FPGA in Wireless Communications Applications

Kiarash Amiri[1], Melissa Duarte[1], Joseph R. Cavallaro[1], Chris Dick[2], Raghu Rao[2], and Ashutosh Sabharwal[1]

[1] Rice University {kiaa,mduarte,cavallar,ashu}@rice.edu
[2] Xilinx Inc. {chris.dick,raghu.rao}@xilinx.com

1 Introduction

In the past decade we have witnessed explosive growth in the wireless communications industry with over 4 billion subscribers worldwide. While first and second generation systems focussed on voice communications, third generation networks (3GPP and 3GPP2) embraced CDMA (code division multiple access) and had a strong focus on enabling wireless data services. As we reflect on the rollout of 3G services, the reality is that first generation 3G systems did not entirely fulfill the promise of high-speed transmission, and the rates supported in practice were much lower than that claimed in the standards. Enhanced 3G systems were subsequently deployed to address the deficiencies. However, the data rate capabilities and network architecture of these systems are insufficient to address the insatiable consumer and business sector demand for the nomadic delivery of media- and data-centric services to an increasingly rich set of mobile platforms.

With these considerations in hand, the move to 4G technologies like 3G LTE (long term evolution) [1] and WiMAX [2] is proceeding at an extremely rapid rate. The goal of next generation systems is to provide high-data rate, low-latency, high reliability (minimize outages and connection drops) employing packet-optimized radio access technology supporting flexible bandwidth allocation. Additional key objectives are to drive down the cost of infrastructure equipment and consumer terminals and to employ a more efficient modulation scheme than the CDMA technology used in 3G systems in order to make more optimal use of precious communication bandwidth.

To meet all of these requirements a significant re-structuring of both the physical layer (PHY) and network architecture is required. Increased spectral efficiency is delivered in all 4G cellular broadband sysems through the use of OFDM (orthogonal frequency division multiplexing) as the preferred modulation scheme. The technology center-piece for delivering improved data rates and communication link robustness is more aggressive use of the spatial dimension of a communication system through multiple-input multiple-output

(MIMO) [3] techniques. 4G wireless systems will employ MIMO processing to equip next generation systems with spatial multiplexing (improved data rates) and diversity (improved reliability) capabilities. MIMO systems (Fig 1) can increase the data rate and provide multiplexing gain by transmitting different data on different antennas [4], also known as spatial multiplexing. MIMO systems can also improve the reliability and error performance in the receiver through diversity gain, i.e. providing the receiver with multiple copies of the transmitted signal. One practical way to realize such diversity gains is to beamform a message across multiple transmit antennas [2, 5]. The beamforming procedure utilizes limited information about the channel state information; this channel information is usually provided through a limited feedback link from the receiver to the transmitter.

The implementation of 4G infrastructure equipment has several challenges. The first is related to the compute requirements of MIMO-OFDM systems. The more sophisticated MIMO-OFDM advanced receiver architectures have compute requirements that are several orders of magnitude greater than that of 3G CDMA systems. A second requirement is related to bridging the gap between *legacy* standards (e.g. 3GPP) and next generation 4G protocols and systems. In many cases 4G infrastructure equipment will need to be multi-mode in nature and support not only the MIMO-OFDM PHY of WiMAX and 3G LTE systems, but also the W-CDMA (wideband CDMA) PHY of 3GPP-based networks. To first order, the silicon technologies deployed in next generation systems will not only need to supply enormous compute capability (many hundreds of giga-operations/second), but they will also require significant flexibility in order to support multi-mode basestation infrastructure equipment.

Field programmable gate arrays (FPGAs) with their inherently parallel structure, are increasingly the technology of choice for addressing the compute and flexibility requirements of next generation systems. One of the fundamental areas in many academic and industrial research organizations is in the area of hardware realization of advanced MIMO receivers.

Fig. 1. Multiple antenna (MIMO) system.

In this chapter, we discuss the architectural challenges associated with spatial multiplexing and diversity gain schemes and introduce FPGA architectures and report experimental results for the FPGA realization of these systems. We introduce a flexible architecture and implementation of a spatial multiplexing MIMO detector, Flex-sphere, and its FPGA implementation. We also present a hardware architecture for beamforming in a WiMAX system as a way to enhance the diversity and performance in the next-generation wireless systems. Finally, we introduce the Rice University Wireless Open Access Research Platform (WARP), an FPGA-centric scalable and programmable research platform, and how it is used for studying the effects of MIMO systems in real-world scenarios.

2 Spatial Multiplexing MIMO Systems

One of the challenges of MIMO systems is detecting the transmitted vector, which often times, translates into finding a minimum distances, and requires considerable resources. For instance, most of the MIMO detection require large sizes of searching and sorting in hardware. In the next two sections, we present the spatial multiplexing MIMO systems as well as a reduced complexity MIMO detector, Flex-Sphere.

We assume a MIMO OFDM system with M_T transmit antennas and $M_R \geq M_T$ receive antennas. We denote the K subcarriers by the subscript $k = 1, ..., K$ throughout this chapter. The input-output model is captured by

$$\tilde{\mathbf{y}}_k = \tilde{\mathbf{H}}_k \tilde{\mathbf{s}}_k + \tilde{\mathbf{n}}_k \tag{1}$$

where $\tilde{\mathbf{H}}_k$ is the complex-valued $M_R \times M_T$ channel matrix, $\tilde{\mathbf{s}}_k = [\tilde{s}_{1,k}, \tilde{s}_{2,k}, ..., \tilde{s}_{M_T,k}]^T$ is the M_T-dimensional transmitted vector, where each $\tilde{s}_{j,k}, j = 1, ..., M_T$, is chosen from a complex-valued constellation Ω_j of the order $p_{j,k} = |\Omega_{j,k}|$, $\tilde{\mathbf{n}}_k$ is the circularly symmetric complex additive white Gaussian noise vector of size M_R and $\tilde{\mathbf{y}}_k = [\tilde{y}_{1,k}, \tilde{y}_{2,k}, ..., \tilde{y}_{M_R,k}]^T$ is the M_R-element received vector. Note that we do not restrict all the parallel M_T streams to use the same modulation order; rather, each stream, which corresponds to one of the antennas of one of the users, may be using either the $4, 16$ or 64-QAM modulation. Also, note that even though we focus on one transmitter with multiple antennas, the results in this section can be extended to multiple users such that the sum of the number of antennas of all of them equal M_T.

We assume that the complete channel information, i.e. the channel matrix coefficients for each subcarrier, is known in the receiver. Moreover, since the detection procedure for each subcarrier is performed independent of other subcarriers, we will drop the k subscript unless it is needed. Therefore, the original MIMO model can be simplified to $\tilde{\mathbf{y}} = \tilde{\mathbf{H}}\tilde{\mathbf{s}} + \tilde{\mathbf{n}}$.

The preceding MIMO equation can be further decomposed into real-valued numbers as follows [5]:

$$\mathbf{y} = \mathbf{H}\mathbf{s} + \mathbf{n} \tag{2}$$

corresponding to

$$\begin{pmatrix} \Re(\tilde{\mathbf{y}}) \\ \Im(\tilde{\mathbf{y}}) \end{pmatrix} = \begin{pmatrix} \Re(\tilde{\mathbf{H}}) & -\Im(\tilde{\mathbf{H}}) \\ \Im(\tilde{\mathbf{H}}) & \Re(\tilde{\mathbf{H}}) \end{pmatrix} \begin{pmatrix} \Re(\tilde{\mathbf{s}}) \\ \Im(\tilde{\mathbf{s}}) \end{pmatrix} + \begin{pmatrix} \Re(\tilde{\mathbf{n}}) \\ \Im(\tilde{\mathbf{n}}) \end{pmatrix} \tag{3}$$

with $M = 2M_T$ and $N = 2M_R$ presenting the dimensions of the new model.

We call the ordering in (2), the conventional ordering. Using the conventional ordering, all the computations can be performed in real values, which would simplify the implementation complexity. Note that after real-valued decomposition, each $s_j, j = 1, ..., M$, in $\mathbf{s}$ is chosen from a set of real numbers, Ω'_j, with $p'_j = \sqrt{p_j}$ elements. For instance, for a 64-QAM modulation, each s_i can take any of the values in the set $\Omega' = \{\pm 7, \pm 5, \pm 3, \pm 1\}$.

The general optimum detector for such a system is the maximum-likelihood (ML) detector which minimizes $\| \mathbf{y} - \mathbf{H}\mathbf{s} \|^2$ over all the possible combinations of the $\mathbf{s}$ vector. Notice that for high order modulations and large number of antennas, this detection scheme incurs an exhaustive exponentially growing search among all the candidates, and is not practically feasible in a MIMO receiver. However, it has been shown that using the QR decomposition of the channel matrix, the distance norm can be simplified [6, 7, 8] as follows:

$$\begin{aligned} D(\mathbf{s}) &= \| \mathbf{y} - \mathbf{H}\mathbf{s} \|^2 \\ &= \| \mathbf{Q}^{\mathbf{H}}\mathbf{y} - \mathbf{R}\mathbf{s} \|^2 = \sum_{i=M}^{1} |y_i{}' - \sum_{j=i}^{M} R_{i,j} s_j|^2 \end{aligned} \tag{4}$$

where $\mathbf{H} = \mathbf{Q}\mathbf{R}$, $\mathbf{Q}\mathbf{Q}^{\mathbf{H}} = \mathbf{I}$ and $\mathbf{y}' = \mathbf{Q}^{\mathbf{H}}\mathbf{y}$. Note that the transition in (4) is possible through the fact that $\mathbf{R}$ is an upper triangular matrix.

The norm in (4) can be computed in M iterations starting with $i = M$. When $i = M$, i.e. the first iteration, the initial partial norm is set to zero, $T_{M+1}(\mathbf{s}^{(M+1)}) = 0$. Using the notation of [9], at each iteration the Partial Euclidean Distances (PEDs) at the next levels are given by

$$T_i(\mathbf{s}^{(i)}) = T_{i+1}(\mathbf{s}^{(i+1)}) + |e_i(\mathbf{s}^{(i)})|^2 \tag{5}$$

with $\mathbf{s}^{(i)} = [s_i, s_{i+1}, ..., s_M]^T$, and $i = M, M-1, ..., 1$, where

$$|e_i(\mathbf{s}^{(i)})|^2 = |y_i{}' - R_{i,i}s_i - \sum_{j=i+1}^{M} R_{i,j}s_j|^2 \tag{6}$$

$$= |J_{i+1}(\mathbf{s}^{(i+1)}) - R_{i,i}s_i|^2. \tag{7}$$

Here, the $T_i(\mathbf{s}^{(i)})$ represents the Partial Euclidean Distances (PEDs) of the children of a parent node; where lower values of PEDs represent a smaller distance, thus a higher reliability, of the partial candidate corresponding with

that node. The PED is comprised of $T_{i+1}(\mathbf{s}^{(i+1)})$, which is the PED of the parent of the current node, and $|e_i(\mathbf{s}^{(i)})|$, which represents the contribution of the current node to its PED. Moreover, it is worth noting that in computing the PED, the $J_{i+1}(\mathbf{s}^{(i+1)})$ can be computed once using the information of the parent node, and the remaining term depends on each separate node.

One can envision this iterative algorithm as a tree traversal with each level of the tree corresponding to one i value, and each node having p'_i children.

The tree traversal can be performed in either a breadth-first or a depth-first manner. In the depth-first tree search [10, 9, 11], only one node is expanded at any time and once the end of the tree is reached, the traversal continues by visiting the new nodes in the higher levels of the tree. Therefore, each level can be visited more than one time.

In the breadth-first tree search [12, 13], however, each level is only visited once, and more than one node per level is expanded. Once the end of the tree, or the leaves level, is reached, the minimum candidate is chosen. A typical breadth-first tree search is the K-best detector. In the $\mathcal{K}$-best detector, at each level, only the best $\mathcal{K}$ nodes, i.e. the $\mathcal{K}$ nodes with the smallest T_i, are chosen for expansion. Note that such a detector requires sorting a list of size $\mathcal{K} \times p'$ to find the best $\mathcal{K}$ candidates. For instance, for a 16-QAM system with $\mathcal{K} = 10$, this requires sorting a list of size $\mathcal{K} \times p' = 10 \times 4 = 40$ at most of the tree levels. This introduces a long delay for the next processing block in the detector unless a highly parallel sorter is used. Highly parallel sorters, on the other hand, consist of a large number of compare-select blocks, and result in dramatic area increase.

3 Flex-Sphere Detector

In order to simplify the sorting step, which significantly reduces the delay of the detector, a sort-free strategy can be utilized. Moreover, we discuss how using a new modified real-valued decomposition ordering (M-RVD) scheme can help in designing a flexible architecture. Finally, we discuss the design and implementation of the flexible sphere detector, Flex-Sphere [14, 15], that can support a range of modulation orders and number of antennas.

3.1 Tree Traversal for Flex-Sphere Detection

Using the sort-free technique, the long sorting operation is effectively simplified to a minimum-finding operation [14, 16]. The detailed steps of this algorithm are described in the Flex-Sphere Tree Traversal algorithm.

An example of this algorithm is illustrated in Figure 2 for a 4×4, 64-QAM system. Note that as described above, the first two levels are fully expanded to guarantee high performance; whereas for the following levels, only the best candidate in the children list of a parent node is expanded. In other words, after passing the first two levels, p_{M_T} nodes are expanded, and for each of

Algorithm 1 Flex-Sphere Tree Traversal

Input: $\mathbf{R}$, $\mathbf{y}'$
$T_{M+1}(\mathbf{s}^{(M+1)}) = 0$
$\mathcal{L} \leftarrow \emptyset$
$\mathcal{L}' \leftarrow \emptyset$
$i \leftarrow M$
\\ Full expansion of the first level:
- Compute T_i with (5),
- $\mathcal{L} \leftarrow \{(\mathbf{s}^{(i)}, T_i(\mathbf{s}^{(i)}))_j | j = 1, ..., p'\}$
- $i \leftarrow i - 1$
\\ Full expansion of the second level:
- **for** each $(\mathbf{s}^{(i+1)}, T_{i+1}(\mathbf{s}^{(i+1)})) \in \mathcal{L}$, **repeat**
  - compute $(\mathbf{s}^{(i)}, T_i(\mathbf{s}^{(i)}))_j$ children pairs, $j = 1, ..., p'$
  - $\mathcal{L}' \leftarrow \mathcal{L}' \cup \{(\mathbf{s}^{(i)}, T_i(\mathbf{s}^{(i)}))_j | j = 1, ..., p'\}$
- **end**
-$\mathcal{L} \leftarrow \mathcal{L}'$
-$\mathcal{L}' \leftarrow \emptyset$
\\ Minimum-based expansion of the next levels:
- **for** $i = M - 2$ down to $i = 1$, **repeat**
  - **for** each $(\mathbf{s}^{(i+1)}, T_{i+1}(\mathbf{s}^{(i+1)})) \in \mathcal{L}$, **repeat**
    - compute $(\mathbf{s}^{(i)}, T_i(\mathbf{s}^{(i)}))_j$ children pairs, $j = 1, ..., p'$
    - $(\mathbf{s}^{(i)}, T_i(\mathbf{s}^{(i)}))_{min} \leftarrow \arg \min_{\{(\mathbf{s}^{(i)}, T_i(\mathbf{s}^{(i)}))_j | j=1,...,p'\}} T_i(\mathbf{s}^{(i)})$
    - $\mathcal{L}' \leftarrow \mathcal{L}' \cup \{(\mathbf{s}^{(i)}, T_i(\mathbf{s}^{(i)}))_{min}\}$
  - **end**
  - $\mathcal{L} \leftarrow \mathcal{L}'$
  - $\mathcal{L}' \leftarrow \emptyset$
  - $i \leftarrow i - 1$
- **end**

- $(\mathbf{s}^{(i)}, T_i(\mathbf{s}^{(i)}))_{detected} \leftarrow \arg \min_{\mathcal{L}} \quad T_i(\mathbf{s}^{(i)})$

those p_{M_T} nodes, the best child node among its p'_M children nodes is selected as the survived node. Therefore, the new node list would contain p_{M_T} nodes in the third level. These p_{M_T} nodes are expanded in a similar way to the forth level, and this procedure continues until the very last level, where the minimum-distance node is taken as the detected node.

Moreover, from the Schnorr-Euchner (SE) ordering [17], we know that finding

$$(\mathbf{s}^{(i)}, T_i(\mathbf{s}^{(i)}))_{min} \leftarrow \arg \min_{\{(\mathbf{s}^{(i)}, T_i(\mathbf{s}^{(i)}))_j | j=1,...,p'\}} T_i(\mathbf{s}^{(i)})$$

basically corresponds to finding the real-valued constellation point closest to $\frac{1}{R_{ii}} J_{i+1}(\mathbf{s}^{(i+1)})$; see Eq. (7). Thus, the long sorting of K-best is avoided.

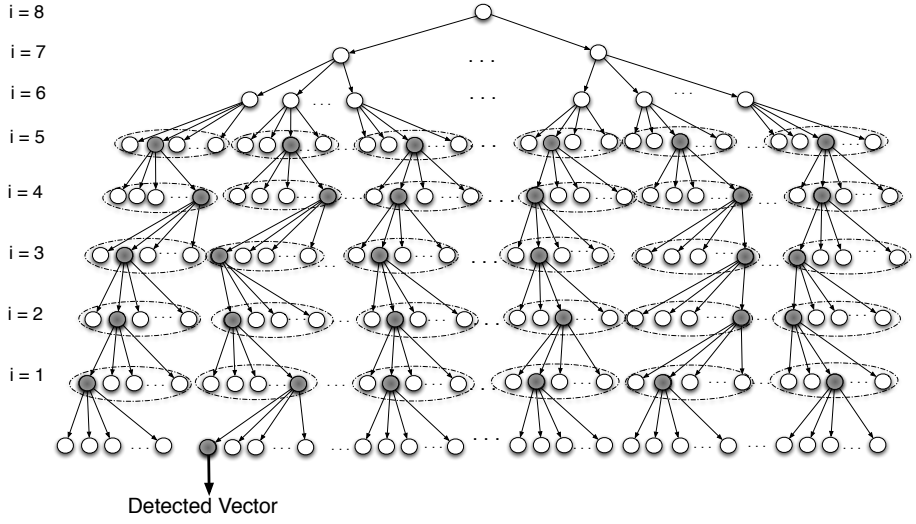

Fig. 2. Flex-Sphere algorithm for a 64-QAM, 4×4 system. The topmost two levels are fully expanded. The nodes marked with black are the minimum in their own set, where each set is denoted by dashed line. Note that because of the real-valued decomposition, each node has only $\sqrt{64} = 8$ children. Also, the number of tree levels are $M = 2 \times M_T = 8$.

3.2 Modified Real-Valued Decomposition (M-RVD) Ordering

For the sort free detector described in the preceding section, we can use a modified real-valued decomposition (M-RVD) ordering which improves the BER performance compared to the ordering given in Eq. (2). The new decomposition is summarized as [18]:

$$\hat{\mathbf{y}} = \hat{\mathbf{H}}\hat{\mathbf{s}} + \hat{\mathbf{n}} \tag{8}$$

or,

$$\begin{pmatrix} \Re(\tilde{y}_1) \\ \Im(\tilde{y}_1) \\ \Re(\tilde{y}_2) \\ \Im(\tilde{y}_2) \\ . \\ . \\ . \\ \Re(\tilde{y}_{M_R}) \\ \Im(\tilde{y}_{M_R}) \end{pmatrix} = \hat{\mathbf{H}} \begin{pmatrix} \Re(\tilde{s}_1) \\ \Im(\tilde{s}_1) \\ \Re(\tilde{s}_2) \\ \Im(\tilde{s}_2) \\ . \\ . \\ . \\ \Re(\tilde{s}_{M_T}) \\ \Im(\tilde{s}_{M_T}) \end{pmatrix} + \begin{pmatrix} \Re(\tilde{n}_1) \\ \Im(\tilde{n}_1) \\ \Re(\tilde{n}_2) \\ \Im(\tilde{n}_2) \\ . \\ . \\ . \\ \Re(\tilde{n}_{M_R}) \\ \Im(\tilde{n}_{M_R}) \end{pmatrix} \tag{9}$$

where $\hat{\mathbf{H}}$ is the permuted channel matrix of Eq. (3) whose columns are re-ordered to match the other vectors of the new decomposition ordering in Eq.

(8). It is worth noting that since the difference between RVD and M-RVD is the grouping of the signals, there is no extra computational cost associated with this modified ordering. We will see in the following sections how this ordering could reduce the latency for Flex-Sphere.

3.3 FPGA Design of the Configurable Detector for SDR Handsets

In this section, the main features of the architecture and the FPGA implementation of the SDR handset detector are presented. We use Xilinx System Generator [19] to implement the proposed architecture. In order to support all the different number of antenna/user and modulation orders, the detector is designed for the maximal case, i.e. $M_T \times M_R$, 64-QAM case, and configurability elements are introduced in the design to support different configurations.

PED Computations

Computing the norms in (7) is performed in the PED (Partial Euclidean Distance) blocks. Depending on the level of the tree, three different PED blocks are used: The PED in the first real-valued level, PED_1, corresponds to the root node in the tree, $i = M = 2M_T = 8$. The second level consists of $\sqrt{64} = 8$ parallel PED_2 blocks, which compute 8 PEDs for each of the 8 PEDs generated by PED_1; thus, generating 64 PEDs for the $i = 7$ level. Followed by this level, there 8 parallel general PED computation blocks, PED_g, which compute the closest-node PED for all 8 outputs of each of the PED_2s. The next levels will also use PED_g. At the end, the Min_Finder unit detects the signal by finding the minimum of the 64 distances of the appropriate level. The block diagram of this design is shown in Figure 3.

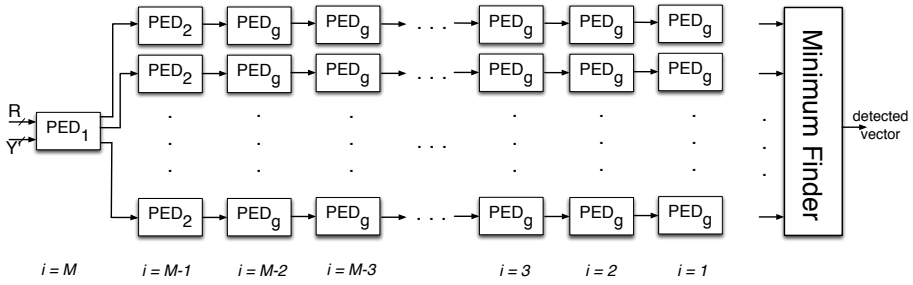

Fig. 3. The block diagram of the Flex-Sphere. Note that there are M parallel PEDs at each level. The inputs to the Min_Finder is fed from the appropriate PED block, as described in section 3.3.

Configurable Design

In order to ensure the configurability of the Flex-Sphere, it needs to support different M_T as well as different modulation orders for different users. The configurability of the detector is achieved through two input signals, M_T and $q^{(i)}$, which control the number of antennas and the modulation order, respectively. These two inputs can change based on the system parameters at any time during the detection procedure. Therefore, this configurability is a real-time operation.

Number of Antennas: The M_T determines the number of detection levels, and it is set through M_T input to the detector, which in turn, would configure the Min_Finder appropriately. Therefore, the minimum finder can operate on the outputs of the corresponding level, and generate the minimum result. In other words, the multiplexers in each input of the Min_Finder block, choose which one of the four streams of data should be fed into the Min_Finder. Therefore, the inputs to the Min_Finder would be coming from the $i = 5, 3$ or 1, if M_T is $2, 3$ or 4; respectively, see Figure 3.

The M_T input can change on-the-fly; thus, the design can shift from one mode to another mode based on the number of streams it is attempting to detect at anytime. Moreover, as will be shown later, the configurability of the minimum finder guarantees that less latency is required for detecting smaller number of streams.

Modulation Order: In order to support different modulation orders per data stream, the Flex-Sphere uses another input control signal $q^{(i)}$ to determine the maximum real value of the modulation order of the i-th level. Thus, $q^{(i)} \in \{1, 3, 7\}$. Moreover, since the modulation order of each level is changing, a simple comparison-thresholding can not be used to find the closest candidate for Schnorr-Euchner [17] ordering. Therefore, the following conversion is used to find the closest SE candidate:

$$\tilde{s} = g(2[\frac{J+1}{2}] - 1) \tag{10}$$

where $[.]$ represents rounding to the nearest integer, $J = (1/R_{ii}) \cdot J_{i+1}$ of Eq. (7), and $g(.)$ is

$$g(x) = \begin{cases} -q^{(i)} & x \leq -q^{(i)} \\ x & -q^{(i)} \leq x \leq q^{(i)} \\ q^{(i)} & x \geq q^{(i)} \end{cases} \tag{11}$$

All of these functions can be readily implemented using the available building blocks of the Xilinx System Generator, see Figure 4. Note that the multiplications/divisions are simple one-bit shifts.

For the first two levels, which corresponds to the in-phase and quadrature components of the last antenna, the PED of the out-of-range candidates are simply overwritten with the maximum value; thus, they will be automatically discarded during the minimum-finding procedure.

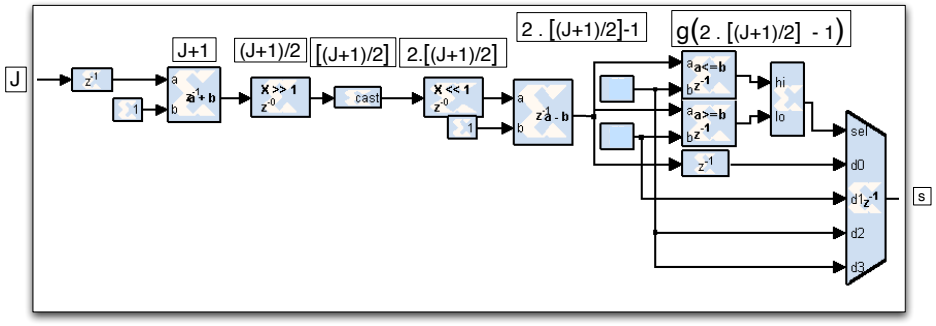

Fig. 4. The pipelined System Generator block diagram for Eq. (10) in the PED_g to support different modulation orders.

Modified Real Valued Decomposition (M-RVD)

Using the real-valued decomposition, the two extra adders that are required per each complex multiplication of the complex representation in Eq (1), can be avoided; thus, avoiding the unnecessary FPGA slices on the addition operations. Moreover, while using the complex-valued operations require the SE ordering of [9], which would be a demanding task given the configurable nature of the detector; with the real-valued decomposition, the SE ordering can be implemented more efficiently and simply for the proposed configurable architecture as described earlier. Also, note that even though some of the multiplications can be replaced with shift-adds in an area-optimized ASIC design; for an FPGA implementation, the appropriate design choice is to use the available embedded multipliers, commonly known as XtremeDSP and DSP48E in Virtex-4 and Virtex-5 devices.

It is noteworthy that if the conventional real-valued decomposition of (3) were employed; then, the results for a 2×2 system would have been ready only after going through all the in-phase tree levels and the first two quadrature levels. However, with the modified real-valued decomposition (M-RVD), every antenna is isolated from other antennas in two consecutive levels of the tree. Therefore, there is no need to go through the latency of the unnecessary levels. Thus, using the M-RVD technique, offers a latency reduction compared to the conventional real-valued decomposition.

Timing Analysis

Each of the PED_g blocks are responsible for expanding 8 nodes; thus, the folding factor of the design is $F = 8$. In order to ensure a high maximum clock frequency, several pipelining levels are introduced inside each of the PED computation blocks. The latency of the PED_1, PED_2 and PED_g blocks

are $7, 17$ and 22, respectively. Note that the larger latency of the PED_g blocks is due to more multiplications required to compute the PEDs of the later levels. The Min_Finder block has a latency of 8.

As mentioned earlier, different values of M_T require different number of tree levels, which incur different latencies. The latencies of the three different configurations of M_T are presented in Table 1. In computing the latencies, an initial 8 cycles are required to fill up the pipeline path.

Table 1. Latency for different values of M_T.

M_T	Latency
$M_T = 2$	$8 + \text{PED}_1 + \text{PED}_2 + 2 \cdot \text{PED}_g + \text{Min\_Finder} = 84$
$M_T = 3$	$8 + \text{PED}_1 + \text{PED}_2 + 4 \cdot \text{PED}_g + \text{Min\_Finder} = 128$
$M_T = 4$	$8 + \text{PED}_1 + \text{PED}_2 + 6 \cdot \text{PED}_g + \text{Min\_Finder} = 172$

Xilinx FPGA Implementation Results for $M_T = 3$

Table 2 presents the System Generator implementation results of the Flex-Sphere on a Xilinx Virtex-4 FPGA, xc4vfx100-10ff1517 [19] for 16-bits precision. The maximum number of detectable streams is set to $M_T = 3$. The maximum achievable clock frequency is 250 MHz. Since the design folding factor is set to $F = 8$, the maximum achievable data rate, i.e. $M_T = 3$ and $p_i = 64$, is

$$D = \frac{M_T \cdot \log w}{F} \cdot f_{max} = 562.5 \text{ [Mbps]}. \tag{12}$$

Table 2. FPGA resource utilization summary of the proposed Flex-Sphere for the Xilinx Virtex-4, xc4vfx100-10ff1517, device.

No. of Antennas	2, 3
Modulation Order	$\{4, 16, 64\}$-QAM
Max. Data Rate	562.5 Mbps
Number of Slices	18,825/42,176 (44%)
Number of Slice FFs	23,961/84,352 (28%)
Number of LUTs	30,297/84,352 (35%)
Number of DSP48E	129/160 (80%)
Max. Freq.	250 MHz

Xilinx FPGA Implementation Results for $M_T = 4$

The System Generator design for $M_T = 4$ antennas is shown in Figure 5. Table 3 presents the System Generator implementation results of the Flex-Sphere

on a Xilinx Virtex-5 FPGA, xc5vsx95t-3ff1136 [19] for 16-bits precision and $M_T = 4$. The maximum achievable clock frequency is 285.71 MHz. Since the design folding factor is set to $F = 8$, the maximum achievable data rate, i.e. $M_T = 4$ and $p_i = 64$, is

$$D = \frac{M_T \cdot \log w}{F} \cdot f_{max} = 857.1 \text{ [Mbps]}. \tag{13}$$

The Flex-Sphere can support different number of antennas and modulation orders, and achieves high data rate requirements of various wireless standards. Table 4 summarizes the data rates for all of the different scenarios of the $M_T = 4$, Virtex-5, implementation.

Table 3. FPGA resource utilization of the proposed Flex-Sphere.

Device	XC5VSX95
No. of Antennas	2, 3, 4
Modulation Order	$\{4, 16, 64\}$-QAM
Max. Data Rate	857.1 Mbps
BER = 10^{-4} @ SNR =	= 25 dB
Number of Slices	11,604/14,720 (78 %)
Number of Registers/FFs	27,115/58,880 (46 %)
Number of Slice LUTs	33,427/58,880 (56 %)
Number of DSP48E/Multipliers	321/640 (50 %)
Number of block RAMs	0 (0 %)
Max. Freq.	285.71 MHz

Table 4. Data rate for different configurations of the 4×4, Table 3, implementation.

	4-QAM	16-QAM	64-QAM
$M_T = 2$	142.7 Mbps	285.7 Mbps	428.4 Mbps
$M_T = 3$	214.1 Mbps	428.4 Mbps	642.7 Mbps
$M_T = 4$	285.7 Mbps	571.4 Mbps	857.1 Mbps

3.4 Simulation Results

In this section, we present the simulation results for the Flex-Sphere, and compare the performance of the FPGA fixed-point implementation with that of the optimum floating-point maximum-likelihood (ML) results. Prior to the M-RVD, introduced in section 3, we employ the channel ordering of [20] to further close the gap to ML. Also, we make the assumption that all the streams are using the same modulation scheme. We assume a Rayleigh fading channel

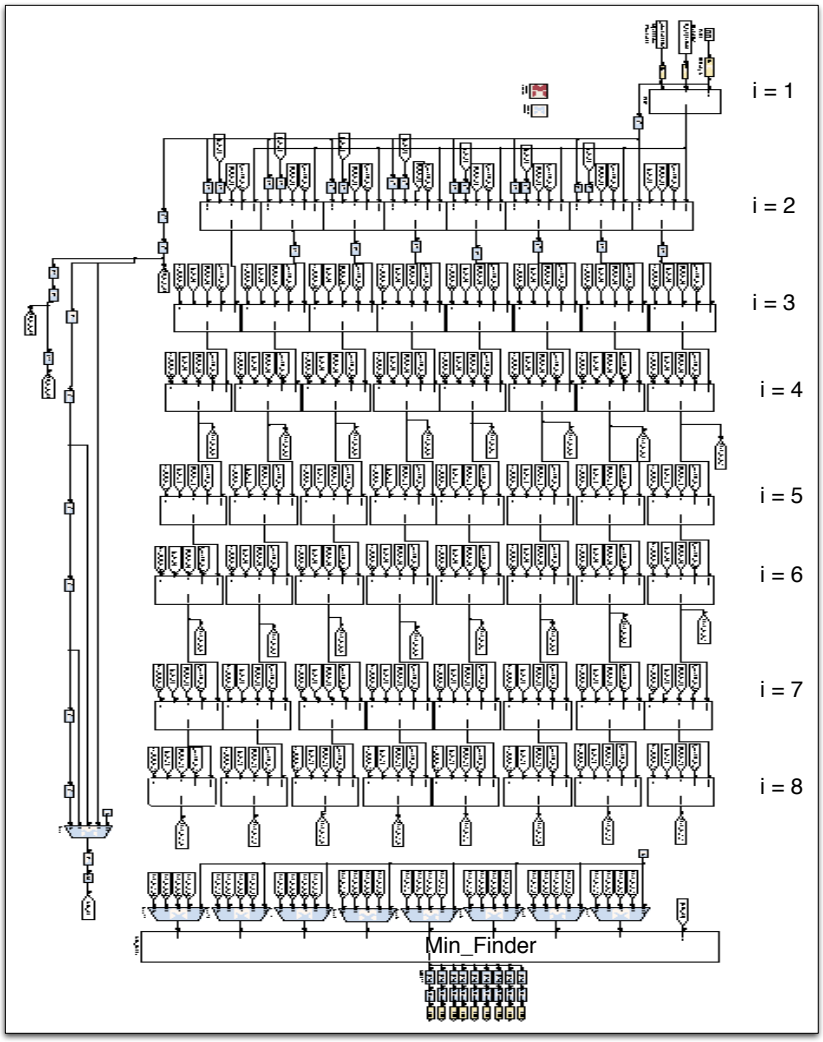

Fig. 5. Xilinx System Generator implementation of the Flex-Sphere detector.

model, i.e. complex-valued channel matrices with the real and imaginary parts of each element drawn from the normal distribution.

In order to ensure that all the antennas in the receiver have similar average received SNR, and none of the users' messages are suppressed due to high power from other users, a power control scheme is employed. Figure 6 shows the simulation results for the maximal 4×4 configuration. As can be seen, the proposed hardware architecture implementation performs within, at most, 1 dB of the optimum maximum-likelihood detection.

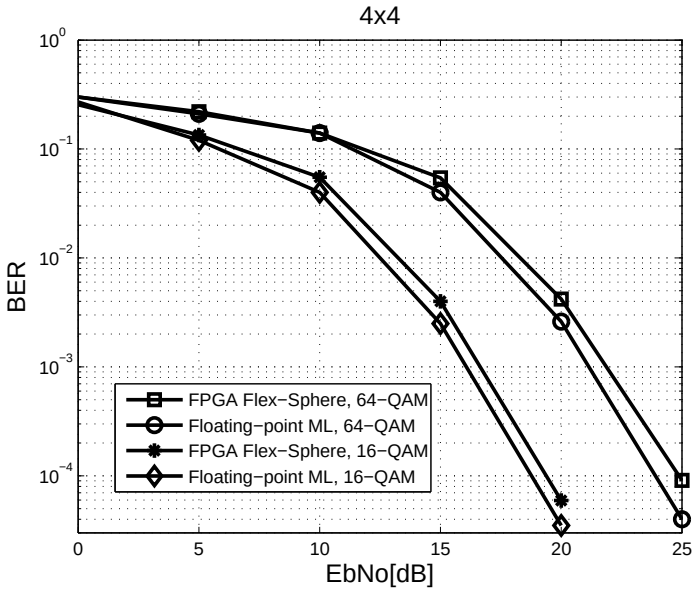

Fig. 6. BER plots comparing the performance of the floating-point maximum likelihood (ML) with the the FPGA implementation. Note that the channel preprocessing of [20] is employed to improve the performance.

4 Beamforming for WiMAX

Multiple Input Mutliple Output (MIMO) antenna systems can be used to increase the data rate (multiplexing gain), improve reliability (diversity gain) or both in certain combinations [1]. The previous section presented schemes for achieving multiplexing gain. This section explains how full diversity gain is achieved in a WiMAX system via beamforming techniques. Implementation challenges of a beamforming WiMAX system are analyzed and results of experiments performed on an FPGA-based testbed are presented.

MIMO schemes that achieve full diversity gain can be either open loop or closed loop. In a closed loop scheme, the transmitter has some knowledge of the instantaneous channel realization and if the forward and reverse channels are not reciprocal, the transmitter acquires channel state information from feedback sent by the receiver. Transmit beamforming is an example of a closed loop scheme that achieves full diversity. In an open loop scheme, the transmitter does not require knowledge of the channel state information; Space Time Codes (STC) are an example of open loop full diversity achieving schemes [8]. Although beamforming and STC schemes can achieve full diversity gain, it is known that beamforming schemes can achieve lower BER and lower probability of outage [5]. The potential for more reliable communication

at higher data rates has made closed loop transmit beamforming techniques part of the physical layer of standards for future wireless communications, like for example 802.11n, WiMAX, and 3GPP [6].

In a beamforming scheme the transmitter adapts to the instantaneous channel conditions. If the forward and reverse channels are reciprocal the transmitter can estimate the channel from pilots sent by the receiver. If the forward and reverse channels are not reciprocal then channel state information is feed back from the receiver to the transmitter via the reverse link. Channel state information at the receiver can be quantized into a few feedback bits using codebook based feedback [5][2], this approach is known as codebook based beamforming and has been considered in WiMAX Frequency Division Duplexing mode, where the forward and reverse channels are not in the same frequency band hence they are not reciprocal.

In a codebook based beamforming scheme the codebook is known to both the transmitter and the receiver; the codebook contains a list of all the possible codewords or beamforming vectors. The receiver estimates the channel and runs a search over the codewords in the codebook to find the beamforming vector that yields the highest Signal to Noise Ratio (SNR) for the given channel conditions. The receiver feeds back to the transmitter the index of the beamforming vector to use for transmission. It has been shown that in a codebook based beamforming system, a few bits of feedback (small codebook size) can achieve performance close to infinite feedback (infinite codebook size), however, the search for the beamforming vector can require a large amount of resources. This section will present implementation requirements and alternatives for efficient implementation of real-time codebook based beamforming WiMAX systems. The performance of different beamforming schemes will be evaluated via simulation and experimental results.

4.1 Beamforming in Wideband Systems

The physical layer of the WiMAX standard achieves wideband communication via Orthogonal Frequency Division Mutliplexing (OFDM). OFDM is a multicarrier scheme in which subcarriers occupy orthogonal narrowband channels. Beamforming is a narrowband scheme that can be easily extended to wideband OFDM systems by applying beamforming techniques per subcarrier. Figure 7 shows a codebook based beamforming MIMO-OFDM system. The system has M_T transmitter antennas, M_R receiver antennas, and K data subcarriers. $\tilde{s}_k$ is used to represent the complex symbol transmitted on subcarrier k, $1 \leq k \leq K$.

In an OFDM system the signal to transmit is first constructed in the frequency domain and then converted to time domain using an Inverse Fast Fourier Transform (IFFT). A Cyclic Prefix (CP) is added to the time domain signal in order to simplify signal processing at the receiver and avoid Inter Symbol Interference (ISI) [1]. At the receiver the CP is removed and the

received signal is converted to frequency domain using a Fast Fourier Transform (FFT). As shown in Fig. 7, in a beamforming MIMO-OFDM system the subcarriers are beamformed in the frequency domain and Maximum Ratio Combining (MRC) at the receiver is also implemented in the frequency domain. For a beamforming MIMO-OFDM system the baseband relationship between the complex symbol transmitted on subcarrier k and the corresponding received signal y_k is given by

$$y_k = \mathbf{z}_k^H \tilde{\mathbf{H}}_k \mathbf{w}_{b_k} \tilde{s}_k + \mathbf{z}_k^H \tilde{\mathbf{n}}_k. \tag{14}$$

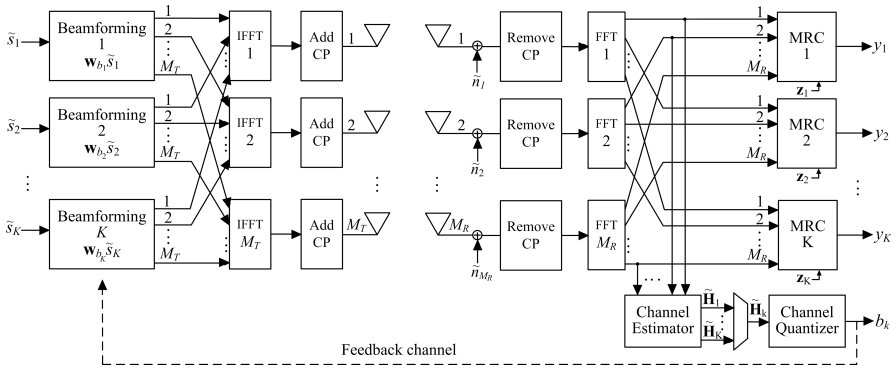

Fig. 7. Baseband representation of a codebook based beamforming MIMO-OFDM system. The FFT and IFFT blocks compute a Fast Fourier Transform and Inverse Fast Fourier Transform respectively. CP is used to denote cyclic prefix.

The $M_T \times 1$ vector used for beamforming subcarrier k is represented by $\mathbf{w}_{b_k}$. The K beamforming vectors $\mathbf{w}_{b_1}, \ldots, \mathbf{w}_{b_K}$ are chosen from a beamforming codebook $\mathcal{W}$ of cardinality $|\mathcal{W}| = 2^B$, hence, $\mathcal{W} = \{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_{|\mathcal{W}|}\}$, and index b_k specifies the index of the beamforming vector chosen to beamform subcarrier k, $1 \leq b_k \leq |\mathcal{W}|$. The channel matrix corresponding to subcarrier k is represented by the $M_R \times M_T$ matrix $\tilde{\mathbf{H}}_k$, the entries of $\tilde{\mathbf{H}}_k$ are $i.i.d$ and each entry is a circularly symmetric complex Gaussian random variable with zero mean and unit variance per complex dimension. To simplify analysis it is assumed that the receiver has perfect knowledge of the channel matrix $\tilde{\mathbf{H}}_k$. The $M_R \times 1$ vector $\mathbf{z}_k$ is the MRC vector for subcarrier k. The $M_R \times 1$ noise vector $\tilde{\mathbf{n}}_k$ has $i.i.d$ entries where each entry is assumed to be circularly symmetric complex Gaussian with zero mean and variance N_0 per complex dimension. Each subcarrier is transmitted with the same average power E_s. This constraint is met by setting $\mathbf{E}\left[|\tilde{s}_k|^2\right] = E_s$ and $\|\mathbf{w}_{b_k}\| = 1$. The WiMAX standard specifies beamforming codebooks for different number of transmitter antennas and codebook cardinalities; all the beamforming vectors in the WiMAX codebooks satisfy $\|\mathbf{w}_{b_k}\| = 1$ ($\|\cdot\|$ is used to denote the vector two-norm and $(\cdot)^H$ denotes the conjugate transposition of a matrix).

The SNR for subcarrier k is given by

$$\mathrm{SNR}_k = \frac{E_s \left| \mathbf{z}_k^H \tilde{\mathbf{H}}_k \mathbf{w}_{b_k} \right|^2}{\|\mathbf{z}_k\|^2 N_0}, \tag{15}$$

the MRC vector and the index of the beamforming vector that maximize SNR_k are [2]

$$\mathbf{z}_k = \frac{\tilde{\mathbf{H}}_k \mathbf{w}_{b_k}}{\|\tilde{\mathbf{H}}_k \mathbf{w}_{b_k}\|} \tag{16}$$

and

$$b_k = \arg \max_{1 \le i \le |\mathcal{W}|} \|\tilde{\mathbf{H}}_k \mathbf{w}_i\|^2 \tag{17}$$

respectively.

In a codebook based beamforming system (like the one considered in WiMAX Frequency Division Duplexing mode) the beamforming codebook $\mathcal{W}$ is known to both the transmitter and the receiver. The K indices $b_1, b_2, \ldots, b_K$ computed by the channel quantizer at the receiver are feedback to the transmitter. Based on these indices the transmitter beamforms subcarriers $1, 2, \ldots, K$ using vectors $\mathbf{w}_{b_1}, \mathbf{w}_{b_2}, \ldots, \mathbf{w}_{b_K}$ respectively. The K indices $b_1, b_2, \ldots, b_K$ computed by the channel quantizer are also used to compute the MRC vectors as shown in (16). Since the codebook has cardinality $|\mathcal{W}| = 2^B$, each index b_k is represented using B bits. The total amount of feedback bits is equal to KB, which corresponds to B bits of feedback information per subcarrier. The WiMAX standard specifies different possible mechanisms to send the KB bits of feedback information from the receiver to the transmitter, in some configurations (e.g. adjacent subcarrier permutation [7]) the amount of feedback bits can be reduced by clustering subcarriers.

The process of quantizing each subcarrier channel $\tilde{\mathbf{H}}_k$ into B bits takes place at the channel quantizer. Since each subcarrier goes through a different channel, each subcarrier channel is quantized independently. In order to reutilize hardware resources the quantizer processes one subcarrier channel at a time, as shown in Figure 7. The input output relationship of the channel quantizer is given by equation (17) which is computed by exhaustive search over the elements in the codebook $\mathcal{W}$.

A codebook based beamforming MIMO system achieves full diversity if $|\mathcal{W}| \geq M_T$ and the beamforming codebook is designed based on the Grassmannian beamforming criterion [2]. The WiMAX standard specifies codebooks for 2, 3, and 4 transmit antennas. For 2 transmit antennas the WiMAX standard specifies a codebook of size $|\mathcal{W}| = 8$, for 3 and 4 transmitter antennas the standard specifies one codebook of size $|\mathcal{W}| = 8$ and one codebook of size $|\mathcal{W}| = 64$. The WiMAX codebooks seem to be designed based on the Grassmannian criterion [4] and it can be verified via Monte Carlo simulation that WiMAX codebooks achieve full diversity.

4.2 Computational Requirements and Performance of a Beamforming System

Implementation of the beamforming and MRC blocks in Figure 7 is straightforward, the most intensive computation in each of the blocks is a vector multiplication. Also, the K beamforming and K MRC blocks can be reduced to one beamforming block and one MRC block by processing one subcarrier at a time.

While implementation of the beamforming and MRC blocks is not resource intensive, implementation of the channel quantizer in Figure 7 can require a large amount of resources depending on the number of transmitter antennas and codebook size. The entries of $\tilde{\mathbf{H}}_k$ are complex numbers and the WiMAX standard specifies that the entries of $\mathbf{w}_i$ are complex numbers rounded to four decimal places, hence, computing $||\tilde{\mathbf{H}}_k\mathbf{w}_i||^2$ for all the $|\mathcal{W}|$ codewords requires $|\mathcal{W}|M_TM_R$ complex multiplications, $|\mathcal{W}|M_TM_R - |\mathcal{W}|M_R$ complex additions, $2|\mathcal{W}|M_R$ real multiplications, and $2|\mathcal{W}|M_R - |\mathcal{W}|$ real additions. After computing $||\tilde{\mathbf{H}}_k\mathbf{w}_i||^2$ for all $|\mathcal{W}|$ codewords the channel quantizer searches for the codeword with the largest $||\tilde{\mathbf{H}}_k\mathbf{w}_i||^2$. Implementing a tree search requires $|\mathcal{W}|-1$ relational blocks, these are blocks that compare two inputs and output the greatest of the two.

The total amount of resources required for channel quantization for one subcarrier in different WiMAX configurations is shown in Table 5. Notice that as the number of antennas and codebook cardinality increases, the number of resources required increases dramatically. The maximum number of embedded multipliers in an FPGA is 512 for the Xilinx Virtex-4 family [10] and 1056 for the Xilinx Virtex-5 family [11], hence, implementing the channel quantizer can become a bottleneck for the implementation of WiMAX systems.

Table 5. Resources required for channel quantization.

	$M_T = 2, M_R = 1$ $\lvert\mathcal{W}\rvert = 8$	$M_T = 3, M_R = 3$ $\lvert\mathcal{W}\rvert = 8$	$M_T = 4, M_R = 4$ $\lvert\mathcal{W}\rvert = 64$
Complex Multipliers	16	72	1024
Complex Adders	8	48	768
Real Multipliers	16	48	512
Real Adders	8	40	448
Relational	7	7	63

The amount of resources required for channel quantization can be reduced by reutilizing resources, but resource reutilization would increase the latency of the channel quantizer. Reutilization is possible as long as the timing constraints are met. In a WiMAX system the timing constraints can be quite tight, especially in high mobility scenarios and in Frequency Division Duplexing (FDD) mode. In a high mobility scenario the channel coherence interval

decreases and feedback information must be sent before it becomes stale and in FDD mode feedback information can potentially be sent as soon as it is available. In both FDD mode and high mobility the faster the feedback information is sent the more throughput the system will have, since more payload data will be sent during a coherence interval.

Another alternative to reduce the amount of resources required for channel quantization is to use a Mixed Codebook (MC) scheme as proposed in [12]. In an MC scheme, the WiMAX codebook is used at the transmitter for beamforming and at the receiver for MRC and a mapped version of the WiMAX codebook is used at the receiver's channel quantizer. The structure of the mapped WiMAX codebook allows a reduction of the number of complex multipliers required for channel quantization. The mapped version of the WiMAX codebook can be obtained via the vector mapping proposed in [12]. Notice that the MC scheme remains WiMAX compliant because the mapped WiMAX codebook is only used for channel quantization. The codebook mapping and the efficient implementation using the mapped WiMAX codebook are explained below.

The vector mapping proposed in [12] finds for each beamforming vector $\mathbf{w}_i$ in codebook $\mathcal{W}$ a corresponding vector $\mathbf{m}_i$ such that

$$\mathbf{m}_i = g_i \mathbf{c}_i, \tag{18}$$

where

$$g_i = \frac{1}{\|\mathbf{c}_i\|}, \tag{19}$$

$$\mathbf{c}_i = \arg\max_{\mathbf{u}\in\mathcal{A}} \left| \left\langle \frac{\mathbf{u}}{\|\mathbf{u}\|}, \mathbf{w}_i \right\rangle \right|, \tag{20}$$

and $\mathcal{A}$ denotes the set of T-dimensional vectors whose entries belong to set α. We label α the codebook alphabet and use $< \mathbf{x}, \mathbf{y} >$ to denote the inner product between $\mathbf{x}$ and $\mathbf{y}$: $< \mathbf{x}, \mathbf{y} >= \mathbf{x}^H \mathbf{y}$. We use $\mathcal{W}_M$ to denoted the mapped version of codebook $\mathcal{W}$. The vectors $\mathbf{m}_1, \ldots, \mathbf{m}_N$ obtained by vector mapping the codewords in codebook $\mathcal{W}$ correspond to codewords $1, \ldots, N$ in codebook $\mathcal{W}_M$, hence, $\mathcal{W}_M = \{\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_N\}$. The vectors $\mathbf{m}_i$ are normalized ((18) and (19)) because beamforming vectors must be unit norm in order to satisfy the transmit power constraint.

If a mapped codebook $\mathcal{W}_M$ is used for channel quantization then the quantization operation in (17) can be rewritten as

$$\begin{aligned} b &= \arg \max_{1\le i\le N} \|\tilde{\mathbf{H}}_k \mathbf{m}_i\|^2 \\ &= \arg \max_{1\le i\le N} \|\tilde{\mathbf{H}}_k g_i \mathbf{c}_i\|^2 \\ &= \arg \max_{1\le i\le N} g_i^2 \|\tilde{\mathbf{H}}_k \mathbf{c}_i\|^2. \end{aligned} \tag{21}$$

We consider the case in which the codebook alphabet is equal to $\alpha = \{0, +1, -1, +j, -j\}$. For this codebook alphabet complex multiplications required to compute $\tilde{\mathbf{H}}\mathbf{c}_i$ can be implemented via simple sign changes and swapping of the real and imaginary parts of the channel matrix entries (notice from (20) that the entries of $\mathbf{c}_i$ belong to α). Table 6 compares the amount of resources required for channel quantization when using a mapped WiMAX codebook with alphabet $\alpha = \{0, +1, -1, +j, -j\}$ with the amount of resources required when using a WiMAX codebook. Observe from Table 6 that when using a mapped WiMAX codebook, the number of multiplications required reduces to zero. This reduction is obtained by increasing the number of five input multiplexers and negators (a negator block is a very simple block that computes the arithmetic negation or two's complement of its input). For both ASIC and FPGA implementations, 2048 five input multiplexers, 2048 negators, and 64 real multipliers, require fewer resources than implementing 1024 complex multipliers.

Using an MC scheme will affect performance because the codebook used for beamforming at the transmitter and MRC at the receiver is different from the codebook used for channel quantization at the receiver. However, the performance loss is very small, since by construction, the mapped WiMAX codebook has quantization regions similar to the quantization regions of the WiMAX codebook it is obtained from [12]. This is an important characteristic of the mapped WiMAX codebook because the quantization regions of the codebook determine the performance of a codebook based beamforming MIMO system. If the WiMAX codebook $\mathcal{W}$ and the mapped WiMAX codebook $\mathcal{W}_M$ have the same quantization regions then the result of channel quantization using codebook $\mathcal{W}$ will be equal to the result of channel quantization using codebook $\mathcal{W}_M$, equivalently

$$\arg \max_{1 \leq i \leq N} \|\tilde{\mathbf{H}}_k \mathbf{w}_i\|^2 = \arg \max_{1 \leq i \leq N} \|\tilde{\mathbf{H}}_k \mathbf{m}_i\|^2. \tag{22}$$

In general, codebooks $\mathcal{W}$ and $\mathcal{W}_M$ will not have exactly the same quantization regions but the are similar enough so that for most of the channel realizations equation (22) holds.

We have only considered alphabet $\alpha = \{0, +1, -1, +j, -j\}$, other alphabets can be used, however, analysis in [12] showed that choosing $\alpha = \{0, +1, -1, +j, -j\}$ results in a good tradeoff between implementation complexity and performance. A larger codebook alphabet can result in better performance but requires more resources for channel quantization.

Figure 8 shows the performance of different beamforimng WiMAX configurations, all simulations were run for a one subcarrier model, the results obtained represent the per subcarrier behavior in an OFDM system. The results in Figure 8 correspond to the best case scenario of perfect channel estimate at the receiver, noiseless and zero delay feedback, and floating point processing. The results in Figure 8 show that in a codebook based beamforming MIMO system, like FDD WiMAX, a few bits of feedback can achieve performance

Table 6. Resources required for channel quantization using a WiMAX codebook and a mapped WiMAX codebook with alphabet $\alpha = \{0, +1, -1, +j, -j\}$. Results correspond to a system with 4 transmitter and 4 receiver antennas and a codebook of 64 codewords.

	WiMAX Codebook	Mapped WiMAX Codebook
Complex Multipliers	1024	0
Complex Adders	768	768
Real Multipliers	512	576
Real Adders	448	448
Negators	0	2048
5-input Multiplexer	0	2048
Relational	63	63

close to the ideal case of infinite feedback. Also, observe that using an MC scheme results in small performance degradation, and as shown in Table 6, the amount of resources required for channel quantization can be significantly reduced by using an MC scheme. Figure 8 also shows the performance of the Alamouti STC [9] which is an open loop scheme. Observe that the beamforming and Alamouti schemes have the same diversity gain (2×1 curves decay with same slope at high EsNo), but beamforming has better performance (lower BER).

4.3 Beamforming Experiments using WARPLab

WARPLab is a framework for rapid prototyping of physical layer algorithms. The WARPLab framework combines the ease of MATLAB with the capabilities of the Wireless Open Access Research Platform (WARP) developed at Rice University [13]. This section describes in detail the WARPLab framework and presents experimental results that show the performance gains that can be obtained using beamforming in a WiMAX system.

WARPLab Framework

WARP provides a unique platform to develop, implement, and test advanced wireless communication algorithms. The platform architecture consists of four main components: custom hardware, platform support packages, open-access repository, and research applications.

The custom hardware allows implementation and scalability of sophisticated signal processing algorithms and provides extensible peripheral options for radios and user interface. The main component of the WARP hardware is a Xilinx Virtex-II Pro FPGA, a new version of the WARP hardware will soon be released, in this new version the Xilinx Virtex-II Pro FPGA has been replaced by a more powerful Xilinx Virtex-4 FPGA. The WARP board, shown

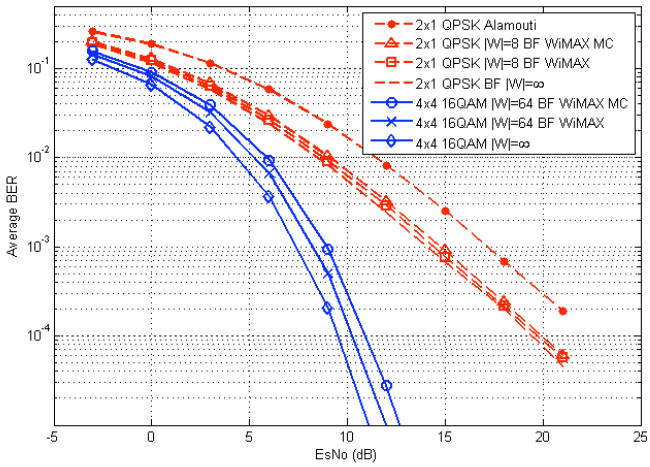

Fig. 8. Performance of a $M_T \times M_R$ beamforming WiMAX system for different values of M_T, M_R, and $|\mathcal{W}|$. BF is used to denote beamforming, MC is used to denote mixed codebook scheme, and $|\mathcal{W}| = \infty$ is used to denote an infinite feedback scenario.

in Figure 9, has four daughter card slots, each slot is connected to a dedicated bank of I/O pins on the FPGA, providing a flexible, high-throughput interface. As shown in Figure 9, the four daughter card slots can be used to connect the FPGA to four different radio boards so that up to a 4×4 MIMO system can be built. The radio boards have been designed by Rice University students, these boards are capable of targeting both the 2.4 GHz and 5 GHz ISM bands and are intended for wide band applications, such as OFDM, with a bandwidth up to 40 MHz.

The platform support packages facilitate seamless use of the WARP hardware by researchers working at all layers of wireless network design. The open-access repository [14], accessible from the Internet, is the central archive for all source codes, models, platform support packages, application building blocks, research applications, design documents, and hardware design files associated with WARP. The contents of the repository are verified by the project administrator at Rice University.

WARP allows clean-slate design and prototyping of physical layer algorithms for wireless communications via two possible design flows: 1) real time implementation and 2) WARPLab framework that allows real time RF transmission and offline processing on a host PC. In real time implementation, all the signal processing is implemented in the FPGA, allowing implementation of a complete end-to-end real-time system. However, many physical layer researchers are interested in rapid prototyping and over-the-air testing of new algorithms without having to deal with the details of FPGA implementation,

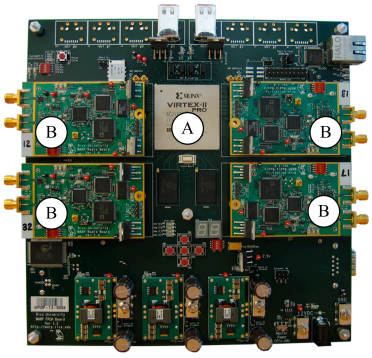

Fig. 9. WARP board with a radio board in each of the four daughtercard slots. A: Xilinx Virtex-II Pro FPGA. B: Radio board.

and without having to implement mechanisms that belong to higher layers, like carrier sensing, contention resolution protocols, and packet detection. To meet these requirements, Rice University has developed the WARPLab framework, which allows generation of waveforms in MATLAB and provides simple m-code functions that allow over-ther-air transmission of these waveforms using the WARP hardware.

The basic WARPLab setup is show in Figure 10, two WARP nodes are connected to a host PC via an Ethernet switch. Up to 16 WARP nodes can be connected to the switch and controlled from the host PC. A user first uses MATLAB to construct waveforms that their algorithm would transmit. These waveforms are loaded into the assigned WARP transmit nodes via Ethernet links that are controlled by custom code on both the PC and WARP nodes. The host PC then triggers the beginning of the experiment by telling the transmit nodes to begin their transmissions and the receive nodes to begin capturing data from the radio. Once transmission and capture are completed, the captured waveforms are passed to the host PC via the Ethernet links. The user can then use MATLAB to process the received waveforms and determine the effects of real radios and wireless channels on their novel algorithm.

The WARPLab framework provides the software necessary for easy interaction with the WARP nodes directly from the MATLAB workspace. The software consists of FPGA code and m-code functions which are all available in the WARP repository [15].

Experiment Setup and Results

This section shows experiment results for a beamforming system with two transmitter antennas and one receiver antenna, the system was designed and tested using the WARPLab framework. Experiments were performed over a wireless channel emulated using Spirent's SR-5500 wireless channel emulator. The experiment setup is shown in Figure 11. Two radios at the transmitter

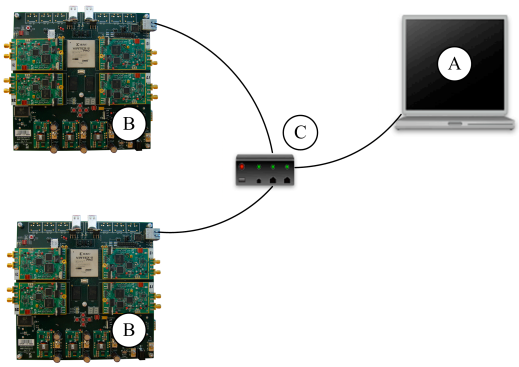

Fig. 10. Basic WARPLab setup. A: Host PC, B: WARP node, C: Switch. The switch is connected to the WARP nodes and the host PC via Ethernet links.

node are connected to the two RF inputs of the channel emulator which emulates two independent RF wireless channels. The two RF outputs of the channel emulator are added to emulate a 2×1 system, the output of the RF adder is connected to one radio at the receiver node. The channel emulator is connected to the host computer via ethernet, the host computer controls the two WARP nodes and the channel emulator via the ethernet links.

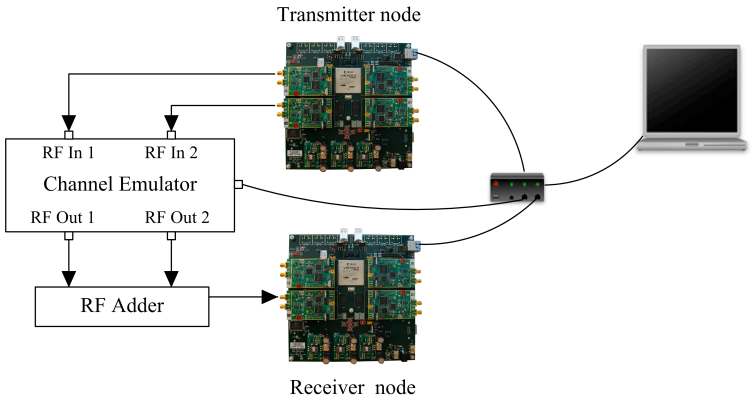

Fig. 11. Experiment setup. The basic WARPLab setup was connected to a channel emulator and an RF adder to emulate a system with two transmitter antennas and one receive antenna.

The system implemented is a single subcarrier system, the results represent the per subcarrier behavior in an OFDM system. Table 7 summarizes the experiment conditions, the two emulated RF channels were both set using exactly the same parameters. Since the delay spread is much smaller than the symbol period, the transmitted signal goes through a flat fading channel.

Table 7. Experiment Conditions

Parameter	Value
Number of transmitter antennas	2
Number of receiver antennas	1
Carrier frequency	2.4 GHz
Number of subcarriers	1
Bandwidth	625 kHz
Sampling frequency	40 MHz
Pulse shaping filter	Squared Root Raised Cosine (SRRC)
SRRC roll-off factor	1
Symbol time	3.2 μs
Modulation	16 QAM
Coding Rate	1 (No error correction code)
Energy per symbol	-20 dBm at input of channel emulator
Paths per emulated RF channel	3
Envelope per path	Rayleigh flat fading all 3 paths
Fading Doppler per path	0.1 Hz (0.04 km/h) in all 3 paths emulates block fading channel
Delay per path	Path 1 = 0 μs , Path 2 = 0.05 μs, Path 3 = 0.1 μs
Relative path loss	Path 1 = 0 dB , Path 2 = 3.6 dB Path 3 = 7.2 dB

Transmission of the payload data was done in packets of 110 symbols. The number of symbols per packet was limited to 110 due to the characteristics of the transmitted signal (symbol time of 3.2μs and sampling frquency of 40 MHz) and the maximum number of samples that can be stored per receiver radio in a node, which is limited to 2^{14} samples [15]. As shown in Figure 12, two pilot sequences were sent before transmission of payload. The first pilot sequence was used for computation of the channel estimate used for channel quantization, the total pilot energy was set equal to twice the energy per symbol. The second pilot sequence was a beamformed pilot sequence which was used to obtain an estimate of the channel times the beamforming vector, this estimate was used for MRC at the receiver, the total pilot energy was equal to the total energy per symbol. The error free feedback channel was implemented in the host PC (the host PC is connected to both transmitter and receiver). The feedback delay was approximately 60 ms, this is the time it took to send training, estimate the channel, quantize the channel estimate, and start transmitting the beamformed signal.

In order to compare the performance of a closed loop scheme like beamforming and an open loop scheme like Alamouti, a 2×1 Alamouti scheme was also implemented and tested using the WARPLab framework. In the Alamouti scheme, transmission of the payload data was also done in packets of 110

symbols. For the Alamouti implementation, only one pilot sequence was sent, payload was sent immediately after the pilot sequence. The total pilot energy was equal to the total energy per symbol.

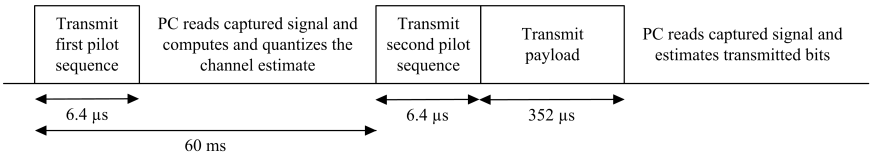

Fig. 12. Time diagram of transmitted signal and PC processing for beamforming experiment.

Simulation results in Figure 8 showed that the beamforming scheme has better performance than the Alamouti scheme and that the MC scheme for efficient implementation results in small performance degradation. These two observations are verified by the experiment results shown in Figure 13. The experiment results in Figure 13 also show that, as expected, few feedback bits have performance close to the performance of infinite feedback. Observe that the plots of the experiment results are not as smooth as the plots of the simulation results in Figure 8, it is very likely that is due to hardware non linearities that were not considered in simulations.

5 Conclusion

In this chapter, we discussed the architectural challenges of spatial multiplexing and diversity gain schemes, and further, introduced FPGA-centric architectures and experiments for these systems. We introduced a flexible architecture and implementation of a spatial multiplexing MIMO detector, Flex-sphere, and its FPGA implementation. We also presented a hardware architecture for beamforming for WiMAX, as a way to enhance the diversity and performance in the next-generation wireless systems. Finally, we showed how we utilized the WARP platform for studying the effects of MIMO systems in real-world scenarios.

References

1. "3GPP Long Term Evolution : http://www.3gpp.org/."
2. "IEEE 802.16 Working Group : http://wirelessman.org/."
3. G. Foschini, "Layered space-time architecture for wireless communication in a fading environment when using multiple antennas," *Bell Labs. Tech. Journal*, vol. 2, 1996.

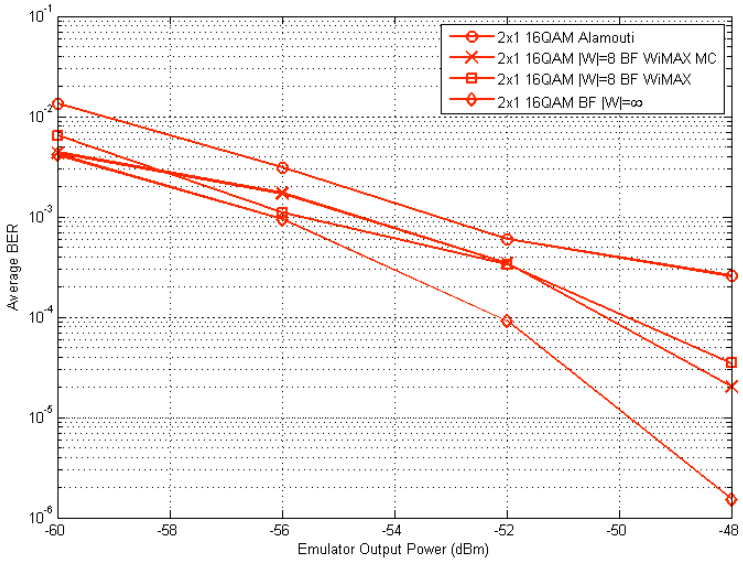

Fig. 13. Experiment results for a 2×1 beamforming system and a 2×1 Alamouti system. BF is used to denote beamforming, MC is used to denote mixed codebook scheme, and $|\mathcal{W}| = \infty$ is used to denote an infinite feedback scenario

4. G.D. Golden, G. J. Foschini, R. A. Valenzuela and P. W. Wolniansky, "Detection algorithms and initial laboratory results using V-BLAST space-time communication architecture," *Electronics Letters*, vol. 35, pp. 14–15, 1999.
5. Z Guo and P. Nilsson, "A 53.3 Mb/s 4×4 16-QAM MIMO decoder in 0.35μm CMOS," *IEEE Int. Symp. Circuits Syst.*, vol. 5, pp. 4947–4950, May 2005.
6. M. O. Damen, H. E. Gamal and G. Caire, "On maximum likelihood detection and the search for the closest lattice point," *IEEE Trans. on Inf. Theory*, vol. 49, no. 10, pp. 2389–2402, Oct. 2003.
7. U. Fincke and M. Pohst, "Improved methods for calculating vectors of short length in a lattice, including a complexity analysis," *Math. Computat.*, vol. 44, no. 170, pp. 463–471, Apr. 1985.
8. B. Hochwald and S. ten Brink, "Achieving near-capacity on a multiple-antenna channel," *IEEE Trans. on Comm.*, vol. 51, pp. 389–399, Mar. 2003.
9. A. Burg, M. Borgmann, M. Wenk, M. Zellweger, W. Fichtner, and H. Bolcskei, "VLSI implementation of MIMO detection using the sphere decoding algorithm," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 7, pp. 1566–1577, July 2005.
10. K. Amiri and J. R. Cavallaro, "FPGA implementation of dynamic threshold sphere detection for MIMO systems," *40th Asilomar Conf on Signals, Systems and Computers*, Nov 2006.
11. D. Garrett, L. Davis, S. ten Brink, B. Hochwald and G. Knagge, "Silicon complexity for maximum likelihood MIMO detection using spherical decoding," *IEEE JSSC*, vol. 39, no. 9, pp. 1544–1552, Sep. 2004.

12. Z. Guo and P. Nilsson, "Algorithm and implementation of the K-Best sphere decoding for MIMO detection," *IEEE JSAC*, vol. 24, no. 3, pp. 491–503, Mar. 2006.
13. K. Wong, C. Tsui, R. S. Cheng and W. Mow, "A VLSI architecture of a K-best lattice decoding algorithm for MIMO channels," *IEEE Int. Symp. Circuits Syst.*, vol. 3, pp. 273–276, May 2002.
14. K. Amiri, J. R. Cavallaro, C. Dick and R. Rao, "A high throughput congurable SDR detector for multi-user MIMO wireless systems," *Springer Journal of Signal Processing*, 2009.
15. K. Amiri, C. Dick, R. Rao and J. R. Cavallaro, "Flex-Sphere: An FPGA Configurable Sort-Free Sphere Detector for Multi-user MIMO Wireless Systems," *Proc. of SDR Forum*, Oct. 2008.
16. L. G. Barbero and J. S. Thompson, "FPGA design considerations in the implementation of a fixed-throughput sphere decoder for MIMO systems," *Field Programmable Logic and Applications, 2006. FPL '06. International Conference on*, Aug. 2006.
17. C. P. Schnorr and M. Euchner, "Lattice basis reduction: improved practical algorithms and solving subset sum problems," *Math. Programming*, vol. 66, no. 2, pp. 181–191, Sep. 1994.
18. K. Amiri, C. Dick, R. Rao and J. R. Cavallaro, "Novel sort-free detector with modified real-valued decomposition (M-RVD) ordering in MIMO systems," *Proc. of IEEE Globecom*, Dec. 2008.
19. "Xilinx : http://www.xilinx.com/."
20. L. G. Barbero and J. S. Thompson, "A fixed-complexity MIMO detector based on the complex sphere decoder," *IEEE 7th Workshop on Signal Processing Advances in Wireless Communications, 2006. SPAWC '06*, Jul. 2006.
21. "WARP : http://www.warp.rice.edu/."

References

1. L. Zheng, D.N.C. Tse (2003). Diversity and multiplexing: A fundamental tradeoff in multiple-antenna channels. IEEE Trans. Inform. Theory 49:1073–1096.
2. D. J. Love, R. W. Heath, T. Strohmer (2003). Grassmannian beamforming for multiple-input multiple-output wireless systems. IEEE Trans. Inform. Theory 49:2735–2747.
3. D.J. Love, R.W Heath, W. Santipach, M. Honig (2004). What is the value of limited feedback for MIMO channels?. IEEE Communications Magazine 42:54–59.
4. D.J. Love, R.W Heath, V. K. N. Lau, D. Gesbert, B. D. Rao, M. Andrews (2008). An overview of limited feedback in wireless communication systems. IEEE JSAC 46:1341–1365.
5. K. Mukkavilli, A. Sabharwal, E.Erkip, B. Aazhang (2003). On beamforming with finite rate feedback in multiple antenna systems. IEEE Trans. Inform. Theory 49:2562–2579.
6. A. Hottinen, M. Kuusela, K. Hugl, J. Zhang, B. Raghothaman (2006). Industrial embrace of smart antennas and MIMO. IEEE Wireless Communications 13:8–16.

7. Air Interface for Fixed and Mobile Broadband Wireless Access system - Amendment for physical and medium acces control layers for combined fixed and mobile operation in licensed bands -Corrigendum 1. IEEE Std. 802.16e, 2006.
8. V. Tarokh, N. Seshadri, A. R. Calderbank (1998). Space-time codes for high data rate wireless communication: Performance criterion and code construction. IEEE Trans. Inform. Theory 44:744–765.
9. S. M. Alamouti (1998). A simple transmit diversity technique for wireless communications. IEEE J. Select. Areas Commun. 16:1451–1458.
10. Xilinx Virtex-4 family overview: http://www.xilinx.com/
11. Xilinx Virtex-5 family overview: http://www.xilinx.com/
12. M.Duarte, A. Sabharwal, C. Dick, R. Rao (2008). A vector mapping scheme for efficient implementation of beamforming MIMO systems. Military Communications Conference MILCOM.
13. WARP: http://warp.rice.edu
14. WARP Repository: http://warp.rice.edu/trac
15. WARP Repository, WARPLab files: http://warp.rice.edu/trac/wiki/WARPLab