Published Online First: November, 2024
https://doi.org/10.21123/bsj.2024.8931
P-ISSN: 2078-8665 - E-ISSN: 2411-7986

Baghdad Science Journal

Design Hybrid Architecture to Implement AES Algorithm on FPGA for IoT Applications

Nada Qasim Mohammed [1], ***Amiza Amir*** [1], ***Muataz Hammed Salih*** [2], ***R. Badlishah Ahmad*** [1], ***Rima Matem*** [1], ***Nisrean Thalji*** [3], ***Qasim Mohammed*** [4], ***Jamal Kamil K. Abbas*** *[5], ***Taha Raad Al-Shakhli*** [5]

[1] Faculty of Electronic Engineering Technology, University Malaysia Perlis, 02600 Arau, Malaysia.
[2] Automation Group, Design and Engineering, Flex, Penang, 11500, Malaysia.
[3] Department of Robotics and Artificial Intelligence, Jadara University, Irbid, Jordan.
[4] Department of Cybersecurity Science, Al-Kunooz University College, Basrah, Iraq.
[5] Cybersecurity Engineering Techniques, Al-Nisour University College, Baghdad, 10036, Iraq.

Received 13/04/2023, Revised 28/05/2024, Accepted 30/05/2024, Published Online First 20/11/2024

Abstract

High-security cryptography algorithms like AES require high computational capabilities to achieve information security. Therefore, it is necessary to use parallel computing architectures that exploit modern technologies in spatial parallelisms to obtain the most conceivable computational power. Various technologies have been introduced to achieve parallel processing. One of them is field-programmable gate arrays (FPGAs), which have good characteristics suitable for implementing parallel architectures with lower power consumption. The paper aims to design and implement an embedded computing processing engine architecture transceiver with high performance to obtain better throughput on FPGA technology to encrypt and decrypt images. In this design, two boards are used, "DE1_Soc and NEEK board" with Altera Quartus prime 18.1, cyclone V 5CSEMA5F31C6 FPGA device for synthesis and simulation. The implementation results show that the proposed architecture has an efficient performance in terms of an operating frequency is 600 MHZ and a throughput is 76.8 GHZ.

Keywords: AES algorithm, Encryption /Decryption, Intent of thing, FPGA, VHDL, Security.

Introduction

Different approaches of cryptographic algorithms are used to secure and protect sensitive information from unauthorized parties; each one has its pros and cons. The Advanced Encryption Standard (AES) is one of the cryptographic algorithms, which is used to secure data, that is trustworthy, efficient and has not yet been cracked. Different approaches are used to implement it with different techniques; each has pros and cons[1,2]. AES is a symmetric-key technique, meaning only duplicate keys for encrypting and decrypting processes are used. Three AES versions are adapted from the United States National Institute of Standards and Technology (NIST); the block size is 128 bits, but they differ in key length sizes: 128, 192, or 256 bits, as well as the number of rounds that depends on the used key length. The AES system is referred to as the Rijndael algorithm, which was developed by Joan Daemen and Vincent Rijmen[3]. Although the AES algorithm provides good security, it needs high computing power to perform its operations within several used rounds, which needs more time for the encryption and decryption of the data [4]. So, using the AES encryption system takes a long to perform its operations, making it unsuitable for devices with limited resources. Whereas IoT processing devices have limited resources and are not able to perform operations in a fast manner in real-time[5, 6]. The growth of image data gathering and processing needs to explore modern technologies that speed up processing, achieve high throughput, and lower power consumption in real time. So, it is necessary to use suitable approaches to speed up its operations to meet these requirements using the AES algorithm by implementing it on FPGA, which is one of the solutions to increase processing speed. The reconfigurable FPGAs make them a very suitable choice for real time processing to get fast computation since FPGAs have a parallel nature that can be used in both spatial and temporal parallel processing [7, 8].

FPGA technology is preferred over other parallel computing options for implementing the AES algorithm in IoT applications due to its high performance, flexibility, energy efficiency, and reconfigurability:

- Superior implementation: FPGAs provide comparable processing power and hardware accelerators for successful implementation. Unlike software implementation, FPGA-based AES schemes provide much higher throughput and shorter idle times. By parallelizing the AES computation process across different processing units, FPGAs can accelerate data encryption and decryption, making them ideal for continuous IoT applications.
- Flexibility: As an extremely flexible solution (in terms of hardware customization) one of the advantages of using FPGAs is that innovators can customize AES implementations according to specific requirements. In contrast to fixed capability ASICs or processors with a fixed range of functions, FPGAs are reconfigurable for use with different AES key sizes, operating modes and performance extensions, yielding them for use with the expanding security needs of IoT applications.
- Power efficiency: Integrated circuits are less power efficient than FPGAs delivering more performance per watt. It is critical for IoT devices; that is, devices that rely on power sources with limited power in, such as batteries; by pushing the AES tasks to FPGAs, the IoT devices can extend battery life and conserve power.
- Reconfigurability: FPGAs are reconfigurable. The hardware can be reconfigured, incrementally, to support different functions, including AES computation. Updating or patching AES implementation is very easy, as is addressing new security risks and incorporating new encryption standards. FPGAs are flexible and make for a future proof path to support changing security requirements of IoT applications.

Because of the high performance, flexibility, power efficiency and reconfigurability offered by FPGA technology, AES algorithms are the preferred implementation option for IoT applications. FPGAs' parallel processing capabilities accelerate data encryption and decryption and make them applicable to real time IoT applications. In addition, FPGAs offer hardware customizability for AES application, so that AES implementation is flexible with various requirements of the application. As they also increase both power efficiency and reconfigurability, they become more suited for resource constrained IoT devices. Thus, one of the main advantages of FPGA technology is that it may exert a significant influence on the security of IoT applications by expeditiously realization of AES algorithms[9].

As with any other technology, however, this implementation suffers certain drawbacks. The objective of this paper is to explore the drawbacks of using AES algorithm in FPGAs and propose a recovery procedure to address these limitations:

- High operational and regulatory costs: To implement FPGAs, one must have a special knowledge; they are expensive to implement and therefore relatively expensive for a small project on the Internet of Things.
- Limited flexibility: FPGAs provide the scalability to critical engineering and infrastructure where replacement or change of equipment may be required. It may make flexibility and resiliency difficult in situations where the number of connected devices goes up and down.

- Power consumption: FPGAs have their isochronous nature, which consumes a lot more power than other computing platforms which sometimes leads to higher operating costs and might not be suitable for power hungry IoT devices.
- Programming complexity: FPGAs are hard to create and hard to program them with special skills needed. Long development cycles and the risk of implementation errors characterize high learning and adaptation required projects.

To address the aforementioned disadvantages, the following recovery procedure is proposed:

- Simplify costs: By using cloud-based FPGA services or open-source FPGA platforms, overhead can be reduced significantly, and a smart solution for performing AES computation on FPGA is offered.
- Privacy and flexibility: The flexibility gained in scaling AES computations on FPGAs is also demonstrated. Separating encryption and decryption into two separate operations will allow for the framework to scale to more numbers of IoT devices without the need for any new hardware.
- Power optimization techniques: Power saving techniques include clock gating, voltage scaling and dynamic power management, all for power reduction with performance penalty.
- Simplified development environment: It reduces FPGA programming to libraries and frameworks that provide easy-to-use development tools. This makes programming on FPGA based projects relatively easy for any range of developers.

Therefore, the architecture techniques is updated to make the processing faster than before without changing the techniques themselves [5,6]. The proposed architectures had been exploit the capabilities which is offered by FPGI technologies in relation to the general purpose configurable logic, as well as specific functions specialization to design an architecture is used to implement an AES algorithm to make the IoT environment data more secure. DE1_Soc and NEEK boards are used for this purpose. Actually, the following contributions are achieved:

- Proposes design of end-to-end architecture suitable for implementing AES algorithms by processing the data in parallel. AES needs high computing power to match the limited capabilities of the Internet of Things. Also, enhance the AES cryptosystem security by dividing and shifting processes, encrypting, sending, decrypting, and collecting files as original with minimum data corruption.
- Design and implement an architecture with quad processing engines, which use parallelism with spatial and temporal to increase the throughput and reduce the processing time. Spatial parallelism is achieved by creating quad processing engines to handle different data at the same time. While the temporal (deep pipelining) divides every engine into sub-modules to process and execute various tasks.
- The proposed design used a frequency of 600 MHz with 128.bit for each engine, an FPGA DE1-SoC board and a NEEK board have been chosen as the computing platforms because they have a high-performance power processing data chipset. It used design tasks because of its capability to handle large amounts of data in parallel[10-13].

The paper is organized to discuss the following related work, components of IoT, IoT data security challenges, AES 128-bit encryption/decryption description, computing processing engine architecture model design on FPGA, results and discussion and conclusion.

Related work

AES, also known as Rijndael, is a symmetric encryption algorithm that operates on fixed-size blocks of data. AES uses a series of substitution, reordering, and permutation operations to provide a high degree of security. Field-programmable gate arrays (FPGAs) are flexible, resilient, and parallel, making them suitable for implementing complex algorithms such as AES; FPGA-based implementations are ideal for real-time applications as they can achieve high operating frequencies and throughput. Researchers have proposed various architectural designs for implementing AES in FPGAs. These designs focus on improving resource utilization and reducing power consumption while maintaining high performance. The proposed FPGA architecture for AES has been extensively evaluated and the results are impressive: the operating frequency of 600MHz indicates the speed at which encryption and decryption operations can be performed; and the high throughput of 76.8Gbps reflects the many advantages of FPGA-based AES applications, such as:

- Flexibility: FPGA designs can be easily modified and reconfigured, allowing for the incorporation of additional features or improvements.
- Parallelism: FPGA designs can exploit parallelism, enabling multiple encryption or decryption operations to be performed concurrently, thereby enhancing overall performance.
- Security: The use of FPGAs provides a higher level of security as the encryption keys and data can be stored within the FPGA, reducing the risk of exposure.

Many architectural designs for implemented AES on FPGA families were used to make use of many advantages, such as increased operating frequency. These approaches aim to increase the processing speed for Advance encryption standard algorithms implementation to secure applications sensitive information. The following are some researchers building the AES with a different approach to FPGA.

In (Ref 1), the Implementation of the AES algorithm on FPGA uses a multistage pipeline with resource sharing to achieve the optimized area, power and delay. Used Device: Virtex 7. But the dis advantage that it still with low frequency is 190 MHZ, So the throughput is also still less.

In (Ref 2), Implement a 128.bit AES Encryption algorithm by improved Sub-Pipelined S-box is integrated into the design to balance the area, power, and performance of the system with less device utilization. Used the Device: Virtex 4. The dis advantage for this researcher used the sub pipeline with limited resources, that will produce low efficiency

In (Ref 3), Implemented an architecture-based parallel hardware implementation of a data hiding scheme for quality access control of images in the discrete cosine transform (DCT) domain on FPGA the disadvantage for the researcher is still low because no parallel processing to speed up the process.

In today's digital era, the need for secure data transmission and storage has become increasingly vital. Image encryption and decryption is a crucial aspect of ensuring data confidentiality, integrity, and authenticity. Field-Programmable Gate Arrays (FPGAs) have gained significant popularity in implementing encryption algorithms due to their parallel processing capabilities and high-performance computing. This paper provides a detailed review of the existing literature on FPGA-based image encryption and decryption, which could offer valuable context and help identify gaps in the research[11-13].

Components of IoT

The Internet of things components are sensors and actuators, embedded processing, and the cloud and connectivity[14-17]. Connectivity and the cloud serve as a medium for communication and data storage, while embedded processing provides smart object intelligence. The components that make up the Internet of Things system include[18]:

- Hardware: In IoT systems, the hardware utilizes devices for control, servers, sensors, actuators, devices for a remote dashboard, a routing device, embedded communication devices, and others. The IoT hardware is used to carry out several activities, including system activation, action specification, communication, and detection, to support specific objectives and security. Developments in wireless transmitting devices with low power and low cost are promising in IoT.
- Middleware: It contains software on-demand and computing resources for data analysis. The software is responsible for data collection and device integration within the networks of the IoT.
- Presentation: It helps in visualizing, providing insight, and comprehension of data by providing tools that can be used for various applications by utilizing different platforms.

Meanwhile, some enabling technologies are required for IoT systems under the above components, such as:

- Radiofrequency identification (RFID) can locate and determine items using radio waves that permit the collection of necessary information.
- Wireless Sensor Networks (WSN) enable the capturing of data by sensor networks from different environments and processes, analyzing and easily disseminating information by utilizing small, tiny devices that are low-powered and coupled with wireless communications.
- Addressing Scheme: To identify the devices uniquely and control the devices remotely through the Internet, each object that may be connected subsequently will undergo unique identification.
- Data storage and analytics: The IoT needs a tool to collect massive amounts of data and manage, process, and store them in real-time. Therefore, artificially intelligent algorithms, methods, and software applications suitable for decision-making are needed to perform these actions.
- Visualization: according to the availability of smartphones, tablets, and iPad devices, the data will be visualized, and the user can read and understand this data easily with minimum effort.

IoT Data Security Challenges

Expanding the number of devices that communicate via networks in the IoT and widely used in human life faces increasing security risks and poses new challenges to data security. These security risks need to consider some security requirements like authentication, confidentiality, trust, and data security. Therefore, there is a need for data transmitted and stored in a secure manner to protect IoT data from destructive forces or unauthorized access, which could pose a serious threat to safety of human life[19-21]. Therefore, there is a need to endorse a highly secure cryptographic system that is able to withstand intruder attacks. One of these cryptographic that not been cracked until now is AES algorithm.

AES 128-bit Encryption/Decryption Description

Both state and keys have 128 bits. Every round here is made up of four transformation functions (Sub bytes, Shift-rows, Mix-columns, add round Key) to accomplish the encryption operation, except the last round, which does not include the AddRoundKey function. The decryption operation involves conducting an inverse state to achieve the original state. The plaintext is an XORed operation with keys. All the first "9" AES algorithm rounds comprise the four stages, excluding the last round in which only three operations are involved (Sub-bytes, Shift-rows, add round key) [22-24], AES Encryption is illustrated in Fig. 1.

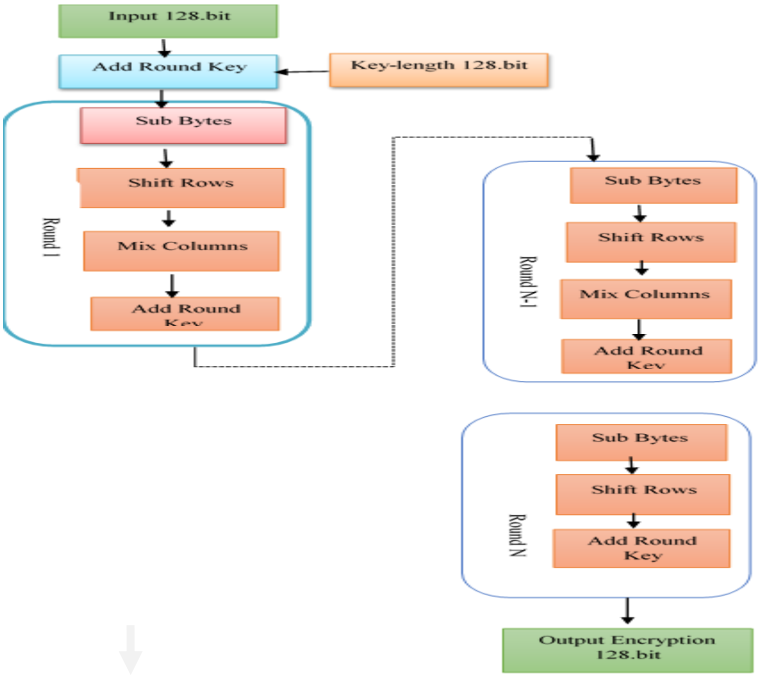


Figure 1. The Block Diagram of 128-AES Algorithm Encrypt

A. Sub Bytes Stage

The first stage uses a 16×16 matrix of bytes, called S-boxes, to substitute each state byte with a different value according to the mapping between the S-Box and the element's value. The initial four bits of the state byte stand for the number of the S-Box row, while the second four bits represent the number S-Box column. The Sub Bytes function is shown in Fig. 2.

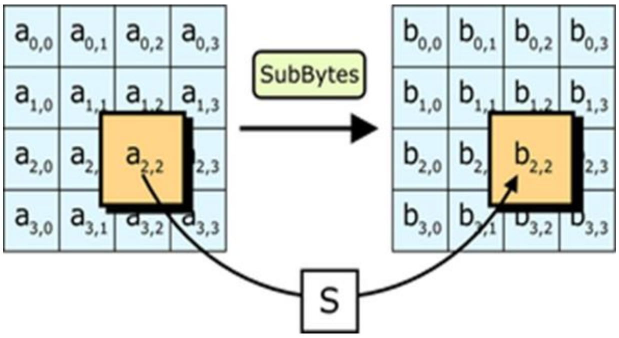


Figure 2. The Sub Byte Operation

The S–box design utilizes the multiplicative inverse over GF(28) to obtain the desired non-linearity properties to achieve resistance to cryptanalytic attacks and interact the input bits with the output bits.

B. Shift Rows Stage

This function performs a simple permutation; each row is shifted depending on its location, and the first row is never shifted. The second row involves the conduction of a 1-byte circular left shift. While the 2-byte and 3-byte left circular left shifts are carried out in the 3rd and 4th rows accordingly. The Shift Rows Function is shown in Fig. 3.

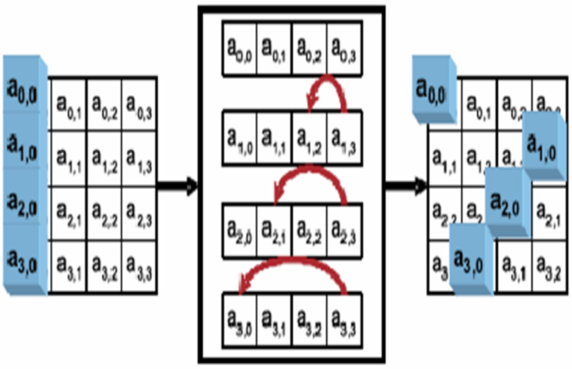


Figure 3. The Operation of the Shift Rows Functions

C. Mix Columns Stage

This function operates on each column individually by carrying out a form of substitution in which arithmetic over GF (28) is involved. Here, a modulo multiplication is of 211four numbers in a column in Rijndeal's Galois field using a particular matrix described in Fig. 4. The function's input is four bytes of a single column, and the output is a novel matrix of 16 bytes in place of the initial column. In the final round of the algorithm, this process is not involved.

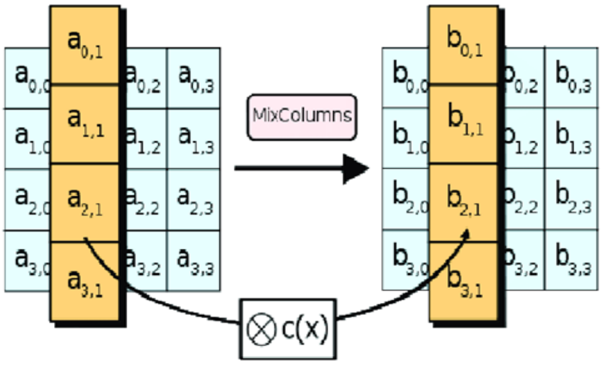


Figure 4. The Operation of the Mix Columns

D. Addroundkey Function

The final function is the AddRoundKey stage, in which a simple bitwise XOR operation is carried out between 128 bits of the round key and the 128 bits of the present state to obtain the output of individual rounds. The 16 bytes of the input matrix are considered 128 bits and are XORed to the 128 bits of the round key. The output is the cipher text when it occurs as the final round; otherwise, the 128 bits obtained are considered as 16 bytes input and return to perform a different round. Fig. 5 shows the AddRoundKey function.

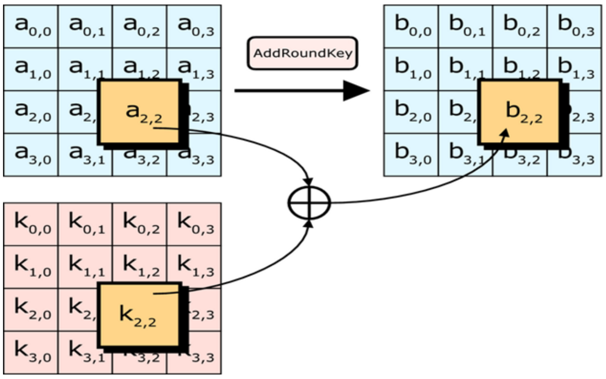


Figure 5. The Operation of the AddAroundKey

Implementing AES algorithms on FPGAs is a promising solution to increase the security and efficiency of IoT applications: With FPGAs providing resource usage and parallelism, hybrid architectures can be designed to accelerate AES implementation with minimal resources. AES implementation with minimum resource usage and acceleration. However, the challenges have to be overcome, though future research and advances in FPGA technology can hopefully mitigate those barriers. At the end of the day, FPGA based AES implementations will revolutionize IoT security and enable adoption of secure communications to various IoT domains in abundance.

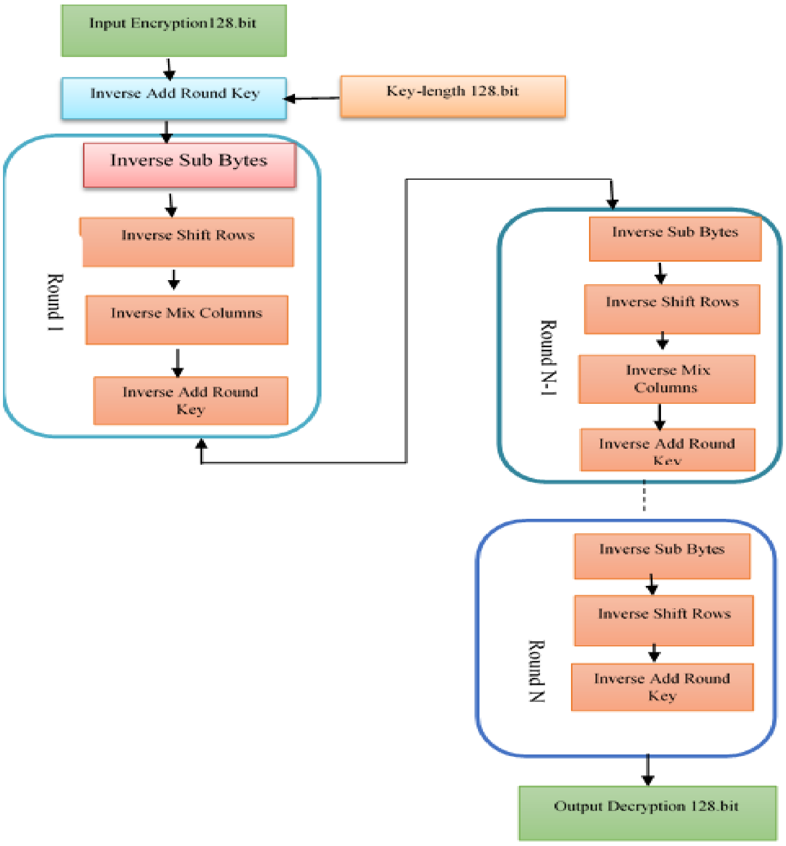


Figure 6. Flowchart for the operation of AES-128-bit Decryption

The Internet of Things (IoT) development has been rapid, which has added many conveniences and benefits to our lives daily. Yet there has been legitimate concern raised that security and privacy of IoT devices and systems will deteriorate with this expansion. These challenges are addressed by using standards and guidelines developed and promoted by the National Institute of Standards and Technology (NIST) to help develop and promote IoT security standards and guidelines. The aim of this paper is to bring to the forefront the value of NIST in improving IoT security as well as the need to adopt its recommendations.

The decryption process of AES algorithm is done in the reverse rounds order of encryption process. Each round consists of the four processes conducted in the reverse order, Add round key, Mix columns, Shift rows, and Byte substitution. Since sub-processes in each round are in a reverse manner, decryption algorithms need to be separately implemented, although they are closely related, as shown in Fig. 6.

A number of modifications were conducted by the researchers at some stages of the AES standard version to increase the processing speed or reduce the area using different kits of FPGAs techniques[15, 25, 26]. But most of the modifications that have been made have not been approved by an authoritative agency like NIST[27-29].

Computing processing engine architecture model design on FPGA

In this architecture model, a processing engine is used to process an image on both board's sides, transmitter and receiver. Fig. 7, shows the top-level design. Two different boards are used, DE1_Soc and NEEK, to make use of their abilities to design circuits for various applications of data processing. The DE1_Soc board is used at the sender part, while the NEEK board is used at the receiver part.

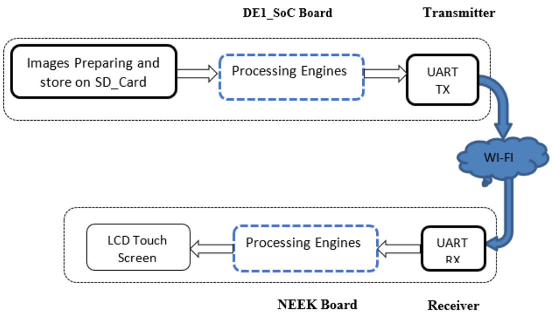


Figure 7. Top Level design

The components of top-level design as follows:

- **Image Preparation and storage on SD Card**: in this stage, images are prepared for processing within the proposed architecture. It includes three activities: reading the image, resizing the image to multiple of 128_byte and storing on SD Card. SD card is 2 Gb for NEEK board and 4Gb for DE1_SoC.
- **Processing Engines (PE):** They simultaneously process the image data in a parallel manner. The processing engine consists of the necessary functions of the encryption/decryption processing unit, on-chip memory, sync, and time/multi-clock units. The processing in each PE depends on using pipelining architecture.
- **UART** is a protocol for asynchronous serial communication with a configurable speed that uses bi-directional for transmitting and receiving the serial data. The synchronizing of the output bits from the transmitting device and the receiving devices does not need a clock signal. The device of UART has two signals: transmitter (Tx) and receiver (Rx)
- **The Wi-Fi controller** transfers images between two boards: DE1-SoC and NEEK boards. Both boards had the same TLDs. Wi-Fi in DE1_SoC operates as (Server) and Wi-Fi in NEEK works as (Client), so the transmission will be easy.

In the proposed architecture, the image is prepared for processing through a series of activities. First, the image is read and then resized to multiples of 128 bytes. This step ensures consistency and efficient processing under the AES algorithm. Finally, the re-edited images are stored on an SD card, which acts as a secure and reliable storage medium. This component ensures that the input data is properly prepared for AES encryption/decryption, which increases security and efficiency. To achieve parallel processing of image data, the proposed architecture leverages a parallel processing engine (PE); the PE consists of core components such as encryption/decryption processing units, on-chip memory, synchronization, and multiple time/clock units. By using a pipelined architecture, each processing engine (PE) can handle multiple tasks simultaneously, increasing processing speed and efficiency. The use of FPGAs enables the implementation of multiple processing engines, increasing the system's ability to process large amounts of data. The Universal Asynchronous Receiver and Transmitter (UART) protocol plays a key role in facilitating communication between devices in the architecture. UART enables asynchronous serial communication at configurable speeds. UART uses a bidirectional communication method, allowing transmitters and receivers to exchange data seamlessly. A notable advantage of UART is that it does not require a clock signal to synchronize output bits. This feature simplifies the communication process and contributes to the overall efficiency of the system.

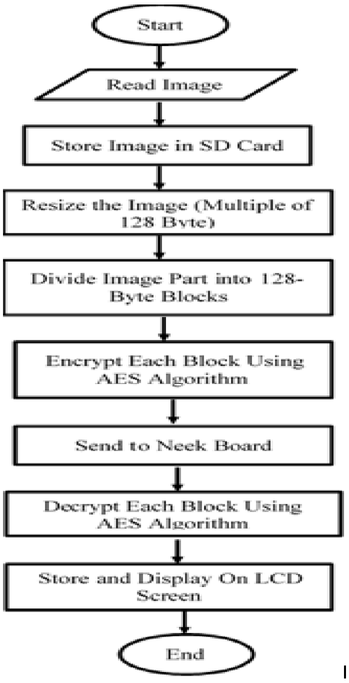

Start
Read Image
Store Image in SD Card
Resize the Image (Multiple of 128 Byte)
Divide Image Part into 128-Byte Blocks
Encrypt Each Block Using AES Algorithm
Send to Neek Board
Decrypt Each Block Using AES Algorithm
Store and Display On LCD Screen
End

Figure 8. Flowchart for Computing Engine.

And the Wi-Fi controller acts as an important part implementing image transmission between two boards, DE1-SoC and NEEK. In terms of the top-level design (TLD) of the two boards they share the same design. In this setup, the Wi-Fi on DE1-SoC is as a server and the Wi-Fi on NEEK is as a client. With this configuration, image transmission and efficient data transfer is possible between devices.

The proposed architecture uses the Wi-Fi technology to support secure and reliable communication, and at the same time, increases the security level of the overall system. This means that AES algorithms can be implemented in FPGAs, which is a promising approach for applications in the IoT and a promising solution to boosting applications' security and efficiency. The proposed architecture enables transmitting and processing data securely between IoT devices by using a high-level design including image preparation, processing engine, UART communication, and Wi-Fi controller. Together these components provide high speed encryption and decryption, parallel throughput, communication integration, and secure data transfer. Increasingly both in IoT and in the broader environment, the need to incorporate AES in FPGAs to protect data and increase the overall efficiency of IoT applications is also growing. Figure 8 shows the operation details of the proposed image processing architecture on both sender and receiver sides with the respective flowcharts. Then one type of images at a time is read in the card and stored on the SD Card. After that block are sent to the engine to process the encoding and decoding. Image segments are encoded and sent to the NEEK board where the decoding of segments is performed, segments of image are combined to reconstruct the original image, and the image is displayed on the LCD. 128 bytes are processed each time.

Results and Discussion

This proposed architecture is implemented using a computing processing engine for the two parties to process an image simultaneously. The processing includes original images of different types (color and gray), where the image has first been resized the size, then saved on the SD Card inside the DE1_SoC board. The computing processing engine does all the important encryption operations and sends them through WIFI to the NEEK board. Both boards have the same top-level design of the processing engine. The NEEK board receives the encrypted image and the decryption operation inside its computing processing engine and sends the image to the LCD screen. Fig. 9 and 10 present examples of the results for color and gray images respectively.

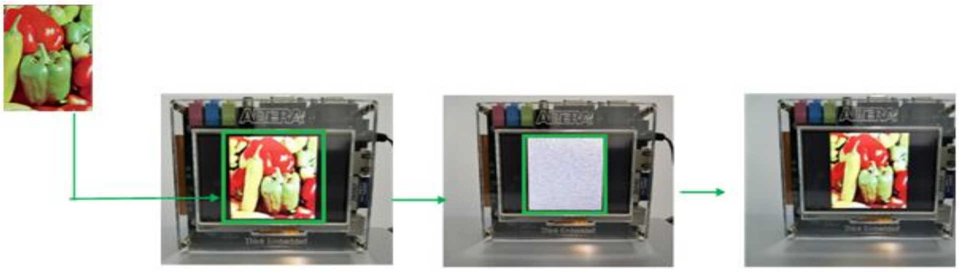


Figure 9. Stages for encryption and decryption color image.

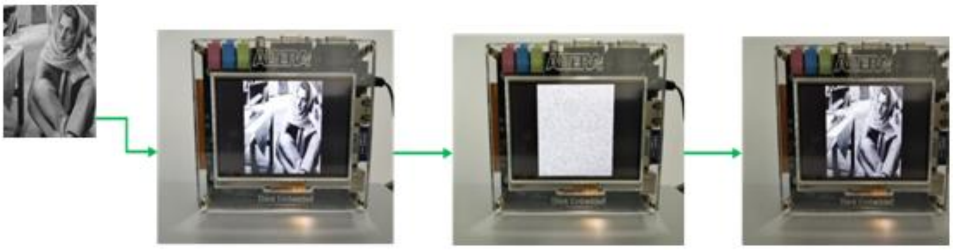
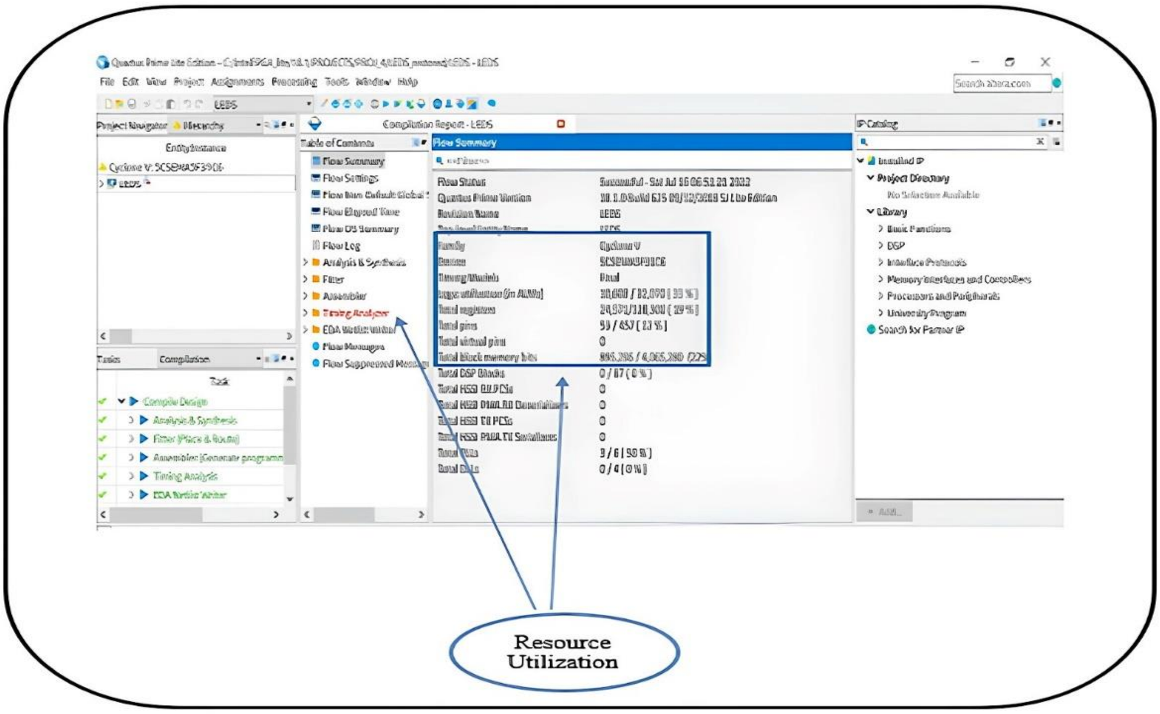


Figure 10. Stages for encryption and decryption gray image.

Fig. 11 and 12 show that the processing engine processes the data based on CAD tools. Quartus prime 18.1 a chipset (5CSEMA5F31C6) cyclone V on FPGA DE1-SoC and NEEK. All these results from Quartus compilation files show that the maximum frequency used in computing processing engine is 600MHz, so the processing time will decrease, and throughput will increase. Also, the

Quartus compilation report presents resource utilization such as register, pin memory bit, logic element, PLL, and others. Table 1 shows the performance of computing processing engine[30-32].

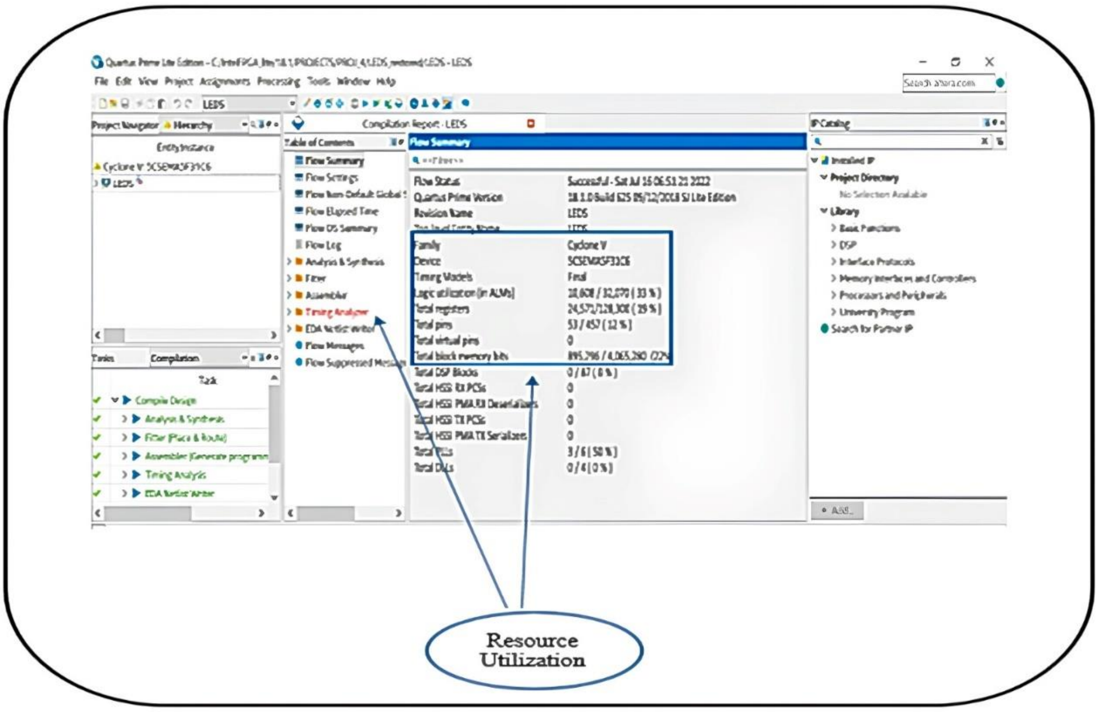
This architecture is characterized by its flexibility; it can be used for any kind of image or data processing by changing the part of the program in the processing engine with a program of the application that the user wants to implement it[33-36].



Flow Summary

Flow Status	Successful - Sat Jul 16 06:51:21 2022
Quartus Prime Version	18.1.0 Build 625 09/12/2018 SJ Lite Edition
Revision Name	LEDS
Top-level Entity Name	LEDS
Family	Cyclone V
Device	5CSEMA5F31C6
Timing Models	Final
Logic utilization (in ALMs)	10,608 / 32,070 (33 %)
Total registers	24,571/128,300 (19 %)
Total pins	53 / 457 (12 %)
Total virtual pins	0
Total block memory bits	895,296 / 4,065,280 (22
Total DSP Blocks	0 / 87 (0 %)
Total HSSI RX PCSs	0
Total HSSI PMA RX Deserializers	0
Total HSSI TX PCSs	0
Total HSSI PMA TX Serializers	0
Total PLLs	3 / 6 (50 %)
Total DLLs	0 / 4 (0 %)

Figure 11. The Compilation Report for Frequency for Processing Engine.



Flow Summary

Flow Status	Successful - Sat Jul 16 06:51:21 2022
Quartus Prime Version	18.1.0 Build 625 09/12/2018 SJ Lite Edition
Revision Name	LEDS
Top-level Entity Name	LEDS
Family	Cyclone V
Device	5CSEMA5F31C6
Timing Models	Final
Logic utilization (in ALMs)	10,608 / 32,070 (33 %)
Total registers	24,571/128,300 (19 %)
Total pins	53 / 457 (12 %)
Total virtual pins	0
Total block memory bits	895,296 / 4,065,280 (22
Total DSP Blocks	0 / 87 (0 %)
Total HSSI RX PCSs	0
Total HSSI PMA RX Deserializers	0
Total HSSI TX PCSs	0
Total HSSI PMA TX Serializers	0
Total PLLs	3 / 6 (50 %)
Total DLLs	0 / 4 (0 %)

Figure 12. The Compilation Report for Resource Utilization for The Processing Engine.

Published Online First: November, 2024
https://doi.org/10.21123/bsj.2024.8931
P-ISSN: 2078-8665 - E-ISSN: 2411-7986

Table 1. Show the Analysis Computing Processing Engine

Features	Result
Frequency Maximum (Fmax)	600MHz
Time	1.6 ns
Throughput	76.8 Gbps
Logic Utilization	10.608/32.070 (33%)
Memory Bit	895.296 (22%)
Total Pin	53/457 (12%)
Total Register Number	24.571/128 (19%)
PLL	3/6 (50%)

Comparison with Other Works

This section compares the results of the previous work and our proposal architecture, including a single computing processing engine and quad computing processing engine design on Altera Quartus Prime 18.1 cyclone V 5CSEMA5F31C6 FPGA platform for Maximum frequency, throughput, and other performance parameters. Also, a frequency comparison between different implementation techniques is performed and our architectures. The frequency comparison of different techniques is shown in Fig. 13.

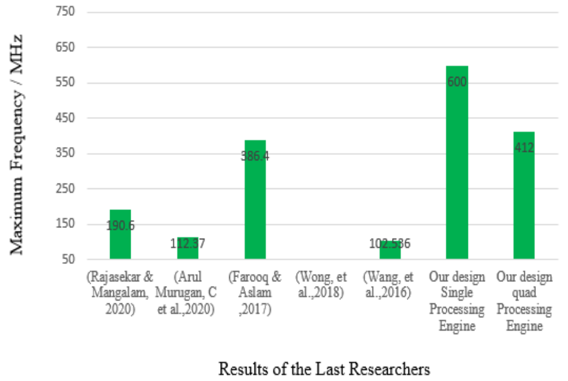


Figure 13. Maximum Frequency Performance of Computing Engines of Image.

It can be seen in the figure below that (Rajasekar & Mangalam, 2020)[2] obtained a max frequency of 190.6 MHz, Arul Murugan et al.[36] obtained Fmax = 112.37 MHz, Farooq & Aslam [37] got a frequency of 886.4 MHz, and Wong et al. [38] obtained a frequency of 102.536 MHz In our architecture, 600 MHz and 412 MHz are received in single and quad processing engines, respectively. Finally, performing a throughput comparison between previous techniques and the proposed architecture is performed. This comparison is presented in Fig. 14. It can be seen from this figure that (Rajasekar & Mangalam)[2] obtained 3.61 Mbps, (Arul Murugan et al.) received 14.383 Mbps, Noor Basha et al. [9] obtained 867Mbps, and Farooq & Aslam[37] obtained 11.35 GB/s. Compared to the proposed architecture, our implementation on DE1 and NEEk boards gives more throughput which are 76.8 Gb/s and 210.9 Gb/s for single engine and quad engine, respectively.

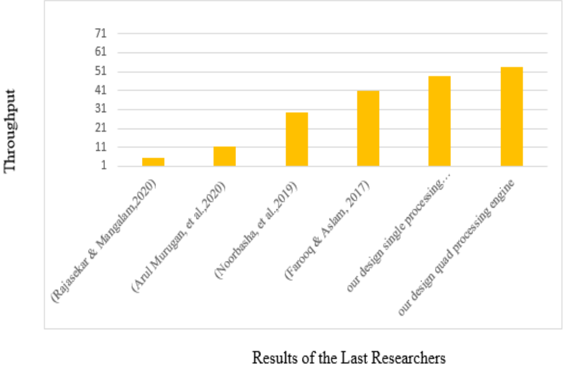


Figure 14. Throughput Performance of Computing Engines of Image.

Conclusion

This paper provides a brief overview of a proposed architecture and partial results that obtained from the implementation. The proposed architectures implemented consist of a processing engine on both sides, transmitter and receiver. The implementation is done using two different boards: Altera® Development and Education DE1_Soc(server), and NEEK board(client) which includes (LUTs, Flip Flops, RAM Tiles) connected using Fi-Wi on both boards, and the software is Altera Quartus Prime 18.1 cyclone V 5CSEMA5F31C6 for synthesis and simulation. To evaluate the performance of the proposed architecture, the numerical results are obtained from the implementation, which are presented its efficiency by making use of modern technologies of FPGA to gain more benefits.

There are several important topics that remain to be developed in this area for future research, as follows:

- Investigation of the possibility of applying this architecture of multiple computing processing engines used with video from digital cameras using FPGA and encoding it using a 128.bit AES algorithm.
- Further investigation recommends architecture performs other algorithms in the spatial and temporal parallelism and makes them work as a package of algorithms to complicate the attacker's attacks.
- Development of the architecture can be enhanced by utilizing modern boards of FPGA technologies to take advantage of resource utilization and improve the VHDL code that was implemented to get better achievement alongside with reducing the resources that are used.
- Expanding to optimizing power consumption or expanding the range of encryption algorithms.

The security of data with implementation of AES algorithm on FPGA provides a highly efficient solution. The superior performance of the proposed architecture is demonstrated by its impressive operating frequency of 600 MHz and the outstanding throughput of 76.8 Gbps. Due to the capability to optimize the resource utilization, to provide flexibility, and to ensure data security, FPGA based implementation of AES is becoming a preferred choice. The implementation of AES on FPGA has huge potential for electronic data transmission security as the demand for that continues to grow. I think implementing the AES algorithm on FPGAs for increasing security and efficiency of the IoT applications has many benefits but it also has many disadvantages to be considered and need to be addressed. Organizations can overcome these challenges by optimizing costs, improving scalability, managing power consumption and simplifying the development environment to enjoy the full AES algorithm processing power provided by FPGAs. This recovery procedure works as a kind of roadmap for those organizations that want to build more secure and more efficient IoT applications with this technology.

Authors' Declaration

- Conflicts of Interest: None.
- We hereby confirm that all the Figures and Tables in the manuscript are ours. Furthermore, any Figures and images, that are not ours, have been included with the necessary permission for re-publication, which is attached to the manuscript.
- No animal studies are present in the manuscript.
- No human studies are present in the manuscript.
- No potentially identified images or data are present in the manuscript.
- Ethical Clearance: The project was approved by the local ethical committee at Faculty of Electronic Engineering Technology, University Malaysia Perlis.

Authors' Contributions statements

The author (N.M.) conceived of the presented idea and developed the theory through discussions with the other authors. The authors (A.A., R.A. and M.S.) supervised the study and verified the main and sub-criteria. The authors (R.M. and N.T.) Verified the analytical methods. The author (Q.M.) verified the statistical analysis. The authors (J.A. and T.A.) assisted in enhancing the manuscript in terms of readability and proofreading. All authors discussed the results and contributed to the final manuscript.

References

1. Jabir H, Fanfakh A. An Overview of Parallel Symmetric Cipher of Messages. J Univ Babylon Pure Appl Sci. 2023 Jun 30; 31(2): 19 -33. https://doi.org/10.29196/jubpas.v31i2.4652.
2. Rajasekar P, Mangalam H, Kumar CS. Logic Realization of Galois Field for AES SBOX using Quantum Dot Cellular Automata. J Supercomputing 2023 Feb; 79(3): 3024-3054. https://doi.org/10.1108/CW-04-2019-0039.
3. Qassir SA, Gaata MT, Sadiq AT. Modern and Lightweight Component-based Symmetric Cipher

Algorithms. Aro Sci J Koya Univ. 2022; 10(2): 152-168. http://dx.doi.org/10.14500/aro.11007.

4. Phadikar A, Mandal H, Chiu TL. Parallel hardware implementation of data hiding scheme for quality access control of grayscale image based on FPGA. Multidimens Syst Signal Process. 2020 Jan; 31:73-101. https://doi.org/10.1007/s11045-019-00650-x
5. Mohammed NQ, Hussein QM, Sana AM, Khalil LA. A Hybrid Approach to Design Key Generator of Cryptosystem. J Comput Theor Nanosci. 2019 Mar 1; 16(3): 971-977. https://doi.org/10.1166/jctn.2019.7985.
6. Nabil M, Khalaf AA, Hassan SM. Design and implementation of pipelined and parallel AES encryption systems using FPGA. Indonesian J Electr Eng Comput Sci. 2020 Oct; 20(1): 287-299. http://doi.org/10.11591/ijeecs.v20.i1.pp287-299
7. Hameed ME, Ibrahim MM, Abd Manap N. Review on improvement of advanced encryption standard (AES) algorithm based on time execution, differential cryptanalysis and level of security. J Telecommun Electron Comput Eng. 2018 Jan 15; 10(1): 139-145.
8. Mane PB, Mulani AO. High speed area efficient FPGA implementation of AES algorithm. Int J Reconfigurable Embed Syst. 2018 Nov; 7(3): 157-165. https://doi.org/10.11591/ijres.v7.i3.pp157-165
9. Noorbasha F, Cheruvu JH, Boina P, Battineni SV. Design of AES based cipher and decipher cryptography system using Verilog HDL. J Phys.: Conf Series. IOP Pub., 1804 (2021): 012170. https://doi.org/10.1088/1742-6596/1804/1/012170.
10. Mohammed NQ, Amir A, Salih MH, Arrfou H, Mohammed Q. Implementation Dual Parallelism Cybersecurity Architecture on FPGA. J Commun. 2022 May; 17(5): 396-402. https://doi.org/10.12720/jcm.17.5.386-392
11. Mohammed NQ, et al. A Review on Implementation of AES Algorithm Using Parallelized Architecture on FPGA Platform. 2023 IEEE Int Conf Adv Sys Emrg Tech. (IC_ASET) 2023 Apr 29 (pp. 1-6). IEEE. https:doi.org/10.1109/IC_ASET58101.2023.10150938
12. Shahid Mustafizur Rahman. Deep learning for Internet of Things (IoT) network security. 2021. PhD Thesis. Inst. Polytechnique de Paris.
13. Rupani A, Sujediya G. A Review of FPGA implementation of Internet of Things Int J Innovative Res Comp Comm Eng. 2016 Sep; 4(9): 16203-16207.
14. Rose K, Eldridge S, Chapin L. The internet of things: An overview. I Soc. 2015 Oct 15; 80: 1-50.
15. Burhan M, Rehman RA, Khan B, Kim BS. IoT elements, layered architectures and security issues: A comprehensive survey. sensors. 2018 Aug 24; 18(9): 2796-2832. https://doi.org/10.3390/s18092796.
16. Marafa FM, Sa'ad S, Tukur A, Mohammed A. A Review on Impact of Internet of Things (IoT) on Individual Privacy in Smart Home Systems. 2021 2nd Int Conf Intel Eng Manag. 2021 Apr 28 (pp. 127-131). IEEE. https://doi.org/10.1109/ICIEM51511.2021.9445330
17. Elkhodr M, Shahrestani S, Cheung H. The internet of things: New interoperability, management and security challenges. arXiv preprint arXiv: 1604.04824. 2016 Apr 17;8(2):85-102 http://dx.doi.org/10.5121/ijnsa.2016.8206
18. Shaukat K, Alam TM, Hameed IA, Khan WA, Abbas N, Luo S. A review on security challenges in internet of things (IoT). 26th IEEE International Conference on Automation and Computing 26th Int conf automation and comp. (ICAC). 2021 Sep 2 (pp. 1-6). IEEE. https://doi.org/10.23919/ICAC50006.2021.9594183.
19. Hussien QM, Habeeba FA. Survey on data security techniques in internet of things. Al-Kunooze Sci J. 2021; 2(2):27-37.
20. Karimian GH, Rashidi B. A high speed and low power image encryption with 128-bit AES algorithm. Int J Electr Comput Eng.. 2012 Jun 1; 4(3): 367-372 .https://doi.org/10.7763/IJCEE.2012.V4.514.
21. Mohammed NQ, Salih MH, Aliana R, Hussein QM, Khalid NA. Design and implementation image processing functional unit using spatial parallelism on FPGA. ARPN J Eng Appl Sci. 2018; 13(15): 4514-4520.
22. Gore M, Deotare V. FPGA Implementation of Area Optimized AES for Image Encryption/Decryption Process. I J N G C A. 2013; 1(9): 23-26. https://doi.org/10.1109/ICCSP.2015.7322746.
23. Rahimunnisa K, Karthigaikumar P, Rasheed S, Jayakumar J, SureshKumar S. FPGA implementation of AES algorithm for high throughput using folded parallel architecture. Secur Commun Netw. 2014 Nov; 7(11): 2225-2236. https:doi.org/1002/sec.651
24. AL-ODAT Zeyad, Mazhar Ali, Assad Abbas, Samme Ullah. Secure hash algorithms and the corresponding FPGA optimization techniques. ACM Comput Surv. 2020; 53(5): 1-36. https://doi.org/10.1145/3311724.
25. Jumaa NK. Survey: internet of thing using FPGA. Iraqi J Electr Electron EngJ. 2017; 13(1): 38-45. https://doi.org/10.33762/eeej.2017.128785.
26. Tausif M, Ferzund J, Jabbar S, Shahzadi R. Towards Designing Efficient Lightweight Ciphers for Internet of Things. KSII Trans Internet Inf Syst. 2017 Aug 1; 11(8):1-10. https://doi.org/10.3837/tiis.2017.08.014
27. Willam S. Cryptography and Network Security: Principles and Practice (Global Edition-). Pearson Education; 2022. http://elib.vku.udn.vn/handle/123456789/2881

28. Zodpe H, Sapkal A. An efficient AES implementation using FPGA with enhanced security features. J King Saud Univ Eng. Sci. 2020 Feb 1; 32(2): 115-122. https://doi.org/10.48550/arXiv.2101.01177.
29. Kamalakkannan K, Mudalige GR, Reguly IZ, Fahmy SA. High-level FPGA accelerator design for structured-mesh-based explicit numerical solvers. 35th IEEE International Parallel and Distributed Processing Symposium (IPDPS) 2021 May 17 (pp. 1087-1096). https://doi.org/10.48550/arXiv.2101.01177.
30. Vaigandla KK, Karne R, Siluveru M, Kesoju M. Review on Blockchain Technology: Architecture, Characteristics, Benefits, Algorithms, Challenges and Applications. Mesopotamian J Cyber Security. 2023 Mar 24; 2023: 73-85. https://doi.org/10.58496/MJCS/2023/012.
31. Al-Amri RM, Hamood DN, Farhan AK. Theoretical Background of Cryptography. Mesopotamian J. Cyber Security. 2023 Jan 26; 2023: 7-15. https://doi.org/10.58496/MJCS/2023/002.
32. Rimani R, Naima HA, Pacha AA, Ramos JA. An Efficient Image Encryption Using a Dynamic, Nonlinear and Secret Diffusion Scheme. Baghdad Sci J. 2021 Sep 1; 18(3): 628-639. https://doi.org/10.21123/bsj.2021.18.3.0628.
33. Khudhair ZN, Nidhal A, El Abbadi NK. Text multilevel encryption using new key exchange protocol. Baghdad Sci J. 2022 Jun 1; 19(3): 619-630. https://doi.org/10.21123/bsj.2023.7315.
34. Abdul-Ghani SA, Abdul-Wahhab RD, Abood EW. Securing text messages using graph theory and steganography. Baghdad Sci J. 2022 Feb 1; 19(1): 0189-. https://doi.org/10.21123/bsj.2022.19.1.0189.
35. Salim KG, Al-alak SM, Jawad MJ. Improved image security in internet of thing (IoT) using multiple key AES. Baghdad Sci J. 2021 Jun 1; 18(2): 417-429 https://doi.org/10.21123/bsj.2021.18.2.0417
36. Arul Murugan C, Karthigaikumar P, Sathya Priya S. FPGA implementation of hardware architecture with AES encryptor using sub-pipelined S-box techniques for compact applications. Automatika. 2020 Oct 1; 61(4): 682-693. https://doi.org/10.1080/00051144.2020.1816388.
37. Farooq U, Aslam MF. Comparative analysis of different AES implementation techniques for efficient resource usage and better performance of an FPGA. J King Saud Univ. Comput Inf Sci. 2017 Jul 1; 29(3): 295-302. https://doi.org/10.1016/j.jksuci.2016.01.004
38. Wong DS, Tabereaux A, Lavoie P. Anode effect phenomena during conventional AEs, low voltage propagating AEs & non-propagating AEs. L Met. 2014. 2016: 529-534. https://doi.org/10.1002/9781118888438.ch90

تصميم بنية هجينة لتنفيذ خوارزمية AES على FPGA لتطبيقات إنترنت الأشياء

ندى قاسم محمد[1]، أميز عامر[1]، معتز حامد صالح[2]، ر.بادليشا أحمد[1]، ريما معتم[1]، نسرين ثلجي[3]، قاسم محمد[4]، جمال كامل خ. عباس[5]، طه رعد الشيخلي[5]

[1] كلية تكنولوجيا الهندسة الإلكترونية، جامعة ماليزيا بيرليس، 02600 أراو، ماليزيا.
[2] مجموعة الأتمتة والتصميم والهندسة، فليكس، بينانج، 11500، ماليزيا.
[3] قسم الروبوتات والذكاء الاصطناعي، جامعة جدارا، إربد، الأردن.
[4] قسم علوم الامن السيبراني، كلية الكنوز الجامعة، بصرة، العراق.
[5] هندسة تقنيات الحاسبات، كلية النسور الجامعة، بغداد، 10036، العراق.

الخلاصة

تتطلب خوارزميات التشفير عالية الأمان مثل AES قدرات حسابية عالية لتحقيق أمن المعلومات. لذلك، من الضروري استخدام بنيات الحوسبة المتوازية التي تستغل التقنيات الحديثة في التوازيات المكانية للحصول على أكبر قدر من القوة الحسابية التي يمكن تصورها. تم إدخال تقنيات مختلفة لتحقيق المعالجة المتوازية. إحداها هي مصفوفات البوابات القابلة للبرمجة ميدانيًا (FPGAs)، والتي تتميز بخصائص جيدة مناسبة لتنفيذ بنيات متوازية مع استهلاك أقل للطاقة. تهدف الورقة إلى تصميم وتنفيذ جهاز إرسال واستقبال مدمج لبنية محرك المعالجة الحاسوبية ذو أداء عالٍ للحصول على إنتاجية أفضل على تقنية FPGA لتشفير وفك تشفير الصور. في هذا التصميم، تم استخدام لوحتين، "DE1_Soc و NEEK board" مع جهاز Altera Quartus prime 18.1، وجهاز cyclone V 5CSEMA5F31C6 FPGA للتوليف والمحاكاة. أظهرت نتائج التنفيذ أن البنية المقترحة تتمتع بكفاءة الأداء من حيث تردد التشغيل 600 ميكا هرتز والإنتاجية 76.8 كيكا هرتز.

الكلمات المفتاحية: خوارزمية AES، التشفير/ فك التشفير، غرض الشيء، FPGA، VHDL، الأمان.