Towards EXtreme scale Technologies and Accelerators for euROhpc hw/Sw Supercomputing Applications for exascale

WP2 New accelerator designs exploiting mixed precision

D2.4 eXtreme Secure Crypto IP, part 1

Revised version

http://textarossa.eu

This project has received funding from the European Union's Horizon 2020 research and innovation programme, EuroHPC JU, grant agreement No 956831

TEXTAROSSA

Towards EXtreme scale Technologies and Accelerators for euROhpc hw/Sw Supercomputing Applications for exascale

Project Start Date: 01/04/2021 **Duration**: 36 months
Coordinator: *AGENZIA NAZIONALE PER LE NUOVE TECNOLOGIE, L'ENERGIA E LO SVILUPPO ECONOMICO SOSTENIBILE - ENEA, Italy.*

Deliverable No	D2.4 (revised)
WP No:	WP2
WP Leader:	CINI-UNIPI
Due date:	M18 (September 30, 2022),
Delivery date:	10/10/2022, revised May 21, 2023

Dissemination Level:

PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Grant Agreement No.: 956831

Deliverable: D2.4 eXtreme Secure Crypto IP, part 1

DOCUMENT SUMMARY INFORMATION

Project title:	Towards EXtreme scale Technologies and Accelerators for euROhpc hw/Sw Supercomputing Applications for exascale
Short project name:	TEXTAROSSA
Project No:	956831
Call Identifier:	H2020-JTI-EuroHPC-2019-1
Unit:	EuroHPC
Type of Action:	EuroHPC - Research and Innovation Action (RIA)
Start date of the project:	01/04/2021
Duration of the project:	36 months
Project website:	textarossa.eu

WP2 New accelerator designs exploiting mixed precision

Deliverable number:	D2.1
Deliverable title:	eXtreme Secure Crypto IP, part 1
Due date:	M18
Actual submission date:	M27 (revised)
Editor:	Sergio Saponara
Authors:	S. Saponara, S. Di Matteo
Work package:	WP2
Dissemination Level:	Public
No. pages:	23, 25 pages in the revised version
Authorized (date):	10/10/2022, revised version 15/5/2023

Responsible person:	Sergio Saponara					
Status:	Plan	Draft	Working	Final	Submitted	Approved

Revision history:

Version	Date	Author	Comment
0.1	2022-09-30	S. Saponara	Draft structure
0.2	2022-10-04	S. Di Matteo	First version completed
0.3	2023-05-05	S. Saponara, S. Di Matteo	Revised according to the reviewer comments

Quality Control:

Checking process	Who	Date
Checked by internal reviewer	Carlos Alvarez	October 10th, 2022
Revised checked by internal reviewer	Daniele Gregori	May 8th, 2023
Checked by Task Leader	Sergio Saponara	October 4th, 2022
Revised checked by Task Leader	Sergio Saponara	May 10th, 2023
Checked by WP Leader	Sergio Saponara	October 4th, 2022
Revised checked by WP Leader	Sergio Saponara	May 10th, 2023
Checked by Project Coordinator	Massimo Celino	October 10th, 2022
Revised checked by Project Coordinator	Massimo Celino	May 15th, 2023

COPYRIGHT

ACKNOWLEDGEMENTS

This project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement no 956831. The JU receives support from the European Union's Horizon 2020 research and innovation programme and Italy, Germany, France, Spain, Poland.
Please see http://textarossa.eu for more information on the TEXTAROSSA project.
The partners in the project are AGENZIA NAZIONALE PER LE NUOVE TECNOLOGIE, L'ENERGIA E LO SVILUPPO ECONOMICO SOSTENIBILE (ENEA), FRAUNHOFER GESELLSCHAFT ZUR FOERDERUNG DER ANGEWANDTEN FORSCHUNG E.V. (FHG), CONSORZIO INTERUNIVERSITARIO NAZIONALE PER L'INFORMATICA (CINI), INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET AUTOMATIQUE (INRIA), BULL SAS (BULL), E4 COMPUTER ENGINEERING SPA (E4), BARCELONA SUPERCOMPUTING CENTER-CENTRO NACIONAL DE SUPERCOMPUTACION (BSC), INSTYTUT CHEMII BIOORGANICZNEJ POLSKIEJ AKADEMII NAUK (PSNC), ISTITUTO NAZIONALE DI FISICA NUCLEARE (INFN), CONSIGLIO NAZIONALE DELLE RICERCHE (CNR), IN QUATTRO SRL (in4). Linked third parties of CINI are POLITECNICO DI MILANO (CINI-POLIMI), Università di Torino (CINI-UNITO) and Università di Pisa (CINI-UNIPI); linked third party of INRIA is Université de Bordeaux; in-kind third party of ENEA is Consorzio CINECA (CINECA); in-kind third party of BSC is Universitat Politècnica de Catalunya (UPC).
The content of this document is the result of extensive discussions within the TEXTAROSSA © Consortium as a whole.

DISCLAIMER

Table of contents

List of Acronyms

Acronym	Definition
ALU	Arithmetic Logic Unit
ASIC	Application Specific Integrated Circuit
AXI	Advanced eXtensible Interface (AXI)
CINI	Consorzio Interuniversitario Nazionale per l'Informatica
CKKS	Cheon-Kim-Kim-Song
CPU	Central Processing Unit
CRYSTALS	CRYptographic SuiTe for Algebraic LatticeS
FPGA	Field Programmable Gate Array
HE	Homomorphic Encryption
HW	Hardware
HPC	High-Performance-Computing
IP	Intellectual Property
IPR	Intellectual Property Rights
JTAG	Joint Test Action Group
XOF	eXtendable Output Function
MPSOC	Multi-Processor SoC
NIST	National Institute for Standard and Technology
PQC	Post Quantum Cryptography
RISC	Reduced Instruction Set Computer
RLWE	Ring Learning With Errors
SE	SEAL Embedded
SEAL	Simple Encrypted Arithmetic Library
SOC	System On Chip
SW	Software
UART	Universal Asynchronous Receiver Transmitter interface
VHDL	VHSIC Hardware Description Language

Executive Summary

This document reports the activities done by TEXTAROSSA partners CINI (UNIPISA), with reference to preliminary HDL design, verification, and synthesis of accelerator IPs in WP2 for cryptography.

Aiming at giving hardware acceleration to advanced secure services, not already covered by EPI1 activities, the work in TEXTAROSSA implements and verifies in FPGA technology:

- a hardware accelerator for Homomorphic Encryption, having as benchmark the SEAL-Embedded Library provided by Microsoft.

- a hardware accelerator for eXtendable Output Functions (XOF) SHAKE-128/256, a new hashing functions used with the algorithms like Crystals-Dilithium recently standardized by NIST for digital signature generation/verification compliant with a Post-Quantum Cryptography (PQC) scenario.

The Secure Crypto IPs are designed according to the specifications defined in the revised D2.1.

Both IPs have been provided with a standard memory-mapped AXI4 interface, so that they can be easily interfaced to any other digital IPs. To this aim results of the integration of the two proposed secure IPs with open source RISC-V 32-bit and RISC-V 64-bit processors are shown.

The repository of the IP is also reported and for each of the two IP a scientific submission to an IEEE journal has been done.

1. Introduction

This document D2.4 reports the activities done by TEXTAROSSA partner CINI (UNIPISA) in WP2.

D2.4 deals with the preliminary HDL design, using SystemVerilog, verification and synthesis of accelerator IPs for cryptography. The Secure Crypto IPs are designed according to the specifications defined in the revised D2.1.

Aiming at giving hardware acceleration to advanced secure services, not already covered by EPI1 activities, the work in TEXTAROSSA implements and verifies:

- Hardware accelerator for Homomorphic Encryption, having as benchmark the SEAL Library provided by Microsoft [1,4,5,11] addressed in Section 2.

- Hardware accelerator for eXtendable Output Functions (XOF) SHAKE-128/256, a new hashing functions used with the algorithms like CRYSTALS -Dilithium [14] recently standardized by NIST [15] for digital signature generation/verification compliant with a Post-Quantum Cryptography (PQC) scenario. This is discussed in Section 3.

The two IPs have been provided with a standard memory-mapped AXI4 interface, so that they can be easily interfaced to any other digital IPs. To this aim results of the integration of the 2 proposed secure IPs with open source RISC-V 32bits [6] and RISC-V 64bits [7] processors are shown.

Implementation results in FPGA technology are shown in Sections 2 and 3 (in TEXTAROSSA FPGA is the target technology for hardware design and to characterize and verify the IP macrocells, since the design of Application Specific Integrated Circuit-ASIC- is not foreseen).

Conclusions and the repository of the IPs are reported in Section 4 and for each of the two IP a scientific submission to an IEEE journal has been done.

2. Accelerator IP for Homomorphic Encryption

This chapter highlights the hardware design and the implementation results on FPGA of the following cryptography function and service: Homomorphic Encryption (HE) in the context of communication between edge devices and a cloud server where the goal is ensuring data privacy to the many users accessing the server from edge devices. As already discussed in Section 3.2 of D2.1, this is a new feature, complementary to the security features present in EPI1.

HE is a form of encryption that allows computations to be performed on ciphertexts, without decrypting them first. This makes it possible to perform computations on sensitive data while keeping it encrypted, which can be useful in applications where privacy and security are paramount. Some of the main HE libraries and their application are:

- Microsoft SEAL [1]: an open source HE library developed by Microsoft Research. It is used for secure cloud computing, secure data sharing and secure machine learning.
- PALISADE [2]: an open source library developed by New Jersey Institute of Technology for secure computation of financial data and secure machine learning.
- HELib [3]: is a HE library developed by IBM research.

Among them, the SEAL library is considered as a reference for the design of homomorphic accelerators since it is adopted by big players of the high-performance computing market like Intel and Nvidia. The SEAL library can be downloaded at: https://github.com/Microsoft/SEAL.

The Intel homomorphic encryption toolkit is optimized for Intel AVX-512 instruction set, https://www.intel.com/content/www/us/en/developer/tools/homomorphic-encryption/overview.html.

NVIDIA that included homomorphic encryption based on SEAL on the framework for federated learning in health applications called CLARA https://developer.nvidia.com/blog/federated-learning-with-homomorphic-encryption/.

2.1. Homomorphic Encryption SEAL-embedded library and benchmarks on RISC-V

Homomorphic Encryption (HE) is a specialized type of encryption that allows specific computations on the encrypted data and generates a cyphertext that, once decrypted, matches the result of operations performed on the plaintext data. HE is nowadays considered a strong privacy-preserving solution that allows users to share data with clouds or any non-secure server.

However, HE requires high computational resources and memory consumption, which limits its use in resources constrained IoT devices. All the HE libraries presented before are not specifically designed for resources-constrained devices. The SEAL-Embedded (SE) library [4] is the first HE library targeted for embedded devices that employs several optimizations to perform the encoding and encryption of data, featuring the Cheon-Kim-Kim-Song (CKKS) HE scheme.

SE follows the lattice-based algorithm Ring Learning With Errors, which states that given R_Q^n ring of integer modulo *Q* with degree less than *n* and given an error distribution χ, the ciphertext can be computed as a couple of polynomials *(a, b)* such that $b = a \cdot s + e \ (mod\ Q)$ where $e \in \chi, a \in R_Q$ and s is the secret polynomial (it assumes the meaning of secret key). Retrieving the secret *s* is considered hard even for quantum computers. In SE polynomial degree *n* is chosen as a power of two. Following the RLWE algorithm on SE all elements are polynomials represented as *n*–length vectors of their unsigned integer coefficients, whose values may vary in the range of $[0, Q-1]$.
SE encrypts data following the CKKS scheme, allowing encryption over floating point values.

The two main functions of the SE library are:

- Encoding: since encryption and decryption work on polynomial rings it is necessary to convert the floating-point message into unsigned integer polynomial without information loss.
- Encryption: that follows the RLWE encryption.

In this work the focus for the hardware acceleration is the symmetric encryption, which has been evaluated as the main bottleneck of the SE library.
In SE, the ciphertext is evaluated as a couple of vectors of 32-bits unsigned integers such that:

$$c_0 = -a \cdot s + m + e$$
$$c_1 = a$$

where *a* is randomly sampled from a uniform distribution, *e* is sampled from a centered binomial distribution, *s* is the secret key and *m* is the message to be encrypted (already encoded from floating-point to unsigned integer polynomial).

The following chapters will show first the results of the benchmark campaign carried out using RISC-V processors on FPGA technology of the symmetric encryption function of the SE library, and next the design strategy and the implementation results of the hardware accelerator.

Benchmark on RISC-V CPUs

The source code of the SEAL-Embedded library can be found in [5]. Two different RISC-V processors have been selected for the benchmark campaign, and two different environments have been implemented on the FPGA Board Zynq UltraScale+ MPSoC ZCU106 equipped with the System-on-Chip (SoC) XCZU7EV-2FFVC1156. Figure 2-1 shows the proposed hardware systems running the benchmark. The selected RISC-V processors are:

- The 32-bit RISC-V RISCY, whose HDL code can be downloaded in [6]. The left side of Figure 2-1 shows the complete system implemented in the target FPGA which encompasses the RISCY CPU, 256KB of on-chip memory, and AXI4 peripherals (i.e. JTAG and serial UART interface).
- The 64-bit RISC-V CVA6, whose HDL code can be downloaded in [7]. The right side of Figure 2-1 shows the complete system implemented in the target FPGA which includes the CVA6 CPU, 512MB of memory (i.e. onboard DDR4), and AXI4 peripherals (i.e. JTAG and serial UART interface).

Both systems run at 100 MHz of frequency on the target FPGA.

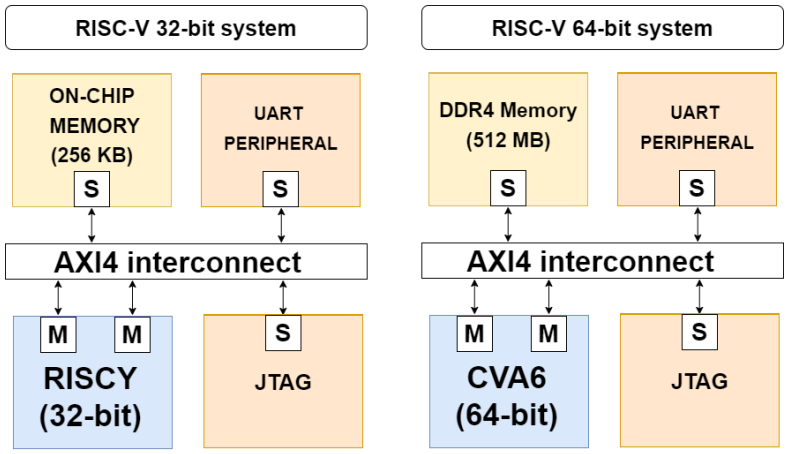

Figure 2-1: RISCV-based systems for benchmarking the SEAL-Embedded library.

Table 2-1 shows the benchmark results of the symmetric encryption function of the SEAL Embedded library on the selected CPUs.

Poly-Degree	Msg size	CVA6 (64-bit)	RISCY (32-bit)
1024	2048 B	17.19 ms	207.10 ms
2048	4096 B	37.09 ms	444.22 ms
4096	8192 B	273.80 ms	2806.43 ms
8192	16384 B	1184.19 ms	--
16384	32768 B	5861.02 ms	--

Table 2-1: Benchmark results for the encryption function of the SEAL-Embedded Library. Column 1 indicates the selected polynomial degree for the RLWE encryption, column 2 indicates the message size in Bytes, column 3 shows the results for the CVA6 processor and column 4 the results for the RISCY processor. Both CPUs run at 100 MHz.

Despite the SEAL-Embedded being targeted for resource-constrained devices, it cannot be successfully executed on the RISCY CPU for Poly-Degree higher than 4096 (256KB of memory are not enough).
In addition, the latency for the encryption process is extremely high: around 3 seconds are required to encrypt 8 KB with 4096 Poly-Degree.
Hence, a hardware accelerator has been designed in Section 2.2 and implemented in FPGA technology.

2.2. Hardware accelerator IP design, interfacing to host processor and implementation results on FPGAs

The following are the main specifications of the hardware accelerator for homomorphic encryption defined:

- Hardware acceleration of the symmetric encryption function of the SEAL-Embedded library. All the poly-degrees must be supported.
- Encryption latency around hundreds of milliseconds;
- Standard AXI4 memory-mapped interface; desiderably a DMA interface.

The target for the hardware acceleration is the RLWE encryption function $c_0 = -a \cdot s + m + e$, where *a* is randomly sampled from a uniform distribution, *e* is sampled from a centered binomial distribution, *s* is the secret key and *m* is the message to be encrypted. Figure 2-2 reports the hardware-software partitioning of the aforementioned function.
Considering the overhead caused by polynomial multiplication, in SE this operation is optimized utilizing the Number Theoretic Transform (NTT), computed following the Harvey Butterfly operations. In a polynomial multiplication between two (*n–1*) degree polynomials, evaluating the NTT of both allow to multiply their coefficient in a point-wise manner, reducing the complexity of the polynomial multiplication from $\mathrm{O}(n^2)$ to $\mathrm{O}(\mathrm{nlogn})$.

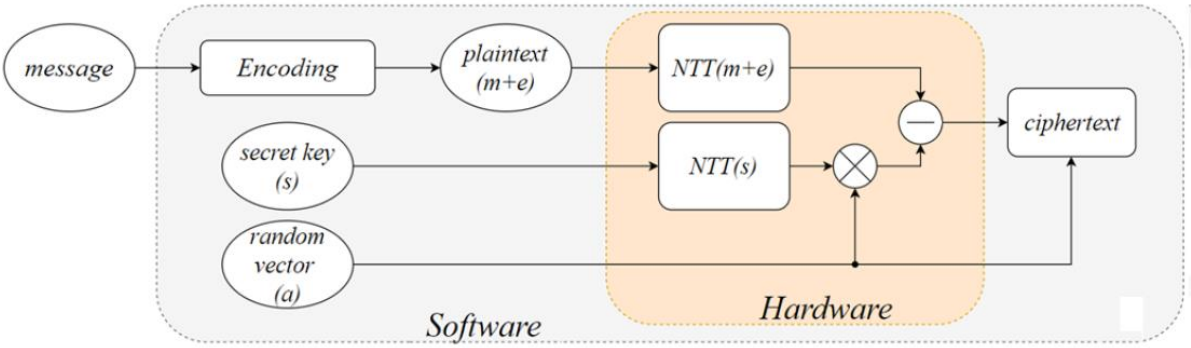

Figure 2-2: Hardware-software partitioning for the acceleration of symmetric encryption function of the SE library.

NTT is a specialized form of the Discrete Fourier Transform except that operates over a prime field instead of a complex field. Given a polynomial $a = (a_0, a_1, \dots, a_{n-1}) \in R_q$ and the primitive root of unity ω, the NTT outputs a vector $A = (A_0, A_1, \dots, A_{n-1})$ following the equation:

$$A_i = \sum_{j=0}^{n-1} a_j \omega^{ij}$$

where $0 \leq j \leq n$. The multiplication between two polynomials *a* and *b* becomes:

$$a \cdot b = \left(1, \omega^{-1}, \dots, \omega^{-(n-1)}\right) \bullet NTT^{-1}(\, NTT(A) \bullet NTT(B)\,)$$

where • indicates the coefficient-wise multiplication. Operatively, the NTT is executed through butterfly-operations among polynomial coefficients and $(n-1)$ powers of a primitive root of unity ω (denoted as twiddle factors). The Harvey Butterfly configuration (reported in Figure 2-3) is used.

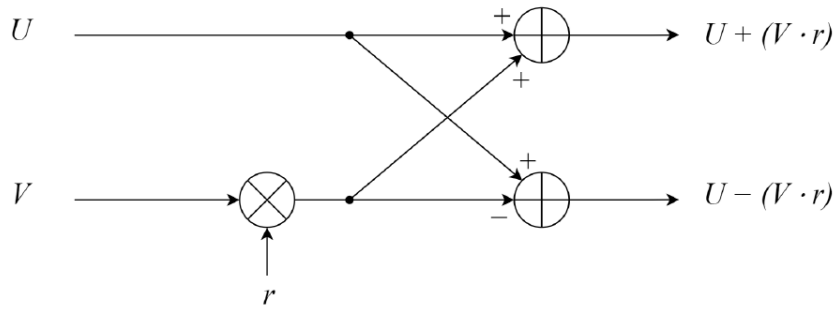

Figure 2-3: Harvey Butterfly configuration. The parameter *r* indicates the twiddle factors, U and V are the polynomial coefficients.

Figure 2-4 reports the overall architecture of the hardware accelerator, and the main blocks are:

- An AXI4 Slave interface (indicated as AXI Slave in Figure 2-4) that can be used to communicated with both the CPU and the DMA. The data size is 32-bit and the address size is 18-bit.
- The interface registers (indicated as Decoding Logic in Figure 2-4) that allows to configure and to check the status of the hardware accelerator.
- The CKKS encryption (indicated as CKKS Encryption in Figure 2-4) module that executes the RLWE encryption algorithm.
- A finite state machine (indicated as CKKS Core Fsm in Figure 2-4) to manage the operation flow.

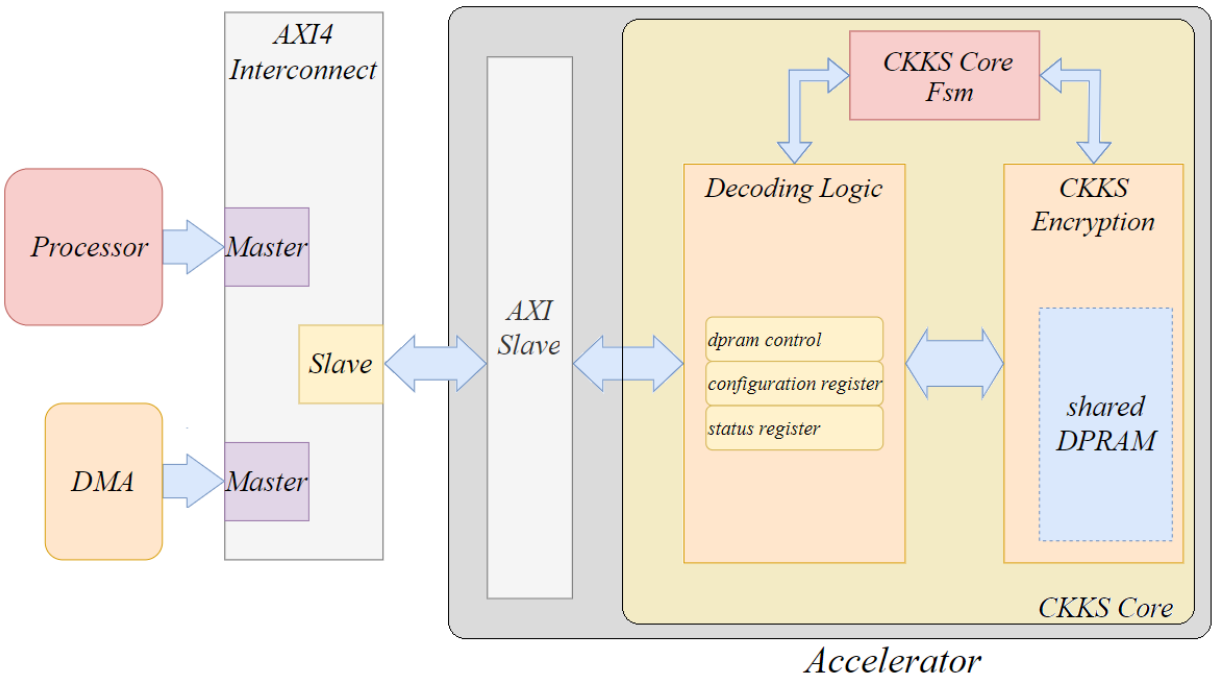

Figure 2-4: Overall architecture of the hardware accelerator for SE.

The internal architecture of the CKKS Encryption module is reported in Figure 2-5; it features the following hardware modules:

- The shared DPRAM, which is a Dual-Port RAM shared among the CPU/DMA and the hardware accelerator. It is used to write the polynomials *a, s* and *(m+e)* and to read the ciphertext.
- DPRAM1 and DPRAM2, which are used to store the polynomial coefficients during the computation of NTT and RLWE encryption.
- ALU Butterfly, which is an Arithmetic Logic Unit (ALU) performing modular addition, subtraction, multiplication and the Harvey Butterfly configuration.
- The Root Generator module and the Root RAM to respectively compute and store the powers of the primitive root of unity ω.

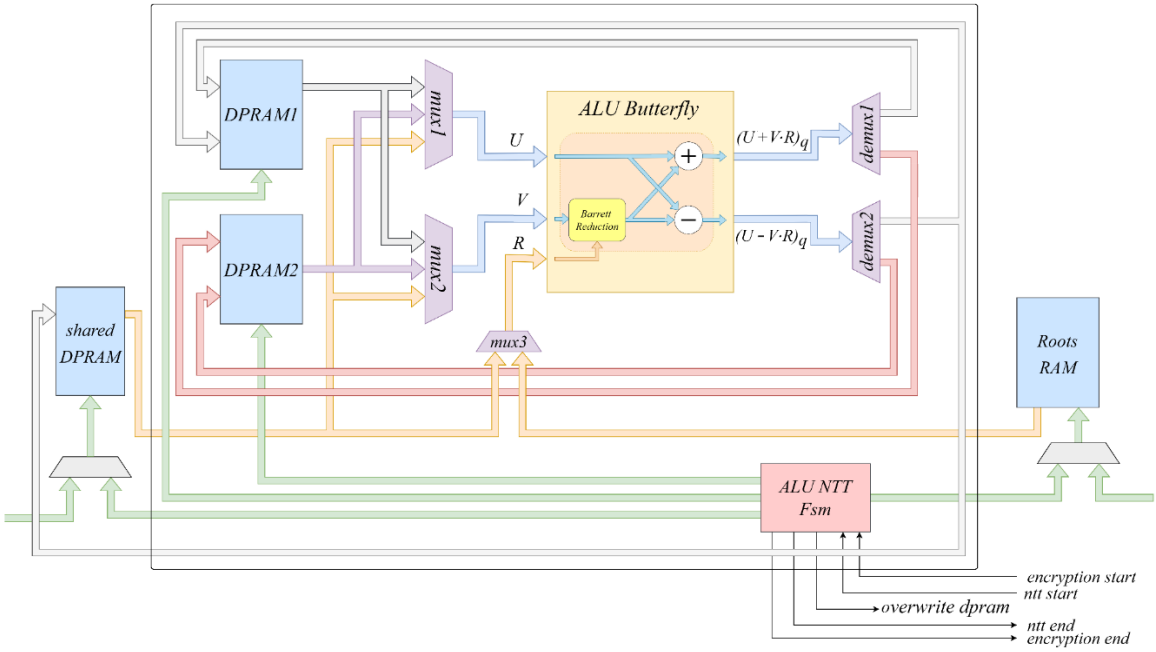

Figure 2-5: Internal architecture of the CKKS encryption module.

The hardware accelerator for SE has been implemented on the FPGA Board Zynq UltraScale+ MPSoC ZCU106 equipped with the System-on-Chip (SoC) XCZU7EV-2FFVC1156.
The block design implemented in the EDA tool Vivado 2020.2 is reported in Figure 2-6. The system includes the RISCY processor (running at 100 MHz), the Xilinx Central DMA, an AXI4 interconnect logic and standard peripherals (UART, JTAG).

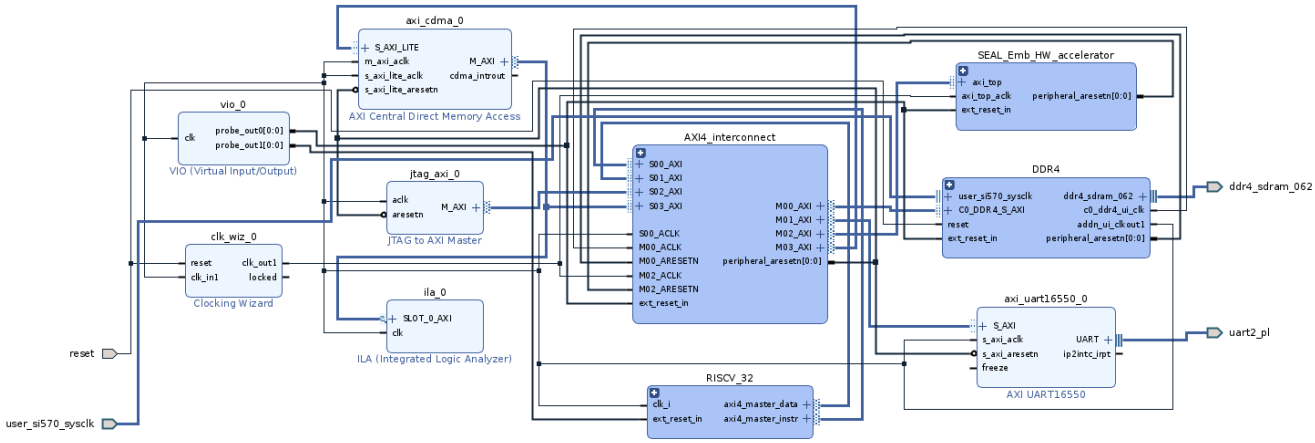

Figure 2-6: Block design implemented in Xilinx Vivado 2020.2 of the proposed system.

Table 2-2 reports a comparison among the software performance (third column) and the software plus hardware acceleration of the symmetric encryption function. In particular, the fourth column includes the results using the CPU as Master in the AXI4 communication instead of the fifth column that considers the Xilinx Central DMA as Master.
Table 2-3 shows the resources consumption on the target FPGA. The software results on the RISCY processor are slightly different respect to the benchmark previously presented since the interconnection network is different.

Polynomial Degree	Msg Size	SW (ms)	HW (ms)	HW DMA (ms)
1024	2048 B	168.35	7.84	0.142
2048	4096 B	364.53	15.68	0.297
4096	8192 B	2352.74	93.72	1.866
8192	16384 B	10032.13	374.83	7.79
16384	32768 B	45856.85	1624.12	35.19

Table 2-2 Performance results of the hardware accelerator for the symmetric encryption function of the SE library. The first column indicates the polynomial degree, the second the message size, the third indicates the execution time for a full encryption using the RISCY processor, the fourth shows the result of the software plus hardware acceleration and the fifth software plus hardware acceleration with data transfers executed using the Xilinx Central DMA.

Module	Max. Frequency	CLB LUTs	CLB REGs	BLOCKs RAM	DSPs
Hardware accelerator	150 MHz	2690 (1.167%)	1408 (0.305%)	87 (27.88%)	42 (2.43%)
RISCY	100 MHz	4487 (1.947%)	2181 (0.473%)	0	5 (0.289%)
AXI4 Interconnect	100 MHz	24058 (10%)	17882 (3.88%)	80 (25%)	0
AXI Central DMA	100 MHz	1221 (0.529%)	2168 (0.47%)	0	0

Table 2-3: Resources consumption of the proposed system on the (SoC) XCZU7EV-2FFVC1156.

3. Accelerator IP for XOF SHAKE-128/256

3.1. eXtendable Output Functions (XOF) SHAKE-128/256 and benchmarking

An eXtendable Output Function (XOF) is a variable-length HASH function in which the length of the output can be chosen to meet the requirements of individual applications.
The XOFs can be specialized to hash functions or used in a variety of other applications. The reference standard for the XOF is the NIST FIPS 202 [8], where two XOFs are specified: SHAKE-128 and SHAKE-256. Several NIST Post-Quantum finalists for both Key Encapsulation Mechanism (KEM) and Digital Signature (DS) adopt the XOF functions SHAKE 128/256: CRYSTALS-Kyber (KEM) and Dilithium (DS), Classic McEliece (KEM), NTRU (KEM), Saber (KEM) and Falcon (DS).
In particular, in DS algorithms the hardware acceleration of XOFs becomes crucial since they are employed to HASH messages of any size. Some IoT applications, for instance Over-The-Air update, requires verifying the DS of large messages (e.g. up to Gigabytes) with low latency.
Next section will show the benchmark results of the DS algorithms CRYSTALS-Dilithium and Falcon running on both RISC-V CVA6 and ARM-A53 CPUs, and the implementation results of a hardware accelerator for SHAKE128/256 functions.

Performance evaluation in Post-Quantum Digital Signature Algorithms

The source code of the Crystals-Dilithium and Falcon algorithms can be downloaded at the NIST official page for the PQC competition:
https://pq-crystals.org/,
https://falcon-sign.info/
In this case, we selected the CPUs RISC-V CVA6 and ARM-A53 because they can be reasonably used for IoT applications. Two different environments have been implemented on the FPGA Board UltraScale+ MPSoC ZCU106 equipped with the System-on-Chip (SoC) XCZU7EV-2FFVC1156:

- A RISC-V CVA6-based system, the one reported on the right side of Figure 2-1. In this case, the entire system is implemented on the target FPGA at 100 MHz of frequency.
- An ARM-A53-based hard-core system running at 1.2 GHz of frequency. The processor is connected to 2 GB of DDR4 memory.

Table 2-3 reports the results for the DS verification function of CRYSTALS-Dilithium and Falcon algorithms with different message lengths (i.e. from 10KB to 100 MB) on the RISC-V CVA6 CPU, while Table 2-4 reports similar results on the ARM-A53 CPU (in this case the message length varies from 10KB to 1 GB).

Message length[byte]	VERIFICATION FUNCTION – RISC-V CVA6 processor			
	Dilithium-2	Dilithium-5	Falcon - 512	Falcon - 1024
10K	30,27 ms	69,21 ms	14,11 ms	16,91 ms
100K	104,85 ms	143,60 ms	83,99 ms	86,88 ms
1M	865,77 ms	904,37 ms	799,24 ms	802,12 ms
10M	8455,38 ms	8492,71 ms	7.933,16 ms	7.936,13 ms
100M	84351,33 ms	84375,76 ms	79.273,83 ms	79.275,83 ms

Table 2-4: Computation time for the DS verification function of CRYSTALS-Dilithium and Falcon algorithms on the RISC-V CVA6 CPU.

Message length [byte]	VERIFICATION FUNCTION – ARM-A53 processor			
	Dilithium-2	Dilithium-5	Falcon - 512	Falcon - 1024
10K	4,65 ms	10,37 ms	4,6 ms	6,26 ms
100K	13,80 ms	19,52 ms	33,08 ms	34,71 ms
1M	107,77 ms	113,49 ms	325,00 ms	326,63 ms
10M	1.045,22 ms	1.050,89 ms	3.236,75 ms	3.238,38 ms
100M	10.419,43 ms	10.424,82 ms	32.354,11 ms	32.355,73 ms
1G	104.161,53 ms	104.167,26 ms	323.527,44 ms	323.529,06 ms

Table 2-5: Computation time for the DS verification function of CRYSTALS-Dilithium and Falcon algorithms on the ARM-A53 CPU.

3.2. Hardware accelerator IP design, interfacing to host processor and implementation results on FPGAs

The eXtendable Output Functions (XOF) SHAKE128/256 are described in the SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions [9]. Unlike SHA-3 functions, the output length of the SHAKE functions can be chosen arbitrarily to meet the requirements of individual applications. The SHAKE functions follow the structure described in [10], named the sponge construction, which is composed of two processes, the absorbing and the squeezing ones.
The first process compresses the input value using the KECCAK-*p* permutations, which are defined by two parameters: the length of the strings that are permuted, called the width or the state (and denoted with b) of the permutation, and the number of iterations called rounds (and denoted with n_r). In particular, the KECCAK-*p*[1600,24] that underlies the SHA-3 family functions, acts on a state of 1600-bit, and requires 24 rounds. A round of the KECCAK-*p* consists of a sequence of five transformations, which are called step mapping, consisting of five processes: θ, ρ, π, χ, and ι. A detailed explanation of these processes can be found in [10]. The state is separated in the first *r* bits named the rate and the remaining *c* = *b* - *r* bits named capacity.
As depicted in Figure 2-7, in the absorbing phase the first message block (of *r* bits) is XORed with the initial state (all zeroes), and then the KECCAK-*p*[1600,24] is applied. For the next message blocks, they are XORed with *r* bits of the state and the KECCAK-*p*[1600,24] is applied again. This operation continues until the whole message is consumed. After that, the squeezing phase outputs an arbitrarily number of *r* bits: after a *r* bits block is squeezed out, the KECCAK-*p*[1600,24] is applied again to the state. If the desired output is shorter than the provided *r* bits blocks, the last block can be truncated.

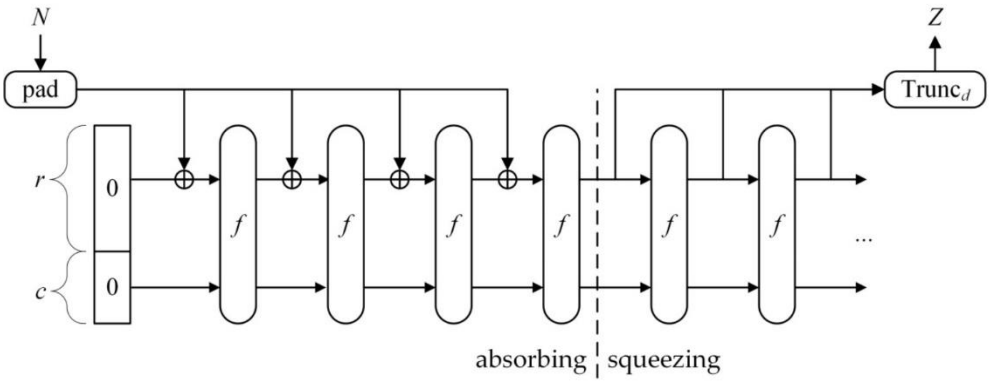

Figure 2-7: Sponge construction for the SHA3 family functions.

The architecture of the hardware accelerator for SHAKE128/256 is represented in Figure 2-8. It includes an AXI4 Memory Mapped Slave interface with a 64-bit data bus, a 1344-bit shift register for input/output from/to the AXI Master interface, a dedicated 1600-bit state register and the combinational logic for the KECCAK-*p* permutations and padding. The configuration register allows to set the number of input blocks to be processed and the last block size for padding. Table 2-6 shows the synthesis results for the proposed hardware accelerators on the FPGA Board Xilinx ZCU106.

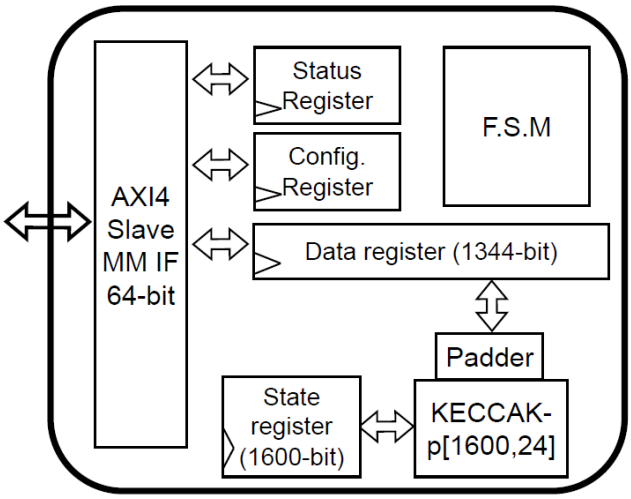

Figure 2-8: Architecture of the hardware accelerator for SHAKE128/256 functions.

Max. frequency	CLB LUTs	CLB REGs	BLOCKs RAM	DSPs
330 MHz	6172	3327	0	0

Table 2-6: Resource consumption of the hardware accelerator for SHAKE128/256 functions.

The hardware accelerator has been integrated in two SoC including based on a RISC-V CVA6 processor and an ARM Cortex-A53, as presented in Figure 2-9.

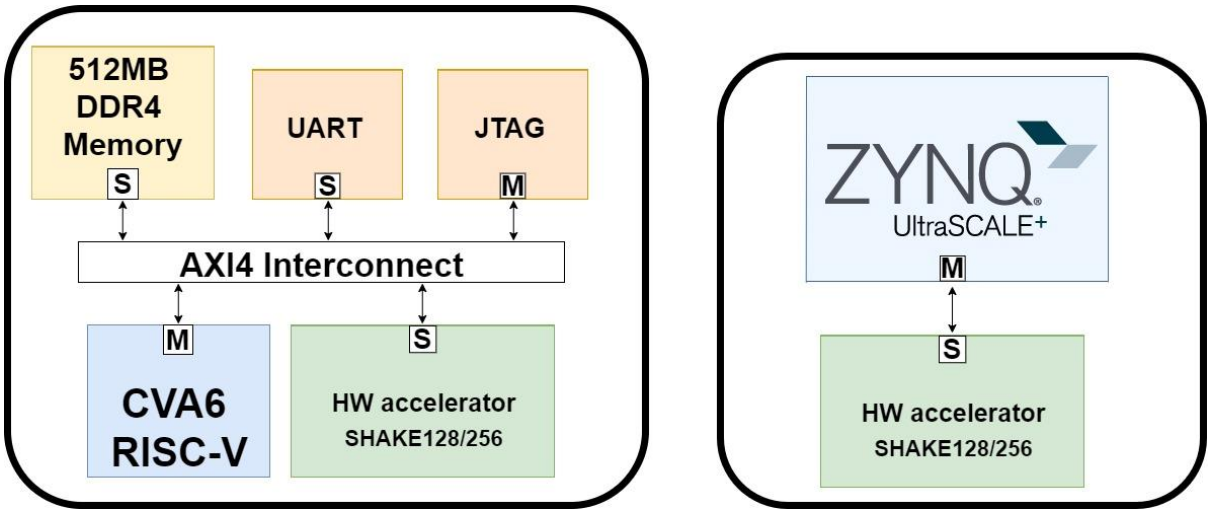

Figure 2-9: On the left side the RISC-V based system and on the right side the ARM-A53 based system. Both systems include the hardware accelerator for SHAKE128/256.

The RISC-V based system is connected to 512MB of DDR4 memory and to standard peripherals (i.e. UART and JTAG) provided as Vivado hardware IPs. The clock frequency for CPU and peripherals/interconnect is 100MHz. The ARM-A53 based system includes 2GB of DDR4 memory, and the UART peripheral to communicate with the host PC. The clock frequency of the ARM-A53 is 1.2 GHz. The accelerator has been synthetized at its maximum frequency reported in Table 2-6. The performance results for the DS verification function of CRYSTALS-Dilithium and Falcon algorithms running on the RISC-V based system (i.e. RISC-V CVA6 plus hardware accelerator, left side of Figure 2-9) are reported in Table 2-7, instead the performance results on the ARM-A53 system (i.e. ARM-A53 plus hardware accelerator, right side of Figure 2-9) are reported in Table 2-8.

Message length[byte]	VERIFICATION FUNCTION – RISC-V CVA6 processor			
	Dilithium-2	Dilithium-5	Falcon - 512	Falcon - 1024
10K	21.42 ms	60.41 ms	6.19 ms	12.12 ms
100K	30.76 ms	69.53 ms	15.06 ms	21.04 ms
1M	125.03 ms	163.78 ms	112.19 ms	115.26 ms
10M	1104.29 ms	1142.70 ms	1091.13 ms	1094.49 ms
100M	11277.23 ms	11314.93 ms	11267.40 ms	11267.40 ms

Table 2-7: Computation time for the DS verification function of CRYSTALS-Dilithium and Falcon algorithms on the RISC-V CVA6 CPU plus the hardware accelerator for SHAKE128/256 functions.

Message length [byte]	VERIFICATION FUNCTION – ARM-A53 processor			
	Dilithium-2	Dilithium-5	Falcon - 512	Falcon - 1024
10K	3.47 ms	9.05 ms	1.21 ms	2.53 ms
100K	4.10 ms	9.68 ms	1.84 ms	3.16 ms
1M	10.55 ms	16.13 ms	8.28 ms	9.62 ms
10M	75.11 ms	80.66 ms	73.25 ms	74.18 ms
100M	720.46 ms	725.71 ms	719.52 ms	719.52 ms
1G	7345.00 ms	7348.19 ms	7342.34 ms	7342.34 ms

Table 2-8: Computation time for the DS verification function of CRYSTALS-Dilithium and Falcon algorithms on the ARM-A53 CPU plus the hardware accelerator for SHAKE128/256 functions.

4. Conclusions and IP repository

This document has reported the activities done by TEXTAROSSA partner CINI (UNIPISA), with reference to the consolidated specifications of accelerator IPs in D2.1, and preliminary (since this is deliverable Part1) HDL design, verification, and synthesis results in FPGA technology for cryptographic IPs.

In Section 2 it has been addressed the design, interfacing, FPGA implementation and verification of a hardware accelerator for Homomorphic Encryption.

The repository for this IP (HE_SEAL_Embedded_HW_IP) is:

https://drive.google.com/drive/folders/1UxmG3t8YLPV6oRAi0B9zwMZK13cAjAET?usp=sharing.

In Section 3 it has been addressed the design, interfacing, FPGA implementation and verification of a hardware accelerator for eXtendable Output Functions (XOF) SHAKE-128/256, a new hashing functions used with the algorithms like CRYSTALS -Dilithium [14] recently standardized by NIST [15] for digital signature generation/verification compliant with a Post-Quantum Cryptography (PQC) scenario.

The repository for this IP (SHAKE_HW_Accelerator_IP) is:

https://drive.google.com/drive/folders/1V07qDMeCX8IdoJrVOXxuQuj82D_gl_a-?usp=sharing.

The two IPs have been provided with a standard memory-mapped AXI4 interface, so that they can be easily interfaced to any other digital IPs. To this aim results of the integration of the 2 proposed secure IPs with open source RISC-V 32bits [6] and RISC-V 64bits [7] processors have been also presented.

Implementation results in Xilinx FPGA technology (FPGA Board Zynq UltraScale+ MPSoC ZCU106 equipped with the System-on-Chip (SoC) XCZU7EV-2FFVC1156.) are also shown in both Sections 2 and 3.

Impact and dissemination

The proposed IPs are interesting, also in view of synergies between TEXTAROSSA and the other initiatives like EPI SGA2 and the European Pilot, since all the proposed accelerators can be integrated with RISC-V computing cores like the RISC-V in the EPAC (European Processor Accelerator).

Moreover, the two IPs can be also used as starting point to enrich in EPI SGA2 the hardware security module, called Cryptotile, initially developed in EPI SGA1 and missing PQC and homomorphic encryption capability. This synergy is useful also since in TEXTAROSSA grant agreement is not foreseen in WP6 a specific system-level application using PQC, which instead is part of the UNIPI work in EPI SGA2.

In the final version of this deliverable (Part2 that is D2.7) UNIPI will present results related to an application for secure firmware over the air update, involving a communication between Cloud server and edge devices, needing signature generation and verification, and implemented using as starting point the IP discussed in Section 3.

The results about the homomorphic accelerator have been submitted to the consideration of the scientific community with the paper submitted to the journal IEEE ACCESS entitled:

S. Di Matteo, M. Lo Gerfo, S. Saponara, “Design and Implementation on FPGA of a Hardware Accelerator for Microsoft SEAL-Embedded”, IEEE ACCESS, under review.

The results about the SHAKE accelerator will be submitted to the consideration of the scientific community with the paper under finalization for submission: S. Di Matteo, M. La Manna, P. Perazzo, G. Dini, S. Saponara, “On Hardware Acceleration of Quantum-secure FOTA Systems in Automotive”.

5. References

[1] Chen, H., Laine, K., & Player, R. (2017). Simple Encrypted Arithmetic Library - SEAL v2.1. Financial Cryptography Workshops.
[2] PALISADE. https://gitlab.com/palisade. New Jersey Institute of Technology (NJIT).
[3] Halevi, S., & Shoup, V. (2020). HElib design principles. Tech. Rep.
[4] Natarajan, D., & Dai, W. (2021). SEAL-Embedded: A Homomorphic Encryption Library for the Internet of Things. IACR Transactions on Cryptographic Hardware and Embedded Systems, 2021(3), 756–779. https://doi.org/10.46586/tches.v2021.i3.756-779.
[5] https://github.com/microsoft/SEAL-Embedded.
[6] https://github.com/pulp-platform/pulpino.
[7] https://github.com/openhwgroup/cva6.
[8] Dworkin, M. (2015), SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions, Federal Inf. Process. Stds. (NIST FIPS), National Institute of Standards and Technology, Gaithersburg, MD.
[9] M. Dworkin, “Sha-3 standard: Permutation-based hash and extendable output functions,” 2015-08-04 2015.
[10] B. Guido, D. Joan, and P. Michaël, “Cryptographic sponge functions,” 2011.
[11] https://github.com/Microsoft/SEAL
[12] https://www.intel.com/content/www/us/en/developer/tools/homomorphic-encryption/overview.html
[13] https://developer.nvidia.com/blog/federated-learning-with-homomorphic-encryption/
[14] https://pq-crystals.org/dilithium/index.shtml
[15] https://csrc.nist.gov/Projects/post-quantum-cryptography/post-quantum-cryptography-standardization