DOI: https://doi.org/10.4038/engineer.v58i2.7691
ENGINEER - Vol. LVIII, No. 02, pp. [25-34], 2025

A MIMO-OFDM Framework for Video Streaming: An Error-Resilient FPGA Design with an Economical Perspective

Patikiri Arachchige Don Shehan Nilmantha Wijesekara

Abstract: The transmission of modulated data in several spatial streams, each of which is made up of many orthogonal subcarriers, is fundamental to Multiple Input Multiple Output (MIMO)-Orthogonal Frequency Division Multiplexing (OFDM). Nevertheless, we discover that current hardware implementations of MIMO-OFDM systems are less appropriate for use, particularly in low-power embedded devices, and need significant hardware resources. In this work, in order to fill this research gap, we develop and build a low-cost MIMO-OFDM system that may be used for video streaming in a Field Programmable Gate Array (FPGA) while still meeting the high-performance criteria required for video communication. In particular, we use configurable pilot symbols and a modulation approach that includes a 3-level BCH code that can handle up to 5-bit corrections per pixel, as well as an intra- and inter-pixel interleaver. To make the error-correcting code less complicated, we employ a simpler third-order generating polynomial and a precalculated Lookup Table (LUT). Karatsuba-inspired multiplications with fixed-point computations, shift register-driven correlators, and LUT-driven square root approximation are used at the receiver for synchronization. By using pipelining inside OFDM phases, multiplexing in the time domain among the four streams, we significantly minimize hardware complexity. In order to determine whether to decompose singular values for calculating the channel matrix inverse, we first approximate the channel matrix utilizing the least squares technique. Then, we perform channel equalization adaptively with reduced noise magnification by calculating the infinity norm difference among the current channel matrix and the most recent recognized channel matrix. Ultimately, the system was put into practice using Artix 7 FPGAs, validated by thorough simulations, and tested for actual data transfer. The outcomes demonstrate that the suggested system is error-resilient, economical, and energy-efficient when compared to current methods.

Keywords: MIMO, video streaming, OFDM, embedded systems, FPGA, hardware-efficiency

1. Introduction

A cable video transmission system's hardware implementation is difficult as it inherently uses a lot of memory and computing power. Video transmission quality of service requires high throughput and low latency [1]. Furthermore, visual communication requires a lot of hardware resources, which also results in a significant power consumption.

There have been a number of earlier studies on 360^0 video communication as of this writing. Xu and colleagues developed an FPGA-based real-time circular 360^0 video communication framework [2]. However, they have presented a technique for simultaneous picture processing here, while our approach uses spatial multiplexing in conjunction with OFDM, which, as demonstrated by the simulation results, improves efficiency while lowering complexity. Although different from a hardware level, Huo et al. created a concept for a RoI-centered 360^0 video communication that prioritizes error prevention above data transfer speed [3]. Techniques like data aggregation [4] and compression [5] are necessary for typical visual communication in non-resource-constrained contexts, but these methods are less practical in embedded systems with limited resources. Fu et al. [6] have devised a mapping technique representing panoramic video to enhance the visual perception, steadiness, and compression performance. In [7], Riegler et al. presented a system that improved the panoramic video quality at a minor expense, resulting in a lower level of experience quality than a maximum-resolution panoramic implementation. It has been demonstrated that careful design is

***Eng. P.A.D.S.N. Wijesekara**, AMIE(SL), B.Sc. Eng. (Hons) (Ruhuna), Ph.D. (reading, Ruhuna), Lecturer in Electrical & Information Engineering, University of Ruhuna,*
Email: nilmantha@eie.ruh.ac.lk
ORCID ID: http://orcid.org/0000-0002-3045-1596

necessary for the hardware implementation of high-throughput communication frameworks to prevent communication demands from overtaxing the hardware's limited resources [8].

The issues are further enhanced when video transmission takes place wirelessly since it requires a large wireless bandwidth and should have a manageable error rate [9]. Since these channels are often dynamic and multi-path propagating and fading channels, interferences among symbols, which are caused by frequency-dependent attenuation, can happen in them. This is the reason why burst bit-errors are common [10]. The difficulties of video transmission hardware implementation are further increased when more parts are incorporated into the hardware design to address these wireless communication issues.

In order to boost wireless data transmission reliability and speed without requiring additional transmission power, the OFDM framework is a highly potential modulation method that uses multiple transceivers for both the transmitting and receiving ends [11]. Through the concurrent transmission of OFDM streams, MIMO-OFDM substantially increases the total communication throughput. However, because of their high latency and resource consumption, current FPGA-based OFDM systems are less appropriate for video transmission [12]. There have been proposals for reducing peak-to-average power ratio in multiuser OFDM, and it has been assessed for computational complexity [13]. In video broadcasting, impulse noise arising from bursty transmission errors in OFDM has been shown to be mitigated by increasing the transmission power [14]. Moreover, COFDM has been utilized for HDTV video transmission and evaluated under imperfect channel conditions [15].

Therefore, we design and implement a low-cost FPGA-based MIMO-OFDM system suitable for embedded systems, using multiple strategies to reduce the hardware implementation cost; however, adhering to high error correction capability, but with the simplest design, reducing unnecessary processing delays, and being able to satisfy high throughput requirements of video communication [16]. Thus, this paper fills a void in the existing body of literature by designing and experimentally validating a MIMO-OFDM framework.

2. Materials and Methods

2.1 Video transmitter

Based on the feedback (*RoI_select*) obtained from the feedback channel, one of the unprocessed data frames taken from several cameras is chosen, as seen in Fig. 1. The block RAM: *Frame_Buffer_TX* will receive this extracted frame for storage. The *OV7670_Capture module* controls when a frame is captured from the camera. Configuring the camera via the I2C interface is done via the *OV7670_controller* module. The OFDM system will read the data from the frame buffer to create OFDM symbols, which will then be sent across the radio frequency transmitters.

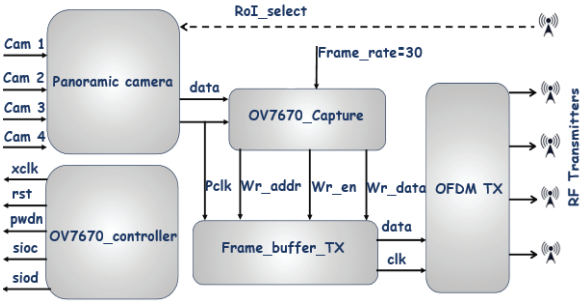


Figure 1 - Video transmitter

2.2 Video receiver

The *OFDM RX* module will first transform the received OFDM symbols into matching data blocks, as seen in Fig. 2. A generating module for write address is then used to obtain a write address to the receiver's frame buffer (*Frame_Buffer_RX*), which is where the *OFDM RX*'s data output is written. While the generating module for read address creates read addresses to read the data contained in the frame buffer so that the *RGB* module may display it, the *VGA* module creates the lateral and upright synchronization signals.

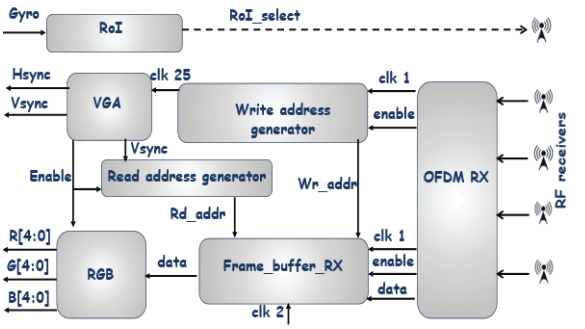


Figure 2 - Video receiver

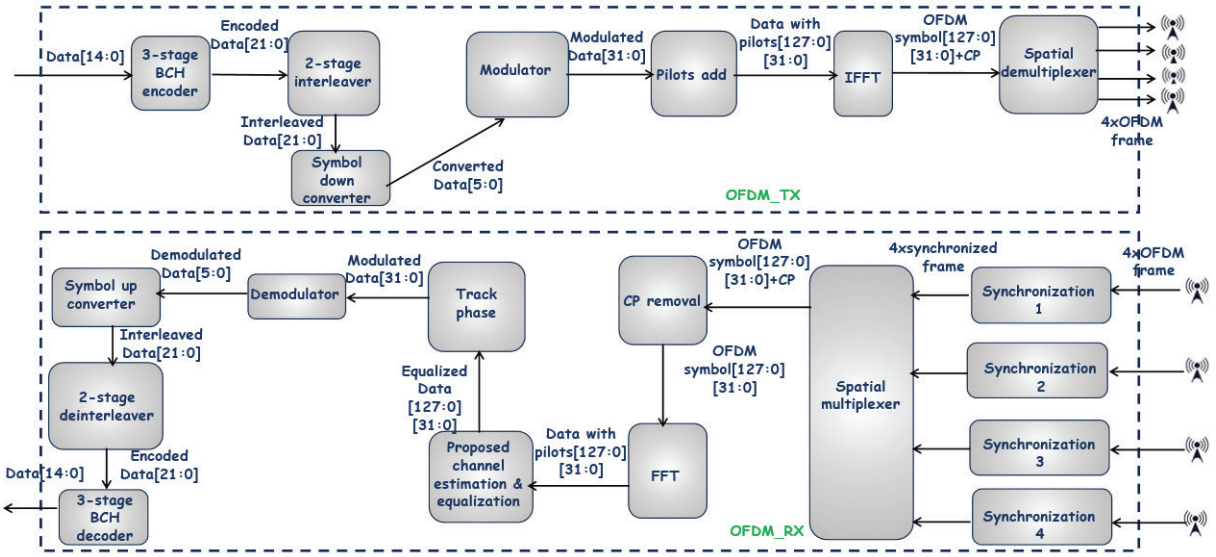


Figure 3 - Design of the suggested MIMO-OFDM framework

2.3 Proposed MIMO-OFDM framework

As seen in Fig. 3, the OFDM system is composed of the following components: data modulators and demodulators, spatial MIMO stream demultiplexers (1 to 4) at transmitter and multiplexers (4 to 1) at receiver, four synchronization occurrences at receiver, channel approximation and channel equalization at receiver, phase tracking modules at the receiver, a BCH encoder (14 to 21) at transmitter and a BCH decoder (21 to 14) at receiver, a 2-level interleaver at transmitter and a deinterleaver at receiver, an IFFT at transmitter and a FFT at receiver, and pilots addition at transmitter. The symbol down converter module in Fig. 3 transforms the interleaved 22-bit data into a symbol size suitable for the modulation that follows. The OFDM_RX's symbol up-converter does the opposite task. Below is a summary of the OFDM system's architecture.

- In order to share the same assets for OFDM symbol creation without generating four occurrences of them for four streams, we employ an acquired clock module for the system, which includes time domain multiplexing of the OFDM symbols. The system is programmable for modulation technique and pilot symbols.
- It consists of a new block-level (inter-pixel) and bit-level (intra-pixel) interleaver design for 2x4 pixels (see Fig. 4). The proposed hierarchical interleaving technique is designed for the video transmission to protect pixels from burst errors. Note that, as demonstrated using the red colour in Fig. 4, before interleaving, there is a 4-bit burst error between pixel 1 and pixel 2. In the first stage of interleaving (block level), the columns 2 and 4 are swapped. In the second stage (bit level), bits are interleaved, such that there is only 1 bit error per pixel.
- Current OFDM synchronizers have complex signal processing algorithms and multiplicative correlators that demand high processing power and can lead to higher latency. Using shift register-driven correlators, LUT-driven square root computation for signal magnitude estimate, and shift register cascades for delaying the received signal, we develop an effective OFDM synchronizer. Additionally, for complex number multiplication, we employ Karatsuba-inspired multiplication during phase tracking. For complex number multiplication, it requires 4 real multiplications and two real additions, and the requirement is reduced to 3 real multiplications and 2 real additions in Karatsuba-inspired multiplication. Thus, we do not use the existing IP cores and achieve complex multiplication using the Karatsuba-inspired technique
- It uses a bitwise negation-driven method in decomposing Singular Values (SV) for inverse matrix computation for channel equalization utilizing a zero-forcing detector (see Equation 1) and a low-complicated simple transposition. We use a zero-forcing detector to drive the interference

among the MIMO streams towards zero. To determine whether to go forward with decomposing SV or not, we perform adaptive equalization by storing the most recently estimated channel matrix (I) and related inverses in a LUT. We then calculate the infinite norm of difference between the approximated channel matrix and LUT entries (see Equation 2). Additionally, in order to save the intermediary pilot matrix (B) in channel approximation utilizing least squares, we use a pre-calculated look-up table to estimate the channel matrix (see Equation 3). Additionally, we use the signal-to-noise ratio (see Equation 4) to standardize noise variance and reduce noise magnification in this procedure.

$$I = PFV^{Hermetian} \quad \ldots(1)$$

The unitary matrix P in Equation (1) includes the eigenvectors of II^T, the diagonal matrix F holds the I singular values, and the eigenvectors of I^TI are included in the V matrix.

$$\left\|E - \left|I^{Hermetian} I\right|\right\|_{\infty} = \max_{1\le i\le 4} \sum_{j=1}^{4} \left|E(ij) - I^{Hermetian} I(ij)\right| \quad \ldots(2)$$

E is the final estimated channel matrix in Equation (2).

$$B = X_{pilot}^{Hermitian} (X_{pilot} I_{pilot}^{Hermitian})^{-1} \quad \ldots(3)$$

$$F^{-1}(ii) = \frac{\lambda_i}{{\lambda_i}^2 + \frac{\mu}{SNR_i}} \quad \ldots(4)$$

In Equation (4), λ_i is defined as the i^{th} singular value of I, where $0 \le \mu \le 1$. Note that noise amplification in the zero forcing detector is reduced by using the signal-to-noise ratio (SNR). Here, note that the proposed approach prevents amplification of noise in low SNR streams, especially when the singular values are low.

- Within the OFDM phases, we employ pipelining, multiplexing in the time domain, fixed-point computations, value truncation, and hardware component sharing among four spatial streams.
- For the composite 3-tier BCH (22, 15) code, a mixed computational approach employing a concurrent parity bit generation and a precalculated LUT for three bits (first stage) and two less complicated cubic generator polynomials (see Equation 5) for the rest of the twelve bits (six bits each in the remaining stages) is used. We select BCH codes owing to lower complexity (due to polynomial arithmetic over finite fields), in comparison to Turbo (due to iterative decoding), Reed-Solomon (due to polynomial interpolation), and LDPC (due to iterative belief propagation algorithms) codes [17]. However, the complexity of the BCH code also increases with the degree of the generator polynomial; thus, we have 3-tier lower complex code with LUT and simpler generator polynomials.

$$h(y) = y^3 + y^2 + y + 1 \quad \ldots(5)$$

In a 20 MHz channel with a sub-carrier gap of 156.25 kHz at a 2.4 GHz carrier frequency, FFT and IFFT are implemented with 128 points to reduce frequency-selective fading, interferences among symbols, Doppler shifts, etc., due to the spreading of data in narrowband subcarriers. Unlike some current systems, we employ clock gating to minimize time-varying power consumption without designing it as a lower-point FFT that forfeits prior gains. On the other hand, selection of a large size FFT can significantly increase the computational complexity, so we avoided it. Moreover, we use a 16 sub-carrier length cyclic prefix to strike a balance between spectral efficiency and mitigation of ISI. Moreover, the system is designed as a 4x4 system, having 4 transmitter and receiver antennas to strike a balance in throughput and hardware complexity.

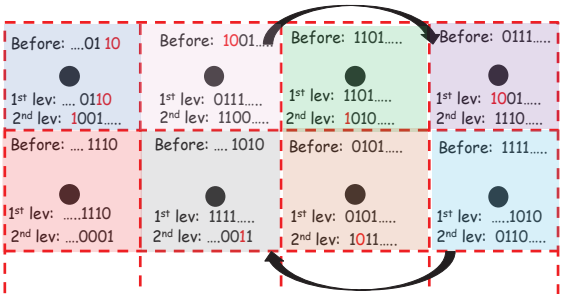


Figure 4 - Hierarchical interleaving

3. Results and Discussion

3.1 Experimental setup

Fig. 5 displays the system in use. Observe that two Artix 7 FPGAs are used to build the OFDM framework, while two Spartan 6 units are used to implement the other components connected to video transmission.

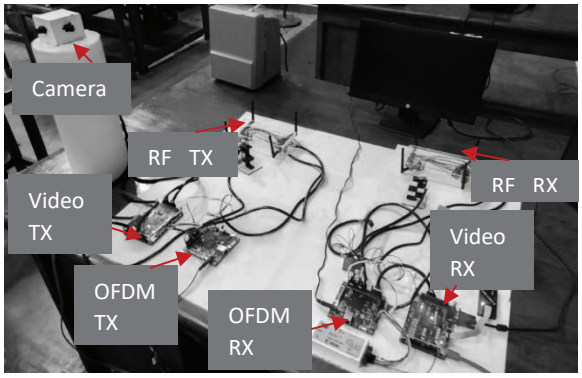


Figure 5 - Hardware implementation

During the transmission and receipt of OFDM frames, the linked computer will receive power consumption figures from programmable power sources that power FPGAs, as seen in Fig. 6. Likewise, the associated symbol along with the time information will be saved in the linked computer at the conclusion of the OFDM signal's transmission and reception. Furthermore, the radio frequency receivers will output the signal power detected at the receiver. The received signal strength from the radio frequency receiver is more precise than the OFDM approximation, even though it may be calculated while synchronizing, leveraging the in-phase and quadrature-phase components. By altering the displacement from the transmitters to the receivers, we may change the power of the received signal. The performance evaluation measures will thereafter be calculated using this data. We transfer 11,000 frames between the transmitter side and receiver side in each experiment in order to obtain one data point. Note that the same physical setup was used for different experiments. In order to vary the received signal strength from -20 dBm to -60 dBm, the transmission distance had to vary approximately from 11 m to 178 m.

3.2 Verification of proposed BCH code

Fig. 7 displays the channel encoder-decoder validation result.

A 5-bit data-swapping module has been added to the experiment between the encoder and decoder. Since the decoder output (*DECODER_DAT_O*) remains unchanged compared to the input data at the encoder, at the end of one clock period subjected to 5-bit channel faults, the data provided to the BCH encoder (*DAT_I*) has been successfully decoded, as shown in Fig. 7. Note that in Fig. 7, the BCH encoder's output (*ENCODER_DAT_O*) is flipped as *FLIPPED_DAT_O,* which is provided to the decoder to obtain the transmitted data at the input of the encoder. The other signals depicted in Fig. 7 are the syndromes, parities, and error flags.

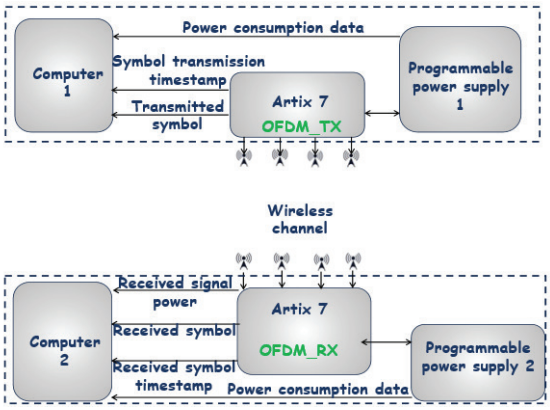


Figure 6 - Experimental setup architecture

3.3 Verification of proposed interleaver

Fig. 8 displays the interleaver-deinterleaver design verification result.

The interleaver is intimately linked with the deinterleaver in this experiment. Since the deinterleaved output (*DEINTERLEAVER_DAT_O*) is identical to the input in the precise same sequence as the input data, the interleaver data input (*DAT_I*) is appropriately deinterleaved, as seen in Fig. 8. The interleaver-deinterleaver design is thus verified.

3.4 Verification of uncoded MIMO-OFDM framework

Fig. 9 illustrates how input data (*dat_in*) is sent to the system. It is necessary to send the complex output signal, *dat_out*, via the channel.

First, a chosen modulation method from 64-QAM, 16-QAM, or QPSK is applied to the input data and pilot symbols. Fig. 9 shows the simulation results for 64-QAM modulation. The output data (*dat_out*) is then obtained by passing the complex symbols through an *IFFT* module *(IFFT_datout)* and adding a cyclic prefix, as shown in the same figure. Note that, as seen from Fig. 9(a), for the modulated data output of the selected modulation scheme (*Modulator_datout*), pilots are added

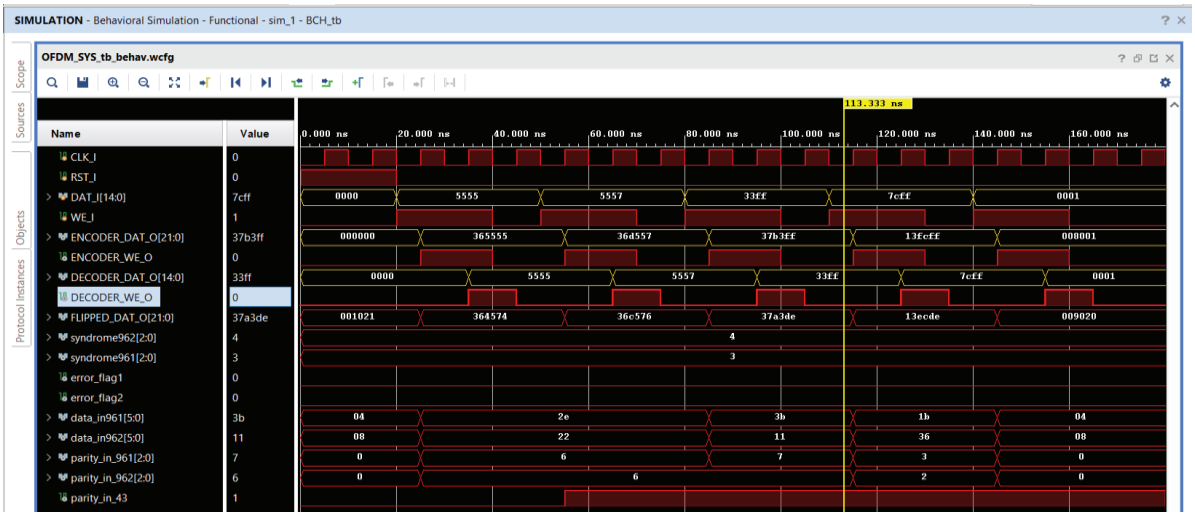


Figure 7 - BCH code simulation

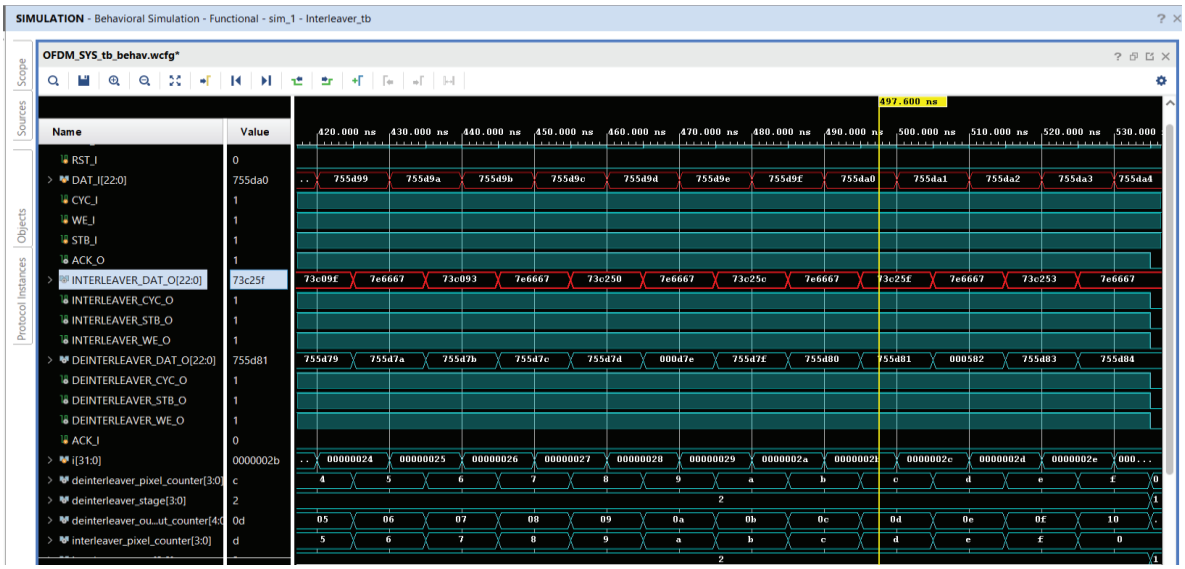


Figure 8 - Interleaver simulation

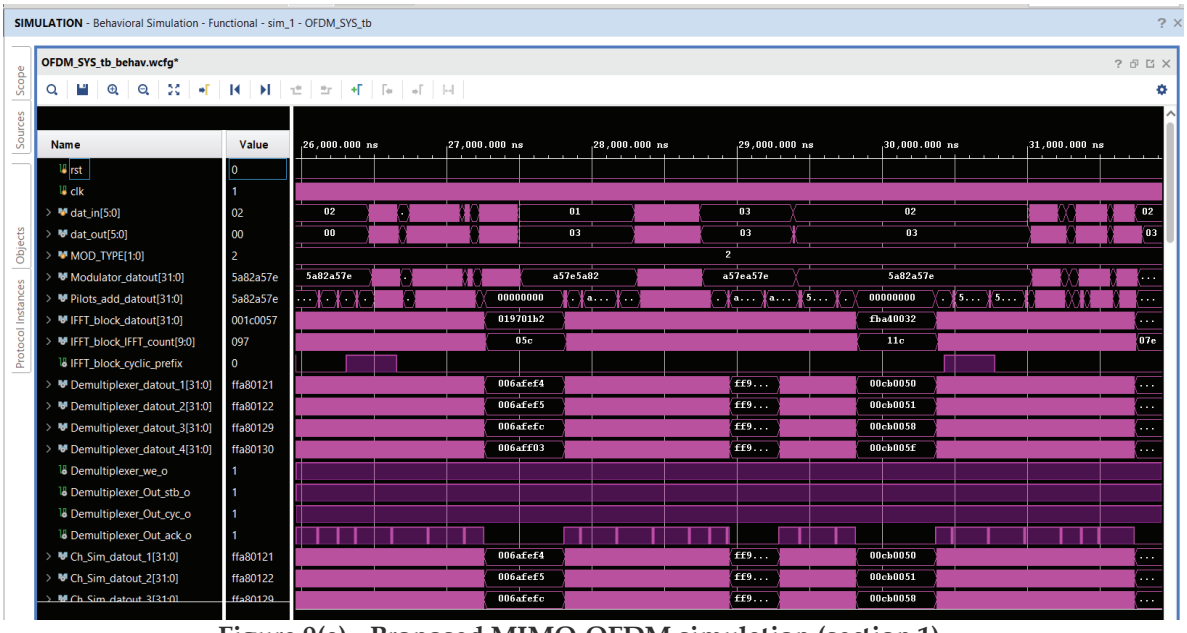


Figure 9(a) - Proposed MIMO-OFDM simulation (section 1)

Figure 9(b) - Proposed MIMO-OFDM simulation (section 2)

(*pilots_add_datout*), sent through the IFFT module, and a cyclic prefix is added. Finally, the IFFT output is demultiplexed into 4 streams (*Demultiplex_datout_1 to 4*) to be transmitted through the 4 transmitters.

As can be seen from the simulation results in Figures 9(a) and 9(b), the system functions optimally because the received data symbols are appropriately synced, equalized, tracked, and demodulated to provide output data with the identical signal pattern as that which was sent by the transmitter.

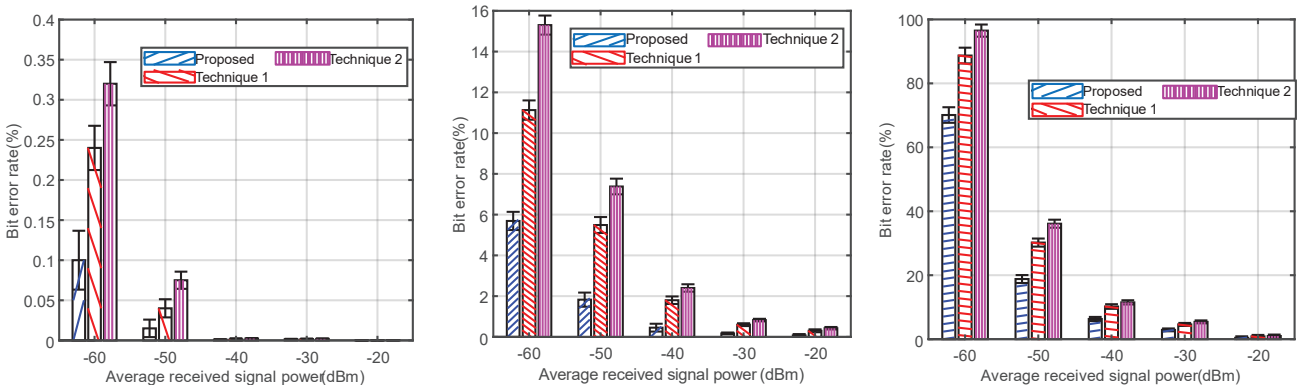
The output signal following the demodulation operation is the *dat_out*, which is the signal that is highlighted. The intended deinterleaver and BCH code will fix the bit faults that are present in the demodulated data in this case. Using four channel simulator instances (*Ch_Sim_datout1*, etc.) for four spatial streams (*Demultiplexer_datout_1*, etc.), we can create bit errors (for the simulation, we have demonstrated for an ideal channel, as seen in Fig. 9). It should be noted that each synchronized data output (*Synchronization_datout1*, etc.) that has had the preamble removed is multiplexed. Cyclic prefixes are then eliminated (see *Cyclic_prefix_removal_datout*) and sent to the FFT occurrence, whose signal output (*FFT_datout*) is used for the proposed channel equalization by approximating a pseudo-inverse matrix and utilizing the zero-forcing technique by decomposing singular values (see output *Channel_estimation_equalization_datout*). The phase of the equalized data symbol is then monitored, and the result is shown in *Track_phase_datout* in Fig. 9. Lastly, as seen in Fig. 9, the phase-estimated data symbols are converted back to down-converted data to yield the demodulated output *dat_out*. For the system in a perfect (*ideal*) channel, the uncoded OFDM error output was zero, indicating that the system was operating as intended.

However, the above validation is for the uncoded MIMO-OFDM transceiver using simulations. We evaluate the hardware-implemented coded system's performance under a real wireless channel using the following subsections.

3.5 Error resilience assessment

The suggested strategy has a lower error rate than current methods, as seen in Fig. 10. The primary cause of this is the two-level interleaver architecture and the three-stage optimized BCH code, which can correct up to five bit corruptions for every 15-bit pixel, incorporating burst faults that can happen in wireless transmission.

The findings show that although convolutional encoders were employed in techniques 1 and 2, they were not as successful as the suggested design's hierarchical BCH code integrated with a 2-tier interleaver for pixel burst error reduction. Furthermore, the timing mistakes generated by the suggested approach are significantly reduced since we employ parallel synchronization modules for each stream in



(a) QPSK (b) 16-QAM (c) 64-QAM

Figure 10 - Bit error rate assessment

comparison to the baseline approaches. This experiment also suggests that the timing and synchronization errors of the proposed technique are quite low.

3.6 Hardware cost

The hardware costs of the suggested 4x4 MIMO-OFDM framework are contrasted with those of alternative systems in Table 1.

Table 1 - Hardware resource consumption comparison

Resource	Technique 1 [12]	Technique 2 [18, 19]	Proposed
Look up tables	11190	31752	8566
Flip flops	21253	28392	11291
Block RAMs	173	94	4
DSP units	95	211	32

The FF, LUT, BRAM, and DSP unit consumption of the suggested technique are significantly lower than those of other comparable systems, as shown in Table 1. As a result, the suggested approach uses far less hardware and is better than current methods.

3.7 Power consumption assessment

As seen in Fig. 11, it is evident that the proposed MIMO-OFDM framework uses less power than current methods. This is mostly because the suggested system consumes fewer dynamic computing resources because of features like value approximations, look-up table-based calculations, interleaver design, hardware-efficient code design, Karatsuba- inspired complex multiplications, and more. However, current implementations do not have these hardware improvements, which results in a comparatively high power consumption.

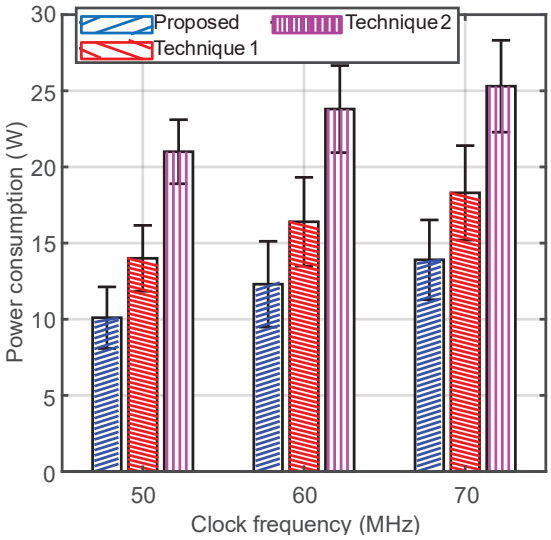


Figure 11 - Power consumption assessment

4. Conclusion

A cost-efficient design of a 4x4 MIMO-OFDM-assisted video streaming system based on FPGA was reported in this study. A cost-efficient matrix inversion estimation for channel equalization, programmable modulation and pilot sequences, a composite 3-tier BCH code (22, 15) to prevent pixel-level faults, a two-stage interleaver to prevent burst errors, and ROI-driven panoramic video recording were all included in the system design. The results show that this system has a lower error rate than current MIMO-OFDM frameworks under actual

channel circumstances. Additionally, the findings demonstrated that the framework implementation resulted in the lowest power usage and the lowest hardware cost. Therefore, it can be said that the suggested approach is quite successful for embedded video communication.

Acknowledgement

The author would like to thank the research coordinator for the immense support provided to carry out the research project.

References

1. Song, D., and Chen, C. W., "Scalable H.264/AVC video transmission over MIMO wireless systems with adaptive channel selection based on partial channel information," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 17, pp. 1218–1226, September 2007.

2. Xu, Y., Zhou, Q., Gong, L., Zhu, M., Ding, X., and Teng, R. K. F., "Highspeed simultaneous image distortion correction transformations for a multicamera cylindrical panorama real-time video system using FPGA," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 24, pp. 1061–1069, June 2014.

3. Huo, Y., Wang, X., Zhang, P., Jiang, J., and Hanzo, L., "Unequal Error Protection Aided Region of Interest Aware Wireless Panoramic Video," in *IEEE Access*, Vol. 7, pp. 80262-80276, 2019.

4. Seneviratne, C., Wijesekara, P.A.D.S.N., and Leung, H., "Performance analysis of distributed estimation for data fusion using a statistical approach in smart grid noisy wireless sensor networks," *Sensors*, Vol. 20, No. 2, pp. 567, 2020.

5. Ma, Y., and Chang, Y., "A Cross-Layer H.264/AVC Video Transmission Method Over Wireless OFDM," *2007 International Workshop on Cross Layer Design*, Jinan, 2007, pp. 100-104.

6. Fu, C. W., Wan, L., Wong, T. T., and Leung, C. S., "The rhombic dodecahedron map: An efficient scheme for encoding panoramic video," *IEEE Transactions on Multimedia*, Vol. 11, pp. 634–644, June 2009.

7. Gaddam, V.R., Riegler, M., Eg, R., Griwodz, C., and Halvorsen, P., "Tiling in interactive panoramic video: Approaches and evaluation," *IEEE Transactions on Multimedia*, Vol. 18, pp. 1819–1831, September 2016.

8. Wijesekara, P.A.D.S.N., Sangeeth, W.M.A.K., Perera, H.S.C., and Jayasundere, N.D., "Underwater Acoustic Digital Communication Channel for an UROV," in *5th Annual Research Symposium (ARS2018)*, 2018, p. E17.

9. Huo, Y., Hellge, C., Wiegand, T., and Hanzo, L., "A tutorial and review on inter-layer FEC coded layered video streaming," *IEEE Communications Surveys and Tutorials*, Vol. 17, pp. 1166–1207, 2nd-quarter 2015.

10. El-Fouly, F.H., Ramadan, R.A., Abd El-Samie, F.E., Kachout, M., Alzahrani, A.J. and Alshudukhi, J.S., "Burst channel error reduction based on interleaving for efficient high-speed wireless communication," *Applied Sciences*, Vol. *12*, No. 7, p.3500.

11. Su, G., Han, Z., Wu, M., and Liu, K. J. R., "A Scalable Multiuser Framework for Video Over OFDM Networks: Fairness and Efficiency," in *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 16, No. 10, pp. 1217-1231, Oct. 2006.

12. Haene, S., Perels, D., and Burg, A., "A real-time 4-stream MIMO-OFDM transceiver: system design, FPGA implementation, and characterization," *IEEE Journal on Selected Areas in Communications*, Vol. 26, No. 6, pp.877-889, 2008.

13. Jiang, T., and Wu, Y., "An overview: Peak-to-average power ratio reduction techniques for OFDM signals", *IEEE Trans. Broadcasting*, Vol. 54, No. 2, pp. 257-268, Jun. 2008

14. Armstrong, J., and Suraweera, H. A., "Impulse noise mitigation for OFDM using decision directed noise estimation," in *Proc. Eighth IEEE International Symposium on Spread Spectrum Techniques and Applications*, Sydney, Australia, 2004, pp. 174-178.

15. Zou, W. Y., and Wu, Y., "COFDM: An overview", *IEEE Trans. Broadcasting*, Vol. 41, No. 1, pp. 1-8, Mar. 1995.

16. Wijesekara, P.A.D.S.N., "Towards Efficient and Reliable Video Communication: A Survey on Scalability, Error Protection, and Multicasting," *The Indonesian Journal of Computer Science*, Vol. 14, No. 1, pp. 34-67, 2025.

17. Alwan, M.H., Singh, M., and Mahdi, H.F., "Performance comparison of turbo codes with LDPC codes and with BCH codes for forward error correcting codes," In Proc. *2015 IEEE Student Conference on Research and Development (SCOReD)*, Dec.2015, pp. 556-560.

18. Pereira, T., Violas, M., Lourenço, J.L., Gameiro, A., Silva, A., and Ribeiro, C., "An FPGA

implementation of OFDM transceiver for LTE applications," *Intrnl. Journal On Advances in Systems and Measurements*, Vol. 6, No. 1-2, pp.224-234, 2013.

19. Srinandhini, K., and Vaithianathan, V., "FPGA implementation of MIMO-OFDM transceiver," In *2014 International Conference on Communication and Signal Processing*, Apr. 2014, pp. 353-357.