Low latency synchronous design in SRAM based physical unclonable function (PUF)

Radhika R [1]B.K.Madhavi[2]

1 Department of ECE, Bhoj Reddy Engineering College for Women, Hyderabad, Telangana, India.
Email: radhi.431@gmail.com

2 Department of ECE, SiddharthaInstitute of Engineering and Technology, Hyderabad, Telangana, India.
Email: bkmadhavi2009@gmail.com

Abstract:
SRAM is developed for memory interface in digital system design. Data stored in SRAM are vulnerable to error due to hardware or software error. Secrete Unknown Ciphers (SUC) is proposed as a mean of provisioning of security in data storage. In digital system design digital clone-resistant functions were developed in overcoming the issues with Physical(ly) Unclonable Functions (PUF). However the desiring has a issue with the synchronization mean for security provisioning. The recent developed approach SRAM-SUC is developed using block cipher coding. However, the latency due to delay metric in the design process limits its application. In this paper, a new synchronous approach for SRAM-SUC design is proposed using delay monitoring parameter in latency controlling. The proposed approach illustrated a higher power saving using different FPGA devices.

Index Term:SRAM-SUC, PUF, security coding, resource optimization, block coding.

I. Introduction

The main concern today is cyber security on connected devices; IoT-connected devices must be able to perform their design functions securely. This requires that each connected device have its own unique clone-resistant or non-classified identity. Non-cloneable physical (LA) functions (PUFs) [1] have emerged as a viable solution for authentication of IoT devices. However, PUFs are analog in nature and require fuzzy extractors (FEs) or auxiliary data algorithms (HDAs) to stabilize their noisy responses [2], resulting in high hardware or software ratios and latency [3].

To overcome the shortcomings of PUFs, the authors proposed a new concept of clone-resistant digital functions, which were formed as secret anonymous ciphers (SUCs). SUC is a secure cipher that is randomly generated internally inside a chip. Because the SUC is digital in nature, it is powerful over the lifetime of the electronic device. Each device connected to the SUC creation process must have a system embedded in a chip (SoC) FPGA. Nowadays, SoC FPGAs are gaining popularity as accelerators; Xilinx recently launched the world's fastest data center and ALVEO AI accelerator card. Meanwhile, Intel introduced programmable acceleration cards (PAC) [4, 5]. Accelerator cards based on FPGAs dramatically enhance the performance of industry-standard servers. SoC FPGAs are also widely used in IoT devices such as Intelligent Vision Acceleration, Industry 4.0 Automation, and Automation in Anything for Vehicle (V2X). In addition to improving performance, SoC FPGAs can be deployed for security applications, for example, non-copy or clone-resistant device identification, faster secure encoding, decoding, and hashing. For example, the SRAM PUF integrated from an internal identifier based on the Intel PAC 5005 is used as a hardware block on Stratics 10 SX, MicroSemi SmartFusion2 and IGLOO2 Class S devices.

An obligatory security condition in IoT is appliance authentication; Due to the exponential growth of connected devices, reliable third-party device authentication and supported device authentication (D2D) on 5G networks will be the cornerstone of network communication performance, especially on the server side. 5G networks are designed to support three services: Enhanced Mobile Broadband (EMBB), Massive Machine Type Communications (MMTC), and Ultra Reliable Low Latency Communications (URLLC).

URLLC [6] is a set of features designed to support latency sensitive applications such as industrial internet, smart grid, and intelligent transport systems. These applications require close security [6]. URLLC has a silicon of 1 millisecond [7]. Therefore, the approval must be robust not only from a safety perspective, but also with minimal latency. PUFs with FEs or HDAs can provide a secure authentication system, but they have two main limitations: (1) a small number of challenge responses because PUFs are compatible with hash functions and (2) a high latency, which makes PUF-based approval inconvenient. Make it a reality for many. Time applications. This work provides an advantage in resisting digital reproduction that circumvents both constraints. The synchronization issue in these

DOI: 10.9756/INT-JECSE/V14I4.164

systems is however not achieved due to varying delay conditions in the memory operation. To achieve a synchronous mode of operation, this paper presents a new method for synchronous operation using delay monitoring operation. The rest of this paper is presented in 7 sections. Section 2 outlines the physical unclonable function design and secrete unknown cipher design. Section 3 presents the SRAM-SUC design. Section 4 present the securitymechanism developed for synchronous option using block cipher coding. Section 5 outlines the simulation result and conclusion is outlined in section 6.

II. Physical(ly) Unclonable Functions (PUF) Design and Secret Unknown Ciphers (SUC)

PUFs are an option that takes advantage of the combined electronic, non-electronic, or physical properties of devices to create a unique identity for each device. It is classified into analog and digital PUF

1) Analog PUFs or Matching PUFs: Several cases of analog PUFs have been suggested in the literature [1] [8]. Electronic, delay-based and memory-based PUFs embody the qualities inherent in electronic devices to bring out unique chip identities. While the creation and/or operation of non-electronic PUFs is inherently non-electronic, electronic circuits are used to process and store PUF responses. Corresponding PUFs have two main drawbacks: incompatibility issues and their vulnerability to cloning attacks.

First, the response sites of the analog PUFs are noisy, and the use of FE or HDA is required to determine their response [2] [3]. During the recording phase, the FE creates and stores the associated data, which will be used to reconstruct the original PUF response from the noisy PUF response. Second, P, PUFs are insecure for many attacks; Modeling attacks on the cloning of powerful PUFs is a powerful risk Presented d. Limit the first attack on the judgment-based PUF model and analyze the line and feed-forward PUF structures. Recently, [11] machine learning has performed PUF modeling attacks on many PUFs. Semi-invasive way was used to identify the status of memory-based PUFs [12].

In [13], the side channel attack was used to analyze the structure and power of the PUF by analyzing the application of vague drains. The current attack mode brings together both side channel and modeling attacks In [15], a hybrid attack was launched that brought together side-channel analysis and machine learning to attack particularly vulnerable PUFs, which prevented attackers from monitoring their results.

2) Digital PUFs or Physical PUFs: There is an erratic response to physical PUFs from whether the control levels are physically connected in the semiconductor. Since these physical connections are not affected by external factors such as temperature and voltage changes, existing physical PUFs can reach full reliability.

Three physical PUFs have been proposed in the literature:

VIA PUF: The vertical interconnection access (VIA) has been proposed in PUF [16][17] and uses possibilities through the structure to generate unique and robust IC responses. VIA PUF is a weak PUF with only one response

SD-PUF: Digital Divided PUF (SD-PUF) [18] The VLSI interface takes advantage of the irregularities of the metal wire, which may or may not be connected. [1]] By deliberately placing the two ends of the inter-c-connection plot line together, the inter-c-connection randomness is achieved, and due to the mask difference, the produced masks will not match. This logical connection (online / offline) is called virtual connection. This inconsistency turns into an uncertain connection state SD-PUF combines multiple digital UFs (D-PUFs) from multiple "building chips". The D-PUF unit consists of N rows and M columns of rows. Each unit cell contains a two-input XOR gate where one of its input is connected to the input switch bit and the other is connected to a power 1 diffraction latch, which is connected to a virtual jumper pin as a source of randomness. SDPUF has multiple reactions to the challenge, and can be classified among such powerful PUFs.

SPN-DPUF: The installation permission was recently proposed in the PUF Digital Network (SPN-DPUF) [19]. SPNDPUF consists of three levels: D-PUF, like Layer X [18], performs the killer and the player [20]. SPNDPUF is a powerful digital PUF, which has more hardware than SD-PUF and has the same statistical properties as SD-PUF.

Digital PUFs or physical PUFs contain some static interactions. However, they have two main limitations:

(1) can only be used for ASIC design and

(ii) Design assumptions do not always arrive in practice, mainly due to the limited variation in drawing resolution when transferring from web to web.

Secret Cipher SUCs are copy-resistant digital functions, which do not have the same instability problems as digital PUF, when the SUC is applied to the system without any changes in the chip design. Also, when FPGA resources are not fully utilized by the functional hardware design, SUC can be implemented at nearly zero cost. SUC designs can use random block cipher or random stream cipher.

III. SRAM-SUC design

For security encryption, a lightweight protocol called Block Cipher is used [1]. The inputs and outputs of the block cipher algorithm contain a 64-bit sequence (a number with a value of 0 or 1). These sequences are sometimes

DOI: 10.9756/INT-JECSE/V14I4.164

referred to as blocks and the number of bits in them will be called their length. For a block cipher algorithm, the cipher is a string of 80, 128 bits. This standard does not allow the use of I/O key lengths and encryption. Tones in this order will start from zero and end with less than the order length (block length or key length). The number I enclose a little bit is known as the index and will be in the range 0 <i< 80, 0 <i< 128 depending on the block length and key length. The basic unit of processing in a block cipher algorithm is a byte, a sequence of eight bits per unit. The input, output, and cipher key sequences are processed as byte arrays, which divide these sequences into eight contiguous bit groups to form byte arrays. For the input, output, or cipher key specified by A, the byte in the resulting array will be denoted using one of two forms, or [n] , where n will be in one of the following fields:
Key length = 80 bits, 0< n<16;
Block length = 128 bits,0< n<6;
All block ciphering byte values {b7, b6, b5, b4, b3, b2, b1, b0 will be presented as a set of individual bit values (0 or 1) in order. These bytes are defined as finite field elements using multilingual presentations:
$b_7 x^7 + b_6 x^6 + b_5 x^5 + b_4 x^4 + b_3 x^3 + b_2 x^2 + b_1 x + b_0 = b_i x^i$
For example, the field element given by {01100011 given} defines x6 + x5 + x + 1.
Internally, block cipher operations are performed on two-dimensional arrays of bytes called states. A state contains four rows of bytes, each containing Nb bytes, where Nb is the length of the block divided by the length of 32 divided by. In the state matrix shown by s, each byte has two indices separately, the number of rows being in the range r. <r <4 and its column number is within the range c 0 <c <Nb This allows one byte in the state to be referred to as sr, c, or s [r, c]. For this standard, Nb = 4, such as 0 <c <4 at the beginning of the encoding and the reverse encoder, input - 0, in1, ... in15 are copied by the state of the byte array.
Coefficients are measured to make the overhead polynomial valid. Polynomials can be defined in GF (28) with coefficients, which are the following limited area elements:
a (x) = a3 x3 + a2 x2 + a1 x + a0 |
Which will be referred to as a word in the form [a0, a1, a2, a3] Note that the polynomials in this section behave slightly differently from the polynomials used to define the domain components, although both types of polynomials use the same adjectives, x. The operations in this section are the same as for limited domain components, i.e. bytes rather than bits. Also, the offset of the four-polynomial uses a different word for down sampling as shown below. The difference should always be clear from the context
Let's describe XOR and transform the operation, let's go
b (x) = b3x 3 + b2 x2 + b1 x1 + b0 |
Determine the second polynomial of the four words XOR is implemented by adding limited domain coefficients of the same capacity as XOR. This XOR corresponds to the XOR operation within each of the bytes associated with each word - in other words, the XOR of the entire word value.
Thus, using the above equation
$a(x)+b(x)=(a_3 \oplus b_3) x^3 + (a_2 \oplus b_2) x^2 + (a_1 \oplus b_1) x + (a_0 \oplus b_0)$
Shift is achieved in two steps. In the first step, the polynomial operation $c(x) = a(x) <<b(x)$ is algebraically expanded, and like powers are collected to give
$C(x) = c_6 x^6 + c_5 x^5 + c_4 x^4 + c_3 x^3 + c_2 x^2 + c_1 x + c_0.$
Where
$C_0=a_0.b_0$ $c_4=a_3.b_1 \oplus a_2 . b_2 \oplus a_1 .b_3$
$C_1=a_1.b_0 \oplus a_0 .b_1$ $c_5= a_3.b_2 \oplus a_2.b_3$
$C_2=a_2.b_0 \oplus a_1 . b_1 \oplus a_0 . b_3$ $c_6=a_3.b_3$
$C_3=a_3.b_3 \oplus a_2.b_1 \oplus a_1.b_2 \oplus a_0 .b_3$

The result, c(x), does not represent a four-byte word so the second step of the transformation is to reduce the c(x) modulo by a maximum of 4 degrees. The score can be reduced to less than 4 degrees max for block cipher algorithms, this is done with a set of x4 + 1s, so
$x_i \text{Mod}(x^4+1)=x^{i \bmod 4}$
The modular operation of $a(x)$ and $b(x)$, denoted by $a(x) \oplus\llcorner b(x)$, is given by the four-term polynomial $d(x)$, defined as follows:
$d (x) = d_3 x^3 + d_2 x^2 + d_1 x + d_0$
with
$d_0=(a_0. b_0) \oplus (a_3 . b_1) \oplus (a_2 .b_2) \oplus (a_1.b_3)$
$d_1=(a_1.b_0) \oplus (a_0.b_1) \oplus (a_3.b_2) \oplus (a_2.b_3)$
$d_2=(a2.b0) \oplus (a_1.b_1) \oplus (a_0.b_1) \oplus (a_3.b_3)$

DOI: 10.9756/INT-JECSE/V14I4.164

$d_3=(a_3.b_0) \oplus (a_2.b_1) \oplus (a_1.b_2) \oplus (a_0.b_3)$

- In most types of blades there is a core-shift modulation in this configuration, part of the central state bit is usually not moved to another location. There is no grip in circular transformation of block blades instead circular transformations consist of three different transformations called layers. By "standardized" we mean that each part of the case is used in the same way differential cryptanalysis is a design method for a large part of the population, based on the application of a wide range of strategy options for different levels, to provide protection against arthritis. In the detailed path strategy, each has its own mission:
- **The Linear mixing layer:** guarantees high diffusion over multiple rounds.
- **The non-linear layer:**parallel application of S-boxes that have optimum worst-case non linearity properties.
- **The key XOR layer:** A simple EXOR of the Round Key to the intermediate State.

Prior to the first round, the main XOR level is applied The inspiration for this XOR startup key is as follows The key can be deliberately removed before any level (or the first in the case of a known plain text attack) after the last XOR of the cipher, and thus will not be helpful for cryptographic protection. (For example, initial and final permission) Primary or terminal XOR switches are implemented in many designs To make the encoder and its reverse structure more even, the line-level mixing level of the last round is different from the mixing level in the other round. It can be shown that crypto does not improve or decrease cryptographic security in any way.

IV. Synchronous SRAM-SUC design

The need for a new secure coding standard came into question after security coding appeared to be insecure in barbarian force attacks. Researchers are developing a new secure coding standard called block cipher. The algorithm can use a variable block length j and the duration length The latest specification allows the combination of 80, 128, and 80128-bit keys. Block Cipher defines a FIPS-certified encryption algorithm that provides secure security for wireless communications, secure network routers, electronic financial transactions, secure video systems, and secure data storage for encrypted data storage.

It is a cipher block designed to operate on a variable length block using a variable length key. The standard describes the use of an 80-bit, 128-bit key to encrypt data blocks of length 80 0r 128 bits; Note that it is possible to block all nine combinations of length and block length. The algorithm is written in such a way that the length and/or length can be easily multiplied or multiplied by 32 bits, and it is designed to run efficiently on hardware or software of a given processor. The schematic diagram of the cyber operations block is shown in Figure 1.

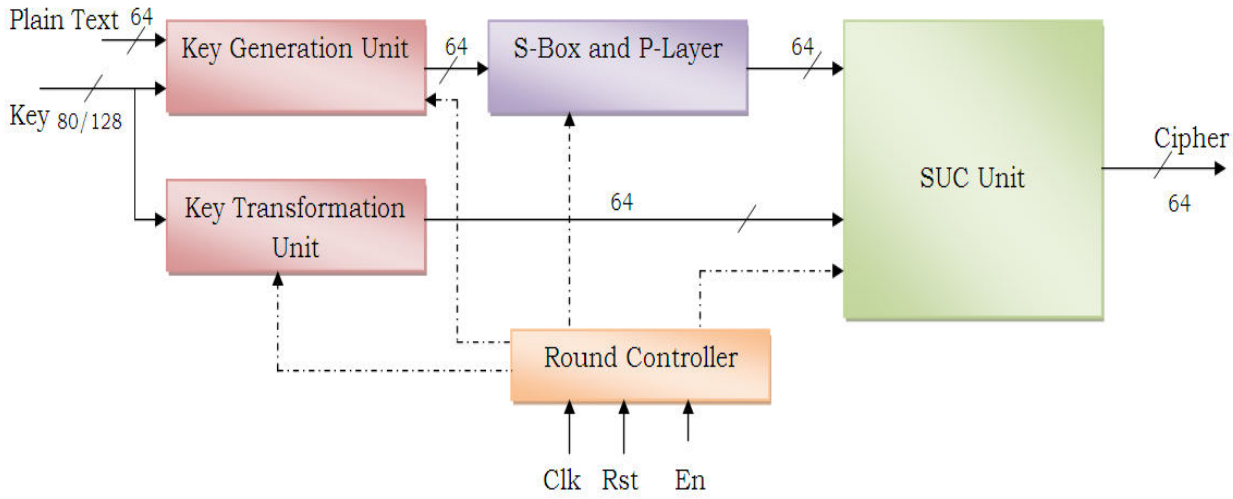


Fig 1: Block Diagram of lightweight block cipher security coding [1]

DOI: 10.9756/INT-JECSE/V14I4.164

The modified structure of the proposed approach is presented in figure 2 below.

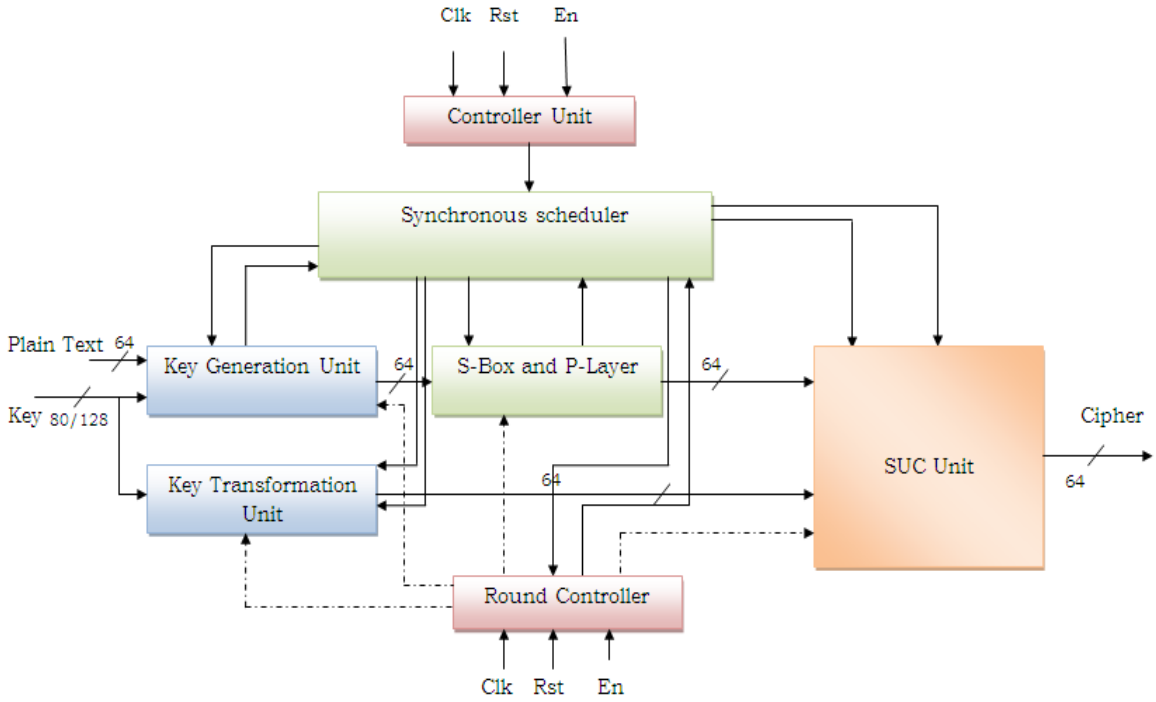


Fig 2: Block diagram of resource controlled light weight block cipher security coding

There is a limit to the upper sources of the previous method, which is the work of large-scale iteration to generate the P and the static method. Although the method is efficient and the source overhead is large enough to overcome the source limit, a balanced source delayed source optimization scheme has been proposed for the record. Here the running clock appears to be running, as the clock is sent along with the processing data along with the processing data to delay the sync. In the case of the switch-scheduling method, it is observed that the conventional clock schedule is proportional to the maximum phase delay, while the operation in the key-scheduling is proportional to the maximum delay difference, resulting in higher clock speeds in the scheduling encoding. The clock distribution is simple to move towards lower load and improve energy saving In such a scheme, the processing history of the banks is given per unit mathematical delay clock pulse. The diagram of the traditional master scheduling pipeline is shown in Figure 1

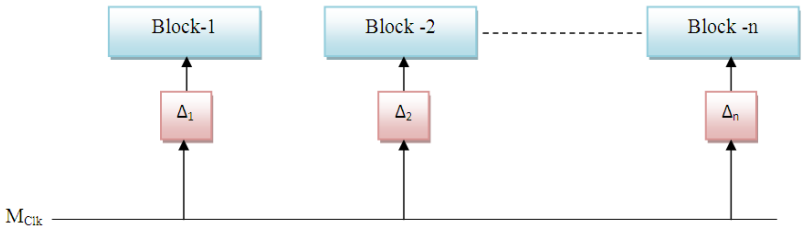


Fig 3: clock allocation in security coding operation

DOI: 10.9756/INT-JECSE/V14I4.164

It parses the delays to the running node based on the instruction execution, and parses each register with the delay Δ_{ci} clock delay based on the instruction type. The hourly delay is calculated on each node based on the instructions, and each computation is handled with an overhead (φ_i). Processing is defined as the sum of all instructions (i = 1… n) that will be determined by the overhead.

$$\varphi_i = \sum_{i=n} O_i + \gamma_i \quad (1)$$

where,γ_i is the processing overhead for the instruction execution and O_i is the clock allocation overhead for i^{th} instruction, which is the aggregated value of clock delay computation (dl_i) and clock allocation overhead (ao_i).

Where, γ_i is the upper processing command and O_i is the upper clock assignment of the ith command, which is the number of clock delay (dl_i) and the general clock assignment (ao_i).

$$\gamma_i = \sum_{i=n} dl_i + ao_i \quad (2)$$

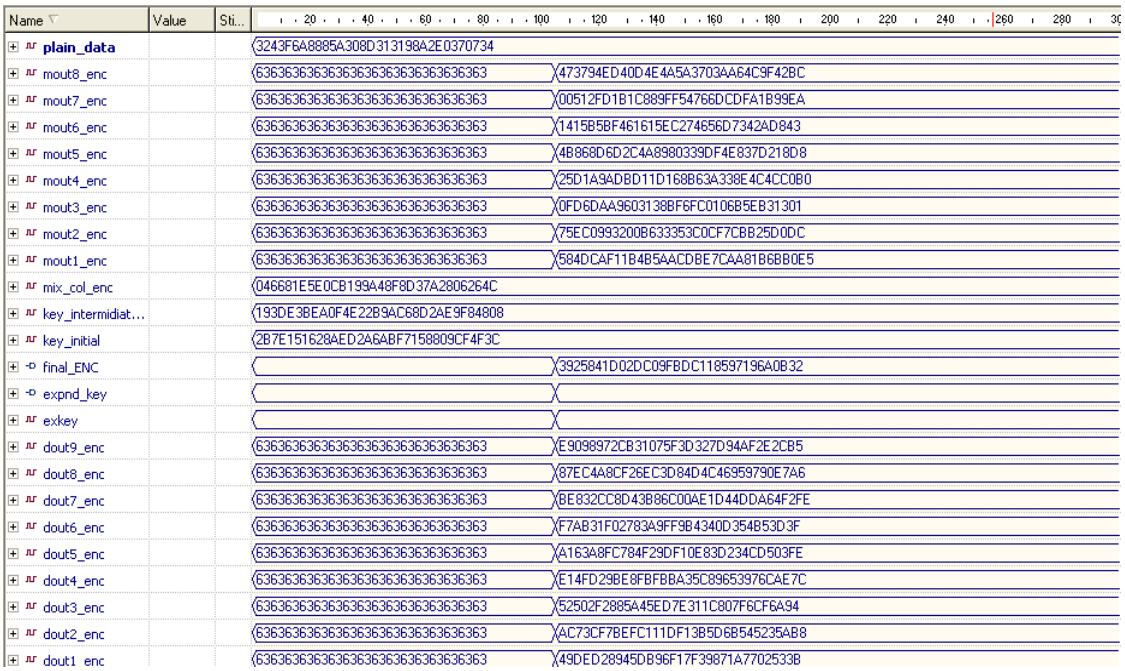
This method reduces the number of hours delay from 15 to 12, and thus reduces the amount of overload. For the given example, the log alignment logic is reduced by 7 overhead cycles as the log is realized. It has been observed that the order of instructions is maintained by switching address pointers however the recommended clock allocation process allocates a continuous clock to each request. However, arithmetic delay is eliminated, switching delay is observed to eliminate this delay, a new instruction-based register alignment using index mapping is proposed. These functions are intended to be used in the best possible way in security encryption the results of the process for the developed method are described below

V. Simulation results

To evaluate the proposed method, the job description is defined using the HDL definition. The timing simulation results of the proposed method are observed as shown in the figure below.

a)

A 128-bit block of data is considered for the, Designed system given as
Plain Text : "3243f6a8885a308d313198a2e0370734".
The Initial Key considered is “2b7e151628aed2a6abf7158809cf4f3c”
The encrypted output is “3925841d02dc09fbdc118597196a0b32”
The encrypted output is a input to the decryption.
After the Whole Process The Obtained Decrypted data is given as
DEC_output: "3243f6a8885a308d313198a2e0370734"
This Shows That the Data is Recovered exactly as Input (Plaintext) . The Intermediate Keys ,subbytes output ,shiftrows Output and mixcolumn outputs .



DOI: 10.9756/INT-JECSE/V14I4.164

Fig 4. Simulation Results of Secure coding

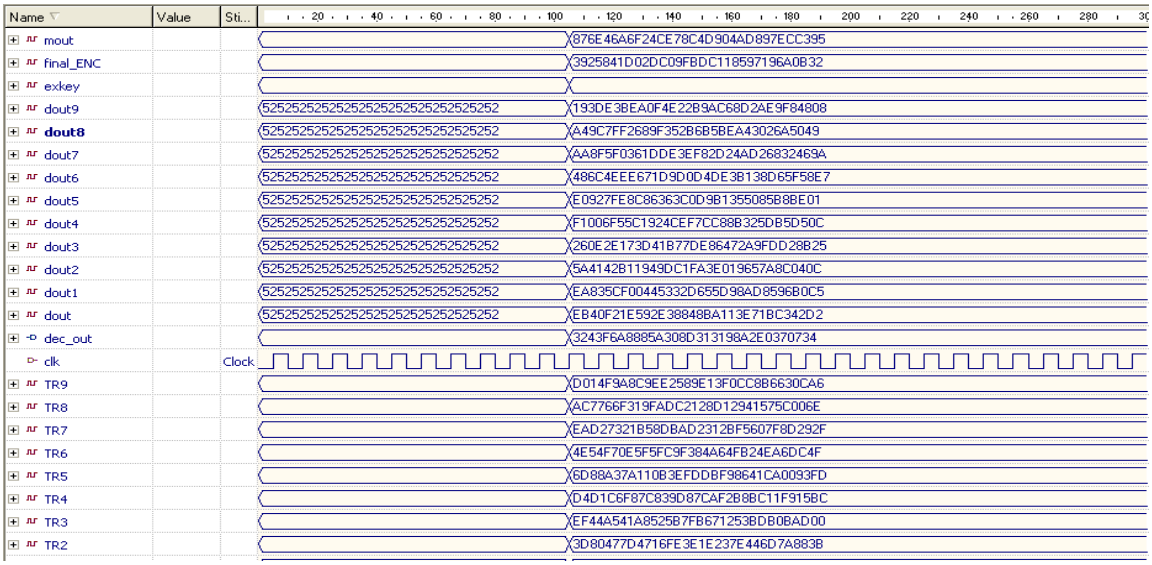


Fig 5. Simulation Results of Decoding

b)

A 128-bit block of data is considered for the, Designed system given as
Plain Text : "00112233445566778899aabbccddeeff".
The Initial Key considered is “000102030405060708090a0b0c0d0e0f”
The encrypted output is “69c4e0d86a7b0430d8cdb78070b4c55a”
The encrypted output is a input to the decryption.
After the Whole Process The Obtained Decrypted data is given as
DEC_output: "00112233445566778899aabbccddeeff "

DOI: 10.9756/INT-JECSE/V14I4.164

The Data is Recovered exactly as Input (Plaintext) . The Intermediate Keys ,subbytes output ,shiftrows Output and mixcolumn outputs

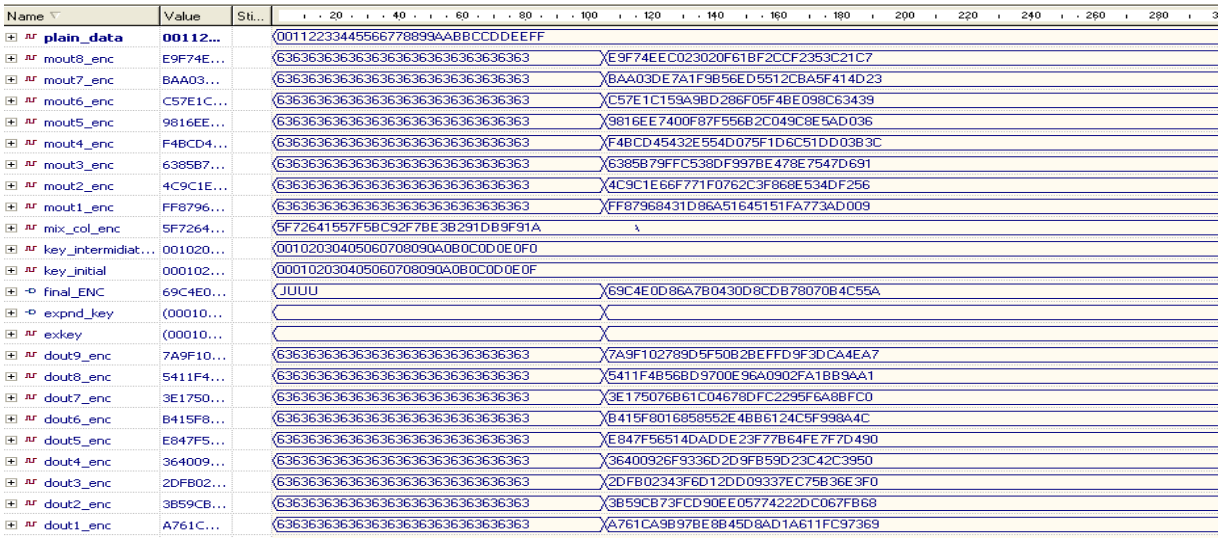


Fig 6. Simulation Results of secure coding

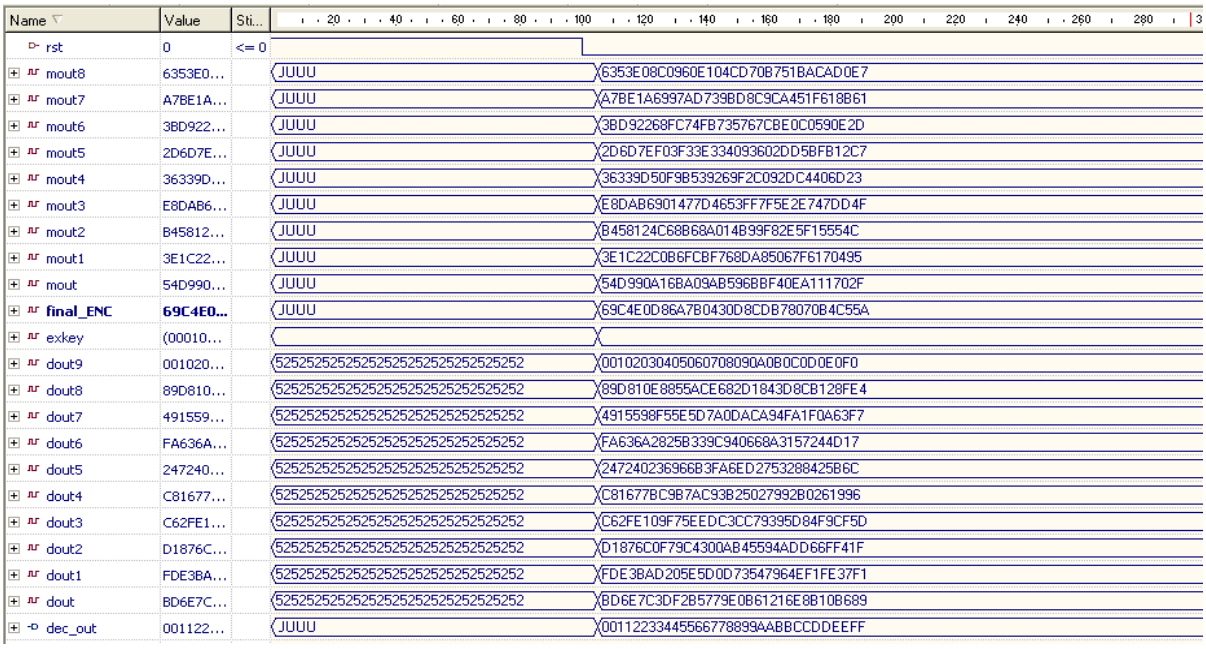


Fig 7. Simulation Results of Decryption

DOI: 10.9756/INT-JECSE/V14I4.164

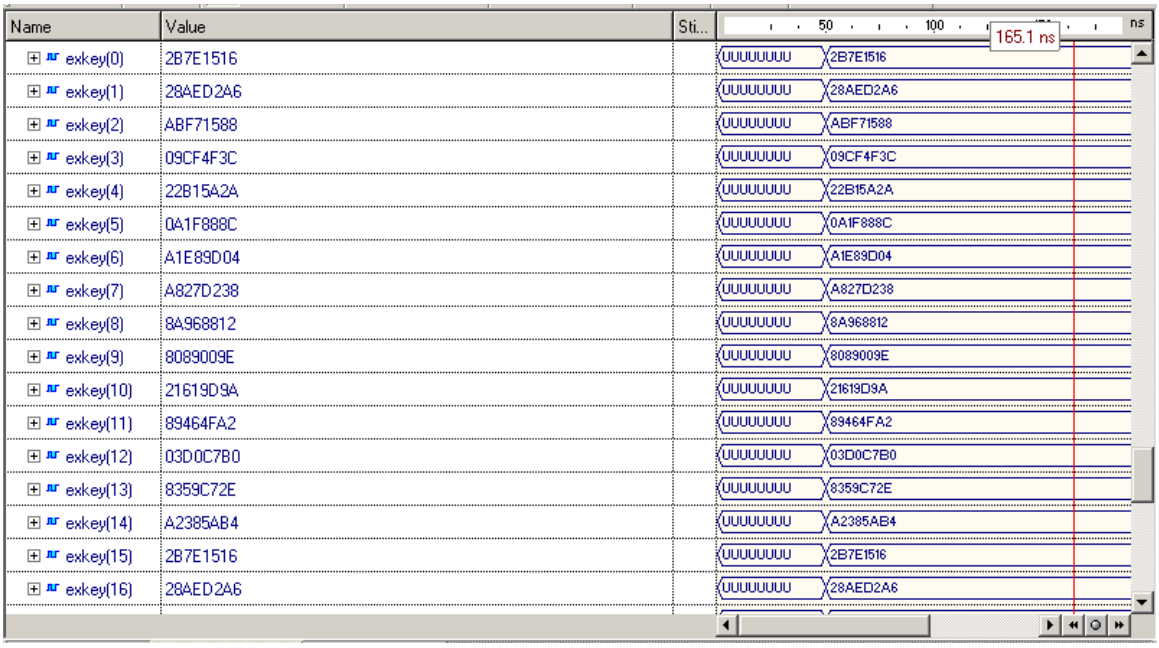


Fig 8 expanded key waveforms from 0 to 16

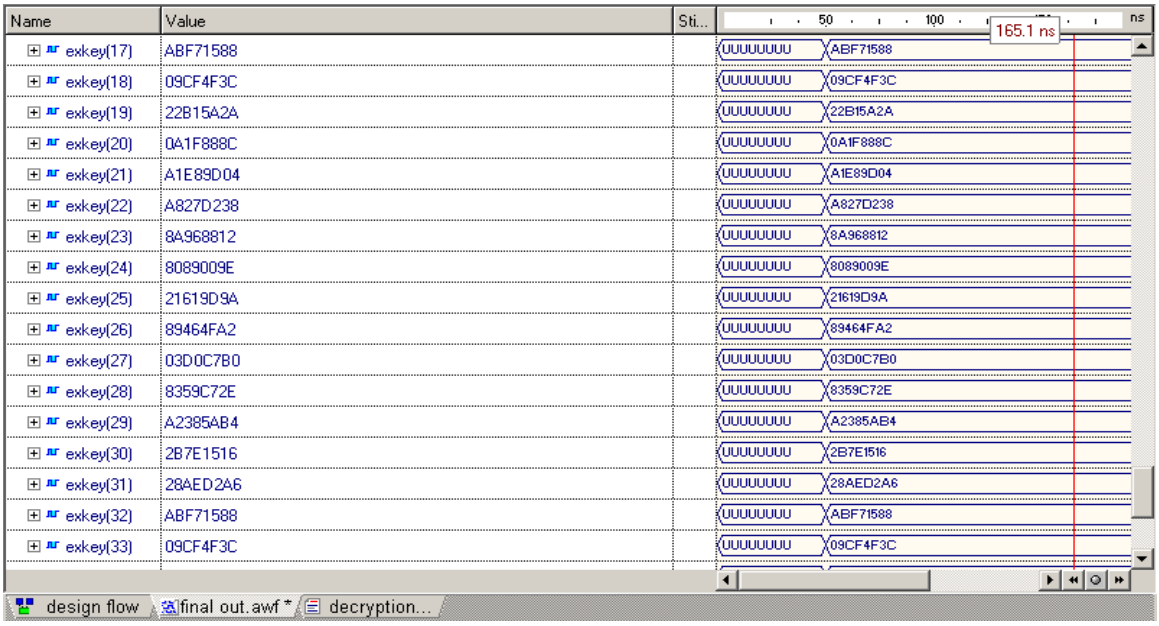


Fig 9 expanded key waveforms from 17 to 33

DOI: 10.9756/INT-JECSE/V14I4.164

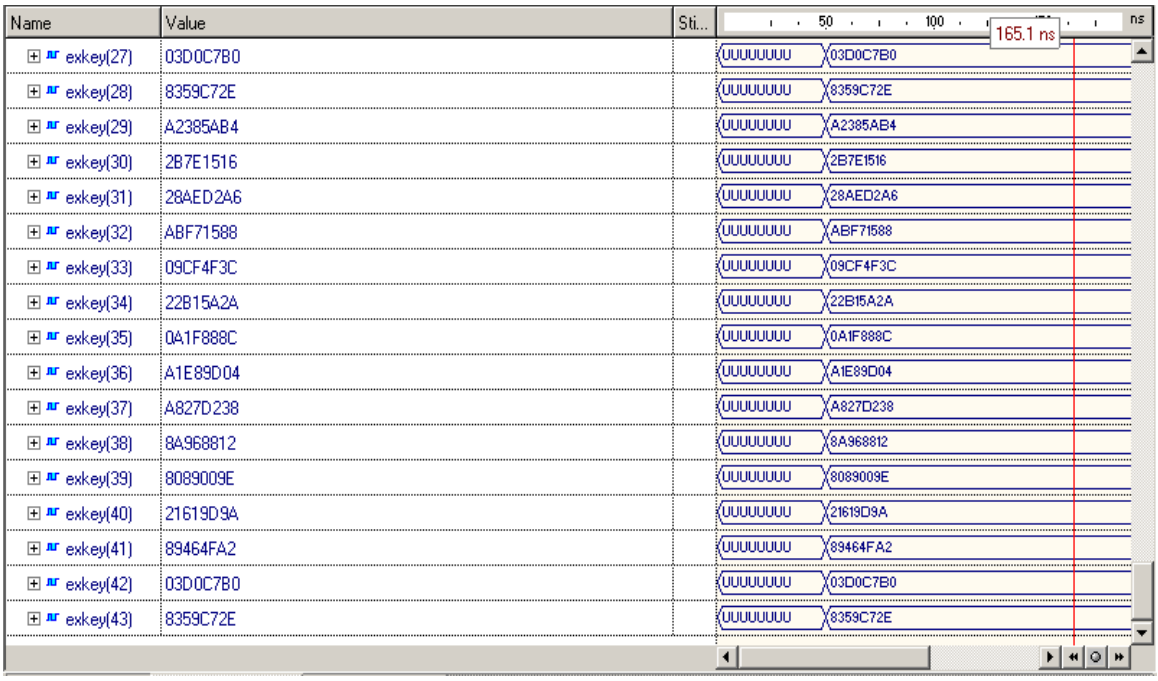


Fig 10 expanded key waveforms 27 to 43

In order to understand the evolving approach on the FPGA target, the developed method was fabricated by targeting Xilinx FPGA devices. A synthesis summary of the proposed approach is shown in Figure 11

Device Utilization Summary (estimated values)			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slices	99	1920	5%
Number of Slice Flip Flops	156	3840	4%
Number of 4 input LUTs	174	3840	4%
Number of bonded IOBs	149	173	86%
Number of GCLKs	1	8	12%

Figure 11. Summarized synthesis report for the developed system.

The power analysis of the system developed on the target FPGA is obtained using the x power analyzer of the Xilinx device. A total of 123mW of power is used for the method developed on the target FPGA.

DOI: 10.9756/INT-JECSE/V14I4.164

The implementation layout on the target FPGA is shown in Figure 12. The RTL is observed for the developed approach with the logical block and its interconnection. The developed system is built on 8 logical units and 14 I/Os.

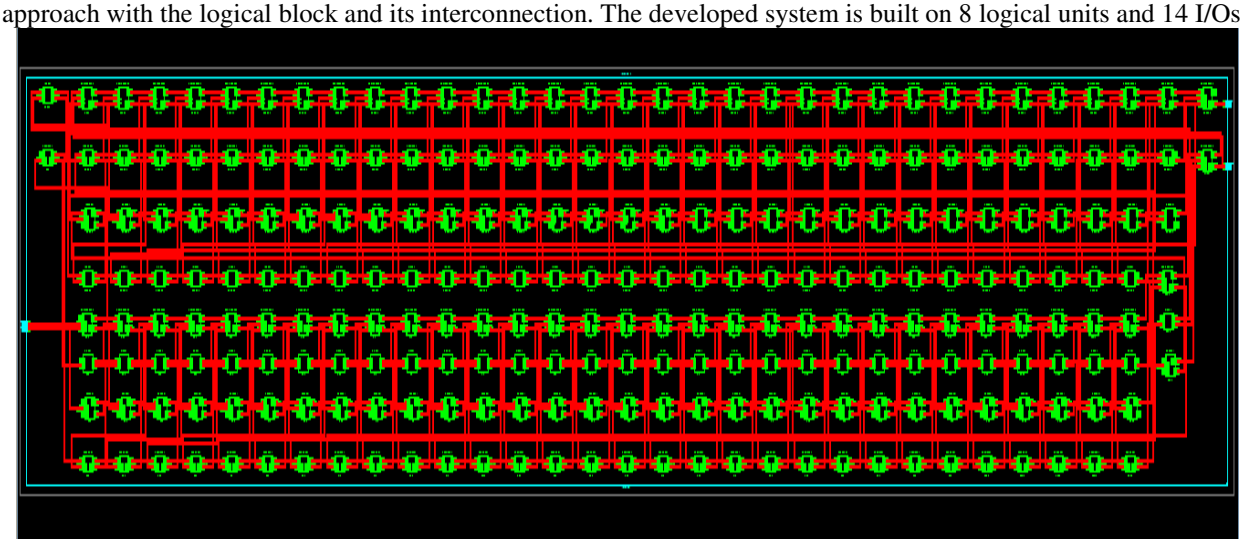


Figure 12: RTL implementation of the FPGA unit for the proposed system.

To evaluate the performance of the system developed on different FPGA devices, the power logic of the Xilinx ISE device is tested using power logic, and the three models are designed as linear (uncontrolled), electronic block, etc. The power and speed is compared as shown in figure 13, 14 respectively. The parameters for the Xilinx Spartan and Vertex FPGA family devices are evaluated. The obtained observations are listed in Table 1, 2.

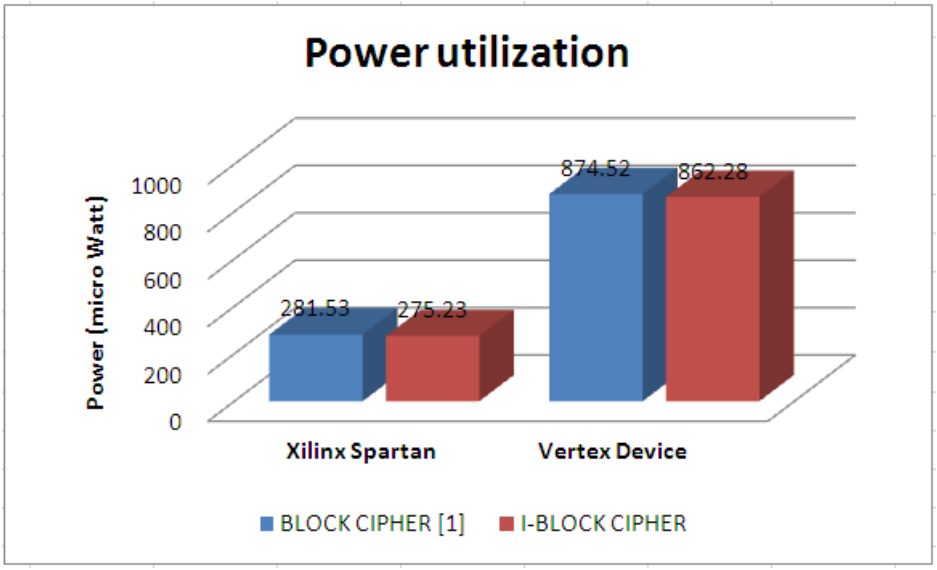


Figure 13: Power utilization for developed model for different FPGA device

Table 1: Power Consumption Analysis

Circuit	**Power (μW)**	
	BLOCK CIPHER [1]	**I-BLOCK CIPHER**
Xilinx Spartan	281.53	275.23

DOI: 10.9756/INT-JECSE/V14I4.164

Vertex family		874.52	862.28

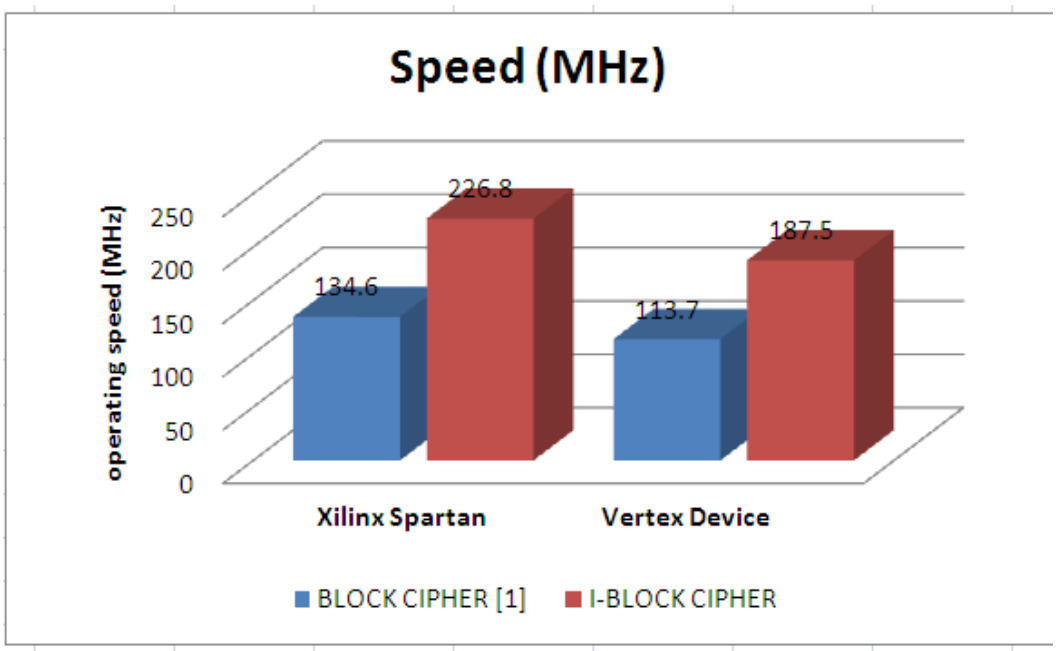


Figure 14: Operational speed for developed model for different FPGA device

Table 2: Delay latency

Circuit	**Speed (MHz)**		
		BLOCK CIPHER [1]	**I-BLOCK CIPHER**
Xilinx Spartan		134.6	226.8
Vertex family		113.7	187.5

VI. Conclusion

In the development of security coding with the use of minimum resources, a new source optimization has been proposed that improves optimization by applying a domain-specific arithmetic method to reduce computation time for message encoding with block sippers. Observations for the implemented Block Cipher Module show that the chip can support 128-bit chips for secure encryption with real-time speeds of 71.8MHz. The target Virtex2p chip shows a faster power processing for the proposed I-Block cipher compared to existing block cipher. A source-optimized secure cyber module is implemented. It can be used for applications that provide high-speed remote device protection.

VII. References

[1] Mars, Ayoub, Hussam Ghandour, and Wael Adi. "SRAM-SUC: Ultra-Low Latency Robust Digital PUF." arXiv preprint arXiv:2106.07105, 2021.

[2] X. Xu and W. Burleson, “Hybrid side-channel/machine-learning attackson pufs: A new threat?” in Proceedings of the conference on Design,Automation & Test in Europe. European Design and AutomationAssociation, 2014, p. 349.

[3] T. Kim, B. Choi, and D. K. Kim, “Zero bit error rate id generation circuitusing via formation probability in 0.18 _m cmos process,” Electronicsletters, vol. 50, no. 12, pp. 876–877, 2014.

DOI: 10.9756/INT-JECSE/V14I4.164

[4] D. Jeon and B.-D. Choi, “Circuit design of physical unclonable functionfor security applications in standard cmos technology,” in 2016 IEEEInternational Conference on Electron Devices and Solid-State Circuits(EDSSC). IEEE, 2016, pp. 86–90.
[5] Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J.Robshaw, Y. Seurin, and C. Vikkelsoe, “Present: An ultra-lightweightblock cipher,” in International workshop on cryptographic hardware andembedded systems. Springer, 2007, pp. 450–466.
[6] W. Adi and A. Mars, “Clone-resistant structures in microsemi soc units,”CryptArchi, 2017.
[7] A. Mars and W. Adi, “Digitally mutating nv-fpgas into physically cloneresistantunits,” arXiv preprint arXiv:1908.03898, 2019.
[8] A. Mars, W. Adi, S. Mulhem, and E. Hamadaqa, “Random stream cipheras a puf-like identity in FPGA environment,” in 2017 Seventh InternationalConference on Emerging Security Technologies (EST). IEEE, 2017, pp.209–214.
[9] W. Adi and A. Mars, “Physical and mechatronic security, technologiesand future trends for vehicular environment,” arXiv preprintarXiv:1805.07570, 2018.
[10] D. Lim, J. W. Lee, B. Gassend, G. E. Suh, M. Van Dijk, and S. Devadas,“Extracting secret keys from integrated circuits,” IEEE Transactions onVery Large Scale Integration (VLSI) Systems, vol. 13, no. 10, pp. 1200–1205, 2005.
[11] M. Majzoobi, F. Koushanfar, and M. Potkonjak, “Testing techniques forhardware security,” in 2008 IEEE International Test Conference. IEEE,2008, pp. 1–10.
[12] U. R¨uhrmair, J. S¨olter, F. Sehnke, X. Xu, A. Mahmoud, V. Stoyanova,G. Dror, J. Schmidhuber, W. Burleson, and S. Devadas, “Puf modelingattacks on simulated and silicon data,” IEEE Transactions on InformationForensics and Security, vol. 8, no. 11, pp. 1876–1891, 2013.
[13] S. Mathew, S. Satpathy, V. Suresh, M. Anders, H. Kaul, A. Agarwal, S. Hsu, G. Chen, and R. K. Krishnamurthy, ``340 mv_1.1 v, 289 gbps/w, 2090-gate nanoaes hardware accelerator with area-optimized encrypt/decrypt gf (2 4) 2 polynomials in 22 nm tri-gate CMOS," IEEE J. Solid-State Circuits, vol. 50, no. 4, pp. 1048_1058, Apr. 2015.
[14] S. Banik, A. Bogdanov, and F. Regazzoni, ``Compact circuits for combined AES Secure coding/decryption," J. Cryptograph. Eng., Vol. 9, No. 1, pp. 69_83, Apr. 2019.
[15] 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. Eurocrypt, Tallinn, Estonia, May 2011, pp. 69_88.
[16] V. Hoang, V. Dao, and C. Pham, ``Design of ultra low power AES Secure coding cores with silicon demonstration in SOTB CMOS process," Electron. Lett., vol. 53, no. 23, pp. 1512_1514, Nov. 2017.
[17] A. Shreedhar, K.-S. Chong, N. K. Z. Lwin, N. A. Kyaw, L. Nalangilli, W. Shu, J. S. Chang, and B.-H. Gwee, ``Low gate-count ultra-small area nano advanced Secure coding standard (AES) design," in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS), May 2019, pp. 1_5.
[18] P. Choi, J.-H. Kim, and D. K. Kim, ``Fast and power-analysis resistant ring lizard crypto-processor based on the sparse ternary property," IEEE Access, vol. 7, pp. 98684_98693, 2019.

DOI: 10.9756/INT-JECSE/V14I4.164