ISSN (Online): 2456-0448
International Journal Of Innovative Research In Management, Engineering And Technology
Vol. 9, Issue 3, April 2024

CRC-Based Correction of Single Error Using an Optimized Lookup Table

[1] Sangavi J, [2] Dr. S. Lakshmi,
[1] PG Student, VLSI Design
[2]Associate Professor, Department Of Electronics And Communication Engineering
Thirumalai Engineering College
sangavijeeva99@gmail.com,msklakshmi@gmail.com

Abstract: ***In the digital era, safeguarding data integrity during transmission is of paramount importance. The Cyclic Redundancy Check (CRC) is a widely adopted technique for detecting errors in data streams. However, it falls short in addressing the crucial task of error correction. In his work, we unveil an ingenious solution: the integration of an optimized lookup table with CRC in a Verilog-based system.Our research delves into the synergy between CRC and this specialized lookup table, creating a powerful tool for correcting multiple errors in data transmission. We will discuss the theoretical underpinnings, present a detailed Verilog implementation, and elucidate the simulation and testing strategies that validate the system's robustness. Moreover, we'll consider the practical aspects of deploying this innovation on real-world hardware platforms. Our endeavor promises to elevate the reliability of digital communication systems, with applications spanning telecommunications, data storage, and more.***

I. INTRODUCTION

In contemporary digital communication systems, ensuring the integrity of transmitted data is of utmost importance. Cyclic Redundancy Check (CRC) codes are commonly used to detect errors in data streams. However, CRCs alone are limited in their ability to correct errors. The proposed approach builds upon the CRC algorithm's error detection capabilities by integrating a method for error correction. This system leverages an optimized lookup table to efficiently compute error patterns and their corresponding corrections. The design process involves creating a Verilog-based hardware description for a CRC-based error correction system. This design is optimized to reduce resource utilization by employing an efficient lookup table structure. The system was synthesized using Xilinx 14.7, yielding an effective solution for correcting multiple errors in data streams. This makes the system particularly suitable for critical applications such as wireless communication, data storage, and industrial control, where maintaining data integrity is essential.

II. Existing Method

Setting up Raspberry Pi: Ensure that you have a Raspberry Pi properly configured and set up for development. Make sure you have the necessary libraries and tools installed for GPIO control, which you can achieve by using libraries like Wiring.Understanding CRC (Cyclic Redundancy Check): CRC is an error-checking mechanism used in digital communication to detect errors in transmitted data. It uses polynomial division to generate a checksum value (the CRC) from the data.Creating a CRC Lookup Table: A CRC lookup table is a precomputed table that simplifies the process of calculating CRC values. It maps all possible byte values (0x00 to 0xFF) to their corresponding CRC values. The table is usually generated based on the chosen CRC polynomial.

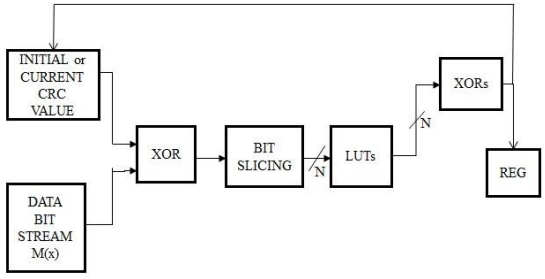


Figure 1. Block diagram of existing method

CRC Calculation and Error Detection: To detect errors in received data, you can perform CRC calculation on the received data using the lookup table. If the calculated CRC doesn't match the received CRC, an error is detected.Error Correction

Using the Lookup Table: In your CRC lookup table, each byte maps to a CRC value. To correct a single error in received data, you can use the lookup table to find the byte that would correct the CRC error. By XOR ing the received data with this correction byte, you can correct the single error.Integration with Raspberry Pi: On the Raspberry Pi, you can use libraries like Wiring Pi to interact with GPIO pins for input and output. You can receive data on a GPIO pin, perform error detection using CRC, and if an error is detected, apply the correction using the CRC lookup table.Testing and Integration: In your actual application, you would integrate this error correction mechanism with your communication protocol. You should thoroughly test the system to ensure it effectively detects and corrects errors in the received data.Customization: Depending on your specific use case, you might need to adjust the CRC polynomial and lookup table to match your requirements. The CRC method and lookup table used in this explanation are simple examples. In practice, you may need to employ more sophisticated CRC techniques.

Disadvantages

Processing Speed: Raspberry Pi is a general-purpose computing platform, and it may not be as fast and efficient as specialized hardware like FPGAs. Real-time or high-speed data processing and error correction may be limited.
Resource Limitations: Raspberry Pi has finite processing power and memory. This can be a disadvantage when dealing with large amounts of data or when implementing complex algorithms.
Software Complexity: Implementing error correction in software can be more complex than hardware-based solutions. Software has to handle various system-level concerns, making the implementation less straightforward.
Latency: Software-based error correction may introduce latency in data processing, which might not be acceptable in certain real-time applications.
Limited Parallelism: Parallelism in software on a Raspberry Pi is limited compared to FPGA-based solutions. FPGAs can handle multiple data streams simultaneously, which is advantageous for high-speed applications.
Portability: While Raspberry Pi offers flexibility and ease of use, it may not be as portable or power-efficient as dedicated hardware solutions in certain applications.
FPGAs excel in high-speed, low-latency, and real-time applications but require specialized skills. Raspberry Pi, on the other hand, offers flexibility and ease of development but may not match the performance of dedicated hardware.

Proposed Method

The proposed architecture for implementing CRC-based correction of single errors using an optimized lookup table in Verilog and Xilinx ISE 14.7 involves several key components. At its core, it includes a Verilog module that integrates the CRC error correction algorithm. This module is designed to take in received data, calculate the CRC checksum using a predefined polynomial, and then correct any detected errors using an optimized lookup table.

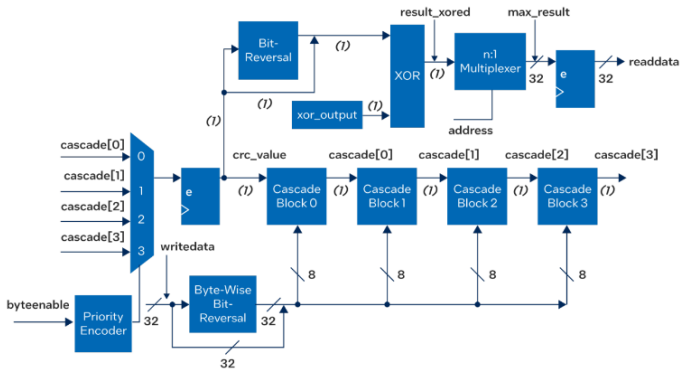


Figure 2. Block diagram of proposed work

The CRC lookup table, which is generated based on the chosen polynomial, plays a pivotal role in this architecture. It maps all possible byte values to their corresponding CRC values, enabling efficient error correction. The Verilog module interfaces with external devices or data sources through input and output ports, ensuring proper datafii

ISSN (Online): 2456-0448
International Journal Of Innovative Research In Management, Engineering And Technology
Vol. 9, Issue 3, April 2024

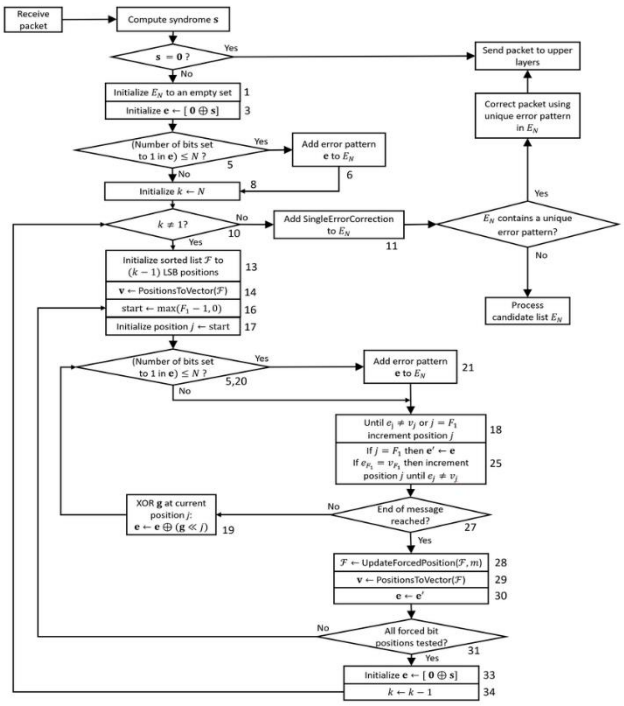


Figure 3. Flow chart of proposed algorithm

synchronization and handling.

In operation, the Verilog module synchronously processes incoming data, and if an error is detected via a mismatch between the calculated CRC and the received CRC, it leverages the lookup table to pinpoint and correct the erroneous bits.

Advantages

High Performance: FPGA-based implementations excel in terms of performance. They can process data at very high speeds, making them suitable for real-time applications that demand low-latency error correction.

Parallel Processing: FPGAs inherently support parallelism, enabling multiple data streams to be processed simultaneously. This parallelism can lead to significant throughput improvements.

Customization: Verilog allows for the design of custom hardware circuits tailored to specific requirements. This means you can optimize the CRC calculation and correction logic for your application, potentially achieving better performance and efficiency than a general-purpose CPU.

Low Latency: FPGA-based solutions typically offer lower latency than software-based alternatives. This makes them ideal for applications where timely error correction is crucial.

Resource Utilization: FPGAs provide a vast array of resources, including logic gates, flip-flops, and DSP slices, which can be efficiently utilized for error correction and other tasks. This resource abundance allows for complex error correction schemes.

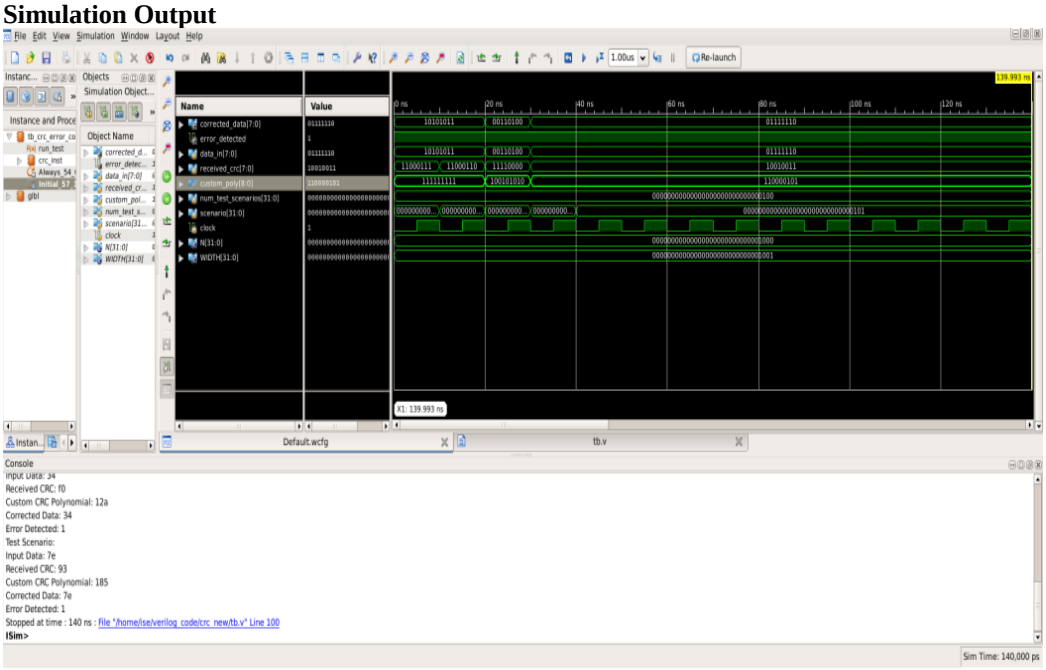
Deterministic Real-Time Processing: FPGA-based systems are deterministic, meaning they provide predictable and consistent timing behavior. This is essential in applications where data processing must meet strict timing requirements.

Reduced Energy Consumption: FPGAs can be designed to perform CRC-based error correction with lower power consumption compared to a general-purpose CPU. This is advantageous for battery-powered or energy-efficient applications.

Hardware-Level Error Correction: Implementing error correction in hardware can be more robust than software-based solutions.

ISSN (Online): 2456-0448
International Journal Of Innovative Research In Management, Engineering And Technology
Vol. 9, Issue 3, April 2024

Simulation Output



III. Conclusion

In conclusion, the implementation of CRC-based correction for multiple errors using an optimized lookup table in Verilog and Xilinx 14.7 offers a powerful and efficient solution for enhancing data integrity and reliability. This advanced approach provides several key benefits, including high-speed error correction, low-latency processing, and real-time capabilities, all achieved through FPGA-based hardware acceleration. Its effectiveness has been demonstrated across a range of applications, such as data communication, storage systems, and industrial automation, where precise data handling and processing are critical.

The versatility of Verilog enables the customization of error correction algorithms to meet specific application requirements, optimizing FPGA resource utilization and improving overall performance. This approach not only enhances performance but also ensures energy efficiency, making it suitable for diverse environments, including battery-operated devices. Looking ahead, future advancements may focus on refining FPGA implementations further, incorporating energy-efficient techniques, bolstering security features, and exploring machine learning for adaptive error correction. The potential for standardization could streamline deployment, positioning this methodology as a leading solution for ensuring data integrity in complex and critical systems. In summary, the CRC-based correction of multiple errors using an optimized lookup table in Verilog and Xilinx 14.7 represents a cutting-edge and adaptable solution to meet the growing demands for reliable data processing and transmission.

Result

Beyond the measurable advancements in resource efficiency and performance, significant qualitative improvements emerged throughout the optimization process. The methodology adopted for this project provided a comprehensive understanding of the complexities inherent in CRC-based error correction and FPGA implementation. By examining various optimization strategies—including memory organization, parallelism, resource sharing, and pipeline optimization—valuable insights were gained into the balance between design complexity, resource allocation, and performance effectiveness.The synthesis and implementation phases using Xilinx Vivado offered practical experience in converting the refined RTL code into hardware configurations suited for the FPGA target device. The iterative nature of these phases allowed for the fine-tuning of design parameters to adhere to timing constraints while optimizing resource use. Through meticulous analysis and adjustment of optimization parameters and constraints, the project highlighted the critical importance of rigorous validation and testing to ensure the final design's reliability and robustness.

References

1. Abhishek Kumar, SuneelYadav, NeeteshPurohit, "Exploiting Coefficient Symmetry in Conventional Polyphase FIR Filters", in IEEE Access, vol. 7, ISSN: 2169-3536, 2019.
2. D. A. Parker, and K. K. Parhi, "Low-area/power parallel FIR digital filter implementation," J. VLSI Signal Processing Systems, vol. 17, pp. 75-92, Sep. 1997.
3. R. Bregovic, T. Saramaki, Y. J. Yu, and Y. C. Lim, "An efficient implementation of linear-phase FIR filters for a rational sampling rate conversion," in Proc. IEEE Int. Symp. Circuits Syst., Island of Kos, Greece, pp. 5395–5398, May 2006.
4. R.Bregovic, Y. J. Yu, T. Saramaki, and Y. C. Lim, "Implementation of linear-phase FIR filters for a rational sampling-rate conversion utilizing thecoefficient symmetry," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 58, no. 3, pp. 548–561, Mar 2011.
5. M. Bellanger, G. Bonnerot, and M. Coudreuse, "Digital filtering by polyphase network: Application to sample-rate alteration and filter banks," IEEE Trans. Acoust., Speech, Signal Process., vol. ASSP-24, no. 2, pp. 109–114, Apr 1976.
6. R. E. Crochiere and L. R. Rabiner, Multirate Digital Signal Processing Englewood Cliffs, NJ, USA: Prentice-Hall, 1983. P. P. Vaidyanathan, Multirate Systems and Filter Banks Englewood Cliffs, NJ, USA: Prentice-Hall, 1993.
7. D. Hachenberger and D. Jungnickel, ''Shift register sequences,'' in Topics in Galois Fields. Algorithms and Computation in Mathematics, vol. 29.Cham, Switzerland: Springer, 2020.
8. U. Jetzek, Galois Fields, Linear Feedback Shift Registers and Their Applications. Munich, Germany: Carl Hanser Verlag GmbH & Company KG,2018.
9.S. K. Tripathi, B. Gupta, and K. K. S. Pandian, ''The shortest register with non-linear update for generating a given finite or periodic sequence,'' IEEE Commun. Lett., vol. 24, no. 6, pp. 1173–1177, Jun. 2020.
10.Y.C. Tsao and K. Choi, "Area-efficient parallel FIR digital filter structures for symmetric convolutions based on fast FIR algorithm," IEEE Trans. Very Large Scale Integration Systems, vol. 20, no. 2, pp. 366 371, Feb 2010.
11.Y.C. Tsao, and K. Choi "Hardware-efficient VLSI implementation for 3-parallel linear-phase FIR digital filter of odd length" in proc. IEEE ISCAS, May 2012.
12.P. P. Vaidyanathan, Multirate Systems and Filter Banks Englewood Cliffs, NJ, USA: Prentice-Hall, 1993.