ELECTRONICS AND ELECTRICAL ENGINEERING
ISSN 1392 – 1215 *2007. No. 3(75)*
ELEKTRONIKA IR ELEKTROTECHNIKA

SIGNAL TECHNOLOGY
T121
SIGNALŲ TECHNOLOGIJA

NoCs Design for Verification

V. Hahanov, O. Yegorov, K. Mostova
Design automation department, Kharkov National University of Radio Electronics,
Lenin av., 14, Kharkov, 61166, Ukraine, tel.: 38 (0 57)702-13-26, e-mail: hahanov@kture.kharkov.ua,
sasha_egorov@kture.kharkov.ua, mostovaya@kture.kharkov.ua

Introduction

In recent time in high-performance computer systems there is a growth of the number of the embedded processors in systems and networks on chip (SoC, NoC). Common bus architecture for the providing interconnection of IP blocks is starting to be not enough sufficient, in spite of architecture simplicity, acceptable price and architecture extendibility. There are appearing some complications that make further usage of such architectures almost impossible. Capacitance for the buses with big length is becoming critically high. Moreover, with increasing number of IP the propagation delay becomes longer, that influence negatively performance of whole system. In such case the threat of breaking main principle of signal propagation during the clock cycle exists. Partitioning of the bus into several segments might be possible solution [1, 2]. In such case signal propagation through each segment will take not more then clock cycle. Such hierarchical bus representation requires additional communication elements as bridges and proper communication protocols (Fig. 1).

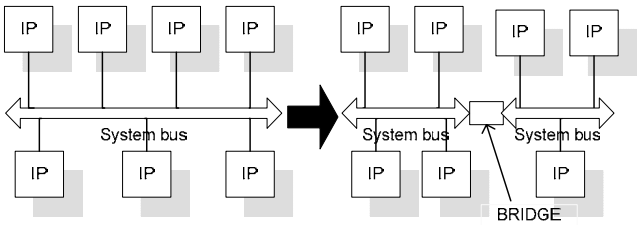

Fig. 1. Bus approach for network-on-chip design

There is a need for a new approaches and automated tools for the verification and test of the networks on chip.

Soc communication architectures

There exist several interface standards for SoC interconnection. Logical structure of the general purpose buses from various vendors, that are widely used today, is similar to each other. Its main idea is that central and most frequently used logic blocks (IP's), such as CPU, Memory are connected by high performance bus, and rarely used peripherals are connected by low performance bus. Low and high performance buses are connected by the bridge. The most widely used general purpose buses are:

Core Connect [3], by (IBM) – consists of high-performance bus – PLB – *Processor Local Bus* for high-performance speed devices and low speed bus for peripheral devices – OPB – *On-Chip Peripheral Bus*. Besides such kind of architecture has third bus – DCR – *Device Control Register Bus* that gathers PLB-connected elements. DCR bus watches over the configuration registers status and increases general bandwidth.

The second wide-spread bus architecture is AMBA [3] – Advanced Microcontroller Bus Architecture. It's structure is very similar to Core Connect architecture. As Core Connect it has high-performance busses – AHB – Advanced High-performance Bus for DSP, CPU and DMA interconnection and ASB Advanced System Bus for micro controllers interconnection. This architecture also has low-performance bus – APB – Advanced Peripheral Bus for peripheral units interconnection. Two high-performance and low-performance threads are separated from each other by the APB bridge. AMBA structure is shown on the Fig. 2.

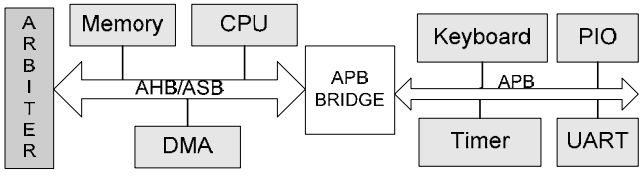

Fig. 2. Scheme of AMBA architecture

The third wide-spread architecture is Wishbone [3] – the bus has the same interface for all IP cores and stands out for simple architecture.

NoC Communication Architectures

Because of the high utilization of the bus and increased length of the interconnect wires the task of the delivering information from point-to-point during one clock cycle becomes hard to implement. That is why there are lots of researches and results in creation of optimal interconnect architecture for the networks on chip. There are five widely used architectures of the networks on chip – SPIN, BFT, CLICHE´, Torus and the Folded Torus. The main principle of those architectures – usage of special buffered routers to store transmitted data.

SPIN. That approach is proposed by Pierre Guerrier and Alain Greiner [4]. Instead of bi-directional throughput wire there are used two 32 bit one-way directional wires that provide point-to point connections between units. The source checks up whether the destination buffer is not overflowed using the counter of free buffer space tracking and the receiver responds to the source how much buffer space were allocated by a separate wire – feedback wire. The SPIN architecture is shown below (Fig. 3).

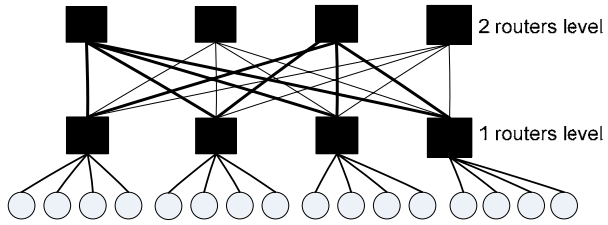

Fig. 3. Scheme of SPIN architecture

Such kind of architecture consists of routers as a nodes and functional units (IP's) as leaves. Among all simple architectures SPIN seems to be complex but despite it is cost-efficient for VLSI.

BFT. Butterfly Fat-Tree (Fig. 4, 5-b) architecture similar to SPIN belongs to fat-tree architectures and has the same concept: the routers are situated in the nodes of a tree and IP units in the leaves. Despite BFT concept has a difference from SPIN. The number of levels depends on a number of IP's:

$$N_{levels} = \log_4 N_{IP's}. \quad (1)$$

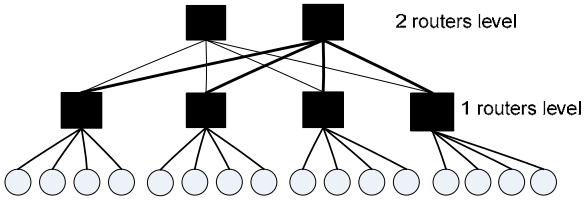

Fig. 4. Scheme of BFT architecture

The number of routers in current level of such kind architecture can be found using the following way:

$$N_{routers} = \frac{N_{IPs}}{2^{l+1}}, \quad (2)$$

where l is a current level.

CLICHÉ. Shashi Kumar et al. [5] proposed methodology called CLICHÉ - Chip-Level Integration of Communicating Heterogeneous Elements (Fig. 5). Each IP unit has a router node. As described before point-to-point connection is supported by two one-directional buses. The router architecture lies in input and output buffers, input and output arbiters, multiplexer, demultiplexer and routing logic.

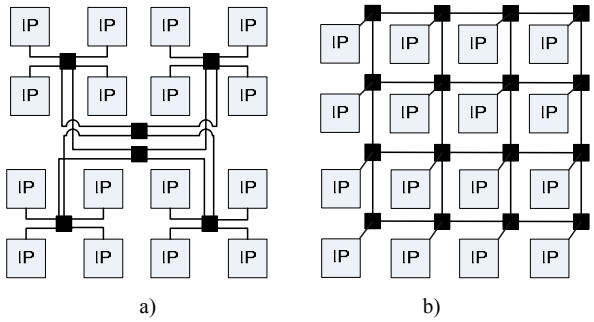

Fig. 5. Scheme of CLICHÉ (a) and BFT (b) architectures

One physical port could have several virtual channels, but only one virtual could have access to a physical port. The arbiter that contained in each router is based on priority matrix as a result gives grants to virtual channel [1, 5].

Such kind of architecture is scalable and has simple structure despite it is not acceptable for parallel computation, data flow, and digital signal processing.

Torus. Such approach was proposed by W.J. Dally et al. It is almost similar to CLICHÉ with the exception of mutual connection of the outermost routers.

Such kind of interconnections could lead to delays after implementation of such architecture. Therefore there was proposed the next architecture called folded torus lies in next nearest routers connection (Fig. 6).

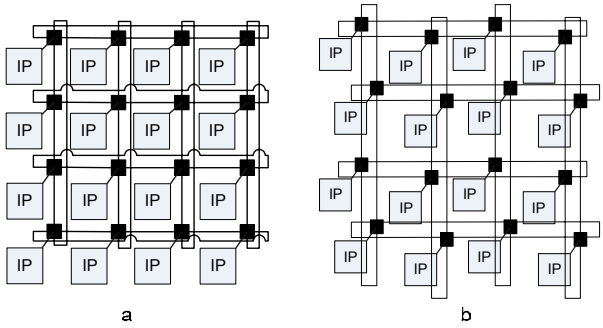

Fig. 6. Torus (a) and folded torus (b)

Octagon. Such approach lies in eight nodes that consist of switch and IP-unit combination that are connected to another three nodes with bi-directional wires (Fig. 7). Such approach is complicated for scaling because of its multiple wires connection despite it is possible to achieve the target node in octagon at least for two steps [2].

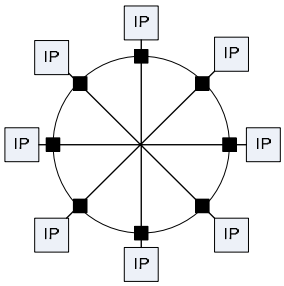

Fig. 7. Scheme of Octagon architecture

Network Communication Protocol

OCP Open Core protocol [6] provides point-to-point interconnection between IP blocks and bus wrappers (bus interface) and describes system level integration requirements. It is system bus independent and provides reusable IP core design. System-on-Chip interconnects provide two types of interfaces: *master* and *slave*. Master interfaces are cores that are capable of generating bus cycles, slave interfaces are cores that are capable of receiving bus cycles. The characteristics of the IP core determine whether the core needs master, slave, or both sides of the OCP. In this case bus interface is an attachment to OCP for each connected IP core (Fig. 8).

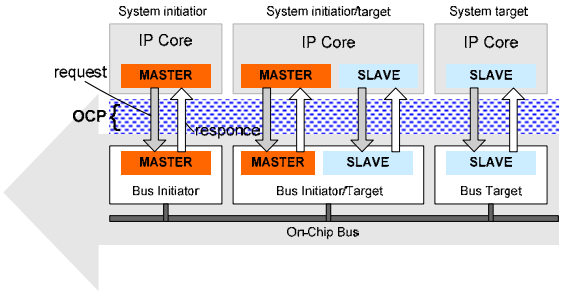

Fig. 8. OCP interface

Verification Approaches

Verification challenges are increasing significantly with the growing number of IPs in NoC. Especially when coming from block level verification to the system level verification, where dozens of IPs simultaneously process, send and receive portions of information.

Today good quality IPs include extra logic for testability after manufacturing [7]. That allows easing the work of test engineers of NoCs. But because of growing complexity of functional verification it is important that IPs also should include additional functionality that will help to automate verification process of the developed system. This additional functionality should help to solve verification problems: to define a bug in the design and perform its localization.

There is no big sense to generate the test for functional test of the IP. When test ones run on IP vendor site it should give same results on customer side (exception can be case with bug in logic simulator or if IP logic has been resynthesized by the customer – but these are not common cases). The best case will be that verification extra logic will check behavior of IP during some workload or regression tests of the designed system.

To implement this approach modern methodology of assertions [8,9,10] suits better then others. It allows to describe functionality of the IP, on the high abstraction level, and check during simulation design constraints and requirements implemented in the form of checkers that have access to the interface and internal structure of the IP (Fig. 9).

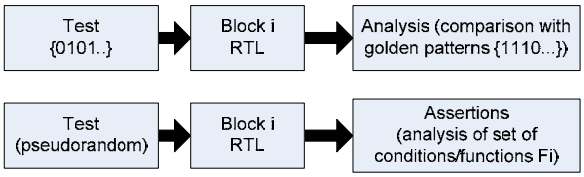

Fig. 9. Verification with assertions versus standard approach

The main advantage is that assertions allow verifying networks on chip with pseudorandom test. When we change input patters assertions analyzer still continue to work and to verify IP or system functionality. Assertions can be used to check communication bus functionality, IP functions and interfaces, chip functions (Fig. 10).

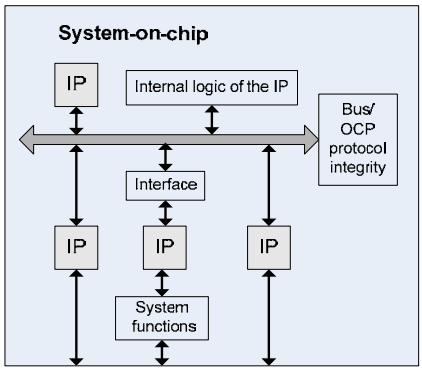

Fig. 10. Placement of the automated checkers during system level verification

The verification process can be presented in such case with general equation T ⊕ F=L, or with the components in more detailed form:

$$\{Tw, Tt\} \oplus \{Fs, Fv\} = \{Lf, Lp, Ls\}, \qquad (3)$$

where {Tw, Tt} – represents workload (pseudorandom) stimulus or specifically generated tests with the defined responses; {Fs ,Fv} – the system description and descriptions of automated verification routines (assertions); L = {Lf, Lp, Ls} – the lists of detectable functional violations (conflicting conditions, functional paths and states).

Conclusions

When we are coming in the area of networks on a chip with dozens and hundreds of IPs on one dice the verification process should also get to the next level. From checking output “0” and “1” with golden patterns to the

smarter approach that is able to check automatically not only bits but design behavior, requirements. Methodology of assertions allows to do it. Assertions reduce work of the test engineer in several times, by covering up to 50% of behavior analysis; by doing checks and notifications at the runtime; by localizing bugs depending of failed assertion in the design; by checking the test coverage and allowing to use pseudorandom generated patterns (workloads) instead of pre-generated tests.

The main drawback of such approach is that smarter verification routines cost more. There are required more resources and more time during simulation run and most likely more expensive tools that supports such approach. The other difficulty is that the procedure of creating assertions is complex and hence costly in terms or engineering and time resources. And it might be not sensible to cover all IPs and interfaces with assertions in terms of one design.

But due to the high reuse nature of the assertions, that ones created for the IP can be used in multiple designs it makes sense to implement some verification standard, similar to IEEE 1149 and IEEE 1500 that will allow vendors to deliver IPs with verification routines, and user will be able easily connect multiple assertions from various IPs in one verification infrastructure during system level verification. That is target of our current and future work.

References

1. **Cristian Grecu, Partha Pratim Pande, Andre Ivanov, Res Saleh.** Timing analysis of network on chip architectures for MP-SoC platforms // Microelectronics Journal.– 2005. – № 36. – P. 833–845.
2. **Partha Pratim Pande, Cristian Grecu, Michael Jones, Andre Ivanov, Resve Saleh.** Performance Evaluation and Design Trade-Offs for Network-on-Chip Interconnect Architectures // IEEE Transactions On Computers. – August 2005. – Vol. 54, No.8.
3. **Rudolf Usselmann.** OpenCores SoC Bus Review. Rev. 1.0, 2001.
4. **Guerrier P., Greiner A.** A generic architecture for on-chip packetswitched interconnections // Proceedings of DATE. – Paris, France, March, 2000. – P. 250–256.
5. **Shashi Kumar, Axel Jantsch, Juha-Pekka Soininen, Martti Forsell, Mikael Millberg, Johny Öberg, Kari Tiensyrjä, Ahmed Hemani**. A network on chip architecture and design methodology // Proceedings of ISVLSI. – 2002. – P. 117–124.
6. **Open** Core Protocol Specification 2.1. Document Revision 1.0.
7. **Rajesh K. Gupta, Zorian Y.** Introducing Core-Based System Design // IEEE Design & Test of Computers, November-December 1997. – P. 15–25.
8. **Maisniemi S., Kalinainen J.** Assertion-Based Verification with PSL Integrated with an Existing RTL Verification Environment // PSL/SUGAR Consortium Meeting DATE. – 2004. – P. 1–5.
9. **Annette Bunker, Ganesh Gopalkrishnan, Sally Mckee**, Formal Hardware Specification Languages for Protocol Compliance Verification // ACM Transactions on Design Automation of Electronic Systems. – 2004. – Vol. 9, No. 1. – P. 1–32.
10. **Open** Verification Library. Assertion Monitor Reference Manual // Accellera. 2004. – V. 3.10.14. – P. 228.

Submitted for publication 2006 12 29

V. Hahanov, O. Yegorov, K. Mostova. NoCs Design for Verification // Electronics and Electrical Engineering. – Kaunas: Technologija, 2007. – No. 3(75). – P. 45–48.

To deploy high performance computing on a chip requires to place the number of the processors in networks on chip (NoC). To fulfill growing market demands number of processors and other IPs on a chip is also increases. Because of that general purpose bus is not efficient to provide communication between IPs and on a chip. In the article there are presented variety of NoC communication architectures and verification approaches on the basis of the hardware assertions. There are proposed design for verification approach, that will allow to use distributed with IP verification routines to ease system level validation. Ill. 10, bibl. 10 (in English; summaries in English, Russian and Lithuanian).

В. Хаханов, А. Егоров, К. Мостова. Проектирование систем на кристалле с учетом возможности верификации // Электроника и электротехника. – Каунас: Технология, 2007. – № 3(75). – С. 45–48.

Реализация механизма высокопроизводительных вычислений требует наличия большого количества процессоров в сети на кристалле. Для удовлетворения требований рынка количество встроенных процессоров и других логических модулей в системах на кристалле растет каждый год. Из-за этого шина с общим доступом уже не удовлетворяет потребности по передаче информации внутри кристалла. Представлены коммуникационные архитектуры и подход к верификации на базе аппаратных предположений. Предложено верификационнопригодное проектирование, которое позволяет упростить системную верификацию благодаря использованию механизмов верификации распространяемых вместе с логическими модулями. Ил. 10, библ. 10 (на английском языке; рефераты на английском, русском и литовском яз.).

V. Hahanov, O. Jegorov, K. Mostova. Sistemų projektavimas kristale atsižvelgiant į verifikacijos galimybes // Elektronika ir elektrotechnika. – Kaunas: Technologija, 2007. – Nr. 3(75). – P. 45–48.

Skaičiavimų našumui padidinti reikia didinti reikia didinti procesorių skaičių kristalo tinkle. Tenkinant paklausą kiekvienais metais didinamas kristalo sistemose įmontuotų procesorių ir kitų loginių modulių skaičius. Šyna su įprasta prieiga jau nebetenkina informacijos perdavimui kristalo viduje keliamų reikalavimų. Pateikiamos komunikacinės architektūros ir prieigos prie verifikacijos remiantis aparatiniais sprendimas. Pasiūlytas verifikacijai tinkamas projektavimas, leidžiantis supaprastinti sisteminę verifikaciją. Il. 10, bibl. 10 (anglų kalba; santraukos anglų, rusų ir lietuvių k.).