Embedded FPGA-Shell:
Emulating RISC-V Architectures at FPGA

Sajjad Ahmed[1], Elias Perdomo[1,2], Joan Teruel[1,2], Mostafa Mojahed[1], Alexander Kropotov[1], Teresa Cervero[1], Xavier Martorell[1,2], Miquel Moretto[1,2], Behzad Salami[1]

[1]Department of Computer Sciences, Barcelona Supercomputing Center
[2]Universidad Politécnica de Cataluña

Abstract

FPGA-level pre-silicon validation of RISC-V-based architectures is crucial; however, it remains a complex and challenging process. FPGA emulation can potentially become a design bottleneck due to the lack of efficient and user-friendly toolsets. To address this challenge, this paper introduces our Embedded FPGA-Shell, a highly customizable, automated, and open-source toolset that effectively facilitates FPGA-level prototyping of RISC-V architectures. Our proposal is built on AMD technology and designed for Alveo accelerator cards, supporting both UltraScale+ and Versal architectures. The fundamental idea behind this tool is simple yet effective: it automatically enables the most common peripherals out of the box, making them readily available for RISC-V cores. Additionally, the tool includes essential components for automatically generating FPGA projects with minimal user intervention. As a demonstration of its capabilities, we integrate Embedded FPGA-Shell with OpenPiton and evaluate its efficiency using multiple in-house and open-source RISC-V cores.

Introduction

The increasing demand for high performance domain specific architectures, driven by the growing adoption of the RISC-V instruction set architecture [1] requires robust multi-core systems. Scalable frameworks, e.g., OpenPiton [2] provides an opportunity to design and develop scalable RISC-V based multi-core and cache coherent systems. In this design process, pre-silicon validation is a crucial step. FPGAs are a commonly used candidate for pre-silicon validation of such large designs [3, 4], thanks to their inherent characteristics, such as reconfigurability and high-performance architecture. However, dealing with different FPGA platforms and architectures becomes laborious for RTL designers without streamlined tools.

To this end, this paper presents the Embedded FPGA-Shell, a highly-customizable, automated, and open source toolset that effectively facilitates the FPGA level prototyping of such RISC-V based architectures. The tool is a QDMA based architecture that seamlessly integrates the most-commonly used peripheral IPs into the design, including DRAM (i.e, DDR4, HBM and HBM2e), UART, JTAG@PCI, Ethernet, Telemetry, QSPI, among others. The tool is based on AMD FPGA technology supporting AMD Alveo data center accelerator cards [5] adapted on both Ultrascale+ and Versal architectures. Our tool also provides the necessary build scripts to ease the design integration process. In summary, the Embedded FPGA-Shell relieves RTL designers and developers of the FPGA emulation phase, allowing them to focus on architectural exploration instead of FPGA-level infrastructure setup. The tool is tightly coupled with OpenPiton enabling several architectures and designs.

We summarize the main contributions of this paper as follows:

- We propose a highly-customizable and easily-configurable toolset to significantly facilitate the emulation of RISC-V architecture at FPGA level by integrating the commonly used peripherals in the design. The tool is equipped with necessary tools and scripts to semi-automatically create the FPGA project with minimal manual intervention.
- We demonstrate the efficiency of our tool by integrating it with a state-of-the-art RISC-V framework, i.e., OpenPiton while adapting it to AMD Alveo Ultrascale+ (i.e., U55, U250, U280) and Versal FPGA (i.e., V80) series.
- We will open source our tool to contribute to the RISC-V community and for further developments and contributions.

Architecture

The high-level architecture of our Embedded FPGA-Shell shown in Figure. 1. As seen, the major components of the tool encompasses the peripheral IPs, toolset for automated FPGA compile, and the tight integration with the chipset module of OpenPiton framework. The key integrated peripheral IPs, mainly based on Xilinx's IP category, are PCIe block (QDMA), Ethernet, UART, Card Management Solution Subsystem

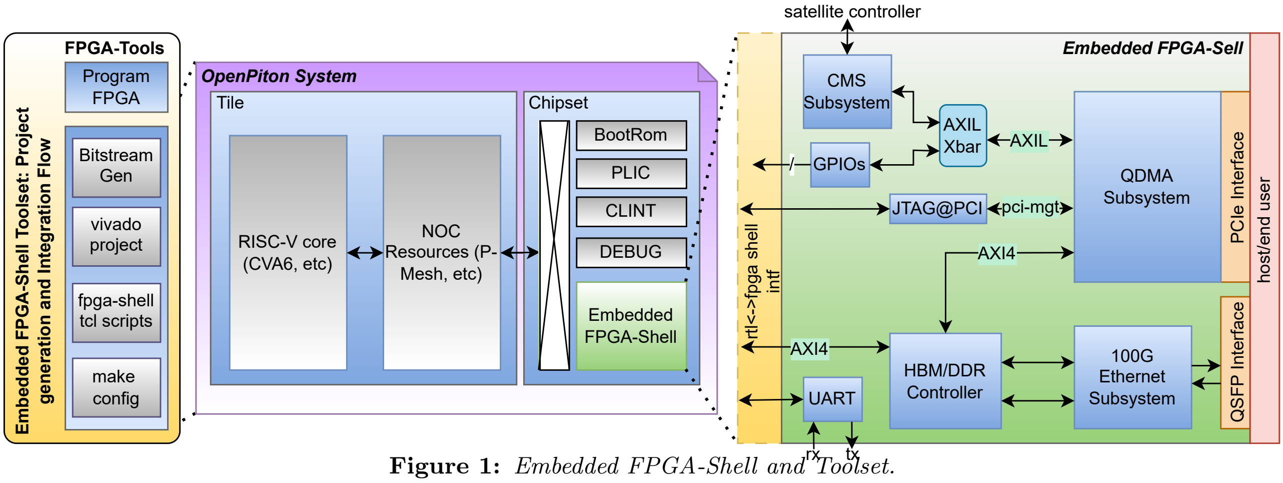

Figure 1: *Embedded FPGA-Shell and Toolset.*

Table 1: *Key integrated peripheral IPs*

Supported IPs	Features	The Role in RISC-V Emulation.
DRAM	DDR4, HBM2, HBM2e (DMA)	Main memory for RISCV based systems, stores Linux and test binaries.
Ethernet	10/100 Gb over QSFP and PCIe	Network access to RISC-V system.
Debug Bridge	JTAG over PCIe	Debug RISCV system in real time in two levels of debugging. Telnet for OpenPiton and GDB for the cores as separated targets using OpenOCD [6].
UART	Debug	Log Linux booting status, remote console for RISC-V system.
CMS	Monitoring (voltage, current, temperature and fan speed)	Telemetry/Monitors and generates reports of FPGA board characteristics and health.
Point 2 Point	MAC Layer Aurora over QSFP	Connects to other FPGAs to extend FPGA platform for larger designs.
Flash Controller	single/dual/quad data transfer rates	Stores Linux images or bootloaders that needs to be persistent across power cycles.

Table 2: *Embedded FPGA-Shell hardware utilization*

IP	LUTs (1303680)	CLB Regs (2607360)	BRAM Tile (2016)
QDMA	49887	51468	81.5
HBM	1073	874	4
Ethernet	47965	68861	29.5
Others	2650	3549	0

(CMS), DRAM controllers, JTAG@PCI with custom RTL designs over AXI4 and AXI-Lite interfaces, and QSPI Flash, as detailed in Table. 1. The tool also provides custom scripts (make files, tcl and bash scripts) for complete project creation and bitstream generation flow.

Evaluation

As part of the evaluation process, we integrated several of our in-house RISC-V cores into the OpenPiton framework and performed the emulation phase using the Embedded FPGA-Shell. In addition, we extended OpenPiton's support to previously unsupported hardware platforms, such as HPC-oriented Alveo boards, and incorporated various hardware modules, HBM, JTAG, Ethernet, and Telemetry. We verify the entire system by a multi-stage Linux boot on FPGA to confirms that our Embedded FPGA-Shell and other enhancements to integrate with OpenPiton effectively are functionally correct. For this emulation, multiple peripherals played different roles summarized in Table. 2. Also, Table. 2 contains the hardware utilization rate for the Embedded FPGA-Shell on Alveo U55c. As seen, the area overhead of the tool is minimal.

Acknowledgment

This paper is co-financed by the Barcelona Zettaescale laboratory which is financed by the Ministry for Digital Transformation and of Public Services, within the framework of the Resilience and Recovery Facility.

References

[1] RISC-V International®. *Specification Status - Home - RISC-V Tech Hub.* URL: `https://wiki.riscv.org/display/HOME/Specification+Status` (visited on 02/08/2025).

[2] Jonathan Balkind et al. "OpenPiton: An open source many-core research framework". In: *ACM SIGPLAN Notices* 51.4 (2016), pp. 217–232.

[3] Elias Perdomo et al. "Makinote: An FPGA-Based HW/SW Platform for Pre-Silicon Emulation of RISC-V Designs". In: *Proceedings of the 16th Workshop on Rapid Simulation and Performance Evaluation for Design*. 2024, pp. 29–34.

[4] Sagar Karandikar et al. "FireSim: FPGA-accelerated cycle-exact scale-out system simulation in the public cloud". In: *2018 ACM/IEEE 45th Annual International Symposium on Computer Architecture (ISCA)*. IEEE. 2018, pp. 29–42.

[5] Xilinx Inc. *ALVEO™ Portfolio Product Selection Guideby Xilinx Inc. XMP451 (v2)*. en. May 2024. URL: `https://docs.amd.com/v/u/en-US/alveo-product-selection-guide` (visited on 03/15/2022).

[6] Hubert Högl and Dominic Rath. "Open on-chip debugger–openocd–". In: *Fakultat fur Informatik, Tech. Rep* (2006).