International Journal of Engineering Research and Reviews ISSN 2348-697X (Online)
Vol. 12, Issue 3, pp: (116-149), Month: July - September 2024, Available at: www.researchpublish.com

A Review Paper on Error Detection and Correction of FSO using FPGA

Rudraunsh Joshi[1], Chinchu Josheph[2], Dr A.A. Bazil Raj[2]

[1] Electronics and Telecommunication Department, Pune Vidyarthi Griha's College of Engineering and Technology and G.K. Patel (Wani) Institute of Management, Vidyanagari, Parvati, Pune, Maharashtra, India, 411009

[2] RF Photonics Laboratory, Department of Electronics Engineering, Defence Institute of Advanced Technology, Girinagar, Pune, Maharashtra, India, 411025

DOI: https://doi.org/10.5281/zenodo.13848647

Published Date: 27-September-2024

Abstract: **Future Free Space Optical (FSO) communication systems have the potential to communicate data at very high rates with very high levels of integrity over distances of up to a few kilometers (for terrestrial links). This technology has also been a candidate for high-speed and highly reliable (BER ~10^-9) communication links between satellites in geosynchronous orbits and ground stations. Since the free space optical medium can induce many forms of distortion (atmospheric turbulence effects, optical beam wander, etc.), the use of a channel code to detect and correct errors during the process of information transfer over the channel is essential. A correctly designed channel code can reduce the raw BER from unacceptable values to values that can be tolerated in many applications. Different kinds of error-correcting codes are used for FSO like Hamming Code, Reed-Solomon Codes, Turbo, and LDPC Codes.**

Keywords: **FSO (free space optics), FPGA (field programmable gate array), BER (Bit error rate), FEC (forward error correction).**

I. INTRODUCTION

Free Space Optical (FSO) [1][2] communication is a wireless technology that uses light to transmit data through air or space which is a free space medium transmission system. Instead of using conventional radio waves as transmission signals for wireless communication, this technology used lasers, infrared as well as ultraviolet signals. This is a kind of open-space wireless communication system (OSWC) [3] . Fig.1 shows the basic block diagram of a Free Space Optical Communication System.

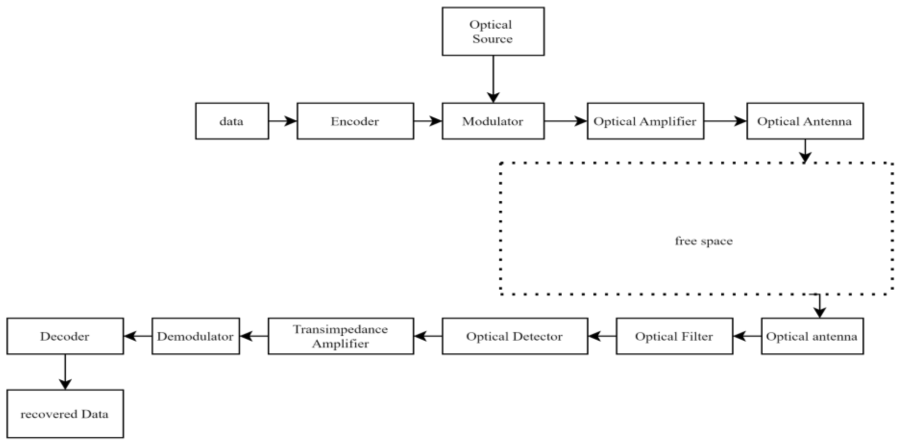


Fig.1 Basic Block Diagram of FSOC system

FSO communications refers to a line-of-sight technology transmitting modulated visible or infrared (IR) beams through the atmosphere to establish optical communications. In some cases, FSO may also use non-line-of-sight (NLOS) technology utilizing scattering or reflectivity [4]. Free-space optical (FSO) communication systems, both in space and within the atmosphere, have been developed to meet the growing demand for high-speed and highly secure communication networks[5]. These systems are particularly valuable for establishing links between various platforms, such as satellites,deep-space probes, ground stations, unmanned aerial vehicles (UAVs), high-altitude platforms (HAPs), aircraft, and other mobile communication partners[. FSO technology is versatile and can be applied in both military and civilian contexts, offering a promising solution for net-centric connectivity. Its advantages in terms of bandwidth, spectrum usage, and security make it an attractive supplement to traditional radio frequency (RF) communications [6].

While FSO links that connect fixed points, such as buildings, have been well-established and are now a staple in local and metropolitan area networks, the application of FSO technology in mobile and long-range scenarios presents unique challenges[7]. The narrow divergence of optical beams in FSO systems necessitates extreme precision in pointing and tracking, which can be difficult to achieve over long distances or in mobile environments. Overcoming these challenges is crucial for fully harnessing the potential of optical links. Additionally, long-distance FSO links that transmit through the atmosphere often experience significant signal fading due to turbulence in the air's refractive index and blockages caused by clouds, snow, and rain, all of which can impact communication reliability [8][9].

FSO communications have many advantages, they typically operate in the unlicensed Tera-Hertz spectrum bands and provide significant improvement in signal bandwidth over RF [10]. FSO channel is immune to electromagnetic interference (EMI) and is secure with a low probability of interception and low probability of detection (LPI/LPD). Some of the other advantages of FSO systems include ease of installation, low costs per bit ratio, and protocol independence to support multiple platforms and interfaces. Due to these advantages, FSO has therefore been considered one of the most viable technologies for next-generation broadband communication and wireless networks. The greatest disadvantage of FSO is the weather. When the communication link includes part of the atmosphere, clear-air turbulence, and possible boundary-layer turbulence (for example, one end of the communication link is on an aircraft) cause serious phase distortions and fading to the link. FSO networks may also suffer because of insufficient availability and high vulnerability to weather turbulence. Weather conditions such as fog, smog, snow, rain, dust particles, etc. affect FSO communications performance. Atmospheric electricity between clouds and between clouds and the earth’s surface can build up lightning flashes with the duration of each flash being between 20–130 msec [4]. Optical communication systems can modulate the light carrier's frequency, amplitude, phase, and polarization. Among these, the most commonly used modulation techniques are amplitude modulation with direct detection and phase modulation combined with homodyne or heterodyne receivers, mainly because of their simpler implementation.

The advancements in Free Space Optical (FSO) communication technology are driven by the need for high-speed, reliable data transmission in various applications[11][12]. Recent studies highlight the importance of robust tracking mechanisms, such as mono-pulse techniques, which enhance alignment accuracy and maintain stable communication links, particularly under dynamic conditions [13][14]. These innovations are crucial for achieving high data rates and reliability in environments prone to atmospheric disturbances[15][16]. Additionally, the integration of localized weather data into predictive models significantly improves the accuracy of atmospheric attenuation predictions, enabling FSO systems to adapt effectively to changing conditions [18].

Research has also focused on mitigating challenges posed by atmospheric turbulence and environmental factors that can disrupt optical signal propagation [19][20]. Techniques such as neural control for beam wandering [21] mitigation and cognitive decision-making controllers for adaptive beam steering have shown promise in enhancing the performance of FSO systems [22]. Furthermore, experimental studies on the effects of specific weather phenomena, like sandstorms [23], provide valuable insights into optimizing FSO communication links in challenging environments. Collectively, these efforts contribute to a deeper understanding of FSO technology and its applications, paving the way for more reliable and efficient optical communication systems [24][25].

The simplest form of digital modulation is amplitude-shift keying (2-ASK), which is known as on-off keying (OOK) in optical systems [26]. In OOK, the light source is switched on to represent a binary "1" and switched off to represent a binary "0." This is a type of intensity modulation. The most basic version of this scheme is called non-return-to-zero (NRZ)-OOK. Another popular coding method is return-to-zero (RZ), which offers better sensitivity and includes the clock frequency within the modulation spectrum. However, both NRZ and RZ can lose clock synchronization if long sequences of ones or zeros are transmitted. This issue can be addressed with coding schemes like Manchester coding, which is a variant of RZ that enables easy clock recovery. The trade-off is that these methods require twice the bandwidth compared to NRZ to satisfy the Nyquist–Shannon theorem, although it has been demonstrated that RZ can also work with the same bandwidth as NRZ, offering around 2 dB more sensitivity [27][28].

Line codes, such as 8B10B, are often used to maintain a constant average signal level, which is crucial for the operation of optical amplifiers and to avoid issues like inter-symbol interference. In fiber systems and free-space optical (FSO) systems, 8B10B coding on top of NRZ requires only about 25% more bandwidth than NRZ alone. It also ensures frequent level changes in the signal, making it easier to recover the clock even with long sequences of identical bits [29].

For On-Off Keying (OOK), the precise wavelength and phase of the carrier signal are not essential for accurate demodulation. The receiver simply detects the incoming optical power and compares it to a threshold. However, OOK is sensitive to amplitude distortions (fading) and signal propagation through different paths, though the latter is less of a concern under clear-sky conditions. Atmospheric conditions like clouds can significantly attenuate the signal, but this is less of an issue for FSO systems in clear weather.

Coherent modulation schemes, such as binary phase-shift keying (BPSK), are also used in optical communications. In BPSK, the phase of the laser light is shifted between two states. Coherent receivers work by combining the received light with a local oscillator. Alternatively, differential phase-shift keying (DPSK) can be used, which is less sensitive than BPSK and doesn't require the precise optical phase-locked loop needed in BPSK systems. While coherent receivers offer sensitivity that is one to two orders of magnitude better than OOK, they come with increased system complexity and are more vulnerable to phase distortions caused by atmospheric propagation. While OOK systems are simpler and more robust against atmospheric distortions compared to coherent systems, coherent systems offer better sensitivity. OOK has been widely used in optical fiber communications due to its low complexity and the availability of reliable, cost-effective components, making it a preferred choice for optical links within the atmosphere [30] [31].

Despite having so many perks, FSO systems are highly susceptible to atmospheric conditions like fog, rain, snow, dust, and turbulence, which can scatter or absorb the optical signal. As a result, we need to use some error correction codes along with the encoder to help transfer the data through FSO. The error in the data is measured by Bit Error Ratio (BER) [32], it simply gives the amount of faulty bits amongst the data. Forward Error Correction (FEC) codes are used to detect and correct errors so that the original data can be recovered at the receiving end. The performance of Hamming code, Low-Density Parity Check (LDPC) code, and Turbo code are some of the popular FECs. These codes can be implemented using FPGA technology [33].

Free-space optical communication (FSOC) links offer several attractive features, such as high data rates, enhanced security, reduced payload sizes, immunity to interference and jamming, and the advantage of not requiring RF spectrum licenses. Despite these benefits, the performance of outdoor optical links over long distances can be significantly affected by atmospheric turbulence, which disrupts the transmitted laser beam. When an optical signal passes through atmospheric turbulence, it experiences considerable fluctuations in amplitude, known as scintillation. This phenomenon is often measured using the scintillation index (SI), defined as:

$$\boldsymbol{\sigma_I^2=\langle I^2\rangle/\langle I\rangle^2-1} \quad \mathbf{(1)}$$

where I represent the optical irradiance and ⟨·⟩ denotes the ensemble average. The temporal behavior of the optical channel, which is characterized by its coherence time τc, is also crucial in understanding signal variations. The bandwidth of scintillation is highly sensitive to crosswinds, leading to significant variability. Additionally, the intensity fluctuations caused by scintillation can be described by statistical distributions such as log-normal or Gamma-Gamma models [35] [36]. These factors depend on several variables, including the distance of the link, the strength of turbulence, the geometry of the transmitter and receiver, height above ground, and local weather conditions. As a result, the variability introduced by atmospheric turbulence adds complexity to both the design and performance testing of laser communication systems [37]. To mitigate the negative impact of atmospheric turbulence on Free-Space Optic Communication (FSOC) links, several strategies are used. These include optical methods like adaptive optics, which adjust the optical wavefront in real time to compensate for distortions, and diversity techniques that use multiple transmitters or receivers to enhance signal robustness. Additionally, communication strategies such as temporal diversity, which involves varying the transmission timing, can help reduce data loss caused by scintillation-induced fades, thereby improving link reliability.

FEC enhances communication reliability by adding redundancy to the transmitted data, allowing the receiver to correct errors without needing to request retransmission [38]. For optical channels, the average fade duration is typically much longer than the time between individual bits [39]. By interleaving data — spreading bits from a single data block across multiple transmission periods — error bursts can be dispersed among several blocks. This increases the chances of successful error correction and enhances the overall efficacy of the FEC [34][40].

II. IMPORTANCE OF ERROR DETECTION AND CORRECTION IN FSO

As we know the vulnerability of FSO systems to atmospheric conditions like fog, rain, snow, dust, and turbulence results in some errors in the data at some level[41]. To overcome this problem we use some error detection and correction algorithms to obtain the desired speed, accuracy, and reliability. Free Space Optical (FSO) communication systems, which rely on light to transmit data through the air, are increasingly recognized for their potential to provide high-speed, high-capacity data links without the need for physical cables. However, the efficiency and reliability of FSO systems are significantly challenged by environmental factors such as atmospheric interference. This interference includes phenomena like fog, rain, dust, and atmospheric turbulence, which can drastically affect the quality of the optical signal [42]. For instance, fog and mist can scatter light due to the water droplets in the air, which are often comparable in size to the wavelength of the optical signal. This scattering can cause significant attenuation, leading to potential data loss. Similarly, rain and snow can absorb and scatter the optical signal, while dust and smoke in the atmosphere can further degrade signal quality by causing additional scattering. Atmospheric turbulence, caused by variations in temperature and pressure, can induce random fluctuations in the FSO beam's path, known as beam wander, and intensity, known as scintillation, leading to phase distortion and signal fading [43][44].

To combat these challenges, error detection and correction techniques are indispensable. These techniques involve the use of advanced error correction codes (ECC) such as Reed-Solomon codes and low-density. Parity-Check (LDPC) codes, and Turbo codes, which are specifically designed to detect and correct errors that occur due to these atmospheric disturbances. Reed-Solomon codes, for example, are particularly effective in correcting burst errors, which are common in FSO systems due to short-term atmospheric changes. LDPC codes provide near-optimal performance in noisy environments, making them ideal for maintaining data integrity even when the error probability is high. Turbo codes, known for their iterative decoding process, offer excellent error correction capabilities, particularly in applications that demand high data integrity, such as satellite communications.

In high-speed data transmission scenarios, which FSO systems are well-suited for, even minor errors can have significant consequences. For example, in financial transactions, where timing and accuracy are critical, an error in data transmission could result in costly mistakes. In military communications, the security and reliability of the communication link are paramount, and any data corruption could have serious implications. In medical imaging, accurate data transmission is essential for correct diagnoses, and errors could lead to incorrect treatment decisions. Error correction techniques are therefore vital in these contexts, ensuring that transmitted data remains accurate and reliable [45].

Moreover, error correction is essential in enhancing the performance of long-distance FSO links, such as those used in satellite-to-ground or inter-satellite communications. In these applications, the signal can degrade significantly over long distances due to atmospheric attenuation and beam divergence. Forward Error Correction (FEC) techniques, which encode additional redundancy into the data stream, allow the receiver to detect and correct errors without needing retransmission, which is particularly crucial in scenarios where retransmission would introduce unacceptable delays or latency. Adaptive modulation and coding schemes can also be employed to optimize the balance between data rate and reliability, adjusting in real time to the current channel conditions.

In addition to improving data integrity and reliability, error correction techniques play a key role in reducing the bit error rate (BER) of FSO systems. A low BER is critical for applications that require high reliability and precision, such as in defense, aerospace, and high-frequency trading systems [46]. Convolutional codes and Hybrid Automatic Repeat Request (HARQ) mechanisms are commonly used to lower BER by correcting errors in continuous data streams and combining error correction coding with retransmission requests when necessary [47]. This not only ensures data accuracy but also enhances the overall reliability of the communication link. From an operational perspective, effective error correction contributes to the cost-effectiveness of FSO systems by reducing the need for retransmissions, which can consume additional bandwidth and power. By reducing the frequency of retransmissions, error correction helps maintain the efficiency of the FSO system, leading to lower operational costs. Furthermore, FSO systems that incorporate robust error correction require less maintenance and experience less downtime, as they are more resilient to errors caused by environmental factors. This results in higher system availability and further cost savings in the long term [48].

In critical and sensitive applications, such as military or financial communications, where the security, precision, and reliability of data transmission are non-negotiable, error detection and correction techniques are indispensable. FSO technology, with its narrow and directed optical beam, already offers a highly secure communication channel that is difficult to intercept. However, the reliability of this communication is further enhanced by robust error correction codes that ensure the integrity of the data even in challenging environments. In financial transactions, where errors can lead to significant financial losses, error correction ensures that the transmitted data is both fast and error-free, reducing the risk of costly mistakes. In scientific and medical data transmission, where accuracy is critical, error correction techniques are essential to maintaining the integrity of the data, ensuring that it remains accurate and reliable throughout the transmission process.

As FSO technology continues to evolve, the role of advanced error detection and correction methods will become even more critical [39]. These methods will support the development of faster, more reliable, and more secure communication networks, enabling FSO systems to meet the growing demand for high-speed, high-capacity data transmission across a wide range of applications. Whether for terrestrial data transmission, satellite communications, or critical infrastructure, robust error correction will remain a cornerstone of FSO technology, ensuring its viability and effectiveness in the face of environmental challenges and the ever-increasing need for reliable communication [49]

III. COMMON ERROR SOURCES IN FSO SYSTEMS:-

1. *Atmospheric Turbulence*

Atmospheric turbulence is one of the most significant challenges faced by Free Space Optics (FSO) communication systems. It occurs when there are rapid fluctuations in the refractive index of the air, primarily due to temperature variations and wind currents [50][51]. This turbulence can lead to two main phenomena: scintillation and beam wander. Scintillation is characterized by rapid, random variations in the intensity of the received optical signal, much like the twinkling of stars seen from Earth[52]. This causes fading and can significantly reduce signal quality, leading to errors in data transmission. Meanwhile, beam wander occurs when the optical beam deviates from its intended path due to varying atmospheric conditions [53] This can result in the beam partially or entirely missing the receiver, causing intermittent signal loss. The erratic nature of atmospheric turbulence makes it a particularly challenging problem to address, as it requires dynamic adjustments to maintain a stable communication link [49][54]

2. *Absorption and Scattering*

Absorption and scattering are additional atmospheric challenges that can significantly degrade the performance of FSO systems. Fog, rain, and snow are particularly detrimental, as water droplets absorb and scatter the optical signal, leading to signal attenuation. In the presence of dense fog, for example, the optical signal's power can be reduced dramatically, increasing the bit error rate (BER) and potentially leading to communication failure. Aerosols such as dust, smoke, and pollution particles can also scatter the optical beam, causing a portion of the signal to be lost and further degrading the quality of the received data. These effects are highly dependent on weather conditions and environmental factors, making FSO systems vulnerable to sudden and unpredictable changes that can severely impact performance.

3. *Misalignment*

FSO systems rely on precise alignment between the transmitter and receiver to maintain a clear line of sight for the optical signal. Pointing errors can occur due to mechanical vibrations, thermal expansion, or equipment shifts, causing the beam to deviate from the receiver, resulting in signal loss and increased error rates. In urban environments, where FSO systems are often mounted on tall buildings, building sway caused by wind or seismic activity can also lead to misalignment. This sway can cause the optical beam to drift away from the receiver, further degrading the signal.

Maintaining proper alignment is crucial for the effective operation of FSO systems, and even small deviations can lead to significant performance degradation.

4. *Background Light Interference*

Interference from background light is another common source of error in FSO systems. Solar interference, caused by direct sunlight or reflections from nearby surfaces, can introduce noise into the FSO receiver, making it more difficult to detect the intended optical signal. This can increase the BER and reduce overall system performance, particularly during daylight hours or in locations with high levels of reflective surfaces. Artificial light sources, such as streetlights, vehicle headlights, and other urban lighting, can also contribute to background noise, complicating signal detection further. The presence of such light interference necessitates the use of sophisticated filtering and signal-processing techniques to minimize its impact on FSO communications.

5. *Multipath Fading*

Multipath fading is an error phenomenon that arises when optical signals reflect off surfaces or are refracted by atmospheric layers, resulting in multiple versions of the signal arriving at the receiver at different times. This effect, known as multipath propagation, can cause signal distortion and interfere with the receiver's ability to accurately reconstruct the transmitted data. The overlapping of these delayed signals can increase the error rate and degrade communication quality. Multipath fading is particularly problematic in environments with many reflective surfaces, such as urban areas or near bodies of water, where reflections can significantly impact signal integrity.

6. *Signal Blockage*

Physical obstructions can completely block the optical path between the FSO transmitter and receiver, leading to a total loss of communication. Physical obstacles such as trees, buildings, or birds can disrupt the line of sight, causing a sudden interruption in the signal. Even temporary obstructions, like moving vehicles or people, can cause intermittent signal interruptions. The reliance on a clear line of sight makes FSO systems particularly vulnerable to such blockages, and even brief interruptions can significantly affect data transmission, particularly in high-speed communication scenarios.

7. *Weather Conditions*

Adverse weather conditions pose a significant challenge to FSO systems. Fog is particularly problematic, as it consists of tiny water droplets that scatter and absorb light, causing severe attenuation and potential loss of signal. Rain and snow also contribute to signal degradation by absorbing and scattering the optical beam, further weakening the signal strength and increasing the likelihood of errors. In hot, arid environments, heat waves can induce air turbulence, distorting the optical signal and causing fading. These weather-induced errors are often unpredictable and can vary greatly in intensity, making it difficult to maintain a stable communication link under adverse conditions.

8. *Laser Safety Regulations*

FSO systems must comply with strict laser safety regulations to ensure that the emitted light does not pose a hazard to humans or animals. These regulations often limit the power of the transmitted laser, which in turn can restrict the effective range of the FSO system. Lower power levels mean that the signal is more susceptible to attenuation and absorption in the atmosphere, particularly over long distances or in adverse weather conditions. This can make it more challenging to maintain a reliable communication link, particularly in environments where high data rates are required.

9. *Chromatic Dispersion*

Chromatic dispersion is a phenomenon that occurs in FSO systems, especially those utilizing Wavelength Division Multiplexing (WDM), where different wavelengths of light travel at different speeds through the atmosphere. This can cause signal spreading and intersymbol interference (ISI), where signals overlap and interfere with each other, leading to increased error rates.

Chromatic dispersion is particularly challenging to manage because it can vary with atmospheric conditions and wavelength, requiring sophisticated compensation techniques to mitigate its effects.

10. *Alignment and Maintenance Issues*

Maintaining optimal performance in FSO systems requires regular alignment and maintenance. Over time, environmental factors such as temperature changes, wind, and physical vibrations can cause the alignment of the optical components to drift, necessitating recalibration to ensure a clear line of sight. Additionally, routine maintenance is necessary to keep optical components free from contaminants such as dirt, dust, or moisture, which can degrade signal quality and increase error rates. Failure to perform regular maintenance can result in a gradual decline in system performance, leading to higher bit error rates and reduced reliability.

Field Programmable Gate Arrays (FPGAs) are powerful, flexible, and reconfigurable integrated circuits that have found widespread use in various applications, particularly in signal processing and communication systems [55]. Unlike traditional processors or application-specific integrated circuits (ASICs), FPGAs offer a unique architecture that can be tailored to perform specific tasks efficiently. This ability to reconfigure hardware on the fly makes FPGAs a compelling choice for applications requiring high performance, adaptability, and real-time processing capabilities [56].

FPGAs are a type of semiconductor device that consists of an array of programmable logic blocks, interconnects, and memory elements. Unlike a CPU, which executes software instructions in a fixed, sequential manner, or an ASIC, which is custom-built for a particular function, an FPGA can be programmed to implement any digital circuit. The key advantage of FPGAs is their reconfigurability: engineers can modify the FPGA configuration post-deployment to perform different functions or optimize for new requirements. This reconfigurability is particularly valuable in fields like signal processing and communication systems, where algorithms and standards frequently evolve [56] [57].

IV. ROLE OF FPGA IN ERROR CORRECTION

Field-Programmable Gate Arrays (FPGAs) are revolutionizing the world of parallel processing with their reconfigurable architecture. Unlike traditional processors that execute instructions sequentially, FPGAs harness the power of an array of configurable logic blocks (CLBs) interconnected by programmable routing channels. These logic blocks can be programmed to perform a wide range of tasks simultaneously, providing true parallelism that accelerates processing speed [58]. The inherent flexibility of FPGAs allows engineers to create customized data paths and processing units tailored to specific applications, optimizing performance through hardware description languages (HDLs) like VHDL or Verilog. By designing hardware that directly implements algorithms in parallel, FPGAs can outshine traditional processors in tasks demanding high throughput and low latency [59].

One of the most remarkable aspects of FPGAs is their ability to implement pipeline designs, where different stages of a task are processed simultaneously across various parts of the chip [60]. This approach contrasts with sequential processing, where each stage must be completed before the next begins, leading to bottlenecks and reduced efficiency. FPGAs can also host multiple independent modules that operate concurrently, including digital signal processing (DSP) blocks, memory controllers, communication interfaces, and custom logic circuits. These modules work together in parallel to handle complex tasks, with the programmable interconnect structure enabling seamless data flow without significant latency. This capability is crucial in applications requiring real-time processing, such as video encoding/decoding, high-frequency trading, and autonomous vehicle systems, where decisions must be made in microseconds.[61]

FPGAs shine in high-speed communication systems, such as those found in Free Space Optics (FSO) or satellite communications, due to their ability to handle significant data throughput and perform real-time processing. These systems often operate at data rates reaching several gigabits per second, requiring fast and efficient error correction to maintain data integrity without slowing down the overall system [62]. FPGAs are well-suited for this task because they can be configured to execute error correction algorithms with minimal latency, ensuring that data processing keeps pace with high transmission speeds. This is crucial in applications like satellite communications, where delays can have serious consequences, such as loss of data or communication link failures. The real-time processing capability of FPGAs ensures that errors are detected and corrected on the fly, providing continuous, reliable data transmission even in challenging conditions [63].

One of the significant advantages of using FPGAs in communication systems is their ability to optimize the use of hardware resources [64]. FPGAs allow designers to implement complex error correction algorithms in a way that maximizes the efficiency of logic gates, memory blocks, and other resources on the chip. This efficiency is particularly important in environments where space, power consumption, and cost are critical factors, such as in satellite systems or remote sensing equipment [65]. By carefully designing the FPGA architecture, engineers can balance the need for high performance with the constraints of power and space [66]. This capability is essential in energy-sensitive applications, where excessive power consumption can reduce the lifespan of a satellite or increase operational costs [66].

FPGAs offer seamless integration with other components of communication systems, such as modulators, demodulators, and signal processors, creating a cohesive and efficient system for data transmission. This integration is vital for minimizing delays and improving overall system performance, as it allows for the smooth flow of data from reception to error correction to retransmission. For example, in a satellite communication system, the FPGA can be integrated with the modulator and demodulator to perform real-time error correction as data is received and prepared for retransmission. This ensures that the data is corrected before it is sent to the next stage, reducing the risk of errors propagating through the system.

As communication technologies continue to evolve rapidly, the ability to adapt to new standards and techniques is crucial. FPGAs provide a level of future-proofing that is invaluable in this context. Because they are reprogrammable, FPGAs can be updated to accommodate new error correction techniques and algorithms as they are developed, without requiring significant hardware changes.Moreover Bit error Rate testers can also be designed using high speed processing properties of FPGA [67]. V This adaptability ensures that communication systems using FPGAs remain relevant and effective even as the technological landscape changes. By allowing systems to evolve with new technologies, FPGAs help protect investments in communication infrastructure, ensuring that they continue to deliver reliable and efficient performance for years to come.

V. ERROR DETECTION TECHNIQUES

Forward Error Correction (FEC) techniques are crucial in modern communication systems for ensuring the accuracy of data transmission. These techniques work by adding extra bits, known as redundant information, to the original message before it is transmitted through a communication channel. This added redundancy is not merely extra data but a carefully designed part of the message that helps in identifying and correcting errors that might occur during transmission [68]. The process of Forward Error Correction involves channel coding, which is the method used to encode the message with these redundant bits. At the receiver end, the redundant bits play a vital role. They are used to detect any errors that may have been introduced during transmission and to pinpoint their exact locations. This capability enables the receiver to correct the errors, ensuring that the original message is reconstructed accurately [69][70].

1. Turbo Codes

Turbo codes, in particular, have become a popular choice in many advanced communication systems due to their efficiency and reliability. They are widely used in diverse applications, including military communications, mobile phone networks, satellite communications, and the Digital Video Broadcasting (DVB) standard. Their ability to approach the theoretical limits of channel capacity ensures that they provide robust error correction even in challenging conditions, contributing to the reliability and effectiveness of modern communication systems[71]. Turbo codes consist of two parts, a turbo encoder in the transmitter, and a turbo decoder on the receiver.

1.1 Turbo encoder:-

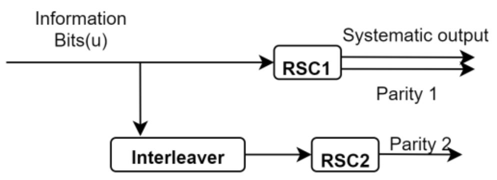


Fig.2 Turbo encoder

A Turbo encoder is constructed using two Recursive Systematic Convolutional (RSC) codes arranged in parallel, with an interleaver placed between them. The role of the interleaver is to shuffle the input sequence, which helps improve error correction by spreading out burst errors [72] [73]. Here's how it works: the original information bits (denoted as U) are fed into the first RSC encoder. This produces both systematic output bits, which closely resemble the original input bits and the first set of parity bits. Meanwhile, the second RSC encoder receives the interleaved version of the information bits and generates the second set of parity bits[74][70]. It does not produce systematic bits, as these are already provided by the first RSC encoder [75]. Consequently, the Turbo encoder has a coding rate of 1/3, meaning for each input bit, it produces three outputs: one systematic bit and two sets of parity bits [76]. To improve the efficiency of the Turbo encoder and increase its coding rate to 1/2, a process called puncturing[77]. is employed. Puncturing involves selectively removing some of the parity bits generated by the Turbo encoder before they are transmitted. This adjustment helps to achieve a higher transmission rate while maintaining effective error correction capabilities [78].

1.2 Turbo decoder:-

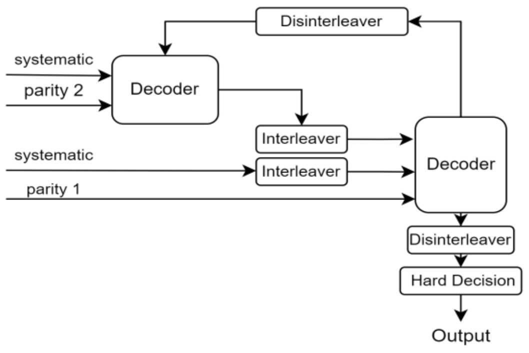


Fig.3 Turbo decoder

The turbo decoder is a crucial component in the receiver of a wireless communication system. Its main role is to decode the data that has been encoded and transmitted using a Turbo encoder. The turbo decoder consists of two convolutional decoders that work together to improve error detection and correction. The inputs to the turbo decoder come from the outputs of the Turbo encoder. Specifically, the systematic bits and the first set of parity bits produced by the first encoder are fed into the first decoder. The output of this decoder is then interleaved and passed to the second decoder along with the second set of parity bits from the second encoder[79]. The second decoder processes this information and generates what is known as extrinsic information. This extrinsic information is then deinterleaved to form the final output of the decoding process [83].

The two decoders continuously exchange information with each other, refining their estimates over multiple iterations. This iterative process continues until the maximum number of iterations is reached or until all detectable errors are corrected. In the final iteration, the output from the second decoder is deinterleaved and then sent to a hard decision unit. This unit converts the decoded data into binary form (1's and 0's) and compares it with the originally transmitted message to verify its accuracy [80] [81].

1.3 FPGA implementation challenges and solutions:-

Implementing Turbo codes on an FPGA presents a unique set of challenges, primarily due to the complexity of Turbo coding algorithms and the specific requirements of FPGA hardware. Turbo codes are based on sophisticated algorithms that involve multiple convolutional encoders and an iterative decoding process. These algorithms are computationally intensive, which can make their implementation on an FPGA challenging. To manage the complexity, break down the Turbo coding process into modular components. Implement each component separately, such as the convolutional encoders, interleavers, and decoders. Use hardware description languages (HDL) like Verilog or VHDL to design these modules. Additionally, leverage FPGA's parallel processing capabilities to handle different parts of the algorithm concurrently. For example, implement parallel paths for encoding and decoding to improve processing speed and efficiency [82]. Turbo codes require significant FPGA resources, including logic elements, memory blocks, and I/O pins. Efficiently using these resources while maintaining performance is a major concern [84].

Optimize resource usage by employing techniques such as pipelining and resource sharing. Pipelining breaks down the processing stages into smaller steps, allowing multiple operations to be performed simultaneously. Resource sharing involves using the same hardware resources for multiple functions, reducing the overall resource footprint. Additionally, perform synthesis and place-and-route optimizations to make the best use of available FPGA resources.[136]

Ensuring that the Turbo code operations meet timing requirements is crucial for real-time performance. The iterative nature of Turbo decoding can exacerbate timing issues. Address timing constraints by carefully analyzing and optimizing the critical path in your design. Use FPGA design tools to perform static timing analysis and identify bottlenecks. Optimize your HDL code and adjust the FPGA design to meet the timing requirements. Techniques such as loop unrolling and optimizing state machine designs can help reduce critical path delays and ensure that the system operates within the required timing constraints.

FPGA implementations can consume substantial power, especially when dealing with high-performance Turbo codes. Power efficiency is a concern, particularly in battery-powered or portable applications. Minimize power consumption by employing low-power design techniques. Use clock gating to turn off unused portions of the FPGA when not needed. Optimize the design for lower clock frequencies where possible and use efficient coding practices to reduce switching activity. Additionally, consider the use of power-efficient FPGA devices and perform power analysis during the design phase to identify and mitigate power hotspots. The iterative decoding process in Turbo codes requires multiple rounds of data exchange between decoders. Implementing this efficiently on an FPGA involves managing complex data flow and synchronization issues. Design the iterative decoding process with careful attention to data flow and synchronization. Use FIFO (First-In-First-Out) buffers to manage data exchanges between decoders and interleavers. Implement efficient handshaking and control mechanisms to coordinate the iterative process. Ensure that the design supports flexible iteration counts and can adapt to varying error conditions.

Testing and verifying the correctness of Turbo code implementations can be complex due to the intricate nature of the algorithms and the high demands on accuracy. Develop a thorough verification strategy that includes simulation, functional testing, and real hardware testing. Create detailed testbenches to simulate various scenarios and ensure that the design performs as expected. Use FPGA debugging tools to monitor and analyze the behavior of the implemented design in real-time. Implement error injection and recovery testing to validate the robustness of the Turbo code implementation [57].

The paper [71] focuses on developing a turbo encoder and decoder for reliable data transmission over noisy communication channels using field-programmable gate arrays (FPGAs). Turbo codes, which consist of two parallel Recursive Systematic Convolutional (RSC) encoders, are designed to minimize transmission errors. The paper outlines the construction of these codes using MATLAB's Simulink and M-file approaches, evaluating performance under varying parameters like code length, iteration numbers, and puncturing operations. The Simulink model is converted into VHDL for practical FPGA implementation, leveraging the parallel processing capabilities of FPGAs. The authors highlight a comparison of turbo codes under different decoding algorithms (Log-MAP, Max-MAP, and SOVA) and their impact on bit error rate (BER).

The practical implementation of the turbo codes was performed on two FPGA boards, demonstrating a successful transmission system without significant errors. Performance tests revealed that increasing code length and iterations improves error correction. Log-MAP outperformed other algorithms in terms of BER performance, though the puncturing mechanism negatively impacted error correction while increasing the data rate. The paper concludes that FPGA-based turbo codes offer effective error correction and can be optimized for specific communication systems, making them suitable for applications requiring high reliability, such as satellite and mobile communications.

In the paper [47] the author introduces a groundbreaking approach to error correction coding. It introduces Turbo Codes, which revolutionizes coding theory by achieving performance near the Shannon limit which is the theoretical maximum efficiency for any coding system. The paper presents Turbo Codes as a form of concatenated coding with two or more simple component codes, typically recursive systematic convolutional (RSC) codes. An interleaver is used between these codes to permute the input bits before being passed to the second encoder, ensuring that error patterns do not concentrate in one place. This architecture provides significant performance improvements. Additionally, the paper explains the iterative decoding algorithm, inspired by the probabilistic method of maximum likelihood, which involves passing information back and forth between decoders to improve the estimation of transmitted data. The authors present simulation results demonstrating the superiority of Turbo Codes, particularly for low signal-to-noise ratios (SNR), proving it to be close to the Shannon limit.

Moreover, the paper highlights the high efficiency of Turbo Codes in various applications such as deep space communications, wireless systems, and other digital communications. Its introduction of soft-input/soft-output (SISO) decoders allows Turbo Codes to iteratively decode received signals with minimal error. The performance of Turbo Codes was noted to be significantly better than previous coding schemes like Reed-Solomon or convolutional codes. The authors predict that Turbo Codes will find practical applications, owing to their excellent error-correcting abilities combined with reasonable complexity in encoding and decoding. This groundbreaking discovery effectively challenged the long-standing view of the coding community and set the stage for future advancements in communication technologies.

The paper [73] compares the performance of three prominent error-correcting codes - Turbo codes, Low-Density Parity-Check (LDPC) codes, and BCH codes. Simulations were conducted in MATLAB to evaluate each code's efficiency in handling noise and errors during data transmission. Turbo codes use an iterative decoding process, LDPC codes employ belief propagation decoding, and BCH codes have fast decoding speeds but limited error correction capabilities.

The results show that BCH codes have the best performance overall, with LDPC codes performing well at higher code rates and approaching Shannon's limit. Turbo codes require more processing power but demonstrate strong performance, especially at lower code rates. The authors conclude that engineers should consider the specific requirements of their communication systems when selecting coding schemes, as each code presents distinct trade-offs in complexity, performance, and error correction capabilities.

2. *LDPC(Low-Density Parity-Check)*

Low-density parity-check (LDPC) [85] codes are a type of linear error-correcting code characterized by their sparse parity-check matrices [86] . A sparse parity check matrix is a type of parity check matrix in which the majority of the elements are zero. Parity check matrices (PCMs) are essential tools used to define linear error-correcting codes, facilitating reliable information transmission over noisy communication channels. These matrices play a crucial role in ensuring that data integrity is maintained, allowing for the detection and correction of errors that may occur during transmission [87]-[90]. By employing PCMs, communication systems can effectively identify discrepancies in the received data and implement corrective measures, thus enhancing the overall robustness of data transmission. This capability is vital in various applications where accurate information relay is critical, especially in environments prone to interference and noise [91]-[93]. This sparsity is not only a defining feature but also a key factor that influences the efficiency of encoding and decoding processes [94][95].

Parity-Check Matrix (H): The matrix H is crucial for defining the LDPC code. It is typically a large, sparse matrix where most of the entries are zero, and only a few entries are one. The sparsity pattern in H determines the structure of the code and affects its performance. For example, a common form is the (3,6) regular LDPC code, where each row of H contains exactly 3 ones and each column contains exactly 6 ones. This structure helps balance the trade-off between performance and computational complexity.

Generator Matrix (G): While the generator matrix G is primarily used for encoding, it is essential for understanding how LDPC codes are constructed. G is derived from H and represents the transformation from the information bits to the codeword. The codeword generated by G must satisfy the constraints imposed by H, ensuring that

$\times \text{codeword}^T = 0$

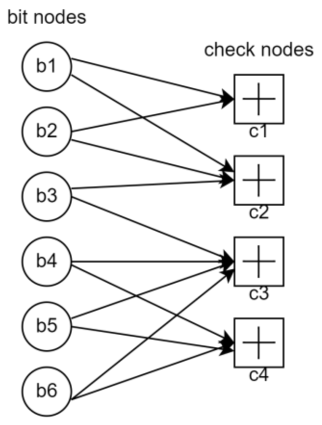
The design of H affects various factors, including the error-correcting performance of the code and the complexity of the decoding process. Well-designed H matrices allow LDPC codes to approach the Shannon limit of channel capacity, making them highly efficient for practical use in communication systems.

2.1 Decoding Algorithms

Decoding LDPC codes involves finding the most likely codeword that matches the received message while satisfying the constraints defined by H. Two prominent algorithms for LDPC decoding are Belief Propagation (BP) and the Min-Sum algorithm [96]-[98].

***2.1.1 Belief Propagation (BP)*:**

Belief Propagation (BP) decoding is a widely used algorithm for decoding Low-Density Parity-Check (LDPC) codes, which are characterized by their sparse parity-check matrices. The process involves constructing a bipartite graph that represents the relationship between the code bits and the parity checks. In this graph, one set of nodes corresponds to the code bits, while the other set corresponds to the parity checks. The decoding process iteratively updates probabilities associated with each bit based on the information received from the checks, allowing the algorithm to converge toward the most likely codeword that satisfies the parity check conditions [99].



The BP algorithm operates in two main steps: the horizontal step and the vertical step. In the horizontal step, the algorithm calculates messages from the check nodes to the variable nodes, updating the probabilities that each bit is a 0 or 1 based on the states of the connected bits. In the vertical step, the variable nodes update their probabilities based on the incoming messages from the check nodes. This iterative process continues until a valid codeword is found or a maximum number of iterations is reached. If the algorithm successfully finds a codeword that satisfies all parity checks, decoding is complete; otherwise, it may declare a failure if the maximum iterations are exhausted without success. This method is effective in approaching the Shannon limit, providing robust error correction capabilities for communication systems operating in noisy environments [100]. BP is known for its accuracy and ability to handle complex codes, but it can be computationally intensive.

Belief Propagation (BP) decoding offers several advantages for Low-Density Parity-Check (LDPC) codes, making it a popular choice in error correction for communication systems. One of the primary benefits is its ability to approach the Shannon limit, which is the theoretical maximum efficiency of data transmission over a noisy channel [89]. BP decoding achieves this by efficiently propagating probabilistic information through the bipartite graph representation of the LDPC code, allowing it to correct errors with high accuracy even in challenging conditions. This capability is particularly useful in applications where maintaining data integrity is crucial, such as in wireless communications and data storage [101].

Another significant advantage of BP decoding is its scalability and flexibility. The algorithm can be easily adapted to various code structures and can handle different levels of sparsity in the parity check matrix. This adaptability allows for the design of LDPC codes that are optimized for specific applications, balancing performance and computational complexity [102]. Additionally, BP decoding is inherently parallelizable, which means it can take advantage of modern computing architectures to speed up the decoding process. This efficiency in both performance and resource utilization makes BP decoding an attractive option for implementing LDPC codes in real-time communication systems, enhancing their reliability and overall effectiveness [103].

In the paper [104] author presents an innovative approach to enhancing the performance of Low-Density Parity-Check (LDPC) codes. Traditionally, short cycles, particularly those of length 4, have been seen as detrimental to the performance of LDPC codes when using standard BP decoding methods. This research challenges that notion by demonstrating that incorporating a significant number of these short cycles can improve decoding performance when a modified BP algorithm is applied. The authors introduce the concept of the Joint Check (JC) bipartite graph, which allows for the simultaneous decoding of two check nodes that share a cycle of length 4.

Through extensive simulations, the authors show that LDPC codes designed with many short cycles outperform those designed to avoid such cycles, particularly when using the proposed JC-BP decoding method. The results indicate that the modified decoding approach yields better Bit Error Rate (BER) performance, achieving improvements of up to 0.5 dB compared to standard BP decoding. This finding is significant as it opens up new avenues for LDPC code design, suggesting that rather than eliminating short cycles, they can be strategically incorporated to enhance performance. Overall, this paper provides valuable insights into LDPC coding strategies and decoding algorithms, contributing to the ongoing development of efficient communication systems.

2.1.2 Min-Sum Algorithm:

The Min-Sum algorithm is a decoding technique used primarily for Low-Density Parity-Check (LDPC) codes. It serves as a simplified version of the more complex Belief Propagation (BP) algorithm, focusing on reducing computational demands while still delivering effective error correction. The core idea behind the Min-Sum algorithm is to utilize a bipartite graph that represents the relationship between code bits and parity checks. During the decoding process, the algorithm iteratively updates messages that represent the probabilities of each bit being a 0 or 1, based on the information received from connected check nodes.

In the Min-Sum algorithm, the calculation of messages from check nodes to variable nodes is simplified by replacing the original BP update rule with a minimum operation. Instead of computing the product of probabilities, the algorithm finds the minimum value of the incoming messages, which allows for faster computations. While this simplification can lead to some degradation in performance, especially in low signal-to-noise ratio (SNR) conditions, various enhancements like the Normalized Min-Sum and Offset Min-Sum algorithms have been developed to improve accuracy. Overall, the Min-Sum algorithm is favored in practical applications due to its balance of efficiency and performance, making it suitable for hardware implementations in communication systems that require reliable error correction.

The paper [105] explores the practical aspects of implementing Min-Sum decoding algorithms for Low-Density Parity-Check (LDPC) codes. The authors focus on the efficiency and performance of Min-Sum decoders in hardware, emphasizing the trade-offs between computational complexity and decoding accuracy. By analyzing various hardware architectures, the study aims to identify optimal configurations that can enhance the decoding process while minimizing resource consumption. The authors provide a detailed examination of the algorithm's performance in terms of speed and error correction capability, highlighting its relevance in modern communication systems.

The results indicate that the Min-Sum algorithm, while simpler than traditional Belief Propagation methods, can achieve competitive error correction performance, especially when implemented in hardware. The paper discusses various implementation strategies, including the use of Field Programmable Gate Arrays (FPGAs) and Application-Specific Integrated Circuits (ASICs), showcasing how these technologies can be leveraged to optimize the decoding process. The authors conclude that with careful design considerations, Min-Sum decoders can be effectively utilized in real-world applications, providing a balance between performance and resource efficiency. Overall, this research contributes valuable insights into the practical implementation of LDPC decoding techniques, underscoring the importance of hardware considerations in achieving reliable communication systems.

2.2 FPGA Implementation Strategies

FPGA implementation of LDPC codes uses the parallel processing property and flexibility of FPGA platforms to achieve high-speed decoding. Effective FPGA design involves several strategies, including parallel processing, pipelining, and efficient memory management.

Parallel Processing in FPGA-Based LDPC Decoders:

FPGAs are highly suitable for parallel processing, making them ideal for LDPC (Low-Density Parity-Check) decoders. LDPC decoding involves multiple variable nodes and check nodes, which can be processed in parallel due to the FPGA's architecture. Each node is treated as an individual processing unit, enabling simultaneous updates across the entire bipartite graph of the LDPC code. For instance, if an LDPC code consists of 100 variable nodes and 100 check nodes, an FPGA can process all these nodes concurrently, significantly reducing the decoding time. This capability makes FPGA-based decoders highly efficient in handling real-time communication tasks like wireless data transfer or video streaming, where high throughput and low latency are essential.

By leveraging this parallelism, FPGAs can sustain high data throughput, crucial for applications like optical communication or satellite links. The scalability of FPGAs allows them to adapt to different LDPC code lengths and rates, ensuring that even large codes can be decoded efficiently without a significant increase in processing time. This parallel processing capability gives FPGAs an edge in high-performance communication systems, as they can manage large volumes of data with minimal delays [106].

Pipelining for Enhanced Efficiency:

Another key technique in FPGA-based LDPC decoders is pipelining. Pipelining divides the decoding process into several stages, allowing different parts of the computation to be handled in parallel. For example, while one stage of the pipeline updates variable nodes, another stage processes check nodes, ensuring that multiple bits are decoded at different points in the pipeline simultaneously. This overlapping of operations enhances the overall throughput of the LDPC decoder by keeping every part of the system active at all times.

The pipelined architecture allows multiple decoding tasks to be handled concurrently, improving the speed and efficiency of the system. This is particularly important for time-sensitive applications like wireless communications or satellite data transfer, where quick and efficient error correction is vital. By combining pipelining with parallel processing, FPGA-based LDPC decoders can deliver the high-performance results required in modern communication systems without compromising on speed or accuracy.[121]

Memory Management for Performance Optimization:

Efficient memory management is a critical aspect of FPGA-based LDPC decoders. FPGAs provide on-chip memory resources, such as block RAM, which can store messages and intermediate results during the decoding process. Storing frequently accessed data in on-chip memory reduces latency, as it eliminates the need to access external memory, which can introduce delays. Efficient memory usage ensures that the FPGA can maintain high decoding speeds, especially in real-time applications.

Moreover, designing effective memory access patterns is important to avoid bottlenecks. Ensuring that memory access is distributed evenly across the FPGA prevents contention between processing units, allowing for smooth data flow throughout the system. Proper memory management also reduces the reliance on external memory interfaces, which can limit performance. By optimizing memory usage, FPGA-based LDPC decoders can handle large amounts of data with minimal delays, ensuring high performance and low latency.

Optimizing FPGA Resources:

To ensure that an FPGA-based LDPC decoder operates efficiently, resource optimization is essential. FPGAs offer a variety of resources, including logic elements, DSP blocks, and memory, which must be managed carefully to fit the design within the available capacity. Resource-efficient algorithms can reduce the use of hardware elements by sharing resources between different stages of the pipeline, such as using the same logic units for both variable and check node updates.

Effective resource management also ensures that the different parts of the decoder do not compete for the same resources, which could cause performance degradation. By minimizing resource contention and optimizing the use of logic and memory blocks, FPGA-based LDPC decoders can achieve high performance within the constraints of the available hardware.

Power Efficiency in FPGA Designs:

In power-sensitive applications like satellite communications or UAVs, power efficiency is a major concern. FPGA-based LDPC decoders can be optimized for low power consumption by using techniques like clock gating, which disables unused parts of the design to reduce dynamic power consumption. By turning off inactive sections of the decoder, the system can operate more efficiently, especially in energy-constrained environments. Additionally, operating the FPGA at lower clock frequencies can further reduce power consumption without compromising performance. Power analysis during the design phase helps identify areas where power can be optimized, allowing designers to address power hotspots and minimize static power usage. These power-saving techniques make FPGA-based LDPC decoders ideal for energy-sensitive applications, ensuring that they deliver high performance while staying within acceptable power limits.

The paper [100] provides a comprehensive comparison of three prominent error-correcting codes used in modern communication systems: Turbo codes, Low-Density Parity-Check (LDPC) codes, and BCH codes. The researchers simulated each coding scheme in MATLAB to assess their performance in terms of efficiency and reliability when dealing with noisy conditions.

The study's findings reveal that BCH codes emerge as the top performer, outshining the other two in terms of overall performance. However, Turbo codes, despite their higher processing demands, also demonstrate impressive reliability. LDPC codes, on the other hand, are noted for their effectiveness at higher code rates, making them particularly suitable for specific applications that require efficient high-rate coding.

The results underscore that while all three coding schemes offer reliable performance, each has its unique advantages and trade-offs. Turbo codes are highlighted for their strong performance, which closely approaches the theoretical Shannon limit. LDPC codes, in contrast, are recognized for their efficiency in high-rate scenarios. The paper emphasizes the importance of carefully selecting the appropriate coding scheme based on the specific requirements of the communication system, such as cost, latency, and processing complexity.

In summary, this research provides valuable insights into the comparative effectiveness of these coding techniques, highlighting their relevance in modern communication technology. By simulating each coding scheme under controlled conditions, the authors offer a comprehensive understanding of the strengths and limitations of Turbo codes, LDPC codes, and BCH codes, empowering communication engineers to make informed decisions when designing reliable and efficient communication systems.

3. BCH

BCH (Bose Chaudhuri Hocquenghem) codes are a class of cyclic error-correcting codes renowned for their ability to handle multiple random errors, making them invaluable in digital communication systems. Named after their creators, these codes play a critical role in applications like Free Space Optical Communication (FSOC), where their robust error correction capabilities help mitigate the challenges of transmission through unpredictable environments. In FSOC, where signals traverse the atmosphere, factors such as turbulence, weather, and scattering can introduce significant noise, fading, and interference [107]. These conditions lead to signal degradation, which BCH codes effectively combat by correcting multiple errors, improving the reliability of the communication link [108]. The error-correcting capacity of BCH codes is represented by t, a parameter based on the design of the code, determining how many errors can be fixed. A defining feature of BCH codes is their cyclic nature not only is a valid codeword considered correct, but any cyclic shift of that codeword is also valid. This cyclic property simplifies encoding and decoding, particularly when implemented in hardware. BCH codes come in both systematic and non-systematic forms, giving engineers flexibility in their design. These codes are constructed over a Galois Field (GF), commonly GF(2^m), which enables efficient polynomial-based calculations during encoding and decoding processes. This algebraic structure allows for rapid and precise error correction, a crucial advantage in FSOC systems where atmospheric distortions, modeled by log-normal or gamma-gamma fading, can result in multiple random errors in the transmitted data[109]. In essence, BCH codes are highly adaptable and effective for environments where data integrity is frequently compromised. Correcting several errors per codeword significantly improves transmission reliability. Their systematic design further ensures that the original message can be recovered efficiently, cementing their importance in both modern communication networks and data storage systems[110].

3.1 BCH Code Encoding

The process of encoding a message using BCH codes [111] involves generating a codeword from a message polynomial.BCH codes are constructed using polynomials over finite fields, specifically GF(2) for binary BCH codes. The key parameters of a BCH code are defined as follows:

- Block Length (n): It is the total number of bits present in the codeword, typically expressed as n=2m−1 for some integer m.
- Message Length (k): It is the number of information bits present in the codeword.
- Error-Correcting Capability (t): It is the maximum number of errors that can be corrected by the BCH code, which is related to the minimum distance d of the code by the formula d≥2t+1.

The BCH code is denoted as BCH(n,k,d) where d is the minimum distance of the code. The table below shows the error correction capability according to the data length(k) and code length(n),

TABLE I. Possible values of n, k, and t for BCH codes

Code length (n)	Data length (k)	Correct up to ‘t’ bits
7	4	1
15	11	1
15	7	2
15	5	3
31	26	1
31	21	2
31	16	3
31	11	5
31	6	7
63	57	1
63	51	2
63	45	3
63	39	4
63	36	5
63	30	6

To encode a message, it is first represented as a polynomial p(x) over GF(2). The encoding process involves the following steps:

1. Generator Polynomial: A generator polynomial g(x) is chosen, which is a product of irreducible polynomials over GF(2). This polynomial has roots that correspond to the error locations in the codeword.

2. Message Polynomial: The message polynomial p(x) is multiplied by x^{n-k} to shift it appropriately for encoding.

3. Division to Find Remainder: The polynomial $p(x)x^{n-k}$ is divided by the generator polynomial g(x) to find the remainder r(x). This remainder is crucial for constructing the codeword.

4. Constructing the Codeword: The codeword c(x) is constructed as follows:

$c(x)=p(x)x^{n-k} + r(x)$

This ensures that c(x) is a multiple of g(x), making it a valid codeword.

3.2 BCH Code Decoding

The decoding process of BCH [112] codes is essential for recovering the original information from potentially corrupted codewords. Let's break down the decoding process into clear, digestible steps.

Step 1: Calculating Syndromes

The first step in decoding a BCH code is to compute what are known as syndromes. When we receive a codeword, it may have been altered by noise or interference. To assess the extent of this corruption, we calculate the syndromes using the formula:

$s_j=R(\alpha^j)=C(\alpha^j)+E(\alpha^j)s_j=R(\alpha^j)=C(\alpha^j)+E(\alpha^j)$

Here, R represents the received codeword, C is the original codeword, and E is the error vector that indicates where changes have occurred. The term α is a primitive element from a finite field, which helps in the calculations. The syndromes give us crucial information about the presence and nature of errors in the received data.

Step 2: Determining the Number of Errors

Once we have the syndromes, the next task is to figure out how many errors have occurred in the codeword. This is done by constructing the error locator polynomial denoted as $\lambda(x)$. This polynomial is derived from the syndromes and plays a critical role in identifying where the errors are located in the received codeword.

Step 3: Finding Error Locations

With the error locator polynomial in hand, we now need to find its roots. Each root corresponds to a position in the codeword where an error has occurred. To accomplish this, we use the Chien search algorithm. This algorithm efficiently searches for the roots of the polynomial, allowing us to pinpoint exactly where the errors are located.

Step 4: Calculating Error Values

After identifying the positions of the errors, the next step is to determine the actual values of the errors at these locations. This is where the Forney algorithm comes into play. The Forney algorithm helps us compute the magnitude of the errors, enabling us to understand how to correct them.

Step 5: Correcting the Errors

Finally, armed with the information about where the errors are and what they are, we can correct the received codeword. This involves flipping the bits at the identified error locations. If the number of detected errors exceeds what the BCH code can handle (its error-correcting capability), the decoder may struggle to recover the original message accurately.

Key Algorithms in BCH Decoding

Several algorithms are commonly used in the BCH decoding process, each playing a vital role:

1. Berlekamp-Massey Algorithm

This algorithm is a cornerstone of BCH decoding. It efficiently computes the error locator polynomial from the syndromes, allowing us to identify the error locations systematically. The Berlekamp-Massey algorithm is known for its effectiveness and speed, making it a popular choice in practical applications.

2. Chien Search

Once we have the error locator polynomial, the Chien search algorithm is employed to find its roots. This step is crucial because it tells us exactly where the errors occurred in the received codeword.

3. Forney Algorithm

Finally, the Forney algorithm helps us determine the specific values of the errors at the identified locations. This ensures that we can accurately correct the errors and recover the original message.

3.3 Implementation of BCH Codes on FPGA for FSOC Systems

In FSOC systems, BCH codes are highly effective for mitigating errors caused by atmospheric disturbances. Their ability to correct multiple random errors makes them ideal for noisy and unpredictable channels. FPGA (Field-Programmable Gate Array) is an ideal platform for implementing BCH codes in FSOC systems due to its ability to handle parallel processing, low-latency operations, and reconfigurable hardware resources. Implementing BCH codes on FPGA provides the computational efficiency, low-latency performance, and flexibility needed for real-time FSOC systems. Through parallelization, careful resource management, and real-time adaptive coding, BCH codes on FPGA offer a robust solution for ensuring reliable communication in FSOC applications [113].

The BCH (Bose-Chaudhuri-Hocquenghem) encoding and decoding processes play a critical role in ensuring error correction within communication systems, especially in high-speed applications like Free Space Optical Communication (FSOC). By implementing these processes on Field Programmable Gate Arrays (FPGAs), significant benefits in speed and efficiency can be realized. The hardware design for BCH encoding incorporates several essential components. A shift register is utilized to manage the input message bits, facilitating the sequential processing of data. Additionally, a polynomial division unit is responsible for dividing the message by the generator polynomial, a fundamental aspect of the encoding process. Control logic is also integrated to oversee the encoding process, determining when it has been completed. The inherent parallel processing capabilities of FPGAs allow for the complete parallelization of the encoding process, which greatly enhances data throughput. This feature is particularly advantageous for FSOC systems that require high-speed data processing to maintain optimal performance under varying conditions [114].

The decoding process for BCH codes is notably more intricate than the encoding phase, primarily due to the requirements for syndrome calculation and error correction. This process typically involves a syndrome calculator that derives syndromes from the received codeword, which is essential for pinpointing errors. To determine the error locator polynomial, the Berlekamp-Massey algorithm is frequently utilized and can be implemented using combinational logic or finite state machines (FSMs) on an FPGA [115]. Additionally, a Chien search module is responsible for locating errors by identifying the roots of the error locator polynomial, while an error correction unit rectifies the detected errors in the received codeword. The decoding process often requires multiple iterations of polynomial evaluations and searches, highlighting the importance of the parallel processing capabilities of FPGAs in expediting these operations.

Despite the significant advantages that FPGAs provide, challenges persist in the implementation of BCH encoding and decoding. While FPGAs excel at parallelizing tasks, some algorithms, such as Berlekamp-Massey and Chien Search, are inherently sequential. Techniques like pipelining or resource-sharing can be employed to optimize these algorithms and improve execution speed [122]. Moreover, BCH decoding can be resource-intensive, especially when dealing with longer block lengths and higher error-correcting capabilities, necessitating careful management of FPGA resources like logic, memory, and processing elements. In real-time FSOC systems, minimizing latency is also critical. FPGA implementations can be fine-tuned to reduce the number of clock cycles needed for encoding and decoding, as well as to implement low-latency error correction methods. Furthermore, in FSOC applications, the ability to adapt to fluctuating channel conditions in real time is vital. By integrating FPGA-based BCH codes with adaptive modulation and coding schemes, the BCH code rate can be dynamically adjusted based on current error rates or signal-to-noise ratios (SNR). This flexibility allows the FPGA to maintain optimal performance without sacrificing bandwidth or power efficiency. Overall, the hardware design for BCH encoding and decoding on FPGAs presents a robust solution for error correction in high-speed communication systems [116].

The paper [117] explores the implementation of a BCH encoder using Field Programmable Gate Arrays (FPGAs) for error correction in digital communication systems. The author emphasizes the significance of BCH codes in effectively correcting multiple random errors, which is crucial for ensuring reliable data transmission. The study delves into the design process of several BCH encoders, focusing on specific code parameters such as (31, 26, 1), (31, 21, 2), (31, 16, 3), (31, 11, 5), and (31, 6, 7). The implementation was carried out using Xilinx-ISE 10.1 and the XC3S700A-4FG484 FPGA, which allowed the researcher to harness the advantages of FPGA technology. These advantages include high calculation rates through parallel processing and the flexibility to modify the design as needed.

The results of the implementation demonstrate the effectiveness of the designed encoders, as evidenced by successful simulations. These simulations showcased the encoding process, where input data was transformed into codewords with added parity bits to detect and correct errors. The paper also delves into the resource utilization of the FPGA, highlighting the efficiency of the encoder designs, which occupy minimal chip area. The author concludes that BCH codes are robust error-correcting codes that are relatively simple to implement and decode, making them suitable for various applications in digital communication. The researcher's findings contribute to the field by providing valuable insights into the practical implementation of BCH codes on FPGA platforms, underscoring their potential for enhancing data reliability in communication systems. The study serves as a testament to the ongoing efforts to improve error correction techniques, ensuring that data transmission remains secure and efficient in an increasingly digital world.

In the paper [116], the author explores the development of a BCH (Bose-Chaudhuri-Hocquenghem) codec designed to enhance data transmission in Free Space Optical (FSO) communication systems. The authors highlight the potential of FSO technology to achieve high data rates over distances of several kilometers, particularly in environments where traditional communication methods may face challenges. Given the susceptibility of optical signals to atmospheric distortions—such as turbulence and scattering—the paper emphasizes the importance of implementing robust error correction mechanisms. The researchers designed a codec for a (31, 16, 3) BCH code on the Nexys-4 FPGA platform, incorporating innovative features like an improved syndrome computation circuit and a parallel Chien search implementation. Their results demonstrate a significant reduction in the bit error rate (BER), showcasing the codec's ability to maintain reliable communication even in challenging conditions.

In addition to the technical design, the paper details the experimental setup used to test the BCH codec, which involved a laser transmitter and a PIN detector operating at a wavelength of 650 nm. The experiments were conducted in an indoor environment, simulating various noise conditions to evaluate the codec's performance. The findings revealed that the BCH codec effectively reduced the number of errors in the received data, with the average BER decreasing from 1.637 x 10^-3 without the channel code to 8.1 x 10^-5 with it. This substantial improvement underscores the codec's effectiveness in mitigating the effects of noise and distortion in FSO systems. Overall, the authors conclude that their implementation of the BCH codec offers a practical solution for enhancing the reliability of FSO communication, paving the way for further advancements in error correction techniques and the development of more powerful coding schemes to address both random and burst errors.

3.4 Applications

BCH codes are utilized in various applications, including:

● Digital Communication: They are widely used in cellular networks, satellite communications, and data storage devices to ensure data integrity by correcting errors that occur during transmission.

● Data Storage: BCH codes are employed in hard drives and SSDs to manage and correct data corruption due to physical defects or wear.

● Cryptography: They are used in code-based cryptographic systems, where their error-correcting capabilities enhance security against certain types of attacks

4. Reed-Solomon Codes

Reed-Solomon (RS) codes are a pivotal advancement in the field of error correction, introduced by Irving S. Reed and Gustave Solomon in 1960. These block error-correcting codes are particularly adept at handling burst errors, which occur when multiple consecutive symbols in a data stream are corrupted. This capability is crucial in various applications, including digital communications and storage systems, such as CDs, DVDs, and satellite communications.

4.1 Key Characteristics of Reed-Solomon Codes

Reed-Solomon codes are defined as RS(n, k), where n represents the total number of symbols in the codeword, which includes both data and parity symbols, while k denotes the number of data symbols. The difference n−k indicates the number of redundant symbols added for error correction. This structure allows RS codes to correct up to t symbol errors, calculated as t = (n−k)/2 . For instance, an RS(255, 223) code can correct up to 16 symbol errors, making it particularly effective for applications that experience burst errors, such as those caused by atmospheric disturbances in satellite communications. The operation of Reed-Solomon codes is grounded in finite fields, specifically Galois Field GF(2^m), where m typically corresponds to the size of the symbols used. In many practical scenarios, 8-bit symbols are employed, leading to operations in GF(2^8). This finite field structure not only facilitates the encoding and decoding processes but also enhances the error correction capabilities by enabling the manipulation of data as polynomial equations. The encoding process for Reed-Solomon codes involves taking the k data symbols and appending n−k parity symbols. These parity symbols are generated using a generator polynomial derived from the Galois field. The resulting codeword, which consists of both data and parity symbols, is then transmitted over potentially noisy channels [118].

On the receiving end, the decoding process employs syndrome-based methods to detect errors. If errors are present, algorithms such as the Berlekamp-Massey algorithm are utilized to identify the locations and values of the erroneous symbols, allowing for their correction. This systematic approach to error detection and correction is what makes Reed-Solomon codes exceptionally robust in various communication and storage applications.

The versatility of Reed-Solomon codes has led to their widespread adoption across numerous technologies. They are integral to the functionality of CDs and DVDs, where they correct errors caused by scratches or other physical damage. Additionally, these codes are employed in digital television, high-speed modems, and even in QR codes, demonstrating their significance in enhancing data integrity and reliability. Reed-Solomon codes represent a foundational technology in error correction, enabling reliable data transmission and storage in an era where digital communication is paramount. Their ability to correct multiple symbol errors efficiently makes them indispensable in both consumer electronics and critical communication systems. In Free Space Optical Communication (FSOC) systems, Reed-Solomon codes are highly effective because FSOC channels often suffer from burst errors due to atmospheric turbulence, absorption, and scattering, as well as occasional deep fading events. RS codes are excellent for correcting these types of errors because they can handle multiple consecutive symbol errors (burst errors) in a block of data. The ability to correct large chunks of corrupted symbols helps maintain communication integrity in varying atmospheric conditions.

FSOC systems require robust error correction schemes because the optical signals used are susceptible to rapid fluctuations in strength due to environmental factors. By employing RS codes, FSOC systems can maintain higher data integrity and quality of service, even in challenging conditions.

4.2 Encoding Process

1. Message Representation:

The message you want to encode is represented as a polynomial, denoted as:

$M(x)=m_0+m_1x+m_2x^2+\cdots+m_{k-1}x^{k-1}$
Here, k represents the number of symbols in the original message, and each m_i is a symbol of the message signal.

2. Generator Polynomial:

A generator polynomial G(x) is chosen based on how many errors the code is meant to correct. It’s typically expressed as:

$G(x)=(x-\alpha_0)(x-\alpha_1)\ldots(x-\alpha_{2t-1})$
Where α is a primitive element of the Galois Field $GF(2^m)$, and t is the number of errors the code can handle.

3. Encoding:
To encode the message, you first multiply the message polynomial M(x) by x^{2t}, producing a new polynomial

$T(x)=x^{2t}M(x)$

Then, divide T(x) by the generator polynomial G(x) using polynomial long division. The remainder of this division forms the parity-check symbols, which are appended to the message, producing the final codeword [120][121].

4.3 Decoding Process

1. Received Polynomial:

Upon receiving the transmitted message, it’s represented as a polynomial R(x), which includes both the message symbols and parity-check symbols. This polynomial might include errors from transmission.

2. Syndrome Calculation:

To check for errors, we compute the syndromes by evaluating R(x) at specific points that correspond to the roots of the generator polynomial G(x). If all syndromes are zero, it means no errors occurred.

3. Error Locator Polynomial:

If any syndrome is non-zero, errors are present. The next step is to compute the error locator polynomial Λ(x), which indicates where the errors are located in the received polynomial. This can be done using algorithms like Berlekamp-Massey or the Euclidean algorithm.

4. Finding Error Locations:

Once Λ(x) is computed, the Chien search algorithm is used to find its roots, which correspond to the error locations in the received message.

5. Calculating Error Values:

After identifying the error locations, the Forney algorithm is applied to calculate the error values at those positions based on the syndromes.

6. Correcting the Errors:

With both error locations and values in hand, the errors are corrected by adjusting the corresponding symbols in R(x). This produces the corrected codeword.

7. Retrieving the Message:

Finally, the original message symbols are extracted from the corrected codeword, ensuring accurate data transmission [120][121].

4.4 FPGA Implementation of Reed-Solomon Codes in FSOC:

Field-Programmable Gate Arrays (FPGAs) are well-suited for implementing Reed-Solomon codes in FSOC systems due to their parallel processing capabilities and the ability to handle the high-speed requirements of optical communication systems. Implementing Reed-Solomon (RS) codes on Field-Programmable Gate Arrays (FPGAs) involves several key steps that leverage the unique capabilities of FPGAs to enhance performance in communication systems, particularly in real-time applications like Free Space Optical Communication (FSOC) [119].

At the core of RS codes is the need for efficient operations in Galois Field GF(2^m), where m=8 is commonly used for byte-level processing. The fundamental operations in this field include addition, multiplication, and division, all of which must be implemented on the FPGA to meet the stringent requirements of real-time communication. Typically, look-up tables (LUTs) are employed to facilitate these Galois Field arithmetic operations, allowing for rapid access and computation. This efficiency is crucial as it directly impacts the overall performance of the RS encoding and decoding processes.

The RS encoder is a vital component that consists of a finite field multiplier and an adder. In this process, data symbols are passed through a generator polynomial to produce parity symbols, which are then appended to the original data stream. The encoding operation can be optimized through pipelining on the FPGA, enabling high-speed performance that is essential for processing FSOC signals in real time. By allowing multiple stages of the encoding process to occur simultaneously, pipelining significantly enhances throughput and reduces latency [120].

The RS decoder is inherently more complex than the encoder, as it must perform error detection and correction. The decoding process typically involves several critical steps:

1. Syndrome Calculation: This initial step detects errors by calculating syndromes from the received data.

2. Error Locator Polynomial: Algorithms such as Berlekamp-Massey or Euclid's algorithm are utilized to derive the error locator polynomial, which identifies the positions of errors within the codeword.

3. Error Magnitude Calculation: The actual values of the errors are determined using Chien search and Forney’s algorithm, allowing for their correction.

These components (syndrome calculator, error locator, and Chien search) are implemented as parallel hardware modules within the FPGA. This parallelism is crucial for meeting the stringent timing requirements of FSOC systems, as it allows for rapid processing of incoming data. FPGAs excel in their ability to perform parallel processing, which is vital for the efficient decoding of RS codes. Each component of the decoder can operate concurrently, drastically reducing the latency associated with the decoding process. Furthermore, pipelining ensures that while one block of data is being decoded, the next block is already in the processing stage. This continuous flow of data not only maintains high throughput but also enhances the overall efficiency of the communication system. In FSOC systems, the parameters such as block length (n) and code rate (k/n) of RS codes may need to be adjusted based on varying atmospheric conditions [116]. FPGAs provide the flexibility to dynamically load different RS code configurations in real-time, adapting to the current channel conditions. Additionally, adaptive modulation and coding can be implemented alongside RS codes on the FPGA, further enhancing the robustness of the communication system [136][139].

Addressing Challenges and Optimizatio*n*

Several challenges must be addressed when implementing RS codes on FPGAs, particularly concerning latency, power consumption, and resource utilization:

● Latency: In real-time FSOC applications, minimizing latency is critical. The RS decoder must be optimized to reduce the number of clock cycles required for each operation, ensuring rapid error correction.

● Power Consumption: Many FSOC systems, especially those deployed in satellites or UAVs, operate under power constraints. FPGA implementations must strike a balance between effective error correction and power efficiency, potentially utilizing low-power FPGA architectures to meet these demands.

● Resource Utilization: The complexity of RS decoding algorithms, such as Berlekamp-Massey and Chien search, can consume significant FPGA resources. Effective management of look-up tables (LUTs), flip-flops (FFs), and memory blocks is essential to optimize resource usage while maintaining high performance.

○ In summary, implementing Reed-Solomon codes on FPGAs involves a careful design of Galois Field arithmetic, efficient encoding and decoding processes, and a focus on parallelism and pipelining. By addressing the unique challenges of real-time communication systems, FPGAs can provide a robust and flexible solution for enhancing data integrity and reliability in various applications, particularly in the dynamic environment of FSOC.

Reed-Solomon codes are a powerful tool in FSOC systems, offering robust error correction in the face of harsh atmospheric conditions. When implemented on FPGAs, they provide the speed and flexibility needed for real-time optical communication. The parallelism and pipelining capabilities of FPGAs make them an excellent platform for efficiently handling the complex arithmetic required for RS encoding and decoding, ensuring reliable communication in variable environments such as those encountered in FSOC applications [120]-[123] [136].

This paper [124] proposes a novel approach to enhance the performance of terrestrial free-space optical (FSO) communication systems by using higher-order pulse position modulation (PPM) schemes combined with Reed-Solomon (RS) error-correcting codes. The authors present simulation results comparing the performance of 2-PPM, 16-PPM, and 256-PPM, both with and without RS coding, under various atmospheric conditions such as ambient light, fog, and scintillation. The results show that using 256-PPM can provide a performance gain of up to 20 dB over the current widely used on-off keying (OOK) modulation. When combined with RS(255,127) codes, the proposed "RS-coded PPM" scheme achieves an overall performance enhancement of up to 25 dB compared to uncoded systems. The authors also discuss the effectiveness of RS codes in combating scintillation fades and the limitations of their approach in dealing with severe fog attenuation. The paper concludes that the RS-coded PPM is a robust and well-performing coded modulation scheme for future terrestrial FSO.

The paper [121] delves into the design and implementation of Reed-Solomon (RS) encoders and decoders on a Field-Programmable Gate Array (FPGA). The authors detail their development of Register Transfer Level (RTL) code for an RS encoder that processes 32-bit input data to produce a 48-bit output. They utilize the Quartus II tool for synthesis and optimization, highlighting the significance of error detection and correction in data transmission, especially in environments susceptible to noise and hardware malfunctions.

The authors provide an in-depth explanation of the Reed-Solomon coding technique, emphasizing its role in error detection and correction. The paper outlines the encoding and decoding processes, incorporating the mathematical principles behind RS codes, including finite fields and Galois fields. Simulation results are presented to validate the functionality of both the encoder and decoder, demonstrating the RS codes' effectiveness in correcting transmission errors. The findings indicate that the developed RS encoder and decoder can substantially improve data reliability in communication systems, suggesting potential future applications in both FPGA and ASIC designs. Overall, the paper offers valuable insights into the practical implementation of error-correcting codes in digital communication systems.

5. Hamming Codes

Hamming Code, created by Richard Hamming in 1950, is a well-known error-correcting code that detects and corrects single-bit errors while identifying two-bit errors. It’s widely employed in digital communication systems to maintain data accuracy, particularly in environments where signal degradation is common. One such example is Free Space Optical Communication (FSOC), where data transmission can be disrupted by atmospheric disturbances, such as turbulence, absorption, or scattering. What sets Hamming codes apart is their innovative use of redundant or parity bits, which are inserted into the original message to provide a way to detect and correct errors. This redundancy ensures the recipient can reconstruct the exact data originally transmitted, even if errors occur along the way. Its simplicity and effectiveness make Hamming codes ideal for systems with limited resources, such as those constrained by power or bandwidth. The straightforward algorithms required to generate and verify these codes make them easy to integrate into both hardware and software systems. Hamming codes find application in various fields, including data storage, telecommunications, and computer networks, where their ability to correct single-bit errors proves invaluable in environments with external interference, such as electromagnetic disturbances. However, while effective for single-bit errors, Hamming codes have limitations in addressing multiple errors, requiring more advanced error-correction methods for systems prone to frequent or complex errors. Hamming codes remain a reliable, resource-efficient method for ensuring data integrity over noisy communication channels. Despite their limitations, ongoing research into error-correction techniques promises to enhance the reliability of data transmission in the future, making systems more robust in the face of increasingly challenging communication environments.[125]

5.1 Structure of Hamming Code

Hamming code is categorized as a linear block code. The basic principle is to add redundant parity bits to the data bits to allow error detection and correction. Hamming codes represent a fascinating and essential aspect of digital communication and data storage, designed to identify and correct errors that can occur during data transmission. Developed by Richard Hamming in the 1950s, these codes enhance the reliability of data transfer by introducing extra bits known as parity bits, which help safeguard the integrity of the original message. At their core, Hamming codes are capable of correcting single-bit errors and detecting double-bit errors. The encoding process involves strategically placing parity bits at specific positions that correspond to powers of two (1, 2, 4, etc.). To determine how many parity bits ‘p’ are needed for a given number of data bits ‘k’, you can use the following formula:

$2^p \geq k + p + 1$

This equation ensures that there is enough redundancy to effectively identify and correct errors in the transmitted data.

Encoding a message with Hamming code is a systematic process that unfolds in several key steps:

1. Calculate Redundant Bits: Start by determining how many parity bits are necessary based on the total number of data bits you have.
2. Position the Parity Bits: Place these parity bits at positions that are powers of two within the message structure.
3. Calculate Parity Values: Each parity bit is computed to ensure that the total number of 1s in the positions it covers is even (this is known as even parity). For instance:

- The first parity bit, r1, checks bits at positions 1, 3, 5, 7, and so on.
- The second parity bit, r2, covers positions 2, 3, 6, 7, etc.
- The third parity bit, r3, looks at positions 4, 5, 6, 7, etc.

The result of this process is a codeword that combines both the original data bits and the calculated parity bits, ready to be sent over the communication channel.

Once the encoded message reaches its destination, the decoding process kicks in, following these steps:

1. Count the Redundant Bits: Just like in encoding, start by figuring out how many parity bits are present in the received codeword.
2. Identify Parity Bit Positions: Determine where the parity bits are located within the received message.
3. Perform Parity Checks: Calculate the parity for each of the parity bits based on the received bits. This step is crucial for spotting any errors:

- If all calculated parities match the expected values, the message is considered error-free.
- If discrepancies arise, the syndrome (a binary number that reflects the results of the parity checks) will point to the position of the erroneous bit.

4. Correct the Error: If a single-bit error is detected, simply flip the erroneous bit to restore the original message.

In essence, Hamming codes are a remarkable tool in the realm of digital communications, ensuring that data remains intact and reliable through the clever use of redundancy [126]. By understanding and applying these codes, we can significantly enhance the accuracy of data transmission, making them invaluable in various applications, from telecommunications to computer memory systems [127]. Hamming code uses even parity or odd parity to detect errors.

If the parity bit detects a change in the expected parity, it identifies an error in one of the data bits.

- **Error detection**: By recalculating the parity bits at the receiver end and comparing them with the received parity bits, the system detects whether an error has occurred.
- **Error correction**: To correct the error, the positions of the bits that fail the parity checks are combined, and the system pinpoints the exact bit that is erroneous. Once the location of the erroneous bit is known, it can be flipped to correct the error.

For example, if the received bits fail at positions 1, 2, and 4, the combination 1+2+4=71+2+4 = 71+2+4=7 tells us that the 7th bit is erroneous and needs to be corrected.

5.2 Hamming Code Implementation on FPGA for FSOC

Implementing Hamming Code on FPGA for Free Space Optical Communication (FSOC) involves several key considerations, including the design of encoder and decoder modules, the choice of FPGA platform, and the optimization for performance and reliability in optical communication environments. Hamming Code is a widely used error-correcting code that can detect and correct single-bit errors in data transmission. It works by adding redundant bits to the data, allowing the detection and correction of errors without needing retransmission. This is particularly useful in communication systems like FSOC, where data integrity is paramount due to potential interference and distortion.

5.2.1 FPGA Implementation Steps

Designing the Hamming code encoder and decoder

the implementation of hamming code on fpga for free space optical communication fsoc typically involves two main encoder components and the decoder components the encoder module takes the data input and generates the hamming code by adding redundant bits the positions of these bits are determined based on the powers of two which allows for efficient error detection and correction the decoder module on the other hand receives the encoded data and checks for errors if an error is detected the decoder identifies the erroneous bit and corrects it ensuring the integrity of the transmitted data.

Choosing the fpga platform

selecting an appropriate fpga platform is crucial for achieving the desired performance in hamming code implementation for fsoc the virtex-5 fpga for instance has been widely used in various implementations due to its high speed and flexibility the choice of fpga affects the complexity of the design and the achievable throughput a well-suited fpga can provide the necessary resources and performance to handle the real-time processing requirements of hamming code in fsoc systems

Programming and simulation

the design of the hamming code encoder and decoder is typically implemented using hardware description languages hdls such as vhdl or verilog tools like xilinx ise or vivado are commonly used for synthesis and simulation the simulation phase is critical for verifying the functionality of the encoder and decoder before deploying them on the fpga by thoroughly testing the design in a simulated environment potential issues can be identified and addressed ensuring the reliability and accuracy of the hamming code implementation

Performance Optimization

to enhance the performance of the hamming code implementation on fpga for fsoc several optimization techniques can be employed parallel processing is one such technique where multiple data streams are processed simultaneously this approach can significantly increase the throughput of the system allowing for real-time data transmission additionally utilizing efficient algorithms for syndrome calculation and error correction can reduce latency and improve the overall efficiency of the hamming code implementation

Testing and Validation

After implementing the hamming code on fpga extensive testing is necessary to ensure that the system performs well under various conditions this includes testing for different types of errors such as single-bit errors burst errors and multiple-bit errors additionally it is crucial to ensure that the system can handle the expected data rates without compromising the integrity of the transmitted data by conducting thorough testing and verification the reliability and robustness of the implementation of the hamming code can be verified making it suitable for deployment in fsoc systems

Implementing Hamming Code on FPGA for FSOC applications is a robust solution for ensuring data integrity in optical communication systems [128]. By carefully designing the encoder and decoder, selecting the right FPGA platform, and optimizing for performance, it is possible to create a reliable communication link that can withstand the challenges posed by atmospheric conditions. This approach not only enhances the reliability of data transmission but also supports the increasing demand for high-speed communication in modern applications[129].

The paper [131] provides a comprehensive analysis of Hamming codes, which are essential for error detection and correction in digital communication. The authors highlight the capabilities of Hamming codes to detect up to two simultaneous bit errors while correcting single-bit errors, contrasting this with simpler parity codes that only detect errors. The study employs simulations using VHDL to demonstrate the effectiveness of Hamming codes in minimizing error occurrences across various data bits. The paper emphasizes the importance of error correction in enhancing throughput, particularly in error-prone channels, and outlines the structure and implementation of Hamming codes, including their mathematical foundations and practical applications in fields such as telecommunications and data storage[130].

In the conclusion, the authors discuss the limitations of Hamming codes, particularly in scenarios involving multiple errors, where additional parity bits may be required to achieve even parity and detect uncorrectable errors. They also suggest future improvements, such as refining the simulation to better handle cases of multiple errors. The paper serves as a valuable resource for understanding Hamming codes' performance and their critical role in ensuring reliable communication in various digital systems. The detailed exploration of both theoretical and practical aspects of Hamming codes makes this study a significant contribution to the field of error correction coding [125] .

VI. ADVANTAGES OF FPGAS IN SIGNAL PROCESSING

1. *Parallel Processing Capabilities*

Signal processing often involves handling vast amounts of data at high speeds, making parallel processing a critical requirement. FPGAs excel in this area due to their inherent parallel architecture [132]. Unlike CPUs, which process instructions sequentially, FPGAs can execute multiple operations simultaneously by configuring multiple logic blocks to operate in parallel. This ability to perform concurrent computations significantly enhances the speed and efficiency of signal-processing tasks [133]. For example, in image and video processing, FPGAs can handle multiple pixel operations at once, enabling real-time processing of high-resolution images [134]. Similarly, in applications like radar signal processing, FPGAs can simultaneously execute multiple filtering and transformation tasks, providing faster and more accurate results [135][136].

2. *Reconfigurability and Flexibility*

FPGAs’ reconfigurability allows them to adapt to changing requirements and standards, which is particularly advantageous in signal processing. New signal processing algorithms and techniques are continually being developed, and the ability to update FPGA configurations to implement these new methods without changing the hardware is invaluable. This flexibility is essential for keeping pace with evolving technologies, such as the transition from 4G to 5G in telecommunications, where new protocols and modulation schemes are continuously introduced. Furthermore, the ability to quickly reconfigure FPGAs means that multiple signal processing tasks can be performed on a single device, reducing the need for multiple specialized chips and simplifying hardware design [137].

3. *Low Latency and Deterministic Performance*

One of the most critical advantages of FPGAs in signal processing is their ability to deliver low-latency, deterministic performance. FPGAs execute operations directly in hardware, avoiding the overhead associated with software execution on general-purpose processors. This direct execution path allows FPGAs to perform complex computations with minimal delay, which is essential for real-time applications such as high-frequency trading, medical imaging, and audio signal processing. In these applications, even microsecond-level delays can be unacceptable.

FPGAs provide predictable timing, ensuring that tasks are completed within a defined timeframe, which is not always possible with general-purpose processors that might experience unpredictable delays due to multitasking and operating system overhead [138].

4. *High Throughput for Data-Intensive Applications*

Modern signal processing applications, such as high-definition video encoding, digital audio processing, and large-scale scientific computations, require the processing of massive amounts of data in real time. FPGAs are well-suited for these tasks because they can process data streams continuously at very high speeds. By leveraging parallel processing and pipelining [136] [139], FPGAs can maintain high throughput, ensuring that data processing keeps up with high input rates. For example, in a digital video broadcast system, FPGAs can handle the compression, encoding, and multiplexing of video streams simultaneously, ensuring smooth transmission without buffering or lag [140].

5. Energy Efficiency and Resource Optimization

FPGAs are generally more energy-efficient than general-purpose processors for specific tasks. This efficiency comes from their ability to execute specific operations in hardware rather than relying on software running on a general-purpose processor, which requires more cycles and therefore consumes more power. In signal processing applications where power consumption is a critical concern—such as in portable medical devices, wireless sensor networks, and remote monitoring systems—FPGAs offer a more sustainable solution. Additionally, FPGAs allow designers to optimize the use of hardware resources, such as logic gates and memory, to meet the specific performance and power requirements of an application, further enhancing their suitability for energy-constrained environments [141].

VII. ADVANTAGES OF FPGAS IN COMMUNICATION SYSTEMS

1. Customization for Complex Communication Protocols

Modern communication systems are increasingly complex, involving a variety of protocols and standards that must be supported simultaneously. FPGAs provide the ability to implement these protocols directly in hardware, allowing for highly optimized and efficient processing. This capability is particularly beneficial in applications like wireless base stations, software-defined radios (SDRs), and satellite communication systems, where different communication standards (e.g., LTE, Wi-Fi, Bluetooth) must be supported. FPGAs can be dynamically reconfigured to switch between different protocols or to update protocol implementations, enabling communication devices to adapt to different network conditions and standards without requiring new hardware [142].

2. Support for Advanced Error Correction and Signal Processing Algorithms

Communication systems often operate in environments with high levels of noise, interference, and signal degradation, requiring robust error correction algorithms to ensure reliable data transmission. FPGAs are ideally suited for implementing advanced error correction techniques, such as

Low-Density Parity-Check (LDPC) codes, Turbo Codes, and Reed-Solomon codes, which require intensive computation. The parallel processing capabilities of FPGAs allow them to execute these complex algorithms more efficiently than traditional processors, reducing the time needed for error detection and correction. This efficiency is crucial in maintaining the integrity of data transmission in high-speed communication systems, such as fiber-optic networks and deep-space communication links [143] [144].

3. Real-Time Processing for High-Speed Communication

In high-speed communication systems, such as those used in 5G networks or satellite communications, the ability to process data in real time is critical. FPGAs can handle the rapid processing of incoming and outgoing data streams, performing tasks such as modulation, demodulation, encoding, and decoding with minimal delay. This real-time capability ensures that communication systems can maintain high throughput and low latency, even under heavy data loads. For example, in a 5G base station, FPGAs can handle the massive MIMO (Multiple Input Multiple Output) processing required to manage multiple user connections simultaneously, providing fast and reliable data transmission [145].

4. Seamless Integration with Other System Components

FPGAs can be easily integrated with other digital and analog components in a communication system, such as digital signal processors (DSPs), analog-to-digital converters (ADCs), and digital-to-analog converters (DACs). This integration capability enables FPGAs to act as a central hub for signal processing tasks, coordinating the flow of data between different components and optimizing overall system performance. In a communication transceiver, for instance, FPGAs can manage the entire signal chain, from signal reception and processing to modulation and transmission, providing a highly integrated and efficient solution [146].

5. Security and Adaptability for Evolving Threats

Security is a significant concern in modern communication systems, particularly in military, financial, and critical infrastructure applications. FPGAs can be programmed to implement robust encryption and decryption algorithms directly in hardware, offering faster and more secure processing than software-based solutions. Additionally, FPGAs' reconfigurability allows them to be updated with new security protocols as threats evolve, providing a future-proof solution that can adapt to new vulnerabilities. This adaptability is particularly important in applications like secure government communications, banking networks, and smart grids, where the ability to quickly respond to emerging threats is crucial [147].

VIII. CONCLUSION

In Free-Space Optics (FSO) communication systems, which use light to transmit data through the air, ensuring reliable data transfer is crucial due to various environmental challenges. Conditions like atmospheric turbulence, rain, and fog can cause significant disruptions and errors in the transmitted signal. To address these issues effectively, robust error correction and detection methods are necessary. Field-Programmable Gate Arrays (FPGAs) have proven to be highly effective tools for implementing these error correction and detection mechanisms in FSO systems. Thanks to their versatility and high processing power, FPGAs can handle complex error correction algorithms in real-time, which is vital for maintaining the integrity and reliability of the communication. As FSO technology advances, we can expect further improvements in FPGA design and error correction techniques. Future developments might include integrating more sophisticated decoding algorithms, incorporating adaptive error correction strategies, and using hybrid FPGA-processor systems. These enhancements will help optimize performance in various and changing environmental conditions.In summary, FPGA-based error correction and detection are key to advancing Free-Space Optics technology. By harnessing the power and flexibility of FPGAs, we can tackle the challenges posed by environmental disturbances and ensure that FSO systems deliver reliable, high-speed data transmission

REFERENCES

[1] V. W. S. Chan, "Free-Space Optical Communications," in Journal of Lightwave Technology, vol. 24, no. 12, pp. 4750-4762, Dec. 2006, doi: 10.1109/JLT.2006.885252.

[2] Ghassemlooy, Zabih, Arun Majumdar, and Arockia Bazil Raj. "Introduction to free space optical (FSO) communications." (2019): 1-26.

[3] D. Kedar and S. Arnon, "Urban optical wireless communication networks: the main challenges and possible solutions," in IEEE Communications Magazine, vol. 42, no. 5, pp. S2-S7, May 2004, doi: 10.1109/MCOM.2004. 1299334.

[4] Arun K. Majumdar, ”Advanced Free Space Optics FSO A Systems Approach”

[5] Arun K. Majumdar; Zabih Ghassemlooy; A. Arockia Bazil Raj, Principles and Applications of Free Space Optical Communications, 2019.

[6] DAS, S., HENNIGER, H., EPPLE, B., MOORE, C., RABINOVICH, W., SOVA, R., YOUNG, D. ‘Requirements and challenges for tactical free-space lasercom’. In IEEE Military Communications Conference. San Diego (USA), 2008, p. 1 – 10.

[7] A Arockia Bazil Raj, Free Space Optical Communication, Nat. Conf. Emerging Trends in Communication Systems (AECOVISION’06),44-48,2006.

[8] Henniger H, Wilfert O. An introduction to free-space optical communications.Radioengineering 2010;19(2):203–11.

[9] WILLEBRAND, H., GHUMAN, B. Fiber optics without fiber. IEEE Spectrum, 2001, vol. 38, no. 8, p. 40 – 45.

[10] Raj, A. Arockia Bazil, Prabu Krishnan, Ucuk Darusalam, Georges Kaddoum, Zabih Ghassemlooy, Mojtaba Mansour Abadi, Arun K. Majumdar, and Muhammad Ijaz. 2023. "A Review–Unguided Optical Communications: Developments, Technology Evolution, and Challenges" Electronics 12, no. 8: 1922. https://doi.org/10.3390/ electronics12081922

[11] A. Arockia Bazil Raj,Mono-pulse tracking system for active free space optical communication, Optik,Volume 127,Issue 19,2016,Pages 7752-7761,ISSN 0030-4026,https://doi.org/10.1016/j.ijleo.2016.05.143.

[12] Chakkaand, Mary Latha, and AA Bazil Raj. "Design and Implementation of A Secure Physical Unclonable Function In FPGA." In IEEE Inter., Conf.,“Systems Inventive Research in Computing Applications (ICIRCA 2020)”, pp. 1-7. 2020.

[13] Arockia Bazil Raj, A. and Arputha Vijaya Selvi, J.. "Comparison of Different Models for Ground-Level

Atmospheric Attenuation Prediction with New Models According to Local Weather Data for FSO Applications" Journal of Optical Communications 36, no. 2 (2015): 181-196. https://doi.org/10.1515/joc-2014-0054

[14] Anthonisamy, Arockia Bazil Raj, James, Arputha Vijaya Selvi, 2016, Formulation of atmospheric optical attenuation model in terms of weather data, Journal of Optics, https://doi.org/10.1007/s12596-016-0325-6

[15] Joseph George Chalil, Shachi Kadam, Chinchu Joseph, K.J. Kulkarni, and A.A.Bazil Raj. “Characterization of Free-space Gaussian Beam Propagation and Recent Developments of FSO Communication Technology: A Review”. International Journal of Engineering Research and Reviews 12, no. 3 (August 14, 2024): 1–24. https://doi.org/ 10.5281/zenodo.13318940.

[16] A Arockia Bazil Raj,FSO channel atmospheric attenuation and refractive index (Cn2) modeling as the function of local weather data, Principles and Applications of Free Space Optical Communications 2019, vol 1, pg 99-125.

[17] Raj, A. B., & Majumder, A. K. (2019). Historical perspective of free space optical communications: From the early dates to today's developments. IET Communications, 13(16), 2405-2419. https://doi.org/10.1049/iet-com.2019.0051

[18] Arockia Bazil Raj, Free Space Optical Communication: System Design, Modeling, Characterization, and dealing with turbulance.

[19] Bazil Raj, Arockia A., et al. "Intensity feedback-based beam wandering mitigation in free-space optical communication using neural control technique." EURASIP Journal on Wireless Communications and Networking 2014 (2014): 1-18

[20] Raj, A. Arockia Bazil, et al. "Design of cognitive decision making controller for autonomous online adaptive beam steering in free space optical communication system." Wireless Personal Communications 84 (2015): 765-799

[21] Joseph, C., Kumar, E. S. V., Raj, A.B., & Sharma, N. (2023, December). A Linear Closed Loop Feedback System for Beam Wander Correction in Medium-Range Optical Link. In 2023 IEEE Pune Section International Conference (PuneCon) (pp. 1-5).IEEE.

[22] Arockia Bazil Raj, A., and Ucuk Darusalam. "Performance improvement of terrestrial freespace optical communications by mitigating the focal-spot wandering." Journal of Modern optics 63.21 (2016): 2339-2347.

[23] De, Sampurna, and AA Bazil Raj. "Experimental study of sand-storm effect on digital fso communication link." 2020 International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT) . IEEE, 2020.

[24] Taher, M.A., Abaza, M., Fedawy, M., Aly, M.H.: Relay selection schemes for FSO communications over turbulent channels. Appl. Sci. 9(7), 1281 (2019)

[25] H. Kaushal, G. Kaddoum Optical communication in space: challenges and mitigation techniques IEEE Commun. Surv. Tutor., 19 (2017), pp. 57-96, 10.1109/COMST.2016.2603518

[26] J. K. Kwon, "Inverse source coding for dimmingin visible light communications using NRZ-OOK on reliable links", IEEE Photon. Technol. Lett., vol. 22, pp. 1455-1457, Oct. 2010.

[27] PAUER, M., WINTER, P., LEEB, W. Bit error probability reduction in direct detection optical receivers using rz coding. Journal of Lightwave Technology, 2001, vol. 19, no. 9, p. 1255 – 1262.

[28] LEEB, W., WINTER, P., PAUER, M. The potential of return-to-zero coding in optically amplified lasercom systems. In IEEE Lasers and Electro-Optics Society 1999 12th Annual Meeting LEOS ’99, vol. 1. San Francisco (USA), 1999, p. 224 – 225.

[29] STREET, A., SAMARAS, K., OBRIEN, D., EDWARDS, D. Closed form expressions for baseline wander effects in wireless ir applications. Electronics Letters, 1997, vol. 33, no. 12, p. 1060 – 1062

[30] MAJUMDAR, A. K., RICKLIN, J. C. Free-Space Laser Communications Principles and Advances. Sew York (USA): Springer, 2008.

[31] DAVID, F. Scintillation loss in free-space optic im/dd systems. In LASE 2004, vol. 5338. San Jose (USA), 2004.

[32] Wei Liu, Wenxiao Shi, Jingtai Cao, Yaowen Lv, Kainan Yao, Shuai Wang, Jihong Wang, Xuefen Chi, Bit error rate analysis with real-time pointing errors correction in free space optical communication systems, Optik, Volume 125, Issue 1,2014, Pages 324-328, ISSN 0030-4026,https://doi.org/10.1016/j.ijleo.2013.06.043.

[33] Mansour, A., Mesleh, R., Abaza, M., 2017. New challenges in wireless and free space optical communications. Opt. Lasers Eng. 89, 95–108

[34] Mudge, Kerry & Grant, Kenneth & Clare, Bradley & Biggs, Colin & Cowley, William & Manning, Sean & Lechner, Gottfried. (2016). Development and characterization of FPGA modems using forward error correction for FSOC. 98330G. 10.1117/12.2238759.

[35] David T. Wayne, Ronald L. Philips, Larry C. Andrews, Troy Leclerc, Paul Sauer, and John Stryjewski "Comparing the Log-Normal and Gamma-Gamma Model to Experimental Probability Density Functions of Aperture Averaging Data, Proc. Of SPIE, 7814, 78140K (2010)

[36] Vetelino FS, Recolons J, Andrews L, Young C, Clare B, Corbett K, Grant K, PDF models of the irradiance fluctuations in Gaussian beam waves, in, SPIE-Int. Soc. Opt. Eng., 6215, Place of Publication: Orlando, FL,The USA. Country of Publication: USA., 2006, pp. 62150A-66219

[37] Hamza Gerçekcioğlu and Yahya Baykal, "Scintillation and bit error rate in bidirectional laser communications between an aerial vehicle and a satellite using annular optical beams in a strong turbulent atmosphere," J. Opt. Soc. Am. A 38, 1391-1399 (2021)

[38] R. Leoraj and J. A. V. Selvi, "Comparative performance analysis of forward error correcting codes for Free Space Optical communication," 2016 International Conference on Emerging Trends in Engineering, Technology and Science (ICETETS), Pudukkottai, India, 2016, pp. 1-6, doi: 10.1109/ICETETS.2016.7603067.

[39] S.L. Sathiya Narayanan, B.C. Dhanush Devappa, Kalyani Pawar, Shreyas Jain, Appala Venkata Ramana Murthy, Implementation of forward error correction for improved performance of free space optical communication channel in adverse atmospheric conditions, Results in Optics, Volume 16, 2024, 100689, ISSN 2666-9501, https://doi.org/10.1016/j.rio.2024.100689.

[40] G. Tzimpragos, C. Kachris, I. B. Djordjevic, M. Cvijetic, D. Soudris and I. Tomkos, "A survey on FEC codes for 100 G and beyond optical networks", IEEE Commun. Surveys Tuts, vol. 18, no. 1, pp. 209-221, 1st Quart. 2016

[41] Arockia Bazil Raj, A., and S. Padmavathi. "Statistical analysis of accurate prediction of local atmospheric optical attenuation with a new model according to weather together with beam wandering compensation system: a seasonwise experimental investigation." Journal of Modern Optics 63.13 (2016): 1286-1296.

[42] Kopeika, Natan, and A. Arockia Bazil Raj. "Special Issue on "Optical and RF Atmospheric Propagation"." Sensors 23.7 (2023): 3644.

[43] Raj Anthonisamy, Arockia Bazil,Padmavathi Durairaj, and Lancelot James Paul. "Performance analysis of free space optical communication in open-atmospheric turbulence conditions with beam wandering compensation control." IET Communications 10.9 (2016): 1096-1103.

[44] Raj, A. Arockia Bazil, J. Arputha Vijaya Selvi, and S. Durairaj. "Comparison of different models for ground-level atmospheric turbulence strength (Cn 2) prediction with a new model according to local weather data for FSO applications." Applied Optics 54.4 (2015): 802-815.

[45] S. Bloom, E. Korevaar, J. Schuster, and H. Willebrand, "Understanding the performance of free-space optics," Journal of Optical Networking, vol. 2, no. 6, pp. 178-200, Jan. 2003.

[46] Raj, A. Arockia Bazil, J. Arputha Vijaya Selvi, and D. Kumar. "Low-cost BER measurement in wireless digital laser communication link with autonomous beam steering system." ICWCSC (2010): 1-6.

[47] Jagdale, Atharva Ninad, and AA Bazil Raj. "Model-Free Ber Measurement in Free Space Laser Communication Link." 2021 International Conference on System, Computation, Automation and Networking (ICSCAN) . IEEE, 2021.

[48] Robert K. Tyson, "Bit-error rate for free-space adaptive optics laser communications," J. Opt. Soc. Am. A 19, 753-758 (2002)

[49] A. Arockia Bazil Raj, J. Arputha Vijaya Selvi, and S. Durairaj, "Comparison of different models for ground-level atmospheric turbulence strength (Cn2) prediction with a new model according to local weather data for FSO applications," Appl. Opt. 54, 802-815 (2015).

[50] Xiaoming Zhu and J. M. Kahn, "Free-space optical communication through atmospheric turbulence channels," in IEEE Transactions on Communications, vol. 50, no. 8, pp. 1293-1300, Aug. 2002, doi: 10.1109/TCOMM.2002. 800829.

[51] Katsilieris, T.D., Latsas, G.P., Nistazakis, H.E., Tombras, G.S.: An accurate computational tool for performance estimation of FSO communication links over weak to strong atmospheric turbulent channels. Computation 5(1), 18 (2017)

[52] X. Zhu and J. M. Kahn, "Free-space Optical communication through atmospheric turbulence channels," IEEE Transactions on Communications, vol. 50, pp. 1293-1300, August 2002.

[53] J Soffia Jennifer, A Arockia Bazil Raj "Formulation of Empirical Model for Atmospheric Turbulence Strength (Cn2) Prediction" Conference Proceedings of Kings College of Engineering- 2015

[54] Raj, A. Arockia Bazil, et al. "Mitigation of beam fluctuation due to atmospheric turbulence and prediction of control quality using intelligent decision-making tools." Applied Optics 53.17 (2014): 3796-3806.

[55] Sulaiman, Z. N., Marhaban, M. H., Hamidon, M. N., & Sidek, R. M. (2009). Design and implementation of FPGA-based systems—a review. Australian Journal of Basic and Applied Sciences.

[56] Kuon, I., Tessier, R., & Rose, J. (2008). FPGA architecture: Survey and challenges. Foundations and Trends® in Electronic Design Automation, 2(2), 135-253.

[57] Monmasson, E., & Cirstea, M. N. (2007). FPGA design methodology for industrial control systems—A review. IEEE Transactions on Industrial Electronics.

[58] Farooq, U., Marrakchi, Z., & Mehrez, H. (2012). FPGA architectures: An overview. In Heterogeneous FPGA Architectures (pp. 15-30). Springer.

[59] A. ALBERT RAJ, T. LATHA, VLSI Design,2008.

[60] A Arockia Bazil Raj Vaishnavi D. Kanzarkar, Kshitija S. Kayat, Akanksha B. Jadhav, Pranav kumar Tingare, Sampurna De, Chinchu Joseph, Vineeth Kumar, Vanita. P. Patil "FPGA Board Ramp Generation for 2 GHz RF Bandwidth in X-Band Application" International Journal of Electrical and Electronics Research-2024

[61] Abel, F., Weerasinghe, J., & Hagleitner, C. (2017). An FPGA platform for hyperscalers. In 2017 IEEE 25th Annual Symposium on Field-Programmable Custom Computing Machines (pp. 15-22). IEEE. Retrieved from IBM.com

[62] Cummings, M., & Haruyama, S. (1999). FPGA in the software radio. IEEE Communications Magazine, 37(2), 108-112. Retrieved from IEEE Xplore

[63] Trimberger, S., Carberry, D., & Johnson, A. (1997). A time-multiplexed FPGA. In Proceedings of the 5th Annual IEEE Symposium on Field-Programmable Custom Computing Machines (pp. 22-30). IEEE. Retrieved from Iowa State University

[64] Chen, D., Cong, J., & Pan, P. (2006). FPGA design automation: A survey. Foundations and Trends® in Electronic Design Automation, 1(3), 195-330. Retrieved from NowPublishers

[65] Mueller, R., Teubner, J., & Alonso, G. (2009). Data processing on FPGAs. Proceedings of the VLDB Endowment, 2(1), 782-793. Retrieved from VLDB.org

[66] Trimberger, S. M., & Moore, J. J. (2014). FPGA security: Motivations, features, and applications. Proceedings of the IEEE, 102(7), 1069-1081.

[67] Raj, A. Arockia Bazil. FPGA-based embedded system developer's guide . CRC Press, 2018.

[68] C. Berrou and A. Glavieux, "Near optimum error-correcting coding and decoding: turbo-codes," IEEE Trans. Commun., vol. 44, no. 10, pp. 1261–1271, 1996.

[69] H. Ljunger, “Turbo decoder with early stopping criteria,” Lund University,2016.

[70] Hui Jin and J.Robert McEliece, “Coding Theorems for Turbo Codes Ensembles”, IEEE Transactions on Information Theory, Vol.48, No.6, pp.1451-1461,2002.

[71] M. A. Fleah and Q. F. Al-Doori, "Design and Implementation of Turbo encoder/ decoder using FPGA," 2019 First International Conference of Computer and Applied Sciences (CAS), Baghdad, Iraq, 2019, pp. 46-51, doi: 10.1109/CAS47993.2019.9075589.

[72] G. Santosh and D. Rajaram, "Design and implementation of turbo coder for LTE on FPGA", International Journal of Electronics Signals and Systems (IJESS), . 3, no. 2, pp. 15–19, 2013.

[73] Todd K Moon. "Error Correction Coding Mathematical Methods and Algorithms”. Published by John Wiley & Sons Inc. 2005.

[74] Lee, S.H., Kwon, J.K., 2012. Turbo code-based error correction scheme for dimmable visible light communication systems. IEEE Photonics Technol. Lett. 24, 1463–1465.

[75] Yaoqiang Han, Anhong Dang, Yongxiong Ren, Junxiong Tang, and Hong Guo, “Theoretical and experimental studies of turbo product code with time diversity in free space optical communication”, Optics Express, Vol. 18, No. 26, pp. 26978-26988,2010

[76] M. F. Mosleh, M.F, Abid and M.Al-Sadoon, "Implementation of Turbo Code Based Xilinx System Generator ", Institute for Computer Science(ICTS), Social Informatics and Telecommunications Engineering 2019 Published by Springer Nature Switzerland AG 2019. BROADNETS, 2018, LNICST 263, pp. 1–8, 2019.

[77] D. Lake and N. Kheriebt, "Performance Analysis of Turbo Codes", MSc. thesis, University M’hamed Bougara – Boumerdes, 2017.

[78] R. M. Deshmukh and S. A. Ladhake, "Analysis of Various Puncturing Patterns and Code Rates: Turbo Code", International Journal of Electronic Engineering Research, vol. 1, no. 2, pp. 79–88, 2009

[79] A. K. Gupta and S. Kumar, “VHDL Implementation of different Turbo Encoder using Log-MAP Decoder,” vol. 2, no. 1, pp. 49–53, 2010.

[80] C. Studer, C. Benkeser, S. Belfanti, and Q. Huang, "Design and implementation of a parallel turbo-decoder ASIC for 3GPP-LTE", IEEE Journal of solid-state circuits, vol. 46, no. 1, pp. 8–17, 2011.

[81] S. K. Chronopoulos, G. Tatsis, and P. Kostarakis, "Turbo Codes — A New PC Design", Scientific research and information technology (SCIRES), vol. 3, pp. 229–234, 2011.

[82] B. Sklar, "Digital Communications Fundamentals and Applications", Second Edition, University of California, Los Angeles, 1988.

[83] S. Gupta, "Design and Implementation of an Optimized Viterbi Decoder", MSc. Thesis, Thapar University, Patiala, 2012.

[84] S. H. Lee and J. K. Kwon, "Turbo Code-Based Error Correction Scheme for Dimmable Visible Light Communication Systems," in IEEE Photonics Technology Letters, vol. 24, no. 17, pp. 1463-1465, Sept.1, 2012, doi: 10.1109/LPT. 2012.2199104.

[85] Gallager, R. G. (1963). Low-Density Parity-Check Codes. MIT Press.

[86] MacKay, D. J. C. (2003). Information Theory, Inference, and Learning Algorithms. Cambridge University Press.

[87] Richardson, T. J., & Urbanke, R. (2008). Modern Coding Theory. Cambridge University Press.

[88] Fossorier, M. P. C. (2007). “Decoding Algorithms for Low-Density Parity-Check Codes.” IEEE Transactions on Information Theory.

[89] MacKay, D. J., & Neal, R. M. (1997). Near Shannon limits performance of low-density parity check codes.Electronics Letters

[90] Johnson, S. J. (2006). Introducing low-density parity-check codes. University of Newcastle, Australia.

[91] Kou, Y., Lin, S., & Fossorier, M. P. C. (2001). Low-density parity-check codes based on finite geometries: a rediscovery and new results. IEEE Transactions on Information Theory.

[92] Richardson, T. J., & Urbanke, R. L. (2001). The capacity of low-density parity-check codes under message-passing decoding. IEEE Transactions on Information Theory.

[93] M. H. Alwan, M. Singh, and H. F. Mahdi, "Performance comparison of turbo codes with LDPC codes and with BCH codes for forward error correcting codes," 2015 IEEE Student Conference on Research and Development (SCOReD), Kuala Lumpur, Malaysia, 2015, pp. 556-560, doi: 10.1109/SCORED.2015.7449398

[94] David, J., C., MacKay., Radford, M., Neal. (1996). 'Near Shannon limit performance of low-density parity-check codes'. Electronics Letters, 33(6):457-458. doi: 10.1049/EL:19970362

[95] B. Barua and S. P. Majumder, "Performance analysis of an LDPC coded FSO communication system with different modulation technique under turbulent condition," 2012 15th International Conference on Computer and Information Technology (ICCIT), Chittagong, Bangladesh, 2012, pp. 240-243, doi: 10.1109/ICCITechn.2012.6509706.

[96] Z. Cui and Z. Wang, "Efficient message passing architecture for high throughput LDPC decoder", Proc. IEEE Int. Symp. Circuits Syst, pp. 917C920, May 2007.

[97] H. Ding, S. Yang, W. Luo and M. Dong, "Design and implementation for high speed LDPC decoder with layered decoding", Proc. WRI Int. Conf. Commun. Mobile Comput, pp. 156C160, Jan. 2009.

[98] M. M. Mansour and N. R. Shanbhag, "High-throughput LDPC decoders", IEEE Trans. Very Large Scale Integr. (VLSI) Syst, vol. 11, no. 6, pp. 976C996, Dec. 2003.

[99] David J.C. MacKay Cavendish Laboratory, 'Parity Check Codes', Cambridge, CB3 OHE, United Kindom.

[100] M. H. Alwan, M. Singh, and H. F. Mahdi, "Performance comparison of turbo codes with LDPC codes and with BCH codes for forward error correcting codes," 2015 IEEE Student Conference on Research and Development (SCOReD), Kuala Lumpur, Malaysia, 2015, pp. 556-560, doi: 10.1109/SCORED.2015.7449398.

[101] Yoni Choukroun, Factor Graph Optimization of Error-Correcting Codes for Belief Propagation Decoding, 2024, Lior Wolf

[102] Li, G., Mu, J., Jiao, X. et al. Enhanced belief propagation decoding of polar codes by adapting the parity-check matrix. J Wireless Com Network 2017, 40 (2017). https://doi.org/10.1186/s13638-017-0825-3

[103] W. Guan and L. Liang, "Check-Belief Propagation Decoding of LDPC Codes," in IEEE Transactions on Communications, vol. 71, no. 12, pp. 6849-6858, Dec. 2023, doi: 10.1109/TCOMM.2023.3308155.

[104] Chung, Kyuhyuk and Jun Heo. "Improved Belief Propagation (BP) Decoding for LDPC Codes with a large number of short cycles." 2006 IEEE 63rd Vehicular Technology Conference 3 (2006): 1464-1466.

[105] Anantharaman, Rajagopal & Kwadiki, Karibasappa & Rao, Vasundara. (2019). 'Hardware Implementation Analysis of Min-Sum Decoders.' Advances in Electrical and Electronic Engineering. 17. 10.15598/aeee.v17i2.3042.

[106] J. Zhang, Z. Yu, J. Zhang, B. Bai, C. Chen and F. Huang, "Speeding up LDPC Decoder by Inter-Frame Pipeline for Wireless Laser Communications," 2019 IEEE/CIC International Conference on Communications in China (ICCC), Changchun, China, 2019, pp. 556-560, doi: 10.1109/ICCChina.2019.8855948.

[107] C. Ding, "Parameters of several classes of BCH codes", IEEE Transactions on Information, vol. 61, no. 10, pp. 5322-5330, Oct. 2015.

[108] Sonali, N. Gupta, A. Dixit and V. K. Jain, "BER Analysis of BCH Codes in Slow Fading Channel with L-PPM and DPPM Scheme," 2018 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), Indore, India, 2018, pp. 1-6, doi: 10.1109/ANTS.2018.8710052

[109] Sonali, N. Gupta, A. Dixit and V. K. Jain, "Performance Analysis of BCH and Repetition Codes in Gamma-Gamma Faded FSO Link," 2019 National Conference on Communications (NCC), Bangalore, India, 2019, pp. 1-5, doi: 10.1109/NCC.2019.8732226.

[110] A. Barrak et al., "Enhancing BER performance limit of BCH and RS codes using multipath diversity", Computers, vol. 6, no. 2, June 2017.

[111] W. Peterson, "Encoding and error-correction procedures for the Bose-Chaudhuri codes," in IRE Transactions on Information Theory, vol. 6, no. 4, pp. 459-470, September 1960, doi: 10.1109/TIT.1960.1057586.

[112] Du Yatao, Zhong Zhi, Zhang Hailong, Gong Fang, Ren Guanghui, and Lan Shulei, 2010. Improved Decoding Algorithm for BCH Code. Information Technology Journal, 9: 1251-1254.

[113] Abdulsada, Hayder. (2013). Design and Implementation of 2-bit BCH Error Correcting Codes using FPGA. JOURNAL OF TELECOMMUNICATIONS. 19. 7

[114] Gnana Prakash, M.Muthamizhan, FPGA Implementation of Bose Chaudhuri Hocquenghem Code (BCH) Encoder and Decoder for Multiple Error Correction Control. International Journal of Innovative Research in Science, Engineering and Technology, Vol. 5, Issue 3, March 2016.

[115] Laurenţiu Mihai, Ionescu Constantin, Anton Ion Tutănescu, Alin Mazăre, Gheorghe Şerban,Hardware Implementation of BCH Error-Correcting Codes on a FPGA. International Journal of Intelligent Computing Research (IJICR), Volume 2, Issue 2, June 2011.

[116] S. Koila, Goutham Simha G D, M. Kulkarni, and U. Sripati, "FPGA implementation of a BCH Codec for free space optical communication system," 2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Delhi, India, 2014, pp. 1822-1826, doi: 10.1109/ICACCI.2014.6968343

[117] Samir Jasim Mohammed, Implementation of Encoder for (31, k) Binary BCH Code based on FPGA for Multiple Error Correction Control. International Journal of Computer Applications (0975 – 8887) Volume 76 – No.11, August 2013.

[118] A.Nazmi Mohammed, R.Mohammed Abaza and H.Moustafa Aly, “Improved Performance of M-ary PPM in Different Free-Space Optical Channels due to Reed Solomon Code Using APD”, International Journal of Scientific & Engineering Research, Vol. 2, Issue 4, pp.1-4,2011.

[119] X. Zhang, "VLSI Architectures for Reed–Solomon Codes: Classic, Nested, Coupled, and Beyond," in IEEE Open Journal of Circuits and Systems, vol. 1, pp. 157-169, 2020, doi: 10.1109/OJCAS.2020.3019403.

[120] G. Durga Priyadharshini, G. Suchitra, ‘PERFORMANCE ANALYSIS OF REED-SOLOMON CODES IN DIGITAL COMMUNICATION SYSTEM USING LABVIEW’, ICTACT JOURNAL ON COMMUNICATION TECHNOLOGY, MARCH 2020, VOLUME: 11, ISSUE: 01.

[121] P. Parvathi and P. R. Prasad, "FPGA based design and implementation of Reed-Solomon encoder & decoder for error detection and correction," 2015 Conference on Power, Control, Communication and Computational Technologies for Sustainable Growth (PCCCTSG), Kurnool, India, 2015, pp. 261-266, doi: 10.1109/PCCCTSG. 2015.7503941.

[122] Y. H. Chen, C. L. Chu, C. C. Yeh, and K. F. Lin, "FPGA Implementation and Verification of Reed-Solomon (63, 47, 8) Code in SDR System," 2012 Second International Conference on Instrumentation, Measurement, Computer, Communication and Control, Harbin, China, 2012, pp. 100-103, doi: 10.1109/IMCCC.2012.30.

[123] Y. H. Chen, C. L. Chu and C. C. Yeh, "FPGA implementation and verification of Reed-Solomon (31, 15, 8) code in SDR system," Proceedings of 2012 2nd International Conference on Computer Science and Network Technology, Changchun, China, 2012, pp. 465-468, doi: 10.1109/ICCSNT.2012.6525978.

[124] S. S. Muhammad, T. Javornik, I. Jelovcan, E. Leitgeb, and O. Koudelka, "Reed Solomon coded PPM for Terrestrial FSO Links," 2007 International Conference on Electrical Engineering, Lahore, Pakistan, 2007, pp. 1-5, doi: 10.1109/ICEE.2007.4287281.

[125] R. W. Hamming, "Error detecting and error correcting codes," in The Bell System Technical Journal, vol. 29, no. 2, pp. 147-160, April 1950, doi: 10.1002/j.1538-7305.1950.tb00463.x.

[126] N. H. Sabbry and A. Levina, "Hamming Code: An Analysis of its Reliability and Efficiency in Computer Networks," 2023 International Conference Automatics and Informatics (ICAI), Varna, Bulgaria, 2023, pp. 356-361, doi: 10.1109/ICAI58806.2023.10339080.

[127] Achmad Fauzi, Nurhayati, Robbi Rahim, BIT ERROR DETECTION AND CORRECTION WITH HAMMING CODE ALGORITHM.

[128] Makhloufi, Assaad El, Samir El Adib and Naoufal Raissouni. “Highly Efficient Security Level Implementation in Radiation-Tolerance FPGA Using a Combination of AES Algorithm and Hamming Code: LST-SW Case.” International Journal of Electrical and Electronic Engineering & Telecommunications (2023): n. pag.

[129] Basirah, Siti and Amir Hamzah. “Hamming Code Using Field Programmable Gate Array (FPGA).” (2014).

[130] C. Hillier and V. Balyan, “Error detection and correction on-board nanosatellites using hamming codes,” Journal of Electrical and Computer Engineering, vol. 2019, #e3905094, Feb. 2019, doi: 10.1155/2019/3905094.

[131] Jaya Kumari Verma, Shipra Varshney, Performance assessment of Hamming Code, Proc. of the International Conference on Science and Engineering (ICSE 2011).

[132] N. Fujita, R. Kobayashi, Y. Yamaguchi, and T. Boku, "Parallel Processing on FPGA Combining Computation and Communication in OpenCL Programming," 2019 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), Rio de Janeiro, Brazil, 2019, pp. 479-488, doi: 10.1109/IPDPSW.2019.00089.

[133] Rene Mueller, Jens Teubner, Gustavo Alonso.Data Processing on FPGAs in Systems Group, Department of Computer Science, ETH Zurich, Switzerland.

[134] Anbuselvi, T., and A. Arockia Bazil Raj. "Design and verification of pipelined parallel architecture implementation in FPGA for bit error rate tester." International Journal of Research in Engineering and Technology 3.2 (2014): 28-34.

[135] Nunez-Yanez, J. L., & Lou, J. (2014). FPGA architecture and tools for digital signal processing applications.Journal of Signal Processing Systems, 76(2-3), 201-218. doi:10.1007/s11265-013-0824-2

[136] Muthiah, D., and A. Arockia Bazil Raj. "Implementation of high-speed LFSR design with parallel architectures." 2012 International Conference on Computing, Communication and Applications. IEEE, 2012.

[137] Compton, K., & Hauck, S. (2002). Reconfigurable computing: A survey of systems and software. ACM Computing Surveys (CSUR), 34(2), 171-210. doi:10.1145/508352.508353

[138] Villasenor, J., & Mangione-Smith, W. H. (1997). Configurable computing. Scientific American, 276(3), 70-75.

[139] Anbuselvi, T., and A. Arockia Bazil Raj. "Design and verification of pipelined parallel architecture implementation in FPGA for bit error rate tester." International Journal of Research in Engineering and Technology 3.2 (2014): 28-34.

[140] Weinhardt, M., & Luk, W. (2001). Pipeline vectorization. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 20(2), 234-248. doi:10.1109/43.902982.

[141] Lau, F. C., Leong, P. H., Siu, W. C., & Kwok, W. K. (2000). Energy-efficient hardware acceleration of adaptive filters. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 8(1), 102-107. doi:10.1109/92.823059

[142] Hentschke, R., & Kreutz, M. (2004). Reconfigurable architectures for wireless communications. Journal of Systems Architecture, 50(2-3), 113-129. doi:10.1016/j.sysarc.2003.07.003.

[143] Blome, J. A., Wu, J., & Zhang, Y. (2007). LDPC codes on FPGAs: High-performance error correction for modern communication systems. IEEE Communications Magazine, 45(3), 104-110. doi:10.1109/MCOM.2007.361575.

[144] Blome, J. A., Wu, J., & Zhang, Y. (2007). LDPC codes on FPGAs: High-performance error correction for modern communication systems. IEEE Communications Magazine, 45(3), 104-110. doi:10.1109/MCOM.2007.361575.

[145] Diniz, P. S., Silva, E. A. B., & Netto, S. L. (2010). Digital signal processing: System analysis and design. Cambridge University Press.

[146] Anderson, J. H., & Smith, M. D. (2005). The role of FPGAs in reconfigurable computing: A survey of the state of the art. ACM Transactions on Design Automation of Electronic Systems (TODAES), 11(3), 442-472. doi:10.1145/1142980.1142984.

[147] Drimer, S., & Kuhn, M. G. (2008). A protocol for secure remote updates of FPGA configurations. In Proceedings of the 16th Annual International Symposium on Field-Programmable Custom Computing Machines (pp. 97-106). IEEE. doi:10.1109/FCCM.2008.44.

[148] D. J. T. Heatley, D. R. Wisely, I. Neild, and P. Cochrane, "Optical wireless: the story so far," in IEEE Communications Magazine, vol. 36, no. 12, pp. 72-74, Dec. 1998, doi: 10.1109/35.735881.

[149] Shu Lin and Daniel J.Costello, “Error Control Coding: Fundamentals and Applications”, Englewood Cliffs, NJ: Prentice Hall, Chapter-6, page no.141-183, 1983.

[150] Todd K Moon, “Error Correction Coding; Mathematical Methods and Algorithms,” John Wiley & Sons, Hoboken, New Jersey, Chapter-6, page no.235-280, May 2005.

[151] Singh, Shree Prakash, Shrivastava, Shreesh Kumar, Sengar, Sujata and Nath, Soyinka. "Fiberless optical communication: issues and challenges" Journal of Optical Communications, 2023. https://doi.org/10.1515/joc-2022-0280

[152] Xiang Yi, Zengji Liu, Peng Yue, Optical scintillations and fade statistics for FSO communications through moderate-to-strong non-Kolmogorov turbulence, Optics & Laser Technology, Volume 47,2013, Pages 199-207, ISSN 0030-3992,https://doi.org/10.1016/j.optlastec.2012.08.008.

[153] Raj, A. Arockia Bazil, and J. Arputha Vijaya Selvi. "Comparison of different models for ground-level atmospheric attenuation prediction with new models according to local weather data for FSO applications." Journal of optical communications. 36.2 (2015): 181.