Cycle-Accurate Verification of the Cryptographic Co-Processor for the European Processor Initiative

Pietro Nannipieri[1 [0000−0002−2538−5440]], Stefano Di Matteo[1 [0000−0002−5711−432X]], Luca Crocetti[1 [0000−0001−8504−8203]], Luca Zulberti[1 [0000−0001−9599−2652]], Luca Fanucci[1 [0000−0001−5426−4974]], and Sergio Saponara[1 [0000−0001−6724−4219]]

Deparment of Information Engineering, University of Pisa
Via G. Caruso 16, Pisa, Italy
pietro.nannipieri@unipi.it

Abstract. This paper presents a cycle-accurate verification environment for the Crypto-Tile, a cryptographic accelerator integrated into the EPI General Purpose Processor. The focus of this work is to provide a robust methodology for validating the functionality and performance of the Crypto-Tile. The verification environment includes an in-depth examination of the internal architecture and operational aspects of the Crypto-Tile, allowing for accurate modelling of hardware components and emulation of Direct Memory Access (DMA) operations. Developers can leverage this environment to simulate and verify their C-Code implementations, utilizing the functions available in the Crypto-Tile library or creating custom libraries. The verification process involves using the 32-bit AXI4 interface for communication between the processor and the Crypto-Tile while emulating DMA operations to ensure accurate testing.

Keywords: AES, ECC, RNG, SHA, RISC-V, EPI, Cryptoprocessor, Hardware, Verification, Cycle-accurate

1 Introduction

In today's interconnected world, where digital information flows seamlessly across networks, the importance of cryptography cannot be overstated. From securing online transactions and safeguarding personal information to protecting national security interests, cryptography plays a pivotal role in upholding the trust and privacy of individuals, organizations, and governments. As the reliance on computing systems continues to grow exponentially, so does the need for robust and efficient cryptographic processors [1,2,5,7,12]. In this context, the European Processor Initiative (EPI) [6] represents a groundbreaking collaborative effort among European Union member states, research institutions and industry partners to develop a cutting-edge high-performance computing ecosystem. The EPI aims to develop a new generation of energy-efficient and high-performance processors.

With digital transformation permeating every aspect of society, achieving self-reliance in processor technology has become imperative for Europe's strategic autonomy. By fostering homegrown expertise and innovation in processor design, the EPI seeks to reduce dependence on non-European technology providers, enhance Europe's technological competitiveness, and strengthen its position in the global market. Developing a secure and efficient cryptographic processor is an integral part of the EPI's broader vision, as it addresses the growing need for robust encryption capabilities to safeguard sensitive data and critical information infrastructures against ever-evolving cyber threats. When dealing with cryptographic algorithms, the computational complexity poses challenges for traditional software execution on Central Processing Units (CPUs). Cryptographic operations involve data manipulations that demand substantial processing power. The execution of these algorithms purely through software implementations can result in significant performance bottlenecks and increased execution times. To overcome these limitations, hardware acceleration emerges as a compelling solution. This approach improves the overall performance of cryptographic operations and ensures the secure processing and protection of sensitive data, making it a preferred choice in scenarios where efficient and high-performance cryptography is paramount. As a solution, the EPI system integrates the Crypto-Tile IP core, which is a dedicated hardware component for cryptography. It plays a crucial role in providing hardware acceleration for cryptographic algorithms, enabling efficient and secure encryption and decryption operations, and robust protection mechanisms for security-critical assets, such as keys. Cycle-accurate verification plays a pivotal role in the development and validation of complex hardware designs, such as the Crypto-Tile. Simulating the hardware at the cycle level enables a more granular and comprehensive analysis of the design's functionality, performance, and compliance with desired specifications. By precisely modelling the hardware behaviour, timing constraints, and interactions with software components, it becomes possible to identify and rectify potential design flaws or bugs. Additionally, cycle-accurate verification enables detailed measurement and analysis of power consumption and latency in software-accelerated hardware primitives. This work aims to provide an overview of the cycle-accurate verification environment developed to validate the Crypto-Tile co-processor. By presenting the verification approach followed using a cycle-accurate verification environment, we strive to contribute to the broader understanding of the rigorous verification processes undertaken for critical hardware components within the EPI framework.

2 The Crypto-Tile Within the European Processor Initiative

The Crypto-Tile [11] is a cryptographic accelerator designed to provide a comprehensive and versatile range of cybersecurity services with advanced security features. It incorporates various engines to support different cryptographic functions. The Advanced Encryption Standard (AES) engine [10] supports both

AES-128 and AES-256 ciphers, offering a minimum security strength of 128 bits in terms of both classical and post-quantum security. The Elliptic-Curve Cryptography (ECC) engine [4] enables operations on elliptic curves with widths of 256 and 521 bits, providing symmetric-key equivalent security strength for classical security. The Secure Hash Algorithm (SHA) engine [8] generates digests of at least 256 bits and 384 bits through SHA2 and SHA-3 functions, meeting the minimum strength requirement of 128 bits. Additionally, the Crypto-Tile features a True Random Number Generator (TRNG) engine [9] that meets the security requirements for cryptographic applications. The internal architecture

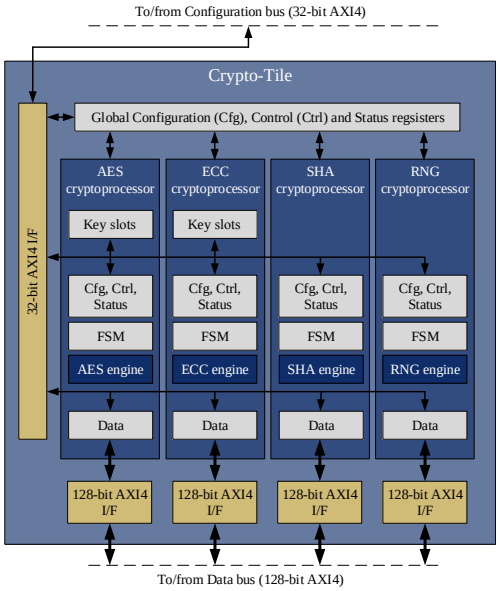

Fig. 1. Crypto-Tile Blocks Scheme

of the Crypto-Tile (Figure 1) consists of several key components. These include a 32-bit AXI4 interface, which serves as a Slave Memory-Mapped interface for accessing the registers of the Crypto-Tile through the Configuration bus. The Global Management Unit handles the global configuration, control, and status of the Crypto-Tile. Each of the four independent crypto-processors is dedicated to a specific class of cryptographic algorithms: AES [10], ECC [4], SHA [8], and Random Number Generator (RNG) [9]. These cryptoprocessors function as coprocessing units for the main processor they are connected to, such as the Secure Micro-Controller Unit (MCU) or the Secure Element (SE, a compact micro-controller that handles the first stage of the secure boot routine). Each cryptoprocessor includes local registers for configuration, control, and status and a Finite State Machine (FSM) for managing cryptographic operations. They also feature engines for hardware acceleration of cryptographic algorithms and functions. The AES and ECC cryptoprocessors incorporate local resources for key storage and management. To facilitate high-bandwidth transfers, four independent 128-bit AXI4 interfaces, one for each cryptoprocessor, provide access to the

data registers. Performances and complexity of the Cryptotile are reported in Table 1.

Table 1. Implementation results for the CryptoTile

Tech	Entity	Max Freq	Complexity
Xilinx VU37P FPGA	**Crypto_Tile**	150 MHz	27507 CLB; 144892 CLB LUTs; 93503 CLB Reg; 64 DSPs
	RNG Engine	260 MHz	2294 CLB; 10154 CLB LUTs; 7122 CLB Reg; 0 DSPs
	SHA Engine	190 MHz	3433 CLB; 10290 CLB LUTs; 10787 CLB Reg; 0 DSPs
	ECC Engine	95 MHz	15151 CLB; 79219 CLB LUTs; 37626 CLB Reg; 64 DSPs
	AES Engine	170 MHz	1253 CLB; 6036 CLB LUTs; 2460 CLB Reg; 0 DSPs
7nm Std-Cell Technology	**Crypto_Tile**	3.7 GHz	1325.16 kGE
	RNG Engine	4.325 GHz	127.16 kGE
	SHA Engine	3.725 GHz	128.32 kGE
	ECC Engine	1.525 GHz	658.90 kGE
	AES Engine	2.425 GHz	56.01 kGE

3 The Crypto-Tile RISC-V Environment

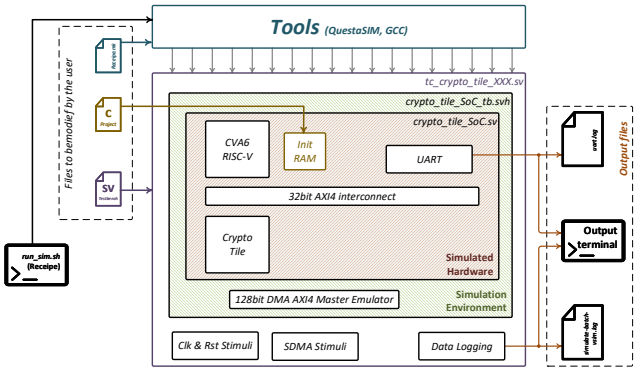

Fig. 2. Crypto-Tile Cycle-Accurate Verification Environment

The verification environment set-up is based on the one presented in [13], which exploits an automated framework for accelerating the design space exploration of hardware/software co-designs in heterogeneous digital systems. It simplifies the customization of design tools, improves designer productivity, and allows the evaluation of different hardware/software choices. The framework integrates the RISC-V toolchain and focuses on post-synthesis analyses for accurate power consumption evaluations. It helps design robust systems against Side-Channel Attacks [3], ultra-low-power and heterogeneous architectures, and space-grade Systems on Chip (SoCs) with varying technology outcomes. The environment architecture (Figure 2) relies on a makefile approach that requires three different inputs: 1) a C project (i.e. the software intended to be executed by the RISC-V processor); 2) a SystemVerilog (SV) testbench (to provide the

necessary inputs to the hardware and simulate its interface with the rest of the world); 3) a Recipe, to configure the work environment. These inputs are fed directly to the framework which processes them by compiling the C code and initializing a simulated RAM containing the compiled executable, then the SV testbench is executed to initialise the cycle-accurate verification. The testbench needs to provide service signals (e.g. clock and reset), and it may send data to the Crypto-Tile via the dedicated 128-bit DMA AXI4 emulator, achieving a data rate much higher than the one provided by the 32-bit AXI4 Internal interconnect. The test environment is then respfor collecting all the generated information, writing a log file of the simulation, and printing the UART output of the RISC-V processor into a dedicated file. To use the verification environment user shall perform the following steps:

1) Write the C code to be executed, by exploiting the provided templates and referring to the Crypto-Tile documentation for details on drivers, register addresses, operation modes et al. to use the functions available in the Crypto-Tile library or by creating an own library. It is noted that the C code provides access only to the 32-bit AXI4 MCU interface on the processor. The DMAs are emulated in System Verilog and cannot be accessed via the C code. However, this limitation will be addressed with the Field Programmable Gate Array (FPGA) system. All the C source files must be added as new targets in the makefile, indicating the main C file (*<c_test>.c*) as the main target (*<c_test>*).

2) Write the SV testbench, which must include at least the clock (*clk*) and reset (*rst*) signals, Also DMA operations can be included by exploiting the System Verilog Secure DMA (SDMA) functions for the AXI4 Master emulation. In this no, no default synchronization mechanisms between the RISC-V processor and the SDMA interfaces are provided, hence they can be used in the Crypto-Tile's internal signals, such as Interrupt Request (IRQ) signals. The name of the SV testbench file (e.g., *<tc_crypto_tile_ID>.sv*) constitutes the top-level and must be specified in the recipe (next step).

3) Run the simulation using the corresponding command. The logs of the compilation process and the simulation are made available in a dedicated folder (*build*), and they can be used to check errors and information.

4 Simulations

Thanks to the simulation environment provided, we were able to establish a comprehensive test plan that stimulates the entire system by focusing on each developed accelerator and service system. Below, we provide a concise overview of the tests performed in each test category:

- **AXI Interfaces:** This test category consists of 20 diverse tests that aim to thoroughly test the AXI communication infrastructure. The focus is mainly on the MCU AXI interconnect and each of the AXI DMA (one for each Cryptoprocessor). All supported AXI operations undergo comprehensive testing.
- **Global Management Unit:** This test category comprises 58 distinct tests that aim to execute write and read attempts on the register file. The Crypto-Tile security specification governs the register file. These tests highlight pos-

sible error situations and evaluate the security of the system against unauthorized attempts to access protected data (both read and write).
- **AES Cryptoprocessor:** This category of tests focuses on the AES cryptoprocessor and includes 138 different tests that need to be executed. These tests cover all the different AES operative modes such as ECB, OFB, CFB, CTR, XTS, CMAC, GCM, and CCM. Additionally, they involve the writing and reading of the cryptographically secured configuration and status register.
- **ECC Cryptoprocessor:** This test category focuses on the ECC cryptographic processor with 62 different tests to be executed. The writing and reading of secured configuration and status registers are tested intensively, together with all the supported ECC engine operational modes.
- **SHA Cryptoprocessor:** This category of tests is centred around the SHA cryptographic processor and includes 63 different tests that need to be performed. There is an extensive focus on testing the secure configuration and status registers' reading and writing, along with both SHA3 and SHA-2 operational modes, each with all the supported key sizes.
- **RNG Cryptoprocessor:** The final category of tests is centred around the RNG cryptoprocessor and includes 45 different tests that need to be performed. These tests heavily evaluate the writing and reading of secured configuration and status registers, as well as the various operational modes, such as changing the entropy source and number generation modes.

5 Conclusions

This work presented a comprehensive cycle-accurate verification environment for the Crypto-Tile, a state-of-the-art cryptographic accelerator integrated into the EPI system. The verification environment provides a systematic approach for validating functionality and performance, by accurately modelling the behaviour of the hardware components and emulating the DMA operations in System Verilog. Developers can leverage this verification environment to simulate and verify their C-Code implementations in conjunction with the Crypto-Tile. The AXI4 MCU interface is the communication channel between the processor and the Crypto-Tile, while DMA operations are emulated to ensure accurate testing. Simulations can be performed by executing the designated command within the verification environment. The generated simulation logs, accessible in the specified *build* folder, facilitate result analysis and troubleshooting in case of compilation or simulation errors.

Acknowledgemnt

This work was partially funded by the European Union's Horizon 2020 research and innovation programme "European Processor Initiative" (grant agreement No. 101036168, EPI SGA2) and partly supported by the Italian Ministry of University and Research (MUR) with the project CN4 - CN00000023 of Recovery

and Resilience Plan (PNRR) program, grant agreement No. I53C22000720001, and in the framework of the FoReLab project (Departments of Excellence).

References

1. Intel Software Guard Extensions (Intel SGX) – Key Management on the 3rd Generation Intel Xeon Scalable Processor. Tech. rep., Intel (August 2019)
2. Coppolino, L., D'Antonio, S., Mazzeo, G., Romano, L.: A comprehensive survey of hardware-assisted security: From the edge to the cloud. Internet of Things **6**, 100055 (2019)
3. Crocetti, L., Baldanzi, L., Bertolucci, M., Sarti, L., Carnevale, B., Fanucci, L.: A simulated approach to evaluate side-channel attack countermeasures for the Advanced Encryption Standard. Integration **68**, 80–86 (September 2019)
4. Di Matteo, S., Baldanzi, L., Crocetti, L., Nannipieri, P., Fanucci, L., Saponara, S.: Secure Elliptic Curve Crypto-Processor for Real-Time IoT Applications. Energies **14**(15) (2021)
5. Gupta, S.: An edge-computing based Industrial Gateway for Industry 4.0 using ARM TrustZone technology. Journal of Industrial Information Integration **33**, 100441 (2023)
6. Kovač, M., et. Al: European Processor Initiative: Europe's Approach to Exascale Computing (2022)
7. McKeen, F., Alexandrovich, I., Berenzon, A., Rozas, C.V., Shafi, H., Shanbhogue, V., Savagaonkar, U.R.: Innovative Instructions and Software Model for Isolated Execution. vol. 10 (June 2013)
8. Nannipieri, P., Bertolucci, M., Baldanzi, L., Crocetti, L., Di Matteo, S., Falaschi, F., Fanucci, L., Saponara, S.: SHA2 and SHA-3 accelerator design in a 7 nm technology within the European Processor Initiative. Microprocessors and Microsystems **87** (2021)
9. Nannipieri, P., Di Matteo, S., Baldanzi, L., Crocetti, L., Belli, J., Fanucci, L., Saponara, S.: True Random Number Generator Based on Fibonacci-Galois Ring Oscillators for FPGA. Applied Sciences (Switzerland) **11**(8) (2021)
10. Nannipieri, P., Matteo, S., Baldanzi, L., Crocetti, L., Zulberti, L., Saponara, S., Fanucci, L.: VLSI Design of Advanced-Features AES Cryptoprocessor in the Framework of the European Processor Initiative. IEEE Transactions on Very Large Scale Integration (VLSI) Systems **30**(2), 177–186 (2022)
11. Nannipieri, P., Crocetti, L., Matteo, S.D., Fanucci, L., Saponara, S.: Hardware Design of an Advanced-Feature Cryptographic Tile within the European Processor Initiative. IEEE Transactions on Computers pp. 1–14 (2023)
12. Pinto, S., Santos, N.: Demystifying Arm TrustZone: A Comprehensive Survey. ACM computing surveys (CSUR) **51**(6), 1–36 (2019)
13. Zulberti, L., Di Matteo, S., Nannipieri, P., Saponara, S., Fanucci, L.: A Script-Based Cycle-True Verification Framework to Speed-Up Hardware and Software Co-Design: Performance Evaluation on ECC Accelerator Use-Case. Electronics (Switzerland) **11**(22) (2022)