Analog Integr Circ Sig Process (2016) 87:155–167
DOI 10.1007/s10470-015-0631-y

# Very low cost chaos-based entropy source for the retrofit or design augmentation of networked devices

**Sergio Callegari**[1] · **Mattia Fabbri**[1] · **Ahmad Beirami**[2]

Received: 1 May 2015 / Accepted: 2 September 2015 / Published online: 23 September 2015


**Abstract** Modern cryptographic protocols require good entropy sources. Unfortunately, many networked devices lack subsystems dedicated to this task, being potentially susceptible to random number generator (RNG) attacks. Yet, most of these systems allow software upgrades and host communication ports, providing the option of a retrofit. This work illustrates how chaotic dynamics can be used to design a sub-10$ entropy source capable of an over 48kbit/s rate and offering multiple serial communication abilities. Operation is based on a standard microcontroller and exploits a loop built around one of its analog to digital converters (ADCs). The design offers self-testing features and enables an experimental validation of some recent results on the choice of the best state quantization function to employ when using chaotic maps as RNGs.

**Keywords** True-RNG · Chaotic map · Hidden Markov process · Analog to digital converter (ADC) · Microcontroller

## 1 Introduction

One of the major areas of concern in information technology is certainly that of security. The issue is aggravated by the recent trend of networking an immense number of devices that were meant to operate in isolation until a recent past. Forecasts suggest that by 2020 more than 25 billion devices will be interconnected [1]. This *internet of things* (IoT) can bring smartness to many environments whose items gain the ability to cooperate but, at the same time, creates an immense attack surface [2]. Moreover, many devices lack important security-oriented subsystems since the IoT paradigm often applies to mundane items for which cost strongly influences engineering and awareness on security is still building up. As a consequence, a legacy of security-critical networked apparatuses is being created, probably to be long-lasting since humble devices are usually replaced only when they break. This outlook is supported by the observation that already half of the networking devices found in business are on the brink of obsolescence [3]. Such a dangerous phenomenon can be contrasted in two major ways: (i) conceiving security subsystems that can be used to economically upgrade/augment current designs; (ii) implementing security subsystems that can be added as low-cost retrofits to current devices.

Both strategies are sufficiently obvious when dealing with software units, less so for subsystems that must necessarily be hardware based. This is the case of good random number generators (RNGs), that are required by virtually all applications based on cryptography [4]. Poor unpredictability results in potential susceptibility to RNG attacks, that cannot be completely excluded unless one employs generators based on *physical* entropy sources [5], namely true-RNGs (TRNGs). Such attacks should not be underestimated since a significant list of past incidents already exists [6]. TRNGs and hardware entropy sources are now becoming mainstream in computers [7] but are still missing from most low-cost/embedded devices.

This paper illustrates the design of a chaos based entropy source suitable for TRNGs. The proposal is declined in two

✉ Sergio Callegari
sergio.callegari@unibo.it

Mattia Fabbri
mattia.fabbri2@studio.unibo.it

Ahmad Beirami
ahmad.beirami@duke.edu

[1] ARCES and DEI, University of Bologna, Bologna, Italy

[2] Duke University, Durham, NC 27708, USA

forms. The first one is a self-contained plugin applicable as a retrofit to existing systems on field. A prototype with Universal Serial Bus (USB) connection is shown in Fig. 1. Attachments based on Inter-Integrated Circuit ($I^2C$), Serial Peripheral Interface (SPI), or Universal Synchronous/Asynchronous Receiver/Transmitter (USART) data transfers are also possible. The second form consists in a schematic upgrade that a manufacturer can apply during revision of existing products with minimal disruption. In the first case, the estimated cost is below 10$, while in the second one it can fall below 5$, subject to the availability of a microcontroller (μC) with spare capacity in the product. In either case, the entropy rate can be in the few tens kbit/s depending on the μC (with a PIC18F2550, rates in excess of 48kbit/s have been experimentally observed[1]).

The market for retrofittable entropy sources is growing, with solutions pricing approximately in the 40–1000$ range (i.e., at far higher costs, even considering selling margins). Closest competitors include the TRNG98 modules [8], the Simtec Entropy Key [9], the OneRNG [10], the NeoG FTS-01. The public domain Infnoise TRNG design [11] is also worth mentioning. Yet, the current proposal differs from all the others in being suitable not just as a pluggable USB dongle, but also as a design augmentation for existing products. Another distinguishing note is the use of chaotic dynamics. Most alternatives rely on the amplification of noise on a specific electronic component (a resistor, an avalanche diode, etc.). This makes shielding mandatory as the latter may also become a noise injection point for tampering. Conversely, the use of chaos makes the system inherently less susceptible to intentional interference and side channels attacks [12].

Chaotic dynamics has already been used in a large number of engineering applications, ranging from spread spectrum communication [13] to signal synthesis [14], from electromagnetic interference reduction [15] to analog testing [16, 17], etc. Certainly, this is not the first time that chaos is proposed for RNGs [18–20]. Incidentally, also the above mentioned Infnoise TRNG is ultimately chaos-based, even if the authors fails to recognize it. However, the current proposal can reuse an analog to digital converter (ADC) readily available on a μC, which simplifies the implementation and, at the same time, makes it more flexible. The use of an ADC as the main building block for a chaotic TRNG was first introduced in [21] and has already seen experimental validation but, until now, only on full custom integrated circuits with specially crafted pipeline data converters [12]. The present design, that made its debut in [22], shows the possibility of relying on a commercial μC ADC with a traditional successive approximation architecture. The approach takes advantage of the theory developed in [17, 21, 23] and is consistent with recent ideas on how TRNG should be validated and designed for test [24–26]. With respect to [22], this followup illustrates more design options and highlights how the flexibility offered by the use of a μC ADC enables an experimental verification of some recent concepts on the fundamental limits of chaotic maps as RNGs [27] and on the choice of the best way to deliver a digital output from the map analog state.

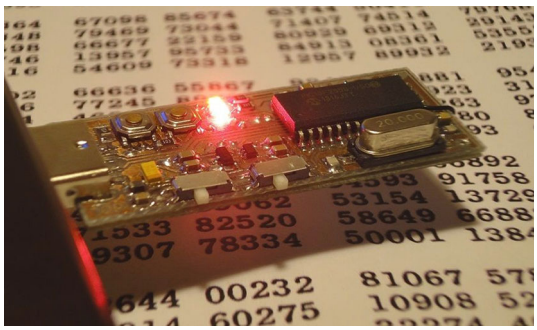


**Fig. 1** Prototype of self-contained entropy source with USB port. Diagnostic LEDs, control switches, trimmers, test-pads, and a wide SOIC μC package significantly enlarge the layout

The paper is organized as follows:

- Section 2 provides background material. It reviews RNG security requirements and discusses TRNG architectures and entropy source classes;
- Section 3 describes the theory of operation of the proposed entropy source;
- Section 4 considers the implementation of the proposed entropy source, also dealing with robustness and testability;
- Section 5 focuses on the design of alternative (and better) ways to obtain a digital output from the chaotic map;
- Section 6 provides experimental results, confirming correct operation and proving the applicability of some recent results on chaos based entropy sources.

## 2 Security, RNGs, and entropy sources

The security of cryptographic systems relies on the existence of some secret data known to authorized users and unpredictable by others. To warrant unpredictability, random strings are often employed (e.g., in *keys*, *salts*, *nounces*, *challenges*, *initialization vectors*, and other one-time quantities). In many cases, recommended lengths for these strings are precisely based on the assumption of a random synthesis (leading to equidistributed and independent bits). However, the generation of these strings is not trivial, because, by nature, digital systems cannot behave randomly. The traditional solution has been to substitute

[1] Considering the overhead of USB data transmission.

*computational complexity* for real unpredictability by the use of so called pseudo-RNGs (PRNGs), namely finite state automata capable of expanding short *seeds* into long sequences with good statistical features [25, 28]. A well designed automaton makes the estimation of the current state from past outputs an exceedingly intensive computational task, so achieving *practical* security. It is worth recalling that discovery of the state by an attacker would result in a disclosure of all the future "random" quantities with a full breakage of the cryptographic measures [4] based on them. Unfortunately, with PRNGs, such an event cannot be *thoroughly* excluded and some high-impact incidents have happened in the past (typically, because of bad automata, bad seeding or information leaks) [6].

The ultimate solution is provided by TRNGs, that typically consist of a physical device (i.e., with analog nature) capable of expressing a noisy behavior, followed by an acquisition (digitalization) mechanism, and by a digital processing stage, often indicated as *entropy distiller*, as in Fig. 2. The first block has the fundamental role of harvesting information from some phenomena so intricated at the microcosmic level to appear random at the macroscopic scale [22, 24]. Together, the physical device and the quantization mechanism constitute an *entropy source*. At its output, the statistical features (distribution and high order moments) may still be not perfect and the entropy rate may be lower than the bit rate (the concept is formalized in Sect. 5). In [24, 25], data at this point is indicated as digitized analog signal (DAS) random numbers. Post-processing is then applied to de-bias and de-correlate the sequences. The amount of post-processing can vary significantly, depending on the quality of the entropy source. The key operation is *mixing*, which generally implies a *dynamical* processing. Data rate reductions are also typically applied to match the bit and information rates, otherwise it is theoretically impossible to have true random data [27, 29, 30]. Hence, the higher the entropy per bit in the DAS numbers, the better. Practical post-processors often involve hash functions or PRNGs constantly re-seeded from the DAS words. Following [24, 25], the entropy distillation delivers so-called *internal* random numbers. Finally, some output logic lets random bits be pulled from TRNG as needed, delivering *external* random numbers.

Nowadays, big players of information technology tend to incorporate full fledged TRNGs in their larger and newer systems [7, 31]. However, being the entropy source the most delicate and expensive part, many low-cost or legacy systems still compromise on it. Frequently, software solutions are employed capable of gathering entropy from peripherals that end up "observing" physical phenomena complex enough to be assumed random as a side effect of their primary duty. For instance, many operating systems have access to mouse movements, key press timings, hard

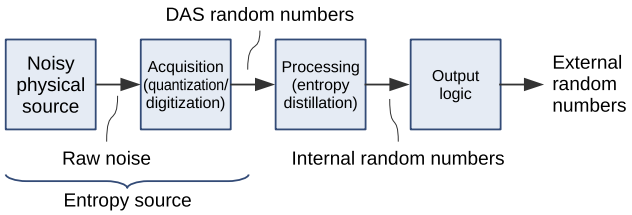



**Fig. 2** Basic building blocks of a TRNG

disk latencies, etc, which are all highly irregular quantities. By their use, one gets what some recent standards call *non-physical* TRNGs [25]. This is somehow a misnomer, since the data *is* ultimately derived from physical sources, yet the latter are not *co-designed* with the rest of the system, nor fully under its control. In this setup, one generally has multiple entropy sources with the distiller taking the additional role of mixing among them. This works fairly well for standard computers, but may fail badly for little devices in an IoT scenario. In fact, the latter may lack most of the peripherals needed to collect entropy (for instance, little routers, networked thermostats, and similar systems do not have mouses, keyboards or hard-disks). With this, entropy distillation may become counterproductive, giving a false sense of security while one falls back to a PRNG behavior due to absence of inputs.

Clearly, the only real fix is to bring physical TRNGs also to humble devices. Doing so in the form of hardware retrofits requires a careful weighing the available options. The most straightforward physical sources are based on the direct amplification and acquisition of noise on an electronic component, such a resistor or a junction device. This can be effective but is inherently exposed to tampering, since interference may be easily exchanged from genuine noise. Hence, shields, that can be expensive and bulky, are generally mandatory [8, 10]. Alternative forms of noise amplification exploit the meta-stability of positive feedback structures, races in signal paths, or phase noise in oscillators that mutually sample each other. The latter techniques enable a "digital" design, since the analog quantity becomes time. Furthermore, tampering is made harder by the more indirect harvesting. Finally, one can adopt circuits based on chaotic dynamics, which is probably the most sophisticated approach. First of all, chaotic-RNGs replicate at the circuit level some traits, including sensitivity to initial conditions and other stochastic features [32, 33], that are conjectured to be at the root of randomness in natural systems. Secondly, they harvest noise in a quite different way than non-chaotic sources. The latter require some element to continuously provide new noise samples. Conversely, chaotic sources could in principle rely just on the uncertainty in their start-up state thanks to sensitivity to initial conditions. In practice, noise will

anyway be present throughout all the operating time, but this ends up as a bonus not strictly required for correct operation. As a side effect, noise and interference collected during operation normally have a very low influence on the output quality (unless they are rather large, a case where tampering can typically be detected with ease).

## 3 Operating principle of the proposed entropy source

When one designs a chaotic source in view of engineering applications, it makes sense to pick the simplest model for the task, choosing among those for which mathematical analysis tools are best developed [34]. For TRNGs, one dimensional (1-D) maps are particularly appealing. These are based on the recurrence relation

$$x_{n+1} = M(x_n), \tag{1}$$

where $x$ is a scalar and $M(\cdot)$ is a nonlinear function that admits a finite invariant set, here normalized to $[0, 1]$ with no loss of generality. A notable mathematical tool is then given by the Perron-Frobenius operator (PFO) $\mathbf{P}_M$ associated to $M(\cdot)$ that lets one observe the propagation of probability density functions (PDFs) through the map [33, 34]. Namely, if $x_0$ is drawn in $[0, 1]$ according to a PDF $\rho_0(\cdot)$, one can compute the PDF associated to $x_1 = M(x_0)$ as $\rho_1(x) = \mathbf{P}_M[\rho_0](x)$. The PFO is *linear*. Under some relatively easy to satisfy conditions over $M(\cdot)$, namely its being *mixing* or *exact*, it can be proved that a single invariant density $\bar{\rho}(\cdot)$ exists (i.e., there is one and only one $\bar{\rho}(\cdot)$ for which $\bar{\rho} = \mathbf{P}_M[\bar{\rho}]$). Furthermore, the trajectory of the map distributes according to it for non-singular initial conditions [34].

Among 1-D maps, it is then convenient to further restrict to piecewise-affine Markov (PWAM) ones. For them, an interval partition $\mathcal{P} = \{I_0, \ldots, I_{P-1}\}$ of $[0, 1]$ exists such that: the map is affine on all the intervals $I_i$; and $I_i \subseteq M(I_j) \vee I_i \cap M(I_j) = \emptyset$ for any couple of intervals $I_i, I_j$ (this last statement corresponds to the requirement that the image of a partition interval is a union of partition intervals). The PWAM property ensures the existence of a finite-dimensional restriction of $\mathbf{P}_M$ to densities that are step-wise over $\mathcal{P}$. Furthermore, $\bar{\rho}(x)$ belongs to such subset of densities, so that it can be found via the restricted PFO. The latter can be expressed through a *kneading matrix* $\boldsymbol{K} = (k_{i,j})$ such that $k_{i,j}$ is the fraction of $I_i$ that is mapped into $I_j$, namely

$$k_{i,j} = \frac{\mu(I_i \cap M^{-1}(I_j))}{\mu(I_i)} \quad \text{for i, j} \in \{0, \ldots, \mathrm{P} - 1\} \tag{2}$$

where $\mu(\cdot)$ is the usual interval measure. For mixing or exact PWAM maps, $\boldsymbol{K}$ can also be interpreted as the transition matrix of a Markov chain describing the coarse dynamics of the system, as observed through a partition-induced quantization function $q(\cdot)$. Clearly, the unique left eigenvector $\mathbf{e}$ of $\boldsymbol{K}$ with unitary eigenvalue, normalized so that $\sum_{i=0}^{P-1} e_i = 1$, provides the probabilities of finding the system state $x_n$ in $I_0, \ldots, I_{P-1}$ via its entries $e_0, \ldots, e_{P-1}$.

Among exact PWAM maps, so-called *shift maps* can be particularly appealing for RNGs. These are based on expressions such as $x_{n+1} = \alpha x_n \bmod 1$ where $\alpha \in \mathbb{N}$ and $\alpha \geq 2$. Here, the Markov partition is obtained by the mere subdivision of [0,1] in $\alpha$ equally sized sub-intervals. In practice, coarse modeling through a Markov chain can also be done with maps that are a slight generalization of the above, as in

$$x_{n+1} = (\alpha x_n + \beta) \bmod 1, \tag{3}$$

where $\beta$ is an arbitrary quantity. Furthermore, one can use any partitioning of $[0, 1]$ in $\alpha$ equally sized "wrapped" intervals such as

$$I_i = \{x : x \in [0, 1] \wedge \lfloor \alpha((x - p) \bmod 1) \rfloor = i\} \tag{4}$$

with $i = 0, \ldots \alpha - 1$ and $p \in \mathbb{R}$, as it can be easily verified by substituting a circle on a plane for the set $[0, 1]$ so that intervals naturally wrap around. In this case, one gets a uniform $\bar{\rho}(\cdot)$ and an $\alpha \times \alpha$ matrix $\boldsymbol{K}$ whose entries are all $1/\alpha$. Figure 3 shows an example with $\alpha = 3$. With the conditions above, the coarse dynamics is modeled by a chain equivalent to that describing the cast of an ideal die with $\alpha$ faces. Thus, the suitability of the model for a TRNG is obvious.

Interestingly, a map $M(\cdot)$ as in Eq. (3) can easily be obtained out of an ADC, as exemplified in Fig. 4. Let $f_{\mathrm{ADC}}(\cdot)$ and $f_{\mathrm{DAC}}(\cdot)$ be used to indicate the static input-output relationships of the ADC and the DAC, respectively. For an $N$ bit unipolar rounding-down[2] ADC, within the conversion range one should ideally have

$$f_{\mathrm{ADC}}(V) = \left\lfloor \frac{2^N}{V_{\mathrm{ref}}} V \right\rfloor, \tag{5}$$

where $V_{\mathrm{ref}}$ is the reference voltage (approximately equal to the full scale) and the quantization step is $V_{\mathrm{ref}}/2^N$. For the complementary $N$ bit DAC one has

$$f_{\mathrm{DAC}}(m) = m \frac{V_{\mathrm{ref}}}{2^N}. \tag{6}$$

so that the quantization error, derived as in the figure, is

$$V_{\mathrm{err}} = V_{\mathrm{in}} - f_{\mathrm{DAC}}(f_{\mathrm{ADC}}(V_{\mathrm{in}})) = V_{\mathrm{in}} - \frac{V_{\mathrm{ref}}}{2^N} \left\lfloor \frac{2^N}{V_{\mathrm{ref}}} V_{\mathrm{in}} \right\rfloor = \frac{V_{\mathrm{ref}}}{2^N} \left( \frac{2^N}{V_{\mathrm{ref}}} V_{\mathrm{in}} - \left\lfloor \frac{2^N}{V_{\mathrm{ref}}} V_{\mathrm{in}} \right\rfloor \right) = \frac{V_{\mathrm{ref}}}{2^N} \left( \frac{2^N}{V_{\mathrm{ref}}} V_{\mathrm{in}} \bmod 1 \right). \tag{7}$$

[2] Similar results can be obtained with any other converter type.

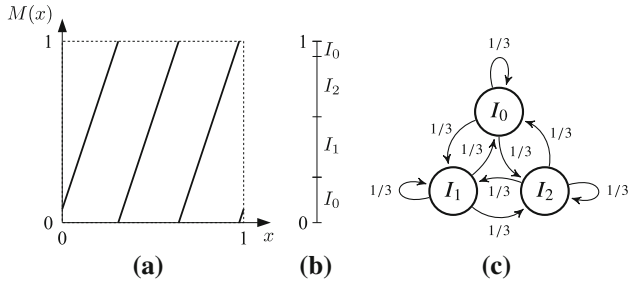



**Fig. 3** Sample shift map with $\alpha = 3$ (a), together with a Markov partition (b) and the corresponding Markov chain (c). For the specific case, $\beta = 3/40$ and $p = -1/12$

Now, let a further block, as shown in Fig. 4(b), deliver $V_{\text{out}} = k(V_{\text{err}} + V_B)$. One gets

$$V_{\text{out}} = \frac{kV_{\text{ref}}}{2^N}\left(\frac{2^N}{V_{\text{ref}}}V_{\text{in}} \bmod 1\right) + kV_B \tag{8}$$

so that

$$\frac{2^N}{kV_{\text{ref}}}(V_{\text{out}} - kV_B) = \left(k\frac{2^N}{kV_{\text{ref}}}(V_{\text{in}} - kV_B) + k\frac{2^N}{V_{\text{ref}}}V_B\right) \bmod 1. \tag{9}$$

By introducing the $(V, x)$ transform pair

$$x = \frac{2^N}{kV_{\text{ref}}}(V - kV_B) \qquad V = \frac{kV_{\text{ref}}}{2^N}x + kV_B \tag{10}$$

it is trivial to see that Eq. (9) defines the map $M(\cdot)$, with $\alpha \equiv k$ and $\beta \equiv k2^N V_B/V_{\text{ref}}$. To achieve a recurrence relation such as Eq. (1), it is then sufficient to feed back $V_{\text{out}}$ into $V_{\text{in}}$ through an analog register, as in Fig. 4(c) which employs two (T/H) blocks in a master–slave arrangement. The resulting chaotic map can be as illustrated in Fig. 5, as long as the parameters obey some obvious constraints, namely $2 \le k \le 2^N$ (with $k \in \mathbb{N}$), $V_B \ge 0$ and $k(V_B + V_{\text{ref}}/2^N) \le V_{\text{ref}}$, i.e., $V_B \le V_{\text{ref}}/k - V_{\text{ref}}/2^N$.

The ADC can also provide the Markov partition and the corresponding quantization function $q(\cdot)$. In fact, while $x$ spans $[0, 1]$, the voltages $V_{\text{in}}$ and $V_{\text{out}}$ span $[kV_B, k(V_B + V_{\text{ref}}/2^N)]$ which is exactly as wide as $k$ quantization intervals. Thus, one can define a partition based on the ADC output value $m$, by simply taking $p = -(2^N V_B/V_{\text{ref}})$, which assures that the condition $x \in I_i$ can be read from $m$ by verifying if $m \bmod k = i$. With this, the quantization function becomes

$$q(x) = m \bmod k. \tag{11}$$

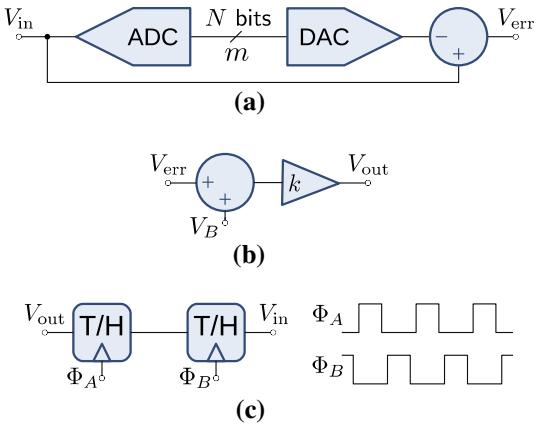



**Fig. 4** Derivation of ADC quantization error via a cascaded DAC (a); its processing (b); and the closing of the loop to create a chaotic recurrent model (c)

## 4 Practical implementation on a microcontroller

The concepts illustrated so far can be translated into a μC based architecture, taking advantage of the fact that most μCs embed at least an ADC. Figure 6 shows a possible implementation corresponding to the prototype in Fig. 1. This assumes no DAC on board, and further simplifications are possible otherwise. The proposed design implements the ADC-DAC connection in Fig. 4(a) via the μC SPI output. The difference block on the right hand side of Fig. 4(a) is obtained together with the processing block in Fig. 4(b) by a difference amplifier, so that its resistors ($R_A$ to $R_F$ in the schematic) allow $k$ and $V_B$ to be controlled. The analog register in Fig. 4(c) is obtained via the cascade of two (T/H) blocks implemented by capacitively loaded operational amplifiers with a high-impedance (shutdown) output mode. Clock signals for the register (2-phases) and the DAC are provided by the μC via 3 digital output lines. For correct operation, a voltage reference shared between the ADC and the DAC is desirable. In the figure, a dedicated component is employed, but simplifications are possible if the μCs can output an adequate $V_{\text{ref}}$.

Obviously, many variants with respect to the prototype implementation are possible (such as the use of a different μC, a different or a μC-embedded DAC, a different gain-and-offset stage or analog register, etc.). In fact, two alternative design targets can be considered:

- a standalone unit, namely an item that the final user can retrofit on an existing system;
- a design augmentation, namely an incremental schematic change that an equipment manufacturer can apply during the revision of a product.

For the first case, a design similar to the proposed prototype can be adopted, possibly with some of the above mentioned

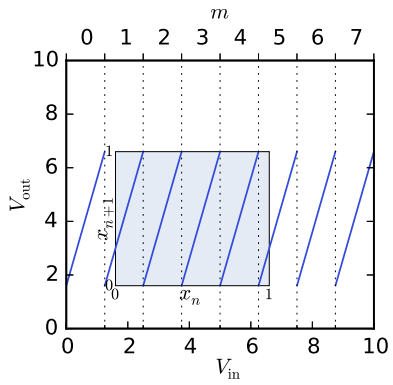



**Fig. 5** Sample map obtained from the quantization error of an ADC. For the sake of representation, $N$ is set at 3. Furthermore, $k = 4$, $V_{ref} = 10V$, $V_B = 0.4V$

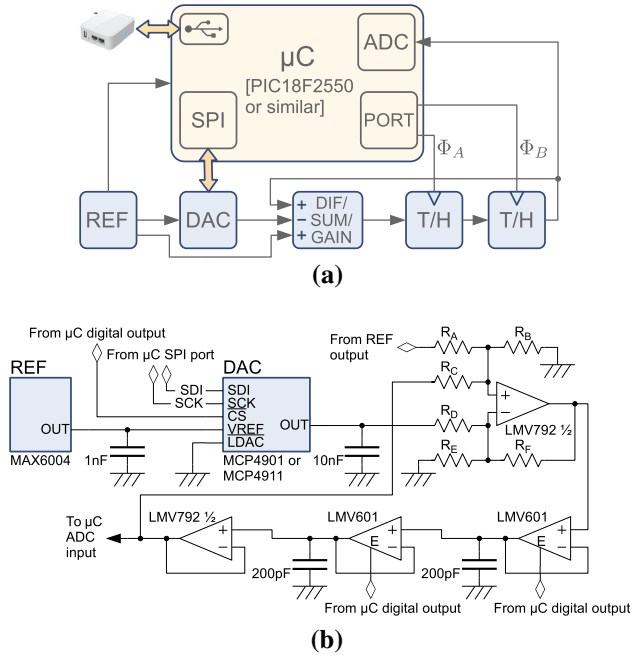



**Fig. 6** µC based architecture implementing the proposed true-RNG. Block diagram (a) and signal processing chain added to the µC (b)

changes. A key aspect is to use a µC supporting the desired communication standards. For instance, the availability of an on-board USB subsystem may be desirable. Based on the prototype bill of materials, costs can be contained below 10$. In the second case, the key aspect is to have as little disruption of the existing schematic as possible. If a µC is available in the original design with a spare ADC channel and spare computation capacity, the incremental change can be restricted to the addition of signal processing chain at the bottom of Fig. 6(b). By saving the µC and a dedicated printed circuit board (PCB), costs can be lowered below 5$.

### 4.1 Implementation requirements and parameter setting

Notwithstanding the apparent simplicity of the implementation, care is required to assure correct operation and robustness in presence of uncertainties, noise and interference. In fact, ADCs embedded in µCs can easily deliver outputs where the least significant bits (lsbs) are erratic, due to noise from digital lines and the dirtying of the power supply caused by the switching operation of large logic blocks. Unless specific arrangements are adopted (ground planes, bypass capacitors, shields, band-limiting filters), the effective number of bits (ENOB) can be much lower than the nominal resolution. The point is that none of these arrangements is desirable or even practicable for the proposed application. On one hand, items such as ground planes, shields or filters should be avoided to contain costs. On the other hand, targeting retrofitting or design augmentation, one has to deal with what is originally available in the retrofitted/augmented unit in terms of power supply quality, PCB arrangements, etc. In other words, setups that are tolerant to a poor ENOB and to relatively large amounts of noise and interference are a necessity.

The issue is addressed by assuming from the very start an ADC resolution $\hat{N}$ lower than the ENOB with some clearance. If the ADC has $\hat{N}$ bits output words and the DAC has $\hat{M}$ bits input words, the arrangement in Fig. 4(a) can be obtained by taking $N < \min(\hat{N}, \hat{M})$. This is practiced by passing to the DAC the value $\left\lfloor \hat{m} * 2^{N-\hat{N}} \right\rfloor * 2^{\hat{M}-N}$ at each cycle, where $\hat{m}$ is the ADC output. Inside the µC, this operation reduces to:

(i) pre-computing the binary word $B_{mask} = -(2^{\hat{N}-N})$ where the negative sign is rendered by two-complement notation (i.e., one has all ones, but for the $\hat{N} - N$ lsbs at zero);

(ii) passing at each cycle $(\hat{m} \,\&\, B_{mask}) \gg (\hat{N} - \hat{M})$ to the DAC, where the operator & indicates a bitwise AND, and $\gg$ indicates a right bit shift.

This arrangement lets one have $V_{err}$ always resolved with sufficiently good relative accuracy. Furthermore, it makes the computation of $m$ from $\hat{m}$ as easy as the computation of $\hat{m} \gg (\hat{N} - N)$. To appreciate what suitable $N$ values can be, consider that conventional ADCs working inside (or alongside) µCs can easily get ENOB values as low as 5-6 bits in absence of noise-limiting measures [35]. Thus, good $N$ values can be in the 3–4 bit range.

Taking $N$ so much lower than $\hat{N}$ (that can easily fall in the 8–12 bit range), may at first appear penalizing because it translates in a data-rate limitation. In fact, it implies the use of small $k$ values, which cause the random symbol

generated at each cycle to be defined on a more restricted alphabet (so carrying less information). Yet, it must be observed that some bounding of $k$ to relatively low values would anyway be required, because the proposed architectures ends up amplifying by $k$ not just the useful signal, but also all the errors. Furthermore, the choice of $N<\hat{N}$ also comes with advantages. Notably, it increases testability (as shown in Sect. 4.2). Moreover, it provides greater flexibility in the choice of the quantization function (as shown in Sect. 5).

For what regards the particular choice of $k$, it is convenient to set it at a power of 2. With this, the symbol generated at each cycle can be perfectly encoded in $N_c = \log_2(k)$ bits, so that, under ideal conditions, the system generates $N_c$ random bits per cycle. If $k$ is a power of 2, under the partition and quantization scheme illustrated in Sect. 3, the DAS number generated at each cycle can be extracted from $\hat{m}$ as

$$w_n = (\hat{m}_n \gg (\hat{N} - N)) \,\&\, (k-1). \tag{12}$$

In the above, note that the bitwise AND operation is equivalent to the modulus operation in Eq. (11). The previous considerations highlight once more that, once constraints on $k$ are put in place to assure $k<2^N$ and robust operation, $k$ should be taken as large as possible within them. For this reason, particularly good $N$, $k$ couples can be those where $k = 2^{N-1}$ as in $N = 4, k = 8$, or $N = 3, k = 4$.

Finally, for what concerns $V_B$, the discussion in Sect. 3 indicates that its accurate setting is not particularly important. It is anyway advisable to pick a $V_B$ value so that the voltage span $[kV_B, k(V_B + V_{\text{ref}}/2^N)]$ is sufficiently centered within the ADC range. This can be useful to prevent noise and interference from bringing the ADC into saturation.

### 4.2 Operation example, robustness and testability

Figure 7 shows the actual map obtained from the prototype. The nominal ADC resolution is $\hat{N} = 10$. Two cases are considered: in plot (a), $k = 4$, $N = 3$, while in plot (b), $k = 8$, $N = 4$. The effects of implementation inaccuracies are evident in the plots, including a small slope error and, most important, a relatively large noise level causing a dispersion of the experimental points around the ideal trace. In fact, $N = 4$, $k = 8$ are limit values for safe operation at the observed noise levels and, in plot (b), the dispersion is already large enough for the experimental map to get more branches than expected (11 branches instead of 9). Nonetheless, the quantization function in Eqs. (11) and (12), can deal with this, being capable to deliver a sensible output even when the analog variables falls outside its nominal range. This is confirmed by the probability distribution of the symbols $b$ generated at each cycle, which is always sufficiently uniform, as illustrated in Fig. 8. The histograms are not completely flat, but one should recall that the item under exam is a raw entropy source, not a full TRNG. In other words, under the schematization in Fig. 2, what is being observed are the DAS random numbers before any entropy distillation, not the internal/external random numbers.

Clearly, the probability distribution of symbols is not a sufficient indicator to characterize the entropy source and more data is provided in Sect. 6. Nonetheless, Figs. 7 and 8 are quite useful to introduce the topic of *testability*. Until recent, there was no assessment criterion specifically dedicated to TRNGs and statistical test suites designed for PRNGs, such as [36, 37] were customarily used. Under the schematization in Fig. 2, this resulted in checking only the external (or at best internal) random numbers, because PRNG tests are too strict for DAS numbers. Furthermore, it pushed designers to hide DAS numbers inside opaque TRNGs structures. Recently, the inadequacy of this approach has been understood [24] and more appropriate testing techniques have been formalized in a standard [25]. The major problem that has been evidenced is that the post-processing block in Fig. 2 can be remarkably similar to a PRNG. Consequently, it can often provide statistically excellent outputs even if there is a failure in the entropy source. In this case, tests succeed even if unpredictability is lost. Conversely, testing on the internal numbers alone should be acceptable only if the distiller is particularly simple, involving no long-term memory, because only in this case up-hill issues can be evidenced. Similar distillers can only be used if the correlations in the DAS numbers are already extremely small. It can be anticipated that proposed source copes quite well with this condition and can be coupled with an extremely lightweight post-processing (see Sect. 6). Still, it is always preferable to perform dedicated tests at the DAS random numbers. The proposed source, not being enclosed into an opaque TRNG package, makes DAS numbers fully accessible for testing, as the histograms in Fig. 8 testify.

In fact, the proposed design goes well beyond this requirement. Thanks to the choice of $N<\hat{N}$, at each cycle the analog variable $V_{\text{in}}$ gets acquired at a much higher resolution than that strictly needed for operation. This brings full *introspection* abilities to the system. In other words, the chaotic entropy source is capable of observing its own analog state (i.e., the main *raw noise* quantity) at a fine resolution (with $\hat{N} - N$ levels in each branch of the chaotic map). This is well proved by the experimental diagrams in Fig. 7 that have been obtained *with no recourse to bench instruments*. Such feature is quite important because modern standards indicate that TRNGs

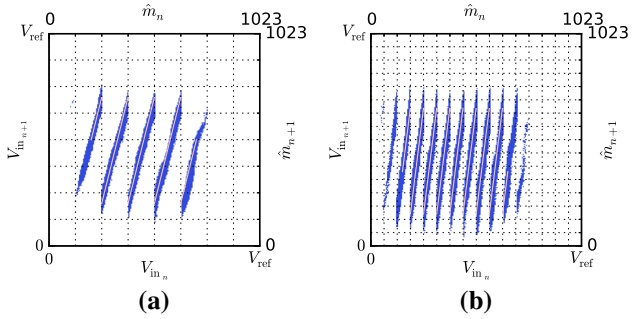



**Fig. 7** Experimental behavior of the prototype, with two different settings: in plot (a), $k = 4$, $N = 3$; in plot (b), $k = 8$, $N = 4$. The ADC resolution is $\hat{N} = 10$ bits. Measured behavior is shown in *blue* (*dark*), and expected behavior is superimposed in *orange* (*bright*) (Color figure online)

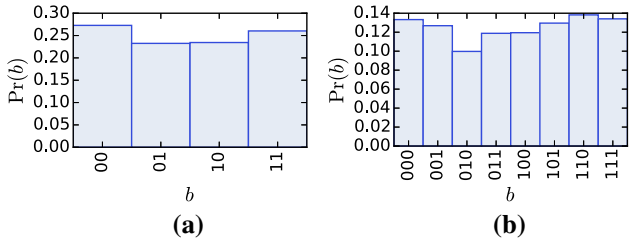



**Fig. 8** Probability distribution of the symbols generated at each cycle by the prototype system for the setups used in Fig. 7(a) in (a) and 7(b) in (b). Estimation based on $32 \times 10^6$ cycles

should be able to perform startup, total failure, and online tests [25]. Comprehensive statistical test suites would certainly work, but are often too computationally intensive for being built-in. Conversely, access to the system analog variables can be an enabler for much simpler functional tests.

## 5 Alternative quantization functions

Some chaos based TRNGs can rely on strong mathematical proofs about the correctness of their design [21, 34]. These let one formally state the ability to generate independent, equidistributed symbols in nominal conditions. Indeed, this is the case for the proposed source, as sketched in Sect. 3. Unfortunately, real world operating conditions are always different from nominal ones. As a consequence, one observes correlation and bias. The best indicator to summarize such defects is *entropy* or more precisely the entropy per output bit [27, 30]. For a truly random source, this is 1, while for any real-world source this is a bound that can only be approached.

Following Shannon, the entropy $H$ associated to a discrete random variable $\Phi$ with possible values $\{\phi_1, \ldots, \phi_l\}$ and probability mass function $P(\phi)$ is $H(\Phi) = E[-\log_2(P(\Phi))]$, where $E[\cdot]$ indicates expectation, so that $H(\Phi) = -\sum_{i=1}^{l} P(x_i)\log_2(P(x_i))$ [5]. In the current discussion, one is particularly interested in *conditional entropy*. Namely, for two events $\Phi$ and $\Psi$ taking values in $\{\phi_i\}$ and $\{\psi_j\}$, one is interested in $H(\Phi|\Psi) = \sum_{i,j} P(\phi_i, \psi_j)\log_2((P(\psi_j)/P(\phi_i, \phi_j))$, which indicates the amount of randomness in $\Phi$, when $\Psi$ is known. Let $\Phi$ be the last bit $B_n$ generated by the entropy source and $\Psi$ the sequence of the previous bits $B^{n-1} = (B_{n-1}, B_{n-2}, \ldots, B_0)$. With this, $H(B_n|B^{n-1})$ measures the amount of additional entropy (randomness) brought by the current bit when the previous $n$ bits are known. What counts is obviously the asymptotic value

$$H_B = \lim_{n\to\infty} H(B_n|B^{n-1}) \tag{13}$$

that indicates the average entropy carried by a bit, when infinitely many previous bits are known, or the average entropy per bit, for short. Such quantity can also be computed as $\lim_{n\to\infty}(1/n)H(B^n)$. Given that the proposed design generates multiple bits per cycle, one can alternatively introduce the quantity $H_W = \lim_{n\to\infty} H(W_n|W^{n-1})$ which is referred to DAS numbers $w$ instead of output bits $b$. Ideally, $H_W$ should be equal to $N_c$. Entropy per bit or per word are quite important qualifiers, being linked to the difficulty by which random quantities used in cryptography can be guessed (see [38] and references therein).

It has recently been observed that implementation errors degrade $H_B$ and $H_W$ because of two combined effects. The first one is related to the *metric entropy* $h_\mu(M)$ of the chaotic map $M(\cdot)$ [32]. Loosely speaking, such quantity indicates the highest possible entropy rate that can be obtained from the chaotic system using the best possible quantization function and represents a bound for $H_W$. Implementation errors affect the map shape and consequently are likely to reduce $h_\mu(M)$. The second effect has to do with the quantization function. Implementation errors reduce the adaptation of $q(\cdot)$ to $M(\cdot)$, degrading the ability to reach the $h_\mu(M)$ bound. Obviously, there is nothing that can be done about the first phenomenon. Yet, the second one (that is often more relevant) can partially be addressed by picking a different quantization function.

In [27], systems capable of generating 1 bit per cycle are considered, defining a sequence of quantization functions $\hat{q}_i(x) = \lfloor ix \rfloor \bmod 2$. As $i \to \infty$, $H_B$ tends to $\min(h_\mu(M), 1)$, regardless of the specific $M$. In other words, $\hat{q}_i(\cdot)$ asymptotically provides a *universal* quantization function giving the best $H_B$, whatever the map. The discussion can be generalized to systems generating $k$ bit per cycle by taking

$$\hat{q}_i(x) = \lfloor ix \rfloor \bmod k. \tag{14}$$

Unfortunately, the asymptotic $\hat{q}_\infty(\cdot)$ has little practical value, being impossible to implement. However, a similar

behavior can be approximated. To this aim, it is important to notice that the key properties of the functions $\hat{q}_i(\cdot)$ consist in: (i) having counter images corresponding to the possible outputs that always have measure $1/k$; (ii) being periodic with a spatial frequency that increases with $i$.

Interestingly, the proposed architecture is capable to compute

$$q_j(x) = (\hat{m} >> (\hat{N} - N - j)) \,\&\, (k-1) \tag{15}$$

with $j = 0, \ldots, \hat{N} - N$. For $j = 0$, $q_j(\cdot) \equiv q(\cdot)$. Furthermore, the family $q_j(\cdot)$ satisfies the two properties above. Specifically, $q_j(\cdot)$ has the same spatial frequency as $\hat{q}_{k\cdot 2^j}(\cdot)$. Thus, the function family $q_j(\cdot)$ shares the same properties as the family $\hat{q}_i(\cdot)$. If it could be extended to infinitely large $j$ values, it would provide the best possible quantization function in terms of robustness to variations in the chaotic map. Since $j$ is cannot exceed $\hat{N} - N$ in the proposed architecture, a further consideration is necessary. Convergence of $\hat{q}(\cdot)$ to the universal quantization function is not necessarily monotonic. In other words, some $\hat{q}_{i_1}(\cdot)$ can behave worse than $\hat{q}_{i_2}(\cdot)$ for finite $i_1 > i_2$. Therefore, there is in principle no guarantee that $q_j(\cdot)$ can get better and better as $j$ is increased, particularly at relatively low $j$ values. In practice, all the $q_j(\cdot)$ end up being *perfect* quantization functions for the nominal chaotic map in Eq. (9). Namely, in absence of implementation errors, they are all equivalent to $q(\cdot)$ in the ability to deliver equidistributed and independent symbols. This is a rather interesting property that shall be discussed in better depth elsewhere. What is relevant here is that, together with the rapid (in fact exponential) increase in the spatial frequency with $j$, such property can make one relatively confident in a regular convergence.

From the discussion carried on so far, one may get the feeling that the best $q_j(\cdot)$ is the one with the largest possible $j$. Jumping to such a conclusion would unfortunately be wrong. In fact, as $j$ is increased, implementation errors have a larger impact on $q_j(\cdot)$ itself. This is easily seen by considering that for $j = \hat{N} - N$, the function $q_j(x)$ returns the $N_c$ lsbs of the ADC output, that can be quite erratic. Consequently, one can expect $H_W$ to increase when adopting quantization functions $q_j(\cdot)$ with a larger $j$, but only up to a certain point. Then, one can presume $H_W$ to decrease as errors on the quantization function become dominant. As this happens, one can also foresee the system to become less tolerant to external interference, since the output gets more directly derived from the lsbs of the converter. The best quantization function is thus the result of a compromise, and one can expect the best $j$ values to be at least a couple of units below $\hat{N} - N$.

## 6 Experimental results

All the data presented in this section has been collected from the prototype system shown in Figs. 1 and 6, operating at approximately $15 \times 10^3$cycles/s for about $32 \times 10^6$ cycles.

### 6.1 Empirical entropy based tests

As mentioned in the previous sections, the best indicator for evaluating the quality of an entropy source is $H_B$. An experimental estimation $\tilde{H}_B$, has here been obtained by first computing $\tilde{H}_{B,n} = (1/n)\tilde{H}(B^n)$, where $\tilde{H}(B^n)$ is the empirical entropy corresponding to $H(B^n)$, for $n$ values $1, 2, \ldots, N$. Then, $\tilde{H}_B$ has been determined as $\min(\{\tilde{H}_{B,n}\})$. Because the effort required to compute $\tilde{H}(B^n)$ grows rapidly with $n$ and so do the requirements on the length of the bit sequence used for the estimation, $N$ has been limited to 12, which is enough for the present analysis. Table 1 shows $\tilde{H}_{B,n}$ both for the map with $N = 4$, $k = 8$ and for the map with $N = 3$, $k = 4$. $\tilde{H}_B$ is marked in bold. For each map the behavior is reported both for the quantization function $q_0(\cdot)$ and for the best $q_j(\cdot)$. From the tabled data, the following observations can be made:

(i) the two maps (corresponding to the two $N$, $k$ parameter sets under test) perform almost equivalently, even if the one with $k = 8$ provides a higher bit rate;
(ii) the achieved $\tilde{H}_B$ is always extremely high;
(iii) there is an evident advantage in using quantization functions designed as in Sect. 5;
(iv) when using $q_0(\cdot)$, lower values of $\hat{H}_{B,n}$ are encountered at $n$ values multiple of $N_c$.

The last point suggest that the main factor negatively affecting entropy is bias in symbols generated at each cycle rather than correlation among them. Indeed, bias is what the advanced quantization functions can improve. For what regards the choice of $j$ in $q_j(\cdot)$, Fig. 9, shows the dependence of $\tilde{H}_B$ on it, confirming the expectation expressed in the previous section that the best $j$ must fall a few units below $\hat{N} - N$.

### 6.2 NIST 800-22 tests and post-processing strategies

Tests from the US National Institute of Standards and Technology (NIST) 800-22 suite [37] have also been run to assess the suitability of the entropy source for a TRNG, basing them on 50 streams with $10^6$ bits each. The quality of the DAS numbers is so high that a surprisingly large

**Table 1** Average entropy in bit aggregates from the DAS numbers for the prototype in various configurations. Estimations based on $32 \times 10^6$ map cycles

| | $\tilde{H}_{B,n}$ | | | |
|---|---|---|---|---|
| | Map with $N = 3$, $k = 4$ | | Map with $N = 4$, $k = 8$ | |
| $n$ | with $q_0(\cdot)$ | with $q_5(\cdot)$ | with $q_0(\cdot)$ | with $q_4(\cdot)$ |
| 1 | 0.9999 | 0.9997 | 0.9999 | 0.9999 |
| 2 | 0.9983 | 0.9996 | 0.9996 | 0.9999 |
| 3 | 0.9987 | 0.9997 | 0.9979 | 0.9997 |
| 4 | 0.9911 | 0.9996 | 0.9995 | 0.9999 |
| 5 | 0.9964 | 0.9997 | 0.9992 | 0.9999 |
| 6 | 0.9859 | 0.9996 | 0.9948 | 0.9997 |
| 7 | 0.9944 | 0.9996 | 0.9984 | 0.9999 |
| 8 | 0.9821 | 0.9995 | 0.9980 | 0.9999 |
| 9 | 0.9927 | 0.9996 | 0.9916 | 0.9997 |
| 10 | 0.9796 | 0.9995 | 0.9973 | 0.9999 |
| 11 | 0.9913 | 0.9996 | 0.9970 | 0.9998 |
| 12 | **0.9777** | **0.9995** | **0.9884** | **0.9996** |

number of individual tests, about 76 % for the map with $N = 4$, $k = 8$ and quantization function $q_4(\cdot)$, is already passed *without any post-processing at all*.

This means that the suite can be passed as a whole with just an extremely lightweight, finite-memory processing. For instance, it is enough to collect bits at the output of the entropy source in small packets and to extract from each of them a slightly smaller output packet by a *discard-permute-and-XOR* mixing, as shown in Fig. 10. Here, two 16-bit words $D_A$ and $D_B$ are obtained from an 18-bit input packet $(d_0, \ldots, d_{17})$ accumulated from DAS numbers, by discarding some bits and scrambling the others. The output is then generated by the bitwise XOR of $D_A$ and $D_B$. The rate reduction ratio is 9 : 8, that is 1.125 bits from the DAS words are required for each output bit. The proposed post-processing is particularly well suited for the map with $k = 8$, since the latter generates $N_c = 3$ bits per cycle. Hence, an input packet is obtained in exactly 6 cycles during which the words $D_A$ and $D_B$ can be assembled. In the same 6 cycles, an output packet is generated that can be exactly delivered in a 2-byte word. This is just convenience, though, and the same scheme can also be applied while extracting $D_A$ and $D_B$ in other ways or based on other packet lengths. What is important is that: (i) is sufficiently long; (ii) the bit ordering in $D_A$ and $D_B$ is sufficiently different; and (iii) the number of bits discarded in building $D_A$ and $D_B$ from the input packet is sufficiently large. Condition (i) is indispensable to assure that the post-processing is sufficiently extended in time to be able to affect all the residual correlations in the DAS numbers. Since the chaotic model used in the source assures *exponentially vanishing* correlation profiles [34], relatively short packets

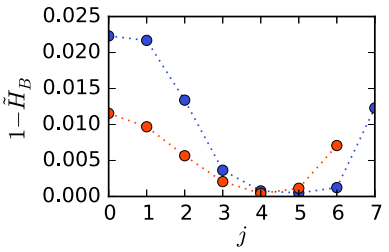



**Fig. 9** Dependency of $\tilde{H}_B$ on the particular quantization function $q_j(\cdot)$, for the map with $N = 3$, $k = 4$ (*blue, dark trace*) and for the map with $N = 4$, $k = 8$ (*orange, bright trace*) (Color figure online)

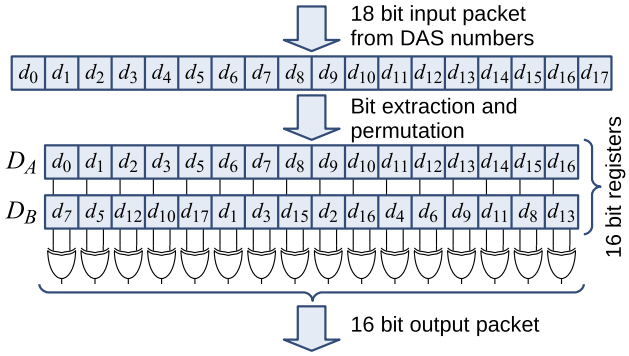



**Fig. 10** Sample *discard-permute-and-XOR* post-processing based on 18 bit input packets and 16 bits output packets

are sufficient. Still, the input packet needs to be at least as long as $N_c$ times the number of cycles required for the correlations to become negligible (5–6 cycles represent a safe value in the prototype implementation). Condition (ii) has to do with the quality of mixing used to destroy the correlations. Finally, condition (iii) refers to the rate reduction being practiced by the post-processor. From theory, the latter must be no less than $1/H_B$, otherwise it is not possible to get equidistributed, independent bits [30]. With a post-processor as simple as the proposed one, such bound must obviously be exceeded with some clearance.

During the validation of the prototype, it has also been verified that it succeeds in the NIST tests with other post-processors. For instance, the test suite is passed when the DAS numbers are treated by the (rather heavyweight) distiller in [39], or when the DAS numbers are processed through a PRNG, even if the latter is a modest (LFSR) as in Fig. 11 (as long as the register is sufficiently long, e.g., more than 12 bits for the proposed prototype).

Note that these two latter distillers are not theoretically correct since they practice no rate reduction and as such cannot enhance the average entropy per bit. Yet, they can trade long-term correlations for short-term ones at a sufficient level to satisfy the requirement of the 800-22 suite. Finally, NIST tests are obviously passed when the DAS numbers are used to fill the *entropy pool* of an operating system that provides its own entropy distiller, such as Linux [40].

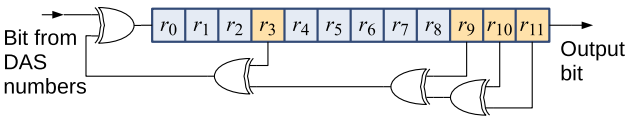



**Fig. 11** Alternative post-processing block using a 12-bit LFSR

To summarize, even if the NIST suite bears only a modest relevance when validating an entropy source (given that it is always possible to find a post-processing strategy that let all tests be passed), experimenting against the 800-22 suite is important in the present context to support two points. The first one is that the entropy source can operate with extremely lightweight distillers. In fact, the simpler the distiller (and the shorter the dependence of its current output on previous inputs), the lower its opacity and the better the TRNG testability, as it gets easier to see defects in the entropy source from the internal random numbers. In this sense the post-processor in Fig. 10 is much better than those in [39] or Fig. 11, because it assures that each bit in the internal numbers depends at most on 18 bits in DAS numbers. The second point is that the entropy source is suitable for direct coupling with entropy distillation solutions that can be found ready available on existing systems. This is important in view of retrofitting or design augmentation.

## 7 Conclusions

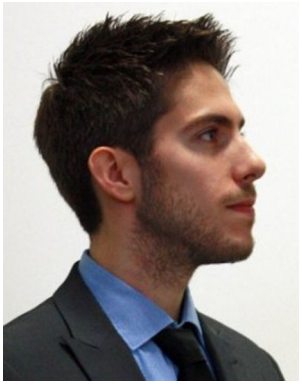
A novel architecture for a μC based entropy source exploiting chaotic dynamics has been proposed, prototyped and validated. The design has some unique features. First of all, it can be used either to implement standalone units suitable for an end user as a retrofit on existing systems or as an incremental schematic change that a manufacturer can apply during the revision of a product. In either case, costs can be low enough not to discourage adoption. Secondly, it follows recent testability guidelines for this kind of objects. In the third place, it is the first design of this sort flexible enough to let one test multiple quantization functions in the delivery of a digital output from the analog state of the internal chaotic systems. This is an important property both because it allows the best performance to be obtained and because it provides the opportunity to validate recent theoretical results in real world system. Finally, the proposed design has excellent (and probably unprecedented) good performance for this class of devices. Specifically: it can generate multiple bits at each iteration of the chaotic system; it delivers more than 0.999 bits of entropy per output bit; it can be transformed into a TRNG passing the NIST 800-22 test suite with the cascading of extremely light-weight post processing units, with minimal data rate loss.

## References


1. Middleton, P., Kjeldsen, P., & Tully, J. (2013). Forecast: The internet of things, worldwide, 2013. Report G00259115, Gartner.
2. Roman, R., Najera, P., & Lopez, J. (2011). Securing the internet of things. *IEEE Computer*, *44*(9), 51–58.
3. Lynn, S. (2011). Survey: Biz network devices vulnerable, almost obsolete. PC magazine (online) http://www.pcmag.com/article2/0,2817,2385833,00.asp.
4. Eastlake, DE., Shiller, JI., & Crocker, SD. (2005). Randomness requirements for security. RFC 4086, http://www.ietf.org/rfc/rfc4086.txt.
5. Gray, RM. (2011). Entropy and information theory. New York: Springer. http://www-ee.stanford.edu/∼gray/it.html.
6. Wikipedia. (2014). Random number generator attack. http://en.wikipedia.org/wiki/Random_number_generator_attack. Accessed 13 May 2014.
7. Taylor, G., & Cox, G. (2011). Digital randomness. *IEEE Spectrum*, *48*(9), 32–58.
8. Dömstedt, B. (2013). TRNG9880 random number processing. White Paper, http://www.trng98.se/getfile.php?file=trng9880_info.pdf.
9. Simtec Electronics. (2009). Entropy key. White Paper, http://www.entropykey.co.uk/res/download/diagram-explanation.pdf.
10. Campbell, P., & Cheetham, J. (2009). OneRNG—theory of operation. White Paper, http://moonbaseotago.com/onerng/theory.html.
11. Cox, B. (2014). Infinite noise TRNG. Manual, https://github.com/waywardgeek/infnoise.
12. Pareschi, F., Scotti, G., Giancane, L., Rovatti, R., Setti, G., & Trifiletti, A. (2009). Power analysis of a chaos-based random number generator for cryptographic security. In *Proceedinds of the IEEE International Symposium on Circuits and Systems, 2009*, (pp. 2858–2861).
13. Kennedy, M. P., Rovatti, R., & Setti, G. (Eds.). (2000). *Chaotic electronics in telecommunications*. Boca Raton, USA: CRC International Press.
14. Callegari, S., Rovatti, R., & Setti, G. (2003). Chaos based FM signals: Applications and implementation issues. *IEEE Transactions on Circuits and Systems I*, *50*(8), 1141–1147. doi:10.1109/TCSI.2003.815222.
15. Callegari, S., Rovatti, R., & Setti, G. (2002). Chaotic modulations can outperform random ones in EMI reduction tasks. *Electronics Letters*, *38*(12), 543–544. doi:10.1049/el:20020381.
16. Callegari, S., Pareschi, F., Setti, G., & Soma, M. (2010). Complex oscillation based test and its application to analog filters. *IEEE Transactions on Circuits and Systems I*, *57*(5), 956–969. doi:10.1109/TCSI.2010.2046956.
17. Callegari, S. (2008). Introducing complex oscillation based test: An application example targeting analog to digital converters. In *Proceedings of ISCAS*, Seattle. (pp. 320–323). doi:10.1109/ISCAS.2008.4541419.
18. Delgado-Restituto, M., Medeiro, F., & Rodríguez-Vázquez, A. (1993). Nonlinear, switched current CMOS IC for random signal generation. *Electronics Letters*, *25*, 2190–2191.
19. Callegari, S., Setti, G., & Langlois, P.J. (1997). A CMOS tailed tent map for the generation of uniformly distributed chaotic sequences. In *Proceedings of ISCAS'97*, Hong Kong (Vol. 2, pp. 781–784). doi:10.1109/ISCAS.1997.621829.
20. Callegari, S., Rovatti, R., & Setti, G. (2005a). First direct implementation of true random source on programmable hardware. *International Journal of Circuit Theory and Applications*, *33*(1), 1–16. doi:10.1002/cta.301.
21. Callegari, S., Rovatti, R., & Setti, G. (2005b). Embeddable ADC-based true random number generator for cryptographic

applications exploiting nonlinear signal processing and chaos. *IEEE Transactions on Signal Processing*, *53*(2), 793–805. doi:10.1109/TSP.2004.839924.
22. Fabbri, M., & Callegari, S. (2014). Very low cost entropy source based on chaotic dynamics retrofittable on networked devices to prevent RNG attacks. In *Proc. 21 st IEEE International Conference on Electronic Circuits and Systems (ICECS)*, Marseille (pp. 175–178). doi:10.1109/ICECS.2014.7049950.
23. Callegari, S., & Setti, G. (2007). ADCs, chaos and TRNGs: A generalized view exploiting Markov chain lumpability properties. In *Proceedings of ISCAS*, New Orleans, (pp. 213–216). doi:10.1109/ISCAS.2007.378314.
24. Schindler, W., & Killmann, W. (2003). Evaluation criteria for true random number generators used in cryptographic applications. In *Cryptographic Hardware and Embedded Systems—CHES 2002*, Springer (pp. 431–449).
25. Killmann, W., & Schindler, W. (2011). A proposal for: Functionality classes for random number generators. Standard AIS-31, German Federal Office for Information Security.
26. Fischer, V. (2012). A closer look at security in random number generators design. *Third International Workshop on Constructive Side-Channel Analysis and Secure Design—COSADE 2012* (pp. 167–182). Lecture Notes in Computer Science: Springer.
27. Beirami, A., Nejati, H., & Callegari, S. (2014). Fundamental performance limits of chaotic-map random number generators. In *Proc. of the 52nd Annual Allerton Conference on Communication, Control, and Computing* (pp. 1126–1131). doi:10.1109/ALLERTON.2014.7028581.
28. ISO 18031. (2011). Information technology—security techniques— andom bit generation. Standard.
29. Vembu, S., & Verdù, S. (1995). Generating random bits from an arbitrary source: Fundamental limits. *IEEE Transactions on Information Theory*, *41*(5), 1322–1332.
30. Beirami, A., & Nejati, H. (2013). A framework for investigating the performance of chaotic-map truly random number generators. *IEEE Transactions on Circuits and Systems II*, *60*(7), 446–450. doi:10.1109/TCSII.2013.2258274.
31. Jun, B., & Kocher, P. (1999). The Intel random number generator. White Paper, Cryptography Research Inc., http://www.cryptography.com/resources/whitepapers.
32. Ott, E. (1993). *Chaos in dynamical systems*. Cambridge: Cambridge University Press.
33. Lasota, A., & Mackey, M. C. (1995). *Fractals and Noise. Stochastic Aspects of Dynamics* (2nd ed.). New York: Springer.
34. Setti, G., Mazzini, G., Rovatti, R., & Callegari, S. (2002). Statistical modeling of discrete time chaotic processes: Basic finite dimensional tools and applications. *Proceedings of the IEEE*, *90*(5), 662–690.
35. Baker, BC. (2004). Techniques that reduce system noise in ADC circuits. Application Note ADN007, Microchip Technologies Inc.
36. Marsaglia, G. (1995). The Marsaglia random number CDROM including the Diehard battery of tests of randomness. CDROM, http://www.stat.fsu.edu/pub/diehard.
37. Rukhin, A., Soto, J., Nechvatal, J., Smid, M., Barker, E., Leigh, S., Levenson, M., Vangel, M., Banks, D., Heckert, A., Dray, J., & Vo, S. (2001). A statistical test suite for random and pseudorandom number generators for cryptographic applications. Special publication SP 800-22, National Institute for Standards and Technology, http://csrc.nist.gov/rnd/SP800-22b.pdf.
38. Beirami, A., Calderbank, R., Duffy, K., & Médard, M. (2015). Quantifying computational security subject to source constraints, guesswork and inscrutability. In *Proc. of the 2015 IEEE International Symposium on Information Theory (ISIT 2015)*, Hong Kong, accepted for publication.
39. Poli, S., Callegari, S., Rovatti, R., & Setti, G. (2004). Post-processing of data generated by a chaotic pipelined ADC for the robust generation of perfectly random bitstreams. In *Proc. of ISCAS'04*, Toronto, CA (Vol. 4, pp. 585–588), doi:10.1109/ISCAS.2004.1329071.
40. Gutterman, Z., Pinkas, B., & Reinman, T. (2006). Analysis of the linux random number generator. In *Proc. of the IEEE Symposium on Security and Privacy*, Oakland, CA, USA.

**Sergio Callegari** received the Dr. Eng. degree (with honors) in electronic engineering and a Ph.D. degree in electronic engineering and computer science from the University of Bologna, Italy, in 1996 and 2000, respectively, working nonlinear circuits and chaotic systems. In 1996, he was a visiting student at King's College London, U.K. He is currently a researcher and Assistant Professor at the Department of Electrical, Electronic, and Information Engineering, University of Bologna, where he teaches Analog Electronics and Signal processing, and Applied Electronics to students of Electronic Engineering and Aerospace Engineering. He is also a faculty member of the Advanced Research Center on Electronic Systems (ARCES) at the University of Bologna. In 2008, 2009, 2011 he was a visiting researcher at the University of Washington in Seattle for short periods. His current research interests include nonlinear signal processing, internally nonlinear, externally linear networks, chaotic maps, delta-sigma modulation, testing of analog circuits, and random number generation. Dr. Callegari has authored or co-authored more than 80 papers in international conferences, journals and scientific books, as well as four national patents and has co-edited a scientific book. In 2004 he was co-recipient of the IEEE Circuit and Systems Society Darlington Award, for the best paper appeared in the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS in the previous biennium. He served as an Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II during 2006–2007 and as an Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS I during 2008–2009. He is currently in the editorial board of the IEICE Nonlinear Theory and Its Applications Journal. He is Chair of the Technical Committee on Nonlinear Circuits and Systems and member of the Technical Committee on Education and Outreach of the IEEE CAS Society. He also served in the Organization Committee and as Publication Co-Chair at NOLTA 2006, as a member of the Organization Committee of Eurodoc 2006, as a Special-Session Co-Chair of NOLTA 2010 and as a Chair for the nonlinear circuits track at ICECS 2012 and ISCAS 2013. In 2005–2007, he has been a member of the board of the Italian Society of Doctoral Candidates and Ph.D. Graduates (ADI).



**Mattia Fabbri** received his Master Degree in Electronic and Communications Engineering at the University of Bologna in 2013. Since then, he has worked as a hardware and firmware designer, focusing on projects related to smart cities, wireless sensor network, embedded systems, process and environmental monitoring.

**Ahmad Beirami** received his B.Sc. in Electrical Engineering from Sharif University of Technology in 2007 and his M.Sc. and Ph.D. in Electrical and Computer Engineering from Georgia Institute of Technology in 2011 and 2014, respectively. He is currently a postdoctoral scholar jointly affiliated with the information initiative at Duke (iiD) and the Research Laboratory of Electronics (RLE) at MIT. Beirami's research interests broadly include information theory, cyber security, machine learning, statistics, and networks. Beirami is the coauthor of a paper that received best student paper nomination in IEEE Midwest Symposium on Circuits and Systems (2008). His Ph.D. work received the Center for Signal and Information Processing Outstanding Research Award (2014), the 2013–2014 School of ECE Graduate Research Excellence Award, and the 2015 Sigma Xi Best Ph.D. Thesis Award all from Georgia Institute of Technology.