ArchitectureComposer

Alexey Kupriyanov∗, Frank Hannig† and Jürgen Teich‡
University of Erlangen-Nuremberg,
Department of Computer Science 12,
Am Weichselgarten 3, D-91058 Erlangen, Germany,
http://www12.informatik.uni-erlangen.de

Dirk Fischer§, Michael Thies¶ and Ralph Weper∥
University of Paderborn,
Department of EE and IT,
Warburger Str. 100,
D-33098 Paderborn, Germany

ArchitectureComposer is a framework for computer architecture and compiler co-generation, which provides convenient tools for semi-automized exploration of optimal architecture/compiler co-designs, and for their efficient bit-true cycle accurate fast simulation. ArchitectureComposer, ArchitectureDebugger, compiled simulator RASIM, and ArCoExplorer are the tools included in this framework, which provide complete design flow for efficient architecture/compiler co-generation and tight interchange between development stages.

In the first development stage, the architecture entry and composition is performed using the tool ArchitectureComposer. Here, architecture’s control- and data path is entered graphically using a library of customizable building blocks such as register files, memories, arithmetic and logic units, busses, etc. This tool is able to generate a cycle accurate model of the register-transfer architecture using the formalism of Abstract State Machines (ASM). At any time, the graphically entered hardware architecture can be automatically transformed into a hardware description language program. Two HDL-generators (ArchitectureComposer modules), namely, for VHDL and Verilog are available. In the near future, it is also planned to extend ArchitectureComposer’s functionality with a SystemC code generation module, which could translate the given graphically described architecture into a SystemC representation.

In the next stage, the tool ArchitectureDebugger, based on an architecture’s ASM model automatically generates a debugging and simulation environment which allows to verify the functionality of the given architecture on the pipeline-cycle accurate or instruction-cycle accurate levels. In this development stage the RASIM - compiled simulator tool also can be used in order to automatically generate an executable stand-alone cycle-accurate and bit-true "C++"-compiled fast simulator which is able to process the very fast architecture simulation with the huge stimuli input data files.

The compiler for given processor architecture can be automatically generated by the tool ArchitectureCompiler. The user initially has to enter the specification of the basic processor’s instruction set and the code-generator generator’s grammar rules. The machine model of the compiler is automatically extracted from the graphical architecture description.

And finally, the simulator generates a protocol of the simulation run which in turn is analyzed by the exploration tool ArCoExplorer. This tool is responsible for walking through the design space of architecture and compiler parameters and denotes the best architecture/compiler co-design solutions for the given application class (benchmark programs) to fulfill given constraints, e.g. code size, execution time, hardware resources, etc.

∗Contact Person and Demonstrator, phone: +49 9131 85-25156, fax: +49 9131 85-25149, email: kupriyanov@cs.fau.de
†Demonstrator, phone: +49 9131 85-25153, fax: +49 9131 85-25149, email: hannig@cs.fau.de
‡phone: +49 9131 85-25150, fax: +49 9131 85-25149, email: teich@cs.fau.de
§phone: +49 5251 60-3059, fax: +49 5251 60-3424, email: fischer@date.upb.de
¶phone: +49 5251 60-6682, fax: +49 5251 60-6697, email: mthies@upb.de
∥phone: +49 5251 60-3922, fax: +49 5251 60-4221, email: weper@date.upb.de