Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, Guangzhou, 18-21 August 2005

THERMAL NOISE RANDOM NUMBER GENERATOR BASED ON SHA-2 (512)

YU-HUA WANG, HUAN-GUO ZHANG, ZHI-DONG SHEN, KANG-SHUN LI

School of Computer Science, Wuhan University, Wuhan 430079, China
E-MAIL: yuhua.w@tom.com

Abstract:

Couple to the rapid development of cryptography, the strength of security protocols and encryption algorithms consumingly relies on the quality of random number. This paper presents a new and security random number generator architecture. The philosophy architecture is based on SHA-2 (512) hash function whose security strength ensures the unpredictability of the produced random numbers. Furthermore, an FPGA-based implementation of architecture is described. The proposed architecture is a flexible solution in many applications taking into account the performance, power consumption, flexibility, cost and area.

Keywords:

Thermal noise; SHA-2(512) Hash Function; Random Number Generator

1. Introduction

With the development of cryptography, the need for random numbers of high quality is sharply growing. Public/private keypairs for asymmetric algorithms are generated from random bit streams [1][2]; random numbers are also used for key generation in symmetric algorithms for generating challenges in authentication protocols to create padding bytes and blinding values [3]. In the Smart Card application field, countermeasures against side-channel attacks can also require the availability of good quality random number source [4]. The security of many cryptographic systems depends on the assumption that future values in the random sequence can be unpredictable [5], so how to design a random generator with good performance is a challenge.

There are two types of random number generators commonly used for cryptographic applications. The true random number generator (TRNG) derives its output from a physical noise source whereas a pseudo random number generator (PRNG) expands a relatively short key (possibly from TRNG) into a long sequence of seemingly random bits based on a deterministic algorithm. A cryptographically secure random number generator is one which produces sequence for which there is no polynomial time algorithm which, on input of the first l bits of the output sequence s, can predict the $(l+1)st$ bit of s with a probability significantly greater than $\frac{1}{2}$ [6][7]. According to Shannon's mathematical theory of communication, for cryptographically secure random number generators, the entropy of a k bit long stream must be as close as possible to k [9].

In many application where a random bit stream is required, pseudo random number generators are frequently employed, which exploits a deterministic algorithm to generate a bit stream from an initial seed that an external observer can exchange for a random sequence. However, since the adopted algorithms are usually public, the seed becomes the only source of randomness. As a consequence, in a PRNG, the output's entropy can never exceed the entropy of the seed. Therefore, it is necessary that a PRNG be properly seeded from a source of true randomness. In nature, there are many physical phenomena with randomness such as timing user processed, peripheral operations, electronic noise and radioactive decay [6]. But for the hardware implementation, thermal noise is ideally performable. However, due to bandwidth limitation, fabrication tolerances, aging and temperature drifts, and deterministic disturbances, the produced bit streams usually show a certain level of correlation even if well-designed. To address this issue, some countermeasures are adopted to deal with the produced bit stream.

In this paper, a new random number generator is proposed. The proposed architecture, which is based on the thermal noise and SHA-2 (512) hash function, can be implemented in high-speed systems with hardware. In the architecture, the physical random number generator produces the raw bit stream, which will be refined by SHA hash function, which remove non-idealities in the output bit stream from physical random number generator. The SHA hash function offers the more unpredictability in the

produced numbers by highly compressing the raw bit stream and improves the security strength of the proposed architecture. The word length of produced random number is 512 bits. This length value is acceptable in all the random number application. A VLSI implementation for FPGA device of the proposed architecture is described, which will be discussed in detail in next sections. The whole design was captured in VHDL language.

This paper is organized as follow: in section two the thermal noise circuit is given. In the next section the SHA-2 hash function is presented. While in the section 4 the proposed architecture of random number generator is analyzed in detail. The VLSI implementation results are described in the section 5. Finally conclusions are discussed in the last section.

2. The architecture of RNG

The proposed architecture is illustrated in Fig. 1. The system includes two basic parts: the physical random number generator based on thermal noise and SHA-2 (512) hash function.

In the system, the quality of the random sequence, which is engendered by the thermal noise, will be affected by many factors that introduce a certain level of correlation into the sequence. In order to eliminate the correlation and to improve the statistical properties, a decorrelating algorithm should be designed to remedy the raw sequence. In this system, SHA-2 (512) hash function is designed to remove the non-idealities the raw sequence through high compression ratio. Furthermore, the high security level of SHA-2 (512) ensures the computational infeasibility to create an initial message for a given message digests, which is important in cryptographic applications. so we elect SHA-2 as a well-designed decorrelating algorithm.

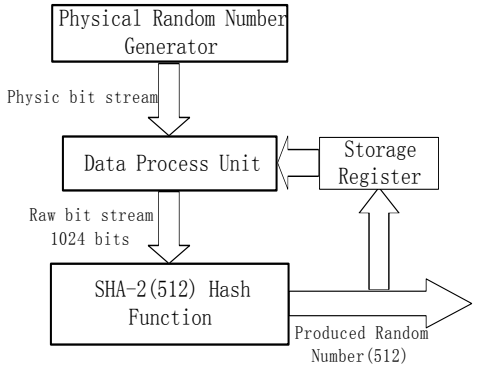

Figure 1.The architecture of the random bit stream

The whole operation scenario of system is simplicity. For the first time, the initial data block, which will be used as the input of SHA-2(512) hash function, is completely contributed by the physical random number generator. After the first operation of SHA-2 (512) hash function, the produced message is equal to the first random number, which will be synchronously feedback to store into the storage register. The second random number generation is different from the first. The only difference is that the initial data block is altered. In this time, the input data of SHA-2 (512) hash function is combined with the previous random number and the random bit stream of the thermal noise, which are processed by Data Process Unit (DPU) and then sent to SHA-2 that will transform them into another random number. The whole process will be repeated continually in the same way..

3. The physical RNG circuits

In general, our physical random number generator consists of three parts: constructing a physical noise source, amplifying the noise and quantifying the noise waveform into a raw bit stream and. The physical random number generator is shown in Fig.2.

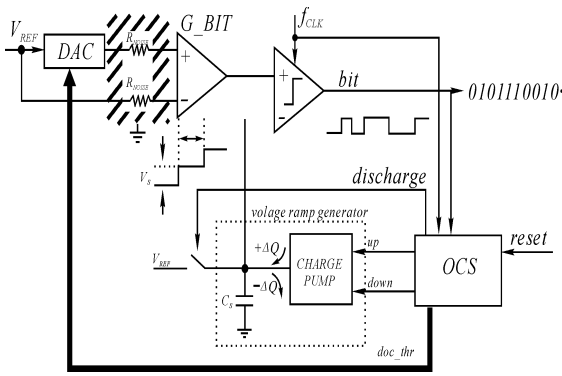

Figure 2.The architecture of physical random number generator

In the above schematics, the integrated resistors[9] contribute thermal noise to the whole system. The amplified noise is compared with a reference voltage of a clocked comparator whose output is a random bit stream. However, there are many factors which can introduce non-random behavior into the raw sequence. Substrate, power supply and external disturbs can induce the periodic patterns into output stream and affect the statistical quality of the random source, which is a main concern in many applications, especially smart card application. In order to eliminate the influences, a number of countermeasures are employed in the above circuits. The noise resistors are designed with an interdigitated layout and surrounded by ground-tied guard

ring. High PSRR topologies are used for the amplifier circuits and a precise offset zeroing system has been developed.

The offset zeroing system also includes a digital to analog (DAC), a voltage ramp generator and an offset control system (OCS). DAC is used to scan the amplifier input bias voltage; once the amplifier bias voltage has been chosen (the amplifier operates in linear region), the comparator reference voltage is changed around V_{REF} to remove the residual offset by means of the voltage ramp generator. The process is controlled by OCS according to the following algorithm:

```
OCS( input: bit; output: dac_thr, up, down, discharge)
    {
      reset:   discharge='1'; wait for T_RST ;
               discharge='0'; dac_thr=0;
      stage1: while (bit='0' and dac_thr< N_TH )
    dac_thr++;
      stage2: while (1) {
                          if (bit='0')   then
                            down='1'
                         else
    up='1';
    wait for  T_S
    down='0'; up='0';
    }
    }
```

where T_{RST} is the capacitor reset time, N_{TH} the DAC threshold number and T_S the duration of the voltage step from the ramp generator.

A topology for a charge pump which allows obtaining a small step size V_S (about 1mV) with a small capacitor C_S (3pF, in the proposed design) is adopted. The feedback loop at comparator's input introduces a high-pass filtering on the amplifier's output noise spectrum with a cutoff frequency $f_c = \frac{V_S}{2\pi T_S}$, where $T_S = n \cdot T_{CLK}$, to remove the offset without affecting the white noise bandwidth.

4. The SHA-2 Hash Function

Hash function is a fundamental primitive category in modern cryptography, often called one-way hash functions. A hash function is computationally efficient function, which maps binary strings of arbitrary length to binary strings of some fixed length, called hash-values [9][11]. That a hash function operates only in one direction is different from the common block ciphers. This means that hash functions produce the output block of any initial value but it is not possible to produce the pre-value for a given hash output. In other words, the security of such an algorithm is ensured due to the computational infeasibility to create an initial message which would corresponds to a given message digest.

The SHA-1 hash function [12] was designed by NIST [13] for the use of the digital signature standard. The SHA-1 algorithm's main operation is to convert the input message of length less than 512 bits into an output message with 160 bits (message digest). The security hash algorithm-1 (SHA-1) is the world's most popular hash function. Unfortunately, the security level of this design is limited to a level comparable to an 80-bit block cipher, so on august 26, 2002 NIST announced the approval of FIPS 180-2[14], Secure hash function which introduces the specification of three new hash function, SHA-2(256,384,512). The hash function of SHA-2 (512) differs most significantly in the number of security bits that are provided for the hashed input data block. Security is directly related to the message digest length, which is useful in the random number generation. In our system, we exploits sufficiently high compression ratio of SHA-2 to remove the effect of the mentioned non-idealities and to improve the statistical properties of the produced random numbers.

The hash function architecture is showed in Fig.2.The PADDER pads the input data and converts them to 1024-bit blocks (padded data), which sequentially process blocks of 1024-bit. The padded message is

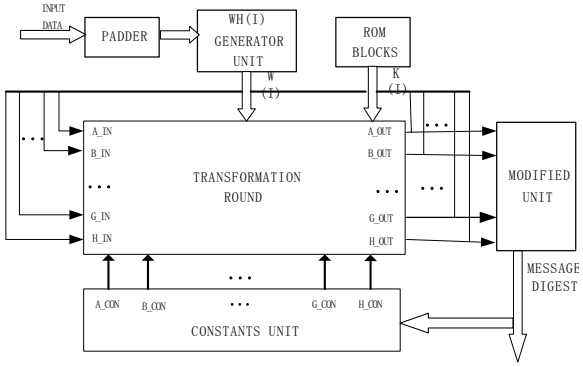

Figure.3 SHA Hash Function Architecture

generated with the following process: a logic "1", followed by m "0"s and followed by a 64-bit integer are appended to the end of the input data message, to produce a padded message to length equal to a multiple of 1024. The 64-bit integer is equal to the length of the input data message. The padded data is a multiple of 1024-bit. The padded data are

divided into N equal data blocks $M^1 \cdots M^N$ and are processed in order, using the following equations:

$$\begin{cases} W_i = M^i & 0 \le i \le 15 \\ W_i = \sigma_1 W_{i-2} + W_{i-7} + \sigma_0 W_{i-15} + W_{i-16} & 15 < i < 80 \end{cases}$$

The algorithm basic transformation round as the specification of the standard defines has to be processed 80 times. The 80×64-bit ROM Blocks are used for predefined K_i constants support. In the last phrase, the modification function processes the data and outputs them.

5. Expermental Results

The proposed architecture has been captured with VHDL and have been synthesized to place and route with XINLINX FPGA Vertex Device (v300pq240). The synthesis result for the proposed RNG implementation is illustrated in the following Table 1.

Table 1. FPGA implementation results

Allocated Area	Used/allocation	Utilization
Fun. Generators	5268/6144	86%
CLB Slices	2710/3072	88%
Dffs or Latches	4182/6144	68%
Frequency	80MHz	

Table 2. Test results before SHA-2(512) process

	P	P	Avg	Pass
Frequency	0.762	0.873	0.805	0.967
Block-Frequency	0.706	0.849	0.810	1.000
Cusum-Forward	0.822	0.861	0.837	0.822
Cusum-Reverse	0.632	0.682	0.664	0.724
Runs	0.242	0.354	0.293	0.872
Long Runs of ones	0.102	0.172	0.135	0.662
Rank(32×32)	0.742	0.812	0.796	0.945
Spectral DFT	0.682	0.752	0.716	0.663
Non-overlapping	0.447	0.506	0.495	0.796
Overlapping	0.114	0.627	0.449	0.801
Universal (L=7,Q=1280)	0.427	0.521	0.483	0.812
Approx Entropy (m=5)	0.462	0.521	0.793	0.897
Lempel-ziv Complexity	0.842	0.865	0.851	0.915
Linear Complexity	0.345	0.521	0.369	0.925
Serial (m=5, $\nabla\Psi_m^2$)	0.632	0.682	0.664	0.724

The statistical qualities is the guideline of random bit sequence, so we adopted two test Standard, FIPS140-1[15] and SP800-22[16], to test the random bit sequences. All the tests have been executed over a 10^6bit long sequence. Both the raw bit sequence of physical random number generator and the bit sequence processed by SHA-2 (512) can completely pass FIPS140-1 test. But, for the SP800-22 test, the two random bit sequences are different. The table 2 and table 3 show the results of the test with SP800-22 for two different kinds of random bit sequence. From the two tables, we can make the conclusion that the random bit stream by processed SHA-2 (512) holds better statistical properties.

Table3. Test results after SHA-2(512) process

	P Low	P High	Avg score	Pass Ratio
Frequency	0.768	0.932	0.886	0.995
Block-Fre	0.811	0.862	0.849	1000
Cusum-Forward	0.917	0.943	0.925	0.987
Cusum-Reverse	0.612	0.689	0.668	0.992
Runs	0.323	0.361	0. 344	0.961
Long Runs of ones	0.187	0.446	0.213	0.885
Rank(32×32)	0.869	0.887	0.878	0.977
Spectral DFT	0.561	0.615	0.573	0.904
Non-overlapping	0.602	0.784	0.671	0.966
Overlapping	0.076	0.083	0.077	0.924
Universal (L=7,Q=1280)	0.437	0.677	0.593	0.992
ApproxEntropy (m=5)	0.691	0.740	0.733	0.948
Lempel-ziv Complexity	0.365	0.419	0.396	0.998
Linear Complexity	0.294	0.315	0.310	0.972
Serial (m=5, $\nabla\Psi_m^2$)	0.685	0.762	0.746	0.997

6. Conclusions

In this paper, a new random number generator architecture and implementation is presented. The thermal noise offers physical source of random bit stream, and the security strength and the advantages of the selected SHA hash function ensure the unpredictability of the produced random number. The proposed architecture is a flexible solution in application cases. In this architecture, we can employ other better algorithms to process the raw random bit stream of the physical random number generator in

order to improve the statistical properties of the random sequence.

Acknowledgements

This paper is supported by the National Natural Science Foundation of China under Grant No. 60373087 and 90104005; the Doctor Foundation Program of Education Department under Grant No: 20020486046.

References

[1] A.J. Menzes, P.C. Van oorschot, and S.A. Vanstone, Handbook of Applied cryptology, CRC Press, New York, 2001.

[2] S.Callegari, R.Rovatti, G.Setti, Embeddable ADC-based true random number generator for cryptographic applications exploiting nonlinear signal processing and chaos, Signal Processing, IEEE Transactions on Volume 53, Issue 2, Feb. 2005 Page(s):793 – 805.

[3] W. Rankl and R.Effing, Smart Card Handbook, 2nd edition, John wiley &Sons, New York, 2000.

[4] P. kocher, J. Jaffe and B.Jun, Differential Power Analysis, Advance in cryptology (crypto 99),M. Wiener,ed., Lecture Notes in computer science 1666, Springe-Verlag,Heidelberg,Germanypp.388-397,1999.

[5] K.H.Tsoi, K.H.Leung and P.H.W.Leong, Compact FPGA-based True and Pseudo Random Number Generators, Field-Programmable Custom Computing Machines, 2003. FCCM 2003. 11th Annual IEEE Symposium on 9-11 April 2003 Page(s):51 - 61

[6] A. J. Menezes, P. C. van Oorschot and A. Vanstone, Handbook of Applied Cryptography, CRC Press, 5th edition, 2001.

[7] K.H.Tsoi, K.H. Leung, P.H.W. Leong, Compact FPGA-based true and pseudo random number generators, Field-Programmable Custom Computing Machines, 2003. FCCM 2003. 11th Annual IEEE Symposium on 9-11 April 2003 Page(s):51 - 61

[8] C.E. Shannon, A Mathematical Theory of Communication, The Bell System Technical Journal, vol.27, pp379-423, July 1948.

[9] N.Stefanou, S.R. Sonkusale, High speed array of oscillator-based truly binary random number generators, Circuits and Systems, 2004. ISCAS '04. Proceedings of the 2004 International Symposium on Volume 1, May 2004 Page(s):I - 505-8 Vol.1.

[10] Jen-shiun chiang and Ming-Da Chiang, ISCAS 2000, (Geneva, Switzerland, 2000)

[11] D. Tom, A. Perez., D. Borrione, E.Bergeret, Electrical, Design of a proven correct SHA circuit, Electronic and Computer Engineering, 2004. ICEEC '04. 2004 International Conference on 5-7 Sept. 2004 Page(s):31 – 34.

[12] Bruce Shinier, Applied Cryptography – protocols, algorithm and source code in c, Second edition, John Wiley and Sons, New York, 1996.

[13] National Institute of Standard and Technology, SHA-1Standard,www.itl.nist.gov/fipspubs/fip180-1.htm.

[14] National Institute of Standard and Technology, SHA-2Standard,www.itl.nist.gov/fipspubs/fip180-2.htm.

[15] National Institute of Standard and Technology, security requirements for cryptographic modules, FIPS 140-1，Jan,1994.

[16] NIST Special Publication 800-22, A Statistical Test for Random and Pseudorandom Number Generators for Cryptographic Application, May 15, 2001.