Technical Sciences
Academy of Romania
www.jesi.astr.ro

Journal of Engineering Sciences and Innovation
Volume 9, Issue 3 / 2024, pp. 343-354

F. Electrical, Electronics Engineering, Computer Sciences and Engineering

Received ***10 May 2024*** *Accepted* ***3 September 2024***
Received in revised form ***12 July 2024***

An efficient algorithm and architecture for the VLSI implementation of type IV DST using short quasi-band correlation structures allowing efficient incorporation of techniques used for hardware security

LAURA TEODORA COTOROBAI[1*], DORU FLORIN CHIPER[1,2,]

[1] Applied Electronics, Technical University Gheorghe Asachi, 43 D. Mangeron Blvd, 700050, Iasi, Romania,
[2]Technical Sciences Academy of Romania (ASTR), 26 Dacia Blvd, 030167, Bucharest, Romania

Abstract. This paper addresses a critical issue in VLSI chip design, specifically the integration of hardware security techniques. It introduces an improved algorithm for the efficient VLSI implementation of Type IV Discrete Sine Transform (DST-IV) using short quasi-band correlation structures. The proposed algorithm is restructured to facilitate the incorporation of key hardware security techniques with low overheads. This integration ensures the security and integrity of the VLSI implementation while maintaining high performance. By leveraging the regular and modular structure of the quasi-band correlation, the architecture aligns with the principles of systolic array architecture, offering advantages in speed and hardware complexity. The proposed method achieves a secure, high-speed, and low-complexity VLSI implementation, making it a robust solution for modern hardware design challenges.

Keywords: VLSI algorithms, discrete transforms, discrete sine transforms, systolic arrays, hardware security.

1. Introduction

Recent, multimedia applications require processing of a huge amount of data at high speed. Moreover many embedded devices are using resource constrained requirements. Some examples of such applications are for example image and

[*]Correspondence address: cotorobai.laura@gmail.com

video via the Internet, digital libraries and telemedicine. This area of research is one of the main beneficiaries of new researches in electronics especially in VLSI technology.

With the advancement of the multimedia applications real-time implementation that support high requirements that use the VLSI technology using FPGA or ASIC has become necessary.

One important algorithm used in data compression is the type IV discrete sine transform (DST IV) [1]. DST IV together with DCT IV have some important applications as: data compression and transmission, spectral analysis and in efficient implementation of DST and DCT. [2]-[6]. DCT IV and DST IV are linked with some other fractional transforms as generalized Fourier transform or generalized Hartley transform.

The above mentioned algorithms are computational intensive and in order to be applied in real-time applications it is necessary to use VLSI implementations but in order to obtain efficient such implementations it is necessary to appropriately reformulate these algorithms [7]..

In order to efficiently reformulate such algorithms a special attention should be afforded to the data flow in the algorithm. This is the reason why using regular and modular computational structures as cycle convolution and circular correlation leads to efficient VLSI implementations [8]-[15] using systolic arrays [16] or distributed arithmetic [17].

In this paper, we are showing that another computational structure called pseudo-band correlation can be used to obtain an efficient VLSI implementation using systolic arrays. Moreover, it can be used to obtain an efficient incorporation of hardware security techniques. Using the proposed VLSI algorithm an efficient VLSI architecture for type IV DST can be obtained with high performances both as high-speed using pipelining and parallelism and a low hardware complexity. Moreover, a low I/O cost can be obtained avoiding the so called speed bottle neck that limits the speed performances of systolic arrays.

2. Modified algorithm presentation

The 1-D type IV Discrete Sine Transform (DST-IV) is defined, for a real input given sequence $x(i): i = 0,1,..., N-1$, as:

$$Y(k) = \sqrt{\frac{2}{N}} \cdot \sum_{i=0}^{N-1} x(i) \cdot \sin[(2i+1)(2k+1)\alpha/2] \quad (1)$$

for k=0, 1,…,N-1

and

$$\alpha = \frac{\pi}{2N} \quad (2)$$

In the initial step, we are removing the constant coefficient from Equation (1) that is used for scaling of the output sequence and we are adding a multiplier at the end of the VLSI architecture to include it at the end of the design.

We are using the efficient VLSI algorithm proposed by us in [18].In this VLSI algorithm in order to reformulate Equation (1), we introduce some restructuring sequences for both input and output, along with the necessary permutations. Using these we obtain a parallel decomposition of the algorithm using the desired computation structure called quasi-band correlation.

The output sequence $\{Y(k): k=1,2,...,N-1\}$ will be determined by utilizing the below equation:

$$Y(k) = x_p(0)\sin[(2k+1)\alpha/2] + 2T_a(k)\cdot\cos[(2k+1)\alpha/2] \tag{3}$$

for k=1,…,N-1

where the below auxiliary output sequence was used $\{T_a(k): k=1,2,...,N-1\}$

Therefore, it can be can be recursively calculated as shown the below:

$$T_a(0) = \sum_{i=0}^{N-1} x_p(i)\sin(i\alpha) = \sum_{i=0}^{N-1} x_S(i) \tag{4}$$

$$T_a(k) = T(k) - T_a(k-1) \tag{5}$$

where we have introduced the auxiliary input sequence defined as follows:

$$x_S(i) = x_p(i)\cdot\sin[i\alpha] \tag{6}$$

The auxiliary input sequence $\{x_p(i): i=0,...,N-1\}$ is recursively determined as follows:

$$x_p(N-1) = x(N-1) \tag{7}$$

$$x_p(i) = (-1)^i x(i) + x_a(i+1) \tag{8}$$

for i=N-2,…,0

Based on the above, the $\{T(k): k=1,2,...,N-1\}$ auxiliary output sequence was obtained. It can be concurrently computed using only 6 short band-correlation structures when N is a prime number. It can be seen that the elements in the matrices from the bellow equations along the secondary diagonal are the same excepting the sign but the first element in each line is not equal with the last elements from the next line as in circular correlation structure. We have called these computational structures quasi band-correlation structures. In this paper, we are specifically focusing on N, the transform length, being equal to 13

Derived from this, we have:

$$T_{1a} = \begin{bmatrix} s(4)-s(5) & -s(3)+s(6) & -(s(1)+s(2)) \\ s(3)+s(6) & (s(1)+s(2)) & s(9)-s(8) \\ -(s(1)+s(2)) & -s(9)-s(8) & s(10)+s(7) \end{bmatrix} \cdot \begin{bmatrix} x(4+9) \\ x(3+10) \\ x(1+12) \end{bmatrix} \tag{9}$$

$$T_{1b} = \begin{bmatrix} s(4) & s(3) & s(1) \\ s(3) & -s(1) & -s(9) \\ -s(1) & s(9) & -s(10) \end{bmatrix} \cdot \begin{bmatrix} x(2+11)-x(4+9) \\ x(5+8)+x(3+10) \\ x(1+12)-x(6+7) \end{bmatrix} \tag{10}$$

$$T_{1c}=\begin{bmatrix} s(3)-s(5) & -s(6)-s(1) & -(s(2)+s(4)) \\ s(6)-s(1) & (s(2)+s(4)) & s(10)-s(8) \\ -(s(2)+s(4)) & -s(10)-s(8) & s(7)-s(12) \end{bmatrix}\cdot\begin{bmatrix} x(2+11) \\ -x(5+8) \\ x(6+7)\_ \end{bmatrix} \tag{11}$$

$$\begin{bmatrix} T(4) \\ T(8) \\ T(10) \\ T(6) \\ T(12) \\ T(2) \end{bmatrix}=\begin{bmatrix} T_{1a}(1)+T_{1b}(1) \\ T_{1c}(1)-T_{1b}(2) \\ -(T_{1a}(2)+T_{1b}(2)) \\ T_{1c}(2)-T_{1b}(3) \\ T_{1a}(3)+T_{1b}(3) \\ T_{1c}(3)-T_{1b}(1) \end{bmatrix} \tag{12}$$

$$T_{2a}=\begin{bmatrix} s(4)-s(5) & s(3)+s(6) & (s(1)+s(2)) \\ s(3)+s(6) & -(s(1)+s(2)) & -(s(9)-s(8)) \\ -(s(1)+s(2)) & s(9)-s(8) & -(s(10)+s(7)) \end{bmatrix}\cdot\begin{bmatrix} x(4-9) \\ x(3-10) \\ -x(1-12) \end{bmatrix} \tag{13}$$

$$T_{2b}=\begin{bmatrix} s(4) & s(3) & s(1) \\ s(3) & -s(1) & -s(9) \\ -s(1) & s(9) & -s(10) \end{bmatrix}\cdot\begin{bmatrix} x(2-11)-x(4-9) \\ x(5-8)+x(3-10) \\ x(1-12)-x(6-7) \end{bmatrix} \tag{14}$$

$$T_{1c}=\begin{bmatrix} s(3)-s(5) & -s(6)-s(1) & -(s(2)+s(4)) \\ s(6)-s(1) & (s(2)+s(4)) & s(10)-s(8) \\ -(s(2)+s(4)) & -s(10)-s(8) & s(7)-s(12) \end{bmatrix}\cdot\begin{bmatrix} x(2-11) \\ -x(5-8) \\ x(6-7)\_ \end{bmatrix} \tag{15}$$

$$\begin{bmatrix} T(9) \\ T(5) \\ T(3) \\ T(7) \\ T(1) \\ T(11) \end{bmatrix}=\begin{bmatrix} T_{2a}(1)+T_{2b}(1) \\ T_{2c}(1)-T_{21b}(2) \\ -(T_{2a}(2)+T_{2b}(2)) \\ T_{2c}(2)-T_{2b}(3) \\ T_{2a}(3)+T_{2b}(3) \\ T_{2c}(3)-T_{2b}(1) \end{bmatrix} \tag{16}$$

In the above, the following notations were considered:

$$x(i+j)=x_C(i)+x_C(j) \tag{17}$$

$$x(i-j)=x_C(i)-x_C(j) \tag{18}$$

with

$$x_C(i)=x_p(i)\cdot\cos[i\alpha] \tag{19}$$

and

$$s(i)=2\cdot\sin(2i\alpha) \tag{20}$$

The obtained computations were reorganized employing the following permutations:

$$\varphi(k)=\begin{cases} <g^{k}>_{N} & if\ k>0 \\ <g^{N+k-1}>_{N} & otherwise \end{cases} \tag{21}$$

$$\zeta(k)=<2k>_{N} \tag{22}$$

For the Galois field formed by the transform index, we considered the primitive root g=3 which was used in the above.

3. Hardware security techniques

Before to introduce the hardware security techniques in our design we will review these techniques.

In an era where cyber threats are ever-evolving, safeguarding the physical components of computing systems, known as hardware security, is critical for maintaining the integrity, confidentiality, and availability of information technology (IT) systems. Hardware security encompasses the strategies, defense mechanisms, and physical constructs designed to resist tampering and ensure that hardware components function as expected, even under attack. It plays a key role in protecting intellectual property (IP). Hardware security techniques are methods or strategies employed to protect devices and their components from physical tampering and external threats, ensuring the integrity and confidentiality of the data they process or store. Unlike hardware security technologies, which are specific implementations or devices (such as Trusted Platform Modules (TPM) or Hardware-Based Full-Disk Encryption (FDE)), techniques can be broader strategies or approaches to securing hardware. The literature presents key hardware security techniques:

Physical Unclonable Functions (PUFs) [19] leverage the inherent physical variations found in hardware components to create unique identifiers or cryptographic keys. These physical variations are intrinsic to the manufacturing process and are nearly impossible to duplicate, providing a robust method for securing devices against unauthorized access and cloning. PUFs can be employed in various applications, including secure key storage, device authentication, and cryptographic key generation. The randomness and uniqueness of PUF responses make them ideal for generating cryptographic keys that are inherently tied to the physical properties of the hardware, thus providing a high level of security. The implementation of PUFs in hardware security frameworks offers a low-cost and effective solution for enhancing device security.

Tamper detection and response mechanisms [20] are critical in designing secure hardware systems. These mechanisms involve embedding sensors and detection circuits within the hardware to monitor for any physical tampering attempts. For instance, opening the casing of a device, altering its components, or attempting unauthorized physical access can trigger these detection systems. Upon detecting tampering, the hardware can automatically initiate protective responses such as erasing sensitive data, shutting down the device, or locking the system to prevent further unauthorized access. These responses are crucial in safeguarding sensitive information and ensuring that the hardware remains secure even when physically

compromised. Implementing tamper detection and response mechanisms enhances the resilience of hardware systems against physical attacks.

Secure Boot [21] is a process that ensures the integrity and authenticity of a device's firmware and software during the boot sequence. This technique involves cryptographic checks to verify that only trusted and authorized software is loaded. Secure Boot begins with a root of trust, typically embedded in hardware, which verifies the bootloader, and subsequently, each stage of the boot process verifies the next stage. If any unauthorized modifications are detected at any point during the boot process, the device can halt the boot sequence or restrict access, thereby preventing potentially malicious code from executing. Secure Boot is essential for preventing rootkits and other persistent threats that can compromise the integrity of a system from the initial start-up phase.

Hardware encryption [22] involves implementing encryption algorithms directly within hardware components to secure data at rest, in use, or in transit. By integrating cryptographic functions into the hardware, this approach offers enhanced performance and security compared to software-based encryption. Hardware encryption can secure data without exposing encryption keys. This paper addresses a critical issue in VLSI chip design, specifically the integration of hardware security techniques. It introduces an improved algorithm for the efficient VLSI implementation of Type IV Discrete Sine Transform (DST-IV) using short quasi-band correlation structures. The proposed algorithm is restructured to facilitate the incorporation of key hardware security techniques, including Physical Unclonable Functions (PUFs), tamper detection and response, secure boot, hardware encryption, and side-channel attack protection. This integration ensures the security and integrity of the VLSI implementation while maintaining high performance.

In the context of hardware security, obfuscation [23] refers to the practice of designing circuits and components in a manner that conceals their true functionality. This technique makes it difficult for attackers to reverse-engineer the hardware or understand its operational mechanisms. By obfuscating the design, manufacturers can protect proprietary algorithms, data processing techniques, and intellectual property from being copied or tampered with. Hardware obfuscation can involve various methods, such as hiding critical paths, using dummy logic gates, and employing complex wiring schemes. This added layer of complexity increases the effort required for an attacker to analyze and compromise the hardware, thus enhancing the overall security of the system.

Side-channel attack [24] protection techniques are designed to defend against attacks that exploit physical emanations from hardware during operation, such as power consumption, electromagnetic radiation, or acoustic signals. These attacks can reveal sensitive information, such as cryptographic keys, by analyzing the physical behavior of the hardware. Protection against side-channel attacks involves implementing countermeasures like noise generation, masking, balancing power consumption, and randomizing operational timing. These techniques obscure the correlation between the physical emanations and the sensitive information, making

it significantly harder for attackers to derive useful data. Effective side-channel attack protection is essential for maintaining the confidentiality of cryptographic operations and other sensitive processes.

Redundancy and fault tolerance [25] involve designing hardware systems with redundant components and architectures that can continue to operate correctly even when parts of the system fail. This technique is crucial for ensuring the reliability and availability of critical systems, particularly in environments where uninterrupted operation is essential. Redundant components can take over the functions of failed parts, while fault-tolerant architectures are designed to detect and correct errors without disrupting system performance. This approach enhances the system's resilience to both accidental failures and malicious attacks, ensuring continuous and reliable operation. Implementing redundancy and fault tolerance is a key strategy for maintaining the dependability of mission-critical hardware systems.

Access control and authentication mechanisms are fundamental for ensuring that only authorized systems or individuals can access or use hardware. These mechanisms often involve the use of cryptographic keys, secure communication protocols, or biometric verification to authenticate

4. Hardware design of the new modified algorithm

The proposed algorithm and architecture allow for an efficient computation of the DST-IV on the same VLSI chip, requiring minimal modifications. By leveraging the regular and modular structure of the quasi-band correlation, the architecture aligns with the principles of systolic array architecture, offering advantages in speed and hardware complexity. The proposed method achieves a secure, high-speed, and low-complexity VLSI implementation, making it a robust solution for modern hardware design challenges.

The VLSI algorithm uses six short quasi-band correlation structures that have a similar structure and the same length and can be computed in parallel. Thus we can obtain high speed performances using parallelism and pipelining techniques and an efficient VLSI implementation using systolic arrays. The proposed VLSI algorithm has been implemented using the systolic array paradigm that allows obtaining high speed performances at a low hardware complexity. The 6 quasi-band correlation structures can be mapped to 6 linear systolic arrays and represent the hardware core for the VLSI implementation of the proposed algorithm. In Fig.1 it is presented the VLSI architecture which is formed from 6 systolic arrays.

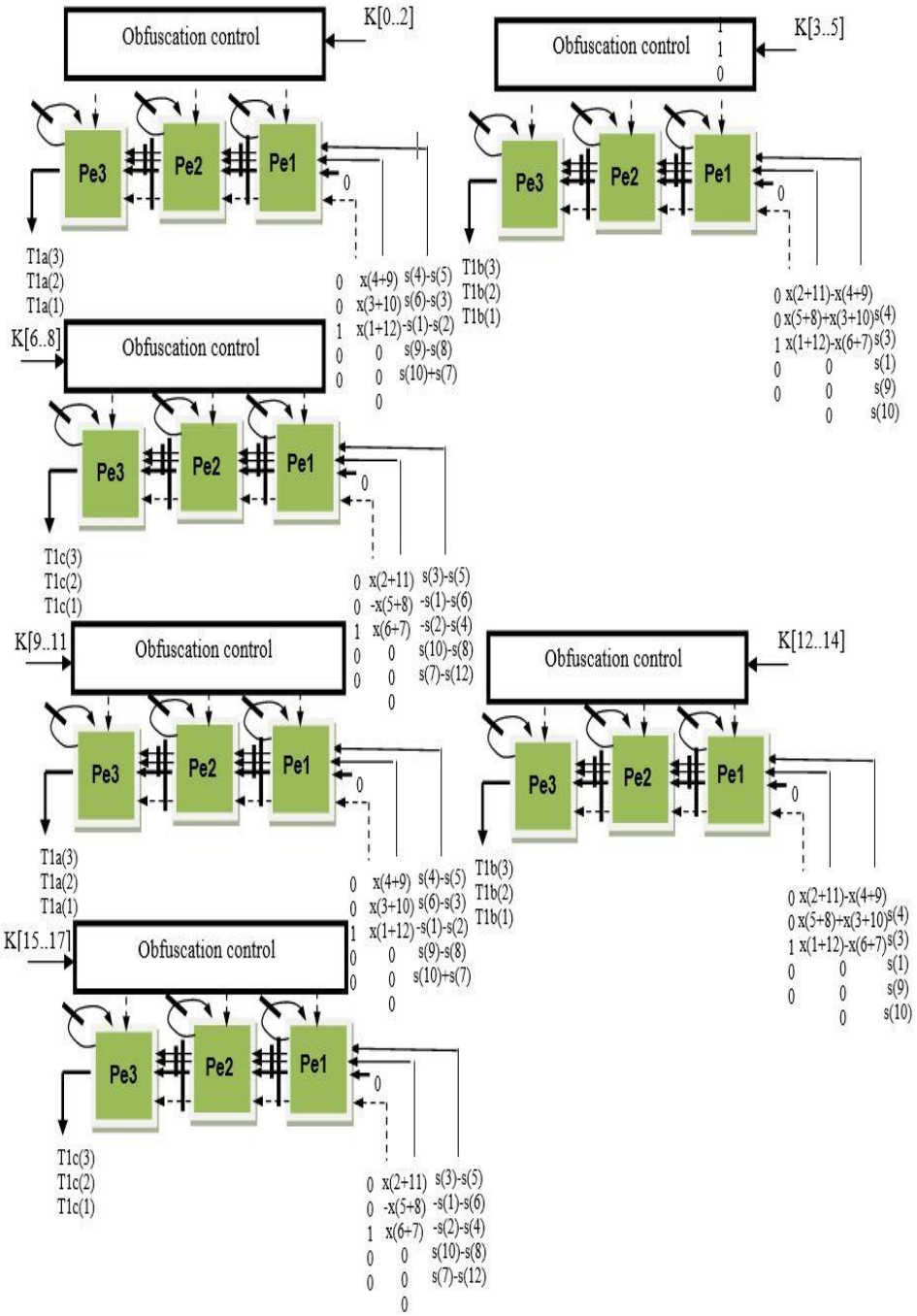
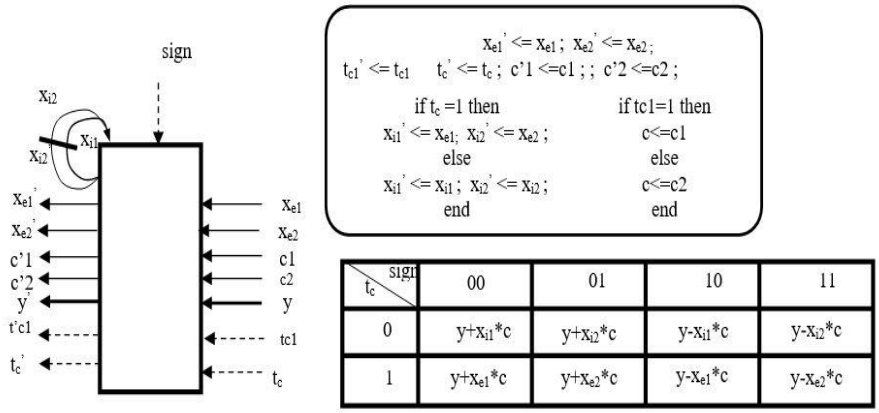


Fig.1. The systolic array for the hardware core of the VLSI implementation of DST IV.

There are 3 processing elements (PE) in each systolic array and the functioning of a PE is shown in Fig. 2.



t_c \ sign	00	01	10	11
0	$y+x_{i1}*c$	$y+x_{i2}*c$	$y-x_{i1}*c$	$y-x_{i2}*c$
1	$y+x_{e1}*c$	$y+x_{e2}*c$	$y-x_{e1}*c$	$y-x_{e2}*c$

Fig. 2. The function of the processing elements PE from Fig. 1 [18].

From Fig.2 we can see that for each processing element PE has and adder, a multiplier and some multiplexers. Using control tag bits, the input data is loaded in each PE from the links at the one end of the liner array. The control tag tc is used to store the input data in each PE. Another control tag "sign" is used for selecting the right sign for the constant multiplier as is shown in Fig.2. We also need a pre-processing and post-processing stages besides the systolic arrays that represent the hardware core. In the pre-processing stage we compute the auxiliary input data using equations (6)-(8) and we reorder using some permutations in order to obtain the desired form. In the post-processing stage we compute the auxiliary output data using equations (5),(12) and (16).

We are using only 6 quasi-band correlation structures instead of 8 as in [26]. Thus the number of processing elements has been reduced from 24 to 18 with no additional overheads and also the number of multipliers while maintaining the same throughput as in [26].

Moreover, as will be explain bellow the obtained VLSI architecture can efficiently incorporate the obfuscation technique and has a good topology with local and regular interconnections that allow an efficient VLSI implementation.

The quasi-band correlation structures have the same dimension, similar structures and due to this we can achieve a reduction of the I/O cost and subsequently of the hardware itself thus leading to an efficient VLSI implementation. This is because the structures are modular and regular, with a simple data-flow within the algorithm which gives good topology of the architecture alongside local and small interconnection links.

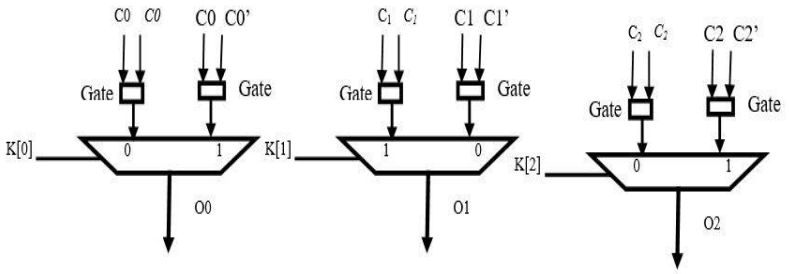


Fig. 3. Obfuscation control circuit for the first systolic array from Fig.1.

In Fig. 3 it is presented the circuit that implements the obfuscation technique. It consists of 3 MUXs controlled by the three bits of the obfuscation key K[0..2] and 6 AND gates that combine the sign bits for each processing element. The correct sign bits for the first systolic array are as follows: [1 0 0] for the Pe1, [1 0 1] for Pe2 and [0 0 1] for Pe3. The correct obfuscation Key for the first systolic array is K[0..2]=[0 1 0]. C1,C2 and C3 are the correct sign bits for the three Processing Elements. The correct bit of the obfuscation Key for the first Pe is 0. So, the AND gate from the left is selected. The first gate combine the correct sign sequence C0=[1 0 0] with the same sign bits given by the sequence C0 but the second AND gate from the right combine the right sequence C0 with the sequence C0' where the bit 1 has been changed to 0. So if the Key bit is wrong, that means 1 instead 0, the MUX select the output of the right gate. So, instead of the correct sign sequence C0=[1 0 0] we have an altered sign sequence C0'=[0 0 0] for the first Pe. So the result will be wrong. The same principle is applied for the other systolic arrays. The obfuscation Key is obtained by the concatenation of the 6 individual Keys: K[0..2], K[3:..5], K[6..8], K[9..11}, K[12..14], K[15:16]. The right sign bits are obtained only when the obfuscation key is correct. When one of the key bits is wrong the result of the VLSI array will be also wrong. The combination formed by K[0],…,K[16]represents the obfuscation key Of course, it is possible to remove the AND gates from the left but they have been introduced in order to have a symmetry of the circuits. Thus, it is not possible to obtain the correct sign sequences by reverse engineering.

5. Conclusions

This paper has addressed a critical issue in VLSI chip design for embedded designs used in common goods specifically the integration of hardware security techniques with very low overheads. Based on an improved VLSI algorithm for type IV DST using short quasi-band correlation structures an efficient VLSI architecture that incorporates the obfuscation technique for hardware security with very low overheads has been obtained. The proposed algorithm is restructured to facilitate the incorporation of key hardware security techniques. This integration ensures the security and integrity of the VLSI implementation while maintaining high

performance. Based on the regular and modular structure of the quasi-band correlation an efficient VLSI implementation using systolic arrays have been obtained that allows the integration of the hardware security techniques with low overheads.

References

[1] Jain A.K., *A fast Karhunen-Loeve transform for a class of random processes*, IEEE Transactions on Communications, **24**, 9, 1976, p.1023-1029.
[2] Jing C.,Tai H-M., *Fast algorithm for computing modulated lapped transform*, Electronics Letters, **37**,12, 2001, p.796-797.
[3] Malvar H.S., *Lapped transforms for efficient transform/subband coding*, IEEE Transactions on Acoustics, Speech, and Signal Processing, **38**, 6, 1990, p. 969 – 978.
[4] Malvar H.S, *Signal Processing with Lapped Transforms*, Norwood, Artech House, 1991.
[5] Britanak V., Rao K.R, *Cosine-/Sine-Modulated Filter Banks*, Cham, Springer Cham, 2018.
[6] Luo C.-H., Ma W-J., Juang W-H., Kuo S-H., Chen C-Y., Tai P-C., Lai S-C., *An ECG Acquisition System Prototype Design With Flexible PDMS Dry Electrodes and Variable Transform Length DCT-IV Based Compression Algorithm*, IEEE Sensors Journal, **16**, 23, 2016, p. 8244 - 8254.
[7] Chiper D.F., *New VLSI Algortihm for a High-Throughput Implementation of Type IV DCT*, 2016 Proceedings of the 2016 International Conference on Communications (COMM), Bucharest, Romania, 2016, p. 17-20.
[8] Meher P.K, *Systolic designs for DCT using low-complexity concurrent convolutional formulation*, IEEE Transactions on Circuits and Systems for Video Technology, **16**, 9, 2006, p.1041-1050.
[9] Chiper D.F., Ungureanu P., *Novel VLSI Algorithm and Architecture with Good Quantization Properties for a High-Throughput Area Efficient Systolic Array Implementation of DCT*, EURASIP Journal on Advances in Signal Processing, 2011, p. 639043.
[10] Chan Y-H., Siu W-C., *Generalized approach for the realization of discrete cosine transform using cyclic convolutions*, Proceedings of the 1993 IEEE Conference on Acoustic, Speeach, and Signal Processing, Miniapolis, USA, 1993, p. 277-280.
[11] Chan Y-H., Siu W-C., *On the realization of discrete cosine transform using the distributed arithmetic*, IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, **39**, 9, 1992, p. 705-712.
[12] Nikara J., Takola J., Akopian D., Saarinen J., *Pipeline architecture for DCT/IDCT* , Proceedings of the ISCAS 2001 - 2001 IEEE International Symposium on Circuits and Systems, **4**, Sydney, Australia, 2001, p. 902-905.
[13] Chiper D.F, Swamy M.N.S., Ahmad M.O., *An efficient unified framework for the VLSI implementation of a prime-length DCT/IDCT with high throughput*, IEEE Transactions on Signal Processing, **55**, 6, 2007, p. 2925 – 2936.
[14] Cheng C., Parhi K.K., *A novel systolic array structure for DCT*, IEEE Transactions on Circuits and Systems II: Express Briefs, **52**, 7, 2005, p. 366–369.
[15] Chiper D.F., Swamy M.N.S., Ahmad M.O., *An efficient systolic array algorithm for the VLSI implementation of prime length DHT*, Proceedings of the ISSCS 2005, International Symposium on Signals, Circuits and Systems vol. I, Iasi, Romania, 2005, p. 167-169.
[16] Kung H.T., *Why systolic architectures*, Computer, **15**, 1, 1982, p. 37-46.
[17] White S.A., *Applications of distributed arithmetic to digital signal processing: A tutorial review*, IEEE ASSP Magazine, **6**, 3, 1989, p. 5-19.
[18] Chiper D.F., Cotorobai L.-T., *An Improved Algorithm for an Efficient VLSI Implementation of Type IV DST using Short Quasi-Band Correlation Structures*, Proceedings of the 2022 14th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), Ploiesti, Romania, 2022, p. 1-4.
[19] Maes R., *Physically Unclonable Functions: Constructions, Properties, and Applications*, Heidelberg, Springer Berlin, Heidelberg, 2013.

[20] National Institue of Standards and Technology, *Guidelines for the Integration of Electronic Security into the Design of Intrinsically Safe Mining Equipment*, NIST Special Publication, 2019, available online at https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-52r2.pdf (Accessed 17 July 2024).
[21] Trusted Computing Group, *TCG EFI Platform Specification For TPM Family 2.0*, 2019, available online at https://trustedcomputinggroup.org/resource/tpm-library-specification (Accessed 17 July 2024).
[22] Katz, J., Lindell, Y., *Introduction to Modern Cryptography*, Boca Raton, CRC Press, 20014.
[23] Barak, B., Goldreich O., Impagliazzo R, Rudich S., Sahai A., Vadhan S., Yang K., *On the (Im)possibility of Obfuscating Programs*, Journal of the ACM(JACM), **59**, 2, 2012, p. 1-48.
[24] Kocher P., Jaffe J., Jun B., *Differential Power Analysis*, Proceedings of the 19th Annual International Cryptology Conference on Advances in Cryptology, Santa Barbara, USA, 1999, p.388–397.
[25] Johnson B.W., *Design and Analysis of Fault-Tolerant Digital Systems*, Boston, Addison-Wesley, 1989.
[26] Chiper D.F., Cotorobai L-T., *A New Approach for a Unified Architecture for Type IV DCT/DST with an Efficient Incorporation of Obfuscation Technique*, Electronics, **10**, 14, 2021, p. 1656.