Model-Based Design of a 5G Uplink Data-Link Layer Receiver from UML/SysML Diagrams

Andrea Enrici, Julien Lallet, Imran Latif
Nokia Bell Labs, Centre de Villarceaux,
91620 Nozay, France
Email: {firstname.lastname}@nokia.com

Ludovic Apvrille, Renaud Pacalet, Adrien Canuel
LTCI, CNRS, Télécom ParisTech
Université Paris-Saclay, 75013 Paris, France
Email: {firstname.lastname}@telecom-paristech.fr

Future 5G networks are expected to provide higher (10x) data-rates and to support use cases such as the Internet of Things and cloud computing. The equipment of current baseband stations is designed with mixed architectures that contain both programmable (DSPs) and configurable (FPGAs) components in which custom HW blocks (IPs) are mapped. To meet the computational requirements of the above 5G use cases, the functions executed by the configurable components will change over time instead of being statically allocated. This will increase the complexity of the stations' design space and raises the need for unified tools capable to rapidly explore, partition and prototype 5G mixed designs.

At DATE 2017 University Booth, we demonstrate our latest achievements in the automatic code generation engine of TTool/DIPLODOCUS [2], a UML/SysML framework for the hardware/software co-design of data-flow systems. Our demonstration, Fig. 1, shows the full design and realization of a 5G data-link layer receiver, as defined in release 14 by the 3GPP consortium [1], for the single antenna case, in the uplink (SC-FDMA), eNodeB side. The purpose of this demonstration is to show that, at system-level of abstraction, it is possible to rapidly explore, partition and prototype mixed architecture designs (DSP-based and IP-based) from a single tool (gray area in Fig. 1).

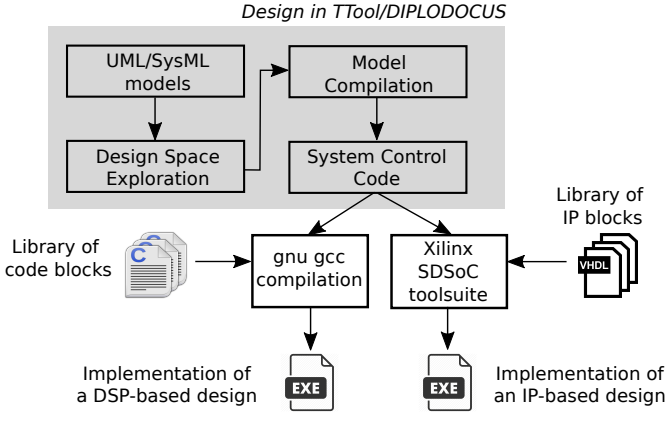

Fig. 1. Our approach for the design of 5G mixed-architecture solutions

In Fig. 1, different views (functionality, architecture, communication protocols) of a 5G data-link receiver are modeled in TTool/DIPLODOCUS with UML/SysML diagrams. Multiple design solutions are explored (Design Space Exploration in Fig. 1) and those satisfying the desired performance requirements (throughput and latency) are translated into control code.

The novelty of our demonstration resides in the model compilation phase. This produces generic control code where the execution of the receiver's data processing (e.g., FFT) and transfer (e.g., DMAs) operations is triggered via calls to a high-level C API. The latter is independent of the technological realization of a target design (e.g., DSP-based, IP-based). The code that describes the implementation of processing and transfer operations is provided by external libraries.

In the context of this demonstration, we show the effectiveness of our approach for a DSP-based design and for an IP-based design. In case of a DSP-based design, we compile the control code and the library of code blocks with a standard compiler such as `gcc`. The result is an executable that runs as an application on top of a general purpose Operating System (e.g., GNU/Linux). This executable programs the control registers of the DSP units of the target platform (Embb [3]) that runs the 5G receiver. In case of an IP-based design, we compile the control code and a library of IP blocks with the Xilinx SDSoC toolsuite [4] into an executable. The latter runs onto a general-purpose control processor's Operating System and governs the execution of the 5G receiver, where each configurable component is implemented as a separated hardware IP block. The setup of this demonstration is composed of two laptops, each connected to a Xilinx FPGA prototyping board, one for Embb and one for the hard IP-based design. Onto each laptop we have installed a complete design chain (TTool/DIPLODOCUS, the compilation and high-level synthesis tools) that will produce code for the receiver chain of our two target designs. The transmitter chain and the communication channel are, instead, emulated in MATLAB.

This work has been partially funded by the French Directorate General for Enterprise (DGE) through the European project SOOGREEN, labelled by Celtic-Plus.

References

[1] `http://www.3gpp.org/release-14`
[2] `http://ttool.telecom-paristech.fr/diplodocus.html`
[3] `http://embb.telecom-paristech.fr/`
[4] `https://www.xilinx.com/products/design-tools/software-zone/sdsoc.html`