IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 10, 2019, accepted November 26, 2019, date of publication December 3, 2019,
date of current version December 16, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2957300

Digital Emulation of a Versatile Memristor With Speech Encryption Application

MOHAMMED F. TOLBA[1,2], WAFAA S. SAYED[3], MOHAMMED E. FOUDA[3,4], HANI SALEH[1], (Senior Member, IEEE), MAHMOUD AL-QUTAYRI[1], (Senior Member, IEEE), BAKER MOHAMMAD[1], (Senior Member, IEEE), AND AHMED G. RADWAN[2,3], (Senior Member, IEEE)

[1]System-on-Chip (SoC) Center, Khalifa University, Abu Dhabi 127788, UAE
[2]Nanoelectronics Integrated Systems Center, Nile University, Giza 12588, Egypt
[3]Engineering Mathematics and Physics Department Faculty of Engineering, Cairo University, Giza 12613, Egypt
[4]Electrical Engineering and Computer Science Department, University of California-Irvine, Irvine, CA 92617, USA

Corresponding author: Mohammed F. Tolba (mohammed.tolba@ku.ac.ae)

This work was supported in part by the Khalifa University of Science and Technology under Award [RC2-2018-020], and in part by the Science and Technology Development Fund (STDF) under Grant 25977.

ABSTRACT Memristor characteristics such as nonlinear dynamics, state retention and accumulation are useful for many applications. FPGA implementation of memristor-based systems and algorithms provides fast development and verification platform. In this work, we first propose a versatile digital memristor emulator that exhibits either continuous or discrete behaviors, similar to valence change memories (VCM) and the electrochemical metallization memories. Secondly, the proposed memristor emulator is used to design a chaotic generator circuit utilizing the memristor's nonlinearity. Finally, the chaotic system is used to design a speech encryption engine to demonstrate its capabilities. The memristor emulator, chaotic generator, and the encryption system were implemented on Nexys 4 Artix-7 FPGA XC7A100T. The implementation results show an efficiency in throughput and hardware resources utilization compared to the previous works. In addition, the encryption system results show good performance against several perceptual, statistical attacks in addition to resistance to security attacks tests including differential attacks, NIST tests, key space analysis, mean square error (MSE), correlation, histogram and spectrogram.

INDEX TERMS Memristor, emulation, FPGA, chaotic generator, S-box, speech encryption.

I. INTRODUCTION

Memristor is a type of resistive memory technology that has some attractive characteristics and hence attracted the attention of industry and academia [1], [2]. Owning to their non-volatility, non linearity, small size and low cost, memristors have many applications such as memristor-based chaotic circuits, encryption, low-power radiation sensing, in-memory computing, Neural Networks (NNs) and neuromorphic applications [3]–[10]. In addition, memristors have become a hot topic in the area of nonlinear systems, where they are used to generate high-frequency chaotic oscillation signals. [11]. As a controllable nanoscale device, the memristor can be utilized to accelerate the development of the conventional nonlinear field [11]. The memristor switching modes are the bipolar and the unipolar memristor. In bipolar, the voltage polarities are used to switch On and Off. In unipolar, the resistance state is changed based on the voltage magnitude and not the polarity [12].

An emulation platform is needed for fast design exploration and development. Hence, different memristors emulators have been proposed in the literature. Memristor emulators mimic the behavior of the memristors based on the modeling equations, which are experimentally deduced from real devices. Memristor emulators can be implemented using either analog or digital design. Analog emulators are built based on MOSFET transistors, diodes, capacitors, resistors, operational amplifiers, and analog multipliers [13]–[16]. The drawbacks of analog emulators are the limitations on power supply voltages, sensitivity to process variations and temperature, number of elements and hardware implementation. In addition, the capacitor needed in the memristor analog model requires a large on-chip area to store the system state [17], [18]. On the other hand,

The associate editor coordinating the review of this manuscript and approving it for publication was Ludovico Minati.

digital emulator does not demand any analog components, where the system's states are stored in a register. The digital models require small area, exhibit good performance, easily programmable, re-configurable and controllable. Consequently, practical implementation of memristors' emulators favors Field-Programmable Gate Arrays (FPGAs) over analog ones [14], [15], [17]–[20].

In addition to circuit optimization, it is important to optimize the algorithm to suit the target implementation. The main contributions of this paper includes:

- Tunable and flexible MR emulator that support both discrete and continuous MR behavior. Previous emulator supports either discrete or continuous behaviors but not both [18], [21]–[23].
- The proposed memristor emulator is designed based on an optimized algorithm have a small area and decent performance when implemented on FPGAs. The proposed memristor emulator is suitable for education and research areas, where it can be used in research labs and integrated with different applications.
- The proposed memristor emulator is used to realize a new memristive chaotic generator, where different chaotic attractors can be achieved. The versatility of the proposed memristor module helps to enhance the chaotic generator which adds independent parameters to control the output range.
- A speech encryption scheme based on S-box, data substitutions, using the proposed memristor chaotic generator, and bit permutations is designed and tested against all standard attacks.
- Finally, all proposed implementations; memristor emulator, chaotic generator and encryption system, have been experimentally verified using Nexys 4 Artix-7 FPGA and compared with the previous works.

This paper is organized as follows; Section II presents the state of the art works including digital memristor, memristor chaotic system and encryption system. Section III introduces the proposed generic discrete and continuous memristor models. In Section IV, a memristor-based chaotic oscillator is presented. Section V introduces memristor chaotic oscillators based speech encryption engine. Section VI validates the proposed circuits experimentally. Finally, the paper is concluded in Section VII.

II. PREVIOUS WORKS

Much research work developed digital memristor emulators such as [18], [21], [22], [24]. In [22], a digital memristor has been presented, which can be used as an independent block such as DSP blocks in FPGAs. The implementation was designed based on the theory of Hewlett-Packard (HP) memristor. Another attempt to build a digital memristor emulator based upon the voltage-controlled threshold-type bipolar memristor was introduced in [18] for ANNs. In addition, the implementation of digital charge- or flux-controlled memristor emulator was developed based on wave digital visualization for neuromorphic circuits [24]. Different research works focused on the implementation of memristor based neuromorphic systems [25]–[27] due to the similarities between memristors and biological synapse.

The continuous and discrete memristor emulators are designed to mimic the valence change memories (VCM) and the electrochemical metallization memories (ECM), respectively [28]. The VCM is associated with continuous evolve of the state variable (resistance) while ECM has the signature of sharp change of resistance value. As an application, this memristor IP core model was used to implement a memristive-Chua chaotic circuit, where all results were verified experimentally.

In [21], a memristor FPGA IP core was presented, which can generate discrete and continuous versatile memristor models. In discrete mode, the memristor emulator hold its memristive state for specific period then increase or decrease the memristive states according to a condition. On the other hand, the continuous model is built based on switching without adding any conditions which means the continuous can not hold the memristive state. Increasing the number of states in the discrete model leads to have a continuous model with a large number of states. But, It comes with a huge increase of the hardware resources as well which is not desirable. In this paper, we propose a new design for the memristor emulator that works in either discrete or continuous modes, Increasing the number of memristive states in the proposed design does not increases the hardware resources unlike [21]. More discussion and comparison between proposed memristor design and previous works are added in the results section.

Due to the memristor's nonlinearity, several memristive chaotic oscillators were presented [11], [29]–[37], and mathematically analyzed. Nevertheless, few were utilized in image encryption applications [38], [39]. Several research papers studied memristor hardware implementation for biologically inspired and neuromorphic systems [27], [40]. Fractional-order models of memristor and memcapacitor were used to design a memfractor chaotic oscillator in [41]. Moreover, a digital implementation for the memristive-Chua chaotic circuit was implemented and experimentally verified in [21].

In [42], the authors introduced FPGA implementation of memristor-chaotic circuit solving the chaotic differential equations including the memristor. There was no hardware dedicated for the memristor emulation. In this paper, we utilize the proposed memristor emulator for chaotic generator to be used as a part of the speech encryption system. The proposed memristor chaotic system offers more flexibility, where the number of memristor states can be easily changed in real time and different chaotic attractors can be generated. In [21], a memristor chaotic system was introduced based on two memristor states only. Moreover, increasing the number of memristor states requires more hardware resources, redesign the system and new synthesis. However, the proposed chaotic system is designed based on multiple memristors states, where increasing the number of memristor states does not demand any change in the hardware resources. In addition

$x(t)$	Comparator	Operation (Op)
$+ve$	$\varphi(t_i) > c_1$	$b = b + \Delta b_1$ $c_1 = c_1 + \Delta c$
$+ve$	$\varphi(t_i) \le c_1$	$b = b$ $c_1 = c_1$
$-ve$	$\varphi(t_i) > c_1$	$b = b$ $c_1 = c_1$
$-ve$	$\varphi(t_i) \le c_1$	$b = b - \Delta b_1$ $c_1 = c_1 - \Delta c$

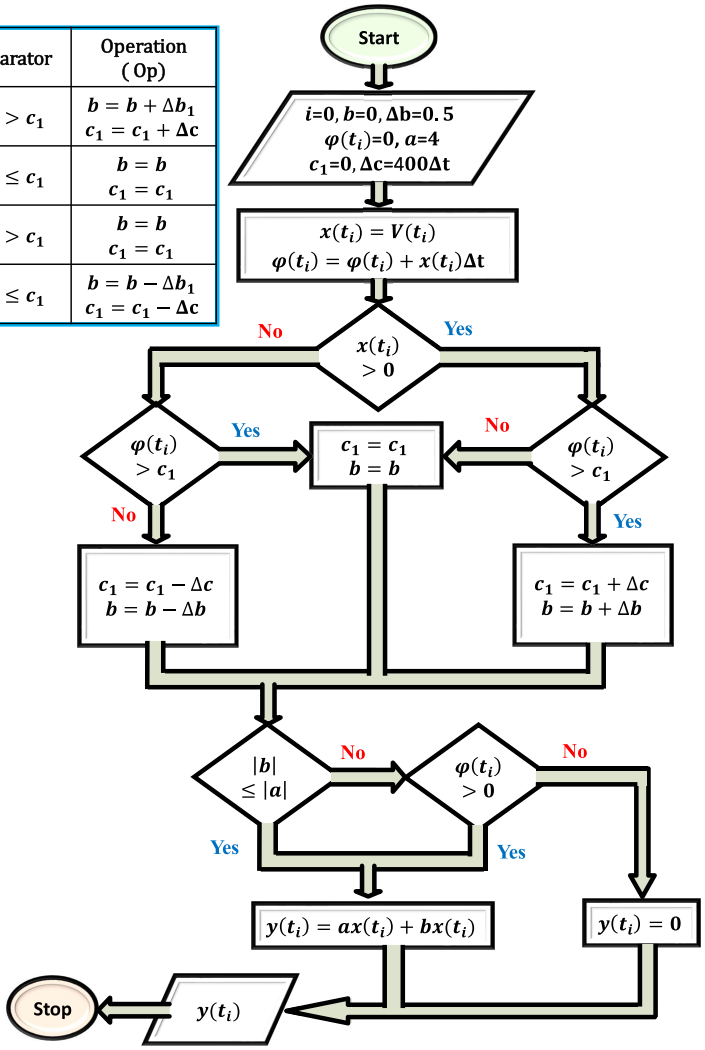


FIGURE 1. The flowchart of the proposed memristor emulator, where a, Δb and Δc are constants.

to that, the change in the memristor states can be done in real time without the need to new synthesis.

Some recent research on memristor have exposed the opportunity of improving the resolution of chaos and neural network models of computation [11]. Chaos-based ciphers can provide lightweight encryption with high performance and efficiency compared to more complicated standard ciphers. Amongst different forms of data, speech encryption is required for many applications such as radio, telephone and IP communication [43].

III. PROPOSED MODEL

A two-state bipolar memristor model was introduced as follows [14]:

$$y(t) = \pm ax(t) \pm b \begin{cases} -x(t) & \varphi\,(t) \le c \\ x(t) & \varphi\,(t) > c \end{cases}. \tag{1}$$

where y, x, and (a, b, c) are normalized output, input and constants, respectively. Two state lines $y = (\pm a \pm b)x$ and $y = (\pm a \mp b)x$ are used to define the memristor hysteresis loop. The condition $|b| \le |a|$ should be held to obtain the memristive behavior. In [21], a multi-state switching model was presented to generalize the two memristive states switching model by adding extra conditions. Discrete and continuous models were introduced alongside their FPGA realizations.

In this work, we follows the standard way to design and validate the proposed memristor emulator. In the first step, we develop an algorithm that shows the memristor functionality. Figure 1 shows a flow chart of the proposed algorithm works. Then, in the second step, we validate the proposed algorithm with numerical MATLAB simulations, where different scenarios are tested. After that, a block diagram is built for the design and implementation, which assist in the RTL coding stage. The proposed design is implemented using Verilog and realized on Xilinx FPGA kit board using existing DSP blocks. The DSP-based implementation reduces both development and verification time, where the functionality of these blocks is already verified.

The proposed implementations are tested in different stages and the implementation is designed based on different modules. The modules are tested by using a test-bench design to verify the design functionality and fulfills the design requirements. The test-benches cover all possible test cases to eliminate any mistakes. After the test-benches have been successfully passed, all modules are connected together to test the whole system. Then, the design is realized on hardware using the existing software tools and FPGA kit. After realizing the proposed system on FPGA, a ïĄnal test is done at the hardware level to ensure the system is functioning correctly.

In this work, a new digital memristor emulator is proposed, which exhibits continuous or distributed behaviors by controlling the model parameters. The continuous memristor model has been introduced based on the idea of switching without adding any conditions where b is changing with the time [21]. In this work, b and c are dynamically changing as shown in the flow chart shown in Fig. 1. The values of b and c are increased and decreased based on the sign of $x\,(t)$ and the comparison of $\varphi\,(t_{i+1})> c_1$. Based on the input sign and the comparator output, b and c are computed as presented in the table in Fig. 1. New memristive state is produced by changing b and c after the addition or the subtraction operations. When b and c are not changed, the memristive state does not changed. The hysteresis loops of the discrete and continuous memristor models are similar to the HP model, however a simple linear equation $I(t) = aV(t) \pm bV(t)$ is used instead of $V(t) = \left(x_d(t)R_{on} + (1 - x_d(t))\,R_{off}\right) i(t)$ in the HP model. The memristor model can be realized by following the steps which summarized in the flow chart, shown in Fig. 1:

- Initialize the variable $\varphi\,(t)$, i, a, b, Δb, c_1, and Δc.
- The first process introduces the integration of the input $x(t_i)$ where $\varphi\,(t_{i+1}) = \varphi\,(t_i) + x\,(t_i)\,\Delta t$.
- Based on the sign of the input $x(t_i)$ and the comparator of $\varphi\,(t_{i+1})> c_1$, b and c are computed.
- In case of $x(t_i)$ is negative, b and c can be computed as follows: if $\varphi\,(t_{i+1})> c_1$ then $c_1 = c_1$ and $b = b$ else $c_1 = c_1 - \Delta c$ and $b = b - \Delta b$.
- In order to keep the condition $|b| \leq |a|$, another comparison is added as follows: if $|b| \leq |a|$ then the output current is computed based on $y(t_i) = ax(t_i) + bx(t_i)$, otherwise the output current is zero.

In order to refer to a device as a memristor; there are three fingerprint should exist. The first and the most important one is the existence of the pinched hysteresis loop, which uniquely identify the memristive devices [1]. Figure 2 illustrates the discrete and continuous pinched hysteresis loop for input signal $\sin(2\pi ft)$ where $a = 4$, $\Delta b = 0.5$ and $\Delta t = 10^{-7}$. The discrete hysteresis loop ($f = 800\ Hz$ and $\Delta c = 1000\Delta t$) and continuous hysteresis loop ($f = 100\ Hz$ and $\Delta c = 100\Delta t$) are presented in Figs. 2(a) and 2(b), respectively. The behavior of c versus the input is also depicted in Figs. 2(c) and 2(d) for discrete and continuous memristor, respectively. Please note that c is continuously/discretely changing for the continuous/discrete cases.

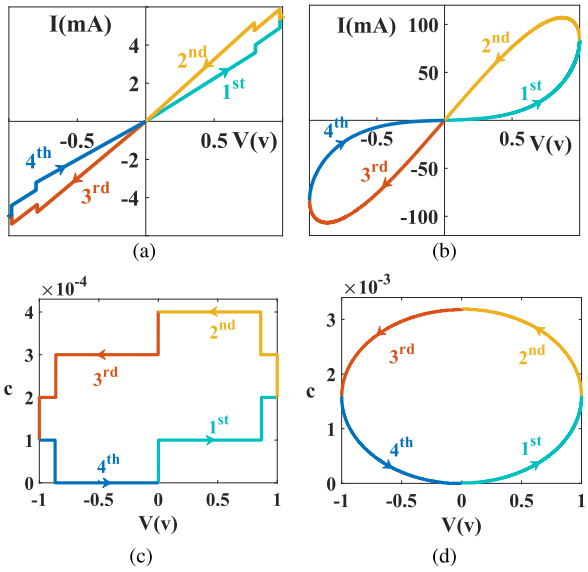


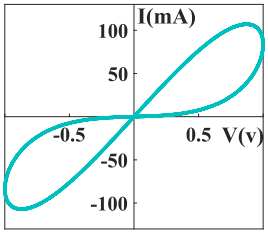
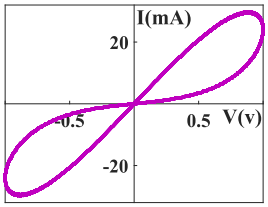
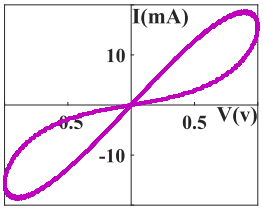
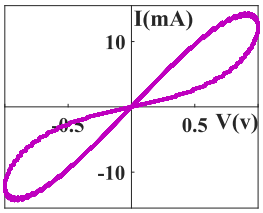
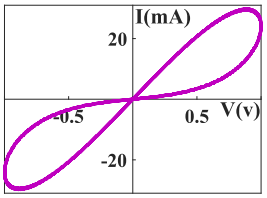
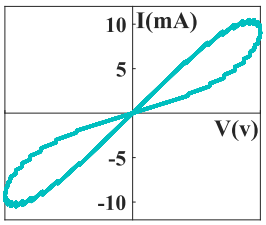
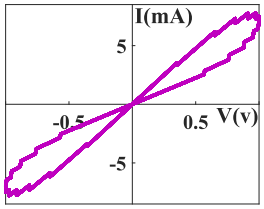
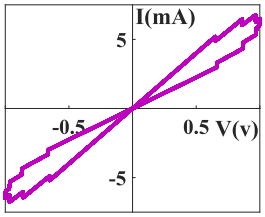
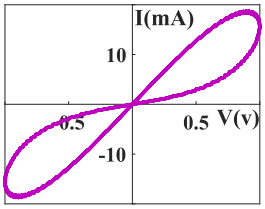
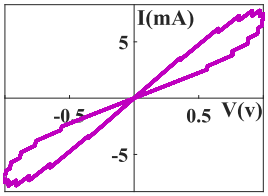
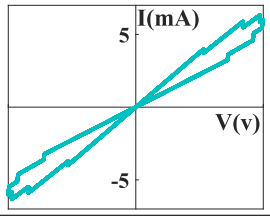
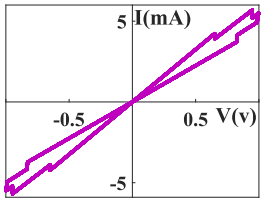
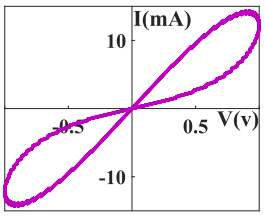
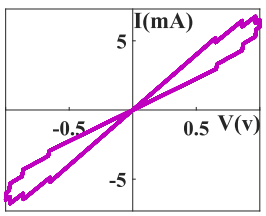
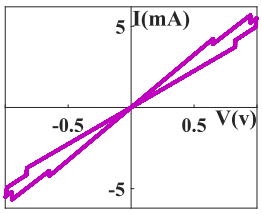
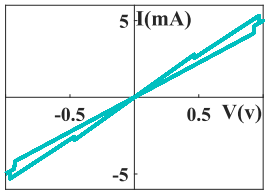
FIGURE 2. Pinched hysteresis loop for input $\sin(2\pi ft)$, where $a = 4$, $\Delta b = 0.5$ and $\Delta t = 10^{-7}$, (a) Hysteresis for $f = 800Hz$ and $\Delta c = 1000$ (b) c vs the input voltage for $f = 800Hz$ and $\Delta c = 1000$, (c) Hysteresis for $f = 100Hz$ and $\Delta c = 100$ and (d) the constant c vs input voltage for $f = 100Hz$ and $\Delta c = 100$.

Table 1 shows various hysteresis loop examples with different values of Δc and different frequencies. Based on the values of the frequency f and Δc, discrete and continuous loops can be generated. As f and Δc increase, the number of states and area inside the hysteresis loop decreases where the hysteresis loop becomes discrete. On the other hand, for the continuous model, f and Δc must be decreased. The second fingerprint of the memristive devices is shrinking the lobe area with increasing the frequency and reaches zero when f tends to ∞, as shown in Table 1. For the same Δc and different frequency, the memristor behavior may change from continuous to discrete modes while the lobe area is decreasing. This change happens because with increasing the frequency, the number of the states inside the lobe becomes smaller which means more discretization. Moreover, the area inside the hysteresis lobe reaching zero when the frequency approaches to infinity. The enclosed area versus the applied signal frequency and Δc in the I-V hysteresis loop of the proposed memristor model is illustrated in Fig. 3. The area decreases with increasing either f or Δc. The enclosed area is computed based on the analysis presented in [21].

A. HARDWARE ARCHITECTURE

Figure 4 presents the hardware architecture for the proposed memristor model. Fixed point arithmetic is used for the design and implementation, where the input is 32 bits divided into 24 bits and 8 bits for the fractional and the integer parts respectively. The output is truncated to 12 bits to be suitable for emulation using FPGA and oscilloscope. The top module of the proposed discrete and continuous memristor model is shown in Fig. 4(a), where $x(t_i)$ is the input signal, Δb, Δc and a are input constants used as control parameters. $y(t_i)$

TABLE 1. Generating discrete and continuous pinched hysteresis loops by increasing and decreasing frequencies f, and Δc, where $\Delta t = 10^{-7}$, $a = 4$, and $\Delta b = 0.5$.

Δc	$f = 100Hz$	$f = 400Hz$	$f = 700Hz$	$f = 1000Hz$
$100\Delta t$				
$400\Delta t$				
$700\Delta t$				
$1000\Delta t$				

depicts the output signal and *sel* is a selector with 2-bits created based on constants c_1 and b operations. The data path circuit for the proposed memristor model is presented in Fig. 4(b). The output $y(t_i)$ is calculated based on $y(t_i) = (a + b)x(t_i)$. The data path of the proposed memristor model is presented in Fig.4(b). The proposed design is developed based on the flow chart shown in Fig. 1, in which, an accumulation is required for the input to compute $\varphi(t_i) = \varphi(t_i) + x(t_i)\Delta t$. This accumulation is done in the design based on the adder. A comparator is required to compare $\varphi(t_i)$ with c_1. While, the multiplexers are required to select the desire operations.

While $\varphi(t_i)$ is computed by accumulating the input samples, the constant c_1 is calculated by adding the outputs of the two left multiplexers. c_{in} is a carry in used to apply the two's complement for the subtraction operation when the multiplexer output is inverted Δc. c_{in} is produced based on an AND gate between the *sel* two bits. As for the two left multiplexers, one is used to get the feedback and one is used to select 0, Δc or inverted Δc. The selection can be done based on *sel* as shown in the table in Fig. 4(a) and the Mux block in Fig. 4(a). Computing the constant b is performed using the same method of computing c_1. Both c_1 and $\varphi(t_i)$ drive the comparator, where the output of the comparator is either 0 or 1 when $\varphi(t_{i+1}) > c_1$ or $\varphi(t_{i+1}) \leq c_1$, respectively. The comparator output is concatenated with the MSB of $x(t_i)$ to create *sel* signal. Finaly, a and b are added and multiplied by the input signal $x(t_i)$. Moreover, a and b drive a comparator where the output is either 1 or 0 when $|b| > |a|$ or $|b| \leq |a|$, respectively. The comparator output is ANDed with the MSB of $\varphi(t_i)$ to drive the multiplexer. The multiplexer selects 0 or the multiplier output when the selector is 1 or 0, respectively.

IV. MEMRISTOR-BASED CHAOTIC OSCILLATOR

Figure 5 presents the proposed chaotic circuit, which consists of a memristor, two capacitors, two inductors and one negative resistor. The circuit has a parallel type Van der Pol oscillator and a series type LC resonator [44]. The two parallel diodes are replaced by the proposed

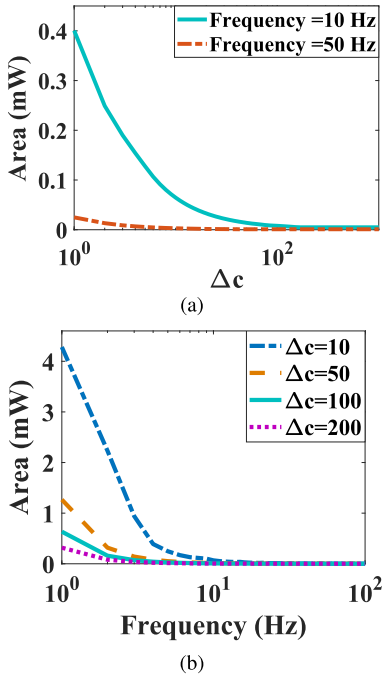


FIGURE 3. Enclosed area versus the applied signal frequency and Δc (a) effect of changing the frequency and (b) effect of changing Δc.

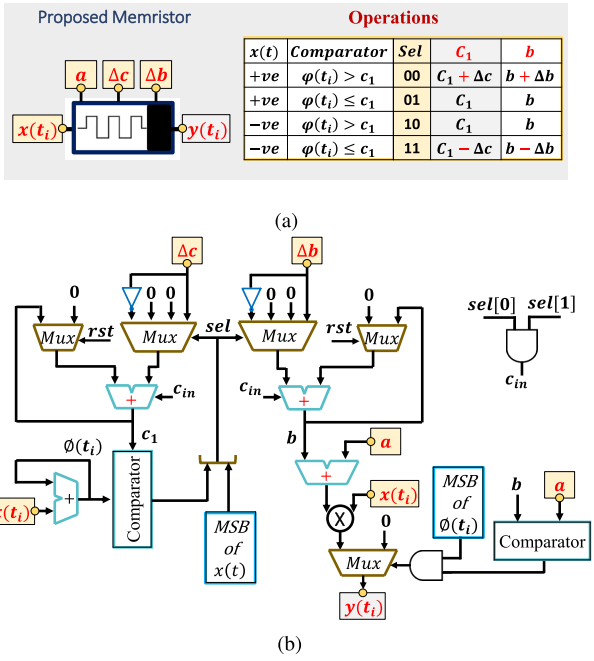


FIGURE 4. Proposed memristor model hardware architecture, (a) top module and *sel* 2-bits created based on constants c, b operations and (b) the data path circuit for the proposed memristor model.

memristor model. The circuit equations are:

$$C_1\frac{dv_1}{dt} = \frac{1}{R_1}v_1 - i_2 - i_4, \tag{2a}$$

$$L_1\frac{di_2}{dt} = v_1, \tag{2b}$$

$$C_2\frac{dv_3}{dt} = i_4, \tag{2c}$$

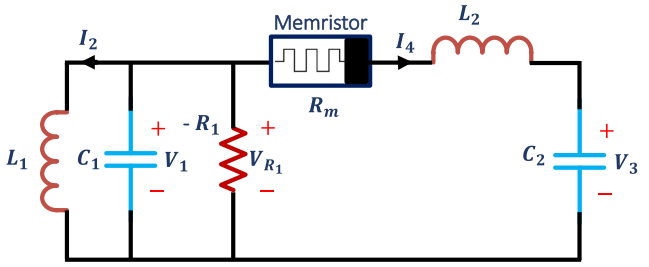


FIGURE 5. The chaotic circuit using the proposed memristor model.

$$L_2\frac{di_4}{dt} = v_1 - v_3 - i_4R_m \tag{2d}$$

$$R_m = a + b, \quad q(t) = \int_0^t i_4(t)dt \tag{3}$$

Following the same scaling presented in [3] for the HP memristor, the variables and parameters in (2) can be scaled as follows: $x = \frac{V_1}{V_0}$, $y = \frac{i_2}{i_0}$, $z = \frac{V_3}{V_0}$, $w = \frac{i_4}{i_0}$, $\alpha = \frac{R_0}{R_1}$, $e_0 = \frac{c_0}{c_1}$, $e_1 = \frac{L_0}{L_1}$, $e_2 = \frac{c_0}{c_2}$ and $e_3 = \frac{L_0}{L_2}$. After scaling (2), the differential equations are normalized as follows:

$$\dot{x} = e_0\left(\alpha x - y - w\right), \tag{4a}$$

$$\dot{y} = e_1\left(x\right), \tag{4b}$$

$$\dot{z} = e_2\left(w\right), \tag{4c}$$

$$\dot{w} = e_3\left(x - z - wR_m\right) \tag{4d}$$

Based on backward Euler method, the following equations are obtained:

$$x_{n+1} = x_n + e_0h\left(\alpha x_n - y_n - w_n\right), \tag{5a}$$

$$y_{n+1} = y_n + e_1h\left(x_n\right), \tag{5b}$$

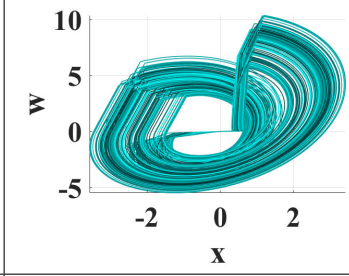
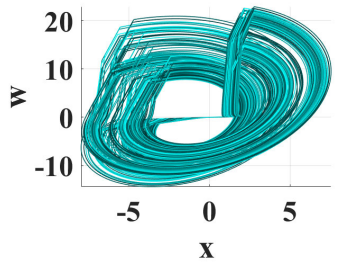
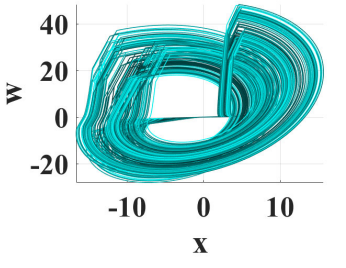
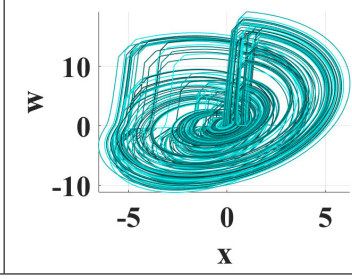
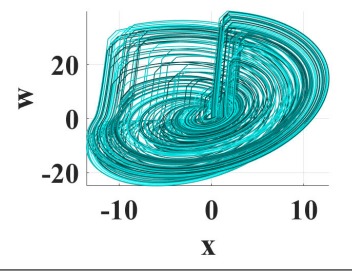
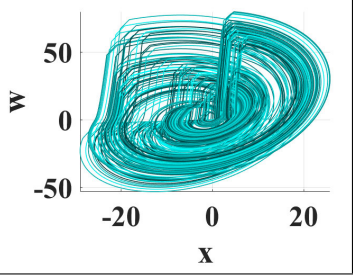
$$z_{n+1} = z_n + e_2h\left(w_n\right), \tag{5c}$$

$$w_{n+1} = w_n + e_3h\left(x_n - z_n - w_nR_m\right) \tag{5d}$$

The mathematical model of the chaotic system is derived from the circuit shown in Fig. 5 and given by (2), which includes four independent state variables corresponding to the circuit currents and voltages. Hyperchaotic systems usually have more than one positive Lyapunov exponent. That is, the dynamics of the system is expanded in two or more directions simultaneously. which result in higher unpredictability, much more complicated structure of the attractors and better performance in chaos based secure communication applications [45].

Figure 6 shows various projections of the 4D chaotic attractor based on the proposed memristor emulator, where $h = 2^{-7}, \alpha = 0.75, e_0 = 1, e_1 = 0.9, e_2 = 0.5, e_3 = 3.25, \Delta b = 0.9, a = 1$, and $\Delta c = 25$. Another chaotic case is plotted in Fig. 7, after changing some design parameters in the chaotic systems which are $\alpha = 0.5$, $e_1 = 1, \Delta b = 0.5$, and $\Delta c = 6$. The memristance R_m for both cases in Figs. 6 and 7 are shown in Fig. 8, where its value fluctuates among different levels. Table 2 presents the effect of changing the memristor parameters (Δc and Δb) on the proposed chaotic generator for $x - w$ projection, where

TABLE 2. Effect of changing the memristor parameters (Δc and Δb) on the chaotic generator for $x - w$ attractor, where $h = 2^{-7}$, $\alpha = 0.75$, $e_0 = 1$, $e_1 = 0.9$, $e_2 = 0.5$, $e_3 = 3.25$, and $a = 10$.

	$\Delta c = 50$	$\Delta c = 100$	$\Delta c = 200$
$\Delta b = 8$			
$\Delta b = 9$			

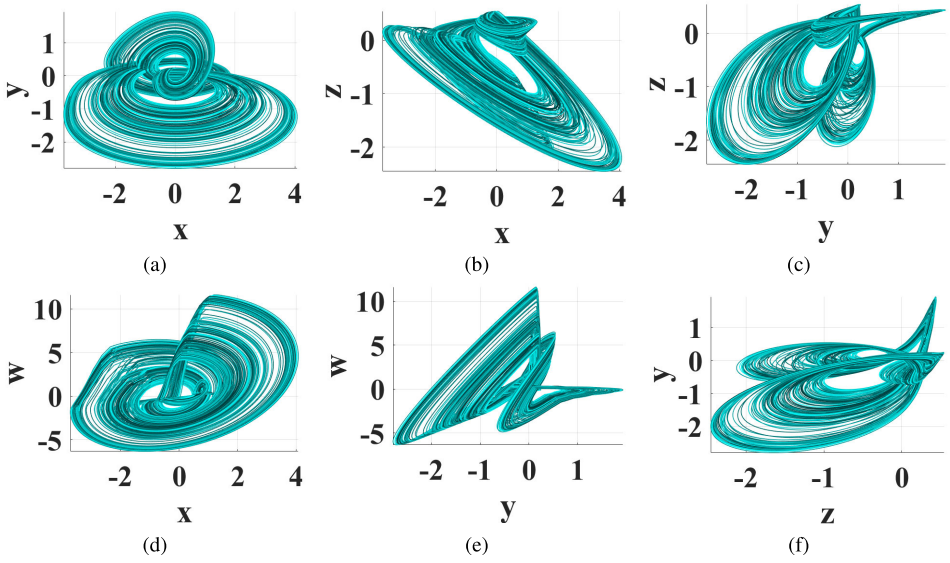


FIGURE 6. Chaotic attractor projections where $h = 2^{-7}$, $\alpha = 0.75$, $e_0 = 1$, $e_1 = 0.9$, $e_2 = 0.5$, $e_3 = 3.25$, $\Delta b = 0.9$, $a = 1$, and $\Delta c = 25$.

$h = 2^{-7}, \alpha = 0.75, e_0 = 1, e_1 = 0.9, e_2 = 0.5, e_3 = 3.25$, and $a = 10$. It can be inferred that the modification of the proposed memristor parameters Δc and Δb enhances the chaotic generator by introducing independent parameters that control the output ranges.

A. PARAMETERS RANGES

Bifurcation diagrams identify the parameters range corresponding to bounded and also chaotic responses. For a chosen bifurcation parameter, the values of x representing local maxima are plotted, i.e., the time series is sampled showing whether the solution is stable, periodic or chaotic. Figure 9 shows the bifurcation diagrams of the chaotic system (5) against five variables, each fixing the other variables to the values $e_0 = e_1 = 1$, $e_2 = 0.5$ and $e_3 = 3.25$ and $\alpha = 0.5$. At these values, Lyapunov Exponents (LEs) are computed using Wolf's algorithm [46] and equal (0.163967, 0.028446; 0.001874, -2.730914), which indicate hyperchaotic behavior with two positive LEs followed by an LE approaching zero and the last one is negative. Moreover, Fig. 10 indicates that the Maximum Lyapunov Exponent (MLE) versus α matches the corresponding bifurcation diagram, where the

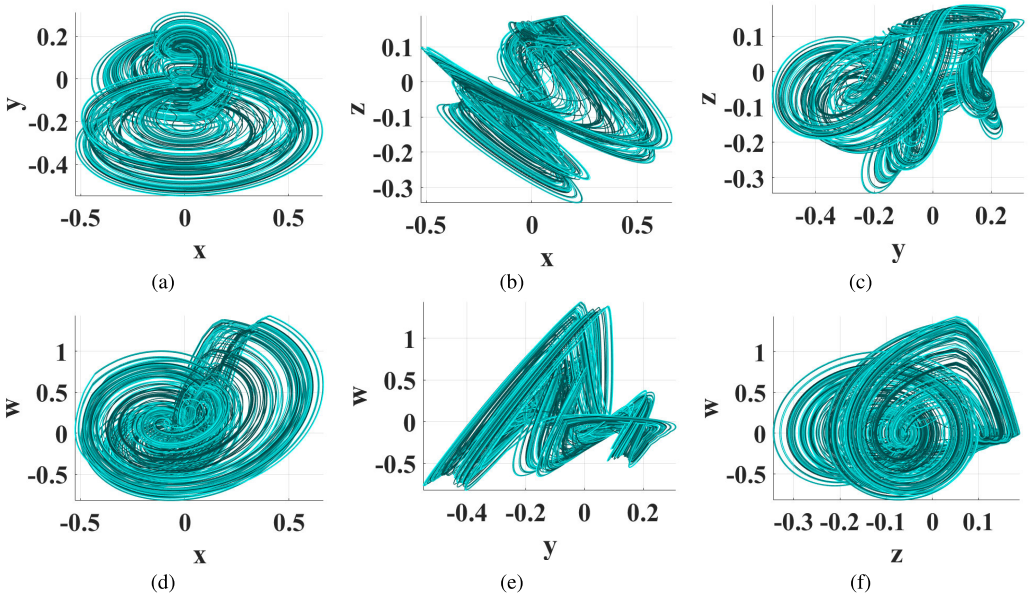


FIGURE 7. Chaotic attractor projections where $h = 2^{-7}$, $\alpha = 0.5$, $e_0 = 1$, $e_1 = 1$, $e_2 = 0.5$, $e_3 = 3.25$, $\Delta b = 0.5$, $a = 1$, and $\Delta c = 6$.

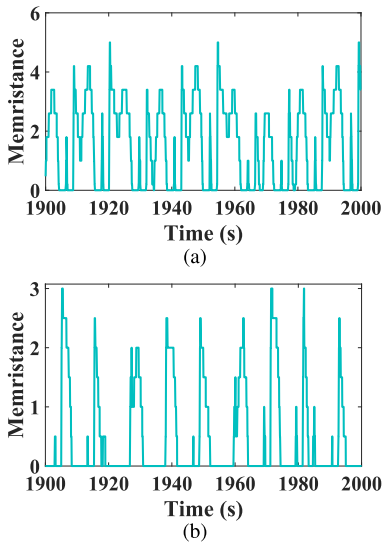


FIGURE 8. Time evolution of the memristance for the cases shown in (a) Fig. 6, and (b) Fig. 7.

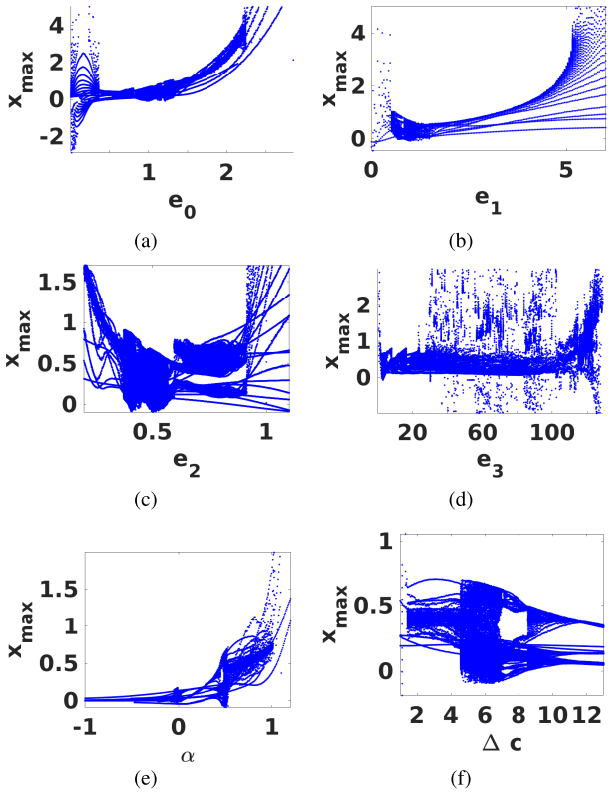


FIGURE 9. Bifurcation diagrams of the chaotic attractor against each variable fixing the other variables to the given values. The memristor parameters are $\Delta b = 0.5$ and $a = 1$.

output undergoes period doubling bifurcation till the chaotic firing around $\alpha > 0.44$.

B. HARDWARE ARCHITECTURE

Figure 11 illustrates the design and methodology of the proposed chaotic generator, where the state variables x, y, z and w are stored in 4 registers. Fixed point format is used for the design, where each register uses 32 bits with 8 bits integer and 24 bits fractional parts. Furthermore, the numerical solutions for x, y, z, and w are computed using the combinational circuits between the registers. The combinational circuits contain the following blocks; Memristor block is used to compute memristance ($R_m = a + b$); multipliers, subtractors and adders are used to perform the multiplication, subtraction and addition operations in (5). In order to improve the performance and power consumption of the proposed chaotic generator, all parameters are chosen as a power of two of

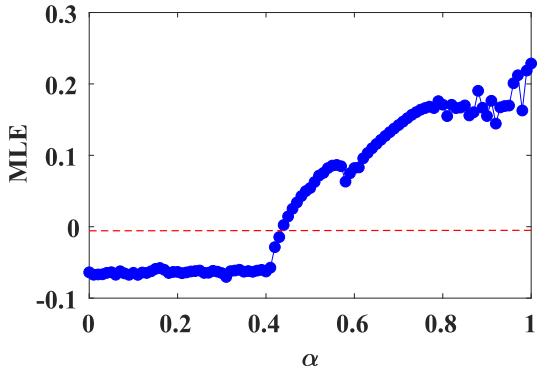


FIGURE 10. Maximum Lyapunov Exponent (MLE) diagrams with respect to the parameter α.

sum of powers of two. After using a power of two, simple operations such as bit-shifts and bit-selects are used instead of multipliers. For $h = 2^{-7}$ which is a power of two, a shift operation block with 7-bits shift right is used.

V. SPEECH ENCRYPTION BASED MEMRISTOR CHAOTIC OSCILLATOR

The proposed memristor chaotic oscillator can be used as a building block for any encryption scheme suitable for text, speech and image encryption. This work focuses on the speech encryption. The proposed encryption process is presented in Fig. 12, where nine sub-keys ($K_0 : K_8$) are used to build the whole encryption key. The sub-keys correspond to four initial values and five independent parameters of the chaotic generator as explained in the next subsection. The encryption sub-keys drive the independent parameters calculator along with the fix parameters to produce the four initial values and the five independent parameters of the proposed chaotic generator. Then, the chaotic generator produces four outputs x, y, z and w to drive the encryption scheme. The encryption system is implemented based on pipelining technique to improve the performance. The pipelining allows each block unit including parameters calculation, chaotic generator and encryption scheme to run concurrently. Only three clock cycles latency are required to produce the first encryption sample.

Figure 13 presents the hardware architecture of the proposed encryption scheme, which utilizes the four outputs of the memristor chaotic generator. The proposed encryption scheme works as follows: each input speech sample is divided into eight parts with 2 bits each. Every 2 bits drive a single S-box block, to use 8 S-box blocks. The general idea of S-Box is taking some number of input bits and substitute them with some number of output bits. The proposed S-box is implemented based on dynamic LookUp Table (LUT), unlike the previous S-box blocks, which were built based on fixed LUT. Fixed LUT S-box are usually required a large LUT to store the data. In contrast, the proposed dynamic S-box is designed to be small to save the hardware resources. For example, 8 bits required $2^8 = 256$ positions with 8 bits for every row in the fixed S-box. On the other hand, the proposed S-box needs only $4 \times 2^2 = 16$ with 2 bits for every row. $x[15 : 0]$ is the input to the S-box blocks and is partitioned as aforementioned. The stored data in the LUT is created as follows: $x[1]$ is inverted twice then repeated twice. $x[0]$ is inverted for each row of the table. Based on the previous process of inverting x, all possibles value x will be covered.

The outputs of the S-box blocks are permuted based on the permutation block, where $y[15 : 0]$ are used as control bits. Two level permutations are performed based on the algorithm introduced in [47]–[49]. Moreover, a multiplexer is used to select between the output of S-box and the permutation block, where the selector is the Least Significant Bit (LSB) of the feedback signal (Fb). Another multiplexer is used to select between z and w based on $Fb[0]$. Finally, the encrypted output signal is produced by XORing the outputs of both multiplexers with Fb. The decryption block diagram is presented in Fig. 14, which is built by reversing the encryption process in Fig. 13.

The speed of encryption block is usually determined by the speed of its primitive operation such as the substitution boxes. To decrease the hardware cost, a smaller S-box datapath is used instead of using large S-Boxes as in the conventional methods. However, using small s-box reduces the security level. To overcome this problem, the proposed S-box is not fixed, and its data change each clock cycle. In [50], a 8×8 S-box was introduced, which required 2048bits LUT. Another attempt to decrease the size has been proposed in [42], with 4-bit S-box in which a 16-bits input require 4 LUT with 64 bits each which means 256-bit LUT in total. The proposed S-box design is an optimized version of the hardware architecture, introduced in [42]. Figure 15(a) illustrates the block diagram of the S-box block, where the input is 2 speech bits (sp) and 2 bits from the chaotic output x. The table in Fig. 15(a) describes the S-box operation where the two sp bits are replaced by two bits generated from x. To build the LUT and cover all possibilities for every 2 bits without repeating any values, the input (sp) is used as an address in the left of the LUT to select among four outputs in the right. The four outputs are created by inverting the two bits from x with different inversion cases. The inverse S-box can be built by reversing the S-box operation as presented in Fig. 15(b). The proposed design requires 8 LUT for 16-bits input and each LUT needs 2-bits with 64-bit in total. Thus, the proposed design uses only 1/4 of LUT used in [42] and 1/32 of LUT used in [50].

A. ENCRYPTION KEY DESIGN

The encryption key consists of nine sub-keys $K_0 : K_8$, which determine the values of the four initial values and the five independent parameters of the chaotic generator. They are used as perturbation values around fixed values for the parameters, which are known to generate chaos from the bifurcation diagrams presented in previous section. Hence, these perturbations should not be increased

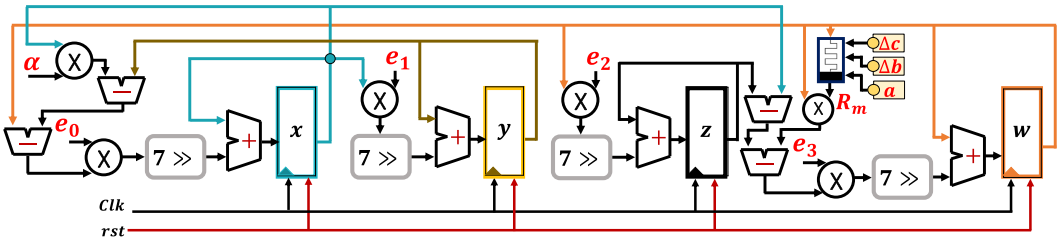


FIGURE 11. Hardware realization of proposed memristor-based chaotic generator, where 4 registers are used to store the numerical solution of x, y, z and w.

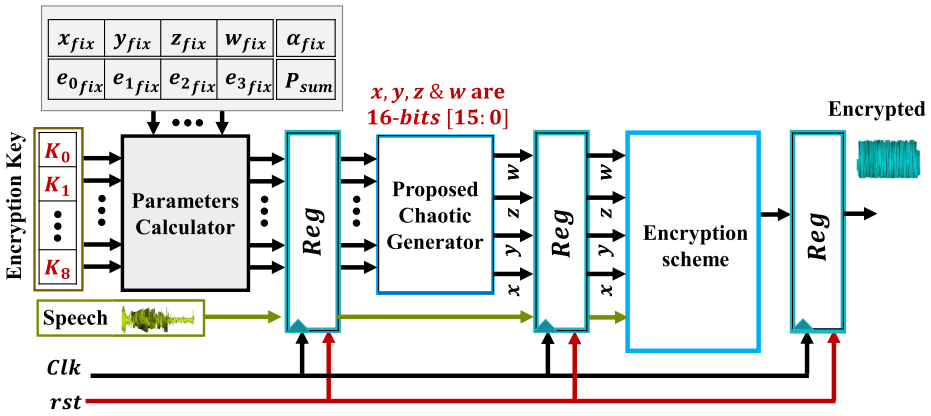


FIGURE 12. Encryption process block diagram, where the design is built based on pipelining technique to improve the performance.

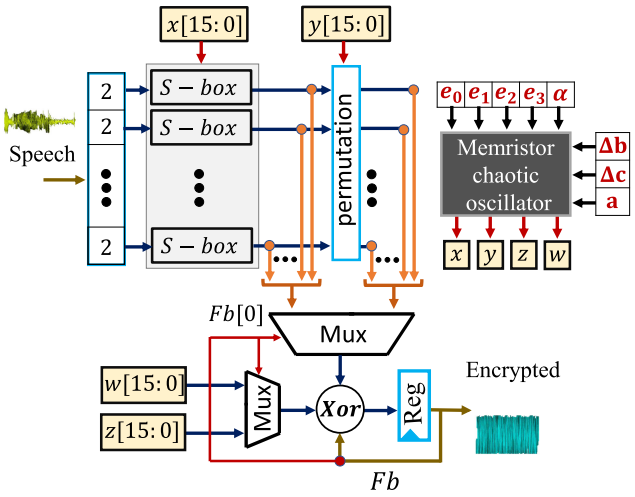


FIGURE 13. Encryption scheme hardware architecture based on s-box, permutation and chaotic generator.

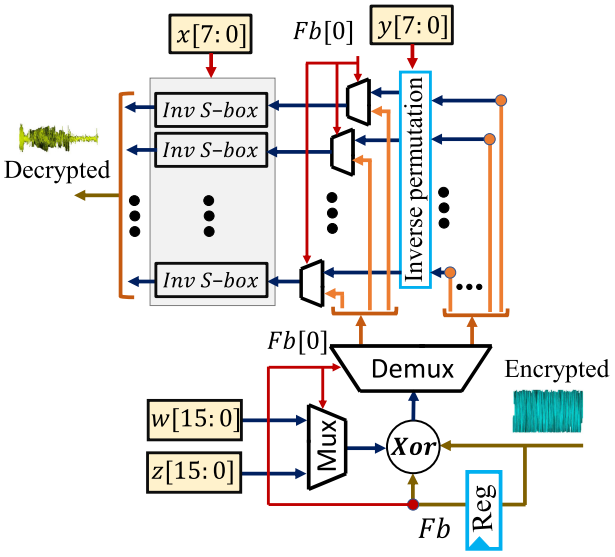


FIGURE 14. Decryption scheme hardware architecture.

above specific limit or the chaotic generator will not be chaotic. Thus, the sub-keys must be downscaled resulting in upper bounded values $x_k, y_k, z_k, \ldots$ and α_k as shown in Fig. 16. The effective number of bits for the encryption key are 24, 24, 24, 24, 20, 20, 20, 20 and 20 for the sub-keys $K_0, K_1, \ldots$ and K_8, respectively. Therefore, the total effective encryption key length is 196 bits, i.e., the key space is 2^{196}, which is resistant to bruteforce attacks [43]. Each sub-key is concatenated with zero's in the MSBs to get 32 bits of $x_k, y_k, \ldots$ and α_k.

B. CHAOTIC PARAMETERS COMPUTATION

The computations of chaotic parameters are given by:

$$\begin{aligned} x_0 &= x_k + x_{fix} + P_{sum}, \quad y_0 = y_k + y_{fix} + P_{sum}, \\ z_0 &= z_k + z_{fix} + P_{sum}, \quad w_0 = w_k + w_{fix} + P_{sum}, \\ e_0 &= e_{0_k} + e_{0_{fix}} + P_{sum}, \quad e_1 = e_{1_k} + e_{1_{fix}} + P_{sum}, \\ e_2 &= e_{2_k} + e_{2_{fix}} + P_{sum}, \quad e_3 = e_{3_k} + e_{3_{fix}} + P_{sum}, \\ \alpha &= \alpha_k + \alpha_{fix} + P_{sum}, \end{aligned} \tag{6}$$

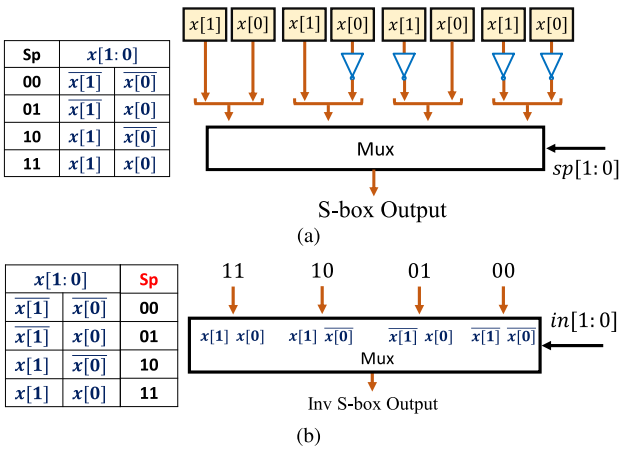


FIGURE 15. (a) S-box block diagram and (b) Inverse S-box block diagram.

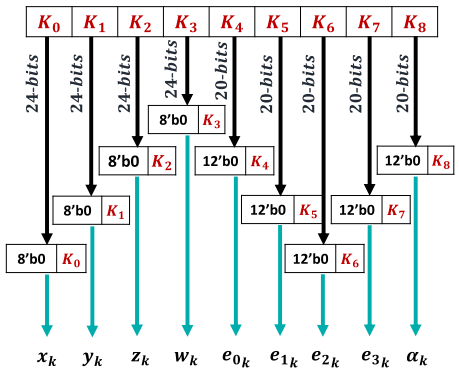


FIGURE 16. Scaling of sub-key values, where each sub-key is concatenated with specific number of bits to complete 40-bit.

where x_k, y_k, ... and α_k are scaling values extracted from the encryption key. x_{fix}, y_{fix}, ... and α_{fix} depict the fixed values of the chaotic parameters. The value P_{sum} depicts an input dependent term, which improves the resistance to differential attacks [49], and is given by:

$$P_{sum} = mod\left(\sum_{i=1}^{N} S_i, 10\right)/1000, \tag{7}$$

where S_i are the speech samples.

C. PERFORMANCE EVALUATION

The performance of the proposed encryption scheme is evaluated using a set of standard tests as follows.

1) PERCEPTUAL AND STATISTICAL TESTS

The original speech time waveforms and spectrograms are characterized by amplitude variation as shown in Table. 3. The spectrogram is a time-frequency decomposition of the magnitude squared of the signal spectrum in a logarithmic scale. It represents the variation of the spectral density with time and frequency, which is specified by the strength of the color. On the other hand, the time waveforms of the encrypted signals are almost random, their spectrograms exhibit scattered power in the entire spectrum over all times/frequencies as shown in Table. 4. The samples of the encrypted speech are uniformly distributed in the histogram.

In order to have a strong encryption, the original and the encrypted signals should be weakly correlated. The correlation coefficient represents this statistical measure and is given by:

$$r_{xy} = \frac{cov(x, y)}{\sqrt{D(x)D(y)}}, \tag{8}$$

where $cov(x, y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y))$, $D(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))^2$, $E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i$ and x and y are the two signals. The correlation coefficients between the original and the encrypted signals should approach zero indicating that the two signals are weakly correlated. The entropy is used to measure the randomness of the speech samples and it is defined as follows:

$$\text{Entropy} = -\sum_{i=1}^{2^p} P(S_i)\log_2 P(S_i), \tag{9}$$

where $P(S_i)$ is the probability of the sample value S_i and p is the number of bits per sample and it is the optimal entropy value. For a random encrypted signal, the entropy value should approach p, which equals 16 for the considered test speech files. The Mean Squared Error (MSE) indicates how far the encrypted signal is from the original signal with larger values indicating more security and is given by:

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(x_i - y_i)^2. \tag{10}$$

Table 5 summaries the correlation coefficients, the entropy and MSE results showing g the good performance of the proposed encryption system which indicate the high randomness of the ciphered data.

2) ROBUSTNESS AGAINST DIFFERENTIAL ATTACKS

Differential cryptanalysis is based on the idea that for a given encryption key, there is a high probability that a specific difference between two plain texts may cause a specific difference between the corresponding ciphered texts. Thus, the differential attack is achieved by applying a pair of plain texts with fixed differences, investigating the differences in the ciphered texts and assigning probabilities to the key. These probabilities are obtained after studying the characteristics of the encryption algorithm. In order to assess the robustness of an encryption scheme against differential attacks, only one of the input speech signal samples is changed and a modified speech signal is obtained. The original and the modified speech signals are encrypted using the same key and the two encrypted signals, A and A', are produced. The encrypted signals are compared based on the Number of Samples Change Rate (NSCR) and the Unified

TABLE 3. Time waveforms, spectrograms, and histograms of the test speech signals.

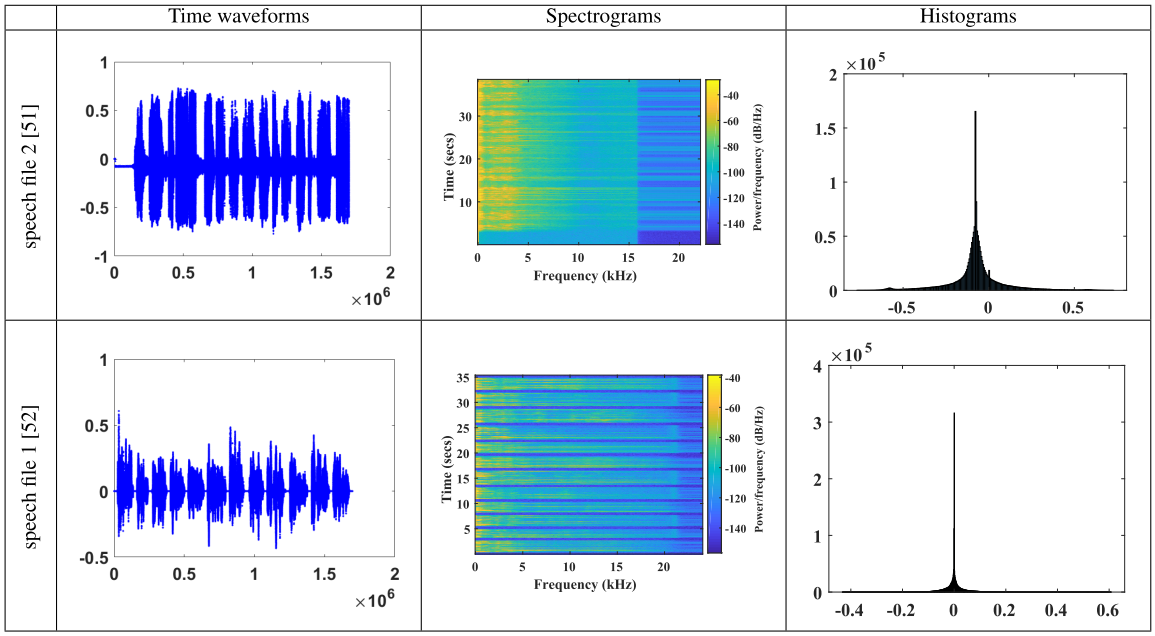
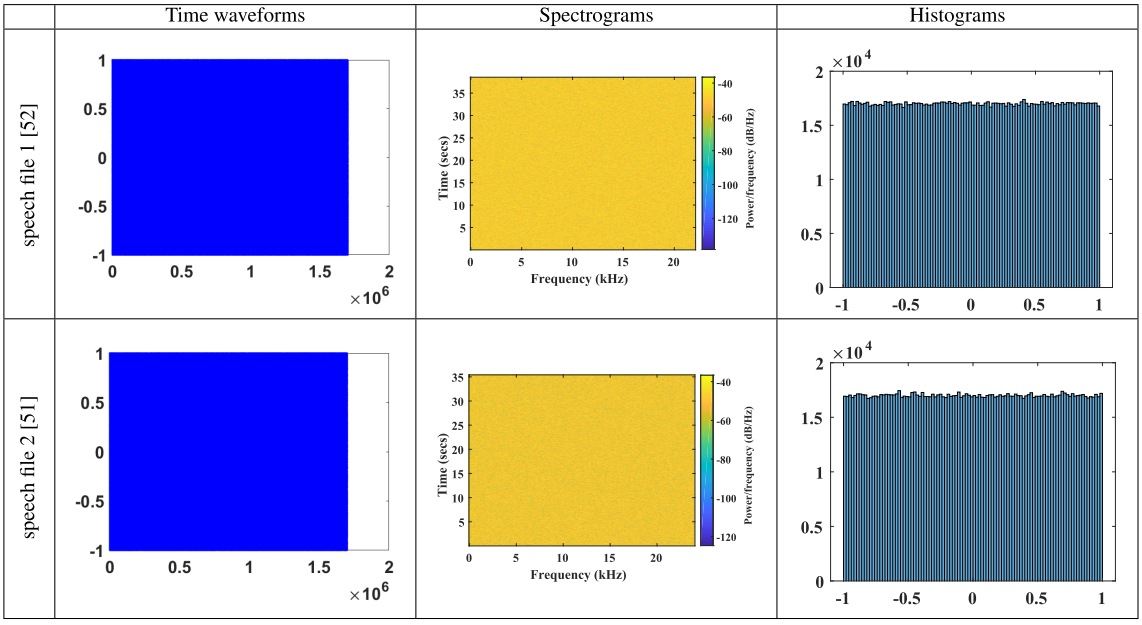


TABLE 4. Time waveforms, spectrograms, and histograms of the corresponding encrypted speech.



Average Changing Intensity (UACI) given by:

$$\text{NSCR} = \frac{\sum_i D_i}{N} \times 100\%, \tag{11a}$$

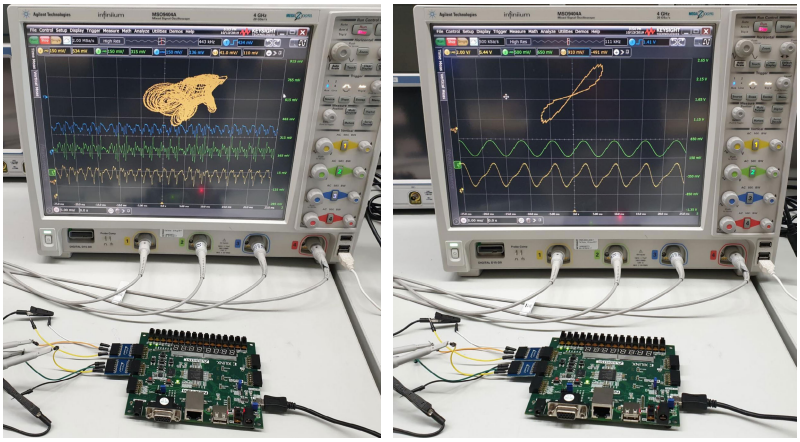
$$\text{UACI} = \frac{1}{N}\left(\sum_i \frac{|A_i - A_i'|}{2^p - 1}\right) \times 100\%, \tag{11b}$$

where $D_i = \begin{cases} 1, & A_i \neq A_i' \\ 0, & \text{otherwise} \end{cases}$.

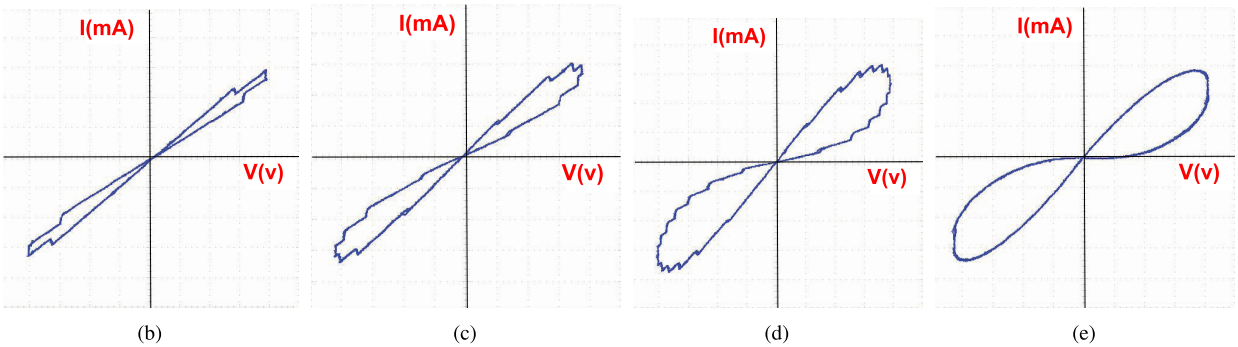
Table 6 gives the minimum, maximum and average values for 10 randomly chosen samples, which successfully approach the recommended values 100 % and 33.3 %, for NSCR and UACI, respectively [43].

3) NATIONAL INSTITUTE OF STANDARDS & TECHNOLOGY (NIST) STATISTICAL TEST SUITE

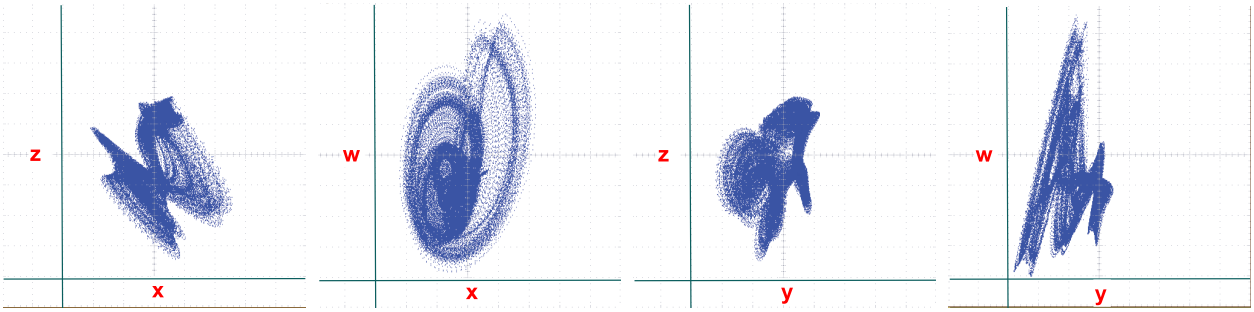
More advanced statistical tests are provided by the NIST statistical test suite [53], which is a statistical test suite for random and pseudo-random number generators for cryptographic applications. The tests are designed to examine the randomness characteristics of a sequence of bits by evaluating the P-value distribution (PV) and the proportion of passing sequences (PP).



(a)



(b) (c) (d) (e)



(f) (g) (h) (i)

FIGURE 17. Different experimental results for proposed memristor model I-V characteristic and chaotic attractors. (a) Experimental setup, (b-e) different emulated memristor results and (f-i) chaotic attractors.

TABLE 5. Statistical measures of the proposed encryption scheme for speech file 1 and file 2.

	Correlation coefficients	Entropy	MSE
File 1 [52]	−0.00039935	15.9719	32646
File 2 [51]	−0.00060053	15.9719	32647

In order to run NIST tests, the encrypted speech files have been saved as ".wav"files in their native int16 representation (i.e. 16 bits/sample) with 16 bits sequence length. The tests are performed on 1, 572, 864 sequences. As an example, NIST tests were performed on two encrypted speech files as shown in Table 3. Fixed-point representation with finite precision slightly differs from numerical simulation of the chaotic generator. The memristive and chaotic behaviors are maintained as may be inferred from comparing (Tables 1 and 2 and Figs. 6 and 7) to (Fig. 17). Moreover, the encryption process is successfully performed comparing Table 4 to Figure 17. The two files pass NIST tests as given in Table 7.

VI. EXPERIMENTAL RESULTS

The proposed discrete and continuous memristor model, chaotic generator, and encryption system are implemented on Nexys 4 Artix-7 FPGA XC7A100T. To simulate the RTL

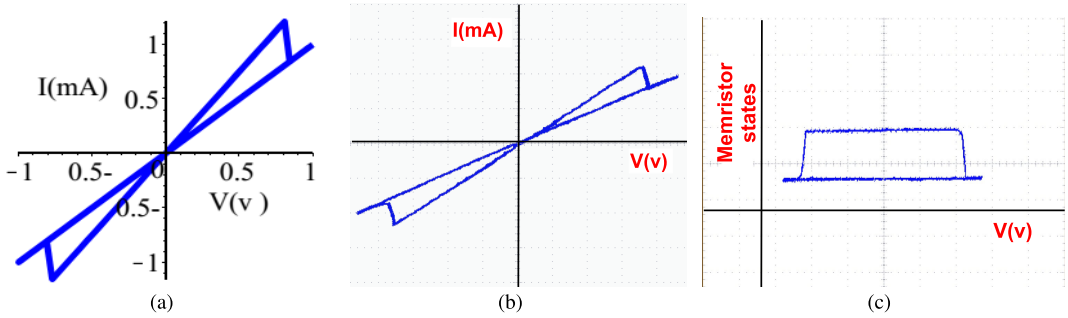


FIGURE 18. (a) Binary memristor MATLAB simulation I-V projection, (b) Binary memristor experimental results I-V projection and (c) memristance-V projection experimental results.

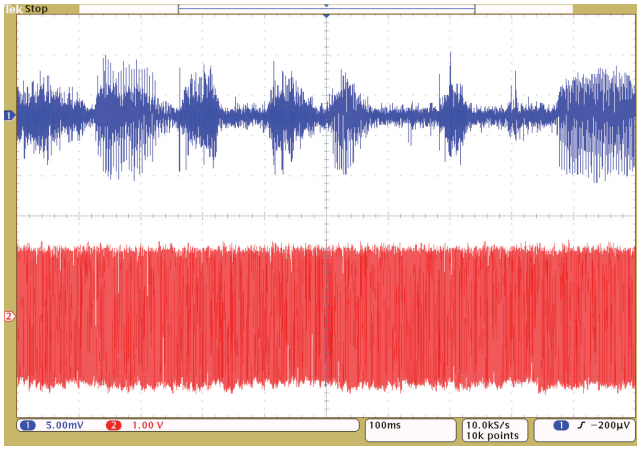


FIGURE 19. Oscilloscope experimental results for proposed encryption system.

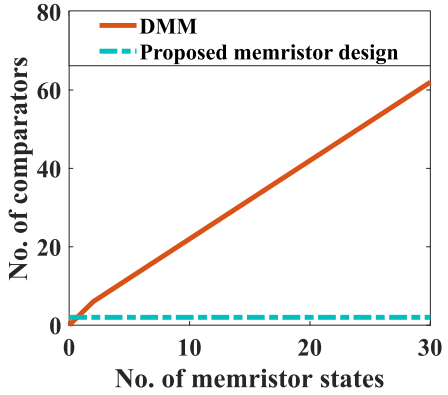


FIGURE 20. Number of comparators vs the number of memristors' states for proposed memristor design and DMM [21].

codes, ISim simulator in Xilinx ISE 14.5 was used. All results were firstly validated with MATLAB software and then compared with FPGA implementation results. Digital to Analog Converter (DAC) Pmod DA2 powered by Texas Instruments is interfaced with the FPGA to convert the digital data to Analog. Two DAC channels of 12-bit are provided allowing users to obtain a resolution up to about 1 mV. For the memristor, the input voltage is produced using the LUT inside the FPGA kit. The FPGA kit generate two channel of serial bits to derive the Pmod DA2, which converts the two serial bits to analog data to be displayed on the oscilloscope. Different experimental results for the proposed memristor model and chaotic generator attractors are illustrated in Fig. 17. The implementation process for the experimental results is done as follows: Xilinix ISE software is used to generate the bit file required for the experimental on FPGA. Since the DA chip only can receive positive numbers, a positive number is added to shift the whole output in the positive part. This number can be chosen based on the simulation results given by MATLAB. The Pmods in the FPGA are used to represents a 12-bit for each output signal. DA2 model is interfaced with the pmod to convert the 12-bits outputs to Analog signal to be displayed on the oscilloscope.

Figure 18 presents the simulation and experimental results for the binary memristor, where Fig. 18(a), 18(b) and 18(c) depict the MATLAB simulation I-V projection, binary experimental results of I-V and memristance-V planes respectively.

TABLE 6. Differential attack measures of the proposed encryption scheme for speech file 1 and file 2.

	UACI (%)			NSCR (%)		
	min	max	avg	min	max	avg
File 1 [52]	33.2280	33.4176	33.3167	99.9985	100	99.9992
File 2 [51]	33.2984	33.6051	33.4979	99.9966	100	99.9983

TABLE 7. NIST results for the encrypted data.

Test	Speech file 1		Speech file 2	
	PV	PP	PV	PP
Frequency	✓	0.979	✓	0.979
Block Frequency	✓	1	✓	1
Cumulative Sums	✓	0.958	✓	0.958
Runs	✓	1	✓	1
Longest Run	✓	1	✓	1
Rank	✓	1	✓	1
FFT	✓	1	✓	1
Non-overlapping Template	✓	0.992	✓	0.988
Overlapping Template	✓	0.958	✓	1
Universal	✓	1	✓	1
Approximate Entropy	✓	1	✓	1
Random Excursions	✓	1	✓	1
Random Excursions Variant	✓	0.987	✓	1
Serial	✓	1	✓	1
Linear Complexity	✓	1	✓	1
Final result	Passed		Passed	

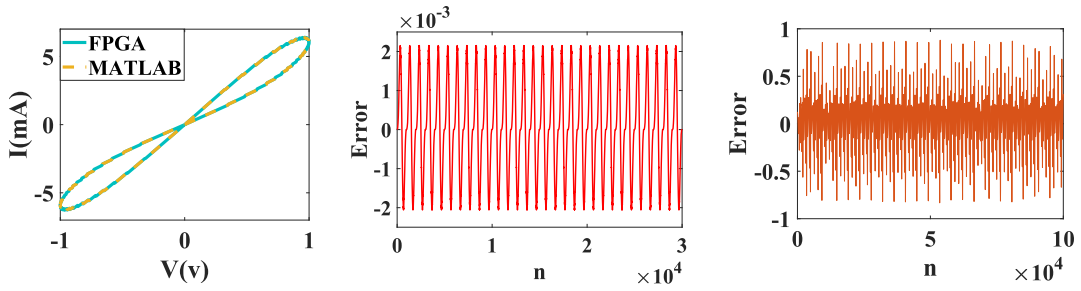


FIGURE 21. (a) I-V plane of proposed memristor model for MATLAB vs FPGA results, (b) error between MATLAB and FPGA for proposed memristor model and (c) error between MATLAB and FPGA for proposed chaotic system.

TABLE 8. A comparison between the proposed memristor model and previous works.

		FPGA	Slice LUT	Slice Reg	Max Freq "MHz"	Throughput bit/sec	Memristor type		Comparator	Multipliers
							Discrete	Continuous		
Proposed Memristor		Artix 7	135	57	189.494	2.2739	✓	✓	2	1
Ref [21]	DMM 2 states		72	44	231	2.772	✓	✗	0	0
	DMM 8 states		289	45	138.17	1.6580	✓	✗	14	0
	CMM		106	58	122.20	1.4664	✗	✓	0	1
Ref [18]		Cyclone II	3266 Logic Elements	32	NA	NA	✗	✓	NA	NA

Experimental results of proposed encryption system are presented in Fig. 19 for speech file 2. For speech encryption and decryption FPGA experimental setup, the AC-97 audio Codec interface is used to pass the input speech signal from a microphone to the FPGA with input speech data at up to 18 bits and 48-kHz sampling is supported. The input speech bits are encrypted based on the proposed encryption scheme. The encrypted signal is delivered to the speaker, which sync wire auxiliary audio cable interfaced with AC-97 line out to get the encrypted signal. The decryption can be done by applying the decryption scheme on the encrypted signal inside the FPGA where the encrypted signal is stored in a register in FPGA.

A comparison between the proposed memristor model and previous works is given in Table 8. Unlike [18], [21], the proposed memristor model can produce Discrete Memristor Model (DMM) or Continuous Memristor Model (CMM). In [21], increasing the number of memrsitor states in DMM to have CMM increases the hardware complexity and resources as shown in Table 8, which requires to redesign of the entire system architecture. It can be inferred that the number of comparators and registers are increased and the performance is decreased with increasing the memristor states. On the other hand, using the proposed memristor model, the number of memristor states can be increased without changing the architecture. It can be simply done by changing the memristor parameters. Figure 20 shows the number of comparators vs the number of memristor states for both memrsitor emulator and DMM [21]. As can be seen for proposed design, the number of comparators does not change with increasing the memristor states. However, the number of comparators are increased with increasing the memristors states in DMM [21]. Reference [18] uses different FPGA kits for the implementation. However, the design requires a division operation which affect the performance and cost more hardware resources. On the other hand, proposed design needs simple arithmetic operations (no needs for division) compared with [18]. Generally, the proposed memristor model can be programmed on the fly by loading the proper configuration parameters. A summary of the FPGA hardware resources of proposed chaotic generator and encryption scheme is presented in Table 9. The proposed chaotic generator and encryption system achieve a maximum frequency of 73 and 73.10 MHz respectively. As can be seen the frequency of the Encryption is higher than the chaotic generator due to the pipelining in the encryption architecture. Also, only 16 LSBs of the chaotic generator outputs are used in the encryption design which have less processing delay.

TABLE 9. Chaotic generator and encryption scheme hardware resources on Artix 7.

Logic Utilization	Slice LUT	Slice Reg	Maximum Frequency "MHz"
Encryption	582	294	73.10
Chaotic generator	520	262	73

Figure 21 presents the MATLAB (floating-point realization) and FPGA (fixed-point realization) results, where Fig. 21(a), 21(b) and 21(c) show the I-V plane of proposed memristor model for MATLAB vs FPGA results, error between MATLAB and FPGA for proposed memristor model and error between MATLAB and FPGA for proposed chaotic system, respectively. It is worth to note that the error

between the numerical simulation and experiment is negligible because it is due to the quantization only. However, the error is large in the chaotic case because the numerical simulation is floating-point realization and the experiments are based on fixed-point where the quantization causes the chaotic behavior to exhibit different trajectories which result in a large difference error. Meaning that the floating-point realization can be used to decipher the ciphered data with chaotic system implemented with fixed-point and vice versa.

VII. CONCLUSION

In this work, a digital memristor emulator was presented that can be tuned to produce both discrete and continuous behaviors. In addition, a 4D chaotic attractor based on the proposed memristor emulator was proposed. Moreover, the developed chaotic circuit was utilized in a speech encryption scheme based on S-box and bit permutations. The performance of the proposed scheme was validated using multiple evaluation criteria. The proposed discrete and continuous memristor model, chaotic system and speech encryption scheme were realized on FPGA. The proposed s-box uses 1/4 of LUT in previous works, which is a significant reduction in the required hardware resources. In addition, the versatility and flexibility of the proposed memristor can generate any number of states with the same hardware cost, which highly reduces the hardware for memristor having large number of states. The proposed memristor achieved maximum frequency of 189.494 MHz compared with previous works that achieved 138.17 MHz and 122.2 MHz for 8-states DMM and CMM respectively. Furthermore, compared to previous related work the proposed hardware architectures offer a number of advantages that include moderate usage of hardware resources and the memristor state is programmable on the fly through a parameter and with no hardware overhead.

REFERENCES

[1] A. G. Radwan and M. E. Fouda, *On the Mathematical Modeling of Memristor, Memcapacitor, and Meminductor*, vol. 26. Cham, Switzerland: Springer, 2015.

[2] H. Abunahla, B. Mohammad, L. Mahmoud, M. Darweesh, M. Alhawari, M. A. Jaoude, and G. W. Hitt, "Memsens: Memristor-based radiation sensor," *IEEE Sensors J.*, vol. 18, no. 8, pp. 3198–3205, Apr. 1, 2018.

[3] A. Buscarino, L. Fortuna, M. Frasca, and L. V. Gambuzza, "A chaotic circuit based on Hewlett-Packard memristor," *Chaos*, vol. 22, no. 2, 2012, Art. no. 023136.

[4] R. Wu and C. Wang, "A new simple chaotic circuit based on memristor," *Int. J. Bifurcation Chaos*, vol. 26, no. 9, 2016, Art. no. 1650145.

[5] S. H. Jo, T. Chang, I. Ebong, B. B. Bhadviya, P. Mazumder, and W. Lu, "Nanoscale memristor device as synapse in neuromorphic systems," *Nano Lett.*, vol. 10, no. 4, pp. 1297–1301, 2010.

[6] M. A. Lebdeh, H. Abunahla, B. Mohammad, and M. Al-Qutayri, "An efficient heterogeneous memristive xnor for in-memory computing," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 64, no. 9, pp. 2427–2437, Sep. 2017.

[7] H. Abunahla, M. A. Jaoude, and C. J. O'Kelly, and B. Mohammad, "Sol-gel/drop-coated micro-thick tio2 memristors for γ-ray sensing," *Mater. Chem. Phys.*, vol. 184, pp. 72–81, Dec. 2016.

[8] A. Mazady, M. T. Rahman, D. Forte, and M. Anwar, "Memristor PUF—A security primitive: Theory and experiment," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 5, no. 2, pp. 222–229, Jun. 2015.

[9] Y. Halawani, B. Mohammad, D. Homouz, M. Al-Qutayri, and H. Saleh, "Modeling and optimization of memristor and STT-RAM-based memory for low-power applications," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 24, no. 3, pp. 1003–1014, Mar. 2016.

[10] Y. Halawani, B. Mohammad, M. Abu-Lebdeh, M. Al-Qutayri, and S. F. Al-Sarawi, "ReRAM-based in-memory computing for search engine and neural network applications," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 9, no. 2, pp. 388–397, Jun. 2019.

[11] C. Li, F. Min, Q. Jin, and H. Ma, "Extreme multistability analysis of memristor-based chaotic system and its application in image decryption," *AIP Adv.*, vol. 7, no. 12, Nov. 2017, Art. no. 125204.

[12] B. Mohammad, M. A. Jaoude, V. Kumar, D. M. A. Homouz, H. A. Nahla, M. Al-Qutayri, and N. Christoforou, "State of the art of metal oxide memristor devices," *Nanotechnol. Rev.*, vol. 5, no. 3, pp. 311–329, 2016.

[13] D. Biolek, V. I. E. R. A. Biolkova, Z. Kolka, and Z. Biolek, "Passive fully floating emulator of memristive device for laboratory experiments," *Adv. Electr. Comp. Eng.*, 2015, pp. 112–116.

[14] A. S. Elwakil, M. E. Fouda, and A. G. Radwan, "A simple model of double-loop hysteresis behavior in memristive elements," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 60, no. 8, pp. 487–491, Aug. 2013.

[15] A. G. Alharbi, M. E. Fouda, Z. J. Khalifa, and M. H. Chowdhury, "Electrical nonlinearity emulation technique for current-controlled memristive devices," *IEEE Access*, vol. 5, pp. 5399–5409, 2017.

[16] A. I. Hussein and M. E. Fouda, "A simple MOS realization of current controlled memristor emulator," in *Proc. 25th Int. Conf. Microelectron. (ICM)*, Dec. 2013, pp. 1–4.

[17] M. T. Abuelma'atti and Z. J. Khalifa, "A new memristor emulator and its application in digital modulation," *Analog Integr. Circuits Signal Process.*, vol. 80, no. 3, pp. 577–584, 2014.

[18] I. Vourkas, A. Abusleme, V. Ntinas, G. C. Sirakoulis, and A. Rubio, "A digital memristor emulator for FPGA-based artificial neural networks," in *Proc. 1st IEEE Int. Verification Secur. Workshop (IVSW)*, Jul. 2016, pp. 1–4.

[19] E. M. Hamed, M. E. Fouda, A. G. Alharbi, and A. G. Radwan, "Experimental verification of triple lobes generation in fractional memristive circuits," *IEEE Access*, vol. 6, pp. 75169–75180, 2018.

[20] E. M. Hamed, M. E. Fouda, and A. G. Radwan, "Multiple pinch-off points in memristive equations: Analysis and experiments," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 66, no. 8, pp. 3052–3063, Aug. 2019.

[21] M. F. Tolba, M. E. Fouda, H. G. Hezayyin, A. H. Madian, and A. G. Radwan, "Memristor FPGA ip core implementation for analog and digital applications," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 66, no. 8, pp. 1381–1385, Aug. 2019.

[22] G. Wang and, "Digital model of TiO_2 memristor for field-programmable gate array," *J. Eng.*, vol. 2014, no. 3, pp. 90–92, Mar. 2014.

[23] A. Solak and S. Herdem, "A piece wise linear memristor model with switches," *Int. J. Model. Optim.*, vol. 6, no. 2, p. 124, Apr. 2016.

[24] E. Solan and K. Ochs, "Generic wave digital emulation of memristive devices," vol. 14, no. 8, 2017, *arXiv:1709.07873*. [Online]. Available: https://arxiv.org/abs/1709.07873

[25] M. Hu, H. Li, Y. Chen, Q. Wu, G. S. Rose, and R. W. Linderman, "Memristor crossbar-based neuromorphic computing system: A case study," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 10, pp. 1864–1878, Oct. 2014.

[26] X. Liu, M. Mao, B. Liu, B. Li, Y. Wang, H. Jiang, M. Barnell, Q. Wu, J. Yang, H. Li, and Y. Chen, "Harmonica: A framework of heterogeneous computing systems with memristor-based neuromorphic computing accelerators," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 63, no. 5, pp. 617–628, May 2016.

[27] V. Ntinas, I. Vourkas, A. Abusleme, G. C. Sirakoulis, and A. Rubio, "Experimental study of artificial neural networks using a digital memristor simulator," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 10, pp. 5098–5110, Oct. 2018.

[28] H. Abunahla, B. Mohammad, D. Homouz, and C. J. Okelly, "Modeling valance change memristor device: Oxide thickness, material type, and temperature effects," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 63, no. 12, pp. 2139–2148, Dec. 2016.

[29] B. Muthuswamy and P. P. Kokate, "Memristor-based chaotic circuits," *IETE Tech. Rev.*, vol. 26, no. 6, pp. 417–429, 2009.

[30] W. Sun, C. Li, and J. Yu, "A memristor based chaotic oscillator," in *Proc. Int. Conf. Commun., Circuits Syst.*, Jul. 2009, pp. 955–957.

[31] P. C. Rech and H. A. Albuquerque, "A hyperchaotic chua system," *Int. J. Bifurcation Chaos*, vol. 19, no. 11, pp. 3823–3828, 2009.

[32] B. Bao, G. Shi, J. Xu, Z. Liu, and S. Pan, "Dynamics analysis of chaotic circuit with two memristors," *Sci. China Technol. Sci.*, vol. 54, no. 8, pp. 2180–2187, Aug. 2011.
[33] B. Bo-Cheng, X. Jian-Ping, Z. Guo-Hua, M. Zheng-Hua, and Z. Ling, "Chaotic memristive circuit: Equivalent circuit realization and dynamical analysis," *Chin. Phys. B*, vol. 20, no. 12, 2011, Art. no. 120502.
[34] H. Xi, S. Yu, C. Zhang, and Y. Sun, "Generation and implementation of hyperchaotic chua system via state feedback control," *Int. J. Bifurcation Chaos*, vol. 22, no. 5, 2012, Art. no. 1250119.
[35] Y. Li, X. Huang, and M. Guo, "The generation, analysis, and circuit implementation of a new memristor based chaotic system," *Math. Problems Eng.*, vol. 2013, Oct. 2013, Art. no. 398306.
[36] Q. Li, H. Zeng, and J. Li, "Hyperchaos in a 4D memristive circuit with infinitely many stable equilibria," *Nonlinear Dyn.*, vol. 79, no. 4, pp. 2295–2308, Mar. 2015.
[37] F. Yuan, G. Wang, and X. Wang, "Extreme multistability in a memristor-based multi-scroll hyper-chaotic system," *Chaos, Interdiscipl. J. Nonlinear Sci.*, vol. 26, no. 7, 2016, Art. no. 073107.
[38] Z.-H. Lin and H.-X. Wang, "Image encryption based on chaos with pwl memristor in chua's circuit," in *Proc. Int. Conf. Comun. Circuit. Syst (ICCCAS)*, Jul. 2009, pp. 964–968.
[39] B. Wang, F. C. Zou, and J. Cheng, "A memristor-based chaotic system and its application in image encryption," *Optik*, vol. 154, pp. 538–544, Feb. 2018.
[40] R. Ranjan, P. M. Ponce, A. Kankuppe, B. John, L. A. Saleh, D. Schroeder, and W. H. Krautschneider, "Programmable memristor emulator ASIC for biologically inspired memristive learning," in *Proc. 39th Int. Conf. Telecommun. Signal Process. (TSP)*, Jun. 2016, pp. 261–264.
[41] K. Rajagopal, A. Karthikeyan, and A. Srinivasan, "Dynamical analysis and FPGA implementation of a chaotic oscillator with fractional-order memristor components," *Nonlinear Dyn.*, vol. 91, no. 3, pp. 1491–1512, Feb. 2018.
[42] A. H. Elsafty, M. F. Tolba, L. A. Said, A. H. Madian, and A. G. Radwan, "FPGA speech encryption realization based on variable S-box and memristor chaotic circuit," in *Proc. 30th Int. Conf. Microelectron. (ICM)*, Dec. 2018, pp. 152–155.
[43] S. Lian, *Multimedia Content Encryption: Techniques and Applications*. Berlin, Germany: Auerbach Publications, 2008.
[44] Y. Hosokawa, Y. Nishio, and A. Ushida, "A design method of chaotic circuits using an oscillator and a resonator," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, vol. 3, May 2001, pp. 373–376.
[45] G. E. Testoni and P. C. Rech, "Dynamics of a particular lorenz type system," *Int. J. Mod. Phys. C*, vol. 21, no. 7, pp. 973–982, 2010.
[46] A. Wolf, J. B. Swift, H. L. Swinney, and J. A. Vastano, "Determining Lyapunov exponents from a time series," *Phys. D, Nonlinear Phenomena*, vol. 16, no. 3, pp. 285–317, 1985.
[47] M. F. Tolba, W. S. Sayed, A. G. Radwan, and S. K. Abd-El-Hafiz, "Chaos-based hardware speech encryption scheme using modified tent map and bit permutation," in *Proc. 7th Int. Conf. Modern Circuits Syst. Technol. (MOCAST)*, May 2018, pp. 1–4.
[48] M. F. Tolba, W. S. Sayed, A. G. Radwan, and S. K. Abd-El-Hafiz, "Fpga realization of speech encryption based on modified chaotic logistic map," in *Proc. IEEE Int. Conf. Ind. Technol. (ICIT)*, Feb. 2018, pp. 1412–1417.
[49] W. S. Sayed, M. F. Tolba, A. G. Radwan, and S. K. Abd-El-Hafiz, "Fpga realization of a speech encryption system based on a generalized modified chaotic transition map and bit permutation," *Multimedia Tools Appl.*, vol. 78, pp. 16097–16127, Jun. 2019.
[50] L. Yi, X. Tong, Z. Wang, M. Zhang, H. Zhu, and J. Liu, "A novel block encryption algorithm based on chaotic s-box for wireless sensor network," *IEEE Access*, vol. 7, pp. 53079–53090, 2019.
[51] *The History Place—Great Speeches Collection: Ronald Reagan Speech 'Tear Down This Wall'*. Accessed: 2018. [Online]. Available: http://www.historyplace.com/speeches/reagan-tear-down.htm
[52] *ITU-T Test Signals for Telecommunication Systems*, Document Rec. ITU-T P.501, 2017, Accessed: Oct. 3, 2017. [Online]. Available: https://www.itu.int/net/itu-t/sigdb/menu.aspx
[53] L. E. Bassham, III, A. L. Rukhin, J. Soto, J. R. Nechvatal, M. E. Smid, E. B. Barker, S. D. Leigh, M. Levenson, M. Vangel, D. L. Banks, N. A. Heckert, J. F. Dray, and S. Vo, "A statistical test suite for random and pseudorandom number generators for cryptographic applications," Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Tech. Rep. SP 800-22 Rev. 1a, 2010.

MOHAMMED F. TOLBA received the B.Sc. degree in electronics and communications engineering from Fayoum University, in 2014, and the M.Sc. degree in micro-electronics system design (MSD) from Nile University. He is currently a Research Associate with SOC, Khalifa University. His research is focused on digital design and implementation of deep learning, convolution neural network (CNN), lightweight encryption, low-power approximation techniques, graphics processing unit (GPU) architectures, computer arithmetic, fractional order circuits, memristor, and chaotic circuits. He has authored or coauthored over 25 journal and conference papers. He received the Best Paper Award in modern circuits and systems technologies (MOCAST) 2017.

WAFAA S. SAYED received the B.Sc. degree (Hons.) from the Electronics and Communications Engineering Department, and the M.Sc. degree from the Engineering Mathematics Department, Faculty of Engineering, Cairo University, Egypt, in 2012 and 2015, respectively, where she is currently pursuing the Ph.D. degree. She is currently an Assistant Lecturer with the Faculty of Engineering, Cairo University. Her research interests include chaos theory, chaotic cryptography, fractional dynamics, and mathematical software. She was awarded by the faculty for being a distinguished Assistant Lecturer. She won the 1st Place Prize in the Masters Forum competition of the 22nd International Conference on Electronics, Circuits, and Systems (ICECS 2015) and Poster Award in the 13th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON 2016).

MOHAMMED E. FOUDA received the B.Sc. degree (Hons.) in electronics and communications engineering and the M.Sc. degree in engineering mathematics from the Faculty of Engineering, Cairo University, Cairo, Egypt, in 2011 and 2014, respectively. He is currently pursuing the Ph.D. degree with the University of California-Irvine, Irvine, CA, USA. His Ph.D. research is focused on enabling the emerging devices (Memristor/RRAM) for memory and neuromorphic applications. He has authored or coauthored over 70 journal and conference papers. His research interests include brain-inspired computing, neuromorphic circuits and systems, resistive memories, modeling and analysis of mem-elements based circuits, fractional-order circuits, and analog mixed circuits. He has served as a Technical Program Committee Member of the International Conference on Microelectronics (ICM) 2018, and the IEEE International Conference on Design & Test Integrated Micro Nano-Systems (DTS), in 2019. He also serves as a Peer-Reviewer for many prestigious journals and conferences. He received the Best Paper Award in ICM 2013 and the Broadcom Foundation Fellowship, from 2016 to 2017.

HANI SALEH received the B.Sc. degree in electrical engineering from the University of Jordan, the M.Sc. degree in electrical engineering from the University of Texas at San Antonio, and the Ph.D. degree in computer engineering from the University of Texas at Austin.

He has been an Associate Professor of electronic engineering with Khalifa University, since 2012. He is a co-founder and an active researcher with the Khalifa University Research Center (KSRC) and the System on Chip Research Center (SOCC), where he led multiple the IoT projects for the development of wearable blood glucose monitoring SOC and a mobile surveillance SOC. He has a total of 19 years of industrial experience in ASIC chip design, microprocessor design, DSP core design, graphics core design, and embedded system design. Prior to joining Khalifa University, he worked for many leading semi-conductor design companies, including a Senior Chip Designer (Technical Lead) at Apple incorporation, Intel (ATOM mobile microprocessor design), AMD (Bobcat mobile microprocessor design), Qualcomm (QDSP DSP core design for mobile SOCs), Synopsys (designed the I2C DW IP included in Synopys DesignWare library), Fujitsu (SPARC compatible high performance microprocessor design), and Motorola Australia. His research interests include the IoT design, deep learning, AI hardware design, DSP algorithms design, DSP hardware design, computer architecture, computer arithmetic, SOC design, ASIC chip design, FPGA design, and automatic computer recognition. He has 12 issued US patents, eight pending patent application, and over 100 articles published in peer-reviewed conferences and journals in the areas of digital system design, computer architecture, DSP, and computer arithmetic.

MAHMOUD AL-QUTAYRI received the B.Eng. degree from Concordia University, Montreal, Canada, in 1984, the M.Sc. degree from the University of Manchester, U.K., in 1987, and the Ph.D. degree from the University of Bath, U.K., in 1992, all in electrical and electronic engineering. He is currently a Full Professor with the Department of Electrical and Computer Engineering and the Associate Dean for Graduate Studies, College of Engineering, Khalifa University, UAE. Prior to joining Khalifa University, he worked at De Montfort University, U.K., and the University of Bath, U.K. He has authored or coauthored numerous technical papers in peer-reviewed international journals and conferences. He has also coauthored a book entitled *Digital Phase Lock Loops: Architectures and Applications* and has edited a book entitled *Smart Home Systems*. This is in addition to a number of book chapters and patents. His current research interests include wireless sensor networks, embedded systems design, in-memory computing, mixed-signal integrated circuits design and test, and hardware security. His professional services include serving on the editorial board of some journals as well as membership of the steering, organizing, and technical program committees of many international conferences.

BAKER MOHAMMAD (M'04–SM'13) received the Ph.D. degree from the University of Texas at Austin, in 2008, the M.S. degree from Arizona State University, Tempe, AZ, USA, and the B.S. degree from the University of New Mexico, Albuquerque, NW, USA, all in ECE.

He is currently the Director of the System on Chip Center, and an Associate Professor with the EECS, Khalifa University. Prior to joining Khalifa University, he was the Senior Staff Engineer/Manager of Qualcomm, Austin, TX, USA, for six years, where he was involved in designing high-performance and low power DSP processor used for communication and multimedia application. Before joining Qualcomm, he worked at Intel Corporation, for ten years, on a wide range of microprocessors design from high-performance, server chips > 100Watt (IA-64), to mobile embedded processor low power sub 1 watt (xscale). He has over 16 year's industrial experience in microprocessor design with an emphasis on memory, low power circuit, and physical design. His research interests include VLSI, power-efficient computing, high yield embedded memory, emerging technology, such as memristor, STTRAM, and in-memory-computing, hardware accelerators for cyber-physical systems. In addition, he is involved in microwatt range computing platform for wearable electronics and WSN focusing on energy harvesting, power management, and power conversion, including efficient dc/dc and ac/dc convertors. He has authored/coauthored over 100 refereed journals and conference proceedings, three books, 18 US patents, multiple invited seminars/panelist, and the presenter of three conference tutorials, including one tutorial on energy harvesting and power management for WSN at the 2015 (ISCAS).

Dr. Mohammad has received several awards, including the KUSTAR Staff Excellence Award in intellectual property creation, the IEEE TVLSI Best Paper Award, the 2016 IEEE MWSCAS Myrill B. Reed Best Paper Award, the Qualcomm Qstar Award for excellence on performance, and leadership, the SRC Techon Best Session Papers for 2016 and 2017, the 2009 Best Paper Award for Qualcomm Qtech Conference, and Intel Involve in the Community Award for volunteer and impact on the community. He is an Associate Editor of the IEEE TRANSACTION ON VLSI (TVLSI), and the *Microelectronics Journal* (Elsevier). He participates in many technical committees at IEEE conferences and reviews for journals, including TVLSI and the IEEE Circuits and Systems.

AHMED G. RADWAN was the Former Director of the Nanoelectronics Integrated Systems Center (NISC), Nile University, Egypt. He was the Former Director of the Technical Center for Carrier Development (TCCD), Cairo University. He was a Visiting Professor of the Computational Electromagnetic Lab (CEL), Electrical and Computer Engineering Department (ECE), McMaster University, Canada, from 2008 to 2009, and then he was selected to be a part of the first foundation research teams to join the King Abdullah University of Science and Technology (KAUST), from 2009 to 2011. He is currently the Vice President for Research with Nile University, and a Professor with the Engineering Mathematics and Physics Department, Cairo University, Egypt. He has more than 290 articles, H-index 37, and more than 4300 citations based on Scopus database. He is the co-inventor of six US patents, author/coauthor of seven international books, as well as 18 book chapters in the highly ranked publishers such as Elsevier and Springer. His research interests include interdisciplinary concepts between mathematics and engineering applications such as fractional-order systems, bifurcation, chaos, memristor, and encryption.

Dr. Radwan is a member of the Applied Research Council, the Specialized Scientific Councils (SSC), the Academy of Scientific Research and Technology (ASRT), Egypt, a member of the National Committee of Mathematics, ASRT, Egypt, and a MC Observer to COST Action CA15225 http://www.cost.eu/COST_Actions/ca/CA15225. He was selected as a member of the First Scientific Council of the Egyptian Young Academy of Sciences (EYAS) as well as in the First Scientific Council of the Egyptian Center for the Advancement of Science, Technology, and Innovation (ECASTI) to empower and encourage Egyptian young scientists in science and technology and build knowledge-based societies. He received the Scopus Award in Engineering and Technology, in 2019, for his publications during 2014 to 2018, the State Excellence Award in Advanced Technological Sciences, in 2018, the Cairo University Excellence Award for research in the engineering sciences, in 2016, the Abdul Hameed Shoman Award for Arab Researchers in basic sciences, in 2015, the State Achievements Award for research in mathematical sciences, in 2012, the Cairo University Achievements Award for research in the engineering sciences, in 2013, and the Best Researcher Awards Nile University 2015 and 2016, respectively.