Automatic Tool Generation from Structural Processor Descriptions

Florian Brandner

Institute of Computer Languages
Christian Doppler Laboratory - Compilation Techniques for Embedded Processors
Technische Universität Wien

Automatic Tool Generation from Structural Processor Descriptions[1]

The success of embedded systems in mobile communication and entertainment devices, in commodity appliances, the domestic environment, as well as in the (safety) critical control systems of cars and airplanes made these small computer systems an indispensable part of every bodies daily lives. The demands on these systems in terms of reliability, efficiency, and computational power are steadily rising, while at the same time the physical dimensions and costs per unit are expected to shrink for every new product generation.

Application specific instruction processors (ASIPs) have become a valuable tool to deliver high computing power under rigid power and area constraints. The development of such a processor is delicate task that requires intimate knowledge of processor design, software development, compilers, and of course the particular problem domain.

Processor description languages – often referred to as architecture description languages (ADLs) – are a promising approach to capture the behavior, structure, and instruction set of an ASIP using a compact and concise description. These languages typically provide various views of the processor at different abstraction levels: (1) the behavioral level is primarily concerned with the abstract behavior of individual instructions, while (2) the structural view defines the hardware organization and resources. Processor description languages that focus on the former are often called behavioral languages, those following the latter approach structural. If a language combines specifications of both layers, it is called mixed. From a processor model, specified in one of these languages, software development tools, such as a compiler, linker, or assembler, can be (semi-)automatically customized for that particular processor. In addition, simulation tools, test cases, and even hardware models can be derived. Mixed languages typically provide the most flexibility in terms of these applications, because both a rather high-level, but also a rather low-level view of the processor is provided.

In this work a structural processor description language is presented that allows to derive a behavioral model from its structural specifications automatically. The language captures the behavioral and structural details of a processor and thus provides great flexibility, but avoids redundancies known from mixed languages. We demonstrate the feasibility of our approach using two generators: (1) a compiler generator that automatically derives a highly optimizing

[1] This work was supported in part by the Christian Doppler Forschungsgesellschaft and OnDemand Microelectronics.

code generator and (2) a simulator generator that derives a high-speed simulator based on dynamic binary translation. Experiments show that the derived tools can compete with hand crafted tools. The code produced by the generated compilers achieves speedups of up to 20%, on average moderate slowdowns of 5-15% have been observed. The simulation framework is similarly competitive and achieves a simulation speed that often matches the speed of the processor implementation in hardware.