Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 45 No. 4 (2024)

VLSI design of a Novel Encryption Standard using Nano Technology

P Raja Lingam[1] , Dr. Ashish Joshi[2]

[1]Research Scholar and [2]Research Guide,

Department of Electronics & Communication Engineering,

Dr. A. P. J. Abdul Kalam University, Indore, MP, India

Abstract

Due to the increasing number of small internet-connected devices, end-to-end security is of the utmost importance in the World Wide Web of Things (IoT). Encryption must be tailored to work with Internet of Things products because they aren't exactly flush with cash. According to this scholarly research, resource-constrained Internet of Things (IoT) devices may benefit from using the newest version of encryption (AES) with a field-programmable gated arrays (FPGA) using 65-nm technology. You may think of the proposed 8-bit data path as having five major components. All of this transitional data, plaintext, as well as keys are stored in two separate registry banks: the Key-Register and the State-Register. The incorporation of Shift-Rows into the State Register allows for more efficient use of space. We built an 8-bit efficient block with four internal registers that read in 8 bits while sending them out; this allowed us to transform the Mix-Column to an 8-bit data channel. In addition, both the password expansion as well as encryption processes share an optimized Sub-Byte. Some Sub-Bytes were consolidated and streamlined in order to increase performance. The clock gating technique is included in the design to reduce power consumption. There was a 35% to 2.4% improvement in application-specific integrated circuit (ASIC) performance when compared to earlier research of a same nature. Based on the findings, it appears that the proposed design is an excellent cryptosystem for IoT devices with limited power consumption.

The following terms are used in the index: lightweight cryptography, Internet of Things (IoT), clock gating, and Advanced Encryption Standard, also algorithm.

Introduction

A vast system of interconnected, tiny devices that are capable of exchanging data is called the "Internet of Things" (IoT). Electronic health records and public transportation systems may both benefit from this information if these tiny devices were to link together to form a sophisticated network capable of sensing, communicating, and processing data. The exponential development in the number of connected devices has made it increasingly complex to ensure the privacy of all supplied information. The bulk of end-node small devices just can't handle encryption. With the proliferation of the Internet of Things (IoT), studies aimed at creating affordable encryption architecture are more important than ever. Internet of Thing end-node devices that operate on batteries are particularly well-suited to lightweight cryptographic methods. Lora Wan and the Internet of Things (IoT) are two of the most secure symmetric cryptography systems now in use. They are both used by several protocols and networks on the Internet. The level of security offered by AES is dependent on the key's lifetime. Many protocols and applications for the Internet of Things (IoT) rely on the Advanced Encryption Standard, also known as AES, with a 256-bit key because it is secure enough to withstand the quantum era. Software implementation of AES has some downsides, such as increased power consumption and a long processing time.

There has been a dramatic increase in the usage of hardware implementations in applications with high performance requirements and devices with limited resources. Low throughput is a problem with AES

implementations for devices with limited resources when using data paths of 8, 16, or 32 bits. In this paper, we provide a data path architecture for devices with limited resources or mobile systems on a chip. Minimal internal wiring is required for 8-bit data path compared to 32-bit data path. Using the lowest part of the design, merging functionalities, and reducing the total amount of blocks are all ways we're trying to make the design smaller. Two distinct register banks that utilize shift-register memory—the State-Register and the Key-Register—are included into our design for the purpose of storing keys, plaintext, and intermediate results. Key multiplication and encryption are two operations that rely heavily on these registers. It is possible to put the proposed design into action using 65-nm technology, which is employed by FPGA and ASIC. Due to its foundation in NIST lightweight cryptography, our proposed architecture's ASIC implementation is well-suited for cryptosystems in IoT devices with little resources. Compared to previous attempts, this hardware implementation seems to be better. Our proposed architecture for 65 nm technology improves the chip's area and region-delay product (ADP) by 2.4%. The opposite side of the power arrival enhances the core region by 22.1%. The main contribution of our study is a small AES architecture that is made specifically for low-powered Internet of Things devices. The following implementation approaches and block designs will be utilized to accomplish this purpose.

First, the Shift-Rows are included into the State-Register to simplify the required logic.

2) We achieve a 15.5% reduction in space need by optimizing the Sub-Bytes block, that we then share between the key expansion as well as encryption stages.

3) We build an 8-bit optimized Mix-Columns module utilizing the 8-bit data route structure that makes use of 8-bit input and output and has a round key, even though 32 bits are required for simultaneous execution of Mix-Columns. Bit by bit, the output is then delivered to Add-Round-Key. To avoid raising the Key-Register data route to 32 bits or saving its results in the registers, it is not necessary to use 32-bit Mix-columns.

4) Applying the clock gating approach in various design components reduces energy use by 18.9% upon 65-nm technology, which is a significant improvement over the original design. What follows is an outline of the remaining content of this piece. After introducing the AES algorithm and its history, this article details the 8-bit data path AES structure and its building pieces. It then goes on to discuss the project's implementation, analysis, and compared to related studies before drawing a conclusion.

RELATED WORK

“Experts in developing low-power, multi-level Internet of Things apps that optimize AES data pathways,"

Connected devices are getting a lot of attention because current Internet of Things (IoT) products don't have any security features. An increase in safety might result from using the tried-and-true advanced identification standard (AES) block ciphers. Nevertheless, a lot of computational power and electricity/energy consumption is required for these security tasks. In this study, we showcase our AES hardware optimization solutions for multi-level secure, ultra-low-power, ultra-energy Internet of Things applications. Power and energy are maximized for data flow and key expansion in our architecture, which provides many degrees of protection through varying key sizes. Using ST FDSOI 28-nm technology, our implementation is projected to reach a maximum transference rate of 28 Mb/s and an energy per piece of around one pJ/b around 10 MHz at 0.6 V, which is comparable to the lightweight standards method PRESENT. An attack utilizing connection power calculations with less than 20,000 traces cannot compromise our suggested data path's 32-bit key, according to the security evaluation. This is in contrast to the 128-bit key.

"Designing the AES S-Box through the optimization of combinational logic"

Among the several symmetric encryption methods in use today, the Advanced Encryption System (AES) ranks high. The AES replacement box (S-box) is one of the most complex and costly parts of the AES hardware since it is the sole non-linear structure in the system. A Virtex II field-programmable gate array (FPGA) chip containing an S-Box hybrid logic architecture is utilized in the proposed project. In order to reduce latency, the

design employs a Boolean adjustment to the truth table of the logic function. The S-Box is built using fundamental gates like the AND, NOT, OR, and multiplexer. Ideally, the architecture would work well for applications that need high-speed performance while also reducing total latency. With respect to gate area, this method is well-suited for use in FPGA implementations. Here we display the hardware, overall area, and delay.

"Advanced Encryption Standard (AES) Development for 65 nm Silicon Displays and Low-Power Operating Systems with Minimum Energy Use per Encryption"

Emerging millimeter-scale IoT devices must provide sufficiently high throughput under tight space and energy constraints, making lightweight encryption circuits essential for appropriate data security. This necessitates the use of dedicated AES accelerators, which provide energy savings of many orders of magnitude compared to solutions based on microcontrollers. Lightweight AES accelerators designed to minimize power consumption are the subject of this paper's architectural investigation. Lightweight AES designs set the minimum amount of cycles per encryption based on the total amount of accessible S-boxes. Coupled with sub-/near-threshold circuit methods, we offer a low-cost ultra-energy-efficient encryption algorithm using AES component for cubic-millimeter platforms. Our test chip's remarkable consumption of electricity of 0.83 pJ/bit under 0.32V places it seven times ahead of the most effective low-cost AES design.

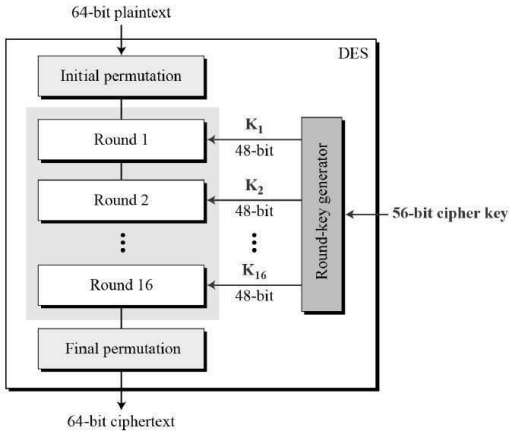
"Enhanced AES Algorithm Performance with High-Speed VLSI Architectures"

This paper introduces novel high-speed designs and the most recent hardware implementations of the AES algorithm. The proposed approach just employs combinational logic, as opposed to earlier systems that relied on look-up tables to accomplish the Sub Bytes as well as InvSubBytes conversions required by the AES algorithm. The entire potential of subpipelining's May now be realized, as the prohibitive delay created by search engine optimization in conventional approaches is no longer an issue. In addition, several inversion approaches are investigated in sector (24), and composite column computing is employed to reduce space requirements. In addition, a practical design for key expansion is revealed, which is compatible with current sub pipelined rounds units. When operating in non-feedback modes, a Xilinx XCV1000 e-8 bg560 device can execute a fully sub pipelined encryption or one comprising seven sub stages that an average rate of 21.56 Gbps, which is 79% faster than the earliest known FPGA implementation when considering equal throughput/slice. **Existing system**

Standard for Data Encryption (DES)

The symmetric-key block cipher known as DES, or the Data Encryption Standard, was published by the National Institute fo Technology and Standards, which is more often known as NIST.

DES is a Feistel Cipher implementation. The 16-round Feistel structure is utilized. A 64-bit block is used. Key size is 64 bits, although DES only uses 56 bits since 8 bits are reserved for checking purposes and aren't used with the encryption method. Here is an illustration of the general structure of DES:



With its foundation in the Feistel Cipher, DES may be defined with as little as _

- Completely recursive

- Final timetable
- Initial as well as final permutation for any further processing.

Standard for Triple Data Encryption (Triple DES)

The Data Encryption Standard, also known as is the foundation of Triple DES, an encryption method. To further strengthen security, this symmetric encryption technique employs multiple cycles of the encrypted data standard (DES). The Data Encryption Standard, or DES for short, is a cypher that employs a three-stage encryption process, which is why it is sometimes called Triple DES. It encrypts data in 64-bit chunks and serves as a block cipher. This new data encryption standard, or DES, is more secure than its predecessor. On the other hand, compared to AES, Triple DES is slower and less efficient.

The Method of Encryption

Triple DES encryption entails the following steps:

Generation of Keys

This is the initial stage of Triple DES encryption. At this stage, a key derivation technique is used to produce three distinct keys.

First Iteration

Key Generation comes before this phase. The process entails rearranging the plaintext bits in accordance with a predetermined permutation table.

Encryption in Three Stages

According to Triple DES experts, this is the most crucial step in the encryption process. In most cases, there are a total of 48 rounds. Here, we generate three levels of encryption by processing the plaintext three times and encrypting it each time using a new key.

Remaining Combinations

This finishes the process of Triple DES encryption. Here, the generated encrypted text block is subjected to a last permutation (FP) operation, the antithesis of the first permutation. The encryption text block's bits are returned to their original arrangement.

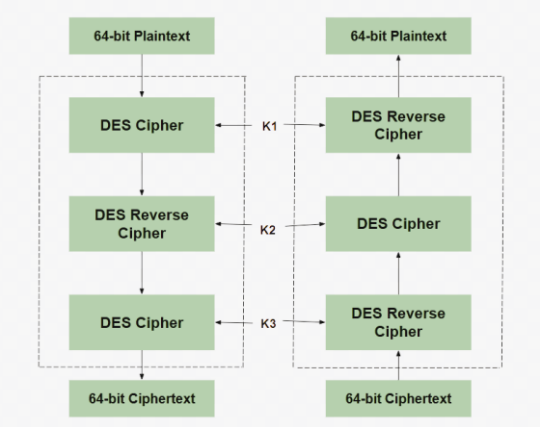


Fig: Triple DES

Elliptic Curve Security Standard (AES)

The Advanced Encryption Standard, also called AES, is a well-known encryption technology that makes data unreadable without the right key, which improves data security. The Advanced Encryption Standard (AES), developed by the National Institute of Standards and Technology (NIST), provides strong protection against infiltration using keys that are 12, 192, or 256 bits long. When it comes to encrypting files, safeguarding

sensitive information, and securing internet connection, this data security solution is effective and extensively used. One of the most important components of modern encryption, AES is well-known all over the world for its effectiveness in protecting data from cybercriminals.

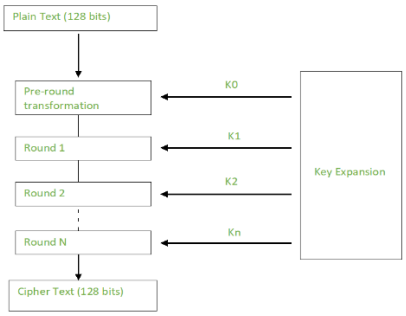


Fig: Creation of Round Keys (AES)

Proposed system:

The proposed Nano-AES architecture with an 8-bit data channel is detailed here. Figure 4.1 displays our proposed design. The system also has a control unit and an RCON block, as well as two register banks, Key-Register and State-Register that temporarily store keys and intermediate results, respectively. Another goal of Mix-Columns and Sub-Bytes is to cut out any extraneous steps.

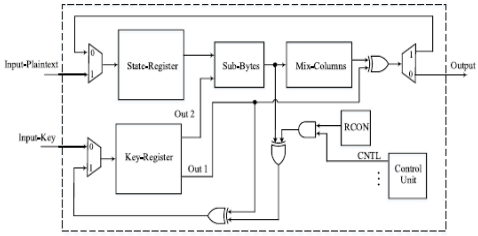


Fig. 4.1The structure of the hypothetical Nano-AES.

In the architecture, there are two channels of input that save intermediate results to register banks during key enlargement and key return. For encryption, the alternate feedback route is used. Figure 2 is a schematic representation of our State-Register concept. With eight bits as well as eight flip-flops apiece, the State-Register boasts sixteen registers. With the use of a shift-register memory structure, the State-Register accepts one 8-bit signal from the original plan and, if necessary, produces one 8-bit output. A subsequent register receives the information from the prior register and uses it to perform encryption.

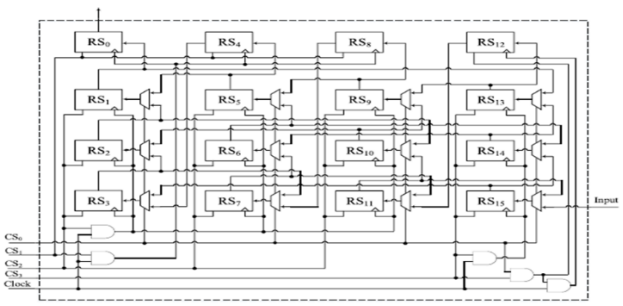


Fig.4. 2. The suggested State-Register architecture, including control circuitry, clock gating, and shift-rows.

The next register receives the encrypted data and processes it. One contains connections between the several State-Register subunits to execute Shift-Rows, as shown in Fig. 4.2, while the other illustrates the horizontal linkages among the registers that exist in each row. Because wiring changes the rows, the logic behind the shift-rows step is no longer needed. Twelve registers were required for the Shift-Row block, according to Jarvinen et al. Because of its bulk, a separate Shift-Rows block is not practical for compact layouts. The primary and secondary control signals (CSs) are responsible for choosing which registers to activate. When CS0 and CS3 are AND end, it produces a signal that activates on the next row of registers; on the first row, it's CS1. When running a state-register, there are primarily five things to do.

1) By setting all control signals to "1" and activating both internal registers, the primary plaintext may be saved into the state.See Figure 4.2 for an illustration of how the architecture receives and stores an 8-bit value to RS15 with each clock cycle. Data stored in the shift-register memory is transferred from RS15 to RS0 in response to the arrival of new 8-bit data over the inter-unit link.

2) The execution function Shift-Rows is achieved by connecting wires to internal registers. As indicated before, registers intersect internally. Activating the entries in the first, third, three fourth columns within the State-Register are necessary for the execution of Shift-Rows. You should put CS0 and CS1 as "0," then CS2 and CS3 as "1."

3) There is a single 8-bit input that controls the design, and the initial Add-Round-Key and the last round's data are stored in shift-register memory (not including Mix-Columns). Activate each of the internal registers as you go along with the process.

4) All control signals must be set to "1" for the mix columns to function. Each column of the national store, RS0 to RS3, which lasts six clock cycles, may be accessed using one 8-bit. Next, enter the information into the layout. The data is provided within four clock cycles so that the fourth column may be prepared to hold the result of Mix-Columns.

5) Using the last column only for storing data from mix-columns: In order to save the outcomes of the Mix-Column calculations, the State-Register requires four clock cycles, as described in Section III-B. The last column within the State-Register should be used to store the data. As a result, only the last column receives data in the Mix-Columns block; no other cells' data is transmitted. While doing so, you'll activate the built-in registers in the first two rows (CS3CS2CS1CS0 should be set to "1001"), which will disconnect the fourth and third portions of the State-Register. The data transfer of the State-Register may be seen in Table I for the initial phase and the Add-Round-Key. Each round will start with the same value for the registers.

Expansion of Keys

Each method iteration necessitates a single 128-bit key. Using the keys of the previous round, new keys are created for the next rounds. Devices with limited resources cannot store all each design's variables due to the large amount of memory required. A single 128-bit register is all that's needed for this ingenious key expansion mechanism. Key-Register, Sub-Bytes, RCON, & XOR all have their final columns changed as they expand. The expansion phase's most crucial stage is this:

$$\begin{aligned}
&\text{Shift}(\text{Col}_4)\\
&\text{Col}'_1 = \text{Sub}(\text{Col}_4) \oplus \text{RCON} \oplus \text{Col}_1\\
&\text{Col}'_2 = \text{Col}'_1 \oplus \text{Col}_2\\
&\text{Col}'_3 = \text{Col}'_2 \oplus \text{Col}_3\\
&\text{Col}'_4 = \text{Col}'_3 \oplus \text{Col}_4. \qquad (10)
\end{aligned}$$

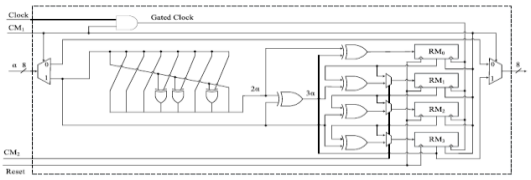


Fig.4. 5. A design for Mix-Columns using the clock gating approach and a bypass circuit is suggested.

The number of key uses determines the constants used to extend the first column of an RCON database. One easy way to use RCON's constant values—0x01, 0x02, 0x04, 0the variables x08, 0x10, 0x20, 0x40, & 0x80—is using a LUT. We opted out of using LUT implementation in our design because of its key benefits—reduced power utilization and a more compact clock network. Earlier efforts used LUT for RCON, but this one does not. By using a basic Karnaugh optimization, Van-Phuc et al. reduced the size of the RCON block.

Their RCON implementation makes use of combinational logic and a round counter. The remaining values, which are not 1B8 and 36, are obtained using an 8-bit shift registers by turning a single "1" instead. The implementation of 1B8 and 36B8 requires additional logic gates. A single "1" generated by a 8-bit shift register, an 8-bit AND, and a 4-bit NOR were the four control signals used to build RCON. An OR gate and a control signal were employed to improve the RCON architecture. Switch for the left side that is round you can see the RCON block in Figure 7.

Figure 8 shows the Key-architecture that has been suggested. The one that specifies the Key-Register is the one that uses the proposed architecture. The 16 8-bit registers, 2-1 MUX, and 8 flip-flops that make up Key-Register may each accept a few inputs. Its eight-bit inputs and outs are a nice feature. According to (10) our Key-Register provides two outputs since expanding the relevant keys requires the two lines of the previous key at the same time.

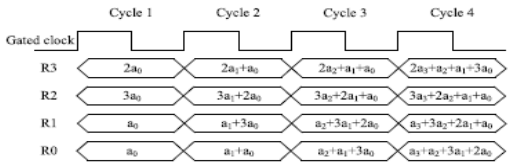


Fig. 4.6. The planned Mix-Columns timing diagram.

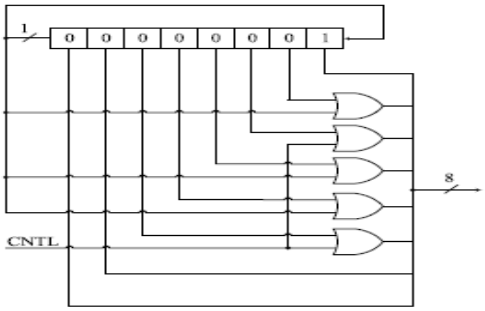


Fig.4. 7. Potential RCON node in the envisioned architecture.

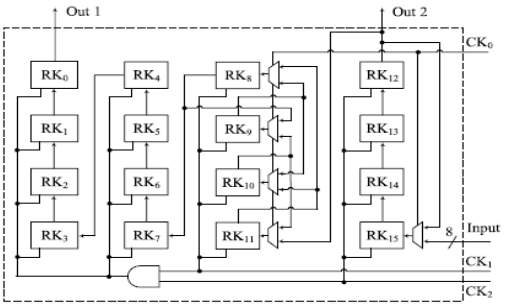
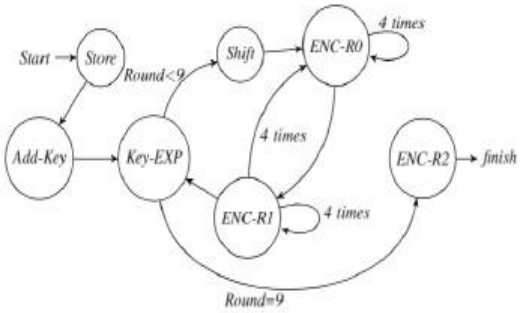


Fig.4. 8. Design of the planned Key-Register.

As seen in Figure 10, our suggested Nano-AES accelerator employs finite-state machines (FSMs). Key-Register and State-Register both allow you to save the original key as well as plain text in the same location. To initiate the first Add-Round-Key, the bypass signaling of the Sub-Bytes and Mix-Columns blocks are activated. One can use Key-EXP to prolong the key after the initial Add-Round-key and after every round that is done. You

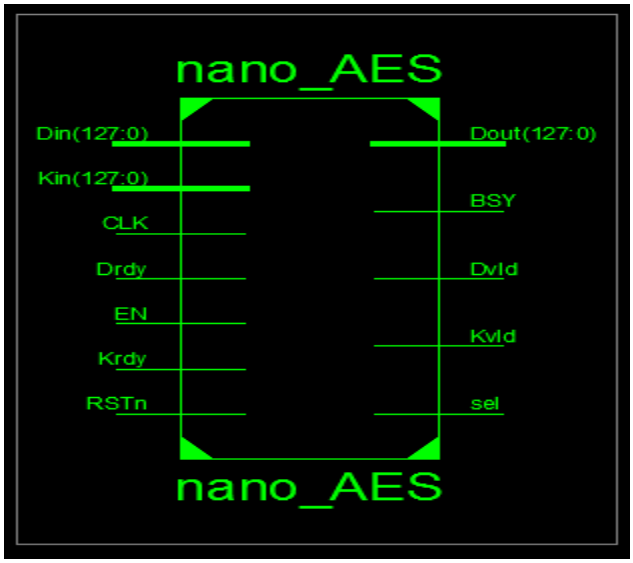


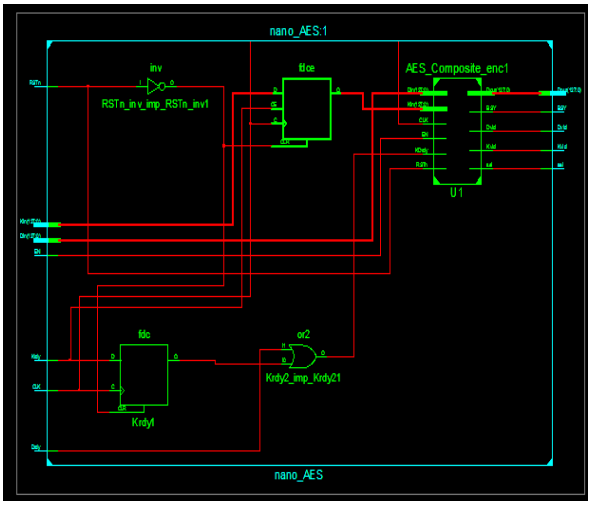
just need one clock to run Shift-Rows.

Fig.4.9. a finite-state machine is used in the suggested layout...

Results

Here is the nano-AES entity diagram.

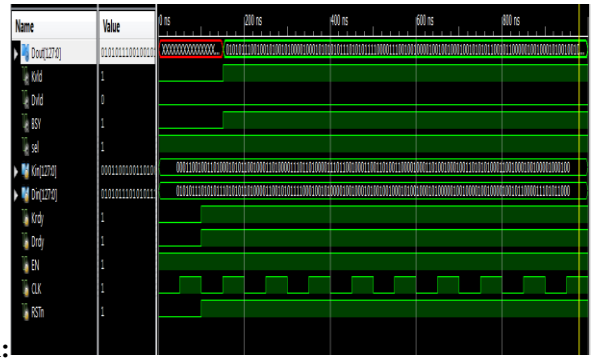




Presented here is the RTL sche

matic for Nano-AES.

Here are the findings of the simulation:



CONCLUSION

Secure communication networks are essential for the expansion of Internet the Things (IoT) connections. This is especially true given that the majority of IoT devices are rather tiny and have low resources. Small and lightweight Internet of Things (IoT) sensors can utilize the AES (Advanced Encryption Standard) symmetric cryptographic approach, which is very safe. For IoT devices with little power, our AES design is ideal because to its 8-bit data stream and two carefully defined register banks.Using 65-nm technology, we were able to save space by 15.5% and lower energy usage by 18.9% using clock gating. Using 65 nm technology and the Virtex-5 FPGA, the design proved to have an optimum power consumption despite an increase in chip area. Despite a 22.1% increase in the core region of the power rings, the suggested approach reduced power compared to previous attempts. Maintaining security using a 256-bit key especially in the quantum era is an objective of the AES design, which aims to offer post-quantum resilience while preserving resource restrictions in IoT devices. Minimizing power consumption as well as space needs is the primary goal of the optimal design architecture.

REFERENCES

1. N. Sornin, M. Luis, T. Eirich, T. Kramp, and O.Hersent, “Lorawan specification,” Lora Alliance, Tech. Rep., Jan. 2015, Pp. 1–82. [Online]. Available: Https://Loraalliance.Org/ResourceHub/Lorawanrspecification-V10.
2. Z. Liu, K.-K. R. Choo, And J. Großschädl, “Securing Edge Devices In The Post-Quantum Internet Of Things Using Lattice-Based Cryptography,”IEEE Commun. Mag., Vol. 56, No. 2, Pp. 158–162, Feb. 2018.

3. D.-H. Bui, D. Puschini, S. Bacles-Min, E. Beigné, and X.-T. Tran, "Aes Datapath Optimization Strategies for Low-Power Low-Energy Multisecuritylevel Internet-Of-Things Applications," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., Vol. 25, No. 12, Pp. 3281–3290, Dec. 2017.
4. C. Patrick and P. Schaumont, "The Role of Energy in the Light weight cryptographic Profile," In Proc. Nist Lightweight Cryptogr. Workshop, 2016, Pp. 1–16.
5. A. Moradi, A. Poschmann, S. Ling, C. Paar, and H. Wang, "Pushing the Limits: A Very Compact and A Threshold Implementation of Aes,"In Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn. Tallinn, Estonia: Springer, 2011, Pp. 69–88.
6. T. Järvinen, P. Salmela, P. Hämäläinen, and J. Takala, "Efficient Byte Permutation Realizations for Compact Aes Implementations," In Proc.13th Eur. Signal Process. Conf., 2005, Pp. 1–4.
7. W. Zhao, Y. Ha, and M. Alioto, "Aes Architectures for Minimum-Energy Operation and Silicon Demonstration in 65 Nm with Lowest Energy per Encryption," In Proc. IEEE Int. Symp. Circuits Syst. (Iscas), May 2015,
8. Pp. 2349–2352.
9. K. Shahbazi, M. Eshghi, And R. Faghih Mirzaee, "Design And Implementation Of An Asip-Based Cryptography Processor For Aes, Idea, And Md5," Eng. Sci. Technol., Int. J., Vol. 20, No. 4, Pp. 1308–1317, Aug. 2017.
10. L. Ali, I. Aris, F. S. Hossain, and N. Roy, "Design Of an Ultra High Speed Aes Processor for Next Generation It Security," Comput. Electr. Eng., Vol. 37, No. 6, Pp. 1160–1170, Nov. 2011.
11. A. Soltani and S. Sharifian, "An Ultra-High Throughput and Fully Pipelined Implementation of Aes Algorithm on Fpga," Microprocessors Microsyst., Vol. 39, No. 7, Pp. 480–493, Oct. 2015.
12. N. Ahmad, R. Hasan, And W. M. Jubadi, "Design Of Aes S-Box Using Combinational Logic Optimization," In Proc. Ieee Symp. Ind. Electron. Appl. (Isiea), Oct. 2010, Pp. 696–699.
13. S. Banik, A. Bogdanov, And F. Regazzoni, "Exploring Energy Efficiency Of Lightweight Block Ciphers," In Proc. Int. Conf. Sel. Areas Cryptogr. Sackville, Nb, Canada: Springer, 2015, Pp. 178–194.
14. H. K. Kim And M. H. Sunwoo, "Low Power Aes Using 8-Bit And 32-Bit Datapath Optimization For Small Internet-Of-Things (Iot)," J. Signal Process. Syst., 2019.