High-Throughput Data Detection for Massive MU-MIMO-OFDM using Coordinate Descent

Michael Wu, Chris Dick, Joseph R. Cavallaro, and Christoph Studer

Abstract—Data detection in massive multi-user (MU) multiple-input multiple-output (MIMO) wireless systems is among the most critical tasks due to the excessively high implementation complexity. In this paper, we propose a novel, equalization-based soft-output data-detection algorithm and corresponding reference FPGA designs for wideband massive MU-MIMO systems that use orthogonal frequency-division multiplexing (OFDM). Our data-detection algorithm performs approximate minimum mean-square error (MMSE) or box-constrained equalization using coordinate descent. We deploy a variety of algorithm-level optimizations that enable near-optimal error-rate performance at low implementation complexity, even for systems with hundreds of base-station (BS) antennas and thousands of subcarriers. We design a parallel VLSI architecture that uses pipeline interleaving and can be parametrized at design time to support various BS antenna configurations. We develop reference FPGA designs for massive MU-MIMO-OFDM systems and provide an extensive comparison to existing designs in terms of implementation complexity, throughput, and error-rate performance. For a 128 BS antenna, 8 user massive MU-MIMO-OFDM system, our FPGA design outperforms the next-best implementation by more than 2.6× in terms of throughput per FPGA look-up tables.

Index Terms—Coordinate descent, equalization, FPGA design, massive multi-user (MU) MIMO, orthogonal frequency-division multiplexing (OFDM), soft-output data detection.

I. INTRODUCTION

MASSIVE multi-user (MU) multiple-input multiple-output (MIMO) technology promises significant improvements in terms of spectral efficiency, coverage, and range compared to traditional, small-scale MIMO [2]–[5]. In fact, massive MU-MIMO is commonly believed to be one of the key technologies for future fifth-generation (5G) wireless systems [6]. The idea underlying this technology is to equip the base-station (BS) with hundreds of antenna elements while communicating with tens of user terminals concurrently and within the same time-frequency resource. However, the large dimensionality of the data detection problem faced in the uplink (where users communicate to the BS), results in excessively high implementation complexity at the BS (see, e.g., [7] and the references therein). Hence, to reduce the implementation costs while enabling throughputs in the Gb/s regime for practical wideband massive MU-MIMO systems with hundreds of antenna elements and thousands of subcarriers, novel algorithms and dedicated hardware implementations on field-programmable gate arrays (FPGAs) or application specific integrated circuits (ASICs) are necessary.

During recent years, various data-detection algorithms [8], [9] and dedicated hardware implementations have been proposed for massive MU-MIMO systems [7], [10]–[13]. All of the existing hardware implementations, however, are either unable to achieve the high throughputs offered by future wideband massive MU-MIMO systems [7], [12], [13], or exhibit excessive hardware complexity [11]. Furthermore, the hardware implementations in [7], [11] only support single-carrier frequency-division multiple-access (SC-FDMA). As demonstrated in [14], however, orthogonal frequency-division multiplexing (OFDM) enables (often significantly) less complex baseband processing which may be a critical design factor for wideband massive MU-MIMO systems with hundreds of BS antennas and thousands of subcarriers.

A. Contributions

We propose a new, low-complexity soft-output data-detection algorithm and a corresponding high-throughput FPGA design for massive MU-MIMO wireless systems that use OFDM. Our algorithm, referred to as optimized coordinate descent (OCD), performs approximate minimum mean-square error (MMSE) or box-constrained equalization, which enables near maximum-likelihood (ML) soft-output data detection performance in massive MU-MIMO systems with a large BS-to-user-antenna ratio. We develop a corresponding high-throughput VLSI architecture with a deep and interleaved pipeline, which can be parametrized at design time to support various BS and user antenna configurations. The algorithmic regularity of OCD and the fact that preprocessing can be implemented at minimum hardware overhead enables high-throughput VLSI designs that require lower complexity than state-of-the-art designs, even for systems with hundreds of BS antennas and thousands of subcarriers. To demonstrate the advantages of OCD compared to existing massive MU-MIMO data-detector designs in terms

1SC-FDMA typically generates baseband signals with a lower dynamic range, but the receiver must perform an additional frequency-to-time conversion (compared to OFDM). This additional conversion step requires one to separate equalization (that is usually carried out in the frequency domain per subcarrier) and data detection (that must be carried out in the time domain). This separation prevents the use of powerful, non-linear equalization methods [15], such as the box-constrained detector proposed in this paper. OFDM, in contrast, causes a slightly higher dynamic range, but requires only one time-to-frequency conversion and enables non-linear data-detection methods that operate directly in the frequency domain on a per-subcarrier basis [16].
of throughput, hardware complexity, and error-rate performance, we provide implementation results on a Xilinx Virtex-7 FPGA.

B. Notation

Boldface lowercase and boldface uppercase letters stand for column vectors and matrices, respectively. For a matrix $\mathbf{A}$, we denote its hermitian transpose by $\mathbf{A}^H$. We use $a_{k,\ell}$ for the entry in the $k$th row and $\ell$th column of the matrix $\mathbf{A}$; the $k$th entry of a column vector $\mathbf{a}$ is denoted by $a_k$. The $\ell_2$-norm of a vector $\mathbf{a}$ is defined as $\|\mathbf{a}\|_2 = \sqrt{\sum_k |a_k|^2}$. The real part of a complex number $a$ is $\Re\{a\}$. Sets are denoted by uppercase calligraphic letters; the cardinality of a set $\mathcal{A}$ is $|\mathcal{A}|$. The expectation operator is designated by $\mathbb{E}\{\cdot\}$.

C. Paper Outline

The rest of the paper is organized as follows. Section II introduces the massive MU-MIMO-OFDM system model and describes data detection using MMSE and box-constrained equalization. Section III details our OCD architecture and shows error-rate simulation results. Section IV and Section V describe our VLSI architecture and shows FPGA implementation results, respectively. We conclude in Section VI.

II. SYSTEM MODEL AND DATA DETECTION

This section introduces the considered OFDM-based uplink model and summarizes efficient methods for linear MMSE and box-constrained soft-output data detection.

A. OFDM-based Uplink System Model

We consider a massive MU-MIMO-OFDM uplink system, where $\mathcal{U}$ single-antenna user terminals send data simultaneously to a BS with $B \gg \mathcal{U}$ antennas over $\mathcal{W}$ subcarriers. Each user $i = 1, \ldots, \mathcal{U}$ encodes its own bit stream (using a forward error-correction scheme) and maps the generated coded bits onto constellation points in a finite set $\mathcal{O}$ (e.g., 64-QAM using a Gray mapping rule), with unit average transmit power, i.e., $\mathbb{E}\{|s|^2\} = 1$ with $s \in \mathcal{O}$, and $Q = \log_2 |\mathcal{O}|$ bits per constellation point. The resulting $W$ frequency-domain symbols $\{s_1^{(i)}, \ldots, s_W^{(i)}\}$ are then transformed into the time domain (TD) using an inverse discrete Fourier transform (DFT) [16]. After prepending the cyclic prefix, all users transmit their TD signals over the frequency-selective wireless channel at the same time.

After removing the cyclic prefixes, the TD signals received at each BS antenna are transformed back to the FD using a DFT. For the sake of simplicity, we assume a sufficiently long cyclic prefix, perfect synchronization, and that perfect channel-state information (CSI) has been acquired via pilot-based training. Under these assumptions, the FD input-output relation on the $w$th subcarrier is commonly modeled as [17]

$$y_w = \mathbf{H}_w \mathbf{s}_w + \mathbf{n}_w,$$  

where $y_w \in \mathbb{C}^B$ is the associated received FD vector, $\mathbf{H}_w \in \mathbb{C}^{B \times \mathcal{W}}$ is the channel matrix, $\mathbf{s}_w \in \mathbb{C}^\mathcal{W}$ contains the symbols transmitted by all $\mathcal{U}$ users, i.e., $[\mathbf{s}_w]_i = s_i^{(w)}$ refers to the symbol transmitted by user $i$ over subcarrier $w$, and $\mathbf{n}_w \in \mathbb{C}^\mathcal{W}$ models thermal noise as i.i.d. complex circularly-symmetric Gaussian vector with variance $N_0$ per complex entry.

B. Equalization-based Data Detection

For the model in (1), optimal data detection in terms of minimizing the symbol error-rate is accomplished by solving the maximum-likelihood (ML) problem [18]

$$\hat{s}_{\text{ML}}^w = \arg\min_{z \in \mathcal{O}^\mathcal{W}} \|y_w - \mathbf{H}_w \mathbf{z}\|^2_2.$$  

Unfortunately, solving (2) exactly for massive MU-MIMO systems quickly results in prohibitive complexity, even with the best-known sphere-decoding algorithms [19]. Equalization-based data detection algorithms [18] enable one to find approximate solutions to the ML problem at low computational complexity. Virtually all linear as well as non-linear equalization methods relax the finite-alphabet constraint $z \in \mathcal{O}^\mathcal{W}$ in (2), which enables the efficient computation of an estimate $\hat{s}$ that is (hopefully) close to the ML solution. The estimate $\hat{s}$ can then either be sliced element-wise onto the nearest constellation point in $\mathcal{O}$ as follows:

$$\hat{s}_i = \arg\min_{z \in \mathcal{O}} \|\hat{s}_i - z\|, \quad i = 1, \ldots, \mathcal{U},$$  

which is known as hard-output data detection, or be used to compute reliability information for each transmitted bit in the form of log-likelihood ratio (LLR) values (see Section II-E), which is known as soft-output data detection [20], [21].

C. Linear MMSE Equalization

The most common equalization-based data detection algorithm is linear MMSE data detection [18], [20]. This method was shown to enable FPGA and ASIC designs that are able to achieve high throughput in massive MU-MIMO systems [7]. Furthermore, for systems with large BS-to-user antenna ratios $\delta = B/\mathcal{U}$ (e.g., two or larger), linear detectors are able to achieve near-ML error-rate performance [3]–[5].

The key idea of MMSE data detection is to relax the constraint $z \in \mathcal{O}^\mathcal{W}$ in the ML problem (2) to the $\mathcal{U}$-dimensional complex space $z \in \mathbb{C}^\mathcal{U}$ and to include a quadratic penalty function. In particular, MMSE equalization solves the following regularized least-squares problem [10], [22]:

$$\hat{s}_{\text{MMSE}}^w = \arg\min_{z \in \mathbb{C}^\mathcal{U}} \|y_w - \mathbf{H}_w \mathbf{z}\|^2_2 + N_0 \|z\|^2.$$  

Since the objective function in (3) is quadratic in $z$, the MMSE equalization problem has a closed-form solution.

An explicit solution to (4) can be computed as follows. First, compute the regularized Gram matrix $\mathbf{A}_w = \mathbf{G}_w + N_0 \mathbf{I}_{\mathcal{U}}$, with $\mathbf{G}_w = \mathbf{H}_w^H \mathbf{H}_w$ and the matched filter vector $\mathbf{s}_{\text{MF}}^w = \mathbf{H}_w^H y_w$. Then, the MMSE estimate in (4) is computed as

$$\hat{s}_{\text{MMSE}}^w = \mathbf{A}_w^{-1} \mathbf{s}_{\text{MF}}^w.$$  

While this closed-form approach was shown to be efficient for traditional, small-scale MIMO systems (e.g., with four antennas at both ends of the wireless link) [21], computing the regularized Gram matrix $\mathbf{A}_w$ and its inverse $\mathbf{A}_w^{-1}$ quickly results in prohibitive complexity in massive MU-MIMO systems with hundreds of BS antennas [11]. In Section III, we present a computationally-efficient equalization algorithm that directly solves (4) in a hardware efficient way, which avoids expensive calculations such as the computation of the regularized Gram matrix $\mathbf{A}_w$ and its inverse $\mathbf{A}_w^{-1}$.
D. Non-Linear Box-Constrained (BOX) Equalization

While linear equalization methods are the most common approach in the MIMO literature, a few non-linear equalizers have recently emerged and were shown to outperform linear methods in terms of error-rate performance [23]. A promising non-linear equalization method, referred to as box-constrained equalization (short BOX equalization) [24], [26], relaxes the constraint $z \in \mathcal{O}^U$ to the convex polytope $\mathcal{C}_\mathcal{O}$ around the constellation set $\mathcal{O}$, which is formally defined as follows:

$$\mathcal{C}_\mathcal{O} = \left\{ \sum_{i=1}^{\left|\mathcal{O}\right|} \alpha_i s_i \mid \alpha_i \geq 0, \alpha_i \in \mathbb{R}, \forall i \right\} \cap \left\{ \sum_{i=1}^{\left|\mathcal{O}\right|} \alpha_i = 1 \right\}. \quad (6)$$

For example, the convex polytope $\mathcal{C}_{\text{QPSK}}$ for QPSK with $\frac{1}{2}$ is given by $\mathcal{C}_{\text{QPSK}} = \{ +1 + j, +1 - j, -1 + j, -1 - j \}$ (7) is simply a box with radius 1 around the square constellation (thus the name BOX equalization). For higher-order QAM alphabets, such as 16-QAM or 64-QAM, we have $\mathcal{C}_\mathcal{O} = \{ x_R + jx_I : x_R, x_I \in [-1, 1] \}$ with $j^2 = -1$; this is simply a box with radius 1 around the square constellation.

BOX equalization solves the following relaxed version of the ML problem in (2):

$$\hat{s}_w^{\text{BOX}} = \arg \min_{\mathbf{w} \in \mathcal{O}^U} \| \mathbf{y}_w - \mathbf{H}_w z \|_2^2. \quad (8)$$

Since this equalization problem (8) is convex, it can be solved exactly using well-established numerical methods from convex optimization [27]. Furthermore, as shown recently in [23], [26], the BOX equalizer exhibits near-ML error-rate performance in the large-antenna limit, where we fix the BS-to-user antenna ratio $\delta = B/U$ so that $\delta > 1/2$ and by letting $B \rightarrow \infty$. In addition, the BOX equalizer does only need knowledge of the transmit constellation $\mathcal{O}$ but not of the noise variance $N_0$, which is in stark contrast to the MMSE equalizer.

Unfortunately, solving (8) exactly with conventional interior-point methods results in prohibitive complexity and requires high numerical precision, which prevents efficient hardware designs that use finite precision (fixed-point) arithmetic. In order to solve (8) at low complexity and in a hardware efficient manner, we propose a new algorithm in Section III.

E. Soft-Output Data Detection

From MMSE and BOX equalization, hard-output estimates can easily be obtained by element-wise slicing of the entries of $\hat{s}_w^{\text{MMSE}}$ and $\hat{s}_w^{\text{BOX}}$ onto the nearest constellation point as in (3), respectively. In systems that use forward error-correction, however, one is generally interested in soft-output detection [28]. From MMSE equalization where $\hat{s}_w = \hat{s}_w^{\text{MMSE}}$, LLR values are typically computed via the max-log approximation [21]

$$L_{w,i,b} = \rho_{w,i} \left( \min_{\mu \in \mathcal{O}_b^U} \left| \frac{\hat{s}_{w,i}}{\mu_{w,i}} - a \right| - \min_{\nu \in \mathcal{O}_b^U} \left| \frac{\hat{s}_{w,i}}{\mu_{w,i}} - a \right|^2 \right), \quad (9)$$

where the sets $\mathcal{O}_b^U$ and $\mathcal{O}_b^U$ contain the constellation symbols for which the $b$th bit is 0 and 1, respectively. For explicit MMSE detection, i.e., the approach discussed in Section II-C that computes $\mathbf{A}_w^{-1}$, the post-equalization signal-to-noise-and-interference-ratio (SINR) $\rho_{w,i}$ and the channel gain $\mu_{w,i}$ can be calculated exactly and in the following efficient way [21]. The SINR is calculated as $\rho_{w,i} = \mu_{w,i}/(1 - \mu_{w,i})$ and the channel gain $\mu_{w,i} = [\mathbf{A}_w^{-1}]_{ij}$, where $[\mathbf{A}_w^{-1}]_{ij}$ is the $i$th row of $\mathbf{A}_w^{-1}$ and $G_{w,i}$ is the $i$th column of $\mathbf{G}_w$.

However, for BOX equalization in Section II-D as well as for data detection algorithms that implicitly solve the MMSE detection problem [4], no efficient methods that exactly compute the SINR $\rho_{w,i}$ are known—this prevents a straightforward computation of the LLR values in (9). In Section III-C we propose an approximate way to compute $\rho_{w,i}$ and $\mu_{w,i}$, which enables us to generate approximate LLR values for such linear and non-linear equalizers.

III. FAST EQUALIZATION VIA COORDINATE DESCENT

While the solution to the implicit MMSE problem (4) can be computed (exactly or approximately) at moderate complexity using iterative conjugate gradient (CG) or Gauss-Seidel (GS) methods, see, e.g., [9], [13], [22], corresponding VLSI designs [10], [13] are unable to achieve high throughput, mainly due to a fairly complex algorithm structure, stringent data dependencies, or the need for high arithmetic precision. We next propose an alternative method to solve both the MMSE equalization (4) and BOX equalization (8) problems at low complexity and in a hardware friendly way.

A. Coordinate Descent (CD)

Coordinate descent (CD) [29] is a well-established iterative framework to exactly or approximately solve a large number of convex optimization problems using a series of simple, coordinate-wise updates. We first define the following function:

$$f(z_1, \ldots, z_U) = f(z) = \| \mathbf{y}_w - \mathbf{H}_w z \|_2^2 + g(z), \quad (10)$$

where $g(z)$ is a convex regularizer. It is now important to realize that both equalization problems (4) and (8) are equivalent to solving the BOXY equalization problem (8) of convex optimization (10) is equivalent to solving the MMSE equalization problem (4). By setting $g(z) = \chi(z) = 0$, for $z \in \mathcal{C}_\mathcal{O}$, where $\chi(z) = 0$ denotes the characteristic function that is zero if $z \in \mathcal{C}_\mathcal{O}$ and infinity otherwise, minimizing (10) is equivalent to solving the BOX equalization problem (8). CD-based equalization simply minimizes the function $f(z_1, \ldots, z_U)$ in (10) sequentially for each variable (or coordinate) $z_u, u = 1, \ldots, U$, in a round-robin fashion [30]. For more details on CD, see [29], [30] and the references therein. We next detail the CD algorithms for MMSE and BOX equalization.

3The performance of CD can often be improved by using a carefully-selected variable-update order [29]; our own experiments have shown that for MMSE and BOX equalization, a simple round-robin update scheme performs well and is easier to implement.
1) **CD-based MMSE Equalization**: Assume we want to find the \( u \)-th optimum value \( z_u \) for the MMSE equalization problem \((4)\), i.e., we seek to compute the solution to

\[
\hat{z}_u = \arg \min_{z_u \in \mathbb{C}} \| y_w - H_u z_u \|^2_2 + N_0 \| z_u \|^2_2,
\]  

(11)

where we hold all other values \( z_j, \forall j \neq u \), fixed. Since this is a quadratic problem, we can solve it in closed form by setting the gradient of the function \((10)\) with respect to the \( u \)-th component to zero:

\[
0 = \nabla_{u} f(z) = h_u^H (Hz - y) + N_0 z_u.
\]  

(12)

By decomposing \( Hz = h_u z_u + \sum_{j \neq u} h_j z_j \), we can solve \((12)\) for \( z_u \) to obtain the following closed-form expression:

\[
\hat{z}_u = \frac{1}{\| h_u \|^2_2 + N_0} h_u^H \left( y - \sum_{j \neq u} h_j z_j \right).
\]  

(13)

This expression is exactly the CD update rule for the \( u \)-th entry of \( z \). For every iteration, we can compute \((13)\) sequentially for each user \( u = 1, \ldots, U \), where we immediately re-use the new result \( \hat{z}_u \) for the \( u \)-th user in subsequent steps. We repeat this procedure for a total number of \( K \) iterations in order to obtain an estimate for \( \hat{z}_u^{(K)} \), where \( z_u^{(K)} \) is the end result of the above-described iterative process.

2) **CD-based BOX Equalization**: Analogously to CD-based MMSE equalization, we can derive the CD update rule for the BOX equalization problem \((8)\). Even though the characteristic function \( j^{\text{BOX}}(z) = \chi(z \in C_{\text{BOX}}) \) is not differentiable, a similar approach that uses subgradients (instead of gradients) enables one to derive the following closed-form expression \(3\):

\[
\tilde{z}_u = \text{proj}_{C_{\text{BOX}}} \left( \frac{1}{\| h_u \|^2_2} h_u^H \left( y - \sum_{j \neq u} h_j z_j \right) \right).
\]  

(14)

Here, \( \text{proj}_{C_{\text{BOX}}} (\cdot) \) is the orthogonal projection onto the convex polytope \( C_{\text{BOX}} \) and is given by

\[
\text{proj}_{C_{\text{BOX}}} (w) = \begin{cases} 
  w & \text{if } w \in C_{\text{BOX}} \\
  \text{arg min}_{q \in C_{\text{BOX}}} \| w - q \| & \text{if } w \notin C_{\text{BOX}}.
\end{cases}
\]  

(15)

In words, if the argument \( w \in \mathbb{C} \) is within the set \( C_{\text{BOX}} \), then the projection outputs \( w \); if \( w \) is outside the set \( C_{\text{BOX}} \), the projection outputs the value \( q \) that is closest to \( w \) within the set \( C_{\text{BOX}} \) in terms of the Euclidean distance. We emphasize that for many practically-relevant constellation sets \( O \), the projection \((15)\) can be carried out efficiently. For any QAM constellation, for example, we independently clip the real and imaginary part of \( w \) onto the interval \([-\alpha, +\alpha]\), where \( \alpha \) is the radius of the tightest box that covers the QAM constellation (see Section \(1\) for the details). For BPSK with \( O = \{-1, +1\} \), we clip the real part of \( w \) onto the interval \([-1, +1]\) and set the imaginary part to zero.\(^3\)

\(^3\)Orthogonal projections for PSK constellations sets are also possible. The development of efficient algorithms for PSK systems is left for future work.

---

**Algorithm 1** Optimized Coordinate Descent (OCD)

1. **inputs**: \( y \), \( H \), and \( N_0 \)
2. **initialization**: 
   3. \( r = y \) and \( z^{(0)} = 0^{U \times 1} \)
   4. MMSE mode: \( \alpha = N_0 \) and \( C = \mathbb{C} \)
   5. BOX mode: \( \alpha = 0 \) and \( C = C_{\text{BOX}} \)
3. **preprocessing**: 
   7. \( d_u^{-1} = (\| h_u \|^2_2 + \alpha)^{-1}, u = 1, \ldots, U \)
   8. \( p_u = d_u^{-1} \| h_u \|^2_2, u = 1, \ldots, U \)
4. **equalization**: 
   10. for \( k = 1, \ldots, K \)
   11. for \( u = 1, \ldots, U \)
   12. \( z_u^{(k)} = \text{proj}_{C_{\text{BOX}}} \left( d_u^{-1} h_u^H r + p_u z_u^{(k-1)} \right) \)
   13. \( \Delta z_u^{(k)} = z_u^{(k)} - z_u^{(k-1)} \)
   14. \( r \leftarrow r - h_u \Delta z_u^{(k)} \)
   15. end for
   16. end for
5. **outputs**: \( \hat{s}^{(K)} = [z_1^{(K)}, \ldots, z_U^{(K)}]^T \)

---

**B. Optimized Coordinate Descent (OCD)**

Instead of blindly computing the updates \((13)\) and \((14)\) for MMSE and BOX equalization, respectively, we perform preprocessing and algorithm restructuring in order to minimize the amount of (recurrent) operations during each of the \( k = 1, \ldots, K \) iterations. These optimizations entail no performance loss, i.e., both methods, OCD and CD, deliver exactly the same results. We refer to the resulting method as the optimized CD algorithm (short OCD), which is summarized in Algorithm 1. OCD supports both BOX and MMSE equalization and the individual optimization steps are as follows.\(^4\)

1) **Preprocessing**: To reduce the computational complexity, OCD precomputes certain key quantities that can be re-used during each of the \( k = 1, \ldots, K \) iterations. This preprocessing step not only results in significant complexity savings during the iterative process (compared to CD), but also simplifies our hardware implementation (see Section \(1\)). In particular, we precompute so-called (regularized) inverse squared column norms of \( H \), i.e., \( d_u^{-1} = (\| h_u \|^2_2 + \alpha)^{-1} \) for \( u = 1, \ldots, U \), with \( \alpha \geq 0 \), as well as regularized gains \( p_u = d_u^{-1} \| h_u \|^2_2 \) for \( u = 1, \ldots, U \). In MMSE mode, the regularization parameter is given by \( \alpha = N_0 \); in BOX mode, the regularization parameter is given by \( \alpha = 0 \), which yields \( p_u = 1 \), \( u = 1, \ldots, U \).

2) **Equalization**: In order to avoid recurrent operations during the equalization process, OCD performs incremental updates and re-uses intermediate quantities during each of the \( k = 1, \ldots, K \) iterations. In essence, we perform sequential updates on the so-called residual approximation vector, which is defined as

\[
r = y - \sum_{j=1}^{U} h_j z_j^{(k)}
\]  

(16)

\(^4\)The OCD algorithm proposed in the conference version of this paper \(1\) differs from the one presented here. The operations in OCD as proposed here have been restructured in order to (i) support MMSE as well as BOX equalization and (ii) reduce the hardware complexity.
at every algorithm iteration $k = 1, \ldots, K$ and for each user $u = 1, \ldots, U$. Note, however, that we do not recompute this residual approximation vector for every iteration and user from scratch. In contrary, we update the residual approximation vector in every iteration and for each user by first computing the symbol estimates $\hat{z}_u^{(k)}$ on line 12 of Algorithm 1. We then compute a so-called delta value $\Delta \hat{z}_u$ on line 13 which enables us to update the residual $r$ on line 14 without calculating the residual $\Delta r$ explicitly.

As mentioned above, OCD delivers exactly the same results as CD, but does so at significantly lower computational complexity. In fact, the original CD algorithm in Section III-A requires one complex-valued inner product and $U - 1$ complex scalar-by-vector multiplications per iteration $k$, whereas the proposed OCD algorithm requires only one inner product and one complex scalar-by-vector multiplication per iteration. More precisely, for MMSE equalization, CD requires $4BU^2 + 2U$ real-valued multiplications per iteration $k$, whereas OCD requires only $8BU + 4U$ real-valued multiplications. Hence, for a large number of BS antennas $B$, OCD requires roughly $U/2$ times lower complexity than CD per iteration.

C. LLR Approximation for OCD

To compute the LLR values (9) for MMSE and BOX equalization using OCD, we must resort to an approximation as we never explicitly compute the inverse $A_w^{-1}$. To this end, we use the approximation put forward in [11], [22] for SC-FDMA-based systems. For OFDM, this approach simplifies significantly and corresponds to approximating the channel gains by $\hat{\rho}_{w,i} = \tilde{d}_{w,i}^{-1}g_{w,i}$, where $\tilde{d}_{w,i}$ is the $i$th main diagonal of the Gram matrix $G_w$ at subcarrier $w$. Furthermore, the approach from [11], [22] applied to OFDM systems results in the following SINR approximation: $\tilde{\rho}_{w,i} = \mu_{w,i}/(1 - \mu_{w,i})$. We refer the interested reader to [22] for more details. As we will show next, this LLR approximation enables near-optimal performance in massive MU-MIMO systems with large BS-to-user-antenna ratios.
D. Error-Rate Performance

In order to assess the error-rate performance for the proposed OCD-BOX algorithm, we perform Monte-Carlo simulations in a coded 20 MHz MIMO-OFDM uplink system with 2048 subcarriers, where 1200 are used for data transmission as in LTE Advanced (LTE-A) \[31\]. We use 64-QAM with Gray mapping and a rate-3/4 turbo code. To account for spatial and frequency correlation, we generate channel matrices using the WINNER-Phase-2 model \[32\] with 7.8 cm antenna spacing as in \[11\], \[22\]. For channel decoding, we use a log-MAP turbo decoder. We report the packet error-rate, which is obtained by coding over one OFDM symbol with 1200 data subcarriers. The signal-to-noise-ratio (SNR) per bit in decibels, defined as

\[
10 \log_{10} \left( \frac{E_b}{N_0} \right) = 10 \log_{10} \left( \frac{\mathbb{E}[|s|^2]}{\mathbb{E}[|n|^2]} \right) 
\]

(17)

Figures 1 and 2 compare the packet error rate (PER) for OCD-BOX with other exact and approximate data-detection methods for massive MU-MIMO systems with various antenna configurations. In particular, we show PER results for Neumann-series detection \[7\], CG-based detection \[10\], and Gauss-Seidel (GS)-based detection \[13\]. We also include an exact linear MMSE equalizer as a reference. For all considered antenna configurations, OCD-BOX outperforms Neumann, CG, and GS detection for the same iteration count. We see that OCD with BOX equalization (OCD-BOX for short) achieves near-exact MMSE performance for only three iterations (\(K = 3\)) for 64 and 128 BS antennas, whereas \(K = 4\) is required for the “not-so-large” system with 32 BS antennas; lower values of \(K\) result in a high error floor. These results confirm that for larger BS-to-user-antenna ratios, approximate linear data detectors approach the performance of the MMSE detector. We note that for the considered antenna configurations, linear MMSE detection achieves near-ML performance \[7\].

Figures 2(a) \[2\] and 2(b) compare the PER for OCD-BOX against OCD with MMSE equalization (short OCD-MMSE). The performance of OCD-BOX is superior than that of OCD-MMSE, especially in the 32 BS antenna, 8 user case. In general, the performance difference is more pronounced for smaller BS-to-user-antenna ratios. This observation is in accordance to recent theoretical results \[23\], and can be addressed to the fact that the box constraint around the constellation is tighter than the quadratic penalty \(\gamma_{\text{MMSE}}(z) = N_0 \|z\|^2\) imposed by MMSE equalization.

We conclude by noting that for many modern wireless communication standards (such as LTE-A \[31\]) achieving a target PER of 10% is sufficient. The proposed OCD detector is able to meet this target performance at only a small SNR loss compared to the exact MMSE-based data detector.

IV. VLSI ARCHITECTURE

We now detail our VLSI architecture for OCD-based MMSE and BOX equalization. The architecture was designed and optimized using Xilinx Vivado HLS (version 2015.2.), which allows us to conveniently simulate, parameterize, and generate different OCD designs that support various antenna configurations at design time. At run-time, the proposed designs can be configured in terms of the numbers of supported users \(U\) and maximum number of iterations \(K\).

A. Architecture Overview

Figure 3 shows two high-level block diagrams of the proposed OCD architecture. The inputs of our architecture are the channel matrix \(H_{\text{in}}\), the residual error vector \(r\) (which is initialized to the received vector \(y_u\)), and the regularization parameter \(\alpha\), which we initialized to \(N_0\) and 0 for MMSE and BOX equalization, respectively. Our architecture supports two operation modes: (a) preprocessing (lines 6–8 of Algorithm 1) and (b) OCD-based qualification (lines 10–16). Preprocessing and equalization are carried out in a \(B\)-wide vector pipeline, i.e., we process \(B\)-dimensional vectors at a time. In the preprocessing mode, we compute the regularized inverse squared column norms \(d_u^{-1}, u = 1, \ldots, U\), as well as the regularized gains \(p_u, u = 1, \ldots, U\). In the equalization mode, we perform the iterations on lines 12–13 of Algorithm 1. In order to support these two operation modes without the need of redundant computation units, the processing pipeline shares the key building blocks used in both modes. In particular, both of the supported modes share the inner-product unit and the right-shift unit (highlighted in red in Figure 3). The inner product unit consists of \(B\) parallel complex-valued multipliers followed by a balanced adder tree. We use multiplexers at the input of the inner product unit, which enables us to switch between preprocessing and equalization on a per-clock cycle basis.

One of the main implementation challenges of the proposed OCD algorithm are data dependencies between successive iterations, which prevent traditional architecture pipelining. In particular, as it can be seen on line 14 of Algorithm 1 each OCD iteration updates the temporary vector \(r\) and the vector \(z_u^{(k+1)}\) given the previous vectors \(r\) and \(z_u^{(k)}\). Hence, in order to achieve high throughput, we deploy pipeline interleaving \[53\], i.e., we simultaneously process multiple subcarriers in a parallel and interleaved manner within the same architecture. For example, after performing an OCD iteration for the first subcarrier, we start an OCD iteration for the second subcarrier in the next clock cycle; we repeat this interleaving process until all pipeline stages are fully occupied. Our final architecture uses a total number of 24 pipeline stages, which enables our design to achieve up to 260 MHz in a Xilinx Virtex-7 FPGA (see Section IV for more details). We note that it is possible to achieve even higher clock frequencies by increasing the number of pipeline stages (especially for smaller small \(B\)); this approach, however, results in a significant hardware overhead.

B. Architecture and Fixed-point Optimization

In order to optimize the hardware efficiency of our architecture, we use fixed-point arithmetic throughout our design. We achieved a negligible implementation loss with 16bit precision with 11 fractional bit for most internal signals; see Figure 1 for the fixed-point (fp) performance. Our design has an implementation loss of less than 0.2 dB SNR (measured at a target PER of 10%) compared to floating-point performance for the considered scenarios, which is a result of the following two optimizations.
1) Inner-product unit: This unit first computes entry-wise products of two $B$-dimensional vectors and then, generates the final sum of these products. We use a balanced adder tree to compute the final sum and 36 bit adders to achieve sufficiently high arithmetic internal precision. During preprocessing, the inner-product unit computes $\|\mathbf{h}_u\|^2_2$ (line 7 of Algorithm 1); during equalization, the same unit computes $\mathbf{h}_u^H \mathbf{r}$ (line 12). As both of these terms are close to $B$ (for large values of $B$), we shift these terms by $b = \lceil \log_2(B) \rceil$ bits to the right in order to reduce the dynamic range. Since we shift $\|\mathbf{h}_u\|^2_2$ by $b$ to the right, when we compute the reciprocal value, $d_u^{-1} = (\|\mathbf{h}_u\|^2_2 + \alpha)^{-1}$, we effectively shift the reciprocal value $d_u^{-1}$ by $b$ bits to the left. In the inner-product unit, we also shift the term $\mathbf{h}_u^H \mathbf{r}$ by $b$ bits to the right. Consequently, we do not need to undo both of these shifts, as they cancel out during the multiplication on line 12 of Algorithm 1.

2) Reciprocal unit: This unit consists of two parts. The first part normalizes the input value to the range $[0, 1]$, which is accomplished using a leading-zero detector and programmable shift to the left. The second part generates a reciprocal value for the normalized input using a look-up table (LUT). We use a FPGA BRAM18 to implement a 18 bit, 2048 entry LUT, where the leading 11 bits of the normalized input value are used to point to the entry in the LUT that stores the associated normalized reciprocal. Finally, we denormalize the normalized reciprocal value by another left shift.

![Diagram](image)

(a) OCD preprocessing mode.

![Diagram](image)

(b) OCD iteration mode.

Fig. 3. High-level block diagram of the proposed OCD-based preprocessing and equalization pipeline. The pipeline is reconfigurable for various BS-antenna configurations at design time, and is able to perform preprocessing as well as MMSE or BOX equalization. The shared computation units between preprocessing and equalization are highlighted in red.

<table>
<thead>
<tr>
<th>Array size</th>
<th>$B = 32$</th>
<th>$B = 64$</th>
<th>$B = 128$</th>
</tr>
</thead>
<tbody>
<tr>
<td># of Slices</td>
<td>2873</td>
<td>6508</td>
<td>11094</td>
</tr>
<tr>
<td># of LUTs</td>
<td>6059</td>
<td>12588</td>
<td>23914</td>
</tr>
<tr>
<td># of FFs</td>
<td>10704</td>
<td>24801</td>
<td>43008</td>
</tr>
<tr>
<td># of DSP48s</td>
<td>198</td>
<td>390</td>
<td>774</td>
</tr>
<tr>
<td># of BRAM18s</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

Max. clock frequency | 261 MHz | 261 MHz | 258 MHz |

V. IMPLEMENTATION RESULTS AND COMPARISON

We now show FPGA implementation results and compare our design to the recently proposed data-detectors for massive MU-MIMO systems in [7], [10], [12], [13].

A. FPGA Implementation Results

We designed three different implementations for the following BS antenna configurations: $B = 32$, $B = 64$ and $B = 128$. For each configuration, we provide post place-and-route implementation results on a Xilinx Virtex-7 XC7VX690T FPGA. All our designs support $U \leq 32$ users and $K \leq 256$ OCD iterations; both of these parameters can be set at run-time.

The hardware complexity, resource utilization, and maximum clock frequency results are summarized in Table I. We note...
The processing latency increases roughly linearly with respect to $K$ and $U$. More specifically, the processing latency of this design is approximately $24(K+1)U + O$ clock cycles, where $O$ is the number of cycles required to flush the pipeline. Typically, 26 cycles are required to flush the pipeline, the exact value of $O$ depends on $B$. The (approximately) linear increase in $K$ can be seen in Table III and for $K = 3$, our design requires only 3.08 $\mu$s to produce its first equalized output.

### B. Comparison

Table IV compares OCD to other, recently proposed large-scale MIMO data detectors, namely the conjugate gradient (CG)-based detector [10], the Neumann-series detector [7], the Gauss-Seidel (GS) detector [13], and triangular approximate semidefinite relaxation (TASER) [12]. All of these detectors have been implemented on the same FPGA and for a 128 BS antenna, 8 user system. We see that for the same system configuration, OCD outperforms all other designs in terms of hardware efficiency, which we define as throughput per FPGA LUTs. Furthermore, our OCD detector achieves superior PER performance than the CG, Neumann, and GS detector (see Figs. 1(b) and 1(c)), which demonstrates the effectiveness of OCD. TASER, in contrast, achieves better error-rate performance for the considered antenna configuration, but only supports QPSK constellations. We note that the throughput of (approximate) linear detectors, such as the ones in [7], [10], [13], scales linearly in the number of bits $Q$ per symbol; for TASER, however, the throughput is limited by QPSK modulation, which prevents this detector to achieve comparable throughputs as the other approximate methods.

In summary, we see that OCD outperforms the next-best design (namely the CG-detector from [10]) by more than $2.6 \times$ in terms of hardware efficiency. The reasons for this advantage are due to the facts that (i) OCD can be implemented in a very regular and parallel manner and (ii) preprocessing requires significantly lower complexity compared to that of the other detectors that require the computation of the regularized Gram

8TASER achieves near-ML performance in “not-so-massive” MIMO systems, where the number of users is comparable to the number of BS antennas.

### TABLE II

<table>
<thead>
<tr>
<th>Main units</th>
<th># of Slices</th>
<th># of LUTs</th>
<th># of FFs</th>
<th># of DSP48s</th>
<th># of BRAM18s</th>
</tr>
</thead>
<tbody>
<tr>
<td>$B = 32$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>r update unit</td>
<td>256 (8.91%)</td>
<td>1024 (16.9%)</td>
<td>0 (0%)</td>
<td>0 (0%)</td>
<td>0 (0%)</td>
</tr>
<tr>
<td>Inner-product unit</td>
<td>811 (28.2%)</td>
<td>1045 (17.3%)</td>
<td>2416 (22.6%)</td>
<td>96 (48.5%)</td>
<td>0 (0%)</td>
</tr>
<tr>
<td>$h_u \Delta u$, scaling unit</td>
<td>265 (9.22%)</td>
<td>249 (4.11%)</td>
<td>1137 (10.5%)</td>
<td>96 (48.5%)</td>
<td>0 (0%)</td>
</tr>
<tr>
<td>Miscellaneous</td>
<td>1541 (53.6%)</td>
<td>3741 (61.74%)</td>
<td>7151 (66.8%)</td>
<td>6 (3.0%)</td>
<td>2 (100%)</td>
</tr>
<tr>
<td>Total</td>
<td>2873 (100%)</td>
<td>6059 (100%)</td>
<td>10704 (100%)</td>
<td>198 (100%)</td>
<td>2 (100%)</td>
</tr>
<tr>
<td>$B = 64$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>r update unit</td>
<td>512 (7.87%)</td>
<td>2048 (16.3%)</td>
<td>0 (0%)</td>
<td>0 (0%)</td>
<td>0 (0%)</td>
</tr>
<tr>
<td>Inner-product unit</td>
<td>1627 (25.0%)</td>
<td>2006 (15.9%)</td>
<td>5776 (23.3%)</td>
<td>192 (49.2%)</td>
<td>0 (0%)</td>
</tr>
<tr>
<td>$h_u \Delta u$, scaling unit</td>
<td>485 (7.45%)</td>
<td>505 (4.01%)</td>
<td>2161 (8.71%)</td>
<td>192 (49.2%)</td>
<td>0 (0%)</td>
</tr>
<tr>
<td>Miscellaneous</td>
<td>3884 (59.7%)</td>
<td>8029 (63.8%)</td>
<td>16864 (68.0%)</td>
<td>6 (1.6%)</td>
<td>2 (100%)</td>
</tr>
<tr>
<td>Total</td>
<td>6508 (100%)</td>
<td>12588 (100%)</td>
<td>24801 (100%)</td>
<td>390 (100%)</td>
<td>2 (100%)</td>
</tr>
<tr>
<td>$B = 128$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>r update unit</td>
<td>1024 (9.23%)</td>
<td>4096 (17.1%)</td>
<td>0 (0%)</td>
<td>0 (0%)</td>
<td>0 (0%)</td>
</tr>
<tr>
<td>Inner-product unit</td>
<td>3447 (31.1%)</td>
<td>4109 (17.2%)</td>
<td>11676 (27.0%)</td>
<td>384 (49.6%)</td>
<td>0 (0%)</td>
</tr>
<tr>
<td>$h_u \Delta u$, scaling unit</td>
<td>1955 (17.6%)</td>
<td>5120 (21.4%)</td>
<td>4211 (9.72%)</td>
<td>384 (49.6%)</td>
<td>0 (0%)</td>
</tr>
<tr>
<td>Miscellaneous</td>
<td>4666 (42.1%)</td>
<td>10589 (44.3%)</td>
<td>27421 (63.3%)</td>
<td>6 (0.8%)</td>
<td>2 (100%)</td>
</tr>
<tr>
<td>Total</td>
<td>11094 (100%)</td>
<td>23914 (100%)</td>
<td>43308 (100%)</td>
<td>774 (100%)</td>
<td>2 (100%)</td>
</tr>
</tbody>
</table>

### TABLE III

<table>
<thead>
<tr>
<th>Throughput and Latency on a Xilinx Virtex-7 XC7VX690T FPGA for $K$ Iterations and 64-QAM, and 128 BS and 8 User Antennas</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K$</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
</tbody>
</table>

that there is no particular critical path in all our designs as Vivado HLS evenly optimizes the delays among all pipeline stages. A detailed area breakdown of the main units is shown in Table III. The “$r$ update unit” corresponds to the output adder in Figure 3(b); the “inner-product unit” corresponds to the unit that computes $h_u^H h_u$ and $h_u^H r$ in Figure 3(a) and Figure 3(b), respectively; the “$h_u \Delta u$, scaling unit” corresponds to the scaling block in Figure 3(a); all remaining circuitry has been flattened by Vivado HLS and is subsumed in “miscellaneous.” Since the proposed architecture performs operations on $B$-dimensional vectors, the resource utilization (excluding the BRAMs) scales linearly with $B$. Since the quantities $H_w$ and $y_w$ are assumed to be stored in external memories, our OCD architecture only uses two BRAM18s: one for the reciprocal LUT and one to store the regularized channel gains $p_u$, $u = 1, \ldots, U$.

The maximum achievable throughput as well as the processing latency are shown in Table III. We see that the throughput only depends on the maximum iteration number $K$ and the clock frequency, but does not depend on $U$. The reason is because the number of bits per subcarrier and the number of clock cycles required to process 24 subcarriers grows linearly with respect to $U$. For example, doubling $U$ doubles the number of bits per subcarrier. However, since the number of OCD updates is $K U$, the number of required clock cycles also doubles; this results in a constant throughput. For $K = 3$ iterations, which was shown in Figure 1 to achieve near-optimal performance, our design achieves 376 Mb/s. Hence, the use of only three parallel instances (to process subcarriers in parallel) would easily exceed 1.1 Gb/s, while consuming less than 65% of the FPGA’s BRAM18s (cf. Table IV).
matrix $A_{\text{eqv}}$, which can be a significant burden in massive MU-MIMO-OFDM systems.

VI. CONCLUSIONS

We have proposed a novel coordinate descent (CD)-based data detector, called optimized CD (OCD), for massive MU-MIMO systems that use orthogonal frequency division multiplexing (OFDM). The proposed OCD detector enables high-performance linear MMSE and non-linear box-constrained data detection using a simple, parallel VLSI architecture that requires low hardware complexity. Our FPGA reference design achieves 376 Mb/s for a 128 BS antenna, 8 user system, and substantially outperforms existing approximate linear data-detection methods in terms of hardware efficiency and/or error-rate performance. Our results show that OCD enables realistic OFDM-based massive MU-MIMO systems to support tens of users communicating with hundreds of BS antennas, while achieving high throughput at low implementation costs.

There are many avenues for future work. OCD can also be used for linear and non-linear precoding in the massive MU-MIMO downlink; a corresponding study is part of ongoing work. Computing exact soft-output values for OCD-based detection (for MMSE and BOX equalization) is an interesting open research problem. Finally, accelerated CD algorithms have been proposed recently [34]; such methods may lead to even faster convergence and hence, could enable higher throughput at the same error-rate performance when implemented in VLSI.

VII. ACKNOWLEDGMENTS

C. Studer would like to thank Tom Goldstein, Charles Jeon, Shahriar Shahabuddin for insightful discussions on the box-constrained equalization method. The work of M. Wu and J. R. Cavallaro was supported in part by Xilinx Inc., and by the US National Science Foundation (NSF) under grants ECCS-1408370, CNS-1265332, and ECCS-1232274. The work of C. Studer was supported in part by Xilinx Inc. and by the US NSF under grants ECCS-1408006 and CCF-1535897.

REFERENCES


TABLE IV

Comparison of 128 × 8 data detectors for massive MU-MIMO system on a Xilinx Virtex-7 XC7VX690T FPGA

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance</td>
<td>near-MMSE</td>
<td>near-MMSE</td>
<td>near-MMSE</td>
<td>near-MMSE</td>
<td>near-MMSE</td>
</tr>
<tr>
<td>Highest modulation</td>
<td>64-QAM</td>
<td>64-QAM</td>
<td>64-QAM</td>
<td>QPSK</td>
<td>64-QAM</td>
</tr>
<tr>
<td>Iteration count $K$</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td># of slices</td>
<td>1,094 (1.0%)</td>
<td>4824 (45%)</td>
<td>n.a.</td>
<td>4350 (4.0%)</td>
<td>11,094 (10%)</td>
</tr>
<tr>
<td># of LUTs</td>
<td>3,324 (0.8%)</td>
<td>148,797 (34%)</td>
<td>18,976 (4.3%)</td>
<td>13,779 (3.2%)</td>
<td>23,914 (5.5%)</td>
</tr>
<tr>
<td># of FFs</td>
<td>3,878 (0.4%)</td>
<td>161,934 (19%)</td>
<td>15,864 (1.8%)</td>
<td>6,857 (0.8%)</td>
<td>43,008 (4.9%)</td>
</tr>
<tr>
<td># of DSP48s</td>
<td>33 (0.9%)</td>
<td>1016 (28%)</td>
<td>232 (6.3%)</td>
<td>168 (5.7%)</td>
<td>774 (21.5%)</td>
</tr>
<tr>
<td>Maximum clock frequency [MHz]</td>
<td>412</td>
<td>317</td>
<td>309</td>
<td>225</td>
<td>258</td>
</tr>
<tr>
<td>Latency [clock cycles]</td>
<td>951</td>
<td>196</td>
<td>n.a.</td>
<td>72</td>
<td>795</td>
</tr>
<tr>
<td>Maximum throughput [Mb/s]</td>
<td>20</td>
<td>621</td>
<td>48</td>
<td>50</td>
<td>376</td>
</tr>
<tr>
<td>Throughput/LUTs</td>
<td>6,017</td>
<td>4,173</td>
<td>2,530</td>
<td>3,629</td>
<td>15,597</td>
</tr>
</tbody>
</table>

*The method uses a special Neumann-series initializer followed by one GS iteration.*
Michael Wu received his B.S. degree from Franklin W. Olin College of Engineering in May of 2007 and his M.S. degree from Rice University in May of 2010, both in Electrical and Computer Engineering. He expects to complete his Ph.D. at Rice University in Electrical and Computer Engineering in 2016 and is currently a staff engineer at Xilinx Inc., San Jose, CA. His research interests are wireless algorithms, software defined radio on GPU, FPGA, other parallel architectures, and high performance wireless receiver designs.

Joseph R. Cavallaro (S’78, M’82, SM’05, F’15) received the B.S. degree from the University of Pennsylvania, Philadelphia, Pa, in 1981, the M.S. degree from Princeton University, Princeton, NJ, in 1982, and the Ph.D. degree from Cornell University, Ithaca, NY, in 1988, all in electrical engineering. From 1981 to 1983, he was with AT&T Bell Laboratories, Holmdel, NJ. In 1988, he joined the faculty of Rice University, Houston, TX, where he is currently a Professor of electrical and computer engineering. His research interests include computer arithmetic, DSP, GPU, FPGA, and VLSI architectures for applications in wireless communications. During the 1996–1997 academic year, he served at the National Science Foundation as Director of the Prototyping Tools and Methodology Program. He was a Nokia Foundation Fellow and a Visiting Professor at the University of Oulu, Finland in 2005 and continues his affiliation there as an Adjunct Professor. He is currently the Director of the Center for Multimedia Communication at Rice University. He is a Fellow of the IEEE and a Member of the IEEE SPS TC on Design and Implementation of Signal Processing Systems and the Chair-Elect of the IEEE CAS TC on Circuits and Systems for Communications. He is currently an Associate Editor of the IEEE Transactions on Signal Processing, the IEEE Signal Processing Letters, and the Journal of Signal Processing Systems. He was Co-chair of the 2004 Signal Processing for Communications Symposium at the IEEE Global Communications Conference and General/Program Co-chair of the 2003, 2004, and 2011 IEEE International Conference on Application-Specific Systems, Architectures and Processors (ASAP), General/Program Co-chair for the 2012, 2014 ACM/IEEE GLSVLSI, Finance Chair for the 2013 IEEE GlobalSIP conference, and TPC Co-Chair of the 2016 IEEE SiPS workshop. He was a member of the IEEE CAS Society Board of Governors during 2014.
Christoph Studer (S’06, M’10, SM’14) received his Ph.D. degree in Information Technology and Electrical Engineering from ETH Zurich in 2009. In 2005, he was a Visiting Researcher with the Smart Antennas Research Group at Stanford University. From 2006 to 2009, he was a Research Assistant in both the Integrated Systems Laboratory and the Communication Technology Laboratory (CTL) at ETH Zurich. From 2009 to 2012, Dr. Studer was a Postdoctoral Researcher at CTL, ETH Zurich, and the Digital Signal Processing Group at Rice University. In 2013, he has held the position of Research Scientist at Rice University. Since 2014, Dr. Studer is an Assistant Professor at Cornell University and an Adjunct Assistant Professor at Rice University. Dr. Studer’s research interests include the design of very large-scale integration (VLSI) circuits, as well as wireless communications, signal and image processing, and convex optimization.

Dr. Studer received ETH Medals for his M.S. and Ph.D. theses, and he was the winner of the Student Paper Contest of the 2007 Asilomar Conf. on Signals, Systems, and Computers, received a Best Student Paper Award of the 2008 IEEE Int. Symp. on Circuits and Systems (ISCAS), and shared the best Live Demonstration Award at the IEEE ISCAS in 2013. In addition, Dr. Studer shared the Swisscom/ICTnet Innovations Award in both 2010 and 2013.