A Metamodeling Component Composition Framemework for Dynamic Partially Reconfigurable Systems

Gilberto OCHOA-RUIZ[1], Pamela WATTEBLED[2], Ouassila LABBANI[1], El-Bay BOURENNANE[1] Jean-Luc DEKEYSER[2]

[1]Laboratoire de Electronique, Informatique et Image, LE2I
Alle Alain Savary, 21000, Dijon, France

[2]INRIA Lille Nord Europe - Equipe DART
40, avenue Halley - Bat A, 59650,Villeneuve dAscq, FRANCE

gilberto_ochoa-ruiz@etu.u-bourgogne.fr, pamela.meftali@inria.fr
ouassila.labbani@u-bourgogne.fr, ebourenn@u-bourgogne.fr, jean-luc.dekeyser@inria.fr

Thème – Langages et outils de développement

Problème traité – We deal with the automatic generation of dynamic partially reconfigurable platforms through the use of the Model Driven Engineering (MDE) paradigm.

Originalité – The proposed MDE methodology aims at facilitating the conception of the DPR design flow using a Metamodeling Component Composition Framework (MCF). The front-end models are described using UML MARTE, and through model transformations, a Xilinx proprietary platform model is created. We make use of the recently introduced IEEE IP-XACT standard as an intermediate representation; this enables us to define a flow agnostic approach in which different back-ends can be targeted

Résultats – We have created different DPR platforms, using different sets and types of PR IPs. The proposed approach have demonstrated to improve the design process and to reduce the required conception effort.

1 Introduction

Despite the efforts by Xilinx and many industrial and academic endeavors, creating complex DPR systems remains a very daunting task. This is due to the complexity of the design flow [1], which requires and in-depth knowledge of many low level aspects of the FPGA technology.

In this paper, we propose a framework to leverage the conception of DPR systems. In fact, the proposed methodology is intended as Metamodeling Composition Framework (MCF) for visually enabling the construction FPGA-based SoC systems. It is based in a Model Driven Engineering (MDE) [2] approach in tandem with a component based approach. Thus, the seamless integration and interoperability of the used IP is a necessity.

Several approaches have made use the UML profile and extensions to support embedded hardware resources modelling. Many of them made use of the UML profile for Modelling and Analysis of Real Time and Embedded Systems (MARTE) [4], which is a proposal for OMG standard profile. However, the mechanisms to move from UML specifications to enriched levels have not been standardized, and every approach manages this issue by defining their own transformation rules and intermediate representations (using custom XMI data-books).

Recently, IEEE IP-XACT standard [5] has been introduced as a means to describe IP meta-data for SoC integration. Several industrial cases studies have demonstrated that the adoption of IP-XACT facilitates the configuration, integration, and verification in multi-vendor SoC design flows [6] and in academic tools [7].

The contributions of this paper relate to presenting an MDE approach that uses the UML MARTE profile that enables moving from high level models to HDL code generation for the implementation of the IP core sub-system description. IP-XACT is used as an intermediate model, used to configure the instantiated IPs in the platform and to automate their parameterization ; this is process is federated by metaprogrammable tool, Sodius MDWorkbench [9], which contains the associated meta-models and transformation rules.. The parameterized system and IPs are then used to generate the necessary inputs to the Xilinx Platform Studio (XPS) tool, in which the final netlist for the system and the IPs are created [8].

2 High-level Platform Description and Automatic Generation

In the Electronic Design Automation (EDA) domain, the management of data and metadata plays an important role in the context of system-level design. Many design methodologies use a pattern which consists of assembling reusable sub-systems and/or IP blocks (IP) to construct complex system platforms. This design process relies heavily on metadata. The formalisms used to describe system architectures within this process are often referred to as models of architecture, and they conform to metamodels that are built upon an XML representations [3]. XML technology serves as a stack for metadata management and its manipulation ; XML trades off brevity, readability, and human comprehension for ease of parsing and use in automate tools and systems.

The most important feature of a MCF an approach for co-design, is the creation of this XML IP library. A reflection mechanism must be set up to identify, extract, and represent IP metadata, and to store it in XML structures. Moreover, these XML structures must conform to an standardized schema for promoting IP reuse ; for this purpose we make use of the IP-XACT standard. The basic idea of the proposed MFC is to the visualization and manipulation of IP cores contained in the XPS tool using an UML MARTE environment. These IPs are described using a proprietary format, the Microprocessor Peripheral Definition file, that abstracts the VHDL implementation of the IP, as depicted on Figure 1 a).

Nevertheless, dealing directly with the MPD description would tie the MCF to an specific technology, which goes against the principles of MDE. Therefore, we transform these files into IP-XACT component descriptions, obtaining a library of XML components that can be interchangeable among different tool vendors and design flows. This transformation is achieved through a MPD to IP-XACT component model transformation (MT), aided by a set of transformation rules (TR). The last phase of the IP creation library is the conversion of the IP-XACT components into a set of XML files that conform to an UML component extension defined in MARTE. In this manner, a set of XML components is created and stored in the MARTE library.

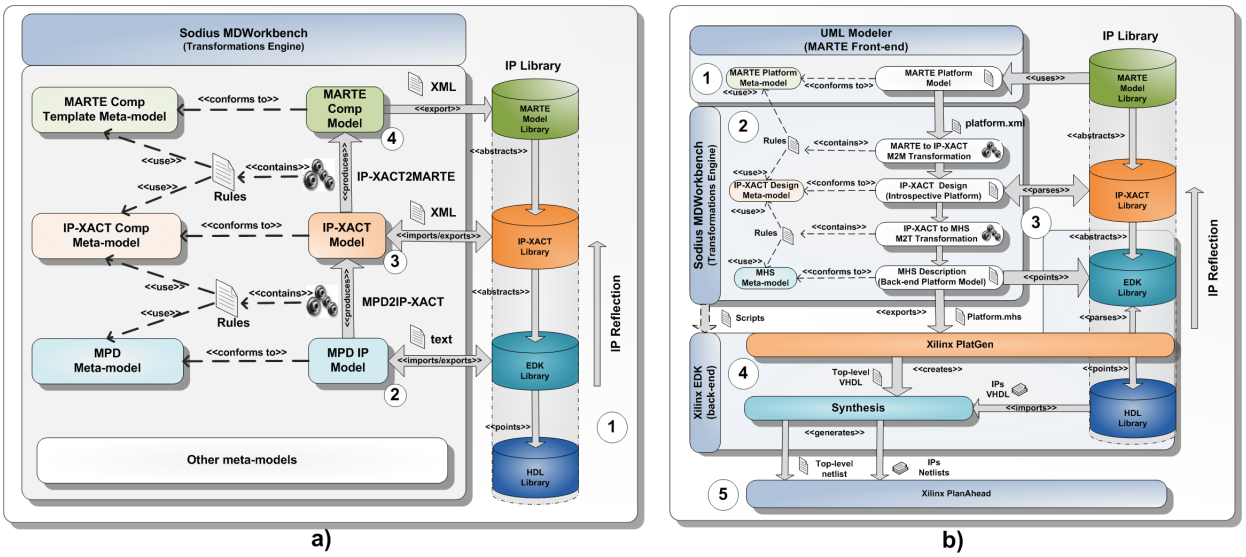

FIGURE 1 – a) Creation of the mult-level IP library. b) Metamodeling Component Composition Framework for DPR systems

Once the multi-level library has been created, it can be used by the Metamodeling Component Composition Framework for building a platform model, as depicted on Figure 1 b). The entry point is a MARTE platform model, which is created by choosing components from the MARTE Library. These components are seen as simple blocks containing interfaces to be connected and parameters to be set by the designer. For this purpose, the extension to the MARTE profile used for creating the library is deployed here for linking the instances in UML model to their IP-XACT counterparts, obtaining their properties automatically [10].

The MARTE model is then imported to Sodius MDWorkbench to obtain an IP-XACT system description, using of a IP-XACT design metamodel, and a set of TR. The IP-XACT XML schemas have been processed by an improved XSD/Ecore meta-model importer in MDWorkbench, which leads to the creation of an IP-XACT metamodel.

The IP-XACT design is then used to perform an MT from IP-XACT to the XPS model for describing the platform. The meta-model for this description has similarly been created in UML and imported into the Sodius tool, and several TR defined. Therefore, we are able to generate the XPSF platform model (MHS, Microprocessor Hardware Specification) through a model-to-text transformation. The MHS file is used by a Xilinx tool, PlatGen, to obtain the top-HDL description of the platform, and which at the same time references to the HDL IPs, that are configured in this phase. Then HDL files can be synthesized an used as input to the DPR design flow, as described in [1]. This work represents just an an axe of the ANR FAMOUS framework, whilst other complementary works deal with the complete modeling and generation of DPR applications.

3 Improving IP and design by reuse through Model-drive Engineering Techniques

In this section we discuss the advantages in terms of the design effort required to implement a system (not detailed in this summary for lack of space), especially if we compare the proposed approach with a purely VHDL approach and, as in the case of the generation back-end of this methodology, using Xilinx EDK. Let us consider for instance typical VHDL top level design, which is generated in around 30 seconds by PlatGen, as depicted in Table 1. The top-level VHDL description contains 7986 lines of code, and contains mainly components instantiations, parameterization and signals declarations for interconnection. It is evident that creating such a design (composed of several components, and multiple sub-components) would take not only hours, but maybe days, in a process very prone to errors, as shown in Table 2.

Table 1
Number of lines of each model and time required for generation

File Type	Nb of Lines	Exec. time	Description
XMI	654	50 sec	MARTE model in XMI for transformation purposes into IP-XACT in MDE Workbench
IP-XACT *Design XML*	998	30 sec	Intermmediate model used in the transformation phase to obtain the MHS file
MHS File Top Level	455	20 sec	This file is used by EDK to instantiating the IP blocks, parameterisation and interconnection
VHDL Top Level	7986	30 sec	HDL description of the system. Obtained by feeding Xilinx Platgen with the MHS and MPD files
Netlist Top Level	N/A	12 min	The system VHDL top level file is used to obtain a NGC for the static part of the platform

Table 2
Design + DPR IP Capture Times

Type of design capture	Time Used	Description
Pure VHDL Approach		This method is the less relaible, long
Manually integrating the platform	Days	and prone to error. Good for small systems
Manually modifying DPR Ips	Hours	Not support for DPR management
Using Xilinx EDK		EDK is justitfiable for systems containing
Platform Integration in XPS	1h30 min	at least one processor (DPR manager)
Manually modifying DPR IP+ MPD	25 min/IP	IP blocks need to be processed separately
Proposed Approach		The time required for a platform creation is
Platform Integration	40 mins	reduced, and the mantainability is improved
		IP blocs creation (VHDL) takes the same
Modifying DPR IP+ MPD	23min/IP	time, but integration is improved

Xilinx EDK, using the Platform Specification Format (PSF, notably MHS and MPD files), makes the design process more amenable : the designer can start creating a design through an easy to use Graphical User Interface (GUI), and then parameterize the design by choosing different options through IP specific TCL files and GUIs. These changes are automatically updated in the MHS files by parsing the corresponding MPD file and checking for any dependencies on parameters. However, the creation of the platform in Xilinx EDK is not completely automated, and a lot of steps still need to be performed manually ; for instance, importing IPs into the platform, their interconnection and parameterization. All these steps require a great deal of design effort and expertise of the tool and this is precisely one of the advantages of using the proposed methodology : by using a high-level description, the designer does not to know all the specificities of the used tools, which often are difficult to grasp by people who are not proficient into FPGA design and VHDL.

Another advantage of using UML and MARTE is the maintainability and improved updatability of the models ; this means that, contrarily to purely VHDL or EDK flows, a change in the platform requires much less effort : since every step of the design flow is automated, the designer does not even need to make use Xilinx EDK or ISE. The IP-XACT descriptions also facilitate the updatability of the approach by changing the vendor extensions or the target meta-models, but not the implementation files.

Références

[1] Xilinx Corporation, *Partial Reconfiguration User Guide, Xilinx UG208* , 2011.

[2] J-L. Dekeyser, P. Boulet, P. Marquet, and S. Meftali, *Model driven engineering for SoC co-design* , in IEEE-NEWCAS Conference, 2005. The 3rd International, vol., no., pp. 21- 25, 19-22 June 2005

[3] S. Decker, S. Melnik, F. van Harmelen, D. Fensel, M. Klein, J. Broekstra, M. Erdmann, I. Horrocks,*The Semantic Web : the roles of XML and RDF* , in IEEE Journal on Internet Computing, 2000

[4] OMG, *Modeling and Analysis of Real-time and Embedded systems (MARTE)* , http ://www.omg.org, 2012

[5] IEEE, *IEEE Standard for IP-XACT, Standard Structure for Packaging, Integrating, and Reusing IP within Tools Flows* , in IEEE Std 1685-2009, Feb. 18 2010.

[6] W. Kruijtzer et al., *Industrial IP Integration Flows based on IP-XACT Standards* , in DATE08, March 2008, pp. 3237.

[7] S. Revol, S. Taha, Safouan, F. Terrier, A. Clouard, S. Gerard, A. Radermacher,J.-L. Dekeyser, *IP Meta-Models for SoC Assembly and HW/SW Interfaces* , in Distributed Embedded Systems : Design, Middleware and Resources, Springer, 2008

[8] Xilinx Corporation, *Embedded System Tools Reference Guide* , Xilinx UG111,September 2012

[9] Sodius,*MDWorkbench* , http ://www.mdworkbench.com/, 2013.

[10] G. Ochoa-Ruiz et al, *A high-level methodology for automatically generating Dynamic Partially Reconfigurable systems using IP-XACT and the UML MARTE profile* , in Design Automation for Embedded Systems, Springer, 2012