A Worst-Case-Aware Design Methodology for Noise-Tolerant Oscillator-Based True Random Number Generator With Stochastic Behavior Modeling

Takehiko Amaki, *Student Member, IEEE*, Masanori Hashimoto, *Senior Member, IEEE*, Yukio Mitsuyama, *Member, IEEE*, and Takao Onoye, *Senior Member, IEEE*

***Abstract*—This paper presents a worst-case-aware design methodology for an oscillator-based true random number generator (TRNG) that produces highly random bit streams even under deterministic noise. We propose a stochastic behavior model to efficiently determine design parameters, and identify a class of deterministic noise under which the randomness gets the worst. They can be used to directly estimate the worst χ value of a poker test under deterministic noise without generating bit streams, which enables efficient exploration of design space and guarantees sufficient randomness in a hostile environment. The proposed model is validated by measuring prototype TRNGs fabricated with a 65-nm CMOS process.**

***Index Terms*—True random number generator, Markov chain, stochastic model, Jitter.**

I. Introduction

RANDOM number generation, which is indispensable for secret- and public-key generation and challenge and response authentication, is a fundamental underlying technology to accomplish highly secure systems. Oscillator-based true random number generators (TRNGs) [1]–[3] are popular circuits that produce physical random numbers. Fig. 1 has a block diagram of a basic oscillator-based TRNG, which consists of a sampler and two distinct oscillators; one is fast and the other is slow. The sampler acquires bits from the fast oscillator (D in Fig. 1) using the signal of the slow oscillator as clock (CK in Fig. 1). The oscillators inherently have jitter because of internal noise, and hence the rise timing of the slow oscillator signal fluctuates from the viewpoint of the rising edges of the fast oscillator. The oscillator-based TRNG generates random numbers exploiting this jitter as a random source. Random period jitter, which is defined as the standard deviation of periods, has been called 'jitter' for the sake of brevity in this paper.

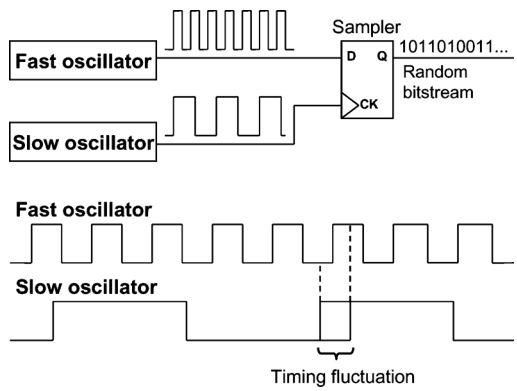

Fig. 1. Basic oscillator-based TRNG.

To design a TRNG that satisfies given performance specifications, we need to estimate the randomness of TRNGs and procure appropriate design parameters. Despite the requirements for randomness estimation, it is difficult to simulate oscillator-based TRNGs because the jitter of oscillators cannot be directly considered in ordinary circuit simulators such as Synopsys HSPICE and NanoSim. They could simulate oscillator-based TRNGs by modeling jitter with pseudorandom numbers through Verilog-A, for example, but it takes an unacceptably long time for simulations since randomness tests require long bit streams. Furthermore, the oscillation periods for the two oscillators and their jitter are on different orders of magnitude and hence the time steps for transient simulations must be kept small to ensure the simulation accuracy. Therefore, an efficient behavioral model and a method of evaluating the randomness of oscillator-based TRNGs are necessary to guide explorations in design space and meet the design specifications. Also, a design method that takes into consideration deterministic noise is required because a TRNG should guarantee its sufficient randomness even when unwanted noises or malicious attacks occur. Note that randomness of TRNG in this paper just means the statistical randomness of output data, and unpredictability is not specifically discussed.

Petrie and Connelly [5], [6] modeled a slow oscillator under random noise and deterministic signals as a voltage-controlled oscillator (VCO). They discussed the required jitter to produce

Manuscript received December 22, 2011; revised May 20, 2012, October 17, 2012, and December 30, 2012; accepted June 10, 2013. Date of publication June 27, 2013; date of current version July 10, 2013. This work was supported by the SCOPE program of the Ministry of Internal Affairs and Communications. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Miodrag Potkonjak.

T. Amaki, M. Hashimoto, and T. Onoye are with the Department of Information Systems Engineering, Osaka University, Suita-shi 565-0871, Japan, and also with JST, CREST, Tokyo 102-0075, Japan (e-mail: amaki.takehiko@ist.osaka-u.ac.jp; hasimoto@ist.osaka-u.ac.jp; onoye@ist.osaka-u.ac.jp).

Y. Mitsuyama is with the School of Systems Engineering, Kochi University of Technology, Kami-shi 782-8502, Japan, and also with JST, CREST, Tokyo 102-0075, Japan (e-mail: mitsuyama.yukio@kochi-tech.ac.jp).

Digital Object Identifier 10.1109/TIFS.2013.2271423

sufficient randomness and the effects of deterministic noise with a poker test [7]. They also investigated the frequency ratio of the two oscillators, but it was limited to be a small number of about 15. Although they [5] claimed that a larger frequency ratio resulted in better randomness, this tendency might depend on an assumption about the behavior of an oscillator-based TRNG and they provided limited quantitative evaluation in terms of frequencies. In addition, their proposed model [5] could not be used to evaluate the effect of deterministic noise accurately when the noise frequencies were higher than that of the slow oscillator, and therefore design with the model could not ensure sufficient randomness under high-frequency noise. Moreover, the model was not validated with hardware measurements. Bucci *et al.* [2] introduced numerical formulas that gave the transition probability between successive bits as a function of the average and the standard deviation of oscillation periods and the initial phase difference between the two oscillators. However, they did not rigorously test randomness with, such as, [7], [8], [10], and did not consider deterministic noise. Bernard *et al.* [11] proposed a mathematical model of a TRNG using two jittery clocks with rationally related frequencies, and the model could be used to evaluate entropy per bit and bias on the generated bit stream. Their model, however, did not take deterministic noise into consideration. Baudet *et al.* [12] modeled the oscillators of a TRNG with a phase-oriented approach, and they provided formulas for entropy rates. They also introduced a method of measuring jitter by filtering out deterministic jitter. Ergün [13] modeled a chaotic oscillator that was used as a slow oscillator, and he provided design guidelines based on estimates of entropies. The model was, however, tailored for a chaotic oscillator and a VCO, and hence it could not be used for other types of oscillator-based TRNGs.

In our preliminary work [14], we proposed a procedure for designing an oscillator-based TRNG with a stochastic behavior model. We determined the design parameters with the model without taking deterministic noise into account, and then evaluated robustness to power-supply noise with bit generation. The procedure, however, required iterations of exploring design space and checking of robustness until ad hoc design modifications attained sufficient robustness to the supply noise. Moreover, the randomness under deterministic noise was evaluated with only a small subset of possible deterministic noises. In reality, the number of possible deterministic noises is infinite. Thus, the identification of the worst-case through a number of simulations is impossible, and hence the procedure did not guarantee the randomness under deterministic noise.

We propose a worst-case-aware design methodology in this paper using a stochastic behavior model. Key design parameters are explored and determined with the stochastic model we propose. The worst randomness under deterministic noise is quickly evaluated with a number of design parameters using a model without bit generation, and appropriate design parameters are determined so that the TRNG passes tests and satisfies the required specifications, which makes design iterations unnecessary. The proposed worst-case-aware design method guarantees enough randomness even under any waveform shapes of deterministic noise, because the worst case derived in this work is the theoretically-proven worst case, and there are no deterministic noises worsen than the worst case. This contribution comes from an identification of the class of deterministic noise which causes the worst situation.

The behavioral model we propose utilizes a Markov chain, and it is used to quickly estimate the worst χ value of a poker test (defined in the FIPS 140-2 [7]) under any deterministic effects without generating a bit stream. The principal design parameters, which are average periods of oscillators, the duty cycle of the fast oscillator, and the use of correctors, are determined guided by the estimated χ values and target χ values. The quality of TRNG outputs can be quantitatively evaluated with other standard randomness tests as well when necessary since the model can also generate a bit stream. It should be noted that the proposed method can be applied to all types of oscillators, since the parameters of interest are independent of the topology or type of oscillator. Furthermore, we test and validate the proposed model with hardware measurements of an oscillator-based TRNG implemented with a 65-nm CMOS process.

The three main contributions our work makes are:

- to propose the worst-case-aware design methodology to guarantee sufficient randomness under deterministic noise,
- to verify the efficiency of the worst-case-aware design method with gate-level TRNG simulation, and
- to identify the class of deterministic noise under which the randomness gets close to the lowest.

The rest of this paper is organized as follows. Section II proposes a behavioral model with a Markov chain and a procedure for evaluating randomness. Section III validates the model with hardware measurements. Section IV proposes a methodology to calculate randomness in the worst case without bit generation. The efficiency of the proposed design methodology is proven with gate-level noise-aware TRNG simulation, which is explained in Section V. Section VI presents an example to illustrate how appropriate design parameters are derived and Section VII gives concluding remarks.

II. Proposed Stochastic Behavior Model

This section proposes a behavioral model of oscillator-based TRNGs using a Markov chain.

A. Behavioral Model of Oscillator-Based TRNG

A Markov chain is a discrete-state/discrete-time stochastic process, $\{X_n\} = \{X_0, X_1, X_2, \ldots\}$, where $\{X_n\}$ is a sequence of random variables, which satisfies, for each r, a Markov property, i.e., [6]

$$P(X_r = x_r | X_{r-1} = x_{r-1}, X_{r-2} = x_{r-2}, \ldots, X_0 = x_0) = P(X_r = x_r | X_{r-1} = x_{r-1}). \quad (1)$$

This means that next state X_{n+1} only depends on current state X_n and is independent of past states $X_0, X_1, \ldots, X_{n-1}$.

Before explaining the proposed model, we will describe an assumption about the model. We have assumed that jitter in the oscillators is temporally uncorrelated, which means that we take into consideration thermal noise, shot noise, and/or 1/f noise but not deterministic noise (this will be considered in Section IV) such as power-supply noise, substrate noise, and external noise.

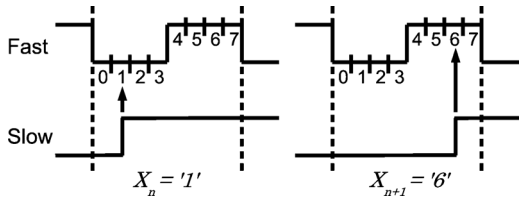

Fig. 2. Example application.

Given this assumption, a Markov chain can be applied to the behavioral modeling of an oscillator-based TRNG.

The fast oscillator waveform of one cycle is divided into m spans and each span is regarded as a state in the proposed model. Thus, a Markov chain that has m-state space is constructed. Let us suppose the n-th rising edge is the timing of a slow oscillator. Here, we define this timing as time n. The fast oscillator at this rise timing stays in one state of the m states defined above. The X_n denotes the state at time n. The TRNG generates the n-th bit corresponding to X_n, since each state corresponds to low or high. Fig. 2 outlines an example where the model is applied to a TRNG where $m = 8$. The TRNG takes state 1 at time n and state 6 at time $n + 1$, and then $X_n = 1$ and $X_{n+1} = 6$. In this example, since states 0, 1, 2, and 3 are low and states 4, 5, 6, and 7 are high, the n-th output is 0 and the $(n + 1)$-th output is 1.

B. Model Construction and Use

This subsection explains the process of evaluating randomness with the Markov model. 1) Transition matrix and 2) state probability vector are calculated, and then 3) random bit streams are generated and evaluated with statistical randomness tests. Each step is explained in what follows.

1) Calculation of Transition Matrix: This step is used to construct transition matrix $\mathbf{P}$ that characterizes the state transition of the Markov chain. The matrix size is $m \times m$ when the model has m-state space. An element of matrix $p_{i,j}$ is the probability of a transition from i to $j (0 \le i, j \le m - 1)$. Transition step a is the number which the state proceeds by and is defined as $\{(j - i) + m\} \bmod m$. Let $q_i(a)$ denote the probability that the next state will advance by a from state i. Assuming a Gaussian distribution, $p_{i,i+a}$ is calculated as:

$$p_{i,i+a} = \sum_{l=-\infty}^{\infty} q_i(a + l \cdot m) = \sum_{l=-\infty}^{\infty} \int_{l \cdot t_{\rm fast} + a \cdot t_{\rm span}}^{l \cdot t_{\rm fast} + (a+1) t_{\rm span}} f_i(x)dx, \quad (2)$$

$$f_i(x) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left(-\frac{(x-\mu)^2}{2\sigma_i^2}\right), \quad (3)$$

where $t_{\rm fast}$ is the average period for the fast oscillator, and $t_{\rm span}$ is the time range for one state and is defined as $t_{\rm fast}/m$. The $f_i(x)$ is the probability density function of a Gaussian distribution whose standard deviation, σ_i, depends on the current state. Note that the proposed model can handle any other distribution shapes as long as they are independent of time, even though a Gaussian distribution has been adopted as a representative shape in this paper. The μ is a remainder where the average period for slow oscillator $t_{\rm slow}$ is divided by that of the fast oscillator. It is most likely that the next timing for sampling will advance by μ from the current. The next sampling timing is distributed more uniformly as σ_i increases. The $p_{i,j} (i > j)$ can also be obtained from (2) since $p_{i,j+m}$ is equal to $p_{i,j}$ while extending the maximum range of j. Thus, $\mathbf{P}$ can be derived with (2).

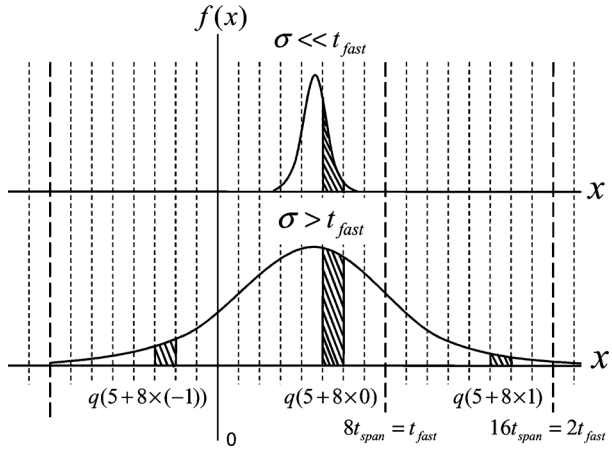

Fig. 3. Example calculation of transition matrix.

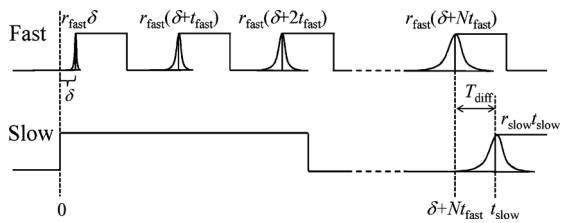

Fig. 4. Jitter accumulation in fast oscillator. Variance of each rise timing is denoted. ($\lfloor \frac{t_{\rm slow}-\delta}{t_{\rm fast}} \rfloor = N$).

Let us explain (2) using the situation in Fig. 2 as a simple example, where m is 8, X_n is '1', and X_{n+1} is '6'. Fig. 3 explains the summation and integration in (2). When $t_{\rm fast}$ is sufficiently large, i.e., $t_{\rm fast} \gg \sigma_i$ (top of Fig. 3), $p_{1,6}$ is approximately obtained as $q(6-1) = \int_{5 \cdot t_{\rm span}}^{6 \cdot t_{\rm span}} f(x)dx$. However, as $t_{\rm fast}$ is comparable to or smaller than σ_i (bottom of Fig. 3), $q(5+8\times(-1)), q(5+8\times1), \cdots$ should not be ignored. As σ_i becomes relatively larger than $t_{\rm fast}$, more terms of q should be summed up, and finally (2) is obtained.

To easily take jitter from both oscillators into consideration, we derive a variance constant and equivalent jitter. An oscillator is composed of stage elements (called gates after this), such as inverters, and the jitter characteristics of gates are an important factor in their design. To discuss this factor, we here define variance constant r as the variance in the stage delay divided by the average stage delay. Due to this definition, the variance constant of an oscillator composed of n gates with r variance constants is conveniently equal to r. The variance constant characterizes the jitter of an oscillator. The variance constant of the fast oscillator, $r_{\rm fast}$, and that of the slow oscillator, $r_{\rm slow}$, can differ. For instance, the variance of periods of the slow oscillator is $r_{\rm slow} t_{\rm slow}$.

Next, Fig. 4 shows an example of waveforms to explain equivalent jitter. Equivalent jitter σ is the time fluctuation between the rise edge of the slow oscillator and the previous rise

edge of the fast oscillator, and is defined as the standard deviation of the time span between the two edges (denoted as $T_{\rm diff}$). The fluctuation of $T_{\rm diff}$ results from the jitter of the slow rising edge at $t_{\rm slow}$ and of the fast rising edge at $\delta + Nt_{\rm fast}$. Here, δ is the initial time difference between the oscillators, and the jitter for the slow edge is $r_{\rm slow}t_{\rm slow}$. N cycles of fast oscillation elapse per cycle of the slow signal where $N = \lfloor \frac{t_{\rm slow}-\delta}{t_{\rm fast}} \rfloor$, and then jitter accumulates in the meantime, since $t_{\rm slow}$ is larger than $t_{\rm fast}$. Then, the variation in the rise edge at $\delta + Nt_{\rm fast}$ is $r_{\rm fast}(\delta + Nt_{\rm fast})$. The two oscillators have different circuits, and hence the rise edges at $t_{\rm slow}$ and $\delta + Nt_{\rm fast}$ are independent of each other. Therefore, the variance in $T_{\rm diff}$ is $r_{\rm slow}t_{\rm slow} + r_{\rm fast}(\delta + Nt_{\rm fast})$. When the current state is i in the Markov chain, the initial time difference is approximated as $\delta \approx t_{\rm fast} - it_{\rm span}$. The error in this approximation, which degrades the accuracy of the model, gets smaller as the size of state-space $m = t_{\rm fast}/t_{\rm span}$ increases, because the error is always less than $t_{\rm span}$. Finally, equivalent jitter is expressed as:

$$\sigma_i = \sqrt{r_{\rm slow}t_{\rm slow} + r_{\rm fast}t_{\rm fast}\left(N + 1 - \frac{i}{m}\right)}. \qquad (4)$$

The parameter of m affects the accuracy and run time for evaluation. To precisely model the behavior, $t_{\rm span}(= t_{\rm fast}/m)$ should be sufficiently smaller than σ_i. The size of m in the experiments will be discussed in Section VI-D.

2) Calculation of State Probability Vector: Given the transition matrix, the next state probability vector, $\boldsymbol{\pi_{n+1}}$, is calculated from the current one, $\boldsymbol{\pi_n}$ as shown in (5) at the bottom of the page.

Transition matrix $\mathbf{P}$ is independent of time n because of the Markov property, and hence $\boldsymbol{\pi_n}$ can be calculated with initial state probability vector $\boldsymbol{\pi_0}$; $\boldsymbol{\pi_n} = \boldsymbol{\pi_0}\mathrm{P}^n$. Fig. 5 plots an example of $\boldsymbol{\pi_n}$ transitions where the initial states are 0 s. Because the average periods of the fast and slow oscillators are 0.3 ns and 50 ns, μ is calculated as 50 ns mod 0.3 ns $= 0.2$ ns. The variance constants of the oscillators are both 1.8×10^{-14} s.

3) Bit Generation and Randomness Tests: Duty cycle d is defined as the ratio of the number of high states to the number of all states m in the model. For example, when states 0 to 29 are low and states 30 to 99 are high $(m = 100)$, d is $(70/100) \times 100 = 0.7$. When the next state probability vector, which can be obtained from the current state, and the duty cycle are given, the next state and the next output are stochastically determined with pseudorandom numbers generated by computer. Repeating this process generates a successive bit stream. Randomness is evaluated by testing the generated bit stream with arbitrary statistical tests.

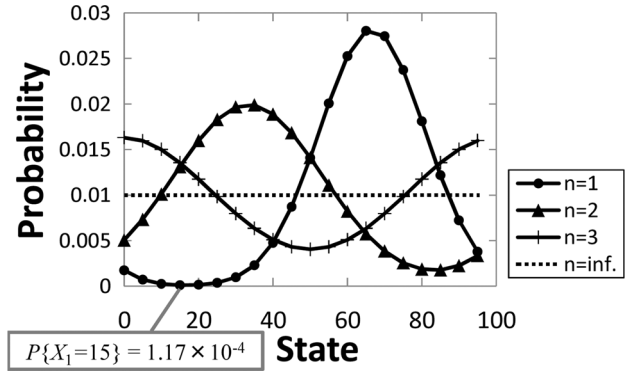

Fig. 5. State probability vectors with progression of time.

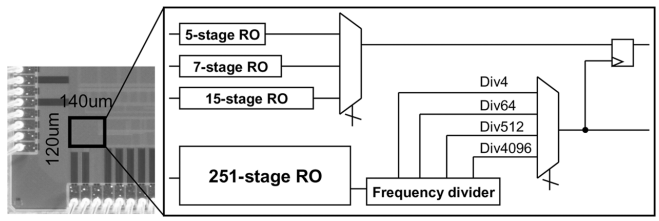

Fig. 6. Chip photos and block diagrams of TRNGs.

Postprocessing with correctors (e.g., the XOR [9] and von Neumann correctors [1]) is a popular technique to improve randomness. When the bit stream is generated by the model, arbitrary correctors can be simply applied to the random numbers and statistical tests are then executed.

III. Model Validation With Hardware Measurements

The proposed Markov model was implemented with MATLAB, and validated with measurements of a prototype TRNG fabricated with a 65-nm process.

A. Test Structure

Fig. 6 shows the test TRNG, chip photos, and block diagrams. The test TRNG was fabricated with e-shuttle 65-nm process. We implemented 5-, 7-, and 15-stage ring oscillators (ROs) as fast oscillators using standard cells with minimum channel length. A 251-stage RO of the slow oscillator is composed of low-leakage standard cells with 10-nm longer channel length. All stage elements of the ROs are static CMOS inverters and 2-input NAND gates. The periods of the ring oscillators are 178.3 ps for the 5-stage RO, 243.2 ps for the 7-stage RO, 502.6 ps for the 15-stage RO, and 10.0 ns for the 251-stage RO from the circuit

$$\boldsymbol{\pi_{n+1}} = \begin{pmatrix} P\{X_n = 0\} \\ P\{X_n = 1\} \\ \vdots \\ P\{X_n = m-1\} \end{pmatrix}^T \begin{pmatrix} p_{0,0} & p_{0,1} & \cdots & p_{0,m-1} \\ p_{1,0} & p_{1,1} & \cdots & p_{1,m-1} \\ \vdots & \vdots & \ddots & \vdots \\ p_{m-1,0} & p_{m-1,1} & \cdots & p_{m-1,m-1} \end{pmatrix} = \boldsymbol{\pi_n}\mathbf{P} \qquad (5)$$

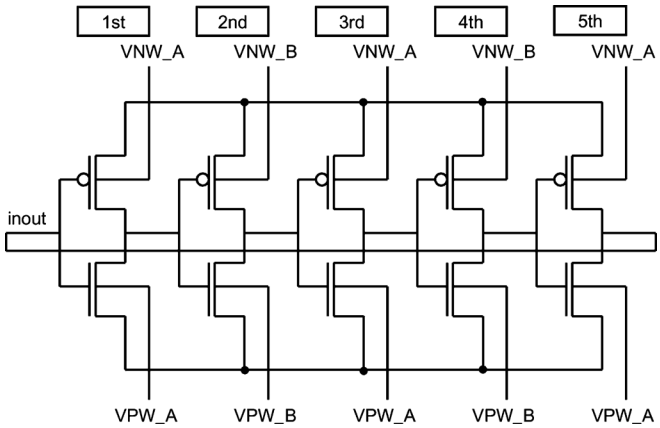

Fig. 7. Example of adjustment of duty cycle with body-biasing technique.

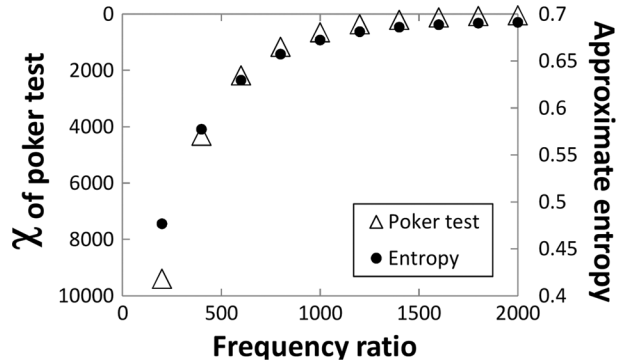

Fig. 8. χ of poker test versus approximate entropy.

simulations. Four-, 64-, 512- and 4096-frequency-dividers were also implemented as slow oscillators.

A body biasing technique was adopted to finely tune the duty cycle of fast oscillators. Although a frequency divider can adjust the duty cycle, a perfect duty cycle of 50% does not necessarily result in a balanced occurrence of 1/0 due to the input offset of the sampler. Fig. 7 has an example of duty cycle adjustment when four body voltages (VNW_A, VNW_B, VPW_A, and VPW_B) are applied to every other inverter in the 5-stage RO. The time when **inout** is high depends on the delay of the NMOSs in the 1st, 3rd, and 5th inverters, and the PMOSs in the 2nd and 4th inverters. The time for low is complimentarily affected by the other MOSs. The duty cycle increases when forward biases are applied to VNW_A and VPW_B, and reverse biases are applied to VNW_B and VPW_A, so that the time for high increases and the time for low decreases. Thus, the duty cycle can be freely chosen by changing the four voltages. This work assumes that the body voltages are provided separately and isolated from VDD, which means deterministic VDD noise does not affect body voltages.

B. Metric of Randomness

We mainly employed the results of a poker test as a randomness metric in this research, because they can easily be calculated in the worst case evaluations of randomness presented in Section IV. However, several randomness tests have been proposed such as the NIST test suite, Diehard tests and FIPS140-2 tests. Even though the NIST tests and Diehard tests are preferred for testing whether test data are sufficiently random or not, they are difficult to use in comparing multiple test streams because they return many p-values as scores for a test stream. Therefore, entropy of a bit stream is widely used for evaluating randomness variations or differences. Fig. 8 compares a poker test and an entropy test. The poker test returns χ values as results, and a pass mark of χ is $2.16 < \chi < 46.17$. The vertical axis for the χ value is inversed because smaller χ indicates higher randomness. It can be seen from the figure that the χ of the poker test is well correlated with the entropies, which indicates that the poker test can be used for approximate evaluations of randomness.

C. Validation With Poker Test

We generated 100 sequences of 20-k random bit streams with the test chip and measured them with a logic analyzer. We also generated the same number of bits using the proposed model. The size of state space was set to 100. The other parameters for the model were determined as follows. We measured the periods of the slow oscillator (the 251-stage ring oscillator with 64-frequency divider) with a real time oscilloscope, and then estimated the variance constant from the measured periods. In this measurement, the fast oscillator was stopped. The trigger jitter of the employed real-time oscilloscope (Tektronix DPO70804) is 1 $\mathrm{ps_{rms}}$, and it is negligibly small because the measured jitter of 251-stage ring oscillator with 64-frequency divider was 113 ps. The estimated variance constant was 2.6×10^{-14} s[1]. On the other hand, since it is difficult to directly measure the signal of the fast oscillator, we obtained the average period from circuit simulation and estimated the jitter of the fast signal assuming that the fast and the slow oscillators have the identical variance constant. Namely, the variance of the fast signal was calculated as the variance constant multiplied by the average period which was from circuit simulation. For the same reason, the duty cycle of the fast oscillator was estimated from measurements assuming that 1/0 probability represented the duty cycle [13], instead of direct waveform measurement. Strictly speaking, the probability of '1' occurrence has a little difference from the actual duty cycle because of the input offset of the sampler. On the other hand, they are highly correlated, and then we regarded the probability of '1' occurrence of the output bit streams as the duty cycle of the fast oscillator.

Fig. 9 plots the measured and simulation results for the poker test with 5- and 15-stage fast oscillators. The horizontal axis plots the frequency ratio of the oscillators, and it was varied by changing the configuration of the frequency divider. The duty cycle for the fast oscillators were adjusted to within $50 \pm 3\%$ by body biasing. We can see from the results for both simulations and measurements in Fig. 9 that increasing sampling sparseness s, which means that the sampler captures data once per s rising edges of the clock, viz., enlarging the jitter of the slow oscillator [15], improves the quality of random bit streams. The χ

[1]The variance constant is different from that used in the other section, since the measured slow oscillator consisted of the low-leakage standard cells while the oscillators in other section consisted of the standard cells with minimum channel length.

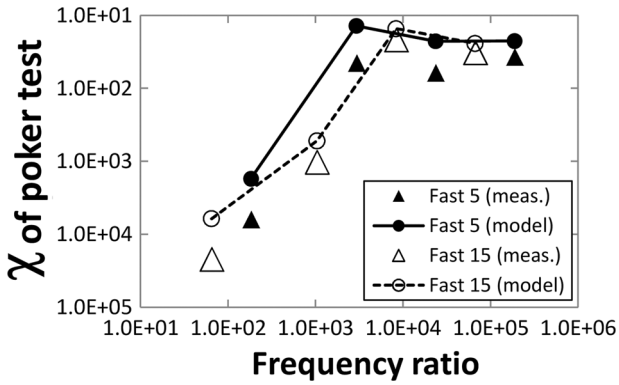

Fig. 9. Randomness versus sampling sparseness and fast average period.

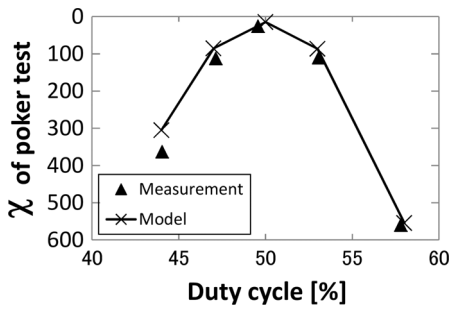

Fig. 10. Randomness versus duty cycle.

value for the 5-stage ring oscillator estimated by the model satisfies the pass mark when the frequency ratio is 2933 and higher. On the other hand, the frequency ratio of 8310 is necessary for the 15-stage ring oscillator. Thus, the fast oscillator with higher frequency reduces the frequency ratio to pass the test, and consequently increases the throughput of the TRNG.

Fig. 10 plots the poker test results obtained by changing the duty cycle for the fast oscillators. We used the 7-stage ring oscillator as the fast oscillator and employed the 512-frequency-divider. In this experimental configuration, the frequency ratio between the oscillators was large enough to pass the poker test. The duty cycle for the fast oscillator varied from 44% to 58% The same figure indicates that the unbalanced duty cycle for the fast oscillator degrades randomness. The results from the simulations and the measurements are well correlated, which means the results from analysis using the proposed model are valid. Also, this result exemplifies that a fast oscillator with unbalanced duty cycle limits the randomness of a TRNG even when the frequency ratio is large enough.

IV. Estimation of Worst-Case Randomness

This section proposes a method of evaluating the worst χ value of a poker test under deterministic noise that utilizes a Markov model but does not require any bit generation.

A. Consideration of Deterministic Noise

Deterministic noise (e.g., power-supply noise, substrate noise, and external noise) induces deterministic fluctuations in the rise timings of oscillators, and it appears as variations of μ in (3) and representative phases x_0. Here, the representative

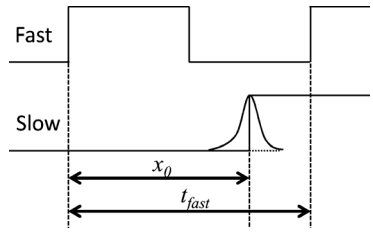

Fig. 11. Example of representative phase.

phase is defined as the time interval from the rising edge of a fast oscillator to the rise timing of a slow oscillator immediately after the fast oscillator without any random jitter, and then $0 \leq x_0 < t_{\rm fast}$. Assuming that jitter has a Gaussian distribution, the representative phase is equal to the average of $T_{\rm diff}$ in Fig. 4. Fig. 11 illustrates the representative phase for oscillating signals.

A state that contains a representative phase corresponds to the mode of states. For example, $n = 1$ in Fig. 5 indicates that the mode value is 65 and the representative phase is $t_{\rm fast} \times 64/100 \leq x_0 < t_{\rm fast} \times 65/100$. Further, the 1/0 occurrence of a bit stream is the most biased where the representative phases of cycles are fixed to the same value, which is the center number of high/low states. Thus, the worst case under deterministic noise can be considered to be a condition where every representative phase is fixed at the middle state of the low states (duty cycle < 0.5) or high states (duty cycle > 0.5). This middle state will be denoted by $s_{\rm middle}$ after this. The proof can be found in the Appendix I.

The discussion in this section focuses on the class of harmful noise rather than waveform shapes and how to deliver such harmful noise to TRNG. Our estimation of the worst-case randomness considers the theoretically worst situation without investigating waveform shapes of the deterministic noise. On the other hand, it is difficult to associate the worst case with the physical attacking method since the noise delivery through physical attacking is totally dependent on the implementation. The mapping of the worst-case to attacking way is another interesting topic to study and one of our future works.

B. Worst Evaluation of χ

Assuming the worst case discussed above, we estimated the worst χ value under deterministic noise using the Markov model. Additional constraints were given to calculations of the transition matrix and state probability vector to estimate the worst χ value.

First, initial vector π_0 in (5) is set so that the probability of state $s_{\rm middle}$ is 1 and the probabilities of the other states are 0. For example, when states 0 to 29 are low and states 30 to 99 are high $(m = 100)$, the initial probability of state 64 (or state 65) is 1 and the others are 0. The representative phases with this constraint are fixed and the bias of 1/0 occurrence becomes the largest, which makes the randomness of output the lowest. Then, μ in (3) is fixed to 0 and does not depend on the periods of oscillators, which means that the representative phase does not change cycle by cycle.

The reason why bit generation is not required is that the temporally-successive probabilities of '1' occurrence can be directly calculated using the proposed model. The computation of χ requires the four successive probabilities of '1' occurrence, $(p_n, p_{n+1}, p_{n+2}, p_{n+3})$, where the probabilities depend on the state at $n-1$, i_{n-1}. On the other hand, from the discussion in Section IV-A, for the worst case evaluation, i_{n-1} can be fixed to s_{middle}. In this case, $(p_n, p_{n+1}, p_{n+2}, p_{n+3})$ can be subsequently computed. Note that the representative phase is fixed in the transition matrix computation for the worst case evaluation. Consequently, the consideration of the worst case enables the worst χ value evaluation without bit generation.

Once the state probability vector is computed, the worst χ value can be directly calculated with the proposed model without bit generation. The probabilities of '1' occurring at successive outputs, $p_1, p_2, \ldots$, are calculated with the corresponding state probability vectors and duty cycle. Note that p_n is not independent of $p_1, p_2, \ldots p_{n-1}$ and the correlation with the past bit stream is taken into consideration in calculating the state probability vector. The worst χ is computed from [7] as:

$$\chi = \frac{16}{5000} \times \sum_{i=0}^{15} (5000 \times \xi_i)^2 - 5000, \tag{6}$$

where ξ_i is the probability that the four successive bits will be equal to i. For instance, ξ_{10} is the probability of $(1010)_2$ and is described as $p_4(1-p_3)p_2(1-p_1)$. The ξ_i can be calculated with the Markov model from the probabilities of the occurrences of '1' or '0', which differs from the conventional way of counting each i in long generated bit sequences.

C. Corrector Considerations

To estimate the worst χ with a corrector, the probabilities of '1' occurring after correction, p'_n, need to be computed from p_n. The p'_n can be computed as $p'_n = p_{2n-1}p_{2n} + (1-p_{2n-1})(1-p_{2n})$ for the XOR corrector, whereas the Von Neumann corrector is difficult to apply since it may discard bits boundlessly and computing p'_n is not easy.

V. Validation of Worst Case-Aware Design

We experimentally confirmed that the proposed worst case computation guaranteed the worst χ with gate-level TRNG simulations by taking power-supply noise into account. Here, the discussion on attacks is not supported by measurement, since it is difficult to accurately inject the noise that we want to give due to measurement environment. For example, though on-chip noise generator [4] can generate power supply noise, it is difficult to control the waveform of the noise due to the distortion by the packages, the bonding wires, and the decoupling capacitances. On the other hand, the simulation is preferable to the chip measurement for analyzing the impact of deterministic noise, because we can inject the noise that we want to give. We therefore have implemented the gate-level TRNG simulator, which enabled us to flexibly control the waveform, frequency, and amplitude of the injected noise.

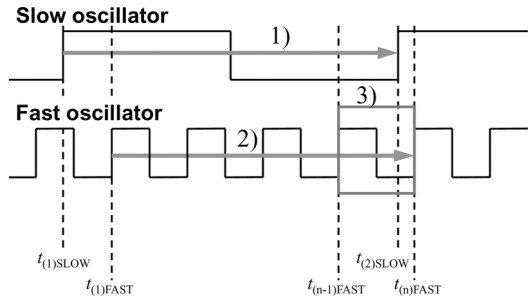

Fig. 12. Concept behind noise-aware gate-level simulation.

A. Simulations Considering Deterministic Noise

A gate-level simulator that took into consideration fluctuations in all gate delays was developed. Each gate delay is denoted as $t_{d,(gate)}$.

$$t_{d,(\mathrm{gate})}(t) = t_{d,\mathrm{offset},(\mathrm{gate})}(\mathrm{Vdd}(t)) + t_{d,\mathrm{random}}, \tag{7}$$

$$t_{d,\mathrm{offset},(\mathrm{gate})}(t) = \frac{a_{(\mathrm{gate})}}{(\mathrm{Vdd(t)} - \mathrm{Vth}_{(\mathrm{gate})})^{\alpha_{(\mathrm{gate})}} + b_{(\mathrm{gate})}}, \tag{8}$$

where (gate) denotes the types of gates. The $t_{d,\mathrm{offset},(\mathrm{gate})}$ is the gate delay without any random noise. To express the dependence of delays on supply noise, we used a gate-delay model ((8)) based on an alpha-power law MOSFET model [18]. Parameters $a_{(\mathrm{gate})}, b_{(\mathrm{gate})}, \alpha_{(\mathrm{gate})}$, and $\mathrm{Vth}_{(\mathrm{gate})}$ are obtained by fitting them to the results from circuit simulations. $\mathrm{Vdd}(t)$ represents the function of a noise-induced supply voltage waveform. The $t_{d,\mathrm{random}}$ represents a random timing fluctuation originating from random noise, and it is calculated as Gaussian random number whose average is zero and variance is $r \times t_{d,\mathrm{offset},(\mathrm{gate})}$ where r is the variance constant of the oscillator.

Fig. 12 explains three-step bit generation, denoting the first rise timings of the fast and the slow ROs as $t_{(1)\mathrm{FAST}}$ and $t_{(1)\mathrm{SLOW}}$ and the timings of n-th rising edges as $t_{(n)\mathrm{FAST}}, t_{(n)\mathrm{SLOW}}$. 1) Calculate the next timing for the rising edge of slow RO $t_{(2)\mathrm{SLOW}}$ from current rising timing $t_{(1)\mathrm{SLOW}}$. 2) From $t_{(1)\mathrm{FAST}}$ and $t_{(1)\mathrm{SLOW}}$, find $t_{(n)\mathrm{FAST}}$ that satisfies equalities $t_{(n-1)\mathrm{FAST}} < t_{(2)\mathrm{SLOW}} < t_{(n)\mathrm{FAST}}$. 3) Generate one bit from $t_{(2)\mathrm{SLOW}}, t_{(n-1)\mathrm{FAST}}$, and $t_{(n)\mathrm{FAST}}$ taking into account the duty cycle of the fast RO.

The time interval between the successive rising edges is the sum of $t_{d,(\mathrm{gate})}(t)$ for two rounds of the slow oscillator. Additionally, when a frequency divider is used for the slow oscillator and the sampling sparseness is s, the gate delays for $2\,s$ rounds are summed.

B. Simulation Results

Randomness under supply noise of various frequencies was evaluated with the simulator, and compared to the worst randomness estimated with the Markov model.

Fig. 13 plots the poker test results (a) without any deterministic noise and under power-supply noise. The figure also plots

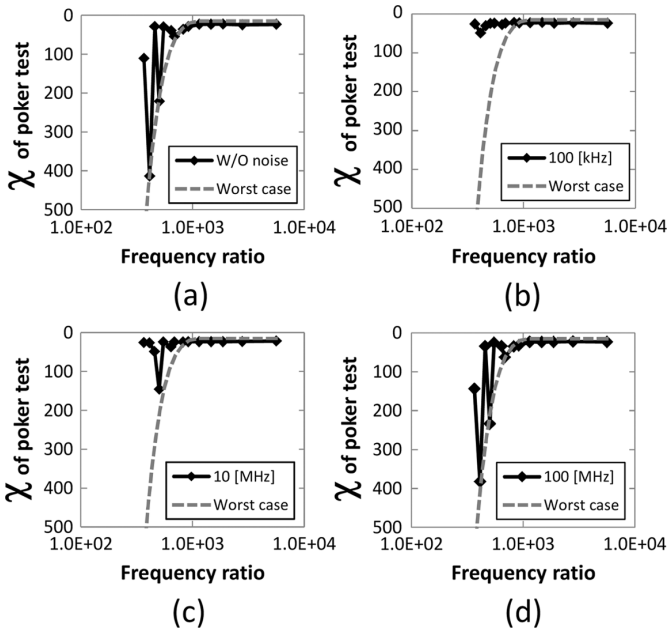

Fig. 13. Evaluation of randomness under and without deterministic noise.

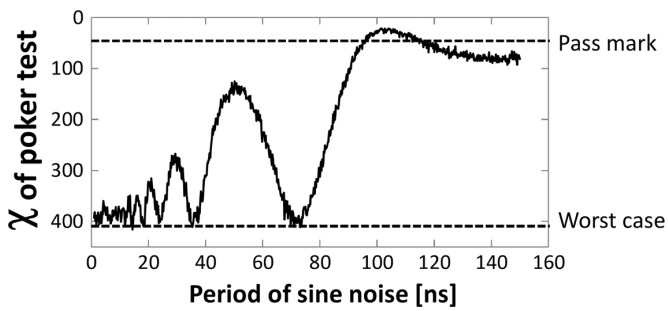

Fig. 14. Evaluation of randomness under deterministic noise with various periods.

the worst χ values estimated with the Markov model. We generated 100 sequences of 20-k bits for the test. (b) One hundred kilohertz, (c) 10 MHz, and (d) 100 MHz of sinusoidal noise whose amplitudes were 100 mV were added to the DC supply voltages of 1.2 V for ROs. We used a 5-stage RO whose average period was 178.3 ps and duty cycle was 51% at 1.2 V as a fast RO. The r of the ROs was 1.77×10^{-14} s referring to the results obtained from measurements of ROs fabricated with the 65-nm process. Equivalent jitter for the Markov model was calculated with r and the periods of ROs. Fig. 13 indicates that randomness depends on the frequency of deterministic noise. It can also be seen that randomness under or without deterministic noise is not worse than the results estimated as being the worst case, which verifies the idea of the worst χ evaluation in Section IV. The χ values in Fig. 13 fluctuate for the low frequency ratios. The fluctuation is caused by μ in (3) and the deterministic noise, since they shift the representative phases x_0. Therefore, the impact of the deterministic noise on the χ value varies depending on the frequency ratio and the noise frequency.

Fig. 14 shows the χ of a poker test when the period of sinusoidal noise was finely varied. The figure also shows the worst χ value and the pass mark for the poker test. Ten sequences of 20-k bits were generated with the simulation. Five-stage RO, whose duty cycle was 51%, and 251-stage RO with a 9-frequency-divider, whose average period was 73.4 ns, were used as the fast and slow ROs. The r of ROs was 1.77×10^{-14} s. It can be seen that the pass/fail for the poker test depends on the period of deterministic noise. In addition, Fig. 14 indicates that the χ values can approach the worst case especially as the period of power-supply noise decreases. Thus, the risk that deterministic noise will degrade randomness to the worst case should not be ignored, and therefore, the proposed worst-case-aware design methodology effectively guarantees randomness even under unwanted noise.

VI. Exploration of Design Space With Proposed Model

This section presents an example to illustrate how to derive appropriate design parameters with the proposed worst-case-aware design methodology. Here, the design space consists of the frequencies of oscillators, the duty cycle of a fast oscillator, and whether the TRNG employs correctors. The design space is explored by evaluating the χ of a poker test when changing the design parameters within feasible values. In Section VI-A, only frequencies of oscillators are explored as a simple example. The allowable shift of duty cycle from 50% can be investigated, though it is not included in this example. Then, Section VI-B shows how to consider the XOR corrector. Section VI-C demonstrates that our design method can consider the injection locking attack by changing variance constant. Section VI-D discusses the required size of state space for meaningful evaluation as a supplement.

A. Design of Fast and Slow Oscillators

This subsection explains the design of fast and slow oscillators for TRNG with given design constraints and circuit information on the 65-nm CMOS process. The variance constant of each gate r is 1.77×10^{-14} s, which was derived from the measurements of ROs in the 65-nm process. Since it is self-evident that the most advantageous duty cycle for the fast oscillator, which is equal to the 1/0 probability here, is 50%, the duty cycle does not need to be explored. The actual duty cycle of the fabricated oscillator has some error from the optimal value due to process variations. Therefore, the duty cycle of the fast RO is within $50 \pm 0.05\%$ here. To simplify the discussion, no correctors have been employed. Ten million bits per second, which is a typical value in a smart card [17], or higher throughput, are required.

First, the periods for several oscillators that could be used as fast ROs were estimated by simulating the circuits. Here, fast oscillators with different numbers of stages (3, 5, and 7) were evaluated for the sake of simplicity, and their periods corresponded to 113.5 ps, 178.3 ps, and 243.2 ps, respectively. Second, the worst χ values for TRNGs with each of the fast ROs were evaluated with the Markov model varying the frequencies of slow ROs.

Fig. 15 plots the estimated χ values. The duty cycle was set to 50.05% assuming the least preferable case. We estimated throughputs that could be achieved for all fast ROs. Now that the required throughput is 10 Mbps, the number of stages of fast ROs should not exceed five. When a 3-stage RO is adopted, the

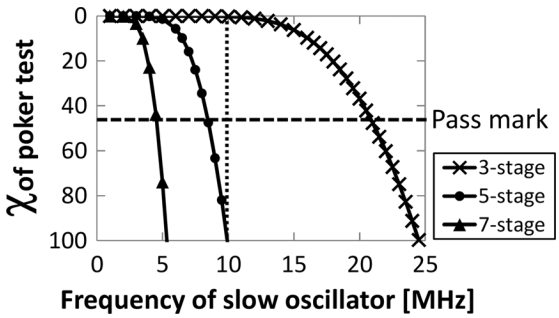

Fig. 15. Evaluation of randomness to design fast and slow oscillators.

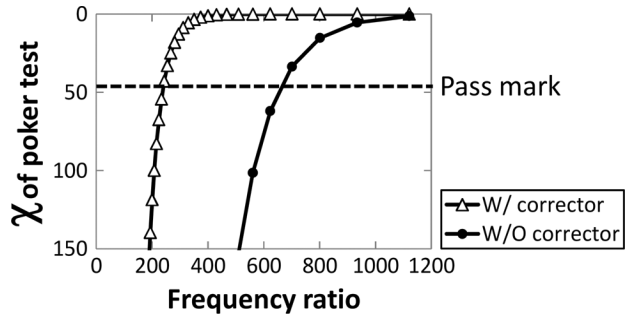

Fig. 16. Improvement in χ value with XOR corrector.

TABLE I
NIST RANDOMNESS TEST RESULTS. p-VALUE/PASS PROPORTIONS ARE LISTED IN EACH CELL. BOLD FONTS ARE USED FOR PASSED TESTS

Test name	21 [MHz]	10 [MHz]
Frequency	**0.1281 / 0.99**	**0.0051 / 0.98**
BlockFrequency	**0.0037 / 0.99**	**0.1626 / 0.99**
CumulativeSums	**0.0805 / 0.99**	**0.0010 / 0.97**
Runs	0.0000 / 0.00	**0.0478 / 0.99**
LongestRun	0.0000 / 0.04	**0.1917 / 0.96**
Rank	**0.3669 / 0.98**	**0.8514 / 1.00**
FFT	0.0000 / 0.70	**0.9781 / 0.99**
NonOverlappingTemplate	0.0000 / 0.00	**0.0060 / 0.98**
OverlappingTemplate	0.0000 / 0.00	**0.0118 / 0.99**
Universal	0.0000 / 0.03	**0.3345 / 0.98**
ApproximateEntropy	0.0000 / 0.00	**0.6163 / 0.99**
RandomExcursions	0.0000 / 0.95	**0.1088 / 1.00**
RandomExcursionsVariant	**0.0106 / 0.98**	**0.0352 / 1.00**
Serial	0.0000 / 0.00	**0.1453 / 1.00**
LinearComplexity	**0.9241 / 1.00**	**0.1223 / 0.99**

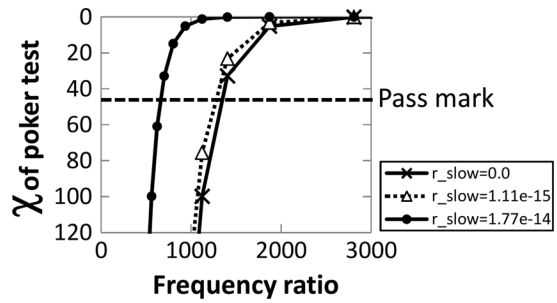

Fig. 17. χ value versus frequency ratio with different variance constants.

number of stages of slow ROs is determined so that the slow oscillator frequency is 20.5 MHz or less. However, for a 7-stage RO, the frequency should not exceed 4.5 MHz, which means more stages, i.e., a larger area is necessary, and furthermore multiple TRNGs are needed to satisfy these requirements.

To further investigate randomness, 100 Mbits of bit streams from the model were evaluated by using the NIST test program. The fast oscillators were 3-stage ROs and the frequencies of the slow oscillators were 10 MHz and 21 MHz. As Fig. 15 shows, the former parameter set achieves sufficient randomness and the latter does not. Table I summarizes the results obtained from the NIST tests. The output bit streams failed nine tests with the 21 MHz slow oscillator, which demonstrates the insufficiency of randomness. However, the 10 MHz slow oscillator attained such a high degree of randomness that it passed all the tests. The efficiency of our design methodology is verified since these results are consistent with those in Fig. 15.

Different oscillator topologies and logic styles, such as the current mode logic for faster ROs, are also explored in actual designs. Here, power consumption, in addition to area, becomes a key performance metric and a more complex design space has to be explored. The proposed evaluation of randomness using the worst χ is effective in terms of CPU time for such purposes.

B. Effect of XOR Corrector

Fig. 16 plots variations in the worst χ with the XOR corrector as the frequency ratio of the oscillators changes. The r is 1.77×10^{-14} s, and the fast oscillators are 5-stage ROs (average period and frequency are 178.3 ps and 5.6 GHz) whose duty cycle is set to 50% Fig. 16 indicates that the XOR corrector improves the estimated χ values. The XOR corrector, however, reduces the throughput of the TRNG by half. The minimum frequency ratios that pass the poker test are 701 without the corrector and 244 with the XOR corrector. Consequently, the throughputs without a corrector and with the XOR corrector correspond to 5.6 Gbps/$701 = 8$ Mbps and (5.6 Gbps/204)/$2 = 11.5$ Mbps. This means that the XOR corrector is effective even when the duty cycle of the fast oscillator is balanced.

C. Injection Locking Attack

Frequency injection [19] is a state-of-the-art attack on oscillator-based TRNGs that utilizes injection locking in ring oscillators to reduce the amount of jitter and degrade randomness. The proposed model can deal with injection locking attacks by decreasing variance constant. In this section, the slow oscillator is injection-locked and its variance constant r_{slow} is reduced while the jitter of the fast oscillator is constant, because an oscillator with high frequency is difficult to be injection-locked. As a preliminary experiment using another small test structure fabricated in the same process, we measured the variance reduction of a ring oscillator under injection locking. A noise generator attacked a 2-input NAND gate in 293-stage ring oscillator. The measurement showed that the variance of the periods decreased to 1/16, and thus we employed 16 as a factor of variance reduction. In addition, we evaluated the randomness of the output when the variance constant for the slow oscillator was zero as an extreme case.

Fig. 17 plots the worst χ as a function of the frequency ratio of oscillators and the three curves correspond to $r_{\mathrm{slow}} = 1.77 \times 10^{-14}$ s, $r_{\mathrm{slow}} (= 1.77 \times 10^{-14}/16) = 1.11 \times 10^{-15}$ s, and $r_{\mathrm{slow}} = 0$ s. Here, the fast oscillator is a 5-stage RO and its duty cycle is 50% Fig. 17 indicates that small r_{slow} requires a

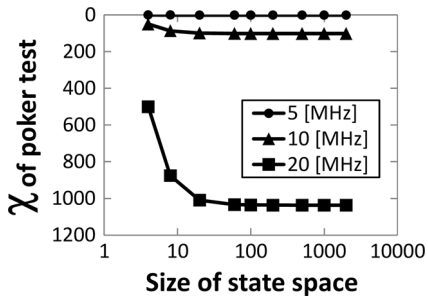

Fig. 18. χ value versus sizes of state space.

low frequency for the slow oscillator to pass the randomness test. If the injection locking reduces $r_{\rm slow}$ from 1.77×10^{-14} s to 1.11×10^{-15} s or 0 s, the throughput decreases by half to sustain sufficient randomness.

Let us examine the throughput reduction above. When the frequency of fast oscillator is constant, the required equivalent jitter to attain sufficient randomness is almost constant. As (4) shows, equivalent jitter, σ_i, originates from slow oscillator, $r_{\rm slow}t_{\rm slow}$, and fast oscillator, $r_{\rm fast}t_{\rm fast}$. When the slow oscillator is under ideal injection locking, the variance constant for slow oscillator becomes zero and $r_{\rm slow}t_{\rm slow} = 0$ while $r_{\rm fast}$ and jitter component from fast oscillator is unchanged. The variance of fast oscillator during a period of slow oscillator, $r_{\rm fast}t_{\rm fast}(N+1-i/m)$, is approximately proportional to frequency ratio, and hence increasing frequency ratio can sustain equivalent jitter. For example, if $r_{\rm slow}t_{\rm slow}$ without injection locking is equal to $r_{\rm fast}t_{\rm fast}(N+1-i/m)$, increasing the frequency ratio by a factor of 2 is reasonable with the slow oscillator under ideal injection locking.

D. Size of State Space

The size of state space m affects the accuracy of the model as explained in Section II-B. Fig. 18 plots the estimated χ values when m is varied. The r is 1.77×10^{-14} s. The fast oscillator is a 5-stage RO and its duty cycle is 50% These are typical settings in our experiments. The frequencies of the slow oscillator are 5, 10, and 20 MHz. Fig. 18 shows that as m becomes larger, the χ value converges, and a large state space is necessary for the conversion when the estimated χ is high (viz., randomness is low). In the range of χ being below 1000, m of 100 enables an approximate estimate of randomness and m of 1000 is sufficient for precise analysis. We therefore employed 100 or 1000 of m in the experiments discussed in this paper.

In the case that m is 100 and the slow frequency is 20 MHz, $t_{\rm span} = t_{\rm fast}/m$ is 1.8 ps, which is $1/23$ of the equivalent jitter σ. This result also suggests a guideline that $t_{\rm span}$ should be less than about $\sigma/25$.

VII. Conclusion

We presented a worst-case-aware design method for oscillator-based TRNGs. A behavioral model of a TRNG and a method of evaluating the worst randomness under deterministic noise were proposed. We confirmed the effectiveness of our model through hardware measurements and comparisons obtained with a gate-level noise-aware TRNG simulator, which was tailored to evaluate randomness under deterministic noise. The proposed design methodology aided us in designing an oscillator-based TRNG that satisfied performance specifications even under a hostile environment.

Appendix A
Mathematical Proof of Worst Case

In Section II, the representative phases of cycles are fixed to the center of the HIGH period (duty cycle > 0.5) or the LOW period (duty cycle < 0.5), to evaluate the worst randomness. We here clarify that such a condition results in the worst entropy, which is a popular metric of randomness.

In this Appendix, t is defined as a time interval from a rising edge of fast oscillator to the next rising edge of slow signal ($0 \le t < t_{\rm fast}$). Assuming a Gaussian distribution, the probability density function of t is

$$f(t) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(t-t_0)^2}{2\sigma^2}\right). \tag{9}$$

where t_0 is the representative phase.

Due to the definition of t, the fast signal is HIGH for $0 \le t < dt_{\rm fast}$ and is LOW for $dt_{\rm fast} \le t < t_{\rm fast}$, where d is the duty cycle of the fast oscillator. Thus, p_n, which is the probability of '1' occurrence at the n-th bit in successive bits, can be calculated by integrating $f(t)$ as follows:

$$p_n = \sum_{l=-\infty}^{\infty} \int_{lt_{\rm fast}}^{lt_{\rm fast}+dt_{\rm fast}} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(t-t_{0n})^2}{2\sigma^2}\right) dt, \quad (l \in \mathbb{Z}) \tag{10}$$

where t_{0n} is the representative phase for the n-th bit. In the following discussion, duty cycle d is $0.5 \le d < 1$, which means that the HIGH period is longer. The same discussion is doable for $0 \le d < 0.5$ with a substitution of $d' = 1-d$. $t_{\rm fast}$ and d are assumed to be independent from n, because their fluctuations are considered as the random jitter or the variation of t_{0n}.

The derivative of p_n with respect to t_{0n} is derived.

$$\begin{aligned}
\frac{\partial}{\partial t_{0n}} p_n &= \frac{1}{\sqrt{2\pi}\sigma} \sum_{l=-\infty}^{\infty} \int_{lt_{\rm fast}}^{lt_{\rm fast}+dt_{\rm fast}} \frac{t-t_{0n}}{\sigma^2} \mathrm{e}^{-\frac{(t-t_{0n})^2}{2\sigma^2}} dt. \\
&= \frac{1}{\sqrt{2\pi}\sigma} \sum_{l=-\infty}^{\infty} \left[\exp\left(-\frac{(t-t_{0n})^2}{2\sigma^2}\right)\right]_{lt_{\rm fast}}^{lt_{\rm fast}+dt_{\rm fast}} \\
&= \frac{1}{\sqrt{2\pi}\sigma} \sum_{l=-\infty}^{\infty} \left\{-\exp\left(-\frac{(lt_{\rm fast}+dt_{\rm fast}-t_{0n})^2}{2\sigma^2}\right) \right. \\
&\quad \left. + \exp\left(-\frac{(lt_{\rm fast}-t_{0n})^2}{2\sigma^2}\right)\right\}
\end{aligned} \tag{11}$$

$(lt_{\rm fast}+dt_{\rm fast}-t_{0n})^2$ is compared with $(lt_{\rm fast}-t_{0n})^2$.

$$\begin{aligned}
&(lt_{\rm fast}+dt_{\rm fast}-t_{0n})^2 - (lt_{\rm fast}-t_{0n})^2 \\
&= -2dt_{\rm fast}t_{0n} + \left(2ldt_{\rm fast}^2 + d^2t_{\rm fast}^2\right).
\end{aligned} \tag{12}$$

When (12) $= 0, t_{0n}$ becomes

$$t_{0n} = t_{\rm fast}\left(l + \frac{d}{2}\right). \tag{13}$$

Here, l is 0 because of the definition $0 < t_{0n} \leq t_{\mathrm{fast}}$, and therefore, $t_{0n} = 0.5dt_{\mathrm{fast}}$.

Under the condition of $t_{0n} < 0.5dt_{\mathrm{fast}}$,

$$(lt_{\mathrm{fast}} + dt_{\mathrm{fast}} - t_{0n})^2 > (lt_{\mathrm{fast}} - t_{0n})^2, \quad (14)$$

$$\int_{lt_{\mathrm{fast}}}^{lt_{\mathrm{fast}}+dt_{\mathrm{fast}}} \frac{t - t_{0n}}{\sigma^2} \exp\left(-\frac{(t - t_{0n})^2}{2\sigma^2}\right) dt > 0, \quad (15)$$

$$\frac{\partial}{\partial t_{0n}} p_n > 0. \quad (16)$$

On the other hand, under the condition of $t_{0n} > 0.5dt_{\mathrm{fast}}$,

$$(lt_{\mathrm{fast}} + dt_{\mathrm{fast}} - t_{0n})^2 < (lt_{\mathrm{fast}} - t_{0n})^2 \quad (17)$$

$$\int_{lt_{\mathrm{fast}}}^{lt_{\mathrm{fast}}+dt_{\mathrm{fast}}} \frac{t - t_{0n}}{\sigma^2} \exp\left(-\frac{(t - t_{0n})^2}{2\sigma^2}\right) dt < 0, \quad (18)$$

$$\frac{\partial}{\partial t_{0n}} p_n < 0. \quad (19)$$

Thus, p_n attains the maximum value with $t_{0n} = 0.5dt_{\mathrm{fast}}$. When t_{0n} is $0.5dt_{\mathrm{fast}}$ for every n, p_n becomes a constant p' irrelevant to n, and obviously $0.5 \leq p' < 1$.

On the other hand, entropy for a bit stream H was defined as

$$H = -p \log p - (1 - p) \log(1 - p), \quad (20)$$

where p is a probability of '1' occurrence across the bit stream. Assuming the condition of $0.5 \leq p < 1$, H becomes the minimum in case that p is the maximum [20]. From these discussion, it is proved that the randomness gets the worst when t_{0n} is always $0.5dt_{\mathrm{fast}}$, namely, the representative phases of cycles are fixed to the middle point of HIGH period.

ACKNOWLEDGMENT

The VLSI chip in this study has been fabricated in the chip fabrication program of VLSI Design and Education Center (VDEC), the University of Tokyo in collaboration with STARC, e-Shuttle, Inc., and Fujitsu Ltd.

REFERENCES

[1] B. Jun and P. Kocher, The Intel random number generator, Cryptography Research Inc., White Paper Prepared for Intel Corporation, Apr. 1999.

[2] M. Bucci, L. Germani, R. Luzzi, A. Trifiletti, and M. Varanonuovo, "A high-speed oscillator-based truly random number source for cryptographic applications on a smart card IC," *IEEE Trans. Computers*, vol. 52, no. 4, pp. 403–409, Apr. 2003.

[3] G. K. Balachandran and R. E. Barnett, "A 440-nA true random number generator for passive RFID tags," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 55, no. 11, pp. 3723–3732, Dec. 2008.

[4] Y. Ogasahara, M. Hashimoto, and T. Onoye, "All-digital ring-oscillator-based macro for sensing dynamic supply noise waveform," *IEEE J. Solid-State Circuits*, vol. 44, no. 6, pp. 1745–1755, Dec. 2009.

[5] C. S. Petrie and J. A. Connelly, "Modeling and simulation of oscillator-based random number generators," in *Proc. IEEE Int. Symp. Circuits and Systems*, May 1996, vol. 4, pp. 324–327.

[6] C. S. Petrie and J. A. Connelly, "A noise-based IC random number generator for applications in cryptography," *IEEE Trans. Circuits Syst. I, Fund. Theory Applicat.*, vol. 47, no. 5, pp. 615–621, May 2000.

[7] Security Requirements for Cryptographic Modules, FIPS Pub. 140-2, May 2001.

[8] A statistical test suite for the validation of random number generators and pseudorandom number generators for cryptographic applications, NIST, Pub. 800-22, May 2001.

[9] R. B. Davies, Exclusive OR (XOR) and Hardware Random Number Generators, Feb. 2002, pp. 1–11 [Online]. Available: http://www.robertnz.net/pdf/xor2.pdf

[10] G. Marsaglia, Diehard Battery of Tests of Randomness, 1995 [Online]. Available: http://stat.fsu.edu/pub/diehard/

[11] F. Bernard, V. Fischer, and B. Valtchanov, "Mathematical model of physical RNGs based on coherent sampling," *Tatra Mountains Math. Pub.*, vol. 45, pp. 1–14, 2010.

[12] M. Baudet, D. Lubicz, J. Micolod, and A. Tassiaux, "On the security of oscillator-based random number generators," *J. Cryptology*, pp. 1–28, 2010, Springer.

[13] S. Ergün, "Modeling and analysis of chaos-modulated dual oscillator-based random number generators," in *Proc. Eur. Signal Process. Conf.*, Aug. 2008, pp. 1–5.

[14] T. Amaki, M. Hashimoto, Y. Mitsuyama, and T. Onoye, "A design procedure for oscillator-based hardware random number generator with stochastic behavior modeling," in *Proc. Int. Workshop on Information Security Applications*, 2010.

[15] D. Schellekens, B. Preneel, and I. Verbauwhede, "FPGA vendor agnostic true random number generator," in *Proc. Int. Conf. Field Programmable Logic and Applications*, 2006, pp. 1–6.

[16] W. Ledermann, *Handbook of Applicable Mathematics*. Hoboken, NJ, USA: Wiley, 1980, vol. 6.

[17] M. Matsumoto, S. Yasuda, R. Ohba, K. Ikegami, T. Tanamoto, and S. Fujita, "1200μ m^2 physical random-number generators based on SiN mosfet for secure smart-card application," in *Proc. IEEE Int. Solid-State Circuits Conf.*, 2008, pp. 414–624.

[18] T. Sakurai and A. R. Newton, "Alpha-power law MOSFET model and its applications to CMOS inverter delay and other formulas," *IEEE J. Solid-State Circuits*, vol. 25, no. 2, pp. 584–594, Apr. 1990.

[19] A. T. Markettos and S. W. Moore, "The frequency injection attack on ring-oscillator-based true random number generators," *Cryptographic Hardware and Embedded Syst.*, pp. 317–331, 2009.

[20] C. E. Shannon, "A mathematical theory of communication," *ACM SIGMOBILE Mobile Comput. Commun. Rev.*, vol. 5, no. 1, pp. 3–55, 2001.

Takehiko Amaki (S'07) received the B.E., M.E., and Ph.D. degrees in information systems engineering from Osaka University, Osaka, Japan, in 2008, 2010, and 2013, respectively.

He is currently with TOSHIBA Corporation. His research interest includes hardware random number generator.

Masanori Hashimoto (S'00–A'01–M'03–SM'11) received the B.E., M.E., and Ph.D. degrees in communications and computer engineering from Kyoto University, Kyoto, Japan, in 1997, 1999, and 2001, respectively.

He has been an Associate Professor with the Department of Information Systems Engineering, Osaka University, Osaka, Japan, since 2004. His current research interests include computer-aided-design for digital integrated circuits and high-speed and low-power circuit design.

Dr. Hashimoto was a recipient of the Best Paper Award at ASP-DAC 2004. He is a member of the Institute of Electronics, Information and Communication Engineers and the IPSJ. He was on the technical program committees of international conferences including DAC, ICCAD, ASP-DAC, DATE, ICCD, and ISQED.

Yukio Mitsuyama (S'97–M'02) received the B.E., M.E., and Ph.D. degrees in information systems engineering from Osaka University, Osaka, Japan, in 1998, 2000, and 2010, respectively.

He was an Assistant Professor with the Graduate School of Engineering, Osaka University. Since 2011, he has been an Assistant Professor with the School of Engineering, Kochi University of Technology, Kami-shi, Japan. His current research interests include reconfigurable architecture and its VLSI design.

Dr. Mitsuyama was a recipient of the Best Paper Award at IEEE ISCE 2004. He is a member of the Institute of Electronics, Information and Communication Engineers and the IPSJ.

Takao Onoye (S'93–M'95–SM'07) received the B.E. and M.E. degrees in electronic engineering and the Dr.Eng. degree in information systems engineering from Osaka University, Osaka, Japan, in 1991, 1993, and 1997, respectively.

He was an Associate Professor with the Department of Communications and Computer Engineering, Kyoto University, Kyoto, Japan. Since 2003, he has been a Professor with the Department of Information Systems Engineering, Osaka University. He has authored or coauthored over 200 research papers in reputed journals and proceedings of international conferences on VLSI design and multimedia signal processing. His current research interests include media-centric low-power architecture and its system-on-a-chip implementation.

Dr. Onoye was a member of the CAS Society Board of Governors from 2008 to 2010, the R10 Student Activities Coordinator from 2010 to 2012, and is now serving as the R10 Treasurer. He is a member of the Institute of Electronics, Information and Communication Engineers, the IPSJ, and the ITE-J.