Design of a true random number generator

Jinsong Wang, Lin Cao and Yi Yang*

School of XiHua University, Sichuan, China

*Corresponding author e-mail: 2140632311@qq.com

Keywords: True random number generator, TRNG, randomness.

Abstract: True random number generator (TRNG) plays an important role in computer information security. Based on the general FPGA technology, this paper implements a TRNG module which can increase the output randomness by using d-trigger sampling stage inverter. The TRNG module is only composed of logic gates and can be integrated into any type of LSIC. The TRNG module is designed with Verilog HDL parameterization and its main design parameters can be modified. The system adopts SOPC structure, completes the random data transmission through DMA, and finally sends it to the superior computer through USB interface. The design of the system was verified and tested on the FPGA of Altera Cyclone IV (EP4CE6F17C8). The random data obtained by sampling 1025 inverters with a 50MHz clock did not need random post-processing, and passed the NIST SP 800-22 randomness test directly. After a series of parameter adjustment experiments, the conclusion is drawn that the output randomness is positively correlated with the inverter series.

1. Introduction

True random number generator (TRNG) plays an important role in computer information security.

True Random Number Generator (TRNG) utilizes unpredictable physical sources in nature to produce unpredictable outputs. Theoretical security can be achieved, that is, an attacker's knowledge of true random number generator will not improve the probability of guessing the unknown output data. In this context, true random Numbers and true random number generators have attracted the attention of scholars at home and abroad. Typical physical sources include nuclear radioactive decay, electrical resistance, transistor thermal noise, atmospheric noise, scintillation noise, etc.

2. Our objectives and implementation plan

The main work and design objectives of this design are as follows:

Design a device that can generate high-speed true random Numbers and send the generated high-speed true random Numbers quickly.

Analyze the randomness quality of the designed true random number generator through the generated experimental data, and calculate and record its statistical characteristics.

In the upper computer by calling the random number generator generated by the random number, to achieve a random number of simple application -- life game.

2.1 System design scheme

The system based on SOPC structure is mainly composed of random number generator, data sending module, Avalon system, data receiving module, communication module and upper computer.

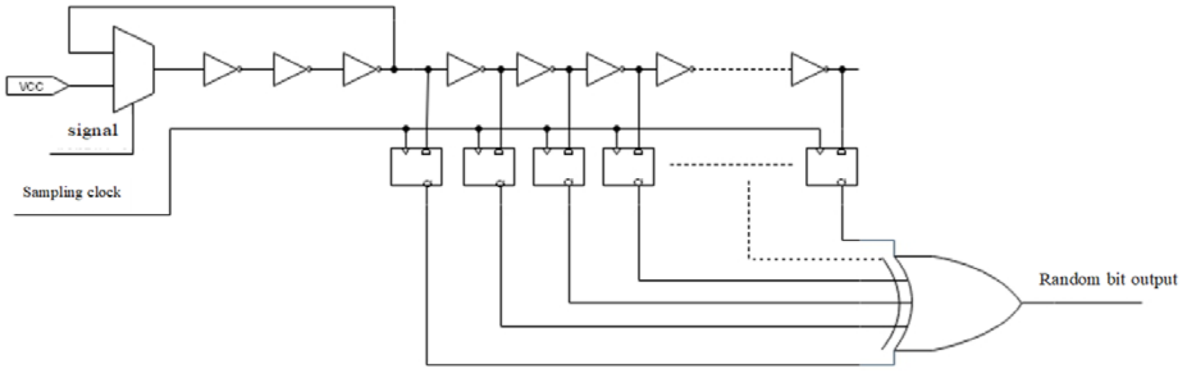


Figure 1. Structure chart of TRNG

The circuit is composed of pure digital circuit, its principle is: when the input signal of D flip-flop does not meet the requirements of the establishment of the trigger time and holding time, the inside of the trigger a node at this time may be in a voltage range, the output cannot be stable in logic 0 or logic 1 state, namely the trigger in the metastable. In this state, the output of the trigger will be uncertain. Is often the case, however, a single trigger every time sampling into metastable probability is very low, so the scheme adopts multiple gate cascade output high frequency oscillation signals, then use multiple gate of cascade D flip-flop output samples at the same time, as long as there is a D flip-flop into metastable, then the final output is uncertain. Finally, the random data of one bit is obtained after different or different outputs of D trigger. Under this structure, as long as the number of D flip-flops is guaranteed, at least one flip-flop can enter the metastable state at each sampling. Therefore, regardless of the sampling frequency, the output value of this scheme has good randomness. On the basis of three non-gates, a double-channel selector is added to form a ring oscillator with enabling signal. When no random number output is needed, the oscillating ring can be closed to reduce the power consumption of the system.

2.1.1 Hardware module design.

Cyclone IV series FPGA chip of ALTERA company is used in this design. Its model is EP4CE6F17C8, and the packaging method is BGA package. There are 256 pins, mainly including User I/O, configuration pins, power supply, clock and special application pins. In addition, there are many pins that need to be connected to GND to ensure a smooth reference ground inside the FPGA. FPGA chip is the core of this circuit. SOPC system is constructed from logic resources inside FPGA chip.

Table 1. Main parameters of the chip

Parameter	Number
Logic elements (LEs)	6272
Embedded memory (Kbits)	270
Embedded 18x18multipliers	15
PLLs	2
Global Clock Networks	10
Number of IO	179
The voltage of kernel	1.15V-1.25V
Working temperature	0-85℃

2.1.2 Design of Clock circuit, FLASH, SDRAM and Buttons and LED.

Adopting 50M active crystal oscillator circuit to provide clock source for FPGA. The crystal oscillator output is connected to the FPGA global input clock pin (CLK1 pin E1), which can be used to drive the user logic circuit inside the FPGA, and the user can realize the clock of other frequencies by configuring the phase-locked loop (PLL) inside the FPGA.

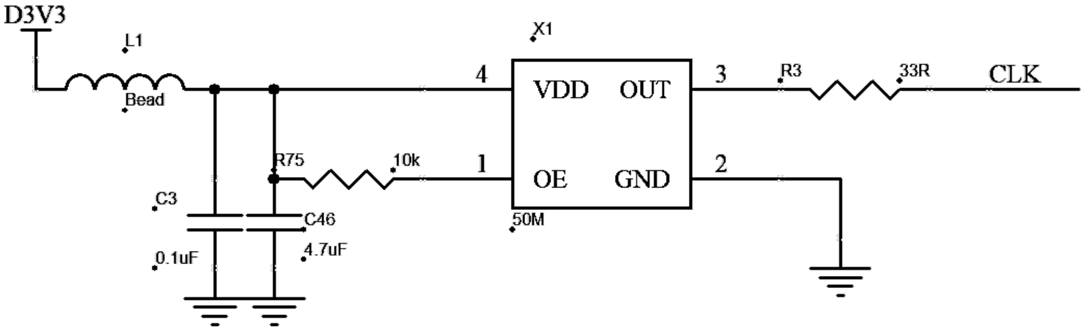


Figure 2. Schematic diagram of active crystal oscillator circuit

The FPGA circuit USES a 16Mbit SPI FLASH chip, model M25P16, which USES 3.3VCMOS voltage standard to completely replace EPCS16, a configuration chip of ALTERA. Because of its non-volatile nature, SPI FLASH can be used as a boot image of an FPGA system. These images mainly include JIC configuration file of FPGA, application code of soft core and other user data files.

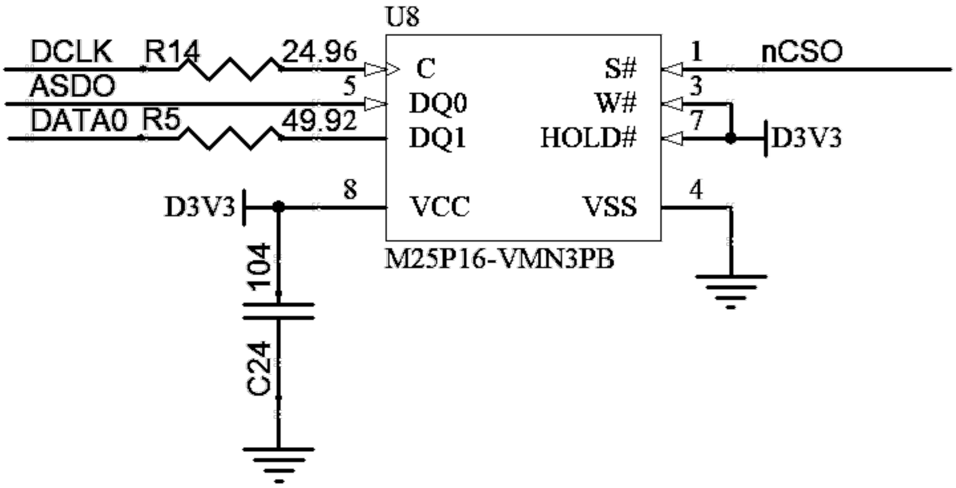


Figure 3. FLASH circuit schematic

The SDRAM chip used in FPGA circuit is HY57V2562GTR with 256Mbit capacity and 16-bit bus length. In SOPC system, SDRAM is mainly used for data caching and storing running programs.

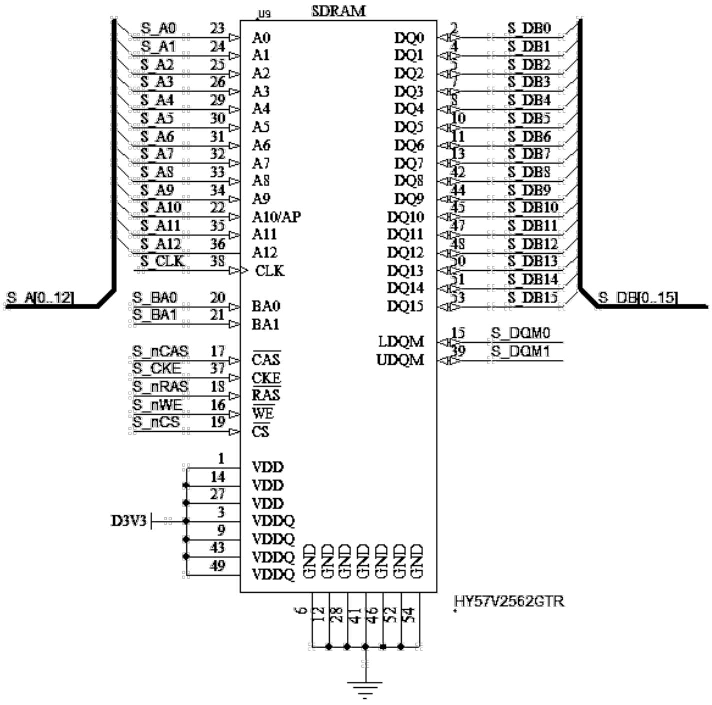


Figure 4. SDRAM circuit schematic

There are 4 independent keys in the circuit: 3 user keys (KEY1~KEY1) and 1 function button (RESET). When pressed, the output is at low level and release is at high level.

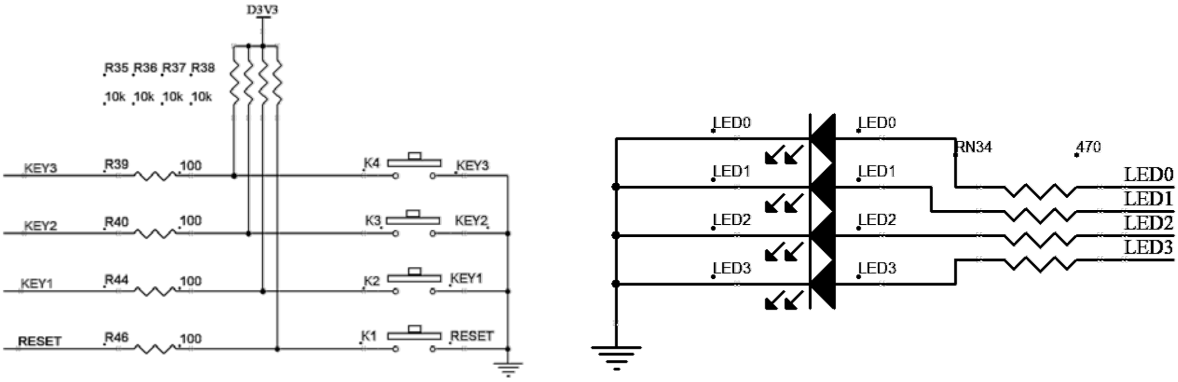


Figure 5. Key and LED circuit schematic diagram

2.2 FPGA logic circuit design

In addition to the USB circuit, the other modules on the FPGA through EDA tools automatically integrated circuit generation. In this paper, Verilog HDL was used to design TRNG, and the structure was simulated and tested. These experiments are implemented on the FPGA platform. Quartus II 12.1 was used to integrate and route the FPGA software.

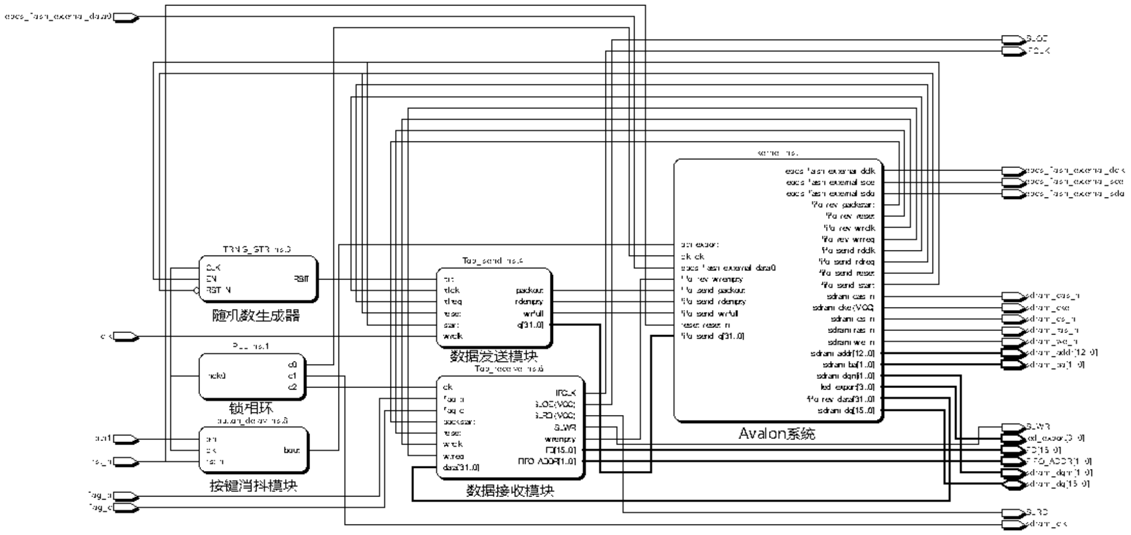


Figure 6. Network list structure diagram

3. System performance test

There are mainly two performance indicators: PROPORTION and p-value. Test software USES these two indicators to determine the randomness of test sequences.

Since each test is independent of the other, the pass rate of the sequence test group can be obtained. Hypothesis test sequence for the m group, significant level for ɑ, there are n sequences of P-Values' Values greater than ɑ, the passing rate $P = n/m$. The confidence interval used to test the pass rate is defined as:

$$\Delta P = \left\{ x \middle| x \in \left(1 - \mathrm{ɑ} - 3\sqrt{\frac{\mathrm{ɑ}(1-\mathrm{ɑ})}{m}}, 1 - \mathrm{ɑ} + 3\sqrt{\frac{\mathrm{ɑ}(1-\mathrm{ɑ})}{m}} \right) \right\} \#(1)$$

If the pass rate p is within the confidence interval Δ p in one or more tests, the likelihood of the test sequence being a random sequence can be considered high. Otherwise, it indicates that the sequence is not random sequence. ɑ = 0.01 in this design, $m = 256$, Then the confidence interval is:

$$\left(1 - 0.01 - 3 * \sqrt{0.01 * \frac{1 - 0.01}{256}}, 1 - 0.01 + 3 * \sqrt{0.01 * \frac{1 - 0.01}{256}} \right) \#(2)$$

This indicates that the p value of the sequence pass rate in the test must be larger than 0.97134, that is, there must be 248 sequences with p-values greater than 0.01.

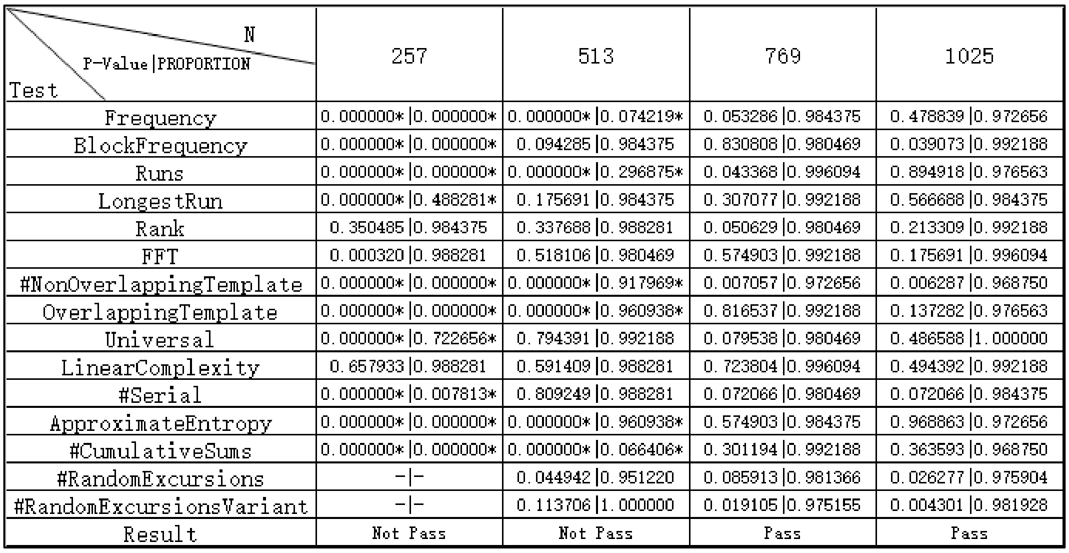
The test of p-value Value is mainly to test the uniformity of sequence distribution. Firstly, the chi-square value of the independent random sequence in group m is calculated as follows:

$$\chi^2 = \sum_{i=1}^{10} \frac{\left(F_i - \frac{m}{10}\right)^2}{\frac{m}{10}} \qquad (i = 1,2,3, \ldots 9,10) \#(3)$$

4. Analysis of test results

When the sampling clock frequency is 50MHz, the cascade number of different inverters N is set and implemented on FPGA to extract the output random data respectively. The results are shown in the table. A "#" sign in the test item indicates that the test item contains multiple subtests. It only gives the minimum result of the subtest item.

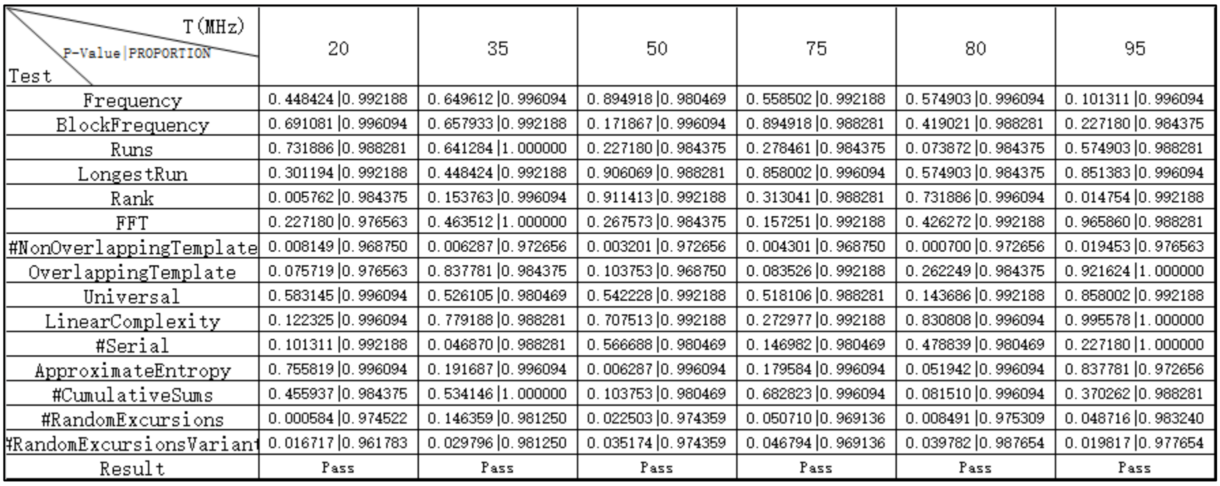
Table 2. Experimental test results.



N P-Value|PROPORTION Test	257	513	769	1025
Frequency	0.000000*|0.000000*	0.000000*|0.074219*	0.053286|0.984375	0.478839|0.972656
BlockFrequency	0.000000*|0.000000*	0.094285|0.984375	0.830808|0.980469	0.039073|0.992188
Runs	0.000000*|0.000000*	0.000000*|0.296875*	0.043368|0.996094	0.894918|0.976563
LongestRun	0.000000*|0.488281*	0.175691|0.984375	0.307077|0.992188	0.566688|0.984375
Rank	0.350485|0.984375	0.337688|0.988281	0.050629|0.980469	0.213309|0.992188
FFT	0.000320|0.988281	0.518106|0.980469	0.574903|0.992188	0.175691|0.996094
#NonOverlappingTemplate	0.000000*|0.000000*	0.000000*|0.917969*	0.007057|0.972656	0.006287|0.968750
OverlappingTemplate	0.000000*|0.000000*	0.000000*|0.960938*	0.816537|0.992188	0.137282|0.976563
Universal	0.000000*|0.722656*	0.794391|0.992188	0.079538|0.980469	0.486588|1.000000
LinearComplexity	0.657933|0.988281	0.591409|0.988281	0.723804|0.996094	0.494392|0.992188
#Serial	0.000000*|0.007813*	0.809249|0.988281	0.072066|0.980469	0.072066|0.984375
ApproximateEntropy	0.000000*|0.000000*	0.000000*|0.960938*	0.574903|0.984375	0.968863|0.972656
#CumulativeSums	0.000000*|0.000000*	0.000000*|0.066406*	0.301194|0.992188	0.363593|0.968750
#RandomExcursions	-|-	0.044942|0.951220	0.085913|0.981366	0.026277|0.975904
#RandomExcursionsVariant	-|-	0.113706|1.000000	0.019105|0.975155	0.004301|0.981928
Result	Not Pass	Not Pass	Pass	Pass

The experiment shows that: The main influence on the randomness of TRNG output is the number of inverter delay chains, independent of the sampling clock. In this design, N=1025 and T=50MHz parameters are used to configure TRNG, and the random number generation rate is 50Mbps. All the NIST tests are passed, which has good randomness and reaches the design index of this design.

Table 3. Experimental test results.



T(MHz) P-Value|PROPORTION Test	20	35	50	75	80	95
Frequency	0.448424|0.992188	0.649612|0.996094	0.894918|0.980469	0.558502|0.992188	0.574903|0.996094	0.101311|0.996094
BlockFrequency	0.691081|0.996094	0.657933|0.992188	0.171867|0.996094	0.894918|0.988281	0.419021|0.988281	0.227180|0.984375
Runs	0.731886|0.988281	0.641284|1.000000	0.227180|0.984375	0.278461|0.984375	0.073872|0.984375	0.574903|0.988281
LongestRun	0.301194|0.992188	0.448424|0.992188	0.906069|0.988281	0.858002|0.996094	0.574903|0.984375	0.851383|0.996094
Rank	0.005762|0.984375	0.153763|0.996094	0.911413|0.992188	0.313041|0.988281	0.731886|0.996094	0.014754|0.992188
FFT	0.227180|0.976563	0.463512|1.000000	0.267573|0.984375	0.157251|0.992188	0.426272|0.992188	0.965860|0.988281
#NonOverlappingTemplate	0.008149|0.968750	0.006287|0.972656	0.003201|0.972656	0.004301|0.968750	0.000700|0.972656	0.019453|0.976563
OverlappingTemplate	0.075719|0.976563	0.837781|0.984375	0.103753|0.968750	0.083526|0.992188	0.262249|0.984375	0.921624|1.000000
Universal	0.583145|0.996094	0.526105|0.980469	0.542228|0.992188	0.518106|0.988281	0.143686|0.992188	0.858002|0.992188
LinearComplexity	0.122325|0.996094	0.779188|0.988281	0.707513|0.992188	0.272977|0.992188	0.830808|0.996094	0.995578|1.000000
#Serial	0.101311|0.992188	0.046870|0.988281	0.566688|0.980469	0.146982|0.980469	0.478839|0.980469	0.227180|1.000000
ApproximateEntropy	0.755819|0.996094	0.191687|0.996094	0.006287|0.996094	0.179584|0.996094	0.051942|0.996094	0.837781|0.972656
#CumulativeSums	0.455937|0.984375	0.534146|1.000000	0.103753|0.980469	0.682823|0.996094	0.081510|0.996094	0.370262|0.988281
#RandomExcursions	0.000584|0.974522	0.146359|0.981250	0.022503|0.974359	0.050710|0.969136	0.008491|0.975309	0.048716|0.983240
#RandomExcursionsVariant	0.016717|0.961783	0.029796|0.981250	0.035174|0.974359	0.046794|0.969136	0.039782|0.987654	0.019817|0.977654
Result	Pass	Pass	Pass	Pass	Pass	Pass

5. Conclusion

In this paper, a true random number generator structure is tried: the output of each level of the inverter delay chain is sampled simultaneously to increase the randomness of the generated sequence. The design adopts pure digital form and is easy to implement the structure on FPGA. At the same time, a SOPC system based on Nios II soft core was designed and built on FPGA to complete the control of true random generator and the transmission process of random Numbers. The data is sent to the upper computer via USB, and the random number generated by TRNG is used to complete the software development in the upper computer. Based on the design idea, set up the corresponding hardware circuit and software design, finished finally got a lot of experimental data, through a series of experiments to verify the performance of this design can meet the design requirements of indicators, and obtained the random properties of this design TRNG has nothing to do with the sampling time and number were positively correlated with inverter delay chain.

Acknowledgments

After continuous efforts and unremitting persistence, the paper was finally completed, and the end of the feeling of hardship and difficult. On the completion of this paper, first of all, I would like to express my heartfelt thanks and sincere respect to the guidance teacher Lin Cao. So many people have helped me. I want to thank everyone who cares and helps me. Thank you.

References

[1] Sunar B, Martin W J, Stinson D R. A Provably Secure True Random Number Generator with Built-In Tolerance to Active Attacks[J].IEEE Transactions on Computers, 2007, 56(1):109-119.

[2] Schindler W, Killmann W.Evaluation Criteria for True (Physical) Random Number Generators Used in Cryptographic Applications[C].Proceedings of Workshop on Cryptographic Hardware and Embedded Systems - CHES 2002, 2003, 2523:431—449.

[3] Jun Benjamin, Kocher Paul. The Intel random number generator, White Paper, 1999.

[4] NIST SP 800-22 rev1a, A Statistical Test Suite for the Validation of Random Number Generators and Pseudo Random Number Generators for Cryptographic Applications[S].

[5] Fuchang Zhao. Design of high-speed image acquisition and transmission system based on CPLD and USB [D]. Harbin university of engineering,2004.

[6] Qian Xin, Ziaoyang Zeng, Guoquan Zhang, etc. Design of array random number generator based on resistance thermal noise [J]. Microelectronics and computers, 2004,21 (7) : 143-146.

[7] Huan Deng, Ronghua Jin, Jun Chen et al. High performance true random number generator based on oscillator [J]. Advances in solid state electronics, 2007,27 (3) : 391. 396.

[8] Istvan, Haller,Suciu, Alin,Cret, Octavian.FPGA based TRNG using automatic calibration[J].2009 IEEE 5TH INTERNATIONAL CONFERENCE ON INTELLIGENT COMPUTER COMMUNICATION AND PROCESSING, PROCEEDINGS,2009:373---376.

[9] Ihor Vasyltsov,Eduard Hambardzumyan,Young-Sik Kim and Bohdan Karpinskyy.Fast Digital TRNG Based on Metastable Ring Oscillator[J].Lecture Notes in Computer Science,2008,5154:164-180.

[10] Jessa, M.,Jaworski, M..Randomness of a combined TRNG based on the ring oscillator sampling method[J].Elektronika,2010,51(12):47---5050.

[11] Lubicz, David,Bochard, Nathalie.Towards an Oscillator Based TRNG with a Certified Entropy Rate[J].IEEE TRANSACTIONS ON COMPUTERS,2015,64(4):1191---1200.

[12] T. Siva.Preventing ADDOS Attack by Using Secure TRNG Based Port Hopping[J].American Journal of Engineering Research,2013,2(5):194-199.

[13] Rahman, Md. Tauhidur,Xiao, Kan,Forte, Domenic.TI-TRNG: Technology Independent True Random Number Generator[J].2014 51ST ACM/EDAC/IEEE DESIGN AUTOMATION CONFERENCE (DAC),2014.