International Journal of Research and Innovation in Multidisciplinary (www.ijrim.com)
|| Volume 2, Issue 2, February 2025 ||

Review of Cyclic Redundancy Check Code for Finite Field Multiplier on FPGA

[1]Nisha Jatav, [2]Prof. Dipti Malviya

[1]M.Tech Scholar, [2]Assistant Professor
Department of Electronics and Communication Engineering,
Sagar Institute of Science & Technology, Bhopal, India

***Abstract*— Cyclic Redundancy Check (CRC) is a widely used error detection technique in digital communication and data storage systems, relying on polynomial division over Finite Fields (Galois Fields, GF) for checksum generation. The implementation of CRC using Finite Field Multipliers (FFMs) on Field-Programmable Gate Arrays (FPGAs) enables high-speed computation with improved efficiency, making it suitable for real-time applications such as networking, wireless communication, and embedded systems. This paper reviews various VLSI-based architectures for CRC implementation, focusing on different multiplier designs, including serial, parallel, and hybrid approaches, along with optimization techniques such as pipeline processing, resource-sharing, and low-power methodologies.**

***Keywords*— CRC, Error, Correction, Speed, Power, Latency.**

1. INTRODUCTION

Error detection plays a critical role in digital communication and storage systems, ensuring that transmitted and stored data remains accurate and free from corruption. Among various error-detecting techniques, Cyclic Redundancy Check (CRC) has gained prominence due to its ability to detect random and burst errors efficiently. CRC is widely used in applications such as wireless communication, networking protocols (Ethernet, 5G, Wi-Fi), storage devices, and embedded systems. The fundamental operation of CRC is based on polynomial division over Finite Fields (Galois Fields, GF), where a transmitted data block is divided by a predetermined generator polynomial to produce a checksum. This checksum is appended to the data and used at the receiver end to verify data integrity. The simplicity and effectiveness of CRC make it a widely adopted method for error detection in high-speed systems.

A key challenge in CRC computation is ensuring high-speed execution while minimizing hardware resources. Traditional software-based CRC implementations introduce latency, making them unsuitable for real-time applications. To overcome this limitation, hardware implementations using Finite Field Multipliers (FFMs) have been developed. FFMs enable efficient polynomial multiplication within Galois Fields, providing a foundation for high-speed CRC calculations. The VLSI (Very Large Scale Integration) architecture of CRC using FFMs is essential for optimizing performance, particularly in systems where real-time error detection is crucial. The choice of an efficient Finite Field Multiplier architecture significantly impacts CRC computation speed, area consumption, and power efficiency.

Field-Programmable Gate Arrays (FPGAs) have emerged as an ideal platform for implementing CRC architectures due to their inherent parallel processing capabilities, flexibility, and reconfigurability. Unlike ASICs, which provide fixed-function implementations, FPGAs allow designers to implement and modify CRC architectures dynamically, making them highly suitable for applications requiring adaptability. FPGA-based CRC designs can leverage parallel and pipeline processing techniques to achieve low-latency checksum computation. Furthermore, the integration of Finite Field Arithmetic in FPGA implementations enhances performance by enabling efficient multiplication and division operations in Galois Fields.

Despite the advantages of FPGA-based CRC implementations, several challenges must be addressed to achieve optimal performance. The

primary trade-offs involve area utilization, power consumption, and computational speed. Different architectural approaches, such as serial, parallel, and hybrid multipliers, have been explored to balance these factors. Serial multipliers consume fewer hardware resources but suffer from lower throughput, while parallel multipliers provide high-speed computation at the cost of increased area. Hybrid architectures aim to achieve an optimal balance by leveraging resource-sharing techniques and pipeline structures. The design choice is heavily influenced by application-specific requirements, including the desired CRC polynomial length, error detection capability, and system constraints.

The choice of CRC polynomial significantly affects error detection efficiency. Various standards, such as CRC-8, CRC-16, CRC-32, and CRC-64, are used in different applications, each offering varying levels of error detection capability. The implementation of FPGA-based CRC architectures must consider the adaptability of these polynomials to support multiple protocols. Adaptive CRC architectures capable of switching between different polynomial configurations are increasingly being explored to enhance flexibility in modern digital systems. Additionally, low-power CRC designs are gaining attention, particularly for energy-efficient applications such as IoT devices and embedded systems.

This review focuses on the latest advancements in VLSI-based CRC architectures for Finite Field Multipliers on FPGA, analyzing different design methodologies, optimization techniques, and real-world applications. The study explores how hardware-efficient multipliers, pipeline processing, and resource-sharing techniques contribute to improved CRC computation. Additionally, it examines recent research trends, including low-power architectures, fault-tolerant CRC implementations, and dynamically reconfigurable designs. The discussion also includes an overview of benchmarking FPGA implementations of CRC, comparing performance metrics such as throughput, area utilization, and energy efficiency.

FPGA-based CRC implementations using Finite Field Multipliers provide a powerful solution for high-speed error detection in digital communication and storage systems. However, achieving an optimal balance between speed, power, and resource utilization remains a critical challenge. Future research in this area should focus on developing more efficient multiplier architectures, adaptive CRC designs, and low-power hardware implementations to enhance the performance of FPGA-based CRC architectures. By leveraging the latest advancements in VLSI design, FPGA architectures, and Finite Field Arithmetic, researchers and engineers can significantly improve the efficiency and reliability of error detection mechanisms in next-generation digital systems.

2. LITERATURE REVIEW

A. Cintas-Canto et al., [1] In order to safeguard cryptographic schemes against both accidental and intentional mistakes, fault detection is gaining significant importance. These designs often make use of finite fields over GF(2m) due to the binary coding of their data for practical reasons. Due to its complexity, multiplication is often the bottleneck operation for many cryptosystems among the various finite field arithmetic. Hence, this study derives fault detection algorithms for finite field multipliers based on cyclic codes that use various fields from both classical and modern cryptography. In addition, we embed these methods into the original designs to conduct a thorough investigation, compare the various overheads, and demonstrate their applicability to embedded systems with severe constraints. Very good error coverage with acceptable overhead is achieved by these implementations on advanced micro devices (AMD)/Xilinx field-programmable gate array (FPGA).

N. N. Qaqos [2] Data storage, communication frameworks, and networking condition domains all rely heavily on Cyclic Redundancy Check (CRC) to identify errors. One of the biggest challenges nowadays is improving the speed of data transmission while simultaneously making better use of available technology. Therefore, the implementation of the framework becomes slowed down by CRC computation. Each of the CRC5 and CRC8 frameworks—used in USB token bundles and ATM standards, respectively—will be structured and implemented in this work. To achieve high throughput data with improved

equipment assets, the suggested CRC engineering for both the CRC encoder and decoder frameworks use an equal pipelining technique. In contrast to previous efforts, the suggested architecture does not use an F-grid or a Look-Up Table (LUT) to store pre-determined CRC properties. The code for the framework was written in Very High-Speed Integrated Circuit Description Language (VHDL), and it was organised and executed using an Austere 3E FPGA processor. Using the Xilinx ISE 9.2i simulator, we functionally duplicate and validate all of the suggested engineering.

Bajarangbali [3] the third In order to satisfy the Ethernet speed limitation, this study details the structure and improvement of an altered CRC computation that is implemented on an FPGA. The algorithm makes use of a decreased lookup table. This computation may be used for data of any length by sequentially processing it in 16-byte squares. There can be less than 16 bytes in the final square. In order to handle a 16-byte square input, the computation begins by building an enhanced table with pre-determined CRC. A lookup is performed in this table according to the input data, and the results of the lookups are merged using XOR operations to generate the final CRC of the input data. The Ethernet data that needs its CRC computed is produced in 128-bit squares at a 312.5 MHz clock repetition in order to achieve a 40Gbps performance.

R. O. S. Juan [4] The CAN standard has a self-correcting mechanism known as Cyclic Redundancy Check (CRC) code to detect and fix errors. Using an optional error correcting technique known as the Hamming code in lieu of the traditional CRC code is the primary objective of this computation. Not to mention, maybe speed up the framework's CAN. From the start-of-frame (SOF) to the control bit frames, the bit floods and the locations of the repeated bits 'r' are determined. In order to register for the necessary r, these bits will be sent into the excess bit controller. The locations of the extra bits are decided by modulo-2 operation, and they are of intensity 2. The suggested method is coordinated by use of a Xilinx Virtex-5 FPGA. Compared to other options for CAN error detection and correction, the simulation results demonstrate a significant improvement in CAN's frame rate while simultaneously reducing the bits packing payload.

Z. Shen [5] An important sub-indicator of the Dark Matter Particle Explorer (DAMPE) is the BGO calorimeter, which provides a broad estimate scope of the necessary huge beam range. A 16-bit Actel ProASIC is used in the calorimeter's readout circuitry. The plan level flip-flops and inserted square random access memory (Slam) of a Streak based field-programmable gate array (FPGA) are also susceptible to single event upset (SEU) under harsh space conditions. The heavy-ion pillar test dissects and tries out SEU mitigation strategies, such as imperfect triple modular redundancy (TMR), CRC checksum, and multi-domain reset, in order to fulfil radiation hardness assurance (RHA). The readout electronics are designed with staggered redundancy in mind and implemented using an FPGA plan that has SEU resilience and minimal asset consumption.

P. Mathew, [6] In accordance with the IEEE 802.15.6-2012 specification, this paper showcases the hardware implementation of a 2.4 GHz narrowband physical layer for a wireless body area network (WBAN). Individually designed and incorporated were the key building components of the PHY handset, such as the CRC, spreader, interleaver, and scrambler. The proposal incorporates a BCH (63, 51, 2) encoder and decoder to counteract the information transfer constraint and provide better, more consistent quality, which is especially important for restoration applications. Pi/2 DBPSK modulation is supported by the spreading technique, which is used before to modulation. The strategy is assessed in terms of its effectiveness and impact.

According to Z. Shen [7] A logical satellite to monitor high-vitamina huge beams in space is being built as the Dark Matter Particle Explorer (DAMPE). The Bismuth Germanium Oxide (BGO) calorimeter, a key detector of DAMPE, measures the deposition of particle vitality, differentiates between hadron foundation positrons, electrons and gamma rays, and provides trigger information. The electrical components are expected to be able to withstand radiation since the satellite is designed to sail in a low Earth orbit at an altitude of 500 km. The Actel ProASCI Streak-

Based Field-Programmable Gate Array The Front-End Electronics (Expense) control segment is chosen to be in addition to (APA). At Lanzhou's Heavy Ion Research Facility (HIRFL), the SEE test was conducted to evaluate the radiation blockage. Despite a convincing LET of 90 MeV-cm2/mg, the chip failed to display latchup. No SEL could compromise the APA Blaze switches, which provide nonvolatile, changeable interface routing and programming nets. The registers created from logic tiles and Slam squares were sensitive to SEU, however, as SEE analysis also showed. So, it's important to find the right ways to reduce the effects of SEU and make sure the chip works OK in a radiation environment.

According to C. Chen [8] Despite some serious security concerns, bank IC cards are now widely used across the globe, particularly in Asia and Europe. In this study, we offer a new approach to transport security for bank IC cards by combining two techniques to protect AMBA (Progressed Microcontroller Transport Engineering), the protocol used to link the 32-bit central processing unit (CPU) to the memories or cryptographic computations within the system on a chip (SoC) construction. Two countermeasures are used here: first, a power balancing mechanism that uses 8B/10B encoding on the transport to prevent power investigations; second, a CRC check mechanism that balances issue assaults on the transport. The FPGA board has verified and duplicated both tactics. The results of the FPGA verification demonstrate that the 8B/10B decipher significantly reduces the intensity quality, and that the CRC can successfully identify 99.4 percent of the faults introduced into the transport.

I. M. Safarulla [9] Radiation may cause transient problems known as single-event upsets (SEUs) or soft errors, and FPGAs based on SRAM are vulnerable to them. In contrast to hard errors, which permanently damage the device, soft errors affect or alter just some logical circumstances of memory components. Any component of static memory may have its reasoning conditions changed or configuration memory altered by a SEU lawfully. Any SRAM-based FPGA with embedded soft centre processors may implement another problem detection framework architecture. Using the Shortcoming tolerant Configuration Engine and the Hamming approach, adaptability to internal failure may be achieved after the defective centre is discovered. The PicoBlaze-based Shortcoming Tolerant Configuration engine finds the source of the problem using the CRC approach, fixes it using frame-based reconfiguration, and makes it flaw-tolerant with the help of triple modular redundancy (TMR). The SEC finds and fixes single bit mistakes using the hammering approach. Shortcoming recovery utilising CRB brought both centres back into harmony. Xilinx ISE 13.2 is used for integration, while ISim is used to replicate the code written in VHDL.

In their work, T. Závodník et al., [10] provide a new approach to handle the calculation of CRC functions, which are often used to check for bit errors in binary data. The CRCs are also applicable to this technology, which is designed for generic hashing in FPGA. Applications that use equal inputs of fixed estimation and need high throughput, such hash tables, are good candidates for the method. So that our method doesn't use any LUTs, it makes use of the DSP squares found in modern FPGAs to do all the necessary XOR operations. To reduce the amount of DSP squares needed for the calculation, it suggests a heuristic based on Monte Carlo methods. Based on our experimental findings, one DSP square with 48 XOR operations may replace about eleven 6-input LUTs.

3. CHALLENGES

1. Hardware Resource Utilization – FPGA implementations must balance speed and area efficiency, as high-performance CRC architectures often require significant hardware resources, increasing FPGA logic utilization.

2. Power Consumption – High-speed CRC computations can lead to excessive power consumption, making it challenging to design energy-efficient architectures, especially for battery-powered or IoT applications.

3. Latency and Throughput Trade-offs – While parallel multipliers improve speed, they also increase hardware complexity. Achieving low-latency, high-throughput CRC computation

without excessive FPGA resource usage remains a challenge.

4. Polynomial Selection and Adaptability – Different applications require different CRC polynomials (e.g., CRC-8, CRC-16, CRC-32). Implementing adaptive architectures that can dynamically support multiple polynomials without compromising efficiency is complex.

5. Finite Field Multiplier Optimization – Efficient multiplication in Galois Fields (GF) is computationally intensive, requiring specialized architectures that minimize redundant operations and logic depth to enhance performance.

6. Pipeline and Parallelism Constraints – Although pipeline and parallel architectures enhance speed, they also introduce design complexity in terms of synchronization, routing congestion, and timing closure issues on FPGA.

7. Fault Tolerance and Reliability – CRC architectures must be resilient against hardware faults, soft errors, and radiation effects, particularly in safety-critical applications like aerospace, automotive, and industrial automation.

8. Scalability and Future-Proofing – With evolving FPGA architectures and CRC requirements, designing scalable and reconfigurable architectures that remain efficient across multiple FPGA generations is an ongoing challenge.

4. CONCLUSION

CRC architectures using Finite Field Multipliers on FPGA is a crucial advancement in error detection for high-speed digital systems. This review highlights various VLSI-based multiplier designs and their trade-offs in terms of performance, area efficiency, and power consumption. FPGA-based CRC implementations benefit from parallelism, reconfigurability, and low-latency computation, making them ideal for modern communication and storage applications. The study also emphasizes the significance of adaptive CRC architectures, which allow dynamic switching between different polynomial configurations to support multiple protocols. Despite advancements, challenges remain in optimizing hardware complexity, power efficiency, and fault tolerance. Future research should focus on developing more efficient multiplier architectures, low-power FPGA implementations, and AI-driven error detection techniques to further enhance CRC performance in next-generation digital communication systems.

REFERENCES

1. A. Cintas-Canto, M. M. Kermani and R. Azarderakhsh, "Reliable Architectures for Finite Field Multipliers Using Cyclic Codes on FPGA Utilized in Classic and Post-Quantum Cryptography," in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 31, no. 1, pp. 157-161, Jan. 2023, doi: 10.1109/TVLSI.2022.3224357.
2. N. N. Qaqos, "Optimized FPGA Implementation of the CRC Using Parallel Pipelining Architecture," *2019 International Conference on Advanced Science and Engineering (ICOASE)*, Zakho - Duhok, Iraq, 2019, pp. 46-51.
3. Bajarangbali and P. A. Anand, "Design of high speed CRC algorithm for ethernet on FPGA using reduced lookup table algorithm," *2022 IEEE Annual India Conference (INDICON)*, Bangalore, 2022, pp. 1-6.
4. R. O. S. Juan, M. W. Jeong, H. W. Cha and H. S. Kim, "FPGA implementation of hamming code for increasing the frame rate of CAN communication," *2021 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*, Jeju, 2021, pp. 684-687.
5. Z. Shen *et al.*, "Study on FPGA SEU Mitigation for the Readout Electronics of DAMPE BGO Calorimeter in Space," in *IEEE Transactions on Nuclear Science*, vol. 62, no. 3, pp. 1010-1015, June 2020.
6. P. Mathew, L. Augustine, D. Kushwaha, V. Desalphine and A. David Selvakumar, "Implementation of NB PHY transceiver of IEEE 802.15.6 WBAN on FPGA," *2018 International Conference on VLSI Systems, Architecture, Technology and Applications (VLSI-SATA)*, Bangalore, 2018, pp. 1-6.

7. Z. Shen *et al.*, "Study on FPGA SEU mitigation for readout electronics of DAMPE BGO calorimeter," *2017 19th IEEE-NPSS Real Time Conference*, Nara, 2017, pp. 1-1.
8. C. Chen, S. You, L. Wu and X. Zhang, "A novel bus security solution for bank IC card with FPGA," *2016 12th IEEE International Conference on Solid-State and Integrated Circuit Technology (ICSICT)*, Guilin, 2016, pp. 1-3.
9. I. M. Safarulla and K. Manilal, "Design of Soft error tolerance technique for FPGA based soft core processors," *2015 IEEE International Conference on Advanced Communications, Control and Computing Technologies*, Ramanathapuram, 2015,
10. T. Závodník, L. Kekely and V. Puš, "CRC based hashing in FPGA using DSP blocks," *17th International Symposium on Design and Diagnostics of Electronic Circuits & Systems*, Warsaw, 2014, pp. 179-182.