Available online at www.sciencedirect.com
ScienceDirect
ELSEVIER
C. R. Physique 7 (2006) 735–750
COMPTES RENDUS
PHYSIQUE
http://france.elsevier.com/direct/COMREN/

Towards reconfigurable and cognitive communications/Vers des communications reconfigurables et cognitives

Reconfigurable architecture for MIMO systems based on CORDIC operators

Hongzhi Wang, Pierre Leray, Jacques Palicot *

Supélec – campus de Rennes, avenue de la Boulaie, 35511 Cesson-Sévigné cedex, France

Available online 8 September 2006

Abstract

The MIMO system is an attractive technology for wireless 3G/4G systems. In this article we propose the realization on FPGA of a MIMO 'V-BLAST Square Root' algorithm based on a variable number of CORDIC operators. The CORDIC operator is highly suitable for this implementation as it only relies on simple techniques of addition and vector offsets. This square root algorithm architecture is reconfigurable in order to adapt itself to different numbers of antennas and different data rates. The proposed architecture can achieve a data rate of 600 Mbit/s in a Virtex-II FPGA circuit from Xilinx for the MIMO system with QPSK modulation. ***To cite this article: H. Wang et al., C. R. Physique 7 (2006).***

Résumé

Architecture reconfigurable pour les systèmes MIMO à base d'opérateurs CORDIC. Le système MIMO est une technologie attractive pour les systèmes 3G/4G sans fil. Dans cet article nous proposons l'implémentation sur FPGA d'un algorithme MIMO « V-BLAST Square Root » basée sur l'utilisation d'un nombre variable d'opérateurs CORDIC. L'opérateur CORDIC convient bien pour l'implémentions car il s'appuie seulement sur de simples techniques d'additions et de décalages entre vecteurs. Cette architecture de l'algorithme *square root* est reconfigurable pour s'adapter à différent nombre d'antennes et différent débit. L'architecture proposée peut atteindre un débit de 600 Mbit/s dans un circuit FPGA VirtexII de Xilinx pour le système MIMO avec une modulation QPSK. ***Pour citer cet article : H. Wang et al., C. R. Physique 7 (2006).***

Keywords: MIMO; V-BLAST Square Root; CORDIC; Configurable architecture

Mots-clés : MIMO ; V-BLAST Square Root ; CORDIC ; Architecture configurable

1. Introduction

With the integration of the Internet and new multimedia applications into wireless communications systems, the demand in terms of data rate is continually increasing. Several techniques have been developed in order to respond to this need. The MIMO (Multiple Input Multiple Output) technique, discovered in 1996 [1] by the researchers at Bell Labs, remains the most promising: it can increase spectral efficiency in a substantial manner. For example, the

* Corresponding author.
E-mail addresses: hongzhi.wang@supelec.fr (H. Wang), pierre.leray@supelec.fr (P. Leray), jacques.palicot@supelec.fr (J. Palicot).

1631-0705/$ – see front matter © 2006 Académie des sciences. Published by Elsevier Masson SAS. All rights reserved.
doi:10.1016/j.crhy.2006.07.018

standards for broadband wireless local networks, such as IEEE 802.11n and Hiperlan2, are going to adopt MIMO systems in their future standards. The data rate for HSDPA (High Speed Downlink Packet Access) transmissions can reach 21.6 Mb/s by using MIMO systems. This technique has received much interest in recent years and has given rise to a number of works. Amongst the difficulties engendered by this technique, the implementation of the MIMO signal demodulation algorithms is a topic currently under discussion. In order to be able to manage the multiplicity of communication standards, the MIMO system has to be capable of supporting various types of modulation and propagation. These are the reasons why a reconfigurable architecture is of considerable interest in MIMO systems.

The MIMO algorithms are generally implemented in a DSP (Digital Signal Processor), so it is difficult to achieve high data-rate performance. As for the traditional ASCI solution, this does not support the diversities of future systems. It is therefore natural to use programmable circuits of the FPGA type to respond to our problem of reconfiguration. These circuits are widely used in Signal Processing, thanks to the possibility of reconfiguration and the high level of parallelism. Concerning the implementation of MIMO reception algorithms, there is at present little published work on the subject.

An implementation of the 'Square Root' algorithm had been produced by Z. Guo in [2], but this realization does not meet the constraints of our problem. In fact, the realization described in [2] is not reconfigurable, as it is implanted on an ASIC.

It is to respond to this that we are proposing in this article an implementation based on CORDIC operators for a reprogrammable FPGA circuit.

The remainder of this article is structured as follows. Section 2 briefly presents the MIMO system. Section 3 describes the various detection algorithms and illustrates our choice of the 'V-BLAST Square Root' algorithm in terms of Performances/Complexity trade-offs. The next section details the 'V-BLAST Square Root' algorithm that we have adopted. Section 5 analyses the functional architecture of the algorithm, and in this section we also describe the CORDIC operator. Section 6 presents the reconfigurable architecture, we analyse the implementation through concrete examples, and then we present the results of the synthesis of the various architectures on an FPGA. Lastly, we highlight the dynamic (real-time) aspect of the reconfiguration.

2. Presentation of the MIMO system

The emergence of MIMO systems has been motivated by the increased data-rate requirements brought about by the arrival of new services such as Internet access and the transmission of images via wireless communications systems, as well as the saturation of resources in transmission channels, in particular in the mobile telephony band. MIMO systems consist of using several antennas for transmission and reception. By taking advantage of the associated space-time processing, these systems have shown a considerable increase in spectral efficiency (proportional to the number of antennas used) [3].

Starting from the viewpoint of the information theory, two researchers at the Bell laboratories, Foschini [4] and Teletar [5] have independently shown that the capacity of multi-antenna systems increases linearly with the number of antennas. These discoveries lie at the origin of MIMO systems (see Fig. 1), primarily aimed at resolving the problems of congestion and the capacity limitation of broadband wireless networks. The basic idea in MIMO systems is space-time processing, where time (a natural dimension) is complemented by a spatial dimension inherent in the use of several antennas. Such a system may be regarded as the extension of intelligent antennas. The key property of a MIMO system is its capacity to turn multipath propagation (traditionally a drawback) into an advantage—in other words, MIMO systems exploit multi-paths instead of suppressing them.

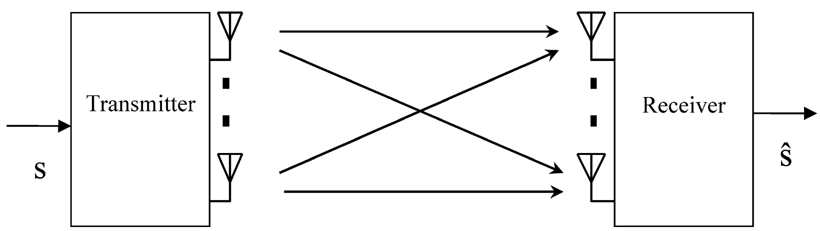


Fig. 1. Principle of a MIMO system.

Conventional techniques used on reception to cancel out the distortion introduced by the MIMO channel often necessitate either knowledge of the channel or the use of a known symbol sequence at the receiver end. In practice, the channel is unknown, and hence an estimation of it is needed. Often the estimation of the channel is based on the use of training sequences multiplexed with the useful data, which quite clearly reduces the useful date rate. For channels that do not change over time, the loss is not significant, as only a single learning cycle is needed.

A distinction can be made between two types of technique for transmission over MIMO systems: the first based on space-time codes, the other on spatial multiplexing [3,6].

2.1. MIMO system based on space–time codes

In order to improve transmission quality, Alamouti [7] and Tarokh [8] have designed systems based essentially on diversity, proposing linked coding and labelling. This *space–time coding* (STC) also allows more reliable communications; it consists of adding redundancy to the transmitted binary data in order to increase the diversity and avoid the drop outs characteristic of the MIMO channel. For more details about space-time codes, refer to [3,2,7,8].

2.2. MIMO system based on spatial multiplexing

Spatial multiplexing or V-BLAST (Vertical Bell Labs Layered Space–Time) [9,10] may be regarded as a special class of space-time codes, its principle consisting of dividing the input data stream into several sub-streams that are transmitted over different antennas. This architecture is principally aimed at increasing system capacity. A simple description is given in Fig. 1.

The transmission antennas each transmit a different symbol, independent of the other antennas, but using the same modulation and the same carrier frequency. The bandwidth used remains identical to that of a classic system, but as several symbols are transmitted, the spectral efficiency increases.

In the remainder of this article, we shall concern ourselves only with spatial multiplexing and more specifically the V-BLAST architecture. In this study, we are using a single-user model in a non-frequency-selective channel. In addition, the model has no memory. This is the most commonly used MIMO channel model, consisting of a matrix H in which each complex coefficient h_{ij} represents the transfer function between receiving antenna i^e and transmitting antenna j^e. For a system with M transmission antennas and N reception antennas, the received vector r may be written:

$$r = Hs + \nu, \quad s \in \xi^M \subset C^M \tag{1}$$

In this equation, $s = [s_1 \ldots s_M]^T$ is the transmitted symbols vector, $(.)^T$ denotes the transpose of $(.)$. H is the channel matrix with dimension $M \times N$ and $\nu = [\nu_1 \ldots \nu_N]^T$ is zero-mean additive Gaussian noise vector with $E[\nu\nu^*] = \sigma^2 I_N$. We are assuming that $E[ss^*] = I_M$ and $E[s\nu^*] = 0$. σ^2 is the variance of the noise per antenna.

The elements of H have a uniformly-distributed phase and amplitude that follows a Rayleigh law. This model is typical of an environment with numerous echoes and sufficient separation between the antennas. We are also assuming that the channel remains constant during the transmission of a block of L data vector $\boldsymbol{s}$ and that the receiver has full knowledge of the channel matrix H. This knowledge may be obtained either by training symbols or by a blind estimation of the channel.

3. Reception algorithms

Numerous reception algorithms exist that could be envisaged for recovering the symbols in MIMO systems. The least complex are the linear receivers based on the criterion of zero forcing (ZF) or minimization of the mean square error (MMSE). The reception algorithm proposed for the V-BLAST system [1] attempts to successively eliminate interference from transmitters other than that whose symbols are being estimated. They are sub-optimal in terms of BER. The optimal detector is based on maximum likelihood (ML), which demands a significant calculation load when the number of antennas or the size of the constellation is large. There exist numerous sub-optimal algorithms based on ML.

The study of new high-performance algorithms is not within the scope of this work. There exist very many algorithms, and it is not our aim to carry out exhaustive research of them all. We only wish in this section to identify, amongst the best-known algorithms, the best algorithms in terms of complexity/performance that we can implant on FPGAs.

3.1. Maximum Likelihood

The Maximum Likelihood (ML) receiver offers the best performance in term of Binary Error Rate (BER). In fact, it is optimal if the transmitted vectors s are equiprobable, which is the case since the symbols s_i are equiprobable, and the N ways transmitted in parallel are independent. It is expressed classically in the following fashion:

$$\hat{s} = \arg\min_{s \in \xi^M} \|r - Hs\|_2^2$$

However, its calculation load rapidly becomes very significant, as it increases exponentially with the number of transmitting antennas M and linearly with the number of receiving antennas N. The problem is known to be NP-complete, and no algorithm exists that makes it possible to find an optimal solution in polynomial time. It is therefore natural to look for simpler algorithms with similar performance. In this category we find the sphere decoding algorithm (SD) [11,12], as well as the 'Branch & Bound' algorithm [13,14].

We shall not list here all the sub-optimal receivers; there is copious literature on that subject. We are confining ourselves to just the best-known.

3.2. 'Branch & Bound'

The principle consists of constructing a tree, each branch of which becomes a sub-problem that is simpler to solve [13,14]. Whether a branch is kept in the algorithm or not is governed by 2 'bounds'. If the bound calculated in the branch under consideration is higher than that of the previously-conserved branch, then this branch is abandoned.

3.3. Sphere decoding

However, it is possible to approach maximum likelihood performance while keeping complexity reasonable [11,12]. The principle of this algorithm is to be positioned at the level of the received signal and to look for the nearest point on the lattice within a sphere of radius C. This greatly reduces the maximum likelihood search field, since only the network points located at a distance less than C from the received signal are considered during minimization of the metric. Hence the choice of C is crucial for the speed of convergence of the algorithm and for the accuracy of the results.

3.4. Linear Zero Forcing (ZF) receiver

This receiver is the simplest and also the poorest performer in term of BER. It seeks to cancel out the contributions of the other transmitters to each symbol. This comes down to inverting the channel transfer matrix:

$$s_{\mathrm{ZF}} = (H^* H)^{-1} H^* r$$

where $(.)^*$ denotes the Hermitian of (.) and $(.)^{-1}$ denotes the inverse of (.). When H is poorly conditioned, inverting it multiplies the noise and thereby seriously degrades the performance at low SNRs.

3.5. Linear MMSE receiver

This criterion minimizes the mean square error due at the same time to the noise and the interference between symbols, unlike the ZF receiver that only deals with the interference between symbols. Its expression is:

$$s_{\mathrm{MMSE}} = (H^* H + \sigma^2 I)^{-1} H^* r$$

This receiver is more resistant to noise than the ZF receiver. At high SNRs, the MMSE receiver tends towards the ZF receiver, since σ^2 tends to 0.

3.6. V-BLAST decisional feedback receiver

The principle of the V-BLAST receiver algorithm was presented in [9,1]. It involves a decisional feedback equalizer, adapted to the structure of MIMO systems. The equalizer can use either the zero forcing criterion or the MMSE criterion. Its principle is as follows: the symbol from the most favoured transmitter (having the best BER according to the criterion considered) is demodulated first. Its contribution to the received vector $\boldsymbol{r}$ is then cancelled out, which increases the SNR on the other transmitters (at each correct decision). This step is repeated until the last transmitter, the least favoured. This receiver is also called OSIC (Ordered Successive Interference Cancellation) in the literature. As for all decisional feedback equalizers, its main drawback is the propagation of errors. Once a wrong decision has been taken over the value of a symbol, a wrong contribution is deducted from the vector $\boldsymbol{r}$, which means that the following symbols will almost certainly be estimated incorrectly. This is why sequencing is used to minimize error propagation.

Many authors have proposed improvements in the algorithm, some based on an iterative QR decomposition like M.O. Damen et al. [11]. Others, like W. Zha and S.D. Blostein [15], use substitution methods. However, the BER is degraded.

3.7. 'V-BLAST Square Root' algorithm

In 2000, B. Hassibi presented the 'Square Root' algorithm [16] which makes it possible to reduce the complexity from $O(M^4)$ to $O(M^3)$ without degrading the BER, in the case where $M = N$. Of the various MIMO signal detection algorithms, the 'V-BLAST Square Root' algorithm achieves a good compromise between the expected performance and low algorithm complexity.

3.8. Succinct performance/complexity comparison

In this section we have presented the various classic detection algorithms such as the linear and non-linear structures or the maximum likelihood structures. Fig. 2 shows an assessment of the BER for the receivers we have just presented. These results are obtained in a MIMO system with 2×2 antennas using QPSK modulation.

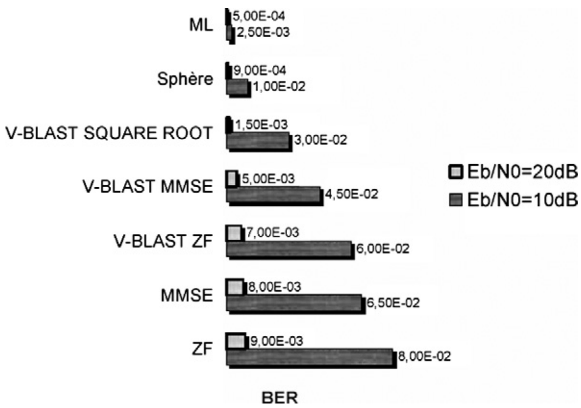


Fig. 2. Comparison of BER performances.

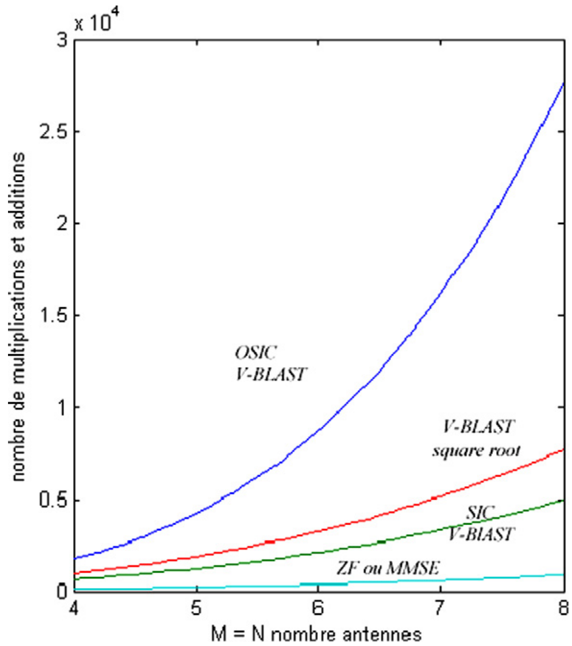


Fig. 3. Comparison of complexity performance as a function of the number of antennas.

Table 1
Comparison of complexity for the various algorithms

Algorithms	Number of operations/symbol	4 × 4 antennas
ZF	$5N^3/3 - 2N/3$	103
MMSE	$3MN^2 + N^3/3 + MN/2 - N^2/2$	107
V-BLAST SIC	$2M^2N + 2MN^2 + 11M^3$	960
V-BLAST OSIC	$M^2N^2 + 2NM^3 + 15/4M^4$	1728
V-BLAST Square Root	$5/3M^3 + 8M^2N + 2MN^2 + 2MN - M^2$	763

Table 1 presents the complexity in terms of number of operations for a 4 × 4 MIMO system.

In Fig. 3 we have shown this same complexity of implementation expressed in number of operations (multiplications and additions) per symbol as a function of the number of antennas.

It is important to note that for the maximum likelihood algorithm, the size of the constellation plays an important part, because the receiver has to calculate the distances for a set of possible vectors within the constellation. The calculation time soon becomes excessive for constellations of large size, even for a small number of antennas. For example, in Table 1, a QAM 16 system with 4 antennas gives us 16^4 possibilities, i.e. 65 536. Hence this algorithm is not adopted. The performance of the linear-type architectures seems correct, but insufficient with respect to non-linear structures.

Considering the figures of Table 1, we have chosen the 'Square Root' algorithm as one of the best Complexity/performance compromises. In order to properly understand the implantation on FPGAs that we are proposing, we will first describe this algorithm in detail in the next section.

4. Description of the 'V-BLAST Square Root' algorithm

4.1. Principle of the V-BLAST receiver

The principle of the V-BLAST receiver algorithm has been presented in [1]. It involves a decisional feedback equalizer, adapted to the structure of MIMO systems. The equalizer can use either the zero forcing criterion or the MMSE criterion. Its principle is as follows: the symbol from the most favoured transmitter (having the best BER according to the criterion considered) is demodulated first. Its contribution to the received vector $\boldsymbol{r}$ is then cancelled out, which increases the SNR on the other transmitters (at each correct decision). This step is repeated until the last transmitter, the least favoured.

4.2. 'V-BLAST Square Root' algorithm

The 'V-BLAST Square Root' algorithm was proposed by B. Hassibi [16]; it avoids the repeated calculation of the pseudo-inverse of the channel matrix and the matrix inversion by using unitary transformations. It makes it possible to reduce the calculation load from O(M^4) to O(M^3) without degrading the BER. To do this, B. Hassibi uses a recurrence relationship well known in adaptive *RLS* (Recursive Least Square) filtering. He demonstrates that if one applies a Givens rotation sequence to the recurrence relationship, one obtains $P^{1/2}$ after i iterations. However, it still remains to calculate Q_α. The best solution will be to apply a relationship that provides $P^{1/2}$ and Q_α at the same time. This is why B. Hassibi put forward a new recurrence relationship starting from the preceding matrix block, to which he adds a block vector.

The algorithm is summarized below.

In order to inversing H, the algorithm begins with the QR decomposition of the augmented channel matrix.

$$\begin{bmatrix} H^{N\times M} \\ \sqrt{\sigma} I^{M\times M} \end{bmatrix} = QR = \begin{bmatrix} Q_\alpha^{N\times M} \\ \times \end{bmatrix} R^{M\times M}$$

The covariance matrix is defined as $P = H_\alpha^+ = (\sigma^2 I + H^* H)^{-1}$, then we can find $R^{-1} = P^{1/2}$ and $H_\alpha^+ = P^{1/2} Q_\alpha^*$, where $P = P^{1/2} P^{*/2}$. Thus the pseudo inverse and covariance matrix can be computed by giving $P^{1/2}$ and Q_α.

The following algorithm is summarized below:

Step 1: Calculation of $P^{1/2}$ and Q_α.

– Initialization

$$\begin{bmatrix} 1 & H_i^{1\times M} P_{i-1}^{1/2 M\times M} \\ 0^{M\times 1} & P_0^{1/2 M\times M} \\ -e_i^{N\times 1} & Q_0^{N\times M} \end{bmatrix} \quad \text{with } P_0^{1/2^{M\times M}} = \beta I,\ Q_0 = 0^{N\times M},\ \beta = \frac{1}{\sqrt{\sigma^2}}$$

– Iteration (A): for $i = 1$ to N

$$\begin{bmatrix} 1 & H_i^{1\times M} P_{i-1}^{1/2 M\times M} \\ 0^{M\times 1} & P_{i-1}^{1/2 M\times M} \\ -e_i^{N\times 1} & Q_{i-1}^{N\times M} \end{bmatrix} \Theta_i = \begin{bmatrix} \times & 0^{1\times M} \\ \times & P_i^{1/2 M\times M} \\ \times & Q_i^{N\times M} \end{bmatrix} \tag{2}$$

End

At the end of these N iterations one obtains $P_N^{1/2} = P^{1/2}$, $Q_N = Q_\alpha$,

- Q_i representing the ith iteration with Q_0 initialized to 0,
- e_i indicates the ith column of the identity matrix,
- Θ_i corresponds to a unitary transformation, which transforms the matrix of Eq. (2) into a lower triangular matrix. The methods for finding this type of unitary transformation are well known [16].

Iteration (B): For $i = 0$ to $M - 1$

Step 2: Determine the minimum norm of the lines of $P^{1/2}$ and permutate this line so that it becomes the last one. This step allows to determine the optimal detecting order to obtain the strongest transmit signal. Likewise permutate the index of the received symbol correspondingly. Perform a unitary transformation that satisfies relationship (3).

$$P^{1/2}\Sigma = \begin{bmatrix} P_{i-1}^{1/2} & P_i^{i-1\times 1} \\ 0 & p_i \end{bmatrix} \tag{3}$$

As for Eq. (2), Σ is a unitary transformation that transforms $P^{1/2}$ into an upper triangular matrix.

Step 3: Update Q_α on the basis of $Q_\alpha \Sigma$.

Step 4: Calculate the MMSE *nulling vectors*

$$w_i = p_i q_{\alpha,i}^* \tag{4}$$

Step 5: Calculate the strongest transmit signal

$$\tilde{y}_i = w_i r_i \tag{5}$$

Step 6: Slice $\hat{y}_i$ to the nearest value in the signal constellation

$$\hat{s}_i = \text{decision}(\tilde{y}_i) \tag{6}$$

Step 7: Cancel the interference of the sliced strongest transmit signal from the vector of received signals and return to step 2.

$$r_{i-1} = r_i - h_i \hat{s}_i \tag{7}$$

End

5. Functional description of the 'V-BLAST Square Root' algorithm

The architecture of the "V-BLAST square-root" MIMO detector is illustrated in Fig. 4. It consists of 6 processing modules. The inputs consist of received messages r and the values of the channel matrix H. The first three modules (M_1, M_2, M_3) perform the decomposition of the matrix H using unitary transformations. These modules calculate the dimensions $P^{1/2}$ (Step 1), Q_a (Step 1), p_i (Step 2) and $q*_{a,i}$ (Step 3). The next module M_4 determines the optimal decoding order and calculates the vector w_i (*nulling vector*) (Step 4). Module M_5 decides the transmitted symbol vector (Step 6) and the last module M_6 performs interference cancellation between the symbols (Step 7).

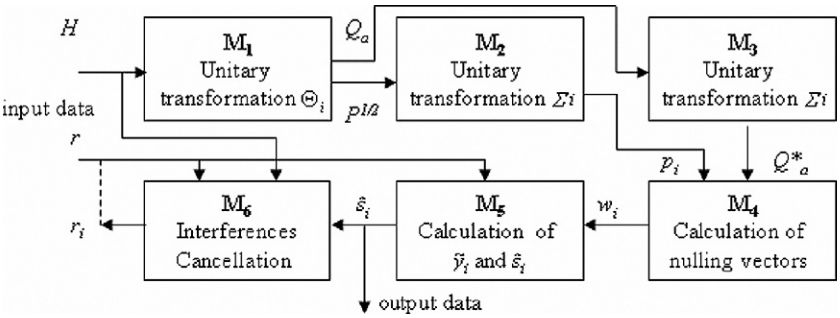


Fig. 4. Functional architecture of the "V-Blast *Square-Root*" MIMO decoder.

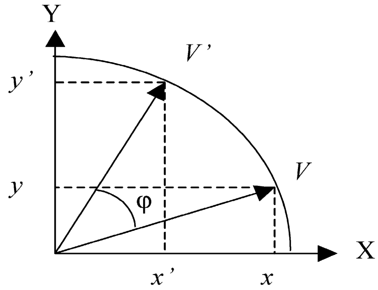


Fig. 5. Rotation of vector V in the Cartesian plane.

The three modules (M_1, M_2, M_3) exhibit similar architecture. These modules are designed using CORDIC operators (see example below for the calculation of $P^{1/2}$ and Q_α in module M_1). Instead of performing QR decomposition by a triangular network, we use a CORDIC-based Givens rotation sequence.

The last three modules (M_4, M_5, M_6) are based on elementary processors (PEs) consisting of a multiplier-accumulator, a subtractor, and a buffer.

5.1. CORDIC operator

The Givens rotation sequences can be implemented on an architecture consisting of a single processor. This architecture based on the CORDIC (COordinate Rotation Digital Computing) operator is proposed by Rader [17]. It's a simple and effective method for calculating a range of complex functions, which relies on a technique of additions and shifters. The algorithm calculates most trigonometric-based functions by approximation. It performs rotations without using multiplication operations.

5.1.1. Principle of the CORDIC algorithm

Resolving functions like sine and cosine by the CORDIC algorithm relies on a method of vector rotation in the Cartesian plane. Let's assume rotation of the vector $V(x, y)$ by an angle ϕ as shown in Fig. 5.

The coordinates of the vector V' are expressed by the equations:

$$x' = x\cos(\phi) - y\sin(\phi)$$
$$y' = y\cos(\phi) + x\sin(\phi)$$

If the angle of rotation is restricted to $\tan^{-1}(\pm 2^{-i})$ where $i = 0, 1, 2, 3, \ldots$, we then obtain ϕ by a series of successive elementary rotations of the order of:

$$a_{i+1} = a_i - d_i \cdot \tan^{-1}(2^{-i}) \quad \text{where } d_i = \pm 1$$

The index d_i indicates the sense of the angle's rotation for each iteration. This index is determined at each iteration according to the result of a comparison. The way this index is used will be explained a little further on. Each iterative vector $V_{i+1}(x_{i+1}, y_{i+1})$ is represented by:

$$x_{i+1} = K_i[x_i - y_i \cdot d_i \cdot 2^{-i}]$$
$$y_{i+1} = K_i[y_i + x_i \times d_i \times 2^{-i}] \quad \text{where } K_i = \cos(\tan^{-1} 2^{-i}) = (1 + 2^{-2i})^{-1/2}$$

It can be shown that, for an increasing number of iterations, the product of the factors K_i tends towards a constant equal to 0.6073. Hence the gain of the rotation algorithm is approximately 1.647. The exact gain is a function of the number of iterations n which may be expressed by:

$$An = \prod_n (1 + 2^{-2i})^{1/2}$$

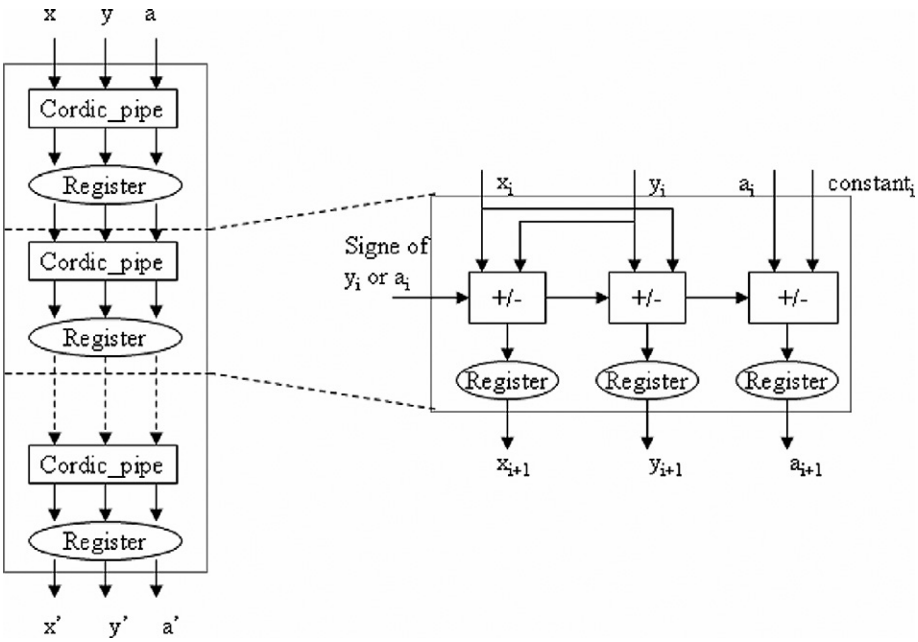


Fig. 6. CORDIC operator pipeline architecture.

Since for a relatively high number of iterations the product tends towards a constant result, we are able to apply this later in the algorithm. Thus we obtain a collection of simplified equations specific to the calculation of the mathematical operations we are looking for:

$$\begin{aligned} x_{i+1} &= x_i - d_i \cdot \mathrm{d}y_i \quad \text{where } \mathrm{d}y_i = y_i \times 2^{-i} \\ y_{i+1} &= y_i + d_i \cdot \mathrm{d}x_i \qquad\qquad \mathrm{d}x_i = x_i \times 2^{-i} \\ a_{i+1} &= a_i - d_i \cdot \mathrm{d}a_i \qquad\qquad \mathrm{d}a_i = \tan^{-1}(2^{-i}) \end{aligned}$$

$d_i = \pm 1$ according to the sign of a_i or y_i.

So these equations serve in calculating the operations. The general principle of the CORDIC algorithm consists of making the rotation vector turn in the appropriate direction by an increasingly small angle until the angle a or the values x and y are approximately equal to 0.

The CORDIC method can be employed in two different modes, known as the 'rotation' mode and the 'vector' mode. In the rotation mode, the co-ordinate components of a vector and an angle of rotation are given and the co-ordinate components of the original vector, after rotation through a given angle, are computed. In the vector mode, the co-ordinate components of a vector are given and the magnitude and angular argument of the original vector are computed.

5.1.2. CORDIC operator architecture

The iterative structure of the CORDIC algorithm makes possible an implantation using a pipeline structure (Fig. 6), thus limiting critical path length in such a way as to speed up operation.

5.2. CORDIC parallel implementation (example: calculating $P^{1/2}$ and Q_α)

Calculation of $P^{1/2}$ and Q_α takes place in the first module. This is the first step in decomposing the channel matrix H. For reasons of numerical stability and ease of implementation, we use a Givens rotation sequence based on CORDIC operators [17].

Let us now consider in the rest of this article a 2 × 2 antenna MIMO model. Then the first term in the second column of Eq. (2) $H_i^{1\times M} P_{i-1}^{1/2M\times M}$ is given below:

$$H_i^{1\times M} P_{i-1}^{1/2M\times M} = \begin{bmatrix} HP_{11} & HP_{12} \\ HP_{21} & HP_{22} \end{bmatrix}$$

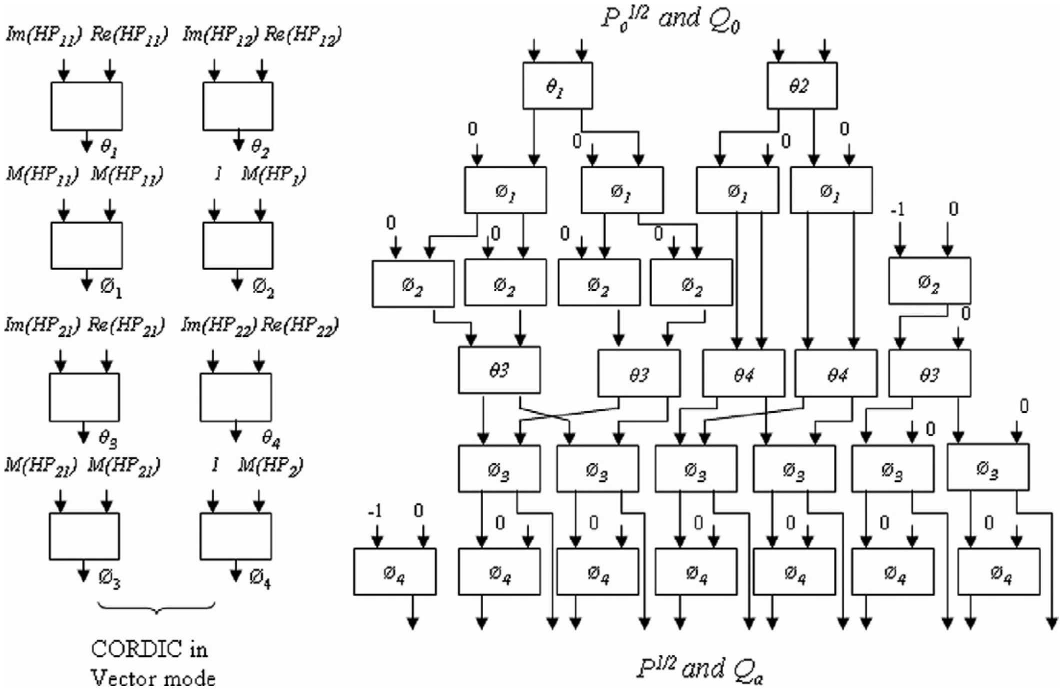


Fig. 7. 29 CORDIC operators for calculating $P^{1/2}$ and Q_a.

The Givens rotation angles are defined from this matrix as below:

$$\theta_1 = \tan^{-1}\frac{\mathrm{Im}(HP_{11})}{\mathrm{Re}(HP_{11})}, \qquad \theta_2 = \tan^{-1}\frac{\mathrm{Im}(HP_{12})}{\mathrm{Re}(HP_{12})}$$

$$\text{Ø}_1 = \tan^{-1}\mathrm{Module}(HP_{11}), \qquad \text{Ø}_2 = \tan^{-1}\frac{\mathrm{Module}(HP_{12})}{\mathrm{Module}(HP_1)},$$

$$\text{with } \mathrm{Module}(HP_1) = \sqrt{\mathrm{Module}(HP_{11})^2 + 1}$$

$$\theta_3 = \tan^{-1}\frac{\mathrm{Im}(HP_{21})}{\mathrm{Re}(HP_{21})}, \theta_4 = \tan^{-1}\frac{\mathrm{Im}(HP_{22})}{\mathrm{Re}(HP_{22})}$$

$$\text{Ø}_3 = \tan^{-1}\mathrm{Module}(HP_{21}), \qquad \text{Ø}_4 = \tan^{-1}\frac{\mathrm{Module}(HP_{22})}{\mathrm{Module}(HP_2)}$$

$$\text{with } \mathrm{Module}(HP_2) = \sqrt{\mathrm{Module}(HP_{21})^2 + 1}$$

Once the angles have been pre-calculated by 8 CORDIC operators in vector mode, we can perform the unitary transformations using the CORDIC operators in rotation mode. Calculating Eq. (2) can then be decomposed into 29 rotations.

Fig. 7 illustrates a fully parallel implementation of the calculations using 29 CORDIC operators. The parallel structure is designed to obtain a very high data rate, but is costly in terms of hardware resources. For lower data-rate applications, not all of the available calculation power is used. To tend towards optimal implementation of the MIMO decoder, we are proposing a reconfigurable architecture whose structure is a function of the transmission characteristics.

6. Reconfigurable architecture

We propose a processing structure where the number of CORDIC operators adapts itself to the data rate and number of antennas characteristics for optimal implementation. A fully parallel architecture, of course, offers significant calculation capacity, but at the expense of very great complexity. On the other hand, an iterative structure constructed around a single operator greatly reduces the complexity, but limits processing speed. It is by combining both these

Table 2
Notation used in Eqs. (8)–(10)

Freq	Clock frequency determined by the calculation speed of a CORDIC operator
NC_{used}	Number of CORDIC operators used
$NC_{optimal}$	Optimal number of CORDIC operators
NC_{total}	Total number of CORDIC operators (depending on M and N)
F_s	Symbol frequency
$N_{symbols/H}$	Number of symbols per frame
$Rate_{symbol}$	Transmitted symbol rate

approaches that it will be possible to obtain an optimal architecture, being guided by the relationships that must link the characteristics of the communication chain (data rate, number of antennas) and the detector architecture (number of CORDIC operators and the way they are organized).

6.1. Relationships

First, we have established a relationship between the data rate and the number of parallel CORDIC operators at the receiver. They are established for a MIMO system with M transmitting antennas and N receiving antennas (see Table 2 for symbols used). The processing data rate is expressed by Eq. (8).

$$\text{Data rate} = (\text{Freq} \times N) \times (\text{NC}_{\text{used}}/\text{NC}_{\text{total}}) \tag{8}$$

In this equation, NC_{total} represents the total number of CORDIC operators to be used for a given MIMO system. This is obtained as a function of the number of transmitting and receiving antennas. Freq is the calculation rate of a CORDIC operator.

Eq. (8) can be re-written as follows:

$$\text{NC}_{\text{used}} = (\text{Data rate} \times \text{NC}_{\text{total}})/(\text{Freq} \times N) \tag{9}$$

Then we have established the relationship with the transmitter. In the communication chain, the data rate of Eq. (9) corresponds to the transmitted symbol rate ($Rate_{symbol}$). We assume that the channel remains stationary during a frame period that is expressed by the number of symbols per frame ($N_{symbols/H}$). Under these conditions, we can express the optimum number of CORDICs as being the minimum number of CORDICs to be used in order for the transmitted symbol rate to be equal to the reception processing rate (cf. Eq. (10)):

$$\text{NC}_{\text{optimal}} = (M/N) \times (\text{NC}_{\text{total}} \times F_s)/(\text{Freq} \times N_{\text{symbols/H}}) \quad \text{with } F_s = \text{Rate}_{\text{symbol}}/M \tag{10}$$

In Eq. (10), when the channel remains stationary during one symbol ($N_{symbols/H} = 1$), the number of CORDIC operators to be used to adapt to Data rate is $NC_{optimal} = NC_{total}$ (under the condition of $M = N$, $\text{Freq} = F_s$).

The relationship between the number of CORDIC operators and the data rate is illustrated in Fig. 8. The other factors in Eq. (10) are considered as constants within a single MIMO system. The minimum number of CORDIC operators to achieve the desired data rate is $NC_{optimal}$. If the number of CORDIC operators is reduced, the data rate will fall linearly. On the other hand, in Fig. 8 the data rate does not increase linearly with the number of operators. It remains constant as shown, because the system is calculating the same channel matrix during the stationary period.

The chronogram of data flow is shown in Fig. 9. As will be presented in Figs. 12 and 13, modules M_1, M_2, M_3 use CORDIC operators and M_4, M_5, M_6 use PEs (Processing Element). These computations are performed in parallel. When modules M_1, M_2, M_3 perform the computation of channel matrix H_1, modules M_4, M_5, M_6 calculate the N symbols of channel matrix H_0 ($N_{symbols/H0}$).

6.2. Example of equation solution and implementation: calculating $P^{1/2}$ and Q_α

With Eqs. (10), we can calculate the number of CORDIC operators necessary to be able to adapt to the given data rates while reducing complexity.

For example, if we define a frame as a packet of 7 symbols ($N_{symbols/H} = 7$), then, to achieve 600 Mbit/s, the optimum number of CORDIC operators is 5 which is calculated by Eq. (10), assuming that the operating frequency

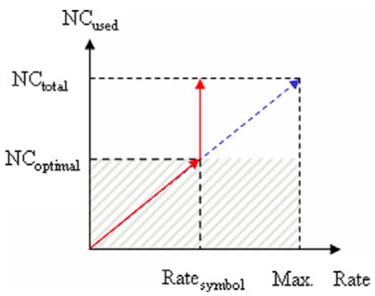


Fig. 8. Relationship between data-rate and number of CORDIC operators used.

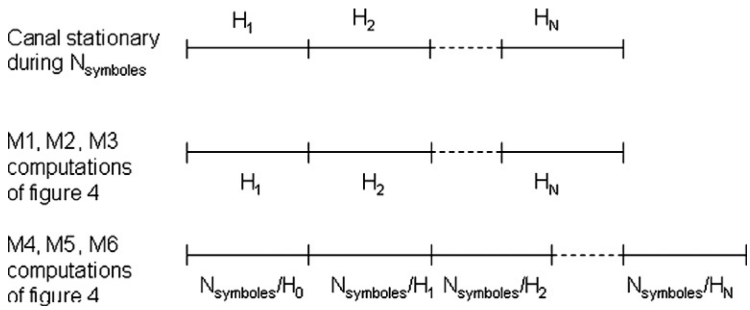
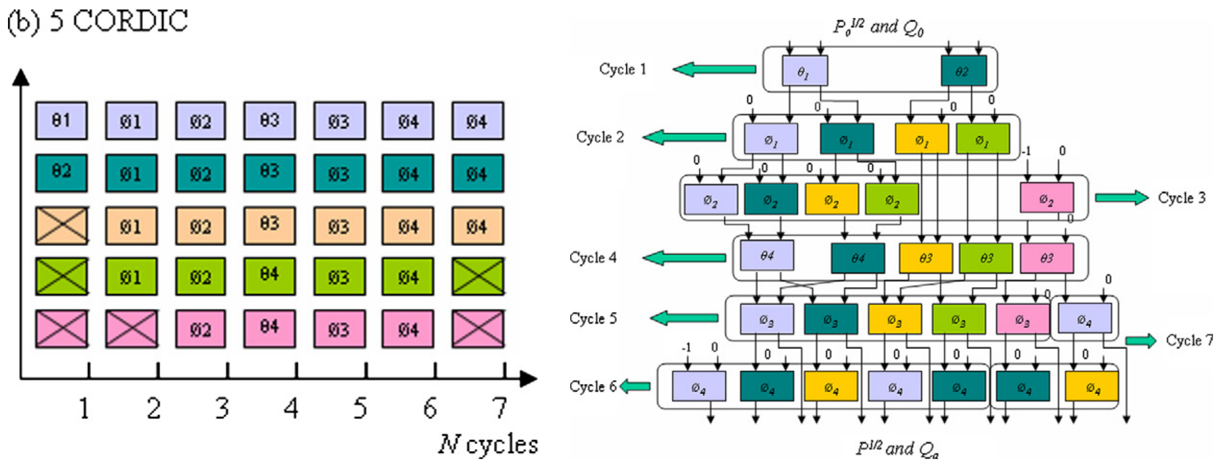
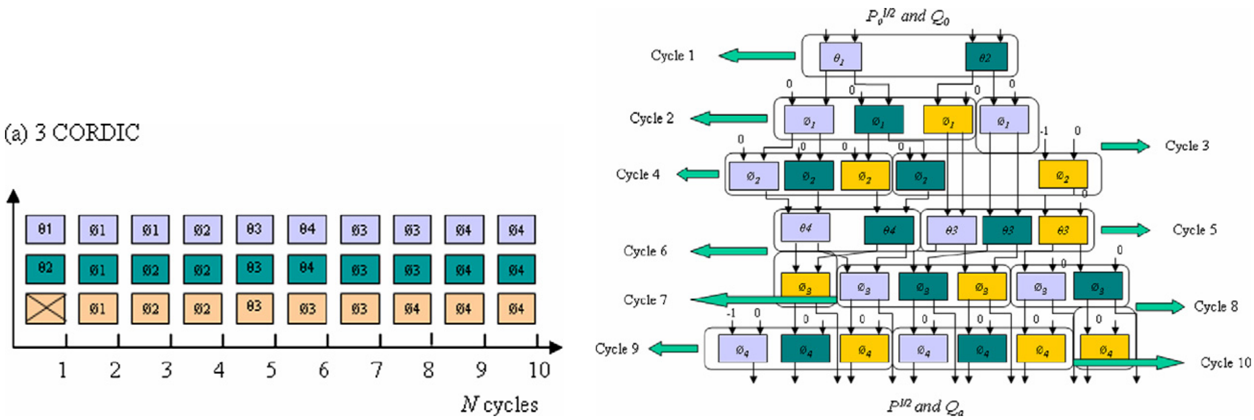


Fig. 9. Chronogram of data flow.



(a) (b)

Fig. 10. (a) Organization of the various structures on a basis of 5 CORDICs. (b) Iterative use of 5 CORDICs.



(a) (b)

Fig. 11. (a) Organization of the various structures on a basis of 3 CORDICs. (b) Iterative use of 3 CORDICs.

of a CORDIC operator is 150 MHz. Calculation of $P^{1/2}$ and Q_α may then be performed by 5 CORDIC operators in parallel in 7 cycles (Figs. 10(a) and (b)).

Now to obtain a lower data rate of the order of 400 Mbit/s, we change the number of CORDIC operators. By applying Eq. (10), the number of operators that need to be implanted is 3. Calculation of $P^{1/2}$ and Q_α can then be performed in 10 cycles. The organization of the architecture of the CORDIC operators is shown in Figs. 11(a) and (b).

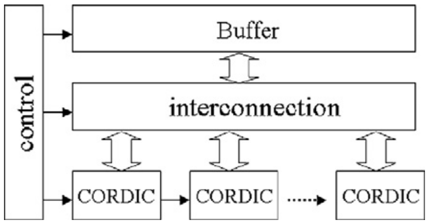


Fig. 12. Diagram of unitary transformation blocks (M_1, M_2, M_3).

Table 3
Result of the synthesis for module M_1

FPGA Xilinx Virtex	Number of slices	Max. freq. (MHz)	Data rate (Mbit/s)
29 CORDICs	16 840	148.6	600
5 CORDICs	4560	148.6	600
3 CORDICs	3726	148.6	400

In the parallel structure in Fig. 7, 29 CORDIC operators were used. In this new architecture, we are now using only 5 CORDIC operators in parallel (see Figs. 10(a) and (b)). We use these 5 CORDIC operators iteratively for 7 cycles. In the first cycle, we use 2 CORDIC operators to perform 2 Givens rotations with angles θ_1 and θ_2.

Then in the next cycle, 4 Givens rotations with the different angles (shown in Fig. 10(b)) are performed by 4 CORDIC operators, the first two of which are the same as in the first cycle and are re-used in the second cycle, and so on for the following cycles. So the whole of the processing is performed by 5 CORDIC operators in 7 cycles. The operation is longer than with the all-parallel structure, but we are optimizing hardware resources.

The architecture of the module is shown in Fig. 12. It thus consists of a variable number operators CORDIC whose ways of connections between them must changed during the treatment. When the required data rate changes, we can modify the number of CORDIC operators and their interconnections. For that reconfigurable architecture will be supplemented by a block of reconfigurable interconnections including of the storage blocks intended to memorize the intermediate results between each reconfiguration of the connections. The controller manages the scheduling of the operations and the transfers.

6.3. Implementation of the calculation of $P^{1/2}$ and Q_α

We compare the implementation of these three architectures in a Virtex FPGA circuit from Xilinx in order to determine the most suitable architecture for hardware realization. The first is a parallel, pipeline architecture that yields the best data-rate performance but with a significant area cost. Then we optimize the area by reducing the number of CORDIC operators. Then another optimization uses 3 CORDICs. This can be used when the data rate is not vital; it makes it possible to reduce the complexity of the architecture. The critical path time is 6.729 ns, and hence the maximum frequency is 148.6 MHz.

Received messages are 16-bit processed. Estimated symbols are calculated in parallel. In each cycle we can obtain 2 symbols in parallel. The modulation used is QPSK. Table 3 summarises the results obtained as a function of the number of CORDIC operators used:

In Table 3, the number of CORDIC operators is the number of CORDIC operators in rotation mode.

6.4. Implementation of the whole algorithm

Modules M_1, M_2, M_3 have been implanted as seen previously, and modules M_4, M_5, M_6 are implanted in the following manner: These modules consist of several PE stages to achieve a high processing speed for the received messages (Fig. 13). Each PE stage consists of a multiplier-accumulator, a comparator, a subtractor and a memory. The speed can be further improved by paralleling several PE stages. If the speed is not a determining criterion, a single PE can be used in an iterative fashion in order to reduce module complexity.

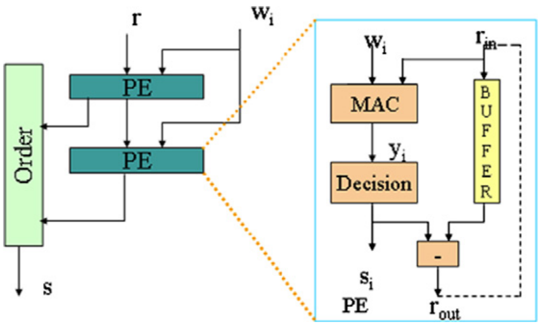


Fig. 13. Architecture of modules M_4, M_5, M_6.

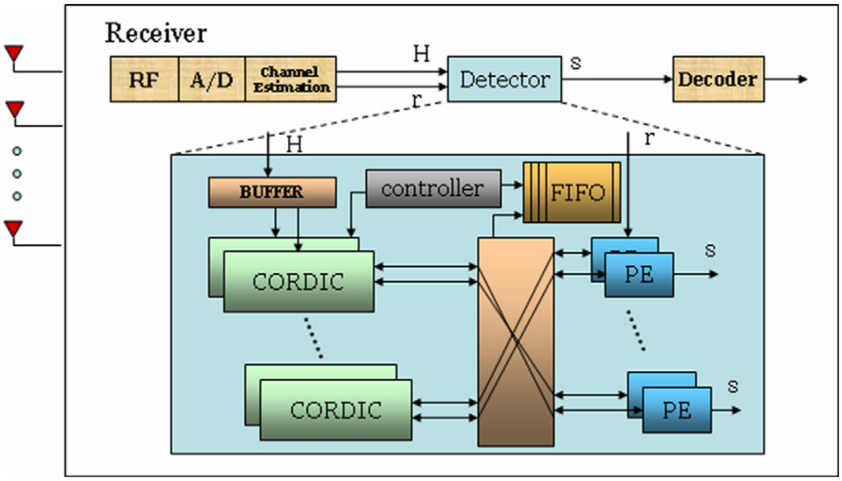


Fig. 14. Architecture of the V-BLAST Square Root detector.

Table 4
Result of the synthesis of the whole algorithm

FPGA Xilinx Virtex	Number of slices	Max. freq. (MHz)	Data rate (Mbit/s)
56 CORDIC	29 036	148.6	600
16 CORDIC	14 380	148.6	600
8 CORDIC	9936	148.6	400

After having implanted all the basic modules, we can incorporate them into an overall architecture. This overall architecture of the detector is illustrated in Fig. 14. We use a controller to organize the interconnections between all the modules. The data are stored in a FIFO memory predefined by Xilinx. The size of the memory depends on the number of operators used (CORDIC and PE). In Table 4, the number of CORDIC operators is the sum of CORDIC operators in vector mode and in rotation mode.

Received messages are 16-bit processed. Estimated symbols are calculated in parallel. Each cycle we can obtain 2 symbols in parallel. The modulation used is a QPSK. Table 4 summarises the results obtained as a function of the number of CORDIC operators used. The modules of the architecture are individually synthesized by Xilinx's ISE 6.3 synthesis tool. With the help of PLANHEAD from Xilinx, we can define the zones of each module of the architecture in a Virtex FPGA. Then we use ISE 6.3 to place and route these modules. Fig. 15 shows a view of the V-BLAST Square Root detector architecture after placement-routing in a Virtex II FPGA.

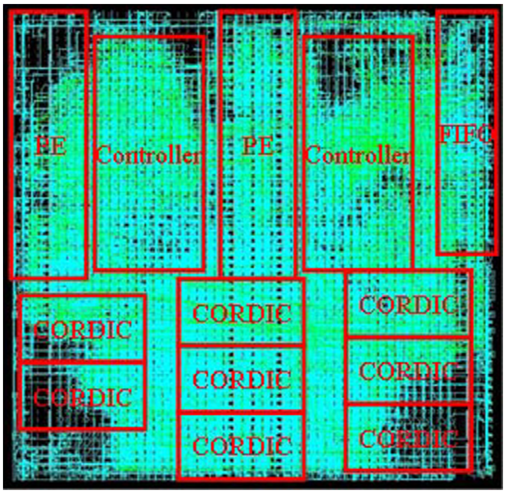


Fig. 15. Implementation of the architecture in an FPGA (FPGA_editor) for an architecture with 8 CORDICs.

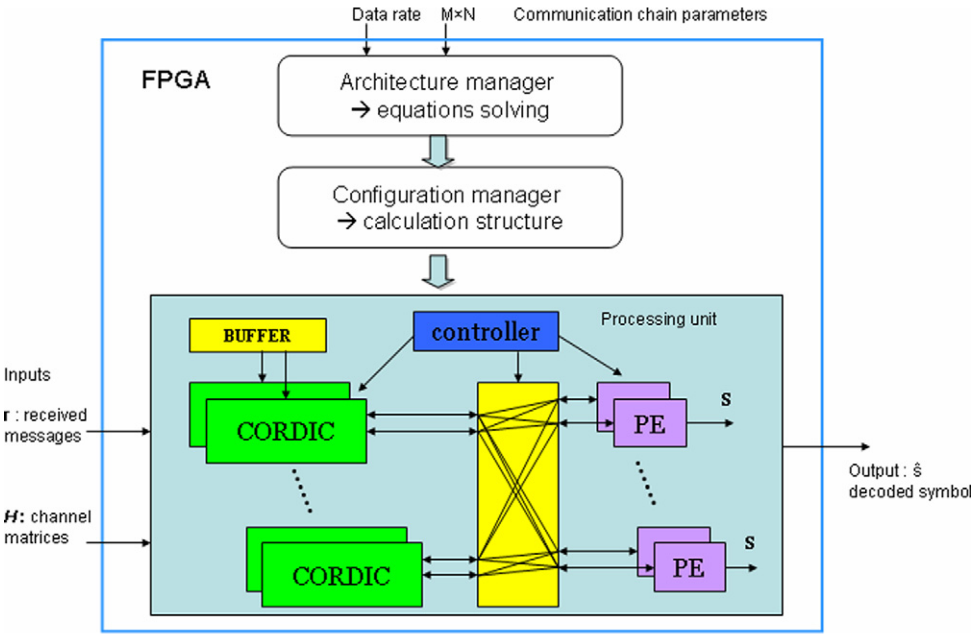


Fig. 16. Reconfiguration management flow.

6.5. Dynamic reconfiguration of the architecture

We have demonstrated that for a set of fixed parameters characterizing the transmission, we can deduce the optimal realization architecture for it that guarantees algorithm performance for minimal hardware complexity. By using FPGA technology, we can reconfigure the processing system in such a way as to implement this optimal architecture, but for a given combination of parameters and having previously produced the different implementations for different combinations.

In a MIMO system where dynamic adaptation of transmission parameters is envisaged (variable data rate, evolutionary antenna structure), it is necessary to design a hardware platform for realizing the decoder that is capable of modifying its architecture as a function of these context changes. FPGA technology with partial dynamic reconfiguration permits of this approach.

The dynamically-reconfigurable architecture we are proposing consists on the one hand of reconfiguration management functions, and on the other, processing functions (Fig. 16).

The architecture manager is applied to solving the equations linking the transmission characteristics to the architecture parameters defined earlier. The data from this analysis are transmitted to the configuration manager, whose task is to construct the optimum processing architecture [18]. To achieve this, the configuration manager has reconfigurable hardware at its disposal; the hardware reconfiguration involves the number of CORDIC operators implemented and the data paths, together with the control logic.

7. Conclusion

We have performed implantation of the MIMO detector based on the 'V-BLAST Square Root' algorithm. This architecture offers great flexibility in terms of the number of antennas and the different data rates given. It adapts to different requirements by using a varying number of CORDIC operators.

Our architecture allows us to obtain significant data rates, thanks to the parallelism of the architecture, though of course at the expense of using a significant area of the FGPA circuit. One of the objectives of our future work will be to reduce this area used to a minimum. Possible research routes are: on the one hand, optimal sequencing, so as to never leave a CORDIC operator inactive; and on the other hand, to realize modules M_4, M_5, M_6 also using CORDIC operators. We shall be studying this possibility by using an interpretation of these modules in the frequency domain, and hence of the PE elementary processors.

Another extension to this work is to develop an architecture adapted to the needs of future radio communication systems by dynamic reconfiguration. This architecture will be defined in blocks, enabling partial reconfiguration of the hardware resources in order to minimize the cost of reconfiguration.

References

[1] G.J. Foschini, Layered space–time architecture for wireless communication in a fading environment when using multi-element antennas, Bell Labs Technical Journal (Autumn 1996) 41–57.

[2] Z. Guo, P. Nilsson, A VLSI implementation of MIMO detection for future wireless communications, in: Proc. IEEE PIMRC'03, vol. 3, 2003, pp. 2852–2856.

[3] D. Gesbert, M. Shafi, D.S. Shiu, P. Smith, A. Naguib, From theory to practice: An overview of MIMO space–time coded wireless systems, IEEE Journal on Selected Areas in Communications 21 (3) (April 2003) 281–302.

[4] G.J. Foschini, M.J. Gans, On limits of wireless communications in a fading environment when using multiple antennas, Wireless Personal Communications 6 (March 1998) 311–335.

[5] E. Teletar, Capacity of multi-antenna Gaussian channels, AT&T Bell Labs, Tech. Rep., June 1995.

[6] D. Gesbert, J. Akhtar, Breaking the barriers of Shannon's capacity: An overview of MIMO wireless systems, Telektronikk Telenor Journal (January 2002).

[7] M. Alamouti, A simple transmit diversity technique for wireless communications, IEEE Journal on Selected Areas in Communications 16 (8) (October 1998) 1451–1458.

[8] V. Tarokh, N. Seshadri, A.R. Calderbank, Space–time codes for high data-rate wireless communication: performance criterion and code construction, IEEE Transactions on Information Theory 44 (2) (March 1998) 744–765.

[9] G.D. Golden, G.J. Foschini, R.A. Valenzuela, P.W. Wolniansky, Detection algorithm and initial laboratory results using V-BLAST space–time communication architecture, Electronics Letters 35 (1) (January 1999) 14–15.

[10] P.W. Wolniansky, G.J. Foschini, G.D. Golden, R.A. Valenzuela, V-BLAST: An architecture for realizing very high data rates over the rich-scattering channel, invited paper, in: Proc. ISSSE-98, Pisa, Italy, September 1998.

[11] M.O. Damen, K. Abed-Meraim, J.C. Belfiore, A generalized sphere decoder for asymmetrical space–time communication architecture, IEE Electronic Letters 36 (January 2000).

[12] M.O. Damen, A. Chkeif, J.C. Belfiore, Lattice code decoder for space–time codes, IEEE Communications Letters 4 (May 2000) 166–169.

[13] A.H. Land, A.G. Doig, An automatic method for solving discrete programming problems, Econometrica 28 (1960) 497–520.

[14] J. Luo, P. Willett, K. Pattipati, L. Brunel, Branch-and-bound-based fast optimal algorithm for multiuser detection in synchronous CDMA, in: Communications, 2003, ICC'03, IEEE International Conference, vol. 5, May 2003, pp. 3336–3340.

[15] W. Zha, S.D. Blostein, Multiuser receivers that are robust to mismatch, in: Special Issue on Multiuser Detection and Blind Estimation, EURASIP Journal on Applied Signal Processing 12 (December 2002) 1355–1364.

[16] B. Hassibi, An efficient square-root algorithm for BLAST, http://mars.bell-labs.

[17] C.M. Rader, VLSI systolic arrays for adaptive nulling, IEEE Signal Processing Magazine 13 (4) 29–49.

[18] J.P. Delahaye, J. Palicot, P. Leray, A hierarchical modeling approach in software-defined radio system design, in: SIPS 2005, Athens, Greece.