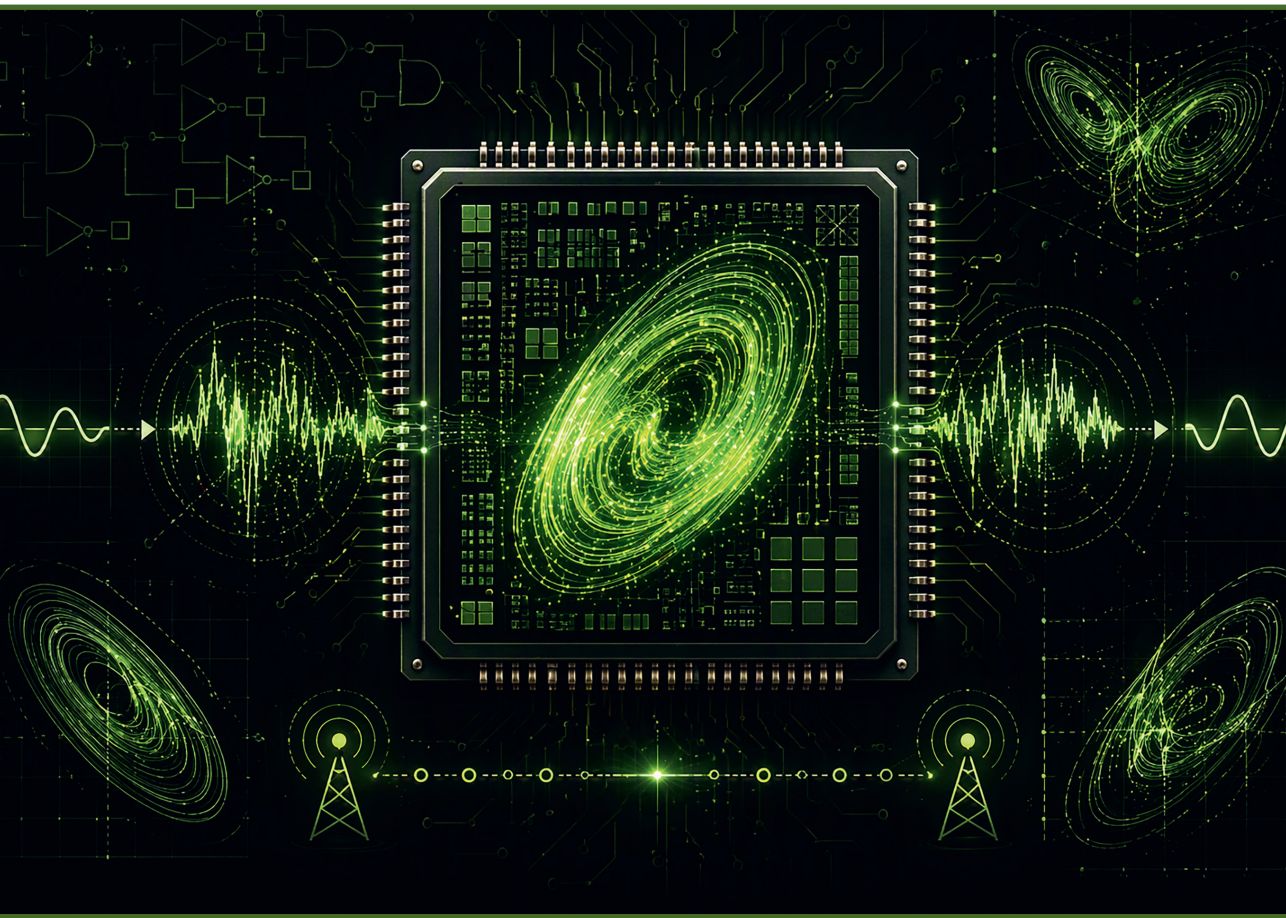
RIGA TECHNICAL UNIVERSITY

Filips Čapligins

FPGA IMPLEMENTATION OF AM-ACSK AND FM-ACSK DIGITAL CHAOTIC COMMUNICATION SYSTEMS

Summary of the Doctoral Thesis



RTU Press
Riga 2026

RIGA TECHNICAL UNIVERSITY

Faculty of Computer Science, Information Technology and Energy

Institute of Photonics, Electronics and Telecommunications

Filips Čapligins

Doctoral Student of the Study Programme "Electronics"

FPGA IMPLEMENTATION OF AM-ACSK AND FM-ACSK DIGITAL CHAOTIC COMMUNICATION SYSTEMS

Summary of the Doctoral Thesis

Scientific supervisors:

Professor Dr. sc. ing.

ANNA LITVIŅENKO

Associate Professor PhD

DENISS KOLOSOVS

RTU Press

Riga 2026

Čapligins, F. FPGA Implementation of AM-ACSK and FM-ACSK Digital Chaotic Communication Systems. Summary of the Doctoral Thesis. – Riga: RTU Press, 2026. – 50 p.

Published in accordance with the decision of the Promotion Council “RTU P-08” of 15 May 2026, Minutes No. 52.

The research was partially supported by research and development grant No. C4835.Dok.1076 under the EU RRF project No. 5.2.1.1.i.0/2/24/I/CFLA/003.

Cover image generated using Microsoft Designer.
https://doi.org/10.7250/9789934373299
ISBN 978-9934-37-329-9 (pdf)

DOCTORAL THESIS PROPOSED TO RIGA TECHNICAL UNIVERSITY FOR PROMOTION TO THE SCIENTIFIC DEGREE OF DOCTOR OF SCIENCE

To be granted the scientific degree of Doctor of Science (PhD), the present Doctoral Thesis has been submitted for defence at the open meeting of RTU Promotion Council on 11 September 2026. at 11:00 at the Faculty of Computer Science, Information Technology and Energy of Riga Technical University, Āzenes iela 12, Room 201.

OFFICIAL REVIEWERS

Associate Professor Dr. sc. ing. Andis Supe
Riga Technical University

Professor Dr Darius Plonis
Vilnius Gediminas Technical University, Lithuania

Professor Dr. sc. ing., Dr. habil. sc. ing. Igor Kabashkin
Transport and Telecommunications Institute, Latvia

DECLARATION OF ACADEMIC INTEGRITY

I hereby declare that the Doctoral Thesis submitted for review to Riga Technical University for promotion to the scientific degree of Doctor of Science (PhD) is my own. I confirm that this Doctoral Thesis has not been submitted to any other university for promotion to a scientific degree.

Filips Čapligins _____

Date: _____

The Doctoral Thesis has been written in English. It consists of an introduction, five sections, conclusions, 56 figures, eight tables, and three appendices. The total number of pages is 96, excluding appendices. The bibliography contains 90 titles.

CONTENTS

ACRONYMS

ACSK	antipodal chaos shift keying
ADC	analog-to-digital converter
AGC	automatic gain control
AM	amplitude modulation
AM-ACSK	amplitude-modulated antipodal chaos shift keying
AWGN	additive white Gaussian noise
BER	bit error rate
C-PSK	chaotic phase shift keying
CDMA	code division multiple access
CIC	cascaded integrator-comb
COOK	chaos on-off keying
CPSK	chaos parameter shift keying
CSK	chaos shift keying
CSS	chaos-based selective symbol
DAC	digital-to-analog converter
DCSK	differential chaos shift keying
DMA	direct memory access
DSP	digital signal processing
FEC	forward error correction
FIFO	first input first output
FIR	finite impulse response
FM	frequency modulation
FM-ACSK	frequency-modulated antipodal chaos shift keying
FPGA	field-programmable gate array
HDL	hardware description language
HPS	hard processing system
HSMC	high speed mezzanine card
IC	integrated circuit
IIR	infinite impulse response
IoT	Internet of Things
IP	intellectual property
LED	light-emitting diode
LFSR	linear feedback shift register
LUT	lookup table
MPU	microprocessor unit
MSB	most significant bit
NCO	numerically controlled oscillator
ODE	ordinary differential equation
OPCL	open-plus-closed-loop
PC	personal computer
PTCFZNN	preset-time convergence fuzzy zeroing neural network
RCO	reconfigurable chaotic oscillator
RK-4	fourth-order Runge-Kutta
RMS	root mean square
SDRAM	synchronous dynamic random-access memory
SFDR	spurious-free dynamic range
SNR	signal-to-noise ratio
SOB	straight offset binary
SoC	system-on-chip
TED	timing error detector
UART	universal asynchronous receiver/-transmitter
USB	universal serial bus
VHDL	very high speed integrated circuit hardware description language
WCDMA	wideband code division multiple access
XOR	exclusive OR

INTRODUCTION

Relevance

In recent decades, the necessity for communication infrastructures with increased signal protection has grown dramatically. Telecommunications play a pivotal role in fostering worldwide connectivity, and the expansion of the Internet of Things (IoT) has further augmented this need. In the context of escalating cyber vulnerabilities and privacy risks, reliable data transmission has emerged as a paramount priority. Conventional security strategies are predominantly reliant on encryption algorithms, which are susceptible to obsolescence due to advancements in computing technology, such as quantum mechanics. These methods primarily prioritize the safeguarding of upper-layer protocols, frequently overlooking the protective measures necessary for the physical transmission layer. Chaotic communication systems present a compelling alternative, taking advantage of the nonlinear and unpredictable nature of chaotic waveforms to enhance protection and deter unauthorized interception.

Research into the application of chaos generators for communication systems dates back to the 1990s, with one of their primary benefits being enhanced physical-layer signal protection [1]. Utilizing chaotic oscillations as carriers renders signal detection and information extraction highly challenging due to the inherently unpredictable signal traits. The incorporation of chaotic synchronization into coherent detection schemes serves to further amplify this protection, thereby necessitating an identical chaos generator at the receiver – with matching parameters and architecture – to achieve demodulation [2]. Furthermore, chaotic signals exhibit robust resilience to interception, demonstrating notable tolerance to multipath effects and deliberate jamming [3]. The inherent nonlinear, ergodic, and pseudorandom characteristics of these systems foster unpredictability, thereby complicating unauthorized access without exact system knowledge. When considered in junction with narrow autocorrelation, reduced cross-signal correlations, and broad spectral spread, these attributes position chaotic systems as a viable option for bolstering physical-layer defenses in wireless transmissions, rendering them a valuable subject for scholarly exploration and practical signal protection applications.

The utilization of chaos generators has historically been confined to the domain of analog circuits. However, the implementation of analog chaos generators within communication systems poses significant challenges, including the absence of initial state control and the inherent parametric instability of components. This results in the complexity of drift compensation and the unreliability of chaotic synchronization [4]. These disadvantages have prompted a shift towards the discrete digital implementations on field-programmable gate arrays (FPGAs) to eliminate drift issues and enable deterministic, high-throughput parallel signal processing via pipelining [3]. In the contemporary context, the reprogrammable nature and hardware description language support of FPGAs have led to a resurgence of interest in chaotic communication application studies [5], [6]. However, despite

this growing interest, the practical implementation of complete chaotic communication transceivers on FPGAs and their experimental performance evaluation remain comparatively limited in the existing literature. Furthermore, there is an ongoing need to optimize resource allocation in order to maintain chaos fidelity while meeting efficiency requirements.

Research Goal and Tasks

The goal of this Thesis is to research the coherent discrete chaotic communication system design options and to develop a set of approaches for the FPGA implementation of such systems. In order to accomplish the objective, the following tasks have been established.

- Analytical study on chaotic communication and FPGA implementation of chaotic systems.
- Design mathematical models for coherent discrete AM-ACSK and FM-ACSK chaotic communication systems.
- Perform the proof-of-concept verification via simulations.
- Development of the FPGA prototypes for the designed AM-ACSK and FM-ACSK chaotic communication systems with VHDL code.
- Perform the experimental verifications of the FPGA prototypes.

Scientific Novelty and Main Results

The study's primary results and scientific novelties are as follows.

- The original discrete adaptation of a modified Chua circuit chaos generator, error-feedback synchronization, and AM-ACSK alongside FM-ACSK modulated communication systems designed for FPGA hardware integration.
- The development of a Simulink mathematical model that precisely reflects the digital components of the proposed AM-ACSK and FM-ACSK communication system prototypes.
- The structural design and FPGA implementation of novel AM-ACSK and FM-ACSK discrete chaotic communication system transceivers, as well as their practical validation through operational tests, have confirmed the viability of both systems and advanced the conceptual development of chaotic communication technologies.

The following components are of particular significance.

- Transmitters: Modified Chua chaos generator (ODE discretized with forward Euler and fixed-point arithmetic), ACSK modulation (8192-sample symbols), NCO-based AM/FM 10.7 MHz passband.
- Receivers: Signal processing including incoherent digital AM/FM demodulation, noise filters, AGC (for AM-ACSK only), coherent ACSK demodulation with error-feedback chaotic synchronization and symbol timing recovery.

The implementation process involves the use of the very high speed integrated circuit hardware description language (VHDL) for prototyping on Altera Cyclone V FPGA with a clock frequency of 50 MHz. This process encompasses the following auxiliary modules, which are necessary for the performance evaluation tests:

- an LFSR is utilized for the purpose of generating pseudorandom transmitted data;
- the AWGN controller is a real-time system that adjusts noise levels;
- the BER controller is utilized for the purpose of detecting and counting bit errors;
- the BER-to-DMA module serves the purpose of transmitting BER data to the memory;
- the software program has been developed in the C programming language for the purpose of reading BER data from memory and displaying the results;
- DAC and ADC peripheral modules facilitate analog channel link interfacing between transmitter and receiver.

The performance of the system was evaluated via bit error rate (BER) in additive white Gaussian noise (AWGN) (and selective fading for FM-ACSK) channels over a range of signal-to-noise ratio (SNR). The emphasis was placed on noise immunity, synchronization reliability, simulation-hardware gaps (fixed-point number format and real-world effects), chaotic communication challenges, and practical viability.

The Theses to be Defended

1. A discrete **AM-ACSK** transceiver based on a modified Chua oscillator employing the forward Euler discrete ODE solving method and error linear feedback chaotic synchronization **is implementable on a single Altera/Intel Cyclone V FPGA** with the data rate of 6.1 kbps at **430 kHz channel bandwidth**.
2. A discrete **FM-ACSK** transceiver based on a modified Chua oscillator employing the forward Euler discrete ODE solving method and error linear feedback chaotic synchronization **is implementable on a single Altera/Intel Cyclone V FPGA** with the data rate of 6.1 kbps at **1.720 MHz channel bandwidth** and **FM modulation index 3.**
3. The discrete **AM-ACSK** transceiver based on a modified Chua oscillator employing the forward Euler discrete ODE solving method, error linear feedback chaotic synchronization and implemented on a single Altera/Intel Cyclone V FPGA **achieves AWGN performance of 10^{-3} BER at 8.7 dB SNR** with an uncoded data rate of 6.1 kbps at **430 kHz channel bandwidth**.
4. The discrete **FM-ACSK** transceiver based on a modified Chua oscillator employing the forward Euler discrete ODE solving method, error linear feedback chaotic synchronization and implemented on a single Altera/Intel Cyclone V FPGA **achieves AWGN performance of 10^{-3} BER at 5.6 dB SNR** with an uncoded data rate of 6.1 kbps at **1.720 MHz channel bandwidth** and **FM modulation index 3**.

Research Methodology

In order to achieve the established goals and tasks, the research methodology incorporates a multi-stage approach integrating theoretical analysis, modeling, simulation, hardware implementation, and empirical validation. This structured progression ensures a comprehensive exploration of chaotic communication systems, from conceptual design to practical deployment on FPGA platforms.

The foundation of the research begins with analytic methods, including a literature review and state-of-the-art analysis on chaotic communication systems. This involves examining historical contexts, properties of chaos, chaos oscillators, synchronization techniques, modulation schemes, and FPGA implementations as alternatives to analog circuits. The review emphasizes FPGA-based chaos generators and communication systems, identifying challenges (e.g., synchronization reliability, noise vulnerability) and metrics (e.g., ordinary differential equation (ODE) discretization, clock speeds, bit precision, resource use), drawing from scholarly sources to support the novelty of the proposed amplitude-modulated antipodal chaos shift keying (AM-ACSK) and frequency-modulated antipodal chaos shift keying (FM-ACSK) schemes.

Utilizing this analysis as a foundation, mathematical modeling is employed to design the AM-ACSK and FM-ACSK systems. ODEs for the modified Chua chaos generator are formulated and discretized using the forward Euler method with fixed-point arithmetic to ensure chaotic behavior suitable for digital implementation. The models under consideration incorporate error feedback synchronization, antipodal chaos shift keying (ACSK) modulation, and digital amplitude modulation (AM)/frequency modulation (FM) passband techniques via numerically controlled oscillators (NCOs). Additionally, receiver signal processing modules are employed, including filtering, timing recovery using the early-late gate method, and bit detection via integration.

Numerical methods and simulations are employed to verify the proof-of-concept and assess system capabilities. Simulink models are designed to replicate the digital components of FPGA prototypes, thereby enabling performance evaluations in simulated channels. A BER analysis was conducted in an AWGN channel to compare AM-ACSK and FM-ACSK. The results demonstrated that FM-ACSK exhibits superior noise immunity. FM-ACSK's performance was further evaluated in a selective fading model.

The subsequent stage involves practical experiments, with a focus on FPGA prototype development and empirical verification. The implementation of the systems is achieved through the utilization of VHDL code compilation, facilitated by Quartus software for Altera Cyclone V FPGA hardware. This implementation incorporates transmitter and receiver signal processing chains, in addition to auxiliary modules that are employed for experimental setup, including AWGN noise controller, BER measurement tools, and reading software. Operational trials entail the implementation of hardware-in-the-loop testing, wherein digital-to-analog converter (DAC) and analog-to-digital converter (ADC) are connected via coaxial cable to establish an analog channel with 10.7 MHz passband frequency. BER measurements are conducted for various AWGN levels and selective fad-

ing scenarios, thereby validating simulation results and confirming the prototypes' viability. This methodological approach guarantees progression from theory to practice, balancing innovation with rigorous verification.

The Object of the Research

The object of the present research is a discrete chaotic communication system. Two such novel systems have been designed with AM-ACSK and FM-ACSK modulations. The research focuses on the FPGA implementation and performance evaluation of these systems.

Practical Significance

The practical importance of this research lies in the advancement of chaotic communication technologies through the development and validation of novel AM-ACSK and FM-ACSK FPGA-implemented prototypes, addressing real-world challenges in physical-layer signal protection and noise resilience. It is noteworthy that the implementation of discrete chaotic communication systems using FPGA is a subject that has received significantly less attention than other areas of communication system research, as demonstrated by a literature analysis. Complementary Simulink models facilitate rapid design and verification, while auxiliary experimental setup modules for the FPGA prototypes support empirical testing of the system noise immunity. These advancements serve as a practical guide for building and FPGA-implementing coherent chaotic communication systems, with several approaches, algorithms, and performance metrics to aid the design process.

Potential applications include:

- secure IoT networks, where the use of noise-like chaotic signals can enhance privacy and anti-jamming in resource-constrained devices;
- wireless military communications, offering covert transmission possibility and relative robustness to interference;
- spread-spectrum systems in the healthcare and transportation infrastructure sectors, which would enhance data integrity and protect against unauthorized access;
- commercial FPGA-based modems, which would enable integration into standards-compliant devices that have low data rate demands;
- smart home local sensor networks or wide area sensor networks, where such infrastructure might benefit from enhanced data protection options.

The transceiver prototypes, developed through iterative simulations and hardware trials, demonstrate the conceptual feasibility of the FPGA implementation, which can be used for further investigation of chaotic communication systems. The proposed design can be adapted to meet the specific application requirements in terms of data rates, spectrum bandwidth, resource and energy usage, and other relevant parameters.

Approbation

The following papers are relevant to the Thesis and have been published in scientific journals (the most relevant ones that reflect the main ideas of the research are marked in bold).

1. **F. Capligins et al. "Frequency-Modulated Antipodal Chaos Shift Keying Chaotic Communication on Field Program Gate Array: Prototype Design and Performance Insights." en. In: *Applied Sciences* 15.3 (Jan. 2025), p. 1156. ISSN: 2076-3417. DOI: 10.3390/app15031156**
2. M. F. Taşdemir et al. "Performance Evaluation of FPGA-Based Design of Modified Chua Oscillator." en. In: *Chaos Theory and Applications* 6.4 (Nov. 2024), pp. 249–256. ISSN: 2687-4539. DOI: 10.51537/chaos.1466919
3. D. Cirjulina et al. "Experimental Study on Colpitts Chaotic Oscillator-Based Communication System Application for the Internet of Things." en. In: *Applied Sciences* 14.3 (Jan. 2024), p. 1180. ISSN: 2076-3417. DOI: 10.3390/app14031180
4. R. Babajans et al. "Synchronization of Analog-Discrete Chaotic Systems for Wireless Sensor Network Design." en. In: *Applied Sciences* 14.2 (Jan. 2024), p. 915. ISSN: 2076-3417. DOI: 10.3390/app14020915
5. R. Babajans et al. "Performance Analysis of Vilnius Chaos Oscillator-Based Digital Data Transmission Systems for IoT." en. In: *Electronics* 12.3 (Jan. 2023), p. 709. ISSN: 2079-9292. DOI: 10.3390/electronics12030709
6. **F. Capligins et al. "FPGA-Based Antipodal Chaotic Shift Keying Communication System." en. In: *Electronics* 11.12 (June 2022), p. 1870. ISSN: 2079-9292. DOI: 10.3390/electronics11121870**
7. F. Capligins et al. "Experimental Study of the Chaotic Jerk Circuit Application for Chaos Shift Keying." In: *Latvian Journal of Physics and Technical Sciences* 58.4 (2021), pp. 55–68. DOI: 10.2478/lpts-2021-0033

The following papers are relevant to the Thesis and have been presented in scientific conferences and published in proceedings (the most relevant ones that reflect the main ideas of the research are marked in bold).

1. **F. Capligins, A. Litvinenko, and D. Kolosovs. "Selective Fading Channel Performance Evaluation for Frequency-Modulated Antipodal Chaos Shift Keying Communication System." In: *2025 IEEE 12th Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE)*. 2025, pp. 1–5. DOI: 10.1109/AIEEE66149.2025.11050770**
2. D. Cirjulina et al. "Fundamental Frequency Impact on Vilnius Chaos Oscillator Dynamics." In: *16th Chaotic Modeling and Simulation International Conference*. Ed. by C. H. Skiadas and Y. Dimotikalis. Cham: Springer Nature Switzerland, 2024, pp. 87–101. ISBN: 978-3-031-60907-7. DOI: 10.1007/978-3-031-60907-7_8
3. R. Babajans et al. "Study on Analog and Discrete Chaos Oscillators Synchronization." In: *16th Chaotic Modeling and Simulation International Conference*. Ed. by C. H. Skiadas and Y. Dimotikalis. Cham: Springer Nature Switzerland, 2024, pp. 33–46. ISBN: 978-3-031-60907-7. DOI: 10.1007/978-3-031-60907-7_4
4. **F. Capligins, A. Litvinenko, and D. Kolosovs. "Performance Evaluation of Frequency Modulated**

Antipodal Chaos Shift Keying Digital Communication System." en. In: *2023 Workshop on Microwave Theory and Technology in Wireless Communications (MTTW)*. Riga, Latvia: IEEE, Oct. 2023, pp. 29–33. ISBN: 9798350393491. DOI: 10.1109/MTTW59774.2023.10320019

5. A. Aboltins et al. "Implementation of chaotic frequency modulation based spread spectrum communication system in software-defined radio." In: *2023 IEEE Wireless Communications and Networking Conference (WCNC)*. 2023, pp. 1–6. DOI: 10.1109/WCNC55385.2023.10118612
6. F. Capligins, A. Litvinenko, and D. Kolosovs. "FPGA Implementation and Study of Antipodal Chaos Shift Keying Communication System." In: *2021 IEEE Microwave Theory and Techniques in Wireless Communications (MTTW)* (2021), pp. 1–6. DOI: 10.1109/mttw53539.2021.9607226
7. F. Capligins et al. "FPGA Implementation and Study of Synchronization of Modified Chua's Circuit-Based Chaotic Oscillator for High-Speed Secure Communications." In: 2021. ISBN: 978-1-66542-538-4. DOI: 10.1109/AIEEE51419.2021.9435783

Structure of the Thesis

The present Thesis is structured into five main sections, followed by overall conclusions.

Section 1 provides an overview of chaos applications in communication systems, covering historical background, properties of chaos, oscillators, synchronization methods, modulation schemes (non-coherent and coherent), advantages of FPGA over analog implementations, a review of existing FPGA-based chaotic systems, key challenges, and an introduction to the AM-ACSK/FM-ACSK research.

Section 2 provides a comprehensive overview of the AM-ACSK system, including its mathematical model, FPGA implementation considerations, transmitter/receiver signal processing, and Simulink model setup.

Section 3 elaborates on the FM-ACSK system, emphasizing architectural modifications from AM-ACSK, design principles, and FPGA implementation.

Section 4 evaluates the performance of both systems via Simulink simulations, by reviewing BER results for AM-ACSK (AWGN) and FM-ACSK (AWGN and fading).

Section 5 assesses the performance of FPGA prototypes, including experimental setups with auxiliary modules, BER analyses in AWGN (both systems) and selective fading (FM-ACSK).

Finally, the conclusions offer a summary of the key findings, innovations, and implications for chaotic communications.

1 CHAOS APPLICATIONS IN COMMUNICATION SYSTEMS

This section provides an overview of using chaos in communication systems and offers some general examples. It also explores the state of the art for FPGA implementation of chaos generators and chaotic communication systems.

1.1 General Context and Historical Background

Chaos manifests as aperiodic long-term behavior in a deterministic system, exhibiting a sensitive dependence on the initial conditions [3]. The use of chaotic oscillations in communication systems is not a novel technique, and it possesses certain advantages. A chaotic signal is difficult to intercept, and it has good resistance to multipath propagation and purposefully created interference [2].

Chaotic systems possess certain characteristics that make them particularly well-suited for use in communication systems.

- **Reduced energy utilization.** Circuits capable of chaotic signal generation can function efficiently with minimal power consumption.
- **Noise-like appearance.** The output signal of a chaotic system is aperiodic and, in certain cases, can be indistinguishable from noise, which complicates the detection of a chaotic signal.
- **Broadband spectrum.** The autocorrelation function of a chaotic system demonstrates a rapid decrease in value as the time interval increases. This property is intrinsic to the nature of a broadband spectrum signal. Furthermore, the cross-correlation function for chaotic signals of the same system from different initial conditions is found to be minimal due to the system's sensitive dependence on the initial conditions. These properties render chaotic generators highly suitable for implementation in spread spectrum communication systems.
- **Self-synchronization.** This property, which will be explained further, allows the receiver to synchronize with the transmitter's state. This can enhance the communication system's resilience to signal interception by utilizing coherent detection techniques.

Chaotic signals, characterized by their inherent wide-band nature, are well-suited for the propagation of narrowband information. Consequently, the utilization of chaotic signals for information encoding results in the generation of spread-spectrum signals characterized by increased bandwidths and reduced power spectral densities. The advantages associated with spread-spectrum signals, including the complexity of uninformed detection, the mitigation of multipath fading, and anti-jamming capabilities, are all features that are inherent to this technology. Therefore, chaos emerges as a cost-effective and adaptable medium for spread-spectrum communications.

1.2 Properties of Chaos and Chaotic Communication Systems

Given the variation in quality, safety level, and application possibilities, communication systems can be developed for both analog and digital transmission based on chaotic generators. The study and utilization of such systems is driven by numerous factors, including the following advantages inherent to chaos-based communications.

Firstly, it is important to note that chaos and noise can be difficult to distinguish from one another. It is possible to generate a chaotic sequence that exhibits a spectrum, statistical characteristics, and autocorrelation function that are very similar to the corresponding parameters of a stochastic noise. This, in turn, facilitates the transmission of a specified chaotic signal, the presence of which is challenging to discern by an external observer. In such cases, covert transmission of messages becomes a viable option, which can be of significant importance, particularly in military applications.

Secondly, if chaotic synchronization is employed, specific hardware is required, with the right parameters, to be able to decode the transmitted message. This complicates the unauthorized interception of the message, increasing the level of transmission protection.

Thirdly, the potential for information obfuscation is a notable consideration. In the conventional communication systems, encryption protocols are utilized, where both the transmitter and the receiver maintain a shared secret key. This key is essential for the decoding of transmitted messages. In the context of chaotic communication systems, the transmitting system itself functions as a dynamic key. The receiver must be capable of synchronizing to the dynamics of the transmitter to successfully retrieve the message.

Finally, numerous chaotic communication systems exhibit notable robustness against interference caused by multipath propagation. The importance of transmission privacy and protection in communications infrastructure is critical, as unauthorized access to such systems could compromise public order and security.

1.3 Chaos Oscillators, Chaotic Synchronization and Modulation

This subsection provides a brief overview of chaos oscillators, chaotic synchronization techniques, and modulation methods employed in chaotic communication systems with both non-coherent and coherent detection.

1.3.1 Chaos Oscillators

Chaos generators are nonlinear dynamic systems that exhibit chaotic behavior at specific parameters and initial conditions. Such systems can be described by a mathematical model using differential equations of state variables, and such systems can also be realized in a physical model (mechanical, electronic, or other). Chaotic oscillations are a possibility in a nonlinear dynamic autonomous system, defined as one that functions without external excitation and contains a minimum of three

state variables [3]. A paradigmatic example of an electronic chaotic generator is the Chua circuit [21], a simple and widely studied autonomous chaotic nonlinear dynamical system for which both a physical and mathematical model are known. The concept was introduced in 1983 by Professor Leon Chua of the University of California.

1.3.2 Chaotic Synchronization Methods

A fundamental property of chaotic nonlinear systems is their high sensitivity to initial conditions. This sensitivity implies that two identical chaos generators, even with minor variations in starting conditions, will generate distinct chaotic outputs. However, when these generators are linked in a specific configuration, their state variables align along identical paths, resulting in chaotic synchronization. The outcome of this process is the alignment of the state variables of the two chaos generators, despite their initial conditions being distinct.

Synchronization is only possible when systems are coupled in a specific manner. Intensive research in the field of chaos communication systems commenced in 1990, when a robust solution for synchronization of chaos generators was proposed [22]. Theoretical and experimental evidence has demonstrated the feasibility of synchronizing two chaotic signals through the implementation of a slave-master configuration within the chaotic system [23].

1.3.3 Chaotic Modulation Scheme Types

Chaotic communication systems operate on principles analogous to spread-spectrum techniques, wherein the transmitter expands the information signal's bandwidth into a broader transmission range, and the receiver compresses it during detection and demodulation. Despite the increased intricacy of these systems relative to their narrowband counterparts, they exhibit superior tolerance to multipath effects and allow for efficient multi-user frequency sharing. A critical constraint that must be addressed is the necessity for precise synchronization between identical chaotic oscillators in the transmitter and receiver. This synchronization requirement necessitates a high degree of precision to ensure optimal performance.

In essence, these systems can be categorized into two distinct classes: coherent and non-coherent. Each of these categories exhibits its own unique set of trade-offs [24]. Coherent systems, exemplified by chaos shift keying (CSK) [25], leverage chaotic synchronization to replicate the transmitter's oscillator state at the receiver, enhancing data safety through this master-slave coupling. However, this is achieved at the expense of reduced noise immunity due to sensitivity to parameter mismatches and channel distortions. Conversely, non-coherent systems, exemplified by differential chaos shift keying (DCSK) [26], manipulate chaotic signal attributes without synchronization, thereby simplifying receiver architecture and enhancing robustness to noise. However, this approach compromises physical-layer signal protection. In practice, non-coherent variants predominate, owing to their straightforward deployment and adequate noise resistance for applications

where advanced signal protection is not crucial. Both types exploit the nonlinear, broadband nature of chaotic signals for effective modulation.

1.4 FPGA Implementation as an Alternative to Analog Implementation

Chaotic oscillators have traditionally been realized through analog electronic circuits, such as the Chua [21], [27] or Rössler [28] designs, as well as variants like the Sprott [13], [29] circuit or Colpitts oscillator [9], [30], incorporating elements like resistors, capacitors, inductors, amplifiers, transistors, and diodes to sustain nonlinear oscillations in a chaotic regime [15]. These systems can be modeled with straightforward ODEs that capture their dynamic behavior. However, deploying analog chaos generators in communication applications presents significant hurdles, including susceptibility to parameter drifts from environmental influences like temperature, humidity, or power supply variations, which cannot be readily corrected during runtime. Such instabilities hinder reliable chaotic synchronization, which is essential for aligning identical master and slave oscillators. Analog circuits also lack control over generators' initial conditions [24].

To overcome these limitations, some researchers have shifted toward discrete digital implementations, particularly using FPGAs, which enable the numerical solution of ODEs to approximate chaotic dynamics in a deterministic manner [10], [16]. This approach eliminates drift issues, supports high-speed parallel processing via pipelining, and facilitates flexible prototyping with hardware description languages (HDLs) [5], [6]. FPGAs thus offer substantial advantages for chaotic communication systems, including enhanced computational throughput and adaptability to standards. Nevertheless, challenges persist in FPGA designs, such as optimizing resource usage, ensuring stability at elevated clock frequencies, selecting appropriate data bus widths in fixed- or floating-point formats, and choosing ODE solving methods, all of which impact signal accuracy, robustness, and overall system efficacy. As FPGA platforms grow more accessible, they revitalize interest in chaotic systems by enabling efficient, reliable prototypes that balance performance with practical constraints.

1.5 Review of the Chaotic System's FPGA Implementations

In contemporary settings, the necessity for advanced communication solutions with increased signal protection across diverse sectors remains on the rise. Nevertheless, a review of various scholarly publications on FPGA-integrated chaos generators and associated communication frameworks (outlined below) indicates that such studies are comparatively scarce relative to chaotic systems relying on alternative analog or digital platforms [24]. This scarcity of research focusing on FPGA in the context of chaotic communications may be attributable to the technology's recent evolution and the complexity of low-level programming, which is crucial for FPGA configuration. Consequently, there is a pressing need to further explore FPGA-based chaotic communication systems to elucidate their strengths, limitations, and viable uses.

Table 1.1 presents an overview of extant research on chaotic communication systems implemented via FPGAs. This development marks a shift from research into chaos generation and synchronization techniques to the design and verification of practical communication systems.

Table 1.1

Overview of FPGA-implemented chaotic communication systems

Paper	**Chaos generator**	**Chaotic synchronization method**	**Modulation**	**FPGA prototype max clock frequency, MHz**	**ODE discrete solving method**	**Max throughput, Gbps**	**Bit width of a system (integers with sign + fraction)**
[31]	modified Chua	Pecora-Caroll	authors', digital algorithm	not mentioned	forward Euler	not mentioned	32(not mentioned)
[23]	Lorenz	author's	chaotic masking	not mentioned	not mentioned	not mentioned	27(not mentioned)
[32]	authors'	Hamilton forms	chaotic masking	275.71	forward Euler	5.24	19(15 + 4)
[33]	author's and Sprott	none	COOK	not mentioned	not mentioned	not mentioned	not mentioned
[34]	Rossler	Pecora-Caroll	chaotic masking	70.9	forward Euler	2.85	19(10 + 9)
[35]	authors'	Hamilton forms	chaotic masking	1	forward Euler	not mentioned	28(4 + 24)
[36]	Sprott	Pecora-Caroll, Hamilton forms, OPCL	chaotic masking	116.4	Forward Euler, trapezoidal, RK-4	3.26	28(7 + 21)
[37]	various	Hamilton forms	16-C-PSK	not mentioned	not mentioned	not mentioned	32(not mentioned)
[38]	van Wyk	none	DCSK-CDMA	not mentioned	RK-4	not mentioned	40(21 + 19)
[39]	authors'	not mentioned	chaotic masking	31.8	RK-4	not mentioned	32(8 + 24)
[40]	authors', nonautonomous digital	direct coupling	chaotic encryption	not mentioned	not applicable	not mentioned	not mentioned
[41]	RCO	error linear feedback	2^{24}-CPSK + CSS + WCDMA	200	not mentioned	not mentioned	not mentioned
[42]	Chen	PTCFZNN	chaotic masking	not mentioned	not mentioned	not mentioned	not mentioned
present	modified Chua	error linear feedback	AM-ACSK/ FM-ACSK	76	forward Euler	1.06	master:14(6 + 8), slave:17(9 + 8)

In comparison with the overviewed previous studies, the present study features both distinctive

and shared characteristics.

- This study incorporates a customized modified Chua generator and an error feedback synchronization method. The former is new and has not been found in the previous studies of other authors, while the latter is found in only one of the studies examined.
- Modulation employs two novel modulation schemes: AM-ACSK and FM-ACSK. These novel schemes are more sophisticated than, for example, chaotic masking, and have not been previously explored in reviewed FPGA systems.
- ODE resolution employs the forward Euler method to ensure adequate precision and stability.
- The maximum clock frequency that can be achieved is 76 MHz; however, the design operates at 50 MHz through the utilization of onboard resources.
- The bit widths in the fixed-point format used are 14 bits in the transmitter's master generator and 17 bits in the receiver's slave generators (including the sign bit), with 8 bits for the fraction.
- The maximum achievable throughput for the master generator is 1.06 Gbps; this capacity is constrained by clock and bit width. Potential enhancements to the system, such as structural redesign and pipelining, could potentially achieve higher performance, but would also incur greater complexity and resource costs. The current approach is sufficient for demonstrating the viability of the systems.

1.6 Challenges in Chaotic Communications

Significant challenges arise in the realm of communication systems based on chaos, particularly with regard to ensuring reliable synchronization, especially in coherent setups, and addressing discrepancies in parameters across transmitting and receiving ends. Another critical issue involves balancing the protective qualities of a chaotic channel against its vulnerability to interference, which must be carefully managed during system development. The predictability of chaotic patterns, in conjunction with the simplicity of non-coherent approaches, has the potential to compromise the effectiveness of protection measures. Conversely, increasing the complexity of chaotic mechanisms may reduce resilience to environmental disturbances and signal disruptions. The pursuit of innovative, more effective, and practical modulation strategies for chaos remains a crucial objective, necessary for elevating the recognition of chaotic communication techniques relative to conventional alternatives.

1.7 Introduction to Research

Despite the challenges previously mentioned, there is a strong sense of confidence in the potential of CSK to provide reliable chaotic communication. The adoption of FPGA platforms has enabled the development of chaos generators that are precisely matched to the desired specifications, with manageable initial states. This development has profoundly impacted the field, particularly in the context of coherent chaotic frameworks and CSK models. This advancement enables empirical assessments

of CSK systems in real-world contexts, thereby necessitating the conceptualization, construction, and evaluation of novel FPGA-based designs as a foundational milestone.

The AM-ACSK and FM-ACSK chaotic communication system transceiver prototypes, presented in this research, incorporate both digital and analog components. The digital segment, realized on the FPGA, incorporates the modulators and demodulators alongside units for processing the transmitted and received signals. Meanwhile, the link between transmitters and receivers occurs in the analog domain, facilitated by a DAC and an ADC.

The rationale underlying the AM-ACSK design is derived from the necessity to integrate the protective advantages of chaotic signals with the convenience of amplitude modulation for transmission on the carrier frequency, resulting in a streamlined hardware configuration that maintains pseudorandom characteristics for covert communication. This blended strategy overcomes the constraints present in conventional chaotic architectures by relocating the baseband ACSK waveform to an intermediate frequency range.

The FM-ACSK iteration builds upon this foundation through the incorporation of frequency modulation, thereby enhancing defenses against carrier shifts and channel irregularities while facilitating refined spectrum management and a uniform envelope [43]. This configuration underlies an unexplored combination of frequency strategies with chaotic signal characteristics, enhancing its applicability to radio-based applications. When implemented on FPGA, it employs rapid execution and adaptable reconfiguration, with a substantiated Simulink simulation of digital components enabling precise comparisons between hardware outcomes and simulated estimations.

1.8 Conclusions

This section was devoted to providing an overview of chaos applications in communication systems, including their historical background, fundamental properties, chaos oscillators, synchronization methods, modulation schemes, the role of FPGA as an alternative to analog implementations, a review of existing FPGA-based chaotic systems, key challenges, and an introduction to the research on novel chaotic communication prototypes. The key topics, including the aperiodic and broadband nature of chaotic signals, the examples of non-coherent and coherent modulation techniques, and the advantages of digital FPGA prototyping for reliability and performance, were examined to establish theoretical foundation for chaotic communications. The subsequent sections will address the design, implementation, and experimental verification of AM-ACSK and FM-ACSK systems.

2 DESCRIPTION OF THE AM-ACSK CHAOTIC COMMUNICATION SYSTEM

This section provides a comprehensive overview of the AM-ACSK chaotic communication system transceiver, which is a novel approach that integrates chaotic dynamics with amplitude modulation. The overview begins with the mathematical modeling of key components, including the chaos generator, chaotic synchronization mechanisms, and baseband modulation. This is followed by an examination of its digital implementation on an FPGA. Next, the discussion explores the signal processing chains employed by the transmitter and receiver, including modulation, demodulation, filtering, gain control, timing recovery, and bit detection. AM-ACSK system is partially based on the the author's ACSK chaotic communication system prototype design [12], [19].

Figure 2.1 depicts the simplified overall architecture of the AM-ACSK communication system. In the AM-ACSK transmitter, the transmitted binary data is first processed by the ACSK modulator, which converts the data into a chaotic waveform. Then, the signal passes through the AM modulator, which shifts the signal to the carrier frequency.

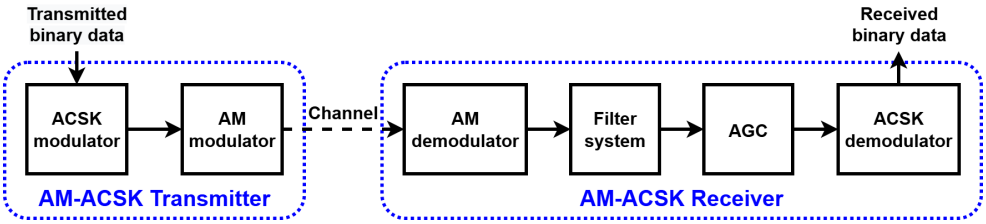


Fig. 2.1. Simplified overall architecture of the AM-ACSK communication system.

On the receiver side, the AM demodulator processes the signal, and a filter removes the channel noise and unwanted frequencies. An automatic gain control (AGC) module stabilizes the signal amplitude. Finally, the ACSK demodulator decodes the chaotic waveform, resulting in the received binary data.

2.1 Mathematical Model of the AM-ACSK Communication System

2.1.1 Modified Chua Chaos Generator

The designed communication system employs a fourth-order chaos generator based on a modified Chua's circuit [44], which is defined by the following set of ODEs:

$$\begin{cases} \frac{dp_1}{dt} = -g(p_1, p_3) - p_2 \\ \frac{dp_2}{dt} = p_1 + \gamma p_2 \\ \frac{dp_3}{dt} = \theta[g(p_1, p_3) - p_4] \\ \frac{dp_4}{dt} = \sigma p_3 \end{cases}. \tag{2.1}$$

The piecewise linear function is specified as

$$g(p_1, p_3) = \begin{cases} c(p_1 - p_3 - d) & (p_1 - p_3) > d \\ 0 & (p_1 - p_3) \leq d \end{cases}. \tag{2.2}$$

Here, p_m represents the system's state variables, and the parameters $\sigma = 1.5$, $\gamma = 0.5$, $c = 3$, and $d = 1$ are real scalar values that ensure the generator maintains stable chaotic behavior, as demonstrated in [20]. The chaotic output from the generator, $r_{out}(t)$, is derived as a weighted combination of the state variables and the piecewise linear function:

$$r_{out} = p_1 k_1 + p_2 k_2 + p_3 k_3 + p_4 k_4 + g(p_1, p_3), \tag{2.3}$$

where $k_1 = -2.6302$, $k_2 = -0.6054$, $k_3 = 0.5870$, and $k_4 = 0.7763$ are weight coefficients. These coefficients enable adjustment of the chaos generator's output signal parameters without altering the primary chaotic ODEs.

2.1.2 Error Feedback Chaotic Synchronization

The chaotic synchronization between the master and slave modified Chua chaos generators is accomplished using an error linear feedback technique, where the master generates a chaotic signal based on equation (2.3) by computing a weighted sum of its state variables p_m and the piecewise linear function $g(p_1, p_3)$. This signal, transmitted as r_{out}, serves as the input r_{in} to the slave generator, which lacks its own piecewise linear function and instead reconstructs $g'(p'_1, p'_3)$ by subtracting the weighted sum of its own state variables $p'_1 k_1 + p'_2 k_2 + p'_3 k_3 + p'_4 k_4$ from the incoming r_{in}, defined as

$$g'(p'_1, p'_3) = r_{in} - (p'_1 k_1 + p'_2 k_2 + p'_3 k_3 + p'_4 k_4), \tag{2.4}$$

enabling a simple yet effective feedback loop for the chaotic synchronization. The slave's output mirrors the master's structure but incorporates the absolute value $|g'(p'_1, p'_3)|$ due to the non-negative nature of $g(p_1, p_3)$, with its inversion occurring when an inverted $r_{out}(t)$ signal is received, causing $g'(p'_1, p'_3)$ to take negative or zero values, which the absolute value then corrects back to positive. This feature allows for the synchronization error signal e_{SYNC} to be derived by subtracting the slave's output from its input:

$$e_{SYNC} = p'_1 k_1 + p'_2 k_2 + p'_3 k_3 + p'_4 k_4 + |g'(p'_1, p'_3)| - r_{in}. \tag{2.5}$$

2.1.3 ACSK Baseband Modulation

Figure 2.2 illustrates the simplified structure of the ACSK [45] modulation and demodulation process over a baseband channel. This scheme employs a master chaos generator at the transmitter to

produce the initial chaotic signal, which is then fed into a conditional inverter. The inverter outputs either the original signal $r_{out}(t)$ or its inverted version, depending on the values of the transmitted data bits $b(t)$. Consequently, the transmitter switches between direct and inverted chaotic signals based on the bit value.

At the receiver, the incoming signal $r_{in}(t)$ is handled by two slave chaos generators: one processes the direct signal, while the other deals with an inverted copy. Chaotic synchronization occurs only at one of these generators, depending on the prior inversion state of the received signal. Error signals are generated for each slave by subtracting its output from its input, followed by smoothing via sliding-window average blocks. The resulting errors, $e_d(t)$ for the direct path and $e_i(t)$ for the inverted path, are subtracted from each other in order to compare the synchronization levels. This difference (Δ_{sync}) is then used in the decision-making unit to detect the values of the received data bits $b'(t)$, with the help of the symbol phase synchronization unit to provide timing recovery.

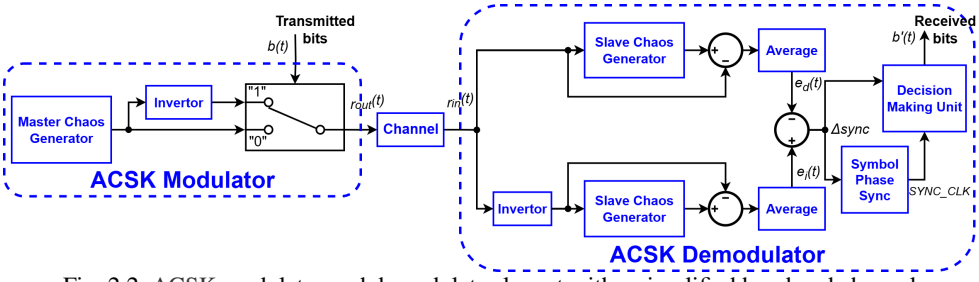


Fig. 2.2. ACSK modulator and demodulator layout with a simplified baseband channel.

The symbol duration spans 8192 samples (2^{13}), with each sample lasting 20 ns, corresponding to the 50 MHz clock oscillator period in the FPGA prototype. This same sampling time is used in the Simulink mathematical model.

Figure 2.3 demonstrates the principles that allow for bit detection by displaying the synchronization errors for both slave generators during transmission of the bit sequence "10101010," as well as their values' averaged difference (Δ_{sync}). Since the sign of Δ_{sync} correlates with the bit value, detecting the transmitted data essentially involves comparing Δ_{sync} with zero within the valid symbol timing frames.

2.1.4 Simulink Model of the AM-ACSK Communication System

Figure 2.4 illustrates the main units of the AM-ACSK communication system within the Simulink environment. All signals used in the model are in fixed-point format to match the digital implementation in FPGA. The signal processing modules generally come from the HDL Toolbox library in Simulink and are suitable for VHDL code generation. The model uses a discrete solver with no continuous states and a fixed sampling step of 20 ns, which matches the 50 MHz sampling frequency used in the FPGA design.

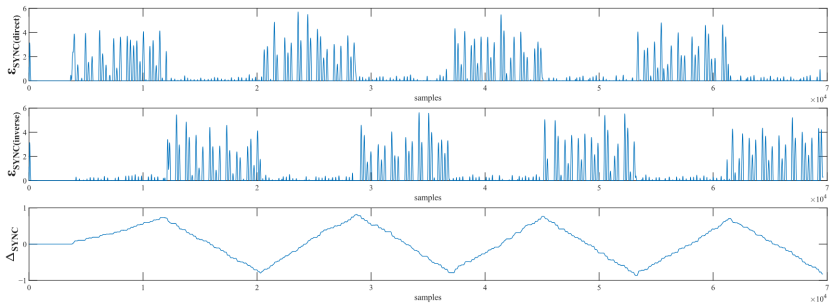


Fig. 2.3. Synchronization error traces (top to bottom): direct input slave generator, inverse input slave generator, and the difference of them averaged (Δ_{sync}) for "10101010" bit sequence transmission.

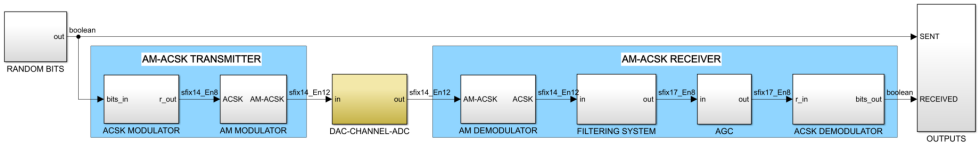


Fig. 2.4. Top-level view of the AM-ACSK system's Simulink model.

2.2 AM-ACSK Communication System Design for FPGA Implementation

This section details the FPGA-based realization of the AM-ACSK transceiver. The prototype setup incorporates a digital transmitter and receiver on an Altera Cyclone V 5CSXFC6D6F31C6N FPGA, driven by an onboard 50 MHz clock. The analog channel is implemented with a 62 cm coaxial cable connecting the AD9767 14-bit DAC and the AD9248 14-bit ADC, both mounted on the Terasic THDB-ADA_HSMC daughterboard, which interfaces with the Arrow SocKit development via high speed mezzanine card (HSMC). The discussion within this section focuses on the digital aspects of the system – specifically, modulation, demodulation, and signal processing.

2.2.1 Digital Implementation of Modified Chua Chaos Generator

To implement the continuous-time nonlinear chaotic system within an FPGA, which is designed primarily for discrete digital circuits, key decisions include selecting an appropriate numerical method for solving ODEs and adopting a suitable binary representation for the data of signals [20]. The forward Euler method is employed for ODE discrete solving, which offers simplicity, stability, convergence, and minimal consumption of FPGA hardware resources compared to more complex alternatives, such as fourth-order Runge-Kutta (RK-4). This method approximates solutions via the iterative formula $y_{n+1} = y_n + hf(y_n)$, where y_n denotes the current state, y_{n+1} – the subsequent approximation, h – the integration step, and $f(y_n)$ – the derivative function; accuracy improves as h diminishes toward zero. In this design, h is set to $(20 \times 50 \times 10^6)/1024$, balancing precision with ease of FPGA deployment, whereas a coarser step like $(20 \times 50 \times 10^6)/256$ leads to divergence due to inadequate approximation fidelity.

Applying this method to the chaotic system's ODEs (2.1) yields the discrete form:

$$\begin{cases} p_{1_{n+1}} = -g(p_{1_n}, p_{3_n}) - p_{2_n} \\ p_{2_{n+1}} = p_{1_n} + \gamma p_{2_n} \\ p_{3_{n+1}} = \theta[g(p_{1_n}, p_{3_n}) - p_{4_n}] \\ p_{4_{n+1}} = \sigma p_{3_n} \end{cases} . \tag{2.6}$$

For numerical handling in the transmitter, signals use a 14-bit signed two's complement fixed-point format, allocating 8 bits to the fractional portion and 5 to the integer and sign bits, enabling a range from -32 to roughly 31.996 with a resolution of about 0.0039 – sufficient for the chaotic attractor's bounds and aligned with the 14-bit constraints of the DAC and ADC interfaces. The receiver generally expands this to 17 bits (8 fractional, 9 integer/sign) for enhanced processing and a gap for additive noise, though signal bit-widths vary within the modules when necessary.

The generator's structure, illustrated in Fig. 2.5, comprises interconnected modular components: integrators (built from adders and registers following the Euler formula, with temporary bit-width expansion and output rounding for accuracy), a piecewise-linear unit $g(p_1, p_3)$ per its defining equation (using adders, a comparator, and a multiplexer without bit adjustments), and coefficient multiplications for θ, σ, and γ.

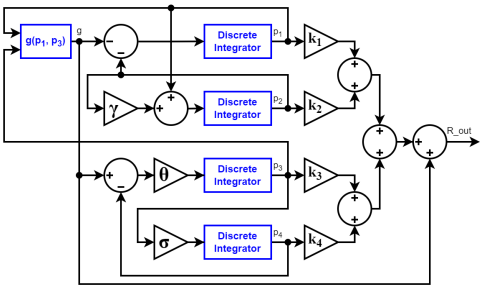


Fig. 2.5. General implementation design of the modified Chua chaos generator.

2.3 AM-ACSK Transmitter Signal Processing

As illustrated in Fig. 2.6, the signal processing in the AM-ACSK transmitter essentially consists of ACSK modulation followed by AM modulation. The process begins with the master chaos generator, which produces a 14-bit fixed-point baseband chaotic signal (r_{out}) that serves as the carrier for the ACSK modulation. This signal is then sent to the inversion controller, a critical module where binary data (inv_en) modulates the carrier through antipodal switching, a hallmark of ACSK. The ACSK-modulated output signal, r_{acsk}, is either inverted ($-r_{out}$) when $inv\_en = 1$ or passed unchanged (r_{out}) when $inv\_en = 0$ over a predetermined symbol duration of 8192 samples.

Next, the ACSK-modulated signal, r_{acsk}, is sent through an AM modulator. This modulator shifts the signal to a passband (r_{am}) in preparation for transmission. Following this, the most sig-

nificant bit (MSB) inversion module converts the signal to offset binary format (r_{dac}), shifting the AM-ACSK signal to positive unsigned values. This ensures compatibility with the DAC's input requirements. Finally, the processed digital signal is sent to the DAC, which converts it into an analog waveform for transmission.

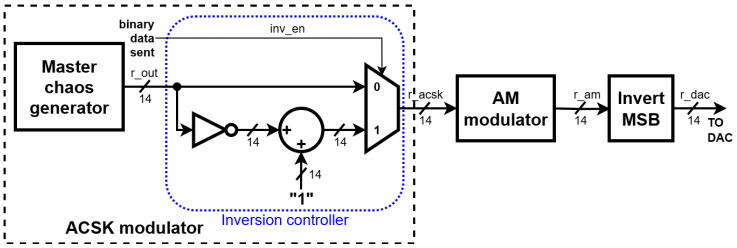


Fig. 2.6. Structure and some design elements of the AM-ACSK transmitter.

2.3.1 AM Modulator

Figure 2.7 shows the AM modulator design, implemented in Simulink for the FPGA. Since the ACSK-modulated signal (14-bit signed fixed-point with an 8-bit fraction) oscillates within maximum values of ± 7.055, it can fit compactly in a 12-bit signed fixed-point with an 8-bit fraction data bus. Using the XOR operation with a high MSB mask results in an inverted MSB value, which essentially adds a DC component of 8. Interpreting this shifted ACSK signal as unsigned causes it to oscillate between the values 15.055 and 0.945. It is then ready to be multiplied by the carrier.

The AM carrier is generated using an NCO, which is a block from the Simulink's digital signal processing (DSP) HDL Toolbox library that digitally synthesizes a sine wave with a frequency of 10.7 MHz using a compressed lookup table (LUT). The NCO employs 26 bits for the accumulator, 13 bits for phase quantization, and an additional 13 bits for phase dithering. These parameters ensure an output frequency resolution of approximately 1 Hz and a spurious-free dynamic range (SFDR) of roughly 90 dB. The carrier increment parameter, equivalent to 14,361,297, establishes the 10.7 MHz carrier output. It is calculated as follows:

$$\text{carrier increment} = \text{round}(F_0 \times 2^N \times dt), \tag{2.7}$$

where F_0 is the desired passband carrier frequency, N is the number of bits used in the accumulator, and dt is the sampling step (20 ns).

Although the NCO output is constrained to an amplitude of 1, shifting the binary point four bits to the right increases the maximum absolute value of the carrier signal from 1 to 16. This ensures that the AM modulation index ($15.055/16 = 0.9409375$) is relatively close to 1. This is a critical factor for the AM-ACSK noise resistance performance. It is important to note that only 215 kHz of the ACSK signal is utilized for demodulation, with the remainder being filtered out. Consequently, the bandwidth of the AM-ACSK signal is $215 \times 2 = 430$ kHz.

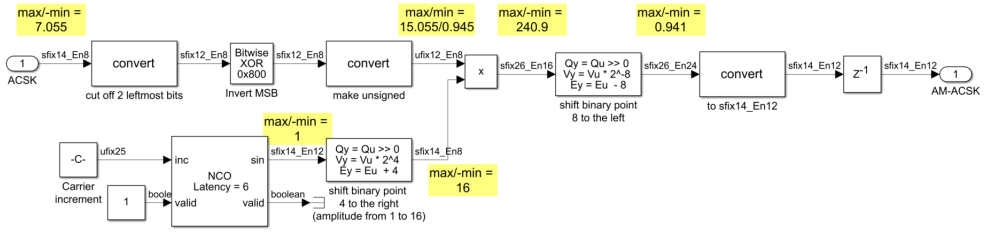


Fig. 2.7. Simulink diagram of the AM modulator design within the AM-ACSK transmitter.

2.4 AM-ACSK Receiver Signal Processing

The incoming signal requires preprocessing before the ACSK demodulator can handle it effectively. As shown in Fig. 2.8, the 14-bit straight offset binary (SOB) digital signal from the ADC (r_{adc}) undergoes an initial transformation in the MSB inversion unit. This shifts the AM-ACSK signal from positive unsigned values to a two's complement signed fixed-point format (r_h). Next, the AM demodulator unit restores the ACSK signal to baseband frequencies (r_l). Subsequent processing includes a filter system stage that removes unwanted frequencies and refines the signal (r_f). Then, an AGC unit adjusts the signal level (r_{in}). The resulting processed signal is fed into the ACSK demodulator, which extracts the transmitted binary data. The design of some of these relevant modules is explained further.

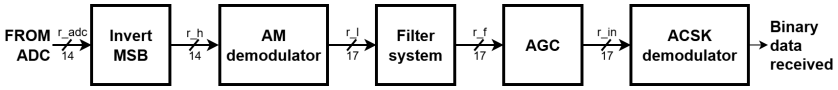


Fig. 2.8. Structure of the AM-ACSK receiver.

2.4.1 AM Demodulator

The AM demodulator in this communication system plays a pivotal role in recovering the baseband ACSK signal from the AM carrier. It is designed for simplicity and efficiency in hardware-constrained environments. Rather than relying on complex carrier phase recovery mechanisms, it uses an incoherent envelope detection approach. As depicted in Fig. 2.9, the first operation computes the absolute value of the input AM-ACSK signal. This effectively rectifies the modulated waveform and extracts its envelope. This step produces a unipolar signal that approximates the original ACSK modulation envelope, but it also introduces harmonic distortions and a DC offset. To mitigate the DC component, the DC Blocker module from the Simulink library is used, which is configured in cascaded integrator-comb (CIC) filter mode with the differential delay equal 8192 samples. CIC filters are well-suited for low-resource efficient FPGA implementation. This filtering centers the envelope around zero, restoring a bipolar representation that more closely resembles the original ACSK chaotic carrier signal, while adding only minor spectral distortion closely to 0 Hz.

It is important to note that AM demodulation is not finalized within this module alone. The absolute value operation generates unwanted high-frequency artifacts, including harmonics of the carrier

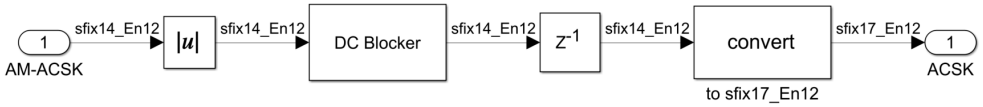


Fig. 2.9. Simulink diagram of the AM demodulator design within the AM-ACSK receiver.

and intermodulation products resulting from the chaotic signal's broadband nature. The subsequent filtering system addresses these spurious components by attenuating frequencies outside the desired baseband.

2.4.2 Filter System

To enhance the AM-ACSK communication system's resilience against channel noise and finalize AM demodulation, the filtering system attenuates signal components in non-critical frequency bands while preserving regions vital for chaotic synchronization, specifically those under 6 kHz and spanning 90 kHz to 215 kHz. Although a unified filter with dual passbands could theoretically achieve this, it would demand a high order and intricate design; thus, the approach employs a cascade of simpler, lower-order filters. The overall filter architecture integrates down-sampling, a bandstop stage, and up-sampling, utilizing just two core designs: a sixth-order infinite impulse response (IIR) halfband lowpass filter with a cutoff at 0.546 times the Nyquist rate, and a twelfth-order IIR bandstop filter. As illustrated in Fig. 2.10, the bandstop filter is applied iteratively six times, with a twofold down-sampling following each iteration, progressively reducing the sampling rate from 50 MHz to 781.25 kHz. On the final application of the halfband filter in this down-sampling phase, its effective cutoff reaches approximately 213.281 kHz, sufficiently approximating the target of 215 kHz.

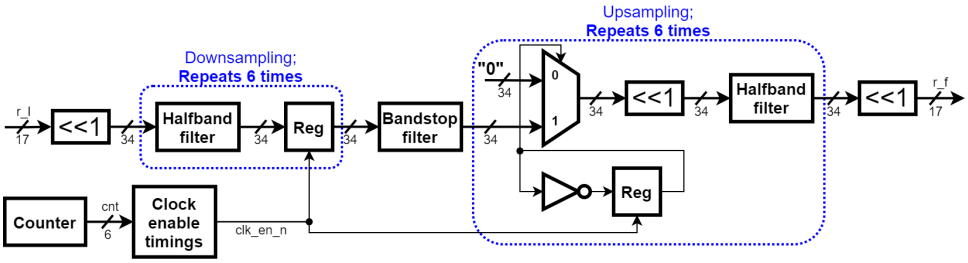


Fig. 2.10. Generalized digital implementation of the filtering system within AM-ACSK receiver.

This down-sampling strategy enables the use of a less complex bandstop filter to eliminate the 6–90 kHz range, which would otherwise be challenging at the full 50 MHz rate. Post-bandstop processing, the signal undergoes six stages of twofold up-sampling to restore the 50 MHz sampling frequency, essential for subsequent demodulation and synchronization. After each up-sampling step, the signal is scaled by a factor of 2 via a single left shift to offset baseband energy dilution, which arises from the spectrum splitting into baseband and half-sampling-frequency components; the extraneous higher-frequency elements are then suppressed by the halfband filter at each stage. Figure 2.10 depicts the generalized digital implementation, where filtering occurs at an expanded 34-bit

bus width. Variable sampling rates are managed through modulated clock enable signals for registers, all synchronized to a common clock: for instance, down-sampling by 2 involves enabling the initial stage register every second cycle, the next every fourth, and so on. Upsampling reverses this logic, incorporating a register with inverted feedback to drive a multiplexer that inserts zeros, effectively duplicating samples by alternating with null values at double the prior rate.

2.4.3 AGC

To compensate for variations in signal levels between the transmitted output and the received input, an AGC module is employed to dynamically scale the received signal toward a predefined reference power level. This ensures that the power delivered to the receiver's slave chaos generators aligns closely with that of the transmitter's master chaos generator, facilitating more reliable chaotic synchronization. The prototype incorporates feedback-based AGC architecture, as detailed in Fig. 2.11, where the input signal is first multiplied by a corrective gain factor derived from the feedback path. This factor is computed by determining the absolute value of the output signal through a combination of a comparator (to detect sign), an arithmetic inverter, and a multiplexer that chooses the inverted version for negative values or the original for positive ones. The resulting absolute value is then subtracted from a reference root mean square (RMS) value of 2.421, derived from the transmitter's modulated chaotic output, before being fed into a gain accumulator. Completing the loop, the accumulator's output undergoes division by 2^K via a logical right shift of K bits. The selection of K influences both the AGC system's dynamic range and response time; evaluations across K values from 10 to 17, with input powers ranging from 0.1 to 10 times the reference, led to K = 14 as a balanced performance option.

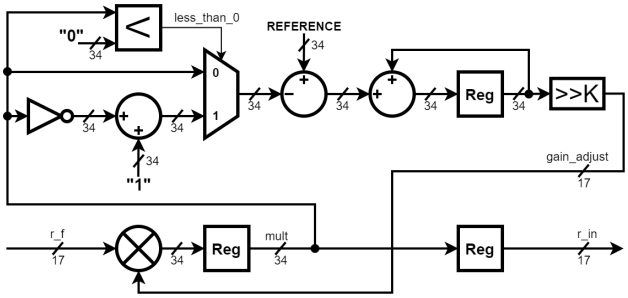


Fig. 2.11. Architecture of the automatic gain control system.

2.4.4 ACSK Demodulation

Current subsection delves into ACSK demodulator's digital architecture, as illustrated in Fig. 2.12. The incoming processed signal is split: one branch feeds directly into a slave chaos generator, whereas the other undergoes arithmetic inversion prior to entering a second slave chaos generator. These slave units largely replicate the master chaos generator's design, with key adap-

tations including a 17-bit signal bus throughout the receiver to handle elevated signal amplitudes during unsynchronized states; the omission of an internal piecewise linear function $g(p_1, p_3)$, which is instead recovered via a negative feedback loop; and the incorporation of the absolute value of this reconstructed function in the output signal.

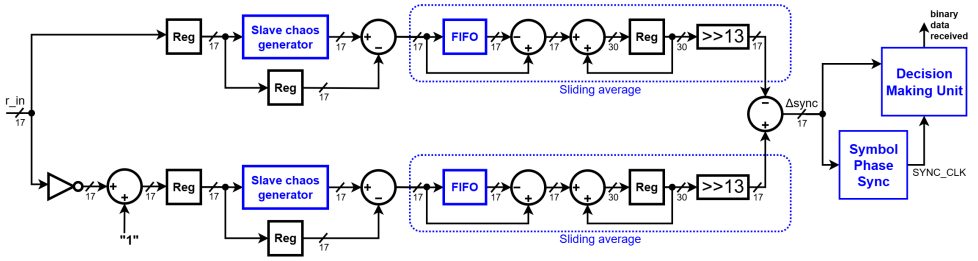


Fig. 2.12. Structure of the ACSK demodulator.

For each slave generator, the synchronization error is derived by subtracting the input from the output, followed by a sliding-window average computed over 8192 (2^{13}) samples. This averaging mechanism relies on a first input first output (FIFO) buffer capable of holding 8192 entries of 17-bit data, initially outputting zeros until full, then cycling through stored values from oldest to newest. The buffer's output is deducted from the current input and routed to an accumulator operating at 30-bit precision to accommodate the summation of 8192 terms without overflow, as $30 = 17 + 13$. The accumulated result is then scaled by dividing by 8192 through a 13-bit right shift, providing the mean synchronization error. The averaged errors from both slaves are differenced and evaluated in the decision-making unit to determine the transmitted bit value, with the process synchronized via a symbol phase synchronization module that recovers timing.

2.4.5 Timing Recovery

In real-world applications, the transmission path between the transmitter and receiver introduces unpredictable delays influenced by physical distance, leading to uncertainty in identifying the precise boundaries of each transmitted symbol (with one bit per symbol in this setup). This challenge is typically addressed through a timing error detector (TED), which analyzes the received signal to identify and correct phase discrepancies. This aligns the receiver's clock with the incoming data stream. Among prevalent TED techniques [46], decision-directed options like zero-crossing and Mueller-Muller depend on symbol estimates and prior data insights, whereas non-data-aided feedback approaches, including Gardner and early-late gate, operate independently of transmitted content knowledge, offering greater flexibility for symbol phase blind acquisition.

These standard methods are primarily intended for traditional digital modulation schemes. However, within this chaotic communication framework, the receiver uses a known symbol duration of 8192 samples at a 50 MHz clock rate, and the characteristics of the ACSK demodulator's Δ_{sync} output correlate with the bit values. This enables a straightforward implementation of blind symbol

phase synchronization via the adapted early-late gate technique.

Figure 2.13 illustrates the operational concept of the early-late gate TED applied to a sample Δ_{sync} signal. It employs two integration gates that sum the signal over the initial and latter halves relative to the assumed clock cycle, where the clock period matches the 8192-sample symbol without adjustments. When the assumed clock aligns perfectly with the symbol timing, the gate outputs differ minimally, signaling successful phase lock. Misalignments produce a significant disparity, with the difference's polarity indicating the direction of the offset (lead or lag) and its magnitude guiding the required correction. This error feeds into a loop that iteratively refines the clock phase until convergence. Post-lock, the system maintains tracking with minor tweaks throughout transmission. The illustration focuses on a "1" bit flanked by "0"s, but since absolute values are compared, it functions equivalently for other types of bit transitions as well.

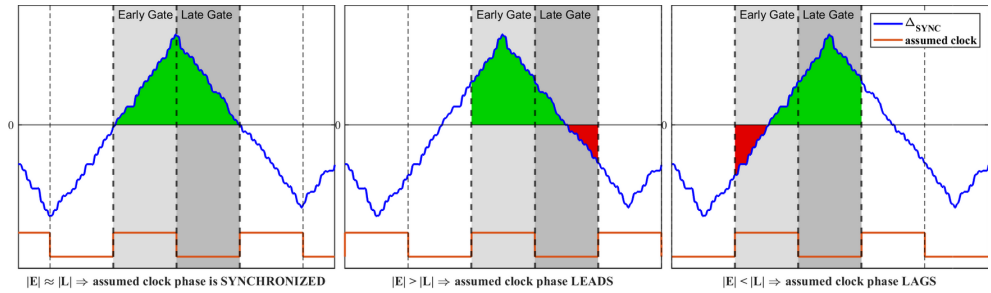


Fig. 2.13. Illustration of the early-late gate timing error detection approach, comparing accumulations in the early and late gates for the Δ_{sync} signal under different assumed clock phases, specifically for a scenario where a received "1" bit is flanked by "0" bits. The green regions highlight positive value accumulations, while red regions denote negative ones.

The developed design adapts a standard early-late gate synchronizer [47] to suit the AM-ACSK system's demands and FPGA constraints, starting with a fixed-point Simulink model converted to VHDL using HDL Coder. As shown in Fig. 2.14, processing commences by left-shifting the Δ_{sync} input ("err_delta_in") logically by 3 bits (equivalent to multiplying by $2^3 = 8$) to amplify variations, followed by separate integrations over the early and late halves, reset by the SYNC_CLK signal. Each integrator comprises an accumulator handling roughly 2^{12} samples (with slight variations from loop adjustments) and an arithmetic right shift by 12 to approximate division by 2^{12}.

A basic timing error estimate could be $|E| - |L|$ (with E as early gate output and L as late), but this limits utility to phase errors under 25% of symbol duration, yielding near-zero results for larger shifts near sign changes in Δ_{sync}. To cover full-range errors, the computation separates magnitude and direction:

$$\text{timing error} = |E - L| \times \text{sign}(|E| - |L|). \tag{2.8}$$

For practical adjustment, a refined version scales the shift and inverts the sign for feedback:

$$\text{PHASE\_ADJUST} = 2^6 \times |E - L| \times \text{sign}(|L| - |E|), \tag{2.9}$$

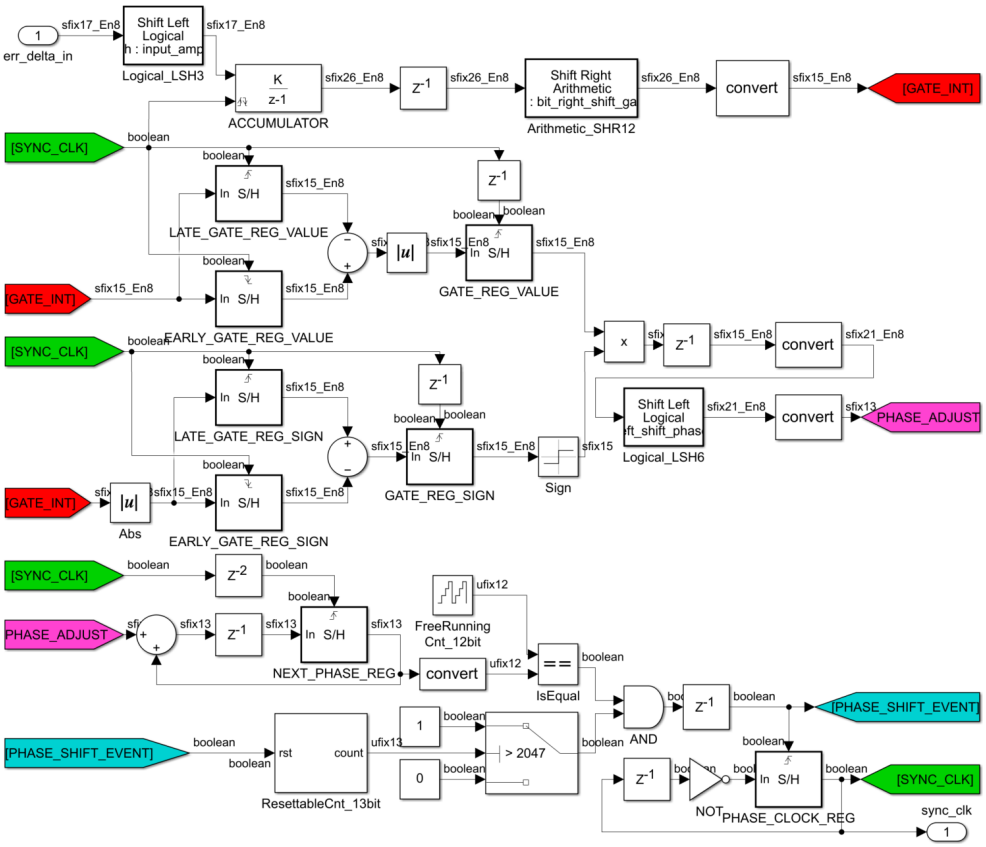


Fig. 2.14. Simulink diagram depicting the implemented early-late gate symbol phase synchronizer. Colored labels denote the signal links, and arrows on clock inputs specify the triggering clock edges.

where the 2^6 factor (via left shift by 6) optimizes loop dynamics. Empirical tests revealed it balances rapid convergence without overshoot or oscillations, with precision.

The PHASE_ADJUST value modulates a 12-bit free-running counter generating SYNC_CLK, while a 13-bit auxiliary counter ensures no premature clock toggles during forward jumps by enforcing at least 2048 samples per half-cycle. The resulting synchronized clock feeds the downstream bit detector module.

2.4.6 Decision-Making Unit (Bit Detector)

The decision-making unit uses the Δ_{sync} signal to determine the values of incoming bits, relying on the phase timing derived from the symbol phase synchronization module. As illustrated in Fig. 2.15, this involves integrating the Δ_{sync} input ("err_delta_in") over the full symbol duration, guided by the recovered bit clock ("sync_clk_in"). Precision in this integration is not critical, since the process solely examines the sign of the outcome for binary classification: a positive result yields a "1," while a negative one produces a "0."

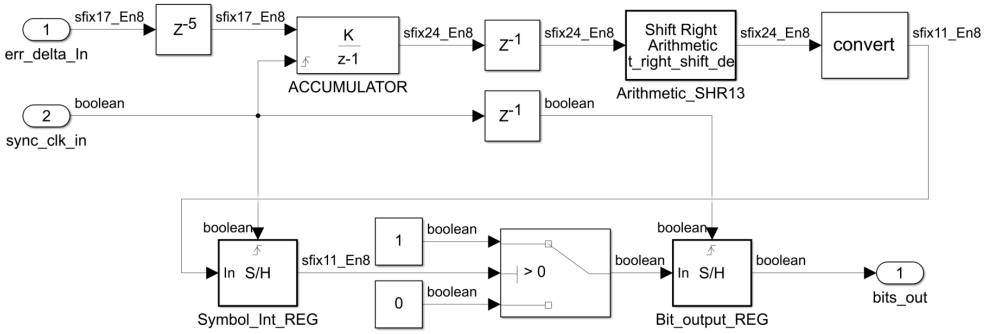


Fig. 2.15. Simulink diagram of the module for detecting received data bits.

2.5 Conclusions

This section was devoted to describing the AM-ACSK chaotic communication system transceiver, including its mathematical model, design principles, and digital implementation on FPGA hardware. The key components, such as the modified Chua chaos generator, error feedback synchronization, ACSK modulation, and signal processing chains for both transmitter and receiver, were examined in detail, with emphasis on their adaptation for efficient prototype development and verification through Simulink simulations. The FPGA implementation of AM-ACSK chaotic communication system transceiver described in this section confirms the first thesis stated in the introduction.

3 DESCRIPTION OF THE FM-ACSK CHAOTIC COMMUNICATION SYSTEM

The FM-ACSK communication system transceiver is very similar to the previously described AM-ACSK system, but it has undergone major design changes. As the name suggests, the main difference is that this system uses frequency modulation instead of amplitude modulation for passband transmission. This section explains the design changes relative to the AM-ACSK system, followed by a description of the digitally implemented FM modulator and demodulator units and the new lowpass filtering system. The design and performance insights of FM-ACSK were published in [7], [17].

3.1 Changes relative to the AM-ACSK Communication System

The overall architecture of the FM-ACSK communication system's transmitter and receiver is illustrated in Fig. 3.1. Since the FM-ACSK system is a variation of the previously described AM-ACSK communication system, its core ACSK modulator and demodulator units are identical to those of the AM-ACSK system. The differences lie in the signal processing parts of the transmitter output and receiver input.

- The **AM modulator and demodulator units are replaced with FM modulator and demodulator units**, respectively. Both systems use a 10.7 MHz carrier, but FM-ACSK uses a wider channel bandwidth (1720 kHz versus 430 kHz for AM-ACSK), which aims to increase the resistance to the channel noise.
- An **AGC unit is no longer necessary** before the ACSK demodulator, since the output signal power of the FM demodulator is independent of its input signal power.
- The **filter system has undergone a complete redesign**. The AM-ACSK employs a downsampling and upsampling approach with low-order halfband IIR filters, as well as a bandstop filter for the 6–90 kHz band. For the FM-ACSK, the 6–90 kHz bandstop filtering has been omitted for simplicity's sake, with filtering of only lowpass frequencies below 215 kHz using a specific combination of two finite impulse response (FIR) filters. An additional 860 kHz lowpass filtering unit is used within the FM demodulator.

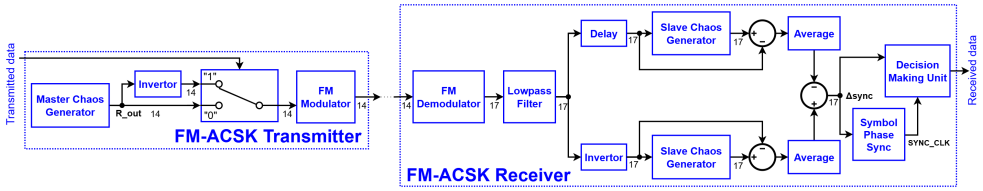


Fig. 3.1. Overall architecture of the FM-ACSK communication system's transmitter and receiver.

3.2 Frequency Modulation

The FM modulator is implemented digitally via an NCO (see Fig. 3.2). Initially, the ACSK signal is interpreted as a 14-bit signed integer, which essentially multiplies the value by the 2^8. This is done for the sake of design convenience, as the NCO expects integer values at the phase increment port input. The deviation increment is a scaling coefficient that, when multiplied by the modified ACSK and added or subtracted from the carrier increment value, ensures the desired deviation of 645 kHz. The NCO internal parameters and carrier increment value are the same as those described for the AM modulator in the previous section. As with the AM modulator, a passband frequency of 10.7 MHz is used, which is a common intermediate frequency in conventional FM communication systems. Given that the maximum ACSK frequency employed in the demodulation is 215 kHz, the deviation of 645 kHz was selected, thereby resulting in a frequency modulation index value of 3 and a 1720 kHz bandwidth for the FM-ACSK transmitted signal, in accordance with Carson's Rule.

The digital FM demodulator, as illustrated in Fig. 3.3, employs a 30-tap FIR Hilbert filter along with a corresponding delay – equivalent to half the Hilbert filter's group delay plus its inherent latency – to form a complex analytic signal from the noise-impacted FM-ACSK input. This analytic signal is then multiplied by a locally generated carrier signal from an NCO, mirroring the one in the FM modulator. This mixing operation shifts one copy of the FM-ACSK analytic signal to the

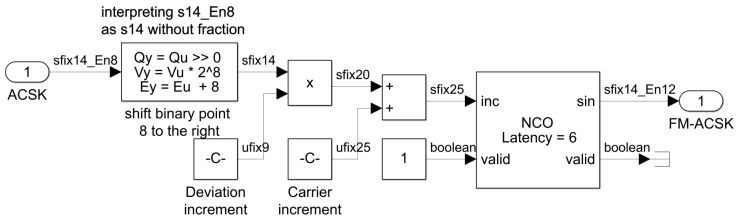


Fig. 3.2. Simulink diagram of the FM modulator unit.

baseband while moving the other copy to twice the carrier frequency. A 100-tap FIR lowpass filter with a cutoff frequency of 860 kHz, which represents half of the FM-ACSK bandwidth, is employed to suppress the high-frequency replica and mitigate a portion of the channel noise.

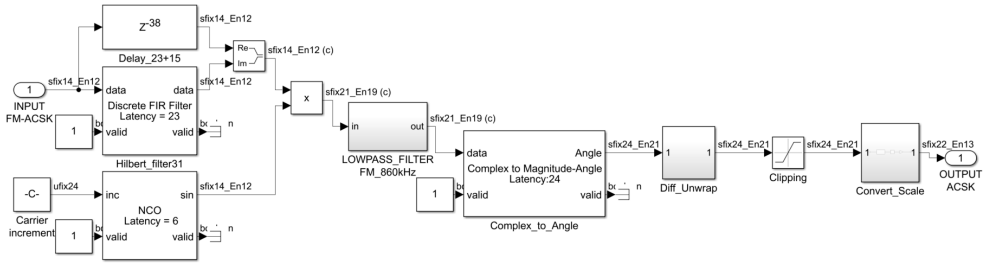


Fig. 3.3. Simulink diagram of the FM demodulator unit.

The instantaneous phase of the resulting complex baseband signal is extracted by computing its angle using a unit from the Simulink HDL Toolbox. Recovery of the modulating ACSK signal, which corresponds to the instantaneous frequency, is achieved through a straightforward two-tap differentiator, which also eliminates any constant offset stemming from carrier phase mismatches. To prevent overflows caused by intense noise, hard limiting is imposed on signal excursions beyond the anticipated range dictated by the frequency deviation. Finally, the signal is amplified to make the power level at the ACSK demodulator input similar to the level at the ACSK modulator output.

3.3 Filters

Prior to forwarding the reconstructed ACSK signal to the ACSK demodulator, supplementary filtering via a 215 kHz lowpass FIR filter is performed to further attenuate residual channel noise. In the FM-ACSK system, the process of filtering is executed by a cascaded arrangement of two lowpass FIR filters (see Fig. 3.4), designated as “RAISED” and “FIX” within the design framework. The objective of this approach is to achieve a satisfactory performance in FIR filtering while minimizing the utilization of hardware resources.

The RAISED FIR filter concept is based on the idea that a filter designed for a cutoff frequency closer to half the Nyquist rate (F_N) requires a lower order than a filter with the same constraints and cutoff slope would require at the edges of the processable frequency span (0 and

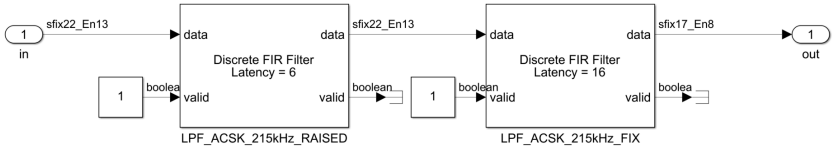


Fig. 3.4. Simulink diagram of the lowpass 215 kHz FIR filter general structure after FM demodulator.

$F_N = 25$ MHz, in this case). One existing approach to address this disadvantage is to employ digital downsampling and upsampling. However, RAISED FIR filter avoids that with a much less complex approach. The initial filter is designed at the higher cutoff frequency, specifically at the target cutoff frequency $F_c = 215$ kHz multiplied by an integer shift coefficient RAISE $= 7$. Subsequently, a RAISE $- 1 = 6$ zeros are appended between every coefficient of the initial filter, thereby providing the coefficients for the RAISED FIR filter. This adjustment effectively resets the cutoff frequency to F_c, while introducing multiple narrow bandpass regions surrounding frequencies $1 \times F_s$/RAISE, $2 \times F_s$/RAISE, $3 \times F_s$/RAISE, and so forth. The selected RAISE coefficient merely adjusts the initial cutoff frequency from 215 kHz to 1.505 MHz.

The FIX FIR filter is a simple low-order lowpass FIR filter whose stopband region begins at the first bandpass region of the RAISED FIR filter, which is located at a frequency near $1 \times F_s$/RAISE $- F_c$. This guarantees that all of the bandpass regions above baseband, introduced by the RAISED FIR filter, are negated to a level of at least the selected stopband attenuation of 60 dB.

The RAISED FIR filter of order 735 exhibits a baseband frequency cutoff slope that is identical to that of a regular FIR filter of the same order, both designed under identical constraints. The primary distinction lies in the fact that the RAISED FIR filter, among all 735 coefficients, has only 104 non-zero coefficients, and the additional passband regions are canceled with a relatively small FIX FIR filter of the order 17.

The frequency response of the entire lowpass 215 kHz FIR filter within the initial 500 kHz range is illustrated in Fig. 3.5. The dotted line indicates the idealized filter frequency response, exhibiting a 200 kHz over 60 dB slope from the F_c. The design constraint for the passband peak-to-peak ripple is set to approximately 0.2 dB to ensure minimum spectral distortion of the ACSK signal. The drawback of this constraint is the relatively slow transition between passband and stopband, where 3 dB attenuation is achieved only at 279 kHz.

The lowpass FIR filter in the FM demodulator unit for the cutoff frequency of 860 kHz is designed by the same approach as the 215 kHz FIR filter described above. The stopband attenuation and passband ripple constraints, as well as the RAISE coefficient, are identical. The number of non-zero coefficients in the 860 kHz RAISED FIR filter (of order 693) is 98, and the corresponding FIX FIR filter has an order of 27.

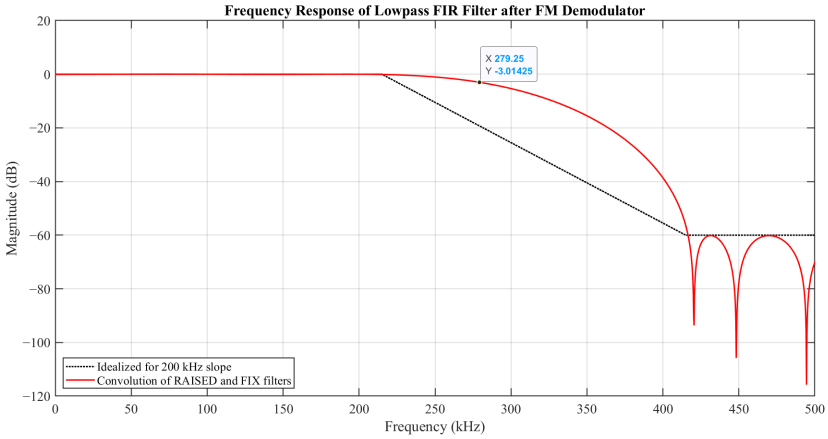


Fig. 3.5. Frequency response of the lowpass 215 kHz FIR filter in the first 500 kHz frequency span.

3.4 Conclusions

This section was devoted to describing the FM-ACSK chaotic communication system transceiver, including its architectural similarities and modifications relative to the AM-ACSK system, design principles, and digital implementation on FPGA hardware. The key components, such as the FM modulator and demodulator units, along with the redesigned lowpass filtering system, were examined in detail. The FPGA implementation of FM-ACSK chaotic communication system described in this section confirms the second thesis stated in the introduction.

4 PERFORMANCE EVALUATION OF THE CHAOTIC COMMUNICATION SYSTEM MODELS USING SIMULATION

To assess the performance of the AM-ACSK and FM-ACSK chaotic communication system transceivers in the Simulink environment, AWGN channel noise performance tests are conducted for both systems, with an additional selective fading channel noise performance test for FM-ACSK. This section describes the experimental setups and results of these simulation tests.

4.1 Simulink Model Experimental Setup

The present subsection focuses on describing the general structure of the Simulink model experimental setup for both systems, as illustrated in Fig. 4.1. Initially, input data bits are generated using a Bernoulli-distributed random binary generator unit from the Simulink library with a timing of one data bit per $T_b = 8192$ samples. A DAC-Channel-ADC unit serves to replicate the approximate

attenuation and delay characteristics present within the FPGA prototype, which is connected via a 62 cm coaxial cable with a 10-sample delay and 0.35 gain.

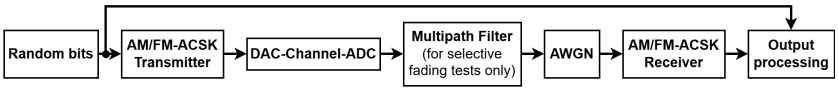


Fig. 4.1. Experimental setup general structure for Simulink performance evaluation tests for both chaotic communication systems.

4.1.1 Selective Fading Channel

The objective of this experimental test is to replicate a basic multipath scenario in a wireless link and assess the system's resistance to noise under various selective fading conditions. This basic model incorporates only two signal paths: a direct line-of-sight route and a single reflected path (as shown in Fig. 4.2). The chosen delays are fixed to create a time-invariant selective fading effect, with the resulting notch – arising from phase interference – aligned at the carrier frequency of the FM-ACSK signal, representing a challenging worst-case scenario.

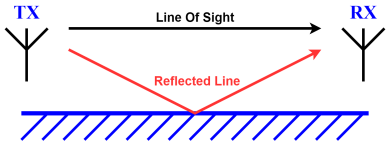


Fig. 4.2. Schematic representation of a two-path propagation model for a selective fading channel.

A straightforward FIR filter was implemented to emulate this two-path, time-invariant selective fading channel (as depicted in Fig. 4.3). The delay parameter τ denotes the time lag of the reflected signal relative to the direct one at the receiver. A delay of 21 samples was selected (with each sample lasting 20 ns at a 50 MHz sampling rate), which determines the fading notch's bandwidth and positions it near the carrier frequency (illustrated in Fig. 4.4). The gain parameter α accounts for losses due to reflection. In the experiment, six distinct gain settings were tested, yielding varying depths of notch attenuation. In order to maintain consistent signal power at the AWGN injection point – thus simplifying SNR computations – a gain adjustment is applied post-FIR filtering. The specific gain (α) and the output correction values are summarized in Table 4.1, with rounding applied to conform to the system's signed 14-bit fixed-point format (12 bits allocated to the fractional part).

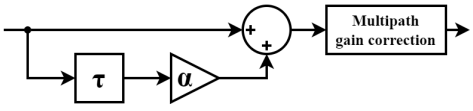


Fig. 4.3. Configuration of the multipath filter, featuring an FIR component for two-path selective fading and subsequent gain normalization.

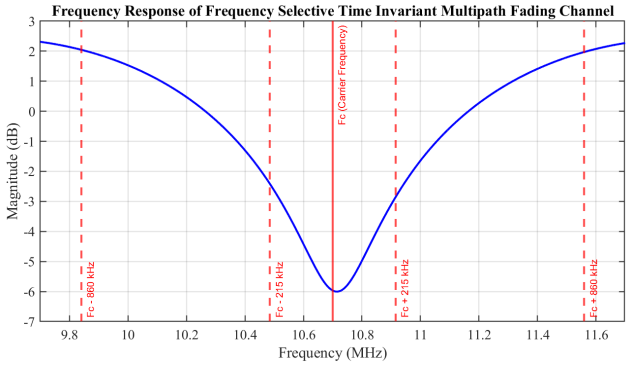


Fig. 4.4. Magnitude response of the selective fading FIR filter exhibiting a 6 dB notch depth.

Table 4.1

Parameters for the multipath filter

Notch attenuation depth, dB	Gain (α)	Multipath gain correction
1.5	0.2919921875	1.292724609375
3.0	0.498779296875	1.548583984375
4.5	0.700927734375	1.76123046875
6.0	0.7490234375	1.79052734375
7.5	0.822265625	1.811279296875
9.0	0.874267578125	1.806884765625

4.2 AM-ACSK Simulink Model AWGN Channel Performance Evaluation

As shown in Fig. 4.5, the simulated AM-ACSK system reaches a BER below 10^{-3} at approximately 7 dB SNR and continues to improve almost monotonically with increasing SNR, falling toward the 10^{-4}–10^{-5} region by roughly 9–10 dB. The simulated curve is smooth and nearly log-linear over the mid-to-high SNR range, indicating stable performance and predictable noise sensitivity for the idealized model.

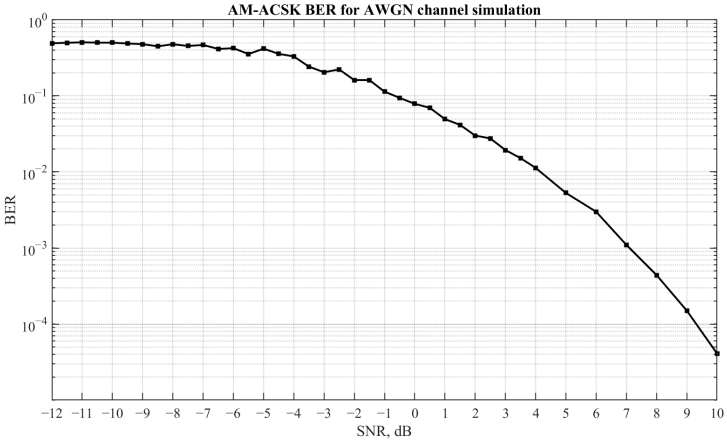


Fig. 4.5. BER for the Simulink simulation of the AM-ACSK system with an AWGN channel.

4.3 FM-ACSK Simulink Model Performance Evaluation

4.3.1 AWGN Channel Performance

The MATLAB Simulink simulation results for the FM-ACSK chaotic communication system, as depicted in Fig. 4.6, exhibit a smooth BER curve under AWGN channel conditions, showcasing strong noise resistance with BER dropping from approximately 0.5 at SNR values below -3 dB to near 10^{-4} at 5.6 dB. Simulations employed 2000 data bits transmitted for most points and 20,000 bits for the lowest four BER values to improve precision at higher SNR, where errors are rare. The curve remains flat around 0.5 BER from -7 dB to -3 dB, indicating a threshold effect, before declining steeply – e.g., from ~ 0.3 at -1 dB to ~ 0.001 at 5 dB – with a decelerating slope from over 2 dB per decade initially to under 1 dB per decade at elevated SNR, supporting accurate noise performance forecasting. Additional testing with denser SNR sampling and larger bit volumes might reveal minor BER slope variations, though the current data shows no such anomalies.

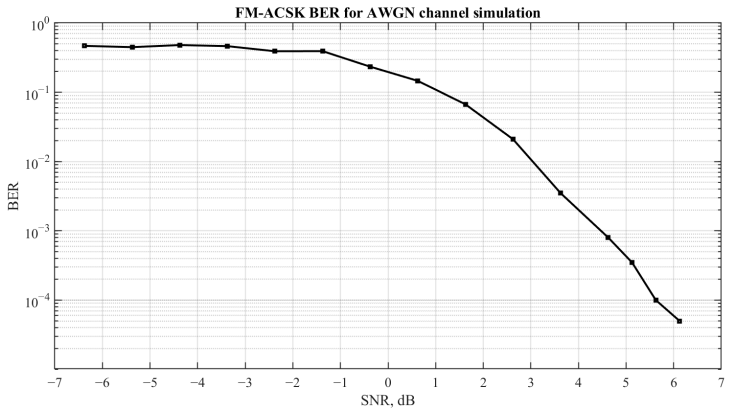


Fig. 4.6. BER for the Simulink simulation of the FM-ACSK system with an AWGN channel.

4.3.2 Selective Fading Channel Performance

Fig. 4.7 presents the BER simulation results for different depths of selective fading notches across various SNR levels, which are published in [14]. At elevated SNR values (exceeding 2–3 dB), the system's performance with shallow notches of 1.5 dB and 3.0 dB remains comparable to that in the absence of fading. However, deeper notches result in substantial BER degradation due to the heightened distortion in the FM-ACSK signal's spectrum.

Conversely, under conditions of high noise (SNR below 2–3 dB), the presence of selective fading yields marginally superior performance compared to non-fading scenario. This enhancement is attributable to the channel's amplification of critical sideband frequencies that carry information, positioned beyond approximately half the deviation from the carrier. While this effect is discernible in BER plots, its practical utility is constrained, as operating at BER levels above 10^{-2} poses significant

challenges, even with sophisticated forward error correction (FEC) techniques.

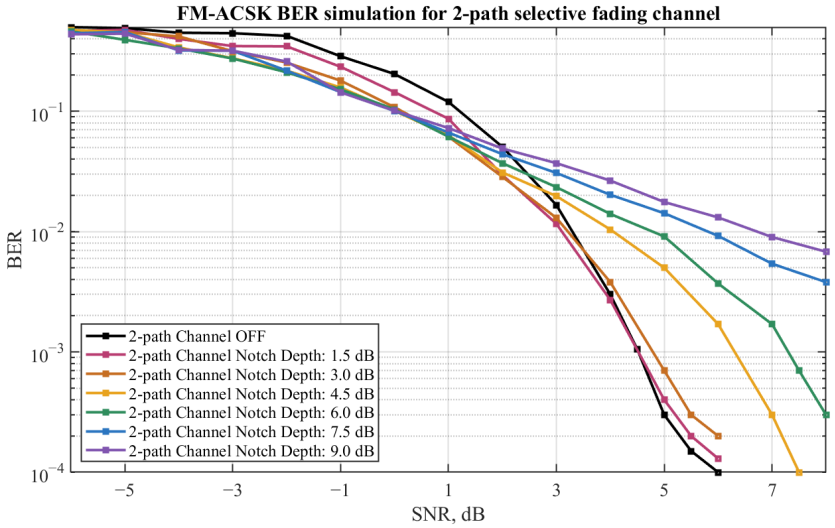


Fig. 4.7. BER curves for varying two-path selective fading notch depths at different SNR values in MATLAB Simulink simulation.

4.4 Conclusions

This section described the performance evaluation of the chaotic communication system mathematical models using the MATLAB Simulink environment. The simulation testing setup for both AM-ACSK and FM-ACSK system transceivers was described in detail, including the definition of the selective fading channel model, followed by BER analyses performed in AWGN (for both systems) and two-path selective fading (for FM-ACSK) channels across a wide range of SNR values.

5 PERFORMANCE EVALUATION OF THE CHAOTIC COMMUNICATION SYSTEM FPGA PROTOTYPES

In a manner consistent with the preceding section, this section examines the performance of AM-ACSK and FM-ACSK for AWGN channels in the absence and the presence of selective fading channels. These evaluations are conducted on the FPGA prototypes, which offer a more comprehensive understanding of the practical performance characteristics of the chaotic communication systems.

5.1 FPGA Prototype Experimental Setup

The FPGA prototypes were developed using the Arrow SoCKit board (see Fig. 5.1), which is equipped with an Altera (Intel) Cyclone V 5CSXFC6D6F31C6N FPGA chip. The establishment

of the transmitter-receiver link is facilitated by an analog channel, employing DAC and ADC on the Terasic THDB-ADA daughterboard. This configuration is connected to the SoCKit through a HSMC interface.

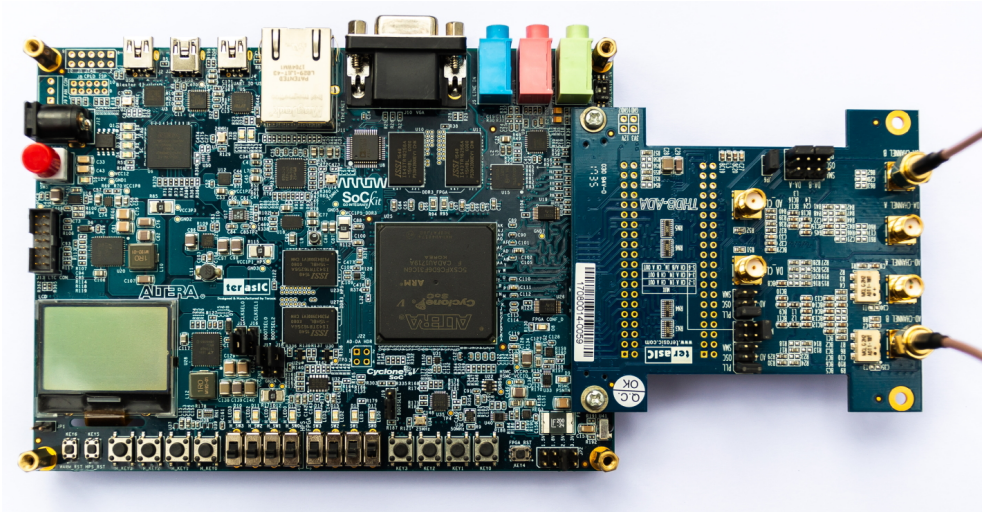


Fig. 5.1. The photograph of the FPGA chaotic communication systems prototype setup on an Arrow SoCKit development board with a Terasic THDB-ADA daughter board and a coaxial cable for analog signal transmission.

The Cyclone V system-on-chip (SoC) exhibits a range of advanced features that extend beyond the fundamental capabilities of FPGAs. It incorporates a hard processing system (HPS) that contains an Arm Cortex-A9 microprocessor unit (MPU) and peripheral controllers. This facilitates the seamless integration of FPGA signal processing with MPU-based software operating on a dedicated Linux platform. The Arrow SoCKit board facilitates the access of peripherals such as clocks, buttons, switches, light-emitting diodes (LEDs), synchronous dynamic random-access memory (SDRAM), and a USB-to-UART bridge, all of which are utilized in the test configuration.

Figure 5.2 provides a high-level view of the experimental setup. Green elements denote custom-designed FPGA modules, blue elements represent Altera/Intel intellectual property (IP) cores from the Quartus library, red blocks indicate integrated HPS modules in the Cyclone V SoC, yellow blocks are external integrated circuits (ICs) on the SoCKit and on the daughterboard, and white blocks encompass miscellaneous external elements.

In essence, the AM-ACSK/FM-ACSK transmitter generates chaotic symbols (8192 samples per modulating bit) from a pseudorandom binary data input sequence. The output is converted to analog via DAC, transmitted over a 62 cm coaxial cable to ADC, and digitized back to a 14-bit signal. Converters are designed to process unsigned signals; therefore, the MSB is inverted both prior to the DAC (where two's complement is converted to offset binary) and after the ADC (where the reverse process occurs). Subsequently, channel noise and selective fading (when necessary) are added to

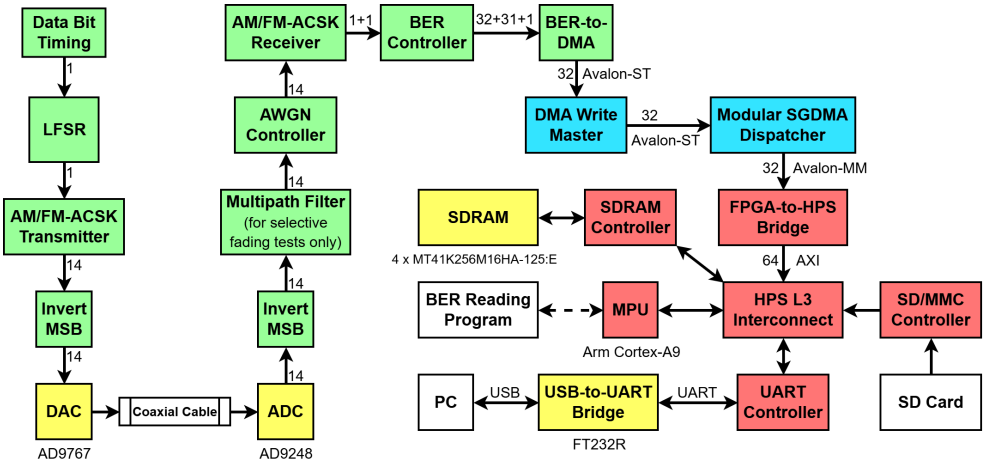


Fig. 5.2. Simplified schematic of the AM-ACSK/FM-ACSK FPGA prototype's overall experimental configuration.

the signal before it is processed by the AM-ACSK/FM-ACSK receiver. The recovered bits are then utilized for BER computation, with the results transmitted to the personal computer (PC) for display, through the HPS.

Beyond the core AM-ACSK/FM-ACSK system (the transceiver), additional FPGA modules were developed for the purpose of evaluating the system's performance (see Fig. 5.2). The following elements are included: data bit timing, linear feedback shift register (LFSR), AWGN controller, BER controller, and BER-to-DMA.

5.2 AM-ACSK FPGA Prototype AWGN Channel Performance Evaluation

As depicted in Fig. 5.3, the FPGA implementation of AM-ACSK yields a BER curve that closely follows the Simulink simulation across the overlapping SNR range, confirming hardware fidelity to the model. The FPGA results extend to 16 dB SNR, achieving BER below 10^{-6} (versus the simulation limit of below 10^{-4} at 10 dB). The observed discrepancies are small and expected for a hardware implementation: analog passband channel imperfections, synchronization offsets and other nonidealities introduce additional effective noise and implementation loss that explain the $\sim$1–2 dB degradation.

5.3 FM-ACSK FPGA Prototype Performance Evaluation

5.3.1 AWGN Channel Performance

As illustrated in Fig. 5.4, the BER performance of the FM-ACSK chaotic communication system is demonstrated under AWGN conditions. Same as in the AM-ACSK case, this figure consists of

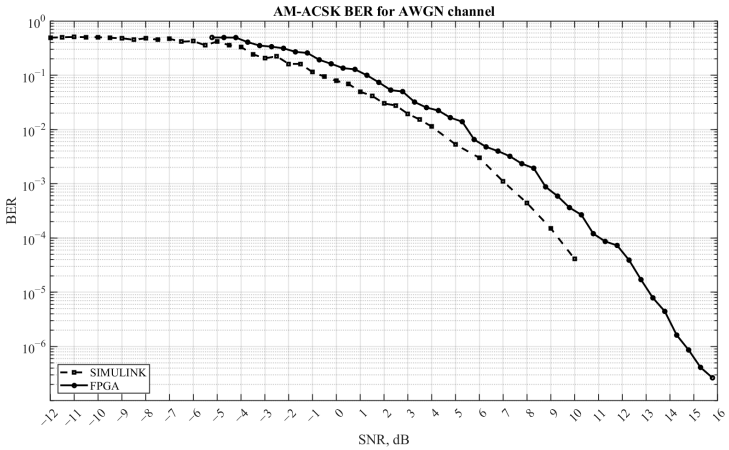


Fig. 5.3. AM-ACSK BER for AWGN channel – MATLAB Simulink and FPGA results.

two components: the MATLAB Simulink simulation curve, which was already presented in the prior section, and the experimental results from the FPGA prototype, which are used for direct comparison. The FPGA results demonstrate a BER curve that is somewhat less smooth than the simulation, with minor irregularities, and exhibit reduced tolerance to noise – shifted by approximately 1 dB in SNR toward higher values for equivalent BER levels. This finding aligns with expectations, given the FPGA implementation's increased complexity, which encompasses analog circuitry and real-world hardware constraints that are not present in the idealized simulation environment.

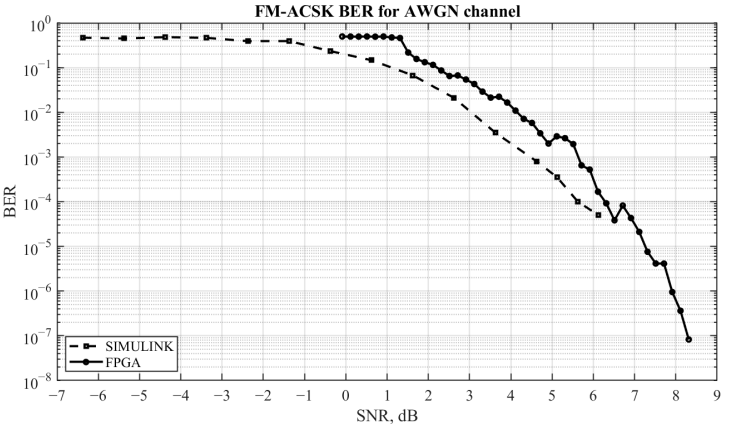


Fig. 5.4. FM-ACSK BER for AWGN channel – MATLAB Simulink and FPGA results.

5.3.2 Selective Fading Channel Performance

Figure 5.5 presents the FPGA prototype BER test results for varying depths of selective fading notches across a range of SNR levels. At elevated SNR values (exceeding 4–5 dB), system performance with shallow notches of 1.5 dB and 3.0 dB remains comparable to that in the absence of

fading. However, deeper notches result in substantial BER degradation due to the heightened distortion in the FM-ACSK signal's spectrum. Conversely, under conditions of high noise (SNR below 2–3 dB), the presence of selective fading yields performance comparable to the non-fading scenario, with most curves overlapping closely around 10^{-1}. Exceptional behavior is demonstrated by the 7.5 dB notch depth test, which showed noticeably worse result than 9.0 dB notch depth test with irregular slope at higher noise levels.

When compared with the MATLAB Simulink simulation results of the same experiment (Fig. 4.7), the FPGA curves exhibit high level of overall agreement in trend and relative ordering of notch-depth effects (except for 7.5 dB notch depth). Both confirm that shallow notches (1.5 dB and 3.0 dB) cause negligible degradation at high SNR, while deeper notches introduce progressive impairment; the hardware prototype, however, attains more than two orders of magnitude lower BER floors and extends to higher SNR values.

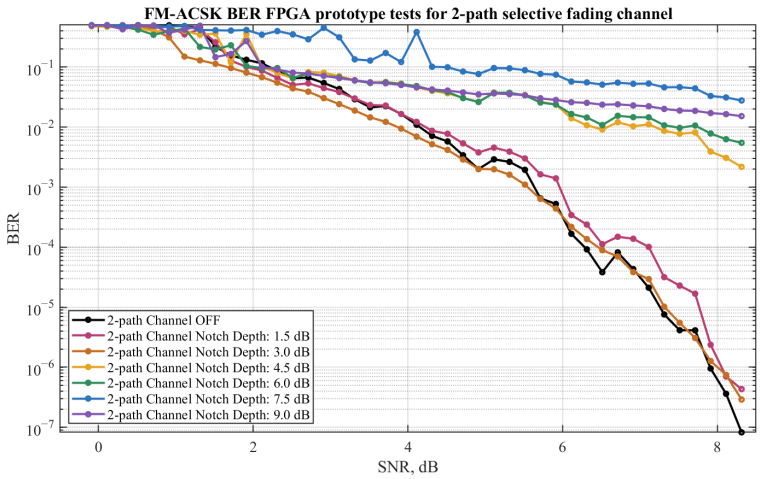


Fig. 5.5. BER curves for varying two-path selective fading notch depths at different SNR values in FPGA prototype tests.

5.4 Conclusions

This section was devoted to the performance evaluation of the chaotic communication system FPGA prototypes. The experimental setup utilizing the Arrow SoCKit board, auxiliary modules (LFSR data generator, AWGN controller, BER controller, and DAC/ADC interfacing via coaxial cable), and real-time testing methodology was presented, followed by BER measurements for both AM-ACSK and FM-ACSK transceiver systems in AWGN and two-path selective fading channels. These empirical findings provide a solid basis for anticipating the noise-handling capabilities of the implemented chaotic communication systems. Experimental validation of the FM-ACSK's superior noise performance, achieving 10^{-3} BER at approximately 5.6 dB SNR (versus 8.7 dB SNR for AM-ACSK), which serve to confirm the third and fourth theses stated in the introduction.

OVERALL CONCLUSIONS

This thesis presents an investigation into the development of FPGA-based digital chaotic communication transceiver systems, centered on a modified Chua oscillator combined with error-feedback synchronization and two innovative hybrid modulations – AM-ACSK and FM-ACSK. These modulations integrate digital coherent baseband antipodal chaos shift keying with analog passband amplitude and frequency modulation to deliver robust physical-layer signal protection while addressing practical constraints in wireless communications and IoT environments. The research encompasses mathematical modeling, simulation validation, FPGA hardware prototyping, and performance benchmarking under realistic channel conditions.

The primary aim – to research the coherent discrete chaotic communication system design options and to develop a set of approaches for the FPGA implementation of such systems – has been realized. Every task set out in the Introduction was carried out: a thorough review of chaotic communication principles and prior FPGA solutions; formulation of system models using forward-Euler discretization of the modified Chua circuit's ODE in fixed-point arithmetic; proof-of-concept testing in Simulink; VHDL-based prototype development on a single Altera/Intel Cyclone V FPGA; and experimental validation through custom auxiliary modules for data generation, noise injection, and BER assessment.

The key achievements of the study can be highlighted as follows.

- A fully digital version of the modified Chua generator, together with error-feedback synchronization and ACSK modulation (fixed at 8192 samples per symbol), was realized and paired with NCO-driven AM and FM passband stages operating at a 10.7 MHz carrier. This architecture maintains chaotic behavior while fitting within the logic resources of a single Cyclone V FPGA device running at 50 MHz.
- Complete transmitter and receiver chains were implemented, incorporating digital incoherent AM and FM demodulators, 215 kHz cutoff filters for noise suppression, AGC (AM-ACSK only), early-late gate timing recovery, and integration-based bit detection. The resulting prototypes deliver a consistent uncoded data rate of 6.1 kbps within 430 kHz (AM-ACSK) and 1.720 MHz (FM-ACSK, modulation index 3) bandwidths.
- High-fidelity Simulink models mirroring the prototypes' digital signal paths enabled direct simulation comparison with hardware results, confirming close agreement once real world effects and analog-channel imperfections are accounted for.
- Comprehensive BER characterization in AWGN channel demonstrated clear performance differences: FM-ACSK exhibits noticeably better noise tolerance owing to its constant envelope and wider spectral occupancy, reaching 10^{-3} BER at 5.6 dB SNR versus 8.7 dB SNR required by AM-ACSK.
- The four theses stated in the Introduction were empirically confirmed through the realized prototypes, establishing the implementability of both chaotic communication systems on an

FPGA hardware with the respective BER levels.

Both system variants were developed and assessed within a unified simulation-to-hardware workflow. Simulink served for rapid design iteration and theoretical benchmarking, while the FPGA prototypes – linked via DAC/ADC and coaxial cable – provided real-time testing under controlled AWGN and multipath conditions using LFSR-generated data and dedicated BER controller for the test data acquisition. This dual validation approach not only verified the theoretical models but also exposed the minor implementation gaps arising from quantization and analog interfacing, providing practical guidance for future refinements.

Taken together, these outcomes illustrate that coherent chaotic communication systems can be deployed on FPGA platforms and utilized for communication purposes with enhanced physical-layer signal protection. The noise-like, spread-spectrum properties of the generated carriers, combined with the strict synchronization requirement at the receiver, inherently raise the barrier for unauthorized interception, while the measured BER performance remains suitable for low-rate applications such as secure IoT sensor links, industrial monitoring, or covert tactical communications. The FM-ACSK variant stands out for its superior noise resilience, suggesting it as a strong candidate for environments with challenging propagation or interference.

In summary, this Thesis successfully fulfills its stated objective, completes all research tasks, and fully substantiates the defended theses. The developed methodologies, prototypes, and performance data establish a practical foundation for advancing FPGA-based chaotic communication systems in real-world communication scenarios.

The results of this study should be interpreted within the context of the selected system configuration, which includes low-rate uncoded transmission, one specific FPGA platform, fixed-point arithmetic, and controlled AWGN/selective fading channel conditions. Subsequent efforts are necessary to attain enhanced data transmission rates, refined synchronization mechanisms, advanced channel coding techniques, and a comprehensive evaluation of signal protection measures.

Directions for continued study include scaling data rates through higher clock frequencies and deeper pipelining level, incorporating adaptive mechanisms that respond to dynamic channel variations, exploring FEC integration for further BER improvement, and conducting detailed signal protection evaluations against modern eavesdropping and jamming techniques. Multi-user access could also widen the application scope of these chaotic communication systems. The frameworks and insights provided here offer a solid starting point for these extensions.

BIBLIOGRAPHY

[1] T. Tami et al. “Chaos secure communication’ implementation in FPGA.” In: *2018 International Conference on Applied Smart Systems (ICASS)*. 2018, pp. 1–6. DOI: 10.1109/ICASS.2018.8652046.

[2] K. Sun. *Chaotic Secure Communication: Principles and Technologies*. De Gruyter, 2016. ISBN: 978-3-11-043406-4.

[3] P. Stavroulakis. *Chaos applications in telecommunications*. CRC Press, 2006. ISBN: 978-0-203-02531-4. DOI: 10.1201/9780203025314.

[4] S. Çiçek, U. E. Kocamaz, and Y. Uyaroğlu. “Secure Chaotic Communication with Jerk Chaotic System Using Sliding Mode Control Method and Its Real Circuit Implementation.” In: *Iranian Journal of Science and Technology – Transactions of Electrical Engineering* 43.3 (2019), pp. 687–698. ISSN: 23641827. DOI: 10.1007/s40998-019-00184-9.

[5] B. Muthuswamy and S. Banerjee. *A Route to Chaos Using FPGAs*. Vol. 16. Springer International Publishing, 2015. ISBN: 978-3-319-18104-2. DOI: 10.1007/978-3-319-18105-9.

[6] E. Tlelo-Cuautle, J. d. J. Rangel-Magdaleno, and L. G. de la Fraga. *Engineering applications of FPGAs: Chaotic systems, artificial neural networks, random number generators, and secure communication systems*. Springer International Publishing, 2016. ISBN: 978-3-319-34115-6. DOI: 10.1007/978-3-319-34115-6.

[7] F. Capligins et al. “Frequency-Modulated Antipodal Chaos Shift Keying Chaotic Communication on Field Program Gate Array: Prototype Design and Performance Insights.” en. In: *Applied Sciences* 15.3 (Jan. 2025), p. 1156. ISSN: 2076-3417. DOI: 10.3390/app15031156.

[8] M. F. Taşdemir et al. “Performance Evaluation of FPGA-Based Design of Modified Chua Oscillator.” en. In: *Chaos Theory and Applications* 6.4 (Nov. 2024), pp. 249–256. ISSN: 2687-4539. DOI: 10.51537/chaos.1466919.

[9] D. Cirjulina et al. “Experimental Study on Colpitts Chaotic Oscillator-Based Communication System Application for the Internet of Things.” en. In: *Applied Sciences* 14.3 (Jan. 2024), p. 1180. ISSN: 2076-3417. DOI: 10.3390/app14031180.

[10] R. Babajans et al. “Synchronization of Analog-Discrete Chaotic Systems for Wireless Sensor Network Design.” en. In: *Applied Sciences* 14.2 (Jan. 2024), p. 915. ISSN: 2076-3417. DOI: 10.3390/app14020915.

[11] R. Babajans et al. “Performance Analysis of Vilnius Chaos Oscillator-Based Digital Data Transmission Systems for IoT.” en. In: *Electronics* 12.3 (Jan. 2023), p. 709. ISSN: 2079-9292. DOI: 10.3390/electronics12030709.

[12] F. Capligins et al. “FPGA-Based Antipodal Chaotic Shift Keying Communication System.” en. In: *Electronics* 11.12 (June 2022), p. 1870. ISSN: 2079-9292. DOI: 10.3390/electronics11121870.

[13] F. Capligins et al. “Experimental Study of the Chaotic Jerk Circuit Application for Chaos Shift Keying.” In: *Latvian Journal of Physics and Technical Sciences* 58.4 (2021), pp. 55–68. DOI: 10.2478/lpts-2021-0033.

[14] F. Capligins, A. Litvinenko, and D. Kolosovs. “Selective Fading Channel Performance Evaluation for Frequency-Modulated Antipodal Chaos Shift Keying Communication System.” In: *2025 IEEE 12th Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE)*. 2025, pp. 1–5. DOI: 10.1109/AIEEE66149.2025.11050770.

[15] D. Cirjulina et al. “Fundamental Frequency Impact on Vilnius Chaos Oscillator Dynamics.” In: *16th Chaotic Modeling and Simulation International Conference*. Ed. by C. H. Skiadas and Y. Dimotikalis. Cham: Springer Nature Switzerland, 2024, pp. 87–101. ISBN: 978-3-031-60907-7. DOI: 10.1007/978-3-031-60907-7_8.

[16] R. Babajans et al. “Study on Analog and Discrete Chaos Oscillators Synchronization.” In: *16th Chaotic Modeling and Simulation International Conference*. Ed. by C. H. Skiadas and Y. Dimotikalis. Cham: Springer Nature Switzerland, 2024, pp. 33–46. ISBN: 978-3-031-60907-7. DOI: 10.1007/978-3-031-60907-7_4.

[17] F. Capligins, A. Litvinenko, and D. Kolosovs. “Performance Evaluation of Frequency Modulated Antipodal Chaos Shift Keying Digital Communication System.” en. In: *2023 Workshop on Microwave Theory and Technology in Wireless Communications (MTTW)*. Riga, Latvia: IEEE, Oct. 2023, pp. 29–33. ISBN: 9798350393491. DOI: 10.1109/MTTW59774.2023.10320019.

[18] A. Aboltins et al. “Implementation of chaotic frequency modulation based spread spectrum communication system in software-defined radio.” In: *2023 IEEE Wireless Communications and Networking Conference (WCNC)*. 2023, pp. 1–6. DOI: 10.1109/WCNC55385.2023.10118612.

[19] F. Capligins, A. Litvinenko, and D. Kolosovs. “FPGA Implementation and Study of Antipodal Chaos Shift Keying Communication System.” In: *2021 IEEE Microwave Theory and Techniques in Wireless Communications (MTTW)* (2021), pp. 1–6. DOI: 10.1109/mttw53539.2021.9607226.

[20] F. Capligins et al. “FPGA Implementation and Study of Synchronization of Modified Chua’s Circuit-Based Chaotic Oscillator for High-Speed Secure Communications.” In: 2021. ISBN: 978-1-66542-538-4. DOI: 10.1109/AIEEE51419.2021.9435783.

[21] L. O. Chua et al. “A Universal Circuit for Studying and Generating Chaos – Part II: Strange Attractors.” In: *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications* 40.10 (1993), pp. 745–761. ISSN: 10577122. DOI: 10.1109/81.246150.

[22] T. Carroll and L. Pecora. “Synchronizing chaotic circuits.” In: *IEEE Transactions on Circuits and Systems* 38.4 (1991), pp. 453–456. DOI: 10.1109/31.75404.

[23] H. G. Chou et al. “A fuzzy-model-based chaotic synchronization and its implementation on a secure communication system.” In: *IEEE Transactions on Information Forensics and Security* 8.12 (2013), pp. 2177–2185. ISSN: 15566013. DOI: 10.1109/TIFS.2013.2286268.

[24] G. Kaddoum. “Wireless Chaos-Based Communication Systems: A Comprehensive Survey.” en. In: *IEEE Access* 4 (2016), pp. 2621–2648. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2016.2572730.

[25] E. Günay, K. Altun, and C. Ünal. “Implementation of CSK communicating system with switched SC-CNN based chaos generator.” In: *2017 10th International Conference on Electrical and Electronics Engineering (ELECO)*. 2017, pp. 1364–1367.

[26] G. Kaddoum and N. Tadayon. “Differential Chaos Shift Keying: A Robust Modulation Scheme for Power-Line Communications.” en. In: *IEEE Transactions on Circuits and Systems II: Express Briefs* 64.1 (Jan. 2017), pp. 31–35. ISSN: 1549-7747, 1558-3791. DOI: 10.1109/TCSII.2016.2546901.

[27] L. Chua et al. “A universal circuit for studying and generating chaos. I. Routes to chaos.” In: *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications* 40.10 (Oct. 1993), pp. 732–744. ISSN: 10577122. DOI: 10.1109/81.246149.

[28] O. Rössler. “An equation for continuous chaos.” In: *Physics Letters A* 57.5 (July 1976), pp. 397–398. ISSN: 0375-9601. DOI: 10.1016/0375-9601(76)90101-8.

[29] J. C. Sprott. “A new class of chaotic circuit.” In: *Physics Letters, Section A: General, Atomic and Solid State Physics* 266.1 (2000), pp. 19–23. ISSN: 03759601. DOI: 10.1016/S0375-9601(00)00026-8.

[30] A. Elwakil and M. Kennedy. “A family of Colpitts-like chaotic oscillators.” en. In: *Journal of the Franklin Institute* 336.4 (May 1999), pp. 687–700. ISSN: 00160032. DOI: 10.1016/S0016-0032(98)00046-5.

[31] Y. Luo, S. Yu, and J. Liu. “Design and implementation of image chaotic communication via FPGA embedded ethernet transmission.” In: *2009 International Workshop on Chaos-Fractals Theories and Applications, IWCFTA 2009* (2009), pp. 148–152. DOI: 10.1109/IWCFTA.2009.38.

[32] E. Tlelo-Cuautle et al. “FPGA realization of a chaotic communication system applied to image processing.” In: *Nonlinear Dynamics* 82.4 (2015), pp. 1879–1892. ISSN: 1573269X. DOI: 10.1007/s11071-015-2284-x.

[33] E. Gunay and K. Altun. “A performance comparison study of programmable platforms: FPAA and FPGA implementation of COOK communication system.” In: *2017 European Conference on Circuit Theory and Design, ECCTD 2017* (2017), pp. 1–4. DOI: 10.1109/ECCTD.2017.8093237.

[34] J. Schmitz and L. Zhang. “Rössler-based chaotic communication system implemented on FPGA.” In: *2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)*. 2017, pp. 1–4. DOI: 10.1109/CCECE.2017.7946729.

[35] E. Tlelo-Cuautle et al. “Dynamics, FPGA realization and application of a chaotic system with an infinite number of equilibrium points.” In: *Nonlinear Dynamics* 89.2 (2017), pp. 1129–1139. ISSN: 1573269X. DOI: 10.1007/s11071-017-3505-2.

[36] O. Guillén-Fernández et al. “On the synchronization techniques of chaotic oscillators and their FPGA-based implementation for secure image transmission.” In: *PLoS ONE* 14.2 (2019), pp. 1–34. ISSN: 19326203. DOI: 10.1371/journal.pone.0209618.

[37] J.-C. Nuñez-Perez et al. “FPGA Realization of an Image Encryption System Using a 16-CPSK Modulation Technique.” en. In: *Electronics* 13.22 (Nov. 2024), p. 4337. ISSN: 2079-9292. DOI: 10.3390/electronics13224337.

[38] M.-A. Estudillo-Valdez, V.-A. Adeyemi, and J.-C. Nuñez-Perez. “FPGA realization of an image encryption system using the DCSK-CDMA technique.” en. In: *Integration* 96 (May 2024), p. 102157. ISSN: 01679260. DOI: 10.1016/j.vlsi.2024.102157.

[39] T. Bonny and W. Al Nassan. “Optimizing Security and Cost Efficiency in N-Level Cascaded Chaotic-Based Secure Communication System.” en. In: *Applied System Innovation* 7.6 (Oct. 2024), p. 107. ISSN: 2571-5577. DOI: 10.3390/asi7060107.

[40] N. F. Karagiorgos et al. “Unconventional Security for IoT: Hardware and Software Implementation of a Digital Chaotic Encrypted Communication Scheme.” en. In: *IEEE Internet of Things Journal* 11.11 (June 2024), pp. 19914–19925. ISSN: 2327-4662, 2372-2541. DOI: 10.1109/JIOT.2024.3371091.

[41] J.-C. Nuñez-Perez et al. “224-CPSK–CSS–WCDMA FPGA-Based Reconfigurable Chaotic Modulation for Multiuser Communications in the 2.45 GHz Band.” In: *Electronics* 14.20 (2025). ISSN: 2079-9292. DOI: 10.3390/electronics14203995. URL: https://www.mdpi.com/2079-9292/14/20/3995.

[42] L. Xiao, L. Zhao, and J. Jin. “Preset-Time Convergence Fuzzy Zeroing Neural Network for Chaotic System Synchronization: FPGA Validation and Secure Communication Applications.” In: *Sensors* 25.17 (2025). ISSN: 1424-8220. DOI: 10.3390/s25175394. URL: https://www.mdpi.com/1424-8220/25/17/5394.

[43] M. H. Abd et al. “Synchronization of Monostatic Radar Using a Time-Delayed Chaos-Based FM Waveform.” In: *Remote Sensing* 14.9 (2022). ISSN: 2072-4292. DOI: 10.3390/rs14091984.

[44] T. Hoang. *Simulink model for observer based synchronization in Chua’s systems*. MATLAB Central File Exchange. URL: https://www.mathworks.com/matlabcentral/fileexchange/26246-observer-based-synchronization-in-chua-s-systems (visited on 02/12/2019).

[45] S. Bendoukha, S. Abdelmalek, and A. Ouannas. “Secure communication systems based on the synchronization of chaotic systems.” In: *Studies in Systems, Decision and Control* 200 (2019), pp. 281–311. ISSN: 21984190. DOI: 10.1007/978-3-030-12232-4_9.

[46] U. Mengali and A. N. D’Andrea. *Synchronization Techniques for Digital Receivers*. en. Boston, MA: Springer US, 1997. ISBN: 978-1-4899-1807-9. DOI: 10.1007/978-1-4899-1807-9.

[47] P. Zicari, P. Corsonello, and S. Perri. “A high flexible Early-Late Gate bit synchronizer in FPGA-based software defined radios.” en. In: *2008 4th European Conference on Circuits and Systems for Communications*. Bucharest, Romania: IEEE, July 2008, pp. 252–255. ISBN: 978-1-4244-2419-1. DOI: 10.1109/ECCSC.2008.4611687.

Filips Čapligins was born in 1993 in Riga, Latvia. He received an Academic Bachelor's degree in Electrical Engineering (2019) and a Professional Master's degree in Electronics and the qualification of Lead Electronics Engineer (2022) from Riga Technical University (RTU). Since 2020, he has been employed at RTU, initially as a research assistant and, since 2022, as a researcher. Since 2026, he has been a radio electronics engineer at JSC "SAF Tehnika". His research interests include the application of chaos theory in telecommunications, digital signal processing, and FPGA technologies.