Microprocessors and Microsystems 36 (2012) 245–256

Contents lists available at SciVerse ScienceDirect

Microprocessors and Microsystems

journal homepage: www.elsevier.com/locate/micpro

An application specific instruction set processor based implementation for signal detection in multiple antenna systems

M. Tamagnone, M. Martina, G. Masera *

Dipartimento di Elettronica, Politecnico di Torino, 10129 Torino, Italy

ARTICLE INFO

Article history:
Available online 25 November 2011

Keywords:
MIMO systems
ASIP
Sphere decoder
VLSI

ABSTRACT

In comparison to single antenna systems, a wireless multiple-input multiple-output (MIMO) system provides higher throughput at no additional cost of bandwidth, but the high complexity of the detection algorithms poses a major challenge to the hardware implementation. Maximum likelihood (ML) MIMO detection guarantees optimal performance but implies huge processing complexity, which makes acceptable this approach only when the number of transmitting antennas is low and the adopted modulation scheme has a small cardinality. Sphere decoding (SD) is an efficient method that significantly reduces the average processing complexity with no performance penalty.

Most of known sphere decoders have been implemented as application specific integrated circuits (ASICs), but the need for high degree of flexibility in MIMO detection makes interesting also application specific instruction set processor (ASIP) implementations. A single programmable ASIP can hardly reach the same processing speed as a fully dedicated ASIC; thus, parallel architectures with multiple concurrent ASIPs must be conceived to guarantee sufficient data throughput.

The objective of this paper is to present a new ASIP-based implementation for the detection of MIMO signals. The processor supports multiple lattice modulation schemes (up to 64-QAM) and up to four transmitting antennas and it is able to run both ML and close to ML algorithms. A parallel architecture has been also designed with multiple ASIPs, which concurrently execute the detection algorithm on received symbols, a central unit acting as task scheduler, and a buffer for the compensation of non constant throughput. A dedicated bus handles the communication among allocated units. Each ASIP occupies a silicon area of 0.093 mm^2 and runs at 400 MHz when implemented on a 90 nm CMOS technology. Achievable throughput depends on the adopted MIMO system and on the number of allocated ASIPs: a detector with 10 ASIPs programmed to run ML detection on a 4×4 MIMO system with 64-QAM modulation offers a throughput of 78 Mbps at signal-to-noise ratio SNR = 18 dB.

1. Introduction

Multiple-input multiple-output (MIMO) systems [1] are digital transmission systems with more than one transmitting and receiving antenna. Such systems, in combination with space-time codes, offer both multiplexing capabilities and transmit diversity, which can be exploited to achieve increased channel capacity and robustness against multipath fading channels. On the receiver side each antenna receives a linear combination of transmitted symbols. Provided that the receiver is able to reconstruct the originally sent information from received signals, such systems have the potential to multiply the achievable data rate by the number of transmitting antennas at no cost of additional bandwidth. MIMO systems are also known to be able to increase the robustness to combat the fading in wireless channels.

The drawback of MIMO systems is the computational complexity required to perform the detection of the originally transmitted symbols, which is the task implemented by MIMO detectors. In the direct implementation of maximum likelihood (ML) detection, the computational complexity grows exponentially with the order of the system, which depends on the size of the selected constellation and on the number of antennas. Therefore, computationally efficient algorithms are required to achieve acceptable bit rates in the detection process, especially for high order MIMO systems. Sphere decoding (SD) techniques [2] are a class of efficient detection algorithms that solve the ML problem with polynomial average complexity in the rate [3]. SD techniques are based on the association of all the possible transmitted vectors to the leaves of a decision tree, but each SD algorithm is different in terms of tree structure and exploring procedure. Since all modern high

* Corresponding author. Tel.: +39 011 5644102; fax: +39 011 5644099.
E-mail address: guido.masera@polito.it (G. Masera).

0141-9331/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.micpro.2011.11.003

performance receivers include iterative soft-input soft-output channel decoders, soft-output detection on MIMO channels is also required [4]. Thus, several SD algorithms have been extended to provide soft-output information instead of hard bits.

This paper mainly deals with depth-first algorithms, both hard-output (HO) and soft-output (SO), as the ones presented in [5,6] respectively. The aim of the paper is to present a new implementation of these algorithms based on an array of application specific instruction set processors (ASIPs) [7]. Each ASIP is optimized to perform the SD on a single vector symbol, and the parallelization is required to achieve acceptable throughput level.

The proposed system also includes a central processor (CP) that works as scheduler, assigning the received symbols to the allocated ASIPs and retrieving the results from them. The scheduling policies employed in the CP are crucial to find practical solutions to the main problem of the SD: its non-constant throughput. This paper covers the ASIP structure and the proposed scheduling policies as well. Notice that the CP has been implemented only on a behavioral level, while the ASIP has been fully synthesized for both FPGA and CMOS standard cell technologies.

The rest of the paper is organized as follows. The MIMO model and the sphere decoding algorithms are introduced in Section 2 together with the notation used in the paper. Section 3 introduces the designed ASIP architecture, while the structure of the multi-ASIP system and the adopted scheduling policies are detailed in Section 4. Simulation results are given in Section 5, where the effects of different architectural choices are evaluated in terms of throughput and BER (bit error rate) performance. Section 6 provides ASIP synthesis results as well as comparisons with other SD implementations. Finally, conclusions are drawn in Section 7.

2. Depth first sphere decoding

This section briefly introduces HO and SO algorithms that employ the depth-first sphere decoding. A complete derivation can be found in [5,6].

2.1. MIMO link model

Let us consider a MIMO channel with M_T transmitting antennas and M_R receiving antennas. The same carrier signal is used for every transmitting antenna, each of which transmits synchronously a different digital symbol belonging to the constellation $\mathcal{O}$. Let $\mathbf{s}$ be the vector of the complex transmitted symbols (one entry for each transmitting antenna) and $\mathbf{y}$ the vector of the complex received signals (one entry for each receiving antenna). Then, the model of the channel is given by:

$$\mathbf{y} = \mathbf{H}\mathbf{s} + \mathbf{n} \text{ with } \mathbf{s} \in \mathcal{O}^{M_T} \tag{1}$$

where $\mathbf{H}$ is the channel matrix ($M_R \times M_T$) and $\mathbf{n}$ is a vector of independent complex gaussian random variables with variance $N_0/2$, which models the noise at the receiver. The purpose of sphere decoding is to find the most likely vector symbol $\hat{\mathbf{s}}$, i.e. the one that minimizes the Euclidean distance from $\mathbf{y}$:

$$\hat{\mathbf{s}} = \arg\min_{\mathbf{s}\in\mathcal{O}^{M_T}} d(\mathbf{s}) = \arg\min_{\mathbf{s}\in\mathcal{O}^{M_T}} \|\mathbf{y} - \mathbf{H}\mathbf{s}\|^2 \tag{2}$$

2.2. Preprocessing and decision trees

A required preliminary step in sphere decoding is QR decomposition, which is a method to triangularize the $M_R \times M_T$ matrix $\mathbf{H}$ [8]:

$$\mathbf{H} = \mathbf{Q}\mathbf{R} \tag{3}$$

where $\mathbf{R}$ is a $M_T \times M_T$ upper triangular matrix, while $\mathbf{Q}$ is a $M_R \times M_T$ matrix with orthonormal columns.

Using this decomposition, a squared euclidean distance (ED) $d(\mathbf{s})$ can be associated to each symbol vector $\mathbf{s}$:

$$d(\mathbf{s}) = c + \|\hat{\mathbf{y}} - \mathbf{R}\mathbf{s}\|^2 \tag{4}$$

where $\hat{\mathbf{y}} = \mathbf{Q}^H\mathbf{y} = \mathbf{R}\mathbf{s}^{ZF}$, the superscript $(\cdot)^H$ stands for the Hermitian transpose, $\mathbf{s}^{ZF}$ is the zero-forcing solution [5] and $c = \|\mathbf{y}\|^2 - \|\hat{\mathbf{y}}\|^2$ is a constant with respect to the symbol $\mathbf{s}$. The problem described by (2) is then equivalently formulated as the problem of finding the $\mathbf{s}$ symbol with the minimum ED. It is worth noticing that constant c does not influence the metric computation expressed in (2) and will then be omitted from now on.

We introduce now the partial symbol vector $\mathbf{s}^{(i)}$, defined as:

$$\mathbf{s}^{(i)} \triangleq [s_i, s_{i+1}, \ldots, s_{M_T}]' \tag{5}$$

where $[\cdot]'$ indicate a transposed vector. These partial symbols can be organized in a tree [5], where $i = M_T$ is the level of the tree root, $i = 1$ is the level of tree leaves, and each possible value of partial symbol $\mathbf{s}^{(i)}$ is associated to one node at level i. A node at level i inherits from its parent at the upper level $i-1$ the corresponding partial symbol $\mathbf{s}^{(i-1)}$ and obtains $\mathbf{s}^{(i)}$ by appending one more vector element, corresponding to a specific choice for s_i.

The leaves of the tree are all the possible $\mathbf{s} \in \mathcal{O}^{M_T}$. Therefore ED values $d(\mathbf{s})$ are associated to tree leaves, while intermediate tree nodes can be associated to partial euclidean distances (PED) $T_i(\mathbf{s}^{(i)})$. PEDs can be additively updated when moving from the root towards the leaves of the tree:

$$\begin{aligned} &T_{M_T+1}(\cdot) \triangleq 0 \\ &T_i(\mathbf{s}^{(i)}) \triangleq T_{i+1}(\mathbf{s}^{(i+1)}) + |e_i(\mathbf{s}^{(i)})|^2 \end{aligned} \tag{6}$$

with $i = M_T, M_T-1, \ldots, 1$; amounts $|e_i(\mathbf{s}^{(i)})|^2$ are called *distance increments*:

$$|e_i(\mathbf{s}^{(i)})|^2 = \left|\hat{y}_i - \sum_{j=i}^{M_T} R_{ij}s_j\right|^2 = |\psi_{i+1}(\mathbf{s}^{(i+1)}) - R_{ii}s_i|^2 \tag{7}$$

where

$$\psi_{i+1}(\mathbf{s}^{(i+1)}) = \hat{y}_i - \sum_{j=i+1}^{M_T} R_{ij}s_j \tag{8}$$

From (6) it is evident that the PED of a node is larger than the one of its parent node, and this property is exploited to achieve an efficient tree exploration. For example, to explore only leaves such that $d(\mathbf{s}) < r$, with a certain r (called *sphere constraint*), we can prune the tree under an intermediate node whose PED violates the constraint, so reducing the number of searched nodes, with no penalty in terms of quality of the found solution.

2.3. Hard output SD

By exploring the tree in a depth first way, we can determine the leaf with the smaller ED, which corresponds to the ML solution. The result is given by the bits associated to the found leaf, hence this algorithm provides output sequences of zeros and ones (hard bits) without any information on the reliability of the detected bits.

During tree exploration, when a leaf is reached with ED value smaller than previously calculated, the constraint on the sphere radius is updated, so as the search space is reduced to all tree nodes with PED lower that the current radius. This allows pruning the tree without the risk of eliminating the ML solution. In exploring the sons of a node, the Schnorr–Euchnerr (SE) enumeration [9] can be used: in this approach, the sons are explored in ascending order of their PEDs. In the SE enumeration, the first explored node