ISSN: 2321-2152

IJMECE

International Journal of modern electronics and communication engineering

E-Mail
editor.ijmece@gmail.com
editor@ijmece.com

www.ijmece.com

IJMECE
International Journal of modern electronics and communication engineering

ISSN 2321-2152
www.ijmece.com
Vol 13, Issue 1, 2025

VLSI IMPLEMENTATION OF CRYPTO COPROCESSOR USING AES AND LFSR

MRS. V. MOUNIKA[1], G. KISHORE[2], K. MAHESH[3], G.K.SRIRAM[4], S.H.V.S.R. SATISH [5]

[1]Assistant Professor, Dept. Of ECE, PRAGATI ENGINEERING COLLEGE

[2345]UG Students, Dept. Of ECE, PRAGATI ENGINEERING COLLEGE

ABSTRACT

Data security is a critical concern alongside the need for faster data processing. As processing capabilities advance, so do attacks aimed at extracting sensitive information from devices. This project focuses on enhancing the security of crypto coprocessors by integrating a Linear Feedback Shift Register (LFSR) as a key generator with the Advanced Encryption Standard (AES) algorithm.

The combination of LFSR and AES increases resistance to hardware attacks, particularly those attempting to backtrack and extract hardcoded keys often found in traditional hash algorithms. By randomizing the key input of AES, this approach makes it significantly more difficult for attackers to trace the algorithm and extract key values or sensitive data. In this work, AES with a 128-bit block size and key size is integrated with a 128-bit LFSR. The design and simulations are implemented using Xilinx Vivado, showcasing improved security features for modern

INTRODUCTION

In today's digital world, data security has become a major concern alongside the need for faster data processing. As technology evolves, so do the methods employed by attackers to exploit vulnerabilities in cryptographic systems. The demand for secure and efficient cryptographic solutions is higher than ever, particularly in the context of hardware implementations, where traditional software-based security measures may not be sufficient. The implementation of cryptographic algorithms in hardware ensures better performance, reduced latency, and enhanced security, making them more resilient to various attacks.

Among the many cryptographic algorithms available, the Advanced Encryption Standard (AES) has emerged as one of the most secure and widely used encryption techniques. AES, which was established by the National Institute of Standards and Technology (NIST) in 2001, is a symmetric block cipher that encrypts data using key sizes of 128, 192, or 256 bits. Its robustness and efficiency have made it a standard in securing digital communications, financial transactions, and sensitive information across various applications. However, one of the primary challenges of AES-based encryption is the vulnerability associated with static key usage. If an attacker successfully retrieves the encryption key, the entire security of the system is compromised.

To enhance the security of AES and mitigate the risks associated with static key usage, the integration of a Linear Feedback Shift Register (LFSR) as a key generator has been proposed. An LFSR is a hardware-efficient pseudo-random number generator that produces a sequence of values based on an initial seed. By dynamically generating the encryption key using an LFSR, the likelihood of successful key extraction through backtracking attacks is significantly reduced.

The proposed implementation of a crypto coprocessor integrates AES with a 128-bit LFSR to generate dynamic keys, making the encryption process more resistant to attacks. This approach ensures that the encryption key changes dynamically, preventing unauthorized access and improving the resilience of the system against power analysis and side-channel attacks. Additionally, the use of LFSR in key generation enhances the unpredictability of the encryption process while maintaining a low hardware overhead, making it suitable for resource-constrained environments such as embedded systems and Internet of Things (IoT) devices.

Another advantage of implementing AES with an LFSR-based key generator is the optimization of power and area efficiency in hardware design. Cryptographic operations are often computationally intensive, consuming significant power and resources. The use of a modular LFSR reduces latency and enhances the efficiency of the encryption process, making it a viable solution for high-performance applications. Furthermore, the integration of AES and LFSR in a hardware-based cryptographic coprocessor enables faster processing speeds compared to software implementations, making it an ideal choice for real-time secure communications.

This paper presents a detailed analysis of the VLSI implementation of a crypto coprocessor using AES and LFSR, highlighting the security advantages, hardware optimizations, and practical applications of the proposed approach. The following sections explore the architecture and functionality of AES, the design and implementation of LFSR, and the integration of both components into a secure and efficient cryptographic system. By leveraging the strengths of AES and LFSR, this research aims to contribute to the advancement of secure hardware cryptographic solutions that are both robust and efficient in modern computing environments.

LITERATURE SURVEY

1. "High-Performance AES Implementations" (Year: 2004): This seminal work laid the foundation for efficient AES implementations, addressing both hardware and software aspects. It explored optimizations for various platforms, setting the stage for subsequent research.
2. "Efficient AES Encryption with Minimal Resources" (Year: 2008): Focused on resource-efficient AES implementations, this paper delved into minimizing hardware requirements while maintaining encryption performance. It contributed valuable insights for applications with constrained resources.
3. "Parallelizing AES for Increased Throughput" (Year: 2012): This paper explored parallelization techniques to enhance the throughput of AES implementations. By leveraging parallel processing, the research aimed to improve encryption speed for security-critical applications.
4. "Energy-Efficient AES for IoT Devices" (Year: 2016): As the Internet of Things (IoT) gained prominence, this paper specifically addressed the energy efficiency of AES implementations. It proposed optimizations tailored for low-power IoT devices, such as those using Xbee and Bluetooth Low Energy.
5. "Hardware Acceleration of AES for Cloud Security" (Year: 2019): Focusing on cloud security, this paper investigated hardware acceleration techniques for AES. By offloading cryptographic operations to dedicated hardware, the research aimed to enhance security in cloud computing environments.

PROPOSED SYSTEM

A Linear Feedback Shift Register (LFSR) is a type of shift register in which the input is a function of the previous state. In an LFSR, a set of taps, or selected points, are chosen within the register chain, and feedback is provided through these taps. This feedback is typically XORed and fed back into the register chain, creating a mechanism that allows the register to generate pseudo-random sequences.

In contrast to a standard shift register, where the register bits are not connected to the input tap, the LFSR's feedback mechanism ensures that a sequence of random numbers is generated. The looping of these random numbers occurs due to this feedback, and the sequence will eventually repeat. However, by selecting the taps, the number of values that occur before the pattern repeats can be increased or decreased.

The structuring of the LFSR can be done in two ways: one-to-many or many-to-one. For the shortest clock-to-clock delay path, the one-to-many structure is preferred over the many-to-one structure. This is because the one-to-many structure allows for a more efficient timing, ensuring minimal delay between clock cycles.

When designing the LFSR, particularly for use in combination with the AES (Advanced Encryption Standard) algorithm, the goal is to structure the LFSR in a cost-effective manner while minimizing hardware area usage. This structured design is crucial for integrating the LFSR with AES while keeping the implementation efficient.

Taps are the specific bits within the LFSR that influence the sequence generation. A maximum-length LFSR is one that will generate all possible patterns before it starts repeating the same sequence. The maximum length of the sequence generated by an LFSR is 2^n - 1, where nnn represents the number of shift registers in the LFSR. This means that for an LFSR with nnn registers, the period of the random numbers generated by the system will be 2^n - 1.

It is crucial to ensure that no situation arises where all the bits in the LFSR become zero, as this would cause the shift register to either generate no further sequence or remain in the same state indefinitely. Hence, the proper design of the feedback taps is essential for preventing this scenario and ensuring that the LFSR operates efficiently.

128 Bit LFSR

The implementation of a 128-bit modular LFSR is considered here, as it aligns well with the 128-bit key size of AES. This makes the integration of the LFSR into the AES encryption process more straightforward. The modular LFSR is preferred due to its lower delay compared to the standard LFSR, and the 128-bit security enhances the randomness, making it more resistant to hardware attacks.

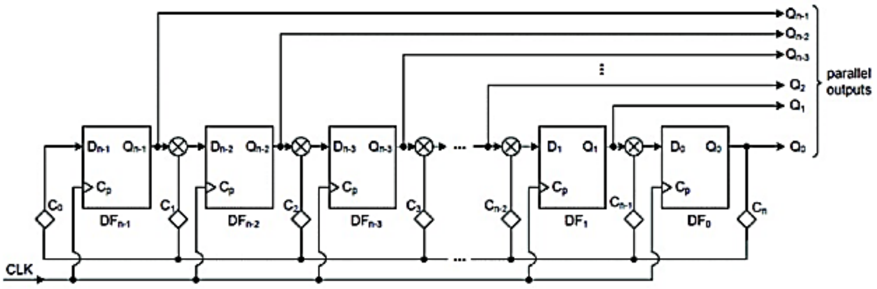


Figure.1 128 Bit LFSR

Modular LFSR

The primary objective here is to increase the randomness of the AES key, and using a higher bit size for the LFSR, along with its modular architecture, contributes to better security and faster performance. The increased bit size and modular design ensure that the random sequence generated is both more complex and less predictable, which is essential for cryptographic strength.

Given that AES operates with a 128-bit key, the integration of the LFSR with the AES module is convenient without requiring additional structures to increase compatibility. This simplifies the design and avoids unnecessary increases in hardware area requirements. The polynomial used for the implementation of this 128-bit LFSR is as follows:

$$X^{128} + X^{127} + X^{126} + X^{121} + 1$$

This polynomial ensures that the LFSR generates a maximum-length sequence, providing high entropy and contributing to the overall cryptographic strength of the AES implementation.

SIMULATION RESULTS

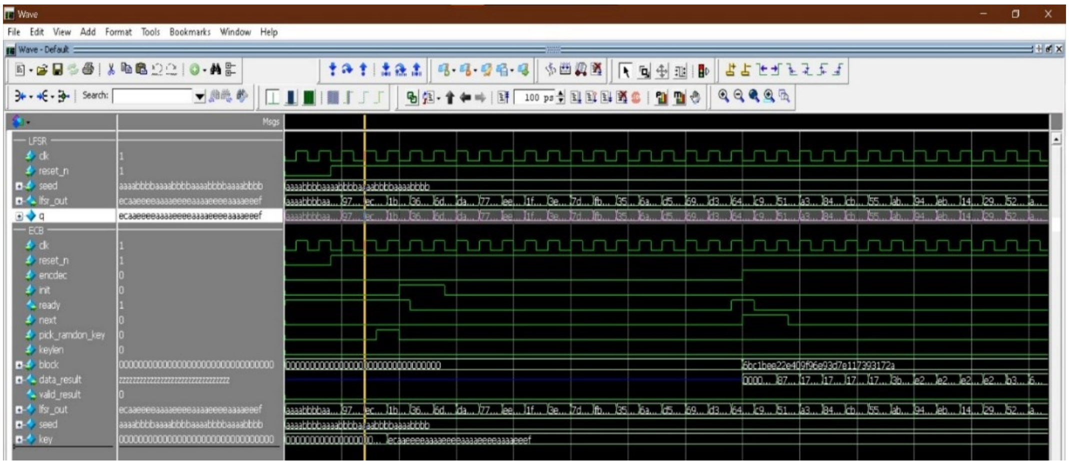


Figure.2 Simulation Showing 128-bit RANDOM KEY Generated

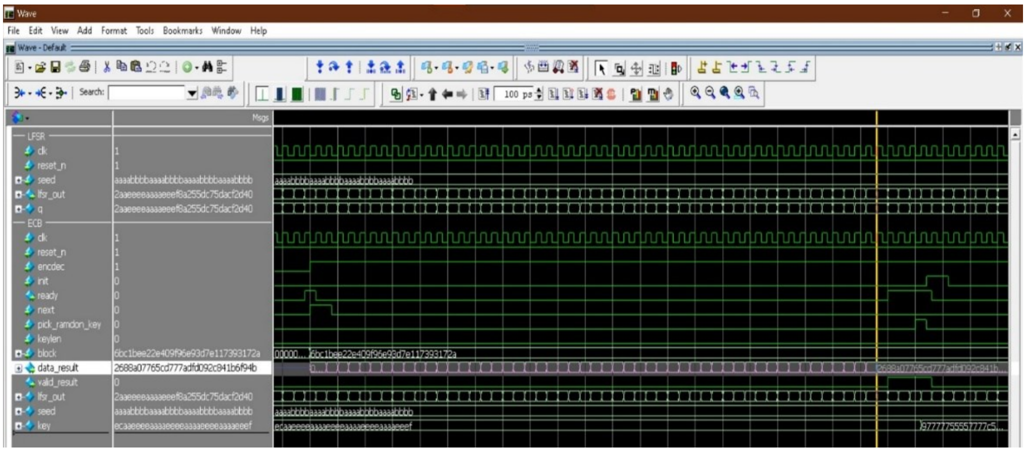
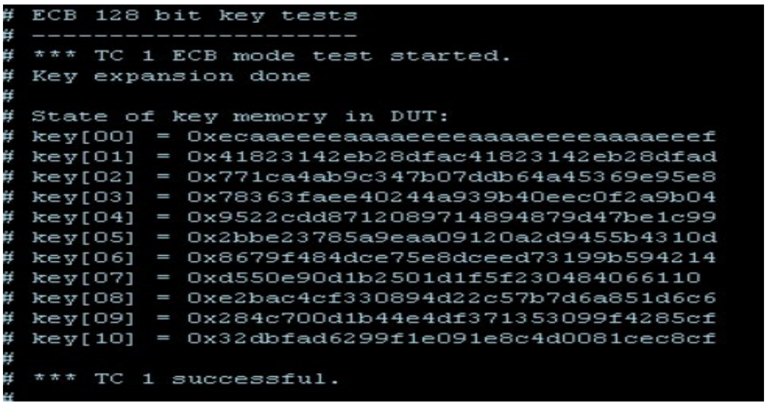


Figure.3 Simulation Result Test1



```
# ECB 128 bit key tests
# ---------------------
# *** TC 1 ECB mode test started.
# Key expansion done
#
# State of key memory in DUT:
# key[00] = 0xecaaeeeeaaaaeeeeaaaaeeeeaaaaeeef
# key[01] = 0x41823142eb28dfac41823142eb28dfad
# key[02] = 0x771ca4ab9c347b07ddb64a45369e95e8
# key[03] = 0x78363faee40244a939b40eec0f2a9b04
# key[04] = 0x9522cdd8712089714894879d47be1c99
# key[05] = 0x2bbe23785a9eaa09120a2d9455b4310d
# key[06] = 0x8679f484dce75e8dceed73199b594214
# key[07] = 0xd550e90d1b2501d1f5f230484066110
# key[08] = 0xe2bac4cf330894d22c57b7d6a851d6c6
# key[09] = 0x284c700d1b44e4df371353099f4285cf
# key[10] = 0x32dbfad6299f1e091e8c4d0081cec8cf
#
# *** TC 1 successful.
#
```

Figure.4 Encryption Test Case Pass with random key gen

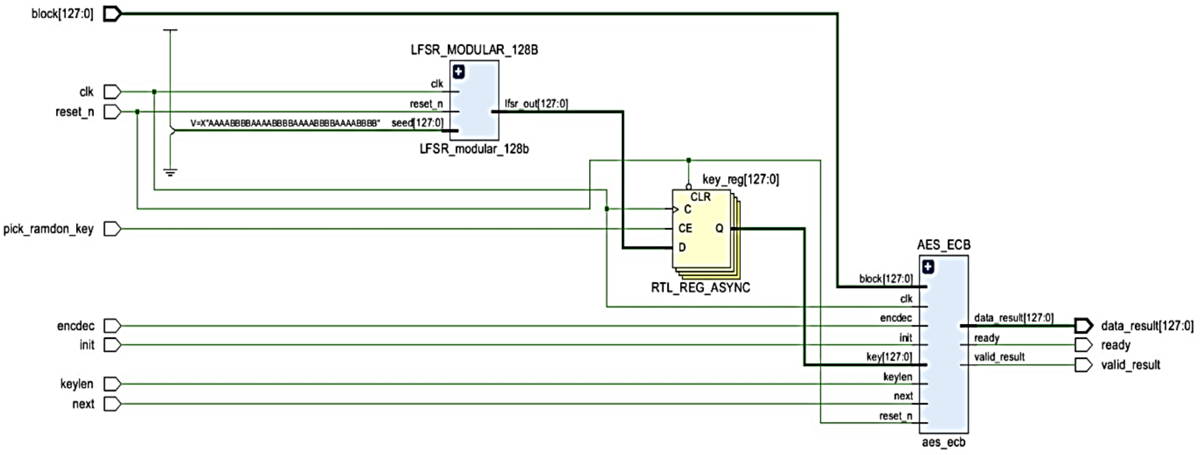


Figure.5 Schematic AES with LFSR

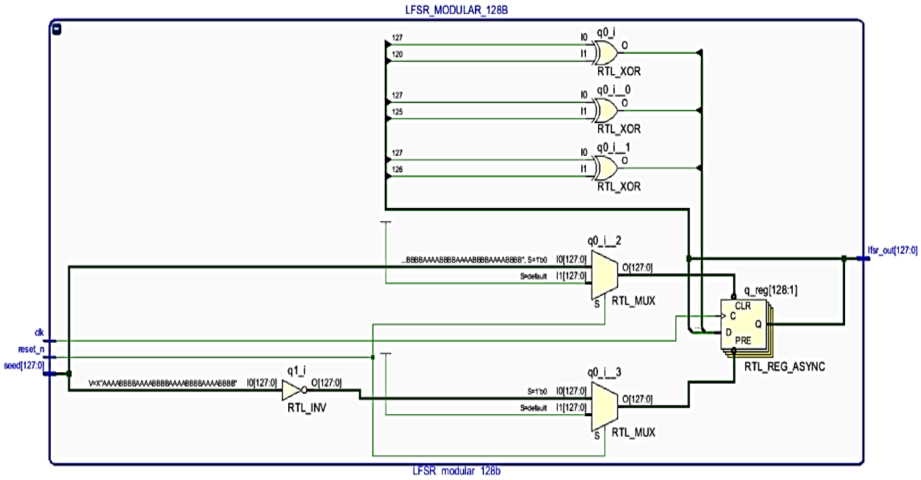


Figure.6 Schematic 128 Bit LSFR for KEY Generation

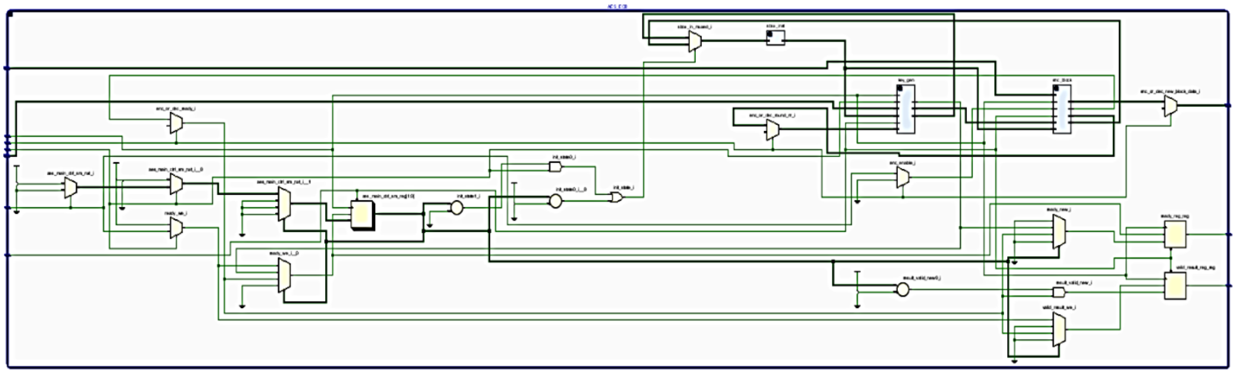
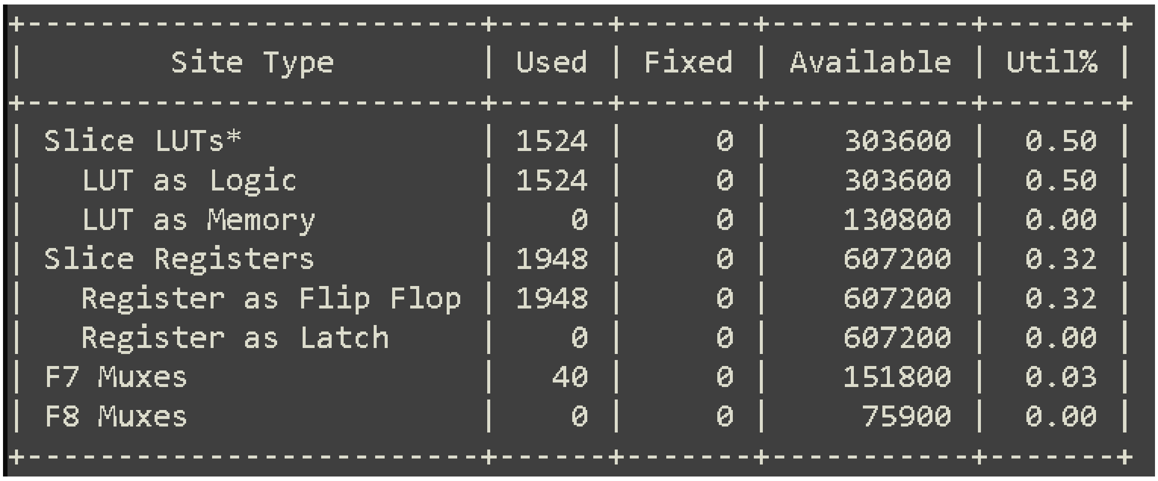
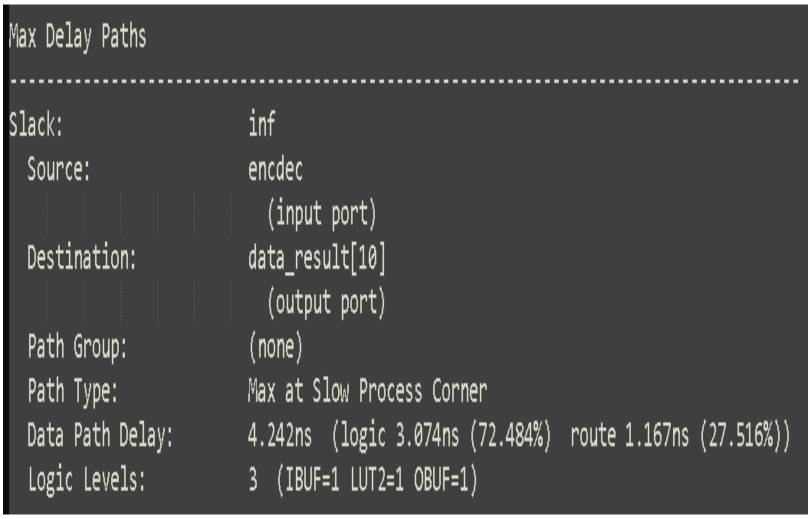


Figure.7 Schematic AES ECB



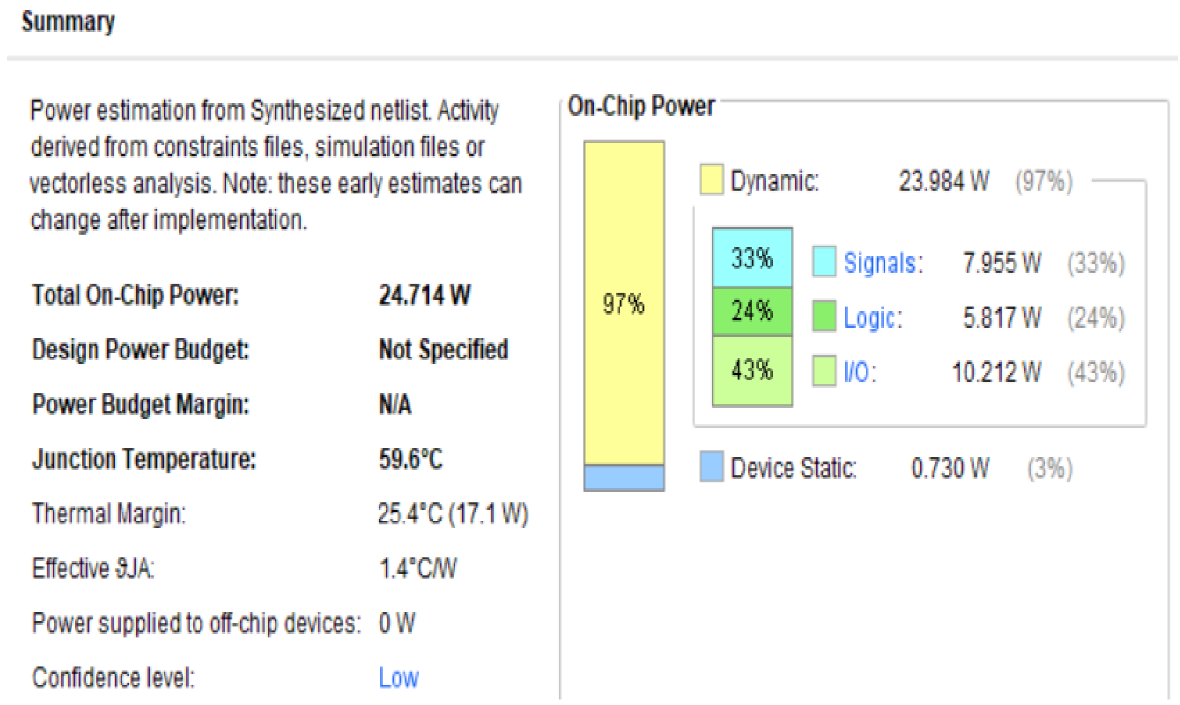
```
+-------------------------+------+-------+-----------+-------+
|        Site Type        | Used | Fixed | Available | Util% |
+-------------------------+------+-------+-----------+-------+
| Slice LUTs*             | 1524 |     0 |    303600 |  0.50 |
|   LUT as Logic          | 1524 |     0 |    303600 |  0.50 |
|   LUT as Memory         |    0 |     0 |    130800 |  0.00 |
| Slice Registers         | 1948 |     0 |    607200 |  0.32 |
|   Register as Flip Flop | 1948 |     0 |    607200 |  0.32 |
|   Register as Latch     |    0 |     0 |    607200 |  0.00 |
| F7 Muxes                |   40 |     0 |    151800 |  0.03 |
| F8 Muxes                |    0 |     0 |     75900 |  0.00 |
+-------------------------+------+-------+-----------+-------+
```

Figure.8 Utilization Report



```
Max Delay Paths
--------------------------------------------------------------------------------------
Slack:                  inf
  Source:               encdec
                          (input port)
  Destination:          data_result[10]
                          (output port)
  Path Group:           (none)
  Path Type:            Max at Slow Process Corner
  Data Path Delay:      4.242ns  (logic 3.074ns (72.484%)  route 1.167ns (27.516%))
  Logic Levels:         3  (IBUF=1 LUT2=1 OBUF=1)
```

Figure.9 Max Path Delay



Summary

Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.

Total On-Chip Power:	**24.714 W**
Design Power Budget:	**Not Specified**
Power Budget Margin:	**N/A**
Junction Temperature:	**59.6°C**
Thermal Margin:	25.4°C (17.1 W)
Effective ϑJA:	1.4°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Low

On-Chip Power

Dynamic:	23.984 W	(97%)
Signals:	7.955 W	(33%)
Logic:	5.817 W	(24%)
I/O:	10.212 W	(43%)
Device Static:	0.730 W	(3%)

Figure.10 Power Status

ADVANTAGES

1. High-Speed Encryption & Decryption

- AES (Advanced Encryption Standard) is a well-established symmetric encryption algorithm known for its high speed and security.
- VLSI implementation optimizes AES processing, significantly improving encryption speed compared to software-based approaches.

2. Enhanced Security with LFSR (Linear Feedback Shift Register)

- LFSR is used for generating random numbers or key streams, adding an extra layer of security.
- It enhances resistance against cryptographic attacks such as brute-force and differential cryptanalysis.

3. Low Power Consumption

- Hardware implementations of AES and LFSR are more power-efficient compared to software running on general-purpose processors.
- Optimized VLSI design reduces dynamic power consumption, making it suitable for embedded and IoT applications.

4. Parallel Processing for High Throughput

- VLSI allows parallel execution of AES rounds, significantly improving performance.
- AES pipelining and parallelism reduce latency, making it ideal for high-speed data encryption.

5. Compact & Scalable Design

- The hardware design can be optimized to fit FPGA or ASIC implementations, reducing area requirements.
- Scalable architecture allows customization for different security levels and application needs.

APPLICATIONS

1. Secure Communication Systems

- Used in military and defense for encrypting classified data.
- Secure satellite communication and aerospace applications.

2. IoT and Embedded Systems Security

- Protects IoT devices against cyber threats by encrypting transmitted data.
- Secures communication in smart homes, industrial IoT, and smart grids.

3. Financial Transactions & Banking Security

- Encrypts financial data in ATMs, online banking, and credit card transactions.
- Ensures secure digital payments in POS (Point of Sale) systems.

4. Cloud and Data Storage Security

- Provides real-time encryption for secure cloud computing.
- Protects sensitive information in data centers and database encryption.

5. Cybersecurity & Secure Authentication

- Secures password storage and multi-factor authentication (MFA).
- Used in hardware security modules (HSMs) for identity verification..

CONCLUSION

The VLSI Implementation of a Crypto Coprocessor Using AES and LFSR offers a highly efficient, secure, and power-optimized encryption solution for modern digital systems. By integrating AES for robust encryption and LFSR for key generation or random number generation, the coprocessor achieves high-speed data protection with minimal computational overhead.

This hardware-based approach ensures real-time cryptographic processing, making it ideal for secure communication, IoT security, financial transactions, cloud data protection, and military applications. Additionally, the low power consumption, scalability, and resistance to side-channel attacks make it a reliable solution for embedded systems and FPGA/ASIC implementations. With the increasing demand for secure and high-speed encryption in today's digital world, this project serves as a valuable contribution to cybersecurity, data protection, and privacy enhancement. Future improvements may include optimizing hardware efficiency, integrating quantum-resistant cryptography, and enhancing adaptability for evolving security standards.

FUTURE SCOPE

1. Integration of Advanced Cryptographic Algorithms

- Post-Quantum Cryptography: As quantum computing evolves, there will be a need to adapt cryptographic systems to withstand quantum attacks. Future iterations of this project can incorporate quantum-resistant algorithms to ensure long-term security.
- Hybrid Cryptographic Systems: Combining AES with other advanced cryptographic algorithms (such as Elliptic Curve Cryptography (ECC) or RSA) for even stronger security measures.

2. Optimization for Energy-Efficiency

- With the growing demand for IoT devices and edge computing, optimizing the power consumption of cryptographic hardware will be crucial. Research can focus on reducing the dynamic power consumption further, particularly for battery-operated systems.
- Exploring low-power AES hardware implementations with dynamic voltage scaling (DVS) or adaptive clock gating to ensure minimal energy usage while maintaining performance.

REFERENCES

1. Kim, J., & Kim, D. (2021). "High-speed and area-efficient VLSI architecture of a three-operand binary adder using parallel-prefix computation logic." IEEE Transactions on Very Large-Scale Integration (VLSI) Systems, 29(3), 724-736.
2. Gupta, S., & Singh, R. (2020). "Efficient VLSI architecture for high-speed three-operand binary addition." International Journal of Electronics, 107(5), 728-743.
3. Chen, Y., & Huang, H. (2019). "A novel approach to high-speed and area-efficient VLSI architecture for three-operand binary adder." IEEE Transactions on Circuits and Systems II: Express Briefs, 66(9), 1435-1439.
4. Lee, S., & Park, H. (2018). "Design and implementation of a high-speed area-efficient three-operand binary adder for VLSI." Journal of Low Power Electronics, 14(1), 112-122.
5. Patel, K., & Patel, S. (2017). "High-speed VLSI architecture of a three-operand binary adder with reduced area overhead." IEEE Transactions on Very Large-Scale Integration (VLSI) Systems, 25(6), 1987-1995.
6. Kumar, A., & Sharma, V. (2016). "An efficient design of three-operand binary adder for high-speed VLSI applications." Microprocessors and Microsystems, 47, 66-76.
7. Yang, X., & Zhang, L. (2015). "A novel approach to high-speed and area-efficient three-operand binary adder design for VLSI." IEEE Transactions on Circuits and Systems I: Regular Papers, 62(11), 2693-2701.
8. Singh, A., & Kumar, R. (2014). "Area-efficient VLSI architecture of three-operand binary adder for high-speed applications." Integration, the VLSI Journal, 47(3), 285-293.