FPGA Design of a Coordinate Descent Data Detector for Large-Scale MU-MIMO

Michael Wu[1,2], Chris Dick[2], Joseph R. Cavallaro[1], and Christoph Studer[3]

[1]Department of ECE, Rice University, Houston, TX; e-mail: cavallar@rice.edu
[2]Xilinx Inc., San Jose, CA; e-mail: miwu@xilinx.com; chris.dick@xilinx.com
[3]School of ECE, Cornell University, Ithaca, NY; e-mail: studer@cornell.edu

***Abstract*—We propose a new, low-complexity data-detection algorithm and a corresponding high-throughput FPGA design for 3GPP LTE-based large-scale (or massive) multi-user (MU) multiple-input multiple-output (MIMO) wireless communication systems. Our algorithm performs approximate minimum mean-square error (MMSE) data detection using coordinate descent (CD), which enables near-MMSE performance at low computational complexity, even for systems with hundreds of antennas at the base station (BS). We design a high-throughput VLSI architecture for 3GPP LTE wideband systems with a deep and interleaved pipeline, which can be parametrized at design time to support various antenna configurations. Our CD-based data detector achieves 379 Mb/s throughout, while using 24 k LUTs and 771 DSP units on a Xilinx Virtex-7 FPGA for a 128 BS antenna, 8 user large-scale MU-MIMO system.**

I. Introduction

Large-scale (or massive) multi-user (MU) multiple-input multiple-output (MIMO) wireless systems use hundreds of antennas at the base-station (BS) receiving data from tens of users that communicate simultaneously and in the same frequency band [1], [2]. While this emerging approach promises significantly improved spectral efficiency, coverage, and range compared to conventional, small-scale MU-MIMO wireless systems, the implementation complexity of suitable baseband processing algorithms increases significantly. In particular, data detection in the uplink (where users transmit data simultaneously to the BS) quickly results in excessive computational complexity at the BS, especially for systems supporting a large number of users. To enable high-throughput data detection for such large-scale MU-MIMO systems, a variety of low-complexity data-detection algorithms [3]–[5] and corresponding hardware designs have been proposed recently, e.g., [6]–[8]. All these detector implementations, however, are either unable to achieve the throughputs offered by modern wideband large-scale MU-MIMO systems, or exhibit excessive hardware complexity.

A. Contributions

In this paper, we propose a new, low-complexity data-detection algorithm and a corresponding high-throughput FPGA architecture for 3GPP LTE-based large-scale MU-MIMO wireless systems. Our algorithm, referred to as optimized coordinate descent (OCD), performs approximate minimum mean-square error (MMSE) data detection using coordinate descent—an iterative method that is able to invert high-dimensional linear systems at low complexity. The regularity of OCD enables near-MMSE performance at very low hardware complexity, even for systems with hundreds of BS antennas. To demonstrate the efficacy of the proposed OCD data detector, we design a corresponding high-throughput VLSI architecture with a deep and interleaved pipeline, which can be parametrized at design time to support a variety of BS and user antenna configurations. We provide reference implementation results on a Xilinx Virtex-7 FPGA and show that our OCD data detector achieves superior error-rate performance and hardware efficiency compared to existing linear data-detector implementations for large MU-MIMO systems.

The work of M. Wu, J. R. Cavallaro, and C. Studer was supported in part by Xilinx Inc., and by the US National Science Foundation under grants ECCS-1408006, CCF-1535897, ECCS-1408370, CNS-1265332, and ECCS-1232274

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 $a_k = [\mathbf{a}]_k$, and $\|\mathbf{a}\|_2 = \sqrt{\sum_k |a_k|^2}$.

II. System Model and Linear Data Detection

A. 3GPP-LTE Uplink System Model

We consider a large-scale 3GPP LTE-based uplink system, where U single-antenna user terminals send data to a BS with $B \gg U$ antennas. The LTE uplink [9] employs single-carrier frequency division multiple access (SC-FDMA), where the ith user encodes its own bit stream and maps the resulting coded bits onto constellation points in a finite set $\mathcal{O}$ (e.g., 16-QAM), with unit average transmit power and $Q = \log_2(M)$ bits per constellation point. The result is an L-dimensional time-domain (TD) vector $\mathbf{x}^{(i)} = \left[x_1^{(i)}, \ldots, x_L^{(i)}\right]^T$. Each user then applies a discrete Fourier transform (DFT) to the TD symbol vector to obtain a frequency domain (FD) symbol vector $\mathbf{s}^{(i)}$, where the vector $\mathbf{s}^{(i)} = \left[s_1^{(i)}, \ldots, s_L^{(i)}\right]^T$ contains the FD symbols transmitted by the ith user. For each user, these FD symbols are assigned to pre-defined data-carrying subcarriers, and transformed to the TD using an inverse DFT. After prepending the cyclic prefix, all users transmit their TD signals simultaneously over the frequency-selective channel.

At the BS, the TD signals received at each antenna are transformed back to the FD using a DFT. The FD input-output relation on the wth subcarrier can be modeled as $\mathbf{y}_w = \mathbf{H}_w \mathbf{s}_w + \mathbf{n}_w$, where

$$\mathbf{y}_w = \begin{pmatrix} y_w^{(1)} \\ \vdots \\ y_w^{(B)} \end{pmatrix}, \quad \mathbf{H}_w = \begin{pmatrix} H_w^{(1,1)} & \cdots & H_w^{(1,U)} \\ \vdots & \ddots & \vdots \\ H_w^{(B,1)} & \cdots & H_w^{(B,U)} \end{pmatrix},$$

$$\mathbf{s}_w = [s_w^{(1)}, \ldots, s_w^{(U)}]^T, \ \ \mathbf{n}_w = [n_w^{(1)}, \ldots, n_w^{(B)}]^T.$$

Here, $y_w^{(i)}$ is the received FD symbol on the wth subcarrier for antenna i, $H_w^{(i,j)}$ models the wireless channel on the wth subcarrier between the ith receive antenna and jth user, $s_w^{(j)}$ denotes the symbol transmitted by the jth user on the wth subcarrier, and $n_w^{(i)}$, $\forall w$, are i.i.d. complex circularly-symmetric Gaussian random variables with variance N_0 per complex entry.

B. Soft-Output MMSE Data Detection

Throughout the paper, we focus on linear data detection algorithms because they enable low-complexity hardware designs in large-scale MU-MIMO systems. We build our algorithm on the soft-output MMSE data detection algorithm proposed in [10]. To obtain estimates of the transmitted FD symbols in SC-FDMA systems, we first perform MMSE equalization on a per-subcarrier basis and then, compute soft-outputs in the form of log-likelihood ratio (LLR) values.

MMSE data detection starts by computing 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 $\mathbf{y}_w^{\text{MF}} = \mathbf{H}_w^H\mathbf{y}_w$. One can then compute the MMSE-equalized FD symbols as $\tilde{\mathbf{s}}_w = \mathbf{A}_w^{-1}\mathbf{y}_w^{\text{MF}}$ [8], [10]. The LLR values are obtained by first performing an IDFT on $\tilde{\mathbf{s}}^{(i)} = [\tilde{s}_1^{(i)}, \ldots, \tilde{s}_L^{(i)}]^T$ to obtain the TD estimate $\tilde{\mathbf{x}}^{(i)} = [\tilde{x}_1^{(i)}, \ldots, \tilde{x}_L^{(i)}]^T$. Then, the max-log LLR value of the jth bit of tth symbol can be computed as [10]

$$L_{(t,j)}^{(i)} = \rho^{(i)}\left(\min_{a\in\mathcal{O}_j^0}\left|\frac{\tilde{x}_t^{(i)}}{\mu^{(i)}} - a\right|^2 - \min_{a\in\mathcal{O}_j^1}\left|\frac{\tilde{x}_t^{(i)}}{\mu^{(i)}} - a\right|^2\right), \quad (1)$$

where $\mathcal{O}_j^0$ and $\mathcal{O}_j^1$ sets of constellation symbols for which the jth bit is 0 and 1 respectively, and $\rho^{(i)} = (\mu^{(i)})^2/\nu_i^2$ where $\nu_i^2 = \mu^{(i)} - |\mu^{(i)}|^2$ for SC-FDMA-based systems [8]. We also need the so-called effective channel gain $\mu^{(i)} = L^{-1}\sum_{w=1}^{L}\mathbf{a}_{i,w}^H\mathbf{g}_{i,w}$, where $\mathbf{a}_{i,w}^H$ is the ith row of $\mathbf{A}_w^{-1}$ and $\mathbf{g}_{i,w}$ is the ith column of $\mathbf{G}_w$.

III. MMSE Detection via Coordinate Descent (CD)

A. MMSE Equalization

The soft-output MMSE data detection algorithm outlined above may result in high complexity for large-scale MU-MIMO systems. In particular, computing the FD MMSE estimates $\tilde{\mathbf{s}}_w = \mathbf{A}_w^{-1}\mathbf{y}_w^{\text{MF}}$, $\forall w$, requires the calculation of the $U\times U$-dimensional matrix inverses $\mathbf{A}_w^{-1}$, which entails prohibitive complexity for large-dimensional systems. Thus, instead of computing $\tilde{\mathbf{s}}_w = \mathbf{A}_w^{-1}\mathbf{y}_w^{\text{MF}}$, which requires the inverse $\mathbf{A}_w^{-1}$, the estimate $\tilde{\mathbf{s}}_w$ can be computed more efficiently by solving the following linear system of equations $\mathbf{A}_w\tilde{\mathbf{s}}_w = \mathbf{y}_w^{\text{MF}}$ for the vector $\tilde{\mathbf{s}}_w$. As it has been realized in [5], this problem is equivalent to solving the following regularized LS optimization problem:

$$\tilde{\mathbf{s}}_w = \arg\min_{\mathbf{z}\in\mathbb{C}^U} \|\mathbf{y}_w - \mathbf{H}_w\mathbf{z}\|_2^2 + N_0\|\mathbf{z}\|_2^2. \quad (2)$$

While the solution to such regularized LS problems can be found (exactly or approximately) at low computational complexity using conjugate gradient (CG) methods, see, e.g., [5], corresponding VLSI implementations [7] are unable to achieve the high throughputs offered by 3GPP LTE-based large-MIMO systems, mainly due to a fairly complex algorithm structure and the need for high arithmetic precision.

B. Coordinate Descent (CD)

We now propose an alternative method to solve (2) by means of coordinate descent (CD) [11]. To this end, we first define

$$f(z_1, \ldots, z_U) = f(\mathbf{z}) = \|\mathbf{y}_w - \mathbf{H}_w\mathbf{z}\|_2^2 + N_0\|\mathbf{z}\|_2^2, \quad (3)$$

and then minimize the function $f(z_1, \ldots, z_U)$ for each variable z_u, $u = 1, \ldots, U$, independently and in a round-robin fashion.[1] To simplify notation, we omit the subcarrier index w in the following discussion. Assume we want to find the optimum uth value z_u. To this end, we hold all other values z_j, $\forall j \neq u$, fixed and set the gradient of the function (3) with respect to the first component to zero, i.e.,

$$0 = \nabla_u f(\mathbf{z}) = \mathbf{h}_u^H(\mathbf{H}\mathbf{z} - \mathbf{y}) + N_0 z_u. \quad (4)$$

[1]The performance of CD can often be improved by randomizing the variable-update order; in our design, we simply perform round-robin updates.

Algorithm 1 Optimized Coordinate Descent (OCD)

inputs: $\mathbf{y}$, $\mathbf{H}$, and N_0
initialization: $\mathbf{t} = \mathbf{0}^{B\times 1}$ and $\mathbf{z}^{(0)} = \mathbf{0}^{U\times 1}$
preprocessing: $d_u^{-1} = (\|\mathbf{h}_u\|_2^2 + N_0)^{-1}$, $u = 1, \ldots, U$
for $k = 1, \ldots, K$ **do**
  for $u = 1, \ldots, U$ **do**
    $\Delta z = d_u^{-1}\mathbf{h}_u^H(\mathbf{y} - \mathbf{t})$
    $z_u^{(k)} = z_u^{(k-1)} + \Delta z$
    $\mathbf{t} = \mathbf{h}_t\Delta z + \mathbf{t}$
  end for
end for
outputs: $\tilde{\mathbf{s}} = [z_1^{(K)}, \ldots, z_U^{(K)}]^T$

By decomposing $\mathbf{H}\mathbf{z} = \mathbf{h}_u z_u + \sum_{j\neq u}\mathbf{h}_j z_j$, we can now solve (4) for z_u and obtain the following expression:

$$z_u = (\|\mathbf{h}_u\|_2^2 + N_0)^{-1}\mathbf{h}_u^H\Big(\mathbf{y} - \textstyle\sum_{j\neq u}\mathbf{h}_j z_j\Big), \quad (5)$$

which is the CD update rule for the uth entry of $\mathbf{z}$. We can now compute (5) for $t = 1, \ldots, U$ and repeat that procedure for K iterations to obtain an estimate for $\tilde{\mathbf{s}} = \mathbf{z}^{(K)}$, where $\mathbf{z}^{(K)}$ is the result of the above-described iteration; see [11] for more details on CD.

C. Optimized Coordinate Descent (OCD)

To reduce the complexity of CD, we precompute the regularized inverse squared column norms of $\mathbf{H}$, i.e., $d_u^{-1} = (\|\mathbf{h}_u\|_2^2 + N_0)^{-1}$ for $u = 1, \ldots, U$. This preprocessing step results in significant complexity reduction as all subsequent CD iterations do not need to recompute this value. To further reduce the complexity of CD, we perform the following iterative updates. In a first step, we update the uth component of $\mathbf{z}$ given the input vectors $\mathbf{h}_u$, $\mathbf{y}$, and $\mathbf{t}$. In a second step, we update the temporary vector $\mathbf{t}$, which is used in the next iteration. We then sequentially update the symbol estimates. While the original CD algorithm in Section III-B requires one complex-valued dot (or inner) product and $U-1$ complex scalar-by-vector multiplications per iteration, our optimized CD algorithm (short OCD) requires only one dot product and one complex scalar-by-vector multiplication; this results in significantly reduced computational complexity. The resulting OCD algorithm is summarized in Algorithm 1.

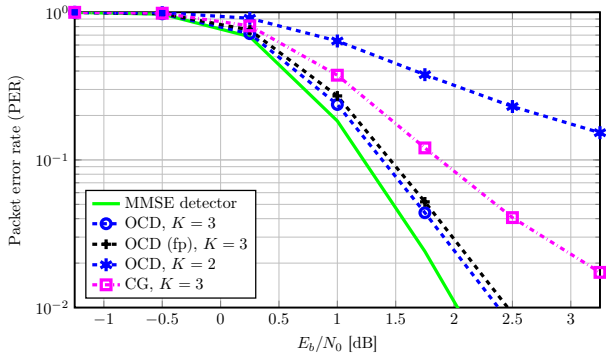
D. LLR Approximation for OCD

To compute the LLR values (1), we must resort to an approximation as we never compute the inverse $\mathbf{A}_w^{-1}$. To this end, we use the approximation in [5], [8], which computes $\mu^{(i)} \approx L^{-1}\sum_{w=1}^{L}(d_{i,w})^{-1}g_{i,w}$, where $(d_{i,w})^{-1}$ is the ith regularized inverse squared column norm of $\mathbf{H}_w$ and $g_{i,w}$ is the entry in the ith diagonal of the matrix $\mathbf{G}_w$. Furthermore, we use the approximation $\rho^{(i)} \approx (1-\mu^{(i)})^{-1}$.

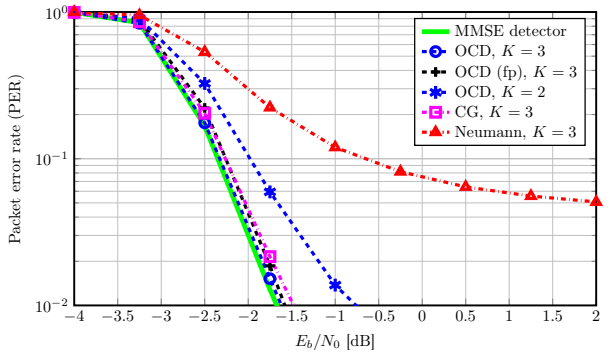
E. Error-Rate Performance

We simulate a 20 MHz 3GPP LTE uplink system using SC-FDMA with 64-QAM and a rate-3/4 turbo code. We also consider a WINNER-Phase-2 channel model with 7.8 cm antenna spacing [5], [8]. For channel decoding, we use a log-map turbo decoder.

Figures 1(a) and 1(b) show the packet error rate (PER) for OCD, as well as other approximate data-detection methods for large-scale MU-MIMO systems, namely the Neumann-series detector [6] and CG-based detection [7]. We also include an exact linear MMSE detector as a reference. We show results for two antenna configurations, 64×8 and 128×8, where we use the notation $B\times U$. OCD with three iterations ($K = 3$) achieves near-MMSE performance, and outperforms the two other approximate detectors for the same iteration count.



(a) 64 BS antennas and 8 users.



(b) 128 BS antennas and 8 users.

Fig. 1. Packet error rate (PER) for a 64×8 and 128×8 massive MIMO system. Optimized coordinate descent (OCD) achieves close-to-MMSE PER performance and outperforms the two other approximate methods [6], [7].

IV. VLSI Architecture

A. Architecture Overview

Figure 2 shows two high-level block diagrams of the proposed OCD architecture. The inputs of our architecture are the channel matrix $\mathbf{H}$, the received vector $\mathbf{y}$, and the noise variance N_0. Our architecture supports two operation modes: (a) preprocessing (line 3 of Algorithm 1) and (b) CD iterations (lines 5–9). Preprocessing and detection are carried out in a B-wide vector processing pipeline, which processes B-dimensional vectors at a time. In the preprocessing mode, we compute the regularized inverse squared column norms d_u^{-1}. In the detection mode, we perform one OCD iteration. To support these two modes without the need of redundant computation units, the processing pipeline shares the key operations used in both modes. In particular, both supported modes share the dot-product unit and the right shifter unit (highlighted in red in Figure 2).

The main implementation challenge of the OCD algorithm are data dependencies between successive iterations, which prevent conventional architecture pipelining. In particular, each OCD iteration updates the temporary vector $\mathbf{t}$ and the vector $\mathbf{z}_u^{(k+1)}$ given the previous vectors $\mathbf{t}$ and $\mathbf{z}_u^{(k)}$. To achieve high throughput, we deploy *pipeline interleaving*, i.e., we simultaneously process multiple subcarriers in an interleaved manner. 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 occupied. We use 24 pipeline stages, which enables our design to achieve up to 260 MHz.[2]

[2]We note that it is possible to achieve even higher clock frequencies by increasing the number of pipeline stages (especially for smaller values of B). This approach, however, entails a significant hardware overhead.

TABLE I
Implementation results on a Xilinx Virtex-7 XC7VX690T FPGA for different BS antenna numbers

Array size	$B = 32$	$B = 64$	$B = 128$
Slices	3 732	6 709	13 447
LUTs	5 909	13 779	23 956
FFs	15 148	27 795	61 335
DSP48s	195	387	771
BRAM18	1	1	1
Max. clock frequency	262 MHz	261 MHz	262 MHz

TABLE II
Throughput on a Xilinx Virtex-7 XC7VX690T FPGA for K iterations and 64-QAM, and 128 BS and 8 user antennas

	$K = 1$	$K = 2$	$K = 3$	$K = 4$
Throughput [Mb/s]	1 379	500	379	304

B. Fixed-point Arithmetic

In order to optimize the hardware efficiency of our architecture, we deploy fixed-point arithmetic. We achieved a negligible implementation loss with 16 bit precision for most internal signals; see Figure 1 for the fixed-point (fp) performance. There are two exceptions.

1) Dot-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 precision. The dot product unit computes $\|\mathbf{h}_u\|_2^2$ (line 3 of Algorithm 1) and $\mathbf{h}_u^H(\mathbf{y} - \mathbf{t})$ (line 6). As both output terms are close to B (for large values of B), we shift these terms by $b = \lceil \log_2(B) \rceil$ bit to the right to reduce the dynamic range. Since we shift $\|\mathbf{h}_u\|_2^2$ by b to the right, when we compute the reciprocal value, $d_u^{-1} = (\|\mathbf{h}_u\|_2^2 + N_0)^{-1}$, we effectively shift the reciprocal term d_u^{-1} by b bit to the left. In the dot-product unit, we also shift the term $\mathbf{h}_u^H(\mathbf{y} - \mathbf{t})$ by b bit to the right. Consequently, we do not need to undo these shifts, as both of them cancel during the multiplication on line 6 of Algorithm 1.

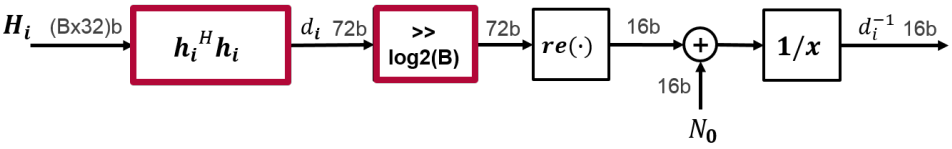
2) Reciprocal unit: This unit consists of two parts. The first part normalizes the input signal to the range $[0.5, 1]$. The second part generates a reciprocal value for the normalized input; we use a BRAM18 as a 18 bit, 2048 entry look-up table (LUT), where the top 11 bit of the normalized value are used to point to the entry in the LUT that stores the associated normalized reciprocal. We then denormalize the reciprocal value (see [6] for more details).

V. Implementation Results

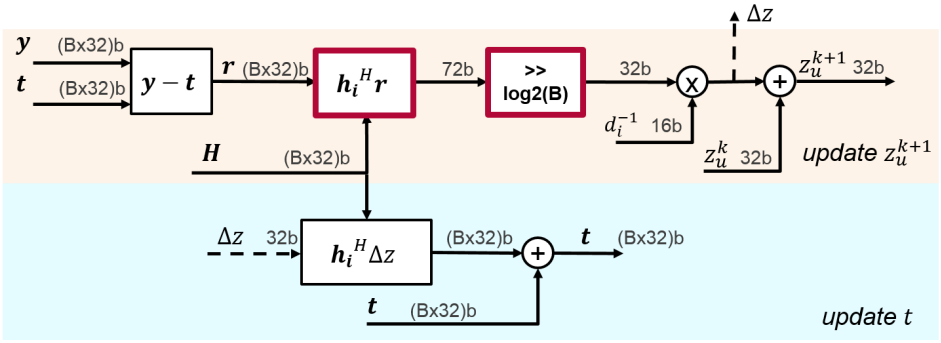
We used Vivado HLS (version 2015.2) to parameterize our OCD architecture for various antenna configurations, i.e., for $B = 32$, $B = 64$ and $B = 128$ BS antennas. For each configuration, we provide post place-and-route implementation results on a Xilinx Virtex-7 XC7VX690T. All implementations support $U \leq 32$ users and $K \leq 256$ OCD iterations, which we can control at run-time.

The hardware complexity and resource utilization results are shown in Table I. Since the proposed architecture performs operations on B-dimensional vectors, the resource utilization (excluding the BRAM) scales linearly with B. Since the quantities $\mathbf{H}$ and $\mathbf{y}$ are assumed to be stored in external memories, our OCD architecture only uses one BRAM18 for the LUT in the reciprocal unit.

The throughput results are shown in Table II. 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



(a) OCD preprocessing mode.



(b) OCD iteration mode.

Fig. 2. High-level block diagram of OCD detector pipeline. The pipeline is reconfigurable at design time for various antenna configurations, and is able to either perform preprocessing or carry out OCD iterations. The shared units between both processing modes are outlined in red.

TABLE III
COMPARISON OF DATA DETECTORS FOR AN 128×8 LARGE-SCALE MIMO SYSTEM ON A XILINX VIRTEX-7 XC7VX690T FPGA

Detector	CG [7]	Neumann [6]	OCD
Slices	1 094 (1.0%)	48 244 (45%)	13 447 (12%)
LUTs	3 324 (0.8%)	148 797 (34%)	23 955 (5.5%)
FFs	3 878 (0.4%)	161 934 (19%)	61 335 (7.1%)
DSP48s	33 (0.9%)	1 016 (28%)	771 (21%)
BRAM18	1	16	1
Clock freq. [MHz]	412	317	262
Latency [cycles]	951	196	795
Throughput [Mb/s]	20	621	379
Throughput/LUTs	6 017	4 173	15 821

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. In contrast, the processing latency increases with respect to both U and K. Specifically, the latency of this design is $24(K+1)U + 26$ clock cycles, where 26 cycles are required to flush the pipeline.

Table III compares OCD to two other large-scale MIMO data detectors, namely the CG-based detector [7] and the Neumann-series detector [6]. Both of these detectors have been implemented on the same FPGA and for a 128 BS antenna, 8 user system. We see that OCD for the same system configuration outperforms these designs in terms of hardware efficiency (measured in terms of throughput per FPGA LUTs). Furthermore, our OCD detector achieves superior PER performance than the other two detectors (see Figs. 1(a) and 1(b)), which demonstrates the effectiveness of OCD.

VI. CONCLUSIONS

We have implemented a new, coordinate descent (CD)-based data detector, called optimized CD (short OCD), for 3GPP LTE large-scale MU-MIMO systems. The proposed OCD detector enables high-performance linear data detection, while enabling VLSI designs that require low hardware complexity. Our FPGA reference design achieves 379 Mb/s for a 128 BS antenna, 8 user system, and outperforms existing approximate linear data-detection methods in terms of hardware efficiency and error-rate performance. Hence, OCD enables realistic 3GPP-LTE large-scale MU-MIMO systems to support tens of users communicating with hundreds of BS antennas.

REFERENCES

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] S. Sesia, I. Toufik, and M. Baker, *LTE, The UMTS Long Term Evolution: From Theory to Practice*, Wiley Publishing, 2009.

[10] 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.

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