Secure and Energy-Efficient Edge Computing Platform with Customized RISC-V

Cuong Pham-Quoc [1,2,*] and Nguyen The Binh [1,2]

[1] Department of Computer Engineering, Faculty of Computer Science and Engineering, Ho Chi Minh City University of Technology (HCMUT), Ho Chi Minh City, Vietnam
[2] Department of Computer Engineering, Vietnam National University-Ho Chi Minh City (VNU-HCM), Thu Duc, Ho Chi Minh City, Vietnam
Email: cuongpham@hcmut.edu.vn (C.P.-Q.); binh.nguyen288@hcmut.edu.vn (N.T.B.)
*Corresponding author

***Abstract*—In recent years, many application domains, such as environmental monitoring and management, smart homes and buildings, and intelligent traffic, including parking and traffic lights, require edge computing platforms for processing data at the stations before sending preliminaries to cloud servers. Although processor-based computing platforms like Raspberry Pi or Jetson Nano/Xavier offer many advantages when applied to this system, the platforms exhibit high energy consumption, security vulnerabilities, and high non-recurring costs. Therefore, this paper introduces a secure and energy efficient edge computing platform built with Field Programmable Gate Array (FPGA) technology. This platform's customized RISC-V processor, developed based on the Open-Source RISC-V architecture, is augmented with the security algorithm SHA-256 to allow data to be encrypted before sending to cloud servers. The security approach is built as a custom instruction processor, allowing users to implement their systems without hardware design skills. The customized processor with SHA-256 extension instruction is developed with Verilog-HDL and built with different low-end/-price FPGA boards for testing and comparing with Micro:Bit and Raspberry Pi embedded boards. Compared with the Micro:Bit system, at the same price as our low-end experimental platform (GW1NR9), our proposed system achieves speedups by up to 27.01×. Compared with Raspberry Pi 4, five times more expensive than our platforms, we need a slightly longer execution time but use 4.24× less energy consumption. For a deep dive evaluating our customized processor, we will use the Dhrystone benchmark to compare it with the ARM-Cortex high-end embedded processor. Experimental results show that we achieve the value DMIPS/Hz of 1.96, better than ARM-Cortex 3 processor and other RISC-V softcore implementations.**

***Keywords*—Field Programmable Gate Array (FPGA), secured and lightweight edge devices, RISC-V, extension instructions**

I. Introduction

The rapid expansion of Internet of Things (IoT) devices dedicated to edge computing has brought about a new era of connectivity and convenience, fundamentally changing our interactions with the world around us [1]. Statista predicts there will be more than 29 B internet-connected IoT devices in 2030. These devices are confronted with a dual challenge. On one side, they must be lightweight, energy-efficient, and possess real-time processing capabilities to excel at the network's edge. Conversely, the growing complexity of IoT ecosystems exposes them to vulnerabilities that threaten data security and user privacy [2].

To confront these challenges head-on, we delve into FPGA-based lightweight IoT edge devices, focusing on security and efficiency and harnessing the potential of customized RISC-V architecture and innovative cryptography techniques. Field Programmable Gate Arrays (FPGAs) facilitate a unique fusion of hardware and software customization, presenting an opportunity to fine-tune performance and security parameters explicitly tailored to the demands of the IoT edge environment. Simultaneously, integrating the RISC-V instruction set architecture empowers developers to design processors that strike a harmonious balance between simplicity and versatility. Lastly, leveraging cryptography techniques ensures secure data transmission over the internet.

This paper introduces a novel approach wherein we propose a tailored version of the RISC-V processor designed to integrate a cryptography engine as an extended instruction seamlessly. Through these extended instructions, the encryption of data before internet transmission is facilitated. Including encryption as an extended instruction simplifies the encryption process for users while significantly bolstering data protection. In contrast to the hardware accelerator method discussed in [3], our architecture offers accelerated performance through hardware components without necessitating extensive hardware expertise from users. Furthermore, the instruction extension methodology typically entails lower overhead regarding design intricacy and resource consumption.

The initial prototype iteration of our system is realized using SystemVerilog, adhering to the single-cycle RISC-V architecture (one instruction per cycle) and employing the SHA-256 algorithm [4]. Leveraging the RISC-V

Manuscript received on June 2, 2024, revised July 10, 2024; accepted September 9, 2024; published December 23, 2024.

architecture and SHA-256 algorithm, our objective is to exploit the capabilities of lightweight yet robust processors capable of accommodating the resource limitations inherent in IoT edge deployments. Our prototype is synthesized and constructed utilizing three FPGA devices, from the cost-effective Gowin GW1NR-9 and Gowin GW2A-18 to the high-performance Xilinx Artix-7 XC7A200T. Considering the cost factor of the platforms, we conduct a comparative analysis between our customized processor implemented on the Tang Nano 9K board and the Micro:Bit system. Experimental findings indicate that our system utilizing the Tang Nano 9K FPGA board achieves up to 13.8× faster than an equivalently priced Micro:Bit board [5]. Additionally, testing conducted with the Dhrystone benchmark [6] demonstrates that our customized processor on FPGA devices achieves a performance metric of 1.32 DMIPS/Hz.

A. *Background*

1) *RISC-V architecture*

The RISC-V architecture emerges as a groundbreaking and open-source Instruction Set Architecture (ISA) that has captured widespread attention and adoption within computer architecture [7]. It has gained prominence due to its remarkable flexibility, scalability, and adaptability across various applications and industries.

Central to RISC-V is its adherence to the Reduced Instruction Set Computing (RISC) philosophy, prioritizing simplicity and efficiency in its design. However, what distinguishes RISC-V is its open nature—unlike numerous proprietary ISAs, RISC-V remains free from the control of a singular entity. Instead, it evolves collaboratively through the contributions of a global community of researchers and engineers. This collaborative effort renders RISC-V accessible for exploration, modification, and implementation without encumbering licensing fees or constraints.

This openness catalyzes innovation, empowering established tech giants and emerging startups to customize custom processors to their unique requirements. Whether for applications in embedded systems, mobile devices, highperformance computing, or specialized domains such as Internet of Things (IoT) devices, RISC-V's modular architecture fosters the development of optimized and cost-effective solutions.

Furthermore, the RISC-V architecture's modularity enables seamless customization, facilitating the integration of specialized features or extensions tailored to specific use cases. This adaptability underscores RISC-V's versatility, making it an attractive choice for a broad spectrum of computing applications.

2) *SHA-256*

Secure Hash Algorithm 256-bit (SHA-256) is a cornerstone in contemporary cryptographic practices, providing robust data integrity and security across digital landscapes [8]. Developed by the National Security Agency (NSA) of the United States, SHA-256 belongs to the SHA2 family of hash functions, serving as a critical tool for safeguarding sensitive information. This paper delves into the significance of SHA-256, exploring its advantages and diverse applications in ensuring data authenticity and protection in digital communication and information systems. SHA-256 offers the following advantages

- Robust Security: SHA-256 relies on mathematical principles that make it computationally infeasible to reverse engineer or tamper with hashed data without access to the original input. Its 256-bit output offers many possible hash values, enhancing resistance against brute-force attacks.
- Data Integrity: SHA-256's unique property, the avalanche effect, ensures that even minor changes in input data produce significantly different hash values. This characteristic makes it invaluable for verifying the integrity of transmitted or stored data and precisely detecting any unauthorized alterations.
- Versatility: SHA-256 is widely adopted across various cryptographic protocols and applications. From digital signatures and password storage to secure communication protocols like TLS and SSL, SHA-256 provides a robust foundation for ensuring data security in diverse scenarios.
- Efficiency: Despite its formidable security features, SHA-256 offers relatively fast computation times, making it suitable for real-time applications where efficiency is crucial. Its efficiency ensures minimal overhead in cryptographic operations while maintaining high levels of security.

Standardization: SHA-256's status as a widely accepted and standardized cryptographic hash function lends credibility and interoperability to systems and applications that employ it. Its endorsement by prominent organizations like NIST enhances its trustworthiness and widespread adoption in commercial and governmental contexts.

B. *Contributions*

The main contributions of the paper can be summarized in three ways:

- We introduce the customized RISC-V instruction extension approach for building secure and energy-efficient FPGA-based edge computing platforms that do not require hardware skills from users but still allow them to exploit advantages of both FPGA technology and general-purpose processor.
- We exploit the SHA-256 cryptographic algorithm as new extension instructions to validate our approach. The extension instructions allow users to execute the algorithm using a hardware-dedicated core and only high-level programming languages.
- We present our prototype version and results with the SHA-256 algorithm implemented with SystemVerilog and various FPGA devices for future study comparison.

C. *Organization*

The remainder of the paper is structured as follows: Section II provides a literature review of the research topic. Our proposed architecture is delineated in Section III-A.

The FPGA-based implementation of our prototype system is detailed in Section III-B. Section IV elaborates on our experimental procedures and analyses. Finally, Section V summarizes our findings and draws conclusions.

II. Literature Review

In recent years, a growing body of research has focused on the design and implementation of secure Internet of Things (IoT) devices and edge computing platforms using Field Programmable Gate Arrays (FPGAs). While several proposed systems have employed hardware accelerator approaches to enhance security, our approach in this paper diverges by utilizing instruction extensions. These extensions allow us to tailor the FPGA architecture for IoT security tasks, resulting in a more flexible and efficient solution. Notably, our work targets cost-effectiveness by implementing these security features on midrange FPGAs, which strikes a balance between performance and affordability.

Numerous researchers, including Samir *et al.* [9], Chen *et al.* [10], Soliman *et al.* [11], Sekar *et al.* [12], Cano-Quiveu *et al.* [13], Gomes *et al.* [14], Meenakshi and Devi [15], Damodharan *et al.* [16], and Lin *et al.* [17] have contributed to this area by presenting their systems for securing edge devices, employing various security algorithms. These systems are typically synthesized on Xilinx FPGAs for testing, with some studies providing details on working frequencies and throughput. Bhoyar *et al.* [18] introduced a VHDL-based implementation of 128-bit AES to enhance IoT data security, as described in their work. Parikibandla *et al.* [19] designed a system that integrates the Lorenz Chaotic Circuit with a Dualport Read Only Memory-based present algorithm on an FPGA Virtex-6 board, specifically tailored for IoT sensor nodes, as discussed in their study. Rajput *et al.* [20] focused on developing VLSI architectures for WiMax/IoT MAES security methods, emphasizing lightweight cryptography with reduced complexity, as outlined in their publication, with experimental results and simulations demonstrating effective operation at 23 MHz. In a separate study, Lin *et al.* [21] detailed an FPGA-based implementation for a secure edge computing device, with a particular emphasis on data confidentiality. Their work, presented in Ref. [21], was evaluated using the Altera Cyclone II DE2-70 board at a working frequency of 50 MHz.

III. Method

In this section, we first present our proposed architecture for a lightweight and secure edge computing platform based on FPGA and RISC-V processors. We then introduce our first prototype implementation to verify the proposed architecture and evaluate the system's lightweight and security.

A. Proposed Architecture

This section first shows an overview of an edge computing system using our proposed platforms with customized RISCV processors for security and energy efficiency. We then focus on the detailed structure of the proposed platform.

1) Overview system

Fig. 1 comprehensively depicts the edge computing system architecture, highlighting the function of our proposed secure and efficient edge device. Our FPGA-based IoT edge device, designed for security and energy efficiency, serves as a gateway for gathering sensor data, performing initial telemetry data processing, and transmitting data to cloud services for further processing. The design of the proposed device is rooted in the RISC-V architecture and is tailored to security measures. Given that IoT edge devices are typically powered by energy harvested from solar panels or batteries, the device's lightweight nature and low energy consumption are critical requirements.

On the one hand, to take the critical requirements of energy consumption into account, the RISC-V processor is customized and built with FPGA devices to keep only necessary machine instructions. On the other hand, to provide security effectively, a dedicated cryptography approach should be developed as an extension instruction so that users can exploit it quickly. Since we target lightweight edge computing platforms with limitations of energy and computational resources, our approach in this paper is to customize the RISC-V processor with an SHA-256 extension instruction targeting FPGA platforms. Our edge device can collect telemetry data from sensors or recording equipment through local networks such as LoRa, wireless networks, or even cable networks. In contrast, the device communicates with cloud services through an internet connection.

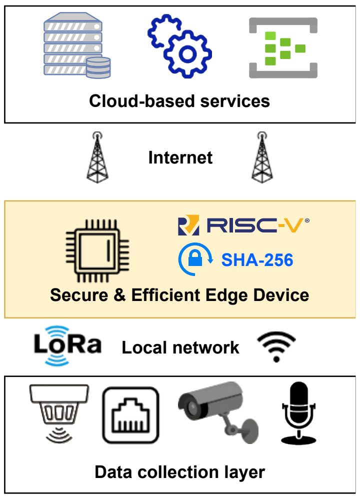


Fig. 1. The overview of an edge computing system with our secure and energy-efficient platform.

2) Customized secured and lightweight RISC-V architecture

The customized single-cycle RISC-V processor, illustrated in Fig. 2, incorporates an SHA-256 cryptography core as an extension instruction. This hardware enhancement integrates the cryptography core directly with the Arithmetic Logic Unit (ALU), facilitating the computation of security protocols without requiring sequential instructions. The controller block manages the activation of the cryptography core, responding to specific instruction opcodes associated with the execution of the extension instruction. This design enables programmers to leverage the benefits of a dedicated hardware core for enhanced performance without necessitating an in-depth understanding of hardware architecture.

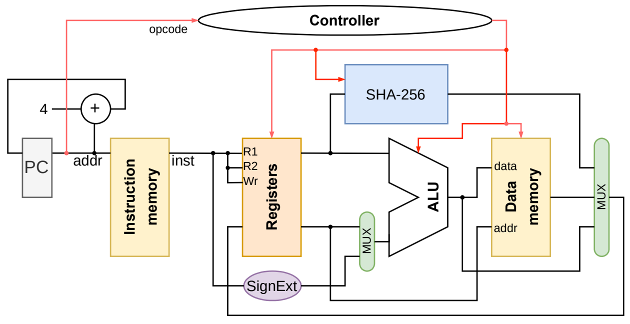


Fig. 2. The detailed architecture of our customized RISC-V processor with the SHA-256 augmented.

Despite the advantages offered by the extension instruction approach compared to traditional hardware accelerator models, it is susceptible to a significant limitation: the elongation of the critical path, potentially leading to a reduction in system operating frequency. However, our target deployment scenario focuses on edge computing platforms where frequency considerations are not paramount. Consequently, this approach effectively streamlines system complexity and minimizes resource demands associated with interconnecting computing cores inherent in the hardware accelerator model.

In addition to incorporating the augmented secured core as an extension instruction, in this work, the customized processor necessitates the ability to communicate with sensor devices via I2C or UART protocols for collecting data. However, accessing devices through these protocols requires access to the data memory address. Hence, this figure does not show this ability with the customized RISC-V processor. When implementing the system with FPGA boards, we explain the details of address multiplexers for accessing different components, including data memory, Ethernet, or UART.

3) SHA-256 extension instruction

We offer extension instructions to allow users to exploit the SHA-256 core efficiently. To conduct an SHA hasing process, the dedicated core should be reset (by sha2rst instruction) before transferring data input to the core (sha2push instruction). All SHA-256 extension instructions are R-type instructions; hence, we use the R-type RISC-V instruction format for encoding them. Fig. 3 presents the R-type instruction format of a RISC-V processor, while Table I shows our proposed extension instructions for SHA-256 computation.

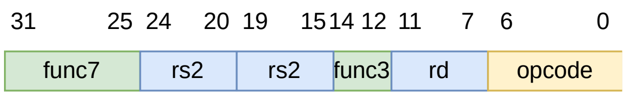


Fig. 3. The R-type instruction format in the RISC-V processor.

TABLE I. SHA INSTRUCTION EXTENSIONS

Instructions	Opcode	Func7	Func3	Example	Description
sha2rst	0x0F	1	0	sha2rst	Start new SHA-256 stream
sha2push	0x0F	2	0	sha2push x0	Push register x0 to current block
sha2start	0x0F	4	0	sha2start	Start new block in current stream
sha2perform	0x0F	8	0	sha2perform	Perform 1 round of SHA-256
sha2finish	0x0F	16	0	sha2finish	Finish current block, update digest
sha2read	0x0F	32	0	sha2read x1, x2	Read a word at index in register x1 to register x2

B. Prototype Implementations with FPGA Boards

This section presents our prototype implementation of the proposed customized secured and lightweight RISC-V on FPGA devices. In this prototype version, the cryptography module is built with the SHA-256 algorithm, i.e., extension instructions for encoding data with SHA-256 will be introduced. We also discussed the details of our interface for interacting with UART, Ethernet, and the main memory.

1) FPGA-based system implementation

We implement our prototype versions with different FPGA boards to evaluate and test the proposed system. Fig. 4 presents the implementation of our system on FPGA boards. To access different peripherals such as main memory, UART connections, and Ethernet ports, we build interfaces for those hardware modules and connect them to an onboard system bus. From the FPGA chip side, along with the customized RISC-V processor, we make an addr_valid module for select hardware modules the processor would like to connect.

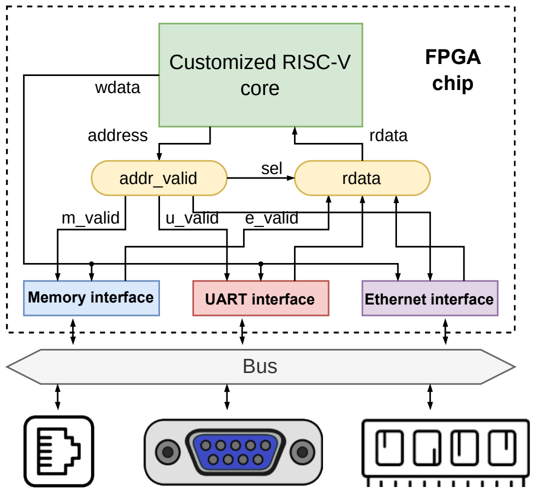
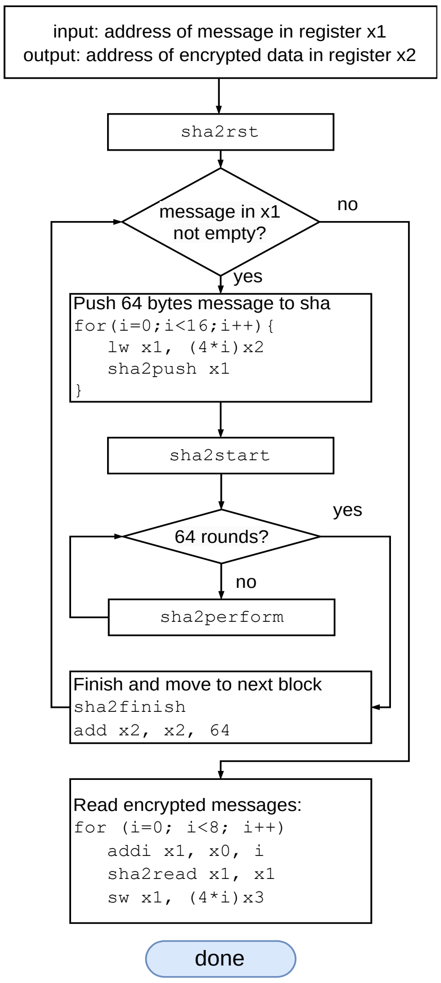


Fig. 4. FPGA-based implementation for our proposed system.

An rdata module is used for multiplex data when the processor needs to load data from these connections. The modules, including the RISCV customized processor, are developed with SystemVerilog and synthesized by design suites provided by FPGA vendors. The experimental results section will introduce details of the FPGA boards used for our experiments.

2) SHA-256 processing with custom instructions

The flowchart in Fig. 5 is used to fully exploit our proposed system with SHA-256 extension instructions for encrypting data. Before starting the new encrypting process, the SHA-256 hardware core should be reset by the sha2rst instruction. The sha2push extension instruction transfers original messages to the core. Since the SHA-256 algorithm requires a 64-byte block for each hashing, 16× 4-byte pushes are performed. The sha2start instruction is invoked to start the hashing process after data movement behaviors.


input: address of message in register x1
output: address of encrypted data in register x2
sha2rst
message in x1 not empty?
no
yes
Push 64 bytes message to sha
sha2start
64 rounds?
yes
no
sha2perform
Finish and move to next block
done

```
for(i=0;i<16;i++){
   lw x1, (4*i)x2
   sha2push x1
}
```

```
sha2finish
add x2, x2, 64
```

Read encrypted messages:

```
for (i=0; i<8; i++)
   addi x1, x0, i
   sha2read x1, x1
   sw x1, (4*i)x3
```

Fig. 5. The flowchart for processing SHA-256 algorithm with our extension instructions.

After issuing the sha2start instruction to activate the SHA-256 core, 64 rounds of hashing according to the SHA algorithm are executed by the sha2perform instruction. The sha2finish instruction is executed following to finalize the encryption process of the current 64-byte block before moving to the next block by increasing the address value in the base address register (x2 register in the flowchart). In case there still exists data for further encrypting, the process is repeated; otherwise, the RISC-V processor starts copying data from the SHA-256 core to the main memory by conducting iterations of sha2read and sw (store word) instructions.

IV. Experimental Results and Discussion

This section delineates the experimental methodologies employed, followed by an exposition of the synthesis outcomes across various FPGA devices. Subsequently, a com prehensive examination and juxtaposition of performance metrics and power consumption are provided for scrutiny.

A. Experimental Setup

For the synthesis of our custom RISC-V processor integrated with SHA-256 functionality, we employed three distinct FPGA devices: the low-end Gowin GW1NR-9 and Gowin GW2A-18, alongside the mid-range Xilinx Artix 7 35T. The former two devices represent cost-effective solutions suited for edge computing applications, while the latter serves as a mid-end benchmark for comparison purposes. The implementation detailed in Section III-B underwent synthesis using Gowin EDA and Vivado 2022 tools for the Gowin and Xilinx FPGA platforms, respectively.

Subsequently, SHA-256 hashing operations were conducted using varying data sizes to facilitate a comparative analysis between our systems and the Microbit and Raspberry Pi 4 platforms. These platforms were also utilized to assess energy consumption metrics. Additionally, the performance evaluation of our custom RISC-V processor was augmented through the utilization of the Dhrystone benchmark.

B. Synthesis Results

Table II presents the synthesis outcomes derived from our prototype implementations on three distinct FPGA platforms: the Gowin GW1NR-9 and GW2A-18, representative of costeffective FPGA solutions, and the Xilinx Artix 7 35T (A35T), exemplifying a mid-range FPGA device. Our system exhibits LUT resource utilization of up to 18% and 42% on the Gowin2A-18 and GW1NR-9 platforms, respectively, as illustrated in the table.

TABLE II. Synthesis Results of Our Prototype Implementation with Various FPGA Boards

Resources	GW1NR-9	GW2A-18	Xilinx A35T
LUTs	3570 41.3%	3707 17.9%	2050 6.2%
FFs	1141 17.6%	1142 7.3%	1243 3.0%
BRAMs (Kbit)	432 92.3%	432 52.2%	432 24%
Freq (MHz)	27	50	75
Price	$19	$29	$159
Price/MHz	$0.70	$0.58	$2.12

Regarding operational frequency, the Gowin GW1NR-9 operates at 27 MHz, while its counterpart offers a maximum frequency of 50 MHz. Conversely, the mid-range Xilinx A35T utilizes a mere 6.2% of its LUT resources and operates at a frequency of 75 MHz.

However, when considering the Price/MHz metric, the Xilinx FPGA device demonstrates less efficiency than the Gowin low-cost FPGA alternatives. Consequently, the Gowin low-cost FPGA devices emerge as optimal choices for lightweight IoT edge devices.

C. Performance Analysis

This section compares the performance (in terms of execution) of our prototype implementations with the Micro:Bit board since the price of Micro:Bit is quite similar to our low end boards. Different blocks of data with various sizes are used for this comparison. Table III shows the execution time when processing the SHA-256 algorithm with varying sizes of block using the Micro:Bit platform and our proposed system with FPGA boards. The table also presents speedups of our prototype implementations concerning the Micro:Bit platform. Experimental results in the table depict that all our prototype implementations offer higher performance than Micro:Bit. The lowest price board, GW1NR-9 achieves speedups by up to 14.6× while Gowin GW2A-18 obtains up to 27.01×. These boards are in the same price range as the Micro:Bit platform. Meanwhile, highly high speedups have been achieved by the implementations with the mid end board, Xilinx Artix A35T, by up to 40.52×. However, the mid-end board is more expensive than Micro:Bit.

D. Energy Consumption Analysis

To further substantiate the efficiency of our proposed system, we conducted additional energy consumption analysis by juxtaposing our system's performance on the Gowin GW1NR-9 board with that of the Raspberry Pi 4, a platform approximately five times costlier than our experimental setup. Table IV presents a comparative analysis of the execution time and energy consumption between the two platforms. Energy consumption is computed by multiplying the reported power consumption by the tools with the execution time. As depicted in the table, while our system exhibits longer execution times compared to the Raspberry Pi 4 platform running OpenSSL 3.0 SHA-256 on Ubuntu, it demonstrates superior energy efficiency, with savings of up to 4.24×. This underscores one of the principal objectives of our proposed system: to optimize efficiency for IoT edge computing devices operating under constrained power environments. However, other FPGA boards cannot be better than Raspberry Pi 4 in terms of power consumption because they are not dedicated to edge computing.

TABLE III. EXECUTION TIME (μS) ON WITH DIFFERENT PLATFORMS AND SPEED-UPS OF OUR PROTOTYPE IMPLEMENTATIONS WITH RESPECT TO MICRO:BIT

Block size	Micro:Bit	Gowin GW1NR-9		Gowin GW2A-18		Xilinx A35T	
		Exec. time	Speed-up	Exec. time	Speed-up	Exec. time	Speed-up
8	330	46	7.17×	25	13.20 ×	16	20.63×
16	416	46	9.04 ×	25	16.64 ×	16	26.00×
64	907	92	9.86×	49	18.51×	33	27.48×
256	2,975	227	13.10×	120	24.79×	81	36.73×
1,024	11,041	769	14.36×	420	26.23×	276	40.00×
8,192	84,959	5,830	14.56 ×	3,150	26.97×	2,100	40.46×
16,384	169,394	11,600	14.60×	6,270	27.01×	4,180	40.52×

TABLE IV. ENERGY CONSUMPTION (μJ) COMPARISON BETWEEN OUR PROTOTYPE IMPLEMENTATIONS AND RASPBERRY PI 4 PLATFORMS

Block size	Raspberry Pi 4		Gowin GW1NR-9		Gowin GW2A-18		Xilinx A35T	
	Time (μs)	Energy	Energy	Reduction	Energy	Reduction	Energy	Reduction
16	1.02	4.49	1.06	4.24×	3.60	1.25×	4,08	1,10×
64	1.49	6.56	2.12	3.10×	7.06	0.93×	8,42	0,78×
256	2.83	12.74	5.22	2.44×	17.28	0.74×	20.66	0.62×
1024	8.25	37.13	17.69	2.10×	60.47	0.61×	70.38	0.53×
8192	58.55	263.48	134.09	1.96×	453.54	0.58×	535.50	0.49×
16384	115.86	544.54	266.80	2.00×	902.77	0.59×	1065.90	0.50×

E. Benchmark Evaluation

TABLE V. DHRYSTONE BENCHMARK RESULTS COMPARISON

Part	DMIPS	DMIPS/MHz
Our on GW1NR-9 @ 27MHz	52.92	1.96
Our on GW2A-18 @ 50MHz	98	1.96
Our on A35T @ 75MHz	147	1.96
VexRISCV (RV32IM) @ 27 MHz	37.26	1.38
PicoRV32 @ MHz 214.65	N/A	0.516
ARM Cortex M3 @	24.42	1.32

We used the Dhrystone benchmark to compare our processor with the referenced RISC-V processor and ARM Cortex M3 processor in terms of DMIPS and normalized to frequency. Table V presents the evaluation results of our processors, VexRISC-V 32 processor [22], PicoRV32 [23], and AMR Cortex M3 at 18.5 MHz [24], according to the Dhystone benchmark. As shown in the table, our processor outperforms others in terms of DMIPS/MHz values.

F. Discussion

In this paper, we introduce the lightweight, secure, and energy-efficient edge computing platform architecture. This architecture addresses edge computing platforms' security and energy consumption challenges by applying the custom RISC-V processor design and FPGA boards. We validate the architecture with low-cost FPGA boards. Hence, the system suffers from the following challenges.

- Computational Power: RISC-V processors, especially those designed to be lightweight, may not match the performance of more established architectures (e.g., ARM or x86) for specific high-demand applications.
- Scalability: Lightweight RISC-V processors may face challenges in scaling performance to handle large-scale or highly complex computations compared to more robust, high-performance computing platforms.
- Maintenance: Maintaining and updating a highly customized hardware platform can be complex, especially when dealing with evolving security threats and performance optimizations.

While these challenges are indeed open issues, they also present exciting opportunities for future research to enhance and refine the proposed system, thereby addressing the current landscape of edge computing and IoT security.

V. Conclusions

Over recent years, various application domains, including environmental monitoring and management, smart home and building automation, and intelligent traffic management encompassing parking and traffic signal control, have necessitated the utilization of edge computing platforms. These platforms enable data processing at local stations prior to transmitting pertinent information to cloud servers. Despite the advantages offered by processor-based computing platforms like Raspberry Pi or Jetson Nano/Xavier in these systems, they are plagued by high energy consumption, security vulnerabilities, and significant non-recurring costs. Consequently, in this paper, we propose the development of a secure and energy-efficient edge computing platform leveraging FPGA technology. Central to this platform is a customized RISC-V processor, constructed based on the Open-Source RISC-V architecture, enhanced with the SHA-256 security algorithm to facilitate data encryption before transmission to cloud servers. The security framework is implemented as a custom instruction within the processor, enabling users to deploy their systems without necessitating hardware design expertise. The customized processor with the SHA-256 extension instruction is realized using VerilogHDL and evaluated across various low-end, cost-effective FPGA boards for comparative analysis with Micro:Bit and Raspberry Pi embedded boards. Notably, when compared with the Micro:Bit system at an equivalent price point to our low-end experimental platform (GW1NR-9), our proposed system achieves speedups of up to 27.01×. Furthermore, in comparison with the Raspberry Pi 4, a platform five times more expensive than ours, although we observe slightly longer execution times, we achieve a remarkable 4.24 times reduction in energy consumption. To delve deeper into the evaluation of our customized processor, we employ the Dhrystone benchmark for comparison with high-end ARM Cortex embedded processors. Experimental findings demonstrate that our system attains a DMIPS/Hz value of 1.96, surpassing that of ARM Cortex 3 processors and other RISCV softcore implementations.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

C.P.-Q. proposed the idea and wrote the paper; N.T.-B. implemented, evaluated, and tested the. All authors had approved the final version.

Acknowledgment

We acknowledge Ho Chi Minh City University of Technology (HCMUT), VNU-HCM for supporting this study.

References

[1] M. Caprolu, R. di Pietro, F. Lombardi, and S. Raponi, "Edge computing perspectives: Architectures, technologies, and open security issues," in *Proc. the 2019 IEEE International Conference on Edge Computing (EDGE)*, 2019, pp. 116–123.

[2] R. Roman, J. Lopez, and M. Mambo, "Mobile edge computing, fog: A survey and analysis of security threats and challenges," *Future Generation Computer Systems*, vol. 78, pp. 680–698, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X16305635

[3] C. Pham-Quoc, J. Heisswolf, S. Werner, Z. Al-Ars, J. Becker, and K. Bertels, "Hybrid interconnect design for heterogeneous hardware accelerators," in *Proc. the 2013 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2013, pp. 843–846.

[4] S. Gueron, S. Johnson, and J. Walker, "SHA-512/256," in *Proc. the 2011 Eighth International Conference on Information Technology: New Generations*, IEEE, 2011, pp. 354–358.

[5] J. Austin, H. Baker, T. Ball, J. Devine, J. Finney, P. de Halleux, S. Hodges, M. Moskal, and G. Stockdale, "The BBC Micro: Bit: From the uk to the world," *Communications of the ACM*, vol. 63, no. 3, pp. 62–69, 2020.

[6] R. P. Weicker, "Dhrystone: A synthetic systems programming benchmark," *Communications of the ACM*, vol. 27, no. 10, pp. 1013–1030, 1984.

[7] A. Dorflinger, M. Albers, B. Kleinbeck, Y. Guan, H. Michalik, R. Klink, C. Blochwitz, A. Nechi, and M. Berekovic, "A comparative survey of open-source application-class RISC-V processor implementations," in *Proc. the 18th ACM International Conference on Computing Frontiers*, 2021, pp. 12–20.

[8] H. Gilbert and H. Handschuh, "Security analysis of SHA-256 and sisters," in *Proc. the International Workshop on Selected Areas in Cryptography*, 2003, pp. 175–193.

[9] N. Samir, A. S. Hussein, M. Khaled, A. N. El-Zeiny, M. Osama, H. Yassin, A. Abdelbaky, O. Mahmoud, A. Shawky, and H. Mostafa, "ASIC and FPGA comparative study for IoT lightweight hardware security algorithms," *Journal of Circuits, Systems and Computers*, vol. 28, no. 12, 2019.

[10] Z. Chen, S. Guo, J. Wang, Y. Li, and Z. Lu, "Toward FPGA security in IoT: A new detection technique for hardware trojans," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 7061–7068, 2019.

[11] S. Soliman, M. A. Jaela, A. M. Abotaleb, Y. Hassan, M. A. Abdelghany, A. T. Abdel-Hamid, K. N. Salama, and H. Mostafa, "FPGA implementation of dynamically reconfigurable iot security module using algorithm hopping," *Integration*, vol. 68, pp. 108–121, 2019.

[12] S. R. Sekar, S. Elango, S. P. Philip, and A. D. Raj, "FPGA implementation of ECC enabled Multi-Factor Authentication (e-MFA) protocol for IoT based applications," in *Proc. the Second International Conference on Microelectronic Devices, Circuits and Systems, ICMDCS 2021*, Springer, 2021, pp. 430–442.

[13] G. Cano-Quiveu, P. R. Vazquez, M. J. Bellido, J. Juan-Chico, J. Viejo-Cortes, D. Guerrero-Martos, and E. Ostua-Aranguena, "Embedded luks (e-luks): A hardware solution to iot security," *Electronics*, vol. 10, no. 23, 3036, 2021.
[14] T. Gomes, P. Sousa, M. Silva, M. Ekpanyapong, and S. Pinto, "FACV: An FPGA-based AES coprocessor for RISC-V," *Journal of Low Power Electronics and Applications*, vol. 12, no. 4, 50, 2022.
[15] S. Meenakshi and M. N. Devi, "Configuration security of FPGA in IoT using logic resource protection," in *Proc. the International Conference on Advances in Electrical and Computer Technologies, ICAECT 2021*, Springer, 2022, pp. 625–633.
[16] J. Damodharan, E. R. S. Michael, and N. Shaikh-Husin, "High throughput present cipher hardware architecture for the medical IoT applications," *Cryptography*, vol. 7, no. 1, 6, 2023.
[17] J.-L. Lin, P.-Y. Zheng, and P. C.-P. Chao, "A new ECC implemented by FPGA with favorable combined performance of speed and area for lightweight IoT edge devices," *Microsystem Technologies*, pp. 1–10, 2023.
[18] D. B. Bhoyar, S. R. Wankhede, and S. K. Modod, "Design and implementation of aes on fpga for security of iot data," in *Proc. 2019 4th International Conference on Internet of Things and Connected Technologies (ICIoTCT)*, Springer, 2020, pp. 376–383.
[19] S. Parikibandla and A. Sreenivas, "FPGA performance evaluation of present cipher using lcc key generation for IoT sensor nodes," in *Proc. the Fifth Micro-electronics, Electromagnetics and Telecommunications (ICMEET 2019*), Springer, 2021, pp. 371–379.
[20] G. S. Rajput, R. Thakur, and R. Tiwari, "VLSI implementation of lightweight cryptography technique for FPGA-IoT application," *Materials Today: Proceedings*, 2023.
[21] W.-C. Lin, P.-K. Huang, C.-L. Pan, and Y.-J. Huang, "FPGA implementation of mutual authentication protocol for medication security system," *Journal of Low Power Electronics and Applications*, vol. 11, no. 4, 48, 2021.
[22] SpinalHDL. A FPGA friendly 32-bit RISC-V CPU implementation. [Online]. Available: https://github.com/SpinalHDL/VexRiscv
[23] Picorv32-a size-optimized RISC-V CPU. [Online]. Available: https://github.com/YosysHQ/picorv32
[24] ARMDeveloper. Dhrystone benchmarking for arm cortex processors—Application note 273. [Online]. Available: https://developer.arm.com/documentation/105958/latest/