Chapter 6
From MARTE to Reconfigurable NoCs: A Model Driven Design Methodology

Imran Rafiq Quadri
CNRS, France

Majdi Elhaji
CNRS, France

Samy Meftali
CNRS, France

Jean-Luc Dekeyser
CNRS, France

ABSTRACT

Due to the continuous exponential rise in SoC’s design complexity, there is a critical need to find new seamless methodologies and tools to handle the SoC co-design aspects. We address this issue and propose a novel SoC co-design methodology based on Model Driven Engineering and the MARTE (Modeling and Analysis of Real-Time and Embedded Systems) standard proposed by Object Management Group, to raise the design abstraction levels. Extensions of this standard have enabled us to move from high level specifications to execution platforms such as reconfigurable FPGAs. In this chapter, we present a high level modeling approach that targets modern Network on Chips systems. The overall objective: to perform system modeling at a high abstraction level expressed in Unified Modeling Language (UML); and afterwards, transform these high level models into detailed enriched lower level models in order to automatically generate the necessary code for final FPGA synthesis.

INTRODUCTION

Since the early 2000s, System-on-Chip (SoC) has emerged as a new methodology for embedded systems design. In a SoC, the computing units (programmable processors, hardware functional units), memories, I/O devices, communication channels, etc.; are all integrated into a single chip. Moreover, multiple processors can be integrated into a SoC (Multiprocessor System-on-Chip, MSoC) in which
the communication can be achieved through Network on Chips (NoCs). These SoCs are generally dedicated to target application domains (such as multimedia video codecs, software-defined radio and radar/sonar detection systems) that require intensive computations. According to Moore’s law, rapid evolution in hardware technology doubles the number of transistors in an Integrated Circuit (IC) nearly every two years. As the computational power increases, more functionalities are expected to be integrated into the system. As a result, more complex software applications and hardware architectures are integrated, leading to a system complexity issue which is one of the main hurdles facing SoC co-design. The fallout of this complexity is that the system design (particularly software design) does not evolve at the same pace as that of hardware due to issues such as development budget limitations, reduction of product life cycles and design time augmentation. This evolution of balance between production and design has become a critical issue and has finally led to the productivity gap. System reliability and verification are also the other issues related to SoC industry and are directly affected by the design complexity. An important challenge is to find efficient design methodologies that raise the design abstraction levels to reduce overall complexity, while effectively handling issues such as accurate expression of inherent system parallelism: such as application loops; and hierarchy.

Network on Chips is considered as an emerging paradigm for resolving the problems related to current highly integrated complex SoCs (Benini, L., and Micheli, G 2001). A SoC may have tens or hundreds of IP (Intellectual Property) cores with each running at different clock cycles resulting in asynchronous clocking. NoCs thus adopt a globally asynchronous, local synchronous (GALS) approach and help to improve the performance: such as throughput; and scalability as compared to other communication structures such as point to point signal wires and shared buses. They are an ideal choice for MPSoC architectures as they allow separation of the communication and the computation concerns while allowing IP reuse by utilization of standard interfaces.

Currently High Level Synthesis (HLS) (or Electronic System Level) is an established approach in SoC industry. This approach raises the design abstraction level to some degrees as compared to traditional hand written HDL (Hardware Description Languages) implementations. The gap between the high abstraction levels and the low abstraction levels is often bridged using one or several Internal Representations (IRs) (Guo et al 2005). The behavioral (algorithmic) description of the system is written in a high level language such as SystemC (OSI 2007) or a similar language, and is then refined into a RTL (Register Transfer Level) implementation using HLS tools. An effective HLS flow and associated tools must be flexible to cope with the rapid hardware/software evolution; and maintainable by the tool designers. The underlying low level implementation details are hidden from users and their automatic generation reduces time to market and fabrication costs. However, usually the abstraction level of the HLS tools is not elevated enough to be totally independent from low level details. Normally, the set of concepts related to an IR are generally difficult to handle due to absence of formal definitions of key concepts and their relations. The text based nature of a system description also results in several disadvantages. Immediate recognition of system information such as related to hierarchy, data parallelism and dependencies is not possible; differentiation between different concepts is a daunting task in a textual description and makes modifications complex and time consuming.

Model Driven Engineering (Planet MDE 2007) (MDE) is an emerging domain and can be seen as a High Level Design Flow in order to resolve the issues related to SoC co-design. MDE enables system level (application/architecture) modeling at a high specification level allowing several abstraction stages (i.e. IRs). Thus a system can be viewed globally or from a specific point of view...
Related Content

Predictive File Replication on the Data Grids
[www.igi-global.com/article/predictive-file-replication-data-grids/38979?camid=4v1a](www.igi-global.com/article/predictive-file-replication-data-grids/38979?camid=4v1a)

Distributed Image Processing on a Blackboard System
[www.igi-global.com/chapter/distributed-image-processing-blackboard-system/20523?camid=4v1a](www.igi-global.com/chapter/distributed-image-processing-blackboard-system/20523?camid=4v1a)

Dynamic Task Assignment with Load Balancing in Cloud Platform
[www.igi-global.com/chapter/dynamic-task-assignment-with-load-balancing-in-cloud-platform/139853?camid=4v1a](www.igi-global.com/chapter/dynamic-task-assignment-with-load-balancing-in-cloud-platform/139853?camid=4v1a)

Architecting Software as a Service for Data Centric Cloud Applications
[www.igi-global.com/article/architecting-software-as-a-service-for-data-centric-cloud-applications/114714?camid=4v1a](www.igi-global.com/article/architecting-software-as-a-service-for-data-centric-cloud-applications/114714?camid=4v1a)