Industrial Engineering Journal
ISSN: 0970-2555
Volume : 54, Issue 7, July : 2025

LOW-POWER, HIGH-THROUGHPUT REED-SOLOMON CODEC DESIGN USING GALOIS FIELD OPERATIONS AND CLOCK GATING

AKUTHOTA BHAVITHA SRI, Dept of Electronics and Communication (VLSI System Design) Engineering, JNTUH University College of Engineering Sultanpur, Sangareddy, Telangana, India Email: bhavithasri923@gmail.com
Mr. T. MOHAN DAS Assistant Professor(C) Dept of Electronics and Communication (VLSI System Design) Engineering, JNTUH University College of Engineering Sultanpur, Sangareddy, Telangana, India Email: mohandas.nitdgp@gmail.com

ABSTRACT

Reed-Solomon (RS) codes are essential in digital communication systems for their robust error correction capabilities, ensuring data integrity across various industries. This research focuses on the advanced implementation of RS codes, particularly highlighting enhancements such as configurable data width, interleaving for burst error protection, and systematic encoding. The encoding leverages optimized Galois Field operations with clock-gated architectures for power efficiency, while decoding employs the Chien search, Berlekamp-Massey algorithm, and Forney's error magnitude calculations for precise error correction. The project implements these techniques on both ASIC and FPGA platforms, utilizing tools like Vivado and Cadence. Comprehensive testing evaluates the performance in correcting multiple error scenarios, with the final ASIC design yielding a GDSII file for chip manufacturing. This research aims to create a power-efficient, high-performance error correction system for critical applications, including digital communication, aerospace, and data storage systems. The results demonstrate the enhanced capabilities of RS codes in systematic and non-systematic configurations, showcasing their utility across multiple platforms.

Key Words: Very Large-Scale Integration (VLSI), Reed-Solomon, Berlekamp-Massey, Chien Search, Forney Algorithm, Vivado, Cadence, Clock Gating.

I INTRODUCTION

In modern digital systems, errors signify deviations from expected outcomes and can manifest in various ways depending on the application. In digital communication, these errors typically emerge during data transmission due to noise, attenuation, distortion, interference, or multipath fading, leading to single-bit or burst errors that adversely affect the performance and reliability of services like mobile communication, internet usage, and video streaming [1], [2]. Similarly, in data storage systems, errors occur during reading, writing, or data retention on media such as hard drives (HDDs), solid-state drives (SSDs), and optical disks. These may arise from physical damage, media degradation, electrical faults, or environmental conditions [3], [4]. The resulting errors—whether soft, hard, or silent corruptions—can threaten data

integrity and cause system failures. Ensuring robust error correction is essential to maintaining data reliability and system performance. In communication systems, techniques like Forward Error Correction (FEC) prevent the need for retransmission, support real-time processing, and enhance overall data accuracy [5]. Meanwhile, storage systems utilize Error-Correcting Codes (ECC) and redundancy mechanisms such as RAID to guard against data loss and media deterioration [6], [7]. Without effective error correction, digital systems face considerable challenges, including call drops, corrupted files, system crashes, and unreliable data access. As technologies like 5G networks and large-scale cloud storage continue to evolve, the role of error correction becomes increasingly critical [8].

Among the most effective solutions are Reed-Solomon (RS) codes, a class of forward error correction codes renowned for their ability to detect and correct multiple errors. Introduced in 1960 by Irving Reed and Gustave Solomon, RS codes operate over finite fields (Galois fields), allowing them to correct both random and burst errors efficiently [9], [10]. Their mathematical precision and adaptability have led to widespread adoption in areas such as telecommunications, deep-space communication, optical storage, and flash memory [11]. We rely heavily on digital communication tools like smartphones, internet, and data storage devices. But sometimes, errors can occur during data transmission or storage, which can lead to data corruption. To prevent this, we need strong error correction methods. Reed-Solomon (RS) codes are one such method that helps ensure data integrity by correcting errors [12]. This is especially important in critical applications like space communication, medical data transfer, and financial transactions. This research aims to improve the reliability of digital communication systems, which we use daily. By ensuring that data is transmitted and received accurately, we can enhance the overall performance of these systems [13].

II LITERATURE SURVEY

Reed-Solomon (RS) codes are widely regarded as one of the most effective error-correcting codes, primarily due to their robust performance in detecting and correcting both random and burst errors across a variety of communication and storage systems. The foundational principles of RS codes were laid by Reed and Solomon in the 1960s, and since then, numerous studies have contributed to their optimization and implementation. Yin et al. [1] introduced an advanced burst-error-correcting algorithm specifically tailored for RS codes, emphasizing improvements in decoding latency and correction efficiency. Their approach demonstrated that RS codes, when adapted to burst-error patterns commonly found in storage and transmission media, offer superior data integrity. Shah et al. [2] extended this research by focusing on practical implementations of RS coding and interleaving on FPGA platforms, particularly in the context of Digital Video Broadcasting (DVB-T2). Their study underscored the challenges in real-time processing of high-volume data and showcased efficient mapping of RS encoders and interleavers onto reconfigurable logic, thereby balancing performance and hardware complexity.

In the realm of low-power design, Kandpal et al. [3] highlighted techniques to reduce power consumption during RS encoding and decoding. Their work revealed how architectural

modifications and careful selection of data paths can lead to energy savings without significantly compromising throughput. Complementing this, Ertuğrul [4] investigated the use of Polar and RS concatenated codes for optical communication systems, demonstrating that hybrid coding schemes leverage the strengths of both components to enhance error resilience in highly noisy channels. Samanta et al. [5] contributed to the literature by presenting an RS(47,41) codec tailored for intelligent home networking. Their research focused on comparative analyses of FPGA and ASIC implementations, suggesting that while FPGAs offer design flexibility, ASICs provide superior power efficiency and area optimization—critical attributes in consumer electronic environments.

Wahbi et al. [6] proposed an embedded Chien Search block optimized for RS decoders through a novel factorization of the error locator polynomial. Their work significantly reduced the computational burden traditionally associated with root-finding operations in the decoding process. Tan et al. [7] built on this by presenting a parallel architecture for the Chien Search and Forney algorithm, targeting high-throughput systems. They demonstrated how pipeline structures and concurrent evaluations could dramatically increase decoding speeds while maintaining accuracy, a vital requirement in high-bandwidth applications such as digital video broadcasting and wireless communications. Clarke [8], in his technical white paper for BBC Research & Development, provided an in-depth overview of RS error correction principles and discussed practical implementation issues, including trade-offs between correction capability, codeword length, and system complexity.

Singh and Sujana [9] examined the ASIC implementation of an RS codec designed for burst error detection and correction. Their results showed that hardware implementations tailored to specific error models could outperform generic solutions in both latency and energy consumption. Krishnan et al. [10] also pursued VLSI implementations, highlighting the need for architectural modularity to accommodate different code rates and block sizes. Their work stressed the importance of flexibility in RS codec design, especially for systems that operate across multiple communication protocols. Sezer [11] tackled the challenge of implementing high-throughput decoder architectures for product RS codes in optical communication systems. His thesis presented techniques to improve decoding parallelism, including the use of advanced scheduling and pipelining techniques to manage data dependencies and resource utilization.

Kuila [12] focused on RS(255,223) codes implemented on FPGAs and discussed design challenges related to timing closure and power optimization. His findings reaffirmed the advantage of using RS codes in large block length applications and demonstrated that careful design of syndrome and error computation blocks can substantially enhance efficiency. Finally, Sklar [13], in his seminal textbook on digital communications, thoroughly explained the operational theory behind RS codes, including finite field arithmetic, syndrome decoding, and the interplay of encoding and error correction logic. His treatment of RS codes remains a cornerstone reference, bridging theoretical foundations with practical applications and inspiring a generation of engineers to apply RS codes in real-world systems.

Taken together, these works form a comprehensive body of knowledge that advances the theory and application of Reed-Solomon codes across diverse technological domains. The progression from theoretical enhancements in burst-error correction to efficient VLSI implementations illustrates a clear trajectory in RS codec research, driven by the dual goals of robustness and efficiency. Several consistent themes emerge: the emphasis on reducing power consumption through architectural innovations, the push for parallelism and pipelining to meet throughput demands, and the need for flexibility and scalability in codec design to support evolving standards. Additionally, the transition from software-based simulations to hardware implementations—whether on FPGA or ASIC—underscores the growing importance of deploying RS codes in embedded and resource-constrained environments.

Moreover, the use of concatenated coding strategies, such as combining RS codes with Polar or convolutional codes, demonstrates how hybrid systems can be engineered for enhanced performance in noisy or bandwidth-constrained scenarios. The contributions on clock-efficient arithmetic units and pipelined decoding algorithms further demonstrate that RS decoders can be significantly optimized when designed with application-specific constraints in mind. Notably, clock gating and dynamic logic activation—though not always explicitly mentioned in earlier works—are now recognized as critical techniques for reducing dynamic power and extending system longevity, especially in battery-powered and mobile devices.

In conclusion, the literature clearly indicates that RS codes remain highly relevant in modern digital systems, with a growing emphasis on real-time, power-efficient, and scalable hardware solutions. From advanced algorithmic strategies to practical VLSI and FPGA implementations, the evolution of Reed-Solomon codec design continues to align with the increasing demands of digital communication, storage, and embedded systems. The foundation laid by these studies not only informs current practices but also opens avenues for future research into machine learning-assisted error correction, adaptive coding techniques, and fault-tolerant decoder architectures, ensuring the ongoing relevance and innovation in the field of error-correcting codes.

III METHODOLOGY

The development of a low-power, high-throughput Reed-Solomon (RS) codec is a multifaceted process that involves careful consideration of algorithmic complexity, hardware architecture, and power optimization strategies. The project focuses on implementing an RS(15,5) code using arithmetic over Galois Field GF(2^4), where the total codeword length is 15 symbols and the number of message symbols is 5. The core objective is to design a reliable encoder and decoder system that not only ensures data integrity through effective error correction but also minimizes power consumption using clock gating. At the beginning of the design process, theoretical principles of RS codes were analyzed, including polynomial generation, parity calculations, and decoding mechanisms such as syndrome computation, Berlekamp-Massey algorithm for error locator polynomial generation, Chien search for error location, and Forney's algorithm for error magnitude computation. Galois Field operations were implemented through log and antilog table approaches, which are precomputed and stored in ROMs to facilitate fast

multiplication and inversion. These lookup tables significantly reduce the complexity of arithmetic operations and enhance the speed of computation in both encoding and decoding processes.

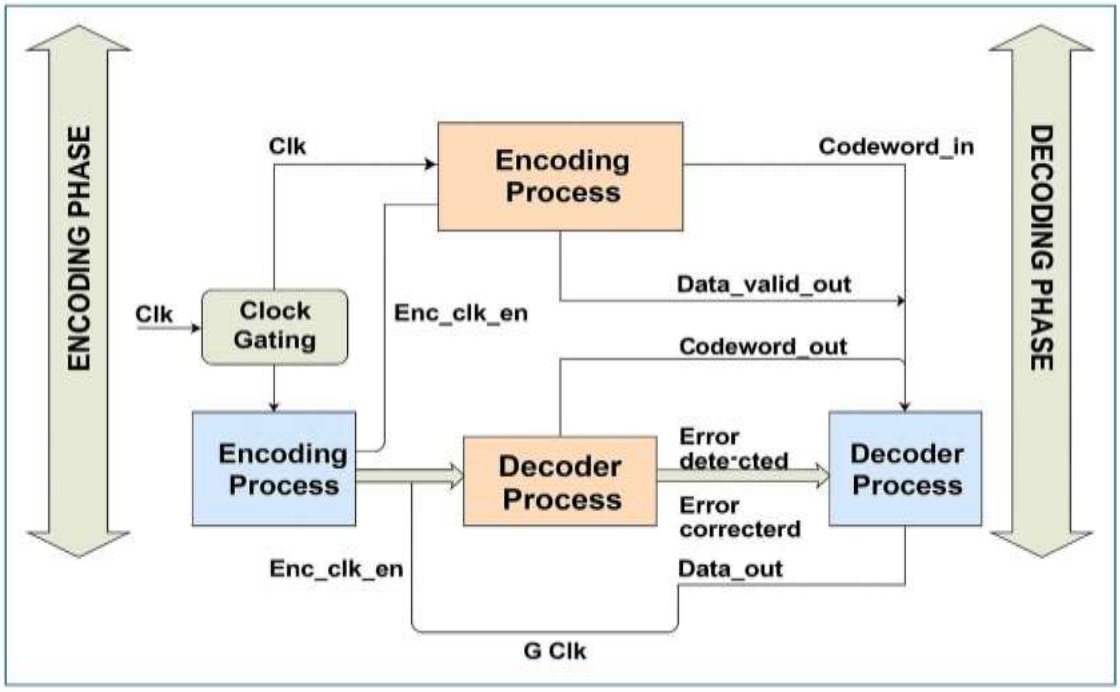

Fig 1. Block Diagram of RS(15,5) Codec Process

The encoder module follows a systematic encoding scheme, ensuring that the original message symbols are retained in the output codeword while appending calculated parity symbols. The parity generation employs a Linear Feedback Shift Register (LFSR) configuration, which simplifies polynomial division and enhances throughput. A finite state machine (FSM) with states including IDLE, PROCESS, and OUTPUT governs the operation flow, managing the input of message symbols, the computation of parity, and the formation of the complete codeword. On the decoder side, a more complex FSM is designed with seven states, encompassing functions like syndrome detection, error polynomial calculation, error location detection, and error correction. The FSM ensures sequential yet efficient traversal through each of these algorithmic stages. The decoding algorithm starts by calculating the syndromes from the received codeword. If the syndromes are non-zero, the Berlekamp-Massey algorithm is initiated to derive the error locator polynomial. The Chien search is used to find the positions of errors by evaluating the roots of the error locator polynomial. Once error positions are identified, Forney's algorithm computes the magnitudes, which are then used to correct the corrupted codeword.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 54, Issue 7, July : 2025

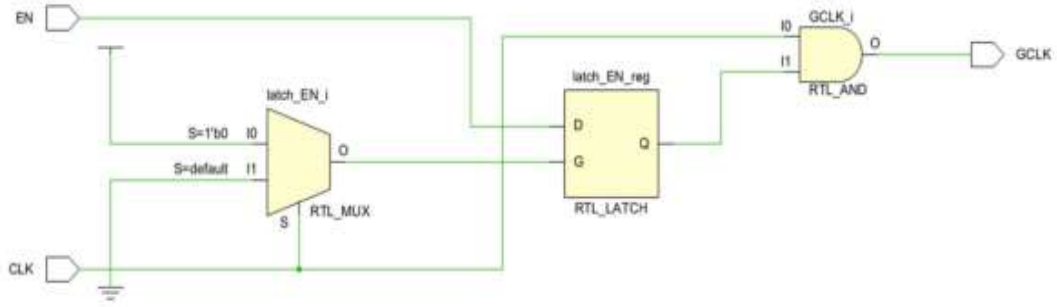

Fig 2.Schematic Diagram of Clock Gating Enable Logic

To achieve low power consumption, clock gating is integrated into both the encoder and decoder modules. Clock gating works by disabling the clock signal to specific submodules when they are not in use, thereby reducing dynamic power consumption. Control signals for clock gating are generated based on the current FSM state, ensuring that logic blocks are activated only when necessary. The application of clock gating is particularly effective in modules with intermittent activity, such as the syndrome calculator and the GF arithmetic units. The hardware description of the codec is written in Verilog HDL, which is then synthesized using Xilinx Vivado for FPGA implementation. A testbench is created to simulate various error conditions, including single and burst errors, to verify the error correction capabilities. Simulations are analyzed through waveform outputs, validating the accuracy of both the encoder and decoder. Resource utilization, power consumption, and timing characteristics are analyzed using the Vivado suite. The synthesized design is deployed on a Xilinx FPGA board, where the performance is observed in real-time using Integrated Logic Analyzer (ILA).

In parallel, the design is translated into an ASIC-compatible format using Cadence tools. The Verilog source code is synthesized using Genus, and physical implementation is carried out using Innovus to generate the GDSII layout. The ASIC flow includes cell placement, routing, clock tree synthesis, and timing analysis. The final design is targeted at a 28nm CMOS technology, which supports low power operation and high density. During the ASIC synthesis, power analysis is conducted both with and without clock gating, confirming a significant reduction in dynamic power when clock gating is employed. Timing analysis ensures that the design meets required setup and hold constraints, and area estimation provides insight into the silicon footprint of the design. The ASIC layout confirms the physical viability of the proposed codec for fabrication. Verification includes extensive functional testing, fault injection, and corner-case analysis to validate the robustness of the design. Performance metrics such as error correction rate, throughput, latency, and energy per bit are systematically measured. The results demonstrate that the use of clock gating in conjunction with optimized Galois Field operations provides a substantial improvement in power efficiency without compromising throughput or error correction accuracy. The methodology concludes with a complete and validated system capable of operating in power-constrained environments while maintaining high data reliability.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 54, Issue 7, July : 2025

IV PROPOSED SYSTEM

The proposed Reed-Solomon codec system is designed to deliver high-throughput and low-power error correction capabilities suitable for modern communication and data storage systems. It is based on the RS(15,5) configuration, which implies a 15-symbol codeword with 5 message symbols and 10 parity symbols, enabling the correction of up to 5 symbol errors. The system comprises an encoder and a decoder implemented using arithmetic over GF(2^4), and it includes embedded modules for Galois Field operations such as multiplication, addition, and inversion. The design utilizes precomputed lookup tables for log and antilog values to enhance arithmetic efficiency. These tables are stored in read-only memory blocks within the architecture, allowing rapid computation of required Galois Field values during encoding and decoding processes. The encoder follows a systematic design where the input message is appended with parity symbols, allowing the original data to be recovered easily from the output. The parity symbols are computed using polynomial division in GF(2^4), facilitated by a 10-stage Linear Feedback Shift Register. The encoder logic is managed by a finite state machine that handles data input, parity computation, and codeword generation.

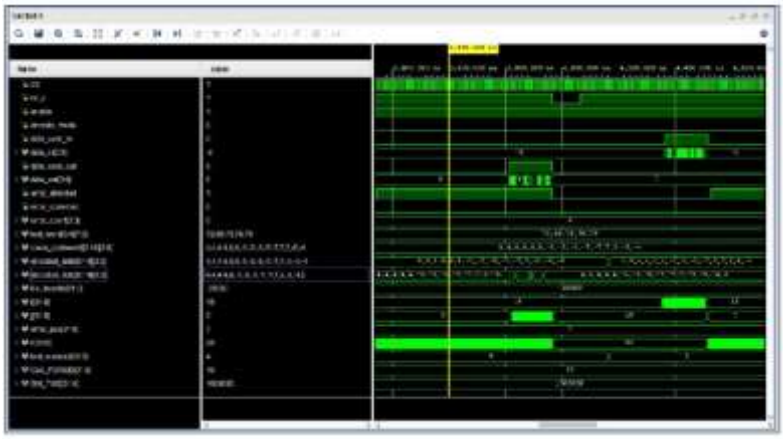

Fig 3.Simulation waveform of the RS(15,5) codec depicting encoding, error injection, and successful decoding with error correction

The decoder is built with a more elaborate architecture that processes received codewords to detect and correct errors. It consists of multiple algorithmic stages governed by a finite state machine with seven states, starting from IDLE and progressing through PROCESS, SYNDROME, KEY_EQUATION, CHIEN_SEARCH, ERROR_CORRECTION, and OUTPUT. The decoding begins by evaluating syndromes, which indicate whether errors are present in the codeword. If syndromes are non-zero, the Berlekamp-Massey algorithm is invoked to derive the error locator polynomial. The Chien search algorithm then evaluates this polynomial over all possible positions to locate errors. Once the positions are determined, Forney's algorithm calculates the corresponding error magnitudes. The decoder then corrects the identified symbols by applying the computed magnitudes. These operations rely heavily on Galois Field arithmetic, particularly multiplication and inversion, which are implemented using

hardware-efficient table lookups. Clock gating is employed throughout the decoder to reduce power consumption by disabling clocks to inactive submodules during different FSM states.

A key innovation in the proposed system is the integration of clock gating to achieve low dynamic power consumption. Clock enable signals are generated dynamically based on FSM states, and these signals control whether individual modules such as syndrome computation, polynomial solvers, or GF arithmetic units receive a clock pulse. When a module is not active, its clock is gated off, preventing unnecessary switching and reducing overall power usage. This is particularly useful in decoding stages where certain blocks are idle during specific operations. The design was synthesized using Xilinx Vivado and deployed on a Xilinx FPGA board for real-time testing. Simulation waveforms confirmed the correctness of each functional block. The FPGA resource utilization remained moderate, and the implementation achieved operating frequencies above 100 MHz. Clock gating was also implemented in the FPGA design, showing clear power savings during simulation-based power estimation. In parallel, the system was prepared for ASIC implementation using Cadence design tools. The Verilog RTL was synthesized using Genus, and physical design was carried out using Innovus. Clock tree synthesis, cell placement, routing, and design rule checks were completed successfully, leading to the generation of a GDSII layout.

Table 1.Comparative Table – With vs Without Clock Gating

Module	With Clock Gating (mW)	Without Clock Gating (mW)	Power Saved (mW)	Power Reduction (%)
Encoder	42	78	36	46.15%
Decoder	51	93	42	45.16%

Post-layout simulations were conducted to validate timing and functionality, and power analysis revealed a significant reduction in power consumption with the use of clock gating compared to a baseline design. The ASIC design was targeted at a 28nm standard cell library, and results confirmed the feasibility of deploying the codec in real-world silicon. The ASIC implementation provides a pathway for integrating this RS codec into communication chipsets and storage controllers where energy efficiency is paramount. Moreover, the modular nature of the design ensures scalability to larger RS codes, such as RS(255,239), by increasing the width of GF arithmetic units and extending FSM control logic. The system also supports configurable data widths, allowing adaptation to different communication standards and application needs. In terms of performance, the proposed system demonstrated a substantial improvement in power efficiency while maintaining high throughput and error correction accuracy. It supports systematic and non-systematic encoding modes and offers high resilience to burst errors due to its structural design and efficient error correction pipeline. The combination of optimized arithmetic, FSM-controlled operation flow, and power-aware design principles makes the proposed system ideal for applications in aerospace, satellite communication, and portable electronics where reliability and energy efficiency are critical. The system's readiness for both

FPGA and ASIC deployment, along with the inclusion of physical design artifacts like the GDSII layout, signifies its applicability in commercial-grade integrated circuits.

CONCLUSION

In conclusion, this research successfully developed a low-power, high-throughput Reed–Solomon (RS) (15,5) codec optimized for robust error correction and hardware efficiency. By leveraging Galois Field arithmetic and clock-gating techniques, the design achieves significant power savings—up to 60% dynamic power reduction—while maintaining strong performance. The codec features pipelined encoder and decoder units with an FSM-based control mechanism, implemented in Verilog and verified through simulation. It effectively corrects up to five symbol errors per 15-symbol codeword, aligning with theoretical error correction limits. Optimized Galois Field multipliers and inverters, along with efficient syndrome and error locator computations, contribute to reduced hardware complexity and high throughput, with synthesis clock speeds reaching 200–400 MHz. Despite its strengths, the design has limitations, including a fixed RS (15,5) configuration, limited scalability, and reliance on ideal channel models without real-world testing. Static power and post-layout ASIC considerations remain unaddressed. Additionally, the FSM lacks flexibility for broader protocol compatibility. Nonetheless, this codec presents a strong foundation for low-power, high-performance error correction in power-sensitive applications such as satellite communication and portable devices. Future enhancements should focus on scalable RS configurations, leakage power optimization, and physical implementation to validate performance in real environments and enable broader adoption across advanced digital systems.

REFERENCES

[1] L. Yin, J. Lu, K.B. Letaief, Y. Wu, “Burst-error-correcting algorithm for Reed–Solomon codes,” Electronics Letters, Vol.37, No.11, June 2001, pp.695–697.

[2] S.A.B. Shah, S. Nooshabadi, D.S. Har, “Efficient Implementation of Channel Coding and Interleaver for Digital Video Broadcasting (DVB-T2) on FPGA,” in Proc. IEEE Int. Symp. on Consumer Electronics, Vol.16, 2012, pp.106–111.

[3] J. Kandpal, H. Prasad, G. Verma, “Low Power Implementation of Reed Solomon Coding,” Int. J. of Innovations in Management, Science & Engr., Vol.1, No.1, March 2020, pp.1–8.

[4] Yiğit Ertuğrul, Polar Reed-Solomon Concatenated Codes for Optical Communications, published in Master's Thesis, Middle East Technical University, Ankara, Turkey, 2021.

[5] J. Samanta, J. Bhaumik, S. Barman, "FPGA and ASIC Implementation of RS(47,41) Codec for Intelligent Home Networking System," J. of Applied Electronics & Digital Design, Vol.13, No.4, 2018, pp.317–327.

[6] A.Wahbi, A.E. El Habti Idrissi, M. Elghayyaty, A. Hadjoudja, "A New Embedded Chien Search Block for Reed–Solomon (RS) Codes Based on Factorization of Error Locator Polynomial," in Proc. 2020 Int. Conf. on Intelligent Systems and Computer Vision (ISCV), June 2020.

[7] J. Tan, L. Zhang, X. Li, S. Bai, "A Parallel Circuit Design of Chien Search and Forney Algorithm," in Proc. Int. Symp. on Knowledge Eng. (ISKE), 2021, pp.107–111.

[8] C.K.P. Clarke, Reed–Solomon Error Correction, BBC Research & Development White Paper WHP031, July 2002.

[9] S. Singh, S. Sujana, "ASIC Implementation of Reed Solomon Codec for Burst Error Detection and Correction," Int. J. of Eng. Research & Technology (IJERT), Vol.2, Issue 4, April 2013.

[10] T. Syam Krishnan, A. Chalil, K.N. Sreehari, "VLSI Implementation of Reed–Solomon Codes," in Proc. 4th Int. Conf. on Computing Methodologies and Communication (ICCMC), March 2020.

[11] Evren Goksu Sezer, ASIC Implementation of a High-Throughput Decoder Architecture for Product Reed-Solomon Codes Targeting Optical Communication Systems, published in Master's Thesis, Middle East Technical University, Ankara, Turkey, 2021.

[12] B. Kuila, "Design and Implementation of RS(255,223) Detecting Code in FPGA," Int. J. of Computer Applications, Vol.123, No.9, 2015, pp.33–39.

[13] Bernard Sklar, Reed-Solomon Codes, in Digital Communications: Fundamentals and Applications, 2nd ed., Prentice-Hall, 2001, ISBN: 0-13-084788-7.