ASIP Meister:
An ASIP Design Environment

M. A. Hassan and Masaharu Imai
Graduate School of Information Science and Technology, Osaka University, Osaka, Japan
info@eda-meister.org
http://www.eda-meister.org/asip-meister/

1. Introduction

An Application Specific Instruction Set Processor (ASIP) is a key component in future lower power platform based architectures, whether they are control oriented or computational oriented. **ASIP Meister** meets these requirements by providing designers with an interactive environment to rapidly specify, design, and generate an ASIP. It generates not only the VHDL description of the target ASIP but also the software development tools including an assembler, compiler, and instruction set simulator. ASIP Meister has the following key features,

- A full support of various architecture parameters:
  - Variable number of pipeline stages.
  - Parameterized hardware resources which are instantiated from a Flexible Hardware Model (FHM) library.
  - Multi-cycle instructions, delayed branches, and handling of external and internal interrupts.
- Designers can input and change all architecture parameters using a graphical user interface (GUI).
- Estimation of design quality such as, area and delay of the target ASIP at an early stage of design phase.
- Both the data path and control logic of the ASIP is automatically generated from the micro-operation description of the instructions.

2. Design Flow with ASIP Meister

The design flow with ASIP Meister is as follows:
1. Set the design goals & architecture parameters of the target ASIP, such as the number of pipeline stages and instruction/data bit-width.
2. Select the hardware resources of the ASIP from FHM database library and set parameters for each resource.
3. Define ASIP storage specification & I/O interface.
4. Define the instruction type, format and exception information.
5. Check design quality with architecture-level estimation.
6. Describe the behavior of each instruction using ANSI-C like syntax.
8. Describe the micro-operation of each instruction per pipeline stage.
9. Check the performance with the generated HDL descriptions and software development tools.

3. Case Study

To demonstrate the effectiveness and flexibility of using ASIP Meister, the complete design flow of a RISC Processor architecture: DLX (32-bit, Harvard, 5-stage pipeline) would be presented. The generated synthesizable code will be mapped to an FPGA Board: Xilinx Multimedia Board (with Virtex-II XC2V2000-FF896) and different software programs execution will be shown.

4. Key Papers

Generation of HDL Description

[1] M. Itoh, S. Higaki, J. Sato, A. Shiomi, Y. Takeuchi, A. Kitajima, M. Imai, "PEAS-III: An ASIP Design Environment," 2000 IEEE International Conference on Computer Design: VLSI in Computers & Processors, pp.430-436, September 2000.

Software Tools Generation

[1] S. Kobayashi, Y. Takeuchi, A. Kitajima, M. Imai, "Compiler Generation in PEAS-III: an ASIP Development System," Proc. of SCOPES 2001, March 2001.

Work in Progress

[1] Y. Kobayashi, S. Kobayashi, K. Okuda, K. Sakanushi, Y. Takeuchi, M. Imai, "Synthesizable HDL Generation Method for Configurable VLIW Processors," Proc. of ASP-DAC 2004, January 2004.
[2] H. Tanaka, S. Kobayashi, Y. Takeuchi, K. Sakanushi, M. Imai, "A Code Selection Method for SIMD Processors with PACK Instructions," Software and Compilers for Embedded Systems, Lecture Notes in Computer Science 2826, pp.66-80, Springer, September 2003.
[3] K. Okuda, S. Kobayashi, Y. Takeuchi, M. Imai, "A Simulator Generator Based on Configurable VLIW Model Considering Synthesizable HW Description and SW Tools Generation," Proc. of SASIMI 2003, pp.152-159, April 2003.

Acknowledgement

This work was partly supported by STARC (Semiconductor Technology Academic Research Center).

Contact Person

Professor Masaharu Imai,
Graduate School of Information Science and Technology,
Osaka University
1-5 Yamadaoka, Suita, Osaka 565-0871, JAPAN
Tel:+81-6-6879-4522, Fax:+81-6-6879-4524
Email: imai@ist.osaka-u.ac.jp