EXPERIMENTAL CHARACTERIZATION OF RANDOM TELEGRAPH NOISE IN FDSOI TECHNOLOGY AND ITS APPLICATION FOR SECURITY PRIMITIVES

By

MARÇAL OLIVÉ I MUÑIZ

A Master thesis submitted in partial fulfillment of
the requirements for the master in

MEE: MASTERS IN ELECTRONICS ENGINEERING

UNIVERSITAT POLITÈCNICA DE CATALUNYA
Departament d'enginyeria electrònica

JUNE 2022

Revision history and aproval record

Revision	Date	Purpose
0	21/6/2022	Document creation
1	26/6/2022	Document Revision
2	30/6/2022	Document Revision
3	1/7/2022	Document Revision

Document Distribution list:

Name	e-mail
Marçal Olivé	marcal.olive.i@estudiantat.upc.edu
Diego Mateo	diego.mateo@upc.edu
Xavier Aragones	xavier.aragones@upc.edu

Written by:		Reviewed and approved by:	
Date	30/6/2022	Date	4/7/2022
Name	Marçal Olivé	Name	Diego Mateo
Position	Project Author	Position	Project supervisor

EXPERIMENTAL CHARACTERIZATION OF RANDOM TELEGRAPH NOISE

IN FDSOI TECHNOLOGY AND ITS APPLICATION FOR SECURITY PRIMITIVES

Abstract

by Marçal Olivé i Muñiz,
Universitat Politècnica de Catalunya
June 2022

: Xavier Aragonès i Diego Mateo

The shrinking of transistors and consequent decrease in operational voltage, specially for Ultra-Low Voltage (ULV) applications such as IoT, has driven current technology to be very sensitive to the effects of random telegraph noise (RTN), the result of trapping and detrapping of carriers in a transistor's oxide trap. Such a source of noise is attractive for the implementation of security primitives due to its resilience to temperature and supply voltage variations. However, it stands out from other noise sources for its low speed. The use of FDSOI technology, through the ability of controlling body bias, can be the key to optimize RTN speed for such applications.

In this project, an experimental characterization of RTN behavior in an FDSOI ROSC-based chip has been conducted to evaluate the potential application of such a technology in security primitives.

Results show that FDSOI technology can be employed to significantly increase or decrease RTN speed and even compensate for the effect of supply voltage and temperature variations.

UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

telecom BCN

TABLE OF CONTENTS

Page

REVISION HISTORY AND APPROVAL RECORD ii

ABSTRACT iii

LIST OF TABLES vi

LIST OF FIGURES vii

CHAPTER

1 Introduction 1

1.1 Gantt Diagram 2

2 State of the Art 4

2.1 RTN in bulk CMOS and FDSOI 4

2.2 RTN in digital circuits 6

2.3 Security primitives based in RTN 8

2.3.1 True Random Number Generators 8

2.3.2 Physical Unclonable Functions 10

3 Methodology for Experimental Analysis of RTN in FDSOI-based ROSCs 12

3.1 The chip under study 12

3.2 Measurement setup 15

3.3 RTN detection and data processing 18

4 Experimental Results 22

4.1 Frequency analysis 22

4.2 RTN characterisation at nominal conditions 23

4.3 Body bias variation effect 25

4.4 Temperature effect 29
4.5 Supply voltage effect 30

5 Potential of RTN in FDSOI for TRNG and PUF implementation . . 32
5.1 RTN in FDSOI for TRNG applications 32
5.1.1 Sequence randomness 32
5.1.2 TRNG performance evaluation 37
5.2 RTN in FDSOI for PUF applications 41

6 Conclusions and future work 43

APPENDIX

REFERENCES 46

GLOSSARY 47

LIST OF TABLES

3.1 PB vs length table for FDSOI-28nm NAND gate 13

3.2 Studied chip measurement and communication setup 16

4.1 Statistics for detected ROSCs presenting RTN over 3 chips 23

4.2 Event rate difference due to bias variation under nominal conditions 26

5.1 NIST test results of $t_{c,e} = te + tc$ count sequences for different ROSC in chip #1 36

5.2 Experimentally obtained technology specs 37

5.3 Calculated TRNG specs 39

5.4 Transistor-based time constant evaluation TRNG specs 39

5.5 Calculated TRNG specs for filtering of devices with fastest RTN 40

LIST OF FIGURES

1.1 Typical RTN-induced waveform 1

1.2 Project's Gantt diagram 3

2.1 Carrier trap diagram 4

2.2 RTN-induced Ids variation 5

2.3 t_c, t_e, τ_c and τ_e definition 6

2.4 τ_c and τ_e for different gate and substrate bias 7

2.5 Time constants and occupancy probability as a function of gate voltage. . . . 7

2.6 Switching frequency effect on RTN 8

2.7 RTN-based TRNG through Ids monitoring 9

2.8 Edge to pulse circuit for RTN detection 10

2.9 RTN-PUF block diagram example 11

3.1 Studied chip core 12

3.2 Studied chip ring oscillator 13

3.3 Used FDSOI technology structure 13

3.4 Full ROSC array 14

3.5 Dual ROSC array for frequency comparison with odometer 15

3.6 Studied chip and measurement board 16

3.7 Measurement mode diagram 17

3.8 RTN detection and time constant extraction process 19

3.9 Algorithm adjusting parameters 20

3.10 Extracted data for an obtained sequence 20

3.11 Automatic filtering results for RTN in ROSC identification for chip #1 . . . 21

4.1 Chip #2 frequency analysis 22

4.2 Found ROSC percentage for the 3 studied chips 23

4.3 Found interference 24

4.4 Time constant distribution for detected ROSCs with RTN under nominal conditions for chip #1 24

4.5 Jump due to RTN value distribution 25

4.6 RTN jump to jitter standard deviation relationship 25

4.7 Detected RTN events as a function of body biases for different ROSC 26

4.8 Body bias corner sequences for ROSC in Col=43, Row=42 in chip #1 27

4.9 Time constants variation due to body bias for two different ROSC in chip #1 28

4.10 Temperature effect on RTN event rate over 16.5ºC difference 29

4.11 Effect of a supply voltage change on RTN event rate for two ROSC on chip #1 30

5.1 Markov chain results for digitised RTN sequence 33

5.2 NIST test suite environment 34

5.3 Debiased $t_{c,e}$ sequence 35

Dedication

I want to express my thanks to my family and friends who supported me throughout this project's development and the rest of my studies. This thesis is dedicated to them.

Special thanks to both of my tutors, Diego Mateo and Xavier Aragones as well as Enrique Barajas, the creator of the studied hardware, for all their guidance in the making of this project.

1. Introduction

The progressive shrinking of transistor sizes and consequent power reduction has led transistor current signals to be comparable to Random Telegraph Noise (RTN), which is generated through trapping and detrapping of a carrier in the gate oxide of a transistor. When a carrier is captured in a trap, an effective transistor threshold voltage variation occurs. RTN is observed as a random fluctuation between discrete level changes in the transistor drain current (Ids). An example of a waveform of Ids through time for a device presenting a 2-state RTN is shown in Fig. 1.1. On top of decreasing the signal to noise ratio, in the case of digital circuits, the presence of RTN can affect performance by modifying delays, resulting in an effective jitter increase, or even generate SRAM cell stability issues.

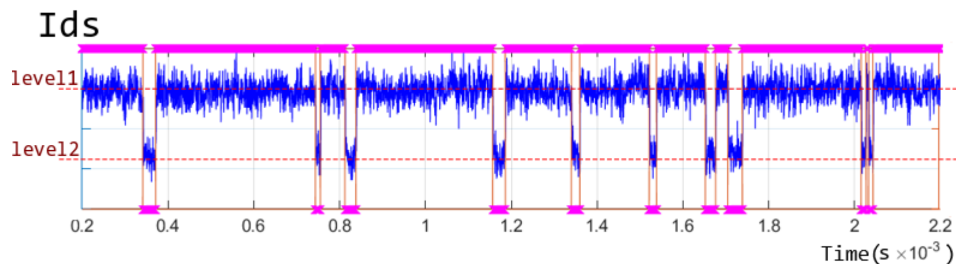


Fig. 1.1. Typical RTN-induced waveform.

Fully-Depleted-Silicon-On-Insulator (FDSOI) technology, through its ability of controlling transistor body bias, can be employed to reduce the effect of RTN. At the same time, it can also be the key to magnifying it with the objective of harvesting the phenomena as a source of randomness for security primitives. The interest in using RTN as a randomness source stems from its reported resilience to temperature and supply voltage variations when compared with other sources such as thermal jitter. However, RTN is notoriously slow, so in order to be able to compete with other sources, a method for increasing its speed needs to be implemented. FDSOI technology shows promise in achieving just that.

The objective of this project has been to perform an experimental characterization of the effect of RTN on FDSOI-based ring oscillators (ROSC) in order to evaluate its behavior. Another objective has been to interpret the obtained results to determine the potential for application of such a technology in the creation of true random number generators (TRNG) and physical unclonable functions (PUF). To do so, the project starts from an already designed FDSOI-based ROSC chip and measurement setup, as well as measurement scripts and a base RTN detection script which was further expanded throughout the project. The studied chip consists on an array of ring oscillators from which their signal can be selected

and driven to its output for analysis. RTN can then be detected from the observed changes in ROSC period over time.

This work as well as the chip under study is part of the project "Novel IC Design Paradigms for Mitigation and Exploitaton" (VIGILANT) which is financed by the Ministerio de Ciencia, Innovacion y Universidades.

1.1 Gantt Diagram

In the developement of this project, the first step has been to study the state of the art as well the chip under test and pre-existing measurement and detection MATLAB scripts.

Following that, the first step was to identify which Oscillators presented RTN and once identified, several sequences were extracted for those identified ROSCs under different biasing, supply and environmental conditions. A total of 5 chips have been evaluated. ROSC sequences have been extracted from chip #1, the probability for a ROSC to present RTN has been evaluated with chips #1 to #3 and all 5 chips were measured to obtain ROSC frequency distribution.

Since the measurements could take several days up to a week to complete depending on the data o be retrieved, post-processing was done simultaneously with measurements. As data was retrieved, it was analysed. All post-processing was performed through MATLAB version R2021b and to evaluate the randomness of extracted random sequences, the NIST test suite was employed [8].

Finally, all data was compiled and the final report was written.

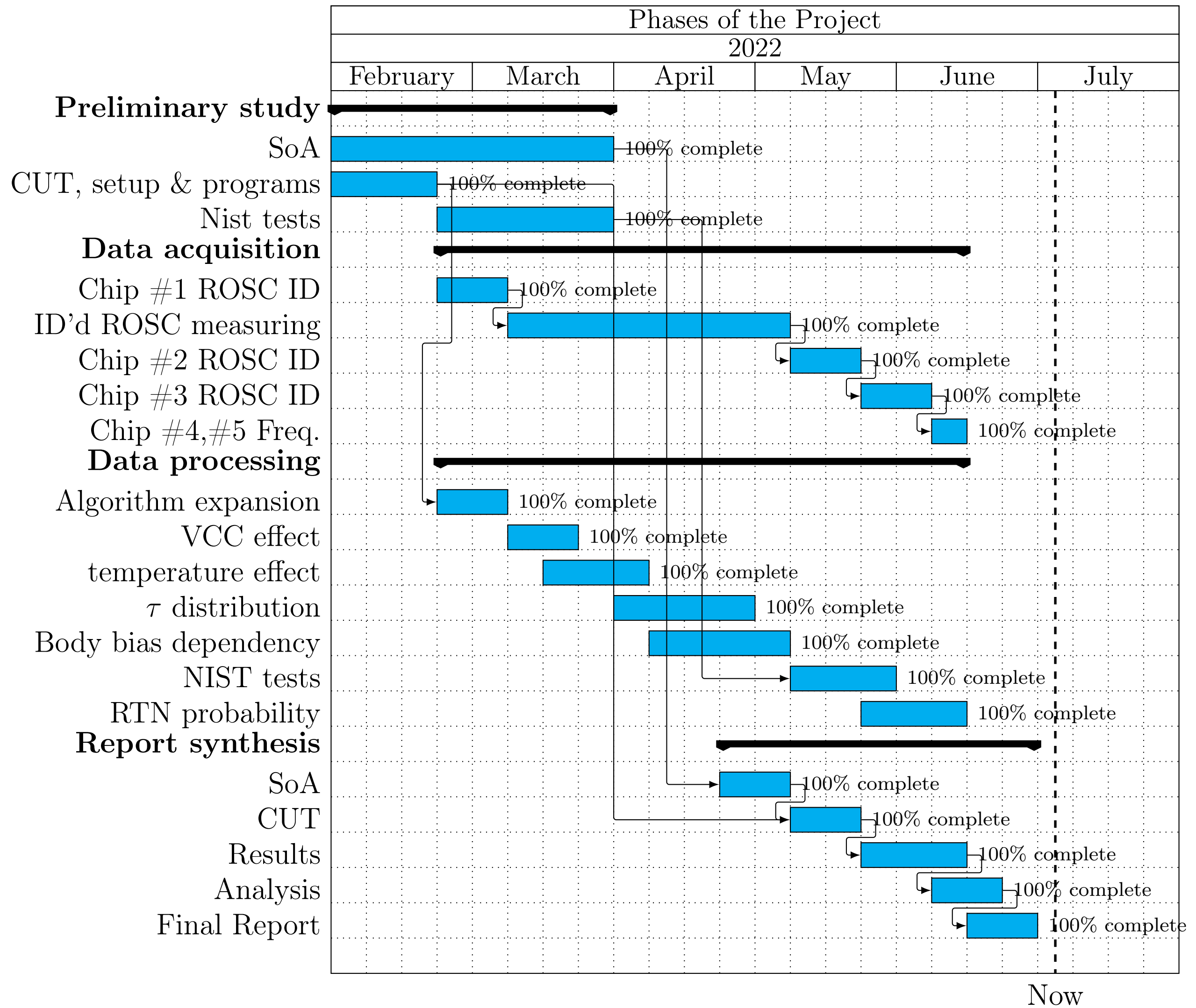
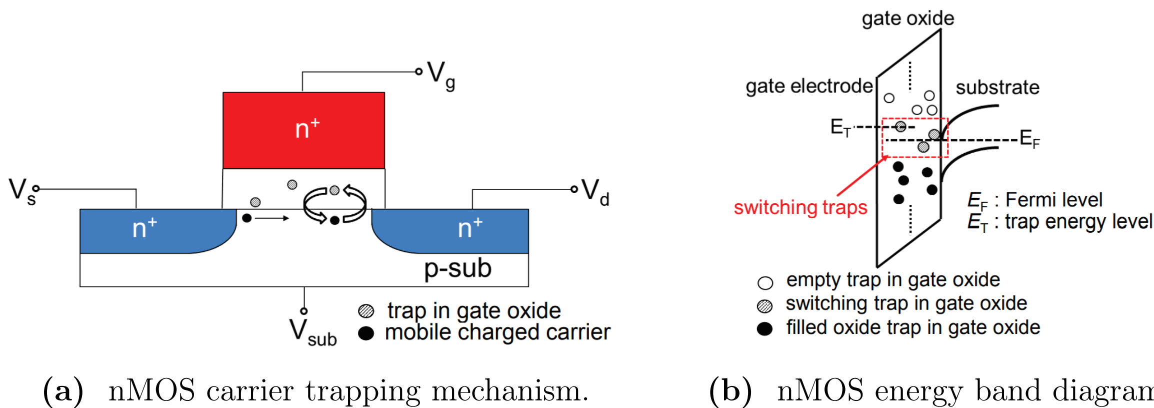


Fig. 1.2. Gantt diagram of the project

2. State of the Art

2.1 RTN in bulk CMOS and FDSOI

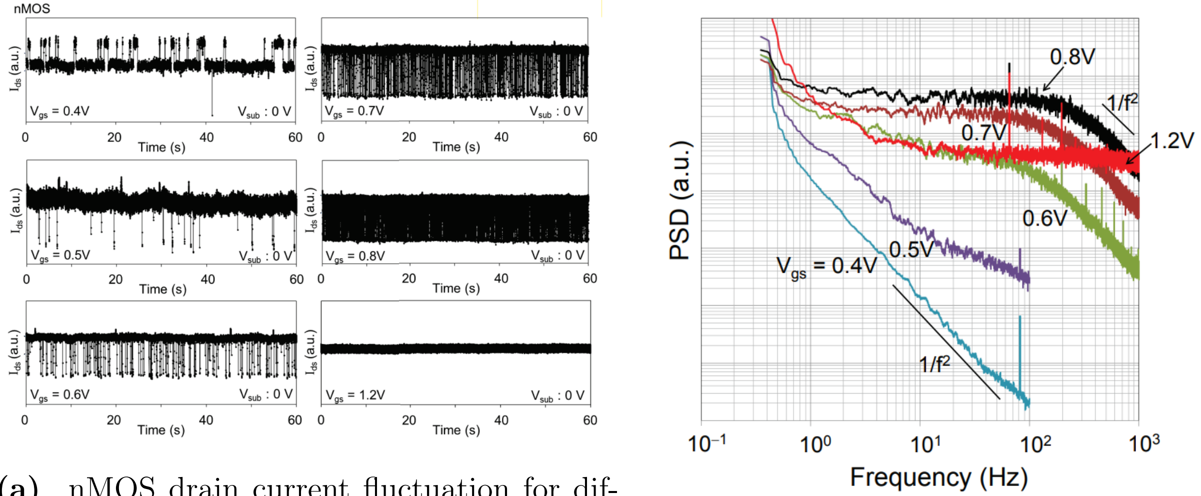
Random Telegraph Noise (RTN) is an intrinsic device noise that produces a fluctuation in performance of a transistor. It is a purely random process caused by the trapping and releasing of mobile charged carriers from the channel in the transistor's gate oxide layer (Fig. 2.1a). This results in a discrete change of effective threshold voltage ΔVth which changes the gate and substrate bias to drain current (Ids) relationship of the transistor. A single trap produces a two-state RTN on capture and release of a carrier. In fact, the number of possible levels created due to RTN is equal to 2^N where N is the number of traps. Switching traps will be those close to the Fermi level of the transistor (Fig. 2.1b). The effects of a trap will vary depending on its position and resulting energy level. Lower trap energy levels will favour the capture of a carrier whilst higher energy levels will favour the emission of a carrier. The presence of switching traps is rare so in most of the cases, only one trap will be present. As a result, two-state RTN is the most commonly found [7].



(a) nMOS carrier trapping mechanism. **(b)** nMOS energy band diagram.

Fig. 2.1. Capture and emission of carriers in nMOS transistors. Extracted from [7].

The typical temporal behaviour induced by RTN in transistors is shown in Fig. 2.2a. It appears as a fluctuation in Ids under constant gate voltage in the order of tens of nA. Two states can be observed: a low current state in which a carrier has been captured by the trap, thus also increasing Vth, and a high current state in which the trap is empty. It can be seen in Fig. 2.2b that under RTN-induced variation, the power specral density (PSD) of the current fluctuation follows a Lorentzian power spectrum of $1/f^2$.



(a) nMOS drain current fluctuation for different gate biases.

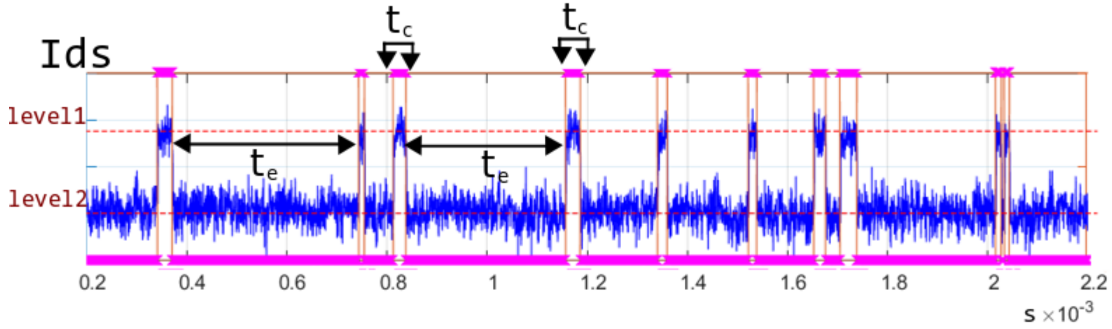
(b) nMOS drain current fluctuation PSD.

Fig. 2.2. 2-level RTN-induced Ids variation. Extracted from [7].

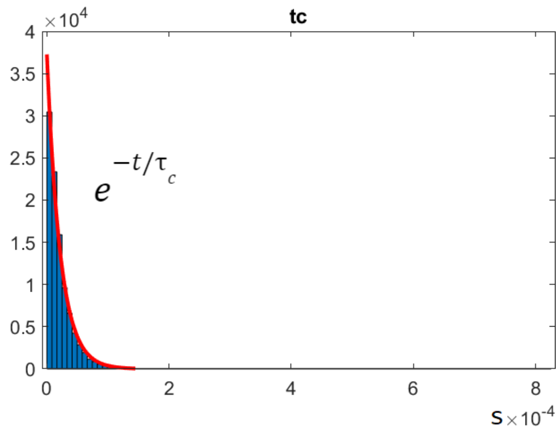
To model RTN, a statistical approach is taken. Parameters τ_c, τ_e and ΔI_{ds} (or ΔV_{th}) characterise the behaviour of a two-state RTN and will vary from transistor to transistor as their trap location is random and thus will be different. ΔI_{ds} is the change of I_{ds} due to RTN under constant voltage biasing. A carrier's capture and emission time are the time it takes for a carrier to be trapped or released to and from the trap. If a carrier's capture and emission times are defined as t_c and t_e, their values are purely random and, if caused by a single trap, will follow exponential distributions ($e^{-t/<\tau>}$) of constants τ_c and τ_e respectively as shown in Fig. 2.3. Being an exponential distribution, these time constants will also be the average of all values under infinite observation time. Additionally, these fitting parameters τ_c and τ_e are characteristic to a device and will follow a logarithmically uniform distribution from 10^{-7}s to 10^{-1}s [11]. The rate at which changes due to RTN are produced depends on both time constants value. So, for a specific device, the maximum RTN rate will be at the conditions for which time constants sum has the smallest value.

Studies show that a device's τ_c and τ_e values will be affected by factors such as temperature, switching frequency of the transistor or, in the case of FDSOI, body bias voltage. As for temperature, its effect can be explained by the τ_c and τ_e expressions Eq. 2.1 where the parameter of interest is T, the temperature. As temperature increases, the exponential fitting constants will decrease resulting in overall more frequent RTN-induced variation [6].

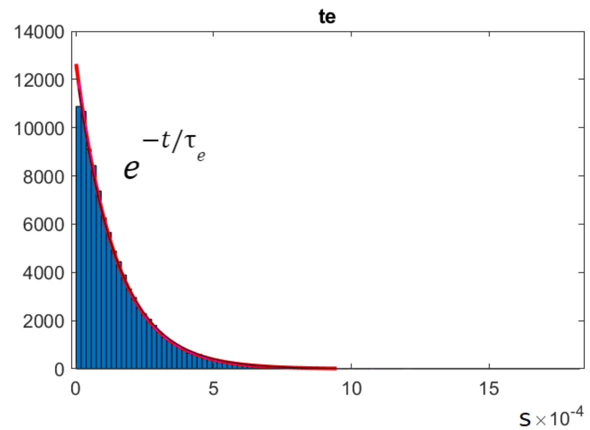
$$\tau_C = \frac{exp(\Delta E_C/KT)}{I_D T \sigma_0 \zeta} \qquad \tau_E = \frac{exp(\Delta E_E/KT)}{T^2 \sigma_0 \eta} \tag{2.1}$$



(a) RTN-induced waveform.



(b) t_c values distribution.



(c) t_e values distribution.

Fig. 2.3. t_c, t_e, τ_c and τ_e definition.

Some studies such as [6] have shown the effects of RTN in FDSOI technology. The case of FDSOI is an interesting one since it introduces the ability of controlling threshold voltage through substrate bias. Fig. 2.4 shows the effect FDSOI front gate bias and substrate bias (VG2) has on τ_c and τ_e. Increasing front gate bias also increases τ_e but decreases τ_c and the slope over τ_c is much faster than for τ_e. Additionally, the increase of substrate bias, for an specific front bias, decreases τ_c but increases τ_e. This is an advantage of FDSOI technology in that it allows for tuning the speed at which RTN is occurring, to either maximise or minimise its effects by controlling substrate bias without having to modify the gate bias.

2.2 RTN in digital circuits

As operating voltage decreases, this discrete change in threshold voltage due to RTN becomes more significant as it translates to a higher and notable percentage of the supply voltage. In the case of digital circuits, this will heavily affect switching delays resulting in altered ring oscillator (ROSC) frequency, or, in the worst case, timing violations in digital circuits.

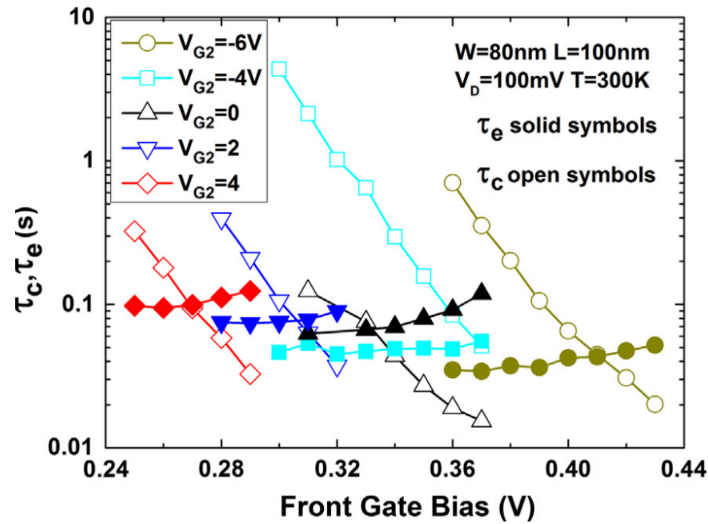


Fig. 2.4. τ_c and τ_e for different gate and substrate bias. Extracted from [6].

A switching transistor will have different effective time constants and trap occupancy probability than in DC behaviour and they will depend on the switching frequency. This is explained by the fact that it will be characterised by two sets of constants. One set for the ON voltage and another for the OFF voltage. Due to gate voltage dependence, as shown in Fig. 2.5, when the transistor is in OFF state the emission constant is smaller than in the ON state, while the opposite is true for the capture constant. Ref. [5][12] also explain that τ_e decreases as frequency increases but τ_c has no observable change due to frequency variation. Trap occupancy probability can be approximated to $P_{AC} = \tau_e(f)/[\tau_c(f)+\tau_e(f)]$. As a result, the trap occupancy probability also decreases as frequency increases until it saturates and cannot decrease any more (Fig. 2.6).

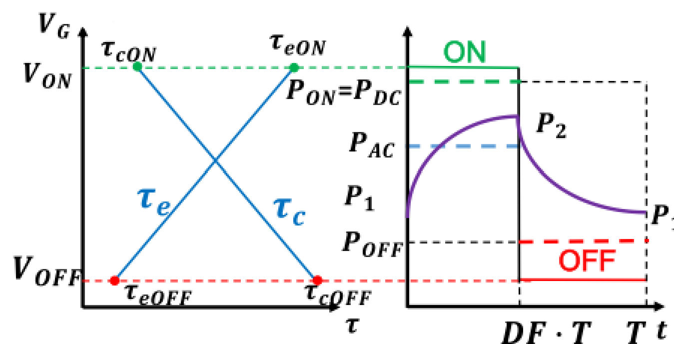
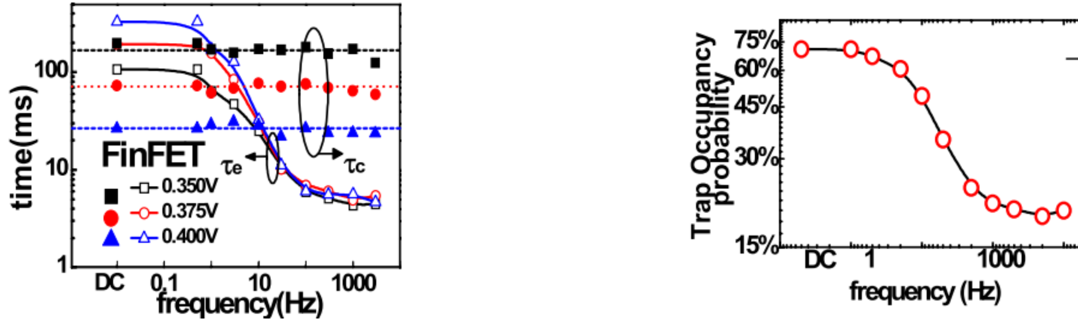


Fig. 2.5. Time constants(left) and occupancy probability(right) as a function of gate voltage. Extracted from [5].



(a) Time constants as a function of switching frequency.

(b) Trap occupancy probability as a function of switching frequency.

Fig. 2.6. Switching frequency effect on RTN. Extracted from [5].

2.3 Security primitives based in RTN

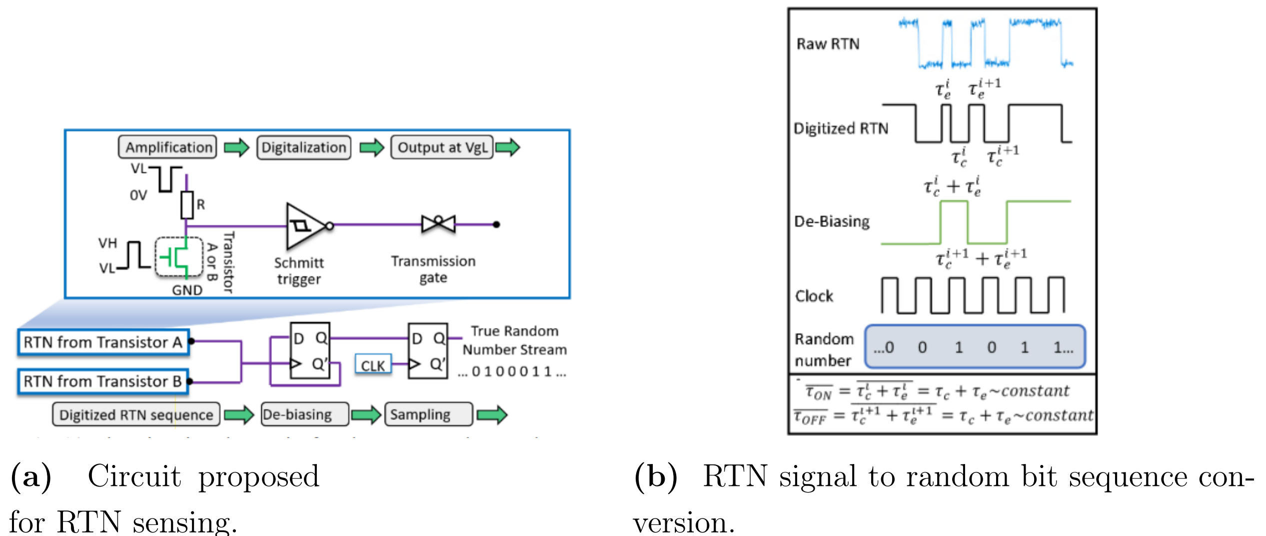
RTN as a noise source can mask randomness sources such as the ones employed in physical unclonable functions. However, in a different perspective, the inherent randomness of RTN makes it a possible source of entropy for security primitives to either generate true random numbers or create physical unclonable functions. Using RTN has both advantages and disadvantages when compared with other sources for certain uses. In the following paragraphs, different options for true random number generators and physical unclonable functions that use RTN as an entropy source will be explored.

2.3.1 True Random Number Generators

A True Number Generator (TRNG) is a device that uses a physical phenomena as a source of entropy to create a truly random digital number. Usually, sources like thermal noise in the form of clock jitter or metastability are used. However, RTN as a source can provide lower power consumption and more robustness to temperature and supply variations. This can be achieved mainly by sacrificing speed and space. More space is required due to the low chance of having a device with RTN, at time constants within a desirable range and relationship, that has to be compensated by increasing redundancy. To evaluate the randomness of a sequence produced by a TRNG, the NIST tests are usually employed. NIST tests are a set of tests developed by the National Institute of Standards and Technology initially intended to evaluate the randomness of a binary sequence that has been generated through an algorithm [8]. They consist of a total of 15 tests which evaluate mainly the frequency of ones and zeroes, the length of a burst of ones and zeroes in relation to the total length of the sequence and the repeatability of a subsequence within the full sequence.

To compare TRNGs on top of randomness, another important metric is the rate at which they can generate random numbers. Additionally, the vulnerability to attacks should also be considered since the ability to externally force a TRNG to generate more predictable numbers would affect its potential use in cryptography.

In the recent years, different approaches for the use of RTN in TRNGs have been proposed. A first option is to obtain RTN by directly sensing the transistor's current over time. Then, that signal can be amplified, quantified to 1's and 0's and then subsampled at a clock signal with a period close to τ_c and τ_e. This is the approach taken in [3], [1] and [4]. A circuit proposal for such approach is given in [1] and shown in Fig. 2.7a. Most of the time, de-biasing is necessary in order to obtain equal probability of 1's and 0's in the output sequence since τ_c and τ_e might not be equal. Ref. [1] also proposes the use of a derived signal from the digitized RTN that switches at every rising edge, making a value last for a time $t_{c,e} = t_c + t_e$ (Fig. 2.7b). This new de-biasing signal allows for creating a sequence of equal probability of 1's and 0's even under very different capture and emission constants. This still results in a random sequence with a 1 and 0 hold time with the same average that follow a hypoexponential distribution, being the sum of two exponential distributions. Other options available for de-biasing are the use of algorithms such as the Von-Newman algorithm [2].



(a) Circuit proposed for RTN sensing.

(b) RTN signal to random bit sequence conversion.

Fig. 2.7. RTN-based TRNG through Ids monitoring. Extracted from [1].

The main challenge of using this approach is the need to tune the sampling period to a value close to $\tau_c + \tau_e$ which is specific to every device. If oversampled, the obtained bits become more predictable. To work around this limitation, Ref. [4] suggests using a larger array of transistors to then select those with constants compatible with the desired sampling frequency.

A second method for using RTN for TRNG is the one given in Ref. [2]. In this case, the RTN signal is passed through an edge-to-pulse circuit to generate a sampling signal that is then used over a constant clock signal to obtain the random sequence. The edge-to-pulse circuit described in the paper is shown in Fig. 2.8 and is, in practice, an edge detector. The rate of RTN events dictates the sampling frequency so no fine tuning of the sampling frequency is necessary and, since the sampling signal only takes into account level changes, devices with more than one trap which produce multi-level RTN can be used effectively. The bit rate of this TRNG is then dependant on the RTN time constants and won't be fixed. In cases where multi-level RTN is present, this topology will bring increased bit rate.

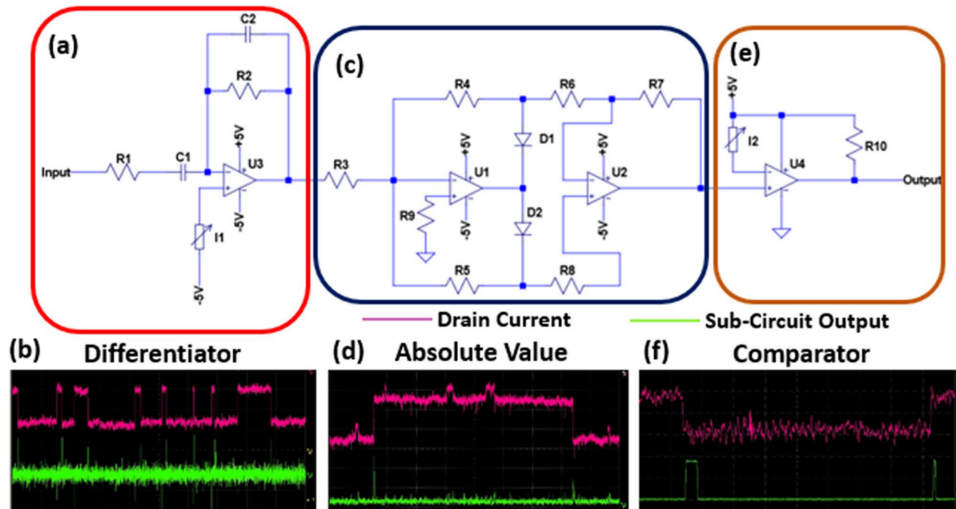


Fig. 2.8. Edge to pulse circuit for RTN detection. (a) differentiator circuit diagram. (b) differentiator input to output example. (c) Circuit diagram for absolute value circuit. (d) absolute value circuit input to output example. (e) Comparator circuit diagram. (f) comparator input to output example. Extracted from [2]

2.3.2 Physical Unclonable Functions

A Physical Unclonable Function (PUF) is an integrated circuit that generates a unique response (output) to a challenge (input). This response has to be secret and different to others ICs with the same design. This can be achieved by taking advantage of inherent physical variations produced in the manufacture of the device. Since these manufacturing variations cannot be replicated, PUFs can then be used in cryptography to either identify the device or create crypto-secure systems where random secret bits do not need to be stored in memory.

To evaluate a PUF's performance, three main metrics are used: uniqueness, reliability and its randomness. Uniqueness refers to the difference in responses between PUFs of the same design by looking at their challenge-response pairs (CRP). Reliability measures the capability of a PUF to consistently create the same response to a challenge under different environmental conditions. Finally, randomness is the PUF's response unpredictability. The attractiveness of using RTN to create a PUF from its randomness which provides uniqueness and is its resistance of deterioration over time, making it a more durable device and increasing its reliability. Two different types of PUF can be identified. Weak PUFs are ones with low number of CRPs but that offer high reliability; these are usually employed for cryptographic key generation. On the other hand, strong PUFs are those that provide a large number of CRPs; they are usually employed for identification and authentication.

One example of RTN being used to implement a PUF is shown in Ref. [11]. In this case, the effect RTN has over ROSC frequency is employed. The PUF in question consists of an array of ROSCs the frequency of which will vary due to RTN each with different time constants. The time constant comparison result between ROSCs will be specific to the PUF. The challenge of this PUF is the pairs of ROSC to compare and the response will be a single bit per ROSC indicating which of the two has faster RTN. The block diagram of the structure detailed in the paper is shown in Fig. 2.9. Counters are used to evaluate the frequency and detect changes due to RTN, which are identifiable over the changes due to jitter by being larger than a pre-defined frequency difference Δf_{ref}. The counts of the two ROSCs can then be compared to produce the response bit.

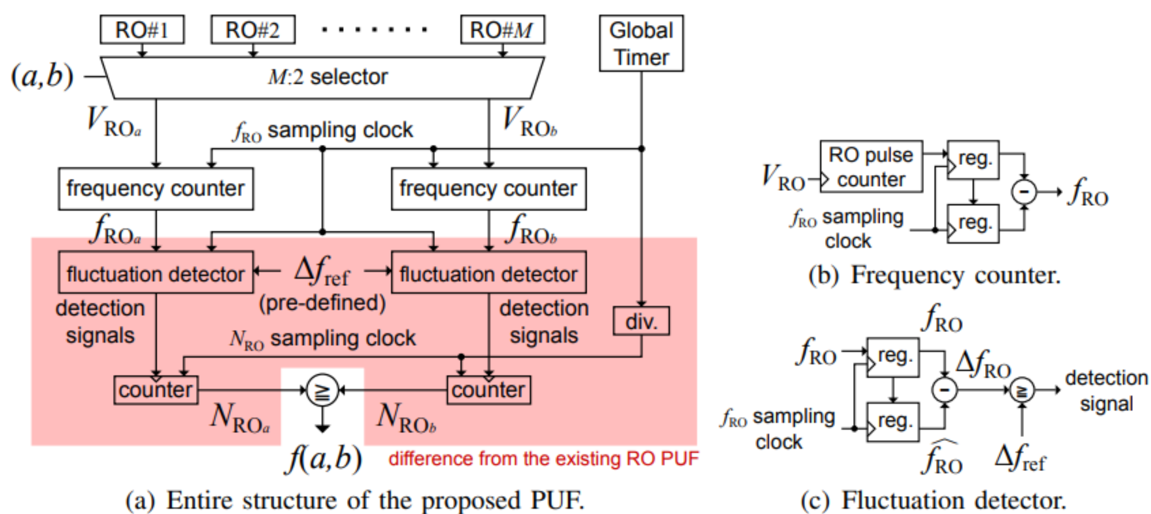


Fig. 2.9. RTN-PUF block diagram example. Extracted from [11].

3. Methodology for Experimental Analysis of RTN in FDSOI-based ROSCs

3.1 The chip under study

To evaluate the effect of RTN on FDSOI ring oscillators, an experimental analysis was performed. The chip used for such experimentation is called ROSQUILLAS, shown in Fig. 3.1. It contains an array of ROSCs in the ROSCON block, a transistor array in TURRON that went unused in this study, an odometer intended for ROSC frequency comparison and a decoder unit that allows for external communication and ROSC selection. In total, it contains an array of 3072 ROSCs developed in ST FDSOI 28-nm technology. Each ROSC consists on 5 stages of NAND gates instead of NOT gates which allows for ROSC deactivation through a control signal (Fig. 3.2).

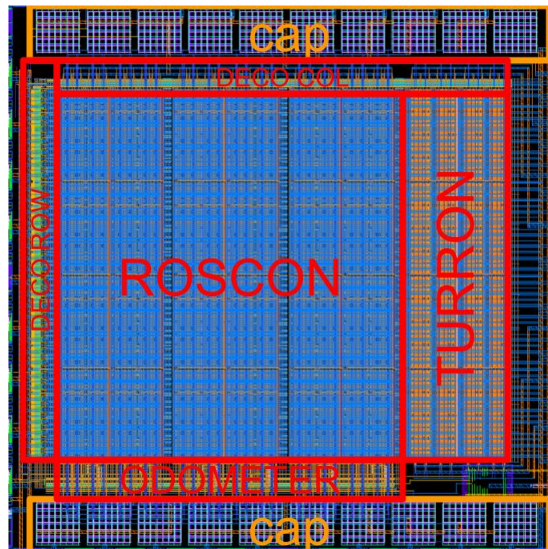
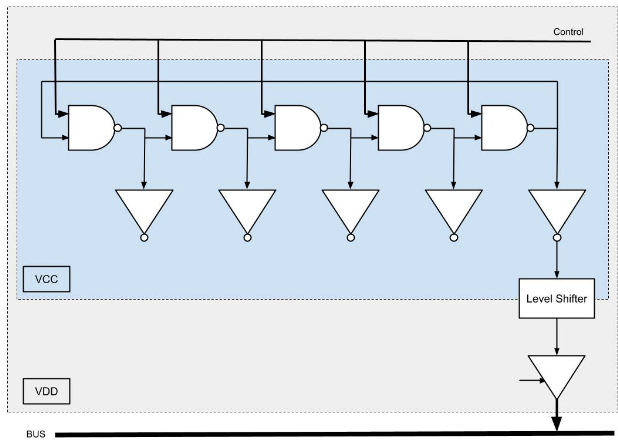
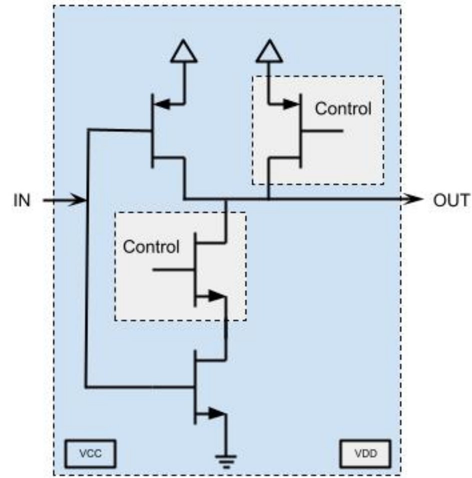


Fig. 3.1. Studied chip core.

The FDSOI transistors used in the NAND gates of the ROSCs have the structure shown in Fig. 3.3. Using FDSOI allows for electrical modulation of the threshold voltage by modifying the body bias. This will modify the ROSC frequency as well as the RTN time constants. The Poly Bias (PB) of a transistor determines its size. The length of the gate is equal to the minimum length plus its PB value in nanometers. Table 3.1 breaks down the length for each PB value. Additionally, for the used NAND gates, the width of the PMOS is 164nm, larger than the 108nm for the NMOS.



(a) ROSC structure.



(b) NAND gate as delay stage.

Fig. 3.2. Studied chip ring oscillator.

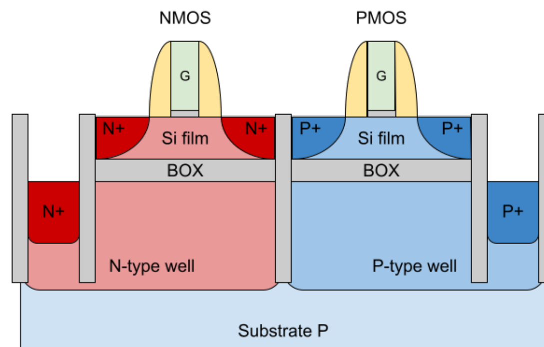


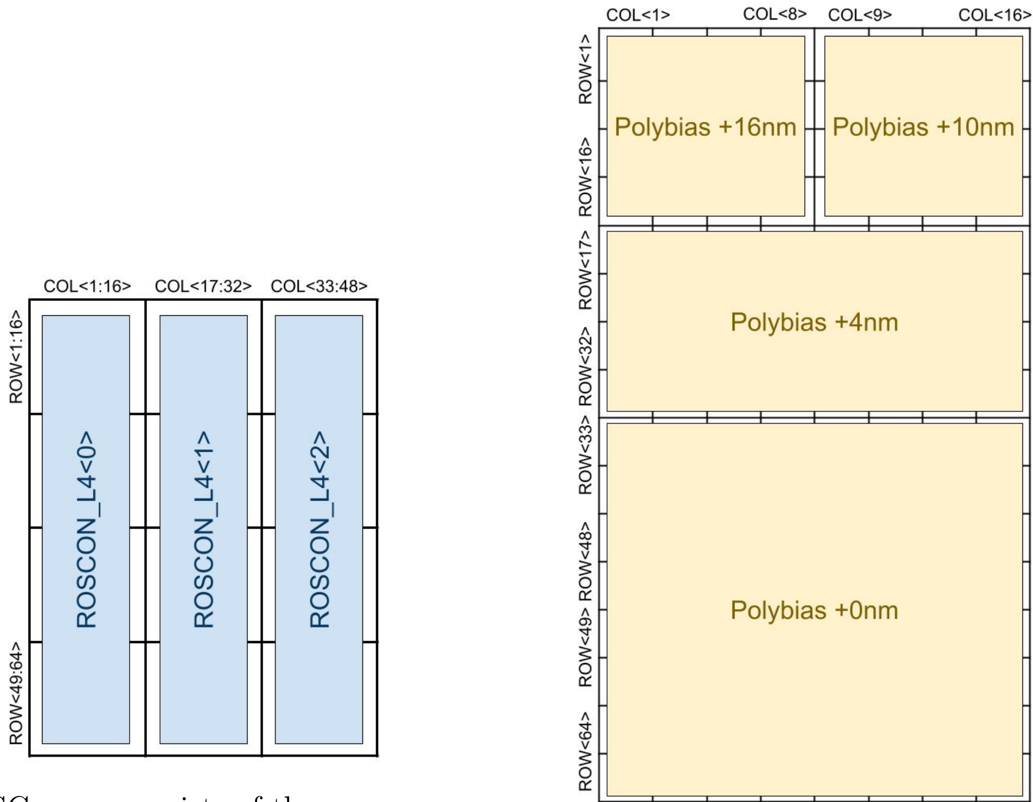
Fig. 3.3. Used FDSOI technology FDSOI structure.

PB	0	4	10	16
L(nm)	30	34	40	46

Table 3.1 PB vs length table for FDSOI-28nm NAND gate.

The chip is powered by two supply voltages: VCC is driven specifically to the ROSCs and VDD gives power to the rest of the chip. The ROSC outputs form the VCC domain (and thus, VCC voltage for the '1' logic level) are converted to the VDD domain with level shifters. This allows for ROSC supply voltage modification without affecting the rest of the circuit. Additionally, the ROSC's NMOS and PMOS body bias, called GNDS and VCCS respectively, are also supplied externally.

Each ROSC in the chip can be identified by its column-row position in the array. The ROSC array structure is shown in Fig 3.4. It is made out of three equal blocs and each of these blocs is divided in sections where the PB of the gates is different. Half of the oscillators will have PB0, 1/4 will have PB4, 1/8 will have PB10 and the remaining 1/8 will have PB16.



(a) ROSC array consists of three identical ROSCON_L4 blocks

(b) ROSCON_L4 block PB distribution

Fig. 3.4. Full ROSC array

The output waveform of the selected ROSC can be driven to the IC output terminals either directly or through a frequency divider in case of oscillators with frequencies too high for the output IO cell. In both cases, the output is driven through a buffer to the output of the chip. The oscillators can also be driven in pairs to a built-in odometer which detects frequency shifts by calculating their frequency relationship (Fig. 3.5). To analyse the effect of RTN over the FDSOI-based ring oscillators, the direct oscillator output has been observed.

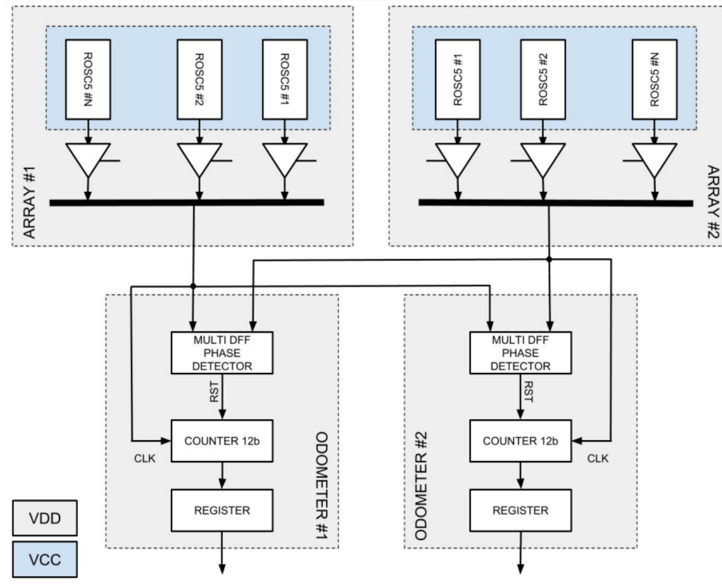


Fig. 3.5. Dual ROSC array for frequency comparison with odometer.

3.2 Measurement setup

To set the supply of the chip and read its output, the setup described in Table 3.2 has been used. Supply voltage VCC and biasing voltages VCCS and GNDS are generated by means of external supplies. The output of the chip is captured by using a digital storage oscilloscope (DSO). A peltier cell has been added on the IC package surface, as shown in Fig. 3.6, to study the effect of a temperature change on RTN. In its used input range of -1 A ($\Delta Temp = 9°C$) to 1 A ($\Delta Temp = -7.5°C$), it provides a temperature variation of 16.5 $°C$. Additionally, to communicate with the chip in order to select an specific oscillator output, a microcontroller has been used. This microcontroller is connected through USB to the measurement PC and translates the instruction coming from MATLAB to a serial data input for the chip. The chip contains a decoder that translates the series input data to the signals that enable and drive the selected ROSC to the output of the chip.

The direct ROSC output is then processed digitally by the DSO to obtain its period through time. Two modes for ROSC period extraction have been used. The first method, named as "continuous", consists on calculating the direct period value at every ROSC rising edge. The other method, named as "averaged", consist on using a trigger signal to periodically calculate the average period in a time window determined by sampling frequency (which is adjusted to ROSC period) and the memory of the capturing device.

Device	Model	Signal	Nominal	Range
DSO	Agilent DSO90804A	Chip output	-	-
Microcontroller	Arduino nano 3	Chip input	-	-
Power Supply	Keysight E3632A	VCC	0.25 V	0.2 to 0.3 V
SMU	Keysight B2912A	VCCS	0 V	-1 V to 0.3 V
		GNDS	0 V	-0.3 V to 1 V
Signal Generator	Agilent 81150A	Oscilloscope trigger	-	10 μs, 100 μs, 1 ms
SMU	Advantest R6340A	Peltier cell supply	0 A	-1 A to 1 A
Peltier cell	Adaptive ET-017-14-11	$\Delta Temp$	0 ºC	-7.5 ºC to 9 ºC

Table 3.2 Studied chip measurement and communication setup

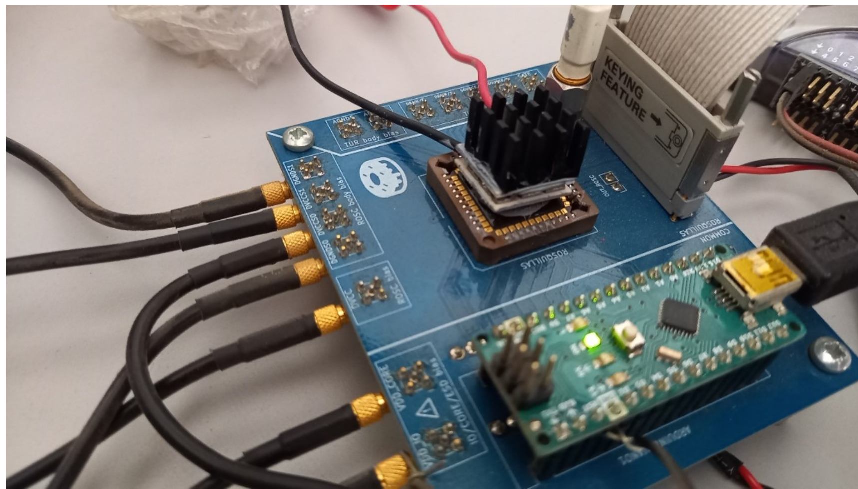


Fig. 3.6. Studied chip and measurement board.

The sampling frequency of the DSO is adjusted to the frequency of the ROSC to ensure enough resolution for period measurement. Before a full sequence reading, a pre-estimation of the ROSC frequency is made and a sampling frequency 500 times that of the ROSC is selected. The oscilloscope has memory for a total of $205 * 10^6$ measured points. When in continuous mode (where measurement is done without trigger signal) the total length of an extracted sequence is all the measured period values within the $205*10^6$ samples. However, in averaged mode (measuring using a trigger signal) the full length is divided in 65636 segments of 3126 points from which the statistics are stored, each segment being separated by the time length dictated by the trigger signal. This results in a sequence of 65636 period values that are the average within each 3126 point segment. Since each segment is 3126 samples at a sampling frequency 500 times that of the ROSC, the period value of each segment will be

the average of 6 measured period values. An schematic showing how both modes work is presented in Fig. 3.7. One thing to note is that, since the time between samples is determined by the ROSC period, the measurement time of the full recording in the continuous mode, or of a segment in the averaged mode, will be variable.

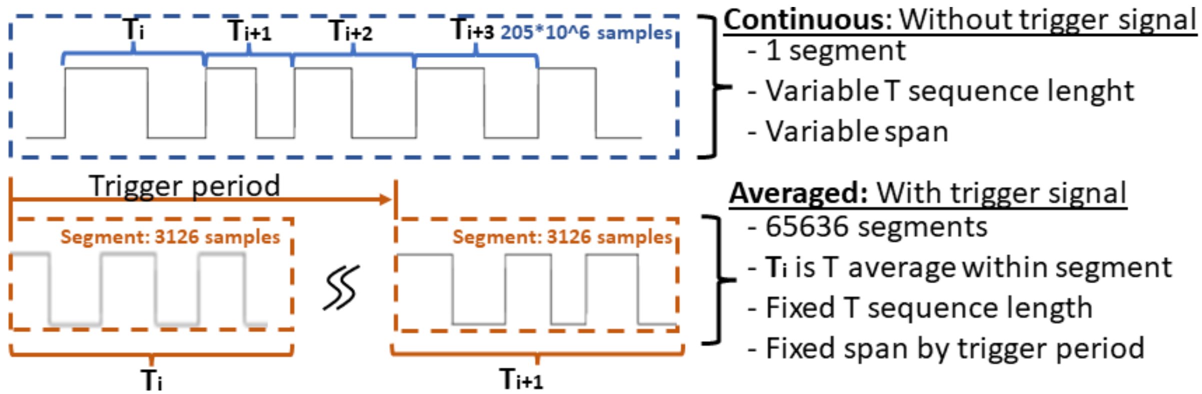


Fig. 3.7. Measurement mode diagram.

In order to cover RTN signals from slow (in the order of μs) to fast (in the order of ms) time constants, the averaged mode actually consists of three options with trigger signals having periods of 1 ms, 0.1 ms or 0.01 ms. Calculating the average of the period instead of saving the raw value allows for longer observation times while using the same amount of space for data storage. Having longer observation time then allows for detecting lower-speed RTN but will suppress higher-speed RTN through the averaging. This means that the trigger should be chosen depending on the estimated τ_c and τ_e. The mode used for each ROSC has been manually selected from a preliminary analysis, looking at the obtained waveforms under each mode. Measuring in continuous mode would be preferable when possible since it has the least amount of processing involved, providing raw period values. Additionally, in averaged mode there's some lost period values due to the unmeasured time between the end of a segment and the next trigger signal. However, averaged mode is still necessary since only the fastest RTN can be detected in continuous mode.

The time required for measurements will depend on the data of interest to extract the measurement mode and trigger used. Additionally, when measuring multiple ROSCs, the time required for ROSC selection becomes significant. For reference, to identify which ROSCs present RTN, a full chip measurement under fixed temperature, supply and bias conditions for both measurement modes and all triggers is necessary. Measuring all ROSC in continuous mode took a full day and 11 hours; measuring with a trigger signal of 0.01 ms took 1 day and 3 hours; measuring with 0.1 ms trigger signal took 1 day and 7 hours and measuring with 1 ms trigger signal took 3 days and 10 hours. This brings the total measuring time for ROSC identification for a single chip under a single bias condition to 7 days and 7 hours.

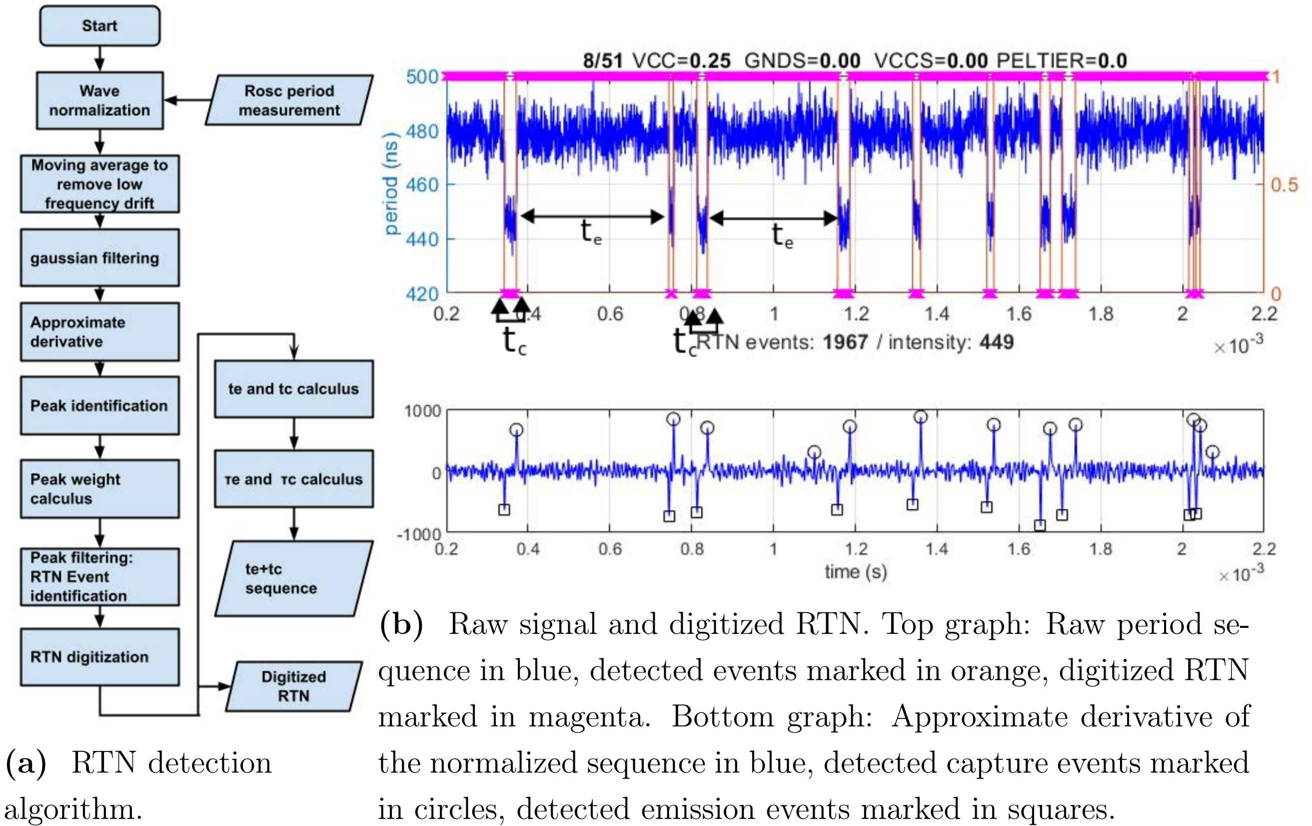
When measuring a single ROSC, the time needed will vary depending on the number of temperature supply and bias conditions to analyse as well as the measurement mode chosen for that ROSC. If only bias condition is to be analysed, the time required can be approximated to $T_{meas} = trigger_time \cdot 65636 \cdot bias_points$ in the case of averaged mode. In the case of continuous mode, the measured time can be approximated to $T_{meas} = \frac{205*10^6}{500 \cdot f_{ROSC}} \cdot bias_points$. This makes the largest measurement time for the full range of 166 bias points, a ROSC measured in average for 1 ms trigger signal, taking a total of 3h 16 min. If multiple VCC or temperature measurements were made, which was most cases, the total time could take days. These long measurement times made unviable the use of all measurement modes and triggers for all ROSC indiscriminately, leading to the need for manual selection of mode and trigger period.

3.3 RTN detection and data processing

The RTN detection process has been done digitally through MATLAB, from either the direct period measurement for the ROSCs with faster RTN or from the averaging at a trigger signal for the slower RTN ones. From these waveforms, the time at each change due to RTN (which were designated as "events") is determined, t_c and t_e are calculated and, through distribution fitting, τ_c and τ_e are extracted.

Two main traits characterise an RTN-induced event. First of all there's the speed of the change: An RTN-induced change should be almost instantaneous, meaning it will be observed in few samples. And the second trait is its magnitude: the size of the step due to RTN should be large when compared with the jitter. If the magnitude is not taken into account, jitter could be falsely detected as RTN due to its high speed. The detection algorithm, which is shown in Fig. 3.8, takes into account both characteristics to identify which period variations are due to RTN.

The algorithm developed starts from a sequence of period readings by the oscilloscope, which is then normalised to facilitate detection across different ROSCs with their own nominal frequency. This new normalised wave is filtered by subtracting the result of a moving average in order to remove low frequency noise. Then, it is Gaussian filtered and run through a differentiator to obtain its derivative, which results in a wave with peaks at times where a sudden change due to RTN is produced. These peaks need to be detected and filtered in order to separate them from those due to jitter. To do so, the level before and after a jump as well as the value of the derivative are taken into account. The level shift is evaluated by calculating the average of the periods in a sample segment, designated as a block, right before and after a jump. By allowing for input parameters of minimum derivative peak height,



(a) RTN detection algorithm.

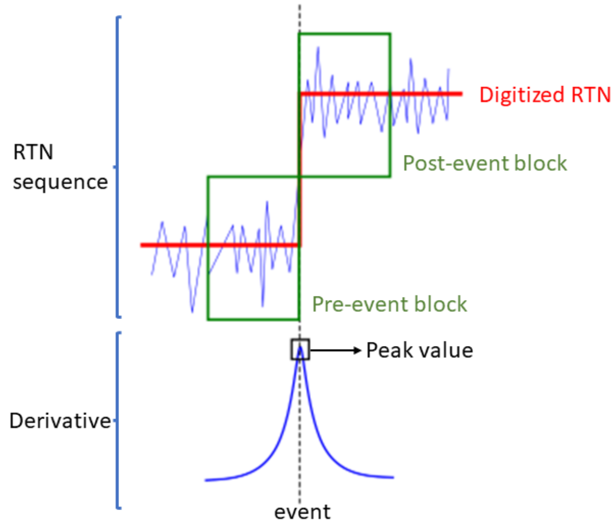
(b) Raw signal and digitized RTN. Top graph: Raw period sequence in blue, detected events marked in orange, digitized RTN marked in magenta. Bottom graph: Approximate derivative of the normalized sequence in blue, detected capture events marked in circles, detected emission events marked in squares.

Fig. 3.8. RTN detection and time constant extraction process.

minimum block difference weighted by the peak value, and the sample size of the Gaussian filter; the detection can be optimised by tailoring it to a specific ROSC behaviour. Fig. 3.9 shows the definition of these input parameters.

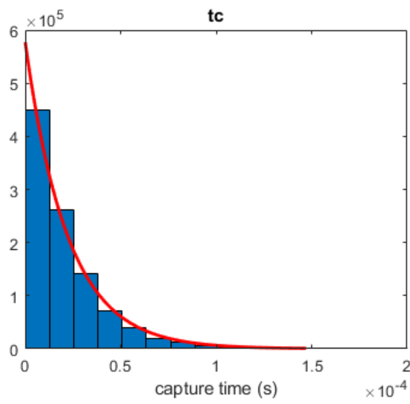
From the detected events, a digitization of the RTN is possible. Each positive peak in the derivative is the result of a carrier capture and each negative peak is the result of a carrier emission. To make the algorithm more robust to detection errors resulting in either false positive or false negative event detection, once an RTN rising or falling edge is detected, the digitised RTN value is held until an opposite edge is detected. This usually results in slightly larger detected t_c or t_e when a detection error occurs.

From the RTN digitization, a t_c and a t_e times sequences are extracted by calculating its pulse width. Then, their distribution fitting constants τ_c and τ_e are obtained through the use of MATLAB's curve fitting toolbox. A de-biased sequence resulting of the addition of consecutive tc and te values is also calculated to evaluate possible implementation in TRNG. An example of the data extracted from a single sequence is shown in Fig. 3.10.

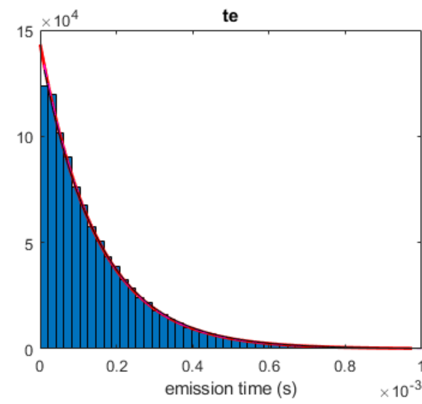


Parameter	Description
`MinPeakHeight`	Minimum derivative peak value
`MinBlockDifference`	Minimum value for the difference between normalised sequence's pre-event and post-event block average values multiplied by the derivative peak value.
`GSize`	Gaussian filter sample size. Determines moving filter, block length and derivative window

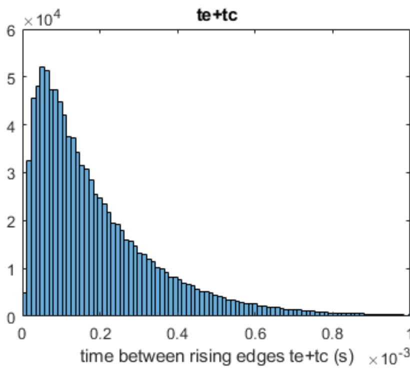
Fig. 3.9. Algorithm adjusting parameters



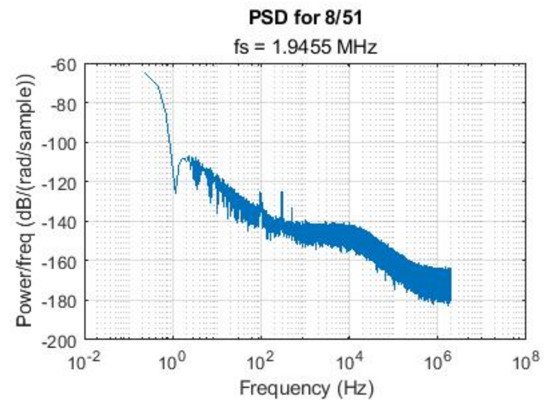
(a) t_c values distribution.



(b) t_e values distribution.



(c) $t_e + t_c$ value distribution



(d) Sequence Power Spectral Density

Fig. 3.10. Extracted data for an obtained sequence.

To identify which oscillators present RTN before exhaustive analysis, a sequence for each ROSC is measured under fixed VCC, GNDS, VCCS and temperature and under different trigger signals. These extracted sequences are run through the algorithm until event identification. This results in the map shown in Fig. 3.11 that acts as a preliminary filtering of the array. To finally identify which ROSCs do present RTN, a final manual check is necessary. An attempt was made at automatic RTN percentage detection, however different parameter values would provide different results. Some cases were able to provide less false positives at the cost of increasing false negatives, a trade-off of false positives and false negatives was present. When choosing the values of the detection parameters, more restrictive values were used to make manual checking feasible. The result provided very little false positives but might have filtered out some false negatives. This, however, did not hurt the main objective which was finding ROSC presenting clear and useful RTN easier to detect for further analysis. After the ROSCs presenting useful RTN are found, when a specific ROSC is measured alone, the parameters are manually chosen in order to optimise detection for that ROSC's behaviour.

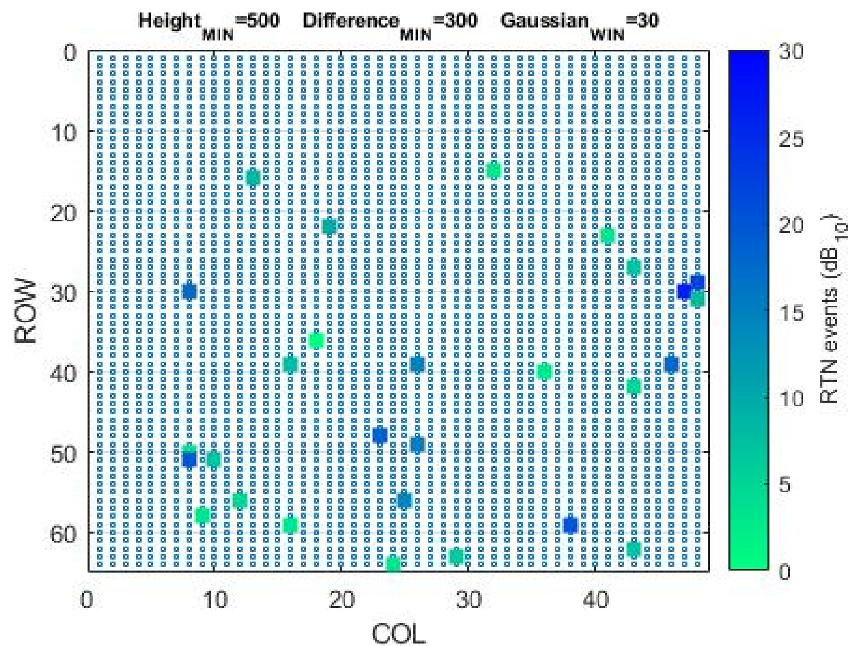


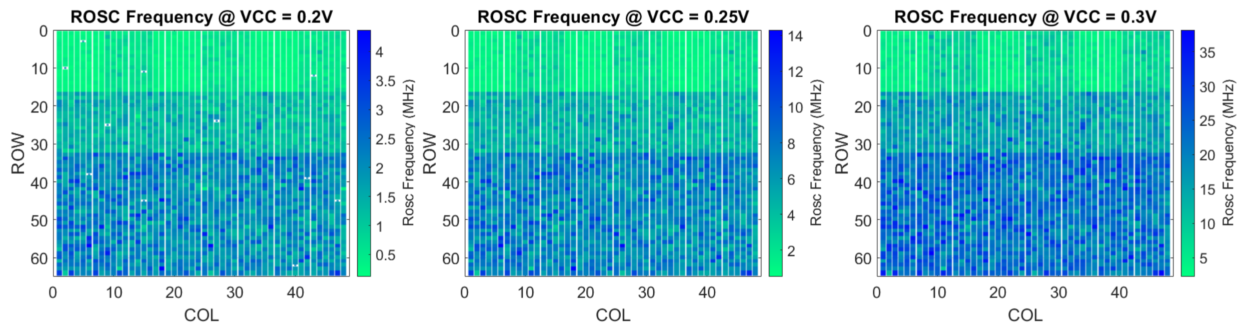
Fig. 3.11. Automatic filtering results for RTN in ROSC identification for chip #1 with no trigger applied.

4. Experimental Results

4.1 Frequency analysis

First of all, the frequency of the full array has been evaluated. In Fig. 4.1 the average frequency readings for different supply voltages are presented. These average frequency values are calculated directly by the DSO, filtering out any jitter present. Under nominal conditions, the frequency of most evaluated chips are between 0.4 MHz and 10.4 MHz. As expected, as supply voltage increases, so does the frequency of the oscillator. At voltages close to VCC=0.2 V and below, some ROSCs cannot operate. Additionally, the frequency of an oscillator will heavily depend on its poly bias, as shown in Fig. 4.1d. The bigger the poly bias, the bigger the transistors are and the lower the frequency of the oscillator will be.

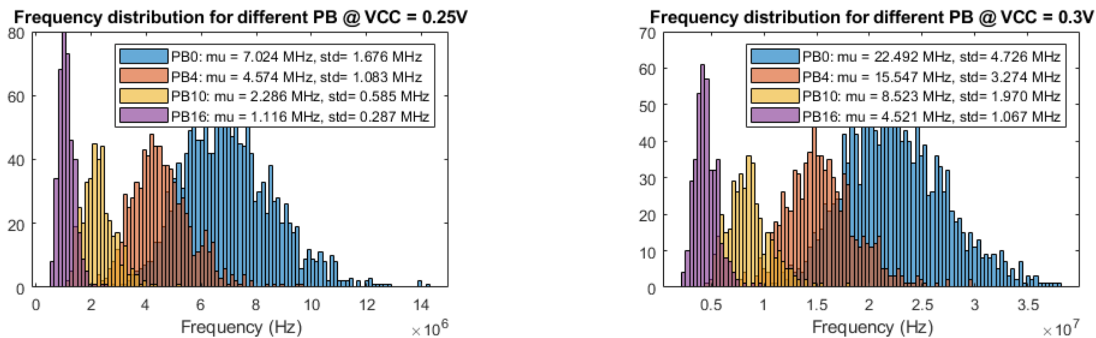
An important thing to mention is that chip #1, on which most RTN measurements were made, shows frequencies about 40% lower than the the rest of the chips. However, such difference in frequency should not significantly effect the results of the RTN analysis.



(a) Frequency map for VCC = 0.2 V

(b) Frequency map for VCC = 0.25 V

(c) Frequency map for VCC = 0.3 V



(d) Frequency distribution for VCC=0.25V. **(e)** Frequency distribution for VCC=0.30V.

Fig. 4.1. Chip #2 frequency analysis.

4.2 RTN characterisation at nominal conditions

A total of 3 chips have been measured to extract ROSCs that present RTN and which of those present usable RTN, at a standard VCC=0.25V, different trigger times and multiple VCCS, GNDS combinations. The results show that about 3,29% of the oscillators present some RTN but a good amount have τ_c and τ_e that are too far apart for reliable analysis. The percentage of ROSCs presenting usable RTN is 0,93%. Table 4.1 and Fig. 4.2 display which percentage of ROSC within each PB group present RTN. Results show that a lower percentage of ROSCs presenting RTN have been found for higher poly bias values which would indicate that either higher PB results in a lower chance of RTN or that it becomes harder to detect. This is most likely the result of masking of the RTN in bigger transistors due to jitter or other noise sources. After all, as poly bias increases, the area also increases and the sensitivity of RTN decreases. With bigger area, the change due to a single electron over V_{th} is les significant.

Poly Bias	ROSC percentage	Total units per chip	Detected RTN	Presenting usable RTN
PB0	50%	1536	3,64%	1,09%
PB4	25%	768	3,60%	1,15%
PB10	12.5%	384	2,86%	0,61%
PB16	12.5%	384	1,74%	0,09%
Total		3072	3,29%	0,93%

Table 4.1 Statistics for detected ROSCs presenting RTN over 3 chips.

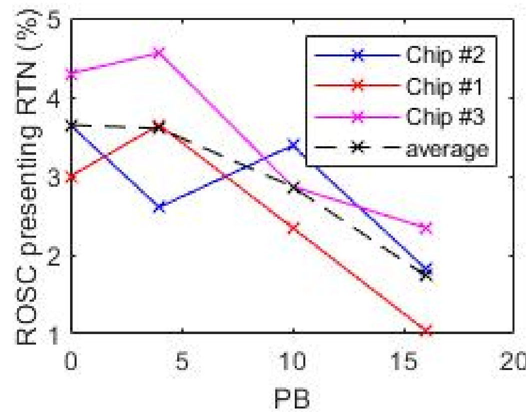
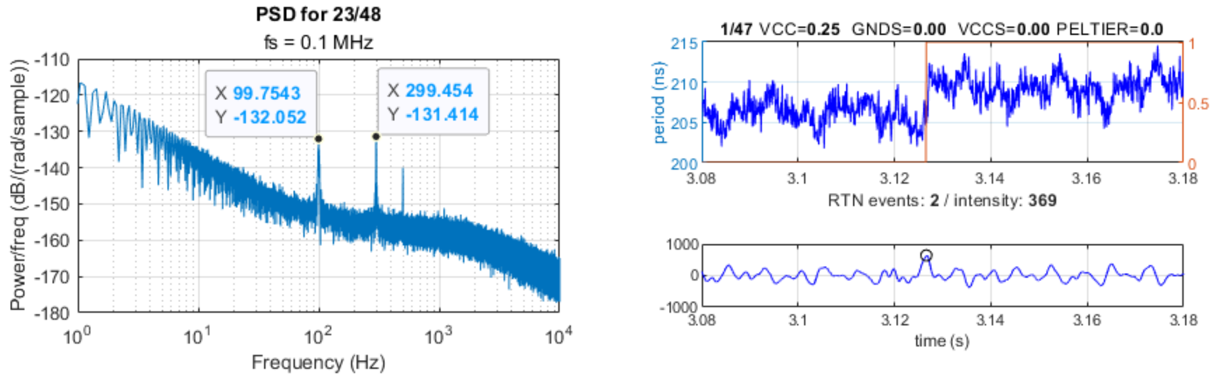


Fig. 4.2. Found ROSC percentage for the 3 studied chips.

One interference common for all ROSCs, which would contribute to the masking of RTN, has been found in the power spectral density function of the analysed sequences. They all present frequency components at multiples of 100Hz, indicating some sort of phase locking of the ROSC of a signal that shifts in frequency at a repetitive pattern every 10 ms (Fig. 4.3).



(a) Sequence PSD presenting 100 Hz interference and odd harmonics

(b) Detected RTN over 100 Hz phase locking interference present in all ROSC

Fig. 4.3. Found interference.

An analysis of the detected ROSCs with usable RTN has shown that most time constants have values between 0.1 ms and 300 ms, and that there's no observable dependence on transistor poly bias (Fig. 4.4). Results seem coherent with the logarithmically uniform distribution described in the literature [11]. However, there's not enough data points to ensure such a behaviour is present in the studied chip.

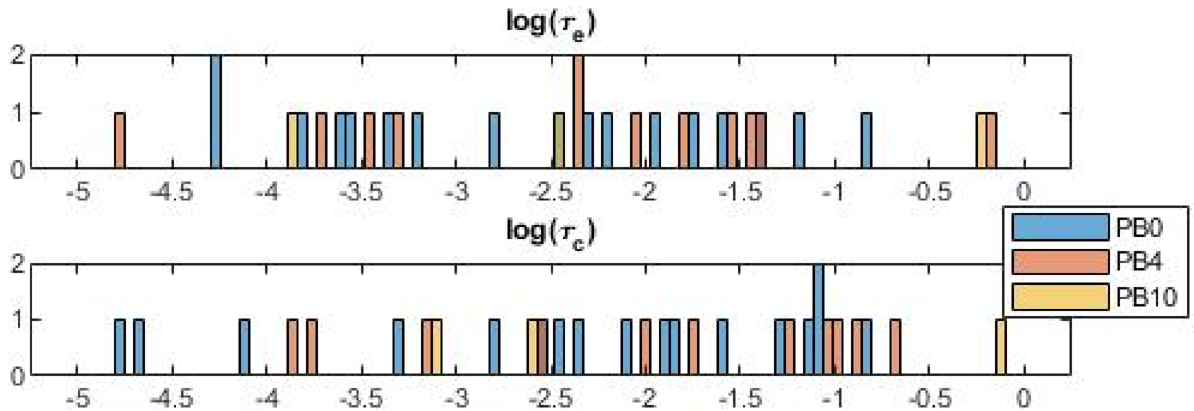


Fig. 4.4. Time constant distribution for detected ROSCs with RTN under nominal conditions for chip #1

Other data manually extracted from identified ROSC in chip #1 are the magnitude of the period value jump due to RTN and the standard deviation of the period measurement due to jitter. To measure the magnitude of the period value due to RTN, a specific point with clear RTN is chosen and the difference between the two levels is calculated. The results presented in Fig. 4.5 show that the jump due to RTN does not seem to depend on the PB of the ROSC. This makes sense since it should depend on the position of the trap which should be random. Although certain dependence should be expected as with higher area, the impact over V_{th} should be smaller. However, when comparing to ROSC jitter, as shown in Fig. 4.6, a more clear PB dependence is observed. As PB increases, the jump due to RTN becomes less pronounced which leads to it being harder to detect.

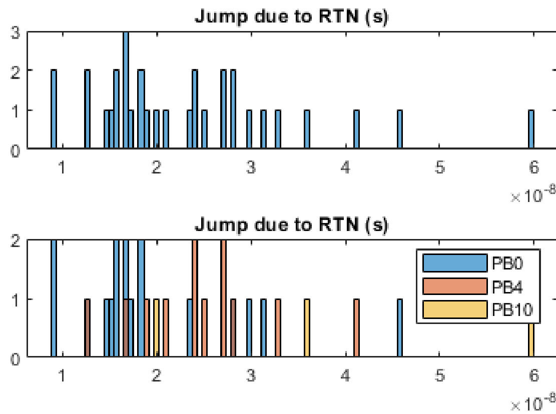


Fig. 4.5. Jump due to RTN value distribution.

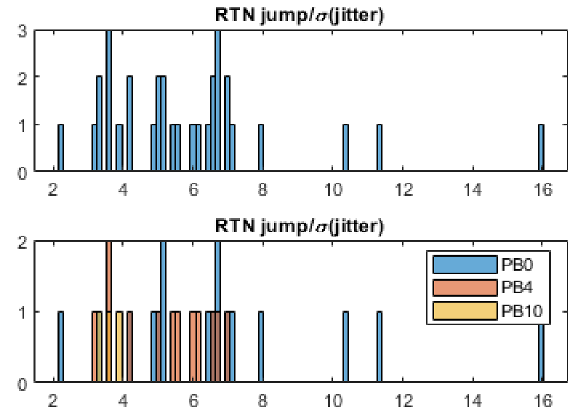
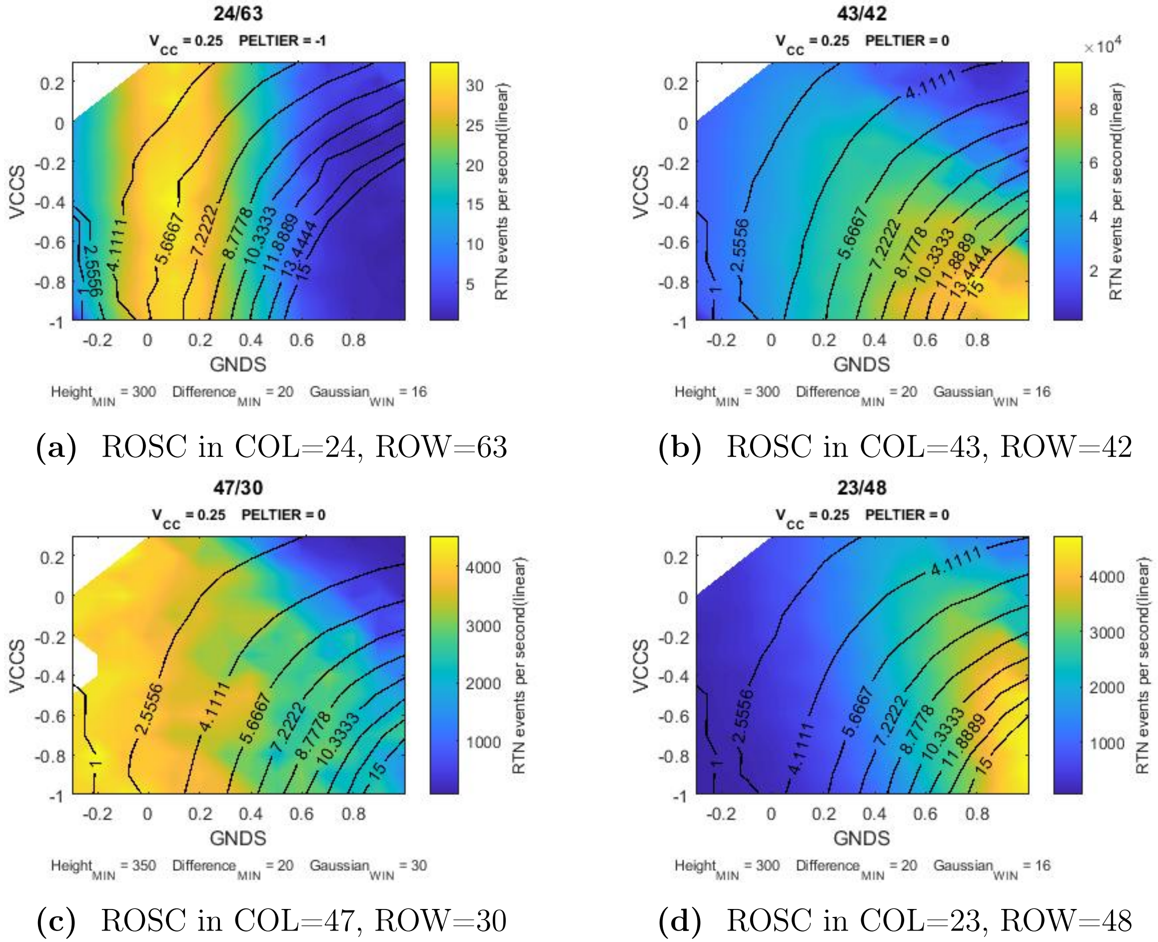


Fig. 4.6. RTN jump to jitter standard deviation relationship.

4.3 Body bias variation effect

Throughout the project, 7 different ROSC have been measured under all possible bias points and under multiple temperature and supply voltage conditions. Fig. 4.7 shows some examples of the effect of varying the body bias of the transistors in RTN speed. Black lines represent iso-frequency points, which show that RTN has no relevant frequency dependence at the working range. The zones at which more RTN is observed are determined by τ_c and τ_e. Wherever their sum is the lowest, the more changes due to RTN will be observed.

Changing the body bias of transistors in propagation gates of an oscillator will affect the amount of RTN events differently for each ROSC. This is the case because it will be dependant on trap location which is random and device-specific. The found increase of RTN rate in nominal conditions from low RTN zones to high RTN zones, shown in table 4.2, can be between 2 times more and 95 times more.



(a) ROSC in COL=24, ROW=63 **(b)** ROSC in COL=43, ROW=42

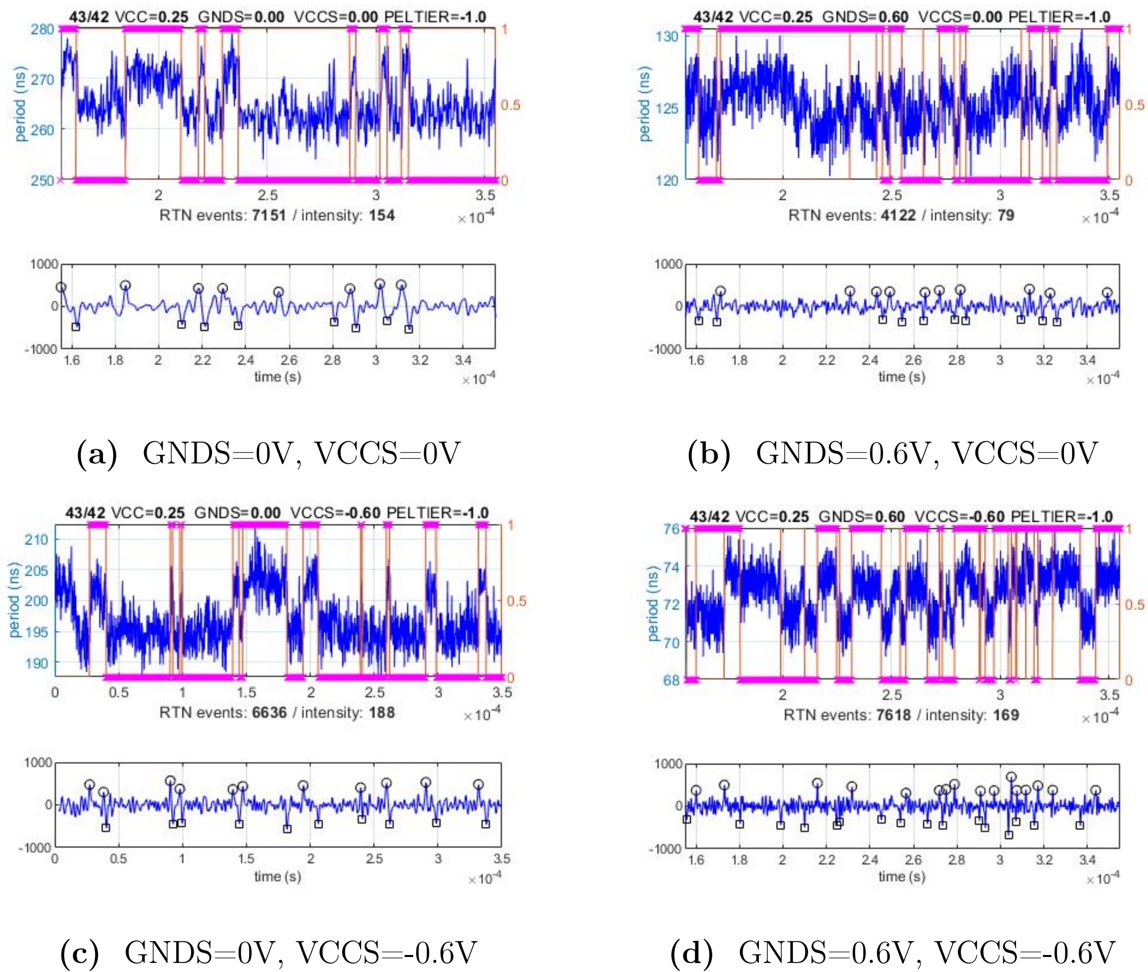
(c) ROSC in COL=47, ROW=30 **(d)** ROSC in COL=23, ROW=48

Fig. 4.7. Detected RTN events and frequency in MHz as a function of body biases for different ROSC in chip #1.

Column	Row	Min event rate (events/s)	Max event rate (events/s)	increase
24	63	0.35	33.26	95 times
43	42	48500	220000	45.36 times
47	30	6000	12000	2 times
23	48	67	4700	71 times
48	29	2246	4830	2.15 times
8	30	174	3600	20.68 times

Table 4.2 Event rate difference due to bias variation under nominal conditions.

One important thing to note about the results shown in Fig. 4.7, is that they are conditioned by detection success. On closer inspection of the sequences at 4 corners of those graphs such as Fig. 4.8, shows an increased difficulty of reliable detection for a combination of higher VCCS and GNDS values like in Fig. 4.8b. There appears to be increased noise combined with smoother transitions which the algorithm will struggle to extract events from. This results in events not being detected which will in turn increase the estimated values of τ_c and τ_e.



(a) GNDS=0V, VCCS=0V

(b) GNDS=0.6V, VCCS=0V

(c) GNDS=0V, VCCS=-0.6V

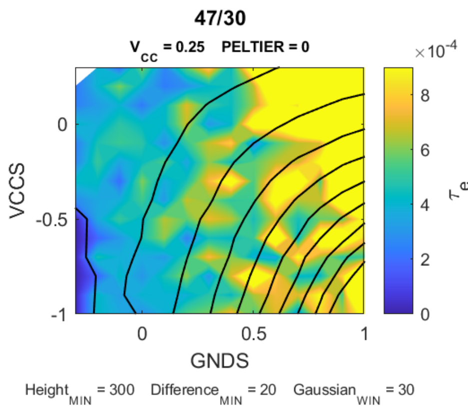
(d) GNDS=0.6V, VCCS=-0.6V

Fig. 4.8. Body bias corner sequences for ROSC in Col=43, Row=42 in chip #1

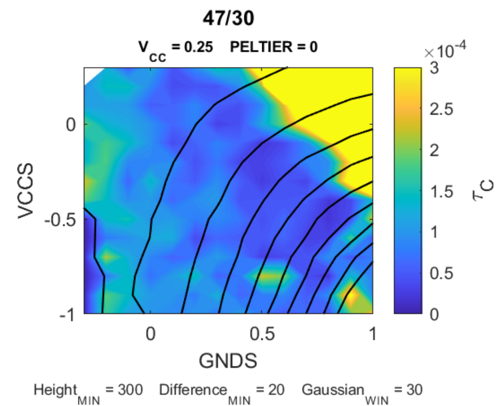
This means that for the case of Fig. 4.7b and Fig. 4.7d, the actual RTN event hot zone is more vertical like in Fig. 4.7a and Fig. 4.7c. This exclusive GNDS dependence would then indicate that the traps are located in the NMOS as opposed to the PMOS where a dominant VCCS dependence would be expected. All 7 analysed ROSC appear to have the RTN trap in the NMOS transistors. According to [10], this technology presents jumps due to RTN in PMOS that are 3 orders of magnitude below that of the NMOS. This would explain the fact

that no traps have been found for a PMOS since any changes due to RTN are easily masked and harder to detect. The body bias position at which the RTN hot zone is centred is entirely random as it depends on trap position; meaning that no definitive general bias dependence can be obtained. However, for a specific device under certain environmental conditions, the hot zone will be unchanging. This means that, through calibration, a bias combinations for either maximum or minimum RTN can be found. In some cases, these RTN changes through body bias could be achieved even without modifying the ROSC's frequency.

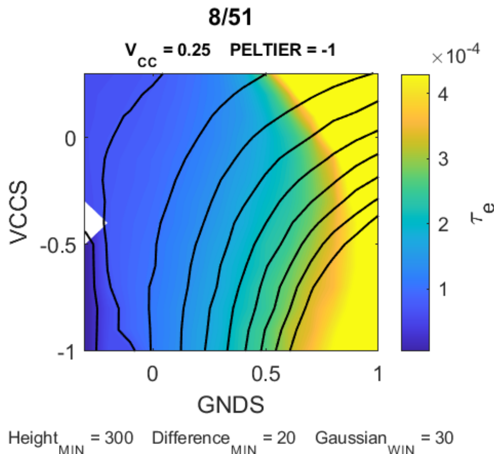
As for the behaviour of the τ_c and τ_e, Fig. 4.9 presents the time constants of two ROSCs under the full range of body bias. Ignoring the upper-right corner with increased time constant value due to the detection issues previously mentioned, results show that τ_c tends to decrease as bias increases while τ_e presents the opposite trend which matches the literature [6].



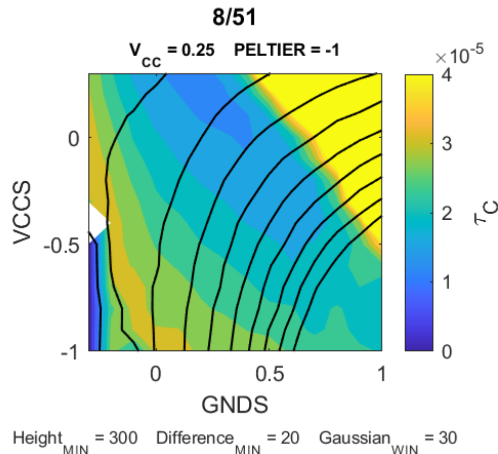
(a) Emission constant for ROSC in position Col=47, Row=30.



(b) Capture constant for ROSC in position Col=47, Row=30.



(c) Emission constant for ROSC in position Col=8, Row=51.

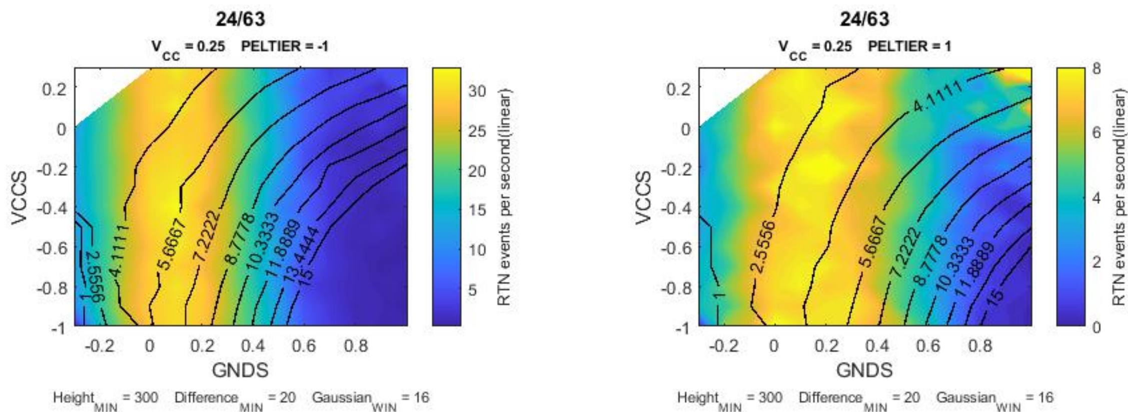


(d) Capture constant for ROSC in position Col=8, Row=51.

Fig. 4.9. Time constants variation due to body bias for two different ROSC in chip #1.

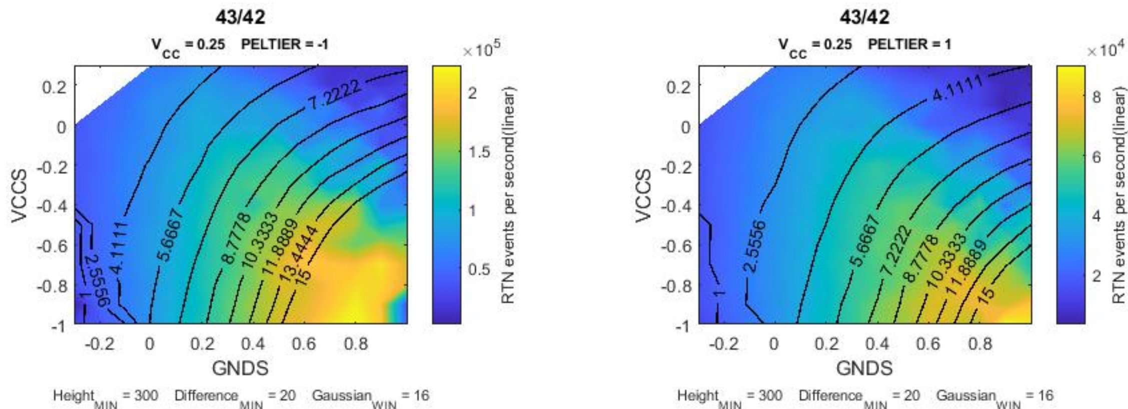
4.4 Temperature effect

Temperature also has an effect on the RTN event rate. Fig. 4.10 shows the results of plotting the body bias dependence of two different ROSC under two different peltier current values. As temperature increases, so does the RTN speed. Over a difference of close to 20 ºC, the rate at which changes due to RTN approximately quadruples. This is consistent with the literature [6] [9] where it is stated that a temperature increase reduces both time constant values. Results show that for the temperature difference analysed, the bias conditions for which the RTN hot spot is found does not change considerably. This means that if an optimal bias point is found, it won't change under these temperature differences. For the optimal bias point to change considerably, the temperature difference would have to be great enough that the relationship between τ_c and τ_e is not only inverted, but also that both time constants are very far apart in value.



(a) ROSC in COL=24, ROW=63 of chip #1 at higher temperature.

(b) ROSC in COL=24, ROW=63 of chip #1 at lower temperature.



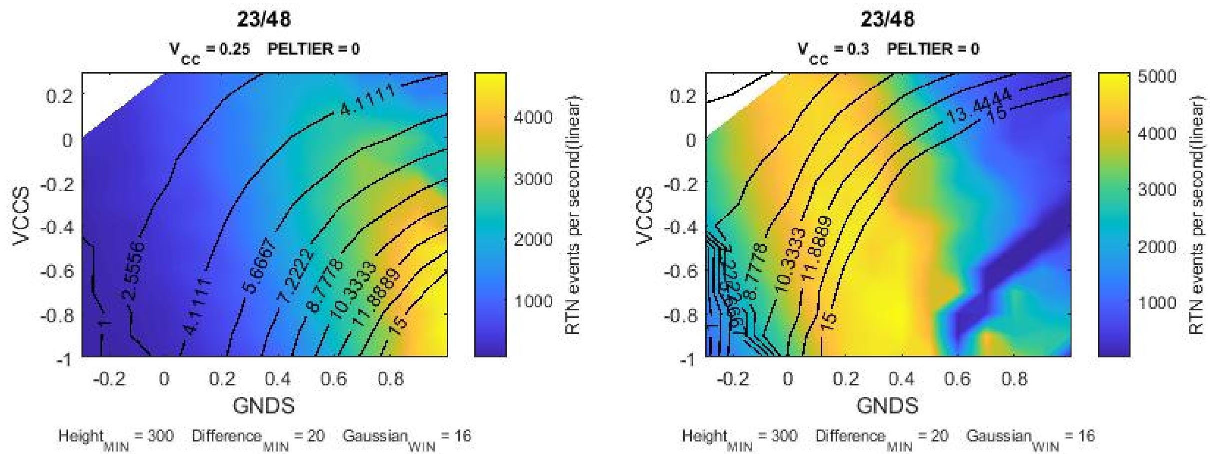
(c) ROSC in COL=43, ROW=42 of chip #1 at higher temperature.

(d) ROSC in COL=43, ROW=42 of chip #1 at lower temperature.

Fig. 4.10. Temperature effect on RTN event rate over 16.5ºC difference.

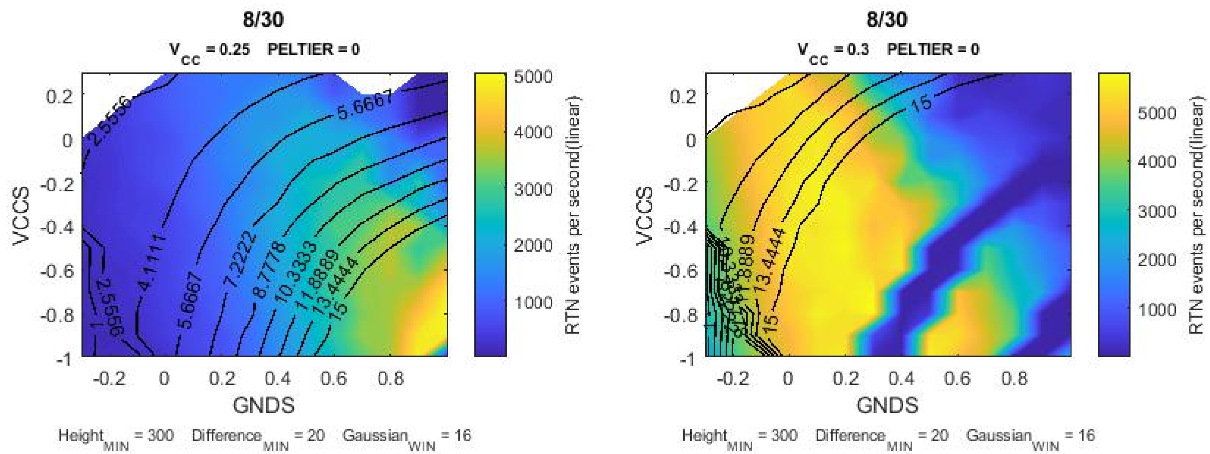
4.5 Supply voltage effect

As for the effect of modifying VCC on the rate of RTN events, it is shown in 4.11. The maximum rate at which RTN is produced stays mostly the same under a supply voltage change, however, there's a shift of the bias dependence by the same amount as the VCC change. This means that a supply voltage change would affect the optimal bias point for RTN.



(a) ROSC in position Col=23, Row=48 at VCC=0.25V.

(b) ROSC in position Col=23, Row=48 at VCC=0.3V.



(c) ROSC in position Col=8, Row=30 at VCC=0.25V.

(d) ROSC in position Col=8, Row=30 at VCC=0.3V.

Fig. 4.11. Effect of a supply voltage change on RTN event rate for two ROSC on chip #1.

If the supply voltage change is great enough, the high RTN area could fall out of the investigated body bias range, thus resulting in an effective decrease of RTN event rate. At the same time, this could also result in an increase of event rate if the high RTN area fell outside the studied body bias and modifying VCC would bring it inside. In any case, one disadvantage of increasing VCC if the aim is using RTN, is that the change in V_{th} due to RTN will be smaller when compared to VCC and will, in turn, be harder to detect. This can be seen, for example, in Fig. 4.11b and Fig. 4.11d. In these figures where VCC is increased, bands of very low detected RTN are present which are the result of detection difficulties.

5. Potential of RTN in FDSOI for TRNG and PUF implementation

In the recent years, different proposals for RTN usage in security primitives have been developed. In this chapter, a discussion of the potential of RTN as a source of randomness for TRNG and PUF applications in FDSOI technology will be discussed. To assess the performance of such a technology, the experimental data previously presented will be employed.

5.1 RTN in FDSOI for TRNG applications

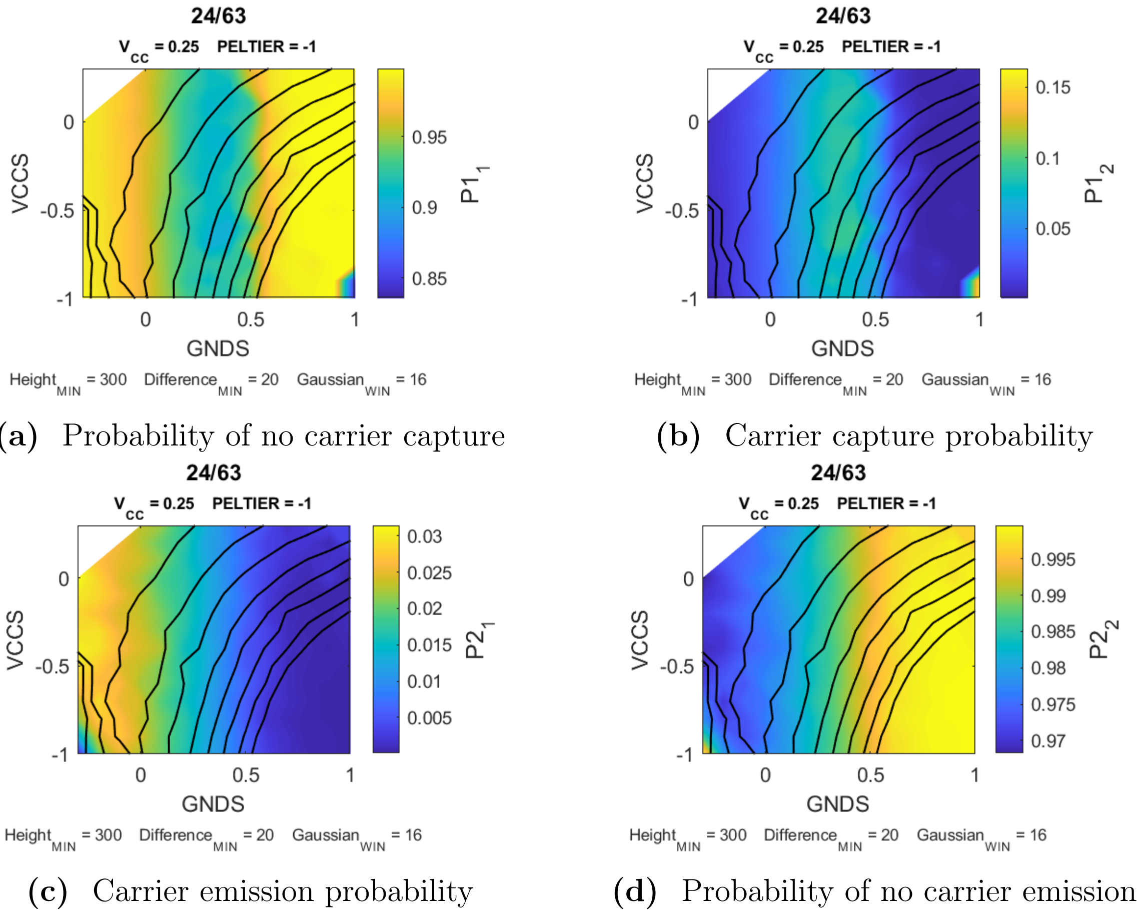
The first step to evaluate a potential TRNG application, is to assess the randomness of the sequences obtained from the ROSCs. From the obtained data, specs for the technology can be obtained and used to calculate a hypothetical rough approximation of TRNG performance.

5.1.1 Sequence randomness

Since the digitised RTN is essentially a sequence of ones and zeroes, it can be modelled as a two-state Markov chain. Fig. 5.1 shows the result of calculating the Markov model of the raw digitised RTN. Markov modelling entails obtaining the probability that the next sample changes or stays at a given RTN state. State 1 indicates RTN=0, meaning there's no captured carrier, while state 2 indicates RTN=1 meaning a carrier is occupying the trap. This can be obtained through the use of the *hmmestimate*() MATLAB function which calculates the hidden Markov model of a sequence.

The graph results show a very high chance for the sequence to stay at a given state. The digitised RTN signal used for the Markov modelling has the same sample rate as the raw period sequence obtained from the ROSC, leading to very long burst of '1's and '0's which results in the great difference between the probability of staying or switching from a state. In order to maximise the randomness in a sequence, the chances of staying at a given state or switching should be as close as possible. This could be achieved by sub-sampling the digitised RTN sequence at a sampling period close to τ_c and τ_e.

However, as previously stated, if the two time constants are not close enough, further de-biasing of the sequence would be necessary in order to ensure equal chance of '1' and '0' in the random sequence. De-biasing entails post-processing the sequence to generate another with equal probability of '1' and '0', which can be achieved by applying algorithms such as the Von-Newmann algorithm [3]. For the sequence to be truly unbiased, meaning τ_c and τ_e



(a) Probability of no carrier capture

(b) Carrier capture probability

(c) Carrier emission probability

(d) Probability of no carrier emission

Fig. 5.1. Markov chain results for digitised RTN sequence.

are equal, the conditions $P1_1 = P2_2$ and $P1_2 = P2_1$ should be met. The further apart these values are, the more biased is the sequence.

In order to evaluate this chip's potential for TRNG applications, the NIST test suite has been employed. The test suite allows for testing the randomness of generated sequences. It works as a separate executable file run through a command prompt window, as shown in Fig. 5.2, in which a binary sequence is fed, as well as the number of bits of the sequence to analyse either in binary or as an ASCII sequence. The sequences are written into a .txt file when processed through MATLAB and are fed later into the test suite. After the program is run, a new .txt file report detailing each test and a general report summarising the results of all tests are generated.

In order to obtain a sequence that is unbiased from any ROSC with usable RTN independently of their time constant relationship, a similar approach to [1] can been taken. In this case, the digitised RTN is used to generate another sequence that changes value at every rising edge, that is, for every carrier's capture and emission. By doing so, a new

```
C:\Users\user1\Desktop\NIST>assess 400000
           G E N E R A T O R    S E L E C T I O N
           ______________________________________

    [0] Input File                  [1] Linear Congruential
    [2] Quadratic Congruential I    [3] Quadratic Congruential II
    [4] Cubic Congruential          [5] XOR
    [6] Modular Exponentiation      [7] Blum-Blum-Shub
    [8] Micali-Schnorr              [9] G Using SHA-1

   Enter Choice: 0


                User Prescribed Input File: 1000loops_suma7b.txt

                S T A T I S T I C A L   T E S T S
                _________________________________

    [01] Frequency                       [02] Block Frequency
    [03] Cumulative Sums                 [04] Runs
    [05] Longest Run of Ones             [06] Rank
    [07] Discrete Fourier Transform      [08] Nonperiodic Template Matchings
    [09] Overlapping Template Matchings  [10] Universal Statistical
    [11] Approximate Entropy             [12] Random Excursions
    [13] Random Excursions Variant       [14] Serial
    [15] Linear Complexity

         INSTRUCTIONS
            Enter 0 if you DO NOT want to apply all of the
            statistical tests to each sequence and 1 if you DO.

   Enter Choice: 1

        P a r a m e t e r   A d j u s t m e n t s
        -----------------------------------------
    [1] Block Frequency Test - block length(M):         128
    [2] NonOverlapping Template Test - block length(m): 9
    [3] Overlapping Template Test - block length(m):    9
    [4] Approximate Entropy Test - block length(m):     10
    [5] Serial Test - block length(m):                  16
    [6] Linear Complexity Test - block length(M):       500

   Select Test (0 to continue): 0

   How many bitstreams? 5

   Input File Format:
    [0] ASCII - A sequence of ASCII 0's and 1's
    [1] Binary - Each byte in data file contains 8 bits of data

   Select input mode:  0

     Statistical Testing In Progress.........

     Statistical Testing Complete!!!!!!!!!!!!

C:\Users\user1\Desktop\NIST>
```

Fig. 5.2. NIST test suite environment.

sequence of ones and zero pulses lasting for $t_{c,e} = tc + te$, is generated (Fig. 5.3). The average of $t_{c,e}$ is a constant through time for a ROSC in static environmental conditions. Their values follow a hypoexponential distribution, being the sum of the distributions of t_e and t_c, which are exponentials of constants τ_e and τ_c. By subsampling this new sequence at a sampling period equal to $< t_{c,e} >$, a highly random and de-biased sequence can be

obtained. This is similar to the approach taken in [4] in which, without applying de-biasing, a sampling frequency equal to $fs = \frac{2}{<\tau_c+\tau_e>}$ is used. In this project, since the de-biasing sequence doubles the duration of '1's and '0's when both time constants are equal, the sampling frequency used in the de-biased sequence is halved. By using this method, a bit rate equal to $Bit_rate_{subampling} = \frac{1}{<\tau_c+\tau_e>}$ is achieved. The sequence resulting of a subsampling of a de-biased sequence of consecutive $t_{c,e} = tc + te$ values, has been able to pass 14 out of the 15 NIST tests. The test that was failed was the universal test which evaluates the compressibility of the sequence and requires a great number of samples that was unable to be acquired within a single sequence.

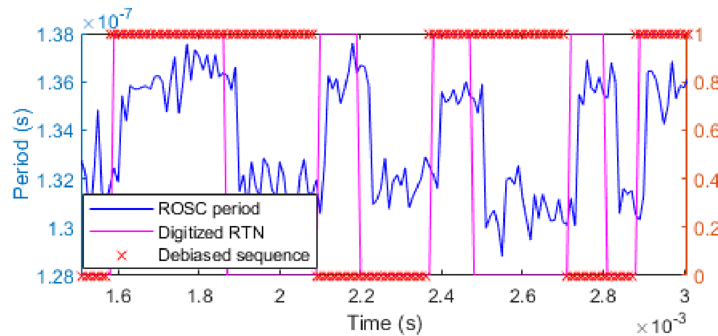


Fig. 5.3. Debiased $t_{c,e}$ sequence.

In order to increase bit rate and obtain longer sequences, a different approach is proposed to create a de-biased sequence. This new sequence will evaluate the difference between $t_{c,e}$ values by measuring the amount of ROSC periods it takes for each carrier to be captured and emitted. It is assumed that the lower bits of the resulting counter will provide higher level of randomness, since the average of $t_{c,e}$ will be a constant through time under static environmental and biasing conditions. The most significant bits are expected to be representative of the average of $t_{c,e}$ and mostly invariable between carrier capture-emission cycles, while the least significant bits will be significantly more variable between $t_{c,e}$ values, providing more randomness.

Table 5.1 shows the results of running the generated sequences through the NIST tests for different ROSCs in chip #1. For some ROSCs, not enough bits were obtained to run the universal test. However, experience has shown that if all other 14 tests are passed, most likely the universal test will be passed too. In cases where the universal test could not be run, only the remaining 14 have been taken into account. The sequences analysed show that the maximum number of bits that pass all the tests will vary depending on the ROSC and temperature. Increasing temperature (lower peltier current value) decreases the maximum number of bits that pass all the test. This, however, also provides higher rate of RTN. Increasing temperature by 16.5 degrees results in loosing 1 bit. At the same time, as seen

before, this temperature increase quadruples the event rate. This means that the bit rate still gets doubled under a 16.5 ºC increase in temperature. In fact, some correlation can be seen between the number of maximum usable bits and $< t_{c,e} >=< t_e + t_c >$. The trend shows that a lower $< t_{c,e} >$ value results in lower maximum usable bits. This makes sense, since the distribution of $t_{c,e} = t_e + t_c$ is the sum of two exponential distributions, so at larger average, a larger variation between values is present. In all studied circumstances, 5 bits are enough to pass all the tests.

Column	Row	Peltier current (A)	Number of bits	Passed tests	$< t_e + t_c >$	Bit rate
8	51	1	7	14/15	0,0003	
8	51	1	6	15/15	0,0003	20 kb/s
8	51	0	7	14/15	0,0001697	
8	51	0	6	15/15	0,0001697	35 kb/s
8	51	-1	7	11/15	0,000072	
8	51	-1	6	13/15	0,000072	
8	51	-1	5	15/15	0,000072	69 kb/s
38	57	0	7	11/15	0,000072	
38	57	0	6	11/15	0,000072	
38	57	0	5	15/15	0,000072	69 kb/s
19	22	0	7	14/15	0,0001999	
19	22	0	6	15/15	0,0001999	30 kb/s
43	42	0	7	13/14	0,0001274	
43	42	0	6	14/14	0,0001274	47 kb/s
47	14	0	8	12/14	0,000964	
47	14	0	7	14/14	0,000964	7 kb/s
47	30	0	10	12/14	0,00066	
47	30	0	9	14/14	0,00066	14 kb/s

Table 5.1 NIST test results of $t_{c,e} = te + tc$ count sequences for different ROSC in chip #1

5.1.2 TRNG performance evaluation

Two approaches have been considered for the application of TRNG: obtaining sequences form ROSC or directly observing single transistor's current variation due to RTN akin to [1].

To extrapolate the obtained data from ROSCs for the case of direct transistor current measurement, the following considerations have been taken into account: All observed ROSCs presented 2-level RTN corresponding to a single trap. This indicates that observed time constants were the results of a single transistor and could be applied to both ROSC and transistor implementations. Each ROSC contains 5 gates with 4 transistors each. Of these transistors, as previously mentioned, PMOS's RTN magnitude is so small that it gets masked. Additionally, of the two NMOS only one contributes to the RTN observable in the output waveform. So, the number of single measured transistors equivalent to a single ROSC would be 5. The transistor would still need to be switching in order to maintain AC excitation for increased RTN speed. On top of that, since more devices are needed, the time required in order to identify which ones present useful RTN would be multiplied by 5.

From the previously presented data, the numbers presented in Table 5.2 can be employed to elaborate on potential TRNG applications. To calculate the ROSC area, the space required for 1024 ROSCs, the decoder and the buffers has been taken into account. The total space of this circuitry in the ROSQUILLAS chip is $150\mu m \cdot 150\mu m = 0.075mm^2$ which, divided by the 1024 ROSCs, results in the value in the area per ROSC shown in the table. As for the current consumption, the consumption of the ROSCs at 0.25V and the buffering circuit powered at 1V have been taken into account. In the case of transistors, the consumption and space of the transistor array in the TURRON block of the same chip has been evaluated. All estimations have been based on the architecture of the studied ROSQUILLAS chip, other design structures would modify the results.

Parameter	Value	Parameter	Value
Area per ROSC	$73.2 \cdot \frac{\mu m^2}{ROSC}$	Area per transistor	$7.32 \cdot \frac{\mu m^2}{ROSC}$
PB0 ROSC current @ VCC=0.25V Buffering current @ VDD=1V	35nA/ROSC $4.87\mu A$/ROSC	trt current @ 0.25V, 1MHz	66 nA
$P(RTN\|PB0)_{ROSC}$	1.09 %	$P(RTN\|PB0)_{trt}$	0.218 %
mean($\frac{1}{<\tau_c+\tau_e>}$)	1219 Hz	$bit_rate_{subsampling}$	$1.2\frac{kbits}{s \cdot ROSC_{withRTN}}$
Max bits for NIST pass	5b	$bit_rate_{Period_Count}$	$6.1\frac{kbits}{s \cdot ROSC_{withRTN}}$

Table 5.2 Experimentally obtained technology specs.

Results have shown that smaller transistors are preferable for easier RTN extraction. By using PB0, 1.09 % of the ROSC will present detectable RTN. At the same time, this detected RTN will have an average capture and emission rate of 1.2 kHz. From these values, an approximated usable ROSC bit rate can be calculated. For the case of a de-biased sequence result of subsampling the $\tau_{c,e}$ sequence:

$$bit_rate_{subsampling} = 1bit \cdot mean(\frac{1}{< \tau_c + \tau_e >}) = 1.2 \frac{kbits}{s \cdot ROSC_{withRTN}} \tag{5.1}$$

If instead, the proposed approach of counting period values to generate the random sequence is implemented, a conservative estimation to obtainable bit rate can be performed. The 5 least significant bits of the period count were able to pass the NIST tests under all studied ROSCs. By assuming 5 bits can be extracted for all usable ROSCs, the estimated obtainable bit rate is the following:

$$bit_rate_{Period_Count} = 5 \cdot mean(\frac{1}{< \tau_c + \tau_e >}) = 6.1 \frac{kbits}{s \cdot ROSC_{withRTN}} \tag{5.2}$$

In the total bit rate calculus, to simplify analysis, it will be assumed that all ROSCs with usable RTN can be used in parallel for sequence generation. Table 5.3 shows a breakdown of the specs of a hypothetical RTN-based TRNG using this technology for three sequence obtention methods: Subsampling of a digitised $t_{c,e}$ sequence from ROSCs, the period measuring method previously mentioned, and a subsampling of a digitised $t_{c,e}$ sequence from direct transistor measurement. All obtained bit rate values are a rough estimate derived from the studied chip's specific structure, meaning that the displayed results are intended for a qualitative analysis rather than a quantitative one. These bit rate values, however, are calculated assuming nominal body bias values. The results could be improved significantly by applying a body bias calibration process to each ROSC in order to find their RTN hot spot. Previous results show that calibration can increase the bit rate of a ROSC with RTN up to a hundred times. In order to calculate the approximate power consumption and space required for the ROSCs and transistors, the total power and space required by the device and the buffering has been taken into account.

Results show a trade-off between ROSC and transistor implementations. Obtaining the $t_{c,e}$ sequence through direct transistor measurement offers better power and space performance. However, implementing the presented method of counting the number of periods between carrier capture and emission allows for heavy reduction of the number of ROSCs needed. By comparing this new method to the direct transistor measurement, a trade-off between power and space appears. Using transistors allows for using only the 1.7 % power consumption used by ROSCs, but using ROSCs the space required is reduced to 40% that of the transistors. In all cases, however, reaching hundreds of Mbits/s becomes a challenge. The space required and power consumption are too significant to become viable.

Method (All devices)	ROSC Subsampling	ROSC Period count	trt Subsampling
Bit rate/device$_{withRTN}$	$1.22\frac{kbits}{s \cdot ROSC_{withRTN}}$	$6.1\frac{kbits}{s \cdot ROSC_{withRTN}}$	$1.22\frac{kbits}{s \cdot trt_{withRTN}}$
P(RTN|PB0)	1.09 %	1.09 %	0.218 %
Devices for 10kbits/s	753	151	3763
Space for 10kbits/s	$0.055mm^2$	$0.011mm^2$	$0.0275mm^2$
Power for 10kbits/s	$39.4\mu W$	$7.9\mu W$	$135nW$
Devices for 1Mbits/s	$75 \cdot 10^3$	$15 \cdot 10^3$	$376 \cdot 10^3$
Space for 1Mbits/s	$5.5mm^2$	$1.1mm^2$	$2.8mm^2$
Power for 1Mbits/s	3.94 mW	0.79 mW	13.5 μW
$Devices_{with_RTN}$ for 1Mbits	820	164	820

Table 5.3 Calculated TRNG specs

Another possible TRNG implementation would be to try and combine the methods of direct transistor measurement with the time constant evaluation done through ROSC period counting. By doing this, both power consumption and space could improve. To do so, an external oscillator at a fixed frequency could be employed in order to quantify the capture and emission times of the trap in the transistor. Approximated calculation of the bit rate obtainable through this method is possible, but only under the assumption that the 5 bits/device determined by passing the NIST tests with the period value counting method is still valid for this case. Further testing would be required in order to ensure such an assumption is correct. The results of this analysis are shown in Table 5.4.

Method	trt τ evaluation
Bit rate/device$_{withRTN}$	$6.1\frac{bits}{s \cdot trt_{withRTN}}$
P(RTN|PB0)	0.218 %
Devices for 10kbits/s	753
Space for 10kbits/s	0.0055 mm^2
Power for 10kbits/s	27 nW
Devices for 1Mbits/s	$75.3 \cdot 10^3$
Space for 1Mbits/s	$0.55mm^2$
Power for 1Mbits/s	2.7 μW
$Devices_{with_RTN}$ for 1Mbits	164

Table 5.4 Transistor-based time constant evaluation TRNG specs

In all studied cases, however, the space required only takes into account the devices themselves. To reach the order of 1Mbits/s, 163 devices are required to provide random sequences in parallel for the case of time constant evaluation. To do so, the sequence extraction circuitry would also be multiplied by 164, heavily increasing the space required. One way to reduce it would be to only select the devices with fastest RTN to decrease the total number of ROSC to measure. The maximum bit rate found for a single ROSC was 69 kbits/s, meaning only 15 ROSC with the fastest RTN could reach the order of 1Mbits/s. The only drawback would be that the total size of the device array would increase about 3.8 times since the probability of having a ROSC with those characteristics is only 0.035 %. Table 5.5 shows the results of such filtering on minimum device array requirements on the period count method for both types of device array.

Method ($\tau_c + \tau_e < 0.0001s$)	ROSC Period count	trt Period count
Bit rate/device$_{withRTN}$	$50\frac{kbits}{s \cdot ROSC_{withRTN}}$	$50\frac{kbits}{s \cdot trt_{withRTN}}$
P(RTN|PB0)	0.035 %	0.007 %
Devices for 1Mbits/s	$57 \cdot 10^3$	$285 \cdot 10^3$
Space for 1Mbits/s	4.17 mm^2	2.1 mm^2
Power for 1Mbits/s	96 μW	330 nW
$Devices_{with_RTN}$ for 1Mbits	20	20

Table 5.5 Calculated TRNG specs for filtering of devices with fastest RTN

The RTN phenomena is slower than other sources of entropy for TRNG implementation. However, it has shown robustness to attacks which makes it an attractive solution for applications where that is an special concern. Supply voltage attacks could modify the rate at which events occur, however, it will not modify the presence of a trap. By implementing the TRNG through FDSOI technology, the effect of a supply voltage modification can be compensated as long as that change is within the body bias range. One added vulnerability of implementing this RTN-TRNG by using ROSCs is that if the supply voltage increases enough, the variation due to RTN ends up being masked by jitter and if the supply voltage is low enough and the ROSC stops oscillating, no RTN can be extracted either.

Temperature attacks are usually performed by increasing temperature. In the case of RTN, that would only improve the bit rate. One case to note is the possibility that the RTN rate increases so much that it cannot be sampled appropriately. However, this could also be remedied by modifying the body bias of the transistor to a manageable RTN event rate through calibration.

Body bias tuning generally serves as a tool to improve resistance to attacks and improve bit rate. However, it can also be a potential point of attack. If the body bias voltage is modified, it could bring an individual ROSC out of the zone of maximum RTN. This could bring the bit rate down considerably, even though RTN won't fully disappear. Yet, in the case that all ROSCs of the proposed TRNG share the same body bias voltage, since the zone of maximum RTN is random for each ROSC, by modifying body bias some ROSC will decrease their bit rate while others will have it increased instead.

One point of concern observed that arises from the use of ROSCs is a possible phase locking. In fact, one such case has been presented that deteriorated the results under specific bias conditions. If a signal is injected into a ROSC that creates a period variation of magnitudes similar to the period value jumps due to RTN, RTN could end up masked and, if the injected signal is considered by the system as RTN, an arbitrary output sequence could be produced.

5.2 RTN in FDSOI for PUF applications

RTN in the case of PUF can either decrease reliability of PUFs that do not employ it, in which case it needs to be minimised, or it can be used in order to implement RTN-based PUF. For the first case, a ROSC-based PUF in which frequency differences due to the manufacture process are compared to create a bit, RTN can decrease reliability by making temporary and unpredictable shifts in ROSC frequency relationship if the two ROSC frequency values are too close together. Experiments results have shown that for high supply voltages, above VDD=0.3V, the effect of RTN is minimised to such a degree that it became undetectable since it produced variations that were masked by other noise sources such as jitter. This means that one way to prevent the effect of RTN would be to work at higher voltages, such as 0.35V or 0.4V which are still fairly low for IoT applications. When comparing the frequency results, however, an advantage of operating at lower supply voltages, besides consuming less, appears. The ratio between the frequency standard deviation and its mean increases as frequency decreases. As an example, for the case of PB16 in VCC=0.25V, the ratio is R=std/freq=0.257, while for PB0 in VCC=0.3V, it decreases to R=0.21. This leads to the conclusion that the supply should be as low as possible for improved frequency comparison but without decreasing enough that RTN starts to take significant effect.

Another solution would be to preemptively identify which ROSCs present RTN and exclude them. Results have shown that around 3.3% of ROSCs present some sort of RTN, so no great loss would be made by excluding them. To detect that 3.3% of ROSCs, an observation of the ROSC period changes through time would be necessary but FDSOI technology could

be employed in order to reduce observation time. Multiple bias points could be observed during significantly less time, as a device with a trap could be identified more clearly under different bias conditions for which an RTN hotspot is present. The total time needed for masking out these ROSCs would still be significant but would be lessened.

In the case of implementing an RTN-based ROSC, one approach would be to compare ROSC RTN speed like in [11]. This can be done by using counters that evaluate how many events are detected within a time window and comparing the result of the counters to generate one bit. If N ROSCs are found to have RTN, the total number of potential challenge response pairs becomes $CRP = \frac{N(N-1)}{2}$. When comparing with a general ROSC-based PUF like the one just mentioned, the total number of CRP is the same expression but N is the total number of ROSCs. This means that to obtain the same number of CRP as a ROSC-PUF of 1024 ROSCs, the RTN-based PUF would need 93945 ROSCs. The space required increases significantly for RTN-PUF, however, by activating only ROSC that were found to have RTN, power consumption would be about the same. One way to increase the number of challenge-response pairs, and reduce space, is to introduce the FDSOI biasing conditions as part of the challenge to compare either between ROSCs or within a single ROSC. Since the bias conditions for the hotspot are random, the time constants under different bias conditions could be compared to provide a random bit in the same way.

RTN-PUF requires a lot of space when comparing with other PUFs. It seems to be better suited for a weak PUF than for a strong PUF. Weak PUFs used for key generation provide less CRP and require high reliability and stability. The main advantage to using RTN is its stability and robustness, as stated by [11]. This is thanks to the great variation between time constant values. Ranging from 0.1 ms to 300 ms and in an expected logarithmically uniform distribution, the values the time constants can take is of 3 orders of magnitude. This is much greater than the frequency variation due to process variation which, for reference, in this technology's case and for the working frequencies employed, the highest frequency value for a PB0 ROSC is only 5 times higher than the minimum frequency found and most ROSCs have a frequency value around the average. As stated before, supply voltage variations can be somewhat compensated by using the body biasing capabilities of FDSOI as long as the voltage difference is known. Results have shown that temperature differences in the range of 16,5ºC affect all ROSCs equally and, thanks to the great separation between ROSC time constants, should not modify the response of the CRP.

6. Conclusions and future work

Through the making of this project, the objective of characterising the chip under test has been fulfilled. In doing so, the behaviour of RTN in FDSOI-based ROSCs has been evaluated. The data obtained included the frequency of the ROSCs, the probability for a ROSC to have RTN and to have useful RTN, how does modifying body bias modify the speed at which RTN occurs, the effect of a supply voltage variation and the effect of modifying temperature.

From the identified ROSC that had useful RTN, random sequences have been extracted and evaluated by applying the NIST test suite to assess the potential use of RTN as an entropy source for security primitives. To do so, an algorithm has been developed that identified changes in ROSC period due to RTN within a sequence. From these detected events, the statistics that define the RTN for each sequence have been obtained and compared for multiple ROSCs under multiple biasing and environmental conditions.

Results have shown that FDSOI's ability of controlling body bias can greatly effect the speed of RTN and can be used in order to maximise or minimise its effect as well as to somewhat compensate temperature and supply voltage variations. A certain body bias change will affect every ROSC in a different way since it will depend on RTN trap location, which is random. The difference found between minimum and maximum RTN event rate for a given device varying only body bias are between 2 times more and 95 times more.

Transistor size is of great importance in the detection of RTN. For larger transistors, the effect of a single trap gets easily masked so if RTN wants to be avoided, it is recommended that larger transistors are used. For reference, 3.64% of ROSCs that had PB0 have been detected to present some kind of RTN whilst for PB16, that value has decreased to 1.74%. On the other hand, if RTN wants to be maximised in order to be harvested, smaller transistors presenting clear and easily identifiable RTN is preferred. For those cases, 1.09% of all PB0 ROSC have presented usable RTN with clear period value jumps and time constants close enough for reliable detection.

By counting the number of periods it takes for a ROSC to have a period value change due to RTN, unbiased random sequences can be generated. Through experimentation, a single ROSC can provide random sequences of up to 69 kbits/s. Using these sequences for TRNG purposes, by selecting only the devices with fastest RTN, the bit rate can be brought to roughly $bit_rate_{Period_Count} = 50\frac{kbits}{s \cdot ROSC_{withRTN}}$. In order to reach bit rate values competitive with current TRNG applications, the minimum number of ROSC needed rises to 57000 ROSC, reaching the order of 1Mbits/s. This bit rate can be achieved by extracting random sequences in parallel of all statistically expected 20 ROSC with RTN time constant sum below 100 μs. Another option presented is the evaluation of RTN time constants by

direct measurement of single transistor's current, which could provide improved power and reduced space to reach the same bit rate. In order to reach 1Mbits/s, approximately $285 \cdot 10^3$ transistors would be needed, occupying $2.1mm^2$ and requiring $330nW$. This is achieved by extracting sequences in parallel of around 20 transistors.

RTN as a randomness source for PUFs based in comparing RTN time constants can provide high reliability thanks to the great range of values those can take. The time constants analysed took values mostly between 0.1 ms and 300 ms, bringing the difference to 3 orders of magnitude. Still, when compared to other PUF designs that do not employ RTN such as ROSC-based PUFs, the amount of ROSC needed to reach the same amount of bits is approximately 100 times larger. However, an added benefit of using FDSOI is that the biasing conditions can be introduced into the challenge for an increase of CRP.

Future work derived from the results obtained in this project would be to propose a structure of TRNG that uses RTN in FDSOI technology and to evaluate how much can the bit rate be improved by using such a technology either by introducing calibration or environmental effect compensation by body bias control. Another possible next step is to keep obtaining measurements for more ROSC and chips in order to gather more data and use the experimentally obtained modelling of RTN to create an emulator. This could then be used to aid in circuit design by accounting for RTN effect when simulating.

REFERENCES

[1] James Brown et al. "A low-power and high-speed True Random Number Generator using generated RTN". In: *2018 IEEE Symposium on VLSI Technology*. IEEE. 2018, pp. 95–96.

[2] James Brown et al. "Random-telegraph-noise-enabled true random number generator for hardware security". In: *Scientific reports* 10.1 (2020), pp. 1–13.

[3] Xiaoming Chen et al. "Modeling random telegraph noise as a randomness source and its application in true random number generation". In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 35.9 (2015), pp. 1435–1448.

[4] Zhigang Ji and Jianfu Zhang. "Understanding Generated RTN as an Entropy Source for True Random Number Generators". In: *2021 International Conference on IC Design and Technology (ICICDT)*. IEEE. 2021, pp. 1–4.

[5] Mulong Luo et al. "Impacts of random telegraph noise (RTN) on digital circuits". In: *IEEE Transactions on Electron Devices* 62.6 (2014), pp. 1725–1732.

[6] Carlos Marquez et al. "Electrical characterization of Random Telegraph Noise in Fully-Depleted Silicon-On-Insulator MOSFETs under extended temperature range and back-bias operation". In: *Solid-State Electronics* 117 (2016), pp. 60–65.

[7] Takashi Matsumoto, Kazutoshi Kobayashi, and Hidetoshi Onodera. "Impact of random telegraph noise on CMOS logic circuit reliability". In: *Proceedings of the IEEE 2014 Custom Integrated Circuits Conference*. IEEE. 2014, pp. 1–8.

[8] Andrew Rukhin et al. *A statistical test suite for random and pseudorandom number generators for cryptographic applications*. Tech. rep. Booz-allen and hamilton inc mclean va, 2001.

[9] Qianying Tang and Chris H Kim. "Characterizing the impact of RTN on logic and SRAM operation using a dual ring oscillator array circuit". In: *IEEE Journal of Solid-State Circuits* 52.6 (2017), pp. 1655–1663.

[10] Christoforos G Theodorou et al. "New LFN and RTN analysis methodology in 28 and 14nm FD-SOI MOSFETs". In: *2015 IEEE international reliability physics symposium*. IEEE. 2015, XT–1.

[11] Motoki Yoshinaga et al. "Physically unclonable function using RTN-induced delay fluctuation in ring oscillators". In: *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE. 2016, pp. 2619–2622.

[12] Jibin Zou et al. "New insights into AC RTN in scaled high-K/metal-gate MOSFETs under digital circuit operations". In: *2012 Symposium on VLSI Technology (VLSIT)*. IEEE. 2012, pp. 139–140.

APPENDIX

Glossary

- **RTN:** Random Telegraph Noise.
- **TRNG:** True Random Number Generator.
- **PUF:** Physical Unclonable Function.
- **ROSC:** Ring Oscillator.
- **FDSOI:** Fully Depleted Silicon on Insulator Technology for transistors.
- **NIST:** National Institute of Standards and technology. Developed the NIST test suite that evaluates the randomness of a binary sequence.
- **ROSQUILLAS:** Chip Under Study. Developed under the VIGILANT project.
- **ROSCON:** ROSC array within ROSQUILLAS chip.
- **TURRON:** Transistor array within ROSQUILLAS chip.
- Ids**:** Transistor drain current.
- Vth**:** Threshold voltage.
- **PSD:** Power spectral density.
- τ_c**:** Carrier capture time constant. Fits the distribution of capture times.
- τ_e**:** Carrier emission time constant. Fits the distribution of emission times.
- t_c**:** Carrier capture time. time it takes for a carrier to be captured in the oxide trap.
- t_e**:** Carrier emission time. time it takes for a carrier to be released from the oxide trap.
- $t_{c,e}$**:** Time for a carrier to be captured and released.
- $VCCS$**:** ROSC PMOS body bias
- $GNDS$**:** ROSC NMOS body bias
- VCC**:** ROSC transistor supply voltage
- VDD**:** Chip supply voltage. Independent from VCC.

- PB**:** Poly Bias. Determines transistor size.
- DSO**:** Digital Storage Oscilloscope.
- SMU**:** Source Measure Unit.
- *Event***:** Detected ROSC period value jump due to RTN.
- ROW**:** Row position of a ROSC within a chip
- COL**:** Column position of a ROSC within a chip