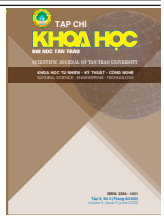
TẠP CHÍ KHOA HỌC ĐẠI HỌC TÂN TRÀO

ISSN: 2354 - 1431

http://tckh.daihoctantrao.edu.vn/



EFFICIENCY INTEGRATED EVALUATION OF ENCRYPTION ALGORITHMS IOT APPLICATIONS ON SPARTAN3E FPGAS

Tong Van Truong, Bui Trung Minh

Tan Trao University, Vietnam

Email address: tongtruong.dhtt@gmail.com

DOI: https://doi.org/10.51453/2354-1431/2023/978

Article info

Received:18/02/2023

Revised: 28/03/2023

Accepted: 20/5/2023

Keywords:

FPGA Spartan3E XC3S100E; encryption algorythms MARS, RC6, TWOFISH, RIJNDAEL.

Abstract:

In the vast ecosystem of IoT applications, selecting the appropriate data encryption solutions and fulfilling the requirements when calculating the cost of hardware and the integration efficiency of encryption algorithms used on FPGA devices is a topic of significant concern. The experimental FPGA device has been selected by the study team.MARS, RC6, TWOFISH, and RIJNDAEL are a few examples of encryption algorithms. We executed the program using the Spartan3E_ XC3S100E chip device family and family along with the Xillinx simulation software version 13.4 based on the design modules of each function block for each algorithm in the VHDL language. The resource costs of each design on the FPGA were provided after the program had run, and these data were compiled for analysis. In the context of this study, we advise system designers to incorporate the RIJNDEAL algorithm into IoT systems in order to achieve data encryption.

TẠP CHÍ KHOA HỌC ĐẠI HỌC TÂN TRÀO
ISSN: 2354 - 1431
http://tckh.daihoctantrao.edu.vn/

ĐÁNH GIÁ HIỆU QUẢ TÍCH HỢP CỦA MỘT SỐ THUẬT TOÁN MÃ HÓA ỨNG DỤNG TRONG IOT TRÊN CHIP FPGA DÒNG SPARTAN3E

Tống Văn Trường, Bùi Trung Minh
Trường Đại học Tân Trào, Việt Nam
Địa chỉ Email: tongtruong.dhtt@gmail.com
DOI: https://doi.org/10.51453/2354-1431/2023/978

Thông tin bài viết

Ngày nhận bài: 18/02/2023

Ngày sửa bài: 28/03/2023

Ngày duyệt đăng: 20/5/2023

Từ khóa:

FPGA Spartan3E XC3S100E; Thuật toán mã hóa MARS, RC6, TWOFISH, RIJNDAEL.

Tóm tắt

Việc lựa chọn các giải pháp mã hóa dữ liệu phù hợp và đáp ứng yêu cầu khi tính toán được chi phí về phần cứng và hiệu quả tích hợp của thuật toán mã hóa sử dụng trên các thiết bị FPGA trong môi trường rộng rãi của các ứng dụng IoT đang là vấn đề rất được quan tâm. Nhóm nghiên cứu đã lựa chọn thiết bị thực nghiệm là FPGA. Các thuật toán mã hóa được lựa chọn gồm: MARS, RC6, TWOFISH và RIJNDAEL. Trên cơ sở các modul thiết kế của từng khối chức năng đối với từng thuật toán trên ngôn ngữ VHDL, chúng tôi đã tiến hành chạy chương trình trên phần mềm mô phỏng Xillinx phiên bản 13.4 với dòng và họ thiết bị chip Spartan3E_ XC3S100E. Sau khi chạy chương trình, các chi phí về tài nguyên của mỗi thiết kế trên FPGA được báo cáo cụ thể, những dữ liệu này đã được tổng hợp để đánh giá. Trong khuôn khổ nghiên cứu này thì thuật toán RIJNDEAL được chúng tôi khuyến cáo các nhà phát triển hệ thống sử dụng trong các hệ thống IoT với mục tiêu mã hóa dữ liệu.

1. Introduction

The internet of things (iot) is the apex of technological advancement at the moment. Because of its originality, application, and ease in a technology environment where the user goal is everything, it is becoming more and more common and intriguing to many scientists [1]. With the capacity to extend beyond space and time, penetrate to touch every nook and cranny, and reach every aspect of life as it is today, it is also the focal point of all attackers and access attackers or stealing data, hijacking attacks... Are issues that still have a lot of room for research, and the field of data security for iot systems is not out of this vortex [6,7]. The resources required for calculation are minimal because IoT devices are designed for low energy consumption and operational costs. At the moment, symmetric encryption and asymmetric encryption are the two most often used encryption methods for data encryption [10]. Although the asymmetric encryption algorithm uses tens of times more energy than symmetric encryption, it is more faster and increases security. It is not appropriate for use in embedded or IoT devices with low-cost processors. So when it comes to encrypting and decrypting data for IoT devices, symmetric encryption techniques are the norm [2].

FPGAS (field programmable gate arrays) are a class of large-scale integrated circuits that use logic element array structures and have two distinguishing characteristics: they can be programmed and changed. Because of their flexibility during the design phase, fpgas help to cut costs and speed up manufacturing. Fpgas are also employed in real-time systems and are

applied to situations that need a lot of computation because to the high density of logic gates [3]. The effectiveness of encryption algorithms on FPGA can be compared and evaluated, but doing so in an absolute sense is challenging. However, this is required to illustrate the two most crucial factors that are frequently examined to assess the performance of any design, namely the cost of time and the cost of materials.

In the research framework of this paper, we seek to assist analysts, designers, and builders of IoT systems have more information on which data encryption algorithm (DEA) is appropriate and how to meet the requirements when estimating the cost of hardware and the integration efficiency of the encryption algorithm used on FPGA devices in the diverse environment of today's IoT applications. To do this, the article is organized as follows: After presenting the research content in the beginning section, Part 3 offers the research findings, contrasts the results, and makes recommendations for IoT system designers before coming to a conclusion.

2. Research content

2.1. Integrated performance evaluation approach for hardware devices

We apply the following method to assess the integrated effectiveness of cryptographic algorithms when implemented on FPGA[4]:

$$\mathbf{IE} = \frac{\mathbf{T}}{\mathbf{R}*\mathbf{F}}$$

IE: Integrated Efficiency

T: Throughput (Mb/s):

$$\mathrm{T} = \frac{\mathrm{Frequency}*\mathrm{bit}}{\mathrm{number\ of\ cycle}}\ (\mathrm{bit/s})$$

R: Resource costs are calculated using expenses based on one of the following criteria: number of slices; number of flip flops; number of Configurable Logic Block; number of LUT (Look-Up Table); number of IOB (Input/Output Block); number of Block Select RAMs (BRAMs). Normally **R** is calculated through the number of CLB

F: Frequency (MHz)

Compare all of the resources listed above that were created on an equivalent line of FPGA devices for the best comparison. However, we must also acknowledge empirically that, even when using the same code, implementing the same device family at various levels still influences the final information flow of the design. Therefore, there will be some differences if the same algorithm is built for 2 separate product lines by 2 different manufacturers. Only by comparing similar gadgets can two solid designs be compared to one another. Therefore, area and information flow both contribute to the evaluation of a design's efficacy. However, people also take into account other variables, such as Flow/Area, which is the ratio to take into account about relevant features of the design, when determining if a design is effective or not.

In order to ensure that the evaluation is the best recommendation for designers to select the most efficient data security solution for IoT systems, we employ all three characteristics in the experimental implementation of this study: flow, area, and flow/area of the designs for comparison.

2.2. Deployment approach

2.2.1. Linguistic use

The VHDL (Very High Speed Integrated Circuit Hardware Description Language) language is the chosen approach because the study's objective is to assess the cost of implementing the algorithm on hardware. It is a hardware description language for extremely high-speed integrated circuits created for the VHSIC (Very High Speed Intergrated Circuit) program of the US Department of Defense. The creation of VHDL was intended to create a standard hardware description language, allowing for the quicker experimental development of digital systems and their simple implementation. VHDL provides several benefits, including [8]:

Promotion: VHDL was created under US government guidance and is now an IEEE standard. There is no single person or organization that owns VHDL. As a result, a large number of device makers and distributors of system simulation design tools support VHDL. This is a key benefit of VHDL that contributes to its rising popularity.

Support for a variety of technologies and design techniques: VHDL supports a variety of design techniques, including top-down design and bottom-up design based on accessible libraries. As a result, VHDL can be used effectively for a wide range of design tasks, including the design of common components and Application Specific ICs.

VHDL is totally independent of the technology used in hardware production. Depending on the hardware

manufacturing process (using CMOS, nMOS, or GaAs), a VHDL system description created at the gate level can be translated into various circuit assemblies. The ability to ignore hardware technology when creating the system is another significant benefit of VHDL. in order for newly developed hardware manufacturing technologies to be quickly applied to the systems that have been developed.

Extended description capabilities: VHDL enables the definition of hardware behavior at all levels, from the digital system level (black box) up to the gate level. Only a cohesive syntax that is consistent for all levels will allow VHDL to describe system behavior on numerous levels. As a result, we are able to simulate a design that has both highly and thoroughly documented subsystems.

Interchangeability: A VHDL model can be run on any simulator that complies with the VHDL standard, and the results defining the system can be shared amongst designers as long as they adhere to the same VHDL standards. Additionally, even while individual subsystems within a system are designed individually, a design team can exchange high-level descriptions of those subsystems.

Support for high-level designs and reuseability of designs: VHDL was created as a high-level programming language, making it possible to design a sizable system involving many individuals. There are numerous capabilities in the VHDL language that allow design management, testing, and sharing.

VHDL also permits the reuse of current components. The study team suggests VHDL as the simulation execution language for this experiment for the reasons listed above.

2.2.2. Experimental tools and chip selection

As was mentioned before, the experimental design process must be carried out and assessed numerous times to determine the best alternative. This is necessary to construct digital systems more quickly, more effectively, and with ease. Because of the device's user-programmable and customizable features, the research team decided to use it as an FPGA (Field-programmable gate array).

The main HDL hardware description languages used for designing or programming FPGAs are VHDL, Verilog, and AHDL [9]. Large FPGA manufacturers like Xilinx and Altera frequently offer software packages and auxiliary tools for the design process. A number of other companies also offer these software packages, including Synopsys and Synplify. With the design code from the HDL as input, these software tools are able to complete every step of the typical IC design process. However, using the analyses mentioned above, the research team decided to test VHDL on FPGA devices.

The research team chose the FPGA for its evaluation method after analyzing its strengths and qualities in the manner described above. However, given the diversity of FPGA families and product lines, we chose the Spartan3E_ XC3S100E family and chip family for our study since we felt they were better suited to the initial objective of IoT applications. Here are the key justifications behind this decision.

A kit called Spartan3E (Figure 1) was created for use in applications that require high performance and bandwidth. Depending on the objectives of the applications' creators, it is utilized in a wide range of market areas. With the SMA interface, the Spartan3E FPGA connector enables design with both conventional and proprietary serial standards.

The Spartan-3E also has some important advantages over the Spartan 3 family, including ease of use, low cost, low power consumption, density of integration of many logic elements, system clock speeds from 5-300 Mhz, five different power consumption levels (3.3V, 2.5V, 1.8V, 1.5V, and 1.2V), integration of up to 376 I/O pins or 156 different signal pairs, and a high data transfer rate.

Figure 1*: Spartan-3E Starter Kit Board*

Spartan3E is composed (Figure 2) from the following parts:

Input/Output Blook (Ios):

Configurable Logic Blocks (CLBs): Made up of look-Up Tables (LUTs).

Block RAM: Supports 16 Kb RAM per RAM Block, the number of RAM Blocks depends on each chip.

Digital Clock Manager(DCM) Blocks: Interconnect

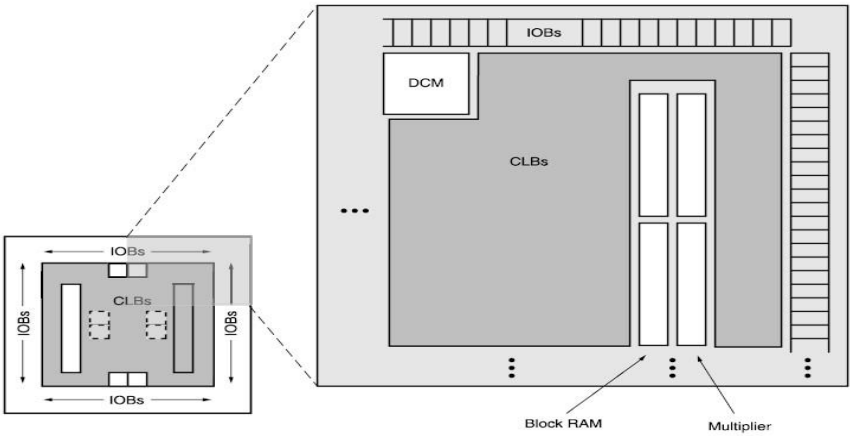


Figure 2: Structure diagram of the chip Spartan3E_ XC3S100E

The chip specifications are described in Table 1: each CLB consists of 4 slices.

Table 1: Specifications of Spartan3E series devices.

Device	System Gates	Equivalent Logic Cell	CLB Array (One CLB = Four Slices)				Distributed RAM bits	Block RAM bits	Dedicated Multiplier	DCMs	Maximum User I/O	Maximum Differential I/O Pairs
			Rows	Columns	Total CLBs	Total Slices						
XC3S100E	100K	2.160	22	16	240	960	15K	72K	4	2	108	40
XC3S250E	250K	5.508	34	26	612	2.448	38K	216K	12	4	172	68
XC3S500E	500K	10.476	46	34	1.164	4.656	73K	360K	20	4	232	92
XC3S1200E	1200K	19.512	60	46	2.168	8.672	136K	504K	28	8	304	124
XC3S1600E	1600K	33.192	76	58	3.688	14.752	231K	648K	36	8	376	156

2.3. Research subjects

Currently, there are many open and closed algorithms used for data encryption in applications in general and IoT in particular. However, within the scope of this study, the research team selected 4 algorithms that are widely used today to perform the evaluation. The selected algorithms include: MARS encryption algorithm, RC6 encryption algorithm, TWOFISH encryption algorithm, RIJNDAEL encryption algorithm. These are the algorithms that have been selected in the final round of the AES (Advanced Encryption Standard) candidates.

2.4. Encryption algorithm MARS

Using variable-size key ciphers and supporting 128-bit data block sizes, MARS is a symmetric key encryption technique. In order to improve the method's efficiency in comparison to earlier traditional encryption algorithms, the algorithm is built on the premise of utilizing the advantages of conducting operations on current generations of computers. There are three steps in the general structure of encryption: Forward mixing, Cryptographic core ,and Backward mixing. At the center is the primary encryption, which uses keyed transformations.

2.3.2. Encryption algorithm RC6

Round shifting is the foundation of RC6, and the data determines how many rounds to shift. The algorithm increases the diffusion after each round, increases security, decreases the number of rounds, and improves algorithm performance by using four registers and integer multiplication. Because RC6 is intended to be straightforward, attacks on it primarily target the security of data-dependent circular shifting. Numerous investigations on its construction and the calculations impacting its safety have been conducted since its proposal.

Although there hasn't been a practical attack on RC6 yet, the research findings have produced some important theoretical analyses and attacks.

There are several additional forms of attacks, however it mostly concentrates on using differential and linear cryptanalysis in RC6 assault.

2.3.3. Encryption algorithm TWOFISH

TWOFISH is a popular symmetric key block cipher that can be utilized in both software and hardware settings. This encryption technique is perfect for applications demanding high security because it is tailored for 32-bit central processing units. It has a variable-length key that can be 128, 192, or 256 bits long. It is a 128-bit block cipher. TWOFISH is an unpatented, open-source encryption algorithm that is freely usable. Block ciphers, rapid encryption and decoding, symmetric key encryption, variable length keys, and open source are some of its features.

2.3.4. Encryption algorithm RIJNDAEL

A symmetric key technique is employed by the widely used encryption standard RIJNDAEL to safely encrypt and decrypt data. It is employed to safeguard private data, including credit card details, passwords, and other sensitive information. It is a symmetric block cipher that uses keys that can be 128, 192, or 256 bits long and has a block/block size of 128 bits. Secure communications, file encryption, and data storage are just a few of the areas where RIJNDAEL encryption is frequently utilized.

2.4. Strategy and design of cryptographic algorithms on the Spartan3E chip

There are typically two options for implementing encryption methods on hardware devices: IL (Iterative Looping) and PP (Pipeline) [5]. However, within the parameters of this study, the team suggests implementing IL mode testing in order to fit the applications of IoT and the features of these systems. In the follow-up study, studies and assessments based on alternative methodologies will be used.

2.4.1. Algorithms created by FPGA program modules

In this section, we only show a few example design results due to the length of the study. As explained in article section 6, aggregate data will be used to compare the test results.

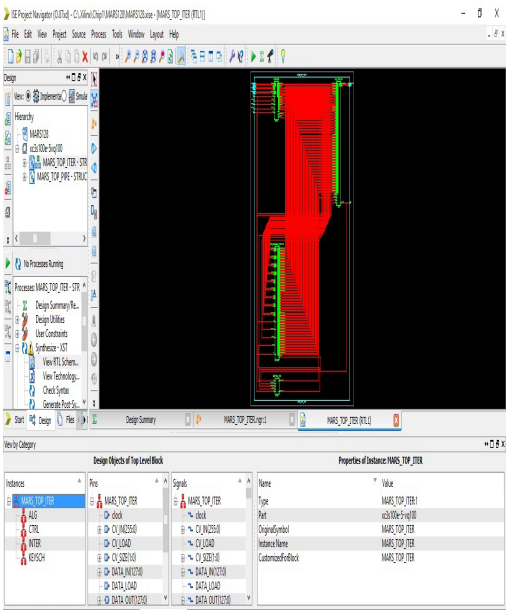
The next 7 files describe the design of the MARS encryption algorithm using VHDL on the FPGA chosen in accordance with the IL strategy. Each file has a corresponding function that is detailed in full in Table 2. The remaining algorithms share comparable program modules, but each algorithm also contains unique modules since it uses a separate encoding/ decoding scheme. For instance, the RC6 encryption technique will be designed on the FPGA using 12 modules, each of which will represent one of the algorithm's 12 functions for encoding and decoding; The RIJNDAEL encryption algorithm contains 19 modules, RC6 only has 12, and TWOFISH only has 11 modules, all of which correlate to the corresponding number of functions in each technique's encryption/ decryption scheme.

Table 2:Illustration of design modules of Encryption algorithm MARS

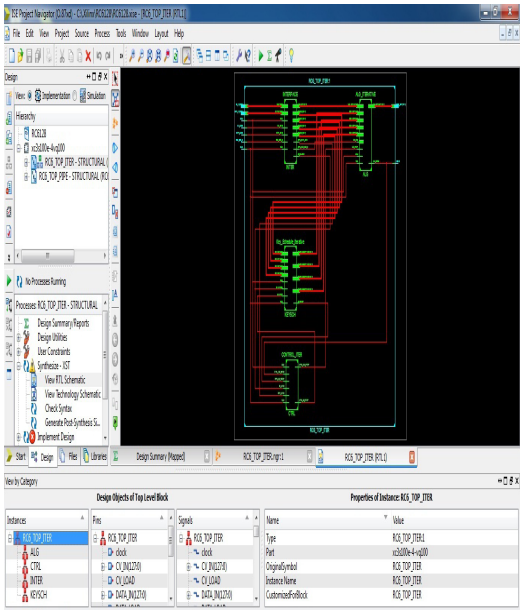
No.	Filename	Function
1	alg_iterative.vhdl	Describe the algorithm's mode of operation.
2	Mars_pack.vhdl	defines the base cipher's control components, algorithmic transformations, and transformation function.
3	Mars_top_Iter.vhdl	Function block descriptions for signals
4	controller_iter.vhdl	a description of the program's control block
5	interface.vhdl	Description of data transfer, control signal, input signal, and output signal
6	Reg128b.vhdl	explains how data is converted between two registers.
7	key_schedule_iter.vhd	Describe the algorithm's key scheme.

2.4.2. Algorithms' circuit design on an FPGA

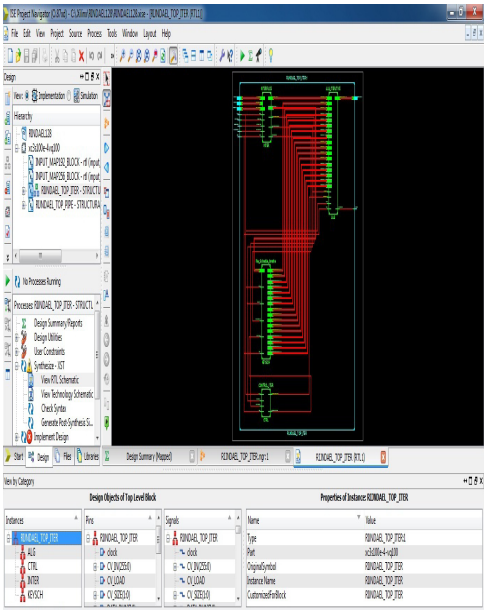
We ran the program on the Xillinx simulation software version 13.4 with the line and family of chip devices introduced above and obtained the overall design diagram of each algorithm on the basis of design modules of each function block for each algorithm (described in 2.4.1) written in VHDL [11]. Figure 3 shows the design outcomes of the algorithms on the FPGA in accordance with the IL method.



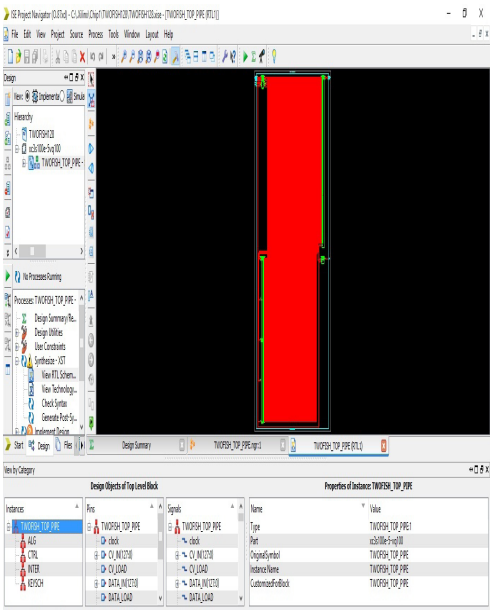
Algorithm Mars



Algorithm RC6



Algorithm RIJNDAEL



Algorithm TWOFISH

Figure 3: Diagram of a circuit using the Spartan3E chip line of algorithms

2.4.3. Findings regarding the cost of programming algorithms for FPGA

The resource expenses of each design on the FPGA are reported in detail in the Design Overview after the program has run (see Figure 4). For instance, the run screen of Spartan3E reports the costs associated with implementing the RIJNDAEL algorithm using the figures in Figure 4. We have additionally implemented and gathered for evaluation the data in part 3 using various algorithms.

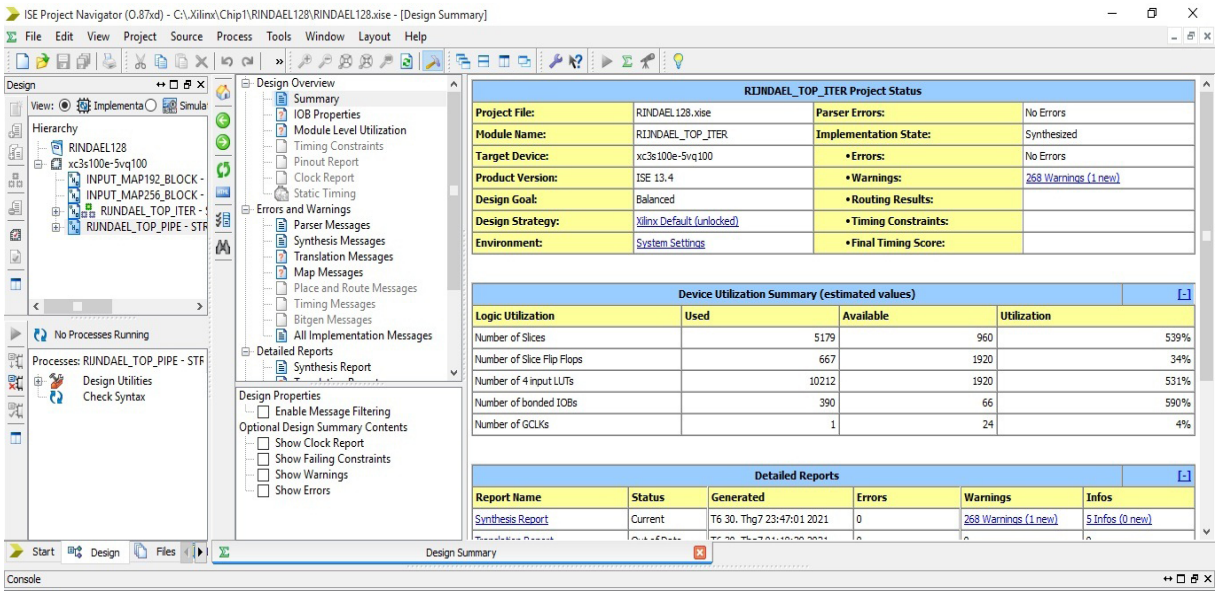


Figure 4: Resource cost parameter of RIJNDAEL algorithm on Spartan3E

3. Evaluation findings and analysis

Table 3 provides a full summary of the algorithm implementation outcomes on Spartan3E. In addition to resource costs with data selection as Slice, we also aggregate additional data in this table to help with the calculation of performance evaluation parameters. This data includes frequency (data obtained from running the program), the number of rounds of the algorithm (as suggested by the author publishing the algorithm), the data block size of each algorithm to determine the throughput, which was shown in section 2, and additional integrated performance evaluation data.

Table 3: Summary of the results of algorithm execution on Spartan3E

Spartan3E_ XC3S100E						
Algorithm	**Block**	**Resource cost (Slice)**	**Frequency (MHz)**	**Number of rounds**	**Throughput (Mpbs)**	**Effect**
MARS	128	21.040	9.579	32	38.316	0,00182
RC6	128	7.816	35.872	20	229.581	0,02937
RIJNDEAL	128	5.846	83.520	10	1.069.056	0,18287
TWOFISH	128	27.137	37.707	16	301.656	0,01112

Graphs have been developed to illustrate these results and make comparisons easier. The comparison chart of the resource cost, frequency, throughput, and integrated efficiency of 4 algorithms on FPGA is shown in Figures 5, 6, 7, and 8, respectively.

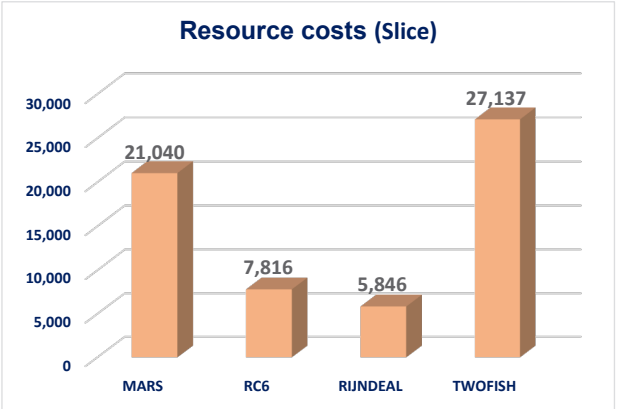


Figure: Compare the resource costs of 4 algorithms on Spartan3E

The benefits are evaluated in the following order to evaluate the cost of resources (the smallest has the advantage), as shown in the graphs: (first). RC6, MARS, RIJNDEAL, RC6, and (4). TWOFISH. Therefore, the system builder can select RIJNDEAL when selecting encryption algorithms for applications that do not care about code/decryption performance.

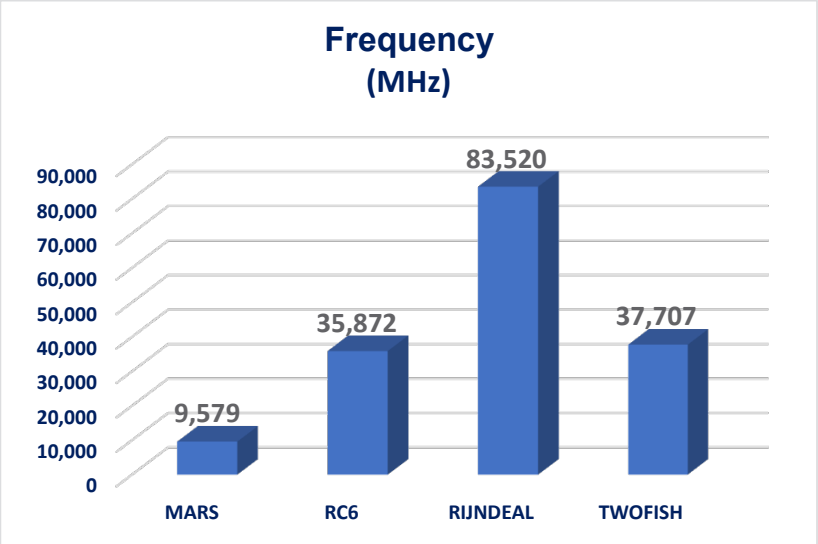


Figure 6: Compare frequencies of 4 algorithms on Spartan 3E

However, the ranking order has a reversal factor that places TWOFISH in second place, RC6 in third place, and MARS in last place in order to analyze the frequency (greater has more advantages). The specific ranking order according to benefits is as follows: (first). RC6; (2) RIJNDEAL; (3) TWOFISH; (4) MARS.

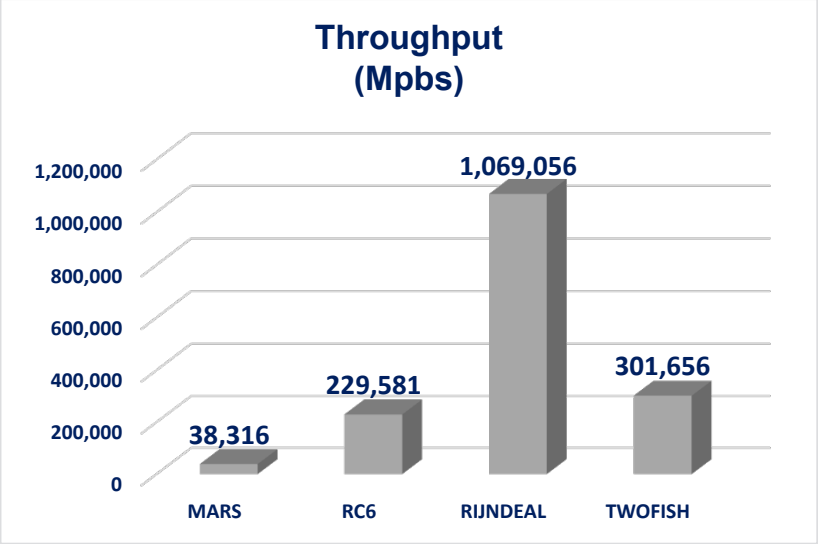


Figure 7: Compare throughput of 4 algorithms on Spartan3E

However, for IoT systems where the throughput factor is a higher coefficient, the aforementioned two comparison criteria are not a deciding factor for selecting a data encryption algorithm. The encoding/decoding traffic element may or may not be given greater weight depending on the nature of each application. With this parameter, the rating order has changed, although RIJNDEAL still holds the top spot. In more detail, the ranking is as follows: RIJNDEAL, TWOFISH, RC6, and MARS are listed in that order.

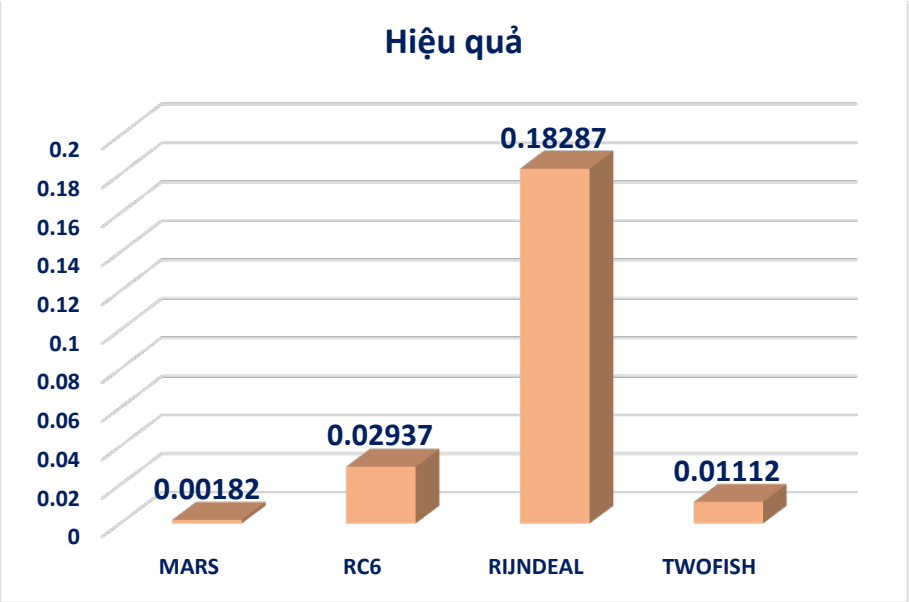


Figure 8: Compare the performance of 4 algorithms on Spartan 3E

Each comparison needs to be assessed against a wide range of evaluation criteria, both in engineering and in practice. Each of the aforementioned assessment criteria has revealed that each algorithm's rating has altered. However, it is determined that the next comparison criterion is a more "fair" comparison criterion than integrated efficiency. Positions 2, 3, and 4 always vary in the order based on this criterion, whereas position 1 always stays the same. The specific order is as follows: (1) RIJNDEAL, (2) RC6, (3) TWOFISH, and (4) MARS.

In conclusion, we also witness distinct genuine photos with 4 different viewpoints on hardware costs, integration efficiency, etc. to compare the order of data encryption techniques. We advise system developers to adopt RIJNDEAL in IoT systems with the aim of encrypting data because it consistently ranks first within the context of our research.

4. Conclusion

The above four block ciphers are perfectly suited for Internet of Things applications because of their high integration efficiency, low cost of area, which ensures compactness, and low power consumption due to high clocks. This is demonstrated by the results of research and implementation of block ciphers on FPGAs with the Spartan3E line code XC3S100E. In addition to the analysis mentioned above and depending on the design model on the FPGA, users must also take into account the choice of security level and resistance to assaults when selecting an algorithm and device family for a certain application. In upcoming papers, we'll have more open follow-up investigations on this subject.

References

1 A. de Saint-Exupery, *"Internet of things, strategic research roadmap,"* European Commision, 2013.

2 A. Riahi, Y. Challal, E. Natalizio, Z. Chtourou, and A. Bouabdallah, *"A systemic approach for iot security,"* in Distributed Computing in Sensor Systems (DCOSS), 2013 IEEE International Conference on. IEEE, 2013, pp. 351–355.

3 C. Chitu and M. Glesner, "An FPGA Implementation of the AES-Rijndaelin OCB/ECB modes of operation", *Microelectronics Journal*, vol. 36, pp. 139-146, 2005.

4 T. B. Do and T. M. Duong, "Assessment of the integrated efficiency of block cipher algorithms for wireless networks on a FPGA chip" *TNU Journal of Science and Technology*, 226(11): 357 – 364, 2021

5 H. M. Nguyen and T. B. Do, "Hybrid Model in the Block Cipher Applications for High-Speed

Communications Networks," *International Journal of Computer Networks & Communications (IJCNC)*, vol. 12, no. 4, July 2020, doi: 10.5121/ijcnc.2020.12404 55.

6 M. Usman, I. Ahmed, M. I. Aslam, S. Khan, and U. A. Shah, "SIT, A Lightweight Encryption Algorithm for Secure Internet of Things," *(IJACSA) International Journal of Advanced Computer Science and Applications*, vol. 8, no. 1, pp. 402-411, 2017.

7 Mirza Abdur Razzaq, Sajid Habib Gill, Muhammad Ali Qureshi, Saleem Ullah. (2017). Security Issues in the Internet of Things (IoT): A Comprehensive Study. International Journal of Advanced Computer Science and Applications, 8(6). DOI: http://dx.doi.org/10.14569/IJACSA.2017.080650.

8 Nguyen Quoc Tuan, (2013), VHDL language for IC design, Ho Chi Minh City National University Publishing House.

9 T. T. Nguyen and N. V. Pham, "Design pulse generator, frequency divider based on FPGA and VHDL" *TNU Journal of Science and Technology*, 226(11): 252 - 259, 2021

10 R. Vignesh, A.Samydurai. (2017). Security on Internet of things (IoT) with Challenges and Countermeasures. *International journal of Engineering development and research,* 5(1), 417-423.

11 Xilinx Inc, "Development System Reference Guide", [E-book], 2022. [Online]. Available: www.xilinx.com [Accessed September 10, 2022].