International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:6, No:3, 2012

SoC Communication Architecture Modeling

Ziaddin Daie Koozekanani and Mina Zolfy Lighvan

Abstract**—**One of the most challengeable issues in ESL (Electronic System Level) design is the lack of a general modeling scheme for on chip communication architecture. In this paper some of the mostly used methodologies for modeling and representation of on chip communication are investigated. Our goal is studying the existing methods to extract the requirements of a general representation scheme for communication architecture synthesis. The next step, will be introducing a modeling and representation method for being used in automatically synthesis process of on chip communication architecture.

Keywords**—**Communication architecture, System on Chip, Communication Modeling and Representation

.

I. Introduction

THE increasing complexity of digital systems opens new discussions and introduces new challenges in this area. One of such issues is OCCA (On Chip Communication Architecture)[2].

Because of some application requirements, most of the current systems are built on a single chip. In these SoCs (System on Chips), establishing the required connections between system components, is one of the major designing problems.

Several parameters related to implementation technology and design constraints influence the on chip communication architecture. Having a proper tradeoff between such factors is a big problem for the designers. In order to overcome this problem, some modeling methodologies are introduced. These methodologies lead the designers in their way of choosing the appropriate communication architecture with the proper tradeoffs between design parameters.

The rest of this paper is organized as follows: In the next section a short introduction to on chip communication architecture modeling is presented. In section III a survey of some approaches in communication modeling is presented. Then paper will be concluded with the conclusion part in section 5.

Ziaddin Daie Koozekanani is Deen of Electrical and Computer Faculty of University of Tabriz, Tabriz, Iran. (phone: +98-411-3393710, fax: +98-411-3300829; e-mail: zdaie@tabrizu.ac.ir).

Mina Zolfy Lighvan is a faculty member of Electrical and Computer Faculty of University of Tabriz, Tabriz, Iran. (phone: +98-411-3393759, fax: +98-411-3300829; e-mail: zdaie@tabrizu.ac.ir).

.

II. OCCA (On Chip Communication Architecture) Modeling

In every design process, modeling the target system helps the designer to represent some of the design parameters and extract some other information consequently. For example in the design flow of an aero dynamic system, before final manufacturing the engineers generate a model of their system and use that model in their experiments. The same scenario exists in SoC design flow. When the designer has decided about the system components and wants to design the communication platform of the entire system, a fine model of the communication architecture would be very useful. But as a matter of fact, any model of a system is an abstract model and carries only a subset of system specification. Therefore, that engineer must choose an appropriate modeling scheme based on his modeling requirements. Several communication architecture modeling methodologies are presented by now at system level. In the following subsections the main categories are pointed out.

A. Model of Performance Exploration

In some methodologies of OCCA modeling, the goal is exploring communication architecture performance and timing[3]. Several source of delay can manipulate OCCA timing. For example wires length, arbitration mechanism, transfer modes and synchronization are some delay sources in any communication architecture. Every performance model must be able to consider such delays.

Some of these performance models are static models[4]. In static models, the performance of the systems is determined analytically. Another class of performance models is dynamic models in which the results are obtained dynamically through some simulation phases. There are also some hybrid models that take advantages of both static and dynamic models.

B. Model of Thermal and Power Estimation

With the technology scale down, power consumption along the communication architecture becomes an essential bottleneck for the designer. Therefore thermal and power estimation of communication architecture is getting more important. There are several methodologies presented for power and thermal modeling[5, 6]. The most important aspects of such models are the bus wire power model and the communication logic power model. Such models help the designer in estimating the power consumption and thermal specification of the designed architecture.

III. Representation Methods

A. Petri Nets

Petri nets[7] were introduced in 1962 as the result of Dr. Carl Adams Petri PHD thesis. They are graphical and mathematical tool which are suitable for modeling concurrent and distributed systems. There is no representation of time and timing in the classic form of Petri nets, but the later generation of Petri nets such as CPN, TPN, DSPN and SPN includes some timing information. Petri nets are appropriate tools for modeling concurrency, sequentially conflict and synchronization issue in parallel systems. In recent times, the application of deterministic and stochastic Petri-Nets (DSPNs) for modeling on-chip communication has been proven to be an striking method to evaluate and explore different communication aspects[1, 8].

The graphical presentation of a Petri net is a bipartite graph which includes two types of nodes: Places and Transitions. The Places usually model resources or partial state of the system where the transitions model state transition and synchronization. In this bipartite graph, arcs are directed and always connect nodes of different types.

Fig. **1**[1] shows a Petri net model which demonstrates the communication scenario in a Avalon-NIOS example. In order to extract required information from a Petri net model three different approach could be used: Mathematical analysis, mathematical approximation and simulation. Because of the important role of Petri nets in modeling methodologies, several tools are implemented for simulation and also for analysis of them.

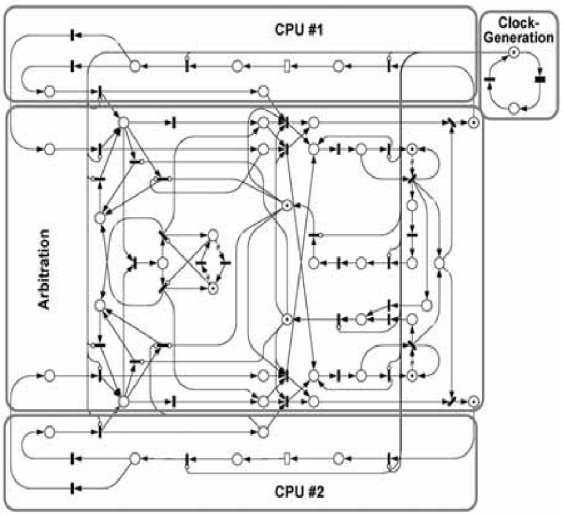

Fig. 1 DSPN model of Avalon-NIOS[1]

B.UML

UML(Unified Modeling Language)[9] is a standardized general-purpose modeling language, which consists of a set of graphic notation techniques for creating visual models of object-oriented software-intensive systems. UML is used to specify, visualize, modify, construct and document the artifacts of an object-oriented software-intensive system. UML is a powerful modeling language but the lack of concurrency and parallelism aspects of it, bound it in the modeling of concurrent systems. For overcoming this limitation and on the other hand for taking advantages of its facility, in several researches UML and actor oriented modeling are used together[9-11]. SAVY[12] is a design pattern which gets a UML model of a system and converts it to a model in OSCI TLM.1.

C. CSP

CSP(Communicating Sequential Processes)[13] is a classical approach for describing the communication in parallel systems. CSP is a mathematical notation which is used in some tools and methods for modeling communication concurrency. Any system model based on CSP is a combination of independent processes. These processes interconnection is synchronized with Message passing communication events. CSP is used in the kernel of several ESL tools such as FDR and FOSSY[14].

One of the important application of CSP was its use for specification and verification of elements of the INMOS T9000 Transputer[15], a complex superscalar pipelined processor designed to support large-scale multiprocessing. CSP was employed in verifying the correctness of the processor pipeline, and also the Virtual Channel Processor which managed off-chip communications for the processor[16].

D. Colif

Colif[17] is an object-based component interconnection model which abstracts MPSoC (Multi Processor SoC) architecture as a virtual architecture. In this virtual model, hardware/software components use and/or provide communication services.

In Colif, based on the three main features, HW/SW component Integration in a higher level of abstraction becomes possible. These three features are:1) Flexible modeling of component communication,2)Separation between component communication and behavior and 3) mixed abstraction level modeling. The colif primary virtual model, converts to an executable model after several refinement steps.

If you are using *Word*, use either the Microsoft Equation Editor or the *MathType* add-on (http://www.mathtype.com) for equations in your paper (Insert | Object | Create New | Microsoft Equation *or* MathType Equation). “Float over text” should *not* be selected.

E. KPN

Kahn Process Network[18] is another model of computation. A KPN is consisting of a group of concurrent processes that are connected in a point to point scheme.

In KPNs the concurrent processes are communicating through unbounded FIFO channels. In such a model the processes are executed independently and their concurrency is applied using some blocking read operations. The control mechanism has a distributed form and no central controller

exists.In [19] KPN is used mapping of a specific application onto a system containing a FPGA and a microprocessor. In this research the KPN model is extracted automatically from a application specification described in MATLAB and then the resulted model is mapped on the hardware systematically.

IV. Conclusion

In the previous sections a survey of on chip communication modeling and the mathematical based representation approaches are presented. The study shows that there are mathematical tools, that are good at representing some aspects of the communication architecture but no one is good enough in illustrating all required features of communication architectures. Therefore for obtaining a better modeling methodology we must manipulate and also combine some of the existing techniques.

References

[1] Blume, H. , Sydow, T. V. , Schleifer, J. , and Noll, T. G. , *Petri Net Based Modeling of Communication in Systems on Chip*, V. Kordic, Editor. 2008: Vienna.

[2] Pasricha, Sudeep and Dutt, Nikil, *On-Chip Communication Architectures (System on Chip Interconnect)*. 2008: Morgan Kaufmann Publisher.

[3] Viaud, E., Pecheux, F., and Greiner, A. *An efficient TLM/T modeling and simulation environment based on conservative parallel discrete event principles*. in *Date 2006*. 2006.

[4] Renner, F, Becker, D., and Glesner, M, *Communication performance models for architecture precise prototyping of real-time embedded systems.* RSP, 1999: p. 108-119.

[5] Gupta, P, Zhong, L, and Jha, N - K. *A high-level interconnect power model for design space exploration*. in *ICCAD 2003*. 2003.

[6] P, Sotiriadis and P., Chandrakasan A., *A bus energy model for deep submicron technology*. IEEE Transactions on Very Large Scale Integration Systems (TVLSI). **10**(3): p. 341-350.

[7] Peterson, J. L. , *Petri Net Theory and the Modeling of Systems*. 1981: Prentice Hall, Englewood Cliffs.

[8] Blume, H., Sydow, T. V., Becker, D., and Noll, T. G., *Applicatio of Deterministic and Schocastic Petrinets for Performance Modeling of NoC Architecture*. Journal of Systems Architecture, 2007. **53**: p. 466-476.

[9] Rumbaugh, J, Jacobson, I, and Booch, G, *The Unified Modeling Language Refrence Manual*. 1998: Addison-Wesley.

[10] Indrusiak, L. and Glesner, A. *SoC Specification using UML and Actor-Oriented Modeling* in *Baltic Electronics Conference, 2006 International*. 2006. Tallin.

[11] Lee, E, Neuendorffer, S, and Wirthlin, M, *Actor Oriented Design of Embedded Hardware and Software Systems*. Journal of Circuits Systems and Computers, 2003. **Vol. 12**(No. 3): p. 231-260.

[12] Perez, J, Sevillano, J, Urcelayeta, S, and Velez, I. *System Behaviour Capture: From UML to SystemC*. in *Forum on Specification and Design Languages* 2008.

[13] Hoare, C. A. R, *Communicating Sequential Processes*. 2004: Prentice Hall International Series in Computing Science. 260.

[14] Brunzema, C and Nebel, W. *CSP with Synthesisable SystemC and OSSS*. in *FDL'07*. 2007.

[15] Allen Kent and Williams, James G., *Encyclopedia of Computer Science and Technology*. 1998.

[16] Barrett, G, *Model checking in practice: The T9000 Virtual Channel Processor*. IEEE Transactions on Software Engineering 1995. **21**(2): p. 69-78.

[17] Cesario, W., Gautheir, L., Lyonnard, D., Nicolescu, G, and Jerraya, A, *Object_based Hardware/Software Component Interaction Model for Interface Design in System-on-a-Chip circuits.* The Journal of Systems and Softwares, 2004: p. 229-244.

[18] Lee, E. A and Parks, T. M, *Dataflow process Networks.* Proceedings of the IEEE, 1995. **83**(5): p. 773-799.

[19] Stefanov, T, Zissulescu, C, Turjan, A, Kienhuis, B, and Deprettere, E. *System Design Using Kahn Process Networks: The Compaan/Laura Approach*. in *Date'04*. 2004.