Verification Challenges in Configurable Processor Design with ASIP Meister

Masaharu Imai[1] and Akira Kitajima[2]

[1] Graduate School of Information Science and Technology, Osaka University, Japan
imai@ist.osaka-u.ac.jp
[2] Department of Computer Science, Osaka Electro-Communication University, Japan
kitajima@isc.osakac.ac.jp

Abstract. In this presentation, several verification problems in configurable processor design synthesis are illustrated. Our research group (PEAS Project) has been developing a novel design methodology of configurable processor, that includes higher level processor specification description, HDL description generation from the specification, Flexible Hardware Model (FHM) for resource management for HDL generation, compiler and ISS (Instruction Set level Simulator) generation. Based on this methodology, we develop a configurable processor design environment named *ASIP Meister*.

The processor design flow using ASIP Meister is as follows: Firstly, a designer describes an instruction set architecture as a specification of a target processor including pipeline specification, instruction formats, behavior description of each instruction and interrupts, data type specification, and so on. Secondly, the designer select resources for modules to implement some functions of instructions from FHM database, that can generate various resources, such as registers, selectors, adders, shifters, etc. Thirdly, the designer describes micro-operation level behavior description with selected resources in each pipeline stages for each instruction and interrupt. Finally, HDL description of the pipeline processor and machine-depend compiler information for a retargetable compiler are generated.

One of the most important issues in such a generation based design methodology is how to keep the consistency between a given instruction set architecture specification and implementations. In the most state-of-the-art processor core generation systems, including ASIP Meister, however, there are no efficient formal methods to guarantee the correctness of a generated HDL description and compiler that implement the given specification of instruction set architecture.

We will explain several problems that are expected to be solved by applying formal verification techniques as reasonable solutions.

D. Borrione and W. Paul (Eds.): CHARME 2005, LNCS 3725, p. 2, 2005.