Hindawi Publishing Corporation
Journal of Engineering
Volume 2013, Article ID 534735, 7 pages
http://dx.doi.org/10.1155/2013/534735

Hindawi

Research Article

A VLSI Architecture for the V-BLAST Algorithm in Spatial-Multiplexing MIMO Systems

Pedro Cervantes-Lozano,[1] Luis F. González-Pérez,[2] and Andrés D. García-García[1]

[1] *ITESM Campus Estado de México, Atizapán de Zaragoza, Mexico*
[2] *ITESM Campus Guadalajara, Guadalajara, Mexico*

Correspondence should be addressed to Pedro Cervantes-Lozano; a00664970@itesm.mx

Received 31 August 2012; Revised 27 November 2012; Accepted 27 November 2012

Academic Editor: Karim Kabalan

This paper presents a VLSI architecture for the suboptimal hard-output Vertical-Bell Laboratories Layered Space-Time (V-BLAST) algorithm in the context of Spatial Multiplexing Multiple-Input Multiple-Output (SM-MIMO) systems immersed in Rayleigh fading channels. The design and implementation of its corresponding data-path and control-path components over FPGA devices are considered. Results on synthesis, bit error rate performance, and data throughput are reported.

1. Introduction

Multiple-Input Multiple-Output (MIMO) communication systems enhance spectral efficiency and bit error rate (BER) performance over wireless communication links [1–3]. MIMO is already considered as the transmission scheme for emerging wireless communication standards such as 802.11 n (WiFi), 802.16 d/e (WiMAX), and 802.11 ac (multi-user MIMO WLAN) [4]. Digital signal processing (DSP) algorithms for symbol decoding in these systems immersed in Rayleigh fading channels require trade-off design challenges regarding BER performance, data throughput, and complexity. Sub-optimal hard-output Spatial-Multiplexing MIMO (SM-MIMO) demodulation techniques [1, 2] offer low complexity aspects with a very high data throughput, but at a penalty in BER performance degradation as compared to Maximum-Likelihood (ML) performance. The V-BLAST (Vertical-Bell Laboratories Layered Space-Time) algorithm [5, 6] is an adequate sub-optimal hard-output SM-MIMO demodulation technique due to the previous mentioned properties. To the best of the authors' knowledge, previous attempts on state-of-the-art VLSI implementations for hard-output V-BLAST-based sub-optimal SM-MIMO demodulation architectures are reported in literature [7, 8]. Seeking for low-power consumption and cost-effectiveness, these ASIC-based (Application-Specific Integrated Circuit) approaches permit no flexibility in meeting these attributes towards prototyping this kind of DSP solutions. In this work, a VLSI architecture for the sub-optimal hard-output detection V-BLAST algorithm is presented. The contribution of this paper is to present a novel VLSI architecture of the V-BLAST algorithm implemented on FPGA devices that perfectly suits SM-MIMO demodulation requirements regarding BER performance, hardware complexity, and data throughput while operating in Rayleigh fading channels, behaving competitively against other earlier approaches. The organization of this paper is as follows: Section 2 presents the MIMO communication model. The V-BLAST algorithm is presented in Section 3. Section 4 highlights the architecture proposal for the V-BLAST algorithm. Implementation results and comparison analysis are exposed in Section 5. Conclusions are covered in Section 6.

2. MIMO Communication Model

The MIMO communication model consists of an antenna array of n_T elements at the transmitter end and n_R elements at the receiver in the presence of an ccs-iid AWGN (circularly-complex symmetric, identically-distributed Additive White Gaussian Noise) Rayleigh fading channel [1–3]. Information signal vector $x = [x_1 \ \cdots \ x_{n_T}]^T$, whose n_T

entries are symbols drawn from a q-QAM (q-ary Quadrature Amplitude Modulation) constellation (known also in this context as a Gaussian finite-integer lattice), is transmitted throughout these n_T antennas. The received signal vector $y = [y_1 \ \cdots \ y_{n_R}]^T$ of dimension n_R can be mathematically described as

$$y = Hx + \eta, \tag{1}$$

where x and y were defined above, $\eta = [\eta_1 \ \cdots \ \eta_{n_R}]^T$ is a n_R AWGN vector, and $H = \begin{bmatrix} h_{11} & \cdots & h_{1n_T} \\ \vdots & & \vdots \\ h_{n_R1} & \cdots & h_{n_Rn_T} \end{bmatrix}$ is the $n_R \times n_T$ MIMO channel matrix (entries ij from H correspond to the fading between the ith receiver and the jth transmitter antennas). System statistics are assumed to be invariant during a MIMO channel realization [1]. Without loss of generality, $n_R = n_T = n$ will be considered in the sequel. Applying QR decomposition [9, 10] to H in (1), that is, $H = QR$, yields

$$\widetilde{y} = Rx + \overline{\eta}, \tag{2}$$

where

$$Q = \begin{bmatrix} q_{11} & \cdots & q_{1n} \\ \vdots & & \vdots \\ q_{n1} & \cdots & q_{nn} \end{bmatrix}, \qquad R = \begin{bmatrix} r_{11} & \cdots & \cdots & r_{1n} \\ 0 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & r_{nn} \end{bmatrix} \tag{3}$$

are $n \times n$ complex orthogonal and upper-triangular matrices, respectively; moreover, $\widetilde{y} = Q^H y$ and $\overline{\eta} = Q^H \eta$, where $[\cdot]^H$ is the conjugate-transpose operator. Obviously speaking, $Q^H H = R$ which reveals that Q^H is equivalent to inverse of Q, that is, Q^{-1}. The problem to solve in this MIMO communication scenario is to find the transmitted vector x among q^n possible candidates given that vector y was received and matrix H has been accurately estimated. As can be seen, an exhaustive search is prohibitively complex. A fast decoding procedure alleviates this complexity constraint by taking advantage of matrix structure of R presented in (2) under a successive interference cancellation (SIC) strategy [5, 6], thus yielding high data throughputs for symbol-decoding purposes. That is why sub-optimal decoding algorithms are preferred. One of the best sub-optimal hard-output decoding algorithms is the V-BLAST which is explained next.

3. The V-BLAST Algorithm

The main idea behind the V-BLAST algorithm, as a sub-optimal hard-output SM-MIMO demodulation technique, is that instead of performing an exhaustive search, symbol decoding of x is performed under an ordered-iterative and back-propagation way (identified also as OSIC: Ordered Successive Interference Cancellation), in which noise (associated with $\overline{\eta} = [\overline{\eta}_1 \ \cdots \ \overline{\eta}_n]^T$) and cochannel interference (related to elements in R) are treated through a SNR (Signal-to-Noise Ratio) optimization criterion that determines the order in which symbol entries of x will be decoded [5, 6]. At each iteration, an entry of vector x is sliced into a q-QAM value at the receiver end assuming that only one transmitter end element possesses the highest SNR (choice based on an optimization criterion), and thus the remaining transmitter elements are considered as interference (besides the presence of AWGN). With these ideas exposed, the V-BLAST algorithm is defined into the following steps (Steps 1–8).

Step 1 (*initial conditions*). Perform the assignments:

$$H_1 = R = \begin{bmatrix} h_{11}^1 & \cdots & h_{1n}^1 \\ \vdots & & \vdots \\ h_{n1}^1 & \cdots & h_{nn}^1 \end{bmatrix},$$

$$G_1 = R^{-1} = \begin{bmatrix} g_{11} & \cdots & g_{1n} \\ \vdots & & \vdots \\ g_{n1} & \cdots & g_{nn} \end{bmatrix}, \tag{4}$$

$$X^1 = \widetilde{y} = [\widetilde{y}_1 \ \cdots \ \widetilde{y}_n]^T.$$

Let $i = 1$ and $\kappa_1 = [1 \ \cdots \ n]$ be an index vector.

Step 2 (*pre/post-detection*). While $i \le n$, and for $1 \le j \le (n - i+1)$, a SNR optimization criterion dictates the order in which symbol entries of x will be decoded according to index k_i, and is stated as

$$\begin{aligned} k_i &= \underset{j}{\operatorname{argmin}} \left\| \left[g_{j1}^i \ \cdots \ g_{jn}^i \right] \right\|_2^2 \\ &= \underset{j}{\operatorname{argmin}} \left(\sum_{m=1}^{n} \left| g_{jm}^i \right|^2 \right), \end{aligned} \tag{5}$$

where elements $g_{j1}^i, \ldots, g_{jn}^i$ are obtained form the Generalized-Inverse G_i (or Left-Pseudoinverse) [11] of the MIMO channel matrix deflated versions H_i, that is,

$$H_i = \begin{bmatrix} h_{11}^i & \cdots & h_{1,(n-i+1)}^i \\ \vdots & & \vdots \\ h_{n1}^i & \cdots & h_{n,(n-i+1)}^i \end{bmatrix}, \tag{6}$$

$$G_i = H_i^\dagger = \begin{bmatrix} g_{11}^i & \cdots & g_{1n}^i \\ \vdots & & \vdots \\ g_{(n-i+1),1}^i & \cdots & g_{(n-i+1),n}^i \end{bmatrix}, \tag{7}$$

with $H_i^\dagger = \left(H_i^H H_i\right)^{-1} H_i^H$. Make $\widetilde{k}_i$ be the k_i-th element of κ_i.

Step 3 (*nulling*). Co-channel interference is mitigated by the use of a nulling vector $g_{\widetilde{k}_i}^i$, that is, obtained from rows of (7), through operation

$$\widehat{y_{\widetilde{k}_i}} = g_{\widetilde{k}_i}^i \cdot X^i = \sum_{m=1}^{n} g_{k_i m}^i \cdot x_m^i, \tag{8}$$

with $X^i = [x_1^i \ \cdots \ x_n^i]^T$, $g_{\widetilde{k}_i}^i = [g_{k_i 1}^i \ \cdots \ g_{k_i n}^i]$, and $\widehat{y_{\widetilde{k}_i}}$ is the signal to be sliced.

Step 4 (*quantization*). A slicing operator $Q[\cdot] : \mathbb{C} \mapsto \mathbb{Z}_{[j]}(q) \subset (\mathbb{Z}^+ + j\mathbb{Z}^+)$ (which maps the signal

$$\widehat{y_{\widetilde{k}_i}} = \overbrace{\widehat{x_{\widetilde{k}_i}}}^{\text{symbol}} + \overbrace{\sum_{\tau=1}^{i-1}\left[\left(\sum_{m=1}^{n} g^i_{k_i m}\cdot h^{\tau}_{m k_\tau}\right)\cdot\left(x_{\widetilde{k}_\tau} - \widehat{c_{\widetilde{k}_\tau}}\right)\right]}^{\text{co-channel interference}} + \overbrace{\sum_{m=1}^{n} g^i_{k_i m}\cdot\overline{\eta}_m}^{\text{noise}} \tag{9}$$

into a constellation point of the q-QAM modulation lattice) is applied to (8), yielding

$$\widehat{c_{\widetilde{k}_i}} = Q\left[\widehat{y_{\widetilde{k}_i}}\right]. \tag{10}$$

This outcome will be the $\widetilde{k}_i$th entry of signal vector $\widehat{x_V}$, that is, $\widehat{x_V}[\widetilde{k}_i] = \widehat{c_{\widetilde{k}_i}}$. If $i = n$, go to Step 8; otherwise, go to Step 5.

Step 5 (*cancellation*). The treatment for co-channel interference and AWGN mitigation after (8)–(10) at iteration i is cancelled out according to

$$X^{i+1} = X^i - h_{\widetilde{k}_i}\cdot\widehat{c_{\widetilde{k}_i}}, \tag{11}$$

where $h_{\widetilde{k}_i} = [h_{1,\widetilde{k}_i} \quad \cdots \quad h_{n,\widetilde{k}_i}]^T$ is the $\widetilde{k}_i$th column of H_1.

Step 6 (*updating*). By, respectively, removing the $\widetilde{k}_i$th element of κ_i and the k_ith column of H_i, deflated versions for the MIMO channel matrix are created, as well as reshaping index vector κ_i, such as:

$$H_{i+1} = \left(\cdots\left(H_1\right)^{\overline{\overline{k}}_1}\cdots\right)^{\overline{\overline{k}}_i}, \tag{12}$$

$$\kappa_{i+1} = \left(\cdots\left(\kappa_1\right)^{\overline{k}_1}\cdots\right)^{\overline{k}_i}. \tag{13}$$

Step 7. Increment i and go to Step 2.

Step 8. $\widehat{x_V} = [\widehat{c}_1 \quad \cdots \quad \widehat{c}_n]^T$ is the transmitted vector.

4. Architecture Proposal

The block diagram of the proposed architecture for the sub-optimal hard-output V-BLAST-based SM-MIMO algorithm is shown in Figure 1. The overall architecture for the V-BLAST algorithm consists of two components: the *Data-Path (DP)* is constituted by processing elements that implement all necessary mathematical operations; the *Control-Path (CP)* provides signaling for synchronization of data-flow for the appropriate decoding of vector x (represented by output $\widehat{x_V}$). In Figure 1, K_{clock} is the system clock, $\widetilde{y}$ and R are the inputs. Additional external signals are employed for initialization of V-BLAST operations through a reset signal ($\overline{RESET}$), detection of valid information at the inputs (signal *DV_inputs*), and indication of valid information at the output (flag *DV_output*). Remember that the V-BLAST architecture is designed in order to perform symbol decoding of $\widehat{x_V}$ generated from a 16-QAM constellation in AWGN Rayleigh fading channels for MIMO systems with $n = 4$. The details concerning the design of the data-path and control-path components are exposed in the subsequent sections.

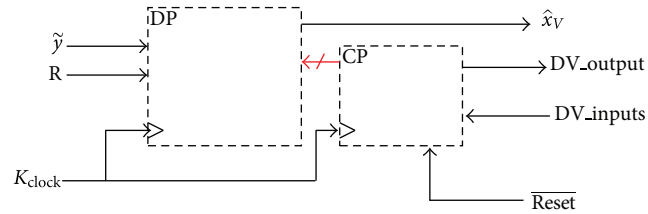

Figure 1: Block diagram for the V-BLAST architecture proposal.

4.1. Finite-Precision Analysis. The configuration of the DP component in the V-BLAST architecture considers design specifications based on fixed-point arithmetic for inputs $\widetilde{y}$ and R (results were carried out by floating and fixed point MATLAB simulations). Figure 2 reveals the BER performance of the V-BLAST algorithm considering specifications mentioned above. Optimal performance is obviously the ML solution (ML-D legend, red line). It can be seen from the figure that using a 16-bit fixed-point word-length for R showed an acceptable performance of the V-BLAST as compared to its floating-point model (blue and blue-dotted lines) with less than 0.01 dB in loss. On the contrary, a reduction in finite-precision, that is, 8-bit word-length, caused a remarkable performance degradation of more than 1 dB (black-dotted line). In addition, 16 bits for both the real and imaginary parts of each entry in $\widetilde{y}$ was also considered.

4.2. Data-Path Architecture. The architecture of the DP component is illustrated in Figure 3, and consists of the following elements: (i) data multiplexors MR and MYG (identified by M_R and $M_{\widetilde{y}}$, resp.) for selecting between $H_1 = R$ and H_i, as well as for $X^1 = \widetilde{y}$ and X^{i+1}; (ii) registers REG_Hi and REG_Xi store, respectively, information regarding H_i in (6) and X^{i+1} in (11); (iii) a block InviPseudo for matrix inversion and left-pseudoinversion [9–11] presented in (7); (iv) P/P-D implements optimization criterion (5); (v) N computes the nulling operation in (8); (vi) slicing operation for quantization (10) are implemented in Q; (vii) C performs cancellation (11); (viii) So is in charge of properly assign decoded symbol entries to vector $\widehat{x_V}$; and (ix) L manages deflated matrix versions in (12) and vector index reshaping in (13).

The V-BLAST architecture provides a decoded symbol entry of $\widehat{x_V}$ at an iteration i (exactly n iterations are required), meaning that multiplexors MYG and MR, and registers REG_Hi and REG_Xi regulate data flow contained in H_1, H_i, X^1, X^i, and X^{i+1}. For $i = 1$ is evident that MR selects matrix R, and MYG does the same for vector $\widetilde{y}$. Whenever

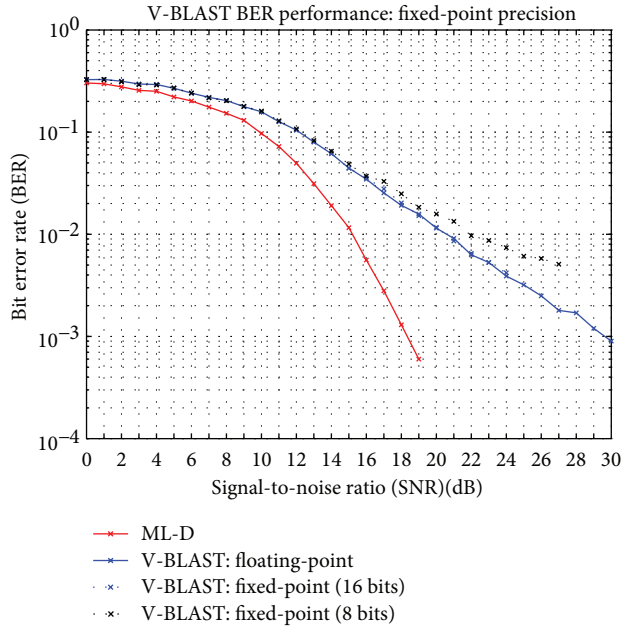

FIGURE 2: BER performance of the V-BLAST algorithm restricted to finite-precision over inputs $\tilde{y}$ and R.

$1 < i \leq n$, the vector X^i is updated into X^{i+1} following (11) through REG_Xi; similarly happens to the matrix deflated versions H_i following (12) through REG_Hi. Moreover, the matrix deflated versions H_i are handled as just zeroing the $\widetilde{k}_i$th column of H_1 after each iteration i, yielding matrices

$$H_i = \begin{bmatrix} \overbrace{\begin{matrix} * & * \\ \vdots & \vdots \\ * & * \end{matrix}}^{(n-i+1)\text{ elements}} & \overbrace{\begin{matrix} \times & \times \\ \vdots & \vdots \\ \times & \times \end{matrix}}^{(i-1)\text{ elements}} \end{bmatrix}, \quad (14)$$

where $[\times \ \cdots \ \times]^T$ denotes a column vector filled with zeros, and $[* \ \cdots \ *]^T$ indicate the remaining column entries, which are properly reordered according to entries contained in index vector κ_i. The P/P-D block computes (5) for every row entry of G_i. With the aid of a Batcher sorting network [12], a nulling vector $g^i_{\widetilde{k}_i}$ is chosen from a set of $n-i+1$ vector candidates based on $\|[g^i_{j1} \ \cdots \ g^i_{jn}]\|_2^2$ which is equivalent to $\sum_{l=1}^{n} |g^i_{jl}|^2$. Each one of these results is labeled consecutively from 1 to $n-i+1$. The index k_i adopts then the value appended to the selected nulling vector. The N block uses the selected nulling vector $g^i_{\widetilde{k}_i}$ in order to perform (8), implementing all complex multiply-and-accumulate operations related to $g^i_{\widetilde{k}_i} \cdot X^i = \sum_{m=1}^{n} g^i_{k_i m} \cdot x^i_n$. The Q block (or slicer) transforms $\widehat{y_{\widetilde{k}_i}}$ into a signal point $\widehat{c_{\widetilde{k}_i}}$ belonging to a q-QAM constellation ($\widehat{c_{\widetilde{k}_i}}$ is coded into a $\log_2 q$-bit word). For $1 \leq m \leq n$, complex additions and multiplications inherent in $x^{i+1}_m = x^i_m - h_{m\widetilde{k}_i} \cdot \widehat{c_{\widetilde{k}_i}}$ complete cancellation in C with the sliced symbol $\widehat{c_{\widetilde{k}_i}}$, the multiplexed X^i vector, and the selected column $h_{\widetilde{k}_i}$ from H_1. The So block registers and accordingly assigns symbol decoded entries $\widehat{c_{\widetilde{k}_i}}$ into elements in vector $\widehat{x_V}$ based on index $\widetilde{k}_i$, in other words $\widehat{x_V}[\widetilde{k}_i] = \widehat{c_{\widetilde{k}_i}}$ (the value of $\widehat{c_{\widetilde{k}_i}}$ is the $\widetilde{k}_i$th element of vector $\widehat{x_V}$). The L block deals with the process of generating deflated versions H_i from H_1, and keeps track of indexes $\widetilde{k}_i$ and k_i. Also, L provides the pertinent value of $h_{\widetilde{k}_i}$ at every iteration i. As mentioned before, in order to perform the deflating operations in (12) the $\widetilde{k}_i$th column of matrix H_1 is removed and substituted by an n-dimensional zero vector. For the index vector update in (13), the k_ith element of the $(n-i+1)$-dimensional array κ_i is assigned to index $\widetilde{k}_i$, that is, $\kappa_i[k_i] = \widetilde{k}_i$, then this element is removed afterwards from κ_i and it is re-sized into a $(n-i)$-dimensional array κ_{i+1}.

The InviPseudo block is the key element for performing iterative generalized-inverses found in (7). The heavy and critical operation to be treated relies on $(H_i^H H_i)^{-1}$, since $G_i = H_i^{\dagger} \doteq (H_i^H H_i)^{-1} H_i^H$. InviPseudo implements a strategy for computing G_i based on a block-matrix Left-Pseudoinverse (hereinafter referred as LPI kernel) approach as proposed in [11]. For the case $n = 4$ developed in this V-BLAST architecture, this LPI kernel is divided into the following entities: A, B, C, D; A^{-1}, D^{-1}; $L = BD^{-1}$, $M = D^{-1}C$, $N = CA^{-1}$; $\phi = A - LC$, ϕ^{-1}; $\alpha = \phi^{-1}L$, $-\alpha$; $\theta = D^{-1} + M\alpha$, $-\theta N$. All of these entities represent 2×2 complex-valued matrices that after all V-BLAST iterations are accomplished, $(H_i^H H_i)^{-1}$ will be reassembled as $\begin{bmatrix} \phi^{-1} & -\alpha \\ -\theta N & \theta \end{bmatrix}$. Depending on the V-BLAST iteration i, data inside each entity is properly initialized for the correct computation of block-matrix inversion within G_i. For example: (i) at $i = 1$: $A = \begin{bmatrix} e_{11} & e_{12} \\ e_{21} & e_{22} \end{bmatrix}$, $B = \begin{bmatrix} e_{13} & e_{14} \\ e_{23} & e_{24} \end{bmatrix}$, $C = \begin{bmatrix} e_{31} & e_{32} \\ e_{41} & e_{42} \end{bmatrix}$, $D = \begin{bmatrix} e_{33} & e_{34} \\ e_{43} & e_{44} \end{bmatrix}$; (ii) at $i = 2$: $A = \begin{bmatrix} e_{11} & 0 \\ 0 & e_{11} \end{bmatrix}$, $B = \begin{bmatrix} e_{12} & e_{13} \\ 0 & 0 \end{bmatrix}$, $C = \begin{bmatrix} e_{21} & 0 \\ e_{31} & 0 \end{bmatrix}$, $D = \begin{bmatrix} e_{22} & e_{23} \\ e_{32} & e_{33} \end{bmatrix}$; (iii) at $i = 3$: $A = \begin{bmatrix} e_{11} & e_{12} \\ e_{21} & e_{22} \end{bmatrix}$, $B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$, $C = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$, $D = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$; (iv) at $i = 4$: $A = \begin{bmatrix} e_{11} & 0 \\ 0 & e_{11} \end{bmatrix}$, $B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$, $C = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$, $D = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. For the cases presented in (i)–(iv), elements e_{ij} are taken accordingly from $H_i^H H_i$ at its corresponding iteration V-BLAST iteration i. Additionally, all arithmetic divisions presented in the LPI kernel and throughout iterations concerning (7) were implemented with CORDIC (Coordinate Rotate Digital Computer) processors [13]. For this purpose, the CORDIC processor (or CORDIC engine) is structured as

$$\begin{bmatrix} x[n+1] \\ y[n+1] \end{bmatrix} = K \begin{bmatrix} 1 & -\sigma_n \cdot 2^{-n} \\ \sigma_n \cdot 2^{-n} & 1 \end{bmatrix} \begin{bmatrix} x[n] \\ y[n] \end{bmatrix}, \quad (15)$$

where

$$\sigma_n = \begin{cases} +1 : y[n] < 0, \\ -1 : y[n] > 0 \end{cases} \quad (16)$$

is associated with the polarity of N micro-rotations in (15) for $0 \leq n \leq N-1$; $x[0] = a > 0$ and $y[0] = b$ are initial conditions in (15); K is an approximated scaling factor; and this engine is customized to perform $b/a = -\sum_{n=0}^{N-1} (\sigma_n \cdot \tan^{-1}(2^{-n}))$.

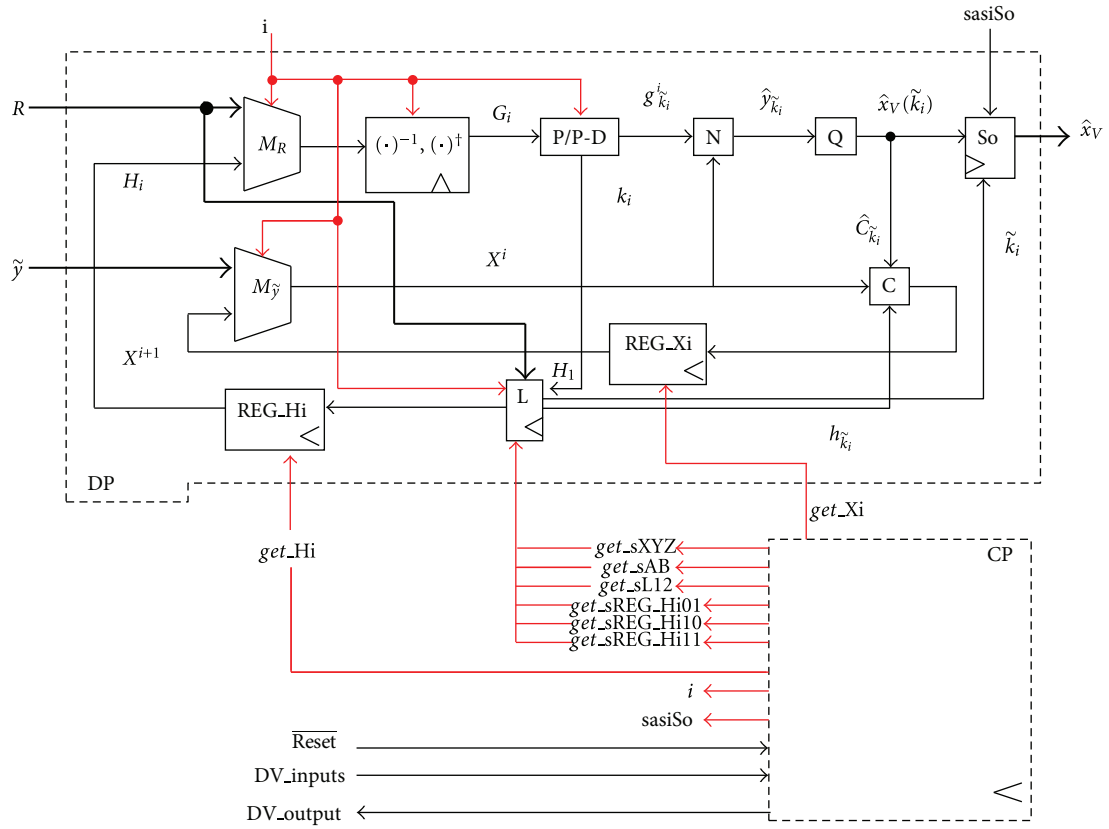

Figure 3: Architecture for the DP component of the V-BLAST. Red arrows are control signals generated by CP.

4.3. Control-Path. The CP component is a finite state machine that handles all required control signals for the DP to properly decode $\widehat{x_V}$ during a time period when channel statistics remain invariant, that is, a MIMO channel realization where inputs $\tilde{y}$ and R do not change in time. Referring to Figures 1 and 3, CP employs the following signals: (i) signals *get_Hi* and *get_Xi* capture data H_i and X^{i+1} in registers REG_Hi and REG_Hi, respectively: (ii) signal *sasiSo* captures every decoded entry $\widehat{c_{\tilde{k}_i}}$ in $\widehat{x_V}$ at block So; (iii) κ_i re-sizing in (13) is regulated by signals *get_*sXYZ, *get_sAB* and *get_sL12*; while deflated versions of H_1 in (12) are generated through signals *get_sREG_Hi01*, *get_sREG_Hi10* and *get_sREG_Hi11*. In addition to these, the roll of signal i is fundamental for synchronizing V-BLAST iterations, because i selects elements in MYG and MR, controls data flow in L, allows the generation of generalized-inverses G_i at InviPseudo, and regulates the choice of nulling vectors in N.

5. Results

The sub-optimal hard-output V-BLAST architecture was designed for operating with transmitted signal vectors generated from 16-QAM modulators immersed in AWGN Rayleigh fading channels with $n = 4$. Simulations for functional validation were programmed with MATLAB. A million of MIMO channel realizations were considered or until a thousand of error blocks were found, that is, a decoded block consisting of $n\log_2 q$ bits. Synthesis was performed with the Altera Quartus II IDE tool over Cyclone III FPGA devices.

5.1. BER Performance. The results for BER performance shown in Figure 2 were corroborated for the finite-precision analysis of the V-BLAST architecture. That is, the FPGA implementation showed the same performance as the one obtained in simulation for the fixed point case, confirming a negligible degradation as compared to theoretical fixed-point performance. Two different simulation scenarios were considered: (i) a MATLAB-based simulation model used for obtaining floating and fixed points (restricted to a 16-bit precision) ML and V-BLAST performances; (ii) a testbench designed for evaluating performance of the device under test, whose test vectors were generated with MATLAB.

5.2. Synthesis Results. The Cyclone III device form Altera FPGA family was selected as implementation target for the V-BLAST architecture. Different synthesis and fitting modes were performed within the Quartus II IDE tool. For instance, synthesis considered speed (sp), balanced (bd), and area (ar) optimization techniques, while fitting considered standard (std) and fast (fst) fitter efforts. Therefore, six different modes were evaluated for implementation purposes, namely: sp-std, bd-std, ar-std, sp-fst, bd-fst, and ar-fst. In all of these cases, hardware complexity resided on logic elements (LEs)

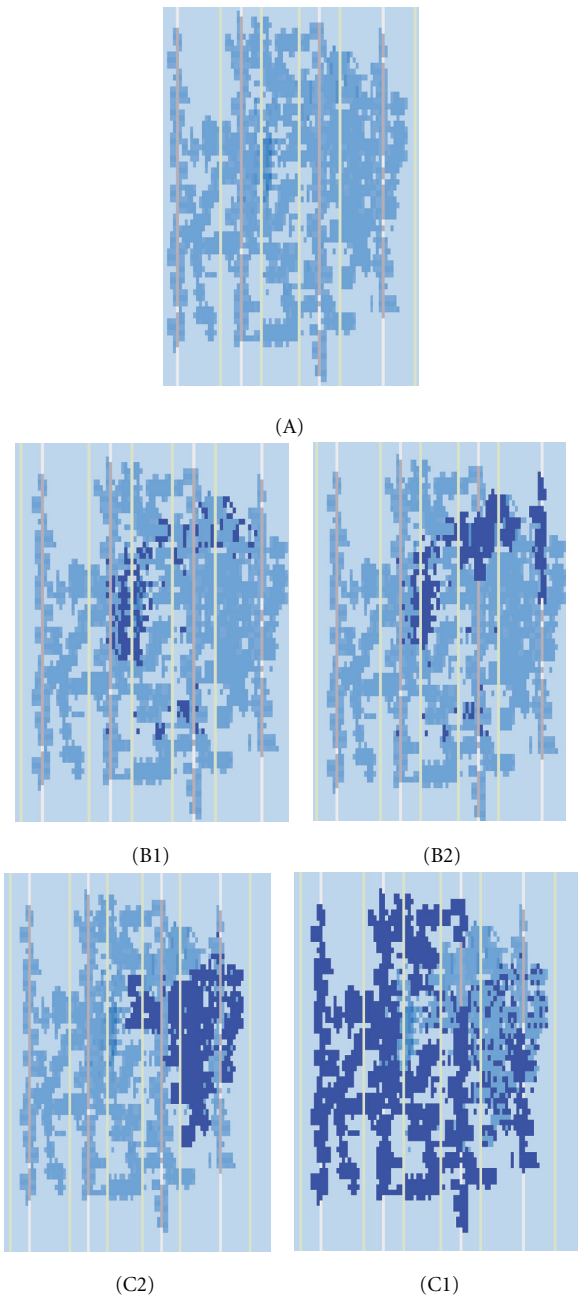

Figure 4: Floorplan results for the V-BLAST architecture.

and embedded multipliers (eMults). The whole V-BLAST architecture demanded respectively a 27% and a 72% usage of the total amount of LEs and eMults available in the FPGA device. In fact, (a) regarding LEs usage: 62.98% belonged to the InviPseudo block, 24.94% to P/P-D, 3.67% to C, 3.29% to L, 2.39% for N, and less than 0.5% for the remaining blocks (MR, MYG, REG_Hi, REG_Xi, Q, and So); (b) regarding eMults usage, 84.61% belonged to InviPseudo block, and 15.39% to N. The best temporal performance of the V-BLAST architecture is offered by the bd-std implementation mode, exhibiting a critical path of 10.056 ns with data throughput of 265.182 Mbps, since 4 bits were decoded after 1.5 clock cycles. This temporal performance perfectly suits SM-MIMO requirements for Rayleigh fading channels as stipulated in 802.11 n, whose specifications are around the 100–200 Mbps [14]. The V-BLAST architecture worked at a maximum clock frequency of 99.443 MHz with an overall decoding latency of 12 clock cycles, equivalent to 120.672 ns. Furthermore, Table 1 shows comparative results against other related works. The floorplan result of the complete V-BLAST architecture is provided in Figure 4: the overall V-BLAST architecture (top-view, label A); MR, MYG, REG_Hi, and REG_Xi (left middle-view, label B1); N, Q, C, So, and L (right middle-view, label B2); InviPseudo (left

Table 1: V-BLAST architecture comparison.

	[7]	[8]	This work
n	4	4	4
Device target	ASIC	ASIC	FPGA
Transmission scheme	QPSK	16-QAM	16-QAM
Data throughput	128 Mbps	212 Mbps	265.18 Mbps
Maximum clock frequency	80 MHz	140 MHz	99.44 MHz

bottom-view, label C1), and P/P-D (right bottom-view, label C2).

5.3. Comparison Analysis. Earlier attempts in providing architectural implementations on sub-optimal hard-output V-BLAST-based SM-MIMO solutions are cited in [7, 8]. Albeit their implementation structured on ASIC devices, which inhibits design flexibility and cost-effectiveness, the FPGA-based VLSI architectural approach developed in this work represents a modular, portable, and scalable implementation of the V-BLAST algorithm as depicted in Figure 3. Moduli constituting the V-BLAST architecture are configured under an RTL level, yielding a moderate capability support for high-dimensional lattices (i.e., q-QAM with 16, 64, 256 values), and high-dimensional MIMO communication scenarios (i.e., $n = 4, 8$). Performance results reported in Table 1 exhibit competitive aspects against other state-of-the-art solutions: a low-dimensional lattice (i.e., QPSK) V-BLAST [7], and a high-dimensional lattice (i.e., 16-QAM) V-BLAST [8]. The heaviest hardware complexity resides on how iterative $H_i^{\dagger}$ operations are performed. For instance, both the LPI kernel and $H_i^{\dagger}$ operations handled in [7] yield the same $O(n^3)$ complexity; however, the LPI kernel significantly avoids matrix unitary transformations, an issue that demands a more sophisticated and complex control-path design. Also, the use of more CORDIC engines in [7, 8] affect data throughput as well as affecting symbol decoding latencies. On the other hand, $H_i^{\dagger}$ operations in [8] are alleviated through level-thresholding, another issue which incurs into BER performance degradation for sub-optimal hard-output SM-MIMO demodulation purposes.

6. Conclusions

In this work, a VLSI architecture for the sub-optimal hard-output SM-MIMO V-BLAST algorithm was proposed. The architecture was designed for operating with symbols drawn from 16-QAM modulators under AWGN Rayleigh fading channels with parameter $n = 4$. Simulation testing and hardware implementation on Altera Cyclone III FPGA devices validated the functionality of the V-BLAST architecture.

Acknowledgment

This work was supported by CONACYT (National Science and Technology Council, Reg. 332852/229015) under the supervision, revision, and sponsorship of ITESM Campus Guadalajara and ITESM Campus Estado de México universities.

References

[1] E. Biglieri, R. Calderbank, A. Constantinides et al., *MIMO Wireless Communications*, Cambridge University Press, 2007.

[2] M. El-Hajjar and L. Hanzo, "Multifunctional mimo systems: a combined diversity and multiplexing design perspective," *IEEE Wireless Communications*, vol. 17, no. 2, pp. 73–79, 2010.

[3] F. P. Fontán and P. M. Espiñera, *Modeling the Wireless Propagation Channel*, Wiley, 2008.

[4] Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Enhancements for Very High Throughput for Operation in Bands below 6GHz, IEEE P802.11ac/D1.0 Std, 2011.

[5] G. J. Foschini, "Layered space-time architecture for wireless communication in a fading environment when using multi-element antennas," *Bell Labs Technical Journal*, vol. 1, no. 2, pp. 41–59, 1996.

[6] P. W. Wolniansky, G. J. Foschini, G. D. Golden, and R. A. Valenzuela, "V-BLAST: an architecture for realizing very high data rates over the rich-scattering wireless channel," in *Proceedings of the URSI International Symposium on Signals, Systems, and Electronics (ISSSE '98)*, pp. 295–300, October 1998.

[7] Z. Guo and P. Nilsson, "A VLSI architecture of the square root algorithm for V-BLAST detection," *Journal of VLSI Signal Processing*, vol. 44, no. 3, pp. 219–230, 2006.

[8] F. Sobhanmanesh, S. Nooshabadi, and K. Kim, "A 212 Mb/s chip for 4 x 4 16-QAM V-BLAST decoder," in *Proceedings of the 50th Midwest Symposium on Circuits and Systems (MWSCAS)*, pp. 1437–1440, August 2007.

[9] G. H. Golub and C. F. V. Loan, *Matrix Computations*, The Johns Hopkins University Press, 3rd edition, 1996.

[10] D. Serre, *Matrices: Theory and Applications*, Springer, 2001.

[11] P. Cervantes, L. F. González, F. J. Ortiz, and A. D. García, "Partition-matrix theory applied to the computation of generalized-inverses for MIMO systems in Rayleigh fading channels," in *Linear Algebra*, pp. 137–162, 2012.

[12] P. Cervantes-Lozano, L. F. González-Pérez, and A. D. García-García, "Analysis of parallel sorting algorithms in K-best sphere-decoder architectures for MIMO systems," in *Proceedings of the International Conference on ReConFigurable Computing and FPGAs (ReConFig '11)*, pp. 321–326, Mexico, 2011.

[13] E. Milos and L. Tomas, *Digital Arithmetic*, Morgan Kauffmann Publishers, 2004.

[14] T. K. Paul and T. Ogunfunmi, "Wireless LAN comes of age: understanding the IEEE 802.11n amendment," *IEEE Circuits and Systems Magazine*, vol. 8, no. 1, pp. 28–54, 2008.