Journal of VLSI Signal Processing 44, 219–230, 2006

DOI: 10.1007/s11265-006-8536-8

A VLSI Architecture of the Square Root Algorithm for V-BLAST Detection

ZHAN GUO AND PETER NILSSON
Department of Electroscience, Lund University, SE-221 00, Lund, Sweden

Received: 20 August 2004; Revised: 4 November 2005; Accepted: 16 March 2006

Published online: 28 July 2006

Abstract. MIMO has been proposed as an extension to 3G and Wireless LANs. As an implementation scheme of MIMO systems, V-BLAST is suitable for the applications with very high data rates. The square root algorithm for V-BLAST detection is attractive to hardware implementations due to its low computational complexity and numerical stability. In this paper, the fixed-point implementation of the square root algorithm is analyzed, and a low complexity VLSI architecture is proposed. The proposed architecture is scalable for various configurations, and implemented for a 4×4 QPSK V-BLAST system in a 0.35 μm CMOS technology. The chip core covers 9 mm^2 and 190 K gates. The detection throughput of the chip depends on the received symbol packet length. When the packet length is larger than or equal to 100 bytes, it can achieve a maximal detection throughput of $128 \sim 160$ Mb/s at a maximal clock frequency of 80 MHz. The core power consumption, measured at 2.7 V and room temperature, is about 608 mW for 160 Mb/s data rate at 80 MHz, and 81 mW for 20 Mb/s at 10 MHz. The proposed architecture is shown to meet the requirements for emerging MIMO applications, such as 3G HSDPA and IEEE 802.11n.

Keywords: VLSI, ASIC, MIMO, BLAST, square root algorithm, fixed-point, CORDIC, 3G, HSDPA, wireless LAN

1. Introduction

As the requirements for wireless packet data services increase and the available radio spectrum becomes scarce, increasing spectral efficiency has been a major focus in researches for future wireless systems. By exploiting the spatial domain, the multiple-input multiple-output (MIMO) antenna systems provide an enormous increase in spectral efficiency compared to the conventional single antenna systems [1]. Thus, there has already been significant interest in building practical MIMO systems for both 3G and Wireless LANs to achieve high data rates over limited spectrum [2, 3].

The vertical Bell Labs layered space-time (V-BLAST) system is an efficient implementation scheme of MIMO systems, which demonstrated spectral efficiencies of 20–40 b/s/Hz at average signal-to-noise ratios (SNR) ranging from 24 to 34 dB in an indoor propagation environment [4]. It uses a vertically layered architecture in which the data bits are dispersed across layers in space-time. At the receiver, these layers are detected by an ordered successive interference cancellation (OSIC) technique, which nulls the interference by linearly weighting the received signal vectors with nulling vectors. The nulling vectors can be derived using either zero-forcing (ZF) [4] or minimum mean squared error (MMSE) criterion [5].

The main computational payload of the OSIC detector lies in the repeated pseudo-inverse computations of the channel matrix, which are required for

the calculation and optimal ordering of the nulling vectors. Moreover, the repeated pseudo-inverse computations might lead to numerical instability when a large number of antennas is employed. To address this, an efficient algorithm is proposed in [6], known as the square root algorithm. The computational complexity is reduced, as well as the numerical stability is significantly improved in the square root algorithm compared to the original OSIC algorithm.

The high data rates promised in V-BLAST systems demand a detector with high detection throughput. Compared to the dedicated hardware implementations, the general purpose DSP devices always have less throughput and consume more power due to the reconfigurable nature. The dedicated hardware solution is thus preferred to a software based DSP solution for V-BLAST detection. The objective of this paper is to present a low complexity VLSI architecture of the square root algorithm for V-BLAST detection. Although the proposed architecture is based on 4-transmit 4-receive antenna V-BLAST system with QPSK modulation for simplicity, it is straightforward to be extended to larger number of antennas and higher modulation size. This architecture is implemented and verified standalone in a 0.35 μm CMOS technology. It can also be used as a core in a baseband ASIC solution, or included as a dedicated hardware core in a DSP solution for V-BLAST systems.

In this paper, the V-BLAST system model and the square root algorithm is briefly described in Sections 2 and 3, respectively. As far as the practical algorithm implementation is concerned, the finite word-length effects are important issues to be considered, which are analyzed in Section 4. A low complexity VLSI architecture of the square root algorithm is proposed in Section 5. This architecture is implemented in a 0.35 μm five-metal-layer 3.3 V CMOS technology. The design methodology and implementation results are presented in Section 6. In Section 7, the implemented chip is shown to be feasible for emerging MIMO systems. The conclusion is presented in Section 8.

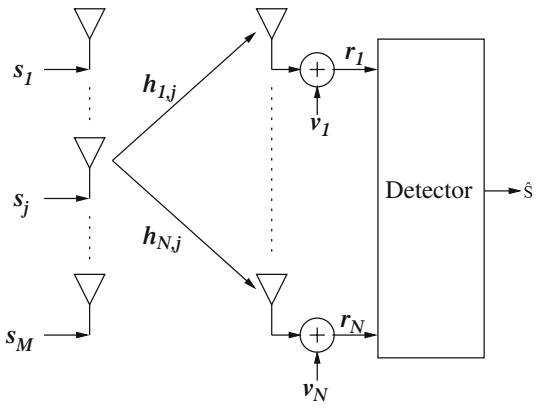

Figure 1. V-BLAST system model with M transmit and N receive antennas.

2. V-BLAST System

In this paper, vectors are denoted by bold lowercase letters, and matrixes are denoted by bold uppercase letters. Other notations used are as follows.

$(\cdot)^T$	Transpose of $(\cdot)$
$(\cdot)^*$	Hermitian of $(\cdot)$
$(\cdot)^{-1}$	Inverse of $(\cdot)$
$\|\|(\cdot)\|\|$	Euclidian norm of $(\cdot)$
$O((\cdot))$	Order of magnitude for $(\cdot)$
$H^{N\times M}$	Matrix of size $N \times M$
$(H)_j$	j-th column of matrix
$(H)_i$	i-th row of matrix

Consider a symbol synchronized and uncoded V-BLAST system with M transmit antennas and $N \geq M$ receive antennas, as shown in Fig. 1. The baseband equivalent model for a V-BLAST system is

$$r = Hs + v \tag{1}$$

where $s = [s_1, s_2, \cdots s_M]^T$ is the transmitted symbol vector, in which each component is independently drawn from a complex constellation such as QPSK and the symbol energy is normalized to unity, $r = [r_1, r_2, \cdots r_N]^T$ is the received symbol vector, $v = [v_1, v_2, \cdots v_N]^T$ is an i.i.d complex zero-mean Gaussian noise vector with variance σ^2 per real dimension. With the assumption of a block-fading and rich-scattering channel [1], H denotes the $N \times M$ channel matrix, whose elements h_{ij} represent the complex transfer functions from the j-th transmit antenna to the i-th receive antenna, and are all i.i.d complex zero-mean Gaussian with variance 0.5 per real dimension. The channel matrix is assumed to be perfectly known to the receiver in this paper.

The essential of V-BLAST detection is to solve [1]. The optimal solution is the maximum likelihood detector (MLD) [7]:

$$\hat{s} = \arg\min_{s} \|r - Hs\|^2 \tag{2}$$

MLD is an exhaustive search scheme, which can efficiently be implemented in well-known Viterbi algorithm. The complexity of MLD increases exponentially with the number of transmit antennas and the signal constellation size. As an alternative to MLD, the sphere decoder can approach the performance of MLD with reasonable complexity [8]. The MLD and sphere decoder have been shown to be feasible in current silicon technology [9, 10], though the implementation complexity is huge. The lower complexity V-BLAST detector using sub-optimal algorithm is thus still desired for low-end applications, that is the focus of this paper.

The MMSE solution to [1] is:

$$\hat{s} = (\alpha I + H^* H)^{-1} H^* r \tag{3}$$

with $\alpha = 2\sigma^2$. It is shown in [4] that nulling with ordering (OSIC) performs better than pure nulling using [3] alone. However, the MMSE-OSIC algorithm suffers from the repeated pseudo-inverse computations required by determining the ordered nulling vectors, which lead to a computational complexity of $O(M^4)$ for $M = N$ and numerical instability for large number of antennas. The square root algorithm is thus proposed in [6].

3. Square Root Detection Algorithm

The square root algorithm for V-BLAST detection successfully avoids the repeated pseudo-inverse computations and reduces the computational complexity to $O(M^3)$without degradation in BER performance. The algorithm calculates the QR decomposition of the augmented channel matrix first

$$\begin{bmatrix} H^{N\times M} \\ \sqrt{\alpha} I^{M\times M} \end{bmatrix} = QR = \begin{bmatrix} Q_\alpha^{N\times M} \\ \times \end{bmatrix} R^{M\times M} \tag{4}$$

where $\times$ denotes the entries that are not relevant at this time, then calculates $P^{1/2} = R^{-1}$. Once $P^{1/2}$ and Q_α are calculated, the repeated pseudo-inverse computations can be completely avoided. Furthermore, a recursion equation is proposed in [6] to avoid inverting R explicitly, that is preferred in hardware implementations. The algorithm is briefly summarized as below:

1. Compute $P^{1/2}$ and Q_α: For $i = 1, 2, \cdots N$,

$$\begin{bmatrix} 1 & \overline{(H)_i A_{i-1}^{M\times M}} \\ 0^{M\times 1} & A_{i-1}^{M\times M} \\ -e_i^{N\times 1} & B_{i-1}^{N\times M} \end{bmatrix} \Theta_i = \begin{bmatrix} \times & 0^{1\times M} \\ \times & A_i^{M\times M} \\ \times & B_i^{N\times M} \end{bmatrix} \tag{5}$$

$$A_0 = \beta I^{M\times M}, \quad B_0 = 0^{N\times M} \tag{6}$$

where e_i is the i-th unit vector of size N, Θ_i is any unitary transformation that block lower triangularizes the pre-array, and $\beta = 1/\sqrt{\alpha}$ is the square root of SNR per transmitted symbol. We thus have

$$P^{1/2} = A_N, \quad Q_\alpha = B_N \tag{7}$$

2. Determine the optimal ordering and nulling vectors: For $i = M,\ M-1, \cdots 1$,
   - Find the minimum length row of $P_i^{1/2}$ and permute it to be the last (i-th) row. Permute s accordingly.
   - Find an unitary Σ_i to block upper triangularize $P_i^{1/2}$:

$$P_i^{1/2}\Sigma_i = \begin{bmatrix} P_{i-1}^{1/2} & \times^{(i-1)\times 1} \\ 0^{1\times(i-1)} & p_i \end{bmatrix} \tag{8}$$

   - Update $Q_\alpha = Q_\alpha \Sigma_i$, then the nulling vector for the i-th transmitted signal is

$$w_i = p_i (Q_\alpha)_i^* \tag{9}$$

3. Perform nulling and cancellation: For $i = M,\ M-1, \cdots 1$,
   - Calculate

$$y_i = w_i \cdot r \tag{10}$$

   - Then the i-th transmitted signal in s is detected as the closest point in the signal constellation Ω.

$$\hat{s}_i = \arg\min_{s\in\Omega} \|s - y_i\|^2 \tag{11}$$

   - Cancel the interferences of the detected signal in the remaining received signals

$$r = r - \hat{s}_i (H)_i \tag{12}$$

Some lower complexity algorithms of $O(M^3)$, compared to the square root algorithm, have been proposed. They are all based on the direct QR decomposition of the channel matrix, but both [11] and [12] have a degradation in the BER performance due to the sub-optimal ordering algorithms employed. The decorrelating decision feedback (DDF) detection algorithm proposed in [13] is a modification to the square root algorithm, which reduces the computational complexity further due to employing the low complexity back substitution method. However, the divisions based back substitution method might be a problem of numerical stability in hardware implementations. Moreover, the back substitution is hard to be parallelized with the QR decomposition, and seems to be somewhat computationally inefficient [14].

The square root algorithm totally avoids pseudo-inversion and division computations, and uses unitary transformations for numerical stability as much as possible. Moreover, the large number of unitary transformations that zeros the predefined entries of given row vectors can be implemented by a sequence of Givens rotations [15]. The sequence of Givens rotations is suitable for CORDIC based hardware implementations in which only shifters and adders are involved [16]. Therefore, it is the BER optimality, numerical stability and lower complexity that make the square root algorithm attractive to hardware implementations for sub-optimal V-BLAST detection.

4. Fixed-Point Implementation Analysis

The quantization schemes of various variables of the square root algorithm have been analyzed in MATLAB. The least possible finite word length required for each variable is firstly determined in turn by assuming that other variables are in infinite precision, then the word length of each variable is further refined with all of variables considered in finite precision.

As in [17], let $q(w,f)$ denote a quantization scheme in which totally w bits are used, of which f bits are used for the fractional part of the value. With this quantization scheme, a value has $(w-f)$ bits of dynamic range and f bits of precision. A fixed-point model is implemented in MATLAB for the square root algorithm. The quantization schemes of all the variables are determined by simulations. All the simulations are performed until 100 frame errors are incurred, or at most 1,000 frames are transmitted with 4,096 bits per frame. It should be noted that the quantization analysis is related to the VLSI architecture proposed later, and the quantization results are dependent on the system model stated above.

The main computational payload of the square root algorithm stems from using unitary transformations to calculate $P^{1/2}$ and Q_α. The quantization of $P^{1/2}$ and Q_α is thus crucial to the algorithm behavior, as it determines the computational accuracy of the pseudo-inverse of channel matrix and optimal ordering. A small wordlength might result in poor performance, though a large wordlength might cost more hardware. However, the hardware overhead of processing $P^{1/2}$ and Q_α is relatively small in the proposed VLSI architecture as shown in the next section. The performance is therefore the most important factor that has to be considered. Based on our simulations, using the $q(16,8)$ scheme for $P^{1/2}$ and Q_α seems to be the optimal trade-off between hardware complexity and BER performance within the investigated range of SNR.

The quantization schemes of all the variables are summarized in Table 1. Floating-point and fixed-point simulation results are shown in Fig. 2. It is clear that the quantization schemes employed in Table 1 perform well compared to the infinite precision schemes within the investigated range of SNR.

5. VLSI Architecture

Based on the square root algorithm, the proposed VLSI architecture consists of four modules, as shown in Fig. 3. The INPUT module stores the values of H and β, which are assumed to be known to the detector. The PINV module uses complex

Table 1. Quantization schemes summary.

No.	Variable	Quantization scheme
1	$P^{1/2}$, Q_α	q(16,8)
2	β	q(16,8)
3	H	q(5,2)
4	p_i, $(Q_\alpha)_i$	q(13,8)
5	w_i	q(12,8)
6	r	q(11,7)
7	y_i	q(10,7)

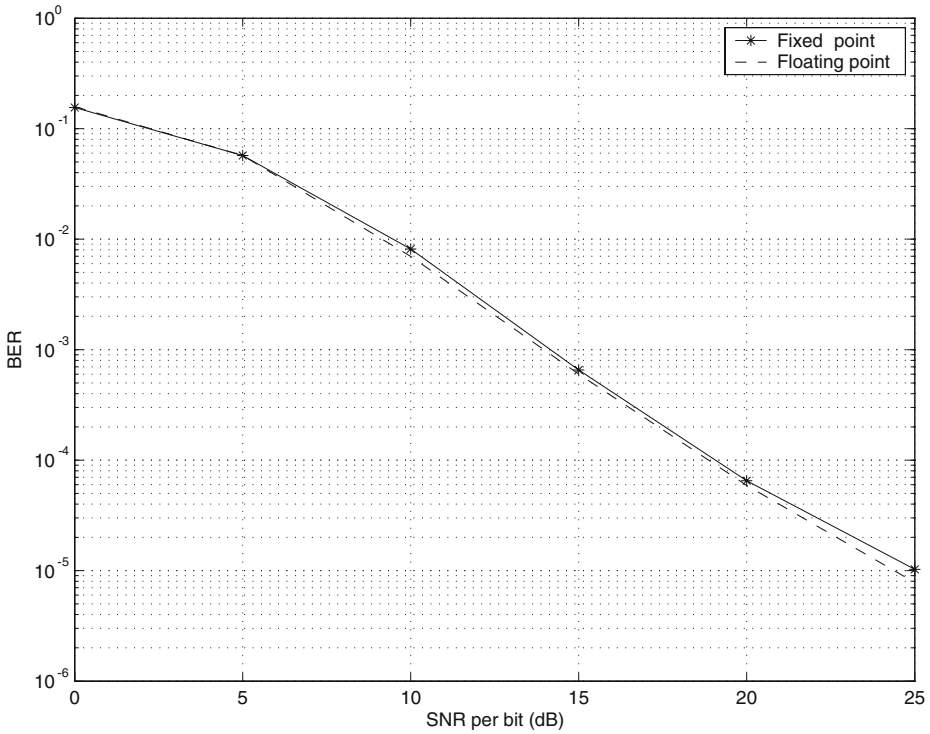

Figure 2. Floating-point vs. fixed-point simulation results.

Givens rotations to calculate $P^{1/2}$ and Q_α. Then, the SORT module also uses complex Givens rotations to calculate the nulling vectors w_i and the optimal ordering *Ord*. The NULL module performs nulling and cancellation, and finally outputs the detected symbols $\hat{s}$. Each module has an independent timing controller, while a timing controller exists on the top level.

In addition, an off-chip memory with appropriate size is assumed to buffer the received symbol packets *r*. The memory size is dependent on the specific applications. The INPUT module covers very few area in the implemented detector (refer to Fig. 12). It mainly consists of three 17×16 bits single-port RAMs. Each RAM is used to buffer a block of H and β. The detector can thus process three continuous symbol packets.

5.1. QR-array vs. Single Processor

In order to perform the QR decomposition of the augmented channel matrix, a large number of complex Givens rotations needs to be performed in the PINV module and the SORT module. This is a well-established technique in applications of adaptive filtering, and known as QR-array [14, 18]. Specifically, calculating $P^{1/2}$ and Q_α in the square root algorithm is similar to the inverse QR algorithm for adaptive channel equalization, which is typically implemented in the form of a triangular array and a linear section connected together [14].

As shown in Fig. 4, the QR-array requires two types of cells, often referred to as boundary and internal cells. In the boundary cell, the length ρ and the angle θ towards the x -axis of a vector (x, y) are computed, which is referred to as vectoring mode. In the internal cell, a vector (x, y) is rotated by an angle

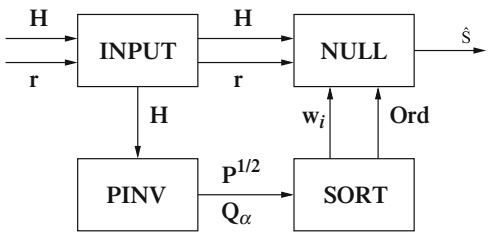

Figure 3. Block diagram of the proposed VLSI architecture.

θ to obtain a new vector (x', y'), which is referred to as rotation mode. Since a CORDIC unit can be operated in the vectoring mode as well as the rotation mode [16], the boundary and internal cell in QR-array can be merged to a single cell based on CORDIC. Furthermore, a supercell is proposed in [19] and adopts CORDIC to implement both vectoring mode and rotation mode for complex number.

However, the number of processors used by the conventional triangular QR-array is often too high and the throughput far in excess of the system requirements. To address this, the triangular array is often mapped to a reduced number of time-shared processors in a linear array, as shown in Fig. 4, using folding and interleaving techniques in the context of adaptive beamforming [18–20].

Note that the QR-array, whether in triangular form or in linear form, is applied to the received symbol data to compute the weight vectors in the case of adaptive beamforming. In the case of V-BLAST detection instead, the QR-array needs to be applied to the estimated channel matrix data. Since the channel matrix H is updated at a much slower rate than the symbol r, with the assumption of block-fading environment, using QR-array to decompose H may lead to an inefficiency in computational capabilities. This motivates us to use a single processor instead of a processor array. The question is how to design the single processor and use it to decompose the whole augmented channel matrix.

5.2. Decomposition of Channel Matrix

The single processor based PINV module consists of three units, as shown in Fig. 5. The multiplier-accumulation (MAC) unit calculates $\underline{(H)_i} A_{i-1}$ in [5]. The unit has a separate output for the complex-number multiplier, since the accumulation operation is not involved in the calculation of $\underline{(H)_1} A_0 = \underline{(H)_1} \beta$. The buffer between the MAC and the supercell is implemented by using two single-port 16×32 bits RAMs, which are operated as a dual-port RAM in order to reduce power consumption. The supercell is used to perform complex Givens rotations, and to calculate $P^{1/2}$ and Q_α. For a detailed discussion on the supercell for complex Givens rotations, the reader is referred to [19]. The employed supercell is a modification to [19] according to the square root algorithm, in which three MUXs are related to the

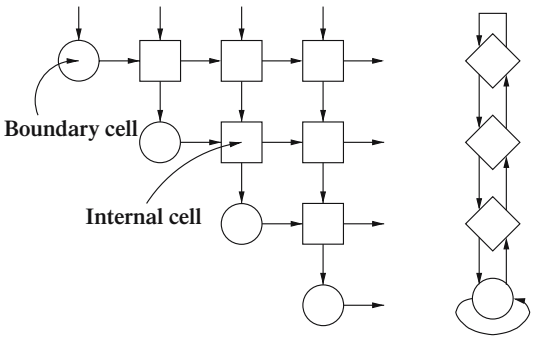

Figure 4. Triangular array (*left*) and linear array (*right*).

subsidiary vector $d_i = [1, 0^{1\times M}, -e_i^{1\times N}]^T$ in the pre-array of (5).

Three CORDIC units in the supercell have the same structure, as shown in Fig. 6. The CORDIC unit is fully pipelined, and each pipelining stage can independently be switched between the vectoring mode and rotation mode as controlled by the signal *Vec_en*. To satisfy an m-bit precision CORDIC operation, $m+1$ iterations are needed and the datapath wordlength has to be $m + 2 + \log_2 m$ [21]. For the 16-bit precision in calculating $P^{1/2}$ and Q_α, 17 stages of pipelining and 22-bit datapath wordlength are thus needed in the CORDIC unit. Furthermore, the scaling operation required by CORDIC is implemented by a fixed coefficient multiplication, which covers three stages of pipelining. The computational latency of a CORDIC unit is thus $\tau = 20$ clock cycles.

To guarantee the correct vectoring operations in CORDIC, the function of the pre-computation unit in the supercell is negating both x and y components if $x < 0$, effectively doing a rotation of radian π, and limiting the vector (x, y) in quadrant 1 or 4 when operated in vectoring mode. The pre-computation unit and three MUXs in the supercell is merged into the first stage of θ-CORDIC and ϕ-CORDIC, respectively. The latency of the supercell is thus $2\tau = 40$ clock cycles.

Every τ clock cycles, the supercell accepts a column vector accompanied with a set of *Vec_en* signals, where each *Vec_en* signal corresponds to an item of the column vector. When the *Vec_en* signals are all high, each element of the inputed column vector is independently operated in the vectoring mode. This scheme is useful in calculating the length of a complex column vector, as used in the

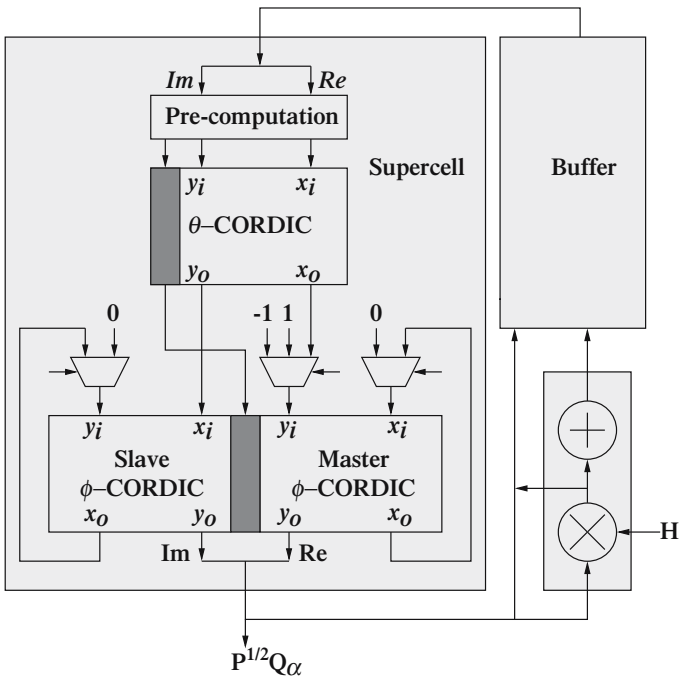

Figure 5. Block diagram of PINV module.

SORT module later. When the first element (leader) of the column vector is enabled by Vec_en while the other elements (follower) not, only the leader is operated in vectoring mode, while the followers are operated in rotation mode in which the rotation angle is determined by the leader. This scheme is used to calculate $P^{1/2}$ and Q_α.

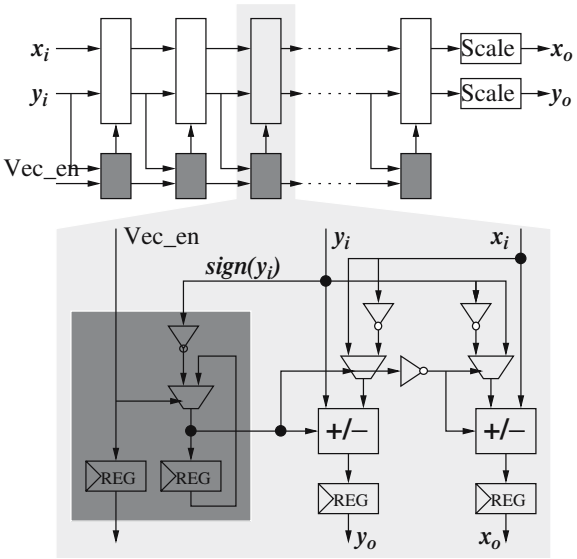

Figure 6. Block diagram of CORDIC unit.

Figure 7 illustrates how to use a single supercell to implement the recursive [5] in the square root algorithm. The complex elements of the pre-array are passed by column to the supercell. The leader of each column is operated in vectoring mode, while the followers are rotated by the leader. At the first τ period, the column 1 is passed to θ -CORDIC. At the second τ period, the updated column 1 is buffered in one of ϕ -CORDIC, while the column 2 is passed to θ -CORDIC. After passing θ -CORDIC, the leaders of the updated column 1 and 2 are all real numbers. Then at the third τ period, the column 1 is output from the supercell with the leader becoming zero, while the column 2 will continue to be rotated with the following column 3. As shown in Fig. 7, this process is repeated for the column 4 and the subsidiary column d until the new column 1 is input, which results in an iteration period of 5τ for updating the pre-array. The leaders of the new columns are calculated in the MAC unit. After four iterations, the outputs of the supercell become $P^{1/2}$ and Q_α, which are to be pipelined into the SORT module. The

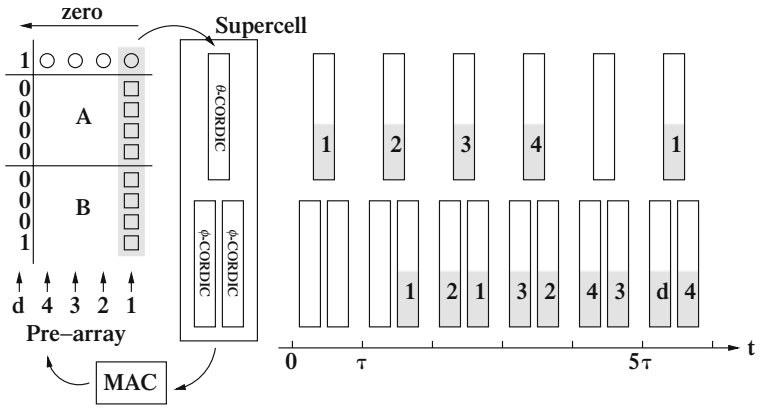

Figure 7. Illustration of one iteration in PINV module.

latency in obtaining $P^{1/2}$ and Q_α is thus $4 \times 5\tau = 400$ clock cycles.

5.3. Calculation and Ordering of Nulling Vectors

The function of the SORT module consists of calculating and ordering the row length of $P^{1/2}$, block upper triangularizing $P^{1/2}$, and generating the nulling vectors. The upper-triangularization of $P^{1/2}$ is also involved in a large number of complex Givens rotations. The structure of the SORT module is thus similar to the PINV module, as shown in Fig. 8. It consists of a supercell, a buffer implemented by a 32×32 bits single-port RAM, and a W unit used to generate the nulling vectors w_i. The W unit is simpler than the MAC unit in the PINV module, since it actually consists of two real-number multipliers as indicated in (9). In addition, an ordering unit is needed to sort the row length of $P_i^{1/2}(i = 1, 2, 3)$ and calculate the optimal ordering *Ord*, but the calculation of the row length is left to the supercell.

As shown in Fig. 9, $P^{1/2}$ is updated every 400 clock cycles. Once $P^{1/2}$ is received by the SORT module, the following 204 clock cycles are needed exclusively for the nulling vectors w_1 and w_2. During the 204 cycles, the row lengths of $P_1^{1/2}$ and $P_2^{1/2}$, corresponding to w_1 and w_2, respectively, can be calculated in θ -CORDIC using vectoring mode. Note that the ordering of $P_4^{1/2}$ is not needed, since $P_4^{1/2}$ is actually a scaler. The question is how to calculate the row length of $P_3^{1/2}$ corresponding to w_3, since the θ -CORDIC has been occupied by w_1 and w_2 at the same time. A possible solution is to force the update period of $P^{1/2}$ to be $400 + 204 = 604$ cycles. This will lead to a computational inefficiency in the PINV module. Our solution is to add the pre-computation unit to the master ϕ -CORDIC in the SORT module, and to activate its vectoring mode, as shown in Fig. 8. The row length of $P_3^{1/2}$ can thus be calculated in the master ϕ -CORDIC, leaving the θ -CORDIC for the newly received $P^{1/2}$.

5.4. NULL Module and Detection Throughput

The NULL module has a flexible structure in linear array as shown in Fig. 11. The structure consists of four fully pipelined PE units in order to make maximal use of the detection throughput provided in a four antennas system. The PE unit in the first three stages consists of a MAC, a slicer, a small buffer implemented in register banks, a subtractor and a pseudo-multiplier used to calculate $\hat{s}_i(H)_i$. The pseudo-multiplier can be implemented in hard-wired

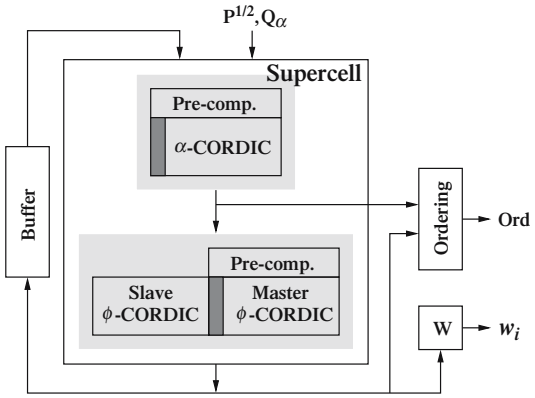

Figure 8. Block diagram of SORT module.

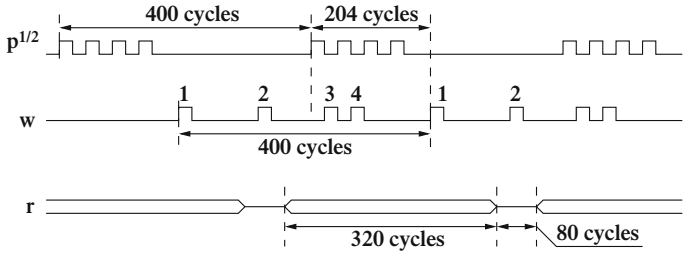

Figure 9. Timing diagram when the update period of $P^{1/2}$ is 400 clock cycles.

shifting and addition, since $\hat{s} \in \Omega$ is the fixed coefficient. The last stage of PE has only a MAC and a slicer, since the cancellation operation is not required in this stage. The ordering unit in the NULL module is used to sort the detected symbols $\hat{s}$ by using the ordering *Ord* which is calculated in the SORT module.

The proposed NULL module is essentially a symbol-serial structure, since the received symbols r_i $(i = 1, 2, 3, 4)$ in vector r are processed one by one. Consequently, the detection throughput would be determined by the symbol rate alone, if the update period L of $P^{1/2}$ and Q_α could be omitted.

However, in the proposed VLSI architecture for V-BLAST detection, L is also a key to the detection throughput due to the fact that L depends on the received packet length. As described above in the PINV module, the minimum of L is $L_{min} = 400$ clock cycles and corresponds to a symbol packet of 100 bytes (400×2 bits due to QPSK employed). When the received packet length is much larger than 100 bytes, i.e., $L \gg L_{min}$, the received symbols r can be processed continuously, as shown in Fig. 10. The detector can thus attain a maximally possible throughput $f_{max} = 2f_c$, where f_c is the clock frequency. As the received packet length decreases to 100 bytes, f_{max} is also reduced to $\frac{320}{400} \times 2f_c = 1.6f_c$ due to the timing overlaps of 80 cycles, as shown in Fig. 9. Finally, when the packet length is smaller than 100 bytes, an off-chip buffer might be needed for the received signals to make the detector functional. The discussion on the buffer size is beyond the scope of this paper, since it depends on the specific application context.

The detection throughput of the detector can be improved further by parallelizing some number of NULL modules in the proposed VLSI architecture. On the other hand, when the throughput requirement is relaxed, the NULL module can be operated with a single PE unit to reduce the chip area, by using feedback mode as indicated in dash line in Fig. 11. The NULL module covers about one third of the chip area (refer to Fig. 12). The power estimation using Synopsys Power Compiler shows that the NULL module contributes to about 40% of total dynamic power consumption of the detector. It is estimated that both area and dynamic power of the chip could be reduced by about 25% if a single PE is employed in the NULL module, but the detection throughput would be reduced to one fourth.

6. Implementation Results

The proposed VLSI architecture is described in Verilog HDL at RTL level, and synthesized in Synopsys Design Compiler. Then a conventional

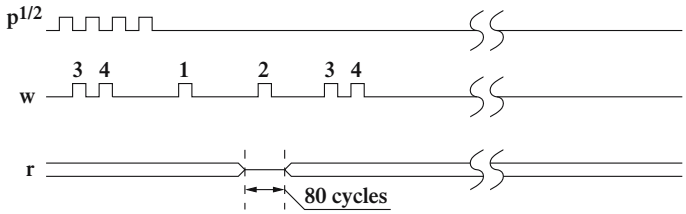

Figure 10. Timing diagram when the update period of $P^{1/2}$ is much larger than 400 clock cycles.

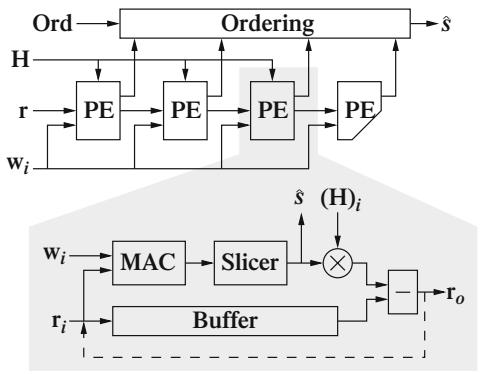

Figure 11. Block diagram of NULL module.

standard cell design flow is employed [22]. The floorplanning, clock tree synthesis, placement and routing are done in Cadence Silicon Ensemble using timing-driven placement with forward constraints annotated. The RTL, annotated pre-layout gate level and post-layout gate level netlists are all verified against the same test vectors generated from the MATLAB fixed-point model. Furthermore, the pre-layout and post-layout timing are verified in Synopsys PrimeTime with net and cell delays back annotated in SDF format.

The V-BLAST detector is fabricated through the EUROPRACTICE *mini@sic* program using AMIS 0.35 μm 5ML CMOS technology [23]. Figures 12 and 13 show the layout and the die photo of the chip, respectively. Table 2 lists the features of the V-BLAST detector. The chip is functional at 2.7 V with a 80 MHz clock at room temperature. The average core power consumption, measured at 2.7 V and room temperature, is about 608 mW at 80 MHz, and 81 mW at 10 MHz.

7. Discussions

It is clear from Fig. 12 that the chip is almost equally divided by the PINV, SORT and NULL modules. This fact shows it necessary, in terms of area and power consumption, to employ a single processor instead of a processor array in the PINV and SORT modules. Furthermore, as shown below, the proposed architecture is also efficient in detection throughput for some potential applications of V-BLAST systems.

Firstly, with the PINV and SORT modules not affected, it is straightforward to extend the NULL module for higher modulation size, and hence attain higher data rate. The following discussions on the detection throughput are thus limited to uncoded data rate, while not taking into account the possible loss in data rate due to coding.

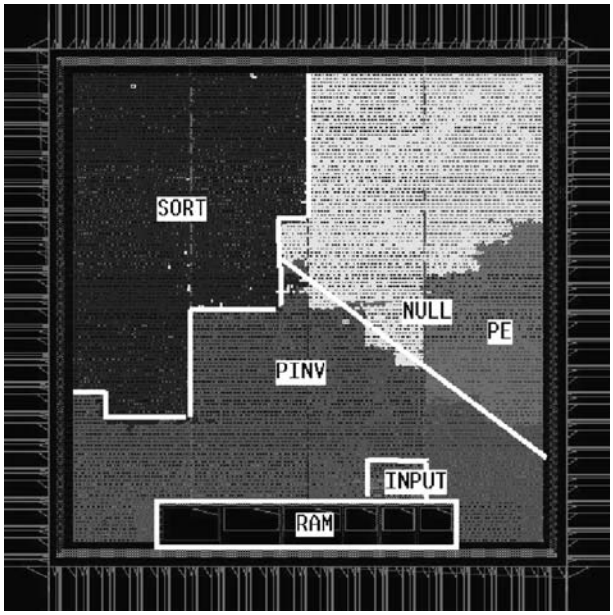

Figure 12. Layout of V-BLAST detector.

As analyzed in Section 5.4, the implemented chip can attain a maximally possible detection throughput of 128 $\sim$ 160 Mb/s at maximal clock frequency of 80 MHz, assuming that the received packet length is larger than or equal to 100 bytes. This condition is trivial to meet in applications of Wireless LANs, e.g., IEEE 802.11a, in which the packet length varies between 1 and 4,095 bytes [24]. The emerging IEEE 802.11n, which is to be backward compatible to 802.11a, will use the MIMO technique to boost a data rate near 150 Mb/s at the physical layer (100 Mb/s at the MAC layer) [3]. The implemented chip proves this potential. For the cases in which the packet length is smaller than 100 bytes, the detection throughput will degrade. A solution is to estimate the channel matrix once every a few packets. Another possibility is that the maximally attainable throughput is fundamentally not needed for those smaller length of packets in real applications.

Another emerging MIMO application is the UMTS HSDPA, in which a maximal data rate of 21.6 Mb/s is required for a $M = N = 4$ system with QPSK modulation [2]. The implemented chip can meet this requirement with a clock at about 11 MHz, which translates to a detection throughput of 17.6 $\sim$ 22 Mb/s. An alternative is to use the reduced-complexity NULL module based on a single PE unit, as described in Section 5.4. When the proposed architecture with the reduced-complexity NULL module operates at 80 MHz clock frequency, it is estimated that a maximal detection throughput of 32 $\sim$40 Mb/s can be achieved, which still meets the requirements of the HSDPA.

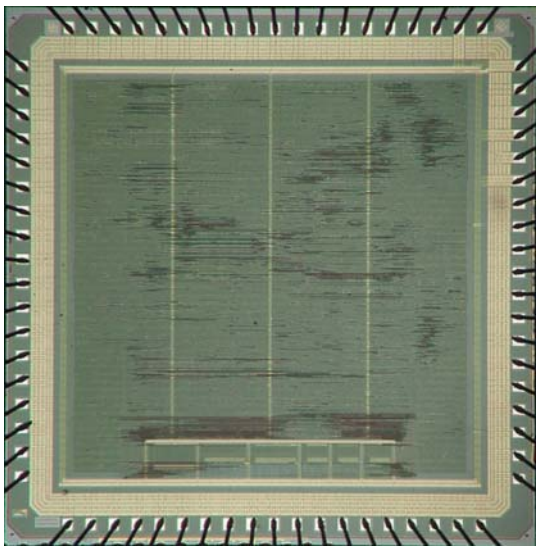

Figure 13. Photomicrograph of V-BLAST detector.

Table 2. Features of V-BLAST Detector ($M = N = 4$, QPSK).

Process	0.35 μm 5ML CMOS
Supply voltage	3.3 V, 2.7 V
Package	JLCC (84 pins)
Max. clock frequency	80 MHz
Latency in obtaining $P^{1/2}$ and Q_α	400 clock cycles
Gate count	190 K
RAM size	$<$ 3 Kbit
Core area	9 mm^2
Core	608 mW (2.7 V @80 MHz) power consumption & 81 mW (2.7 V @10 MHz)

8. Conclusion

In this paper, the fixed-point implementation effects are analyzed on the square root algorithm for V-BLAST detection. Instead of the complex QR-array that employs a large number of processors, a single processor based VLSI architecture is proposed to reduce the implementation complexity for V-BLAST detection. The proposed architecture is implemented in a 0.35 μm CMOS technology for a 4-transmit 4-receive antenna system with QPSK modulation. The detection throughput of the chip depends on the received symbol packet length. When the packet length is larger than or equal to 100 bytes, the chip can achieve a maximal detection throughput of 128 $\sim$ 160 Mb/s at a maximal clock frequency of 80 MHz. The core power consumption, measured at 2.7 V and room temperature, is about 608 mW for 160 Mb/s data rate at 80 MHz, and 81 mW for 20 Mb/s at 10 MHz. The proposed architecture is scalable for various configurations, and shows potentials for emerging MIMO applications.

Acknowledgment

This work was supported by the INTELECT program under SSF.

References

1. G. J. Foschini, "Layered Space-time Architecture for Wireless Communication in Fading Environments when using Multiple Antennas," *Bell Labs. Tech. J.*, vol. 1, no. 2, pp. 41–59, Autumn 1996.
2. 3GPP, "Physical Layer Aspects of UTRA High Speed Downlink Packet Access," 3GPP TR25.848 V4.0.0(2001–2003).
3. V. K. Jones, G. Raleigh and R. van Nee, "MIMO answers high-rate WLAN call," http://www.eetimes.com/story/OEG20031231S0008.
4. 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 *Proc. ISSSE*, Pisa, Italy, Sept. 1998.
5. S. Bäro, G. Bauch, A. Pavlic and A. Semmler, "Improving BLAST Performance Using Space-time Block Codes and Turbo Decoding," in *IEEE Global Telecommunications Conference*, 2000.
6. B. Hassibi, "An efficient square-root algorithm for BLAST," http://mars.bell-labs.com/.
7. J. G. Proakis, *Digital Communications*, 4th ed. McGraw-Hill, 2001.
8. M. O. Damen, A. Chkeif and J.-C. Belfiore, "Lattice Code Decoder for Space-time Codes," IEEE Communications Letters, vol. 4, no. 5, pp. 161–163, May 2000.
9. D. Garrett, L. Davis, S. ten Brink, and B. Hochwald, "APP Processing for High Performance MIMO Systems," in *Proceedings of the IEEE 2003 Custom Integrated Circuits Conference (CICC)*, San Jose, California, Sept. 2003.
10. Z. Guo and P. Nilsson, "A VLSI Architecture of the Schnorr-euchner Decoder for MIMO Systems," in *IEEE 6th CAS Symposium on Emerging Technologies: Frontiers of Mobile and Wireless Communication*, Shanghai, China, June 2004.
11. M. O. Damen, K. Abed-Meraim, and S. Burykh, "Iterative QR detection for BLAST," *Wireless Personal Communications*, vol. 19, pp. 179–191, 2001.
12. D. Wubben, R. Bohnke, J. Rinas, V. Kuhn, and K. D. Kammeyer, "Efficient Algorithm for Decoding Layered Space-time Codes," *Electronics Letters*, vol. 37, pp. 1348–1350, 2001.
13. W. Zha and S. D. Blostein, "Modified Decorrelating Decision-feedback Detection of BLAST Space-time System," in *IEEE International Conference on Communications*, vol. 1, pp. 335–339, June 2002.
14. S. Haykin, *Adaptive Filter Theory*, 4th ed. Prentice-Hall, 2002.
15. G. H. Golub and C. F. V. Loan, *Matrix Computations*, 3rd ed. John Hopkins University Press, 1996.
16. Y. H. Hu, "CORDIC-based VLSI Architectures for Digital Signal Processing," *IEEE Signal Processing Magazine*, July 1992.
17. Z. Wang, H. Suzuki, and K. K. Parhi, "Finite Wordlength Analysis and Adaptive Decoding for Turbo/Map Decoders," *Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology*, vol. 29, pp. 209–221, 2001.
18. R. Walke, R. Smith, and G. Lightbody, "Architectures for adaptive weight calculation on ASIC and FPGA," in *33rd Asilomar Conference on Signals, Systems and Computers*, 1999.
19. C. Rader, "VLSI Systolic Arrays for Adaptive Nulling," *IEEE Signal Processing Magazine*, vol. 13, no. 4, pp. 29–49, 1996.
20. G. Lightbody, R. Walke, R. Woods, and J. McCanny, "Linear QR Architecture for a Single Chip Adaptive Beamformer," *Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology*, vol. 24, pp. 67–81, 2000.
21. J. Valls, M. Kuhlmann, and K. K. Parhi, "Evaluation of CORDIC algorithms for FPGA design," *Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology*, vol. 32, 2002, pp. 207–222.
22. Z. Guo, "A Short Introduction to ASIC Design Flow in AMIS Library," online available: http://www.es.lth.se/home/zgo/, April 2003.
23. http://www.imec.be/europractice/europractice.html.
24. IEEE, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," IEEE Std 802.11a/D7.0–1999.

Dr. Zhan Guo received the B.Sc. degree in Electrical Engineering from Xi'an Jiaotong University, Xi'an, China in 1996, the M.Sc. degree in Microelectronics from Tsinghua University, Beijing, China in 1999, and the Ph.D. degree in Electrical Engineering from Lund University, Lund, Sweden in 2005. His main research interests include digital ASIC design for wireless systems and system-on-chip design.
zhan.guo@es.lth.se

Peter Nilsson reached the Master of Science degree in Electrical Engineering at Lund Institute of Technology, Lund University. In May 1992, he reached the degree Licentiate of Engineering and in May 1996 he reached the degree Doctor of Philosophy in engineering both at Lund University. After the Ph.D. degree he began as an Assistant Professor at Department of Applied Electronics (now Department of Electroscience), Lund University, Lund, Sweden. In November 1997 he became Associate Professor at the same department and in December 2003, the degree "Docent" was awarded. Peter is also the Program Manager for Socware Research & Education, a national program for research and Master's education on System-on-Chip. He is also an Associate Editor for IEEE Transactions on Circuits and Systems I and a member of the VLSI Systems and Applications Technical Committee in IEEE Circuits and Systems Society. Peter's main interest is in the field of implementation of digital circuits.
peter.nilsson@es.lth.se