IACR Transactions on Cryptographic Hardware and Embedded Systems
ISSN 2569-2925, Vol. 2022, No. 4, pp. 88–109. DOI:10.46586/tches.v2022.i4.88-109

An energy and area efficient, all digital entropy source compatible with modern standards based on jitter pipelining

Adriaan Peetermans and Ingrid Verbauwhede

imec-COSIC, KU Leuven, Leuven, Belgium, firstname.lastname@esat.kuleuven.be

Abstract. This paper proposes an energy and area efficient entropy source, suitable for true random number generation, accompanied with a stochastic model in a 28 nm CMOS technology. The design uses a jitter pipelining architecture together with an increased timing resolution to achieve a maximal throughput of 298 Mbit/s and a best energy efficiency of 1.46 pJ/bit at a supply of 0.8 V. The generated random bits pass the NIST SP 800-90B IID tests with a min entropy rate of 0.933 bit/bit, which is more than required by the AIS-31 standard. The all digital design allows for effortless transfer to other technology nodes, taking advantage of all benefits related to further technology scaling.

Keywords: TRNG · Entropy · AIS-31 · NIST SP 800-90B

1 Introduction

Modern cryptographic systems require a substantial amount of true random data (e.g. key material, masks, initialisation, etc.). This demand for cryptographic grade randomness only tends to increase in the near future with the emergence of post quantum secure cryptographic algorithms. By providing high quality randomness, True Random Number Generators (TRNGs) form a solid foundation that allows for the implementation of higher level algorithms and protocols.

Validating the performance of an Entropy Source (ES) (entropy generating component of a TRNG) is often done by solely assessing the quality of the generated random data [TRA21, KLK17]. Following an iterative approach (Fig. 1, top), the verifier generates a certain amount (determined by the selected test) of random data and applies that data to a series of statistical tests. The tests determine if the data can be regarded as "random" with a predefined significance level. If the data fails the tests, certain design parameters of the ES circuit are fine-tuned and the tests are run again, until the ES output passes. Previous examples [Dic03] have shown that this approach can lead to TRNGs that are prone to prediction attacks. Statistical tests that only work with the output of the ES/TRNG cannot differentiate between sequences generated by deterministic algorithms, TRNGs, or a combination of both [BBF09].

To overcome this concern, a new approach (Fig. 1, bottom) was proposed by international standardisation bodies [TBK+18, KS11, ISO19]. The new workflow is centered on the existence of a stochastic model characterising the entropy extraction process taking place in the ES circuitry. Entropy requirements (by the standard and/or the application), model assumptions (e.g. the existence of a certain type of noise source), and platform parameters (e.g. the intrinsic gate delay) form the input to this model. From the model, an optimisation procedure can be determined to select the value of the ES design parameters

Received: 2022-04-15 Accepted: 2022-06-15 Published: 2022-08-31

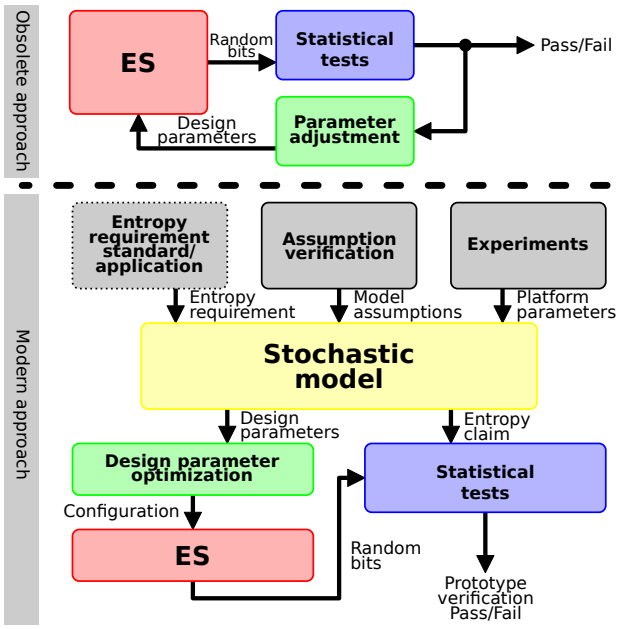

Figure 1: Obsolete versus modern ES verification approach.

(e.g. number of Ring Oscillator (RO) stages, jitter accumulation duration, etc.). Additionally, the model makes a prediction on the amount of entropy being generated. Statistical tests can then be used to verify this prediction and serve as a sanity check for individual prototypes.

ES designs that are compatible with this modern approach include the Transition Effect Ring Oscillator (TERO) ES, first proposed by [VD10] and implemented in [YBS16] (oscillator jitter based), a cross-coupled inverter pair ES, proposed and implemented by [MSA+12] (metastability based), or the Self-Timed Ring (STR) ES, first proposed by [CFAF13] and implemented in [CCF+18] (oscillator jitter based).

Following the modern approach, this work has the following contributions:

- A novel all digital ES architecture based on the unpredictable timing jitter in inverter Delay Chain (DC) ROs is proposed. The timing jitter is resolved by a Time to Digital Converter (TDC) based on two free running ROs, similar as in [KG04]. The benefits of having an all digital architecture are twofold: it allows for an easy integration into more complex digital systems, and it enjoys the benefits of further CMOS scaling, without having to redesign the entire circuit.

- Reducing the TDC resolution and accumulating independent timing jitter concurrently (creating a jitter pipeline), decreases the jitter accumulation time required and a throughput of several 100 Mbit/s could be achieved. This throughput is substantially larger than previously reported for oscillator jitter based ESs [YBS16, KLK17, CCF+18, YFH+14, KHL21].

- A detailed stochastic model is provided, capable of estimating the generated output entropy.

- Effort was done to measure out and estimate the magnitude of the jitter strength parameter (required by the stochastic model).

- A parameter optimisation strategy is provided to optimise the ES throughput.

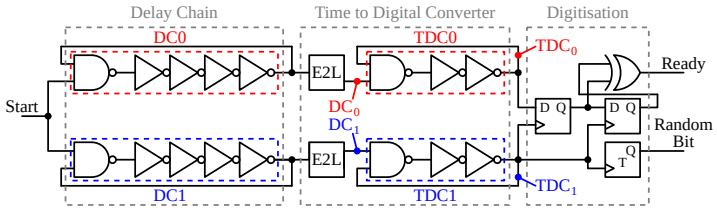

Figure 2: ES jitter pipelining architecture.

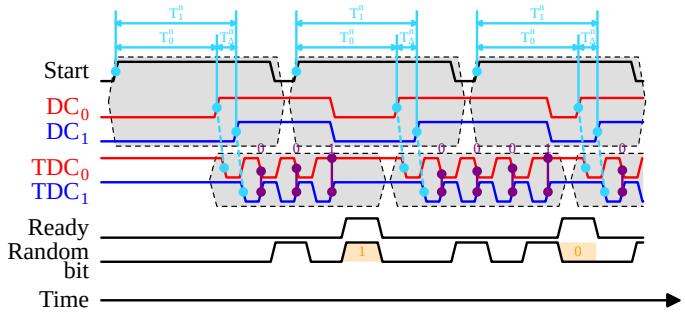

Figure 3: ES jitter pipelining timing diagram.

- The design is implemented in a 28 nm CMOS technology and measurement results are available.

2 ES architecture

This section will provide a high-level description of the ES architecture and jitter pipeline principle, before heading to a more detailed mathematical analysis of the design in Sect. 3.

2.1 Jitter pipeline

The proposed ES architecture is depicted in Fig. 2. Three components can be differentiated: a DC, a TDC and a digitisation block. Both the DC and the TDC consist of two ROs: DC0, DC1 and TDC0, TDC1 respectively. Timing jitter will naturally accumulate in all four ROs, when left running for a specified accumulation time interval. The TDC ROs are used to resolve the timing jitter generated by the DC ROs with a resolution related to the period difference of the two TDC ROs. The resolvement action leads to a digital representation of the timing difference created by the DC. This representation is then used to construct a random output bit.

To minimise idle time, the DC can already be restarted to accumulate jitter for the following output bit, during the resolvement phase of the current output bit. Timing jitter further accumulates during the resolvement phase, as the TDC contains free running ROs as well. Both phases therefore provide independent contributions to the output bit entropy, effectively creating a jitter pipeline, where jitter is being generated in a first (DC) stage, before being handed over to a second (TDC) stage, where it accumulates further. The pipelining principle is indicated by the shaded boxes in Fig. 3.

The concept of jitter pipelining is not limited to the jitter accumulation and resolvement using a DC and a TDC structure as showcased in this work, but should be regarded as a broader concept, that might be exploited in other ES architectures as well.

2.2 Architecture timing description

A start edge is applied to both DC ROs. Each DC RO consists of a chain of four delay-configurable inverters. The Edge To Level (E2L) blocks in Fig. 2 react to the n-th positive edge generated by the DC ROs by disabling the DC ROs and outputting a positive edge (DC_0 and DC_1). The time it takes for the start edge to propagate through both DC ROs for n cycles, to the output of the E2L block is indicated as T_0^n and T_1^n, for DC0 and DC1 respectively at the timing diagram in Fig. 3. Random timing jitter variations make the timing difference: $T_\Delta^n = T_0^n - T_1^n$, a random variable over multiple evaluations. The E2L outputs (DC_0 and DC_1 in Fig. 2) enable the TDC ROs to start oscillating (TDC_0 and TDC_1).

Both TDC ROs are configured to have a slightly different oscillation period, which defines the TDC resolution: $res = |P_{TDC0} - P_{TDC1}|$. The ROs start with an initial phase difference determined by T_Δ^n and keep oscillating until the phase difference is either 0° or 180° (π radians). The digitisation circuitry detects this phase synchronisation as the bottom RO (TDC1) will start to sample a different logic value from the top RO (TDC0) by means of an XOR gate. A T flip-flop will determine if during the phase synchronisation, TDC1 experienced an odd or an even amount of cycles. The output of this T flip-flop is used as the random output bit.

3 Stochastic model

This section elaborates a mathematical characterisation of the proposed circuit in Sect. 2 and quantifies the amount of entropy being extracted from the available timing jitter. The entropy estimation presented in this work, will be solely based on the existence of unmanipulatable thermal noise. Other noise sources will inevitably also be present. As we assume thermal noise is independent from all other sources of noise, the coexistence of these other noise sources will not lead to an entropy reduction and the estimation provided here is certainly a lower bound.

3.1 Model assumptions

To start off, four main assumptions made in the model are listed below:

- Thermal noise is unmanipulatable and independent from other noise sources.
- DC and TDC ROs are all mutually independent oscillators, affected by thermal noise.
- RO phase affected by thermal noise behaves as a Wiener process with drift.
- Jitter strength (defined in Sect. 3.2.2) is small: $F_{noise} \ll 1\,\mathrm{s}$.

3.2 Prerequisites

3.2.1 Notation

In this text, random variables and their realisations are denoted as uppercase and lowercase characters respectively. A stochastic process through time $t \geq 0$ is represented as an uppercase function (e.g. $\{X(t)\}_{t\geq 0}$) and a realisation as a lowercase function (e.g. $\{x(t)\}_{t\geq 0}$). The Probability Mass Function (PMF) or Probability Density Function (PDF) for a discrete or continuous random variable respectively Y is denoted as $f_Y(\cdot)$. The Cumulative Distribution Function (CDF) of a random variable Y is represented as $F_Y(\cdot)$. The expected value and variance of a random variable Y are:

$$\begin{aligned}\mathbf{E}[Y] &= \sum_i y_i f_Y(y_i), \quad \text{for } Y \text{ discrete} \\ &= \int_{-\infty}^{\infty} y f_Y(y) dy, \quad \text{for } Y \text{ continuous,}\end{aligned} \tag{1}$$

$$\mathbf{Var}[Y] = \mathbf{E}[(Y - \mathbf{E}[Y])^2]. \tag{2}$$

The probability of an event E is noted as $\mathcal{P}[E]$. A conditioned random variable X, given the knowledge of another random variable Y is denoted as: $X|Y$.

A random variable Y following a distribution $\mathcal{D}$, with parameters p_i is represented as: $Y \sim \mathcal{D}(p_1, p_2, \cdots)$. The distributions being used in this text are:

- Gaussian distribution: $\mathcal{N}(a, b^2)$, with mean a and variance b^2.
- Inverse Gaussian distribution: $IG(a, b^2)$, with mean a and variance $\frac{a^3}{b^2}$.

The PDF and CDF of a standard normal distributed variable (Gaussian distributed with mean 0 and variance 1) are denoted as $\varphi_{norm}(\cdot)$ and $\Phi_{norm}(\cdot)$ respectively.

3.2.2 Description of a noisy oscillating signal

Some prerequisite knowledge on timing jitter in free running ROs is given.

Noiseless The phase of an oscillating noiseless signal is a continuous linear function through time t:

$$\varphi(t) = \mu t + \varphi_0, \tag{3}$$

with μ defining the oscillation speed or angular frequency and φ_0 determining the phase at time zero. The phase of an oscillator cannot be explicitly observed. The observable waveform $e(\varphi)$ (current flow or node voltage) is defined as a function of the implicit phase, some examples are:

$$\begin{aligned}e(\varphi) &= A\sin(\varphi), \\ e(\varphi) &= \begin{cases} A, & \text{if } \varphi \bmod 2\pi < \pi \\ 0, & \text{if } \varphi \bmod 2\pi \geq \pi, \end{cases} \\ e(\varphi) &= \frac{A}{2\pi}\varphi \bmod 2\pi,\end{aligned} \tag{4}$$

representing sinusoidal, square and sawtooth waveforms respectively, with amplitude A. The operator $\cdot \bmod a$, is shorthand notation for $\cdot - \left\lfloor \frac{\cdot}{a} \right\rfloor a$, or the positive remainder after division by a. The waveform can then be described as a composite function of time: $w(t) = (e \circ \varphi)(t)$ Each of these waveforms has a period: $P_w = \frac{2\pi}{\mu}$, meaning that $w(t + P_w) = w(t)$ for any t. The waveform frequency is the inverse of the period: $F_w = \frac{\mu}{2\pi}$.

Noisy In this work, we assume the phase of an oscillator affected by thermal noise, $\{\Phi(t)\}_{t\geq 0}$, to behave as a Wiener process with drift:

$$\Phi(t) = \mu t + \varphi_0 + \sigma W(t), \quad \text{for } t \geq 0, \tag{5}$$

with φ_0 again the phase at time zero, μ the drift and σ^2 the infinitesimal variance. $\{W(t)\}_{t\geq 0}$ represents a Wiener process without drift. The oscillator is assumed to start at

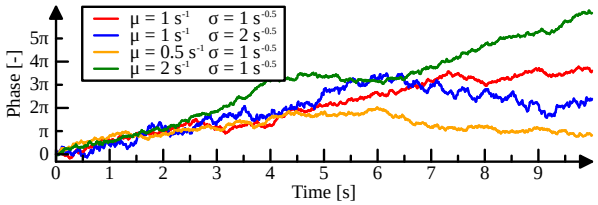

Figure 4: Example instances of a random phase process.

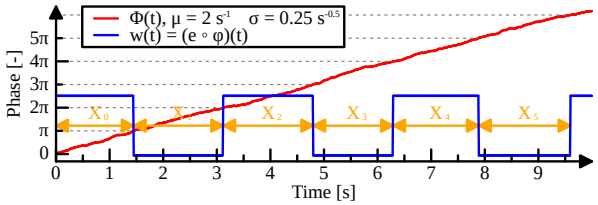

Figure 5: RO waveform and corresponding phase versus time.

time zero, as the Wiener process is undefined for negative time. The assumption is related to the fact that a Wiener process with drift describes the integration of currents with a white (thermal) noise component onto a load capacitor, as was explained by [Abi06]. Some example instances of this phase process are given in Fig. 4. At any moment in time t_a, the value of the phase is Gaussian distributed:

$$\Phi(t_a) \sim \mathcal{N}(\mu t_a + \varphi_0, \sigma^2 t_a). \tag{6}$$

Noisy Ring Oscillator An RO is modelled using the square waveform with amplitude equal to one as the observable waveform phase function ($e(\varphi)$), the phase is modelled by a Wiener process with drift and zero initial phase: $\Phi(t) = \mu t + \sigma W(t)$, as shown in Fig. 5.

The half-period duration of the k-th half period is represented by the random variable X_k. Due to the independent increment property of the Wiener process, each half period duration of the RO output is Independent and Identically Distributed (IID) compared to all other half periods and can be represented by a single random variable: X. This time duration is equal to the time it takes for the oscillator phase to reach a multiple of π. Again due to the independent increment property, we only look at the time required to reach a phase of π, starting from phase zero. All other half periods have identical distribution. The time required for a Wiener process with drift to hit a certain level, α, for the first time is inverse-Gaussian distributed: $IG(\frac{\alpha}{\mu}, (\frac{\alpha}{\sigma})^2)$. The half-period duration distribution is then given as:

$$X \sim IG(\frac{\pi}{\mu}, (\frac{\pi}{\sigma})^2). \tag{7}$$

From this, the expected value and variance for X can be calculated: $\mathbf{E}[X] = \frac{\pi}{\mu}$ and $\mathbf{Var}[X] = \frac{\pi\sigma^2}{\mu^3}$. The jitter strength, controlling the rate at which jitter accumulates in the RO is then equal to:

$$F_{noise} = \frac{\mathbf{Var}[X]}{\mathbf{E}[X]} = (\frac{\sigma}{\mu})^2, \tag{8}$$

with units of time. In practical applications, this quantity is in the order of femtoseconds [YRG+17].

Note that an assumption was made in case of positive drift μ and phase started at zero, the phase would not return back and cross zero into negative values. Zero however is also a multiple of π and will therefore produce an edge at the output when crossed. This is related to the fact that the inverse-Gaussian distribution only describes the first passage time. For small values of drift relative to the infinitesimal variance ($\frac{\mu}{\sigma} \ll 1$), the phase could pass a certain level multiple times, with each passage creating an edge at the output.

The assumptions made will therefore only hold when $\frac{\mu}{\sigma} \gg 1$, which is true in most applications ($F_{noise} \ll 1\,\mathrm{s}$). The probability of the phase returning back to its starting value and crossing it is equal to:

$$\mathcal{P}[\Phi(t) \leq 0] = \Phi_{norm}(-\frac{\mu}{\sigma}\sqrt{t}), \tag{9}$$

This probability diminishes rapidly in time when $F_{noise} \ll 1\,\mathrm{s}$. For very small time instances ($t \approx (\frac{\sigma}{\mu})^2$ or lower), the assumption will also not hold, as the RO output waveform cannot be seen as an ideal digital signal anymore.

3.3 Delay Chain time difference distribution

The DC consists of two noisy free running ROs. The RO phase is described as a stochastic process:

$$\begin{aligned} \Phi_{DC_0}(t) &= \mu_{DC_0} t + \sigma_{DC_0} W_{DC_0}(t) \quad \text{for } t \geq 0, \\ \Phi_{DC_1}(t) &= \mu_{DC_1} t + \sigma_{DC_1} W_{DC_1}(t) \quad \text{for } t \geq 0. \end{aligned} \tag{10}$$

The DCs start at time zero with a phase equal to zero. Both DCs run a prescribed number of periods n and cause the E2Ls to activate at times T_0^n and T_1^n respectively:

$$\begin{aligned} \Phi_{DC_0}(T_0^n) &= n2\pi, \\ \Phi_{DC_1}(T_1^n) &= n2\pi. \end{aligned} \tag{11}$$

The first passage time at a level $n2\pi$ of a Wiener process with drift is, as before, described by the inverse-Gaussian distribution:

$$\begin{aligned} T_0^n &\sim IG(\frac{n2\pi}{\mu_{DC_0}}, (\frac{n2\pi}{\sigma_{DC_0}})^2), \\ T_1^n &\sim IG(\frac{n2\pi}{\mu_{DC_1}}, (\frac{n2\pi}{\sigma_{DC_1}})^2). \end{aligned} \tag{12}$$

The DC time difference distribution T_Δ^n after n periods is equal to:

$$T_\Delta^n = T_0^n - T_1^n, \tag{13}$$

which is defined by a subtraction of two independent random variables. Its CDF can be calculated by integrating the PDFs of T_0^n and T_1^n.

$$\begin{aligned} F_{T_\Delta^n}(t) &= \mathcal{P}[T_\Delta^n \leq t] = \mathcal{P}[T_0^n \leq t + T_1^n] \\ &= \begin{cases} \int_0^\infty f_{T_1^n}(t_1) \int_0^{t+t_1} f_{T_0^n}(t_0)\, dt_0\, dt_1 & \text{if } t \geq 0 \\ \int_0^\infty f_{T_0^n}(t_0) \int_{t_0-t}^\infty f_{T_1^n}(t_1)\, dt_1\, dt_0 & \text{if } t < 0. \end{cases} \end{aligned} \tag{14}$$

The PDF for T_Δ^n is then equal to:

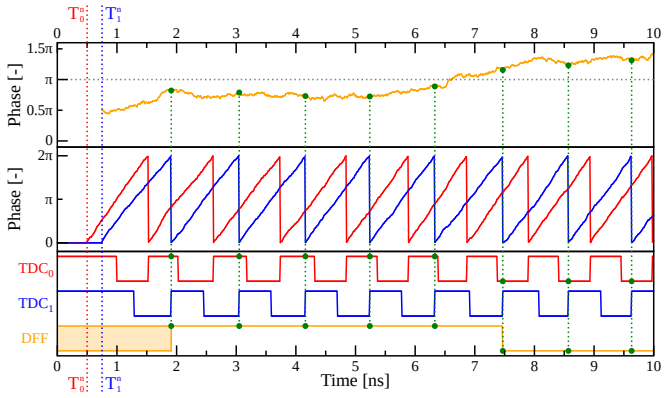

Figure 6: Relation between TDC phases and sampling time instances. The middle graph shows TDC phases modulo 2π ($\Phi_{TDC_0}(t) \bmod 2\pi$ and $\Phi_{TDC_1}(t) \bmod 2\pi$), the top graph shows the TDC phase difference ($\Phi_{TDC_0}(t) - \Phi_{TDC_1}(t)$), and the bottom graph shows the output waveform of TDC0 (top), TDC1 (middle), and sampling flip flop (bottom).

$$f_{T^n_\Delta}(t) = \frac{\partial F_{T^n_\Delta}(t)}{\partial t}. \tag{15}$$

Note that in this model, T_0^n and T_1^n are assumed to be independent. Effort was made in the design and layout of all four ROs to make sure coupling is minimised by introducing separate supply networks and placing each RO in its own N-well. If a dependency would still be present, this leads to a reduced jitter strength estimate in Sect. 4 and therefore reduces the entropy claim made by this model.

3.4 Time to Digital Converter run time distribution

The TDC oscillators start oscillating when the respective DC has finished running n cycles (times T_0^n and T_1^n respectively). Each TDC is a free running RO and the phases can be represented as a stochastic process:

$$\begin{aligned} \Phi_{TDC_0}(t) &= \mu_{TDC_0}(t - T_0^n) + \sigma_{TDC_0} W_{TDC_0}(t - T_0^n) \quad \text{for } t \geq T_0^n, \\ \Phi_{TDC_1}(t) &= \mu_{TDC_1}(t - T_1^n) + \sigma_{TDC_1} W_{TDC_1}(t - T_1^n) \quad \text{for } t \geq T_1^n. \end{aligned} \tag{16}$$

Note that both T_0^n and T_1^n are random variables, following the distributions from Eq. 12. $\Phi_{TDC_0}(t)$ and $\Phi_{TDC_1}(t)$ are therefore representing random Wiener processes with drift and a random starting time instance.

TDC1 samples TDC0 by using a Data Flip Flop (DFF). Figure 6 shows the relation between the TDC phases and the sampling time instances. Whenever the sampled value (DFF output) toggles, the TDCs stop oscillating and the number of TDC1 periods is outputted. From this figure, it can be seen that the toggling of the DFF output happens whenever the two TDC phases have diverged by a value of more than π. The TDCs stop at the next positive edge of TDC1. The TDC phase difference $\Phi_\Delta(t)$ is defined only for time instances after the second TDC has started:

$$\begin{aligned} \Phi_\Delta(t) &= \Phi_{TDC_0}(t) - \Phi_{TDC_1}(t) \\ &= (\mu_{TDC_0} - \mu_{TDC_1})t + \mu_{TDC_1} T_1^n - \mu_{TDC_0} T_0^n \\ &\quad + \mathcal{N}(0, \sigma^2_{TDC_0}(t - T_0^n) + \sigma^2_{TDC_1}(t - T_1^n)) \quad \text{for } t \geq \max(T_0^n, T_1^n). \end{aligned} \tag{17}$$

The notation: $\cdot + \mathcal{N}(a, b^2)$ in Eq. 17 indicates the addition of a normal distributed variable X, such that $X \sim \mathcal{N}(a, b^2)$. This normal distributed variable follows from the properties of Wiener processes and addition of normal variables: $aW(b) \sim \mathcal{N}(0, a^2 b)$, for any $b \geq 0$, and $\mathcal{N}(a, b^2) - \mathcal{N}(c, d^2) \sim \mathcal{N}(a - c, b^2 + d^2)$, for two independent normal distributed random variables. Here, three cases can be distinguished:

1. $\boldsymbol{T_\Delta^n > 0}$ or $\boldsymbol{T_0^n > T_1^n}$: the clock TDC (TDC1) starts running first. A time shift is performed to $\Phi_\Delta(t)$, such that: $t' = t - T_0^n$. Substituting this into Eq. 17:

$$\begin{aligned}\Phi_\Delta(t' + T_0^n) = (\mu_{TDC_0} - \mu_{TDC_1})(t' + T_0^n) + \mu_{TDC_1} T_1^n - \mu_{TDC_0} T_0^n \\ + \mathcal{N}(0, \sigma_{TDC_0}^2 t' + \sigma_{TDC_1}^2 (t' + T_0^n - T_1^n)) \quad \text{for } t' \geq 0.\end{aligned} \tag{18}$$

   This can be further simplified into:

$$\begin{aligned}\Phi_\Delta(t' + T_0^n) = (\mu_{TDC_0} - \mu_{TDC_1}) t' + \sqrt{\sigma_{TDC_0}^2 + \sigma_{TDC_1}^2} W'_{TDC}(t') \\ - \Phi_{TDC_1}(T_0^n) \quad \text{for } t' \geq 0.\end{aligned} \tag{19}$$

   Equation 19 shows that the TDC phase difference $\Phi_\Delta(t)$, for $t \geq T_0^n$, can be written as a new Wiener process with drift $\mu_\Delta = \mu_{TDC_0} - \mu_{TDC_1}$ and infinitesimal variance $\sigma_\Delta^2 = \sigma_{TDC_0}^2 + \sigma_{TDC_1}^2$, subtracted with the accumulated phase $\Phi_{TDC_1}(T_0^n)$ (from time T_1^n to T_0^n). This accumulated phase in TDC1 is independent of the Wiener process $W'_{TDC}(t')$, as this process only starts at time $t' = 0$ or $t = T_0^n$ and due to the nature of Wiener processes, phase accumulated over non-overlapping time intervals is independent.

2. $\boldsymbol{T_\Delta^n < 0}$ or $\boldsymbol{T_0^n < T_1^n}$: the clock TDC (TDC1) starts running last. The reasoning from the case $T_\Delta^n > 0$ can be repeated with a time shift: $t' = t - T_1^n$. This will produce:

$$\begin{aligned}\Phi_\Delta(t' + T_1^n) = (\mu_{TDC_0} - \mu_{TDC_1}) t' + \sqrt{\sigma_{TDC_0}^2 + \sigma_{TDC_1}^2} W'_{TDC}(t') \\ + \Phi_{TDC_0}(T_1^n) \quad \text{for } t' \geq 0.\end{aligned} \tag{20}$$

   Again, the phase accumulated in TDC0 ($\Phi_{TDC_0}(T_1^n)$) from T_0^n to T_1^n is independent of the Wiener process $W'_{TDC}(t')$, starting at time $t' = 0$ or $t = T_1^n$.

3. $\boldsymbol{T_\Delta^n = 0}$ or $\boldsymbol{T_0^n = T_1^n}$: both TDCs start at exactly the same time. Note that this is only a theoretical case, as T_Δ^n is described by a continuous probability density function and the probability of this case happening is effectively equal to zero.

   To be complete however, the TDC phase difference is now equal to (with a time shift: $t' = t - T_0^n = t - T_1^n$):

$$\begin{aligned}\Phi_\Delta(t' + T_0^n) = (\mu_{TDC_0} - \mu_{TDC_1}) t' + \sqrt{\sigma_{TDC_0}^2 + \sigma_{TDC_1}^2} W'_{TDC}(t') \\ = \Phi_\Delta(t' + T_1^n) \quad \text{for } t' \geq 0.\end{aligned} \tag{21}$$

   In this case, the subtraction with the phase accumulated in one of the TDCs disappears, as no TDC had been running before the second one starts.

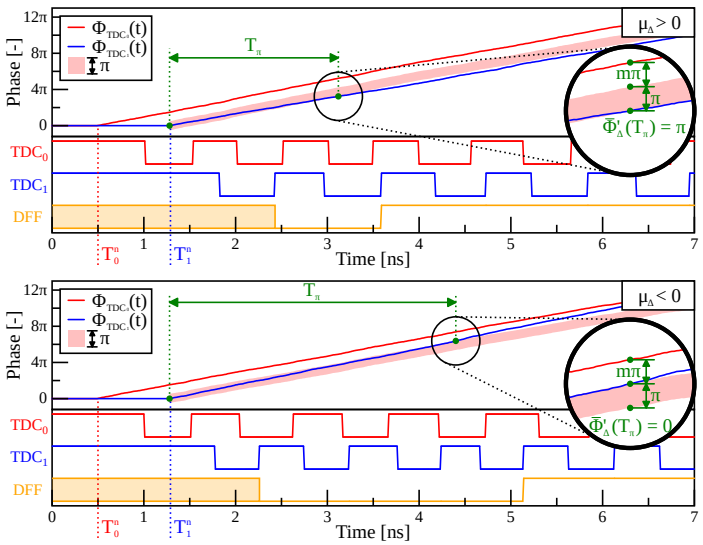

Figure 7: Relation between TDC phases and sampling time instances for $\mu_\Delta > 0$ (top) and $\mu_\Delta < 0$ (bottom).

The shifted TDC phase difference $\Phi'_\Delta(t)$ is now introduced:

$$\Phi'_\Delta(t) = \Phi_\Delta(t + \max(T_0^n, T_1^n)) \quad \text{for } t \geq 0. \tag{22}$$

The TDC phase difference starts off at a random variable, determined by the DC time difference T_Δ^n:

$$\Phi_\Delta^0 = \Phi'_\Delta(0) = \begin{cases} -\Phi_{TDC_1}(T_0^n) & \text{if } T_\Delta^n > 0 \\ \Phi_{TDC_0}(T_1^n) & \text{if } T_\Delta^n < 0 \\ 0 & \text{if } T_\Delta^n = 0. \end{cases} \tag{23}$$

From the description of the random processes in Eq. 16, Φ_Δ^0, conditioned to T_Δ^n, is distributed as follows:

$$\Phi_\Delta^0 | T_\Delta^n \sim \begin{cases} \mathcal{N}(-\mu_{TDC_1} T_\Delta^n, \sigma^2_{TDC_1} T_\Delta^n) & \text{if } T_\Delta^n > 0 \\ \mathcal{N}(-\mu_{TDC_0} T_\Delta^n, -\sigma^2_{TDC_0} T_\Delta^n) & \text{if } T_\Delta^n < 0 \\ 0 & \text{if } T_\Delta^n = 0, \end{cases} \tag{24}$$

with $X \sim 0$ indicating that the variable X follows a degenerate distribution centred at zero, with PDF equal to the Dirac delta function $\delta(\cdot)$. From the start on, the TDC phase difference $\Phi'_\Delta(t)$ will behave as a Wiener process with drift, added to this initial phase difference. The TDCs will stop oscillating, whenever the value of $\Phi'_\Delta(t)$ crosses for the first time a multiple of π at time T_π:

$$T_\pi = \min_{t \geq 0}(t | \Phi'_\Delta(t) = m\pi \text{ and } m \in \mathbb{Z}). \tag{25}$$

An example phase instance of the two TDCs is shown in Fig. 7. Because we are only interested in the first passage time of $\Phi'_\Delta(t)$ at a multiple of π, the initial phase difference can be reduced modulo π:

$$\bar{\Phi}'_\Delta(t) = (\Phi'_\Delta(t) - \Phi_\Delta^0) + \Phi_\Delta^0 \bmod \pi \quad \text{for } t \geq 0. \tag{26}$$

Both $\bar{\Phi}'_\Delta(t)$ and $\Phi'_\Delta(t)$ will have equal first passage times at a multiple of π: T_π. Note that for the reduced phase difference $\bar{\Phi}'_\Delta(t)$ the first passage level at a multiple of π will be either zero or π:

$$\bar{\Phi}'_\Delta(T_\pi) = 0 \text{ or } \pi. \tag{27}$$

Depending on the sign of the phase drift difference (μ_Δ), the reduced phase difference $\bar{\Phi}'_\Delta(t)$ will drift towards one of the two boundaries for $\mu_\Delta > 0$ or $\mu_\Delta < 0$, as shown in Fig. 7.

When started too close to an opposite boundary 0 in case $\mu_\Delta > 0$ and π in case $\mu_\Delta < 0$, the phase difference could hit this boundary, prematurely ending the oscillations. We can now determine the CDF for T_π, conditioned to Φ^0_Δ:

$$\begin{aligned} F_{T_\pi|\Phi^0_\Delta}(t|\varphi) &= \mathcal{P}[T_\pi \leq t|\Phi^0_\Delta = \varphi] = 1 - \mathcal{P}[T_\pi > t|\Phi^0_\Delta = \varphi] \\ &= \begin{cases} 1 - \mathcal{P}[0 < \bar{\Phi}'_\Delta(t) \leq \pi|\Phi^0_\Delta = \varphi] & \text{if } t \geq 0 \\ 0 & \text{if } t < 0. \end{cases} \end{aligned} \tag{28}$$

The condition for the oscillations to continue can be written more explicitly:

$$0 < \mu_\Delta t + \sigma_\Delta W'_{TDC}(t) + \varphi \bmod \pi \leq \pi. \tag{29}$$

Moving all deterministic parts to the outside:

$$-\frac{\mu_\Delta t + \varphi \bmod \pi}{\sigma_\Delta} < W'_{TDC}(t) \leq \frac{\pi - \mu_\Delta t - \varphi \bmod \pi}{\sigma_\Delta}. \tag{30}$$

From the property of Wiener processes: $W(a) \sim \sqrt{a}\mathcal{N}(0,1)$, we can rewrite the boundaries from Eq. 30:

$$-\frac{\mu_\Delta t + \varphi \bmod \pi}{\sigma_\Delta\sqrt{t}} < X \leq \frac{\pi - \mu_\Delta t - \varphi \bmod \pi}{\sigma_\Delta\sqrt{t}}, \tag{31}$$

with $X \sim \mathcal{N}(0,1)$ (standard normal distributed). Substituting this result into Eq. 28 gives:

$$F_{T_\pi|\Phi^0_\Delta}(t|\varphi) = \begin{cases} 1 - \Phi_{norm}(\frac{\pi - \mu_\Delta t - \varphi \bmod \pi}{\sigma_\Delta\sqrt{t}}) & \\ \quad + \Phi_{norm}(-\frac{\mu_\Delta t + \varphi \bmod \pi}{\sigma_\Delta\sqrt{t}}) & \text{if } t \geq 0 \\ 0 & \text{if } t < 0. \end{cases} \tag{32}$$

Figure 8 depicts how these boundaries will evolve through time for different drift difference (μ_Δ), infinitesimal difference variance (σ_Δ), and initial phase difference condition ($\Phi^0_\Delta = \varphi$). The conditional PDF for T_π can then be obtained by differentiating the CDF.

3.5 Output bit probability distribution

The TDCs will stop oscillating at the first positive edge of TDC1 after time T_π. The number of cycles of TDC1 will then be used to construct the output random bit. The number of completed cycles, R, is equal to:

$$R = \lceil \frac{\Phi_{TDC_1}(T_\pi + \max(T^n_0, T^n_1))}{2\pi} \rceil. \tag{33}$$

The term $\max(T^n_0, T^n_1)$ is added to T_π, as T_π was defined for the shifted phase difference. This term represents the accumulated phase in case TDC1 was started first.

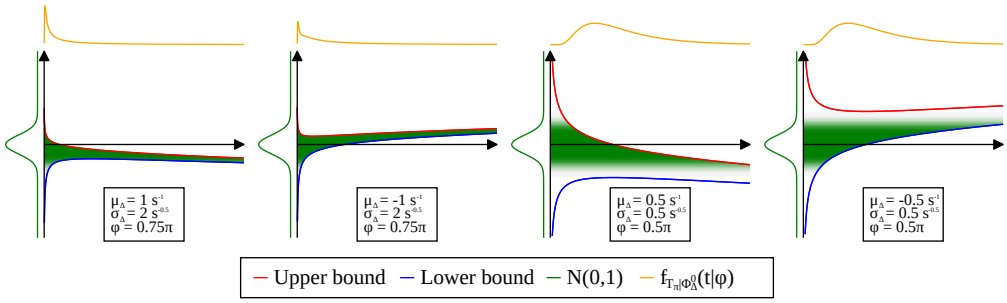

Figure 8: Lower and upper boundaries from Eq. 31, for different μ_Δ, σ_Δ, and $\Phi_\Delta^0 = \varphi$.

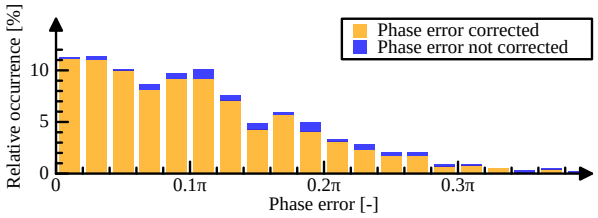

Figure 9: Histogram of 1000 repeated simulations, showing the absolute Φ_{TDC_1} phase error introduced by using the simplified relation in Eq. 34. In most cases, the phase error will be corrected by the ceil operation. The simulations were performed with jitter strength equal to 30 fs.

There exists a dependency between T_π and the Wiener process determining $\Phi_{TDC_1}(\cdot)$. This makes deriving an analytical expression for the distribution of R not straightforward. To circumvent this issue, the noise contributing to $\Phi_{TDC_1}(\cdot)$ is neglected: $\Phi_{TDC_1}(t) \approx \mu_{TDC_1}(t - T_1^n)$. This simplification is justified by the fact that the jitter strength in the phase difference signal is much larger than in a single TDC: $\frac{\sigma_\Delta}{\mu_\Delta} \gg \frac{\sigma_{TDC_1}}{\mu_{TDC_1}}$. This is true for decently matched TDCs ($\mu_{TDC_0} \approx \mu_{TDC_1}$ and $|\mu_\Delta| \ll \mu_{TDC_1}$). Simulation results shown in Fig. 9 further justify the simplification as the introduced error in the phase Φ_{TDC_1} is low. The relative error (average deviation for R) is 0.2 %. Using this simplification, Eq. 33 is transformed to:

$$R = \left\lceil \frac{\mu_{TDC_1}(T_\pi + \max(T_0^n, T_1^n) - T_1^n)}{2\pi} \right\rceil. \tag{34}$$

Depending on the sign of T_Δ^n, we have:

$$R = \begin{cases} \left\lceil \frac{\mu_{TDC_1}(T_\pi + T_\Delta^n)}{2\pi} \right\rceil & \text{if } T_\Delta^n > 0 \\ \left\lceil \frac{\mu_{TDC_1} T_\pi}{2\pi} \right\rceil & \text{if } T_\Delta^n \leq 0. \end{cases} \tag{35}$$

The term $\mu_{TDC_1} T_\Delta^n$ can be more accurately replaced by $-\Phi_\Delta^0$ for $T_\Delta^n > 0$ from Eq. 23, as this term represents the accumulated phase in TDC1 before TDC0 was started. The conditional CDF for R can now be calculated:

$$\begin{aligned} F_{R|\Phi_\Delta^0, T_\Delta^n}(r|\varphi, t) &= \mathcal{P}[R \leq r | \Phi_\Delta^0 = \varphi, T_\Delta^n = t] \\ &= \begin{cases} \mathcal{P}[T_\pi \leq \frac{2\pi r + \varphi}{\mu_{TDC_1}} | \Phi_\Delta^0 = \varphi] & \text{if } t > 0 \\ \mathcal{P}[T_\pi \leq \frac{2\pi r}{\mu_{TDC_1}} | \Phi_\Delta^0 = \varphi] & \text{if } t \leq 0 \end{cases} \quad \text{for } r \in \mathbb{N}. \end{aligned} \tag{36}$$

R is a discrete random variable:

$$f_{R|\Phi^0_\Delta,T^n_\Delta}(r|\varphi,t) = \mathcal{P}[R = r|\Phi^0_\Delta = \varphi, T^n_\Delta = t] \\ = \begin{cases} F_{R|\Phi^0_\Delta,T^n_\Delta}(r|\varphi,t) - F_{R|\Phi^0_\Delta,T^n_\Delta}(r-1|\varphi,t) & \text{if } r \in \mathbb{N}_1 \\ F_{R|\Phi^0_\Delta,T^n_\Delta}(0|\varphi,t) & \text{if } r = 0. \end{cases} \tag{37}$$

Removing the conditionals, to obtain the joint distribution:

$$f_{R,\Phi^0_\Delta,T^n_\Delta}(r,\varphi,t) = f_{R|\Phi^0_\Delta,T^n_\Delta}(r|\varphi,t) f_{\Phi^0_\Delta|T^n_\Delta}(\varphi|t) f_{T^n_\Delta}(t), \tag{38}$$

with $f_{\Phi^0_\Delta|T^n_\Delta}(\varphi|t)$ and $f_{T^n_\Delta}(t)$ obtained from Eqs. 24 and 15 respectively. The random variables Φ^0_Δ and T^n_Δ are integrated out, to obtain the distribution for R:

$$f_R(r) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f_{R,\Phi^0_\Delta,T^n_\Delta}(r,\varphi,t)dtd\varphi. \tag{39}$$

The produced random bit B is now equal to the least significant bit of R. From this, the bit probability can be calculated:

$$\mathcal{P}[B = b] = \sum_{i=0}^{\infty} f_R(2i+b) \quad \text{for } b \in \{0,1\}. \tag{40}$$

As the entire system does not contain a state that is transferred from one bit generation to another, individual bits are IID by design.

4 Jitter strength measurement

The entropy estimate provided by the model in Sect. 3 is highly influenced by the platform dependent parameter: jitter strength (F_{noise}). This parameter determines the rate at which timing jitter will naturally accumulate in a free running RO. In contrast to model parameters (e.g. RO frequency), the value for the jitter strength cannot be measured out directly. As was proposed by [YRG+17], jitter measurement should happen on-chip and preferably with a differential measurement setup, to minimise external (non-thermal noise) influences that might lead to an overestimation of the available timing jitter.

It is important not to overestimate the jitter strength parameter, and use a conservative method here for the following two reasons: firstly, due to the nature of the measurement, measurement errors (e.g. external noise sources other than thermal noise that might be manipulable) will always manifest themselves as a positive bias. Intuitively, this can be explained by the fact that the jitter strength parameter will be determined based on observed measurement variance. If external, independent sources of error exist, they will always lead to an increase of observed measurement variance (adding two independent random variables will increase variance) and lead to an overestimation. Secondly, based on the estimated output entropy, a security claim will be made. If the available jitter strength was overestimated, this will lead to an overestimation of the produced output entropy, and therefore to an invalid security claim.

To get accurate results, the jitter measurement experiment was repeated on five separate devices (chips). The most conservative estimate will further be used to estimate entropy for all devices tested.

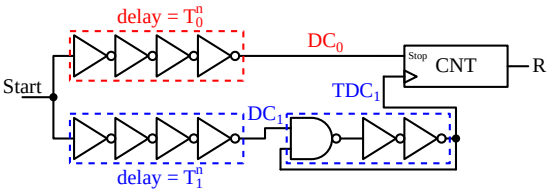

Figure 10: Jitter measurement circuit architecture.

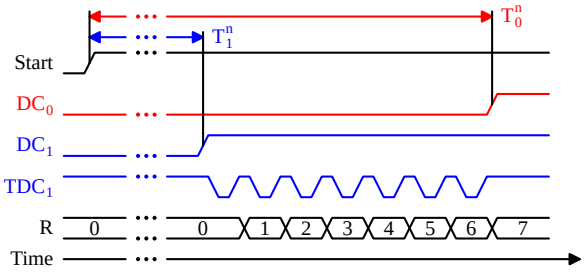

Figure 11: Jitter measurement time diagram.

4.1 On-chip measurement setup

The proposed ES architecture in Sect. 2 allows for on-chip differential jitter strength measurement as well. A circuit diagram, showing only the relevant parts for the jitter measurement, together with a timing diagram are shown in Figs. 10 and 11. By configuring TDC0 and TDC1 to have a long and short oscillation period respectively ($P_{TDC0} > 2P_{TDC1}$), it can be ensured that a positive edge of TDC1 will occur each half-period of TDC0. When each half period of TDC0 is sampled, the TDCs will stop oscillating as soon as both DCs have finished propagating. DC0 and DC1 are configured such that DC1 has a shorter propagation delay than DC0 ($T_0^n > T_1^n$). Therefore, TDC1 will oscillate during the time interval when DC1 finished propagating, but DC0 did not. A counter, counting the number of oscillations of TDC1 during this time interval, therefore produces an output proportional to the propagation delay difference between DC0 and DC1. By observing the counter output variance over multiple evaluations, an estimation for the differential DC propagation variance and, therefore, also for the available jitter strength in DC0 and DC1 is obtained.

4.2 Theoretical jitter analysis

Based on the stochastic model from Sect. 3, an estimate for the observed counter output variance can be made dependent on the value of F_{noise}. In this work, we choose the highest value for F_{noise}, that will still lead to an underestimation of the observed variance, as the final jitter strength estimate. The timing jitter accumulated by TDC1, will also influence the counter output variance. The model from Sect. 3 is extended here, to get an estimate for the counter output variance. The delay chain timing difference distribution (T_Δ^n) is given by Eq. 14. Due to a hardware constraint, the TDCs are only allowed to stop oscillating after both have gone through two full periods. The jitter accumulation time interval is therefore given as:

$$T_A^n = T_\Delta^n + T_{2TDC0}, \tag{41}$$

with T_{2TDC0} a random variable describing the time required for TDC0 to oscillate for two full periods. From Sect. 3, T_{2TDC0} is IG distributed:

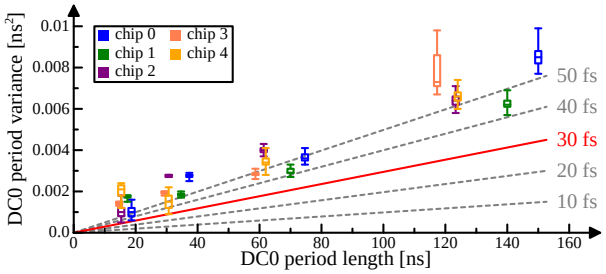

Figure 12: Jitter measurement results.

$$T_{2TDC0} \sim IG(\frac{4\pi}{\mu_{TDC_0}}, (\frac{4\pi}{\sigma_{TDC_0}})^2). \tag{42}$$

TDC1 will oscillate during this accumulation time interval T_A^n. The phase of TDC1 at the end of this interval (assuming it started with zero phase), conditioned on the accumulation time interval length, is Gaussian distributed:

$$[\Phi_J | T_A^n = t_a] = \Phi_{TDC_1}(t_a) \sim \mathcal{N}(\mu_{TDC_1} t_a, \sigma^2_{TDC_1} t_a) \quad \text{for } t_a \geq 0. \tag{43}$$

The condition to the accumulation time interval length can be removed similarly as was done in Sect. 3:

$$f_{\Phi_J}(\varphi) = \int_0^\infty \frac{1}{\sigma_{TDC_1}\sqrt{t_a}} \varphi_{norm}(\frac{\varphi - \mu_{TDC_1} t_a}{\sigma_{TDC_1}\sqrt{t_a}}) f_{T_A^n}(t_a) dt_a. \tag{44}$$

The TDC oscillations will only stop after a positive edge has occurred in TDC1. All phases of TDC1 in the interval $(2\pi(r-1), 2\pi r]$ will therfore produce the same counter output: r. The probability of the counter output R being equal to a value r (PMF) is then given by:

$$f_R(r) = \mathcal{P}[R = r] = \int_{2\pi(r-1)}^{2\pi r} f_{\Phi_J}(\varphi) d\varphi \quad \text{for } r \in \mathbb{N}_1. \tag{45}$$

From this result, the variance $\mathbf{Var}[R]$ can be calculated and compared with the measurements.

4.3 Measurement results

Figure 12 shows the jitter measurement results for five different devices. The experiment was repeated for four different DC accumulation time differences (T_Δ^n), determined by the parameter: $n \in \{1, 2, 4, 8\}$. For each device, 100 times 2^{16} counter outputs are collected. The counter output variance ($\mathbf{Var}[R]$) was calculated over the 2^{16} collected samples. Each device at each value for n is represented as a box plot, showing the experimental distribution for $\mathbf{Var}[T_0^n]$ derived from measuring $\mathbf{Var}[R]$, and using the model from Sect. 4.2, over the 100 repeated measurements. The straight lines in Fig. 12 represent the theoretical DC0 period length variance ($\mathbf{Var}[T_0^n]$) for different jitter strength magnitudes (F_{noise}), calculated as: $\mathbf{Var}[T_0^n] = F_{noise}\mathbf{E}[T_0^n]$. A conservative estimate equal to 30 fs is obtained.

5 Design parameter selection criteria

The proposed ES design has four design parameters that can be freely chosen by the designer: μ_{DC_0}, μ_{DC_1}, μ_{TDC_0} and μ_{TDC_1}. The infinitesimal variances (σ_X) are related to the phase drifts by the obtained jitter strength and Eq. 8. This section provides a selection strategy for these four parameters.

5.1 Pipeline balance

As for all pipelined architectures, balancing the propagation times for both stages is necessary. The propagation delay of the DC stage is given as:

$$d_{DC} = \max(T_0^n, T_1^n), \tag{46}$$

the slowest DC will determine when the TDCs can start resolving the timing difference. The maximal TDC resolving time is determined by the TDC resolution (res), defined as:

$$res = |P_{TDC_0} - P_{TDC_1}| = |\frac{2\pi}{\mu_{TDC_0}} - \frac{2\pi}{\mu_{TDC_1}}|. \tag{47}$$

Each period of TDC0, the TDC1 positive edge will have shifted with an amount of res compared to the positive edge of TDC0. The TDCs will stop oscillating as soon as TDC1 samples a different value from TDC0. This means at most $\frac{P_{TDC_0}}{2res}$ cycles of TDC1 are required. The maximal TDC resolving time is then given as:

$$d_{TDC} = \frac{P_{TDC_0} P_{TDC_1}}{2res}. \tag{48}$$

To make sure the TDCs finish resolving before the DCs finish accumulating jitter for the next output bit, the TDC resolving time should be smaller than the maximal DC propagation delay: $d_{TDC} < d_{DC}$. This constraint imposes an upper bound to the TDC resolution:

$$res > \frac{P_{TDC_0} P_{TDC_1}}{2\max(T_0^n, T_1^n)}. \tag{49}$$

5.2 Entropy density

According to [KS11], a minimal Shannon entropy density of 0.997 bit/bit at the output is required. The stochastic model from Sect. 3 is used to determine the theoretical entropy density at the output. Timing jitter will accumulate proportional to a square root with respect to accumulation time (addition of independent variances). A maximal TDC resolution size is required to be able to extract the required entropy from the accumulated DC timing jitter. This observation leads to an upper bound on the required TDC resolution, given as:

$$res < \alpha \sqrt{F_{noise} \max(T_0^n, T_1^n)}, \tag{50}$$

with α a constant related to the required entropy density and the shape of accumulated timing jitter distribution. The value of α can now be determined by evaluating the model from Sect. 3 for multiple values of DC accumulation time and searching for the upper bound on the required resolution.

Figure 13 shows the model results. As can be observed, the obtained α is not perfectly constant. A lower bound (horizontal line in Fig. 13) is selected, such that Eq. 50 will always produce a valid upper bound for the TDC resolution. The value for α used in the remainder of this work equals 1.94.

5.3 ES Throughput

The ES throughput is related to the DC accumulation time:

$$throughput = \frac{1}{\max(T_0^n, T_1^n)}. \tag{51}$$

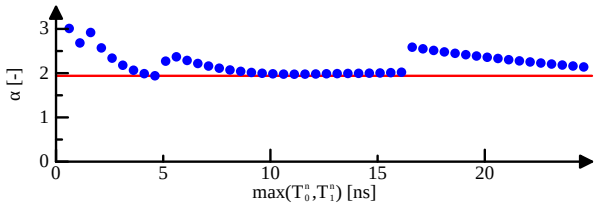

Figure 13: Minimal α required for Eq. 50 to produce a high enough entropy density, with $F_{noise} = 30\,\text{fs}$.

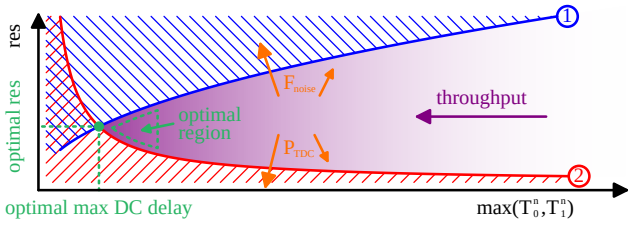

Figure 14: TDC resolution versus DC accumulation time optimisation.

A sketch, showing both constraints for the TDC resolution versus DC accumulation time is depicted in Fig. 14. The top curve (1) and bottom curve (2) visualise Eqs. 50 and 49 respectively. Valid values for the TDC resolution are marked by the shaded region in between the two constraint curves. Higher ES throughput favours points more to the left of the graph. The optimal resolution/accumulation time point is at the intersection of both constraint curves. To have a sufficiently stable ES implementation, some margin from the constraint borders is required, which is indicated by the optimal region in Fig. 14.

5.4 Delay control circuit

To enable the throughput optimisation procedure, a fine control over the DC accumulation time (T_0^n and T_1^n) and the TDC resolution is required. Figure 15 shows a circuit breakdown for both the DC and TDC ROs. Each DC RO consists of four stages and each stage contains five inverters of decreasing effective length.

Some inverters (indicated in Fig. 15) can be switched on/off by controlling a configuration input as shown at the right of Fig. 15. When an inverter is switched on, its output current is used to accelerate (dis)charge of the load capacitance, reducing the stage propagation delay. When the inverter is switched off, it still contributes to the load capacitance seen by the previous stage, further increasing the propagation delay.

Both TDC ROs consist of two stages and each stage contains eight identical minimal sized inverters that have the same on/off control. The inverters do not require analog voltages to be configured, they are either turned fully on or fully off. Having an all digital design removes the need for additional circuitry to generate analog voltages on chip.

Both DC and TDC ROs have 16 configuration bits each, driven by a controller circuit external to the device. Given the architecture in Fig. 15, for all devices tested, a configuration in the optimal region could always be found.

6 Experimental results

Five devices containing the proposed ES architecture have been manufactured using a 28 nm CMOS technology. Unless explicitly stated otherwise, all measurements are performed

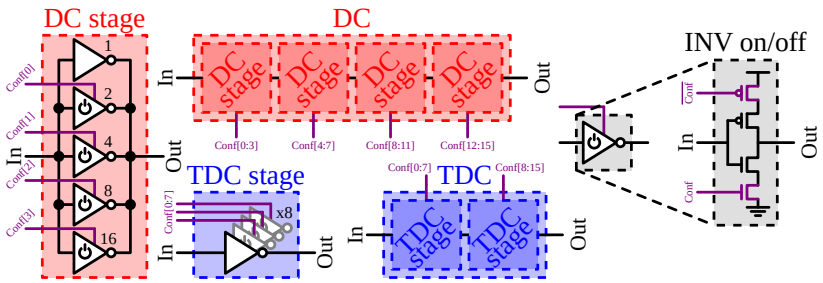

Figure 15: Detailed DC/TDC architecture.

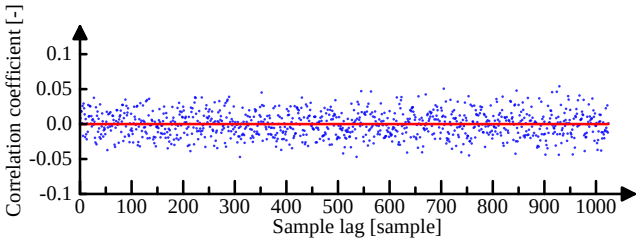

Figure 16: Measured sample correlation from 4096 samples.

at nominal conditions: 20 °C environment temperature and 0.9 V supply voltage. For each device, the DC and TDC ROs are configured to obtain an operating point inside the optimal region, as explained in Sect. 5. This was achieved by scanning all DC and TDC RO frequencies and selecting an optimal combination.

6.1 IID claim verification

As claimed in Sect. 3.5, the output bits are by design IID. Two experiments are performed to verify this claim: a correlation analysis of the generated counter (R) values, and the NIST SP 800-90B IDD test [TBK+18].

Correlation analysis The sample correlation coefficient of 4096 consecutively generated counter samples (realisations of R) from chip 0 is calculated for sample lags ranging from 1 to 1024. The sample correlation coefficient is calculated as:

$$correlation(lag) = \frac{\sum_{i=1}^{3072}(r_i - \bar{r})(r_{i+lag} - \bar{r})}{\sqrt{\sum_{i=1}^{3072}(r_i - \bar{r})^2 \sum_{i=1}^{3072}(r_{i+lag} - \bar{r})^2}}, \tag{52}$$

with r_i the i-th generated sample and $\bar{r}$ the sample mean. The results in Fig. 16 show no significant sample correlation, further strengthening the IID assumption.

NIST SP 800-90B IID test All five devices pass the NIST SP 800-90B IID test, using 1 Mbit of consecutively generated bits.

6.2 Entropy validation

As minimally required by [KS11], the estimated output bit min entropy should be larger than 0.91 per bit (equals 0.997 bits of Shannon entropy). In Sect. 5, the ES design parameters have been selected to output at least 0.91 bit of min entropy, higher entropy levels are possible at the cost of reduced throughput. Table 1 provides an overview of

Table 1: Min entropy estimates.

Chip	0	1	2	3	4
Model estimate	0.99988	0.99861	0.99811	0.99895	0.99963
Test estimate	0.93341	0.94475	0.94722	0.95255	0.96221
Minimum	0.93341	0.94475	0.94722	0.95255	0.96221

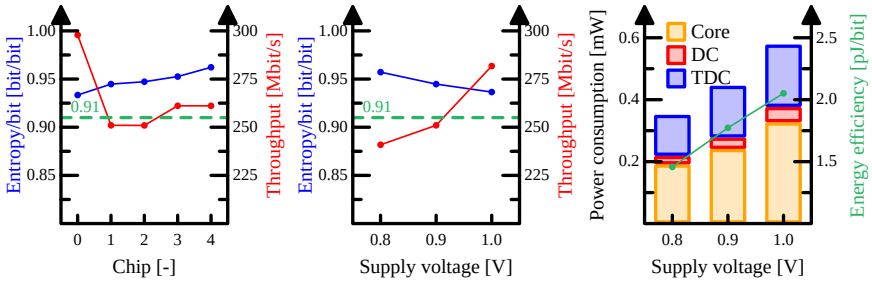

Figure 17: Measurement results.

the output min entropy estimates for all five devices tested, using 1 Mbit of consecutive data at nominal conditions. Each of the devices reach the required min entropy level. The entropy estimate obtained from the NIST SP 800-90B tests with 1 Mbit of data is even more conservative than the one obtained from the stochastic model, which is expected as these tests tend to underestimate the available entropy [Saa21]. The counter output (R) could be used as a health metric, indicating a possible entropy reduction.

6.3 Power and throughput

All five devices tested achieve a throughput of over 250 Mbit/s at nominal conditions, as can be seen in the left graph of Fig. 17. Process variations in the DC/TDC ROs can lead to some devices having better/worse performance. One device (chip 1) has been extensively tested at different voltage conditions. The experimental results in the middle graph of Fig. 17 show that for all supply voltage levels tested, the output bit entropy remained above the 0.91 bit/bit threshold.

The right graph of Fig. 17 shows the power consumption breakdown and energy efficiency per generated bit. Best energy efficiency is achieved at 0.8 V supply: 1.46 pJ/bit, which is lower than previous reported. The power breakdown shows that the Core, DC and TDC consume 54.2 %, 8.2 % and 37.6 % of the total power consumption respectively at nominal conditions. The core module contains the digitisation and synchronisation circuitry.

7 Conclusion and comparison

7.1 Comparison

Compared to previous work in Table 2, the proposed design achieves best energy and second best area efficiency (throughput generated per unit of normalised area). The jitter pipelining architecture together with high TDC time resolutions allows for high throughput at a modest area and power requirement. A chip photo is depicted in Fig. 18. The ES circuitry (DC, TDC and core) occupies 750.7 μm^2. Additional configuration flip-flops, to store the DC/TDC configuration (Conf) and interfacing logic (Send) are added to measure out the devices.

Table 2: Comparison with previous work.

	This work	ISSCC 2021 [TRA21]	ISSCC 2017 [KLK17]	JSSC 2016 [YBS16]	JSSC 2012 [MSA+12]	Cryptogr. 2021 [KHL21]
Technology [nm]	28	28	65	40	45	65
Entropy source	Edge jitter	SRAM leakage jitter	Edge jitter	Edge jitter	Meta-stability	Edge jitter
Stochastic model available	✓	✗	✗	✓	✓	✓
All digital	✓	✓	✗	✓	✗	✓
Area [kF^2]	957.5	**36***	218	522.5	1977	59.2
Max throughput [Mbit/s]	298	3.6	9.9	2	**2400**	8.27
Best energy efficiency [pJ/bit]	**1.46**	9.6	35.5	11	2.9	128.2
Best area efficiency [bit/s/F^2]	311.2	100	45.4	3.83	**1214**	139.7
Supply voltage range [V]	0.8 - 1.0	0.8 - 1.0	1.08 - 1.2	0.6 - 0.9	0.28 - 1.35	-

* SRAM area not included

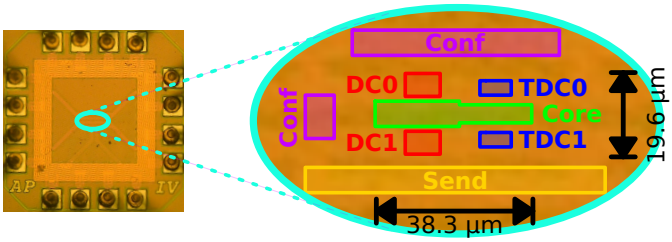

Figure 18: Chip photo with zoomed in region on the ES area.

7.2 Conclusion

The proposed ES architecture was designed and verified following an approach compatible with modern standards. Thanks to the digital nature of the circuits used, this design gains all benefits related to digital CMOS, such as scaling and design integration. A stochastic model capable of estimating the output bit entropy is presented, together with an on-chip jitter measurement methodology to quantify the jitter strength platform parameter. An optimisation scheme is presented to guide the design parameter selection process and to ensure maximal throughput is obtained for the given platform parameters. The jitter pipelining structure allows for efficient (both in terms of area and energy usage) on-chip entropy generation.

Acknowledgements

This work was supported by CyberSecurity Research Flanders with reference number VR20192203, in part supported by the Research Council KU Leuven (C16/15/058) and by the European Commission through the Horizon 2020 research and innovation programme Cathedral ERC Advanced Grant 695305. Adriaan Peetermans is funded by an FWO fellowship.

References

[Abi06] Assad Abidi. Phase noise and jitter in CMOS ring oscillators. *IEEE J. Solid State Circuits*, 41(8):1803–1816, 2006.

[BBF09] Nathalie Bochard, Florent Bernard, and Viktor Fischer. Observing the randomness in RO-based TRNG. In Viktor K. Prasanna, Lionel Torres, and René Cumplido, editors, *ReConFig'09: 2009 International Conference on Reconfigurable Computing and FPGAs, Cancun, Quintana Roo, Mexico, 9-11 December 2009, Proceedings*, pages 237–242. IEEE Computer Society, 2009.

[CCF+18] Mathieu Coustans, Abdelkarim Cherkaoui, Laurent Fesquet, Christian Terrier, Stephanie Salgado, Thomas Eberhardt, and Maher Kayal. Subthreshold logic for low-area and energy efficient true random number generator. In *2018 IEEE Symposium in Low-Power and High-Speed Chips, COOL CHIPS 2018, Yokohama, Japan, April 18-20, 2018*, pages 1–3. IEEE Computer Society, 2018.

[CFAF13] Abdelkarim Cherkaoui, Viktor Fischer, Alain Aubert, and Laurent Fesquet. A self-timed ring based true random number generator. In *19th IEEE International Symposium on Asynchronous Circuits and Systems, ASYNC 2013, Santa Monica, CA, USA, May 19-22, 2013*, pages 99–106. IEEE Computer Society, 2013.

[Dic03] Markus Dichtl. How to predict the output of a hardware random number generator. In Colin D. Walter, Çetin Kaya Koç, and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2003, 5th International Workshop, Cologne, Germany, September 8-10, 2003, Proceedings*, volume 2779 of *Lecture Notes in Computer Science*, pages 181–188. Springer, 2003.

[ISO19] Information technology – Security techniques – Test and analysis methods for random bit generators within ISO/IEC 19790 and ISO/IEC 15408. Standard, International Organization for Standardization, Geneva, CH, October 2019.

[KG04] Paul Kohlbrenner and Kris Gaj. An embedded true random number generator for FPGAs. In Russell Tessier and Herman Schmit, editors, *Proceedings of the ACM/SIGDA 12th International Symposium on Field Programmable Gate Arrays, FPGA 2004, Monterey, California, USA, February 22-24, 2004*, pages 71–78. ACM, 2004.

[KHL21] Netanel Klein, Eyal Harel, and Itamar Levi. The cost of a true random bit - on the electronic cost gain of ASIC time-domain-based TRNGs. *Cryptogr.*, 5(3):25, 2021.

[KLK17] Eunhwan Kim, Minah Lee, and Jae-Joon Kim. 8.2 8Mb/s 28Mb/mJ robust true-random-number generator in 65nm CMOS based on differential ring oscillator with feedback resistors. In *2017 IEEE International Solid-State Circuits Conference, ISSCC 2017, San Francisco, CA, USA, February 5-9, 2017*, pages 144–145. IEEE, 2017.

[KS11] Wolfgang Killmann and Werner Schindler. A proposal for: Functionality classes for random number generators, 2011.

[MSA+12] Sanu Mathew, Suresh Srinivasan, Mark Anders, Himanshu Kaul, Steven Hsu, Farhana Sheikh, Amit Agarwal, Sudhir Satpathy, and Ram Krishnamurthy. 2.4 Gbps, 7 mW all-digital PVT-variation tolerant true random number generator for 45 nm CMOS high-performance microprocessors. *IEEE J. Solid State Circuits*, 47(11):2807–2821, 2012.

[Saa21] Markku-Juhani O. Saarinen. On entropy and bit patterns of ring oscillator jitter. In *Asian Hardware Oriented Security and Trust Symposium, AsianHOST 2021, Shanghai, China, December 16-18, 2021*, pages 1–6. IEEE, 2021.

[TBK+18] Meltem Sönmez Turan, Elaine Barker, John Kelsey, Kerry A. McKay, Mary L. Baish, and Mike Boyle. Recommendation for the entropy sources used for random bit generation, 2018.

[TRA21] Sachin Taneja, Viveka Konandur Rajanna, and Massimo Alioto. 36.1 unified in-memory dynamic TRNG and multi-bit static PUF entropy generation for ubiquitous hardware security. In *IEEE International Solid-State Circuits Conference, ISSCC 2021, San Francisco, CA, USA, February 13-22, 2021*, pages 498–500. IEEE, 2021.

[VD10] Michal Varchola and Milos Drutarovský. New high entropy element for FPGA based true random number generators. In Stefan Mangard and François-Xavier Standaert, editors, *Cryptographic Hardware and Embedded Systems, CHES 2010, 12th International Workshop, Santa Barbara, CA, USA, August 17-20, 2010. Proceedings*, volume 6225 of *Lecture Notes in Computer Science*, pages 351–365. Springer, 2010.

[YBS16] Kaiyuan Yang, David T. Blaauw, and Dennis Sylvester. An all-digital edge racing true random number generator robust against PVT variations. *IEEE J. Solid State Circuits*, 51(4):1022–1031, 2016.

[YFH+14] Kaiyuan Yang, David Fick, Michael B. Henry, Yoonmyung Lee, David T. Blaauw, and Dennis Sylvester. 16.3 A 23Mb/s 23pJ/b fully synthesized true-random-number generator in 28nm and 65nm CMOS. In *2014 IEEE International Conference on Solid-State Circuits Conference, ISSCC 2014, Digest of Technical Papers, San Francisco, CA, USA, February 9-13, 2014*, pages 280–281. IEEE, 2014.

[YRG+17] Bohan Yang, Vladimir Rozic, Milos Grujic, Nele Mentens, and Ingrid Verbauwhede. On-chip jitter measurement for true random number generators. In *2017 Asian Hardware Oriented Security and Trust Symposium, AsianHOST 2017, Beijing, China, October 19-20, 2017*, pages 91–96. IEEE Computer Society, 2017.