EURASIP Journal on
Information Security

RESEARCH Open Access

ASIC implementation of random number generators using SR latches and its evaluation

Naoya Torii[1,2*], Hirotaka Kokubo[1], Dai Yamamoto[1], Kouichi Itoh[1], Masahiko Takenaka[1] and Tsutomu Matsumoto[2]

Abstract

A true random number generator (TRNG) is proposed and evaluated by field-programmable gate arrays (FPGA) implementation that generates random numbers by exclusive-ORing (XORing) the outputs of many SR latches (Hata and Ichikawa, IEICE Trans. Inf. Syst. E95-D(2):426–436, 2012). This enables compact implementation and generates high-entropy random numbers.

In this paper, we fabricate and evaluate 39 TRNGs using SR latches on 0.18 μm ASICs. Random numbers are generated by XORing the outputs of 256 SR latches. Our TRNGs pass the SP800-90B health tests and the AIS20/31 statistical tests in changing temperatures (from −20 to 60 °C) and voltages (1.80 ± 0.15 V). We also perform an independent and identically distributed (IID) test and calculate min-entropy according to the SP800-90B. With these tests, we are able to confirm that our TRNGs are highly robust against environmental stress. The power consumption and circuit scale of our TRNGs are 0.27 mW and 1240.5 gates, respectively. Our TRNGs that use SR latches are small enough to be implemented in embedded devices.

Keywords: Random number generator, SR latch, AIS20/31, SP800-90B

1 Introduction

In the evolving era of the Internet of things (IoT), the security of embedded devices such as sensors, smart cards, and smart phones has become important for various applications, such as services that collect sensor information, electronic money, and online banking. To achieve security, the devices have to be installed with cryptographic hardware for secure communication and identification/authentication.

Cryptographic hardware achieves high-level security such as through encryption processors, random number generators, and tamper detection circuits.

One important element in the hardware is the random number generator because it generates keys for symmetric-key/public-key ciphers. Random numbers with a low randomness make secret keys and private keys predictable, which enables attackers to eavesdrop on communication and forge signatures. Hence, the quality of random numbers affects the security of a system using embedded devices.

In general, the output sequences of high-quality random number generators are required to have two properties: *randomness* and *unpredictability*. Randomness means that “all elements of the sequence are generated independently of each other, and the value of the next element in the sequence cannot be predicted” [1]. Unpredictability means that “no correlation between a seed and any value generated from that seed should be evident; each element of the sequence should appear to be the outcome of an independent random event whose probability is 1/2” [1], when a random sequence is generated from a random seed.

Random number generators are classified into two subgroups: *true random number generators (TRNGs)* and *deterministic random number generators (DRNGs)*. TRNGs use an analog physical process as an entropy source, such as electrical thermal noise or the jitters of a clock signal. The source is considered to be unpredictable. Hence, if the output sequence of a TRNG has the property of randomness, the TRNG is considered to generate high-quality random numbers. DRNGs use one or more inputs

*Correspondence: torii.naoya@jp.fujitsu.com
[1] FUJITSU LABORATORIES LTD., 4-1-1 Kamikodanaka, Nakahara-ku, 211-8588 Kawasaki, Japan
[2] Yokohama National University, 79-7 Tokiwadai, Hodogaya, 240-8501 Yokohama, Japan

called "random seeds" and generate random sequences by using a deterministic algorithm. DRNGs generate random numbers by programming and satisfy the randomness requirement when they are well designed. Hence, the random seed for DRNGs needs unpredictability because the output sequence is predictable when the seed is known.

Additionally, random number generators are required to generate high-quality random numbers regardless of environmental changes. That is because embedded devices are often exposed to environmental changes. Attackers could intentionally reduce the quality of random numbers by freezing embedded devices or by supplying a lower voltage than rated.

Moreover, for resource-limited embedded devices, TRNGs should be integrated on an application-specific integrated circuit (ASIC). TRNGs that can be integrated on digital ASICs have been proposed; however, there are many problems in terms of noise, power consumption, circuit scale, and design cost. A TRNG using SR latches was proposed as a method to solve these problems [2]. This TRNG has been implemented only on field-programmable gate arrays (FPGAs). ASIC implementation is necessary for the mass production of this TRNG because ASIC has the advantage of lower chip cost, lower power consumption, and faster processing than FPGA. It is unknown whether or not a TRNG on an ASIC is able to generate high-quality random numbers. It is necessary to implement and evaluate a TRNG on an ASIC because random numbers are affected by the characteristics of the semiconductor. TRNGs on FPGAs [2] have only been evaluated with the NIST SP800-22 tests [1]. They have not been evaluated by tests dedicated to physical random numbers, namely BSI AIS20/31 [3] and NIST SP800-90B (first draft) [4] (the latest is the second draft [5]). These tests for physical random numbers will be widely used in the near future. Moreover, the robustness of TRNGs against temperature and voltage fluctuations must be evaluated.

Our contributions In this paper, we implemented TRNGs using SR latches on ASICs on the basis of the TRNG on an FPGA [2]. The reason we focus on this latch-based TRNG is that its design cost is small and high-quality random numbers are expected to be generated in any environment. This paper makes two contributions. First, we fabricated TRNGs on 0.18-μm CMOS ASICs. We evaluated whether or not TRNGs are able to generate random numbers on these ASICs by using the AIS20/31 statistical tests and SP800-90B IID test for true random numbers and examined whether our TRNGs have robustness against temperature and voltage fluctuations. Additionally, we examined the SP800-22 tests for reference. Second, we measured the power consumption and the circuit scale of the TRNGs and examined whether they can be installed in embedded devices.

As a result, our TRNGs on ASICs were able to generate high-quality random numbers even if the environment changed. Thus, our TRNGs are considered to improve the security of embedded devices. In addition, they were found to be small and low-power enough to be implemented in embedded devices.

Organization of this paper This paper is organized as follows. In Section 2, we briefly introduce work related to our research. Section 3 is an outline of a TRNG using SR latches. In Section 4, we describe an ASIC implementation of the TRNG. In addition, we measured the power consumption of the TRNG on an ASIC. In Section 5, we evaluate the quality of the true random numbers from the TRNG by using the AIS20/31 and SP800-90B randomness statistical tests. In Section 6, we discuss randomness by increasing the number of latches, the result of the statistical tests, and a comparison with the previous implementation. Finally, Section 7 is a summary of this research.

Parts of the content of this paper were published as a peer-reviewed conference paper in [6].

2 Related work

The various TRNGs on ASICs and FPGAs have been proposed so far. TRNGs are classified into three types: direct amplification of a noise source, jittered oscillator sampling, and metastable circuits.

The first type is a direct amplification of a noise source [7, 8]. This type uses a random noise source as an entropy source. A random sequence is generated by amplifying a random noise source such as thermal noise digitized by a comparator. This type of TRNG generates a high-quality random number. However, it is difficult to integrate in a high-density digital ASIC because it requires thermal analog sensors and analog amplifiers.

The second type is jittered oscillator sampling. This type uses the jitter of a clock oscillator as an entropy source. A random number is generated by sampling a high-frequency clock with a low-frequency clock by using a D-type flip-flop (DFF). Some TRNGs were proposed [9–13]. In [11], the low-frequency clock is generated by connecting the FPGA to external components (resistances and capacitors). In [10], it is done by using a PLL embedded in the FPGA. In [12], it is done by using a CMOS clock generator in an LSI. This type of TRNG needs analog components. Therefore, it is not cost effective because embedded analog elements and analog/digital mixed ASICs are needed.

The other implementations of the second type use the jitter of a ring oscillator as an entropy source [14–16]. A ring oscillator has a feedback structure composed of an odd number of NOT gates. Random numbers are generated by the exclusive-OR (XOR) of multiple ring oscillator outputs. This type of TRNG generates high-quality random numbers and has robustness against temperature fluctuations. However, it would be not suitable for embedded devices with limited resources because the ring oscillator has large power consumption, noise, and circuit scale.

The third type uses metastable circuits [2, 17–22]. This type of TRNG is suitable for embedded devices because of its small-scale and low-power consumption. The prototypes of these proposed TRNGs were fabricated on CMOS and can generate high-quality random numbers. However, they need an additional dynamic adjustment for the voltage or for the internal elements in the metastable circuit. For example, in [17], a negative feedback loop using a switched capacitor network is used to adjust the DC bias of their proposed bistable circuit. In [19], floating-gate memory is used to adjust the DC bias of the SR latch efficiently. In [18], the bias of the SR latch is controlled by measuring the metastable resolution time. This adjustment needs a dedicated full-custom circuit including analog components, which leads to a large design cost at the transistor level. Moreover, it is necessary to re-design the circuits when implementing them with different CMOS technology because the TRNGs often do not work as expected under a different CMOS technology. In [20], a structure called transition effect ring oscillator (TERO) was proposed and implemented on an FPGA. TERO retrieves the entropy using the oscillatory metastable operation. A TRNG was implemented by the XOR of two TERO outputs on an FPGA, and the output sequences passed the NIST800-22 tests. TERO structure does not require feedback circuits and is expected to be efficiently fabricated on CMOS. In [22], the stochastic model of TERO was validated by implementation of an 28-nm CMOS. However, the hardware size and the throughput are not reported.

Hata and Ichikawa proposed a TRNG using the metastability of SR latches and implemented it on an FPGA [2]. The design cost of this TRNG is quite small because it uses only digital synchronous circuits and no feedback circuits are required. In addition, the TRNG can save power consumption by stopping the clock signal input to the SR latches when random number generation is not required. The random numbers from the TRNG pass the NIST SP800-22 statistical tests. For the abovementioned reasons, the TRNG proposed by Hata and Ichikawa is expected to have better properties for embedded devices than other TRNGs.

3 Random number generator using SR latches

3.1 General physical true random number configuration

A TRNG generally consists of three components: an entropy source, a conditioning component, and a health test component [3, 4]. A block diagram of a general TRNG is shown in Fig. 1 [4]. The entropy source is “a component, device, or event that generates unpredictable output values” [3]. The conditioning component reduces the bias and outputs a random number. If the entropy source output has a high entropy, the conditioning component is not always necessary. Therefore, it is optional. The health test component checks the failures of the entropy source continuously. If the health test detects failures, it outputs an alarm signal.

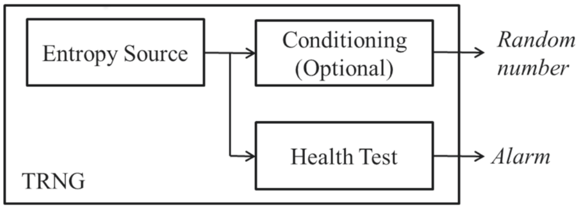

Fig. 1 General TRNG. A general TRNG generally consists of three components: an entropy source, a conditioning component, and a health test component. The entropy source is “a component, device, or event that generates unpredictable output values” [3]. The conditioning component reduces the bias and outputs random numbers. The health test component checks the failures of the entropy source continuously

In this paper, we implemented a new entropy source to evaluate its entropy. We call the source a “TRNG” and call the entropy source output a “random number.” We evaluated the entropy to decide whether the conditioning component is necessary.

3.2 Random number generator using SR latches

In this section, we explain the method for generating random numbers that was proposed by Hata and Ichikawa in [2]. Their TRNG generates random numbers on the basis of the metastability of SR latches. An SR latch consists of two NAND gates and is commonly used to store one bit of information. In this TRNG, the same signal is input to the $\overline{S}$ and $\overline{R}$ of the SR latch, as shown in Fig. 2. When input $= 0$, the SR latch is stable with output $= 1$. When input changes from 0 to 1, the SR latch temporarily enters

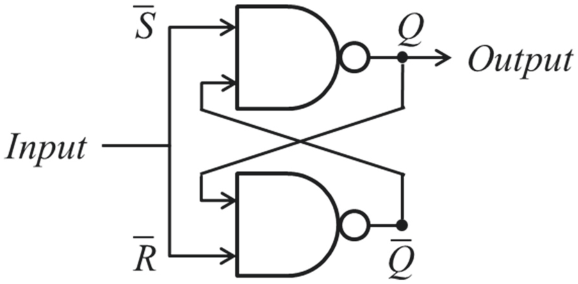

Fig. 2 SR latch. An SR latch consists of two NAND gates and is commonly used to store one bit of information. In this TRNG, the same signal is input to the $\overline{S}$ and $\overline{R}$ of the SR latch

a metastable state, and then it is stable with output $= 0$ or 1. Random numbers can be obtained from output by giving input clock signals by using this behavior. Ideally, the probability of outputting 0 and 1 is equal; however, this probability is actually biased. This is because of the difference in wiring delay between gates or the difference in drive capability between two NAND gates. In many cases, this SR latch generates only 0's or only 1's.

In this paper, we divide the SR latch into two types. One is a random latch, and the other is a constant latch. We defined a "random latch" as an SR latch whose output sequence includes at least one transition between 0 and 1 and defined a "constant latch" as an SR latch that generates only 0's or only 1's.

As described in Section 2, it is difficult to generate high-quality random numbers by using only one SR latch without controlling the latch characteristics continuously on the basis of feedback from output sequences. Hata and Ichikawa [2] proposed a new TRNG consisting of multiple SR latches and an XOR gate without feedback. This TRNG generates random numbers by XORing multiple SR latches' outputs. Hata and Ichikawa describe the implementation method on FPGA that was used to increase the number of random latches. Additionally, the number of latches to be XORed is evaluated. As a result, 64 or more latches are necessary to generate random numbers to pass the NIST SP800-22 statistical tests [1]. This TRNG can reduce the bias of each latch by XORing many latches to generate high-quality random numbers.

Problems There are three problems in [2]. First, this TRNG was implemented only on FPGAs. Second, it was not evaluated in various environments. Third, only one TRNG was evaluated. It is difficult to implement an FPGA in mass-produced embedded devices such as smart cards due to a large power consumption and chip cost, so ASIC implementation is necessary for mass production. TRNGs for embedded devices must be able to generate high-quality random numbers in any environment. If a TRNG generates random numbers with low entropy due to environmental changes, the security of an embedded device is compromised because secret information may be predicted by attackers. In general, the characteristics of a semiconductor, for example, drive capability and wire delay, are influenced by both temperature and voltage changes. Therefore, the quality of random numbers from TRNGs is affected by both changes. Hence, robustness against changes should be evaluated, but as yet, it has not been. In addition, the semiconductor characteristics are slightly different from chip to chip. Hence, multiple chips should be evaluated.

4 ASIC implementation

We fabricated TRNGs using SR latches on 0.18-μm CMOS ASICs (Fujitsu CS86 series [23]). The TRNGs generate random numbers from the XOR of the outputs of 256 SR latch units, as shown in Fig. 3.

The SR latches were custom-designed on the circuit layout and implemented as a hard macro. This macro is called an "SR latch unit" in this paper. 256 SR latch units, XOR, and DFF are implemented automatically by using circuit design tools. Hence, the design cost is quite small.

A block diagram of the SR latch unit is shown in Fig. 4. It consists of two NAND gates and two buffer gates. To increase the number of random latches, the SR latch unit was designed as follows. Both two NAND gates and two buffer gates were selected to have the same characteristics. The wiring lengths between the NAND gates and buffer gates were designed to be the same length; that is, they were the same length from the SR latch unit input to each NAND input, from the NAND output to the other NAND input, and from the NAND output to the buffer input. This enabled the corresponding wiring capacitance and resistance within the SR latch unit to be equal. In FPGA implementation [2], DFFs are used instead of the buffer gates. DFF implementation may increase the number of random latches; however, in our implementation, we used the buffer gate to reduce the hardware size. This SR latch unit structure is expected to increase the number of the random latches.

The TRNGs were assembled as 28-pin dual in-line (DIP) packages. Two types of TRNGs were fabricated: 20 standard TRNGs (using CS86MN, called "MN-TRNG") and 19 low-power-consuming TRNGs (using CS86ML, called "ML-TRNG").

The cell sets of CS86MN have standard transistor characteristics, and those of CS86ML have transistor characteristics with a low leak current. CS86ML is suitable for designing mobile devices that require low power consumption. For example, the delay time, power consumption, and leakage power of 2-input NAND cells is 88 ps, 40.1 nW/MHz, and 0.023 nW for CS86MN and 136 ps, 38.3 nW/MHz, and 0.0067 nW for CS86ML [23].

4.1 Power consumption and circuit scale

We measured the power and current consumption of the TRNGs with a direct current ammeter. According to our

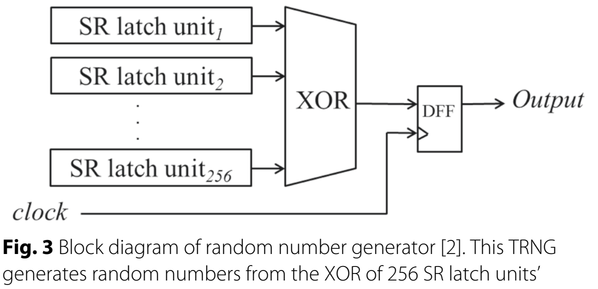

Fig. 3 Block diagram of random number generator [2]. This TRNG generates random numbers from the XOR of 256 SR latch units' outputs

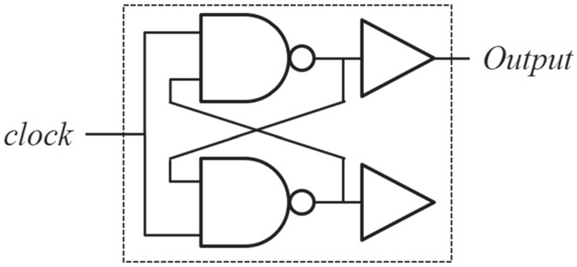

Fig. 4 SR latch unit. A SR latch unit consists of NAND gates and buffer gates. The wiring lengths between gates are designed to be equal to increase the possibility of the latches being random latches

experimental measurements, the average power/current consumption of both the MN-TRNG and ML-TRNG ASICs is 270 μW/150 μA and 252 μW/140 μA, respectively. The current consumption of common ASICs used for contactless smart cards is approximately 1 mA [24]. The current consumption of our TRNGs was much smaller than this value, so the TRNGs are practical and useful.

Additionally, we measured the circuit scale of our TRNGs. In the following discussion, one gate is equivalent to a 2-1 NAND gate (2-bit input and 1-bit output). The TRNG consists of 256 SR latch units including two 2-1 NAND gates and two buffer gates, a 256-1 XOR gate, and a 1-bit flip-flop to store a random number temporarily. Our TRNG was synthesized with Design Compiler 2003.03, and the circuit scale was 1240.5 gates. This evaluation did not include the 256-1 multiplexer (256-1 MUL) and the 2-1 multiplexer (2-1 MUL) which will be described in Section 5.1 because they were for the evaluations of each latch output and not necessary for practical use. On the other hand, we measured the power consumption of our chip including the 256-1 MUL and the 2-1 MUL. We considered the power consumption was negligible because it was measured when the 256-1 MUL was not used and 2-1 MUL passed the random numbers.

We consider our TRNGs to be small enough to be implemented in a chip for embedded devices.

4.2 Clock frequency

We selected the clock frequency (that is, the throughput of the TRNG) as 2.5 MHz which was fast enough for embedded systems to generate session keys and session random numbers. We implemented the TRNG on an FPGA and confirmed the evaluations of the clock frequency described in [2]. From this evaluation, we selected the clock frequency of the ASIC implementation with enough margins. The fastest throughput of our TRNG is the future work to be considered.

5 Evaluation

In this section, we evaluate whether our TRNGs fabricated on ASICs generate high-quality random numbers regardless of environmental changes. As mentioned in Section 3, TRNGs may be influenced by both temperature and voltage fluctuations.

5.1 Evaluation system

Our experimental system for the acquisition of random numbers is shown in Fig. 5. This figure shows the main function blocks. The system consists of two boards: a custom-made board for the ASICs of the TRNGs and a Spartan-3E starter kit board with a Xilinx FPGA for controlling the TRNGs [25] (hereafter, called "FPGA board"). The core voltage to the TRNGs was supplied by using a stabilizing power supply, which was able to adjust the supply voltage at intervals of 0.01 V. The clock signals were input to the TRNGs through the FPGA board. The clock by the 50-MHz oscillator was divided into 2.5 MHz by a digital clock manager (DCM) on the FPGA board and inputted to the TRNG chip. Random numbers generated by the TRNGs were written to a micro SD card via a block RAM and SD writing module on the FPGA board. We acquired not only random numbers but also the output of each latch for further evaluation. The PC set commands to the command register in the TRNG chip via a PC interface on the FPGA board. By setting the command register, two multiplexers, the 256-1 multiplexer (256-1 MUL in Fig. 5) and 2-1 multiplexer (2-1 MUL in Fig. 5), were controlled to output either random numbers or each latch. The random numbers or the outputs of each latch in SD card were moved to a hard disk in the PC manually.

In this environment, we evaluated the random numbers generated by all of the 39 TRNGs while changing the temperature and voltage. The core voltage was changed to 1.65 V (1.80–0.15 V), 1.80 V (standard), and 1.95 V (1.80 + 0.15 V) by the stabilizing power supply. The temperature was maintained at −20 and 60 °C by using a constant temperature oven. We also collected data at room temperature (≈ 27 °C). Only the custom-made board for the TRNGs was put in the constant temperature oven. The FPGA board was always operated at the rated voltage and room temperature. These two boards were connected through a low/high temperature resistant cable.

5.2 Evaluation of randomness

5.2.1 Preliminary: health test by SP800-90B

As a preliminary test, we evaluated random numbers with our TRNGs by using a health test defined in SP800-90B [4]. This test checks for failures of the entropy source continuously. Therefore, a health test circuit should be embedded in our TRNG chips. However, we did not implement this circuit on our fabricated chips because the purpose of the implementation was to evaluate the

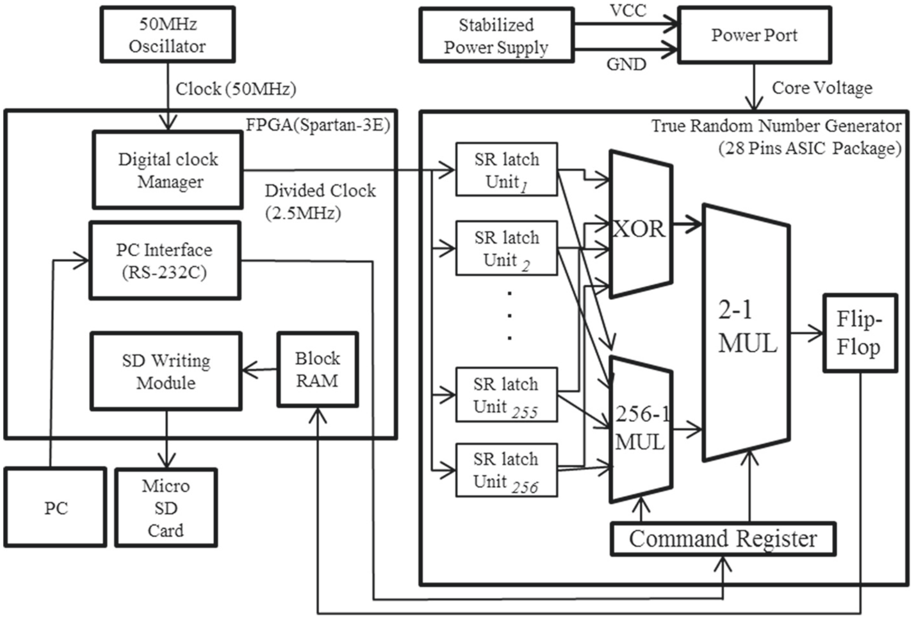

Fig. 5 Experimental system for the acquisition of random numbers. Our experimental system consists of two boards: a custom-made board for the ASICs of the TRNGs and a Spartan-3E starter kit board with a Xilinx FPGA for controlling the TRNGs. Random numbers generated by the TRNGs were written to a micro SD card via a block RAM and SD writing module on an FPGA board. We acquired not only the random numbers but also the output of each latch

entropy of a latch-based random number generator. Instead, we performed the health test off-line on a PC using the random numbers acquired by the evaluation system in Section 5.1 because the TRNGs would not be suitable for practical use if the random numbers fail the health test. We acquired approximately 5.5 Mbits of random numbers from each TRNG while changing the temperature and voltage.

The health test consists of two tests: a repetition count test and an adaptive proportion test. A *false positive rate*, which is the probability of ideal true random numbers failing these tests, is set to 2^{-30} as recommended in SP800-90B.

[***Repetition count test***] The goal of the repetition count test is "to quickly detect a catastrophic failure that causes the noise source to become 'stuck' on a single output value for a long time" [4].

The test procedure is as follows. If the same value (0 or 1) appears consecutively c times or more in a sequence of random numbers, the random numbers are a failure, where $c = \text{ceiling}\ (1 + 30/\text{min-entropy})$. In this paper, c is 32, and *min-entropy* will be mentioned in Section 5.2.3.

[***Adaptive proportion test***] The goal of the adaptive proportion test is "to detect a large loss of entropy, such as might occur as a result of some physical failure or environmental change affecting the noise source" [4].

The test procedure is as follows. First, we obtain a 1-bit value from the beginning of the random numbers as a reference value. Second, we obtain one *block* from the succeeding random numbers. The bit length of a block is represented by the *window size*, and if the reference value appears greater than the *cutoff* times in a block, the random numbers are a failure. The size of the *cutoff* is defined by the *false positive rate*, *min-entropy*, and *window size*. This procedure is repeated until the end of the random numbers. In our evaluations, the *window size* and *cutoff* were 64 and 55 for test settings I and 4096 and 2240 for test settings II. That is, about 84,700 blocks were evaluated for test settings I, and about 1350 blocks were evaluated for test settings II, in each case of random numbers.

In this test, the cutoff value is different from that in SP800-90B when the window size is 64 because the value is expected to be incorrect on the basis of the calculation written in the footnote on page 35 in SP800-90B [4].

The random numbers from the MN-TRNGs and ML-TRNGs passed all of the health tests in the nine cases of temperatures and voltages. In [6], ML-TRNGs failed some of the tests in some of the conditions. However, by correcting the cutoff value, the ML-TRNGs passed all health tests in the nine cases. From this test, our TRNGs would pass the health test continuously if the health test circuit were embedded in our TRNG chip.

We did not evaluate the total failure test in AIS20/31 for the same purpose, because "the test should be selected with regard to the stochastic model of the noise source" [3] and the stochastic model of our TRNG is further work to be considered.

5.2.2 AIS20/31 statistical tests

We evaluated the random numbers in various temperatures and voltages by using the AIS20/31 statistical tests [3]. The data set for this test was same as that in Section 5.2.1. AIS20/31 includes an evaluation criterion for the true random number generators defined by BSI, i.e., the German Federal Office for Information Security.

AIS20/31 includes eight standard statistical tests such as the poker test, the long run test, and the uniform distribution test. We evaluated our TRNGs by using the statistical tests for test procedure A in AIS20/31. In this procedure, five statistical tests out of eight are used. For an ideal random number, the probability of passing the tests is ≈ 0.9987, and that of failing more than two tests is ≈ 0. If one test is failed, a second run is done. If the second test is failed, test procedure A is failed [3].

AIS20/31 classifies TRNGs into two classes: the PTG.1 class and PTG.2 class. The TRNGs in the PTG.1 class can be used for random number generation for challenge and response authentication. The TRNGs in the PTG.2 class can be used for key and seed generation for pseudo random number generators, so these TRNGs provide higher security than those in the PTG.1 class.

Figure 6 shows the rate of TRNGs that passed the AIS20/31 tests. The horizontal axis shows the environment at various temperatures and voltages. The vertical axis shows the rate of the PRNGs that passed the tests. The MN-TRNGs passed the tests in all cases, so our MN-TRNGs have robustness against temperature and voltage fluctuations. The ML-TRNGs, however, failed tests only in two cases out of 171 cases. Two of the 19 ML-TRNG chips failed the tests, and each ML-TRNG chip failed one case out of nine. In other words, 17 chips out of 19 passed all AIS20/31 statistical tests. For reference, we retested the random numbers of two failed chips by retrieving the numbers from the random number files by deleting the first 10,000 bits and adding the bits at the end of the original random numbers. Both of the random numbers passed the AIS20/31 statistical tests. We believe that the ML-TRNGs also have robustness against temperature and voltage fluctuations regardless of the failures of the first tests. In Section 6.1, we discuss the ML-TRNGs from the aspect of the number of SR latches.

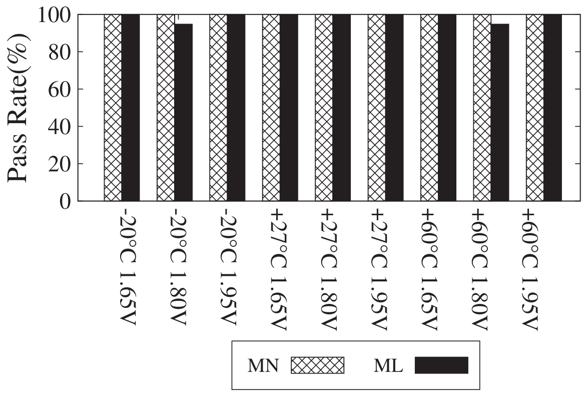

Fig. 6 AIS20/31 pass rate. The MN-TRNGs passed the AIS20/31 tests in all 180 cases. The ML-TRNGs, however, failed tests in two cases out of 171

From these evaluations, our TRNGs passed the standard statistical test for PTG.1 and PTG.2 from the aspect of AIS20/31. More tests including the total failure test and the online test are required to regard our TRNGs as PTG.1, and additionally, a stochastic model of the entropy source is required for PTG.2; however, this further work is to be considered.

5.2.3 SP800-90B IID test and min-entropy

We use *min-entropy* as the objective criterion of randomness. Min-entropy is defined as the lower bound of the amount of information of a random variable [4]. The min-entropy per bit of ideal true random numbers is 1 because the proportion of 0's and 1's is ideally 0.5. The method of estimating the min-entropy differs depending on whether the TRNG is independent and identically distributed (IID). A random sequence is evaluated as IID when "each element of the sequence has the same probability distribution as the other values and all values are mutually independent" [4]. The evaluation consists of six shuffling tests and a statistical test. Therefore, first, we implemented software for IID verification tests in accordance with SP800-90B and evaluated our TRNGs by using this test. The data set for this test was same as that in Section 5.2.1.

Figure 7 shows the IID test pass rate for the nine cases of temperatures and voltages. From the results, the MN-TRNGs passed all 180 cases, so we performed min-entropy estimation for IID sources (see Section 9.2 in [4]). In contrast, the ML-TRNGs passed 166 out of 171 cases. We, however, regarded the ML-TRNGs as IID sources because all 19 chips passed the tests under normal conditions, i.e., 27 °C and 1.80 V, and 14 chips out of 19 passed all tests in the nine cases of temperatures and voltages. To pass all tests, we believe that a ML-TRNG is necessary to connect an appropriate conditioning component or to increase the number of latches, as discussed in Section 6.

Figures 8 and 9 show the results of min-entropy estimation in MN-TRNGs and ML-TRNGs, respectively. The x indication, the upper line, and the lower line show the average, maximum, and minimum of the min-entropy per bits in all test cases, respectively. The min-entropy is very close to "1" (i.e., ideal min-entropy) in both types of

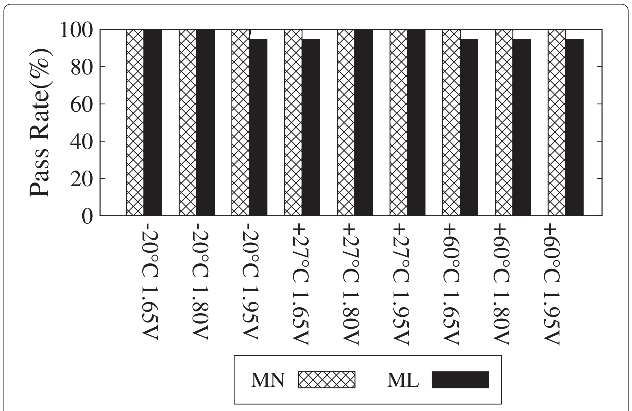

Fig. 7 IID evaluation test pass rate. The MN-TRNGs passed the IID evaluation test in all 180 cases. In contrast, the ML-TRNGs passed 166 out of 171 cases. We, however, regarded the ML-TRNGs as IID sources because they passed the tests under the normal conditions, i.e., 27 °C and 1.80 V

PRNGs. Hence, our PRNGs have a very high min-entropy regardless of the temperature or voltage.

5.2.4 SP800-22 statistical tests

We evaluated the random numbers by using the SP800-22 statistical tests [1] for reference because these tests are said to be useful for analyzing random number sources in AIS20/31 [3]. We used an NIST statistical test suite called "sts-2.1.2" [26] and the recommended parameter settings of AIS 20/31. We selected a MN-TRNG chip and acquired random numbers at the room temperature (25 °C) and the rated voltage (1.8 V). Two data sets (data set 1 and 2) were evaluated and the size of each data set was about 2^{30} bits. The results of the tests are shown in Table 1. The asterisk denotes that a test consisted of several sub-tests and that the minimum p value is shown. The data set 1 passed the all tests and the data set 2 passed tests except for non-overlapping template matching test. The values of the failed test are shown in *bold* type. We consider the number of chips should increase to evaluate the sequence of MN-TRNG correctly. If the non-negligible number of chips fail the SP800-22 statistical tests, MN-TRNG would need the appropriate conditioning component or increase the number of latches in the MN-TRNG to pass all statistical tests in SP800-22.

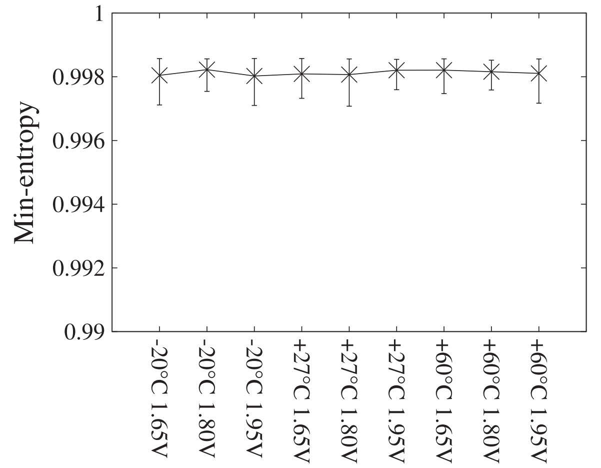

Fig. 8 Min-entropy estimation (MN). The min-entropy is very close to "1" (i.e., ideal min-entropy) in the nine cases of temperatures and voltages. The *x* indication, the *upper line*, and the *lower of line* show the average, maximum, and minimum of the min-entropy in the test cases, respectively

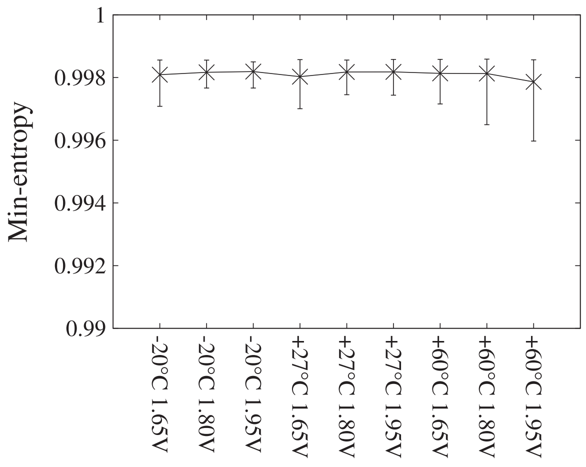

Fig. 9 Min-entropy estimation (ML). The min-entropy was very close to "1" (the ideal min-entropy) in the nine cases of temperatures and voltages. The *x* indication, the *upper line*, and the *lower line* show the average, maximum, and minimum of the min-entropy in the test cases, respectively

5.3 Evaluation of random latches

In this section, we evaluated the behavior of each SR latch in order to clarify the reasons for the robustness against temperature and voltage fluctuations of our TRNGs.

We defined the quality of the random latches based on the proportion of 1's in an output sequence. If the sequence of a random latch has an equal distribution of 0 and 1, it is considered to be a good random number. Therefore, 50 % is the highest quality, and 0 % (all 0) or 100 % (all 1) is the worst. In [2], it is reported that the output of the TRNG passed the SP800-22 statistical tests when there were one high-quality random latch and 17 low quality ones. Therefore, our TRNG generates good random numbers, if there are a few high-quality random latches against temperature and voltage fluctuations.

We focused on two evaluation axes: the number of random latches and the quality of random numbers from each random latch in our TRNG. We acquired an output

Table 1 NIST 800-22 statistical test result. We evaluated the random numbers by using the SP800-22 statistical tests [1] for reference. We used an NIST statistical test suite called "sts-2.1.2" [26] and the recommended parameter settings of AIS 20/31. We selected a MN-TRNG and acquired random numbers at the room temperature (25 °C) and the rated voltage (1.8 V)

	Data set 1		Data set 2	
Statistical test	*p* value	Proportion	*p* value	Proportion
Frequency	0.352760	0.9897	0.981009	0.9869
Block frequency	0.198458	0.9814	0.065912	0.9944
Cumulative sums*	0.017970	0.9888	0.100602	0.9879
Runs	0.392050	0.9888	0.423587	0.9879
Longest run	0.054577	0.9860	0.193532	0.9870
Rank	0.013864	0.9897	0.773966	0.9888
FFT	0.405158	0.9935	0.428692	0.9832
Non-overlapping template*	0.009459	0.9897	0.763056	*0.9786*
Overlapping template	0.725810	0.9851	0.067077	0.9907
Universal	0.571253	0.9879	0.731471	0.9907
Approxi. entropy	0.163454	0.9916	0.144099	0.9934
Random excursions*	0.095691	0.9985	0.120217	0.9807
Random excursions varian*	0.097473	0.9923	0.035873	0.9985
Serial*	0.034210	0.9935	0.557853	0.9925
Linear complexity	0.051110	0.9860	0.750162	0.9888

Two data sets (data sets 1 and 2) were evaluated and the size of each data set was about 2^{30} bits. The asterisk denotes that a test consisted of several sub-tests and that the minimum *p* value is shown. Data set 1 passed all the tests and data set 2 passed the tests except for non-overlapping template matching test. The values of failed tests are shown in italic type

sequence of about 21K bits from each SR latch in the nine cases of temperatures and voltages per chip.

5.3.1 The number of random latches

We evaluated the number of random latches in the 256 latches. Figures 10 and 11 show the number of random latches in MN-TRNGs and ML-TRNGs, respectively. The bars graphs and the upper and lower lines show the average, maximum, and minimum of the number of random latches in the test cases.

The higher the temperature and voltage were, the larger the average number of random latches was for both types

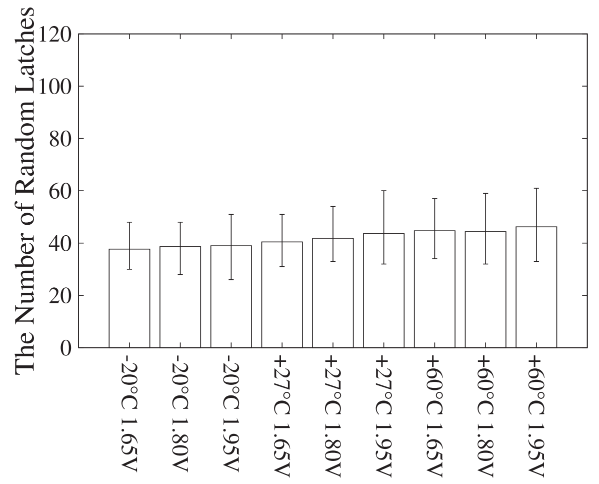

Fig. 10 Number of random latches (MN). The *bar graphs* and the *upper* and *lower lines* show the average, maximum, and minimum of the number of random latches in the nine cases of temperatures and voltages, respectively

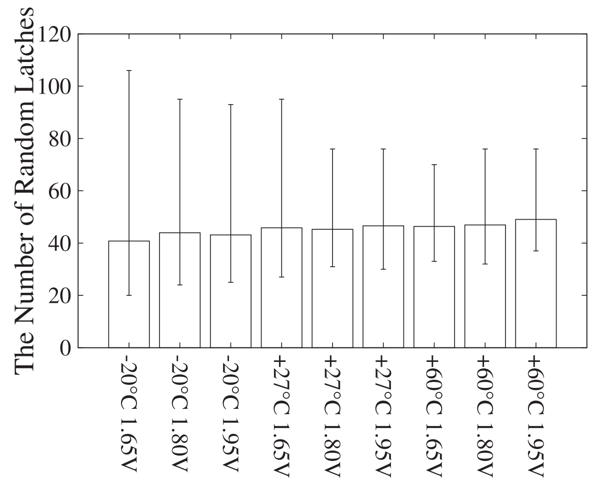

Fig. 11 Number of random latches (ML). The *bar graphs* and the *upper* and *lower lines* show the average, maximum, and minimum of the number of random latches in the nine cases of temperatures and voltages, respectively

of TRNGs, except in some cases. Even at −20 °C and 1.65 V, the number of random latches reached a minimum of 20 for the ML-TRNGs. In addition, we observed that some constant latches changed to random latches when the environment changed.

The number of random latches for the MN-TRNGs, whose average was approximately 42, was more stable than for the ML-TRNGs, whose average was approximately 45.

5.3.2 The quality of random latches

Figures 12 and 13 show the quality of random latches in MN-TRNGs and ML-TRNGs, respectively. Here, [a,b] and (a,b) represent closed and open intervals, respectively. For example, [30 %, 40 %) represent 30 % $\leq x <$ 40 %, where x is the proportion of 1's in the output sequence. We considered the quality of random numbers to be the same between 0–10 % and 90–100 %. Thus, the quality of the output sequence was divided into five intervals. The lowest part of the bar graph indicating [40 %, 60 %] represents random latches outputting high-quality random numbers. Approximately 5 % of random latches were of high-quality in any environment. In addition, we observed that a high-quality random latch in a temperature and a voltage became low quality one in other temperatures and voltages. Hence, high-quality random latches was changed when the environment changed.

5.3.3 Discussion of our TRNG

In any environment, there were approximately 42 random latches, and approximately 5 % of all random latches were high-quality random latches. Hence, there is expected to be about two high-quality random latches in any environment. From the experimental result in [2], we expect that the TRNG including one high-quality random latch generated high-quality random numbers. Therefore, our TRNGs can generate high-quality random numbers. Moreover, the number of random latches including biased ones is expected to contribute to improving the quality of random numbers produced by XOR operation [27].

6 Discussion

6.1 Randomness by increasing the number of latches

We expected that the quality of random numbers would be improved if the number of implemented SR latches increases. This is because our TRNGs generated random numbers as the XOR of 256 SR latch outputs. To verify this, we regarded the XOR of two actual TRNG outputs as random numbers obtained from a virtual TRNG with 512 built-in SR latches and evaluated whether or not the quality of the random numbers was improved. The virtual TRNGs were generated as follows. We focused on the TRNGs failing at least one test. If there were even numbers of TRNGs that failed the same test in the same environment, the XOR of each pair was regarded as the virtual TRNG. Otherwise, the XOR of outputs from the failing TRNGs and the TRNGs with the lowest min-entropy in the same test/environment was regarded as the virtual TRNG.

We evaluated the virtual TRNGs by using the AIS20/31 statistical tests. As a result, all the virtual PRNGs passed the test. Through this evaluation, we verified that the 256 latches were not sufficient for the ML-TRNGs to pass all AIS20/31 statistical tests, while the quality of the random numbers could be improved by increasing the number of SR latches.

6.2 Statistical tests and conditioning comportment

Table 2 shows the pass rates of the SP800-90B health test, AIS20/31 statistical tests, and SP800-90B IID test.

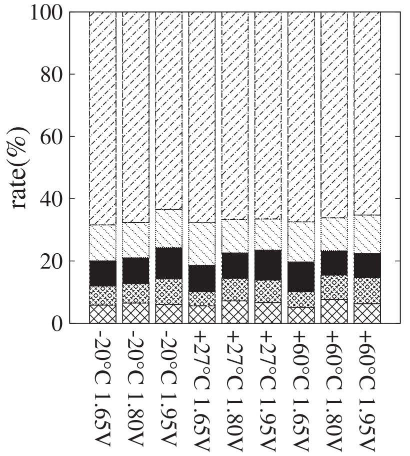

Fig. 12 Quality of random latches (MN). Approximately 5 % of the random latches were high-quality in the nine cases of temperatures and voltages

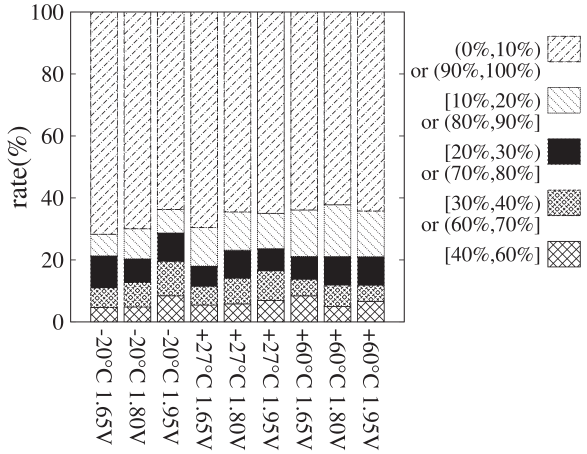

Fig. 13 Quality of random latches (ML). Approximately 5 % of the random latches were high-quality in the nine cases of temperatures and voltages

Table 2 Results of statistical tests

	SP800-90B	AIS20/31	SP800-90B
	Health test	Statistical test	IID test
MN-TRNG	100 %	100 %	100 %
ML-TRNG	100 %	99 %	97 %

The pass rates of SP800-90B health test, AIS20/31 statistical test, and SP800-90B IID test are shown. The pass rate based on the number of cases: the pairs of voltages and temperatures. The chips of MN-TRNG passed the all tests (180 cases = 20 chips × 9 cases), and those of ML-TRNG passed all tests except two cases of AIS20/31 statistical tests and five cases of SP800-90B out of 171 cases (19 chips × 9 cases). We consider that all MN-TRNGs and ML-TRNGs generate high-quality random numbers

The pass rate was based on the number of cases involving pairs of voltages and temperatures. MN-TRNG chips passed all of the tests (180 cases = 20 chips × cases), and the ML-TRNG chips passed all of the tests except for two cases of the AIS20/31 statistical tests and five cases of the SP800-90B IID test out of 171 cases (19 chips × 9 cases).

From these results, we found that all MN-TRNGs generated high-quality random numbers. We observed that our MN-TRNGs generated unbiased random sequences that had enough entropy. Therefore, their output random sequences can be used without the conditioning component in Section 3.1. We believe that the ML-TRNGs also generated high-quality random numbers. However, the quality of the numbers is expected to be improved by increasing the number of latches per chip or by connecting conditioning components, such as a linear-feedback shift register (LFSR) [28] and other conditioning components shown in AIS20/31 and SP800-90B.

6.3 Comparison

Table 3 shows a performance comparison of recent proposed TRNGs that were fabricated on a CMOS ASIC. In the table, it is difficult to compare the power consumption and the throughput because the CMOS processes are different. In addition, the evaluation methods of randomness are different. Therefore, Table 3 shows state-of-the-art TRNGs for reference. We believe that our TRNGs are well balanced from the viewpoint of power and throughput and are suitable for embedded systems.

7 Conclusions

In this paper, we fabricated two types of TRNGs using SR latches on ASICs and evaluated the robustness of the TRNGs against temperature and voltage fluctuations.

We validated that the TRNGs can generate random numbers at a standard voltage and room temperature. Furthermore, we evaluated that the random numbers were generated in various conditions, where the temperature was between −20 and 60 °C and the voltage was between 1.65 and 1.95 V, in line with the AIS20/31 statistical tests [3], SP800-90B health tests, IID verification tests, and min-entropy estimation [4].

As a result, we found that all of the MN-TRNGs (the TRNGs on CS86MN with a standard power consumption) generated high-quality random numbers that pass all of the abovementioned tests in various environments. Our TRNGs also generated high-quality random numbers continually because the min-entropy is stable at high values. Some of the ML-TRNGs (the TRNG on CS86ML with low power consumption) failed some of the AIS20/31 statistical tests and IID evaluation tests. However, the quality of random numbers is expected to be improved by increasing the number of SR latches or connecting an appropriate conditioning component.

For these reasons, our TRNGs that use SR latches on an ASIC have robustness against temperature and voltage fluctuations. The circuit scale and power consumption of the TRNGs were 1240.5 gates and 270 μW, respectively. Hence, our TRNGs were small in size and had a low power consumption, which is suitable for embedded devices.

We have come to the conclusion that our TRNGs as an entropy source can be used, for example, generating cryptographic keys, nonces for authentication, and seeds for pseudo random number generators.

Table 3 Comparison with previously proposed TRNG. The performances comparison of recent proposed TRNGs were fabricated on CMOS ASIC. It is difficult to compare the power consumption and the throughput fairly because the CMOS processes are different. In addition, the evaluations of randomness are different. Therefore, this shows state-of-the-art TRNGs for reference. We think that our TRNG is good balanced TRNG from the viewpoint of the power and the throughput

Entropy	Reference	Technology	Power	Throughput	Post processing	Random number evaluation
Direct amplification	[8]	0.18 μm	3.6 mW	5 Mbps	–	FIPS140-1, Knuth 2nd ed.
Jitter oscillator sampling	[12]	90 nm	240 μW	1.74 Mbps	LSFR	AIS31, Entopy dist.
	[13]	65 nm	–	7.5 Mbps	None	SP800-22Basic, DIEHARD
Metastable circuit	[18]	0.13 μm	1 mW	40 Kbps	5:1 decimation	SP800-22 Basic
	[19]	0.35 μm	9.4 μW	5 Kbps	NSFR	SP800-22 Basic
	[21]	45 nm	7 mW	2.4 Gbps	–	SP800-22
	This work(MN)	0.18 μm	270 μW	2.5 Mbps	–	SP800-90B, AIS20/31
	This work(ML)	0.18 μm	252 μW	2.5 Mbps	–	SP800-90B, AIS20/31

Future work will include considering hardware size and power consumption including the conditioning component, the health circuit component, and I/O interface, discussion on the experiment in terms of larger fluctuations of temperature, voltage, and clock frequency. In addition, we will evaluate a version of our TRNGs that has been fabricated by advanced LSI processes.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

NT carried out the evaluations of the implemented random number generators, participated in the analyzing the experimental results and drafted the manuscript. HK carried out the collection of data, implemented a statistical test program in accordance with SP800-90B, participated in analyzing the results of the program, and drafted the manuscript. DY participated in the experiment design, carried out the collection of data, participated in the analyzing the experimental results, and helped to draft the manuscript. KI implemented a statistical test program in accordance with AIS20/30 and participated in analyzing the results of the program. MT conceived of the study and participated in its design and coordination. TM participated in the evaluation of the experimental results and helped to draft the manuscript. All authors read and approved the final manuscript.

Acknowledgements

The authors would like to thank anonymous referees for their constructive feedback and useful comments.

Received: 31 July 2015 Accepted: 4 May 2016
Published online: 16 May 2016

References

1. NIST, Special Publication 800-22, A statistical test suite for random and pseudorandom number generators for cryptographic applications (2010). http://csrc.nist.gov/publications/nistpubs/800-22-rev1a/SP800-22rev1a.pdf. Accessed 11 May 2016
2. H Hata, S Ichikawa, FPGA implementation of metastability-based true random number generator. IEICE Trans. Inf. Syst. **E95-D**(2), 426–436 (2012). doi:10.1587/transinf.E95.D.426
3. W Killmann, W Schindler, A proposal for: functionality classes for random number generators version 2.0 (2011). https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Zertifizierung/Interpretationen/AIS_20_Functionality_classes_for_random_number_generators_e.html. Accessed 11 May 2016
4. NIST, Special Publication 800-90B(first draft), *Recommendation for the entropy sources used for random bit generation.* (National Institute of Standards and Technology, Gaithersburg, MD, 2012). http://csrc.nist.gov/publications/drafts/800-90/draft-sp800-90b.pdf. Accessed 11 May 2016
5. NIST, *Special Publication 800-90B (second draft), Recommendation for the entropy sources used for random bit generation.* (National Institute of Standards and Technology, Gaithersburg, MD, 2016). http://csrc.nist.gov/publications/drafts/800-90/sp800-90b_second_draft.pdf. Accessed 11 May 2016
6. H Kokubo, D Yamamoto, M Takenaka, K Itoh, N Torii, *Evaluation of ASIC Implementation of Physical Random Number Generators Using RS latches.* (Springer International Publishing, Switzerland, 2014), pp. 3–15. doi:10.1007/978-3-319-08302-5_1
7. WT Holman, JA Connelly, AB Dowlatabadi, An integrated analog/digital random noise source. IEEE Trans. Circ. Syst. I Fundam. Theory Appl. **44**(6), 521–528 (1997). doi:10.1109/81.586025
8. M Bucci, L Germani, R Luzzi, A Trifiletti, M Varanonuovo, A high-speed oscillator-based truly random number source for cryptographic applications on a smart card IC. IEEE Trans. Comput. **52**(4), 403–409 (2003). doi:10.1109/TC.2003.1190581
9. RC Fairfield, RL Mortenson, KB Coulthart, in *Advances in Cryptology, Proc. of CRYPTO 84, LNCS Vol.195*. An LSI Random Number Generator (RNG) (Springer, Heidelberg, 1985), pp. 203–230. doi:10.1007/3-540-39568-7_18
10. V Fischer, M Drutarovsk, in *Cryptographic Hardware and Embedded Systems-CHES2002, LNCS Vol.2523*. True Random Number Generator Embedded in Reconfigurable Hardware (Springer, Berlin, Heidelberg, 2002), pp. 415–430. doi:10.1007/3-540-36400-5_30
11. KH Tsoi, KH Leung, PH Leong, PHW Leong, in *11th Annual IEEE Symposium on Field-Programmable Custom Computing Machines-FCCM 2003(April 2003)*. Compact FPGA-based true and pseudo random number generators (IEEE, New York, 2003), pp. 51–61. doi:10.1109/FPGA.2003.1227241
12. M Bucci, R Luzzi, Fully digital random bit generators for cryptographic applications. IEEE Trans. Circ. Syst. I Reg. Papers. **55**(3), 861–875 (2008). doi:10.1109/TCSI.2008.916446
13. T Amaki, M Hashimoto, T Onoye, A process and temperature tolerant oscillator-based true random number generator. IEICE Trans. Fundam. **E97-A**(12), 2393–2399 (2014)
14. B Sunar, WJ Martin, DR Stinson, A provably secure true random number generator with built-in tolerance to active attacks. IEEE Trans. Comput. **56**(1), 109–119 (2007). doi:10.1109/TC.2007.250627
15. D Schellekens, B Preneel, I Verbauwhede, in *Proceedings - 2006 International Conference on Field Programmable Logic and Applications (August 2006)*. FPGA vendor agnostic true random number generator (IEEE, New York, 2006), pp. 1–6. doi:10.1109/FPL.2006.311206
16. K Wold, CH Tan, Analysis and enhancement of random number generator in FPGA based on oscillator rings. Int. J. Reconfigurable Comput. **2009**(501672), 8 (2009). doi:10.1155/2009/501672
17. DJ Kinniment, EG Chester, in *Proceedings of the 28th European Solid-State Circuits Conference (September 2002)*. Design of an on-chip random number generator using metastability (IEEE, New York, 2002), pp. 595–598. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1471597. Accessed 11 May 2016
18. C Tokunaga, D Blaauw, T Mudge, in *2007 IEEE International Solid-State Circuits Conference(Feburary 2007). Digest of Technical Papers*. True Random Number Generator with a Metastability-Based Quality Control (IEEE, New York, 2007), pp. 404–406. doi:10.1109/ISSCC.2007.373465
19. J Holleman, S Bridges, BP Otis, C Diorio, A 3 μW CMOS true random number generator with adaptive floating-gate offset cancellation. IEEE J. Solid State Circ. **43**(5), 1324–1336 (2008). doi:10.1109/JSSC.2008.920327
20. M Varchola, M Drutarovsky, in *Cryptographic Hardware and Embedded Systems - CHES 2010. LNCS Vol.6225*. New High Entropy Element for FPGA Based True Random Number Generators (Springer, Heidelberg, 2010), pp. 351-365. doi:10.1007/978-3-642-15031-9_24
21. SK Mathew, S Srinivasan, MA Anders, H Kaul, SK Hsu, F Sheikh, RK Krishnamurthy, 2.4 Gbps, 7 mW all-digital PVT-variation tolerant true random number generator for 45 nm CMOS High-Performance Microprocessors. IEEE J. Solid State Circ. **47**(11), 2807–2821 (2012). IEEE. doi: 10.1109/JSSC.2012.2217631
22. P Haddad, V Fischer, F Bernard, J Nicolai, in *Cryptographic Hardware and Embedded Systems – CHES 2015, LNCS Vol.9293*. A physical approach for stochastic modeling of TERO-based TRNG (Springer, Heidelberg, 2015), pp. 357–372. doi:10.1007/978-3-662-48324-4
23. Fujitsu Semiconductor, Semicustom CMOS standard cell CS86 series (2009). http://www.fujitsu.com/us/Images/e620209_CS86_ASIC.pdf. Accessed 28 Mar 2016
24. K Finkenzeller, *RFID Handbook: Fundamentals and, Applications in Contactless Smart Cards and Identification, Second Edition*. (Wiley, West Sussex, 2003)
25. Xilinx, Spartan-3E starter kit board user guide. http://www.xilinx.com/support/documentation/boards_and_kits/ug230.pdf. Accessed 28 Mar 2016
26. NIST, NIST statistical test suite (2014). http://csrc.nist.gov/groups/ST/toolkit/rng/documents/sts-2.1.2.zip. Accessed 28 Mar 2016
27. RB Davies, Exclusive OR (XOR) and hardware random number generators Independent biased pairs (2002). http://www.robertnz.net/pdf/xor2.pdf. Accessed 28 Mar 2016
28. R Ward, T Molteno, Table of linear feedback shift registers.Technical Reports, University of Otago, New Zealand (2007). http://courses.cse.tamu.edu/csce680/walker/lfsr_table.pdf. Accessed 28 Mar 2016