Implementation of a Chaotic Time-Delay RNG Based Secure Communication System on FPGA

Latif Akçay[1], Erdem Çil[1], Alptekin Vardar[1], İlayda Yaman[1], Ramazan Yeniçeri[2], Müştak E. Yalçın[1]

[1] Department of Electronics and Communication Engineering
[2] Aerospace Research Center
Istanbul Technical University
Istanbul, TR-34469, Turkey
E mail: {akcayl, cile, vardara, yamanil, yenicerir, mustak.yalcin} @itu.edu.tr

Abstract

Security is one of the most important design parameters in communication systems. Security of cryptographic systems depends on the unpredictability of keys. Chaotic random number generators have become an alternative method for random number generation instead of physical noise based ones. In this work, we describe a system-on-chip design which includes a chaos-based random number generator. Key generation, encryption-decryption blocks and control unit are designed to run on the same chip. All blocks are connected to the Microblaze softcore processor and implemented on a Xilinx FPGA. Structural details of the system and the results are shared.

1. Introduction

With the rapid increase of interconnectedness of autonomous systems in the recent years, safe communications became a fundamental issue. Thus the need for a secure cryptologic hardware in Internet of Things (IoT) devices must be fully addressed. The increase in the necessity to cryptology resulted in with an interest in statistically compatible random number generators. A good random number generator (RNG) is vital for cryptographic security and consequently, any security hardware must have an RNG block[1].

Random number generators are widely used in cryptology as key generators. Thus the security of the system greatly depends on the randomness of the source. To obtain unpredictable systems, true random number generators (TRNG) must be used as a source. Classical TRNGs use different kinds of physical phenomenons as sources. Nowadays, most commonly used phenomenon in embedded TRNGs is the jitter noise of digital clock signals[2]. Chaotic circuits offer a promising alternative way. Chaotic systems are described as aperiodic systems which generate unpredictable noise like signals[3]. They have a sensitivity dependence on the initial condition. Thus, even a small amount of difference in the initial signal eventually results with a big difference in the state of the system. Because of such uncertainties, if the state variable is unknown, the output of the system cannot be predicted, meaning infinite entropy[4].Nevertheless, short term predictability is the biggest drawback of chaotic systems. One should note that because of this future, in order to use a chaotic system in random number generation, it must be quantized.

In today's world, communication applications that require security should be flexible and use little chip area. At the same time, the speed factor is one of the parameters to be considered. For all these reasons, a flexible microprocessor was chosen in this study and cryptographic blocks covering a small chip area were used.

In the first part of the work, summary information related to RNG structure was presented. In the second part, encryption and decryption algorithms are explained. In addition, how all the peripheral units are connected to the Processor Local Bus (PLB) interface and the general flow of the communication system is shown. In the third part, the results are shared and commented.

2. Chaotic Time-Delay Sampled Data System Based TRNG

Because of their relatively simple system model, chaotic time-delay systems present great potential to generate chaotic dynamic behaviors. Time-delay part of the system is basically a memory unit with a future of storing its input and transferring it to its output after a certain amount of time.

This time-delay circuit is generally implemented with LCL or Bessel-type filters. With those implementations, it is hard to reduce the number of physical components and the number of idealized elements to model the circuit. Circuit realization of these filters is the main drawback at the implementation of chaotic time-delay systems. In order to overcome this negative effect, a feedback system which offers a binary feedback function is presented[5]. Usage of this binary feedback system results in with a highly simplified implementation of the chaotic system. This binary feedback system is composed of D-type flip-flops. The output state of flip-flop will be changed to its input state on the rising edge of the clock signal. Thus, D flip-flop samples its input states at every clock edge and delays them. Usage of a flip-flop chain with the purpose of delaying the binary output of the nonlinear feedback part of the introduced system results in with a new system that is a sampled-data feedback system[6].

In this system, the chaotic sampled-data system which is proposed in [6] is used as the key generator. The mathematical model of the introduced system is presented in the following equation;

$$\dot{x}(t) = -x(t) + \alpha f(x(t_k - \tau)),\ t_k \leq t < t_k + T_s \qquad (1)$$

where x represents the state, α represents the feedback weight, τ is the delay amount applied $t_k - th$ sample of the x and Ts represents the sampling period[8-EB]. The nonlinear binary

feedback function $f(x)$ is depicted by the following equation;

$$f(x) = 4\,(u(x-1) + u(-x-1)) - 2 \qquad (2)$$

where $u(\cdot)$ is the unit step function. Between tk and $tk + Ts$, the behavior of the system constantly changes because of the sampled and delayed feedback x(t_k- τ). For the values α = 2, τ = 8, Ts = 0.1, system has a strange attractor in$x(t) - x(t-\tau)$ plane shown in Figure 1.

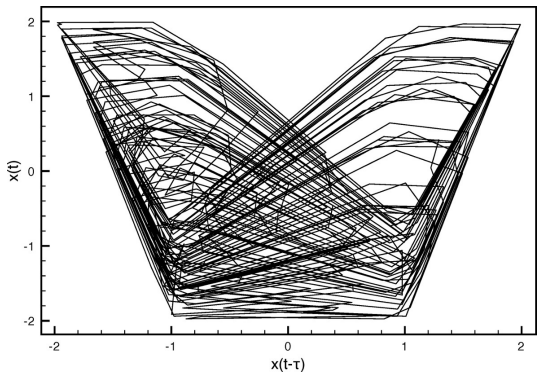

Figure 1. Chaotic dynamic of the system in $x(t) - x(t-\tau)$. The state $x(t) \in [-2, 2]$ [7].

2.1. Digitalization

For integration, which is vital for a random number generator, the implementation and full digital design of the system must be done. This system has been discretized by forward Euler method with integration step h, and given by the following equation;

$$x[k+1] = x[k] + h(-x[k] + f(x[k - \frac{\tau}{h}])). \qquad (3)$$

In the original system $x(t) \in [-2, 2]$ but here, the amplitude of the state variable $x(t)$ has been quantized between -4 and 4 and the amplitude of the system has been quantized by the 8-bit digital state with twos complement representation(signed Q3.5 format). The block which realizes the system is named INTCOMP (integrate and compare). Block diagram of this block is given in Figure 2.

REG is the 8-bit state register. It has the initial value INIT and can be reset to that value by RST signal. FT represents the delayed signal of nonlinear feedback. Depending on the value of FT signal, state registers value can be increased either $h(-x[k]+2)$ or $h(-x[k]-2)$. Period of the CLK signal is equal to h. In this implementation h is equal to 1/4(normalized value) so the multiplication process can be done by shifting operation. The purpose of the comparator block is to generate F, which will then be driven in to the delay line. An alternative delay line composed of Look-up-tables (LUT) can provide both the delayed signal and a delay uncertainty that breaks the periodic trajectory. This alternative delay line which functions as a non-inverting buffer is named DLUT and shown in Figure 3.

The source of the propagation delay is the low-pass characteristics of digital gates. At the output of each gate, voltage is increased or decreased exponentially. Thus it takes some amount of time to cross the threshold voltage level of the next gate. The combination of this effects at each gate generates the delay line

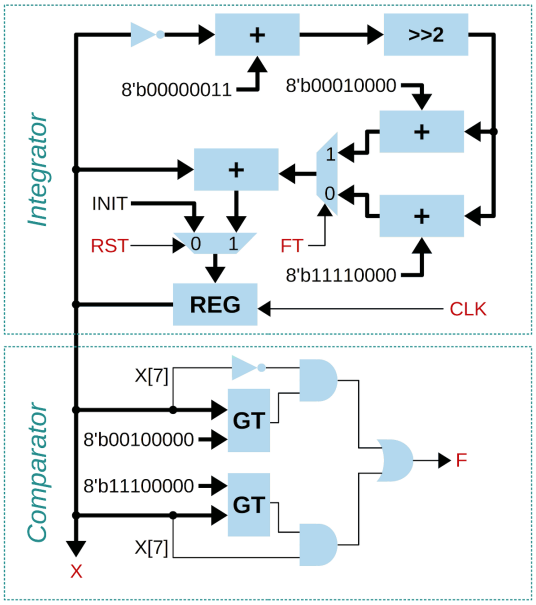

Figure 2. Schematic diagram of integrate and compare block (INTCOMP) [7].

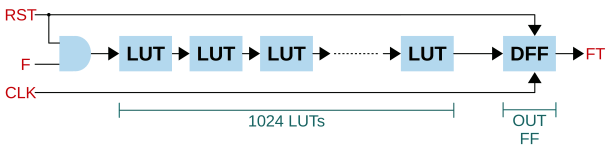

Figure 3. Schematic diagram of Look-up-table based delay line (DLUT) [7].

on LUT chain. In addition, process and environmental (PVT) variations effects on delay line and PVT variations induce randomly varying delay ($\tau(tk)$). The system which is designed based on DLUT is described by the following equation;

$$x[k+1] = x[k] + h(-x[k] + f(x[k - \frac{\tau(k)}{h}])) \qquad (4)$$

where $\tau(k)$ is a random variable.

DLUT line has a significant jitter on it. This jitter is caused by the noise and it causes the random process of this line. Chaotic basis of INTCOMP has a sensitivity to initial conditions. Hence, INTCOMP dramatically changes its trajectory when the delayed signal has a slight change with respect to the expected one. The resultant system is shown in the Figure 4.

More detailed information about the TRNG may be found our previous papers [5],[6],[7],[8].

3. Implementation of Communication Sytstem

The protection of the information transmitted is one of the most important issues in communication technologies. It is one of the main concerns of cryptology to develop different systems that provide protection of the data against outside threats, which means securing the data.

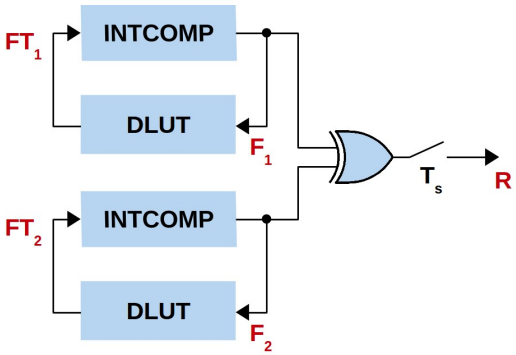

Figure 4. Block diagram of paired RNG.

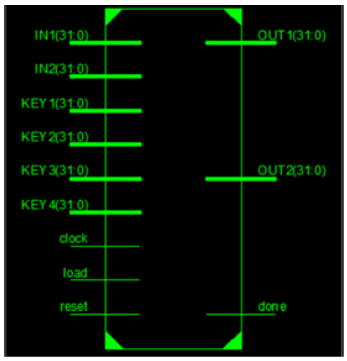

Figure 5. Block diagram of Cryptographic Modules.

Microblaze is a small and flexible softcore processor by Xilinx. In this work, it is used as a control unit of the entire desing. The system consists of the TRNG, encryption block, decryption block and UART. In addition to the TRNG, encryption and decryption blocks are also implemented as hardware in Verilog HDL. All peripherials are connected to the processor with PLB interface by using Xilinx Platform Studio (XPS). After the hardware design, the software design has been done and the test codes written have been compiled for Microblaze in the Xilinx Software Development Kit (SDK) environment.

The Federal Information Processing Standards (FIPS) provide some statistical tests to be used to evaluate the randomness of RNGs in cryptographic-based security systems. Four different statistical tests are determined for the randomness of a random number generator. In each of these tests, a static value is computed from the generated numbers and some boundaries are given that must be satisfied by this static value. These boundaries change in different FIPS publications. These four tests are performed on a 20000 bits stream generated by the random number generator. The generator cannot satisfy the standards if any of these tests fail[9]. We have tested our TRNG design by coding these tests in C language and using the SDK. For real tests, we have used Digilent Atlys Board which have Xilinx Spartan6 SLX45-CGS324 FPGA on it. We have also obtained NIST(National Institute of Standards and Technology) SP800-22REV1A statistical tests results. More detailed information about the tests and results can be found in our previous works[5],[6],[7],[8].

3.1. Tiny Encryption Algorithm (TEA)

The encryption module included in this project is coded using Tiny Encryption Algorithm. In Tiny Encryption Algorithm, a 64-bits information, which is divided into two 32-bits words, is encrypted using a 128-bits key, which consists of 4, 32-bits keywords[10].

The decryption module works similar to the encryption module. The first input of the decryption module is the second output of the encryption module. The second input of the decryption module is the first output of the encryption module. The keywords must be selected the same as in the encryption module. The I/O diagram of these modules can be seen in Figure 5. These two modules are simulated using Xilinx ISE Simulator. The inputs of the encryption module are obtained in the output of the decryption module.

```
always@(posedge clk)begin////clk/(2*RECCOUNT)
    if( RECCOUNT < sample_rate )begin   RECCOUNT <= RECCOUNT + 8'd1; end
    else begin   RECCOUNT <= 8'd0;   CLKREC = ~CLKREC; end
end
```

Figure 6. Changing the throughput value.

3.2. Connecting the TRNG to PLB Interface

The TRNG module is added to the Microblaze as a custom peripheral. Hence, a new peripheral with the name “trng” and its templates are created. Again, PLB is selected as the bus to which the peripheral will be attached. The number of software accessible registers is selected as 10. The HDL of the user logic template is selected as Verilog. The TRNG system is added to user logic template as a submodule. The inputs and outputs of the RNG module are connected to slave registers. The TRNG module includes a Block RAM and the netlist core of this Block RAM must be instantiated in the custom peripheral.

The throughput value of DFF-DLUT line TRNG was changed for different values. This value shows how many times the sample is taken from a sample pool. These changes were done in the Verilog code of the TRNG. The value RECCOUNT in Figure 6 was changed in the range of 0-255. When different throughput values are tested, it has been seen that unless RECCOUNT is 0, the tests are passed most of the time. When it is zero, the tests are barely passed. The passing ratio falls dramatically. RECCOUNT being zero means that each sample is included as a result. In conclusion, it can be stated that the TRNG passes all of the four statistical RNG tests (FIPS) except for some special situations. Therefore, it can be used in a cryptographic system.

In order to TRNG module works properly, DLUT block included by it must be placed together. Therefore, an area constraint is added to UCF of the system. This constraint can be seen in Figure 7. Another requirement of the RNG module is that it must be used with a clock frequency of 120 Mhz, but the system clock frequency of Spartan 6 FPGA is 100 Mhz. Therefore, the clock generator of Microblaze is used. CLKOUT0 output of the clock generator is configured to have a clock frequency of 120 Mhz and connected to the “trng” peripheral. Finally, “Treat timing closure as an error” option in Project/Project Options/Design Flow is unselected (it is selected by default).

```
INST "number_0/number_0/USER_LOGIC_I/AB/DLUT1/*" AREA_GROUP = "DLUT" ;
AREA_GROUP "DLUT" RANGE = SLICE_X2Y2:SLICE_X30Y30;
```

Figure 7. Placement Constraint

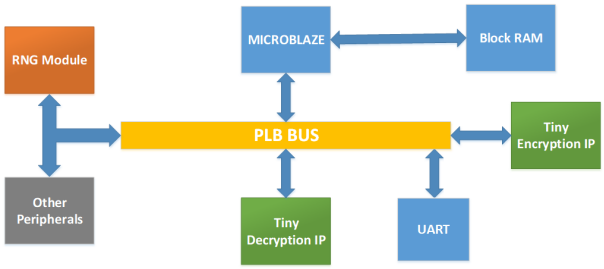

Figure 8. System-on-chip structure of the system.

3.3. Overall Structure of The System

Overall system has been created using the encryption module, the decryption module and the TRNG with the configuration that passes the tests. These three modules are added to the Microblaze as custom peripherals by using XPS tools. Hence, three separate modules are combined to have a system-on-chip design. The duty of the TRNG system is to provide key words to the process. In this complete system, keywords are taken from 20000 bits random numbers array. The block diagram of the overall system is shown in the Figure 8.

The system is tested with different input values. Each time, numbers given to two inputs of the system are obtained as two outputs of the decryption process. Two examples can be seen in the Figure . In the first example, inputs are 3 and 5. In the second example, inputs are 500 and 1000 (1f4 and 3e8 in hexadecimal format).

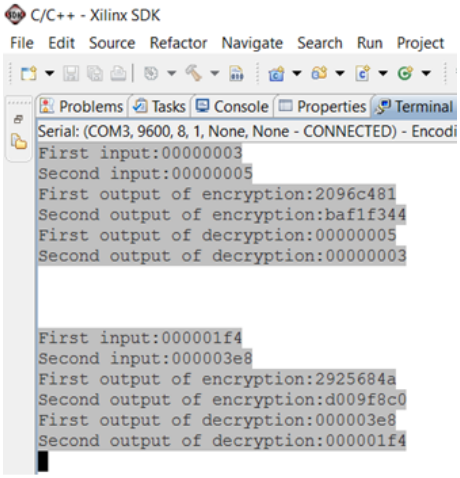

Figure 9. Result of the system work for two different values.

4. Conclusions

In this project, a system on chip implementation of an example secure communication system on FPGA was realized. The TRNG and cryptographic IPs are embedded with the soft processor Microblaze. Two different TRNG structures were used; DFF-DLUT line RNG and DLUT-DLUT line RNG. The SOC implementation of the RNG makes it possible to evaluate and test the generated numbers in a rather quick way.

The FIPS tests were implemented in C language and run on Digilent Atlys Board. According to the results of the tests, the system can be optimized for better results. By examining the results of the tests, it is determined that the random number generator satisfies all of these standards most of the time. After that, the random number generator system was implemented again with some parameter changes and the effect of these changes on the results was examined. Besides, the whole system was tried with crypto applications and accurate results were obtained and shared. This provides a secure communication method that can be used especially in applications where less chip areas are required.

5. References

[1] Menezes, Alfred J., Paul C. Van Oorschot, and Scott A. Vanstone.1996. Handbook of applied cryptography. CRC press.

[2] Galajda, M. D. P. (2006). True Random Number Generator Embedded in a Reconfigurable Hardware. Journal of Electrical Engineering, 57(4), 218-225.

[3] Pareschi, Fabio, Gianluca Setti, and Riccardo Rovatti. "Implementation and testing of high-speed CMOS true random number generators based on chaotic systems." IEEE transactions on circuits and systems I: regular papers 57.12 (2010): 3124-3137.

[4] Callegari, Sergio, Riccardo Rovatti, and Gianluca Setti. "Embeddable ADC-based true random number generator for cryptographic applications exploiting nonlinear signal processing and chaos." IEEE Transactions on Signal Processing 53.2 (2005): 793-805.

[5] R. Yeniceri and M. E. Yalcin, "True random bit generation with sampled-data feedback system", Electronics Letters, vol. 49, no. 8, pp. 543-545, April 2013

[6] Yalcin, Müştak E., Ramazan Yeniceri, and Serdar Özoğuz. "A chaotic time-delay sampled-data system and its implementation." International Journal of Bifurcation and Chaos 24.03 (2014): 1450039.

[7] Yeniceri, R. and Vardar, A. and Yalcin, M. E., "Full Digital Implementation of A Chaotic Time-delay Sampled-data System", Circuits 2017 IEEE International Symposium on and Systems (ISCAS), May 2017, accepted.

[8] Yeniceri, R., Vardar, A., Cil, E., Akcay, L., Goncu, E., and Yalcin, M. E., "A Chaotic Time-delay System Based Digital RNG and Integrated Autonomous Test Suite", The 23 European Conference on Circuit Theory and Design (ECCTD), September 2017, accepted.

[9] Vithanage, A., and Shimizu, T. (2003). Fips 140-2 (change notice 1) random number tests. Available: http://www. fdk. co. jp/cyber-e/pdf/HM-RAE103. pdf, 140-2.

[10] Wheeler, David J., and Roger M. Needham. "TEA, a tiny encryption algorithm." International Workshop on Fast Software Encryption. Springer Berlin Heidelberg, 1994.