DESIGN ASPECTS OF DISCRETE TIME CHAOS BASED TRUE RANDOM NUMBER GENERATORS

by

İhsan Çiçek

B.S., Electronics and Telecommunication Engineering,

İstanbul Technical University, 2002

M.Sc., Electronics Engineering, Sabancı University, 2004

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

Graduate Program in Electrical and Electronics Engineering

Boğaziçi University

2014

DESIGN ASPECTS OF DISCRETE TIME CHAOS BASED
TRUE RANDOM NUMBER GENERATORS

APPROVED BY:

Prof. Günhan Dündar ....
(Thesis Supervisor)

Assoc. Prof. Arda Deniz Yalçınkaya .....

Assoc. Prof. Müştak Yalçın .....

Assoc. Prof. Sıddıka Berna Örs Yalçın .....

Assist. Prof. Ali Emre Pusane .....

DATE OF APPROVAL: 11.06.2014

ACKNOWLEDGEMENTS

There are many people to whom I would like to thank for their support and encouragement in completion of this thesis. First of all, all my gratitude goes to my supervisor Prof. Günhan Dündar. He has guided my PhD research at Boğaziçi University for about eight years. Over these years, he has truly inspired me during my research with his innovative ideas and, provided helpful hints and always supported me through the good and the bad times. So, it has been an honor and pleasure to work with him. In addition, I also would like to thank Assist. Prof. Ali Emre Pusane for always being helpful and encouraging me. My thanks also go to my respectable thesis jury members: Prof. Ercan Solak, Assoc. Prof. Arda Yalçınkaya, and Assoc. Prof. Müştak Yalçın, and Assoc. Prof. Sıddıka Berna Örs Yalçın for spending their rarely available time on inspecting my thesis.

I am indebted to TÜBİTAK-BİLGEM for not only motivating its employees to pursue an academic career, but also for providing research equipment and scientific research expenditures. Also, my huge thanks go to my project managers Dr. Fatih Kara and Hikmet Aşmer for giving me an opportunity to complete my PhD study, consistently encouraging me to go on working and giving valuable advises. My thanks also go to my friends Mustafa Parlak, Tayyar Güzel, Özgür Ören, İmran Ergüler, Ülkühan Güler, Recep Küyük and my colleagues for the pleasant and productive environment and their long lasting friendship.

This thesis could not have been carried out without the financial support kindly provided by Boğaziçi University Research Projects Fund. I would like to express my gratitude to institution for supporting my research.

Additionally, to many thanks go to the distinguished members of the Beta Lab: Okan Batur, Baykal Sarıoğlu, and Engin Afacan for being always there to help me, and for opening the doors of their humble house during the chip design process, Berk

Çamlı for the Latex related life saving help, Ahmet Unutulmaz for kindly sharing his cubicle with me, and all others for their hospitality.

Last but not least, I would like to express my deepest gratitude to my loving parents, my sister, my aunt, my mother and father in law for their unconditional support and patience. They have always encouraged me when I took important decisions and always been with me when I needed them. Finally, my deepest thanks go to my wife Betül, and my lovely son M. Ahmet for their love, patience, and enduring support. Without their patience and encouragement, this thesis would never have been created.

I would like to dedicate this thesis to the loving memory of my uncle, the true hero of my life, M. İlhan Öktem (1938 - 2013), a civil engineer who saved the lives of hundreds in the earthquake of 17th August 1999 with his properly constructed buildings in Yalova. No words are sufficient to describe his inspiration, and contribution to my life.

ABSTRACT

DESIGN ASPECTS OF DISCRETE TIME CHAOS BASED TRUE RANDOM NUMBER GENERATORS

In this thesis, design aspects of discrete time chaos based cryptographic grade TRNGs are studied starting from chaotic map equations to in depth analyses of the generated entropy. Custom mathematical models of discrete time chaos based TRNGs are developed to predict the randomness performance ahead of the physical implementation. A practical information measure, T-entropy, is used to characterize the entropy capacity of discrete time chaos based TRNGs, since conventional statistical tests can only provide pass/fail type binary outputs. Maximum allowable parameter variation boundaries for hardware design are determined using T-entropy calculations. A new dual entropy core TRNG architecture is introduced along with its mathematical model. The superiority of the proposed architecture over conventional single entropy core TRNG architecture is presented through a comparative study of generated entropy, and its sensitivity to parameter variations. A proof of concept novel dual entropy core discrete time chaos based TRNG circuit is implemented on a reconfigurable analog platform, and measurement results are presented. A novel integrated dual entropy core discrete time chaos based TRNG circuit is designed, and implemented in 180nm CMOS technology using a new, matching driven design methodology, and the measurement results of the prototype chip are presented.

ÖZET

AYRIK ZAMANLI KAOS TABANLI GERÇEK RASTSAL SAYI ÜRETEÇLERİNİN TASARIM İNCELİKLERİ

Bu tezde kriptografik ayarda ayrık zaman kaotik gerçek rastsal sayı üreteçlerinin tasarım incelikleri kaotik denklemlerden başlayıp üretilen entropinin derin analizine uzanarak incelenmiştir. Fiziksel gerçeklemeden önce rastsallık performanslarını öngörebilmek amacıyla ayrık zaman kaotik gerçek rastsal sayı üreteçleri için özel matematiksel modeller geliştirilmiştir. Geleneksel istatistiksel testlerin ikilik geçti/kaldı tipi sonuçları incelenen sistemlerin entropi kapasitelerini karakterize edemediği için pratik bir bilgi kuramı ölçütü olan T-entropi, bu amaçla kullanılmıştır. Donanım tasarımı için izin verilebilir maksimum parametre saçılma sınırları T-entropi ölçütü hesaplamaları kullanılarak belirlenmiştir. Yeni bir çift entropi kaynaklı ayrık zaman kaotik gerçek rastsal sayı üreteç mimarisi matematiksel modeliyle birlikte literatüre tanıtılmıştır. Önerilen yeni mimarinin tek entropi kaynaklı ayrık zaman kaotik gerçek rastsal üreteçlerine göre üstünlükleri üretilen entropi ve kontrol parametre hassasiyetleri perspektifinden karşılaştırmalı olarak sunulmuştur. Yeni çift entropi kaynaklı ayrık zaman kaotik gerçek rastsal sayı üreteç mimarisi, prototip olarak sahada programlanabilir analog diziler üzerinde gerçeklenerek ölçüm sonuçları verilmiştir. Yeni bir çift entropi kaynaklı ayrık zaman kaotik gerçek rastsal sayı üreteç tümdevresi $180nm$ CMOS teknolojisinde eşleştirmeye dayanan yeni bir yaklaşımla tasarlanarak gerçeklenmiş, ve üretilen tümdevrenin ölçüm sonuçları sunulmuştur.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS iii
ABSTRACT v
ÖZET vi
LIST OF FIGURES x
LIST OF TABLES xvi
LIST OF SYMBOLS xviii
LIST OF ACRONYMS/ABBREVIATIONS xxi
1. INTRODUCTION 1
1.1. Classification of TRNGs 3
1.1.1. Amplified Noise Sampling Based TRNGs 4
1.1.2. Multiple Oscillator Sampling Based TRNGs 5
1.1.3. Chaotic Signal Sampling Based TRNGs 6
1.2. Randomness Performance Evaluation of TRNGs 8
1.3. Contributions 9
1.4. Publication List 11
1.5. Organization of the Thesis 12
2. CHARACTERIZATION OF CHAOTIC ENDOMORPHIC MAPS FOR TRNG APPLICATIONS 14
2.1. Logistic Map 18
2.1.1. Dynamic Characteristics of the Logistic Map 18
2.1.2. Spectral Characteristics of the Logistic Map 20
2.1.3. Statistical Characteristics of the Logistic Map 22
2.2. Tent Map 25
2.2.1. Dynamic Characteristics of the Tent Map 26
2.2.2. Spectral Characteristics of the Tent Map 28
2.2.3. Statistical Characteristics of the Tent Map 29
2.3. Bernoulli Map 31
2.3.1. Dynamic Characteristics of the Bernoulli Map 32

2.3.2. Spectral Characteristics of the Bernoulli Map 34
2.3.3. Statistical Characteristics of the Bernoulli Map 35
3. THEORETICAL ASPECTS AND MATHEMATICAL MODELING OF DT CHAOS BASED TRNGS 39
3.1. Mathematical Modeling of Single Entropy Core DT Chaos Based TRNGs 40
3.1.1. Calculation of the Optimum Threshold for Identically Distributed Bit Generation 42
3.1.2. Calculation of the Optimum Threshold for Independent Bit Generation 47
3.2. Statistical Testing of Single Entropy Core DT Chaos Based TRNG . . 50
3.3. Mathematical Modeling of Dual Entropy Core DT Chaos Based TRNGs 52
3.4. Statistical Testing of Dual Entropy Core DT Chaos Based TRNG . . . 55
4. RANDOMNESS PERFORMANCE EVALUATION USING A PRACTICAL INFORMATION MEASURE 57
4.1. Calculation of the Entropy of a Finite Bit Stream Using T-Complexity 58
4.2. Randomness Performance Evaluation of Single Entropy Core TRNG . . 61
4.3. Randomness Performance Evaluation of Dual Entropy Core TRNG . . 66
5. FPAA IMPLEMENTATION OF DT CHAOS BASED TRNGS 69
5.1. FPAA Implementation of a Single Entropy Core DT Chaos Based TRNG 70
5.1.1. Measurement Results 71
5.1.2. Statistical Test Results 72
5.2. FPAA Implementation of a Dual Entropy Core DT Chaos Based TRNG 73
5.2.1. Measurement Results 74
5.2.2. Statistical Test Results 76
6. ASIC IMPLEMENTATION OF DT CHAOS BASED DUAL ENTROPY CORE TRNGS 78
6.1. Design of the Entropy Cores 80
6.1.1. Minimalist Circuit Implementation of the Bernoulli Map Function 83
6.1.2. Improved Circuit Implementation of the Bernoulli Map Function 88
6.1.3. Minimalist Circuit Implementation of the Tent Map Function . 91
6.1.4. Improved Circuit Implementation of the Tent Map Function . . 95

6.2. Design of the Sample and Hold Circuit 98
6.3. Design of the Current Mode Comparator for Random Bit Generation . 102
6.4. Integrated Circuit Implementation 106
7. CHIP MEASUREMENTS AND STATISTICAL TEST RESULTS 108
7.1. Test Fixture Board of the Prototype Integrated Circuit 108
7.2. Measurement Setup and Data Acquisition System 110
7.3. Measurement Results of the Prototype Integrated Circuit 111
7.3.1. Measurement Results of the Bernoulli Map Entropy Core 111
7.3.2. Measurement Results of the Tent Map Entropy Core 112
7.3.3. Measurement Results of the Dual Entropy Core TRNG 114
7.4. Statistical Test Results of the Dual Entropy Core TRNG 115
8. CONCLUSIONS AND FUTURE WORK 117
8.1. Summary 117
8.2. Future Work 119
REFERENCES 120

LIST OF FIGURES

Figure 1.1. Universal architecture of a typical true random number generator. 3

Figure 1.2. A unified classification of TRNGs according to entropy source, number generation method, and implementation technology. . . . 4

Figure 1.3. Electrical noise based TRNGs 5

Figure 1.4. Jitter based TRNG. 5

Figure 1.5. Chaos based TRNG. 7

Figure 2.1. Logistic map function. 18

Figure 2.2. Bifurcation, and Lyapunov exponent diagram of the logistic map. 19

Figure 2.3. Power spectral density of the logistic map generated time series. . 21

Figure 2.4. Autocorrelation of the logistic map generated time series. 21

Figure 2.5. Histogram based probability mass function, and its distribution fit. 24

Figure 2.6. Tent map function. 26

Figure 2.7. Bifurcation, and Lyapunov exponent diagram of the tent map. . . 27

Figure 2.8. Power spectral density of the tent map generated time series. . . . 28

Figure 2.9. Autocorrelation of tent map generated time series. 29

Figure 2.10. Histogram based probability mass function, and its distribution fit. 30

Figure 2.11. Bernoulli map function. 32

Figure 2.12. Bifurcation, and Lyapunov exponent diagram of the Bernoulli map. 33

Figure 2.13. Power spectral density of the Bernoulli map generated time series. 34

Figure 2.14. Autocorrelation of the Bernoulli map generated time series. 35

Figure 2.15. Histogram based probability mass function, and its distribution fit. 36

Figure 2.16. A comparative outlook of common chaotic maps, and their characteristics. 38

Figure 3.1. Single entropy core DT chaos based TRNG model. 41

Figure 3.2. Random number generation using the phase portrait of skew tent map. 42

Figure 3.3. Unified 3D projection of bifurcation, and statistical characteristics of the trajectories generated by skew tent map. 44

Figure 3.4. Partitioning of the PDF with respect to a threshold parameter. . . 44

Figure 3.5. Entropy as a function of bit generation threshold, T_h. 46

Figure 3.6. Dual entropy core DT chaos based TRNG Model. 52

Figure 3.7. Probability density function of the composite random variable Y. . 55

Figure 4.1. T-entropy of the bitstream generated by logistic map based single entropy core TRNG model, and calculated maximum allowable deviation boundaries for an entropy loss of 0.01. 63

Figure 4.2. T-entropy of the bitstream generated by tent map based single entropy core TRNG model, and calculated maximum allowable deviation boundaries for an entropy loss of 0.01. 64

Figure 4.3. T-entropy of the bitstream generated by Bernoulli map based single entropy core TRNG model, and calculated maximum allowable deviation boundaries for an entropy loss of 0.01. 65

Figure 4.4. T-entropy of the bitstream generated by single entropy core TRNG model. 67

Figure 4.5. T-entropy of the bitstream generated by dual entropy core TRNG model. 67

Figure 4.6. Vertical Projection of the T-entropy plots for the bitstreams generated by single, and dual entropy core TRNG models. 68

Figure 5.1. FPAA implementation of logistic map based single entropy core TRNG. 70

Figure 5.2. Phase portrait of the logistic map implemented on FPAA. 71

Figure 5.3. FPAA implementation of Bernoulli map based dual entropy core TRNG. 73

Figure 5.4. Measurement setup for the dual entropy core DT chaos based TRNG. 74

Figure 5.5. Proof of concept implementation of the dual entropy core DT chaos based TRNG architecture. 75

Figure 5.6. Operation of the dual entropy core DT chaos based TRNG architecture implemented on FPAA. 76

Figure 6.1. Dual entropy core DT chaos based TRNG architecture. 80

Figure 6.2. Schematic of the minimalist Bernoulli map non linear function block. 83

Figure 6.3. Layout of the the minimalist Bernoulli map occupies $15 \times 15\mu m^2$. 86

Figure 6.4. DC transfer function of the Bernoulli map circuit without inverters. 87

Figure 6.5. Improved DC transfer function of the Bernoulli map circuit. . . . 87

Figure 6.6. Schematic of the improved Bernoulli map circuit. 88

Figure 6.7. Layout of the the improved Bernoulli map occupies $20 \times 30\mu m^2$. . 90

Figure 6.8. DC transfer function of the improved Bernoulli map circuit. 90

Figure 6.9. Schematic of the minimalist tent map non linear function block. . 91

Figure 6.10. Layout of the the minimalist tent map occupies $17 \times 14\mu m^2$. . . . 92

Figure 6.11. DC transfer function of the minimalist tent map circuit. 93

Figure 6.12. The effect of entropy reduction as a result of smooth slope transition at the discontinuity point of the tent map. 94

Figure 6.13. Schematic of the improved tent map non linear function block. . . 95

Figure 6.14. Layout of the the improved tent map occupies $29 \times 17\mu m^2$. 97

Figure 6.15. DC transfer function of the improved tent map circuit. 97

Figure 6.16. Schematic of the improved sample and hold circuit. 98

Figure 6.17. Layout of the sample and hold circuit occupies $23 \times 38\mu m^2$. 101

Figure 6.18. Transient operation of the improved Bernoulli map circuit. 101

Figure 6.19. Transient operation of the improved tent map circuit. 102

Figure 6.20. Schematic of comparator circuit. 103

Figure 6.21. Layout of the the current comparator circuit occupies $19 \times 16\mu m^2$. 104

Figure 6.22. DC transfer characteristics of the current mode comparator circuit. 105

Figure 6.23. DC transfer characteristics of the current mode comparator circuit. 105

Figure 6.24. Layout of the prototype integrated circuit occupies a die area of $1525 \times 1525\mu m^2$ designed using UMC $180nm$ CMOS technology. . 106

Figure 6.25. Microphotograph of the prototype integrated circuit fabricated in $180nm$ CMOS technology provided by UMC. 107

Figure 7.1. Custom designed test fixture board 109

Figure 7.2. Measurement and data acquisition system. 110

Figure 7.3. Time domain measurement results of the Bernoulli map entropy core. 112

Figure 7.4. Phase portrait measurement results of the Bernoulli map entropy core. 112

Figure 7.5. Time domain measurement results of the tent map entropy core. . 113

Figure 7.6. Phase portrait measurement results of the tent map entropy core. 113

Figure 7.7. Time domain measurement results of the integrated dual entropy core TRNG circuit. 114

Figure 7.8. Phase portrait measurement results of the integrated dual entropy core TRNG circuit. 115

LIST OF TABLES

Table 3.1. Joint and marginal bit generation probabilities for consecutively generated bits $\{b_n, b_{n+1}\}$ using skew tent map as the entropy source. 49

Table 3.2. NIST800.22 statistical test results for the bitstream generated by logistic map based single entropy core TRNG model. 50

Table 3.3. NIST800.22 statistical test results for the bitstream generated by tent map based single entropy core TRNG model. 51

Table 3.4. NIST800.22 statistical test results for the bitstream generated by Bernoulli map based single entropy core TRNG model. 51

Table 3.5. NIST800.22 statistical test results for the bitstream generated by Bernoulli map based dual entropy core TRNG model. 56

Table 4.1. Maximum allowable parameter tolerances for single entropy core DT chaos based TRNG models. 66

Table 5.1. NIST800.22 statistical test results for the bitstream generated by logistic map based single entropy core TRNG. 72

Table 5.2. NIST800.22 statistical test results for the bitstream generated by Bernoulli map based dual entropy core, TRNG. 77

Table 6.1. Device dimensions of the minimalist Bernoulli map circuit. 85

Table 6.2. Device dimensions of the improved Bernoulli map circuit. 89

Table 6.3. Device dimensions of the minimalist tent map circuit. 92

Table 6.4. Device dimensions of the improved tent map circuit. 96

Table 6.5. Device dimensions of the sample and hold circuit. 100

Table 6.6. Device dimensions of the comparator circuit. 104

Table 7.1. NIST800.22 statistical test results for the bitstream generated by integrated dual entropy core TRNG. 116

LIST OF SYMBOLS

ACC	DC accuracy
A_{K}	Technology proportinality constant of K
A_{VT}	Technology proportinality constant of V_T
b_n	Digital binary encoding of a chaotic real valued sample
$B(\alpha, \beta)$	Standard beta function
BW	Bandwidth
C	Capacitor or capacitance
C_{gs}	Gate-source capacitance of a transistor
C_{ox}	Oxide capacitance
C_{T}	T-complexity
$f_{\mathrm{e}}(x)$	Fraction of states that fall into a specific bin in histogram
$f_{\mathrm{x}}(x)$	Probability density function
$f_{\mathrm{x,x}}(x)$	Joint probability density function
f_{T}	Unity gain frequency of a transistor
$F_{\mathrm{x}}(x)$	Cumulative distribution function
g_m	Transconductance of a MOSFET
H	Shannon entropy
I_{D}	Drain current of a MOSFET
I_{j}	Current source with index j
I_{in}	Input current
I_{out}	Output current
I_{T}	T-information
k'	Transconductance parameter
k'_{n}	Transconductance parameter of an nMOS transistor
k'_{p}	Transconductance parameter of a pMOS transistor
K	Current gain of a MOSFET
L	Channel length of a MOSFET
L_{min}	Minimum channel length of a MOSFET

$Li(z)$	Logarithmic integral of z
Log_{n}	Logarithm function in base n
M_{j}	MOSFET with index j
$M(x)$	Map function
n	Index for state variable
N	Number of iterations or elements
p_{i}	string subpattern
P	Frobenius-Perron operator
P_{i}	Marginal probability of bit i
P_{ij}	Joint probability of consecutive bits (b_i, b_j)
r_{out}	Output resistance
R	Chaos control parameter of the logistic map
R_{xx}	Autocorrelation function
S	Subspace of a map function
t	Time
T_{N}	Trajectory of a map function for N iterations
T_{h}	Bit generation threshold
V_{GS}	Gate to source voltage of a MOSFET
V_{OV}	Overdrive voltage of a MOSFET
V_{T}	Threshold voltage of a MOSFET
$(W/L)_{\mathrm{i}}$	Aspect ratio of a MOSFET with index i
W	Channel width of a MOSFET
x_0	Initial condition of discrete time chaotic system
$x_{\mathrm{j,n}}$	State variable for j^{th} entropy core at iteration index n
x_{n}	State variable of a discrete time chaotic system
X_{j}	Random variable corresponding to x_n of jth entropy core
y_{n}	state variable of dual entropy core TRNG system
Y	Composite random variable corresponding to y_n

β	Chaos control parameter of the Bernoulli map
β_j	Non-overlapping j^{th} bin in histogram
δ	An infinitesimal change
δx_0	An infinitesimal separation in the initial conditions
ΔV_T	Variation of V_T
ΔK	Variation of K
η	Chaos control parameter
$\Delta\eta$	Variation of η
λ	Lyapunov Exponent
$\Psi(\hat{\alpha})$	Shape parameter estimator for beta distribution
$\Psi(\hat{\beta})$	Shape parameter estimator for beta distribution
μ	Chaos control parameter of the tent map
π	Pi number
Φ	Interval within the domain of a map function
Ψ_{ij}	Statistical independence metric for consecutive bits (b_i, b_j)
σ	Variance
τ	Time delay or lag
$\#$	Cardinality operator
∞	Infinity symbol
d/dx	Derivative operator
$\int$	Integral operator

LIST OF ACRONYMS/ABBREVIATIONS

1D	One-dimensional
3D	Three-dimensional
AIS	Application notes and interpretations
ASIC	Application specific integrated circuit
BW	Bandwidth
CAB	Configurable analog block
CDF	Cumulative distribution function
CLK	Clock signal
CMOS	Complementary metal-oxide semiconductor (technology)
CT	Continuous time
DC	Direct current
DT	Discrete time
ESD	Electrostatic discharge
FFT	Fourier transform
FIPS	Federal information processing standards
FP	Frobenius-Perron
FR4	Flame Retardant 4
FPAA	Field programmable analog array
FPGA	Field programmable gate array
IC	Integrated circuit
KHz	Kilohertz
KS	Kolmogorov-Sinai
ln	Natural logarithm
Mbps	Mega bits per second
MEMS	Micro-electro-mechanical system
MHz	Megahertz
MIM	Metal-insulator-metal (capacitor)

MOS	Metal-oxide semiconductor
MOSFET	Metal-oxide semiconductor field effect transistor
NIST	National institute of standards and technology
OPAMP	Operational amplifier
OTA	Operational transconductance amplifier
PCB	Printed circuit board
PDF	Probability density function
PMF	Probability mass function
PRNG	Pseudo random number generator
PSD	Power spectral density
PSRR	Power supply rejection ratio
RFID	Radio frequency Identification
RHPC	Recursive hierarchical pattern copying
RLPC	Recursive linear pattern copying
RNG	Random number generator
RO	Ring oscillator
RS232	EIA serial communication standard
SC	Switched-capacitor
SI	Switched-current
S/H	Sample and hold
Taug	T-augmentation
TRNG	True random number generator
UMC	United Microelectronics Corporation
VLSI	Very large scaled integration
XOR	Exclusive logic OR

1. INTRODUCTION

In many fields of engineering, and science, random numbers are used for specific purposes. A diverse spectrum of applications such as Monte Carlo simulations, statistics, gambling, and cryptology demand high quality random numbers. Since the validity of the application outcome depends on the quality of random numbers, the generation process is fundamentally important. Random number generation is basically producing a sequence of independent numbers with a specified distribution. A random number generator (RNG) can be defined as a computational, or a physical device which is designed to generate a stream of statistically independent numbers, or symbols. RNGs are classified as true, or pseudo, depending on the source of randomness.

Pseudo random number generators (PRNGs) are based on deterministic algorithms which can be implemented in software, or hardware form. PRNGs are capable of yielding long sequences of random numbers with good statistical properties, but these sequences are usually periodic, making them vulnerable to prediction. Progressive developments in computational systems prevent PRNGs from being a trusted source of randomness in information security related applications. Studies up to date agree on the fact that it is not possible to create true randomness from within a deterministic system. John von Neumann underlined this fact with his famous statement: "Anyone who considers arithmetical methods of producing random digits is, of course, in a state of sin" [1]. While PRNGs are low cost, and easy to integrate with digital systems, certain applications such as cryptology, lottery, and gambling need a source of true randomness.

True random number generators (TRNGs) are based on unpredictable physical phenomena such as electrical noise, radioactive decay, or atmosphere events. One of the most important application field of TRNGs is cryptology, in which strict quality measures are enforced to satisfy specific security requirements. True random numbers are used in cryptographic authentication protocols, initialization vectors, nonces,

padding values, encryption, and decryption keys. In cryptology, encryption process is defined to be the conversion of a plaintext into an unintelligible ciphertext, and decryption process is the inverse operation of encryption. A cipher is based on a group of algorithms that carry out both the encryption, and decryption processes. The operation of the cipher depends on the algorithms, and the cryptographic key, which is a secret parameter only known by communicating parties. In contemporary cryptology, ciphers are accepted to be publicly known, and the secrecy is constituted on cryptographic keys. Throughout the historical evolution of cryptology, the security paradigm has shifted from ciphers to keys as a result of the fact that many of the ciphers operating without keys have been broken. A cryptographic key is basically a string of symbols with a specific length determined by the cipher. It is the secret that authorized communicating parties must protect against potential threats. No one should be capable of predicting, or computing the key. For this reason, cryptographic keys are created using TRNGs. Accordingly, TRNGs are accepted as a vital building block of a cryptographic system, since no deterministic cryptographic function is capable of generating more randomness at the output than what is available at the input [2, 3]. As a consequence, the unpredictability, and the security of a cryptographic system depend heavily on the TRNG, rendering it as the most critical, and crucial component of a cryptographic system.

First examples of generating true random numbers by electrical means in the literature appeared in the early 70s. Pioneers of electronic TRNGs used radioactive emissions as the entropy source [4]. In the late 70s, discretely implemented true random bit generators using the avalanche noise of a reverse biased zener diode were introduced [5]. Similar discrete designs have appeared until early 80s [6–8]. With the evolution of integrated circuit technology, design trends in electronics started to drift towards silicon as a result of better performance opportunity with lower cost. The first integrated TRNG circuit was fabricated in 1984 [9]. The circuit was based on digital mixing of two oscillators. Since then many oscillator based TRNGs have used more, or less the same approach. In the late 80s, TRNG designs based on chaos started to appear in the literature [10–13].

Every TRNG employs one, or more non-deterministic, unpredictable source of entropy to produce random numbers. A typical TRNG architecture usually consists of three components presented in Figure 1.1: One, or more entropy source creating the random variable, a sampler for harvesting randomness from the entropy source, and a post-processor accounting for potential statistical bias.

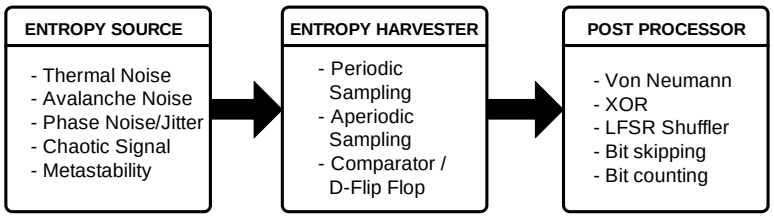


Figure 1.1. Universal architecture of a typical true random number generator.

In addition to these components, TRNGs can also incorporate a security monitor in practical applications for enhanced security. Critical system parameters such as the entropy of the generated bitstream, temperature, or power supplies are monitored against hardware failures, or side channel attacks. In the case of an anomaly, the host cryptographic system is informed for taking proper counter measures.

1.1. Classification of TRNGs

TRNGs can be classified according to the entropy source, number generation method, or implementation technology. A unified classification chart portraying most of the possible TRNG types is illustrated in Figure 1.2. A wide range of the TRNGs in the literature can be classified using the chart in Figure 1.2. The innermost circle of the chart presents the classification according to the source of randomness. The circle in the middle region of the chart shows the classification based on the method of randomness extraction, and classification with respect to the implementation type is shown by the outer circle of the chart.

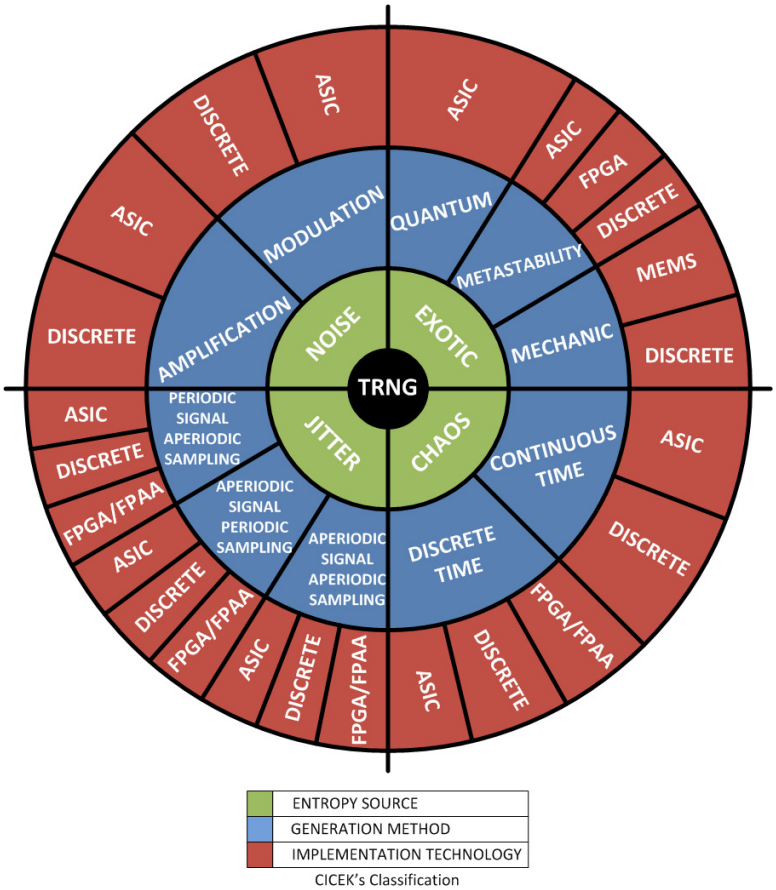


Figure 1.2. A unified classification of TRNGs according to entropy source, number generation method, and implementation technology.

While the chart in Figure 1.2 offers a diverse spectrum of classification possibilities, the source of randomness is the most important parameter that affects the performance. TRNGs can be classified into three major categories according to the source of randomness:

1.1.1. Amplified Noise Sampling Based TRNGs

Pioneers of electronic true random number generators exploited the unpredictable Brownian motion of electrons readily available as band-limited thermal noise in resistive components [14]. In succeeding variants, other semiconductor noise sources such as avalanche noise, shot noise, or 1/f noise were also considered as entropy sources. A common design feature of this type is that the generated noise is amplified to a level where it can be compared with a threshold to generate random bits as shown in

Figure 1.3. The main shortcoming of this approach is the limited throughput due to the band-limited entropy source. Inherent high sensitivity to interfering signals such as

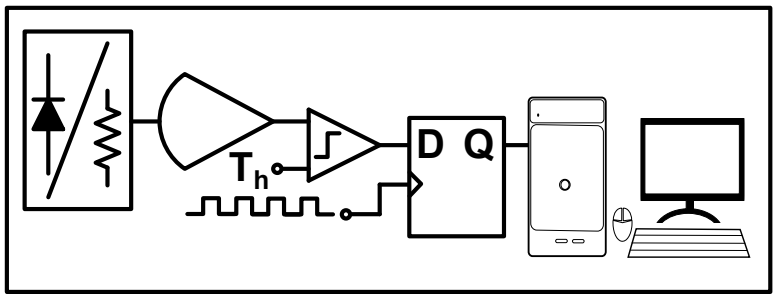


Figure 1.3. Electrical noise based TRNGs

substrate, and power rail coupled noise as a result of poor power supply rejection sets a fundamental limit on their integration capability with digital circuits [15]. Furthermore, post-processing units are required to deal with statistical deviations at the expense of reduced throughput.

1.1.2. Multiple Oscillator Sampling Based TRNGs

In order to meet the demand for high throughput TRNGs required by emerging high-speed secure computing, and communication applications, multiple oscillator based TRNGs were introduced. The architecture of such a TRNG is based on the sampling of a low-jitter fast oscillator, with a high-jitter slow oscillator as shown in Figure 1.4 [2].

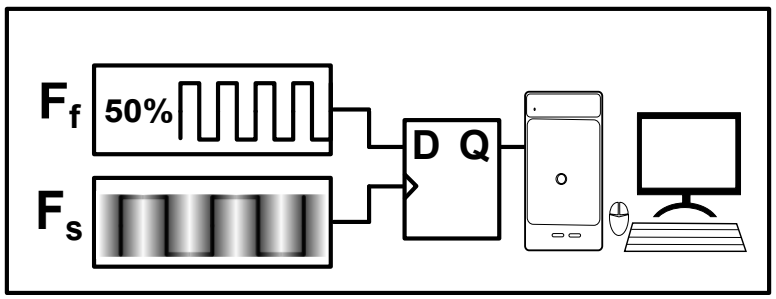


Figure 1.4. Jitter based TRNG.

In contrast to TRNGs that sample the amplified noise, jitter based TRNGs perform better in terms of throughput, sensitivity to external interferers, and integration with

digital circuits. However, this approach requires a perfectly matched, 50% duty cycle low-jitter fast oscillator for generating equiprobable bits. A high-jitter slow oscillator that has jitter-to-period ratio in the excess of 10% is needed to generate high entropy bits [16]. In addition, large frequency separation is mandatory for reducing potential correlations. As the throughput requirements increase, duty cycle matching becomes a challenge, and a post processor is required to deal with statistical bias introduced by the duty cycle mismatch.

Recently introduced variants of this class of TRNGs utilize arrays of high power consuming ring oscillators (RO) whose outputs are combined with an XOR tree, and sampled by flip-flops to harvest randomness contributed by the jitter of each RO [17–19]. Although they offer high integration potential with digital integrated circuits, the injection locking problem [20–23], weak power supply rejection against interfering signals, layout placement constraints required for large number of ROs, and high power consumption make this type inappropriate for many applications.

1.1.3. Chaotic Signal Sampling Based TRNGs

Chaos can be defined as an irregular, and complex behavior exhibited by dynamic systems for certain values of system parameters. Exponentially divergent, and aperiodic nature of chaotic dynamics is driven, and characterized by the underlying positive Lyapunov exponent(s), making the dynamic system extremely sensitive to variations in the initial conditions. The term butterfly effect is popularly used to describe this behavior. Any small variation in the initial conditions is transformed into large deviations throughout the spatio-temporal evolution of chaotic orbits.

Chaos based TRNGs use the underlying chaotic dynamics as the entropy source since a dynamic system operating in the chaotic regime can act as an information source according to the ergodic theory [24]. Although the non-linear dynamics of chaotic systems is theoretically defined in deterministic terms, their high sensitivity to small perturbations in the initial conditions render them practically unpredictable. The unpredictability of the TRNG is established with the help of electrical noise readily

available in circuit nodes. When continuous wandering of the initial conditions, and state variables are combined with the divergent behavior of the chaotic system, it becomes impossible to determine the initial conditions exactly as a result of limited measurement precision in practice.

A typical chaos based TRNG uses a non-linear dynamical system operating in chaotic regime as the entropy source at the front-end. An appropriate sampler is used to harvest random bits, and a post processor is utilized at the back-end to cope with the potential statistical imperfections in the signal processing chain as presented in Figure 1.5.

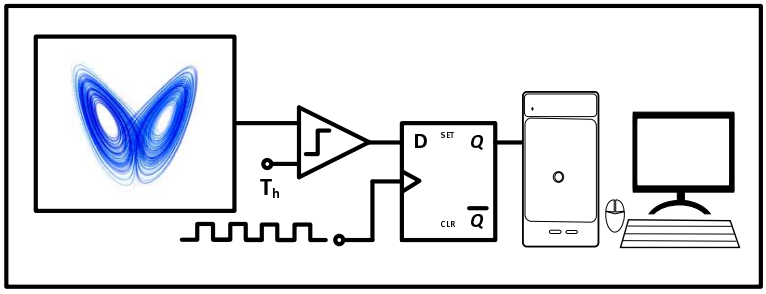


Figure 1.5. Chaos based TRNG.

Chaos based TRNGs can be classified as continuous time (CT), and discrete time (DT) depending on underlying dynamics. In CT chaotic systems, differential equations define the future state in terms of the rate of change associated with the current state variables [25], whereas in DT chaotic systems, the future state defined by the difference equations depends only on the value of current state [26]. Usually CT chaotic TRNG implementations require large area, and high power consuming analog circuit blocks, such as OPAMPs, OTAs, or oscillators [27, 28], as compared to their DT counterparts which can be built with lower number of components, [29, 30]. DT chaos based TRNGs require an external clock signal to drive the chaotic dynamics. Therefore, the generation, and evolution speed of the time-series forming the chaotic trajectories depends on the clock frequency. It is possible to adjust the clock frequency dynamically at runtime within available limits to enable low power, or high throughput operation without requiring any topological modifications. Moreover, unlike their CT counterparts, DT chaos based TRNGs do not need components that occupy large area such

as inductors used in cross coupled chaotic oscillators [28]. This makes DT chaos based TRNGs more compatible with the standard digital CMOS processes in which cryptographic circuits are fabricated [31]. Their simple implementation with small silicon footprint makes them convenient candidates for applications that demand lightweight, and hardware efficient TRNGs such as RFID systems, smartcards, smartphones, etc. DT chaos based TRNGs can be implemented both by using switched current (SI), or switched capacitor (SC) design methods.

1.2. Randomness Performance Evaluation of TRNGs

Suitable metrics are required to investigate the degree of randomness for bitstreams produced by RNGs [32]. However, few standards exist in the field such as NIST800.22, FIPS140-2, and AIS31 that address practical statistical analysis techniques in evaluation of RNGs [33–35].

In practice, statistical testing is used to make sure that a random number generator produces numbers that appear to be random. National Institute of Standards and Technology (NIST) has developed various metrics that may be employed to investigate the randomness of cryptographic grade random number generators [33]. NIST800.22 statistical test suite is used in this thesis for validating TRNG model, and circuit generated bitstreams, since it is widely accepted, and popular in testing of random number generators.

Traditionally, the quality of the random numbers generated by TRNGs is evaluated using statistical tests. In hardware design, in order to harvest maximum entropy from a system, the boundaries of maximum entropy should be known. Unfortunately, conventional statistical tests only provide a pass-fail type output, and the test results cannot be used in circuit design. In performance evaluation of the entropy generated by DT chaotic systems of interest, a practical information measure is required to extract the necessary information about the critical parameters affecting the randomness performance, and their maximum allowable variation boundaries for proper circuit design.

In this thesis, we focus on cryptographic grade random number generation using discrete time chaos as the entropy source. Design aspects of DT chaos based TRNGs are explored starting from map equations to in depth analysis of the entropy generated by the maps. A new design approach, and a new TRNG architecture are introduced. We introduce architectural novelties, and show their superiority over conventional designs through a comparative study of generated entropy, and its parameter sensitivity. Mathematical models of TRNGs are constituted, and used to predict the randomness performance using information theoretic measures to explore the maximum allowable parameter variation boundaries. The outcome of the theoretical work is used to design DT chaotic circuits which are implemented in 180nm CMOS technology provided by UMC. Additionally, a dual entropy core TRNG circuit is implemented on a reconfigurable analog platform as a proof of the concept.

1.3. Contributions

In this dissertation, we study the design aspects of DT chaos based true random number generators, and make the following contributions:

- *Modeling, and characterization of DT chaos based TRNGs:* We develop custom mathematical models for DT chaos based TRNGs. We obtained the dynamical, statistical, and spectral characteristics of the chaotic maps using numerical simulations of the developed models. Entropy capacities of the maps are studied using a practical information metric, T-entropy. The randomness performance, and its dependence on parameter variations are revealed for the chaotic maps of interest. The maximum allowable parameter variation boundaries for generating high entropy random bits are calculated. Introduced characterization methods are universal in nature, and allow the prediction of the randomness performance, and its parameter sensitivity for any chaotic map, ahead of its physical implementation.

- *Introduction of a new dual entropy core DT chaos based TRNG architecture:* We introduce a new dual entropy core DT chaos based TRNG architecture that em-

ploys dual DT chaotic maps as the entropy sources. The entropy generated by the proposed architecture is shown to be higher than that of its single entropy counterpart. Moreover, T-entropy calculations revealed that proposed architecture achieves much better immunity to control parameter variations when compared to its conventional single entropy core predecessor.

- *A novel FPAA implementation of the dual entropy core DT chaos based TRNG architecture:* A novel FPAA implementation of dual Bernoulli map based TRNG is introduced to the literature. The operation of the circuit is confirmed with measurement, and statistical test results.

- *Introduction of a new design method for integrated dual entropy core DT chaos based TRNGs:* We propose A new matching driven design methodology for DT chaos based TRNGs using a practical information measure, T-entropy. The maximum allowable variation boundaries for critical system parameters which affect the randomness performance have been calculated using T-entropy of the bitstreams generated by numerical simulations of the custom DT chaos based TRNG mathematical models. Calculated maximum allowable parameter variations are mapped to circuit parameter tolerances in the proposed matching driven design methodology to create efficient cryptographic grade entropy sources for TRNG applications.

- *A novel ASIC implementation of the dual entropy core DT chaos based TRNG architecture:* Using the new matching driven design methodology, a novel integrated circuit implementation of the dual entropy core DT chaos based TRNG employing Bernoulli, and tent maps as the entropy sources is introduced. The operation of the circuit is validated with measurement, and statistical test results.

1.4. Publication List

The results of this thesis have been submitted, or published in the following:

- Cicek, I., A. Pusane, and G. Dundar, *"A Novel Dual Entropy Core True Random Number Generator,"* in Journal of Analog Integrated Circuits and Signal Processing (AICSP), http://dx.doi.org/10.1007/s10470-014-0324-y, Springer, 2014. *Invited paper, recently published online.*
- Cicek, I., A. 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.
- Cicek, I., A. Pusane, and G. Dundar, *"A Novel Dual Entropy Core True Random Number Generator,"* in 8th International Conference on Electrical and Electronics Engineering (ELECO), pp. 1-4, 2013.
- Cicek, I., and G. Dundar, *"A Chaos Based Integrated Jitter Booster for True Random Number Generators,"* in 21th IEEE European Conference on Circuit Theory and Design (ECCTD), pp. 1-4, 2013.
- Cicek, I., A. Pusane, and G. Dundar, *"Random Number Generation Using Field Programmable Analog Array Implementation of Logistic map,"* in 21st Signal Processing and Communications Applications Conference (SIU), pp. 1-4, 2013.
- Cicek, I., A. Pusane, and G. Dundar, *"Field Programmable Analog Array Implementation of Logistic Map,"* in 21st Signal Processing and Communications Applications Conference (SIU), pp. 1-4, 2013.
- Cicek, I., A. Pusane, and G. Dundar, *"A Feasibility Study of A 1D Chaotic Map for True Random Number Generation,"* in 20th Signal Processing and Communications Applications Conference (SIU), pp. 1-4, 2012.
- Cicek, I., and G. Dundar, *"A Hardware Efficient Chaotic Ring Oscillator Based True Random Number Generator,"* in 18th IEEE International Conference on Electronics, Circuits and Systems (ICECS), pp. 430-433, 2011.

1.5. Organization of the Thesis

The rest of this thesis is organized as follows:

In Chapter 2, dynamical, statistical, and spectral characterization of chaotic systems of interest that will be used as entropy sources is introduced. Although DT chaotic systems are studied, the methods can also be used to study the properties of continuous time chaotic systems with little modification.

In Chapter 3, theoretical aspects of true random number generation using single, and dual DT chaotic systems are explored through mathematical models. Theoretical conditions for generating equiprobable, and independent bits are discussed, and optimum parameter sets for generating high entropy random bits from DT chaotic systems of interest are calculated.

In Chapter 4, a practical information measure, T-entropy is introduced which will be used as a metric for randomness performance evaluation of the proposed models in Chapter 3. In depth analysis of entropy generation capability of chaotic systems of interest are investigated. 3D T-entropy plots constructed with the help of bitstreams generated by respective TRNG models of Chapter 3 are presented. Maximum allowable parameter variation boundaries for hardware design are calculated.

In Chapter 5, Field Programmable Analog Array (FPAA) implementation of single, and dual entropy core DT chaos based TRNG circuits is explained.

In Chapter 6, Application Specific Integrated Circuit Implementation of DT chaos based TRNG circuits are discussed.

In Chapter 7, Measurement setup for evaluation of the practical implementations are introduced. Measurement results, and statistical test results of the bitstream acquired from prototype TRNG circuits are presented.

And finally, Chapter 8 summarizes the main contributions of this thesis, and portrays the main conclusions along with potential research directions. As previously mentioned, the results of this thesis have been submitted, or published in [36–43].

2. CHARACTERIZATION OF CHAOTIC ENDOMORPHIC MAPS FOR TRNG APPLICATIONS

Intrinsic dynamical, spectral, and statistical properties of a chaotic system play a key role in determining performance of TRNG applications. In order to create an efficient TRNG design, it is important to study, and understand the characteristics, and the parameter dependencies ahead of hardware design. Chaotic systems exhibit extraordinary properties such as sensitive dependence on initial conditions, aperiodicity, and invariant measure. The divergent chaotic behavior of a non-linear dynamic system is driven by the underlying Lyapunov exponents. Lyapunov exponent characterize the rate of separation of infinitesimally close trajectories. In quantitative terms, two trajectories with an initial separation of δx_0 diverge at a rate given by Equation 2.1 in the phase space.

$$|\delta x(t)| \approx e^{\lambda t}|\delta x_0|. \tag{2.1}$$

Different orientations of the initial separation vector result in different rates of separation, consequently yielding a spectrum of Lyapunov exponents, equal in number to the dimension of the phase space. The largest one is called maximal Lyapunov exponent since it determines the notion of predictability of a dynamic system. A positive maximum Lyapunov exponent is widely accepted as an indicator of chaos. Infinitesimally small valued initial separation vector translates into huge variations at the state variables of the system with the help of positive Lyapunov exponents. The maximal Lyapunov exponent is defined as

$$\lambda = \lim_{t\to\infty} \lim_{\delta x_0 \to 0} \frac{1}{t} ln \frac{|\delta x(t)|}{|\delta x_0|} \tag{2.2}$$

which translates into

$$\lambda(x_0) = \lim_{N\to\infty} \frac{1}{N} \sum_{i=0}^{N-1} ln|f'(x_i)| \tag{2.3}$$

for a DT system with an initial value of x_0, and N iterations. Another efficient method for exploring the dynamic behavior of a chaotic system is the bifurcation diagram which shows the possible long term values of the state variable such as fixed points, or periodic orbits of a system as a function of a bifurcation parameter of the system. Usually, stable solutions are represented with a solid line, and unstable solutions are represented by a dotted line in a bifurcation diagram.

Frequency spectrum of chaotic signals has wideband characteristics as a result of aperiodic behavior. Fourier transform of the time series of a chaotic signal can be used to study the frequency domain characteristics since many physical processes are described as a sum of various individual frequency components. The Fourier transform of a signal generates a frequency spectrum that contains all of the information about the original signal, but in a different form, which means that the original signal can be completely reconstructed by an inverse Fourier transform. The Fourier transform of a noise like stochastic waveform like is also random.

Autocorrelation is defined as the cross correlation of a signal with itself. It is basically a measure of the similarity between observations as a function of the time lag between them. It is a useful time domain instrument for finding repeating patterns,

such as the presence of a periodic signal masked by noise. For a signal $f(t)$, the continuous autocorrelation $R_{ff}(\tau)$ is defined as

$$R_{ff}(\tau) = (f(t) * f^*(-t))(\tau) = \int_{-\infty}^{\infty} f(t+\tau) f^*(t) dx = \int_{-\infty}^{\infty} f(t) f^*(t-\tau) dx \quad (2.4)$$

where f^* represents the complex conjugate. Similarly the discrete autocorrelation R at lag k for a discrete signal x_n is expressed as

$$R_{xx}(k) = \sum_n x_n x_{n-k}^*. \quad (2.5)$$

Autocorrelation of a periodic function is also periodic with the same period. The autocorrelation of noise like signal will exhibit a strong peak at $\tau = 0$, and will be zero elsewhere. The power spectral density of a signal is a positive real function of a frequency variable associated with a stationary stochastic process, or a deterministic function of time. The power spectrum decomposes the content of a stochastic process into different frequencies present in that process, and helps to identify the periodic content within the signal.

While chaotic systems exhibit sensitive dependence on small variations in the initial conditions, their long term statistical characteristics do not depend on the initial conditions. Throughout the time evolution of a DT chaotic system, there exists a positive probability measure at step $n = N$ which is independent of the probability distribution at initial step $n = 0$. Also known as ergodicity property, meaning that there exists a measure of the phase space that is invariant by the dynamics [24]. In ergodic chaotic systems, trajectories have initial condition independent, and constant statistical characteristics also known as the invariant measure. With their well

defined statistical characteristics, and invariant measure, chaotic systems can act as information sources which promote their use as entropy sources in TRNG applications.

Although the dynamics of chaotic systems are defined in deterministic terms, their high sensitivity to the changes in initial conditions, when combined with the exponentially divergent aperiodic behavior driven by the underlying positive Lyapunov exponents, along with the invariant measure, make them efficient entropy sources for TRNG applications. Furthermore, continuous drift of the initial conditions due to electrical noise, and the lack of infinite measurement precision at circuit nodes make it impossible to determine the initial conditions exactly, hence providing the required security, and unpredictability for TRNG application [40].

An endomorphic map is a mathematical function whose range is equal to its domain such that it always maps subspaces to subspaces $f : S \to S$ with $S \subset \mathbf{R}$. In most cases the set S will be $S = [0, 1]$. The iteration values are always within the value set. In mathematical terms, they can also be expressed in the form of $x_{n+1} = f(x_n)$ where $n = 0, 1, 2, ...$, and $x_0 \in [0, 1]$.

In the following paragraphs, mathematical definitions of some circuit implementable DT chaotic maps will be introduced along with their dynamical, spectral, and statistical properties. Although we have chosen well-known, and well-studied DT chaotic systems, the characterization methods introduced in this section are universal in nature, and can be applied to explore the features of other chaotic systems.

2.1. Logistic Map

The logistic equation is used to model the growth dynamics of a biological population [44]. The chaotic behavior of the logistic map was first discovered by Robert May in 1976 [45]. Logistic map is mathematically expressed by Equation 2.6 with parameter R being the growth rate, and x_n being the population at year n.

$$x_{n+1} = Rx_n(1 - x_n). \tag{2.6}$$

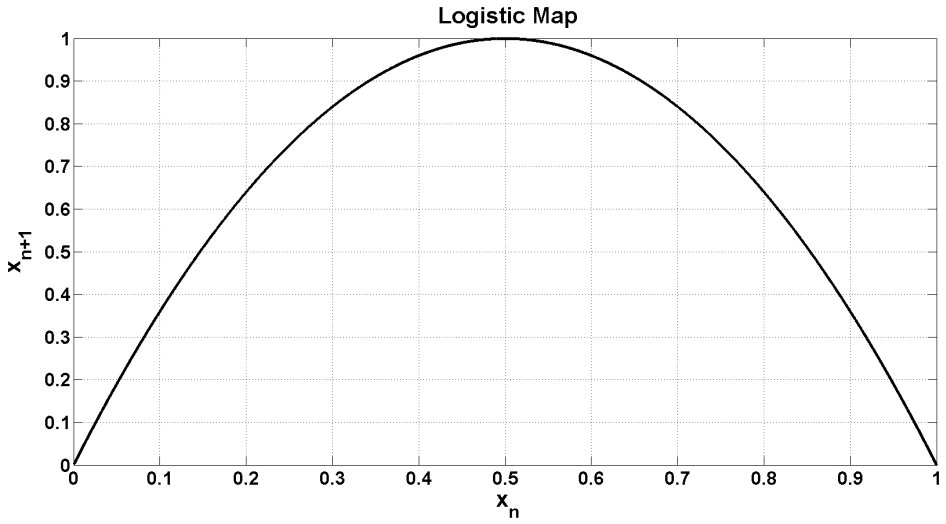


Figure 2.1. Logistic map function.

2.1.1. Dynamic Characteristics of the Logistic Map

The logistic map presented in Figure 2.1 exhibits interesting dynamic behavior with respect to the changes in parameter R as shown in Figure 2.2 which can be summarized as follows:

For $0 < R < 1$, the population will eventually die, independent of the initial population.

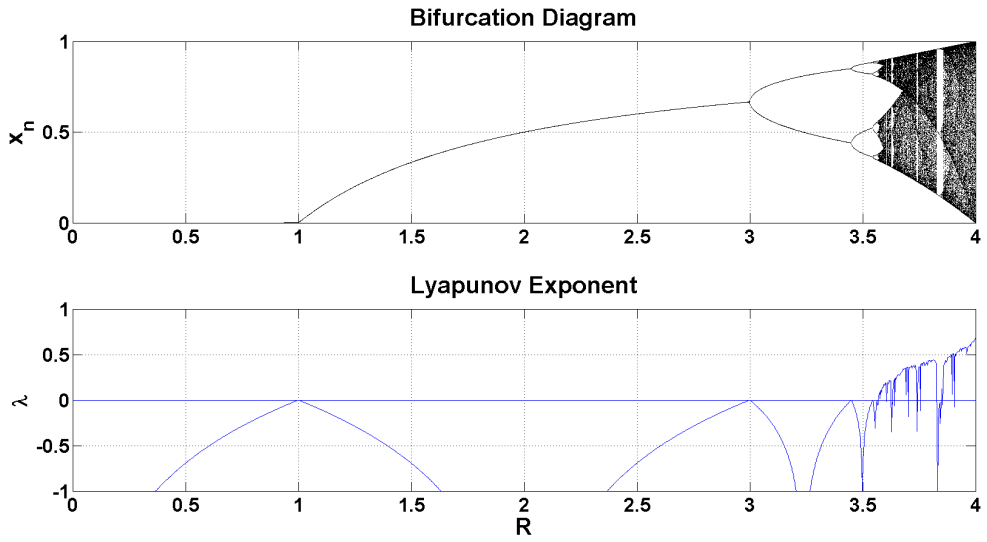


Figure 2.2. Bifurcation, and Lyapunov exponent diagram of the logistic map.

For $1 < R < 2$, the population will quickly reach the value $(R-1)/R$, independent of the initial population.

For $2 < R < 3$, the population will also eventually reach the same value $(R-1)/R$, but first will fluctuate around that value for some time. The rate of convergence is linear, except for $R = 3$, when it is dramatically slow, less than linear.

For $3 < R < 3.45$, from almost all initial conditions the population will approach permanent oscillations between two values. These two values are control parameter R dependent.

For $3.45 < R < 3.54$, from almost all initial conditions the population will approach permanent oscillations among four values.

For $R > 3.54$, from almost all initial conditions the population will approach oscillations between 8 values, then 16, 32, etc. The lengths of the parameter intervals which yield oscillations of a given length decrease rapidly; this behavior is an example of a period-doubling cascade.

The value $R = 3.57$, is the onset of chaos, at the end of the period-doubling cascade. From almost all initial conditions, any oscillations of finite period can no longer be observed. Small variations in the initial population yield drastically different results over time, meaning sensitive dependence on initial conditions, and aperiodicity which are two prime characteristics of chaos.

For $3.57 < R < 3.83$, the system exhibits chaotic behavior, but there are still certain isolated ranges of R that show non-chaotic behavior; these are sometimes called islands of stability. For instance, beginning at (approximately 3.83) there is a range of parameters, R, which exhibit oscillation within three values, and for slightly higher values of R, oscillation between 6 values, then 12 etc.

For $R > 4$, the values eventually leave the interval $[0, 1]$, and diverge for almost all initial values. All described behavior can be depicted with a bifurcation diagram along with the Lyapunov exponent graph generated by numerical simulations as shown in Figure 2.2. Maximal Lyapunov exponent of the logistic map is numerically calculated as $ln2 = 0.693$ when the control parameter R attains its maximum value at $R = 4$.

2.1.2. Spectral Characteristics of the Logistic Map

Spectral properties of the logistic map can be explored by using the time series forming the chaotic trajectory, and fast Fourier transform (FFT) to calculate the power spectral density of the map using the Welch's method [46] as presented in Figure 2.3. Starting from a random initial condition, the logistic map is iterated for 10^5 steps with parameter $R = 4$. the generated time series data is used to numerically calculate the Welch power spectral density estimate of the logistic map as shown in Figure 2.3.

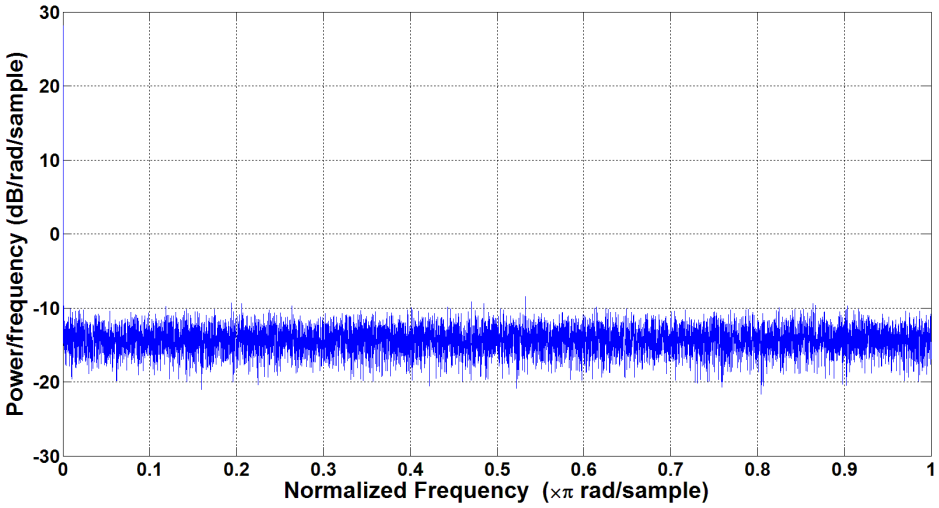


Figure 2.3. Power spectral density of the logistic map generated time series.

The power of the chaotic signal is distributed uniformly over the frequency band which is a desirable property for TRNG applications. The autocorrelation of the chaotic time series is also numerically calculated to confirm the noise like aperiodic behavior of the chaotic signal generated by the logistic map as presented in Figure 2.4.

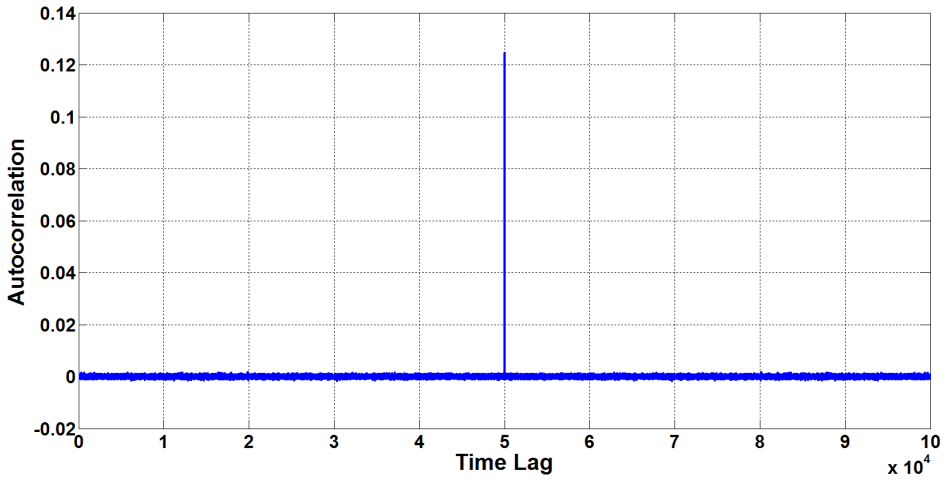


Figure 2.4. Autocorrelation of the logistic map generated time series.

Autocorrelation plot in Figure 2.4 shows that there is no linear correlation between the samples of the time series generated by logistic map. Consequently, both autocorrelation, and power spectral density of chaotic signals generated by the logistic map have noise like spectral properties.

2.1.3. Statistical Characteristics of the Logistic Map

Statistical properties of the logistic map can be studied using both theoretical, and numerical methods. In theoretical calculations Frobenius-Perron operator can be used to explore the time evolution of the underlying probability density function (PDF) of the logistic map, and in the limit case where PDF settles as a result of ergodicty, the analytical expression can be obtained [47]. The theoretical calculations of Ulam orand Von Neumann [48] yielded the PDF of the logistic map as

$$f_x(x) = \frac{1}{\pi\sqrt{x(1-x)}}. \tag{2.7}$$

Despite the highly sensitive, and divergent behavior, long term statistical characteristic of a chaotic system is usually constant, and independent of the initial conditions. It is not always possible to find an easy theoretical derivation of the underlying PDF. For such cases, a practical numerical approach is used to find a known PDF to match the underlying probability mass function. It requires the modeling, and simulation of the dynamic system of interest to generate time series data. PDF of the logistic map represented by Equation 2.7 can also be obtained by distribution fitting of empirically constructed probability mass function to a known PDF.

An empirical probability density function can be constructed using a histogram to display the frequency with which states along a trajectory fall into given bins composing the phase space. Imagine that the phase space $[0, 1]$ is composed of n discrete non-overlapping bins so that the jth bin is defined by

$$\beta_j = [\frac{j-1}{n}, \frac{j}{n}), \qquad j = 1, 2, ..n. \tag{2.8}$$

A trajectory of length N with $N >> n$ generated by a DT chaotic map $M(x)$, starting from the initial condition x_0 can be denoted as

$$T_N = \{x_0, M(x_0), M^2(x_0), ..., M^N(x_0)\}. \tag{2.9}$$

If we define the fraction f_e of the N states of the system that falls into the ith bin, we obtain

$$f_e = \frac{n}{N}\{\#M^i(X_0) \in \beta_j, j = 1, 2, .., N\}, \tag{2.10}$$

where # denotes the cardinality operator. It is possible to obtain an empirical PDF using large values of N in order to guarantee the settling of the invariant measure. For instance, using $n = 100$ bins, when the logistic map defined by Equation 2.6 with $R = 4$ is iterated for $N = 10^5$ time steps starting from a random initial condition x_0, Equation 2.10 implements a histogram as plotted in Figure 2.5. Although the time series generated by the logistic map has sensitive dependence on the initial state x_0, statistical distribution of the states over the phase space is invariant, and independent of x_0 [24]. It can be observed that the empirically obtained PDF is symmetric around 0.5. A literature search on well known statistical distributions revealed that the shape of the empirical PDF resembles the Beta distribution expressed by Equation 2.11.

$$f_x(x) = \frac{(x-p)^{\alpha-1}(q-x)^{\beta-1}}{(q-p)^{\alpha+\beta-1}B(\alpha,\beta)}, \qquad x \in [p,q], \quad \alpha,\beta > 0, \tag{2.11}$$

where $B(\alpha, \beta)$ is the standard Beta function, and it is calculated using the equation

$$B(\alpha, \beta) = \int_0^1 t^{\alpha-1}(1-t)^{\beta-1}dt. \tag{2.12}$$

Shape parameters (α, β) can be obtained by using maximum likelihood estimation method when the range of random variable $[p, q]$ is known. In the case of logistic map $[p, q]$ corresponds to $[0, 1]$ respectively.

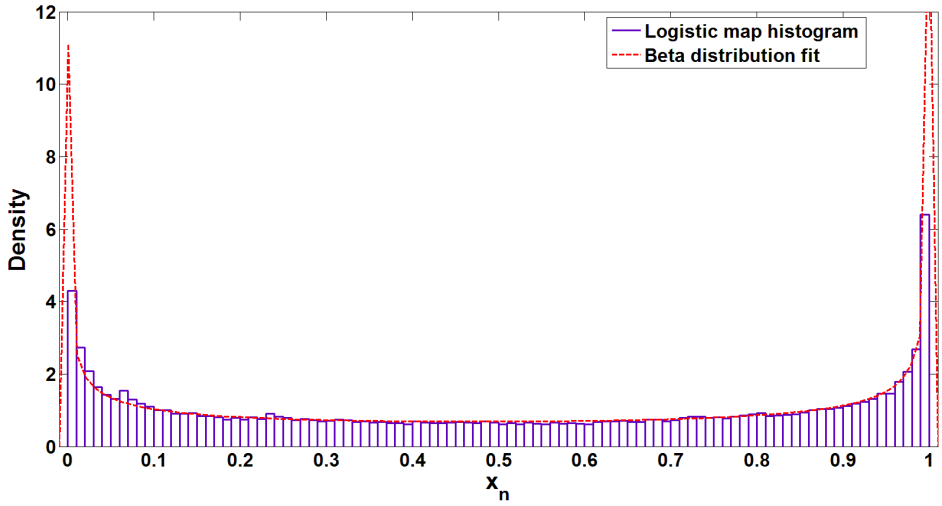


Figure 2.5. Histogram based probability mass function, and its distribution fit.

The following set of equations need be solved in order to obtain shape parameter estimators:

$$\begin{aligned} \psi(\hat{\alpha}) - \psi(\hat{\alpha}+\hat{\beta}) &= \frac{1}{N}\sum_{i=1}^{n} log(\frac{x_i - p}{q-p}), \\ \psi(\hat{\beta}) - \psi(\hat{\alpha}+\hat{\beta}) &= \frac{1}{N}\sum_{i=1}^{n} log(\frac{q - x_i}{q-p}), \end{aligned} \tag{2.13}$$

where $\{p = 0, q = 1\}$, and x_i is the logistic map time series data. Shape parameters are obtained as $\{\alpha = 0.5, \beta = 0.5\}$ using the maximum likelihood estimator equation

set. When the parameter values are substituted in Equation 2.11, and Equation 2.12 the probability density function for logistic map is obtained as

$$f_x(x) = \frac{1}{\pi\sqrt{x(1-x)}}, \tag{2.14}$$

which is equal to the theoretically calculated PDF Equation 2.7 [48].

2.2. Tent Map

Tent map is a piece-wise linear function that is composed of two linear sections that combine into a tent like structure (hence the name) as illustrated by Figure 2.6. It is one of the circuit implementable chaotic map functions easily implementable by circuits with a uniform invariant measure, and white noise like spectrum. In mathematical terms, it is expressed as

$$x_{n+1} = \begin{cases} \mu x_n, & 0 \leq x_n \leq 0.5 \\ \mu(1-x_n), & 0.5 < x_n \leq 1 \end{cases} \tag{2.15}$$

where μ is the control parameter. Skew tent map is a similar but more general form of the tent map which is mathematically defined as

$$x_{n+1} = \begin{cases} \mu x_n, & 0 \leq x_n \leq \frac{1}{\mu} \\ \frac{\mu}{\mu-1}(1-x_n), & \frac{1}{\mu} < x_n \leq 1 \end{cases} \tag{2.16}$$

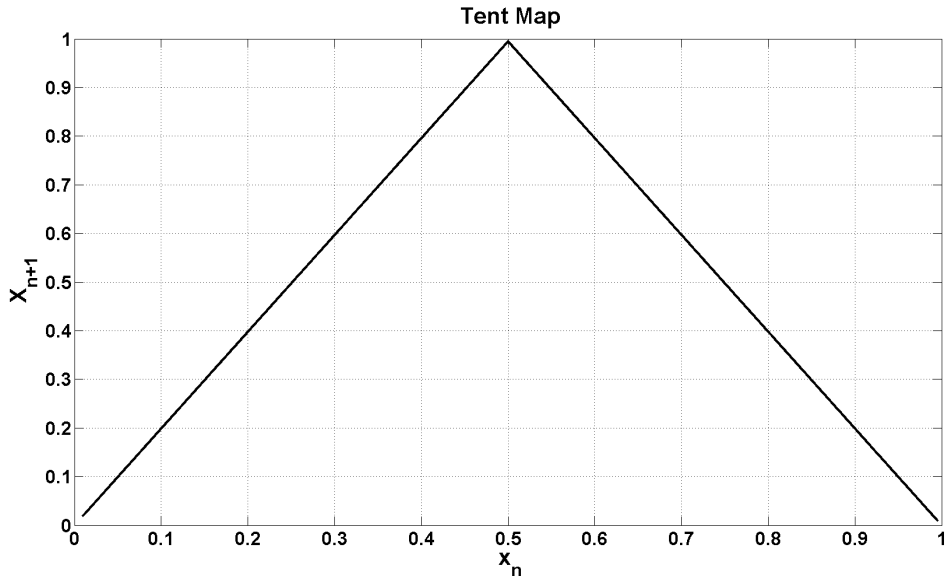


Figure 2.6. Tent map function.

where μ is the control parameter. The slope transition point of the skew tent map can move freely with respect to the control parameter. Thus, skew tent map is asymmetric for all $\mu \neq 2$, and it is symmetric, and equal to the tent map for $\mu = 2$.

2.2.1. Dynamic Characteristics of the Tent Map

Tent map presented in Figure 2.6 shows a robust dynamic behavior with respect to the changes in control parameter μ as presented by the bifurcation diagram in Figure 2.7. The dynamic characteristic of the tent map can be summarized as follows

For $0 < \mu < 1$, tent map dynamics do not exhibit chaotic behavior, and the dynamics are stable.

For $1 < \mu \leq 2$, the tent map enters the chaotic regime as a result of the positive Lyapunov exponent driven divergent behavior. It is important to note that, no stability regions exists as in the case of logistic map presented in Figure 2.2.

Maximal Lyapunov exponent of the tent map can be calculated analytically using the Equation 2.3, and taking the derivative of Equation 2.15 as follows,

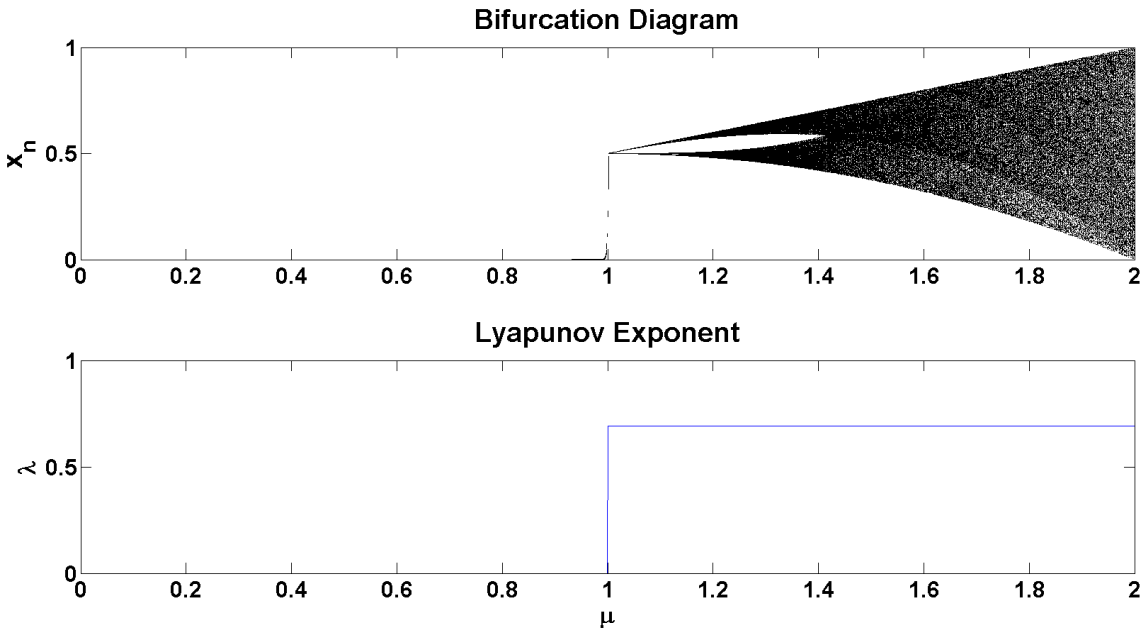


Figure 2.7. Bifurcation, and Lyapunov exponent diagram of the tent map.

$$f'(x_n) = \begin{cases} +\mu, & 0 \leq x_n \leq 0.5 \\ -\mu, & 0.5 < x_n \leq 1 \end{cases} \tag{2.17}$$

from which the Lyapunov exponent of the map can be calculated as

$$\lambda(x_0) = \lim_{n\to\infty} \frac{1}{n} \sum_{i=0}^{n-1} ln|f'(x_i)| = \lim_{n\to\infty} \frac{1}{n} \sum_{i=0}^{n-1} ln|\mu| = ln\mu. \tag{2.18}$$

Maximal Lyapunov exponent of the tent map is calculated as $ln2 = 0.693$ when control parameter achieves its maximum at $\mu = 2$. A similar result can be obtained for the case of skew tent map using the same approach yielding $ln2 = 0.693$ for $\mu = 2$.

2.2.2. Spectral Characteristics of the Tent Map

Spectral properties of the tent map are studied by using the time series forming the chaotic trajectory, and fast Fourier transform to calculate the power spectral density of the map using Welch's method [46] as presented in Figure 2.8. Tent map is iterated for 10^5 steps with parameter $\mu = 2$ starting from a random initial condition. Time series generated by the map is used to numerically calculate the Welch power spectral density estimate of the tent map as shown in Figure 2.8. Tent map has a flat power spectrum, the power of the chaotic signal is distributed uniformly over the frequency band which is a desirable property for TRNG applications.

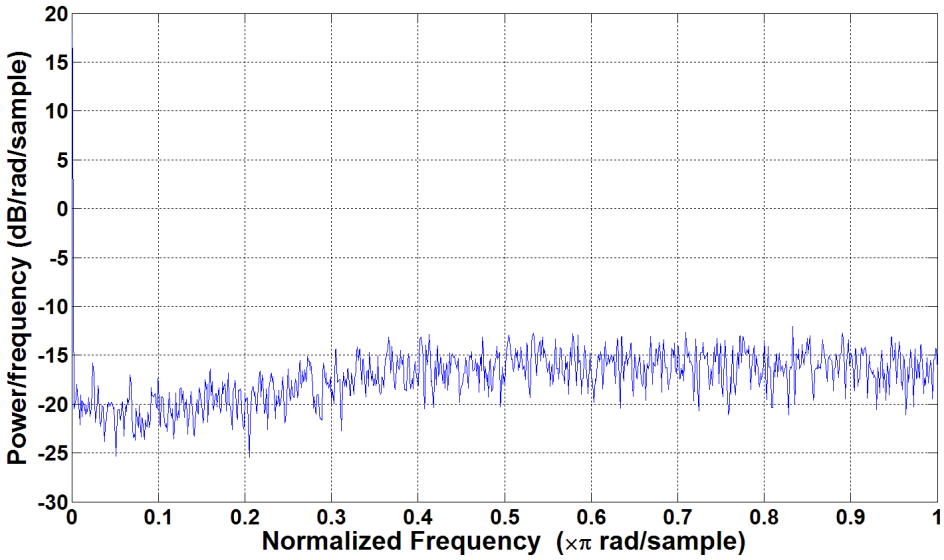


Figure 2.8. Power spectral density of the tent map generated time series.

In addition, the autocorrelation of the chaotic time series is numerically calculated for the tent map to confirm the noise like aperiodic behavior of the generated time series as presented in Figure 2.9. No linear correlation between the samples of the time series generated by tent map exists as suggested by the autocorrelation plot presented in Figure. 2.9. Chaotic signals generated by the tent map have noise like spectral properties both in terms of power spectral density, and autocorrelation.

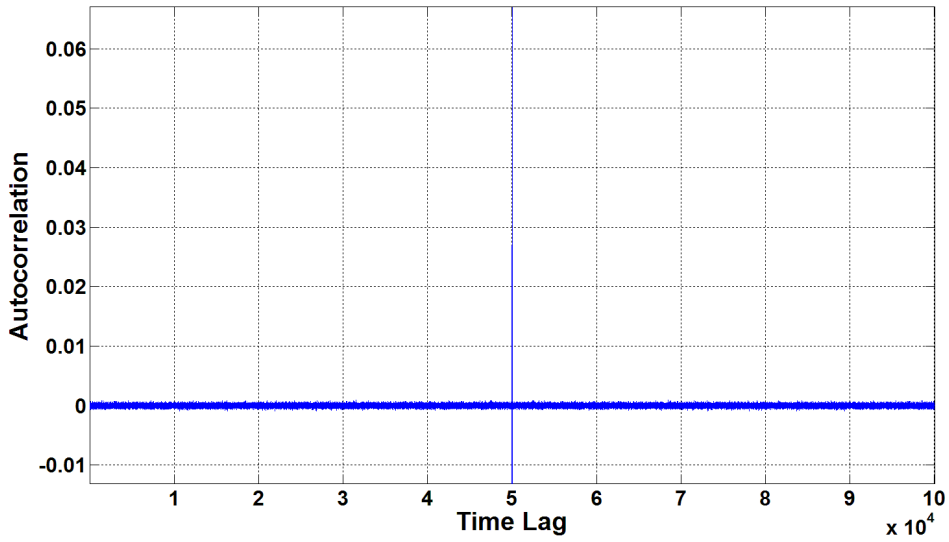


Figure 2.9. Autocorrelation of tent map generated time series.

2.2.3. Statistical Characteristics of the Tent Map

Statistical properties of the tent map can be studied using both theoretical, and numerical methods. The numerical approach used for calculation of the underlying PDF of the logistic map in the previous section can be used to reveal the underlying probability distribution of the tent map. An empirical probability density function can be constructed using a histogram to display the frequency with which states along a trajectory fall into given bins forming the phase space. Using a large number of iterations in numerical simulations helps to guarantee settling of the PDF. For instance, skew tent map, defined by Equation 2.15, with $\mu = 2$ is numerically simulated for 10^5 iterations starting from a random initial condition x_0. The empirical PDF is calculated using a histogram with 100 bins, and plotted as shown in Figure 2.10. Although the time series generated by the map is sensitive to the initial state x_0, the statistical distribution of the states evolving over the phase space is invariant, and independent of x_0. At first glance, the statistical characteristic defined by the empirical PDF shown in Figure 2.10 suggests a uniform distribution, convenient for TRNG applications. Numerically calculated empirical PDF, and the PDF obtained by statistical distribution fitting practically confirms that tent map has uniform invariant measure.

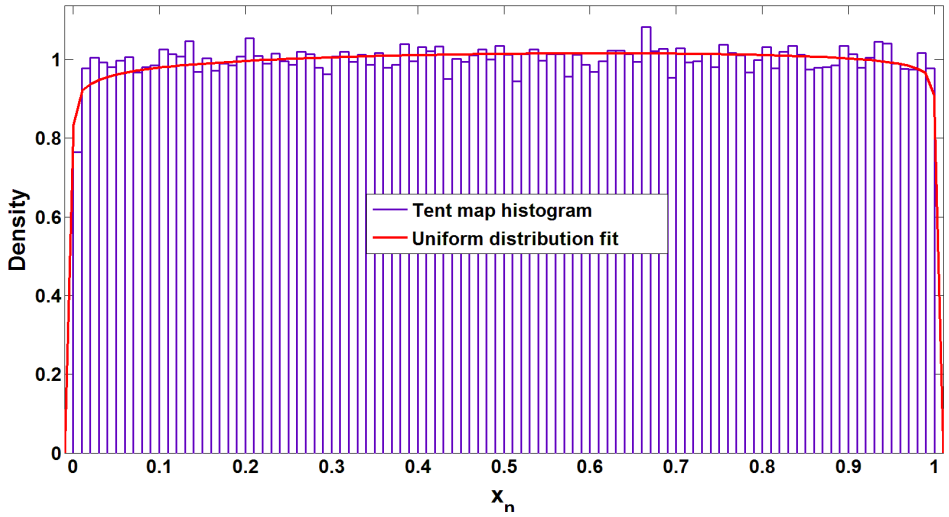


Figure 2.10. Histogram based probability mass function, and its distribution fit.

Underlying PDF can also be derived analytically by calculating the Frobenius-Perron (FP) operator for the skew tent map in the asymptotical case as the DT chaotic system iterates to infinity [47]. The general expression of FP operator for a map function $M(x):\ [0,1] \to [0,1]$ is expressed as

$$Pf(x) = \frac{d}{dx} \int_{M^{-1}(\Phi)} f(u)du, \tag{2.19}$$

where $M^{-1}(\Phi)$ is the counter image of the interval $\Phi = [0, x]$ on the x_{n+1} axis, which corresponds to x_n after one iteration in the phase space. In the case of the skew tent map, it is easy to show that $M^{-1}([0,x]) = [0, \frac{x}{\mu}] \cup [1-(1-\frac{1}{\mu})x, 1]$. Using Equation 2.19 we calculated the Frobenius-Perron operator of the skew tent map as

$$Pf(x) = \frac{d}{dx}\{\int_{0}^{\frac{x}{\mu}} f(u)du \quad + \int_{1-(1-\frac{1}{\mu})x}^{1} f(u)du\} = 1. \tag{2.20}$$

It is convenient to choose an initial density of $f(x) \equiv 1$ since initial conditions for the DT chaotic system are assumed to be determined by the uniformly distributed thermal noise. If we substitute the expression of $Pf(x)$ for $f(x)$ in Equation 2.20 to obtain the evolution of densities in parallel with the time iterations, asymptotically we obtain

$$f_\infty(x) = \lim_{n\to\infty} P^n f(x) = 1. \tag{2.21}$$

It is obvious that PDF of the skew tent map corresponds to the uniform distribution as estimated by the numerical approach.

2.3. Bernoulli Map

Bernoulli map is a piece-wise linear chaotic map function easily implementable by circuits with a uniform invariant measure, and white noise like spectrum. In mathematical terms it is defined as

$$x_{n+1} = \begin{cases} \beta x_n, & 0 \le x_n \le 0.5 \\ \beta x_n - 1, & 0.5 < x_n \le 1. \end{cases} \tag{2.22}$$

It is composed of two lines with a discontinuity point between them as shown in Figure 2.11. Bernoulli map is also known as the dyadic, or bit shift map. When x_n is written in binary notation, x_{n+1} is calculated by shifting the binary point one bit to the right, and replacing the new binary point with a zero, if the bit to the left of it is one.

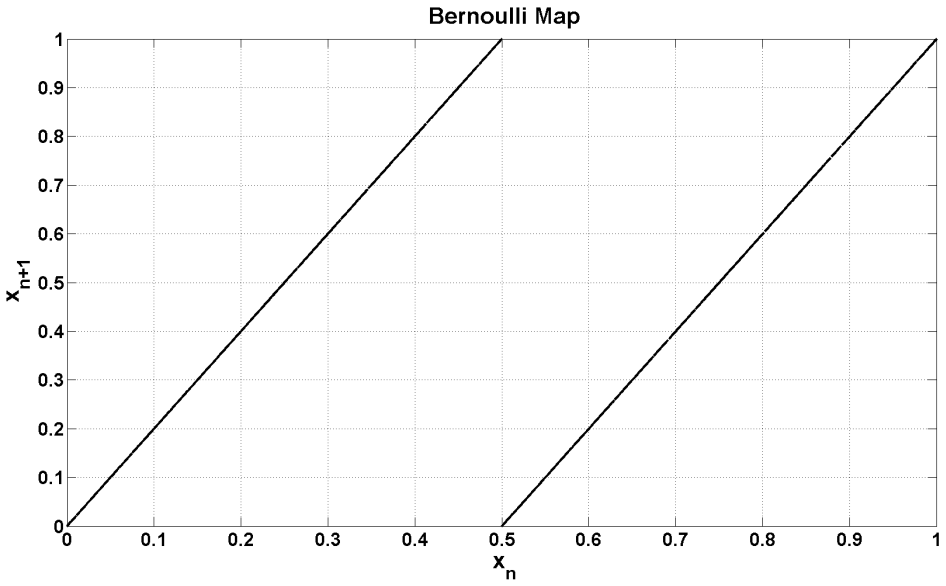


Figure 2.11. Bernoulli map function.

2.3.1. Dynamic Characteristics of the Bernoulli Map

The Bernoulli map presented in Figure 2.11 shows a robust dynamic behavior with respect to the changes in control parameter μ as presented by the bifurcation diagram in Figure 2.7. The dynamic characteristic of the Bernoulli map can be summarized as follows

For $0 < \beta < 1$, dynamics of the Bernoulli map do not exhibit chaotic behavior, and the map has stable behavior.

For $1 < \beta \leq 2$, Bernoulli map exhibits chaotic behavior as a result of the positive Lyapunov exponent driven divergent characteristics. It is important to note that, no stability regions exists as in the case of logistic map presented in Figure 2.2.

Maximal Lyapunov exponent of the Bernoulli map can be calculated analytically using the Equation 2.3, and taking the derivative of Equation 2.22 as follows,

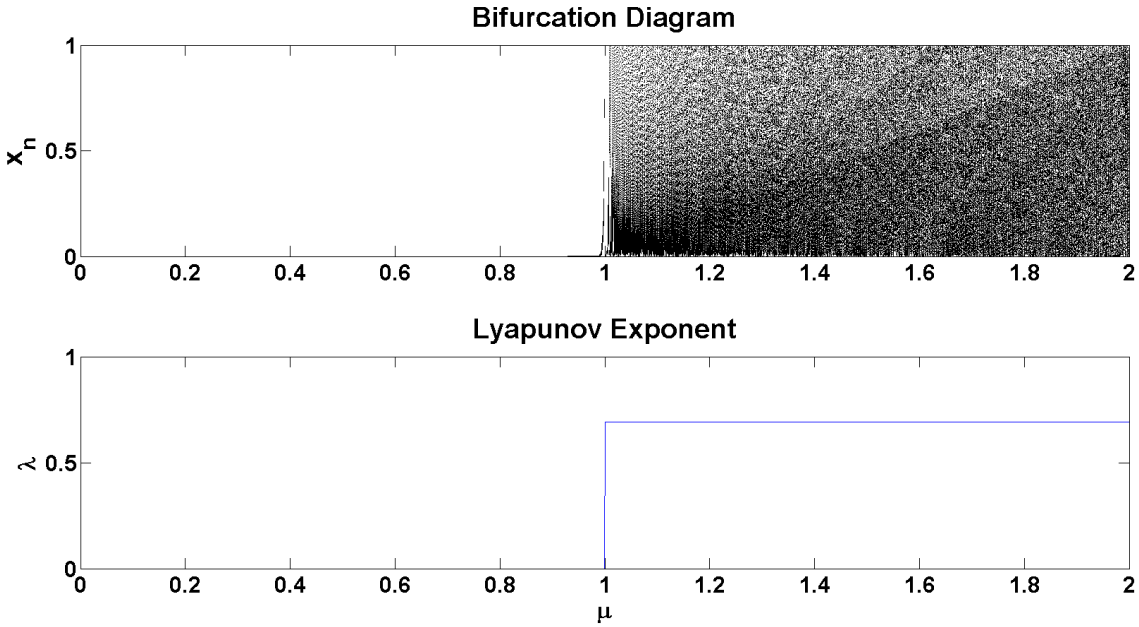


Figure 2.12. Bifurcation, and Lyapunov exponent diagram of the Bernoulli map.

$$f'(x_n) = \begin{cases} +\beta, & 0 \leq x_n \leq 0.5 \\ -\beta, & 0.5 < x_n \leq 1 \end{cases} \tag{2.23}$$

from which the Lyapunov exponent of the map can be calculated as

$$\lambda(x_0) = \lim_{n\to\infty} \frac{1}{n} \sum_{i=0}^{n-1} ln|f'(x_i)| = \lim_{n\to\infty} \frac{1}{n} \sum_{i=0}^{n-1} ln|\beta| = ln\beta. \tag{2.24}$$

Maximal Lyapunov exponent of the Bernoulli map is calculated as $ln2 = 0.693$ for $\beta = 2$.

2.3.2. Spectral Characteristics of the Bernoulli Map

Spectral features of the Bernoulli map are studied by using the chaotic trajectory formed by the time series generated by the map. The power spectral density of the Bernoulli map is calculated using fast Fourier transform, and Welch's method [46]. Starting from a random initial condition, Bernoulli map is iterated for 10^5 time steps with the control parameter set as $\mu = 2$. Time series generated by the map is used to numerically calculate the Welch power spectral density estimate of the Bernoulli map as shown in Figure 2.13.

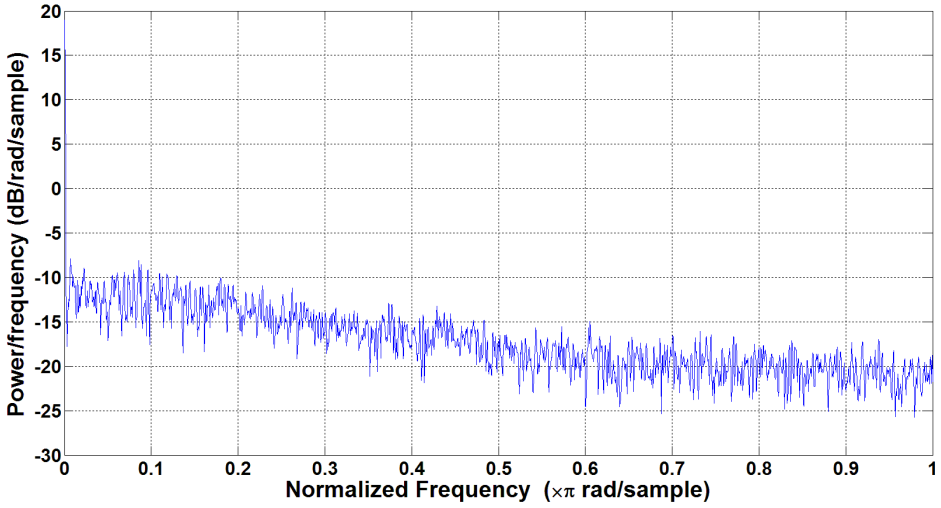


Figure 2.13. Power spectral density of the Bernoulli map generated time series.

Bernoulli map has a flat power spectrum, the power of the chaotic signal is distributed uniformly over the frequency band which is a desirable property for TRNG applications as in the case of tent map.

In addition, the autocorrelation of the chaotic time series is numerically calculated for the Bernoulli map to confirm the noise like aperiodic behavior of the generated time series as presented in Figure 2.14. Autocorrelation is the cross-correlation of a signal with itself. Namely, it is the similarity between observations as a function of the time delay between them. Autocorrelation is useful for finding repeating patterns, such as the presence of a periodic signal buried under noise. Since chaotic signals are aperiodic in nature, a noise like spectrum of autocorrelation is observed.

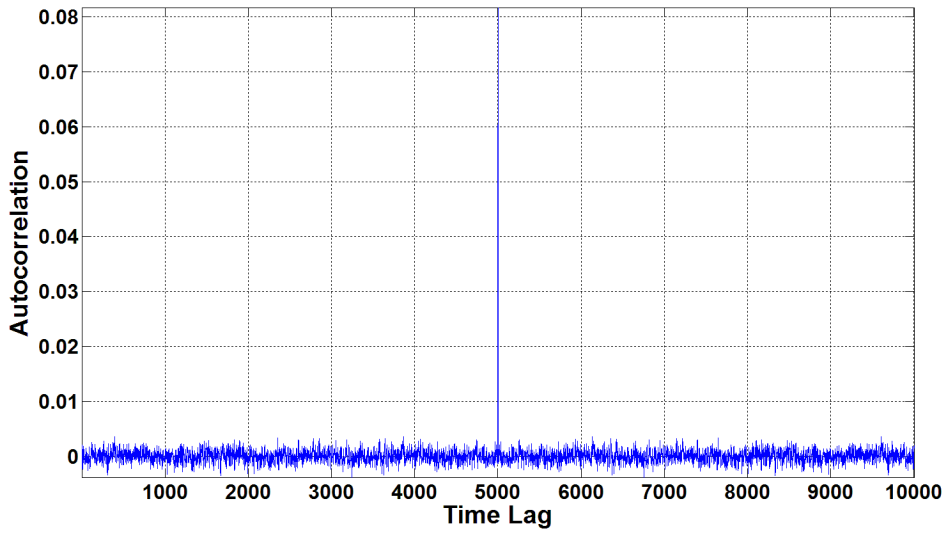


Figure 2.14. Autocorrelation of the Bernoulli map generated time series.

2.3.3. Statistical Characteristics of the Bernoulli Map

Both theoretical, and numerical methods can be used to study the statistical properties of the Bernoulli map. Numerical approach is used to obtain the empirical uniform probability distribution of the Bernoulli map as shown in Figure 2.15. The amplitude distribution of the Bernoulli map generated time series is plotted as a histogram to obtain the empirical probability mass function. Using an appropriate distribution fitting method, the underlying PDF of the Bernoulli map is calculated. Although the time series generated by the map is sensitive to the initial state x_0, the statistical distribution of the states evolving over the phase space is invariant, and independent of x_0. Statistical characteristics defined by the empirical PDF shown in Figure 2.15 suggest a uniform distribution, convenient for TRNG applications.

The underlying probability distribution of the Bernoulli map can also be calculated in theoretical terms using the previously presented Frobenius-Perron operator. It can be used to explore the time evolution of the underlying probability density function of the Bernoulli map, and in the limit case where PDF settles as a result of ergodicity, the analytical expression of the PDF can be obtained [47].

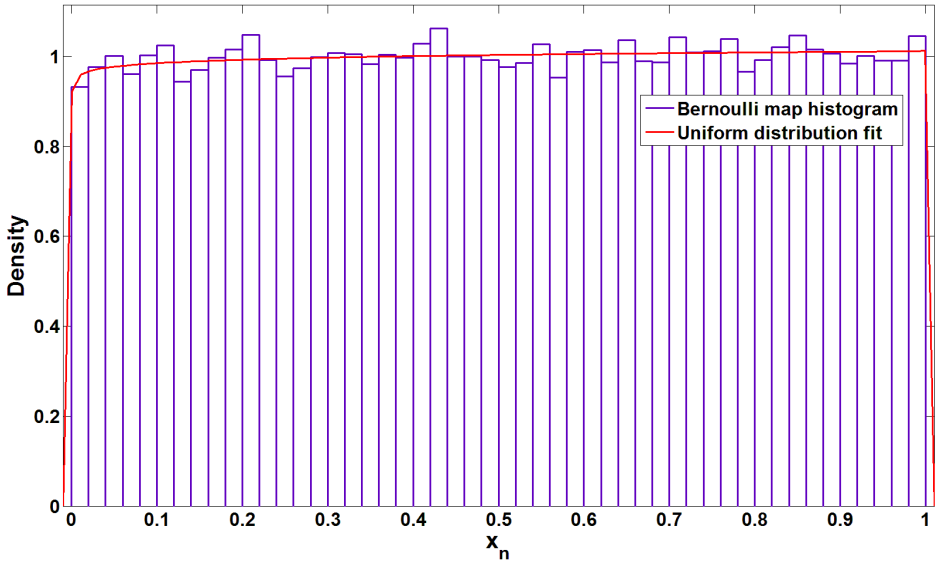


Figure 2.15. Histogram based probability mass function, and its distribution fit.

Since the general expression of FP operator defined in Equation 2.19 for an endomorphic map $M(x)$, we need to calculate the counter image $M^{-1}([0,x])$ of the interval $\phi = [0,x]$ on the x_{n+1} axis that corresponds to x_n. For the Bernoulli map, it is calculated to be $M^{-1}([0,x]) = [0, \frac{x}{\beta}] \cup [\frac{1}{\beta}, \frac{1+x}{\beta}]$. The FP operator for the Bernoulli map can be calculated as

$$Pf(x) = \frac{d}{dx}\{\int_{0}^{\frac{x}{\beta}} f(u)du \quad + \int_{\frac{1}{\beta}}^{1} f(u)du\} = \frac{2}{\beta}. \tag{2.25}$$

Under the assumption of uniform initial density $f(x) \equiv 1$ induced by the thermal noise, and for control parameter $\beta = 2$, the underlying PDF of the Bernoulli map is calculated to be the uniform distribution as in the case of tent map.

To sum up, while there exist a diverse spectrum of chaotic maps, those maps which are easily implementable by circuits are of interest in this work. Different aspects of such maps are explored using both analytical, and numerical methods. Properties

such as dynamic behavior, spectral, and statistical features important for TRNG applications are investigated. The logistic map exhibits interesting, and rich dynamical behavior for different values of the control parameter. There are stability islands in the bifurcation diagram shown in Figure 2.2 that have to be avoided due to periodic oscillations generated in those regions. Those stability regions are not encountered in the tent map, or the Bernoulli map. Both maps have continuous bifurcations, and generate robust chaotic signals for a wide range of control parameter values. Another advantage shared by the tent, and Bernoulli maps is their uniform distribution that allows generation of independent, and identically distributed random numbers. All three maps have noise like spectral characteristics as a result of aperiodic chaotic behavior. Chaotic signal power is spread all over the spectrum. No linear correlation can be observed in the generated time series clue to the divergent behavior driven by positive Lyapunov exponents. All maps of interest have the maximal Lyapunov exponent of $ln2 = 0.693$. A comparative table that incorporates different features of chaotic maps of interest is presented in Figure 2.16 to have a better understanding of the chaotic maps, and their properties.

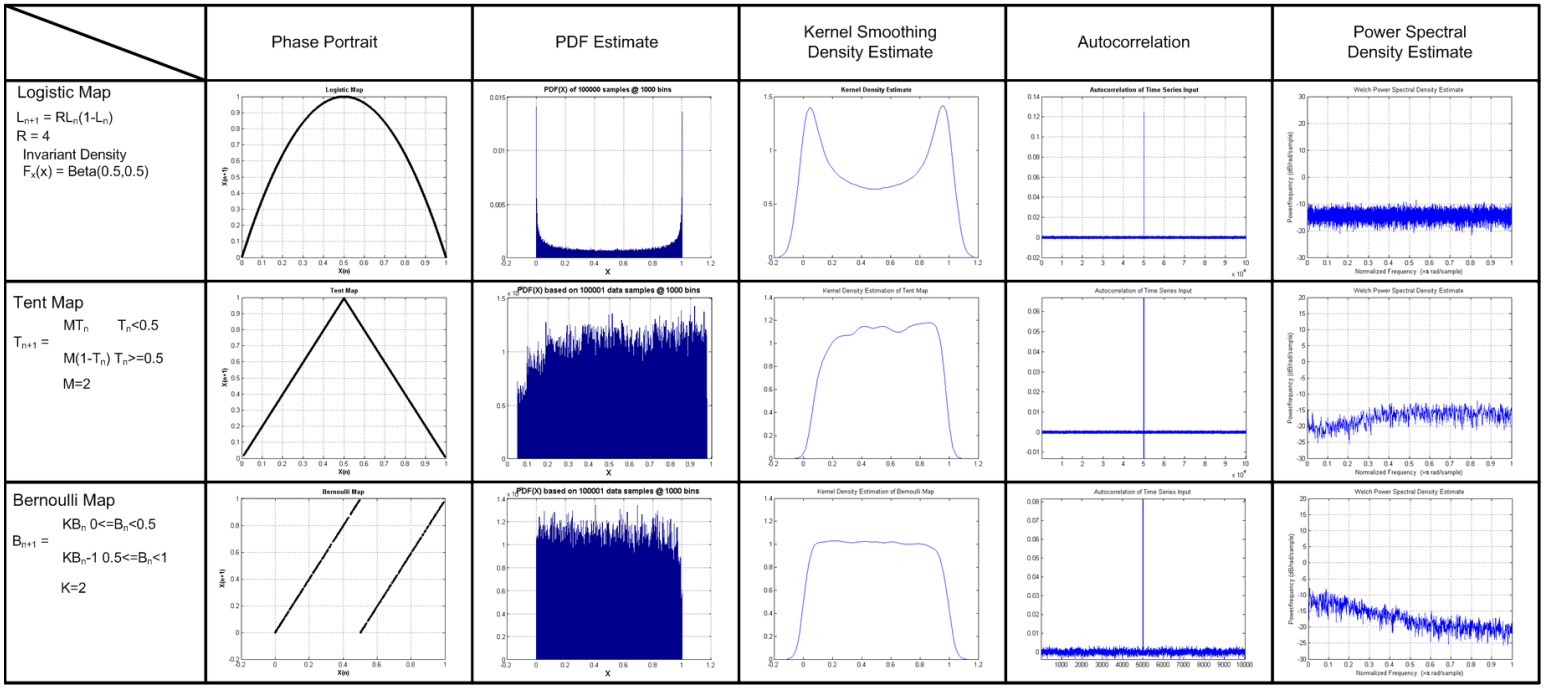


Figure 2.16. A comparative outlook of common chaotic maps, and their characteristics.

3. THEORETICAL ASPECTS AND MATHEMATICAL MODELING OF DT CHAOS BASED TRNGS

In this chapter, theoretical aspects of generating independent, and identically distributed random bits from a DT chaotic system will be studied. Mathematical models for numerical simulations of DT chaos based true random number generators will be constructed. Essential conditions for equiprobability, and statistical independence require special treatment, and will be addressed through theoretical calculations.

True random number generators are accepted to be the most crucial building block of any cryptographic system, since no other component is capable of generating more entropy at the output than what is available at the input [40]. A chaotic system can act as an information source with well defined statistical characteristics which are also known as the invariant measure. The information generation capacity of a chaotic system is, in principle, bounded by the sum of its Lyapunov exponents [49]. The nonlinear maps of interest have confined number of Lyapunov exponents which put a fundamental limit on the maximum entropy they can produce. So it is essential to develop an optimum harvesting method that allows the most efficient use of the limited randomness available from within the chaotic system. Chaotic systems are deterministic in nature by their theoretical definitions. In a practice, the unpredictability is constituted on the fact that thermally induced zero mean noise fluctuations in the initial conditions will be translated into huge variations at the state variables with the help of the divergent characteristic driven by the positive Lyapunov exponents. Because of the lack of infinite measurement precision, it is practically impossible to predict the past values of the time series, or initial conditions by observing the current values. Thus, the required unpredictability for the TRNG applications is established with the help of positive Lyapunov exponents, and sensitive dependence on the initial conditions.

3.1. Mathematical Modeling of Single Entropy Core DT Chaos Based TRNGs

Since Kolmogorov's original work [50], it has been well known that the phase space of a dynamic system can be partitioned for studying the evolution of dynamics in terms of a finite set of symbols. The trajectory generated throughout the spatio-temporal evolution of a chaotic system can be encoded using an alphabet composed of finite number of elements for creating a symbolic representation of the time evolution. For this translation, the required mapping between the real valued time series, and the discrete symbols can be established by partioning the phase space using single, or multiple threshold levels. When the phase space of a dynamic system is partitioned into non-overlapping symbol generating regions, the discrete encoding of a chaotic trajectory will be a stream of consecutive symbols with certain statistical characteristics inherited from the original chaotic time series. Statistical distribution of the bits is going to be a strong function of the threshold used to divide the phase space, and parameters controlling the dynamics. Partitioning can be practically implemented using a number of comparators with appropriate thresholds. If the number of elements in the alphabet is confined to two, the symbolic encoding of the time series will correspond to digital representation of the chaotic trajectory in terms of bits. A single threshold is required in this case, which divides the phase space of the chaotic system into two bit generating partitions. The single entropy core architecture basically encodes the chaotic trajectory using a binary alphabet whose elements are equally probable, provided the threshold is correctly calculated. In practice, the threshold generator must be capable of tracking the chaotic signal, and must be able to generate the optimum threshold value for yielding equiprobable random bits. From an implementation point of view, this requirement complicates the hardware design.

Although there is no restriction in the number of partitions, for the sake of simplicity, and reduced complexity, a binary partition (bipartition) formed by the use of a comparator, and a threshold generator is preferred for modeling. Any chaotic map, and its feasibility for TRNG application can be studied using the proposed model shown in Figure 3.1 which is composed of a non-linear function block implementing

the 1D chaotic map, a sample and hold block that drives the DT dynamics, and a threshold generator that divides the phase space into two partitions with the help of a comparator operating as a one bit analog to digital converter. Aforementioned circuit-friendly 1D chaotic maps such as the logistic map represented by Equation 2.6, skew tent map defined by Equation 2.15, or Bernoulli map expressed by Equation 2.22 can be used as the entropy source in the proposed TRNG model.

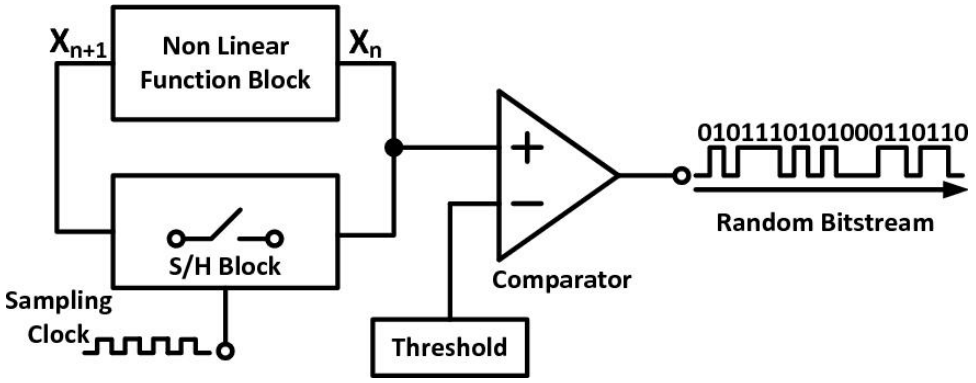


Figure 3.1. Single entropy core DT chaos based TRNG model.

Among the available candidates, skew tent map defined by Equation 2.15, is selected as the working example. The threshold comparator can be mathematically defined as

$$b_n = B(x_n) = \begin{cases} 0, & 0 \leq x_n \leq T_h \\ 1, & T_h < x_n \leq 1. \end{cases} \tag{3.1}$$

It is used to divide the phase space into two bit generating regions with respect to the threshold parameter T_h such that $T_h \leq \frac{1}{\mu}$. The DT chaos based TRNG model presented in Figure 3.1 generates random bits by comparing the spatio-temporal location of evolving chaotic trajectory in the partitioned phase space with respect to the threshold parameter as shown in Figure 3.2. Starting from an arbitrarily chosen initial state which is assumed to be determined by the thermal noise at circuit nodes in practice, chaotic evolution of the system on each iteration step will generate a bit depending on the location of chaotic samples within the partitioned phase space.

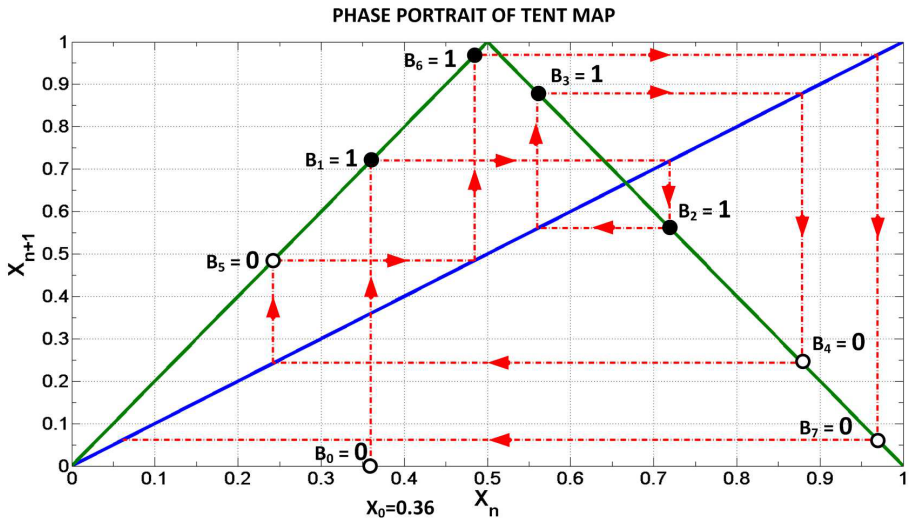


Figure 3.2. Random number generation using the phase portrait of skew tent map.

Equation 2.16, and Equation 3.1 form the single entropy core DT chaos based TRNG model. The model allows the evaluation of randomness performance of the chaotic map of interest ahead of physical implementation thus helps to save time, and resources. The proposed model generates random bits at each iteration of the map. For instance, starting from an arbitrarily chosen initial condition of $x_0 = 0.36$ the chaotic trajectory for eight time iterations of the map, is $x_n = \{0.36, 0.7196, 0.5604, 0.8787, 0.2425, 0.4848, 0.9691, 0.0618\}$. This trajectory is then encoded as a bitstream $B(x_n) = \{0, 1, 1, 1, 0, 1, 1, 0\}$ using Equation 3.1, and an arbitrarily chosen threshold value of $T_h = 0.4$ as presented in Figure 3.2. It is easy to see that T_h has direct impact on the statistics of the bitstream. The optimum value of the T_h has to be calculated accurately for generating high entropy bits with guaranteed identical distribution, and statistical independence.

3.1.1. Calculation of the Optimum Threshold for Identically Distributed Bit Generation

In order to choose a threshold for generating identically distributed bits from the entropy source, the underlying statistical distribution must be explored. This can be achieved using the time series generated by the TRNG model. Ergodic nature of chaotic systems imposes that long term statistical characteristics are independent of

the initial conditions [24]. Birkhoff's ergodic theorem states that finding the invariant measure of the chaotic map is equivalent to studying the evolution of the initial statistical distribution, such that the chaotic map of interest should be applied to each point in the initial distribution.When the invariant distribution is obtained, this corresponds to the invariant measure of the infinite aperiodic orbit [51]. As it was shown previously in Chapter 2, the skew tent map has a uniform invariant measure. Readily, the cumulative distribution function (CDF) for the skew tent map can be calculated using Equation 2.21 as the following

$$F(x) = \int_{-\infty}^{T_h} f_\infty(x)dx = T_h. \tag{3.2}$$

In physical implementations, circuit parameters corresponding to the dynamic system equation parameters are subject to variations as a result of fabrication imperfections, and matching issues [52–55]. Thus, it is a good practice to inspect the effect of control parameter variation over the invariant measure. For this purpose the proposed model presented in Figure 3.1 is used to generate random bits for all possible values of the control parameter of the map. Then a PDF slice for each step of control parameter is calculated, and combined in a 3D plot as shown in Figure 3.3. Bifurcation diagram is also embedded into the 3D projection of the PDF for presenting the unified chaotic, and statistical behavior of the dynamic system with respect to control parameter μ.

In the case of $\mu < 1$, for any initial value, the system is attracted towards the fixed point at $x = 0$. As μ approaches closer to two, underlying PDF becomes more uniform as demonstrated in Figure 3.3. According to the bifurcation diagram shown in Figure 3.3, it is possible to conclude that PDF of the skew tent map is insensitive to the variations in the chaos control parameter μ, thus yielding a robust source of chaos.

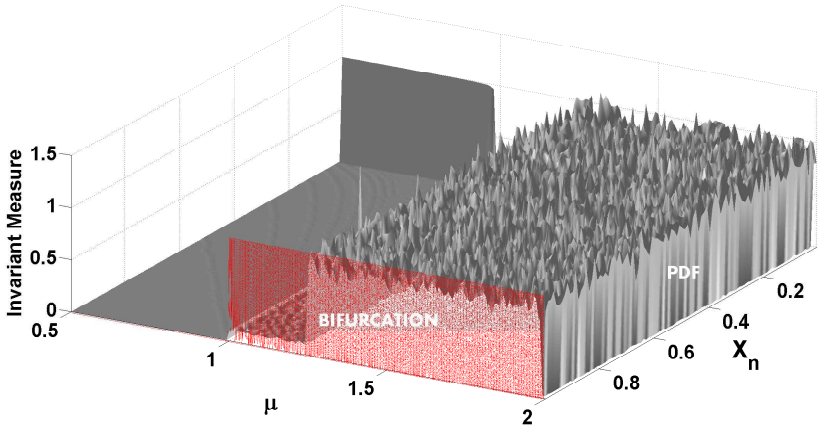


Figure 3.3. Unified 3D projection of bifurcation, and statistical characteristics of the trajectories generated by skew tent map.

In order to generate numbers from the source with high level of randomness, entropy must be harvested where it is maximum. In the context of the proposed TRNG model, the problem can be simplified to the determination of optimum threshold T_h that enables the generation of high entropy bits from an information source with a known PDF. The threshold parameter is used to define the boundary of non-overlapping bit generating partitions as shown in Figure 3.4.

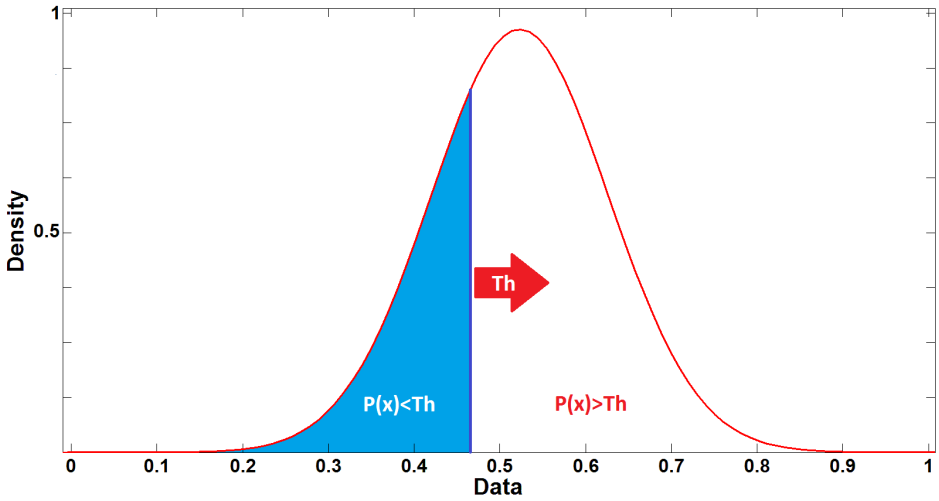


Figure 3.4. Partitioning of the PDF with respect to a threshold parameter.

Let us assume that bit zero is generated for $x \in [0, T_h]$ with probability P_0, and bit one is generated for $x \in [T_h, 1]$ with probability P_1 such that $P_0 + P_1 = 1$. Since the

area under the PDF curve determines the probability of a generated bit, it is possible to define the bit zero, and one generation probabilities in terms of the PDF $f_x(x)$ of the entropy source. Bit zero generation probability can be defined as

$$P_0 \;=\; P(x \leq T_h) = \int_{-\infty}^{T_h} f_x(x)dx = T_h. \tag{3.3}$$

Similarly, bit one generation probability can be defined as

$$P_1 \;=\; P(x > T_h) = \int_{T_h}^{\infty} f_x(x)dx = 1 - P_0 = 1 - T_h. \tag{3.4}$$

When the generated bits are considered independent, the entropy definition of Shannon [56, 57]

$$H = -\sum_{n=0}^{1} P_n log_2 P_n = -(P_0 log_2 P_0 + P_1 log_2 P_1) \tag{3.5}$$

can be used to calculate the optimum threshold that yields maximum entropy bits. If the definitions of P_0, and P_1 in Equation 3.3, and Equation 3.4 are substituted in Equation 3.5, the following expression for the entropy of a randomly generated bit is obtained.

$$H = -\left\{ (\int_{-\infty}^{T_h} f_x(x)dx) log_2 (\int_{-\infty}^{T_h} f_x(x)dx) + (\int_{T_h}^{\infty} f_x(x)dx) log_2 (\int_{T_h}^{\infty} f_x(x)dx) \right\} \tag{3.6}$$

$$= -\{T_h log_2 T_h + (1 - T_h) log_2 (1 - T_h)\}.$$

Maximum entropy yielding threshold can be calculated by solving the equation

$$\frac{dH(T_h)}{dT_h} = 0 \tag{3.7}$$

which attains its maximum at $T_h = 0.5$ as shown in Figure 3.5. This result points out that in order to obtain maximum entropy bits, zero, and one generation probabilities must be equal which can be achieved by finding the T_h that divides the area under the PDF curve into two equal partitions. It is possible to show that $T_h = 0.5$ for logistic, and Bernoulli maps using the same approach.

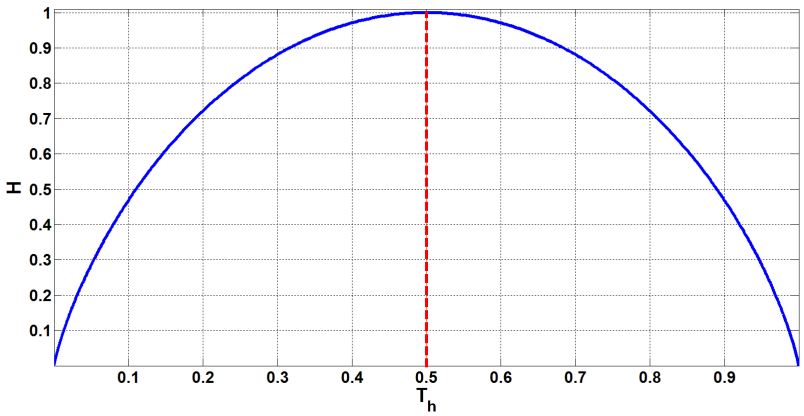


Figure 3.5. Entropy as a function of bit generation threshold, T_h.

According to Figure 3.5, Shannon entropy is maximized for randomly generated bits having equal probabilities, but in the context of TRNG application, the statistical dependence relation between consecutively generated bits still needs to be clarified.

3.1.2. Calculation of the Optimum Threshold for Independent Bit Generation

An ideal TRNG, by definition, is required to generate independent, and identically distributed numbers. Although samples from a chaotic system can never be regarded as truly independent due to the inherent deterministic relation defining the dynamics, it may be possible to establish statistical independence between consecutively generated bits using the binary quantized samples [58]. For an independent pair of bits $\{b_n, b_{n+1}\}$ generated from the skew tent map, it is possible to express their joint probability as the product of marginal probabilities having the form,

$$P_{ij} = P(b_n = i, b_{n+1} = j) \quad = P(b_n = i)P(b_{n+1} = j) \quad \text{for } i, j \in \{0, 1\} \tag{3.8}$$

where $P(b_n = \ i, b_{n+1} = j)$ represents the probability of concurrent conditions that $b_n = i$, and $b_{n+1} = j$ for $i, j \in \{0, 1\}$. In the proposed TRNG model, marginal bit generation probabilities are defined by Equation 3.3, and Equation 3.4 in terms of the threshold T_h. In order to calculate the joint probabilities, let us consider the case of $P_{00} = P(b_n = 0, b_{n+1} = 0)$. Keeping in mind that $T_h \leq \frac{1}{\mu}$, and taking into account the separation of cases at $X_n = \frac{1}{\mu}$, it is possible to derive the joint probability P_{00} as

$$P_{00} = \ \ P(b_n = 0, b_{n+1} = 0) = P(X_n \leq T_h, X_{n+1} \leq T_h). \tag{3.9}$$

By using additivitiy property,

$$P_{00} = \quad P(X_n \leq T_h, X_{n+1} \leq T_h, X_n \leq \frac{1}{\mu}) + P(X_n \leq T_h, X_{n+1} \leq T_h, X_n > \frac{1}{\mu}), \quad (3.10)$$

and by using the definition of X_{n+1} from Equation 2.16,

$$\begin{aligned} P_{00} = & P(X_n \leq T_h, \mu X_n \leq T_h, X_n \leq \frac{1}{\mu}) + \\ & P(X_n \leq T_h, \frac{\mu}{\mu - 1}(1 - X_n) < T_h, X_n > \frac{1}{\mu}) \end{aligned} \quad (3.11)$$

is obtained. It can equivalently be written as,

$$P_{00} = \quad P(X_n \leq min(T_h, \frac{T_h}{\mu}, \frac{1}{\mu})) + P(X_n \leq T_h, X_n \geq max(1 - \frac{T_h}{\mu}, \frac{1}{\mu})). \quad (3.12)$$

Consequently we obtain:

$$P_{00} = \quad \int_0^{\frac{T_h}{\mu}} f_\infty(u)du = \frac{T_h}{\mu}. \quad (3.13)$$

An independence metric expressed by Equation 3.14 can be defined by using the difference between the joint probability, and product of marginal probabilities for consecutively generated bits b_n, b_{n+1}. The independence metric will be zero if consecutively generated bits are independent.

$$\Psi_{ij} = P_{ij} - P_i P_j, \quad i, j \in \{0, 1\}. \quad (3.14)$$

In the case of $\{b_n = 0, b_{n+1} = 0\}$, the independence metric ψ_{00} is defined as the difference between Equation 3.13, and product of marginal probabilities P_0 defined by Equation 3.3. Generated random bits are considered to be independent when $\Psi_{00} = 0$.

If we substitute $\mu = 2$, and solve Equation 3.15, the optimum T_h value that guarantees statistical independence can be calculated as the following

$$\begin{aligned} \Psi_{00} &= P_{00} - P_0 P_0 = \frac{T_h}{\mu} - T_h^2|_{\mu=2} = 0 \\ &\Longrightarrow T_h = 0.5. \end{aligned} \tag{3.15}$$

The same approach can be used to calculate the joint probabilities for consecutively generated bits in the other three cases (P_{01}, P_{10}, P_{11}). The optimum threshold value for these three cases is calculated to be $T_h = 0.5$ for maximum entropy yielding control parameter value of $\mu = 2$. Table 3.1 lists the calculated joint, and marginal probabilities for the skew tent map. It is easy to see that the joint probabilities will be equal to 0.25, and marginal probabilities will be equal to 0.5 which makes $\Psi_{i,j} = 0, \quad i, j \in \{0, 1\}$, implying the existence of statistical independence between all consecutively generated bits for the optimum parameter set $\{\mu = 2, T_h = 0.5\}$.

Table 3.1. Joint and marginal bit generation probabilities for consecutively generated bits $\{b_n, b_{n+1}\}$ using skew tent map as the entropy source.

	$\mathbf{P(b_{n+1} = 0)}$	$\mathbf{P(b_{n+1} = 1)}$	$\mathbf{b_{n+1}}$
$\mathbf{P(b_n = 0)}$	$\frac{T_h}{\mu}$	$T_h(1 - \frac{1}{\mu})$	T_h
$\mathbf{P(b_n = 1)}$	$T_h(1 - \frac{1}{\mu})$	$1 - T_h(1 - \frac{1}{\mu}) - T_h$	$1 - T_h$
$\mathbf{b_n}$	T_h	$1 - T_h$	1

Similar results are obtained for other types of 1D chaotic maps using the same approach. We calculated the optimum set of parameters that enable the generation of independent, and identically distributed bits for the logistic map as $\{R = 4, T_h = 0.5\}$, and for the Bernoulli map as $\{\beta = 2, T_h = 0.5\}$. Theoretical calculations confirm

that skew tent map, defined by Equation 2.15, can be used as an entropy source for generating independent, and identically distributed bits.

3.2. Statistical Testing of Single Entropy Core DT Chaos Based TRNG

NIST800.22 statistical test suite v2.0 was used for testing the statistical performance of the single entropy core model. The chaotic maps of interest in Chapter 2 were used as entropy source, in the single entropy core TRNG model with respective optimum parameter set calculated above. Then, for each chaotic map, 400Mbits were generated using the model for statistical testing. In the NIST800.22 test results presented in Table 3.2, Table 3.3, and Table 3.4, each p-value corresponding to a particular test describes the probability of the bitstream that is generated by an ideal TRNG [33]. NIST800.22 statistical test suite divides the raw bitstream into 1Mbit blocks, and applies the suite of statistical tests. Proportion column in the Table 3.2, Table 3.3, and Table 3.4 shows the ratio of 1Mbit sequences passing the particular NIST800.22 test. According to the results in Table 3.2, Table 3.3, and Table 3.4, the generated bitstreams from each chaotic map successfully pass all NIST800.22 statistical tests.

Table 3.2. NIST800.22 statistical test results for the bitstream generated by logistic map based single entropy core TRNG model.

Test	**P-Value**	**Proportion**
Frequency	0.896380	0.9928
Block Frequency	0.808085	0.9952
Cumulative Sums	0.360448	0.9928
Runs	0.450891	0.9833
Longest-Run	0.310631	0.9809
Rank	0.916162	0.9833
FFT	0.637119	0.9928
Universal	0.546481	0.9905
Apen	0.657426	0.9833
Serial	0.821369	0.9857
Linear-Complexity	0.095035	0.9809

Table 3.3. NIST800.22 statistical test results for the bitstream generated by tent map based single entropy core TRNG model.

Test	**P-Value**	**Proportion**
Frequency	0.780592	0.9737
Block Frequency	0.536606	0.9857
Cumulative Sums	0.789883	0.9761
Runs	0.108321	0.9785
Longest-Run	0.348514	0.9952
Rank	0.986480	0.9881
FFT	0.556410	0.9857
Universal	0.859040	0.9857
Apen	0.314287	0.9809
Serial	0.842729	0.9881
Linear-Complexity	0.432784	0.9952

Table 3.4. NIST800.22 statistical test results for the bitstream generated by Bernoulli map based single entropy core TRNG model.

Test	**P-Value**	**Proportion**
Frequency	0.317972	0.9928
Block Frequency	0.167071	0.9952
Cumulative Sums	0.393516	0.9928
Runs	0.830035	0.9833
Longest-Run	0.727654	0.9881
Rank	0.130351	0.9833
FFT	0.850975	0.9857
Universal	0.512185	0.9857
Apen	0.884326	0.9881
Serial	0.928167	0.9928
Linear-Complexity	0.303409	0.9881

Although all model generated bitstreams pass the NIST tests, we still need an information metric capable of quantifying the entropy generated by the TRNG model in order to compare the randomness performances, and parameter variation tolerances of chaotic maps. For this reason, it is essential to explore the randomness of the

generated finite bitstream in depth using a practical information measure, as will be done in Chapter 4.

3.3. Mathematical Modeling of Dual Entropy Core DT Chaos Based TRNGs

The dual entropy core TRNG presented in Figure 3.6 generates random bits using the difference between two uniformly distributed, independent, and uncorrelated random variables. The proposed new architecture is based on symbolic dynamics, which translates the generated chaotic time series into symbolic binary strings of ones, and zeros by comparing chaotic samples from each entropy source [42]. Entropy core redundancy is used to increase the maximum achievable entropy, which is fundamentally limited by the Lyapunov exponent of the single entropy core architecture. While different endomorphic maps exhibiting chaotic behavior can be used as entropy sources, the Bernoulli map presented in Figure 2.11, and defined by Equation 2.22 is chosen for both entropy cores, since the map is capable of generating robust chaos. The similarities between the two entropy cores also reduce hardware complexity, and design overhead.

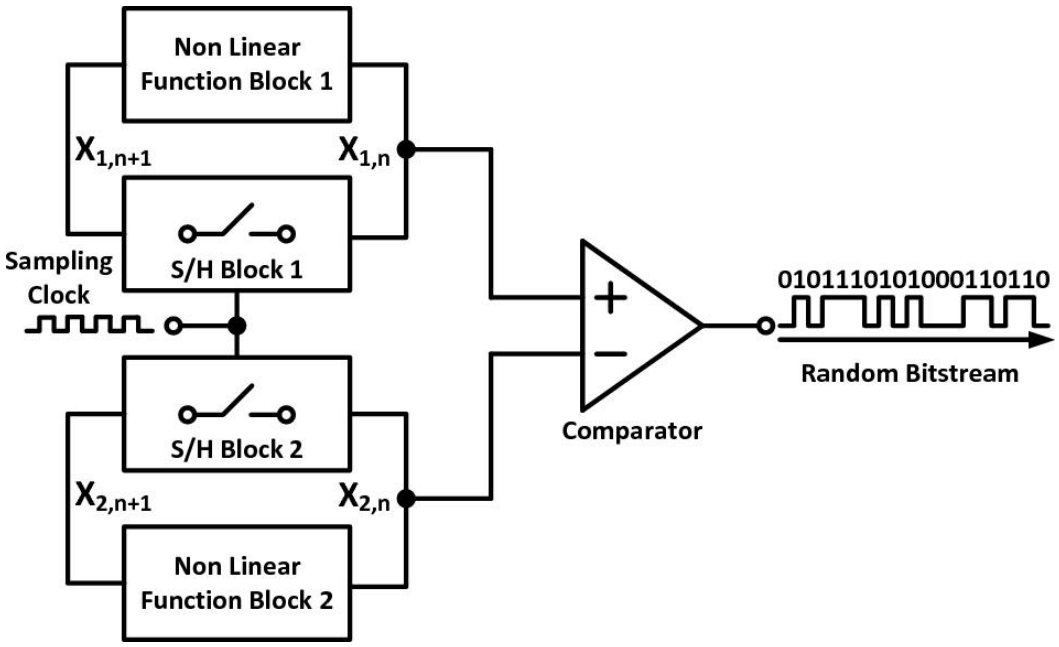


Figure 3.6. Dual entropy core DT chaos based TRNG Model.

A custom mathematical model of the dual entropy core TRNG architecture has been developed for studying the randomness performance using basic probability measures. Assume that we have two independent, uncorrelated, and uncoupled Bernoulli maps defined by Equation 2.22 that are guaranteed to start operating from different initial conditions with chaos controlling parameters $\{\mu_1, \mu_2\}$. Then, a bit extractor function can be defined such that,

$$b_n = B(x_{1,n}, x_{2,n}) = \begin{cases} 0, \ x_{1,n} \leq x_{2,n} \\ 1, \ x_{1,n} > x_{2,n} \end{cases} \tag{3.16}$$

where $x_{i,n}, \quad i = 1, 2, ...,$ corresponds to the chaotic time series generated by the ith entropy core. The mathematical model of the dual entropy core TRNG architecture is constructed using Equation 2.22, and Equation 3.16. As it can be inferred from Equation 3.16, random bits are generated by using the sign of the difference between the samples of two uniformly distributed random variables. The bit extractor function represented by Equation 3.16 can be implemented using a comparator in practice.

In order to have a better understanding of the statistical properties of the dual entropy core TRNG architecture, the joint PDF is calculated as follows: Let X_1, X_2 be independent, uncorrelated, and uniformly distributed random variables. A random variable Y composed of X_1, and X_2 can be defined as,

$$Y = X_1 - X_2. \tag{3.17}$$

The joint probability density function of X_1, X_2 is,

$$f_{X_1,X_2}(x_1,x_2) = 1, \qquad x_1, x_2 \in (0,1). \tag{3.18}$$

The cumulative distribution function of Y can be calculated using,

$$\begin{aligned} F_Y(y) &= P(Y \leq y) = P(X_1 - X_2 \leq y) \\ &= \begin{cases} \int_0^{1+y} \int_{x_1-y}^{1} 1 \, dx_2 dx_1, & -1 < y < 0 \\ 1 - \int_y^1 \int_0^{x_1-y} 1 \, dx_2 dx_1, & 0 \leq y < 1 \end{cases} \\ &= \begin{cases} \frac{1}{2}y^2 + y + \frac{1}{2}, & -1 < y < 0 \\ -\frac{1}{2}y^2 + y + \frac{1}{2}, & 0 \leq y < 1. \end{cases} \end{aligned} \tag{3.19}$$

The probability density function of Y is obtained by taking the derivative of cumulative distribution function as,

$$f_Y(y) = \frac{dF_Y(y)}{dy} = \begin{cases} 1 + y, -1 < y < 0 \\ 1 - y, & 0 \leq y < 1. \end{cases} \tag{3.20}$$

It is interesting to note that the probability density function of Y has zero mean, and symmetric distribution around zero, which allows generation of equiprobable bits. Using the proposed model, we numerically simulated the TRNG system using different random initial conditions for each entropy core, and constructed an empirical probability density function of composite random variable Y as presented in Figure 3.7 following a similar approach as explained in Chapter 2. The theoretically derived joint

probability function, and empirically constructed probability density functions are in good agreement as shown in Figure 3.7.

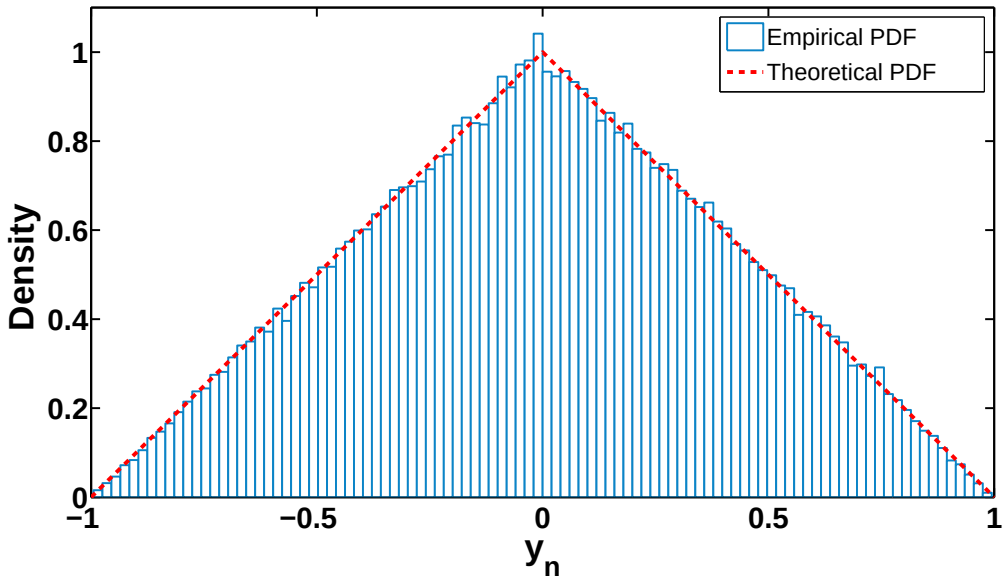


Figure 3.7. Probability density function of the composite random variable Y.

3.4. Statistical Testing of Dual Entropy Core DT Chaos Based TRNG

NIST statistical test suite v2.0 is used for evaluating the statistical performance of the model generated bitstream. 400Mbits of random data is recorded for testing purpose. In the NIST test results presented in Table 3.5, each p-value corresponding to a particular test describes the probability of the bitstream under test is generated by an ideal TRNG [33]. NIST statistical test suite divides the raw bitstream into 1Mbit blocks, and applies the tests. The proportion column in Table 3.5 shows the ratio of 1Mbit sequences passing the particular NIST test. According to the results in Table 3.5, the generated bitstream successfully passes all NIST statistical tests.

Table 3.5. NIST800.22 statistical test results for the bitstream generated by Bernoulli map based dual entropy core TRNG model.

Test	**P-Value**	**Proportion**
Frequency	0.966244	0.9900
Block Frequency	0.219006	0.9975
Cumulative Sums	0.838645	0.9875
Runs	0.304126	0.9875
Longest-Run	0.783973	0.9775
Rank	0.724817	0.9950
FFT	0.143279	0.9925
Universal	0.470189	0.9825
Apen	0.935716	0.9925
Serial	0.117089	0.9825
Linear-Complexity	0.779188	0.9925

4. RANDOMNESS PERFORMANCE EVALUATION USING A PRACTICAL INFORMATION MEASURE

A true random number generator produces random numbers by sampling the entropy source, and converting analog samples into digital bits. The theoretical requirements for generating independent, and identically distributed bits from a DT chaotic system is studied in Chapter 3 using the proposed mathematical models. The TRNG models introduced in Chapter 3 are based on symbolic dynamics that allow translation of real valued chaotic time series into symbolic binary strings of ones, and zeros using the spatio-temporal location of the samples in the partitioned phase space.

In order to make a comparative study of the randomness performances of chaotic systems of interest, a practical information measure that can qualify the randomness of the finite bitstream generated by mathematical models is needed. Conventional statistical tests are designed for detecting any periodic, or correlated bit patterns, but their binary, pass/fail type outputs can not be used to understand the underlying randomness capacity of the entropy source. Thus, it is necessary to use an information measure, and calculate the entropy of finite length bit stream to compare the randomness performance of chaotic maps.

A vocabulary based practical information measure, called T-entropy, has been used to calculate the entropy of the generated finite bitstream [59]. T-entropy calculation is based on a recursive hierarchical pattern copying (RHPC) algorithm, called T-decomposition, which parses the bitstream in terms of bit patterns, while accounting for consecutive repetitions of each pattern. Recursive bit pattern identification approach enables the algorithm to detect any existing long range dependencies, and structures in patterning. T-entropy of a finite bitstream is calculated based on the complexity of the RHPC algorithm [60].

4.1. Calculation of the Entropy of a Finite Bit Stream Using T-Complexity

Entropy has long been regarded as an indicator of disorder. Boltzmann was first to provide a statistical formalism to this notion in the 19th century. Much later, in 1948, following the foundation of the mathematical theory of communication, Shannon used the entropy as a measure of information in coded messages [56, 57]. Shannon's main contribution was the interpretation of entropy as information by providing a probabilistic formulation of entropy in which the source probability distribution is assumed to be known. On the other hand, in many practical applications, any a priori information about the information source does not exist. While practically being an attractive subject, it is not possible to talk over the information content of a finite length string as it is not defined by Shannon's theory. This fact led Kolmogorov to develop the algorithmic information theory in 1958, allowing one to express the corresponding entropy of a dynamic system by using Shannon's entropy through symbolic encoding of the phase space using partitions [50]. Kolmogorov-Sinai entropy (KS-entropy) which can be defined as the supremum of the Shannon entropies calculated over all possible phase space partitions, is a measure of the average rate at which information is lost by a dynamic system about its former state, and therefore it can precisely quantify the unpredictability of the dynamic system. While direct calculation is not possible, in 1977, Pesin proved that the KS-entropy of certain non-linear dynamical systems is given precisely by the sum of corresponding positive Lyapunov exponents [49]. There exist a number of practical techniques for quite precisely computing Lyapunov exponents from the observed time series [60, 61]. Thus the KS-entropy can be computed indirectly without a priori knowledge of source statistics.

In 1997, Titchener proposed a new measure of information [59] based on recursive hierarchical pattern copying (RHPC) algorithm performing better than recursive linear pattern copying (RLPC) algorithm based complexity measure developed by Lempel and Ziv [62]. Although more sophisticated, so called T-decompositon algorithm consumes less vocabulary for entropy calculation of a string.

The RHPC algorithm parses a string of interest composed of symbols from alphabet A from left to right, and lifts the patterns p_i from the string right to left, and accounts for consecutive repetitions of each pattern with corresponding integer parameters, k_i. The decomposition derived for string x is of the form $x = p_t^{k_t} p_{t-1} ... p_1^{k_1} \alpha$ such that $\alpha \in A$, constrained by further requiring each sub pattern to have the form $p_i = p_{i-1}^{m_{i-1,i-1}} p_{i-2}^{m_{i-2,i-1}} ... p_1^{m_{i-1,1}} A$, with $0 \leq m_{j-1} \leq k_i$. The resulting decomposition of a string is thus given in terms of patterns p_i, which tend to maximize the reuse of patterns p_j, such that $j < i$, minimizing the total number of steps t required to generate the string. t measures the depth of recursive pattern copy hierarchy [63]. Instead of using depth t to measure the complexity, a further adjustment is done to allow for the repetition of prefix patterns where $k_i > 1$. The RHCP complexity can be defined as $C_T(x) = \sum_i log_2(k_i + 1)$. The steps are called taugs (abbreviation of T-augmentation steps) [63].

For instance, let $A = \{0, 1\}$, and $x = 0100010101101$. The algorithm works as the following [63]

(i) Select the ultimate (pattern) character (left to right)
$x = 010001010110\underline{\mathbf{1}}$ thus set $\alpha = 1$.

(ii) Select the penultimate character in $x = 01000101011\underline{\mathbf{0}}\mathbf{1}$ so $p_1 = 0$.

(iii) Determine the number k_1 of times this pattern repeats as a consecutive run, in this position: p_1 appears just once so $k_1 = 1$

(iv) Parse string left to right grouping each occurrence of the pattern p_1 (or consecutive run of up to k_1 occurrences of p_1 if $k_1 > 1$) with an immediately following pattern yielding $x = .01.00.01.01.01.01.1.01$.

(v) Repeat step (ii): select the penultimate pattern in the newly grouped sequence, thus set $p_2 = 1$. Repeat step (iii): p_2 does not, in the positions immediately to the left, repeat more than once so set $k_2 = 1$. Repeat step (iv), re-parse left to right to make new groups from each occurrence (or run less than equal to k_2) of p_2 with an immediately following pattern: $x = .01.00.01.01.01.101$.

(vi) Repeating steps (ii, iii, iv): select the penultimate pattern $p_3 = 01$. (Since p_3 repeats three times set $k_3 = 3$). Parsing left to right to make new groups

$x = .0100.010101101.$is obtained. Since $k_3 > 1$, the process of forming groups, allows $0 < m_i \leq k_i$ consecutive copies of the current pattern p_3, to be grouped with the next available pattern.

(vii) Repeating steps (ii, iii, iv) for the last time: we select penultimate pattern $p_4 = 0100$ occurring just once so $k_4 = 1$. Parsing left to right to form new groupings results $x = .0100010101101$. Since only the one group is remaining, calculation is finished.

Finally the result is calculated as $x = (0100)^1(01)^3(1)^1(0)^10$. Each pattern may be written in terms of its predecessor patterns that is $p_4 = 0100 = (01)^1(1)^0(0)^10$, $p_3 = 01 = (0)^0(1)^1, p_2 = (0)^01, p_1 = 0$. Instead of using hierarchy depth t, $C_T(x) = \sum_i log_2(k_i+1)$ can be defined, and used to calculate the steps:

$$log_2(1+1) + log_2(3+1) + log_2(1+1) + log_2(1+1) = 5 \quad steps, (5 \quad taugs)$$

T-augmentation process can be applied to produce strings that for a given length n have a maximum vocabulary t, and therefore complexity t. Titchener was able to derive an empirical expression for the upper bound of complexity as $C_T(x) \leq li(log_e(\#A)|x|), x \in A^+$ where # denotes the cardinality, and the logarithmic integral function is given by $li(z) = \int_0^Z \frac{du}{log_e(u)}$. One may derive the lower bound for RHPC complexity by considering a single repeating character as $C_T(x) \geq log_2(|x|)$.

The T-information $I_T(x(n))$ is defined to be the inverse logarithmic integral of the T-complexity divided by a scalar constant $ln2$ with units in nats [63].

$$I_T(x(n)) = li^{-1}(\frac{C_T(x(n))}{ln2}) \tag{4.1}$$

is the total information for $x(n)$ in nats (It is possible to convert units to bits by dividing I_T by $ln2$). In the limit case as $n \to \infty$, $I_T(x(n)) \leq ln(\#A^n)$ where the right hand side expression corresponds to n-block Shannon entropy [56]. The average T-information rate per symbol, referred as the average T-entropy of $x(n)$ is denoted

as

$$h_T^-(x(n)) = (I_T(x(n)))/n \quad (nats/symbol). \tag{4.2}$$

T-information, and average T-entropy can be calculated from T-complexity. Recursive approach to identifying pattern structures along the string enables the algorithm to detect long range dependencies, and structures in patterning [63]. T-entropy calculation is highly sensitive to both local, and global pattern structures, and produces entropy values that are indicators of the Shannon n-block entropy for large $n \approx (10^4 - 10^6)$ [56].

4.2. Randomness Performance Evaluation of Single Entropy Core TRNG

A reliable true random number generator is required to produce an acceptable level of randomness throughout its operational lifetime. Thus, it is important to understand the effects of potential parameter variations on the entropy of the generated bitstream. T-entropy can be used as a practical information metric for estimating the randomness performance of a chaotic system which will be employed as an entropy source in the single entropy core TRNG architecture. T-entropy of the generated bitstream reveals the randomness characteristics of a chaotic system as a function of critical system parameters such as chaos control parameter, or bit generation threshold. Thus, the effects of parameter variation on the randomness can be estimated ahead of physical implementation using the mathematical model of the TRNG architecture. In addition, maximum allowable limits of parameter variations can also be determined for circuit design. Consequently, proposed approach can be used to qualify the feasibility of a chaotic system that will be as a source of randomness in a TRNG application.

In order to qualify the randomness characteristic of a chaotic system, and determine the maximum allowable boundaries of parameter variations that attain an acceptable level of entropy, formerly introduced recursive hierarchical pattern copying T-decomposition algorithm is employed, and the T-entropy of the bitstream generated by the mathematical model has been calculated. The elements of the parameter set that affects the randomness performance (chaos control parameter, and threshold)

are varied throughout their respective domains, and at each step, T-entropy of the bitstream is calculated, and recorded for each particular parameter set. Then, the recorded data set is used to construct a 3D projection of the T-entropy of the TRNG model generated bitstream. The appearing projection is basically a 3D bifurcation diagram with z-axis being the entropy of the model generated bitstream. The 3D T-entropy projection can be used to explore the limits of maximum permissible variations in chaos control parameter, and comparison threshold for a particular loss in the entropy level. It is possible to find out a maximum entropy harvesting window with respect to comparison threshold, and chaos control parameter intervals. The size of this window provides very valuable information to the hardware designer by showing the variation boundaries of parameters for a certain level of entropy loss.

The single entropy core DT chaos based TRNG model shown in Figure 3.1, has been implemented in MATLAB to generate random bits. For each stepped value of chaos control parameter R, a vector of random bits were generated by comparing the map iteration vector to comparison threshold vector. Then, the T-entropy of the vector has been calculated, and recorded, hence creating entropy slices on each iteration. When all the slices are joined at the end of calculation process, a 3D projection of T-entropy is obtained. It is interesting to note that the vertical bird eye view of the 3D projections in Figure 4.1, Figure 4.2, and Figure 4.3, correspond to the bifurcation diagrams of the respective chaotic maps of interest as presented in Figure 2.2, Figure 2.7, and Figure 2.12. These 3D bifurcation plots are strong functions of Lyapunov exponents, and practically establish the relation between the entropy of the generated bitstream, and Lyapunov exponents in accordance with the Pesin Theorem [49].

The rich chaotic dynamics of the logistic map appears in the T-entropy plot presented in Figure 4.1. The stability islands found in the bifurcation diagram shown in Figure 2.2 can be observed in Figure 4.1 as in the form of entropy gaps, meaning that no usable randomness exists at those regions, and they should be avoided in TRNG applications. Apparently, logistic map is not the best choice of entropy source unless strict control of the chaos control parameter is possible. Variations in chaos control parameter can easily put the dynamic system out of chaos, and disable random number

generation. The maximum achievable entropy level is calculated as 0.693 which is in good agreement with the maximal positive Lyapunov exponent of the logistic map calculated in Chapter 2.

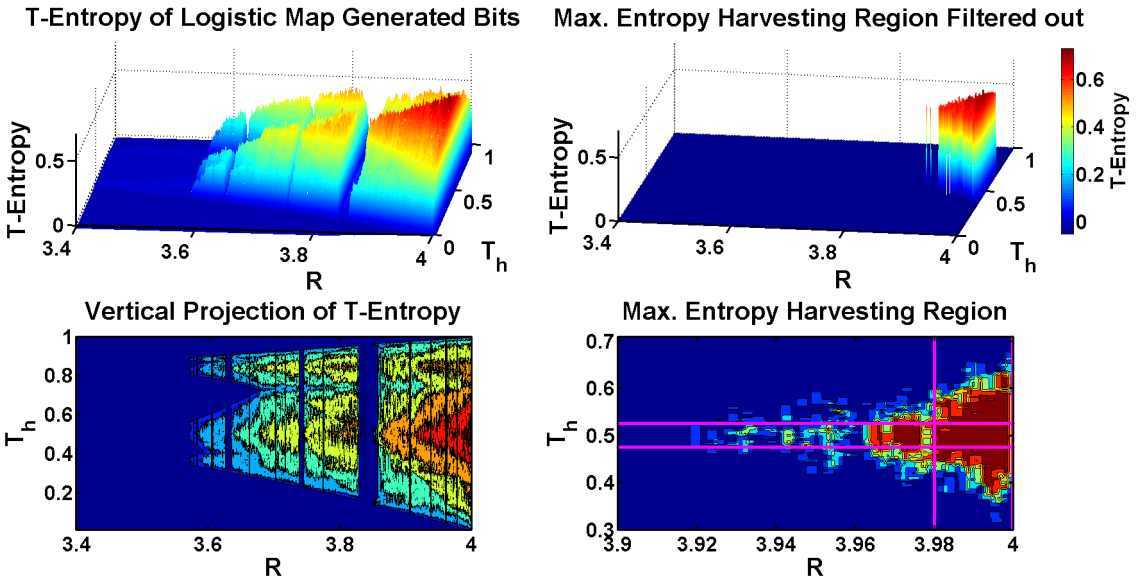


Figure 4.1. T-entropy of the bitstream generated by logistic map based single entropy core TRNG model, and calculated maximum allowable deviation boundaries for an entropy loss of 0.01.

Maximum entropy region is detected using custom filtering of T-entropy data with respect to an entropy loss of 0.01. Hence, the lower bound on entropy level is chosen as 0.683, and the regions below this boundary are filtered out. The vertical projection of the 3D plot shown in Figure 4.1 is windowed for determining maximum entropy region in both axes. When the center of geometry is taken as the reference point, deviation limits in R, and T_h axes are obtained. The single entropy core DT chaos based TRNG model using logistic map as the entropy source is found to have, 1% deviation tolerance for chaos control parameter R, and 2.5% deviation tolerance for bipartition threshold T_h, under the assumption of 0.01 maximum allowable entropy loss.

The robust chaotic dynamics of the tent map can be observed in the T-entropy plot shown in Figure 4.2. No stability islands appear in Figure 4.2 as in the case of bifurcation diagram of the logistic map shown in Figure 2.2. Tent map exhibits no

entropy gaps, meaning that a wide range of usable randomness exists which renders the tent map as a robust entropy source. This is a very desirable feature for an chaotic system that will be employed as the source of randomness in a TRNG application. Tent map is more immune to control parameter variations when compared to logistic map. The tent map is a convenient choice of entropy source, since parameter variations cannot put the dynamic system out of chaos easily. The maximum achievable entropy level is calculated as 0.693 which corresponds to the maximal positive Lyapunov exponent of the tent map that has been theoretically calculated in Chapter 2.

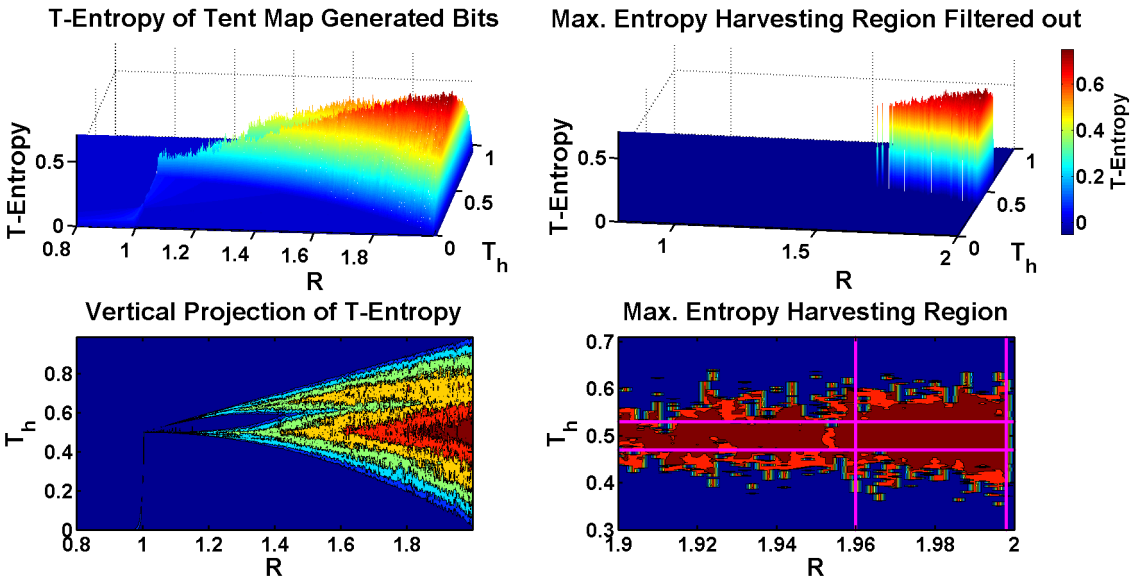


Figure 4.2. T-entropy of the bitstream generated by tent map based single entropy core TRNG model, and calculated maximum allowable deviation boundaries for an entropy loss of 0.01.

The single entropy core DT chaos based TRNG model using tent map as the entropy source is found to have, 2% deviation tolerance for chaos control parameter R, and 1% deviation tolerance for bipartition threshold T_h, under the assumption of 0.01 maximum allowable entropy loss.

T-entropy for the Bernoulli map based TRNG model generated bitstream has been calculated as shown in Figure 4.3. Bernoulli map exhibits no entropy gaps, offering a wide range of usable randomness which renders itself as a robust entropy source. As in the case of tent map, Bernoulli map has no stability islands which pro-

motes its use as an entropy source in TRNG applications. Bernoulli map becomes an appropriate choice of entropy source as a result of its robust chaotic behavior, variations in chaos control parameter cannot easily put the dynamic system out of chaos. The maximum achievable entropy level for the Bernoulli map has been calculated as 0.693 matching to its maximal positive Lyapunov exponent, which was theoretically calculated in Chapter 2.

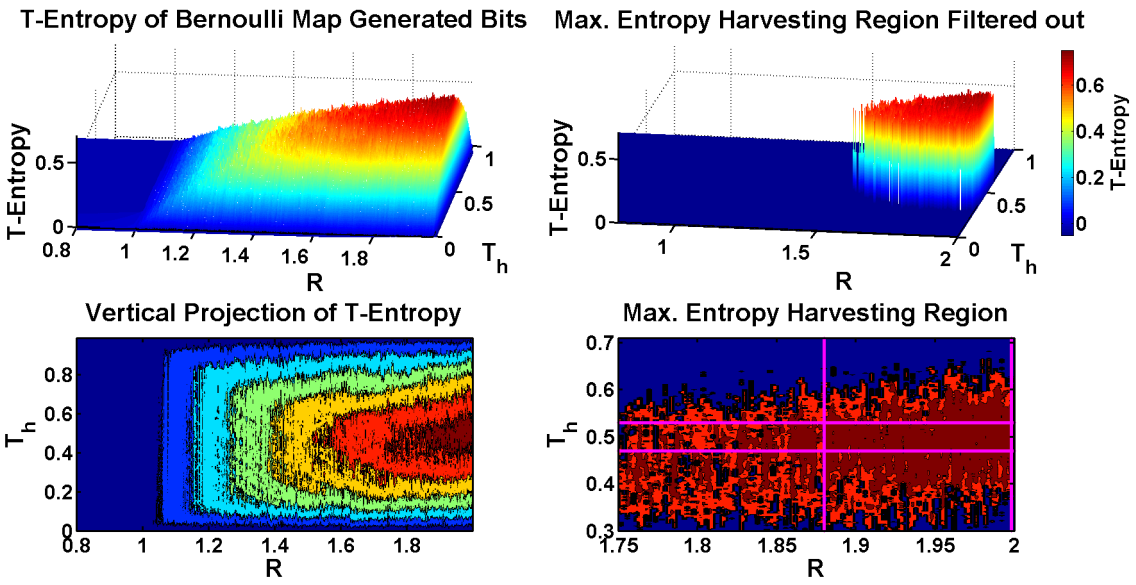


Figure 4.3. T-entropy of the bitstream generated by Bernoulli map based single entropy core TRNG model, and calculated maximum allowable deviation boundaries for an entropy loss of 0.01.

Table 4.1 summarizes maximum allowable variation limits for the 1D chaotic maps of interest where η is the chaos control parameter that corresponds to R for logistic map defined by Equation 2.6, μ for the tent map defined by Equation 2.15, β for Bernoulli map defined by Equation 2.22, and T_h is the comparison threshold. According to Table 4.1, tent map, and Bernoulli map have better immunity to parameter variations when compared to the logistic map. On the other hand logistic map has wider tolerance for threshold deviations. The parameter variation analysis of entropy showed that as long as the deviations are within estimated limits given in Table 4.1, entropy of the generated bitstream is guaranteed to be very close to the maximum achievable value.

Table 4.1. Maximum allowable parameter tolerances for single entropy core DT chaos based TRNG models.

Maps	$\boldsymbol{\eta}$	$\boldsymbol{\Delta\eta}$	$\mathbf{T_h}$	$\boldsymbol{\Delta T_h}$
Logistic	3.99	$\mp 1\%$	0.5	$\mp 2.5\%$
Tent	1.98	$\mp 2\%$	0.5	$\mp 1\%$
Bernoulli	1.940	$\mp 6\%$	0.5	$\mp 1\%$

4.3. Randomness Performance Evaluation of Dual Entropy Core TRNG

The randomness characteristics of the dual entropy core DT chaos based TRNG architecture can be explored using the same approach that has been employed to study its single entropy core precursor. In randomness performance evaluation, T-entropy has been used as a metric to calculate the randomness of the generated finite bitstream [59]. We have chosen the Bernoulli map for each entropy core, in order to make a fair performance comparison with the single entropy core architecture. The T-entropy of the bitstreams generated by both single, and dual entropy core TRNG models have been calculated with the help of numerical simulations, for all possible values of parameters affecting the randomness performance. 3D projections of T-entropy calculations are plotted in Figure 4.4, and Figure 4.5 for comparing the randomness performances.

In both cases, it is possible to observe that T-entropy increases as the chaos control parameter(s) μ approach to its maximum value, two, at which the entropy of the generated bitstream achieves its maximum. In the single entropy core case, any deviation in the comparator threshold parameter T_h can drastically reduce the maximum achievable entropy level of 0.693 as observed in Figure 4.4. The entropy dependency on the threshold parameter T_h is noticeably higher than that of the chaos control parameter μ for the single entropy core architecture.

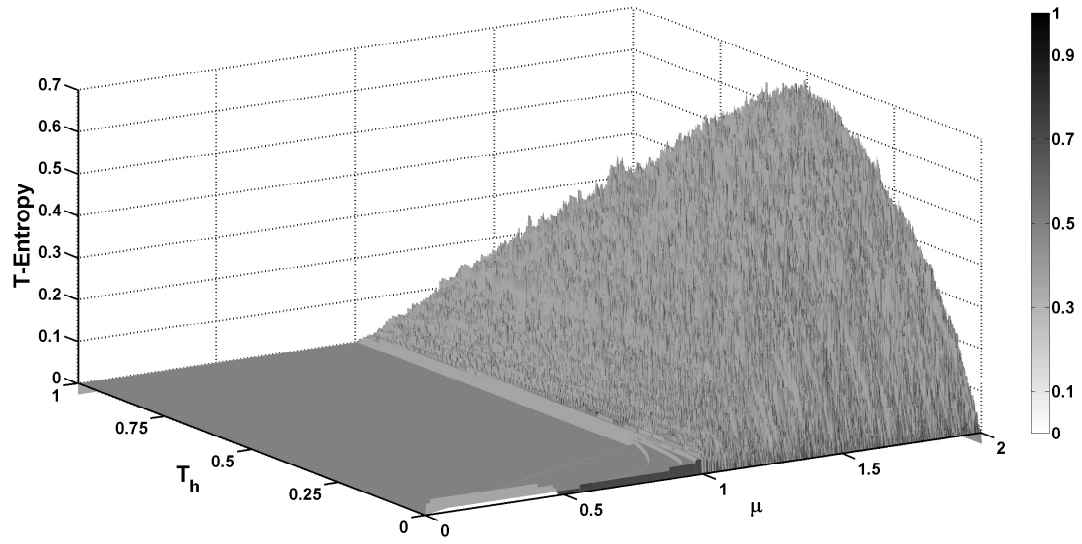


Figure 4.4. T-entropy of the bitstream generated by single entropy core TRNG model.

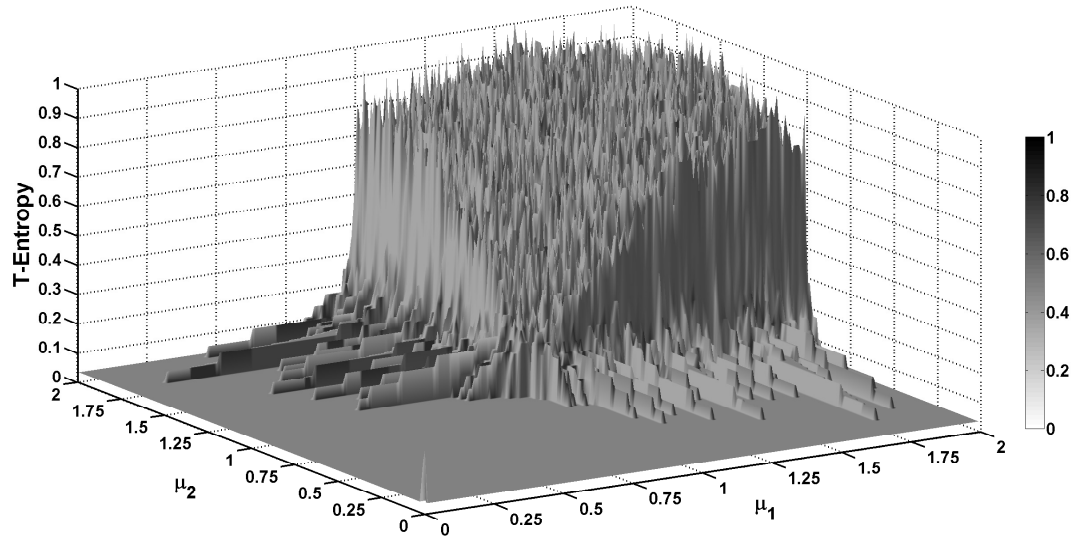


Figure 4.5. T-entropy of the bitstream generated by dual entropy core TRNG model.

On the other hand, in the dual entropy core case shown in Figure 4.5, both the maximum achievable entropy level, and the associated control parameter intervals are larger, which enables generation of high entropy bits for a wider range of parameter values. Figure 4.6 provides a better understanding of randomness characteristics of

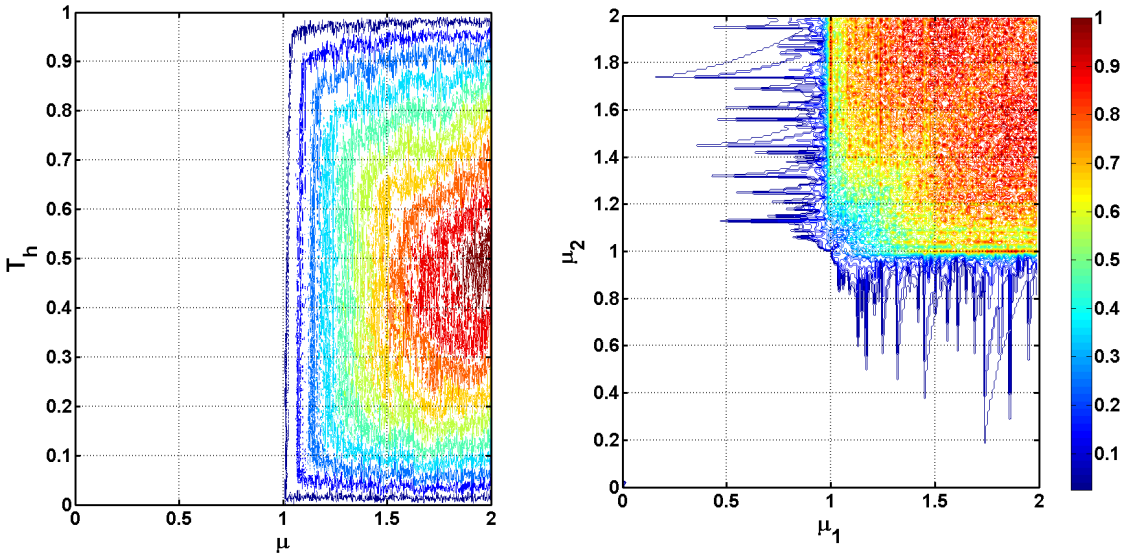


Figure 4.6. Vertical Projection of the T-entropy plots for the bitstreams generated by single, and dual entropy core TRNG models.

both architectures using the side by side vertical projections of the T-entropy graphs of single, and dual entropy core TRNG model generated bitstreams. According to Figure 4.6, the dual entropy core TRNG model generates higher entropy for a wider range of chaos control parameters μ_1, μ_2, and it is less sensitive to parameter variations. Dual entropy core TRNG architecture is distinguished by its maximum achievable entropy level in the excess of 0.9 for a wide range of chaos control parameter values, which outperforms its highly parameter sensitive counterpart that can achieve a maximum entropy level of 0.693.

The dual entropy core TRNG architecture is remarkably less sensitive to deviations in the chaos control parameters, which manifests itself as a unique advantage in the implementation process. In analog design, sensitive parameters such as T_h of the single entropy core TRNG architecture can not be controlled precisely as a result of matching, and resolution limitations associated with the implementation technology [40]. The critical dependence of entropy on T_h renders single entropy core architecture practically inefficient. In addition, it should be noted that absence of the threshold generator, and all the design complexity associated with it reduces design overhead only at a small expense of hardware redundancy.

5. FPAA IMPLEMENTATION OF DT CHAOS BASED TRNGS

Field programmable analog arrays (FPAA) are cost effective reconfigurable platforms for fast prototyping of analog circuits [38, 39, 42]. FPAAs are basically integrated circuits, that contain configurable analog blocks (CABs) with interconnects between them. Unlike their digital counterpart, the field programmable gate array (FPGA), they may be voltage mode, or current mode devices. For voltage mode variants, each block usually contains an operational amplifier which is combined with programmable array of passive components. These blocks can operate as summers, integrators, comparators, or filters. There are mainly two operating modes: continuous time, and discrete time. Continuous time devices operate like an array of transistors, or op amps which can be utilized at their full bandwidth. The components can be connected in a particular arrangement through reconfigurable arrays of switches. Usually, parasitic inductance, and capacitance of the switch matrix, and noise contributions are primary performance limiting factors that must be taken into account in the design process [64].

Discrete time operating FPAAs employ a reference system clock, and are active only during a portion of the time. An internal programmable clock synthesizer generates all the required sampling clock signals for CABs within the chip. In a switched capacitor type FPAA, all CABs sample their input signals with a sample and hold circuit that consists of a semiconductor switch, and a capacitor. Analog samples are fed to a programmable OPAMP circuit which can be routed to a number of other CABs. This flexibility comes at the cost of increased complexity, and reduced bandwidth. Switched current type FPAAs, offer simpler architecture, and omit the input capacitor, but they have less accuracy, and limited fan out.

We have chosen a commercial FPAA integrated circuit (AN231E04) based on switched capacitor technology as the implementation platform for the proof of concept design of the DT chaos based TRNGs, since it allows the realization of DT systems [65].

5.1. FPAA Implementation of a Single Entropy Core DT Chaos Based TRNG

In order to realize a single entropy core DT chaos based TRNG, a non linear transfer function for implementing the map, a sample and hold circuit for driving the chaotic dynamics, a threshold generator, and a comparator for generating random bits are required as suggested by Figure 3.1.

The aforementioned FPAA chip contains all the blocks required for implementation of the three chaotic maps discussed previously. The logistic map defined by Equation 2.6 is chosen since it exhibits rich dynamic behavior, and its implementation as a chaos generator is popular in the literature [38, 39, 66–71]. Logistic map is implemented around a signed adder, a multiplier, and an inverting gain block as presented by Figure 5.1.

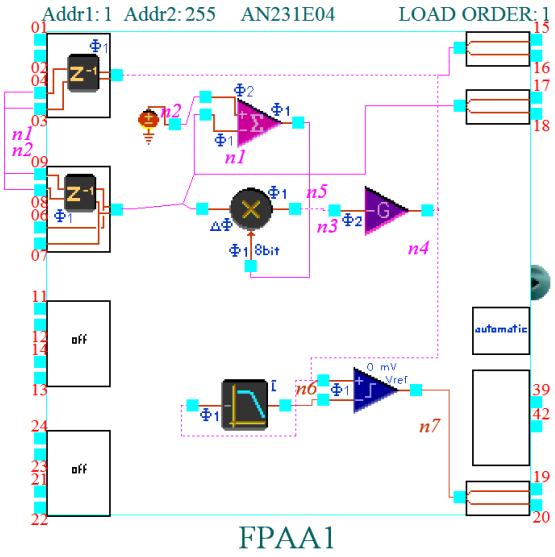


Figure 5.1. FPAA implementation of logistic map based single entropy core TRNG.

A low pass filter is used to calculate the running average of the chaotic signal generated by the logistic map. The cutoff frequency of the filter is set to a value much lower than the half of the operating frequency of the multiplier component for creating an adaptive threshold for the comparator. The comparator generates bits by comparing the chaotic signal with its running average to generate an identically distributed bitstream in which every bit is equally likely to appear. The FPAA chip is powered by 3.3V DC, and driven

by a 16MHz master clock, readily available on the development board. Master clock is used to synthesize the internal clock signals required by CAB components [72].

5.1.1. Measurement Results

The non-linear dynamics of logistic map based TRNG is driven by sample-hold circuits operating at 2MHz. The phase portrait of the logistic map is measured as presented in Figure 5.2

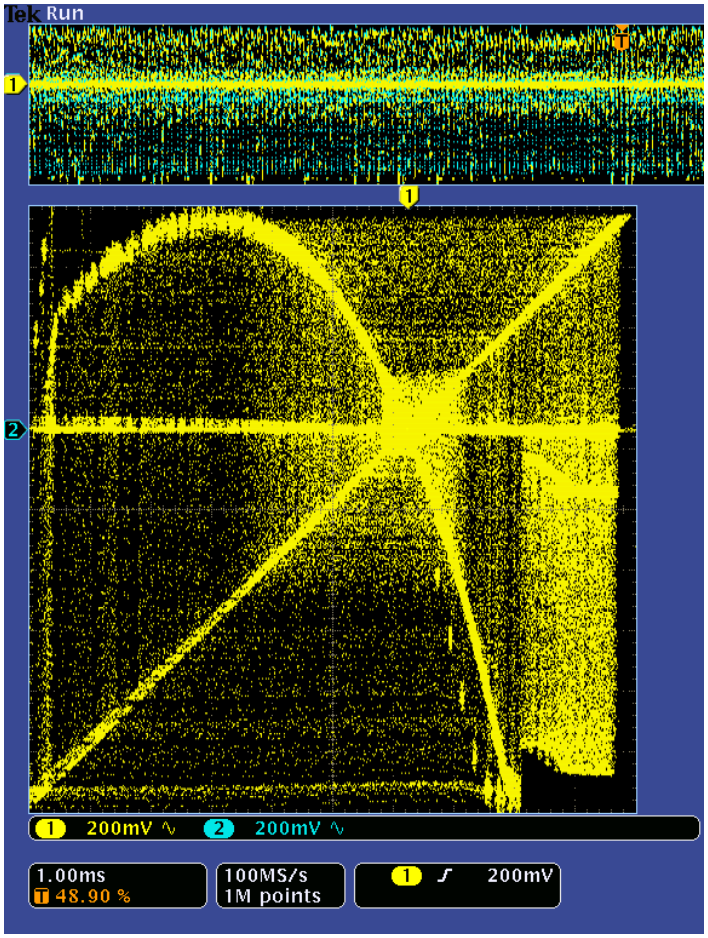


Figure 5.2. Phase portrait of the logistic map implemented on FPAA.

Although the analog blocks of the FPAA can operate up to 4MHz, the multiplier block puts a fundamental limit on the throughput as a result of its maximum clock limitation at 250KHz. Thus, an average throughput around 200Kbps is obtained. A Spartan XC3S1600E Microblaze FPGA development board is used to acquire bits for statistical testing.

5.1.2. Statistical Test Results

400Mbits of bitstream acquired by the FPGA is transferred to the computer using the UART interface on the FPGA development board [73]. An average throughput of 200Kbps is achieved at first, but the acquired bitstream failed the frequency test which accounts for the uniform distribution of ones, and zeros. The source of statistical bias may be switching noise, component tolerances, or offsets. In order to solve the bias problem, a post processor that XORes consecutive bits has been employed in the FPGA which effectively reduced the average throughput to 100Kbps. NIST800.22 statistical test suite v2.0 is used for the statistical testing of the processed bitstream. As presented in Table 5.1, each p-value corresponding to a particular test describes the probability of the bitstream generated by an ideal TRNG [33].

Table 5.1. NIST800.22 statistical test results for the bitstream generated by logistic map based single entropy core TRNG.

Test	**P-Value**	**Proportion**
Frequency	0.474986	1.0000
Block Frequency	0.075719	0.9800
Cumulative Sums	0.719747	1.0000
Runs	0.115387	0.9800
Longest-Run	0.249284	0.9800
Rank	0.61630	0.9800
FFT	0.514124	0.9900
Universal	0.037566	0.9700
Apen	0.678686	0.9900
Serial	0.334538	0.9800
Linear-Complexity	0.153763	0.9900

NIST800.22 statistical test suite divides the raw bitstream into 1Mbit blocks, and applies the test suite. The proportion column in Table 5.1 shows the ratio of 1Mbit sequences passing the particular NIST800.22 test. According to the results in Table 5.1, the acquired bitstream passes all NIST800.22 test with the help of XOR post processing.

5.2. FPAA Implementation of a Dual Entropy Core DT Chaos Based TRNG

The dual entropy core architecture introduced in Chapter 3 is composed of two DT chaotic maps that evolve in time with the help of respective sample and hold blocks that drive the non linear dynamics by a clock signal as shown in Figure 3.6. Two chaotic signals created by uncorrelated, and uncoupled maps are connected to the inputs of a comparator to generate random bits. For the sake of simplicity, both maps are configured as Bernoulli maps with identical parameters. The initial conditions for the maps are guaranteed to be different with the help of existing additive white Gaussian noise at the circuit nodes of the chip. Different CABs are used for implementing each entropy core to prevent any coupling, and synchronization between chaotic maps [8].

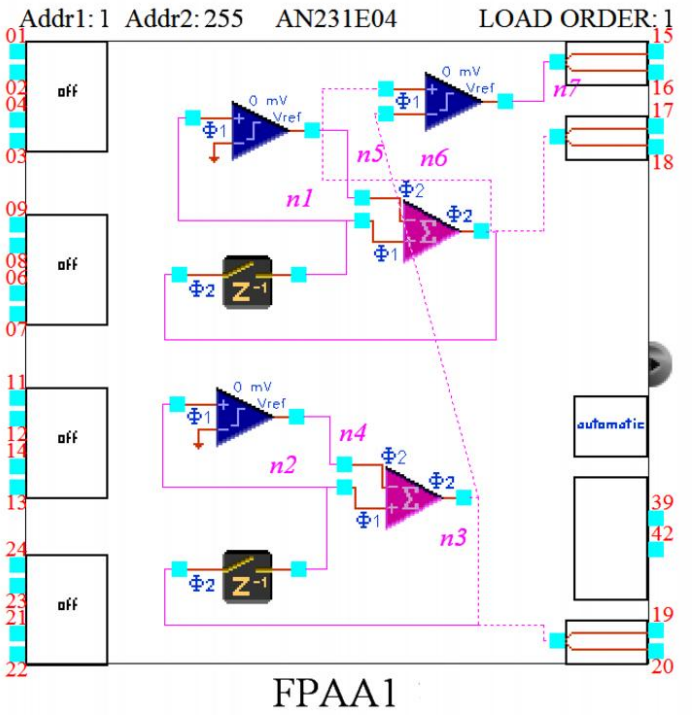


Figure 5.3. FPAA implementation of Bernoulli map based dual entropy core TRNG.

Each entropy core requires a non-linear function block, and a sample and hold block that can be built using the resources already available in the CABs within the FPAA chip [65]. The non-linear function block is designed around an analog adder, and a comparator circuit to implement the Bernoulli map defined by Equation 2.22

as shown in Figure 5.3 [74–76]. In addition, the bit extractor function defined by Equation 3.16 is realized by an additional comparator within the available CABs as shown in Figure 5.3 [42]. The FPAA chip is powered by 3.3V DC, and driven by a 16MHz master clock on the development board shown in Figure 5.5, which is used to synthesize the internal clock signals required by CAB components [72]. The non-linear dynamics of dual Bernoulli map based TRNG is driven by sample-hold circuits operating at 2MHz within CABs.

5.2.1. Measurement Results

The measurement setup illustrated in Figure 5.4 is built around an AN231E04 FPAA development board, and a Spartan XC3S1600E Microblaze FPGA development board for evaluating the randomness performance of the proof of concept dual entropy core TRNG architecture as demonstrated in Figure 5.5 [72, 73].

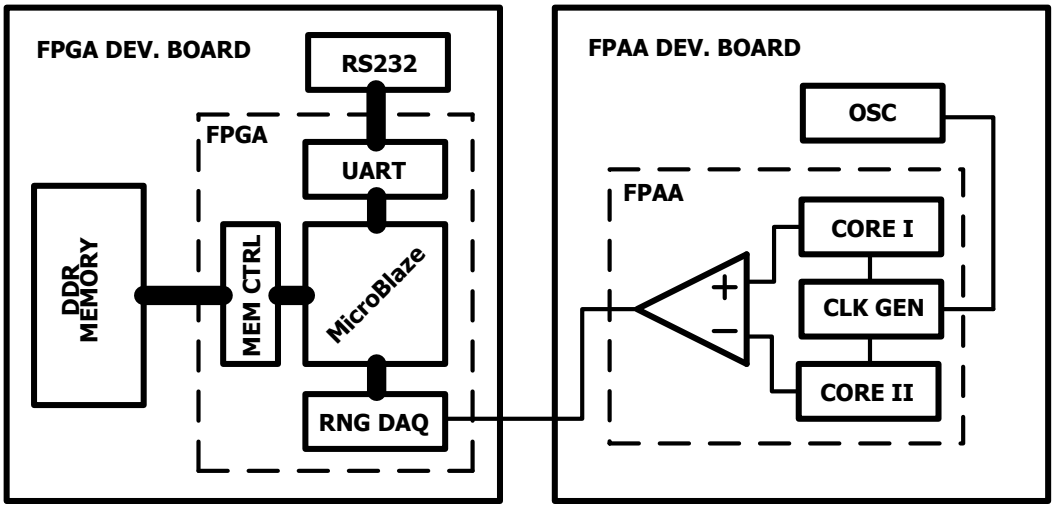


Figure 5.4. Measurement setup for the dual entropy core DT chaos based TRNG.

The FPGA development board shown in Figure 5.5 is used to acquire, and transfer the bitstream generated from FPAA development board. For this purpose, a custom 32bit single core Microblaze microcontroller running at 50MHz with DDR Memory, and UART interfaces is implemented using the embedded development kit of the chip vendor. A data acquisition module with programmable sampling clock has been integrated as a custom peripheral within the processor. A 32bit shift register which is driven by the programmable clock converts the serially acquired random bits into 32 bit

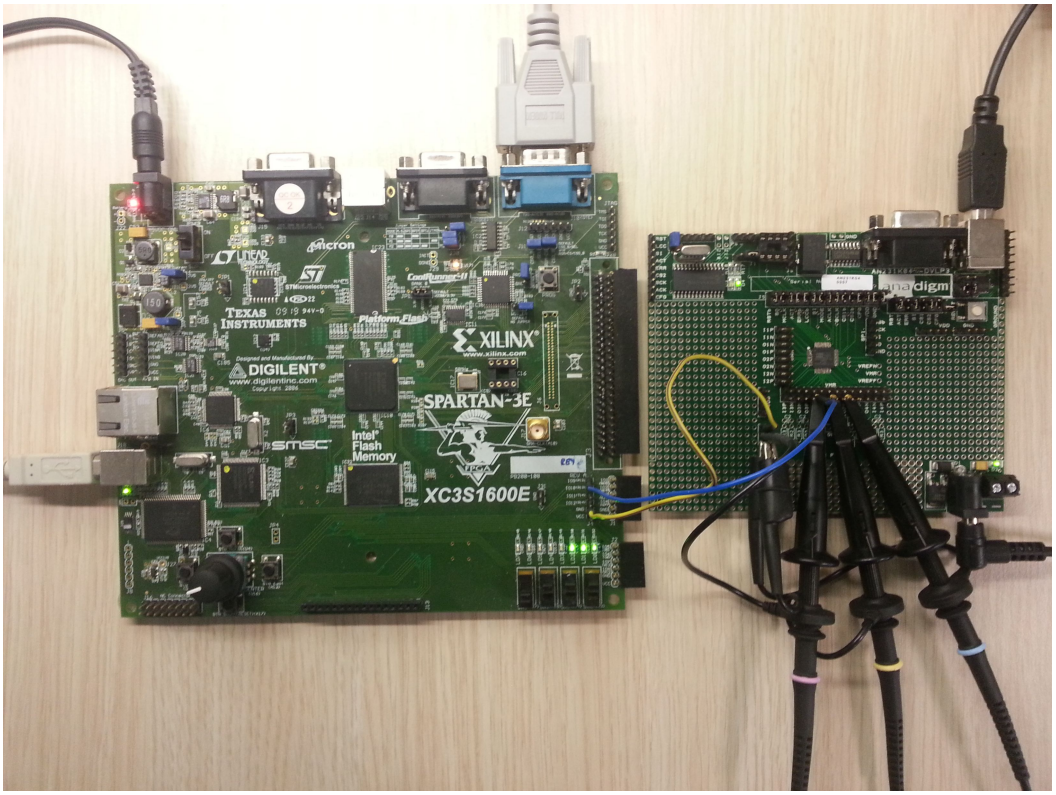


Figure 5.5. Proof of concept implementation of the dual entropy core DT chaos based TRNG architecture.

parallel data which is then read by the software running on microblaze, and transferred to external DDR SDRAM for temporary storage. After the end of data acquisition session, random number data stored in the external DDR SDRAM is transferred to computer using the ubiquitous RS232 interface. A custom software on the computer side records the incoming data to a file in binary format for statistical analyses.

Chaotic signals generated by each entropy core in the FPAA chip, and random bits generated by the comparator are shown in the oscilloscope screenshot presented in Figure 5.6. As a result of the clock speed limitations imposed by the implementation technology of the FPAA chip, an average throughput in the excess of 1.5Mbps is achieved by the proof of concept circuit as shown in Figure 5.6.

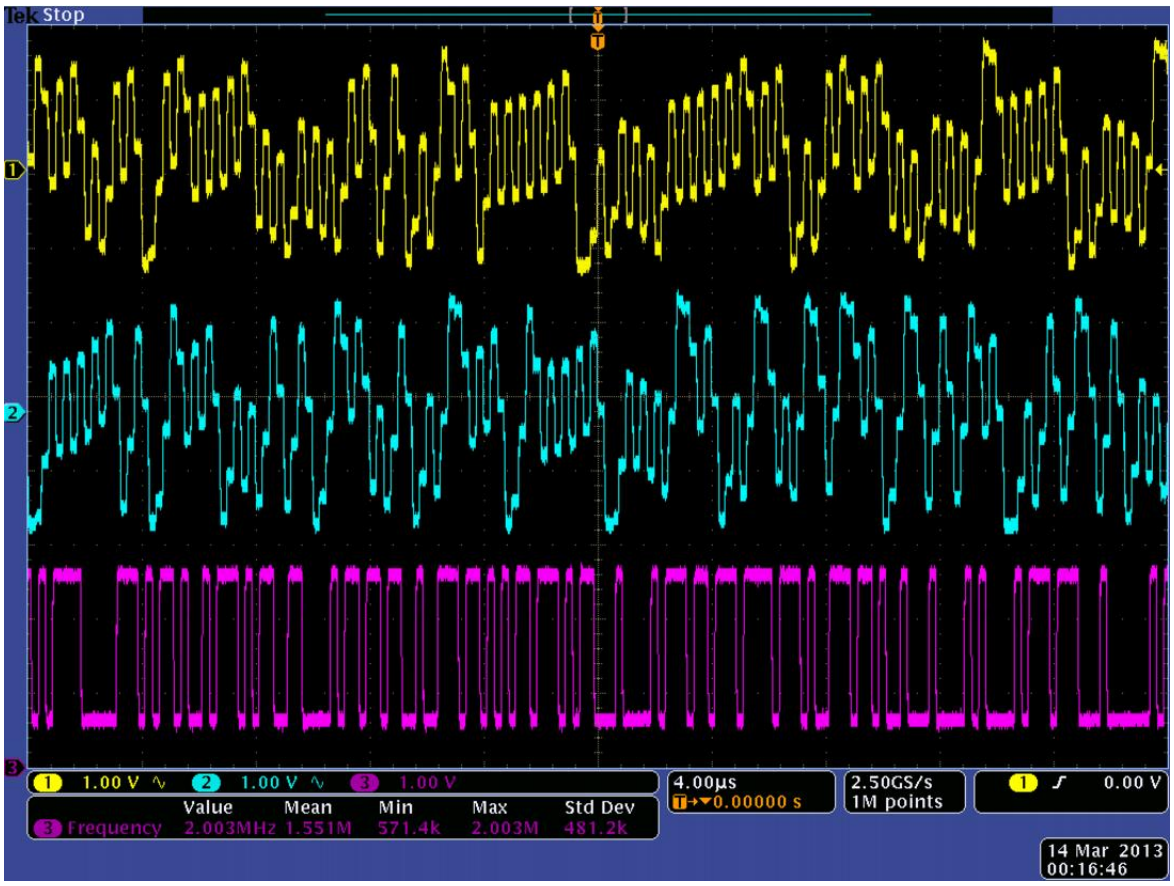


Figure 5.6. Operation of the dual entropy core DT chaos based TRNG architecture implemented on FPAA.

5.2.2. Statistical Test Results

While no set of statistical tests can absolutely qualify a TRNG, they are useful in determining the statistical performance for the cryptographic applications. NIST800.22 statistical test suite v2.0 is used for the statistical testing of the acquired bitstream. 400Mbits of data have been captured, and transferred to computer for statistical tests using the setup shown in Figure 5.5. As presented in Table 5.2, each p-value corresponding to a particular test describes the probability of the bitstream generated by an ideal TRNG [33]. NIST800.22 statistical test suite divides the raw bitstream into 1Mbit blocks, and applies the test suite. Proportion column in Table 5.2 shows the ratio of 1Mbit sequences passing the particular NIST800.22 test. According to the results in Table 5.2, the acquired bitstream successfully passes all NIST800.22 statistical tests.

Table 5.2. NIST800.22 statistical test results for the bitstream generated by Bernoulli map based dual entropy core, TRNG.

Test	**P-Value**	**Proportion**
Frequency	0.904708	0.9975
Block Frequency	0.783973	0.9900
Cumulative Sums	0.549331	0.9850
Runs	0.605916	0.9800
Longest-Run	0.432672	0.9950
Rank	0.783973	0.9900
FFT	0.366918	0.9900
Universal	0.319084	0.9800
Apen	0.585209	0.9975
Serial	0.968128	0.9925
Linear-Complexity	0.978072	0.9875

Dual entropy core DT chaos based true random number generator architecture can enhance the randomness of the generated bitstream using hardware redundancy as suggested by Figure 4.5, and Figure 4.6. Proposed architecture can be used to improve the randomness performance of 1D chaotic map based TRNGs. Unfortunately, the design flexibility offered by the reconfigurability of FPAA comes with the inevitable cost of reduced speed. However, an ASIC implementation of the dual entropy core DT chaos based TRNG architecture can overcome the throughput limitations of the FPAA chip.

6. ASIC IMPLEMENTATION OF DT CHAOS BASED DUAL ENTROPY CORE TRNGS

The pioneering efforts in very large scale integrated (VLSI) circuit technology has led the creation of application specific integrated circuits (ASIC)s which are customized chips designed to satisfy the requirements of a particular purpose, or application. The evolutionary trend in VLSI circuit technology has been traditionally driven by furious industrial competition to bring down the cost, and area of the integrated circuits. As of today, progressive advances in the VLSI fabrication technology made it possible to design, and implement transistors with channel widths as small as $20nm$.

The use of field programmable analog arrays reduces the complexity of analog design, decreases time to market, and allows designs to be easily updated, and improved in the operating environment. However, as we have witnessed in Chapter 5, their flexibility comes with the inevitable cost of reduced performance due to factors associated with their complex architecture. High power consumption, and low throughput demote their use in emerging mobile cryptographic applications that require both low power, and high speed operation. Performance penalty of FPAAs can be addressed by using ASIC implementation technology. Dedicated nature of the ASIC approach can achieve much higher performance than what could the flexibility of FPAAs offer. Additionally, in modern cryptographic applications, integration of the TRNG with the cryptographic core is more preferred to minimize the side channel attack risk. In discrete implementations, the connection between TRNG, and cryptographic chip may possess a security flaw that can be used by attackers.

In the single entropy core TRNG model shown in Figure 3.1, chaos control parameter, and bipartition threshold determine the statistical properties of the generated bitstream. Parameter variations have drastic effects on the performance of single entropy core TRNG architecture. Any variation in the threshold creates a statistical bias in the output bitstream, which has to be addressed by a post processor as we

have experienced in the FPAA implementation of the logistic map based TRNG in Chapter 5. Furthermore, the effect of chaos control parameter variation manifests itself as entropy reduction which has been discussed in Chapter 3, and Chapter 4.

We introduced the dual entropy core TRNG model shown in Figure 3.6 as a solution to the chronic problems of its single entropy core counterpart. Dual entropy core architecture omits the threshold generator, and all the complexity associated with its design. Instead, it incorporates another core which can be built using two similar, or dissimilar chaotic maps. We showed that random variable at the output of the proposed architecture will have a symmetric PDF with zero mean that allows the generation of equiprobable bits for two similar entropy cores having uniform underyling distribution. It is easy to observe that regardless of the chaotic dynamics, the concept is valid for all kinds of chaotic maps with uniform underlying distribution. The dual entropy core TRNG architecture performs better in terms of randomness as shown in Chapter 3. The T-entropy calculations in Figure 4.5, and Figure 4.6 also showed that proposed architecture is more immune to parameter variations which make it a convenient candidate for integrated circuit implementation.

In this chapter, we introduce several circuit topologies to implement basic building blocks of a dual entropy core DT chaos based TRNG in ASIC form. Starting from minimalist, and easy to understand circuits to more complex topologies, various circuits will be introduced, and their design aspects will be discussed. Current mode approach has been used in the design of circuits since it allows hardware efficient, and compact implementation of chaotic map equations.

6.1. Design of the Entropy Cores

A typical dual entropy core DT chaos based TRNG consists of two non-linear function blocks that implement chaotic maps of interest, two sample and hold blocks that drive the chaotic dynamics, and a comparator to generate random bits by comparing the outputs of each entropy core as presented in Figure 6.1. An entropy core is defined as the union of a non-linear function block, and a sample and hold block as shown in Figure 6.1.

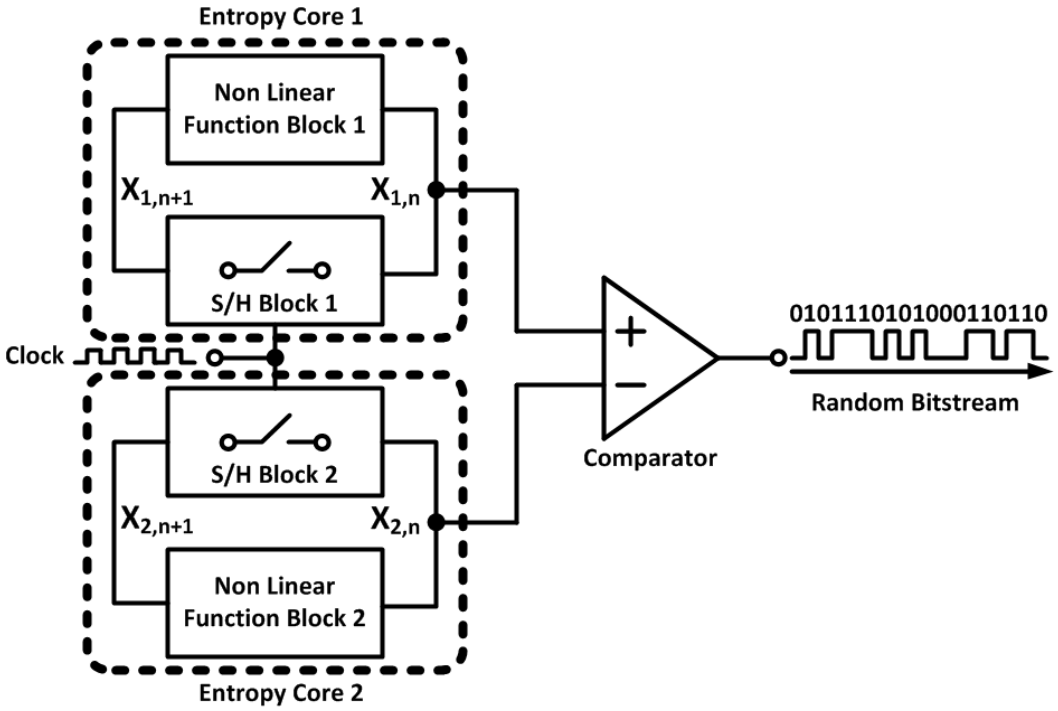


Figure 6.1. Dual entropy core DT chaos based TRNG architecture.

Each entropy core is driven by an external clock. Different clock signals could also be used, but for the sake of simplicity, a single clock signal is used to drive the chaotic dynamics of both entropy cores. Clock signal determines the evolution speed of the chaotic dynamics, and therefore, the throughput, and dynamic power consumption. One of the major advantages of discrete time operation is that the throughput, and dynamic power consumption can be adjusted to meet the requirements of the cryptographic application. Whether it is high speed, or low power, clock signal can be adjusted within available limits to conform to each scenario without requiring any topological changes. The scalable performance offered by the dual entropy core TRNG

can fit into a wide range of applications. Continuous time bulky chaotic circuits have little, or no degree of freedom to offer such functionality.

Comparator block in Figure 6.1 implements the bit extractor function defined by the model presented in Chapter 3. A logic one is generated when the amplitude of the chaotic signal generated by entropy core 1 is greater than the amplitude of the chaotic signal generated by entropy core 2. Comparator generates a logic zero for the opposite case. The throughput of the TRNG is determined by the evolution speed of the chaotic dynamics, which is driven by the external clock signal. The comparator has to operate fast enough to handle the chaotic signals. A post processor may be used to address any potential statistical bias at the cost of reduced throughput.

It is important to note that, in the dual entropy core TRNG model introduced in Chapter 3, entropy cores are accepted to be isolated, uncoupled, and guaranteed to start from different initial conditions. Isolation can be established by following appropriate layout rules. The use of guard rings help to reduce unwanted potential coupling that can cause synchronization of the entropy cores. Because of the existing additive thermal noise, the initial conditions will never be identical in practice. We used two different maps for each entropy core in integrated circuit implementation. Bernoulli map, and tent map both have uniform underlying distribution which meets the basic theoretical requirement for the dual entropy core TRNG architecture. In addition, both maps have continuous bifurcation, and they are capable of generating robust chaos for a wide range of control parameter values as observed in Figure 2.7, and Figure 2.12. This is a desirable feature for a chaotic system, since parameter variations in practice can put the system out of chaos if the control parameter falls into an existing stability island within its bifurcation diagram. The spectral characteristics of both maps have white noise like characteristic. This is also a desirable feature for a chaotic system that will be used as an entropy source in TRNG applications.

DT chaotic maps can be implemented using switched capacitor [10–13, 31, 70, 71, 77–81], or switched current design methods [29, 67, 82–91]. The operation of switched capacitor circuits is based on charge transfer, and requires linear floating

capacitors that only come with special process options (e.g. double polysilicon layers) at increased cost. On the other hand, the switched current technique relies on the ability of MOSFET to maintain its drain current by the charge stored on its gate capacitor when its gate is open circuited. Thus, switched current circuits require only grounded capacitors that do not need to be linear, and they are readily available at the gate of any MOSFET. This renders switched current circuits more compatible with standard digital VLSI processes where cryptographic cores are implemented.

A key performance feature of the current mode approach is its inherent wide bandwidth capability since as a current amplifier, the transistor is useful almost up to its full bandwidth f_T. The simplicity of the switched current circuits offers high operating frequencies with smaller silicon footprint when compared to switched capacitor circuits. Process technology driven shrinking device dimensions lead to integrated circuits with predominately capacitive parasitics. Current mode circuits can achieve high speed operation at low impedance with low voltage swing as a result of minimal capacitive charging, and discharging.

As the device dimensions shrink down, supply voltages have to be reduced for reliable operation. Technology driven diminishing power supplies create a burden on the useful dynamic range. This problem can be addressed by shifting signal representation paradigm from voltage to current, in which the signal dynamic range is no more limited by supply voltage, but it depends on impedance levels. Since switched current circuits operate in the current domain, the supply voltage can be reduced to save power. Thus, switched current circuits can operate consuming less power not only due to their simpler architecture, but also due to the low voltage operation ability.

In switched current design, current transfer functions are linear, and basic analog signal processing functions like inversion, scaling, and summation can be conveniently implemented using the current mirror as a building block. While linear capacitors, and transconductors are not required, precision depends on transistor matching. Good transistor matching is more difficult to attain than good capacitance matching. Thus, careful design of the circuit layout is required for matching.

6.1.1. Minimalist Circuit Implementation of the Bernoulli Map Function

Bernoulli map defined by Equation 2.22 can act as an entropy source with a uniform underlying distribution. Piecewise linear chaotic maps can be implemented using current mirrors. There are many different current mirror topologies in the literature, but for the sake of simplicity, and reduced hardware complexity, basic current mirrors are used to obtain a minimalist, area, and power efficient circuit implementation of the Bernoulli map as shown in Figure 6.2. The minimalist Bernoulli map circuit is composed of 12 MOSFETs. Chaos control parameter β of Equation 2.22 is implemented at the sample and hold stage using the ratio of device dimensions.

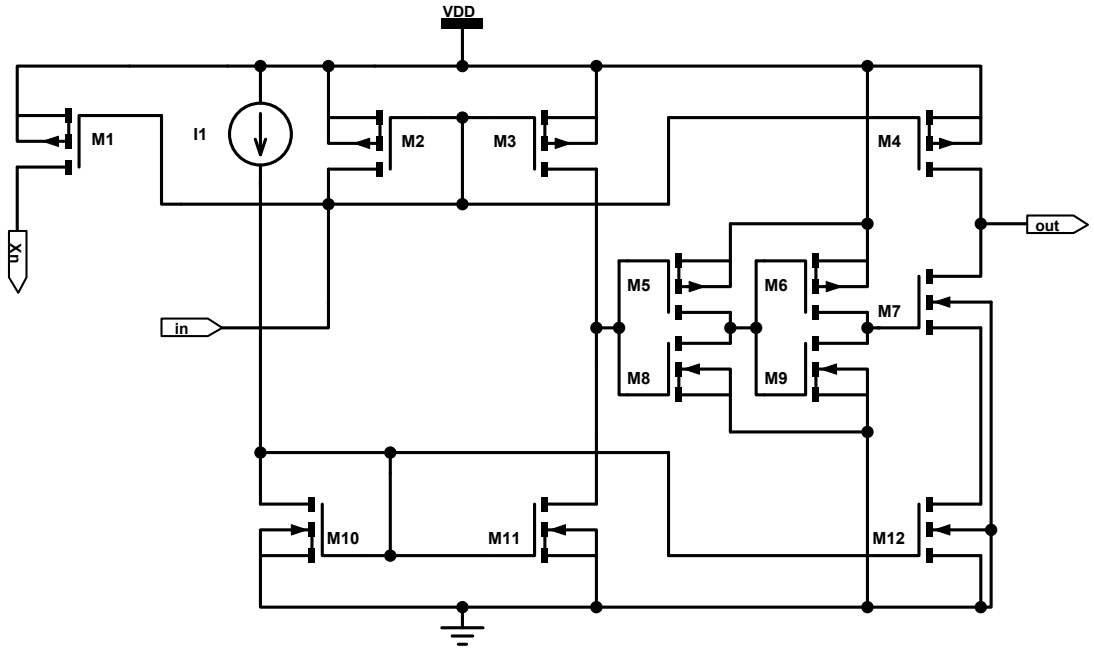


Figure 6.2. Schematic of the minimalist Bernoulli map non linear function block.

In the circuit schematic shown in Figure 6.2, the current source I_1 connected to the drain of MOSFET M_{10}, sets the half-scale current that corresponds to switching point at 0.5 in Equation 2.22, and it is mirrored by M_{11}, and M_{12}. The input current is copied by the current mirror formed by MOSFETs M_1-M_4. M_1 is used to create an auxiliary copy of the state variable x_n for further signal processing. The drains of MOSFETs M_3, and M_{11} are connected together such that the voltage at this node will change proportional to the difference of drain currents. The voltage will not rise until

I_{in} exceeds I_1 which can be used to trigger MOSFET M_7 that handles the switching action of the Bernoulli map by connecting, or disconnecting the drains of MOSFETs M_4, and M_{12}. When $I_{in} > I_1$, M_7 turns on, and the difference between drain currents of M_4, and M_{12} sets the output current. Two stage cascaded inverters inserted between node of interest, and the gate of M_7 improve switching characteristic of the map.

Inevitable process, and device parameter variations exist in practice due to fabrication imperfections, which can be classified as systematic, or random. Systematic variations can be addressed by using proper biasing, and layout techniques. On the other hand, device to device variations, also known as device mismatch, can not be predicted at design stage. A designer can only use device dimensions, layout, and bias point to control matching [54]. Threshold voltage differences ΔV_T, and current gain differences $\Delta K = \mu C_{ox} \Delta W / \Delta L$ are two dominant sources of mismatch which are considered as independent random variables in practice. These random variables have Gaussian distribution with zero mean, and both have device area dependent variances defined by

$$\sigma(\Delta V_T) = \frac{A_{VT}}{\sqrt{WL}}, \tag{6.1}$$

$$\sigma(\frac{\Delta K}{K}) = \frac{A_K}{\sqrt{WL}}, \tag{6.2}$$

where W is the width, and L is the length of the device. Proportionality constants A_{VT}, and A_K are technology dependent parameters. In the simple current mirror, current matching is bias point dependent, and a direct relation between the accuracy, and device area exists

$$ACC \approx \frac{V_{OV}\sqrt{2WL}}{24A_{VT}}. \tag{6.3}$$

Thus, the accuracy requirements of the simple current mirror, impose a minimum device area as suggested by Equation 6.3. For 1% accuracy, which is a reasonable

value in practice, a minimum device area can be calculated by using an appropriate overdrive voltage $V_{OV} = V_{GS} - V_T$, and technology provided A_{VT} constant. According to calculations, a minimum device area of $3.2\mu m^2$ is necessary to meet the required accuracy. While the accuracy improves with increased device area, bandwidth is reduced as a result of increased parasitic capacitance. Simple current mirror operating in saturation region has a bandwidth of

$$BW \approx \frac{g_m}{4\pi C_{GS}}. \tag{6.4}$$

Since $C_{GS} \approx \frac{2}{3} C_{ox} WL$, and $g_m = 2I_D/V_{OV}$, for a MOSFET operating in saturation, we can calculate the required bias current for a target bandwidth of 100MHz using

$$I_D \approx \frac{4}{3}\pi C_{ox} WLV_{OV} BW. \tag{6.5}$$

as $15\mu A$. This current is set by I_1 in the circuit schematic shown in Figure 6.2. We can calculate W/L ratio using current voltage relation of the MOSFET

$$I_D = \frac{1}{2} K \frac{W}{L} V_{OV}^2 \tag{6.6}$$

as $W/L = 2I_D/KV_{OV}^2 = 5$. Using $W \times L = 3.2\mu m^2$, and $W/L = 5$ we get $W = 4\mu m, L = 0.8\mu m$. The minimalist Bernoulli map based on simple current mirrors, is designed to operate for a half-scale current of $15\mu A$ with the device dimensions provided in Table 6.1.

Table 6.1. Device dimensions of the minimalist Bernoulli map circuit.

MOSFET	$\mathbf{M_1}$	$\mathbf{M_2}$	$\mathbf{M_3}$	$\mathbf{M_4}$	$\mathbf{M_5}$	$\mathbf{M_6}$	$\mathbf{M_7}$	$\mathbf{M_8}$	$\mathbf{M_9}$	$\mathbf{M_{10}}$	$\mathbf{M_{11}}$	$\mathbf{M_{12}}$
W (μm)	4	4	4	4	2	2	4	1	1	4	4	4
L (μm)	0.8	0.8	0.8	0.8	0.18	0.18	0.8	0.18	0.18	0.8	0.8	0.8

The layout of the minimalist Bernoulli map circuit presented in Figure 6.3 is composed of 12 MOSFETs, and occupies $15 \times 15\mu m^2$ area on silicon.

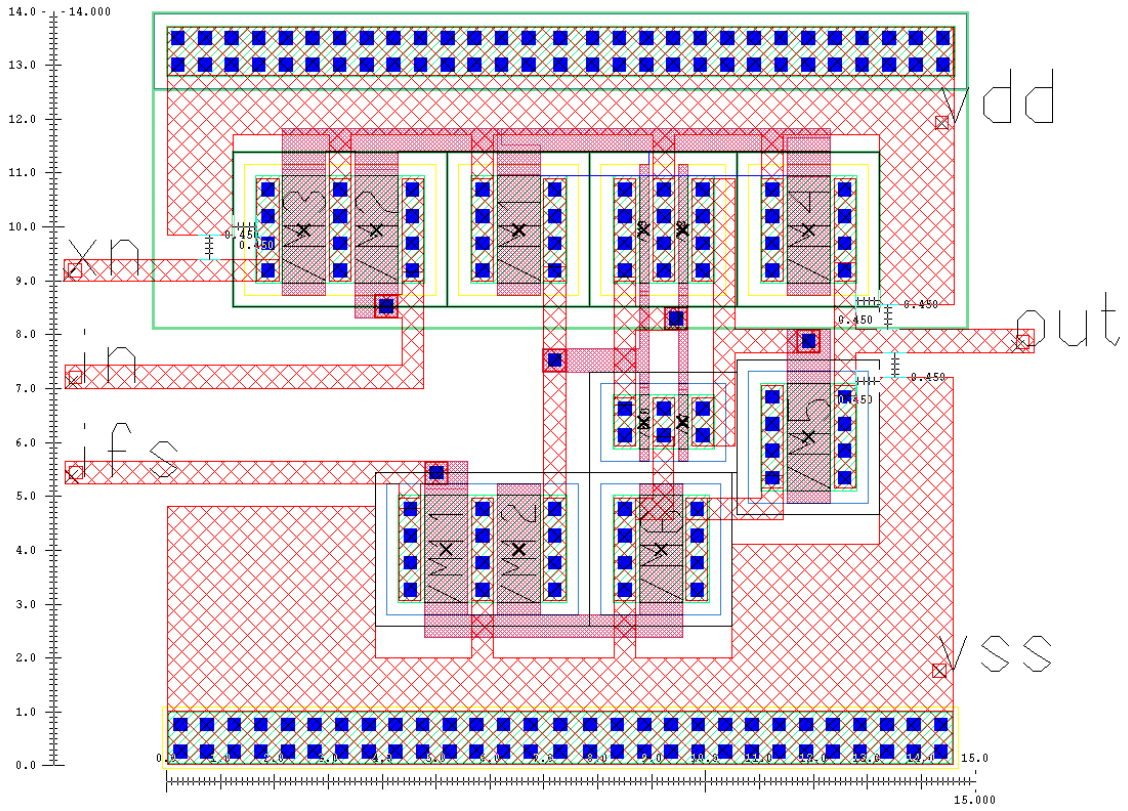


Figure 6.3. Layout of the the minimalist Bernoulli map occupies $15 \times 15\mu m^2$.

In early phase of the circuit design, the voltage at the node where drains of M_3, and M_{11} are connected, was not capable of triggering M_7 due to the low output impedance of simple current mirrors. As a result, current transfer function of the circuit had smooth transitions at the discontinuity point as shown in Figure 6.4. The smooth switching characteristic would manifest itself as entropy reduction, since the chaotic trajectory will not be able to visit certain regions in the phase space [40]. In order to improve the switching characteristic of the Bernoulli map circuit, two stage cascaded inverters were inserted between the node of interest, and the gate of M_7 as shown in Figure 6.2. It has been confirmed with DC transfer function simulations presented in Figure 6.5 that applied solution improves the switching characteristic of the Bernoulli map circuit at the expense of four additional MOSFETs.

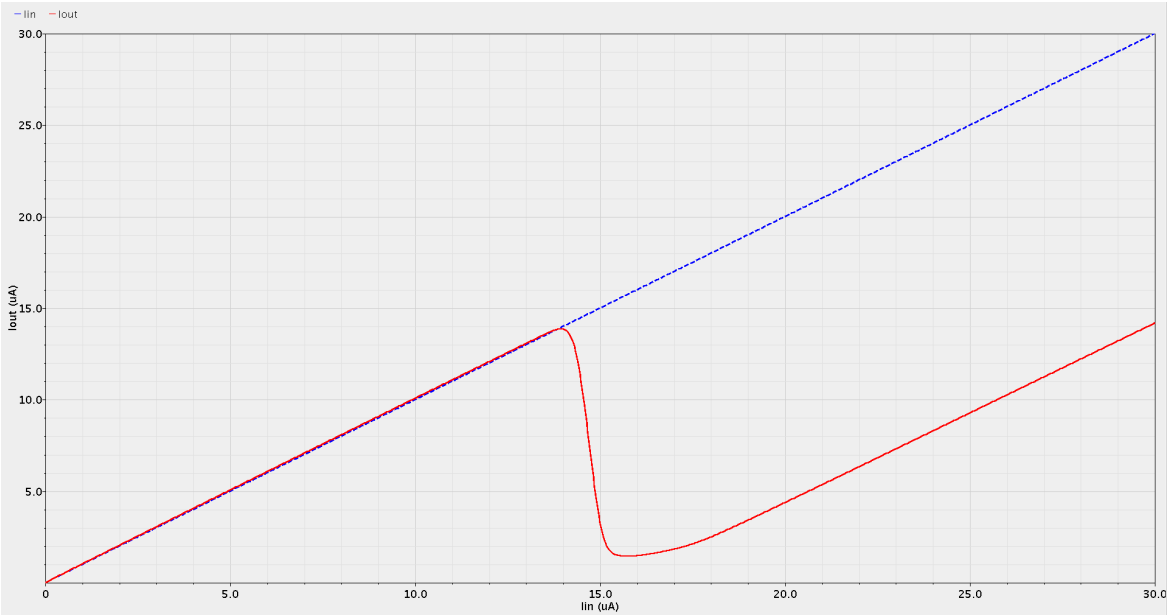


Figure 6.4. DC transfer function of the Bernoulli map circuit without inverters.

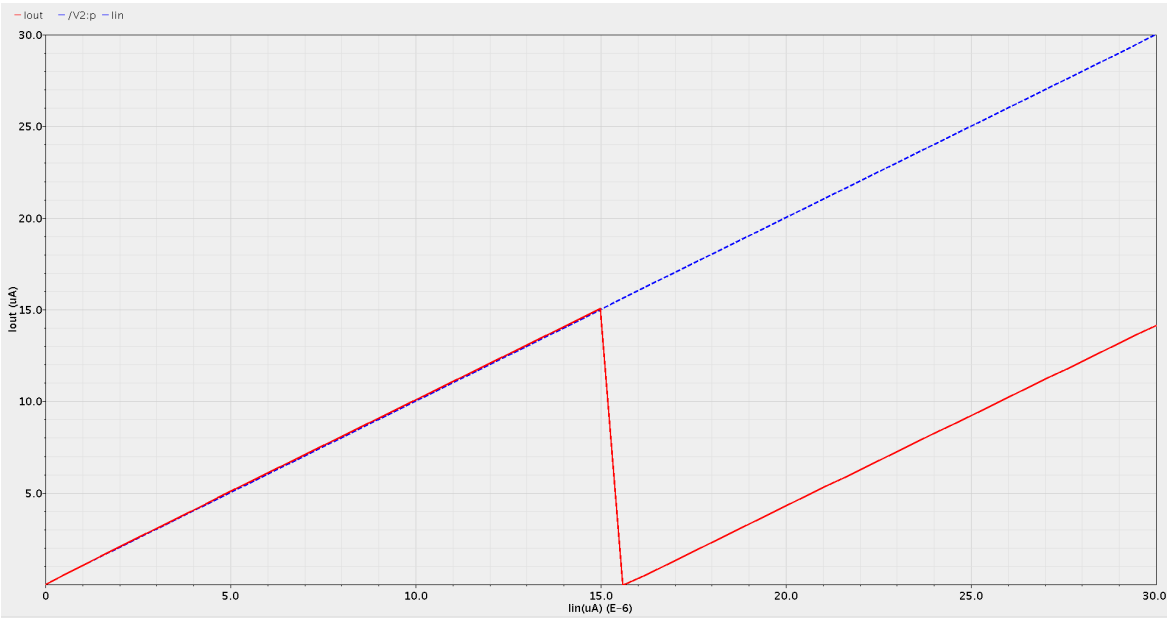


Figure 6.5. Improved DC transfer function of the Bernoulli map circuit.

In the minimalist Bernoulli map circuit, overdrive voltage of the MOSFETs stacked between the power rails can become large enough to saturate the carrier velocity in the channel. The immunity to power rail coupled signals is modest due to relatively low output impedance of the simple current mirrors.

6.1.2. Improved Circuit Implementation of the Bernoulli Map Function

The problems of the minimalist Bernoulli map circuit can be addressed by choosing cascode current mirror topology as the basic building block, without sacrificing the goal of simplicity. The use of cascode current mirror reduces the device overdrive voltages, boosts the output impedance of current mirrors, and improves rejection against supply rail induced signals at the expense of slightly increased hardware complexity. Chaos control parameter β of Equation 2.22 is implemented at the sample and hold stage using the ratio of device dimensions. The improved Bernoulli map circuit composed of 19 MOSFETs is presented in Figure 6.6.

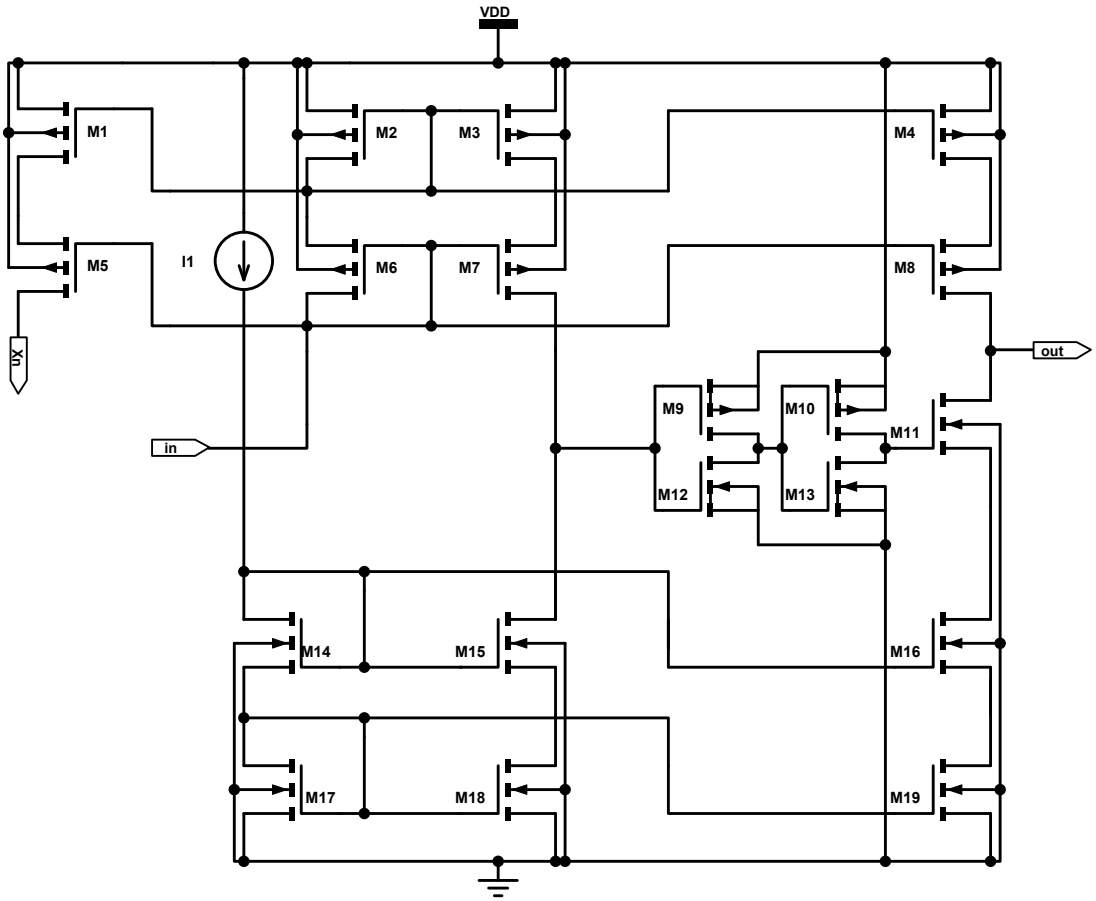


Figure 6.6. Schematic of the improved Bernoulli map circuit.

The operation of the circuit in Figure 6.6 is similar to its minimalist cousin shown in Figure 6.2. The current source I_1 connected to the drain of MOSFET M_{14} determines the half-scale current that stands for the switching point of 0.5 in Equa-

tion 2.22. This current is copied with the help of cascode current mirror formed by MOSFETs M_{14}-M_{19}. The input current is mirrored by the cascode current mirror formed by MOSFETs M_1-M_8. Cascode current mirror MOSFETs M_1, M_5 create an auxiliary copy of the state variable x_n for further signal processing. The drains of MOSFETs M_7, and M_{15} are connected together such that the voltage at this node changes proportionally to the difference of drain currents. The voltage at this circuit node is used to trigger the MOSFET M_{11} through previously introduced two stage cascaded inverters that improve the switching characteristics of the DC transfer function. When the mirrored input current, I_{in}, is greater than the mirrored half-scale current, I_1, the voltage at the node of interest will increase, and M_{11} will be turned on through cascaded inverters. The difference between drain currents of M_8, and M_{16} sets the output current. The improved Bernoulli map circuit has been designed using the previously introduced device matching driven design methodology. It is interesting to note that for the same half-scale current $15\mu A$, and target bandwidth, an improved accuracy level of 0.5% could be achieved using similar device dimensions. Improved Bernoulli map is composed of 19 MOSFETs, and device dimensions are are provided in Table 6.2. Layout of the improved Bernoulli map circuit presented in Figure 6.7 is composed of 19 MOSFETs, and occupies an area of $20 \times 30\mu m^2$ on silicon.

Table 6.2. Device dimensions of the improved Bernoulli map circuit.

MOSFET	$\mathbf{M_1}$	$\mathbf{M_2}$	$\mathbf{M_3}$	$\mathbf{M_4}$	$\mathbf{M_5}$	$\mathbf{M_6}$	$\mathbf{M_7}$	$\mathbf{M_8}$	$\mathbf{M_9}$	$\mathbf{M_{10}}$
W (μm)	4	4	4	4	4	4	4	4	2	2
L (μm)	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.18	0.18
MOSFET	$\mathbf{M_{11}}$	$\mathbf{M_{12}}$	$\mathbf{M_{13}}$	$\mathbf{M_{14}}$	$\mathbf{M_{15}}$	$\mathbf{M_{16}}$	$\mathbf{M_{17}}$	$\mathbf{M_{18}}$	$\mathbf{M_{19}}$	
W (μm)	2	1	1	4	4	4	4	4	4	
L (μm)	0.8	0.18	0.18	0.8	0.8	0.8	0.8	0.8	0.8	

DC transfer function of the improved Bernoulli map circuit is shown in Figure 6.8. Switching characteristics of the Bernoulli map circuit have become remarkably sharper

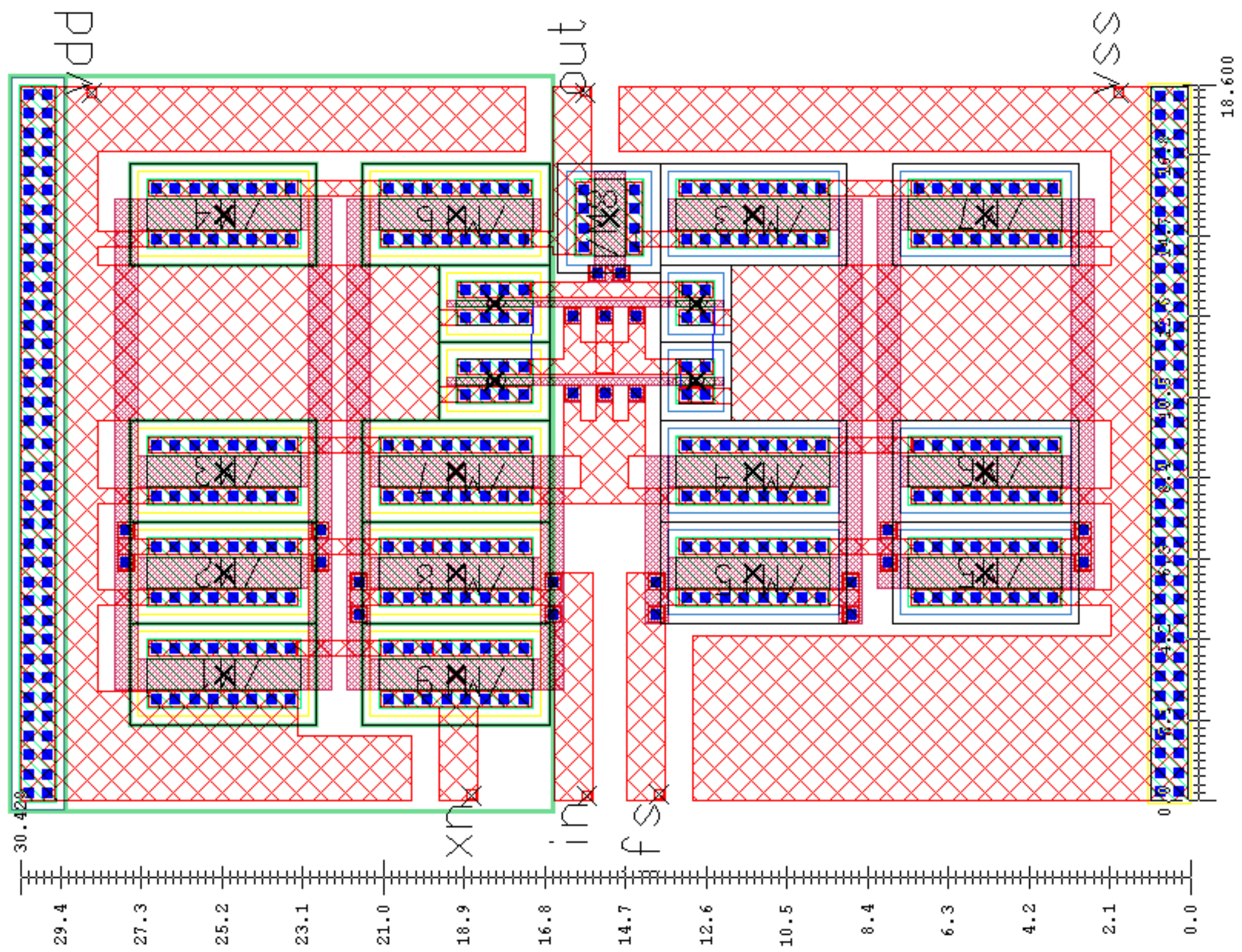


Figure 6.7. Layout of the the improved Bernoulli map occupies $20 \times 30\mu m^2$.

than its minimalist predecessor presented in Figure 6.5 with the help of cascode current mirrors used to implement the Bernoulli map function.

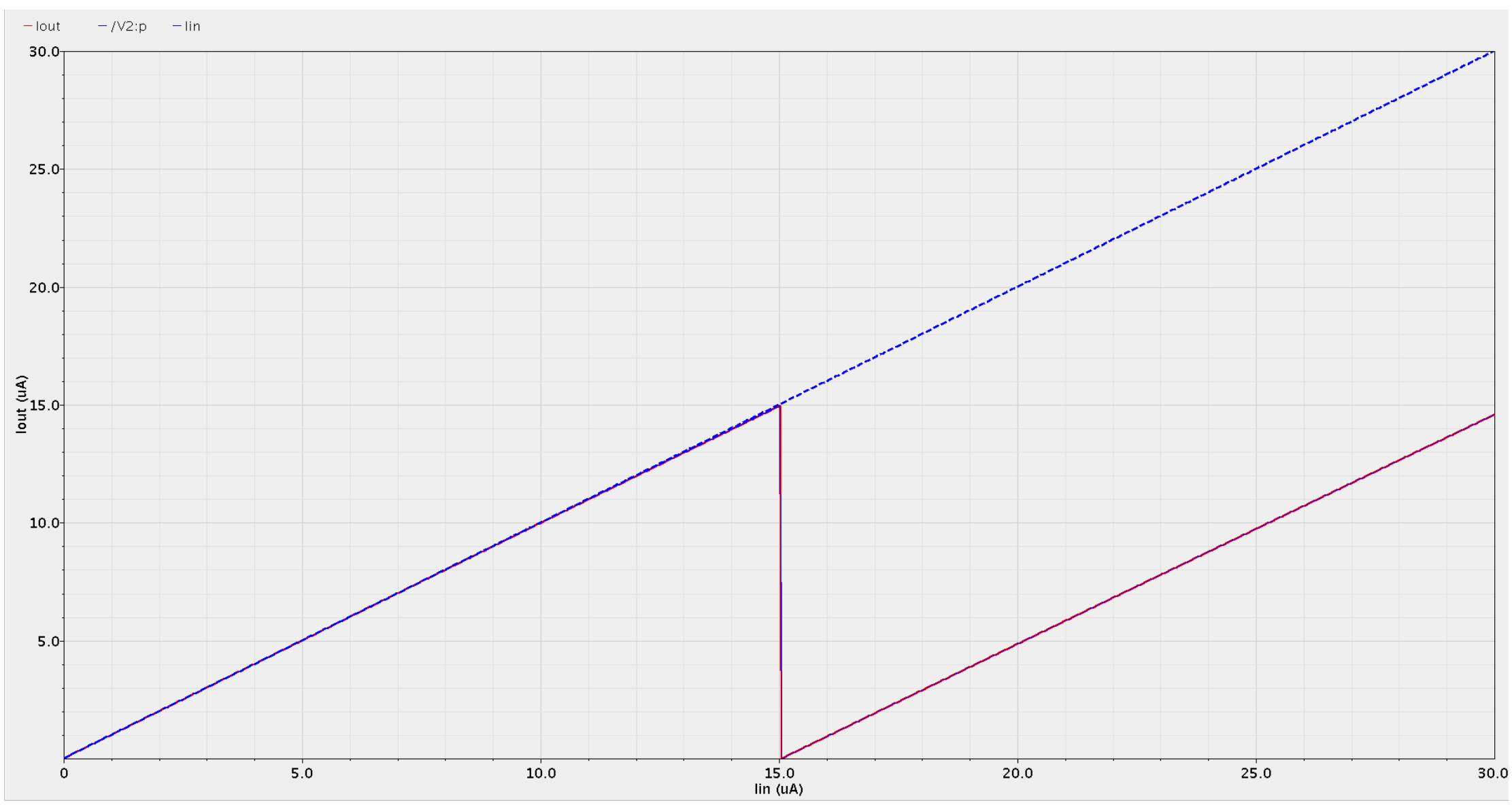


Figure 6.8. DC transfer function of the improved Bernoulli map circuit.

6.1.3. Minimalist Circuit Implementation of the Tent Map Function

Tent map defined by Equation 2.15 can operate as an entropy source with a uniform underlying distribution. Piecewise linear characteristic of the tent map can be implemented using simple current mirrors to create a minimalist, area, and power efficient circuit implementation as shown in Figure 6.9.

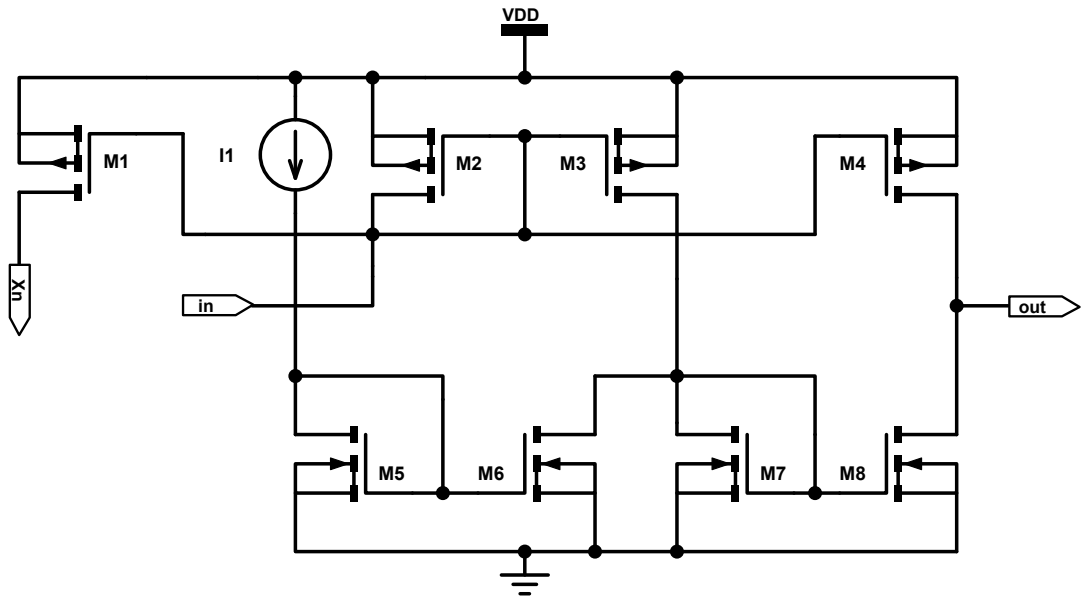


Figure 6.9. Schematic of the minimalist tent map non linear function block.

In the minimalist tent map circuit composed of 8 MOSFETs shown in Figure 6.9, current source I_1 sets the half-scale current, and it is mirrored by M_6. Current mirror formed by M_1, M_2, M_3, and M_4 copies the input current. M_1 is used to create an auxiliary copy of the state variable x_n for further signal processing. The difference between drain currents of M_3, and M_6 determines the drain current of M_7, which is further mirrored by M_8, and the difference between the drain currents of M_4, and M_8 forms the output current. In a simple current mirror, current matching is bias point dependent [54]. The accuracy requirements of the simple current mirror, impose a minimum device area according to Equation 6.3. A minimum device area of $3.2\mu m^2$ is needed for a practically useful value of 1% accuracy. Although the accuracy improves with increasing device area according to Equation 6.3, the bandwidth of the simple current mirror defined by Equation 6.4 is reduced because of increased parasitic ca-

pacitance. The transconductance parameter, g_m, of current mirror transistors can be calculated for a particular bandwidth using Equation 6.4. Assuming saturation, the half-scale bias current for a target bandwidth of 100MHz is calculated using Equation 6.5, and Equation 6.6 as $15\mu A$. Then using the device current-voltage relation Equation 6.6, $W/L = 5$ is obtained. Knowing $W \times L = 3.2\mu m^2$, and $W/L = 5$ we get device dimensions as $W = 4\mu m, L = 0.8\mu m$. The device length is larger than the technology minimum, and it helps to improve the matching, and reduce the channel length modulation effect. The minimalist tent map circuit composed of 8 MOSFETs has been implemented with the device dimensions provided in Table 6.3. The layout of the minimalist tent map circuit presented in Figure 6.10 is composed of eight MOSFETs, and occupies $17 \times 14\mu m^2$ area on silicon.

Table 6.3. Device dimensions of the minimalist tent map circuit.

MOSFET	$\mathbf{M_1}$	$\mathbf{M_2}$	$\mathbf{M_3}$	$\mathbf{M_4}$	$\mathbf{M_5}$	$\mathbf{M_6}$	$\mathbf{M_7}$	$\mathbf{M_8}$
W (μm)	4	4	8	4	4	4	4	4
L (μm)	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8

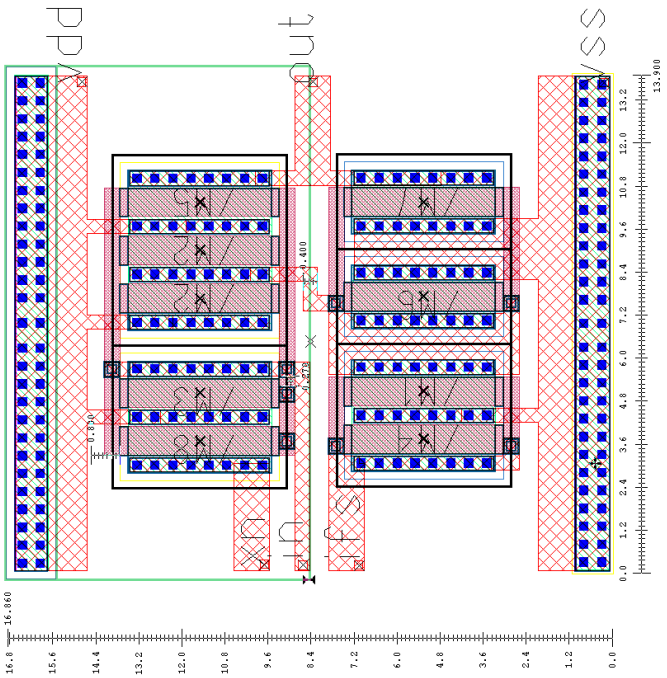


Figure 6.10. Layout of the the minimalist tent map occupies $17 \times 14\mu m^2$.

The overdrive voltage of the MOSFETs stacked between the power rails is in the range of $0.9Vs$ for $V_{DD} = 1.8V$ which is large enough to saturate the carrier velocity in device channel. Under velocity saturation, device will operate with reduced g_m, and degrade device performance. While the ideal tent map, shown in Figure 2.6, has a sharp slope transition, the circuit implementation has smooth transition at the discontinuity point as observed in DC transfer characteristic presented by Figure 6.11.

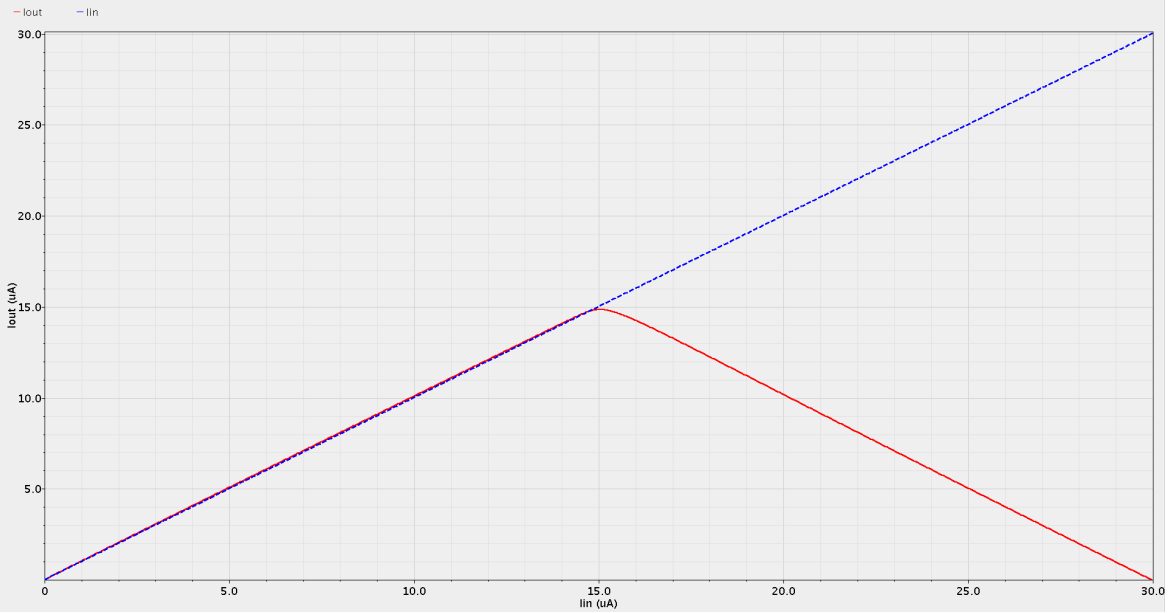
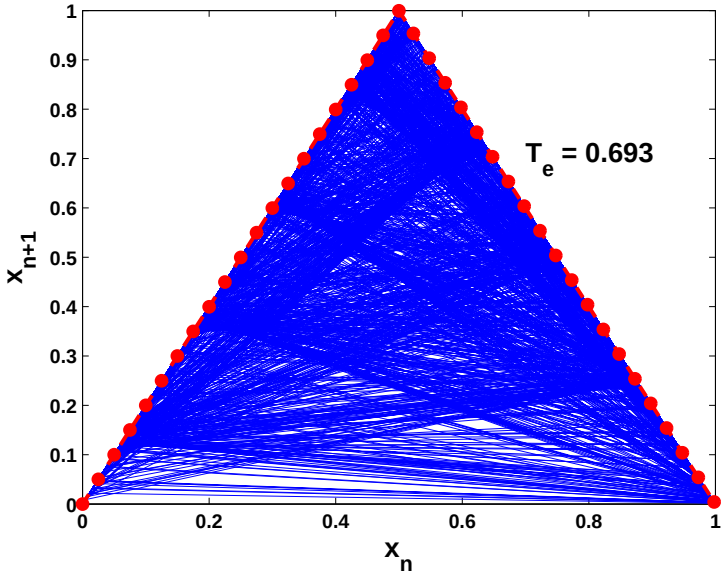


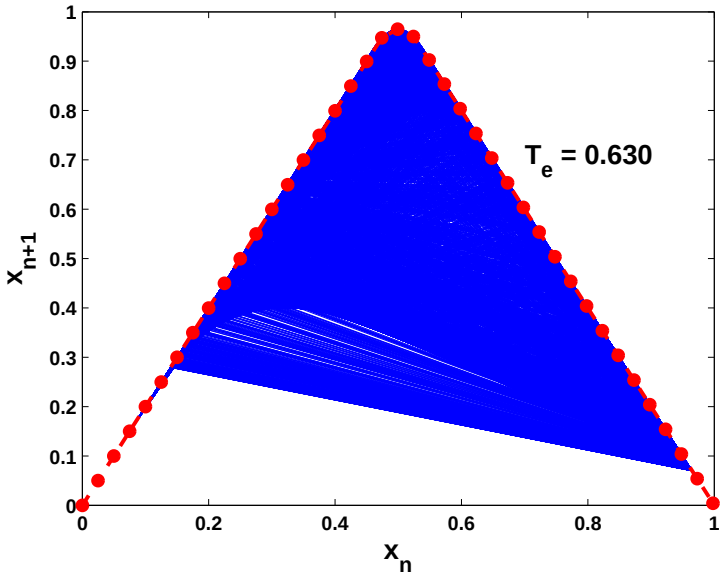
Figure 6.11. DC transfer function of the minimalist tent map circuit.

The slope of the tent map affects the chaos control parameter, and this can reduce the entropy [40, 42]. A custom tent map model with smooth slope transition has been developed to understand the effect of smooth slope transition on entropy. The ideal tent map has been modified by interposing a parabola at the discontinuity point. Ideal, and modified tent maps have been numerically simulated using the single entropy core TRNG model, and the T-entropy of the generated bitstream for each map has been calculated. Phase portrait plots of ideal, and modified tent map are presented in Figure 6.12. A comparative glance at the Figure 6.12 reveals that the chaotic trajectories are unable to visit certain regions in the phase space of the modified tent map. This vacancy has been translated into an entropy loss of 0.063, as indicated by the calculated T-entropy values presented in Figure 6.12. The effect of entropy reduction is a result of smooth slope transition at the discontinuity point [40, 92]. Thus, smooth slope transition induced entropy reduction effect should be taken into

account for creating efficient entropy sources. Design blocks that improve switching characteristics should be employed to increase sharpness at the discontinuity point where slope transition occurs.



(a)



(b)

Figure 6.12. Phase portraits of the ideal tent map (a), and modified tent map (b) with calculated T-entropy values for generated bitstreams from each map.

6.1.4. Improved Circuit Implementation of the Tent Map Function

The improved circuit implementation of the tent map composed of 16 MOSFETs is presented in Figure 6.13. The switching characteristic of the minimalist tent map has been improved using cascode current mirror as a building block without sacrificing the goal of lightweight design. Cascode current mirror is chosen because of its good current matching, and high output impedance. With the help of cascode topology, the overdrive voltage of each device is reduced to avoid velocity saturation, and the output impedance of the current mirrors is boosted to improve current matching, and immunity to power rail coupled signals.

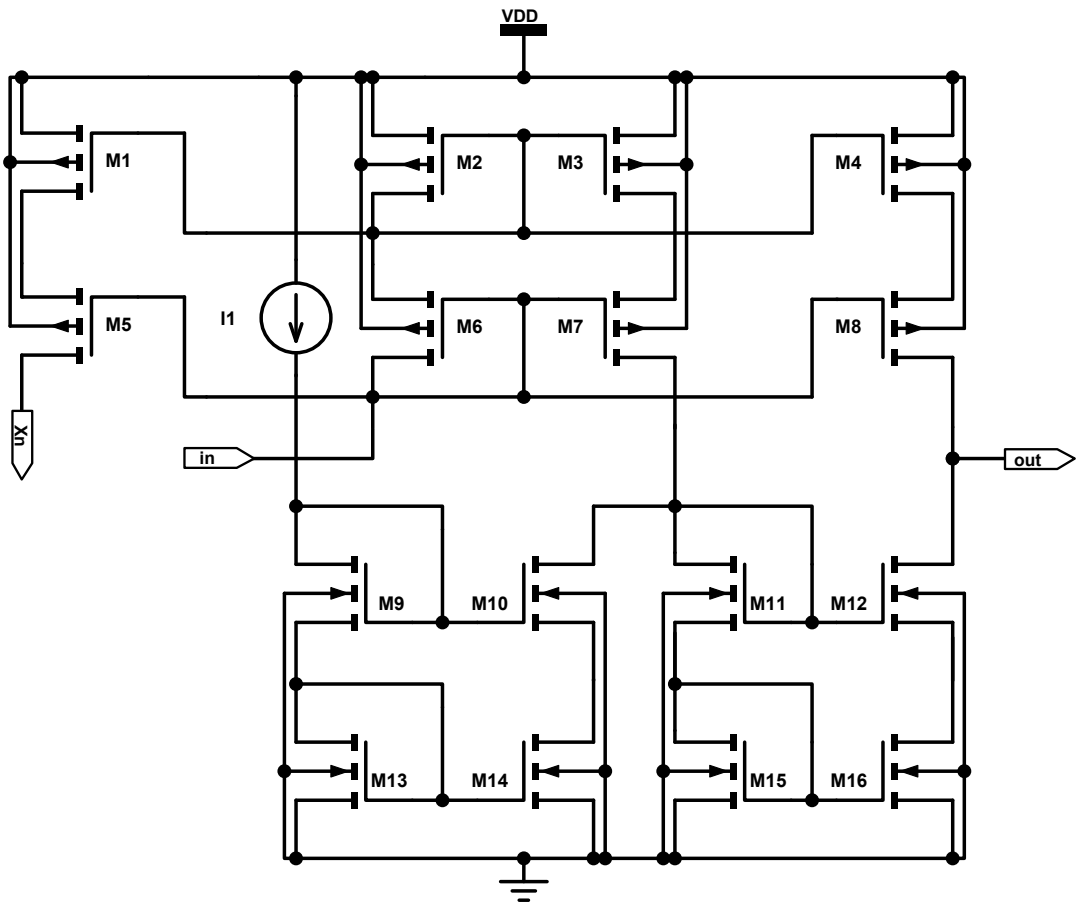


Figure 6.13. Schematic of the improved tent map non linear function block.

The operation of the improved tent map circuit is similar to its minimalist precursor. The circuit requires a half-scale bias current that will correspond to the domain of the map which is provided by the current source I_1. The half-scale bias current is copied by cascode current mirror composed of M_9, M_{10}, M_{13}, and M_{14}. Cascode current mirror formed by M_1-M_8, create copies of the input current on respective circuit branches. M_1, M_5 create an auxiliary copy of the state variable x_n for further signal processing. The difference between drain currents of M_3, M_7, and M_{10}, M_{14} determines the drain current of M_{11}, M_{15} which is mirrored by M_{12}, M_{16}. The difference current between drain currents of M_8, and M_{12} forms the output current.

Using the previously introduced device matching driven design methodology, the improved tent map circuit is designed to operate for a half-scale current of $15\mu A$ targeting an accuracy level of 1%. Device dimensions of the improved tent map circtuit are provided in Table 6.4. The layout of the improved tent map circuit composed of 16 MOSFETs is presented in Figure 6.14, and occupies $29 \times 17\mu m^2$ area on silicon. DC transfer function of the improved tent map circuit is shown in Figure 6.15. Slope transition characteristics of the tent map circuit have become better than its minimalist counterpart presented in Figure 6.11 with the help of cascode current mirrors.

Table 6.4. Device dimensions of the improved tent map circuit.

MOSFET	$\mathbf{M_1}$	$\mathbf{M_2}$	$\mathbf{M_3}$	$\mathbf{M_4}$	$\mathbf{M_5}$	$\mathbf{M_6}$	$\mathbf{M_7}$	$\mathbf{M_8}$
W (μm)	4	4	8	4	4	4	8	4
L (μm)	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8
MOSFET	$\mathbf{M_9}$	$\mathbf{M_{10}}$	$\mathbf{M_{11}}$	$\mathbf{M_{12}}$	$\mathbf{M_{13}}$	$\mathbf{M_{14}}$	$\mathbf{M_{15}}$	$\mathbf{M_{16}}$
W (μm)	4	4	4	4	4	4	4	4
L (μm)	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8

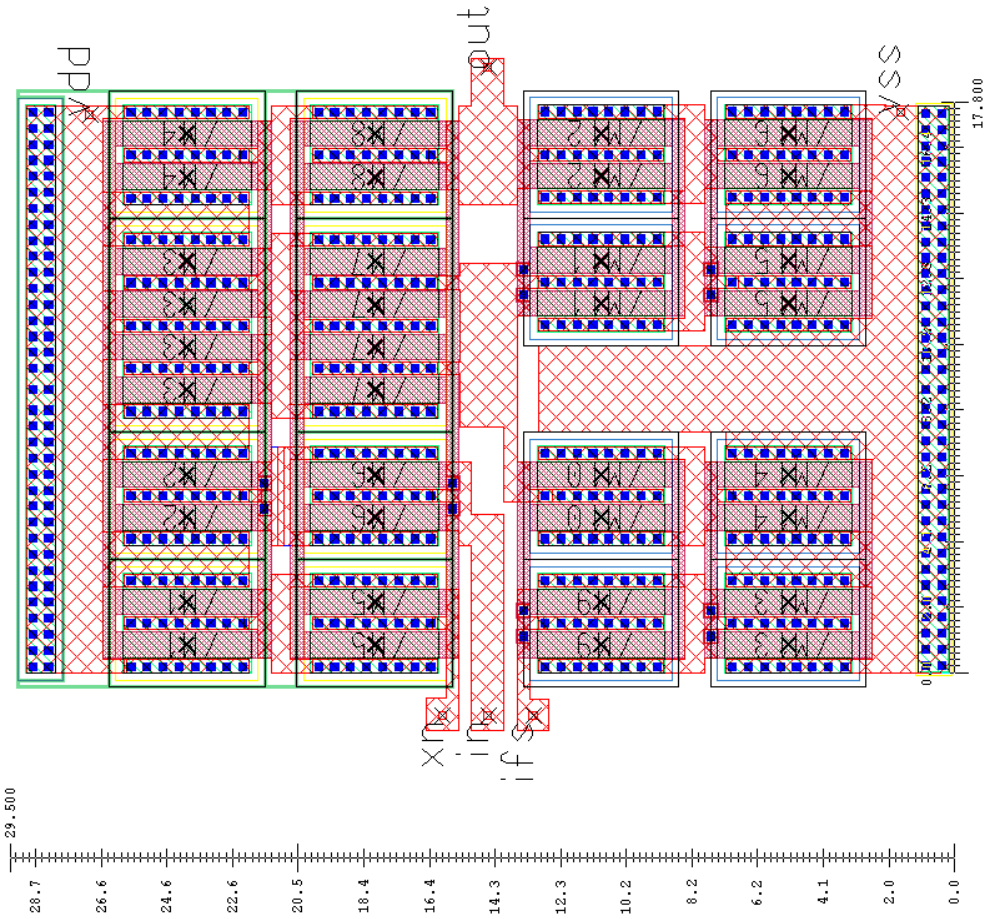


Figure 6.14. Layout of the the improved tent map occupies $29 \times 17 \mu m^2$.

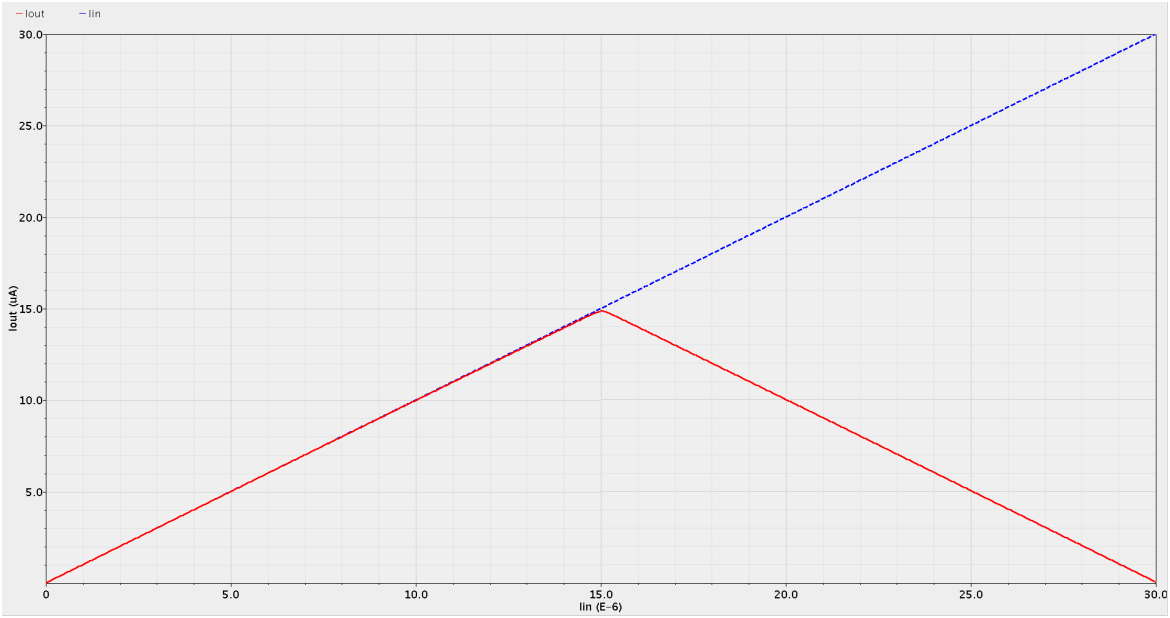


Figure 6.15. DC transfer function of the improved tent map circuit.

6.2. Design of the Sample and Hold Circuit

The sample and hold block in Figure 6.1 drives the dynamics of the chaotic map circuit with the help of an external clock signal. A simple current mirror can be modified to act as a sample and hold stage using a switch between the gates of mirror MOSFETs. A voltage can be stored in a capacitor connected at the gate of the mirror MOSFET during the sampling phase in order to use the associated current later in the hold phase. This retained current can be propagated, and scaled with the help of other current mirror circuits. A two stage current mirror based sample and hold circuit composed of 23 MOSFETs shown in Figure 6.16 has been designed for driving chaotic dynamics of the map circuits.

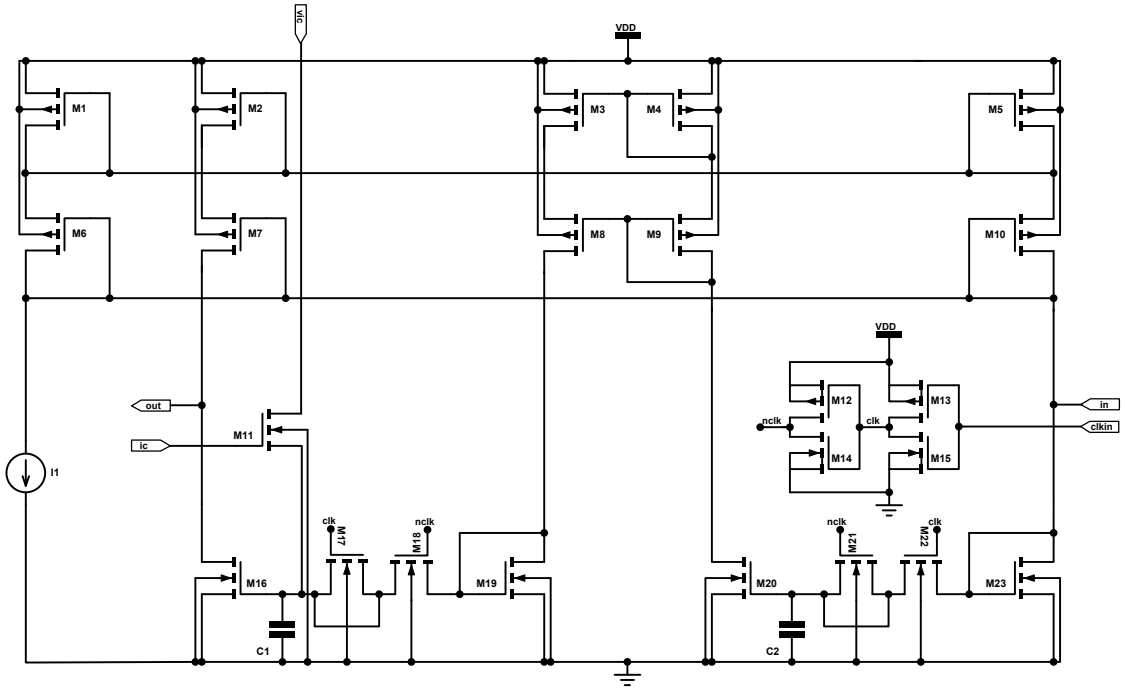


Figure 6.16. Schematic of the improved sample and hold circuit.

Current mirror formed by MOSFETs M_1, M_2, M_5, M_6, M_7, M_{10} provide a bias current I_1, to control the operation of current mirror based sample and hold circuit. The input current is sampled onto C_2 via M_{23}, M_{22}, and M_{21} at the positive cycle of the clock signal. On the negative clock cycle, charge stored in C_2 creates the copy of the input current by biasing M_{20}. This current is copied to second sample and hold stage over the cascode current mirror formed by M_3, M_4, M_8, and M_9. Previously

sampled copy of the input current is sampled into C_1 over M_{19}, M_{18}, and M_{17}. On the next positive cycle of the clock, a ratio of this current will be flowing through M_{16} with the help of the charge stored at C_1, which biases the MOSFET. The chaos control parameter is implemented by the current transfer ratio of the current mirror formed by M_{16}-M_{19} to reduce sensitivity to parameter deviations. M_{11} is used to initialize the dynamics by charging C_1 to an externally applied voltage that sets the initial value of the map. M_{17}, and M_{21} are half sized dummy MOSFETs that are used to reduce clock feed-through effect. Sampling switch transistors M_{18}, and M_{22} are sized at minimum to cope for charge sharing phenomena. Two stage cascaded inverters formed by M_{12}, M_{13}, M_{14}, and M_{15} have been used to drive the switch transistors with the help of the externally applied clock signal.

The design of the circuit follows a matching driven design methodology. As previously mentioned, in practical implementations device mismatches occur due to process variations. Implementation technology imposes a minimum area for a certain variation of device dimensions according to Equation 6.1, and Equation 6.2. The minimum required device area can be calculated by substituting the T-entropy estimated maximum allowable parameter variance in Equation 6.2, and using the technology provided A_K parameter. Then, for a target sampling frequency the required g_m is calculated using Equation 6.4 under the assumption of saturation mode operation where the gate capacitance of a mirror MOSFET is $C_{GS} \approx \frac{2}{3}C_{ox}WL$. Required bias current can be obtained using $g_m = 2I_D/V_{OV}$ by choosing appropriate overdrive voltage. After calculating required drain current I_D, the W/L ratio of the mirror MOSFETs can be calculated using Equation 6.6. Device dimensions are determined for a particular operating condition with the knowledge of W/L, and WL using maximum allowable parameter variation, bandwidth, and overdrive voltage as independent design parameters. Chaos control parameter has been implemented using the ratio of device dimensions to reduce sensitivity to parameter deviations. The proposed novel design approach in this section establishes a link between previously developed information theory based parameter variation estimation, and circuit design processes, thus help the designer in bridging the gap between theory, and application.

During the operation of sample and hold stages, clock signal can be coupled to the signal being sampled by the switch as a result of the coupling between the gate driving clock signal, and the MOSFET channel over the gate-to-source capacitance. This phenomena is known as the clock feedthrough, and it can introduce excessive voltage in the sampling capacitors which can put the system out of chaos if the dynamic range of the map circuit has been exceeded [93, 94]. In order to minimize this harmful effect, small switch MOSFETs have been used to reduce the coupling [95]. Additionally, a good control of the slew rate of the clock signal can help to reduce the clock feedthrough effect. When the switch transistors are turned off, the stored charge in the channel that flows into the sampling capacitor can cause a signal dependent charge injection error [96–98]. This phenomena can put the system out of chaos if excessive amount of charge is injected. In order to cope for this problem half size dummy switches have been placed in series with the switch transistors [95, 99]. UMC technology provides single polysilicon, and six metal layers for circuit implementation. Since the polysilicon capacitors are extinct, technology provided metal insulator metal (MIM) capacitors have been used for sample and hold charge storage. The use of uppermost metal layers for MIM capacitors provides an advantage against substrate coupled noise, since the MIM capacitors are located far above transistors. Device dimensions are provided in Table 6.5. Layout of the sample and hold circuit presented in Figure 6.17 is composed of 23 MOSFETs, and two MIM capacitors. which occupy a silicon area of $23 \times 38 \mu m^2$.

Table 6.5. Device dimensions of the sample and hold circuit.

MOSFET	$\mathbf{M_1}$	$\mathbf{M_2}$	$\mathbf{M_3}$	$\mathbf{M_4}$	$\mathbf{M_5}$	$\mathbf{M_6}$	$\mathbf{M_7}$	$\mathbf{M_8}$	$\mathbf{M_9}$	$\mathbf{M_{10}}$	$\mathbf{M_{11}}$	$\mathbf{M_{12}}$
W (μm)	4	4	4	4	4	4	4	4	4	4	2	2
L (μm)	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.18
MOSFET	$\mathbf{M_{13}}$	$\mathbf{M_{14}}$	$\mathbf{M_{15}}$	$\mathbf{M_{16}}$	$\mathbf{M_{17}}$	$\mathbf{M_{18}}$	$\mathbf{M_{19}}$	$\mathbf{M_{20}}$	$\mathbf{M_{21}}$	$\mathbf{M_{22}}$	$\mathbf{M_{23}}$	
W (μm)	1	1	1	3.95	4	1	2	2	1	2	2	
L (μm)	0.18	0.18	0.18	0.8	0.18	0.18	0.8	0.8	0.18	0.18	0.8	

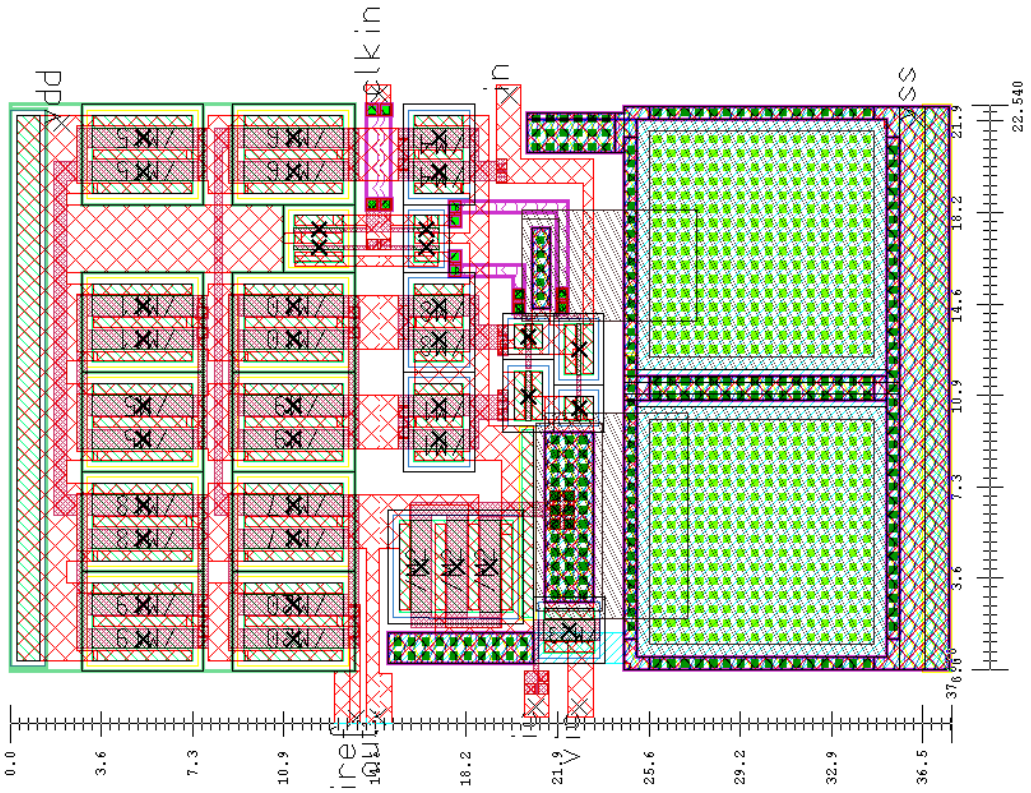


Figure 6.17. Layout of the sample and hold circuit occupies $23 \times 38 \mu m^2$.

Transient simulation results presented in Figure 6.18, and Figure 6.19 show the chaotic signals generated by improved Bernoulli, and tent map circuits that operate with their respective sample and hold circuits.

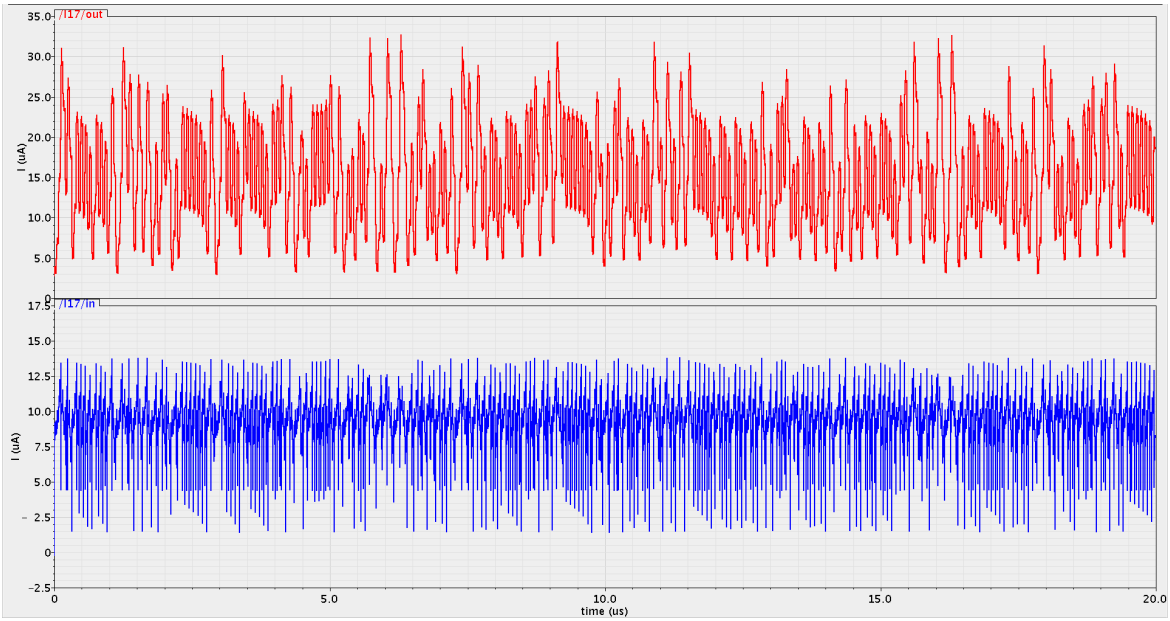


Figure 6.18. Transient operation of the improved Bernoulli map circuit.

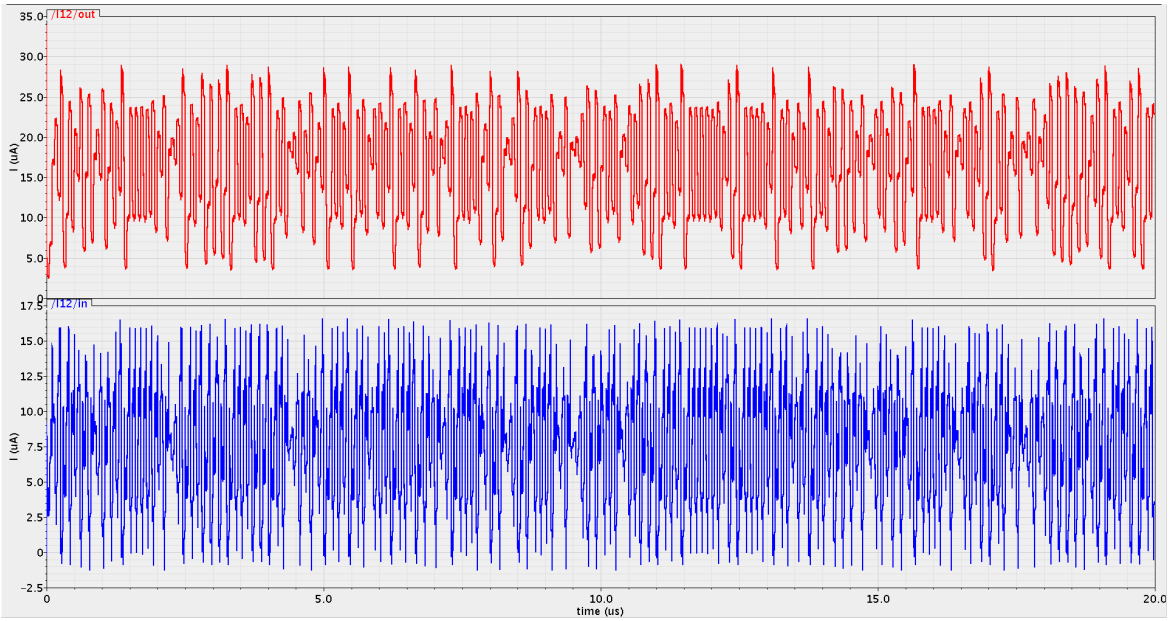


Figure 6.19. Transient operation of the improved tent map circuit.

6.3. Design of the Current Mode Comparator for Random Bit Generation

The current mode comparator has been designed to implement Equation 3.1 that generates random bits by comparing the amplitudes of analog samples from the entropy cores. The first examples of current comparators in the literature use the current mirrors as the basic building block [100]. Unfortunately, high output resistance of the mirrors degrade the bandwidth, especially at low input current levels. Various current mode circuits that use non-linear feedback have been developed to improve the transient performance at the cost of increased complexity, and power consumption [101–104]. Current mode comparators can be distinguished according to the input impedances. In the low impedance types, input current is sensed at a low impedance node, and then amplified to generate an output. On the other hand in high impedance variants, the input current is sensed at a high impedance node, and usually amplified using a non-linear feedback mechanism. The first type provides high speed at the cost of reduced accuracy, while the second type provides high resolution at the expense of increased voltage swings that fundamentally limit the operation speed.

The first type has been preferred for dual entropy core TRNG architecture, in order not to create a performance bottleneck on the throughput. A simple yet

effective current input voltage output comparator circuit presented in Figure 6.20 has been designed. For moderate to large input currents MOSFETs will be operating in saturation mode as intended but, at small input current levels, MOSFETs tend to operate in the subthreshold region, and the mode change of their operation back in saturation region can cause undesirable long delays in the transient response.

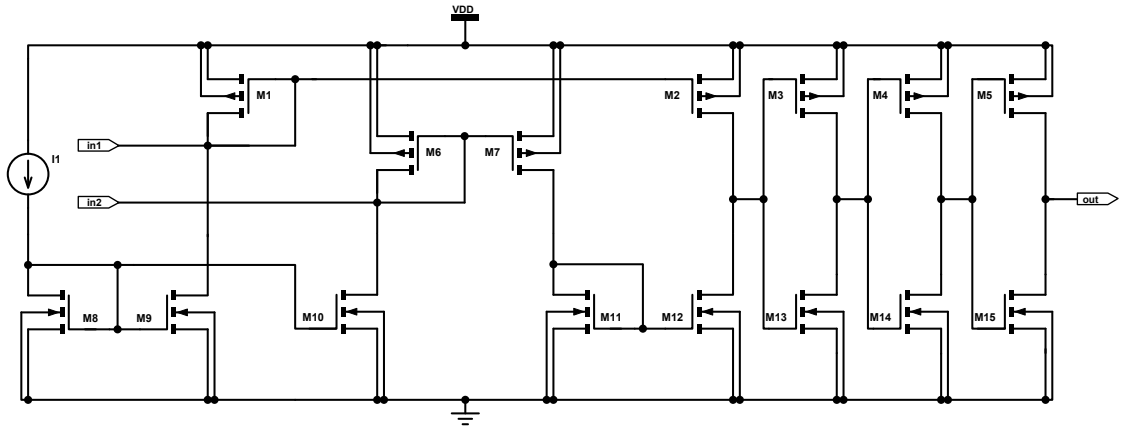


Figure 6.20. Schematic of comparator circuit.

In order to overcome this problem, a current source I_1 is used for generating bias currents to the input current mirrors, such that slow response time due to subthreshold operation at low input current levels could be avoided. The first input current is added to the bias current provided by current mirror formed by M_8, and M_9, then this current is copied by the current mirror formed by M_1, and M_2. Diode connected MOSFETs M_1, and M_6 maintain low impedance at the inputs. The second input current is added with the bias current copied by current mirror composed of M_8, M_9, and M_{10}. Then this current is tranferred with the help of current mirrors formed by M_6, M_7, M_{11}, M_{12}. At the circuit node, where the drains of M_7, and M_{11} meet, a voltage proportional to the difference of input currents is generated. Then this voltage is amplified, and digitized by a chain of inverters composed of M_3, M_{13}, M_4, M_{14}, M_5, and M_{15}. Device dimensions are provided in Table 6.6. The layout of the current mode comparator circuit composed of 15 MOSFETs presented in Figure 6.21 occupies a $19 \times 16\mu m^2$ area on silicon.

Table 6.6. Device dimensions of the comparator circuit.

MOSFET	$\mathbf{M_1}$	$\mathbf{M_2}$	$\mathbf{M_3}$	$\mathbf{M_4}$	$\mathbf{M_5}$	$\mathbf{M_6}$	$\mathbf{M_7}$	$\mathbf{M_8}$
W (μm)	4	4	4	4	4	4	4	4
L (μm)	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8
MOSFET	$\mathbf{M_9}$	$\mathbf{M_{10}}$	$\mathbf{M_{11}}$	$\mathbf{M_{12}}$	$\mathbf{M_{13}}$	$\mathbf{M_{14}}$	$\mathbf{M_{15}}$	
W (μm)	4	4	2	2	2	1	1	
L (μm)	0.8	0.8	0.8	0.18	0.18	0.18	0.18	

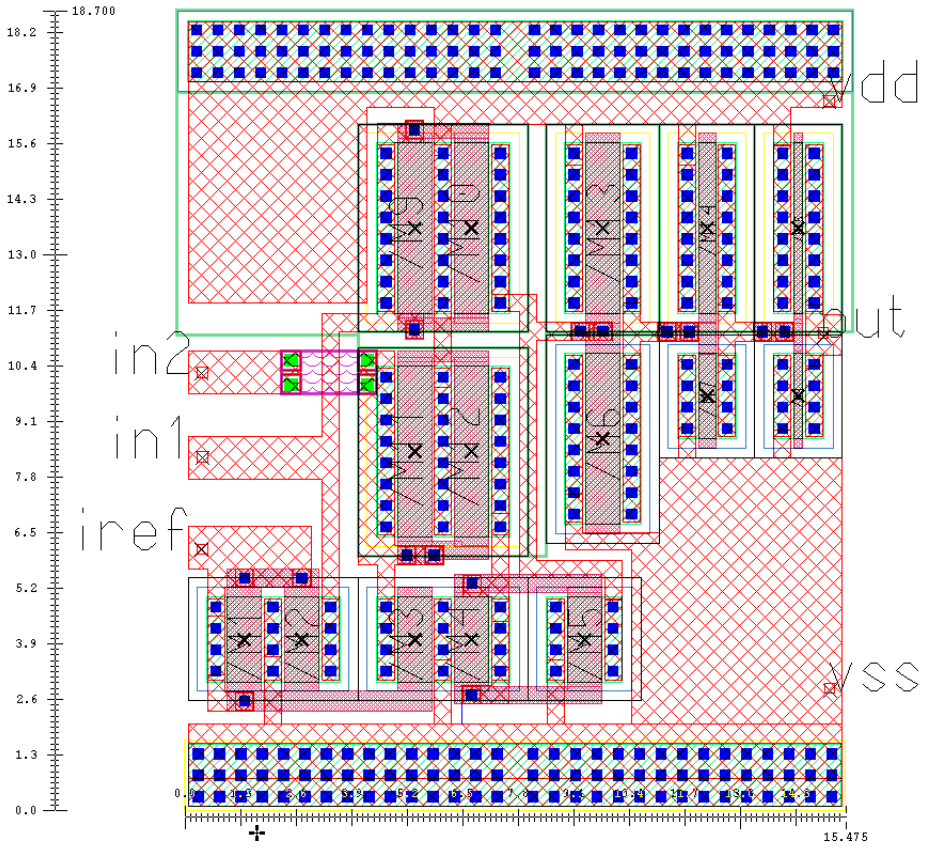


Figure 6.21. Layout of the the current comparator circuit occupies $19 \times 16\mu m^2$.

The DC transfer characteristics of the current mode comparator circuit is presented in Figure 6.22 that have a convenient interval to operate with chaotic map circuits. The output of the comparator switches when the swept input current exceeds the half-scale current at the second input as observed in Figure 6.22.

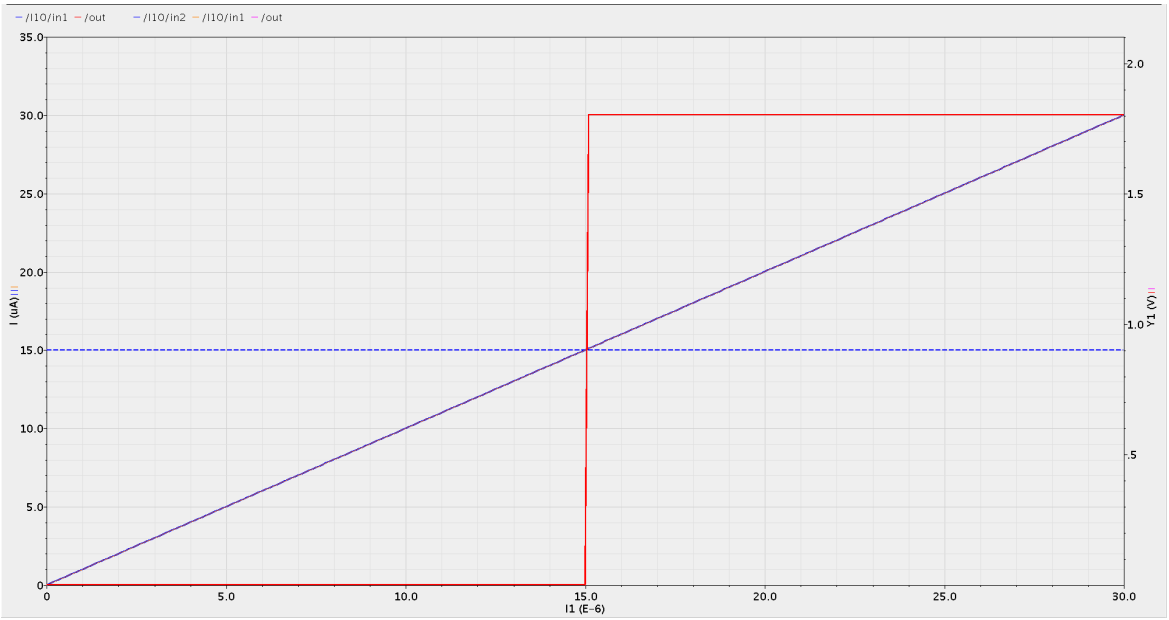


Figure 6.22. DC transfer characteristics of the current mode comparator circuit.

Time domain behavior of the circuit is studied using a time varying input current against a constant input. Transient simulations shown in Figure 6.23 reveals a propagation delay of $3.6ns$ that corresponds to a bandwidth in the excess of $270MHz$.

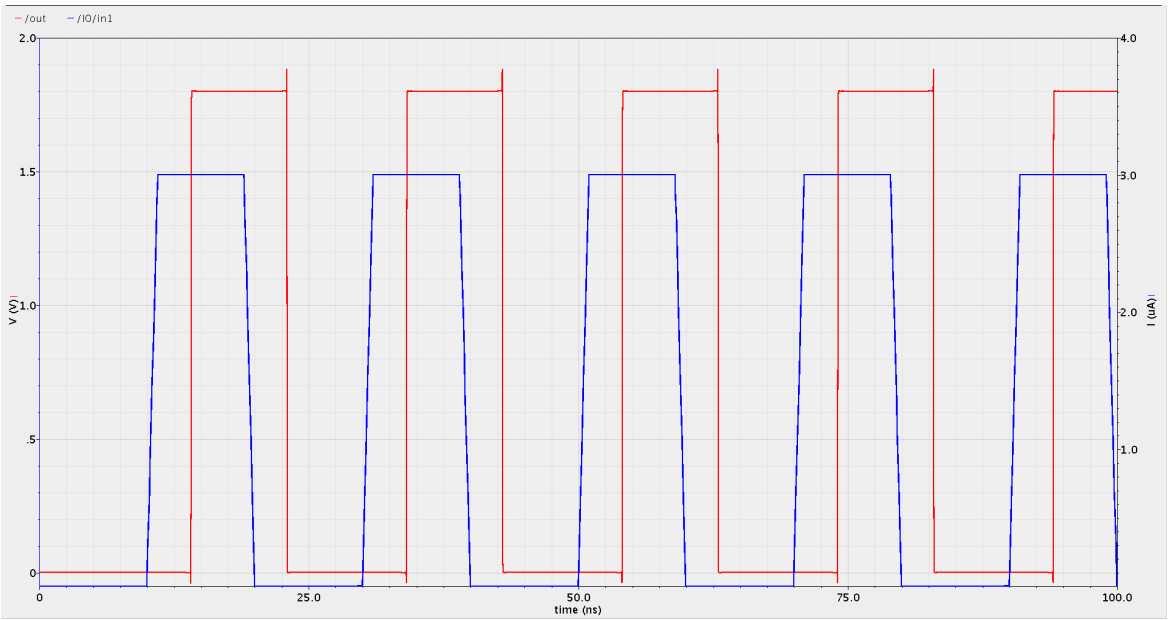


Figure 6.23. DC transfer characteristics of the current mode comparator circuit.

6.4. Integrated Circuit Implementation

180nm CMOS technology provided by UMC through Europractice service has been used for the fabrication of the designed circuits. Technology provides a single polysilicon, and six metal layers for circuit implementation. A silicon die area of $1525 \times 1525 \mu m^2$ has been used for the prototype chip. The final layout of the prototype integrated circuit is shown in Figure 6.24. The prototypes are encapsulated in 44 pin ceramic leaded chip carrier (JLCC44) packages provided by the integrated circuit foundry.

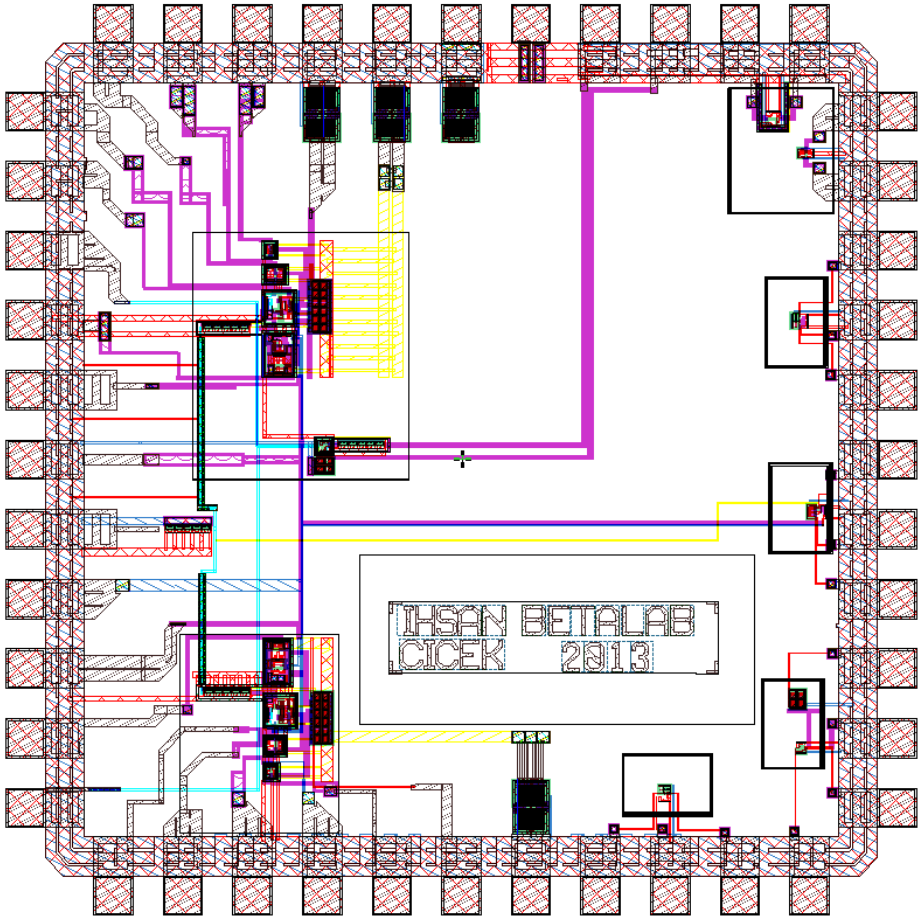


Figure 6.24. Layout of the prototype integrated circuit occupies a die area of $1525 \times 1525 \mu m^2$ designed using UMC 180nm CMOS technology.

In order to reduce substrate coupled noise, each circuit block is guarded using dual guard rings connected to appropriate power rails. Isolation of circuit blocks also helps to avoid potential synchronization of chaotic entropy core circuits.

The microphotograph of the fabricated silicon die is presented in Figure 6.25. Experimental circuit blocks have been placed on the right hand side of the bilateral axis of the chip in order to improve silicon utilization.

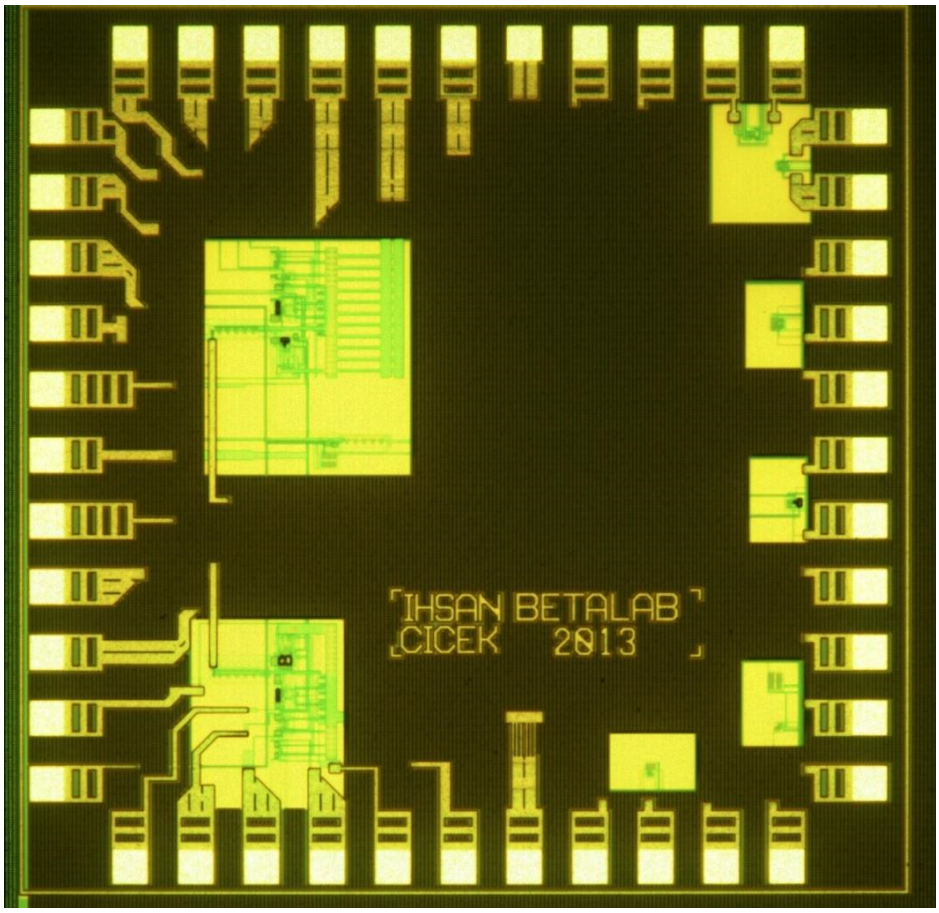


Figure 6.25. Microphotograph of the prototype integrated circuit fabricated in $180nm$ CMOS technology provided by UMC.

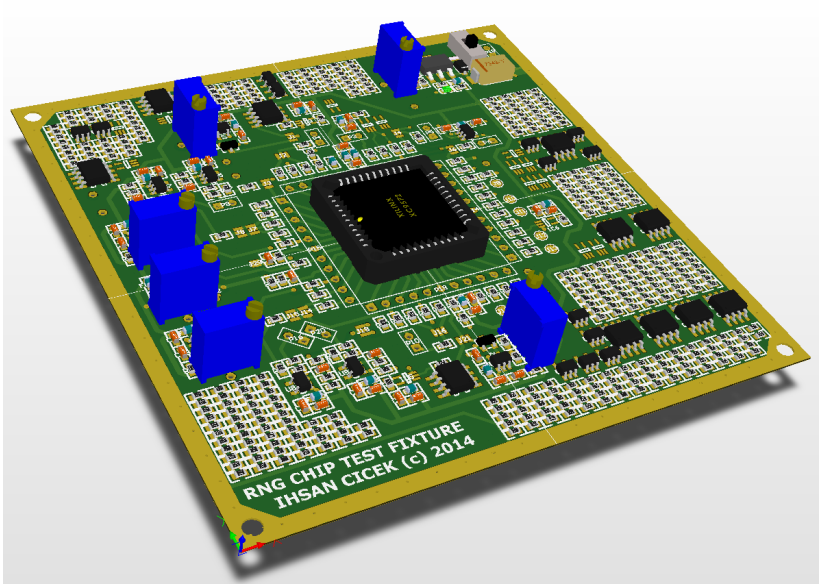
7. CHIP MEASUREMENTS AND STATISTICAL TEST RESULTS

In this chapter, we present the measurement results of the integrated dual entropy core DT chaos based true random number generator architecture. Measurements have been conducted in a well controlled, ESD safe laboratory environment. A custom printed circuit board has been designed, and fabricated as a test fixture for the prototype chip. All the required supplementary circuits such as the voltage regulators, current sources, buffers, and clock generators have been implemented on the test fixture board. The time domain measurements were taken using a Tektronix MSO4034 mixed signal oscilloscope. A Tektronix AFG3101 function generator was used as an external clock source for driving the chaotic dynamics. An off-the-shelf available Xilinx Spartan XC3S1600E based FPGA development board has been used as a data acquisition instrument for transferring generated bitstream to the computer.

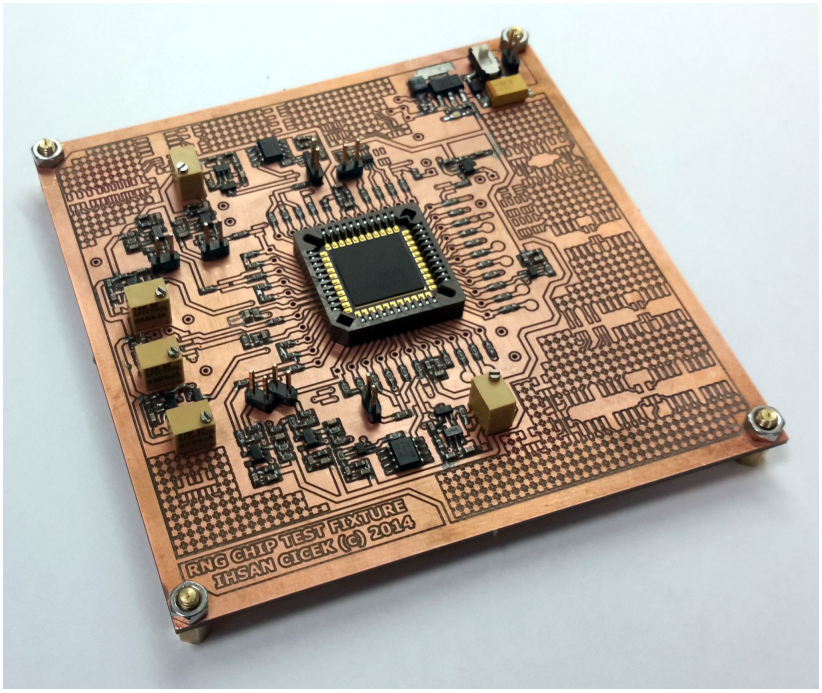
7.1. Test Fixture Board of the Prototype Integrated Circuit

A custom test fixture printed circuit board shown in Figure 7.1 has been designed for functional verification of the integrated dual entropy core DT chaos based TRNG circuit designed in Chapter 6. Designed test chip requires a single ended $1.8V$ DC power supply which is provided with the help of an onboard voltage regulator. In order to minimize power rail coupled noise, appropriate decoupling capacitors, and ferrite beads have been placed at a close location near the PLCC44 socket. A variable onboard oscillator has been included in the design as an alternative auxiliary clock source to the external clock, which is required for driving the chaotic dynamics. Current to voltage converting buffers were designed using low offset, high bandwidth OPAMPs in transimpedance configuration for measuring the current mode state variables in voltage form. Special purpose adjustable current reference integrated circuits have been employed on the test fixture board to provide the required bias, and reference currents for the device under test. Additional bread-board area has been allocated for

hardware debugging. 3D projection of the designed board, and fabricated 2-layer FR4 PCB have been presented in Figure 7.1.



(a)

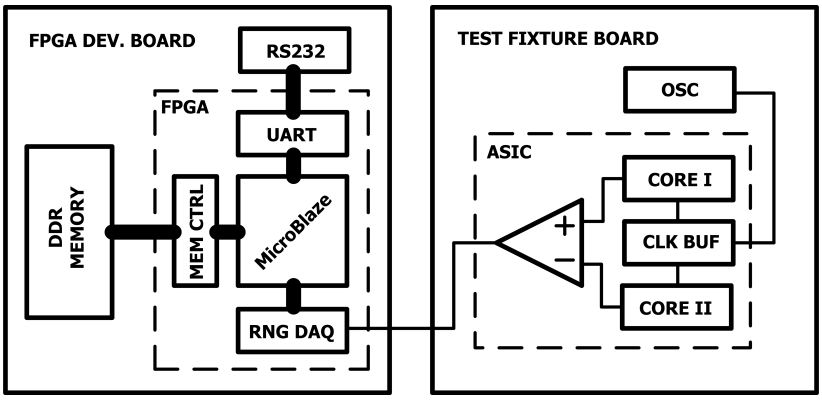


(b)

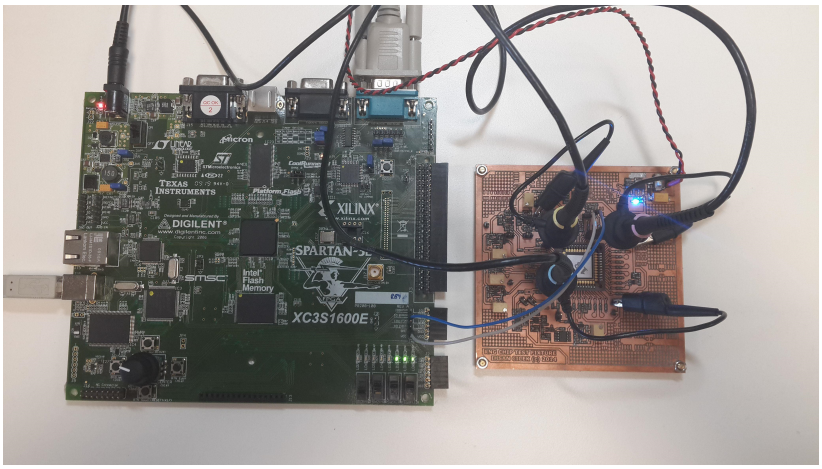
Figure 7.1. 3D picture of the designed test fixture board (a), and its physical realization (b) with prototype integrated circuit mounted in PLCC44 socket.

7.2. Measurement Setup and Data Acquisition System

The measurement setup diagram shown in Figure 7.2 is realized using the custom designed test fixture board, and an off-the-shelf available Xilinx Spartan XC3S1600E development board for evaluating the randomness performance of the integrated dual entropy core TRNG circuit prototype as shown in Figure 7.2.



(a)



(b)

Figure 7.2. Measurement and data acquisition system concept (a), and the physical realization of the measurement setup (b).

We converted current mode state variables into voltage form using the transimpedance amplifiers employed on the test fixture board. These current to voltage conversion buffer amplifiers help to isolate the device under test, and reduce the loading effect of the oscilloscope probes on the pins of the chip.

FPGA development board shown in Figure 7.2 has been utilized to acquire, and transfer the bitstream generated by the integrated dual entropy core TRNG circuit on the test fixture board. A custom 32bit single core Microblaze microcontroller operating at 50MHz with DDR Memory, and UART interfaces has been synthesized using Xilinx embedded development kit. A data acquisition module with programmable sampling clock has been integrated as a custom peripheral within the processor. Inside the data acquisition modue, a 32bit shift register driven by a programmable clock converts serially acquired random bits into 32bit parallel data which is then read by the software running on the Microblaze, and transferred to external DDR SDRAM for temporary storage. After the end of data acquisition session, random number data saved in the external memory is transferred to computer using the standard RS232 interface. A custom software on the computer side writes the incoming data to a file in binary format for statistical analyses.

7.3. Measurement Results of the Prototype Integrated Circuit

In this section, we present the measurement results. Signals generated by the prototype integrated circuit have been measured using Tektronix MSO4034 mixed signal oscilloscope. The device under test was driven by an external clock signal generated by Tektronix AFG3101 signal generator.

7.3.1. Measurement Results of the Bernoulli Map Entropy Core

Bernoulli map entropy core is based on the improved Bernoulli map circuit, and its chaotic dynamics is driven by the externally applied clock signal. External clock source was utilized, since it allowed better control on the slew rate of the clock signal. The chaotic state variable in current form has been converted into voltage with the help of transimpedance buffers on the test fixture board. Chaotic operation of the improved Bernoulli map has been confirmed with time domain, and phase portrait measurements presented in Figure 7.3, and Figure 7.4 respectively.

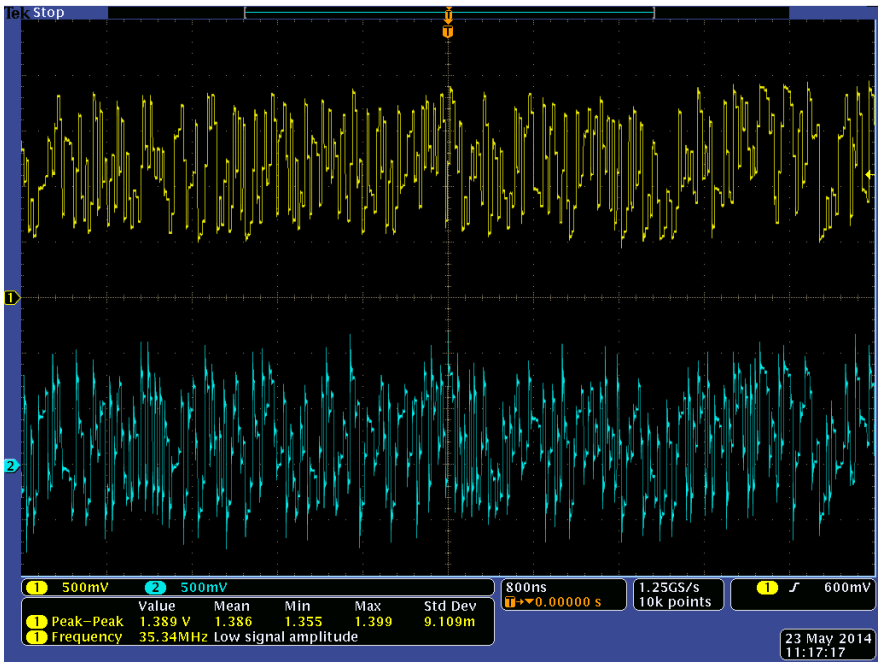


Figure 7.3. Time domain measurement results of the Bernoulli map entropy core.

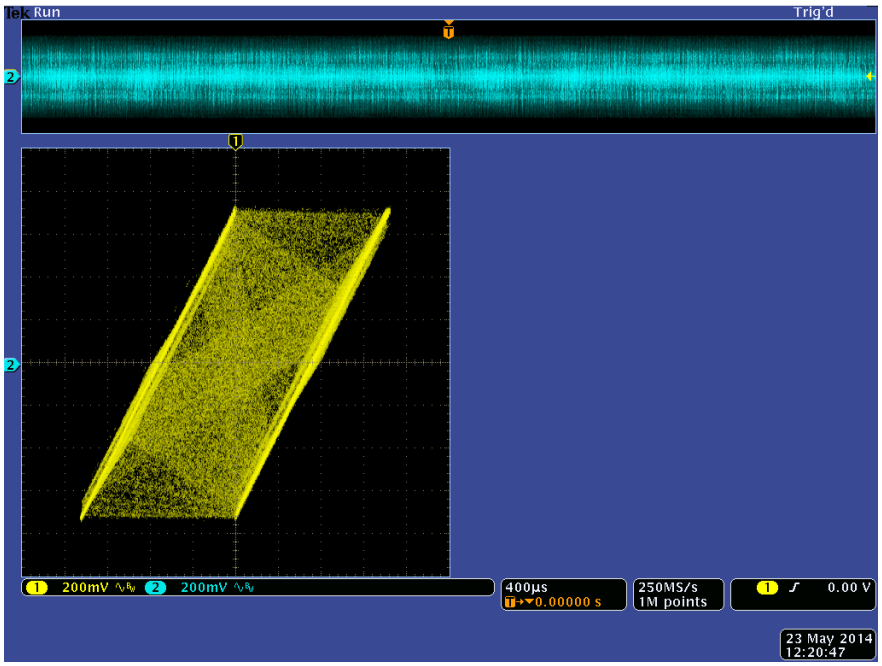


Figure 7.4. Phase portrait measurement results of the Bernoulli map entropy core.

7.3.2. Measurement Results of the Tent Map Entropy Core

Tent map entropy core is based on the improved tent map circuit, and it is operated with the help of an externally applied clock signal. External clock source was preferred because it allows the precise control of the slew rate of the clock signal.

State variable of the chaotic system in current form has been converted into voltage with the help of transimpedance buffers on the test fixture board. Chaotic operation of the improved tent map entropy core has been confirmed with time domain, and phase portrait measurements shown in Figure 7.5, and Figure 7.6 respectively.

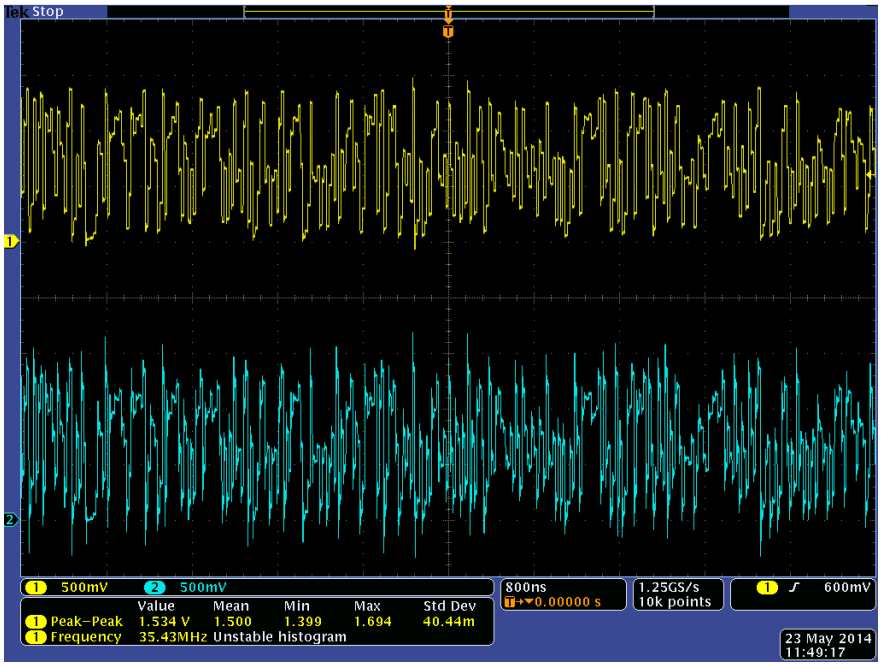


Figure 7.5. Time domain measurement results of the tent map entropy core.

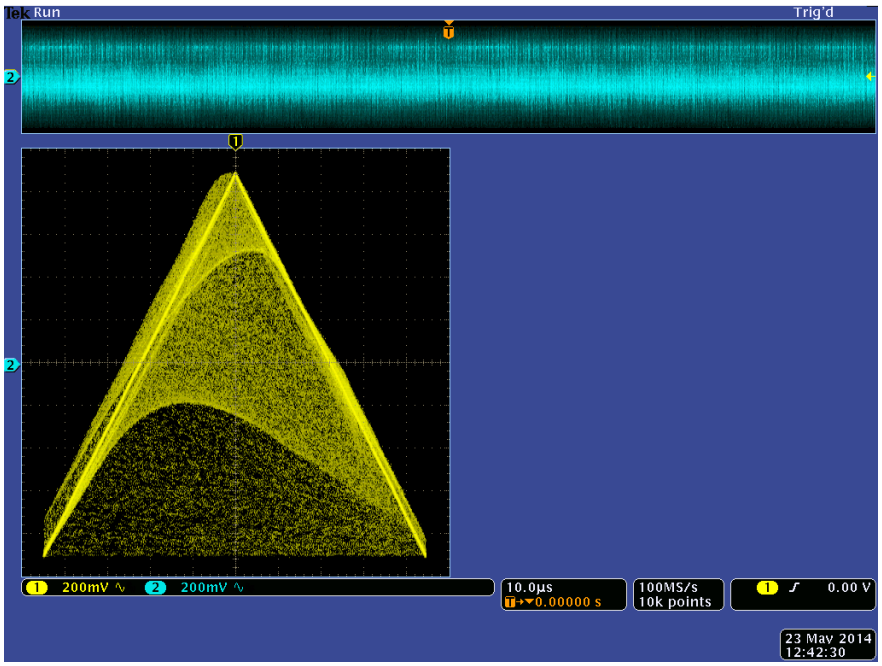


Figure 7.6. Phase portrait measurement results of the tent map entropy core.

7.3.3. Measurement Results of the Dual Entropy Core TRNG

Integrated dual entropy core DT chaos based TRNG is composed of two entropy cores, and one comparator. We used improved Bernoulli, and tent maps for the entropy cores, and a current input, voltage output comparator has been employed to generate random bits. Both entropy cores were operated with the help of an externally applied clock signal. On each clock pulse the chaotic current mode state variables are compared, and random bits are generated through the output of the comparator. Time domain measurements in Figure 7.7 present the chaotic signals generated by each DT chaos based entropy core, and the generated bitstream. The prototype chip acheived approximately 35Mbps throughput.

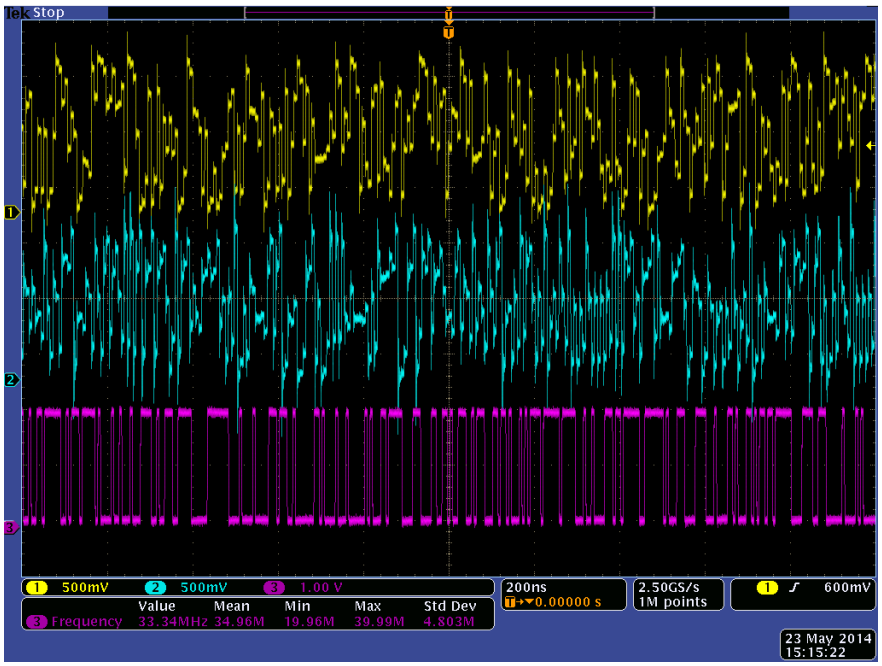


Figure 7.7. Time domain measurement results of the integrated dual entropy core TRNG circuit.

The phase portrait plot has been constructed using the Bernoulli map, and tent map generated chaotic signals as presented in Figure 7.8. It is interesting to note that a homogeneous distribution of the states can be observed in the phase portrait plot of Figure 7.8.

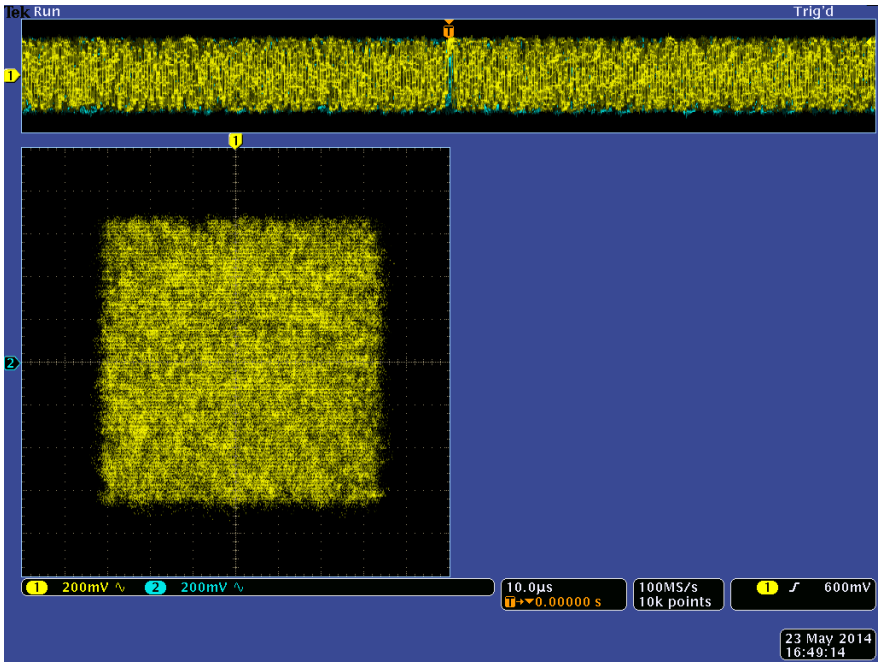


Figure 7.8. Phase portrait measurement results of the integrated dual entropy core TRNG circuit.

The ASIC implementation of the dual entropy core DT chaos based TRNG achieved approximately 35Mbps throughput. The different between design goals, and measurement results needs to be analyzed. We believe that the parasitics of the chip package, and socket when combined with the parasitics induced by the test fixture PCB create a limit on the throughput. We also observed that the operation of the prototype integrated circuit is highly clock slew rate dependent. Consequently, we might not have handled the clock feedthrough phenomena properly. More efficient circuit techniques for addressing the clock feedthrough problem should be employed in future designs. ASIC implementation of the dual entropy core operates faster than its FPAA implementation as anticipated in Chapter 5. We think that higher throughput levels are achievable by using proper techniques for clock signal induced problems, and by using chip packages with lower intrinsic parasitics.

7.4. Statistical Test Results of the Dual Entropy Core TRNG

We used the data acquisition setup shown in Figure 7.2 for capturing, and transferring the random bits generated by the prototype integrated circuit. The bitstream

acquired by the FPGA was written to the external DDR memory until 400Mbits of data have been captured. Then, the random numbers stored in the DDR memory were transferred to the computer using the readily available onboard UART interface upon data acquisition. Although no suite of statistical tests can perfectly qualify a TRNG, they are still the only trusted source in detecting statistical imperfections. NIST800.22 statistical test suite v2.0 was used for the statistical testing of the acquired bitstream. NIST800.22 statistical test suite divides the raw bitstream into 1Mbit blocks, and applies the tests. Statistical test results are presented in Table 7.1. Each p-value that corresponds to a particular test describes the probability of the bitstream generated by an ideal TRNG [33]. Proportion values in Table 7.1 show the ratio of 1Mbit sequences passing the corresponding NIST800.22 test. The statistical test results presented in Table 7.1 confirms that the acquired bitstream successfully passed all the NIST800.22 statistical tests.

Table 7.1. NIST800.22 statistical test results for the bitstream generated by integrated dual entropy core TRNG.

Test	**P-Value**	**Proportion**
Frequency	0.334538	0.9875
Block Frequency	0.951205	0.9900
Cumulative Sums	0.379555	0.9875
Runs	0.585209	0.9900
Longest-Run	0.637119	0.9875
Rank	0.673507	0.9850
FFT	0.788728	0.9825
Universal	0.169512	0.9925
Apen	0.699313	0.9875
Serial	0.759756	0.9925
Linear-Complexity	0.350485	0.9950

8. CONCLUSIONS AND FUTURE WORK

To conclude this dissertation, we summarize the main contributions of the study, and then point out some future work.

8.1. Summary

In this thesis, we have studied the design aspects of DT chaos based true random number generators starting from equations that define the chaotic system, to circuit implementations in FPAA, and ASIC form.

After giving an overview of TRNG systems in the literature, and their classification in Chapter 1, we studied the dynamic, spectral, and statistical characterization of DT chaotic systems that will be used as entropy sources in Chapter 2.

In Chapter 3, we introduced mathematical modeling of DT chaos based TRNGs. The mathematical model of single entropy core TRNG architecture has been presented. we explored the theoretical aspects of true random number generation using the developed models. A novel dual entropy core TRNG architecture, and its mathematical model have been introduced to overcome the limited entropy capacity, and the threshold generation complexity of the single entropy core TRNG architecture. Bitstreams generated by each model were statistically tested using NIST800.22 statistical test suite.

In Chapter 4, a practical information measure, T-entropy, has been used to characterize the randomness performance of the discrete chaotic systems of interest through the digital bitstream generated by the mathematical models developed in Chapter 3. We calculated the maximum allowable parameter variation boundaries

using T-entropy for helping to make the right design decisions in hardware design.

In Chapter 5, FPAA implementations of single, and dual entropy core DT chaos based TRNGs have been introduced. We have witnessed the practical problems of the single entropy core architecture, as predicted before in Chapter 3. FPAA implementation of the novel dual entropy core architecture outperformed its single entropy core counterpart with its promising randomness performance as provisioned by the mathematical models, and T-entropy calculations in Chapter 3, and Chapter 4. Main performance bottleneck for both architectures was the throughput limitation as a result of the implementation technology.

In Chapter 6, we introduced the design aspects for ASIC implementation of the novel dual entropy core DT chaos based TRNG. A new matching driven design methodology has been introduced to implement the DT chaos based entropy cores, sample and hold block, and a current mode comparator. The operation, and design of the circuit blocks have been explained. DC transfer, and time domain characteristics of the entropy cores have been presented by simulation results. Layouts of the designed circuits, and the final layout of the prototype integrated circuit was presented.

In Chapter 7, we presented the measurement results of the fabricated prototype chip. A custom, test fixture PCB has been developed. The time domain operation of the entropy cores composed of Tent map, and Bernoulli map has been shown along with their phase portrait measurements that confirm chaotic operation. Statistical test results of the bitstream acquired using an off-the-shelf available FPGA board show that, the integrated dual entropy core TRNG passed all the tests, without requiring any post processing, practically reassert the validity of the proposed design method, and the superiority of the novel dual entropy core architecture.

8.2. Future Work

We suggest some directions for future research related to this study.

- Foundation of new, efficient, lightweight, and circuit friendly DT chaotic systems would create an application field for the characterization, and design methods introduced in this dissertation. Two dimensional chaotic systems should seriously be considered as potential entropy sources for future lightweight TRNG systems. Discovery of new TRNG architectures capable of improving the randomness performance of the existing single, and dual entropy core TRNGs will be considered as complementary contributions to this study.

- Design aspects of continuous time chaos based TRNG systems might be studied using the characterization, and evaluation methods introduced in this dissertation with little, or no modification. Continuous time chaotic systems usually have large number of Lyapunov exponents, hence they have potential for generating more randomness. Although the additional randomness comes with the cost of increased hardware complexity, a universal design methodology that can be applied to any continuous time chaotic system will be an indispensable tool for reducing design time, and predicting randomness performance ahead of realization.

- The effects of potential side channel attacks on the generated randomness can be studied for exploring the requirements of robust DT chaos based TRNG design. Although the proof of concept design is functional, its utilization requires an in depth analysis, and understanding of its weaknesses. The steps required to create an attack resistant DT chaos based TRNG design will have both theoretical, and practical value.

REFERENCES

1. Neumann, J. V., "Various Techniques Used in Connection with Random Digits", *Applied Math Series.*, Vol. 12, pp. 36–38, 1951.

2. Jun, B. and P. Kocher, "The Intel Random Number Generator", *Cryptography Research, Inc. white paper prepared for Intel Corp.*, 1999.

3. Bock, H., M. Bucci and R. Luzzi, "An Offset-Compensated Oscillator-Based Random Bit Source for Security Applications", M. Joye and J.-J. Quisquater (Editors), *Cryptographic Hardware and Embedded Systems - CHES 2004*, Vol. 3156 of *Lecture Notes in Computer Science*, pp. 268–281, Springer Berlin Heidelberg, 2004.

4. Vincent, C., "The Generation of Truly Random Binary Numbers", *Journal of Physics E.*, Vol. 3, No. 8, pp. 594–598, 1970.

5. Yarza, A. and P. Martinez, "A True Random Pulse Train Generator", *Electronic Engineering*, Vol. 50, No. 614, pp. 21–23, 1978.

6. Bungay, H. and R. Martin, "Truly Random Numbers", *Kilobaud*, Vol. 3, No. 4, pp. 46–47, 1979.

7. Mayhugh, T., "Build a Noise-Based Random Number Generator", *Byte*, pp. 452–456, 1981.

8. Y. Tang, A. M. and L. Chua, "Synchronization and Chaos", *IEEE Transactions on Circuits and Systems*, Vol. 30, No. 9, pp. 620–626, 1983.

9. Fairfield, R. C., R. L. Mortenson and K. B. Coulthart, "An LSI Random Number Generator", *in Proceedings of the CRYPTO84 Conference*, pp. 203–230, 1984.

10. Espejo-Meana, S., J. Martin-Gomez, A. Rodriguez-Vazquez and J. Huertas, "Application of Piecewise-Linear Switched-Capacitor Circuits for Random Number Generation", *in Proceedings of the 32nd Midwest Symposium on Circuits and Systems*, Vol. 2, pp. 960–963, 1989.

11. S. Espejo-Meana, J. H., A. Rodriquez-Vazquez and J. Quintana, "Application of Chaotic Switched-Capacitor Circuits for Random Number Generation", *European Conference on Circuit Theory and Design*, pp. 440–444, 1989.

12. Espejo, S., J. D. Martin, A. Rodriguez-Vazquez and J. Huertas, "Design of an Analog/Digital Truly Random Number Generator", *in Proceedings of the IEEE International Symposium on Circuits and Systems*, Vol. 2, pp. 1368–1371, 1990.

13. Rodriguez-Vazquez, A., S. Espejo-Meana, J. Huertas and J. Martin, "Analog Building Blocks for Noise and Truly Random Number Generation in CMOS VLSI", *Sixteenth European Solid-State Circuits Conference, ESSCIRC '90.*, Vol. 1, pp. 225–228, 1990.

14. Maddocks, R. S. e. a., "A Compact and Accurate Generator for Truly Random Binary Digits", *Journal of Physics E: Scientific Instruments*, Vol. 5, p. 542, 1972.

15. Petrie, C. and J. Connelly, "A Noise-Based IC Random Number Generator for Applications in Cryptography", *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, Vol. 47, No. 5, pp. 615–621, 2000.

16. Bucci, M., L. Germani, R. Luzzi, P. Tommasino, A. Trifiletti and M. Varanonuovo, "A High-Speed IC Random-Number Source for Smartcard Microcontrollers", *IEEE Transactions on Circuits and Systems I*, Vol. 50, No. 11, pp. 1373–1380, 2003.

17. Sunar, B., W. Martin and D. Stinson, "A Provably Secure True Random Number Generator with Built-in Tolerance to Active Attacks", *IEEE Transactions on Computers*, Vol. 56, No. 1, pp. 109–119, 2007.

18. Wold, K. and C. Tan, "Analysis and Enhancement of Random Number Generator in FPGA Based Oscillator Rings", *Proceedings of the International Conference on ReConfigurable Computing and FPGAs (Reconfig 2008)*, pp. 385–390, 2008.

19. Wold, K. and C. H. Tan, "Analysis and Enhancement of Random Number Generator in FPGA Based on Oscillator Rings", *International Journal of Reconfigurable Computing*, Vol. 2009, pp. 4:1–4:8, 2009.

20. Markettos, A. T. and S. W. Moore, "The Frequency Injection Attack on Ring-Oscillator-Based True Random Number Generators", *Proceedings of the 11th International Workshop on Cryptographic Hardware and Embedded Systems*, CHES '09, pp. 317–331, Springer-Verlag, Berlin, Heidelberg, 2009.

21. Poucheret, F., K. Tobich, M. Lisarty, L. Chusseau, B. Robisson and P. Maurine, "Local and Direct EM Injection of Power Into CMOS Integrated Circuits", *Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC)*, pp. 100–104, 2011.

22. Poucheret, F., L. Chusseau, B. Robisson and P. Maurine, "Local Electromagnetic Coupling with CMOS Integrated Circuits", *8th Workshop on Electromagnetic Compatibility of Integrated Circuits (EMC Compo)*, pp. 137–141, 2011.

23. Bayon, P., L. Bossuet, A. Aubert, V. Fischer, F. Poucheret, B. Robisson and P. Maurine, "Contactless Electromagnetic Active Attack on Ring Oscillator Based True Random Number Generator", *Proceedings of the Third international conference on Constructive Side-Channel Analysis and Secure Design*, COSADE'12, pp. 151–166, Springer-Verlag, Berlin, Heidelberg, 2012.

24. Eckmann, J. P. and D. Ruelle, "Ergodic Theory of Chaos and Strange Attractors", *Reviews of Modern Physics*, Vol. 57, No. 3, pp. 617–656, 1985.

25. Ott, E., *Chaos in Dynamical Systems*, Cambridge University Press, 2002.

26. Alligood, K., T. Sauer and J. Yorke, *Chaos: An Introduction to Dynamical Systems*, Chaos: An Introduction to Dynamical Systems, New York, NY, 1997.

27. Yalcin, M., J. Suykens and J. Vandewalle, "True Random Bit Generation from a Double Scroll Attractor", *IEEE Transactions on Circuits and Systems I*, Vol. 51, No. 7, pp. 1395–1404, 2004.

28. Tavas, V., A. Demirkol, S. Ozoguz, A. Zeki and A. Toker, "Integrated Cross-Coupled Chaos Oscillator Applied to Random Number Generation", *Circuits, Devices Systems, IET*, Vol. 3, No. 1, pp. 1–11, 2009.

29. T. Stojanovski, J. P. and L. Kocarev, "Chaos-Based Random Number Generators-Part II: Practical Realization", *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, Vol. 48, No. 3, pp. 382–385, 2001.

30. Wang, Y.-H., H.-G. Zhang, Z.-D. Shen and K.-S. Li, "Thermal Noise Random Number Generator Based on SHA-2 (512)", *in Proceedings of the International Conference on Machine Learning and Cybernetics*, Vol. 7, pp. 3970–3974, 2005.

31. F. Pareschi, G. S. and R. Rovatti, "A Fast Chaos-based True Random Number Generator for Cryptographic Applications", *in Proceedings of the IEEE 32nd European Solid-State Circuits Conference, ESSCIRC 2006*, pp. 130–133, 2006.

32. Soto, J., "Statistical Testing of Random Number Generators", *in Proceedings of the 22nd National Information Systems Security Conference*, 1999.

33. Bassham, L. E., III, A. L. Rukhin, J. Soto, J. R. Nechvatal, M. E. Smid, E. B. Barker, S. D. Leigh, M. Levenson, M. Vangel, D. L. Banks, N. A. Heckert, J. F. Dray and S. Vo, *SP 800-22 Rev. 1a. A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*, Tech. rep., Gaithersburg, MD, United States, 2010.

34. NIST, *FIPS PUB 140-2, Security Requirements for Cryptographic Modules*, FIPS-140-2, U.S. DoC/National Institute of Standards and Technology, 2002.

35. Schindler, W. and W. Killmann, "Evaluation Criteria for True (Physical) Random Number Generators Used in Cryptographic Applications", *Revised Papers from the 4th International Workshop on Cryptographic Hardware and Embedded Systems*, CHES '02, pp. 431–449, Springer-Verlag, London, UK, 2003.

36. Cicek, I. and G. Dundar, "A Hardware Efficient Chaotic Ring Oscillator Based True Random Number Generator", *18th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, pp. 430–433, 2011.

37. Cicek, I., A. Pusane and G. Dundar, "A Feasibility Study of a 1D Chaotic Map for True Random Number Generation", *20th Signal Processing and Communications Applications Conference (SIU)*, pp. 1–4, 2012.

38. Cicek, I., A. Pusane and G. Dundar, "Field Programmable Analog Array Implementation of Logistic Map", *21st Signal Processing and Communications Applications Conference (SIU)*, pp. 1–4, 2013.

39. Cicek, I., A. Pusane and G. Dundar, "Random Number Generation Using Field Programmable Analog Array Implementation of Logistic Map", *21st Signal Processing and Communications Applications Conference (SIU)*, pp. 1–4, 2013.

40. Cicek, I., A. 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.

41. Cicek, I. and G. Dundar, "A Chaos Based Integrated Jitter Booster for True Random Number Generators", *21th IEEE European Conference on Circuit Theory and Design, ECCTD 2013*, pp. 1–4, 2013.

42. Cicek, I., A. Pusane and G. Dundar, "A Novel Dual Entropy Core True Random

Number Generator", *8th International Conference on Electrical and Electronics Engineering (ELECO)*, pp. 1–4, 2013.

43. Cicek, I., A. E. Pusane and G. Dundar, "A New Dual Entropy Core True Random Number Generator", *Analog Integrated Circuits and Signal Processing*, pp. 1–10, 2014, `http://dx.doi.org/10.1007/s10470-014-0324-y`, invited paper, in press.

44. Verhulst, P. F., "Recherches sur la loi d'Mathematics Accro-Issement de la Population.", *L'Acad'emie Royale de Bruxel-les et de l'Universit' e Louvain*, Vol. 18, No. 1, pp. 1–42, 1845.

45. May, R. M., "Simple Mathematical Models with Very Complicated Dynamics", *Nature*, Vol. 261, No. 5560, pp. 459–467, 1976.

46. Welch, P. D., "The Use of Fast Fourier Transform for the Estimation of Power Spectra: A Method Based on Time Averaging Over Short, Modified Periodograms", *IEEE Transactions on Audio and Electroacoustics*, Vol. 15, pp. 70–73, 1967.

47. Lasota, A. and M. C. Mackey, *Chaos, Fractals, and Noise Stochastic Aspects of Dynamics. Second Edition. Applied Mathematical Sciences, 97*, Vol. 97 of *Applied Mathematical Sciences*, Springer-Verlag, New York, 1994.

48. Ulam, S. and J. von Neumann, "On Combination of Deterministic and Stochastic Processes", *The Summer Meeting in New Haven*, Vol. 53, pp. 1120+, 1947.

49. Pesin, Y. B., "Characteristic Lyapunov Exponents and Smooth Ergodic Theory", *Russian Mathematical Surveys*, Vol. 32, No. 4, p. 55, 1977.

50. Kolmogorov, A. N., "Invariant Measures of Tunable Chaotic Sources: Robustness Analysis and Efficient Estimation", *Doklady Akademii Nauk SSSR*, Vol. 119, pp. 861–864, 1958.

51. Hoppensteadt, F., *Analysis and Simulation of Chaotic Systems*, Analysis and Simulation of Chaotic Systems, Springer, 2000.

52. Pelgrom, M., A. C. J. Duinmaijer and A. Welbers, "Matching Properties of MOS Transistors", *IEEE Journal of Solid-State Circuits*, Vol. 24, No. 5, pp. 1433–1439, 1989.

53. Pelgrom, M., H. Tuinhout and M. Vertregt, "Transistor Matching in Analog CMOS Applications", *Electron Devices Meeting, 1998. IEDM '98. Technical Digest., International*, pp. 915–918, 1998.

54. Kinget, P., "Device Mismatch and Tradeoffs in the Design of Analog Circuits", *IEEE Journal of Solid-State Circuits*, Vol. 40, No. 6, pp. 1212–1224, 2005.

55. Kinget, P., "Device Mismatch: An Analog Design Perspective", *in Proceedings of the IEEE International Symposium on Circuits and Systems, ISCAS 2007*, pp. 1245–1248, 2007.

56. Shannon, C. E., "A Mathematical Theory of Communication", *The Bell System Technical Journal*, Vol. 27, pp. 379–423, 623–656, 1948.

57. Shannon, C. and W. Weaver, *The Mathematical Theory of Communication.*, University of Illinois Press, 1949.

58. Lawrance, A. J. and R. C. Wolff, "Binary Time Series Generated by Chaotic Logistic Maps", *Stochastics and Dynamics*, Vol. 3, No. 4, pp. 529–544, 2003.

59. Titchener, M., "Deterministic Computation of Complexity, Information and Entropy", *in Proceedings of the IEEE International Symposium on Information Theory*, p. 326, 1998.

60. Steuer, R., W. B. Ebeling and M. R. Titchener, "Partition Based Entropies of Dynamic and Stochastic Maps", *Stochastics and Dynamics*, Vol. 1, No. 1, pp.

45–61, 2001.

61. Titchener, M. R. and W. B. Ebeling, "Deterministic Chaos and Information Theory.", *Data Compression Conference*, p. 520, IEEE Computer Society, 2001.

62. Ziv, J. and A. Lempel, "A Universal Algorithm for Sequential Data Compression", *IEEE Transactions on Information Theory*, Vol. 23, No. 3, pp. 337–343, 1977.

63. Titchener, M., "A Measure of Information", *Data Compression Conference, 2000. Proceedings. DCC 2000*, pp. 353–362, 2000.

64. Odame, K., C. M. Twigg, A. Basu and P. E. Hasler, "Studying Nonlinear Dynamical Systems on a Reconfigurable Analog Platform", *in Proceedings of the IEEE International Symposium Circuits and Systems, ISCAS 2007*, pp. 445–448, 2007.

65. Anadigm, *The AN231E04 dpASP Dynamically Reconfigurable Analog Signal Processor AN231E04 Datasheet Rev. 1.1*, 2008, `http://www.anadigm.com/_doc/DS231000-u001.pdf`, accessed in May 2011.

66. Lopez-Hernandez, J., A. Diaz-Mendez, R. Vazquez-Medina and R. Alejos-Palomares, "Analog Current-Mode Implementation of a Logistic-Map Based Chaos Generator", *in Proceedings of the 52nd IEEE International Midwest Symposium Circuits and Systems, MWSCAS '09*, pp. 812–814, 2009.

67. Diaz-Mendez, A., J. Marquina-Perez, M. Cruz-Irisson, R. Vazquez-Medina and J. Del-Rio-Correa, "Chaotic Noise {MOS} Generator Based on Logistic Map", *Microelectronics Journal*, Vol. 40, No. 3, pp. 638 – 640, 2009.

68. Ding, Q., Y. Zhu, F. Zhang and X. Peng, "Discrete Chaotic Circuit and The Property Analysis of Output Sequence", *in Proceedings IEEE International Symposium on Communications and Information Technology, ISCIT 2005*, Vol. 2, pp. 1043–1046, 2005.

69. McGonigal, G. and M. Elmasry, "Generation of Noise by Electronic Iteration of the Logistic Map", *IEEE Journal of Circuits And Systems*, Vol. 34, No. 8, pp. 981–983, 1987.

70. Rodriguez-Vazquez, A., J. L. Huertas, A. Rueda, B. Perez-Verdu and L. O. Chua, "Chaos from Switched-Capacitor Circuits: Discrete Maps", *IEEE Journal of Proceedings*, Vol. 75, No. 8, pp. 1090–1106, 1987.

71. Rodriguez-Vazquez, A., J. Huertas and L. Chua, "Chaos in Switched-Capacitor Circuit", *IEEE Journal of Circuits And Systems*, Vol. 32, No. 10, pp. 1083–1085, 1985.

72. Anadigm, *AN231K04-DVLP3 Anadigm Apex Development Board User Manual*, 2009, `http://www.anadigm.com/_doc/UM231000-K001.pdf`, accessed in May 2011.

73. Xilinx, *Xilinx Spartan 3E-1600 Development Board Reference Manual*, 2007, `http://www.digilentinc.com/Data/Products/S3E1600/ug257.pdf`, accessed in May 2011.

74. Callegari, S., R. Rovatti and G. Setti, "First Direct Implementation of a True Random Source on Programmable Hardware", *International Journal of Circuit Theory and Applications*, Vol. 33, pp. 1–16, 2005.

75. Callegari, S., R. Rovatti and G. Setti, "Embeddable ADC-Based True Random Number Generator for Cryptographic Applications Exploiting Nonlinear Signal Processing and Chaos", *IEEE Transactions on Signal Processing*, Vol. 53, No. 2, pp. 793–805, 2005.

76. Addabbo, T., M. Alioto, A. Fort, S. Rocchi and V. Vignoli, "Uniform-Distributed Noise Generator Based on a Chaotic Circuit", *in Proceedings of the IEEE Instrumentation and Measurement Technology Conference, IMTC 2006*, pp. 1156–1160, 2006.

77. Rodriguez-Vazquez, A., A. Rueda, B. Perez-Verdu and J. Huertas, "Chaos via a Piecewise-Linear Switched-Capacitor Circuit", *Electronics Letters*, Vol. 23, No. 12, pp. 662–663, 1987.

78. Rodriguez-Vazquez, A., M. Delgado, S. Espejo and J. Huertas, "Switched-Capacitor Broadband Noise Generator for CMOS VLSI", *Electronics Letters*, Vol. 27, No. 21, pp. 1913–1915, 1991.

79. Delgado-Restituto, M., A. Rodriguez-Vazquez and J. Huertas, "1/fy Noise Generation Through a Chaotic Nonlinear Switched-Capacitor Circuit", *in Proceedings of the 34th Midwest Symposium on Circuits and Systems*, pp. 52–55 Vol.1, 1991.

80. Delgado-Restituto, M. and A. Rodriguez-Vazquez, "Mixed-Signal Map-Configurable Integrated Chaos Generator for Chaotic Communications", *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, Vol. 48, No. 12, pp. 1462–1474, 2001.

81. Delgado-Restituto, M. and A. Rodriguez-Vazquez, "Integrated Chaos Generators", *Proceedings of the IEEE*, Vol. 90, No. 5, pp. 747–767, 2002.

82. Wang, C.-C., J.-M. Huang, H.-C. Cheng and R. Hu, "Switched-Current 3-bit CMOS 4.0-MHz Wideband Random Signal Generator", *IEEE Journal of Solid-State Circuits*, Vol. 40, No. 6, pp. 1360–1365, 2005.

83. Wang, C.-C., Y.-L. Tseng, H.-C. Cheng and R. Hu, "Switched-Current 3-bit CMOS Wideband Random Signal Generator", *Southwest Symposium on Mixed-Signal Design*, pp. 186–189, 2003.

84. Katz, O., D. Ramon and I. Wagner, "A Robust Random Number Generator Based on a Differential Current-Mode Chaos", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 16, No. 12, pp. 1677–1686, 2008.

85. Ramirez-Angulo, J., E. Sanchez-Sinencio and A. Rodriguez-Vazquez, "A

Piecewise-Linear Function Approximation Using Current Mode Circuits", *in Proceedings of the IEEE International Symposium on Circuits and Systems, ISCAS '92*, Vol. 4, pp. 2025–2028, 1992.

86. Espejo, S., A. Rodriguez-Vazquez, R. Dominguez-Castro, B. Linares and J. Huertas, "A Model for VLSI Implementation of CNN Image Processing Chips Using Current-Mode Techniques", *in Proceedings of the IEEE International Symposium on Circuits and Systems, ISCAS '93*, pp. 970–973, 1993.

87. Degaldo-Restituto, M., F. Medeiro and A. Rodriguez-Vazquez, "Nonlinear Switched-Current CMOS IC for Random Signal Generation", *Electronics Letters*, Vol. 29, No. 25, pp. 2190–2191, 1993.

88. Delgado-Restituto, M. and A. Rodriguez-Vazquez, "Current-Mode Building Blocks for CMOS-VLSI Design of Chaotic Neural Networks", *IEEE International Conference on Neural Networks, IEEE World Congress on Computational Intelligence*, Vol. 3, pp. 1993–1997, 1994.

89. Delgado-Restituto, M. and A. Rodriguez-Vazquez, "CMOS Current-Mode Chaotic Neurons", *1994 IEEE International Symposium on Circuits and Systems, ISCAS '94*, Vol. 6, pp. 499–502, 1994.

90. Delgado-Restituto, M. and A. Rodriguez-Vazquez, "Current-Mode Building Blocks for CMOS-VLSI Design of Chaotic Neural Networks", *IEEE International Conference on Neural Networks, IEEE World Congress on Computational Intelligence*, Vol. 6, pp. 3973–3977, 1994.

91. Delgado-Restituto, M., R. de Ahumada and A. Rodriguez-Vazquez, "Secure Communication Through Switched-Current Chaotic Circuits", *IEEE International Symposium on Circuits and Systems, ISCAS '95*, Vol. 3, pp. 2237–2240, 1995.

92. Addabbo, T., A. Fort, D. Papini, S. Rocchi and V. Vignoli, "Invariant Measures of Tunable Chaotic Sources: Robustness Analysis and Efficient Estimation", *IEEE*

Transactions on Circuits and Systems I: Regular Papers, Vol. 56, No. 4, pp. 806–819, 2009.

93. Van Peteghem, P. and W. Sansen, "Single Versus Complementary Switches: A Discussion of Clock Feedthrough in S.C. Circuits", *Twelfth European Solid-State Circuits Conference, ESSCIRC '86*, pp. 143–145, 1986.

94. Eichenberger, C. and W. Guggenbuhl, "Charge Injection of Analogue CMOS Switches", *IEE Proceedings G: Circuits, Devices and Systems*, Vol. 138, No. 2, pp. 155–159, 1991.

95. Eichenberger, C. and W. Guggenbuhl, "On Charge Injection in Analog MOS Switches and Dummy Switch Compensation Techniques", *IEEE Transactions on Circuits and Systems*, Vol. 37, No. 2, pp. 256–264, 1990.

96. Wegmann, G., E. Vittoz and F. Rahali, "Charge Injection in Analog MOS Switches", *IEEE Journal of Solid-State Circuits*, Vol. 22, No. 6, pp. 1091–1097, 1987.

97. Shieh, J.-H., M. Patil and B. Sheu, "Measurement and Analysis of Charge Injection in MOS Analog Switches", *IEEE Journal of Solid-State Circuits*, Vol. 22, No. 2, pp. 277–281, 1987.

98. Wang, C., "A Minimization of the Charge Injection in Switched-Current Circuits", *in Proceedings of the International Symposium on Circuits and Systems, ISCAS '04*, Vol. 1, pp. I–905–8, 2004.

99. Eichenberger, C. and W. Guggenbuhl, "Dummy Transistor Compensation of Analog MOS Switches", *IEEE Journal of Solid-State Circuits*, Vol. 24, No. 4, pp. 1143–1146, 1989.

100. Freitas, D. and K. Current, "CMOS Current Comparator Circuit", *Electronics Letters*, Vol. 19, No. 17, pp. 695–697, 1983.

101. Dominguez-Castro, R., A. Rodriguez-Vazquez, F. Medeiro and J. Huertas, "High Resolution CMOS Current Comparators", *Eighteenth European Solid-State Circuits Conference, ESSCIRC '92*, pp. 242–245, 1992.

102. Traff, H., "Novel Approach to High Speed CMOS Current Comparators", *Electronics Letters*, Vol. 28, No. 3, pp. 310–312, 1992.

103. Tang, A. and C. Toumazou, "High Performance CMOS Current Comparator", *Electronics Letters*, Vol. 30, No. 1, pp. 5–6, 1994.

104. Ravezzi, L., D. Stoppa and G.-F. Dalla Betta, "Simple High-Speed CMOS Current Comparator", *Electronics Letters*, Vol. 33, No. 22, pp. 1829–1830, 1997.