IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 14 January 2026, accepted 28 January 2026, date of publication 6 February 2026, date of current version 11 February 2026.

Digital Object Identifier 10.1109/ACCESS.2026.3662104

RESEARCH ARTICLE

Efficient Hardware and Software Co-Design of Adaptive Digital Beamforming With DoA Estimation in RFSoC for 5G Communication Systems

NAGESWARA RAO DUSARI, (Graduate Student Member, IEEE), AND MEENAKSHI RAWAT, (Member, IEEE)
Department of Electronics and Communication Engineering, Indian Institute of Technology Roorkee, Roorkee, Uttarakhand 247667, India
Corresponding author: Meenakshi Rawat (meenakshi.rawat@ece.iitr.ac.in)

ABSTRACT Adaptive digital beamforming is a crucial technique used in 5G communication systems for transmitting or receiving signals only in the desired directions. The estimation of the desired signal's direction of arrival (DoA) is used to implement beamforming. Implementing computationally complex beamforming and DoA algorithms together in hardware with low latency, less hardware resource utilization, and good accuracy for 5G communication applications is challenging. Therefore, this paper proposes an efficient hardware and software co-design approach for adaptive beamforming with DoA estimation using a radio frequency system on chip (RFSoC) evaluation board for the first time. The DoA intellectual property (IP) is created in the programmable logic (PL) part of RFSoC. The most popular high-resolution multiple signal classification (MUSIC) algorithm is used for DoA estimation. Eigenvalue decomposition (EVD) is one of the major operations of the MUSIC algorithm. Therefore, in this work, EVD is calculated efficiently in the PL after converting a complex covariance matrix to a real-valued matrix, which significantly reduces resource utilization. Adaptive beamforming is implemented in the processing system (PS) of RFSoC by utilizing the DoA of the desired signal, which is calculated using the DoA IP. The proposed PL and PS co-design of adaptive beamforming requires less resource utilization. To validate the proposed design, beamforming measurement is done in an anechoic chamber with a 1 × 4 antenna array with a 20 MHz bandwidth 5G new radio (NR) signal. The measurement results show that accurate DoA is estimated, and the beam is formed in the desired direction.

INDEX TERMS Digital beamforming, direction of arrival (DoA) estimation, hardware and software co-design, radio frequency system on chip (RFSoC).

I. INTRODUCTION

The 5G technology in wireless communication provides high data rates, low latency, and a large spectrum, but path losses are high due to higher frequencies. Beamforming is one of the crucial techniques to overcome path losses by transmitting and receiving signals only in the desired direction instead of omnidirectionally. Beamforming also increases the signal-to-interference plus noise ratio (SINR), spectral efficiency, energy efficiency, and system security [1].

Generally, adaptive beamforming can be done using a reference signal or the direction of arrival (DoA) of the desired signal. The major advantage of DoA-based adaptive beamforming is that it does not require transmitted signal data at the receiver. The minimum variance distortionless response (MVDR) is the most popular DoA-based adaptive digital beamforming (ADBF) technique [2]. The multiple signal classification (MUSIC) algorithm is a high-resolution

The associate editor coordinating the review of this manuscript and approving it for publication was Tiago Cruz.

and most accurate DoA estimation algorithm, but its computational complexity is high [3]. Calculating the inverse of the covariance matrix in the MVDR algorithm and eigenvalue decomposition (EVD) in the MUSIC algorithm are the main operations, and these operations are computationally complex. Implementing such computationally complex operations in hardware is challenging. Moreover, complexity will increase further when modulated complex IQ signals are considered for 5G applications instead of real-valued signals.

To address these challenges, a programmable logic (PL) and processing system (PS) co-design approach is essential. The PL in a modern radio frequency system on chip (RFSoC) enables high-throughput and parallel computation. Implementing computationally intensive operations like EVD in the PL, the system achieves significantly lower latency and higher throughput, which are essential for real-time signal processing. The PS is typically composed of Advanced RISC Machines (ARM) cores, which are more appropriate for sequential control logic, parameter configuration, and coordinating data movement between functional blocks. Implementing complex algorithms through hardware-software partitioning ensures computational efficiency, system-level flexibility, and optimal resource utilization and performance [4].

In [5], [6], [7], [8], and [9], authors presented the implementation of beamforming and DoA estimation using software-defined radio (SDR) hardware platforms. However, in these works, the DoA and ADBF algorithms are implemented in MATLAB/LabVIEW instead of hardware, which is not suitable for real-time applications. Few works have been reported in the literature on the implementation of algorithms in hardware [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21]. In [10] and [11], efficient methods are proposed to estimate DoAs of both coherent and uncorrelated sources, and algorithms are implemented in the TMS320C6678 digital signal processor for practical realization. However, hardware measurements are not performed in real time using a physical antenna array. References [12], [13], [14], [15] implemented the DoA algorithm in field programmable gate array (FPGA) hardware. In [12] and [13], the covariance matrix was decomposed into LU, Cholesky, and LDL factorization and used to estimate the DoAs. These methods reduced the complexity without using traditional EVD for DoA estimation. However, the implementation of these methods achieves low latency at the cost of higher resource utilization. Additionally, the integration of beamforming with DoA estimation on FPGA platforms intensifies resource demands. In [14], a low-latency and resource-efficient DoA algorithm was implemented using an algebraic method. However, an algebraic method can only be suitable for solving up to 3 × 3 real matrices. Moreover, there is no general algebraic equation for polynomial order more than four according to the Abel-Ruffini theorem [22]. In [15], the PL-PS co-design of DoA estimation using RFSoC for complex-valued signals is presented. The EVD for complex-valued signals is calculated directly without any transformation, which requires many hardware resources. Moreover, the experimental validation is also not performed in [14] and [15].

Digital beamforming with DoA estimation is implemented in [16], [17], and [18]. In [16], robust adaptive beamforming is developed by jointly estimating the interference location and array calibration errors in partly calibrated arrays. However, the proposed method is validated only through simulations and not implemented on hardware. In [17], beamforming is implemented in an FPGA. However, DoA is estimated using LabVIEW software, and the estimated DoA is provided to the FPGA for beamforming. In [18], only the beamforming output calculation is implemented in the FPGA, and the remaining beamforming coefficient generation and DoA estimation are implemented in the LabVIEW software on the PC, which is not suitable for real-time applications.

The beamforming using RFSoC is implemented in [19], [20], and [21]. In [19], receiver beamforming is implemented in an RFSoC device using the Xilinx ZC1275 evaluation board. However, beamforming weights are computed offline in MATLAB and subsequently transferred to the PL through a joint test action group (JTAG)–advanced extensible interface (AXI). Within the PL, the received analog-to-digital converter (ADC) data streams are multiplied by the precomputed complex weights and digitally summed to produce the beamformed output. In [20], digital beamforming based on a conventional weight-and-sum approach is implemented on an RFSoC platform. The beamforming weights are predefined for specific steering angles. However, no adaptive DoA estimation or adaptive beamforming algorithm is implemented on the RFSoC. In [21], adaptive beamforming using a diagonally loaded MVDR algorithm is implemented in RFSoC. However, DoA estimation is not included, and resource utilization is very high.

This paper proposes the real-time implementation of ADBF and DoA estimation together for complex-valued 5G new radio (NR) signals in RFSoC based on efficient hardware and software co-design for the first time to the best of our knowledge. Specifically, DoA intellectual property (IP) is created in PL to accelerate the EVD and peak search operations, while the major part of beamforming is implemented in PS. Therefore, PL and PS co-design will help to utilize fewer resources and achieve low latency in an optimized way. Moreover, the inverse of the covariance matrix for the MVDR algorithm is calculated by utilizing eigenvalues and eigenvectors of the DoA IP to accelerate the beamforming algorithm. The unitary transformation method (UTM) is used to convert a complex covariance matrix into a real-valued covariance matrix. Consequently, both EVD using the Jacobi iteration method and the MUSIC peak search for DoA estimation are performed entirely using real-valued operations. This transformation significantly reduces computational complexity and improves numerical efficiency compared to conventional complex-valued operations.

The RFSoC ZCU216 is used in this work for hardware implementation of algorithms as well as experimental validation. RFSoC integrates an FPGA, ARM processors, and direct RF-high sampling rate data converters in a single chip, which will have a small footprint, high flexibility, and low power solution for 5G/beyond radio applications [23]. Moreover, ZCU216 contains 16 transmitters and 16 receivers, which is also suitable for implementing large array beamforming and massive multiple-input multiple-output (massive MIMO) techniques for next-generation networks. The proposed design is experimentally validated in an anechoic chamber using a hardware measurement setup mainly consisting of RFSoC ZCU216, a horn antenna for transmission, and a 1 × 4 microstrip antenna array for receiving 5G NR signals.

The main contributions of this paper are summarized as follows:

- A real-time adaptive digital beamforming system integrated with DoA estimation is developed on an RFSoC platform for the first time.
- A novel PL-PS co-design framework is proposed to improve computational efficiency and reduce both hardware resource usage and overall system latency.
- A real-valued transformation method based on unitary matrices is applied to the complex covariance matrix, allowing the MUSIC algorithm to be executed with lower arithmetic complexity. This transformation significantly reduces resource utilization.
- A two-stage spectral peak search method with variable step sizes is employed to accelerate peak detection in the MUSIC spectrum, thereby reducing execution time without compromising estimation accuracy.

The remainder of this paper is organized as follows. Section II presents the MVDR and MUSIC algorithms for DoA-based adaptive beamforming, along with the UTM and the Jacobi iteration method used for eigenvalue and eigenvector computation. Section III introduces the proposed PL–PS co-design architecture for DoA estimation and beamforming. Section IV provides a detailed complexity analysis of the UTM-based MUSIC algorithm. Section V discusses simulation results of the MUSIC algorithm with UTM under various operating scenarios. Section VI describes the hardware measurement setup and experimental results. Finally, Section VII concludes the paper.

II. DOA-BASED ADAPTIVE BEAMFORMING

In beamforming, the direction of the beam can be changed by changing the phase and amplitude of array signals, which are referred to as weights. The weights are dynamically updated according to changes in user directions in adaptive beamforming. The weights in the adaptive beamforming can be calculated based on the reference signal or the DoA of the desired signal from the received signals. The DoA-based adaptive beamforming is more popular and suitable for real-time applications because it does not require the desired signal at the receiver. The MVDR algorithm is one of the efficient DoA-based adaptive beamforming algorithms for updating weights. The block diagram of DoA-based ADBF at the receiver is shown in Fig. 1. First, the DoA of the desired signal is estimated using the MUSIC algorithm after receiving signals from the antenna array. The MVDR algorithm updates the weights by using the estimated DoA. Finally, the beamforming output is calculated using updated weight vectors and received signals.

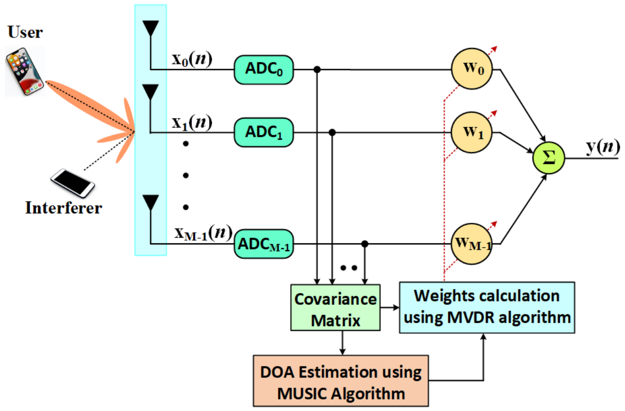

FIGURE 1. Block diagram of the adaptive digital beamforming (ADBF) system based on direction-of-arrival (DoA) estimation.

A. ARRAY SIGNAL MODEL

Consider a uniform linear array (ULA) consisting of M antenna elements and K uncorrelated source signals impinging on the array. Let $s_0(\ell)$ denote the signal from the desired source, while the remaining $[s_1(\ell), s_2(\ell), \ldots, s_{K-1}(\ell)]$ are interference signals. The signal received at the m^{th} antenna element at the discrete-time instant ℓ can be expressed as:

$$\boldsymbol{x}_m(\ell) = \boldsymbol{a}(\theta_0)\, s_0(\ell) + \sum_{k=1}^{K-1} \boldsymbol{a}(\theta_k)\, \boldsymbol{s}_k(\ell) + \boldsymbol{n}_m(\ell) \qquad (1)$$

where $\boldsymbol{a}(\theta_k)$ is steering vector corresponding to k^{th} source signal and $\boldsymbol{n}_m(\ell)$ is noise vector at m^{th} antenna element. Here $m = 0, 1, 2, \ldots M - 1$. The steering vector $\mathbf{a}(\theta_k)$ can be expressed as:

$$\boldsymbol{a}(\theta_k) = [1, e^{j(\frac{2\pi}{\lambda})d \sin\theta_k}, \ldots, e^{j(\frac{2\pi}{\lambda})(M-1)d \sin\theta_k}] \qquad (2)$$

where λ is carrier wavelength and d is the distance between two consecutive antenna elements.

B. MUSIC ALGORITHM

The MUSIC algorithm is a high-resolution method used for accurate DoA estimation. It consists of three main computational steps to determine the DoA of incoming signals. First, the covariance matrix $\boldsymbol{R}_{xx}$ of the received signal data is computed. Second, EVD is performed on the covariance matrix to separate the signal and noise subspaces. Finally, a spectral peak function is evaluated over a predefined angular range, typically from $[-90°, 90°]$ with a specified step size to identify the angle corresponding to the signal's arrival direction by locating the spectral peaks.

The EVD and spectral peak search (SPS) functions are computationally intensive and require significant hardware resources for practical implementation. Hardware-efficient approaches for eigenvalue and eigenvector computation include the Jacobi iteration method, approximate Jacobi methods, algebraic methods, and QR decomposition [24], [25]. The EVD is done using the Jacobi method for a complex-valued covariance matrix to estimate DoA using the MUSIC algorithm in [25]. However, performing EVD directly on complex-valued signals inherently increases complexity due to the necessity of complex multiplications. Moreover, the SPS function also becomes computationally heavier when handling complex eigenvectors. To address these challenges, in this work, the complex-valued covariance matrix is converted to a real-valued matrix using the UTM, and then EVD and SPS functions are calculated without any complex multiplications [26].

1) CALCULATING COVARIANCE MATRIX

The covariance matrix ($\boldsymbol{R}_{xx}$) of the received signal matrix ($\boldsymbol{X}$) is defined as:

$$\boldsymbol{R}_{xx} = \mathbb{E}[\boldsymbol{X}\boldsymbol{X}^H] \tag{3}$$

where $\boldsymbol{X} = [\boldsymbol{x}_0(\ell), \boldsymbol{x}_1(\ell), \ldots \boldsymbol{x}_{M-1}(\ell)]$, $\mathbb{E}$ is expectation operator and $(.)^H$ represents conjugate transpose.

This complex-valued covariance matrix (CVCM) is converted to a real-valued covariance matrix (RVCM) using the UTM approach. The UTM utilizes the hermitian and persymmetric properties of the covariance matrix and finds the RVCM as:

$$\boldsymbol{R} = \boldsymbol{U}\boldsymbol{R}_{xx}\boldsymbol{U}^H \tag{4}$$

where $\boldsymbol{R}$ is RVCM with size of $M \times M$. $\boldsymbol{U}$ is the unitary matrix, which is defined as:

$$\boldsymbol{U} = \frac{1}{\sqrt{2}} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{S} \\ j\boldsymbol{S} & -j\boldsymbol{I} \end{bmatrix} \tag{5}$$

where $\boldsymbol{I}$ is an identity matrix, and $\boldsymbol{S}$ is a matrix in which all secondary diagonal elements are one, and the remaining elements are zero with the size of $M/2 \times M/2$.

2) EIGENVALUE DECOMPOSITION

The Jacobi method finds the eigenvalues and eigenvectors of a symmetric RVCM. The RVCM is transformed into a diagonal matrix using the sequence of Givens rotation [24].

$$\boldsymbol{R}^{k+1} = \boldsymbol{J}_{pq}^T \boldsymbol{R}^k \boldsymbol{J}_{pq} \tag{6}$$

where $k = 1, 2, \ldots$ denotes the iteration index and, $\boldsymbol{J}_{pq}$ is Jacobi rotation matrix. For p and q values of k^{th} rotation, the elements of $\boldsymbol{J}_{pq}$ matrix are defined as

$$\begin{aligned} J_{pq}[p,p] &= J_{pq}[q,q] = cos(\theta) \\ J_{pq}[p,q] &= -J_{qp}[q,q] = sin(\theta) \\ J_{pq}[i,i] &= 1 \quad \forall \quad i \neq p \,\&\, q \end{aligned} \tag{7}$$

where the θ is the rotation angle. The rest of the elements of $\boldsymbol{J}_{pq}$ are zero. The p and q values are calculated based on the maximum value of off-diagonal elements of $\boldsymbol{R}$

$$|R^k[p,q]| = \max_{i \neq j} |R^k[i,j]| \tag{8}$$

The rotation angle θ can be calculated as:

$$\theta = \frac{1}{2} \tan^{-1} \left(\frac{2R^k[p,q]}{R^k[p,p] - R^k[q,q]} \right) \tag{9}$$

The diagonal elements of $\boldsymbol{R}$ will become the eigenvalues when all off-diagonal elements of $\boldsymbol{R}$ are zero by iterative processing (6). The eigenvectors can be calculated iteratively as:

$$\boldsymbol{V}_{RVCM}^{k+1} = \boldsymbol{V}_{RVCM}^{k} \boldsymbol{J}_{pq} \tag{10}$$

The initial eigenvector ($\boldsymbol{V}_{RVCM}^0$) is taken as the identity matrix. The eigenvectors of CVCM and RVCM are related by

$$\boldsymbol{V}_{CVCM} = \boldsymbol{U}^H \boldsymbol{V}_{RVCM} \tag{11}$$

where $\boldsymbol{V}_{CVCM}$ and $\boldsymbol{V}_{RVCM}$ are eigenvectors of CVCM and RVCM respectively. The eigenvalues of CVCM and RVCM are the same. The Jacobi algorithm for finding eigenvalues and eigenvectors of RVCM is described in algorithm 1.

Algorithm 1 The EVD of RVCM using Jacobi Algorithm

```
Input: RVCM ($\boldsymbol{R}$)
Output: Eigenvalues (VA),Eigenvectors ($\mathbf{V}_{RVCM}$)
VE ← I
while iteration < max_iteration do
    $R_{p,q}$ = max_off_diagonal_element($\boldsymbol{R}$)
    if (max_value < tolerance) then break
    $\theta = 0.5 \times \tan^{-1}\left(\frac{2R_{p,q}}{R_{p,p} - R_{q,q}}\right)$
    $\boldsymbol{J} \leftarrow \boldsymbol{I}$
    $A = \cos\theta,\ B = \sin\theta$
    $\boldsymbol{J}_{p,p} = A,\ \boldsymbol{J}_{p,q} = -B$
    $\boldsymbol{J}_{q,p} = B,\ \boldsymbol{J}_{q,q} = A$
    $\boldsymbol{R} \leftarrow \boldsymbol{J}^T\boldsymbol{R}\boldsymbol{J}$
    $\boldsymbol{V}_{RVCM} \leftarrow \boldsymbol{V}_{RVCM}\boldsymbol{J}$
end while
VA ← $\boldsymbol{R}$
return VA, $\boldsymbol{V}_{RVCM}$
```

The eigenvectors are divided into signal subspace (SS) and noise subspace (NS). The SS contains the eigenvector corresponding to the highest eigenvalue $\boldsymbol{V}_{SS} = \mathbf{v}_0$, and the NS contains remaining eigenvectors $\boldsymbol{V}_{NS} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots \boldsymbol{v}_{M-1}]$. The $\boldsymbol{V}_{SS}$ and $\boldsymbol{V}_{NS}$ are orthogonal to each other.

3) SPECTRAL PEAK SEARCH

In this stage, the MUSIC pseudo-spectrum is computed using the steering vector and the noise subspace eigenvectors. Peaks in this pseudo-spectrum indicate the directions of the incoming signals. The highest peaks correspond to the DoAs of the desired signals. The SPS function is mathematically expressed as:

$$\boldsymbol{P}(\theta_i) = \frac{1}{\boldsymbol{a}^H(\theta_i)\boldsymbol{V}_{NS}\boldsymbol{V}_{NS}^H\boldsymbol{a}(\theta_i)} \tag{12}$$

where $\theta_i \in [-90°, 90°]$ with some step size $\Delta\theta$. The denominator term of (12) will be zero for the desired signal angle due to the orthogonality between NS and SS. However, its value will be minimum instead of zero due to the existence of noise. Therefore, the DoA will be the angle for which the SPS function value is maximum.

The noise space vector in (12) is a complex-valued vector. To reduce the computational complexity, the SPS function can be modified by substituting the NS vector from (11) in (12):

$$\boldsymbol{P}(\theta_i) = \frac{1}{\boldsymbol{a}_R^T(\theta_i)\boldsymbol{V}_{RVNS}\boldsymbol{V}_{RVNS}^T\boldsymbol{a}_R(\theta_i)} \tag{13}$$

where $\boldsymbol{V}_{RVNS}$ is the real valued noise space (RVNS) vector obtained from RVCM eigenvector ($\boldsymbol{V}_{RVCM}$) in (10).

$$\begin{aligned}\boldsymbol{a}_R(\theta_i) &= \boldsymbol{U}\boldsymbol{a}(\theta_i)\\ &= \sqrt{2}\begin{bmatrix}\cos\left(\frac{M-1}{2}\phi(\theta_i)\right), \ldots, \cos\left(\frac{1}{2}\phi(\theta_i)\right),\\ \sin\left(\frac{1}{2}\phi(\theta_i)\right), \ldots, \sin\left(\frac{M-1}{2}\phi(\theta_i)\right)\end{bmatrix}^T\end{aligned} \tag{14}$$

Therefore, calculating the SPS function involves only real computations, which reduces the hardware computational complexity.

The step size value plays a critical role in computing the SPS function. A smaller step size yields higher angular resolution but increases execution time, whereas a larger step size reduces computation time at the cost of accuracy. Therefore, a two-stage adaptive step size approach is employed to achieve both high accuracy and less execution time simultaneously. The coarse DoA denoted by θ_c will be estimated in the first stage with a large step size ($\Delta\theta_l$) over the entire angular range $\theta_i \in [-90° : \Delta\theta_l : 90°]$. The fine DoA will be calculated in the second stage with a small step size ($\Delta\theta_s$) over a localized angular window around the coarse DoA, starting from ($\theta_c - \Delta\theta_l$) to ($\theta_c + \Delta\theta_l$). This strategy significantly reduces computational time while preserving high estimation accuracy.

C. MVDR ALGORITHM

The MVDR algorithm finds the weights by minimizing output power while maintaining a unitary steering array response at the estimated DoA of the desired signal, which will maximize the SINR. Mathematically, it can be expressed as:

$$\min_{\mathbf{w}} \boldsymbol{w}^H\boldsymbol{R}_{xx}\boldsymbol{w}, \quad \text{subject to } \boldsymbol{w}^H\boldsymbol{a}(\theta_0) = 1 \tag{15}$$

Using the Lagrange multiplier method, the optimum weights can be obtained as:

$$\boldsymbol{w}_{opt} = \frac{\boldsymbol{R}_{xx}^{-1}\boldsymbol{a}(\theta_0)}{\boldsymbol{a}^H(\theta_0)\boldsymbol{R}_{xx}^{-1}\boldsymbol{a}(\theta_0)} \tag{16}$$

The beamformer output ($\boldsymbol{y}(n)$) can be calculated after obtaining weights using the MVDR algorithm as:

$$\boldsymbol{y}(n) = \boldsymbol{w}_{opt}^H\boldsymbol{X} \tag{17}$$

Finding the inverse of CVCM is one of the major operations in the MVDR algorithm, and it is computationally complex if it is calculated directly. Therefore, in this work, the inverse of CVCM is calculated using eigenvalues and eigenvectors, which are obtained through the MUSIC algorithm. The inverse of CVCM is calculated as:

$$\boldsymbol{R}_{xx}^{-1} = \boldsymbol{V}_{CVCM}\left(\frac{1}{\boldsymbol{D}}\right)\boldsymbol{V}^H\boldsymbol{R}_{xx}^H \tag{18}$$

where $\boldsymbol{D}$ is the eigenvalues matrix of CVCM. The diagonal elements of $\boldsymbol{D}$ are eigenvalues, and the remaining elements are zeros.

III. PL AND PS CO-DESIGN ARCHITECTURE

In RFSoC platforms, algorithms can be implemented in two distinct but complementary domains: One is PL, which contains reconfigurable hardware logic blocks, and the other is PS, which includes general-purpose processors such as ARM cores. The PL provides specialized, fixed-function hardware acceleration designed explicitly for computationally intensive, parallelizable tasks, which reduces latency significantly and increases throughput compared to traditional software processing. On the other hand, the PS employs programmable general-purpose processors optimized for flexibility, ease of configuration, and management of sequential or less parallelizable tasks, but typically at the expense of higher latency and reduced throughput. Therefore, this work adopts the PL-PS co-design to get the advantages of both PS and PL by strategically partitioning tasks between hardware acceleration and software execution for optimal performance and efficient resource utilization.

The most computationally intensive operations in DoA-based ADBF systems are EVD and the inversion of the covariance matrix. To efficiently handle these operations, this work employs a strategic partitioning between the PL and PS. Specifically, the DoA estimation, which critically relies on EVD, is accelerated in the PL to ensure low latency and real-time performance. In contrast, the MVDR beamforming algorithm, requiring flexibility for varying parameters such as the number of snapshots and adaptive reconfiguration, is executed in the PS. Additionally, computationally demanding covariance matrix inversion is simplified by utilizing the eigenvalues and eigenvectors calculated in the PL, which significantly reduces complexity in the PS. This hierarchical approach, starting with the rapid DoA estimation in PL followed by flexible beamforming weight computation in PS, ensures optimal latency and resource utilization.

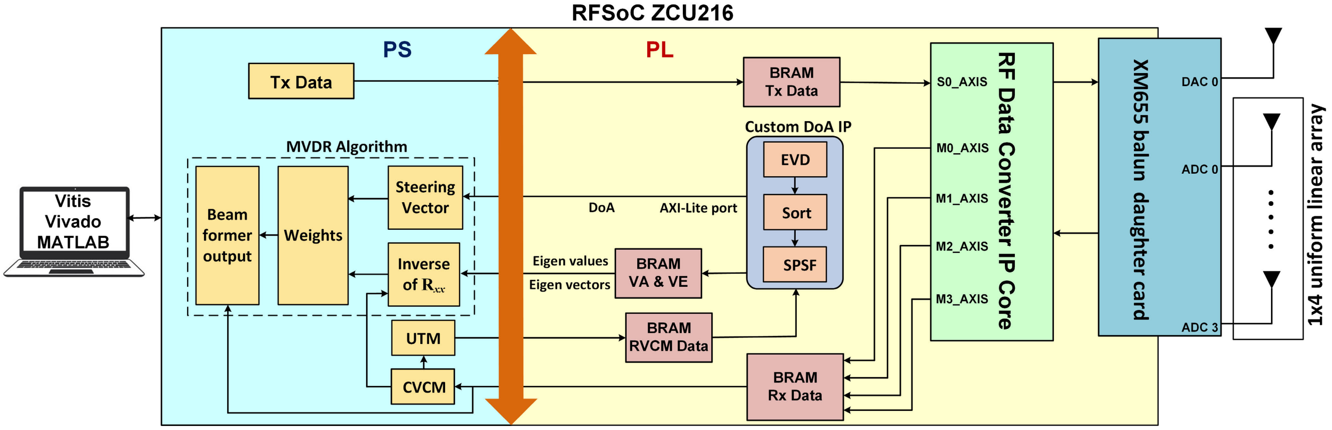

FIGURE 2. Block diagram of the proposed PL-PS co-design architecture for adaptive digital beamforming with the DoA estimation using RFSoC.

The following methodology is proposed to optimize the design:

1) The CVCM is converted to an RVCM using the UTM in the PS.
2) The DoA estimation IP is created in the PL using the high-level synthesis (HLS) design approach. The DoA IP performs EVD and SPS functions and returns the estimated DoA angle along with eigenvalues and eigenvectors to the PS.
3) The inverse of the CVCM for the MVDR algorithm in the PS is calculated by utilizing eigenvalues and eigenvectors obtained through DoA IP using the MUSIC algorithm.

The architecture of the proposed design is shown in Fig. 2. The receiver antenna array is connected to ADCs through the XM655 balun daughter card. The digital-to-analog converter (DAC) and ADCs are configured using an RF data converter IP core. The signals received by the ADCs are stored in block random-access memories (BRAMs) in the PL and then transmitted to the PS via the AXI4 interface. The complex covariance matrix is calculated using the received data in the PS. The CVCM is converted to RVCM using the UTM and then transmitted to the DoA IP in the PL. The DoA IP contains two BRAM interface ports. The DoA IP performs EVD using the Jacobi iteration method and then estimates the DoA by computing the SPS function.

A. IMPLEMENTATION OF EVD IN PL

Fig. 3 illustrates the hardware architecture of the EVD based on the Jacobi method. The RVCM (R) is the input to the EVD function, which is received from the PS. The decomposition process begins with the identification of the largest off-diagonal element in magnitude within the covariance matrix. This is achieved by selecting the non-diagonal elements of R one by one and comparing their absolute values iteratively to determine the global maximum and its corresponding indices (p,q). The abs() function from the HLS math library is used to find the absolute value. The iterative Jacobi process terminates if the maximum off-diagonal value is less than a predefined tolerance value (Tol). Otherwise, a Jacobi rotational transformation is computed to reduce the non-diagonal elements to zero or less than the tolerance value.

The rotation angle θ is calculated from the matrix elements ($R_{p,p}$, $R_{p,q}$, $R_{p,q}$) using basic arithmetic operations (multiplication, subtraction, and division), followed by an arctangent evaluation as defined in (9). The arctangent (inverse of tan) is calculated based on Taylor's series approximation by considering the first three terms ($x - x^3/3 + x^5/5$) to reduce the resource utilization. Calculating x^3 and x^5 in the traditional way will consume many digital signal processing (DSP) slices. Therefore, this expression is modified based on Horner's rule for effective implementation. The modified equation can be expressed as:

$$\arctan(x) = x\left(1 - 0.3333\, x_1 + 0.2\, x_1^2\right) \tag{19}$$

where $x_1 = x^2$. This structure requires only three multipliers.

Following angle calculation, its sine and cosine are computed to construct the orthogonal rotational matrix J as expressed in (7). These trigonometric functions are implemented using coordinate rotation digital computer (CORDIC) algorithm-based lookup tables (LUTs) in HLS for efficient hardware-friendly computation. Once the rotational matrix is formed, the covariance matrix is updated using orthogonal similarity transformation as described in (6). The eigenvectors matrix is updated by multiplying it with the rotational matrix. This process is iterated for a fixed number of sweeps (N) or until the maximum off-diagonal element becomes smaller than the Tol. After convergence, the diagonal elements of the transformed matrix contain the eigenvalues, and the $\boldsymbol{V}_{RVCM}$ matrix contains the corresponding eigenvectors.

B. IMPLEMENTATION OF SPS FUNCTION IN PL

The SPS function is computed using the steering vector and the noise subspace matrix, as defined in (13). For each angle θ, the steering vector $a(\theta)$ of size $1 \times M$ is

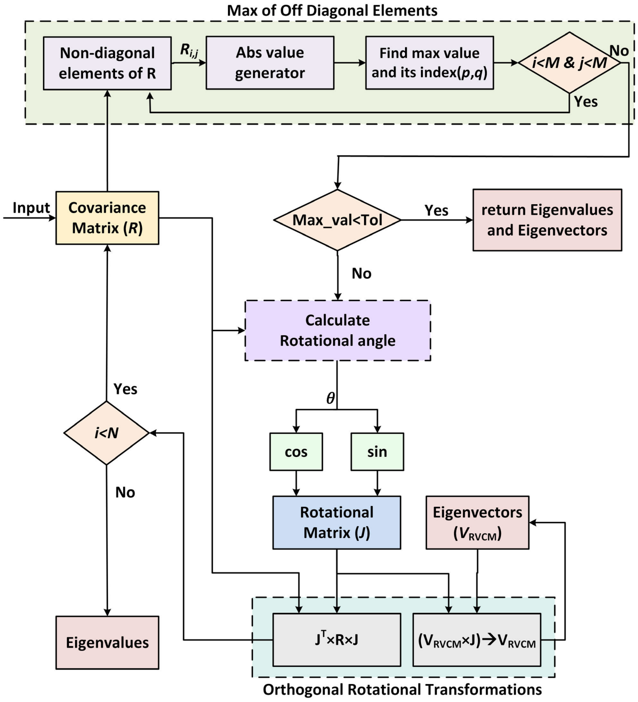
Max of Off Diagonal Elements
Non-diagonal elements of R
$R_{i,j}$
Abs value generator
Find max value and its index(p,q)
i<M & j<M
No
Yes
Input
Covariance Matrix (R)
Max_val<Tol
Yes
return Eigenvalues and Eigenvectors
No
Calculate Rotational angle
θ
cos
sin
Yes
i<N
No
Rotational Matrix (J)
Eigenvectors (V_RVCM)
Eigenvalues
J^T×R×J
(V_RVCM×J)→V_RVCM
Orthogonal Rotational Transformations

FIGURE 3. Block diagram of eigenvalue decomposition (EVD) using the Jacobi method.

generated, where $\theta \in [-90^\circ, 90^\circ]$ with a specific step size. Conventionally, the steering matrix is computed using trigonometric functions such as sine and cosine, which introduces hardware complexity due to repeated evaluations across all angular values. To optimize hardware resource utilization, the steering vectors corresponding to all discrete θ_i values are precomputed with a small step size and stored in BRAM as a steering matrix. During run-time, the appropriate steering vector is fetched from BRAM, thereby eliminating the need for real-time trigonometric computations.

Fig. 4 shows the hardware architecture of the SPS function. The eigenvectors of the noise subspace and steering vector are input to a matrix multiplication unit, which computes the denominator of the MUSIC spectrum expression (13). The computation involves a sequence of three matrix multiplications: the Hermitian transpose of the steering vector multiplied by the noise subspace matrix, followed by its conjugate transpose, and then re-multiplied with the steering vector to obtain the denominator term of the SPS function. These operations are implemented using DSP blocks to handle multiply–accumulate (MAC) operations efficiently.

Minimum value search is used by considering only the denominator term of the SPS function instead of the peak search of the SPS function to reduce the resource utilization. The comparator block compares the current value of the denominator term with the previous minimum value and then gives the updated minimum value (m). The index value (k) corresponding to the minimum value is stored in the temporary register. The DoA is estimated based on the final updated index value (k) and angle vector (θ) after completing all iterations in every stage. The estimated DoA will be equal to $\theta_i[k]$.

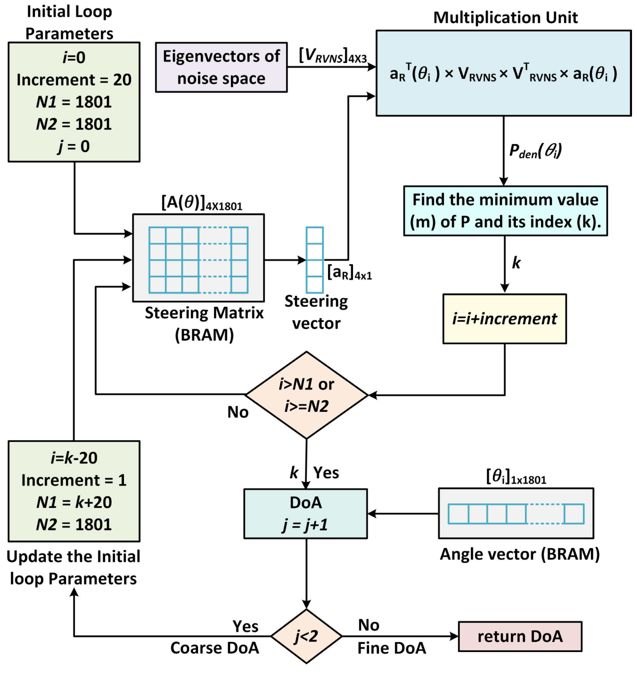

FIGURE 4. Block diagram of spectral peak search (SPS) function.

The DoA is estimated in two stages with adaptive step sizes. The coarse DoA is estimated with a large step size in the first stage, and then the fine DoA is estimated with a small step size in the second stage. The steering matrix created with a small step size can also be used as a steering matrix with a large step size by changing the initial value i, final value $N1$, and increment value of the loop. The increment value for the first-stage search is defined as $(\Delta\theta_l/\Delta\theta_s)$, which is equal to 20 for $\Delta\theta_l = 2$ and $\Delta\theta_s = 0.1$. In the first stage, the angular search is performed over the full range $\theta \in [-90^\circ, 90^\circ]$. Therefore, the initial value is set to zero, and the final value will be

$$\left(\frac{90-(-90)}{\Delta\theta_s}\right)+1 = 1801 \tag{20}$$

which defines the loop bounds for the first-stage search. The initial value and final value for the second-stage search are defined based on the coarse DoA estimated in the first stage. The second-stage angular search is performed over the interval

$$(\theta_c - \Delta\theta_l) \leq \theta \leq (\theta_c + \Delta\theta_l) \tag{21}$$

Accordingly, the initial and final values are given by

$$i = k - \left(\frac{\Delta\theta_l}{\Delta\theta_s}\right) = k - 20 \tag{22}$$

and

$$N_1 = k + \left(\frac{\Delta\theta_l}{\Delta\theta_s}\right) = k + 20 \tag{23}$$

respectively, where k represents the index corresponding to the coarse DoA. The increment value is set to 1 because the step size is 0.1 for the second stage.

This architecture enhances hardware efficiency by reducing real-time computations through a two-stage approach with an adaptive step size, storing precomputed steering matrix and angular vector values in BRAM, and employing parallel DSP-based matrix operations for high-throughput SPS computation.

C. IMPLEMENTATION OF MVDR ALGORITHM

The MVDR algorithm is implemented in the PS part of RFSoC based on bare-metal design in Vitis. The PS contains high-performance quad-core ARM Cortex-53 processors, which are suitable for implementing algorithms efficiently. The high-level language C with some basic libraries is used to program the MVDR algorithm in Vitis 2021.1. Calculating the inverse of the complex covariance matrix is more challenging using standard C math libraries. Therefore, the inverse of the matrix is calculated based on eigenvalue decomposition, which does not require calculating the direct inverse function. Moreover, the MUSIC algorithm is also based on eigenvalue decomposition. Therefore, the eigenvalues and eigenvectors calculated for RVCM in the PL are reused in the PS instead of being calculated again in the PS, which reduces the execution time significantly. The real-valued eigenvectors are converted to complex-valued eigenvectors in the PS using (11) after receiving eigenvalues and eigenvectors from the PL for obtaining eigenvectors of CVCM. The inverse of CVCM is calculated using (18). The steering vector is also calculated in the PS using (2) after receiving the estimated DoA value from the PL. Afterward, optimal weights are calculated using the steering vector and the inverse of CVCM using (16). Finally, the beamformer output can be calculated using (17).

IV. COMPLEXITY ANALYSIS OF MUSIC ALGORITHM WITH UTM

In the conventional approach, covariance computation, EVD, and SPS are carried out in the complex domain [25]. This significantly increases the arithmetic cost because one complex multiplication requires four real multiplications and two real additions, and one complex addition requires two real additions. After applying the UTM, the EVD is performed on RVCM using the real-valued Jacobi algorithm. In this approach, each Jacobi rotation requires only real multiplications and real additions. For an $M \times M$ matrix with L Jacobi iterations, both the real-valued and complex-valued Jacobi algorithms exhibit the same asymptotic computational complexity of $\mathcal{O}(M^3L)$. However, the constant factor is significantly lower in the real-valued case. Specifically, each rotation in the complex Jacobi method involves approximately four times more real multiplications and three to four times more real additions compared to the real Jacobi method, in addition to extra square-root and division operations required to compute the complex rotation parameters.

The computational overhead of the UTM itself is also minimal. In the general case, a full matrix multiplication involving three $M \times M$ matrices, i.e., computing $\mathbf{U}^H\mathbf{RU}$, requires $\mathcal{O}(M^3)$ operations. However, in the proposed implementation, the UTM exploits the sparsity and structure of the unitary matrix, whose non-zero elements are ± 1 or $\pm j$. Consequently, the UTM is implemented using only additions, subtractions, sign changes, and a final scalar multiplication, reducing the overall complexity to approximately $\mathcal{O}(M^2)$. For the 4×4 case considered in this work, the complete UTM requires only 16 additions/subtractions and 16 real multiplications by 0.5, resulting in negligible hardware overhead.

A detailed step-by-step complexity comparison is summarized in Table 1. Here, N denotes the number of snapshots, L the maximum number of Jacobi iterations, and P the number of scanning angles in the SPS. As shown in Table 1, the overall computational complexity of the MUSIC algorithm based on UTM is reduced by a factor of four in multiplications and by three to four times in additions/subtractions when compared to the conventional complex-valued MUSIC algorithm.

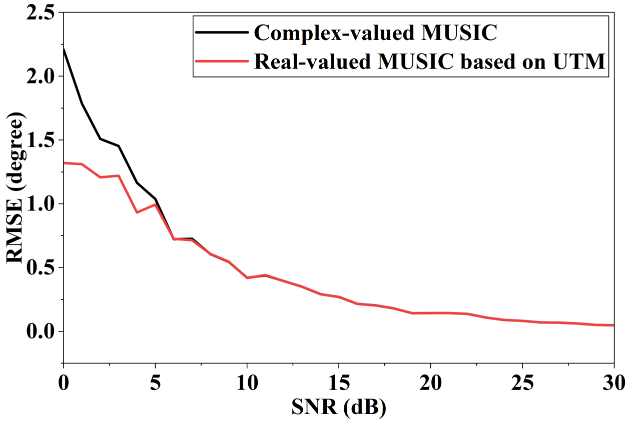

FIGURE 5. RMSE vs. SNR: Performance comparison of complex-valued MUSIC algorithm and real-valued MUSIC algorithm based on UTM and two-stage SPS function under different SNR conditions.

V. SIMULATION RESULTS OF MUSIC ALGORITHM WITH UTM

The complex-valued MUSIC algorithm and the real-valued MUSIC algorithm based on UTM exhibit comparable estimation performance. The UTM transforms the CVCM into an RVCM, thereby reducing computational complexity and numerical instability. This transformation enables a more reliable eigen-decomposition, leading to a more accurate separation between the signal subspace and the noise subspace. Fig. 5 shows the root mean square error (RMSE) vs. signal-to-noise ratio (SNR) curves for the case of a single source located at 20°, 1×4 ULA and 100 snapshots. SNR varies from 0 dB to 30 dB. As observed, the estimation accuracy improves with increasing SNR, and both algorithms achieve nearly

TABLE 1. Complexity analysis comparison (in terms of real arithmetic operations).

Step	Big-O Notation	Conventional Complex MUSIC		MUSIC based on UTM	
		No. of Multiplications	No. of Adds/Subs	No. of Multiplications	No. of Adds/Subs
Covariance Matrix	$\mathcal{O}(M^2N)$	$4M^2N$	$4M^2N-2M^2$	$4M^2N$	$4M^2N-2M^2$
UTM	$\mathcal{O}(M^2)$	–	–	M^2	$4M^2$
EVD (Jacobi)	$\mathcal{O}(M^3L)$	$(12M^3+13)L$	$(12M^3-6M^2+7)L$	$\mathbf{(3M^3+5)L}$	$\mathbf{(3M^3-3M^2+3)L}$
SPS	$\mathcal{O}(M^2PK)$	$(8M(M-K)+4M)P$	$(8M(M-K)+2K-2)P$	$\mathbf{(2M(M-K)+M)P}$	$\mathbf{(2M(M-K)-(M-K-1))P}$

identical performance at moderate and high SNR levels. However, at low SNR, the MUSIC algorithm incorporating UTM demonstrates slightly improved performance compared to the conventional complex-valued MUSIC algorithm, due to enhanced numerical robustness in subspace estimation.

A. TWO STAGE SPS

The two-stage SPS significantly reduces the number of iterations required to find the peak values compared to the full-resolution SPS, while preserving DoA estimation accuracy and sidelobe characteristics. To quantitatively assess its effectiveness, two scenarios are considered to compare the two-stage SPS with the conventional full-resolution MUSIC algorithm, particularly under closely spaced source conditions with an angular separation of 5°. In both scenarios, the number of antenna elements is $M = 8$, the number of snapshots is 100, and the SNR is 5 dB. The two-stage SPS consists of a coarse search with a step size of 1°, followed by a fine search with a step size of 0.1° centered around the coarse-stage estimates.

Fig. 6 illustrates the scenario in which the source angles are 20° and 25°, which coincide with the coarse grid points. In this case, the coarse-stage estimates are already well aligned with the true directions, and the subsequent fine-stage refinement preserves both the mainlobe shape and the sidelobe levels. Consequently, the spatial spectrum obtained using the two-stage SPS closely matches that of the full-resolution MUSIC algorithm. Fig. 7 shows the case where the source angles are 20.5° and 25.5°, which lie between the coarse grid points. Although the coarse-stage estimates alone are insufficient to accurately capture the true DoAs, the fine-stage refinement successfully recovers the correct source locations with high precision. Compared to the full-resolution MUSIC algorithm, a slight increase in sidelobe levels is observed when using the two-stage SPS approach. Nevertheless, the estimated DoAs remain accurate, demonstrating that the two-stage SPS effectively balances computational efficiency and estimation performance. A quantitative comparison among conventional full-resolution MUSIC, full-resolution MUSIC with UTM, and two-stage SPS MUSIC with UTM in terms of estimated DoAs and peak sidelobe level (PSLL) is shown in Table 2.

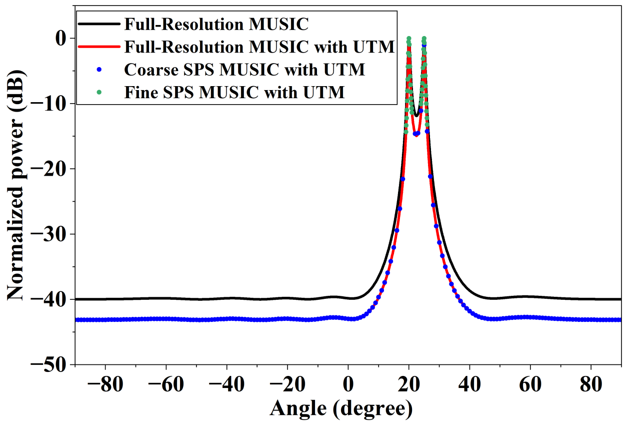

FIGURE 6. DoA estimation when sources are 20° and 25° using full resolution and two-stage SPS MUSIC algorithm.

B. PERFORMANCE UNDER COHERENT SOURCES

The MUSIC algorithm can accurately estimate DoA when the sources are either uncorrelated or partially correlated. However, it encounters challenges in scenarios involving perfectly correlated sources, such as multipath propagation environments. In these cases, the covariance matrix becomes singular and rank-deficient, and it is difficult to estimate an accurate DoA.

Spatial smoothing (SS) is one of the preprocessing techniques that addresses the rank deficiency of the covariance matrix caused by coherent sources [27]. This step is performed prior to applying the UTM, which converts the complex-valued covariance matrix into a real-valued form suitable for real-domain EVD and SPS in the MUSIC algorithm. To evaluate the effectiveness of spatial smoothing, simulations were conducted using an antenna array with $M = 8$ antenna elements, two coherent sources impinging from $\theta = 20°$ and 25°, and 1000 snapshots. The number of subarrays used for spatial smoothing is set to four. The performance of DoA estimation using the MUSIC algorithm without spatial smoothing, the MUSIC algorithm with spatial smoothing, and the MUSIC algorithm with UTM and spatial smoothing is shown in Fig. 8. Without spatial smoothing, the MUSIC algorithm fails to resolve

TABLE 2. Comparison of DoA estimates and PSLL for full-resolution and two-stage SPS MUSIC with UTM.

True Angles	Full resolution MUSIC		Two stage SPS MUSIC with UTM		
	Estimated DoA	PSLL (dB)	Estimated coarse DoA	Estimated fine DoA	PSLL (dB)
(20 °, 25 °)	(20 °, 25 °)	-39.55	(20 °, 25 °)	(20 °, 25 °)	-42.71
(20.5 °, 25.5 °)	(20.4 °, 25.5 °)	-36.59	(20 °, 25 °)	(20.4 °, 25.5 °)	-35.82

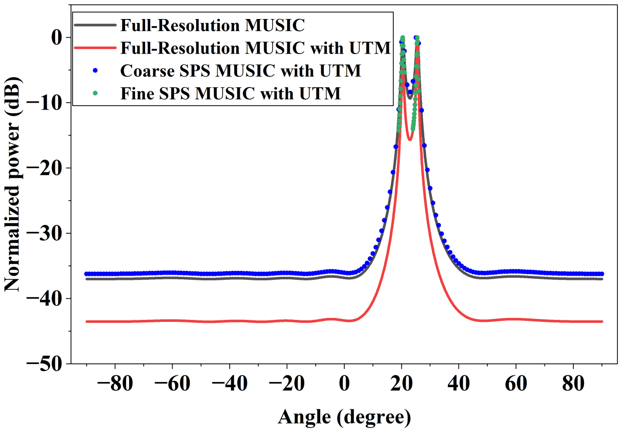

FIGURE 7. DoA estimation when sources are 20.5° and 25.5° using full resolution and two-stage SPS MUSIC algorithm.

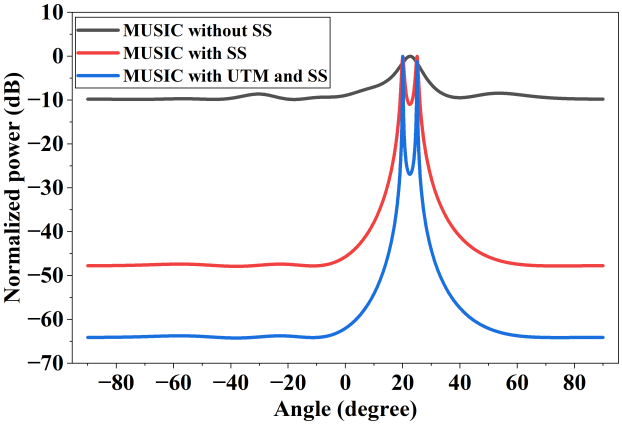

FIGURE 8. DoA estimation using the MUSIC algorithm for coherent signals.

coherent sources due to rank deficiency. The spectrum is broad and does not exhibit distinguishable peaks. With spatial smoothing, distinct peaks appear at the correct DoAs, indicating successful decorrelation and accurate estimation.

Furthermore, when spatial smoothing is combined with the UTM, the resulting spectrum exhibits improved sharpness and enhanced spectral resolution. The real-valued covariance matrix obtained after UTM improves numerical conditioning and stabilizes the eigen-decomposition process, leading to narrower main lobes and better sidelobe suppression even for closely spaced coherent sources. This combination enables efficient and accurate DoA estimation in the real-valued domain while preserving estimation accuracy.

C. PERFORMANCE UNDER RAYLEIGH FADING

Rayleigh fading conditions are considered to evaluate the performance of the DOA estimation based on UTM in the absence of a dominant line-of-sight (LoS) component. In this scenario, the received signal consists entirely of randomly scattered multipath components, resulting in Rayleigh-distributed channel amplitudes and uniformly distributed phases [28], [29]. The Rayleigh fading channel is modeled as a complex circularly symmetric Gaussian process, and simulations are carried out using an 8-element ULA, 1000 snapshots, 10 dB SNR, and two sources located at $-10°$ and 20°. The MUSIC pseudospectrum under Rayleigh fading is shown in Fig. 9. The MUSIC algorithm based on UTM and the conventional standard MUSIC algorithm are able to accurately estimate DoA under Rayleigh fading. Moreover, the MUSIC algorithm based on UTM has better sidelobe levels compared to the conventional standard MUSIC algorithm.

VI. MEASUREMENT SETUP AND RESULTS

The proposed design is validated through hardware measurement. The measurement setup is shown in Fig. 10. The measurement is done in the anechoic chamber. One DAC channel of RFSoC ZCU216 is connected to a horn antenna through a power amplifier (PA) for transmitting a 20 MHz bandwidth 5G NR signal. The RF lambda PA (RAMP01M22GA) is used to increase the power of the transmitting signal. The four ADCs of RFSoC are connected to a microstrip ULA to receive signals from different directions.

A. MULTI TILE SYNCHRONIZATION

Zynq UltraScale^{+} RFSoC ZCU216 is the third generation of the RFSoC evaluation board. ZCU216 contains 16 DACs and 16 ADCs, which support high sampling rates up to 9.85 gigasamples per second (GSPS) and 2.5 GSPS, respectively. All 16 ADCs are grouped into 4 tiles, and each tile contains 4 ADCs. ADCs of different tiles are not synchronized due to different phase-locked loop (PLL) clocks for each tile, which leads to random phase shifts between different ADC channels. Therefore, multi-tile synchronization (MTS) is done using a system reference (SYSREF) signal and PL clock before receiving signals for adaptive beamforming measurement [23].

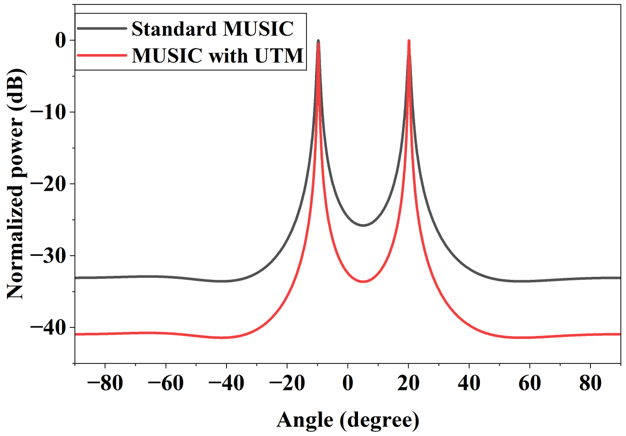

FIGURE 9. Spectrum of the standard MUSIC and UTM-based MUSIC algorithm for DoA estimation under Rayleigh fading.

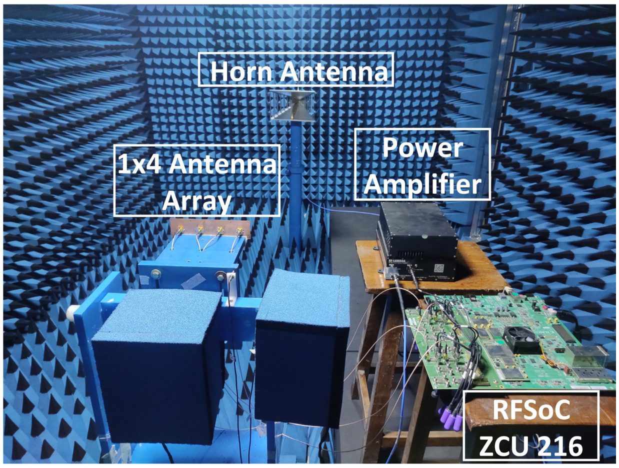

FIGURE 10. Adaptive beamforming measurement setup in the anechoic chamber.

The selection of these clock signals and the sampling rate of DAC and ADC play a very important role in synchronization and transmitting signals at the required frequency without distortions. The radio-frequency data converter (RFDC) is configured with a DAC/ADC sampling rate of 2.4 GHz and a numerically controlled oscillator (NCO) frequency of 1 GHz, operating in Nyquist zone 1, to enable signal transmission and reception at 3.4 GHz.

The PL clock should be an integer multiple of the AXI4 stream clock for MTS. The AXI4 stream clock depends on several parameters like sampling rate, interpolation, decimation, samples per AXI4 stream cycle, and complex or real data output. The AXI4 stream clock value can be calculated as:

$$\text{AXI4 stream clock} = \frac{(\text{Fs} \times \text{IQ mode})/\text{IR}}{\text{PL}_{\text{word}}} \tag{24}$$

where Fs is the sampling rate of DAC, IR is the interpolation rate, PL_{word} is the number of samples per AXI4 stream cycle, and IQ mode is 1 for complex output and 0 for real output.

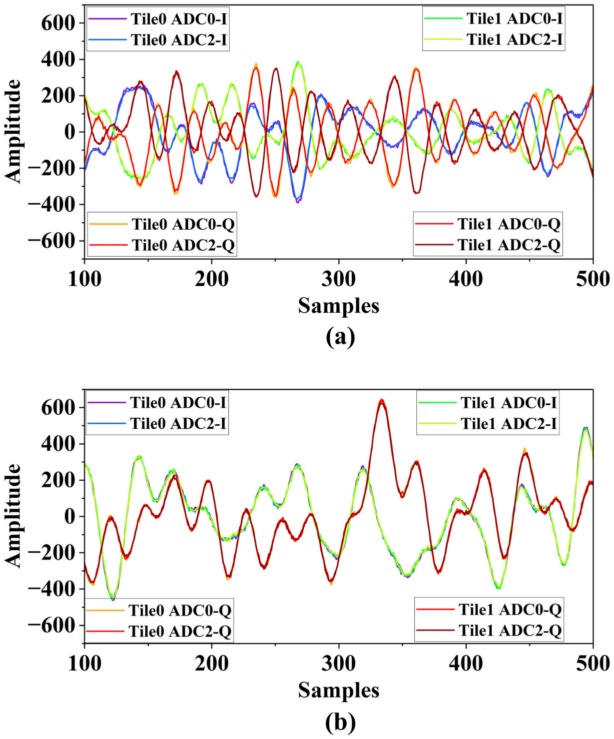

FIGURE 11. Received signals from multiple ADC tiles before and after MTS. (a) Shows the I/Q components of the received signals across four ADC channels before MTS, which have phase and amplitude misalignment between the tiles. (b) Shows the same channels after MTS, demonstrating successful synchronization with improved alignment in both I and Q components without any phase and timing offsets across RFSoC ADC tiles.

The AXI4 stream clock frequency will be 150 MHz when the interpolation and decimation factors are set to 4, and the number of samples per AXI4 stream cycle is 8. The SYSREF clock of 10 MHz and the PL clock of 150 MHz are used. The on-chip PLL is used to produce sample clocks by taking a reference clock of 150 MHz from the CLK104 module. The Xilinx XRFdc software driver in the PS is used to program the CLK104 module, and the RFDC driver application programming interfaces (APIs) are used to run the MTS [30].

To verify the MTS of RFSoC ZCU216, the complex IQ data is transmitted from DAC and received from 4 channels (ADC0 and ADC2 of tile0, and ADC0 and ADC2 of tile1). Fig. 11 shows the signals received before MTS and after MTS. There is a random phase shift between signals of different tiles before MTS. All signals received from different tiles are exactly aligned with zero phase shift after MTS.

B. ANTENNA ARRAY

A 1 × 4 ULA patch antenna is designed using a full-wave simulator CST Microwave Studio. The antenna is designed and fabricated on 20 mils thick dielectric substrate Rogers 4350B ($\varepsilon_r = 3.66$ and $tan\delta = 0.0037$) with copper cladding 0.017mm. The inter-element spacing between

TABLE 3. Resource utilization, execution time, and power consumption of DoA IP using different design approaches.

Resource Type	Total Available Resources	Design Approach-1 (FLP)	Design Approach-2 (FLP)		Design Approach-3 (FLP)		Design Approach-3 (FIP)	
		Utilized	Utilized	Savings (%)	Utilized	Savings (%)	Utilized	Savings (%)
BRAM	1080	19.5	13	33.33	13	33.33	**8**	**58.97**
DSP	4272	186	66	64.52	76	59.14	**27**	**85.48**
FF	850560	34219	13542	60.43	15177	55.65	**5536**	**83.82**
LUT	425280	31602	12423	60.69	13984	55.75	**5920**	**81.27**
Execution Time (ms)		1.83	1.46	20.22	0.12	93.44	**0.063**	**96.56**
Power consumption (W)		0.585	0.185	68.38	0.205	64.96	**0.059**	**89.91**

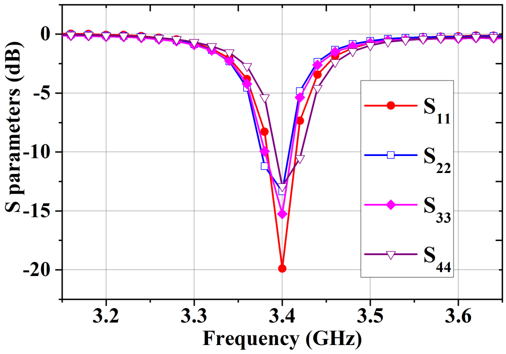

FIGURE 12. Measured reflection coefficients of the antenna elements across the frequency range of 3.2 GHz to 3.6 GHz, showing a resonance near 3.4 GHz.

adjacent radiating patch of ULA is $\lambda/2$. The ULA is designed with an operating frequency of 3.5 GHz in CST Studio software. The measured S-parameters are shown in Fig. 12. The operating frequency is slightly shifted to 3.4 GHz after fabrication due to fabrication tolerances. The bandwidth of ULA is 30 MHz at -10 dB return loss, which is sufficient for receiving 20 MHz 5G NR signals.

C. DOA IP

The DoA IP is implemented in PL using the Vitis HLS 2021.1 tool. The HLS design approach enables faster design and verification compared to the traditional hardware description language (HDL) approach [31], [32]. The DoA IP is generated from HLS, which performs EVD and SPS functions to generate the eigenvalues and eigenvectors and estimate the DoA for a given covariance matrix. The EVD function is implemented using the Jacobi iteration method, as described in Section III. The maximum number of iterations and tolerance values are taken as 100 and 10^{-5}, respectively, for the Jacobi algorithm. The "While" loop will terminate if the algorithm reaches either a maximum number of iterations or a tolerance. In the two-stage SPS function, the step size value used for coarse DoA estimation is 2°, and the step size value for fine DoA estimation is 0.1°. Therefore, a total number of iterations required for estimating the final DoA will be 132 in which 91 iterations are used for coarse DoA and 41 iterations for fine DoA estimation. If the conventional method is used with a small step size of 0.1° for the entire angular space [−90° : 90°], then the number of iterations required to calculate the DoA will be 1801. Therefore, 92.6% iterations were reduced by using a two-stage approach with adaptive step size compared to the conventional process.

TABLE 4. BRAM utilization for Design Approach-3 with fixed point implementation.

S. No	Variable	BRAMs Utilized
1	Angle vector	1
2	Steering matrix	4
3	Covariance matrix	1
4	EVD matrix	1
5	CORDIC cosine table	1
	Total	**8**

DoA IP is implemented using different design approaches. Design Approach-1 is based on the conventional method in which the complex Jacobi method is used for performing EVD, and a single-stage SPS function with a fixed step size of 0.1° is used for peak search. Design Approach-2 is based on the real-valued Jacobi method to perform EVD after converting the CVCM to RVCM using UTM as discussed in Section II. Therefore, Design Approach-2 contains only real-valued operations without any complex-valued operations. As a result, it achieves a significant reduction in hardware resources utilization and execution time compared to the Design Approach-1. Design Approach-3 also uses UTM for real-valued operations similar to Design Approach-2. Additionally, Design Approach-3 incorporates a two-stage approach with an adaptive step size for peak search to minimize latency while maintaining accuracy.

D. RESOURCE UTILIZATION

The resource utilization and execution time of DoA IP based on different design approaches are shown in Table 3. More than 33% of hardware resources are reduced in Design Approach-2 compared to Design Approach-1. The single-precision floating-point (FLP) data type is used in both Design Approaches 1 and 2. Choosing appropriate data types is very important while implementing algorithms in the

TABLE 5. DSPs utilization for Design Approach-3 with fixed point implementation.

Signal Processing Operations	No. of Multiplications	No. of DSPs Used	Comments
Rotational angle calculation using (9) and (19)	7 single-valued multiplications.	7	1 DSP for each multiplication.
Orthogonal rotational transformation using (6) and (10)	3 matrix multiplications. The size of each matrix is 4 × 4.	12	4 DSPs for each matrix multiplication.
SPS function denominator using (13)	3 matrix multiplications with different sizes of matrices.	8	4,3 and 1 DSPs are used for 1st, 2nd and 3rd matrix multiplications respectively.
Total	6 matrix multiplications. 7 single-valued multiplications.	27	The total number of DSPs will be equal to $5M + 7$.

TABLE 6. Comparison of DoA estimation.

Actual Location	DoA estimation using MATLAB	DoA estimation using DoA IP Based on Design Approach-3	
		FLP	FIP
15°	14.1°	14.09°	14.1°
-20°	-19°	-19°	-18.99°
-30°	-29.8°	-29.79°	-29.79°

TABLE 7. Sensitivity analysis showing the effect of varying fractional bit-widths on DoA estimation error and hardware resource utilization.

Data type	FIP with variation of fractional bits					FL
Word length (W,I)	(10,4)	(16,4)	(20,4)	(22,4)	(26,4)	32
Fractional bits	6	12	16	18	22	NA
BRAM	6	7.5	8	8.5	9.5	12
DSP	27	27	27	28	53	86
FF	3129	4615	5920	5987	6789	16136
LUT	2618	3783	5536	7794	6439	13795
DoA error	8.52 °	0.31 °	0.21 °	0.21 °	0.21 °	0.21 °

hardware to achieve the desired trade-off between resource utilization and accuracy. Resource utilization can be reduced by using a fixed-point (FIP) data type with an appropriate word length (WL) compared to the FLP data type. The final design, which is Design Approach-3, is implemented using both floating and fixed-point data types. The fixed-point word length is selected based on a detailed sensitivity analysis, as shown in Table 7. The results show that a (20, 4) fixed-point format achieves floating-point accuracy with significantly reduced hardware resource utilization, making it well-suited for the proposed architecture.

The Design Approaches 2 and 3 with FLP utilize almost the same amount of resources, but Design Approach 3 has less execution time compared to Design Approach 2 due to the two-stage SPS function with adaptive step size. Design Approach-3 with FIP exhibits the highest resource utilization savings, including a 58.97% reduction in BRAMs, 85.48% in DSPs, 83.82% in flip-flops (FFs), and 81.27% in LUTs, while achieving a 96.56% reduction in execution time compared to Design Approach-1. Tables 4 and 5 show how BRAMs and DSPs are utilized in implementing DoA IP based on Design Approach-3 using FIP data type. BRAMs are used to store precomputed angle values (θ_i), the steering matrix, the covariance matrix, the EVD, and the CORDIC table to find $cos(\theta)$ and $sin(\theta)$. The angle vector contains values from $-90°$ to $90°$ with a step size of $0.1°$. The total number of values in the angle vector is 1801. One 36-Kb BRAM is sufficient to store 1801 values with 20 bits for each value. The steering matrix contains 7204 values, and the number of 36-Kb BRAMs used to store the steering matrix will be 4.

The DSPs are used for performing multiplication operations in the algorithm. The design involves both matrix multiplications and single-value multiplications, where each scalar multiplication is mapped to one DSP48E2 slice. For efficient matrix multiplication in HLS, the first operand matrix is partitioned in a row-wise manner, while the second operand matrix is partitioned column-wise using array partition pragmas. This memory organization enables all elements of a row from the first matrix and all elements of a corresponding column from the second matrix to be accessed simultaneously. Consequently, the MAC operations required to compute each output element can be executed in parallel. With appropriate loop unrolling and pipelining, this approach significantly improves throughput by allowing concurrent MAC operations for each output element. The number of DSPs required for the matrix multiplication is therefore equal to the number of multiplications needed to compute a single element of the output matrix. For example, multiplying two matrices with size 4×4 requires only four DSP slices, and multiplying matrices with sizes 1×3 and 3×4 requires only three DSP slices. Therefore, the total number of DSPs used in Design Approach-3 with fixed-point implementation is equal to 27, as shown in Table 5. The LUTs and FFs are used in almost all operations of the algorithm. All the additions and subtractions are implemented using LUTs and FFs.

In addition to resource utilization and execution time, the power consumption of the DoA IP is evaluated for different design approaches, as summarized in Table 3. Design Approach-1 consumes 0.585 W, whereas Design Approach-2 reduces the power consumption to 0.185 W, achieving a 68.38% reduction due to improved resource utilization based on the UTM approach. Design Approach-3 (FIP) further enhances energy efficiency, consuming only 0.059 W, which corresponds to an 89.91% reduction in power compared to Design Approach-1. Overall, the PL-based implementation using Design Approach-3 (FIP) achieves superior resource

TABLE 8. Resource utilization for different sizes of ULA.

Resource Type	Total Available Resources	1×4 ULA		1×8 ULA		1×16 ULA		1×32 ULA	
		Resources Utilized	Resources Utilized (%)	Resources Utilized	Resources Utilized (%)	Resources Utilized	Resources Utilized (%)	Resources Utilized	Resources Utilized (%)
BRAM	1080	8	0.74	12	1.11	20	1.85	36	3.33
DSP	4272	27	0.63	47	1.10	87	2.04	167	3.91
FF	850560	5536	0.65	8714	1.02	11037	1.30	31030	3.65
LUT	425280	5920	1.39	7053	1.66	9745	2.29	20519	4.82
Latency (ms)		0.063		0.553		1.715		6.433	

utilization, lower latency, and reduced power consumption, demonstrating that the proposed PL–PS co-design framework is suitable for energy-efficient, software-defined 5G/6G wireless applications.

The estimation of DoA for different source location angles is shown in Table 6. It can be observed that the DoA values estimated in real time using the DoA IP closely match the estimated DoA values obtained from MATLAB simulations. The fixed point with (20,4) implementation also estimates an accurate DoA compared to the floating point implementation. Therefore, the DoA IP based on Design Approach-3 with FIP design utilized fewer resources and also estimated the correct DoA.

E. SENSITIVITY ANALYSIS OF FIXED-POINT WORD LENGTH

The sensitivity analysis is performed by varying the fractional bit-width of the fixed-point representation. This analysis evaluates the impact of fractional precision on DoA estimation accuracy and hardware resource utilization. To analyze the trade-offs systematically, the fractional bits from 6 to 22 are varied, while fixing the integer bit-width (including sign bit) at 4 bits. As shown in Table 7, the word length (20,4) offers an excellent trade-off, achieving high accuracy with minimal resource usage. Greater than (20,4) and floating-point data yield almost the same accuracy with higher resource utilization. The integer width (including sign bit) of 4 was found to be sufficient for normalized covariance matrices, and this choice was made after extensive empirical validation. Formats with fewer than 4 integer bits exhibited higher estimation errors due to saturation effects, whereas wider integer widths did not yield noticeable accuracy gains.

F. SCALABILITY ANALYSIS

The proposed architecture is implemented and validated using RFSoC ZCU216 for 1×4 ULA. However, it can be extended to a large array size as well. For the scalability analysis purpose, the authors have considered different array sizes and implemented DoA IP in the RFSoC ZCU216 without performing hardware measurements. Table 8 summarizes the resource utilization of the DoA estimation IP for different array sizes. As expected, resource utilization increases with the number of array elements M. However, even for a 1×32 ULA, less than 4% of the available BRAMs, DSP slices, and FFs are utilized. The LUT utilization is below 5%, indicating substantial headroom for further scaling. The LUT usage can be further reduced by mapping selected addition and subtraction operations to DSP slices, thereby improving resource balance.

To quantify scalability limits, the number of DSP slices required for a $1 \times M$ ULA is expressed as

$$\text{DSPs} = 5M + 7 \tag{25}$$

Based on the total DSP resources available on the RFSoC ZCU216 platform, the architecture theoretically scales up to $M = 853$ before DSP saturation occurs.

BRAM utilization is primarily determined by the memory requirements of the steering matrix, covariance matrix, eigenvector matrix, and angle vector, with dimensions $M \times N$, $M \times M$, $M \times M$, and $1 \times N$, respectively. The total number of required 36-Kb BRAMs is given by:

$$\frac{MN \times 20}{36864} + \frac{M^2 \times 20}{36864} + \frac{(M^2 + M) \times 20}{36864} + \frac{N \times 20}{36864} \tag{26}$$

where $N = 1801$ corresponds to a step size of 0.1° and the word length is 20 bits. In addition, one BRAM is allocated for storing the $\cos(\theta)$ and $\sin(\theta)$ values using a CORDIC lookup table. Based on the available BRAM resources on the RFSoC device, the architecture can scale up to approximately $M \approx 643$ antenna elements before BRAM capacity becomes the limiting factor.

Overall, the scalability analysis demonstrates that the proposed architecture efficiently supports large antenna arrays with modest hardware resource utilization. Therefore, the proposed architecture can be extended to a large array size ULA for massive MIMO scenarios targeted by 5G and beyond.

G. ADAPTIVE BEAMFORMING RESULTS

The adaptive beamforming performance of the proposed PL–PS co-design framework is evaluated using a 1×4 ULA. A 5G NR signal is captured from different directions in an anechoic chamber environment, and the received data from the ADCs are transferred to the PS. The number of samples received from each ADC is 4096, and each sample contains both real and imaginary data.

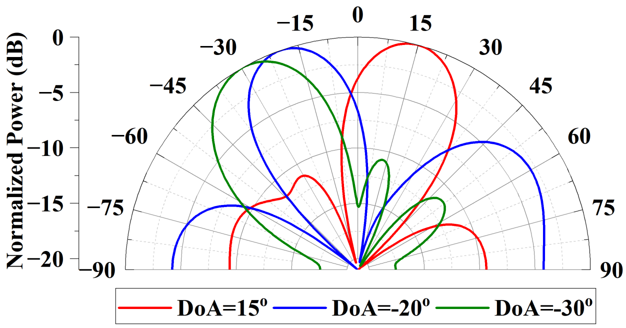

FIGURE 13. Measured radiation pattern of adaptive digital beamforming at different DoAs.

TABLE 9. Execution time for operations implemented in PS.

Operations	Execution Time (μs)
CVCM	235.8
UTM	8.5
Weights	36.03
Beamforming output for 4096 samples	998.39
Total	1278.72

TABLE 10. Comparison of PL and PS execution time for EVD and SPS.

Operations	PS (ms)	PL (ms)	Reduction (%)
EVD	133.4	23.26	82.31
SPS	230.06	34.32	85.08
Total Execution Time	363.46	57.92	84.06

TABLE 11. Performance of adaptive beamforming using MVDR algorithm.

Desired Angle	Estimated beam direction	HPBW	PSLL (dB)
-30 °	-29.8 °	30.35 °	-9.35
-20 °	-19.5 °	31.6 °	-13.4
15 °	15.1 °	27.45 °	-8.3

The adaptive beamforming using the MVDR algorithm is implemented in PS based on the estimated DoA from PL. Table 9 summarizes the execution time of various operations implemented in the PS. CVCM is computed in the PS using 100 samples per antenna element. The UTM converts the complex-valued covariance matrix into a real-valued matrix, enabling subsequent real-valued EVD and SPS processing in the PL and thereby eliminating the need for complex arithmetic in hardware. The UTM operation involves only simple add/subtract operations and a few multiplications by constants, because most entries of the unitary matrix consist of 0, ±1, or $\pm j$. As a result, the overhead introduced by the UTM step in the PS is extremely small and requires only 8.5 μs. The beamforming weights computation also requires less execution time of 36.03 μs because the inverse of the covariance matrix is calculated by utilizing eigenvalues and eigenvectors from PL. The beamforming output computation time for 4096 samples is 998.39 μs. This corresponds to an effective beamforming output computation latency of 0.243 μs per sample. The total execution time for all these operations in the PS is 1278.72 μs.

The EVD and SPS blocks were implemented in the PL and operate at a clock frequency of 100 MHz. Table 10 presents a direct comparison of PL and PS execution time. Implementing EVD and SPS in the PL results in a 84.06% reduction in execution time compared to the PS implementation. This clearly demonstrates that the PL significantly accelerates the computationally intensive stages of the MUSIC and MVDR algorithms. The total execution time required to compute the adaptive beamforming with DoA estimation using the proposed PL and PS co-design architecture is 1.34 ms.

The radiation pattern of the output signal after beamforming is shown in Fig. 13, which shows that the beam is formed in the transmitted signal direction. The measured half-power beamwidth (HPBW) values are close to the theoretical ideal value of approximately 25.5 °for a 1 × 4 ULA, indicating accurate beamwidth formation. Table 11 summarizes the quantitative beamforming performance for different desired steering angles. As observed, the estimated beam directions closely match the desired angles, while the measured PSLL values confirm effective sidelobe suppression.

VII. CONCLUSION

The adaptive digital beamforming is implemented in the RFSoC ZCU216 evaluation board based on the PL and PS co-design approach for complex-valued 5G NR signals. The computational complexity is reduced by converting the complex-valued covariance matrix to a real-valued matrix based on UTM. Then, the eigenvalues and eigenvectors were calculated using the Jacobi iteration method in PL of RFSoC, which accelerated the execution. The DoA IP is designed in PL, while the MVDR algorithm is designed in PS by utilizing eigenvalues and eigenvectors calculated through the DoA IP, which gave optimum resource utilization and execution time. The proposed PL and PS co-design of the ADBF algorithm with DoA is verified with measurements in the anechoic chamber using a 1 × 4 antenna array, which has formed the beam in the desired direction.

REFERENCES

[1] S. Kutty and D. Sen, "Beamforming for millimeter wave communications: An inclusive survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 949–973, 2nd Quart., 2016.

[2] I. P. Gravas, Z. D. Zaharis, T. V. Yioultsis, P. I. Lazaridis, and T. D. Xenos, "Adaptive beamforming with sidelobe suppression by placing extra radiation pattern nulls," *IEEE Trans. Antennas Propag.*, vol. 67, no. 6, pp. 3853–3862, Jun. 2019.

[3] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Trans. Antennas Propag.*, vol. AP-34, no. 3, pp. 276–280, Mar. 1986.

[4] J. Teich, "Hardware/software codesign: The past, the present, and predicting the future," *Proc. IEEE*, vol. 100, no. 5, pp. 1411–1430, May 2012.

[5] C. Codau, B. Rares, T. Palade, A. Pastrav, P. Dolea, R. Simedroni, and E. Puschita, "Experimental evaluation of a beamforming-capable system using NI USRP software defined radios," in *Proc. 18th RoEduNet Conf., Netw. Educ. Res. (RoEduNet)*, Galati, Romania, Oct. 2019, pp. 1–6.

[6] V. Goverdovsky, D. C. Yates, M. Willerton, C. Papavassiliou, and E. Yeatman, "Modular software-defined radio testbed for rapid prototyping of localization algorithms," *IEEE Trans. Instrum. Meas.*, vol. 65, no. 7, pp. 1577–1584, Jul. 2016.

[7] N. R. Dusari and M. Rawat, "Multi tile synchronization and calibration of Xilinx RF SoC ZCU216 for digital beamforming," in *Proc. IEEE Wireless Antenna Microw. Symp. (WAMS)*, Rourkela, India, Jun. 2022, pp. 1–5.

[8] D. Marinho, R. Arruela, T. Varum, and J. N. Matos, "Application of digital beamforming to software defined radio 5G/Radar systems," in *IEEE MTT-S Int. Microw. Symp. Dig.*, Nov. 2019, pp. 1–3.

[9] N. R. Dusari and M. Rawat, "Efficient implementation of direction of arrival using RFSoC platform," in *Proc. IEEE Microw., Antennas, Propag. Conf. (MAPCON)*, Ahmedabad, India, Dec. 2023, pp. 1–6.

[10] V. Dakulagi and J. He, "Improved direction-of-arrival estimation and its implementation for modified symmetric sensor array," *IEEE Sensors J.*, vol. 21, no. 4, pp. 5213–5220, Feb. 2021.

[11] L. Zhang, D. Veerendra, M. Nagabushanam, B. R. Babu, and A. Singh, "Efficient direction estimation of both coherent and uncorrelated sources without prior knowledge of source count," *IEEE Sensors J.*, vol. 23, no. 18, pp. 21739–21746, Sep. 2023.

[12] A. A. Hussain, N. Tayem, M. O. Butt, A.-H. Soliman, A. Alhamed, and S. Alshebeili, "FPGA hardware implementation of DOA estimation algorithm employing LU decomposition," *IEEE Access*, vol. 6, pp. 17666–17680, 2018.

[13] A. A. Hussain, N. Tayem, A.-H. Soliman, and R. M. Radaydeh, "FPGA-based hardware implementation of computationally efficient multi-source DOA estimation algorithms," *IEEE Access*, vol. 7, pp. 88845–88858, 2019.

[14] U. M. Butt, S. A. Khan, A. Ullah, A. Khaliq, P. Reviriego, and A. Zahir, "Towards low latency and resource-efficient FPGA implementations of the MUSIC algorithm for direction of arrival estimation," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 68, no. 8, pp. 3351–3362, Aug. 2021.

[15] M. Gupta, S. Sharma, H. Joshi, and S. J. Darak, "Reconfigurable architecture for spatial sensing in wideband radio front-end," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 69, no. 3, pp. 1054–1058, Mar. 2022.

[16] J. He, T. Shu, V. Dakulagi, and L. Li, "Simultaneous interference localization and array calibration for robust adaptive beamforming with partly calibrated arrays," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 57, no. 5, pp. 2850–2863, Oct. 2021.

[17] H. Xu, H. Aliakbarian, and G. A. E. Vandenbosch, "Off-the-shelf low-cost target tracking architecture for wireless communications," *IEEE Syst. J.*, vol. 9, no. 1, pp. 13–21, Mar. 2015.

[18] H. Xu, H. Aliakbarian, E. Van der Westhuizen, R. Wolhuter, and G. A. E. Vandenbosch, "An architectural scheme for real-time multiple users beam tracking systems," *IEEE Syst. J.*, vol. 11, no. 4, pp. 2905–2916, Dec. 2017.

[19] M. Fosberry and M. Livadaru, "Digital synthetic receive beamforming with the Xilinx ZC1275 evaluation board," in *Proc. IEEE Int. Symp. Phased Array Syst. Technol. (PAST)*, Waltham, MA, USA, Oct. 2019, pp. 1–2.

[20] S. Pulipati, V. Ariyarathna, U. D. Silva, N. Akram, E. Alwan, A. Madanayake, S. Mandal, and T. S. Rappaport, "A direct-conversion digital beamforming array receiver with 800 MHz channel bandwidth at 28 GHz using Xilinx RF SoC," in *Proc. IEEE Int. Conf. Microw., Antennas, Commun. Electron. Syst. (COMCAS)*, Nov. 2019, pp. 1–5.

[21] R. Fagan, F. C. Robey, and L. Miller, "Phased array radar cost reduction through the use of commercial RF systems on a chip," in *Proc. IEEE Radar Conf.*, Apr. 2018, pp. 0935–0939.

[22] E. Dehn, *Algebraic Equations: An Introduction To the Theories of Lagrange and Galois*. New York, NY, USA: Columbia Univ. Press, 1930.

[23] A. Collins, "All programmable RF-sampling solutions," Xilinx, Inc., San Jose, CA, USA, Tech. Rep. WP489 (v1.0.1), 2017, vol. 489.

[24] Y. Liu, C.-S. Bouganis, P. Y. K. Cheung, P. H. W. Leong, and S. J. Motley, "Hardware efficient architectures for eigenvalue computation," in *Proc. Design Autom. Test Eur. Conf.*, Munich, Germany, 2006, pp. 1–6.

[25] Z. Li, W. Wang, R. Jiang, S. Ren, X. Wang, and C. Xue, "Hardware acceleration of MUSIC algorithm for sparse arrays and uniform linear arrays," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 69, no. 7, pp. 2941–2954, Jul. 2022.

[26] K.-C. Huarng and C.-C. Yeh, "A unitary transformation method for angle-of-arrival estimation," *IEEE Trans. Signal Process.*, vol. 39, no. 4, pp. 975–977, Apr. 1991.

[27] Q. Chen and R. Liu, "On the explanation of spatial smoothing in MUSIC algorithm for coherent sources," in *Proc. Int. Conf. Inf. Sci. Technol.*, Nanjing, China, Mar. 2011, pp. 699–702.

[28] D. Astely and B. Ottersten, "The effects of local scattering on direction of arrival estimation with MUSIC," *IEEE Trans. Signal Process.*, vol. 47, no. 12, pp. 3220–3234, Dec. 1999.

[29] Y. S. Cho, J. Kim, W. Y. Yang, and C. G. Kang, *MIMO-OFDM Wireless Communications With MATLAB*. Hoboken, NJ, USA: Wiley, 2010.

[30] AMD. (2023). *Zynq UltraScale+ RFSoC RF Data Converter V2.6 Gen 1/2/3/DFE LogiCORE IP Product Guide (PG269)*. Accessed: Dec. 2024. [Online]. Available: https://docs.xilinx.com/r/en-U.S./pg269-RF-data-converter/Introduction

[31] J. Cong, B. Liu, S. Neuendorffer, J. Noguera, K. Vissers, and Z. Zhang, "High-level synthesis for FPGAs: From prototyping to deployment," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 30, no. 4, pp. 473–491, Apr. 2011.

[32] M. Imtiaz Rashid and B. Carrion Schafer, "Robust and efficient RTL to c compiler optimized for high-level synthesis," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 44, no. 2, pp. 559–567, Feb. 2025.

NAGESWARA RAO DUSARI (Graduate Student Member, IEEE) received the B.Tech. degree in electronics and communication engineering from Acharya Nagarjuna University (ANU), India, in 2011, and the M.Tech. degree from the Motilal Nehru National Institute of Technology (MNNIT), Allahabad, India, in 2013, with a specialization in digital systems. He is currently pursuing the Ph.D. degree with the Department of Electronics and Communication Engineering, IIT Roorkee, India. His research interests include software-defined radios, digital beamforming, direction of arrival, digital predistortion, and hardware and software co-design. His research has resulted in several publications in reputed IEEE journals and conferences.

MEENAKSHI RAWAT (Member, IEEE) received the Ph.D. degree in electrical and computer engineering from the University of Calgary, Calgary, AB, Canada, in 2012.

She was a Postdoctoral Research Fellow with the University of Calgary, from September 2012 to June 2013. She was also with The Ohio State University as a Postdoctoral Project Researcher/Scientist, from July 2013 to June 2014. She is currently a Professor with Indian Institute of Technology (IIT) Roorkee, Uttarakhand, India. She is one of the founding directors of the IITR-led startup "Linearized Amplifier Technologies and Services Private Ltd." Her research interests include software-defined radio (SDR), digital and analog predistortion for radio-frequency power amplifiers, multiband and ultra-wideband transmitter architectures, and implementation aspects of MIMO and beamforming. She has developed several single-band, multi-band, and ultra-wideband solutions for distortion-free transmitter front ends, resulting in multiple industrial projects, patents, and numerous publications in IEEE Transactions and international conferences. She has received the University of Calgary Research Production Award on multiple occasions and the Best Paper Award at the 82nd and 83rd Automatic RF Techniques Group (ARFTG) Conferences, in 2013 and 2014. She also received the Young Faculty Research Fellowship from the Digital India Laboratories.