Scalable Truly Random Number Generator

Wang Liao, Meilin Wan, Kui Dai and Xuecheng Zou

Abstract—**Despite the low cost and simple structure to generate pseudo random number, truly random number still has its own place in many security systems. We have proposed a truly random number generator with scalable parameters, different requirements focused on cost or performance can be satisfied with proper voltage and working frequency and the randomness has passed the test under NIST SP 800-22. The entropy source is based on ring oscillator and Von Neumann corrector is adopted to improve the bias. Additionally, some post-process circuit based on LFSR is introduced to flatten the distribution of sequence out from entropy source. Detailed optimization of each part is also discussed. Finally, power evaluation and customization have been done. The power cost can be down to 12.4 uW and the output bit rate can be up to 300Mb/s.**

Index Terms—**Truly random number; ring oscillator; LFSR; scalable**

I. INTRODUCTION

As the basic resource of many information systems, random numbers which can be divided into pseudo and truly ones, have played an important role in various essential areas, such as cryptography algorithms, build in self-test (BIST), telecommunication systems and simulation of complex phenomena. Lots of previous works to generate random numbers of both types have been done.

Compared with certain analog circuits like oscillator and amplifier for truly random number, digital circuits made up by simple logic gates or software running on microprocessor based on specific algorithm are sufficient for pseudo random number. Attracted by the low power, low cost and stable output sequence, many applications choose pseudo random number generator (PRNG) rather than truly random number generator (TRNG). But the fact that PRNG is based on certain algorithm means the randomness is unreliable and the sequence can be predicted. For some critical area like public-key cryptosystem, TRNG is still irreplaceable. This paper has proposed a TRNG based on simple digital circuits, suitable for various applications focused either on cost or performance. The output sequence has also passed the randomness test suite provided by National Institute of Standards and Technology (NIST) [1].

The rest of essay is organized as follow. Section II proposed the entropy source to generate truly random noise. Post-process circuits to optimize the truly random noise both in stability and randomness are introduced in Section III. The relationship between performance and power consumption, along with comparison to related works are discussed in Section IV. Test result of randomness based on the statistical test suite provided by NIST is shown in Section V. Finally, Section VI gives the conclusion and remarks.

II. RANDOM NOISE GENERATION

According to previous works, the source of truly random noise can be divided into three types.

(1) Amplification of noise generated by typical components like resistors [2].

(2) Sampling a jittered high frequency oscillator by another low frequency clock [3].

(3) Discrete and chaos systems generated by certain analog circuits [4].

Method (1) is the most straightforward way and is proposed early. The thermal noise generated by single component is very weak so an amplifier with large gain and wide bandwidth is needed, which means great power consumption and has an unpredictable impact to other part of the system. Chaos system in method (3) is very popular recently, but the switch network and large current used to generate chaos is very complicated and not suitable for our purpose of scalability. Compared to method (1) and method (3), advantages of method (2) are obvious, the oscillator is easy to integrate into circuits and can be implemented by various ways with simple structure.

A. Entropy source

Most oscillators are based on amplifier and comparator [5], we decide to use digital circuits instead of analog ones to achieve better stability and lower the difficulty of integration. The ring oscillator is composed by inverters is shown in Fig.1. The number of inverters must be odd to oscillate. If the latency of each inverter is *t* and the number of inverters is *2n-1*, the period of oscillation will be *2(2n-1)t*.

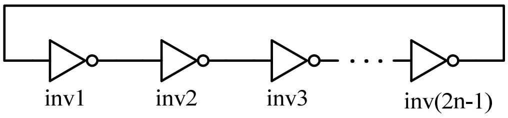

Fig.1. Ring oscillator formed by inverters

The jitter occurs when the latency of inverters varies according to the thermal and shot noise existed in circuit. The output waveform is shown in Fig.4A, and it can be figured that the jitter is not obvious as the influence factor is often very tiny in practice. Some further optimization is needed.

B. Multiple ring oscillator in parallel

Multiple ring oscillator mentioned above is introduced to increase the jitter so that it can be sampled by clock of lower

Wang Liao, Meilin Wan, Kui Dai and Xuecheng Zou are with the School of Optical and Electronic Information, Huazhong University of Science and Technology, Wuhan, CO 430074 P.R.China
Wang Liao (e-mail: liaowangww@163.com).
Meilin Wan: (email: D201277512@hust.edu.cn)
Kui Dai (email: josh.maxview@gmail.com)
Xuecheng Zou (email: estxczou@gmail.com)

frequency. To determine the length of each ring, firstly they should not be the same, as the same length will lead to same frequency, so the entropy will be wasted due to the overlap of transition zones. Furthermore, to minimize the overlapped area, the frequency of each ring should be relatively-prime, which means the length should be relatively-prime, so that the overlap only occurs at the lowest common multiple of each periods. The final architecture is shown in Fig.2, after XOR with each other the output of oscillator is sampled by a D flip-flop.

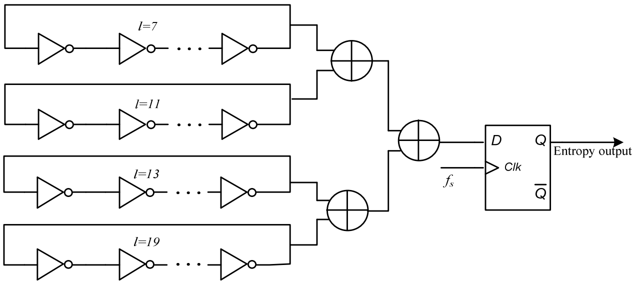

Fig.2. Four ring oscillator in parallel

C. Elimination of bias

Under ideal conditions, after sampled by lower frequency clock, the jitter in oscillator should be totally random and unbiased, which means the possibility of being zero or one is equal. But in practice due to the environmental factors like temperature and voltage, there is always some bias need to be eliminated, especially the bias is also amplified along with the entropy when multiple ring oscillators XOR with each other. Here Von Neumann Corrector is adopted [6], whose function can be described by Table I.

Table I
VON NEUMANN CORRECTOR

Input Value	Output
00	Discard
01	0
10	1
11	Discard

The implementation is shown in Fig.3, only a few logic gates and D flip-flop is needed. It can be figured from Fig.4C that the throughput rate decreased to quarter as the sequence 00 and 11 are abandoned.

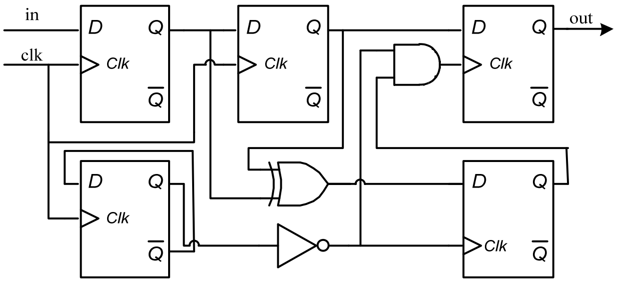

Fig.3. Typical Von Neumann corrector

III. POST-PROCESS CIRCUIT

Like its entropy source derived from noise inside logic gates, the output of Von Neumann Corrector follow normal distribution. Although the mean value approaches zero, the standard of TRNG requires uniform distribution. Post-process circuit based on Linear Feedback Shift Register (LFSR) is proposed to flatten the distribution.

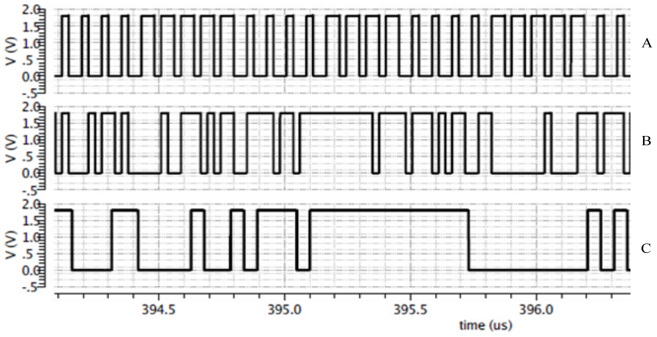

Fig.4. Entropy source waveform

A. Preliminaries

In fact, LFSR has been widely adopted as Pseudo Random Number Generators (PRNG), and its typical structure is shown in Fig.5. The feedback loop can be described by characteristic polynomial as follow and usually $f_m = f_0 = 1$.

$$p(x) = \sum_{i=0}^{m} f_i x_i = f_m x^m + f_{m-1} x^{m-1} + ... + f_1 x + f_0 1$$

Despite its simple structure and low cost, LFSR has two main problems when involved in random number generation. Firstly, the output sequence will repeat after certain cycles for each given initial value of LFSR, and the number of cycles reaches maximum only when the characteristic polynomial is primitive. Secondly, the sequence lacks complexity and fails to past the serial test proposed by Knuth [7]. For example, if the value of an LFSR with length *m* at time t is *v*, which can be represented by the value of each register as $v = (v_m v_{m-1} ... v_1 v_0)$, then at time $t+1$ the value w will be $w = (p(v) v_m v_{m-1} ... v_2 v_1) = v/2 \ or \ v/2 + 2^{m-1}$, where p(v) is the characteristic polynomial. That means there is always a 50% possibility the next value can be predicted.

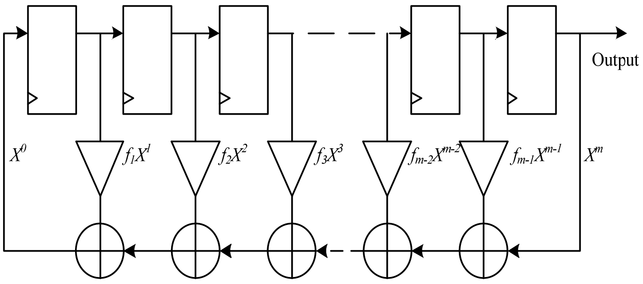

Fig.5. Typical structure of LFSR

In fact the two problems of LFSR mentioned above should be considered together. The feedback circuit is restricted to follow the primitive polynomial to achieve the maximum length of random sequence. But in our work, LFSR is connected with entropy source to be the post-process circuit. The existence of input value makes the state machine of LFSR no longer fixed, so the feedback circuit can be more flexible to become nonlinear.

B. Feedback circuit customization

Without the restriction of primitive, we can customize our own nonlinear feedback circuit. The feedback destination can be any register and the n-th register has n possible

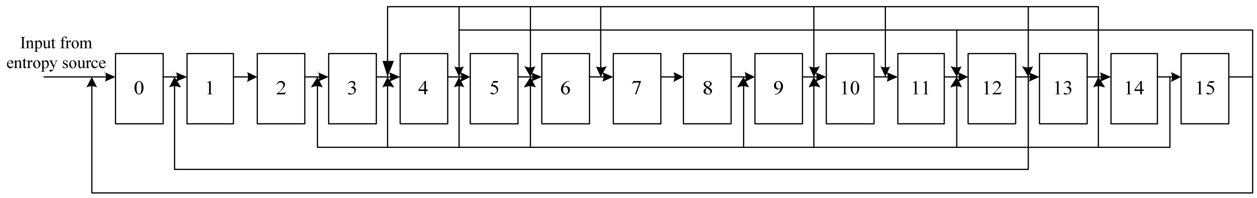

Fig.6. Nonlinear feedback circuit adopted

feedback sources. For an LFSR with length *m*, the feedback circuit can be represented by a variable of $m(m-1)/2$ bits. The variable needs to be adjusted to achieve a good uniform distribution.

For our design, the feedback circuit of 16 bits is shown in Fig.6. It should be reminded that the structure is not fixed, some popular solutions like Genetic Algorithm (GA) can be adopted to customize different structure focused on either performance or cost [8].

C. Final optimization

In fact, the essence of randomness for one bit, is the possibility of being 0 or 1 keeps equal. To achieve this goal, the output of LFSR need proper further optimization. In order to minimize the impact to performance and cost of original LFSR, just some basic combinational logic is sufficient. Firstly, the 1/2 possibility can be made up by

$$\frac{1}{2}=\frac{1}{8}+\frac{1}{8}+\frac{1}{8}+\frac{1}{8}=(\frac{1}{2}\times\frac{1}{2}\times\frac{1}{2})+(\frac{1}{2}\times\frac{1}{2}\times\frac{1}{2})+(\frac{1}{2}\times\frac{1}{2}\times\frac{1}{2})$$
$$+(\frac{1}{2}\times\frac{1}{2}\times\frac{1}{2})$$

Where the multiplication can be reflected by AND gate with 3 inputs, and addition can be reflected by OR gate with 4 inputs. For our LFSR with 16 bits, the equation above can be defined as

$$out=(16\times\overline{13}\times\overline{6})+(13\times9\times\overline{3})+(\overline{16}\times\overline{9}\times1)+(6\times3\times\overline{1})$$

Six out of sixteen bits are selected and after proper inversion and combination, only one AND gate will get 1 at any time, which means this structure is unique and exclusive. The architecture of combinational logic is shown in Fig.7.

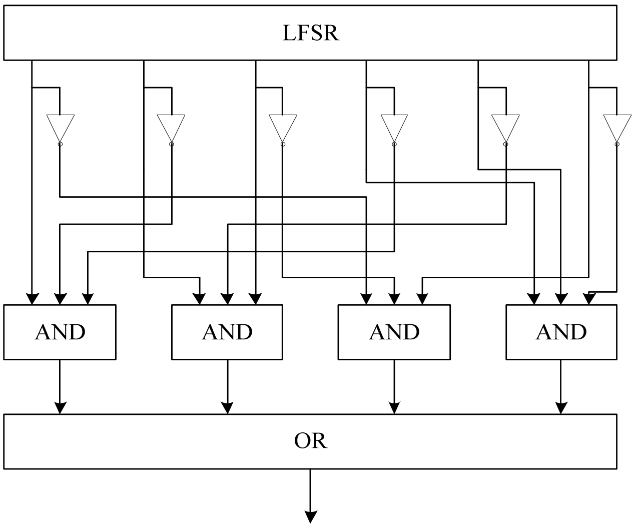

Fig 7. Combinational optimization logic

IV. EVALUATION FOR POWER AND AREA

As we know, for a certain analog or digital circuit with supply voltage and working frequency. Furthermore, there is also a trade-off between voltage and frequency themselves. Apart from the sample clock in entropy source, there are two clocks in our design need to be considered.

A. Oscillation frequency of entropy source

According to our test result, the relationship between oscillation frequency and power consumption of entropy source under different supply voltage is shown in Fig.8. We can figure that lower voltage or lower frequency will both lead to lower power consumption, but with lower voltage, the maximum frequency is limited.

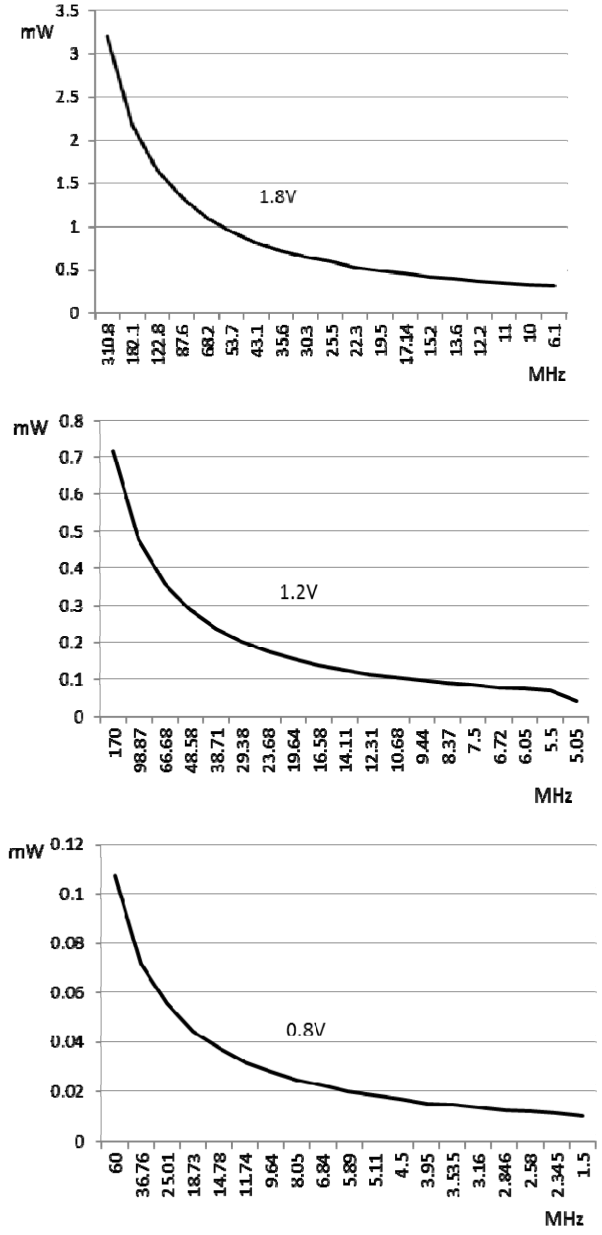

Fig.8. Power of entropy source under different frequency

B. Clock in post-process circuit

Similar as above, the relationship between frequency and power of post-process circuit is shown in Table II. To keep the stabilization of digital circuit, here the minimal voltage has been limited to 1.2V. Moreover, this clock must be slower than or equal with the one in entropy source, because if this clock is faster, the randomness will weaken due to the repeated sample.

Table II
POWER OF POST-PROCESS CIRCUIT UNDER DIFFERENT FREQUENCY

Frequency (MHz)	Power (uW @ 1.8V)	Power (uW @ 1.2V)
1000	4900	2178
100	423	188
10	43	19
1	4.5	2
0.1	0.51	0.227

Having these test results, we can customize proper clock frequency and supply voltage, to satisfy various specific applications focused either on cost or performance. The detail data and comparison between some previous works is shown in Table III. The (c) means when focused on cost, the voltage of entropy source can be set down to 0.8V and the oscillation frequency can be set down to 1.5MHz. The (p) means when focused on performance, the voltage of entropy source can be set up to 1.8V and the oscillation frequency can be set up to 310MHz. The area of high performance is smaller because higher frequency leads to shorter ring oscillator.

Table III
PERFORMANCE AND COST COMPARISON

Design	Tech	Power (uW)	Area (mm2)	Bit Rate (Mb/s)
Our work (c)	0.18 um	12.4	0.02	1
Our work (p)	0.18 um	3540	0.007	300
[5]	0.35 um	3	\	0.2
[9]	0.18 um	1420	0.0304	40
[10]	0.18 um	20000	0.038	125

V.RANDOMNESS TEST

Finally, the random sequence generated from our TRNG is tested under NIST SP 800-22. The 100Mbits test data have been divided into 100 groups with 1Mbits in each group. The p-value and pass rate among all 100 groups of every test is shown in Table IV.

VI. CONCLUSION REMARKS

A truly random number generator with scalable structure is proposed in this paper. The entropy source and post-process circuit both have been optimized in various aspects, to ensure the quality of random sequence generated. The relationship between power and working frequency is discussed, and then evaluation and comparison with previous work has been done. Due to simple digital structure, its supply voltage and working frequency can vary among large range. For different applications, the power cost can down to only 12.4 uW, or the output bit rate can up to 300Mb/s, by customizing the proper parameters. Finally, test results under NIST SP 800-22 shows the randomness is highly reliable.

Table IV
RANDOMNESS TEST RESULTS BY NIST SP 800-22

Statistical Test	P-value (average)	Proportion	Result
Approximate Entropy	0.105937	100%	Success
Block Frequency	0.794631	100%	Success
Cumulative Sums	0.330820	100%	Success
FFT	0.832839	100%	Success
Frequency	0.434215	100%	Success
Linear Complexity	0.379357	100%	Success
Longest Runs	0.656813	98%	Success
Non Overlapping Template	0.793349	99%	Success
Overlapping Template	0.081594	98%	Success
Random Excursions	0.887366	100%	Success
Random Excursions Variant	0.692495	100%	Success
Rank	0.668013	100%	Success
Runs	0.808306	100%	Success
Serial	0.715651	100%	Success
Universal	0.182778	100%	Success

REFERENCES

[1] NIST, A Statistical test suite for random and pseudo-random number generators for cryptographic applications. http://csrc.nist.gov/groups/ST/toolkit/rng/.

[2] Huang Zhun, Chen Hongyi, “A truly random number generator based on thermal noise”, Proceedings of 4th international conference on ASIC, pp.862-864, 2001.

[3] Güler.U, Ergün.S, “A high speed IC Random Number Generator based on phase noise in ring oscillators”, Proceedings of 2010 IEEE International Symposium on Circuits and Systems (ISCAS), pp.425-428, 2010.

[4] Angulo.J.A.A, Kussener.E, et al, “Discrete chaos-based Random Number Generator”, IEEE Faible Tension Faible Consommation, pp.1-4, 2014.

[5] Angulo.J.A.A, Kussener.E, et al, “A new oscillator-based Random Number Generator”, IEEE Faible Tension Faible Consommation, pp.1-4, 2012.

[6] J. V. Neumann, “Various techniques used in connection with random digits”, Applied Math Series, vol. 12, pp. 36–38, 1951.

[7] Knuth, D.E.: The art of computer programming, vol. 2: Seminumerical algorithms, 2nd ed. Reading, MA: Addison-Wesley 1981.

[8] Wang Yuhua, Wang Hongyong, et al, “Evolutionary Design of Random Number Generator”, International Joint Conference on Artificial Intelligence, pp.256-259, 2009.

[9] Tong Zhou, Mingyan Yu, Yizheng Ye, “A Robust High-Speed Chaos-Based Truly Random Number Generator for Embedded Cryptosystems”, 49th IEEE International Midwest Symposium on Circuits and Systems, pp.117-120, 2009.

[10] Güler.U, Ergun.S, Dundar.G, “A digital IC Random Number Generator with logic gates only”, 17th IEEE International Conference on Electronics, Circuits, and Systems, pp.239-242, 2010.