Verification of CoreSwap: Replacing ARM Cortex-A5 with RISC-V CVA6 in ARM SoC Environment

Muhammad Hammad Bashir*[2], Umer Shahid[2], Yazan Hussnain[1], M Tahir[2], and Fatima Saleem[1]

[1]10xEngineers
[2]Department of Electrical Engineering, U.E.T Lahore

Abstract

This paper presents the verification methodology and results of the CoreSwap project, where the ARM Cortex-A5 core in ARM Educational Kit SoC was replaced with the open-source RISC-V OpenHW CVA6 core. The project demonstrates the feasibility of integrating a RISC-V core into an existing SoC ecosystem while maintaining functionality and system stability. We detail the comprehensive verification process, including core-level Architecture Compliance Tests (ACTs), manually written system-level tests, FPGA synthesis on a Kintex-7 platform, and running performance benchmarks. The verification strategy highlights the challenges and solutions to validate such a large-scale System on Chip (SoC). This seamless integration and verification of the CoreSwap underscores the potential of RISC-V in proprietary SoC environments.

Introduction

The increasing adoption of RISC-V, an open-source Instruction Set Architecture (ISA) [1], has driven significant interest in integrating RISC-V cores into existing SoC designs, offering flexibility and granular control over system customization. This paper outlines the verification methodology for CoreSwap, a project designed to swap RISC-V core in ARM-based SoC environments. Verification is essential to ensure that the RISC-V core integrates seamlessly into the existing system while meeting architectural and system-level requirements. Our contributions include the development of a comprehensive verification strategy for core- and system-level, FPGA synthesis, and the execution of performance benchmarks to validate real-world functionality.

Methodology

The CoreSwap project involved replacing the ARM Cortex-A5 core with the RISC-V CVA6 core in an ARM Educational Kit SoC. The CVA6 core [2], an open-source, 64-bit RISC-V implementation, was chosen for its compatibility with the existing SoC infrastructure. Furthermore, in replacement of ARM Generic Interrupt Controller (GIC), the Platform-Level Interrupt Controller (PLIC) was integrated to multiplex peripheral interrupts onto the external interrupt lines of the HART context, while the Core-Level Interrupt Controller (CLINT) was included to provide inter-processor interrupt (IPI) and timer functionalities. The complete block diagram is shown in Figure 1. Verification was performed at multiple levels to ensure the correctness of the CoreSwap project.

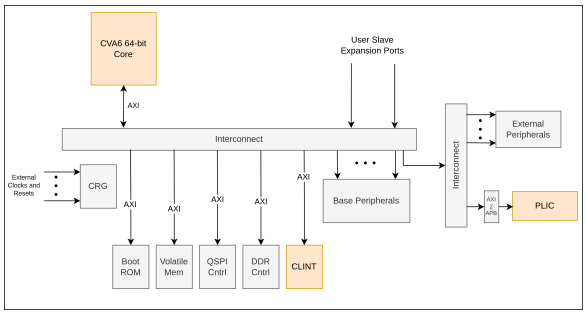

Figure 1: *High-level architecture of the modified system-on-chip (SoC) featuring a RISC-V core*

Core-Level Verification

We used the RISC-V Architecture Compliance Tests (ACT) [3] to validate the adherence of the CVA6 core to the RISC-V ISA. ACTs ensured that the CVA6 adhered to the RISC-V ISA specifications, successfully passing all mandatory test cases. These tests covered instruction and control flow operations, ensuring that the core executed the instructions correctly. Additionally, these tests verified correct memory accesses, such as loads and stores, which require the core to send requests through the network interconnect (NIC) that routes them to the appropriate memory. This process validates the proper integration of the core within the system.

*Corresponding author: hammad.bashir543@gmail.com

System-Level Verification

System-level tests were run to validate the interaction between the CVA6 core and the SoC peripherals. These self-checking tests include the verification of the memory map, interrupt handling, and correct configuration setup. The following steps outline the complete test sequence.

1. **Boot Sequence and Trap Handler setup:** The test execution starts from a designated instruction address known as the test entry point. From there, the boot sequence is initiated, involving the initialization of general-purpose registers, control and status registers, stack setup, and the trap handler.
2. **System Initialization:** After the boot sequence, the test proceeds to system initialization, which involves configuring each peripheral (e.g., UART, timers, and GPIO). This test environment requires at least one UART for logging and debugging purposes.
3. **Interrupts Configuration:** External and core-level interrupts are managed by the Platform-Level Interrupt Controller (PLIC) and the Core-Level Interrupt Controller (CLINT), respectively. This step involves configuring the interrupts, such as enabling/disabling them, setting priorities, applying masking, and setting up the corresponding interrupt handlers.
4. **Main test Entry:** The test finally enters the main function, where different test cases are executed. The following table summarizes a subset of the system-level test cases that we executed.

Table 1: *Subset of System Level Test cases*

Test Case	Test Description	Test Target
Integration Sanity Test	Generic Test for the verification of correct boot flow and system initialization.	Whole System Integration
Memory Map Test	Verifies the correct memory configurations	Memory Map
Interrupts Test	Verifies the behavior of CLINT, PLIC including generation of timer, software and external interrupts with correct handling.	CLINT, PLIC, External Peripherals

FPGA Synthesis and Validation

The modified SoC was synthesized on a Xilinx Kintex-7 FPGA to validate its functionality in a real-world environment. The synthesized SoC on the Xilinx Kintex-7 FPGA demonstrated stable operation under real-world conditions. We executed a series of performance benchmarks of Mibench suite [4] in a bare-metal environment, focusing on key metrics such as instruction throughput, memory latency, and computational efficiency in terms of Instruction per cycle (IPC). The results are shown in Figure 2. The IPC results showed that the CVA6 core achieved competitive performance. demonstrating the system's capability to perform efficiently and reliably under practical conditions.

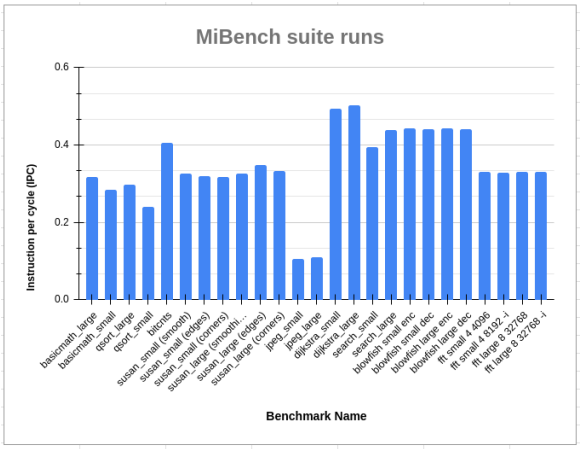

Figure 2: *Mibench suite runs on the modified system-on-chip (SoC) featuring CVA6 core*

Conclusion

The CoreSwap project demonstrates the feasibility of replacing a proprietary ARM core with an open-source RISC-V core in an existing SoC while maintaining full system functionality. Our verification strategy ensured correctness at multiple levels, from core-level ISA compliance to full system operation. This work highlights the importance of a robust verification process in enabling the adoption of RISC-V in diverse SoC environments and paving the way for future open-source architecture transitions.

References

[1] Andrew Waterman et al. *The RISC-V Instruction Set Manual, Volume I: User-Level ISA, Version 2.0*. Tech. rep. UCB/EECS-2014-54. May 2014. URL: `http://www2.eecs.berkeley.edu/Pubs/TechRpts/2014/EECS-2014-54.html`.

[2] OpenHW Group. *CVA6: An Open-Source RISC-V Out-of-Order Core*. `https://github.com/openhwgroup/cva6`. Accessed: 2025-02-06. 2024.

[3] RISC-V International. *RISC-V Architecture Test Framework*. `https://github.com/riscv-non-isa/riscv-arch-test`. Accessed: 2025-02-06. 2025.

[4] *mibench*. GitHub. Accessed: 2025-02-06. Feb. 2025. URL: `https://github.com/embecosm/mibench`.