Chapter 14
The Co-Design of Test Structure and Test Data Transfer Mode for 2D-Mesh NoC

Ying Zhang, Ning Wu and Fen Ge

Abstract NoC(Network-on-Chip) has been proposed as a new solution to deal with the global communication problem of complex SoC(System-on-Chip). However, there are many difficulties in testing and verification for NoC. We propose novel co-design of test architecture and data transfer schemes for 2D-Mesh topology NoC to improve the parallelism of test packets transmission. The testing efficiencies of different structures or transfer modes are evaluated under a coverage-driven and hierarchical NoC testbench, which is based on the VMM verification methodology and SystemVerilog language. The evaluation results of testing cost, testing time and hardware overhead show that the shortening of transmission path and parallel testing effectively decreases the power consumption and testing time. Furthermore, one of these test structures can be proved to the optimal scheme.

Keywords Data transfer · NoC · Parallel testing · Testbench · Test structure · VMM

Y. Zhang (✉) · N. Wu · F. Ge
The College of Information Science and Technology, Nanjing University
of Aeronautics and Astronautics, 29# Yudao Street, College No.4, NUAA,
210016 Nanjing, Jiangsu Province, China
e-mail: tracy403@nuaa.edu.cn

N. Wu
e-mail: wunee@nuaa.edu.cn

F. Ge
e-mail: gefen@nuaa.edu.cn

H. K. Kim et al. (eds.), *IAENG Transactions on Engineering Technologies*,
Lecture Notes in Electrical Engineering 170, DOI: 10.1007/978-94-007-4786-9_14,

14.1 Introduction

With the development of the semiconductor industry and growing scale of the chip design, NoC (Network-on-Chip) will become the solution to the design of complicated system chip instead of traditional bus system chip SoC (System-on-Chip), which will also pose the greatest challenge to associated validation and testing. Furthermore, validation and testing are expected to account for more than 70 % of the whole design work for complex system chip [1]. There is urgent demand for novel testing methods or schemes aiming at testing complicated system chips, especially NoCs.

However, there are many difficulties in NoC testing. Such as the controllability and observability are poor for the network structure and communication scheme, GALS (Global Asynchronous Local Synchronous) method make it necessary for adopting various testing clocks and there is no standard benchmark for evaluation NoC testing, etc. As far as the first difficulty is concerned, the test structure of NoC should be distributed to cooperate with the communication architecture and decrease the hardware overhead. The test control components are inserted to the network and the basic strategy is to transmit test vectors to resources under test through tested components [2]. There are two steps of NoC testing, the first one is to test communication fabric when the router (switch) can be regarded as a IP core and we can apply the same test vectors to all the routers, the second one is to reuse the communication fabric as the test access to resources and transfer test data packets through particular testing input. The verification of communication fabric can be accomplished by the associated testbench [3–5], while the key problem of the second step is how to choose or configure the testing input.

Authors of [6] choose one switch to be connected to the source of test data (called Test Access Switch, TAS) and all the test data are broadcasted from TAS. Though the test scheme is prior to traditional bus-based SoCs in test time and power consumption, it needs complicated routing algorithm and may cause the increase of test time especially for large NoCs. The same situation may be worse when the TAS is the center of the network [7]. Because the center switch will be the hot-spot when all its neighbors communicate with it, the congestion will happen and greatly increase the test time. The solution to this problem is to increase the input/output port number so as to improve the efficiency of parallel testing. Sedghi et al. [8] apply two TASs, one is located on the lower left of the network and the other is at the upper right. The advantage in test time was showed, but the location of TASs can not change and specific routing algorithm is needed.

On the other side, the parallelism of test data transfer is needed to consider [9, 10]. Network communication technology of test data has been widely applied for NoC testing process [11–13]. The communication means can be divided into unicast (point-to-point packet transfer) and multicast (one-to-many packet transfer). Among them, multicast can efficiently improve the parallelism of data transmission, so how to apply multicast transfer mode to testing is worth studying. Fang Fang et al. [9] proposed a multicast paths testing method which

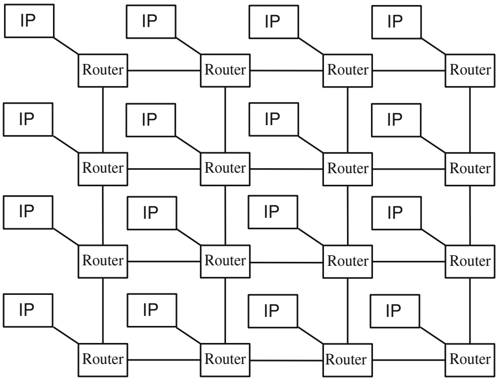


Fig. 14.1 4 × 4 Mesh network

modify multicast communication protocol for test and improve the testing parallelism based on Virtual Channels. However, the testing scheme is based on ordinary test structure.

We propose a novel test structure of 2D-Mesh NoC which has configurable TASs and applied with multicast transfer mode. It is greatly adaptable for parallel testing of routers and resources. The evaluation on the VMM-based testbench shows the reasonable test structure configuration is effective in improving testing efficiency and reducing testing cost.

Hereafter, Sect. 14.2 give a brief introduction of the NUT (NoC Under Test) and the associated testbench. Section 14.3 presents the co-design of NoC test structures and multicast test data transfer. Section 14.4 explains the results of performance evaluation for the test schemes. Section 14.5 draws the conclusions.

14.2 NOC Under Test and the Testbench

14.2.1 NoC Under Test

NoCs can be defined as a set of structured routers and point-to-point channels interconnecting IP cores (Resources). The topology of NoC can be represented as an undirected connected graph $G(N,L)$, where $N = \{n_1,n_2,\ldots n_i\}$ is the set of nodes and $L = \{l_1,l_2, \ldots l_j\}$ is the set of links in the corresponding network. The widely applied topologies are Mesh, Torus, Ring, Hypercube and Fat-tree [14]. Figure 14.1 shows a 4 × 4 2D-Mesh network, which is the structure of our NoC.

For regularity of its structure, Mesh network are easy to implement and have good scalability. Each node in Mesh network is connected to neighbors through regular grid point-to-point links.

The key component of the network fabric is communication node (router). Our router under test is realized in RTL description. The architecture of router is shown in Fig. 14.2. It includes SRAM, Cross Switches, Read/Write control, Routing

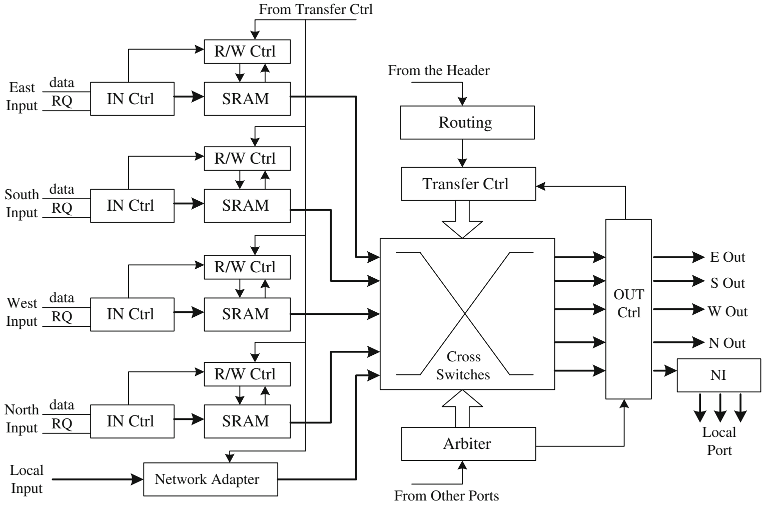


Fig. 14.2 Architecture of router under test

module, Arbiter, Transfer control, Input/Output control, and Network Interface (NI).

Each router has four input channels and four output channels and one network interface for providing communication service to resource nodes. The input control port will test the first packet, checking 32 bits data is in accordance with the parity, if parity checking is correct, data will be written into SRAM, otherwise it will be rejected and retransmission request will be sent to the upper router. Routing module obtains the destination address from the packet header and transfer data with the confirmation of Arbiter. Routing and arbitration module can be adjusted according to certain routing algorithms.

Our NoC employs XY routing algorithm, which is the most commonly used deterministic routing algorithm. XY routing algorithm is related to source node and destination node addresses while irrelevant to network status. It is suitable for 2D-Mesh topology and other similar topology NoCs. It is known to be deadlock-free in meshes (unicast mode) and easy to implement in hardware.

NoCs typically use the message communication model and messages are formed in packets. Each packet is composed of a header and some payloads [15]. The original data packets format of our NoC is shown in Fig. 14.3.

Packet_Type demonstrates the type of data packets, *Destination_ID* and *Source_ID* are destination and source address, *Packet_length* gives the total packet number of the message, *Reserve* field is for extension or user-defined function. *HEC* is used to validate the packet. *Payload* is the actual communication data.

The header of Multicast data packet has many flits, in addition to original unicast packet header information, it will also define the sorting of each destination node of the data in the whole multicast data packets. If the number of the

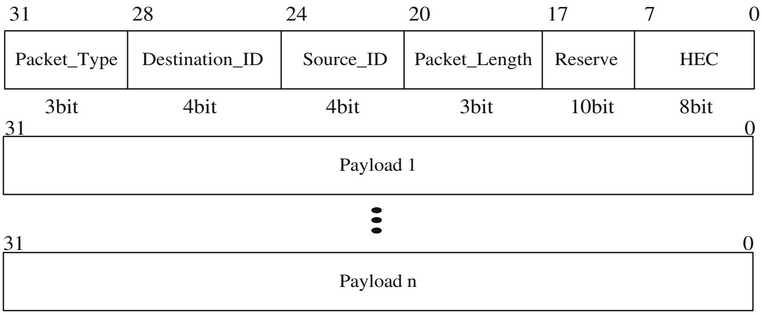


Fig. 14.3 Format of transmission data packets

destination nodes is more than two, the destination nodes information will be added to the header flits, at most 32 destination nodes address. The last flit of the header is the length of sub-packet information, which is the flits number of each sub-packet.

14.2.2 Coverage-Driven and VMM-Based Testbench

The hierarchical NoC testbench is based on the VMM methodology, which integrates assertions, abstraction, automation, and reuse mechanism to improve validation efficiency and productivity [16]. The testbench consists of Test Layer, Scenario Layer, Function Layer, Instruction Layer and NoC Under Test. The transmission between different layers is achieved through channel mechanism [17].

The test process contains four steps and they are generating random stimuli, establishing the testbench environment, executing the test and producing the verification report.

Each module of the testbench is realized in the form of certain class, the UML class diagram to explain their relationship is shown as Fig. 14.4.

The *Environment* class is the core of the NoC testbench. It is almost related to all the modules and controls the building, operating and ending of test process.

The *Config* class determines whether send packets to the specific node or not and the quantity of packets, etc. According to the configuration information, *RU_generator* is responsible for generating random packets with constraints and sending them to next layer through the channel.

The *Driver* receives packets from *RU_generator*, then sends them to NoC and transmits to the *Scoreboard* through callback at the same time. *Monitor* concentrates on output packets from NoC and transmit data to the *Scoreboard* and *Coverage* class. *Scoreboard* achieves the expected packets from *Driver*, gets actual packets from *Monitor*, then compares them and induces the result.

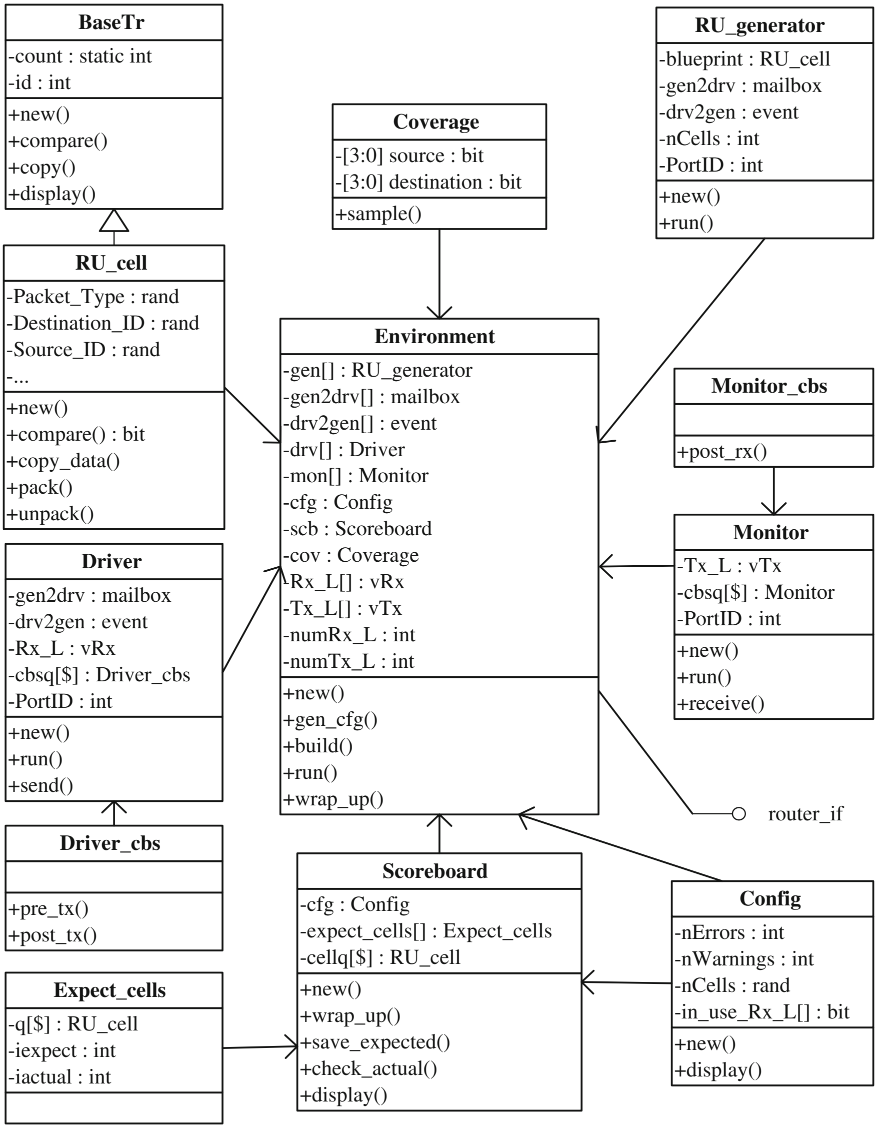


Fig. 14.4 UML class diagram of NoC testbench

The *Coverage* class defines the related cover point and can modify restraint based on the result of simulation. Function coverage is the important parameter to evaluate the efficiency of the testbench, which refers to the validated function of all functions percentage.

Adopting different random seeds and modifying design repeatedly, our testbench finally achieves 100 % function coverage. The testbench can accomplish

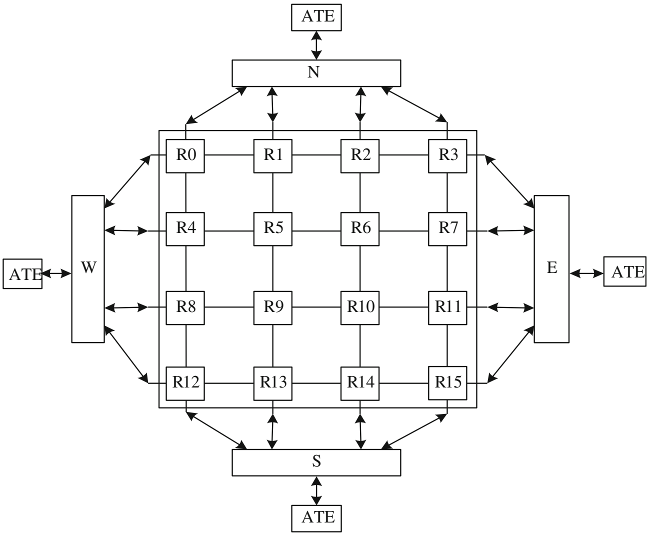


Fig. 14.5 Test structure of NoC with 16 TASs

function verification effectively with good adaptability and expansibility. Only provided the packets format and the transmission timing sequence are known, the testbench can be easily applied to different NoCs without the consideration of topology structure or routing algorithm.

14.3 Design of NoC Test Structures and Test Data Transfer Mode

We propose the test structure which adopts edge of Mesh as TASs, as shown in Fig. 14.5. ATE and peripheral circuits are connected in four directions, while TAS selection can be flexible with the configuration of related circuit. The structure has less test pins and can greatly shorten the transmission path of test data packets, which will be helpful to the parallel testing and decrease testing time as well.

The structure of periphery circuits is shown in Fig. 14.6. The input of periphery circuits comes from output ports of routers and input ports of ATE, while the output of periphery is connected to input ports of routers and output ports of the ATE.

Test data packets are transferred from the ATE and sent to certain router according to the input configuration. Test response packets are sent into the peripheral circuits and passed through selectors to output port of the ATE. Moreover, the arbitration mechanism is added to prevent the competition between routers.

In order to reduce the area overhead, we can configure partial peripheral circuits as shown in Fig. 14.7. Only two ports are selected in each direction. The number

Fig. 14.6 Architecture of periphery circuits

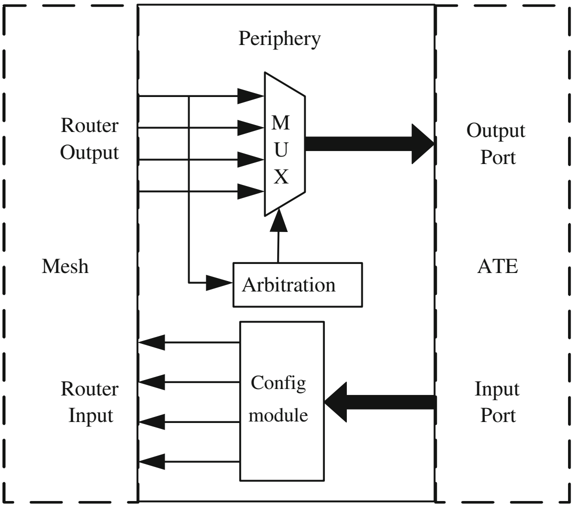
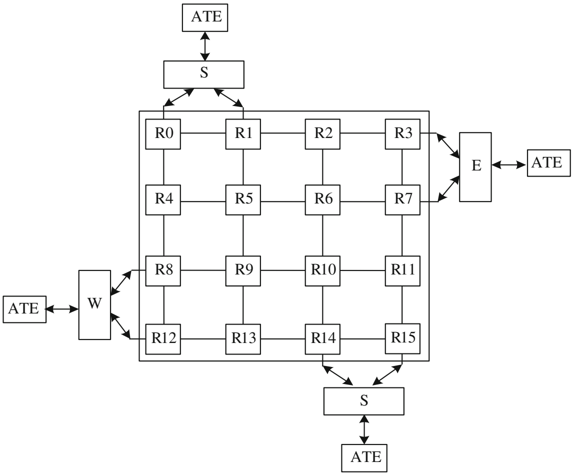


Fig. 14.7 Test structure of NoC with 8 TASs



and position of TASs can be adjusted according to the network size and the application characteristic.

The NoC test data transfer modes can be divided into unicast transmission mode and multicast mode. Unicast mode is sending packets from one port to other single port, while multicast is from one port to more than one ports. Compared with unicast mode, multicast has average shorter delay and smaller network bandwidth, which will improve the efficiency of the NoC testing. Moreover, its advantage will be more apparent when the number of transmission nodes increases.

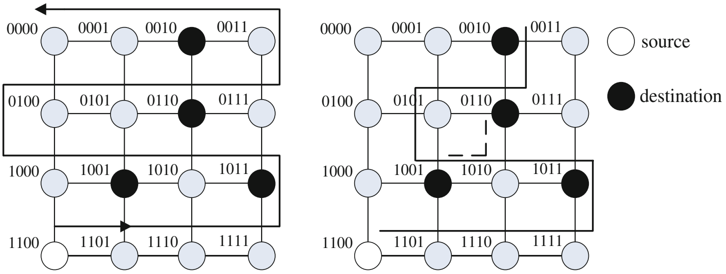


Fig. 14.8 Sorting method and transmission path when source node in four corners

However, there is a key problem need to be solved for multicast mode, that is deadlock avoiding. Provided that the transfer information is test data, we proposed the deadlock-free routing schemes as Figs. 14.8 and 14.9 show.

As the design of test structure is described, source node is located in the external 12 router of NoC. When the source node belongs to one of the four corners, shown in Fig. 14.8, starting from the source node, draw a line cross all the routers, the first destination node is the last multicast node while the last destination node is the first multicast one. The destination node sequence for Fig. 14.8 is 0010, 0110, 1001, and 1011.

When the source node is the other eight edge routers, the line starts from source node, connected to the four corners of the router firstly and then line to the destination node in the same way as the source nodes is the four corners, the sorting mode is shown in Fig. 14.9.

Furthermore, XY routing algorithm also needs to be improved when multicast transfer mode is applied. The rules are as the follows.

(1) The line turning to the 180° angel is not allowed;
(2) When data packets are sent to the last destination node in one row, if the next destination node of the X direction underside the line in the current node (as shown in Fig. 14.8, destination node 1001 is on the next column of 0110), the packet will send along the Y direction to next node;
(3) For source nodes not located in four corners of the router, packet is transmitted to nearest source node in the four corner routers.

After the improvement of destination node sorting and routing algorithm, multicast transmission path become a forward line, which will never repeat the path has taken, so as to efficiently avoid the deadlock.

14.4 Performance Evaluation

For 4 × 4 Mesh network, Fig. 14.10 give four testing schemes with different TAS configuration and each resource chooses the nearest TAS for test vectors transmission.

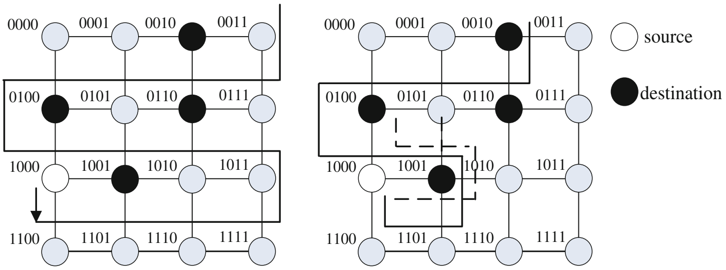


Fig. 14.9 Sorting method and transmission path when source node not in four corners

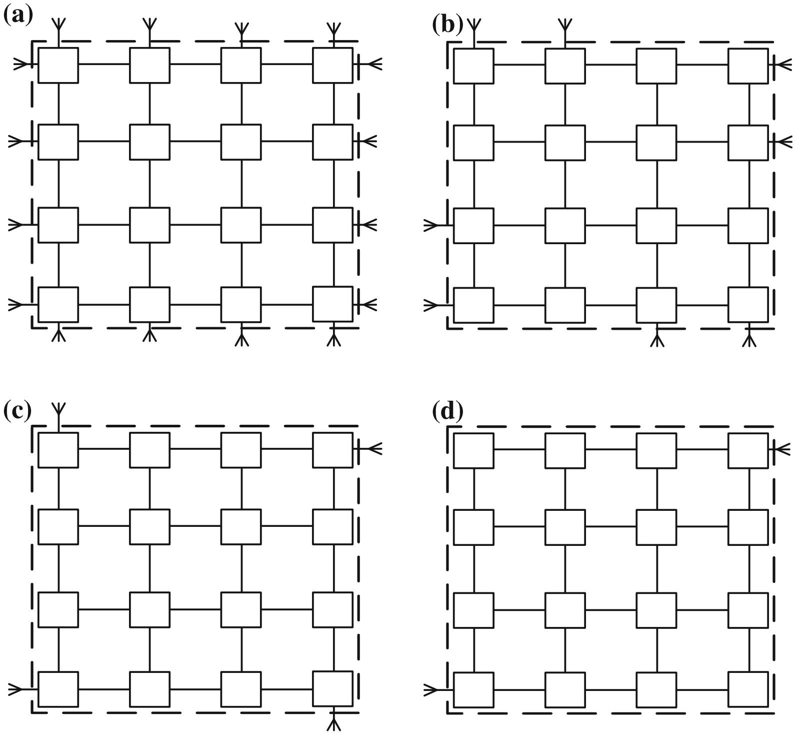


Fig. 14.10 Schemes of different test ports configuration. **a** 16 test ports. **b** 8 test ports. **c** 4 test ports. **d** 2 test ports

How to choose TASs greatly affects test cost, test time and corresponding synthesis area. The test cost mainly refers to test power, which is relevant to the number of test packets and the length of transmission path [18, 19]. The test time can be evaluated by the modified VMM-based testbench and the area overhead of Scheme (a) and Scheme (b) are calculated and compared with the original Mesh NoC on Synopsys EDA platform.

Table 14.1 Comparison of test costs

Scheme	$N_{Ri} = 1$	$N_{Ri} = 2$	$N_{Ri} = 3$	$N_{Ri} = 4$	Sum
(a)	12	4	0	0	20
(b)	8	8	0	0	24
(c)	4	8	4	0	32
(d)	2	4	6	4	44

14.4.1 Estimation of Test Cost

For simplifying and standardizing the estimation of test cost, we assume that the number of test data packets for each resource to be constant *C*. $\mathrm{N_{Ri}}$ and $\mathrm{N_{Ci}}$ are respectively the number of routers and channels in the test path for $Resource_i$ and $\mathrm{N_{Ci}} = \mathrm{N_{Ri}} - 1$. $\mathrm{T_R}$ and $\mathrm{T_C}$ are respectively the average cost of test packets passing through one router and through one channel. *n* is the total number of resources, so the total test costs $\mathrm{T_{all}}$ can be calculated as follows.

$$\begin{aligned} T_{all} &= \sum_{i=1}^{n} (N_{Ri} * T_R + N_{Ci} * T_C) * C \\ &= \sum_{i=1}^{n} [(N_{Ri} * T_R + (N_{Ri} - 1) * T_C] * C \\ &= \sum_{i=1}^{n} [(T_R + T_C) * N_{Ri} - T_C] * C \end{aligned} \tag{14.1}$$

Most parameters in equation (14.1) are constants except N_{Ri}, so the total test cost can be measured by $\sum_{i=1}^{n} N_{Ri}$, that is the number of routers in test paths for all resources. Test costs of four schemes in Fig. 14.10 are shown as Table 14.1.

It is evident that the test cost increases with the decrease of TASs and obviously Scheme (c) and scheme (d) have too high test costs to be good testing schemes. Scheme (a) has lowest test cost but 16 TASs will lead to unacceptable size overhead. Compared with Scheme (a), Scheme (b) has a little increase on test cost and much decrease on test ports, so it should be optimal scheme. Reasonable choice of the position and number of TASs is extremely important especially when the scale of network increases.

14.4.2 Evaluation of Test Time

For the function verification, the driver and monitor module of the testbench are connected to each router to verify the communication between resources. However, for the testing, the driver and monitor module should be connected to periphery circuits to make the original test platform equivalent to the ATE.

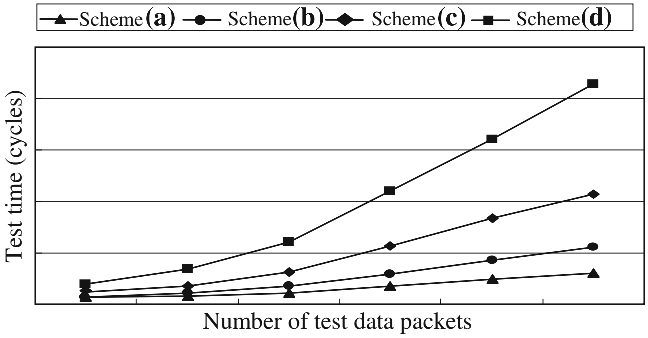


Fig. 14.11 Test time comparison

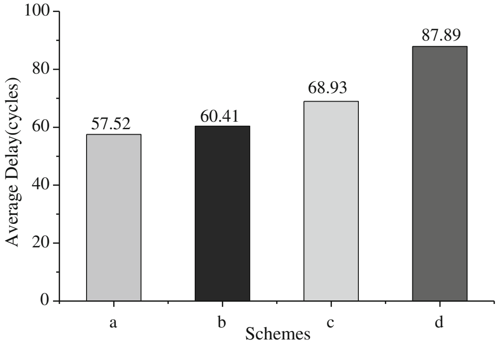


Fig. 14.12 Average delay time comparison

The test time of four testing schemes is shown in Fig. 14.11. X axis is the number of test data packets received by each resource node and Y axis is the time when the last packet is received. When packets number is small, there is no obvious different in test time for all the schemes, however, with increase in packets number, the test time of four schemes gradually increase and the difference between them also becomes increasing. The increase of TASs greatly enhances testing parallelism and effectively reduces the test time.

The average delay time between sending test data packets and receiving response packets is shown as Fig. 14.12. With the decrease of TASs, the average delay also increases gradually, because average path from TAS to resource gets longer. However the difference between Scheme (a) and Scheme (b) is not obvious.

Based on the analysis of the test time with variable packets number and the statics of test packet transmission time, we can conclude that test time is mainly affected by the number of test ports, especially when the amount of communication information increases greatly.

Table 14.2 Area overhead comparison

	Original Mesh (μm^2)	Scheme(a) (μm^2)	Ratio of increase	Scheme(b) (μm^2)	Ratio of increase
Combination logic	959583.24	972296.74	1.325 %	964227.32	0.484 %
Sequence logic	2134660.71	2156475.23	1.022 %	2145984.61	0.530 %
Interconnect	30881190.62	31257792.38	1.220 %	31008210.76	0.411 %
Total size	33975434.56	34386564.36	1.210 %	34118422.69	0.421 %

14.4.3 Evaluation of Area Overhead

For the advantage in test cost and test time, the Scheme (a) and Scheme (b) were synthesized and optimized based on Synopsys DC Compiler and SMIC 0.18 CMOS technology library. Table 14.2 gives the area overhead of original Mesh network and the Scheme (a) and (b). The ratio of increase is relative to the synthesized area of original Mesh network.

The majority of synthesized areas are interconnection lines, so the increase of TASs will surely increase the area overhead. The results of Table 14.2 show the Mesh network with additional test structure has bigger size than original Mesh network and Scheme (b) is superior to Scheme (a) on size overhead.

The testability design can reduce the testing time, but it would increase the cost of the area. We manage to find the strategy to make balance between two aspects. This paper puts forward four testing schemes, Scheme (b) is the best one which has relatively shorter test time and delay time, small synthesized area and its test costs is modest.

14.5 Conclusion and Future Work

We proposed a configurable co-design of test structure and test data transfer mode for 2D-Mesh NoC and given four practical schemes with different TASs configuration. The evaluation platform is the coverage-driven and VMM-based testbench which is originally designed for communication verification on NoC. By adjusting the packet format and routing mechanism, the testbench is applied to evaluating the performance of different testing schemes. Combined with the circuit synthesis results, the optimal scheme can be verified. The experimental results showed that the shortening of transmission path and parallel testing effectively decreases the test cost and test time.

We will research on the optimization of test schedule algorithm based on the designed test structure later.

Acknowledgments This work was supported by the Natural Science Foundation of China under Grant 61076019 and 61106018, the Aeronautical Science Foundation of China under Grant 20115552031, the China Postdoctoral Science Foundation under Grant 20100481134.

References

1. Chris Spear (2006) SystemVerilog for Verification, Synopsys, Inc
2. Grecu C, Pande P, Wang B et al. (2005) Methodologies and algorithms for testing switch-based NoC interconnects. In: Proceedings of the 20th. DFT conf. defect and fault tolerance in VLSI systems, Montery(CA), pp 238–246
3. Li Y, Liang L (2010) An NoC modeling and simulating method with systemC, microelectronic & computer, China, 27(3):78-82
4. Banerjee N, Vellanki P, Chaha KS (2004) “A power and performance model for Network-on-Chip architectures.” In: Proceedings of the DATE Conf. France, pp1250–1255
5. Kreku J, Hoppari M, Kestilä T (2008) Application-platform performance modeling and evaluation. In Proceedings of the forum on FDL conf. specification, verification and design languages, Sttargart, pp 43–48
6. Hosseinabady M, Banaiyan A, Bojnordi MN et al. (2006) A concurrent testing method for NoC switches. In Proceedings of the design automation & test in Europe conf. vol.1, Munich, p 244
7. Zhang F (2007) Design-for-testability optimal strategy and design-for-testability research based on network-on-chip, M.S. thesis, Dept. Electron. Eng., Tsinghua Univ., China
8. Sedghi M, Koopahi E, Alaghi A et al. (2008) An NoC test strategy based on flooding with power, test time and coverage considerations. In: Proceedings of the VLSI design conf. Hyderabad, pp 409–414
9. Fang F, Dong J, Han Y et al. (2010) Parallel testing method for NoC using multicast path. J Electron Meas Instrum. 4(10):911–917
10. Giles G, Wang J, Sehgal A et al. (2008) Test access mechanism for multiple identical cores. In: Proceedings of the international test conference (ITC), pp 1–10
11. OuYang YM, Feng W, Liang HG (2008) An optimized test ports selecting method under power constraint in NoC. J Comput Appl. 28(4):1026–1028
12. Li J, Xu Q, HU Y, et al. (2008) Channel width utilization improvement in testing NoC-based systems for test time reduction. In: Proceedings of electronic design, test and applications, 4th IEEE International Symposium 2008, pp 26–31
13. Zhao JW, Shi YB, Wang Zh G (2009) Research on test strategy for hierarchical network-on-chips interconnection infrastructure. J Electron Meas Instrum. 23(5):34–39
14. Tayan O (2009) Networks-on-chip: challenges, trends and mechanisms for enhancements. In Proceedings of the ICICT’09 conf. information and communication technologies, Karachi, pp 57–62
15. Zhang Y, Wu N, Ge F (2011) Novel test structures for 2D-Mesh NoC with evaluation on the coverage-driven&VMM-based testbench, Lecture Notes in Engineering and Computer Science. In: Proceedings of the world congress on engineering, WCECS 2011, 19–21 October, 2011, San Francisco, USA, pp 797–801
16. Bergeron J, Cerny E, Hunter A, Nightingale A (2006) Verification methodology manual for system verilog, Synopsys & ARM Inc
17. Mao Ye (2008) Research and implementation of VMM-based verification platform, M.S. thesis, Dept. Electron. Eng., Huazhong University of Science and Technology
18. Cota E, Carro L, Wagner F et al. (2002) Power-aware noc reuse on the testing of core-basedsystems. In Proceedings of the ITC’02 Conf. Baltimore, pp 612–621
19. Zeferino CA, Susin AA (2003) SoCIN: a parametric and scalable network-on-chip. In Proceedings of the intergrated circuits and systems design (SBCCI)’03 conf. Sao Paulo, pp 169–174