# A Universal Hardware Emulator for Verification IPs on FPGA: A Novel and Low-Cost Approach

Saeid Jamili(✉), Antonio Mastrandrea, Abdallah Cheikh, Marcello Barbirotta, Francesco Menichelli, Marco Angioli, and Mauro Olivieri

Sapienza Universitá di Roma, Via Eudossiana, 18, 00184 Roma, RM, Italy
{saeid.jamili,antonio.mastrandrea,abdallah.cheikh, marcello.barbirotta,francesco.menichelli, marco.angioli,mauro.olivieri}@uniroma1.it

**Abstract.** Efficient and cost-effective functional verification strategies are more and more essential in digital integrated system design. This paper presents a low-cost approach to meet this challenge, introducing a universal hardware emulator designed for verifying various Intellectual Property (IP) cores on FPGA. We demonstrate the practicality and performance of this emulator through a use-case of verifying an advanced Integer Arithmetic Logic Unit (ALU) for the RISC-V ISA vector extension. The process of integration and the results obtained from the verification are presented and discussed. Preliminary findings indicate that the emulator can perform more than 1.1 million tests per second. This research contributes to the advancement of hardware verification techniques, providing researchers, universities, and small businesses with an accessible and effective solution.

**Keywords:** Hardware emulator · Verification · Co-simulation · FPGAs · UVM

## 1 Introduction

The escalating complexity of digital integrated systems compels us to adopt robust and comprehensive functional verification strategies, which are critical in assuring the dependability of the final product [1]. Recent studies have highlighted that verification efforts consume about 50% of development time [2,3]. Functional design verification is typically done via logic simulation, pre-silicon hardware emulation, or ultimately silicon prototyping. Logic simulation offers low cost, flexibility, and precision but may be slow for big designs. Emulation facilitates very extensive test patterns thanks to a much higher speed, yet the implementation of an ad-hoc emulation environment from scratch is timeconsuming and error prone. Silicon prototyping is superior in terms of speed for almost final designs, but its implementation is costly and offers limited debugging capabilities [4,5]

Hardware emulation has garnered significant attention due to its ability to accelerate verification by modelling hardware designs on a configurable devices hardware. FPGAs provide a suitable platform for the emulation of various IPs, enabling the rapid testing of new designs. Nevertheless, building a versatile and cost-effective hardware emulator on FPGA, which is able to accommodate a wide range of IPs, is highly demanding.

This work proposes a universal hardware emulator that can verify a variety of IPs on FPGA, also offering different verification modes that may involve hardware test-benches generated via High-Level Synthesis (HLS), or software test-benches running on a built-in processor on the FPGA chip, or even co-emulation with test-benches concurrently simulated on a host PC. To illustrate the utility and performance of our proposed universal emulator, we report the use-case of verifying an advanced arithmetic unit for vector processors. We demonstrate how the IP is integrated into the emulation system and present the results obtained from this exercise.

The paper concludes by elucidating the future directions for enhancing the capabilities of our emulator and exploring potential applications and improvements.

## 2 Background and Related Work

Numerous methodologies, including logic simulation, hardware emulation, prototyping, and formal verification, have been devised over the years to confront the challenge of ensuring the functional correctness of digital chip designs. Each of these approaches exhibits its unique set of advantages and limitations [4,5]. Here we refer to the works related to hardware emulation of systems or individual IP blocks.

Existing literature documents several endeavours aimed at establishing specialized emulators on FPGA platforms, geared towards specific IPs [6–8]. However, these solutions inherently suffer from a lack of versatility and universality, as they are constrained by their tailored design.

In the commercial arena, universal hardware emulators such as those belonging to Cadence Palladium series, Synopsys ZeBu and Mentor Graphics Veloce series represent the pinnacle of performance and versatility. However, these high-end emulators come with substantial financial requisites, rendering them inaccessible for many researchers, educational institutions, and small-scale enterprises.

Addressing the apparent gaps in the existing body of work, this paper seeks to design and implement a universal hardware emulator that provides a cost-effective, yet high-performance solution for verifying a diverse array of IPs on FPGA.

## 3 Proposed Universal Hardware Emulator Design

### 3.1 Design Principles

The proposed approach aims at maximizing flexibility, enabling it to universally support a diverse range of Devices Under Test (DUTs) at a low cost. The system architecture incorporates three distinct environments that we refer to as the Personal Computer (PC), Programmable Logic (PL), and Soft Processor (SP), facilitating the implementation of both the verification process and the reference model. Moreover, the environment permits the parallel verification of multiple instances of the same IP blocks. A significant portion of the system has been implemented using HLS.

### 3.2 Architecture of the Emulator

Figure 1 presents the architecture of the proposed hardware emulator. The three environments PC, PL, SP are suitably connected via high-performance interfaces to facilitate simulation, co-simulation, and hardware emulation. The system includes a Universal Verification Methodology (UVM) environment, with a reference model that can be implemented in C++. Thanks to HLS technology, this model can be placed in any of the three environments. Running the reference model on FPGA PL environment offers speed and the potential for use in system prototyping. However, one must consider limitations for large models in terms of resource requirements and library usage constraints. In the PL environment, besides running the synthesized reference model, we may also use an already verified IP as a reference. This is particularly useful when one needs to update or modify existing IPs and may run a regression test comparing with the previously verified version.

A crucial part of the system is a constrained random generator module for generating stimuli. We utilized the Linear Feedback Shift Register (LFSR) method for random generation due to its simplicity and speed. Through the control software, initial values or seeds are updated or reseeded at regular intervals to maximize randomness and prevent output signal repetition [9]. In addition, this module can generate all the signals in parallel.

A clock generator is included in the architecture. It is designed to control the clock of the DUTs, thereby supporting cycle-accurate verification. Additionally, transaction-level modeling aids in efficiently checking functionality, and an operational mode that utilizes an array of stimuli can expedite the verification process. In this work, we utilized the VCU108 Xilinx platform, and for enhancing the ease of system configuration, we developed a user-friendly Graphical User Interface (GUI) using C++ and C# for the implementation.

As a result, the presented hardware emulator integrates multi-environment verification, synthesizable reference model implementation, and parallale random stimuli generation. By leveraging HLS technology, the emulator allows for reference model deployment across these environments, further enhancing testing flexibility.

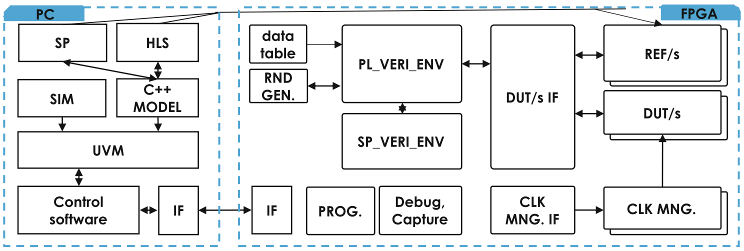

**Fig. 1.** Architecture of the proposed universal hardware emulator

## 4 Use Case: Verification of an Advanced Arithmetic Unit for Vector Processors

### 4.1 Overview

In this section, we present a use case that demonstrates the application of our proposed universal hardware emulator for the verification of an advanced Arithmetic and Logic Unit (ALU) conceived to be integrated in vector processors. The ALU was specifically designed at the Barcelona Supercomputing Center (BSC) to support the RISC-V "V" vector extension v1.0 [10]. The ALU, expressed in SystemVerilog, supports a wide range of operations including addition, subtraction, multiplication, comparison, bitwise operations, shifting, and division on data widths of 64, 32, 16 and 8-bits, operating in packed-SIMD fashion.

### 4.2 Integration of ALU Unit into Emulation Process

The integration of the ALU i nthe verification system started with the design of a behavioral reference model, implemented in C++, whose block diagram is illustrated in Fig. 2. Using the HLS tool, the C++ model was converted into RTL to be synthesized on FPGA. The verification process was carried out within the PL environment, enabling high-speed execution. The integration and setup of the system consisted of several distinct steps:

1. Defining Input/Output Ports of the DUTs.
2. Defining Test Scenarios and Signal Constraint for generating stimuli.
3. Creating and integrating the Behaviornal Reference Model.
4. Defining the Verification Process: this step consists of programming the test sequences.
5. Running, Debugging, and Reporting.

### 4.3 Results

Our experimental evaluation of the proposed emulation process involved conducting more than 37 billion tests across 25 distinct scenarios to examine the

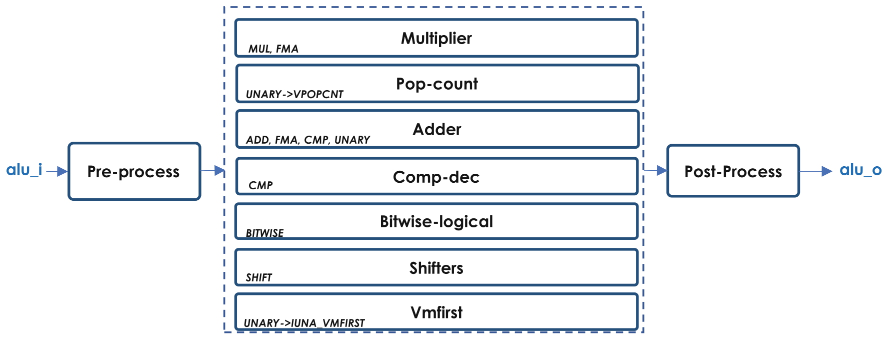

**Fig. 2.** Schematic of the advanced integer ALU unit used for verification

functional operations and control signals of the ALU, as shown in Table 1. The emulator executed approximately 1.1 million tests per second, showcasing its considerable speed. In contrast, traditional simulation methods managed about 33 thousand tests per second, showing the advantage of our emulation approach.

**Table 1.** Test results of ALU unit for 37 billion tests with 25 scenarios

| Tested Func. | Key features | | Number of tests (M) | Number of failed | Execution Time (min) |
|---|---|---|---|---|---|
| | | Ctrl. sig | | | |
| | Average: × | × | 2 K | 0 | 29 |
| ADD/SUB | | × | 4 K | 0 | 58 |
| | | ✓ | 500 | 0 | 7 |
| | fi mode: × | × | 2 K | 0 | 29 |
| Multiplier | | × | 2 K | 0 | 29 |
| | | ✓ | 500 | 0 | 7 |
| FMA | Add/sub: ✓ | × | 2 K | 0 | 34 |
| | Add/sub: ✓ | ✓ | 500 | 0 | 8 |
| Bitwise | | × | 2 K | 0 | 28 |
| | | ✓ | 500 | 0 | 7 |
| | VPOPCNT, VMFIRST | × | 2 K | 0 | 33 |
| | VPOPCNT VMFIRST \| Input src1: –1 | × | 2 K | 0 | 34 |
| | VMSIF, VMSIF, VMSIF | × | 2 K | 0 | 33 |
| Unary | VMSIF, VMSIF, VMSIF \| Input src1: 0 | × | 2 K | 0 | 34 |
| | VPOPCNT, VMFIRST | ✓ | 500 | 1.2 M | 7 |
| | VMSIF, VMSIF, VMSIF | ✓ | 500 | 0 | 7 |
| | VMSIF, VMSIF, VMSIF \| Input src1: 0 | ✓ | 500 | 750 K | 7 |
| Comp. | | × | 2 K | 0 | 31 |
| | | ✓ | 500 | 0 | 8 |
| | Left \| Round: ×\| Arith.: ×\| Out. Mode: BASE, WIDEN | × | 2 K | 0 | 27 |
| | Left \| Round: ×\| Arith.: ×\| Out. Mode: MASK | × | 2 K | 500 | 29 |
| Shifter | Out. Mode: NARROW | × | 500 | 0 | 7 |
| | Right \| Round \| Out. Mode: BASE, WIDEN | × | 2 K | 0 | 28 |
| | Right \| Round \| Out. Mode: MASK | × | 2 K | 0 | 28 |
| | Right \| Round \| Out. Mode: MASK | ✓ | 500 | 0 | 7 |

*Note* ✓ = parameter enabled; ×= disabled

## 5 Conclusion and Future Work

In this study, we introduced a hardware emulator designed to test various IPs on FPGAs. Our initial results indicated that our emulator exhibited exceptional speed. However, our testing was restricted and did not include a comparison with other emulators. For future work, We intend to evaluate our emulator using a wider range of IPs, including floating-point units, artificial intelligence IPs, and reconfigurable HW accelerators [11]. This comprehensive analysis will enable us to better understand the strengths of our emulator, as well as areas that may require improvement.

**Acknowledgment.** This work was partially funded by The European Pilot project. The European PILOT project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 101034126. The JU receives support from the European Union's Horizon 2020 research and innovation programme and Spain, Italy, Switzerland, Germany, France, Greece, Sweden, Croatia and Turkey.

## References

1. Vasudevan S (2017) Still a fight to get it right: verification in the era of machine learning. IEEE Int Conf Rebooting Comput (ICRC) 2017:1–8
2. Siemens (2022) Part 3: The 2022 wilson research group functional verification study [Online]. Available: https://blogs.sw.siemens.com/verificationhorizons/2022/10/30/part-3-the-2022-wilson-research-group-functional-verification-study/
3. Akram A, Sawalha L (2019) A survey of computer architecture simulation techniques and tools. IEEE Access 7:78120–78145
4. Scheffer L, Lavagno L, Martin G (2006) EDA for IC system design, verification, and testing (Electronic Design Automation for Integrated Circuits Handbook). CRC Press, Inc.
5. Mehta AB (2018) ASIC/SoC functional design verification. Springer, Berlin
6. Shi K et al (2023) ENCORE: efficient architecture verification framework with FPGA acceleration. In: Proceedings of the 2023 ACM/SIGDA international symposium on field programmable gate arrays (FPGA 2023). ACM, Monterey, CA, USA, pp 209–219
7. Lagos-Benites J et al (2011) An FPGA-emulation-based platform for characterization of digital baseband communication systems. IEEE Int Symp Defect Fault Toler VLSI Nanotechnol Syst 391–398
8. Tomas BJ, Jiang Y, Yang M (2014) SoC scan-chain verification utilizing FPGA-based emulation platform and SCE-MI interface. In: 2014 27th IEEE international system-on-chip conference (SOCC), pp 398–403
9. Goss C (2011) Improving verification productivity with the dynamic load and reseed methodology. Cadence Des Syst
10. Minervini F et al (2023) An area-efficient RISC-V decoupled vector coprocessor for high performance computing applications. ACM Trans Arch Code Optim 20(2):1–25
11. Jamili S et al (2022) Implementation of dynamic acceleration unit exchange on a RISC-V soft-processor. In: *International conference on applications in electronics pervading industry, environment and society*. Springer, Berlin, pp 300–306