Received: 9 March 2020 | Revised: 8 June 2020 | Accepted: 2 July 2020

DOI: 10.4218/etrij.2020-0083

SPECIAL ISSUE

ETRI Journal WILEY

Efficient hardware implementation and analysis of true random-number generator based on beta source

Seongmo Park[1] | Byoung Gun Choi[1] | Taewook Kang[1] | Kyunghwan Park[1] | Youngsu Kwon[1] | Jongbum Kim[2]

[1]Department of Intelligent SoC Research, Electronics and Telecommunications Research Institute, Daejeon, Rep. of Korea

[2]Department of Radioisotope Research, Korea Atomic Energy Research Institute, Daejeon, Rep. of Korea

Correspondence
Seongmo Park, Department of Intelligent SoC Research, Electronics and Telecommunications Research Institute, Daejeon, Rep. of Korea.
Email: smpark@etri.re.kr

Funding information
This work was supported by the Nuclear Research and Development Project grant funded by the Korean government [20JS1100, Random number generation circuit development and integration technology development].

This paper presents an efficient hardware random-number generator based on a beta source. The proposed generator counts the values of "0" and "1" and provides a method to distinguish between pseudo-random and true random numbers by comparing them using simple cumulative operations. The random-number generator produces labeled data indicating whether the count value is a pseudo- or true random number according to its bit value based on the generated labeling data. The proposed method is verified using a system based on Verilog RTL coding and LabVIEW for hardware implementation. The generated random numbers were tested according to the NIST SP 800-22 and SP 800-90B standards, and they satisfied the test items specified in the standard. Furthermore, the hardware is efficient and can be used for security, artificial intelligence, and Internet of Things applications in real time.

KEYWORDS
beta source, NIST SP800-22, NIST SP800-90B, RTL, true random number

1 | INTRODUCTION

Random-number encryption is used for information security in various fields such as electronic commerce and authentication. In addition, random numbers are used in various applications such as games, lotteries, sampling, and simulations [1]. Various methods for securing random numbers have been studied.

In practice, random-number generation is implemented artificially in the software as a pseudo-random-number generation technique. When generating a pseudo-random number, a device enters a seed, which is a series of bits, as an input to the deterministic random-number generator. The stability of the random number depends on the seed. Examples of seeds include a keyboard input by a user. Noises provided by the operating system or noises due to memory access contention among the graphical processing unit cores can also be used as the seed. The existence of the seed renders the "random" numbers from the generator not completely random.

Frequently used pseudo-random numbers are generated with computer algorithms, and highly complex computer algorithms have generated random numbers with increased complexity as well. Nevertheless, as pseudo-random numbers exhibit a unique pattern that depends on the algorithm, pseudo-random-number generators are vulnerable to hacking (eg, by prediction software). In fact, modern security technologies that use cryptographic system intelligence, such as Rivest-Shamir-Adleman, have a problem involving random pattern exposure [2]. These random number patterns can be extracted with the latest hacking techniques using quantum computers and artificial intelligence.

Consequently, technologies generating unpredictable, pure random numbers are being rapidly developed in areas

such as secure communications, security systems, and online banking. However, current true random-number generators are expensive, and researchers have been developing alternative, affordable true random-number generators. For example, true random numbers generated from hardware based on random physical phenomena are attracting attention. Random numbers can be extracted from quantum mechanical random phenomena, such as an emission of radioactive isotopes, alpha rays, and beta rays. Studies are being conducted to increase the randomness of numbers extracted from the random phenomena of hardware further.

Several tests have been developed to determine whether sequences were randomly generated. Here, we focus on a test class known as statistical tests for randomness [3]. AIS 20/31 [4], developed by the German Federal Office for Information Security, is a certification for entropy estimation that allows the designers of true random-number generators to estimate the theoretical entropy based on a provided probabilistic model of the entropy source. Another testing method is the NIST Special Publication (SP) 800-90B, published in 2016 as the second version [5].

Unlike AIS 20/31, NIST SP 800-90B is a statistical algorithm for estimation. Entropy is the final output value of AIS, whereas minimum entropy is the output value of SP 800-90B. Statistical tests also verify whether the output is independent and identically distributed (IID) and use the default tools on the IID track. Further reconstructions are performed with additional entropy estimators on tracks other than IID, and the minimum estimates of all the estimators are selected as the final result. AIS and SP 900-90B emphasize different aspects. As entropy estimation depends heavily on the probabilistic model provided, AIS 20/31 is more rigorous, but also error-prone. Consequently, NIST certification is convenient for designers and verifiers to estimate the entropy of entropy sources.

As a statistical method, the second draft of NIST SP 800-90B adopted some of the methods based on the output data of the entropy example. The basic method is a frequency coefficient estimator, which approximates the power distribution over frequency as shown in Ref. [6]. Alternatively, Bayesian methods are to estimate entropy, such as in Ref. [7]. In addition to these two main methods, compression algorithms are used to estimate entropy for compression entropy estimation, for example, to propose the concept of entropy statistics for entropy estimation [8]. Kelsey and others [9] proposed a new prediction model for entropy prediction. This method is based on simple conversion and decision knowledge, and it is a new approach included in the second draft of NIST SP 800-90B.

In this paper, we propose a method of generating random numbers using the natural decay of beta radioisotopes, which is a true random-number generation technology using beta decay, which can be produced in small quantities and mass. This study shows that the generated random numbers satisfy the SP 800-22 [10] and SP 800-90B [11] standards established by the National Standards Institute.

TABLE 1 Comparison of radioactivity types

Radioactive decay type	Size	Portability	Generation rate	Semiconductor damage
Alpha (Am-241)	Small	High	Low	High
Beta (Ni-63)	Small	High	High	Low

Radioisotopes spontaneously release energetic particles, such as alpha or beta particles (alpha or beta rays), for stabilization, which is a natural decay phenomenon. The spontaneous decay of radioisotopes has the advantage of being able to produce small-scale and mass-produced products that are not affected by the randomness of the decay events and the random number generated, and the change in the physical environment. Intrinsic random-number generators have been constructed using quantum mechanical random phenomena and specifically, the natural decay of radioisotopes [12–14]. In particular, beta source-based random-number generation shows the advantages of small size, portable line, high speed, mass production, low generation rate, and low semiconductor damage compared with that based on alpha sources, as indicated in Table 1. Furthermore, a beta cell can be constructed from a p-i-n junction based on silicon carbide fabricated on a custom semiconductor and Ni foil plated with the Ni-63 radioisotope [15].

In this paper, we propose a new method to improve true random-number generation. The proposed method provides a strategy for counting the values of "0" and "1" of random numbers based on a beta source by storing P-values, labeling them, and comparing the new random numbers. The proposed scheme is implemented in RTL [16], and the generated true random number is verified using SP 800-22 and SP 800-90B standard tests. Section 2 discusses the proposed algorithm and architecture. Section 3 presents the system architecture and RTL simulation results of the LabVIEW-based random-number generator and describes the SP 800-22 and SP 800-90B test methods. Section 4 concludes the paper.

2 | PROPOSED ALGORITHM AND ARCHITECTURE

2.1 | True random-number generator system

The random-number generation system consists of a beta source, silicon PN diode, preamplifier, pulse shaper, discriminator, counter, random-number output circuit, and controller. The random-number generator generates intrinsic random numbers from events, entropy, or particles emitted from the beta source. The source detector consists of silicon PN diodes and can detect particles emitted from a source such as a radioisotope. The source of the silicon PN diode is a beta source, and the source detector generates a detection signal based on

the detected particles. The detection signal is an analog electrical signal, such as voltage or current, and is used as the input of the preamplifier. As an example, the time at which particles are detected and the time at which they are not detected in the detection signal are distinguished. The preamplifier amplifies the detection signal to generate an amplification signal, and amplifies the magnitude of the detection signal, so that the time at which the particles are detected and the time at which they are not detected can be easily distinguished. The amplified signal allows the output signal of the preamplifier to remain in the short section for a long time, maintains the peak point for a long time, and solves the problem that the output decreases for a long time through the pulse shaper and the discriminator. Logic pulses greater than the noisy level are sent out to allow the times at which particles are detected to be determined based on the magnitude of the detection signal.

The counter counts the number of clocks during the time interval between two adjacent pulses in the pulse signal. The counter receives a clock signal from the controller to count the clocks. However, the clock signal is generated directly inside the counter. The counter generates binary count values as the count result of each time interval between the pulses. Each binary count value includes a "0" or "1" value of the set number of bits. The number of binary count values with the set number of bits corresponds to the number of measured time intervals. Each binary count value depends on the length of the measured time interval. That is, as the length of the measured time interval increases, the corresponding binary count value increases. The random-number output circuit can validate the binary count values generated by the counter based on the conversion result. Validating binary count values indicates extracting binary count values from real random numbers.

As the sample size of true random numbers becomes larger, the probability that the random number (binary count value) will have a certain value becomes more uniform across different values. For example, if there are n bits in a binary count value, the probability that one of them converges to 1 is $1/2^n$. That is, if the sample is larger, the binary count values have evenly distributed values in the range of the random number, and the numbers of "1" and "0" values become nearly equal. That is, when the difference between the numbers of "0" and "1" values included in each of the binary count values is large, the binary count values would be pseudo-random numbers. If there is no or a small difference between the numbers of "0" and "1" values included in each of the binary count values, the binary count values would be true random numbers.

Accordingly, the random-number output circuit can perform the conversion and decision. The random-number output circuit extracts "0" and "1" values from the binary count value. Based on the numbers of extracted "0" and "1" values, the random-number output circuit generates the conversion data. Here, the converted data directly or indirectly indicate the numbers of extracted "0" and "1" values. Training transform data, test transform data, and target transform data are all generated by the particles emitted from the source as shown in Figure 1. That is, the binary count value generated according to the time interval at which the particles are detected is converted into conversion data.

The random-number output circuit classifies the binary count value into a true random number or a pseudo-random number based on the training conversion data. Based on the classification result, the random-number output circuit labels the training transform data to generate labeled data. The labeled data include conversion data, and the binary count value corresponding to the conversion data indicates the type of random number. For example, if the binary count value is a true random number, the labeled data include conversion transform data and indicate that the transform datum is a true random number. As another example, when the binary count value is a pseudo-random number, the labeled data include training transform data and indicate that the training transform datum is a pseudo-random number. In the following descriptions, the training transform data and labeled data generated based on the binary count value, which is a pseudo-random number, are represented as being included in a pseudo class. In addition, the transform and labeled data generated based on the binary count value, which is an intrinsic random number, are represented as being included in the intrinsic class. The random-number output circuit obtains reference data for classifying the labeled data into a pseudo or intrinsic class as a conversion result, based on the labeled data.

The reference data include information about a hyperplane. A hyperplane is a boundary for classifying labeled data as a pseudo or intrinsic class. For example, the labeled data displayed below the hyperplane are classified into an intrinsic class, and those displayed above the hyperplane are classified into a pseudo class. The random-number output circuit determines whether the random number is truly random based on the test conversion data. By using the reference data generated

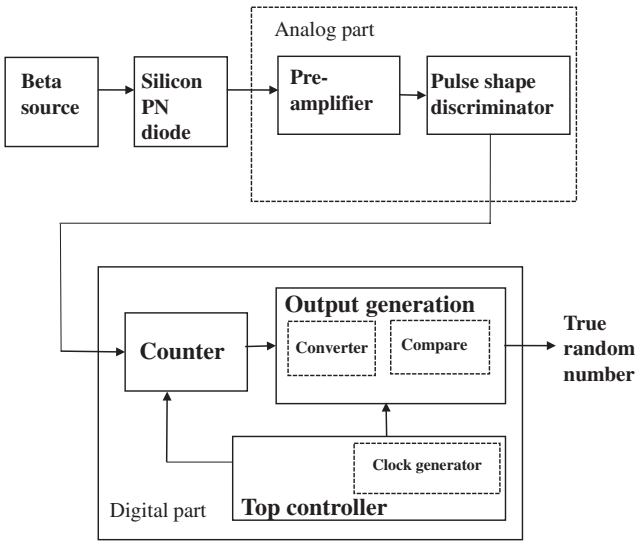


FIGURE 1 True random-number generator system

through the random-number output circuit conversion, the test conversion data can be classified into a pseudo or intrinsic class. The random-number output circuit determines whether the test conversion data classified into the intrinsic class satisfy the intrinsic condition. The intrinsic condition is intended for determining whether the binary count value corresponding to the test conversion data, classified into the intrinsic class, is a true random number. The random number output circuit in Figure 1 can change the parameters under the control of the controller. With further conversion, the existing hyperplane can be modified to have a different value. If sufficient conversion is performed, the random-number output circuit classifies the target transform data into a pseudo or intrinsic class based on the reference data. Based on the classification result, the random-number output circuit can output a true random number.

2.2 | Random-number generation algorithm

Figure 2 is a flowchart illustrating the operation of the random-number generator. Conversion is a process in which the random-number device generates the reference data based on the converted data. In the secondary conversion process, the random-number generating device adjusts the reference data based on the test conversion data. In operation, the random-number generating device performs an initialization operation. For example, the reference data previously stored are deleted through the initialization operation. Then, the random-number generating device measures the time interval at which particles are detected, to generate a conversion binary count value. It generates the training conversion data based on the numbers of "0" and "1" values included in the training binary count value. It generates the labeled data by labeling the training conversion data. As shown in Figure 2, "conversion data" refer to a conversion binary count value, transform data, or labeled data. The labeled data can be used to perform first-order conversion. The random-number generator generates reference data through the conversion data. It generates a test binary count value by measuring the time interval at which particles are detected. Then, it generates test conversion data based on the numbers of "0" and "1" values included in the test binary count value. It operates based on the test conversion data. Then, it classifies the test conversion data into a pseudo or intrinsic class using the reference data generated through the operation. The random-number generator determines whether the test conversion data are classified into an appropriate class. If the test conversion data are not classified into a suitable class, the random-number generator generates new reference data. When the test conversion data are classified into a suitable class, the random-number generator classifies the target data using the reference data. The time interval at which particles are detected can be measured, to produce a target binary count value. Based on the numbers

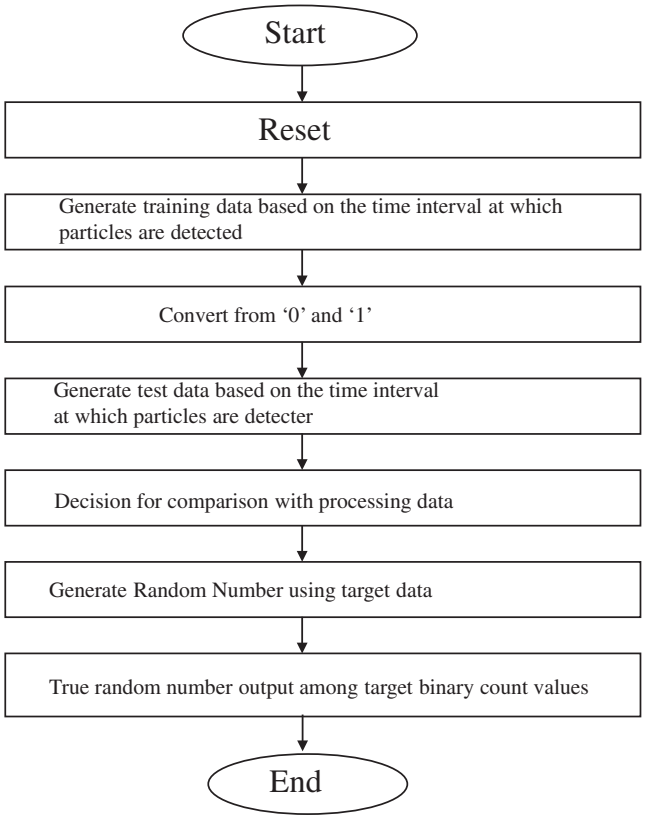


FIGURE 2 Flowchart of the operation of the random-number generator

of "0" and "1" values included in the count value, target conversion data are generated. Based on the newly generated reference data, the validity of the target binary count values is verified. That is, the random-number generator classifies the target transform data into a pseudo or intrinsic class. It outputs a target binary count value corresponding to the target conversion data included in the intrinsic class.

Figure 3 is a flowchart illustrating the operation of the random-number output device. The training data are generated based on the count value. Through conversion, the reference data, which indicate information on the hyperplane for classifying values into a pseudo or intrinsic class, are generated. Test data are generated based on the count values, and the generated reference data classify the test data. The proper classification is determined by checking whether conversion has been performed sufficiently. If the test data are not properly classified, the parameters are changed, and the aforementioned steps are repeated. Through this iterative process, random data classified as an intrinsic class among target data are output.

Figure 4 is a flowchart illustrating the updated method. The comparator extracts the "0" and "1" values included in the binary count value. The numbers of extracted "0" and "1" values are counted. The conversion circuit generates the conversion data based on the numbers of "0" and "1" values. Based on the conversion data, it is determined whether

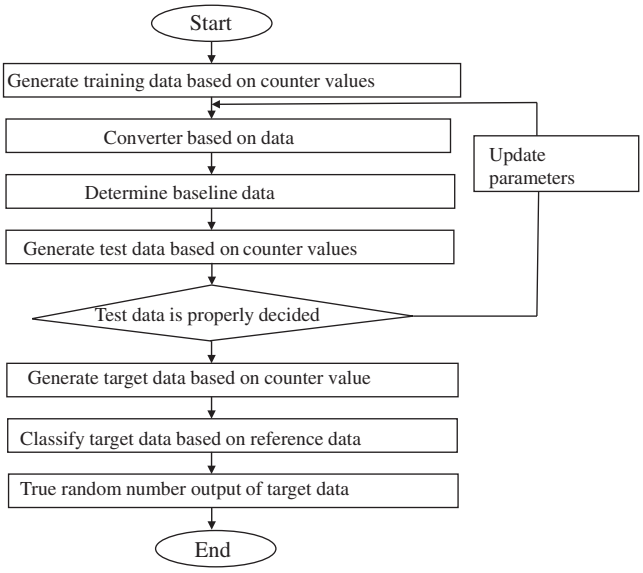


FIGURE 3 Flowchart decision for true random number

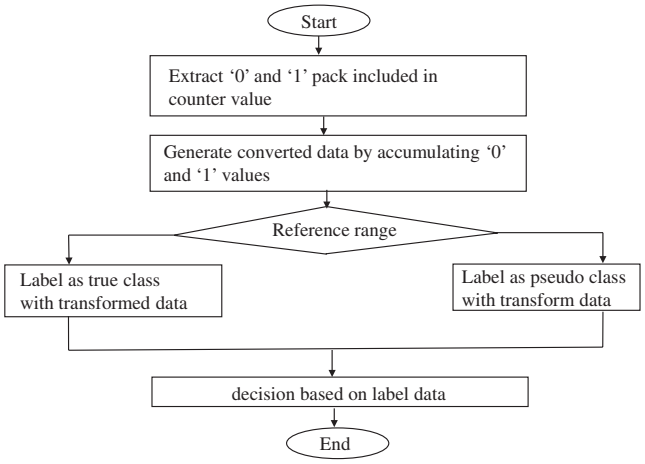


FIGURE 4 Decision for true random number

the numbers of "0" and "1" values are within a reference range. For example, the classification circuit determines whether the ratio of the number of "1" values to the number of "0" values is within a reference range. In this case, the reference range is between 0.49 and 0.50. If the numbers of "0" and "1" values are within the reference range, the classification circuit labels the transform data with a true class to generate labeled data. If the numbers of "0" and "1" values are not within the reference range, the classification circuit labels the transform data with a pseudo class to generate labeled data.

2.3 | Proposed architecture

Figure 5 is a block diagram showing the digital part. The block consists of a flip-flop, counter, converter, multiplier-accumulator (MAC), memory, comparator, total controller, and clock generator. The counter measures the time intervals between pulses to produce a count value. It works by synchronizing the influx pulse signal of the flip-flop and the detection time of the particles, that is several pulses corresponding to the time between the pulses. The number of clock ticks during the interval is counted. The output of the counter accumulates the numbers of "0" and "1" values in the converter. In the MAC block, the cumulative product of the output value of the converter is calculated and is stored in memory. When a new random value is input, the comparator compares it with the stored value to determine the pseudo-random number and true random number, and outputs the true random number.

Figure 6 shows a block diagram of a 32-bit fixed-bit multiplier. The block comprises an 8-bit adder, 9-bit subtractor, 24×24 multiplier, XOR, normalizer, and final output. It has 32-bit single-precision IEEE 754 floating-point operation, with 1-bit sign, 8-bit exponent, and 23-bit mantissa place values.

3 | SIMULATION AND VERIFICATION

3.1 | RTL simulation

This platform design architecture uses a dedicated hardware engine with a system clock operating at 200 MHz.

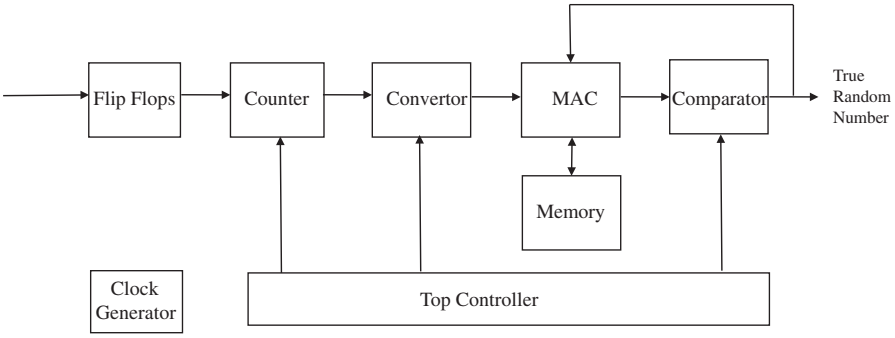


FIGURE 5 Block diagram of random generator

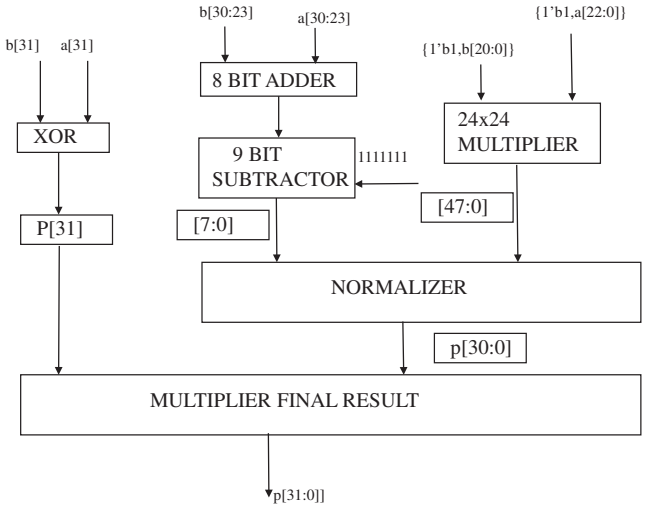


FIGURE 6 32-bit floating-point multiplier architecture

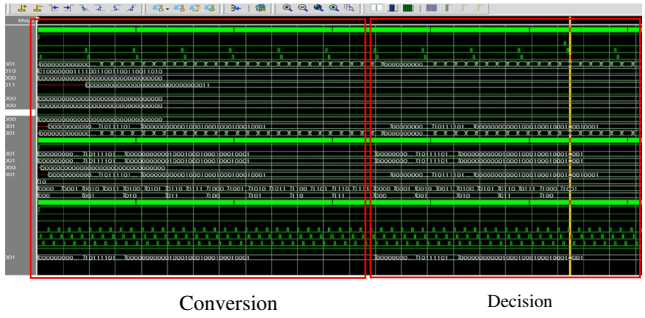


FIGURE 7 RTL simulation [Colour figure can be viewed at wileyonlinelibrary.com]

Figure 7 shows the simulation result of the proposed module using Verilog hardware description language and verification. The system is implemented using the 0.45 µm library (TSMC). The simulation shows the conversion and classification of results for the lightweight true-random generator.

TABLE 2 Logic synthesis*

Block	Area (µm²)	Gate counter (Gates)
Adder	2607.2	3725
Multiplier	3300.8	4715
Accumulator	264.1	377
Flip-flops	404	577
Counter	905	816
Converter	571	1387
Top control	188	269
Sum	8240.1	11 866

*Target Lib: TSMC 45 nm tcbn45gsbwplvtbc Gate Counter: 2-input NAND gate = 0.7 μm^2.

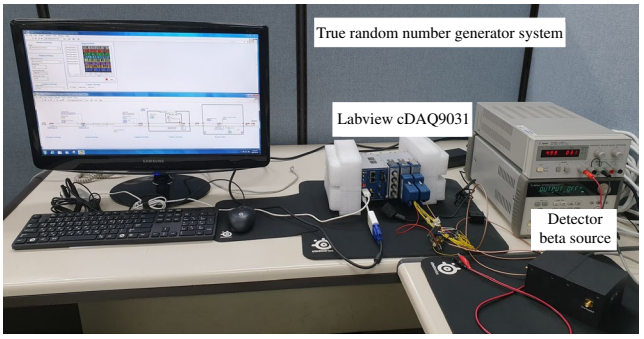


FIGURE 8 True random-number generator system [Colour figure can be viewed at wileyonlinelibrary.com]

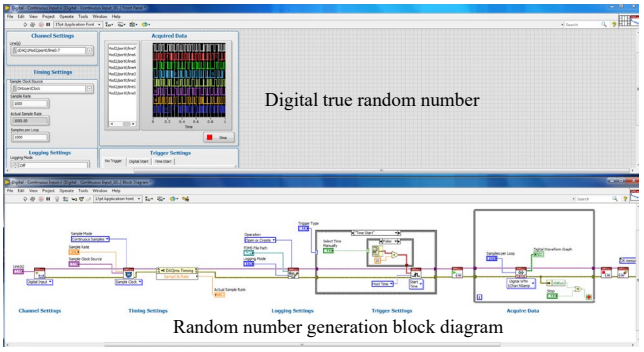


FIGURE 9 Digital true random-number generator system based on LabVIEW [Colour figure can be viewed at wileyonlinelibrary.com]

3.2 | Logic synthesis

Table 2 shows the result of logical synthesis. The TSMC 40 nm target library was used to synthesize logic using the Synopsys design compiler. The logic synthesis resulted in a logic gate count of approximately 11,866.

3.3 | System design

The beta source-based true random-number generator is shown in Figure 8. The system was designed using a beta source, detector, and LabVIEW (cDAQ9031) system.

Figure 9 shows a LabVIEW-based digital random-number generator system. The digital random output of the FPGA is shown via a digital waveform.

3.4 | SP 800-22 Test

Table 2 shows the statistical test suite for the random numbers based on Python. The statistical tests on the randomness of the random numbers consisted of 15 types of test methods using pseudo-random numbers and true random numbers on a 1.6M bit dataset. The experiment showed that the true random numbers generated using the beta source increase

TABLE 3 NIST SP800-22 test

Items	*P*-value of pseudo-random number	*P*-value of Proposed method	Results
Monobit test	0.292462749356	0.0501884500565	Pass
Frequency within block test	0.230443081147	0.0150075360195	Pass
Runs test	0.4984920945	0.31234747611	Pass
Longest run ones in a block test	0.326054130504	0.242788674655	Pass
Binary matrix rank test	0.169187100709	0.230412253352	Pass
DFT test	0.460252039688	0.783303943576	Pass
Non-overlapping template matching test	0.999999815175	1.00014332003	Pass
Overlapping template matching test	0.313392365595	0.894431974104	Pass
Maurer's universal test	0.999909497336	0.999554384562	Pass
Linear complexity test	0.106412399501	0.800437373096	Pass
Serial test	0.132634957334	0.0175486078436	Pass
Approximate entropy test	0.131667431003	0.0175004296891	Pass
Cumulative sums test	0.305577573144	0.0287624305175	Pass
Random excursion test	0.0746419980822	0.185145908489	Pass
Random excursion variant test	0.22010219462	0.0875273565741	Pass

the reliability of the marginal line with a significant level of probability of making an error.

3.5 | SP 800-90B test

For each IID and non-IID test, 1M data are collected and a restart test is performed. SP 800-90B is a statistical evaluation of noise sources and is used to evaluate random-number generators in cryptographic module verification systems [7,8]. As a representative standard to evaluate the safety of noise sources, a second draft complementing the existing SP 800-90B was published by NIST on January 27, 2016 [14]. This revision of the SP 800-90B introduces a new method of estimating using a predictor to estimate the minimum entropy of non-IID tracks. The predictors are known to detect the periodic characteristics of a noise source easily, but no specific reference is made to those characteristics. The noise source based on the beta source was tested using the software provided in the standard.

All test conditions have a minimum entropy of 5.768870 in the non-IID track. Tables 3, 4, and 5 show a random number SP 800-90B test using pseudo-random numbers and beta sources. The test results showed that the proposed random number generation passed SP 800-90B. Furthermore, the pseudo-random number obtained a Chi-square independence score of 65 212.6, and the proposed method obtained a value of 64 757.5. The minimum entropy was 7.86511 in the IID test, which exceeded that of the pseudo-random number, and 5.78597 in the non-IID test.

TABLE 4 NIST SP800-90B test

Test items	Min-entropy of pseudo-random number	Min-entropy of proposed method	
IID	7.86511	7.85964	True
Restart	7.86511	7.85964	True
Non-IID	5.78597	5.76887	True
Restart	5.78597	5.76887	True

4 | CONCLUSION

This paper presents an efficient hardware random-number generator based on beta sources. The proposed random-number generator uses a method to distinguish between pseudo- and true random numbers via simple cumulative comparison operations in real time, which is then implemented in the hardware. The random-number generator generates labeled data indicating whether the count value is a pseudo- or true random number according to the bit value of the binary count value based on the generated labeling data. Random and pseudo-random numbers are used as reference data. The proposed method is verified with Verilog RTL coding and a LabVIEW-based system for hardware implementation. The generated random numbers are tested according to the NIST SP 800-22

TABLE 5 NIST SP800-90B test

Test items	Pseudo-random number	Proposed method
Chi-square independence score	65 212.6	64 757.5
Chi-square goodness-of-fit score	2449.48	2373.33
IID permutation	Pass	Pass

and SP 800-90B standards and satisfy the test items provided therein.

ORCID

Seongmo Park https://orcid.org/0000-0001-8656-9094
Taewook Kang https://orcid.org/0000-0001-9147-3898
Kyunghwan Park https://orcid.org/0000-0001-9852-6155

REFERENCES

1. A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*, Chapter 3. CRC Press, 1996, pp. 87–131.
2. H. Corrigan-Gibbs and S. Jana, *Recommendations for randomness in the operating system: How to keep evil children out of your pool and other random facts*, in Proc. USENIX HotOS (Switzerland), May 2015, pp. 1–5.
3. G. Marsaglia, *The diehard test suite*, 2003. Available at http://www.csis.hku.hk/~diehard/
4. W. Killmann and W. Schindler, *A proposal for: Functionality classes for random number generators*, AIS 20/31. http://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Zertifizierung/Interpretationen/AIS_20_Functionality_classes_for_random_number_generators_e.pdf;jsessionid=A837238C46E2F9205B0C2AF043153011.2_id294?__blob=publicationFile
5. M. S. Turan et al., *Recommendation for the entropy sources used for random bit generation*, (Second DRAFT) NIST Special Publication 800–90B, Jan. 2016. http://csrc.nist.gov/publications/drafts/800-90/sp800-90b_second_draft.pdf
6. P. Grassberger, *Entropy estimates from insufficient samplings*, arXiv:physics/0307138, 2003.
7. D. H. Wolpert and D. R. Wolf, *Estimating functions of probability distributions from a finite set of samples, part 1: Bayes estimators and the Shannon entropy*, arXiv preprint comp-gas/9403001, 1994.
8. P. Hagerty and T. Draper, *Entropy bounds and statistical tests*, http://csrc.nist.gov/groups/ST/rbg_workshop_2012/hagerty_entropy_paper.pdf
9. J. Kelsey, K. A. McKay, and M. S. Turan, *Predictive models for min-entropy estimation. In Cryptographic Hardware and Embedded Systems*, in Proc. Int. Workshop, Saint-Malo (Saint-Malo, France), Sept. 13–16, 2015, pp. 373–392.
10. National Institute of Standard and Technology, *A statistical test suite for random and pseudorandom number generators for cryptographic applications*, Special Publication 800-22,15, 2001.
11. NIST, *Recommendation for the Entropy Sources Used for Random Bit Generation, NIST DRAFT Special Publication 800-90B*, Aug. 2012.
12. H. Park et al., *Key derivation functions using the dual key agreement based on QKD and RSA crypto-system*, J. KICS **41** (2016), no. 4, 479–488.
13. K. J. Ha, C. H. Seo, and D. Y. Kim, *Design of validation system for a crypto-algorithm implementation*, J. KICS **39B** (2014), no. 4, 242–250.
14. NIST, *Recommendation for the Entropy Sources Used for Random Bit Generation*, (Second DRAFT) NIST Special Publication 800-90B, Jan. 2016.
15. T. Kang et al., *Evaluation of a betavoltaic energy converter supporting scalable modular structure*, ETRI J. **41** (2019), 254–261.
16. S. Park et al., *A Sing-Chip Video/Audio CODEC for Low Bit Rate Application*, ETRI J. **22** (2000), 20–29.

AUTHOR BIOGRAPHIES

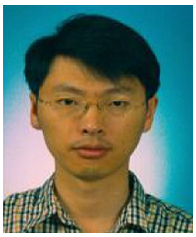
Seongmo Park received the BS, MS, and PhD degrees in electronic engineering from Kyungpook University, Taegu, Korea, in 1985, 1987, and 2006, respectively. In 1987, he joined the GoldStar Semiconductor Company Gumi, Korea, where he was involved in the research and development of Mask ROM design, and SoC design. He joined the Electronics and Telecommunications Research Institute (ETRI), Daejeon, Korea, in 1992. From 1992 to 1995, he was involved in the research and development of ASIC design. From 1996 to 2000, he was involved in the development project of H.263 (Video Compression SoC). From 2001 to 2003, he was in charge of the government projects on MPEG-4 (Moving Picture Expert Group) SoC for the IMT-2000 application. He successfully implemented MPEG-4 SoC with the application of video conferencing. From 2004 to 2006, he successfully implemented H.264 SoC with a 90 nm process for mobile device applications. From 2007 to 2009, he worked at Multi-Format Multimedia SoC based on MPCore Platform as a team leader. From 2010 to 2011, he worked toward the development of SVC codec for IPTV Terminal as a project leader, and as a project member on Video Codec SoC for UHD mobile devices and smart TVs. He received Marquis Who's Who 2014–2017 Achievement Award, the IBC Cambridge "2000 Outstanding Intellectuals of the 21st Century 2016" Achievement Award, and the 2017 Albert Nelson Marquis Lifetime Achievement Award. He is currently working as a project member on neuromorphic SoC design and the development of a true random-number generator device. He is a senior member of IEEE, a principal member of the engineering staff at ETRI, and a professor of UST. His interests include machine learning algorithms, neuromorphic architecture design, video compression algorithms, SoC architecture design, and true random-number generator devices based on beta-voltaic battery.

Byoung Gun Choi received the BS degree in electronic engineering from Yeungnam University in 1995 and the MS and PhD degrees from the Information and Communications University (ICU) in 2000 and 2005, respectively. He was with Samsung Electronics Co. as a semiconductor engineer from 1995 to 1996. His research interests include beta-voltaic battery technology and semiconductor devices. Since 2005, he has been with the Electronics and Telecommunications Research Institute, Daejeon, Republic of Korea, where he is currently a Principal Researcher.

Taewook Kang was born in Daejeon, Korea, in May 1980. He received the BS and MS degrees in electrical engineering from Pohang University of Science and Technology (POSTECH), Pohang, Korea, in 2005 and 2007, respectively. Since February 2007, he has been with the Electronics and Telecommunications Research Institute (ETRI), Daejeon, Korea, where he is currently a Senior Researcher. He has been primarily studying human body communications. His research interests include communication systems, radio channel modeling, and power management of energy harvesting systems.

Kyunghwan Park received the MS and PhD degrees in electrical and electronic engineering from Korea Advanced Institute of Science and Technology, Daejeon, Korea, in 1993 and 1997, respectively. From 1997 through 2000, he worked at DACOM R&D Center, Daejeon Korea. Since January 2001, he has been with the Electronics and Telecommunications Research Institute, Daejeon, Korea as a Principal Member of Engineering Staff. His research interests include RF/Analog IC for wireless communications, RFID chips, radiation detection circuits, and random-number generators based on radiation and beta-voltaic battery.

Youngsu Kwon received the BS, MS, and PhD degrees from Korea Advanced Institute of Science and Technology (KAIST), Republic of Korea in 1997, 1999, and 2004, respectively. He worked toward designing 3-dimensional FPGA at Microsystems Technology Laboratory (MTL), Massachusetts Institute of Technology as a Postdoctoral Associate from 2004 to 2005. He has been with the AI SoC Research Department, Electronics and Telecommunications Research Institute (ETRI), Republic of Korea, since 2005. At ETRI, he is a Director and Principal Member of Research Staff of the AI SoC Research Department devoted to the design of the AI processor, AB. He has special interests in many-core architecture, AI processor design, low-power architecture design, computer-aided design, and algorithmic optimizations of circuits and systems. He received the Presidential Prize from the Korean Government in 2016, Official Commendations from the Ministry of Science and ICT as well as Ministry of Industry in 2016, the Excellent Researcher Award from Korea Research Council in 2013, the Industrial Contributor Award from Korean Federation of SMEs in 2013, and medals from Samsung's Thesis Prizes in 1997 and 1999.



Jongbum Kim received the MS degree in electrical engineering from ChungNam National University and the PhD degree in nuclear and quantum engineering from Korea Advanced Institute of Science and Technology, Daejeon, Korea, in 2000 and 2011, respectively. Since January 2000, he has been with the Korea Atomic Energy Research Institute, Daejeon, Korea as a Principal Researcher. His research interests include radioisotope application, radiation measurement, and quantum random-number generator.