Optimizing Delay and Area in One-Bit and Two-Bit Data Error Correction Using CRC

Gireesha K[1], Thejaswini B M[2]

Department of ECE Bangalore Institute of Technology, Bangalore, India

Abstract

In digital communication and storage systems, ensuring data integrity is crucial for reliable performance. This paper examines how Cyclic Redundancy Check (CRC) mechanisms for correcting both one-bit and two-bits in data transmissions. While CRCs are traditionally employed for error correction, this study extends their utility by introducing methods for error correction, aiming to enhance data reliability. This presents a new approach that leverages CRC-based techniques to rectify, single-bit and double-bit errors within transmitted data streams. The suggested method combines encoding and decoding capabilities of CRC using the diagonal bits, parity bits and check bits. Demonstrating significant improvements in error resilience over standard CRC implementations. Experimental findings indicate the proposed techniques offer effective solutions for error correction, with a focus on optimizing performance and accuracy. This approach primarily emphasizes optimizing the area and latency associated with error correction. The work contributes to the advancement of reliable error correction frameworks, which can be integrated into various communication and storage systems to maintain data integrity and mitigate the effects of transmission errors.

Index terms: Cyclic Redundancy Check, error correction, parity bit, diagonal bit and data transmission.

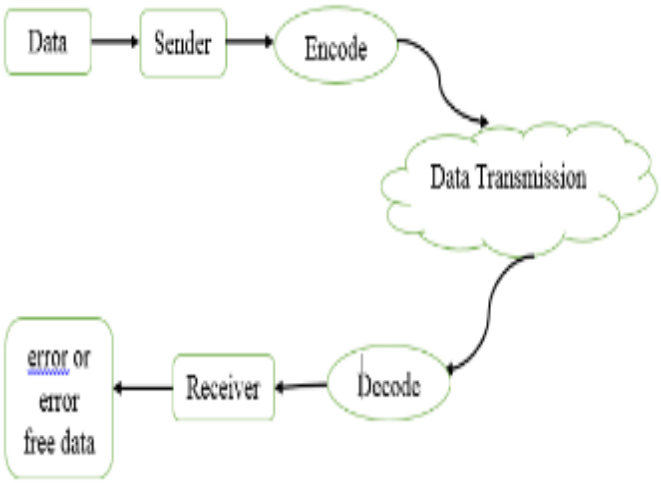
INTRODUCTION

In modern digital communication and data storage systems CRC (Cyclic Redundancy Check) is a widely used error-checking technique in computing the data communication. It's a simple and efficient method for detecting and correcting flipped bits in data transmission. CRC is commonly used in network protocols, file storage and data transmission. The CRC is extensively utilized in wireless communications to increase the reliability of data transmission between transmitters and receivers. It aids in identifying typical fault that may appear, like bits flipping caused by noise disturbance in communication channel.

It serves as a crucial tool for detecting fault at different surfaces of the protocol stack. CRCs safeguard packet headers at the physical layer. CRCs are employed to identify whether transmitted packets contain bit errors, leading to the discarding of erroneous packets. CRC checks for data transmission errors on lines at data link layer or transport layer.

CRC consist of two elements, the protected bit stream, referred as the payload or transmitted data and a generator polynomial. The generator polynomial remains constant and is determined based on specific transmission standard.CRC works by appending a fixed-length sequence of bits to the data before transmission. Sender and receiver both have a shared polynomial (called the CRC polynomial) used in the calculation. The sender computes the CRC value by xor. The data appends CRC bits using CRC polynomial and sends the entire message, including the CRC value. The receiver performs the same computation and compares the computed CRC value with the one received. If they match, the input is considered intact; if they don't match, an error is detected.

Fig.1 Basic CRC unit



Single-bit and double-bit errors frequently arise in digital communication. A one-bit error affects a single bit of data, while a two-bit error affects double bits. Addressing these errors is crucial for systems that demand for high reliability, such as computer memory, data storage devices and communication protocols. Single-bit errors affect individual bits of transmitted data and can often be managed using error detection and correction techniques like parity checks. These methods involve appending redundant bits to the data stream, enabling the receiver to detect and in some cases correct errors automatically. In contrast, double-bit errors pose greater challenges as they can lead to more substantial data corruption.

Sender side

- **Data Encoding:** The sender prepares the data to be transmitted by encoding, it includes appending the CRC value, which is calculated based on data being sent.
- **Transmission:** The encoded data, now containing the CRC is transmitted over the communication channel to the receiver.

Receiver Side

- **Data Reception:** The receiver captures the transmitted data, which includes both the original data and the appended CRC value.
- **Decoding:** The receiver processes the incoming data to extract the original content and separate it from the CRC.
- **Error Detection:** The receiver recalculates the CRC based on the received data and compares it with the received CRC value. If they match, the data is considered error-free, if not a fault had appeared during transmission.

The paper structured as below:
Section II, Briefs about the Literature Survey.
Section III, covers the foundation concept of CRC unit.
Section IV, introduces proposed model of CRC architecture.
Section V presents the results and performance evaluations of architecture.
Section VI offers the conclusion and final thoughts of the study.

Literature Survey

Vivien Boussard, Stephane Coulombe, et.al., [1] presents a new approach for multiple error correction in wireless communications over error-prone networks, utilizing the CRC syndrome with an optimized lookup table that eliminates the need for arithmetic operations. The method achieves correction performance comparable to state-of-the-art techniques while significantly reducing computational complexity. The lookup table is designed for easy navigation to correct multiple bit errors, maintaining a constant size even when

handling more than two errors, providing a substantial advantage over previous table-based methods. Simulations conducted using a C implementation on a Raspberry Pi 4 show that, this method can process single and double error corrections for large payloads in 100 ns and 642 µs, respectively, compared to 300 µs and 1.5 s required by existing CRC multiple error correction techniques. Additionally, when multiple candidate error patterns are present, numerous errors can be corrected by incorporating a checksum cross-validation step.

Niloofar Yazdani, Nikolaos Kouvelas, et.al., [2] prescribed a novel coding technique at application layer designed to increase the recovery of corrupted frames in Long Range Wide Area Networks (LoRaWAN). Due to high frame corruption in LoRaWAN caused by network coexistence and MAC layer characteristics, LoRa's Forward Error Correction (FEC) mechanism often falls short in retrieving corrupted data, with real-life measurements showing a significant portion of received transmissions being corrupted. ReDCoS aims to suggest this issue by employing lightweight coding techniques to pre-encode transmitted data, followed by computing the CRC based on this pre-encoded data. At the receiver end, both the CRC and coded data are utilized to recover information from corrupted frames beyond the built-in Error Correcting Code (ECC).

Xinmiao Zhang, Yok Jye Tang, et.al., [3] proposed a cyclic redundancy check (CRC) that is widely utilized in digital communication and storage systems to ensure data integrity. The process of CRC encoding and decoding involves linear feedback shift registers (LFSRs), with parallel LFSRs implemented using registers that undergo feedback matrix multiplication and input pre-processing matrix multiplication. Achieving high throughput, essential for modern applications, requires significant parallelism. Previous designs have simplified the complexity of parallel LFSRs by employing state transformation and adjusting the input tap. By leveraging the transformation, the pre-processing matrix in highly-parallel LFSRs can be streamlined without altering the feedback matrix. Furthermore, it is revealed that post-processing matrix multiplication in state-transformed designs can be eliminated without compromising the CRC's error detection capability.

Zeyu Han, Chongbin Xu, et.al., [4] presented a sparse Kronecker-product (SKP)coding scheme is proposed for unsourced multiple access. Specifically, the data of each active user is encoded as the Kronecker product of two component codewords with one being sparse and other being forward-error-correction (FEC) coded. At the receiver, an iterative decoding algorithm is developed, consisting of matrix factorization for the decomposition of the Kronecker product and soft-in soft-out decoding for the component sparse code and the FEC code. The CRC aided interference cancelation technique is further incorporated for performance improvement. Numerical results show that the proposed scheme outperforms the state-of the-art counterparts, and approaches the random coding.

Juhee Yun, Nulibyul Kim, et.al., [6] presents CRC codes that are utilized for error correction by toggling a few unreliable bits within a packet to convert negative acknowledgments (NAK) to acknowledgments (ACK). The challenge with employing CRC codes to correcting errors lies in the high worst-case complexity, as the number of potential error patterns is 2NUR, where NUR represents the count of unreliable bits. The concept of optimality in error pattern sets using soft decisions, focusing on fixed set sizes and unreliable bit numbers. Optimal error patterns are incrementally generated based on this optimality definition, ensuring a low worst-case complexity in the proposed error pattern estimation method known as fixed complexity CRC-aided error pattern estimation (fcCRCEPE).

Vivien Boussard, Firouzeh Golaghazadeh, et.al., [7] proposed an innovative single error correction method related to cyclic redundancy check (CRC) for robust H.264 Baseline video decoding. Unlike existing methods, this correction algorithm does not rely on lookup tables; instead, it identifies the error location using binary operations derived from the computed link layer CRC syndrome. Because multiple errors can produce the same CRC syndrome as a single error, the corrected packet is verified through a non-desynchronizing bits validation (NDBV) process, which only forwards compliant packets to the video decoder. Simulations using the H.264 Baseline profile demonstrate an average gain of 3.04 dB and 2.36 dB over the latest spatio-temporal error concealment (STBMA) and NDBV+STBMA reconstruction methods, respectively, at a residual bit error rate of 10^{-6}.

Xijin Liu, Shaohua Wu, et.al., [8] describes the novel successive cancellation list (SCL) decoding schemes for Polar codes that leverage the fault correcting capability of CRC codes. The proposed schemes aim to enhance CRC-aided Polar decoding by harnessing the potential error correction capabilities of CRC. The first scheme, CRC error-correction-aided successive cancellation list (CRC-EC-SCL) decoding, integrates a look-up-table-based CRC EC method into the Polar SCL decoding process. Building on this, the segmented-CRC-

EC-SCL decoding scheme is introduced, where the information block is grouped into segments, every group with a shorter CRC sequence for error correction. This allows for early correction of multiple erroneous bits, effectively limiting error propagation. Simulation results demonstrate a significant performance improvement with both proposed schemes.

Evgeny Tsimbalo, Xenofon Fafoutis, et.al., [9] proposed the utilization of iterative decoding algorithms for errors in CRC codes, particularly in scenarios where traditional error correction codes are impractical due to energy limitations at the transmitter. The research demonstrates how two iterative techniques, typically employed for low-density parity check (LDPC) codes—Belief Propagation (BP) and the Alternating Direction Method of Multipliers (ADMM)—can be adapted for use with a high-density parity check matrix. By conducting simulations, the study evaluates the performance of both techniques, highlighting a substantial improvement.

FOUNDATION CONCEPT OF CRC UNIT

The Cyclic Redundancy is a powerful error detection technique used in digital networks and storage systems to detect accidental changes to raw data by appending extra bits to data block before transmission. These bits are computed using a xor operation based on input bits themselves. The XOR operation is used extensively to generate the checksum.

The data which has to be transmitted or stored is divided into blocks and a mathematical algorithm is appended to each block to create the checksum. This algorithm consists, series of XOR operations between the bits and a predefined polynomial, known as the generator polynomial.

The generator polynomial is a crucial component of the CRC calculation. It determines the bits in the checksum and plays a significant role in error detection capabilities of CRC. After completing all the XOR operations, we obtain the CRC checksum. This checksum is then appended to initial data block, resulting in the transmitted or stored data.

The sender appends the computed CRC checksum to the info before transmission. The receiver performs the same polynomial operation on the received data and compares the calculated CRC checksum with the one received. If they match, the data supposed to be intact; if not, an error is detected.

CRC is efficient in detecting errors such as single-bit errors, burst errors, and most random errors. CRC algorithms can be applied in hardware (for speed and efficiency) and software (for flexibility and compatibility). In essence, CRC error detection relies on polynomial mathematics to generate and verify checksums that can reliably detect errors while transmission or storage of digital data.

XOR is a operation that requires two inputs and produces an output of true (1) if the inputs are different, and false (0) if they are the same. In terms of binary numbers, it can thought of as a bit-wise operation where each corresponding bit of two bits is compared. The XOR operation is commutative, meaning that the order of the operands does not affect the result.

X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	0

Table 1: XOR truth table

Advantages of CRC:

- Simplicity
- High Error Detection Probability
- Efficiency
- Compatibility
- Real-time Error Detection

Applications of CRC:

- Data Communication
- Data Storage
- Network Protocols
- Wireless Communications
- File Transfer Protocols
- Disk Image Verification

PROPOSED MODEL OF CRC

In the Suggested model, one and two bit error correction, the encryption and decryption are part of an error correction and detection. The encoder is answerable for transforming the input data into a coded format, while the decoder performs the reverse operation, converting the encoded information revers into its original form. The breaking down of 16-bit input data into four different groups Wi, Xi, Yi and Zi. These groups form the basis of subsequent operations. The encoder then arranges these groups into a matrix form.

W1	W2	W3	W4
X1	X2	X3	X4
Y1	Y2	Y3	Y4
Z1	Z2	Z3	Z4

In the Encoding the following bits are calculated before transmission from the sender. The Diagonal bits is obtained by performing the xor operation of the matrix above,

D1 → W1 ^ X2 ^ Y1 ^ Z2
D2 → W2 ^ X1 ^ Y2 ^ Z1
D3 → W3 ^ X4 ^ Y3 ^ Z4
D4 → W4 ^ X3 ^ Y4 ^ Z3

Parity bits are computed to ensure that the number of bits set to 1 is even (or odd, depending on the parity scheme). Each parity bit is calculated by performing XOR operations across the bits in each column of the matrix. It helps in detecting any errors in each group of bits.

P1 → W1 ^ X1 ^ Y1 ^ Z1
P2 → W2 ^ X2 ^ Y2 ^ Z2
P3 → W3 ^ X3 ^ Y3 ^ Z3
P4 → W4 ^ X4 ^ Y4 ^ Z4

Check bits are designed to provide additional redundancy by XORing bits from different rows and columns. These check bits are crucial for verifying the integrity of each group and identifying potential errors.

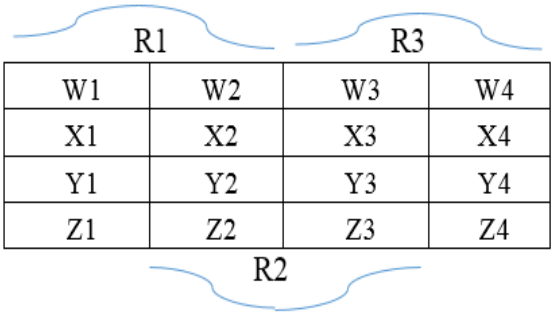
Cw13 → W1 ^ W3
Cw24 → W2 ^ W4
Cx13 → X1 ^ X3
Cx24 → X2 ^ X4
Cy13 → Y1 ^ Y3
Cy24 → Y2 ^ Y4
Cz13 → Z1 ^ Z3
Cz24 → Z2 ^ Z4

Both the encoder and decoder operate as combinational logic circuits, implying their actions are solely determined by current input states, without reliance on previous states or memory elements. This inherent characteristic makes combinational circuits ideal for tasks requiring immediate responses to input changes, a crucial aspect in error correction and detection.
In the decoding process the Syndrome value is calculated for the diagonal, parity and check bits. The Syndrome is performed by performing the xor operation with the redundancy bits.

$$SDi \rightarrow Di \wedge RDi$$
$$SPi \rightarrow Pi \wedge RPi$$
$$SCi \rightarrow Ci \wedge RCi$$

Where, i = 1,2,3,4 and
R = Redundancy bit

Armed with information about errors, the decoder strategically selects regions within the coded message that are affected. Applying correction mechanisms specific to these regions, the decoder aims to rectify errors and restore the integrity of the data.
The region selection criteria for error bits are determined by analyzing the Syndrome Diagonal (SD) bits and Syndrome Parity (SP) bits. The purpose of these criteria is to identify the most appropriate region where errors might be occurring, based on the sum of specific syndrome values.



At the decoder the region is selected for the error correction, that in which position the bits are flipped. Based on the regions the errors are corrected.

Regions	Selection Criteria
Region 1	SD1 + SD2 +SP1 + SP2 > SD3 + SD4 + SP3 + SP4
Region 2	SD1 + SD2 +SP1 + SP2 < SD3 + SD4 + SP3 + SP4
Region 3	SD1 + SD2 +SP1 + SP2 = SD3 + SD4 + SP3 + SP4

The selection of regions based on these criteria is critical for optimizing the overall efficiency and reliability of the system. By dynamically choosing the most appropriate region, the system can adapt to varying conditions, ensuring that it operates at optimal performance levels while minimizing errors and delays. This approach is particularly valuable in high-stakes environments such as space engineering, where precise timing and power management are crucial for mission success.
The ultimate goal of decoding process is to recover the original 16-bit data. By eliminating the redundancy bits added during encoding, the decoder meticulously reconstructs the initial data, ensuring its fidelity to the information before encoding.

Results and Performance Evaluations
The developed CRC design consists of design entry, simulation and synthesis. The come up with architecture

has been coded in Verilog language and verified their functionality using using Vivado tool. Once the functional verification is done, the RTL model is taken to the synthesis, where the optimized netlist is viewed and Area, power, timing is obtained.

Our proposed CRC unit is designed for delay & area efficiency. It utilizes a diagonal, parity and check bits where the encoding and decoding scheme of data is performed. This reduces the area by directly reducing the delay. The design also incorporates the xor operation of bits.

Consequently, the processed output of delay and area is balanced to referenced papers. The simulated and synthesized results are as follows...

Simulation Result:

Fig.2 Encoder waveform

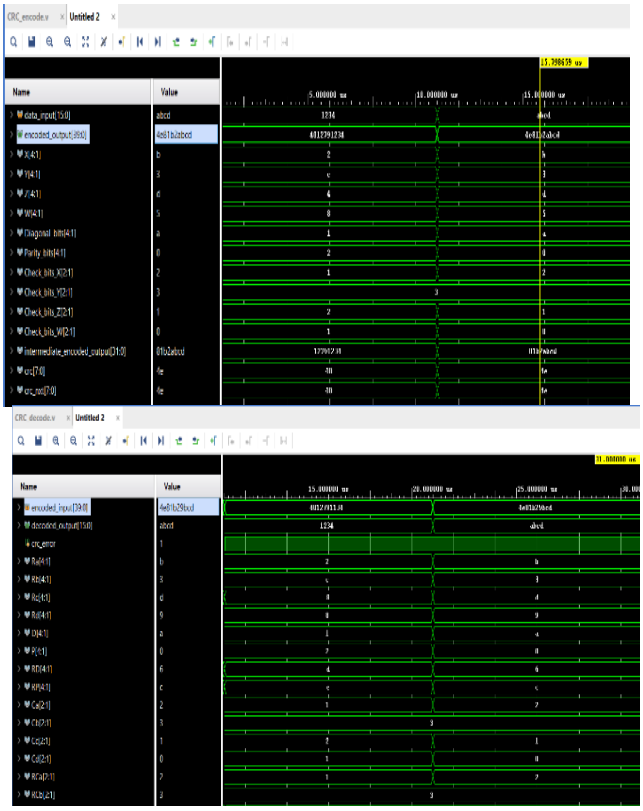


Fig.3 Decoder waveform

Synthesis Result:

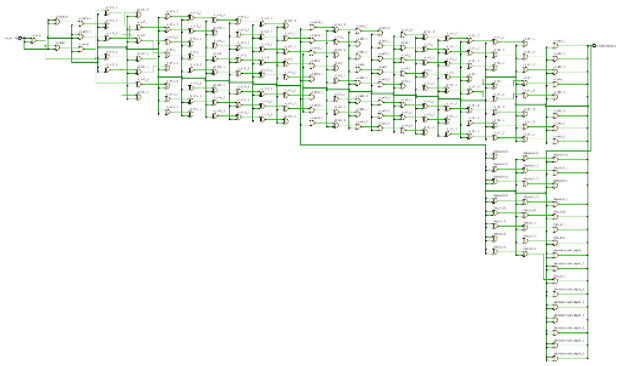
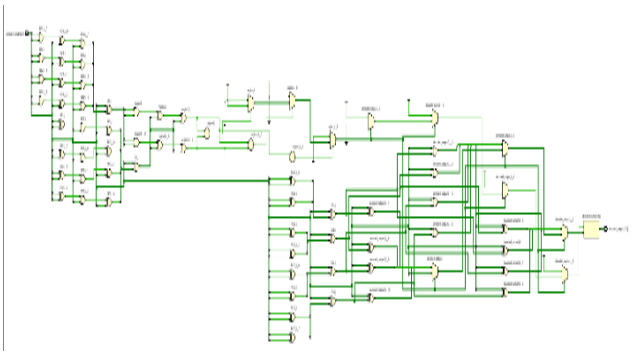


Fig.4 Encoder Synthesized circuit

Fig.5 Decoder Synthesized circuit



Timing Report:

Fig.6 Encoder Timing report

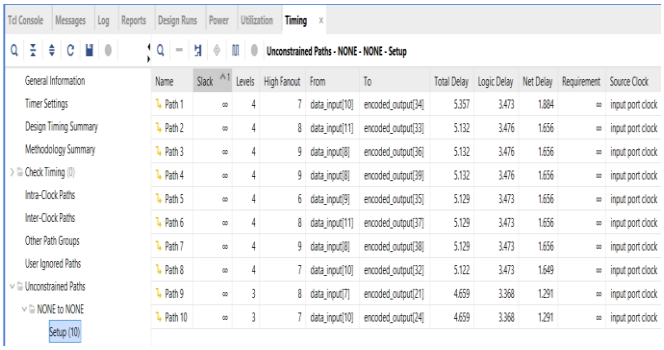
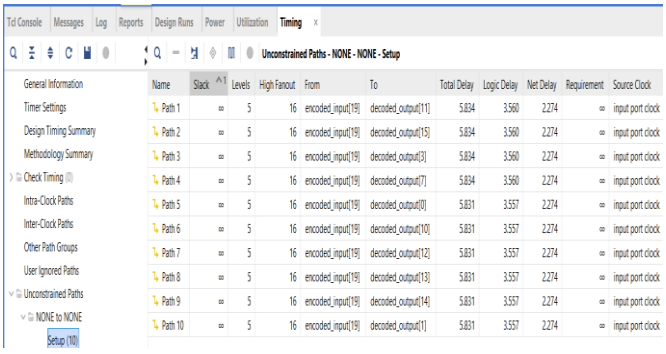


Fig.7 Decoder Timing report



Area Report:

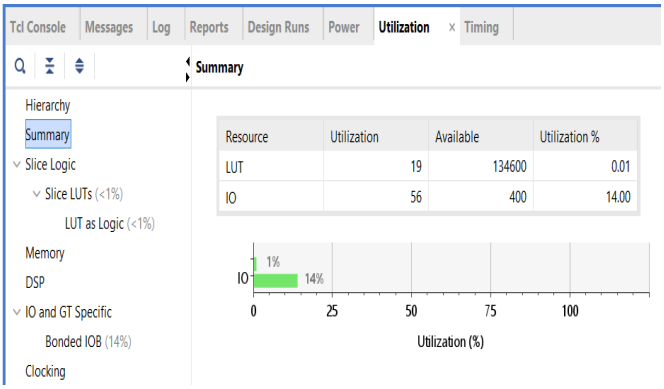


Fig.8 Encoder Area report

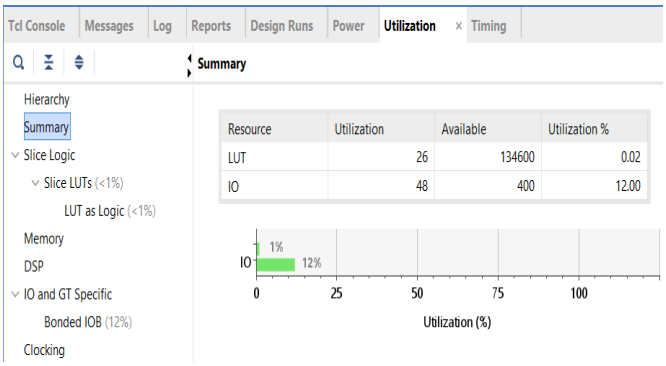


Fig.9 Decoder Area report

Area & Delay of Encoder

	Encoder
Area (%)	14.01
Delay (ns)	5.1

Area & Delay of Decoder

	Decoder
Area (%)	12.02
Delay (ns)	5.8

Area and Delay Report Comparison

	Proposed work	Ref [1]	Ref [5]	Ref [11]
Year	2024	2022	2021	2013
Area (%)	14.01	___	25.5	___
Delay (ns)	5.1	100	___	13.7us

Conclusion

The proposed CRC architecture is highly effective in phrase of both delay and space. The design effectively combines an encoder and decoder in the information transmission, each optimized for their respective. The entire design is coded in Verilog and verified using the Vivado tool, demonstrates the proposed CRC unit outperforms traditional designs as referenced in other papers. Synthesis provided the optimized netlists that confirm these improvements. Overall, the proposed CRC architecture demonstrates a superior balance of delay & area efficiency, validated by both simulation and synthesized results, making it a valuable advancement in the CRC unit designs.

In future the CRC unit can be implemented by utilizing additional coding schemes and redundancy to detect and locate the errors position in the data, so that can be useful for the data transmission with multiple errors.

REFERENCES

1. Vivien Boussard, Stephane Coulombe, "CRC-Based Correction of Multiple Errors Using an Optimized Lookup Table" IEEE Access, vol.10, pp 23931-23947, 2022.
2. Niloofar Yazdani, Nikolaos Kouvelas, R Venkatesha Prasad, "Energy Efficient Data Recovery from Corrupted LoRa Frames" IEEE Global Communication Conference, 2021.
3. Xinmiao Zhang and Yok Jye Tang, "Low-Complexity Parallel Cyclic Redundancy Check" IEEE International Symposium on Circuits and Systems, 2021.
4. Zeyu Han, Chongbin Xu, "Sparse Kronecker-Product Coding For Unsourced Multiple Access" IEEE Wireless Communications Letters, Vol. 10, Pp 2274 – 2278, 2021.
5. Alvaro Cintas Canto and Reza Azarderakhsh, "CRC-Based Error Detection Constructions for FLT and ITA Finite Field Inversions Over GF(2^m)" IEEE Transactions on VLSI, vol-29, pp 1033-1037, 2021.
6. Juhee Yun, Nulibyul Kim, "CRC-Aided Fixed Complexity Error Pattern Estimation Technique" IEEE Access, vol.10, pp 59048-59056, 2020.
7. Vivien Boussard and Patrick Corlay, "Robust H.264 Video Decoding using CRC-Based Single Error correction and Non-Desynchronizing Bits Validation" IEEE, PP 1098-1102, 2020.
8. Xijin Liu, Shaohua Wu, "Improved Polar SCL Decoding By Exploiting the Error Correction Capability of CRC" IEEE Access, Vol. 7, Pp 7032 – 7040, 2019.
9. Evgeny Tsimbalo, Xenofon Fafoutis, "CRC Error Correction for Energy-Constrained Transmission" 26th International Symposium on Personal, Indoor and Mobile Radio Communications, pp 430 – 434, 2015.
10. Juhee Yun and Jaekwon Kim, "Fixed Complexity Error Pattern Estimation" International Conference on Computing, Networking and Communications, pp 968 – 971, 2015.
11. Sheng-Shih Wang and Ting-Rong, "CPR: A CRC-Based Packet Recovery Mechanism for Wireless Networks" IEEE Wireless Communication and Networking Conference, pp 321-326, 2013.
12. Yanbin Zhang, "Error Correction Application of CRC in the RFID System" IEEE International Conference, pp 443 – 446, 2011.
13. Wen-Kang and Yaw-Chung, "A Novel CRC Based Error Correction Scheme in OFDM/OFDMA Wireless Networks", ISITA, Taichung, Taiwan, pp 668-672, 2010.
14. Shahram Babaie and Ahmad Khadem Zadeh, "Double Bits Error Correction Using CRC Method", Fifth International Conference on Semantics, pp 254-257, 2019
15. Xiaodong Deng, Tao Liu, Mengtian Rong, "Segmented Cyclic Redundancy Check: a Data Protection Scheme for Fast Reading RFID Tag's Memory" IEEE Communications Society subject matter, pp 1576 – 1581, 20