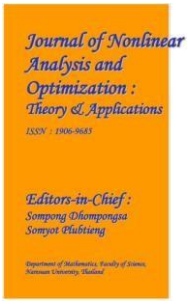
Journal of **N**onlinear **A**nalysis and **O**ptimization
Vol. 15, Issue. 1 : 2024
ISSN : **1906-9685**



A HIGHLY SECURE FPGA-BASED DUAL-HIDING ASYNCHRONOUS-LOGIC AES ACCELERATOR AGAINST SIDE-CHANNEL ATTACKS

SYED USMAN, M.SAHITHYA, K. CHARISHMA SAI, L. LAHARI, M. AJAY, Department of Electronics & Communication Engineering, NRI Institute of Technology, Pothavarappadu (V), Agiripalli (M), Eluru (Dt)-521212

Abstract: Data security is crucial in applications such as e-commerce, internet banking, military, satellite communication, wireless communication, and signal and digital image processing. Cryptography plays a vital role in ensuring data confidentiality by transforming data into an unreadable format for unauthorized users. This technique provides a reliable and cost-effective way to protect data and verify data integrity. However, cryptography is vulnerable to side-channel attacks, which can compromise data security and integrity. To address this issue, a novel AES accelerator is proposed. It enhances vertical (amplitude) side-channel attack (SCA) resistance through an efficient dual-rail mapping approach and a zero-value (ZV) compensated substitution-box (S-Box). Additionally, it improves horizontal (temporal) SCA resistance by employing a timing-boundary-free input arrival time randomizer and a skew-delay controller. This design reduces area overhead and provides robust protection against side-channel attacks. The proposed design is implemented using Xilinx ISE 14.7 and validated through simulation results.
Keywords: - Advanced encryption standard (AES), asynchronous logic design, delay randomizer, dualhiding, side-channel attack (SCA), Data security, Cryptography.

1. Introduction

The AES algorithm is implemented using four operations: Sub Bytes, Shift Rows, Mix Columns, and Add Round Key. The architecture of the 256-bit AES algorithm consists of 14 rounds for encryption and 14 rounds for decryption. After encryption, the cipher text is transmitted across the channel and decrypted by the receiver using the same key used for encryption. In the 128-bit AES algorithm, the key size is 128 bits, and all data sizes are also 128 bits, including the message to be encrypted, the cipher text, and the decrypted message. The internal data structure of 128-bit data involves using a 4x4 matrix, where each element of the matrix is 8 bits. Since all operations are performed on a column basis, the 128-bit data is converted into a 4x4 matrix with each element being 8 bits. The 128-bit AES encryption block is implemented in 14 rounds, each consisting of Add Round Key, Sub Bytes, Shift Rows, and Mix Columns. Round 0 only involves the Add Round Key operation, while Round 14 consists of Sub Bytes, Shift Rows, and Add Round Key operations, requiring 3 clock cycles. Rounds 1 to 13 include all four operations, with each operation occurring in a distinct clock cycle. Therefore, the same hardware can be used for all 14 rounds once it has been implemented for Add Round Key, Sub Bytes, Shift Rows, and Mix Columns. The AES algorithm is a serial process, where the output of the first round serves as the input to the second round, allowing for the use of the same hardware for each round. This paper proposes a secure FPGA architecture for implementing cryptographic devices, focusing on flexibility and tamper-resistance in embedded systems design. Unlike ASICs, which have high costs and long design times, FPGAs offer a more adaptable and cost-effective solution for security

applications. The FPGA design is based on an asynchronous methodology and is resilient to various side-channel attacks, including Power and Fault Attacks. The implementation of the AES algorithm demonstrates the inherent resistance of the SCAR-FPGA to side-channel analysis (SCA). SCA is a method used to deduce cryptographic keys by analyzing power consumption patterns, making hardware countermeasures crucial to protect cryptographic circuits. The proposed SCA-resistant methodology involves a machine learning trained power compensation module that adjusts the probability of hamming distance (HD) of the intermediate data. This adjustment makes it difficult to distinguish between correct and incorrect sub-keys, thereby enhancing resistance to SCA. With neural dynamic programming, the machine learning algorithm determines the optimal HD redistribution mapping. Experimental SCA results demonstrate that the AES-128 encryption algorithm circuit, implemented on a Xilinx Spartan-6 FPGA placed on a SAKURA-G board, can give more than 200 times measures to disclosure and still shows no indication of disclosing the advanced encryption standard (AES) sub-key. Furthermore, it has zero frequency overhead, minimal power and area overhead, and is suitable for hardware implementation of SCA countermeasure. This research presents an embedded solution for hardware trojan (HT) and counterfeit detection. The suggested approach is based on the fingerprinting of the static distribution of the supply voltage (Vdd) throughout the whole surface of an integrated circuit, taking into account that HTs are invariably added to production batches rather than to a single device. An array of sensors sensitive to the local Vdd value is used to measure this fingerprint, and a novel variation model of CMOS logic performance is the basis for fingerprint extraction. This model considers the impact of design (layout, supply route, etc.) in addition to process changes. This study presents an adaptive distinguisher in addition to the fingerprinting method to address the challenging issue of p-value fixing on huge sets of statistical tests. On a set of 24 FPGA boards, the effectiveness of the entire detection process is empirically shown. With the volume of personal information saved in the digital domain increasing at an unprecedented rate, security and privacy on modern computing systems have emerged as a critical design parameter. One of the main risks to these devices' security and privacy has been shown to be side-channel assaults. Thus, comprehending the fundamentals of side-channel attacks has emerged as a crucial area of research. This paper proposes an effective preprocessing method for an attack scenario where physical leakage (PL) collection time and device access time are constrained. By improving the quality of the leakage signal, the suggested preprocessing method makes use of several side-channel distinguishers to reduce the number of PL measurements needed for a given success rate. This work combines mutual information and Pearson correlation, two widely utilised distinguishers. For the first time, the preprocessing and attack processes perform better when coupled distinguishers are applied. For both masked and unmasked advanced encryption standard (AES), the success rate of the suggested attack framework is 30% higher than that of the traditional single distinguisher side-channel attacks, by 33% and 33%, respectively. In order to create a highly efficient yet reliable Side Channel Attack (SCA), we present a Profiling through Relevance-Learning (PRL) technique on Physical Leakage Information (PLI) to extract highly correlated PLI with processed data. Our suggested PRL consists of four essential components. To find the boundaries of the clusters and the objects within them, variance analysis on PLI is first used. Secondly, the low variance sampling points of PLI measurements (traces) are discarded and the high variance sampling points of PLI measurements are clustered using the nearestneighbor k-NN variance clustering technique to minimise the sample points of PLI. Relevant leaking information about the secret key is contained in these clustered sampling points, which have a strong correlation with the processed data. Third, there are multiple nearby sampling locations where the information linked to the secret key is dispersed, with varying degrees of leakage. To quantify the level of leakage related to the secret key, we analytically estimate the Key-leakage relevance factor for each clustered sample point. Fourth, depending on the values of the relevance factor and the traces of the sample points, a weight proportional to the Key-leakage relevance factor is updated repeatedly using Hebbian learning. To improve the correlation between the PLI and the processed data, the converged weights assigned to the clustered sample points are connected to the corresponding PLI. As a result, it is possible to drastically lower the number of PLI measurements needed to disclose the secret key. Furthermore, we demonstrate analytically that our suggested PRL has an O(n)

computational difficulty, in contrast to the O(n2) and O(n3) computational complexities of the profiling techniques that have been disclosed. Based on our proposed PRL's tests conducted on the PLI of the AES-128 method, the findings show that k-NN variance clustering reduces the PLI's sample points by 87%. In just 538 iterations, the converged weight with learning error rate <1% is reached. We test the robustness of our proposed PRL using four distinct frequencies (corresponding to four noise levels), a variety of previously published profiling approaches, and two concealing countermeasures (horizontal and vertical hidings) applied to the PLI. At four different frequencies, our suggested PRL successfully reduces 94.53 to 98.19 percent of traces. The weights converge at 7,517 iterations based on the hiding countermeasures applied to the PLI, and the SCA only needs 523 traces to disclose the secret key. In comparison to documented methods that necessitate over 106 traces, our suggested PRL is approximately 2,000 times more effective at executing SCA. We describe an asynchronous-logic (asynchronous) accelerator for the Advanced Encryption conventional (AES) that is immune to sidechannel attacks (SCAs) and is based on conventional library cells. We implement the dual-rail logic to reduce SCA, and we suggest the data flow control to stop the reset operation at the last round, as well as a delayed completion tree to introduce delay variations. Using power simulations, we further conduct a thorough SCA evaluation (using 7 attacking/power models). To the best of our knowledge, no other asynchronous AES or its sub-block designs have ever been the subject of a thorough SCA examination. We demonstrate the unbreak ability of our proposed asynchronous AES accelerator using 5k power simulations. Our suggested asynchronous AES accelerator dissipates 2nJ/encryption @ 1.2V and measures 420µm×420µm @ 65nm CMOS [1-6].

2. Proposed Method

The block design of our dual-hiding asynchronous logic AES accelerator is shown in Figure 1. It consists of an asynchronous logic core, a preround TBF input arrival-time randomizer, input–output flip-flops with sync-logic state machines, and a single-to dual-rail conversion module. The syncasynchronous interface circuit is made up of input-output flip-flops with sync-logic state machines that store the inputs (plaintext and keys) and outputs (ciphertext). The primary functions of the preround TBF input arrival-time randomizer are to safeguard the preround operation from SCA and to randomly start the dual-rail asynchronous-logic operation. The primary AES computing engine is the asynchronous-logic core, which is composed of multiple building parts, including dual-rail core modules (S-Box, Shift-Row, MixColumn, etc.) and rings of asynchronous-logic pipeline [latches with competition detection (CD)].

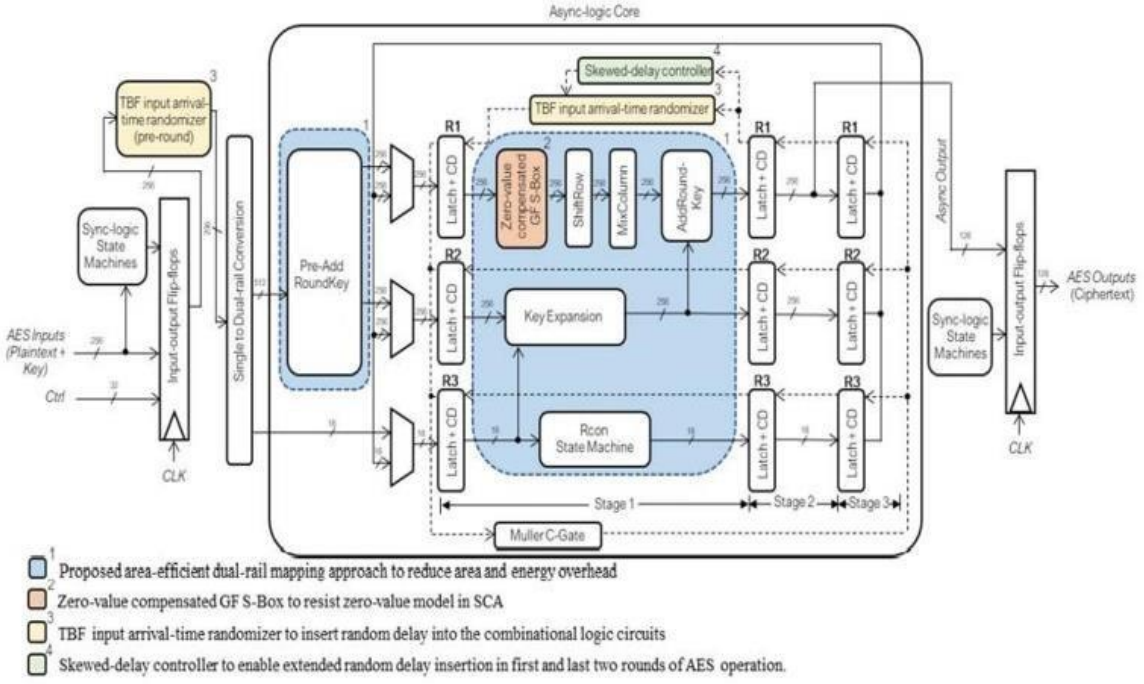


Figure.1. Block diagram of our dual-hiding asynchronous-logic AES accelerator

We first describe the encryption process of our proposed asynchronous-logic AES accelerator. Upon receiving the AES inputs, the input–output flip-flops pass the inputs into the asynchronous-logic core via the TBF input arrival-time randomizer and the single- to dual-rail conversion module. The TBF input arrival-time randomizer is our proposed countermeasure to protect against FR SCA attacks (see technical discussion later). The single- to dual-rail conversion module converts the single-rail data of the AES inputs into their corresponding dual-rail data.In the asynchronous-logic core, the dual-rail data first undergo a pre-round key addition (Pre- AddRoundKey), followed by ten rounds of AES transformation in rings of asynchronous-logic pipelines. Each ring of the asynchronous-logic pipelines is composed of three latches with CD. Lack serves as the handshake signal to control the latch, allowing either valid data (for evaluation) or null data (for reset) to pass through. When the Lack signal is asserted (logic "1"), the latch will wait for valid data, and once the valid data have arrived, the latch will hold the valid data. When the Lack signal is negated (logic "0"), the latch will wait for null data, and once the null data have arrived, the latch will hold the null data. Each valid–null data sequence constitutes one round of AES transformation. There are three rings of asynchronous-logic pipeline, R1–R3 rings (i.e., latches labeled with R1–R3, respectively, in Fig.4.2). The R1 ring performs AES round transformation (S-Box, ShiftRow, MixColumn, and AddRoundKey), the R2 ring performs key expansion, and the R3 ring controls ten rounds of AES transformation via Rcon state machine.When the encryption is started, the latches in Stage 1 will first receive the data from the Pre-AddRoundKey module via the multiplexers (MUXes). Based on the successful receival of the data by the latches in Stage 1, the MUXes will be switched to establish three rings, R1–R3. For each ring, we need a minimum of three stages (i.e., Stages 1–3 in Fig.4.2) to form a complete asynchronous cyclic data propagation [41].Each ring is formed by passing the data from Stage 1 to Stage 2, Stage 2 to Stage 3, and Stage 3 back to Stage 1 again. The dotted lines in Fig.4.2 represent the handshake Lack signals of the latches (in each ring), thus controlling the propagation of the valid and null data in Stages 1–3. The Muller C-Gate synchronizes all the Lack signals (from the latches of R1–R3) in Stage 1.

This synchronization in Stage 1 ensures that the AES data, AES key, and Rcon are all updated/stored in the latches in Stage 1 before a new round of AES transformation is initiated. Ten rounds of asynchronous-logic AES transformation are performed with the valid and null data cycling through Stages 1–3.After the tenth round, the Rcon state machine stops the asynchronous-logic operation, and the ciphertext is stored in the R1 latch in Stage 2, ready to be retrieved by the input–output flip-flops. The sync-logic state machines control the flip-flops to fetch the data stored in the latches after ten clock cycles. Hence, conceptually, our asynchronous-logic AES accelerator is globally synchronous, locally asynchronous (GSLA), with the assumption that the AES encryption will be finished within ten clock cycles.

2.1Area-Efficient Dual-Rail Mapping Approach

The proposed approach integrates dual-rail cells with fewer than five inputs into a single 6-input LUT, such as two-input AND, OR, NAND, NOR, XOR, and XNOR gates, resulting in a more area-efficient mapping approach that requires just one LUT.

2.2 ZV Compensated GF S-Box

The hardware implementation of S-Boxes is crucial for reducing area and power consumption. While a Galois Field (GF) S-Box is area-efficient, it is susceptible to Zero Value (ZV) Side-Channel Analysis (SCA) attacks. To address this issue, a ZV-compensated GF S-Box is proposed. This approach includes two MUXes: the first MUX detects if the S-Box input is zero and passes a dummy data value, p, if true; the second MUX waits for the dummy data to pass through until it detects q in the output of GF(24), where q is the expected output of GF(24) when the S-Box input is p. For the proposed ZVcompensated S-Box, p is chosen as "8" and q as "1." Once q is detected, the second MUX returns the correct output, zero, to the subsequent circuits.

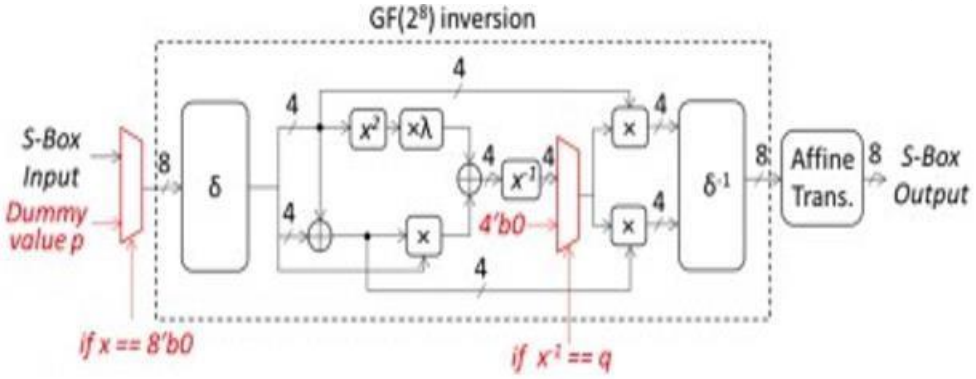


Figure.2. Proposed ZV compensated GF S-Box

2.3 TBF Input Arrival-Time Randomizer

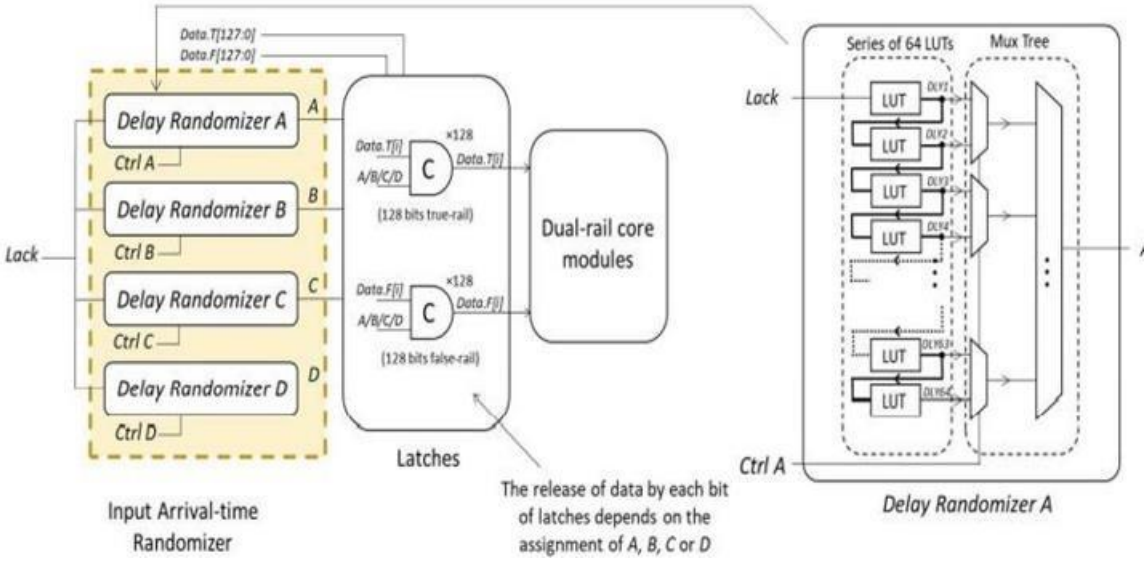


Figure.3. Block diagram of our proposed TBF input arrival-time randomizer. (b) Internal circuitries of a delay randomizer.

To increase the horizontal hiding feature, we propose a TBF input arrival-time randomizer to insert random delays to the local handshake signals (Lack) which control the "release" of the data into dualrail core modules. Fig. 3 depicts our proposed TBF input arrival-time randomizer which comprises four delay randomizers (i.e., delay randomizers A, B, C, and D). Each delay randomizer is made up of a series of 64 LUTs and of a MUX tree allowing the inserted delay to vary from 1 to 64 unit- delays. The Ctrl signal is a primary input, serving as a random signal to our proposed asynchronous-logic AES accelerator that defines the number of unit delays to be incurred. For simplicity and for the purpose of side-channel attack (SCA) evaluation, we generate the Ctrl signal using plaintext via external XOR operations for the first encryption, and based on the previous ciphertext via external XOR operations for subsequent encryptions, to achieve randomness. The delay randomizers A-D are connected to the latches, manipulating the input arrival time of the data to the dual-rail core modules. In our asynchronous-logic design, each bit of the latch can release data independently, meaning each bit of data may have a different input arrival time. With four delay randomizers in our proposed TBF input arrival-time randomizer, we can group the latches (128-bits true-rail + 128-bits false-rail) into four groups by assigning the outputs of each delay randomizer to particular latches through wire

connections. Such an assignment is denoted as the configuration of the TBF input arrival-time randomizer.

2.4 Skewed-Delay Controller

In general, side-channel attack (SCA) evaluations only apply to the first round (FR) or last round (LR) operations. Therefore, applying wide delay variations to all the rounds is not efficient. We propose to increase the delay variations only in the first and last two rounds. The skewed-delay controller comprises a state machine and two MUXes. The first MUX, known as the stabilizing MUX, stabilizes all the internal nodes in the series of LUTs and the MUX tree. The second MUX, known as the bypass MUX, allows the handshake signal (Lack) to be bypassed without incurring input delay during the second to eighth rounds of AES operation. The state machine counts the number of handshake signals, which corresponds to the number of rounds of AES operation.

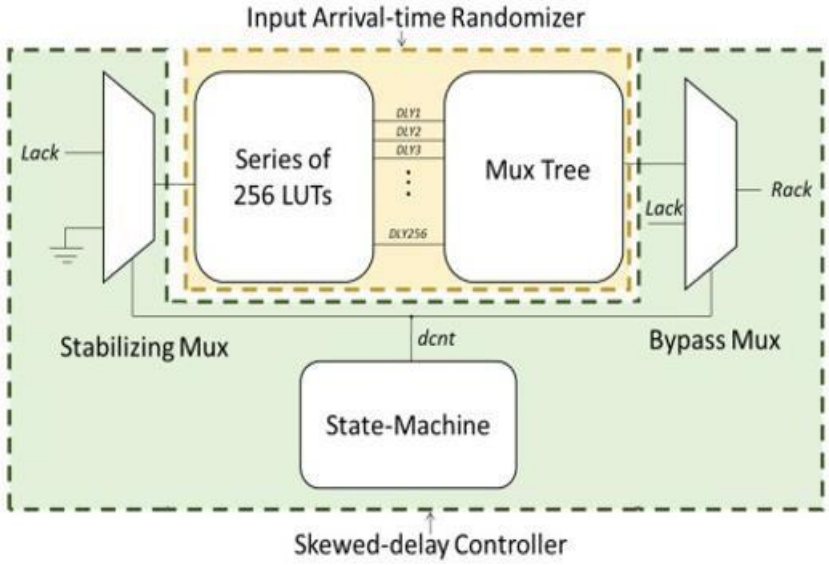


Figure.4. Block diagram of our proposed skewed-delay controller

3. Results and Discussion

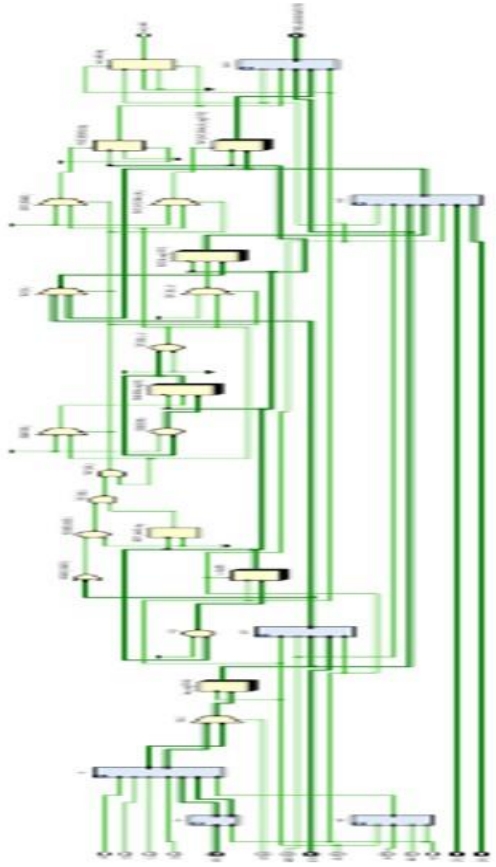


Figure.5. RTL schematic

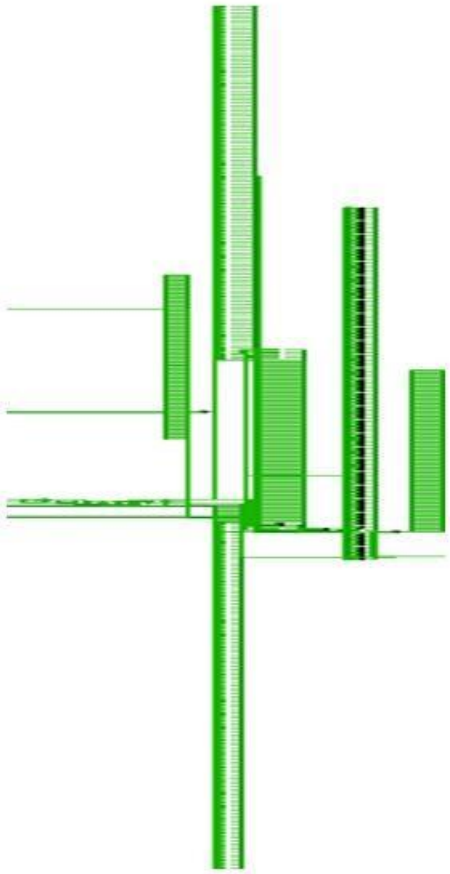
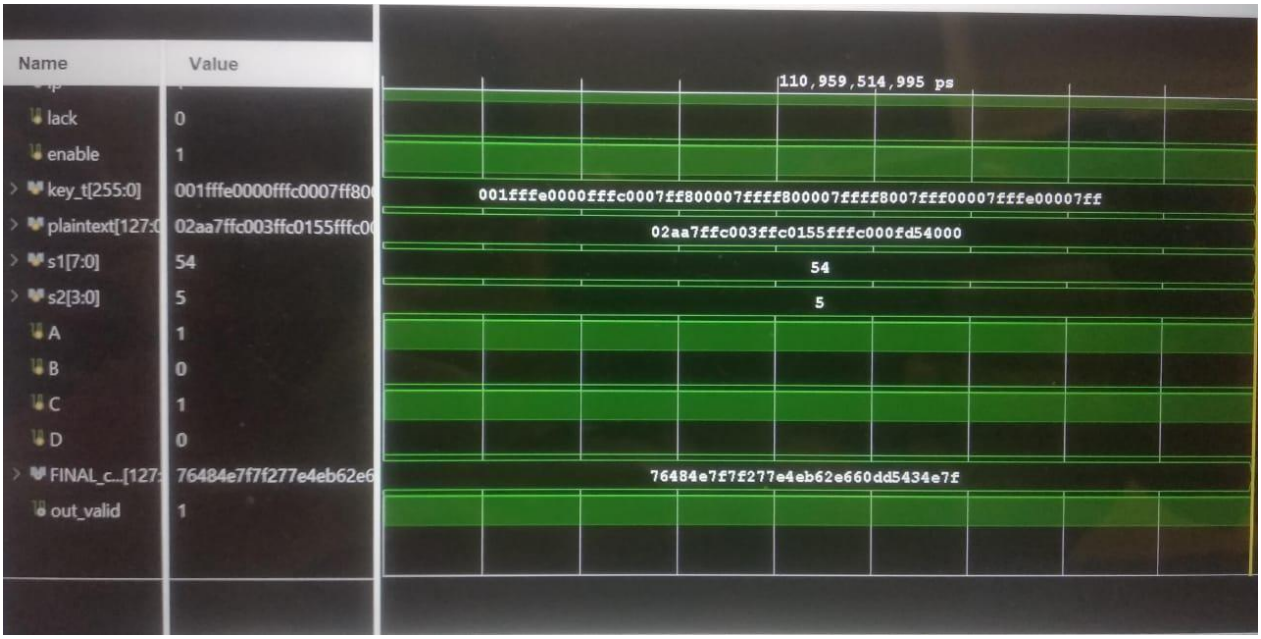
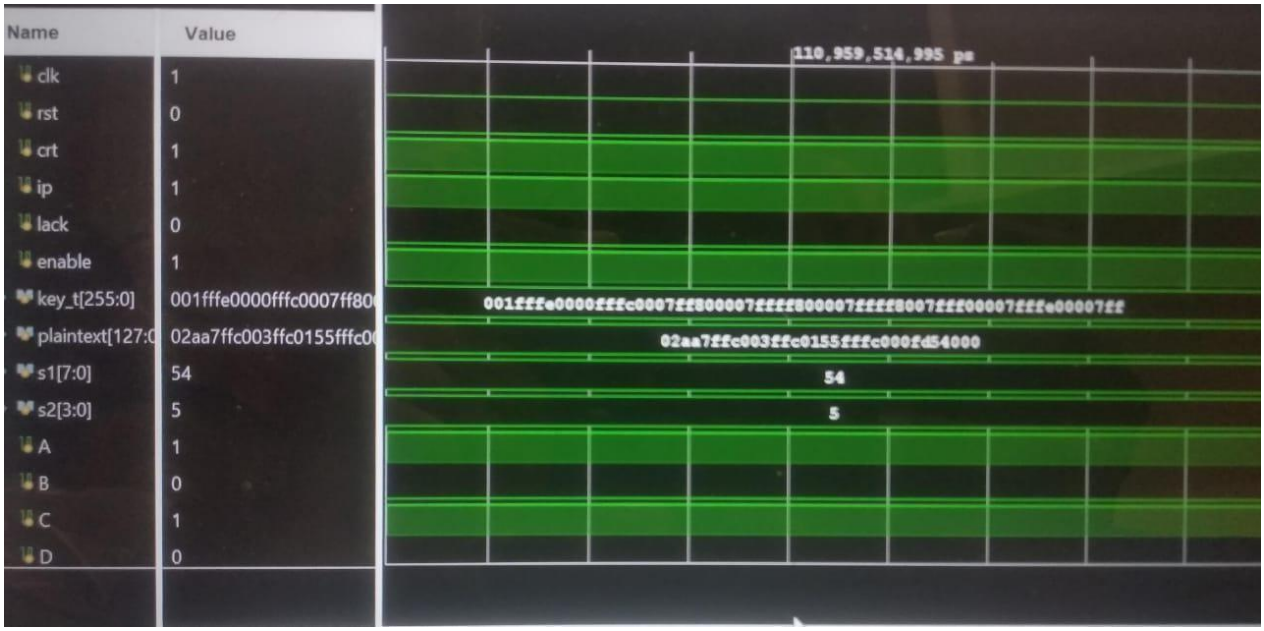


Figure.6. Technology schematic

3.2 Simulation results





3.3 Evaluation table for Area, Delay:

	Area (LUT's)	**Delay (ns)**
Proposed	2914	8.248

4. Conclusion

We proposed an asynchronous-logic AES accelerator for secure FPGAs, employing "by-design" approaches to achieve fine-grain dual-hiding (horizontal and vertical) for mitigating side-channel attacks (SCA) while maintaining low area and energy overheads. Our accelerator incorporates an areaefficient dual-rail mapping approach, ZV compensated S-Box, TBF input arrival-time randomizer, and skewed-delay controller to enable low overhead-for-security design.We extensively evaluated our TBF input arrival-time randomizer with various configurations and selected the most secure configuration for our asynchronous-logic AES accelerator. The evaluation results demonstrate that our accelerator can withstand SCA with 20 million traces, making it the most secure FPGA-based design, with area and energy overheads of 4.3× and 1.5×, respectively. Our proposed design achieved a figure

of merit (Area × Energy/MTD(All)×106) of only 0.3, which is 403× smaller than the sync-logic WDDL and 95× smaller than the reported asynchronous-logic design.

Reference

1. A. Mokari, B. Ghavami and H. Pedram, "SCAR-FPGA : A novel side-channel attack resistant fpga," 2009 5th Southern Conference on Programmable Logic (SPL), 2009, pp. 177-182
2. Shan, Weiwei & Zhang, Shuai & He, Yunkun. (2017). Machine Learning based Side-ChannelAttack Countermeasure with Hamming-distance Redistribution and its application on AES. Electronics Letters. 53. 10.1049/el.2017.1460.
3. M. Lecomte, J. Fournier and P. Maurine, "An On-Chip Technique to Detect Hardware Trojans and Assist Counterfeit Identification," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 25, no. 12, pp. 3317-3330, Dec. 2017.
4. S. Seçkiner and S. Köse, "Preprocessing of the Physical Leakage Information to Combine SideChannel Distinguishers," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 29, no. 12, pp. 2052-2063, Dec. 2021.
5. A. A. Pammu, K. -S. Chong, Y. Wang and B. -H. Gwee, "A Highly Efficient Side Channel Attack with Profiling through Relevance-Learning on Physical Leakage Information," in IEEE Transactions on Dependable and Secure Computing, vol. 16, no. 3, pp. 376-387, 1 May-June 2019.
6. K. -S. Chong et al., "Side-Channel-Attack Resistant Dual-Rail Asynchronous-Logic AES Accelerator Based on Standard Library Cells," 2019 Asian Hardware Oriented Security and Trust Symposium (AsianHOST), 2019, pp. 1-7.