IEICE TRANS. FUNDAMENTALS, VOL.E96–A, NO.3 MARCH 2013

PAPER

Jitter Amplifier for Oscillator-Based True Random Number Generator*

Takehiko AMAKI[†,††a)], ***Student Member***, **Masanori HASHIMOTO**[†,††], ***and*** **Takao ONOYE**[†,††], ***Members***

SUMMARY We propose a jitter amplifier architecture for an oscillator-based true random number generator (TRNG). Two types of latency-controllable (LC) buffer, which are the key components of the proposed jitter amplifier, are presented. We derive an equation to estimate the gain of the jitter amplifier, and analyze sufficient conditions for the proposed circuit to work properly. The proposed jitter amplifier was fabricated with a 65 nm CMOS process. The jitter amplifier with the two-voltage LC buffer occupied 3,300 μm^2 and attained 8.4x gain, and that with the single-voltage LC buffer achieved 2.2x gain with an 1,700 μm^2 area. The jitter amplification of the sampling clock increased the entropy of a bit stream and improved the results of the NIST test suite so that all the tests passed whereas TRNGs with simple correctors failed. The jitter amplifier attained higher throughput per area than a frequency divider when the required amount of jitter was more than two times larger than the inherent jitter in our test-chip implementations.

key words: *true random number generator, jitter*

1. Introduction

High-quality random number generation is essential for security. True random numbers are produced from physical random sources. All bits in bit streams are independent of the other bits and the probabilities of 1/0 occurrences are identical. Because true random numbers cannot be predicted by computational methods, they are very advantageous for security purposes. For example, they are used as the keys and initial vectors for the cipher block chaining (CBC) mode in common key cryptosystems. Challenge-and-response authentication also requires true random numbers for validation.

On-chip TRNGs have been widely studied [1]–[3] to avoid the need for special hardware to capture random sources. Oscillator-based TRNGs [4]–[9], which utilize the jitter of oscillators as random sources, have been popular on-chip TRNGs. Oscillator-based TRNG can be easily implemented with CMOS gates or FPGA [8], and it is insensitive to 1/f noise and external deterministic interference [7]. However, the amount of internal noise, i.e., jitter is so small that it is very difficult to generate highly random bit streams. Frequency dividers help increase the amount of jitter by accumulating the jitter of the oscillator, but they significantly degrade throughput [4]. Another approach is to be accompanied by a post-processor that consumes additional area and dissipates power, while post-processing does not ensure that random numbers have the sufficient quality.

Note that, jitter in this paper is the random period jitter of an oscillating signal, originating from internal random noise such as thermal, shot, and random telegraph noise. Then, the amount of the jitter can be defined as the standard deviation of periods. In addition, the jitter is assumed to be temporally independent in this paper. Let us show an example of auto correlation function (ACF) of the jitter of an oscillator. A normalized ACF of an 251-stage ring oscillator with 16-frequency divider fabricated in 65 nm CMOS process is shown in Fig. 1. The period of 512 cycles were measured with a real-time oscilloscope, the mean of the periods was subtracted from each measured period, and then the normalized ACF was calculated. Figure 1 shows that the ACF of the jitter was similar to the Dirac delta function, that means the jitter of the ring oscillator can be assumed to be temporally independent. This assumption is used to analyze the behavior of the jitter amplifier in Sect. 2.

This paper proposes a jitter amplifier for an oscillator-based TRNG. Figure 2 illustrates the structure and the operation of a TRNG employing the jitter amplifier. The fast oscillating signal (D in Fig. 2) is sampled with a jittery slow clock whose jitter is amplified with the jitter amplifier (CK in Fig. 2), which results in a random bit stream. Note that the timing jitter relative to the fast oscillator output is the source of the randomness in the TRNG. This timing jitter increases as the period jitter of the slow oscillator, and hence we focus on the period jitter of the slow oscillator and discuss how to amplify it. Though the throughput of the TRNG is unstable since the frequency of the clock for the sampler is slightly but randomly fluctuated, a first-in first-out (FIFO) interface

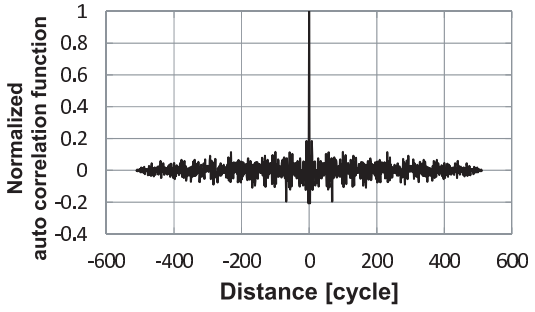

Fig. 1 Normalized auto correlation function of jitter of ring oscillator.

Manuscript received September 7, 2012.

[†]The authors are with the Department of Information Systems Engineering, Graduate School of Information Science and Technology, Osaka University, Suita-shi, 565-0871 Japan.

[††]The authors are with JST CREST, Kawaguchi-shi, 332-0012 Japan.

*A preliminary version of this paper was presented in [13].

a) E-mail: amaki.takehiko@ist.osaka-u.ac.jp

DOI: 10.1587/transfun.E96.A.684

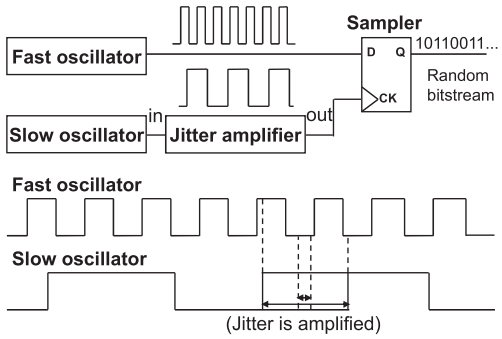

Fig. 2 Oscillator-based TRNG with jitter amplifier.

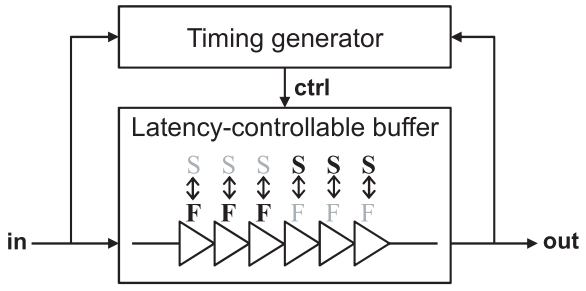

Fig. 3 Block diagram of jitter amplifier.

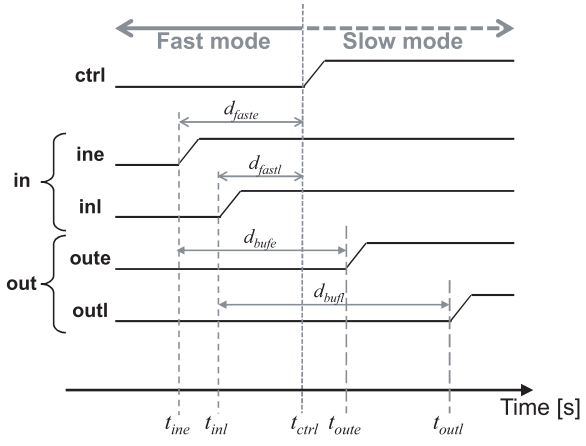

Fig. 4 Timing chart explaining concept behind jitter amplification.

can easily stabilize the throughput. We therefore focus on the randomness of the bitstream from the sampler to clarify the efficiency of the proposed jitter amplifier. To obtain the theoretical substantiation of jitter amplification, the gain of the jitter amplification is analytically estimated. Furthermore, sufficient conditions for proper amplification is analyzed, which helps the jitter amplifier to be integrated with the TRNGs. Two kinds of test chips were fabricated with a 65nm process to validate the proposed amplifier. The measurements demonstrated that the proposed circuit improved randomness with a small increase in area without degrading throughput. To the best of our knowledge, this work is the first to realize the intentional jitter amplification.

The reminder is organized as follows. Section 2 presents the proposed jitter amplifier and analyzes its behavior. Section 3 presents and compares two types of implementations. Section 4 explains the results obtained from measurements. Section 5 concludes the paper.

2. Behavior of Jitter Amplifier

2.1 Concept Behind Jitter Amplification

Figure 3 shows a block diagram of the jitter amplifier. The proposed jitter amplifier consists of an LC buffer and a timing generator. The LC buffer is designed so that each buffer delay t_d could be changed by ctrl from t_{df} to t_{ds}, where $t_{df} < t_{ds}$. That is, the buffer operates in fast mode until the ctrl rise edge arrives, and after that it works in slow mode. Details on the implementations of the LC buffer and the timing generator will be given in Sect. 3.

Figure 4 illustrates the concept behind jitter amplification, where the jitter of the input oscillating signal in is amplified. The timings of rise edges of in fluctuate since the input signal has jitter. To exemplify how the temporally fluctuated signals are processed in the jitter amplifier, let us consider an early rising signal ine and a late rising signal inl, and their outputs oute and outl, respectively. ine rises at t_{ine} and inl rises at t_{inl} ($t_{ine} < t_{inl}$), and oute rises at t_{oute} and outl rises at t_{outl}. The total latencies of the LC buffer for ine and inl are $d_{bufe} = t_{oute} - t_{ine}$ and $d_{bufl} = t_{outl} - t_{inl}$. Here, ctrl rises while in is propagating through the LC buffer, namely, the rise timing of ctrl t_{ctrl} is $t_{ine} < t_{ctrl} < t_{oute}$ and $t_{inl} < t_{ctrl} < t_{outl}$. Note that t_{ctrl} is constant for both t_{ine} and t_{inl} since ctrl is generated by the timing generator which is distinct from the slow oscillator. The time intervals of fast mode are $d_{faste} = t_{ctrl} - t_{ine}$ for ine and $d_{fastl} = t_{ctrl} - t_{inl}$ for inl, where $d_{faste} > d_{fastl}$ from $t_{ine} < t_{inl}$. Longer time for fast mode reduces time for slow mode and results in smaller total latency of LC buffer, and therefore, $d_{bufe} < d_{bufl}$. This means that the later rising edge of in causes larger latency in the LC buffer. In order to validate the jitter amplification, time differences at in and out are compared as follows:

$$\begin{aligned} t_{outl} - t_{oute} &= (t_{inl} + d_{bufl}) - (t_{ine} + d_{bufe}) \\ &= (t_{inl} - t_{ine}) + (d_{bufl} - d_{bufe}) \\ &> (t_{inl} - t_{ine}). \end{aligned} \tag{1}$$

Therefore, the time difference between the early and the late rise timings at in, which indicates the input jitter, is intensified at out by the variable latency of the LC buffer. Thus, the jitter of in is amplified.

2.2 Analysis of Behavior

2.2.1 Preparation

In this section, the behavior of the jitter amplifier is analyzed and an equation to estimate the gain of the jitter amplification is presented. Figure 5 shows a timing chart to explain the behavior of the jitter amplifier. A timing of the n-th ris-

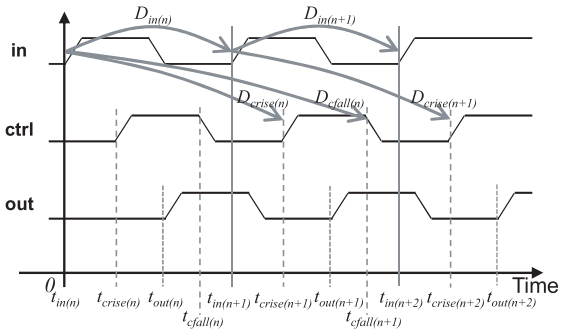

Fig. 5 Behavior of jitter amplifier.

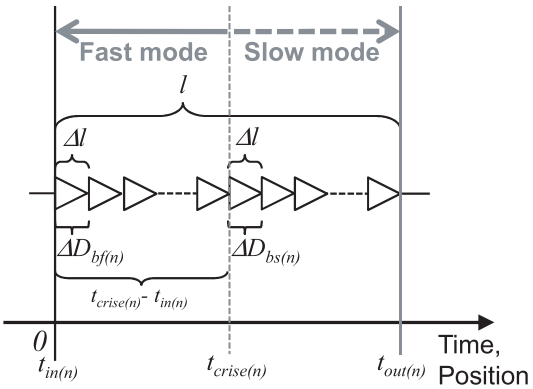

Fig. 6 Fast and slow modes of LC buffer.

ing edge of in is defined as $t_{in(n)} = 0$ ($n \in \mathbb{N}$), and the corresponding rise edges of ctrl and out are $t_{crise(n)}$ and $t_{out(n)}$, where $t_{in(n)} < t_{crise(n)} < t_{out(n)}$. Here, a rise edge of ctrl is generated by the timing generator from a previous edge of in, and $t_{crise(n+1)}$ depends on $t_{in(n)}$. The amount of input jitter given to the jitter amplifier is represented as the standard deviation during the time interval of $t_{in(n+2)} - t_{in(n+1)}$ and output jitter is the standard deviation of $t_{out(n+2)} - t_{out(n+1)}$, and gain of the jitter amplifier is output jitter divided by input jitter. $t_{in(n)}$ needs to be considered to derive the output jitter since $t_{crise(n+1)}$ depends on $t_{in(n)}$ and $t_{crise(n+1)}$ affects the rise timing of out, $t_{out(n+1)}$. Also, timing information before $t_{in(n)}$ is not needed since $t_{in(n+1)}$ and $t_{crise(n+1)}$, which are the necessary timings for deriving the output jitter, are both generated from the same timing $t_{in(n)}$.

The n-th period of in is represented as a stochastic variable $D_{in(n)} = t_{in(n+1)} - t_{in(n)}$. The time interval from the n-th rising edge of in to the $(n+1)$-th rise edge of ctrl is a stochastic variable $D_{crise(n)} = t_{crise(n+1)} - t_{in(n)}$. $D_{in(n)}$ represents the period of the slow oscillator and $D_{crise(n)}$ represents the latency of the timing generator. The stochastic process $\{D_{in(n)}\}$ is assumed to be independent and identically distributed, and its element $D_{in(n)}$ is assumed to be normally distributed. $\{D_{crise(n)}\}$ also assumed to be independent and identically distributed, and $D_{crise(n)}$ is assumed to be normally distributed. The above assumptions are reasonable because the delay elements which construct $D_{in(n)}$ and $D_{crise(n)}$ are fluctuated by the internal noises and the noises are temporarily independent. The assumption is also supported by the measured ACF of the jitter of the ring oscillator in Fig. 1. The mean of $D_{in(n)}$ is μ_{in} and its variance is σ_{in}^2, and the mean of $D_{crise(n)}$ is μ_{crise} and its variance is σ_{crise}^2. That is, $D_{in(n)} \sim N(\mu_{in}, \sigma_{in}^2)$ and $D_{crise(n)} \sim N(\mu_{crise}, \sigma_{crise}^2)$. $D_{in(n_1)}$ is independent of $D_{crise(n_2)}$ for arbitrary n_1 and n_2 ($n_1, n_2 \in \mathbb{N}$) since the slow oscillator is distinct from the timing generator.

Figure 6 shows the behavior of the LC buffer for n-th rising edge of in. Here, let us introduce an analogy that a signal is propagating on a line at a certain speed. The length of the line, that is the distance between start and end points, is l and it corresponds to the length of the LC buffer. Note that the length l is an abstract length rather than a concrete size of the buffer chain. The latencies for a sufficiently small length Δl are $\Delta D_{bf(n)}$ for fast mode and $\Delta D_{bs(n)}$ for slow mode. The stochastic process $\{\Delta D_{bf(n)}\}$ is assumed to be independent and identically distributed, and $\{\Delta D_{bs(n)}\}$ are also assumed to be independent and identically distributed. $\Delta D_{bf(n)}$ and $\Delta D_{bs(n)}$ are assumed to be normally distributed, their means are μ_{bf} and μ_{bs}, and their variances are σ_{bf}^2 and σ_{bs}^2, namely, $\Delta D_{bf(n)} \sim N(\mu_{bf}, \sigma_{bf}^2)$ and $\Delta D_{bs(n)} \sim N(\mu_{bs}, \sigma_{bs}^2)$. This is because the delay elements which constructs $\Delta D_{bf(n)}$ and $\Delta D_{bs(n)}$ are fluctuated by the internal noise, and the noise is temporarily independent. Then, for example, the latency of LC buffer in fast mode is calculated as $l\Delta D_{bf(n)}/\Delta l$. Practically, the amount of jitter is much smaller than the periods, therefore, σ_{bf} and σ_{bs} are much smaller than μ_{bf}[†]. Here, $D_{in(n_1)}$, $D_{crise(n_2)}$, $\Delta D_{bf(n_3)}$ and $\Delta D_{bs(n_4)}$ are independent of each other for arbitrary n_1, n_2, n_3 and n_4 ($n_1, n_2, n_3, n_4 \in \mathbb{N}$).

2.2.2 Gain Derivation

From now, an analytical expression of gain is derived. Firstly, the amount of input jitter is σ_{in} since $t_{in(n+2)} - t_{in(n+1)}$ is $D_{in(n+1)}$.

Here, $\Delta D_{bf(n)}$ is rewritten as $\mu_{bf} + D_{bfr(n)}$, where $D_{bfr(n)} \sim N(0, \sigma_{bf}^2)$. Now that σ_{bf} is much smaller than μ_{bf}, it can be considered as $\left|-D_{bfr(n)}/\mu_{bf}\right| \ll 1$. Though, strictly speaking, $\left|D_{bfr(n)}\right|$ is not always smaller than μ_{bf} due to the normal distribution, the probability of $\left|D_{bfr(n)}\right| \geq \mu_{bf}$ is so small that it can be ignored in actual situations. According to Taylor expansion, $1/\Delta D_{bf(n)}$ can be approximated as follows;

$$\begin{aligned}\frac{1}{\Delta D_{bf(n)}} &= \frac{1}{\mu_{bf}} \frac{1}{1 + \frac{D_{bfr(n)}}{\mu_{bf}}} \\ &= \frac{1}{\mu_{bf}} \left\{ 1 - \frac{D_{bfr(n)}}{\mu_{bf}} + \sum_{k=2}^{\infty} \left(-\frac{D_{bfr(n)}}{\mu_{bf}} \right)^k \right\} \\ &\approx \frac{1}{\mu_{bf}^2} \left(\mu_{bf} - D_{bfr(n)} \right). \end{aligned} \tag{2}$$

[†]If the jitter amount is sufficiently large, the jitter amplification itself is not required.

In order to identify the dominant factors, σ_{bf}^k/μ_{bf}^k and σ_{bs}^k/μ_{bf}^k ($k \geq 2$) are approximated as zeros since σ_{bf} and σ_{bs} are much smaller than μ_{bf}.

Under these conditions, the variance of $t_{out(n+2)}-t_{out(n+1)}$ is calculated as;

$$\mathrm{Var}[t_{out(n+2)}-t_{out(n+1)}] \approx \left\{2(a+1)x^2-2(2a+1)x+(2a+1)\right\}\sigma_{in}^2 + 2\sigma_{bs}^2\left\{\frac{l}{\Delta l}-\frac{\mu_{crise}-\mu_{in}}{\mu_{bf}}\right\}, \quad (3)$$

where $a = \sigma_{ctrl}^2/\sigma_{in}^2$ and $x = \mu_{bs}/\mu_{bf}$ ($a > 0, x > 0$). The detailed derivation can be found in Appendix A. The first term of Eq. (3) shows that the input jitter is magnified by the mechanism explained in Sect. 2.1 This magnification of the input jitter is independent of the rise timing of ctrl. The second means the additional jitter appended during slow mode.

From Eq. (3), the gain of the jitter amplifier is calculated as follows;

$$Gain^2 \approx \left\{2(a+1)x^2-2(2a+1)x+(2a+1)\right\} + 2\frac{\sigma_{bs}^2}{\sigma_{in}^2}\left\{\frac{l}{\Delta l}-\frac{\mu_{crise}-\mu_{in}}{\mu_{bf}}\right\}. \quad (4)$$

Because the second term of Eq. (4) is positive, a sufficient condition for $Gain > 1$ is;

$$2(a+1)x^2-2(2a+1)x+(2a+1) > 1, \quad 0 < x < \frac{a}{a+1}, 1 < x. \quad (5)$$

In actual situations, $Gain > 1$ is attained since μ_{bs} is larger than μ_{bf}, i.e., $x > 1$. In addition, in case of $x > 1$, $Gain$ becomes larger monotonically as x increases. On the other hand, when x is $0 < x < a/(a+1)$, the circuit amplifies the jitter in the inverse way. In this case, μ_{bs} is smaller than μ_{bf}, and then the early rising edge at in rises late at out and the late edge of in rises early at out. Though the analysis suggests such an implementation, the following discussion focuses on $1 < x$.

2.3 Constraints on LC Buffer and Input Signal

In the discussion so far, it is assumed that $t_{in(n)} < t_{crise(n)} < t_{out(n)}$ holds for arbitrary n. In addition, the circuit should be initialized without hindering the amplifying operation. Thus, the timings of ctrl should be appropriately adjusted by the timing generator. However, the precision of the adjustment is limited and furthermore the signals have jitter, which restricts the length of the LC buffer and the frequency of input signal. The conditions for the proposed circuit to amplify the jitter properly are explained here.

In Fig. 5, the timing of the n-th falling edge of ctrl is defined as $t_{cfall(n)}$. The time interval from the n-th in to the $(n+1)$-th fall edge of ctrl, which represents the latency in the timing generator, is $D_{cfall(n)} = t_{cfall(n+1)} - t_{in(n)}$. $D_{cfall(n)}$ is normally distributed, and its mean is μ_{cfall} and its variance is σ_{cfall}^2, that is, $D_{cfall(n)} \sim N(\mu_{cfall}, \sigma_{cfall}^2)$. Here, $D_{cfall(n_1)}$ is independent of $D_{in(n_2)}$, $D_{crise(n_3)}$, $\Delta D_{bf(n_4)}$ and $\Delta D_{bs(n_5)}$ for arbitrary n_1, n_2, n_3, n_4 and n_5 ($n_1, n_2, n_3, n_4, n_5 \in \mathbb{N}$).

The sufficient condition for the proper function is $t_{in(n+1)} < t_{crise(n+1)} < t_{out(n+1)} < t_{cfall(n+1)} < t_{in(n+2)}$. If this condition is satisfied, a rise edge of in propagates through the LC buffer in fast mode firstly, and then propagates in slow mode until the edge goes through the buffer.

In actual design, μ_{rise} and μ_{fall}, which are the average latencies in the timing generator and correspond to μ_{crise} and μ_{cfall}, are discretely controlled rather than continuously. Therefore, μ_{rise} and μ_{fall} are expressed as $\mu_{rise} = \mu_{rise\_offset} + s\Delta\mu_{rise}$ and $\mu_{fall} = \mu_{fall\_offset} + t\Delta\mu_{fall}$ ($s, t \in \mathbb{Z}, s, t \geq 0$), where $\Delta\mu_{rise}$ and $\Delta\mu_{fall}$ represent the adjustment steps of μ_{rise} and μ_{fall}. For example, as will be discussed in Sect. 3.1, our implemented timing generator employs counters whose clock signal is given by internal ring oscillators, and then $\Delta\mu_{rise}$ and $\Delta\mu_{fall}$ are equal to the periods of the clocks. Note that, with $\Delta\mu_{rise} \to 0$ and $\Delta\mu_{fall} \to 0$, the following discussion can be applied to an ideal timing generator which can control μ_{rise} and μ_{fall} continuously.

Here, let us suppose $\mu_{rise\_offset} < \mu_{in} - m\sqrt{\sigma_{rise}^2+\sigma_{in}^2} + (l/\Delta l)\mu_{bf} - m\sigma_{bf}\sqrt{(l/\Delta l)}$ and $\mu_{fall\_offset} < \mu_{in} - m\sqrt{\sigma_{fall}^2+\sigma_{in}^2} + (\mu_{in} - m\sigma_{in})$ are satisfied, as will be derived in Appendix B. These conditions mean that the offsets of the timing generator, μ_{rise_offset} and μ_{fall_offset}, are small enough for the rise and fall edges of ctrl to be adjusted into the proper range. Under these conditions, the sufficient conditions are expressed as the following two equations;

$$\Delta\mu_{rise} + 2m\sqrt{\sigma_{rise}^2+\sigma_{in}^2} < \frac{l}{\Delta l}\mu_{bf} - m\sigma_{bf}\sqrt{\frac{l}{\Delta l}}, \quad (6)$$

$$\Delta\mu_{fall} + 2m\sqrt{\sigma_{fall}^2+\sigma_{in}^2} < (\mu_{in} - m\sigma_{in}) - \left(\frac{l}{\Delta l}\mu_{bs} + m\sigma_{bs}\sqrt{\frac{l}{\Delta l}}\right). \quad (7)$$

The derivations will be presented in Appendix B. Here, a coefficient m ($m > 0$) is introduced to bound the normal distribution. To be more precise, the upper bound of a normal distribution $N(\mu, \sigma^2)$ is defined as $\mu + m\sigma$ and the lower bound is $\mu - m\sigma$. Intuitively, Eq. (6) means that the range of $t_{crise(n+1)}$ added by the step of μ_{rise} is smaller than the minimum latency of the LC buffer. Eq. (7) means that the range of $t_{cfall(n+1)}$ added by the step of μ_{fall} is smaller than the minimum time interval between the rise edges of out and the next in. The number of stages of LC buffer is limited because the length of LC buffer is restricted from the Eqs. (6), (7) and l is proportional to the number of stages.

Equations (6), (7) represent the constraints for designing the jitter amplifier. For obtaining a proper jitter amplification, the timings of ctrl rise and fall edges should be controlled by configuring the timing generator, as will be shown in Sect. 3.1. The edges of ctrl, however, cannot be adjusted into the ranges of proper function if Eqs. (6), (7) are not satisfied. Thus, the designer should confirm that the constraints are satisfied in designing the jitter amplifier.

Also, with rearranging the Eq. (7) for μ_{in}, the constraint

on the period of input signal is derived;

$$\Delta\mu_{fall}+2m\sqrt{\sigma_{fall}^2+\sigma_{in}^2}+m\sigma_{in}+\left(\frac{l}{\Delta l}\mu_{bs}+m\sigma_{bs}\sqrt{\frac{l}{\Delta l}}\right)<\mu_{in}. \tag{8}$$

Thus, the input frequency, $1/\mu_{in}$, is limited by Eq. (8).

As an example, the constraints are verified for a jitter amplifier we have implemented, which will be shown in Fig. 12(a). μ_{in} and σ_{in} are from measurement results of a 251-stage ring oscillator with a 64-frequency divider, employing 1.2 V and 0.7 V of supply voltages. $\Delta\mu_{rise}$, $\Delta\mu_{fall}$, σ_{rise}^2, σ_{fall}^2, $l\mu_{bf}/\Delta l$, $l\mu_{bs}/\Delta l$, $l\sigma_{bf}^2/\Delta l$ and $l\sigma_{bs}^2/\Delta l$ are calculated from the measurement results considering that the average and the variance of the latency are proportional to the number of stages. m is set to 3. Then, the left and right terms of Eq. (6) are calculated as 3.7×10^{-9} and 1.5×10^{-8}, and those of Eq. (7) are 3.8×10^{-9} and 4.1×10^{-7}. Consequently, our implemented circuit can amplify the jitter properly.

3. Implementation

3.1 Implementation of Timing Generator

The timing generator is responsible for generating ctrl, and its implementation is illustrated in Fig. 7. When in rises, the edge detector generates a negative pulse, and the 1-bit counter selects the path which the reset signal is delivered (xrst_e/xrst_o). When a negative pulse is generated, a ring oscillator is enabled (en_e/o) and its corresponding counter starts to increment after being initialized. Every time the counter value (cnt_e/o) exceeds predefined values (cstart/cend), a pulse generator produces rise and fall edges. The timing generator has two (even and odd) paths to generate ctrl for every in rise edge because the increment of the counter starts at the rising edge of in and ends after the next rising edge. The mismatch between the even and odd paths due to process variation affects the timing of ctrl rising edge, μ_{crise}. However, the impact on the gain of jitter amplifier is limited, since μ_{crise} affects only the second term of Eq. (4) and the second term can be ignored with large x. Debug signals (dbug) show ctrl when in and out rise. dbug must be

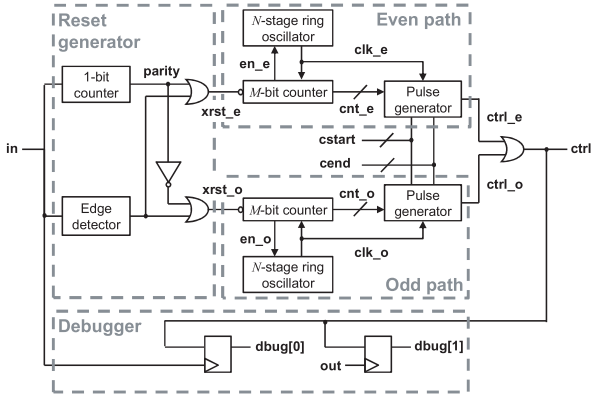

Fig. 7 Block diagram of timing generator we implemented.

2'b10 since ctrl is low at the rising edge of in and high at the rising edge of out for proper function. If dbug is 2'b11, for example, it means that ctrl rises before the edge of in, and therefore the rise timing of ctrl should be delayed by cstart. Thus, the debug signal can be used for adjusting the rise/fall timings of ctrl.

Figure 8 shows the behavior of the signals in the timing generator. When in rises, the 1-bit counter flips parity. If parity is zero, a negative pulse is generated as xrst_e, and the M-bit counter cnt_e is initialized and starts increment. While cnt_e is between cstart and cend-1, the pulse generator makes a pulse ctrl_e. In the same way, when parity is one, xrst_o resets cnt_o, and ctrl_o is produced. Finally, ctrl_e and ctrl_o are ORed and its output becomes ctrl. With properly selected values of cstart and cend, ctrl rises between rise edges of in and out, and falls between out and the next in, namely, Eqs. (6) and (7) are satisfied.

3.2 Implementations of LC Buffers

Various LC buffer implementations are possible that change the element delay depending on the control signal. We present two implementations here that use voltage scaling, i.e., two-voltage and single-voltage implementations. Other implementations of the LC buffer could, e.g., change the loading and/or drivability of buffer elements.

Figure 9 shows the two-voltage LC buffer. The VDD of the buffer (VDBUF) is varied to change the buffer delay from high voltage (VDBUFH) to low (VDBUFL) by using PMOS switches according to ctrl. The jitter gain varies depending on what voltages are selected for VDBUFH and VDBUFL. The sizes of the PMOSs should be determined so that the switching time while VDBUF is changed from VDBUFH to VDBUFL should be sufficiently small comparing to the latency of the LC buffer.

Figure 10 depicts the single-voltage LC buffer. The VDD of the buffer (VDBUF) can be gated from global VDD with PMOS, and can be shorted to the ground by the NMOS transistors. The decoder generates select signals of the multiplexers, where the number of HIGH select signals is determined by the external signal scnum. The delay element, XOR and AND make a short pulse whose width is equal to the delay of the delay element. Each multiplexer passes the input pulse signal when its select signal is HIGH. The ctrl and sc are LOW in fast mode, and VDBUF is close to VDD. When the operation mode is switched to slow mode by the rising edge of ctrl, the HIGH signal is first input to the PMOS, which makes the VDBUF float. Parasitic capacitances connecting to VDBUF are discharged, VDBUF is lowered, and consequently, the buffer element delay increases.

Figure 11 shows simulated waveforms of the single-voltage LC buffer. Parasitic capacitances and resistances of the wire and the diffusion layers were extracted from the layout, and the well capacitances were also taken into account in the simulation. The number of HIGH sc is three. out_buffered is out signal in Fig. 10 after propagating

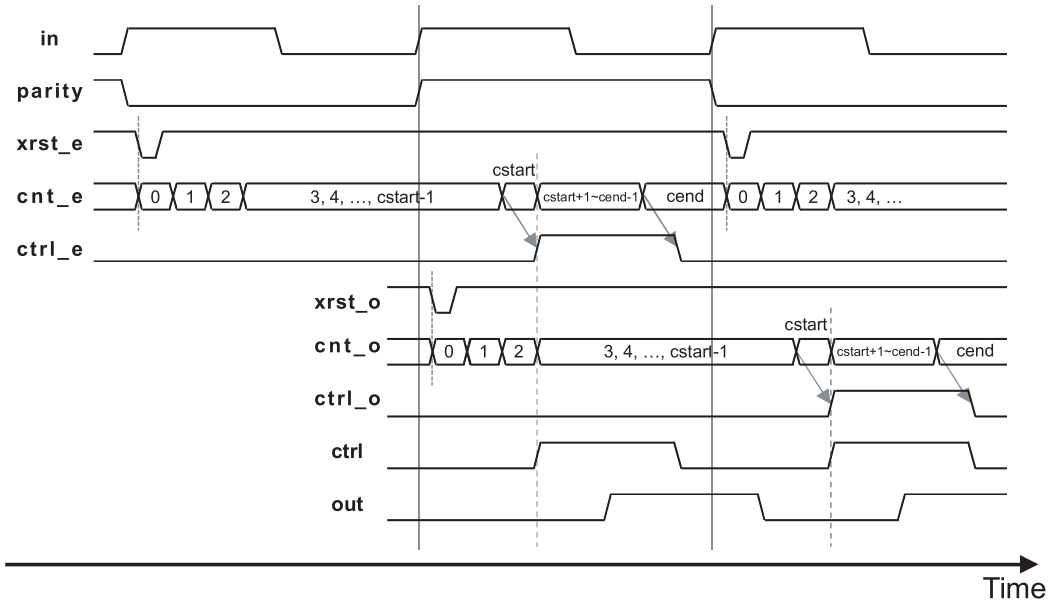

Fig. 8 Timing chart of timing generator. Clock signals (clk_e/o) are omitted.

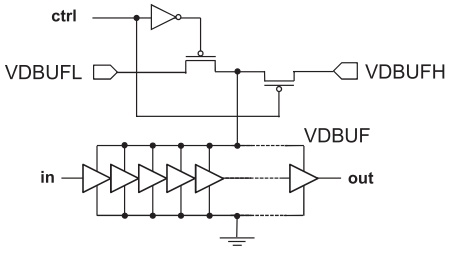

Fig. 9 Implementation of two-voltage LC buffer.

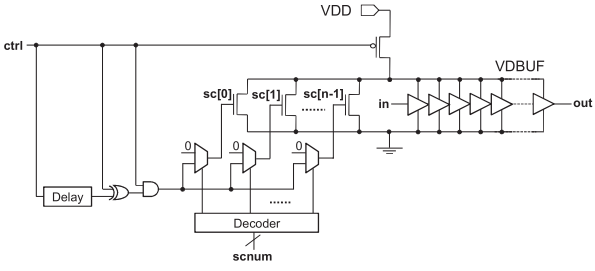

Fig. 10 Implementation of single-voltage LC buffer.

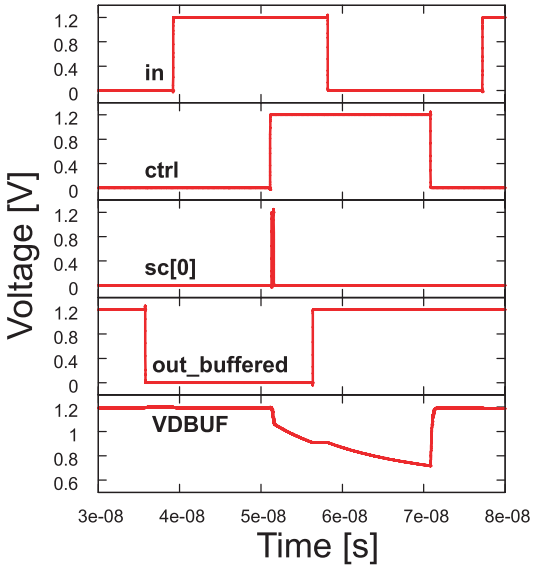

Fig. 11 Waveform example of single-voltage LC buffer.

through a buffer. It can be seen that VDBUF drops when pulses are input to sc, and VDBUF recovers after the fall edge of ctrl. The edges of sc are sharp enough because sufficiently large buffers were inserted after the multiplexers in Fig. 10. In contrast to the two-voltage LC buffer, after VDBUF gets float and dropped, the voltage of the LC buffer is decreasing gradually as the rise and the fall edges of in propagate through the buffer. As Eq. (1) suggests, on the other hand, the single-voltage LC buffer amplifies the jitter when the delay in the LC buffer increases from fast mode to slow mode. Since the discussion in Sect. 2.2 assumes that the average delay in slow mode, μ_{bs}, is constant, the gain estimation with Eq. (4) is not accurate for the single-voltage LC buffer. The jitter gain changes depending on the number of shorted NMOSs, which can be changed by scnum, and how long the duration of sc is, since they affect voltage drop at the beginning of slow mode. The pulse width, which is determined by the delay element in Fig. 10, and the sizes of the switching transistors should be specified considering parasitic capacitance of VDBUF since the time interval during changing VDBUF should be sufficiently small.

Even though the two-voltage LC buffer requires an additional one or two analog pins for VDBUFH and VDBUFL, they can provide stable VDBUF, which makes the estimate of gain Eq. (4) reasonably accurate. The single-voltage LC

buffer, on the other hand, is suitable for low cost implementation since no additional pins are necessary. The jitter gain, however, is difficult to accurately estimate due to the gradual decrease in VDBUF in slow mode. Another issue is the difficulty of predicting the amount of potential drop because it is not easy to accurately estimate parasitic capacitance such as well junction capacitance at the design time.

4. Results from Experiments

4.1 Implementation of Chips

Prototype oscillator-based TRNGs with a two-voltage LC buffer (chip A) and with a single-voltage LC buffer (chip B) were fabricated with a 65 nm CMOS process (Fig. 12). A 31-stage ring oscillator and a 251-stage ring oscillator with a 64-frequency divider were implemented as fast and slow oscillators in chip A. A 7-stage ring oscillator and a 251-stage ring oscillator with a four-frequency divider were the fast and slow oscillators of chip B. The number of frequency division for the slow oscillator in chip B was set to four so that the oscillating frequency became lower than 50 MHz taking into account some safety margin, since the signal with more than 100 MHz cannot be delivered to the outside of the chip due to bonding wire inductance. On the other hand, the slow oscillator in chip A was accompanied with 64-frequency divider, since the internal ring oscillators in the timing generator is slower than that of chip B and then lower frequency was required to achieve jitter amplification[†]. Basically, higher frequency of fast oscillator is desirable for randomness [5], and then 7-stage ring oscillator was selected for chip B. On the other hand, the slow oscillator of chip A was slower as mentioned above and then it had larger intrinsic jitter. To clearly demonstrate the contribution of the jitter amplification to randomness improvement, a slower fast oscillator was selected in chip A. Note that the slower slow oscillator means lower throughput, and hence it is not desirable. For attaining higher throughput, a faster slow oscillator with insufficient jitter should be adopted and in this case jitter amplification becomes necessary to achieve sufficient randomness.

The duty cycles of the fast oscillators, which determined the probabilities of 1/0 occurrences, could be finely adjusted by using body biasing technique [10]. The VDD for the fast oscillators was supplied through a dedicated external pin. A 1,000-stage inverter chain and a 800-stage inverter chain were employed for the LC buffers of chip A and chip B, respectively. The number of stages of LC buffer was determined to satisfy the constraints in Eqs. (6) and (7). Since the slow oscillator of chip B is faster than chip A, the buffer in chip B was set smaller. The areas of the jitter amplifiers were 3,300 μm^2 for chip A and 1,700 μm^2 for chip B. This area difference mainly comes from the implementations of the fast oscillator. In chip A, P-wells of every gate were separated to supply distinct body voltages. On the other hand, the gates share the identical body voltage in chip B, and therefore, the area was smaller than chip A.

In the following, we first evaluate the jitter gain of two implementations of the jitter amplifier with two-voltage and single-voltage LC buffers. On the other hand, now that the implementations of the timing generators and the oscillators as well as the LC buffers are different between the Chip A and the Chip B, the impact of the jitter amplification on the improvement in randomness cannot be directly compared. Therefore, the randomness after the jitter amplifiers will be discussed separately for each chip.

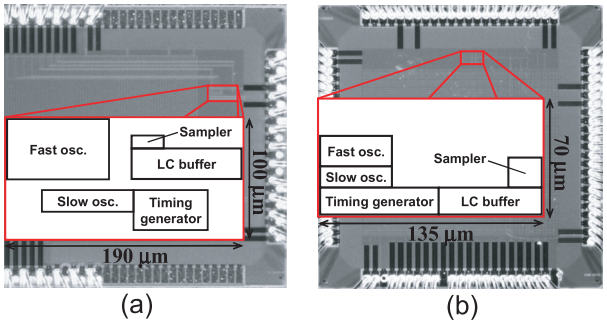

Fig. 12 Chip photos. (a) Chip A employing two-voltage LC buffer. (b) Chip B using single-voltage LC buffer.

4.2 Jitter Gain

The gains of the jitter amplifiers were measured using a real-time oscilloscope. Figure 13 plots the measured jitter gains for the two-voltage LC buffer under different temperatures. The gains estimated with Eq. (4) have also been

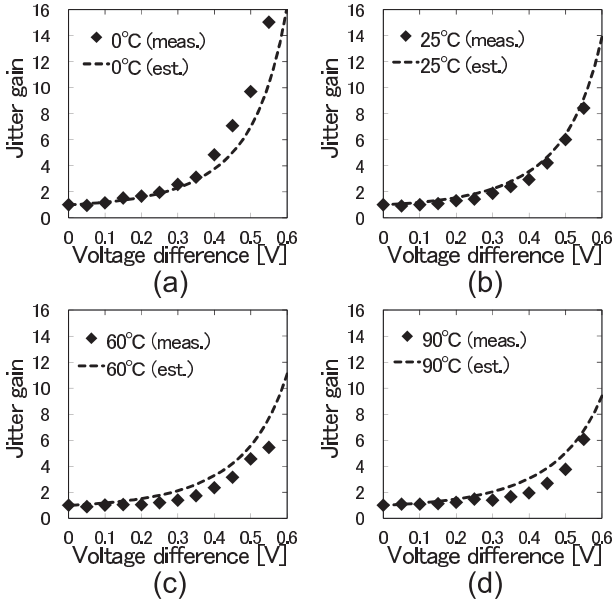

Fig. 13 Jitter gain for two-voltage LC buffer. Temperatures are (a) 0°C, (b) 25°C, (c) 60°C and (d)90°C.

[†]The timing generator of Chip B is the revised version of Chip A, and therefore its internal oscillator was designed faster in order to decrease $\Delta\mu_{rise}$ and $\Delta\mu_{fall}$.

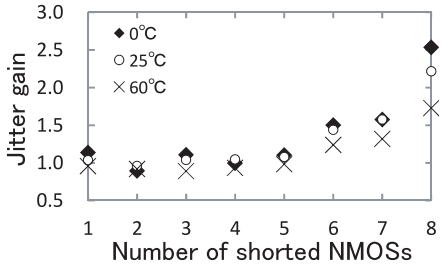

Fig. 14 Measured jitter gain for single-voltage LC buffer.

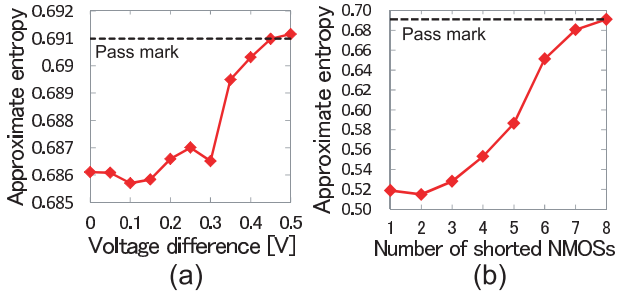

Fig. 15 Approximate entropies. Pass marks (=0.69099) have also been given. Temperature was 25°C. (a) Two-voltage LC buffer. (b) Single-voltage LC buffer.

shown. Here, assuming that the variance of the delay is proportional to the number of gates, a is calculated as the number of gates through which the rising edge of in propagates until the rise edge of ctrl divided by that during a cycle of the slow oscillator. The internal ring oscillator was 83-stage ring oscillator, cstart was 195, and the slow oscillator was 251-stage ring oscillator with 64-frequency divider. Then, a was $\{83 \times 2 \times (195 + 1)\}/(251 \times 2 \times 64) = 1.01$. x was computed from simulation results of 251-stage ring oscillator at various VDDs. For example, because the periods of the ring oscillator were 8.2 ns with 1.2 V of VDD and 62.8 ns with 0.6 V, then x at 1.2 V of VDBUFH and 0.6 V of VDBUFL was $62.8/8.2 = 7.7$. The second term of Eq. (4) is ignored because it gets sufficiently small comparing to the first term as x increases. The X-axis is the difference in voltage defined as VDBUFH - VDBUFL, where VDBUFH is fixed to 1.2 V. We can see that a larger difference in voltage achieves higher gain. This is consistent with Eq. (4), since the larger voltage difference increases x. Also, decreasing temperature increases jitter gain because the sensitivity of the buffer delay to supply voltage becomes larger and x increases at lower temperatures. It attains 8.4 times gain at 25°C. The estimated gain agrees well with the measurements.

Figure 14 shows the measured gain for the single-voltage LC buffer, where the number of shorted NMOSs is varied. The estimation is not shown since, as referred in Sect. 3.2, it is difficult to calculate the gain of the single-voltage LC buffer. Larger numbers of shorted NMOSs yield higher gain of jitter amplification. The gain increases as temperature decreases, which is consistent with the results in Fig. 13. It should be noted that randomness monotonously improves as the jitter of the slow clock increases, and hence the magnitude of jitter amplifier gain is important yet its stability is not required.

4.3 Approximate Entropy

Approximate entropies were calculated for the output bit streams of 16 Mbits measured by a logic analyzer at 25°C, following the NIST SP800-22 [11]. Figure 15 shows the approximate entropies when (a) the difference in voltage in the two-voltage LC buffer and (b) the number of shorted NMOSs in the single-voltage LC buffer were changed. The pass marks for the NIST tests (= 0.69099) have also been plotted, where the entropy of an ideal RNG is $\log_e 2 =$ 0.693. Fast oscillators for (a) and (b) are the 31-stage ring oscillator whose VDD is 0.9 V and the 7-stage ring oscillator whose VDD is 1.2 V, respectively, and their duty cycles are adjusted within 50 ± 0.8%. Figure 15 clearly demonstrates that the proposed jitter amplifiers improve randomness and enable sufficient entropies. The approximate entropy of Fig. 15(a) without jitter amplification (leftmost point) is relatively high compared to Fig. 15(b) and is improved less significantly than (b), because the slow oscillator of (a) had lower frequency and its intrinsic jitter was larger.

Table 1 Setup for NIST randomness tests. The other necessary parameters are automatically decided by the testing program provided by NIST.

Name	Value
Block length for BlockFrequency	20000
Block length for NonOverlappingTemplate	9
Block length for OverlappingTemplate	9
Block length for ApproximateEntropy	10
Block length for Serial	16
Sequence length for LinearComplexity	500
Significance level	0.01
Test data for Runs	20 Kbits × 5000 seqs.
Test data for the other tests	1 Mbits × 100 seqs.

4.4 Comparison with Post-Processors Using NIST Test

We will discuss the advantages of the proposed jitter amplifier here by comparing it with simple post-processors, i.e., a XOR corrector [12] and a von Neumann corrector [9].

After a sufficient number of bits were generated from the TRNG without any correctors or jitter amplifiers, 100 Mbits of streams were obtained with the XOR corrector and von Neumann corrector. The same amount of random bit stream was also generated by the TRNG with jitter amplification with the two-voltage LC buffer. Then, their qualities were evaluated with the NIST test suite. The 31-stage ring oscillator at 0.9 V was the fast oscillator for the TRNG, and its duty cycle was adjusted within 50±0.1%. Depth of the XOR corrector is one. We applied 1.2 V of VDBUFH and 0.7 V of VDBUFL for the LC buffer, and then, the voltage difference was 0.5 V. Temperature was 25°C. Table 1 lists the test parameters we employed, and the parameters satisfy recommendations given in NIST SP800-22. The test gives p-values and pass proportions for each test. If the

Table 2 NIST randomness test results. P-value/pass proportion have been listed in each cell. Bold fonts indicate passed tests.

Test name	Plain	XOR corrector	Von Neumann corrector	Jitter amplifier
Frequency	**0.6993 / 0.99**	0.0000 / 0.00	**0.0270 / 0.99**	**0.1296 / 0.97**
BlockFrequency	**0.0095 / 1.00**	0.0000 / 0.00	**0.8832 / 0.98**	**0.2133 / 0.99**
CumulativeSums	**0.4944 / 0.98**	0.0000 / 0.00	**0.2248 / 0.99**	**0.0032 / 0.97**
Runs	0.0000 / 0.26	0.0000 / 0.02	0.0000 / 0.94	**0.1376 / 0.99**
LongestRun	0.0000 / 0.01	0.0000 / 0.00	0.0000 / 0.91	**0.9558 / 1.00**
Rank	**0.0156 / 1.00**	**0.3838 / 1.00**	**0.1917 / 1.00**	**0.6163 / 1.00**
FFT	**0.8165 / 1.00**	0.0000 / 0.79	**0.7981 / 1.00**	**0.3669 / 0.99**
NonOverlappingTemplate	0.0000 / 0.00	0.0000 / 0.00	0.0000 / 0.15	**0.0072 / 1.00**
OverlappingTemplate	0.0000 / 0.00	0.0000 / 0.00	0.0000 / 0.28	**0.1626 / 1.00**
Universal	0.0000 / 0.00	0.0000 / 0.00	**0.0028 / 0.98**	**0.3041 / 0.99**
ApproximateEntropy	0.0000 / 0.00	0.0000 / 0.00	0.0000 / 0.31	**0.8514 / 0.98**
RandomExcursions	**0.0267 / 1.00**	- / -	**0.0805 / 0.98**	**0.0554 / 1.00**
RandomExcursionsVariant	**0.0190 / 1.00**	- / -	**0.0127 / 0.98**	**0.0909 / 0.98**
Serial	0.0000 / 0.00	0.0000 / 0.00	0.0000 / 0.94	**0.3669 / 0.97**
LinearComplexity	**0.4559 / 1.00**	**0.3838 / 1.00**	**0.3191 / 0.97**	**0.7981 / 0.99**

p-values are 0.0001 or more and the pass proportions are within $(1-\alpha) \pm 3\sqrt{\alpha(1-\alpha)/n_{seq}}$, then it passes the test. Here, α is the significance level and n_{seq} is the number of sequences. From Table 1, the pass range of the pass proportion for runs test is between 0.986 and 0.990, and the pass range for the other tests is between 0.961 and 1.000.

Table 2 summarizes the NIST test results. With neither a corrector nor a jitter amplifier (plain), seven tests failed, which clarified the low randomness for the outputs. The XOR corrector degraded the results for the NIST tests because XOR operation for a poorly random bit stream unbalanced its occurrences of 1/0, and what is worse, the corrector reduced throughput by half. Though the results can be improved with employing the depth of two or more, it unacceptably decreases throughput. Even though the von Neumann corrector increased the number of passed tests, six tests still failed. Additionally, the throughput after the corrector was 0.26 times smaller than that of "plain" in this case, where the reduction in throughput depended on the original bit stream. The jitter amplifier significantly improved the randomness of TRNG output to pass all the tests. Note that the deteriorations in p-values found in the frequency and FFT tests could be ignored since they were within the pass range. In this experimental setup, the voltage difference higher than or equal to 0.5 V was necessary to pass all NIST tests, and below 0.5 V some NIST tests failed. Because the LC buffer kept the sampling frequency of TRNG unchanged, the same throughput as that for "plain" could be achieved. Thus, the jitter amplifier enabled the target TRNG to generate a sufficiently random bit stream without degrading throughput.

4.5 Comparison with Frequency Divider

Dividing the slow oscillator output with a frequency divider, which often consists of serially-connected two-frequency dividers, is a simple solution to obtain large jitter. The two-frequency divider means the simplest asynchronous frequency divider which consists of an inverter and a DFF. 2^n-frequency divider is easily constructed by connecting n two-frequency dividers in series. For example, 16-frequency divider consists of a series of four two-frequency dividers. Though there are many frequency dividers whose factors of divisions are not 2^n, their areas depend on the implementations. Thus, we considered only 2^n-frequency dividers which consist of n two-frequency dividers in this discussion. Here, the p-th period of the slow oscillator is $t_{slow(p)} \sim N(\mu_{slow}, \sigma^2_{slow})$, where μ_{slow} is the average of the slow periods and σ^2_{slow} is the variance. $t_{slow(p)}$ is independent of $t_{slow(p+k)}$, for $k \in \mathbb{Z}$. When the slow oscillator is divided by m_d, the variance is accumulated during m_d cycles of the slow oscillator. And then, the average period of the divided signal is $m_d\mu_{slow}$ and the variance is $m_d\sigma^2_{slow}$. Since the jitter is defined as a standard deviation of the periods, the jitter of the slow oscillator is σ_{slow} and that after a m_d-frequency divider is $\sqrt{m_d}\sigma_{slow}$, and thus the amount of the jitter is multiplied by $\sqrt{m_d}$ with a m_d-frequency divider, whereas the frequency becomes $1/m_d$ times smaller.

The jitter amplifier and the frequency divider are compared here using throughput per area of TRNG as a metric. Let us improve the jitter of a slow oscillator whose area is A_{osc}, frequency is F_{osc}, and jitter is σ_{osc}. When the required jitter is σ_{req}, the required magnification of jitter is $M_{req} = \sigma_{req}/\sigma_{osc}$. To attain M_{req} with two-frequency dividers whose area is A_{div}, $\lceil \log_2 M^2_{req} \rceil$ dividers are necessary. The required area is $A_{osc} + A_{div}\lceil \log_2 M^2_{req} \rceil$ and the throughput is $F_{osc}/2^{\lceil \log_2 M^2_{req} \rceil}$. On the other hand, when a jitter amplifier is used whose area is A_{ja} and achievable jitter gain is G_{ja}, $\lceil M_{req}/G_{ja} \rceil$ amplifiers are necessary. The area is $A_{osc} + A_{ja}\lceil M_{req}/G_{ja} \rceil$ and the throughput is F_{osc}. Here, the throughput per area of the jitter amplifier is divided by that of the frequency divider, and this value indicates whether the jitter amplifier is superior or not. The condition under which the throughput per area of the jitter amplifier is superior to that of the frequency divider is:

$$1 < \frac{F_{osc}}{A_{osc} + A_{ja}\lceil \frac{M_{req}}{G_{ja}} \rceil} \Big/ \frac{F_{osc}/2^{\lceil \log_2 M^2_{req} \rceil}}{A_{osc} + A_{div}\lceil \log_2 M^2_{req} \rceil}. \quad (9)$$

This means the jitter amplifier is superior in throughput per

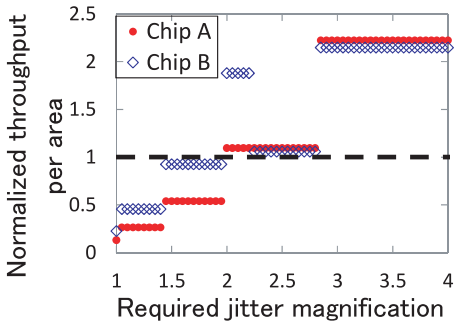

Fig. 16 Normalized throughput per area vs. required magnification.

area, when the right hand term of Eq. (9), called as the normalized throughput per area, is more than one.

Figure 16 has the normalized throughput per area as the required magnification M_{req} changes. The normalized throughput per area was calculated from the right term in Eq. (9) using the parameters value below. For chip A, $A_{osc} = 495.0\ \mu\text{m}^2$, $A_{ja} = 3300.0\ \mu\text{m}^2$, $A_{div} = 8.0\ \mu\text{m}^2$, and $G_{ja} = 8.4$. As for chip B, $A_{osc} = 478.0\ \mu\text{m}^2$, $A_{ja} = 1663.0\ \mu\text{m}^2$, $A_{div} = 8.0\ \mu\text{m}^2$, and $G_{ja} = 2.2$. The areas for the slow oscillator, two-frequency divider, and jitter amplifier are based on the layouts of chips A and B. The gain of the jitter amplifier was set to the largest value observed in the measurement. Figure 16 indicates an oscillator-based TRNG should employ a jitter amplifier when required jitter magnification is larger than 2.0. With this jitter magnification, chip A with the proposed jitter amplifier passed all NIST tests as shown in Sect. 4.4. On the other hand, TRNGs with the same magnification by frequency divider and chip B are assumed to pass NIST tests while NIST tests were not carried out for them. Large jitter magnification is required when designing a good TRNG that generates highly random numbers at high throughput yet occupies a small area. The jitter amplifier is more suitable than the frequency divider for such purposes.

5. Conclusion

A jitter amplifier for an oscillator-based TRNG has been developed with a 65 nm CMOS process. The results from measurements demonstrated that it efficiently amplified the jitter of the sampling signal and enabled a TRNG with high throughput yet a small area with sufficient random bit stream quality. In addition, we tested and confirmed that the jitter amplifier was better than simple correctors and a frequency divider in most cases.

Acknowledgments

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.

This work is supported by SCOPE program of Ministry of Internal Affairs and Communications.

References

[1] F. Pareschi, G. Setti, and R. Rovatti, "Implementation and testing of high-speed CMOS true random number generators based on chaotic systems," IEEE Trans. Circuits Syst. I, Regular Papers, vol.57, no.12, pp.3124–3137, Dec. 2010.
[2] N. Liu, N. Pinckney, S. Hanson, D. Sylvester, and D. Blaauw, "A true random number generator using time-dependent dielectric breakdown," Proc. IEEE VLSI Circuits, pp.216–217, June 2011.
[3] S. Srinivasan, S. Mathew, R. Ramanarayanan, F. Sheikh, M. Anders, H. Kaul, V. Erraguntla, R. Krishnamurthy, and G. Taylor, "2.4 GHz 7 mW all-digital PVT-variation tolerant true random number generator in 45nm CMOS," Proc. IEEE VLSI Circuits, pp.203–204, June 2010.
[4] M. Bucci and R. Luzzi, "Fully digital random bit generators for cryptographic applications," IEEE Trans. Circuits Syst. I, Regular Papers, vol.55, no.3, pp.861–875, April 2008.
[5] 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. Comput., vol.52, no.4, pp.403–409, April 2003.
[6] C.S. Petrie and J.A. Connelly, "A noise-based IC random number generator for applications in cryptography," IEEE Trans. Circuits Syst. I, Regular Papers, vol.47, no.5, pp.615–621, May 2000.
[7] C.S. Petrie and J.A. Connelly, "Modeling and simulation of oscillator-based random number generators," Proc. IEEE ISCAS, vol.4, pp.324–327, May 1996.
[8] K.H. Tsoi, K.H. Leung, and P.H.W. Leong, "High performance physical random number generator," IET Comput. Digit. Tech., vol.1, no.4, pp.349–352, July 2007.
[9] B. Jun and P. Kocher, "The Intel random number generator," cryptography research inc., white paper prepared for Intel corp., April 1999.
[10] T. Amaki, M. Hashimoto, Y. Mitsuyama, and T. Onoye, "A design procedure for oscillator-based hardware random number generator with stochastic behavior modeling," Proc. WISA, Aug. 2010.
[11] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray, and S. Vo, "A statistical test suite for the validation of random number generators and pseudorandom number generators for cryptographic applications," NIST, pub. 800-22, April 2010.
[12] R.B. Davies, "Exclusive OR (XOR) and hardware random number generators," pp.1–11, Feb. 2002. (http://www.robertnz.net/pdf/xor2.pdf)
[13] T. Amaki, M. Hashimoto, and T. Onoye, "An oscillator-based true random number generator with jitter amplifier," Proc. IEEE ISCAS, pp.725–728, May 2011.

Appendix A: Calculation of Output Jitter

In Sect. 2.2, the gain of the proposed jitter amplifier is discussed. We will here detail the calculation of the amount of the output jitter. Note that the definition of the notations are found in Sect. 2.2.

In order to obtain the output jitter, $t_{out(n+2)} - t_{out(n+1)}$ is calculated;

$$\begin{aligned} t_{out(n+1)} &= t_{crise(n+1)} + \left(l - \frac{t_{crise(n+1)} - t_{in(n+1)}}{\Delta D_{bf(n)}}\Delta l\right)\frac{\Delta D_{bs(n)}}{\Delta l} \\ &= D_{crise(n)} + \left(l - \frac{D_{crise(n)} - D_{in(n)}}{\Delta D_{bf(n)}}\Delta l\right)\frac{\Delta D_{bs(n)}}{\Delta l}, \end{aligned} \tag{A·1}$$

$$\begin{aligned} t_{out(n+2)} &= t_{crise(n+2)} + \left(l - \frac{t_{crise(n+2)} - t_{in(n+2)}}{\Delta D_{bf(n+1)}}\Delta l\right)\frac{\Delta D_{bs(n+1)}}{\Delta l} \\ &= D_{in(n)} + D_{crise(n+1)} \\ &\quad + \left(l - \frac{D_{crise(n+1)} - D_{in(n+1)}}{\Delta D_{bf(n+1)}}\Delta l\right)\frac{\Delta D_{bs(n+1)}}{\Delta l}, \qquad (A\cdot 2) \end{aligned}$$

$$\begin{aligned} t_{out(n+2)} - t_{out(n+1)} &= (D_{in(n)} - D_{crise(n)} + D_{crise(n+1)}) \\ &\quad - (D_{in(n)} - D_{crise(n)})\frac{\Delta D_{bs(n)}}{\Delta D_{bf(n)}} \\ &\quad + (D_{in(n+1)} - D_{crise(n+1)})\frac{\Delta D_{bs(n+1)}}{\Delta D_{bf(n+1)}} \\ &\quad - \frac{l}{\Delta l}(\Delta D_{bs(n)} - \Delta D_{bs(n+1)}). \qquad (A\cdot 3) \end{aligned}$$

From Eq. (2), $1/\Delta D_{bf(n)}$ is approximated as follows;

$$\begin{aligned} \frac{1}{\Delta D_{bf(n)}} &\approx \frac{1}{\mu_{bf}^2}\left(\mu_{bf} - D_{bfr(n)}\right) \\ &= \frac{\Delta D'_{bf(n)}}{\mu_{bf}^2}, \qquad (A\cdot 4) \end{aligned}$$

where $\Delta D'_{bf(n)} = \mu_{bf} - D_{bfr(n)}$ follows normal distribution whose mean and variance are μ_{bf} and σ_{bf}^2. Here, $\Delta D'_{bf(n_1)}$ is independent of $D_{in(n_2)}$, $D_{crise(n_3)}$, $\Delta D_{bf(n_4)}$ and $\Delta D_{bs(n_5)}$ for arbitrary n_1, n_2, n_3, n_4 and n_5 ($n_1, n_2, n_3, n_4, n_5 \in \mathbb{N}$). From Eq. (A·4), Eq. (A·3) is approximated as;

$$\begin{aligned} t_{out(n+2)} - t_{out(n+1)} &\approx (D_{in(n)} - D_{crise(n)} + D_{crise(n+1)}) \\ &\quad - \Delta D_{bs(n)}\left\{\frac{1}{\mu_{bf}^2}(D_{in(n)} - D_{crise(n)})\Delta D'_{bf(n)} + \frac{l}{\Delta l}\right\} \\ &\quad + \Delta D_{bs(n+1)}\left\{\frac{1}{\mu_{bf}^2}(D_{in(n+1)} - D_{crise(n+1)})\Delta D'_{bf(n+1)} + \frac{l}{\Delta l}\right\}, \qquad (A\cdot 5) \end{aligned}$$

where the first, second and third terms are called hereafter as (A), (B), and (C).

(A) and $(B + C)$ are not independent of each other since they share the same random variables. Therefore, $\mathrm{Var}[A + B + C] = \mathrm{Var}[A] + \mathrm{Var}[B + C] + 2\mathrm{Cov}[A, B + C]$. The mean and the variance of (A) are;

$$\mathrm{E}[A] = \mu_{in} - \mu_{crise} + \mu_{crise} = \mu_{in}, \qquad (A\cdot 6)$$

$$\mathrm{Var}[A] = \sigma_{in}^2 + \sigma_{crise}^2 + \sigma_{crise}^2 = \sigma_{in}^2 + 2\sigma_{crise}^2. \qquad (A\cdot 7)$$

The mean of $(B + C)$ is;

$$\begin{aligned} \mathrm{E}[B + C] &= -\mu_{bs}\left\{\frac{1}{\mu_{bf}^2}(\mu_{in} - \mu_{crise})\mu_{bf} + \frac{l}{\Delta l}\right\} \\ &\quad + \mu_{bs}\left\{\frac{1}{\mu_{bf}^2}(\mu_{in} - \mu_{crise})\mu_{bf} + \frac{l}{\Delta l}\right\} \\ &= 0. \qquad (A\cdot 8) \end{aligned}$$

(B) is independent of (C) since they do not share the same random variables, and therefore, $\mathrm{Var}[B + C] = \mathrm{Var}[B] + \mathrm{Var}[C]$. From $\mathrm{Var}[XY] = \mathrm{Var}[X]\mathrm{Var}[Y] + \mathrm{E}[X]^2\mathrm{Var}[Y] + \mathrm{E}[Y]^2\mathrm{Var}[X]$ where X and Y are independent of each other, the variance of (B) is calculated as follows;

$$\begin{aligned} \mathrm{Var}[B] &= \mathrm{Var}\left[\Delta D_{bs(n)}\left\{\frac{1}{\mu_{bf}^2}(D_{in(n)} - D_{crise(n)})\Delta D'_{bf(n)} + \frac{l}{\Delta l}\right\}\right] \\ &= \mathrm{Var}[\Delta D_{bs(n)}]\mathrm{Var}\left[\frac{1}{\mu_{bf}^2}(D_{in(n)} - D_{crise(n)})\Delta D'_{bf(n)} + \frac{l}{\Delta l}\right] \\ &\quad + \mathrm{E}[\Delta D_{bs(n)}]^2\mathrm{Var}\left[\frac{1}{\mu_{bf}^2}(D_{in(n)} - D_{crise(n)})\Delta D'_{bf(n)} + \frac{l}{\Delta l}\right] \\ &\quad + \mathrm{E}\left[\frac{1}{\mu_{bf}^2}(D_{in(n)} - D_{crise(n)})\Delta D'_{bf(n)} + \frac{l}{\Delta l}\right]^2 \mathrm{Var}[\Delta D_{bs(n)}]. \qquad (A\cdot 9) \end{aligned}$$

Here,

$$\mathrm{E}\left[\frac{1}{\mu_{bf}^2}(D_{in(n)} - D_{crise(n)})\Delta D'_{bf(n)} + \frac{l}{\Delta l}\right] = \frac{\mu_{in} - \mu_{crise}}{\mu_{bf}} + \frac{l}{\Delta l}, \qquad (A\cdot 10)$$

$$\begin{aligned} &\mathrm{Var}\left[\frac{1}{\mu_{bf}^2}(D_{in(n)} - D_{crise(n)})\Delta D'_{bf(n)} + \frac{l}{\Delta l}\right] \\ &= \frac{1}{\mu_{bf}^4}\mathrm{Var}[(D_{in(n)} - D_{crise(n)})\Delta D'_{bf(n)}] \\ &= \frac{1}{\mu_{bf}^4}\left\{(\sigma_{in}^2 + \sigma_{crise}^2)\sigma_{bf}^2 + (\mu_{in} - \mu_{crise})^2\sigma_{bf}^2 + \mu_{bf}^2(\sigma_{in}^2 + \sigma_{crise}^2)\right\}. \qquad (A\cdot 11) \end{aligned}$$

From Eqs. (A·10), (A·11), Eq. (A·9) is calculated as;

$$\begin{aligned} \mathrm{Var}[B] &= \frac{\mu_{bs}^2 + \sigma_{bs}^2}{\mu_{bf}^4}\left\{(\sigma_{in}^2 + \sigma_{crise}^2)\sigma_{bf}^2 + (\mu_{in} - \mu_{crise})^2\sigma_{bf}^2\right\} \\ &\quad + \frac{(\mu_{bs}^2 + \sigma_{bs}^2)(\sigma_{in}^2 + \sigma_{crise}^2)}{\mu_{bf}^2} + \sigma_{bs}^2\left\{\frac{\mu_{in} - \mu_{crise}}{\mu_{bf}} + \frac{l}{\Delta l}\right\}. \qquad (A\cdot 12) \end{aligned}$$

$\mathrm{Var}[C]$ is also calculated in the same manner;

$$\begin{aligned} \mathrm{Var}[C] &= \frac{\mu_{bs}^2 + \sigma_{bs}^2}{\mu_{bf}^4}\left\{(\sigma_{in}^2 + \sigma_{crise}^2)\sigma_{bf}^2 + (\mu_{in} - \mu_{crise})^2\sigma_{bf}^2\right\} \\ &\quad + \frac{(\mu_{bs}^2 + \sigma_{bs}^2)(\sigma_{in}^2 + \sigma_{crise}^2)}{\mu_{bf}^2} + \sigma_{bs}^2\left(\frac{\mu_{in} - \mu_{crise}}{\mu_{bf}} + \frac{l}{\Delta l}\right). \qquad (A\cdot 13) \end{aligned}$$

Therefore, $\mathrm{Var}[B + C]$ becomes;

$$\begin{aligned} \mathrm{Var}[B+C] &= \frac{2(\mu_{bs}^2 + \sigma_{bs}^2)}{\mu_{bf}^4}\left\{(\sigma_{in}^2 + \sigma_{crise}^2)\sigma_{bf}^2 + (\mu_{in} - \mu_{crise})^2\sigma_{bf}^2\right\} \\ &\quad + \frac{2(\mu_{bs}^2 + \sigma_{bs}^2)(\sigma_{in}^2 + \sigma_{crise}^2)}{\mu_{bf}^2} + 2\sigma_{bs}^2\left\{\frac{\mu_{in} - \mu_{crise}}{\mu_{bf}} + \frac{l}{\Delta l}\right\}. \qquad (A\cdot 14) \end{aligned}$$

In order to obtain the variance of (A)+$(B$+$C)$, the covariance

of (A) and $(B+C)$ is calculated.

$$\begin{aligned}\mathrm{Cov}[A, B+C] &= \mathrm{E}[A(B+C)] - \mathrm{E}[A]\mathrm{E}[B+C]. \\ &= \mathrm{E}[A(B+C)] \end{aligned} \quad (\mathrm{A}\cdot 15)$$

$(A)+(B+C)$ is calculated as follows;

$$\begin{aligned}A(B+C) =& -\frac{\Delta D_{bs(n)}\Delta D'_{bf(n)}}{\mu_{bf}^2}(D_{in(n)}-D_{crise(n)})^2 \\ &-\frac{\Delta D_{bs(n)}\Delta D'_{bf(n)}}{\mu_{bf}^2}(D_{in(n)}-D_{crise(n)})D_{crise(n+1)} \\ &+\frac{\Delta D_{bs(n+1)}\Delta D'_{bf(n+1)}}{\mu_{bf}^2}D_{in(n+1)}(D_{in(n)}-D_{crise(n)}+D_{crise(n+1)}) \\ &-\frac{\Delta D_{bs(n+1)}\Delta D'_{bf(n+1)}}{\mu_{bf}^2}D_{crise(n+1)}(D_{in(n)}-D_{crise(n)}) \\ &-\frac{\Delta D_{bs(n+1)}\Delta D'_{bf(n+1)}}{\mu_{bf}^2}D_{crise(n+1)}^2 \\ &-\frac{l}{\Delta l}\Delta D_{bs(n)}(D_{in(n)}-D_{crise(n)}+D_{crise(n+1)}) \\ &+\frac{l}{\Delta l}\Delta D_{bs(n+1)}(D_{in(n)}-D_{crise(n)}+D_{crise(n+1)}). \end{aligned} \quad (\mathrm{A}\cdot 16)$$

Here, because $(D_{in(n)} - D_{crise(n)})$ and $D_{crise(n+1)}$ are normally distributed, they can be changed into standard normal distribution by standardization. Also, the square of the standard normal random variable has chi-squared distribution whose degrees of freedom is one. Therefore, the means of $(D_{in(n)} - D_{crise(n)})^2$ and $D_{crise(n+1)}^2$ can be obtained from the means of chi-squared distribution. Because of $(D_{in(n)} - D_{crise(n)}) \sim N(\mu_{in} - \mu_{crise}, \sigma_{in}^2 + \sigma_{crise}^2)$ and $D_{crise(n+1)} \sim N(\mu_{crise}, \sigma_{crise}^2)$, the means of their squares are;

$$\mathrm{E}[(D_{in(n)}-D_{crise(n)})^2] = \sigma_{in}^2+\sigma_{crise}^2+(\mu_{in}-\mu_{crise})^2, \quad (\mathrm{A}\cdot 17)$$

$$\mathrm{E}[D_{crise(n+1)}^2] = \sigma_{crise}^2 + \mu_{crise}^2. \quad (\mathrm{A}\cdot 18)$$

Then, $\mathrm{Cov}[A, B+C]$ is calculated;

$$\mathrm{Cov}[A, B+C] = -\frac{\mu_{bs}}{\mu_{bf}}(\sigma_{in}^2 + 2\sigma_{crise}^2). \quad (\mathrm{A}\cdot 19)$$

From Eqs. (A· 7), (A· 14), (A· 19), $\mathrm{Var}[A+B+C]$, is calculated as;

$$\begin{aligned}\mathrm{Var}[A+B+C] =& \left(1-2\frac{\mu_{bs}}{\mu_{bf}}\right)(\sigma_{in}^2+2\sigma_{crise}^2) \\ &+\frac{2(\mu_{bs}^2+\sigma_{bs}^2)}{\mu_{bf}^4}(\sigma_{in}^2+\sigma_{crise}^2)\sigma_{bf}^2 \\ &+\frac{2(\mu_{bs}^2+\sigma_{bs}^2)}{\mu_{bf}^4}(\mu_{in}-\mu_{crise})^2\sigma_{bf}^2 \\ &+\frac{2(\mu_{bs}^2+\sigma_{bs}^2)}{\mu_{bf}^4}\mu_{bf}^2(\sigma_{in}^2+\sigma_{crise}^2) \\ &+2\sigma_{bs}^2\left(\frac{\mu_{in}-\mu_{crise}}{\mu_{bf}}+\frac{l}{\Delta l}\right). \end{aligned} \quad (\mathrm{A}\cdot 20)$$

σ_{bf}^k/μ_{bf}^k and σ_{bs}^k/μ_{bf}^k $(k \geq 2)$ are approximated as zeros since σ_{bf} and σ_{bs} are much smaller than μ_{bf}. With $a = \sigma_{crise}^2/\sigma_{in}^2$ and $x = \mu_{bs}/\mu_{bf}$, the output jitter $\mathrm{Var}[A+B+C]$ is approximated;

$$\begin{aligned}\mathrm{Var}[A+B+C] \approx& \left\{2(a+1)x^2-2(2a+1)x+(2a+1)\right\}\sigma_{in}^2 \\ &+2\sigma_{bs}^2\left(\frac{l}{\Delta l}-\frac{\mu_{crise}-\mu_{in}}{\mu_{bf}}\right). \end{aligned} \quad (\mathrm{A}\cdot 21)$$

Appendix B: Derivation of Constraints

In Sect. 2.3, the constraints on LC buffer is discussed. Here, we will present the process when Eqs. (6), (7) are derived.

The sufficient condition for the jitter amplifier to work properly can be expressed as two conditions;

$$t_{in(n+1)} < t_{crise(n+1)} < t_{out(n+1)}, \quad (\mathrm{A}\cdot 22)$$

$$t_{out(n+1)} < t_{cfall(n+1)} < t_{in(n+2)}. \quad (\mathrm{A}\cdot 23)$$

The constraint on $t_{crise(n+1)}$, Eq. (A· 22), is rewritten as follows;

$$0 < t_{crise(n+1)} - t_{in(n+1)} < t_{out(n+1)} - t_{in(n+1)}. \quad (\mathrm{A}\cdot 24)$$

The right side of Eq. (A· 24) is the delay in the LC buffer, and then it is not less than the latency in the LC buffer in fast mode. Then, the sufficient condition of $t_{crise(n+1)}$ is;

$$\begin{aligned}&0 < t_{crise(n+1)} - t_{in(n+1)} < \frac{l}{\Delta l}\Delta D_{bf(n+1)}, \\ &0 < D_{rise(n)} - D_{in(n)} < \frac{l}{\Delta l}\Delta D_{bf(n+1)}. \end{aligned} \quad (\mathrm{A}\cdot 25)$$

With introducing the coefficient m, Eq. (A· 25) is expressed as the following two conditions;

$$0 < (\mu_{rise} - \mu_{in}) - m\sqrt{\sigma_{rise}^2 + \sigma_{in}^2}, \quad (\mathrm{A}\cdot 26)$$

$$(\mu_{rise}-\mu_{in})+m\sqrt{\sigma_{rise}^2+\sigma_{in}^2} < \frac{l}{\Delta l}\mu_{bf} - m\sigma_{bf}\sqrt{\frac{l}{\Delta l}}. \quad (\mathrm{A}\cdot 27)$$

They can be rewritten as conditions of μ_{rise}.

$$\mu_{in} + m\sqrt{\sigma_{rise}^2 + \sigma_{in}^2} < \mu_{rise}, \quad (\mathrm{A}\cdot 28)$$

$$\mu_{rise} < \mu_{in} - m\sqrt{\sigma_{rise}^2+\sigma_{in}^2} + \frac{l}{\Delta l}\mu_{bf} - m\sigma_{bf}\sqrt{\frac{l}{\Delta l}}. \quad (\mathrm{A}\cdot 29)$$

Here, μ_{rise} is expressed as $\mu_{rise} = \mu_{rise\_offset} + s\Delta\mu_{rise} (s \in \mathbb{Z}, s \geq 0)$. Also, μ_{rise_offset} assumed to be less than the right side of Eq. (A· 29). Then, the sufficient condition is;

$$\begin{aligned}\Delta\mu_{rise} <& \mu_{in} - m\sqrt{\sigma_{rise}^2+\sigma_{in}^2} + \frac{l}{\Delta l}\mu_{bf} - m\sigma_{bf}\sqrt{\frac{l}{\Delta l}} \\ &-\left(\mu_{in}+m\sqrt{\sigma_{rise}^2+\sigma_{in}^2}\right), \\ \Delta\mu_{rise} &+ 2m\sqrt{\sigma_{rise}^2+\sigma_{in}^2} < \frac{l}{\Delta l}\mu_{bf} - m\sigma_{bf}\sqrt{\frac{l}{\Delta l}}. \end{aligned} \quad (\mathrm{A}\cdot 30)$$

Next, the condition of $t_{cfall(n+1)}$, Eq. (A· 23), is;

$$t_{out(n+1)} - t_{in(n+1)} < t_{cfall(n+1)} - t_{in(n+1)} < t_{in(n+2)} - t_{in(n+1)}. \quad (A\cdot 31)$$

The left side of Eq. (A· 31) is the delay in the LC buffer, and then it is not more than the latency in the LC buffer in slow mode. Then, the sufficient condition of $t_{cfall(n+1)}$ is;

$$\frac{l}{\Delta l}\Delta D_{bs(n)} < t_{cfall(n+1)} - t_{in(n+1)} < t_{in(n+2)} - t_{in(n+1)},$$
$$\frac{l}{\Delta l}\Delta D_{bs(n)} < D_{fall(n)} - D_{in(n)} < D_{in(n+1)}. \quad (A\cdot 32)$$

Equation (A· 31) is expressed as the two equations with the coefficient m;

$$\frac{l}{\Delta l}\mu_{bs} + m\sigma_{bs}\sqrt{\frac{l}{\Delta l}} < (\mu_{fall} - \mu_{in}) - m\sqrt{\sigma_{fall}^2 + \sigma_{in}^2}, \quad (A\cdot 33)$$
$$(\mu_{fall} - \mu_{in}) + m\sqrt{\sigma_{fall}^2 + \sigma_{in}^2} < \mu_{in} - m\sigma_{in}. \quad (A\cdot 34)$$

They can be rewritten as conditions of μ_{fall}.

$$\frac{l}{\Delta l}\mu_{bs} + m\sigma_{bs}\sqrt{\frac{l}{\Delta l}} + \mu_{in} + m\sqrt{\sigma_{fall}^2 + \sigma_{in}^2} < \mu_{fall}, \quad (A\cdot 35)$$
$$\mu_{fall} < \mu_{in} - m\sqrt{\sigma_{fall}^2 + \sigma_{in}^2} + (\mu_{in} - m\sigma_{in}). \quad (A\cdot 36)$$

Here, μ_{fall} is expressed as $\mu_{fall} = \mu_{fall\_offset} + t\Delta\mu_{fall}$ ($t \in \mathbb{Z}, t \geq 0$). In addition, μ_{fall_offset} is assumed to be less than the right side of Eq. (A· 36). Then, the sufficient condition is;

$$\Delta\mu_{fall} < \mu_{in} - m\sqrt{\sigma_{fall}^2 + \sigma_{in}^2} + (\mu_{in} - m\sigma_{in}) - \left(\frac{l}{\Delta l}\mu_{bs} + m\sigma_{bs}\sqrt{\frac{l}{\Delta l}}\right) - \mu_{in} - m\sqrt{\sigma_{fall}^2 + \sigma_{in}^2},$$
$$\Delta\mu_{fall} + 2m\sqrt{\sigma_{fall}^2 + \sigma_{in}^2} < (\mu_{in} - m\sigma_{in}) - \left(\frac{l}{\Delta l}\mu_{bs} + m\sigma_{bs}\sqrt{\frac{l}{\Delta l}}\right). \quad (A\cdot 37)$$

Takehiko Amaki received the B.E. and M.E. degrees from Osaka University, Osaka, Japan, in 2008 and 2010, respectively, where he is currently working toward the Ph.D. degree in the Department of Information Systems Engineering. His research interest includes hardware random number generator.

Masanori Hashimoto 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. Since 2004, he has been an Associate Professor in Department of Information Systems Engineering, Graduate School of Information Science and Technology, Osaka University. His research interest includes computer-aided-design for digital integrated circuits, and high-speed circuit design. Dr. Hashimoto served on the technical program committees for international conferences including DAC, ICCAD, ASP-DAC, DATE, ISPD, ICCD and ISQED. He is a member of IEEE and IPSJ.

Takao Onoye received the B.E. and M.E. degrees in Electronic Engineering, and Dr.Eng. degree in Information Systems Engineering all from Osaka University, Japan, in 1991, 1993, and 1997, respectively. He is currently a professor in the Department of Information Systems Engineering, Osaka University. His research interests include media-centric low-power architecture and its SoC implementation. He is a member of IEEE, IPSJ, and ITE-J.