UNIVERSITÀ
DI SIENA
1240

Department of Information Engineering and Mathematics

PhD Program in
INFORMATION ENGINEERING AND SCIENCE

RESEARCH LINE: Electronics, Electrical Engineering and Electronic Measurement

# Digital Nonlinear Oscillators: A Novel Class of Circuits for the Design of Entropy Sources in Programmable Logic Devices

**Advisor:**
**Prof. Tommaso Addabbo**

**Candidate:**
**Riccardo Moretti**

**XXXIII Cycle**

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

In recent years, cybersecurity is gaining more and more importance. Cryptography is used in numerous applications, such as authentication and encryption of data in communications, access control to restricted or protected areas, electronic payments [1–3]. It is safe to assume that the presence of cryptographic systems in future technologies will become increasingly pervasive, leading to a greater demand for energy efficiency, hardware reliability, integration, portability and security.

However, this pervasiveness introduces new challenges, such as the implementation of cryptographic primitives with improved performance in terms of timing, chip area, power and computational resource consumption, addressing the increasing demand of low-complexity hardware devices, like systems for the Internet of Things (IoT). In response to this limitation, lightweight cryptography comes into play - a branch of cryptography that provides tailor-made solutions for resource-limited devices.

One of the fundamental classes of cryptographic hardware primitives is represented by Random Number Generators (RNGs), that is, systems that provide sequences of binary symbols that are deemed unpredictable [4].

The circuits and systems that implement RNGs can be divided into two categories, namely Pseudo Random Number Generators (PRNGs) and True Random Number Generators (TRNGs).

PRNGs are deterministic and eventually periodic finite state machines, capable of generating sequences that appear to be random. In other words, a PRNG is a device that generates and repeats a finite random sequence, saved in memory or generated by calculation.

A TRNG, on the other hand, is a device that generates random numbers based on stochastic physical processes. Typically, a hardware TRNG consists of a mixed-signal circuit that is classified according to the stochastic process on which it is based. Specifically, the most commonly used sources of randomness are [5]:

- chaotic circuits;
- high jitter oscillators;
- circuits that measure other stochastic processes.

A chaotic circuit is an analog or mixed-signal circuit in which currents and voltages vary over time according to systems of nonlinear differential equations [6]. The time evolution of these currents and voltages can be understood as the evolution of the state of a chaotic nonlinear dynamical system.

Jitter can instead be defined as the deviation of the output signal of an oscillator from its true periodicity, due to electronic noise, which causes uncertainty in its transition times [7].

Other possible stochastic processes that a TRNG can use may involve radioactive decay, photon detection, or electronic noise in semiconductor devices [8–15].

TRNGs presented in the literature are typically designed in the form of Application Specific Integrated Circuits (ASICs). On the other hand, in recent years an increasing number of researchers are investigating the design of TRNGs in Programmable Logic Devices (PLDs) [16–25]. A PLD offers, compared to an ASIC, clear advantages in terms of cost and versatility. At the same time, however, there is currently a widespread lack of trust in these PLD-based architectures, starting from specific cryptographic weaknesses found in well known solutions based on Ring Oscillators [26–28].

In this work we propose a novel class of circuits suitable for being implemented in digital devices, as PLDs, as a valid alternative to traditional solutions proposed in literature to generate random bits.

## 1.1 Thesis Organization

In the next chapters a new class of nonlinear circuits based on digital hardware is introduced that can be used as entropy sources for TRNGs implemented in PLDs, named Digital Nonlinear Oscillators (DNOs).

More in detail:

- Chapter 2 provides the definition of Digital Nonlinear Oscillator, supported by notable examples capable of demonstrating experimentally how different circuit topologies referable to this class can express significantly different performance;
- Chapter 3 introduces the analysis methods needed to evaluate the performance of a Digital Nonlinear Oscillator, thus establishing an approach for the design od DNO-based TRNGs;
- Chapter 4 proposes a circuit topology usable as a high performance entropy source;
- Chapter 5 describes an algorithmic procedure, suitable for being implemented in low-complexity PLDs, aiming to select, within a given set of random binary sources, the one with highest entropy.

# Chapter 2

# Digital Nonlinear Oscillators

In this chapter we introduce a novel class of circuits that can be used to design entropy sources for True Random Number Generation, called Digital Nonlinear Oscillators (DNOs). DNOs constitute nonlinear dynamical systems capable of supporting complex dynamics in the time-continuous domain, although they are based on purely digital hardware. By virtue of this characteristic, these circuits are suitable for their implementation on Programmable Logic Devices. Focusing on the analysis of Digital Nonlinear Oscillators implemented in FPGAs, a preliminary comparison is proposed between three different circuit topologies belonging to the introduced class, in order to demonstrate how circuits of this type can have different characteristics, depending on their dynamical behavior and hardware implementation.

## 2.1 Definition

We open this chapter with an informal definition [29].

**Definition 2.1.** *A Digital Nonlinear Oscillator (DNO) is a network of electronic circuits, originally designed to behave as digital logic gates, which implements an autonomous nonlinear dynamical system that exhibits complex (periodic or chaotic) dynamics in the time-continuous domain.*

Therefore, a DNO is a circuit capable of generating entropy on the basis of two possible dynamical behaviors. In case of chaotic dynamics, the generated entropy mainly depends on the dynamical characteristic of the implemented circuit topology. Alternatively, the circuit acts as a periodic oscillator, and the information generation mechanism depends on the electronic noise (e.g. causing phase noise or jitter).

What links both the described cases is the nonlinearity of DNOs, which depends on the intrinsically nonlinear nature of the electronic circuits necessary for the design of digital logic gates present in the system. In fact, these circuits typically use transistors as switches, with the aim, on a 'large signal' scale, of bringing the output voltages to saturation towards ground or power supply voltages, representing binary logic levels. As it is clarified in this thesis, DNOs can be understood as analog systems implemented in digital devices.

## 2.2 DNO Design in Programmable Logic Devices

As anticipated by Definition 2.1, a DNO is a dynamical system that can be implemented using only digital hardware. By virtue of this characteristic, focusing the design of this class of circuits in Programmable Logic Devices (PLDs) is of particular interest.

In digital electronics, a PLD is an integrated circuit which at the time of manufacture is not configured to perform any specific boolean operation and therefore, before being able to use it, must be programmed (i.e. configured). This implies that, using PLDs, different logic circuits, although functionally different from each other, are implemented by programming the same hardware.

Without losing generality, we can refer to FPGAs, that are a special class of PLDs [30].

Giving a simplified description, an FPGA can be seen as a 2D matrix of cells, called Configurable Logic Blocks (CLBs). A CLB is the fundamental logical resource for the implementation of sequential or combinatorial circuits. Each CLB connects with the others via a local switch matrix, which then allows to access to the general routing matrix.

Each CLB includes a defined number of slices, within which there are programmable hardware elements for the implementation of the different logic functions. The set of elements in a slice is referred to as Elementary Logic Block (ELB).

An ELB basically includes three elements:

- a Look-Up Table (LUT), that can be used to store truth tables of arbitrary 1-bit logic functions;
- a flip-flop, through which it is possible to memorize the logic output state of the function, for the implementation of synchronous gates;
- a multiplexer for the synchronous or asynchronous configuration of the logic port.

Using this generic structure, it is possible to program any logic function characterized by one output bit and a given number of input bits up to the maximum supported by the ELB hardware, as shown in Fig. 2.1.

Based on this observation and recalling Definition 2.1, it is possible to state that, in the context of the design of a DNO on PLDs, each ELB represents a node of the topology. In this sense, a DNO can be represented as an oriented graph whose nodes are the ELBs and whose arcs are the connections between the output and the inputs of each ELB, as shown in Fig. 2.2.

On the other hand, each autonomous network of ELBs capable of supporting periodic or chaotic dynamics constitutes a DNO. It is clear that a statement of this

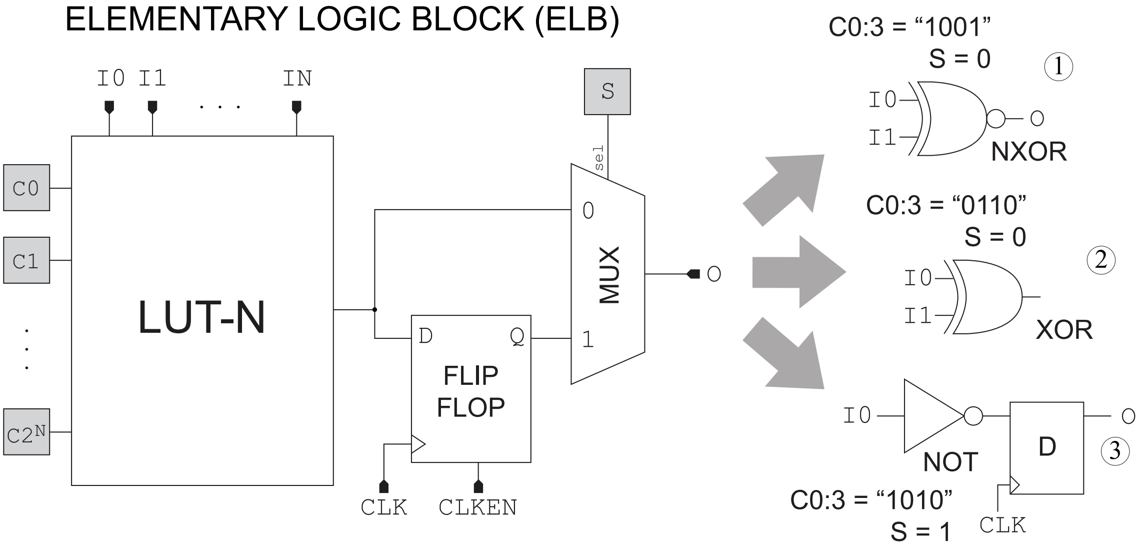


Figure 2.1: Simplified structure of an FPGA configurable Elementary Logic Block (ELB), representing the set of elements required to implement a logic port.

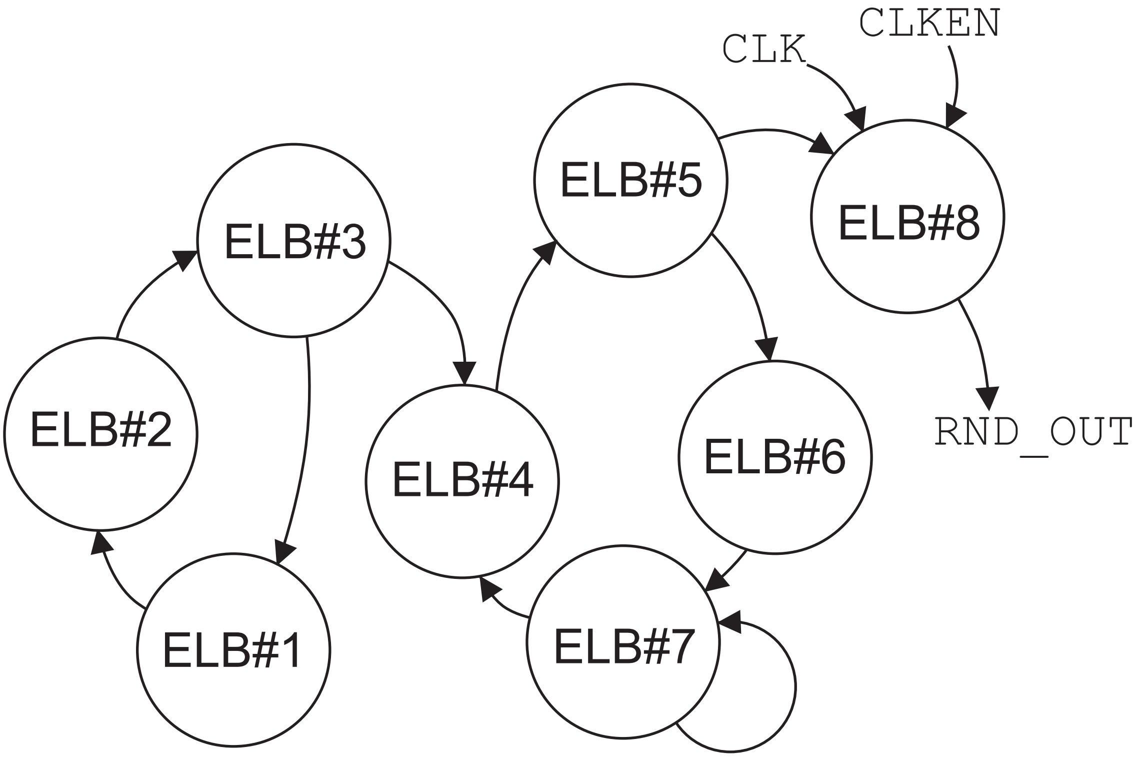


Figure 2.2: Oriented graph representation of a possible network topology for a 7-nodes DNO designed in FPGAs. The graph nodes are the FPGA ELBs, while the arcs are the connections between the output and the inputs of each ELB.

type defines a broad family of circuits, as it does not limit the complexity or size of the considered networks.

In addition, it should also be noted that the performance of a DNO topology is strictly dependent on the characteristics of the specific hardware implementation. This means that, to evaluate a DNO, it is not sufficient to define the network topology and the logical function performed by each node, but it is necessary to analyze also the effects introduced by the use of a specific technology for the implementation of the circuit, including logic gates and routing elements.

## 2.3 Example Cases: from Ring Oscillator to Custom DNOs

In this section, three notable examples of DNOs characterized by the same number of nodes are provided. These examples, that constitute networks of different complexity, are used to show how different DNOs can be characterized by particularly different performance from a dynamical behavior point of view, and therefore from their entropy.

The DNOs taken into consideration, shown in Fig. 2.3, are:

- a Ring Oscillator;
- a Galois Ring Oscillator;
- a custom DNO topology.

The three topologies are all composed of seven ELBs, plus an eighth node (ELB#8) used for the uniform sampling of the output signal, in order to generate a random sequence of bits.

### 2.3.1 Ring Oscillator

The Ring Oscillator is a circuit composed by an odd number of NOT gates closed in a loop. Its output oscillates between two voltage levels with a frequency $f_{\mathrm{RO}}$ which depends on the propagation time $\tau_p$ of a NOT gate and on the number $N$ of inverters in the chain:

$$f_{\mathrm{RO}} \approx \frac{1}{2\tau_p N}. \tag{2.1}$$

The random component linked to this structure resides in the jitter to which the oscillator is subjected.

The Ring Oscillator is a well known topology in the context of entropy sources based on digital oscillators [31–35], with controllable and repeatable dynamical characteristics; for this reason, in this section it plays the role of reference benchmark for the other two topologies.

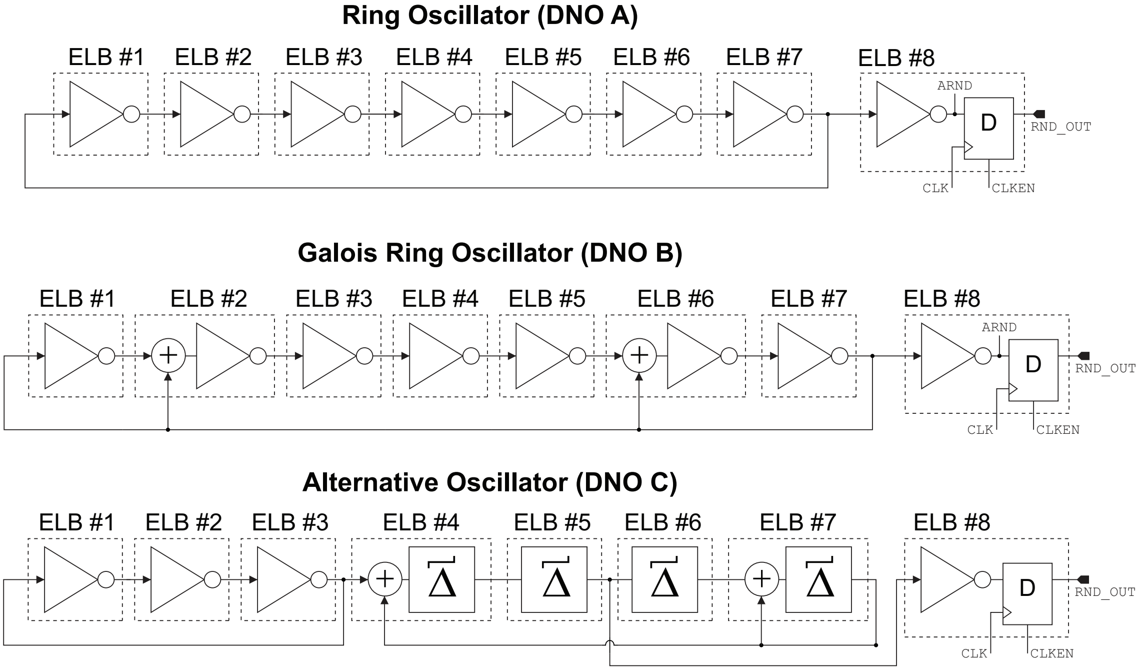


Figure 2.3: Three different DNO topologies. The three topologies have the same number of nodes, but they constitute networks of different complexity thanks to the routing between nodes.

### 2.3.2 Galois Ring Oscillator

The Galois Ring Oscillator was proposed by Golić in 2006 [36].

Inspired by Ring Oscillators and Linear Feedback Shift Registers (LFSRs), Golić discussed two topologies based on loops of inverters combined with XOR gates. The proposed structures have the appearance of LFSRs but, instead of registers, have inverters used as delay elements. Golić proposed two topologies, called Fibonacci Ring Oscillator and Galois Ring Oscillator. The difference between the two lies in the fact that in the Fibonacci topology a single feedback network controls the first node of the loop, while in the Galois topology the feedback signals are distributed over multiple nodes, similarly to the Fibonacci or Galois topologies of LFSRs.

Golić investigated these structures as synchronous finite state machines, identifying theoretical conditions such to have no fixed points or to maximize the period of oscillation. From the perspective of DNOs, what is missing in the Golić approach is an assessment of the dynamical behavior of the physical circuit.

To investigate the weight of this aspect, it can be useful to analyze the dynamical behavior of the signals involved in the Galois Ring Oscillator shown in Fig. 2.4 by means of numerical transient simulations.

Taking as a reference the UMC 180 nm technology, we designed the LUT structure shown in Fig. 2.1 at the CMOS transistor level in Cadence Virtuoso, as shown in Fig. 2.5. Using the LUTs we built the circuit corresponding to the Galois Ring Oscillator topology and we carried out simulation campaigns, subjecting the circuit

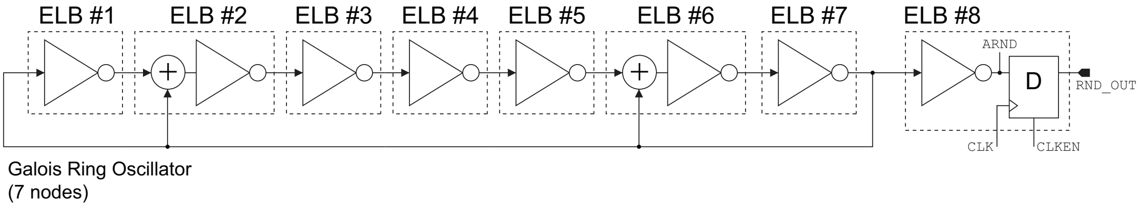


Figure 2.4: A possible 7-nodes Golić system. The topology defines a Galois Ring Oscillator. ELB#7 output serves as feedback signal, distributed over the ELBs#[2,6]. ELB#7 output is also the output signal, uniformly sampled by ELB#8.

to additive white noise.

What emerges from the simulations, summarized in Fig. 2.6, is that the oscillator can exhibit limit cycles with a relatively short duration, in disagreement with what Golić theorized. Furthermore, the circuit also appears to be quite robust to perturbations, as its periodic behavior remains recognizable even in presence of unrealistic high noise levels.

### 2.3.3 Custom DNO Topology

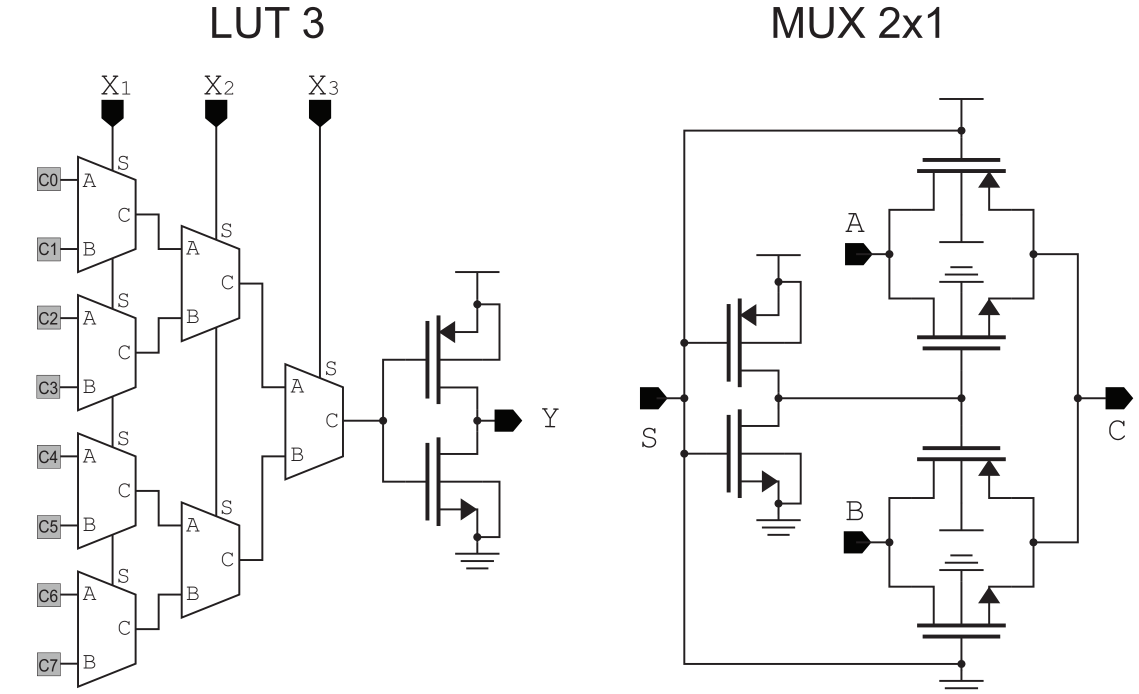
The third system analyzed in this section consists of an original DNO topology, obtained by combining a Ring Oscillator with loop structures composed of digital delays and XOR gates, as shown in Fig. 2.7. The digital delays are marked in the figure by a special symbol, which has the purpose of highlighting how, from an analogical point of view, they constitute signal rectifiers.

The considered topology is able to exhibit complex dynamics, as can be observed through simulations in Cadence Virtuoso based on the use of the LUTs built using the CMOS UMC 180 nm technology (Fig. 2.5).

For example, consider the transient simulations shown in Fig. 2.8, obtained by forcing the initial conditions of the ELBs#[4-7] to voltages (0,0,0,0) V.

Considering that the ELBs#[1-3] constitute a Ring Oscillator, we can exclude the presence of stable fixed points for the entire structure. Focusing on the output dynamics of ELBs#[4-7], we observe that the first low-high transition of ELB#3 propagates in subsequent ELBs until it triggers the self-oscillation of the loop composed by ELB#7 (evidence mark A). This oscillation is then transferred to ELB#4 and mixed with the signal from the Ring Oscillator. All this can lead to the creation of complex periodic dynamics, depending on the ratio between the time constants of the two subsystems.

Bringing our attention to the evidence mark B, it is possible to notice the nonlinear behavior of the digital delays, as the high gain in each stage tends to saturate the input signals towards ground or power supply voltages.



$$Y = \bar{C}_0\bar{X}_1\bar{X}_2\bar{X}_3 + \bar{C}_1\bar{X}_1\bar{X}_2X_3 + \bar{C}_2\bar{X}_1X_2\bar{X}_3 + \bar{C}_3\bar{X}_1X_2X_3 + \bar{C}_4X_1\bar{X}_2\bar{X}_3 + \bar{C}_5X_1\bar{X}_2X_3 + \bar{C}_6X_1X_2\bar{X}_3 + \bar{C}_7X_1X_2X_3$$

$$C = SB + \bar{S}A$$

Figure 2.5: Schematic representation of a 3-inputs Look-Up Table (LUT) and of the 2-inputs multiplexers composing the LUT, designed in Cadence Virtuoso at transistor level using the UMC 180 nm technology.

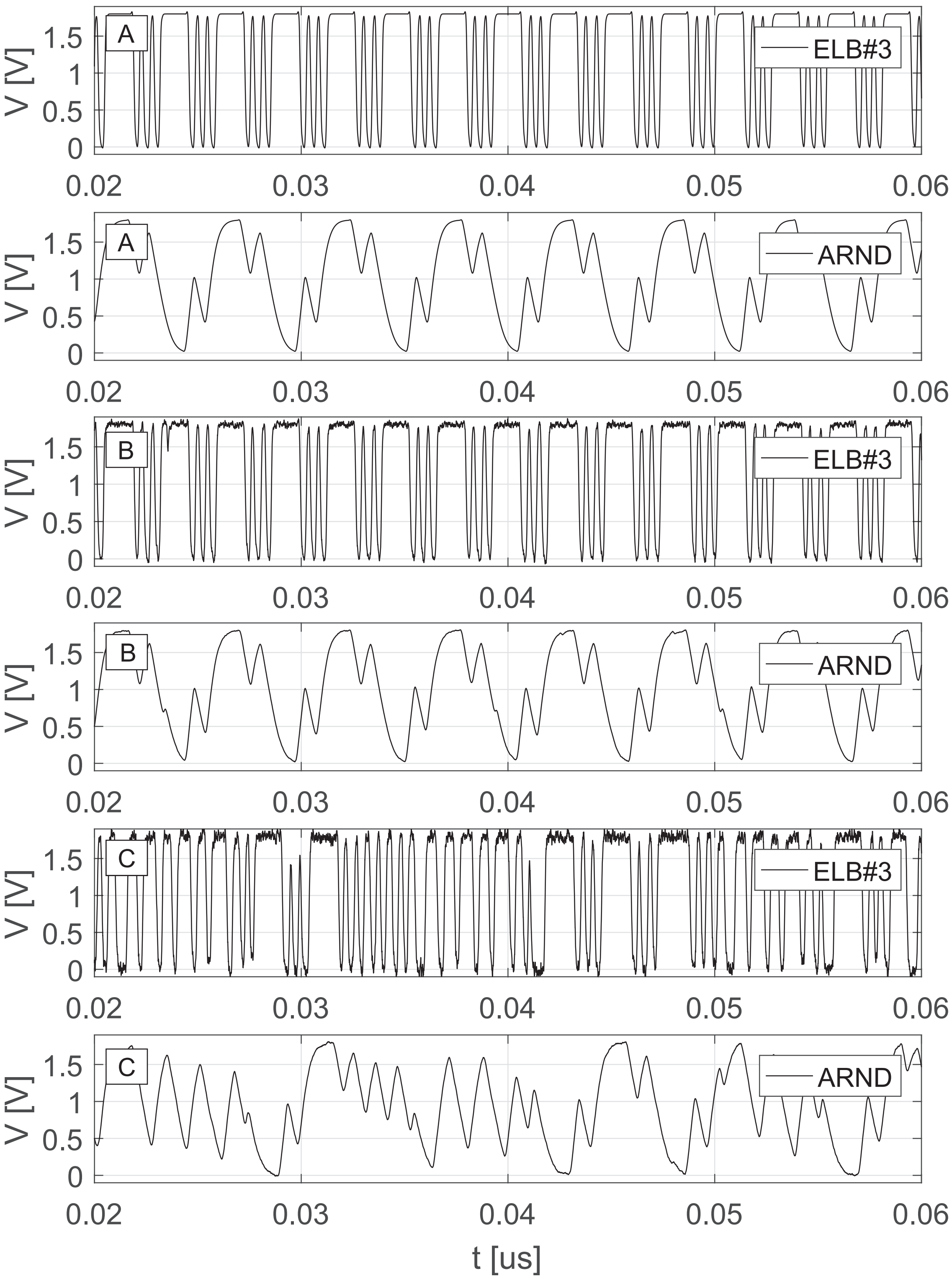


Figure 2.6: Cadence Virtuoso transient simulations of the Galois Ring Oscillator shown in Fig. 2.4, designed using the LUT reference structure shown in Fig. 2.5 (UMC 180nm CMOS technology). Case A: no additive noise; case B: relevant additive noise; case C: abnormal additive noise.

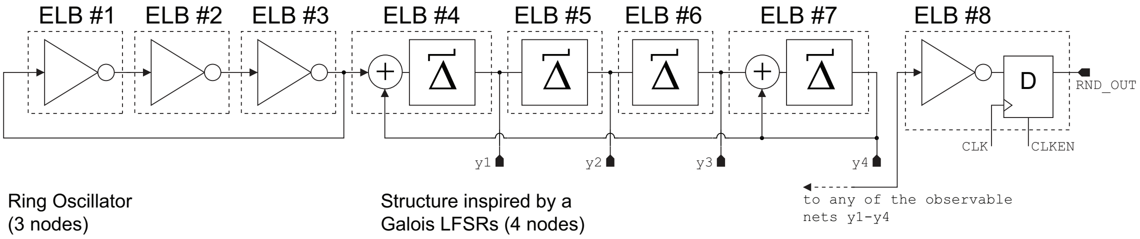

ELB #1
ELB #2
ELB #3
ELB #4
ELB #5
ELB #6
ELB #7
ELB #8
D
RND_OUT
CLK
CLKEN
y1
y2
y3
y4
to any of the observable nets y1-y4
Ring Oscillator (3 nodes)
Structure inspired by a Galois LFSRs (4 nodes)

Figure 2.7: 7-nodes system belonging to the new class of proposed DNOs. The topology combines a 3-nodes Ring Oscillator (ELBs#[1-3]) with loop structure composed of digital delays (ELB#[5,6]) and XOR gates (ELBs#[4,7]). ELB#8 uniformly samples the output signal, selectable among the ELBs#[4-7].

### 2.3.4 Experimental Analysis

We implemented the three topologies on a Xilinx Artix 7 xc7a35 FPGA to analyze the performance of the three DNOs from the point of view of generating random numbers. The same hardware resources were used for each of them, to ensure a fair comparison between the three implementations. To do this, manual control of FPGA resources place and route phases was applied. More specific details regarding the procedure by which DNOs are implemented on FPGAs are provided in Section 3.5.

The output signal of each implementation was sampled at different frequencies, defined on a range between 100 kHz and 100 MHz. For each sampling frequency, one million bits long sequences were acquired, on which analyzes were then performed aimed at evaluate the level of randomness.

Since the goal of this analysis was to compare the performance of three different topologies, rather than evaluating the outcome of standard statistical tests, such as NIST 800.22 [4], we adopted the following conventional metrics:

- pattern distribution of subsequent generated bytes;
- average Shannon redundancy;
- autocorrelation function;
- runs distribution;
- probability distribution of generated bytes.

In this way it is possible to compare imperfect sources, avoiding the typical “saturated to fail” results of standard high sensitivity cryptographic statistical tests.

Fig. 2.9 shows the pattern distributions of successive generated bytes for the three topologies, evaluated at different frequencies. From the image, it can be seen that the Ring Oscillator loses the uniform pattern for frequencies higher than 100 kHz, the Galois Ring Oscillator for frequencies higher than 500 kHz, the custom system maintains uniformity up to 5 MHz.

Fig. 2.10 shows the average Shannon redundancy (defined as the complement the average Shannon entropy) for binary words up to 16 bits for the three systems,

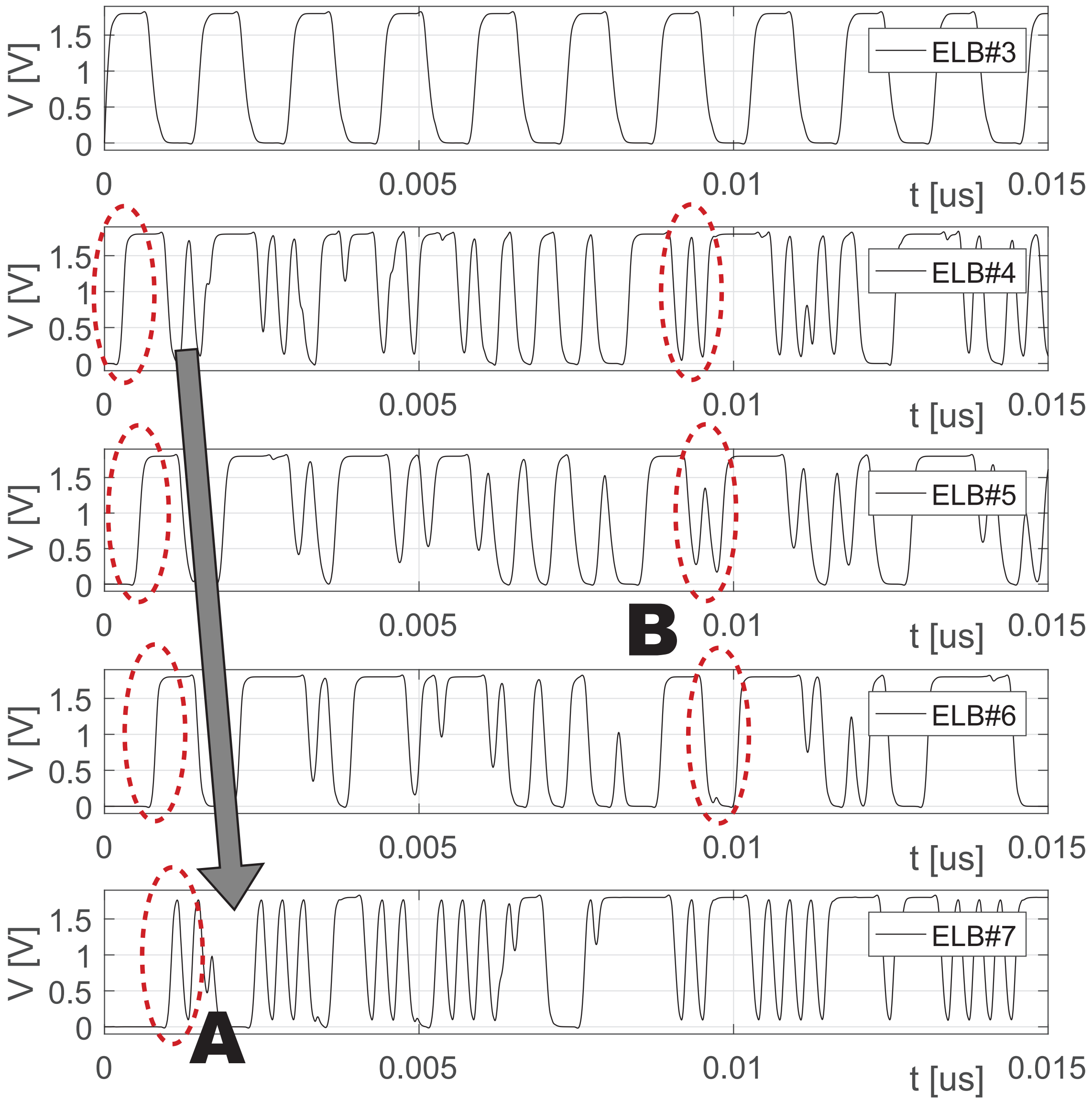


Figure 2.8: Cadence Virtuoso transient simulations of the custom DNO topology shown in Fig. 2.7, designed using the LUT reference structure shown in Fig. 2.1 (UMC 180 nm CMOS technology). The simulations are performed without additive noise. Mark A highlights the propagation in the circuit of the first low-high transition of ELB#3. Mark B highlights the nonlinear behavior of the digital delays.

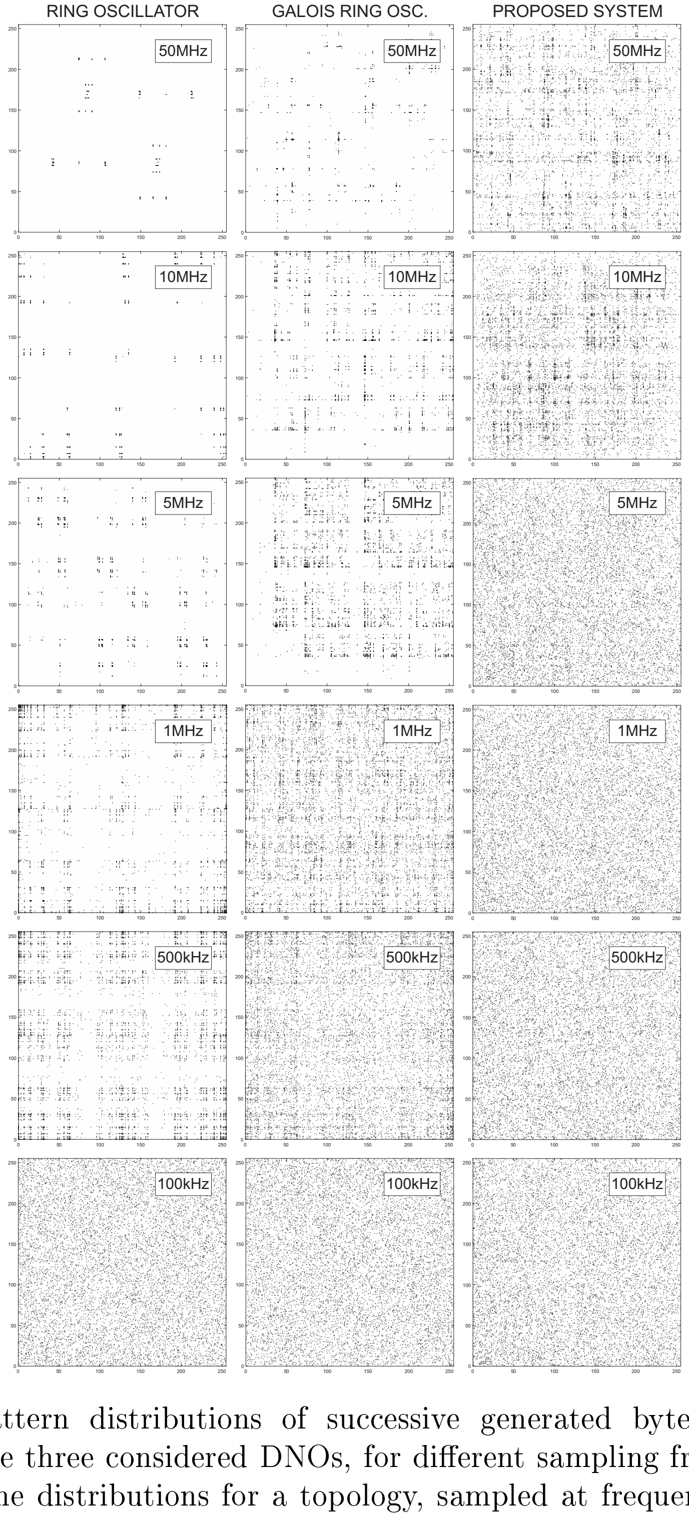


Figure 2.9: Pattern distributions of successive generated bytes on the plane $(b_n, b_{n+1})$, for the three considered DNOs, for different sampling frequencies. Each column shows the distributions for a topology, sampled at frequencies going from 100 kHz (lower plots) to 50 MHz (upper plots).

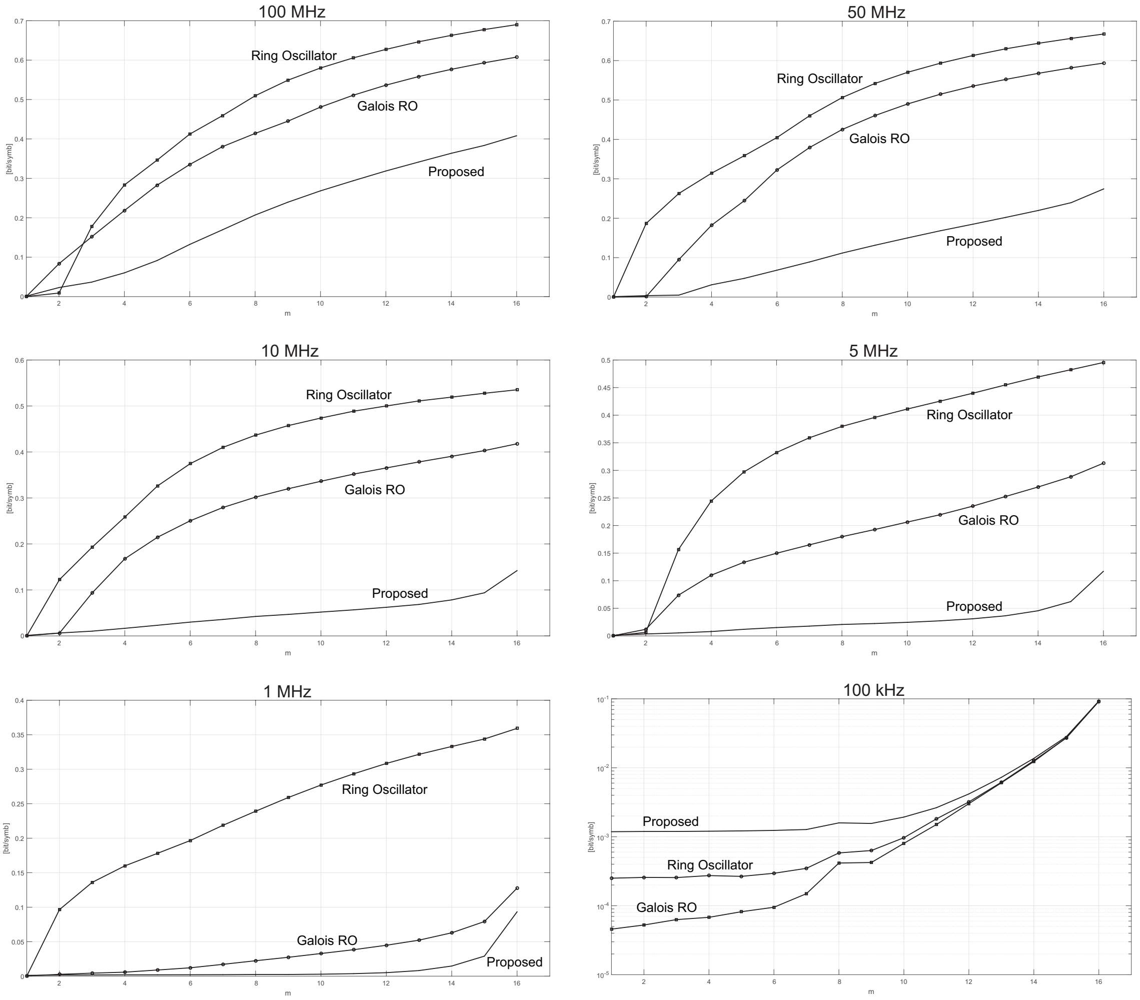


Figure 2.10: Average Shannon redundancy for binary words up to 16 bits, for the three systems, for different sampling frequencies.

evaluated at different frequencies. The custom DNO has lower redundancy than the other two systems at all frequencies except 100 kHz. In this case, the entropy is limited by a residual biasing of the mean value of the generated sequences.

Fig. 2.11 shows the autocorrelation function of the binary sequences evaluated for the three systems up to a time lag equal to 40, at different frequencies. The gray dashed line represents the ideal level for time lags greater than 0, which for an ideal binary random source should be $0.5^2 = 0.25$. The red dashed line represents the asymptotic value of the estimated autocorrelation function, equal to the square of the mean value of the sequence. The custom DNO achieves the asymptotic autocorrelation value much faster that the other two DNOs, regardless of the sample rate.

Fig. 2.12 shows the runs statistics, that are sequences of consecutive equal bits, evaluated for both 0s and 1s, up to runs of 6 bits, comparing the three systems at different sampling frequencies. The dashed gray line represents the ideal reference

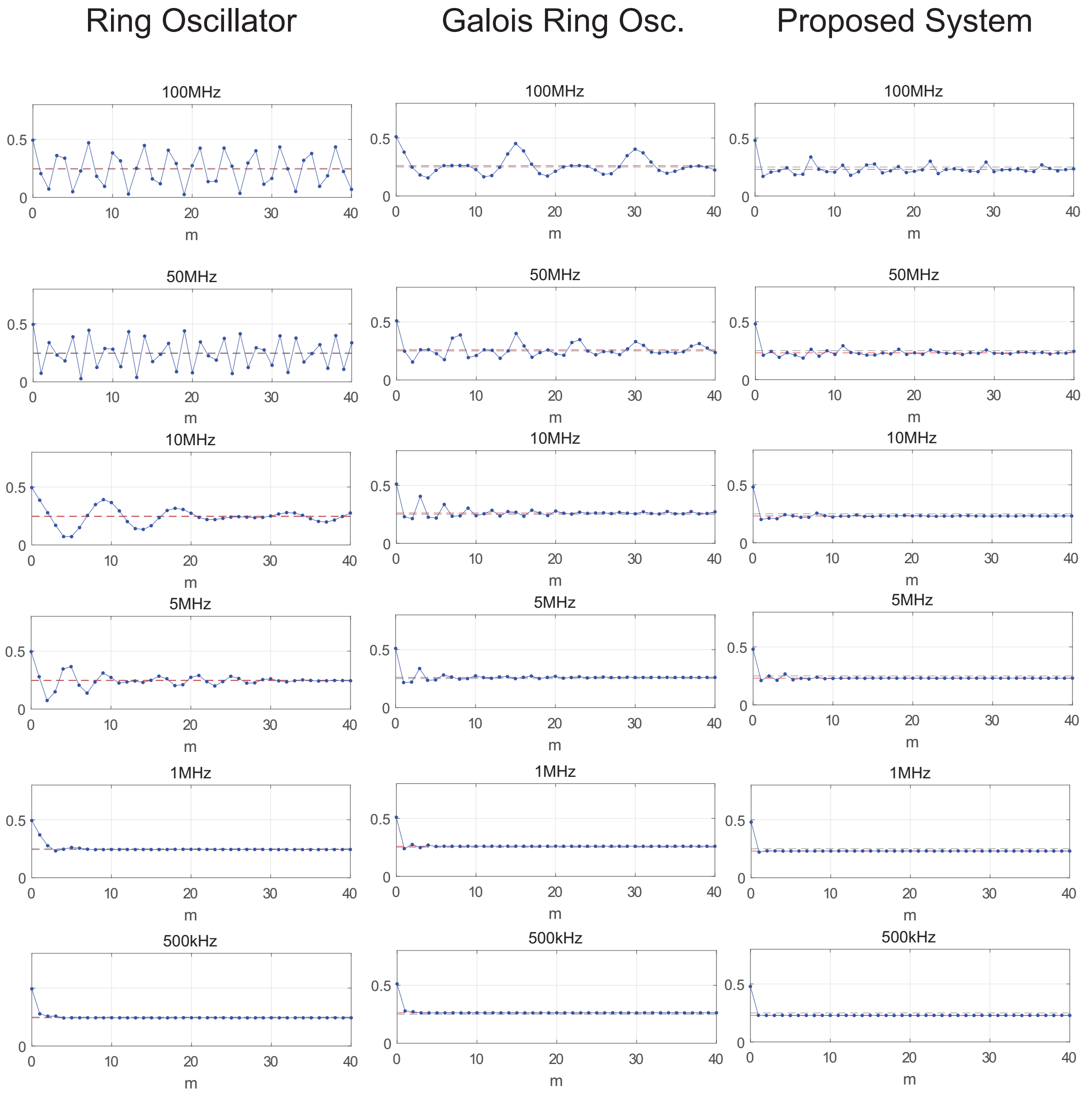


Figure 2.11: Autocorrelation function of the collected binary streams up to the time lag 40, for different sampling frequencies. Gray-dashed line: ideal level for time lag $m > 0$; red-dashed line: asymptotic value of the estimated autocorrelation function.

level. The custom DNO manages to approach the ideal level at all sampling frequencies, unlike the Ring Oscillator and the Galois Ring Oscillator, which instead approach the ideal level only up to 500 kHz and 1 MHz, respectively.

Fig. 2.13 shows the probability distributions of the 8-bits symbols generated by the three systems at the different frequencies. The dashed red line represents the ideal reference level, corresponding to a uniform distribution characterized by symbols with probability equal to 1/256. It is evident that the symbols generated by the custom DNO are distributed more evenly than the Ring Oscillator and the Galois Ring Oscillator regardless of the sampling rate.

Summarizing the observed data, we can affirm that the three systems are characterized by different performance. In particular, the custom DNO reaches levels of randomness higher than the other two systems.

In conclusion, the example shows, through informal methods of investigation, that the DNOs constitute a class of entropy sources with very different characteristics, justifying the need to define new methodologies for their analysis, aimed at the conscious design of circuit solutions, capable of achieving satisfactory performance for cryptographic applications.

## 2.4 Conclusion

We introduced the circuit class of Digital Nonlinear Oscillators (DNOs), i.e. circuits that can be used as entropy sources for the design of True Random Number Generators. DNOs are nonlinear dynamical systems capable of supporting complex dynamical behaviors in the time-continuous domain, although they are based on purely digital hardware.

We explored the possibility of implementing such circuits on Programmable Logic Devices, with a particular focus on their implementation on FPGAs. In this sense, we analyzed the internal structure of a chip of this type, investigating the role of their basic circuit elements in the design of DNOs.

Finally, we presented a comparison of the performance in terms of entropy generation of three notable topologies (Ring Oscillator, Galois Ring Oscillator and a custom topology), built using the same amount of hardware resources for each of them, so as to be able to perform a comparison mainly related to their dynamical characteristics.

From the comparison, it emerged that circuits with similar hardware complexity can offer particularly different dynamical characteristics based on how the topology is defined, thus justifying our interest in deepening the study of this class of circuits and in defining formalized methods for their design.

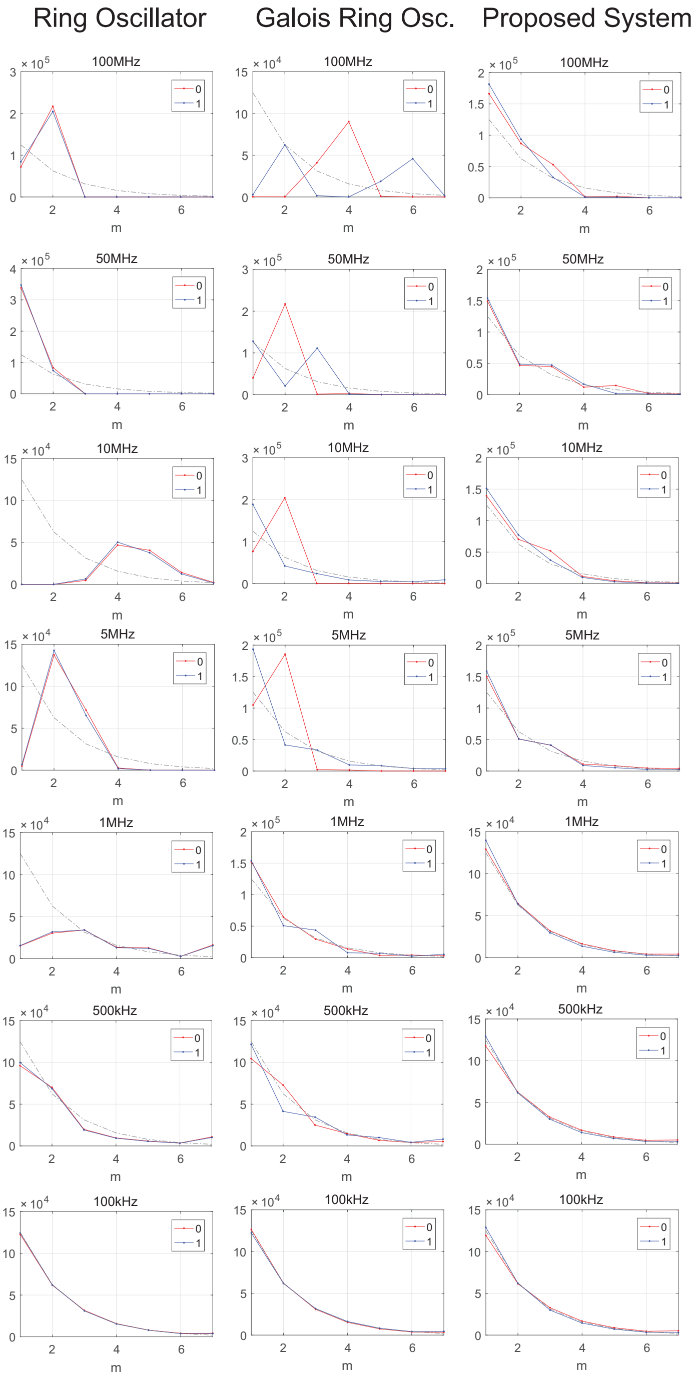


Figure 2.12: Runs occurrencies for 0s and 1s up to 6 bits, for different sampling frequencies. Gray-dashed line: reference ideal level.

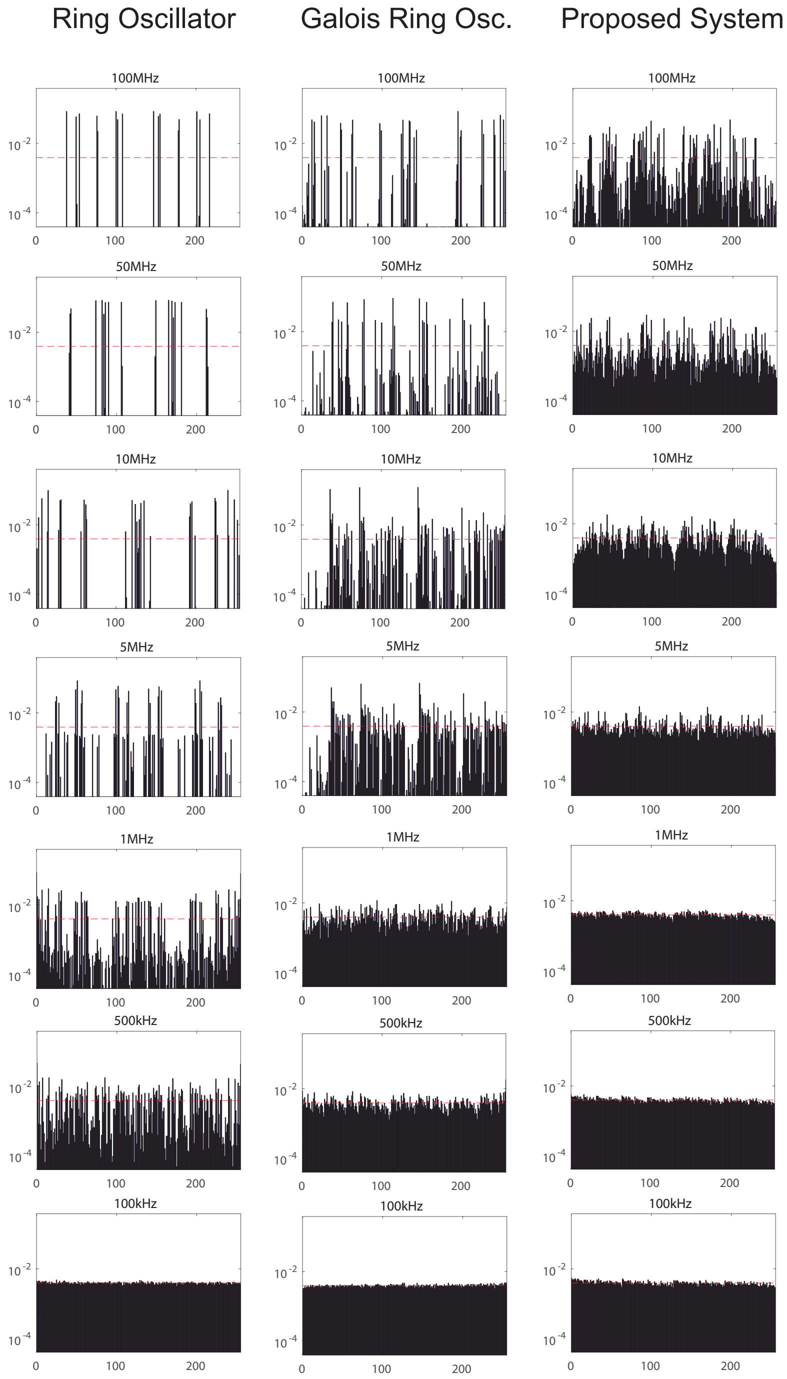


Figure 2.13: Probability distribution for 8-bit words, for the three systems, for different sampling frequencies. Red-dashed line: reference ideal level.

# Chapter 3

# Investigation Methods and Implementation Techniques

In this chapter we formalize a methodology for the analysis and design of Digital Nonlinear Oscillators based on the evaluation of their electronic implementation, their dynamical behavior and the information rate they can generate. The presented methodology makes use of different tools, such as figures of merit, simplified dynamical models, advanced numerical simulations and experimental tests carried out through implementation on FPGA. Each of these tools is analyzed both in its theoretical premises and through explanatory examples.

## 3.1 DNO Analysis: a Need for Investigation Methods

In Chapter 2 we introduced the class of circuits called Digital Nonlinear Oscillators (DNOs). The proposed arguments allow us to state that a DNO can be understood in three possible ways:

- a DNO is an analog electronic circuit built using digital hardware;
- a DNO is a complex dynamical system capable of supporting periodic or chaotic dynamics;
- a DNO is a source of entropy that can be used for the generation of random numbers.

Obviously, physical implementation, dynamical behavior and generated entropy are closely linked and interdependent aspects in DNOs.

For this reason, to formalize a methodology for the analysis and design of circuits belonging to this class, it is necessary to take into consideration all three natures of the considered systems. From this derives a complex approach to the study of DNOs, which requires the use of multiple tools to evaluate the performance from every point of view. Specifically, the analysis methods and the implementation techniques that we employed in the context of this work can be framed as follows:

- figures of merit for the evaluation of the statistical characteristics;
- simplified dynamical models for the assessment of relevant aspects related to system stability;
- circuit simulation of solutions based on CMOS technologies for more in-depth dynamical evaluations;
- physical implementation of circuits in FPGA for the experimental validation of the theorized and simulated behaviors.

In the next sections each of these tools is analyzed, providing their theoretical premises and some explanatory examples.

### 3.1.1 References

The material presented in this chapter includes results that have been published in the following publications:

- T. Addabbo, A. Fort, R. Moretti, M. Mugnaini, V. Vignoli, and M. G. Bosque, "Lightweight true random bit generators in plds: Figures of merit and performance comparison," in *2019 IEEE International Symposium on Ciruits and Systems (ISCAS)*. IEEE, 2019, pp. 1-5 [37].
- T. Addabbo, A. Fort, R. Moretti, M. Mugnaini, and V. Vignoli, "Analysis of a circuit primitive for the reliable design of digital nonlinear oscillators," in *2019 15th Conference on Ph. D Research in Microelectronics and Electronics (PRIME)*. IEEE, 2019, pp. 189-192 [38].

## 3.2 Figures of Merit

The first tools that we consider are two figures of merit for the comparative evaluation of the statistical characteristics of DNOs.

A DNO is as a device that can be used for the generation of random numbers. To evaluate the quality of an object of this type, the state of the art provides the application of standard statistical tests [4, 39].

However, this approach constitutes a poorly informative method regarding the actual statistical characteristics of the circuit. In fact, the statistical tests are limited to providing an absolute pass/fail outcome, which only establishes whether the considered system complies with the minimum quality that agrees with the standard. Taking a set of systems capable of passing these tests, we are unable to determine which of these systems are better or worse relying solely on the tests outcome. In addition to this, given a set of arbitrary tests, it is always possible to identify an adequate invertible post-processing algorithm capable of manipulating the data generated by a system in order to make it pass the tests [5].

For this reason it is useful to introduce figures of merit that allow to evaluate, in the comparison between two or more sources, which ones are capable of offering better performance [37]. Obviously this type of analysis has a comparative value only, and is not intended to replace standard statistical tests, which instead establish in absolute terms whether a source of entropy used in the generation of random numbers is valid or not for a specific application.

### 3.2.1 Decorrelation Time

The first figure of merit we consider is the Decorrelation Time. To provide a definition of this figure, we must first introduce some notations and definitions.

**Definition 3.1.** *Given an ergodic information source that generates a binary sequence $S = \{s_i : i \in \mathbb{N}\}$, we say that the source has a vanishing statistical dependence if for each k-tuple of random variables $\{s_{j_1}, s_{j_2}, \dots, s_{j_k} : j \in \mathbb{N}, k \in \mathbb{N}, 0 \leq j_1 < j_2 < \cdots < j_k\}$ and for each $\varepsilon \in \mathbb{R}^+$, an index $m_0 \in \mathbb{N}$ exists such that if $m \geq m_0$ then $|P(s_{j_k+m}|s_{j_1}, s_{j_2}, \dots, s_{j_k}) - P(s_{j_k+m})| < \varepsilon$, or more succintly:*

$$\lim_{m\to\infty} P(s_{j_k+m}|s_{j_1}, s_{j_2}, \dots, s_{j_k}) = P(s_{j_k+m}) = P(s), \tag{3.1}$$

*where $P$ is a measure of probability and $P(A|B) = P(A\cap B)/P(B)$ is the conditional probability for two events $A$ and $B$.*

In general, Definition 3.1 is valid for any circuit characterized by free oscillations and affected by electronic noise, uniformly sampled by adopting a 1-bit quantization resolution [37].

**Theorem 3.1.** *Given an ergodic information source with vanishing statistical dependence that generates a binary sequence $S = \{s_i : i \in \mathbb{N}\}$, the limit of the autocorrelation function associated with the sequence $R_S(m) = E[s_i s_{i+m}]$ for $m \to \infty$ is equal to $[P(s = 1)]^2 = R_S^2(0)$.*

*Proof.* The autocorrelation function associated with the sequence $S$ depends on the expected value of the symbols in the sequence and their covariance:

$$R_S(m) = E[s_i s_{i+m}] = E[s_i]E[s_{i+m}] + \mathrm{Cov}(s_i, s_{i+m}) = (E[s])^2 + \mathrm{Cov}(s_i, s_{i+m}). \tag{3.2}$$

The expected value of a binary random variable is equal to the probability that the variable has a value of 1:

$$E[s] = 0 \cdot P(s = 0) + 1 \cdot P(s = 1) = P(s = 1), \tag{3.3}$$

and it is also equal to the autocorrelation function for $m = 0$. Since the information source has a vanishing statistical dependence, according to (3.1), for $m \to \infty$ the two symbols $s_i$ and $s_{i+m}$ can be considered statistically independent. This implies that the covariance for $m \to \infty$ tends to 0. In conclusion, we have that:

$$\lim_{m\to\infty} R_S(m) = [P(s=1)]^2 = R_S^2(0). \tag{3.4}$$

□

Let us now consider a DNO whose output is sampled at a frequency $f_s$ to acquire a test sequence with finite length of $N$ bits.

The source autocorrelation function $R_S(m) = E[s_i s_{i+m}]$, with $0 \le m \le M \le N-1$ can be estimated using the following formula:

$$\tilde{R}_S(m) = \frac{1}{N-m} \sum_{i=0}^{N-1-m} s_i s_{i+m}. \tag{3.5}$$

Assuming that the DNO is an ergodic source with vanishing statistical dependence, by Theorem 3.1 the autocorrelation function tends asymptotically to the value $\tilde{R}_S^2(0)$. We then introduce the normalized autocorrelation function $\phi_S : \{0, 1, \ldots, M\} \to [0,1] \subset \mathbb{R}$:

$$\phi_S(m) = \left| \frac{\tilde{R}_S(m) - \tilde{R}_S^2(0)}{\tilde{R}_S(0) - \tilde{R}_S^2(0)} \right|. \tag{3.6}$$

**Definition 3.2.** *Given a DNO that respects the condition of an ergodic source with vanishing statistical dependence, sampled at frequency $f_s$ to acquire an $N$-bits long sequence $S$, the Decorrelation Time $\tau_S(M,\eta)$ associated to the sequence $S$ on a window of $M+1 \le N$ bits with energy ratio $\eta$, where $\eta \in [0,1] \subset \mathbb{R}$, is defined as the minimum time necessary for the residual normalized energy associated to the normalized autocorrelation function $\phi_S$ to be less than $1-\eta$, that is:*

$$\tau_S(M,\eta) = \frac{k_{min}}{f_s} \quad [s], \tag{3.7}$$

*where:*

$$k_{min} = \min_{k \le M} \frac{\sum_{m=0}^{k} \phi_S^2(m)}{\sum_{m=0}^{M} \phi_S^2(m)} \ge \eta. \tag{3.8}$$

As shown in Fig. 3.1, the product between $f_s$ and $\tau_S(M,\eta)$ defines the minimum number of sampling periods to reach the energy ratio $\eta$ estimated on the interval $[0, M]$.

### 3.2.2 Average Shannon Entropy

The second figure of merit to evaluate the performance of a DNO is the Average Shannon Entropy.

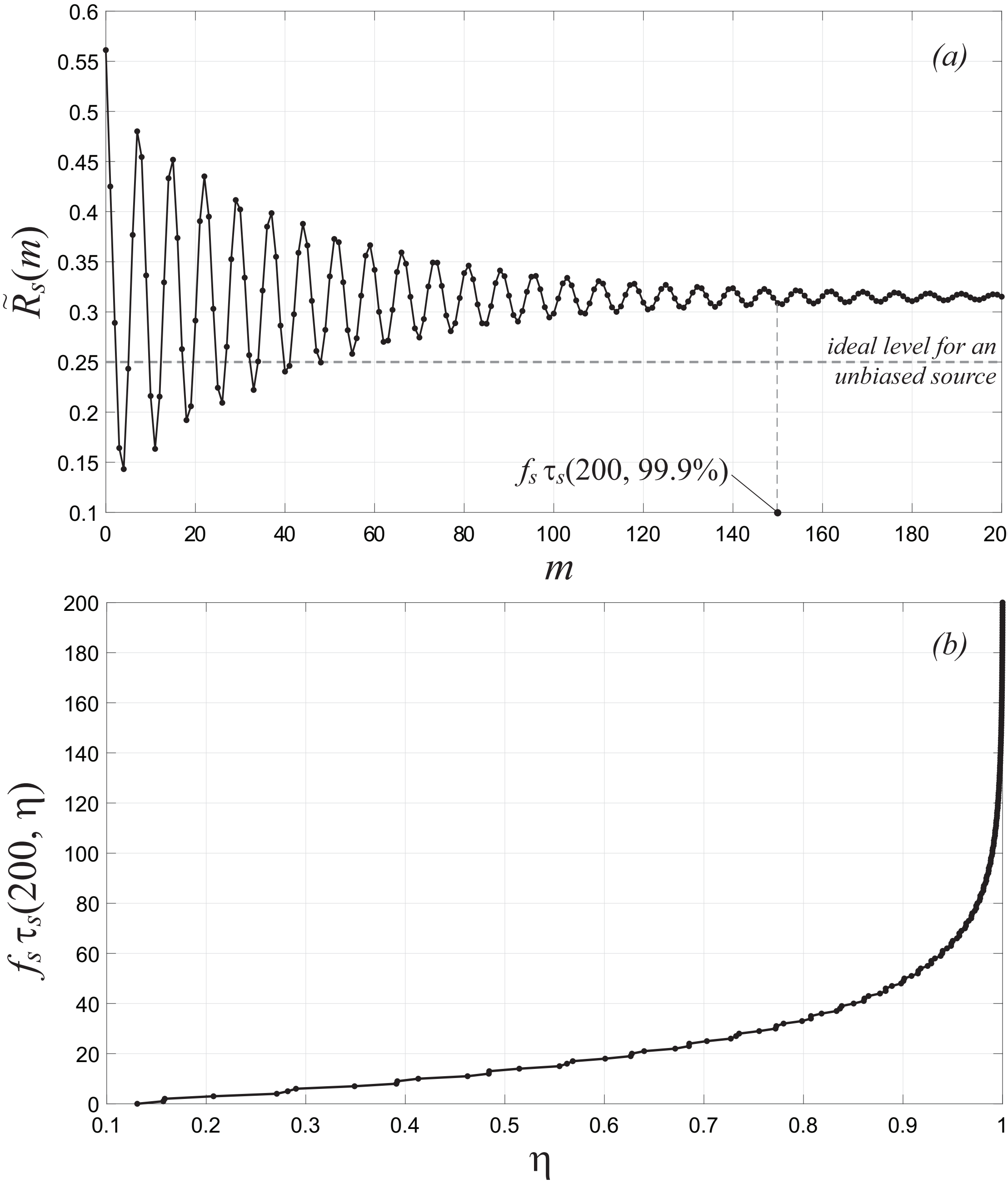


Figure 3.1: The vanishing autocorrelation function (a) of a DNO under test, sampled uniformly with $f_s = 50MHz$, and the correspondent Decorrelation Time (here normalized and represented as $f_s \cdot \tau_S(M, \eta)$), as a function of $\eta$ for $M = 200$ (b).

Let us again consider a DNO sampled at frequency $f_s$ to generate a $N$-bits long sequence. Suppose to collect the generated bits grouping them into $n$-bits long symbols, thus obtaining a sequence of $\lfloor N/n \rfloor$ symbols.

By indicating with $\{B_i : i = 0, 1, \ldots, 2^n - 1\}$ the set of all possible $n$-bits symbols, the probability of generating the $i$-th symbol can be estimated as follows:

$$\tilde{P}(B_i) = \frac{\#B_i}{\lfloor N/n \rfloor}, \tag{3.9}$$

where $\#B_i$ is the number of occurrences of the $i$-th symbol in the generated sequence.

**Definition 3.3.** *Given a DNO sampled at frequency $f_s$ to generate a sequence of $N$ bits grouped in $n$-bits words, thus obtaining a sequence of $\lfloor N/n \rfloor$ symbols, the Average Shannon Entropy (ASE) is defined as:*

$$ASE(n) = -\frac{1}{n} \sum_{i=0}^{2^n-1} \tilde{P}(B_i) \log_2 \tilde{P}(B_i) \quad \textit{[bit/sym]}. \tag{3.10}$$

The product of the ASE and the sampling frequency $f_s$ defines the average amount of information per second generated by the DNO.

### 3.2.3 Example: Comparison of three DNO Topologies

To show an example of application of the introduced figures of merit, let's consider the three topologies analyzed in Section 2.3, shown again in Fig. 3.2.

The three DNOs were implemented in five Xilinx Artix 7 xc7a35 FPGAs, designing in each chip and for each DNO 16 oscillators in different positions (same positions for each analyzed topology), obtaining a total of 80 DNO instances. The different topologies differ in the LUTs thruth tables and routing, whereas using the same amount of slices. Each implementation was sampled at defined frequencies ranging between 100 kHz and 100 MHz, collecting one million bits long sequences in any case. Each sequence was used to calculate both the Decorrelation Time and the Average Shannon Entropy.

The Decorrelation Time was estimated by setting in (3.7) $M = 200$ and $\eta = 0.999$, properly selecting, among the chosen set of sampling frequencies, the highest $f_s$ such to experience the adequate vanishing of the autocorrelation function in the observation time window $[0, M/f_s]$. The choice of $M$ and $\eta$ influences the result of the estimate in absolute terms, but a reasonable choice of parameters, based on heuristic considerations, allowed for a reliable comparison of the systems under test.

Fig. 3.3 summarizes the obtained results, reporting the statistics of the Decorrelation Times (average, minimum, maximum, $10^{\text{th}}$ and $90^{\text{th}}$ percentiles) for each of the five tested chips. It is evident that the three DNOs are characterized by significantly different Decorrelation Times, although the implementations used the same hardware resources. It is also interesting to note that, taking a topology, the average values of the Decorrelation Times are weakly variable between among the

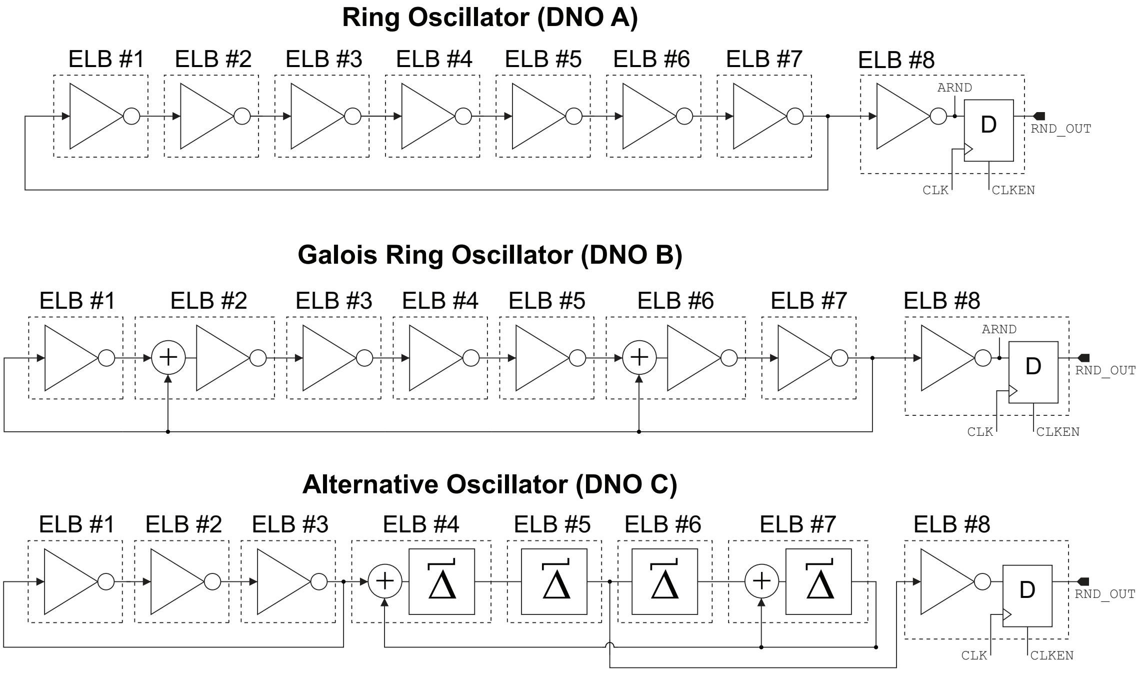


Figure 3.2: The different DNO architectures considered for comparison according to the evaluation of their Decorrelation Times and Average Shannon Entropies.

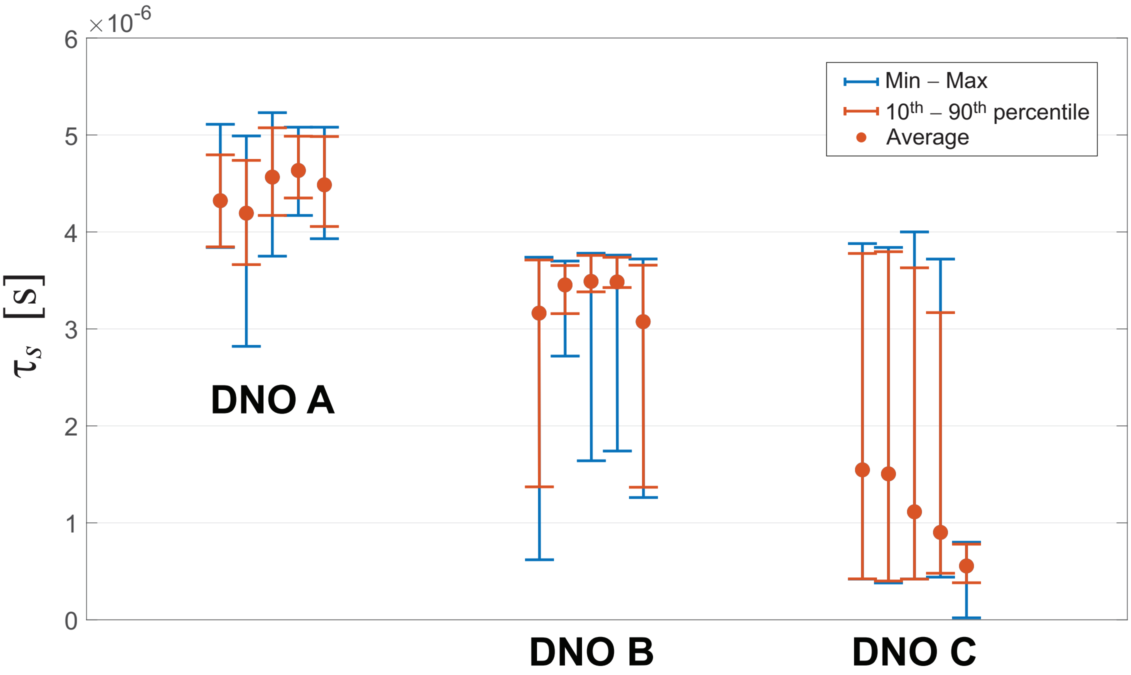


Figure 3.3: Decorrelation Times of the three DNO topologies, evaluated for 80 instances of each topology, implemented on 5 Xilinx Artix 7 xc7a35 FPGAs. For each chip, the average, minimum, maximum, 10th and 90th percentiles of the 16 instances implemented on that chip are reported.

chips, suggesting that the Decorrelation Time is intrinsically related to the specific topology.

The Average Shannon Entropy was evaluated for symbols with a length between 1 and 16 bits. The obtained results are similar among the tested chips, regardless of the sampling frequency, therefore in Fig. 3.4 we report the results obtained for a single FPGA by evaluating the ASE for 10-bits long symbols. Again, the figure shows the average, minimum, maximum, $10^{\text{th}}$ and $90^{\text{th}}$ percentiles of the ASE, comparing the values for the different sampling frequencies.

Putting together the results shown in Fig. 3.3 and Fig. 3.4, it is possible to find a link between Decorrelation Time and Average Shannon Entropy: on average, the shorter is the Decorrelation Time the higher is the ASE.

Fig. 3.4 also highlights how ASE and sampling frequency are linked by a nonlinear relationship, for which a variation in the sampling frequency involves a marginal variation of the ASE. In this sense, the frequency at which we sample our source has a significant weight in terms of the rate of generated information, as evidenced by the Average Shannon Entropy per second (ASEpS) shown in Fig. 3.5.

## 3.3 Study of Simplified Dynamical Models

Another method of analysis we employed to study Digital Nonlinear Oscillators is the study of DNOs simplified dynamical models.

As already highlighted several times in the previous sections, a DNO is a network composed of circuits that in the digital domain implement logic functions, but which in the analog domain are characterized by DC nonlinear transfer functions. From the dynamics point of view, the parasitic components linked to the technology used for the design of the circuits and to the connections between one circuit and another determine not negligible signal propagation times, which can trigger more or less complex dynamics at the DNO level.

Taking these characteristics into account, we defined a simplified model for the description of a DNO, having the purpose, given a certain topology, to investigate which conditions favor compatibility with complex dynamics on the basis of the stability of its fixed points [38]. The model is designed to be used in a preliminary analysis of the DNO, in which it is not intended to evaluate the transient behavior of the circuit. The observation of these aspects, in fact, requires more advanced tools, such as numerical simulators based on BSIM4 models.

Our proposal, shown in Fig. 3.6, foresees to represent each node of a DNO with a first order cell, composed of a voltage controlled voltage generator that controls a resistance-capacitance ($RC$) cell of the first order:

$$\frac{dv_o}{dt} = \frac{g(\boldsymbol{v}_i) - v_o}{RC}. \tag{3.11}$$

$v_o$ is the output voltage of the node, $\boldsymbol{v}_i \in \mathbb{R}^m$ is a column vector that collects the input voltages of the node, $g : \mathbb{R}^m \to \mathbb{R}$ is the DC analog transfer function of the node.

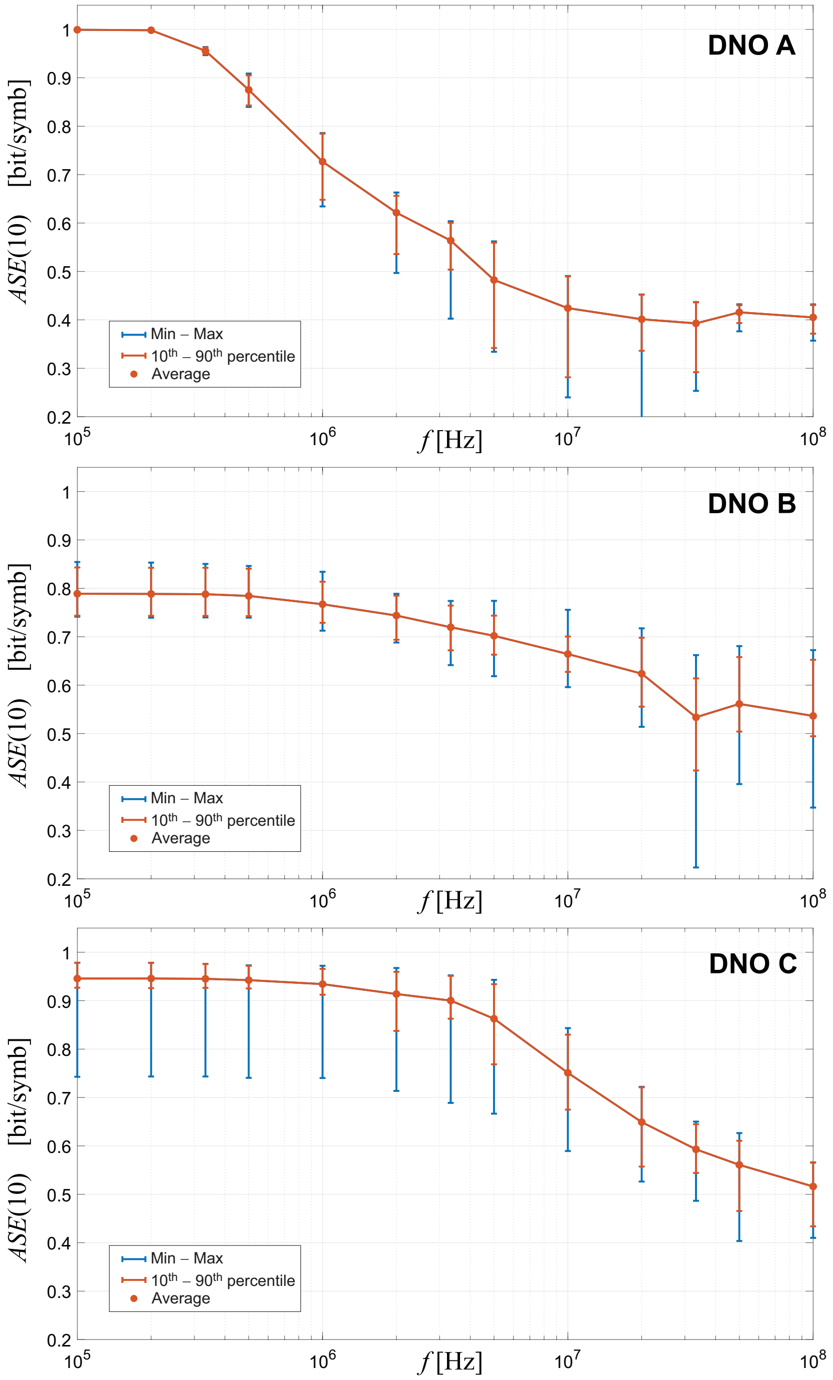


Figure 3.4: Average Shannon Entropies of the three DNO topologies for 10-bit words according to different sampling frequencies, evaluated for 16 instances of each topology, implemented on a Xilinx Artix 7 xc7a35 FPGAs. The average, minimum, maximum, $10^{\text{th}}$ and $90^{\text{th}}$ percentiles of the 16 implemented instances are reported. Similar results were obtained repeating the measurements on four other chips.

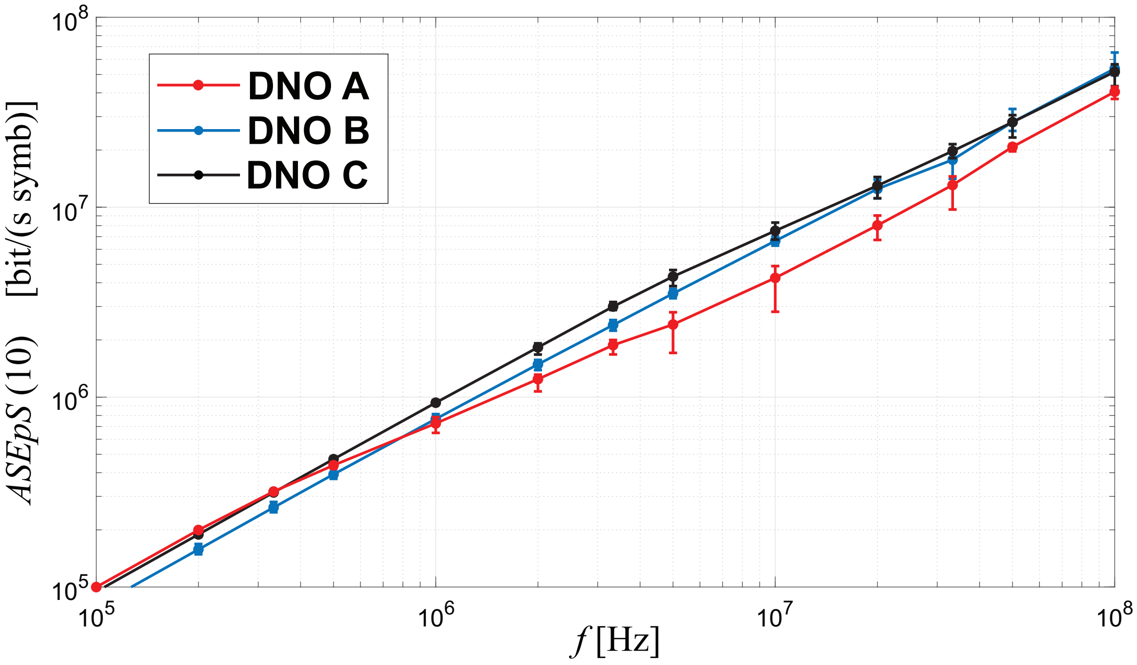


Figure 3.5: Average Shannon Entropies per Second of the three DNO topologies for 10-bit words according to different sampling frequencies, evaluated for 16 instances of each topology, implemented on a Xilinx Artix 7 xc7a35 FPGAs. The average, minimum and of the 16 implemented instances are reported. Similar results were obtained repeating the measurements on four other chips.

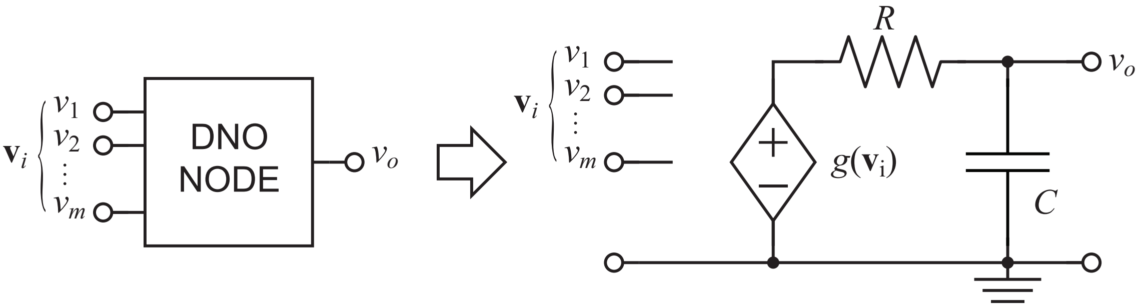


Figure 3.6: A first-order simplified nonlinear dynamical model used to investigate the DNOs fixed points and their stability. Each node of a DNO is represented with a first order cell, given by a voltage controlled voltage generator that controls a resistance-capacitance ($RC$) cell of the first order.

By adopting this representation, a $N$-nodes DNO can be investigated by means of the following nonlinear generalized dynamical system of order $N$:

$$\frac{d\boldsymbol{v}_o}{dt} = \boldsymbol{F}[\boldsymbol{g}(\boldsymbol{v}_o) - \boldsymbol{v}_o] = \boldsymbol{G}(\boldsymbol{v}_o). \tag{3.12}$$

$\boldsymbol{v}_o = \{v_{oi} : i = 1, 2, \ldots, N\} \in \mathbb{R}^N$ is a column vector representing the state of the DNO (defined by the output voltages of all nodes), $\boldsymbol{g} : \mathbb{R}^N \to \mathbb{R}^N$ is the column vector of the DC analog transfer functions of each node of the DNO, $\boldsymbol{F} \in \mathbb{R}^{N\times N}$ is a diagonal matrix whose diagonal elements are the reciprocals of the time constants defined by the $RC$ cells of each node.

The fixed points of this system are the values of $\boldsymbol{v}_o$ for which the following condition holds:

$$\boldsymbol{g}(\boldsymbol{v}_o) = \boldsymbol{v}_o. \tag{3.13}$$

Assuming that $\boldsymbol{g}$ is smooth and differentiable, the stability of the fixed points can be evaluated by studying the real part of the eigenvalues $\lambda$ of the Jacobian matrix $\boldsymbol{J} = (\frac{\partial \boldsymbol{G}}{\partial v_{o1}}, \frac{\partial \boldsymbol{G}}{\partial v_{o2}}, \ldots, \frac{\partial \boldsymbol{G}}{\partial v_{oN}})$ calculated at the fixed points themselves.

The use of this model requires providing a description of the DC transfer functions of the DNO nodes. Obviously, the quality of this description influences the accuracy of the obtained result. Without losing generality with respect to the presented approach, our choice was to build the transfer functions based on the analytical composition of parametrized normalized sigmoids such as:

$$\phi(x, a, b) = \frac{1}{1 + e^{a(x-b)}}, \tag{3.14}$$

where $x \in [0, 1] \subset \mathbb{R}$, $a \in \mathbb{R}\backslash\{0\}$ and $b \in (0, 1) \subset \mathbb{R}$. On the basis of the sign of $a$, it is possible to represent through the sigmoid the transfer function associated with an inverter or a digital buffer, as shown in Fig. 3.7.

In a practical application, through an appropriate choice of $a$ and $b$, the sigmoids can be used for the nonlinear fitting of the DC transfer functions of real logic gates, as shown in Fig. 3.8, where the DC transfer function of a NOT gate is modeled as follows:

$$v_0(v_i) = \mathrm{NOT}(v_i) \approx \phi(v_i, a, b), \quad a > 0. \tag{3.15}$$

We defined the sigmoid in (3.14) as normalized, as its domain and codomain are equivalent to those of a DC transfer function of CMOS digital circuits with 1 V power supply.

More complex logic functions than an inverter or a digital buffer can be obtained by combining sigmoids. For example, a two-inputs XOR gate can be represented using the following analytical model in 2D, as shown in Fig. 3.9:

$$\begin{aligned} v_o(v_{i1}, v_{i2}) &= \mathrm{XOR}(v_{i1}, v_{i2}) = \\ &= \phi(v_{i1}, a_1, b_1)\phi(v_{i2}, a_2, b_2) + \phi(v_{i1}, a_3, b_3)\phi(v_{i2}, a_4, b_4), \\ &a_1, a_4 < 0, \quad a_2, a_3 > 0. \end{aligned} \tag{3.16}$$

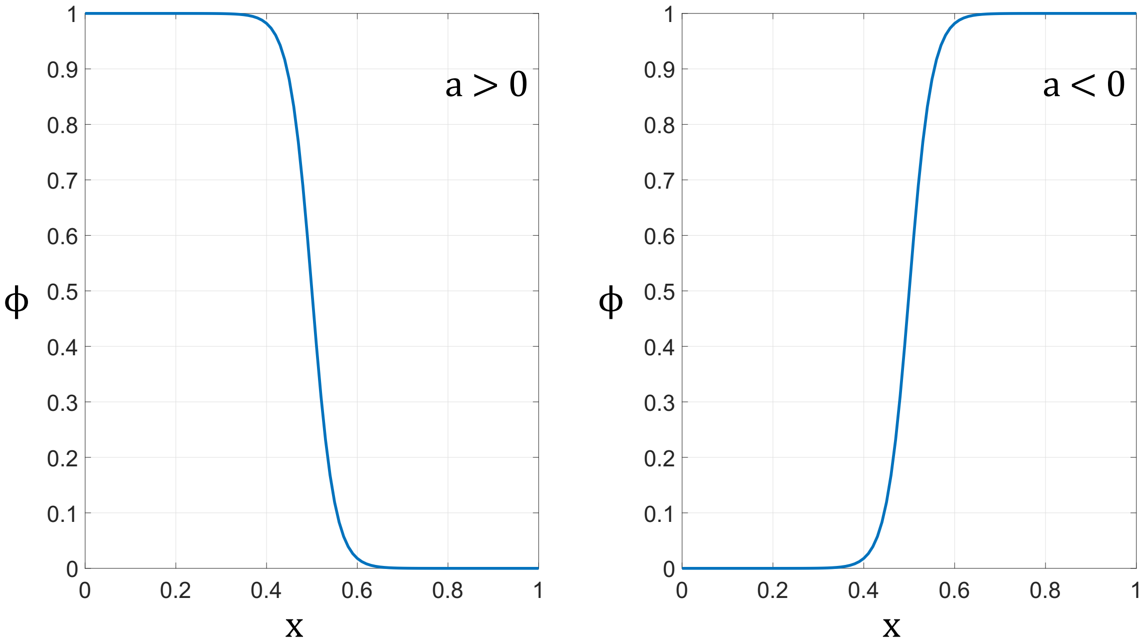


Figure 3.7: Sigmoids computed according to (3.14) setting $a = \pm 40$ and $b = 0.5$. $a > 0$ provides the primitive model for an inverter, while $a < 0$ provides the primitive model for a digital buffer.

### 3.3.1 Example: Analysis and Optimized Design of a DNO Sub-Circuit Primitive

To provide a practical example of the application of the simplified dynamical model, we refer to the Galois Ring Oscillators proposed by Golić. As already explained in Subsection 2.3.2, a Galois Ring Oscillator consists of an array of $N > 1$ digital gates combined with multiple feedbacks, as shown in Fig. 3.10.

Regardless of the complexity of the topology, a Galois Ring Oscillator always terminates with a feedback loop having the structure represented in Fig. 3.11.

This sub-circuit, consisting of a first node that implements a two-inputs logic function and a cascade of $k$ nodes with one input, acts as a trigger for the dynamics of the entire topology. For this reason, it is interesting to understand what are the minimum necessary conditions to make it start to oscillate.

More in detail, without loss of generality, suppose, with reference to Fig. 3.11, that the block $f_2$ defines a XOR function and that the blocks $f_{1,j}$, $j >= 0$, define digital buffers, from now on called DEL blocks, as shown in Fig. 3.12.

In this case, the purpose of applying the model is to understand the minimum number of DEL blocks that must be inserted in the feedback loop to trigger its oscillation. We build the model by applying (3.12) and (3.14), assuming $|a| = \alpha > 10$, $b = 0.5$ and $1/RC = \psi$.

We divide the study by assuming the independent input signal $x$ equal to 0 V and equal to 1 V. Limiting ourselves to these two situations and indicating with $v_i$ the feedback input signal, the transfer function of the XOR gate can be expressed

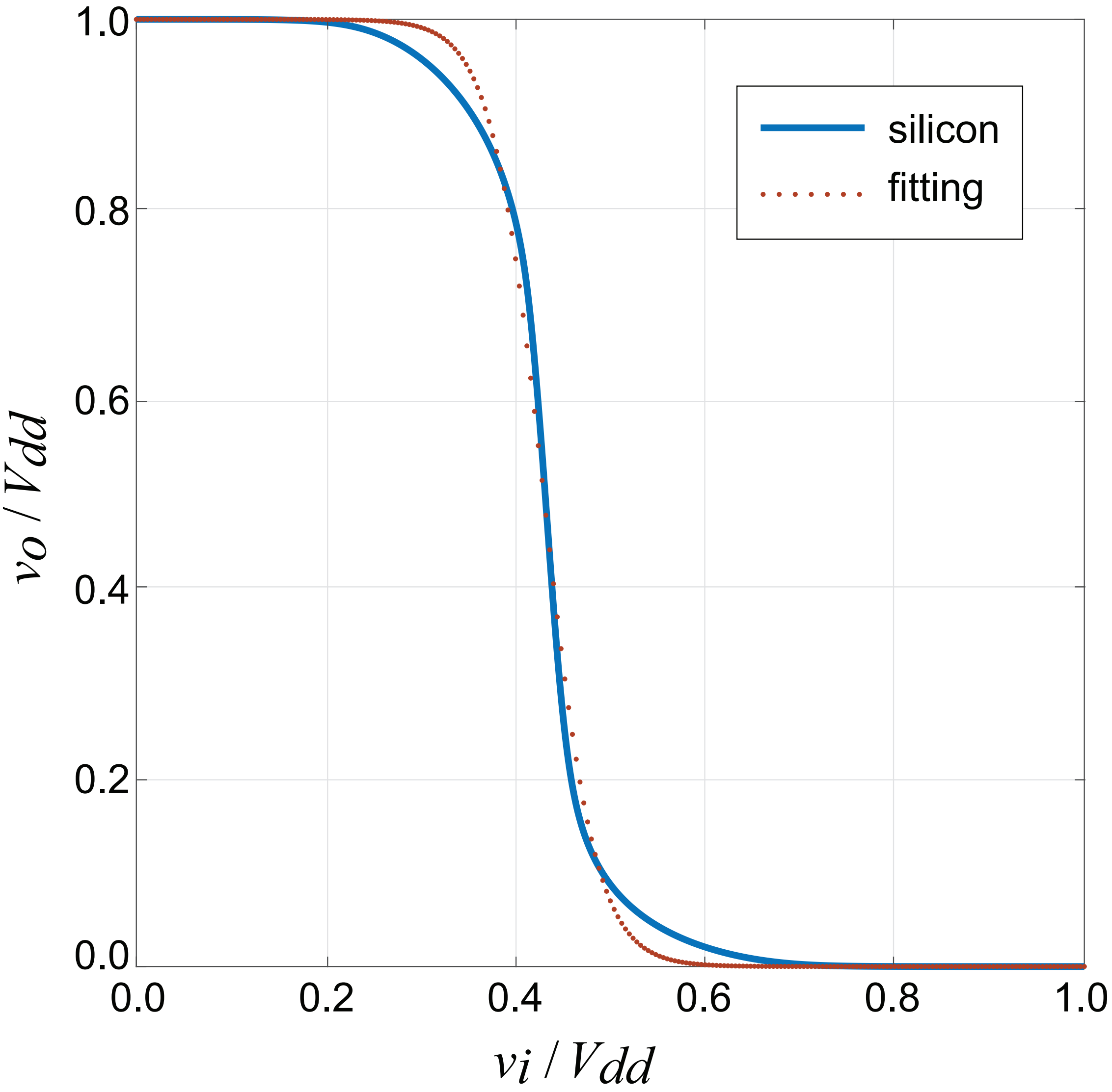


Figure 3.8: The DC transfer function of a CMOS inverter (UMC 180nm technology, 1.8V) and the fitting model (3.15), with $a \approx 36.81$, $b \approx 0.43$.

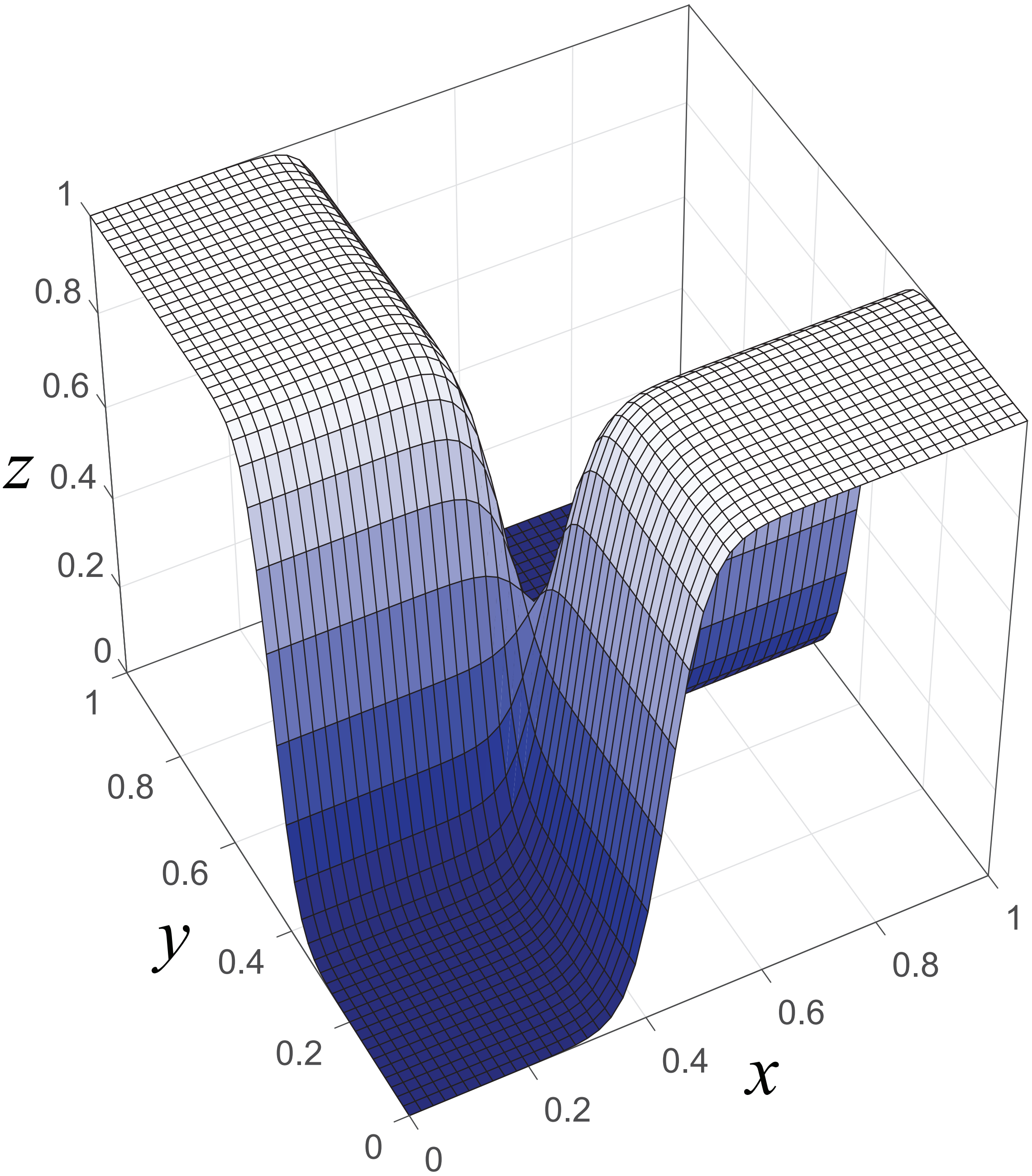


Figure 3.9: The sigmoid model of a two-inputs transfer function $z = \mathrm{XOR}(x, y)$ defined according to (3.16) for $a \approx 36.81$, $b \approx 0.43$.

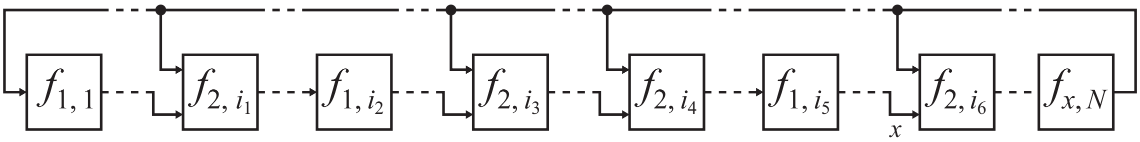


Figure 3.10: A low-complexity DNO topology, derived from the Galois Ring Oscillators proposed by Golić [24, 36].

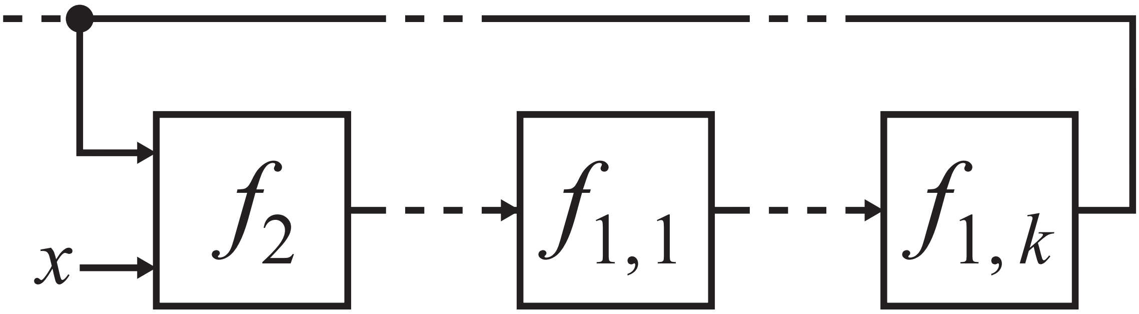


Figure 3.11: Feedback loop sub-network terminating any Galois Ring Oscillator, architecture, such as the one shown in Fig. 3.10.

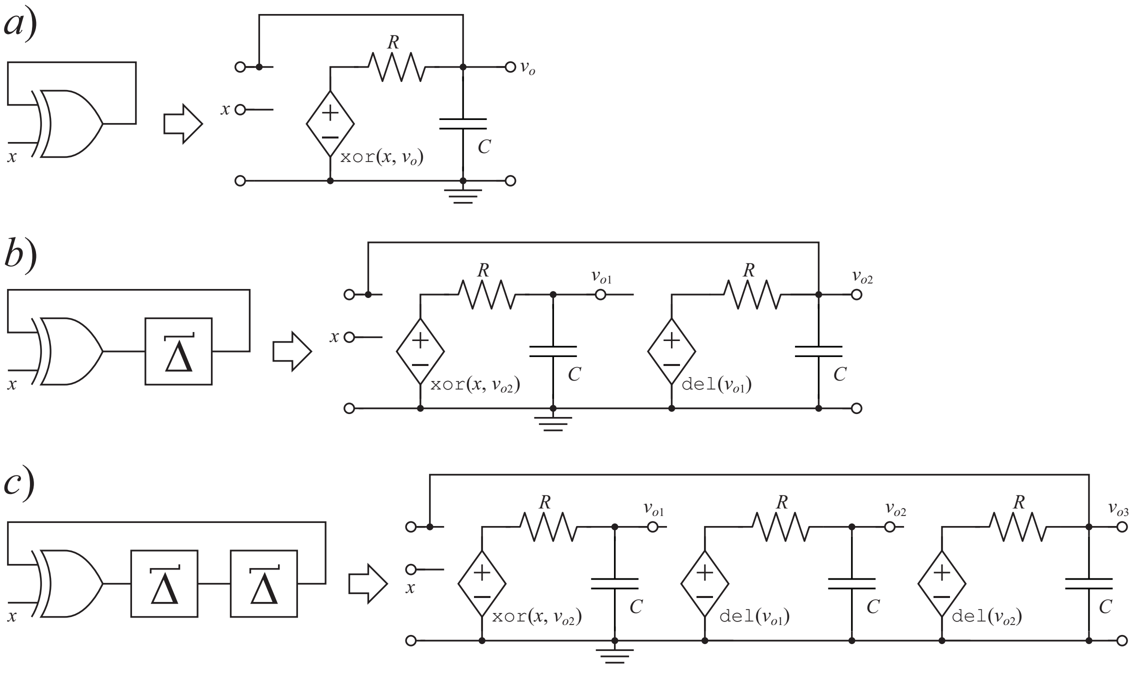


Figure 3.12: The simplified models to investigate the fixed points stability, for different implementations of the system shown in Fig. 3.11. With respect to the generic system of Fig. 3.11, it was assumed that the block $f_2$ defines a XOR function and that the blocks $f_{1,j}$, $j >= 0$, define digital buffers.

as follows:

$$\mathrm{XOR}(x, v_i)|_{x=p\in\{0,1\}} = \frac{1}{1+e^{a_p(v_i-0.5)}}, \tag{3.17}$$

where $a_0 = -\alpha$ and $a_1 = \alpha$.

Let's start our analysis from the simplest situation, in which the loop is composed solely of the XOR gate ($k = 0$). For the notation of the signals, refer to Fig. 3.12.a.

The dynamical system is described by a single equation:

$$\frac{dv_o}{dt} = [g(x, v_o) - v_o]\psi. \tag{3.18}$$

$g : \mathbb{R}^2 \to \mathbb{R}$ is the XOR gate transfer function (3.17).

According to (3.13), the fixed points are the solutions of the equation:

$$v_o = \frac{1}{1+e^{a_p(v_o-0.5)}}. \tag{3.19}$$

For $x \approx 0$ V, (3.18) has three solutions, namely $v_{0,A} = 0.5$ V, $v_{0,B} \approx 0$ V, $v_{0,C} \approx 1$ V. For $x \approx 1$ V, (3.18) has one solution, namely $v_{1,A} = 0.5$ V.

To determine the stability of these fixed points, we calculate the Jacobian matrix of the system, which in this case is limited to:

$$J(x, v_o) = \frac{\partial}{\partial v_o}\left[\frac{dv_o}{dt}\right] = \left[-\frac{a_p e^{a_p(v_o-0.5)}}{(1+e^{a_p(v_o-0.5)})^2} - 1\right]\psi. \tag{3.20}$$

The eigenvalues calculated on the fixed points are:

$$\begin{aligned}\lambda_1(v_{0,A}) &= J(0, v_{0,A}) = (\frac{\alpha}{4} - 1)\psi,\\ \lambda_1(v_{0,B}) &= J(0, v_{0,B}) \approx -\psi,\\ \lambda_1(v_{0,C}) &= J(0, v_{0,C}) \approx -\psi,\\ \lambda_1(v_{1,A}) &= J(1, v_{1,A}) = -(\frac{\alpha}{4} + 1)\psi.\end{aligned} \tag{3.21}$$

Having fixed $\alpha > 10$, we observe that $v_{0,A}$ is unstable (positive real eigenvalue), while the other fixed points are all stable (negative real eigenvalues). This implies that for $x \approx 0$ V the circuit has a bistable behavior, while if $x \approx 1$ V the circuit is stable. In both cases, the system cannot support oscillations.

Let us now consider a loop composed of a XOR gate and a DEL block ($k = 1$, Fig. 3.12.b).

The dynamical system is defined as follows:

$$\begin{cases}\frac{dv_{o1}}{dt} = [g_1(x, v_{o2}) - v_{o1}]\psi\\ \frac{dv_{o2}}{dt} = [g_2(v_{o1}) - v_{o2}]\psi\end{cases}. \tag{3.22}$$

$g_1 : \mathbb{R}^2 \to \mathbb{R}$ is the transfer function of the XOR gate (3.17), while $g_2 : \mathbb{R} \to \mathbb{R}$ is the transfer function of the DEL gate:

$$\mathrm{DEL}(v_i) = \frac{1}{1+e^{-\alpha(v_i-0.5)}}. \tag{3.23}$$

$v_i$ indicates the input voltage of a generic DEL gate.

The fixed points of the system are the solutions of:

$$\begin{cases} v_{o1} = g_1(x, v_{o2}) = \frac{1}{1+e^{ap(v_{o2}-0.5)}} \\ v_{o2} = g_2 \circ g_1(x, v_{o2}) = \frac{1}{1+e^{-\alpha\left(\frac{1}{1+e^{ap(v_{o2}-0.5)}}-0.5\right)}} \end{cases} . \tag{3.24}$$

For $x \approx 0$ V, (3.24) has three solutions, namely $\boldsymbol{v}_{0,A} = (0.5; 0.5)$ V, $\boldsymbol{v}_{0,B} \approx (0; 0)$ V, $\boldsymbol{v}_{0,C} \approx (1; 1)$ V. For $x \approx 1$ V, (3.24) has a solution, that is $\boldsymbol{v}_{1,A} = (0.5; 0.5)$ V.

In this case, the Jacobian matrix takes on 2x2 dimensions:

$$\begin{aligned} J(x, \boldsymbol{v}_o) &= \begin{bmatrix} \frac{\partial}{\partial v_{o1}} \left[\frac{dv_{o1}}{dt}\right] & \frac{\partial}{\partial v_{o2}} \left[\frac{dv_{o1}}{dt}\right] \\ \frac{\partial}{\partial v_{o1}} \left[\frac{dv_{o2}}{dt}\right] & \frac{\partial}{\partial v_{o2}} \left[\frac{dv_{o2}}{dt}\right] \end{bmatrix} = \\ &= \begin{bmatrix} -\psi & -\frac{a_p e^{ap(v_{o2}-0.5)}}{(1+e^{ap(v_{o2}-0.5)})^2}\psi \\ \frac{\alpha e^{-\alpha(v_{o1}-0.5)}}{(1+e^{-\alpha(v_{o1}-0.5)})^2}\psi & -\psi \end{bmatrix}. \end{aligned} \tag{3.25}$$

The eigenvalues associated with the generic fixed point $(x^*, \boldsymbol{v}_o^*)$ are the values of $\lambda$ for which the determinant of the matrix $\boldsymbol{J}(x^*, \boldsymbol{v}_o^*) - \lambda \boldsymbol{I}$ is zero:

$$\det(\boldsymbol{J}(x^*, \boldsymbol{v}_o^*) - \lambda \boldsymbol{I}) = 0. \tag{3.26}$$

By doing the calculations, the following eigenvalues are obtained:

$$\begin{aligned} \lambda_{1,2}(\boldsymbol{v}_{0,A}) &= -(1 \pm \frac{\alpha}{4})\psi, \\ \lambda_1(\boldsymbol{v}_{0,B}) &\approx -\psi, \\ \lambda_1(\boldsymbol{v}_{0,C}) &\approx -\psi, \\ \lambda_{1,2}(\boldsymbol{v}_{1,A}) &= -(1 \pm j\frac{\alpha}{4})\psi. \end{aligned} \tag{3.27}$$

Similarly to the $k = 0$ case, all the fixed points are stable (eigenvalues with negative real part), except for $\boldsymbol{v}_{0,A}$ which is unstable, as it has a positive real part eigenvalue. Again, the circuit appears to be bistable for $x \approx 0$ V and stable for $x \approx 1$ V, excluding the possibility of oscillation.

Since not even a DEL block is sufficient to trigger oscillations, we add an additional delay element ($k = 2$, Fig. 3.12.c).

Accordingly, the dynamical system is modified as follows:

$$\begin{cases} \frac{dv_{o1}}{dt} = [g_1(x, v_{o3}) - v_{o1}]\psi \\ \frac{dv_{o2}}{dt} = [g_2(v_{o1}) - v_{o2}]\psi \\ \frac{dv_{o3}}{dt} = [g_2(v_{o2}) - v_{o3}]\psi \end{cases} . \tag{3.28}$$

The fixed points of the system are the solutions of:

$$\begin{cases} v_{o1} = g_1(x, v_{o3}) = \frac{1}{1+e^{ap(v_{o3}-0.5)}} \\ v_{o2} = g_2 \circ g_1(x, v_{o3}) = \frac{1}{1+e^{-\alpha\left(\frac{1}{1+e^{ap(v_{o3}-0.5)}}-0.5\right)}} \\ v_{o3} = g_2 \circ g_2 \circ g_1(x, v_{o3}) = \frac{1}{1+e^{-\alpha\left(\frac{1}{1+e^{-\alpha\left(\frac{1}{1+e^{ap(v_{o3}-0.5)}}-0.5\right)}}-0.5\right)}} \end{cases} . \tag{3.29}$$

For $x \approx 0$, (3.29) has three solutions, namely $\boldsymbol{v}_{0,A} = (0.5; 0.5; 0.5)$ V, $\boldsymbol{v}_{0,B} \approx (0; 0; 0)$ V, $\boldsymbol{v}_{0,C} \approx (1; 1; 1)$ V. For $x \approx 1$ V, (3.29) has a solution, that is $\boldsymbol{v}_{1,A} = (0.5; 0.5; 0.5)$ V.

In this case, the Jacobian matrix takes on 3x3 dimensions:

$$\begin{aligned} J(x, \boldsymbol{v}_o) &= \begin{bmatrix} \frac{\partial}{\partial v_{o1}}\left[\frac{dv_{o1}}{dt}\right] & \frac{\partial}{\partial v_{o2}}\left[\frac{dv_{o1}}{dt}\right] & \frac{\partial}{\partial v_{o3}}\left[\frac{dv_{o1}}{dt}\right] \\ \frac{\partial}{\partial v_{o1}}\left[\frac{dv_{o2}}{dt}\right] & \frac{\partial}{\partial v_{o2}}\left[\frac{dv_{o2}}{dt}\right] & \frac{\partial}{\partial v_{o3}}\left[\frac{dv_{o2}}{dt}\right] \\ \frac{\partial}{\partial v_{o1}}\left[\frac{dv_{o3}}{dt}\right] & \frac{\partial}{\partial v_{o2}}\left[\frac{dv_{o3}}{dt}\right] & \frac{\partial}{\partial v_{o3}}\left[\frac{dv_{o3}}{dt}\right] \end{bmatrix} = \\ &= \begin{bmatrix} -\psi & 0 & -\frac{a_p e^{ap(v_{o3}-0.5)}}{(1+e^{ap(v_{o3}-0.5)})^2}\psi \\ \frac{\alpha e^{-\alpha(v_{o1}-0.5)}}{(1+e^{-\alpha(v_{o1}-0.5)})^2}\psi & -\psi & 0 \\ 0 & \frac{\alpha e^{-\alpha(v_{o2}-0.5)}}{(1+e^{-\alpha(v_{o2}-0.5)})^2}\psi & -\psi \end{bmatrix} . \end{aligned} \tag{3.30}$$

Starting from (3.30), the following eigenvalues are obtained:

$$\begin{aligned} &\lambda_{1,2}(\boldsymbol{v}_{0,A}) = -(\frac{\alpha+8}{8} \pm j\frac{\alpha\sqrt{3}}{8})\psi, \quad \lambda_3(\boldsymbol{v}_{0,A}) = (\frac{\alpha}{4} - 1)\psi, \\ &\lambda_1(\boldsymbol{v}_{0,B}) \approx -\psi, \\ &\lambda_1(\boldsymbol{v}_{0,C}) \approx -\psi, \\ &\lambda_{1,2}(\boldsymbol{v}_{1,A}) = (\frac{\alpha-8}{8} \pm j\frac{\alpha\sqrt{3}}{8})\psi, \quad \lambda_3(\boldsymbol{v}_{1,A}) = -(\frac{\alpha}{4} + 1)\psi. \end{aligned} \tag{3.31}$$

In this case it is observed that the unstable fixed points are $\boldsymbol{v}_{0,A}$ and $\boldsymbol{v}_{1,A}$, as they are associated to eigenvalues with positive real part. Consequently, the circuit is still bistable for $x \approx 0$ V, but it is unstable for $x \approx 1$ V. Similar results are obtained for $k > 2$.

At this point we can conclude that a necessary condition for the structure represented in Fig. 3.11 to support oscillations is that the loop is composed of the two-inputs function and at least two blocks with one input.

**Example Application: Design of Ultra-Fast Oscillators in PLDs.** When we intend to design an oscillator in digital hardware, the simplest and compact solution that we can use according to the state of the art is the Ring Oscillator [31–35].

As already indicated in Subsection 2.3.1, a Ring Oscillator consists of a loop of $N$ NOT gates, where $N$ is an odd number greater than or equal to 3. This means that, in the case of a PLD project, a Ring Oscillator must be composed by at least three LUTs, each of which implements a NOT gate. Therefore, apparently three can be considered the minimum number of programmable hardware resources needed to build an oscillating circuit.

However, within a PLD the input and output pins of the LUTs are not directly connected; being the device programmable, to build the connections between logic gates, it is necessary to pass through active switch matrices. In addition to this, the output signal of a LUT, before reaching these switch matrices, passes through active digital elements that are part of the Elementary Logic Blocks described in Section 2.2.

Together with the analysis of the simplified dynamical model presented in Subsection 3.3.1, these considerations on the hardware structure of a PLD suggests that to design an oscillating circuit in a PLD is not necessary to use three or more LUTs, but it may suffices to use just one LUT, allowing the routing circuitry to take the role of the remaining stages.

To verify this assumption, we implemented the two topologies shown in Fig. 3.13 on a Xilinx Artix 7 xc7a35 FPGA, taking control of the synthesizer place and route policies at the lowest level. Given the compactness of both topologies, we expected to reach high oscillation frequencies (in the order of GHz). Since the I/O FPGA pins were designed to operate at bit rates much lower than the expected oscillation frequencies, they could not be used to extract signal out from the FPGA for direct measurements. Rather, we 1-bit sampled the oscillators locally, exploiting the registers in the Configurable Logic Blocks, adopting a sampling frequency of 100 MHz. The sampled bits were then collected in sequences of 1 million elements, which were then used for the calculation of the Decorrelation Time, already defined in Subsection 3.2.1.

In Fig. 3.14 the autocorrelation functions and the relative Decorrelation Times, evaluated on a window of $2\mu$s with $\eta$ fixed at 99.9%, are reported. From the figure we can obtain two important information:

- both systems are oscillating circuits, characterized by vanishing autocorrelation functions;
- the Decorrelation Time of the Ring Oscillator is double that of the oscillator using a single LUT.

Together with the knowledge of the used hardware, the use of the simplified dynamical model allowed to determine optimization methods for the design of DNOs (interpreting the oscillator with one LUT as a sub-element of a more complex DNO). Studying the model, we identified specific conditions such as to guarantee the oscillation of the signals and an increase in the oscillation frequency, to the benefit of the generated information entropy per second.

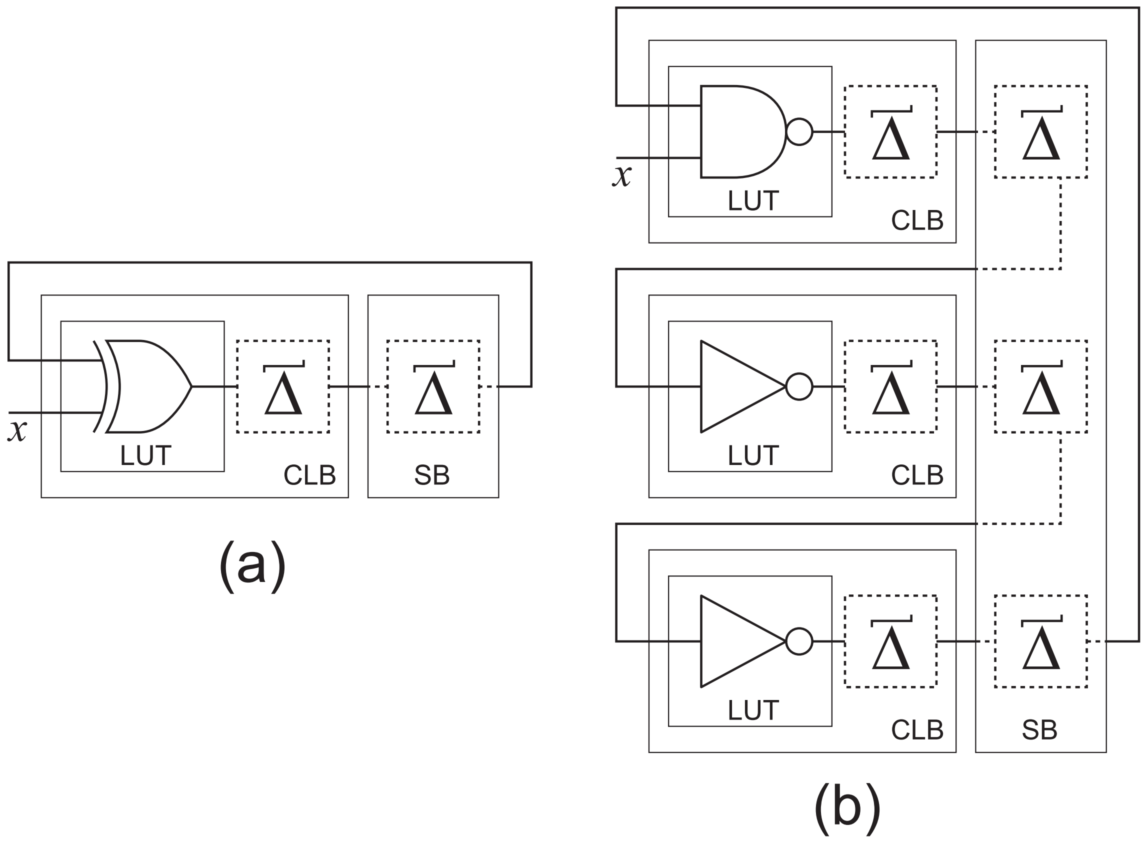


Figure 3.13: A schematic representation of the two compared systems: (a) a single LUT feedback loop that should support oscillations in FPGAs according to the study of the simplified dynamical model; (b) a three-nodes conventional Ring Oscillator. Dashed DEL nodes result from the routing/configuration multiplexers present in the Configurable Logic Block and the local Switch Matrices.

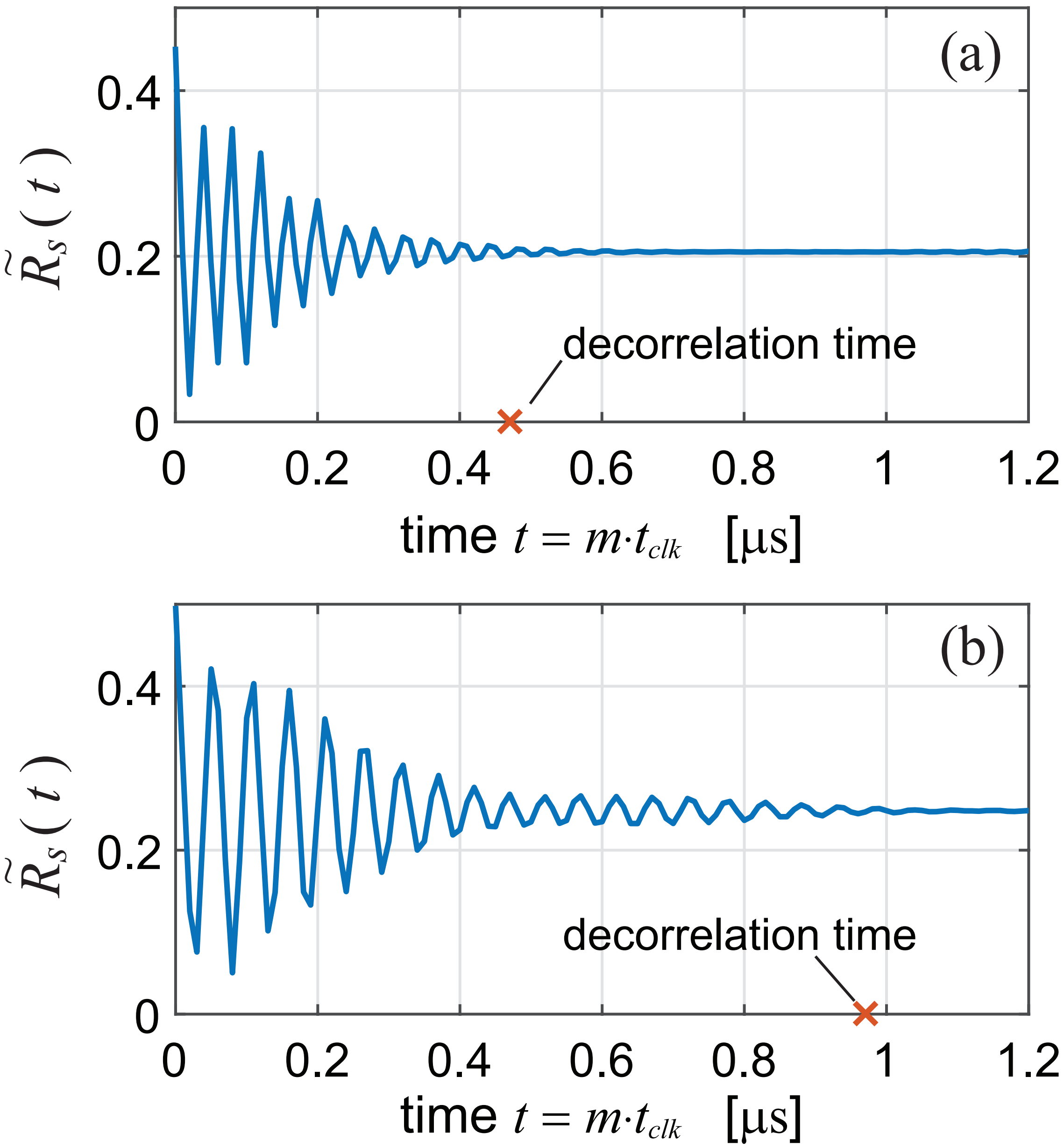


Figure 3.14: The autocorrelation functions of the binary streams collected from the oscillators shown in Fig. 3.13, performing the uniform sub-sampling of the oscillating signal, with a sampling frequency of 100 MHz and a 1-bit A/D quantization strategy.

## 3.4 Advanced Numerical Simulations

An extension of the studies performed on the simplified dynamical model relies on the use of advanced numerical simulators.

The simplified dynamical model allows to investigate the minimum necessary conditions that favor compatibility with complex dynamical behaviors for the DNO topology under consideration. However, the model does not evaluate what the actual behavior of a DNO implementation is, as it is based solely on the circuit topology and does not take into account its hardware characteristics.

The main advantage of the simplified model is the possibility to provide a complete analysis of the system, thanks to the low complexity of the resulting circuit. However, a real DNO is affected by the presence of parasitic components, which cause the resulting dynamical system to have an higher dimension with respect to the system defined through the application of the simplified model; as a consequence, the real system results, in general, too complex to be solved through direct calculations.

For this purpose, it is necessary to resort to advanced numerical simulation tools, which allow, still at a simplified level, to build the circuit based on real technologies at the transistor level. In this way, we can evaluate its dynamics taking into account not only its functional topology, but also the parasitic physical elements linked to its implementation and its operation. Specifically, our goal is to understand what could be the behavior of a certain topology in the case of its implementation on FPGA.

In our analyzes, without loss of generality, we defined a simulation setup in Cadence Virtuoso based on CMOS UMC 180 nm technology. Obviously, this setup can be implemented in any simulation environment and referring to any technology.

We built at the transistor level a library of simplified fundamental hardware elements used for the construction of an asynchronous circuit in FPGA, i.e. Look-Up Tables (LUTs) and multiplexers (MUXes) with various numbers of inputs. In Figs. 3.15 and 3.16 are shown the schematics of a 2-inputs multiplexer and a 3-inputs LUT.

The LUTs can be used to build any logic gate characterized by a number of inputs less than or equal to the maximum number supported by the used LUT. For example, in Fig. 3.17 the LUT of Fig. 3.16 is configured to implement a 3-inputs XOR function.

The MUXes instead can be used to emulate the active programmable routing elements by means of which the connections between the ports of the LUTs are built for the definition of the circuit. Fig. 3.18 shows an example of a DNO topology composed of LUTs and routing elements based on the components just described.

The built circuits represent a still simplified version of a possible real implementation, but they allow to take into account in the evaluation of the dynamical behavior of the signals aspects related to the physics of the device. It is important to underline that the purpose of these simulations is not to investigate the real behavior of the signals involved in the circuit dynamics, but to extend the anal-

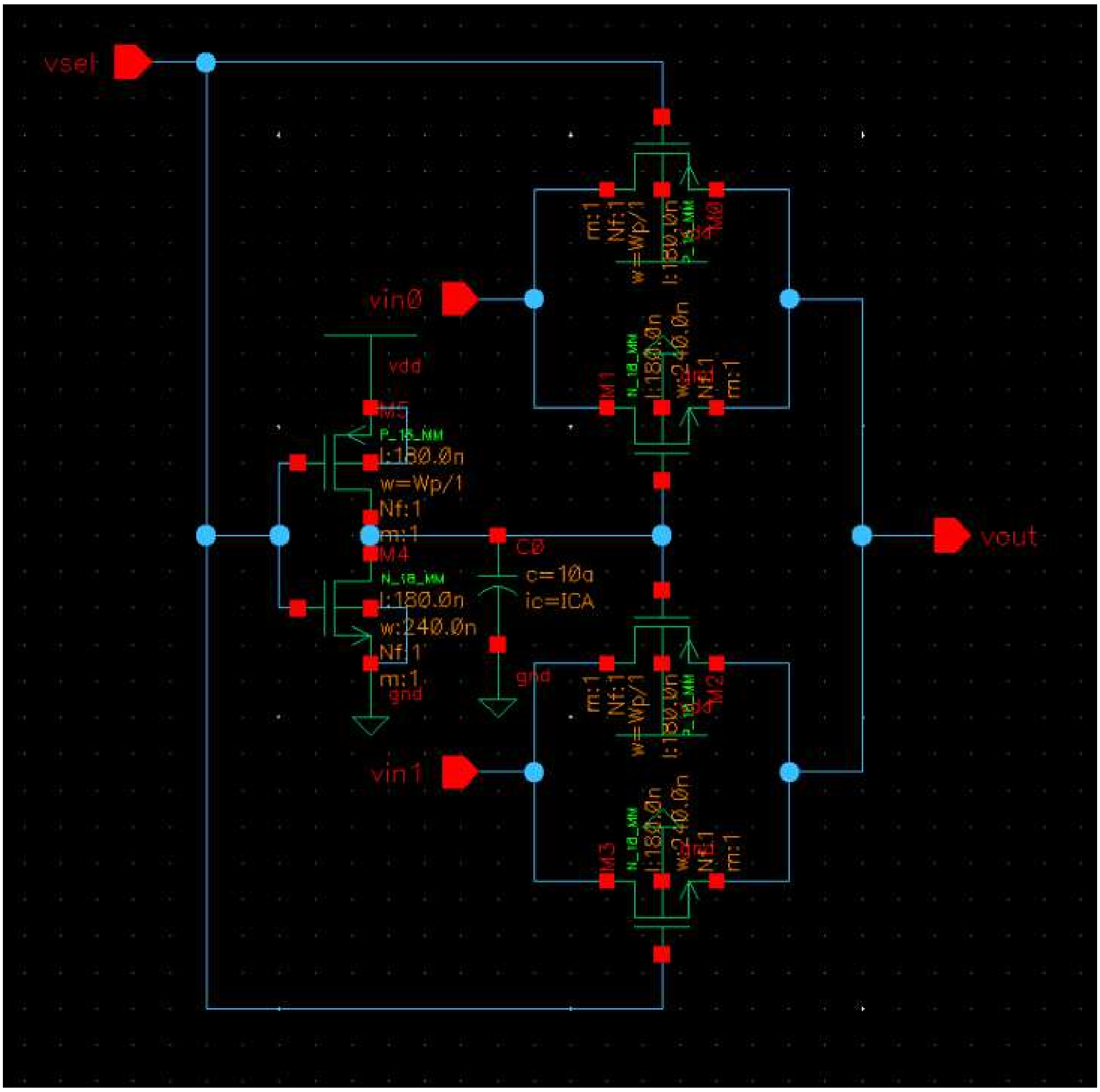


Figure 3.15: Schematic representation of a two-inputs multiplexer designed in Cadence Virtuoso at transistor level using the UMC 180 nm technology.

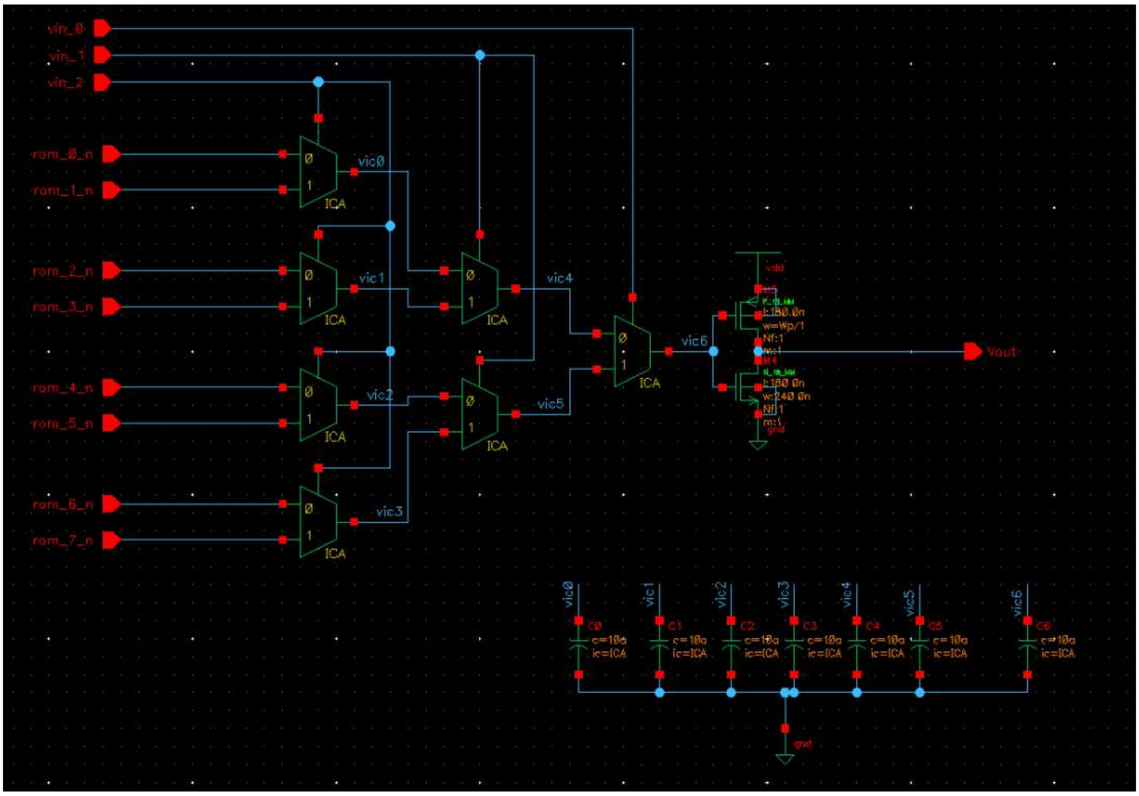


Figure 3.16: Schematic representation of a three-inputs Look-Up Table designed in Cadence Virtuoso at transistor level using the UMC 180 nm technology, making use of the two-inputs multiplexers whose schematic is shown in Fig. 3.15.

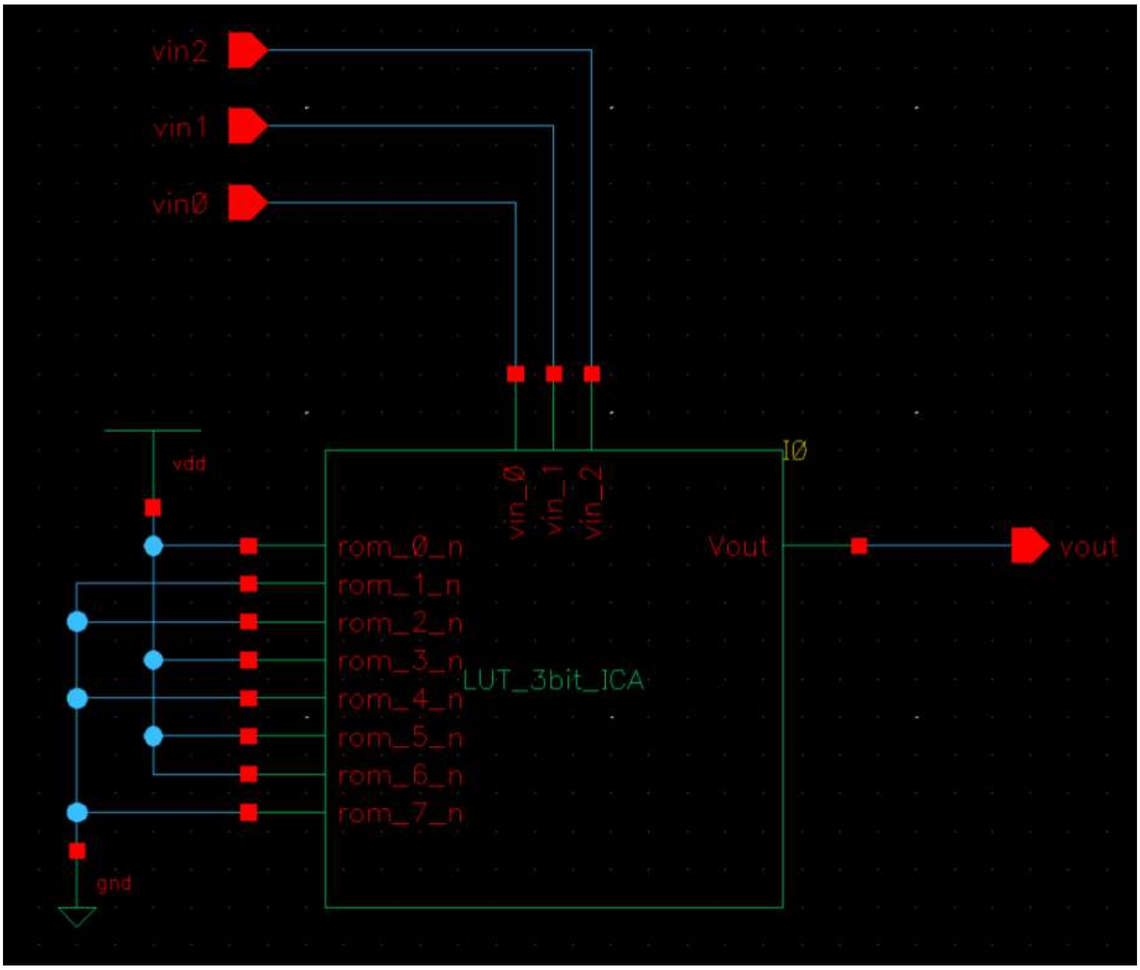


Figure 3.17: Configuration of a three-inputs Look-Up Table (whose schematic is shown in Fig. 3.16) to implement a XOR function in Cadence Virtuoso using the UMC 180 nm technology.

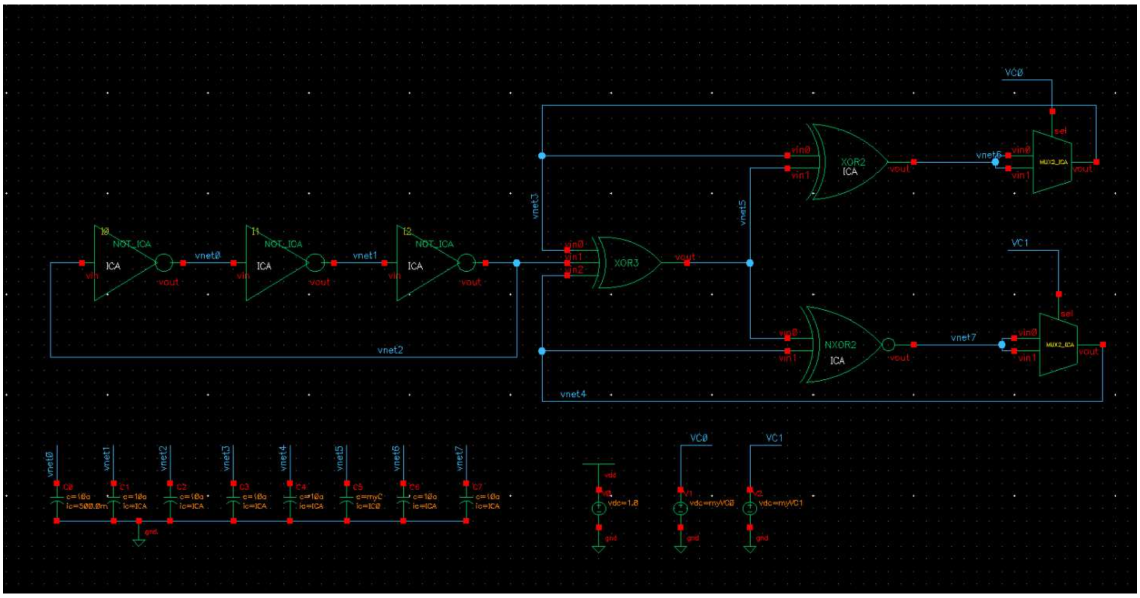


Figure 3.18: Schematic representation of a DNO topology based on the LUTs and the MUXes designed in Cadence Virtuoso at transistor level using the UMC 180 nm technology. Each logic gate is designed by configuring a three-inputs LUT as the one shown in Fig. 3.16.

ysis performed with the simplified dynamical model to higher dimension systems, exploiting the advanced numerical simulators functionality. This is the reason for which we designed our own components library, instead of using more rigorous and reliable standard models, simulating digital circuits with analog simulators.

The DNOs are analyzed by means of transient simulations, carrying out simulation campaigns based on noise injection or on Monte Carlo analysis. There are two parameters on which Monte Carlo simulations operate:

- the nonlinear DC transfer functions of the logic gates and their parasitic capacitances, parametrized by means of the form factor of the PMOS transistors, as can be observed in the schematic of the MUX in Fig. 3.15;
- the initial conditions of each node in the circuit, controlled by capacitors with negligible capacity (magnitude order of the aF) whose voltages at instant $t = 0$ of the simulation are parametrized; an example of such capacitors can be seen in Fig. 3.16.

Other analyzes include the sensitivity evaluation with respect to temperature and power supply fluctuations.

An application case of the above analyzes is presented in Chapter 4.

## 3.5 FPGA Implementation Techniques

In this last section, we provide the implementation techniques we employed to design DNOS in FPGAs.

In the course of the previous sections, various examples of DNOs were reported, showing each time in practical terms the functioning or characteristics of the addressed issues. Each of these examples included the analysis of binary sequences obtained by sampling implementations in FPGAs.

The FPGA implementation of a DNO represents the practical validation of the observations and theoretical results found starting from the simplified dynamical model and numerical simulations. For this reason, this passage cannot be carried out freely, but must respect specific design rules, independent of the considered topology.

These rules are intended to provide control over the FPGA hardware resources through the use of device primitives. The argument presented here refers to designs on Xilinx Artix 7 FPGAs made with the Vivado Design Suite in VHDL language, but its value is general, as it can be easily adapted to any FPGA device.

### 3.5.1 Combinatorial Loops

Normally, in digital design, combinatorial loops should be avoided: a combinatorial loop consists of a feedback of logic elements without registers, which can create logic race conditions or ruin the timing analysis in the synthesis and implementation phases. For this reason, design tools typically generate Design Rule Check (DRC) errors when such a loop is identified during synthesis.

A DNO topology, however, is by definition based on combinatorial loops. For this reason, to implement a DNO, it is necessary to provide the design tool with special directives to enable the synthesis of combinatorial loops required by the designer. In Vivado, it is possible to reduce the severity of the compiler blocking message, reducing the presence of a combinatorial loop from an error condition to a simple warning condition. To do this, we need to add to the project a Tcl (Tool command language) script containing the command `set_property SEVERITY {Warning} [get_drc_checks LUTLP-1]`.

### 3.5.2 Design of Elementary Logic Blocks

The logic resources in a Xilinx Artix 7 FPGA are organized in a matrix of Configurable Logic Blocks (CLBs), each containing two slices, and each slice is composed of four 6-inputs Look-Up Tables (LUTs) and eight storage items [30]. Each slice is identified by two values `X` and `Y`, which define its physical position within the FPGA. Similarly, the LUTs in a slice are identified by four letters `A`, `B`, `C`, `D`.

From the implementation point of view, in Section 2.2 we defined a DNO as a network whose nodes consist of predefined hardware structures called Elementary Logic Blocks (ELBs). The logical functionality of an ELB can be implemented

```vhdl
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
library UNISIM;
use UNISIM.VComponents.all;

entity ELB1NOT is
  port (
    A : in std_logic;
    NOT_A : out std_logic );
end ELB1NOT;

architecture Behavioral of ELB1NOT is
  attribute DONT_TOUCH : string;
  attribute KEEP_HIERARCHY : string;
  attribute BEL : string;
  attribute LOC : string;
  attribute DONT_TOUCH of Behavioral : architecture is "yes";
  attribute KEEP_HIERARCHY of Behavioral : architecture is "yes";
  attribute DONT_TOUCH of NOTGate : label is "yes";
  attribute KEEP_HIERARCHY of NOTGate : label is "yes";
  attribute BEL of NOTGate : label is "A6LUT";
  attribute LOC of NOTGate : label is "SLICE_X0Y0";
begin
  NOTGate : LUT1
    generic map (
      INIT => "01" )
    port map (
      O => NOT_A,
      I0 => A );
end Behavioral;
```

Figure 3.19: VHDL code for the low-level design of an ELB with a NOT boolean functionality. In this example, the solution uses the `6LUT` primitive, resource `A`, in the slice located at the coordinates `X0Y0`.

through a LUT. Since a DNO is an asynchronous circuit, it does not require the use of registers.

For clarity of presentation, each ELB can be associated with a VHDL entity. As shown in Figs. 3.19, 3.20, where VHDL codes are reported to describe a NOT function and a XOR function, the implementation of an ELB requires the use of special directives to force the use of specific hardware resources within the chip [40], and primitives of the device accessible through the invocation of the UNISIM library from Xilinx (in the case of the examples `LUT1` and `LUT2`). The logical operation carried out by the LUT is described through the constant `INIT`, which contains the corresponding truth table, as shown in Tabs. 3.1, 3.2.

```vhdl
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
library UNISIM;
use UNISIM.VComponents.all;

entity ELB5XOR2 is
  port (
    A : in std_logic;
    B : in std_logic;
    XOR_AB : out std_logic );
end ELB5XOR2;

architecture Behavioral of ELB5XOR2 is
  attribute DONT_TOUCH : string;
  attribute KEEP_HIERARCHY : string;
  attribute BEL : string;
  attribute LOC : string;
  attribute DONT_TOUCH of Behavioral : architecture is "yes";
  attribute KEEP_HIERARCHY of Behavioral : architecture is "yes";
  attribute DONT_TOUCH of XORGate : label is "yes";
  attribute KEEP_HIERARCHY of XORGate : label is "yes";
  attribute BEL of XORGate : label is "A6LUT";
  attribute LOC of XORGate : label is "SLICE_X1Y0";
begin
  XORGate : LUT2
    generic map (
      INIT => "0110" )
    port map (
      O => XOR_AB,
      I0 => A,
      I1 => B );
end Behavioral;
```

Figure 3.20: VHDL code for the low-level design of an ELB with a XOR boolean functionality. In this example, the solution uses the 6LUT primitive, resource A, in the slice located at the coordinates X1Y0.

| I0 | O |
|---|---|
| 0 | INIT[0] = 1 |
| 1 | INIT[1] = 0 |

Table 3.1: Truth table to implement a NOT gate by means of the LUT1 device primitive generic INIT.

| I1 | I0 | O |
|---|---|---|
| 0 | 0 | INIT[0] = 0 |
| 0 | 1 | INIT[1] = 1 |
| 1 | 0 | INIT[2] = 1 |
| 1 | 1 | INIT[3] = 0 |

Table 3.2: Truth table to implement a XOR gate by means of the LUT2 device primitive generic INIT.

### 3.5.3 Synchronization Interface

To acquire bit sequences starting from the implemented DNO, it is necessary to connect the output pin of the circuit to a synchronization interface, which can be reduced to a single D flip-flop that simultaneously performs the 1-bit analog-to-digital (A/D) conversion and the uniform sampling of the output signal.

To implement the flip-flop we use an FF device primitive, which consists of a D type flip-flop with clock enable and synchronous reset, identified by the FDRE entity accessible through the UNISIM library. Fig. 3.21 shows an example of the VHDL code through which the synchronization interface is designed.

Again, it is important to have control over the placement of the resource on the chip, so the VHDL code must also include special directives for this purpose. While in the case of ELBs a selection of the position of the component in the chip is made for reasons of dynamical characteristics of the implemented circuit, as regards the synchronization interface it is necessary to manually select its position since it participates in the timing analysis of the entire design, therefore some positions may not respect the timing constraints.

### 3.5.4 Placing and Routing

In an FPGA, routing is constructed using programmable switches and connection boxes according to a hierarchical architecture that offers local and global connectivity.

Once we arranged the ELBs in the desired positions, the connection between the pins takes place according to policies left to the compiler. To minimize the impact related to routing, it is advisable to concentrate the ELBs in a few slices, placing them next to each other.

## 3.6 Conclusion

We introduced a set of tools that define a methodology for the analysis and design of Digital Nonlinear Oscillators; these tools have the purpose of evaluating a Digital Nonlinear Oscillator from different points of view, considering its characteristics as an analog electronic circuit, as a complex dynamical system and as a source of entropy.

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
library UNISIM;
use UNISIM.VComponents.all;

entity SYNC_INT is
  port (
    ANALOG_IN : in std_logic;
    CLK : in std_logic;
    RST : in std_logic;
    RND_OUT : out std_logic );
end SYNC_INT;

architecture Behavioral of SYNC_INT is
  attribute DONT_TOUCH : string;
  attribute KEEP_HIERARCHY : string;
  attribute BEL : string;
  attribute LOC : string;
  attribute DONT_TOUCH of Behavioral: architecture is "yes";
  attribute KEEP_HIERARCHY of Behavioral: architecture is "yes";
  attribute DONT_TOUCH of BitRegister : label is "yes";
  attribute KEEP_HIERARCHY of BitRegister : label is "yes";
  attribute BEL of BitRegister : label is "DFF";
  attribute LOC of BitRegister : label is "SLICE_X1Y0";
begin
  BitRegister : FDRE
    generic map (
      INIT => '0' )
    port map (
      Q => RND_OUT,
      C => CLK,
      CE => '1',
      R => RST,
      D => ANALOG_IN );
end Behavioral;
```

Figure 3.21: VHDL code for the low-level design of the D flip-flop used to perform both 1-bit A/D conversion and uniform sampling of the output signal provided by a DNO. In this example, the solution uses the `FF` primitive, resource `D`, in the slice located at the coordinates `X1Y0`.

We defined two figures of merit (Decorrelation Time and Average Shannon Entropy) which allows to evaluate, in the comparison between two or more sources, which one is capable of offering the best performance in terms of generated information. The Decorrelation Time establishes what is the minimum sampling period capable of guaranteeing decorrelation between consecutive symbols generated by a DNO; the Average Shannon Entropy offers an estimate of the entropy generated by the circuit.

An application example for these figures of merit was provided: we compared three different DNO topologies (Ring Oscillator, Galois Ring Oscillator and a custom topology) implementing them in 5 Xilinx Artix 7 FPGAs, acquiring sequences of bit sampled at different frequencies, and evaluating their Decorrelation Times and Average Shannon Entropies. We observed that the three DNOs are characterized by different performance. We found a link between Decorrelation Time and Average Shannon Entropy, as, on average, the shorter is the Decorrelation Time the higher is the ASE. The analyzes also highlighted that Average Shannon Entropy and sampling frequency are linked by a nonlinear relationship, for which a variation in the sampling frequency involves a marginal variation of the ASE. In this sense, the frequency at which we sample our source has a significant weight in terms of the rate of generated information.

We introduced a simplified dynamical model for the description of a DNO having the purpose to investigate the minimum necessary conditions that favor its compatibility with complex dynamical behaviors, on the basis of the stability of its fixed points. In the model, each node of the DNO is represented with a first order cell, given by a voltage controlled voltage generator that controls a resistance-capacitance cell of the first order.

The model was used to study the stability of a circuit primitive that is often used within complete DNO topologies, evaluating the minimum complexity at the dynamical system level that this primitive must possess in order to oscillate. The results obtained through the analysis based on the simplified model were then verified by implementation on FPGA: we showed experimentally that it is possible to design an oscillating sub-circuit composed by a single Look-Up Table (LUT) feedback loop. By comparing the designed subcircuit with a three-nodes Ring Oscillator, we noticed that with a topology of this kind it is possible to reach dynamical speeds higher than the DNO with minimum complexity that can be designed at a logical level.

We showed the simulation setup built in Cadence Virtuoso in order to deepen the dynamical behavior of the signals involved in a DNO. This setup makes use of UMC 180 nm technology to replicate at transistor level, in a simplified form, the fundamental hardware structures of an FPGA used in the design of a DNO, i.e. LUTs for the design of logic gates and MUXes for the emulation of the active routing elements. These structures are used to design the circuits to be analyzed, which are then subjected to different types of simulations, such as noise injection and Monte Carlo analysis.

Finally, we explained the design rules that must be applied when implementing

DNOs on FPGAs, showing the syntax in VHDL language for their use. These rules have the purpose of:

- allowing the synthesis of combinatorial loops, normally not allowed as they can create logic race conditions or ruin the timing analysis;
- using specific low-level resources, such as Look-Up Tables and Flip-Flops, which must be explicitly selected by indicating their position within the chip;
- defining the overall layout of the circuit to partially control the routing between the output and input pins of the designed logic gates.

# Chapter 4

# High Performance DNO

In this chapter we use the analysis and design methodologies of Digital Nonlinear Oscillators formalized in Chapter 3 to describe the complete workflow followed for the design of a novel DNO topology. This DNO is characterized by chaotic dynamical behaviors, and is capable of achieving high performance in terms of generated entropy, downstream of a reduced hardware complexity and high sampling frequencies. By exploiting the simplified dynamical model, the advanced numerical simulations in Cadence Virtuoso and the FPGA implementation, the presented topology is extensively analyzed both from a theoretical point of view (notable circuit sub-elements that make up the topology, bifurcation diagrams, internal periodicities) and from an experimental point of view (generated entropy, source autocorrelation, sensitivity to routing, temperature sensitivity, application of standard statistical tests).

## 4.1 Chaos in Fully Digital Hardware

In Chapter 3 we defined the tools needed to analyze the characteristics and performance of a DNO. In the course of this chapter, a practical example of how these tools could be used in order to design a DNO from scratch is shown.

Starting from a theoretical analysis of the dynamics of the proposed topology and subsequently evaluating the characteristics resulting from its implementation, a DNO is proposed characterized by complex dynamical behaviors (including chaos), capable of achieving high performance in terms of generated entropy, downstream of a reduced hardware complexity and high sampling frequencies [41–43].

### 4.1.1 References

The material presented in this chapter includes results that have been published in the following publications:

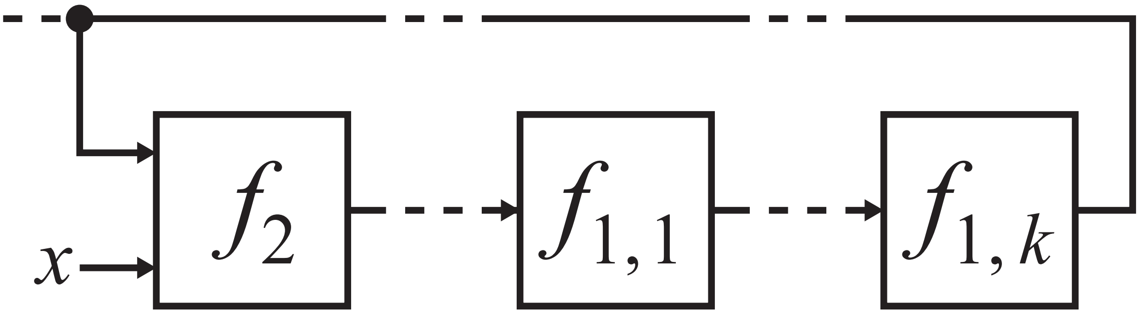


Figure 4.1: The sub-network analyzed in Subsection 3.3.1, consisting of a feedback loop composed of a gate with two inputs (one of feedback and the other independent) and a cascade of gates with one input.

- T. Addabbo, A. Fort, R. Moretti, M. Mugnaini, H. Takaloo, and V. Vignoli, “Chaos in fully digital circuits: A novel approach to the design of entropy sources,” in *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2020, pp. 1-5 [41];
- T. Addabbo, A. Fort, R. Moretti, M. Mugnaini, H. Takaloo, and V. Vignoli, “A new class of chaotic sources in programmable logic devices,” in *2020 IEEE International Workshop on Metrology for Industry 4.0 & IoT*. IEEE, 2020, pp. 6-10 [42];
- T. Addabbo, A. Fort, R. Moretti, M. Mugnaini, H. Takaloo, and V. Vignoli, “A new class of digital circuits for the design of entropy sources in programmable logic,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 67, no. 7, pp. 2419-2430, 2020 [43].

## 4.2 Topology

The first step in designing a DNO is choosing the topology.

In Subsection 3.3.1 we analyzed the dynamical characteristics of a sub-circuit consisting of a feedback loop composed of a gate with two inputs (one of feedback and the other independent) and a cascade of gates with one input, which we report in Fig. 4.1.

By selecting a XOR logic function for the two-inputs gate and a digital delay (DEL) for the one-input gates, we observed that such subcircuit supports oscillations, provided the one-input gates cascade is composed by at least two elements and that the independent input of the XOR is fixed at a logic 1, while with a logic 0 it assumes a bistable behavior.

Repeating the calculations by replacing the XOR with a NXOR, a complementary result is obtained (we do not report the calculations as the procedure is equivalent to the one already presented in Subsection 3.3.1): the sub-circuit supports oscillations provided it has a cascade of at least two DEL gates and that the independent input of the NXOR is set at a logic 0, and is bistable when the independent input is forced to a logic 1.

Assuming to connect two configurations of these types together by short circuiting the independent inputs, and to inject a periodic digital signal in them (e.g. the output of a Ring Oscillator), we expect therefore to see the two sub-circuits oscillate at alternate moments, with the sub-circuit that is not oscillating that maintains the last reached logic state.

Combining the injected periodic digital signal with the output signals of the two loops and acting on the periods of these three signals, it is reasonable to think that this process of switching the oscillations on and off gives rise to complex dynamics.

This assumption arises from the fact that the dynamics of coupled oscillators is studied since centuries, starting from the well known synchronization of weakly coupled mechanical pendulums. This phenomenon is known as phase-locking, and is generally present in dissipative systems with competing frequencies. Depending on both the system parameters and the coupling strength, different kind of dynamics can be observed, ranging from periodic-locked, quasi-periodic (i.e., the ratio between the two oscillator frequencies is irrational) and chaotic. To have a chaotic dynamics, a fundamental role is played by the nonlinear nature of both the oscillators and the coupling between them [44–48].

A special case of coupled oscillators is obtained when an autonomous dynamical system $\mathbf{x}$ is used to generate a driving signal exciting a second dynamical system $\mathbf{y}$. In this situation, referring to a wide theoretical framework, the overall system can be described by the generic system of nonlinear differential equations

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) \\ \dot{\mathbf{y}} = \mathbf{g}(\mathbf{x}, \mathbf{y}) \end{cases} , \tag{4.1}$$

being $\mathbf{x} : \mathbb{R} \to \mathbb{R}^N, \mathbf{y} : \mathbb{R} \to \mathbb{R}^M$ real-valued functions of time $t$, and $\mathbf{f}, \mathbf{g}$ nonlinear smooth real-valued functions of $\mathbf{x}$ and $\mathbf{y}$, respectively. If $\dot{x} = \mathbf{f}(\mathbf{x})$ and $\dot{\mathbf{y}} = \mathbf{g}(\mathbf{0}, \mathbf{y})$ define two periodic dynamical systems, we may call $\mathbf{y}$ in (4.1) the forced oscillator, being $\mathbf{x}$ the forcing periodic driver.

According to the just presented considerations, we elaborated the topology shown in Fig. 4.2, where the ELBs#[1-3] have the task of generating the periodic injected signal, while the loops composed respectively of ELBs#[5-7] and ELBs#[8-10] are the previously analyzed sub-circuits. ELB#4 is the combination element of the output signals of the three loops and ELB#11 plays the role of a 1-bit quantization A/D converter of the circuit output signal.

## 4.3 Dynamical Analysis

Having established the topology, the next step is to investigate if conditions that favor compatibility of the topology with complex dynamics exist. To do this, we have to resort to the simplified dynamical model described in Section 3.3.

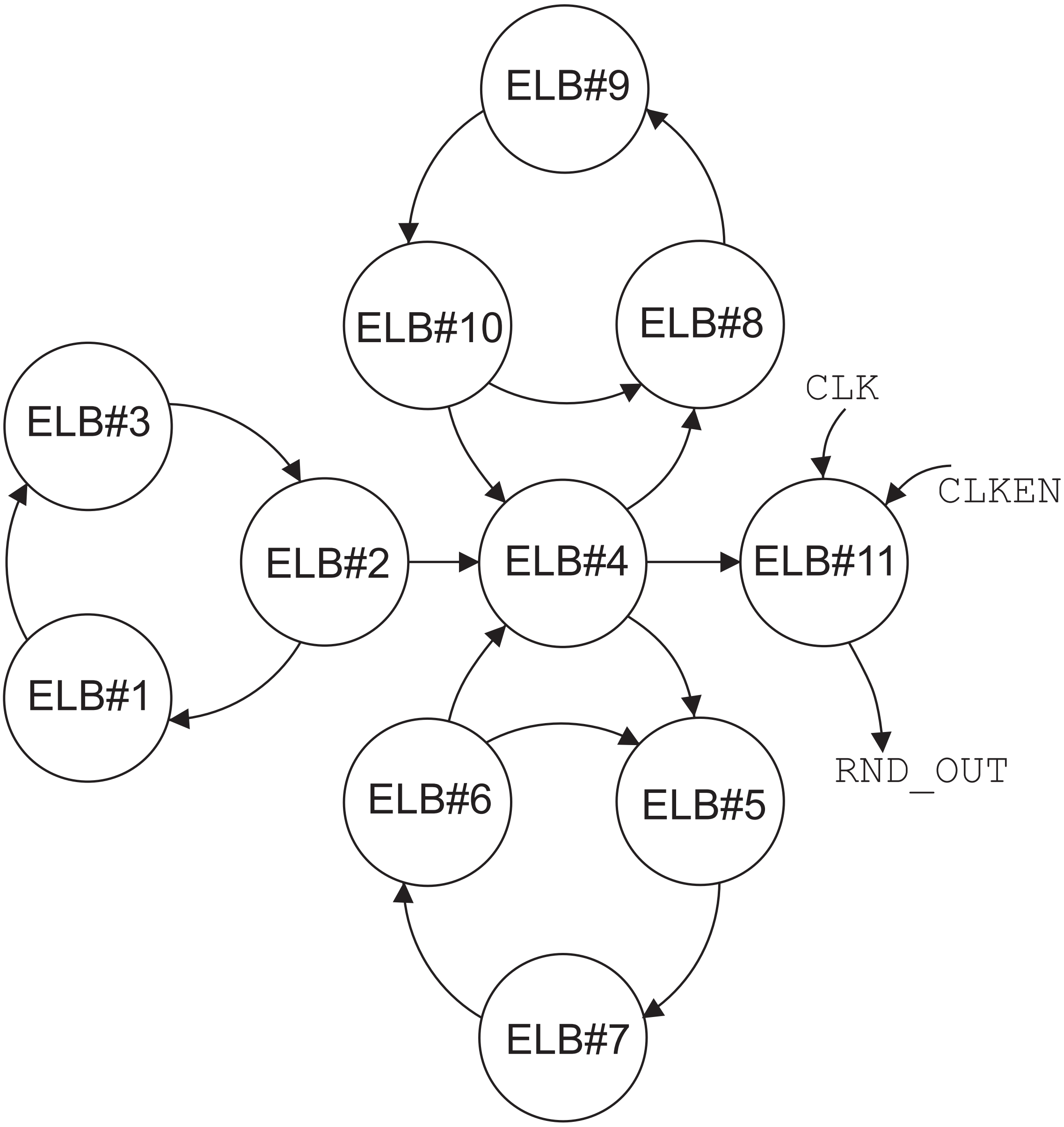


Figure 4.2: The analyzed DNO topology, characterized by an independent loop generating an excitation signal, controlling the dynamics of the sub-circuit loops having the structure analyzed in Subsection 3.3.1.

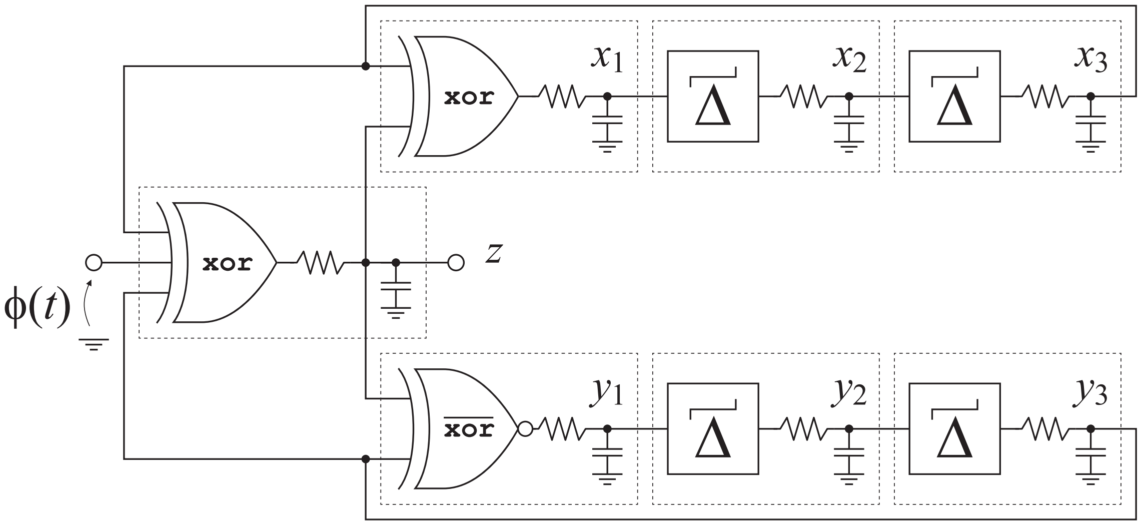


Figure 4.3: A simplified model to investigate the core DNO sub-network implementing the non-autonomous dynamical system (4.2).

Observing the topology in Fig. 4.2, we note that it can be divided in two parts: the ELBs#[1-3] constitute a structure that is totally independent from the rest, having the purpose of exciting the remaining part of the network. Therefore, at a first approximation it is possible to decide to ignore the subnetwork given by the ELBs#[1-3], replacing it with an excitation wave generator.

Applying this simplification and imposing that the loop given by the ELBs#[5-7] is the sub-circuit composed by one NXOR and two DELs, the loop given by the ELBs#[8-10] is the sub-circuit composed by one XOR and two DELs, and the ELB#4 combines the output signals of the two loops with the excitation signal by means of a XOR operation, the resulting simplified dynamical model looks as shown in Fig. 4.3.

This circuit defines a non-autonomous nonlinear dynamical system, which operates in a normalized phase space defined on the domain $[0,1]^7 \subset \mathbb{R}^7$, that is:

$$\begin{cases} \frac{dx_1}{dt} = \alpha_1[\text{XOR2}(x_3, z) - x_1] \\ \frac{dx_2}{dt} = \alpha_2[\text{DEL}(x_1) - x_2] \\ \frac{dx_3}{dt} = \alpha_3[\text{DEL}(x_2) - x_3] \\ \frac{dy_1}{dt} = \beta_1[\text{NXOR2}(y_3, z) - y_1] \\ \frac{dy_2}{dt} = \beta_2[\text{DEL}(y_1) - y_2] \\ \frac{dy_3}{dt} = \beta_3[\text{DEL}(y_2) - y_3] \\ \frac{dz}{dt} = \gamma[\text{XOR3}(x_3, \phi(t), y_3) - z] \end{cases} . \tag{4.2}$$

$\alpha_i, \beta_i, \gamma \in \mathbb{R}^+, i = 1, 2, 3$ are positive parametric constants that describe the reciprocals of the *RC* time constants associated with each node of the circuit, $\phi : \mathbb{R} \to [0,1]$ is the arbitrary signal of excitation, DEL : $\mathbb{R} \to [0,1]$, XOR2 : $\mathbb{R}^2 \to [0,1]$, NXOR2 : $\mathbb{R}^2 \to [0,1]$, XOR3 : $\mathbb{R}^3 \to [0,1]$ are the functions that fit the DC analog transfer functions of the respective logic gates, defined as analytic combinations of the sigmoids of the form:

$$\sigma(x, a, b) = \frac{1}{1 + e^{a(x-b)}}, \tag{4.3}$$

where $a, b \in \mathbb{R}$.

We limit $a$ and $b$ on the intervals $a > 20$ and $0.3 < b < 0.7$, and define, for $i \in \mathbb{N}$, the fundamental transfer functions of rectification and inversion as follows:

$$x_i(v_i) = x_i = \frac{1}{1 + e^{-a(v_i-b)}}, \tag{4.4}$$

$$\overline{x}_i(v_i) = \overline{x}_i = \frac{1}{1 + e^{a(v_i-b)}}. \tag{4.5}$$

(4.4-4.5) can be used to express the transfer functions indicated above:

$$\begin{aligned}
\mathrm{DEL}(v_i) &= x_i, \\
\mathrm{XOR2}(v_i, v_j) &= x_i\overline{x}_j + \overline{x}_i x_j, \\
\mathrm{NXOR2}(v_i, v_j) &= x_i x_j + \overline{x}_i\overline{x}_j, \\
\mathrm{XOR3}(v_i, v_j, v_k) &= (x_i x_j + \overline{x}_i\overline{x}_j)x_k + (x_i\overline{x}_j + \overline{x}_i x_j)\overline{x}_k.
\end{aligned} \tag{4.6}$$

### 4.3.1 System Analysis: Turned-Off Excitation

If the excitation of the circuit in Fig. 4.3 is turned off ($\phi(t) = 0$), the dynamical behavior of the system (4.2) depends on the parameters $\alpha_i$, $\beta_i$, $\gamma$, $a$, $b$, $i = 1, 2, 3$.

Assuming that the parameters assume non-pathological values, i.e. that $\alpha_i$, $\beta_i$ and $\gamma$ are defined on similar order of magnitudes, the resulting autonomous system has a stable and globally attractive limit cycle. In other words, the autonomous simplified circuit obtained by switching off $\phi(t)$ belongs to the DNO family.

Fig. 4.4 shows the results of exhaustive simulations of the system, obtained by integrating (4.2) with standard numerical methods. The figure shows how the system is stable to parametric perturbations (Fig. 4.4.a) and that the output signal $z$ is characterized by regular oscillations (Fig. 4.4.b).

### 4.3.2 System Analysis: Periodic Excitation

If the excitation of the circuit in Fig. 4.3 is turned on, the system (4.2) describes a forced nonlinear oscillator [45–48].

To carry out an analysis that includes the parametric space, taking into account the problem dimensions, it is not possible to adopt an analytical approach with respect to the system (4.2), therefore it is necessary to resort to numerical investigation methods.

To perform these analyzes, we reduced the complexity of the problem, assuming $\alpha_i = \beta_i = \gamma = \xi > 0$ in (4.2) and $a = 30$ and $b = 0.5$ in (4.4). For the excitation signal, we used an adapted full-scale sinusoidal signal with frequency $f_0 = 1/T_0$:

$$\phi(t) = \frac{1}{2}(1 + \sin(2\pi f_0 t)). \tag{4.7}$$