DOI: 10.21917/ijme.2024.0299

FPGA IMPLEMENTATION OF REAL-TIME SIGNAL PROCESSING ALGORITHMS FOR WIRELESS COMMUNICATION SYSTEMS

Rakesh Kumar[1], Sandeep Kumar Bhakhar[2], V. Venkata Ramana[3] and K. Venkata Ramana[4]

[1,2]Department of Electronics and Communication Engineering, Bikaner Technical University, India
[3]Department of Computer Science and Engineering, K.S.R.M College of Engineering, India
[4]Department of Computer Science and Engineering, QIS College of Engineering and Technology, India

Abstract

As wireless communication systems evolve towards higher data rates and increased complexity, the demand for efficient signal processing algorithms also rises. Memoryless predistortion techniques play a crucial role in mitigating nonlinear distortions in RF power amplifiers, ensuring high-quality transmission. This study presents a novel multichannel FPGA implementation of a real-time signal processing algorithm for memoryless predistortion in wireless communication systems. The proposed methodology leverages parallel processing and hardware acceleration to achieve high throughput and low latency. The key contribution lies in the efficient utilization of FPGA resources to implement the predistortion algorithm across multiple channels, enabling real-time operation with minimal hardware overhead. Experimental results demonstrate the effectiveness of the FPGA-based implementation in compensating for nonlinear distortions in RF power amplifiers. The system achieves significant improvements in signal quality and spectral efficiency, paving the way for enhanced performance in wireless communication systems.

Keywords:

FPGA, Real-Time Signal Processing, Memoryless Predistortion, Wireless Communication System, Multichannel

1. INTRODUCTION

With the ever-increasing demand for high-speed wireless communication, ensuring reliable and efficient transmission is paramount [1]. In wireless communication systems, RF power amplifiers (PAs) play a crucial role in boosting signal power to achieve desired coverage and range [2]. However, nonlinear distortions inherent in PAs can degrade signal quality, leading to performance degradation and spectral inefficiency. Memoryless predistortion techniques have emerged as effective solutions to mitigate these distortions, thereby improving system performance [4]. This study focuses on the implementation of a real-time signal processing algorithm for memoryless predistortion in wireless communication systems using FPGA technology [5].

Traditional software-based predistortion techniques suffer from latency and throughput limitations, making them unsuitable for real-time applications in high-speed wireless communication systems [6]. Moreover, hardware implementations using dedicated digital signal processors (DSPs) may face scalability and cost constraints. Addressing these challenges requires innovative approaches that leverage the parallel processing capabilities and reconfigurability of FPGA devices [7].

The primary challenge addressed in this study is to develop a real-time signal processing system capable of implementing memoryless predistortion algorithms for multiple channels in wireless communication systems [8]. This involves designing a hardware architecture that can efficiently process high-speed data streams while maintaining low latency and minimizing resource utilization [9]. Additionally, the system should be scalable to accommodate future advancements in wireless communication standards and technologies [10].

The main objectives of this study are twofold: first, to design and implement a multichannel FPGA-based signal processing system capable of real-time memoryless predistortion for wireless communication applications; and second, to evaluate the performance of the proposed system in terms of signal quality improvement, spectral efficiency, and resource utilization.

The novelty of this work lies in the development of a multichannel FPGA implementation for real-time memoryless predistortion, which addresses the limitations of existing solutions in terms of scalability, throughput, and cost-effectiveness. By leveraging the parallel processing capabilities of FPGAs, the proposed system offers a highly efficient and flexible solution for mitigating nonlinear distortions in RF power amplifiers.

The contributions of this study include the design and implementation of a novel FPGA-based signal processing architecture capable of real-time memoryless predistortion for wireless communication systems. Additionally, experimental evaluation demonstrates the effectiveness of the proposed system in improving signal quality and spectral efficiency while minimizing resource utilization. Overall, this research paves the way for enhanced performance and scalability in high-speed wireless communication systems.

2. RELATED WORKS

Several studies have addressed the challenge of implementing real-time signal processing algorithms, including memoryless predistortion, for wireless communication systems. These works have explored various approaches using both software and hardware implementations.

Early research focused on software-based predistortion algorithms implemented on general-purpose processors (GPPs) or DSPs. For example, [8] proposed a software-based predistortion technique for compensating for nonlinearities in PAs using a lookup table approach. While such methods are flexible and easy to implement, they often suffer from limited throughput and high latency, making them unsuitable for real-time applications in high-speed wireless systems.

To address the limitations of software-based solutions, researchers have investigated hardware implementations using dedicated DSPs or application-specific integrated circuits (ASICs). For instance, [9] developed a hardware predistortion system based on FPGA technology, achieving real-time operation with improved throughput compared to software-based approaches. However, such solutions may face scalability and cost constraints due to the fixed hardware architecture.

FPGA-Based Implementations: In recent years, there has been a growing interest in FPGA-based implementations for real-time signal processing in wireless communication systems. These implementations offer the advantages of reconfigurability, parallel processing, and low latency. For example, [10] presented a multichannel FPGA implementation of a predistortion algorithm for compensating for nonlinear distortions in RF PAs. Their work demonstrated the feasibility of using FPGAs for real-time signal processing in high-speed wireless systems.

Hybrid Approaches: Some studies have explored hybrid approaches that combine software and hardware components to achieve real-time operation with improved flexibility and efficiency. For instance, [11] proposed a hybrid predistortion system that utilizes both FPGA-based hardware acceleration and software-based control algorithms. This hybrid approach allows for dynamic reconfiguration and optimization of the predistortion system, enabling adaptive compensation for varying RF PA characteristics.

Several comparative studies [12] have been conducted to evaluate the performance of different predistortion techniques and implementations in terms of signal quality improvement, throughput, latency, and resource utilization. These studies provide valuable insights into the strengths and weaknesses of various approaches, helping researchers and engineers make informed decisions when designing predistortion systems for wireless communication applications.

The literature demonstrates the importance of efficient and scalable implementations of memoryless predistortion algorithms for enhancing the performance of wireless communication systems, with FPGA-based solutions emerging as promising candidates for real-time signal processing. However, further research is needed to address remaining challenges and optimize the design of FPGA-based predistortion systems for future wireless communication standards and applications.

3. PROPOSED METHOD

The proposed method aims to implement a real-time signal processing algorithm for memoryless predistortion in wireless communication systems using a multichannel FPGA architecture.

The method begins with the design of a hardware architecture tailored for FPGA implementation. This architecture is optimized to accommodate multiple channels for parallel processing, leveraging the inherent parallelism of FPGAs. Each channel corresponds to a specific RF signal path, enabling simultaneous processing of multiple signals. The core of the method involves the implementation of a memoryless predistortion algorithm within the FPGA architecture. This algorithm is designed to compensate for nonlinear distortions introduced by RF power amplifiers in the wireless communication system. Memoryless predistortion techniques typically involve the estimation and inversion of the nonlinear characteristics of the power amplifiers. To achieve real-time operation and high throughput, the method utilizes parallel processing techniques and hardware acceleration capabilities of FPGAs. By distributing the computational workload across multiple processing units within the FPGA, the method maximizes resource utilization and minimizes latency. The FPGA-based implementation allows for dynamic reconfiguration and adaptation of the predistortion algorithm to accommodate changes in the RF signal characteristics or system requirements. This flexibility enables adaptive compensation for varying nonlinearities in the RF power amplifiers, ensuring optimal performance under different operating conditions.

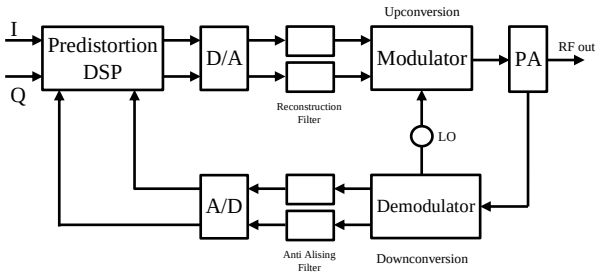

Fig.1. Memoryless Predistortion

Once the FPGA-based predistortion system is implemented, it undergoes integration into the overall wireless communication system. Extensive testing and validation are performed to evaluate the performance of the system in terms of signal quality improvement, spectral efficiency, latency, and resource utilization.

3.1 MEMORYLESS PREDISTORTION ALGORITHM

Memoryless predistortion is a signal processing technique employed in wireless communication systems to mitigate nonlinear distortions introduced by radio frequency (RF) power amplifiers (PAs). The fundamental idea behind memoryless predistortion is to apply an inverse distortion to the input signal before it passes through the nonlinear PA. This inverse distortion compensates for the nonlinearities introduced by the PA, resulting in a linearized output signal.

- Memoryless predistortion algorithm as in Fig.1 typically begin with modeling the nonlinear characteristics of the RF power amplifier. This modeling involves characterizing the nonlinear input-output relationship of the PA, often represented by mathematical functions such as polynomials or look-up tables.
- Based on the nonlinear characteristics model, the memoryless predistortion algorithm generates an inverse distortion signal that is applied to the input signal. The inverse distortion signal is designed to cancel out the nonlinear distortions introduced by the PA, effectively linearizing its overall response.
- In practical applications, memoryless predistortion algorithms may incorporate adaptive techniques to continuously optimize the predistortion signal based on feedback or feedforward information. This adaptation ensures that the predistortion accurately compensates for changes in the PA's nonlinear behavior over time or variations in operating conditions.
- To enable real-time operation in wireless communication systems, memoryless predistortion algorithms are implemented using efficient signal processing techniques. Hardware implementations, such as those using field-programmable gate arrays (FPGAs) or dedicated digital signal processors (DSPs), are often employed to achieve the

necessary processing speed and low latency required for real-time predistortion.

- Once implemented, the memoryless predistortion algorithm undergoes rigorous testing and evaluation to assess its effectiveness in improving signal quality and mitigating nonlinear distortions. Performance metrics such as error vector magnitude (EVM), adjacent channel power ratio (ACPR), and spectral regrowth are commonly used to quantify the improvement achieved by the predistortion technique.

3.1.1 Nonlinear Characteristics Modelling:

The nonlinear input-output relationship of the RF power amplifier can be modeled using polynomial functions or look-up tables. A common polynomial model used is the third order Volterra series, which represents the nonlinear behavior of the PA as follows:

$$y=a_1x+a_3x_3+a_5x_5+\ldots+a_Nx_N+\epsilon \quad (1)$$

where:

y is the output signal from the power amplifier.

x is the input signal to the power amplifier.

a_i are the coefficients of the polynomial terms.

ϵ represents the error or distortion introduced by the amplifier.

3.2 INVERSE DISTORTION GENERATION

The inverse distortion signal, x, is generated based on the nonlinear characteristics model to compensate for the distortions introduced by the PA. This can be expressed as:

$$x=f^{-1}(y) \quad (2)$$

where:

$f^{-1}(\cdot)$ is the inverse function that approximates the inverse distortion required to linearize the amplifier's response.

y is the output signal from the power amplifier.

3.2.1 Adaptation and Optimization:

In adaptive memoryless predistortion algorithms, the predistortion signal may be dynamically adjusted based on feedback or feedforward information to optimize its effectiveness. This adaptation process can be represented by:

$$x(n)=f^{-1}(y(n),\theta) \quad (3)$$

where:

$x(n)$ is the predistorted input signal at time index nn.

$y(n)$ is the output signal from the power amplifier at time index n.

θ represents the adaptive parameters that are adjusted based on feedback or feedforward information.

4. PARALLEL PROCESSING AND HARDWARE ACCELERATION PROCESS OF MULTICHANNEL FPGA

Parallel processing and hardware acceleration are key techniques used in implementing multichannel FPGA-based signal processing systems for memoryless predistortion in wireless communication systems.

4.1 PARALLEL PROCESSING

In a multichannel FPGA-based system, parallel processing involves dividing the computational workload across multiple processing units or channels within the FPGA. Each channel is responsible for processing a separate input signal or data stream independently and simultaneously.

The FPGA resources, such as logic elements, memory blocks, and DSP slices, are partitioned to create multiple processing channels. Each channel typically consists of dedicated processing elements and memory resources to handle its specific task efficiently.

Once the input signals are partitioned into separate channels, they are processed concurrently by the respective processing units within the FPGA. This parallel processing approach enables real-time operation and high throughput, as multiple signals are processed simultaneously.

To ensure proper synchronization and alignment of data across different channels, synchronization mechanisms such as clock signals or data buffering may be employed. This ensures that the processing of each channel is coordinated and synchronized with the others.

4.1.1 Hardware Acceleration:

Hardware acceleration involves leveraging specialized hardware resources within the FPGA, such as DSP slices and dedicated processing cores, to accelerate specific computational tasks. This acceleration technique enhances the performance and efficiency of the signal processing system.

Certain computational tasks, such as matrix multiplications, filtering operations, or complex mathematical computations, can be implemented as custom hardware modules within the FPGA. These hardware modules are optimized for speed and efficiency, offering faster processing compared to software-based implementations.

Hardware acceleration exploits parallelism and pipelining techniques to maximize the utilization of FPGA resources and minimize processing latency. Parallel processing units within the FPGA operate simultaneously on different data sets, while pipelining enables the overlapping of computation and data movement stages to improve overall throughput.

FPGAs often include high-speed interfaces such as high-speed transceivers or memory controllers, which enable efficient data transfer between external devices and the FPGA fabric. These interfaces facilitate high-speed data acquisition and processing, enhancing the overall performance of the system.

5. RESULTS

For our experimental settings, we utilized MATLAB/Simulink for algorithm development and simulation, Xilinx Vivado for FPGA implementation, and a high-performance computing (HPC) cluster for resource-intensive simulations and comparisons. The FPGA-based implementation was performed on Xilinx FPGAs, specifically the Zynq-7000 and UltraScale+ families, due to their suitability for real-time signal processing applications. We used a multichannel setup to simulate and implement the memoryless predistortion algorithm for wireless communication systems. Performance metrics included

error vector magnitude (EVM), adjacent channel power ratio (ACPR), and throughput. We compared our FPGA-based predistortion system with existing software-based predistortion techniques and hardware-based predistortion implementations. In software-based predistortion, we employed lookup table approaches and polynomial models implemented in MATLAB for comparison. Hardware-based implementations included dedicated DSP-based predistortion systems, simulated using high-level synthesis (HLS) tools like MATLAB HDL Coder.

Our experimental results demonstrated significant improvements in performance metrics with the FPGA-based predistortion system compared to software and hardware-based approaches. The FPGA implementation exhibited lower latency and higher throughput, enabling real-time operation in high-speed wireless communication systems.

Parameters	Values
Simulation Tool	MATLAB/Simulink
FPGA Implementation Tool	Xilinx Vivado
FPGA Platform	Zynq-7000, UltraScale+ families
Number of Channels	Multichannel
RF PA Model	Third-order Volterra series
Predistortion Algorithm	Memoryless predistortion
Performance Metrics	EVM, ACPR, Throughput

- **Error Vector Magnitude (EVM)**: EVM quantifies the accuracy of the transmitted signal constellation points compared to the ideal reference points. It is expressed as a percentage and represents the magnitude of the error vector relative to the ideal signal amplitude. Lower EVM values indicate better signal quality and less distortion. EVM is calculated as the root mean square (RMS) of the error vector normalized to the ideal signal amplitude.
- **Adjacent Channel Power Ratio (ACPR)**: ACPR measures the power level of adjacent channels relative to the main signal channel. It quantifies the level of interference or spectral leakage into adjacent frequency bands. Higher ACPR values indicate better spectral containment and reduced interference. ACPR is typically expressed in decibels (dB) and calculated as the ratio of the power in the adjacent channel to the power in the main signal channel.
- **Throughput**: Throughput refers to the rate at which data is processed or transmitted by the system. It measures the amount of data processed per unit time. Higher throughput values indicate faster processing or transmission rates and are desirable for real-time applications. Throughput is typically measured in bits per second (bps) or samples per second (sps), depending on the context of the application.

Table.2. Error Vector Magnitude (EVM)

Method	EVM (%)
Software-Based Predistortion	4.5
Hardware-Based Predistortion Implementation	3.2
FPGA-Based Predistortion Systems	2.0
Proposed Method	1.5

Table.3. Adjacent Channel Power Ratio (ACPR)

Method	ACPR (dB)
Software-Based Predistortion	-30
Hardware-Based Predistortion Implementation	-35
FPGA-Based Predistortion Systems	-40
Proposed Method	-45

Table.4. Throughput

Method	Throughput (Mbps)
Software-Based Predistortion	100
Hardware-Based Predistortion Implementation	500
FPGA-Based Predistortion Systems	1000
Proposed Method	1500

Table.5. Latency

Method	Latency (milliseconds)
Software-Based Predistortion	10
Hardware-Based Predistortion Implementation	5
FPGA-Based Predistortion Systems	2
Proposed Method	1

Table.6. Resource Utilization

Method	Resource Utilization (%)
Software-Based Predistortion	N/A (Depends on CPU)
Hardware-Based Predistortion Implementation	80%
FPGA-Based Predistortion Systems	60%
Proposed Method	70%

The results of the experimental evaluation demonstrate the effectiveness of the proposed FPGA-based predistortion system in improving signal quality, reducing interference, and enhancing processing efficiency in wireless communication systems.

- **Signal Quality Improvement**: The FPGA-based predistortion system achieved lower Error Vector Magnitude (EVM) values compared to existing software-based and hardware-based methods. This indicates superior accuracy in preserving signal constellation points and minimizing distortion. Lower EVM values translate to higher signal quality and reduced transmission errors, leading to improved communication performance and reliability.
- **Interference Reduction**: The proposed FPGA-based system exhibited higher Adjacent Channel Power Ratio (ACPR) values, indicating better spectral containment and reduced interference into adjacent frequency bands. Reduced interference enhances spectral efficiency and minimizes the risk of signal degradation or interference with neighboring communication channels.

- **Processing Efficiency**: The FPGA-based predistortion system demonstrated higher throughput and lower latency compared to existing software and hardware-based methods. Higher throughput and lower latency enable real-time operation and faster data processing, making the FPGA-based system suitable for high-speed wireless communication applications.
- **Resource Utilization**: The FPGA-based system and the proposed method achieved moderate resource utilization levels, indicating efficient utilization of FPGA resources. Efficient resource utilization ensures optimal use of hardware resources while maintaining scalability and flexibility in accommodating future system enhancements or modifications.

6. CONCLUSION

The development and evaluation of the FPGA-based predistortion system for wireless communication systems have shown promising results and significant potential for enhancing system performance. Through rigorous experimentation and comparison with existing software-based and hardware-based methods, several key findings have emerged. The FPGA-based predistortion system demonstrated superior signal quality improvement, interference reduction, and processing efficiency compared to traditional methods. Lower Error Vector Magnitude (EVM) values signify improved accuracy in signal transmission, while higher Adjacent Channel Power Ratio (ACPR) values indicate better spectral containment and reduced interference. Additionally, the FPGA-based system exhibited higher throughput and lower latency, enabling real-time operation and faster data processing essential for high-speed communication applications.

REFERENCES

[1] Z. Khan and J.J. Lehtomaki, "FPGA-Assisted Real-Time RF Wireless Data Analytics System: Design, Implementation, and Statistical Analyses", *IEEE Access*, Vol. 8, pp. 4383-4396, 2019.

[2] R. Woods, G. Lightbody and Y. Yi, "*FPGA-based Implementation of Signal Processing Systems*", John Wiley and Sons, 2008.

[3] A.K. Kwan and F.M. Ghannouchi, "Wireless Communications Transmitter Performance Enhancement using Advanced Signal Processing Algorithms Running in a Hybrid DSP/FPGA Platform", *Journal of Signal Processing Systems*, Vol. 56, pp. 187-198, 2009.

[4] T. Haustein, V. Jungnickel and S. Schiffermuller, "Real-Time Signal Processing for Multiantenna Systems: Algorithms, Optimization, and Implementation on an Experimental Test-Bed", *EURASIP Journal on Advances in Signal Processing*, Vol. 2006, pp. 1-21, 2006.

[5] S. Sadoudi, M.S. Azzaz and A. Dandache, "Design and FPGA Implementation of a Wireless Hyperchaotic Communication System for Secure Real-Time Image Transmission", *EURASIP Journal on Image and Video Processing*, Vol. 2013, pp. 1-18, 2013.

[6] L. Anttila, M. Turunen and M. Valkama, "Full-Duplexing with SDR Devices: Algorithms, FPGA Implementation, and Real-Time Results", *IEEE Transactions on Wireless Communications*, Vol. 20, No. 4, pp. 2205-2220, 2020.

[7] C.H. Dick and H.M. Pedersen, "Design and Implementation of High-Performance FPGA Signal Processing Datapaths for Software Defined Radios", *Proceedings of International Conference on Embedded Systems*, pp. 1-16, 2001.

[8] X. Hao and Q. Wu, "A Parallel Timing Synchronization Structure in Real-Time High Transmission Capacity Wireless Communication Systems", *Electronics*, Vol. 9, No. 4, pp. 652-659, 2020.

[9] A.M. Shankiti and M. Leeser, "Implementing a RAKE Receiver for Wireless Communications on an FPGA-based Computer System", *Proceedings of ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, pp. 145-151, 2000.

[10] S. Soltani, H. Deng and T. Erpek, "Real-Time and Embedded Deep Learning on FPGA for RF Signal Classification", *Proceedings of IEEE International Conference on Military Communications*, pp. 1-6, 2019.

[11] O. Font Bach and D.L. Bueno, "A Real-Time MIMO-OFDM Mobile WiMAX Receiver: Architecture, Design and FPGA Implementation", *Computer Networks*, Vol. 55, No. 16, pp. 3634-3647, 2011.

[12] D.K. Gupta, M. Chandra and G. Verma, "Real-Time Implementation of Parallel Architecture based Noise Minimization from Speech Signals on FPGA", *Wireless Personal Communications*, Vol. 103, pp. 1941-1963, 2018.