Random Number Generation Based on Digital Differential Chaos

M. Affan Zidan[1], A. G. Radwan[1,2] and K. N. Salama[1]
[1]Electrical Engineering Program
King Abdullah University of Science and Technology (KAUST)
Thuwal, Kingdom of Saudi Arabia 23955-6900
Email: {mohammed.zidan, ahmed.radwan, khaled.salama}@kaust.edu.sa
[2]Department of Engineering Mathematics,
Faculty of Engineering, Cairo University, Egypt

***Abstract*—In this paper, we present a fully digital differential chaos based random number generator. The output of the digital circuit is proved to be chaotic by calculating the output time series maximum Lyapunov exponent. We introduce a new post processing technique to improve the distribution and statistical properties of the generated data. The post-processed output passes the NIST Sp. 800-22 statistical tests. The system is written in Verilog VHDL and realized on Xilinx Virtex® FPGA. The generator can fit into a very small area and have a maximum throughput of 2.1 Gb/s.**

I. Introduction

Chaos is a nonlinear deterministic system that expresses random behavior. Many analog chaos generators have been introduced ranges from using discrete elements and Op-amps [1], to completely MOS based [2]–[4], to fully integrated [5], and digitally based chaotic generators [6]. Implementation for multi-scroll chaos was also introduced [7]. Chaos generators find applications in chaotic based digital communication systems [8], cryptography [9], and random number generation [10].

Analog chaotic generators are sensitive to the operating conditions, process variations, and temperature. In addition, the initial conditions cannot be set precisely in analog generators. Analog circuit implementation typically requires a large on-chip area for the state capacitor realization. According to [11] tolerances in practical chaos implementations decrease the quality of the output (Entropy). Many theoretically useful properties of RNG such as, efficiency, repeatability, portability [12], cannot be achieved using analog circuit realizations.

In this paper we propose an area efficient high speed digital differential chaos based random number generator generator, where the initial conditions and the states are stored in registers rather than capacitors. The system is characterized by wide noise margins and high reliability. The introduced generator is based on the chaos system given in [1], which is described as,

$$-\dddot{X} = \ddot{X} + B\dot{X} + X \tag{1}$$

where the nonlinear element is defined as,

$$B\left(\dot{X}\right) = \begin{cases} \alpha, & \dot{X} \geq 1 \\ 0, & \dot{X} < 1 \end{cases} \tag{2}$$

The system is written in Verilog VHDL and realized on Xilinx Virtex® 4 FPGA. While the used FPGA is not the latest generation, our circuit fits into a very small area and the output can reach a throughput of 2.1 Gb/s. The digital output is proved to be chaotic by calculating the Maximum Lyapunov Exponent (MLE), and showing a better chaotic behavior than the original differential equation. A new post processing technique is also introduced to improve the distribution and the statistical properties of the generated data, and the output was analyzed using the NIST Sp. 800-22 statistical tests [13].

II. Digital Chaotic Differential Equation

Differential based chaotic generator can be digitally implemented by realizing the numerical solution of its differential equations. The system given in equation (1) was solved using three different numerical methods, Runge-Kutta fourth order, mid-point, and Euler techniques. The generated time series is proved to be chaotic by calculating its MLE for all three systems. Although Runge-Kutta fourth order and mid-point solutions are considered to be more accurate than Euler, the last technique adds an extra nonlinearity to the chaotic system, which appears from the calculated Lyapunov exponent, which saturates at 0.203, 0.276, and 0.377 for Runge-Kutta fourth order, mid-point, and Euler techniques respectively. Euler technique is used to implement the digital system. Let $Y = \dot{X}$ and $Z = \ddot{X}$, the numerical solution for equation (1) is evaluated as,

$$X_{t+h} = X_t + hY_t \tag{3a}$$

$$Y_{t+h} = Y_t + hZ_t \tag{3b}$$

$$Z_{t+h} = Z_t - h\left(Z_t + Y_t\, B\left(Y_t\right) + X_t\right) \tag{3c}$$

where t expresses the time and h is the time step. These equations can be realized using a simple register transfer module, where the state variables X, Y and Z are implemented as registers, rather than capacitors in the analog realizations. Each of the three equations is realized as a combinational logic unit. Fig. 1 shows the schematic of the digital implementation of the chaos generator. Such digital implementation can be thought of as a nonlinear extension of the linear feed-back shift registers.

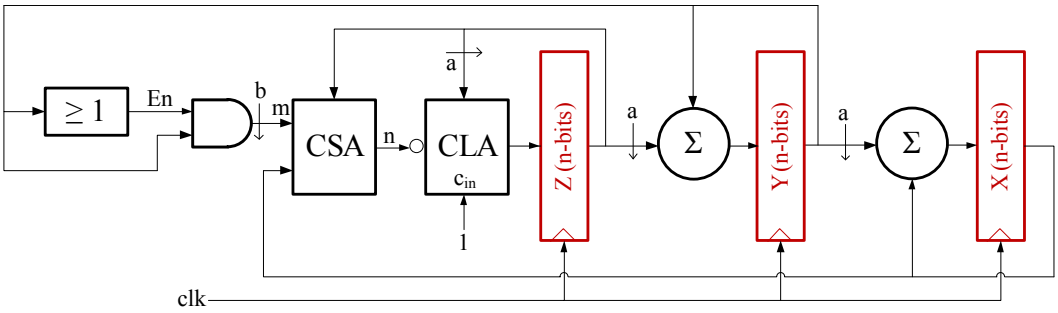


Fig. 1. The schematic of the digital implementation of the chaos generator. The arrows are used to donate to shift in wires. CSA and CLA are Carry Save Adder and Carry Lookahead Adder respectively. $m = Y\,B\,(Y)$ and $n = Z + Y\,B\,(Y) + X$.

In the native form of the equations (3a), (3b) and (3c), there are four required multiplication operations. Eliminating such multiplications will reduce the system's area significantly. Since the system will be chaotic for intervals of h and α, so they are selected such that, $h = 2^{-a}$ and $\alpha = 2^{b}$, where a and b are positive integers. This will transform the area-consuming multiplication operations into simple shifts. Since equation (3c) is the bottleneck of the digital generator pipeline, fast Carry-Look-Ahead adder (CLA) and carry save adder (CSA) were used in its implementation. However, normal carry-propagate adders were used within equations (3a) and (3b) for area saving.

The outputs of the chaotic generator are within intervals of bounded maximums and minimums. Therefore, fixed-point numbers representation is an excellent selection for the system realization. Such selection will reduce the circuit area and delay significantly. A 16-bits fixed-point representation is used to describe the numbers in the system which adds an extra nonlinearity and improves the chaotic behavior. The most significant bit is used for the sign, the following three bits for the integer part, and the rest of the bits for the fraction part. The nonlinear element is simply realized by an enable line and an array of AND gates as shown in Fig. 1. The enable line will be active in case of $Y \geq 1$, neutralizing the AND gates and passes a shifted version of Y. In the other case the enable will be reset, which passes zeros through the gates array. The enable Boolean equation is given by,

$$Enable = y\,(15)' \cdot (y\,(14) + y\,(13) + y\,(12)) \qquad (4)$$

III. Random Number Generation

Chaos circuit realizations are considered to be one of the main techniques to create random number generators (RNG) [10], [14], beside jittered oscillator sampling [15], amplification of a noise source [16], and Quantum based RNGs [17]. According to [10] chaos generators suffers from a short term predictability. In general, post-processing techniques are used to improve the statistical properties of generated random sequences, but in return these techniques reduce the throughput of the generator [14]. We introduce a new post-processing technique to improve the chaos generator output to meet the RNG statistical requirements. The proposed post-processing performance is compared to previously known systems, Von Neumann technique, and bit-counting [14]. Also, the output of the introduced RNG is tested using the standard NIST Sp. 800-22 random tests package [13].

Based on chaos generator properties and statistical observations, we introduce a new post-processing technique. This technique steps is described as,

Step 1: According to [10] chaos generators suffers from a short term predictability. We found that discarding the highest significant bits and keeping only the lowest ones removes the short term predictability.

Step 2: Based on statistical observations, it was found that the distribution of the odd and even numbers, and so the ones and zeros, are not balanced. The output distribution was balanced by simply discarding the lowest significant bit.

Step 3: The experimental results show that attaching the three outputs of the generator in one sequence improves both of the throughput and the statistical properties of the output.

In general our post-processing is based on discarding the group of bits with statistical defects, specially the highest significant bits for chaotic systems. The exact number of bits to keep is selected according to the experimental results. This selection is made such that bits from 1 to 3 from each variable are attached together forming the final output, with symbols range from 0 to 511.

IV. Experimental Results

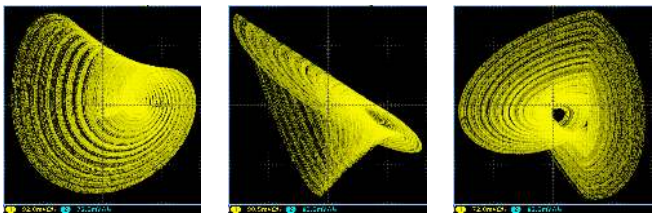
The chaos generator and the RNG were written in Verilog VHDL and realized on Xilinx Virtex® 4 XC4VSX35 FPGA. Table I shows the details FPGA area utilization, maximum frequency and maximum throughput of both of the chaos generator and the RNG. This table shows that purposed generator fits into an extremely small area, and reaches a maximum throughput of 2.1 Gb/sec.

TABLE I
AREA UTILIZATION, MAXIMUM FREQUENCY AND THROUGHPUT OF THE SYSTEM REALIZATION ON THE XILINX VIRTEX® 4 XC4VSX35 FPGA. THE CIRCUITS WERE SYNTHESIZED USING XILINX ISE® 11, WITH OPTIMIZED FOR TIME OPTION SELECTED.

	Slices (15,360)	**Slices FF (30,720)**	**LUTs (30,720)**	**Freq. (MHz)**	**Throughput (Mb/s)**
Chaos	154	65	243	128.3	2052
RNG	136	57	233	131.1	1180

A. Chaotic Behavior

Fig. 2 shows the attractor of the digitally generator output as captured from the oscilloscope, which show a perfect chaotic behavior. While the visual inspection shows a chaotic phasor trajectory diagram, it is not a sufficient proof for chaos. The generated time series is proved to be chaotic by calculating its MLE for the digitally created time series. The calculation is made using the software provided in [18]. The numerically calculated exponent saturates at a value of 0.391 after 250,000 iterations. The positive value of the exponent proves the chaotic behavior of the generated data. The calculated exponent shows an improvement over the numerical solution, due to the nonlinearity added to the system by the fixed-point digital implementation, as shown in Fig. 3.



(a) X-Y (b) X-Z (c) Y-Z

Fig. 2. Snapshots of the attractor projections, captured using TekTronix™ MSO 414 mixed signal oscilloscope.

B. Post-processing

The proposed post-processing improves the symbols distribution compared to the original chaos output, as shown in Fig. 4. This figure shows the symbols histogram changes from a Gaussian-like before applying the post processing to almost a uniform distribution after the proposed technique applied. The figure also shows that the post-processing technique balances the zeros-ones ratio. The calculations are made on an 11 million symbols dataset.

The proposed post-processing improves the frequency response of the circuit output. Fig. 5 shows the spectrum digital output of the circuit before and after post processing. The figure shows that the post processing spreads the power over the spectrum to be more likely as white noise. The spectrum of

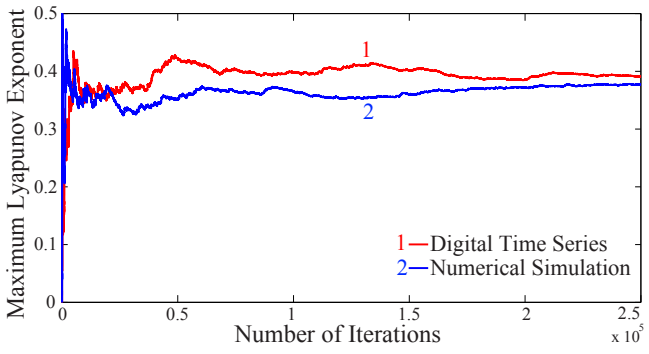


Fig. 3. The MLE plotted versus number of iterations for the digitally generated time series and the Euler numerical simulation.

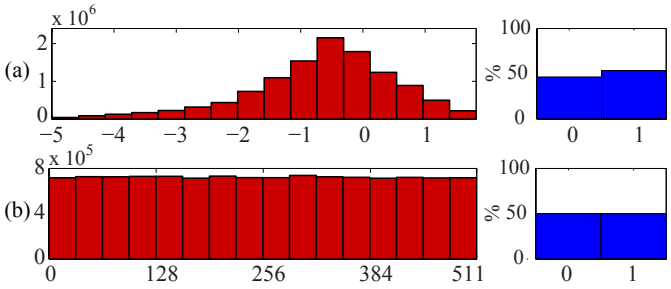
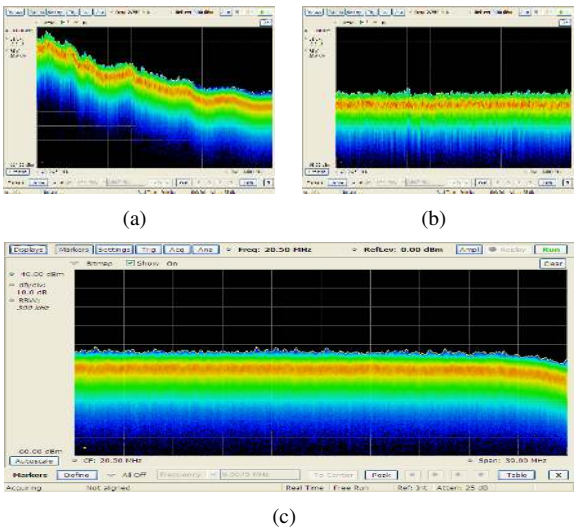


Fig. 4. Histogram of the symbols (on the left) and the distribution of zeros and ones (on the right) are plotted for, (a) the original generated data without post-processing for x variable, and (b) after applying the post-processing step 3.

the $f_s/2$ range shows slit deformation at the higher frequency due to the zero order hold effect of the digital to analog converter used for adapting the output for the spectrum analyzer. Fig. 6(a) and Fig. 6(b) shows a sample of the circuit digital output versus time before and after applying post-processing. It clearly appears from these figures that the applied post-processing decreases the predictability of the output. Also the post-processing increases the power of the higher frequency components. Fig. 6(c) shows X-Z attractor projections after applying step 2 of the post-processing, which is uniformly distributed over square shape attractor.

Statistical Test: We use the NIST Sp. 800-22 random tests package [13] to demonstrate the statistical improvement introduced by applying the purposed post-processing. The system output passes all the 15 Sp. 800-22 tests with a success rate of 100%. Table shows the results of the NIST Sp. 800-22 statistical test result. The results are given for the original output and different post processing techniques. Bit-counting post-processing, Von Numman, and our purposed post-processing



(a) (b) (c)

Fig. 5. Snapshot for the spectrum, using TekTronix™ RSA 6120 real-time spectrum analyzer, (a) chaos generator output up to 4.5MHz, (b) post-processed RNG output up to 4.5MHz, and (c) post-processed RNG output up to 40MHz ($f_s/2$).

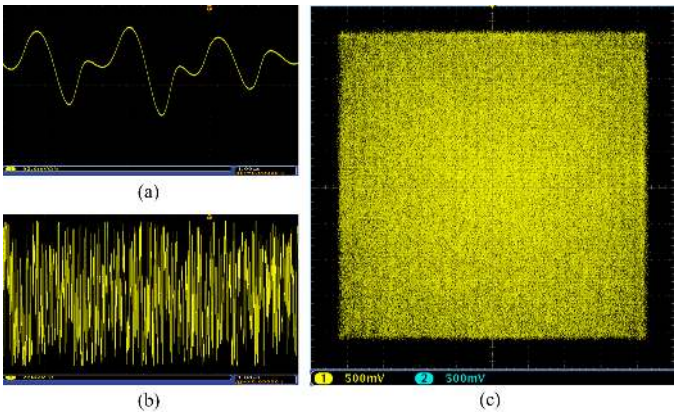


Fig. 6. Sample of the circuit output, captured using TekTronix™ MSO 414 mixed signal oscilloscope, for, (a) the original generated data of variable x for time interval of 1μSec, (b) the original generated data of variable x after applying the post-processing for the same interval, and (c) X-Z attractor projections after applying step 2 of the post-processing.

are all compared. The table shows the proportion Value (PP) as a fraction of one, and the validity of the P-Values distribution, the P-Value of the P-values (PV). According to [13] the PV is vialed for values greater than 0.0001. For the final output the minimum PV was 0.066 and the maximum was 0.91. The NIST results verify that bit-counting and Von Numman show improvement over the original data but are not sufficient to pass all the tests for our system. The proposed post-processing successfully passes all of the tests.

TABLE II
THE PROPORTION VALUE (PP), AND THE VALIDITY OF THE P-VALUES DISTRIBUTION (PV) OF THE NIST SP. 800-22 TEST SHOWING RESULTS FOR (A) NO POST-PROCESSING IS USED, (B) BIT-COUNTING TECHNIQUE (EACH FOUR BIT IS COUNTED TOGETHER), (C) VON NEUMANN POST-PROCESSING, AND (D) PROPOSED POST-PROCESSING. THE THROUGHPUT AS BITS PER CYCLE IS GIVEN FOR EACH CASE.

	(a)		(b)		(c)		(d)	
	PV	PP	PV	PP	PV	PP	PV	PP
Frequency	×	0.5	✓	0.5	✓	0.9	✓	1
B. Frequency	×	0.3	×	0.3	✓	0.9	✓	1
C. Sums	×	0.4	✓	0.4	✓	0.9	✓	1
Runs	×	0.3	✓	0.8	✓	0.9	✓	1
Longest Run	×	0.2	✓	0.8	✓	1	✓	1
Rank	×	0	×	0	✓	1	✓	1
FFT	✓	1	✓	1	✓	0.9	✓	1
N. O. Temp.	✓	1	✓	1	✓	1	✓	1
O. Temp.	×	0	×	0	×	0	✓	1
Universal	×	0	×	0	×	0	✓	1
App. Entropy	×	0	×	0	×	0	✓	1
R. Excur.	×	0	×	0	×	0	✓	1
R. Excur. V.	×	0	×	0	×	0	✓	1
Serial	×	0	×	0	×	0.1	✓	1
L. Complexity	✓	0.9	✓	1	✓	1	✓	1
TP [bits/cycle]	16		4		3.36		9	

V. CONCLUSION

In this paper a fully digital, area efficient high speed chaos based random number generator has been introduced. The output of the digital circuit is proved to be chaotic by calculating the output time series MLE, which shows a better chaotic response than the original differential equations. The technique described in this paper can be applied to other more complicated analog generators. A new post processing technique is introduced to improve the distribution and statistical properties of the generated data. The post-processed output passes successfully the NIST Sp. 800-22 statistical tests. The generator was realized using Verilog HDL, and implemented on Xilinx Virtex® 4 FPGA.

REFERENCES

[1] A. Elwakil and M. Kennedy, "Construction of classes of circuit-independent chaotic oscillators using passive-only nonlinear devices," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 48, no. 3, pp. 289–307, 2001.

[2] A. G. Radwan, A. M. Solman, and A. El-Sedeek, "MOS realization of the modified Lorenz chaotic system," *Chaos, Solitons and Fractals*, vol. 21, pp. 553–561, 2004.

[3] A. Demirkol, S. Ozoguz, V. Tavas, and S. Kilinc, "A cmos realization of double-scroll chaotic circuit and its application to random number generation," in *IEEE International Symposium on Circuits and Systems (ISCAS'08)*, May 2008, pp. 2374–2377.

[4] A. G. Radwan, A. M. Soliman, and A.-L. El-Sedeek, "An inductorless CMOS realization of Chua's circuit," *Chaos, Solitons and Fractals*, vol. 18, pp. 149–158, 2003.

[5] A. S. Elwakil, K. N. Salama, and M. P. Kennedy, "An equation for generating Chaos and its monolithic implementation," *International Journal of Bifurcation and Chaos*, vol. 12, no. 12, pp. 2885–2895, 2002.

[6] A. S. Mansingka, A. G. Radwan, M. A. Zidan, and K. N. Salama, "Analysis of bus width and delay on a fully digital signum nonlinearity chaotic oscillator," in *IEEE International Midwest Symposium on Circuits and Systems (MWSCA'11)*, August 2011.

[7] K. N. Salama, S. Ozguz, and A. S. Elwakil, "Generation of n-scroll chaos using nonlinear transconductors," in *International symposium on circuits and systems (ISCAS'03)*, May 2003, pp. 176–179.

[8] P. Chiang and C. Hu, "Chaotic pulse-position baseband modulation for an ultra-wideband transceiver in cmos," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 57, no. 8, pp. 642–646, 2010.

[9] F. Pareschi, G. Scotti, L. Giancane, R. Rovatti, G. Setti, and A. Trifiletti, "Power analysis of a chaos-based random number generator for cryptographic security," in *IEEE International Symposium on Circuits and Systems. ISCAS*, May 2009, pp. 2858–2861.

[10] M. Yalcin, J. Suykens, and J. Vandewalle, "True random bit generation from a double-scroll attractor," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 51, no. 7, pp. 1395–1404, 2004.

[11] T. Addabbo, M. Alioto, A. Fort, S. Rocchi, and V. Vignoli, "A feedback strategy to improve the entropy of a chaos-based random bit generator," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 53, no. 2, pp. 326–337, 2006.

[12] P. L'Ecuyer, "Uniform random number generators: A review," in *Proceedings of the 1997 Winter Simulation Conference*, December 1997, pp. 127–134.

[13] A. R. et. al., "A statistical test suite for random and pseudorandom number generators for cryptographic applications," 2001.

[14] T. Stojanovski, J. Pihl, and L. Kocarev, "Chaos-based random number generators. part ii: practical realization," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 48, no. 3, pp. 382–385, 2001.

[15] M. Bucci and R. Luzzi, "Fully digital random bit generators for cryptographic applications," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 55, no. 3, pp. 861–875, 2008.

[16] W. Holman, J. Connelly, and A. Dowlatabadi, "An integrated analog/digital random noise source," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 44, no. 6, pp. 521–528, 1997.

[17] S. Pironio, A. Acin, S. Massar, A. B. de la Giroday, D. N. Matsukevich, P. Maunz, S. Olmschenk, D. Hayes, L. Luo, T. A. Manning, and C. Monroe, "Random numbers certified by Bell's theorem," *Nature*, vol. 464, pp. 1021–1024, 2010.

[18] M. Perc, "User friendly programs for nonlinear time series analysis," 2010, [Retrieved 7 August, 2010]. [Online]. Available: http://www.matjazperc.com/ejp/time.html