

FPGA-based Power & EM Side-Channel Cryptanalysis on GIFT-128 and Development of secure Trojan-free physical design for GIFT-128 with minimized transistors

SRIRAMUDU^{1,2,3}, NAGENDAR YERUKALA¹, G. PADMAVATHI¹, NEELIMA GUNTUPALLI²

¹C R Rao AIMSCS, UoH Campus, Gachibowli,
Hyderabad, Telangana,
INDIA

²Department of Computer Science and Engineering,
Acharya Nagarjuna University, Guntur, Andhra Pradesh,
INDIA

³TigOs Secure Hardware Solution-C3ihub,
Hyderabad,
INDIA

Abstract: - The rapidly evolving field of the Internet of Things revolves around the transmission of data through wired or wireless networks. A crucial aspect within this domain is ensuring the security of the data being transmitted within optimal timeframes. PRESENT is a well-known lightweight cryptosystem for wireless communication. GIFT is an improved version of it. It is designed to optimize throughput. This paper presents the hardware implementation of the GIFT cipher on an FPGA, including simulation with a testbed. The process involves the generation of random plaintext and keys, which are inputs to the cipher, followed by conducting power leakage analysis and electromagnetic (EM) leakage analysis on GIFT. Subsequently, based on these analyses, a secure Trojan-free VLSI Physical design of GIFT has been presented. The paper also outlines the schematic designs of the GIFT cipher, along with the verification of layout and simulation results.

Key-Words: - GIFT Block Cipher, Power analysis, Electromagnetic analysis, Side channel Cryptanalysis, FPGA, Hardware design.

Received: March 29, 2026. Revised: April 26, 2026. Accepted: June 21, 2026. Published: October 12, 2026.

1 Introduction

The Internet of Things (IoT) introduces a novel network environment where not only devices but also various objects are interconnected, sharing data. The term "thing" encompasses any smart entity possessing data. Networks comprised of such entities may be compact and constrained in battery usage. Minimizing power consumption is crucial for operations like data transfer between devices. In this context, data security heavily relies on the integration of cryptographic algorithms into the devices. Absent these algorithms, the confidentiality and integrity of data become highly questionable.

To mitigate such risks, it is imperative to embed a robust cipher in smart devices. In this scenario, a cryptographic primitive with high security is essential, such as 'Lightweight Cryptography,' specifically designed for resource-constrained

environments. An example of this is the GIFT Cipher, a Lightweight Symmetric Block Cipher. Optimizations are required to seamlessly integrate the cipher into the IoT environment.

This paper addresses the hardware implementation, side-channel cryptanalysis focusing on power and electromagnetic leakages, and the development of a secure VLSI physical design free from Trojans. It emphasizes layout verification and simulation of the physical design, providing insights to effectively integrate the 'GIFT Cipher' into the IoT landscape.

The presentation of the paper is as follows. The structure of the GIFT cipher has been discussed in section-2. Section 3 describes the literature survey related to the GIFT cipher, and the hardware implementation of the GIFT block cipher is presented in Section 4. Communication between FPGA and PC happens by Serial communication using FIFO, as

described in Section 5. Power model, EM model leakages, and side channel correlation have been discussed in Sections 6, 7, and 8, respectively. Section 7 discusses the secure hardware physical design of the cipher GIFT. Finally, conclusive remarks have been discussed in Section 8.

2 GIFT Cipher Design

A small, fast, energy-efficient, and more secure SPN (Substitution Permutation Network) block cipher, named GIFT, is proposed in [1] by improving PRESENT. GIFT is a family of two very lightweight block ciphers: GIFT64(28-round with 64-bit block size) and GIFT128(40-round with 128-bit block size). Both versions have a 128-bit key size. Each round of GIFT consists of 4 steps: SubCells, PermBits, AddRoundKey, and Constant XOR. GIFT is composed of only a Substitution box (S-Box) and bit-wiring, but its natural bit-slice data flow ensures excellent performance in all scenarios, from area-optimized hardware implementations to very fast software implementations on high-end platforms. It gives better performance than existing lightweight block ciphers in terms of area, throughput, and energy.

The following GIFT structure shown in Figure 1 gives the flow of converting plaintext into ciphertext.

**Plaintext---->SBOX---->Permutation---
->AddRoundKey----->ConstantXOR**

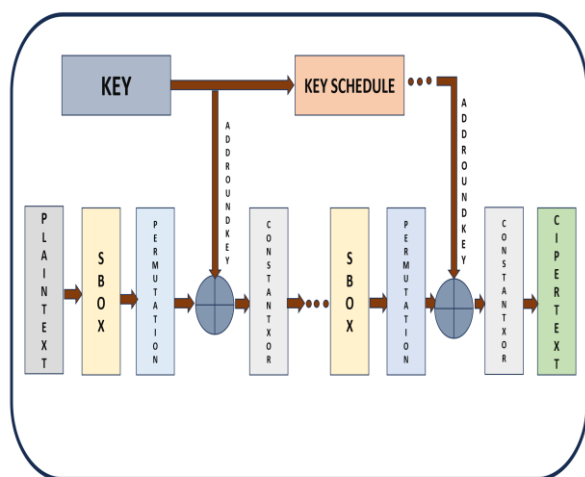


Fig. 1: GIFT Structure
Source: Created by the authors

A brief overview of the structure is as follows:

Round keys have been generated using the Key Scheduling Algorithm (KSA) and the master secret key. These round keys will be used in each round.

Substitution is performed using a predefined S-Box (substitution box) to introduce non-linearity into

the cipher, which adds more confusion between the plaintext and ciphertext.

GIFT employs a permutation operation to shuffle the bits within each half of the block, which adds the diffusion property to the structure.

The output of one round becomes the input for the next round. It is an iterative process; after performing a specified number of rounds, the output of the last round is the ciphertext. A detailed description of the GIFT cipher can be seen in [1].

The S-Box is a key component of the round function and introduces non-linearity into the cipher. The S-Box in GIFT is constructed based on the Rijndael S-Box, which is also used in the AES algorithm. The GIFT S-Box is a 4x4 matrix where each element is a 4-bit value. The S-Box is applied independently to each nibble (4 bits) of the input data during the substitution phase of the round function. The specific values in the S-Box are chosen to provide desirable cryptographic properties such as confusion and diffusion. Table 1 shows the inputs and outputs of the S-Box used in the GIFT algorithm. It's worth noting that the choice of the S-Box and its properties play a crucial role in the security of a block cipher. The design of the GIFT S-Box is based on established principles in cryptography, and the S-Box is carefully crafted to resist various attacks.

After substitution, the result is combined with other components such as the linear permutation, key mixing, and XOR operations to complete one round of the GIFT cipher. The permutation introduces a linear transformation to the bits in the block. The permutation is specifically applied to shuffle the bits within the block. By which one can achieve the diffusion. S-Box and permutation both add confusion and diffusion properties to the cipher so that the cipher structure becomes more secure.

Table 1. GIFT S-Box

Input	Output
0000	0001
0001	1010
0010	0100
0011	1100
0100	0110
0101	1111
0110	0011
0111	1001
1000	0010
1001	1101
1010	1011
1011	0111
1100	0101
1101	0000
1110	1000
1111	1110

Source: [1]

3 Literature Survey

The hardware implementation aspects of GIFT have been discussed in [1]. The focus of this paper revolves around the extensive implementation area dominated primarily by storage and S-Boxes. Opting for a more economical S-Box could result in a considerably more vulnerable proposal, establishing robust constraints against both differential and linear attacks.

GIFT stands out with a robust security margin and a small gate footprint. In [2], the authors have presented the optimization techniques, including bit slicing and $\times$ slicing, to attain optimal implementations for both GIFT-64 and GIFT-128 using the RV 32I ISA. A quantitative performance assessment is conducted using a physical board. Bit slicing effectively reduces the number of clock cycles, enhancing efficiency.

The paper [3] explores comprehension and hardware implementation, shedding light on implementation vulnerabilities such as side-channel attacks (SCAs) and providing countermeasures against potential leakages. The authors have claimed that their secure implementation incorporates a novel lightweight countermeasure, comprising two balanced and masked S-Boxes. In contrast, their suggested countermeasure intelligently combines balancing and masking, ensuring comprehensive protection with minimal overhead.

Securing image data is a paramount consideration. The paper [4] presents hardware solutions dedicated to preserving image data, utilizing the lightweight block cipher GIFT for integrated encryption and decryption operations on an FPGA, specifically the Xilinx Zynq Ultrascale+ xczu9eg. The system operates at a frequency of up to 590 MHz, consuming 673 mW of power. Notably, the design exhibits proficiency in processing standard

gray-level and RGB color image sizes, including 256_256, 512_512, and 1024_1024. Substantial reductions in the average correlation coefficient are observed in the horizontal (99:59%), vertical (99:97%), and diagonal (99:92%) directions.

A quantum circuit design for the GIFT cipher has been introduced in [5]. The proposed method optimizes PRESENT and GIFT by minimizing qubits, quantum gates, and circuit depth, comparing the proposed quantum circuits. Finally, the quantum resources needed for the Grover search algorithm's oracle

with PRESENT and GIFT block ciphers are estimated.

Lightweight cryptography plays a crucial role as a cornerstone in securing embedded systems, especially for resource-constrained devices with

stringent requirements. Its key attributes include reduced power consumption, minimal memory usage, and more. The paper [6] conducts a survey on diverse lightweight block ciphers like PRESENT, GIFT, RECTANGLE, PICO, and LED. The strengths and weaknesses of each algorithm, along with a summary of various attacks targeting them, are discussed.

A novel cache attack on GIFT, termed GRINCH, has been presented in [7], which successfully recovers the full key with fewer than 400 encryptions.

GIFT stands as one of the lightweight cryptography (LWC) candidates in the NIST LWC competition. The selected schemes from this competition are anticipated to be implemented in Internet-of-Things (IoT) devices, marking a pivotal development for the present and future information and communication technology market. Several authors have implemented the GIFT cipher and have given some suggested countermeasures against side channel attacks. Some attacks are performed on GIFT, and some discuss the quantum implementation aspects. Our aim is to show how one can perform the power/electromagnetic leakages on the design, and also how to develop a trojan-free secure physical design at the transistor level. For this, we have initially implemented on the SAKURA-G FPGA board and tried to capture traces by power analysis and EM analysis. Based on this, a secure hardware implementation has been done, and also a secure physical design with minimized transistors has been developed.

4 Hardware Implementation

Initially, GIFT was implemented using the C language (software) and also Verilog code (Hardware Description Language). These two codes have to be verified and compared by the Golden model testbed.

4.1 Simulation with Golden Model Testbed (C and Verilog)

A simulation with a golden model testbed is a common practice in the field of digital design and verification, particularly when working with hardware description languages like Verilog and software languages like C or C++ [8], [9], [10]. This technique is used to verify the correctness and functionality of a hardware design by comparing it with a known, trusted, or reference model, often referred to as the "golden model". This model serves as the reference for the correct behaviour of the system or component we are designing. It's essential

to ensure that the golden model is thoroughly tested and debugged to serve as a reliable benchmark. The detailed process is described below.

Design Under Test (DUT): This is the component or system that you are developing.

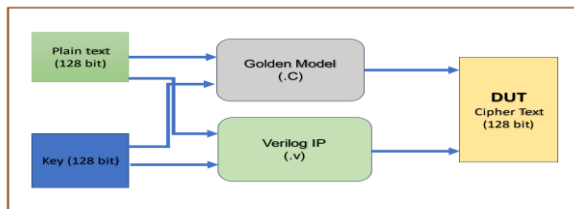


Fig. 2: Golden Model Testbed

Source: Created by the authors

Design Under Test (DUT): This is the component or system that you are developing. The DUT is the

hardware design or Verilog module used to test and verify. It contains stimulus generation, input signal

monitoring, and comparison logic to check the DUT's outputs against the golden model's expected outputs. This simulation helps validate that the DUT behaves as expected and matches the behaviour of the golden model under various test cases and scenarios. Any discrepancies or differences between the two outputs arranged as potential issues or bugs in the DUT. A testbench is a simulation environment used to verify the DUT's (Design Under Test) functionality.

Debugging: If there are discrepancies or issues, debugging tools and techniques can be used to identify the source of the problem within the DUT. Debugging might involve examining waveforms, logs, or other simulation data to trace the issue.

This simulation and debugging process is often iterative. Once the DUT passes all the tests and meets the required specifications, a verification report is generated that contains the verification process, results, and any deviations from the golden model's behaviour.

GIFT Golden Model Simulation:

GIFT-128 Block Cipher has been implemented in both hardware code (Verilog) and software code (.C) for testing with a Golden model. In this method, random plaintext and keys are generated using file operations shown in Figure 2. A single software tool has been designed to facilitate testing for both the Verilog and C code. This Verilog simulation has been done on GTKWave.

GIFT-128 has been implemented using Verilog and simulated on the GTKWave tool. The following Figure 3 shows the simulation results.

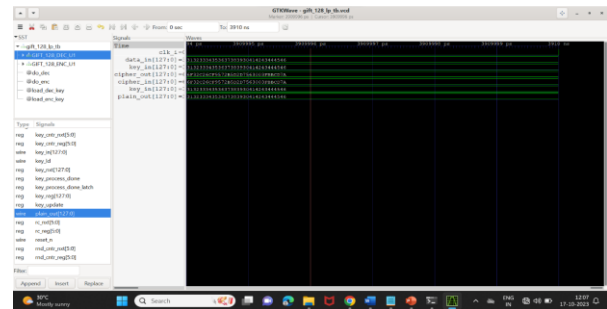


Fig. 3: Hardware Simulation

Source: Created by the authors

GIFT-128 has been implemented using the C language (Golden model/software). The following Figure 4 shows the results.



Fig. 4: Software Simulation

Source: Created by the authors

The Golden model and the Verilog simulation take plaintext and the key as inputs and output ciphertexts. DUT verifies whether the generated ciphertexts from both implementations match. Multiple test cases are employed for thorough testing. Results of DUT can be found in Figure 5.

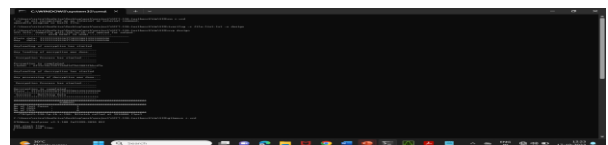


Fig. 5: DUT verification Results

Source: Created by the authors

Register-Transfer Level (RTL): RTL is a level of abstraction in digital design that represents the flow of data between registers within a digital circuit. It describes how data moves between registers and the logical operations performed on the data.

Synthesis: Synthesis is the process of translating a high-level hardware description (often written in an HDL like Verilog or VHDL) into a lower-level representation that can be implemented in hardware. RTL synthesis specifically focuses on the conversion of RTL-level descriptions into a netlist, which is a representation of the circuit in terms of logic gates and flip-flops.

In summary, RTL synthesis is a pivotal step in transforming a high-level hardware description into a detailed netlist(gate-level) that can be implemented in hardware. It plays a crucial role in optimizing the design for performance, area, and power, as shown in

Figure 6. In this work, Xilinx-ISE design suite has been used for RTL synthesis to generate a BIT File (.bit). The bit file contains all of the information necessary to properly configure the device that was created by the implementation tools for our design.

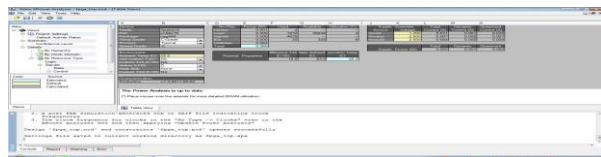


Fig. 6: RTL Summary
Source: Created by the authors

4.2 SAKURA-G FPGA Architecture

The SAKURA-G board has been used for implementing the GIFT-128 block cipher. SAKURA-G FPGA consists of two FPGAs: the main FPGA and the controller FPGA. GIFT-128 has been implemented, and the BIT File (.bit) (generated from RTL synthesis) has been configured on the main FPGA. The controller FPGA is used for FIFO and data communication purposes, and it communicates with the main FPGA and PC. i.e., it works as a data transmitter/receiver between the main FPGA and the PC. This can be understood from Figure 7.

The hardware implementation of the GIFT Algorithm requires 40 clock cycles and is configured in the SAKURA-G FPGA Spartan 6 x75 family. Additionally, the Controller FPGA in the Spartan-6 x9 family features FIFO and buffer register implementations. Each FPGA is equipped with an individual oscillator operating at a frequency of 48 MHz. The main goal is to target the Main FPGA, identify any leakages, and retrieve the secret information of the crypto-algorithm. The crucial aspect is to ensure synchronization among all devices.

The crypto-algorithm operates at a frequency of 24 MHz, as the oscillator frequency is 48 MHz. The crypto-algorithm runs on the positive clock edge. To capture the leakages, the starting and ending points of the algorithm cannot be identified. Therefore, other reference clocks are used for debugging and triggering the capture. These reference clocks are generated from the main FPGA and sent to GPIO pins to provide an external clock reference for the oscilloscope. The following Figure 7 shows the configuration of the GIFT algorithm on an FPGA board and the communication flow.

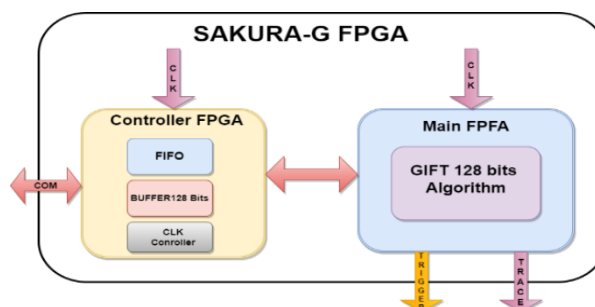


Fig. 7: SAKURA-G FPGA configuration
Source: Created by the authors

5 Communication

Communication between the FPGA and the PC happens via serial communication using a FIFO. The FPGA functions as a physical cryptographic system where random plaintext and random keys are provided as input, and the resulting ciphertext is produced as output and transferred back to a PC through communication. To facilitate this, a software tool (using a Python program) has been developed which generates random plaintexts and keys.

5.1 Serial Communication using FIFO

Serial communication using FIFO (First In, First Out) often involves the use of a buffer or queue to manage data transfer between devices.

Future Technology Devices International (FTDI) is well-known for its USB interface chips and modules, which enable communication between a host device (like a computer) and other devices through USB connections. This communication happens in the following three ways.

FTDI USB-to-Serial Chips: FTDI's USB-to-serial chips serve as a connection between USB and standard serial communication interfaces. They enable the integration of USB connectivity into devices utilizing UART (Universal Asynchronous Receiver-Transmitter) for serial communication.

Virtual COM Port Drivers: FTDI offers Virtual COM Port (VCP) drivers, enabling USB-to-serial chips to emulate standard COM ports on a computer. This streamlines the integration of these devices into pre-existing software designed for communication over serial ports.

FTDI FIFO Mode: FTDI chips frequently include support for FIFO (First In, First Out) modes, enabling efficient data transfer between USB and serial interfaces. This capability enhances the performance of data-intensive applications by utilizing a buffer (FIFO) for temporary storage of data.

Software tool for Random number generation and Communication:

This tool generates a 16-byte plaintext and a 16-byte key, supplies them to the FPGA, and then retrieves the resulting 16-byte ciphertext back to the PC using 245 FIFO communication. The communication occurs through FTDI with FT_prog. The software tool operates in conjunction with the FT_prog software, communicating through the serial communication port and identifying the FPGA FTDI by allocated serial numbers, shown in the Figure 1. SAKURA GIFT SCA Communication was designed to generate random data along with the transmission of plaintexts and keys, and then receive the outputs in the form of ciphertexts, as shown in Figure 9 below. Using FTDI communication, all communications were done through FIFO communication.

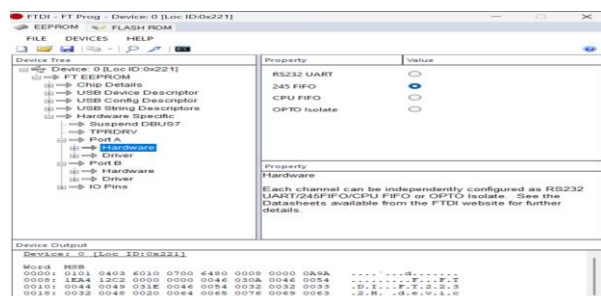


Fig. 8: FTDI Program tool
Source: Created by the authors

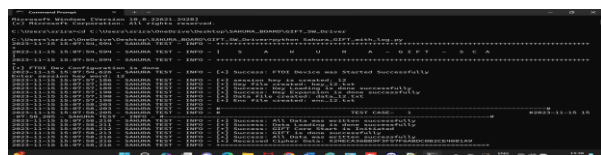


Fig. 9: Communication output
Source: Created by the authors

6 Side Channel Cryptanalysis

Now the complete hardware design and communication between the FPGA and PC have been set up for side channel cryptanalysis [11], [12]. The power leakage and electromagnetic leakage side channel analysis on GIFT have been performed, and the detailed leakage models are discussed below.

6.1 Power Leakage Model

The proposed GIFT algorithm with 128 bits was modeled in Verilog and was tested through simulation, synthesis, and place & route operations via Xilinx ISE design suite. The created .bit file was loaded onto SAKURA-G via JTAG programming cable. Once the FPGA board has been configured, the communication has been set up using a software program, and power traces were captured while the algorithm runs on the platform. This experiment was

performed on the SAKURA-G board shown in Figure10.

As mentioned earlier, one of the advantages of employing the SAKURA-G development kit in this experiment is due to the fact that it comes equipped with a 1-ohm resistor in parallel connection with the FPGA power source. Hence, this allows the power measurement to be done using an SMA cable connected to the power source. Therefore, distinct power traces per GIFT encryption round have been successfully captured in the experiment.

For cryptanalysis purposes, where the secret key will be recovered from power traces obtained in this experiment, several traces should be gathered using different plaintexts [13], [14]. There are several ways to analyze power traces, such as statistical methods, machine learning [15], and neural network techniques [16],[17],[18],[19],[20].



Fig. 10: Power leakage model
Source: Created by the authors

In this paper, the following tools and equipment were utilized: SAKURA-G FPGA, mixed-signal oscilloscope, SMA to BNC cable, Xilinx ISE 14.4 suite, FTDI with FIFO communication, Python communication tool with Master key, PT and CT, and analysis tools. GIFT, a 128-bit block cipher, was implemented on the SAKURA-G FPGA. Subsequently, a side-channel power attack was conducted, and power traces were successfully captured shown in Figure 11.



Fig. 11: Power leakage traces of GIFT-128
Source: Created by the authors

6.2 Electromagnetic (EM) Leakage Model

An electromagnetic side-channel attack uses the electromagnetic emissions from electrical equipment to steal data [21]. In this attack model, a hardware (Verilog) implemented GIFT128-bit crypto-algorithm, FIFO communication, simulated using the Golden model testbed and synthesis, PnR, generated

a .bit file, and configured in the SAKURA-G FPGA. The other part is the generation of a RAND plaintext and key with FIFO communication using Python programming. SAKURA-G FPGA has an FTDI communication chip for power and communication. For communication between PC and FPGA through FT Prog software tool with 245 FIFO.

SAKURA-G FPGA, PC, Mixed signal Oscilloscope (MSO), Trigger probe, and EM probe are required to set up an EM Side channel attack. In this method, the FPGA works as a Crypto-system, the PC sends the RAND Plaintext & key, and receives the ciphertext through serial FIFO communication. During the process of encryption, EM radiation has to be captured using an EM Probe and stored in MSO. Multiple plaintext and ciphertext pairs are required for a differential EM side-channel attack. EM leakage setup on SAKURA-G FPGA can be seen in Figure 12.

In this attack, one cannot notice the start and end of the encryption process in the EM spectrum. To notice the starting and ending points of encryption, the following needs to be done. A reference clock is allocated to a GPIO pin, which is then connected to one of the channels of the MSO. This reference clock will be synchronized with the Edge Trigger on MSO. This can be seen in Figure 13.



Fig. 12: EM leakage model
Source: Created by the authors

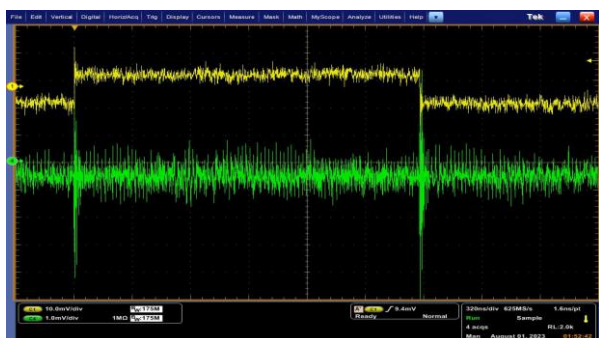


Fig. 13: EM traces of GIFT-128
Source: Created by the authors

6.3 Side Channel Correlation Analysis

Correlation Power Analysis (CPA) can be performed on the GIFT-128 implementation after capturing real-time power and electromagnetic (EM) leakage traces. CPA is a statistical side-channel attack that takes advantage of the correlation between the captured leakage traces and the internal processes of the cryptographic algorithm during the encryption process. Both power and EM leakages are utilized to develop the CPA attack model from the FPGA implementation. The same can be applied to the GIFT-128 algorithm.

Power analysis provides the overall device activity, and EM analysis provides the ability to observe the internal activities of a device that involves the cryptographic algorithm. The process of preparing the CPA attack model and performing the side-channel analysis is described in detail in [22].

The Pearson correlation equation is really useful for people who analyze things. They use it to do something called Correlation Power Analysis, or CPA for short, to see how the power a device uses is related to what they thought it would use. The Pearson correlation equation helps with this. When they try to figure out the key, they make an educated guess about how much power the device will use for each possible key. They use models, like the Hamming Weight model, to make these guesses. Then they compare these guesses to the power used by the device at many different points. The key that they think is the most likely to be correct is the one that matches the power usage to what they actually saw. This is because the Pearson correlation equation helps them find the key with similar power consumption. The Correlation Power Analysis technique is important here.

The Pearson correlation coefficient is defined in equation 1 as follows:

$$\rho_{k,j} = \frac{\sum_{i=1}^N (H_i - \bar{H})(t_{ij} - \bar{t}_j)}{\sqrt{\sum_{i=1}^N (H_i - \bar{H})^2 \sum_{i=1}^N (t_{ij} - \bar{t}_j)^2}} \quad (1)$$

where H_i represents the hypothetical leakage obtained from the leakage model, t_{ij} denotes the measured power sample at the time instant j for trace i , N is the total number of traces, and $\bar{H}$ and $\bar{t}_j$ correspond to the mean values of the hypothetical leakage and measured traces, respectively.

7 GIFT Block Cipher Physical Design

VLSI physical design is a crucial phase in the overall chip design process. It involves transforming a logical representation of a circuit into a physical layout that can be fabricated. Cadence tool is used for the development of integrated circuits and electronic

systems. In this paper, the Cadence tool has been used for the physical design. Dsch tool and Microwind tools have been used for designing schematic designs and layout designs, respectively.

Microwind features a high-precision CMOS layout editor that accommodates technologies ranging from 1.2 μ m to 22nm, showcasing exceptional illustration capabilities. The tool's advanced editing commands and layout control are designed to significantly reduce our development times beyond our expectations.

GIFT cipher has 16- 44 S-Boxes. Some basic gates are required to design an S-Box. The schematic design of the required basic gates at the transistor level and also the verification of those designs have been done using the Dsch tool. Microwind tool has been used for the design of physical layout, simulation, and verification.

7.1 Physical Design of Basic Gates

Optimized implementation of basic gates at the transistor level and physical layout has been presented. From the Figure 14 and Figure 15, and Figure 16 below, one can visualize the difference between the normal hardware design of gates and their physical design

NOR gate: Schematic design of the NOR gate with 4-MOS transistors is shown in Figure 14. 2 P-MOS transistors are used as full-up transistors, and 2N-MOS transistors are used as full-down transistors, and the same has been verified using the Dsch tool. The physical layout design of the NOR gate, verified by the Microwind tool, has been developed. Figure14 shows the schematic design and physical layout of a NOR Gate.

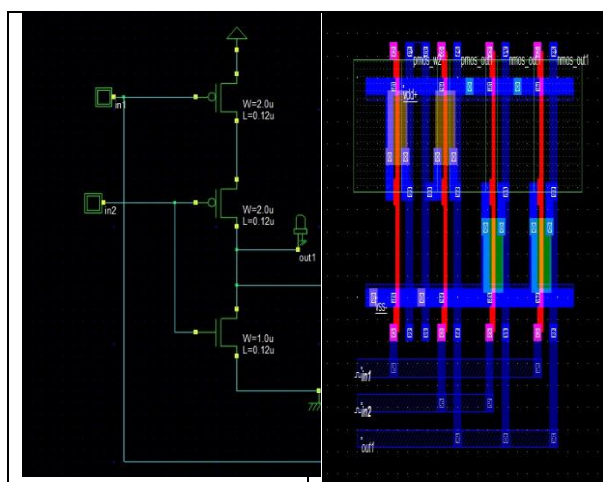


Fig. 14: NOR Schematic Designs and Layout
Source: Created by the authors

NAND Gate: Schematic design of a NAND gate with 4-MOS transistors is shown in Figure 15. 2 P-MOS transistors are used as pull-up transistors, and

2N-MOS transistors are used as pull-down transistors, and the same has been verified using the Dsch tool. The physical layout design of the NAND gate, verified by the Microwind tool, has been developed. Figure 15 shows the schematic design and physical layout of a NAND Gate.

XNOR GATE: XNOR using CMOS transistors Design requires 10 transistors [23]

But only 4 transistors have been used in the optimized XNOR design. Optimized schematic design of XNOR gate with 4-MOS transistors is shown in Figure 16. 2 P-MOS transistors are used as full-up transistors, and 2N-MOS transistors are used as full-down transistors, and the same has been verified using the Dsch tool. The layout design of the XNOR gate, verified by the Microwind tool, has been developed. Figure 16 shows the schematic design and physical layout of the XNOR Gate.

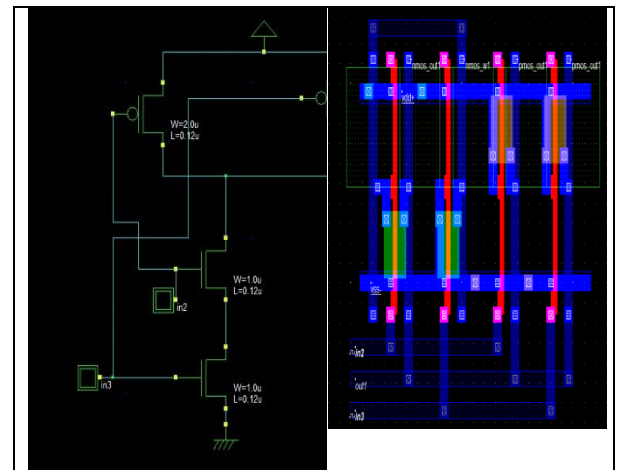


Fig. 15: NAND Schematic Design and Layout
Source: Created by the authors

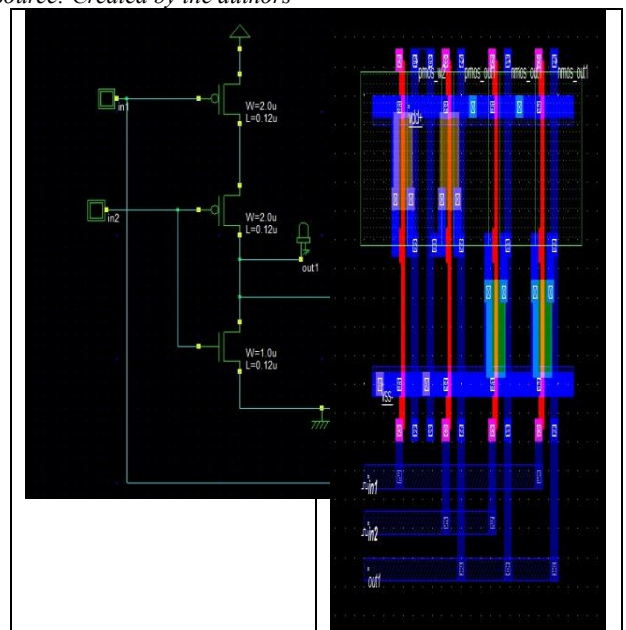


Fig. 16: XNOR Schematic Design and Layout
Source: Created by the authors

7.2 Hardware Design of GIFT S-Box

The optimized design of a 4X4 S-Box requires 6-
XNOR, 3-NAND, and 1-NOR gates as shown in
Table 2. The schematic design of the S-Box is shown
in Figure 17.

S-Box layout using Microwind: The non-linear
function's (S-Box 4x4) physical design employs a
minimization approach, emphasizing a Trojan-free
design [24]. Our design strategy involves minimizing
transistor usage to mitigate vulnerability to Trojan
injections. The fundamental blocks incorporate fewer
transistors across various design layers, such as
metal-to-metal, polysilicon layers, and connections.
Layout design of 4x4 S-Box is shown in Figure 18.
This approach ensures heightened resistance against
Trojan attacks.

Simulation of S-Box physical layout:
Verification of CMOS circuits in Microwind doesn't
require SPICE or an external simulator. The program
incorporates an internal analogy-like simulator that
supports MOS Level 1, Level 3, or BSIM4 models.
With capabilities such as rapid time-domain analysis,
voltage and current estimation, intuitive post-
processing, frequency estimation, and delay
estimation, PROthumb emerges as a time- saving
tool. Additionally, on-screen power estimation for
circuit simulation is also available. Post-simulation
results of the S-Box physical layout are shown in
Figure 19.

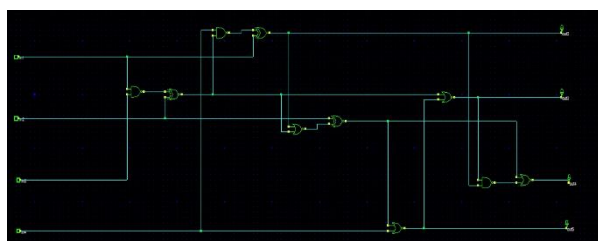


Fig. 17: Schematic design of S-Box

Source: Created by the authors

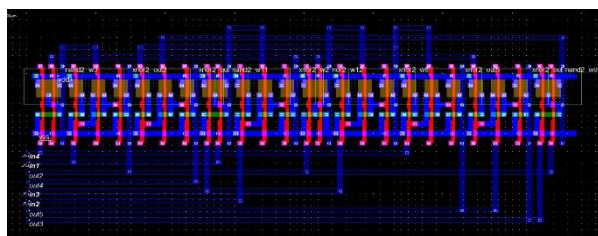


Fig. 18: Layout Design of 4x4 S-Box

Source: Created by the authors

Table 2. S-Box implementation

Input: 4-bit input (A3, A2, A1, A0) (before entering S-Box).		
Output: 4-bit output (A3, A2, A1, A0) (after performing S-Box).		
Step	Operation	Test input
Input	A3 A2 A1 A0	1 0 1 0
1	A1 = A1 XNOR (A0 NAND A2)	1 XNOR (0 NAND 0) = 1
2	A0 = A0 XNOR (A1 NAND A3)	0 XNOR (1 NAND 1) = 1
3	A2 = A2 XNOR (A0 NOR A1)	0 XNOR (1 NOR 1) = 1
4	A3 = A3 XNOR A2	1 XNOR 1 = 1
5	A1 = A1 XNOR A3	1 XNOR 1 = 1
6	A2 = A2 XNOR (A0 NAND A1)	1 XNOR (1 NAND 1) = 0
Output	A3 A2 A1 A0	1 0 1 1

Source: Created by the authors

7.3 Counter for GIFT 40 Rounds

GIFT-128 has 40 rounds, Up/down counter has been
used for counting 40 rounds. This counter counts
from 00 to FF(255), but the requirement is for only
40 rounds. The 40-bit counter design uses an 8-bit
full adder, an 8-bit DFF with a synchronization
clock. .

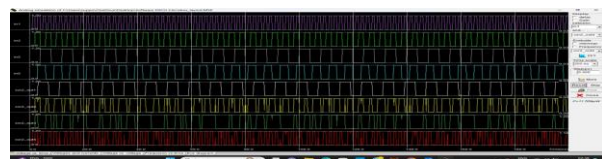


Fig. 19: S-Box layout simulation

Source: Created by the authors

LFSR: The GIFT 128-bit cipher has a 6-bit LFSR
for 40 rounds of constant value generation. LFSR has
a 5-degree polynomial. In this connection, the design
uses 6 D-FF and one XOR gate. The schematic
design and layout designs of the LFSR have been
shown in Figure 20 and Figure 21.

The use of LFSRs is prevalent in various
applications, including digital signal processing,
cryptography, and error detection. The 6-bit LFSR
produces a sequence of $2^6 - 1$ (63) unique states
before repeating, making it suitable for certain
applications requiring pseudo-random sequences of
moderate length.

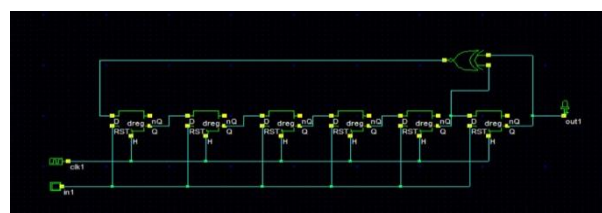


Fig. 20: LFSR

Source: Created by the authors

Designing a 6-bit LFSR with a VLSI layout using Microwind involves utilizing the software's tools to place and connect transistors to implement the LFSR functionality.

The layout aims to optimize the use of space on the chip while meeting performance and power consumption requirements.

Simulation within Microwind allows engineers to verify the design's functionality and performance before actual fabrication.

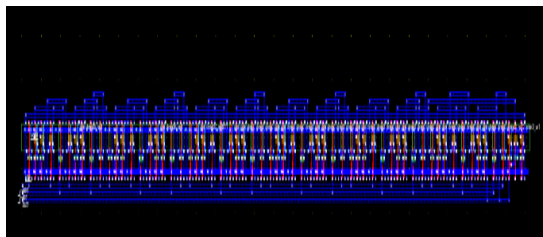


Fig. 21: LFSR layout

Source: Created by the authors

Counter for GIFT 40 rounds: GIFT 128 has 40 rounds, so counting 40 rounds using an up/down counter. This counter counts 00 to FF, but the requirement is 40 rounds. 40-bit counter design using 8-full adders, 8 DFFs, and a synchronization clock. GIFT 128 has a 40-round designed layout using Microwind with 22nm technology.

The schematic design and layout designs of the counter have been shown in Figure 22 and Figure 23

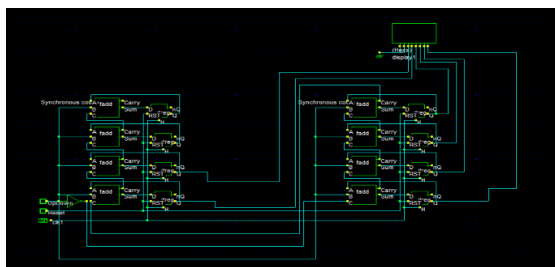


Fig. 22: Counter for GIFT 40 rounds

Source: Created by the authors

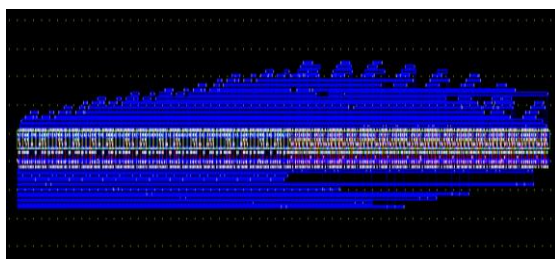


Fig. 23: Counter for GIFT 40 rounds layout

Source: Created by the authors

The development of the new GIFT nonlinear function design involves using the minimized transistor-level design technique, which helps in the

formation of a small and secure hardware architecture. By minimizing the logic gates, unused nodes, and other routing paths used in the design, the possible locations for inserting a Hardware Trojan can be minimized, since Hardware Trojans require additional circuits to trigger the payload circuit. Verification of the nonlinear functions involving the S-box was done through schematic simulations, test cases, layout design, and finally post-layout simulations. Analysis through power side-channel and electromagnetic side-channel tests confirmed that the proposed hardware design exhibits stable behaviour and there is no unusual switching activity in the device. Additionally, tests for Design Rule Checks (DRCs) and Layout Versus Schematic (LVS) tests have been performed on the physical layout. A DRC test shows that the physical layout conforms to all necessary fabrication rules needed in chip fabrication, including spacing and proper routing. LVS test ensures that the physical layout does not involve any extra transistors or unnecessary connections in the circuit. Passing the DRC and LVS tests confirms the correctness of the final implementation of the hardware architecture without additional structures.

8 Conclusion

GIFT is suitable for applications where there are resource constraints such as low power or limited computation capabilities. This makes it an excellent option for implementing cryptographic solutions that prioritize efficiency and lightweight performance. In this paper, the GIFT Cipher algorithm has been coded, simulated, synthesized, and implemented on an FPGA successfully for encryption and decryption of a text message. And also captured power and EM traces for power leakage analysis and EM leakage analysis, respectively. Initially, GIFT is implemented on the SAKURA-G FPGA board and captured traces by power analysis and EM analysis. The GIFT cipher has a 4×4 S-Box, which is a non-linear function. Nonlinear functions are more vulnerable to side-channel attacks. So, the non-linear function S-Box in GIFT has been redesigned with optimized gate-level and transistor-level designs by minimizing the number of transistors in CMOS designs, which yields a secure physical design with minimized transistors. This secure design uses fewer transistors so that the utilization of power is reduced, which resists side-channel analysis.

Declaration of Generative AI and AI-assisted Technologies in the Writing Process

During the preparation of this work, the author(s) utilized artificial intelligence (AI) technologies

solely for the purpose of linguistic refinement and grammatical correction. Following the application of these tools, the author(s) comprehensively reviewed, verified, and edited the resulting text. The author(s) maintain full accountability and responsibility for the accuracy, integrity, and final content of the publication.

References:

- [1] Banik, S., Pandey, S.K., Peyrin, T., Sasaki, Y., Sim, S.M., Todo, Y. (2017). GIFT: A Small Present. In: Fischer, W., Homma, N. (eds) Cryptographic Hardware and Embedded Systems CHES 2017. Lecture Notes in Computer Science, vol 10529. Springer, Cham. doi: [10.1007/978-3-319-66787-4_16](https://doi.org/10.1007/978-3-319-66787-4_16).
- [2] Gheorghe Pojoga and Kostas Papagiannopoulos. 2022. Low-latency implementation of the GIFT cipher on RISC-V architectures. In Proceedings of the 19th ACM International Conference on Computing Frontiers (CF '22). Association for Computing Machinery, New York, NY, USA, 287-295. <https://doi.org/10.1145/3528416.3530996>.
- [3] A. Aljuri, C. Reinbrecht, S. Hamdioui, M. Taouil and J. Sepulveda, "Balanced Dual-Mask Protection Scheme for GIFT Cipher Against Power Attacks," 2022 IEEE 40th VLSI Test Symposium (VTS), San Diego, CA, USA, 2022, pp. 1-6, doi: <https://doi.org/10.1109/VTS52500.2021.9794230>.
- [4] Jangir, A., Pandey, J.G. GIFT cipher usage in image data security: hardware implementations, performance and statistical analyses. J Real-Time Image Proc 18, 2551-2567 (2021). <https://doi.org/10.1007/s11554-021-01146-3>.
- [5] Jang K, Song G, Kim H, Kwon H, Kim H, Seo H. Efficient Implementation of PRESENT and GIFT on Quantum Computers. Applied Sciences. 2021; 11(11):4776. <https://doi.org/10.3390/app11114776>.
- [6] Hatzivasilis, George & Fysarakis, Konstantinos & Papaefstathiou, Ioannis & Manifavas, Harry. (2018). A review of lightweight block ciphers. Journal of Cryptographic Engineering. 8. 1-44. doi: [10.1007/s13389-017-0160-y](https://doi.org/10.1007/s13389-017-0160-y).
- [7] C. Reinbrecht, A. Aljuri, S. Hamdioui, M. Taouil and J. Sepulveda, "GRINCH: A Cache Attack against GIFT Lightweight Cipher," 2021 Design, Automation & Test in Europe Conference & Exhibition (DATE), Grenoble, France, 2021, pp. 549-554, doi: <https://doi.org/10.23919/DAT51398.2021.9474201>.
- [8] Kelvin Ng. 2008. Challenges in using system-level models for RTL verification. In Proceedings of the 45th annual Design Automation Conference (DAC '08). Association for Computing Machinery, New York, NY, USA, 812-815. <https://doi.org/10.1145/1391469.1391676>.
- [9] Mathur, Anmol & Krishnaswamy, Venkat. (2007). Design for Verification in System-level Models and RTL. 193-198, <https://dl.acm.org/doi/10.1145/1278480.1278528>. (Accessed Date: July 20, 2026).
- [10] Doulos, The Verilog Golden Reference Guide: https://class.ece.iastate.edu/cpre488/resources/verilog_reference_guide.pdf, (Accessed Date: July 20, 2026).
- [11] Gattu, N., Khan, M. N. I., De, A., & Ghosh, S. (2020, November). Power side channel attack analysis and detection. In Proceedings of the 39th International Conference on Computer-Aided Design, Virtual Event, San Diego, CA USA, pp. 1-7, Article No.: 65, November 2020, [10.1145/3400302.3415692](https://doi.org/10.1145/3400302.3415692).
- [12] Das, D., & Sen, S. (2020). Electromagnetic and power side-channel analysis: Advanced attacks and low-overhead generic countermeasures through white-box approach. Cryptography, 4(4), 30, <https://doi.org/10.3390/cryptography4040030>.
- [13] S. Mangard, A simple power-analysis (SPA) attack on implementations of the AES Key Expansion, in: International Conference on Information Security and Cryptology, Springer, Berlin, Heidelberg, 2002, pp. 343-358. [10.1007/3-540-36552-4_24](https://doi.org/10.1007/3-540-36552-4_24).
- [14] Kocher, P., Jaffe, J. and Jun, B. (1999) Differential Power Analysis. In: Wiener, M., Ed., Annual International Cryptology Conference, Springer, Berlin, 388-397. https://doi.org/10.1007/3-540-48405-1_25.
- [15] Lerman, Liran; Bontempi, Gianluca; Markowitch, Olivier (1 January 2014). "Power analysis attack: an approach based on machine learning". International Journal of Applied Cryptography. 3 (2): 97-115, doi: [10.1504/IJACT.2014.062722](https://doi.org/10.1504/IJACT.2014.062722).

- [16] Hospodar, G., Gierlichs, B., De Mulder, E. et al. Machine learning in side-channel analysis: a first study. *J Cryptogr Eng* 1, 293-302 (2011). <https://doi.org/10.1007/s13389-011-0023-x>,
- [17] Valk, D.V., & Picek, S. (2019). Bias-variance Decomposition in Machine Learning-based Side-channel Analysis. *IACR Cryptology{ePrint} Archive.*, 2019/570, <https://eprint.iacr.org/2019/570>, (Accessed Date: July 20, 2026)
- [18] Hettwer, B., Gehrer, S. & Güneysu, T. Applications of machine learning techniques in side-channel attacks: a survey. *Journal of Cryptographic Engineering*. 10, 135-162 (2020). <https://doi.org/10.1007/s13389-019-00212-8>,
- [19] W. Shan, S. Zhang, J. Xu, M. Lu, L. Shi and J. Yang, "Machine Learning Assisted Side-Channel-Attack Countermeasure and Its Application on a 28-nm AES Circuit," in *IEEE Journal of Solid-State Circuits*, vol. 55, no. 3, pp. 794-804, March 2020, doi: [10.1109/JSSC.2019.2953855](https://doi.org/10.1109/JSSC.2019.2953855).
- [20] M. Mushtaq, A. Akram, M. K. Bhatti, R. N. B. Rais, V. Lapotre and G. Gogniat, "Run-time Detection of Prime + Probe Side-Channel Attack on AES Encryption Algorithm," 2018 Global Information Infrastructure and Networking Symposium (GIIS), Thessaloniki, Greece, 2018, pp. 1-5, doi: [10.1109/GIIS.2018.8635767](https://doi.org/10.1109/GIIS.2018.8635767).
- [21] D. -H. Seo, M. Nath, D. Das, S. Ghosh and S. Sen, "Improved EM Side-Channel Analysis Attack Probe Detection Range Utilizing Coplanar Capacitive Asymmetry Sensing," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 42, no. 12, pp. 4583-4596, Dec. 2023, doi: [10.1109/TCAD.2022.3227077](https://doi.org/10.1109/TCAD.2022.3227077).
- [22] Sriramudu, V. Nalla, D. Narendar and G. Padmavathi, "Model for capturing noise free traces for Side Channel Power Cryptanalysis based on SAKURA-G FPGA and Case study of AES," 2022 International Conference on Recent Trends in Microelectronics, Automation, Computing and Communications Systems (ICMACC), Hyderabad, India, 2022, pp. 1-7, doi: [10.1109/ICMACC54824.2022.10093672](https://doi.org/10.1109/ICMACC54824.2022.10093672).
- [23] CMOS implementation of XNOR: <https://eepower.com/technical-articles/cmos-implementation-of-xor-xnor-and-tg-gates/#>, (Accessed Date: July 20, 2026).
- [24] H. Pearce, V. R. Surabhi, P. Krishnamurthy, J. Trujillo, R. Karri and F. Khorrami, "Detecting Hardware Trojans in PCBs Using Side Channel Loopbacks," in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 30, no. 7, (July 2022), 926–937. <https://doi.org/10.1109/TVLSI.2022.3171174>

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

The authors equally contributed in the present research, at all stages from the formulation of the problem to the final findings and solution.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

No funding was received for conducting this research.

Conflict of Interest

The authors have no conflicts of interest to declare that are relevant to the content of this article

Creative Commons Attribution License 4.0 (Attribution 4.0 International, CC BY 4.0)

This article is published under the terms of the Creative Commons Attribution License 4.0 https://creativecommons.org/licenses/by/4.0/deed.en_US