Efficient and High-Speed CGRA Accelerator for Cryptographic Applications

Vu Trung Duong Le[1], Hoai Luan Pham[1], Thi Hong Tran[2], Thi Sang Duong[1], and Yasuhiko Nakashima[1]
[1] Nara Institute of Science and Technology, 8916–5 Takayama-cho, Ikoma, Nara, 630-0192 Japan.
[2] Osaka Metropolitan University, Osaka 558-8585, Japan.
Email: le.vu_trung_duong.lp4@is.naist.jp, pham.luan@is.naist.jp, hong@osaka-cu.ac.jp,
duong.sang.do4@is.naist.jp, nakashim@is.naist.jp

***Abstract*—Efficient cryptographic hardware architectures must exhibit both high performance and flexibility to effectively meet the diverse demands of data security in multi-domain applications. However, current research faces challenges in achieving a balance between high processing speed and flexibility, leading to low hardware efficiency and limitations in applicability. Therefore, this paper proposes the ultra-efficient crypto-processor (UECP), a coarse-grained reconfigurable arrays (CGRA) accelerator designed to support various cryptographic algorithms, offering excellent performance and hardware efficiency. To significantly enhance performance and configuration efficiency, the UECP implements two optimal techniques: a 4x4 pipelined processing element array (PEA) and an effective configurable arithmetic logic unit (C-ALU). The performance evaluation of the UECP on the Xilinx ZCU102 FPGA at the system-on-chip level demonstrates a throughput ranging from 1.45 to 4.35 Gbps, power consumption of 0.357 W, and remarkable energy efficiency of 12.19 Gbps/W. The UECP on FPGA outperforms modern embedded CPUs such as ARM Cortex A53 and ARM Cortex A57 by 6.23-24.1× in various algorithm computations. Moreover, when compared to current low-flexibility FPGA-based designs, the flexible UECP demonstrates a throughput superiority of 1.58-10.71×. Finally, the ASIC synthesis comparison results show that the UECP outperforms related works with 7.16-22× higher throughput, 7.92-22.5× better area efficiency, and 1.76-15.36× superior energy delay products (EDP).**

***Index Terms*—FPGA, ASIC, CGRA, Area efficiency, Energy efficiency, Flexibile crypto-processor, Reconfigurable architecture, Cryptography**

I. Introduction

Cryptography plays a crucial role in guaranteeing data privacy, integrity, and authentication, making it an increasingly popular and essential tool for safeguarding data across diverse applications for communications and the internet. In more detail, hash function algorithms such as SHA256 and BLAKE256 are applied in blockchain mining, data integrity verification, password storage, and digital signatures [1]. Moreover, stream cipher algorithms such as Chacha20 and Salsa20 are popularly used for SSH and TLS encryption [2]. In addition, symmetric encryption algorithm such as AES is widely operated in securing communication, VPN, and data encryption systems [3]. To fulfill the security demands, there arises a need for a flexible hardware architecture capable of executing various cryptographic algorithms with superior performance and hardware efficiency across diverse application domains.

Numerous studies propose hardware architectures for cryptography to improve performance and hardware efficiency for multiple cryptographic applications. In [4]–[8], the authors propose application-specific integrated circuit (ASIC)-based cryptographic accelerators to achieve extremely high speed and power efficiency. Nonetheless, these architectures are limited in low flexibility due to supporting only one or two algorithms, which restricts the range of their applicability. In contrast, the authors in [9] introduce an application-specific instruction set processor (ASIP) for cryptography, aiming to offer outstanding flexibility for multiple algorithm calculations. However, these processors face challenges with extremely low performance because of the high latency for executing many instructions. These issues arise from high configuration costs caused by ineffective frequent memory access for intermediate value storage, as well as the lack of sufficient support for pipeline computing. Moreover, to enhance both flexibility and hardware efficiency, the authors in [10], [11] propose coarse-grained reconfigurable arrays (CGRA), enabling programmability for multiple cryptographic algorithm executions with high performance to adapt security tasks in various domains. However, these CGRAs face limitations in several cryptographic algorithms containing complex logical procedures due to the inefficiency of processing element architecture and crossbar configuration, leading to the increment in configuration cost, latency, and hardware resources. Typically, there still lacks a programmable hardware architecture capable of efficiently supporting multiple cryptographic algorithms with high hardware efficiency to meet high-security system requirements.

In this paper, the ultra-efficient crypto-processor (UECP), a novel CGRA-based accelerator for cryptography, is proposed to reach high flexibility and hardware efficiency. Particularly, the UECP uses a 4x4 pipeline processing element array (PEA), integrating configurable arithmetic logic units (C-ALU) with local data memories inside. This design aims to minimize memory access, facilitate pipeline processing, and ultimately improve both configuration efficiency and hardware utilization.

The structure of the paper is as follows: In Section II, the background of cryptographic applications and the essential requirements for a dedicated hardware architecture in cryptography are discussed. Section III outlines the specifics of the proposed UECP architecture. Further insights into

2379-1896/23/$31.00 ©2023 IEEE
DOI 10.1109/CANDAR60563.2023.00033

the flexibility and performance evaluations of the UECP are provided in Section IV. The paper concludes in Section V, summarizing the findings and exploring potential avenues for future research.

II. Background

A. Multi-cryptography applications

Cryptography plays a crucial role in safeguarding data security and maintaining its accuracy. Notably, various widely recognized cryptographic algorithms contribute to secure communication and data protection in different fields. Some prominent examples include SHA256, BLAKE256, Chacha20, Salsa20, and AES. SHA256 and BLAKE256 find extensive use in systems like blockchain mining, digital signatures, certification generation, and Merkle tree construction [1]. Chacha20 and Salsa20 serve as efficient stream cipher solutions, ensuring secure communication and file protection in protocols like SSH and TLS [2]. Finally, AES is a commonly used symmetric encryption method applied in different areas like safeguarding data, securing files, protecting databases, enabling VPNs, and ensuring password confidentiality. It's utilized across multiple fields, including government, healthcare, and business.

The integration of multiple cryptographic algorithms can significantly boost system security through the adoption of a multi-layered defense strategy. For instance, in the context of secure web browsing, SHA256 is employed to verify SSL/TLS certificates, and AES is used to encrypt both browsing and cookie data. In specific cloud storage platforms like Google Cloud Storage, Amazon S3, and Dropbox, AES takes on the role of data encryption, while SHA-256 is utilized to sign user requests [12]. In hypothetical scenarios involving secure communication systems, SHA256 is utilized for certificate validation and trust establishment, while BLAKE256 performs rapid and secure hashing of internal data. Additionally, to protect general user information, Chacha20 and Salsa20 are combined as stream cipher encryption. Lastly, AES encrypts critical data such as images, videos, credit details, and payment information [13].

The use of a combination of cryptographic techniques is becoming more prevalent as a means to achieve robust data security. With the increasing demands for strong security measures, developing high-performance hardware capable of supporting diverse cryptosystems is critical. Such specialized hardware plays a vital role in contemporary computing systems by enhancing security protections and aiding in essential information-safeguarding duties.

B. Preliminary analysis on cryptography

As mentioned above, the need for efficient and versatile programmable architectures for cryptography has become increasingly apparent. There are three main properties of cryptography that lead to the requirements for a novel programmable hardware architecture to reach high performance and flexibility for multiple security applications.

- **Support for multiple operations per instruction**: Cryptographic algorithms involve various basic operations, including logical (and, or, xor), shift-rotation, and addition, for their computations. In addition to supporting these operations, a programmable architecture must offer multiple operations per instruction to reduce latency and configuration costs, thereby ensuring hardware efficiency.
- **Support for pipelined processing**: Cryptography often contains complex calculations in multiple loops leading to high latency and long critical paths. To address this challenge, the hardware should support pipeline processing to enable multiple data parallel processing. The pipeline helps shorten the critical path and reduce latency, leading to substantial improvements in performance and hardware efficiency.
- **Support for complex operations**: Certain cryptographic algorithms, like AES and SHA256, involve complex procedures such as SubBytes, MixColumns, ShiftRows, σ_0^{256}, σ_1^{256}, Σ_0^{256}, and Σ_1^{256}, which cannot be easily configured using basic operation instructions alone. To address this challenge, an efficient hardware architecture must offer customized instructions to program and execute these complex procedures more efficiently. These instructions reduce latency, minimize configuration costs, and substantially enhance overall performance.

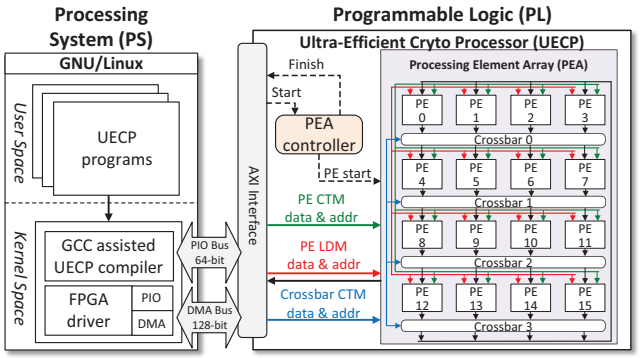

Fig. 1. Overview of the UECP architecture at the FPGA SoC level

By considering these critical aspects, we propose a CGRA-based accelerator for cryptography, offering versatility and high performance across various security applications. This accelerator, named UECP, is thoroughly explained in the next section, highlighting its high flexibility and hardware efficiency for serving different security tasks.

III. Proposed Architecture

A. Overview architecture

As illustrated in Fig. 1, the architecture of the UECP was implemented on field programmable gate arrays (FPGA) at the system-on-chip (SoC) level to achieve high performance and hardware efficiency. The architecture consists of two primary components: a processing system (PS) and programmable logic (PL), communicating via AXI buses.

The PS, under the control of the embedded CPU running on GNU/Linux, stores UECP programs in the user space. The UECP compiler, assisted by the GNU Compiler Collection

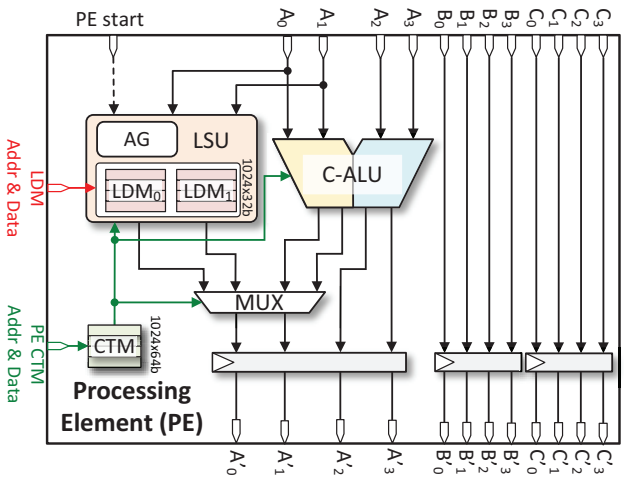

Fig. 2. Processing element architecture

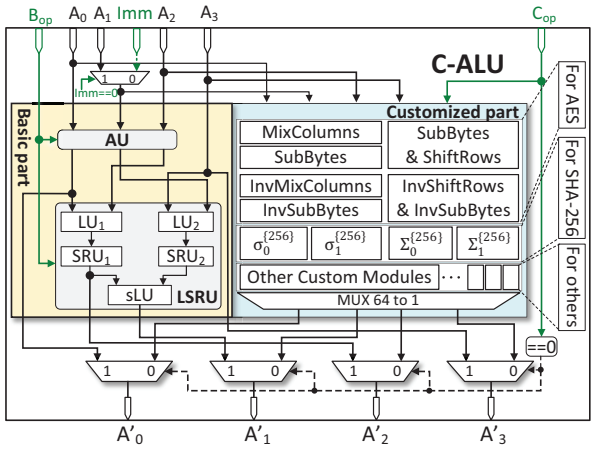

Fig. 3. Configurable ALU architecture

(GCC), compiles these programs in kernel space. The PS and PL exchange data through two mechanisms. Infrequently accessed configuration parameters are stored in context memories (CTMs) and transferred via a simple parallel input/output (PIO) method. For high-throughput computational data stored in local data memories (LDMs), direct memory access (DMA) transfers are utilized. DMA provides a more efficient data transfer modality compared to PIO for the frequent movement of processing data. The UECP within the PL is comprised of three components: an AXI interface decoding transferred data, a processing element array (PEA) controller managing PEA operations, a 4x4 PEA performing primary processing, and crossbars routing PEs data. The PEA controller generates control data to effectively manage UECP operations, while the PEA and crossbar require configured context and computational data to execute their functions.

The architecture of a processing element (PE) is described in Fig. 2, containing three primary components: a load-store unit (LSU), data buffers, and the configurable arithmetic logic unit (C-ALU). The LSU, containing two 1024x32-bit LDMs, stores executed data transferred between the UECP and PS. Specifically, the LSU enables the PE to store computational data received from or written to the PS utilizing its LDMs instead of relying on access to global memories as demonstrated in [11]. Additionally, a PE consists of three groups of 4x32-bit buffers named A, B, and C to store immediate computational data after each pipeline processing stage. Finally, the C-ALU is the main processing component of a PE. It reads configured context from CTM and processes A data group, operating according to an instruction set designed for efficient cryptography. Further details about the C-ALU hardware architecture and instruction set are presented in the following section.

B. *Efficient Configurable ALU for cryptography*

To meet the rising demand for efficient and versatile cryptographic hardware architectures, the C-ALU is proposed as a key component within each PE of the UECP. Particularly, C-ALU can support various cryptographic operators with low configuration costs, enable parallel processing for loop calculations, and perform complex data manipulations required by popular algorithms. This C-ALU architecture aims to enhance security and efficiency for cryptographic operations, meeting the diverse requirements for robust data protection. In this section, we provide a detailed discussion of the C-ALU design and functionalities.

Fig. 3 depicts the architecture of the configurable arithmetic logic unit (C-ALU), designed to perform two types of operations: basic (B_{op}) and customized (C_{op}). The basic operations include common cryptographic functions such as logic, shift, rotate, and arithmetic operations, providing a foundation for various cryptographic algorithms. The basic portion of the C-ALU implements these operations. The customized portion of the C-ALU is intended to execute the complex data manipulations required by specific algorithms. It incorporates dedicated modules to process intricate procedures, including MixColumns, SubBytes, and ShiftRows for the AES as well as σ_0^{256}, σ_1^{256}, Σ_0^{256}, and Σ_1^{256} for the SHA256. Additionally, the C-ALU can be customized with modules to support other cryptographic algorithms. Integrating these specialized modules enables the C-ALU to optimize performance and substantially reduce configuration costs for fixed programmable operations.

In the C-ALU, basic operators are covered by some fundamental operation units: the arithmetic unit (AU), logic unit (LU), sub-logic unit (sLU), and shift-rotate unit (SRU). The AU handles basic arithmetic operations, such as addition and subtraction, while the LU and sLU perform logical cryptographic operations. Additionally, the SRU facilitates shift and rotation operations, which are crucial for various cryptographic techniques that rely on data permutation and reordering. These units are strategically proposed based on frequently used operators in various cryptographic algorithms, ensuring the most efficient configuration. By the proposed basic part, C-ALU can simultaneously execute one arithmetic operation, three logic operations, and two shift-rotation operations in a single instruction, which allows UECP to achieve high

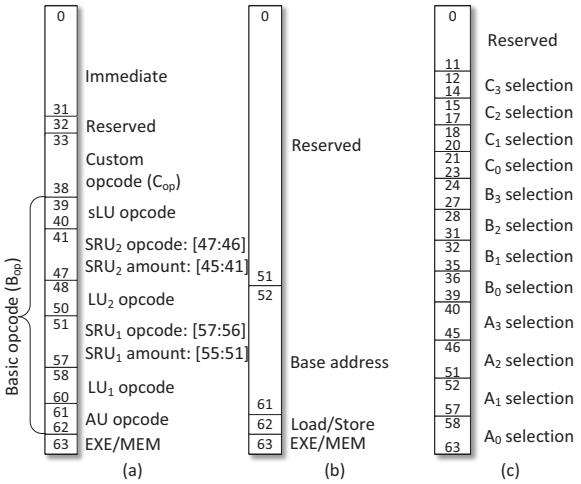

Fig. 4. Instruction format: (a) EXE, (b) MEM, and (c) MAP format

processing speed and minimal configuration costs.

The UECP utilizes 64-bit instructions for configuration, employing three distinct formats as shown in Fig. 4: execution (EXE), memory (MEM), and mapping (MAP). The EXE instruction configures C-ALU computations, with the basic opcode (B_{op}) and customized opcode (C_{op}) occupying 24 and 6 bits, respectively. The MEM instruction manages load/store operations for the LSU, utilizing a 10-bit base address to access the 1024x32-bit LDMs. The MAP instruction configures the crossbar, selecting data sources (A_{0-3}, B_{0-3}, and C_{0-3}) for the PE, as depicted in Fig. 5. Notably, data in group A can be chosen from any PEs in the previous row, while groups B and C are limited to data from the same or adjacent PEs in the prior row. This minimizes multiplexer resources in the crossbar, improving efficiency.

In summary, the proposed C-ALU architecture provides high efficiency and flexibility for cryptographic applications. By supporting basic and customized operations, the C-ALU can handle diverse, complex algorithms such as AES and SHA256. Additionally, executing multiple concurrent operations per instruction assists in minimizing configuration overhead and enhancing performance. In the following section, we demonstrate the UECP hierarchical pipeline mechanism for substantial data processing in real-time systems.

C. Hierarchical pipeline for tremendous data processing

In this part, the hierarchical pipeline mechanism is presented for the UECP to perform real-time computation on large volumes of data, accommodating various application requirements. Fig. 6 depicts in more detail the two-level pipeline of the proposed mechanism. The first-level pipeline enables concurrent system data transfer and hardware execution, increasing data transfer bandwidth. Moreover, the second pipeline enhances UECP performance when processing massive input datasets.

The scheduling of the system-level pipeline is displayed in Fig. 6(a). There are two AXI bus types: a 64-bit parallel

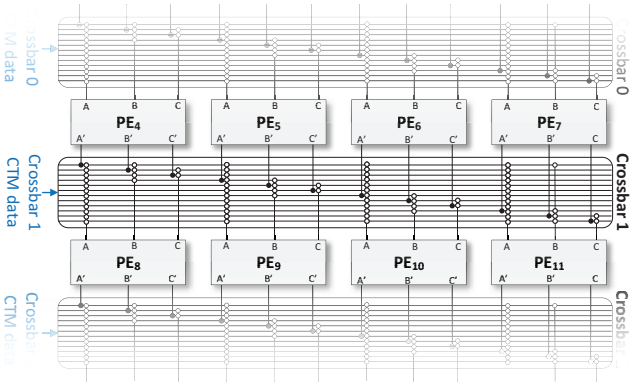

Fig. 5. The mapping of the UECP crossbars

input/output (PIO) bus and a 256-bit direct memory access (DMA) bus. The PIO bus transfers UECP configuration context and control data, while the DMA bus transfers processing data. The system can handle a batch of N input data in one go, where N is half the storage capacity of LDMs inside each PE. The pipeline utilizes a memory ping-pong technique, processing half the stored data while moving the other half. In the system-level pipeline mechanism, the context data for PEA and crossbars are first transferred once via the PIO bus to program the assigned functions for the UECP. After that, when the PEA executes for a data batch stored in half of the LDMs, the CPU can read the output data from and transfer a new data batch to the remaining half of the LDMs for the next PEA execution. The pair of start and finish signals are control signals that notify when the PS reads the output data batch and when the UECP starts a new execution. Overall, this pipeline process enhances system performance and data transfer efficiency by enabling continuous UECP processing without interrupting computations to move large datasets.

Fig. 6(b) illustrates how the UECP-level pipeline is scheduled. This pipeline divides the UECP into eight pipeline stages, including four 4xPE row stages and four crossbar stages, placed alternatively. In this setup, every pair of the crossbar and PE row works together to handle eight input cases with the same programmed operators over eight cycles. The last pipeline stage, comprising the last PE row, produces eight output data, which are fed back to the first crossbar for the next cycle. To change the programmed context, the accessing addresses of context memories are automatically increased to the next context for all PEs and crossbars. In a working session, the data feedback and context-switching processes repeat in a loop until the final results are computed after M iterations. In particular, the first loop iteration loads the initial eight data inputs, while the last loop iteration stores the final eight data outputs. After 8xM cycles, a working session for processing eight input cases is finished. The bias address of LDMs is then increased by eight to point to the next eight input data. Obviously, the UECP needs to perform N/8 working sessions to complete a batch of N inputs. During each session, all PEs and crossbars are constantly active without any periods

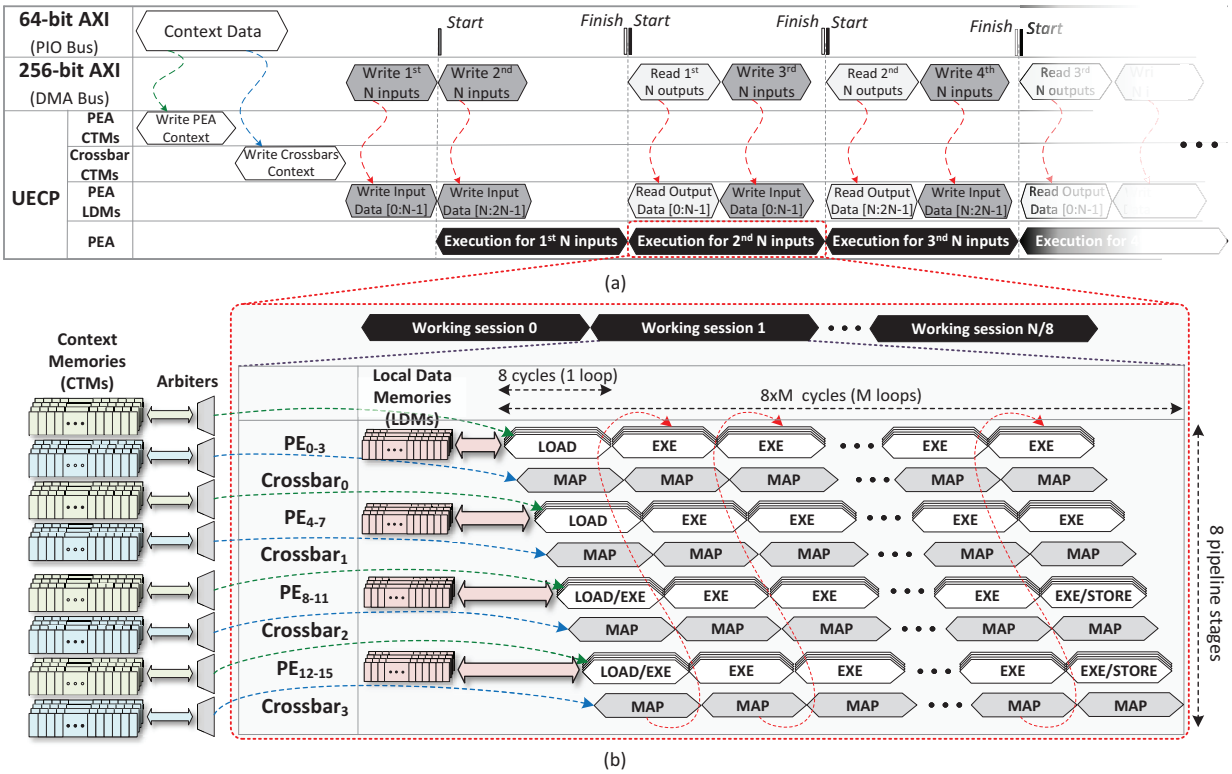

Fig. 6. Two-level pipeline schedule includes (a) system level and (b) UECP level

of inactivity. This is because of the ongoing feedback and continuous transfer of pipelined data between loops. Moreover, there is no waiting time between two working sessions because of the automated address increment mechanism for context and local data memories. Therefore, by minimizing periods of inactivity, hardware efficiency can be optimized, potentially achieving a 100% utilization rate.

In summary, the hierarchical pipeline architecture substantially enhances UECP performance for real-time applications requiring security and efficiency. The following section covers verification and evaluation results, highlighting the architecture's exceptional improvements in flexibility and hardware efficiency.

IV. Evaluation Results

A. Implementation and Comparison with CPUs

To verify the accuracy, our UECP design was implemented on the Xilinx ZCU102 FPGA at the SoC level for verification and evaluation. In the PS, the ARM Cortex-A53 CPU on Petalinux 2021.2 operating system prepares data and configures the UECP contexts. Particularly, the UECP is configured to execute 100,000 test cases of five different algorithms, namely BLAKE256, BLAKE2s, SHA256, Chacha20, and AES128. The system utilizes **15,916 FFs, 92,118 LUTs,** and **96 BRAMs** and consumes a power of **0.357 W** at an operation frequency of **187 MHz**. After verification, the UECP finished computations for all test cases with an accurate rate of 100%.

Additionally, Fig. 7 shows the execution time comparison of the following test cases set between the proposed UECP

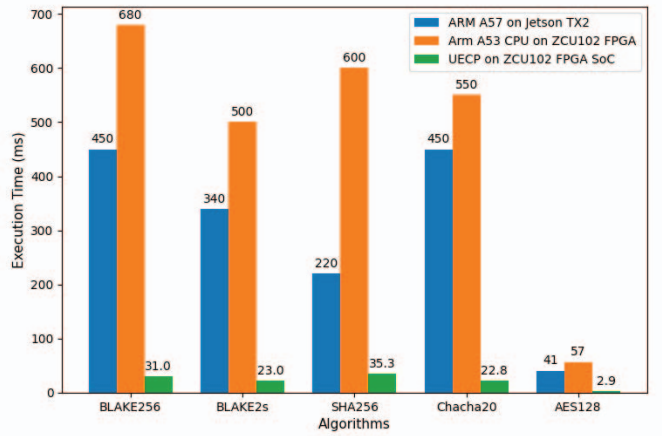

Fig. 7. Performance comparison of the proposed UECP, ARM Cortex-A53 CPU, and ARM Cortex-A57 CPU on various cryptographic algorithms

and several embedded CPUs such as ARM Cortex A57 on Jetson TX2 developer kit and ARM Cortex A53 on ZCU102 FPGA to prove its outstanding performance. Specifically, the UECP's execution time reaches the fastest value of 2.9 ms at the AES128 mode and the slowest value of 35 ms for the SHA265 mode. These results reveal that the UECP is faster than the ARM Cortex A57 CPU from **6.23**× (220 vs. 35.3 ms) to **19.73**× (450 vs. 22.8 ms). Similarly, the UECP outperforms the ARM Cortex A53 CPU from **17**× (600 vs. 35.3 ms) to **24.1**× (550 vs. 22.8 ms).

The implementation results have demonstrated that the

TABLE I
COMPARISON OF THE PROPOSED UECP WITH EXISTING FPGA-BASED WORKS

Design level		Standalone	Standalone	Standalone	Standalone	Standalone	Standalone	**SoC**			
Reference		[3]	[14]	[15]	[16]	[17]	[18]	**Proposed UECP**			
Flexibility		No	No	No	Low	No	No	**High**			
FPGA Device		Virtex5	Arria10GX	CycloneV	Virtex6	Virtex6	Zynq7000	**ZCU102**			
Technology (nm)		65	20	28	28	28	28	**16**			
Algorithm		AES128	AES128	AES128	Chacha20	BLAKE256	SHA256	**AES128**	**SHA256**	**BLAKE256**	**Chacha20**
Resources	LUT	3352	-	6618	-	-	-	**92,118**			
	FF	-	-	-	-	-	25,190	**15,916**			
	Slice	428	4729	-	77	267	-	**31,262**			
	DSP	20	-	-	-	-	-	**0**			
Frequency (MHz)		268.2	667	150	345	154	181	**187**			
Power (W)		-	-	-	-	-	0.38	**0.357**			
Throughput (Gbps)		1.597	0.854	0.7	0.21	0.25	0.917	**4.35**	**1.45**	**1.65**	**2.25**
Eng_eff (Gbps/W)		-	-	-	-	-	2.41	**12.19**	**4.06**	**4.6**	**6.28**

UECP can substitute embedded CPUs in executing cryptographic algorithms, providing substantially higher performance.

B. Comparison with existing FPGA-based work

In order to elucidate the exceptional flexibility, performance, and hardware efficiency, exhaustive benchmarking was undertaken, contrasting the UECP on Xilinx ZCU102 FPGA SoC against existing inflexible FPGA-based standalone core and SoC architectures. Comparative evaluations of throughput and energy efficiency were performed for four prevalent cryptographic algorithms: AES128, SHA256, BLAKE256, and Chacha20. Throughput and energy efficiency were quantified as described in 1 and 2, respectively. The results, tabulated in Table I, demonstrate the significant advantages granted by the UECP.

$$Throughput = \frac{Block\_Size \times Frequency}{\#Execution\_Cycle} \tag{1}$$

$$Energy\ Efficiency = \frac{Throughput}{Power} \tag{2}$$

For AES128 throughput, the UECP outperformed standalone designs in [3], [14], [15] by a remarkable **2.73**× to **6.21**×, achieving speeds of 4.35 Gbps compared to 1.597 Gbps and 0.7 Gbps, respectively. In Chacha20 and BLAKE computation, the UECP's throughput surpassed [16] and [17] by an impressive **10.71**× (2.25 Gbps vs. 0.21 Gbps) and **6.6**× (1.65 Gbps vs. 0.25 Gbps), respectively. Moreover, in SHA256 computation, the UECP on SoC exhibited superior performance compared to SoC design in [18], with throughput and energy efficiency ratios of **1.58**× (1.45 Gbps vs. 0.917 Gbps) and **1.68**× (4.06 Gbps/W vs. 2.41 Gbps/W), respectively.

The evaluation results clearly demonstrate the exceptional capabilities of the UECP compared to FPGA-based research, making it an exceptional choice for cryptographic tasks across various applications.

C. Comparison with existing ASIC-based work

To further prove the outstanding performance, hardware efficiency, and flexibility, the UECP architecture is synthesized on the ASIC FreePDK 45nm library. In Table II, the comparison results with other ASIC-based related works in the area, power consumption, throughput, area efficiency, energy efficiency, and energy-delay products (EDP) of 1,000,000 execution cases are displayed to demonstrate the excellence of the proposed architecture. Accordingly, the area efficiency and EDP are calculated as eq 3 and 4, respectively, where the lower EDP value means better efficiency.

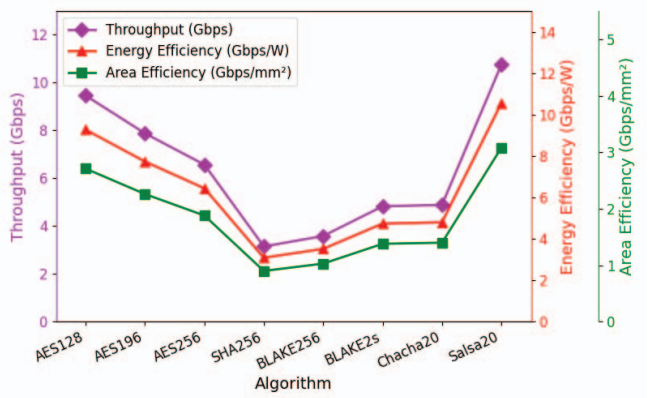

Fig. 8. Throughput, area efficiency, and energy efficiency of UECP on different cryptographic algorithms

$$Area\ Efficiency = \frac{Throughput}{Area} \tag{3}$$

$$EDP = Execution\_Time^2 \times Power \tag{4}$$

Obviously, the UECP utilizes the area of **3.49 mm**^{2} and consumes the power of **1.02 W** at the operating frequency of **406.5 MHz**. In AES128 computation mode, the proposed UECP is better than the NanoAES design in [19] **22**× (9.46 vs. 0.43 Gbps), **16.9**× (2.71 vs. 0.16 Gbps/mm^2), and **6**× (2 vs. 12) in terms of throughput, area efficiency, and EDP, respectively. For SHA256 mode, the UECP continues to show surpass the PVHArray architecture in [11] **7.16**× (3.15 vs. 0.44 Gbps) in throughput, **22.5**× (0.9 vs. 0.04 Gbps/mm^2) in area efficiency, and **1.76**× (269 vs. 473) in EDP. Finally, in BLAKE256 processing mode, the UECP has throughput higher than PUFs BLAKE design in [20] **10.88**× (3.59 vs.

TABLE II
COMPARISON OF THE PROPOSED UECP ARCHITECTURE WITH OTHER STATE-OF-THE-ART ASIC WORKS

References	Tech. (nm)	Config-urable	Area (mm^2)	Freq. (MHz)	Power (mW)	Algorithm	Throughput (Gbps)	Area Eff (Gbps/mm^2)	Energy Eff (Gbps/W)	EDP
NanoAES [19]	22	No	2.61	1,100	13	AES128	0.43	0.16	33.08	12
PVHArray [11]	55	Yes	12.25	110	35	SHA256	0.44	0.04	15.71	473
PUFsBLAKE [20]	65	No	2.62	780	91	BLAKE256	0.33	0.13	3.63	2,195
Proposed UECP	**45**	**Yes**	**3.49**	**406.5**	**1,020**	**AES128**	**9.46**	**2.71**	**9.27**	2
						SHA256	**3.15**	**0.9**	**3.09**	269
						BLAKE256	**3.59**	**1.03**	**3.52**	208
						Chacha20	**4.89**	**1.4**	**4.8**	112

(*) In the 180nm technology, one kGE (kilo-Gate Equivalent) is approximately equivalent to an area of 2.25 mm^2.

0.33 Gbps), the higher area efficiency of **7.92**× (1.03 vs. 0.13 Gbps/mmm^2), and the better EDP of **15.36**× (208 vs. 2,195).

Moreover, Fig. 8 illustrates the impressive flexibility of the UECP through a benchmark evaluation of its throughput, energy efficiency, and area efficiency across seven execution tasks: AES128, AES196, AES256, SHA256, BLAKE256, BLAKE2s, Chacha20, and Salsa20. The UECP achieves remarkable results, delivering throughput ranging from **3.15 to 10.76 Gbps**, energy efficiency between **3.09 and 10.55 Gbps/W**, and area efficiency from **1.03 to 3.08 Gbps/mm^2**. These comparison results on ASICs prove the UECP's outstanding performance and ability to adapt to diverse cryptographic applications.

V. CONCLUSION

In summary, the ultra-efficient cryptography processor (UECP) is an ideal proposed hardware architecture for high performance and flexibility adapting to cryptographic applications. Through its coarse-grained reconfigurable design and pipelined processing element array (PEA) with an efficient configurable ALU, the UECP achieves excellent throughput, hardware efficiency, and flexibility. Evaluations on an FPGA demonstrate significant gains over modern CPUs and inflexible FPGA implementations across various algorithms. Moreover, ASIC synthesis comparisons exhibit advantages in throughput, area, and energy-delay products (EDP). Future work will focus on expanding the programmable ALU to support more algorithms and increasing the PEA size to boost performance and flexibility further.

ACKNOWLEDGMENT

This work was supported by Projects Supported by NAIST Foundation FY2023 under Grant R5190015 and the Japan Science and Technology Agency (JST) under Grant JPMJPR20M6. This work was also supported through the activities of VDEC, The University of Tokyo, in collaboration with NIHON SYNOPSYS G.K.

REFERENCES

[1] R. Martino and A. Cilardo, "SHA-2 Acceleration Meeting the Needs of Emerging Applications: A Comparative Survey," *IEEE Access*, vol. 8, pp. 28 415–28 436, 2020.

[2] R. S. Salman, A. K. Farhan, and A. Shakir, "Lightweight Modifications in the Advanced Encryption Standard (AES) for IoT Applications: A Comparative Survey," in *CSASE 2022*, Duhok, Iraq, 2022, pp. 325–330.

[3] A. E. Makhloufi, N. Tagmouti, N. Chekroun, S. E. Adib, J. A. Sobrino, and N. Raissouni, "AES/FPGA Encryption Module Integration for Satellite Remote Sensing Systems: LST-SW case," in *CommNet2020*, Marrakech, Morocco, 2020, pp. 1–7.

[4] H. L. Pham, T. H. Tran, V. T. D. Le, and Y. Nakashima, "A High-Efficiency FPGA-Based Multimode SHA-2 Accelerator," *IEEE Access*, vol. 10, pp. 11 830–11 845, 2022.

[5] V. D. Le, T. Tran, H. Pham, D. Lam, and Y. Nakashima, "MRSA: A High-Efficiency Multi ROMix Scrypt Accelerator for Cryptocurrency Mining and Data Security," *IEEE Access*, vol. 9, 2021.

[6] P. Luan, T. Tran, V. D. Le, and Y. Nakashima, "A Flexible and Energy-Efficient BLAKE-256/2s Co-Processor for Blockchain-based IoT Applications," in *SBCCI 2022*, Porto Alegre, Brazil, 2022.

[7] H. L. Pham, T. H. Tran, V. T. D. Le, and Y. Nakashima, "Compact Message Permutation for a Fully Pipelined BLAKE-256/512 Accelerator," *IEEE Access*, vol. 10, pp. 68 740–68 754, 2022.

[8] T. H. Tran, H. L. Pham, T. D. Phan, and Y. Nakashima, "BCA: A 530-mw Multicore Blockchain Accelerator for Power-Constrained Devices in Securing Decentralized Networks," *TCAS-I: Regular Papers*, vol. 68, no. 10, pp. 4245–4258, 2021.

[9] Y. Zhang, J. Han, X. Weng, Z. He, and X. Zeng, "Design Approach and Implementation of Application Specific Instruction Set Processor for SHA-3 BLAKE Algorithm," *IEICE Trans. Electron.*, vol. E95, no. 8, pp. 1415–1426, 2012.

[10] H. L. Pham, T. H. Tranl, V. T. D. Le, and Y. Nakashima, "A Coarse Grained Reconfigurable Architecture for SHA-2 Acceleration," in *IPDPSW 2022*, Lyon, France, 2022, pp. 671–678.

[11] Y. Du, W. Li, Z. Dai, and L. Nan, "PVHArray: An Energy-Efficient Reconfigurable Cryptographic Logic Array With Intelligent Mapping," *IEEE Transactions on VLSI Systems*, vol. 28, no. 5, 2020.

[12] A. Pagán and K. Elleithy, "A Multi-Layered Defense Approach to Safeguard Against Ransomware," in *CCWC 2021*, 2021, pp. 0942–0947.

[13] J. Katz and Y. Lindell, *Introduction to Modern Cryptography*. CRC press, 2014.

[14] A. Sideris, T. Sanida, and M. Dasygenis, "Hardware Acceleration of the AES Algorithm using Nios-II Processor," in *PACET 2019*, Volos, Greece, 2019, pp. 1–5.

[15] T. K. Tran, T. P. Dang, T. T. Bui, and H. T. Huynh, "A Reliable Approach to Secure IoT Systems Using Cryptosystems Based on SoC FPGA Platforms," in *ISEE 2021*, Ho Chi Minh, Vietnam, 2021.

[16] N. At, J. L. Beuchat, E. Okamoto, San, and T. Yamazaki, "Compact Hardware Implementations of ChaCha, BLAKE, Threefish, and Skein on FPGA," *TCAS-I: Regular Papers*, vol. 61, no. 2, pp. 485–498, 2014.

[17] J.-P. Kaps *et al.*, "Lightweight Implementations of SHA-3 Candidates on FPGAs," in *INDOCRYPT 2011*, ser. Lecture Notes in Computer Science, vol. 7107. Berlin, Heidelberg: Springer, 2011.

[18] M. Kammoun, M. Elleuchi, M. Abid, and M. S. BenSaleh, "FPGA-based implementation of the SHA-256 hash algorithm," in *DTS 2020*, 2020, pp. 1–6.

[19] S. Mathew *et al.*, "340mv–1.1v, 289gbps/w, 2090-gate NanoAES hardware accelerator with area-optimized encrypt/decrypt gf(24)2 polynomials in 22nm tri-gate CMOS," in *2014 Symposium on VLSI Circuits Digest of Technical Papers*, Honolulu, HI, USA, 2014, pp. 1–2.

[20] Y. Zhang, P. Wang, X. Zhang, X. Weng, and Z. Yu, "A PUFs-based hardware authentication BLAKE algorithm in 65 nm CMOS," *International Journal of Electronics*, vol. 103, no. 6, 2016.