International Journal of Control and Automation
Vol.6, No.5 (2013), pp.247-262
http://dx.doi.org/10.14257/ijca.2013.6.5.23

Novel Core Test Wrapper Design Supporting Multi-mode Testing of NoC-based SoC

Zhang Ying[1], Wu Ning[1], Ge Fen[1], Chen Xin[1] and Zhou Lei[2]

[1]*Nanjing University of Aeronautics and Astronautics,* [2]*Yangzhou University*
tracy403@nuaa.edu.cn

Abstract

According to embedded IP core structure and various DFT strategies, it is difficult to test NoC-based SoC especially for reuse situation. The paper proposes the design of a compatible IEEE 1500 standard core test wrapper, which makes different IP cores homogeneous from the point of test integration. A test response comparator is added to the test wrapper, which is capable of verifying directly whether there is failure in this IP. The comparator not only improves testing efficiency and also supports various test modes. The function and timing simulation of proposed wrapper are accomplished on ISCAS '89 circuits. A 2D Mesh NoC was constructed based on these wrappered IP cores and implemented unicast and multicast testing. The experimental results show that the wrapper can effectively realize embedded IP test access and isolation, and has the good adaptability to support various NoC test strategies.

Keywords: *Wrapper, NoC, Testing, Multicast*

1. Introduction

The design of NoC is based on reuse of IP cores in the same way as SoC and a large number of IP cores are designed and validated by third party, which greatly promote the design efficiency. Moreover, the reusing of IP cores is not only for the logic design, but also includes testing reuse [1]. Through there are many new issues on complicated SoCs [2], the testing is one of the most significant research contents for SoC and NoC. Since testing data packets need to transfer from external pins to embedded IP cores, the test scheme for single IP core is no longer suitable for lacking the original observability and controllability. In addition, the diversity of test architecture and strategy make it more difficult for testing integration of embedded IP cores. Flexible and effective test structure design is urgently expected for NoC testing.

IEEE 1500 is the standard aimed to test access and isolation of the SoC embedded IP core testing [3]. Wrapper is the key component of the standard, which surrounds the under test IP and makes various IP cores homogenous in the point view of test integration, so as to simplify the reuse of test architecture in system level. IEEE 1500 standard also can apply to the NoC embedded IP testing as NoC is the special SoC, while the test scheme will need to be modified because the test access of NoC mostly relies on the network architecture in stead of the bus for ordinary SoCs. IEEE 1500 only gives the concept of standard wrapper and the compatible wrapper should be implemented on the specific requirement of NoC testing.

There had been some research on test wrapper design for NoC. A. M. Amory [4, 5] proposed the NoC test wrapper design within the constraints of bandwidth and delay and presented simulation results on the ITC benchmark. F. A. Hussin [6, 7] described two other NoC wrappers, which were aimed to increase the utilization rate of bandwidth and decrease

ISSN: 2005-4297 IJCA

test time at the same time. The simulation results were based on the ITC P93791 circuit. The wrappers in [4-7] are designed for under test IP cores on system level and there is no information about the application on specific NoC. Babak Aghaei [8] presented the wrapper for an 8-bit microcontroller and provided specific application information. Test data is directly transferred to the IP in serial through the function port, but no test result or performance analysis is provided based on the application of NoC.

The above wrappers are only considered unicast testing, which transfer test stimulus and test result in packets. The routers control the packet transmission from the specific test source node to destination node and only one packet can be transferred at one time. On the other side, multicast communication is increasingly applied to support parallel computing [9-11], which can be also implemented to parallel testing and greatly improve the test efficiency. However, the wrapper should be improved for support the multicast testing mode besides unicast testing.

The purpose of this paper is to design the compatible IEEE 1500 Wrapper for NoC embedded IP cores unicast and multicast testing. A test response comparator is added to the wrapper, so that the comparison of actual test response and ideal response can be accomplished in the wrapper and transfer to the next nodes at the same time, which will accelerate the testing process and increase the flexibility of the test architecture. A 2D Mesh NoC is constructed with ISCAS '89 benchmark circuits (as IP cores) and applied to unicast and multicast testing process. The simulation results and performance analysis on this NoC indicate that the wrapper can effectively control test data transmission for unicast or multicast testing and has good adaptability.

The paper is composed of 5 sections. Following this introduction, Section 2 presents the typical test process of NoC embedded cores. Section 3 describes in detail the design of the wrapper and test interface. Section 4 presents experimental results of the wrapper function, unicast and multicast testing on specific NoC. Section 5 concludes the paper.

2. Test Process of NoC Embedded IP Core

Basic NoC topologies include Mesh, Torus, BFT, Ring, etc [12] and our design is based on 2D Mesh structure for its simplicity and good scalability [13]. Each IP core is connected to the corresponding router, which implements the data transmission, the best path selection and error correction. The structure of the router in our NoC is shown as Figure 1.

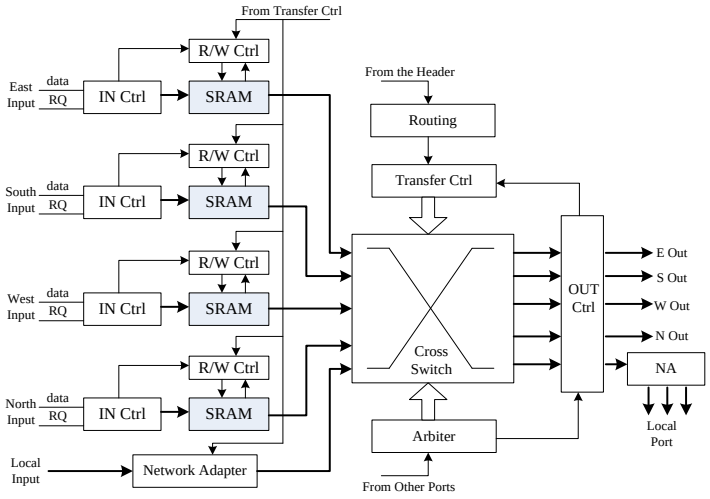


Figure 1. Block Diagram of the Router

NoC typically relies on packet switching communication, so each IP and the router is linked by the Network Interface (NI), which pack the data from IP core, unpack the data from router and realize other functions. A data packet is composed of a header flit and some payload flits. The header flit carries the routing information, packet type (function or testing) and length of total flits or other information. Payload flits carries the communication data for normal function, test stimulus and test response for testing process. Usually routers work in full duplex mode, so the East, West, South, North and Local part each has its own input and output port.

The router module in Figure 1 is divided into several functional parts: read/write control, input/output control, data buffer SRAM, routing unit, transmission control, arbiter, cross switch and network adapter. When the router receives data from other routers, its input control module will check the data flits for 32-bit even parity, if no error, data will be written into SRAM, otherwise flits will be discarded and a request be sent to the upward router to retransmit these flits. Routing module decides the transmission direction based on the header flit and sends a transfer request to output port. If arbiter module agrees to the request, the packet will transfer to the output module through the cross switch, if not, the packet will be suspended until obtain permission. The transfer channel will be released for other packets when current packet transmission ends. The router often relies on dedicated routing algorithms, such as X-First routing [14] for the 2D-Mesh topology. However, the router maybe applies with other routing algorithms for specific function or testing process and the modules in the router are adoptable for the variation.

The NoC data transfer modes can be divided into unicast transmission mode and multicast mode. The former mode is sending packets from one port to other single port, while multicast is from one port to more than one ports.

Unicast transmission is the basic NoC data transmission mode. Source node sends data with routing information in the header flit, and then transfer through the connected routers, the routers determine the direction of the packet transmission, and finally the packet will be sent to the destination node. Because the header of unicast transmission contains only one destination node information, the reliability of information transmission is high and the design of the corresponding router is relatively simple. However, this data transmission mode consumes much time and will influence the efficiency of the whole system.

Compared with unicast mode, multicast has average shorter delay and smaller network bandwidth, which will improve the efficiency of the NoC communication. Moreover, its advantage will be more apparent with the increase of the nodes number. Multicast destination addresses constitute a group and each destination address is referred to as a member of the multicast group. The multicast transmission can be mainly divided into three modes in the following [15].

(1) The multicast communication mode based on unicast: source node continuously sends data to each multicast group members in unicast form. In this way the network bandwidth will be wasted and the burden of the source will increase due to the need to have a complete list of members of the multicast group.

(2) The multicast communication mode based on the tree: multicast tree is to cover all members of the multicast group, and the tree can self-adjust dynamically with the change of the multicast group members. The source node only needs to send the data once, and then the data will be copied and transferred forwarding through the multicast tree branches, so as to reach each destination node. Though the transmission efficiency is improved, the hardware

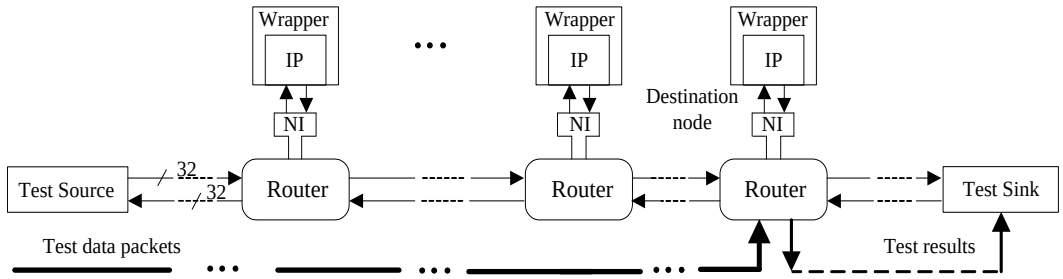
implementation is relatively complex and if one multicast tree node is blocked, then the subsequent node will be blocked.

(3) The multicast transmission mode based on the path: the header flit of the data transmitted by the source node includes the node addresses of all group members in the process of transmission, and with a certain way each multicast group members are accessible and got a copy of data sent to the current network interface, and ultimately the data is sent to all the destination addresses. Source node also sends data once in this method and the hardware implementation is relatively simple. This transmission mode is adopted in our design.

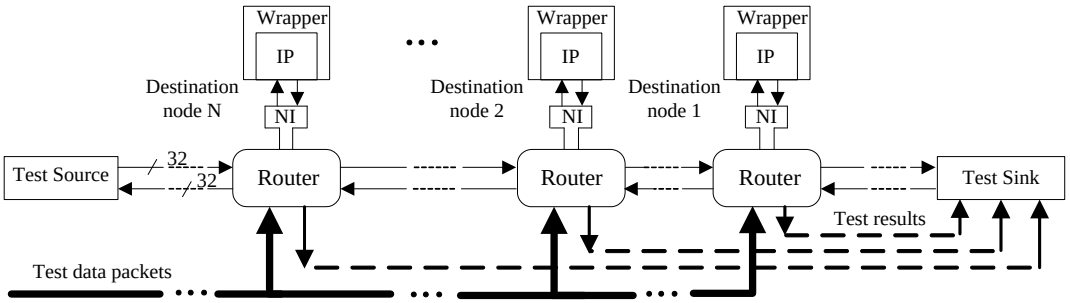
As far as the NoC testing is concerned, since a large number of IP cores are integrated into NoC, the traditional sequential test methods will be inapplicable for too much time overhead. On the other side, the IEEE 1500 describes the systematic test process of embedded IP cores in SoC. The test stimulus is sent from the test source and the test result is transferred to the test sink through the TAM (Test Access Mechanism), usually the TAM is the bus. However, the NoC test generally adopts parallel testing for increasing the test efficiency. The parallel test of multi-core SoC commonly needs to set up the dedicated test channel, which is too difficult to implement for NoC due to the complexity of its wiring. How to effectively transmit test vectors in parallel to multiple embedded IP cores based on the NoC resources so as to implement the NoC parallel testing, which is one of the great challenges of NoC testing.

Reuse NoC Structure as the test access mechanism is a good solution to this problem [16]. The NoC test stimulus and test response are transferred through the routers and communication architecture which are originally designed for function data transmission. In order to realize reuse NoC structure scheme, firstly the test data needs to be divided into flits (the almost same format as the function data) and add routing information and core wrapper control information (such as test mode) to the header flit, which will enhance the testing data transmission efficiency. Secondly, the wrapper of embedded IP cores should have great versatility so as to maintain the stability of data packet format. Finally, after the insertion of wrapper, the multiple scan chains should be approximately the same length to avoid test inefficiency caused by the lack of the consistency in the scan chains.

Test packets will be transferred through the network in unicast or multicast mode. When the NoC is in unicast testing, test stimulus is firstly translated into packet. Then the packet is set into FIFO of the router from the test source. The routing module makes requests to the transmission direction according to the header information, then transmits the packet between routers and eventually sends the packet to the destination node. Therefore, the data packets need to be constructed into the IEEE 1500 standard compatible format by NI and sent to the IP. Finally, test response is added to the header flit, formed the result packet and sent to test sink. Unicast testing process is shown in Figure 2(a). The header flit format of unicast testing is shown in Figure 3(a). *Source_ID* and *Dest_ID* are destination and source address, *P_function* demonstrates the type of data packets (function or testing), *P_length* gives the total packet number of the message, *Parity_b* is used to validate the packet. *Payload* is the actual communication data.



(a) Unicast testing process



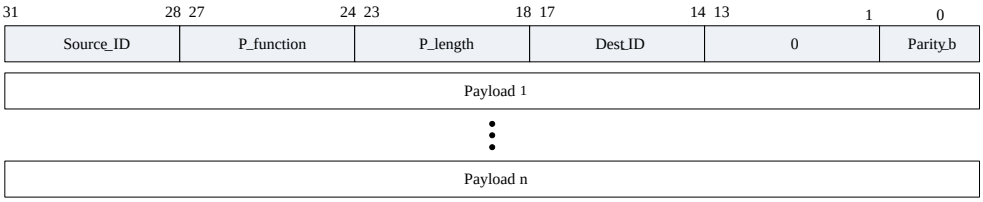
(b) Multicast testing process

Figure 2. Testing Process for Unicast and Multicast mode

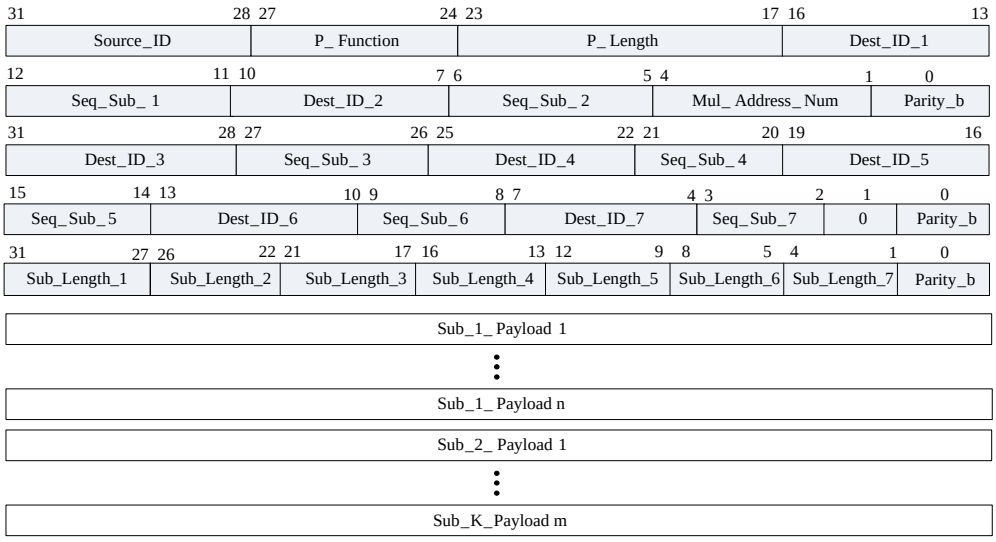
Multicast testing mode is to send multicast packets from test source to multiple embedded IP cores for parallel testing. The test data packets that belong to the IP are received and kept by its NI, while other packets are discarded. The test stimulus is sent to the wrapper and the IP core testing is processing according to the preplanned test mode. The correct test response is sent to our designed wrapper and the response comparator in the wrapper will directly generate the comparison result, which determines whether there is fault in the IP core or not. The test result is sent into the NI, then the reply packet will be generated and transmit through the NoC network into the test sink. Multicast testing process is shown in Figure 2(b).

The black solid lines in Figure 2(b) behalf of multicast data packets and these packets will firstly transmit to the most far destination node and continue to transfer to other nodes until the nearest destination node. The dotted lines show test results from each router and the results packets are eventually transferred to test sink according to the routing information. The testing process will not finish until all test results are sent to test sink.

The multicast header flit format is shown in Figure 3(b). The maximum number of multicast address is 31 and when the number is more than 2, the header is formed by several flits. These flits provide the information of source address (*Source_ID*), all destination addresses (*Dest_ID_x*), the type of data packets (*P_function*), the number of data packets (*P_length*), even parity check (*Parity_b*), the length of sub-packets (*Sub_length_x*) and ordinal number of sub-packets (*Seq_sub_x*), etc. The payload packets are made up of test data for transmission, which consist of sub-packets for under test IP cores. Each sub-packet contains test stimulus and correct test response.



(a) Unicast testing packet



(b) Multicast testing packet

Figure 3. Format of testing data packets

3. Wrapper and Test Interface Design

As the interface between TAM and IP cores, wrapper is used to enhance the interconnectivity of different IP cores and simplify the reuse of core test design in system level, so as to enhance the efficiency of embedded IP testing. IEEE 1500 standard mainly includes the following port: Wrapper Serial Input/Output port (WSI, WSO), Wrapper Parallel Input/Output port (WPI, WPO) and Wrapper Interface Port (WIP). The testing control signal is provided by Wrapper Interface Port (WIP). Under the control of the WIP signal, test instructions are transferred to Wrapper Instruction Register (WIR) from WSI, test data also input from WSI and the test results are transferred to the test sink from WSO. Besides these ports, there are other two important registers: Wrapper Bypass Register (WBY) and Wrapper Boundary Register (WBR).

Chosen the ISCAS '89 reference sequence circuit S444 as the example, we will introduce the wrapper implementation in detail and describe the process of wrapper loading and test data transmission. The circuit S444 with the designed wrapper is shown in Figure 4.

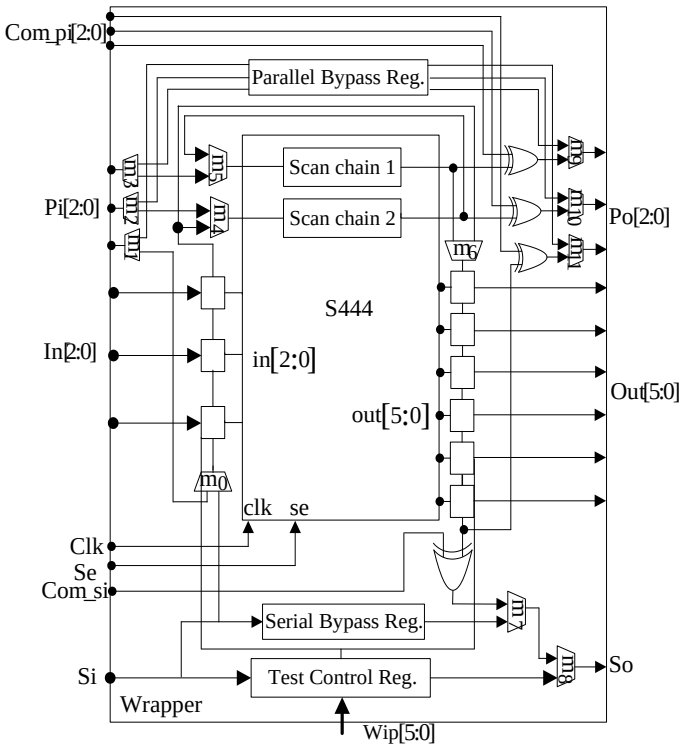


Figure 4. Circuit S444 with the designed wrapper

The small blank boxes in Figure 4 represent Wrapper Boundary Cells (WBC), *Si*, *So* are serial input/output signal, *Pi [2:0]*, *Po [2:0]* are parallel input/output port; *In [2:0]*, *Out [5:0]* are normal function input and output port; *Com_si* and *Com_pi[2:0]* are serial and parallel input port of correct test response for testing comparator. *Wip [5:0]* is the input signal of the test control register and the test control register contains the instruction register and update register. Its output signal is the 12-bit multiplexer enable signal (*m0~m11*) for the realization of the different test mode control.

After adding the test response comparator, the wrapper can locally verify whether there is fault in the IP core or not. The XOR gates in Figure 4 are the test response comparators and in conjunction with the comparators, there are the parallel comparison data input port (*Com_pi[2:0]*) and the serial comparison input port (*Com_si*) added in the wrapper. At the end of IP core testing, the correct test response is transferred through the *Com_si* or *Com_pi* into the wrapper and compared with the real test response by the XOR gates. If the IP is fault-free, the output of XOR gate will be all zero, otherwise not. The output of comparator will be sent to *Po* or *So* according to the chosen test mode and the test results are transmitted to NI for packing. The response packet has only a header flit and bit 13 to bit 0 will behalf the test result, which is full zero for fault-free. Because only one flit is sent to the NoC through the TAM, the burden of network is reduced and the transmission efficiency is improved.

Wrapper interface circuit diagram is shown in Figure 5, which mainly includes buffer module, control module and output module. Buffer module temporarily stores the test data on the influence of the control module, while control module determines the buffer data into Wrapper from WSI or WPI according to the input *Data_function* signal. Control module sends signal *Receive_ack* to the adapter according to the Wrapper operating condition and the signal *Receive_ack* means whether agree to receive data or not (high level represents agree).

Then control module decides the number of data packets into the buffer based the length information and at the same time generates control signal WIP following the principles of IEEE 1500. Output module is mainly used to receive test data from the wrapper, generate *Message_in* signal according to the *Data_mode* signal. When *Data_mode* is testing, if output data from WSO/WPO are all 0 (which means IP core is fault-free), *Message_in* signal will output all 0. In other cases, *Message_in* signal will output all 1. When the output data is ready, signal *Trans_req* will be transferred to get feedback signal. If *Trans_ack* is high level, *Message_in* will be sent into network adapter and at the same time test mode signal *Data_function* will be sent to pack the transmission data.

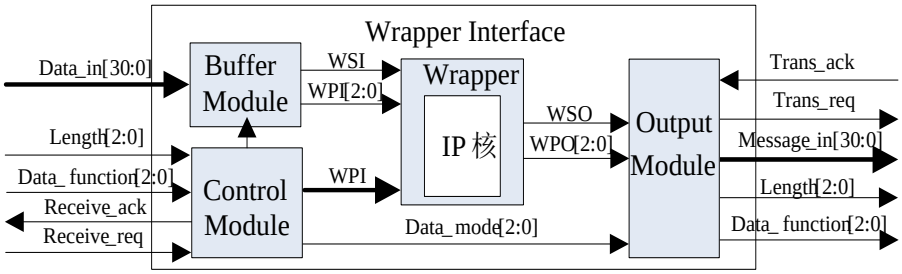
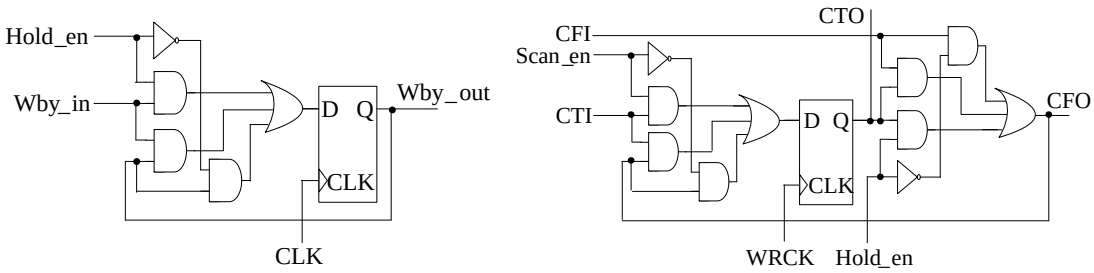


Figure 5. Architecture of wrapper interface circuit

The scan chain uses special trigger instead of ordinary trigger and the most commonly used is multiple choice trigger (Multiplexed_flip_flop). These scan triggers are linked to form a scan chain, which can work in normal or test mode. The length of inserted scan chains should keep consistent, for the difference of scan chains length may cause test time unnecessarily increase.

WBY provides a rapid transmission channel for data that no need to transfer through the IP core and it is one of the necessary components in the IEEE 1500 Wrapper. WBY in this paper consists of multiple selector and D flip-flop. When the data need to transfer through the WBY, Signal *Hold_en* is in high level and data is transmit from *Wby_in*, through D flip-flop and output to *Wby_out*. However, when no bypass transmission, *Hold_en* will be low level and flip-flop maintains the current value. WBY structure diagram is shown in Figure 6(a).



(a) WBY circuit (b) WBC circuit

Figure 6. Architecture of WBY and WBC circuit

WBC is the basic component of WBR and there should be a WBC in input/output port (not including clock, reset signal) of IP core for implementing observability and controllability. A WBC should include at least four part, those are Cell Function Input (CFI), Cell Function Output (CFO), Cell Test Input (CTI) and Cell Test Output (CTO). The functional data can be

transmitted to the CFO from CFI or transmitted to the CTO from CTI on the function of different control signal. The WBC can be inserted into multiplexer and connected together with other WBC to form a WBR. The realization of the WBC model is shown as Figure 6 (b).

When signal *Hold_en* and *Scan_en* are 0, function data transmits to CFO from CFI and then move into function port of the connected IP core; while *Hold_en* and *Scan_en* are 1, the test data transmits to CTO from CTI. The WBR can work in two test mode: internal and external test mode. For external test mode, both the function data and test data transfer from WBR, so set *Hold_en* to be 0, *Scan_en* be 1; for internal test mode, only test data transfer from WBR, so set *Hold_en* and *Scan_en* to be 1. In this paper the WBCs are inserted multiple selector and their CTI, CTO are connected end to end into the WBR, while different testing mode are realized by the control of each multiplexer.

The wrapper supports seven operation modes: normal function, serial bypass, parallel bypass, serial in-test, parallel in-test, serial ex-test and parallel ex-test. So instruction code needs three bits to encode which is named *Data_function [2-0]* in the wrapper interface. Instruction code transfers serially from WSI to WIR when the control signal *ShiftWR* is high level, while *ShiftWR* is low level, the code is hold in WIR. when *UpdataWR* signal is high level, the data in the WIR is decoded in the falling edge of the clock, generating 12-bit multi-channel selector enable signal (*m0~m11*) and WBR enable signal *Scan_en* and *Hold_en*, control the data transmission of multiplexer and WBR, so as to realize different testing mode.

WIR configuration flow diagram is shown in Figure 7.

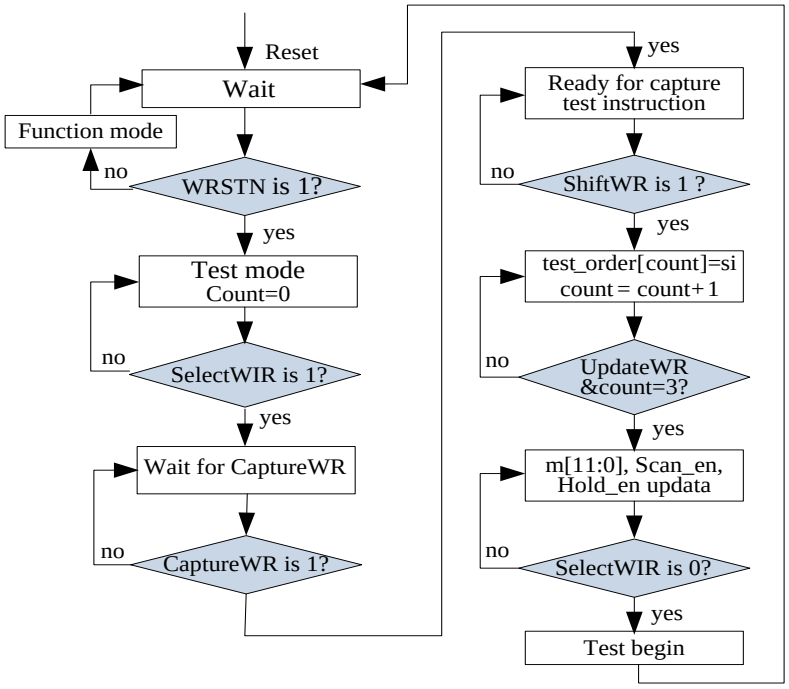


Figure 7. WIR configuration flow chart

If reset signal *WRSTN* is low level, the state will convert to normal function mode, and if the communication packets are function data, the *WRSTN* maintains low level, otherwise means test mode. In the test mode, high and low level of signal *SelectWIR* respectively means transferring test instruction or test data, *CaptureWR* in high level means the test instruction is

about to input and WIR is ready to receive instruction. *ShiftWR* in high level shows transferring test instruction status and WSI port will prepare test instruction to WIR. Three clock cycles later, test instruction will be sent into WIR and at the same time *ShiftWR* converts to low level. *UpdataWR* is high level means the decoding state and the test instruction is decoded on the falling edge of the clock, generating control signal for WBR and multiple selector. Then *SelectWIR* converts to low level and begins to transmit test stimulus. At the end of the test, *WRSTN* returns to low level, normal function resumes and waiting for the next test.

4. Experiment Results

The basic function of the wrapper is to accomplish correct testing on different function modes with the prescribed timing sequences. The simulation platform is Synopsys VCS tools. The test process for the wrapper is to set up each of function modes according to the standard WIP sequences and check the configuration of WBC and output results. For clarity, we only provide the parallel in-test simulation waveform as the example, which is shown in Figure 8. The signal sequence accords with the standard and the output is the expected result.

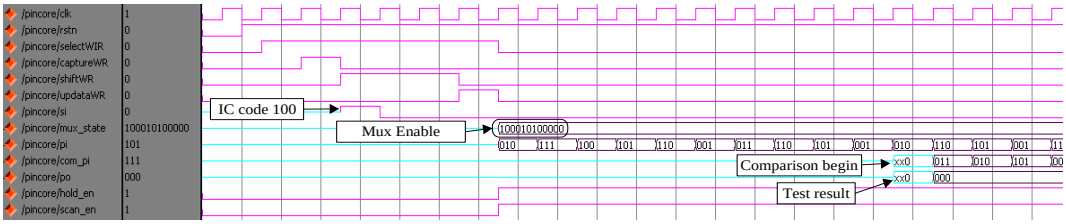


Figure 8. Wrapper simulation waveform (parallel in-test)

We evaluated the wrapper size overhead of four ISCAS '89 reference circuits which are applied as the IP cores and they are circuit S386, S444, S526 and S1238. The comparison between original size and the size with wrappers of four IP cores is shown in Figure 9. The size is increased by 1/3 on the average and the growth rate decreases with the increase of the original IP core size. Furthermore, considering the increase of the observability and controllability for embedded IP cores and the flexibility for various testing modes, the size overhead is acceptable.

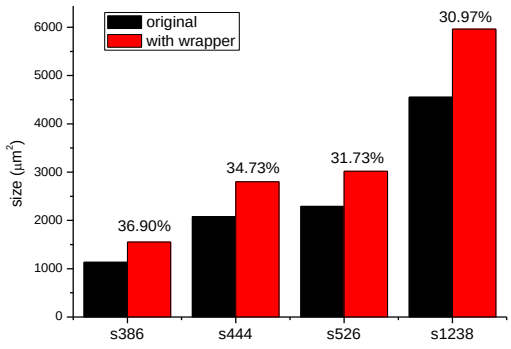


Figure 9. Comparison of original size and the size with wrapper

To evaluate the adaptability of wrapper, we will construct the under test NoC and apply the associated unicast and multicast testing. The circuit S386, S444, S526 and S1238 are adopted as the embedded IP cores and randomly inserted into the NoC structure. The NoC with inserted IP cores is shown in Figure 10.

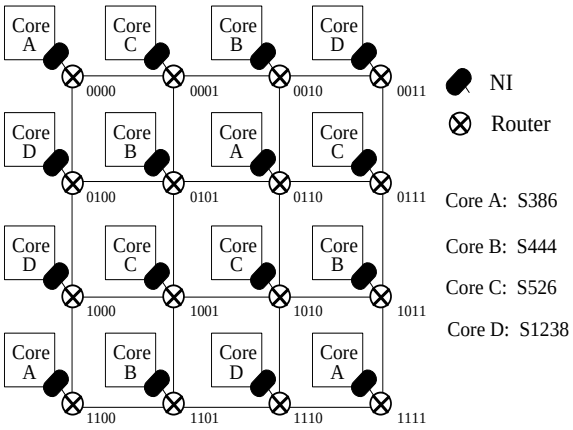


Figure 10. NoC structure with inserted IP cores

In order to compare with multicast testing, the unicast testing applies subnet to realize parallel test. The NoC can be dynamically divided into two, four or eight subnets according to the configuration. The external mapping module consists of three components: input module, output module and dynamic mapping module. Dynamic mapping module controls the subnet partition based on the corresponding block pattern. Block pattern encodes in 2 bits, 00 for no subset, 01 for eight subnets, 10 for four subnets, and 11 for two subnets. The subnet structure is shown in Figure 11.

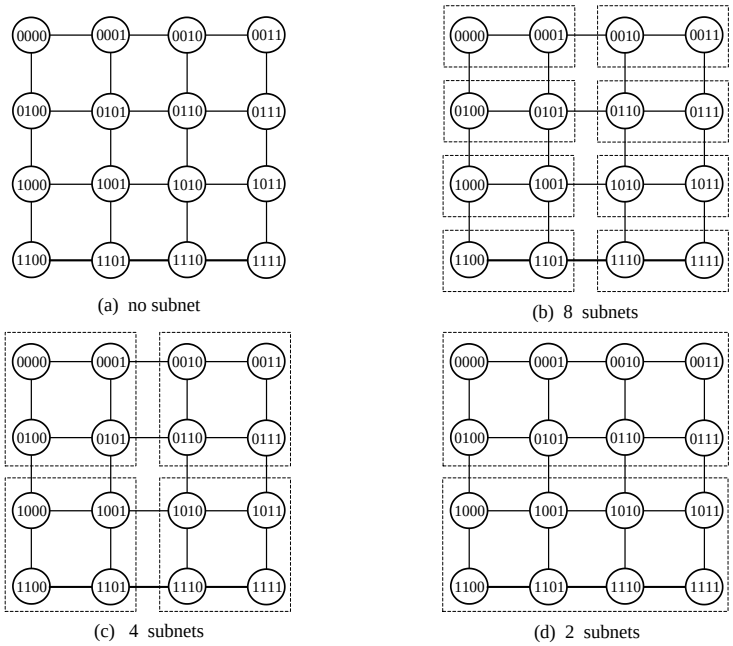


Figure 11. The structure of subnets partition

After the partition, the dynamic mapping module sends all subnet IP tables to the input module, which will control the sending sequence of testing data packets. At the same time, two 16-bit control signals are generated according to the block pattern and sent to input/output modules to connect the input/output modules with 16 boundary interfaces of the Mesh network.

The input/output modules directly connect to edge ports of Mesh structure and 16-bit enable signal from the dynamic mapping module provides the three states gating between peripheral circuit and router external port, so as to realize the interconnection of test input port and the router. The input module configures unicast packets according to the test address and sends test packets into the network. Reusing NoC edge ports avoids the special design of test input/output port and decreases the transmission delay caused by the transfer of test data in multiple peripheral circuits, thus eventually reduce the test time.

As far as the multicast testing is concerned, the data transmission between source node and destination nodes is based on XY routing algorithm. When the packet arrives the router, the address of current router will be checked with the last destination address in multicast nodes. If they are different, then continue to transmit the packet according to XY routing algorithm. If they are the same, the current node will get a copy of the packet and send to connected IP core. At the same time, the number of multicast destination nodes decrease by 1 and the packet will continue to transmit to the current last destination address node, until the packet arrives all the multicast destination nodes.

However, there is a key problem need to be solved for multicast testing, that is deadlock avoiding. As far as testing data transfer is concerned, we proposed the routing schemes as Figure 12 and Figure 13 shows.

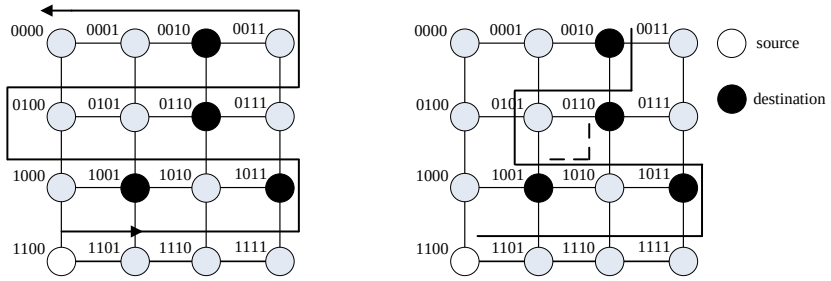


Figure 12. Sorting and transmission path when source node in four corners

When the source node belongs to one of the four corners, shown in Figure 12, 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 node. The destination node sequence for Figure 12 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 as shown in Figure 13.

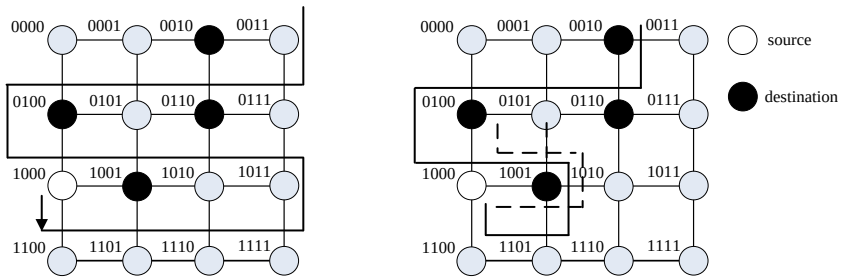


Figure 13. Sorting and transmission path when source node in four corners

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

(1) When data packets are sent to the last destination node in one row, if the next destination node of X direction below the line of the current node (as shown in Figure 12 destination node of 1001 is on the next column of 0110), the packet will send along the Y direction to next node;

(2) For source nodes not located in four corners of the router, packet is transmitted to nearest source node in the four corner routers;

(3) The line turning to the 180º angel is not allowed.

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

To be comparable with subnet model test, the numbers of multicast destination nodes are set to 2, 4, 8 and 16, while the destination addresses of multicast packets are the complete addresses of corresponding subnet. Moreover, the test data of multicast mode and subnet mode are the same to guarantee the undifferentiated testing condition. We gave the comparison of test data transmission power and time for unicast and multicast testing in the text follows.

Test data transmission power

The transmission power of NoC test data packet is mainly related to the number of test data packets and the length of the transmission path [17]. Assumed that the total number of under test IP core is *n*, number of test data flits for IP core i is *Ci*, N_{Ri} is the total number of router in testing data transmission path, N_{Ci} is the channel number in the same path and there is $N_{Ri} = N_{Ci} + 1$. T_R, T_C is respectively the consumption power when a test data flit transmits through a router or a channel. Total power transmission of testing all IP cores can be calculated based on Eq. (1).

$$\begin{aligned} T_{all} &= \sum_{i=1}^{n} C_i \times (N_{Ri} \times T_R + N_{Ci} \times T_C) \\ &= \sum_{i=1}^{n} C_i \times [(N_{Ci} + 1) \times T_R + N_{Ci} \times T_C] \\ &= \sum_{i=1}^{n} C_i \times [(T_R + T_C) \times N_{Ci} + T_R] \end{aligned} \quad (1)$$

In Eq. (1), T_R and T_C are both constants, so the test transmission power T_{all} can be measured by $\sum_{i=1}^{n} C_i \times N_{Ci}$, in which N_{Ci} is equal to hop count of each test packet and that is the number of passing routers from the source IP core to the destination IP core. Therefore, the total number of hop counts can measure power overhead on test data packets transmission.

For unicast testing, scheme a, b, c, d respectively means the NoC is divided into 8 subnets, 4 subnets, 2 subnets and no partition pattern; for multicast testing, scheme a, b, c, d means broadcast packets with 2 addresses, 4 addresses, 8 addresses and 16 addresses. Total hop counts and used port numbers of unicast and multicast mode with 4 schemes are shown in Table 1.

Table 1. Comparison of hop count and port number

Scheme	*Hop count*		*Port number*	
	Subnet	Multicast	Subnet	Multicast
a	16	16	12	12
b	32	28	8	8
c	64	52	4	4
d	96	63	2	2

Based on the data in Table 1, the hop counts of scheme a to scheme d gradually increase, while the port numbers decrease. That means the test transmission power of scheme a is minimal and the port overhead of scheme d is minimal. And it is obvious that the power consumption of multicast mode is much lower than unicast mode.

Testing time

Based on the designed wrapper, the testing time of subnet and multicast modes can be evaluated on Synopsys platform. Applying four schemes of two testing modes on the NoC as Figure 10, test time results are shown in Figure 14.

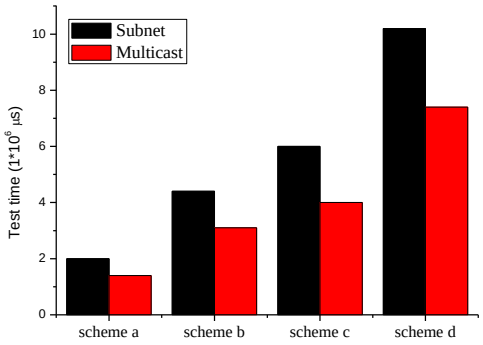


Figure 14. Comparison of test time

It can be concluded from the experiment results that unicast mode is more time-consuming than the multicast mode. Although the unicast test (subnets) as a whole is parallel, the data transmission in each subnet is still serial. On the other side, multicast testing for each IP core is really concurrent.

5. Conclusion

This article designed an improved IEEE 1500 compatible wrapper for NoC embedded IP core, which is available for unicast and multicast testing. The particular test response comparator in the wrapper can directly confirm whether faults occur in the IP core or not, so that it will greatly improve the testing efficiency and flexibility. Unicast and multicast testing of embedded IP cores on 2D Mesh NoC are implemented based on the proposed wrapper and the test power consumption and test time are evaluated. The experiment results indicate that the wrapper is perfectly functional and has good adaptability. How to further enhance parallelism is the goal for wrapper optimization and we will continue the research on this respect in future.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (No. 61106018 and No. 61106029) and the Defense Industrial Technology Development Program under contract No. B2520110008.

References

[1] S. Lu, X. Yan, H. Li, H. Shen and L. He, "Test Wrapper Design for IP-core Test Reuse", Journal of Zhejiang University (Engineering Science), vol. 38, no. 4, **(2004),** pp. 93-97.
[2] D. Son, H. Choi, H. Jeon and C. Kim, "Thermal-aware 3D Multi-core Processor Design using Core and Level-2 Cache Placement", International Journal of Control and Automation, vol. 6, no. 1, **(2013)**, pp. 25-31.
[3] IEEE 1500 Embedded Core Test, http://grouper.ieee.org/groups/1500/, **(2005)**
[4] A. M. Amory, K. Goossens, E. J. Marinissen, M. S. Lubaszewski and F. G. Moraes, "Wrapper Design for the Reuse of Networks-on-Chip as Test Access Mechanism", Proceedings of the Eleventh IEEE European Test Symposium, **(2006)** May 21-24, Southampton, United Kingdom.
[5] A. M. Amory, K. Goossens, E. J. Marinissen, M. S. Lubaszewski and F. G. Moraes, "Wrapper Design for the Reuse of a Bus, Network-on-chip, or Other Functional Interconnect as Test Access Mechanism", IET Computers & Digital Techniques, vol. 1, no. 3, **(2007)**, pp. 197-206.
[6] F. A. Hussin, T. Yoneda and H. Fujiwara, "Optimization of NoC Wrapper Design under Bandwidth and Test Time Constraints", Proceedings of the 12th IEEE European Test Symposium, **(2007)** May 20-24, Freiburg, Germany.
[7] F. A. Hussin, T. Yoneda and H. Fujiwara, "NoC-Compatible Wrapper Design and Optimization under Channel-Bandwidth and Test-Time Constraints", IEICE Transactions on Information and Systems, vol. E91-D, no. 7, **(2008)**, pp. 2008-2017.
[8] B. Aghaei and S. Babaei, "The New Test Wrapper Design for Core Testing in Packet-switched Micro-network on Chip", Proceedings of the 2nd International Conference on Power Electronics and Intelligent Transportation System (PEITS), **(2009)** December 19-20, Shenzhen, China.
[9] K. Goossens, J. Dielissen and A. Radulescu, "AEthereal Network on Chip-concepts, Architectures, and Implementations", IEEE Design & Test of Computers, vol. 22, no. 5, **(2005)**, pp. 414-421.
[10] Z. Lu, B. Yin and A. Jantsch, "Connection-oriented multicasting in wormhole-switched networks on chip", Proceedings of IEEE Computer Society Annual Symposium on VLSI, **(2006)** March 2-3, Karlsruhe, Germany.
[11] F. A. Samman, T. Hollstein and M. Glesner, "Multicast parallel pipeline router architecture for network-on-chip", Proceedings of Design, Automation and Test in Europe, **(2008)** March 10-14, Munich, Germany.
[12] L. Benini and G. De Micheli, "Networks on Chips: A New SoC Paradigm", IEEE Computer, vol. 35, no. 1, **(2002)**, pp. 70-80.
[13] Y. Choi and N. Park, "Multicore and Mesh Network-based Parallel Performance Evaluation using Intra Prediction Algorithms", International Journal of Control and Automation, vol. 5, no. 4, **(2012)**, pp. 49-54.
[14] W. Dally and C. Seitz, "Deadlock-free Message Routing in Multiprocessor Interconnection Networks", IEEE Transactions on Computers, vol. 36, no. 5, **(1987)**, pp. 547–553.
[15] H. Liu, "Key Techniques of Multicast Communication for Network on Chip", Master thesis, Wuhan University of Technology, **(2011)**

[16] C. Grecu, P. Pande, B. Wang, A. Ivanov and R. Saleh, "Methodologies and Algorithms for Testing Switch-based NoC Interconnects", Proceedings of the 20th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems, **(2005)** October 3-5, Monterey, U.S.A.
[17] E. Cota, L. Carro, F. Wagner and M. Lubaszewski, "Power-aware Noc Reuse on the Testing of Core-based Systems", Proceedings of the International Test Conference, **(2003)** September 30-October 2, Charlotte, U.S.A.

