Taking the Risk out of Optimizing Your Own RISC-V Architecture Design

Gert Goossens[1], Patrick Verbist[1], Dominik Auras[1]

[1]Synopsys

Abstract

The use of the RISC-V ISA is a growing trend. An important driver is the ability to customize or create RISC-V ISA and microarchitectural extensions to differentiate designs across application areas [1]. This results in application-specific processors (ASIPs) composed of a RISC-V baseline architecture, extended with specialized datapaths and storage structures. While supporting all standard RISC-V instructions, such architectures can also execute custom instructions that are encoded either in RISC-V's reserved opcode space or in additional parallel slots of an extended long instruction word.

Application-specific processors that have a RISC-V baseline provide compatibility with the RISC-V ecosystem, since they can execute software programs and libraries developed for general-purpose RISC-V architectures, and existing hardware peripherals designed for RISC-V can be reused. However, depending on the degree of specialization, much higher workloads can be executed with lower power consumption. In this sense, they can obsolete the need for hardwired accelerators that are typically used to offload standard microprocessors. Application-specific processors can offer similar acceleration but remain software-programmable, which provides for the flexibility that is needed to make post-silicon changes in the acceleration domain.

Designing application-specific processors can be challenging, for multiple reasons. First, within a large architectural search space it is not obvious to decide which extensions are best for the target application domain. Second, software developers expect to get access to a high-quality software development kit (SDK) for the application-specific processor. Third, a reliable register-transfer level (RTL) implementation of the processor must become available for silicon implementation, with excellent power, performance, and area (PPA) characteristics.

This presentation will show how electronic design automation (EDA) tools can solve these challenges. Synopsys' ASIP Designer is a leading EDA tool to design and program application-specific processors [2]. From a user-defined processor model, capturing the instruction-set and micro-architecture in the architecture description language nML, ASIP Designer automatically creates both a complete SDK with an efficient C/C++ compiler, and a synthesizable RTL implementation of the processor [3]. ASIP Designer’s unique Compiler-in-the-Loop and Synthesis-in-the-Loop methodologies enable performing software compilation and RTL synthesis runs multiple times while designing the ASIP, using the results to guide the ongoing architectural exploration. Extensive architectural iterations can be completed in only days of time, resulting in high design productivity.

ASIP Designer comes with a family of processor example models representing standardized RISC-V ISAs, which are freely available to ASIP Designer users [4]. Currently these models include support for RISC-V's I, M, C and F instruction classes. Both 32 and 64-bit variants are available, and different depths of the instruction pipeline are supported. ISA compliance has been verified using the RISC-V Compliance Framework [5]. ASIP Designer customers are successfully using these models as a baseline architecture, to which they add specialization using ASIP Designer's architectural exploration capabilities. This can range from scalar RISC-V based architectures with highly specialized functional units, over very-long instruction word architectures with a RISC-V scalar issue slot combined with any number of parallel slots encoding custom instructions, to specialized architectures with wide SIMD (single-instruction, multiple-data) processing. Such processors have been deployed by ASIP Designer users in industrial designs for diverse markets including wireless, security, automotive, and supercomputing. ASIP Designer users own the RISC-V based processor intellectual property that they create using the tool.

Other approaches have been described in literature to support the concept of extending RISC-V processors. Often the focus is on only one aspect of tool support, such as the creation of instruction-set simulators for RISC-V extensions [6], or adding custom instructions to existing C compilers [7]. Codasip’s approach integrates compiler generation, simulation and RTL generation in a single design environment [8]. However, users must define multiple processor models with different abstractions for different tools, which may introduce consistency risks. Also, the design examples described seem to be limited to scalar RISC-V processors without instruction word or data-level (SIMD) parallelism. In contrast, ASIP Designer uses a single processor model expressed in the nML language, and supports a broad architectural scope.

In this presentation, ASIP Designer's design flow for application-specific processors based on RISC-V will be presented. The approach will be illustrated with a few case studies, showing real-life examples of such processors.

ASIP Designer provides full interoperability with other EDA tools from Synopsys, to enable efficient synthesis and physical design implementation, PPA analysis, virtual prototyping, debugging, and both simulation-based and formal verification of the generated processor. Altogether, these enabling tools take the risk out of RISC-V design optimization, enabling users to differentiate their designs while maintaining compatibility and contributing to the RISC-V ecosystem.

References

[1] A. Waterman, Y. Lee, D.A. Patterson, K. Asanović, “Extending RISC-V”, Chapter 10 in: “The RISC-V Instruction Set Manual, Volume I: User-Level ISA, Version 2.1, University of California at Berkeley, May 2016.

[2] Synopsys, “ASIP Designer”, https://synopsys.com/asip.

[3] G. Goossens, “Under the Hood of ASIP Designer - Application-Specific Processor Design Made Possible by Tool Automation”, Synopsys DesignWare Technical Bulletin Q4/21, October 2021.

[4] Synopsys, “Technology Feature: Wide scope of RISC-V ASIP models ready for ASIP accelerator development’, ASIP eUpdate, Feb. 2022.

[5] “RISC-V Architectural Testing Framework”, https://github.com/riscv/riscv-compliance.

[6] K. McDermott, “Methodology for Implementation of Custom Instructions in RISC-V Architecture”, Embedded World, February 2019.

[7] A. Bradbury, “Diving into RISC-V LLVM: Supporting Custom Instruction Set Extensions”, RISC-V Workshop, Barcelona, May 2018.

[8] Codasip, “Extending RISC-V ISA With a Custom Instruction Set Extension”, https://www.design-reuse.com/articles/46237/extending-risc-v-isa-with-a-custom-instruction-set-extension.html.

Biographies

Gert Goossens is a Senior Director of Engineering at Synopsys, where he is currently leading the company’s tool development group for Application-Specific Instruction-set Processors (ASIPs). Previously, he was a co-founder and the CEO of Target Compiler Technologies, the company that pioneered the concept of ASIP tools. Gert Goossens holds MS and Ph.D. degrees from KU Leuven, Belgium.

Patrick Verbist is the Product Manager for Synopsys’ ASIP Designer tools. Previously he was a Business Development Manager and Field Application Engineer for the ASIP Designer tools and, prior to the acquisition by Synopsys in 2014, Director of Sales at Target Compiler Technologies. Before that Patrick worked for 12 years as Business Development Manager for Imec in Belgium and San Jose (US). He holds a Master’s degree in Electrical Engineering from KU Leuven, Belgium.

Dominik Auras is a Senior Applications Engineer for ASIP Tools at Synopsys. Before he joined the ASIP Tools group in 2018, he worked as a post-doctoral researcher at the Chair for Integrated Signal Processing Systems at the RWTH Aachen University. Dominik holds Dr.-Ing. (PhD) and Dipl.-Ing. (MSc) degrees in Computer Engineering from RWTH Aachen University. His university research focused on the design of ASICs for mobile receivers.