ABSTRACT This paper presents a low complexity chip design supporting dual-mode beamforming, i.e., sampling matrix inversion (SMI) and the minimum variance distortionless response (MVDR), for wireless Multiple-Input Multiple-Output (MIMO) communication systems. The auto-correlation matrix inversion is the critical computing kernel shared by the two beamforming schemes. To alleviate the computing complexity, the auto-correlation matrix is approximated by a Toeplitz counterpart, which can be decomposed efficiently by applying the Cholesky decomposition and the Schur algorithm. This leads to an $O(N^3)$ to $O(N^2)$ complexity reduction, where $N$ is the matrix size, while preserving computing parallelism for the hardware design. In addition, a diagonal loading technique is employed to mitigate the stability problem when the matrix is ill-conditioned. Simulation results indicate that no performance loss is observed due to the algorithm simplification measures. A systolic array based mapping procedure converts the two beamforming algorithms to a unified hardware accelerator design with 80% shared circuitry. Complex-valued divisions are achieved by adopting a hardware efficient coordinate rotation digital computer (CORDIC) scheme. In chip implementation, a TSMC 90nm UTM process technology is used and the design specs largely follow the requirements of IEEE 802.11ac standard. The core size of the chip design is $0.68 \text{mm}^2$. The measurement results show that the chip can operate up to 200MHz with a power consumption of 49.03mW. It can complete the computation of a new beamforming vector (of size 8) every 0.64us and exhibits the highest throughput among the 6 compared designs.

INDEX TERMS Adaptive beamforming, Cholesky decomposition, Schur algorithm, systolic array, hardware accelerator, chip design.

I. INTRODUCTION

Beamforming, also known as spatial filtering, is an essential technique of array signal processing and was traditionally applied to radar systems to achieve the directional transmission/receiving of signals. With the prevailing of wireless communication systems, beamforming schemes have also been employed extensively to enhance the transmission performance, such as extending the service range or improving SNR values at the receiving end. Beamforming controls the phase and relative amplitude of the signal at each transmitter (or receiver) for directional signal transmission (or reception). For wireless communication, this feature is useful in combating path loss problems and supporting spatial multiplexing. For mmWave based wireless communication systems such as 5G, the path and penetration loss problem is even more prominent and the beamforming technique is a must [1]. The directionality of beamforming depends on the size of antenna array. The larger the array size, the better the directionality. The implementation of beamforming can be achieved in the RF section or in the baseband (digital) section. The former one employs a phase shifter for each antenna and requires an additional signal combiner at the receiving end [2]–[4]. The accuracy of phase shift / signal combination and the incurred signal attenuation, which negatively affecting the overall RF link budget, are the major hurdles of this approach. The digital approach [5], although requires extra RF chains (one per antenna) and ADC/DACs,
A. ADAPTIVE BEAMFORMING

Instead of receiving signal along a fixed direction, adaptive beamforming can steer the beam in a specific direction according to the target source position, and can dynamically adjust the beam weight as the signal source moves. By changing of the null direction of receiving, it can also suppress possible interferences to enhance the received signal quality. Various beamforming weight calculation algorithms have been developed subject to different optimization criteria. These criteria include minimum mean square error (MMSE), minimum output energy (MOE), and maximum output signal to interference plus noise ratio (Max SINR). The MMSE algorithm tries to minimize the error with respect a reference signal $d(t)$. The autocorrelation matrix $R_{xx}$ of the received signals $x(t)$ and the cross correlation vector $r_{xd}$ between $x(t)$ and $d(t)$ are first calculated. The solution

$$W_{MMSE} = R_{xx}^{-1} r_{xd}$$

is commonly known as the Wiener filter. The MOE algorithm minimizes the total output energy while keeping the gain of the array of desired signal constant. This algorithm does not require a reference signal as in the case of MMSE, but needs the knowledge of spatial signature (or steering vector) $h_0$. In line of sight communications, this corresponds to the direction of arrival (DoA). The weight vector can be calculated as

$$W_{MOE} = c \frac{R_{xx}^{-1} h_0}{h_0^H R_{xx}^{-1} h_0}.$$  

(2)

When the constant $c$ is set to 1, it leads to a Minimum Variance Distortionless Response (MVDR) beamformer. The DoA estimation can be obtained by using schemes such as Multiple Signal Classification (MUSIC) and Estimation of Signal Parameters via Rotational Invariance Technique (ESPRIT) algorithms. The Max SINR algorithm aims at maximizing the signal to interference plus noise ratio. It also requires the spatial signature information. The optimal weight vector of the max SINR algorithm is similar to that of the MOE scheme by replacing the signal autocorrelation matrix $R_{xx}$ with noise covariance matrix $R_n$.

$$W_{MaxSINR} = c \frac{R_{xx}^{-1} h_0}{h_0^H R_n^{-1} h_0}.$$  

(3)

It can be shown, these three beamforming vectors, although being derived by using different optimal criteria, are actually equivalent.

In implementing practices, the autocorrelation matrix $R_{xx}$ and cross correlation vector $r_{xd}$ can be estimated by taking average of multiple samples of received data. The inverse of the autocorrelation matrix estimate ($\hat{R}_{xx}^{-1}$) is then calculated to obtain the optimal beamforming weight vector. This is considered the direct matrix inversion (DMI) or sample matrix inversion (SMI) approach. Note that the computing complexity of matrix inversion is $O(N^3)$, where $N$ is matrix size. To alleviate the computing overheads, iterative algorithms based on the principle of steepest descent were developed. These include least mean square (LMS) and recursive least square (RLS) algorithms. In LMS algorithm, autocorrelation matrix and cross-correlation vector are replaced by their snapshot values. The convergence, however, takes a long time and the selection of step size $\mu$ is critical. The RLS algorithm is considered to achieve a faster convergence than the LMS algorithm. It computes the matrix inversion incrementally in each iteration and the total computing complexity is actually higher than the DMI approach. A RLS digital beamformer is presented in [6], it uses modified inverse matrix operations to reduce computational complexity. A SMI-MVDR beamformer was proposed in [7], it changes the dimensions of matrix operations and implements matrix decomposition using recursive updates.

As for the implementation platforms, due to intensive computing efforts required for adaptive beamforming, either high end programmable DSP processors or hardwired solutions are plausible for real time operations. The former one is flexible in implementing different algorithms. For example, a Steered Minimum Variance (STMV) adaptive beamforming was proposed in [8] and implemented on the TigerSHARC Digital Signal Processor platform. The programmable option, however, consumes much more power than the hardwired alternative. It is also less efficient in achieving high throughput computations. An ASIC design using QR decomposition to realize adaptive beamforming is proposed [9]. The ASIC is implemented by using a TSMC 0.13nm process technology and can operate up to 150MHz. In [10], a Real-Time QRD-Based MVDR Beamforming is implemented in a Xilinx Virtex-4 platform and can operate up to 250MHz.

B. PROPOSED WORK AND CONTRIBUTIONS

In this paper, a low-complexity dual-mode adaptive digital beamformer design for packet based wireless communication
systems is investigated. In particular, the prevailing MIMO OFDM system is considered as the target. The beamforming technique can be applied to either the time domain (pre-FFT) or the frequency domain (post-FFT) in a MIMO OFDM system. Taking the receiving end as an example, the time domain approaches [11]–[14] combine the baseband signals received from the antenna array by multiplying them with a set of beamforming weights. The frequency domain approaches [15]–[18] performs the beamforming on a per subcarrier basis after transforming the FFT transform and each sub-carrier can have a distinct beamforming weight. This, however, calls for much higher computing efforts. In this paper, the time domain approach is used with a potential application to hybrid beamforming schemes for massive MIMO systems.

Instead of adopting iterative algorithms with uncertainty in convergence speed, the DMI schemes are employed. The beamforming vector computations are designed to be accomplished within the fixed time frame of packet preamble. In particular, two popular DMI schemes are supported, i.e. MMSE and MVDR. The former one performs the MMSE beamforming using the preamble of packet data as the reference signal while the latter one achieves the beamforming based on given DoA information. The computation of the autocorrelation matrix inverse is the most computationally intensive part in both schemes. A new computing scheme is developed to alleviate its computing complexity. Instead of direct matrix inversion, Cholesky decomposition of the auto-correlation matrix is performed first, which decomposes the matrix into the product of a lower and an upper triangular matrix. We further take advantages of the near Toeplitz structure property of auto-correlation matrix and apply a fast computing Schur algorithm to obtain the decomposition result. This measure successfully reduces the computing complexity by an order, i.e. from $O(N^3)$ to $O(N^2)$, where $N$ is the matrix size. After matrix decomposition, the matrix inversion is converted to two successive matrix vector multiplications, which also facilitates efficient pipelining in hardware implementation. In hardware accelerator design, we use the framework of 802.11ac as a reference in determining timing and throughput specs. A multi-stage systolic array design is developed. It serves as the shared computing kernel of two beamforming modules and reduces the overall circuit complexity by 41%. Various circuit simplification and power reduction techniques are applied. The design is implemented in a 90nm process technology and measured results are provided.

The remaining of the paper is organized as follows: In Section II, the proposed auto-correlation matrix inversion computing scheme is presented. This includes the Schur algorithm based Cholesky decomposition and extra algorithm modifications to improve hardware design efficiency. In section III, the performance evaluations, as well as the bit true simulations, of the two adaptive beamforming schemes are elaborated. In addition, a diagonal loading technique to tackle matrix rank deficiency problem is also discussed. Section IV addresses the algorithm to hardware mapping issue and discusses various optimization techniques employed to obtain a unified dual-mode beamformer design. Finally, the chip design summary and implementation results are given in section V. A performance comparison with existing works is included. Possible extension to a more general MIMO configurations is also discussed.

II. MMSE/MVDR DUAL-MODE BEAMFORMING COMPUTING SCHEME

A. SCHUR ALGORITHM FOR MATRIX DECOMPOSITION

Refer to Eqs (1) and (2), auto-correlation matrix inversion, $\mathbf{R}_{xx}^{-1}$, is common to the weight vector computations of MMSE and MVDR beamforming scheme. Since direct matrix inversion is too costly in computing complexity, it is seldom computed explicitly. If the matrix inversion is meant for solving a linear system $\mathbf{A} \cdot \mathbf{b} = \mathbf{c}$ with the solution $\mathbf{b} = \mathbf{A}^{-1} \cdot \mathbf{c}$, by decomposing the matrix $\mathbf{A}$ into the product of a lower triangular matrix $\mathbf{L}$ and an upper triangular matrix $\mathbf{U}$ (LU decomposition), i.e., $\mathbf{A} = \mathbf{L} \cdot \mathbf{U}$, the solution vector $\mathbf{b}$ can be computed by solving two equations successively.

$$\mathbf{L} \cdot \mathbf{d} = \mathbf{c}$$
$$\mathbf{U} \cdot \mathbf{b} = \mathbf{d}$$

Because of the special structures of $\mathbf{L}$ and $\mathbf{U}$ matrices, vector $\mathbf{d}$ and then vector $\mathbf{b}$ can be calculated by using forward substitution and backward substitution schemes, respectively. If the matrix is Hermitian and positive definite, e.g., an auto-correlation matrix with full rank, the LU decomposition can be converted to a LDL decomposition, a variant of the classical Cholesky decomposition.

$$\mathbf{R}_{xx} = \mathbf{L} \cdot \mathbf{D} \cdot \mathbf{L}^H,$$  (5)

where $\mathbf{L}$ is a unit lower triangular matrix, $\mathbf{D}$ is a diagonal matrix, and the superscript $H$ indicates the conjugate transpose. The algorithm computing complexity of LDL decomposition is roughly one half that of regular LU decomposition [19]. In spite of this, the algorithm complexity is still of $O(N^3)$, where $N$ is the matrix size, and poses a big challenge to real time computation.

To further reduce the computing complexity, we take advantage of the special property of the signal vectors received from the antenna array. Consider a uniform linear array (ULA) of antennas, assume there are $l$ signal sources each arrives from a distinctive direction. Define the corresponding steering vectors (or array response vector) $\mathbf{a}(\theta_i), i = 1 \sim l$, as

$$\mathbf{a}(\theta_i) = [1, e^{1j2\pi \sin(\theta_1)}, \ldots, e^{(N-1)j2\pi \sin(\theta_l)}]^T$$  (6)

The autocorrelation matrix $\mathbf{R}_{xx}$ of the received signal vector $\mathbf{x}(t)$ can be expressed as

$$\mathbf{R}_{xx} = \mathbf{A} \mathbf{R}_{ss} \mathbf{A}^H + \mathbf{R}_{\text{bry}},$$  (7)

where $\mathbf{R}_{ss}$ is the auto-correlation matrix (l by l in dimension) of the l signal sources multiplied by the respective path attenuation factor, and $\mathbf{A} = [\mathbf{a}(\theta_1) \cdot \ldots \cdot \mathbf{a}(\theta_l)]_{N \times l}$. $\mathbf{R}_{\text{bry}}$ is the covariance matrix of the noise vector $\mathbf{n}(t)$ observed at the receiving
antenna array. If \( f \) signal sources and the channel model are both stationary, \( \mathbf{R}_{xx} \) becomes a Toeplitz one, so does \( \mathbf{A} \mathbf{R}_{xx} \mathbf{A}^H \). If all noises are AWGN (additive white Gaussian Noise) and mutually independent, \( \mathbf{R}_{nn} = \sigma_n^2 \mathbf{I} \). This suggests \( \mathbf{R}_{xx} \) is also a Toeplitz one. A Toeplitz matrix (or diagonal-constant matrix) has each of its descending diagonals from left to right as constant. Eq. (8) shows a \( 4 \times 4 \) matrix example of \( \mathbf{R}_{xx} \), which is both Toeplitz and Hermitian.

\[
\mathbf{R}_{xx} = \begin{bmatrix}
t_0 & t_1 & t_2 & t_3 \\
t_1^* & t_0 & t_1 & t_2 \\
t_2^* & t_1^* & t_0 & t_1 \\
t_3^* & t_2^* & t_1^* & t_0 \\
\end{bmatrix},
\]

(8)

Note that the diagonal values of the matrix are real-valued and identical. Due to the special matrix structure of Toeplitz matrices, fast decomposition algorithms with a computing complexity of \( O(N^2) \) can be applied. There are basically two types of fast algorithms, i.e. Levinson type and Schur type. The classical Levinson-Durbin recursion was first proposed by Norman Levinson in [20] and later improved by James Durbin in [21]. It constructs the decomposition of the inverse of the Toeplitz matrix by solving the last columns of the inverses of the leading principal submatrices recursively. A typical Schur type algorithm was presented in [22] as fast Cholesky factorization for positive definite Toeplitz matrices. The computations proceed progressively in arrow-wise manner to obtain the decomposition of the Toeplitz matrix. Both types of algorithms have similar computing complexities. More specifically, Schur algorithm uses \( O(2N^2 - N\kappa) \) time complexity, whereas Levinson-Durbin recursion uses only \( O(N^2) \) space, which suggests a smaller memory footprint of the latter in software implementation. Schur algorithm, however, is numerically stable but Levinson-Durbin algorithm is not for some ill-conditioned matrix. From the aspect of hardware implementation, the computing parallelism of the algorithm is essential to the efficiency of hardware acceleration. Both Schur and Levinson-Durbin algorithms are recursive. Full computing parallelism (with an \( O(1) \) time complexity) can be exploited from the vector addition and scalar times vector operations in Schur algorithm. However, only partial computing parallelism (\( O(\log N) \) time complexity) can be extracted from the vector inner product operations in Levinson-Durbin algorithm. Based on these two reasons (numerical stability and time complexity), Schur algorithm is employed as the basic framework of our auto-correlation matrix inversion scheme. Figure 2 shows the pseudo code of the Schur algorithm, which triangularizes a Toeplitz and Hermitian matrix \( \mathbf{R}_{xx} \) as

\[
\mathbf{L} \cdot \mathbf{R}_{xx} = \mathbf{U},
\]

(9)

where \( \mathbf{L} \) is a unit lower triangular matrix and \( \mathbf{U} \) is an upper triangular matrix. The elements of matrix \( \mathbf{R}_{xx} \) are \( r_{ij} = r_{ij}^* = t_{i-j}, \) for \( i, j = 1 \sim N \) and \( i \geq j \).

The scheme performs matrix triangularization by calculating the rows of \( \mathbf{L} \) and \( \mathbf{U} \) matrices progressively (one row per iteration) from top to bottom. Calculating the first rows of \( \mathbf{L} \) and \( \mathbf{U} \) matrices (iteration 1) is trivial because \( \mathbf{L}_1 = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}^T \) and \( \mathbf{u}_1^T \) is identical to the first row of \( \mathbf{R}_{xx} \). Starting from iteration 2, two row vectors \( \mathbf{v}_l \) and \( \mathbf{u}_l \) are updated based on the results of \( \mathbf{v}_{l-1} \) and \( \mathbf{u}_{l-1} \) from the previous iteration. The initial setting of \( \mathbf{v}_1 \) and \( \mathbf{u}_1 \) are equal to the first two rows of \( \mathbf{R}_{xx} \), respectively. The \( \mathbf{u} \) vector is first shifted right by one position with a new entry added to the leading position. A reflection coefficient \( \kappa_l \) is then calculated and the \( \mathbf{v}_l \) and \( \mathbf{u}_l \) vectors are updated by performing a butterfly operation, which consists of two elementary row operations using \( \kappa_l \) as the multiplication constant. The leading entry of the \( \mathbf{u}_l \) vector will be nullified and becomes the \( i \)th row of the \( \mathbf{U} \) matrix.

**Algorithm 1:** Schur decomposition

**INPUT:** \( \mathbf{t} \) /* the first row of the Toeplitz matrix */

**process the 1st row of \( \mathbf{L} \) and \( \mathbf{U} \) matrices**

01. For \( j = 1 \) to \( N \)
02. \( \mathbf{v}_j = \mathbf{u}_j = t_{r_{-1}} \)
03. \( \mathbf{l}_{1,j} = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix} \) /* reflection coefficient */

**Element-wise operations of row vectors**

04. For \( i = 2 \) to \( N \)
05. \( \kappa_i = -v_{i-1}(i) / u_{i-1}(i-1) \) /* reflection coefficient */

**process the 2nd to the \( N \)th row of \( \mathbf{L} \) and \( \mathbf{U} \) matrices**

06. For \( j = i+1 \) to \( N \)
07. \( \mathbf{v}_j = \kappa_i \mathbf{u}_{i-1}(j-1) \) /* row operation of \( \mathbf{V} \) matrix */
08. \( \mathbf{v}_j = \mathbf{v}_j + \mathbf{v}_{i-1} \)
09. End
10. For \( j = i \) to \( N \)
11. \( \mathbf{u}_j = \mathbf{u}_{i-1}(j-1) + \mathbf{v}_i \) /* row operation of \( \mathbf{U} \) matrix */
12. \( \mathbf{u}_j = \mathbf{u}_j \)
13. End
14. \( \mathbf{l}_{1}(i) = 1 \)
15. \( \mathbf{l}_{1}(i) = \mathbf{v}_i \)
16. For \( j = 2 \) to \( i-1 \)
17. \( \mathbf{l}_{i-1,j} = \mathbf{v}_i \mathbf{u}_{i-1}(i-j) \) /* row operation of \( \mathbf{L} \) matrix */
18. \( \mathbf{l}_{1,j} = \mathbf{v}_i \mathbf{u}_{i-1}(i-j) + \mathbf{v}_{i-1} \)
19. End
20. End

**OUTPUT:** \( \mathbf{L} \) /* the lower triangular matrix */

**OUTPUT:** \( \mathbf{U} \) /* the upper triangular matrix */

**FIGURE 2.** The Schur scheme for matrix triangularization.

**B. EFFICIENT AUTO-CORRELATION MATRIX INVERSION SCHEME**

Refer to Eq. (5), after Schur decomposition we have

\[
\mathbf{L} = \mathbf{L}^{-1}, \quad \mathbf{U} = \mathbf{D} \cdot \mathbf{L}^H \quad (10)
\]

Our goal is calculating the product of \( \mathbf{R}_{xx}^{-1} \) with a vector \( \mathbf{p} \). The first step result indicated in Eq. (4) can be obtained without performing a forward substitution. Since \( \mathbf{L} = \mathbf{L}^{-1} \), we can compute \( \mathbf{g} = \mathbf{L}^{-1} \cdot \mathbf{p} \) by using a matrix vector


\[ \mathbf{g} = \tilde{\mathbf{L}} \cdot \mathbf{p} \]  

(11)

Although it’s possible to obtain the final result by performing a backward substitution using the upper triangular matrix \( \mathbf{U} \), the computing sequence is actually opposite to that of the decomposition process. From the hardware implementation aspect, this indicates that these two computation phases, i.e. decomposition and backward substitution, cannot overlap. This leads to a prolonged computing latency. To mitigate the problem, the inverse of matrix \( \mathbf{U} \) is computed explicitly. The backward substitution is then replaced by a matrix vector multiplication \( \mathbf{M} \cdot \mathbf{g} \), where \( \mathbf{M} = \mathbf{U}^{-1} \). The two computation phases can be chained together to shorten the latency. Refer to Eq. (10), Matrix \( \mathbf{M} \) can be calculated as

\[ \mathbf{M} = \mathbf{U}^{-1} = (\mathbf{D} \cdot \mathbf{L}^H)^{-1} = \tilde{\mathbf{L}} \cdot \mathbf{D}^{-1} \]

(12)

\( \mathbf{D} \) is a diagonal matrix consists of the main diagonal of matrix \( \mathbf{U} \). \( \mathbf{D}^{-1} \) can be easily obtained by taking the reciprocal of diagonal elements. The pseudo code for computing \( \mathbf{R}_{xx}^{-1} \cdot \mathbf{p} \) based on the Schur decomposition result \( \tilde{\mathbf{L}} \cdot \mathbf{R}_{xx} = \mathbf{U} \) is shown in Figure 3.

### Algorithm 2: computing \( \mathbf{R}_{xx}^{-1} \cdot \mathbf{p} \) based on Schur decomposition result

**INPUT:** \( \mathbf{L} \) /* L matrix of Schur decomposition */

**INPUT:** \( \mathbf{D} = \text{diag}(\mathbf{U}) \) /* U matrix of Schur decomposition */

**INPUT:** \( \mathbf{p} \) /* Cross-correlation vector */

1. For \( i = 1 \) to \( N \)
2. For \( j = i \) to \( N \)
3. \( \mathbf{m}_i(j) = \text{conj}(\mathbf{L}_i(j) / d_j) \)
4. End
5. For \( j = 1 \) to \( i \)
6. \( \tilde{\mathbf{p}} = \mathbf{L}_i(j) \cdot \mathbf{p}_j \)
7. \( g_i = g_j \cdot \tilde{\mathbf{p}} \)
8. End
9. For \( j = 1 \) to \( i \)
10. \( \tilde{\mathbf{w}} = \mathbf{m}_i(j) \cdot g_i \)
11. \( \mathbf{w}_j = \mathbf{w}_j + \tilde{\mathbf{w}} \)
12. End
13. OUTPUT: \( \mathbf{w} \) /* weight vector of Sample matrix inversion */

**FIGURE 3.** Efficient computing scheme of multiplying an auto-correlation inverse matrix with a vector.

### III. ALGORITHM PERFORMANCE SIMULATION RESULTS AND ANALYSES

There are two basic assumptions made in developing the efficient beamforming computing schemes as shown in section II. The first one is that the auto-correlation matrix \( \mathbf{R}_{xx} \) is positive definite. The second one is that the received signal vector \( \mathbf{x} \) is stationary so that becomes \( \mathbf{R}_{xx} \) Toeplitz. The first assumption does not always hold and some remedial measures should be applied. As for the second assumption, in implementation practices, the auto-correlation matrix can only be estimated by taking average over a limited length of data. If the length is not sufficiently long, the estimated \( \mathbf{R}_{xx} \) may not be a Toeplitz one even if the signal is truly stationary. Some form of approximation has to make, which would inevitably lead to some beamforming performance loss. This effect should also be examined through the simulations.

The wireless communication system adopted to evaluate the performance of beamforming schemes largely follows the IEEE 802.11ac standard. A \( 1 \times 8 \) SIMO (single-input multiple-output) OFDM system configuration with 8 receiving antennas is assumed. The data modulation format is 64-QAM. The propagation path of the channel pulse response is set to 11-taps, and the azimuth direction of arrival (DoA) angle of the signal is assumed to be 0 degree. One VHT-LTF (very high throughput – long training field) is used as the training symbol to calculate the autocorrelation matrix and cross-correlation vector. The training symbol length is 512.

**FIGURE 4.** Simulated beam patterns of SMI and MVDR scheme.

### A. DIAGONAL LOADING SCHEME

For the positive definite property of the auto-correlation matrix, refer to Eq. (7), if the noise term \( \mathbf{R}_{nn} \) is ignored, \( \mathbf{R}_{xx} \) is positive definite if \( \mathbf{AR}_{ss} \mathbf{A}^H \) is full ranked. The full rank condition holds if rank(\( \mathbf{R}_{ss} \)) equals \( N \), the number of receiving antennas. This means there should be \( N \) independent signal sources, which may not be true if the channel environment is not scattering rich. It is now up to the noise term \( \mathbf{R}_{nn} \) to ensure the full rank property of \( \mathbf{R}_{xx} \). However, when the SNR (signal to noise ratio) value of the received signal is high, the noise term becomes negligible and \( \mathbf{R}_{xx} \) becomes ill-conditioned. This will adversely affect the beamforming performance. Figure 4 shows the beam patterns of SMI and MVDR beamforming under two SNR values, i.e. 0dB and 15dB. In the 0dB cases where significant AWGNs (additive White Gaussian noises) are observed, the beam patterns of the beamforming schemes both exhibit a dominant peak at degree zero, the assumed DoA. However, in the 15dB cases, the beam patterns of both schemes fail to match the...
assumed DoA. To improve the robustness of the beamformer, diagonal loading schemes [23]–[25] can be applied by adding a scaled identity matrix to the original $\mathbf{R}_{xx}$. The scaling factor (or loading factor) is $\gamma$.

$$\tilde{\mathbf{R}}_{xx} = \mathbf{R}_{xx} + \gamma \cdot \mathbf{I},$$  \hspace{1cm} (13)

A heuristic rule to determine the loading factor is setting its value proportional to the trace of $\mathbf{R}_{xx}$, i.e., the summation of the main diagonal elements.

$$\gamma = \varepsilon \cdot \text{trace} \{ \mathbf{R}_{xx} \},$$  \hspace{1cm} (14)

where $\varepsilon$ is the power ratio of the loading factor, and different magnification values are set according to different antenna array size. Generally, the value is greater than the reciprocal of the number of antennas [26]. Fig. 5 shows the simulation of the SMI scheme power beam patterns under different $\varepsilon$ values when SNR = 20dB. It can be observed that a large $\varepsilon$ value can achieve better sidelobe suppression. Fig. 6 shows the symbol error rate (SER) simulations of the MVDR scheme with ($\varepsilon = 0.01 \sim 2.2$) and without ($\varepsilon = 0$) diagonal loading technique. The results show that SER improves with the increase of the $\varepsilon$ value. When $\varepsilon$ value reaches 0.05 and beyond, the SER curves almost overlap. A magnified plot further reveals the proposed schemes with Toeplitz approximation even exhibit a small performance edge over their counterpart schemes (0.25db and 0.13dB, respectively). This means the Toeplitz approximation technique can reduce the beamforming computing complexity by an order without causing any performance loss.

### IV. LOW COMPLEXITY DUAL MODE BEAMFORMER PROCESSOR DESIGN

#### A. ACCELERATOR DESIGN SPECIFICATIONS AND SYSTEM ARCHITECTURE

The design specs of the beamformer processor, the hardware accelerator of the proposed beamforming scheme, are...
TABLE 1. Wireless system setting and beamformer design specs.

| Parameter                          | Value     | Parameter                          | Value |
|------------------------------------|-----------|------------------------------------|-------|
| Channel BW/system clock            | 160 MHz   | FFT size(N)                        | 512   |
| Spatial streams(Nss)               | 1         | IFFT/FFT(TFFT)                     | 3.2 μs|
| MCSs                               | 9         | Short guard interval(TGd)          | 0.4 μs|
| Modulation                         | 64-QAM    | Long guard interval(TGll)          | 0.8 μs|
| Coding rate(R)                     | 3/4       | Data Rate                          | 867 Mbps|
| Autocorrelation matrix estimation  | 128 clock cycles | Beamforming vector calculation     | 128 clock cycles |

The system configuration is largely based on the specifications of Very High Throughput Physical Layer (VHT PHY) of the IEEE 802.11ac standard. The channel bandwidth is 160MHz, which also defines the sampling rate. The beamforming is assumed to be performed, along with packet detection, synchronization, and channel estimation) within the time frame bounded by a VTH-LTF symbol, which includes a long preamble of 512 points (the FFT size) and a long guard interval of 128 points. The total time span is thus 4 μs. Since there are other tasks to be accomplished in the period of long preamble, the time budget for beamforming is set to 256 clock cycles, where 128 of them are allocated for auto-correlation matrix and cross-correlation vector estimations. The remaining 128 clock cycles are reserved for beamforming vector calculation. Note that the DoA estimation hardware is not implemented in this design. If it were included in the design, it should be performed within the time frame of auto-correlation matrix estimation.

We will next present a unified architecture design, which is basically a collection of distinct DSP modules needed in both algorithms. With the hardware centric deliberation in algorithm development, these DSP modules can be seamlessly pipelined and the number of distinct DSP modules can be minimized, as well. Fig. 8 shows the system architecture of the proposed SMI and MVDR dual mode beamformer. There are 9 modules in total and 7 of them are shared by the two beamforming schemes. An input buffer (SV module) is used to collect data from different antennas. The Toeplitz matrix $\mathbf{R}_{xx}$ is generated by the autocorrelation matrix (AM) module. The matrix is updated with the diagonal loading scheme in the DL module. The Schur decomposition is performed next in the SD module to obtain the $\tilde{\mathbf{L}}$ and $\mathbf{U}$ matrices, respectively. Then the $\mathbf{M}$ matrix and the $\mathbf{g}$ vector are calculated in the MM and GV modules, respectively, for Cholesky decomposition. Finally the SMI weights $\mathbf{w}_{SMI} = \mathbf{R}_{xx}^{-1} \cdot \mathbf{p}$ are obtained in the SW module where $\mathbf{p}$ is cross-correlation vector. In the MVDR mode, refer to Eq. (15), $\tilde{\mathbf{w}} = \mathbf{R}_{xx}^{-1} \cdot \mathbf{a}(\theta)$ is first calculated in the same way as in the SMI mode. Parameter $\lambda$ is next calculated, and finally the MVDR weights are performed in the MW module.

\[
\begin{align*}
\mathbf{w}_{SMI} &= \mathbf{R}_{xx}^{-1} \cdot \mathbf{p} \\
\tilde{\mathbf{w}} &= \mathbf{R}_{xx}^{-1} \cdot \mathbf{a}(\theta) \\
\Rightarrow \lambda &= \frac{1}{\mathbf{a}(\theta)^{H} \tilde{\mathbf{w}}} \\
\mathbf{w}_{MVDR} &= \lambda \tilde{\mathbf{w}},
\end{align*}
\]

**FIGURE 8.** System architecture of SMI / MVDR dual mode beamformer.

**FIGURE 9.** 4 × 4 Schur decomposition method mapping block diagram.

### B. ALGORITHM TO HARDWARE ARCHITECTURE MAPPING

A canonical mapping procedure [27] from algorithm representations, i.e., dependence graphs, to systolic array designs is applied to derive the hardware design of each computing module. The mapping starts with selecting a scheduling vector and a projection vector. An algebraic method called space-time transform is next applied to derive the scheduling and hardware allocation. Due to space limitation, the mapping process of each module is not elaborated and only the final results are given. Fig. 9 shows the systolic array design of a Schur decomposition scheme. For illustration simplicity, a 4 × 4 matrix size is assumed. It consists of two rows of processing elements (PE), each row of size 5 (N + 1). The first row of $\mathbf{R}_{xx}$ (with diagonal loading) enters the array
from the top. This two-row PE array design are responsible for the update of two row vectors, $v_{ti}$ and $u_{ti}$. The leading PE of the first row is also in charge of the computation of reflection coefficient. This calls for a complex-valued division. The final $\tilde{L}$ and $U$ matrices output from the bottom of the array, with one row generated per computing iteration.

Figure 10 shows the pipelined architecture of the SMI weight vector computing module. The $\tilde{L}$ and $U$ matrices from the Schur decomposition module are fed to a 3-stage PE array. Refer to Eq. (12), stage one computes matrix $M$ row by row. A divider is needed to calculate the reciprocal of the diagonal element $d_i$. $N$ multipliers are employed to compute all elements in one $M$ row in parallel. Stage two performs a matrix–vector multiplication to obtain the $g$ vector. A linear systolic array containing $N$ MAC (multiply & accumulate) modules is employed. Stage three performs another matrix–vector multiplication $M \cdot g$ for the SMI beamforming vector result. Its hardware mapping is identical to that of stage 2. Note that the data flows across stages are consistent and can be seamlessly pipelined.

Fig. 11 shows the hardware design of an $8 \times 8$ Schur decomposition corresponding to the linear systolic array of the first computing stage in Figure 10. It takes 8 computing iterations, one for the update of a row, to complete the $8 \times 8$ Schur decomposition.

The two decoders (A and B) generate the multiplexer control signals for operand selection. The circuit designs for $M$ matrix, $g$ vector and the final SMI weight vector can be derived straightforwardly from the systolic arrays shown in Fig. 10. Extra pipeline retiming schemes, however, are applied to reduce the critical path delay of the inner product operation. Fig. 12 depicts the hardware designs of computing an $8 \times 8$ auto-correlation matrix and an $8 \times 1$ cross correlation vector. Both assume a linear systolic array structure. For the auto-correlation matrix module, due to the special structure of Toeplitz matrices, the design requires only one row of calculations, which reduces the hardware complexity by 87% when compared with a full-blown design.

Figure 13 describes the $\lambda$ parameter calculation module unique to the MVDR mode beamforming. It is basically a linear array of complex-valued multiply-and-accumulate (MAC) modules. To shorten the critical path delay, extra
pipelined registers are placed between the multipliers and the adders, and also along the accumulation path.

C. COMPLEX-VALUED HARDWARE MODULE DESIGN

Note that all arithmetic operations in this design are complex-valued operations. A complex-valued multiplication requires 4 real-valued multiplications and two real-valued additions. Complex-valued divisions needed in Schur decomposition, M matrix and \( \lambda \) parameter calculations call for extra design efforts to simplify the circuitry. Refer to Eq. (16), in a conventional approach, the denominator is first converted to a real value by multiplying with its complex conjugate (applied to both the numerator and the denominator). After performing a complex-valued multiplication is the numerator, two real-valued divisions follow to obtain the final result. In total, 6 multiplications, 3 additions, and 2 divisions, all real-valued, are needed.

\[
\frac{(a_r + ja_i)}{(b_r + jb_i)} = \frac{(b_r - jb_i)}{(b_r + jb_i)} \left( \frac{(a_r + ja_i)}{(b_r + jb_i)} \right) = \frac{(a_r b_r + a_i b_i) + j(a_r b_i - a_i b_r)}{(b_r^2 + j b_i^2)}
\]

(16)

To reduce the circuit complexity, the Coordinate Rotation Digital Computer (CORDIC) is adopted. The CORDIC scheme was first introduced in 1959 by Jack E. Volder and can compute various arithmetic functions by applying a sequence of micro-rotations iteratively using shift and add operations only. There are three types of CORDIC computations, i.e. circular, linear and hyperbolic. Each type of computation is further divided to vector and rotation modes. Trigonometric functions can be realized in the circular type computing while multiplications/divisions are supported in the linear type computing. Note that the accuracy of CORDIC scheme improves with the increase of computing iterations. A complex-valued division is divided into two computing phases. In phase 1, the denominator is treated as a linear type vector to complete the real-valued division. The numerator is rotated out each other.

In the second phase, the real-valued divisions are then performed by applying linear type vector mode CORDIC computations. Depending on the area-time complexity trade-off, the implementation of a CORDIC computing module ranges from a purely sequential form to a fully pipelined one. In a purely sequential form, (or a folded architecture), a single set of adder and shifter is used repetitively to complete all micro-rotations. On the contrary, a fully pipelined design performs one micro-rotation in a dedicated pipeline stage. The former design is most hardware economical but requires the longest computing latency while the one achieves the highest throughput at the expense of hardware complexity. In this design, a balance between the circuit complexity and the throughput is sought. Figure 14 (a) shows the circuit design of phase 1 computation, i.e., converting the denominator to a real value and updating the numerator accordingly. The leading CORDIC module, operating in a circular type vector mode, is in charge of converting the denominator to a real value by nullifying its imaginary part. Two micro-rotations are performed at a time and the results are feedback for the subsequent computing iterations. This can reduce the computing latency by one half with a mild increase in circuit complexity when compared with a purely sequential design. Extra simulations were conducted to determine the required iteration number of the CORDIC scheme. The result indicates 8 iterations are sufficient and the computing latency in this design is thus 4 clock cycles. Note that in the M matrix calculation, all elements in the same row of the auto-correlation matrix need to perform the same complex-valued divisions. Therefore, the micro-rotation sequence generated by the leading CORDIC module is passed to the trailing \( N \) CORDIC modules, all operating in a circular type rotation mode, to update their values, respectively. This technique gives rise to an additional 21.8% gate count saving. One more design optimization measure is applied here. According to the original CORDIC algorithm, a scaling factor \( K \) must be multiplied after the sequence of rotations to obtain the correct magnitude of the rotated vector. Since the rotations are applied equally to both numerator and denominator, the scaling factors cancel out each other.

Figure 14(b) shows two CORDIC modules configured in the linear type vector mode to complete the real-valued divisions for the real part and imaginary part, respectively. \( N \) pair of such CORDIC modules are employed to compute all elements in the same matrix row concurrently. The circuit complexity reduction by using the CORDIC scheme is significant. The synthesis results reveal that the gate counts for the complex-valued vector division design using conventional dividers and CORDIC modules are 35.6k and 4.7k, respectively. This means an 86% gate count saving.
D. SYSTEM TIMING DIAGRAM

The timing diagram of the proposed dual-mode 1 × 8 beamformer design is shown in Fig. 15. 128 out of the 512 sampling points from the VTH-LTF symbol are dedicated to estimations of the auto-correlation (AM) matrix and cross-correlation vector (CV). After the estimation results are ready, it takes one clock cycle to perform the diagonal loading (DL). SD stands for the Schur decomposition module and takes 10 clock cycles to complete one row of computations. The M matrix (MM) and the g vector (gV) computations are performed concurrently followed by the calculation of one element of SMI weight vector. Because of this highly parallel computing scheme, it takes only 82 clock cycles to complete the SMI weight vector calculations. For the MVDR mode beamforming, 11 additional clock cycles (and 93 clock cycles in total) are needed to complete the task. This leads to a total latency of 210 and 221 clock cycles, respectively, for the two beamforming modes.

V. CHIP IMPLEMENTATION AND PERFORMANCE EVALUATION

Based on the design presented in section IV, the corresponding hardware accelerator chip design is developed. The synthesis is based on an ARM standard cell library and the implementation technology is a TSMC 90nm UTM CMOS process. The first step is conducting fixed point simulations to obtain a bit true design and evaluating the implementation loss.

A. FIXED POINT SIMULATIONS FOR BIT RATE DESIGN

Fixed point simulations take the windowing and the quantization effects into consideration, and determine the best windowing position and the minimum number of bits (word length) of each variable so that the performance deviation from the floating point version is acceptable. Figure 16 shows the fixed point simulation trials to determine the fractional length of the SMI weight variable. The integral length is bounded by the maximum value collected in the simulation trials, which is 3 bits in this case, so as not to introduce overflow. The fractional length ranges from 1 to 9 bits in simulations. The result shows that the SER improvement saturates when the fractional length reaches 8 bits. The SNR degradation with respect to the floating point version at SER equal to 10⁻³ is less than 0.05dB. The word length for the SMI weight variable is thus set to 11 bits. Similar simulations are conducted, proceeding from input to output, to determine the word lengths of variables at different computing modules. Table 2 summarizes the derived word lengths of output variables in each module. The variable word length of the auto-correlation matrix and the cross-correlation vector modules is larger because of the 128-point-long accumulation phase needed to obtain the estimation results. As mentioned in sub-section IV.C, the CORDIC module iteration number is set to 8.

Based on the word length settings specified in Table 2, Fig. 17 shows the SER curve simulations of the fixed point design versus the floating point version design and the original SMI scheme without any algorithmic simplifications. The results show that the three SER curves basically overlap. The magnified plot indicates the hardware implementation loss is merely 0.2dB (compared with the floating point design). A slight performance edge over the original scheme remains, which is consistent with the simulation results shown in Figure 7.

B. CHIP IMPLEMENTATION AND MEASUREMENT RESULTS

For a design to achieve its desired throughput, 8 complex-valued samples received from the 1 × 8 antenna array must be admitted every clock cycle. While the input data bandwidth is not an issue for on chip communications, it is problematic for off chip I/O communications when IC implementation is

| Module                        | Word length |
|-------------------------------|-------------|
| Autocorrelation matrix        | 17          |
| Cross-correlation vector      | 17          |
| Schur decomposition           | 11          |
| g vector                      | 11          |
| MVDR weight                   | 11          |
| CORDIC                        | 8 iterations|

FIGURE 15. The timing diagram of the proposed dual-mode beamformer design.

FIGURE 16. Fixed-point simulation of the SMI weight module.

TABLE 2. The word length of each function module.
K.-T. Chen et al.: Design and Chip Implementation of a SMI/MVDR Dual-Mode Beamformer

FIGURE 17. Evaluation of the hardware implementation loss of the fixed point design.

FIGURE 18. The hardware complexity breakdown of the design.

FIGURE 19. (a) The 8-antenna SMI/MVDR integrated chip design layout. (b) The photograph of the fabricated chip.

FIGURE 20. Shmoo plot of chip measurements.

considered because of the extremely large pin count. In this implementation, a CLCC 84-pin packaging is used. To cope with the pin count limitation, only one complex-valued sample can be input per clock cycle. The target clock rate, as stated in Table 1, is equal to the baseband bandwidth, i.e., 160MHz. To receive all eight samples through a single input data port within one clock period, the I/O frequency must be 8 times higher, which is considered not feasible. Since our goal is to achieve silicon verification of the kernel design, a prolonged input latency plus an input data buffer to provide the required data bandwidth internally are acceptable. Similarly, only one data port is allocated for the output of weight vector, which takes 8 clock cycles to complete. A mode selection pin is used to switch the operations between SMI and MVDR beamforming. The synthesis result indicates the circuit complexity of the design is about 205k in gate count. 60% of them is for combinational logic and the remaining 40% is for FFs and sequential logic. To facilitate design testability, a scan chain consisting of 5,990 additional FFs is added to the chip design. This makes the total gate count slightly rising to 216k. Synopsys TetraMax is used to generate test patterns. With a set of 251 test patterns, the fault coverage is up to 99.16%. Fig. 18 shows the hardware complexity breakdown of the design.

The Schur decomposition module consumes the largest portion of the gate count. All other computing modules have similar gate count numbers. This is because their designs are all based on a linear systolic array structure. Fig. 19(a) shows the chip layout accomplished by using Synopsys IC Compiler, and Fig. 19(b) is a die photograph after fabrication. The core size is 0.68 mm\(^2\) and the chip die size reaches 2.13 mm\(^2\) after adding I/O pads for CLCC 84-pin packing. The chip measurements were conducted on an Agilent 93000 SOC Test System and the Shmoo plot subject to different power supply voltages and clock rates is illustrated in Figure 20. The green tiles indicate the conditions tests pass. For a nominal power supply voltage (1V), the maximum clock rate is 208 MHz.

Note that when we developed the chip design, some design headroom is added to ensure the speed performance in face of process variations. This leads to an implementation slightly outpacing the required clock rate. 208MHz is the corner case and 200MHz can be considered a nominal clock rate functioning correctly in most cases. In real applications, the clock rate must be in line with the baseband bandwidth,
i.e., 160MHz. We may lower the Vdd voltage setting to achieve power saving. The power consumption measurement results of the entire chip with a nominal 1V Vdd setting are 44.11mW @160MHz and 49.03mW @200MHz.

Finally, the chip design summary is listed in Table 3. As mentioned before, the prolonged input latency is due to the pin count limitation. If the design were used as an IP in the entire baseband chip design, this limitation can be lifted. The initiation interval is defined as the minimal time span between two successive initiations of beamforming vector weight computations. This is actually bounded by the interval of performing the auto-correlation matrix estimation. The throughput is calculated as the reciprocal of the initiation interval. It is 1.25M BF vectors/sec when the clock rate is 160MHz.

| Table 3. Chip design summary. |
|-------------------------------|
| **Item**                     | **Specification**                  |
| Technology process           | TSMC 90nm UTM                       |
| Cell library                 | ARM Cell-based Design Kit           |
| Package                      | CLCC 84                             |
| Voltage (Core/IO)            | 1.0/3.3 V                           |
| Nominal Clock Rates          | 160/200 MHz                         |
| Maximum clock rate           | 208MHz                              |
| Core Size                    | 0.68 mm²                            |
| Die Size                     | 2.13 mm²                            |
| Gate Count                   | 216.64 k                            |
| Power                        | 41.44/49.03 mW                      |
| Input latency                | 128×8 clock cycles                  |
| Computing latency            | 82(SMI)/93(MVDR) clock cycles       |
| Output latency               | 8 clock cycles                      |

C. DESIGN COMPARISONS

In the proposed dual-mode beamformer design, the SMI algorithm and the MVDR algorithm share most of the computing modules, i.e., the Autocorrelation matrix, the Schur decomposition, \( M \) matrix, \( g \) vector, diagonal loading and the SMI weight calculation. The shared hardware accounts for about 70% of the entire design. This leads to a significant circuit complexity reduction when implementing two beamforming modes in a unified design.

Although there are many papers addressing the beamforming schemes (particularly those using the time domain approach), very few of them offer the hardened solutions. Table 4 shows the performance comparison with 5 related designs found in the literature. Two [8], [10] of them support MVDR beamforming and the remaining three [9], [28], [29] use RLS beamforming. Besides the proposed one, design [9] is the only one with an ASIC implementation. It employs a QR decomposition scheme to accomplish the RLS algorithm based beamforming. It is designed for an antenna array of size 41. The design is implemented in a TSMC 0.13um process technology and can operate at 150MHz. The gate count is roughly 5 times the that of the proposed one and the power consumption is formidable (4W). No throughput information is available. Design [10] also adopts a QR decomposition scheme to solve the matrix inversion needed in the MVDR beamforming. The design is implemented in an FPGA platform and the clock rate is 250MHz. It takes 14,190 clock cycles to complete the beamforming weight calculation. Compared with the 128 clock cycles needed in the proposed design, its throughput rate is only one 50th of the proposed one after normalization with the matrix size. The normalization factor is proportional to the \( N^3 \). Designs [28], [29] implement a variant of RLS algorithm based on dichotomous coordinate descent iterations and the implementation technology is FPGA. The reported performance indexes are update rates, which are different from the throughput rate used in the proposed and other MVDR scheme based designs. In the RLS algorithm based design, the beamforming weights are updated incrementally until convergence. For the comparison purpose, we assume the iteration number needed for convergence is equal to the number of input sample vectors needed for the auto-correlation matrix estimation. This means the update rate should be divided by 128 to obtain an equivalent throughput rate. The simulation results in [28], [29] also indicate that hundreds of input sample vectors are needed for the convergence. The equivalent throughput rates of these two designs are much smaller than that of the proposed one even after the normalization. Design [8] realizes a Steered Minimum Variance (STMV) beamforming, which can compute beamforming vector weights from a single time-series snapshot and thus belongs to the MVDR category. The design adopts a programmable solution where a TiherSHARC digital signal processor (DSP) is used. It supports matrix size operations from 16 × 16 to 48 × 64. The cycle count is up to 30,457. Even the design can operate at the highest clock rate (600MHz) among all designs, its throughput rate still falls behind to the proposed one.

Because of the differences in implementation technology, problem size and beamforming scheme, it is difficult to develop other composite performance indexes for the comparison purpose. However, the throughput rate advantage of the proposed design is obvious. This is because the proposed design is the only one adopts a \( O(N^2) \) computing scheme, which gives it an inherent advantage in computing throughput.

VI. EXTENSION TO GENERAL MIMO CONFIGURATIONS

The target 1 × 8 MIMO system configuration suggests that there is only signal source (or data stream) from the transmitter. The proposed scheme, however, can be easily extended to handle the \( M \times N \) configurations by treating it as a superposition of \( M \) of \( 1 \times N \) sub-systems at the receiving end. In particular, the most complex computation in the beamforming schemes, i.e., the inverse of the auto-correlation matrix \( (\mathbf{R}^{-1}_{xx}) \), is common to all signal sources and needs to be computed only once. So is the calculation of auto-correlation matrix. Only the parts distinct to each signal source need
TABLE 4. Performance comparison of related designs.

| Platform  | Proposed scheme | RLS [9] | QRD [10] | RLS-DCC [28] | RLS-DCC [29] | STMV [8] |
|-----------|-----------------|---------|----------|--------------|--------------|---------|
| Antenna size | 8               | 41      | 10       | 16/64        | 9/16         | 16      |
| Operating frequency | 200MHz          | 150 MHz | 250MHz   | 100MHz       | 100MHz       | 600MHz  |
| Platform   | ASIC            | ASIC    | FPGA     | FPGA         | FPGA         | DSP     |
| process    | 90nm            | 130nm   | --       | --           | --           | --      |
| scheme     | SMV/MVDR       | RLS     | MVDR     | RLS          | MVDR         |
| Gate count | 216.64k        | 106k    | --       | --           | --           |
| power      | 49mW           | 4W      | --       | --           | --           |
| Cycle count | 128            | 14,190  | 567/3,219 | 490/6,336    | 30,457       |
| Computing latency | 0.64us       | --      | 56.76us  | --           | 50.76us     |
| Update rate | --             | --      | 176k/31k | 207k/16k  | 19.7k      |
| Throughput (weights/sec) | 1.56M          | 17.6k   | --       | --           | 157.6k      |
| Normalized throughput | 1.56M         | 30.44k  | 11k      | 2.3k/8k     |

The reported numbers are for one update iteration only.

as the product of a beamforming matrix and a precoding matrix. To reduce the computing complexity, the beamforming matrix is constructed by choosing appropriate steering vectors from a codebook using a heuristic scheme such as orthogonal matching pursuit. Interested readers are referred to [30], [31] for details. In most real system implementations, a straightforward beam search scheme is often employed. In this approach, the transmitter and the receiver simply enumerate all possible beam direction combinations to determine the best of each. This approach, however, is applicable to the case of one signal source.

VII. DISCUSSION AND CONCLUSION

A low complexity dual mode SMI/MVDR beamformer chip design was proposed. The design was carefully developed starting from the algorithm development, architecture mapping, circuit optimization to the final chip implementation. Since algorithm complexity plays a crucial role in hardware implementation, various algorithm simplification measures were applied. These include identifying the shared modules of two beamforming modes, approximating the Hermitian type auto-correlation matrix with a Toeplitz type counterpart, and developing an efficient computing scheme based on Cholesky and Schur decompositions for matrix inversion. The computing complexity was successfully reduced by an order, from $O(N^3)$ to $O(N^2)$. The ill-conditioned matrix inversion problem was properly tackled by a diagonal loading technique and no noticeable performance loss arises due to these algorithm relaxation / simplification measures.

The developed computing schemes were next mapped to a systolic array design containing 6 pipelined computing stages. The two beamforming designs coexist in this unified hardware architecture with 80% of the circuitry are shared. The timing specs of the chip design is compliant with the IEEE 802.11ac standard. To reduce the overall design gate count, complex-valued divisions were implemented by using the CORDIC scheme. A scan chain was also added to improve the testability.

The implementation result on a TSMC 90nm UTM process technology indicates a core size of 0.68 mm² and a die size of 2.13 mm². The equivalent logic gate is merely 216.64k. The chip can work successfully at a nominal 160MHz clock.
rate and can further operate up to 200MHz with a power consumption of 49.03mW. It shows the highest throughput among 6 related beamforming hardware designs. The initiation interval of successive beamforming vector weight updates for a sized 8 antenna array is as small as 128 clock cycles, which is only 0.64us when operating at the highest clock rate (200MHz). The low complexity and high throughput features of the proposed design make it very suitable for applications in massive MIMO communication systems.

Regarding the future work, we have discussed the design extension to a general $M \times N$ MIMO system. The principle is treating the system as a superposition of $M \times 1 \times N$ sub-systems. Since the inverse of the auto-correlation matrix ($R_{xx}^{-1}$), is common to all signal sources, it is computed once and the corresponding computing module can be shared by $M$ sub-systems. Only the weight vectors need to be calculated for each signal source. We may either introduce $M$ parallel computing modules to maintain the throughput or use one module only to process them sequentially.

REFERENCES

[1] H. Yan, S. Ramesh, T. Gallagher, C. Ling, and D. Cabric, “Performance, power, and area design trade-offs in millimeter-wave transmitter beamforming architectures,” *IEEE Circuits Syst. Mag.*, vol. 19, no. 2, pp. 33–58, May 2019.

[2] M. Novrati, S. Shahsavari, S. Lee, H. Wang, and N. Tavassolian, “A concurrent dual-beam phased-array Doppler radar using MIMO beamforming techniques for short-range vital-signs monitoring,” *IEEE Trans. Antennas Propag.*, vol. 67, no. 4, pp. 2390–2404, Apr. 2019.

[3] Y.-S. Su and C.-C. Chang, “Designs of deformed eight-way butler matrix with port reduction for massive array beamforming,” *IET Microw., Antennas & Propag.*, vol. 13, no. 6, pp. 864–869, May 2019.

[4] S. Jiang, R. Lu, J. Jeong, and M. P. Flynn, “A 1-GHz 16-element four-beam true-time-delay digital beamformer,” *IEEE J. Solid-State Circuits*, vol. 54, no. 5, pp. 1304–1314, May 2019.

[5] Y. Yu, W. Hong, Z. H. Jiang, H. Zhang, and C. Guo, “Multibeam generation and measurement of a DDS-based digital beamforming array transmitter at Ka-band,” *IEEE Trans. Antennas Propag.*, vol. 67, no. 5, pp. 3030–3039, May 2019.

[6] V. V. Zaharov and M. Teixeira, “Modified implementations of RLS digital beamformer,” in *Proc. IEEE Antennas Propag. Soc. Int. Symp.*, vol. 3, Jul. 2005, pp. 275–278.

[7] V. V. Zaharov and M. Teixeira, “SMI-MVDR beamformer implementations for large antenna array and small sample size,” *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 55, no. 10, pp. 3317–3327, Nov. 2008.

[8] C. S. Mohan, S. B. Menon, K. E. Nissar, and K. P. A. Kumar, “Adaptive beamforming with instantaneous convergence & its efficient implementation in DSP hardware,” in *Proc. Ann. IEEE India Conf. (INDICON)*, Dec. 2012, pp. 700–704.

[9] Z. Liu, J. V. McCanny, G. Lightbody, and R. Walké, “Generic SoC QR array processor for adaptive beamforming,” *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 50, no. 4, pp. 169–175, Apr. 2003.

[10] C. Dick, F. Harris, M. Pajic, and D. Vuletic, “Real-time QRD-based beamforming on an FPGA platform,” in *Proc. 40th Asilomar Conf. Signals, Syst. Comput.*, Nov. 2006, pp. 1200–1204.

[11] M. Budabathon, Y. Hara, and S. Hara, “Optimum beamforming for pre-FFT OFDM adaptive antenna array,” *IEEE Trans. Veh. Technol.*, vol. 53, no. 4, pp. 945–955, Jul. 2004.

[12] B. Liu, R. Jin, and Y. Fan, “Modified pre-FFT OFDM adaptive antenna array with beam-space channel estimation,” *Electron. Lett.*, vol. 40, no. 5, pp. 287–288, Mar. 2004.

[13] L. Fan, H. Zhang, and C. He, “Minimum bit error rate beamforming for pre-FFT OFDM adaptive antenna array,” in *Proc. IEEE 62nd Veh. Technol. Conf. (VTC-Fall)*, vol. 6, Sep. 2005, pp. 359–363.

[14] S. Hara, M. Budabathon, and Y. Hara, “A pre-FFT OFDM adaptive antenna array with eigenvector combining,” in *Proc. IEEE Int. Conf. Commun.*, vol. 4, Jun. 2004, pp. 2412–2416.

[15] Z. Lei and F. P. S. Chiu, “Post and pre-FFT beamforming in an OFDM system,” in *Proc. IEEE 9th Veh. Technol. Conf. (VTC-Spring)*, vol. 1, May 2004, pp. 39–43.

[16] H. Matsuoka and H. Shoki, “Comparison of pre-FFT and post-FFT processing adaptive arrays for OFDM systems in the presence of co-channel interference,” in *Proc. 14th IEEE Proc. Pers., Indoor Mobile Radio Commun. (PIMRC)*, vol. 2, Sep. 2003, pp. 1603–1607.

[17] H. Matsuoka, H. Kasami, M. Tsuruta, and H. Shoki, “A smart antenna with pre-and post-FFT hybrid domain beamforming for broadband OFDM system,” in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2006, pp. 1916–1920.

[18] C.-Y. Chi, C.-H. Peng, K.-C. Huang, T.-H. Tsai, and W.-K. Ma, “A block-by-block blind post-FFT multistage beamforming algorithm for multiuser OFDM systems based on subcarrier averaging,” *IEEE Trans. Wireless Commun.*, vol. 7, no. 8, pp. 3328–3351, Aug. 2008.

[19] H. W. Press, A. S. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*, vol. 5, 2nd ed. Cambridge, U.K.: Cambridge Univ., 1992, p. 994.

[20] N. Levinson, “The Wiener RMS error criterion in filter design and prediction,” *J. Math. Phys.*, vol. 25, nos. 1–4, pp. 261–278, 1946.

[21] J. Durbin, “The fitting of time-series models,” *Rev. Stat. Inst.*, vol. 28, no. 3, pp. 233–243, 1960.

[22] E. H. Bareiss, “Numerical solution of linear equations with Toeplitz and vector Toeplitz matrices,” *Numerische Math.*, vol. 13, no. 5, pp. 404–424, Oct. 1969.

[23] C. Jianhuo and W. Xuegang, “Diagonally loaded SMI algorithm based on inverse matrix recursion,” *J. Syst. Eng. Electron.*, vol. 18, no. 1, pp. 160–163, Mar. 2007.

[24] B. D. Carlson, “Covariance matrix estimation errors and diagonal loading in adaptive arrays,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-24, no. 4, pp. 397–401, Jul. 1988.

[25] J. Li, P. Stoica, and Z. Wang, “On robust capon beamforming and diagonal loading,” *IEEE Trans. Signal Process.*, vol. 51, no. 7, pp. 1702–1715, Jul. 2003.

[26] Z. Tian, K. L. Bell, and H. L. Van Trees, “Robust constrained linear receivers for CDMA wireless systems,” *IEEE Trans. Signal Process.*, vol. 49, no. 7, pp. 1510–1522, Jul. 2001.

[27] S. Y. Kung, *VLSI Array Processors*. Englewood Cliffs, NJ: Prentice-Hall, 1988.

[28] J. Liu and Y. Zakharov, “Dynamically regularized RLS-DCD algorithm and its FPGA implementation,” in *Proc. 42nd Asilomar Conf. Signals, Syst. Comput.*, Oct. 2008, pp. 1876–1880.

[29] J. Liu and Y. Zakharov, “FPGA implementation of RLS adaptive filter using dichotomous coordinate descent iterations,” in *Proc. IEEE Int. Conf. Commun.*, Jun. 2009, pp. 1–5.

[30] K.-T. Chen, Y.-T. Hwang, and Y.-C. Liao, “VLSI design of a high throughput hybrid precoding processor for wireless MIMO systems,” *IEEE Access*, vol. 7, pp. 85925–85936, 2019.

[31] Y.-Y. Lee, C.-H. Wang, and Y.-H. Huang, “A hybrid RF/baseband precoding processor based on parallel-index-selection matrix-inversion-bypass simultaneous orthogonal matching pursuit for millimeter wave MIMO systems,” *IEEE Trans. Signal Process.*, vol. 63, no. 2, pp. 305–317, Jan. 2015.
YIN-TSUNG HWANG (Member, IEEE) received the B.S. and M.S. degrees in electronic engineering from National Chiao Tung University, Hsinchu, Taiwan, in 1983 and 1985, respectively, and the Ph.D. degree from the Department of Electrical and Computer Engineering, University of Wisconsin–Madison, Madison, in 1993.

Since 1993, he had been a Faculty Member with the Department of Electronic Engineering, National Yunlin University of Science and Technology, Taiwan. In 2004, he joined the Department of Electrical Engineering, National Chung Hsing University, where he is currently a Professor. He was the Director of the Meng Yao Chip Center, National Chung Hsing University, from 2007 to 2010. His research interests include VLSI designs for wireless baseband signal processing, beamforming and radar signal processing, image compression, and low-power VLSI circuit designs. He has also been a Council Member of the Taiwan Integrated Circuits Design Association, since 2014.

CHENG-YI HUANG received the B.S. degree from the Department of Electronic Engineering, National Yunlin University of Science and Technology, Taiwan, in 2012, and the M.S. degree from the Department of Electrical Engineering, National Chung Hsing University, Taichung, Taiwan, in 2014. He has been with MediaTek Inc., Hsinchu, Taiwan, since 2014. His work has been focusing on low-power digital circuit designs, including thermal, power, and performance management for advanced CMOS.