# Collaborative Verification and Testing Platform for NoC-based SoC

Ying Zhang, *Member, IEEE,* Ning Wu, Fen Ge and Xin Chen

***Abstract*—Verification and testing play important roles in complex system chip design, especially for NoC-based SoC, which has been increasingly applied to complex system design. This paper proposes a collaborative verification and testing platform for NoC-based SoC. Based on VMM verification methodology, a hierarchical NoC validation platform is constructed and assigned to the function verification of NoC communication. Simulations on Mesh and Torus NoCs validated the satisfying function coverage of the software verification platform. Besides, a reconfigurable NoC testing platform is implemented based on PowerPC and FPGA. Some experiments of the testing platform are performed on NoCs with ITC'02 benchmark circuits. The experiment results showed that the testing platform established the hardware verification and capable of evaluating testing time and other parameters. The function verification information can be transmitted to the testing platform by certain interfaces to reconstruct NoC, meanwhile the hardware simulation results can be acquired by the software platform to improve the NoC design. So that, the collaborative hardware and software verification is accomplished.**

***Index Terms*—verification, testing platform, NoC, co-verification, VMM**

## I. Introduction

WITH the increasing development of the semiconductor technology and chip integration, NoC (Network on Chip) becomes an important solution to complex SoC (System on Chip) architecture[1]. In the mean time, the corresponding validation and testing assignments face unprecedented challenges. According to statistics, design verification and testing are expected to account for more than 70% of the whole design work for complex system chips [2]. The verification and testing platform for NoC will accomplish the function verification and performance evaluation efficiently. Therefore, how to construct the verification and testing platform has great significance for NoC research and application.

There have been some research [3-5] discussing on the design of verification platforms for NoC-based SoC. Authors of [3] proposed an automated verification environment for NoC. The simulation results on Spidergon-STNoC with Cadence Specman are satisfying, but the verification platform only focuses on function verification. S. Chai *et al.* [4] proposed a NoC simulation and verification platform based on SystemC. Simulated by loading different network traffic patterns, the platform has been applied to verification of NoC architecture and routing algorithm. Also, the platform does not involve the hardware verification. T. Huang et al. [5] introduced a verification platform for NoC with 16 microprocessors based on FPGA. 4×4 Mesh topology and wormhole routing mechanism are applied to the verification platform and an OFDM transceiver is mapped on the platform. The platform can be effective to hardware verification, while its scalability is limited and size overhead is relatively high. Our proposed solution is to take into account the needs of the hardware and software verification at the same time, which will greatly decrease the comprehensive testing time and increase the testing efficiency.

We propose a software/hardware co-verification platform for NoC-based SoC. Based on SystemVerilog and VMM methodology, a hierarchical verification platform is designed to confirm the functions of NoC communication. At the same time, a reconfigurable NoC testing platform is implemented based on PowerPC and FPGA. The verification information will transfer between software platform and hardware platform, in order to modify the NoC mapped to FPGA and improve the system design, thus forming the effective co-verification of NoC-based SoC.

Hereafter, Section II gives a brief introduction of the software/hardware co-verification platform for NoC. Section III presents the design of collaborative verification and testing platform. Section IV explains the experiment results on NoCs constructed with ITC'02 benchmark circuits. Section V draws the conclusions.

## II. Hardware/Software Co-verification Platform for Noc-based SoC

The collaborative verification and testing platform proposed in this paper is applied to NoC-based SoCs. NoCs can be defined as a set of structured routers and point-to-point interconnecting IP cores (Resources). The topology of NoC can be represented as an undirected connected graph $G(N,L)$, where $N=\{n_1,n_2,...n_i\}$ is the set of nodes and $L=\{l_1,l_2, ...l_j\}$ is the set of links in the corresponding network. The widely applied topologies are Mesh, Torus, Ring, Hypercube, and Fat-tree [6]. For regularity of its structure, Mesh network are easy to

Manuscript received July 10, 2015; revised August 4, 2015. This work was supported in part by the National Natural Science Foundation of China (61376025), the Aeronautical Science Foundation of China under Grant (20140652008), and Prospective Joint Research Project of on the Integration of Industry, Education and Research of Jiangsu Province (2014003-05).

Ying Zhang is with the College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, China (e-mail: tracy403@nuaa.edu.cn).

Ning Wu, Fen Ge and Xin Chen are with the same place of Ying Zhang. (e-mail:wunee@nuaa.edu.cn,gefen@nuaa.edu.cn, xin_chen@nuaa.edu.cn).

implement and have good scalability. Each node in Mesh network is connected to neighbors through regular grid point-to-point links.

The architecture of software/hardware co-verification platform for NoC-based SoC is shown in Fig. 1. Software verification platform is meant to accomplish the validation process by certain software tools and construct the environment with different abstraction levels. Furthermore, the verification platform is aimed to ensure correct function and satisfying performance parameters. Hardware testing platform is based on the configurability of FPGA to realize modules, structure and system level hardware verification of the NuT (NoC under Test). The reusability of hardware testing components will effectively improve adaptability and expandability. Through the Ethernet or serial interface, information can share or exchange between software and hardware platform, so as to realize the collaborative verification and testing.

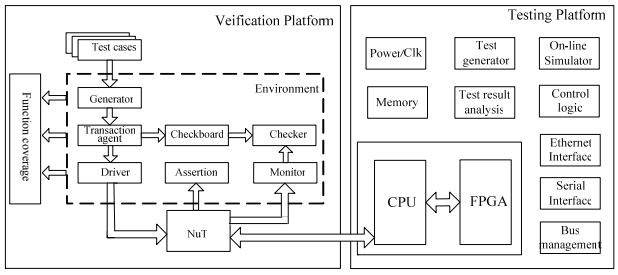



Fig. 1. Diagram of the software/hardware co-verification platform

The software validation environment in Fig. 1 provides the transaction level based on VMM verification method and the RTL level verification based on SystemVerilog or Verilog, etc. The input files of software verification platform include the test data files and hardware description files of NuT. After acquiring data from the files, the platform will come into effect according to the predetermined validation process. Validation results will be compared with ideal results in the test files and comparison results will be in the form of assertions in the output of the verification platform. If the results do not agree with the ideal results, which means design errors exist, information will be sent to the test files and design modules. And there will be the corresponding revise and resolution later. If the results of the hardware testing platform show the need to modify the structure design, the analysis report will be transmitted to the verification platform by the interfaces between two platforms.

The hardware testing platform in Fig. 1 is based on microcontroller and FPGA and under test NoC is mapped to the FPGA. The function and performance of the system chip can be tested through JTAG or Chipscope tool and test results will compare with the ideal responses. The comparison results will be analyzed on-line or transmitted to the PC and the analysis results will feedback to software platform, or remap directly to the hardware platform after modification.

The verification and testing platform for NoC-based SoC has the following features:

1. Parallel implementation of verification and testing.

As long as respectively obtained the required input, verification and testing platform can implement independent verification and testing process. So that parallel application of verification and testing can be realized in theory and in practice.

2. Information sharing between verification and testing.

Errors found by verification and testing platform will be easily notified to each other. Furthermore, the information can be used as the reference of the other. When there is an error in any one of platforms, verification or testing process will be retriggered.

3. Hierarchical and reusable feature of validation and testing methods.

Verification platform is built based on the VMM verification methodology and has the characteristics of the hierarchical and reusability. Testing platform consists of modular components and test structure can be online reconfigured to provide reusable characteristics.

The design process of proposed collaborative verification and testing platform is shown in Fig. 2.

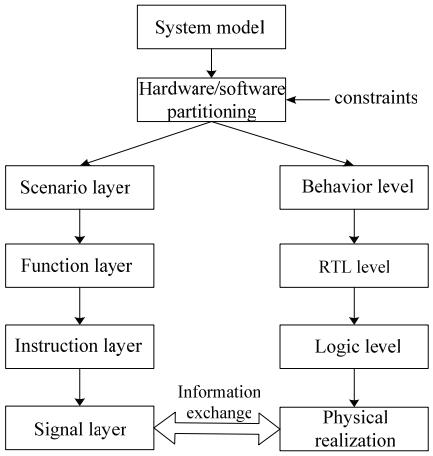



Fig. 2. Design process of the collaborative platform

First of all, according to the requirement of the validation plan and system testing parameters, validation and testing behavior is divided into hardware and software verification objective. Software platform adopts the VMM methodology to build hierarchical structure and consist of testing layer, scenario layer, function layer, command layer, and signal layer. Each level provides a series of services to the upper level hiding low-level details through the abstract processing. Test cases are usually located in the top of the verification platform, but in the practical application, they can bypass other levels or interact with any component in order to achieve direct access to under test parts. The flexibility of test cases and testing configuration makes it easy to implement the design of software verification platform. In the hardware platform, the behavior level, RTL level and logic level of under test NoC can be mapped to the FPGA, cooperating with the microcontroller and basic peripheral interfaces provided by IP library, along with testing controller, memory modules, etc. In that way, an efficient hardware testing platform can be configured. Through appropriate modification and module replacement, NoC can be easily remapped on the software and hardware verification platform.

ISBN: 978-988-19253-6-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

## III. Design of Collaborative Verification and Testing Platform for NoC-based SoC

The hierarchical NoC verification platform is based on the VMM methodology, which adopts assertions, abstraction, automation, reuse mechanism in order to improve validation and productivity [10]. The platform is composed of multiple levels of description and can react to each layer in the whole design process. Each level can provide services to its upper level or test cases, and hide low-level details by means of abstraction. Different Layers are neatly partitioned and the flow transmission is achieved through channel mechanism.

The verification platform contains multiple-layered test modules and controls four steps of the test process, which includes generating random test vectors, establishing verification environment, executing verification and producing the reports. The basic structure of the platform is showed as Fig. 3, which consists of test cases, test vector generator, test environment and NoC under test. The environment includes driver module, monitor module, checker module, scoreboard module and function coverage module.

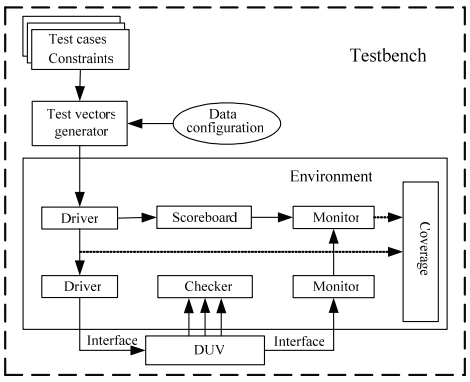



Fig. 3. Basic structure of the hierarchical NoC testbench

According to the multi-layered infrastructure, the verification platform can be divided into Test layer, Scenario layer, Function layer, Instruction layer and NoC under Test.

Test layer provides test cases and the realization of test cases is the combination of the following elements: additional constraints of generator, new definition of random scenes (scripts), synchronization mechanism between transaction processors, state monitor and directed stimulation for the original design. Scenario layer provides generators of data and transactions which can be controlled and synchronized. Function layer contains high-level driver and monitor modules and can confirm verification results. Instruction layer is related to various interfaces and physical protocols of NuT. It provides a consistent and low-level interface which is not relevant to the modeling manner. The RTL model is connected to the upper level module through the interface. An interface can be regarded as a bundle of intelligent connections, which contain connection, synchronization, communication between two or more modules.

Each module of the verification platform is realized in the form of certain class, the UML class diagram to explain their relationship is shown as Fig. 4.

The *Environment* class is the core of the verification platform. 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. 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.

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

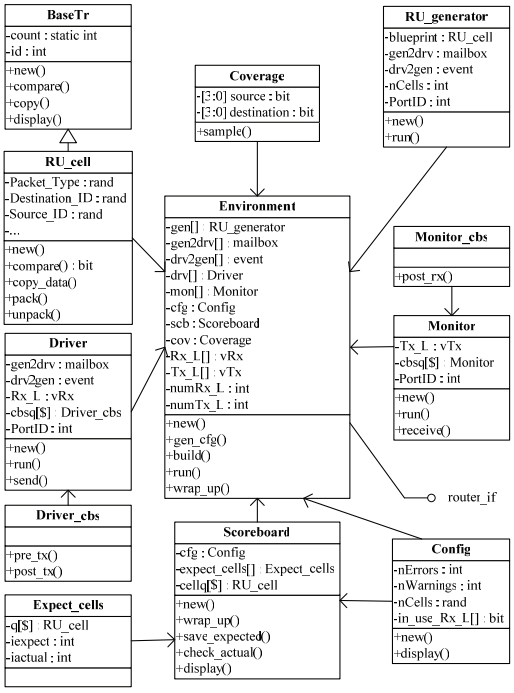



Fig. 4. UML class diagram of NoC verification platform

In order to simulate system chip in module level and circuit level, we proposes a reusable multi-level testing platform for NoC-based SoC. The testing platform structure is shown as Fig. 5.

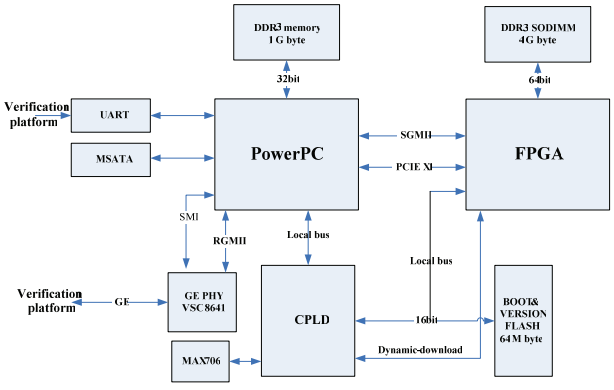



Fig. 5. Architecture of NoC testing platform

ISBN: 978-988-19253-6-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Acquiring function verification information from software platform, testing platform is constructed based on configurable interconnection modules. Moreover, test generation and test data analysis module are designed to support offline and online testing. The hardware platform not only can be applied to NoC performance evaluation, but also can implement hardware validation for testing structure and testing strategy.

PowerPC (CPU) is responsible for interfaces management and controlling data exchange between the PC and the FPGA on the testing platform. Test stimulus and response data can be stored in the DDR or SATA and the FPGA configuration can be accomplished through the JTAG input or online by the PROM. Hardware configuration and test process for NoC is shown in Fig. 6. Firstly, according to specific application, to the size, topology and routing rules of the NoC will be determined. After that, the NoC mapping can be realized with the corresponding test structure and traffic status. Then, configuration information will be provided to generate reconfigurable routers and corresponding configuration file. Test stimulus and ideal response data are generated by the software on PC and transmitted to the testing platform. Finally, test response data can be stored in the chip or transmitted to the PC and test results can be analyzed online or done post-process by the software on the PC.

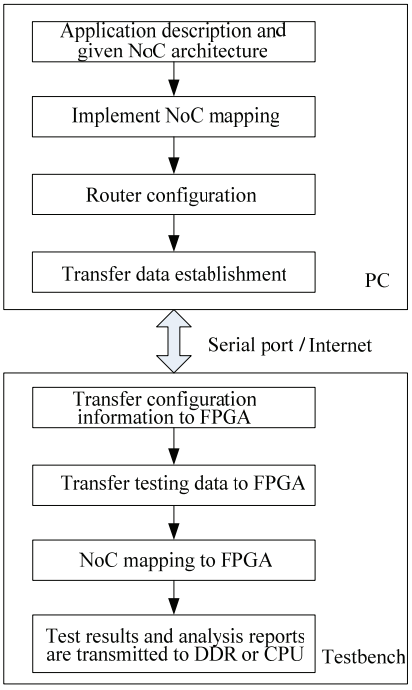



Fig. 6. Flow chart of hardware configuration and testing

Dynamic reconfiguration routing module is one of the main components on testing platform and it is composed of Traffic Generator (TG), Traffic Receiver (TR), Network Interface (NI) and the reconfigurable router. Fig. 7 describes the main function modules of FPGA and interface modules on test platform. It adopts g1023 circuit of ITC'02 benchmark [8] to map on 4×4 Mesh as an example. Test data are transmitted into the Mesh network by the PowerPC or DDR module, and will be forwarded to another router or local IP core according to the routing algorithm, while each IP core being with the IEEE 1500 compatible test wrapper supporting multi testing modes [9]. NoC mapping is realized based on collaborative optimization of testing and mapping for NoC [10], and the NoC mapping in Fig. 7 is on the condition of W = 16, $\alpha = \beta = 0.5$ . TAS (Test Access Switch) in Fig. 7 means the testing data input/output port. To achieve the systemic reusability of the testing platform, the interconnection mechanism with adaptable routing algorithm and the construction of online configuration mode are required in testing platform. The structure of reconfigurable network interface and other modules will be demonstrated in detail later.

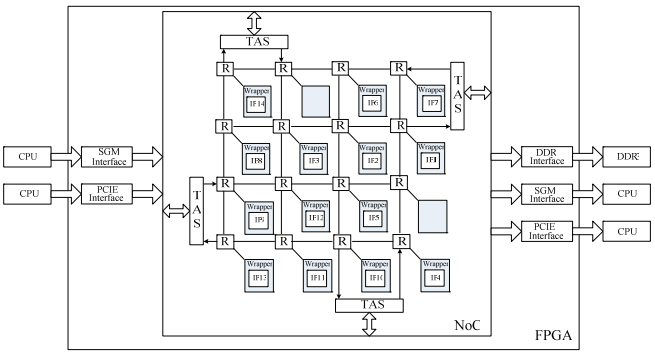



Fig. 7. Diagram of internal modules in FPGA (W = 16, g1023)

● Network interface module

Network interface (NI) is the bridge between local synchronization system and NoC data transmission channel. Its main function is to provide network adapter services for IP cores, similar to the modem of computer network. It implements the separation of the communication and computing of NoC and this makes the IP core selection more concentrated to the function realization instead of considering the communication architecture. The upper module of NI is the Core Interface (CI), while the lower connection module is routing interface module. Besides realizes the network protocol conversion between different levels, NI can also feedback part of the state information to the local IP Core. So that, dynamic configuration will implement and obtain better performance and functional flexibility. The structure of network interface circuit is shown as Fig. 8.

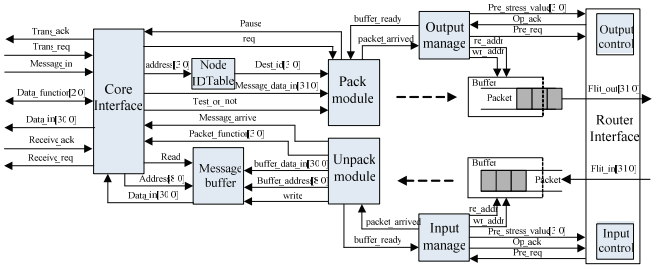



Fig. 8. Architecture of network interface module

NI is mainly composed of routing interface, data buffer, input/output channel management, pack/unpack module, etc. Routing module is applied to transform between different clock domains, transfer order, end-to-end flow control, etc. Input/output management modules acknowledge to the transmission network according to the buffer situation, the modules are mostly reconfigurable to support various communication architectures.

ISBN: 978-988-19253-6-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

● Configuration module

Configuration information is in a binary file generated by the PC, which contains three types of information: information of interconnection network (nodes, interconnect, network structure), information of the router structure (reconfigurable options), information of test data (TAS access, test data packet format, transmission bandwidth). The configuration file are transferred to CPLD from the CPU and stored in the Block Ram and registers of the CPLD. The disassembly module of CPLD will translate binary data in configuration file into corresponding control information of NoC and modify them into synthesizable RTL codes, finally downloaded to the FPGA. The process of configuration related modules structure is shown in Fig. 9.

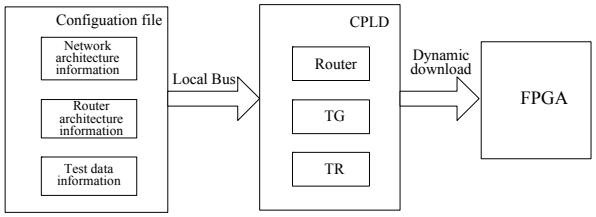



Fig. 9. Diagram of configuration module

● Ethernet module

Diagram of Ethernet data transmission processing is shown in Fig. 10. Ethernet module implements the data communication between the FPGA chip and Ethernet physical layer (PHY). Data in the FPGA chip will packet and output through the PHY port and the data sent to PHY will be filtered and unpacked for transmition to the FPGA. Ethernet module mainly includes MAC IP core and UDP/IP protocol stack.

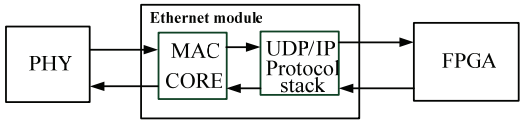



Fig. 10. Diagram of Ethernet data transmission

MAC IP core module includes client FIFO, AXI4-Lite state control machine and the physical layer interface module.

UDP/IP protocol stack module implements the UDP, IPv4, ARP protocol. The connection delay of the MAC layer control module is zero. The source port and destination port of sending UDP packets are completely controllable. When data are sent, they are allowed to be filtered through the source port to the destination port. Sending and receiving data from the IP layer broadcast address are both supported. The sending and receiving clock domain are independent of each other. The function of protocol stack has been verified in the 1 GBPS Ethernet field.

## IV. Performance Evaluation of Collaborative Verification and Testing Platform

In order to evaluate the proposed collaborative verification and testing platform, some experiments are applied to NoCs with circuits of ITC'02 benchmark. Firstly, the simulation of verification platform is provided. The simulation on Synopsys VCS with the 4×4 Torus NoC is shown as Fig. 11. In addition, the simulation on ModelSim with 4×4 Mesh NoC is shown as Fig.12. The results in Fig.11 and Fig.12 show that the data transmission among different IP cores is consistent with the preset data, which verifies the correctness of the data transmission function of the NoC. The software platform can be easily adapted to other topology and dimension NoCs.

```
Router[0]: exp=198, act=198
Router[1]: exp=192, act=192
Router[2]: exp=200, act=200
Router[3]: exp=191, act=191
Router[4]: exp=191, act=191
Router[5]: exp=191, act=191
Router[6]: exp=191, act=191
Router[7]: exp=193, act=193
Router[8]: exp=200, act=200
Router[9]: exp=192, act=192
Router[10]: exp=184, act=184
Router[11]: exp=197, act=197
Router[12]: exp=186, act=186
Router[13]: exp=201, act=201
Router[14]: exp=189, act=189
Router[15]: exp=195, act=195

total_time: 1865680 ns
actual cells received: 3091
Average latency : 60 ns
last_time: 51305 ns
Throughput*100 (flit/cycle): 48.000000
$finish at simulation time                55825
           V C S   S i m u l a t i o n   R e p o r t
Time: 55825 ns
CPU Time:      5.450 seconds;       Data structure size:   1.2Mb
Fri Aug  1 10:29:47 2014
```

Fig. 11. Veification platform simulation results on VCS

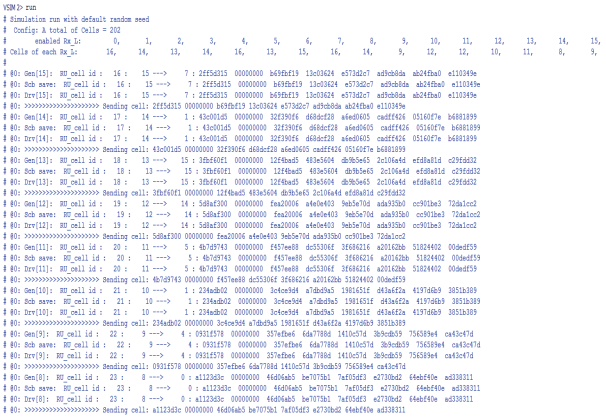


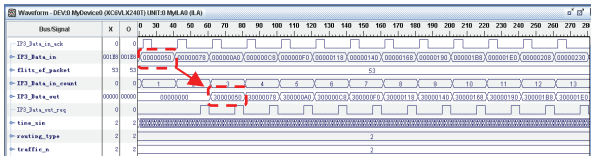
Fig. 12. Veification platform simulation results on ModelSim

Choosing g1023 circuit as the example and setting the system clock to 100 MHZ, test optimization mapping result of testing platform is shown in Fig. 7. The test data of IP3 are output from FPGA to PowerPC through SGMII or PCIE, then transferred to the PC by serial or Ethernet ports, and the data can also be stored in DDR3 of the testing platform. The test data transmission process is shown in Fig. 13. Because the capture range of Xilinx ISE Chipscope is limited, only one segment of the test response data is provided.

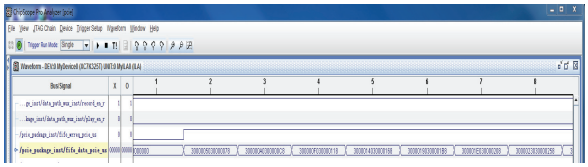
Since the IP core in testing platform is not concrete logic circuit, the experiment applies the preset test data, for example, the input data for IP3 is 00000050, the corresponding output data for 30000050, which can be seen in Fig 13 (a). Figure 13 (b) (c) show the corresponding test data of the PCIE and Ethernet interface, later the data will be transmitted to the PC, which means to be sent to the verification platform. The same test data can be stored in the DDR, which are available for online test analysis, as Figure 5.21 (d) shows. Note that the word width of the PCIE,

ISBN: 978-988-19253-6-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

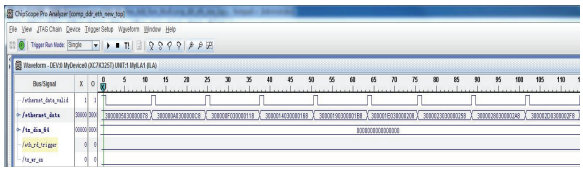
Ethernet, DDR is 64bit, while the original test data is 32bit.



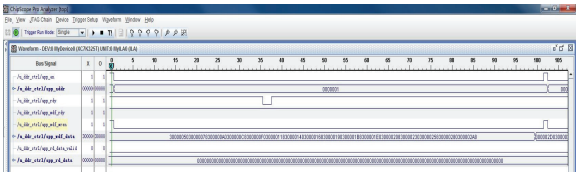
a) The test data of IP3 in the FPGA



b) Output of test data through PCIE



c) Output of test data through Ethernet



d) Test data is output to DDR3

Fig. 13. Test data transmission process

The testing optimized NoC mapping for g1023 circuit are applied to different optimization proportions under testing platform and comparison results of test time is shown in TABLE I. (the time unit in TABLE I is the number of clock cycles).

TABLE I
THE TEST TIME OF TEST PLATFORM

| Optimization proportion | | Tc_1 | Tc_2 | Tc_3 | Tc_4 | Tmax |
|---|---|---|---|---|---|---|
| W=16 | $\alpha=\beta=0.5$ | 7405 | 16863 | 15597 | 15003 | 15597 |
| | $\alpha=1,\beta=0$ | 14570 | 11275 | 14923 | 14108 | 14923 |
| | $\alpha=0.8,\beta=0.2$ | 13727 | 15003 | 10990 | 15148 | 15148 |
| W=32 | $\alpha=\beta=0.5$ | 15597 | 12444 | 6369 | 15338 | 15597 |
| | $\alpha=1,\beta=0$ | 12721 | 10875 | 14916 | 11237 | 14916 |
| | $\alpha=0.8,\beta=0.2$ | 12149 | 14989 | 11057 | 11553 | 14989 |

The result in TABLE I show that test time optimization results of hardware emulation are consistent with theoretical algorithm analysis [10], which not only verifies the validity of the testing optimized NoC mapping algorithm, but also showed the effectiveness of the collaborative software and hardware platform.

Fig. 14 shows the photo of the hardware platform. The IC under the heat sink of bottom right is Freescale PowerPC P1010 and under heat sink of upper right (beside DDR slots) is Xilinx FPGA ffg900 XC7k325T-2.

Fig. 14. Photo of the test platform

## V. Conclusion

This paper presents the design of the software and hardware collaborative verification platform for the NoC. Based on SystemVerilog and VMM methodology, a multi-level NoC verification platform is constructed. The software platform can validate the NoC communication function and easily adaptable to various network structures. With reconfigurable router module and RTL model of the communication structure, NoCs are mapped on the FPGA of testing platform. The experiments on NoCs with ITC '02 benchmark circuits indicated that the verification and testing platform can effectively realize the on-chip NoC parallel testing and can accomplish dynamic configuration. Data exchange between the testing platform and the verification platform is implemented by the Ethernet and serial interfaces, so as to realizing the software and hardware co-verification. Moreover, the collaborative verification and testing platform can be applied to analysis function and testing parameters.

## References

[1] T. Bjerregaard and S.A. Mahadevan. “Survey of Research and Practices of Network-on-Chip,” *ACM Computing Surveys*, vol. 38, no. 1, 2006, pp. 1–51.
[2] Chris Spear, SystemVerilog for Verification, Synopsys,Inc. 2006.
[3] F. Vitullo, S. Saponara, E. Petri, et al. “A reusable coverage-driven verification environment for Network-on-Chip communication in embedded system platforms,” in *Proc. 7th IEEE Workshop on Intelligent solutions in Embedded Systems*, Ancona, 2009, pp. 71–77.
[4] S. Chai, C. Wu, Y. Li, et al. “A NoC simulation and verification platform based on SystemC,” in *Proc. 2008 Computer Science and Software Engineering Conf.*, Wuhan, 2008, pp. 423–426.
[5] T. Huang, Y. Chen, J. Hu, et al. “Network-on-chip emulator design with FPGA array,” in *Proc. 2010 ICCCAS Conf.*, Chengdu, 2010, pp. 886–890.
[6] Tayan, O. “Networks-on-Chip: Challenges, trends and mechanisms for enhancements,” in *Proc. ICICT'09 Conf. Information and Communication Technologies*, Karachi, 2009, pp. 57–62.
[7] J. Bergeron, E. Cerny, A. Hunter and A. Nightingale, “Verification Methodology Manual for SystemVerilog”, Synopsys & ARM, Inc. 2006.
[8] E.J. Marinissen, V. Iyengar and K. Chakrabart
[9] y. “A Set of Benchmarks for Modular Testing of SOCs,” in *Proc. ITC'02 Conf.*, 2002, pp. 519–528.
[10] Y. Zhang, N. Wu, F. Ge and L. Zhou. “Novel Core Test Wrapper Design Supporting Multi-mode Testing of NoC-based SoC,” *IJCA*, vol. 6, no. 5, 2013, pp. 247–262.
[11] Y. Zhang, N. Wu, F. Ge and X. Chen. “Collaborative Optimization of Testing and Mapping for Network on Chip,” *IEEE Trans. On CAD/ICS(TCAD),* submitted for publication.

ISBN: 978-988-19253-6-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)