

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

185,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Efficient Implementation of MIMO Decoders

Muhammad S. Khairy¹, Mohamed M. Abdallah² and S. E.-D. Habib³

¹*Electrical Engineering and computer Science Dept.,*

Univ. of California, Irvine, Irvine, CA 92697-2625,

²*Electrical and Computer Eng. Dept., Texas A&M Univ. at Qatar, PO Box 23874, Doha,*

³*Electronics and Comm. Dept., Faculty of Eng., Cairo Univ., Giza 12613,*

¹*USA*

²*Qatar*

³*Egypt*

1. Introduction

Multiple-Input Multiple-Output (MIMO) technology has emerged as a promising technology for achieving the high data rates of next generation wireless communication systems. MIMO systems improve either the bit-error rate (BER) performance by using spatial diversity or the data rate via spatial multiplexing. However, Maximum-likelihood (ML) detection for high order MIMO systems faces a major challenge in computational complexity that grows exponentially with the number of transmit and receive antennas. This limits the practicality of these systems from an implementation point of view, particularly for mobile battery-operated devices.

This reality motivated researchers to consider other suboptimal approaches for MIMO decoding, such as Zero Forcing (ZF), Minimum Mean Square Error (MMSE) and VBLAST (Proakis & Salehi, 1994; Guo & Nilsson, 2003; Myllyla et al., 2005). All of these suboptimal approaches vary in performance and complexity. Recently, the sphere decoding (SD) algorithm which is a tree-based search algorithm enabled the implementation of efficient MIMO decoders that achieve near ML performance together with reduced complexity (Burg et al., 2005; Barbero et al., 2005; Khairy et al., 2009). Instead of the exhaustive search over all possible combinations of the transmitted symbols, the SD algorithm reduces the complexity by searching only over a finite number of symbols within a circle of radius R centered at the received vector. While the SD approach provides a near ML solution, the runtime measured by the required operations to find the optimum solution is highly dependent on the received signal to noise ratio and the channel conditions. Consequently, the SD algorithm experiences variable throughput problems as the search radius R for each symbol varies according to the noise levels and the channel coefficients. Moreover, the sequential search results in hardware implementations that are not fully pipelined.

To alleviate these problems, the fixed sphere decoding (FSD) algorithm was recently proposed (Barbero & Thompson, 2006 b). The FSD algorithm achieves a fixed throughput performance and enables fully-pipelined hardware by performing fixed number of operations per detected symbol, independent of the noise level. The FSD algorithm is performed in two stages where the first one is a pre-processing stage for the received

symbols that incorporates a QR decomposition of the channel matrix for the MIMO system. This pre-processing allows for employing a tree-based search algorithm at the second stage that finds the optimal symbol which has the minimum distance to the received symbol.

In this chapter, we present an efficient FPGA prototype of the FSD that implements an efficient QR decomposition of the channel matrix for a MIMO system using CORDIC array techniques followed by the FSD tree-based search algorithm. This prototype provides an improved hardware implementation in terms of area and throughput compared to up-to-date published results in the literature. The FPGA implementation is incorporated with a MATLAB simulation model of an OFDM based MIMO system to validate the hardware design.

As a practical application, we employed the FSD in the receiver of the IEEE802.16e WiMAX MIMO system. The IEEE 802.16e standard supports three types of MIMO space time codes (STC), referred to as matrix A, B, and C. These STC codes achieve different levels of throughput and diversity depending on the quality of the MIMO channels. We employed the FSD as a STC detector to decode the different STC schemes. In achieving that, the received signal is adaptively pre-processed according to the STC type prior feeding it to the FSD. At the end of the chapter, we present the performance of a two by two STC-WiMAX system with the FSD in terms of bit error rate and throughput. The results indicate that our FPGA implementation achieves throughput values that satisfy the requirements of the WiMAX standard.

2. System model

Consider the OFDMA-based 2x2 WiMAX MIMO system shown in Fig. 1. At the transmitter, the input data is first encoded then modulated using QAM modulation technique. The STC operates on input data symbols sequentially and distributes the data symbols to each antenna path. The WiMAX standard supports three STC referred to in the standard by matrix A, B and C, respectively. In particular, the STC matrix A achieves full diversity by employing the Alamouti coding, while the STC matrix B achieves full rate by employing spatial multiplexing, and the STC matrix C achieves full rate and diversity, by employing the Golden code. The transmitter encodes the incoming symbol streams into two spatial streams using one of the matrices A, B or C. The block of subcarrier mapping and PseudoRandom Binary Sequence (PRBS) function denotes data truncation or puncturing, pilot insertion, and IFFT input packing. Then IFFT is applied to generate the OFDMA symbol.

At the receiver, a FFT operation is first performed at each receive antenna, followed by subcarrier de-mapping and PRBS. The STC Decoder is implemented using the MIMO FSD decoder. The received signal, after performing the FFT operation, can be expressed as follows

$$r = Hs + n \quad (1)$$

where s denotes the vector of transmitted symbols, n is the complex additive white Gaussian noise of zero mean and variance N_0 , and r is the received vector. H denotes the 2×2 channel matrix where each element, h_{ij} , is modeled as Rayleigh fading channel. The channel matrix is assumed to be perfectly known at the receiver.

We develop a rapid prototype of the system model of interest whereby the FSD algorithm is implemented using Stratix II FPGA board and the rest of the system blocks are implemented using MATLAB. Such prototype allows for testing the WiMAX MIMO detectors for several OFDM systems with different FFT sizes.

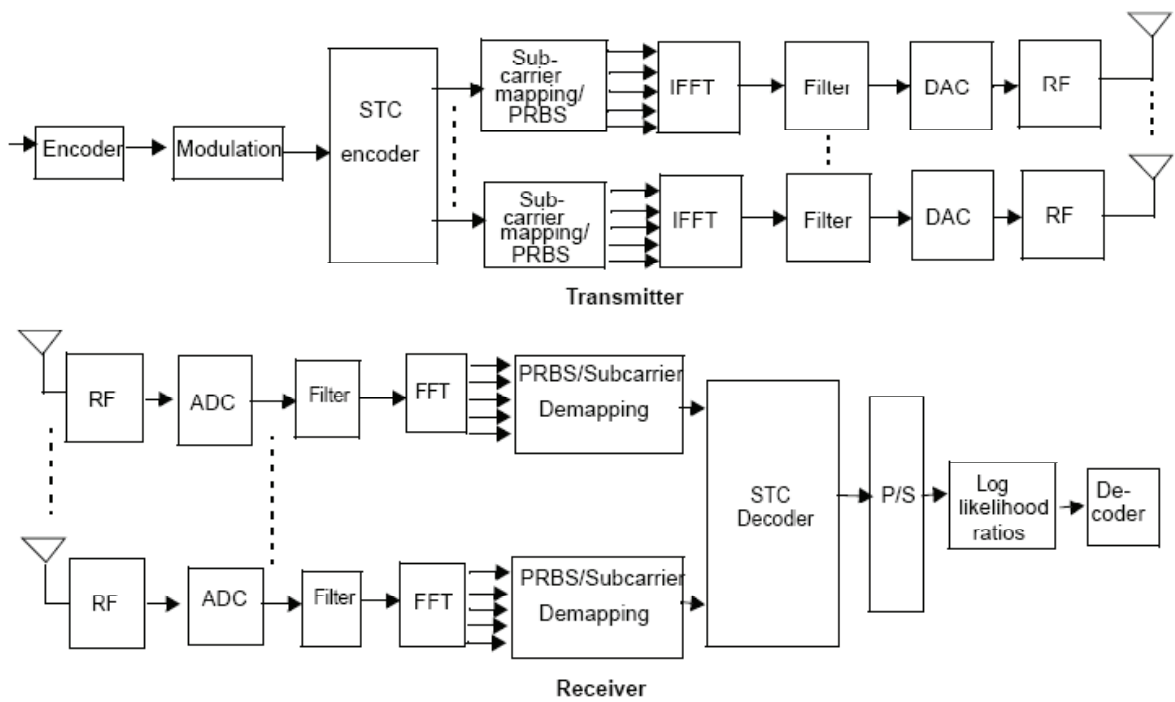


Fig. 1. WiMAX System Model

3. FSD algorithm

The Fixed Sphere Decoder searches over a fixed number of vectors within a circle of radius C centered at the received vector r . This results in fixed complexity and hence fixed throughput. The symbol with the minimum Euclidian distance to the received vector, \hat{s}_{FSD} , can be obtained as follows,

$$\hat{s}_{FSD} = \arg\{\min_{s \in S} \|r - Hs\|^2 \leq C^2\}$$

(2)

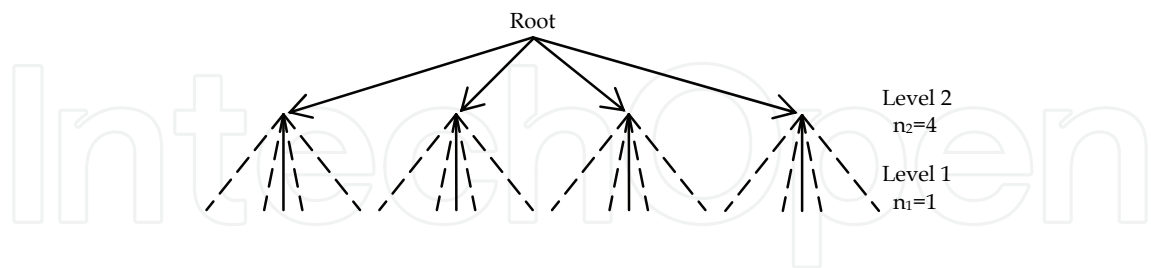


Fig. 2. Tree Search for 2x2 MIMO system and QPSK

The FSD search for the transmitted vectors can be viewed as a tree search of N levels. Fig. 2 shows the tree for 2x2 MIMO system with QPSK modulation. Starting from the root, there exists 4 branches corresponding to 4 QPSK symbols and for each node in the top level there exists another 4 different branches. In general, the search process starts from level N . The number of nodes considered per level is denoted as n_i . (Barbero & Thompson, 2006 b) showed that n_i during the tree search satisfies the following relation

$$E[n_N] \geq E[n_{N-1}] \geq \dots \geq E[n_1]$$

where $E[n_i]$ represents the expected number of visited nodes per level i . The numbers of visited nodes per level are pre-determined. All node symbols of level N are visited where only one node per level is chosen for the next levels. Given that the channel matrix H can be decomposed using QR to an upper triangular matrix R and a unitary matrix Q .

$$H = QR \quad (3)$$

where, $Q^H Q = Q Q^H = I$. Since Q is a unitary matrix, the expression given in (1) can be rewritten as follows,

$$\|r - Hs\|^2 = \|Q^H(r - Hs)\|^2 = \|\hat{y} - Rs\|^2 \quad (4)$$

where $\hat{y} = Q^H r$

Hence the expression in (2) can be equivalently expressed as

$$\hat{s}_{FSD} = \arg\{\min_{s \in S} \|\hat{y} - Rs\|^2 \leq C^2\} \quad (5)$$

The second norm can be expressed as sum of a partial Euclidian distance d_i where:

$$D = \|\hat{y} - Rs\|^2 = \sum_{i=1}^N d_i \quad (6)$$

The partial Euclidian distance d_i can be evaluated as:

$$d_i = |b_i - R_{ii}s_i|^2 \quad (7)$$

where $b_i = \hat{y}_i - \sum_{j=i+1}^N R_{ij}s_j$

We note that using the QR decomposition of the channel matrix H , rather than the Cholesky considered in (Barbero & Thompson, 2005 ; Barbero & Thompson, 2006 b), and subsequently casting the FSD by Eqns. (5-8) leads to improvements in hardware and throughput as will be proved in Section 6 and 7 of this chapter (Khairy et. al, 2009) . In the subsequent sections, we first present an overview of the QR decomposition and the implementation of the FSD, and then we present the decoding technique for the different STC matrices.

4. QR decomposition using CORDIC

The QR decomposition is done using CORDIC (Coordinate Rotation Digital Computer) (Volder, J.E, 1959). CORDIC is used to rotate 2-point vector (x, y) by angle ζ_v to a new vector with coordinates (x', y')

$$x' = \cos(\zeta_v) \times (x - y \tan \zeta_v) \quad (9a)$$

$$y' = \cos(\zeta_v) \times (y + x \tan \zeta_v) \quad (9b)$$

By selecting ζ_v as follows

$$\tan \zeta_v = 2^{-v}, \quad (10)$$

The trigonometric functions of equation 9 are then implemented via right-shifts and add/subtract operations only. The first step of the CORDIC algorithm is to express any arbitrary angle ξ by a sequence of forward or backward rotations obeying equation 10 as follows

$$\zeta = \sum_v \rho_v \zeta_v \tag{11}$$

where $\rho_v = \pm 1$. Throughout all these macro rotations, the multiplication by $\cos \zeta_v$ can be collected together into a single constant

$$K = \prod_v \cos \zeta_v \tag{12}$$

The CORDIC algorithm can be seen as composed of stages performing mini rotations followed by a gain correction. In general, at the mini rotation number i the new rotated vector can be expressed as:

$$y_{i+1} = y_i + \rho_i x_i 2^{-i} \tag{13a}$$

$$x_{i+1} = x_i - \rho_i y_i 2^{-i} \tag{13b}$$

$$\zeta_{i+1} = \zeta_i - \rho_i \times \tan^{-1} 2^{-i} \tag{13c}$$

The CORDIC method provides a pipelined implementation of the given rotations using only shifts and addition/subtractions. There exist two modes of CORDIC Units. The CORDIC vector mode rotates the input vector by the angle necessary to align it along the x-axis. The result of the vector mode is a rotation angle and a scaled magnitude of the original vector. The vectoring function works by minimizing the y component at each rotation. The second CORDIC mode is the rotate mode, which rotates the incoming vector by a certain angle. The QR is implemented using systolic array as shown in Fig.3. The QR operation can be accomplished by a sequence of circular row operations to nullify elements below the main diagonal of the matrix. The round boundary cell performs the vectoring operation, i.e., it computes the angles needed for annihilation of the incoming data samples. The boundary cell sends the angle values to the inner square cells in the same row. The inner square cell calculates the new rotated sample values based on the angle values given from the boundary cell.

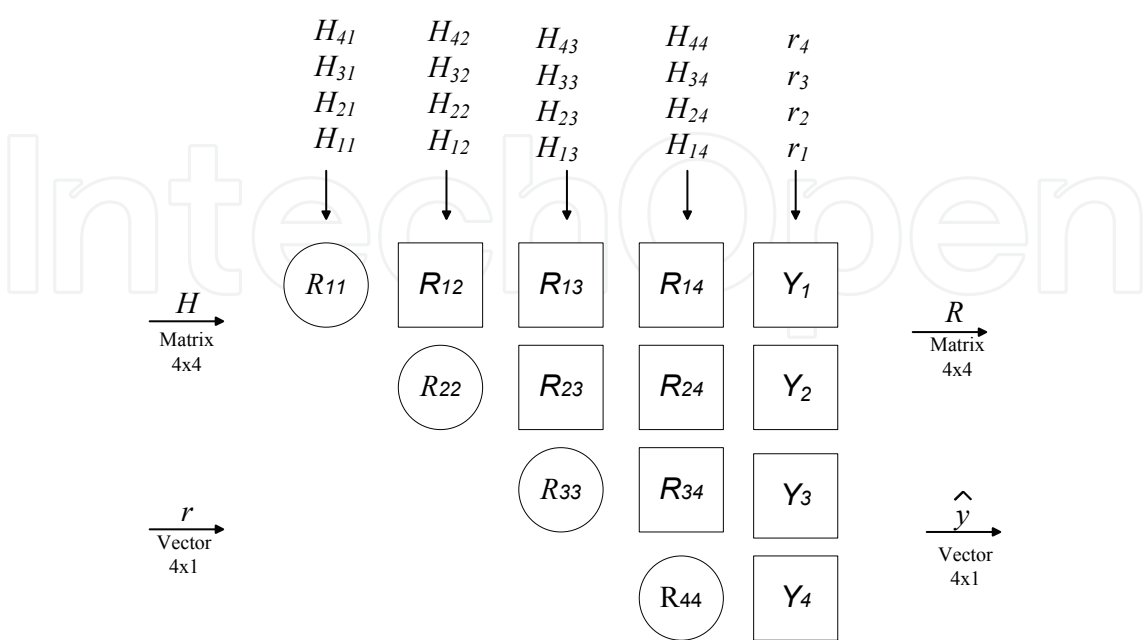


Fig. 3. QR Systolic Array for a 4x4 channel matrix

5. Architecture and FPGA implementation

In this section we consider the architecture and implementation of the FSD for a 4x4 uncoded MIMO system with 16-QAM modulation techniques. The FSD tree search has [16, 1, 1, 1] visited nodes per level as going from the root of the tree until the leaf respectively. Figure 4 shows the block diagram for the MIMO detector.

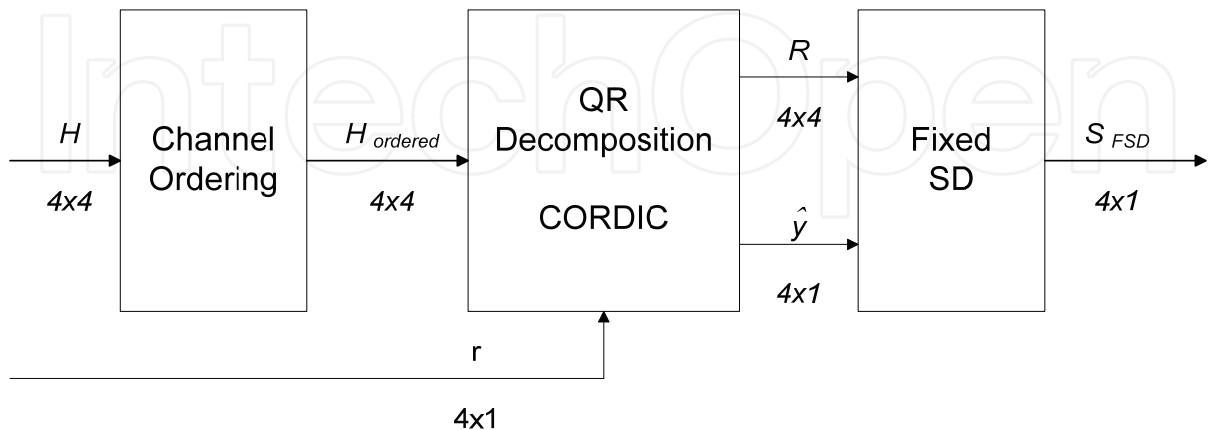


Fig. 4. FSD with QR and channel Ordering

The channel matrix H is assumed to be perfectly known at the receiver. The ordered channel matrix as well as the received vector is decomposed to upper triangular matrix R and a vector Q^Hr using the CORDIC QR decomposition.

5.1 Channel ordering

Channel ordering is performed using the columns of the channel matrix H . Channel columns are ordered in an iterative way according to the norm of each column. The received symbol with the largest noise amplification is selected for level N as this level considers all node symbols. For the remaining $N-1$ columns, we choose a channel column corresponding to the received symbols with the smallest noise amplification (Barbero & Thompson, 2006 b).

5.2 CORDIC systolic array

The basic CORDIC unit is implemented using only adder/subtract and shifter units. It can be adapted for vector mode and rotate mode by setting the sign of ρ_i .

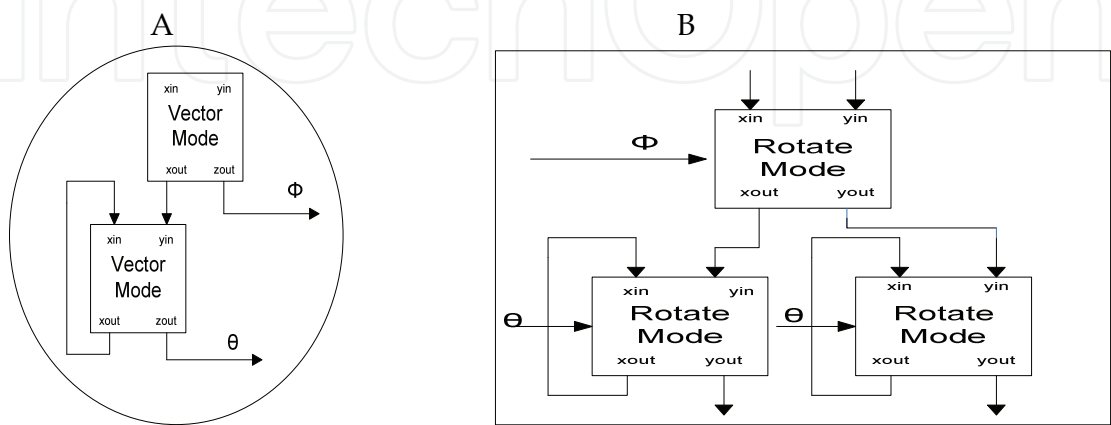


Fig. 5. (A) Vector mode CORDIC. (B) Rotate mode CORDIC.

For complex data, the boundary cell which performs vector mode consists of two real CORDIC blocks. The first one rotates the vector formed from the real and imaginary parts of the input, x , to set its imaginary part to zero producing an angle Φ . The second block computes the angle Θ needed to annihilate the rotated real part of the first CORDIC block against the stored one of the previous row.

The angle Φ calculated from the boundary cell is used to rotate both the real and imaginary part of the input vector. Both the rotated real and imaginary part, with the stored real and imaginary parts of the previous row, are rotated with angle Θ calculated from the boundary cell. On the other hand, the inner square cell which performs the complex rotate mode consists of three real CORDIC rotate blocks.

5.3 Fixed sphere decoder:

The search stages are composed of four pipelined levels and a decision unit. Considering 16-QAM modulation scheme, level 4 computes the 16 different Euclidian distances D_4 . At levels 3 to 1 only the symbol with minimum d_i is chosen. The calculation of the 16 minimum chosen branch distances (the minimum branch distance for each node) at each level from 3 to 1 can be done fully parallel or sequential (Hess et al., 2007) or in between (Barbero & Thompson, 2006 b). We chose to divide them into 2 iterations each calculates 8 different distances. This approach increases the throughput of the system as we can detect a new MIMO symbol every 2 clock cycles.

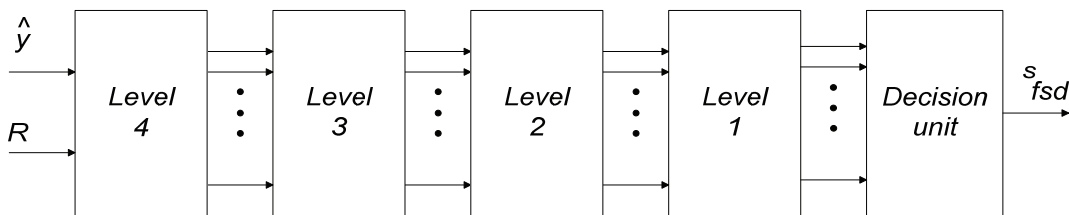


Fig. 6. Sphere Decoder Search stages

5.3.1 Level 4

Level 4 is the top level in tree after the root. At this level, the 16 different branch distances d_4 are calculated based on equation (7). The sphere ALU for the exhaustive search SD in (Barbero et al., 2006 b) which is used to calculate the branch distance is adapted and modified to calculate the Partial Euclidian distance of the FSD algorithm. For 16-QAM modulation, the partial Euclidian distance is calculated as follow

$$D_4 = d_4 = |b_4 - R_{44}s_4|^2 \tag{14a}$$

$$D_4 = R_{44}^2 \times |s_4|^2 + |b_4|^2 - 2 \times \text{Re} \{s_4^* \times b_4\} \tag{14b}$$

The term $|b_4|^2$ is added to all the 16 partial distances. So, removing this term won't affect the calculation since we are looking for the symbol with the minimum Euclidian distance. The term $R_{44}^2 |s_4|^2$ is equivalent to $2 |R_{44}|^2$ or $10 |R_{44}|^2$ or $18 |R_{44}|^2$ since we adopt 16-QAM modulation scheme. This calculation can be done using shift and add as shown in the architecture of Fig. 7. Registers are inserted to divide the operation into two clock cycles.

5.3.2 Levels 3 to 1

Each node in the tree has 16 branches. We need only to find the symbol that minimizes the distance d_i .

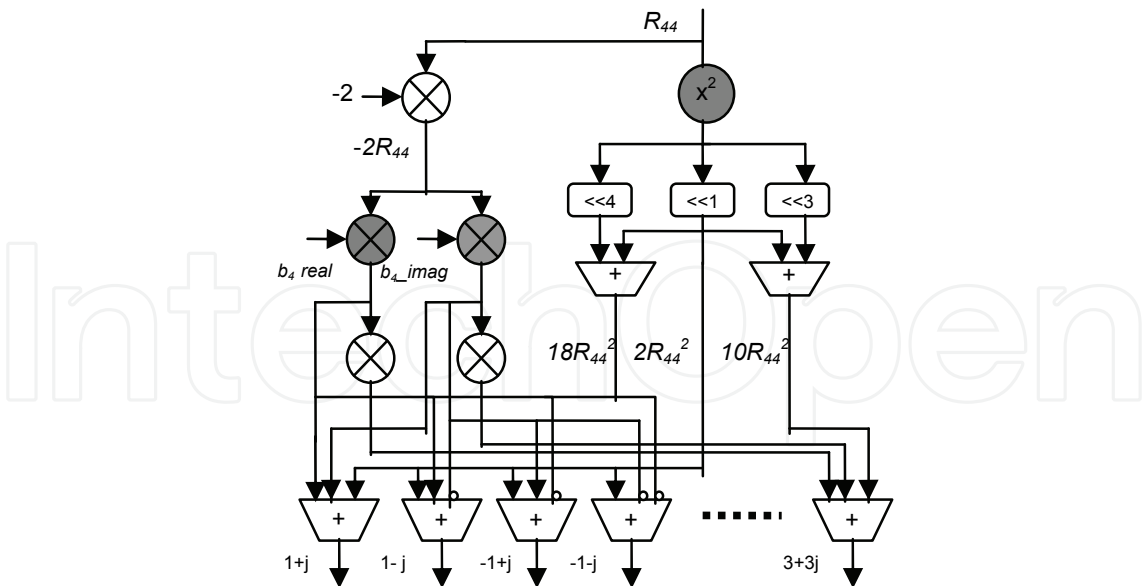


Fig. 7. Architecture of Level 4 FSD

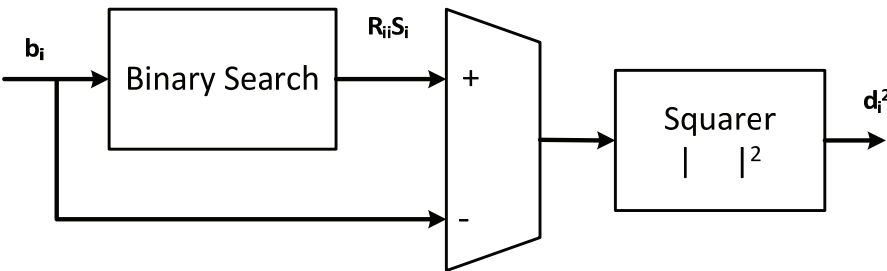


Fig. 8. Hardware Architecture for processing each node of Levels 3 to 1

$$d_i = |b_i - R_{ii}s_i|^2 \tag{15}$$

This is done by searching for the symbol s_i so that b_i is closest to $R_{ii}s_i$. This can be accomplished using binary search by comparing the real and imaginary of b_i with the boundaries of the used constellation.

The calculation of the distance d_i is performed in parallel for 8 branches out of 16. Therefore, 2 iterations are required to perform all distance calculation and only two clock cycles are required to perform the search of each stage. The calculation of b_i , using (8), requires the knowledge of the detected symbol of all the previous levels, $R_{ij}s_j$. Since we consider all the symbols in level 4, calculation of $R_{34}s_4$ can be done in parallel with level 4 as shown in figure 9. The 16 values of $R_{34}s_4$ are calculated over two clock cycles since we use two iterations over the parallel units for calculating the Euclidian distances.

For the case of b_2 and b_1 , the calculation of $R_{ij}s_j$ can't be applied in parallel with the previous levels as only one symbol is considered for the previous levels 3 and 2. However, if we consider equation (8) we can find that $R_{i4}s_4$ can be pre-calculated in the same manner shown in Fig. 9 which reduces the hardware area. Also, to reduce the hardware complexity, the different values of $R_{ij}s_j$ are pre-calculated, and then, using a multiplexer for each branch distance unit we choose the one corresponding to the symbol s_j . The architecture used for calculating $R_{ij}s_j$ is shown in fig. 10. We use shifters and adders instead of multipliers so the hardware complexity is significantly reduced.

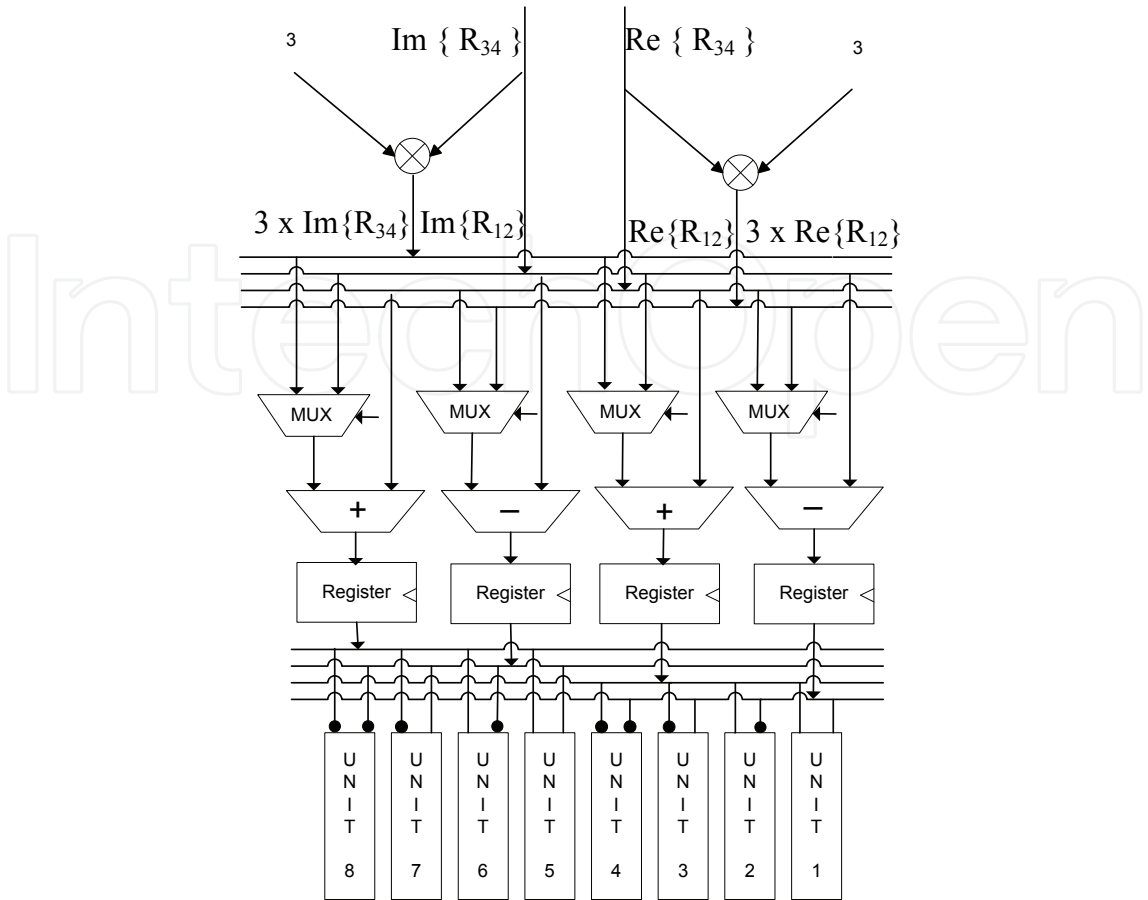


Fig. 9. Calculation of $R_{34}^* s_4$ for the 8 parallel units of Level 3

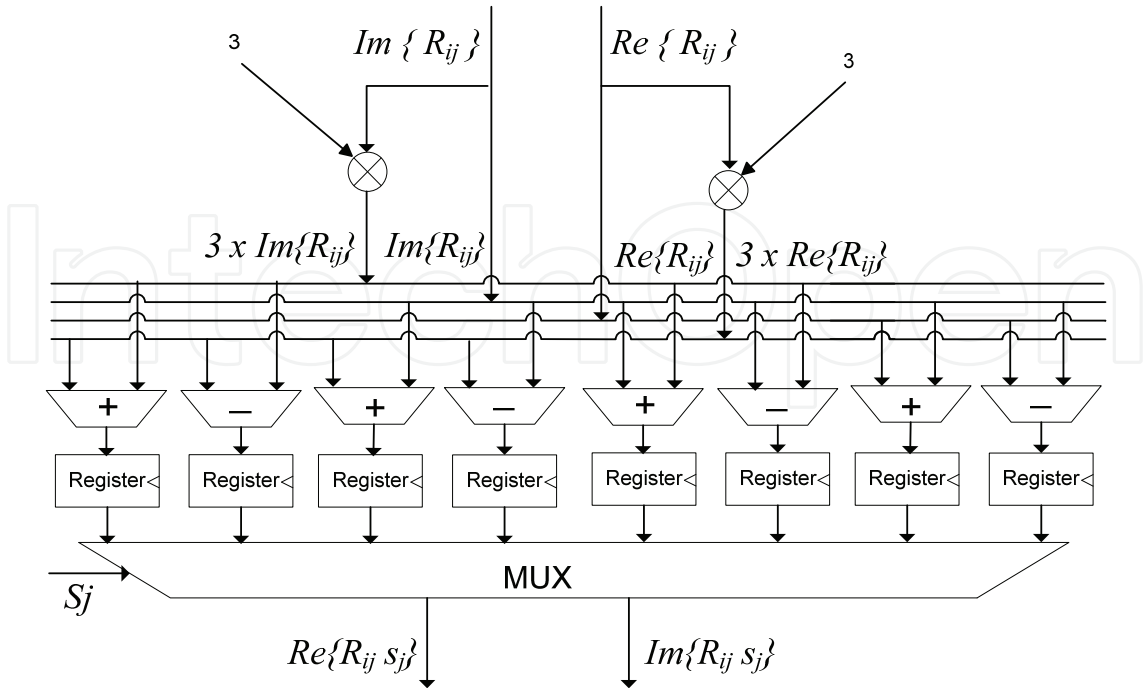


Fig. 10. Hardware Circuit for $R_{ij} s_j$ calculation

5.3.3 Decision unit

The decision unit adds the branch distances of level 4 to level 1 and then searches over the calculated 16 Euclidian distances to find the vector with the minimum distance. The transmitted vector with the minimum Euclidian distance is selected as the detected FSD symbol.

6. Symbol decoding

The WiMAX standard of the transmission scheme for two transmit and two receive antennas includes three matrices that define the transmission format with the row index indicating antenna number and column index indicating OFDMA symbol time (IEEE Standard 802.16e, 2005). One of the following three transmission matrices can be used.

6.1 Matrix A:

Matrix A achieves full diversity, and hence, improves the bit error rate given a rate one. This code is the Alamouti code.

$$A = \begin{bmatrix} s_i & -s_{i+1}^* \\ s_{i+1} & s_i^* \end{bmatrix} \quad (16)$$

The two received vector symbols at two successive time symbols can be written as:

$$\begin{bmatrix} r_0^{(0)} \\ r_1^{(0)} \end{bmatrix} = \begin{bmatrix} h_{00} & h_{01} \\ h_{10} & h_{11} \end{bmatrix} \times \begin{bmatrix} s_0 \\ s_1 \end{bmatrix}$$

$$\begin{bmatrix} r_0^{(1)} \\ r_1^{(1)} \end{bmatrix} = \begin{bmatrix} h_{00} & h_{01} \\ h_{10} & h_{11} \end{bmatrix} \times \begin{bmatrix} -s_1^* \\ s_0^* \end{bmatrix}$$

They can be combined into a matrix equation as follow:

$$\begin{bmatrix} r_0^{(0)} \\ r_1^{(0)} \\ r_0^{*(1)} \\ r_1^{*(1)} \end{bmatrix} = \begin{bmatrix} h_{00} & h_{01} \\ h_{10} & h_{11} \\ h_{01}^* & -h_{00}^* \\ h_{11}^* & -h_{10}^* \end{bmatrix} \times \begin{bmatrix} s_0 \\ s_1 \end{bmatrix} \quad (17)$$

resulting in a modified received vector $r'_{4 \times 1}$ and a channel matrix $H'_{4 \times 2}$. To use the sphere decoder, the channel matrix H must be first decomposed using the QR.

$$H'_{4 \times 2} = Q_{4 \times 4} R_{4 \times 2} = [Q'_{4 \times 2} \quad Q''_{4 \times 2}] \begin{bmatrix} R'_{2 \times 2} \\ O_{2 \times 2} \end{bmatrix} \quad (18)$$

The sphere decoder is searching for the vector s that minimizes the following equations:

$$\hat{s}_{\text{FSD}} = \arg \left\{ \min_{s \in S} \|r' - H's\|^2 \leq C^2 \right\} \quad (19)$$

$$\|r' - H's\|^2 = \|Q^H(r' - Hs)\|^2 = \left\| \begin{bmatrix} Q'^H_{2 \times 4} \\ Q''^H_{2 \times 4} \end{bmatrix} r' - \begin{bmatrix} R' \\ O \end{bmatrix} s \right\|^2 \quad (19a)$$

So,

$$\|r' - H's\|^2 = \|\hat{y} - R's\|^2 + \|Q_{2 \times 4}^{H'} \times r'\|^2 \tag{19b}$$

Since the second term doesn't depend on the symbol s , so equation 19 can be written as

$$\hat{s}_{FSD} = \arg\{\min_{s \in S} \|\hat{y} - R's\|^2 \leq C^2\} \tag{20}$$

where $\hat{y}_{2 \times 1} = Q_{2 \times 4}^{H'} \times r'_{4 \times 1}$
The Hardware architecture required to decode the symbols coded by Matrix A is shown Figure 11.

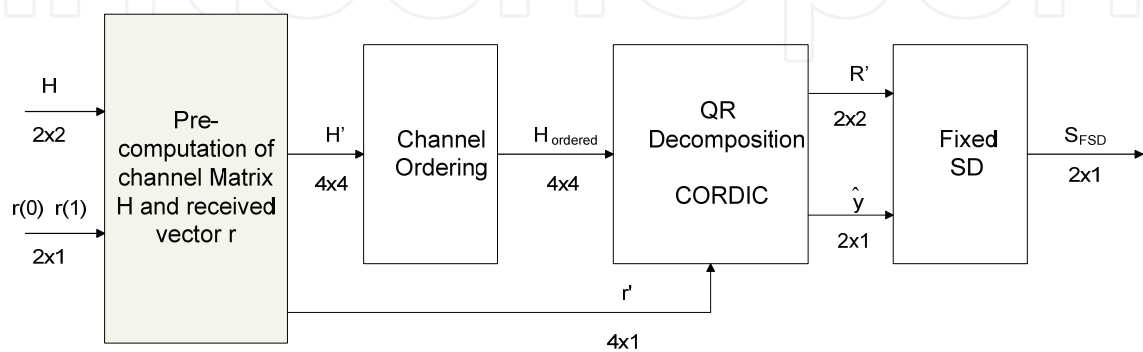


Fig. 11. Hardware Architecture for Matrix A decoding

The composed channel matrix H' is first ordered and then applied to the QR block. The CORDIC systolic array is fed by a non square matrix H' of dimension 4×2 . By using the given rotation technique, the upper triangular matrix R and the vector y are obtained. The R Matrix has a special feature of having the off diagonal element R_{12} set to zero at the last step of the decomposition. Thus, R is a diagonal matrix. So the decoding process is simplified since each symbol can be decoded independently. For each stage we are looking for the symbol s_i that minimizes equation 7. This is accomplished by searching inside the scaled constellation, shown in figure 12, for the symbol s so that $R_{ii}s_i$ is closest to b_i .

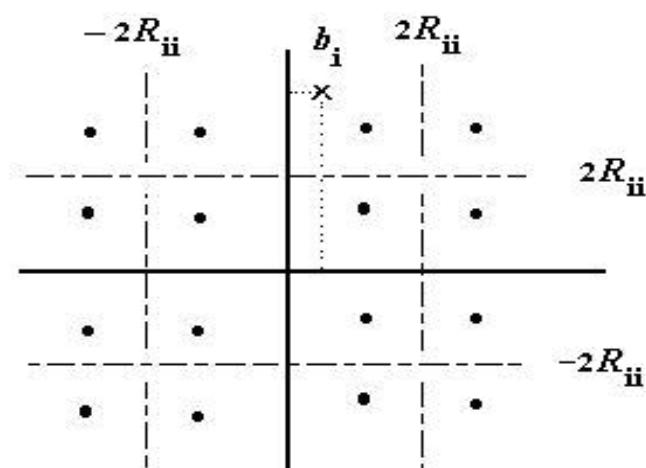


Fig. 12. 16-QAM Constellation

We compare the real and the imaginary value of b_i with the threshold levels 0 , $2R_{ii}$ and $-2R_{ii}$ that define the boundaries of each symbol. This comparison is done using comparators and multiplexers as shown in figure 13.

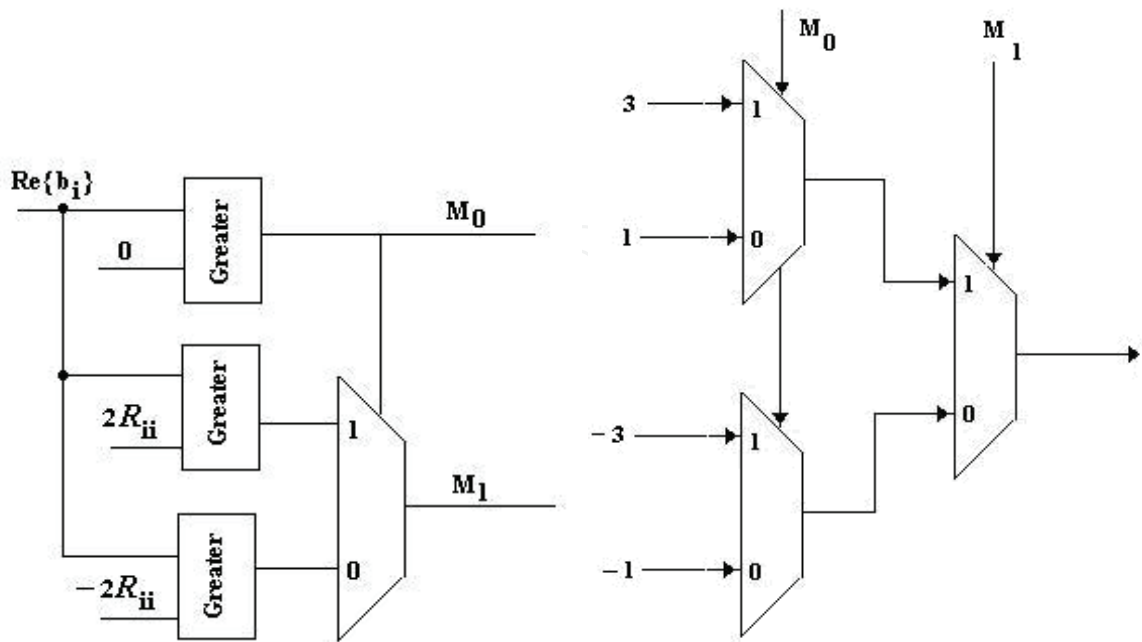


Fig. 13. Binary search Architecture

Therefore, there is no use for the tree search of the FSD. Only two parallel binary search blocks are required to find the transmitted symbols.

6.2 Matrix B:

Matrix B achieves spatial multiplexing with rate two employed for good channel conditions. It is similar to 2x2 un-coded MIMO system. Hence, it can be decoded directly using the traditional 2-level FSD. It doesn't require any additional adjustment over the usual FSD for the 2x2 MIMO systems.

$$B = \begin{bmatrix} s_i \\ s_{i+1} \end{bmatrix} \tag{21}$$

6.3 Matrix C:

Matrix C achieves full diversity and full rate. This code is called the golden code.

$$C = \frac{1}{\sqrt{1+p^2}} \times \begin{bmatrix} s_i + jp \times s_{i+3} & p \times s_{i+1} + s_{i+2} \\ s_{i+1} - p \times s_{i+2} & jp \times s_i + s_{i+3} \end{bmatrix}, p = \frac{-1+\sqrt{5}}{2} \tag{22}$$

The received vectors are:

$$\begin{bmatrix} r_0^{(0)} \\ r_1^{(0)} \end{bmatrix} = \begin{bmatrix} h_{00} & h_{01} \\ h_{10} & h_{11} \end{bmatrix} \times \begin{bmatrix} s_1 + jp \times s_4 \\ s_2 - p \times s_3 \end{bmatrix}$$
$$\begin{bmatrix} r_0^{(1)} \\ r_1^{(1)} \end{bmatrix} = \begin{bmatrix} h_{00} & h_{01} \\ h_{10} & h_{11} \end{bmatrix} \times \begin{bmatrix} p \times s_2 + s_3 \\ jp \times s_1 + s_4 \end{bmatrix}$$

They can be combined as:

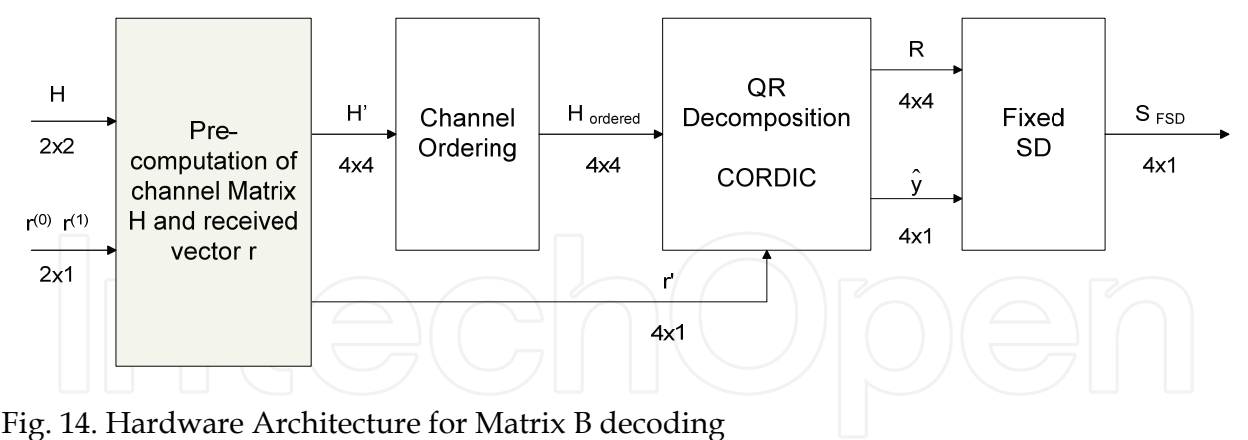


Fig. 14. Hardware Architecture for Matrix B decoding

$$\begin{bmatrix} r_0^{(0)} \\ r_1^{(0)} \\ r_0^{(1)} \\ r_1^{(1)} \end{bmatrix} = \begin{bmatrix} h_{00} & h_{01} & -ph_{10} & jph_{00} \\ h_{10} & h_{11} & -ph_{11} & jph_{10} \\ jph_{01} & ph_{00} & h_{00} & h_{01} \\ jph_{11} & ph_{10} & h_{10} & h_{11} \end{bmatrix} \times \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{bmatrix} \tag{23}$$

$$r_{4 \times 1}' = H_{4 \times 4}' \times s_{4 \times 1} \tag{24}$$

Although the system is 2x2 but we should use 4 Levels FSD to be able to decode the four transmitted symbols. Pre-computations of the channel matrix H and the received vector r have to be done to adjust them into a form suitable for the FSD as explained by equations (23-24). A block diagram describing the decoding process is shown in figure 14. The formed matrix H' and the vector r' are applied to the QR systolic structure to decompose them into the upper triangular matrix R and the vector y . The QR decomposition and FSD tree search is equivalent to the algorithms described in section 4 and 3 respectively.

7. Area report

The CORDIC algorithm is implemented with 9 iterations and 12 bit internal word length while a 16 bit word length is used for internal data width of FSD. ALTERA STRATIX II FPGA is used. The area report is given in Table I. Since, a new MIMO symbol is obtained every two clock cycles, the throughput for 4x4 system using 16-QAM modulation is calculated according to

$$\text{Throughput} = 4 \times 4 \times (\text{fclk}/2) \tag{25}$$

Operation over a frequency of 100 MHz results in a throughput of 800 Mbps. Compared to (Barbero & Thompson, 2006 b), we have achieved the double the throughput of the presented FSD-1 and 33% increase in the throughput compared to the optimized FSD2. Compared to the implementation of Ref. (Barbero & Thompson, 2006 b) given in Table I (using Vertex II Pro XC2VP70 FPGA), the number of multipliers are drastically reduced where only 2 multipliers (4 DSP Blocks) are required for the FSD and no multipliers are needed for the QR. Additionally, the proposed FSD requires less number of registers and Look-Up Tables LUT's.

	QR-CORDIC	4x4 FSD	FSD1/2[5]
Target FPGA	EP2S60F672C3	EP2S60F672C3	Xilinx XC2VP70
Combinational ALUTs/LUT	(5700) 11.78 %	(13,743) 28.2%	(16,119) 24%
Dedicated Logic Registers	(1520) 3.14%	(1412) 2.94%	15,332/16,855 23%/25.47%
DSP Block/Multiplier	-	4	(82) 25%
Memory bits/Block RAM	6080 < 1%	-	(82) 25%
Maximum clock freq.	160 MHz	102 MHz	100 MHz
Throughput	-	800 Mbps	400/600 Mbps

Table 1. Area Report and maximum clock frequency of 4x4 MIMO FSD

8. Simulation results

We developed a MATLAB Model for the FUSC WiMAX MIMO system with 2 transmitting antennas and 2 receiving antennas. A floating point simulation has been conducted for the QPSK modulation scheme. Fig. 15 shows bit error rate (BER) versus E_b/N_o for QPSK for the three different Matrices A, B and C. As seen, STC matrix A achieves the best BER but it only achieves data rate 1. STC matrix B has a data rate of 2 but its performance is about 6dB lower that Matrix A at 10^{-4} BER. Finally the STC matrix C has a full data rate of 2 and full diversity on the expense of higher bit error rate compared to the STC matrix A.

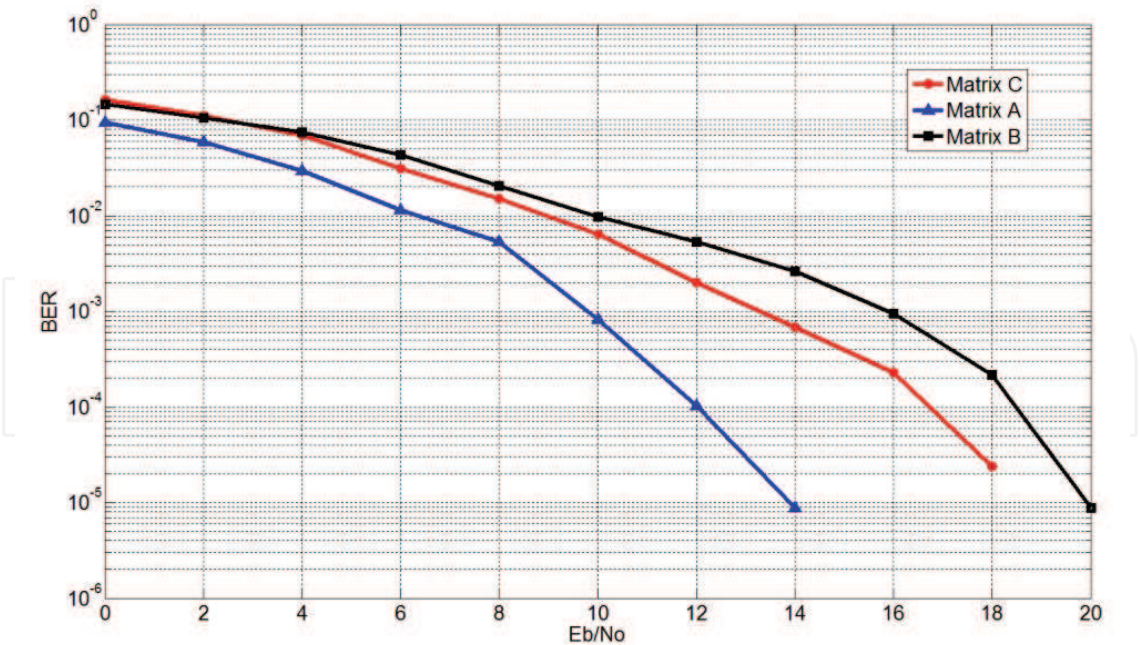


Fig. 15. Simulation Results Matrix A, B and C

Fig. 16 shows that the fixed-point BER performance of the fixed sphere decoder when employed to decode the symbols coded with Matrix A, B and C. The quantization effect has been simulated for 16, 14, 12 and 10 data bits.

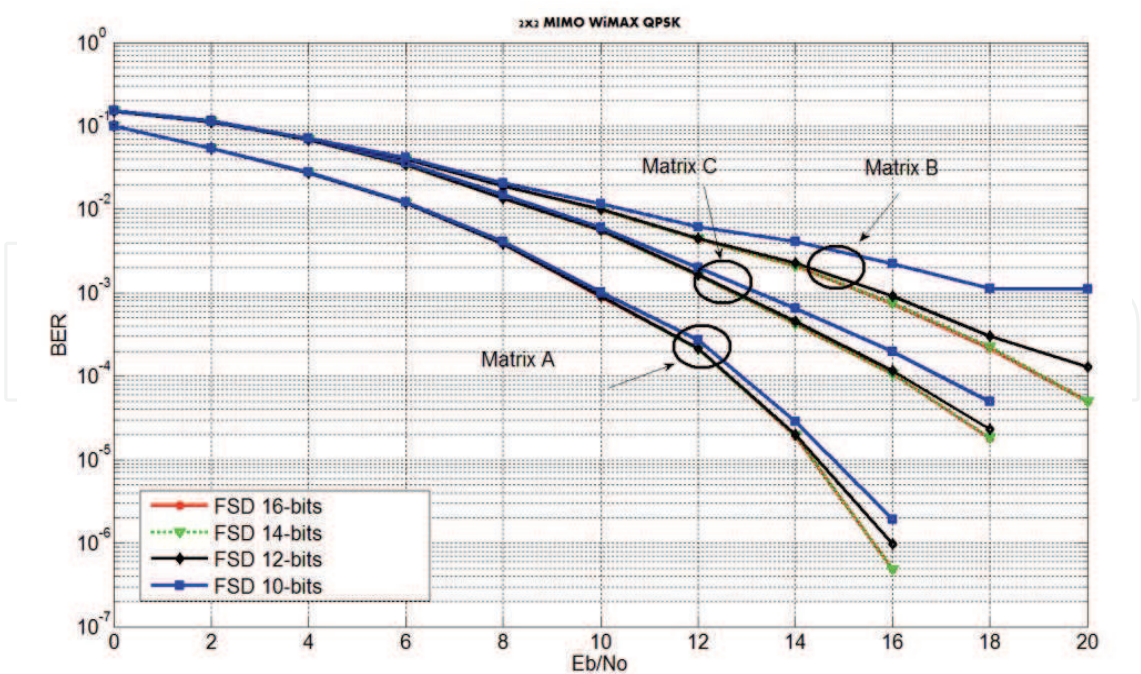


Fig. 16. Fixed Point Simulation of FSD for Matrix A, B and C

Finally the quantization effect of QR is taking into effect, where 12-bits per internal data word with 9 iterations have been used while a 16 bit data word is used for the FSD. Fig. 17 shows the degradation effect due to the quantization of the QR. This effect is apparent at high SNR where the noise effect is negligible compared to the quantization effect. Yet, this BER degradation at high SNR is less pronounced for the QR Decomposition than the same effect if Cholesky - based inversion of the H matrix is applied (Barbero & Thompson, 2006 a).

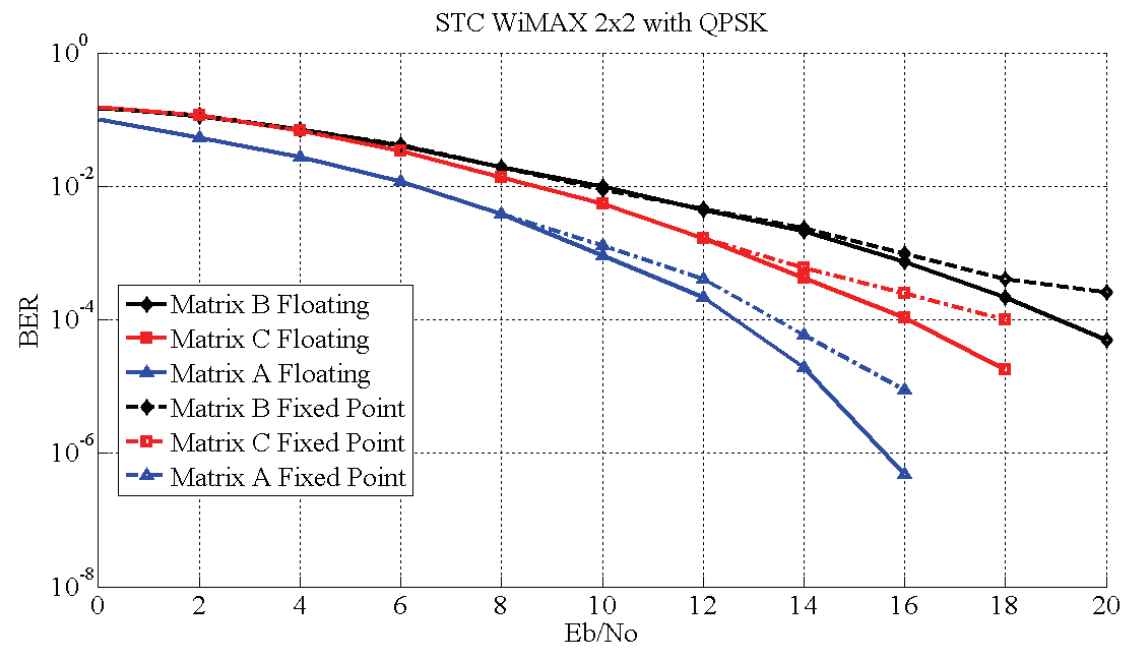


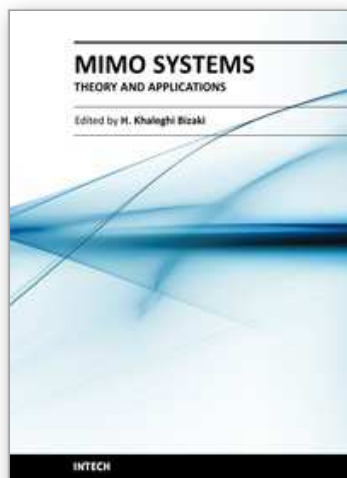
Fig. 17. QR quantization effect on Fixed Point Simulation

9. Conclusion

We have presented a FPGA architecture of the FSD as a MIMO decoder for the WiMAX system which achieves close-to ML BER performance. We have shown how the FSD is adaptively used to decode the three different STC. The presented FSD achieves fixed and higher throughput compared to (Barbero & Thompson, (2006) b) and uses less number of multipliers and LUTs.

10. References

- "Air Interface for Fixed and Mobile Broadband Wireless Access Systems", IEEE Standard 802.16e, 2005
- Barbero, L.G.; Thompson, J.S. (2005); "Rapid prototyping of the sphere decoder for MIMO systems," in *Proc. IEEE/EURASIP Conference on DSP Enabled Radio (DSPeR '05)*, vol. 1, Southampton, UK, Sept. 2005, pp. 41–47
- Barbero, L.G.; Thompson, J.S (2006) a, "Real Time implementation of a sphere decoder based MIMO Wireless system", in *EURASIP European Signal Processing Conference (EUSIPCO '06)*, Florence, Italy, Sep. 2006
- Barbero, L.G.; Thompson, J.S (2006) b, "Rapid Prototyping of a Fixed-Throughput Sphere Decoder for MIMO Systems," *IEEE International Conference on Communication 2006*, vol. 7, pp. 3082-3087, June 2006.
- 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 Circuits*, vol.40, no.7, pp. 1566- 1577, July 2005
- Guo, Z. and Nilsson, P. (2003), "A VLSI implementation of MIMO detection for future wireless communications," in *Proc. IEEE PIMRC'03*, vol. 3, pp. 2852–2856, 2003.
- Hess, C.; Wenk, M. ; Burg, A.; Luethi, P. ; Studer, C.; Felber, N. and Fichtner, N. (2007) "Reduced Complexity MIMO Detector with Close to ML Error Rate Performance ", *Proc. of ACM Great Lakes Symposium on VLSI (GLSVLSI)*, 200-203, Stresa-Lago Maggiore, Mar 2007
- Khairy, M.S.; Abdallah, M.M.; Habib, S.E.-D. (2009), "Efficient FPGA implementation of MIMO decoder for mobile WiMAX system," in *Proc. IEEE International Conference on Communications 2009 (ICC 2009)*, Dresden, Germany, June 2009.
- Khairy, M.S.; Mehlhauer, C.; Rupp, M. (2010), "Boosting sphere decoding speed through Graphic Processing Units," *16th European Wireless Conference (EW)*, Lucca, Italy, 12-15 April 2010.
- Myllyla, M.; Hintikka, J.-H.; Cavallaro, J.R.; Juntti, M.; Limingoj, M.; Byman, A (2005)., "Complexity Analysis of MMSE Detector Architecture for MIMO OFDM Systems," in *Proceedings of the 2005 Asilomar conference*, pp 75-81, Pacific Grove, CA, 2005.
- Proakis, J; Salehi, M (1994); "Communication Systems Engineering," Prentice Hall PTR
- Volder, J.E. (1959) "The CORDIC Trigonometric Computing Technique," *IRE Trans. on Electronic Computers*, vol. EC-8, no. 3, pp. 330–4, 1959.



MIMO Systems, Theory and Applications

Edited by Dr. Hossein Khaleghi Bizaki

ISBN 978-953-307-245-6

Hard cover, 488 pages

Publisher InTech

Published online 04, April, 2011

Published in print edition April, 2011

In recent years, it was realized that the MIMO communication systems seems to be inevitable in accelerated evolution of high data rates applications due to their potential to dramatically increase the spectral efficiency and simultaneously sending individual information to the corresponding users in wireless systems. This book, intends to provide highlights of the current research topics in the field of MIMO system, to offer a snapshot of the recent advances and major issues faced today by the researchers in the MIMO related areas. The book is written by specialists working in universities and research centers all over the world to cover the fundamental principles and main advanced topics on high data rates wireless communications systems over MIMO channels. Moreover, the book has the advantage of providing a collection of applications that are completely independent and self-contained; thus, the interested reader can choose any chapter and skip to another without losing continuity.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Muhammad S. Khairy, Mohamed M. Abdallah and S. E.-D. Habib (2011). Efficient Implementation of MIMO Decoders, MIMO Systems, Theory and Applications, Dr. Hossein Khaleghi Bizaki (Ed.), ISBN: 978-953-307-245-6, InTech, Available from: <http://www.intechopen.com/books/mimo-systems-theory-and-applications/efficient-implementation-of-mimo-decoders>

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2011 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](https://creativecommons.org/licenses/by-nc-sa/3.0/), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen