Czech Technical University in Prague

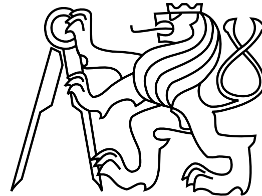
Faculty of Electrical Engineering

Doctoral Thesis

August 2018

Ing. Vlastimil Kotě

Czech Technical University in Prague
Faculty of Electrical Engineering
Department of Microelectronics



Design of True Random Number Generators Suitable for Integrated Circuits

Doctoral Thesis

Ing. Vlastimil Kotě

Prague, August 2018

Ph.D. Programme: P 2612 Electrical Engineering and Information Technology
Branch of study: 2612V015 Electronics
Supervisor: doc. Ing. Jiří Jakovenko, Ph.D.

Declaration

I declare I have completed my doctoral thesis on my own with the contribution of my supervisor and consultants. I used only materials (literature, projects, articles) specified in the attached list.

I agree with the use of the information presented in my doctoral thesis pursuant to Copyright Act 121/2000 Coll., Sec. 60.

In Prague, August the 31st, 2018 ....

Vlastimil Kotě

Acknowledgements

I would like to thank my supervisor doc. Ing. Jiří Jakovenko, Ph.D. for the professional guidance, valuable comments and especially for willingness and patience in consultation. I would also like to thank my colleagues from CTU in Prague and company STMicroelectronics for their patience and valuable expertise they have provided me. Special thanks to Ing. Patrik Vacula, Ing. Vladimír Molata, and Ing. Adam Kubačák for their ideas, consultations, and great cooperation. And last but not least, I would like to thank my wife Michaela, my daughter Anastázie, my son Vlastimil, and the whole family for their overall sustained support during the time of my work.

Abstract

This doctoral thesis deals with the design of true random number generators (TRNGs) suitable for integrated circuits (ICs). These devices exploit randomness of physical phenomena, in this case, noises in semiconductors structures. The first proposal is a TRNG with time multiplexed metastability-based sources of randomness using the principle of pipelining. Digitized random signals coming from sources of randomness are interleaved in a time multiplexer, which allows increasing output data rate. This TRNG has been fabricated in a 130 nm HCMOS9GP bulk CMOS technology from STMicroelectronics, occupies an area of 0.029 mm^2, and can operate in changing environmental conditions. Random bit sequences can be generated up to the data rate of 20 Mb/s without any corrective mechanisms while its power consumption is 72.48 μW at temperature of 25 °C.

Another proposal is enhanced generic architecture of TRNGs, which can detect attempts of attackers trying to manipulate with properties of random number sequences. Protective mechanisms detecting sudden changes in sequence properties are based on features of the von Neumann corrector and an approximately entropy estimation. A presented reconfigurable noise source can produce random sequences in two settings between which is switched when entropy decreases. These circuits have been designed in a 130 nm HCMOS9A bulk CMOS technology also from STMicroelectronics.

Simulations of the presented TRNGs are very time-consuming. Therefore, for verification of systems containing these generators, their behavioral models have been created in the Verilog-A HDL. They approximate properties of the TRNGs and are able to shorten duration of the simulations to allow revealing possible errors of systems.

The developed TRNGs are analog and mixed-signal (AMS) circuits containing very sensitive parts. Their precise handmade physical design is very time-consuming. Hence new methodology steps of the physical design have been proposed. Introduced functions can automatically sort electrical devices according to their topological, structural and electrical properties, control layout objects without filling forms, or search an IC design database based on similarity of object properties. They speed up the physical design, help to prevent errors and make the design more robust.

Keywords: True random number generator, integrated circuit, source of randomness, behavioral model, physical design of analog and mixed-signal circuits.

Abstrakt

Tato doktorská práce se zabývá návrhem generátorů pravých náhodných čísel (TRNG) vhodných pro integrované obvody (IC). Tato zařízení využívají náhodnost fyzikálních jevů, v tomto případě šumů v polovodičových strukturách. Prvním návrhem je TRNG s časově multiplexovanými metastabilními zdroji náhodnosti využívající tzv. princip řetězení. Digitalizované náhodné signály produkované zdrojem náhodnosti jsou prokládány v časovém multiplexoru, který umožňuje zvýšení výstupní datové rychlosti. Tento TRNG byl vyroben ve 130nm CMOS technologii nazvané HCMOS9GP od společnosti STMicroelectronics na ploše 0,029 mm^2 a je schopen pracovat v proměnných podmínkách. Sekvence náhodných bitů mohou být generovány až do datové rychlosti 20 Mb/s bez použití opravných mechanizmů při výkonové spotřebě 72,48 µW a teplotě 25 °C.

Dalším návrhem je rozšířená obecná architektura TRNG, jenž je schopná detekovat pokusy útočníků snažících se manipulovat s vlastnostmi sekvencí náhodných čísel. Ochranné mechanismy detekující náhlé změny vlastností těchto sekvencí jsou založeny na charakteristikách von Neumannova korektoru a na přibližném odhadu entropie. Uvedený rekonfigurovatelný zdroj šumu je schopen produkovat náhodné sekvence ve dvou nastaveních, mezi nimiž je přepínán při poklesu entropie. Tyto obvody byly navrženy ve 130nm CMOS technologii nazvané HCMOS9A od společnosti STMicroelectronics.

Simulace prezentovaných TRNG jsou velmi časově náročné. Proto pro verifikace systémů obsahujících tyto generátory byly vytvořeny v jazyce Verilog-A HDL jejich behaviorální modely. Napodobují vlastnosti TRNG a jsou schopné zkrátit dobu trvání simulací, aby bylo možné odhalit případné chyby systému.

Vyvinuté TRNG jsou analogové a smíšené (AMS) obvody obsahující velmi citlivé části. Jejich precizní ručně vytvářený fyzický návrh je velmi časově náročný. Proto byly navrženy nové metodologické kroky fyzického návrhu. Zavedené funkce umí automaticky třídit elektronické součástky podle jejich topologických, strukturních a elektrických vlastností, ovládat databázové objekty bez vyplňování formulářů nebo prohledávat databáze na základě podobnosti databázových objektů. Tyto funkce zrychlují fyzický návrh, pomáhají předcházet chybám a dělat design robustnější.

Klíčová slova: Generátor pravých náhodných čísel, integrovaný obvod, zdroj náhodnosti, behaviorální model, fyzický návrh analogových a smíšených obvodů.

Contents

List of Acronyms XV

List of Symbols XVII

List of Figures XXI

List of Tables XXV

1 Introduction 1

1.1 Organization of This Thesis . . . 3

1.2 Author's Scientific Contributions . . . 3

1.3 State of the Art . . . 5

1.4 Solution Methods of the Work . . . 12

2 TRNG Fundamentals 15

2.1 Generic Architecture . . . 16

2.1.1 Noise Source . . . 17

2.1.2 Digitizer . . . 17

2.1.3 Post-processing Block . . . 19

2.1.4 Output Interface . . . 25

2.2 TRNGs with Direct Noise Amplification . . . 25

2.2.1 Source of Randomness Based on Thermal Noise of Resistors . . . 26

2.2.2 Direct Amplification of Noise in Semiconductor Junction . . . 27

2.3 TRNG Based on Ring Oscillators . . . 29

2.4 Metastability-Based TRNG . . . 31

2.5 TRNG Based on Chaos 34
2.6 Noises Occurring in Electronic Circuits 37
2.6.1 Thermal Noise 38
2.6.2 Flicker Noise 39
2.6.3 Shot Noise 40
2.6.4 Generation-Recombination Noise 41
2.6.5 Random Telegraph Signal Noise 41
2.6.6 Avalanche Noise 42

3 Evaluation of Random Number Sequences 43
3.1 NIST Test Suite 44
3.1.1 Monobit Test 45
3.1.2 Frequency Test within a Block 46
3.1.3 Runs Test 46
3.1.4 Test for the Longest Run of Ones in a Block 46
3.1.5 Binary Matrix Rank Test 46
3.1.6 Discrete Fourier Transform Test 47
3.1.7 Non-overlapping Template Matching Test 47
3.1.8 Overlapping Template Matching Test 47
3.1.9 Maurer's "Universal Statistical" Test 47
3.1.10 Linear Complexity Test 48
3.1.11 Serial Test 48
3.1.12 Approximate Entropy Test 48
3.1.13 Cumulative Sums Test 48
3.1.14 Random Excursions Test 49
3.1.15 Random Excursions Variant Test 49
3.2 FIPS Test Suite 49
3.2.1 Monobit Test 50
3.2.2 Poker Test 50
3.2.3 Runs Test 50
3.2.4 Long Runs Test 51
3.3 Shannon Entropy 51

4 TRNG with Time Multiplexed Sources of Randomness **53**

4.1 Principle of Time Multiplexed TRNG 54

4.2 Circuit Implementation 56

4.2.1 Noise source 58

4.2.2 Digitizer 72

4.2.3 Time Multiplexer 73

4.3 Measurement Results 77

4.3.1 Arrangement of Measuring Instruments 77

4.3.2 Evaluation of Generated Random Number Sequences 78

4.3.3 Current Consumption 84

4.3.4 Required Energy per Random Bit 85

4.4 Comparison 87

5 Protective Mechanisms for TRNGs **89**

5.1 Enhanced Generic Architecture 90

5.2 Protective Mechanisms 92

5.2.1 Attack Detector 92

5.2.2 Low Entropy Detector 96

5.3 Reconfigurable Source of Randomness 99

5.3.1 Reconfigurable Noise Source 100

5.3.2 Differential Digitizer 105

5.3.3 Power Supply 108

5.4 Achieved Results 109

5.5 Future Work on Development of TRNGs 112

5.5.1 EGA with Time Multiplexed Sources of Randomness 113

5.5.2 Noise source with Automatic Zeroing 113

6 Behavioral Models of TRNGs **117**

6.1 Behavioral Model of TRNG with Time Multiplexer 118

6.1.1 Description of Model 119

6.1.2 Properties of Model 122

6.2 Behavioral Model of EGA 124

6.2.1 Structure of Model 124

6.2.2 Evaluation of Model 126

7 New methodology steps of physical design of AMS ICs 129

7.1 Automated Pre-placement Phase 130

7.1.1 Definition of Rules 131

7.1.2 Algorithm Implementation 137

7.1.3 Productivity Improvement 141

7.2 Incremental Control of Layout Objects 145

7.2.1 Implementation in CAD Environment 146

7.2.2 Productivity Gain 147

7.3 Search for Objects based on Their Similarities 148

7.4 Classification of Matched Structures 150

7.5 Future Work on Physical Design Methodology 150

8 Conclusions 151

References 155

Appendix i

List of Author's Publications iii

A.1 Publications Related to the Topic of This Work iii

A.1.1 Publications in Impacted Journals iii

A.1.2 Publications in Reviewed Journals iii

A.1.3 Publications Excerpted by WoS iv

A.1.4 Other Publications iv

A.1.5 Functional Samples v

A.2 Publications Not Related to the Topic of This Work v

A.2.1 Patents v

A.2.2 Publications Excerpted by Scopus vi

A.2.3 Other Publications vi

A.2.4 Functional Samples vi

Recognition and Review vii

List of Acronyms

AMS Analog and Mixed-Signal

ApEn Approximate Entropy

ASIC Application Specific Integrated Circuit

BCD Bipolar-CMOS-Double Diffused MOS

BJT Bipolar Junction Transistor

CAD Computer-Aided Design

CMOS Complementary Metal-Oxide-Semiconductor

DC Direct Current

DRC Design Rule Check

DRM Design Rule Manual

EGA Enhanced Generic Architecture

ESD Electrostatic Discharge

FIB Focused Ion Beam

FinFET Fin Field Effect Transistor

FIPS Federal Information Processing Standards

FPGA Field Programmable Gate Array

FPPM Final Pre-Placement Matrix

GR Generation-recombination

HDL Hardware Description Language

IC	Integrated Circuit
ICDB	Incremental Control Database
IoT	Internet of Things
LDO	Low Dropout Voltage Regulator
LFSR	Linear Feedback Shift Register
LVS	Layout Versus Schematic
MOSFET	Metal-Oxide-Semiconductor Field Effect Transistor
NBTI	Negative Bias Temperature Instability
NIST	National Institute of Standards and Technologies
PBTI	Positive Bias Temperature Instability
PC	Personal Computer
PDF	Probability Density Function
PRNG	Pseudo-Random Number Generator
PSD	Power Spectral Density
RMS	Root Mean Square
RNG	Random Number Generator
RO	Ring Oscillator
RTS	Random Telegraph Signal
SCR	Silicon Controlled Rectifier
SoC	System on Chip
SOI	Silicon On Insulator
TRNG	True Random Number Generator
VAM	Virtual Analog Matrix
VCO	Voltage Controlled Oscillator
VDM	Virtual Digital Matrix
VHDL	Very-High-Speed Integrated Circuit Hardware Description Language
XOR	Exclusive Or

List of Symbols

a_{a}	Offensive signal (V)
b	Bias of random number generation (-)
e	Coefficient of resilient function (-)
d	Binary derivative (b)
$d_{\mathrm{A,B}}$	Relative difference between t_{A} and t_{B} (%)
f	Frequency (Hz)
g	Random number sub-sequence (V)
g_{ds}	Output conductance of the MOSFET small signal model (S)
g_m	Transconductance (AV^{-1})
h	Coordinate of resilient function (-)
i	Normalized period (-)
j	Order of ring oscillator (-)
k_{B}	Boltzmann constant ($1.38064852 \cdot 10^{-23}\,\mathrm{JK}^{-1}$)
l	Normalized period (-)
m	Length of bit sequence (b)
n	Number of source of randomness (-)
o	Occurrences
p	Probability
q	Identifier of source of randomness (-)

q_e	Elementary charge ($q = 1.60217662 \cdot 10^{-19}$ C)
r	Internal random numbers (V)
r_{ds}	Output resistance of the MOSFET small signal model (Ω)
r_{out}	External random numbers (V)
s	Digitized noise signal (V)
t	Time (s)
x_{t}	Sample test statistic value
z	Constant of resilient function (-)
z_{t}	Test statistic
A	Amplification (-)
B	Random data rate (bit/s)
C	Capacitance (F)
E	Energy (J)
F	Resilient function
G	Conductance (S)
H	Entropy (b)
I	Current (A)
K_{f}	Coefficient of flicker noise for MOSFETs (C^2/m^2)
$K_{\text{f},R}$	Material-dependent coefficient of flicker noise for resistors (Sm^2)
K_{rts}	Coefficient of random telegraph signal noise (-)
L	Length (m)
M	Input parameter of the NIST test suite (b)
N	Quantity (-)
O	Order of binary derivative (-)
P	Power (W)
Q	Input parameter of the NIST test suite (b)
R	Resistance (Ω)

S	Spectral density (W/Hz)
T	Temperature (K)
V	Voltage (V)
W	Width (m)
X	Random bit
Y	Random bit
Z	Random bit
α	Significance level (-)
γ	Technology dependent coefficient of thermal noise (-)
ζ	Skew tent map coefficient (-)
η	Coefficient of flicker noise (-)
ι	Number of bits (-)
κ	Constant of metastable circuit (-)
μ	Mean value
μ_{m}	Mobility ($\mathrm{m^2V^{-1}s^{-1}}$)
ν	Analog noise signal (V)
ξ	Number of logic gates (-)
ρ	Relative ratio (-)
σ^2	Variance
τ	Delay (s)
υ	Base of logarithm (-)
Θ	Period (s)
Ψ	Periodic square waveform signal (V)

List of Figures

2.1 The generic architecture of a TRNG [97] 16
2.2 The digitization model with the clocked comparator 18
2.3 The digitization model composed of digital circuits 18
2.4 The block diagram of the TRNG exploiting thermal noise of resistors [15] 26
2.5 The principle of VCO controlled sampling [15] 27
2.6 The block diagram of the TRNG with two independent noise sources [25] 28
2.7 The noise source based on noise occurring in the reverse polarized P-N junction [25] 28
2.8 The source of randomness based on a architecture with ring oscillators [27] 29
2.9 The RO output signal $\Psi_j(t)$ with the jitter displayed 30
2.10 The mechanical analogy of metastable systems 32
2.11 Waveforms of signals in the metastability based noise source [41] 33
2.12 The noise source exhibiting the metastable behavior [98] 34
2.13 The source of randomness based on the discrete time non-linear dynamical system [58] 35
2.14 The noise source as the CMOS compatible current mode circuit published in [58] 36
2.15 The transient simulation of the noise source published in [58] 37
2.16 The phase portrait plot of the noise source published in [58] 37

4.1 Principle illustration of the proposed TRNG [4] 55
4.2 The block diagram of the designed TRNG with time multiplexed sources of randomness [4] 57

4.3 The schematic diagram of the metastability-based noise source [4] 58
4.4 Waveforms of signals inside the designed TRNG simulated by the Mentor Eldo simulator at the transistor level [4] 59
4.5 The schematic diagram of the proposed metastable element used for offset analysis 60
4.6 The small signal model of the metastable element 61
4.7 The simplified small signal model of the metastable element used for gain calculation 61
4.8 The schematic diagram of the designed digitizer [4] 72
4.9 The layout of the metastable element connected to input parts of the digitizers with a drawn symmetry axis 73
4.10 The schematic diagram of the time multiplexer [4] 74
4.11 Waveforms illustrating the function of the time multiplexer [4] 74
4.12 The layout of the multi-project test chip without displayed tiles for better layer planarization 75
4.13 Photo of the fabricated die. The proposed TRNG containing all described parts – the noise sources, the digitizer, the time multiplexer, and the output buffer – occupies the marked area [4] 76
4.14 One of the smallest emblems of the Czech Technical University in Prague with the width of 59.55 µm and the heigth of 44.85 µm is present on the fabricated chip 76
4.15 Diagram of the measuring instrument arrangement 77
4.16 The approximate entropy of output random number sequences generated in the temperature range of −35 °C to 85 °C [4] 84
4.17 The current consumption of the designed circuit measured in the temperature range of −35 °C to 85 °C [4] 85
4.18 The energy per random bit of the designed TRNG calculated in the temperature range of −35 °C to 85 °C 87

5.1 The block diagram of the proposed EGA 91
5.2 The model of the side channel attack aimed at the noise source 93
5.3 Relation between B_{VN} and B_{IR} depending on the bias b 94

5.4 The principle schematic diagram of the valid bit generator without buffers for proper timing . . . 95
5.5 The simplified schematic diagram of the reconfigurable noise source . . . 101
5.6 Waveforms of signals inside the reconfigurable noise source simulated by the Mentor Eldo simulator at the transistor level . . . 102
5.7 The layout of the reconfigurable noise source with a drawn symmetry axis 103
5.8 The schematic diagram of the differential digitizer with the check of complementarity . . . 106
5.9 Waveforms of signals inside the differential digitizer simulated by the Mentor Eldo simulator at the transistor level. . . . 107
5.10 The block diagram of the TRNG connecting the EGA with time multiplexed sources of randomness . . . 114

6.1 Structure of the model of the source of randomness . . . 119
6.2 Waveforms of signals generated by the behavioral model of the TRNG with time multiplexed sources of randomness simulated by transient analysis of the Mentor Eldo simulator . . . 120
6.3 Waveforms of signals generated by the behavioral model of the TRNG based on the EGA simulated by the Mentor Eldo simulator . . . 125

7.1 Physical design flow enhanced by the pre-placement phase [8] . . . 131
7.2 Symbolic view of layout instances before (a) and after (b) the automated pre-placement phase [8] . . . 132
7.3 Cross sections of usually used devices in CMOS and BCD designs: (a) Junction insulated N-channel MOSFET. (b) PNP BJT. (c) Polysilicon – N-well capacitor. (d) Polysilicon resistor with a junction insulated bulk terminal. (e) Junction insulated N+ – P-well diode [8] . . . 133
7.4 Parasitic thyristor structures in CMOS or BCD designs: (a) Example of a frequently used circuit – an inverter. (b) Cross section with depicted parasitic components. (c) Schematic diagram of the parasitic thyristor structure. (d) Solution eliminating the parasitic thyristor structure. (e) Schematic diagram of the parasitic structure without thyristor configuration [8] . . . 135

7.5 The flow chart of the automated pre-placement phase [8] 138
7.6 VAM filling with outlined way of indexing [8] 140
7.7 The process of the wire width modification with the shown status bar . . . 145
7.8 The flow chart of the incremental control of layout objects [9] 147
7.9 Use of the similar search in the schematic editor (a) and layout editor (b) . 149

List of Tables

2.1 The truth-table of the XOR operation 21

2.2 Examples of the mean value calculated using the formula 2.16 22

2.3 The table describing the basic function of the von Neumann corrector . . 23

3.1 The required ranges for runs with different lengths of the Runs test 50

4.1 Summary of found recommendations for the noise source proposal [4] . . 70

4.2 Results of the FIPS tests (P – Passed; F – Failed) 80

4.3 Results of the NIST tests at the output random data rates 10 Mb/s and 20 Mb/s 81

4.4 Results of the NIST tests at the output random data rates 30 Mb/s and 40 Mb/s 82

4.5 Results of the NIST tests at the output random data rates 50 Mb/s and 60 Mb/s 83

4.6 Parameters including required energy per random bit of published TRNGs 86

5.1 The 4-bit sub-sequences with the low values of the approximate entropy estimation 98

5.2 The 8-bit sub-sequences with the low values of the approximate entropy estimation 99

5.3 Dimensions and numbers of elements of MOSFETs used in the reconfigurable noise source 104

5.4 Results of the NIST tests for both settings of the reconfigurable noise source which have been generated directly (Directly), using the XOR corrector (XOR), or using the Von Neumann corrector (VN) 110
5.5 Results of the FIPS tests for both settings of the reconfigurable noise source which have been generated directly (Directly), using the XOR corrector (XOR), or using the Von Neumann corrector (VN) 111
5.6 Used settings of the proposed low entropy detector 112

6.1 Results of the FIPS tests of the behavioral model of the TRNG with time multiplexed sources of randomness. The sequences were generated directly (Directly), using the XOR corrector (XOR), or using the Von Neumann corrector (VN) 122
6.2 Results of the NIST tests of the behavioral model of the TRNG with time multiplexed sources of randomness 123
6.3 Results of the FIPS tests for both settings of the behavioral model of the TRNG based on the EGA 126
6.4 Results of the NIST tests for both settings of the behavioral model of the TRNG based on the EGA 127

7.1 Results of productivity improvement based on design time measurement for different type of AMS circuits [8] 142
7.2 The results demonstrating productivity gain created by the incremental control [9] 148
7.3 The results demonstrating productivity increase created by the similar search [12] 149

Chapter 1

Introduction

Can we imagine our lives without modern communication systems? Internet and modern hand-held devices are used every day. These modern communication systems connect people and allow us to reduce geographical distances between different places in the world. Almost immediately through these systems, we can transmit or receive the latest information about development of the newest events and things. Important transactions in business, negotiations with offices, and sending of sensitive information can be also made by these communication systems. To make it all possible, one condition must be met. Security of transfers made by the modern communication systems has to be ensured. For this purpose, different session keys for authentication and encryption are created. Therefore parts of these systems are devices that are able to generate unpredictable and nonrecurring numbers – random numbers [1]. These very important devices – random number generators (RNGs) – generate random numbers with specific features, for example, uniform distribution and high entropy.

Random numbers are usually generated by mathematical computational algorithms known as pseudo-random number generators (PRNGs). Number sequences generated by PRNGs only approximate to properties of true random number sequences and are not appropriate for use in cryptographic devices and communication systems because they are generated by the computational algorithm using present and past values. The initial state usually called seed is set externally. Thus, under certain conditions, future values could be theoretically predictable [2].

For higher security level, devices based on a physical source of randomness can be incorporated into structures of modern communication or cryptographic systems instead of

PRNGs. These generators are commonly called true random number generators (TRNGs) and are based on physical phenomena exhibiting a random behavior. However, magnitudes of these sources of randomness are very small. Therefore TRNGs are very sensitive to disturbance. The basic function of these generators can be affected by sources of undesired deterministic noise such as temperature variations or power supply noise. In an extreme case, this weakness allows deliberate attacks that are executed by malicious noise. Hence, the TRNGs described in this work contain structures that are able to suppress temperature and process variation or are able to react to states caused by deterministic noises. Another fundamental parameter of the TRNGs usable in the systems above-mentioned is sufficient output random data rate. Therefore the presented TRNGs are proposed to generate high-quality random numbers with high output random data rate.

It is not possible to prove that an arbitrarily long sequence of random numbers is really random. Therefore the quality of generated output random number sequences is evaluated using well-known statistical test suites, which are composed of certain amount of exactly defined tests and are able to find any undesirable hidden dependencies or unbalanced distribution.

In order to be possible to put the proposed TRNGs on the market, it is necessary to come up with a functional product on time. This can be achieved by shortening design time because fabrication time is usually fixed. However, shorter design time must not adversely affect the quality and reliability of fabricated devices. Therefore, during development of described TRNGs, new design methodology steps have been proposed and used. For correct function of TRNGs, the high quality of physical design is essential because any inaccuracy in physical design can cause an undesirable distortion of output random data or even generator failure. Hence the introduced methodology steps are also focused on physical design of integrated circuits (ICs).

The development of TRNGs is an actual topic. The main goals of this thesis are following:

- Development of a new TRNG architecture allowing an increase of output random data rate.
- Proposal of mechanisms detecting a significant decrease of the quality of random number sequences.

- Development of behavioral models of TRNGs allowing acceleration of system simulations.
- Development of extensions of physical design methodology speeding up the design process and making designs more robust.

1.1 Organization of This Thesis

This thesis is organized as follows. The topic of this work together with descriptions of author's contributions and the state of the art are introduced in chapter 1 followed by summary of TRNG fundamentals in chapter 2. Methods of random number sequence evaluation are presented in chapter 3.

Principle and design of the improved TRNG structure with time multiplexed sources of randomness are shown and discussed in chapter 4. Chapter 5 deals with the TRNG completed by protective mechanisms detecting deliberate malicious attacks.

Simulations of systems on chip (SoCs) containing TRNGs are very time-consuming. Therefore behavioral models of TRNGs, which are able to shorten simulation time, are presented in chapter 6. Physical design time of TRNGs can be reduced by new analog and mixed-signal (AMS) methodology steps described in chapter 7. Finally, the work is concluded in chapter 8.

1.2 Author's Scientific Contributions

This work is aimed at the development of improved structures of true random number generators that produce high-quality random number sequences and can be integrated into modern communication and cryptographic systems quickly and easily. The presented TRNGs are proposed as parts of complex SoCs, which can be fabricated in standard complementary metal-oxide-semiconductor (CMOS) or bipolar-CMOS-double diffused MOS (BCD) [3] processes on silicon substrates. In other words, they are designed both from the perspective of the generated random number sequences quality and from the perspective of system integration. More specifically, this work brings the following contributions:

- **Development of the TRNG with time multiplexed sources of randomness to get higher output random data rate.**

In the proposed TRNG presented in chapter 4, the random data rate is increased by a transition from a serial approach to the parallel use of more independent sources of randomness. The so-called principle of pipelining has been implemented. This new architecture together with obtained results has been published in the impacted journal Radioengineering [4].

- **Implementation of protective mechanisms into TRNG structure.**
  The new enhanced generic architecture introduced in chapter 5 includes mechanisms, which can detect a bias in sequences of random numbers caused by deliberate malicious attacks and can reveal a significant decrease in the entropy of sources of randomness. This idea has been published in the reviewed journal ElectroScope [5] and presented at an international conference [6].

- **Development of behavioral models of designed TRNGs allowing faster simulations and verifications at a system level.**
  Simulations of SoCs containing TRNGs at the transistor level are enormously time-consuming. This work presents behavioral models of the developed TRNGs, which approximate their properties, speed up the simulations, and allow revealing system errors. The behavioral models of TRNGs have been presented at an international conference [7].

- **Development of methodology steps of AMS IC physical design allowing faster and more robust design of the TRNGs.**
  This thesis introduces new features suitable for the physical design of the developed TRNGs. A tool automatically sorting electrical devices according to their topological, structural and electrical properties has been published in the impacted journal named Integration, the VLSI Journal [8]. Other functions controlling layout objects without filling forms, searching an IC design database based on similarity of object properties, or classifying matched structures regarding systematic mismatch have been published in the reviewed journal named Advances in Science, Technology and Engineering Systems Journal [9] and presented at international conferences [10], [11], [12], and [13].

1.3 State of the Art

Cryptographic devices and secure communication systems use random numbers for their operations [1]. In other words, RNGs help to ensure the security. Random number generators can be also used for other purposes. They are commonly used to model and simulate various natural phenomena or physical processes. Random numbers are the basis for simulations of electrical circuits and elements, semiconductor structures or simulators of physical fields. Specifically, the Monte Carlo simulation method is based on generating random number sequences [14]. In statistics, random numbers are used, among other things, for selection of random samples from larger data sets. Very often for measurements in acoustics, noises with different spectral characteristics are generated by random number generators. Sequences of random numbers are used not only for scientific purposes but they appear in art or in commercial applications such as various lottery games and gambling slot machines.

Functions of many electronic devices and systems depend on generating of high-quality random number sequences with sufficient output random data rate. Any failure of the RNG leads to total failure of the whole system. Therefore it is essential to use random number sequences generated by any TRNG. In the past, different variations of TRNGs were only presented in specialized cryptographic devices such as hardware security modules, cryptographic accelerators, or smart cards. With the advent of modern compact fast computers and more sophisticated programs, the need for available high-quality random number sequences has increased. TRNGs have been incorporated into more complex systems such as motherboard chipsets [15], platforms or processors [16]. Today, TRNGs are individual blocks of SoCs [17].

True random number sequences can be extracted from the chaos, which occurs in semiconductor lasers [18], from randomness occurring in Josephson junctions in superconductive integrated circuits [19]. Other TRNGs are based on radioactive decay [20] and [21] or magnetic tunnel junctions [22]. However, these principles are not suitable for TRNGs, which are parts of SoCs, usually fabricated in standard CMOS processes on silicon substrates.

Thus TRNGs integrable into these SoCs are based on more principles appropriate such as amplification of electrical noise, oscillator jitter, metastable states of electronic circuits, or chaotic based non-linear dynamical systems. The amplification of electrical

noise is pure analog technique. In this case in a part called a noise source, a type of analog noise is extracted and amplified directly. In 1997, Holman et al. described the technique of direct noise amplification in [23]. This technique was also used by Intel in the design of the RNG, which was described by Jun and Kocher in 1999 [15]. Eberlein and Bakar proposed a TRNG based on CMOS channel noise, which is suitable for smart card applications in 2007 [24]. The new structure based on random current spikes appearing in a reverse polarized p-n junction and influencing a circuit with an unstabilized operating point was proposed by Kotě et al. in 2012 [25].

The phenomenon called the jitter is the basis for TRNGs, which are composed of standard digital cells without any analog parts. This phenomenon arises in all digital clocked circuits and can be described as an uncertainty in the exact timing of the rising edge or the falling edge in a square wave signal. Thus basic parts of these TRNGs are ring oscillators. This type of TRNGs is usually designed in Field Programmable Gate Arrays (FPGAs) or in ICs. In 2004, Kohlbrenner and Gaj proposed a TRNG using a FPGA [26], which do not contain any phase-locked loops and only uses configurable logic blocks. A detailed description of the behavior of TRNGs based on a large number of ring oscillators with the same ring length and a post-processing based on resilient functions was developed by Sunar et al. [27]. In 2006, this proposal was implemented into a FPGA by Schellekens et al. [28]. This TRNG needed minimally 110 ring oscillators of three inverters.

Fischer et al. analyzed the jitter generated in ring oscillators [29]. They used a simple physical model of jitter sources to show that the random jitter accumulates slower than the global and manipulable deterministic jitter. In 2009, Bochard et al. continued with the analysis of a improved TRNG structure based on ring oscillators and showed that the proportion of the pseudo-randomness compared to the true-randomness in the generated random raw signal is much bigger than expected [30]. In 2010, Güler and Ergün presented a high speed RNG based on the jitter, which was design and fabricated as the first known application specific integrated circuit (ASIC) [31]. For better extraction of randomness from an oscillator based TRNG, Amaki et al. proposed a jitter amplifier [32], which was fabricated in a 65 nm CMOS process in 2013. A classical ring oscillator in TRNGs was replaced by self-timed rings. A new self-timed based TRNG was presented by Cherkaoui et al. [33] where self-timed rings allowed to adjust the time lapse between two successive events propagating around the ring.

Effect of flicker noise on a CMOS ring oscillator based TRNG was described by Güler et al. in 2014, which showed that flicker noise has a positive effect on increasing the entropy [34]. Then in 2017, Coustans et al. exploited subthreshold properties of jitter of events propagating in a self-timed ring oscillator and in an inverter based ring oscillator for a design of a TRNG in a 180 nm CMOS process [35]. A concept of capacitive coupling between two ring oscillators amplifying jitter and a dual-edge sampling scheme increasing the output random data rate was used by Do and Liu for a design of a low-power TRNG fabricated in a 65 nm CMOS process [36]. In 2018, Saxl et al. published an ultra-low power TRNG based on ring oscillators composed of current-starved inverter design [37], which is intended for passive radio frequency identification tags and fabricated in a 55 nm semiconductor process.

In integrated circuits, metastable states appear and have the random behavior caused by the presence of noises. Therefore this phenomenon is especially suitable for ICs and is the basis for modern TRNGs. A TRNG based on metastable behavior was published by Kinniment and Chester in 2002 [38] and fabricated in an AMS 0.6 µm process. This solution uses a feedback loop to set the ratio between output ones and zeros. In 2008, Tokunaga et al. proposed a metastability based TRNG fabricated in a 130 nm CMOS process [39], which grades the probability of randomness regardless of the output bit value by measuring the metastable resolution time. The proposed system determines the original random noise level at the time of metastability and tunes itself to achieve a high probability of randomness. TRNGs based on metastability of digital latches and fabricated in a 350 nm CMOS process were presented by Holleman et al. [40]. Used floating-gate memory cells allow compensating offsets causing any nonrandom behavior.

A pure digital TRNG was designed by Srinivasan et al. [41] in 2009. In this TRNG, programmable cross-coupled inverters and programmable delay components reducing parasitic deterministic influences were implemented. For this design, a 45 nm CMOS technology was used. In 2010, Suresh and Burleson used this design for comparison of the digital post-processing effectiveness of using the XOR corrector and the von Neumann corrector [42], which are the most used correctors for direct post-processing in TRNGs. Then in 2015, Suresh and Burleson followed up on this work with an article [43] that described a stochastic model for metastability based TRNG and an impact of intra-die variations. Using this model, they compared three basic post-processing tech-

niques the XOR corrector, the von Neumann corrector, and the PRESENT cipher corrector. Meanwhile, in 2012, they proposed a sub-vdd pre-charge technique [44], which improve the tolerance of the used metastability based TRNG to device mismatch. Mathew et al. designed TRNGs based on the above-mentioned unstable cross-coupled inverter pair architecture in a 45 nm CMOS process [45] and in a 14 nm fin field effect transistor (FinFET) CMOS process [46]. Another TRNG linked a jitter generator as a noise pre-amplification and a metastable latch as jitter edge detector [47]. This solution was developed by Kuan et al. in 2014 and was fabricated in a 40 nm CMOS technology. In 2017, Tao and Dubrova presented a TRNG fabricated in a 65 nm CMOS process [48], in which randomness was extracted from a circuit formed by latch comparators and a metastable detector allowing ternary valued outputs. Kotě et al. developed a TRNG with time multiplexed metastability-based sources of randomness in 2018 [4]. This generator fabricated in a 130 nm CMOS technology is suitable for ICs used in modern hand-held devices and is resistant to changing environmental conditions.

Metastability based TRNGs have been mostly implemented into ICs. However, in 2012, Hata and Ichikawa proposed a pure digital metastability based solution of TRNGs [49] that can be implemented in any FPGAs. The RS latch was implemented as a source of randomness. The XOR tree was applied on the outputs of the 256 RS latches for improving the quality of output random sequences. Then in 2013, Wieczorek presented another solution [50] and [51], which is based on a response time difference of a pair of nearly metastable D flip-flops and can also be implemented in any FPGAs. Another TRNG developed by Li et al. in 2017 uses cross-coupled NAND gates and manual routing [52]. The proposed TRNG was implemented in a FPGA and contained self-calibration and post-processing circuits.

Another type of TRNGs also suitable for ICs uses chaotic based non-linear dynamic systems and their high sensitivity to initial condition changes. A TRNG developed by Petrie and Connelly [53] combined direct noise amplification, oscillator sampling, and discrete-time chaotic system based on the Bernoulli shift map and was fabricated in 2 μm CMOS technology in 1998. A TRNG architecture based on a circuit called a double-scroll attractor was published by Yalcin et al. in 2004 [54]. This generator was realized by current feedback operational amplifiers and is appropriate for integration on a chip. Drutarovsky and Galajda proposed a TRNG extracting randomness from a chaotic

switched-capacitor circuit in 2007 [55]. This proposal is embedded into a mixed-signal reconfigurable device without any external components. In 2010, Pareschi et al. designed two TRNG prototypes [56] developed from a pipeline analog-to-digital converter (ADC) design, which was modified to operate as a set of piecewise-linear chaotic maps. The prototypes were fabricated in a 350 nm respectively in a 180 nm CMOS processes. In 2014, Cicek et al. developed two chaos based TRNGs. The first [57] used the dual Bernoulli map as a source of randomness and was implemented in a field programmable analog array IC. The second [58] operated with a current mode skew tent map circuit and was designed in a 250 nm CMOS technology.

A compact discrete-time chaotic oscillator is proposed to approximate a V-shape map and is a core of a TRNG [59], which was designed in a 180 nm CMOS process and presented by Jiteurtragool et al. in 2015. In the same year, Park et al. proposed a TRNG using a discrete Boolean chaotic oscillator [60] fabricated in a 350 nm CMOS technology. Also, the chaotic based TRNGs can be implemented in a FPGA. In 2017, Koyuncu and Ozcerit published a solution suitable for FPGAs [61] that used the Sundarapandian-Pehlivan chaotic system. Next year, Wieczorek and Golofit introduced a new concept of a TRNG combining metastability based source of randomness and a chaotic circuit using a time-continuous random variable in a feedback loop [62]. This concept was verified in a FPGA and is also appropriate for gate level on-chip implementations.

Safety of contemporary systems is dependent on the unpredictability of random number sequences. Safety of these systems can be violated if there is a possibility that the values of generated random numbers can be detected or affected by a deliberate attack. Therefore the random number sequences are generated by TRNGs, which are tested and qualified by statistical tests. For testing of the majority of TRNGs, two well-known statistical test suites are usually used. The first test suite – FIPS PUB 140-2, Security Requirements for Cryptographic Modules [63] (the FIPS test suite) – was published by the National Institute of Standards and Technology in 2001. The second test suite – A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications [64] (the NIST test suite) – was published by the National Institute of Standards and Technology in 2010. This strict test suite is suitable for applications where the strongest requirements are needed. In some publications, the third statistical test suite – The Diehard Battery of Tests of Randomness [65] – developed by Marsaglia in 1995

appears. A necessary condition for implementation of a TRNG into modern systems is positive results of these test suites.

Simulations of proposed circuits are an integral part of IC design. As described above, TRNGs can be parts of SoCs. Simulations of these complex systems at a transistor level is becoming unrealistic due to enormous time demands [66]. Therefore researchers and designers develop behavioral models of individual parts of SoCs, which allow to simulate and verify the whole system in an acceptable time. Petrie and Connelly presented a model of the proposed TRNG in [53]. Properties of output random data were evaluated by statistical tests. The TRNG structure published by Kotě et al. in [25] was verified by a developed behavioral model, which was created in analog hardware description language Verilog-A. Then, they described an enhanced architecture of TRNGs [5], which was also modeled in Verilog-A and can detect deliberate malicious attacks. For evaluation of well-known post-processing techniques, Suresh and Burleson developed a model of the metastability-based TRNG [43]. Behavioral models of TRNGs developed on the base of usually used architectures were presented by Kotě et al. in [7]. These types of generators are suitable for implementation in SoCs.

Another integral part of IC design is physical design. TRNGs are sensitive circuits, whose physical design has to be created with the utmost care because any error can cause generator failure. Moreover, TRNGs are often not only composed of digital cells. They are a typical representative of AMS circuits. This part of state of the art can also be found in the author's publication [8].

In the early years of IC design and production, a lot of time was spent on an inspection of the semiconductor structures drawn on tracing paper as all geometrical parameters were checked manually by rulers and protractors. As time passed, this inefficient technique was replaced by computer-aided design (CAD) applications, which are beneficial for both design and inspection of the semiconductor structures. Development of CAD applications for IC physical design began in the second half of the 20th century [67], [68]. Today's CAD systems [69], [70], [71] allow modifications of physical design flow and implementation of new algorithms. Unlike digital synthesis and implementation tools, AMS physical design still remains handwork. Several automated tools for analog layout were presented [72]. Some of them are capable of good quality routing, some of them help with generation of devices, unfortunately still there is no tool widely commer-

cially used for analog placement. There are two main reasons for this situation. At first, placement of AMS circuits requires much more constraints compared to digital ones. As described by Scheible and Lienig in [73], these constraints can be implicit or explicit, and in AMS physical design flow it is very difficult to express all of them explicitly to be then accessible for algorithmic processing.

The second obstacle in AMS physical design flow, well-stated in [74], is somehow a reluctance of analog designers to use a complete automated analog flow. They are missing some kind of "aesthetics" in an automatically proceeded layout. Aesthetics which enables human beings to verify with a naked eye if electrical and matching requirements are met. Instead, analog designers prefer small automated steps, using so-called in-design assistants [73] that design environments from different vendors such as [69], [70], [71] are full of. Moreover, in-design assistants are not only developed by the vendors, but they are also developed by researchers. Vacula et al. developed an incremental control of layout objects [10], which removes form filling based on incremental approach. They also introduced a way of search in IC design database [12], which is based on similarity of object properties. Systematic mismatch arising in matched structures can be minimized by a method of matched structure classification presented in [13]. Designers can use in-design assistants for time-consuming tasks, but they keep the control of the whole layout.

In the final phase of physical design, lithographic design shapes are transformed into text or vector files and their geometrical and electrical parameters are checked by automatic algorithms [75]. The principle of these checks has remained similar up to now, of course, modern and more efficient algorithms [76], [77] are used in the contemporary CAD environments. Unfortunately, this saves time only during the inspection at the end of design.

However, another amount of time can be saved by anticipating human failure in early stages of layout. Again, automatic algorithms implemented in some CAD environment can do a part of the job. They are able to lay out simple integrated circuits [78], typical circuit structures [79], [80], basic analog building blocks [81] or to optimize floorplan using multi-objective optimization algorithm based on topological benchmark [82]. They can replace lengthy and complex manual work, in which humans are most likely to make a mistake. Moreover, if they keep respecting particular stages of IC layout, these algo-

rithms can be appropriately combined, so results of one algorithm can be used as an input for another one. The analog layout flow is then becoming partially automated.

Numerous works describing almost fully autonomous physical design synthesis have been published. The approach [83] introduced by Eick et al. describes automatically generated constraints used for an automatic placer and then for an automatic router. Another automatic physical design flow [84] developed by Habal and Graeb takes into account influence of each device reshaping and is focused on finding an optimal layout by applying a non-slicing placement algorithm and numerical simulation to evaluate device sizing procedure and effect of layout parasitics.

Automated methods [83], [84] look very promising, but there are several drawbacks which limit their usage in practice. The core of these methods is an extraction of matched structures such as current mirrors, cascaded current mirrors, and differential pairs. High precise designs use trimmed structures which contain a number of pass-gates and switches to be able to configure required circuit parameters. Also in low power designs used in mobile applications and the Internet of Things (IoT), the power consumption is very optimized. Due to this need, AMS circuits contain switches disconnecting dedicated parts. In high reliable design, the negative bias temperature instability (NBTI) [85] and the positive bias temperature instability (PBTI) phenomena [86] are taken into account. Therefore inputs of differential amplifiers are usually reconfigurable. The additional switches do not allow simple detection of basic analog structures. It is another reason why until today there is no universal tool for automated AMS physical design synthesis and AMS physical designs are still made manually. Therefore Kotě et al. introduced a pre-placement phase of IC AMS physical design flow [8], which automatically sorts electrical devices used in planar IC technologies according to their topological, structural and electrical properties. The automated pre-placement phase is able to prevent a creation of hardly detectable errors occurring during physical design and speeds up the entire design process.

1.4 Solution Methods of the Work

Known and world-wide spread environment Cadence Virtuoso has been used to development TRNGs suitable for ICs. As described above, proposed TRNGs are the representative of AMS electronic circuits. For proposal implementation, the Analog-on-Top

approach has been applied. Thus schematic diagrams have been created in Cadence Virtuoso Schematic Editors [87], [88]. Functions of proposed circuits have been simulated by known tools such as Mentor Eldo [14] or Virtuoso Spectre Circuit Simuator [89]. The TRNGs have been developed in 130 nm bulk CMOS technologies known as HCMOS9GP and HCMOS9A and in 160 nm BCD technologies known as BCD8sP and SOIBCD8S. All of these technologies have been provided by STMicroelectronics. To be possible to simulate SoCs containing TRNGs in an acceptable time, behavioral models of these generators have been created in the Verilog-A hardware description language (HDL) [90]. The models have been simulated in the simulators above-mentioned.

Physical design of proposed TRNGs has been created in Cadence Virtuoso Layout Editors [91], [69]. New methodology steps of physical design flow have been developed in the Cadence SKILL programming language [92] and in the Cadence SKILL IDE environment [93]. Before pattern generation tape-out, the proposed topologies were checked by Calibre DRC and LVS [94] whether they were compatible with design rule manual (DRM) respectively with the proposed schematic diagrams. Parasitic capacitors and resistors have been extracted from completed physical designs by Synopsys Star-RCXT [95]. A fabricated TRNG has been characterized and validated by standard laboratory equipment such as precise voltage and current sources, oscilloscopes, logic analyzers, etc. All generated and measured data have been processed by MATLAB [96].

Chapter 2

TRNG Fundamentals

Development of true random number generators progresses dynamically, as it is mentioned in state of the art description in section 1.3. Modern communication and cryptographic systems still have higher requirements on safety. In this regard, random number sequences generated by PRNGs have insufficient properties such as periodicity of the generated sequences. Therefore it is necessary to develop new and more sophisticated TRNGs. Although the development of TRNGs is advanced, it is essential to describe fundamentals of the TRNGs as first.

Before the description of basic structures of TRNGs, random numbers are defined. The random numbers can only be considered in the case when the number is a part of a sequence of several other numbers, which were generated independently within each other of a defined set or an interval with a probability distribution. Moreover, future values of random numbers are not possible to predict based on knowledge of present or past values of the number sequence. Cores of the TRNGs usually generate digital values in the form of logic bits, which can be processed in additional blocks creating any more complex format of numbers. Therefore, in this case, sequences of logic zeros and logic ones are assumed.

In this chapter, a generic architecture of TRNGs is introduced. The basic principles of TRNGs suitable for utilization in ICs are thermal noise generated by resistors, oscillator jitter, metastable states of electronic circuits, or chaotic behavior of non-linear dynamic systems. Description of these principles is followed by explanations of basic physical phenomena with the random behavior, which serve as a source of randomness.

2.1 Generic Architecture

True random number generators are devices, whose output digital signals exhibit non-deterministic properties. Cores of these devices are able to generate random bits, and by their processing, random numbers can be formed. The generated sequences of logic zeros and logic ones are divided into frames with a fixed length. These created numbers are written in the frames in the binary form and can be marked as the random numbers. For modifications of some of their properties, a post-processing algorithm can be incorporated into a TRNG architecture. Majority of TRNGs follows a generic architecture that was also presented in [97], [28] and [98]. The generic architecture consists of several blocks. A noise source and a digitizer form a source of randomness. Into the generic architecture, a post-processing block and an output interface are also incorporated. The arrangement of these blocks is shown in figure 2.1.

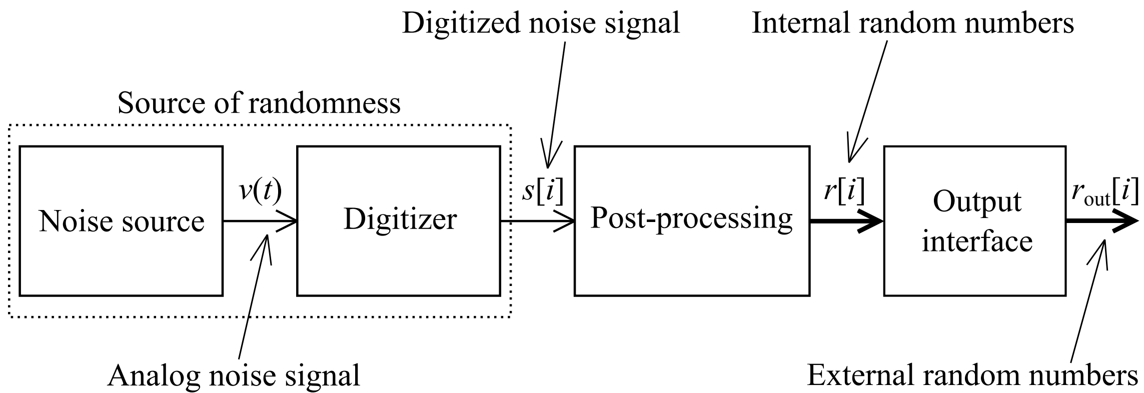


Fig. 2.1: The generic architecture of a TRNG [97]

A noise source produces a signal with non-deterministic features commonly called an analog noise signal $v(t)$, which is a continuous signal in the time t. This signal $v(t)$ is digitized in the next block – a digitizer – that creates a digitized noise signal $s[i]$ where i is a normalized period, $i = t/\Theta_{\mathrm{c}}$, and Θ_{c} is a period of a clock signal. Occurring imperfections of the digitized noise signal $s[i]$ such as a bias, hidden dependencies, or unbalanced distribution can be solved in the post-processing block, which produces internal random numbers $r[i]$. The format of the internal random numbers must be adapted to the required output data format. This is a function of an output interface that produces an external random numbers $r_{\mathrm{out}}[i]$ compatible with system requirements.

2.1.1 Noise Source

The noise source is a unit creating the analog noise signal $v(t)$ based on a physical phenomenon. If this phenomenon has the random behavior, the output analog noise signal $v(t)$ can also have random features. An ideal noise source generates the analog noise signal $v(t)$ with the uniform probability distribution. However, it is tough to develop such a source of randomness. Therefore some correction data techniques are usually used in combination with an appropriate source of randomness, whose properties are closer to the properties of the ideal source of randomness.

Physical phenomena such as radioactive decay, chaos occurring in lasers, or randomness occurring in superconductive circuits can be used as noise sources for TRNGs. However, the TRNGs are usually implemented in SoCs. For these purposes, it is essential to use the noise source, which can be integrated into a chip. In real integrated TRNGs, electrical noises generated in semiconductor structures, jitter arising in ring oscillators, metastable states appearing in ICs, or chaos emerging in non-linear dynamic systems can be used.

2.1.2 Digitizer

The contemporary SoCs works with digital signals. For this reason, TRNGs generate a random signal in digital form. To be possible to produce the digitized random signal, the generic architecture of TRNGs contains the digitizer. For demonstration in this part, there are shown basic models of analog noise signal digitization.

The main component of the first model of digitization is a clocked comparator. Structure of this model is shown in figure 2.2. The internal analog noise signal is amplified to have sufficient amplitude to be able to drive the input of the clocked comparator. The analog noise signal $v(t)$ is compared with a reference in each clock period. Thus the signal $v(t)$ is converted to a signal with discrete levels, which is synchronized with the system clock. The output signal can be marked as the digitized noise signal $s[i]$ and processed in next blocks. The topology containing comparator was described in [23].

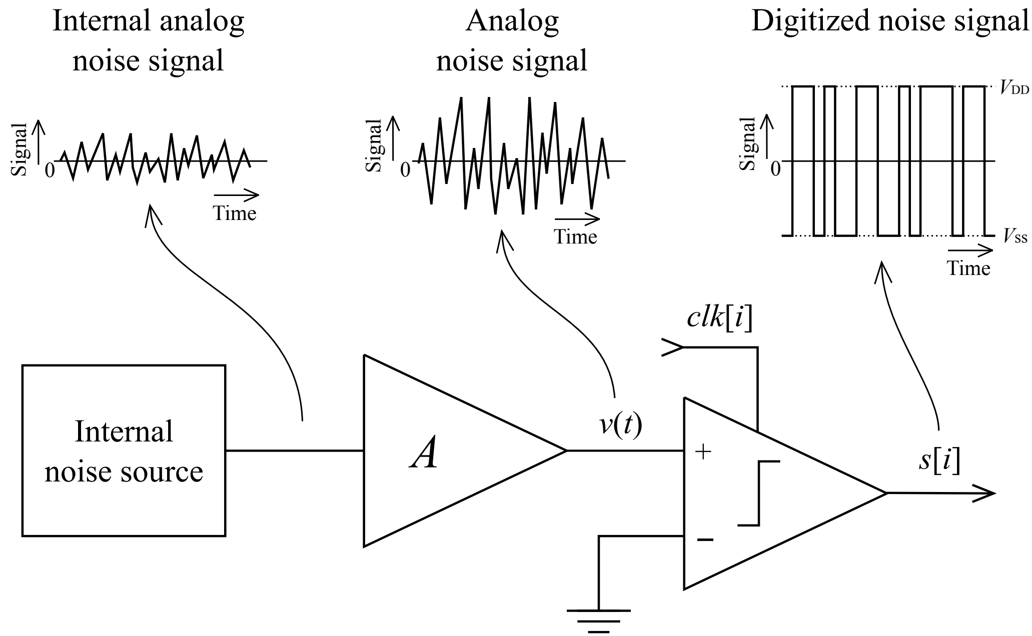


Fig. 2.2: The digitization model with the clocked comparator

Another digitization model is composed of digital circuits, namely an inverter and a D-type flip-flop, and is shown in figure 2.3. In cases where the analog noise signal $v(t)$ has sufficient amplitude and its DC component is set at a level of the transition region of the inverter, any deflection of $v(t)$ causes inverter flip, which creates a signal with discrete levels. Then this signal is sampled by the D flip-flop. In this manner, the digitized noise signal $s[i]$ is formed. This principle was used in [25].

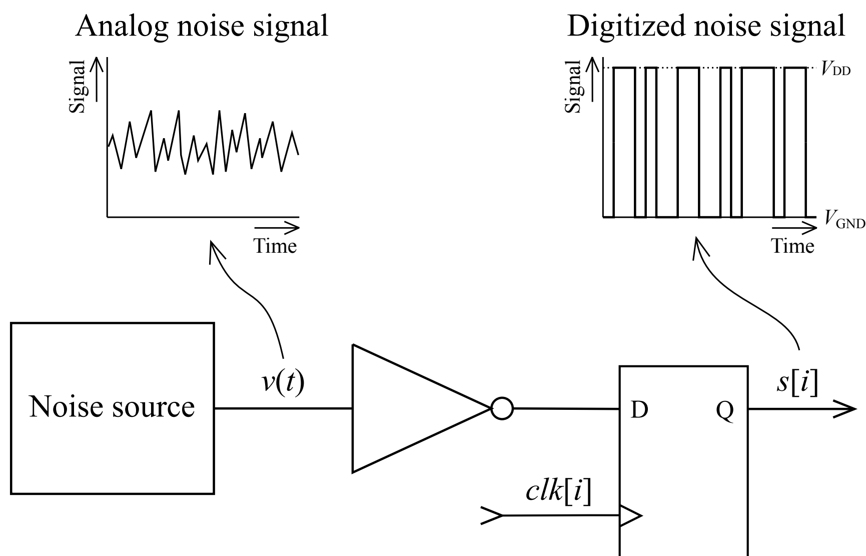


Fig. 2.3: The digitization model composed of digital circuits

The jitter can be defined as uncertainty of exact timing of a rising edge or a falling edge of digital signals. In principle, TRNGs working on the base of this phenomenon contain an element, which only samples the noise signal. This signal is generated by ring oscillators composed of digital cells and therefore has only discrete levels as it is described in [26] and [27]. In other words, the digitized noise signal $s[i]$ is produced by the D flip-flip, which samples the noise signal.

2.1.3 Post-processing Block

The post-processing block can solve imperfections of random data that arise in the noise source or the digitizer. The probability distribution of the digitized noise signal $s[i]$ does not have to be purely uniform, and hence it is appropriate to adjust it. For example, an input offset of the clocked voltage comparator depicted in figure 2.2 creates statistical defects in the digitizer and distorts the uniform probability distribution of the digitized noise signal $s[i]$. After post-processing, a probability distribution of the adjusted internal random numbers $r[i]$ is closer to the uniform distribution than a probability distribution of the digitized noise signal $s[i]$.

The digitized noise signal $s[i]$ may also exhibit low entropy. Then entropy per bit of the internal random numbers $r[i]$ can be increased by a suitable post-processing algorithm applied on the digitized noise signal $s[i]$. The result is higher entropy of the internal random numbers $r[i]$, however, lower random data rate. In other words, inclusion of the post-processing block into a TRNG structure allows to use a source of randomness with lower entropy per bit and to increase resistance to deliberate malicious attacks and deterministic environmental changes such as temperature variations or power supply distortion.

In the post-processing block, several different functions can be implemented for reduction of a distortion of internal random numbers $r[i]$. Fundamental and mostly used functions for the post-processing block in TRNGs are the XOR corrector [99] and the von Neumann corrector [100]. The post-processing can be based on other more complex algorithms such as cryptographic hash functions [101], linear feedback shift registers (LFSRs) [36], or linear codes [102]. The more complex algorithms can affect the power consumption of TRNGs, which is undesirable especially in modern hand-held devices

where it could affect the overall power consumption of the system and shorten the time between charging the battery.

For description of properties of the mostly used post-processing functions, basic statistic operations are defined. Symbols X, Y, and Z denote random bits having logic zero or logic one. Thus, in this case, the mean value $\mathrm{E}(X)$ can be defined as the average value of a large number of consecutive random bits and thus

$$\mathrm{E}(X) = \Pr(X = 1) \tag{2.1}$$

where $\Pr(X = 1)$ is the probability when the random bit is equal to 1. The variance of X is measure of the variability of X and is determined by

$$\mathrm{var}(X) = \mathrm{E}\{[X - \mathrm{E}(X)]^2\} = \mathrm{E}(X)\,[1 - \mathrm{E}(X)]\,. \tag{2.2}$$

The correlation of X and Y is defined as

$$\mathrm{cor}(X, Y) = \frac{\mathrm{cov}(X, Y)}{\sqrt{\mathrm{var}(X)\mathrm{var}(Y)}} \tag{2.3}$$

where $\mathrm{cov}(X, Y)$ is the covariance of X and Y defined by

$$\mathrm{cov}(X, Y) = \mathrm{E}\{[X - \mathrm{E}(X)]\,[Y - \mathrm{E}(Y)]\}. \tag{2.4}$$

The correlation is equal to 0, when X and Y are independent. However, when X and Y are identical, the correlation is equal to 1.

An ideal TRNG generates logic ones with the probability of $\frac{1}{2}$. Thus according to (2.1), the mean value of the digitized noise signal $s[i]$ is equal to $\frac{1}{2}$. However, a real TRNG generates logic ones with the probability, which may be slightly deviated from the ideal case. Therefore the mean value also deviates. In other words, the real TRNG generate random bits with a bias b, which can be defined as

$$\Pr(X = 1) = \frac{1}{2} + b \tag{2.5}$$

and

$$\Pr(X = 0) = \frac{1}{2} - b \tag{2.6}$$

where the bias b is a real number in the range of $-\frac{1}{2}$ to $\frac{1}{2}$.

The XOR corrector is based on logic exclusive or (XOR) operation and its function is described in detail [98] and derived in [99]. This known operation can be defined by

a truth table 2.1 and marked by a symbol ⊗. Basic properties of the XOR operation is commutativity

$$X \otimes Y = Y \otimes X \tag{2.7}$$

and associativity

$$(X \otimes Y) \otimes Z = X \otimes (Y \otimes Z). \tag{2.8}$$

X	Y	$X \otimes Y$
0	0	0
0	1	1
1	0	1
1	1	0

Tab. 2.1: The truth-table of the XOR operation

The introduced bias can be reduced by implementation of the XOR corrector in the post-processing block. Amount of bias reduction is determined by the probability when the XOR corrector creates logic one from biased input data assuming that random bites are generated independently. Thus

$$\Pr[(X \otimes Y) = 1] = \Pr[(X = 1)(Y = 0) \cup (X = 0)(Y = 1)] \tag{2.9}$$

and after apparent adjustments, the probability can be transferred into

$$\Pr[(X \otimes Y) = 1] = \Pr(X = 1)\Pr(Y = 0) + \Pr(X = 0)\Pr(Y = 1). \tag{2.10}$$

By inserting the formulas (2.5) and (2.6) into the equation (2.10), the bias reduction is expressed as

$$\Pr[(X \otimes Y) = 1] = \frac{1}{2} - 2b^2. \tag{2.11}$$

The form (2.11) shows that the XOR corrector reduces the bias from b to $2b^2$. However, output random data rate B_{XOR} is halved.

The bias reduction can be also described from the point of view of the mean value. Moreover, in this way, an effect of correlated X and Y can be determined. Thus if X and Y are independent then the mean value of the XOR operation is described by

$$\mathrm{E}(X \otimes Y) = \frac{1}{2} - 2\left[\mathrm{E}(X) - \frac{1}{2}\right]\left[\mathrm{E}(Y) - \frac{1}{2}\right]. \tag{2.12}$$

From this equation, it is obvious that

$$|\mathrm{E}(X \otimes Y) - \frac{1}{2}| \leq \min\left(|\mathrm{E}(X) - \frac{1}{2}|, |\mathrm{E}(Y) - \frac{1}{2}|\right). \tag{2.13}$$

Thus the bias is reduced in all cases.

In case when X and Y have any non-zero correlation, the mean value of the XOR operation is given by

$$\mathrm{E}(X \otimes Y) = \frac{1}{2} - 2\left[\mathrm{E}(X) - \frac{1}{2}\right]\left[\mathrm{E}(Y) - \frac{1}{2}\right] - 2\,\mathrm{cor}(X, Y)\sqrt{\mathrm{var}(X)\mathrm{var}(Y)}. \tag{2.14}$$

The mean value of generated random data by a real well-functioning TRNG is usually very close to $\frac{1}{2}$. Therefore, for these cases, formula (2.14) can be approximated by

$$\mathrm{E}(X \otimes Y) \approx \frac{1}{2} - 2\left[\mathrm{E}(X) - \frac{1}{2}\right]\left[\mathrm{E}(Y) - \frac{1}{2}\right] - \frac{1}{2}\mathrm{cor}(X, Y). \tag{2.15}$$

As can be seen from formulas (2.14) and (2.15), the correlation of X and Y can affect the result and add bias.

In some cases, the XOR corrector can processed more than two random bits. The mean value of the XOR operation of independently generated random bits $X_1, X_2, \ldots, X_w$ is expressed by

$$\mathrm{E}(X_1 \otimes X_2 \otimes \ldots \otimes X_w) = \frac{1}{2} + (-2)^{w-1}\left[\mathrm{E}(X) - \frac{1}{2}\right]^w. \tag{2.16}$$

This formula (2.16) describes the basic property of the XOR corrector. The resulting bias decreases with the increased amount of the processed independent random bits. A bias reduction of random bit sequences with the same mean values is shown in table 2.2.

	$\mathrm{E}(X_1 \otimes \cdots \otimes X_w)$			
$\mathrm{E}(X)$	$w = 2$	$w = 3$	$w = 4$	$w = 5$
0.48	0.4992	0.499968	0.49999872	0.4999999488
0.49	0.4998	0.499996	0.49999992	0.4999999984
0.50	0.5000	0.500000	0.50000000	0.5000000000
0.51	0.4998	0.500004	0.49999992	0.5000000016
0.52	0.4992	0.500032	0.49999872	0.5000000512

Tab. 2.2: Examples of the mean value calculated using the formula 2.16

In 1951, John von Neumann proposed a method [100], which is able to remove the bias from random bit sequences. This method known as the von Neumann corrector is often implemented in the post-processing block of TRNGs such as [15], [42], [43], or [25]. The basic function of the corrector is defined by a table 2.3 and marked by a symbol $\oslash$ in this work. The corrector converts two different input bits into one valid output bit, but the same input bits do not generate any new output bit. This operation is neither commutative nor associative.

X	Y	$X \oslash Y$
0	0	none
0	1	1
1	0	0
1	1	none

Tab. 2.3: The table describing the basic function of the von Neumann corrector

The bias removal can be determined by the probability when the von Neumann corrector produces logic one from biased input data. Because at the input bit equality, the corrector does not produce any output bit, the probability is determined under the condition of the valid input, that is, in other words, the different input bits [102]. Thus

$$\Pr\left[(X \oslash Y) = 1\right] = \Pr\left[(X \oslash Y) = 1 | (X = 1)(Y = 0) \cup (X = 0)(Y = 1)\right]. \tag{2.17}$$

After adjustments, the probability is

$$\Pr\left[(X \oslash Y) = 1\right] = \frac{\Pr\left\{\left[(X = 0)(Y = 1)\right]\left[(X = 1)(Y = 0) \cup (X = 0)(Y = 1)\right]\right\}}{\Pr\left[(X = 1)(Y = 0) \cup (X = 0)(Y = 1)\right]}. \tag{2.18}$$

This equation can be modified to the following form

$$\Pr\left[(X \oslash Y) = 1\right] = \frac{\Pr(X = 0)\Pr(Y = 1)}{\Pr(X = 1)\Pr(Y = 0) + \Pr(X = 0)\Pr(Y = 1)}. \tag{2.19}$$

By inserting the formulas (2.5) and (2.6) into the equation (2.19), the effect of the bias on processed bit sequences is expressed as

$$\Pr\left[(X \oslash Y) = 1\right] = \frac{\frac{1}{4} - b^2}{2(\frac{1}{4} - b^2)} = \frac{1}{2}. \tag{2.20}$$

So, as is apparent from equation (2.20), the von Neumann corrector completely eliminates the bias. However, this extraordinary feature is at the cost of lower output random data

rate B_{VN} than the data rate of the source of randomness B_{SR}. The output data rate B_{VN}, which is even variable, can be determined by the probability of occurrence of valid input bit pairs. This corrector creates one valid bit from the input pair. Therefore the probability must be divided by two [102]. Thus

$$B_{\text{VN}} = \frac{\Pr\left[(X=1)(Y=0) \cup (X=0)(Y=1)\right]}{2} B_{\text{SR}}. \tag{2.21}$$

After adjustments above-used and by inserting the formulas (2.5) and (2.6), the output random data rate B_{VN} is expressed as

$$B_{\text{VN}} = \left(\frac{1}{4} - b^2\right) B_{\text{SR}}. \tag{2.22}$$

This formula (2.22) describes a considerable disadvantage of the von Neumann corrector. Thus, at zero bias, the maximal output random data rate B_{VN} is only one-quarter of the data rate of the source of randomness B_{SR}. With the increasing bias, the output random data rate B_{VN} falls further.

A resilient function as a method of bias reduction is used in some cryptographic applications. This technique can reduce the bias but at the cost of reducing the random data rate. Its definition is taken from [27]. An (e_1, e_2, e_3)-resilient function is a function

$$F(x_1, x_2, \ldots, x_{e_1}) = (y_1, y_2, \ldots, y_{e_2}) \tag{2.23}$$

from $\mathbb{Z}_2^{e1}$ to $\mathbb{Z}_2^{e2}$ enjoying the property that, for any e_3 coordinates $h_1, \ldots, h_{e_3}$, for any constants $z_1, \ldots, z_{e_3}$ from $\mathbb{Z}_2$ and any element y of the codomain

$$\Pr\left[F(x) = y | x_{h_1} = z_1, \ldots, z_{h_t} = z_t\right] = \frac{1}{2^{e_2}}. \tag{2.24}$$

In the computation of this probability, all x_h for $h \notin \{h_1, \ldots, h_t\}$ are viewed as independent random variables, each of which takes on the logic one or the logic zero with probability 0.5.

In other words, the input to the resilient function is the digitized noise signal $s[i]$, which is composed of a large number (for example 0.9ι) of random bits infiltrated by a small number (for example 0.1ι) of bits produced by a source of randomness. If an $(\iota, e_2, 0.1\iota)$-resilient function processes this signal with length ι bits, the output marked as the internal random numbers $r[i]$ consist of e_2 corrected random bits [27]. From this, it follows that the output random data rate is given by the ratio of e_2 to e_1.

The resilient function is preferably constructed by a linear code, which is given by

$$F(\mathbf{x}) = \mathbf{x}\mathbf{G}^T \tag{2.25}$$

where F is the (e_1, e_2, e_3)-resilient function and $\mathbf{G}$ is a generator matrix for the so-called $[e_1, e_2, e_3 + 1]$ linear code. The resilient functions can be implemented at the hardware level by LFSRs. A short code length is more accessible to implement and requires smaller buffers, but TRNGs using such a code are more threatened by deterministic noise or possible deliberate malicious attacks. Therefore there must be done a trade-off between code length and the size of the buffers.

2.1.4 Output Interface

TRNGs are usually implemented in complex SoCs, and therefore they have to meet requirements of the system for an output data format to be able to communicate with other parts. The output interface, which is a part of TRNGs, modifies the internal random numbers r, adjusts the format of output data, and creates the external random numbers r_{out} with required properties such as suitable voltage levels of logic ones and logic zeros, correct data timing, and the sufficient fan-out.

In generators using a post-processing algorithm with variable output data rate, a buffer or a shift register are incorporated into the output interface. They store random bits and provide random data stream with constant data rate.

2.2 TRNGs with Direct Noise Amplification

Electrically measurable characteristics resulting from random behavior of charge carriers in conductors of semiconductors are the basis for randomness extraction in TRNGs. Direct amplification of noise is a pure analog type of the randomness extraction. This method exploits noises arising in resistors, semiconductor junctions, or CMOS structures, where noises described in 2.6 occur. An internal noise signal $v_{\text{int}}(t)$ formed by noises in a circuit is directly amplified and the processable analog noise signal $v(t)$ is created according to

$$v(t) = A\, v_{\text{int}}(t) \tag{2.26}$$

where A is amplifier amplification. This principle is a part of figure 2.2. The amplifier has to preserve all components of randomness of the internal analog noise signal $v_{\text{int}}(t)$. In order to minimize the bias of the analog noise signal $v(t)$, the circuit used for the amplification should have very good properties such as a high degree of linearity and a high bandwidth. Then the signal $v(t)$ can be processed by some digitization method described in section 2.1.2 or its derivative.

2.2.1 Source of Randomness Based on Thermal Noise of Resistors

The method of direct noise amplification was implemented in the TRNG described in [15]. Its block diagram is shown in figure 2.4. The internal noise signal $v_{\text{int}}(t)$ is generated by subtracting signals from two undriven well-matched resistors, in which thermal noise occurs. Subtraction of the signals eliminates deterministic environmental changes such as fluctuations of temperature, power supply, or electromagnetic radiation. The internal noise signal $v_{\text{int}}(t)$ is amplified by an amplifier, which does not cause any additional bias.

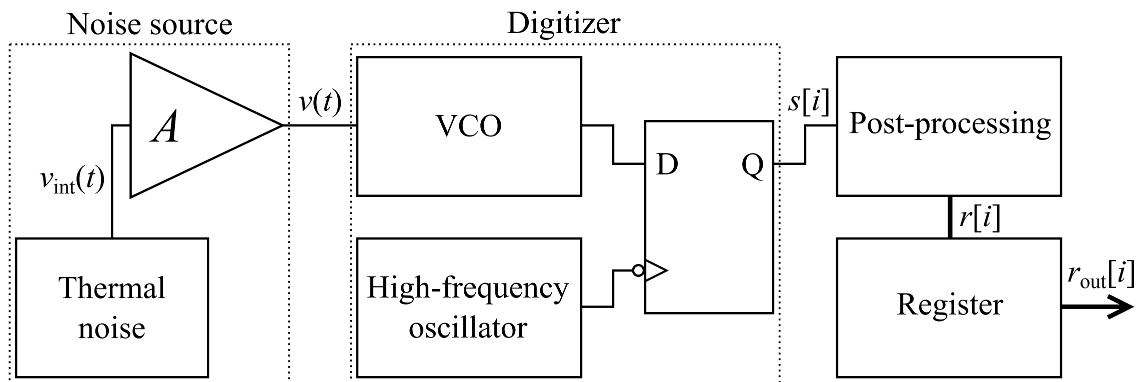


Fig. 2.4: The block diagram of the TRNG exploiting thermal noise of resistors [15]

The analog noise signal $v(t)$ is not directly sampled, but it modulates a basic frequency of a voltage controlled oscillator (VCO). The output signal of the VCO with the changing frequency controls sampling a square-wave signal, which a high-frequency oscillator produces. The process of sampling is shown in figure 2.5. The basic frequency of the VCO is hundred times lower than the frequency of the high-frequency oscillator. Maximal deflection of the modulated frequency corresponds to 20 periods of the high-frequency oscillator signal.

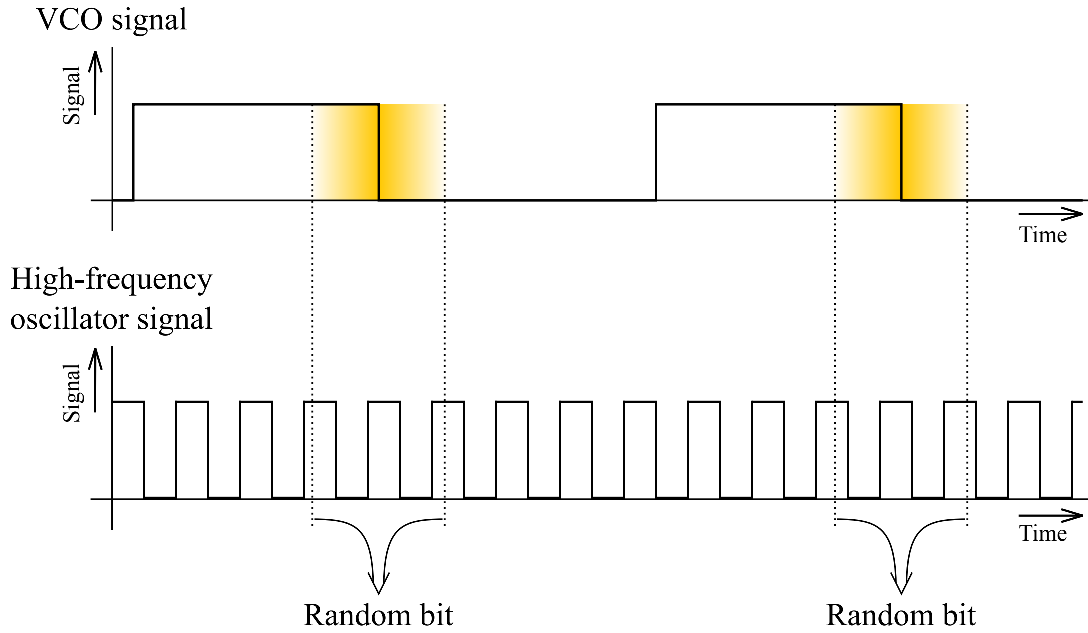


Fig. 2.5: The principle of VCO controlled sampling [15]

Any possible bias of the digitized noise signal was removed by the von Neumann corrector implemented in the post-processing block of this TRNG. A 32-bit register was incorporated into the output interface due to the variable random data rate, which exceeded 75 kb/s. The correct function of this proposal was confirmed by successful statistical tests.

2.2.2 Direct Amplification of Noise in Semiconductor Junction

A TRNG with direct amplification of analog noise arising in P-N junctions of semiconductor structures was published in [25], and its block diagram is depicted in figure 2.6. Randomness is extracted by two identical independent noise sources, whose schematic diagram is shown in figure 2.7. A reverse polarized P-N junction emitter-base of the NPN bipolar transistor Q_1 works as the source of random noise, more precisely, as the source of random spikes occurring in current that flows through the junction. This current flows into the base of the transistor Q_2. Then the internal noise signal $v_{\text{int}}(t)$ appearing on the collector of the transistor Q_2 is amplified in next part because the analog noise signal $v(t)$ must have sufficient amplitude to be able to drive digitizer inputs. Therefore the amplifier composed of the same bipolar transistors is incorporated. The part of the amplifier input is a capacitor separating DC component of the internal noise signal $v_{\text{int}}(t)$.

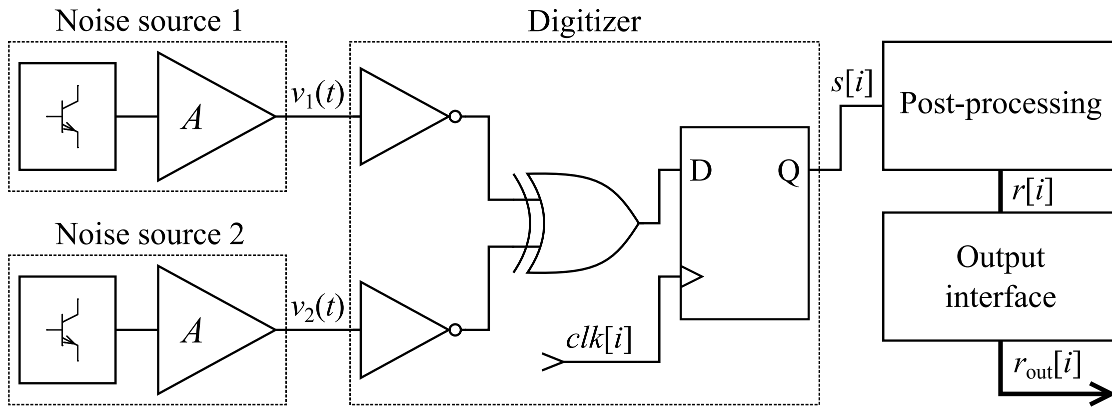


Fig. 2.6: The block diagram of the TRNG with two independent noise sources [25]

The digitizer processes both analog noise signals $v_1(t)$ and $v_2(t)$. Inverters create signals with discrete voltage levels, which are processed together by the XOR gate and then sampled by the D flip-flop, as can be seen in figure 2.6. Any possible bias of the digitized noise signal $s[i]$ can be reduced in the post-processing block where the XOR and von Neumann correctors are implemented. A user can choose the corrector type or disable them. Then the internal random numbers $r[i]$ are formed directly from the digitized noise signal $s[i]$ without modifications.

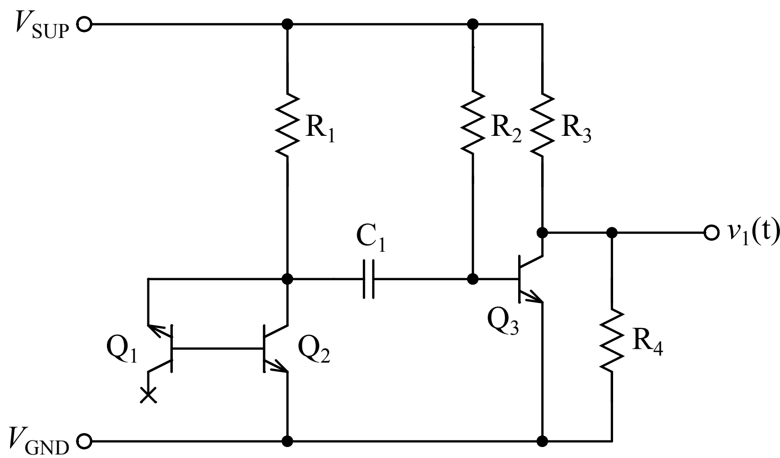


Fig. 2.7: The noise source based on noise occurring in the reverse polarized P-N junction [25]

Random number sequences generated by this TRNG were evaluated by the FIPS [63] and NIST [64] statistical test suites. The NIST test suite found a small bias in sequences

generated without corrector. However, this bias was removed to a large extent by the correctors, especially by the von Neumann corrector.

2.3 TRNG Based on Ring Oscillators

A pure digital solution of randomness extraction is based on the phenomenon commonly-called the jitter, which is described in section 2.1.2. This type of the TRNGs was described in detail in [27], allows efficient design without any analog parts, and is suitable for FPGAs and ICs. Its implementation in an FPGA was presented in [28]. The proposal of the source of randomness is shown in figure 2.8.

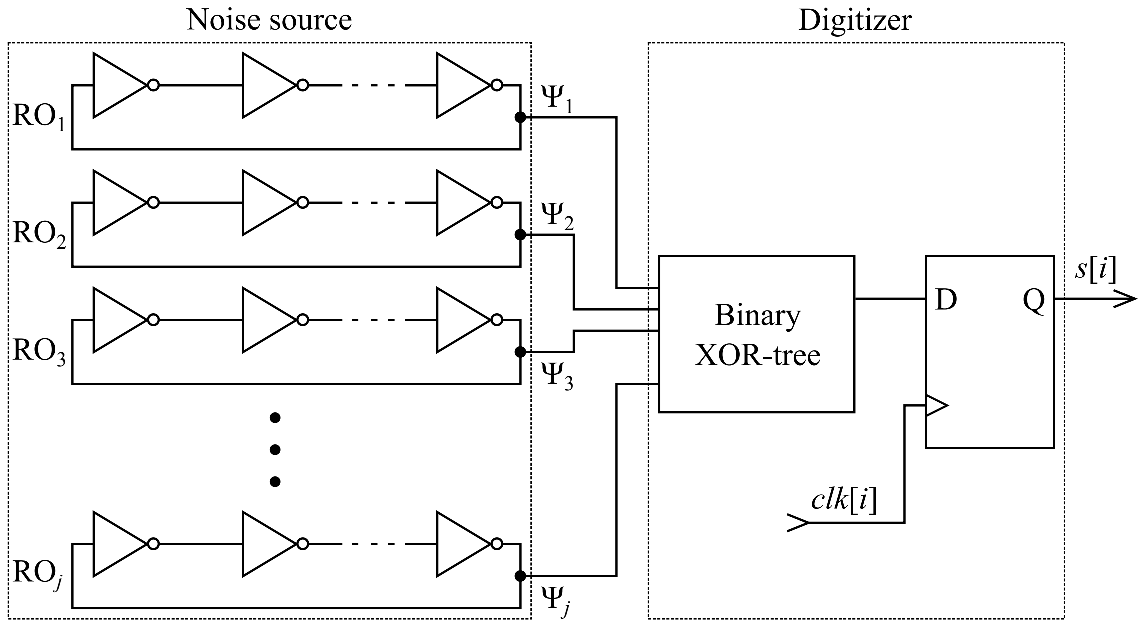


Fig. 2.8: The source of randomness based on a architecture with ring oscillators [27]

The fundamental part of this solution is the ring oscillator (RO), which is a sources of a square waveform signal exhibiting the jitter in rising or falling edges. The ring oscillator is composed of an odd number of inverters arranged in the rings. The output of each inverter oscillates between logic zero and logic one. Thereby the periodic square waveform signal $\Psi_j(t)$ is formed and can be described as

$$\Psi_j(t) = \Psi_j(t + \Theta_{\mathrm{RO}}) \tag{2.27}$$

where j is a number of the ring oscillator, Θ_{RO} is a period of its output signal, which is determined by the number of inverters ξ and the delay of the inverters τ_{inv}. Thus the

period Θ_{RO} can be expressed as

$$\Theta_{\mathrm{RO}} = \xi \tau_{\mathrm{inv}}. \tag{2.28}$$

However, in the real world, the output signal $\Psi_j(t)$ does not have the ideal square waveform. The period Θ_{RO} is not fixed but consists of a constant part $\Theta_{\mathrm{RO,c}}$ and of a random variable part $\Theta_{\mathrm{RO,r}}$, whose value occurs in the range from $-\Theta_{\mathrm{RO,c}}/2$ to $\Theta_{\mathrm{RO,c}}/2$ with the normal distribution and the mean value in the middle of this interval. Thus the period Θ_{RO} is given by

$$\Theta_{\mathrm{RO}} = \Theta_{\mathrm{RO,c}} + \Theta_{\mathrm{RO,r}}. \tag{2.29}$$

The random variable part $\Theta_{\mathrm{RO,r}}$ – the jitter – appears around the rising edge or the falling edge of the output signal $\Psi_j(t)$ and can be divided into a random variable part around the rising edge $\Theta_{\mathrm{RO,r,r}}$ and a random variable part around the falling edge $\Theta_{\mathrm{RO,r,f}}$. Similarly the constant part $\Theta_{\mathrm{RO,c}}$ can be divided into a constant part of logic one $\Theta_{\mathrm{RO,c,O}}$ and a constant part of logic zero $\Theta_{\mathrm{RO,c,Z}}$. The output signal $\Psi_j(t)$ with all described parts is depicted in figure 2.9. The jitter width depends on the number of used inverters ξ [28].

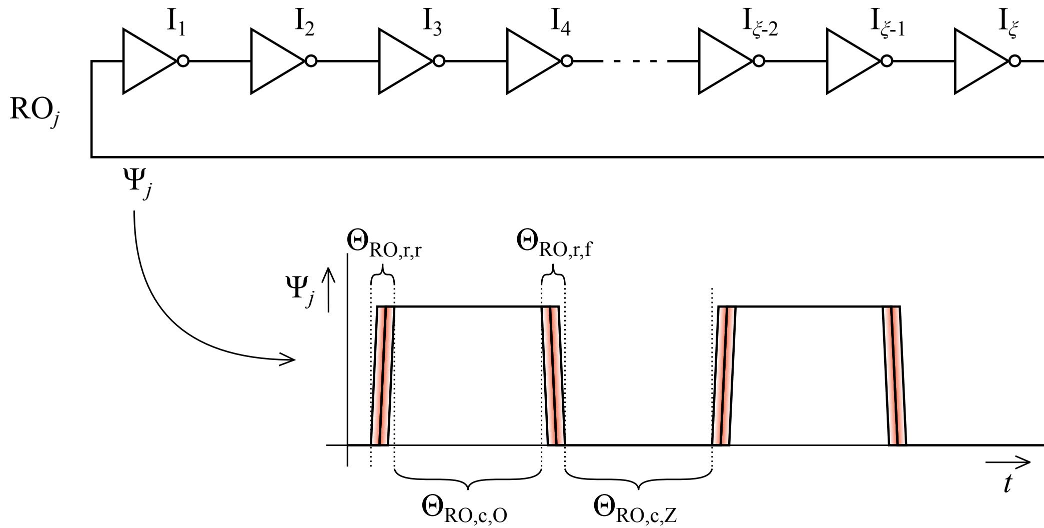


Fig. 2.9: The RO output signal $\Psi_j(t)$ with the jitter displayed

It is not appropriate to sample the jitter directly because this part of the periodic square waveform signal $\Psi_j(t)$ is very short and so the deterministic part of this signal could be sampled. The result would be that the digitized noise signal would contain a significant deterministic component and the quality of generated random number sequences would be low. Therefore a more complex method described in [27] is usually used. More ring

oscillators produce more the square waveform signals $\Psi_j(t)$ that are processed in the XOR tree, which is composed of the standard XOR logic gates. The digitized noise signal $s[i]$ is formed by sampling the XOR tree product.

A part of the TRNG based on ring oscillators is the post-processing block, in which the method of bias reduction called the resilient function is implemented. This function is described in section 2.1.3. Successful results of the NIST and Diehard [65] statistical test suites confirmed fine properties of random number sequences generated by the described TRNG with the random data rate of 2.5 Mb/s [28].

TRNGs based on ring oscillators are usually composed of standard digital cells. Hence they are suitable for implementation in FPGAs and ICs. However, to work it correctly, this type of TRNGs use a large number of ROs and proper timing of internal signals must be ensured. Thus this TRNG type has significant placement constraints. Also with the large number of ROs, high power consumption is connected. Based on these inconveniences, this type is not suitable for mobile and hand-held devices. Moreover, immunity to power supply distortion is weak. By signal injected into the power supply, frequencies of ROs can be locked, and the jitter can be eliminated [103].

2.4 Metastability-Based TRNG

Randomness can also be extracted from circuits with a so-called metastable behavior. In modern systems during recent years, the number of applications of the metastability based TRNGs in ICs grows primarily due to their compact design [41], [45], [46], or [48]. For an easy description of the metastable system, a mechanical analogy shown in figure 2.10 can be used. Two ideally slant planes are symmetrically inclined to each other. An ideal spherical bead is precisely placed on the created top. If any external noises do not influence this system, the bead stays on the top. This state can be called as the metastable state. However, if some external noises influence this system, the bead falls to a stable position. The stable positions are on both sides of the slant planes and are marked with the numbers “0” and “1”, which represent logic zero and logic one. External noises can have a deterministic character or a random character. If the random character prevails over the deterministic character, then the selection of the side, on which the bead ended, can be considered as random.

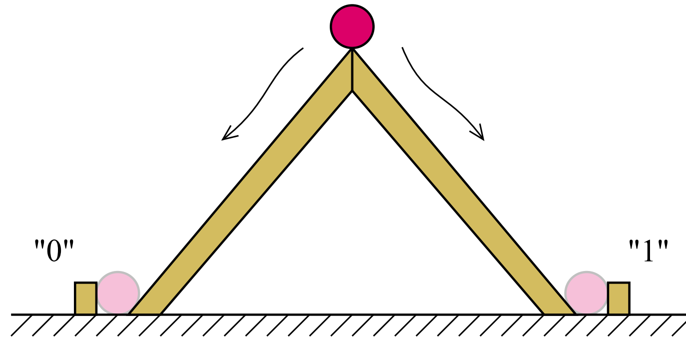


Fig. 2.10: The mechanical analogy of metastable systems

A TRNG with an architecture corresponding to the above-mentioned mechanical analogy was published in [41]. This circuit composed of a pair of cross-coupled inverters and two pre-charged metal-oxide-semiconductor field effect transistors (MOSFETs) is a typical structure exhibiting the metastable behavior, which is shown in figure 2.11. The pair of P-channel MOSFETs M_1 and M_2 initialize the metastable state in the circuit. Both transistors M_1 and M_2 are controlled by a square waveform clock signal, which is represented by V_{CLK}. When the clock signal is in logic zero, signals in both nodes “a” and “b” switch to logic one. Then when the rising edge of the clock signal appears, the circuit goes to the metastable state. In other words, voltages in both nodes V_a and V_b drop to a metastable voltage V_{meta}. Random noise present in the nodes “a” and “b” causes a transition of the circuit to the stable state. Then, the resulting logic levels in both nodes are dependent on noise difference between the nodes during the metastable state. Specifically, thermal and flicker noise described in section 2.6 occurs in channels of MOSFETs fabricated in CMOS technologies. Thus the logic levels are generated randomly. Thanks to the circuit structure, a new random bit can be generated each period of the clock signal. The digitized noise signal $s[i]$ is usually formed by a sampling of voltages V_a and V_b.

Waveforms of signals in the above-described noise source are depicted in figure 2.12. Voltages V_a and V_b in the nodes “a” and “b” are formed in the following sequence. In the first part, the circuit is reset by switched on MOSFETs M_1 and M_2. So in fact, the voltages V_a and V_b are connected to the power supply V_{SUP}. Then after switching off the transistors M_1 and M_2, the voltages V_a and V_b go through the metastable voltage V_{meta} into a stable state based on present noise. Time of circuit flipping is known as the resolution time and is defined as the time, which the metastable system needs for the transition

from the metastable state to the stable state. The resolution time t_r is given by several parameters [39] and can be modeled as

$$t_r = \tau_{meta} \ln \left(\frac{\kappa_f \Delta V_f}{\Delta V_i} \right) \quad (2.30)$$

where τ_{meta} and κ_f are system dependent constants, ΔV_f is a final voltage difference in the nodes “a” and “b” in the stable state, and ΔV_i is a initial voltage difference in the metastable state. Theoretically, if there were no noise in the nodes “a” and “b”, the described circuit would not go into the stable state. However, in reality, there is always present some type of noise, and the circuit flips into the stable state.

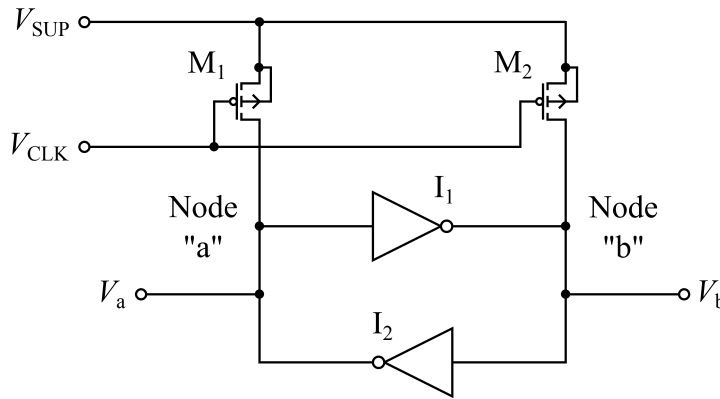


Fig. 2.11: Waveforms of signals in the metastability based noise source [41]

This type of TRNGs allows compact design especially suitable for ICs. During the metastable state, the power consumption is higher, but in rest of period, the power consumption is comparable to the consumption of conventional digital circuits. The metastability based TRNGs are resistant to deterministic noise of power supply, which is injected into the circuit during the metastable state [41]. However, the metastable systems are very sensitive to a mismatch between core devices such as the cross-coupled inverters of the above-mentioned TRNG. In the extreme case, the mismatch can cause that the random behavior changes to deterministic and the system fails. In AMS design of ICs, the mismatch between devices can be eliminated by proper physical design, but which can be can be very time-consuming. Therefore it is advisable to use AMS physical design assistants or automatized functions [10], [13], or [8]. Due to its properties, this type of TRNGs is suitable for modern hand-held devices.

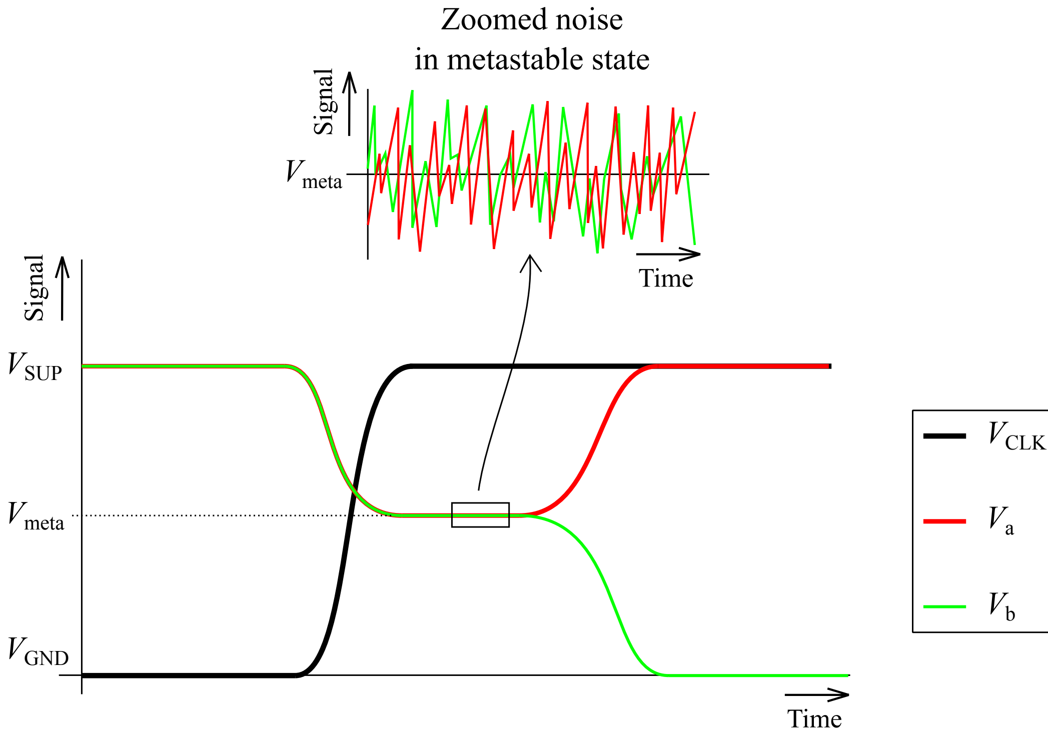


Fig. 2.12: The noise source exhibiting the metastable behavior [98]

2.5 TRNG Based on Chaos

Random number sequences can be generated by TRNGs, which are based on chaos arising in non-linear dynamical systems. A principle of the chaos based TRNGs is well described in [58]. The non-linear dynamical systems are defined by deterministic equations, but they are very sensitive to changes of initial conditions. Their behavior has exponentially divergent aperiodic characteristic given by the so-called positive Lyapunov exponents. Noise present in the non-linear dynamical systems continuously affects the initial conditions and causes unpredictable output values. Moreover, these systems can be implemented in circuits fabricated in commercially available CMOS and BCD technologies [58], [59], in FPGAs [61], or in circuits composed of discrete components [54]. So the non-linear dynamical systems are suitable for utilization in noise sources of TRNGs integrated into ICs.

The non-linear dynamical systems suitable for TRNGs can be categorized from the viewpoint of the processed signal on the systems working in continuous time and the systems working in discrete time. The continuous time systems are described by differential

equations and a typical representative of TRNGs based on these systems is the TRNG using the double-scroll attractor published in [54]. These TRNGs are usually composed of analog blocks such as operational amplifier, which can mean that they require a larger silicon area.

The discrete time systems are described by difference equations and are usually composed of lower number of components. A fundamental architecture of the TRNG with the discrete time non-linear dynamical system is shown in figure 2.13. The noise source consists of a circuit with a non-linear transfer function and a sample and hold circuit. A signal produced by the noise source is digitized by a clocked comparator, which thus generates the digitized noise signal $s[i]$.

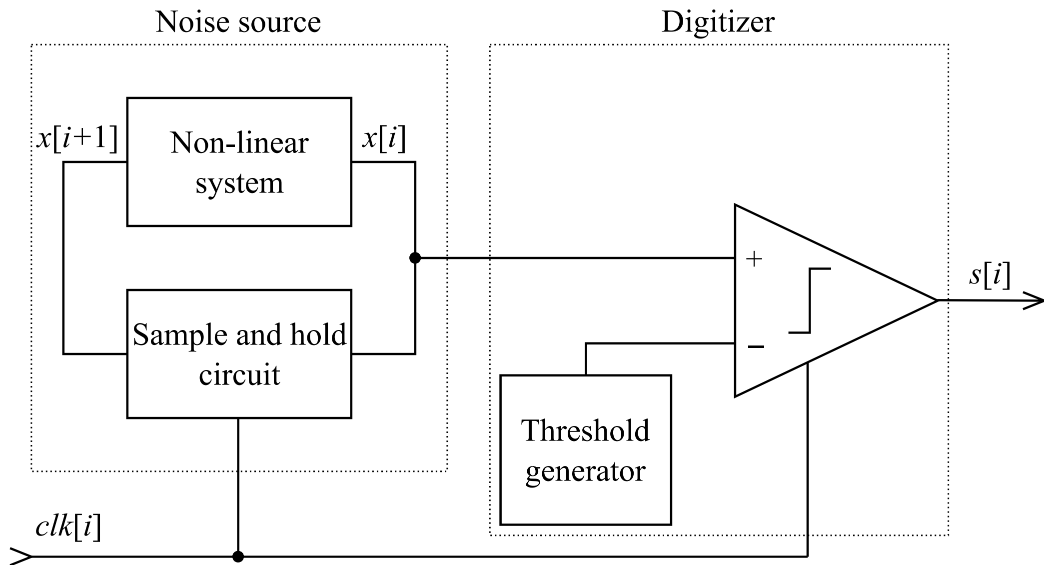


Fig. 2.13: The source of randomness based on the discrete time non-linear dynamical system [58]

An example of this TRNG type was published in [58]. As the discrete time non-linear dynamical system, a CMOS compatible current mode circuit using skew tent map was proposed and is depicted in figure 2.14. The skew tent map is defined by

$$x\,[i+1] = \begin{cases} \zeta x\,[i], & 0 \leq x\,[i] \leq \frac{1}{\zeta} \\ \frac{\zeta}{\zeta-1}\,(1 - x\,[i]), & \frac{1}{\zeta} < x\,[i] \leq 1 \end{cases} \tag{2.31}$$

where ζ is a skew tent map coefficient, $x\,[i]$ is an actual value, and $x\,[i+1]$ a future value. The current mirror composed of N-channel MOSFETs M_{N1} and M_{N2} copies a reference

current I_{REF}. Similarly, the input current I_{IN} is mirrored by P-channel MOSFETs M_{P1}, M_{P2} and M_{P3}. The current difference between currents flowing through M_{P3} and M_{N2} is mirrored by N-channel MOSFETs M_{N3} and M_{N4}. The output current I_{OUT} is formed by combination of currents flowing through M_{P3} and M_{N4}.

The sample and hold circuit working on the current mirror principle samples the output current I_{OUT} and iterates the one-dimensional skew tent map. The ratio of the current mirror composed of M_{N5} and M_{N7} is the main chaos controlling parameter. Results of a transient simulation and a phase portrait plot taken from [58] are shown in figure 2.15 respectively in figure 2.16. Random number sequences generated by this TRNG were tested by the NIST test suite [64] and passed all the tests.

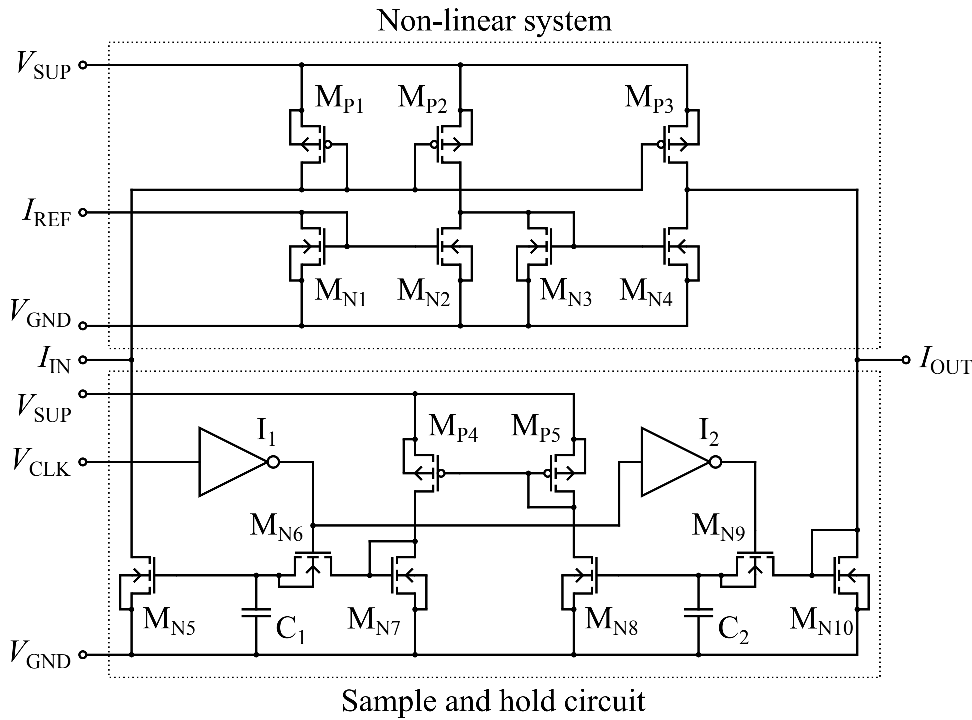


Fig. 2.14: The noise source as the CMOS compatible current mode circuit published in [58]

The chaos controlling parameter must be very precisely tuned so that the chaotic based TRNG operates in the chaotic regime. Variations of this parameter can reduce the quality of random number sequences and, in extreme case, system failure. Therefore tolerances of the chaos controlling parameter must be large enough. Chaos controlling parameter

variations are a weak point of the TRNGs based on chaos. Nevertheless, these TRNGs can be integrated into ICs and are suitable for mobile devices.

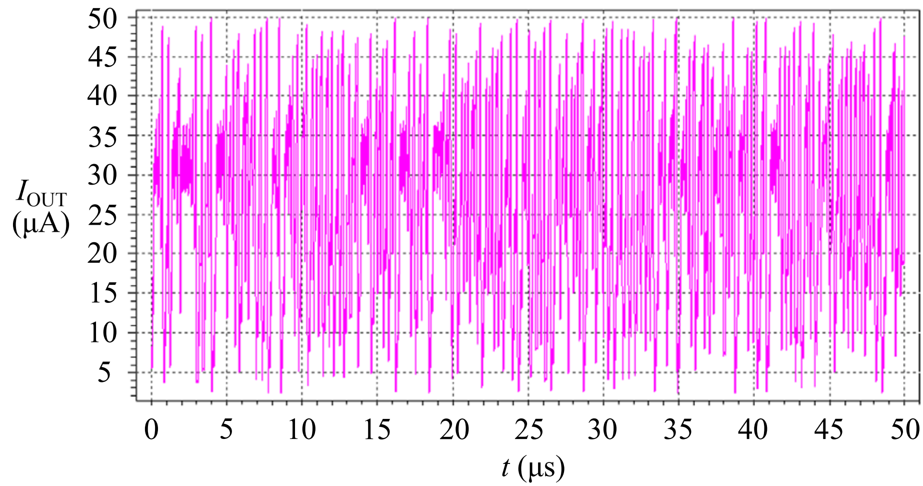


Fig. 2.15: The transient simulation of the noise source published in [58]

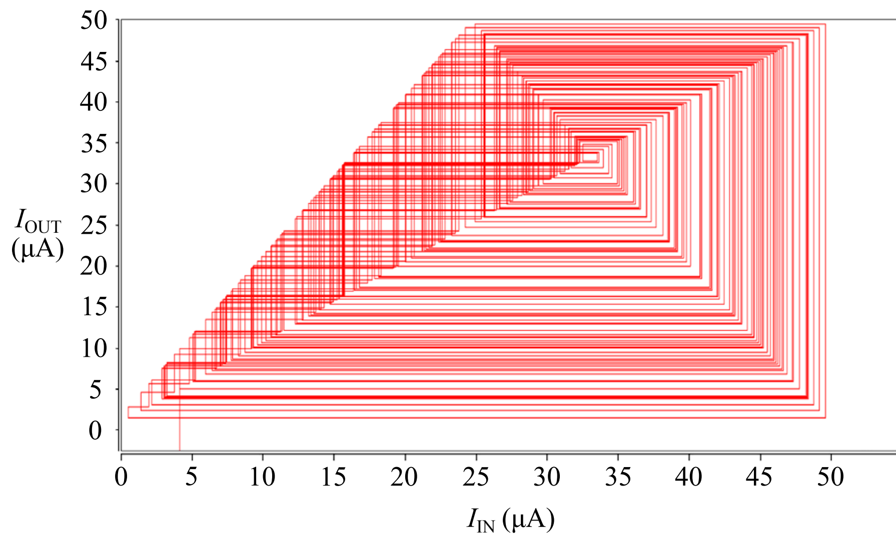


Fig. 2.16: The phase portrait plot of the noise source published in [58]

2.6 Noises Occurring in Electronic Circuits

TRNGs suitable for implementation in SoCs fabricated in CMOS or BCD technologies exploit physical phenomena with the random behavior specifically noises occurring in electronic semiconductor circuits. Therefore, in this section, electronic noises most fre-

quently used in the TRNGs are summarized and described. A similar summary can also be found in [98].

Electric noise occurs in all real electronic circuits and devices and can be observed as random disturbances in electrical signals. In circuit theory, electrical signals are explicitly described by mathematical equations, and it is possible to determine their values at all moments. However, it is not possible to predict any future values of noise. Therefore noises are generally described by a probabilistic approach thus by characteristics such as the mean value, the variance, or the standard deviation. In most cases, the mean value of some noise is zero. Thus the noise variance corresponds to the noise power. Amplitude distribution of noise is characterized by the probability density function (PDF).

Noise can be classified from the point of view of the mechanism of origin and contribution to total noise. Some types of noise are always present in circuits due to its physical nature, but some types are dependent on the quality of manufactured devices, which is given by, for example, defects of the gate oxide or the silicon bulk in CMOS technologies.

2.6.1 Thermal Noise

Thermal noise described by Johnson [104] and by Nyquist [105] is generated by a spontaneous motion of charge carriers in a real electrical conductor. Thus it arises regardless of an applied voltage on a real device. The charge carriers are usually electrons, but thermal noise appears in other types of conducting materials where the charge carriers are, for example, ions in an electrolyte. Thermal noise is a kind of white noise with the normal amplitude distribution and the constant power spectral density (PSD) throughout a frequency range, which is expressed as

$$\frac{\mathrm{d}\overline{v_{\mathrm{n,th}}^2}}{\mathrm{d}f} = 4k_{\mathrm{B}}TR \tag{2.32}$$

where k_{B} is the Boltzmann constant, T is ambient temperature, and R is total resistance of the real conductor. In CMOS or BCD technologies used for applications abovementioned, thermal noise occurring in channels of MOSFETs can be modeled as an equivalent input noise voltage source [106] with the PSD

$$\frac{\mathrm{d}\overline{v_{\mathrm{n,th}}^2}}{\mathrm{d}f} = 4k_{\mathrm{B}}T\gamma\frac{1}{g_m} \tag{2.33}$$

where g_m is MOSFET transconductance, and γ is a technology dependent coefficient, whose value depends on the used technology. Often used value is 2/3, but increases for submicron CMOS technologies [107].

The root mean square (RMS) of the thermal noise voltage is used to set parameters of noise sources implemented in behavioral models of TRNGs. A real resistor can be modeled as an ideal resistor in series with the thermal noise voltage source. Then the RMS value can be calculated according to

$$V_{\mathrm{n,th},R} = \sqrt{4k_{\mathrm{B}}TR\Delta f} \tag{2.34}$$

where Δf is an assumed frequency bandwidth. Thermal noise also occurs in real capacitors and is known as kT/C noise. The real capacitor can be modeled by the ideal resistor R, the ideal capacitor C and the thermal noise voltage source connected as a one-pole low pass filter. Then the RMS of voltage V_C on the capacitor C is given by

$$V_{\mathrm{n,th},C} = \sqrt{\frac{k_{\mathrm{B}}T}{C}}. \tag{2.35}$$

Derivation of $k_{\mathrm{B}}T/C$ noise can be found in [98]. Thermal noise occurring on resistors is basis for the TRNG described in [15].

2.6.2 Flicker Noise

Flicker noise is a fluctuation of the conductance with the PSD, which can be described by $f^{-\eta}$ where η equals 1 ± 0.1 in a wide frequency range [108]. It is also often referred to as $1/f$ noise according to its specific PSD and is considered a kind of pink noise [109]. Its origin has not yet been fully elucidated and is subject to discussion. This noise cannot be observed at higher frequencies because it disappears in above-mentioned thermal noise, which is always present and has the flat PSD.

Fluctuations of resistance can be seen, among other things, in real resistors. Generated noise is proportional to the DC current flowing through the resistor and is also dependent on the quality of fabricated components. The flicker noise PSD of a resistor R [110] can be described by

$$\frac{\mathrm{d}\overline{v_{\mathrm{n,f},R}^2}}{\mathrm{d}f} = \frac{K_{\mathrm{f},R}R_{\square}V_R^2}{W_R L_R}\frac{1}{f} \tag{2.36}$$

where $K_{\mathrm{f},R}$ is a material-dependent coefficient of flicker noise, $R_{\square}$ is sheet resistance, V_R is voltage of the resistor R with dimensions W_R and L_R. Resistors fabricated from

homogeneous material produce smaller flicker noise. Specifically, diffusion resistors from the crystalline silicon have smaller coefficient $K_{\mathrm{f},R}$ than poly-silicon resistors [106].

Flicker noise can be observed in all electronic devices such as in base current of bipolar transistors, current of diodes, or even cathode current of vacuum tubes. Also MOSFETs exhibit this type of noise due to a contribution of surface states occurring on the boundary between the silicon crystal and the gate oxide. Thus flicker noise of MOSFETs can be modeled as an equivalent input noise voltage source [106] with the PSD

$$\frac{\mathrm{d}\overline{v_{\mathrm{n,f}}^2}}{\mathrm{d}f} = \frac{K_\mathrm{f}}{WLC_{\mathrm{ox}}^2}\frac{1}{f} \tag{2.37}$$

where K_f is a flicker noise coefficient with low technology dependence, C_{ox} is the gate oxide capacitance per unit area, W is the gate width and L the gate length.

2.6.3 Shot Noise

Unlike flicker noise, the origin of shot noise is known. Shot noise occurs in semiconductor materials with junctions. It is associated with DC current flowing through a semiconductor component with a junction and is caused by passages of charge carriers across the junction. Charge carriers with sufficient velocity and energy pass the junction at random time. Thus, in fact, the current flowing through the junction is composed of a large number of random independent pulses.

Shot noise has the PDF corresponding to the normal distribution and is considered white [109]. Thus its PSD is flat and is given by

$$\frac{\mathrm{d}\overline{i_{\mathrm{n,s}}^2}}{\mathrm{d}f} = 2q_e I \tag{2.38}$$

where q_e is the elementary charge and I is the current flowing through a device [108]. Also MOSFETs in subthreshold regime exhibit shot noise while its RMS value [111] can be described by

$$I_{\mathrm{n,f,D}} = 2q_e I_{\mathrm{sat}}\left(1 + e^{\frac{-q_e V_{\mathrm{DS}}}{kT}}\right)\Delta f \tag{2.39}$$

where V_{DS} is the drain-source voltage of a MOSFET and I_{sat} is its saturation current in subtreshold conduction defined in [112].

2.6.4 Generation-Recombination Noise

Generation-recombination (GR) noise described in [108] occurs in semiconductor components and is caused by fluctuations of free electrons in the conduction band. The free electrons fluctuate due to generation and recombination process between the band and traps. Thus GR noise is a fluctuation of the conductance G with the spectral density S_G described by

$$\frac{S_G(f)}{G^2} = \frac{S_{N_\mathrm{cc}}(f)}{N_\mathrm{cc}^2} = \frac{\overline{(\Delta N_\mathrm{cc})^2}}{N_\mathrm{cc}^2} \frac{4\tau_\mathrm{r}}{1 + (2\pi f \tau_\mathrm{r})^2} \tag{2.40}$$

where τ_r is a relaxation time characteristic for the trap, N_cc is the number of charge carriers, and $\overline{(\Delta N_\mathrm{cc})^2}$ is their variance, which is given for one type of traps by

$$\frac{1}{\overline{(\Delta N_\mathrm{cc})^2}} = \frac{1}{N_\mathrm{cc}} + \frac{1}{N_\mathrm{ot}} + \frac{1}{N_\mathrm{et}} \tag{2.41}$$

where N_ot is the average number of occupied traps and N_et is the average number of empty traps.

2.6.5 Random Telegraph Signal Noise

Random telegraph signal (RTS) noise also frequently called burst noise or popcorn noise is another type of noise, which is significant in very small MOSFETs and at low frequencies. This noise can be observed as signal switching between two discrete levels at random time. It is caused by trapping of an individual carrier in a single active trap in the oxide, or by a scattering center in the region of the inversion layer of the device [113]. Trapping centers in the vicinity of Fermi levels generate RTS noise and are the result of heavy metal ions contamination or defects of crystalline lattice [114]. If there are more traps in near-interface of the gate oxide, RTS noise is more complicated and randomly switches among more discrete levels [115].

The PSD of RTS noise [113] can be expressed by

$$\frac{\mathrm{d}\overline{i_\mathrm{n,rts}^2}}{\mathrm{d}f} = \frac{4K_\mathrm{rts}\tau_\mathrm{c}\Delta I_\mathrm{D}^2}{1 + (2\pi f \tau_\mathrm{c})^2} \tag{2.42}$$

where K_rts is a coefficient of RTS noise, I_D is drain current of a MOSFET, and τ_c is the characteristic time constant of the trap given by

$$\frac{1}{\tau_\mathrm{c}} = \frac{1}{\overline{\tau_\mathrm{l}}} + \frac{1}{\overline{\tau_\mathrm{h}}} \tag{2.43}$$

where $\overline{\tau_l}$ is the average time of captures and $\overline{\tau_h}$ is the average time of emissions. The coefficient of RTS noise K_{rts} represents the trap occupation probability, which can be calculated according to

$$K_{rts} = \frac{\overline{\tau_h \tau_l}}{(\overline{\tau_h} + \overline{\tau_l})^2}. \tag{2.44}$$

A TRNG published in [116] is based on RTS noise, which causes perturbation of a Sigma-Delta modulator. This TRNG was integrated into a chip and fabricated in a 55 nm process.

2.6.6 Avalanche Noise

Another type of noise – avalanche noise – occurs in semiconductor devices with reverse-biased P-N junctions and is well described in [117]. The condition is that the P-N junction is biased by a strong electric field. Thus a free charge carrier acquires sufficient energy in the avalanche zone and has a significant probability that it may collide with a crystalline lattice and create a new hole-electron pair in the zone. If there is the sufficiently strong electric field then one or both carriers probably cause another collision and generate another hole-electron pair before they leave the zone. In this way, a steady avalanche is created and is free of noise.

However, in reality, some carriers with high energy do not produce any hole-electron pair, and some produce two or more hole-electron pairs while their average number is the same as in the steady avalanche. This process creates a fluctuation of the avalanche. Thus avalanche noise is composed of a random sequence of avalanche fluctuations. Its PSD can be described by

$$\frac{\mathrm{d}\overline{i_{n,a}^2}}{\mathrm{d}f} = \frac{2q_e I}{(2\pi f \tau_a)^2} \tag{2.45}$$

where I is the current flowing through the P-N junction and τ_a is the average interval between two consecutive hole-electron pair generations.

Avalanche noise can be used in TRNGs composed of discrete components or fabricated in BCD technologies. A solution of a discrete TRNG presented in [118] compares outputs from two noise sources, which are based on diodes producing avalanche noise.

Chapter 3

Evaluation of Random Number Sequences

Modern communication systems are secured by different session keys, hash functions, or key protocols [1], which are based on random numbers. Therefore parts of these systems are RNGs namely PRNGs or TRNGs producing random numbers in the form of logic ones and zeros. The generators based on the random behavior of physical phenomena – the TRNGs – provide higher security level. Features of the RNGs are evaluated, among other things, from properties of generated random number sequences. As already mentioned in chapter 2, it is possible to evaluate only random number sequences, not the individual random numbers. The number sequences generated by a TRNG have to have properties, which are expected from random numbers sequences. For example, the sequence should be composed of the same number of logic ones and zeros while this property results from an assumption that each new bit is generated with the probability equal to 1/2 that its value will be logic one. It is also necessary in order no deterministic fluctuations occur in the produced sequences. Any deviations from the expected properties might be deliberately misused. In this way, the communication systems might be attacked.

Properties of random number sequences are classified by statistical test suites. Functionality of most TRNGs mentioned in section 1.3 was tested by A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications [64] (the NIST test suite) and FIPS PUB 140-2, Security Requirements for Cryptographic Modules [63] (the FIPS test suite) both defined by the National Institute of Standards and Technology. Therefore these test suites are used to evaluate the proposed TRNG structures, which are described in the following chapters. This chapter explains basic

principles of the statistical test suites. The quality of random number sequences can also be evaluated using entropy introduced by Shannon in [119], which is also mentioned here.

3.1 NIST Test Suite

The NIST test suite is designed to detect different types of non-randomness in arbitrarily long number sequences produced by RNGs. This test suite is a package consisting of 15 statistical test and some of them are composed of other subtests. All tests defined in [64] are briefly described in the following sections. Source codes in the C programming language were provided by the National Institute of Standards and Technology.

Each test tests a null hypothesis, which in this case is that the tested number sequence is random. Thus an alternative hypothesis is that the tested sequence is not random. For each test, a suitable reference distribution is used. From the reference distribution, a critical value is determined. From the tested sequence, a value of test statistic is computed and compared to the critical value. If the value of test statistic exceeds the critical value, the null hypothesis is rejected. Otherwise, the null hypothesis is not rejected [64].

The tests use the normal distribution or the chi-square (χ^2) distribution as reference distributions. If some non-randomness is revealed, the calculated test statistic falls in extreme regions of the reference distribution. The test statistic for the normal distribution is described by

$$z_{\mathrm{t}} = \frac{(x_{\mathrm{t}} - \mu)}{\sigma} \tag{3.1}$$

where x_{t} is the sample test statistic value, μ is the mean value of the test statistic, and σ^2 is the variance of the test statistic. Good fit of observed occurrences of a sample measure and expected occurrences of the hypothesized distribution is classified by the χ^2 distribution. Thus the test statistic for the χ^2 distribution is expressed by

$$\chi^2 = \sum_{q=1}^{N_{\mathrm{o}}} \frac{\left(o_{\mathrm{o},q} - o_{\mathrm{e},q}\right)^2}{o_{\mathrm{e},q}} \tag{3.2}$$

where N_{o} is the number of occurrences, o_{o} are observed occurrences, and o_{e} are expected occurrences.

The test statistic is used for calculation of the so-called P-value, which is the probability that a perfect RNG would have produced a sequence less random than the tested sequence, given the kind of non-randomness assessed by the test [64]. The P-value equal

to 0 means that the tested sequence is absolutely non-random. On the other hand, the P-value equal to 1 determines that the sequence is entirely random. A decisive value is given by a significance level α, which is usually equal to 0.01. This value has been used for all tests in this work. Thus the tested sequence is considered random when the P-value is equal to or greater than α. Otherwise, if the P-value is less than α, the null hypothesis is rejected.

The P-value is calculated from the test statistic using so-called special functions. If the reference distribution is normal, the P-value is computed by the error function

$$\mathrm{erfc}\,(z_{\mathrm{t}}) = \frac{2}{\sqrt{\pi}} \int_{z_{\mathrm{t}}}^{\infty} e^{-x^2} \mathrm{d}x. \tag{3.3}$$

In the second case, when the χ^2 distribution is reference, the P-value is determined by the incomplete gamma function

$$\mathrm{igamc}\left(a_{\mathrm{t}}, \frac{\chi^2}{2}\right) = \frac{\int_{\frac{\chi^2}{2}}^{\infty} y^{a_{\mathrm{t}}-1} e^{-y} \mathrm{d}y}{\Gamma\,(a_{\mathrm{t}})} \tag{3.4}$$

where a_{t} is a test parameter and $\Gamma\,(a_{\mathrm{t}})$ is the gamma function defined by

$$\Gamma\,(a_{\mathrm{t}}) = \int_{0}^{\infty} u^{a_{\mathrm{t}}-1} e^{-u} \mathrm{d}u. \tag{3.5}$$

In the provided algorithm, the exact reference distributions are replaced by asymptotic distributions to be possible to compute the desired values. Therefore, for each test, a recommended minimum length of the test sequence is defined. For testing most TRNGs mentioned in section 1.3, number sequences with the length 1 Mb were used.

3.1.1 Monobit Test

The Monobit test is based on the assumption that the random number sequence contains the same number of logic ones and logic zeros. Therefore this test checks their numbers in the tested sequence and decides if the numbers are approximately the same. The reference distribution is normal for this test. The minimum length of the tested sequence should be 100 b. This test is crucial. If this test fails, then other tests will probably also fail.

3.1.2 Frequency Test within a Block

The Frequency test within a block tests the ratio between ones and zeros within M_{f}-bit blocks. The number of ones in the M_{f}-bit block should be approximately $M_{\mathrm{f}}/2$ while M_{f} is an input parameter of this test. The block size M_{f} should be equal to or greater than 20 and greater than $0.01N_{\mathrm{ts}}$ where N_{ts} is the length of the tested sequence. When M is set to 1, this test degenerates to the Monobit test above-mentioned. The tested sequence should be longer than 100 b. For the test statistic, the reference distribution is the χ^2 distribution.

3.1.3 Runs Test

The Runs test detects the total number of uninterrupted sequences of identical bits with different lengths. The number of runs with different lengths should be as expected for the random sequence. This test is also able to determine the rate of oscillation between values partially. The recommended minimum length of the tested sequence is 100 b. The reference distribution for the test statistic is the normal distribution.

3.1.4 Test for the Longest Run of Ones in a Block

The Test for the longest run of ones in a block finds out the longest run of ones within M_{l}-bit blocks and determines if its length is consistent with the length of the longest run occurring in the random sequence. The minimum length of the tested sequence is dependent on the block length M_{l}. If M_{l} is 8 b, the minimum length of the tested sequence is 128 b. If M_{l} is 128 b, the minimum length is 6 272 b. Finally, if M_{l} is 10^4 b, the minimum length is $75 \cdot 10^4$ b. In this case, the χ^2 distribution is the reference.

3.1.5 Binary Matrix Rank Test

The test called the Binary matrix rank test determines linear dependence in the tested sequence, which is divided into sub-strings with fixed length Q_{b}. From the sub-strings, disjoint matrices are formed while M_{b} is a number of rows, Q_{b} is a number of columns, and the default value of M_{b} and Q_{b} is set to 32 b. Then linear dependence in each matrix is analyzed using the binary rank. The recommended minimum length of the tested sequence has to be higher equal to or greater than $38M_{\mathrm{b}}Q_{\mathrm{b}}$. Thus the minimum length

is 38 912 b for the default values M_b and Q_b. Also for this test, the χ^2 distribution is the reference.

3.1.6 Discrete Fourier Transform Test

The Discrete Fourier transform test detects periodic features appearing in the tested sequence. It is mainly focused on repetitive patterns, which are near each other. Heights of peaks produced by the Discrete Fourier transform are compared with a threshold value. This test is based on the assumption that 95 % of the peaks in the random sequence do not exceed the threshold value. The reference distribution is the normal distribution. The length of the tested sequence should be at least 1 000 b.

3.1.7 Non-overlapping Template Matching Test

The Non-overlapping template matching test evaluates too many occurrences of predefined non-periodic patterns with the length Q_{no}. The pattern is searched in an Q_{no}-bit window. The window slides one-bit position when the pattern is not found. Otherwise, the window is moved to the first following bit, and a new search is started. So no overlaps are created. During the test, the tested sequence is divided into eight M_{no}-bit blocks, which are further processed. The recommended and usually used length of the non-periodic patterns Q_{no} is 9 b. Another condition for valid test results is that M_{no} is greater than $0.01 N_{ts}$. For this test, the reference distribution is the χ^2 distribution.

3.1.8 Overlapping Template Matching Test

The Overlapping template matching test uses the same mechanism as the test abovementioned but with one difference and that the window slides only one bit when the predefined non-periodic pattern is found. So this procedure creates overlaps. The recommended minimum length of the tested sequence is 10^6 b. As in the previous case, the reference distribution is the χ^2 distribution.

3.1.9 Maurer's "Universal Statistical" Test

The Maurer's "Universal Statistical" test verifies if the tested sequence can be significantly compressed without loss of information. This test is based on the assumption that

the random sequence cannot be significantly compressed. For this test, the reference distribution is the so-called half-normal distribution, which is a one-sided case of the normal distribution. The tested sequence should be at least 387 840 b long.

3.1.10 Linear Complexity Test

Some RNGs generate random number sequences by LFSRs. Therefore the Linear complexity test detects the length of the LFSR and evaluates if the tested sequence is sufficiently complex. The sequence is non-random when the detected length is too short. For the test statistic, the reference distribution is the χ^2 distribution. The recommended minimum length of the tested sequence is 10^6 b.

3.1.11 Serial Test

The frequency of all possible overlapping M_s-bit patterns in the tested sequence is determined by the Serial test, which is based on the assumption that each M_s-bit pattern can appear with the same probability as like any other M_s-bit pattern. Thus this test detects uniformity of occurrences of all M_s-bit patterns. The recommended minimum length of the tested sequence is given by $M_s < (\log_2 N_{ts}) - 2$. The reference distribution is the χ^2 distribution.

3.1.12 Approximate Entropy Test

The Approximate entropy test determines frequencies of all possible overlapping M_{ApEn}-bit and $\left(M_{ApEn} + 1\right)$-bit patterns in the tested sequence. Then, on the basis of the so-called Approximate Entropy (ApEn) described in [120], the detected frequencies are compared with expected results for the random sequence. The reference distribution for the test statistic is the χ^2 distribution and the recommended minimum length of the tested sequence is given by $M_{ApEn} < (\log_2 N_{ts}) - 5$.

3.1.13 Cumulative Sums Test

The Cumulative sums test evaluates cumulative sums of the tested sequence, which is adjusted before processing so that values 0 are replaced by -1. The cumulative sum of partial sub-sequences also named the excursion from zero of the random walk should

be near zero for the random sequence. A more significant value of the excursion indicates non-random properties of the tested sequence. The reference distribution is the χ^2 distribution. The recommended minimum length of the tested sequence is 100 b.

3.1.14 Random Excursions Test

The Random excursions test determines the number of cycles, which have the defined number of visits to a particular state of a cumulative sum random walk. Also in this case, values 0 of the tested sequence are replaced by -1. The cycle of the random walk is a series of steps with unit length beginning and ending at zero. This test evaluates whether the number of visits to a particular state within a cycle is approximately equal to the expected number for the random sequence. The reference distribution is the χ^2 distribution. The minimum length of the tested sequence should be equal to or greater than 10^6 b.

3.1.15 Random Excursions Variant Test

The Random excursions variant test counts the total number of visits to a particular state in a cumulative sum random walk. The tested sequence is modified as in the previous test. The total number is compared with the expected number of visits to various states for the random sequence. Also in this case, the minimum length of the tested sequence should be equal to or greater than 10^6 b. The reference distribution for the test statistic is the half-normal distribution.

3.2 FIPS Test Suite

The statistical test suite – the FIPS test suite – defined in [63] consist of 4 statistical tests. As in the NIST test suites, each test focuses on some property, which is significant for the random sequence. The basic condition is that all tests are intended for sequences containing exactly 20 000 b. All tests have been implemented in MATLAB according to their definitions.

3.2.1 Monobit Test

As well as the test with the same name from the NIST test suite, the Monobit test determines if the ratio between logic ones and zeros in the tested sequence is approximately the same. So in other words, if the number of ones in the tested sequence is in the range of 9 725 to 10 275, then the Monobit test from the FIPS test suite passes.

3.2.2 Poker Test

The Poker test reveals too high occurrences of 4-bit patterns in the tested sequence. Immediately after the start of the test, the tested sequence is divided into 5 000 sub-strings with the length 4 b. Each sub-string has one from 16 possible values. Occurrences of each value are counted across all sub-strings. The number of occurrences $N_{\mathrm{o,FIPS}}$ is compared with expected results of the random sequence. The evaluation is done using the formula

$$X_{\mathrm{FIPS}} = \frac{16}{5000}\left(\sum_{q=0}^{15} N_{\mathrm{o,FIPS},q}^{2}\right) - 5000 \tag{3.6}$$

while the Poker test passes when X_{FIPS} is in the range of 2.16 to 46.17.

3.2.3 Runs Test

The run is defined in section 3.1.3 where the Runs test of the NIST test suite is briefly described. The Runs test defined in the FIPS test suite works similarly. The number of occurrences of runs with different lengths is counted across the tested sequence. If the numbers of occurrences are within ranges defined in table 3.1, the Runs test passes.

Length of run	Required range
1	2 315 – 2 685
2	1 114 – 1 386
3	527 – 723
4	240 – 384
5	103 – 209
6 or more	103 – 209

Tab. 3.1: The required ranges for runs with different lengths of the Runs test

3.2.4 Long Runs Test

The Long runs test detects very long runs in the tested sequence. The long run can be defined as the run consisting of consecutive ones or zeros with the length of 26 or more. If the tested sequence does not contain any long runs, the long runs test passes.

3.3 Shannon Entropy

The entropy is often used for quality evaluation of random number sequences because it expresses a measure of unpredictability of possible events. This fundamental function was described by Shannon in [119] and is defined as follows. Individual events are marked as $x_1, x_2, \ldots, N_\mathrm{e}$. The probability that some event x_q occurs is always greater than or equal to zero where $q = 1, 2, \ldots, N_\mathrm{e}$. Sum of all probabilities is

$$\sum_{q=1}^{N_\mathrm{e}} \Pr\left(x_q\right) = 1. \tag{3.7}$$

Then the entropy is expressed as

$$H = -\sum_{q=1}^{N_\mathrm{e}} \Pr\left(x_q\right) \log_\upsilon \Pr\left(x_q\right) \tag{3.8}$$

where υ is the base of the logarithm while $\upsilon > 0$ and $\upsilon \neq 1$. If the probability of any event is zero, then

$$\lim_{\Pr(x_q)\to 0+} \Pr\left(x_q\right) \log_\upsilon \Pr\left(x_q\right) = 0. \tag{3.9}$$

Functional values of the entropy are in the range of 0 to 1. If the value is 0, it is possible to exactly determine the event. However, if the value of the entropy is 1, the event cannot be predicted. So the entropy of random number sequences produced by TRNGs should be very close to 1.

Chapter 4

TRNG with Time Multiplexed Sources of Randomness

TRNGs provide unpredictable random number sequences but at a cost of lower data rate than the data rate of PRNGs. Rising level of system security increases demands on the amount of high-quality random numbers generated in the shortest possible time. Amount of generated random data is highly dependent on the used manufacturing process of TRNGs. For increasing generated random data at a given time, modern technologies such as superconductive circuits [19] or semiconductor lasers [18] can be used. However, TRNGs implemented in complex SoCs are proposed in technologies, which are suitable for these systems. It allows true random data generation directly in modern hand-held devices. Contemporary SoCs are usually designed in standard submicron CMOS technologies, in which the random data rate is limited. Therefore for these systems, the random data rate is increased by a transition from serial approach to the parallel use of more independent sources of randomness.

A TRNG working on the basis of electrical and thermal noises present in CMOS structures has been proposed and manufactured. Concretely for generation of random bits, metastability-based sources of randomness are used. Randomness is extracted from thermal noise and flicker noise, which are generated in metal-oxide-semiconductor field effect transistor (MOSFET) channels. For a random data rate increase, the parallelism of more sources of randomness is implemented. The designed TRNG can generate typically tens of random megabits per second, which is several times more than a conventional TRNG. Parts of this chapter has been published by the author of this thesis in [4].

4.1 Principle of Time Multiplexed TRNG

While increasing amount of random data at any defined time, technological limits of random data generation can be reached. Then while maintaining the same technology, the same way of random number generation and the same operating conditions of a TRNG – a further increase in the amount of random data in the defined time is impossible. Therefore it is necessary to find new options in TRNG architecture that allows increasing rate of random data whereas strict criteria of random sequences for cryptographic devices will be satisfied [64].

In this proposal, the parallelism of several sources of randomness is introduced. Thus a composition of generated random data in parallel into a serial random data stream is defined on the hardware level. For the above composition, the principle of pipelining is used. Thus individual raw random data streams are generated synchronously with a phase shift and with a proportionally lower clock frequency rather than a clock frequency of an output random data stream. The output random data are created by sequential mixing of raw random data streams according to

$$\{r\,[l]\}_{l=0}^{\infty} = \left\{ s_1\,[i],\ s_2\left[i + \frac{1}{n}\right], s_3\left[i + \frac{2}{n}\right],\ \ldots,\ s_n\left[i + \frac{n-1}{n}\right] \right\}_{i=0}^{\infty} \tag{4.1}$$

where n is number of independent serial random data streams, $s_1,\ s_2,\ \ldots,\ s_n$ are random data streams, r is the output random data, i is a normalized period of the random data stream while $i = t/T_\mathrm{c}$, t is time, and T_c is a period of a clock signal, l is a normalized period of the output data stream while $l = ni$. Values of l and i are non-negative integers because random data is generated after the system starts at $t = 0$. The formula (4.1) may only be used under condition when the raw random data streams are generated in digitized sources of randomness independently. A bit rate of the raw random data stream is given by a clock frequency where $f_\mathrm{c} = 1/T_\mathrm{c}$. Naturally from the above-defined output random data, their bit rate can be derived according to

$$l = ni = n\frac{t}{T_\mathrm{c}} = nf_\mathrm{c}\,t \tag{4.2}$$

where f_c is a frequency of the clock signal. Thus the result of mixing is that the data rate of the output random data is n times higher than the bit rate of one digitized source of randomness. The used principle is depicted in figure 4.1 where $clk\,[i]$ denotes a clock signal with the period T_c.

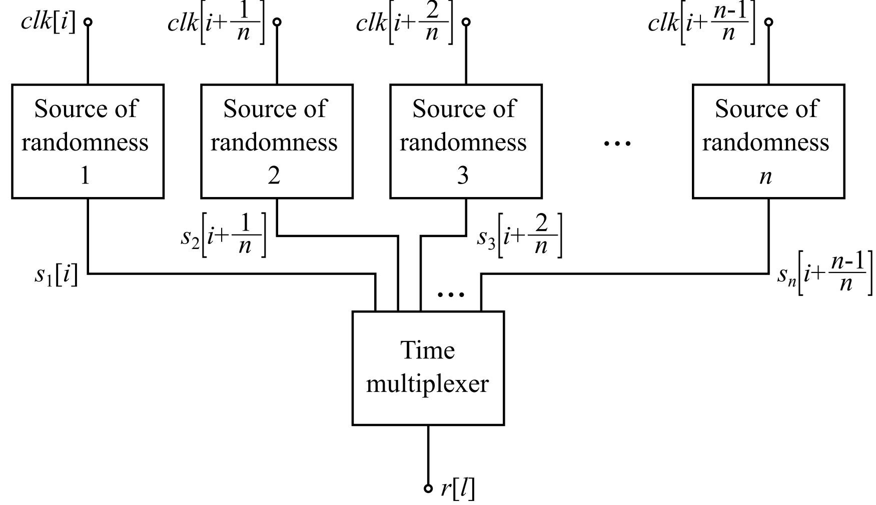


Fig. 4.1: Principle illustration of the proposed TRNG [4]

During mixing the randomness of generated data must be preserved. TRNGs do not generate a continuous analog signal but usually produce a digital signal with two voltage levels, which are represented by logic one and logic zero. Random data sequences composed of logic ones and zeros have significant properties, which must not be changed by data mixing. The most important one is the mean value of a bit sequence, which has to be close to $1/2$ in case of a random bit sequence. Thus the mean value of output random data r has to be close $1/2$ after sequential mixing of single data streams $s_1,\ s_2,\ \ldots,\ s_n$, which are generated by fully functional and independent sources of randomness. One data stream can be described as

$$s_q = \left(s_1^q,\ s_2^q,\ s_3^q,\ \ldots,\ s_m^q\right) \tag{4.3}$$

where m is length of the data stream, s_k^q is a random value of the data stream, $k = 1, 2, 3, \ldots, m$, and q is an identifier of the source of randomness, $q = 1, 2, 3, \ldots, n$. In general, a random signal u with the mean value $\bar{u}$ is composed of consecutive data streams $s_1,\ s_2,\ \ldots,\ s_n$ so that

$$u = (s_1,\ s_2,\ s_3,\ \ldots,\ s_n) = \left(s_1^1,\ s_2^1,\ s_3^1,\ \ldots,\ s_m^n\right). \tag{4.4}$$

A random signal u_σ is formed by any permutation without repetition of values s_k^q of the random signal u and has the mean value $\bar{u}_\sigma$. At this point the distributive property of summation is used and then

$$\bar{u} = \frac{1}{n}\sum_{q=1}^{n}\left(\frac{1}{m}\sum_{k=1}^{m} s_k^q\right) = \frac{1}{nm}\sum_{q=1}^{n}\sum_{k=1}^{m} s_k^q. \tag{4.5}$$

Because the random signal u_σ is composed of values s_k^q and summation has the commutative property it can be shown that

$$\bar{u} = \frac{1}{nm} \sum_{q=1}^{n} \sum_{k=1}^{m} s_k^q = \bar{u}_\sigma. \tag{4.6}$$

Output random data r is a permutation of the random signal u, which means that the mean value of output random data is not changed during mixing.

4.2 Circuit Implementation

The introduced TRNG with time multiplexed sources of randomness has been designed and fabricated in the 130 nm CMOS technology from STMicroelectronics known as HCMOS9GP in a standard variant with power supply voltage 1.2 V. This design is based on four independent sources of randomness. Therefore output random data r are composed of four independent raw random data streams while equation (4.1) changes into the form

$$\{r\,[l]\}_{l=0}^{\infty} = \left\{ s_1\,[i],\ s_2 \left[i + \frac{1}{4} \right], s_3 \left[i + \frac{2}{4} \right],\ s_4 \left[i + \frac{3}{4} \right] \right\}_{i=0}^{\infty} \tag{4.7}$$

because n is a number of independent random data streams and is equal to 4 in this case. Thus the normalized period l of output random data stream r is four times higher than the normalized period i of each raw random data stream s_n. And according to equation (4.2) the bit rate of output random data r is four times higher than the bit rate of used sources of randomness. If a source of randomness works with a maximal bit rate the bit rate of output random numbers can be increased by implementation of above-described principle.

A block diagram of the proposed TRNG is shown in figure 4.2. In presented design, four independent sources of randomness based on metastable states of electronic circuits are integrated and they generate digitized random signals V_{sq} where q is the identifier of the source of randomness and $q = 1, 2, 3, 4$. The source of randomness is composed of the noise source and the digitizer. Proposed noise sources use fine reference currents $I_{\mathrm{REF}q}$, which are generated in a reference current generator. All digitized random signals V_{sq} are mixed in a time multiplexer, which is together with all sources of randomness controlled by a clock signal generator where control signals are derived from the reference signal of external oscillator V_{OSC}. Output buffer processes output of time multiplexer and is able to drive external digital circuits or inputs of measuring instruments.

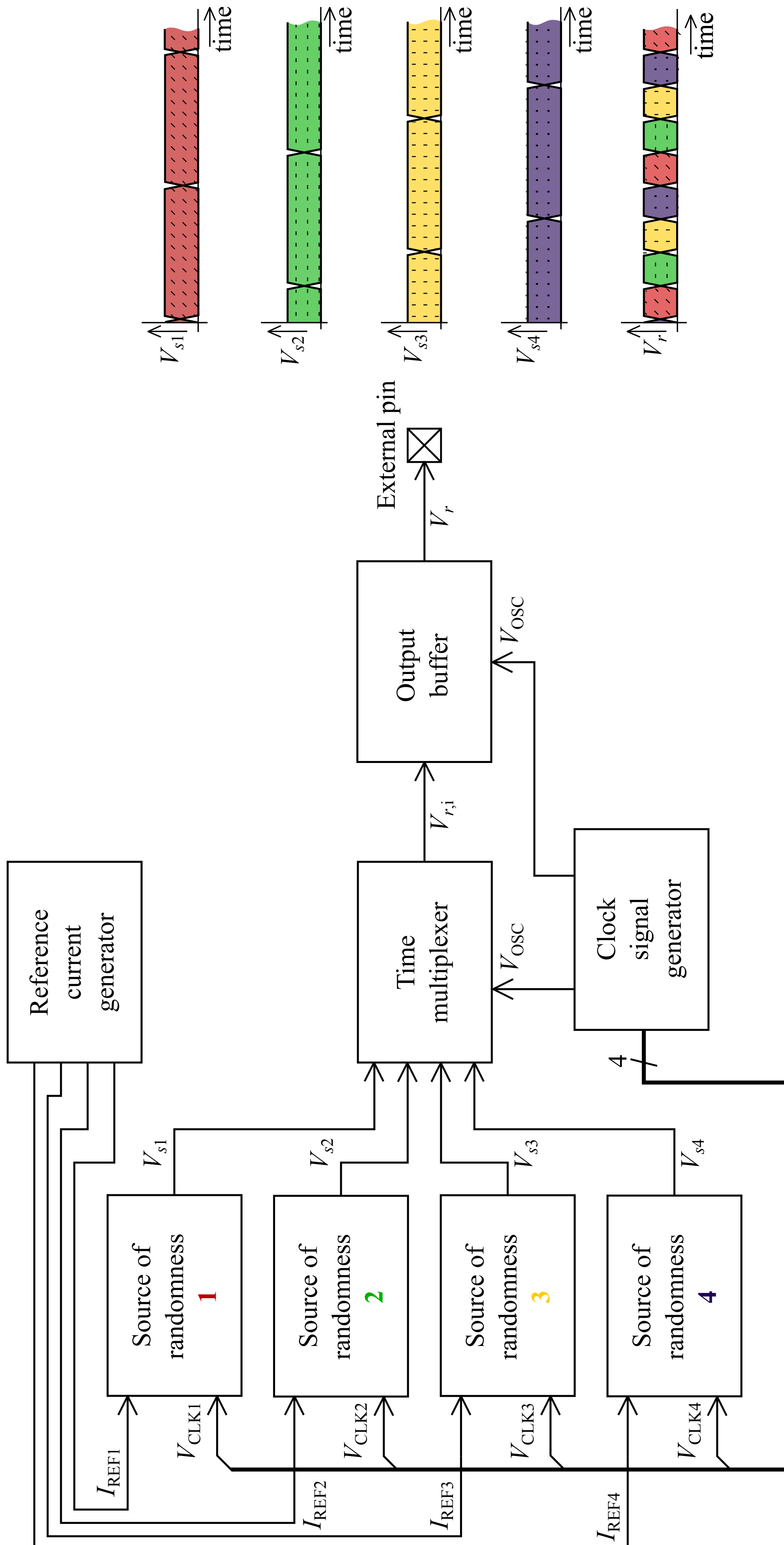


Fig. 4.2: The block diagram of the designed TRNG with time multiplexed sources of randomness [4]

4.2.1 Noise source

A noise source is a part of the presented circuit, which is able to extract randomness from a physical phenomenon appearing in silicon semiconductors. In this case the proposed noise source structure shown in figure 4.3 is based on a fast comparator, whose inputs are connected to the same voltage V_{IN} created on the transistor $\mathrm{M_{N3}}$. Thus a noise present in circuit decides on an output value. The noise source has to work periodically. Therefore the circuit is extended by a reset transistor $\mathrm{M_{N6}}$, which resets the proposed noise source and allows to generate a new random value each period. This is shown in figure 4.4 where a random value of the output signal $V_{\mathrm{O,M1}}$ is generated when the clock signal V_{CLK1} is in logic zero. The noise source is reset by the transistor $\mathrm{M_{N6}}$ when the clock signal V_{CLK1} is in logic one. Decision phase arises at the beginning of generating each random value when the circuit is in a metastable state and metastable voltage V_{meta} is equal to 274 mV. The noise present in the circuit causes a transition to a stable state. The final random value of the output signal $V_{\mathrm{O,M1}}$ is given by the noise in the circuit during decision phase.

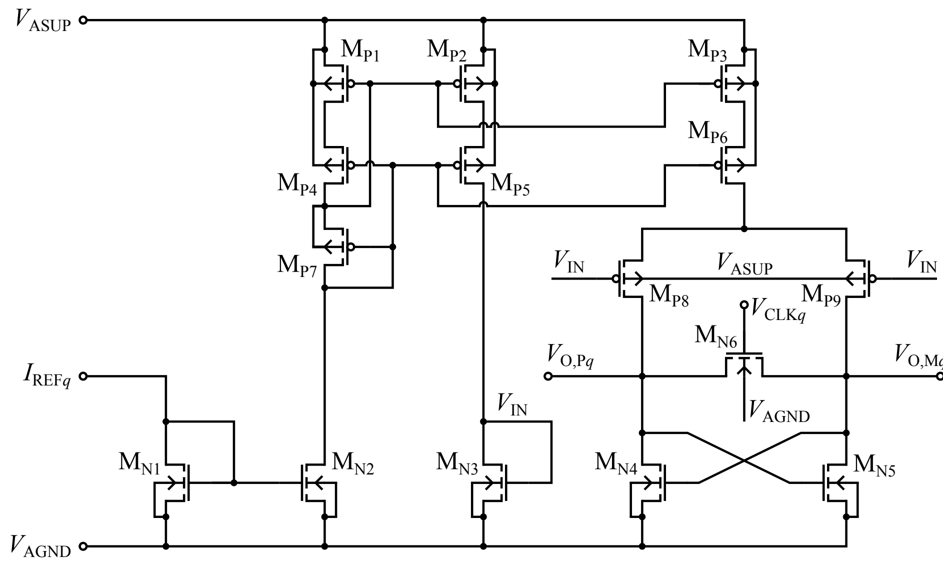


Fig. 4.3: The schematic diagram of the metastability-based noise source [4]

Systematic errors can cause malfunction of the manufactured device. The noise source structure is proposed with maximum regard to symmetry, whose breach would adversely affect output random data, which would result in a deviation of the random data mean value. Therefore the important assumption is the conformity of dimensions of transistors $\mathrm{M_{P8}}$ and $\mathrm{M_{P9}}$ and also transistors $\mathrm{M_{N4}}$ and $\mathrm{M_{N5}}$. Not only the same dimensions of transis-

tor pairs but also totally symmetric topology design including metal interconnections and well-matched elements are prerequisites for the properly functioning noise source. Hence the proposed noise source is composed of two equal branches, which differ only in small details namely vias among metal layers. To reduce appreciable mismatches among parameters of paired transistors, a common-centroid configuration is used so that first-order gradients are canceled [107], [121].

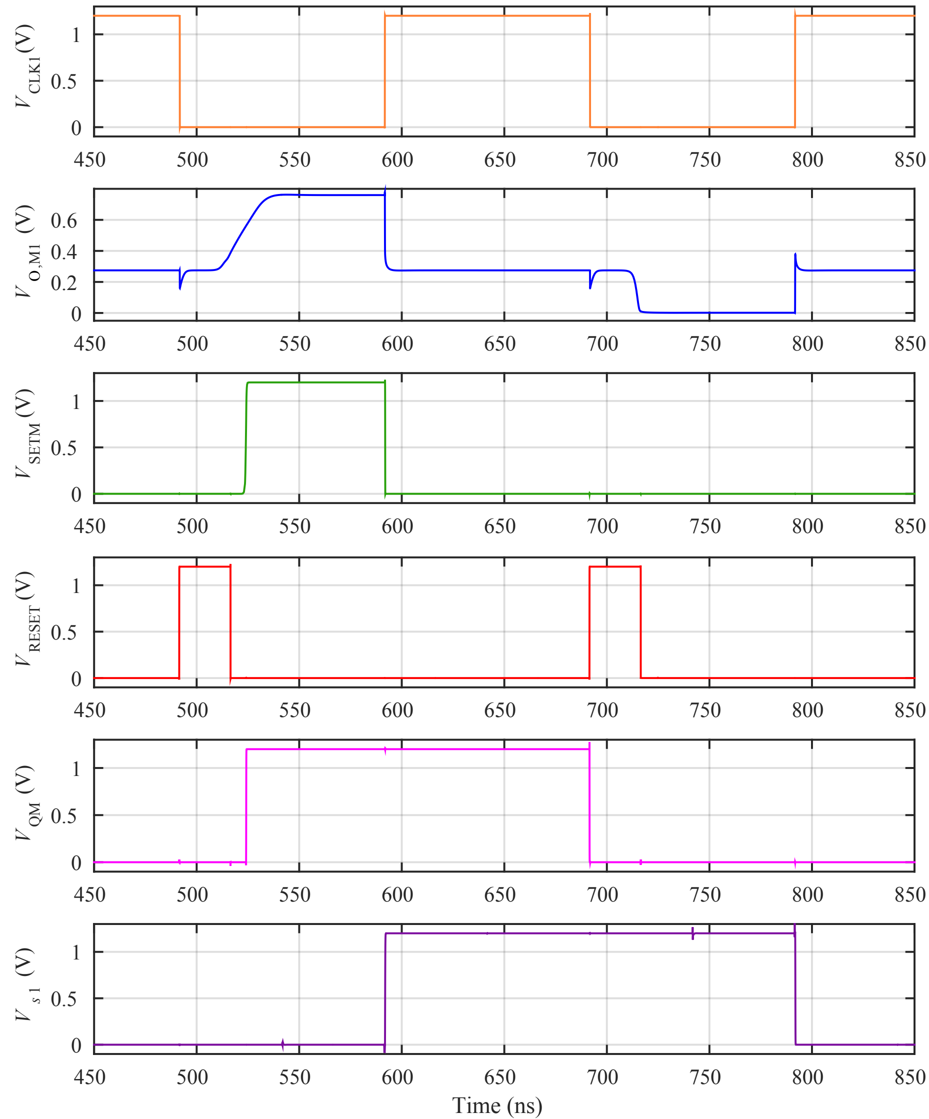


Fig. 4.4: Waveforms of signals inside the designed TRNG simulated by the Mentor Eldo simulator at the transistor level [4]

The noise source is designed to minimize the offset voltage, which is able to cause an undesirable distortion of output random data, such as a deviation of the mean value of

output random data or even circuit locking in one logical value. Contribution to the offset voltage created by any difference between output voltages $V_{\mathrm{O,P}q}$ and $V_{\mathrm{O,M}q}$ is eliminated by proper circuit design and layout. Thus the offset voltage can be caused by a mismatch of MOSFET parameters such as a threshold voltage V_{TH} or a β parameter which is defined as

$$\beta = \mu_{\mathrm{m}} C_{\mathrm{ox}} \frac{W}{L} \tag{4.8}$$

where μ_{m} is mobility of charge carriers in MOSFET, C_{ox} is the gate oxide capacitance per unit area, W is the gate width and L the gate length.

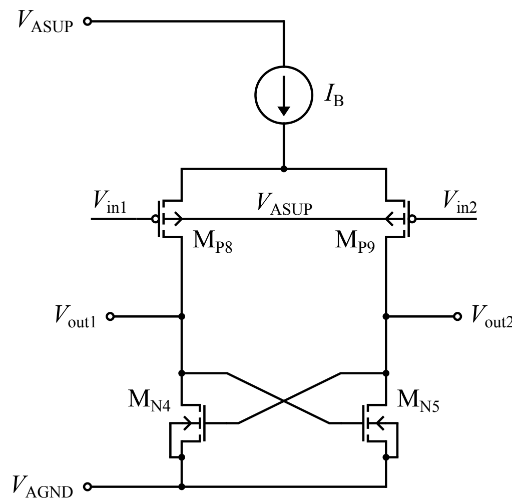


Fig. 4.5: The schematic diagram of the proposed metastable element used for offset analysis

The proposed noise source contains a metastable element, which is a core of the circuit and extracts randomness from noise present. The schematic diagram of the metastable element is shown in figure 4.5. To minimization the offset voltage, the metastable element is analyzed. First, it is necessary to calculate gain. Thus the metastable element is replaced by its small signal model depicted in figure 4.6. Difference between v_{out1} and v_{out2} is caused by circular current and is not affected by $r_{\mathrm{ds,B}}$. For this reason, the small signal model can be simplified as shown in figure 4.7.

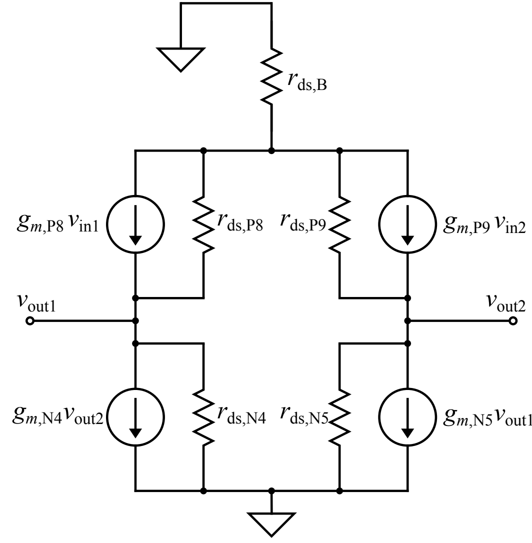


Fig. 4.6: The small signal model of the metastable element

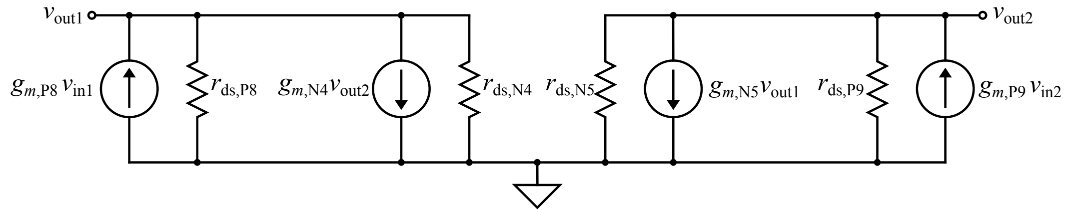


Fig. 4.7: The simplified small signal model of the metastable element used for gain calculation

For gain calculation, the above mentioned simplified small signal model depicted in figure 4.7 can be described by equations

$$\frac{v_{\text{out1}}}{r_{\text{ds,P8}}} + \frac{v_{\text{out1}}}{r_{\text{ds,N4}}} + g_{m,\text{N4}} v_{\text{out2}} - g_{m,\text{P8}} v_{\text{in1}} = 0 \tag{4.9}$$

and

$$\frac{v_{\text{out2}}}{r_{\text{ds,P9}}} + \frac{v_{\text{out2}}}{r_{\text{ds,N5}}} + g_{m,\text{N5}} v_{\text{out1}} - g_{m,\text{P9}} v_{\text{in2}} = 0 \tag{4.10}$$

where g_m is transconductance of the relevant MOSFETs, r_{ds} is a output resistance of the MOSFET small signal model, v_{in1} and v_{in2} are input voltages, and v_{out1} and v_{out2} are output voltages. Then both equations (4.9) and (4.10) are expressed as

$$\frac{v_{\text{out1}}}{r_{\text{ds,P8}}} + \frac{v_{\text{out1}}}{r_{\text{ds,N4}}} + g_{m,\text{N4}} v_{\text{out2}} - g_{m,\text{P8}} v_{\text{in1}} = \frac{v_{\text{out2}}}{r_{\text{ds,P9}}} + \frac{v_{\text{out2}}}{r_{\text{ds,N5}}} + g_{m,\text{N5}} v_{\text{out1}} - g_{m,\text{P9}} v_{\text{in2}}. \tag{4.11}$$

This equation can be modified to the following form

$$v_{\text{out1}}\left(\frac{1}{r_{\text{ds,P8}}}+\frac{1}{r_{\text{ds,N4}}}-g_{m,\text{N5}}\right)-v_{\text{out2}}\left(\frac{1}{r_{\text{ds,P9}}}+\frac{1}{r_{\text{ds,N5}}}-g_{m,\text{N4}}\right)=$$
$$=g_{m,\text{P8}}v_{\text{in1}}-g_{m,\text{P9}}v_{\text{in2}}. \quad (4.12)$$

In case of good design and high-quality layout of the circuit, conditions mentioned below are fulfilled

$$g_{m,\text{P8}}=g_{m,\text{P9}};g_{m,\text{N4}}=g_{m,\text{N5}};r_{\text{ds,P8}}=r_{\text{ds,P9}};r_{\text{ds,N4}}=r_{\text{ds,N5}}. \quad (4.13)$$

Under these assumptions, the equation (4.12) can be simplified

$$(v_{\text{out1}}-v_{\text{out2}})\left(\frac{1}{r_{\text{ds,P8}}}+\frac{1}{r_{\text{ds,N4}}}-g_{m,\text{N4}}\right)=g_{m,\text{P8}}\left(v_{\text{in1}}-v_{\text{in2}}\right). \quad (4.14)$$

Thus the gain is expressed as

$$A_v=\frac{v_{\text{out1}}-v_{\text{out2}}}{v_{\text{in1}}-v_{\text{in2}}}=\frac{g_{m,\text{P8}}}{\frac{1}{r_{\text{ds,P8}}}+\frac{1}{r_{\text{ds,N4}}}-g_{m,\text{N4}}}=\frac{g_{m,\text{P8}}}{g_{\text{ds,P8}}+g_{\text{ds,N4}}-g_{m,\text{N4}}} \quad (4.15)$$

where g_{ds} is a output conductance of the MOSFET small signal model. In case $g_{\text{ds,N4}} << g_{m,\text{N4}}$ and $g_{\text{ds,P8}} << g_{m,\text{P8}}$, the gain can be expressed as

$$A_v\approx-\frac{g_{m,\text{P8}}}{g_{m,\text{N4}}}. \quad (4.16)$$

The circular current i_{c} creates difference between v_{out1} and v_{out2} and is given by

$$i_{\text{c}}=g_{m,\text{P8}}\left(v_{\text{in1}}-v_{\text{in2}}\right). \quad (4.17)$$

For calculation of the difference between v_{out1} and v_{out2}, the equation (4.14) is used. Then

$$v_{\text{out1}}-v_{\text{out2}}=\frac{g_{m,\text{P8}}\left(v_{\text{in1}}-v_{\text{in2}}\right)}{\left(\frac{1}{r_{\text{ds,P8}}}+\frac{1}{r_{\text{ds,N4}}}-g_{m,\text{N4}}\right)}. \quad (4.18)$$

In the next step, the equation (4.17) describing the circular current is inserted into the equation (4.18) and

$$v_{\text{out1}}-v_{\text{out2}}=\frac{i_{\text{c}}}{\left(\frac{1}{r_{\text{ds,P8}}}+\frac{1}{r_{\text{ds,N4}}}-g_{m,\text{N4}}\right)}=\frac{i_{\text{c}}}{\left(g_{\text{ds,P8}}+g_{\text{ds,N4}}-g_{m,\text{N4}}\right)}. \quad (4.19)$$

If $g_{\text{ds,N4}} << g_{m,\text{N4}}$ and $g_{\text{ds,P8}} << g_{m,\text{P8}}$, the difference between v_{out1} and v_{out2} can be expressed as

$$v_{\text{out1}}-v_{\text{out2}}\approx-\frac{i_{\text{c}}}{g_{m,\text{N4}}}. \quad (4.20)$$

The offset voltage can be caused by mismatch of the threshold voltage V_{th} of P-channel MOSFETs $\mathrm{M_{P8}}$ and $\mathrm{M_{P9}}$. Thus the threshold voltage can be described by

$$V_{\mathrm{th}}^{*} = V_{\mathrm{th}} + \Delta V_{\mathrm{th}} \tag{4.21}$$

where ΔV_{th} is change of the threshold voltage. If considered MOSFETs operate in saturation mode, their drain current is given by

$$I_{\mathrm{D}} = \frac{1}{2}\mu_{\mathrm{m}} C_{\mathrm{ox}} \frac{W}{L} (V_{\mathrm{GS}} - V_{\mathrm{th}})^2 \tag{4.22}$$

where V_{GS} is voltage between the gate terminal and the source terminal. Then the change of the MOSFET threshold voltage affects its drain current, which can be expressed by

$$I_{\mathrm{D,P8,9}}^{*} = \frac{1}{2}\mu_{\mathrm{m,P}} C_{\mathrm{ox}} \frac{W_{\mathrm{P8,9}}}{L_{\mathrm{P8,9}}} (V_{\mathrm{GS,P8,9}} - V_{\mathrm{th,P8,9}}^{*})^2. \tag{4.23}$$

By substitution of the equation (4.21) into the equation (4.23), the drain current is written as

$$I_{\mathrm{D,P8,9}}^{*} = \frac{1}{2}\mu_{\mathrm{m,P}} C_{\mathrm{ox}} \frac{W_{\mathrm{P8,9}}}{L_{\mathrm{P8,9}}} (V_{\mathrm{GS,P8,9}} - V_{\mathrm{th,P8,9}} - \Delta V_{\mathrm{th,P8,9}})^2 \tag{4.24}$$

and then this equation is appropriately modified to a form

$$I_{\mathrm{D,P8,9}}^{*} = \frac{1}{2}\mu_{\mathrm{m,P}} C_{\mathrm{ox}} \frac{W_{\mathrm{P8,9}}}{L_{\mathrm{P8,9}}} \left((V_{\mathrm{GS,P8,9}} - V_{\mathrm{th,P8,9}}) - \frac{\Delta V_{\mathrm{th,P8,9}} (V_{\mathrm{GS,P8,9}} - V_{\mathrm{th,P8,9}})}{V_{\mathrm{GS,P8,9}} - V_{\mathrm{th,P8,9}}} \right)^2 \tag{4.25}$$

to be possible to express influence on the fundamental equation of the MOSFET drain current (4.22) as

$$I_{\mathrm{D,P8,9}}^{*} = \frac{1}{2}\mu_{\mathrm{m,P}} C_{\mathrm{ox}} \frac{W_{\mathrm{P8,9}}}{L_{\mathrm{P8,9}}} (V_{\mathrm{GS,P8,9}} - V_{\mathrm{th,P8,9}})^2 \left(1 - \frac{\Delta V_{\mathrm{th,P8,9}}}{V_{\mathrm{GS,P8,9}} - V_{\mathrm{th,P8,9}}} \right)^2. \tag{4.26}$$

As can be seen from the equation (4.26), the current $I_{\mathrm{D,P8,9}}^{*}$ is a function of $\Delta V_{\mathrm{th,P8,9}}$. So this equation can be linearized for $\Delta V_{\mathrm{th,P8,9}} << V_{\mathrm{GS,P8,9}} - V_{\mathrm{th,P8,9}}$. The first two members of the Taylor series in $\Delta V_{\mathrm{th,P8,9}} = 0$ sufficiently approximate $I_{\mathrm{D,P8,9}}^{*}$. To simplify the next steps, coefficients in the equation (4.26) are replaced by the $\beta_{\mathrm{P8,9}}$ parameter according to the equation (4.8). Thus the first member of the Taylor series is given by

$$I_{\mathrm{D,P8,9}}^{*}(\Delta V_{\mathrm{th,P8,9}} = 0) = \frac{1}{2}\beta_{\mathrm{P8,9}} (V_{\mathrm{GS,P8,9}} - V_{\mathrm{th,P8,9}})^2. \tag{4.27}$$

The second member of the Taylor series is determined using derivative of $I_{\mathrm{D,P8,9}}^{*}$, which is expressed as

$$\frac{\mathrm{d}I_{\mathrm{D,P8,9}}^{*}}{\mathrm{d}\Delta V_{\mathrm{th,P8,9}}} = \beta_{\mathrm{P8,9}} (V_{\mathrm{GS,P8,9}} - V_{\mathrm{th,P8,9}})^2 \left(1 - \frac{\Delta V_{\mathrm{th,P8,9}}}{V_{\mathrm{GS,P8,9}} - V_{\mathrm{th,P8,9}}} \right) \frac{-1}{V_{\mathrm{GS,P8,9}} - V_{\mathrm{th,P8,9}}} \tag{4.28}$$

and after simplification, this equation is written as

$$\frac{\mathrm{d}I^*_{\mathrm{D,P8,9}}}{\mathrm{d}\Delta V_{\mathrm{th,P8,9}}} = \beta_{\mathrm{P8,9}}\Delta V_{\mathrm{th,P8,9}} - \beta_{\mathrm{P8,9}}(V_{\mathrm{GS,P8,9}} - V_{\mathrm{th,P8,9}}). \tag{4.29}$$

So the second member of the Taylor series is calculated using the equation (4.29). Thus

$$\left(\frac{\mathrm{d}I^*_{\mathrm{D,P8,9}}}{\mathrm{d}\Delta V_{\mathrm{th,P8,9}}}(\Delta V_{\mathrm{th,P8,9}} = 0)\right)\Delta V_{\mathrm{th,P8,9}} = -\beta_{\mathrm{P8,9}}(V_{\mathrm{GS,P8,9}} - V_{\mathrm{th,P8,9}})\Delta V_{\mathrm{th,P8,9}}. \tag{4.30}$$

Then it is suitable to modify this equation into following form

$$\left(\frac{\mathrm{d}I^*_{\mathrm{D,P8,9}}}{\mathrm{d}\Delta V_{\mathrm{th,P8,9}}}(\Delta V_{\mathrm{th,P8,9}} = 0)\right)\Delta V_{\mathrm{th,P8,9}} = \\ = -\frac{1}{2}\beta_{\mathrm{P8,9}}(V_{\mathrm{GS,P8,9}} - V_{\mathrm{th,P8,9}})^2\frac{2\Delta V_{\mathrm{th,P8,9}}}{V_{\mathrm{GS,P8,9}} - V_{\mathrm{th,P8,9}}}. \tag{4.31}$$

Based on the calculated members of the Taylor series (4.27) and (4.31), the linearized form of $I^*_{\mathrm{D,P8,9}}$ can be expressed as

$$I^*_{\mathrm{D,P8,9}} \approx \frac{1}{2}\beta_{\mathrm{P8,9}}(V_{\mathrm{GS,P8,9}} - V_{\mathrm{th,P8,9}})^2 - \frac{1}{2}\beta_{\mathrm{P8,9}}(V_{\mathrm{GS,P8,9}} - V_{\mathrm{th,P8,9}})^2\frac{2\Delta V_{\mathrm{th,P8,9}}}{V_{\mathrm{GS,P8,9}} - V_{\mathrm{th,P8,9}}} \tag{4.32}$$

while the drain current can be written as

$$I^*_{\mathrm{D,P8,9}} = I_{\mathrm{D,P8,9}} - \Delta I_{\mathrm{D,P8,9}} \tag{4.33}$$

where $\Delta I_{\mathrm{D,P8,9}}$ is a change of the drain current caused by $\Delta V_{\mathrm{th,P8,9}}$. From the equations (4.32) and (4.33), $\Delta I_{\mathrm{D,P8,9}}$ is expressed in following form

$$\Delta I_{\mathrm{D,P8,9}} = I_{\mathrm{D,P8,9}}\frac{2\Delta V_{\mathrm{th,P8,9}}}{V_{\mathrm{GS,P8,9}} - V_{\mathrm{th,P8,9}}}. \tag{4.34}$$

The offset voltage V_{OS1} caused by $\Delta V_{\mathrm{th,P8,9}}$ can be calculated by

$$V_{\mathrm{OS1}} = \frac{v_{\mathrm{out1}} - v_{\mathrm{out2}}}{A_v} \tag{4.35}$$

under the condition of the equality of input voltages. The difference between output voltages is caused by the circular current, which is $\Delta I_{\mathrm{D,P8,9}}$ in this case. By inserting the above described equations (4.19) and (4.15) into the equation (4.35) and subsequent simplifications, the offset voltage is given by

$$V_{\mathrm{OS1}} = \frac{\Delta I_{\mathrm{D,P8,9}}}{g_{m,P8,9}}. \tag{4.36}$$

At this moment, $\Delta I_{\mathrm{D,P8,9}}$ can be substituted by the equation (4.34) and $g_{m,P8,9}$ by commonly used equation

$$g_{m,\mathrm{P8,9}} = \frac{2I_{\mathrm{D,P8,9}}}{V_{\mathrm{GS,P8,9}} - V_{\mathrm{th,P8,9}}}. \tag{4.37}$$

Thus, after further simplification, the the offset voltage corresponds to

$$V_{\mathrm{OS1}} = \Delta V_{\mathrm{th,P8,9}}. \tag{4.38}$$

Also mismatch of the threshold voltages of N-channel MOSFETs M_{N4} and M_{N5} affects the offset voltage. For derivation of influence of $\Delta V_{\mathrm{th,N4,5}}$, the similar approach as in the previous paragraph is used. Therefore the drain current is expressed as

$$I^*_{\mathrm{D,N4,5}} = \frac{1}{2}\mu_{\mathrm{m,N}} C_{\mathrm{ox}} \frac{W_{\mathrm{N4,5}}}{L_{\mathrm{N4,5}}} (V_{\mathrm{GS,N4,5}} - V^*_{\mathrm{th,N4,5}})^2. \tag{4.39}$$

By inserting the equation (4.21) into the equation (4.39) and appropriate adjustment, the required form is prepared

$$I^*_{\mathrm{D,N4,5}} = \frac{1}{2}\mu_{\mathrm{m,N}} C_{\mathrm{ox}} \frac{W_{\mathrm{N4,5}}}{L_{\mathrm{N4,5}}} (V_{\mathrm{GS,N4,5}} - V_{\mathrm{th,N4,5}})^2 \left(1 - \frac{\Delta V_{\mathrm{th,N4,5}}}{V_{\mathrm{GS,N4,5}} - V_{\mathrm{th,N4,5}}}\right)^2. \tag{4.40}$$

Assuming $\Delta V_{\mathrm{th,N4,5}} << V_{\mathrm{GS,N4,5}} - V_{\mathrm{th,N4,5}}$, the equation (4.40) is linearized by the first two members of the Taylor series in $\Delta V_{\mathrm{th,N4,5}} = 0$ as in the previous case. So the resulting equation sufficiently approximating $I^*_{\mathrm{D,N4,5}}$ is written as

$$\begin{aligned} I^*_{\mathrm{D,N4,5}} \approx \frac{1}{2}\beta_{\mathrm{PN4,5}}(V_{\mathrm{GS,N4,5}} - V_{\mathrm{th,N4,5}})^2 - \\ - \frac{1}{2}\beta_{\mathrm{N4,5}}(V_{\mathrm{GS,N4,5}} - V_{\mathrm{th,N4,5}})^2 \frac{2\Delta V_{\mathrm{th,N4,5}}}{V_{\mathrm{GS,N4,5}} - V_{\mathrm{th,N4,5}}} \end{aligned} \tag{4.41}$$

while the drain current is given by

$$I^*_{\mathrm{D,N4,5}} = I_{\mathrm{D,N4,5}} - \Delta I_{\mathrm{D,N4,5}} \tag{4.42}$$

where $\Delta V_{\mathrm{th,N4,5}}$ causes $\Delta I_{\mathrm{D,N4,5}}$, which can be expressed from the equations (4.41) and (4.42) as

$$\Delta I_{\mathrm{D,N4,5}} = \frac{1}{2}\mu_{\mathrm{m,N}} C_{\mathrm{ox}} \frac{W_{\mathrm{N4,5}}}{L_{\mathrm{N4,5}}} (V_{\mathrm{GS,N4,5}} - V_{\mathrm{th,N4,5}})^2 \frac{2\Delta V_{\mathrm{th,N4,5}}}{V_{\mathrm{GS,N4,5}} - V_{\mathrm{th,N4,5}}}. \tag{4.43}$$

This change of drain current causes difference between output voltages according to the equation (4.19). Then for offset voltage calculation, the equation (4.35) is used. So the offset voltage V_{OS2} caused by $\Delta V_{\mathrm{th,N4,5}}$ is described by

$$V_{\mathrm{OS2}} = \frac{\Delta I_{\mathrm{D,N4,5}}}{g_{m,\mathrm{P8,9}}}. \tag{4.44}$$

By inserting the equation (4.43) into the equation (4.44) and subsequent simplification, the the offset voltage V_{OS2} can be written as

$$V_{\mathrm{OS2}} = \frac{\mu_{\mathrm{m,N}} C_{\mathrm{ox}} \frac{W_{\mathrm{N4,5}}}{L_{\mathrm{N4,5}}} (V_{\mathrm{GS,N4,5}} - V_{\mathrm{th,N4,5}}) \Delta V_{\mathrm{th,N4,5}}}{g_{m,\mathrm{P8,9}}}. \tag{4.45}$$

Transconductance of MOSFETs can be expressed by commonly used equation

$$g_m = \mu_{\mathrm{m}} C_{\mathrm{ox}} \frac{W}{L} (V_{\mathrm{GS}} - V_{\mathrm{th}}). \tag{4.46}$$

By inserting the equation (4.46) into the equation (4.45), the offset voltage is given by

$$V_{\mathrm{OS2}} = \frac{g_{m,\mathrm{N4,5}}}{g_{m,\mathrm{P8,9}}} \Delta V_{\mathrm{th,N4,5}}. \tag{4.47}$$

From design point of view, it is appropriate to adjust this equation (4.47) into the form containing geometrical dimensions of the MOSFETs, which can be modified during design of the TRNG. Therefore transconductance in the equation (4.47) are replaced by the equation (4.46). After simplification, the resulting form of the offset voltage caused by $\Delta V_{\mathrm{th,N4,5}}$ is

$$V_{\mathrm{OS2}} = \Delta V_{\mathrm{th,N4,5}} \sqrt{\frac{\mu_{\mathrm{m,N}} \frac{W_{\mathrm{N4,5}}}{L_{\mathrm{N4,5}}}}{\mu_{\mathrm{m,P}} \frac{W_{\mathrm{P8,9}}}{L_{\mathrm{P8,9}}}}}. \tag{4.48}$$

The offset voltage is also caused by mismatch of the β parameters of P-channel MOSFETs M_{P8} and M_{P9}. A change of the β parameter is defined by

$$\beta^* = \beta + \Delta\beta \tag{4.49}$$

and affects the drain current flowing through the MOSFET, which is thus expressed as

$$I^*_{\mathrm{D,P8,9}} = \frac{1}{2} (\beta_{\mathrm{P8,9}} + \Delta\beta_{\mathrm{P8,9}})(V_{\mathrm{GS,P8,9}} - V_{\mathrm{th,P8,9}})^2. \tag{4.50}$$

After appropriate adjustment, the drain current is given by

$$I^*_{\mathrm{D,P8,9}} = \frac{1}{2} \beta_{\mathrm{P8,9}} (V_{\mathrm{GS,P8,9}} - V_{\mathrm{th,P8,9}})^2 + \frac{1}{2} \beta_{\mathrm{P8,9}} (V_{\mathrm{GS,P8,9}} - V_{\mathrm{th,P8,9}})^2 \frac{\Delta\beta_{\mathrm{P8,9}}}{\beta_{\mathrm{P8,9}}} \tag{4.51}$$

and it can also be written as

$$I^*_{\mathrm{D,P8,9}} = I_{\mathrm{D,P8,9}} + \Delta I_{\mathrm{D,P8,9}}. \tag{4.52}$$

The change of the drain current is caused by $\Delta\beta_{\mathrm{P8,9}}$. Using the equations (4.51) and (4.52), this change is described by

$$\Delta I_{\mathrm{D,P8,9}} = I_{\mathrm{D,P8,9}} \frac{\Delta\beta_{\mathrm{P8,9}}}{\beta_{\mathrm{P8,9}}} \tag{4.53}$$

and it creates difference between output voltages. As in previous cases, the offset voltage can be determined by the equation (4.35) where the output voltage difference is replaced

by the equation (4.19) and the gain by the equation (4.15). After simplifications, it follows that

$$V_{\mathrm{OS3}} = \frac{\Delta I_{\mathrm{D,P8,9}}}{g_{m,\mathrm{P8,9}}}. \tag{4.54}$$

By inserting the equation (4.53) into the equation (4.54), the offset voltage caused by $\Delta\beta_{\mathrm{P8,9}}$ can be expressed as

$$V_{\mathrm{OS3}} = \frac{I_{\mathrm{D,P8,9}}}{g_{m,\mathrm{P8,9}}} \frac{\Delta\beta_{\mathrm{P8,9}}}{\beta_{\mathrm{P8,9}}}. \tag{4.55}$$

To express geometrical dimensions of MOSFETs, the well-known form of transconductance is used

$$g_m = \sqrt{2\mu_{\mathrm{m}} C_{\mathrm{ox}} \frac{W}{L} I_{\mathrm{D}}}. \tag{4.56}$$

So after replacement of $g_{m,P8,9}$ in the equation (4.55) by the equation (4.56) and subsequent simplifications, the resulting form of the offset voltage caused by $\Delta\beta_{\mathrm{P8,9}}$ is

$$V_{\mathrm{OS3}} = \sqrt{\frac{I_{\mathrm{D,P8,9}}}{2\mu_{\mathrm{m,P}} C_{\mathrm{ox}} \frac{W_{\mathrm{P8,9}}}{L_{\mathrm{P8,9}}}}} \frac{\Delta\beta_{\mathrm{P8,9}}}{\beta_{\mathrm{P8,9}}}. \tag{4.57}$$

Mismatch of the β parameters of N-channel MOSFETs M_{N4} and M_{N5} also contributes to the offset voltage of the metastable element. The offset voltage caused by $\Delta\beta_{\mathrm{N4,5}}$ is derived in the same way as the offset voltage caused by $\Delta\beta_{\mathrm{P8,9}}$. Therefore the whole derivation is not repeated. The difference between the β parameters creates a change of the drain current expressed by

$$\Delta I_{\mathrm{D,N4,5}} = I_{\mathrm{D,N4,5}} \frac{\Delta\beta_{\mathrm{N4,5}}}{\beta_{\mathrm{N4,5}}}. \tag{4.58}$$

This change causes difference between output voltages. Thus the offset voltage is calculated using the equation (4.35). After appropriate substitutions and following simplifications, the offset voltage is given by

$$V_{\mathrm{OS4}} = \frac{I_{\mathrm{D,N4,5}}}{g_{m,\mathrm{P8,9}}} \frac{\Delta\beta_{\mathrm{N4,5}}}{\beta_{\mathrm{N4,5}}}. \tag{4.59}$$

In this case, the drain current flowing through the transistor M_{P8} respectively the transistor M_{P9} is equal to the drain current flowing through the transistor M_{N4} respectively M_{N5}. Therefore it can be assumed that

$$I_{\mathrm{D,P8,9}} = I_{\mathrm{D,N4,5}} = I_{\mathrm{D}}. \tag{4.60}$$

By substitution of $g_{m,\mathrm{P8,9}}$, the offset voltage caused by $\Delta\beta_{\mathrm{N4,5}}$ is also expressed using geometrical dimensions of MOSFETs as

$$V_{\mathrm{OS4}} = \sqrt{\frac{I_{\mathrm{D}}}{2\mu_{\mathrm{m,P}}C_{\mathrm{ox}}\frac{W_{\mathrm{P8,9}}}{L_{\mathrm{P8,9}}}}}\frac{\Delta\beta_{\mathrm{N4,5}}}{\beta_{\mathrm{N4,5}}}. \tag{4.61}$$

The total offset voltage V_{OS} of the used circuit can be described by a sum of individual contributors V_{OS1}, V_{OS2}, V_{OS3}, and V_{OS4}. Thus it can be expressed as

$$V_{\mathrm{OS}} = \Delta V_{\mathrm{th,P8,9}} + \Delta V_{\mathrm{th,N4,5}}\sqrt{\frac{\mu_{\mathrm{m,N}}\frac{W_{\mathrm{N4,5}}}{L_{\mathrm{N4,5}}}}{\mu_{\mathrm{m,P}}\frac{W_{\mathrm{P8,9}}}{L_{\mathrm{P8,9}}}}} + \sqrt{\frac{I_{\mathrm{D}}}{2\mu_{\mathrm{m,P}}C_{\mathrm{ox}}\frac{W_{\mathrm{P8,9}}}{L_{\mathrm{P8,9}}}}}\left(\frac{\Delta\beta_{\mathrm{P8,9}}}{\beta_{\mathrm{P8,9}}} + \frac{\Delta\beta_{\mathrm{N4,5}}}{\beta_{\mathrm{N4,5}}}\right) \tag{4.62}$$

where $\Delta V_{\mathrm{th,P8,9}}$ is the threshold voltage error between paired transistors $\mathrm{M_{P8}}$ and $\mathrm{M_{P9}}$ with the same widths $W_{\mathrm{P8,9}}$ and the same lengths $L_{\mathrm{P8,9}}$, $\Delta V_{\mathrm{th,N4,5}}$ is the threshold voltage error between paired transistors $\mathrm{M_{N4}}$ and $\mathrm{M_{N5}}$ with the same widths $W_{\mathrm{N4,5}}$ and the same lengths $L_{\mathrm{N4,5}}$, $\mu_{\mathrm{m,N}}$ is mobility of charge carriers in N-channel MOSFETs, $\mu_{\mathrm{m,P}}$ is mobility of charge carriers in P-channel MOSFETs, $\frac{\Delta\beta_{\mathrm{P8,9}}}{\beta_{\mathrm{P8,9}}}$ is a normalized error of β parameter between paired transistors $\mathrm{M_{P8}}$ and $\mathrm{M_{P9}}$, $\frac{\Delta\beta_{\mathrm{N4,5}}}{\beta_{\mathrm{N4,5}}}$ is the normalized error of β parameter between paired transistors $\mathrm{M_{N4}}$ and $\mathrm{M_{N5}}$. Thus according to the derived equation (4.62) for offset voltage minimization, the ratio $W_{\mathrm{P8,9}}/L_{\mathrm{P8,9}}$ should be maximized, the ratio $W_{\mathrm{N4,5}}/L_{\mathrm{N4,5}}$ minimized, and the bias current also minimized because it is obvious that $I_{\mathrm{B}} = 2I_{\mathrm{D}}$.

However, the proposed circuit has to be able to use non-deterministic noise as much as possible. In the CMOS technologies used for applications above-mentioned, thermal noise and flicker noise occur mainly [106]. Thermal noise is a noise with flat frequency spectrum so-called white noise, which is usually modeled as an equivalent input noise voltage source of MOSFET [106] with the noise density described by the equation (2.33). MOSFETs also exhibit flicker noise, which is briefly described in section 2.6.2 and is modeled as the equivalent input noise voltage source of MOSFET [106] with the noise density given by the equation (2.37).

The input-referred noise density of the metastable element is calculated according to a principle mentioned in [107] when

$$\frac{\mathrm{d}\overline{v_{\mathrm{n,in}}^2}}{\mathrm{d}f} = \frac{\frac{\mathrm{d}\overline{v_{\mathrm{n,out}}^2}}{\mathrm{d}f}}{A_v^2} \tag{4.63}$$

where A_v is given by the equation (4.15) and $\frac{\mathrm{d}\overline{V^2_{\mathrm{n,out}}}}{\mathrm{d}f}$ is output power spectral noise density based on the equation (4.19) and given by

$$\frac{\mathrm{d}\overline{v^2_{\mathrm{n,out}}}}{\mathrm{d}f} = \frac{\frac{\mathrm{d}\overline{i^2_{\mathrm{n}}}}{\mathrm{d}f}}{\left(g_{\mathrm{ds,P8}} + g_{\mathrm{ds,N4}} - g_{m,\mathrm{N4}}\right)^2} \tag{4.64}$$

where $\frac{\mathrm{d}\overline{i^2_{\mathrm{n}}}}{\mathrm{d}f}$ is PSD of noise current. Using the equations (4.63) and (4.64), the input-referred noise density can be expressed as

$$\frac{\mathrm{d}\overline{v^2_{\mathrm{n,in}}}}{\mathrm{d}f} = \frac{\frac{\mathrm{d}\overline{i^2_{\mathrm{n}}}}{\mathrm{d}f}}{g^2_{m,\mathrm{P8}}}. \tag{4.65}$$

Each of the transistors $\mathrm{M_{P8}}$, $\mathrm{M_{P9}}$, $\mathrm{M_{N4}}$, and $\mathrm{M_{N5}}$ generates noise, which contributes to the total noise occurring in the metastable element. PSD of drain noise current a MOSFET is given by the sum of PSD of thermal noise current [107] described by

$$\frac{\mathrm{d}\overline{i^2_{\mathrm{n,th}}}}{\mathrm{d}f} = 4k_{\mathrm{B}}T\gamma g_m \tag{4.66}$$

and PSD of flicker noise current [107] described by

$$\frac{\mathrm{d}\overline{i^2_{\mathrm{n,f}}}}{\mathrm{d}f} = \frac{K_{\mathrm{f}}}{WLC^2_{\mathrm{ox}}}\frac{1}{f}g^2_m. \tag{4.67}$$

The metastable element is designed symmetrically for the above reasons. Therefore PSD of noise current is given by

$$\begin{aligned}\frac{\mathrm{d}\overline{i^2_{\mathrm{n}}}}{\mathrm{d}f} = 2\,\Bigg(&4k_{\mathrm{B}}T\gamma g_{m,\mathrm{P8,9}} + 4k_{\mathrm{B}}T\gamma g_{m,\mathrm{N4,5}} + \\ &+ \frac{K_{\mathrm{f,P}}}{W_{\mathrm{P8,9}}L_{\mathrm{P8,9}}C^2_{\mathrm{ox}}}\frac{1}{f}g^2_{m,\mathrm{P8,9}} + \frac{K_{\mathrm{f,N}}}{W_{\mathrm{N4,5}}L_{\mathrm{N4,5}}C^2_{\mathrm{ox}}}\frac{1}{f}g^2_{m,\mathrm{N4,5}}\Bigg)\end{aligned} \tag{4.68}$$

where $K_{\mathrm{f,P}}$ is the flicker noise coefficient for P-channel MOSFETs and $K_{\mathrm{f,N}}$ for N-channel MOSFETs. Using equations (4.68) and (4.65) and after subsequent simplifications, the input-referred noise density can be expressed as

$$\begin{aligned}\frac{\mathrm{d}\overline{v^2_{\mathrm{n,in}}}}{\mathrm{d}f} = 2\Bigg[&\frac{4k_{\mathrm{B}}T\gamma}{g_{m,\mathrm{P8,9}}}\left(1 + \frac{g_{m,\mathrm{N4,5}}}{g_{m,\mathrm{P8,9}}}\right) + \\ &+\frac{K_{\mathrm{f,P}}}{W_{\mathrm{P8,9}}L_{\mathrm{P8,9}}C^2_{\mathrm{ox}}}\frac{1}{f}\left(1 + \frac{K_{\mathrm{f,N}}W_{\mathrm{P8,9}}L_{\mathrm{P8,9}}}{K_{\mathrm{f,P}}W_{\mathrm{N4,5}}L_{\mathrm{N4,5}}}\left(\frac{g_{m,\mathrm{N4,5}}}{g_{m,\mathrm{P8,9}}}\right)^2\right)\Bigg].\end{aligned} \tag{4.69}$$

From design point of view, it is important to express the equation (4.69) using geometric dimensions of the MOSFETs. Thus the total input-referred noise density is given by

$$\frac{\mathrm{d}\overline{v_{\mathrm{n,in}}^2}}{\mathrm{d}f} = 2\left[\frac{4k_{\mathrm{B}}T\gamma}{\sqrt{2\mu_{\mathrm{m,P}}C_{\mathrm{ox}}\frac{W_{\mathrm{P8,9}}}{L_{\mathrm{P8,9}}}I_{\mathrm{D}}}}\left(1+\sqrt{\frac{\mu_{\mathrm{m,N}}\frac{W_{\mathrm{N4,5}}}{L_{\mathrm{N4,5}}}}{\mu_{\mathrm{m,P}}\frac{W_{\mathrm{P8,9}}}{L_{\mathrm{P8,9}}}}}\right)+\right.$$
$$\left.+\frac{K_{\mathrm{f,P}}}{W_{\mathrm{P8,9}}L_{\mathrm{P8,9}}C_{\mathrm{ox}}^2}\frac{1}{f}\left(1+\frac{\mu_{\mathrm{m,N}}K_{\mathrm{f,N}}}{\mu_{\mathrm{m,P}}K_{\mathrm{f,P}}}\left(\frac{L_{\mathrm{P8,9}}}{L_{\mathrm{N4,5}}}\right)^2\right)\right]. \quad (4.70)$$

The first part of the equation (4.70) represents thermal noise influence, which can be maximized by decreasing the ratio $W_{\mathrm{P8,9}}/L_{\mathrm{P8,9}}$, increasing the ratio $W_{\mathrm{N4,5}}/L_{\mathrm{N4,5}}$, and decreasing the drain current I_{D}. Similarly, the second part of the equation (4.70) describes flicker noise influence, which can be maximized by decreasing the product $W_{\mathrm{P8,9}}L_{\mathrm{P8,9}}$ and increasing the ratio $L_{\mathrm{P8,9}}/L_{\mathrm{N4,5}}$.

Influence	Recommendation	
Offset voltage ↓	$W_{\mathrm{P8,9}}/L_{\mathrm{P8,9}}$ ↑	$W_{\mathrm{N4,5}}/L_{\mathrm{N4,5}}$ ↓
Thermal noise ↑	$W_{\mathrm{P8,9}}/L_{\mathrm{P8,9}}$ ↓	$W_{\mathrm{N4,5}}/L_{\mathrm{N4,5}}$ ↑
Flicker noise ↑	$W_{\mathrm{P8,9}}L_{\mathrm{P8,9}}$ ↓	$L_{\mathrm{P8,9}}/L_{\mathrm{N4,5}}$ ↑

Tab. 4.1: Summary of found recommendations for the noise source proposal [4]

Transistor dimensions of this most sensitive part are proposed in the light of the above considerations, which are summarized in table 4.1. Naturally, dimensions of all used components are chosen to fit the designed circuit into a predefined area on a die. Recommendations for minimizing the offset voltage and prerequisites for maximizing thermal noise are contradictory. Therefore MOSFETs M_{P8}, M_{P9}, M_{N4}, and M_{N5} have been proposed with the same dimensions. In order to maximize flicker noise influence, channel areas of transistors M_{P8} and M_{P9} have been minimized by shortening their lengths L_{P8} and L_{P9}. The proposed MOSFET dimensions have been fine-tuned by numerical simulations of the whole circuit, especially by a noise transient simulation in the Mentor Eldo simulator [14]. Thus the final transistors dimensions are $W_{\mathrm{P8,9}} = W_{\mathrm{N4,5}} = 5\,\mu\mathrm{m}$ and $L_{\mathrm{P8,9}} = L_{\mathrm{N4,5}} = 0.5\,\mu\mathrm{m}$.

The operating conditions of the circuit depicted in figure 4.3 are set by a reference current $I_{\mathrm{REF}q}$ of $1\,\mu\mathrm{A}$. Current distribution inside the circuit is made by current mirrors.

The current mirror composed of P-channel MOSFETs M_{P1}, M_{P2}, and M_{P3} is used in a cascode variant, which is created from transistors M_{P4}, M_{P5}, and M_{P6}. A current branch generating the voltage V_{IN} for inputs of the differential pair copies the reference current in a ratio of 1 : 1. A ratio of the differential pair bias current created by the MOSFET M_{P3} is proposed with regard to offset voltage minimization when according to equation (4.62) currents flowing transistors M_{P8} and M_{P9} should be decreased. The total current consumption of the whole circuit has to be also minimized due to minimization of the power consumption of the whole proposed TRNG. However, this bias current also has to be set with regard to a random data rate because higher data rate requires the higher bias current. The presented TRNG is designed for the output random data rate of 20 Mb/s. Therefore, after verification by simulations, the current for the differential pair is set in the ratio 4 : 1 to the reference current $I_{\mathrm{REF}q}$.

The described TRNG is designed for use in complex ICs where a number of different parts work at the same time. In these ICs, deterministic disturbance arises and spreads across the chip, especially via the substrate and the power supply. This deterministic disturbance can occur in the output random data stream as a deterministic component. In other words, the output random data stream can be deliberately influenced and a system using some TRNG can be attacked in this way. Therefore systems with TRNGs must be resistant to this type of attack.

In presented design, the power supply and ground are divided into two separated parts. The first part is intended to supplying sensitive analog blocks, that is all noise sources. Moreover, the power supply and ground of each noise source are star-routed due to prevention of mutual distortion. Digital blocks of the proposed TRNG are supplied with the other part of the power supply and ground. These blocks operate with digital signals and cannot be easily affected by the described non-invasive attack. The presented generator does not show any deterministic components in the output random data stream during simulations of the power supply distortion as well as during on-chip verification. Nevertheless, to increase security, a low drop out regulator without any external capacitor integrated into the same die is suitable to use because there is no possibility to attack via external supply pin.

4.2.2 Digitizer

A block transferring random values from the analog part of the proposed TRNG to the digital part is marked as the digitizer and is shown in figure 4.8. The random signal $V_{\mathrm{O,M}q}$ in analog supply domain is transformed into the digital signal V_{sq} in digital supply domain. The random values have to be transferred without any damage and synchronized with rest of the system. This block separates both power supply domains as well. The sources of randomness shown in figure 4.2 are created by connection of the digitizer to the noise source.

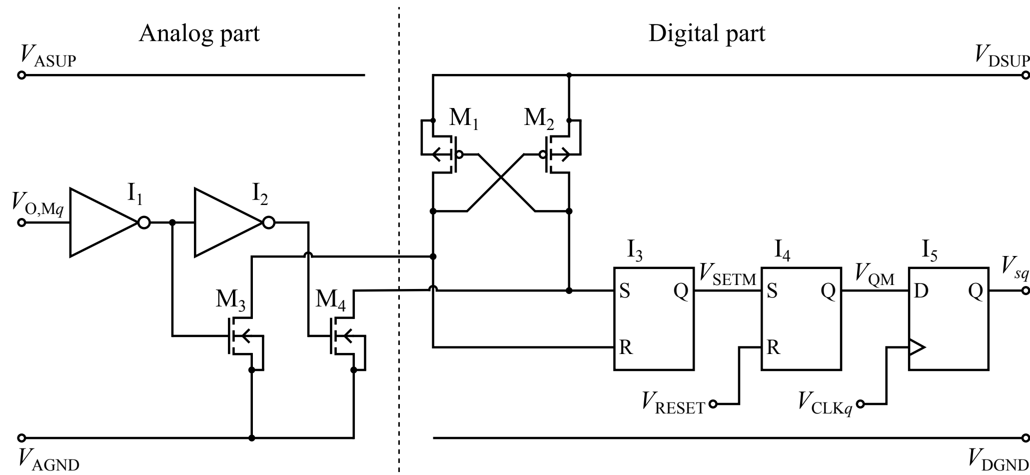


Fig. 4.8: The schematic diagram of the designed digitizer [4]

The random values are transferred between both supply domains by differential signals, which are created by standard CMOS inverters I_1 and I_2. These standard cells also suitably shape the input signal $V_{\mathrm{O,M}q}$. A structure consisting of transistors M_1, M_2, M_3, M_4 and commonly used in level shifters perform the transmission itself and drives an RS latch I_3 composed of standard CMOS NOR gates. The large digital circuits in complex ICs usually create supply voltage distortion in form of voltage glitches, which can cause malfunctions of other circuits connected to the same supply voltage. The incorporated RS latch I_3 is able to hold a logic value during supply voltage fluctuations and thus increase system reliability.

The output signal V_{SETM} of the RS latch I_3 is still not suitable for further processing in the time multiplexor because a logic value of the signal cannot be reliably read in the next rising edge of the clock signal $V_{\mathrm{CLK}q}$ as it is shown in figure 4.4. Therefore the signal

V_{RESET} is formed in the clock generator. Using this, a part of the signal V_{SETM} containing the random logic value is extended in the RS latch I_4 so that it can be synchronized with $V_{\mathrm{CLK}q}$ in the D latch I_5, which produces a further usable signal V_{sq}.

Slightly asymmetric or too large parasitic capacitances at the noise source outputs can cause the TRNG failure. The noise source can be locked in one logic value. Therefore from the layout point of view to maintain symmetry, the digitizer is connected to each noise source output but only one of them is used for next signal processing. The digitizer is also placed as near as possible to the noise source outputs due to parasitic capacitance reduction of interconnections $V_{\mathrm{O,M}q}$ and $V_{\mathrm{O,P}q}$. For the same reason, inverters with the smallest possible MOSFETs are used at the digitizer input. A layout of the most sensitive part of the TRNG – the symmetric metastable element connected to input parts of the digitizers – is depicted in figure 4.9.

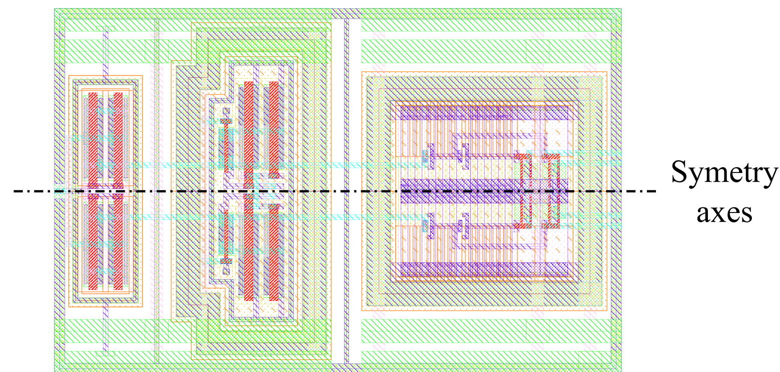


Fig. 4.9: The layout of the metastable element connected to input parts of the digitizers with a drawn symmetry axis

4.2.3 Time Multiplexer

Each source of randomness is designed to generate the digital random signal V_{sq} with clock frequency 5 MHz. The time multiplexer depicted in figure 4.10 combines all four generated signals V_{sq} shown in figure 4.11 and forms the internal raw random signal $V_{r,\mathrm{i}}$ with four times higher clock frequency. Transmission gates are opened consecutively by signals $V_{\mathrm{SN}q}$ controlling N-channel MOSFETs $\mathrm{M}_{\mathrm{SN}q}$ and their negations controlling P-channel MOSFETs $\mathrm{M}_{\mathrm{SP}q}$. All control signals are generated in the clock generator and derived from the fast clock signal V_{OSC}.

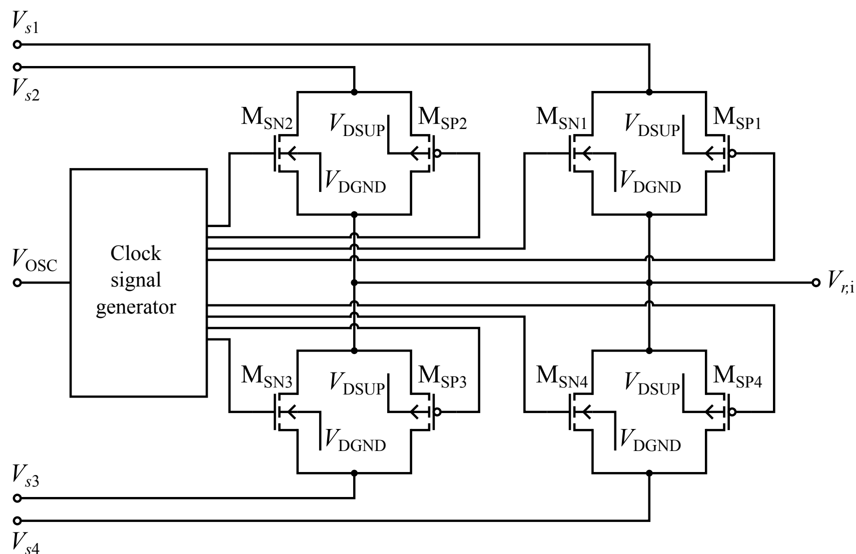


Fig. 4.10: The schematic diagram of the time multiplexer [4]

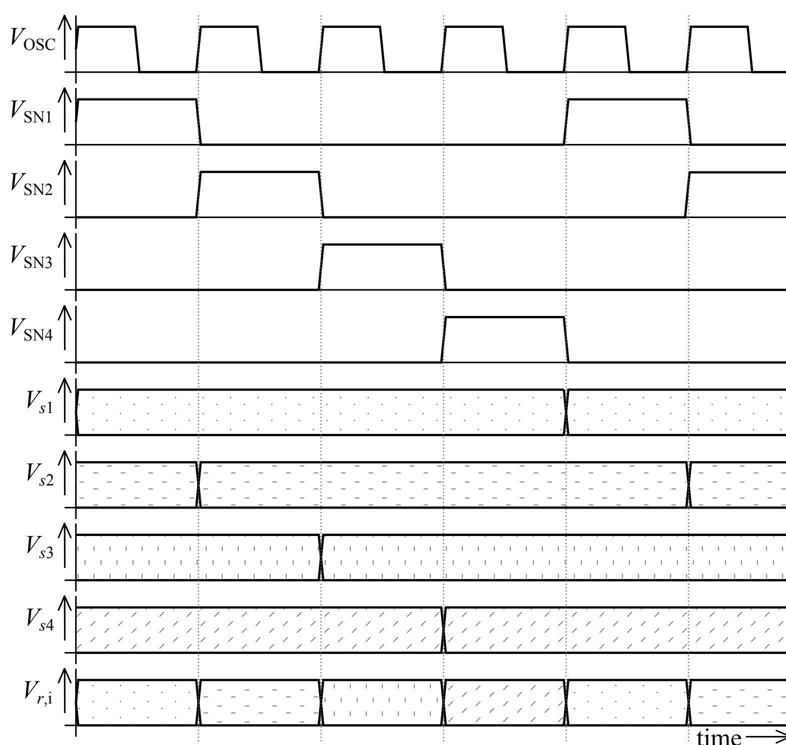


Fig. 4.11: Waveforms illustrating the function of the time multiplexer [4]

More circuits switch on at the same time then peaks on the power supply can appear. The sources of randomness generate the digital random signals V_{sq} with a shift of one

quarter the period $V_{\mathrm{CLK}q}$ among them. It reduces peak current consumption and thereby power supply distortion.

The internal raw random signal $V_{r,\mathrm{i}}$ does not have sufficient capability to drive external circuits or measuring instruments. Hence an output buffer increasing load-driving capability is incorporated into the proposed TRNG structure and produces the raw random signal V_r, which is accessible from a pin on the manufactured chip.

The designed TRNG is a part of a multi-project test chip, which consists of four projects and was funded by the Grant Agency of the Czech Technical University in Prague, grant No. SGS17/188/OHK3/3T/13. The second project contains a block for switch-mode power supply. A mismatch among different types of MOSFETs is studied in the third project. The aim of the last project is testing new electrostatic discharge ESD protective structures in customized pads. The first three project are protected by pads containing standard ESD protections available in the used process. A layout of the whole test chip is depicted in figure 4.12.

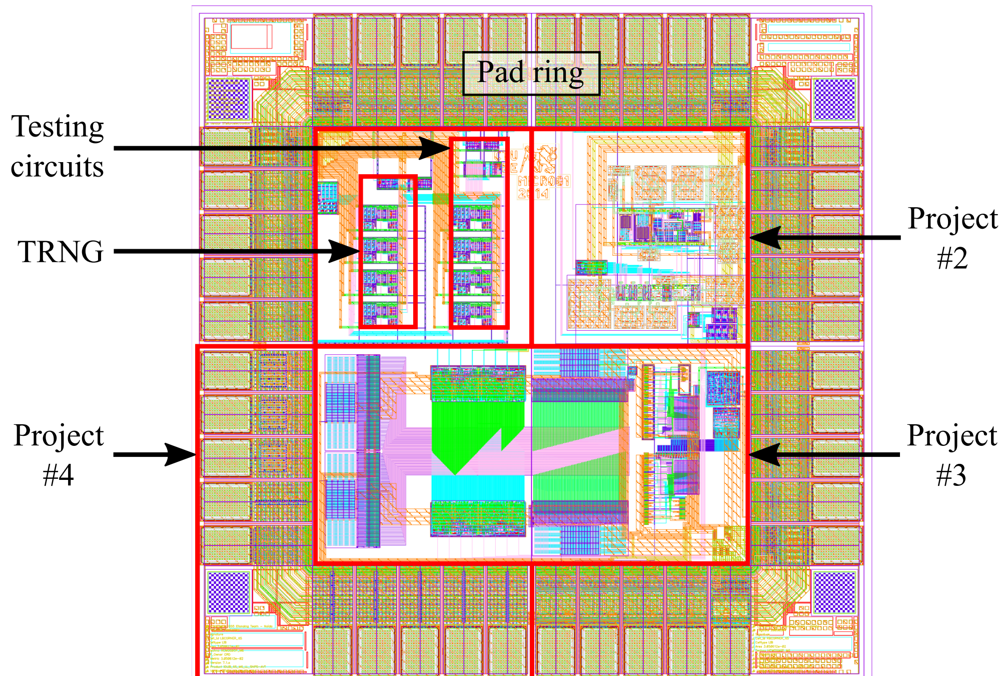


Fig. 4.12: The layout of the multi-project test chip without displayed tiles for better layer planarization

The die photo is shown in figure 4.13. The designed device occupies an area of 0.029 mm^2 including all described parts – the noise sources, the digitizer, the time mul-

tiplexer, and the output buffer. In figure 4.13, there are not almost visible any structures, which are visible in figure 4.12 because the fabricated die contains so-called tiles, which are intended to better planarization of individual layers during chemical-mechanical polishing. For unambiguous identification, the fabricated chip includes one of the smallest emblems of the Czech Technical University in Prague, whose microphotography is shown in figure 4.14.

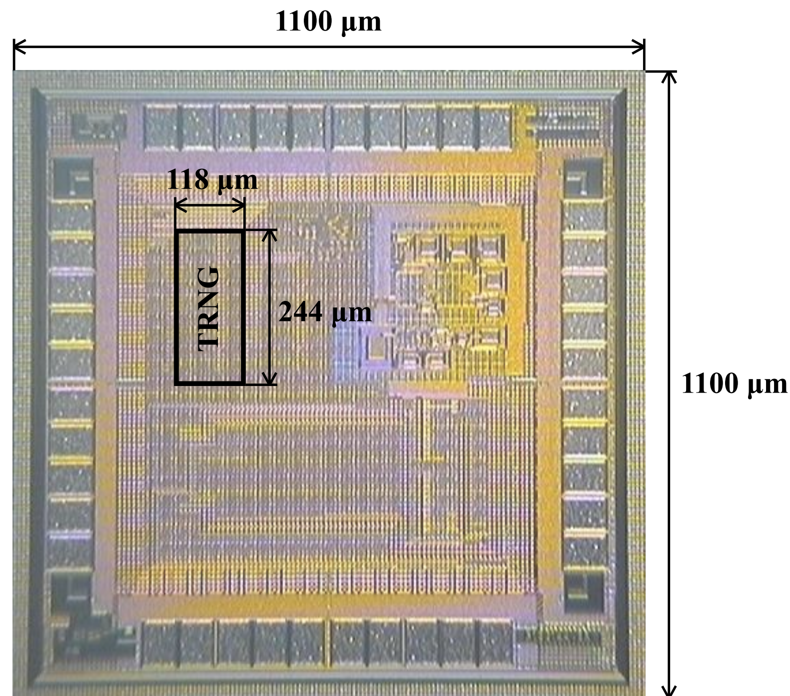


Fig. 4.13: Photo of the fabricated die. The proposed TRNG containing all described parts – the noise sources, the digitizer, the time multiplexer, and the output buffer – occupies the marked area [4]

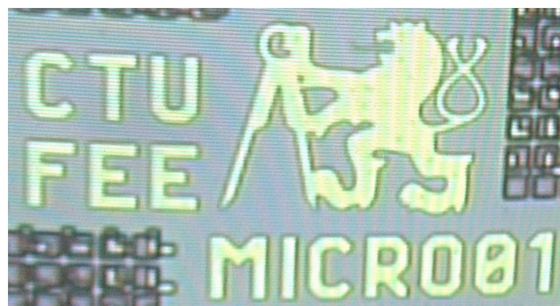


Fig. 4.14: One of the smallest emblems of the Czech Technical University in Prague with the width of 59.55 µm and the heigth of 44.85 µm is present on the fabricated chip

4.3 Measurement Results

The designed TRNG integrated on the fabricated chip has been characterized and validated both from the perspective of quality of generated random number sequences and from the perspective of integration into a system. All random sequences have been tested by the statistical test suite FIPS [63] and by the stricter statistical test suites NIST [64]. These statistical test suites are briefly described in chapter 3 and can reveal a bias, repeating patterns, or unbalanced distribution of random data.

4.3.1 Arrangement of Measuring Instruments

Before the presentation of the measured results, an arrangement of measuring instruments shown in figure 4.15 is described. First, it was necessary to ensure the power supply of the manufactured chip, which was supplied by DC voltage of 1.2 V. The supply voltage was generated by Keithley 2400 SourceMeter, which provides precision voltage and current sourcing as well as measurement capabilities.

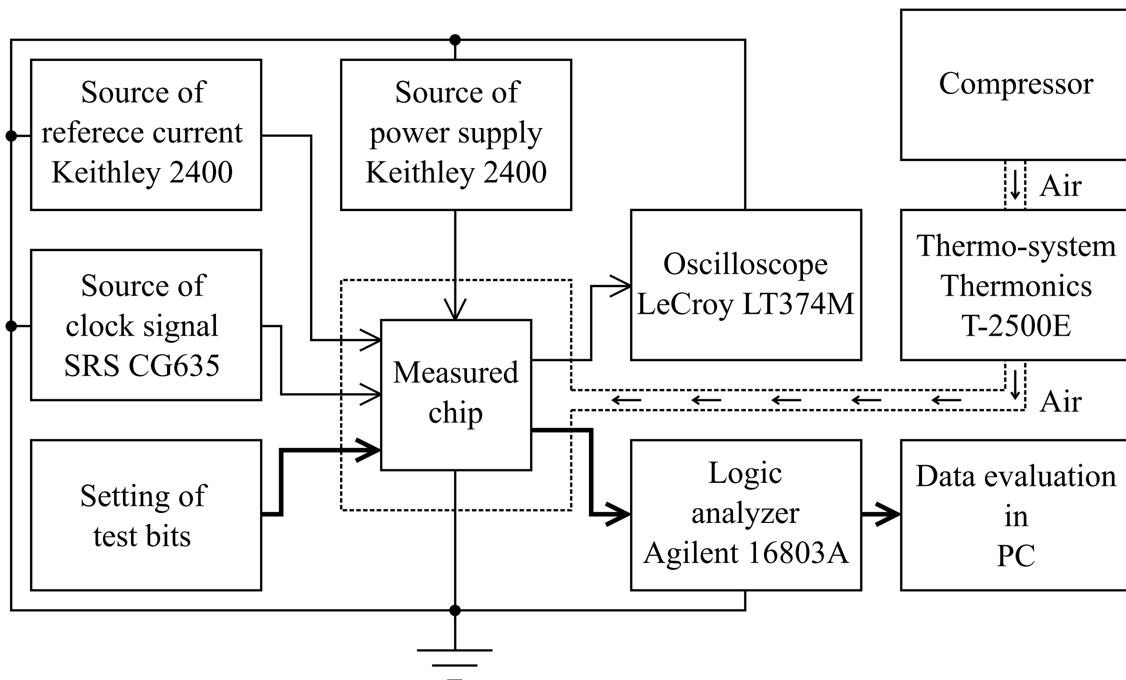


Fig. 4.15: Diagram of the measuring instrument arrangement

The measured TRNG has been designed to be integrated as one part of SoC where the reference current and the clock signal are generated in so-called common blocks,

which contain oscillators and sources of reference voltages and currents for the whole system. Thus, during measurement, it was necessary to deliver the reference current and the clock signal V_{OSC}. The reference current was also generated by the second Keithley 2400 SourceMeter. A clock generator named Stanford Research Systems CG635, which is able to produce very stable square wave clock signals, generated the clock signal V_{OSC} for all measurement runs.

During measurement, the temperature of each measured sample has been controlled by a precision temperature forcing system Thermonics T-2500E, which uses compressed air for its operation. Thus the compressed air was produced by a compressor with an incorporated filter removing oil residues. This arrangement allows measuring random number sequences generated at various temperatures, as can be seen from the measured results mentioned below.

The fabricated chip also contains testing structures, which allow connecting some internal signals to an external pad. This function is important during a debugging phase when the internal signals are checked if their waveforms and behavior of the TRNG are as expected. The internal signals were switched by a testing multiplexer, which was set by three testing bits in this case. Available signals were observed by a digital oscilloscope LeCroy LT374M.

The random number sequences generated in all defined operation points of the TRNG were read by a 102-channel logic analyzer Agilent 16803A. Subsequent evaluation of random number sequences was performed by the statistical test suites described in chapter 3 in a personal computer (PC).

4.3.2 Evaluation of Generated Random Number Sequences

All 15 statistical NIST tests have been performed with a significance level α equal to 0.01. Thus each test has passed, if the computed P-value would have been equal or greater than 0.01. For NIST tests, 1 Mb long random sequences have been generated by the output random data rate from 10 Mb/s to 60 Mb/s in the temperature range of −35 °C to 85 °C. For FIPS tests, the random sequences have been shortened to the desired 20 kb.

The generator presented has been optimized for the random data rate of 20 Mb/s. However, the theoretical limit is slightly more than 60 Mb/s because the duration of the decision phase of the noise source output signal $V_{\mathrm{O,M}q}$ described in section 4.2.1 does

not depend on the clock frequency. While increasing the clock frequency, the duration of the stable state shortens but the duration of the decision phase does not change. For this reason, when increasing the clock frequency above the limit, the noise source is not able to move to the stable state and generate any random bits. In other words, when the output random data rate rises above the theoretical limit, there is not enough time after the decision phase for generation of a new random bit. Faultless FIPS and NIST tests have confirmed functionality of the designed TRNG at the output random data rate of 20 Mb/s and at the ambient temperature of 25 °C. Further, the TRNG was able to produce the output random data without failure even with the rate 60 Mb/s, which coincides with the assumed theoretical limit. At the ambient temperature of 25 °C and all measured data rates, the generated random data passed the FIPS tests but failed in some NIST tests as can be seen in tables 4.2, 4.3, 4.4, and 4.5 where the total number of subtests and the number of passed subtests are displayed. An essential part of the results has been published in [4].

As described above, deterministic disturbance of the power supply can affect the quality of generated random sequences. Hence the random sequences were generated with artificially created power supply distortion, which simulated power supply distortion with the voltage spike frequency in the range of 10 MHz to 100 MHz coming from digital circuits. Results of the FIPS and NIST tests are consistent with the results obtained from the output random data generated at the undistorted power supply. Of this comparison, it can be assumed that the above-described distortion of the power supply does not affect the output random data.

From the results obtained, it is obvious that the proposed TRNG has worked well at the output random data rates 10 Mb/s and 20 Mb/s at all measured temperatures. However, at maximum data rates, the NIST tests have revealed periodic features in the output sequences, which has been caused by deterministic noise present in the circuit. At the highest measured temperatures and the highest data rates, an imbalance between zeros and ones has been observed, which has been demonstrated by unmet Frequency tests from both NIST and FIPS test suites. This bias has been caused by the occasional locking of the generator in one logic value. All results of NIST and FIPS tests are listed in tables 4.2, 4.3, 4.4, and 4.5.

The approximate entropy (ApEn) defined in [120] is used to determine the amount of regularity and the unpredictability of fluctuations in random bit streams. Thus small

	Results at output random data rates																	
	10 Mb/s									20 Mb/s								
Temperature (°C)	-35	-20	-5	10	25	40	55	70	85	-35	-20	-5	10	25	40	55	70	85
Monobit	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
Poker	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
Runs	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
Long run	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
	30 Mb/s									40 Mb/s								
Monobit	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
Poker	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
Runs	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
Long run	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
	50 Mb/s									60 Mb/s								
Monobit	P	P	P	P	P	P	P	P	F	P	P	P	P	P	P	P	F	F
Poker	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	F	F
Runs	P	P	P	P	P	P	P	P	F	P	P	P	P	P	P	P	P	F
Long run	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P

Tab. 4.2: Results of the FIPS tests (P – Passed; F – Failed)

	Number of subtests	Number of passed subtests at output random data rates																	
		10 Mb/s									20 Mb/s								
Temperature (°C)		-35	-20	-5	10	25	40	55	70	85	-35	-20	-5	10	25	40	55	70	85
Monobit	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Frequency	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Runs	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Longest runs	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Binary matrix rank	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Spectral DFT	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Non-overlapping template	148	148	148	148	148	148	148	147	148	147	148	147	148	148	148	148	148	147	148
Overlapping template	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Universal statistical	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Linear complexity	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Serial	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
Approximate entropy	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Cumulative sums	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
Random excursions	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
Random excursions variant	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18

Tab. 4.3: Results of the NIST tests at the output random data rates 10 Mb/s and 20 Mb/s

		Number of passed subtests at output random data rates																	
	Number of subtests	30 Mb/s									40 Mb/s								
Temperature (°C)		-35	-20	-5	10	25	40	55	70	85	-35	-20	-5	10	25	40	55	70	85
Monobit	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Frequency	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Runs	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Longest runs	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Binary matrix rank	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Spectral DFT	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Non-overlapping template	148	146	147	148	147	147	147	146	147	145	147	148	146	145	146	148	147	144	145
Overlapping template	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Universal statistical	1	1	1	1	1	0	0	1	1	0	1	1	0	1	0	1	1	0	1
Linear complexity	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Serial	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
Approximate entropy	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1
Cumulative sums	2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	2	2	2	1
Random excursions	8	8	8	8	8	8	8	8	8	8	8	8	7	8	7	8	8	7	7
Random excursions variant	18	18	18	18	18	18	18	18	18	18	18	18	18	18	16	18	18	15	15

Tab. 4.4: Results of the NIST tests at the output random data rates 30 Mb/s and 40 Mb/s

	Number of subtests	Number of passed subtests at output random data rates																	
		50 Mb/s									60 Mb/s								
Temperature (°C)		-35	-20	-5	10	25	40	55	70	85	-35	-20	-5	10	25	40	55	70	85
Monobit	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	0
Frequency	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	0
Runs	1	1	1	1	1	0	1	0	0	0	0	1	1	0	0	1	1	0	0
Longest runs	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Binary matrix rank	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Spectral DFT	1	1	1	1	1	0	1	1	0	0	0	1	1	0	0	1	0	0	0
Non-overlapping template	148	134	141	131	128	121	134	114	107	102	98	121	134	109	104	112	107	92	71
Overlapping template	1	1	1	1	1	1	1	1	1	1	0	1	1	1	0	1	1	0	0
Universal statistical	1	1	1	1	1	0	1	0	1	0	1	0	1	1	0	1	1	0	0
Linear complexity	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Serial	2	2	2	2	2	0	1	2	1	0	1	2	1	1	2	1	1	2	0
Approximate entropy	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1	1	0	0
Cumulative sums	2	2	2	2	2	2	2	2	2	2	1	2	2	1	2	1	1	0	0
Random excursions	8	8	6	8	6	8	6	8	4	4	2	7	8	5	4	2	0	0	0
Random excursions variant	18	18	18	16	18	15	18	15	12	12	10	16	18	10	12	2	0	0	0

Tab. 4.5: Results of the NIST tests at the output random data rates 50 Mb/s and 60 Mb/s

values of ApEn imply strong regularity or persistence, and large values of ApEn imply substantial fluctuation or irregularity [120]. The ApEn calculation is a part of the NIST test suite. The ApEn values of all sequences are shown in figure 4.16. The approximate entropy of sequences generated with the lower data rates is high and close to the maximum value of the natural logarithm of 2. However, the ApEn values of sequences generated with the higher data rates are lower. Even the ApEn values at high temperatures and data rates are very low, which proves the presence of the already mentioned bias in these data sequences.

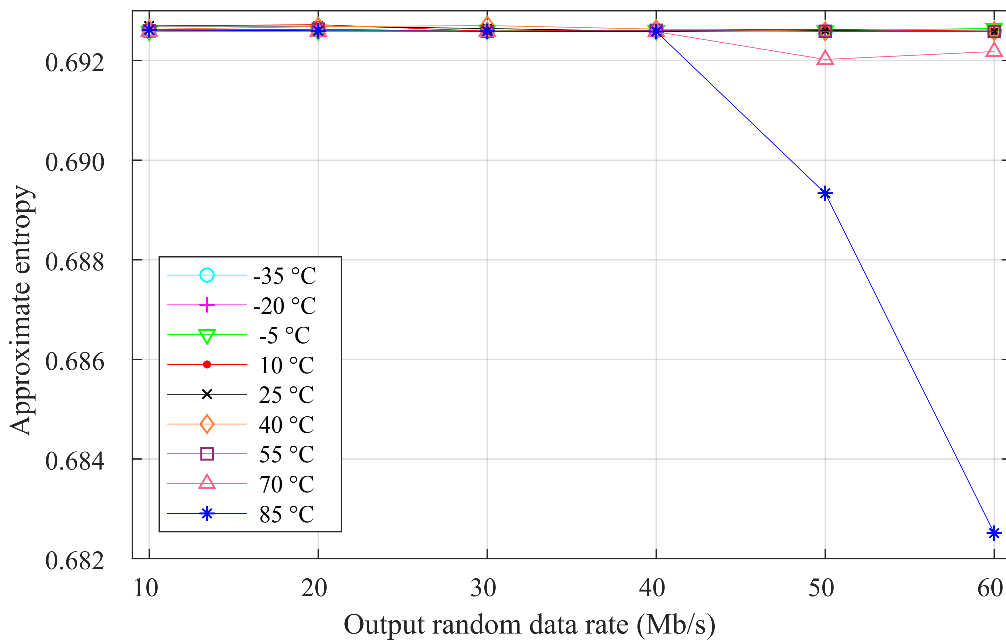


Fig. 4.16: The approximate entropy of output random number sequences generated in the temperature range of −35 °C to 85 °C [4]

4.3.3 Current Consumption

This TRNG is designed for use within complex SoCs. Therefore, from the point of view of the whole system, it is necessary to know the consumption in defined operation points. The used instrument arrangement depicted in figure 4.15 allowed its measurement in all defined operation points. Thus the current consumption was also measured in the temperature range of −35 °C to 85 °C for the output random data rate from 10 Mb/s to 60 Mb/s and results are depicted in figure 4.17. At the ambient temperature of 25 °C and at the output random data rate of 20 Mb/s, the current consumption is 60.4 µA.

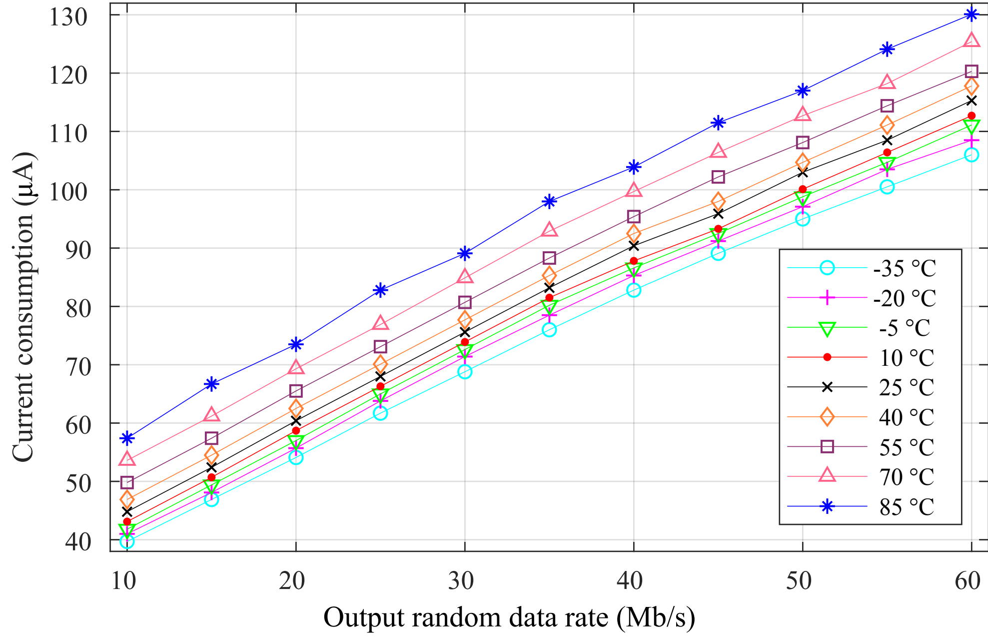


Fig. 4.17: The current consumption of the designed circuit measured in the temperature range of −35 °C to 85 °C [4]

4.3.4 Required Energy per Random Bit

Some published TRNGs integrated into a chip can be evaluated according to required energy per random bit, which is usually calculated according to

$$E_\mathrm{b} = \frac{P_\mathrm{cons}}{B_\mathrm{out}} \tag{4.71}$$

where P_cons is the power consumption of the TRNG and B_out is its output random data rate. This measure expresses an average amount of energy, which is needed to generate one random bit. However, it does not evaluate the quality of random number sequences. Therefore it is usually stated together with the results of statistical test suites. The energy per random bit of the designed TRNG is shown in figure 4.18 where it can be seen that the generator produces random number sequences with very low values of E_b in all operation points.

Also using the energy per random bit, the already published TRNGs can be compared. Table 4.6 shows a comparison among selected TRNGs based on used technology node, source of randomness type, occupied area on a chip A_TRNG, power supply voltage V_SUP, power consumption P_cons, output random data rate B_out, and energy per random bit E_b.

TRNG	Technology node (nm)	Source of randomness type	A_{TRNG} (mm^2)	V_{SUP} (V)	P_{cons} (µW)	B_{out} (Mb/s)	E_b (pJ/b)
Mathew [46]	14	Metastability	0.001008	0.75	1500	162.5	9.23
Yang [122]	28	Ring oscillators	0.000375	0.9	540	23.2	23.28
Srinivasan [41]	45	Metastability	0.001024	0.5 – 1.1	2280	4000	0.57
Srinivasan [123]	45	Metastability	0.004004	0.28 – 1.35	7000	2400	2.92
This work	130	Metastability	0.029000	1.2	72.48	20	3.62
Tokunaga [39]	130	Metastability	0.145000	1.2	1000	0.2	5000.00
Pareschi [56]	180	Chaos	0.126000	1.8	22000	48	458.33
Coustans [35]	180	Ring oscillators	0.003172	1.0 – 1.8	0.015	0.0005	30.00
Eberlein [24]	180	Noise amplification	0.008008	1.8	25	5	5.00
Guler [31]	250	Ring oscillators	0.043000	2.5	40000	16.5	2424.24
Park [60]	350	Chaos	0.057000	1.0 – 5.0	26100	300	87.00
Holleman [40]	350	Metastability	0.031000	5.0	9.39	0.005	1878.00
Petrie [53]	2000	Combination	1.500000	3.0	39000	1.4	27857.14

Tab. 4.6: Parameters including required energy per random bit of published TRNGs

From this table, it is obvious that the designed TRNG produces random number sequences with one of the lowest energy per random bit while the total power consumption is low.

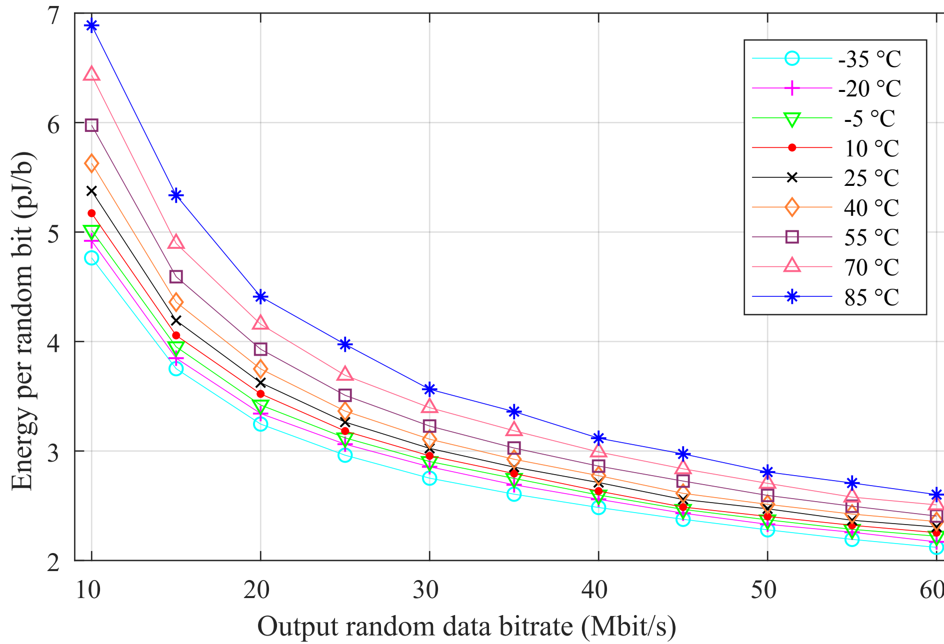


Fig. 4.18: The energy per random bit of the designed TRNG calculated in the temperature range of −35 °C to 85 °C

4.4 Comparison

Based on measurements and statistical test results, it might be said that the fabricated TRNG can be integrated inside SoCs for generation of random bit sequences with the bit rate up to 20 Mb/s at above-mentioned temperatures without incorporating any corrector. When used at higher random data rates, the TRNG should be supplemented with any corrector mentioned in section 2.1.3 to improve the properties of random sequences. However, use of the generator at the highest random data rate is not appropriate because the results of both statistical test suites at the highest temperatures show failures.

The designed TRNG has been fabricated in the 130 nm bulk CMOS technology as the part of the multi-project test chip where occupies the area of 0.029 mm^2 and consumes 72.48 µW at the ambient temperature of 25 °C and the output random data rate of 20 Mb/s. A direct comparison can be done with the TRNG published in [39] and fabricated by the similar 130 nm bulk CMOS process. This generator occupies a larger area of 0.145 mm^2,

has a higher power consumption of 1 mW, and a lower random data rate of 200 kb/s. Both TRNGs resist temperature and power supply variations.

Very high random data rates have been achieved by TRNGs published in [41], [123], [45], or [46], which have been fabricated in advanced node CMOS processes optimized for higher clock frequencies of digital circuits. They have much higher power consumption, and that can be an obstacle for use in some applications such as hand-held devices due to energy saving. The introduced architecture in this work allows reducing the power consumption, while the output random data rate remains high enough. Because the designed TRNG does not contain passive analog devices it simplifies its migration to advanced process nodes to increase TRNG random data rate and decrease the power consumption even more.

Chapter 5

Protective Mechanisms for TRNGs

Safety of contemporary communication systems is dependent on the unpredictability of random number sequences produced by RNGs. Therefore it is advisable to generate the random number sequences using TRNGs for the reasons described in previous chapters. However, in these cases, the safety of the systems may be impaired by deliberate malicious attacks on the TRNGs integrated into these communication systems. Then the produced random number sequences are manipulated, and attackers are able to guess parts or even entire sequences.

The attacks against TRNGs differ according to the method of execution. Attack types are summarized in [124]. Passive attacks do not directly affect the generation of random number sequences, but the sequences are estimated based on measurements of external manifestations of the system such as actual power consumption or emissions of the electromagnetic field. Active attacks can be conducted non-invasively when the sequences are affected by changing environmental conditions such as intentional changes of surrounding electromagnetic field, supply voltage, or temperature. In this way, a bias can be introduced into the probability distribution of logic values produced by the TRNG. The TRNG behavior can also be influenced by a destruction of the whole generator or its parts, which can be marked as the invasive active attack.

TRNGs are usually parts of SoCs. Therefore these systems should not have any accessible external pin dedicated only to a power supply of the TRNG and the TRNG power consumption should be negligible compared to the power consumption of the whole system. In this manner, it is possible to limit the passive attacks. Also the invasive active attacks are very complicated. The destruction of the TRNG only is almost impossible

without damaging the other parts of the system because the systems are usually designed as complex ICs fabricated in submicron technologies. So it is necessary to have a very expensive device such as a focused ion beam (FIB) instrument and to know details of the physical design of the whole SoC containing the TRNG. The most dangerous attacks are active non-invasive because the attackers do not need any expensive equipment. The proposed mechanisms described in this chapter are able to detect and eliminate the non-invasive active attacks. Parts of this chapter have been published by the author of this thesis in [5].

5.1 Enhanced Generic Architecture

The conventionally used generic architecture of TRNGs described in section 2.1 does not allow testing of random number sequences during their generation. Thus any deliberately introduced bias of random number sequences cannot be detected, and the communication system can be attacked without noticing. The disadvantage mentioned above is eliminated by the new enhanced architecture of TRNGs that allows testing of random number sequences at the hardware level during their generation.

Each TRNG can be attacked by deliberate malicious attacks for the purpose to affect random number sequences. Then these sequences lose random properties. Such attacks threaten the security of communication and thus cause security failure of the whole system. However, not only attacks threaten the system security. Loss of randomness can also be caused by deterministic noises such as deterministic distortion of power supply and chip substrate, or regular temperature fluctuations. This deterministic disturbance naturally occurs in complex ICs where a lot of various blocks work at the same time. Therefore, the proposed architecture contains mechanisms, which can detect the bias of random number sequences and also reveal a significant decrease in the entropy of sources of randomness.

Security elements of TRNGs were published in [97] but this thesis presents the new complex extension of the generic architecture with a description of used mechanisms. A block diagram of the proposed enhanced generic architecture (EGA) of TRNGs is shown in figure 5.1. After attack detection, implemented mechanisms create a notification, which stops the output stream of random numbers in order to the communication

system is not endangered. The notification is sent to other TRNG blocks – the post-processing block and the output interface. It is also available to a master system, which is informed about the state of generated sequences.

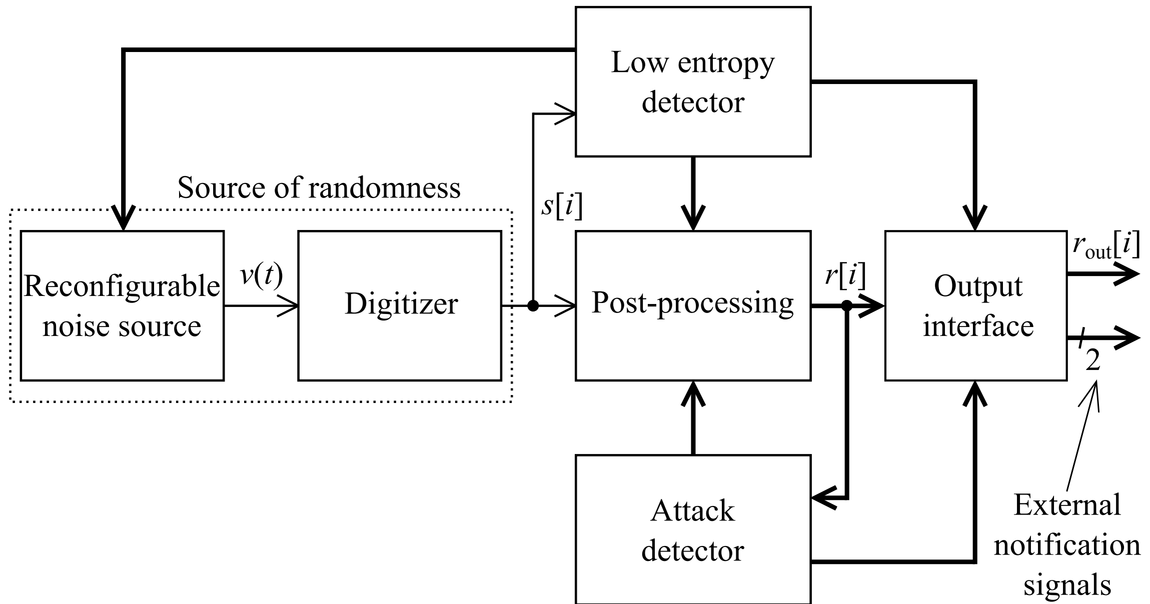


Fig. 5.1: The block diagram of the proposed EGA

The attacker tries to modify properties of the source of randomness. However, the post-processing block incorporated in the TRNG structure is capable of removing imperfections of random data to some extent, but large imperfections caused by an attack cannot be solved by the post-processing. Therefore a block detecting possible attacks – an attack detector – works with the internal random numbers $r[i]$, which are produced by the post-processing block and should have suitable properties. If known parameters of the internal random numbers $r[i]$ are changed, the attack detector creates the notification that the malicious attack is detected.

The external random numbers $r_{out}[i]$ produced by the TRNG must be unpredictable. In other words, $r_{out}[i]$ must meet requirements of the statistical test suites and have sufficient entropy, which is described in section 3.3. The decrease in the entropy of $r_{out}[i]$ can be evaluated as loss of randomness. Therefore it is appropriate to continuously monitor the level of the entropy and create the notification when the entropy significantly drops.

If the entropy of the digitized noise signal $s[i]$ is sufficient, then the entropy of the internal random numbers $r[i]$ is also sufficient for the reason mentioned in section 2.1.3.

Thus the post-processing block solves imperfections of $s[i]$. Moreover, the output interface does not affect the quality of random number sequences. So the entropy of the external random numbers $r_{\mathrm{out}}[i]$ is the same as the entropy of the internal random numbers $r[i]$. Based on these claims, it is evident that if the entropy of the digitized noise signal $s[i]$ is sufficient, then the generated external random numbers $r_{\mathrm{out}}[i]$ must also have sufficient entropy and can be considered as really random. Therefore a new block – a low entropy detector – monitors the entropy level of the digitized noise signal $s[i]$.

If the entropy decreases under a certain level, the notification is generated and sent to other blocks and the master system. The certain level of the sufficient entropy – a low entropy threshold – is determined on the basis of results of the successfully passed statistical test suites.

The proposed EGA allows using a reconfigurable noise source, which is able to respond to the low entropy notification and change its configuration. So the noise source in the new configuration generates the analog noise signal $v(t)$ under different conditions, which means higher power consumption and lower random data rate but also entropy increase.

5.2 Protective Mechanisms

Currently, modern communication and cryptographic systems face attack attempts every day, which threatens the safety of all electronic equipment and causes problems in everyday life. A part of the attacks is directed to TRNGs. Therefore the described EGA contains protective mechanisms, which are able to detect the deliberate malicious attacks and generator malfunctions characterized by the entropy decrease of produced sequences of random numbers.

5.2.1 Attack Detector

The attack detector is based on a model of the attack when an offensive signal $a_{\mathrm{a}}(t)$ is superimposed to the analog noise signal $v(t)$. This attack model is depicted in figure 5.2. In normal conditions, the TRNG produces random values with a specific probability distribution, but which is changed during the attack. In other words, the change of the probability distribution of the analog noise signal $v(t)$ is caused by the offensive signal $a_{\mathrm{a}}(t)$. How-

ever, one condition must be met. The standard deviation of the noise source probability distribution must be smaller than an amplitude of the attack. This type of the attack may be considered the so-called side channel attack. After digitization, the attack appears as the bias b of the probability, with which random bits are generated and which is defined in section 2.1.3.

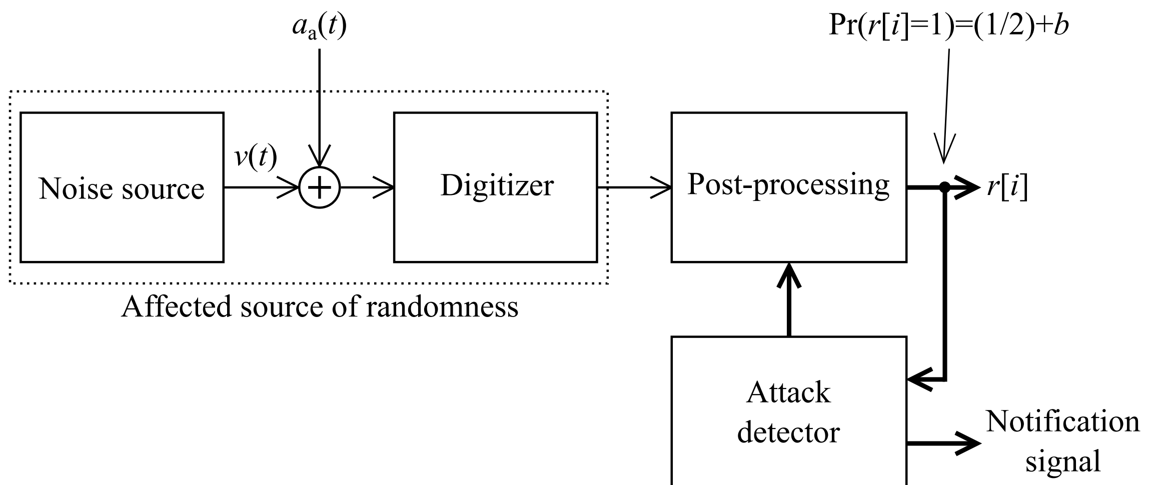


Fig. 5.2: The model of the side channel attack aimed at the noise source

To be possible to detect the potential attack, the attack detector monitors the bias occurring in the internal random numbers $r[i]$. A sudden increase of the bias causes the creation of the attack notification on the basis of which the generator can stop delivering random bits. Bias monitoring is based on a principle resulting from the fundamental purpose of the von Neumann corrector, whose function is described in section 2.1.3. If the internal random numbers $r[i]$ are generated without any bias b, the random data rate of the von Neumann corrector B_{VN} is exactly four times lower than the random data rate of the internal random numbers B_{IR}. However, if the random bits are generated with the bias, the random data rate B_{VN} decreases depending on the magnitude of the bias. So the ratio between B_{VN} and B_{IR} is a function of the bias b given by the formula

$$\frac{B_{\mathrm{VN}}}{B_{\mathrm{IR}}} = \frac{1}{4} - b^2, \tag{5.1}$$

which is derived in section 2.1.3. Relation between B_{VN} and B_{IR} depending on the bias b is shown in figure 5.3.

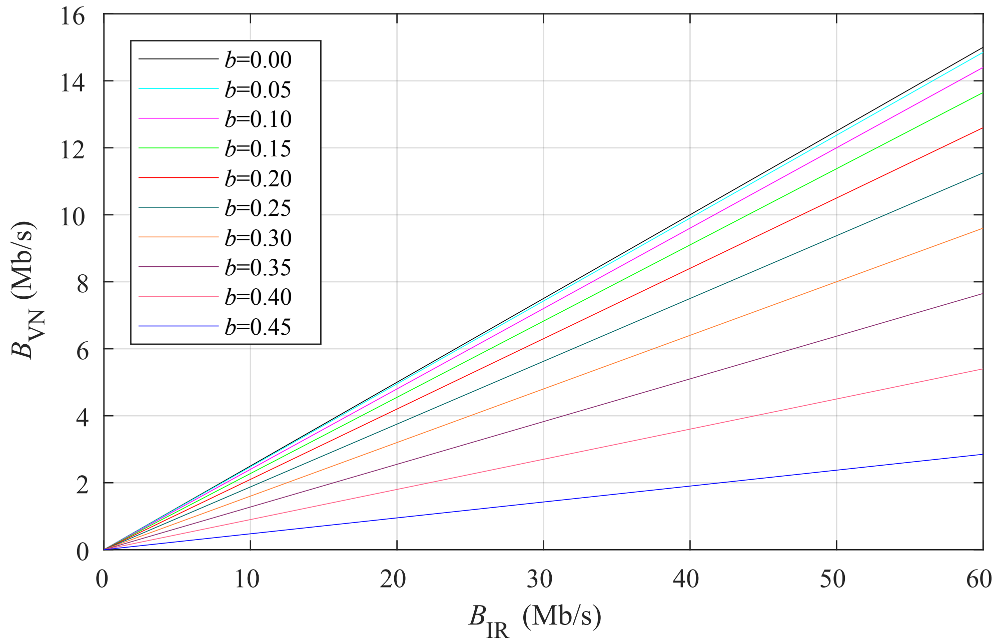


Fig. 5.3: Relation between B_{VN} and B_{IR} depending on the bias b

At this moment, it is necessary to determine an acceptable level of the bias and the bias, which is considered the possible attack. As mentioned above, limit values of the bias can be determined on the basis of results of the successfully passed statistical test suites. Thus the bias limit values $b_{\mathrm{l,l}}$ and $b_{\mathrm{l,h}}$ can be calculated using the probability when the post-processing block generates logic one assuming the entire sequence passes the statistical test suite. The Monobit test of the FIPS statistical test suite briefly described in section 3.2.1 defines the admissible numbers of logic ones in the random number sequences, which is in the range of $N_{\mathrm{lo,l}}$ to $N_{\mathrm{lo,h}}$. So the probability is in the range

$$\frac{N_{\mathrm{lo,l}}}{N_{\mathrm{FIPS}}} \leq \Pr\left(s\left[i\right] = 1 \,|\, \text{FIPS Monobit test passed}\right) \leq \frac{N_{\mathrm{lo,h}}}{N_{\mathrm{FIPS}}} \tag{5.2}$$

where N_{FIPS} is the required length of the random number sequence by the FIPS test suite. Using the definition of the bias (2.5) and the formula (5.2), the bias limits can be expressed as

$$b_{\mathrm{l,l}} = \frac{N_{\mathrm{lo,l}}}{N_{\mathrm{FIPS}}} - \frac{1}{2} \tag{5.3}$$

and

$$b_{\mathrm{l,h}} = \frac{N_{\mathrm{lo,h}}}{N_{\mathrm{FIPS}}} - \frac{1}{2}. \tag{5.4}$$

By insertion of the parameters defined in the FIPS test suite, the specific values of the bias limit are calculated. So the low bias limit $b_{\mathrm{l,l}}$ equals to -0.01375 and the high bias

limit $b_{l,h}$ equals to 0.01375. Assuming a fully functional and tested TRNG, if the absolute value of the bias suddenly rises above the calculated limits, the attack detector produces the notification.

The attack detector is realized as a counter of valid bits coming from the von Neumann corrector. The case, where this corrector generates a value, is marked as the valid bit. So the detector is composed of the standard counter and a very simple generator of the valid bits, which is shown in figure 5.4. An advantage of the solution described above versus the solution based on the definition of the bias is that this solution saves a considerable amount of logic gates. The solution based on the definition of the bias counts logic ones in the internal random numbers $r[i]$. The number of logic ones can be more than a half the length of the considered sequence. However, the solution described above counts only valid bits, which is slightly less than a quarter the length of the sequence. It also means that this solution saves an area on a chip, which is an important not only economic indicator. A disadvantage of this solution is a lower resolution of the bias, which is given by the factor b^2 in the equation 5.1.

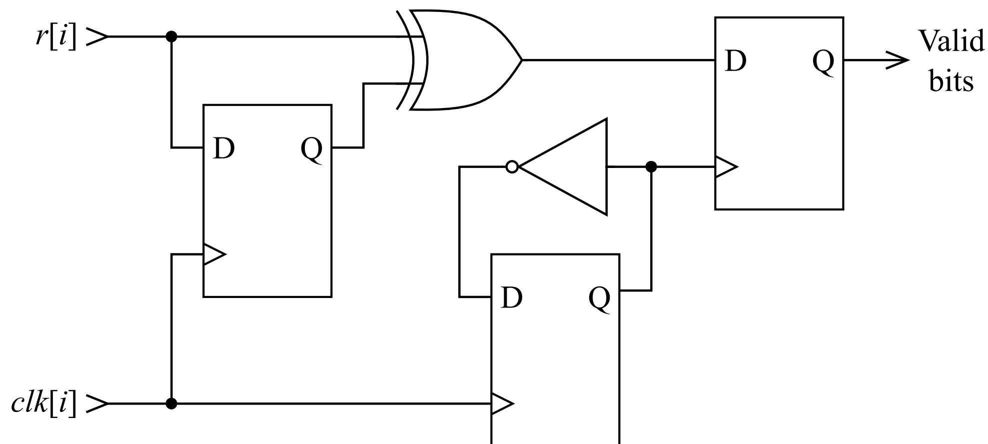


Fig. 5.4: The principle schematic diagram of the valid bit generator without buffers for proper timing

The above-described solution of the attack detector has been implemented as a block described in the Verilog HDL [125], which can be automatically compiled by Synopsys Design Compiler [126] into a CMOS gate logic. Then its physical design can be automatically realized using Synopsys IC Compiler [127], which ensures proper timing of all logic gates and their optimal placement and routing for a chosen CMOS technology.

5.2.2 Low Entropy Detector

The low entropy detector should determine the entropy of the random number sequences during their generation. However, it is not an easy task. The definition of the Shannon entropy given in 3.3 is based on the knowledge of the whole number sequence. Therefore it is not possible to determine the value of the entropy directly from the definition, but a way must be found how to estimate the entropy of the generated sequences.

The new block of the EGA – the low entropy detector – estimates the entropy of the sequences produced by the source of randomness. So the low entropy detector processes the digitized noise signal $s[i]$, which is the digital signal composed of logic ones and zeros. Under this assumption, an alphabet $\mathcal{A}$ is introduced while $\mathcal{A} = \{0, 1\}$. Then the probabilities are $\Pr(s[i] = 1) = p_1$ and $\Pr(s[i] = 0) = p_0$. At this moment, the mentioned probabilities are used to determine the entropy. So

$$H(p_1) = -\sum_{q \in \mathcal{A}} p_q \log_2 p_q = -p_1 \log_2 p_1 - p_0 \log_2 p_0 \tag{5.5}$$

and because $p_0 = 1 - p_1$. The entropy is

$$H(p_1) = -p_1 \log_2 p_1 - (1 - p_1) \log_2 (1 - p_1) . \tag{5.6}$$

As can be seen from the formula (5.6), it is not possible to determine the exact value of the entropy of random number sequences, which are being generated, because the probability p_1 is calculated from the entire sequence. Moreover, the probability can vary during its generation. Therefore, in this case, an approximation of the entropy is determined from consecutive sub-sequences with a fixed length. The sub-sequence is defined as $g[v] = s\left[m_g i + v\right]$ where $0 \leq v \leq m_g - 1$ and m_g is the length of the sub-sequence.

According to the formula (5.6), periodically repeated sequences of ones and zeros have the maximal value of the entropy. However, generation of these sequences by a TRNG is impermissible. Thus the approximate entropy of sub-sequences must be computed more complexly. For this reason, a method of the approximately entropy estimation published in [128] is adopted. First, the term binary derivative of the binary sub-sequence $d[v]$ is defined. So this term published in [129] is described by

$$d[v] = \begin{cases} 1, & g[v] = g[v+1], \\ 0, & g[v] \neq g[v+1] . \end{cases} \tag{5.7}$$

Marking $d^O[v]$ represents an application of $d[v]$ O times. In other words, O is order of the binary derivative. By the application of the binary derivative, it is possible to detect periodically recurring patterns in the sub-sequence $g[v]$.

Using the formula (5.6), the entropy of each binary derivative of the binary sub-sequence $g[v]$ can be computed and marked as $H(p_{1,O})$ where $p_{1,O}$ is the probability given by $p_{1,O} = \Pr\left(d^O[v] = 1\right)$. For estimation of the approximate entropy H_{est}, a mechanism using the weighting method was especially developed and presented in [128]. Thus the approximate entropy estimation H_{est} can be calculated as

$$H_{\mathrm{est}} = \frac{1}{\sum_{q=0}^{m_g-1} w[q]} \sum_{q=0}^{N_{\mathrm{BD}}-1} H\left(p_{1,q}\right) w[q] \tag{5.8}$$

where $w[k]$ is a weighting function.

For application in the TRNGs, the suitable weighting function is

$$w[q] = 2^q. \tag{5.9}$$

This method is usually marked as a power weighting. Every order of the derivative has own weight, and every binary derivative has a specific value of the Shannon entropy according to the formula (5.6). A higher order of the binary derivative has a higher weight. Therefore it is possible to eliminate sub-sequences with the periodic pattern.

The highest binary derivative is not used for computing the approximate entropy estimation H_{est} because its contribution is not essential. Therefore the first $m_g - 2$ binary derivatives of the sub-sequence $g[v]$ are used for computation of a weighted average of the entropy as

$$H_{\mathrm{est}} = \frac{1}{\sum_{q=0}^{m_g-2} 2^q} \sum_{q=0}^{m_g-2} H\left(p_{1,q}\right) 2^q \tag{5.10}$$

where $p_{1,q}$ is the probability and $p_{1,q} = \Pr\left(d^q[v] = 1\right)$.

The formula suitable for computation of the approximate entropy estimation H_{est} is obtained by substituting the equation (5.6) into the equation (5.10). Then the partial sum of the geometrical series is applied and the approximate entropy estimation H_{est} of the binary sub-sequence is

$$H_{\mathrm{est}} = \frac{1}{2^{m_g-1} - 1} \sum_{q=0}^{m_g-2} \left(-p_{1,q} - \log_2 p_{1,q} - \left(1 - p_{1,q}\right) \log_2 \left(1 - p_{1,q}\right)\right) 2^q. \tag{5.11}$$

As written above, the thresholding method is used in the low entropy detector. Therefore a threshold of the approximate entropy estimation of the binary sub-sequence is set.

On this basis, the low entropy detector decides whether the sub-sequence is valid or, conversely, is marked as the sub-sequence with the low entropy. The disadvantage of this method of the approximate entropy estimation computation is large time requirements.

Online calculation of the approximate entropy estimation is not a trivial task. The suitable length of the sub-sequences m_g can be determined on the basis of properties of the statistical test suites described in chapter 3 and possibilities to evaluate the sub-sequence in real time. The statistical test suite define so-called runs, which are uninterrupted sequences of identical bits, and measure their occurrence. For this reason, it is suitable to use 8-bit or eventually 4-bit sub-sequences. For these sub-sequences, a preferable solution is to compute their approximate entropy estimations and choose the sub-sequences with low values, which will be detected during the operation of the TRNG. The sub-sequences with the low values of the approximate entropy estimation are listed in tables 5.1 and 5.2. The low entropy detector has been implemented as a block described in the Verilog HDL.

Sub-sequence	H_{est}
0000	0.000000
1111	0.000000
0101	0.142857
1010	0.142857

Tab. 5.1: The 4-bit sub-sequences with the low values of the approximate entropy estimation

The low entropy detector creates a control signal for the noise source, which is reconfigured so that the random number sequence does not contain more consecutive sub-sequences with the low entropy. A number of the sub-sequences with the low-entropy results from the so-called long run defined by the FIPS test suite and described in 3.2.4. Thus, in other words, the sequence including at least 26 identical consecutive bits should not be produced. From this statement, the primary function of the low entropy detector is derived. Thus the noise source is reconfigured when the low entropy detector detects three equal consecutive 8-bit sub-sequences with low entropy or six equal consecutive 4-bit sub-sequences with low entropy. These states are further referred to as low entropy runes. If this situation occurs, the noise source is very likely affected by deterministic

noise or even is locked in some output logic value. Therefore the noise source is reconfigured into the second functional state. By early reconfiguration, there is a chance that any corrupted sequence will not be produced and the TRNG will not fail.

Sub-sequence	H_{est}	Sub-sequence	H_{est}
00000000	0.000000	00001111	0.107277
11111111	0.000000	11110000	0.107277
01010101	0.007874	01011010	0.107277
10101010	0.007874	10100101	0.107277
00110011	0.023389	00111100	0.111551
01100110	0.023389	01101001	0.111551
10011001	0.023389	10010110	0.111551
11001100	0.023389	11000011	0.111551
00010001	0.053399	00011110	0.114125
00100010	0.053399	00101101	0.114125
01000100	0.053399	01001011	0.114125
01110111	0.053399	01111000	0.114125
10001000	0.053399	10000111	0.114125
10111011	0.053399	10110100	0.114125
11011101	0.053399	11010010	0.114125
11101110	0.053399	11100001	0.114125

Tab. 5.2: The 8-bit sub-sequences with the low values of the approximate entropy estimation

5.3 Reconfigurable Source of Randomness

The proposed EGA allows utilization of a reconfigurable source of randomness, which consists of a reconfigurable noise source and an appropriate digitizer. Therefore an example of the solution is presented in this section. This solution is convenient for ICs. Circuits are proposed in the 130 nm bulk CMOS technology known as HCMOS9A from STMi-

croelectronics. This technology is not only suitable for digital systems but also analog and AMS systems.

5.3.1 Reconfigurable Noise Source

Randomness is extracted from a CMOS circuit exhibiting the metastable behavior, which is described in section 2.4. This circuit can generate individual random bits directly in each period and is derived from the noise source described in section 4.2.1. Therefore it is proposed based on the methodology shown in chapter 4. Immunity against deterministic disturbance of a power supply, which can affect the quality of random number sequences, is increased by implemented circuit topologies such as cascode connections of transistors.

The proposed noise source is capable of generating sequences of random bits in two settings. If randomness in the first setting of the noise source is lost, the system mentioned above switches the circuit to the second setting. There is a chance that the noise source could again start producing the random sequences. This approach could be called as a method of dual protection.

The analog noise signal is generated by the reconfigurable noise source containing the metastable element. The schematic diagram of this noise source is depicted in figure 5.5. The noise source is composed of the current mirrors, the internal voltage generator, two amplifiers, and the metastable element. The current mirror created from the N-channel MOSFETs M_{N12} and M_{N13} copies the fine reference current I_{REF} of the value of $1\,\mu A$ into the circuit. The current mirrors composed of P-channel MOSFETs M_{P3} – M_{P15} serve as the current sources for the internal voltage generator, both amplifier, and the metastable element. They contain the cascode transistors due to reducing the influence of deterministic power supply distortion.

The metastable element works as a fast comparator with inputs connected to the voltage V_{SET}. This voltage is generated in the simple internal voltage generator, which is composed of one branch of the P-channel MOSFET current mirror and N-channel MOSFETs M_{N8} and M_{N9}. In the default setting, the cross-coupled N-channel MOSFETs M_{N1} and M_{N2} create a positive feedback during the decision phase while the transistors M_{N3} and M_{N4} are disabled by switches SW_1 and SW_2. In figure 5.6, the switches are drawn as symbols, but in reality, they are made up of MOSFETs. The N-channel MOSFETs M_{N7}, M_{N8}, and M_{N9} reset the proposed noise source and keep it in the well-defined state when the

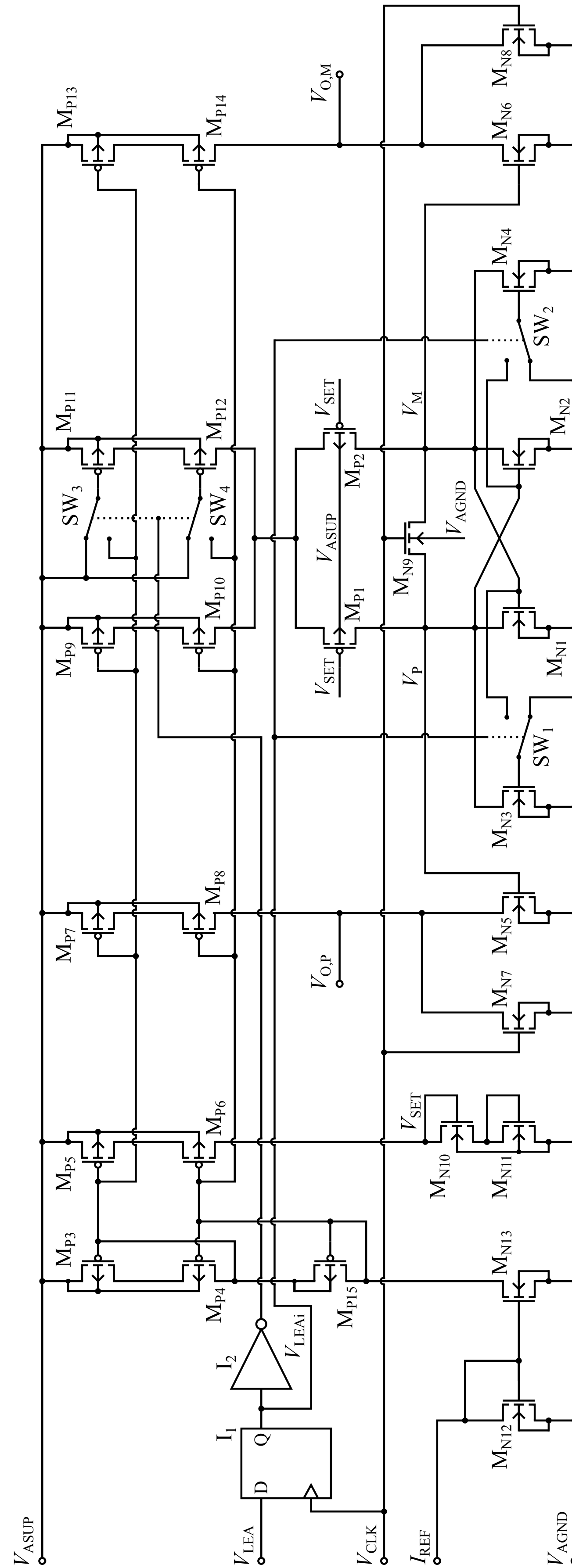


Fig. 5.5: The simplified schematic diagram of the reconfigurable noise source

clock signal V_{CLK} is in logic one. At this moment, the circuit is in the reset phase. When the clock signal goes into logic zero, the circuit goes into the decision phase. The voltage in the nodes V_{P} and V_{M} goes from the metastable voltage V_{meta} of 985 mV to one of the stable voltages as can be seen in figure 5.6. The final stable voltage value is given by the random noise present in the metastable element during the decision phase. As well as the noise source proposed in section 4.2.1, this noise source extracts randomness from thermal and flicker noise, which appear in CMOS circuits and are described in section 2.6.1 respectively in section 2.6.2.

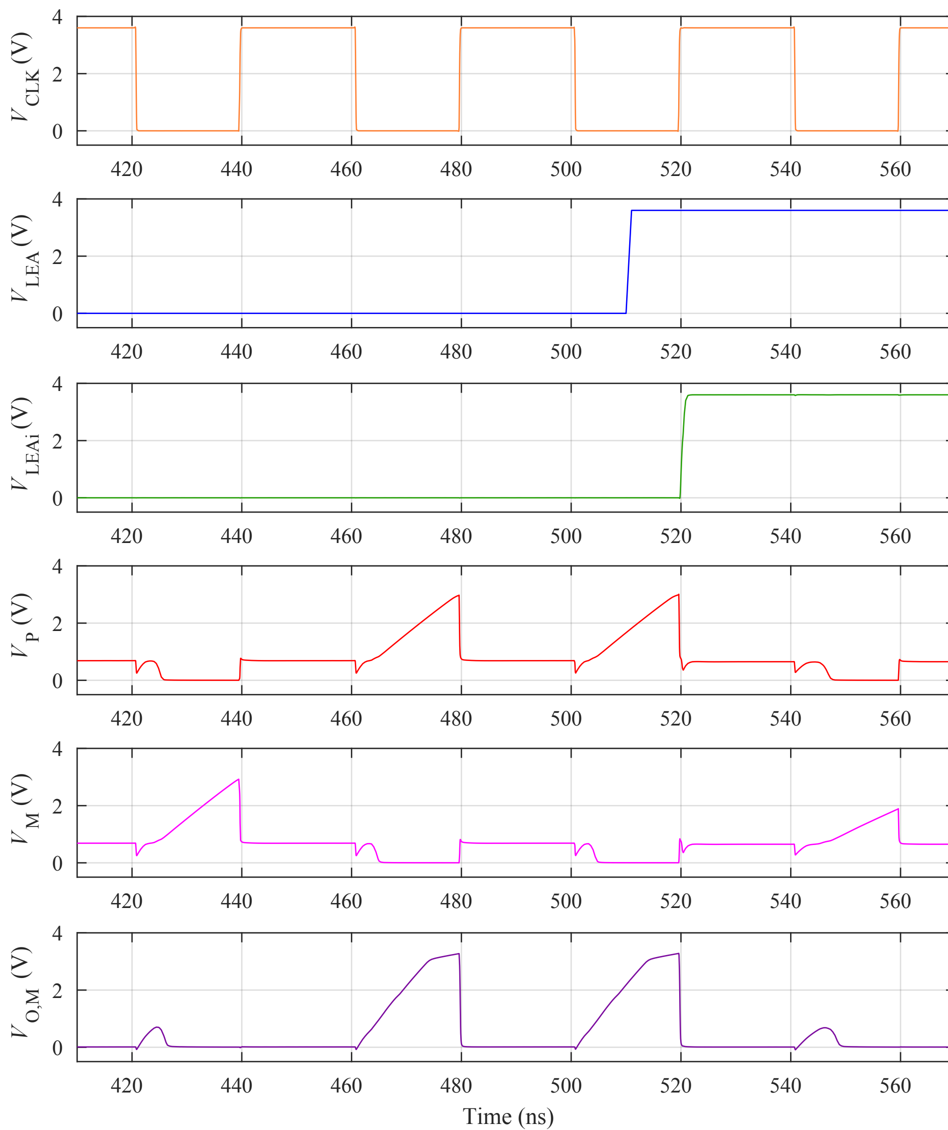


Fig. 5.6: Waveforms of signals inside the reconfigurable noise source simulated by the Mentor Eldo simulator at the transistor level

Any asymmetry of the metastability element can cause a systematic error during random bit generation. Thus the random number sequences could be biased, and their quality would be low. Therefore the metastable element is proposed symmetrically. In other words, not only all active components of the metastable element are designed symmetrically, but also parasitic capacitances and resistances on both sides of the element must be the same. As well as the case of the TRNG described in the previous chapter, the high quality of physical design of this circuit is essential. Therefore it is necessary to make it manually with the utmost caution. Any deviation from the symmetry of circuit layout can cause some bias in the produced random number sequences. A symmetrical layout of the reconfigurable noise source is shown in figure 5.7.

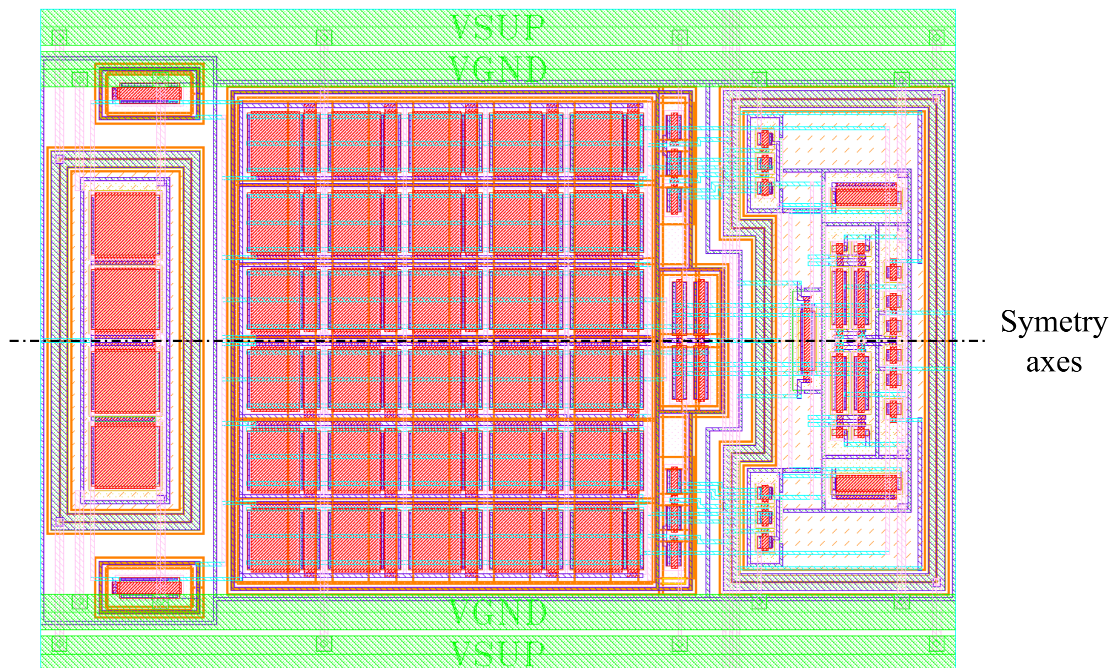


Fig. 5.7: The layout of the reconfigurable noise source with a drawn symmetry axis

The single-stage amplifiers created from N-channel MOSFETs M_{N5} and M_{N6} and two branches of the P-channel MOSFET current mirror are incorporated into the circuit of the noise source to shape and create the analog noise signals, which are represented by the voltages $V_{O,M}$ and $V_{O,P}$ and should be complementary under normal conditions. They also cause slightly faster reaction of digitizer input circuits. Waveforms of signals inside the reconfigurable noise source depicted in figure 5.6 were created by the Mentor Eldo simulator at the transistor level when noise transient simulations were performed.

Into the second functional setting, the reconfigurable noise source is set by the control signal, which can be marked as a low entropy alert and is produced by the low entropy detector based on the conditions defined above. The circuit is designed based on the assumption that the low entropy alert represented by the voltage V_{LEA} can occur at any time. Therefore the low entropy alert is synchronized with the clock signal V_{CLK} using a D-type flip-flop I_1 in order circuit parameters are not changed during the decision phase when a random bit is generated. So the circuit is reconfigured during the reset phase. As can be seen in figure 5.6, the coming alert V_{LEA} is synchronized and a generated internal alert signal V_{LEAi} switches the circuits parameters using the switches SW_1 – SW_4. During reconfigurable noise source design, the low entropy alert was forced into the circuit to be possible to test both functional settings.

MOSFETs	Number of elements (-)	Width (μm)	Length (μm)
M_{N1}, M_{N2}	2	4.5	0.5
M_{N3}, M_{N4}	2	2	0.5
M_{N5}, M_{N6}	2	1	0.5
M_{N7}, M_{N8}	1	1	0.5
M_{N9}	1	5	0.5
M_{N10}, M_{N11}	1	1	5
M_{N12}, M_{N13}	2	5	5
M_{P1}, M_{P2}	2	4.5	0.5
M_{P3}, M_{P5}	1	5	4
M_{P4}, M_{P6}	1	5	1
M_{P7}, M_{P13}	6	5	4
M_{P8}, M_{P14}	6	5	1
M_{P9}	14	5	4
M_{P10}	14	5	1
M_{P11}	2	5	4
M_{P12}	2	5	1

Tab. 5.3: Dimensions and numbers of elements of MOSFETs used in the reconfigurable noise source

After creation of the low entropy alert, the switches SW_1 – SW_4 enable additional circuit structures, which change parameters of the noise source. Specifically, the switches SW_3 and SW_4 enable an additional branch of the P-channel MOSFET current mirror formed by M_{P11} and M_{P12}. By this step, the quiescent current flowing into the metastable element is increased by 14.3 %, namely from 14 µA to 16 µA. Furthermore, the total ratio of width to length W/L of transistors M_{N1} and M_{N2} creating the positive feedback is changed by switching SW_1 and SW_2 when parallel-connected transistors SW_3 and SW_4 are turned on. So the ratio is larger by 44.4 % while the total width of the parallel-connected transistor pairs SW_1 – SW_3 and SW_2 – SW_4 increases from 9 µm to 13 µm and their length remains unchanged. Dimensions and numbers of elements of all MOSFETs are listed in table 5.3.

Also in this setting, the noise source is able to generate random bit sequences as can be seen in section 5.4 but at the cost of higher power consumption. The reconfigurable noise source is designed to be able to operate at the clock frequency of 25 MHz and the supply voltage 3.6 V. The simulated power consumption of the noise source together with the digitizer is 203.6 µW in the first parameter setting. In the second parameter setting, the power consumption is higher and is equal to 222.7 µW. Both simulations of the power consumptions were performed at ambient temperature 25 °C.

5.3.2 Differential Digitizer

The presented differential digitizer shown in figure 5.8 transforms the analog noise signal into a digital form and exploits the basic feature of the proposed noise source mentioned above that the output signals of the noise source $V_{O,P}$ and $V_{O,M}$ depicted in figure 5.9 are complementary during the decision phase when a random bit is generated. In other words, in this phase, the signal $V_{O,P}$ randomly goes into one of the logic values and the other signal $V_{O,M}$ goes into the opposite logic value.

As in the previous chapter, the power supply and ground are divided into two separated parts and star-routed due to the prevention of mutual distortion. All sensitive parts of this source of randomness are connected to the quiet supply voltage V_{ASUP} respectively to the quiet ground voltage V_{AGND}. The digital part is supplied by the power domain formed by the supply voltage V_{DSUP} and the ground voltage V_{DGND}. Transmission of signals between both power domains is one of the digitizer tasks.

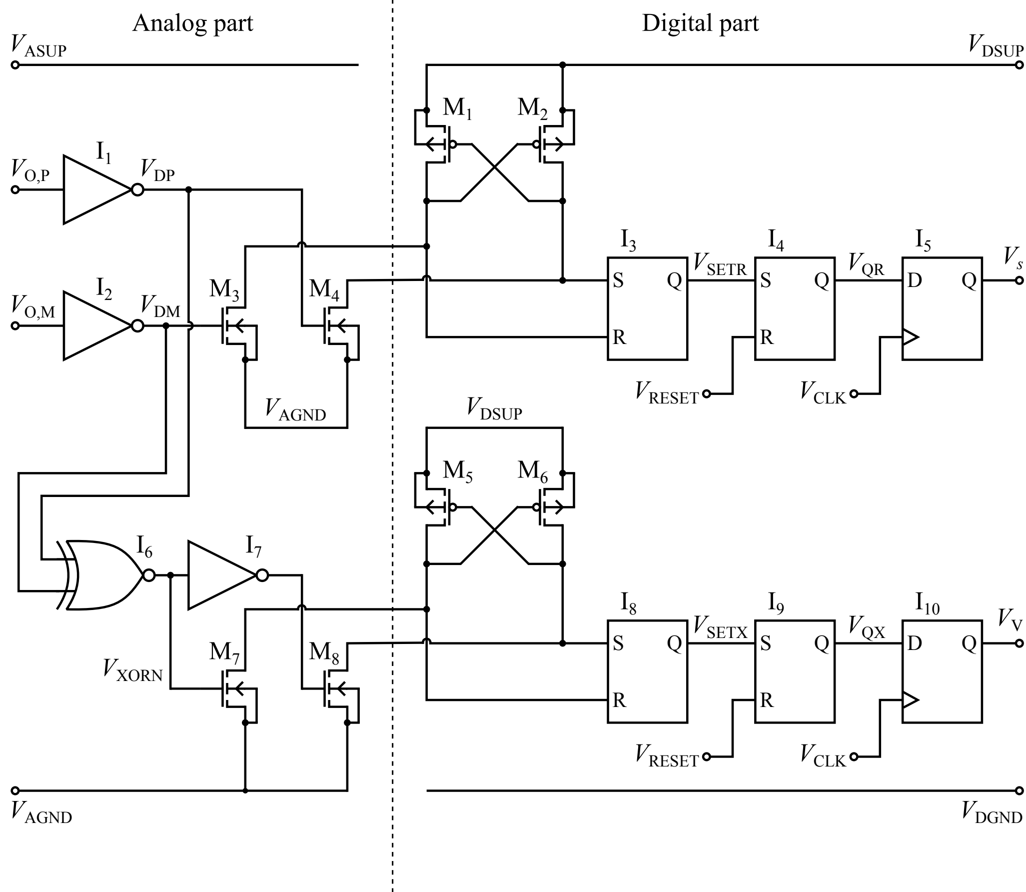


Fig. 5.8: The schematic diagram of the differential digitizer with the check of complementarity

Input inverters I_1 and I_2 shape the signals $V_{O,P}$ and $V_{O,M}$ and form the signals V_{DP} and V_{DM}, which are transferred by the circuit part composed of MOSFETs M_1, M_2, M_3, and M_4 from an analog power supply domain to a digital power supply domain. A RS latch I_3 created by CMOS NAND gates is incorporated to hold the transferred logic value during large fluctuations of the power supply when it could lead to erroneous transmission of the logic value. In this part, the signal V_{SETR} is available with suitable logic values in the digital power supply domain, but it is not synchronized with the clock signal V_{CLK} so that it can be further processed. Therefore another RS latch also composed of CMOS NAND gates produces the signal V_{QR}, whose duration of logic values are extended to be possible to be reliably sampled by a D flip-flop I_5. In this way, the output signal V_s is synchronized with the clock signal V_{CLK} and can be further processed in other digital parts of the TRNG.

The signal V_{RESET} resetting the RS latches I_4 and I_9 is generated together with the clock signal V_{CLK} in the master system.

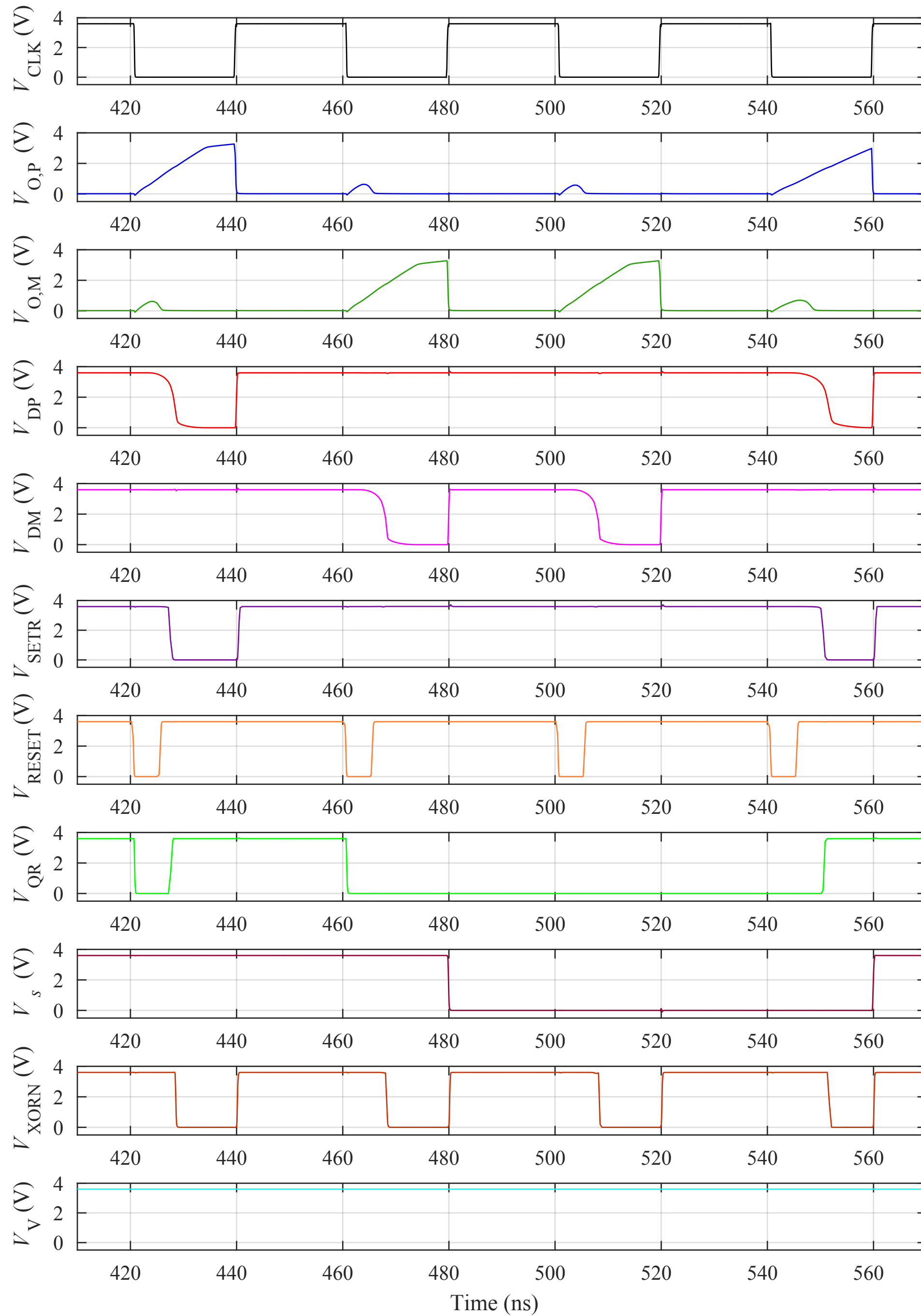


Fig. 5.9: Waveforms of signals inside the differential digitizer simulated by the Mentor Eldo simulator at the transistor level.

This digitizer is also able to check the functionality of the proposed noise source. So the assumption is that a correctly generated bit is formed from the signals $V_{\mathrm{O,P}}$ and

$V_{O,M}$, which are complementary during the decision phase under standard conditions. This complementarity is checked by an additional part of the introduced digitizer. This part is composed of a CMOS XOR gate at the input. The resulting value is then transferred into the digital power supply domain in the same way as described above. If the random bits are generated correctly, the signal checking validity V_V is in the logic one. However, if the TRNG was attacked, a situation could occur that the signals $V_{O,P}$ and $V_{O,M}$ would not be complementary. Thus the signal V_V would indicate invalid random bits generation and would be in the logic zero.

The inputs of the differential digitizer are connected directly to the most sensitive part of the TRNG – the noise source. Therefore the physical design of the noise source together with the digitizer inputs needs to be absolutely symmetrical. Any asymmetry can introduce the bias into random number sequences or even cause failure of the TRNG. So the physical design has been done manually using the methods described in chapter 7, which allow creating high-quality layouts of AMS circuits in a short time. Also all passive parasitic components such as parasitic resistors and capacitors must be minimized in order the circuit parameter would be minimally affected.

5.3.3 Power Supply

The proposed source of randomness has been developed with the power supply voltage of 3.6 V. Generally, it is suitable to minimize deterministic noises in the power supply, which might appear as a deterministic component in random number sequences. In extreme cases, this possibility might be abused, and little resistant TRNGs might be deliberately affected due to an intentional security breach of the system. Therefore, to minimize attack potential, a capacitor-less low dropout voltage regulator (LDO) with a high power supply rejection ratio and with no accessible external pin should be used to supply systems containing TRNGs.

For the power supply of the presented source of randomness, a capacitor-less LDO has been proposed and published in [130]. This LDO with the output power P-channel MOSFET is designed in the same CMOS technology, does not require any external component, is stable in a wide range of load currents, has fast transient response performance and low current consumption. Therefore it can be integrated with the presented circuit in one chip together. Also, for the power supply of this circuit, another variant of a LDO

with the output power N-channel MOSFET [131] has been developed and can be used. This variant provides the similar parameters and even higher load current but at the cost of higher power consumption of a control part.

5.4 Achieved Results

The designed blocks have been tested both separately and also together as a system. The reconfigurable source of randomness designed in the 130 nm HCMOS9A bulk CMOS technology has been simulated. The quality of generated random number sequences has been tested by FIPS and NIST test suites described in chapter 3. These statistical tests can find any discrepancies in random number sequences such as the bias or periodically repeating patterns. For evaluation of the proposed circuit, random number sequences with the length of 1 Mb were generated for both configurations of the noise source mentioned above. The noise transient simulations lasted an enormous time. Therefore it is not possible to generate a large number of random number sequences. The produced sequences were processed by commonly used correctors – the XOR corrector and the von Neumann corrector, whose principles and features are described in section 2.1.3. At first, all generated and processed sequences of random numbers were tested by the FIPS test suite, which did not detect any bias or repeating patterns. Thus all sequences passed this test suite and results for the random number sequence generated at the ambient temperature of 25 °C and the clock frequency of 25 MHz are listed in table 5.5. The proposed source of randomness can generate random bit during each period of the clock signal. Thus if the clock frequency is set to 25 MHz, the data rate of the digitized noise signal is 25 Mb/s.

The same random number sequences were also tested by the strict NIST test suite, whose results are listed in table 5.4. The results are in the form $N_\mathrm{P}/N_\mathrm{F}$ where N_P is the number of the passed sub-tests and N_F is the number of the failed ones. In the case of sequences generated by the noise source in the default setting and not processed by any corrector, the Non-overlapping template test fails in only one subtest out of 148. The sequences processed by the XOR corrector and the von Neumann corrector passed all test from the NIST test suite, which is a very good result. In the other case, when the noise source was switched to the non-default setting, the random number sequences generated directly without any corrector also passed all test except the Non-overlapping

template test, which fails in three subtest. The results of the sequences processed by the XOR corrector show that the negligible distortion in the form of repeating patterns was not completely eliminated. The Non-overlapping template test fails in only one subtest. However, the von Neumann corrector removed this distortion to a large extent as can be seen from the achieved results. Thus all NIST statistical tests passed. Based on the obtained results, the reconfigurable noise source shows very good properties. Subtest failures of the Non-overlapping template test in the above-mentioned cases do not indicate any major problems because the P-value always approached the decisive value.

	Default setting of reconfigurable noise source			Non-default setting of reconfigurable noise source		
	Directly	XOR	VN	Directly	XOR	VN
Monobit	1/0	1/0	1/0	1/0	1/0	1/0
Frequency within block	1/0	1/0	1/0	1/0	1/0	1/0
Runs	1/0	1/0	1/0	1/0	1/0	1/0
Longest runs	1/0	1/0	1/0	1/0	1/0	1/0
Binary matrix rank	1/0	1/0	1/0	1/0	1/0	1/0
Spectral DFT	1/0	1/0	1/0	1/0	1/0	1/0
Non-overlap. template	147/1	148/0	148/0	145/3	147/1	148/0
Overlapping template	1/0	1/0	1/0	1/0	1/0	1/0
Universal statistical	1/0	1/0	1/0	1/0	1/0	1/0
Linear complexity	1/0	1/0	1/0	1/0	1/0	1/0
Serial	2/0	2/0	2/0	2/0	2/0	2/0
Approximate entropy	1/0	1/0	1/0	1/0	1/0	1/0
Cumulative sums	2/0	2/0	2/0	2/0	2/0	2/0
Random excursions	8/0	8/0	8/0	8/0	8/0	8/0
Random excursions var.	18/0	18/0	18/0	18/0	18/0	18/0

Tab. 5.4: Results of the NIST tests for both settings of the reconfigurable noise source which have been generated directly (Directly), using the XOR corrector (XOR), or using the Von Neumann corrector (VN)

	Default setting of reconfigurable noise source			Non-default setting of reconfigurable noise source		
	Directly	XOR	VN	Directly	XOR	VN
Monobit	PASSED	PASSED	PASSED	PASSED	PASSED	PASSED
Poker	PASSED	PASSED	PASSED	PASSED	PASSED	PASSED
Runs	PASSED	PASSED	PASSED	PASSED	PASSED	PASSED
Long runs	PASSED	PASSED	PASSED	PASSED	PASSED	PASSED

Tab. 5.5: Results of the FIPS tests for both settings of the reconfigurable noise source which have been generated directly (Directly), using the XOR corrector (XOR), or using the Von Neumann corrector (VN)

Integration of any of above-mentioned correctors into the TRNG architecture improve the quality of random number sequences at the cost of a lower output data rate. Moreover, in the case of utilization of the von Neumann corrector, the output data rate is not constant but varies in time. It must be taken into account when designing the output interface. The presented circuit has been also simulated with deterministic disturbance of the power supply but no deterministic component has not appeared in the tested random number sequences.

The reconfigurable noise source has also been simulated together with the proposed low entropy detector, which has been tested in settings listed in table 5.6 where the used threshold value of the approximate entropy estimation $H_{\mathrm{est,th}}$ and the used length of the sub-sequence m_g are stated. The low entropy detector has not detected any low entropy run in random number sequences generated by the proposed noise source. Therefore, to verify functionality, random number sequences, into which the low entropy runs have been added, have been forced into the simulated system based on the EGA. Then the reconfigurable noise source has been switched to the non-default setting and back according to the proposed mechanism mentioned above.

A similar situation occurred when testing the attack detector. During simulations of the system based on the EGA, there has been no case when an attack has been detected. Therefore, to verify the functionality of the proposed attack detector, random number sequences with an introduced bias have been forced into the system. The attack detector

has responded as expected and announced the possible attack. This information has been available to the master system on the output terminal of the proposed TRNG.

Setting number	$H_{\text{est,th}}$	m_g
1	0.10	4
2	0.01	8
3	0.03	8
4	0.06	8
5	0.11	8
6	0.12	8

Tab. 5.6: Used settings of the proposed low entropy detector

As mentioned above, the power consumption of the reconfigurable noise source is 203.6 µW in the first parameter setting and 222.7 µW in the second parameter setting. Thus the required energy per random bit defined in section 4.3.4 is 8.14 pJ/b in the first parameter setting and 8.91 pJ/b in the second setting. When compared to values in table 4.6, the achieved values are very low, but they are slightly higher than the achieved values of the TRNG with time multiplexed sources of randomness described in chapter 4. It is due to the use of another CMOS technology, which is preferable for power supply systems and less suitable for signal processing applications.

The designed system has been verified by noise transient simulations in the Mentor Eldo simulator [14] and is ready for integration into a chip, which will be fabricated in the 130 nm HCMOS9A bulk CMOS process. This system can be directly compared with the TRNG published in [39], which produces the quality random number sequences and is designed in the similar 130 nm bulk CMOS process. However, it achieves a lower output data rate of 200 kb/s and do not contain any protective mechanisms.

5.5 Future Work on Development of TRNGs

The aim of new TRNGs is to produce a large amount of data in a short time but with the lowest possible power consumption so that they might be integrated into SoCs, which are parts of mobile hand-held applications. It is also necessary to ensure safety in order the

TRNGs cannot be attacked by deliberate malicious attacks. Therefore the development of these generators still continues.

5.5.1 EGA with Time Multiplexed Sources of Randomness

Future development of TRNGs is based on connection of the new introduced EGA with the TRNG described in chapter 4. In a future proposal, randomness will be extracted in four reconfigurable sources of randomness, which are described in section 5.3. Each digitized noise signal produced by sources of randomness will be checked by the low entropy detector mentioned in section 5.2.2. The low entropy detector will control the setting of the reconfigurable noise source and pass its status to the control logic. If the reconfigurable noise sources work properly, the produced digitized noise signals will be multiplexed as described in the previous chapter.

If the low entropy detectors detect corrupted random data produced by some of the reconfigurable sources of randomness, the affected sources will be turned off, and the internal random numbers will be generated by the functional ones. Therefore the control logic will drive the time multiplexer. Turning off some sources of randomness will mean a reduction of output random data rate. The produced internal random numbers will be checked by the attack detector, whose function is described in section 5.2.1. If any possible attack is detected, the detector will stop random number generation. The control logic will inform the master system about TRNG status. This idea is depicted in figure 5.10.

All parts of the future TRNG except the control logic have been proposed and tested by the functional sample or simulations. The main task will be a proposal of the control logic and testing of the whole system. In the first phase, the system will be simulated using behavioral models, which are described in chapter 6. Then the behavioral models will be replaced by designs at the transistor level. After simulations, the physical design of this proposal will be done using the methodology steps mentioned in chapter 7.

5.5.2 Noise source with Automatic Zeroing

A TRNG can be influenced by any mismatch among used components. Therefore future work is aimed at a proposal of the noise source, which will be able to reduce an influence of the mismatch actively. The main idea of this proposal is an introduction of a mechanism of automatic zeroing, which will compensate for the consequences of the mismatch. The

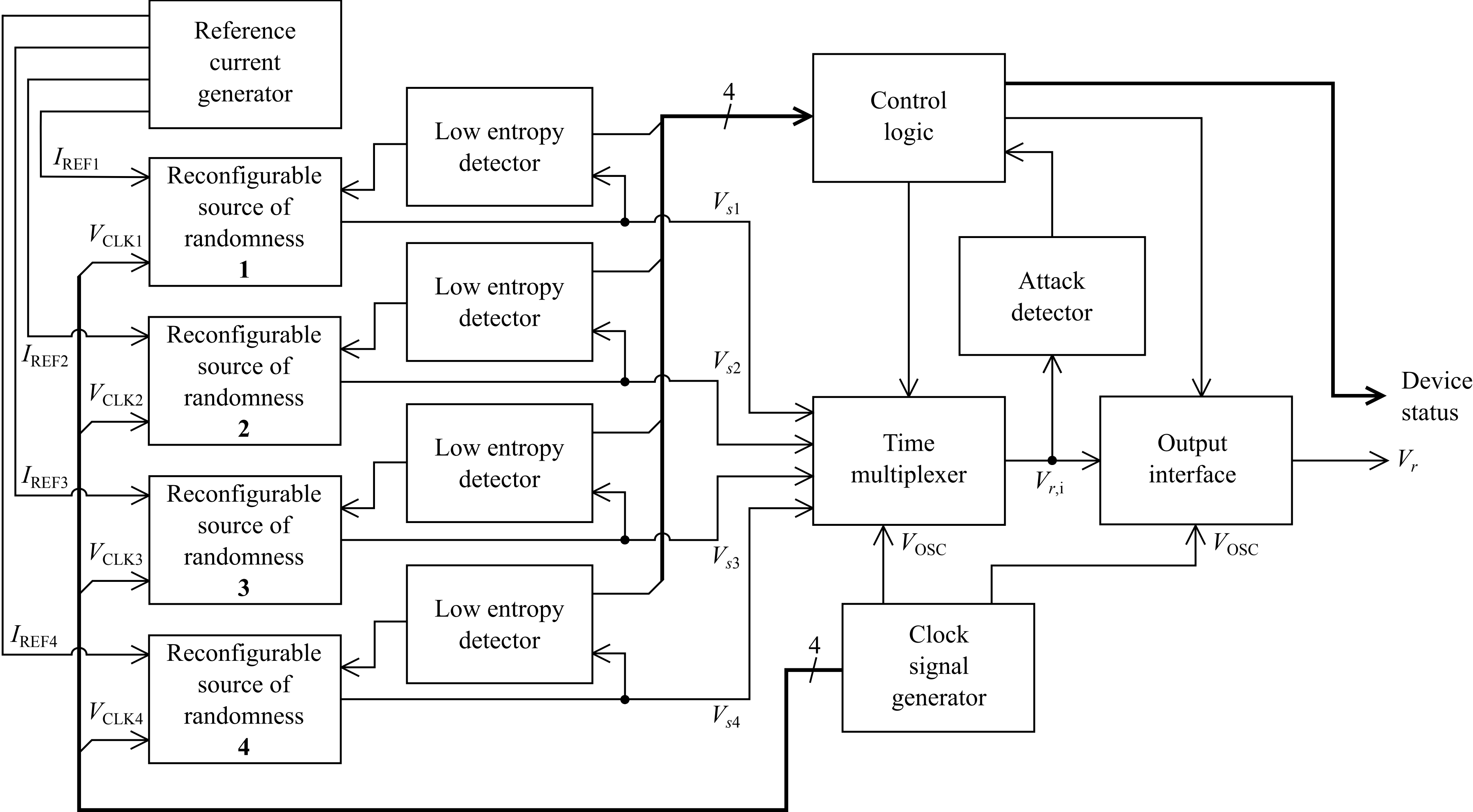


Fig. 5.10: The block diagram of the TRNG connecting the EGA with time multiplexed sources of randomness

new noise source with automatic zeroing will be derived from the noise sources mentioned above and also will work periodically. During the reset phase, the noise source will be reconfigured to be possible to measure the offset voltage and save it. Then, during the decision phase, the noise source will be reconfigured back to its default form and the saved offset voltage will be summed up with input voltages. This approach should actively eliminate the influence of the mismatch.

Chapter 6

Behavioral Models of TRNGs

Behavioral models of electronics systems are used for the architecture definition of these systems and then during final verification of the designed device. It is not possible to manufacture all versions of proposed systems because fabrication of hardware prototypes is very expensive and together with testing, long-lasting process. The number of fabricated versions and duration of development phase is reflected in a price of the finished product and its competitiveness in the market. To use the behavioral models allow greater degree of optimization, reducing the number of bugs and also the manufactured prototypes. Therefore, in the result, to use these models during development phase increases competitive of the product.

Systems working with digital signals are usually modeled by so-called hardware description language (HDL) such as VHDL (Very-high-speed IC HDL) or Verilog HDL [125]. An advantage of these models is that they can be directly synthesized by tools such as Synopsys Design Compiler [126] into a gate level netlist with which the physical design can be automatically compiled by Synopsys IC Compiler [127].

Analog systems working with continuous signals and AMS systems working with both signal types can be modeled by derivatives of the Verilog HDL such as Verilog-A or Verilog-AMS. These extensions allow solving differential equations describing the behavior of the analog systems. The TRNGs proposed in this thesis are typical representatives of AMS circuits extracting randomness from a physical phenomenon and transferring random information into the digital domain. For simulations of the created behavioral models, simulators such as Mentor Eldo [14] or Virtuoso Spectre Circuit Simuator [89] can be used.

Creation and following simulations of precise models are time-consuming. Therefore physical phenomena exploited in the systems are modeled only approximately. A compromise between the time consumption and accuracy of the behavioral model must be done [66]. In other words, the behavioral models approximate properties of the designed circuits at the transistor level.

TRNGs are individual parts of a SoC, which are usually fabricated in CMOS or BCD processes. Simulations of the whole complex SoC at the most accurate transistor level are almost impossible due to enormous time demands. Therefore, during simulations, individual parts of the SoC can be substituted by their behavioral models. This replacement significantly speeds up simulations of the whole system, which makes it easier to find system bugs.

The behavioral models of the designed TRNGs must exhibit the same properties as the designed TRNGs themselves. Their properties are evaluated based on produced random number sequences, which are tested by the statistical test suites FIPS [63] or NIST [64] described in chapter 3. Thus the random number sequences generated by the behavioral models should have very similar results of the statistical test suites to the random number sequences produced by the real TRNGs. The behavioral models of TRNGs have been published by the author of this work in [7] and [98].

6.1 Behavioral Model of TRNG with Time Multiplexer

The models of the AMS systems are usually described by system structure and individual modules, which are defined by mathematical relations among input and output signals. The relations can be created by events defined in the used Verilog-A HDL and allowing better control of the models. Then the system structure can be defined hierarchically on multiple levels, in which the modules are placed and interconnected. After the definition of the system structure and description of all modules, a set of equations describing the system is created using Kirchhoff's circuit laws and solved by a simulator to be obtained the system response. For simulations of the behavioral model of the TRNG with time multiplexed sources of randomness, both simulators mentioned above have been used.

This behavioral model of the TRNG described in chapter 4 has been analyzed by a transient analysis when the used simulator replaces nonlinear differential equations by

discrete-time finite difference approximations. Then, at each time point of a simulation time interval, the set of equations is iteratively solved by the Newton-Raphson method. The first convergence criterion is fulfilled when the solution of present iteration is close to the solution of the previous iteration. Similarly, the second criterion is met when the Kirchhoff's Flow law is satisfied. The iteration process terminates if both convergence criteria are satisfied [90].

6.1.1 Description of Model

The structure of the behavioral model of the TRNG with time multiplexed sources of randomness is derived from the block diagram shown in figure 4.2. Individual blocks are modeled as the modules by event-based approach described in Verilog-A HDL [90]. The fundamental module is the source of randomness composed of the noise source and the digitizer as stated in section 2.1. The noise source extract randomness from the metastable behavior of the metastable element, which is explained in section 2.4. The metastable element is a symmetric structure and therefore is modeled by two identical modules.

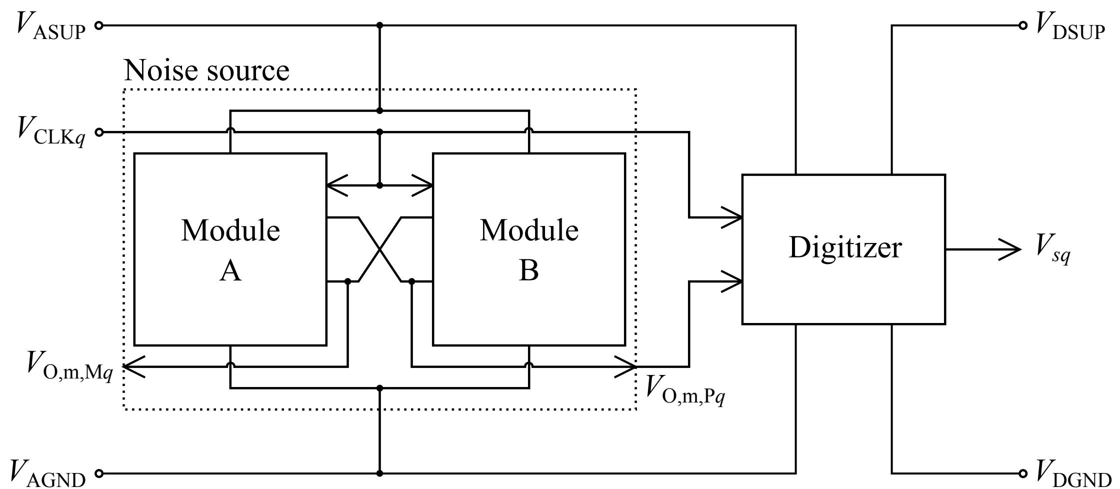


Fig. 6.1: Structure of the model of the source of randomness

Output voltages $V_{\mathrm{O,m,P}q}$ and $V_{\mathrm{O,m,M}q}$ are generated in several phases as described in section 4.2.1. In the first phase – so-called decision phase, the output voltage is equal to the metastable voltage V_{meta} and the metastable element is in the metastable state. In CMOS ICs, thermal noise and flicker noise briefly described in section 2.6 are significant. Noise in this behavioral model is generated by a PRNG, which is available in the

Verilog-A HDL [90]. The fundamental disadvantage of the used PRNG is that it generates the same number sequences with the same seed. For this reason, the only differences between both modules of the noise source are different seeds of the PRNG. Structure of the model of the source of randomness is shown in figure 6.1.

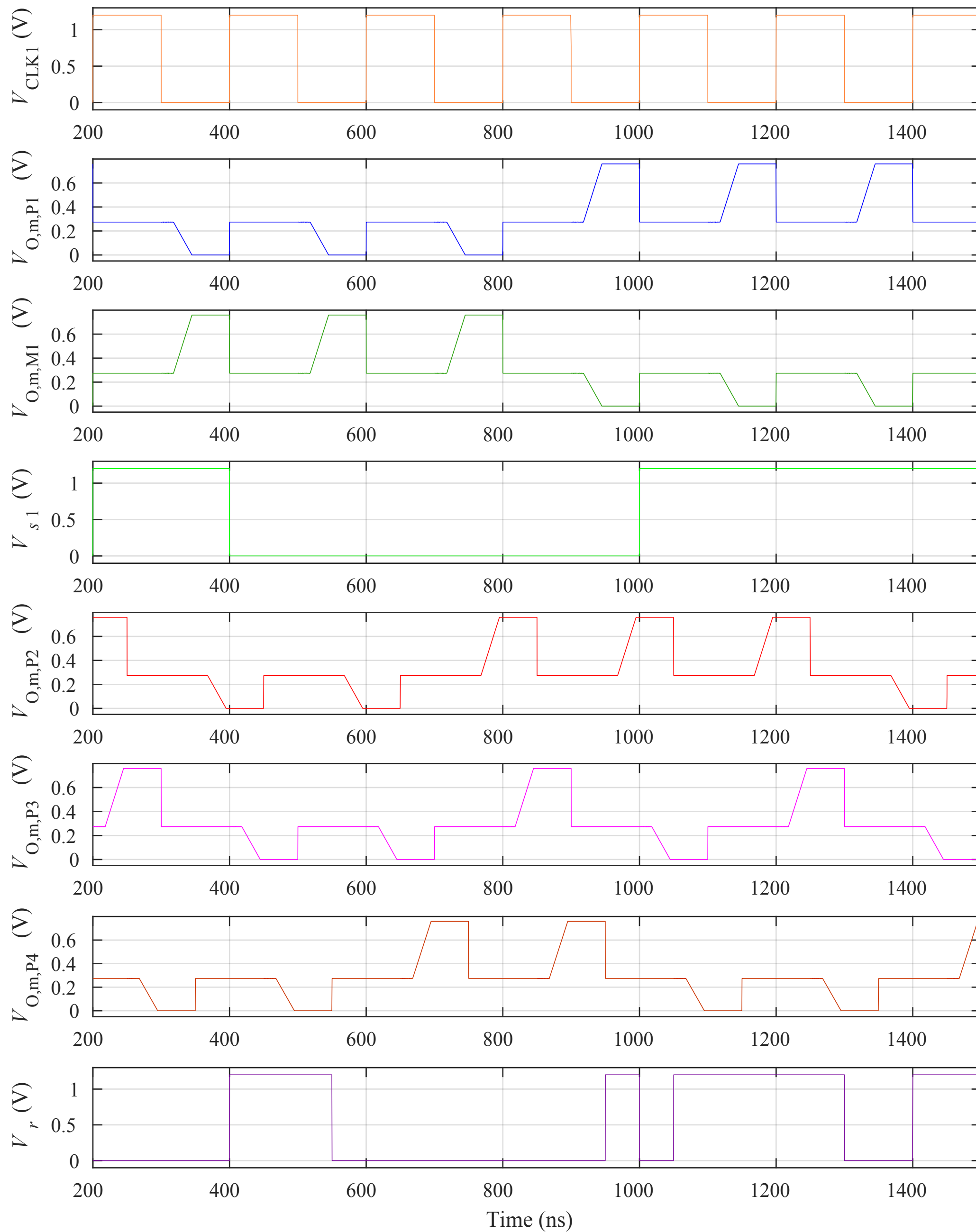


Fig. 6.2: Waveforms of signals generated by the behavioral model of the TRNG with time multiplexed sources of randomness simulated by transient analysis of the Mentor Eldo simulator

Thus transition to the stable state is caused by the noise present in the model. The final value of the stable state is given by the modeled noise, which is not based on any physical phenomena with random behavior but a result of the computational algorithm. This model has to work periodically. Therefore it is controlled by the clock signal. If its value is in logic one, the model is reset. In the opposite value, the model generates a random value according to the principle mentioned above. The modeled output signals of the noise source are shown in figure 6.2. This solution is designed to verify the system. However, it can not be used to obtain random number sequences for real applications because they are not truly random but only pseudo-random.

The second part of the source of randomness module is the digitizer, which transfers the analog noise signal to the digital form. Another task of the digitizer is a separation of power supply domains. Thus the produced digitized noise signal shown in figure 6.2 is composed of two logic levels in the digital domain and is synchronized with the relevant clock signal. The digitizer is modeled by an event called Cross Event, which is generated when the required expression crosses zero in a specified direction.

The behavioral model of this TRNG contains four sources of randomness. Their output signals are composed in the time multiplexer according to the principle defined in section 4.1. It allows generating random number sequences with four-time higher random data rate. This block is modeled by another basic event known as Time Event, which is produced at defined time points.

A part of the TRNG proposed in chapter 4 is the output interface creating the desired format of output random data to be possible to read them. In the case of this behavioral model, the output interface is a module, which saves generated random values into a text file so that the random number sequences can be further tested by the described statistical test suites. The name and location of the text file can be specified as parameters of the module. Moreover, the model of the output interface produces an output digital signal with defined voltage levels, which may be marked according to the definition mentioned in section 2.1 as the external random numbers. This signal can be further processed by the master system.

The created behavioral model also contains correctors described in section 2.1.3 to be possible to compare properties of produced sequences before and after processing. The correctors are able to some extent eliminate the bias present in generated random number

sequences. The module containing the correctors processes the internal random numbers and thus is incorporated between the time multiplexer and the output interface. Therefore the output interface also saves resulting sequences produced by the correctors into other text files.

6.1.2 Properties of Model

As mentioned above, tasks of the developed model are to approximate the behavior of the TRNG with time multiplexed sources of randomness and generate random number sequences with very similar properties as random number sequences produced by the fabricated TRNG. The properties are visible from the obtained results of the applied statistical test suites FIPS [63] and NIST [64]. Each test of these suites evaluates some property, which is expected from random number sequences. As described in chapter 3, the sequences with the length 1 Mb was tested by the NIST test suite and then they were shortened to 20 kb to be evaluated by the FIPS test suite.

Random data were generated during the transient simulation by Mentor Eldo simulator [14]. The output random data rate was set to the nominal value of 20 Mb/s according to the value, on which the fabricated TRNG is proposed. The results obtained using the statistical test suites are listed in tables 6.1 and 6.2 while the results of the NIST test suite are in the already used form $N_\mathrm{P}/N_\mathrm{F}$ where N_P is the number of the passed sub-tests and N_F is the number of the failed ones.

	Directly	XOR	VN
Monobit	PASSED	PASSED	PASSED
Poker	PASSED	PASSED	PASSED
Runs	PASSED	PASSED	PASSED
Long runs	PASSED	PASSED	PASSED

Tab. 6.1: Results of the FIPS tests of the behavioral model of the TRNG with time multiplexed sources of randomness. The sequences were generated directly (Directly), using the XOR corrector (XOR), or using the Von Neumann corrector (VN)

The evaluated sequences were generated both directly without any post-processing and with use of the correctors. The FIPS test suite did not detect any bias or repeating

patterns. As well as the more strict NIST test suite did not reveal any major shortcomings. During testing the sequence produced by the XOR corrector, the Non-overlapping template test failed in only one subtest out of 148. It is not an obstacle for the use of this behavioral model, which is able to produce number sequences having very similar properties as the sequences generated by the developed TRNG. As explained in the description of the model in previous section, this model is written in the Verilog-A HDL. Therefore it need not be tested in any temperature range because various temperature does not affect the created source code.

	Directly	XOR	VN
Monobit	1/0	1/0	1/0
Frequency within block	1/0	1/0	1/0
Runs	1/0	1/0	1/0
Longest runs	1/0	1/0	1/0
Binary matrix rank	1/0	1/0	1/0
Spectral DFT	1/0	1/0	1/0
Non-overlap. template	148/0	147/1	148/0
Overlapping template	1/0	1/0	1/0
Universal statistical	1/0	1/0	1/0
Linear complexity	1/0	1/0	1/0
Serial	2/0	2/0	2/0
Approximate entropy	1/0	1/0	1/0
Cumulative sums	2/0	2/0	2/0
Random excursions	8/0	8/0	8/0
Random excursions var.	18/0	18/0	18/0

Tab. 6.2: Results of the NIST tests of the behavioral model of the TRNG with time multiplexed sources of randomness

Simulations of the TRNG time multiplexed sources of randomness at the transistor level are very time-consuming. The very important task for the developed behavioral models is to accelerate simulations of SoCs containing TRNGs. Simulation of this model takes approximately 145 minutes for generation of number sequence with the length of

1 Mb. Simulation of the TRNG at the transistor level takes several days using a distributive mode of the simulator. Thus the duration of the simulation of the behavioral model is incomparably shorter than the duration of the transistor level simulation. Therefore, for verification of a SoC, it is necessary to use the behavioral model instead of the transistor level proposal.

6.2 Behavioral Model of EGA

The TRNG based on the enhanced generic architecture (EGA) introduced in chapter 5 is designed to be able to be a part of a SoC. Therefore its behavioral model was created in the Verilog-A HDL.

6.2.1 Structure of Model

The structure of this model is based on the block diagram shown in figure 5.1. The reconfigurable noise source is implemented as a module in the Verilog-A HDL, which is derived from the noise source module described in section 6.1.1. This derivation is possible because parameters of the model mentioned above are not so-called hard-coded, but they can be elegantly set as attributes of Verilog-A modules. Moreover, the module of the reconfigurable noise source is complemented by switching, which allows changing its setting. As defined in section 5.3, the change of reconfigurable noise source setting is performed when the entropy of produced random sequences decreases and there is a risk that generated sequences would not have passed the statistical test suites. The change of model setting affects the output signals $V_{\mathrm{O,m,P}}$ and $V_{\mathrm{O,m,M}}$ as can be seen in figure 6.3. It corresponds to the behavior of the proposed reconfigurable noise source, which is shown in figure 5.6.

Also a module of the differential digitizer is derived from the module of the digitizer described in section 6.1.1. This module is modified to be able to processed the complementary signals $V_{\mathrm{O,m,P}}$ and $V_{\mathrm{O,m,M}}$ produced by the reconfigurable noise source module according to the proposal in figure 5.8. Therefore this module is also extended by a part, which checks whether random bits are generated correctly. In other words, it monitors complementarity of the formed signals $V_{\mathrm{O,m,P}}$ and $V_{\mathrm{O,m,M}}$ during the phase, in which the reconfigurable noise source module creates a random bit.

Parts of this new architecture is the low entropy detector and the attack detector as can be seen in figure 5.1. They represent newly introduced protective mechanisms of TRNGs, which are capable of protecting communication systems against possible deliberate malicious attacks or can detect an unexpected drop in entropy of generated random number sequences. These blocks are designed in the Verilog HDL. Therefore they are used in the created behavioral model. The advantage is that these blocks were verified during the development of the TRNG based on the EGA.

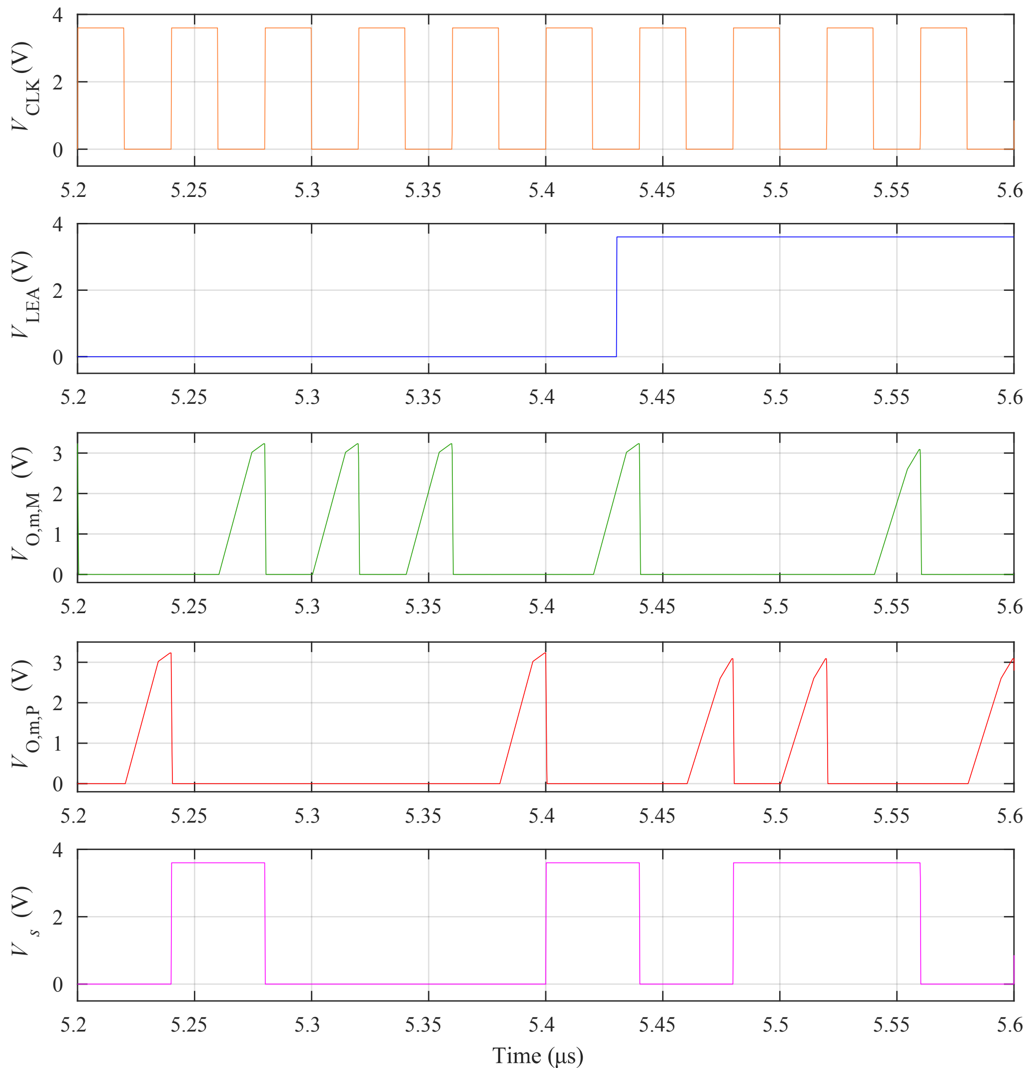


Fig. 6.3: Waveforms of signals generated by the behavioral model of the TRNG based on the EGA simulated by the Mentor Eldo simulator

The EGA allows implementation of random number sequence post-processing. Therefore its behavioral model contains correctors described in section 2.1.3 for the same reason

as mentioned in section 6.1.1. Equally the same module of the output interface creating for the model of the TRNG with time multiplexed sources of randomness is used in this behavioral model. Thus the generated and processed random values are saved into text files so that the random number sequences can be further tested by the described statistical test suites.

6.2.2 Evaluation of Model

The behavioral model of the TRNG based on the EGA was simulated by Mentor Eldo simulator [14]. The random number sequences were generated with the nominal value of the output random data rate equal to 25 Mb/s, on which the TRNG is proposed. The main module of this model – the source of randomness – is derived from the behavioral model described in section 6.1. Therefore the assumption is that this model should have very similar properties, which confirms the results of the statistical test suites.

	Default setting			Non-default setting		
	Directly	XOR	VN	Directly	XOR	VN
Monobit	PASSED	PASSED	PASSED	PASSED	PASSED	PASSED
Poker	PASSED	PASSED	PASSED	PASSED	PASSED	PASSED
Runs	PASSED	PASSED	PASSED	PASSED	PASSED	PASSED
Long runs	PASSED	PASSED	PASSED	PASSED	PASSED	PASSED

Tab. 6.3: Results of the FIPS tests for both settings of the behavioral model of the TRNG based on the EGA

Random number sequences were generated by this behavioral model in the first and second settings. The results of the FIPS statistical test suite are listed in table 6.3. This test suite did not reveal any non-random features of generated random number sequences. The results of the NIST statistical test suite are listed in table 6.4 where the form of writing the results is the same as in the previous case in section 6.1.2. As with the previous model, the NIST test suite failed only in some subtests of the Non-overlapping template test. However, their the P-values did not deviate fundamentally from the decisive value. Based on the results obtained, it can be said that this behavioral model has very similar properties as the proposed TRNG described in chapter 5.

	Default setting			Non-default setting		
	Directly	XOR	VN	Directly	XOR	VN
Monobit	1/0	1/0	1/0	1/0	1/0	1/0
Frequency within block	1/0	1/0	1/0	1/0	1/0	1/0
Runs	1/0	1/0	1/0	1/0	1/0	1/0
Longest runs	1/0	1/0	1/0	1/0	1/0	1/0
Binary matrix rank	1/0	1/0	1/0	1/0	1/0	1/0
Spectral DFT	1/0	1/0	1/0	1/0	1/0	1/0
Non-overlap. template	147/1	147/1	147/1	147/1	146/2	148/0
Overlapping template	1/0	1/0	1/0	1/0	1/0	1/0
Universal statistical	1/0	1/0	1/0	1/0	1/0	1/0
Linear complexity	1/0	1/0	1/0	1/0	1/0	1/0
Serial	2/0	2/0	2/0	2/0	2/0	2/0
Approximate entropy	1/0	1/0	1/0	1/0	1/0	1/0
Cumulative sums	2/0	2/0	2/0	2/0	2/0	2/0
Random excursions	8/0	8/0	8/0	8/0	8/0	8/0
Random excursions var.	18/0	18/0	18/0	18/0	18/0	18/0

Tab. 6.4: Results of the NIST tests for both settings of the behavioral model of the TRNG based on the EGA

The behavioral model has been created to be used in top-level simulations of a SoC because any top-level simulation with all blocks at the transistor level is extremely time-consuming. Therefore some blocks can be replaced by their behavioral models and the system can be verified in a shorter time. Simulation of this model based on the EGA takes approximately 35 minutes for generation of number sequence with the length of 1 Mb. As in the previous case, simulation of the relevant TRNG at the transistor level takes several days using a distributive mode of the simulator. Without this simulator setting, to produce number sequences with length of 1 Mb is enormously time-consuming.

Chapter 7

New methodology steps of physical design of AMS ICs

Design of semiconductor AMS ICs can be divided into several subsequent and mutually dependent steps. Out of those, a very crucial one is the physical implementation of designed circuit topologies, in other words, layout of ICs, which is in majority of companies usually done by a person other than the one working on microelectronic circuit design and its simulations. This work-partitioning is essential to achieve desired project timing.

The designed circuit topologies are usually not mature, not enough verified when the process of layout starts and as a deadline for sending lithographic data to a plant is getting closer, arising time pressure has to be dealt with. Thus the right timing of all the layout tasks is crucial from the beginning. Implementation of procedures and methods for saving time and preventing human failure can reasonably influence final time-to-market [132].

TRNGs developed in this work are typical representatives of AMS ICs. Moreover, they contain parts, which are very sensitive and their physical design has to be created with the utmost care. Any inaccuracy can cause the bias of generated random number sequences or even failure of the entire TRNG. Creating a precise physical design is a very time-consuming task. Therefore it is appropriate to use tools, which speeds up this development phase, helps to prevent errors, and makes the design more robust.

In practice, AMS physical design of all analog blocks and the whole ICs created by the Analog-on-Top approach is still handmade. No fully automated AMS physical design flow is accepted by physical design engineers because this has not achieved the quality of manually crafted physical designs, so far [72]. However, physical design engineers like

using assistant functions [73] published in [69], [70], [71] facilitating work on individual design steps. The assistant functions can save a considerable amount of valuable time, save human labour, and eliminate some types of errors such as non-compliance with the design rules, wrong metallic interconnections, or wrong current capability and too high resistance of metallic wires.

This chapter describes new features, which have been developed during the physical design of the TRNGs and are further used to create other AMS IC layouts. The newly introduced features are able to automatically sort electrical devices according to their topological, structural and electrical properties [8], control layout objects without filling forms [10], search an IC design database based on similarity of object properties [12], and classify matched structures regarding systematic mismatch [13]. The descriptions of these features can also be found in the publications [8], [10], [12], and [13] mentioned above.

7.1 Automated Pre-placement Phase

The most critical place to create fundamental errors passing through the entire physical design flow is the step before the placement itself when even hundreds of individual instances are sorted manually. These errors are usually detected only during the final verification, and their detection returns the physical design to the beginning. Presence of these errors leads to a wrong area estimation, and thus it can affect the surrounding blocks as well. Additionally, the finished placement must be changed in most cases as well as all follow-up phases. Works and tools published so far do not solve this problem. Therefore an algorithm automating this phase named a pre-placement phase has been proposed. The standard physical design flow enhanced by this phase ensuring primary sorting based on types of electrical devices and their electrical parameters is shown in figure 7.1. Using the automated algorithm, the pre-placement phase is accelerated, the entire physical design flow is faster, and the errors caused by manual work are eliminated. It also means that the cost of the development of devices such as TRNGs can be lowered. Results of the algorithm can be used for very fast and precise area estimation, manual placement as well as basic constraints for automatic placers. This section has been published by the author of this work in [8]. This publication contains a more detailed description of this algorithm.

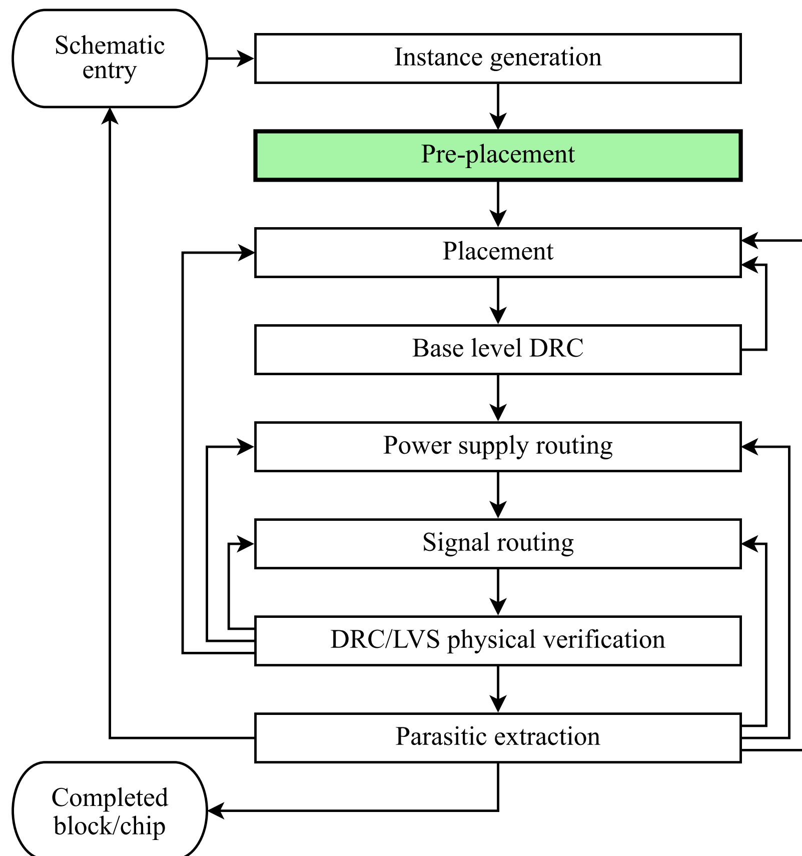


Fig. 7.1: Physical design flow enhanced by the pre-placement phase [8]

The pre-placement phase is governed by rules, which are derived from physical structures of devices used in a semiconductor technology and from usually used building blocks in IC design. Compliance with the rules is the fundamental prerequisite for flawless placement. Outputs of the proposed algorithm are groups of instances sorted according to the rules. Then the physical design engineers do not have to work with a large number of instances but they only work with a smaller number of defined groups which improves the clarity of the whole design and thus its quality. In a case of using an automatic analog placer, the proposed algorithm works as a generator of fundamental constraints without their definition is not possible to perform correct automatic placement [72].

7.1.1 Definition of Rules

At first, layout instances corresponding to a schematic diagram are generated onto a canvas of used layout editor [69]. Unfortunately, placement of these instances is usually based on their position in the schematic, which is quite useless for following layout cre-

ation. Different types of components are inappropriately mixed as can be seen in figure 7.2(a). At this time, components have to be sorted into groups of the same type and similar interconnection. Such groups can then be used in following layout phase – placement. Manual sorting and formation of the mentioned groups is a slow process with a high probability of an error requiring a lot of additional time to be fixed. Therefore the pre-placement phase is introduced. Groups of similar components are formed based on criteria, which take into account topology, structure and electrical interconnection of particular components, so that these groups are becoming matrix elements as shown in figure 7.2(b).

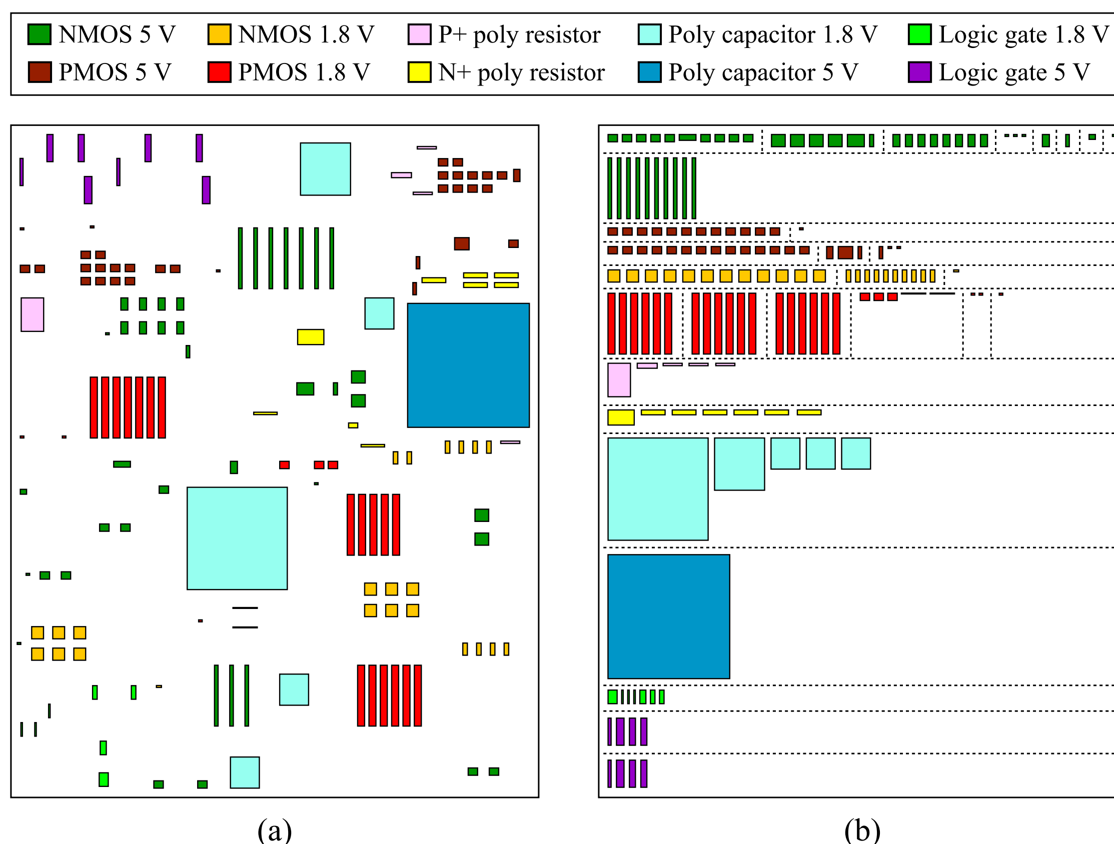


Fig. 7.2: Symbolic view of layout instances before (a) and after (b) the automated pre-placement phase [8]

Instances of the same types are located after the pre-placement phase in the same rows of the final pre-placement matrix (FPPM). The second important rule regarding MOSFETs is sorting according to bulk terminal connection, as can be seen in figure 7.3(a), where a cross section of N-channel MOSFET insulated by deep N-well is depicted and

bulk terminal is highlighted by a rectangle. MOSFETs with the same bulk connection can be placed into a common insulation well in order to save an area on a die. Therefore instances of the same type and with the same bulk terminal connection are placed in one row of FPPM.

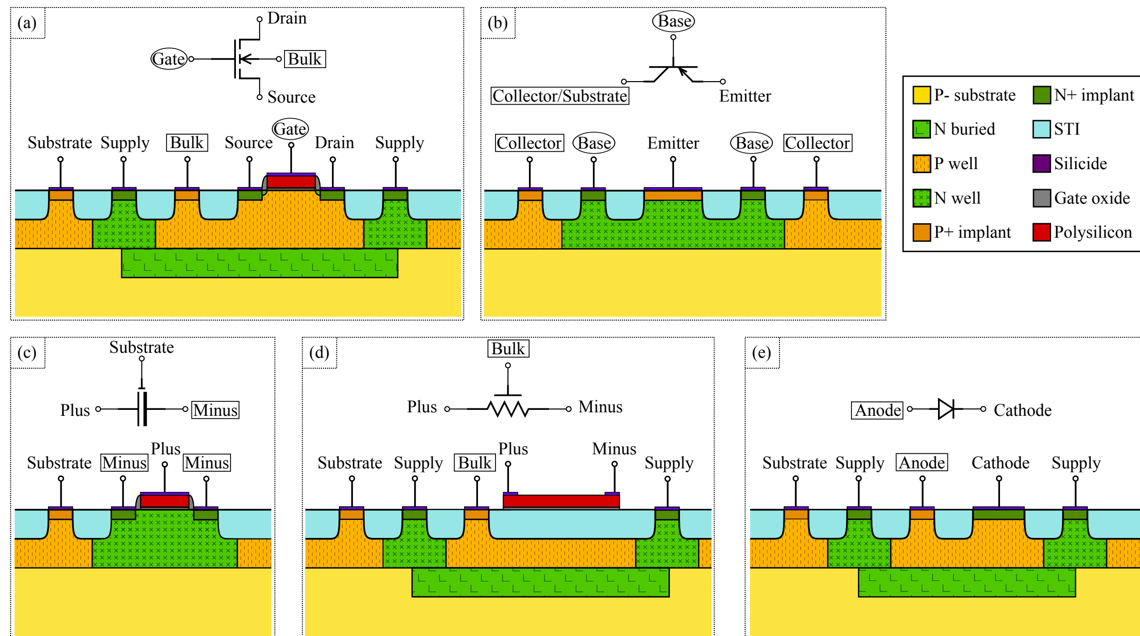


Fig. 7.3: Cross sections of usually used devices in CMOS and BCD designs: (a) Junction insulated N-channel MOSFET. (b) PNP BJT. (c) Polysilicon – N-well capacitor. (d) Polysilicon resistor with a junction insulated bulk terminal. (e) Junction insulated N+ – P-well diode [8]

Often occurring errors in MOSFET bulk connection, which are created during manual sorting, are eliminated using this rule. This type of errors is usually detected up in the final phase of design, when LVS check is performed. Their removal can be very complicated and time-consuming because instance placement inside block often has to be comprehensively changed while design rules and proper electrical connection have to be respected. Simultaneously, a suitable topology of designed block has to be preserved. When the above-mentioned type of error occurs, all phases from placement phase must be completed and verified again, as can be seen in figure 7.1.

The design time can still be optimized by introduction of a further rule regarding MOSFETs, which is derived from circuit connection of usually used building blocks such as current mirrors, differential pairs, active loads, switching stages and amplifiers. There-

fore MOSFETs are sorted according to electrical connection of the gate terminal, which is highlighted by an ellipse in figure 7.3(a). According to this rule, rows of FPPM are divided into columns while transistors with common gate connection are located in individual columns. An basic example is a current mirror where gate terminals of all MOSFETs are usually connected to the same net and bulk terminals are also connected together. Therefore in this case, all transistors are located in one cell of FPPM after the pre-placement phase. This feature makes detail layout of matched structures easier.

Failure of ICs manufactured in CMOS and BCD technologies is often caused by turning a parasitic thyristor structure on, which occurs mainly in the usually used bulk type of semiconductor wafers [133]. The parasitic thyristor known also as the parasitic silicon controlled rectifier (SCR) is a semiconductor structure composed of four alternating P-type and N-type layers with three P-N junctions as can be seen in figure 7.4(b) where is an example of one of the basic circuits – an inverter with schematic diagram shown in figure 7.4(a). A result of triggering this structure also shown in figure 7.4(c) is a creation of a low impedance path between power supply and ground domains. The so-called latch-up is created when low impedance path persists after removing the trigger source. Both parasitic bipolar junction transistors (BJTs) Q_P and Q_N in figure 7.4(c) have to be biased into the forward-active region [121]. In other words, a product of the common-emitter current gains β_F of Q_P and Q_N is equal or greater than one. These conditions for latch-up formation are usually met in standard ICs where P+ diffusions in N-wells are connected to the highest voltage and N+ diffusions in P-wells to the lowest voltage.

The parasitic thyristor structures are usually turned on by several mechanisms. So-called forward-voltage triggering occurs when a supply voltage exceeds the absolute maximum rating. Gate triggering arises when a N-well located close to the parasitic structure creates another parasitic NPN BJT, which can be turned on by voltage spikes on the substrate or the N-well. This leakage current can create drop on a parasitic well resistance R_p of the parasitic thyristor shown in figure 7.4(c) and turn it on. The parasitic thyristor can also be triggered by very fast voltage change between anode and cathode, voltage spikes on pins exceeding their supply voltage by more than a diode drop or by an ESD event. Increasing temperature can cause an increase of the leakage current through the parasitic thyristor and create latch-up too. In products for space applications, latch-up can be caused by ionizing radiation [134].

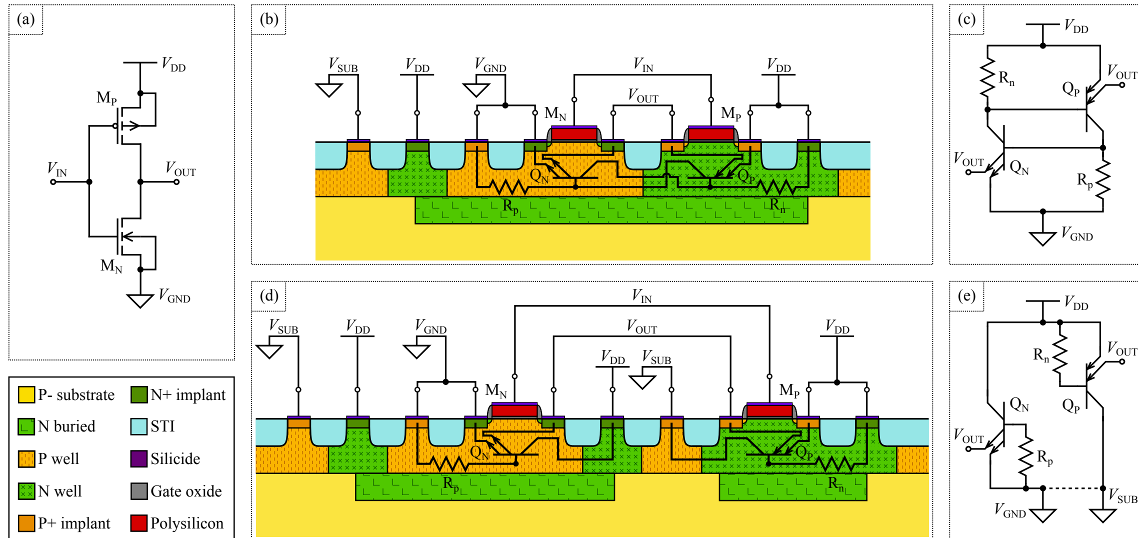


Fig. 7.4: Parasitic thyristor structures in CMOS or BCD designs: (a) Example of a frequently used circuit – an inverter. (b) Cross section with depicted parasitic components. (c) Schematic diagram of the parasitic thyristor structure. (d) Solution eliminating the parasitic thyristor structure. (e) Schematic diagram of the parasitic structure without thyristor configuration [8]

For suppression of the parasitic structures include the thyristors, use of the silicon on insulator (SOI) substrates is an option. However, this solution can be used only in cases of substantiated cost increase. In other cases, the bulk type of semiconductor wafers is chosen and the parasitic structures have to be considered and degraded by highly doped guard rings around threatened structures [121] or distance increase between these structures. Both methods decrease the gain of the parasitic BJT. However, the preferable solution is the separation of N-channel and P-channel MOSFETs as shown in figure 7.4(d) where the parasitic thyristor structure is removed as can be seen in figure 7.4(e). The proposed algorithm separates both types of MOSFETs and eliminates the thyristor structures inside designed devices.

The rules mentioned are sufficient to sort separate MOSFETs, but ICs are composed of other component types such as BJTs, diodes, passive components like resistors or capacitors, and also logic gates. Pre-placement rules are naturally defined for these component types as well. BJTs are sorted into rows according to collector terminal connection. Thus, similarly as MOSFETs, instances of same type and with same collector terminal connection are placed in one row of FPPM. As illustrated in figure 7.3(b) in some CMOS

technologies, PNP type BJTs lie directly on the substrate, which means that the collector terminal is connected directly to a substrate potential. Then, in this case, all PNP BJTs will be sorted into one row. Also for BJTs, individual rows are divided into columns while individual columns are occupied by transistors with the same base terminal connection.

In technologies mentioned above, passive components are mainly represented by resistors and capacitors. Both these components occur in various types such as diffusion, polysilicon and metallic resistors or capacitors formed by MOS structures, polysilicon-polysilicon capacitors and metallic capacitors. A cross section of a capacitor formed by MOS structure is shown in figure 7.3(c) and polysilicon resistor in figure 7.3(d). The fundamental rule is also used for passive components. Each type thus occupies given row of FPPM. But for some passive components, local substrate electrical connection is defined. The local P-well is insulated by suitably polarized deep N-well and can be connected to defined node, which is different from the substrate, as is shown in figure 7.3(d). Rows thus contain only components of the same type and with the same local terminal electrical connection. Passive components do not have any control terminals. Therefore individual rows of FPPM are not divided into columns.

In modern planar electronic circuits, there are used suitable polarized diodes, both diode of N+ – P-well or P+ – N-well type for antenna effect reduction and Zener diodes which can create, for example, a simple voltage limitation. These components also respect the fundamental pre-placement rule when individual rows correspond to individual diode types. Further row division follows the electrical connection of deeper diffusion layer. For example, the anode terminal determines partition of N+ – P-well diodes whose a cross section is shown in figure 7.3(e).

Logic parts of circuits are not designed with separated transistors but are designed with predefined standard cells – logic gates, which occupy a substantial part of AMS chips and are parts of almost all blocks. Therefore pre-placement rules for logic gates have been defined as well. A basic pre-placement rule for logic gates is their distribution into rows according to their voltage class. However logic gates of one voltage class can be connected to different supply or ground nodes. Hence the gates with the same voltage class and with the same supply and ground terminal connections located in one row of FPPM. A product of the automatic pre-placement phase with simplified layout view is shown in figure 7.2(b) where logic gates are located in lower rows and last two rows

contain gates with same voltage class but with different supply and ground connections. Such a grouping of logic gates allows simple spatial optimization without creating short circuits among individual supply or ground voltage domains and significantly simplifies the selection of components for manual placement or for use of an automatic placer.

Building blocks of ICs are not designed in one level but in a suitable hierarchy. Therefore these blocks not only contain the active and passive components but also the previously finished sub-blocks, which must be considered in the automatic pre-placement phase. Thus a pre-placement rule is also defined for sub-blocks, when all same sub-blocks are located in one common row of FPPM. An example of layout instance placement after the pre-placement phase is shown in figure 7.2(b) where instances have been replaced by symbolic views.

7.1.2 Algorithm Implementation

The algorithm of the automated pre-placement phase explained in the previous section has been implemented in Cadence CAD environment, which supports the Cadence SKILL programming language and allows implementation of new features [92]. Then a created program is used in Cadence Virtuoso Layout editor [69]. This program works with layout instances, which have been generated in the layout editor according to a reference schematic at the start of physical implementation.

A flow chart describing the pre-placement implementation is shown in figure 7.5. The program begins when user selects the relevant item in a custom menu in the layout editor. After the start, initial input checks are performed. Compatibility with the layout editor version, pre-placement rules definition, and correct database format is checked. The reference schematic of the created layout view is needed because some instance parameters are accessible in this schematic only. Therefore in the next step, checks of the reference schematic existence and of its readability are executed.

Some information about connectivity is not available in the layout view but they are stored in the reference schematic. Therefore, in this phase, the schematic is analyzed and checked if each layout instance has a corresponding schematic instance. When one of above mentioned checks fails, the program creates a log file with a list of appropriate error messages and then ends. After successful accomplishment of all checks, all layout instances are moved into 3rd quadrant of layout editor canvas to avoid overlays of layout

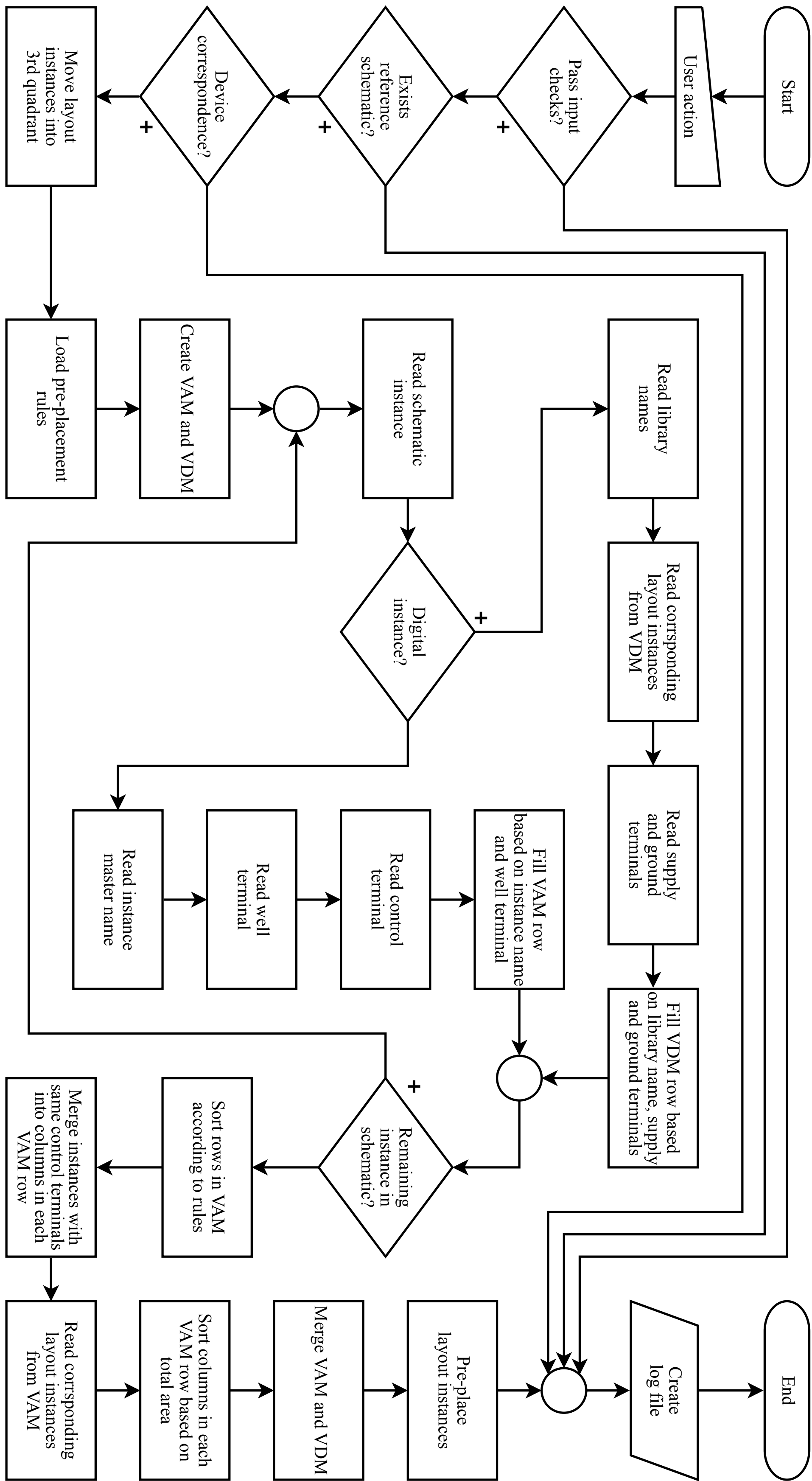


Fig. 7.5: The flow chart of the automated pre-placement phase [8]

instances during the final step of pre-placement. In the next step, rules for pre-placement defined in the previous section are loaded.

The program does not move physically each layout or schematic instance, but operates with their database representation – virtual objects – only. This procedure can save processing time, which would otherwise be used for repetitive movement and rendering layout and schematic instances. Therefore two virtual matrices, which will be filled by virtual objects and in which these virtual objects will be sorted and arranged, are created. Basic AMS circuits are usually composed of digital standard cells and of analog active and passive components. The virtual digital matrix (VDM) is created for digital standard cells and the virtual analog matrix (VAM) for analog components.

A core loop starts by reading an unread schematic instance in each run. If the read schematic instance is an analog component, the instance master name is read. Then according to the loaded pre-placement rules, which are described in previous section, well and control terminals are read. The well terminal is common designation for bulk or substrate terminals of analog components. For example, the well terminal of a N-type MOSFET is the bulk connection to a P-well. Similarly the control terminal is a common designation for gate and base terminals of analog components. Specifically for BJTs, the control terminal is the base terminal. For reading of well and control terminals, the schematic instance is used because the well terminal is not accessible in some types of layout instances.

In the next step, the prepared VAM is filled by database representations of schematic instances row by row on the base of created index. Therefore each row is indexed by a text string, which is composed of the master instance name and the well terminal of the appropriate virtual object. So in the result, each row of VAM is filled by virtual objects of the same type with the same well terminal connections. This way of indexing simplifies this naturally three-dimensional problem to two-dimensional, saving computing time and shortening the whole physical design. The process of VAM filling is shown in figure 7.6.

If the read schematic instance is the digital standard cell, the library name is found out and the corresponding layout instance is detected. Then supply and ground terminals are read from the layout instance because this connectivity information cannot be easily accessible in the original schematic instance due to inherited definitions of ground and supply connections. In this moment the rows of VDM are filled by database represen-

tations of layout objects on the base of indexes which consist of library name, name of supply terminal and name of ground terminal formed in strings. Similarly as VAM filling, this way of indexing also simplifies the problem just to two-dimensional. As a result, each row of VDM is filled by virtual objects of logic gates coming from the same library.

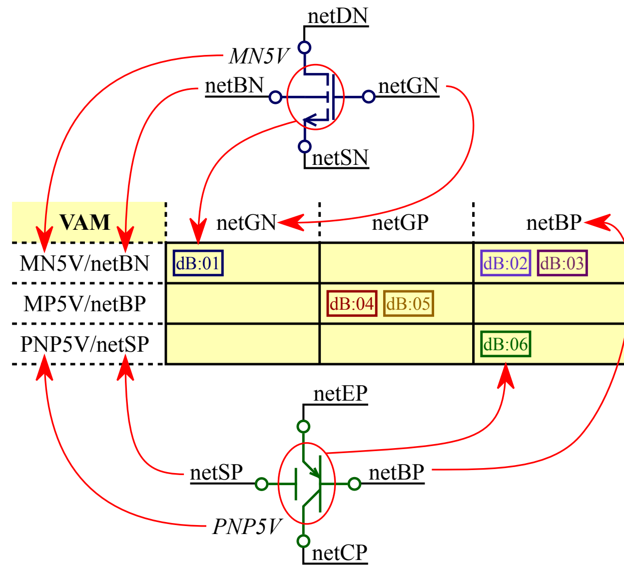


Fig. 7.6: VAM filling with outlined way of indexing [8]

If all schematic instances are read, the core loop is closed and the program goes into next phase, when an order of rows in VAM is specified according to loaded rules. Rows of VAM are filled by virtual objects of the same type and with common connection of well terminals but virtual objects inside rows are not grouped into columns. This is performed in the next step. Thus separated columns containing only virtual objects with common connection of control terminal are created. After this, all virtual objects in VAM have to be converted into layout form. Therefore corresponding layout instances are found out and the whole VAM is transferred from schematic virtual objects into layout virtual objects. This type of virtual objects contains information about all physical device dimensions which are used in next moment. Individual columns inside rows are arranged according to size of the total area, which all components in a column occupy.

Before moving the layout instances to positions given by arranged virtual objects, both virtual matrices VAM and VDM are being merged. A virtual matrix is formed so that the whole VAM is put on its beginning followed by VDM. This final virtual matrix is then read object by object, while coordinates of each virtual object are used for placement of

corresponding layout instances on canvas of the layout editor. Before the very end of the program, the log file is generated and it contains information about number of processed analog and digital instances and about the total computing time of the pre-placement step. At the end of the log file, a record about processed instances is listed, from which fundamental constraints in a required format [72] can be generated.

7.1.3 Productivity Improvement

Integration of the pre-placement phase into the standard flow allows effectiveness improvement of the whole design process. For quantification of productivity increase, time of manual standard sorting t_{MS} and time of automatic pre-placement phase t_{PP} to reach the final sorted matrix are introduced. The time saving between manual sorting and automated pre-placement is described as a relative ratio ρ between t_{MS} and t_{PP} according to

$$\rho = \frac{t_{MS}}{t_{PP}}. \tag{7.1}$$

Results based on time measurement for different type of AMS circuits and for individual methods of final matrix generation are listed in table 7.1 where circuit complexity is described by parameters of FPPM such as the number of rows N_{ROW}, the number of non-empty FPPM elements N_{NE}, and the number of layout instances N_{LI}. The durations t_{MS} of all measured circuits have been measured using the same equipment under the same conditions. Manual sorting has been performed by one reference layout engineer. Sorting generated by other engineers having different experiences, or changes of conditions have caused variations of measured t_{MS} but not fundamentally considering ρ.

Also all t_{PP} durations have been measured under the same conditions using the same equipment, in this case 32 processors system with clock frequency 2.6 GHz and 256 GB of random access memory. The Cadence Virtuoso Layout Editor [69] is a single thread application. Each measurement has always been made on an idle processor. Before the program starts, no initial setting is required. In other words, the setup time is zero and is no considered further. Therefore, the durations t_{PP} have been measured by automatic script implemented inside the program from pressing the start button till the program end.

The measured AMS circuits have been designed in 160 nm technologies BCD8sP and SOIBCD8S from STMicroelectronics. Based on obtained results, the introduced auto-

Type of circuit	N_{LI} (-)	N_{ROW} (-)	N_{NE} (-)	t_{MS} (s)	t_{PP} (s)	ρ (-)	t_{PPF} (min)	t_{MSF} (min)	$d_{PPF,MSF}$ (%)	t_{ER} (min)	t_{MSFE} (min)	$d_{PPF,MSFE}$ (%)
Comparator	29	4	11	110	0.028	3 929	724	746	-2.99	47	793	-8.74
Comparator with hysteresis	32	6	13	155	0.038	4 079	819	844	-2.98	52	896	-8.61
Signal clamper	40	9	17	246	0.032	7 688	1 029	1 064	-3.29	75	1 140	-9.77
Operational amplifier	52	7	16	249	0.065	3 831	1 177	1 218	-3.35	69	1 287	-8.54
Supply selector	66	6	17	167	0.047	3 553	1 345	1 395	-3.54	71	1 466	-8.21
Over voltage protection	78	8	15	174	0.055	3 164	1 411	1 471	-4.11	82	1 551	-9.03
Rail-to-rail input comparator	94	4	23	431	0.106	4 066	1 686	1 773	-4.92	102	1 875	-10.09
Zero crossing detector	99	6	20	362	0.068	5 324	1 660	1 737	-4.45	96	1 833	-9.46
Error amplifier	112	9	19	299	0.092	3 250	1 716	1 802	-4.75	122	1 922	-10.72
Operational amplifier with auto-trimming	112	8	26	572	0.117	4 889	1 879	1 976	-4.90	148	2 124	-11.53
Low drop out regulator	122	40	57	2 280	0.089	25 618	3 470	3 604	-3.72	286	3 890	-10.80
Soft start	129	7	25	390	0.098	3 980	1 941	2 042	-4.95	132	2 178	-10.87
Over current protection	197	13	32	746	0.143	5 217	2 423	2 574	-5.88	169	2 743	-11.68
Programmable zero crossing comparator	213	15	33	742	0.173	4 289	2 506	2 669	-6.10	192	2 861	-12.40
Fast comparator with auto-trimming	324	9	28	677	0.196	3 454	2 698	2 856	-5.53	178	3 024	-10.79
Current reference	358	44	79	3 420	0.199	17 186	6 793	7 157	-5.09	324	7 481	-9.20

Tab. 7.1: Results of productivity improvement based on design time measurement for different type of AMS circuits [8]

mated pre-placement is in the range of 3 164 times to 20 099 times faster compared to manual pre-placement.

A comparison has been performed among durations of the entire physical design flow with the pre-placement phase t_{PPF}, the standard physical design flow t_{MSF}, and the standard physical design flow including the time required for bulk error removal t_{MSFE}. Therefore the relative difference $d_{\mathrm{PPF,MSF}}$ between t_{PPF} and t_{MSF} has been calculated by

$$d_{\mathrm{PPF,MSF}} = \frac{t_{\mathrm{PPF}} - t_{\mathrm{MSF}}}{t_{\mathrm{MSF}}} \cdot 100 \tag{7.2}$$

and the results are listed in table 7.1. Thus the presented automatic pre-placement has a positive influence on duration of the entire physical design flow, it is shortened by 4.41 % on average. Use of the proposed pre-placement phase brings multiple advantages to floor-planning and placement flow. It is well known that the complexity of placement or automatic placement increases with devices count [72]. By applying the automatic pre-placement phase and creating groups of devices related to insulation pockets described in section 7.1.1, the task is simplified and N_{LI} is substituted by N_{ROW}. In other words, components of the same type with the same bulk terminal connection are usually grouped and placed into one common pocket. Then the final placement is performed with all pockets instead of all instances. This substitution simplifies and accelerates placement flow because the number of pockets is smaller than the number of layout instances. This way of placement is marked as a pocket based approach and can be seen in figures 4.9 and 5.7. This phase is one step of the entire physical design flow but is able to shorten it. Modern ICs are composed of tens and hundreds of blocks. If 4.41 % of design time is saved on each block in an IC, it means saving units of days in total which can be the decisive time for IC delivery.

If MOSFETs require electrically separated bulk terminals, then it is necessary to have wells below these MOSFETs electrically separated. This is usually done by suitably biased PN junctions. Wells and buried well implant layers are used for lateral and vertical isolation of different bulks [121]. Wells are usually very deep and thick implantation and for robust design rules require larger spacing for separation. This is the reason why for bulk separation there is a lot of area sacrificed. So if an error of bulk connections is made and is detected in final verification, while whole compact and optimized layout is almost done, then the correction is very complex and time-consuming. These errors may negatively affect the quality of the final layouts.

The error of bulk connection has been simulated for all mentioned circuits by reconnection of the bulk terminal of a MOSFET. Then, durations t_{ER} needed to fix the error in the bulk connection have been measured and are listed in table 7.1. In general, the error of wrong bulk connection can be corrected much faster if N_{LI} and also N_{NE} are lower. Errors due to wrong bulk connections are usually one of the most time consuming modification in AMS layout and the correction process depends on the circuit complexity and requires many hours. By applying the automated pre-placement phase, wrong bulk connection errors are effectively suppressed and hours of later reworks are saved. It is shown on the relative difference $d_{\mathrm{PPF,MSFE}}$ between t_{PPF} and t_{MSFE}, which is 10.03 % on average.

In practice, for the final IC to be competitive, there is a great deal of pressure on the smallest possible size of the semiconductor die. The occupied area of silicon is thus becoming a key parameter. The automatic pre-placement phase does not only save physical design time but also helps to save an area on the chip. Technology design rules do not allow to place different types of components as close to each other as the same types. In other words, the smallest device spacing and layout area are reached when devices of the same type share the same bulk connection. Due to this fact, placing them together is a preferred solution.

The advantage of the described flow is that interconnections of components with the same bulk terminal are very short because these components are as close as possible. Thus the short interconnections represent small parasitic resistances and capacities, which helps to create a good quality design [135]. So only interconnections among pockets must be optimized.

Frequently used blocks such as differential pairs or current mirrors usually have common bulk terminal connection. Therefore, by this approach, they are placed in the smallest area possible. So an effect of the thermal gradient well-described in [136] is minimized [121].

In advanced sub-micron technology nodes, a physical effect known as the well-edge proximity effect become more significant. This effect arises during the ion implantation of wells by ion scattering at edges of the photoresist and causes considerable mismatch among MOSFET parameters [137], [138]. Therefore it is advisable to keep sensitive analog components at a suitable distance from the edges of the wells. Thus if the components are kept far from the well edges and are not grouped according to component type, the

area increases considerably. When the AMS physical design flow with the automatic pre-placement phase is used, the relevant components are grouped and the number of the well edges is limited. In this manner, influence of the well-edge proximity effect is minimized and the occupied area is smaller. Incorporation of the pre-placement phase into AMS physical design flow helps to create ICs with high-quality layout.

7.2 Incremental Control of Layout Objects

During the physical design process, layout objects are often modified through filling various complicated forms, which is not user-friendly and slows down the design process. Therefore a new concept of control of layout objects has been proposed. Complicated form filling is replaced by an incremental approach. The idea has been presented in [10], and the complete description has been published in [9].

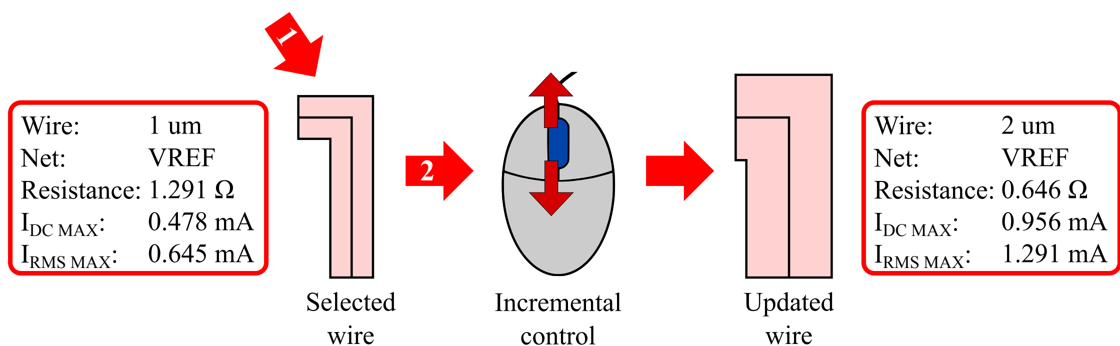


Fig. 7.7: The process of the wire width modification with the shown status bar

The incremental control allows modifications of fundamental properties of layout objects. Typing exact values to forms is replaced by a very simple mouse scrolling as can be seen in figure 7.7 where a width of a wire is changed. Users to be able to have modifications under control, fundamental properties of layout objects including electrical parameters are immediately displayed in a status bar. In this way, it is possible to edit, for example, the width of wires, the number of via cuts, an aspect ratio of rectangles, dimensions of pins and labels, or even the number of gate fingers of MOSFETs or an aspect ratio of capacitors. Discrete values such as the number of gate fingers or the number of via cuts are naturally incremented or decremented. Values exactly entered such as the wire width or the label height are circularly selected from a predefined set, which is composed of the

most used values. The change of the aspect ratio is performed with a suitable chosen step. A property to be modified is chosen by a modifier on a keyboard.

After selection of any layout object, the status bar immediately displays the actual parameters. Then, after an increment or a decrement, the parameters are updated, and the values on the bar are recalculated as can be seen in figure 7.7 where wire parameters such as resistance, maximum electromigration current, and connectivity are displayed. Use of the status bar allows checking fundamental electrical properties of layout object during layout creation, which increases the robustness of produced designs and reduces the number of created errors.

7.2.1 Implementation in CAD Environment

The incremental control of layout object has been implemented in Cadence Virtuoso Layout Editor [69] using the SKILL programming language [92]. This implementation has been presented in [11]. A flow chart of an implemented program is shown in figure 7.8. The program starts when the user presses a modifier and performs an increment by the mouse wheel. Then the program identifies selected layout objects and reads the incremental control database (ICDB), which contains a list of changeable properties of each object type and determines a way of the modification. In the case that continuous values to be entered, the ICDB contains the predefined set of values, which is read incrementally.

During the modification, an actual value of a changeable property can be slightly different than the desired value due to an alignment of layout objects to design grid. After repeated modifications, the actual value can shift from the desired value and an error can occur. Therefore an initial value of the changeable property is introduced, which allows calculating the actual value based on the correct original value. Hereby the maximal possible accuracy of consecutive incremental changes is ensured. If the initial value is required and not yet set, the program stores the initial value as a parameter of the layout object. Together with this, the program creates a record of the last updated value, which serves as a decisive criterion whether the layout object has been changed by this program or in a different way. When the initial value is set, then it is read and compared with the actual value. If these values equal, the layout object has not been modified by an external intervention and the initial value is untouched. If these values are different, some external

intervention has been performed and the initial value is set to the actual value. Thus the program respects the external intervention.

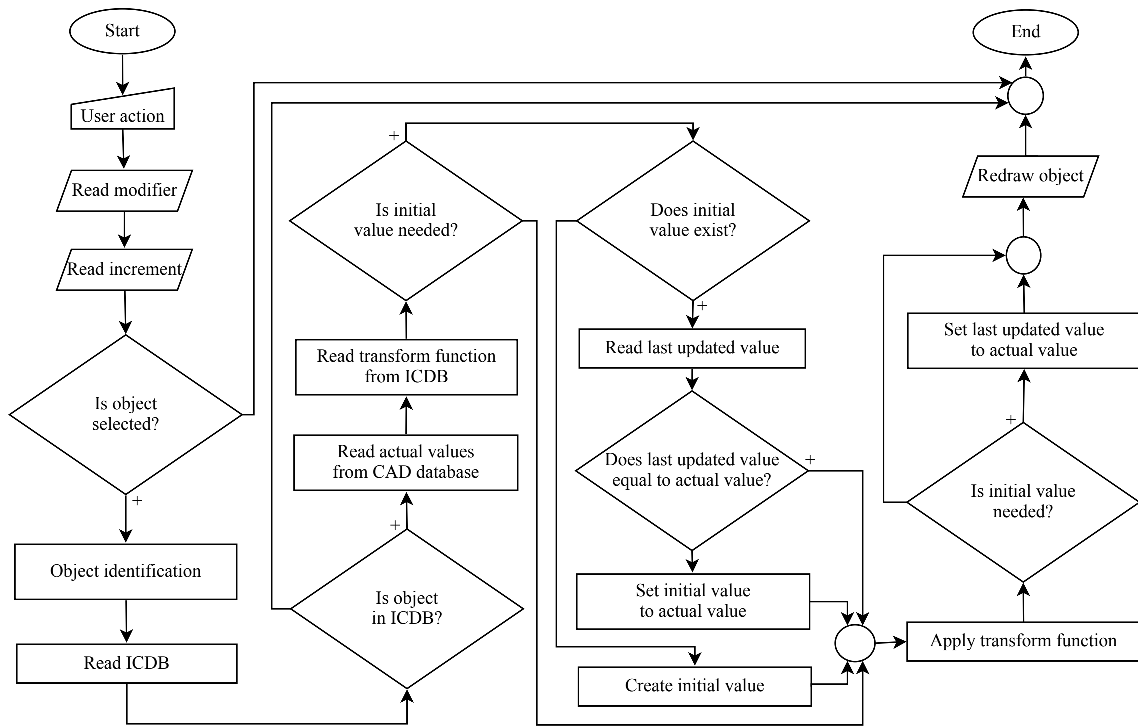


Fig. 7.8: The flow chart of the incremental control of layout objects [9]

After these steps, the update of a current state is executed based on obtained parameters and a function, which is defined in the ICDB. When the layout object type needs to set the initial value, the last updated value is set to the actual value. So the last updated value is the parameter of the layout object. It allows that its value will be used in the next run of this program. By cyclic applications of this program, the layout object is controlled incrementally.

7.2.2 Productivity Gain

The incremental control of layout objects reduces the number of actions needed to reach the optimal result. The average time of modifications performed by a standard approach t_{SA} and the average time of modifications made by the incremental control t_{IC} are introduced to reach a target form of layout objects with a random initial value. To quantify layout productivity gain, the relative difference $d_{\mathrm{IC,SA}}$ between t_{IC} and t_{SA} is calculated for various types of layout objects according to the equation (7.2). The results listed in

table 7.2 show that the use of the incremental control saves the time in the range of 23 % to 67 %. So the incremental control brings an intuitive control concept, which is able to increase productivity and helps to create more robust designs of AMS circuits including TRNGs.

Object type	t_{SA} (s)	t_{IC} (s)	$d_{IC,SA}$ (%)
Via	5.6	2.0	-64.7
Wire	4.6	3.5	-23.4
Pin	7.0	4.5	-36.1
Label	5.8	3.5	-39.9
Rectangle	5.2	3.2	-38.5
MOSFET	5.6	3.7	-33.6
Capacitor	5.4	3.5	-35.2
Pin with Label	15.1	5.0	-66.7

Tab. 7.2: The results demonstrating productivity gain created by the incremental control [9]

7.3 Search for Objects based on Their Similarities

Designs of AMS circuits contain a large number of objects. During creation or modification of designs, it is often necessary to search for different objects in the CAD database. A currently used search flow of the objects is a complicated process, which is based on filling parameters and their exact values into a search form. Therefore a new search concept based on similarities of objects has been proposed. This concept named similar search has been integrated into the Cadence Virtuoso Schematic and Layout editors [88], [69] through a program created in the SKILL programming language [92], and has been presented in [12].

The currently used search flow does not allow to search based on a name of the object, their property, and their connectivity together. Moreover, it cannot be realized without typing. However, the developed similar search allows to search based on properties and connectivity of currently selected object or more objects and even typing is not needed as

can be seen in figure 7.9. The proposed search process takes place in the following phases. Some object is selected and set as a reference. The similar search detects its parameters and connectivity and offers possibilities to search. After choice an option, objects with similar parameters are selected.

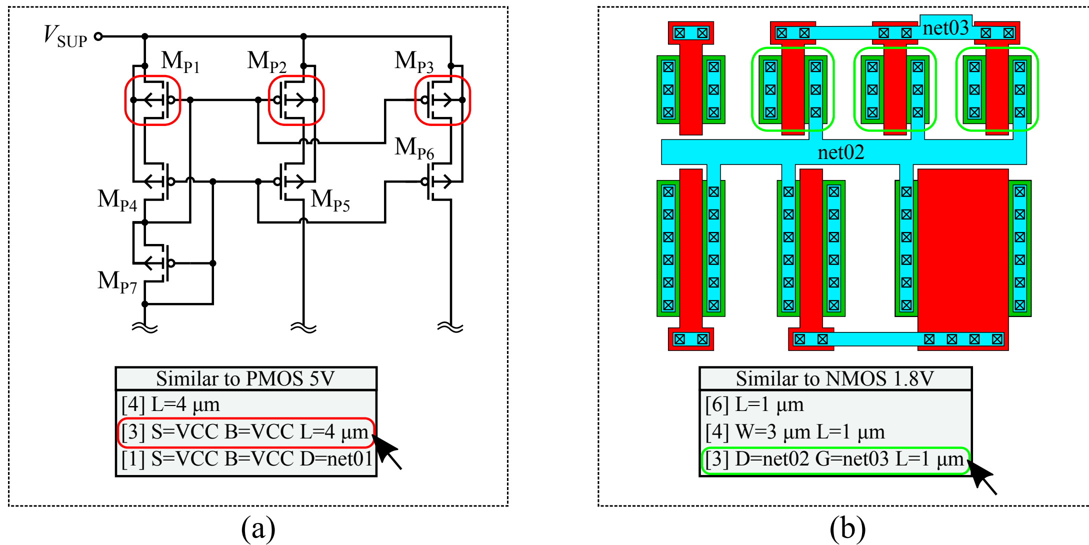


Fig. 7.9: Use of the similar search in the schematic editor (a) and layout editor (b)

Selected circuit part	Editor type	N_O	t_{MA} (s)	t_{SS} (s)	$d_{SS,MA}$ (%)
Cascoded current mirror	Schematic	17	49.6	7.0	-85.9
Cascoded current mirror	Layout	73	76.0	6.1	-92.0
Digital decoder	Schematic	14	20.3	4.9	-75.9
Digital decoder	Layout	18	22.4	5.2	-76.8

Tab. 7.3: The results demonstrating productivity increase created by the similar search [12]

The similar search creates a common approach to object search in schematic and layout editors and increases design productivity. It allows finding a set of objects, which cannot be selected by the currently used search flow. The productivity increase is demonstrated by the results listed in table 7.3 where the relative difference $d_{SS,MA}$ between the average time of selection by the similar search t_{SS} and the average time of manual selection t_{MA} is calculated according to to the equation (7.2). Shown examples in table 7.3

cannot be selected by the currently used search flow. Their complexity is given by the number of objects N_{O}. The results show that the use of the similar search accelerates the design flow of AMS circuits containing both individual transistors and standard cells.

7.4 Classification of Matched Structures

The systematic mismatch occurs in ICs and negatively affects sensitive analog and AMS circuits. This effect can be reduced by a proper layout technique so-called matching. In practice, there is a need to decide, which pattern of a matched structure is better regarding systematic mismatch reduction. Therefore a new method classifying matched structures has been introduced and presented in [13].

Matched structures composed of active or passive devices are classified based on an estimation of a parameter gradient function, which is modeled up to fifth order. Because a final position of the matched structure on a wafer is not usually known, evaluation is done in different directions and the worst cases in each order form a five element mismatch vector. The pattern is usually composed of more than two devices, for example, A, B, and C while A is a reference. Then mismatch vectors representing matching A to B and A to C are produced. The developed program allows setting weights for each matching. The mismatch vectors are summed and weighted. The result is the evaluation vector representing the quality of the matched structure while lower vector values means better systematic mismatch suppression. The patterns are extracted from the Cadence Virtuoso Layout editor [69] by a script created in the SKILL programming language [92]. The program evaluating the pattern quality has been developed in MATLAB [96]. This method helps to design more robust analog or AMS circuits. All details are available in [13].

7.5 Future Work on Physical Design Methodology

AMS blocks and ICs created by the Analog-on-Top approach are still handmade. To be possible to use an automatic analog placer and create correct placement, a large number of constraints must be defined. Therefore further development is aimed at an automatic generation of physical design constraints, which will be extracted from huge CAD databases by modern processing big data.

Chapter 8

Conclusions

True random number generators (TRNGs) are essential devices for ensuring the security of modern communication systems. They generate random number sequences based on physical phenomena with the random behavior, in this case, electronic noises occurring in CMOS ICs. Therefore the produced sequences cannot be predicted. This work presents new structures of TRNGs, which are integrable in chips, occupy small areas, have low power consumption, and can be a part of multi-system chips. For these reasons, they are suitable for modern hand-held devices.

An increasing level of system security requires a higher amount of quality random data generated with sufficient data rate. Therefore the first proposal named the TRNG with time multiplexed metastability-based sources of randomness uses the so-called principle of pipelining, which has not been used in already published works. The process of random data generation is described in section 4.1. This TRNG has been designed and fabricated in the 130 nm HCMOS9GP bulk CMOS technology from STMicroelectronics and occupies an area of $0.029\,\mathrm{mm}^2$.

Randomness is extracted by four noise sources from thermal and flicker noise present in CMOS circuits. All digitized random signals are combined in the time multiplexer, which increases the output random data rate. Random number sequences have been generated by various output random data rates at various temperatures and verified by the FIPS and NIST statistical test suites. Their results listed in tables 4.2, 4.3, 4.4, and 4.5 have shown that the proposed TRNG is able to operate without any correctors at the output data rate up to 20 Mb/s. Measured power consumption depicted in figure 4.17 is low compared to already published generators as mentioned in table 4.6. Moreover,

the required energy needed for the generation of a random bit is one of the lowest. The designed TRNG does not contain any passive devices. This feature simplifies its migration to advanced technology nodes, which allows increasing the output random data rate and decreasing the power consumption even more. The measurements made have shown that the changing environmental conditions do not affect properties of random number sequences. Based on achieved parameters, it can be said that the presented TRNG is suitable for being integrated into a SoC, which is a part of modern mobile devices. The obtained results have been published in the impacted journal Radioengineering [4].

Security of the systems containing TRNGs may be impaired by deliberate malicious attacks. Attackers try to manipulate with properties of random number sequences and guess parts or even entire sequences. Therefore the new enhanced generic architecture (EGA) introduced in chapter 5 includes mechanisms, which are able to detect the bias of random number sequences caused by a possible attack and also reveal a significant decrease in the entropy of sources of randomness. This idea has been published in the reviewed journal ElectroScope [5]. The function of the introduced attack detector is based on features of the von Neumann corrector and explained in section 5.2.1. Biased random number sequences cause variations of corrector random data rate, and these variations are detected.

The low entropy detector is based on the approximately entropy estimation described in section 5.2.2. With the sudden decrease in the entropy, the detector reconfigures the designed reconfigurable noise source and thus try to recover the quality of random number sequences. The reconfigurable source of randomness introduced in section 5.3 is derived from the source of randomness described in section 4.2 and is capable of producing sequences of random bits in two settings. If the entropy detector detects lost of randomness, the reconfigurable source of randomness is switched to the second setting, where is a chance that it could again start producing the random sequences. These circuits have been designed in the 130 nm HCMOS9A bulk CMOS technology from STMicroelectronics and tested by the statistical test suites mentioned above. The results listed in tables 5.5 and 5.4 have confirmed that this source of randomness can generate the quality random numbers with the output random data rate of 25 Mb/s. The power consumption of this circuit is 203.6 µW respectively 222,7 µW in the second circuit setting. Thus the required energy per random bit is 8.14 pJ/b respectively 8.91 pJ/b. These achieved values are very

low compared to values in table 4.6. However, they are slightly higher than the values achieved by the TRNG with time multiplexed sources of randomness described in chapter 4. It is mainly due to the use of another variant of technology.

Development of TRNGs still continues and deals with the connection of the proposed EGA with the architecture of the introduced TRNG with time multiplexed metastability-based sources of randomness as described in section 5.5.1. Future work mentioned in section 5.5.2 will explore the idea of active reduction of an influence of the mismatch among components in the noise source.

The presented TRNGs are designed to be parts of SoCs. Simulations of the whole SoCs at the most accurate transistor level are almost impossible due to enormous time demands. Therefore individual parts of the system can be substituted by their behavioral models, which speed up the simulations. The models have been created using Verilog-A HDL and approximate properties of the designed TRNGs at the transistor level. Therefore randomness has been modeled by an available PRNG in Verilog-A HDL. The behavioral models of the presented TRNGs are described in chapter 6. They are able to generate random number sequences with very similar properties as random number sequences produced by the designed generators, which show the results of the statistical test suites listed in tables 6.1, 6.2, 6.3, and 6.3. Simulations of the created behavioral models take tens of minutes when a number sequence with the length of 1 Mb is generated as can be seen in sections 6.1.2 and 6.2.2. However, simulations of the designed TRNGs at the transistor level take several days. Thus the duration of simulations of the behavioral models is incomparably shorter than the duration of the transistor level simulations. Therefore it is necessary to use the behavioral models instead of the transistor level proposals during verifications of SoCs.

The developed TRNGs are typical AMS circuits containing very sensitive parts, whose physical implementation have to be done very carefully because any inaccuracy can cause the bias of generated random number sequences. However, the precise handmade physical design is very time-consuming. Therefore, during the development of the TRNGs, new methodology steps of physical design of AMS ICs have been proposed. They speed up the physical implementation, help to prevent errors and make the design more robust. The newly introduced feature named the automated pre-placement phase categorizes electrical devices according to their topological, structural and electrical properties, replaces human

labor, and can prevent the creation of hardly detectable errors occurring at the beginning of AMS physical design. Moreover, the product of the proposed algorithm can be used for very fast and precise area estimation, manual placement as well as basic constraints for automatic placers. As can be seen from the obtained results listed in table 7.1, introduction of the automated pre-placement phase has a positive influence on duration of the entire physical design flow, which is shortened by 4.41 % on average. If 4.41 % of design time are saved on each block of a modern IC composing of tens and hundreds of blocks, the new design phase allows saving units of days in total, which may be the decisive time for IC delivery. Additional time mentioned in section 7.1.3 is saved when this method prevents frequently-occurring errors such as the wrong bulk connection of MOSFETs. The results and description of the automated pre-placement phase have been published in the impacted journal named Integration, the VLSI Journal [8].

This thesis also presents the new way of control of layout objects when standard complicated form filling is replaced by an incremental approach. Users have all modifications under control because fundamental properties of layout objects including electrical parameters are immediately displayed in the created status bar. The incremental control saves the time needed for the modifications in the range of 23 % to 67 % as can be seen in table 7.2. This method has been published in the reviewed journal named Advances in Science, Technology and Engineering Systems Journal [9]. During development of the introduced TRNGs, other features searching an design database based on similarity of object properties and classifying matched structures regarding systematic mismatch have been proposed. They are also mentioned in chapter 7. Future work mentioned in 7.5 is aimed at an automatic generation of physical design constraints, which will be extracted from huge CAD databases by modern processing big data.

The results of this doctoral thesis are summarized in section 1.2 and come from the development, which has been supported by the Grant Agency of the Czech Technical University in Prague, grant No. SGS17/188/ OHK3/3T/13 (Mikro a nanostruktury a součástky), grant No. SGS14/195/OHK3/3T/13 (MiNa), grant No. SGS11/156/OHK3/ 3T/13, the GAČR project No. 02/09/160, and Ministry of the Interior grant No. VG2010 2015015. Parts of this thesis have been developed in cooperation with STMicroelectronics.

References

[1] A. J. Menezes, P. C. V. Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*. Boca Raton, FL: CRC Press, 1996.

[2] W. Schindler and W. Killmann, “Evaluation criteria for true (physical) random number generators used in cryptographic applications,” in *Cryptographic Hardware and Embedded Systems – CHES 2002*, Berlin, Heidelberg, Germany, 2003, pp. 431–449.

[3] A. Andreini, C. Contiero, and P. Galbiati, “A new integrated silicon gate technology combining bipolar linear, CMOS logic, and DMOS power parts,” *IEEE Trans. Electron Devices*, vol. ED-33, no. 12, pp. 2025–2030, 1986.

[4] V. Kotě, P. Vacula, V. Molata, O. Veselý, O. Tláskal, D. Barri, J. Jakovenko, and M. Husák, “A true random number generator with time multiplexed sources of randomness,” *Radioengineering*, 2018, (In press).

[5] V. Kotě, V. Molata, and J. Jakovenko, “Enhanced generic architecture for safety increase of true random number generators,” *ElectroScope*, vol. 2014, no. 3, 2014.

[6] V. Kotě, V. Molata, and J. Jakovenko, “New structure of true random number generators with protective mechanisms,” in *Electronic Devices and Systems IMAPS CS International Conference 2014*, Brno, Czech Republic, 2014, pp. 19–24.

[7] V. Kotě, V. Molata, and P. Vacula, “Behavioral models of true random number generators,” in *Proceedings of the International Student Scientific Conference Poster – 22/2018*, Prague, Czech Republic, 2018, pp. 1–6.

[8] V. Kotě, A. Kubačák, P. Vacula, J. Jakovenko, and M. Husák, "Automated pre-placement phase as a part of robust analog-mixed signal physical design flow," *Integration, the VLSI Journal*, 2018, (In press).

[9] P. Vacula, V. Kotě, A. Kubačák, M. Lžíčař, R. Zelený, M. Husák, and J. Jakovenko, "Incremental control techniques for layout modification of integrated circuits," *Advances in Science, Technology and Engineering Systems Journal*, vol. 2, no. 3, pp. 1196–1201, 2017.

[10] P. Vacula, V. Kotě, A. Kubačák, M. Lžíčař, R. Zelený, M. Husák, and J. Jakovenko, "Incremental control techniques for layout modification," in *11th International Conference on Advanced Semiconductor Devices & Microsystems (ASDAM)*, Smolenice, Slovakia, 2016, pp. 239–242.

[11] P. Vacula, V. Kotě, A. Kubačák, M. Lžíčař, R. Zelený, M. Husák, and J. Jakovenko, "Modern control techniques for layout creation," in *CDNLive Cadence User Conference EMEA 2016*, Munich, Germany, 2016.

[12] P. Vacula, V. Kotě, A. Kubačák, S. Cliquennois, M. Lžíčař, M. Husák, and J. Jakovenko, "Modern search techniques for layout creation," in *CDNLive Cadence User Conference EMEA 2017*, Munich, Germany, 2017.

[13] P. Vančura, V. Kotě, P. Vacula, A. Kubačák, and J. Jakovenko, "Matched structure classification," in *CDNLive Cadence User Conference EMEA 2017*, Munich, Germany, 2017.

[14] *Eldo User's Manual: Software Version 6.8_3 Release AMS 2006.2b*, Mentor Graphics Corporation, Wilsonville, OR, 2006.

[15] B. Jun and P. Kocher, *The Intel Random Number Generator*, Cryptography Research, Inc., San Francisco, CA, 1999.

[16] *Evaluation of VIA C3 Nehemiah Random Number Generator*, Cryptography Research, Inc., San Francisco, CA, 2003.

[17] *STM32F405xx, STM32F407xx – Datasheet*, STMicroelectrnoics, 2016.

[18] X. Tang, Z. M. Wu, J. G. Wu, T. Deng, J. J. Chen, L. Fan, Z. Q. Zhong, and G. Q. Xia, “Tbits/s physical random bit generation based on mutually coupled semiconductor laser chaotic entropy source,” *Optics Express*, vol. 23, no. 26, pp. 33 130–33 141, 2015.

[19] T. Sugiura, Y. Yamanashi, and N. Yoshikawa, “Demonstration of 30 Gbit/s generation of superconductive true random number generator,” *IEEE Trans. Appl. Supercond.*, vol. 21, no. 3, pp. 843–846, 2011.

[20] A. Alkassar, T. Nicolay, and M. Rohe, “Obtaining true-random binary numbers from a weak radioactive source,” in *Computational Science and Its Applications – ICCSA 2005*, Singapore, 2005, pp. 634–646.

[21] D. Rüschen, M. Schrey, J. Freese, and I. Heisterklaus, “Generation of true random numbers based on radioactive decay,” in *Proceedings of the International Student Scientific Conference Poster – 21/2017*, Prague, Czech Republic, 2017, pp. 1–4.

[22] S. Oosawa, T. Konishi, N. Onizawa, and T. Hanyu, “Design of an STT-MTJ based true random number generator using digitally controlled probability-locked loop,” in *IEEE 13th International New Circuits and Systems Conference (NEWCAS)*, Grenoble, France, 2015, pp. 1–4.

[23] W. T. Holman, J. A. Connelly, and A. B. Dowlatabadi, “An integrated analog/digital random noise source,” *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 44, no. 6, pp. 521–528, 1997.

[24] M. Eberlein and R. A. Bakar, “An integrated channel noise-based true random number generator,” in *7th International Conference on ASIC (ASICON ’07)*, Guilin, China, 2007, pp. 391–394.

[25] V. Kotě, V. Molata, and J. Jakovenko, “Improved structure of true random number generator with direct amplification of analog noise,” *ElectroScope*, vol. 2012, no. 6, 2012.

[26] P. Kohlbrenner and K. Gaj, “An embedded true random number generator for FPGAs,” in *Proceedings of the 2004 ACM/SIGDA 12th International Symposium on Field Programmable Gate Arrays (FPGA ’04)*, Monterey, CA, 2004, pp. 71–78.

[27] B. Sunar, W. J. Martin, and D. Stinson, “A provably secure true random number generator with build-in tolerance to active attacks,” *IEEE Trans. Comput.*, vol. 56, no. 1, pp. 109–119, 2007.

[28] D. Schellekens, B. Preneel, and I. Verbauwhede, “FPGA vendor agnostic true random number generator,” in *International Conference on Field Programmable Logic and Applications (FPL ’06)*, Madrid, Spain, 2006.

[29] V. Fischer, F. Bernard, N. Bochard, and M. Varchola, “Enhancing security of ring oscillator-based TRNG implemented in FPGA,” in *International Conference on Field Programmable Logic and Applications (FPL 2008)*, Heidelberg, Germany, 2008, pp. 245–250.

[30] N. Bochard, F. Bernard, and V. Fischer, “Observing the randomness in RO-based TRNG,” in *International Conference on Reconfigurable Computing and FPGAs (ReConFig ’09)*, Quintana Roo, Mexico, 2009, pp. 237–242.

[31] U. Güler and S. Ergün, “A high speed IC random number generator based on phase noise in ring oscillators,” in *Proceedings of 2010 IEEE International Symposium on Circuits and Systems (ISCAS)*, Paris, France, 2010, pp. 425–428.

[32] T. Amaki, M. Hashimoto, and T. Onoye, “Jitter amplirier for oscillator-based true random number generator,” *IEICE Trans. Fundamentals*, vol. E96–A, no. 3, pp. 684–696, 2013.

[33] A. Cherkaoui, V. Fischer, A. Aubert, and L. Fesquet, “A self-timed ring based true random number generator,” in *IEEE 19th International Symposium on Asynchronous Circuits and Systems (ASYNC)*, Santa Monica, CA, 2013, pp. 99–106.

[34] U. Güler, A. E. Pusane, and G. Dündar, “Investigating flicker noise effect on randomness of CMOS ring oscillator based true random number generators,” in *International Conference on Information Science, Electronics and Electrical Engineering (ISEEE)*, Sapporo, Japan, 2014, pp. 845–849.

[35] M. Coustans, C. Terrier, T. Eberhardt, S. Salgado, A. Cherkaoui, and L. Fesquet, “A subthreshold 30pJ/bit self-timed ring based true random number generator for

internet of everything," in *IEEE SOI-3D-Subthreshold Microelectronics Technology Unified Conference (S3S)*, Burlingame, CA, 2017.

[36] A. T. Do and X. Liu, "25 fJ/bit, 5Mb/s, 0.3 V true random number generator with capacitively-coupled chaos system and dual-edge sampling scheme," in *IEEE Asian Solid-State Circuits Conference (A-SSCC)*, Seoul, South Korea, 2017, pp. 61–64.

[37] G. Saxl, M. Ferdik, and T. Ussmueller, "Ultra-low-power ring oscillator based true random number generator for passive UHF RFID tags," in *IEEE Topical Conference on Wireless Sensors and Sensor Networks (WiSNet)*, Anaheim, CA, 2018, pp. 99–102.

[38] D. J. Kinniment and E. G. Chester, "Design of an on-chip random number generator using metastability," in *Proceedings of the 28th European Solid-State Circuits Conference (ESSCIRC 2002)*, Florence, Italy, 2002.

[39] C. Tokunaga, D. Blaauw, and T. Mudge, "True random number generator with a metastability-based quality control," *IEEE J. Solid-State Circuits*, vol. 43, no. 1, pp. 78–85, 2008.

[40] J. Holleman, S. Bridges, B. P. Otis, and C. Diorio, "A 3 μW CMOS true random number generator with adaptive floating-gate offset cancellation," *IEEE J. Solid-State Circuits*, vol. 43, no. 5, pp. 1324–1336, 2008.

[41] S. Srinivasan, S. Mathew, V. Erraguntla, and R. Krishnamurthy, "A 4Gbps 0.57pJ/bit process-voltage-temperature variation tolerant all-digital true random number generator in 45nm CMOS," in *22nd International Conference on VLSI Design*, New Delhi, India, 2009, pp. 301–306.

[42] V. B. Suresh and W. P. Burleson, "Entropy extraction in metastability-based TRNG," in *IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*, Anaheim, CA, 2010.

[43] V. B. Suresh and W. P. Burleson, "Entropy and energy bounds for metastability based TRNG with lightweight post-processing," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 62, no. 7, pp. 1785–1793, 2015.

[44] V. B. Suresh and W. P. Burleson, "Robust metastability-based TRNG design in nanometer CMOS with sub-vdd pre-charge and hybrid self-calibration," in *13th International Symposium on Quality Electronic Design (ISQED)*, Santa Clara, CA, 2012, pp. 298–305.

[45] S. K. Mathew, S. Srinivasan, M. A. Anders, H. Kaul, S. K. Hsu, F. Sheikh, A. Agarwal, S. Satpathy, and R. K. Krishnamurthy, "2.4 Gbps, 7 mW all-digital PVT-variation tolerant true random number generator for 45 nm CMOS high-performance microprocessors," *IEEE J. Solid-State Circuits*, vol. 47, no. 11, pp. 2807–2821, 2012.

[46] S. K. Mathew, D. Johnston, S. Satpathy, V. Suresh, P. Newman, M. A. Anders, H. Kaul, A. Agarwal, S. K. Hsu, G. Chen, and R. K. Krishnamurthy, "μRNG: A 300–950 mV, 323 Gbps/W all-digital full-entropy true random number generator in 14 nm FinFET CMOS," *IEEE J. Solid-State Circuits*, vol. 51, no. 7, pp. 1695–1704, 2016.

[47] T. K. Kuan, Y. H. Chiang, and S. I. Liu, "A 0.43pJ/bit true random number generator," in *IEEE Asian Solid-State Circuits Conference (A-SSCC)*, KaoHsiung, Taiwan, 2014, pp. 33–36.

[48] S. Tao and E. Dubrova, "TVL-TRNG: Sub-microwatt true random number generator exploiting metastability in ternary valued latches," in *IEEE 47th International Symposium on Multiple-Valued Logic (ISMVL)*, Novi Sad, Serbia, 2017, pp. 130–135.

[49] H. Hata and S. Ichikawa, "FPGA implementation of metastability-based true random number generator," *IEICE Trans. Inf. & Syst.*, vol. E95-D, no. 2, pp. 426–436, 2012.

[50] P. Z. Wieczorek, "Dual-metastability FPGA-based true random number generator," *Electronics Letters*, vol. 49, no. 12, pp. 744–745, 2013.

[51] P. Z. Wieczorek and K. Golofit, "Dual-metastability time-competitive true random number generator," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 61, no. 1, pp. 134–145, 2014.

[52] C. Li, Q. Wang, J. Jiang, and N. Guan, "A metastability-based true random number generator on FPGA," in *IEEE 12th International Conference on ASIC (ASICON)*, Guiyang, China, 2017, pp. 738–741.

[53] C. S. Petrie and J. A. Connelly, "A noise-based random bit generator IC for applications in cryptography," in *Proceedings of the 1998 IEEE International Symposium on Circuits and Systems (ISCAS '98)*, Monterey, CA, 1998, pp. 197–200.

[54] M. E. Yalcin, J. A. K. Suykens, and J. Vandewalle, "True random bit generation from a double-scroll attractor," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 51, no. 7, pp. 1395–1404, 2004.

[55] M. Drutarovsky and P. Galajda, "A robust chaos-based true random number generator embedded in reconfigurable switched-capacitor hardware," in *17th International Conference Radioelektronika*, Brno, Czech Republic, 2007.

[56] F. Pareschi, G. Setti, and R. Rovatti, "Implementation and testing of high-speed CMOS true random number generators based on chaotic systems," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 57, no. 12, pp. 3124–3137, 2010.

[57] I. Cicek, A. E. Pusane, and G. Dundar, "A new dual entropy core true random number generator," *Analog Integr. Circ. Sig. Process.*, vol. 81, no. 1, pp. 61–70, 2014.

[58] I. Cicek, A. E. Pusane, and G. Dundar, "A novel design method for discrete time chaos based true random number generators," *Integration, the VLSI Journal*, vol. 47, no. 1, pp. 38–47, 2014.

[59] N. Jiteurtragool, C. Wannaboon, and T. Masayoshi, "True random number generator based on compact chaotic oscillator," in *15th International Symposium on Communications and Information Technologies (ISCIT)*, Nara, Japan, 2015, pp. 315–318.

[60] M. Park, J. C. Rodgers, and D. P.Lathrop, "True random number generation using CMOS Boolean chaotic oscillators," *Microelectronics Journal*, vol. 46, no. 12, pp. 1364–1370, 2015.

[61] I. Koyuncu and A. T. Ozcerit, "The design and realization of a new high speed FPGA-based chaotic true random number generator," *Computers & Electrical Engineering*, vol. 58, pp. 203–214, 2017.

[62] P. Z. Wieczorek and K. Golofit, "True random number generator based on flip-flop resolve time instability boosted by random chaotic source," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 65, no. 4, pp. 1279–1292, 2018.

[63] *Security Requirements for Cryptographic Modules: FIPS PUB 140-2*, Federal Information Processing Standards, National Institute of Standards and Technology, Gaithersburg, MD, 2001.

[64] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray, and S. Vo, *Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications: Special Publication 800-22*, National Institute of Standards and Technology, Gaithersburg, MD, 2010.

[65] G. Marsaglia, *The Marsaglia Random Number CDROM including the Diehard Battery of Tests of Randomness*, Florida State University, 1995.

[66] Y. Wang, Y. Wang, and L. He, "Behavioral modeling for operational amplifier in sigma-delta modulators with Verilog-A," in *IEEE Asia Pacific Conference on Circuits and Systems (APCCAS 2008)*, Macao, China, 2008, pp. 1612–1615.

[67] A. Spitalny and M. J. Goldberg, "On-line graphics applied to layout design of integrated circuits," *Proceedings of the IEEE*, vol. 55, no. 11, pp. 1982–1988, 1967.

[68] D. K. Lynn, "Computer-aided layout system for integrated circuits," *IEEE Trans. on Circ. Theory*, vol. 18, no. 1, pp. 128–139, 1971.

[69] *Virtuoso® Layout Suite XL User Guide: Product Version IC6.1.7*, Cadence Design Systems, Inc., San Jose, CA, 2016.

[70] *Pyxis™ Layout User's Manual for the Pyxis Custom Design Platform: Software Version 17.1*, Mentor Graphics Corporation, Wilsonville, OR, 2017.

[71] *Custom Compiler™ Layout Editor User Guide: Version M-2017.03*, Synopsys, Inc., Mountain View, CA, 2017.

[72] R. Martins, N. Lourenco, and N. Horta, *Analog Integrated Circuit Design Automation*. Cham, Switzerland: Springer, 2017.

[73] J. Scheible and J. Lienig, "Automation of analog IC layout – challenges and solutions," in *Proceedings of the 2015 Symposium on International Symposium on Physical Design (IESM)*, Monterey, CA, 2015, pp. 33–40.

[74] R. C. Johnson, "Analog eda finally automated," *EETimes*, 2015. [Online]. Available from: https://www.eetimes.com/document.asp?doc_id=1326192

[75] E. Barke, "A network comparison algorithm for layout verification of integrated circuits," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 3, no. 2, pp. 135–141, 1984.

[76] W. H. Chang, S. D. Tzeng, and C. Y. Lee, "A novel subcircuit extraction algorithm by recursive identification scheme," in *The 2001 IEEE International Symposium on Circuits and Systems (ISCAS 2001)*, Sydney, NSW, Australia, 2001, pp. 491–494.

[77] N. Zhang and D. C. W. II, "Speeding up VLSI layout verification using fuzzy attributed graphs approach," *IEEE Trans. Fuzzy Syst.*, vol. 14, no. 6, pp. 728–737, 2006.

[78] A. Turgeman and J. Katzenelson, "EC-PI: an integrated circuit layout program in the enhanced C language," in *The Sixteenth Conference of Electrical and Electronics Engineers in Israel*, Tel-Aviv, Israel, 1989, pp. 1–4.

[79] N. Lourenco, M. Vianello, J. Guilherme, and N. Horta, "LAYGEN – automatic layout generation of analog ICs from hierarchical template descriptions," in *Ph.D. Research in Microelectronics and Electronics (PRIME)*, Otranto, Italy, 2006, pp. 213–216.

[80] R. Martins, N. Lourenco, and N. Horta, "LAYGEN II – automatic layout generation of analog integrated circuits," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 32, no. 11, pp. 1641–1654, 2013.

[81] I. Lomeli-Illescas, S. A. Solis-Bustos, V. H. Martínez-Sánchez, and J. E. Rayas-Sánchez, "Synthesis tool for automatic layout generation of analog structures," in *IEEE ANDESCON*, Arequipa, Peru, 2016.

[82] R. Martins, N. Lourenco, and N. Horta, "Analog IC placement using absolute coordinates and a hierarchical combination of pareto optimal fronts," in *Ph.D. Research in Microelectronics and Electronics (PRIME)*, Glasgow, UK, 2015, pp. 61–64.

[83] M. Eick, M. Strasser, K. Lu, U. Schlichtmann, and H. E. Graeb, "Comprehensive generation of hierarchical placement rules for analog integrated circuits," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 30, no. 2, pp. 180–193, 2011.

[84] H. Habal and H. Graeb, "Constraint-based layout-driven sizing of analog circuits," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 30, no. 8, pp. 1089–1102, 2011.

[85] D. K. Schroder, "Negative bias temperature instability: What do we understand?" *Microelectronics Reliability*, vol. 47, no. 6, pp. 841–852, 2006.

[86] J. F. Zhang and W. Eccleston, "Positive bias temperature instability in MOSFET's," *IEEE Trans. Electron Devices*, vol. 45, no. 1, pp. 116–124, 1998.

[87] *Virtuoso® Schematic Editor User Guide: Product Version 5.1.41*, Cadence Design Systems, Inc., San Jose, CA, 2008.

[88] *Virtuoso® Schematic Editor XL User Guide: Product Version IC6.1.7*, Cadence Design Systems, Inc., San Jose, CA, 2016.

[89] *Virtuoso® Spectre® Circuit Simulator Reference: Product Version 15.1*, Cadence Design Systems, Inc., San Jose, CA, 2016.

[90] *Cadence® Verilog®-A Language Reference: Product Version 6.1*, Cadence Design Systems, Inc., San Jose, CA, 2006.

[91] *Virtuoso® Layout Editor User Guide: Product Version 5.1.41*, Cadence Design Systems, Inc., San Jose, CA, 2008.

[92] *Virtuoso® Spectre® Layout Suite SKILL Reference: Product Version IC6.1.7*, Cadence Design Systems, Inc., San Jose, CA, 2016.

[93] *Cadence SKILL IDE User Guide: Product Version IC6.1.7*, Cadence Design Systems, Inc., San Jose, CA, 2016.

[94] *Calibre® Interactive™ and Calibre® RVE™ User's Manual: Software Version 2015.3*, Mentor Graphics Corporation, Wilsonville, OR, 2015.

[95] *Star-RCXT™ User Guide: Version B-2008.12*, Synopsys, Inc., Mountain View, CA, 2008.

[96] *MATLAB® Data Analysis*, The MathWorks, Inc., Natick, MA, 2018.

[97] M. Bucci and R. Luzzi, "Design of testable random bit generators," in *Cryptographic Hardware and Embedded Systems – CHES 2005*, J. R. Rao and B. Sunar, Eds. Berlin, Heidelberg: Springer, 2015, pp. 147–156.

[98] V. Kotě, "True random number generator," Master's thesis, Czech Technical University in Prague, 2011.

[99] R. B. Davies, *Exclusive OR (XOR) and hardware random number generators*, 2002. [Online]. Available from: http://www.robertnz.net/pdf/xor2.pdf

[100] J. von Neumann, "Various techniques used in connection with random digits," *J. Res. Nat. Bur. Stand. Appl. Math. Series 3*, pp. 36–38, 1951.

[101] B. Barak, R. Shaltiel, and E. Tromer, "True random number generators secure in a changing environment," in *Cryptographic Hardware and Embedded Systems – CHES 2003*, C. D. Walter, Ç. K. Koç, and C. Paar, Eds. Berlin, Heidelberg: Springer, 2003, pp. 166–180.

[102] S. H. Kwok, Y. L. Ee, G. Chew, K. Zheng, K. Khoo, and C. H. Tan, "A comparison of post-processing techniques for biased random number generators," in *Information Security Theory and Practice. Security and Privacy of Mobile Devices in Wireless Communication (WISTP 2011)*, C. A. Ardagna and J. Zhou, Eds. Berlin, Heidelberg: Springer, 2011, pp. 175–190.

[103] A. T. Markettos and S. W. Moore, "The frequency injection attack on ring-oscillator-based true random number generators," in *Cryptographic Hardware and*

Embedded Systems – CHES 2009, C. Clavier and K. Gaj, Eds. Berlin, Heidelberg: Springer, 2009, pp. 317–331.

[104] J. B. Johnson, "Thermal agitation of electricity in conductors," *Physical Review*, vol. 32, pp. 97–109, 1928.

[105] H. Nyquist, "Thermal agitation of electric charge in conductors," *Physical Review*, vol. 32, pp. 110–113, 1928.

[106] W. M. C. Sansen, *Analog Design Essentials*. Boston, MA: Springer, 2006.

[107] B. Razavi, *Design of Analog CMOS Integrated Circuits*. New York, NY: McGraw-Hill, 2001.

[108] F. N. Hooge, "$1/f$ noise source," *IEEE Trans. Electron Devices*, vol. 41, no. 11, pp. 1926–1935, 1994.

[109] P. Horowitz and W. Hill, *The Art Of Electronics*, 2nd ed. Cambridge, UK: Cambridge University Press, 1989.

[110] L. K. J. Vandamme and H. J. Casier, "The 1/f noise versus sheet resistance in poly-Si is similar to poly-SiGe resistors and Au-layers," in *Proceedings of the 30th European Solid-State Circuits Conference*, Leuven, Belgium, 2004, pp. 365–368.

[111] R. Sarpeshkar, T. Delbruck, and C. A. Mead, "White noise in MOS transistors and resistors," *IEEE Circuits Devices Mag.*, vol. 9, no. 6, pp. 23–29, 1993.

[112] Y. Tsividis, *Mixed Analog-Digital VLSI Devices and Technology*. Singapore: World Scientific Publishing, 2002.

[113] C. Leyris, S. Pilorget, M. Marin, M. Minondo, and H. Jaouen, "Random telegraph signal noise SPICE modeling for circuit simulators," in *ESSDERC 2007 - 37th European Solid State Device Research Conference*, Munich, Germany, 2007, pp. 187–190.

[114] B. M. Wilamowski and J. D. Irwin, *Fundamentals of Industrial Electronics*, 2nd ed. Boca Raton, FL: CRC Press, 2011.

[115] N. Tega, H. Miki, F. Pagette, D. J. Frank, A. Ray, M. J. Rooks, W. Haensch, and K. Torii, “Increasing threshold voltage variation due to random telegraph noise in FETs as gate lengths scale to 20 nm,” in *Symposium on VLSI Technology*, Honolulu, HI, 2009, pp. 50–51.

[116] T. Figliolia, P. Julian, G. Tognetti, and A. G. Andreou, “A true random number generator using RTN noise and a sigma delta converter,” in *IEEE International Symposium on Circuits and Systems (ISCAS)*, Montreal, Canada, 2016, pp. 17–20.

[117] M. E. Hines, “Noise theory for the read type avalanche diode,” *IEEE Trans. Electron Devices*, vol. ED-13, no. 1, pp. 158–163, 1966.

[118] B. Lampert, R. S. Wahby, S. Leonard, and P. Levis, “Robust, low-cost, auditable random number generation for embedded system security,” in *SenSys ’16 Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems*, New York, NY, 2016, pp. 16–27.

[119] C. E. Shannon, “A mathematical theory of communication,” *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.

[120] S. Pincus and B. H. Singer, “Randomness and degrees of irregularity,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 93, no. 5, pp. 2083–2088, 1996.

[121] A. Hastings, *The Art of Analog Layout*. Upper Saddle River, NJ: Prentice-Hall, 2001.

[122] K. Yang, D. Fick, M. B. Henry, Y. Lee, D. Blaauw, and D. Sylvester, “A 23Mb/s 23pJ/b fully synthesized true-random-number generator in 28nm and 65nm CMOS,” in *IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC) 2014*, San Francisco, CA, 2014, pp. 280–281.

[123] S. Srinivasan, S. Mathew, R. Ramanarayanan, F. Sheikh, M. Anders, H. Kaul, V. Erraguntla, R. Krishnamurthy, and G. Taylor, “2.4GHz 7mW all-digital PVT-variation tolerant true random number generator in 45nm CMOS,” in *Symposium on VLSI Circuits*, Honolulu, HI, 2010, pp. 203–204.

[124] M. Simka, M. Drutarovsky, and V. Fischer, "Testing of PLL-based true random number generator in changingworking conditions," *Radioengineering*, vol. 20, no. 1, pp. 94–101, 2011.

[125] *IEEE Standard Verilog® Hardware Description Language*, IEEE Computer Society, Park Avenue, NY, 2001.

[126] *Design Compiler® User Guide*, Synopsys, Inc., Mountain View, CA, 2013.

[127] *IC Compiler™ Implementation User Guide*, Synopsys, Inc., Mountain View, CA, 2014.

[128] G. J. Croll, "BiEntropy – The approximate entropy of a finite binary string," *CoRR*, pp. 1–16, 2013.

[129] T. Goka, "An operator on binary sequences," *SIAM Review*, vol. 12, no. 2, pp. 264–266, 1970.

[130] V. Molata and V. Kotě, "Capacitor-less LDO regulator with low consumption," in *Proceedings of the International Student Scientific Conference Poster – 17/2013*, Prague, Czech Republic, 2013, pp. 1–5.

[131] V. Molata, V. Kotě, and J. Jakovenko, "Capacitor-less linear regulator with NMOS power transistor," *ElectroScope*, vol. 2013, no. 5, 2013.

[132] M. V. Barros, O. Possamai, L. V. O. D. Valentina, and M. A. Oliveira, "Analysis of time to market complexity:a case study of application of Bayesian networks as a forecasting tool," in *International Conference on Industrial Engineering and Systems Management (IESM)*, Seville, Spain, 2015.

[133] Y. Huh, K. Min, P. Bendix, V. Axelrad, R. Narayan, J.-W. Chen, L. D. Johnson, and S. H. Voldman, "Chip level layout and bias considerations for preventing neighboring I/O cell interaction-induced latch-up and inter-power supply latch-up in advanced CMOS technologies," in *Electrical Overstress/Electrostatic Discharge Symposium*, Tucson, AZ, 2005, pp. 1–8.

[134] M. Jianga, G. Fu, B. Wan, M. Jia, and Y. Qiu, "Research on single event latch-up effect of CMOS based on TCAD," in *The Second International Conference on Reliability Systems Engineering (ICRSE 2017)*, Beijing, China, 2017, pp. 1–5.

[135] B. Xu, S. Li, X. Xu, N. Sun, and D. Z. Pan, "Hierarchical and analytical placement techniques for high-performance analog circuits," in *Proceedings of the 2017 ACM on International Symposium on Physical Design*, Portland, OR, 2017, pp. 55–62.

[136] M. P. H. Lin, H. Zhang, M. D. F. Wong, and Y. W. Chang, "Thermal-driven analog placement considering device matching," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 30, no. 3, pp. 325–336, 2011.

[137] Y.-M. Sheu, K.-W. Su, S. Tian, S.-J. Yang, C.-C. Wang, M.-J. Chen, and S. Liu, "Modeling the well-edge proximity effect in highly scaled MOSFETs," *IEEE Trans. Electron Devices*, vol. 53, no. 11, pp. 2792–2798, 2006.

[138] H. C. Ou, K. H. Tseng, J. Y. Liu, I. P. Wu, and Y. W. Chang, "Layout-dependent effects-aware analytical analog placement," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 35, no. 8, pp. 1243–1254, 2016.

Appendix

List of Author's Publications . . . iii
Recognition and Review . . . vii

Appendix A: List of Author's Publications

A.1 Publications Related to the Topic of This Work

A.1.1 Publications in Impacted Journals

V. Kotě, P. Vacula, V. Molata, O. Veselý, O. Tláskal, D. Barri, J. Jakovenko, and M. Husák, "A true random number generator with time multiplexed sources of randomness," *Radioengineering*, 2018, (In press). Co-authorship: 53 %

V. Kotě, A. Kubačák, P. Vacula, J. Jakovenko, and M. Husák, "Automated pre-placement phase as a part of robust analog-mixed signal physical design flow," *Integration, the VLSI Journal*, 2018, (In press). Co-authorship: 35 %

A.1.2 Publications in Reviewed Journals

P. Vacula, V. Kotě, A. Kubačák, M. Lžíčař, R. Zelený, M. Husák, and J. Jakovenko, "Incremental control techniques for layout modification of integrated circuits," *Advances in Science, Technology and Engineering Systems Journal*, vol. 2, no. 3, pp. 1196–1201, 2017. Co-authorship: 40 %

V. Kotě, V. Molata, and J. Jakovenko, "Enhanced generic architecture for safety increase of true random number generators," *ElectroScope*, vol. 2014, no. 3, 2014. Co-authorship: 75 %

Cited in:

- G. J. Croll, "BiEntropy of knots on the simple cubic lattice," in *Unified Field Mechanics II: Formulations and Empirical Tests*, 2018, pp. 447–453.

V. Molata, V. Kotě, and J. Jakovenko, "Capacitor-less linear regulator with NMOS power transistor," *ElectroScope*, vol. 2013, no. 5, 2013. Co-authorship: 15 %

Cited in:

- N. Bansal and R. Gupta, "An NMOS low drop-out voltage regulator with -17dB wide-band power supply rejection for SerDes in 22FDX," in *31st International Conference on VLSI Design and 2018 17th International Conference on Embedded Systems (VLSID)*, Pune, India, 2018, pp. 341–346.

V. Kotě, V. Molata, and J. Jakovenko, "Improved structure of true random number generator with direct amplification of analog noise," *ElectroScope*, vol. 2012, no. 6, 2012. Co-authorship: 80 %

A.1.3 Publications Excerpted by WoS

P. Vacula, V. Kotě, A. Kubačák, M. Lžíčař, R. Zelený, M. Husák, and J. Jakovenko, "Incremental control techniques for layout modification," in *11th International Conference on Advanced Semiconductor Devices & Microsystems (ASDAM)*, Smolenice, Slovakia, 2016, pp. 239–242. Co-authorship: 40 %

A.1.4 Other Publications

V. Kotě, V. Molata, and P. Vacula, "Behavioral models of true random number generators," in *Proceedings of the International Student Scientific Conference Poster – 22/2018*, Prague, Czech Republic, 2018, pp. 1–6. Co-authorship: 60 %

P. Vacula, V. Kotě, A. Kubačák, S. Cliquennois, M. Lžíčař, M. Husák, and J. Jakovenko, "Modern search techniques for layout creation," in *CDNLive Cadence User Conference EMEA 2017*, Munich, Germany, 2017. Co-authorship: 43 %

P. Vančura, V. Kotě, P. Vacula, A. Kubačák, and J. Jakovenko, "Matched structure classification," in *CDNLive Cadence User Conference EMEA 2017*, Munich, Germany, 2017. Co-authorship: 12.5 %

P. Vacula, V. Kotě, A. Kubačák, M. Lžíčař, R. Zelený, M. Husák, and J. Jakovenko, "Modern control techniques for layout creation," in *CDNLive Cadence User Conference EMEA 2016*, Munich, Germany, 2016. Co-authorship: 40 %

V. Kotě, V. Molata, and J. Jakovenko, "New structure of true random number generators with protective mechanisms," in *Electronic Devices and Systems IMAPS CS International Conference 2014*, Brno, Czech Republic, 2014, pp. 19–24. Co-authorship: 80 %

V. Molata, V. Kotě, T. Nápravník, and J. Jakovenko, "Capacitor-less LDO regulator with NMOS power transistor," in *Electronic Devices and Systems IMAPS CS International Conference 2013*, Brno, Czech Republic, 2013, pp. 50–55. Co-authorship: 20 %

V. Molata and V. Kotě, "Capacitor-less LDO regulator with low consumption," in *POSTER 2013 – 17th International Student Conference on Electrical Engineering*, Prague, Czech Republic, 2013, pp. 1–5. Co-authorship: 20 %

V. Kotě, T. Nápravník, V. Molata, and J. Jakovenko, "Structure, modeling and realization of true random number generator with analog noise amplification," in *Proceedings of Electronic Devices and Systems EDS 2012*, Brno, Czech Republic, 2012, pp. 145–150. Co-authorship: 70 %

V. Molata, V. Kotě, T. Nápravník, and J. Jakovenko, "Capacitor-less LDO regulator in CMOS technology," in *Proceedings of Electronic Devices and Systems EDS 2012*, Brno, Czech Republic, 2012, pp. 54–59. Co-authorship: 10 %

A.1.5 Functional Samples

V. Kotě, V. Molata, P. Vacula, and J. Jakovenko, Fully Integrated Pipelined Noise Source Based on Metastability for True Random Number Generators. Functional sample. 2014. Co-authorship: 75 %

A.2 Publications Not Related to the Topic of This Work

A.2.1 Patents

P. Vacula, M. Vacula, V. Kotě, A. Kubačák, and M. Lžíčař, Transistors with dissimilar square waffle gate pattern. Patent application. 2016. Co-authorship: 30 %

A.2.2 Publications Excerpted by Scopus

T. Nápravník, V. Kotě, V. Molata, and J. Jakovenko, "Differential Evolutionary Optimization Algorithm Applied to ESD MOSFET Model Fitting Problem," in *IEEE 15th International Symposium on Design and Diagnostics of Electronic Circuits & Systems (DDECS)*, Tallinn, Estonia, 2012, pp. 155–158. Co-authorship: 10 %

A.2.3 Other Publications

P. Vacula, V. Kotě, and D. Barri, "Trench MOS Having Source with Waffle Patterns," in *Proceedings of the International Student Scientific Conference Poster – 22/2018*, Prague, Czech Republic, 2018, pp. 1–5. Co-authorship: 20 %

T. Nápravník, V. Kotě, V. Molata, and J. Jakovenko, "Utilization of Differential Evolutionary Optimization Algorithm for ESD MOSFET Model Fitting," in *Proceedings of Electronic Devices and Systems EDS 2012*, Brno, Czech Republic, 2012, pp. 33–38. Co-authorship: 10 %

A.2.4 Functional Samples

V. Molata, V. Kotě, P. Vacula, and J. Jakovenko, Integrated single-ended low-power ramp generator for DC-DC converters. Functional sample. 2014. Co-authorship: 15 %

T. Nápravník, V. Kotě, P. Vacula, and J. Jakovenko, Single-finger and multi-finger MOST structures for ESD characterization and model development. Functional sample. 2014. Co-authorship: 10 %

P. Vacula, T. Jeřábek, V. Molata, V. Kotě, J. Jakovenko, and M. Husák, Low specific on-resistance MOSFET with waffle gate pattern for power management. Functional sample. 2014. Co-authorship: 5 %

Appendix B: Recognition and Review

IEEE MTT/AP/ED/EMC Joint Chapter awarded Diploma for the best C+EI paper "Behavioral Models of True Random Number Generators presented at Poster 2018." *22nd International Student Conference on Electrical Engineering POSTER 2018*. Prague, Czech Republic, May 10, 2018.

Program Committee of POSTER 2018 awarded Certificate of Achievement in the Best Poster Competition for the work "Behavioral Models of True Random Number Generators." *22nd International Student Conference on Electrical Engineering POSTER 2018*. Prague, Czech Republic, May 10, 2018.

Certificate of Participation for the contribution "Modern Search Techniques for Layout Creation." *CDNLive Cadence User Conference EMEA 2017*. Munich, Germany, May 15–17, 2017.

Program Committee of POSTER 2013 awarded Diploma for the work "Capacitor-Less LDO Regulator With Low Consumption." *17nd International Student Conference on Electrical Engineering POSTER 2013*. Prague, Czech Republic, May 16, 2013.

Best Ph.D. Poster Award – 1st Place for the work "Structure, modeling and realization of true random number generator with analog noise amplification." *IMAPS Electronic Devices and Systems International Conference 2012*. Brno, Czech Republic, June 29, 2012.

Review for Advances in Science, Technology and Engineering Systems Journal