# FPGA-based Secured and Efficient Lightweight IoT Edge Devices with Customized RISC-V

Nguyen The Binh[1,2], Binh Kieu-Do[3], Trong-Thuc Hoang[3], Pham Cong-Kha[3], Cuong Pham-Quoc[1,2,*]
[1]Ho Chi Minh City University of Technology (HCMUT), Ho Chi Minh City, Vietnam
[2]Vietnam National University - Ho Chi Minh City (VNU-HCM), Ho Chi Minh City, Vietnam
[3]University of Electro-Communications (UEC), Tokyo, Japan
* corresponding author: cuongpham@hcmut.edu.vn

2023 RIVF International Conference on Computing and Communication Technologies (RIVF) | 979-8-3503-1584-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/RIVF60135.2023.10471777

***Abstract*—In recent years, the IoT edge computing paradigm has been used in various domains such as environmental management, smart office and home, video surveillance, etc. However, most systems rely on processor-based embedded boards like Raspberry Pi or Micro that require high power consumption but lack security. This paper presents our FPGA-based secured and lightweight architecture and implementation for IoT edge devices. The proposed approach is based on the RISC-V architecture with our customization to allow data to be encrypted and decrypted securely with the SHA-256 algorithm. The secured algorithm is designed and then implemented by Verilog as an extension instruction of the RISC-V processor to accelerate encryption performance. Our proposed system is built with Gowin and Xilinx FPGA to compare performance with Micro:bit and Raspberry Pi embedded systems. For testing customized RISC-V processors, the Dhrystone benchmark is used. The experimental results show that our system achieves speed-ups by up to 13.8× compared to Mirco:bit. Although our system requires more execution time than Raspberry Pi 4, it is up to 3.10× energy-efficient than the Raspberry platform. When evaluated with the Dhrystone benchmark, our proposed customized RISC-V processor obtains the value DMIPS/MHz of 1.32, better than ARM Cortex 3 running at 18.5 MHz.**

***Index Terms*—FPGA technology, Secured and lightweight edge devices, RISC-V, Extension instructions**

## I. Introduction

The rapid proliferation of Internet of Things (IoT) devices for edge computing has ushered in an era of interconnects and convenience, revolutionizing how we interact with our surroundings [1]. These devices, however, face a dichotomy of challenges. On one hand, they need to be lightweight, energy-efficient, and capable of real-time processing to thrive at the network's edge. On the other hand, the increasing complexity of IoT ecosystems exposes them to vulnerabilities that can compromise data security and user privacy [2].

To address these dual challenges, we delve into the domain of FPGA-based secured and efficient lightweight IoT edge devices, leveraging the power of customized RISC-V architecture [3] and novel cryptography approaches. Field-Programmable Gate Arrays (FPGAs) enable a unique synergy between hardware and software customization, offering the potential to optimize performance and security parameters tailored to the IoT edge environment. Meanwhile, the integration of the RISC-V instruction set architecture empowers developers to craft processors that strike a balance between simplicity and versatility. Finally, exploiting cryptography approaches allows data to be transferred securely over the internet.

In this paper, we propose a customized version of the RISC-V processor to integrate a cryptography engine as an extended instruction. The extended instructions allow data to be encrypted before being transferred over the internet. As an extended instruction, the encryption process becomes simple for users but powerful for data protection. Compared to the hardware accelerator approach [4], our architecture allows performance to be accelerated by hardware components while not requiring much hardware knowledge from users.

The first prototype version of our system is implemented by SystemVerilog based on the single-cycle RISC-V architecture (one instruction per cycle) and SHA-256 algorithm [5]. Leveraging the RISC-V architecture and SHA-256 algorithm, we aim to harness the potential of lightweight yet robust processors that are capable of addressing the resource constraints of IoT edge deployments. Our prototype version is synthesized and built with three FPGA devices ranging from cheap Gowin GW1NR-9 and Gowin GW2A-18, to high-end Xilinx Artix-7 XC7A200T. Taking the prices of the platforms into account, we compare our customized processor built in the Tang Nano 9K board and the Micro:bit system. Experimental results show that our system with the Tang Nano 9K FPGA board achieves up to 13.8× speed-ups compared to the same price Micro:bit board [6]. When testing with the Dhrystone benchmark [7], our customized processor on FPGA devices achieves 1.32 DMIPS/Hz.

The key contributions of the paper can be summarized as follows.

1) We propose customizing the RISC-V processor to integrate security approaches for lightweight and secured FPGA-based IoT edge devices. The customization is scalable when different security algorithms can be deployed. Using the instruction extension approach, the complexity of the hardware system is hidden from users. Users do not have much hardware knowledge for building systems like the hardware accelerator approach but still exploit the high-performance characteristics of FPGA devices.
2) We present our prototype version and results with the SHA-256 algorithm implemented with SystemVerilog and various FPGA devices for future study comparison.

The rest of the paper is organized as follows. Section II

979-8-3503-1584-4/23/$31.00 ©2023 IEEE

presents the background of our work and related work in the literature. We present our proposed architecture in Section III. The FPGA-based implementation of our prototype system is introduced in Section IV. We discuss our experiments with different analysis in Section V. Finally, Section VI concludes our paper.

## II. Background and Related work

In this section, we briefly introduce the RISC-V architecture and SHA-256 algorithm. We summarize IoT security proposals in the literature.

### A. Background

*1) RISC-V architecture:* The RISC-V architecture is a revolutionary and open-source instruction set architecture (ISA) that has garnered significant attention and adoption in computer architecture. Born out of the need for a flexible, scalable, and customizable ISA, RISC-V offers a foundation for designing processors and computing systems across various applications and industries.

At its core, RISC-V follows the Reduced Instruction Set Computing (RISC) philosophy, emphasizing simplicity and efficiency in its design. What sets RISC-V apart is its open nature – unlike many proprietary ISAs, RISC-V is not controlled by a single entity. Instead, it is developed collaboratively by a global community of researchers and engineers, making it accessible for anyone to study, modify, and implement without licensing fees or restrictions.

This openness has led to a surge of innovation, allowing established tech companies and startups to create custom processors tailored to their specific needs. Whether for embedded systems, mobile devices, high-performance computing, or even specialized applications like Internet of Things (IoT) devices, RISC-V's modular architecture enables the creation of optimized and cost-effective solutions.

*2) SHA-256:* SHA-256, which stands for Secure Hash Algorithm 256-bit, is a widely used cryptographic hash function that plays a crucial role in ensuring data integrity and security across digital communication and information systems. Developed by the National Security Agency (NSA) of the United States, SHA-256 is part of the SHA-2 family of hash functions.

At its core, SHA-256 takes an input message of variable length and processes it into a fixed-size 256-bit hash value. This hash value is designed to be unique to the input data, making even the slightest change in the input result in a vastly different output hash. This property, known as the avalanche effect, is essential for verifying data authenticity and detecting tampering.

SHA-256's applications are diverse and far-reaching. It is commonly used in digital signatures, password storage, data verification, and cryptographic protocols such as Transport Layer Security (TLS) and Secure Sockets Layer (SSL). By generating a unique and deterministic hash value, SHA-256 ensures that the integrity of data remains intact during transmission and storage, making it extremely difficult for unauthorized parties to modify the data without detection.

### B. Related work

In recent years, many studies have existed for the design and implementation of secured IoT devices on FPGA. However, most of the proposed systems use the hardware accelerator approaches instead of instruction extensions like our approach in this paper. Moreover, these systems are mainly implemented on high-end technology FPGAs, which are not cost-efficient like our work.

Samir et al. [8], Z. Chen et al. [9], Meenakshi et al. [10], Soliman et al. [11], Sekar et al. [12], Cano-Quiveu et al. [13], Gomes et al. [14], Damodharan et al. [15], and Lin et al. [16] present their systems for secured IoT devices with various security algorithms. All of them are synthesized with Xilinx FPGAs for testing and validating the proposed system. Working frequencies and throughput are reported in some studies.

Bhoyar et al. presented a 128-bit AES implementation using VHDL to enhance the security of IoT data, as documented in their work [17]. The simulation was conducted using ISIM. Parikibandla et al. devised a system involving the Lorenz Chaotic Circuit integrated with a Dual-port Read Only Memory-based PRESENT Algorithm on an FPGA Virtex-6 board, designed for IoT sensor nodes, as discussed in their study [18]. Rajput et al. developed VLSI architectures for WiMax/IoT MAES security methods, focusing on lightweight cryptography with reduced complexity, as outlined in their publication [19]. Experimental results, coupled with simulations, demonstrate the system's effective operation at 23 MHz.

In another endeavor, Lin et al. detailed an FPGA-based implementation for a secure edge computing device emphasizing safeguarding data confidentiality. Their contribution, presented in [20], was assessed using the Altera Cyclone II DE2-70 board at a working frequency of 50 MHz.

## III. Proposed architecture

This section presents an overview of the IoT system architecture based on our proposed secured and efficient lightweight IoT devices. We then dive deeply into the architecture of the customized RISC-V processor to support the security and lightweight ability.

### A. Overview system

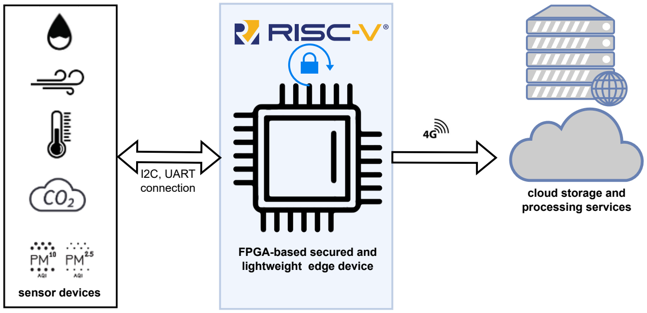
RISC-V
I2C, UART connection
4G
PM10
PM2.5
AQI
AQI
sensor devices
FPGA-based secured and lightweight edge device
cloud storage and processing services

Fig. 1. The overview IoT system and the role of our proposed FPGA-based secured and lightweight edge device

Figure 1 illustrates the overview of an IoT system architecture and the role of our proposed edge device. Our FPGA-based secured and lightweight IoT edge device is used as a gateway for collecting data from sensors, preprocessing telemetry data, and transferring data to cloud storage. The proposed device is designed based on the RISC-V architecture and customized with security approaches. Since IoT edge devices are usually deployed with energy harvesting from solar panels or batteries, lightweight and low power consumption are essential demands.

### B. Customized secured and lightweight RISC-V architecture

Figure 2 presents our customized single-cycle RISC-V processor augmented with a cryptography core as an extension instruction. As shown in the figure, the hardware cryptography core is built along with the ALU for calculating security approaches instead of using a sequence of instructions. The controller block considers instruction opcodes to activate the cryptography core if the extension instruction is executed. By using this approach, programmers do not need to have much hardware knowledge, but they can exploit the advantages of a hardware-dedicated core to improve performance. Although the extension instruction approach offers benefits compared to the hardware accelerator model, it suffers from the main drawback of the long critical path that may reduce the working frequency of the system. However, we are targeting edge computing platforms where frequency is not a critical parameter. Hence, the approach helps minimize the system's complexity and resources required for interconnecting computing cores in the hardware accelerator model.

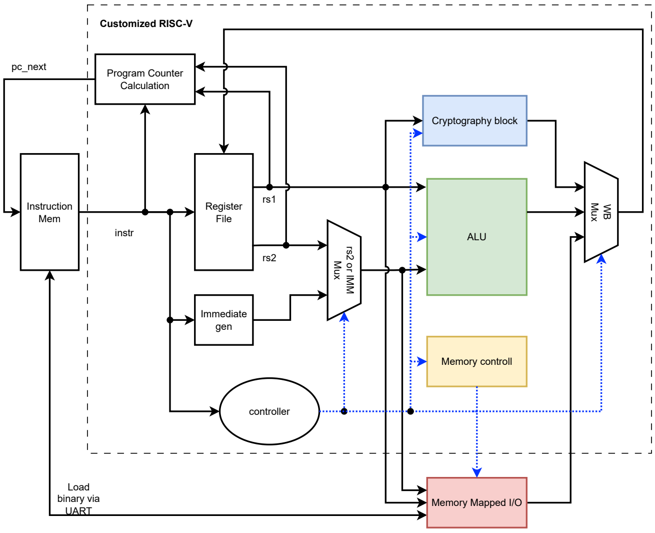

Fig. 2. The proposed customized RISC-V architecture augmented with a cryptography core

Along with the augmented secured core as an extension instruction, the customized processor should support data communication with sensor devices through I2C or UART protocol. Therefore, we build the memory-mapped I/O module for handling data access. Depending on instructions and specific platforms, the module will access data from UART communication, Ethernet, or main memory. Details of this implementation will be discussed in the next section.

## IV. FPGA-based Edge Computing Platform Implementation

In this section, we present our first 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. Besides, we discuss the details of our memory-mapped IO module for interacting with UART, Ethernet, and the main memory.

### A. SHA-256 extension instruction

Table IV-A presents all the extension instructions used for SHA-256 and executed by the cryptography block. We use the RISC-V R-type format for the extension. The most significant bits from 31 to 25 are used as opcodes. In this implementation, we use one-hot encoding for our SHA instructions.

The pseudo code in Algorithm 1 illustrates the use of these instructions for encoding data with the SHA-256 method.

**Algorithm 1:** The process for encoding data with SHA-256 using our extension instructions

```
Data: reg x2 store byte stream
Result: reg x3 sha256 digest of length 32 bytes
        sha2_hash
sha2rst x0, x0, x0; //reset hardware block
while still have data on x2 do
    //push 64 bytes data for the next block
    for $i \leftarrow 0; i < 16; i++$ do
        lw x1, (4*i)(x2);
        sha2push x0, x1, x0;
    end
    sha2start x0, x0, x0 ;  //start performing the block
    repeat
        sha2perform x0, x0, x0 ;  //perform one round
    until 64;
    sha2finish x0, x0, x0 ;   //finish current block
    add x2, x2, 64 ;   //move on to next 64 bytes block
end
//read the 32 bytes digest to x3
for $i \leftarrow 0; i < 8; ++i$ do
    ori x1, x0, i;
    sha2read x1, x1, x0 ; //read digest bytes [4*i+:4] to x1
    sw (4*i)(x3), x1 ; //store the result in x3
end
```

TABLE I
SHA INSTRUCTIONS EXTENSION

| Instruction | [31:25] | [24:20] | [19:15] | [14:12] | [11:7] | [6:2] | [1:0] | Description |
|---|---|---|---|---|---|---|---|---|
| sha2rst | 1 | x | x | 0 | x | 2 | 3 | Start new SHA256 stream |
| sha2push | 2 | x | rs1 | 0 | x | 2 | 3 | Push register to current block |
| sha2start | 4 | x | x | 0 | x | 2 | 3 | Start new block in current stream |
| sha2perform | 8 | x | x | 0 | x | 2 | 3 | Perform 1 round of SHA256 |
| sha2finish | 16 | x | x | 0 | x | 2 | 3 | Finish current block, update digest |
| sha2read | 32 | x | idx[2:0] | 0 | rd | 2 | 3 | Read a word at index idx in SHA destination register |

- x: don't care values
- opcode: [31:25]

### B. Memory mapped IO

As mentioned above, the first prototype version of our secured and efficient lightweight IoT edge devices with customized RISC-V processors aims to communicate with sensors through UART and Ethernet protocol. Therefore, the memory-mapped IO module should manage these communication channels based on the addresses of memory access instructions. Figure 3 presents the implementation of this module. The figure shows that we use the 16 high-significant memory address bits to distinguish peripherals. A decoder module will enable a corresponding interface for interacting with UART, Ethernet, or the processor's main memory. In this way, the customized processor accesses data from or to UART, Ethernet, or main memory only by data transfer instructions load word (lw) or store word (sw).

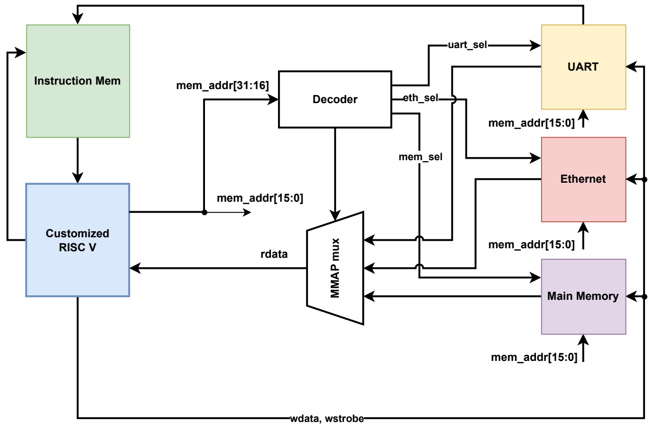

Fig. 3. FPGA-based implementation for IoT edge devices

## V. Experiments

This section presents our setups for experiments and then introduces our synthesis results with different FPGA devices. Finally, an analysis and comparison regarding performance and power consumption are discussed.

### A. Experimental setup

To synthesize our customized RISC-V processor with SHA-256, we use three FPGA devices, including Gowin GW1NR-9 and Gowin GW2A-18, to high-end Xilinx Artix-7 XC7A200T. The two first devices represent the low-cost devices suitable for edge computing, while the last is a high-end device used for comparison. The implementation presented in Section IV is synthesized with Gowin EDA and Vivado 2022 tools for the Gowin and Xilixn FPGA, respectively.

We use the Tang Nano 9K board as our edge device for the performance comparison purpose and compare the results with Microbit and Raspberry Pi 4 platforms. These platforms are also used for energy consumption analysis. To further evaluate the performance of our customized RICS-V processor, the Dhrystone benchmark is used.

### B. Synthesis results

Table II shows the synthesis results of our first prototype implementation on three different FPGA devices, including the Gowin GW1NR-9 and GW2A-18 representing low-cost FPGA platforms and Xilinx XC7A200T representing high-end FPGA devices. As shown in the table, with the low-cost FPGA devices, our system uses up to 20% and 66% LUT resources of the Gowin2A-18 and GW1NR-9, respectively. Regarding working frequency, the Gowin GW1NR-9 from the Tang Nano 9K board offers 27 MHz, while the other provides up to 50 MHz working frequency. Meanwhile, the high-end FPGA Xilinx XC7A200T consumes only 1% LUT resource and offers 92.3 MHz working frequency. However, in terms of the Price/MHz parameter, the Xilinx FPGA device is not as efficient as the Gowin low-cost FPGA devices. Therefore, the Gowin low-cost FPGA devices are suitable for lightweight IoT edge devices.

### C. Performance and energy consumption analysis

In this section, we compare the performance of the Tang Nano 9K board equipped with Gowin GW1NR-9 device and Micro:bit board due to their similar price. We also compare the Tang Nano 9K board with our customized RISC-V and the Raspberry Pi 4 board regarding performance and energy consumption.

Table V-C shows the execution time when processing the SHA-256 algorithm with different block sizes using the Micro:bit platform and our proposed system with Tang Nano 9K board. The table shows that our system outperforms the same price Micro:bit board for all block sizes. Figure 4 illustrates the speed-ups of our system compared to Micro:bit. The achieved speed-ups are stable by up to 13.8× when block sizes become large.

More energy consumption analysis to prove our proposed system's efficiency, we compare our system on Tang Nano

TABLE II
THE SYNTHESIS RESULTS OF OUR FIRST PROTOTYPE VERSION

| Part | LUT6 | LUT4 | FF | BRAM 4KB | DSP | % LUT | Max Freq | Price | Price/MHz |
|---|---|---|---|---|---|---|---|---|---|
| Xilinx XC7A200T | 1894 | N/A | 1091 | 36 | 0 | 1% | 92.3MHz | $529 | $5.73 |
| Gowin GW1NR-9 | N/A | 5622 | 1334 | 4 | 0 | 66% | 27MHz | $19 | $0.70 |
| Gowin GW2A-18 | N/A | 4147 | 1319 | 14 | 0 | 20% | 50MHz | $29 | $0.58 |

TABLE III
EXECUTION TIME ($\mu$S) COMPARISONS WITH DIFFERENT BLOCK SIZES

| Block size | Micro:bit | Tang Nano 9K |
|---|---|---|
| 8 | 330 | 64 |
| 16 | 416 | 63 |
| 32 | 588 | 61 |
| 64 | 907 | 112 |
| 128 | 1,567 | 160 |
| 256 | 2,975 | 255 |
| 512 | 5,622 | 445 |
| 1,024 | 11,041 | 825 |
| 2,048 | 21,501 | 1,585 |
| 4,096 | 42,668 | 3,106 |
| 8,192 | 84,959 | 6,147 |
| 16,384 | 169,394 | 12,229 |
| 32,768 | 338,475 | 24,394 |

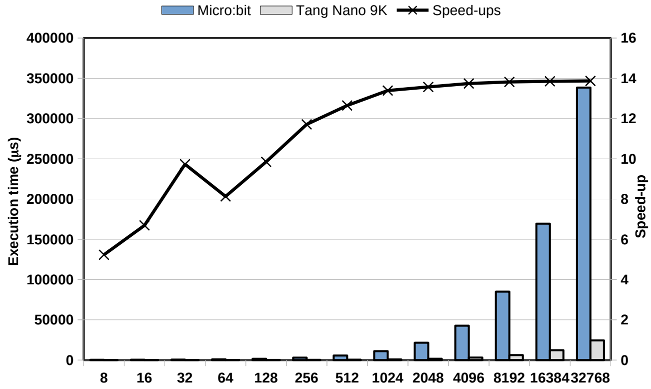

Fig. 4. Speed-ups comparison with different data size

TABLE IV
COMPARISON BETWEEN OUR PROTOTYPE VERSION ON TANG NANO 9K AND RASPBERRY PI 4 IN TERMS OF EXECUTION TIME ($\mu$S) AND ENERGY CONSUMPTION ($\mu$J)

| Block size | Raspberry Pi 4 | | Tang Nano 9K | | Energy reduction |
|---|---|---|---|---|---|
| | Time | Energy | Time | Energy | |
| 16 | 1.02 | 4.49 | 63 | 1.45 | 3.10× |
| 64 | 1.49 | 6.56 | 112 | 2.58 | 2.54× |
| 256 | 2.83 | 12.74 | 255 | 5.87 | 2.17× |
| 1024 | 8.25 | 37.13 | 825 | 18.98 | 1.96× |
| 8192 | 58.55 | 263.48 | 6,147 | 141.38 | 1.86× |
| 16384 | 115.86 | 544.54 | 12,229 | 281.27 | 1.93× |

9K board with Raspberry Pi 4 (5× more expensive than our experiment platform). Table V-C compares the execution time of the two platforms and the energy consumption (energy consumption is computed by the product of power consumption, reported by tools, and execution time). As shown in the table, our system is worse than the Raspberry Pi 4 platform, which is running OpenSSL 3.0 SHA256 on Ubuntu in terms of execution time. However, we are more efficient by up to 3.10× in energy consumption. That is one of the two goals of our proposed system: efficiency for IoT edge computing devices where power is limited.

### *D. Dhrystone benchmark evaluation*

We use the Dhrystone benchmark to assess performance and further evaluate the proposed customized RISC-V processor. Table V-D presents the evaluation results. As reported in the table, due to the high-end FPGA device, when implemented on Xilinx XCA200T, the customized processor achieves up to 122.17 DMIPS (Dhrystone millions instruction per second), that is 3.42× and 1.84×. This achievement comes from the higher working frequency of Xilinx FPGA. Regarding DMIPS/MHz, the processor achieves the value of 1.32, better than the Cortex-M3 processor running at 18.5MHz [21].

TABLE V
DHRYSTONE BENCHMARK EVALUATION FOR THE PROPOSED CUSTOMIZED PROCESSOR

| Part | Dhrystone/s | DMIPS | DMIPS/MHz |
|---|---|---|---|
| Xilinx XC7A200T | 214,651 | 122.17 | 1.32 |
| Gowin GW1NR-9 | 62,790 | 35.74 | 1.32 |
| Gowin GW2A-18 | 116,279 | 66.18 | 1.32 |

## VI. CONCLUSION

In recent times, the IoT edge computing paradigm has found application in various fields such as environmental management, smart offices and homes, video surveillance, among others. Nonetheless, a majority of these systems rely on processor-based embedded boards like Raspberry Pi or Micro:bit. These boards tend to consume high levels of power and suffer from security limitations. In this paper, we introduce our secure and lightweight architecture for IoT edge devices, which is built upon FPGA technology. Our approach is centered around the RISC-V architecture, which we have customized to enable secure encryption and decryption using the SHA-256 algorithm. To enhance encryption performance, we have designed and implemented the secure algorithm in Verilog, extending the RISC-V processor as an additional instruction. Our proposed system is prototyped using Gowin and Xilinx FPGA platforms, allowing for performance comparisons against Micro:bit and Raspberry Pi embedded systems. The Dhrystone benchmark is employed to assess the customized RISC-V processor's performance. Empirical findings reveal that our system achieves notable speed-ups of up to 13.8 times when compared to the Mirco:bit system. While our system exhibits slightly longer execution times than the Raspberry Pi 4, it demonstrates an energy efficiency improvement of up to 3.10 times over the Raspberry Pi platform. Evaluation using the Dhrystone benchmark indicates that our custom RISC-V processor achieves a DMIPS/MHz value of 1.32, outperforming the performance of an ARM Cortex 3 operating at 18.5 MHz.

## ACKNOWLEDGEMENT

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 *2019 IEEE International Conference on Edge Computing (EDGE)*, 2019, pp. 116–123.

[2] R. Roman, J. Lopez, and M. Mambo, "Mobile edge computing, fog et al.: 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] A. Waterman, Y. Lee, D. Patterson, K. Asanovic, V. I. U. level Isa, A. Waterman, Y. Lee, and D. Patterson, "The risc-v instruction set manual," *Volume I: User-Level ISA', version*, vol. 2, 2014.

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

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

[6] 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.

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

[8] 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, p. 1930009, 2019.

[9] 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.

[10] S. Meenakshi and M. Nirmala Devi, "Configuration security of fpga in iot using logic resource protection," in *Advances in Electrical and Computer Technologies: Select Proceedings of ICAECT 2021*. Springer, 2022, pp. 625–633.

[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 *Microelectronic Devices, Circuits and Systems: Second International Conference, ICMDCS 2021, Vellore, India, February 11-13, 2021, Revised Selected Papers 2*. Springer, 2021, pp. 430–442.

[13] G. Cano-Quiveu, P. Ruiz-de-clavijo 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, p. 3036, 2021.

[14] T. Gomes, P. Sousa, M. Silva, M. Ekpanyapong, and S. Pinto, "Fac-v: An fpga-based aes coprocessor for risc-v," *Journal of Low Power Electronics and Applications*, vol. 12, no. 4, p. 50, 2022.

[15] J. Damodharan, E. R. Susai Michael, and N. Shaikh-Husin, "High throughput present cipher hardware architecture for the medical iot applications," *Cryptography*, vol. 7, no. 1, p. 6, 2023.

[16] 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.

[17] D. B. Bhoyar, S. R. Wankhede, and S. K. Modod, "Design and implementation of aes on fpga for security of iot data," in *4th International Conference on Internet of Things and Connected Technologies (ICIoTCT), 2019: Internet of Things and Connected Technologies*. Springer, 2020, pp. 376–383.

[18] S. Parikibandla and A. Sreenivas, "Fpga performance evaluation of present cipher using lcc key generation for iot sensor nodes," in *Microelectronics, Electromagnetics and Telecommunications: Proceedings of the Fifth ICMEET 2019*. Springer, 2021, pp. 371–379.

[19] G. S. Rajput, R. Thakur, and R. Tiwari, "Vlsi implementation of lightweight cryptography technique for fpga-iot application," *Materials Today: Proceedings*, 2023.

[20] 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, p. 48, 2021.

[21] ARMDeveloper, "Dhrystone benchmarking for arm cortex processors - application note 273," https://developer.arm.com/documentation/105958/latest/, visited on August 20, 2023.