International Journal of Electrical and Computer Engineering (IJECE)
Vol. 15, No. 4, August 2025, pp. 3813~3823
ISSN: 2088-8708, DOI: 10.11591/ijece.v15i4.pp3813-3823

High-speed field-programmable gate array implementation for mmWave orthogonal frequency-division multiplexing transmitters: design and evaluation

Kidsanapong Puntsri[1], Bussakorn Bunsri[1], Puripong Suthisopapan[2]
[1]Department of Electronics and Telecommunication Engineering, Rajamangala University of Technology Isan, Khon Kaen Campus (RMUTI KKC), Khon Kaen, Thailand
[2]Department of Electrical Engineering, Faculty of Engineering, Khon Kaen University (KKU), Khon Kaen, Thailand

Article Info

Article history:

Received Sep 6, 2024
Revised Mar 31, 2025
Accepted May 24, 2025

Keywords:

Field-programmable gate array
Inverse fast Fourier transform
Millimeter wave
Orthogonal frequency-division multiplexing
Radix-2 algorithm
Wireless communications

ABSTRACT

This paper presents a field-programmable gate array (FPGA)-based implementation of an orthogonal frequency-division multiplexing (OFDM) transmitter signal processing chain optimized for high-speed millimeter wave (mmWave) communication systems. The design prioritizes real-time processing efficiency and flexibility. A high-throughput 2048-point inverse fast Fourier transform (IFFT) module, realized using a Radix-2 algorithm, forms the core of the design, showcasing efficient hardware resource utilization. The implementation further includes cyclic prefix (CP) insertion and configurable support for various quadrature amplitude modulation (QAM) orders and pilot arrangements. The design is implemented in VHSIC hardware description language (VHDL) using Vivado 2020 and evaluated on the Zynq UltraScale+ RFSoC ZCU111 evaluation kit. The processing pipeline employs eight parallel lanes for concurrent data computation. Experimental results demonstrate a mean squared error (MSE) of only 0.00013 between the FPGA-generated waveform and its MATLAB-simulated counterpart. Additionally, post-implementation resource utilization analysis shows efficient usage of FPGA resources. These findings validate the efficacy and real-time capability of the proposed FPGA-based OFDM transmitter leverages parallelism and high-speed architecture to efficiently process massive data streams, making it suitable for a wide range of mmWave OFDM applications. In contrast to recent works that focus on lower-order IFFT modules, this paper employs a high-throughput IFFT computation, showcasing efficient hardware resource utilization for high-speed mmWave applications.

Corresponding Author:

Kidsanapong Puntsri
Department of Electronics and Telecommunication Engineering, Rajamangala University of Technology Isan, Khon Kaen Campus (RMUTI KKC)
150 Sri Chan Rd, Mueang KhonKaen District, Khon Kaen 40000, Thailand
Email: kidsanapong.pu@rmuti.ac.th

1. INTRODUCTION

Millimeter wave (mmWave) frequency bands are a promising candidate for future wireless communication systems due to their vast bandwidth, and capable of supporting ultra-high data rates [1]. However, mmWave propagation characteristics present challenges such as high path loss and multipath fading [2]. Orthogonal frequency-division multiplexing (OFDM) modulation format is a key technology for mitigating these challenges. By dividing the wideband channel into narrow subcarriers, OFDM offers

robustness against frequency-selective fading in mmWave channels. Furthermore, OFDM remains a research focus for 5G and beyond due to its numerous advantages [3], [4], and it is already a standard modulation format for digital video broadcasting (DVB) [5] and wireless local area networks (WLAN) [6]. The main advantages of OFDM are its simple implementation and its ability to significantly reduce intersymbol interference (ISI), especially when the cyclic prefix (CP) is longer than the channel impulse response (CIR). However, realizing high-speed OFDM necessitates efficient hardware architectures for real-time signal processing. Field-programmable gate arrays (FPGAs) provide a flexible platform for such implementations due to their reconfigurability and potential for high throughput. The inverse fast Fourier transform (IFFT) operation, which converts the digital baseband signal frequency domain to the time domain for transmission, is a key element in the OFDM transmitter chain. Efficient IFFT algorithms on FPGAs are crucial for achieving real-time processing with constrained resources [7]–[9]. Thus, the real-time implementation of OFDM for high-speed mmWave communication systems poses a significant challenge. In addition to its role in high-speed wireless communication, mmWave OFDM has gained traction in big data applications involving real-time analytics [10]. The increasing volume of data generated by modern applications demands high-throughput processing capabilities, especially in areas such as real-time analytics within big data environments. The proposed FPGA-based OFDM transmitter, with its inherent parallelism and high-speed architecture, is well-suited to address this need by enabling efficient processing of massive data streams. Another important application of OFDM lies in ensuring secure and high-speed communication in cloud-based environments, particularly for internet of things (IoT) deployments that handle sensitive data [11]. The integration of FPGA-based OFDM transmitters into cloud security frameworks offers a promising solution to address latency and security challenges by enabling robust and high-speed encrypted communications [12], [13].

Various studies have explored FPGA-based IFFT implementations for OFDM transmitters, as shown in recent work [14]. Common approaches include pipelined and distributed architecture. Pipelined architectures achieve high throughput by processing data in stages but may require additional memory and logic resources [15], [16]. Conversely, distributed architectures utilize multiple processing units to parallelize the IFFT operation, potentially offering lower latency but at the cost of increased resource utilization [16]. The choice between these approaches depends on specific application requirements.

Previous researchers have sought to optimize IFFT implementations. In study [17], a low-latency Radix-2 based single-path delay feedback (SDF) IFFT architecture for OFDM systems is proposed, focusing on reducing memory size in the reordering method for the first stage of SDF IFFT architectures. This approach achieves a 41% memory reduction compared with conventional architectures. In [18], a method to enhance IFFT efficiency, which was done by replacing twiddle multipliers with simpler "pass-logic" for common OFDM input values is introduced, although it is limited to phase-shift keying (PSK) modulation. In study [19], an FPGA-based software-defined radio (SDR) system for OFDM is described, emphasizing power reduction techniques for mobile applications. In [20], an FPGA implementation of an OFDM transceiver for Wi-Fi is presented, but it relies on MATLAB/Simulink and is limited to lower speeds. In response to hardware resource constraints within interleaved frequency division multiple access (IFDMA) transceivers, a multi-priority scheduling (MPS) algorithm, detailed in [21], was developed to optimize the execution of butterfly computations. The resulting FFT implement, designated MPS-FFT, demonstrates a significant reduction in computational latency compared to conventional FFT methods when applied to IFDMA signal processing.

In this paper, an FPGA-based implementation of an OFDM transmitter signal processing chain optimized for high-speed mmWave communication systems is presented. The design prioritizes real-time processing efficiency and flexibility. A high-throughput 2048-point IFFT module, realized using a Radix-2 algorithm, forms the core of the design, showcasing efficient hardware resource utilization. The processing pipeline employs 8 parallel lanes for concurrent data computation. The implementation further includes CP insertion and configurable support for various quadrature amplitude modulation (QAM) orders and pilot arrangements. The design is implemented in VHDL using Vivado 2020 and evaluated on the Zynq UltraScale+ RFSOC ZCU111 evaluation kit. Experimental results demonstrate a mean squared error (MSE) of only 0.00013 between the FPGA-generated waveform and its MATLAB simulation. From the previous works [17]–[21], we introduce a novel FPGA-based mmWave OFDM transmitter architecture that pushes the boundaries of speed and throughput. A key innovation is the utilization of a significantly larger FFT size compared to existing FPGA implementations. This advancement unlocks the potential for high spectrum efficiency and larger bandwidth in mmWave communication systems.

2. OFDM for mmWave communication systems

OFDM for mmWave works like normal or conventional OFDM. The difference is that the bandwidth, where the mmWave operates from 400 MHz up to 800 MHz [1]-[3], at the carrier of 24.25-29.5 GHz. This

frequency is widely used for 5G. The transmitter uses IFFT for upconverting the QAM symbols to a specific frequency, called frequency spacing. The IFFT output, denoted by $x(n)$, can be written as (1) [22]:

$$x(n) = \frac{1}{N}\sum_{k=0}^{N-1} X(k)\, e^{j2\pi kn/N}, \quad (1)$$

where $k = 0,1,\ldots,N-1$, and N is the number of FFT points (number of samples or periods). $X(k)$ is arbitrary binary phase shift keying (BPSK), quadrature phase shift keying (QPSK) or quadrature amplitude modulation (QAM) mapping. The higher bit mapping is the higher order and the higher spectrum efficiency. The CP is next appended, denoted by $x_{cp}(n)$, the index of OFDM symbol, including CP sample can be expressed by (2).

$$x_{cp}(n) = \left[\underbrace{x(-N_{cp})x(-N_{cp}+1)x(-N_{cp}+2)\ldots x(-N_{cp}+N_{cp}-1)}_{CP}\underbrace{x(0)x(1)\ldots x(N\ \ -1)}_{Useful\ data}\right], \quad (2)$$

At the receiver end, the received signal with quantization noise due to the fixed-point format and without RF implements is computed by (3).

$$y(n) = x_{CP}(n) \otimes h(n) + z(n) + q(n), \quad (3)$$

hence, $y(n)$ represents the received distorted replica of the transmitted signal. The parameters $h(n)$, $z(n)$and $q(n)$ are channel impulse response, AWGN component, and quantization noise component due to fixed-point number, respectively. From (3), the signal to noise ratio (SNR) is defined by (4).

$$SNR = \frac{E\left(x_{cp}(n)\right)}{E(z(n))+E(q(n))}, \quad (4)$$

where $E(\cdot)$ is the expectation operator. The SNR in dB can be calculated by $SNR_{dB} = 10\,log(SNR)$.

The received frequency domain, denoted by $X(k)$, is done by taking FFT of (3), the calculation is expressed by (5).

$$\begin{aligned} Y(k) &= \sum_{n=0}^{N-1} y(n)\, e^{-j2\pi kn/N} \\ &= X(k)\cdot H(k) + Z(K) + Q(k), \end{aligned} \quad (5)$$

where $H(k)$ is communication channels, while $Z(K)$ and $Q(K)$ are AWGN and quantization noise components in frequency domain, respectively. Finally, the recovery of the received signal can be simply divided by $H(k)$, expressed by (6).

$$\begin{aligned} \tilde{Y}(k) &= \frac{Y(k)}{H(k)} \\ &= X(k) + (Z(k) + Q(k))/H(k), \end{aligned} \quad (6)$$

As can be seen, the receiver processing is done in frequency domain, which is simple to implement in hardware.

3. OFDM transmitter implementation

3.1. Hardware constraints

Hardware resource utilization of the OFDM transmitter (Tx) processing unit and design choices are detailed in this section. The Tx comprises three primary stages: QAM mapping, IFFT, and CP appending. All processing units employ an 18-bit fixed-point representation for optimal efficiency, with the first bit denoting sign, followed by 4 bits for the integer portion and 13 bits for the fractional part. This fixed-point format [23] balances computational accuracy with resource utilization constraints. Additionally, *fi (v, s, w, f)* is used in MATLAB command to convert from the floating-point value to fixed-point format, where *v* is value, *s* is signed property, *w* is word length, and *f* is fraction length. A 2048-point Radix-2 IFFT is implemented, necessitating 11 computation stages. Additionally, the CP is 256 samples. To enhance throughput, the design incorporates parallel processing, allowing 8 simultaneous input computations. This parallelism, coupled with

a 100 MHz internal clock, facilitates an 800 MHz data throughput. Under the chosen configuration, the design achieves an 800 MHz data throughput. If 1024-point QAM modulation is adopted, a net data rate of 8 Gbps can be attained. This illustrates the inherent trade-off between data throughput and spectral efficiency. Higher-order QAM schemes offer increased data rates but demand more complex processing, potentially leading to elevated resource usage. Additionally, all the processing is implemented on the Zynq UltraScale+ RFSoC ZCU111 Evaluation Kit [24].

3.2. Quadrature amplitude modulation mapping

Quadrature amplitude modulation (QAM) mapping is a critical process in digital communication systems, where binary bit sequences are assigned to specific symbols within a constellation diagram. In this work, a look-up table (LUT) based approach is proposed for efficient QAM mapping. The incoming data bits serve as addresses for the LUT, retrieving pre-calculated values representing the corresponding I (in-phase) and Q (quadrature) components of the QAM symbol. The symbol power is normalized to unity to simplify the mapping process. For a comprehensive explanation of QAM mapping, refer to [2], [3]. The number of points within the constellation diagram directly corresponds to the number of unique addresses utilized in the mapping process. For instance, QPSK with its four constellation points employs four unique addresses, while 64-QAM, with 64 points, utilizes 64 addresses. LUTs offer an efficient hardware implementation for QAM mapping, allowing seamless switching between different QAM schemes (*e.g.*, QPSK, 16-QAM) by simply modifying the pre-loaded values within the LUT. To achieve high-speed communication, the proposed approach incorporates eight parallel processing units, enabling simultaneous calculation of multiple QAM symbols. Further details regarding this parallel processing implementation can be found in Figure 1. As can be seen, the proposed LUT-based QAM mapping approach offers a flexible and efficient solution for digital communication systems, facilitating high-speed data transmission while maintaining the flexibility to adapt to different QAM schemes.

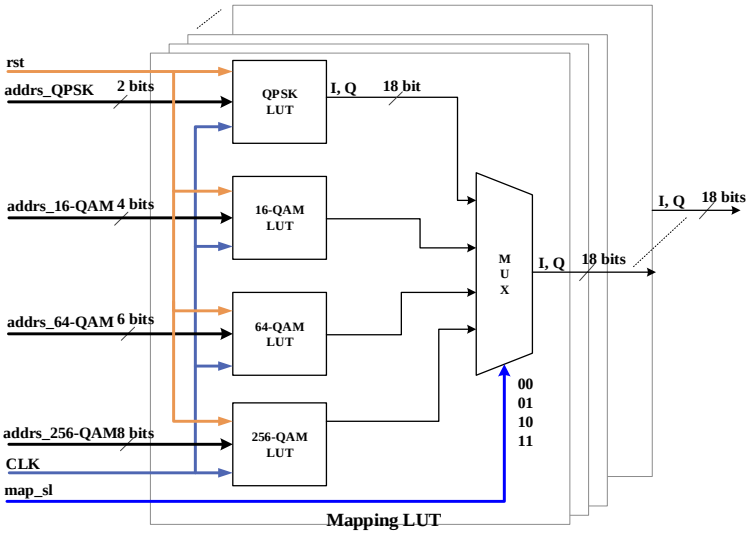

Figure 1. The proposed QAM mapping with 8 parallel processing

3.3. CP appending

In this section, the proposed CP added method in FPGA is presented. The output from the IFFT processing is fed to the dual core RAM to store all the 2048 samples, where the input address of 256 (=2048/8) is counted from 0 to 255. For the address for reading out, there are two counts needed. The first count is for the last samples of 32 (=256/8) from the useful samples itself, this needs 0, …, 31 address. The second count is for useful samples. Therefore, the counter length is equal to input address. Additionally, since the read-out address is longer than the written address, the CLK_B should be faster than CLK_A. The ratio is expressed by $\frac{(N_{sf}+N_{cp})}{N_{sf}}$, where N_{sf} is a number of useful samples or the FFT size and N_{cp} is the number of CP in samples. In this work, $N_{sf} = 2048$ and $N_{cp} = 288$, the ratio is $\frac{(2048+256)}{2048} = 1.125$. For example, if the speed of CLK_A = 100 MHz is assumed, the CLK_B will be 112.5 MHz.

3.4. 2048-points IFFT/FFT implementation

This section investigates the application of the IFFT in mmWave high-speed wireless data communication systems. The IFFT is a critical component of OFDM, facilitating the transformation of digital data, expressed as complex symbols in the frequency domain, into a corresponding time-domain signal. A comparative evaluation of general IFFT/FFT implementation methodologies, encompassing speed, hardware complexity, and performance advantages, is presented in Table 1. Specifically, the proposed hybrid implementation, combining Radix and RAM architectures, demonstrates a notable improvement in speed and efficiency relative to alternative methodologies. Additionally, Radix-2 is a very effective for implementation on hardware; especially on FPGA, and that is widely used in many digital signal processing (DSP) applications. The divide and conquer strategy are used. The method divides the 2048-point DFT input progressively into smaller DFTs. In this case, $N = 2048$, the number of stages is $v = log_2(N) = 11$. In this work, computing the inverse FFT using forward FFT is employed, where the FFT can be used for the OFDM receiver, in the future. The FFT calculation is given by [25],

$$\begin{aligned} X(k) &= FFT(x) \\ &= \sum_{n=0}^{N-1} x(n) W_N^{kn}, \end{aligned} \tag{7}$$

where $X(k)$ is frequency domain and $x(n)$ is discrete time domain input. $W_N^{kn} = e^{-j2\pi kn/N}$ is complex number twiddle factors (TW). $n = 0,1,\ldots,N-1$ and $k = 0,1,2,\ldots,N/2-1$. In Radix-2, the sampling input signal is computed separately for the even-indexed, $x_0, x_2, \ldots, x_{N-2}$ and the odd-indexed, $x_1, x_3, \ldots, x_{N-1}$. Additionally, the algorithm divides the calculation into two parts, expressed by (8).

$$\begin{aligned} X(k) &= \sum_{\substack{neven \\ N/2-1}} x(n)\, W_N^{nk} + \sum_{\substack{nodd \\ N/2-1}} x(n)\, W_N^{nk} \\ &= \sum_{n=0} x(2n)\, W_N^{2nk} + \sum_{n=0} x(2n+1) W_N^{(2n+1)k} \\ &= \sum_{n=0}^{N/2-1} [x(2n) + x(2n+1)]\, W_N^{2nk}, \end{aligned} \tag{8}$$

hence, from the complex exponential notation, $W_N^{2nk} = W_{N/2}^{nk}$. Then, the first half can be calculated by (9).

$$X(k) = \sum_{n=0}^{N/2-1} x(2n)\, W_{N/2}^{nk} + W_N^k \sum_{n=0}^{N/2-1} x(2n+1) W_{N/2}^{nk}\,, \tag{9}$$

and the second half is calculated by (10).

$$X(k+N/2) = \sum_{n=0}^{N/2-1} x(2n)\, W_{N/2}^{n(k+N/2)} + W_N^{k(N/2)} \sum_{n=0}^{N/2-1} x(2n+1) W_{N/2}^{n(k+N/2)}\,. \tag{10}$$

We know that $W_N^{k+N/2} = -W_N^k$. Therefore

$$X(k+N/2) = \sum_{n=0}^{N/2-1} x(2n)\, W_{N/2}^{nk} - W_N^k \sum_{n=0}^{N/2-1} x(2n+1) W_{N/2}^{nk}\,. \tag{11}$$

The butterfly structure is usually employed in hardware to implement (9) and (11). Additionally, by using (7), IFFT can be calculated by [21].

$$x(n) = \frac{1}{N} conj\left(FFT\left(conj(X(k))\right)\right), \tag{12}$$

where *conj* is configuration operator. From (12), this implies that IFFT can be computed by using FFT. The implement of (12) is detailed in Figure 2, where each stage uses the butterfly structure, as shown insert A in Figure 2. It is recursively 256 times to achieve 2048-points. The total stage is 11. Additionally, the term $\frac{1}{N}$ will be spread by $\frac{1}{2}$ to each stage. This approach allows for the conservation of integer bits while increasing the number of fractional bits. Consequently, the bit precision is enhanced, leading to improved accuracy. Additionally, the computation speed can be potentially accelerated up to 8 times, assuming ideal conditions. With an internal clock frequency of 100 MHz, the achievable total throughput reaches 800 MHz. These specifics are visually represented in Figure 3. The Twiddle factor (TW) values are pre-computed and stored in internal read-only memory (ROM) for efficient access during processing.

Table 1. Common methods for IFFT/FFT implementation

Method	Speed (Approximate)	Gate Consumption (Approximate)	Advantages	Disadvantages
CORDIC-based [25]	Low	Moderate (thousands)	Low power consumption, efficient for trigonometric calculations	Can be slower than other methods, accuracy trade-offs
Pipeline [26]	Moderate	Moderate to high	High throughput, suitable for real-time processing	Increased latency, can be resource-intensive for large FFT sizes
Radix+ RAM (Proposed)	High	Moderate to high	Flexible design, high throughput and efficient resource usage	Can be complex to design, may require additional control logic

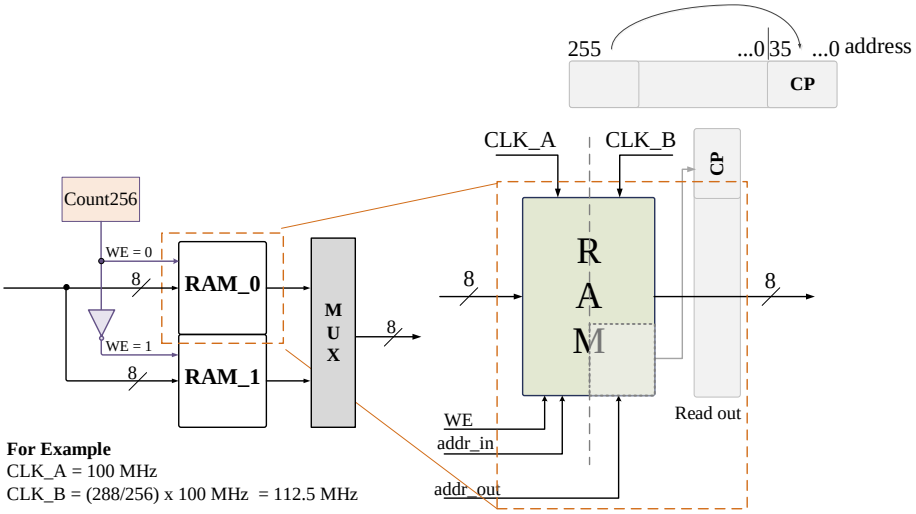

Figure 2. The proposed CP appending using dual core RAM at the transmitter

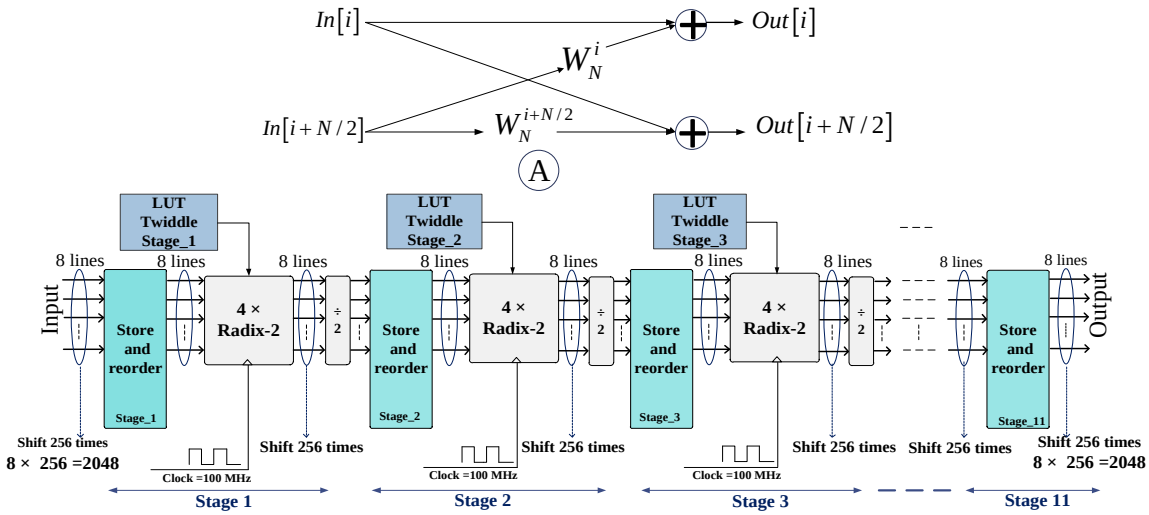

Figure 3. FFT implementation for 8 parallels processing with 11 stages of calculation

During each processing stage, the system needs to reorder data based on the output index for the next stage. To achieve this, the output data on each stage is stored in a random-access memory (RAM). The read address of the RAM is dynamically controlled based on the required index for the next stage. For feed-forward processing without data delays, dual-Banks RAM architecture is employed. The Banks consist of Bank 0 and Bank 1. A write enable (WE) signal controls which bank is active. When data is written to the first Bank (*e.g.*, Bank 0 with WE = 1), data from each stage is simultaneously read from the other bank

(Bank 1). Conversely, when WE = 0 for Bank 0, data is written to Bank 1 and read from Bank 0. The WE signal typically toggles every 256 clock cycles. Please refer to Figure 4 for further understanding.

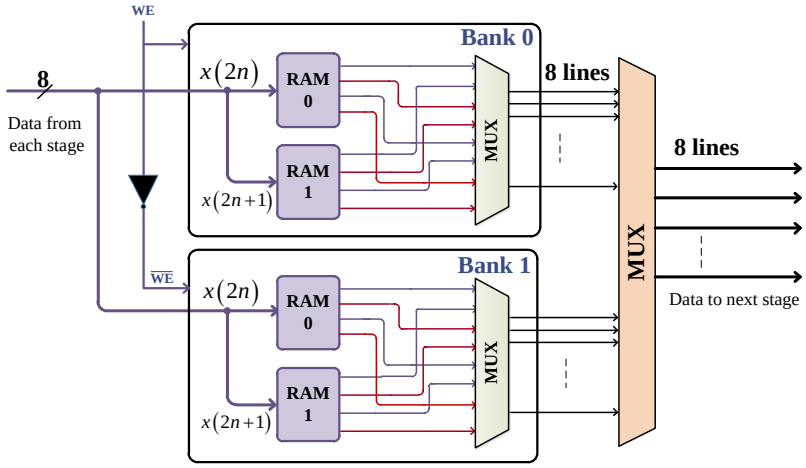

Figure 4. Details of the algorithm for storing and reordering data indices in preparation for the next stage

4. IMPLEMENTATION RESULTS

In this section, the FPGA-based transmitter signal processing chain for mmWave OFDM systems is reported. First, the effect of AWGN and quantization noise on OFDM transmitter using fixed-point format is simulated and analyzed. The constraints of the designed are shown in section 3.1, specifically, the 2048 FFT size and the 18-bit word length are used, and the bit error rate (BER) performance was evaluated via numerical simulation. Only AWGN and quantization noise are considered. The subcarrier indices from 21 to 632 and 1424 to 2030 are modulated with 64-QAM, and the rest are zeros. Figure 5 shows the distortion impact due to the fixed-point, which is evident that the quantization noise is not the sole contributor. As can be seen, the inter-band interference also plays a role, as evidenced by the power leakage spreading across subcarriers. Additionally, the results demonstrate a clear increase in BER as the number of fractional bits decreased, as illustrated in Figure 6. Furthermore, it indicates that higher-order QAM modulation schemes exhibit more sensitivity to the SNR. The findings from Figures 5 and 6 collectively suggest an optimal fractional bit allocation exceeding 10 bits. Consequently, for the subsequent investigations within this study, we have judiciously selected a 13-bit fractional representation. This choice aims to achieve an optimal trade-off between computational accuracy, BER performance, and the suppression of inter-band interference.

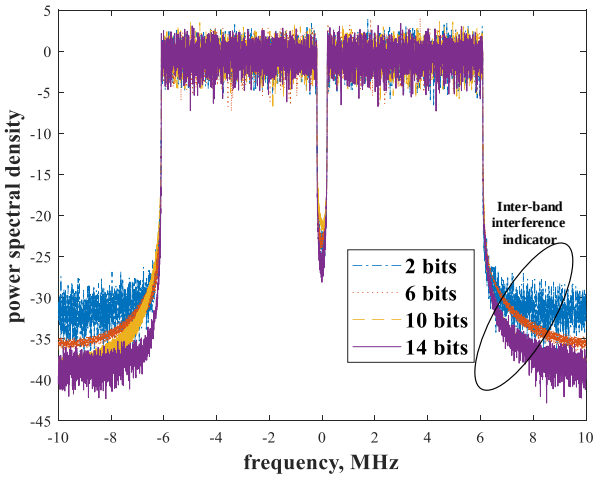

Figure 5. The Inter-band interference effect on OFDM transmitter

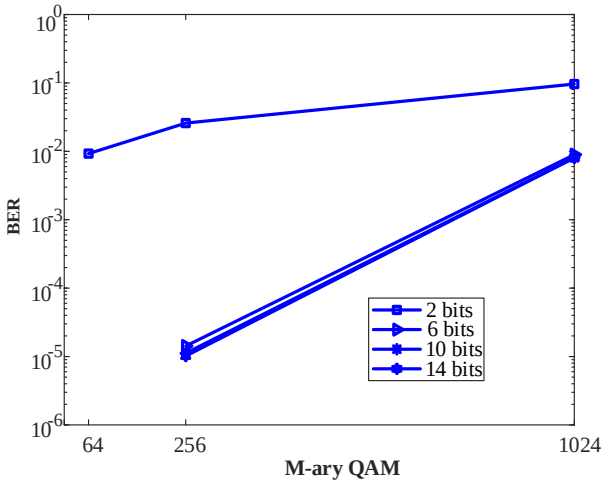

Figure 6. BER versus M-ary QAM with various fractional bits is considered

Next, a comprehensive evaluation of the high-speed OFDM waveform generator implemented on an FPGA for mmWave communication systems was conducted. The experimental setup is illustrated in inserted in Figure 7(a), and the generated waveform is presented in Figure 7(b). The design and implementation were carried out using VHDL within the Vivado 2020 environment, targeting the Zynq UltraScale+ RFSoC ZCU111 evaluation kit.

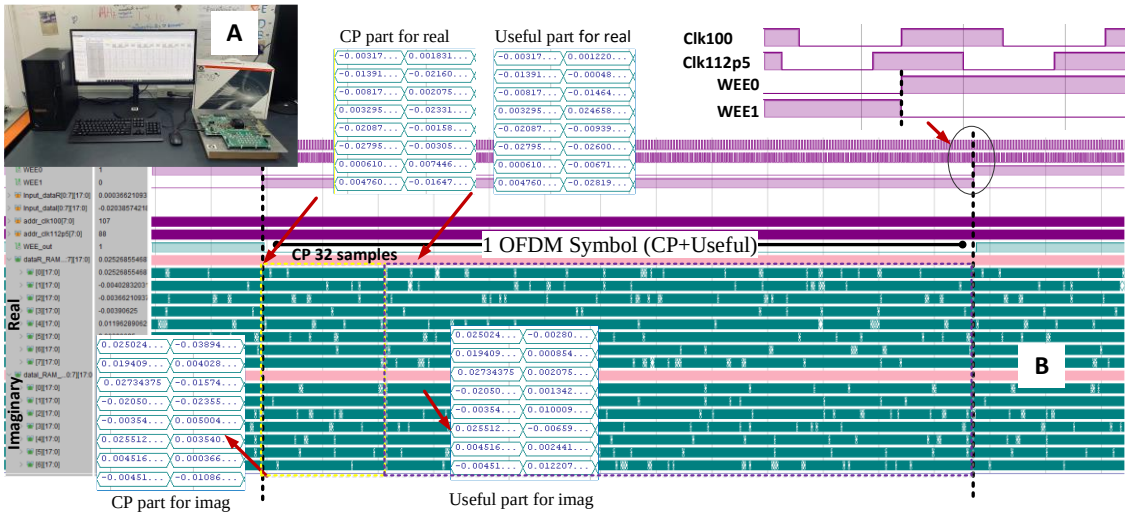

Figure 7. OFDM transmitter waveform generator implemented on an FPGA for mmWave communication systems: (a) system setup and (b) OFDM output waveform

The processing architecture is structured as 8 parallel lanes, facilitating the concurrent computation of multiple data streams. As depicted in Figure 2, the CP is appended over 32 clock cycles (8 lanes × 32 cycles = 256 samples). Subsequently, 256 clock cycles are allocated for 2048 of the useful OFDM data symbols (8 lanes × 256 cycles = 2048 samples). Thus, each complete OFDM symbol comprises 2304 samples (=2048 + 256). The design incorporates two distinct clock domains: a 100 MHz system clock (clk100) and a 112.5 MHz clock multiplexer (clk112p5) responsible for outputting the CP and OFDM symbols. This allows for higher bandwidth and increases spectral efficiency. Moreover, to ensure adequate temporal separation between write and read operations on memory elements, clk100 is intentionally phase-shifted by 250 degrees. A quantitative comparison between the FPGA-generated waveform and its MATLAB-simulated reveals a mean squared error (MSE) of only 0.00013.

Finally, Table 2 provides a breakdown of the post-implementation resource utilization of the FPGA. Notably, key resources such as the CLB LUTs, RAM, and DSPs exhibit utilization percentages of only 7.91%, 0.74%, and 3.7%, respectively. Additionally, there is no number of failing endpoint. Specifically, our implementation demonstrates superior performance in terms of both speed and throughput compared to the state-of-the-art. To the best of our knowledge, our design achieves the largest FFT size among FPGA-based mmWave communication systems reported in the literature. There are only simulation works, as shown [27], [28], and no implementation has been reported yet. The results demonstrated that the efficacy and efficiency of the proposed FPGA-based high-speed OFDM transmitter design for mmWave communication systems are achieved. To evaluate the power efficiency of the proposed FPGA-based OFDM transmitter, we conducted a power consumption analysis using Xilinx Power Estimator tool in Vivado 2020. The analysis considers the dynamic power consumption of the FPGA resources utilized by the design, including the CLBs, block RAMs, and DSPs. The estimated power consumption operating at 100 MHz is 48.667 Watts. Furthermore, the real-world applicability of our FPGA-based OFDM transmitter was assessed by considering its robustness to channel impairments and compatibility with existing mmWave infrastructure.

Table 2. The Zynq UltraScale+ RFSoC ZCU111 evaluation kit resource usage

Resources	Usage/Values
CLB LUTs	7.91%
CLB Registers	6.98%
LUT as Memory	9.65%
LUT as Logic	3.06%
Block RAM Tile	0.74%
DSPs	3.70%
Number of failing endpoints	0

5. CONCLUSION

This paper presented a resource-efficient FPGA implementation of an OFDM transmitter signal processing chain optimized for high-speed mmWave communication systems. The core of the design is a 2048-point IFFT module, realized using a Radix-2 algorithm to minimize hardware footprint. Additionally, the implementation incorporates flexible CP insertion, adaptable QAM modulation, and configurable pilot patterns, allowing for dynamic trade-offs between spectral efficiency, system robustness, and hardware resource utilization. Experimental results demonstrate a high degree of correlation between the FPGA-generated waveform and MATLAB simulations, validating the design's accuracy. Furthermore, efficient resource utilization underscores the practicality of the proposed solution for real-time mmWave OFDM transmitter applications. The high-throughput processing capabilities pave the way for future mmWave communication systems capable of multi-gigabit per second data rates is achieved. The inherent low-latency and high-speed processing offered by this FPGA-based approach are crucial for supporting Big Data applications and enabling advanced AI functionalities. Particularly, this work holds significant relevance for 6G technology, where real-time, high-bandwidth data processing is paramount.

ACKNOWLEDGMENTS

This research project is supported by Rajamangala University of Technology Isan.

FUNDING INFORMATION

This research project is supported by Rajamangala University of Technology Isan. Contract No. ENG9/67.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Kidsanapong Puntsri	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Bussakorn Bunsri			✓			✓				✓	✓	✓		
Puripong Suthisopapan						✓				✓				

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

The authors state no conflict of interest.

INFORMED CONSENT

This study does not involve human participants and therefore informed consent was not required.

ETHICAL APPROVAL

This study did not involve human participants or animals, and therefore, ethical approval was not required.

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

[1] D. Wu, J. Wang, Y. Cai, and M. Guizani, "Millimeter-wave multimedia communications: Challenges, methodology, and applications," *IEEE Communications Magazine*, vol. 53, no. 1, pp. 232–238, 2015, doi: 10.1109/MCOM.2015.7010539.
[2] K.-C. Huang and Z. Wang, *Millimeter wave communication systems*. Wiley-IEEE Press.
[3] T. S. Rappaport *et al.*, "Millimeter wave mobile communications for 5G cellular: It will work!," *IEEE Access*, vol. 1, pp. 335–349, 2013, doi: 10.1109/ACCESS.2013.2260813.
[4] Y. Kim, H. Y. Lee, J. Oh, J. Lee, W. Roh, and K. Cheun, "Feasibility of mobile cellular communications at millimeter wave frequency," in *Proceedings - IEEE Global Communications Conference, GLOBECOM*, 2015, pp. 589–599, doi: 10.1109/GLOCOM.2014.7417433.
[5] I. Eizmendi *et al.*, "DVB-T2: The second generation of terrestrial digital video broadcasting system," *IEEE Transactions on Broadcasting*, vol. 60, no. 2, pp. 258–271, 2014, doi: 10.1109/TBC.2014.2312811.
[6] I. 802.11, "IEEE Standard for Information technology-Telecommunications and information exchange between systems Local and metropolitan area networks- Specific requirements Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications," *IEEE Std 802.11-2012 (Revision of IEEE Std 802.11-2007)*, vol. 11, pp. 1–2793, 2012, doi: 10.1109/IEEESTD.2012.6178212.
[7] B. Bunsri, K. Puntsri, and A. Yindeemak, "FPGA implementation of IFFT-2048 points for high speed transmitter OFDM communication systems with 64-QAM mapping," 2024, doi: 10.1109/iEECON60677.2024.10537973.
[8] K. M. Mohan, K. Naresh, C. H. Ganesh, V. S. Reddy, and K. H. Kishore, "Implementation of N-Point FFT/IFFT processor based on Radix-2 Using FPGA," in *Proceedings - 2022 International Conference on Recent Trends in Microelectronics, Automation, Computing and Communications Systems, ICMACC 2022*, 2022, pp. 499–504, doi: 10.1109/ICMACC54824.2022.10093290.
[9] S. Dhanasekar, P. M. Bruntha, T. M. Neebha, N. Arunkumar, N. Senathipathi, and C. Priya, "An area effective OFDM transceiver system with multi-radix FFT/IFFT algorithm for wireless applications," in *2021 7th International Conference on Advanced Computing and Communication Systems, ICACCS 2021*, 2021, pp. 551–556, doi: 10.1109/ICACCS51430.2021.9441694.
[10] M. S. Husain, M. Z. Khan, and T. Siddiqui, *Big data concepts, technologies, and applications*. CRC, Taylor & Francis group, 2023.
[11] M. Z. Khan, M. Shoaib, M. S. Husain, K. Ul Nisa, and M. T. Quasim, "Enhanced mechanism to prioritize the cloud data privacy factors using AHP and TOPSIS: a hybrid approach," *Journal of Cloud Computing*, vol. 13, no. 1, 2024, doi: 10.1186/s13677-024-00606-y.
[12] M. Z. Khan, K. U. Nisa, M. T. Quasim, M. A. Khalifa, and M. M. Mobarak, "Cloud-based data protection: A framework for authorizing data movement," in *Proceedings - 2024 International Conference on Expert Clouds and Applications, ICOECA 2024*, 2024, pp. 271–275, doi: 10.1109/ICOECA62351.2024.00057.
[13] A. S. Alluhaidan, M. Z. Khan, N. Ben Halima, and S. Tyagi, "A diversified context-based privacy-preserving scheme (DCP2S) for internet of vehicles," *Alexandria Engineering Journal*, vol. 77, pp. 227–237, 2023, doi: 10.1016/j.aej.2023.06.073.
[14] M. Garrido, "A survey on pipelined FFT hardware architectures," *Journal of Signal Processing Systems*, vol. 94, no. 11, pp. 1345–1364, 2022, doi: 10.1007/s11265-021-01655-1.
[15] H. N. Nguyen, S. A. Khan, C. H. Kim, and J. M. Kim, "A pipelined FFT processor using an optimal hybrid rotation scheme for complex multiplication: Design, FPGA implementation and analysis," *Electronics (Switzerland)*, vol. 7, no. 8, 2018, doi: 10.3390/electronics7080137.
[16] K. K. Parhi, "A low-latency FFT-IFFT cascade architecture," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2024, pp. 181–185, doi: 10.1109/ICASSP48485.2024.10447370.
[17] I. G. Jang and G. Do Jo, "Low latency IFFT design for OFDM systems supporting full-duplex FDD," *International Conference on Ubiquitous and Future Networks, ICUFN*, pp. 642–646, 2017, doi: 10.1109/ICUFN.2017.7993870.

[18] V. Arunachalam and A. N. Joseph Raj, "Efficient VLSI implementation of FFT for orthogonal frequency division multiplexing application," *IET Circuits, Devices and Systems*, vol. 8, no. 6, pp. 526–531, 2014, doi: 10.1049/iet-cds.2013.0457.
[19] T. R. D. Kumar, V. Madhavan, M. A. Krishna, and R. C. Abishek Ronjan, "Area efficient implementation of software-defined radio using FPGA," in *Proceedings of the 8th International Conference on Communication and Electronics Systems, ICCES 2023*, 2023, pp. 222–225, doi: 10.1109/ICCES57224.2023.10192804.
[20] A. Mecwan and D. Shah, "Implementation of OFDM transceiver on FPGA," *2013 Nirma University International Conference on Engineering, NUiCONE 2013*, 2013, doi: 10.1109/NUiCONE.2013.6780121.
[21] Y. Du, S. C. Liew, and Y. Shao, "Efficient FFT computation in IFDMA transceivers," *IEEE Transactions on Wireless Communications*, vol. 22, no. 10, pp. 6594–6607, 2023, doi: 10.1109/TWC.2023.3244553.
[22] A. F. Demir, M. H. Elkourdi, M. Ibrahim, and H. Arslan, "Waveform design for 5G and beyond," *5G Networks: Fundamental Requirements, Enabling Technologies, and Operations Management*, pp. 51–76, 2018, doi: 10.1002/9781119333142.ch2.
[23] R. G. Lyons, *Understanding digital signal processing*, 3rd Ed. Pearson Education, Inc., 2011.
[24] AMD Adaptive Computing, "ZCU111 evaluation board user guide, UG1271 (v1.4)," *AMD Adaptive Computing*, 2023. https://docs.amd.com/r/en-US/ug1271-zcu111-eval-bd (accessed Sep. 06, 2024).
[25] J. S. Chitode, *Digital signal processing*. Technical Publications Pune, 2008.
[26] H. Mahdavi and S. Timarchi, "Area-time-power efficient FFT architectures based on binary-signed-digit CORDIC," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 10, pp. 3874–3881, 2019, doi: 10.1109/TCSI.2019.2922988.
[27] P. M. Kandan and K. Narmadha, "High performance 2048 point FFT/IFFT design for IEEE 802.16e standard," *2014 International Conference on Electronics and Communication Systems, ICECS 2014*. pp. 1–5, 2014, doi: 10.1109/ECS.2014.6892531.
[28] N. Mahdavi, R. Teymourzadeh, and M. Bin Othman, "VLSI implementation of high speed and high resolution FFT algorithm based on Radix 2 for DSP application," in *2007 5th Student Conference on Research and Development, SCORED*, 2007, pp. 1–4, doi: 10.1109/SCORED.2007.4451381.

BIOGRAPHIES OF AUTHORS

Kidsanapong Puntsri received B.Eng. degree in telecommunication engineering from Mahanakorn University of Technology (MUT), Thailand, in 2002, and M.Eng. degree in telecommunication engineering from King Mongkut's Institute of Technology Ladkrabang (KMITL), Thailand, in 2004. In 2014, he obtained Dr.-Ing in electrical engineering from the University of Paderborn, Germany. At the present, he is an associate professor at the Department of electronics and telecommunication engineering, Rajamangala University of Technology Isan, Khon Kaen Campus, Thailand. His main research interests include multi-carrier communication in both optical and wireless systems, and realization of communication systems by field-programmable gate array (FPGA). He has published in IEEE more than 40 contribution papers. He is a senior IEEE member. He can be contacted at email: kidsanapong.pu@rmuti.ac.th.

Bussakorn Bunsri received B.Eng., and M.Eng. degree in electronics and telecommunication engineering from Rajamangala University of Technology Isan, Khon Kaen Campus (RMUTI KKC), Thailand, in 2019 and 2024 respectively. Currently, she makes Dr.Eng. degree in electrical engineering, At Rajamangala University of Technology Isan, Khon Kaen Campus (RMUTI KKC), Thailand. Her research interests include multi-carrier communication and field-programmable gate array (FPGA) implementation. She can be contacted at email: bussakorn.bu@rmuti.ac.th.

Puripong Suthisopapan received the B.Eng., M.Eng. and Ph.D. degrees in electrical engineering from Khon Kaen University, Thailand in 2007, 2009 and 2012, respectively. Since January 2020, he has been an associate professor in the Department of Electrical Engineering, Faculty of Engineering, Khon Kaen University, Thailand. His current research interests are error correction codes, signal processing techniques for modern digital communications and quantum error correcting codes. He can be contacted at email: purisu@kku.ac.th.