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 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 $2.6\times$ 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[1], 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

MW and JRC are with the Department of ECE, Rice University, Houston, TX; e-mail: {mbw2,cavallar}@rice.edu

MW and CD are with Xilinx Inc., San Jose, CA; e-mail: {miwu, chris.dick}@xilinx.com

CS is with the School of ECE, Cornell University, Ithaca, NY; e-mail: studer@cornell.edu

A short version of this paper for a single-carrier frequency-division multiple access (SC-FDMA) massive MU-MIMO systems has been presented at the IEEE International Symposium on Circuits and Systems (ISCAS) [1].

[1]SC-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 [14].

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 kth row and ℓth column of the matrix $\mathbf{A}$; the kth entry of a column vector $\mathbf{a}$ is denoted by $a_k = [\mathbf{a}]_k$. The ℓ_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 algorithm 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 U single-antenna user terminals send data *simultaneously* to a BS with $B \gg U$ antennas over W subcarriers. Each user $i = 1, \ldots, U$ 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., $\mathbb{E}\big[|s|^2\big] = 1$ 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.[2] Under these assumptions, the FD input-output relation on the wth subcarrier is commonly modeled as [17]

$$\mathbf{y}_w = \mathbf{H}_w \mathbf{s}_w + \mathbf{n}_w, \quad w = 1, \ldots, W, \tag{1}$$

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

[2]These assumptions are common in the MIMO-OFDM literature [16].

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]

$$\tilde{\mathbf{s}}_w^{\text{ML}} = \arg\min_{\mathbf{z} \in \mathcal{O}^U} \|\mathbf{y}_w - \mathbf{H}_w \mathbf{z}\|_2^2. \tag{2}$$

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 $\mathbf{z} \in \mathcal{O}^U$ in (2), which enables the efficient computation of an estimate $\tilde{\mathbf{s}}$ that is (hopefully) close to the ML solution. The estimate $\tilde{\mathbf{s}}$ can then either be sliced element-wise onto the nearest constellation point in $\mathcal{O}$ as follows:

$$\hat{s}_i = \arg\min_{z \in \mathcal{O}} |[\tilde{\mathbf{s}}]_i - z|, \quad i = 1, \ldots, U, \tag{3}$$

which is known as hard-output data detection, or 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 $\delta = B/U$ (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 $\mathbf{z} \in \mathcal{O}^U$ in the ML problem (2) to the U-dimensional complex space $\mathbf{z} \in \mathbb{C}^U$, and to include a quadratic penalty function. In particular, MMSE equalization solves the following regularized least-squares problem [10], [22]:

$$\tilde{\mathbf{s}}_w^{\text{MMSE}} = \arg\min_{\mathbf{z} \in \mathbb{C}^U} \|\mathbf{y}_w - \mathbf{H}_w \mathbf{z}\|_2^2 + N_0 \|\mathbf{z}\|_2^2. \tag{4}$$

Since the objective function in (4) is quadratic in $\mathbf{z}$, 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 $\mathbf{A}_w = \mathbf{G}_w + N_0 \mathbf{I}_U$ with $\mathbf{G}_w = \mathbf{H}_w^H \mathbf{H}_w$ and the matched filter vector $\tilde{\mathbf{s}}_w^{\text{MF}} = \mathbf{H}_w^H \mathbf{y}_w$. Then, the MMSE estimate in (4) is computed as

$$\tilde{\mathbf{s}}_w^{\text{MMSE}} = \mathbf{A}_w^{-1} \tilde{\mathbf{s}}_w^{\text{MF}}. \tag{5}$$

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 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 $\mathbf{z} \in \mathcal{O}^U$ to the convex polytope $\mathcal{C}_\mathcal{O}$ around the constellation set $\mathcal{O}$, which is formally defined as follows:

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

For example, the convex polytope $\mathcal{C}_{\text{QPSK}}$ for QPSK with[3]

$$\mathcal{O} = \{+1+j, +1-j, -1+j, -1-j\} \quad (7)$$

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

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

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

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 $B \to \infty$. In addition, the BOX equalizer does only need knowledge of the transmit constellation $\mathcal{O}$ but not of the noise variance N_0.

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 way, 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 $\tilde{\mathbf{s}}_w^{\text{MMSE}}$ and $\tilde{\mathbf{s}}_w^{\text{BOX}}$ 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 $\tilde{\mathbf{s}}_w = \tilde{\mathbf{s}}_w^{\text{MMSE}}$, LLR values are typically computed via the max-log approximation [21]

$$L_{w,i,b} = \rho_{w,i} \left(\min_{a \in \mathcal{O}_b^0} \left| \frac{[\tilde{\mathbf{s}}_w]_i}{\mu_{w,i}} - a \right|^2 - \min_{a \in \mathcal{O}_b^1} \left| \frac{[\tilde{\mathbf{s}}_w]_i}{\mu_{w,i}} - a \right|^2 \right), \quad (9)$$

where the sets $\mathcal{O}_b^0$ and $\mathcal{O}_b^1$ contain the constellation symbols for which the bth 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 is $\mu_{w,i} = [\mathbf{A}_w]_i^H [\mathbf{G}_w]_i$, where $[\mathbf{A}_w]_i$ is the ith row of $\mathbf{A}_w^{-1}$ and $[\mathbf{G}_w]_i$ is the ith 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 generate $\rho_{w,i}$ and $\mu_{w,i}$, which enables us to compute 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 equalizaton (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:

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

where $g(\mathbf{z})$ is a convex regularizer. It is now important to realize that both equalization problems (4) and (8) are special cases when minimizing (10). In fact, by setting $g^{\text{MMSE}}(\mathbf{z}) = N_0 \|\mathbf{z}\|_2^2$, minimizing (10) is equivalent to solving the MMSE equalization problem (4). By setting $g^{\text{BOX}}(\mathbf{z}) = \chi(\mathbf{z} \in \mathcal{C}_\mathcal{O})$, where $\chi(\mathbf{z} \in \mathcal{C}_\mathcal{O})$ denotes the characteristic function that is zero if $\mathbf{z} \in \mathcal{C}_\mathcal{O}$ 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.[4] For more details on CD, see [29], [30] and the references therein. We next detail the CD algorithms for MMSE and BOX equalization.

[3] We note that this constellation is not normalized to unit expected power.

[4] The 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 uth optimum value z_u for the MMSE equalization problem (4), i.e., we seek to compute the solution to

$$\hat{z}_u = \arg\min_{z_u \in \mathbb{C}} \|\mathbf{y}_w - \mathbf{H}_w \mathbf{z}\|_2^2 + N_0 \|\mathbf{z}\|_2^2, \tag{11}$$

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 uth component to zero:

$$0 = \nabla_u f(\mathbf{z}) = \mathbf{h}_u^H (\mathbf{H}\mathbf{z} - \mathbf{y}) + N_0 z_u. \tag{12}$$

By decomposing $\mathbf{H}\mathbf{z} = \mathbf{h}_u z_u + \sum_{j \neq u} \mathbf{h}_j z_j$, we can solve (12) for z_u to obtain the following closed-form expression:

$$\hat{z}_u = \frac{1}{\|\mathbf{h}_u\|_2^2 + N_0} \mathbf{h}_u^H \left(\mathbf{y} - \sum_{j \neq u} \mathbf{h}_j z_j \right). \tag{13}$$

This expression is exactly the CD update rule for the uth entry of $\mathbf{z}$. 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 uth user in subsequent steps. We repeat this procedure for a total number of K iterations in order to obtain an estimate for $\tilde{\mathbf{s}}^{\text{MMSE}} = \mathbf{z}^{(K)}$, where $\mathbf{z}^{(K)}$ 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 update rule for the BOX equalization problem (8). Even though the characteristic function $g^{\text{BOX}}(\mathbf{z}) = \chi(\mathbf{z} \in \mathcal{C}_{\mathcal{O}})$ is not differentiable, a similar approach that uses subgradients (instead of gradients) enables one to derive the following closed-form expression [30]:

$$\hat{z}_u = \text{proj}_{\mathcal{C}_{\mathcal{O}}} \left(\frac{1}{\|\mathbf{h}_u\|_2^2} \mathbf{h}_u^H \left(\mathbf{y} - \sum_{j \neq u} \mathbf{h}_j z_j \right) \right). \tag{14}$$

Here, $\text{proj}_{\mathcal{C}_{\mathcal{O}}}(\cdot)$ is the orthogonal projection onto the convex polytope $\mathcal{C}_{\mathcal{O}}$ and is given by

$$\text{proj}_{\mathcal{C}_{\mathcal{O}}}(w) = \begin{cases} w & \text{if } w \in \mathcal{C}_{\mathcal{O}} \\ \arg\min_{q \in \mathcal{C}_{\mathcal{O}}} |w - q| & \text{if } w \notin \mathcal{C}_{\mathcal{O}}. \end{cases} \tag{15}$$

In words, if the argument $w \in \mathbb{C}$ is within the set $\mathcal{C}_{\mathcal{O}}$, then the projection outputs w; if w is outside the set $\mathcal{C}_{\mathcal{O}}$, the projection outputs the value q that is closest to w within the set $\mathcal{C}_{\mathcal{O}}$ in terms of the Euclidean distance. We emphasize that for many practically-relevant constellation sets $\mathcal{O}$, 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 α is the radius of the tightest box that covers the QAM constellation (see Section II-D for the details). For BPSK with $\mathcal{O} = \{-1, +1\}$, we clip the real part of w onto the interval $[-1, +1]$ and set the imaginary part to zero.[5]

[5]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)

inputs: $\mathbf{y}$, $\mathbf{H}$, and N_0
initialization:
  $\mathbf{r} = \mathbf{y}$ and $\mathbf{z}^{(0)} = \mathbf{0}^{U \times 1}$
  MMSE mode: $\alpha = N_0$ and $\mathcal{C} = \mathbb{C}$
  BOX mode: $\alpha = 0$ and $\mathcal{C} = \mathcal{C}_{\mathcal{O}}$
preprocessing:
  $d_u^{-1} = (\|\mathbf{h}_u\|_2^2 + \alpha)^{-1}$, $u = 1, \ldots, U$
  $p_u = d_u^{-1} \|\mathbf{h}_u\|_2^2$, $u = 1, \ldots, U$
equalization:
for $k = 1, \ldots, K$ **do**
  for $u = 1, \ldots, U$ **do**
    $z_u^{(k)} = \text{proj}_{\mathcal{C}} \left(d_u^{-1} \mathbf{h}_u^H \mathbf{r} + p_u z_u^{(k-1)} \right)$
    $\Delta z_u^{(k)} = z_u^{(k)} - z_u^{(k-1)}$
    $\mathbf{r} \leftarrow \mathbf{r} - \mathbf{h}_u \Delta z_u^{(k)}$
  end for
end for
outputs: $\tilde{\mathbf{s}} = [z_1^{(K)}, \ldots, z_U^{(K)}]^T$

B. Optimized Coordinate Descent (OCD)

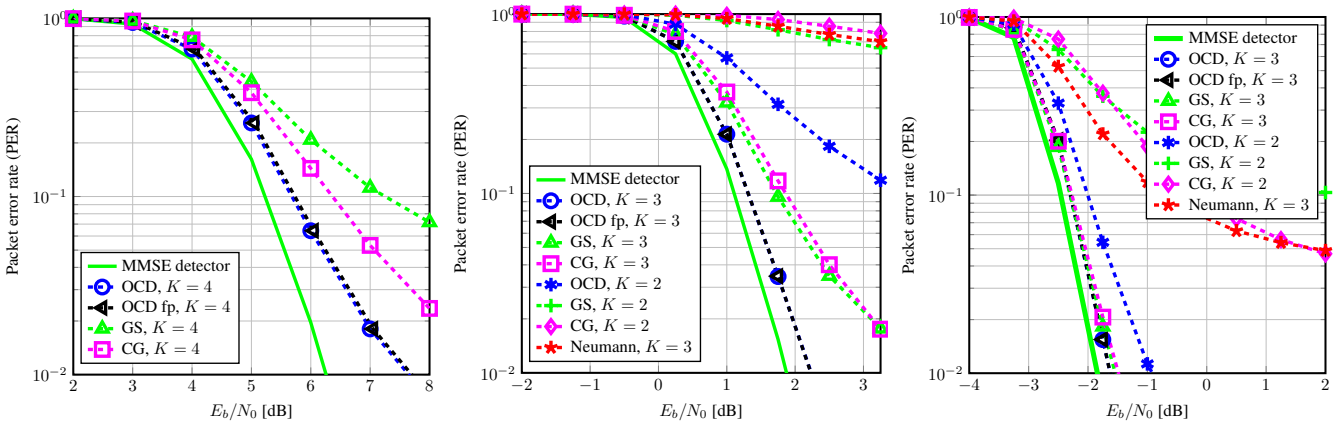
Instead of blindly computing the updates (13) and (14) for MMSE and BOX equalization, respectively, we perform preprocessing and algorithm restructuring in order to minimize the amount of (recurrent) operations during each of the $k = 1, \ldots, K$ iterations. These optimizations entail *no* performance 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.[6]

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 IV). In particular, we precompute so-called (regularized) inverse squared column norms of $\mathbf{H}$, i.e., $d_u^{-1} = (\|\mathbf{h}_u\|_2^2 + \alpha)^{-1}$ for $u = 1, \ldots, U$, with $\alpha \geq 0$, as well as regularized gains $p_u = d_u^{-1} \|\mathbf{h}_u\|_2^2$ 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

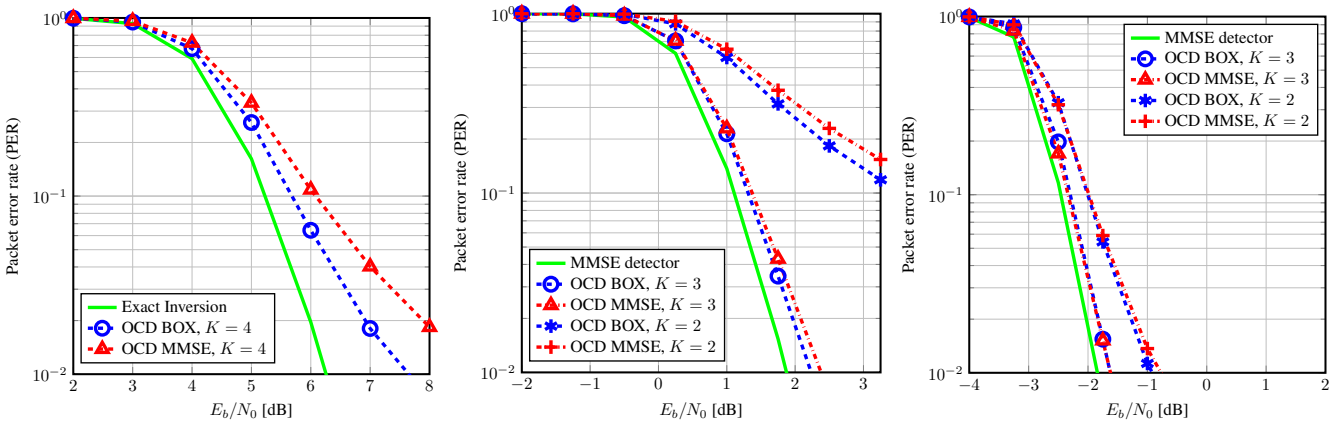
$$\mathbf{r} = \mathbf{y} - \sum_{j=1}^{U} \mathbf{h}_j z_j^{(k)} \tag{16}$$

[6]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.



(a) 32 BS antennas and 8 users. (b) 64 BS antennas and 8 users. (c) 128 BS antennas and 8 users.

Fig. 1. Packet error rate (PER) for a massive MU-MIMO-OFDM system ("fp" denotes fixed-point performance). Optimized coordinate descent (OCD) with box-constrained equalization achieves close-to-MMSE PER performance and outperforms the other three approximate equalization methods [7], [10], [13].



(a) 32 BS antennas and 8 users. (b) 64 BS antennas and 8 users. (c) 128 BS antennas and 8 users.

Fig. 2. Packet error rate (PER) for a massive MU-MIMO-OFDM system. BOX equalization outperforms MMSE equalization, especially for systems with a smaller BS-to-user antenna ratio. Furthermore, both approximate equalization methods achieve near-exact MMSE performance for a small number of iterations.

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 approximate 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 $z_u^{(k)}$ on line 12 of Algorithm 1. We then compute a so-called delta value $\Delta z_u^{(k)}$ on line 13, which enables us to update the residual $\mathbf{r}$ on line 14 without calculating the residual (16) 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. More precisely, for MMSE equalization, CD requires $4BU^2 + 2U$ real-valued multiplications[7] 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.

[7]We count 4 real-valued multiplications per complex-valued multiplication.

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 $\mathbf{A}_w^{-1}$. 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 $\tilde{\mu}_{w,i} = d_{w,i}^{-1} g_{w,i}$, where $d_{w,i}^{-1}$ is the ith regularized inverse squared column norm of $\mathbf{H}_w$ and $g_{i,w}$ is the entry in the ith main diagonal of the Gram matrix $\mathbf{G}_w$ at subcarrier w. Furthermore, the approach from [11], [22] applied to OFDM systems results in the following SINR approximation: $\tilde{\rho}_{w,i} = \tilde{\mu}_{w,i}/(1-\tilde{\mu}_{w,i})$. 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

$$10\log_{10}\left(\frac{E_{\mathrm{b}}}{N_0}\right) = 10\log_{10}\left(\frac{\mathbb{E}[\|\mathbf{s}\|^2]}{Q\mathbb{E}[\|\mathbf{n}\|^2]}\right). \qquad (17)$$

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(b), and 2(c) 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 more accurate than the quadratic penalty $g^{\mathrm{MMSE}}(\mathbf{z}) = N_0\|\mathbf{z}\|_2^2$ 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.

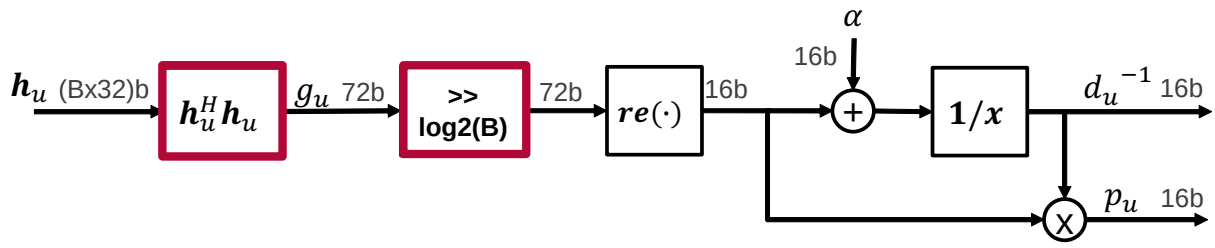
A. Architecture Overview

Figure 3 shows two high-level block diagrams of the proposed OCD architecture. The inputs of our architecture are the channel matrix $\mathbf{H}_w$, the residual error vector $\mathbf{r}$ (which is initialized to the received vector $\mathbf{y}_w$), and the regularization parameter α, 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 qualization (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.

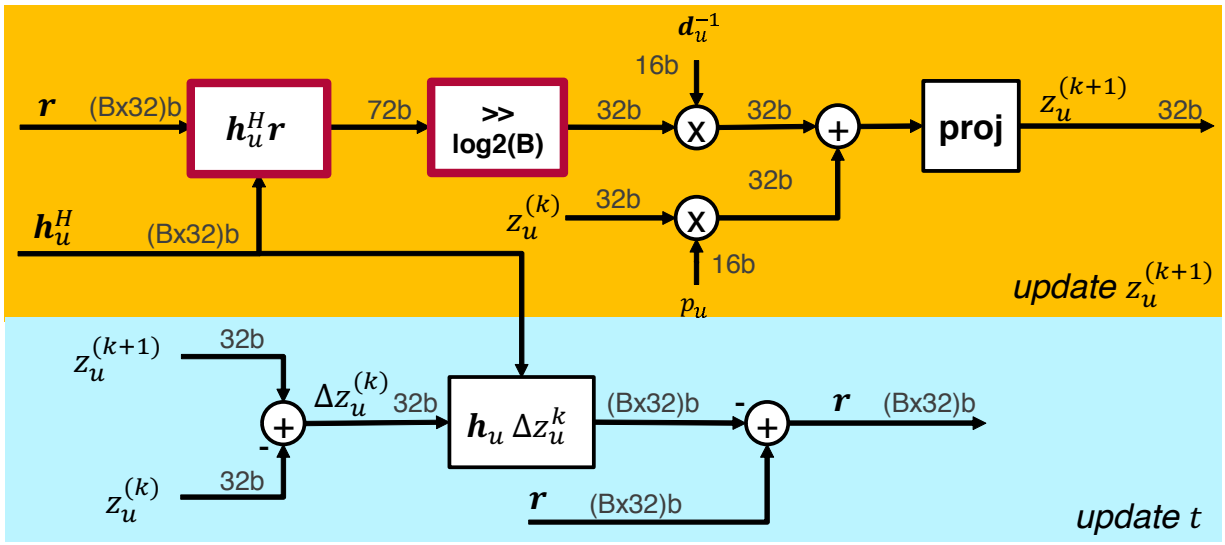
One of the main implementation challenges of the proposed OCD algorithm are data dependencies between successive iterations, which prevent traditional architecture pipelining. In particular, as it can be seen on line 14 of Algorithm 1, each OCD iteration updates the temporary vector $\mathbf{r}$ and the vector $\mathbf{z}_u^{(k+1)}$ given the previous vectors $\mathbf{r}$ and $\mathbf{z}_u^{(k)}$. Hence, in order to achieve high throughput, we deploy *pipeline interleaving* [33], 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 V 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 16 bit 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.



(a) OCD preprocessing mode.



(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.

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 $[0.5, 1]$, 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.

TABLE I
IMPLEMENTATION RESULTS ON A XILINX VIRTEX-7 XC7VX690T FPGA FOR DIFFERENT BS ANTENNA NUMBERS

Array size	$B = 32$	$B = 64$	$B = 128$
# of Slices	2 873	6 508	11 094
# of LUTs	6 059	12 588	23 914
# of FFs	10 704	24 801	43 008
# of DSP48s	198	390	774
# of BRAM18s	2	2	2
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].

TABLE II
AREA BREAKDOWN ON A XILINX VIRTEX-7 XC7VX690T FPGA FOR DIFFERENT BS ANTENNA NUMBERS

	Main units	# of Slices	# of LUTs	# of FFs	# of DSP48s	# of BRAM18s
$B = 32$	$\mathbf{r}$ update unit	256 (8.91%)	1 024 (16.9%)	0 (0%)	0 (0%)	0 (0%)
	Inner-product unit	811 (28.2%)	1 045 (17.3%)	2 416 (22.6%)	96 (48.5%)	0 (0%)
	$\mathbf{h}_u \Delta z_u$ scaling unit	265 (9.22%)	249 (4.11%)	1 137 (10.6%)	96 (48.5%)	0 (0%)
	Miscellaneous	1 541 (53.6%)	3 741 (61.74%)	7 151 (66.8%)	6 (3.0%)	2 (100%)
	Total	2 873 (100%)	6 059 (100%)	10 704 (100%)	198 (100%)	2 (100%)
$B = 64$	$\mathbf{r}$ update unit	512 (7.87%)	2 048 (16.3%)	0 (0%)	0 (0%)	0 (0%)
	Inner-product unit	1 627 (25.0%)	2 006 (15.9%)	5 776 (23.3%)	192 (49.2%)	0 (0%)
	$\mathbf{h}_u \Delta z_u$ scaling unit	485 (7.45%)	505 (4.01%)	2 161 (8.71%)	192 (49.2%)	0 (0%)
	Miscellaneous	3 884 (59.7%)	8 029 (63.8%)	16 864 (68.0%)	6 (1.6%)	2 (100%)
	Total	6 508 (100%)	12 588 (100%)	24 801 (100%)	390 (100%)	2 (100%)
$B = 128$	$\mathbf{r}$ update unit	1 024 (9.23%)	4 096 (17.1%)	0 (0%)	0 (0%)	0 (0%)
	Inner-product unit	3 447 (31.1%)	4 109 (17.2%)	11 676 (27.0%)	384 (49.6%)	0 (0%)
	$\mathbf{h}_u \Delta z_u$ scaling unit	1 955 (17.6%)	5 120 (21.4%)	4 211 (9.72%)	384 (49.6%)	0 (0%)
	Miscellaneous	4 668 (42.1%)	10 589 (44.3%)	27 421 (63.3%)	6 (0.8%)	2 (100%)
	Total	11 094 (100%)	23 914 (100%)	43 308 (100%)	774 (100%)	2 (100%)

TABLE III
THROUGHPUT AND LATENCY ON A XILINX VIRTEX-7 XC7VX690T FPGA FOR K ITERATIONS AND 64-QAM, AND 128 BS AND 8 USER ANTENNAS

	$K = 1$	$K = 2$	$K = 3$	$K = 4$
Max. throughput [Mb/s]	1 363	496	376	302
Latency [µs]	1.58	2.33	3.08	3.82

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 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 II. The "$\mathbf{r}$ update unit" corresponds to the output adder in Figure 3(b); the "inner-product unit" corresponds to the unit that computes $\mathbf{h}_u^H \mathbf{h}_u$ and $\mathbf{h}_u^H \mathbf{r}$ in Figure 3(a) and Figure 3(b), respectively; the "$\mathbf{h}_u \Delta z_u$ 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 $\mathbf{H}_w$ and $\mathbf{y}_w$ 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 KU, 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).

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 µs 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[8] 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

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

TABLE IV
COMPARISON OF 128×8 DATA DETECTORS FOR MASSIVE MU-MIMO SYSTEM ON A XILINX VIRTEX-7 XC7VX690T FPGA

Detector	CG [10]	Neumann [7]	Gauss-Seidel [13]	TASER [12]	OCD
Performance	near-MMSE	near-MMSE	near-MMSE	near-ML	near-MMSE
Highest modulation	64-QAM	64-QAM	64-QAM	QPSK	64-QAM
Iteration count K	3	3	1[a]	3	3
# of slices	1 094 (1.0%)	48 244 (45%)	n.a.	4 350 (4.0%)	11 094 (10%)
# of LUTs	3 324 (0.8%)	148 797 (34%)	18 976 (4.3%)	13 779 (3.2%)	23 914 (5.5%)
# of FFs	3 878 (0.4%)	161 934 (19%)	15 864 (1.8%)	6 857 (0.8%)	43 008 (4.96%)
# of DSP48s	33 (0.9%)	1 016 (28%)	232 (6.3%)	168 (5.7%)	774 (21.5%)
# of BRAM18s	1	16	6	0	2
Maximum clock frequency [MHz]	412	317	309	225	258
Latency [clock cycles]	951	196	n.a.	72	795
Maximum throughput [Mb/s]	20	621	48	50	376
Throughput/LUTs	6 017	4 173	2 530	3 629	15 597

[a]The method uses a special Neumann-series initializer followed by one GS iteration.

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 matrix $\mathbf{A}_w$, 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

[1] M. Wu, C. Dick, J. Cavallaro, and C. Studer, "FPGA design of a coordinate-descent detector for large-MIMO," in *Proc. IEEE Intl. Conf. on Circuits and Systems (ISCAS)*, May 2016.

[2] T. L. Marzetta, "Noncooperative cellular wireless with unlimited numbers of base station antennas," *IEEE Trans. Wireless Commun.*, vol. 9, no. 11, pp. 3590–3600, Nov. 2010.

[3] F. Rusek, D. Persson, B. K. Lau, E. G. Larsson, T. L. Marzetta, O. Edfors, and F. Tufvesson, "Scaling up MIMO: Opportunities and challenges with very large arrays," *IEEE Signal Process. Mag.*, vol. 30, no. 1, pp. 40–60, Jan. 2013.

[4] J. Hoydis, S. Ten Brink, and M. Debbah, "Massive MIMO in the UL/DL of cellular networks: How many antennas do we need?," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 2, pp. 160–171, Feb. 2013.

[5] E. Larsson, O. Edfors, F. Tufvesson, and T. Marzetta, "Massive MIMO for next generation wireless systems," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 186–195, Feb. 2014.

[6] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. Soong, and J. C. Zhang, "What will 5G be?," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 6, pp. 1065–1082, June 2014.

[7] M. Wu, B. Yin, G. Wang, C. Dick, J. R. Cavallaro, and C. Studer, "Large-scale MIMO detection for 3GPP LTE: algorithms and FPGA implementations," *IEEE J. Sel. Topics in Sig. Proc.*, vol. 8, no. 5, pp. 916–929, Oct. 2014.

[8] H. Prabhu, J. Rodrigues, O. Edfors, and F. Rusek, "Approximative matrix inverse computations for very-large MIMO and applications to linear pre-coding systems," in *Proc. IEEE WCNC*, 2013, pp. 2710–2715.

[9] Y. Hu, Z. Wang, X. Gaol, and J. Ning, "Low-complexity signal detection using CG method for uplink large-scale MIMO systems," in *Proc. IEEE ICCS*, Nov 2014, pp. 477–481.

[10] B. Yin, M. Wu, J. Cavallaro, and C. Studer, "VLSI Design of Large-Scale Soft-Output MIMO Detection Using Conjugate Gradients," in *Proc. IEEE ISCAS*, May 2015, pp. 1498–1501.

[11] B. Yin, M. Wu, G. Wang, C. Dick, J. R. Cavallaro, and C. Studer, "A 3.8 Gb/s large-scale MIMO detector for 3GPP LTE-Advanced," in *Proc. IEEE ICASSP*, May 2014, pp. 3907–3911.

[12] O. Castañeda, T. Goldstein, and C. Studer, "FPGA design of approximate semidefinite relaxation for data detection in large MIMO wireless systems," in *Proc. IEEE Intl. Conf. on Circuits and Systems (ISCAS)*, May 2016.

[13] Z. Wu, C. Zhang, Y. Xue, S. Xu, and Z. You, "Efficient architecture for soft-output massive MIMO detection with Gauss-Seidel method," in *Proc. IEEE Intl. Conf. on Circuits and Systems (ISCAS)*, May 2016.

[14] N. E. Tunali, M. Wu, C. Dick, and C. Studer, "Linear large-scale mimo data detection for 5g multi-carrier waveform candidates," in *Proc. Asilomar Conference on Signals, Systems, and Computers*, Nov. 2015.

[15] M. Wu, C. Dick, J. R. Cavallaro, and C. Studer, "Iterative detection and decoding in 3GPP LTE-based massive MIMO systems," in *22nd European Signal Processing Conference (EUSIPCO)*, Sept. 2014, pp. 96–100.

[16] R. Prasad, *OFDM for Wireless Communications Systems*, Artech House, Inc., Norwood, MA, USA, 2004.

[17] D. Gesbert, M. Shafi, D. Shiu, P. J. Smith, and A. Naguib, "From theory to practice: an overview of MIMO space-time coded wireless systems," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 3, pp. 281–302, 2003.

[18] A. Paulraj, R. Nabar, and D. Gore, *Introduction to Space-Time Wireless Communications*, Cambridge University Press, New York, USA, 2008.

[19] D. Seethaler, J. Jaldén, C. Studer, and H. Bölcskei, "On the complexity distribution of sphere decoding," *IEEE Trans. Inf. Theory*, vol. 57, no. 9, pp. 5754–5768, Sept. 2011.

[20] D. Seethaler, G. Matz, and F. Hlawatsch, "An efficient MMSE-based demodulator for MIMO bit-interleaved coded modulation," in *Proc. Global Telecommunications Conference (GLOBECOM)*, Nov. 2004, vol. 4, pp. 2455–2459.

[21] C. Studer, S. Fateh, and D. Seethaler, "ASIC implementation of soft-input soft-output MIMO detection using MMSE parallel interference cancellation," *IEEE J. Solid-State Circuits*, vol. 46, no. 7, pp. 1754–1765, Jul. 2011.

[22] B. Yin, M. Wu, J. R. Cavallaro, and C. Studer, "Conjugate gradient-based soft-output detection and precoding in massive MIMO systems," in *Proc. IEEE GLOBECOM*, Dec 2014, pp. 4287–4292.

[23] C. Jeon, A. Maleki, and C. Studer, "On the performance of mismatched data detection in large MIMO systems," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2016, pp. 1227–1231.

[24] P. H. Tan, L. K. Rasmussen, and T. J. Lim, "Constrained maximum-likelihood detection in CDMA," *IEEE Trans. Commun.*, vol. 49, no. 1, pp. 142–153, Jan. 2001.

[25] A. Yener, R. D. Yates, and S. Ulukus, "CDMA multiuser detection: A nonlinear programming approach," *IEEE Trans. Commun.*, vol. 50, no. 6, pp. 1016–1024, June 2002.

[26] C. Thrampoulidis, E. Abbasi, W. Xu, and B. Hassibi, "BER analysis of the box relaxation for BPSK signal recovery," *IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2016.

[27] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge Univ. Press, New York, NY, USA, 2004.

[28] G. Caire, G. Taricco, and E. Biglieri, "Bit-interleaved coded modulation," *IEEE Transactions on Information Theory*, vol. 44, no. 3, pp. 927–946, May 1998.

[29] S. J. Wright, "Coordinate descent algorithms," *Mathematical Programming*, vol. 151, no. 1, pp. 3–34, 2015.

[30] G. Gordon and R. Tibshirani, "Coordinate descent," Tech. Rep., Lecture Notes, Optimization 10-725, Carnegie Mellon University, 2015.

[31] *3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Layer Procedures (Release 10)*, 3GPP Organizational Partners TS 36.213 version 10.10.0, Jul. 2013.

[32] L. Hentilä, P. Kyösti, M. Käske, M. Narandzic, and M. Alatossava, "Matlab implementation of the WINNER phase II channel model ver 1.1," Dec. 2007.

[33] H. Kaeslin, *Digital integrated circuit design: from VLSI architectures to CMOS fabrication*, Cambridge University Press, 2008.

[34] Y. T. Lee and A. Sidford, "Efficient accelerated coordinate descent methods and faster algorithms for solving linear systems," in *IEEE 54th Annual Symposium on Foundations of Computer Science (FOCS)*, Oct. 2013, pp. 147–156.