm for nocs," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 31, no. 5, pp. 726–739, May 2012.

[10] M. R. Kakoee, V. Bertacco, and L. Benini, "At-speed distributed functional testing to detect logic and delay faults in nocs," *IEEE Transactions on Computers*, vol. 63, no. 3, pp. 703–717, Mar. 2014.

[11] S. Y. Jiang, Q. Wu, S. Y. Chen, J. S. Wang, E. Masoumeh, L. T. Huang, and Q. Li, "Optimizing dynamic mapping techniques for on-line noc test," in *Design Automation Conference (ASP-DAC), 2018 23rd Asia and South Pacific*. IEEE, 2018, pp. 227–232.

[12] D. Wentzlaff, P. Griffin, and H. Hoffmann, "On-chip interconnection architecture of the tile processor," *IEEE Micro*, vol. 27, no. 5, pp. 15–31, 2007.

[13] Y. Li, S. Makar, and S. Mitra, "Casp: Concurrent autonomous chip self-test using stored test patterns," in *Proceedings of the conference on Design, automation and test in Europe*. ACM, 2008, pp. 885–890.

[14] W. J. Dally and B. P. Towles, *Principles and Practices of Interconnection Networks*. Elsevier, 2004.

[15] S. Mitra, M. Mitzenmacher, S. S. Lumetta, and N. Patil, "X-tolerant test response compaction," *IEEE Design and Test of Computers*, vol. 22, no. 6, pp. 566–574, Nov. 2005.

[16] Z. Zhang, D. Refauvelet, and A. Greiner, "On-the-field test and configuration infrastructure for 2-d-mesh nocs in shared-memory many-core architectures," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 6, pp. 1364–1376, Jun. 2014.

[17] B. Bhowmik, S. Biswas, and J. K. Deka, "Impact of noc interconnect shorts on performance metrics," in *2016 Twenty Second National Conference on Communication (NCC)*, Mar. 2016, pp. 1–6.