Generation of deterministic MCU/FPGA hybrid systems from UML activities

Rüdiger Willenberg, Zamira Daw, Christian Englert, Marcus Vetter

1Department of Electrical and Computer Engineering, University of Toronto, Canada
2Institute of Embedded Systems, University of Applied Sciences, Mannheim, Germany

Motivation

Model-driven engineering allows the development of stable embedded systems in the face of growing complexity. The Unified Modeling Language (UML) is the most widely used modeling language in this area. UML activity diagrams combine control and data flow semantics. They allow a behavioral description while facilitating structural partitioning. Code generation from models avoids programming errors and maintains a direct relation between code and documentation.

DMOSES Development Process

The DMOSES (Deterministic MOdels for Signal-processing Embedded Systems) process supports the development of embedded systems from modeling to implementation. Our profile extends UML semantics to allow executable models ensuring deterministic behavior. A two-step transformation converts DMOSES models into compilable code.

VHDL framework for FPGA components

The data flow modeled in activity diagrams is realized by implementing input and output pins as FIFOs. FIFOs are generated with data-type-specific dimensions, mirroring template-based pin classes in an equivalent C++ framework.

Cross-platform Modeling and Implementation

During modeling, each node in an activity is assigned to a CPU or FPGA execution platform. Correspondingly, either C++ or VHDL code is generated and compiled into target binaries.

As an example hybrid system for mixed use of C++ and VHDL components, a PowerPC system on a Virtex-5 FPGA was built. Activity edge connections between C++ and VHDL nodes are implemented through CPU bus peripherals with FIFO functionality. On the software side, Linux character devices connect as node pins to C++ nodes. Through the FIFO respectively the device driver interfaces, different types of CPU-FPGA connections can be abstracted.

Case Study: Configurable audio filter loop

An audio signal is sampled, FIR-filtered and output again as well as Fourier-transformed by VHDL components. A new coefficient set for the filter can be received via RS232 by the CPU and be entered into the filter. The spectral data won by the FFT is further processed by the CPU and displayed as a spectral bargraph on the system’s LCD display.

Conclusion and Outlook

Our design process allows comprehensive modeling of multi-platform embedded systems, including assured determinism through UML extensions, performance analysis and re-usability of C++ and VHDL components for other designs. Future work will include extension to DSP platforms and implementation of the UML state diagram, a useful semantical supplement to the activity diagram.

Acknowledgments

This research is supported by the European Union, European Regional Development Fund (ERDF) and the federal state of Baden-Wuerttemberg, Germany.