Analog Integr Circ Sig Process
DOI 10.1007/s10470-012-9894-8

Low complexity scalable MIMO sphere detection through antenna detection reordering

Michael Wu · Chris Dick · Yang Sun · Joseph R. Cavallaro

Received: 29 February 2012 / Revised: 20 May 2012 / Accepted: 5 June 2012

Abstract This paper describes a novel low complexity scalable multiple-input multiple-output (MIMO) detector that does not require preprocessing and the optimal squared l^2-norm computations to achieve good bit error (BER) performance. Unlike existing detectors such as Flexsphere that use preprocessing before MIMO detection to improve performance, the proposed detector instead performs multiple search passes, where each search pass detects the transmit stream with a different permuted detection order. In addition, to reduce the number of multipliers required in the design, we use l^1-norm in place of the optimal squared l^2-norm. To ameliorate the BER performance loss due to l^1-norm, we propose squaring then scaling the l^1-norm. By changing the number of parallel search passes and using norm scaling, we show that this design achieves comparable performance to Flexsphere with reduced resource requirement or achieves BER performance close to exhaustive search with increased resource requirement.

Keywords MIMO detection · Spatial multiplexing · Soft output detection · FPGA · Sphere decoding

M. Wu (✉) · Y. Sun · J. R. Cavallaro
Electrical and Computer Engineering, Rice University Houston, Houston, TX 77005, USA
e-mail: mbw2@rice.edu

Y. Sun
e-mail: ysun@rice.edu

J. R. Cavallaro
e-mail: cavallar@rice.edu

C. Dick
Xilinx, Inc., 2100 Logic Drive, San Jose, CA 95124, USA
e-mail: chris.dick@xilinx.com

1 Introduction

Multiple-input multiple-output (MIMO) is a key technique of many wireless standards such as 3G LTE, WiMAX and 802.11n. By exploiting multipath propagation, MIMO uses multiple transmit antennas and receive antennas to improve both reliability and throughput of a receiver. As the received signal at each antenna consists of a combination of multiple data streams from multiple transmit antennas, the main challenge for a MIMO receiver is to recover the transmit signals from the receive signals. This challenge, the MIMO detection problem, is known to be an integer least-squares problem that can be solved with an exhaustive search. An exhaustive search algorithm, however, is cost prohibitive for wireless system with strict latency and power requirements. As a result, a number of suboptimal MIMO detection algorithms have been developed to reduce the computational complexity.

The search process can be viewed a tree traversal where unlikely paths with large distances are excluded and likely path with small distances are kept. The traversal algorithms can be classified into two categories: depth-first algorithms such as depth-first sphere detection [1], and breadth-first algorithms such as K-best [2]. The main problem with depth-first sphere detection is that the number of tree nodes visited is large in the low signal to noise ratio (SNR) range, while the number of tree nodes visited is small in the high SNR range. As a result, depth-first sphere detection algorithms typically have variable throughput which is undesirable in systems with strict latency requirements. The alternative is a fixed throughput breadth-first algorithm, K-best detection, which visits a constant number of nodes independent of SNR. As a large K value is required to achieve performance close to exhaustive search, the primary challenge of a high throughput low latency K-best

Published online: 30 June 2012

detector design is sorting required to find the K best path at each step of the algorithm.

To address the sorting complexity of the K-best detection algorithm, a number of modified algorithms [3, 4] attempts to find a fast approximation to sort. For example, instead of sorting N values to find K smallest values, SSFE [3] partitions N values into M arrays, where M is the modulation order. Fast enumeration is used to find the best K/M values for each subarray without sorting each subarray. However, this approximation reduces the BER performance of the detector. To recover some of the performance loss, authors in [5] present a hard decision detector called Flexsphere. While the search process is still SSFE, Flexsphere perform V-BLAST-like antenna reordering and modified real-value decomposition (RVD) to improve BER performance. In addition, the design also use l^1-norm in place of the optimal squared l^2-norm to reduce the number of multipliers required. The final design can achieve accuracy close to optimal exhaustive search for hard decision.

Since a typical wireless system usually combines a soft-output detector and a soft-input decoder to maximize performance, we aim to design a soft output MIMO detector by extending the design outlined in [5]. An existing implementation of the Flexsphere detector [6] for WiMAX shows that the V-BLAST-like preprocessing block uses significant amount of FPGA resources due to matrix inversions, while the QR decomposition and detection blocks combined use significantly less FPGA resources. In addition, to generate soft-output, the candidate list generated by Flexsphere does not guarantee bit-level reliability information, the log likelihood ratio (LLR), cannot be found for all bits, leading to the need for LLR clipping [7–9]. In addition, soft-output values should be generated with squared l^2-norm. As shown in [10], although l^1-norm is accurate enough for hard decision, it leads to severe performance loss for soft-output detection. The author in [10] propose scaling l^1-norm heuristically through measured data. We aim for an analytical method.

In our previous work [11], we proposed the following change to reduce complexity. Instead of computing the optimal antenna detection order with a V-BLAST-like preprocessing block before detection, we schedule multiple search passes through the search space, where each search uses a different permuted antenna detection order. In this paper, we describe this improved design in detail. In addition, as l^1-norm reduce performance for soft-output detection, we propose generating more accurate soft-output values by squaring then scaling the l^1-norm. We show that compared to the design in [6], the proposed design can either achieve comparable performance with reduced resource requirement, or it can achieve better performance with increased resource requirement. In addition, we will show this design can avoid the problem of LLR clipping when the number of search passes through the search space is the same as the number of antennas.

This paper is organized as follows: Section 2 gives an overview of the system model. Section 3 describes the proposed detection algorithm. Section 4 presents the performance of the proposed detector. In Sect. 5, we present the corresponding FPGA implementation. Finally, we conclude in Sect. 6.

2 MIMO system model

For an $N_r \times N_t$ MIMO system, the source transmits N_t signals and the destination receives signals on N_r antennas. The received signal, $\mathbf{y} = [y_0, y_1, \ldots, y_{N_r-1}]$, can be modeled by

$$\mathbf{y} = \mathbf{Hs} + \mathbf{n}, \tag{1}$$

where $\mathbf{H} = [\mathbf{h_0}, \mathbf{h_1}, \ldots, \mathbf{h_{N_t-1}}]$ is the $N_r \times N_t$ channel matrix, in which each element of the column vector $\mathbf{h_i}$, h_{ij}, is an i.d.d. zero mean circular symmetric complex Gaussian random variable (ZMCSCG) with σ_h^2 variance. Given a binary vector $\mathbf{x} = [x_0, x_1, x_2 \ldots x_{L-1}]$, where $L = \log_2 M \cdot N_t$, the function map$(\cdot)$ translates the binary vector $\mathbf{x}$ onto $\mathbf{s} = [s_0, s_1, \ldots, s_{N_t-1}]$. Each element of $\mathbf{s}$, s_i, is an element drawn from a finite alphabet $\mathbf{\Omega}$ with cardinality M and average power E_s per symbol. For example, the constellation alphabet for QPSK is $\{-1-j, -1+j, 1-j, 1+j\}$ with $M = 4$. Finally, the receiver noise, $\mathbf{n} = [n_0, n_1, \ldots, n_{N_r-1}]$, is an independent ZMCSCG with $\frac{\sigma_n^2}{2}$ variance per dimension.

We first obtain an equivalent system model in the real domain by performing real-valued decomposition:

$$\begin{pmatrix} \Re(\mathbf{y}) \\ \Im(\mathbf{y}) \end{pmatrix} = \begin{pmatrix} \Re(\mathbf{H}) & -\Im(\mathbf{H}) \\ \Im(\mathbf{H}) & \mathbf{R}(\mathbf{H}) \end{pmatrix} \begin{pmatrix} \Re(\mathbf{s}) \\ \Im(\mathbf{s}) \end{pmatrix} + \begin{pmatrix} \Re(\mathbf{n}) \\ \Im(\mathbf{n}) \end{pmatrix} \tag{2}$$

We then permute the real-value decomposed channel matrix such that the in-phase and the quadrature parts of the same complex symbol are adjacent neighbors [5]:

$$\begin{pmatrix} \Re(y_0) \\ \Im(y_0) \\ \Re(y_1) \\ \Im(y_1) \\ \cdot \\ \cdot \\ \cdot \\ \Re(y_{N_R-1}) \\ \Im(y_{N_R-1}) \end{pmatrix} = \tilde{\mathbf{H}} \begin{pmatrix} \Re(s_0) \\ \Im(s_0) \\ \Re(s_1) \\ \Im(s_1) \\ \cdot \\ \cdot \\ \cdot \\ \Re(s_{N_R-1}) \\ \Im(s_{N_R-1}) \end{pmatrix} + \begin{pmatrix} \Re(n_0) \\ \Im(n_0) \\ \Re(n_1) \\ \Im(n_1) \\ \cdot \\ \cdot \\ \cdot \\ \Re(n_{N_R-1}) \\ \Im(n_{N_R-1}) \end{pmatrix} \tag{3}$$

$$\tilde{\mathbf{y}} = \tilde{\mathbf{H}}\hat{\mathbf{s}} + \tilde{\mathbf{n}} \tag{4}$$

In the real-value model, the number of elements in each vector is doubled and both dimensions of the matrix $\tilde{\mathbf{H}}$ are doubled. Furthermore, each element of the equivalent transmit vector, $\tilde{s}_i$, is an element drawn from a finite alphabet $\mathbf{\Omega}'$ with cardinality $Q = \sqrt{M}$.For example, the real value decomposed constellation alphabet for QPSK is $\{-1,1\}$ and $Q = 2$.

Given $\tilde{\mathbf{y}}$ and the channel matrix $\tilde{\mathbf{H}}$, the goal of the soft-output MIMO detector at a MIMO receiver is to compute the logarithmic a-posteriori probability (APP) ratio, $L_D(x_k|\tilde{\mathbf{y}}, \tilde{\mathbf{H}})$, per bit. Assuming no prior knowledge of the transmitted bits, the soft-output value per bit can be approximated with the following equation using max-Log approximation [12]:

$$L_D(x_k|\tilde{\mathbf{y}}, \tilde{\mathbf{H}}) \approx \min_{\mathbf{x}\in\mathbb{X}_{k,-1}} \frac{\|\tilde{\mathbf{y}} - \tilde{\mathbf{H}}\tilde{\mathbf{s}}\|_2^2}{2\sigma_n^2} - \min_{\mathbf{x}\in\mathbb{X}_{k,+1}} \frac{\|\tilde{\mathbf{y}} - \tilde{\mathbf{H}}\tilde{\mathbf{s}}\|_2^2}{2\sigma_n^2}, \tag{5}$$

where $\mathbb{X}_{k,-1}$ is the list of all binary vectors with the kth component equal to -1, $\mathbb{X}_{k,+1}$ is the list of all binary vectors with the kth component equal to $+1$, and $\tilde{\mathbf{s}} = \mathrm{map}(\mathbf{x})$.

3 Proposed soft-output N-way MIMO detector

Although max-Log approximation reduces complexity of the detector, evaluating $L_D(x_k|\tilde{\mathbf{y}}, \tilde{\mathbf{H}})$ with Eq. 5 is still computationally intensive as the detector needs to search through the set of all possible binary vectors to find the hypothesis and the counter-hypothesis per transmitted bit. The complexity of the detector can be reduced by constraining the size of the set. A soft-output MIMO detector first finds a smaller set of transmit vectors, or a candidate list, $\mathcal{L}$, by eliminating vectors with large distances. To compute $L_D(x_k|\mathbf{y}, \mathbf{H})$, the candidate list is divided into $\mathcal{L}_{k,-1}$ and $\mathcal{L}_{k,+1}$, where $\mathcal{L}_{k,-1}$ is the list of candidates with the k^{th} bit equal to -1 and $\mathcal{L}_{k,+1}$ is the list of candidates with the kth bit equal to $+1$. We find the candidate with the smallest distance for each set and the difference between the two distances is the soft value:

$$L_D(x_k|\tilde{\mathbf{y}}, \tilde{\mathbf{H}}) \approx \underbrace{\min_{\mathbf{x}\in\mathcal{L}_{k,-1}} \frac{\|\tilde{\mathbf{y}} - \tilde{\mathbf{H}}\tilde{\mathbf{s}}\|_2^2}{2\sigma_n^2}}_{hypothesis} - \underbrace{\min_{\mathbf{x}\in\mathcal{L}_{k,+1}} \frac{\|\tilde{\mathbf{y}} - \tilde{\mathbf{H}}\tilde{\mathbf{s}}\|_2^2}{2\sigma_n^2}}_{counter-hypothesis}. \tag{6}$$

Although SSFE performs close to exhaustive search as a hard decision detector, it does not perform close to exhaustive search since the generated candidate list size is very small. When computing soft-output for the kth bit, the list $\mathcal{L}_{k,-1}$ or the list $\mathcal{L}_{k,+1}$ may be an empty set, in which case $L_D(x_k|\tilde{\mathbf{y}}, \tilde{\mathbf{H}})$ cannot be computed since the hypothesis or the counter-hypothesis is unknown. When an empty set occurs, we can still generate an LLR value by clipping the LLR to a predetermined value [8]. This, however, leads to performance degradation, especially when the clipping value is not picked appropriately. Furthermore, $L_D(x_k|\tilde{\mathbf{y}}, \tilde{\mathbf{H}})$ may not be accurately approximated due to the small candidate list size. Although the V-BLAST-like preprocessing in Flexsphere [5] ensures optimal detection order, the preprocessing block is very expensive. Furthermore, to reduce the number of multipliers required during the search, l^1-norm [1] is used in place of l^2-norm (squared Euclidean distance). For soft-output, we can use l^1-norm in the APP computation:

$$L_D(x_k|\tilde{\mathbf{y}}, \tilde{\mathbf{H}}) \approx \min_{\mathbf{x}\in\mathcal{L}_{k,-1}} \frac{\|\tilde{\mathbf{y}} - \tilde{\mathbf{H}}\tilde{\mathbf{s}}\|_1}{2\sigma_n^2} - \min_{\mathbf{x}\in\mathcal{L}_{k,+1}} \frac{\|\tilde{\mathbf{y}} - \tilde{\mathbf{H}}\tilde{\mathbf{s}}\|_1}{2\sigma_n^2}. \tag{7}$$

Since the hypothesis and the counter-hypothesis for each bit are a function of l^2-norm, the use of l^1-norm reduces BER performance significantly.

In the proposed detection algorithm, we leverage the fact that the search algorithm is cheap and propose a novel detection method in which the preprocessing block is removed in favor of performing multiple search passes. In addition, we keep the reduce the performance loss due to l^1-norm by squaring then scaling l^1-norm before applying Eq. 7 to compute APP per bit.

3.1 MIMO detection search pass

The proposed MIMO detection scheme uses the same search algorithm as Flexsphere and SSFE. Since the search algorithm is discussed in-depth in [5], we will give an overview of the algorithm in this section.

Given $\tilde{\mathbf{y}}$ and $\tilde{\mathbf{H}}$, we first perform QR decomposition on $\tilde{\mathbf{H}}$ for an equivalent system model, where the l^1-norm distance of a transmit vector $\tilde{\mathbf{s}}$ is:

$$d(\tilde{\mathbf{s}}) = \|\hat{\mathbf{y}} - \mathbf{R}\tilde{\mathbf{s}}\|_1. \tag{8}$$

where $\mathbf{R}$, an upper-triangular matrix, is the effective channel matrix and $\hat{\mathbf{y}}$ is the effective received vector.

To generate a smaller candidate list, the MIMO detector excludes transmit vectors with large l^1-norm distances and searches for transmit vectors with small l^1-norm distances. During the search process, the MIMO detector can evaluate the l^1-norm distance of a transmit vector level by level since $\mathbf{R}$ is upper triangular. As a result, the search process can be viewed as a traversal through a tree. The MIMO detection search algorithm can be viewed as a greedy tree search. The algorithm traverses the tree breadth first and prunes unlikely branches level by level until there are a few complete paths left. Figure 1 demonstrates an example of

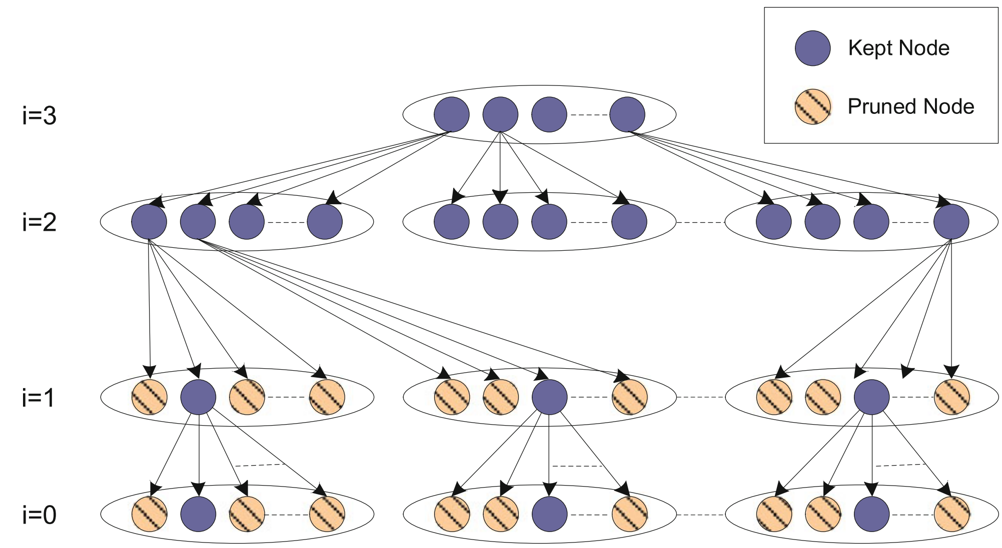


Fig. 1 Proposed parallel search

the search process for a 2 × 2 16-QAM MIMO system. Since all branches in the first two levels are kept, the first two levels of the tree are fully expanded. At the subsequent levels, the algorithm evaluates all branches out of each node and prunes unlikely branches with large l^1-norm distances. The selectivity of the pruning process is controlled by how many branches a node keeps in the pruning process. This search algorithm only keeps the best branch for each node after the first two tree levels. All surviving paths at the last tree level are in our candidate list.

We will now describe the pruning function. Figure 2 shows the data flow at the jth node at stage i. Given one incoming path with path history $\mathbf{p} = (p_0, p_1, \ldots, p_i)$ and distance d_m, we wish to extend the incoming path to the next level $i + 1$ by picking the best out-going path among Q paths. The updated cumulative weight after connecting node p_i to the kth node in level $i + 1$ is:

$$d'_k = d_m + w_{j,k}^{<i+1>}, \quad 0 \le k \le Q-1, \tag{9}$$

where $w_{i,k}^{<i+1>}$ is defined as:

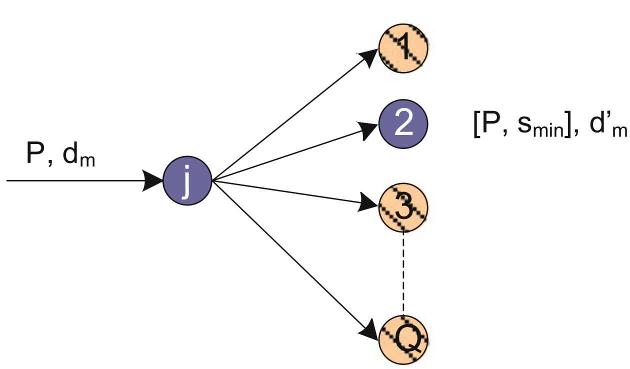


Fig. 2 An example of the pruning process at node j

$$w_{j,k}^{<i+1>} = \left\| \hat{y}_k - R_{k,k}s_k - \sum_{j=k+1}^{M} R_{k,j}p_j \right\|_1, \tag{10}$$

$$= ||b_{i+1}(\mathbf{p}) - R_{k,k}s_k||_1. \tag{11}$$

The best connected node, $s_{\min}$, in level $t + 1$ that minimizes $w_{i,k}^{<i+1>}$ is simply the closest constellation point in Ω' to $\frac{b_{i+1}(\mathbf{p})}{R_{k,k}}$. The algorithm to find the best node can be implemented with a simple round function,

$$\tilde{s}_{min} = \text{Round}\left[\frac{1}{2}\left(\frac{b_{i+1}(\mathbf{p})}{R_{k,k}} + Q - 1\right)\right] \cdot 2 - Q + 1, \tag{12}$$

followed by a threshold function:

$$s_{\min} = \begin{cases} \tilde{s}_{\min} & |\tilde{s}_{\min}| < Q-1 \\ \text{sign}(\tilde{s}_{\min})(Q-1) & |\tilde{s}_{\min}| \ge Q-1 \end{cases} \tag{13}$$

When the best node is found, the path history at level $i + 1$ is updated by appending the best outgoing node to $\mathbf{p}$ and the Euclidean distance at level $i + 1$ is updated by saving d'_m as d_m.

3.2 N-way scheduled search

For the proposed N-way scheduled MIMO detector, we remove the V-BLAST-like preprocessing block in Flexsphere in favor of running multiple search passes in parallel. The proposed algorithm is summarized in Algorithm 1. The inputs for all search passes are the same, consisting of the channel matrix $\mathbf{H}$ and the received vector $\mathbf{y}$. Since the detection order affects the performance of the detector, we schedule multiple search passes where each pass uses a different antenna detection order to generate different lists of candidates. Without the V-BLAST-like preprocessing block, the optimal detection order is not known. We

Algorithm 1 Proposed MIMO detection algorithm

Initialization: $\mathcal{L} \leftarrow$
for i = 0 to N-1 **do**
  $\mathbf{y_i} \leftarrow$ circular rotate rows of $\mathbf{y}$ i times
  $\mathbf{H}_i \leftarrow$ circular rotate columns of $\mathbf{H}$ i times
  $(\hat{\mathbf{y}}_i, \mathbf{R}_i) \leftarrow \mathrm{MRVD} - \mathrm{QRD}(\mathbf{y}_i, \mathbf{H}_i)$
  $\mathcal{L}_i \leftarrow \mathrm{search}(\hat{\mathbf{y}}_i, \mathbf{R}_i)$
  $\mathcal{L} \leftarrow \mathcal{L} \bigcup \mathcal{L}_i$
end
Compute LLR values using candidate list $\mathcal{L}$

propose an antenna detection order that can be obtained by a simple circular rotation of columns of $\mathbf{H}$. A search pass, consisting of RVD decomposition, QR decomposition and tree search, is performed for each permutation. The result is the ith search pass processes the ith antenna first. Since $\mathbf{H}$ has N_t columns, the proposed detector runs up to N_t passes for N_t possible permutations. Each detection pass generates more candidates for the candidate list. The total number of candidates generated is NC, where C is the number of candidate generated by one search pass and N is the number of search passes. Figure 3 illustrates the blocks required for the $N = N_r$ case.

After each search pass, the hypothesis and the counter-hypothesis for each bit are updated using the generated candidate lists. After N search passes, the LLR values are computed for each bit by finding the difference between the hypothesis and the counter-hypothesis according to Eq. 7. Compared to Flexsphere, a larger list increases the probability of finding the best hypothesis and the counter-hypothesis per transmitted bit. We note that each search pass guarantees that the hypothesis and the counter-hypothesis can be found for the bits corresponding to the first antenna detected. The first two levels of the tree are completely expanded. Due to the property of the pruning function, there is a complete path through all nodes in the first two levels of the tree at the end of the search. This means the candidate list will have a candidate through each possible symbol transmitted by the first antenna. This

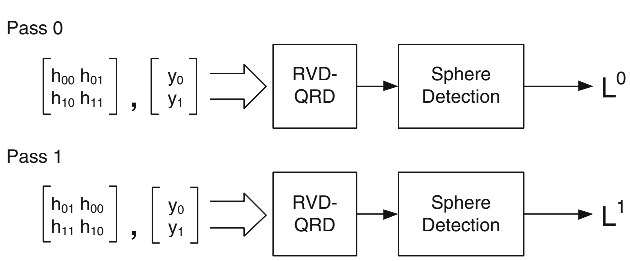


Fig. 3 Proposed detector for a 2 × 2 MIMO system

property is illustrated by Fig. 1. As a result, the two sublists for each bit transmitted by the first antenna level are always non-empty and the hypothesis and the counter-hypothesis both can be found. The LLR values corresponding to the antenna expanded first by each of the search passes do not require clipping. When $N = N_t$, there is no need for LLR clipping since each antenna is fully expanded once and each bit has a hypothesis and a counter-hypothesis, increasing performance.

3.3 L1 transformation

The use of l^1-norm in place of squared l^2-norm reduces the accuracy of the soft-value generated. Instead of using l^1-norm directly as in 7, we propose using squared then scaled version of l^1-norm to increase accuracy of the soft-value generated:

$$L_D(x_k|\tilde{\mathbf{y}}, \tilde{\mathbf{H}}) \approx \min_{\mathbf{x} \in \mathcal{L}_{k,-1}} \frac{\gamma\|\hat{\mathbf{y}} - \mathbf{R}\tilde{\mathbf{s}}\|_1^2}{2\sigma_n^2} - \min_{\mathbf{x} \in \mathcal{L}_{k,+1}} \frac{\gamma\|\hat{\mathbf{y}} - \mathbf{R}\tilde{\mathbf{s}}\|_1^2}{2\sigma_n^2}. \tag{14}$$

We propose picking a scaling value γ such that the expected value of the transformed l^1-norm is the same as the expected value of the optimal squared l^2-norm:

$$\mathbb{E}[\|\hat{\mathbf{y}} - \mathbf{R}\tilde{\mathbf{s}}\|_2^2] = \gamma\mathbb{E}[\|\hat{\mathbf{y}} - \mathbf{R}\tilde{\mathbf{s}}\|_1^2] \tag{15}$$

The value γ is the ratio between the squared l^1-norm and squared l^2-norm. This ratio is a fixed value which depends on the receive antenna. We will now solve for γ:

We first compute for $\mathbb{E}[\|\hat{\mathbf{y}} - \mathbf{R}\tilde{\mathbf{s}}\|_2^2]$:

$$\mathbb{E}[\|\hat{\mathbf{y}} - \mathbf{R}\tilde{\mathbf{s}}\|_2^2] = \mathbb{E}[(\hat{\mathbf{y}} - \mathbf{R}\tilde{\mathbf{s}})^T(\hat{\mathbf{y}} - \mathbf{R}\tilde{\mathbf{s}})] \tag{16}$$

$$= \mathbb{E}[(\mathbf{Q^T n})^T(\mathbf{Q^T n})] \tag{17}$$

$$= \mathbb{E}[\mathbf{n}^T Q^T Q n] \tag{18}$$

$$= \sum_{i=0}^{N_r} \mathbb{E}[n_i n_i] \tag{19}$$

$$= N_r \sigma_n^2 \tag{20}$$

We then compute for $\|\hat{\mathbf{y}} - \mathbf{R}\tilde{\mathbf{s}}\|_1^2$, the expected value of the square of l^1-norm. The square of l^1-norm can be written as:

$$\mathbb{E}[\|\hat{\mathbf{y}} - \mathbf{R}\tilde{\mathbf{s}}\|_1^2] = \mathbb{E}\left[\left(\sum_{i=0}^{N_r-1} |\mathbf{q_i n}|\right)^2\right], \tag{21}$$

where $\mathbf{q_i}$ is a row of $\mathbf{Q}$. The summation can be by grouping like terms together:

$$\mathbb{E}[\|\hat{\mathbf{y}} - \mathbf{R}\tilde{\mathbf{s}}\|_1^2] = \mathbb{E}\left[\left(\sum_{i=0}^{N_r-1} |\mathbf{q_i n}||\mathbf{q_i n}|\right)\right] + \mathbb{E}\left[\left(\sum_{i,j,i\neq j} |\mathbf{q_i n}||\mathbf{q_j n}|\right)\right]. \quad (22)$$

Since $\mathbf{Q}$ is a unitary matrix, the variable $\mathbf{q_i n}$ is also a zero mean Gaussian random variable with the same variance as n_i. We can now solve for $\mathbb{E}[\|\hat{\mathbf{y}} - \mathbf{R}\tilde{\mathbf{s}}\|_1^2]$:

$$\mathbb{E}[\|\hat{\mathbf{y}} - \mathbf{R}\tilde{\mathbf{s}}\|_1^2] = \sum_{i=0}^{N_r-1} \mathbb{E}[n_i n_i] + \mathbb{E}\left[\left(\sum_{i,j,i\neq j} |n_i||n_j|\right)\right], \quad (23)$$

$$= \sum_{i=0}^{N_r-1} \mathbb{E}[n_i n_i] + 2\binom{N_r}{2}\mathbb{E}[|n_i||n_j|], \quad (24)$$

$$= N_r\sigma_n^2 + 2\binom{N_r}{2}\frac{2}{\pi}\sigma_n^2. \quad (25)$$

We can now solve for γ:

$$\gamma = \frac{N_r}{N_r + 2\binom{N_r}{2}\frac{2}{\pi}}, \quad (26)$$

$$= \frac{1}{1 + (N_r - 1)\frac{2}{\pi}}. \quad (27)$$

We measured the quality of these two scaling through numerical simulation two ways. Given two vectors x and y, the mean square error (MSE) metric is defined as $\sum_i (x_i - y_i)^2$. The other metric is maximum relative error defined as $\max_i(\frac{|x_i - y_i|}{y_i})$. We use a vector of eight i.d.d Gaussian unit variance random variables as the test case and measured the average MSE and MRE between l^1-norm the l^2-norm. In addition, we measured MSE and MRE between transformed l^1-norm and l^2-norm. As seen in Table 1, l^1-norm without any scaling performs much worse, suggests l^1-norm is not a good approximation of squared l^2-norm.

We also derived the ratio that minimizes the MSE between the squared l^1-norm and the squared l^2 norm. Since the difference between the ratios is very minimal, the derivation and the result are omitted in this paper.

4 Performance

We compared the BER performance of our proposed N-way scheduled MIMO detector against the soft-output Flexsphere detector and the K-best detector in a flat fading Rayleigh fading channel. In our simulation, the soft output of the detector is fed to a length 2304, rate 1/2 WiMAX LDPC decoder, which performs up to 20 decoding iterations. For the K-best detector, we choose a large K value of 64 for 16-QAM and 256 for 64-QAM. For the proposed algorithm, we show the results for up to 4 detection passes. We used an LLR clipping value of 8 for all the detector configurations with the exception of the proposed detection algorithm with 4 detection passes where LLR clipping was not required.

Table 1 MSE and MRE of l^1-norm and transformed l^1-norm

	l^1-norm	Transformed l^1-norm
MSE	2.201	0.504
MRE	4.341	0.684

We first look at the performance of the detector when the optimal l^2-norm is used. Figure 4(a) compares the performance of detectors for 16-QAM. We see that Flexsphere gains more than 1 dB of performance at FER of 10^{-3} compared to the N-way scheduled MIMO detector where N = 1. This shows the benefits of V-BLAST-like preprocessing. The proposed N-way scheduled MIMO detector with N = 2 performs better than Flexsphere, while the proposed N-way scheduled MIMO detector with N = 3 performs better than the K-best MIMO detector. Finally, the proposed detector with N = 4 performs within 0.25 dB of exhaustive search at FER of 10^{-3}. This is expected as N = 4 avoids LLR clipping and generates more accurate LLR values due to a larger candidate list. Figure 4(b) compares the performance of the detector for 64-QAM. The result is similar to 16-QAM. For N = 2, the N-way scheduled MIMO detector performs similarly to the soft-output Flexsphere. For N = 3, the N-way scheduled MIMO detector performs similarly to the K-best detector. For N = 4, the N-way scheduled MIMO detector's performance is close to exhaustive search. The results suggest that we can remove the costly V-BLAST-like preprocessing and improve performance by increasing the number of detection passes.

We then look at the performance of the detector when the non-optimal l^1-norm and the squared then scaled l^1-norm are used. As shown in Fig. 5, there is a significant improvement between transformed l^1-norm and l^1-norm. By scaling, we can achieve performance within 1 dB of a soft-output detector using l^2-norm squared. This is expected as the transformed l^1-norm is much closer than l^1-norm.

5 FPGA hardware implementation

In this paper, we implemented our design on a Virtex®-5 XC5VFX130T-2FF1738 FPGA. A complete soft-output Flexsphere detector is implemented with Xilinx System Generator by extending the hard-decision Flexsphere design presented by authors in [6]. The hard-decision Flexsphere detector design consists of three components: a channel preprocessor, RVD/QRD block, and a sphere

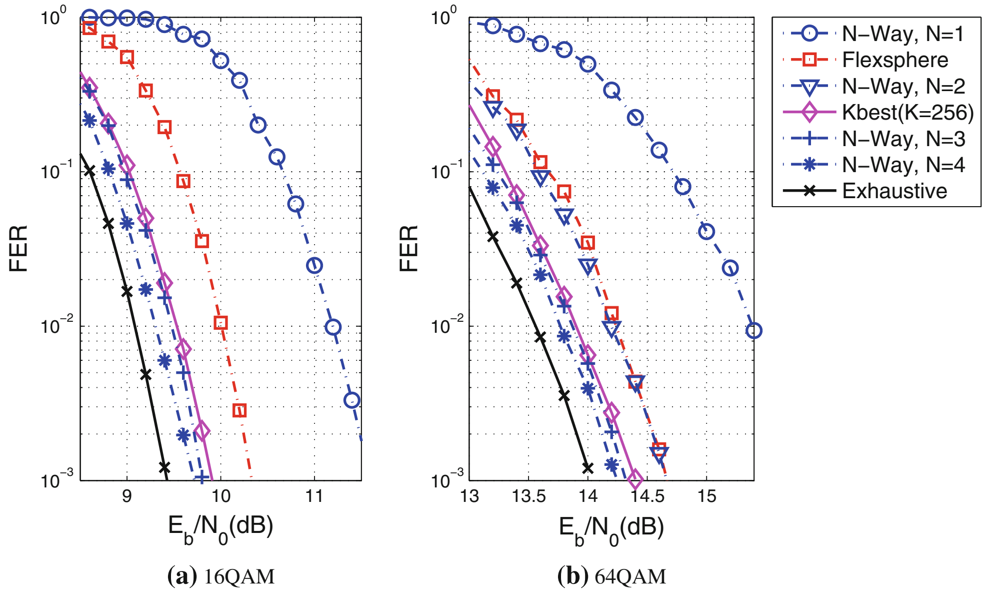


Fig. 4 FER performance, squared l^2-norm

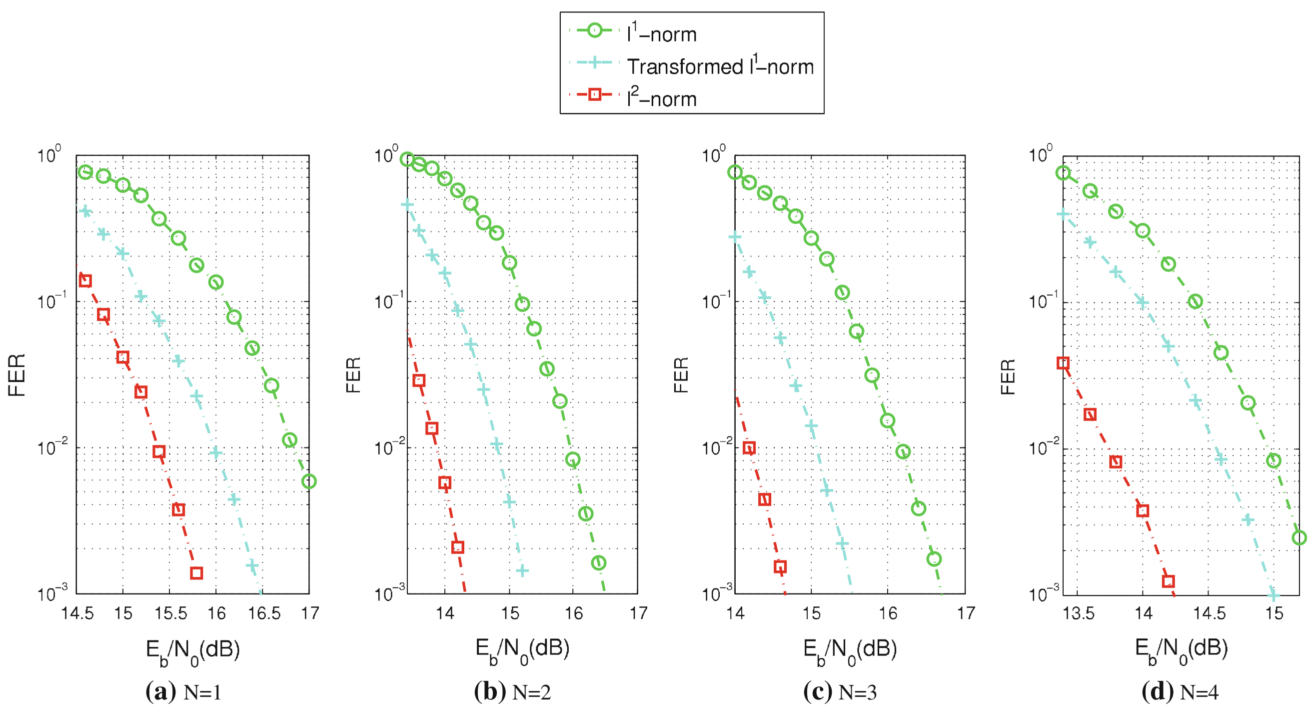


Fig. 5 FER performance of l^1-norm, squared l^2-norm, and transformed l^1-norm for different N

detector block. This hard decision design outputs the constellation points of the candidate with the smallest distance among the 64 candidates at the last level. Instead of outputting the candidate with the smallest distance, we added an LLR computation block which uses the 64 candidates and the associated distances to generate soft-output LLR values.

To meet the target data rate of 83.965 Mbps which corresponds to a 360 sub-carriers WiMAX system, the sphere detector has a throughput of one candidate per cycle. To match the data rate, the LLR generator needs to process 64 candidates to generate 24 bit-level soft values every 64 cycles with a minimal clock frequency of 225 MHz. The LLR generator is straight forward. Since the sphere detector uses l^1-norm as the distance metric for each candidate, we first square then scale the distance metric associated with each candidate. We then demodulate each incoming candidate into bits using a look-up table. To find the hypothesis and the counter-hypothesis, we allocate two registers per demodulated bit. In total, we have 48 18-bit registers which are partitioned into two sets–one set for the hypothesis and one set for the counter-hypothesis. Depending on whether a demodulated bit is 0 or 1, we compare the Euclidean distance against the value in the corresponding register. If the current Euclidean distance is smaller, the value in the register is replaced by the current Euclidean distance of the candidate. All 24 bits are evaluated in parallel to meet the throughput requirement. The resources required for the blocks from [6] and the LLR computation block are shown in Table 2.

Since the proposed design is a modified Flexsphere detector, we used the design proposed by authors in [6] to implement our current proposed design. The RVD-QRD, the sphere detector, and the LLR Computation blocks are reused while the channel preprocessor is removed. Each search pass is one pass through all three blocks. Therefore, the throughput of an implementation with one copy of these three blocks is reduced N times for N search passes.

To improve the achievable throughput of the proposed detector, we use more resources and perform search passes in parallel. We simply replicate the serial implementation *N* times, where each instance of the detector performs a detection pass independently to generate the hypothesis and the counter-hypothesis per transmitted bit. The output is N lists of the hypotheses and counter-hypotheses per transmitted bit that need to be merged into one list. The merger block is a fairly simple block, where the minimum value among N values is found for each transmitted bit.

Table 2 Resource usage of soft-output Flexsphere

Block	Slices	LUTs/FFs	DSP48	Block RAM
Preprocessing	14,684	28,801/39,563	120	54
RVD/QRD	4,233	5,672/5,556	30	20
Sphere detector	2,678	4,767/6,031	41	12
LLR computation	1,667	2,042/3,991	2	4
Total	22,676	42,090/53,095	191	86

Table 3 Resource usage of merge blocks

N	Slices	LUTs/FFs	DSP48	Block RAM
2	65	100/201	0	0
3	137	208/401	0	0
4	202	312/602	0	0

Table 4 Total resource usage of N-way sphere detector

N	Slices	LUTs/FFs	DSP48	Block RAM
1	7,992	13,289/13,532	71	32
2	16,515	28,566/28,411	142	64
3	24,939	43,151/42,808	213	96
4	33,174	57,736/57,205	284	128

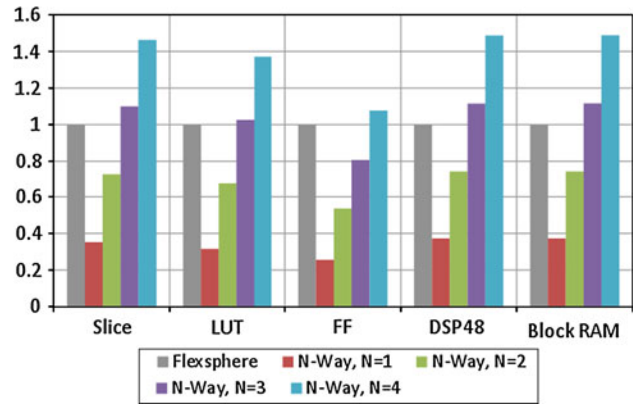


Fig. 6 Resource comparison

Since the resource cost depends on N, the resources required by merging N lists are listed in Table 3.

The overall resource required is approximately N times the serial implementation of the proposed detector plus the cost of merger block for value N and is shown in Table 4.

Figure 6 shows the cost of different realizations of the detector compared to the soft-output Flexsphere, where the cost of the Flexsphere is normalized to one. To achieve comparable performance to soft-output Flexsphere with preprocessor, we need at least two search passes. Although we need an additional RVD/QRD block, sphere detector block and LLR generator block for N = 2, we save a substantial amount of resources compared to the Flexsphere detector because we eliminate the expensive preprocessing block. The amount of resources required for N = 4 increased around 50 percent compared to the soft-output Flexsphere detector. However, increasing the number of search passes to four increases the performance of the detector by another 0.5 dB over the soft-output Flexsphere design.

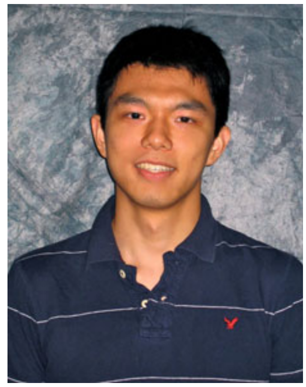
6 Conclusion

In this paper, we presented a scalable detection algorithm that does not require preprocessing to achieve good performance. We propose scheduling search passes with different antenna detection order, where the *i*th antenna is positioned as the first layer for the *i*th search pass. We show that by changing the number of search passes, we can achieve BER performance 0.25 dB from exhaustive search and eliminate LLR clipping. We also show that we can achieve comparable performance to Flexsphere with two search passes and eliminate the costly processing, resulting in a more area-efficient design. In addition, in our design, we maintain the use of l^1-norm to reduce complexity of our FPGA implementation. We show that we can recover some performance loss through squaring then scaling the l^1-norm such that we are within 1dB of the optimal l^2 norm.

Acknowledgments The work of the Rice University authors was supported in part by the National Science Foundation under grants EECS-0925942 and CNS-0923479.

References

1. Burg, A., Borgmann, M., Wenk, M., Zellweger, M., Fichtner, W., & Bolcskei, H. (2005). VLSI implementation of MIMO detection using the sphere decoding algorithm. *IEEE Journal of Solid-State Circuit, 40*, 1566–1577.
2. Wong, K., Tsui, C., Cheng, R., & Mow, W. (2002). A VLSI architecture of a K-best lattice decoding algorithm for MIMO channels. In *IEEE International Symposium on Circuits and Systems, Kos* (Vol. 3, pp. 273–276). Kos.
3. Li, M., Bougard, B., Lopez, E., Bourdoux, A., Novo, D., Van Der Perre, L., & Catthoor, F. (2008). Selective spanning with fast enumeration: A near maximum-likelihood MIMO detector designed for parallel programmable baseband architectures. In *ICC '08, IEEE International Conference on Communications, Beijing*. Beijing.
4. Sun, Y., & Cavallaro, J. R. (2009). High throughput vlsi architecture for soft-output mimo detection based on a greedy graph algorithm. In *Proceedings of the 19th ACM Great Lakes symposium on VLSI, GLSVLSI '09, Boston*, Boston: ACM.
5. Amiri, K., Dick, C., Rao, R., & Cavallaro, J. R. (2011). A high throughput configurable SDR detector for multi-user MIMO wireless systems. *Journal of Signal Processing Systems, 62*, 233–245.
6. Dick, C., Trajkovic, M., Denic, S., Vuletic, D., Rao, R., Harris, F., & Amiri, K. (2009). FPGA implementation of a near-ML sphere detector for 802.16E broadband wireless systems. In *Proceedings of the 2009 SDR Technical Conference and Product Exposition, SDR '09, Washington*. Washington.
7. de Jong, Y., & Willink, T. (2005). Iterative tree search detection for MIMO wireless systems. *IEEE Transactions on Communications, 53*(6), 930–935.
8. Zimmermann, E. (2007) *Complexity aspects in near-capacity MIMO detection-decoding*. PhD thesis, TU, Dresden.
9. Zimmermann, E., Milliner, D., Barry, J., & Fettweis, G. (2008). Optimal LLR clipping levels for mixed hard/soft output detection. In *IEEE GLOBECOM 2008, New Orleans*. New Orleans.
10. Myllyl, M. (2011) *Detection algorithm and architectures for wireless spatial multiplexing in MIMO-OFDM systems*. PhD thesis, University of Oulu.
11. Wu, M., Dick, C., Sun, Y., & Cavallaro, J. R. (2011). Improving MIMO sphere detection through antenna detection order scheduling. In *Proceedings of the 2011 SDR Technical Conference and Product Exposition, SDR '11, Washington*. Washington.
12. Hochwald, B., & Brink, S. (2003). Achieving near-capacity on a multiple-antenna channel. *IEEE Transactions on Communications, 51*, 389–399.



Michael Wu received his B.S. degree from Franklin W. Olin College of Engineering in May of 2007 and his M.S. degree from Rice University in May of 2010, both in Electrical and Computer Engineering. He is currently a Ph.D. candidate in the E.C.E. department at Rice University. His research interests are wireless algorithms, software defined radio on GPGPU and other parallel architectures, and high performance wireless receiver designs.

Chris Dick is the DSP Chief Architect at Xilinx and the engineering manager for the Xilinx Communications Signal Processing Group (CSPG) in the Communications Business Unit (CBU). Chris has worked with signal processing technology for two decades and his work has spanned the commercial, military and academic sectors. Prior to joining Xilinx in 1997 he was a professor at La Trobe University, Melbourne Australia for 13 years and managed a DSP Consultancy called Signal Processing Solutions. He has been an invited speaker at many international signal processing symposiums and workshops and has authored more than 100 journal and conference publications, including many papers in the fields of parallel computing, inverse synthetic aperture radar (ISAR), FPGA implementation of wireless communication system PHYs and the use of FPGA custom computing. Chris' work and research interests are in the areas of fast algorithms for signal processing, digital communication, MIMO, OFDM, 3G LTE MODEM design, software defined radios, VLSI architectures for DSP, adaptive signal processing, synchronization, hardware architectures for real-time signal processing, and the use of Field Programmable Arrays (FPGAs) for custom computing machines and real-time signal processing. He holds a bachelor's and Ph.D. degrees in the areas of computer science and electronic engineering.

Yang Sun received the B.S. degree in Testing Technology & Instrumentation in 2000, and the M.S. degree in Instrument Science & Technology in 2003, both from Zhejiang University, Hangzhou, China. From 2003 to 2004, he worked at S3 Graphics Co. Ltd. as an ASIC design engineer, developing 3D Graphics Processors (GPU) for computers. From 2004 to 2005, he worked at Conexant Systems Inc. as an ASIC design engineer, developing Video Decoders for digital satellite-television set-top boxes (STBs). He received the Ph.D. degree in the Department of Electrical and Computer Engineering at Rice University, Houston, TX, in 2010. His research interests include parallel algorithms and VLSI architectures for wireless communication systems, especially forward-error correction (FEC) systems. He received the 2008 IEEE SoC Conference Best Paper Award, the 2008 IEEE Workshop on Signal Processing Systems Best Paper Award (Bob Owens Memory Paper Award), and the 2009 ACM GLSVLSI Best Student Paper Award.

Joseph R. Cavallaro (S'78, M'82, SM'05) received the B.S. degree from the University of Pennsylvania, Philadelphia, PA, in 1981, the M.S. degree from Princeton University, Princeton, NJ, in 1982, and the Ph.D. degree from Cornell University, Ithaca, NY, in 1988, all in electrical engineering. From 1981 to 1983, he was with AT& T Bell Laboratories, Holmdel, NJ. In 1988, he joined the faculty of Rice University, Houston, TX, where he is currently a Professor of electrical and computer engineering. His research interests include computer arithmetic, VLSI design and microlithography, and DSP and VLSI architectures for applications in wireless communications. During the 1996–1997 academic year, he served at the National Science Foundation as Director of the Prototyping Tools and Methodology Program. He was a Nokia Foundation Fellow and a Visiting Professor at the University of Oulu, Finland in 2005 and continues his affiliation there as an Adjunct Professor. He is currently the Director of the Center for Multimedia Communication at Rice University. He is a Senior Member of the IEEE. He was Co-chair of the 2004 Signal Processing for Communications Symposium at the IEEE Global Communications Conference and General/Program Co-chair of the 2003, 2004, and 2011 IEEE International Conference on Application-Specific Systems, Architectures and Processors (ASAP), and Program Co-chair for the 2012 ACM/IEEE GLSVLSI.