High-Level Simulations of On-Chip Networks

Claas Cornelius Frank Sill Dirk Timmermann
Institute of Applied Microelectronics and Computer Engineering
University of Rostock, Germany
{claas.cornelius, frank.sill, dirk.timmermann}@uni-rostock.de

1. Introduction

The International Technology Roadmap for Semiconductors [1] prognoses up to four billion transistors on a single chip with feature sizes below 45 nm operating in the 10 GHz range. This will be achieved by continuous scaling of transistor dimensions and offers the opportunity to further increase the performance into the Tera-Flop era. Unfortunately, the scaling is about to approach physical limits which causes new challenges and serious problems that endanger the future development. Such issues emerge at diverse abstraction levels and they have to be considered during all design and production steps. Leakage currents, power density, clock distribution, reliability, verification, memory-bottleneck, and design-productivity-gap are just an extract of all these issues.

Network-On-Chip (NOC) has been proposed as a new design methodology to cope with these problems [2]. NOCs consist of heterogeneous resources that are encapsulated from another which allows the use of different voltages, frequencies or even diverse technologies for each resource. Furthermore, encapsulation eases the exchange and reuse of resources as well as the integration of intellectual property (i.e. predefined macro blocks). The independent resources are connected (by a determined interface) to an on-chip network which affords the communication among the resources whereas the network consists of routers and physical links. An example of two possible topologies with the mentioned components is given in figure 1.

Up to date, large effort has already been put into the development of elementary modules to operate NOCs in principle. For instance repeater, booster, and phase coding have been suggested for signal transmission, coding and error correction schemes have been analyzed as well as bufferless or asynchronous routers have been integrated in various topologies (e.g. mesh, torus, fat-tree) applying diverse routing schemes (e.g. XY, adaptive, deflective). First prototypes have also been produced and their functionality has successfully been proven [3][4]. However, those prototypes are application specific NOCs with a limited number of resources so that a multitude of issues of future general purpose NOCs with several hundred resources, as predicted by [5] and others, does not have to be considered.

Because corresponding technologies to implement such large NOCs are not available yet and the expenditure of time as well as the cost-value ratio of large functional test-chips is also questionable, the behavior of NOCs needs to be simulated to obtain results for performance, power, area, and further network related metrics like bandwidth usage or areas of increased packet congestion. Such simulations are necessary both for designers to exploit different hardware architectures and solutions as well as for vendors to evaluate different existing chips for their application-specific domains.

2. Targeted Problems

Even though, first prototypes have been presented, there is still a large number of open questions that need to be solved to make the use of NOCs generally applicable. Therefore, a functional simulator is understood as an essential tool to investigate problems and assess solutions. For instance, different architectural approaches during the design phase need to be compared which includes amongst others topology, routing scheme, bus width, and link implementation.

However, to efficiently implement applications for a given NOC architecture, appropriate programming models are necessary to exploit the specific properties like the parallel structure of resources or distributed memory. If such models and programs exist, it needs to be determined

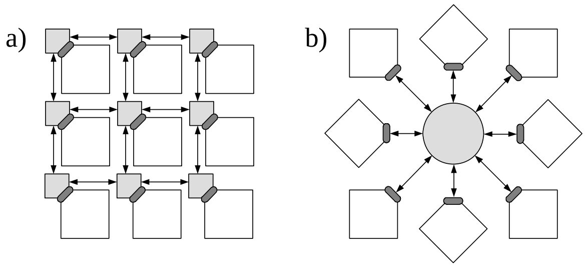

Fig. 1 Simple example of a a) mesh and a b) star topology of a Network-On-Chip (NOC)

how the programs can be mapped onto a given NOC architecture. This so called application mapping can occur both during the development phase and during online operation and requires the consideration of communication-computation trade-offs, power issues, quality of service requirements, or thermal characteristics.

But, online application mapping is just one element of a required operating system for the management of a NOC. Load balancing, power issues, hardware reconfiguration, availability of resources and services as well as more administrative tasks need to be taken care of by such a system. If a centralized or decentralized approach is more suited for these purposes, is application specific and depends for instance on control overhead and the number of resources. Because sporadic errors will be unavoidable and malfunction of resources will have to be dealt with, testing and verification also needs to be considered at some stage. This will allow static adaptation or even online self-healing and can also increase the yield and reliability of large integrated circuits. Finally, it demands for appropriate metrics and benchmarks to assess own solutions or to compare achieved results within the research community.

3. Proposed structure

The NOC configuration in figure 2 implies several steps that can be found in standard design flows like definition of system requirements, design of an architecture, and hardware/software partitioning. Subsequently, the primary input for the simulator is a topology as a result of the configuration or alternatively from an existing chip. The topology includes the information of what type the different used modules are, at which logical position they are situated and which other modules they are connected to. This refers to all applied modules as the resources, interfaces, routers, and links.

The simulator sets up the given configuration by using modules from a NOC library that have previously been defined in terms of behavior as well as delay, power consumption, area, and further necessary parameters. Such a module can be a link with repeaters covering a certain distance and offering a determined bus width. The routers and interfaces are defined in a similar way. In contrast to all other modules, the resources possess additional parameters that describe their network related behavior like packet injection rate. For instance, a resource of a crypto core will receive and send packets continuously to only one destination requiring dedicated bandwidth whereas a resource that coordinates activities within the NOC will rather send and receive short control messages to varying destinations requiring minimal delay. The classification of resources and their behavior allows the benchmarking of large general purpose NOCs for different application domains without the need to know the operating applications in advance.

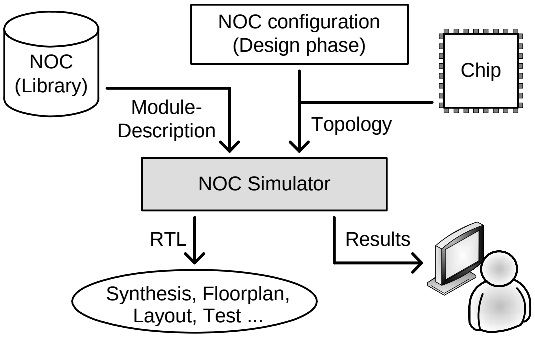

Fig. 2 Integration of the NOC simulator into an automated design flow

The simulation itself is performed on the level of modules and not on gate or even transistor level. For example, the delay for a transaction between two resources is derived from the sum of the delays for each involved module. Network related metrics like link usage and packet congestion can also be determined by the use of activity counters for the links and repeaters. Due to the high abstraction, this approach can possibly mask specific behavior and is not as exact as cycle accurate simulations. But, the promising advantage is the extraordinary reduction of simulation time. Finally, the results are written to a text file, but the future implementation shall also include the output on a graphical user interface to ease the evaluation of the various parameters.

The secondary output shall be a Register Transfer Level (RTL) description to allow the integration into an automated design flow which is absolutely mandatory to implement complex integrated circuits. The further steps of such a flow, like synthesis, placement, or routing, will have to incorporate and deal with issues of ultra deep submicron technologies that are not solely related to NOCs like reliability and signal integrity, yield optimization, as well as parameter variations.

References

[1] Semiconductor Industry Association (SIA), “International Technology Roadmap for Semiconductors”, 2003.

[2] W. Dally and B. Towles, “Route packets, not wires: On-chip interconnection networks”, DAC, pp. 684-689, 2001.

[3] R. Mullins, A. West, and S. Moore, “The design and implementation of a low-latency on-chip network”, ASP-DAC, pp. 164-169, 2006.

[4] K. Lee et. al., “A 51mW 1.6GHz on-chip network for low-power heterogeneous SoC platform”, ISSCC, pp. 152-153, 2004.

[5] A. Jantsch, "NoCs: A new contract between hardware and software", DSD, pp. 10-16, 2003.