Evaluation of ASIC Implementation of Physical Random Number Generators using RS Latches

Hirotaka Kokubo, Dai Yamamoto, Masahiko Takenaka,
Kouichi Itoh, and Naoya Torii

Fujitsu Laboratories Ltd., Secure Computing Lab,
4-1-1 Kamikodanaka, Nakahara-ku, Kawasaki, Kanagawa 211-8588, Japan
{kokubo.hirotaka,yamamoto.dai,
ma,ito.kouichi,torii.naoya}@jp.fujitsu.com

Abstract. Embedded devices such as smart cards and smart phones are used for secure systems, for example automated banking machines and electronic money. The security of an embedded device depends strongly on secret information; cryptographic keys, nonces for authentication or seeds for a pseudo random number generator, which is generated by a Physical True Random Number Generator (PTRNG). If a PTRNG generates random numbers with a low entropy, the security of the embedded device has a vulnerability because secret information may be predictable by attackers due to the low entropy. Hence PTRNGs are required to provide high-quality physical random numbers even in an undesirable environment, that is, low/high temperature or supply voltage. PTRNGs also must be small-scale and consume low power due to the limited hardware resources in embedded devices.
In this paper, we fabricate and evaluate 39 PTRNGs using RS Latches on 0.18μm ASICs. Physical random numbers were generated from the exclusive-OR of 256 RS latches' outputs. Our PTRNGs passed the SP800-90B Health Tests and the AIS31 Tests while changing both temperature (from -20°C to 60°C) and voltage (1.80V ±10%), and thus, we were able to confirm that our PTRNGs have high-robustness against environmental stress. The power consumption and circuit scale of our PTRNG are 0.27mW and 984.5 gates, respectively. Our PTRNG using RS latches is small enough to be implemented on embedded devices.

Keywords: Random Number Generator,RS Latch, Metastability, AIS31, SP800-90B

1 Introduction

Embedded devices such as smart cards and smart phones have become widespread in applications where high security is necessary, such as employee ID cards, electronic money and online banking. These embedded devices have cryptographic hardware for secure communications and identification/authentication. Cryptographic hardware achieves high-level security by using cryptographic technologies such as symmetric-key cryptography and a pseudo random number generator. One of the security aspects for these cryptographic technologies depends

on random numbers. This is because the random numbers are used for key generations for symmetric-key/public-key ciphers and seed generations for pseudo random number generators amongst other things. Random numbers with a low randomness cause the risk of prediction of the secret key and seed, which enables attackers to eavesdrop on communication contents and forge signatures. Hence, the quality of random numbers affects the security of embedded devices. Generally, random numbers are generated with physical random number generators (PTRNGs). Embedded devices with high-level security require PTRNGs which can generate high-quality random numbers. Additionally, embedded devices such as smart card and smart phone are often exposed to environmental changes, so attackers could intentionally lower the quality of the random numbers by freezing embedded devices. Therefore, PTRNGs should be able to generate high-quality random numbers regardless of the environmental changes. Moreover, PTRNGs should be able to integrate as an Large Scale Integration (LSI) for resource-limited embedded devices.

Some of the PTRNGs that can be integrated as digital LSI have been previously proposed, but there are many problems in terms of noise, power consumption, circuit scale and design cost. A PTRNG using RS latches has been proposed as a method to solve these problems. This PTRNG has been implemented only on FPGAs. Application specific integrated circuit (ASIC) implementation is necessary for the mass production of the PTRNGs because ASIC has the advantage of lower chip cost, lower power consumption and faster processing than FPGA. It is unknown whether or not a PTRNG on ASIC is able to generate high-quality random numbers. It is necessary to implement and evaluate the PTRNG on ASIC because random numbers are affected by the characteristics of the semiconductor, but as yet no evaluation has been made of such a PTRNG on ASIC and PTRNGs [1] have only been evaluated with the NIST SP800-22 randomness statistical tests[2]. It has not been evaluated by the tests dedicated to physical random numbers, namely AIS31[3] and SP800-90B[4]. PTRNGs should be evaluated by these tests because the importance of PTRNGs has recently been gathering attention, and these tests for physical random numbers will be widely used in the future. Moreover, the robustness of PTRNG against temperature and voltage fluctuations must be evaluated.

Our Contributions In this paper, we implement a PTRNG using RS latches on an ASIC based on the PTRNG on an FPGA[1]. The reason why we focus on this latch-based PTRNG is that its design cost is small and high-quality random numbers are expected to be generated in any environment. This paper makes four contributions; (1) We fabricated the PTRNG on a 0.18 μm CMOS ASIC. We evaluated whether or not the PTRNG is able to generate random numbers on this ASIC. (2) We measured the power consumption and the circuit scale of the PTRNGs, and examined whether it can be installed in embedded devices. (3) We evaluated the quality of random numbers generated by our PTRNGs according to the AIS31 and SP800-90B randomness statistical tests for physical random numbers. (4) We examined whether our PTRNGs have the robustness against temperature and voltage fluctuations. As a result, our PTRNGs on an

ASIC were found to be small and low-power enough to be implemented on embedded devices, and able to generate high-quality random numbers even if the environment changes, thus our PTRNGs can improve the security of embedded devices.

Organization of This Paper This paper is organized as follows: Section 2 briefly introduces some work related to our research. Section 3 gives an outline of a PTRNG using RS Latches. Section 4 describes an ASIC implementation of the PTRNG. In addition, we measured the power consumption of the PTRNG on an ASIC. Section 5 evaluates the quality of the physical random numbers from the PTRNG by using the AIS31 and SP800-90B Health Tests. Finally, Section 6 gives a summary of this research.

2 Related Work

Figure 1 shows various PTRNGs on LSIs which have been proposed until now. The PTRNGs are classified into two types; analog-based one and digital-based one. Analog-based PTRNGs are based on random noise signals such as thermal noise, and they are known to be high-quality random number generators. However, the weak point of these PTRNGs is that they are difficult to integrate in high-density in an LSI due to the large-scale thermal sensors. Digital-based PTRNGs are categorized by entropy sources. One is to use the jitter of oscillators as an entropy source, for example ring oscillators-based PTRNGs [5]. A ring oscillator has a feedback structure composed of an odd number of NOT gates. Random numbers are obtained from the exclusive-OR of multiple ring oscillator outputs, and they have the robustness against temperature change. However, the PTRNGs in this category would be not suitable for embedded devices with limited resources because the ring oscillator has large power consumption, noise, and circuit scale. The other is to use the metastability of digital circuits. This type of PTRNG is suitable for embedded devices because of the small scale and low-power consumption. The prototypes of this PTRNG can generate high-quality random numbers [6][7][8]. However they need an additional dynamic adjustment for the voltage or of internal elements. This adjustment needs a dedicated full-custom circuit, which causes the large design cost at the transistor level. Moreover, it is necessary to re-design them when implementing on different CMOS technology because the PTRNGs often do not work as expected under a different CMOS technology.

Hata et al. have proposed a PTRNG using the metastability of RS latches and implemented it on an FPGA [1]. The design cost of this PTRNG is quite small because it uses only digital synchronous circuits. In addition, the PTNRG can save power consumption by stopping the clock signal inputted to the RS latches when the random numbers generation is not required. The random numbers from the PTRNG passes the NIST SP800-22 statistical tests[2]. For the abovementioned reasons, the PTRNG proposed by Hata et.al. has better properties for embedded devices than other PTRNGs.

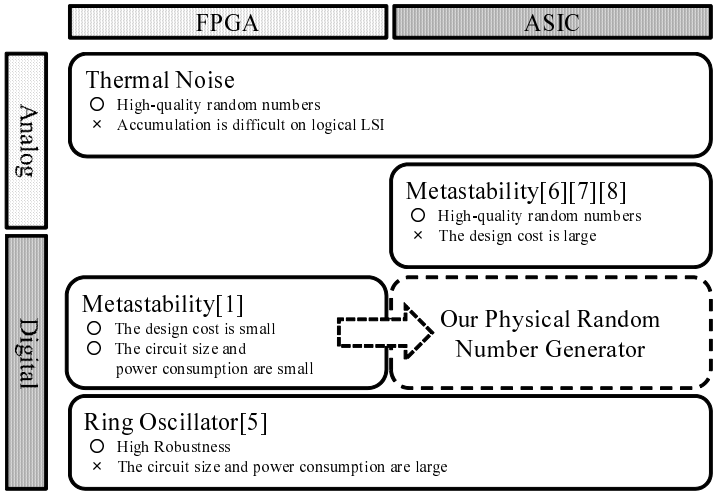

Fig. 1. Variety of Physical Random Number Generators.

3 Random Number Generator using RS Latches

This section explains the method for generating physical random numbers that was proposed by Hata et al. in [1]. The PTRNGs that are using this method generates physical random numbers based on the metastability of RS latches.

Figure 2 shows an RS latch. An RS latch consists of 2 NAND gates, and is commonly used to store one bit information. When $input = 0$, the RS latch is stable with $output = 1$. When $input$ changes from 0 to 1, the RS latch temporarily enters a metastable state, and then, it is stable with $output = 0$ or 1. Physical random numbers can be obtained from $output$ by giving $input$ clock signals using this behavior. Ideally, the probability of outputting 0 and 1 is equal, but this probability is actually biased. This is because of the difference in wiring delay between gates, or the difference of drive capability between two NAND gates. In many cases, this RS latch generates only '0's or only '1's, so it is difficult to generate high-quality random numbers using only one RS latch. A PTRNG consisting of multiple RS latches and an exclusive-OR gate is proposed in [1]. This PTRNG generates random numbers from the exclusive-OR of multiple RS latches' outputs. This enables the PTRNG to exclude the biases and to generate high-quality random numbers.

Problems There are two problems in [1]. (1) This PTRNG has implemented only on FPGAs. (2) This PTRNG has not been evaluated in various environments. 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. The PTRNGs for embedded devices must be able to generate high-quality physical random numbers in any environments. If the PTRNG generates random numbers with low entropy due to environmental changes, the security of the embedded device is compromised

because secret information may be predictable by attackers due to the low entropy. 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 PTRNGs is affected by the both changes. Hence, the robustness against to the changes should be evaluated, but as yet it has not. In addition, the PTRNGs should be evaluated based on the SP800-90B Health Tests published in 2012, which is introduced for the tests of physical random number generators.

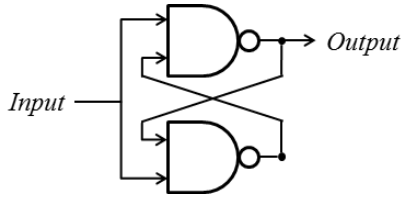

Fig. 2. RS Latch

4 ASIC Implementation

We fabricate PTRNGs using RS latches on a 0.18μm CMOS ASIC (Fujitsu CS86 series [9]). This PTRNG generates random numbers from the exclusive-OR of 256 RS latches' outputs. The RS latch was custom-designed on the circuit layout so that the wire lengths between the two NAND gates are the same, and was implemented as hard macro. Thus, the probability of the RS latch generating random numbers is expected to improve. 256 RS latches are implemented automatically by using circuit design tools. Hence, the design cost is quite small. The PTRNGs are assembled as DIP28 packages. Two types of the PTRNG were fabricated, namely 20 standard PTRNGs (using CS86MN, called MN-PTRNG) and 19 low-power-consuming PTRNGs (using CS86ML, called ML-PTRNG).

4.1 Measurement of Power Consumption and Circuit Scale

Embedded devices require low-power-consuming PTRNGs. We measured the power and current consumption of the PTRNGs with a direct current ammeter. According to our experimental measurements, the average power/current consumption of both MN-PTRNG and ML-PTRNG ASICs is 0.27mW/0.15mA and 0.252mW/0.14mA respectively. The current consumption of common ASICs used for contactless smart cards is approximately 1mA [10]. The current consumption of our PTRNG was much smaller than this value, so is practical and useful. Additionally, we measured the circuit scale of our PTRNG. In the following discussion, one gate is equivalent to a 2-1 NAND gate (2-bit input and 1-bit

output). The PTRNG consists of 256 RS latches, a 256-1 exclusive-OR gate, and a 1-bit flip-flop to store a random number temporarily. Our PTRNG was synthesized with the Design Compiler 2003.03, and the circuit scale was 984.3 gates. This circuit scale was smaller than the implementation of the PRESENT cipher which is one of the most famous ultra-lightweight ciphers [11]. In addition, this circuit size is smaller than the circuit size of Triple DES which is one of the most widely used in smart cards (e.g. MIFARE DESFire MF31CD40). We achieved PTRNGs with the very small circuit scale on an ASIC.

5 Evaluation

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

5.1 Evaluation System

Figure 3 shows our experimental system for the acquisition of random numbers. This figure is omitted excluding important parts. It consists of two boards: a custom-made board for the ASICs of the PTRNGs and a Spartan-3E starter kit board with a Xilinx FPGA for controlling the PTRNGs[12]. The core voltage to the PTRNGs was supplied by using a stabilizing power supply, which was able to adjust the supply voltage at intervals of 0.01V. The clock signals were input to the PTRNGs through the FPGA board. Random numbers generated by the PTRNGs were written to a micro SD card via a block RAM of the FPGA. We acquired not only the random numbers but also the output of each latch for our further evaluation.

In this environment, we evaluated the random numbers generated by all of the 39 PTRNGs while changing the temperature and voltage. The core voltage is changed to 1.65V (1.80V–10%), 1.80V (standard) and 1.95V (1.80V+10%) by the stabilizing power supply. The temperature was maintained at -20°C, 27°C, and 60°C by using a constant temperature oven. Only the custom-made board for the PTRNGs 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

We acquired approximately 5.5M bits of random numbers from each PTRNG while changing the temperature and voltage. 351 cases of random numbers (3 temperatures× 3 voltages×39 PTRNGs, 180 cases for MN-PTRNGs and 171 cases for ML-PTRNGs) was exhaustively evaluated according to both the SP800-90B Health Tests and the AIS31 Tests.

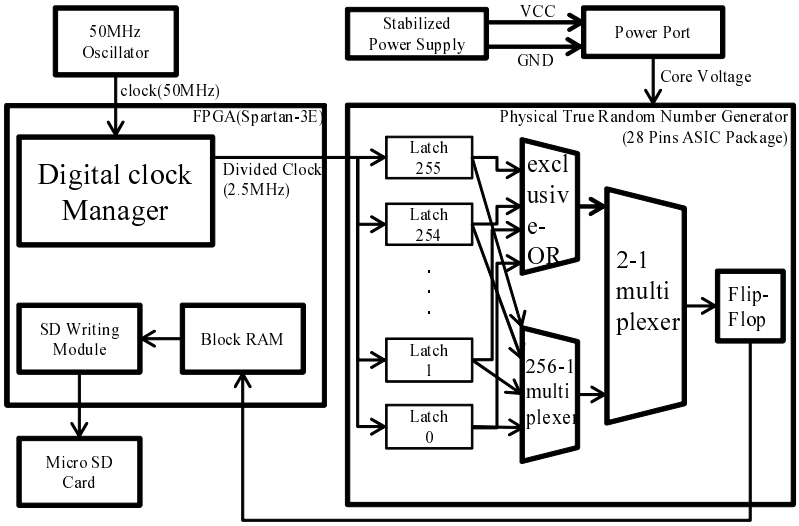

Fig. 3. Experimental System for the Acquisition of Random Numbers

The NIST SP800-22 statistical tests[2], which are well known as tests for *pseudo* random numbers, had been used for physical random numbers. However, there is SP800-90B and AIS31 which are tests dedicated to *physical* random numbers now. We evaluated our PTRNGs according not to SP800-22 but to these tests in this paper.

NIST SP800-90B Health Tests

We evaluated whether our PTRNGs could generate high-entropy random numbers according to the repetition count test and the adaptive proportion test defined in SP800-90B[4]. The random numbers at various temperatures and voltages were tested as follows. 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***]

If the same value (0 or 1) appears consecutively c times or more in the sequence of random numbers, the random numbers are a failure, where $c = ceiling(1 + 30/min\text{-}entropy)$. In this paper, c is 32. *min-entropy* will be mentioned in Section 5.3.

[***Adaptive Proportion Test***]

Firstly, we obtained a 1-bit value from the beginning of the random numbers as a reference value. Secondly, we obtained one *block* from the succeeding random numbers. The bit length of a block is represented by *window size*, and if the reference value appears greater than *cutoff* times in a block, the random numbers are failure. The size of the *cutoff* is defined by the *false positive rate*, *min-entropy* and *window size*. This procedure was repeated until the end of the

random numbers. In our evaluations, the *window size* and *cutoff* were 64, 51 in Test Settings I and are 4096, 2240 in Test Settings II, respectively. That is, about 84,700 blocks are evaluated in "Test Settings I" and about 1,350 blocks are evaluated in "Test Settings II", in each case of random numbers. We consider the PTRNGs pass the SP800-90B Health Tests if all blocks pass in both test settings. This means that the PTRNGs continuously generate random numbers with high-entropy.

Figure 4 and 5 show the rate of the PTRNGs that passed the SP800-90B Health Tests. The horizontal axis shows the environment at various temperatures and voltages. The vertical axis shows the rate of the PTRNGs that passed the tests. In the MN-PTRNGs, all cases pass this test as shown in Fig. 4. In the ML-PTRNGs, six cases failed the test in Fig. 5, and four cases out of the six happened when the temperature was -20°C. This may be because the ML-PTRNGs have a small number of RS latches outputting random numbers at a low temperature (details are discussed in Section 5.4). In contrast, the MN-PTRNGs can generate high-entropy random numbers even when the temperature and voltage change. Hence an MN-PTRNG is more suitable for generating physical random numbers than an ML-PTRNG.

AIS31 Tests

We evaluated the random numbers in various temperatures and voltages according to AIS31 Tests [3]. AIS31 is an evaluation criterion for the physical random number generators defined by BSI (i.e. the German Federal Office for Information Security). Tests in AIS31 include various statistical tests such as the Poker Test, the Long Run Test and the Uniform Distribution Test. AIS31 classifies PTRNGs into two classes; P1 Class and P2 Class. PTRNGs in P1 Class pass P1 Tests, and PTRNG in P2 Class pass P2 Tests. The PTRNGs in the P1 Class can be used for random number generation for challenge and response authentication. The PTRNGs in the P2 Class can be used for key and seed generations for pseudo random number generators, which provide higher security than PTRNGs in the P1 Class. It is desirable for PTRNGs to pass both of the tests because PTRNGs for embedded devices are used for various applications.

Figures 6 and 7 show the rate of PTNRGs that passed the AIS31 Tests. The horizontal and the vertical axises are the same as Fig. 4 and 5. If the PTRNG fails either of the P1 or P2 Tests, we regarded it as *failed* PTRNG. The MN-PTRNGs pass the tests in all cases as shown in Fig. 6, so our MN-PTRNGs have the robustness against temperature and voltage fluctuations, and can thus be used for secure embedded systems including key generation. The ML-PTRNGs, however, failed tests only in two cases out of 171, one of which was the same PTRNG as the failed PTRNG in the SP800-90B Health Tests. The next section discusses whether ML-PTRNGs are able to generate high-quality random numbers.

Further Evaluation by Increasing the Number of Latches

We expected that the quality of random numbers would be improved by in-

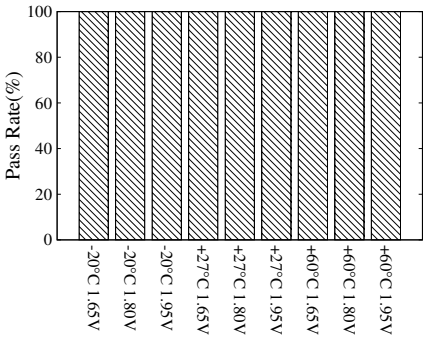

Fig. 4. SP800-90B Pass Rate (MN)

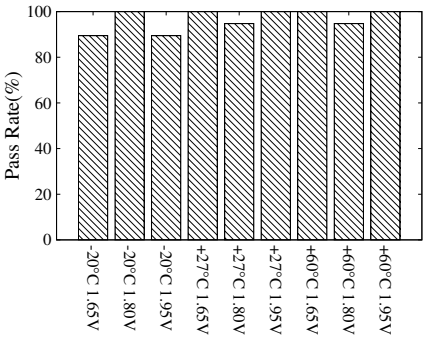

Fig. 5. SP800-90B Pass Rate (ML)

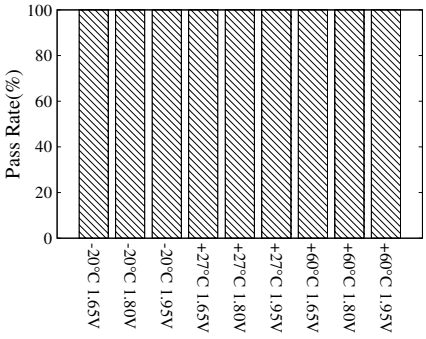

Fig. 6. AIS31 Pass Rate (MN)

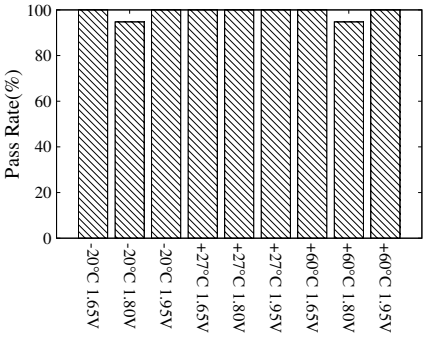

Fig. 7. AIS31 Pass Rate (ML)

creasing the number of implemented RS latches. This is because our PTRNGs generated random numbers as the exclusive-OR of 256 RS latch outputs. To verify this, we regarded the exclusive-OR of 2 actual PTRNGs outputs as random numbers obtained from a virtual PTRNG with built-in 512 RS latches, and evaluated whether or not the quality of the random numbers was improved. The virtual PTRNGs were generated as follows. We focused on the PTRNGs failing at least one test. If there were even numbers of PTRNGs that failed the same test in the same environment, the exclusive-OR of each pair was regarded as the virtual PTRNG. Otherwise, the exclusive-OR of outputs from the failing PTRNG and the PTRNG with the lowest min-entropy in the same test/environment was regarded as the virtual PTRNG.

We evaluated the virtual PTRNGs according to the NIST SP800-90B Health Tests and AIS31 Tests. As a result, all the virtual PTRNGs passed both tests. Through this evaluation, we verify that the 256 latches are not sufficient for the ML-PTRNGs, while the quality of random numbers could be improved by increasing the number of RS latches. Hence we should carefully decide the number of implemented RS latches in consideration of both the quality of random numbers and the circuit space.

5.3 Min-Entropy Estimation

We use *minimum entropy* (i.e. 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 the 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 PTRNG is IID (Independent and Identically Distributed). Therefore, first, we implemented a software for IID verification tests [4], and evaluated our PTRNGs using this tests. As a result, the MN-PTRNGs passed all 180 cases, so we performed min-entropy estimation for IID sources (see Section 9.2 in [4]). In contrast, the ML-PTRNGs passed 166 out of 171 cases. We, however, we regarded the ML-PTRNGs as IID sources because they passed at least all tests under the normal conditions (i.e. 27°C and 1.80V). Min-entropy estimation as a non-IID sources will be part of our future work.

Figures 8 and 9 show the results of min-entropy estimation. The middle line, 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 PTRNGs. Hence our PTRNGs have a very high min-entropy regardless of the temperature or voltage.

5.4 Evaluation of Output from each RS Latch

As mentioned previously, our PTRNGs, especially MN-PTRNGs, have the robustness against temperature and voltage fluctuations. This section evaluates the behavior of each RS latch in order to clarify the reason for the robustness. We defined a "random latch" as the RS latch whose output sequence includes

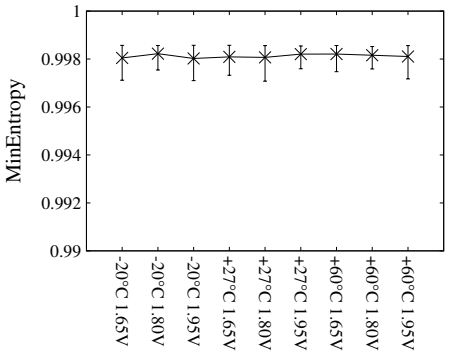

Fig. 8. MinEntropy of Physical Random Number Generator (MN)

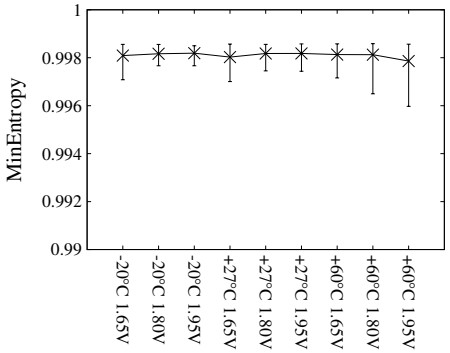

Fig. 9. MinEntropy of Physical Random Number Generator (ML)

at least one transition between 0 and 1, and defined a "constant latch" as the RS latch that generates only '0's or only '1's. We focus on two evaluation axes: the number of random latches and the quality of random numbers from each random latch.

The Number of Random Latches

We acquired 21K bits of output sequence from each RS latch while changing the temperature and voltage, and evaluated the number of random latches. In Fig. 10 and 11, the bar graphs, the upper and lower of lines show the average, maximum and minimum of the number of random latches in the test cases respectively. The higher temperature and voltage, the larger average number of random latches in both types of PTRNGs, except in some cases. At -20°C and 1.65V, the number of random latches reaches a minimum of 20 in ML-PTRNG. In addition, there are some transitions from constant latch to random latch whenever the environment changes. The number of random latches in an MN-PTRNGs, whose average is approximately 40, is more stable than in an ML-PTRNG.

The Quality of Random Numbers from each Random Latch

The quality of each random latch is one of the most important metrics for the quality of a random number as well as the number of random latches. Thus, we evaluated the quality of the output sequence from each random latch. We acquired approximately 21K bits of output from each RS latch and examined the proportion of '1's in the output sequence. For the ideal random latch, the proportion of '1' is 50%.

Figure 12 and 13 show the rate of the number of random latches by the proportion of '1's. Here, [a,b] and (a,b) represent closed and open intervals respectively. For example, [30%,40%) and (60%,70%] represents $30\% \leq x < 40\%$ and $60\% < x \leq 70\%$, where x is the proportion of '1's in the output. The lowest part of bar graph indicating [40%,60%] represents random latches outputting

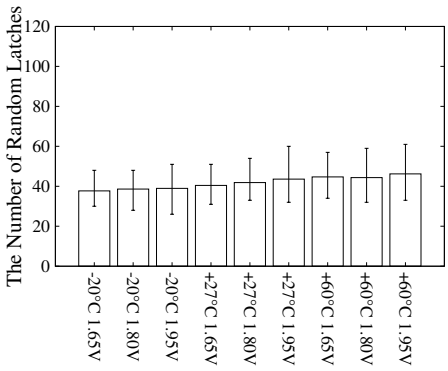

Fig. 10. Number of Random Latches (MN)

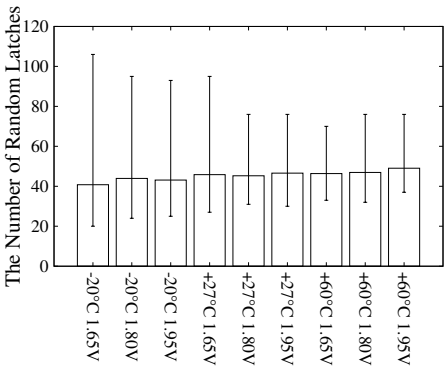

Fig. 11. Number of Random Latches (ML)

high-quality random numbers (i.e. "high-quality random latch"). There are approximately 5% high-quality random latches in any environment. Additionally, most random latches generate biased output.

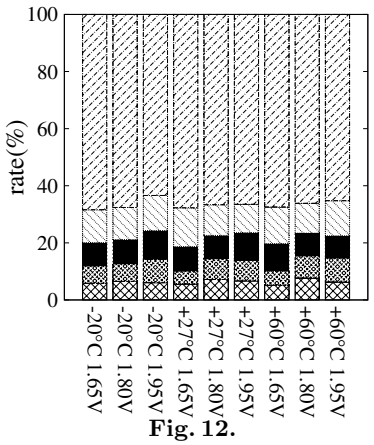

Fig. 12. The Characteristics of Random Latches (MN)

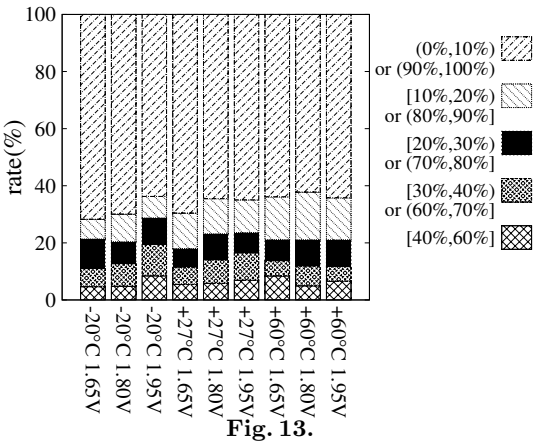

Fig. 13. The Characteristics of Random Latches (ML)

5.5 Discussion

We consider the results of Section 5.4. In any environment, there are approximately 40 random latches, and approximately 5% of all random latches are the high-quality random latch. Hence, there is expected to be about 2 high-quality random latches in any environment. Moreover, a number of random latches including biased ones are expected to contribute to improve the quality of random numbers.

As mentioned in Section 5.2, our PTRNGs can generate high-quality random numbers that pass the tests for physical random number generators in any environment. The min-entropy is stable at high level, and they can be used effective entropy source. We validate our PTRNGs using RS latches work stably and effectively on an ASIC. Circuit scale and power consumption of our PTRNGs are quite small. Our PTRNG generates high-quality random numbers in even worse conditions. Hence, our PTRNG is very suitable for embedded devices.

6 Conclusion and Future Work

In this paper, we fabricated 2 types of the PTRNGs using RS latches on ASICs, and evaluated the robustness of the PTRNGs against temperature and voltage fluctuations. We validated that the PTRNG can generate random numbers at a standard voltage and room temperature. Furthermore, we evaluated the random numbers generated in various conditions, where the temperature was between -20°C and 60°C and the voltage was between 1.65V and 1.95V, in line with the AIS31 Tests[3], SP800-90B Health Tests[4], IID Verification Tests[4] and Min-Entropy Estimation[4]. As a result, we found that all MN-PTRNGs (the PTRNG on CS86MNwith a standard power consumption) generates high-quality random numbers which pass all of the above-mentioned tests in various environments. Our PTRNGs also generated high-quality random numbers continually because the min-entropy is stable at high values. Some of the ML-PTRNGs (the PTRNG on CS86MLwith low power consumption) failed some tests, but the quality of random numbers, however, is expected to be improved by increasing the number of RS latches implemented. For these reasons, our PTRNGs that use RS latches on an ASIC have the robustness against temperature and voltage fluctuations. The circuit scale and the power consumption of the PTRNGs were 984.3 gates and $0.27m$W respectively. Hence our PTRNGs were small-size and had a low power consumption, which is suitable for embedded devices. Our PTRNGs are high-quality entropy sources and can be used for various purposes such as cryptographic keys, nonces for authentication and seeds for pseudo random number generators. Future work will include discussion on the experiment in larger fluctuations of temperature, voltage and clock-frequency and countermeasure against side-channel attacks.

References

1. H.Hata, S.Ichikawa, FPGA Implementation of Metastability-Based True Random Number Generator, IEICE Transactions on Information and Systems, vol.E95-D, no.2, pp.426-436, 2012.
2. NIST, Special Publication 800-22, A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications, 2010.
3. BSI, AIS31, Functionality classes and evaluation methodology for true (physical) random number generators, 2001.
4. NIST, Special Publication 800-90B, Recommendation for the Entropy Sources Used for Random Bit Generation, 2012.

5. B.Sunar, W.J.Martin, and D.R.Stinson, A provably secure ture random number generator with built-in tolerance to active attacks, IEEE Transactions on Computers, vol.56, no.1, pp.109-119, 2007.
6. M.Bellido, A.Acosta, M.Valencia, A.Barriga, and J.Huertas, Simple binary random number generator, Electronics Letters, vol.28, no.7, pp.617-618, 1992.
7. D.kinniment, and E.Chester, Design of an on-chip random number generator using metastability, Proc.ESSCIRC 2002, vol.4, no.6, pp.595-598, 2002.
8. C.Tokunaga, D.Blaauw, and T.Mudge, True random number generator with a metastability-based quality control, IEEE Journal of Solid-State Circuits, vol.43, no.1 pp.78-84, 2008.
9. Fujitsu Semiconductor, Semicustom CMOS Standard Cell CS86 Series, `http://edevice.fujitsu.com/fj/DATASHEET/e-ds/e620209.pdf`, 2011.
10. Klaus Finkenzeller, RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification, Second Edition, Wiley, 2003.
11. A.Bogdanov et al., PRESENT: An Ultra-Lightweight Block Cipher, CHES 2007 LNCS, vol.4727, pp.450-466, 2007.
12. Xilinx: Spartan-3E starter kit board, `http://www.xilinx.com/products/boards-and-kits/HW-SPAR3E-SK-US-G.htm`