A Multi-mode Sphere Detector Architecture for WLAN Applications

Ramin Shariat-Yazdi and Tad Kwasniewski

Department of Electronics, Carleton University
Ottawa, Ontario, Canada

ABSTRACT
Designing high performance multi-mode MIMO detectors is the key to successful implementation of multiple-input multiple-output (MIMO) communication systems. As an attempt to address this issue, this paper presents a VLSI architecture for implementation of depth-first sphere decoding algorithm that can operate in multiple operating modes. To demonstrate the effectiveness of the proposed architecture, a configurable sphere detector has been implemented that can support 2×2/3×3/4×4 BPSK/QPSK/16-QAM MIMO systems.

I. INTRODUCTION
Multiple-input multiple-output (MIMO) communication offers a significant increase in data throughput with high spectral efficiency. MIMO systems use multipath propagations to improve overall bit error ratio (BER) and data rate in the communication link [1]. The IEEE standards for WLANs (IEEE 802.11n) and WMANs (IEEE 802.16) are based on the combination of MIMO with orthogonal frequency division multiplexing (OFDM) modulation. In these standards, receivers are required to have the ability to support a number of transmitter/receiver antenna configurations, modulation schemes and coding rates.

In this paper, the problem of developing a flexible and configurable architecture capable of detecting spatially multiplexed MIMO signals is addressed. We propose a dynamically configurable MIMO detector architecture based on depth-first sphere decoding algorithm and stack algorithm for searching tree graphs [3]. The proposed architecture has the following features:

- Number of transmitter and receiver antennas can be set to 2×2, 3×3 and 4×4.
- It supports BPSK, QPSK and 16-QAM modulation with the ability to have different modulation schemes on different transmitting antennas.

The rest of the paper is organized as follows. After a brief review of MIMO detection using sphere decoding algorithm, a configurable architecture for MIMO detection is introduced in section III. In section IV, the implementation results for a CMOS 0.18 μm target technology are presented. Finally, the conclusions are drawn in section V.

II. MIMO CHANNEL MODEL

In a flat-fading narrow band MIMO system comprising $N_t$ transmitter and $N_r$ receiver antennas ($N_t \geq N_r$), the channel model can be described as:

$$y = Hs + n$$

(1)

Where $y$ is the $N_r \times 1$ vector of received symbols, $H = [h_{ij}]$ is the $N_r \times N_t$ equivalent flat-fading channel matrix, $s = [s_1, ..., s_{N_t}]^T$ is an $N_t \times 1$ vector of transmitted symbols. The $N_r \times 1$ vector $n$ represents the thermal noise as independent identically distributed additive complex gaussian noise at the receiver. The transmit vector $s$ corresponds to a binary vector, containing $N_t b$ bits, where $b$ is the number of bits per symbol in the complex constellation $\Lambda = \{1, \ldots, 2^{b-1}\}$. The Maximum Likelihood (ML) detector in hard-output detectors solves (1) by finding the solution to (2).

$$s_{ML} = \arg \min_{s \in \Lambda} ||y - Hs||$$

(2)

By applying QR decomposition on channel matrix $H$, we can further simplify (1) and (2) to obtain (3) [5], [6]:

$$s_{ML} = \arg \min_{s \in \Lambda} ||y - Rs||$$

(3)

such that

$$H = QR$$

(4)
and
\[ y_q = Q^H y \]  
(5)
expanding vector norm in (3) yields
\[ s_{\text{sc}} = \arg \min \sum_{i=1}^{N_t} \left| y_{\omega} - \sum_{j=1}^{N_r} R_{ij} s_j \right|^2 \]  
(6)
starting from \( i=N_t \), (6) can be solved recursively as follows:
\[ T_i(P_i) = T_{i+1}(P_{i+1}) + e_i(P_i) \]  
(7)
\[ T_{N_t+1}(P_{N_t+1}) = 0 \]
where \( T_i(P_i) > T_{i+1}(P_{i+1}) \)
(8)
and
\[ e_i(P_i) = y_{\omega i} - \sum_{j=1}^{N_r} R_{ij} s_j \]  
(9)
In (7)-(9), \( P_i = [s_i, s_{i+1}, \ldots, s_{N_t}]^T \) is known as partial symbol vector [6]. Using the above simplification, the original optimization problem (2) can be solved recursively. The cost function of the resulting optimization problem is \( T_i(P_i) \) which is known as "Partial Euclidean Distance (PED)" [6].

III. DETECTOR ARCHITECTURE

A. Depth-First Sphere Detector

In sphere decoding algorithm the search for ML solution is limited to only those vector symbols \( s \) for which \( Hs \) lies inside a hyperspace with radius \( r \) around the received point \( y \) as described in (10) [2], [7].
\[ d(s) = \|y - Hs\| < r \]  
(10)
All possible combinations of transmitted symbols in a MIMO channel can be mapped on to a tree graph such that the root of the tree is on level \( N_t+1 \) and each node at level \( i \) corresponds to the transmitted symbols from \( N_t \) to \( i \)-th antenna. In a depth-first sphere detector, the tree is traversed such that at each node after calculating the partial metric \( T_i \) the sphere radius test is performed. The result of this test determines the direction of tree traverse. It is possible to improve the throughput of the sphere detector by shrinking the sphere radius dynamically [5]. Following Schnorr-Euchner (SE) ordering rule further reduces the complexity of the tree search [4].

B. Search Algorithm

The flow chart of the configurable sphere detector algorithm is shown in Fig.1, where \( T_i(c_m) \) and \( P_i(c_m) \) are defined as:
\[ P_i(c_m) = [c_m, s_{i+1}, \ldots, s_N] \]  
(11)
\[ T_i(c_m) = T_i(P_i(c_m)) \]  
(12)
The algorithm has two main branches; the node processing and the buffer control. In an iteration of the algorithm, the processing branch receives the relevant information of a parent node from data buffers and expands it to all of its children nodes. This information includes the accumulated metric \( T_{i+1} \) and the partial symbol vector of the node from root of the tree \( P_{i+1} \). The new PED metrics are calculated for each of the children nodes of the selected parent node and the results are compared to the sphere radius \( r \). Vector \( V \) shows the result of this comparison for each of the children nodes. The mask vector \( m \) defines the validity of each of the children nodes based on the type of modulation scheme selected for that tree level. Vector \( V \) is the final validity vector of children nodes and is stored in buffers along with updated metrics and partial symbol vectors. The buffer control branch of flow chart determines the search direction. In the forward traverse mode, the search continues with symbols that pass the sphere radius test until the search reaches the first level of search tree (\( i = 1 \)). After each sphere radius test, the minimum value of all the metrics that pass the test are found and are directly sent to the node processing branch. The rest of metrics along with their associated partial symbols (\( P_i \)) are stored in the buffers assigned to level \( i \) (buf). In hardware implementation, this saves one clock cycle and improves the overall throughput of the detector. If no metrics pass the sphere radius test or the algorithm reaches the first level of the tree (\( i = 1 \)) a pop operation occurs and a parent node from previous level of the tree is used to continue forward traverse of the tree. The search stops when all the buffers (buf, \( m = 2, \ldots, N_t \)) are empty. Fig. 2 shows an example of forward and backward movement in search tree.

C. Proposed Flexible Architecture

The top level architecture for the detector has been shown in Fig. 3, depicting the main blocks.
In every clock cycle, one of the parent nodes stored in level $i$ of the Data Buffer is expanded to all its children nodes; this includes calculating the PED metrics for all children, comparing the metrics with the sphere radius, selecting the nodes that pass the sphere radius test and applying the modulation mask to filter out the unwanted nodes. All partial symbol vectors and their associated metrics ($T_i$) are stored in the Data Buffer block. The Metric Computation block computes the PEDs of all children of a parent node in parallel considering 16-QAM constellation points. The role of the parallel comparators in this architecture is to compare the values of the updated metrics with the sphere radius. The resulting vector $V_i$ shows the list of valid children nodes. Using a series of parallel AND gates and a multiplexer we can implement a configurable architecture to support different modulation schemes. The constellation points in QPSK modulation can be considered as a subset of 16-QAM modulation. In case of BPSK or QPSK modulation the mask value selected by the multiplexer, sets the valid bits associated with the unwanted nodes to zero. The resulting vector $V_i$ shows the validity of the final metrics and is stored in Data Buffer. The type of modulation for each of the transmitting antennas is selected by $m_i$ inputs. At each level of depth-first tree search control unit selects the appropriate mask value to reflect the modulation scheme for that tree level. In Data Buffer block consists of 3 buffers assigned to levels 4, 3, and 2 of the search tree (Fig. 4).

It can be configured to support 3 or 2 antennas by disabling $buf_2$ and $buf_3$ from Buffer Control block. The Buffer Control selects the buffer module based on the tree level and number of antennas. It also controls the write (push) and read (pop) operations in each buffer modules. The write operation to all the locations within each module is simultaneous. Each module selects the next entry that should be read based on the SE ordering rule. In Fig. 5 the internal architecture of the data buffer modules has been depicted. Each module stores the information related to nodes including metric, partial symbol vector, and valid bit. A set of parallel comparators continuously compare the metrics for each of the valid nodes with the updated sphere radius and set the corresponding valid bits to zero. The metrics that fail to satisfy sphere condition are eliminated internally before entering main processing pipeline.

IV. IMPLEMENTATION RESULTS

Based on the proposed architecture, a $2\times2/3\times3/4\times4$ (BPSK/QPSK/16-QAM) MIMO detector has been designed, synthesized and mapped to a 0.18 µm CMOS standard cell library.
After synthesis and mapping, a clock frequency of \( f_{\text{clk}} = 65 \text{ MHz} \) has been achieved with a logic complexity of 87K equivalent gates (GE). Equivalent gates number is calculated by dividing total area by the area of a two-input drive-1 NAND gate. All the arithmetic computations are based on 16-bit fixed point numbers. Data buffers have been implemented using registers to increase the speed at the expense of silicon area. The average throughput of the detector as a function of signal to noise ratio (SNR) for 4×4 16-QAM mode is shown in Fig. 6. In Table I. the performance figures of the proposed architecture have been compared with the previously reported results in the literature.

V. CONCLUSION

Configurability in terms of number of supported antennas and modulation schemes is one of the key features of the future wireless standards. In this paper a multi-mode VLSI architecture for sphere decoding algorithm has been proposed. The proposed architecture is configurable for different modulation schemes (BPSK, QPSK, and 16-QAM) and antenna configurations (2×2, 3×3, and 4×4).

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Antennas</td>
<td>4×4</td>
<td>4×4</td>
<td>4×4, 3×3, 2×2</td>
</tr>
<tr>
<td>Modulation</td>
<td>16-QAM</td>
<td>16-QAM</td>
<td>16-QAM, QPSK, BPSK</td>
</tr>
<tr>
<td>Detector</td>
<td>Sphere detector</td>
<td>K-best</td>
<td>Sphere detector</td>
</tr>
<tr>
<td>Tech [( \mu \text{m} )]</td>
<td>0.25 ( \mu \text{m} )</td>
<td>0.35 ( \mu \text{m} )</td>
<td>0.18 ( \mu \text{m} )</td>
</tr>
<tr>
<td>Equivalent gate number</td>
<td>117 K</td>
<td>91 K</td>
<td>87 K</td>
</tr>
<tr>
<td>Clock Frequency</td>
<td>51 MHz</td>
<td>100 MHz</td>
<td>65 MHz</td>
</tr>
<tr>
<td>Throughput @ 20 dB</td>
<td>73 Mbps</td>
<td>53 Mbps</td>
<td>112 Mbps (4×4 16-QAM)</td>
</tr>
</tbody>
</table>

REFERENCES