Integration, the VLSI Journal, Elsevier, vol. 72, pp. 163-170, May 2020.

FPGA Implementation of a Chaotic Oscillator With Odd/Even Symmetry and Its Application

M. F. Tolba*

System on Chip Center (SoCC), Khalifa University of Science and Technology, Abu Dhabi, 127788, Emirates

A. S. Elwakil†

Department of Electrical and Computer Engineering, University of Sharjah, 27272, Emirates

H. Orabi, M. Elnawawy, F. Aloul, A. Sagahyroon

Department of Computer Science and Engineering, American University of Sharjah, Emirates

A. G. Radwan‡

Nanoelectronics Integrated Systems Center (NISC) Research Center, Nile University, Cairo, Egypt

We propose a mathematical system capable of exhibiting chaos with a chaotic attractor which is odd symmetrical in the $x-y$ phase plane but even symmetrical in the $x-z$ and $y-z$ phase planes respectively. A hardware implementation of the system is done on a digital FPGA platform for verification. The system is also attractive in the sense that (i) its dynamics are single-parameter controlled and (ii) it inherently generates two chaotic clock signals. As an application, an FPGA design methodology using this oscillator for speech encryption is demonstrated. The security of the proposed encryption scheme is evaluated and results confirm its robustness. Due to the efficient hardware resource utilization, the encrypted system delivers a throughput of 1.3Gbit/sec using a Xilinx Kintex 7.

1. Introduction

Numerous chaotic systems and chaotic oscillator circuits have been introduced over the years and employed in many applications. Some of these systems have unique features from a nonlinear dynamical point of view [1] while others are more focused on simplicity and suitability for circuit implementation [2]. Some of the simplest

*Also with the Nanoelectronics Integrated Systems Center (NISC) Research Center, Nile University, Cairo, Egypt

†Also with the Department of Electrical and Computer Engineering, University of Calgary, Alberta, Canada and the Nanoelectronics Integrated Systems Center (NISC), Nile University, Cairo, Egypt

‡Also with the Dept. of Mathematica and Physics, Faculty of Engineering, Cairo University, Egypt

chaotic circuits include the Wien-type oscillator of [3] and the Colpitts-based family of [4]. Generation of chaos requires the existence of at least one nonlinear function which can be asymmetric (typical of diode characteristics for example) [4], odd-symmetric [5], even-symmetric [6,7], periodic [8], containing hysteresis [9], or based on discrete maps [10]. Chaotic dynamics are widely used to produce pseudo-random number generators and for secure communications and encryption applications [11, 12]. Several recent contributions have particularly focused on speech encryption [13,14].

It is important for verifying the chaotic behavior of a system to choose a proper numerical integration algorithm with a suitable step size particularly if the system contains slow-fast dynamics [15]. Different numerical methods have been used to simulate and to implement chaotic systems such as the Euler, 4^{th}-order Runge-Kutta, or trigonometric polynomials methods. The trigonometric polynomials method provides better accuracy compared with Euler and 4th-order Runge-Kutta [16]. However, the choice of a numerical integration method has its implications on the hardware realization of the system and a compromise between accuracy and hardware complexity is usually exercised [17].

Chaotic oscillators have been implemented using discrete electronic devices and in CMOS technology using integrated circuits to achieve features such as low voltage or low power operation [18]. However, FPGAs have recently become the most common platform for experimental validation of chaotic systems because they are faster, more reliable [19–21] and modular [22]. A memristive chaotic oscillator based on an FPGA emulator was recently introduced and verified in [23]. In [24], an efficient and secure embedded chaotic cryptosystem was introduced to process digital images and perform voice recognition. Three technologies were used for the implementations: (1) a Spartan 3151-1600 Xilinx FPGA kit; (2) a 64-bit Raspberry Pi 3 single-board computer; and (3) a speech recognition chip (SRC) manufactured by Sunplus. This shows that FPGAs can be easily integrated with other components.

Meanwhile, a system with an X-shaped nonlinearity; which is both even- and odd- symmetric was recently introduced in [25] and verified using an FPGA. The clear complexity and many parameters of that system is the main motivation behind the work presented here, which introduces a single-parameter controlled chaotic oscillator with both types of nonlinear functions. The proposed system is inherently capable of generating two chaotic clock signals. Chaotic clocking has become increasingly important in several applications as a replacement of simple periodic clocking [26] particularly in power electronic systems [27]. A variable fractional-order version of this system was demonstrated in [28]. Furthermore, we employ the chaotic oscillator in a speech encryption application which is increasingly important in multimedia tools [29]. The purpose is to hide perceptual and statistical characteristics of the speech file by scrambling it. The reverse operation at the decryption side ensures the retrieval of the information [30]. Due to the simplicity and efficiency of the proposed chaos generator, its FPGA realization achieves a throughput of 1.5Gbit/sec and when employed in the speech encryptor, a throughput of 1.3Gbit/sec is obtained.

2. Chaotic Oscillator System

The proposed third-order system is described by the following set of differential equations.

$$\begin{aligned}\dot{x} &= -ax + y \cdot \mathrm{sgn}(z)\\ \dot{y} &= x\\ \dot{z} &= 1 - |y|\end{aligned} \tag{1}$$

Parameter a is the only constant control parameter. It is clear that the even-symmetrical nonlinearity in this system is the absolute value function $|y|$ while the odd-symmetrical nonlinearity is the signum function $\mathrm{sgn}(z)$. However, $|y|$ is also equal to $y \cdot \mathrm{sgn}(y)$ which simplifies the implementation of the system on a digital platform significantly since the signum functions require simple switches. Furthermore, this means that the system can inherently generate two independent chaotic clocks; one related to the odd-symmetrical nonlinearity and the other related to the even-symmetrical nonlinearity. It can be shown that the linearized dynamics of this system are governed by the second-order oscillator

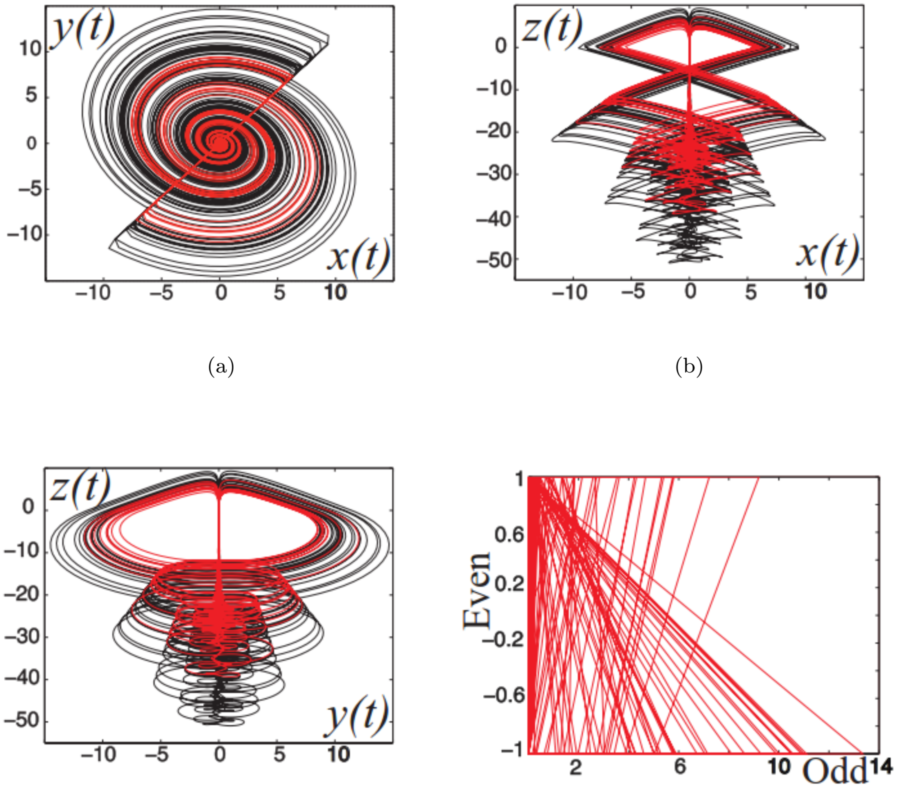
$$\ddot{y} + a\dot{y} + y = 0 \tag{2}$$

which means that the real equilibrium point at the origin $(x_0, y_0, z_0) = (0, 0, 0)$ is always unstable for any $a > 0$. We shall not attempt to study in further detail the chaotic dynamics of this system as our focus here is on the implementation and encryption application.

Figures 1(a), 1(b) and 1(c) show the projections of the chaotic attractor in the $x-y$, $x-z$ and $y-z$ phase spaces, respectively, obtained from numerical simulations of (1) using a 4^{th} Runge-Kutta algorithm with $a = 0.3$ and a step size of 0.01. The odd-symmetry of the attractor is clear in the $x - y$ plane while the even-symmetry is clear with respect to the z axis. In Fig. 1(d), the dynamic even and odd functions; i.e. $|y(t)|$ and $\mathrm{sgn}(z(t))$, are plotted versus each other showing an intersecting X-shape as expected. Figure 2 presents the bifurcation diagram of z_{max} versus a. This diagram shows that the magnitude of z_{max} increases significantly with increasing the value of a. To avoid this, we operate the system at values of $a < 0.4$. A single positive Lyapunov exponent equal to $+0.083$ was calculated from the $z(t)$ time series at $a = 0.3$.

2.1. *Hardware Implementation*

The hardware architecture of the proposed system is shown in Fig. 3, where the design is built using three registers to store the state variables. Fixed point 32-bit



(a)

(b)

(c)

(d)

Fig. 1: Numerical simulation results of the chaotic oscillator system with $a = 0.3$. Phase space projections are shown in (a,b,c) and the odd versus even nonlinearities are shown in (d)

format and Euler discretization method with a step-size h are used for implementation [31] and the realized equations are

$$x_{n+1} = \left(-ax(n) + y(n)\mathrm{sgn}(z(n))\right) h + x(n), \tag{3a}$$

$$y_{n+1} = \left(x(n)\right) h + y(n), \tag{3b}$$

$$z_{n+1} = \left(1 - y(n)\mathrm{sgn}(y(n))\right) h + z(n). \tag{3c}$$

The Euler method is used for the hardware implementation because it requires less hardware compared with the Runge-Kutta method. Three combinational circuits are proposed to compute the numerical solution of variables x, y, and z. In each computation step, one adder and an 8 bit shift right operation are used to add

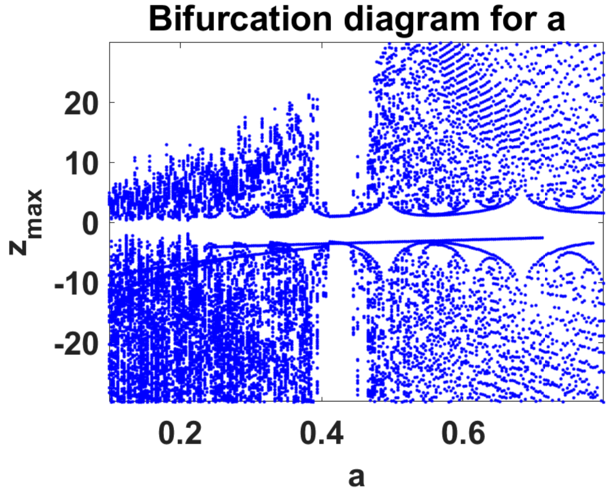


Fig. 2: Bifurcation diagram of z_{max} against variable a

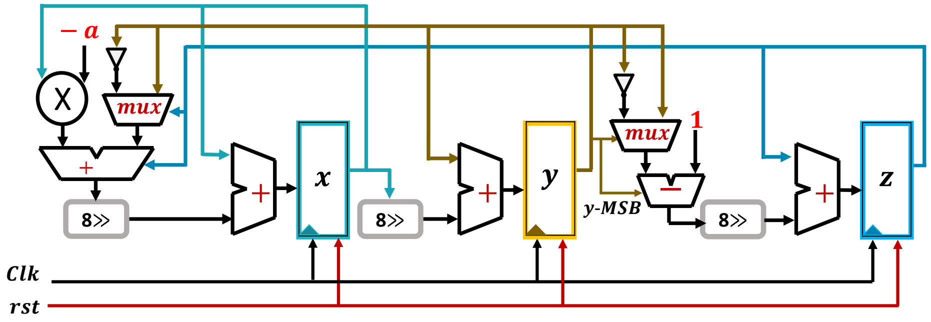
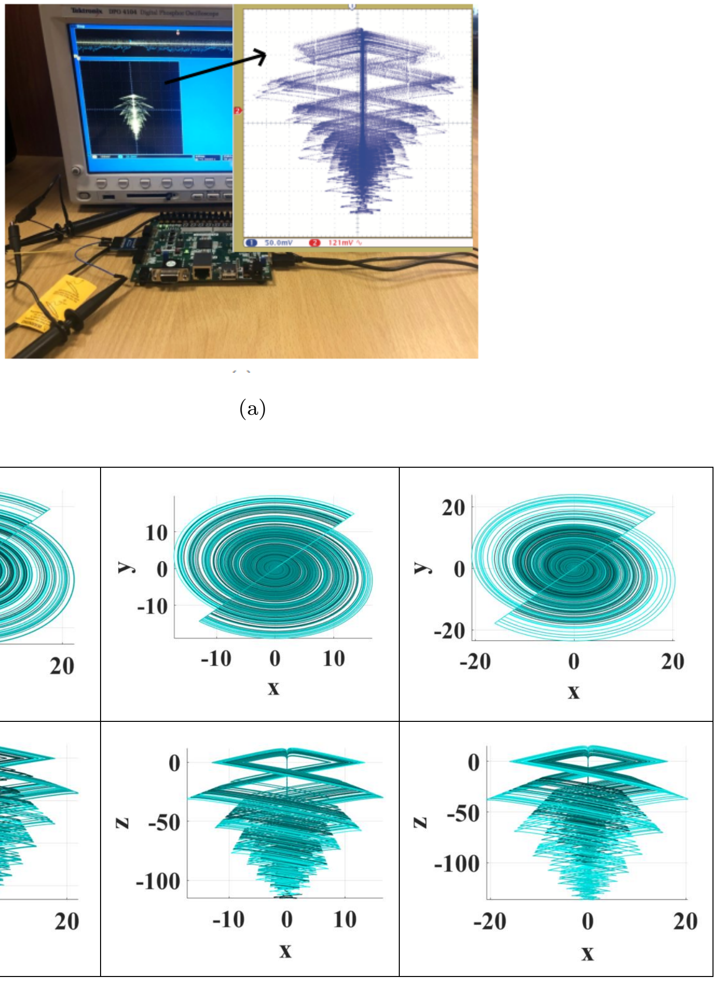


Fig. 3: Complete hardware architecture of the chaotic system implemented on the FPGA

previous values of the state variable with the multiplication of the step size and the current value. The step size h value used in this work is a power of two in order to use simple shift right operation instead of multipliers.
The experimental setup is shown in Fig. 4(a) based on the Nexys 4 XC7A100T FPGA platform. MATLAB and Xilinx ISE 14.5 softwares were used for simulations, RTL code generation and testing. The FPGA generates 12-bit pair outputs which are converted to analog signals using 12-bit digital-to-analog converters with a 1mV resolution. Figure 4(b) shows the experimental projections of the chaotic attractor at three different discretization step values confirming the robustness of the implementation.



(a)

(b)

Fig. 4: Experimental setup using an FPGA platform and chaotic attractor projections when $h = 2^{-5}$ (left), $h = 2^{-6}$ (middle) and $h = 2^{-7}$ (right).

3. Encryption Application

3.1. *Scheme*

Figure 5 shows the proposed encryption algorithm process chart of the chaotic system parameters calculation from the encryption sub-keys. As shown in Fig. 5(a), the encryption sub-keys k_0, k_1, k_2, k_3 drive the parameters calculator block to compute the chaotic parameters x_0, y_0, z_0 and a required for the chaotic oscillator. Then, the oscillator generates three signals x, y and z to drive the encryption scheme to encrypt the input speech signal. Figure 5(b) presents the process of computing the chaotic parameters where each encryption sub-key is concatenated with zero's in the MSB side to complete 40-bits for x_k, y_k, z_k and 34-bits for a_k. The computations of the parameters depend on the following equations [32]:

$$x_0 = x_{fix} + (7'b0000000, k_0) + P_{sum}, \tag{4a}$$

$$y_0 = y_{fix} + (7'b0000000, k_1) + P_{sum}, \tag{4b}$$

$$z_0 = z_{fix} + (9'b000000000, k_2) + P_{sum}, \tag{4c}$$

$$a = a_{fix} + (3'b000, k_3) + P_{sum}. \tag{4d}$$

where $x_{fix}, y_{fix}, z_{fix}$ and a_{fix} are the chosen initial values of the chaotic parameters. These values are added to the sub-keys and P_{sum} which is the term matching to input dependence and represents the summation of all 16-bits signed integer speech samples. The value P_{sum} improves the resistance to differential attacks based on [33], and is calculated as follows:

$$P_{sum} = mod\left(\sum_{i=1}^{N} S_i, 10\right)/1000, \tag{5}$$

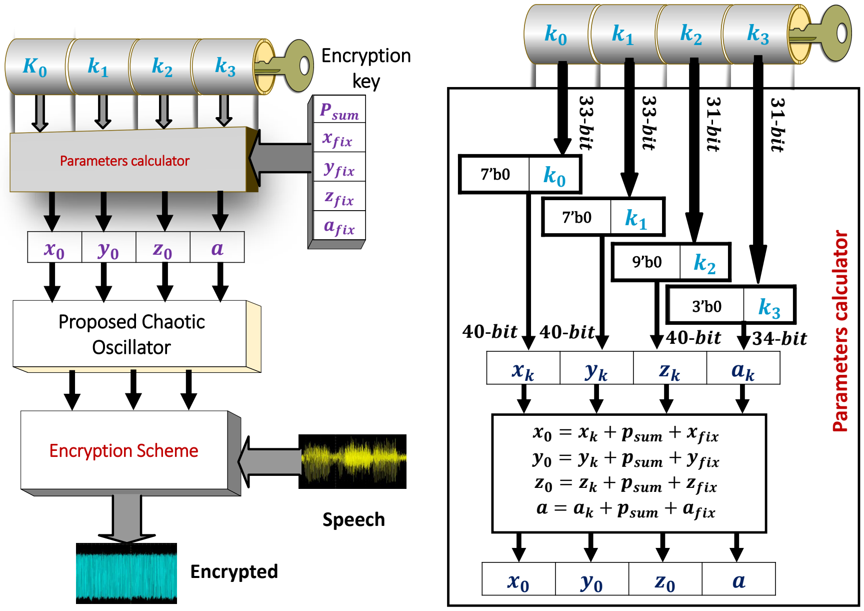
where S_i depict all the sample values of the input speech with N samples. One of the most popular attacks is the Brute force attack which all secured ciphers must be protected against. A strong encryption algorithm must have a large key space and in the proposed encryption system, a key space of 2^{128} distributed for the encryption sub-keys k_0, k_1, k_2, k_3 as $(33 + 33 + 31 + 31) = 128$.

The pipeline hardware architecture of the proposed encryption system is shown in Fig. 6(a), the four sub-keys are based on initial values and one independent parameter of the proposed chaotic generator. The encryption scheme is based on an XOR gate and feedback, as shown in Fig. 6(b). The input speech signal is represented in 16-bits fixed-point format and a multiplexer selects between x and y where the selector is z. The output of the multiplexer is XORed with the feedback and the speech signal to produce the encrypted signal. In the decryption hardware architecture, the reverse process takes place, as shown also in Fig. 6.

3.2. *Performance Evaluation*

Two test files were used to evaluate the performance of the encryption system. The first file is *this is a test* with original sampling frequency of 16kHz while the second file is *Ronald Reagan said Mr. Gorbachev, tear down this wall!* with a sampling frequency of 44.1kHz.

Figure 7 illustrates the time waveforms, spectrograms, and histograms of the first original speech file and encrypted signal. The spectrogram represents the variation of the spectral density with time and frequency, which is specified by the strength of the color. The results of the second speech test file are shown in Fig. 8 showing similar excellent performance.



(a) (b)

Fig. 5: (a) The proposed encryption algorithm chart and (b) key process and parameters calculator

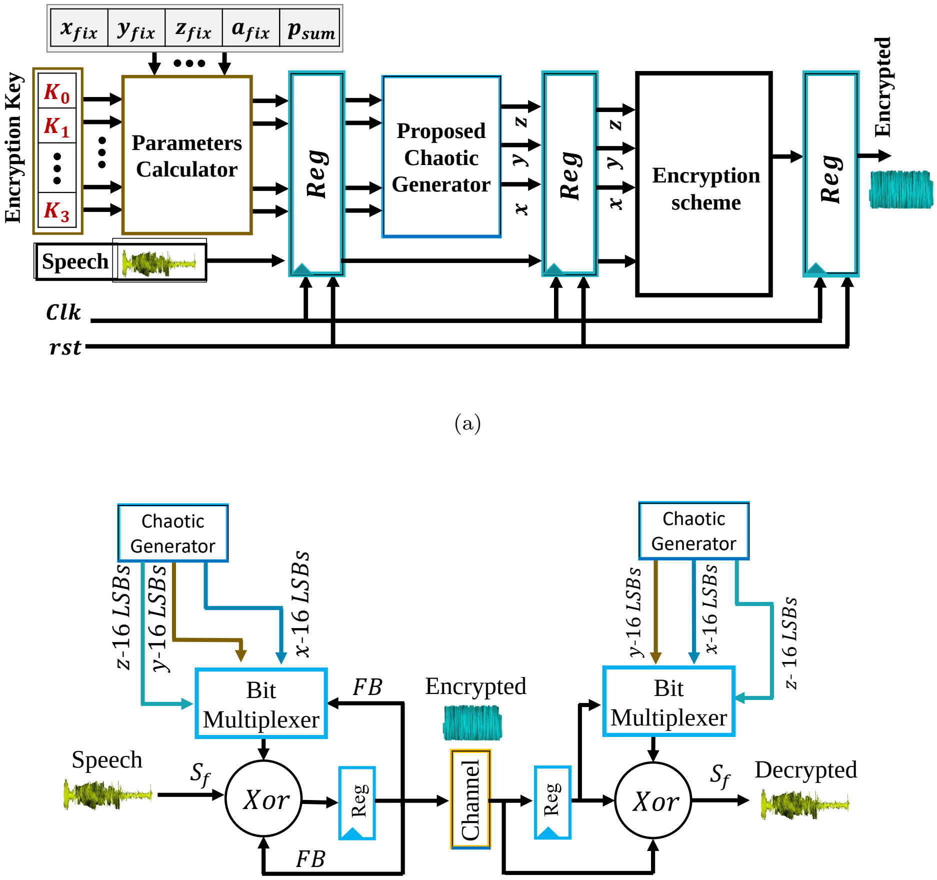
The correlation coefficient is computed as follows:

$$r_{xy} = \frac{cov(x,y)}{\sqrt{D(x)D(y)}}, \tag{6}$$

where $cov(x,y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y))$, $D(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))^2$, $E(x) = \frac{-1}{N}\sum_{i=1}^{N} x_i$, N is the number of samples and x and y are the two signals. The entropy of a speech signal measures the randomness of the signal samples as follows

$$\text{Entropy} = -\sum_{i=1}^{2^p} P(S_i)\log_2 P(S_i), \tag{7}$$

where the best value of entropy is p for a quite encrypted speech signal encoded in p bits/sample, and $P(S_i)$ is the probability of the sample value S_i. The Mean Squared Error (MSE) describes how far the encrypted (or the wrong decrypted signal) is from the original signal. The MSE between the original signal x and another signal



(a)

(b)

Fig. 6: (a) Encryption process block diagram and (b) proposed encryption scheme

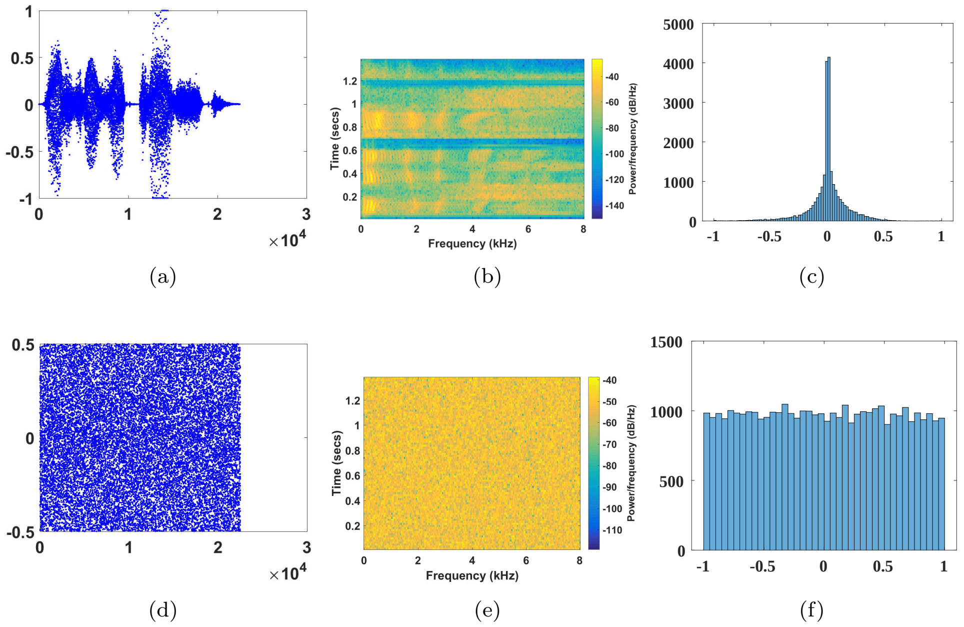
y, each of length N samples, is calculated as follows:

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(x_i - y_i)^2 . \tag{8}$$

For the proposed encryption scheme, Table 1 shows correlation coefficients, MSE and entropy for the two tested speech files.

3.3. *Resistance against differential attacks*

To evaluate the robustness of the proposed encryption scheme to differential attacks, one of the original input speech samples is changed to create a new one $A_i^{'}$. Both original input signal A_i and the new one $A_i^{'}$ (with the same encryption key) are



(a) (b) (c)

(d) (e) (f)

Fig. 7: Time waveforms ((a) and (d)), spectrograms ((b) and (e)), and histograms ((c) and (f)) of the original signal (top) and encrypted signal (bottom) for the speech file *this is a test*

Table 1: r_{xy}, MSE, entropy, UACI (%) and NSCR for the proposed encryption scheme.

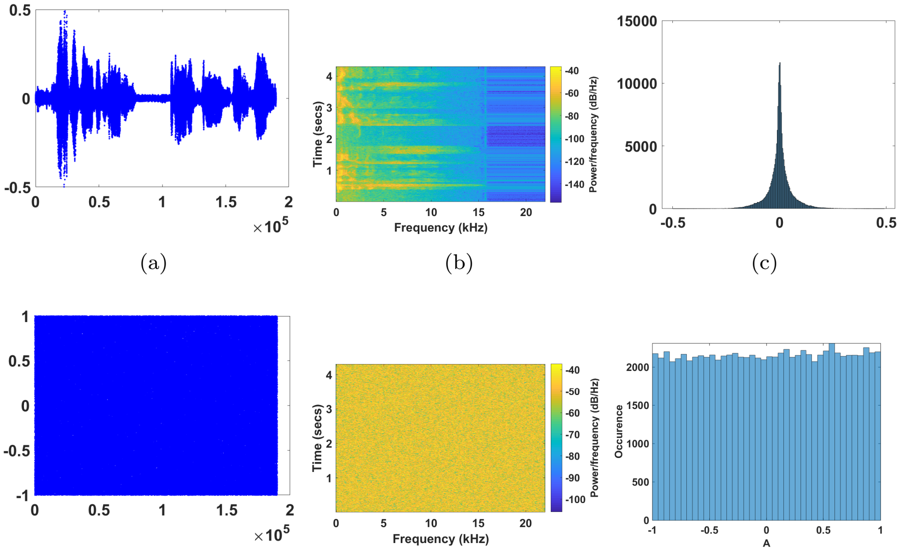
File	r_{xy}	MSE	Entropy	UACI (%)	NSCR (%)
speech file 1	0.0015	32547	15.9617	33.115	99.982
speech file 2	0.0034	32563	15.9613	33.303	99.991

compared in terms of The Unified Average Changing Intensity (UACI) and Number of Samples Change Rate (NSCR) which are given by

$$UACI = \frac{1}{N}(\sum_{i} \frac{|A_i - A_i^{'}|}{2^p - 1})100\%, \tag{9}$$

$$NSCR = \frac{\sum_i D_i}{N} 100\%, \tag{10}$$

where $D_i = 1$ only at $A_i \neq A_i^{'}$ and 0 otherwise. Table 1 presents UACI and NSCR values for the proposed encryption scheme in the last two columns. As can be seen,



(a) (b) (c)

(d) (e) (f)

Fig. 8: Time waveforms ((a) and (d)), spectrograms ((b) and (e)), and histograms ((c) and (f)) of the original signal (top) and encrypted signal (bottom) for speech file *Ronald Reagan said Mr. Gorbachev, tear down this wall!*

the proposed encryption scheme successfully reached the recommended percentage of 33.3 % and 100 % for UACI and NSCR, respectively [34].

3.4. *Results and Discussions*

The AC-97 audio Codec interface is used to pass the input speech signal from a microphone to the FPGA with input speech data resolution up to 18 bits and 48kHz sampling rate. The input speech bits are encrypted as described in the previous section and delivered to the speaker through an auxiliary audio cable interfaced with AC-97 line out. Figure 9 illustrates the oscilloscope experimental waveforms of this speech signal before and after encryption.

A comparison between the proposed encryption system and previous works is given in Table 2. The proposed system achieved a throughput of 1.316 GB/sec compared with 1.1 GB/sec and 0.8 GB/sec for [35] and [36] respectively. The FPGA hardware resources of the proposed chaotic system implementation is shown in Table 2. The encryption cipher presented in [35] has been implemented based on 64-bit registers and key space of 3×2^{124}. In contrast, the proposed encryption system is built based on an effective key space of 2^{128} and is resistant to brute-force attacks. The chaos generator clearly utilizes most of the resources and produces a chaotic clock with throughput of 1.517GB/sec when operating from a clock frequency of

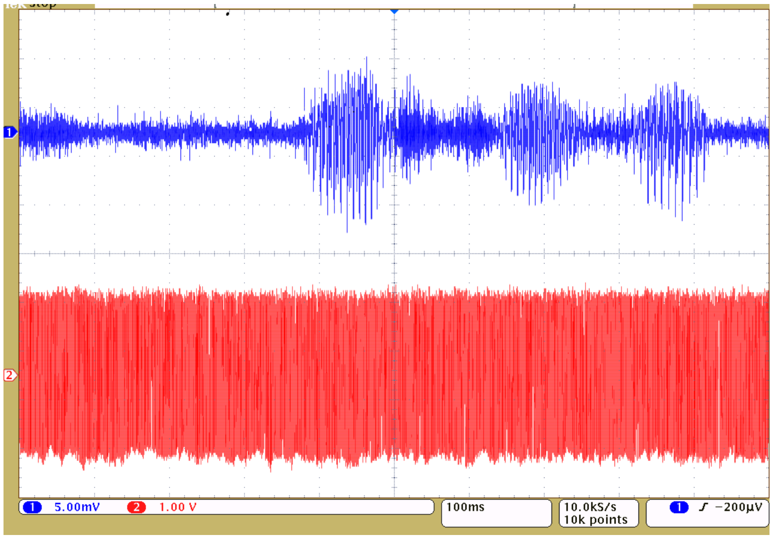


Fig. 9: Oscilloscope shot of experimental results of the proposed speech encryption system. Original signal (top trace) and encrypted signal (lower trace)

Table 2: FPGA hardware resources comparison.

	No. of Slices	No. of Registers	Max Frequency MHz	Throughput Gbit/s	Key space
Chaos generator	560	152	47.401	1.517	-
Encryption system	152	68	82.257	1.316	2^{128}
[35]	354	228	69	1.1	2^{125}
[36]	44	84	51.9	0.8	-

47.4MHz. The encryption part uses less resources and can operate from a maximum clock frequency of 82.3MHz.

4. Conclusion

An extremely simple chaotic generator showing both odd and even symmetries was introduced and experimentally verified on a modular FPGA platform. A speech encryption scheme (based on the proposed generator) was also designed and implemented on an FPGA using Verilog HDL. The chaotic oscillator can be used as a dual chaotic-clock generator. This feature was not the main focus of this work and will be explored further in the future. It is worth noting that chaotic clocks are becoming increasingly important particularly in power electronic applications for improving efficiency and reducing electromagnetic interference [37].

References

1. J. Sprott, Simplest dissipative chaotic flow, *Physics letters A* **228**(4-5) (1997) 271–274.
2. J. C. Sprott, A new chaotic jerk circuit, *IEEE Transactions on Circuits and Systems II: Express Briefs* **58**(4) (2011) 240–243.
3. A. Elwakil and M. Kennedy, High frequency Wien-type chaotic oscillator, *Electronics Letters* **34**(12) (1998) 1161–1162.
4. A. Elwakil and M. Kennedy, A family of Colpitts-like chaotic oscillators, *Journal of the Franklin Institute* **336**(4) (1999) 687–700.
5. G.-Q. Zhong, Implementation of Chua's circuit with a cubic nonlinearity, *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications* **41**(12) (1994) 934–941.
6. M. A. Zidan, A. G. Radwan and K. N. Salama, Controllable V-shape multiscroll butterfly attractor: System and circuit implementation, *International Journal of Bifurcation and Chaos* **22**(06) (2012) p. 1250143.
7. S. Bhowmick, S. C. Saha, M. Qiao and F. Xu, Transition to a chaotic flow in a V-shaped triangular cavity heated from below, *International Journal of Heat and Mass Transfer* **128** (2019) 76–86.
8. B. A. Márquez, J. J. Suárez-Vargas and J. A. Ramírez, Polynomial law for controlling the generation of n-scroll chaotic attractors in an optoelectronic delayed oscillator, *Chaos: An Interdisciplinary Journal of Nonlinear Science* **24**(3) (2014) p. 033123.
9. S. Kilinç, M. E. YalÇin and S. Özoguz, Multiscroll chaotic attractors from a hysteresis based time-delay differential equation, *International Journal of Bifurcation and Chaos* **20**(10) (2010) 3275–3281.
10. I. Cicek, A. E. Pusane and G. Dundar, A novel design method for discrete time chaos based true random number generators, *Integration* **47**(1) (2014) 38 – 47.
11. C. Li, B. Feng, S. Li, J. Kurths and G. Chen, Dynamic analysis of digital chaotic maps via state-mapping networks, *IEEE Trans. Circuits and Systems I: Regular Papers* **66**(6) (2019) 2322–2335.
12. S. Ozoguz, A. S. Elwakil and S. Ergun, Cross-coupled chaotic oscillators and application to random bit generation, *Circuits, Devices and Systems, IEE Proceedings* **153**(5) (2006) 506–510.
13. M. F. Tolba, W. S. Sayed, M. E. Fouda, H. Saleh, M. Al-Qutayri, B. Mohammad and A. G. Radwan, Digital emulation of a versatile memristor with speech encryption application, *IEEE Access* **7** (2019) 174280–174297.
14. C. Volos, A. Akgul, V.-T. Pham, I. Stouboulos and I. Kyprianidis, A simple chaotic circuit with a hyperbolic sine function and its use in a sound encryption scheme, *Nonlinear Dynamics* **89**(2) (2017) 1047–1061.
15. V. H. Carbajal-Gomez, E. Tlelo-Cuautle, J. M. Muñoz-Pacheco, L. G. de la Fraga, C. Sanchez-Lopez and F. V. Fernandez-Fernandez, Optimization and CMOS design of chaotic oscillators robust to PVT variations, *Integration* **65** (2019) 32–42.
16. A. Pano-Azucena, E. Tlelo-Cuautle, G. Rodriguez-Gomez and L. De la Fraga, FPGA-based implementation of chaotic oscillators by applying the numerical method based on trigonometric polynomials, *AIP Advances* **8**(7) (2018) p. 075217.
17. E. Gungor, E. Çavuş and I. Pehlivan, A logistic map Runge kutta-4 solution for FPGA using fixed point representation, *Chaos Theory and Applications* **1**(1) (2019) 19–28.
18. J. Jin and L. Zhao, Low voltage low power fully integrated chaos generator, *Journal of Circuits, Systems and Computers* **27**(10) (2018) p. 1850155.
19. T. Bonny and A. S. Elwakil, FPGA realizations of high-speed switching-type chaotic oscillators using compact VHDL codes, *Nonlinear Dynamics* **93**(Jul 2018) 819–833.
20. S. Soliman, M. A. Jaela, A. M. Abotaleb, Y. Hassan, M. A. Abdelghany, A. T. Abdel-

Hamid, K. N. Salama and H. Mostafa, FPGA implementation of dynamically reconfigurable IoT security module using algorithm hopping, *Integration* **68** (2019) 108 – 121.

21. M. Bakiri, J.-F. Couchot and C. Guyeux, CIPRNG: A VLSI family of chaotic iterations post-processings for F2-linear pseudorandom number generation based on Zynq MPSoC, *IEEE Trans. Circuits and Systems I: Regular Papers* **65**(5) (2018) 1628–1640.
22. P. Tertel and L. Hedrich, Real-time emulation of block-based analog circuits on an FPGA, *Integration* **63** (2018) 373 – 382.
23. 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 Transactions on Circuits and Systems II: Express Briefs* **66**(August 2018) 1381–1385.
24. E. Rodríguez-Orozco, E. García-Guerrero, E. Inzunza-Gonzalez, O. López-Bonilla, A. Flores-Vergara, J. Cárdenas-Valdez and E. Tlelo-Cuautle, FPGA-based chaotic cryptosystem by using voice recognition as access key, *Electronics* **7**(12) (2018) p. 414.
25. N. S. Soliman, M. F. Tolba, L. A. Said, A. H. Madian and A. G. Radwan, FPGA implementation of X-and heart-shapes controllable multi-scroll attractors, in *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, IEEE2018, pp. 1–5.
26. D. Zhong, G. Yang, Z. Xiao, Y. Ding, J. Xi, N. Zeng and H. Yang, Optical chaotic data-selection logic operation with the fast response for picosecond magnitude, *Optics express* **27**(16) (2019) 23357–23367.
27. V. Nguyen, H. Huynh, S. Kim and H. Song, Active EMI reduction using chaotic modulation in a buck converter with relaxed output LC filter, *Electronics* **7**(10) (2018) p. 254.
28. M. Tolba, H. Saleh, B. Mohammad, M. Al-Qutayri, A. S. Elwakil and A. Radwan, Enhanced fpga realization of the fractional-order derivative and application to a variable-order chaotic system, *Nonlinear Dynamics* **10.1007/s11071-019-05449-w** (2020).
29. H. Ghasemzadeh, M. Tajik Khasss and H. Mehrara, Cipher text only attack on speech time scrambling systems using correction of audio spectrogram, *The ISC International Journal of Information Security* **9**(2) (2017) 33–47.
30. P. Sathiyamurthi and S. Ramakrishnan, Speech encryption using chaotic shift keying for secured speech communication, *EURASIP Journal on Audio, Speech, and Music Processing* **2017**(1) (2017) p. 20.
31. W. Sayed, A. Radwan, M. Elnawawy, H. Orabi, A. Sagahyroon, F. Aloul, A. Elwakil, H. Fahmy and A. El-Sedeek, Two-dimensional rotation of chaotic attractors: Demonstrative examples and FPGA realization, *Circuits, Systems, and Signal Processing* **38**(Oct. 2019) 4890–4903.
32. 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 and Applications* **78**(12) (2019) 16097–16127.
33. A. G. Radwan, S. H. AbdElHaleem and S. K. Abd-El-Hafiz, Symmetric encryption algorithms using chaotic and non-chaotic generators: A review, *Journal of advanced research* **7**(2) (2016) 193–208.
34. S. Lian, *Multimedia content encryption: techniques and applications* (CRC press, 2008).
35. A. Pande and J. Zambreno, A chaotic encryption scheme for real-time embedded systems: design and implementation, *Telecommunication Systems* **52**(2) (2013) 551–561.
36. 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 *2018 IEEE International Conference on Industrial Technology (ICIT)*, IEEE2018, pp. 1412–1417.

37. Z. Cao and Y. Zhang, Variable frequency modulation for EMI suppressing in power converter, *Energy and Power Engineering* **5**(04) (2013) p. 1147.