# FPGA Design of Approximate Semidefinite Relaxation for Data Detection in Large MIMO Wireless Systems

Oscar Castañeda[1,3], Tom Goldstein[2], and Christoph Studer[3]

[1]Department of EE, Del Valle de Guatemala University, Guatemala City, Guatemala; e-mail: cas11086@uvg.edu.gt
[2]Department of CS, University of Maryland, College Park, MD; e-mail: tomg@cs.umd.edu
[3]School of ECE, Cornell University, Ithaca, NY; e-mail: studer@cornell.edu

***Abstract*—We propose a novel, near-optimal data detection algorithm and a corresponding FPGA design for large multiple-input multiple-output (MIMO) wireless systems. Our algorithm, referred to as TASER (short for triangular-approximated semidefinite relaxation), relaxes the maximum-likelihood (ML) detection problem to a semidefinite program and solves a non-convex approximation using a preconditioned forward-backward splitting procedure. We show that TASER achieves near-ML performance at low computational complexity, even for large-dimensional MIMO systems. We develop a systolic array that implements TASER and achieves high throughput at low hardware complexity. To demonstrate the effectiveness of our solution, we develop reference designs on a Xilinx Virtex-7 FPGA for various antenna configurations. One of our TASER designs achieves up to 98 Mb/s for a 32-user system employing QPSK, while consuming only 150 k FPGA look-up tables.**

## I. Introduction

Large multiple-input multiple-output (MIMO) is believed to be a key technology for 5G wireless communication systems [1], [2]. The idea is to equip the base station (BS) with hundreds (or more) of antennas, while serving a typically smaller number of (single- or multi-antenna) users at the same time in the same frequency band. While large-MIMO promises improved spectral efficiency compared to more traditional, small-scale MIMO systems, the potentially large number of user antennas requires computationally expensive data-detection algorithms. To enable high-throughput uplink communication (where users transmit data to the BS), a variety of low-complexity data-detection algorithms [3]–[5], as well as corresponding FPGA implementations [6]–[8] and application-specific integrated circuit (ASIC) designs [9] have been recently proposed. All existing data detectors for large-MIMO, however, rely on linear data detection (and approximations thereof). Such algorithms enable high-throughput VLSI designs, but entail a significant performance loss in "not-so-large" MIMO systems, where the ratio between the number of BS antennas and user antennas is rather small (e.g., two or lower).

### A. Contributions

In this paper, we develop a novel data detection algorithm and a corresponding FPGA design for large-MIMO systems, which we refer to as TASER (short for triangular-approximated semidefinite relaxation). Our detector builds upon semidefinite relaxation (SDR) [10], [11], which enables near-ML data detection performance, even for systems where the number of BS antennas is equal to the number of users [12]. TASER approximates the SDR formulation of the ML problem using a Cholesky factorization, and solves the resulting non-convex problem using forward-backward splitting (FBS) [13]. We develop a corresponding systolic array, which enables high throughput at manageable implementation costs. We provide implementation results for a Xilinx Virtex-7 FPGA and perform a comparison with the recently-proposed large-MIMO data detectors [6]–[8] in terms of error-rate performance, throughput, and FPGA-implementation complexity.

### B. Notation

Lowercase boldface letters stand for column vectors; uppercase boldface letters denote matrices. For a matrix $\mathbf{A}$, we denote its transpose by $\mathbf{A}^T$. 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 vector $\mathbf{a}$ is denoted by $a_k = [\mathbf{a}]_k$. The $\ell_2$-norm of $\mathbf{a}$ is $\|\mathbf{a}\|_2 = \sqrt{\sum_k |a_k|^2}$. The identity matrix is $\mathbf{I}$ and the all-ones vector is $\mathbf{1}$. The real and imaginary part of a complex vector $\mathbf{a}$ is denoted by $\Re(\mathbf{a})$ and $\Im(\mathbf{a})$, respectively.

## II. System Model and Semidefinite Relaxation

We consider a large MIMO wireless uplink system with $B$ BS antennas and $U \leq B$ user antennas. We use the standard input-output relation to model the (flat-fading) wireless channel: $\mathbf{y} = \mathbf{H}\mathbf{s} + \mathbf{n}$. Here, $\mathbf{y} \in \mathbb{C}^B$ is the BS receive-vector, $\mathbf{H} \in \mathbb{C}^{B\times U}$ is the MIMO channel matrix, $\mathbf{s} \in \mathcal{O}^U$ is the transmit vector containing the data symbols from all users ($\mathcal{O}$ refers to the constellation set), and $\mathbf{n} \in \mathbb{C}^B$ is i.i.d. circularly-symmetric Gaussian with variance $N_0$ per entry. For this model, maximum-likelihood (ML) detection corresponds to

$$\hat{\mathbf{s}}^{\text{ML}} = \arg\min_{\mathbf{s}\in\mathcal{O}^U} \|\mathbf{y} - \mathbf{H}\mathbf{s}\|_2, \quad (1)$$

and its runtime complexity scales exponentially with the number of users $U$, even with computationally efficient sphere-decoding algorithms [14]. Semidefinite relaxation (SDR) of (1) is a well-known ML approximation [10] that enables significantly lower (i.e., polynomial) complexity for large-MIMO systems employing BPSK and QPSK constellations[1], while achieving full ML-diversity [12].

SDR starts with the real-valued decomposition of the MIMO system, i.e., $\bar{\mathbf{y}} = [\Re(\mathbf{y}); \Im(\mathbf{y})]$ and $\overline{\mathbf{H}} = [\Re(\mathbf{H}), -\Im(\mathbf{H}); \Im(\mathbf{H}), \Re(\mathbf{H})]$, and solves the following semidefinite program (SDP) [11]:

$$\widehat{\mathbf{X}} = \arg\min_{\mathbf{X}} \operatorname{trace}(\mathbf{T}\mathbf{X}) \text{ subject to } \operatorname{diag}(\mathbf{X}) = \mathbf{1},\, \mathbf{X} \succeq 0. \quad (2)$$

Here, $\mathbf{T} = [\overline{\mathbf{H}}^T\overline{\mathbf{H}}, -\overline{\mathbf{H}}^T\bar{\mathbf{y}}; -\bar{\mathbf{y}}^T\overline{\mathbf{H}}, \bar{\mathbf{y}}^T\bar{\mathbf{y}}]$ is of dimension $N \times N$ with $N = 2U + 1$ for QPSK, and the constraint $\mathbf{X} \succeq 0$ ensures that $\mathbf{X}$ is a positive semidefinite (PSD) matrix. An estimate of the ML solution in (1) is obtained by taking the signs of the leading eigenvector of $\mathbf{X}$ (see [11] for the details). While (2) can be solved exactly using interior-point methods [11], these algorithms require a large number of iterations, where each iteration involves the computation of eigenvalue decompositions (or matrix inverses) and transcendental functions. We believe that these are the main reasons that—until now—*no* VLSI design of an SDR detector has been described in the open literature.

## III. TASER: Triangular-Approximated Semidefinite Relaxation

We now detail our algorithm, referred to as TASER, which computes an approximate solution to the SDP in (2) at low complexity.

The work of O. Castañeda and C. Studer was supported in part by Xilinx Inc., and by the US National Science Foundation (NSF) under grants ECCS-1408006 and CCF-1535897. The work of T. Goldstein was supported in part by the US NSF under grant CCF-1535902 and by the US Office of Naval Research under grant N00014-15-1-2676.

[1] SDR methods for other constellations exist; see, e.g., [15] for more details.

**Algorithm 1** TASER

```
1: inputs: $\widetilde{\mathbf{T}}$, $\mathbf{D}$, and $\tau = 1/\|\widetilde{\mathbf{T}}\|_2$
2: initialization: $\widetilde{\mathbf{L}}^{(0)} = \mathbf{D}$
3: for $t = 1, \ldots, t_{\max}$ do
4:     $\mathbf{V}^{(t)} = \widetilde{\mathbf{L}}^{(t-1)} - \mathrm{tril}(2\tau\widetilde{\mathbf{L}}^{(t-1)}\widetilde{\mathbf{T}})$
5:     $\widetilde{\mathbf{L}}^{(t)} = \mathrm{prox}_{\tilde{g}}(\mathbf{V}^{(t)})$
6: end for
7: outputs: $\bar{s}_k = \mathrm{sign}(\widetilde{L}^{(t_{\max})}_{N,k}), k = 1, \ldots, N-1$
```

### A. Triangular SDP Formulation via the Cholesky Decomposition

The key idea of TASER builds upon the fact that PSD matrices can be factorized using the Cholesky decomposition $\mathbf{X} = \mathbf{L}^T\mathbf{L}$, where $\mathbf{L}$ is an $N \times N$ lower-triangular matrix. This allows us to reformulate the SDR problem in (2) as

$$\widehat{\mathbf{L}} = \arg\min_{\mathbf{L}} \ \mathrm{trace}(\mathbf{L}\mathbf{T}\mathbf{L}^T) \ \text{subject to} \ \|\boldsymbol{\ell}_k\|_2 = 1, \forall k, \qquad (3)$$

where we replaced the constraint $\mathrm{diag}(\mathbf{L}^T\mathbf{L}) = \mathbf{1}$ of (2) by an equivalent $\ell_2$-norm equality constraint on the $k$th column $\boldsymbol{\ell}_k = [\mathbf{L}]_k$. For BPSK and QPSK, we take the signs of the last row of the solution matrix $\widehat{\mathbf{L}}$ from (3); see our planned journal paper [16] for the details.

### B. Forward-Backward Splitting (FBS)

Since the problem (3) is non-convex, finding an optimal solution is difficult. For TASER, we apply FBS [13] (a computationally efficient method to solve convex optimization problems) to the non-convex problem in (3). While this approach is not guaranteed to converge to the optimal solution of the non-convex problem (3), our simulation results in Section V show excellent error-rate performance.

FBS is an efficient, iterative method to solve convex optimization problems of the form $\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} f(\mathbf{x}) + g(\mathbf{x})$, where the function $f$ is smooth and convex, and $g$ is convex but non-smooth, using the following iterative process (for $t = 1, 2, \ldots, t_{\max}$) [13]:

$$\mathbf{x}^{(t)} = \mathrm{prox}_g(\mathbf{x}^{(t-1)} - \tau\nabla f(\mathbf{x}^{(t-1)}); \tau).$$

Here, $\tau > 0$ is a suitably-chosen step size, $\nabla f(\mathbf{x})$ is the gradient of $f$, and the proximal operator for the function $g$ is [13]

$$\mathrm{prox}_g(\mathbf{z}; \tau) = \arg\min_{\mathbf{x}} \left\{\tau g(\mathbf{x}) + \tfrac{1}{2}\|\mathbf{x} - \mathbf{z}\|_2^2\right\}. \qquad (4)$$

### C. The TASER Algorithm

To solve (3) using FBS, we set $f(\mathbf{L}) = \mathrm{trace}(\mathbf{L}\mathbf{T}\mathbf{L}^T)$ and $g(\mathbf{L}) = \chi(\|\boldsymbol{\ell}_k\|_2 = 1, \forall k)$, where $\chi$ is the characteristic function (which is zero if the constraint is met and infinity otherwise). For these definitions, the gradient is given by $\nabla f(\mathbf{L}) = \mathrm{tril}(2\mathbf{L}\mathbf{T})$, where $\mathrm{tril}(\cdot)$ extracts the lower-triangular part; the proximal operator (4) is given by $\mathrm{prox}_g(\boldsymbol{\ell}_k; \tau) = \boldsymbol{\ell}_k/\|\boldsymbol{\ell}_k\|_2$, $\forall k$. We use a step size of $\tau = 1/\|\mathbf{T}\|_2$, where $\|\mathbf{T}\|_2$ is the spectral norm of the matrix $\mathbf{T}$.

To ensure fast convergence of FBS, we precondition (3). To this end, we compute a diagonal matrix $\mathbf{D} = \mathrm{diag}(\sqrt{T_{1,1}}, \ldots, \sqrt{T_{M,M}})$ which allows us to precondition the matrix $\widetilde{\mathbf{T}} = \mathbf{D}^{-1}\mathbf{T}\mathbf{D}^{-1}$ so that it has an all-ones diagonal. We then run FBS on a normalized lower-triangular matrix $\widetilde{\mathbf{L}} = \mathbf{D}\mathbf{L}$ until a maximum number of iterations $t_{\max}$ has been reached. Preconditioning also requires a modified proximal operator: $\mathrm{prox}_{\tilde{g}}(\tilde{\boldsymbol{\ell}}_k) = D_{k,k}\tilde{\boldsymbol{\ell}}_k/\|\tilde{\boldsymbol{\ell}}_k\|_2$. We next propose a systolic array that enables us to implement TASER as summarized in Algorithm 1; more details can be found in [16].

## IV. Systolic VLSI Architecture

### A. Architecture Overview

Figure 1 shows the proposed triangular systolic array consisting of $\frac{1}{2}N(N+1)$ processing elements (PEs). Each PE contains an entry $\widetilde{L}^{(t-1)}_{i,j}$ of the lower-triangular matrix $\widetilde{\mathbf{L}}^{(t-1)}$. All PEs in the same column and row receive data from a column-broadcast unit (CBU) and a row-broadcast unit (RBU), respectively. Both of these broadcast units enable the computation of the $N \times N$ matrix-matrix multiplication on line 4 of Algorithm 1 in $N$ clock cycles. In the $k$th cycle during the $t$th TASER iteration, the RBU of the $i$th row sends the value $\widetilde{L}^{(t-1)}_{i,k}$ to all PEs on row $i$, while the $j$th CBU sends $\widehat{T}_{k,j}$ to all PEs on column $j$. We assume that the (scaled) matrix $\widehat{\mathbf{T}} = 2\tau\widetilde{\mathbf{T}}$ has been computed in a pre-processing step and is stored in distributed FPGA look-up tables (LUTs), instead of block RAMs. With the data from the RBU and CBU, each PE then performs a multiply-accumulate (MAC) operation until the matrix-matrix multiplication is complete. The subtraction operation on line 4 is carried out by initializing the accumulator with $\widetilde{L}^{(t-1)}_{i,j}$ and by sequentially subtracting $\widetilde{L}^{(t-1)}_{i,k}\widehat{T}_{k,j}$.

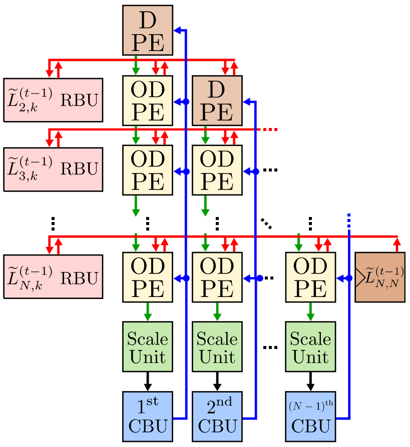


Fig. 1. High-level block diagram of TASER. We use a systolic array of processing elements (PEs) for the diagonal (D) and off-diagonal (OD) elements, which enables high throughput at moderate hardware complexity.

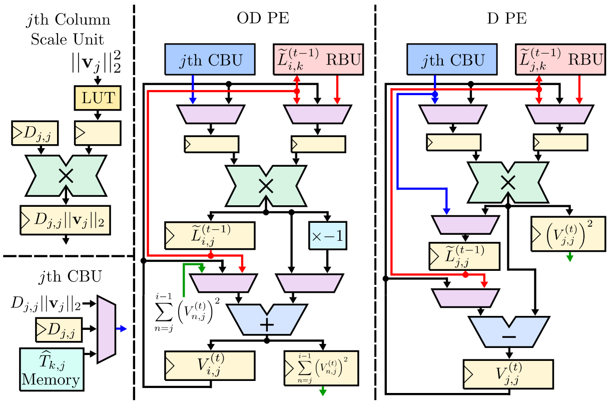
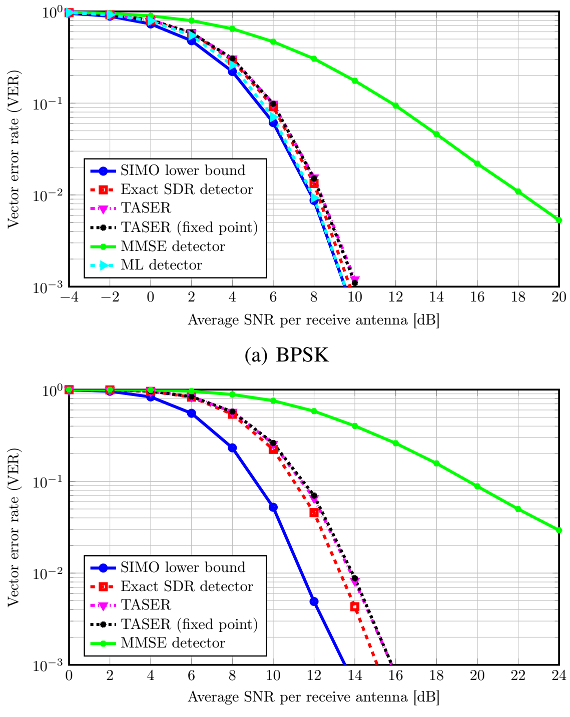

$j$th Column Scale Unit
$\|\mathbf{v}_j\|_2^2$
LUT
$D_{j,j}$
$D_{j,j}\|\mathbf{v}_j\|_2$
$j$th CBU
$D_{j,j}\|\mathbf{v}_j\|_2$
$D_{j,j}$
$\widehat{T}_{k,j}$ Memory
OD PE
$j$th CBU
$\widetilde{L}^{(t-1)}_{i,k}$ RBU
$\widetilde{L}^{(t-1)}_{i,j}$
$\times -1$
$\sum_{n=j}^{i-1}(V^{(t)}_{n,j})^2$
$V^{(t)}_{i,j}$
$\sum_{n=j}^{i-1}(V^{(t)}_{n,j})^2$
D PE
$j$th CBU
$\widetilde{L}^{(t-1)}_{j,k}$ RBU
$(V^{(t)}_{j,j})^2$
$\widetilde{L}^{(t-1)}_{j,j}$
$V^{(t)}_{j,j}$

Fig. 2. Architecture details of the column-broadcast unit (CBU), the column-scale unit, and the off-diagonal (OD) and diagonal (D) PEs.

Since $\widetilde{\mathbf{L}}$ is lower-triangular, the $V^{(t)}_{i,j}$ value from line 4 can be computed for all PEs in the $i$th row in only $i$ clock cycles. To implement the $\mathrm{prox}_{\tilde{g}}$ function, during the $(i+1)$th cycle each PE on the $i$th row squares $V^{(t)}_{i,j}$ and passes it downwards to the next PE in the same column (the green arrows in Figure 1). In the $(i+2)$th cycle, the PEs of the $(i+1)$th row square their $V^{(t)}_{i+1,j}$ and add the result to the value from the previous row. This enables the calculation of the squared $\ell_2$-norm in $N+1$ cycles. For the $j$th column, the squared $\ell_2$-norm is passed to a scale unit, which computes the inverse square root and multiplies it with $D_{j,j}$. The result is then sent to all the PEs in the same column via the CBU. All PEs then multiply this scaling factor to their $V^{(t)}_{i,j}$ value to obtain the next iterate $\widetilde{L}^{(t)}_{i,k}$, thus



(a) BPSK

(b) QPSK

Fig. 3. Uncoded vector error rate (VER) for a 16 BS antenna, 16 user antenna large MIMO system. TASER achieves near-optimal VER performance (close-to-ML and the SIMO lower bound) and achieves similar performance as the exact SDR detector; linear MMSE data detection performs only poorly.

completing the proximal operation on line 5.

### *B. Processing Element*

We designed two slightly distinct types of PEs in our systolic array: (i) off-diagonal (OD) PEs and (ii) diagonal (D) PEs (cf. Figure 2). Both PE types support the following four operation modes:

*1) Initialization of* $\mathbf{L}$*:* This mode is used for line 2 of Algorithm 1. All off-diagonal PEs initialize $\widetilde{L}_{i,j}^{(t-1)}=0$; the diagonal PEs initialize their states with $D_{j,j}$ received from the CBU.

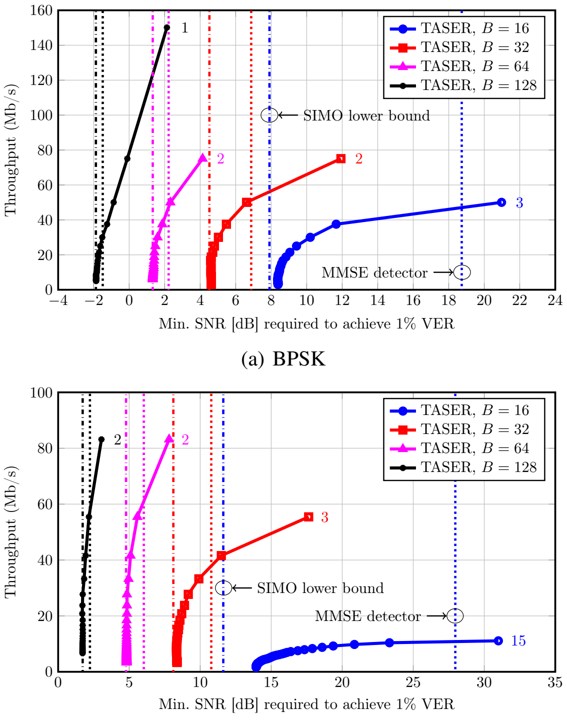
*2) Matrix multiplication:* This mode is used to compute line 4 of Algorithm 1. The multiplier uses the inputs from both broadcast signals. In the first cycle of the matrix-matrix multiplication procedure, the multiplier's output is subtracted from $\widetilde{L}_{i,j}^{(t-1)}$; in all other cycles, it is subtracted from the accumulator. In the $k$th cycle, all the PEs in the $k$th column use their internal $\widetilde{L}_{i,k}^{(t-1)}$ to feed the multiplier, instead of the signals coming from the RBU.

*3) Squared $\ell_2$-norm calculation:* This mode is used for line 5 of Algorithm 1. Both of the multiplier's inputs are $V_{i,j}^{(t)}$. For the D-PEs, the result is passed to the next PE in the same column. For the OD-PEs, the output of the multiplier is added to the value from the preceding PE in the same column; the result is sent to the next PE.

*4) Scaling:* This mode is used to complete line 5 of Algorithm 1. One of the multiplier's inputs is $V_{i,j}^{(t)}$ and the other is $D_{j,j}/\|\mathbf{v}_j\|_2$ (which was computed previously by the scale unit, being $\mathbf{v}_j$ the $j$th column of $\mathbf{V}^{(t)}$) received through the CBU. The result of this operation is $\widetilde{L}_{i,j}^{(t)}$ and is stored in every PE.

### *C. Implementation Details*

We use 14 bit fixed-point values in the entire design. All PEs except for the bottom row use 10 fraction bits to represent $L_{i,j}^{(t-1)}$ and $V_{i,j}^{(t)}$; the PEs in the bottom row use 9 fraction bits. For the element $\widetilde{L}_{N,N}$, we use a register as its value remains constant. There is no RBU for the first row. We implemented the inverse square-root in the scale unit using LUTs, which consist of $2^{11}$ entries with 14 bits per word.



(a) BPSK

(b) QPSK

Fig. 4. Throughput vs. performance trade-off for a 16 user system. Vertical dash-dot lines represent the SIMO lower bound; dashed lines represent linear MMSE performance. TASER outperforms linear detectors in almost all regimes. The number next to the points corresponds to the number of iterations.

TABLE I
IMPLEMENTATION RESULTS ON A XILINX VIRTEX-7 XC7VX690T FPGA FOR DIFFERENT TASER ARRAY SIZES

| Array size | $N=9$ | $N=17$ | $N=33$ | $N=65$ |
|---|---|---|---|---|
| BPSK users | $U=8$ | $U=16$ | $U=32$ | $U=64$ |
| QPSK users | $U=4$ | $U=8$ | $U=16$ | $U=32$ |
| Slices | 1 467 | 4 350 | 13 787 | 60 737 |
| LUTs | 4 790 | 13 779 | 43 331 | 149 942 |
| FFs | 2 108 | 6 857 | 24 429 | 91 829 |
| DSP48s | 52 | 168 | 592 | 2 208 |
| Max. clock frequency | 232 MHz | 225 MHz | 208 MHz | 111 MHz |
| Min. latency (cycles) | 16 | 24 | 40 | 72 |
| Max. throughput | 116 Mb/s | 150 Mb/s | 166 Mb/s | 98 Mb/s |

The RBUs and CBUs were implemented using multiplexers. The $i$th RBU requires a multiplexer with $i$ inputs, whose output connects to $i$ PEs. This results in larger fan-out for large values of $i$, eventually becoming the critical path of the systolic array; the same applies to the CBUs. To improve the overall throughput, we put registers at the inputs and outputs of the broadcast multiplexers.

## V. IMPLEMENTATION RESULTS AND COMPARISON

### *A. Error-Rate Performance*

Figures 3(a) and 3(b) show vector error rate (VER) simulation results for TASER (for $t_{\max}=20$ iterations) with BPSK and QPSK modulation, respectively. All simulations are for a $16\times16$ large-MIMO system (we use the notation $B\times U$) with i.i.d. flat Rayleigh fading. We also show the performance of the single-input multiple-output (SIMO) lower bound, ML detection (for BPSK only), exact SDR detection (2), and linear MMSE detection. We see that TASER achieves near-ML performance and outperforms MMSE detection

TABLE II
Comparison of large-MIMO detectors for $128 \times 8$ large-MIMO systems on a Xilinx Virtex-7 XC7VX690T FPGA

| Detection algorithm | TASER | TASER | CGLS [7] | Neumann [6] | CD [8] |
|---|---|---|---|---|---|
| Error-rate performance | Near-ML | Near-ML | Near-MMSE | Near-MMSE | Near-MMSE |
| Modulation scheme | BPSK | QPSK | 64-QAM | 64-QAM | 64-QAM |
| Preprocessing | No | No | Yes | Yes | Yes |
| Iterations | 3 | 3 | 3 | 3 | 3 |
| Slices | 1 467 (1.35 %) | 4 350 (4.02 %) | 1 094 (1 %) | 48 244 (44.6 %) | 13 447 (12.4 %) |
| LUTs | 4 790 (1.11 %) | 13 779 (3.18 %) | 3 324 (0.76 %) | 148 797 (34.3 %) | 23 955 (5.53 %) |
| FFs | 2 108 (0.24 %) | 6 857 (0.79 %) | 3 878 (0.44 %) | 161 934 (18.7 %) | 61 335 (7.08 %) |
| DSP48s | 52 (1.44 %) | 168 (4.67 %) | 33 (0.9 %) | 1 016 (28.3 %) | 771 (21.4 %) |
| Clock frequency | 232 MHz | 225 MHz | 412 MHz | 317 MHz | 262 MHz |
| Latency (clock cycles) | 48 | 72 | 951 | 196 | 795 |
| Throughput | 38 Mb/s | 50 Mb/s | 20 Mb/s | 621 Mb/s | 379 Mb/s |
| Throughput/LUTs | 7 933 | 3 629 | 6 017 | 4 173 | 15 821 |

(note that ML detection and exact SDR detection entail excessive complexity). We also show the fixed-point performance of our TASER design, which demonstrates virtually no implementation loss.

Figures 4(a) and 4(b) show the trade-off between the throughput of TASER and the minimum SNR required to achieve 1% VER. We also include the SIMO lower bound and the performance of linear MMSE detection as a reference; this detector serves as a fundamental performance limit of the conjugate gradient least-squares (CGLS) detector in [7], the Neumann-series detector in [6], and the recent coordinate-descent (CD) detector in [8]. The maximum number of TASER iterations $t_{\max}$ enables us to tune the performance/complexity trade-off; only a few iterations are sufficient to outperform linear detection. We furthermore see that TASER delivers near-ML performance, while achieving throughputs ranging from 5 Mb/s to 50 Mb/s, depending on the antenna configuration and modulation scheme.

### B. Implementation Results

To demonstrate the effectiveness of TASER, we developed several FPGA designs for systolic array sizes of $N = 9$, $N = 17$, $N = 33$ and $N = 65$, which either support 8, 16, 32, and 64 BPSK users, or 4, 8, 16, 32 QPSK users, respectively. The corresponding implementation results on a Xilinx Virtex-7 XC7VX690T are shown in Table I. As expected, the resource utilization increases quadratically with the array size $N$. For the $N = 9$ and $N = 17$ arrays, the critical path is in the PEs' MAC unit; for the $N = 33$ and $N = 65$ arrays, the critical path is in the row broadcast multiplexers, which limits the throughput of the $N = 65$ array that supports up to 64 BPSK users.

In Table II, we compare TASER to the CGLS detector [7], the Neumann-series detector [6], and the CD detector [8], which have been implemented on the same FPGA and for a $128 \times 8$ large-MIMO system. TASER achieves comparable throughput to the CGLS design and significantly lower latency than the Neumann-series and CD detectors. In terms of the hardware efficiency (measured in terms of throughput per FPGA LUTs), our design performs similarly to CGLS and Neumann, and inferior to the CD design. Nevertheless, when taking into account the error-rate performance (see Figures 3(a) and 3(b)), TASER significantly outperforms the error-rate performance of these reference designs for BPSK and QPSK constellations.

## VI. Conclusions

In this paper, we have implemented—to the best of our knowledge—the first MIMO data detector that uses semidefinite relaxation. We have proposed TASER, a novel data-detection algorithm, and a corresponding systolic array. Our reference FPGA implementation results show that TASER achieves comparable hardware-efficiency compared to existing large-MIMO data detectors, while providing near-ML performance. For systems supporting a large number of low-rate users (e.g., 16 user or more) where BPSK and QPSK transmission is sufficient, TASER provides a viable alternative to sub-optimal, linear data-detection methods. We conclude by noting that due to stringent space constraints, we have ignored soft-output detection and a convergence analysis of TASER; both of these issues will be addressed in a planned journal version of this paper [16].

## 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] M. Wu, C. Dick, J. Cavallaro, and C. Studer, "FPGA design of a coordinate descent data detector for large-scale MIMO," *submitted to ISCAS*, 2016.

[9] 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.

[10] B. Steingrimsson, Z.-Q. Luo, and K. M. Wong, "Soft quasi-maximum-likelihood detection for multiple-antenna wireless channels," *IEEE Trans. Sig. Proc.*, vol. 51, no. 11, pp. 2710–2719, Nov. 2003.

[11] Z.-Q. Luo, W.-k. Ma, A. M.-C. So, Y. Ye, and S. Zhang, "Semidefinite relaxation of quadratic optimization problems," *IEEE Sig. Proc. Mag.*, vol. 27, no. 3, pp. 20–34, May 2010.

[12] J. Jaldén and B. Ottersten, "The diversity order of the semidefinite relaxation detector," *IEEE Transactions on Information Theory*, vol. 54, no. 4, pp. 1406–1422, Apr. 2008.

[13] T. Goldstein, C. Studer, and R. G. Baraniuk, "A field guide to forward-backward splitting with a FASTA implementation," *arXiv preprint: 1411.3406*, Nov. 2014.

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

[15] A. Wiesel, Y. C. Eldar, and S. Shamai, "Semidefinite relaxation for detection of 16-QAM signaling in MIMO channels," *IEEE Sig. Proc. Letters*, vol. 12, no. 9, pp. 653–656, Sep. 2005.

[16] O. Castañeda, T. Goldstein, and C. Studer, "Approximate semidefinite relaxation for data detection in large wireless systems," *in preparation for a journal.*