Design and implementation of an object-oriented framework for dynamic partial reconfiguration

N. Abel, A. Beyer, F. Grüll, N. Meier, and U. Kebschull
Kirchhoff Institute of Physics, Heidelberg University, Heidelberg, Germany

Nowadays, two innovative future trends regarding embedded hardware development and hardware description can be found. The first trend concerns the hardware itself. Modern FPGAs (Field Programmable Gate Arrays) provide the possibility that parts of the configuration can be exchanged while the rest of the circuit is running untouched – which is called dynamic partial reconfiguration (DPR).

The second trend concerns the way hardware is described. Currently, the most important hardware description languages (HDLs) are VHDL and Verilog. Although they allow to describe hardware on a very high level, the developer still has to handle registers, clocks and clock domains. Using an HDL operating on the algorithmic level, this is not necessary any longer – the corresponding synthesis process is called high level synthesis (HLS).

Although both, DPR and HLS are very important future trends regarding hardware design, they develop rather independently. Most of today's software-to-hardware compilers focus on conventional hardware and therefore have to remove dynamic aspects, such as the instantiation of calculating modules at runtime. On the other hand, DPR tools work on the lowest possible layer regarding FPGAs: the bitfile level. Thus, currently the use of DPR leads to a struggle with architectural details of the FPGAs and the corresponding synthesis and implementation tools.

Our workgroup focuses on a combination of DPR and HLS, since DPR can change the programming paradigm in future HDLs with regard to dynamic instantiations. Dynamic aspects would not be a problem any longer, but could be realized on the target FPGA using DPR. Beyond that, a high level language support of DPR could help it to become a commonly used method. The aim of our work is to find a solution how HDLs on algorithmic level and DPR can be combined, solely using language constructs which are already well-known to software-developers.

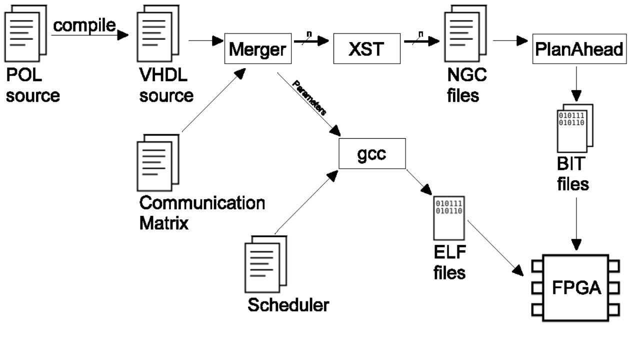

Figure 1: The implementation toolflow of the framework

As a first step, the typical structure and behavior of reconfigurable hardware has been analyzed. Thereby it turned out that the best way to describe such hardware is to make use of the object-oriented paradigm combined with multi-threading. In consequence, an enriched subset of Java, forcing the programmer to make use of multiple objects running in parallel, has been defined: POL (Parallel Object Language).

The specification of POL comes with a set of requirements. The most challenging part is the high degree of flexibility regarding object instantiation and inter-object communication. POL allows the user to instantiate and to destroy objects as well as to establish and to dissolve their connection at *any* position in the code. Beyond that, POL allows an overmapping of the FPGA.

In order to enable the evaluation of the possibilities and limitations of POL, a development framework has been implemented. This framework includes an emulator which allows the execution of POL in software [1], a compiler which is responsible for the translation from POL to VHDL [2], a so called Communication Matrix which serves as fast and flexible communication structure on the FPGA [3], and a scheduler (running on an embedded processor) that decides which hardware module is loaded (Fig. 1).

For performance analysis, an audio filter has been implemented. It shows that overmapping can be used in environments with a data rate of ∼100 000 samples/s, while scenario-based scheduling algorithms can be used in streaming applications with data rates of ∼100 000 000 samples/s. These maximum data rates are solely possible due to the usage of object-orientation in POL and the corresponding optimizations of the reconfiguration times [4].

References

[1] A. Beyer, *Development of an Emulator for the execution of POL-Code inside a Java-Environment*, bachelor thesis, University of Heidelberg, Germany, 2008

[2] F. Grüll, *The Parallel Object Language*, diploma thesis, University of Heidelberg, Germany, 2009

[3] N. Meier, *Development of a Framework for Dynamic Partial Reconfiguration serving the Object Oriented Hardware Programming Language POL*, diploma thesis, University of Heidelberg, Germany, 2008

[4] N. Abel, *Design and Implementation of an Object-Oriented Framework for Dynamic Partial Reconfiguration*, PhD thesis, University of Heidelberg, Germany