

# Enhanced Dual Strategy based VLSI Architecture for Computing Pseudo Inverse of Channel Matrix in a MIMO Wireless System

<sup>1</sup>Zahid Khan, <sup>1,2</sup>Tughrul Arslan, <sup>1</sup>John S. Thompson, <sup>1,2</sup>Ahmet T. Erdogan

<sup>1</sup>System Level Integration Group  
School of Engineering and Electronics  
The University of Edinburgh,  
Edinburgh, Scotland, UK

<sup>2</sup>Institute of System Level Integration  
The ALBA Campus,  
Livingston, Edinburgh,  
Scotland, UK

## Abstract

*Multiple Input Multiple Output (MIMO) wireless technology involves highly complex signal processing which is directly related to increased power and area consumption in VLSI architecture. This paper proposes an enhanced dual strategy based VLSI architecture developed for computing the pseudo inverse of augmented channel matrix used in MIMO systems. The architecture concurrently addresses algorithmic optimization of number of multipliers while at the same time allowing for intelligent selective clock gating to disable the clock to those portions of the architecture that remain inactive during period of computation. Results indicate overall 36% power and 31% area reduction compared to previous architecture without degrading the BER performance.*

## 1. Introduction

Multiple Input - Multiple Output (MIMO) wireless communication is a new technology that promises to remove the limits of wireless networks by providing spectral efficiency near Shannon's bound [1]. MIMO multiplies range, reliability and data speed of existing wireless systems [2]. Because of its advantages, MIMO is entering into almost every wireless network such as CDMA2000, and WCDMA as an example [2]. However, such benefits of MIMO come at the expense of highly complex signal processing which directly contributes to high power consumption [3].

Communication is becoming more wireless and portable. The weight and size are the bottlenecks of portable electronic systems. Power supplies are a major, if not dominant factor contributing to the weight and size of portable electronic systems and these are directly affected by the power dissipation due to electronic circuits.

In CMOS, sources of power consumption include short circuits, leakage currents and switching. The switching or dynamic power equation is described as

$$P = kC_L V^2 f \quad [4]$$

where  $k$  represents the switching activity factor,  $C_L$  the total physical capacitance,  $V$  the supply voltage and  $f$  the frequency of operation. Algorithmic power optimization includes reduction of both physical capacitance and switching activity factor. Physical capacitance can be reduced by reducing the area of hardware through efficient implementation [5]. Switching activity reduction either comes from area reduction that reduces the number of nodes or from reducing the switching frequency of nodes. One of the algorithmic optimizations is reducing redundancy [6] from a design. By reducing the redundant operations or hardware, unnecessary switching of the clock as well as other signals can be avoided, thereby saving power.

VBLAST (Vertical Bell Labs Layered Space Time) is a MIMO detection algorithm [7] that provides a good trade-off between BER (Bit Error Rate) performance and computational complexity compared to its counter parts. Zero Forcing (ZF) and Minimum Mean Square Error (MMSE) detectors [8] are computationally less expensive than VBLAST; however, they provide inferior BER performance compared to VBLAST. The optimal solution is maximum likelihood (ML) [8] detection which provides best BER performance. However, it is highly expensive regarding computational complexity. This increases exponentially with the number of antennas used and is prohibitively high for antennas more than 4. Therefore, the ML algorithm cannot be implemented on mobile platforms due to its high overhead of area and power [3].

VBLAST can provide BER performance close to ML at a computational complexity much less than ML [8]. In VBLAST itself, the bottlenecks are repeated pseudo inverse calculation required to compute optimal ordering and nulling vectors. This repeated computation leads to numerical instability in hardware implementation and complexity which can be reduced using alternative algorithms.

The pseudo inverse can be computed using the singular value decomposition method (SVD) [9]. However, pseudo inverse computation through SVD is expensive both in silicon and power consumption. For equal transmit and receive antennas ( $M=N$ ), the

complexity of pseudo inverse through SVD in MMSE-VBLAST is  $(27/4)M^4$  [10]. The complexity grows as the fourth power of  $M$  which is quite huge. The square root algorithm [10] not only computes pseudo inverse but also avoids repeated pseudo inverse computation and reduces the computational complexity of VBLAST to  $O(M^3)$  without degrading performance [10].

A VLSI architecture for computing the pseudo inverse module through the square root algorithm has been devised in [11] in which a 2-CORDIC based supercell has been used. The architecture presented in [11] is an improved version of the architecture presented in [12] which is a straight forward implementation without regard to area and power optimizations. In this paper, two strategies have been suggested to the architecture proposed in [11]. The first is the use of two multipliers in the MAC (Multiply and Accumulate) unit instead of the conventional four multipliers used to carry out multiplication and accumulation of complex numbers. The second strategy is based on disabling the clock to some of the portions of the architecture that remain inactive for some time during period of computation.

## 2. MIMO System Model and Square Root Algorithm for VBLAST

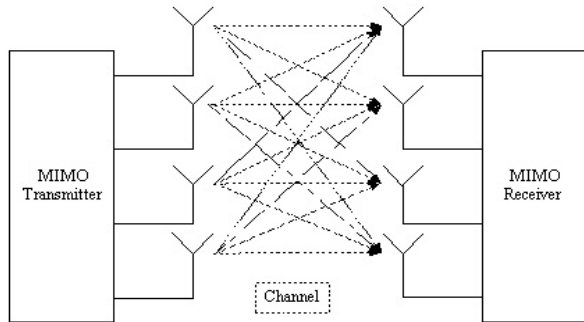


Figure 1: (MIMO System Model)

In MIMO communication systems, more than one antenna is used at the transmitter to transmit symbols and more than one antenna is used at the receiver to receive them. In the diagram of Figure 1, spatial multiplexing is used and  $M$  transmit antennas transmit  $M$  symbols simultaneously while each symbol is received by the  $N$  receive antennas. Each symbol transmitted is received by all the receiving antennas thus making multiple channel paths. These paths, if combined, make a matrix of channel elements. Each symbol makes  $N$  channel paths and is received by  $N$  receive antennas. Since there are  $M$  symbols transmitted simultaneously, the channel becomes a  $N \times M$  matrix.

If  $s = [s_1, s_2, s_3, s_4, \dots, s_M]^T$  denotes the symbol vector transmitted,  $H$  denotes the  $N \times M$  channel matrix between the receive and transmit antenna array, and  $v$  denotes the AWGN independent and identically distributed noise vector, then the corresponding receive vector  $r$  at the input of the MIMO receiver is given by

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

To recover the transmitted symbol vector  $s$ , it is necessary to invert the channel matrix  $H$ . The inversion can be done depending upon the detection method used. For Zero Forcing, the channel matrix is simply inverted and the detector output is given by (2)

$$z = s + H^\dagger v \quad (2)$$

where  $\dagger$  is the pseudo inverse. For MMSE, the channel matrix is augmented by the noise variance ( $\alpha$ ) and the detector output is as in (3)

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

where  $*$  represents complex conjugate transpose.

In VBLAST, successive nulling and cancellation is used to detect the transmitted symbols. The channel matrix is first inverted and then reordered to detect that symbol first which has the highest post detection Signal to Noise ratio (SNR). This corresponds to the row of the inverted channel matrix having minimum Euclidean distance. The symbol after detection is subtracted from the received symbol vector. The corresponding column of the  $H$  matrix is zeroed down and the process is repeated with the deflated channel matrix until all the symbols are detected. In this research, MMSE is used for channel inversion. The pseudo inverse of a generic matrix  $H$  is given by

$$H^+ = (H^* H)^{-1} H^* = R^{-1} Q^* \quad (4)$$

The pseudo inverse can be computed using either singular value decomposition (SVD) or QR decomposition. The square root algorithm [10] is developed for MMSE-VBLAST and computes the QR decomposition of the augmented channel matrix.

$$\begin{bmatrix} H^{N \times M} \\ \sqrt{\alpha} I^{M \times M} \end{bmatrix} = QR = \begin{bmatrix} Q_a^{N \times M} \\ x \end{bmatrix} R^{M \times M} \quad (5)$$

here  $x$  denotes the entries that are not relevant. The algorithm first decomposes the channel matrix into QR  $a_r + ja_i$  and then computes  $P^{1/2} = R^{-1}$ . Once  $Qa$  and  $P^{1/2}$  are computed, the repeated pseudo inverse can be avoided. Both  $Qa$  and  $P^{1/2}$  are computed together in a

series of unitary transformations. The algorithm is described below:

Compute  $Qa$  and  $P^{1/2}$  using equation (6).

$$1. \quad B \Theta_i = X \quad (6)$$

$$B = \begin{bmatrix} 1 & H_i P_{|i-1}^{1/2} \\ 0^{M \times 1} & P_{|i-1}^{1/2} \\ -e_i^{N \times 1} & B_{i-1} \end{bmatrix} \quad X = \begin{bmatrix} x & 0^{1 \times M} \\ x & P_i^{1/2} \\ x & B_i \end{bmatrix}$$

Here  $i$  represents iterations and  $i=1, \dots, N$   
 $B$  is the prearray matrix and has dimension of

$(I+M+N) \times (I+M)$  and  $P_{|0}^{1/2} = 1/\sqrt{\alpha}I, B_0 = 0_{N \times M}$   
 $e_i^{N \times 1}$  is the  $i$ -th unit vector of dimension  $N$  and  $\Theta_i$  is any unitary transformation (Jacobi rotation) that block lower triangularizes the pre-array denoted by  $M$ . After  $N$  iterations,

$$P_{|0}^{1/2} = P_{|N}^{1/2} \text{ and } Qa = B_N \quad (7)$$

Equations (6) and (7) are used in pseudo inverse computation. For the rest of the algorithm, the reader is referred to [10].

### 3. Sequence of operations in Pseudo Inverse computation

A MatLab program has been developed to model computation of pseudo inverse using the square root algorithm. The program first generates independent and identically distributed (iid) complex channel elements and assumes a particular value for the Signal to Noise Ratio (SNR). After generating channel elements, the rest of the code models pseudo inverse the way it will be implemented in hardware.

Computation of pseudo inverse is done in  $N$  iterations. Each iteration consists of three steps. The first step starts with forming/updating the prearray matrix  $B$  which for a  $M=N=4$  antenna system is a matrix of dimension  $9 \times 5$ . The weighted channel

elements in the first row described by  $H_i P_{|i-1}^{1/2}$  in which

$H_i$  is a  $1 \times 4$  vector and  $P_{|i-1}^{1/2}$  is a  $4 \times 4$  matrix is obtained using complex multiplication and addition. In the second step, each complex weighted channel element in the first row is made real by rotating its imaginary part to zero [13] using Jacobi's rotation. Jacobi's rotation is carried out using CORDICs (Coordinate Rotation Digital Computers). Jacobi's rotation is described by equations (9) and (10) [9].

$$\begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} a_r \\ a_i \end{bmatrix} = \begin{bmatrix} r \\ 0 \end{bmatrix} \quad (9)$$

$$\theta = -\tan^{-1}(a_i / a_r) \quad (10)$$

Given a complex number, the angle is first calculated using equation (10) and then the complex number is rotated using equation (9) to make its imaginary part 0. Both rotation and angle calculation are carried out using CORDICs.

The first row of the prearray matrix is given by (11)

$$[1 \quad h_{11} \quad h_{12} \quad h_{13} \quad h_{14}] \quad (11)$$

where  $h_{11}, h_{12}, h_{13}$  and  $h_{14}$  are the weighted channel elements. These are the first elements of their respective columns which consist of 9 elements per column in the case of  $M=N=4$  antenna system. The first element of each column (described in 11) inside the prearray is made leader with others as followers. The leaders as in (11) are applied to the CORDIC for angle calculation. The leaders together with their corresponding followers are applied to the CORDIC for rotation. The effect of this rotation on the leader is to make its imaginary part equal to zero while its effect on the followers is to change their phase by the phase angle of the leader. At the end of this step, all leaders become real and the followers of each leader have their phase angle changed.

The third step involves zeroing the leaders. However, before proceeding to this step, it is necessary to explain the parallelism inside Jacobi's rotation in which the process of zeroing elements can be done in parallel. Each Jacobi's rotation involves either two rows or two columns. For the prearray  $B$  two Jacobi's rotations can be done in parallel and leaders can be zeroed in 3 instead of 4 iterations as shown in Figure 2. Only leaders are shown and all symbols represent real numbers. The vertical axis represents iteration number.

1	$h_{11}$	$h_{12}$	$h_{13}$	$h_{14}$	1	$h_{11}$	$h_{12}$	$h_{13}$	$h_{14}$
1	$\alpha$	0	$h_{13}$	$h_{14}$	1	0	$\beta_{12}$	0	$\alpha_{14}$
2	$\beta$	.0	$h_{13}$	$h_{14}$	1	0	0	0	$\lambda_{14}$
3	$\chi$	0	0	$h_{14}$	$\alpha_{11}$	0	0	0	0
4	$\lambda$	0	0	0					
					(a)				(b)

Figure 2: (Sequential (a) and Parallel (b) Jacobi's rotation)

In the sequential case described in Figure 2(a),  $(1, h_{11})$  is rotated which zeros  $h_{11}$  and changes  $1$  to  $\alpha$  where  $\alpha$  is any real number. Then  $(\alpha, h_{12})$  is rotated to  $(\beta, 0)$ . After this,  $(\beta, h_{13})$  is rotated to  $(\chi, 0)$  and lastly  $(\chi, h_{14})$  is rotated to  $(\lambda, 0)$ . In the parallel case described in 2(b),  $(h_{11}, h_{12})$  and  $(h_{13}, h_{14})$  are rotated to  $(0, \beta_{12})$  and  $(0, \alpha_{14})$  simultaneously. After this, there are only three

columns left, rotation is done sequentially and  $(\beta_{12}, \alpha_{14})$  is rotated to  $(0, \lambda_{14})$ . Lastly,  $(1, \lambda_{14})$  is rotated to  $(\alpha, 0)$ . This implies that one iteration can be removed if rotation is carried out in parallel.

The leaders are then zeroed out using the parallel Jacobi rotation and this completes one iteration. After  $(N=4)$  such iterations,  $P^{1/2}$  and  $Qa$  give the pseudo inverse of the channel matrix. Initially the three pipelined CORDIC based pseudo inverse module proposed in [12] and shown in Figure 3, was implemented in MatLab. The module consists of a  $\Theta$ -CORDIC to calculate the angle as according to equation (10) and two  $\Phi$ -CORDICs to perform rotation as according to equation (9).

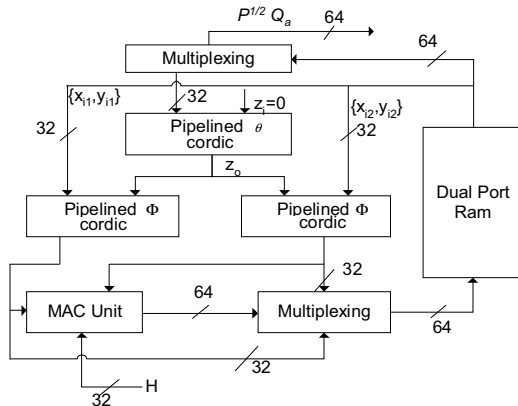


Figure 3: (Previous pseudo inverse module)[12]

In this implementation the number of iterations as well as the first step in each iteration is the same as described above. In step 2, all the four leaders are applied to the  $\Theta$ -CORDIC in pipeline. When the angle comes out of the  $\Theta$ -CORDIC, the leaders and their corresponding followers are applied to the two  $\Phi$ -CORDICs for making leaders real and changing the phase angle of the followers. In step 3 and in the process of zeroing the leaders, the  $\Theta$ -CORDIC is fed first with  $(h_{11}, h_{12})$  and then  $(h_{13}, h_{14})$ . Thus two angles need be calculated. When the angle for  $(h_{11}, h_{12})$  rotation comes out of the  $\Theta$ -CORDIC, it is applied to the two  $\Phi$ -CORDICs together with columns 2 and 3 of the prearray. The two columns are applied in parallel. After all elements of columns 2 and 3 are applied to the  $\Phi$ -CORDICs, the next angle is applied for rotating  $h_{13}$  to zero, thus 4th and 5th columns are applied in parallel for rotation.

The output is stored in the prearray. With this process, two of the four leaders are zeroed out. This corresponds to iteration 1 in Figure 2. In iteration 2, the leaders  $\beta_{12}$  and  $\alpha_{14}$  are applied for zeroing  $\beta_{12}$ . First these are applied to the  $\Theta$ -CORDIC and after calculating the angle, the angle together with columns 3 and 5 of the prearray are applied to the  $\Phi$ -CORDICs. The output from the  $\Phi$ -CORDICs is stored in the

prearray with  $\beta_{12}$  rotated to zero and  $\alpha_{14}$  modified to  $\lambda_{14}$ . In the last iteration  $(1, \lambda_{14})$  are applied to the  $\Theta$ -CORDIC, the angle is then applied to the  $\Phi$ -CORDICs for rotation and this rotates the last leader to zero. This completes one iteration. After four such iterations,  $P^{1/2}$  and  $Qa$  give the pseudo inverse of the augmented channel matrix.

## 4. Proposed Pseudo Inverse module

The proposed architecture [11] for the pseudo inverse (Figure 4) consists of two independent and generic pipelined CORDICs and a dual port ram to support the two CORDICs. Each CORDIC is developed to have 13 micro-cells. The number system used is 16Q8 with 8 bits for precision, 7 bits for dynamic range and 1 bit for sign. The process of computing the pseudo inverse is the same as described in section 3; however, the difference is that the two CORDICs are used both in vectoring as well as in rotation mode. The control unit of each micro-cell is designed to configure the cell for either vector or rotation mode. The inputs to the control unit of each micro-cell are the sign bits of x, y and z inputs as well as CORDIC mode signal. The CORDIC mode signal determines the vectoring or rotation mode for the micro-cell. This signal is propagated together with x, y and z data from the first micro-cell to the last one. Initially the two CORDICs are in the vectoring mode and calculate the angles. The angles are then applied as input to both CORDICs and both CORDICs perform rotation together in the way described in section 3. The rotated vector needs scale correction in which each coordinate of the rotated vector is multiplied with the scale correction constant which is 0.6057 [13]. A simple shift and add circuit instead of multiplier is used to perform this correction. The scale correction in 16Q8 format is 10011011 which is equal to  $2^7+2^4+2^3+2^1+2^0$  and can be implemented.

## 5. Dual Strategy in Proposed Architecture

### 5.1 Algorithmic Optimization of number of multipliers

The function of the MAC unit inside the pseudo inverse module is to perform the complex matrix multiplication given by  $H_i P_{i-1}^{1/2}$  where  $H_i$  (1x4) is the ith row of the channel matrix,  $P_{i-1}^{1/2}$  (is the 4x4) inverse triangular matrix and  $H_i P_{i-1}^{1/2}$  is 1x4 vector. The MAC unit shown in Figure 5b computes one value of  $H_i P_{i-1}^{1/2}$  in 4 clock cycles using 4 multipliers and



compared to the power consumption of the conventional MAC module.

The clock gating is just a simple ANDing of the clock signal with a control signal. Clock gating has produced negligible overhead in terms of area. By clock gating, the power consumption of Cordic-2 has been reduced from 29.409mw to 20.808mw which is a reduction of about 29%. With the two modifications, the overall power and area has been reduced by 36% and 31% respectively compared to the architecture proposed in [12].

## 7. Conclusion

The authors have presented an area and power efficient VLSI architecture for computing pseudo inverse through square root algorithm. The architecture reduces area and power consumption by reducing the redundant hardware as much as possible and achieves 31% area and 36% power efficiency. The architecture exploits parallelism inherent in Jacobi's rotation. The architecture is simulated within MMSE-VBLAST system and imposes no degradation in performance as well as latency.

## 8. References

- [1] G. J. Foschini, "Layered Space-Time architecture for wireless communication in fading environments when using multiple antennas," Bell Labs Tech. J., vol 2, Autumn 1996
- [2] G. Lawton, "Is MIMO the future of wireless communications?", computer, vol: 37, issue 7, July 2004 Pages 20-22
- [3] D. Garrett, L. Davis, St. Brink, B. Hochwald, G. Knagge, "Silicon complexity for maximum likelihood MIMO detection using spherical decoding", Solid-State Circuits, IEEE Journal, vol. 39, Issue 9, Sept. 2004, Pp(s):1544-52
- [4] M. Chandrakasam, M. Potkonjak, R.Mehra, J. Rabaey, and R. Broderon, "Optimizing power using transformations", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 14, no.1 pp. 12-31, January 1995
- [5] M.Pedram,"Power Minimisation in IC Design: Principles and Applications", ACM Transactions on Design Automation of Electronic Systems, vol. 1, no. 1, pp. 3-56, January 1996.
- [6] X. Wu, M. Pedram, L. Wang, "Multi-code state assignment for low power design", Circuits, Devices and Systems, IEE Proceedings vol. 147, Issue 5, Oct. 2000 Page(s):271 - 275
- [7] 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", Proc. ISSSE'98, Sept. 1998
- [8] A. Adjoudani, E.C. Beck, A.P. Burg, G.M. Djuknic, T.G. Gvoth, D. Haessig, S. Manji, M.A. Milbrodt, M. Rupp, D. Samardzija," Prototype experience for MIMO

- BLAST over third-generation wireless system", Selected Areas in Communications, IEEE Journal on Vol. 21, Issue 3, April 2003 Page(s):440 - 451
- [9] Gene. H. Golub, Charless F. Van Loan, "Matrix Computation"
- [10] B. Hassibi, "An efficient square-root algorithm for BLAST", Acoustics, Speech, and Signal Processing, 2000. ICASSP '00. Proceedings. 2000 IEEE International Conference on, Volume 2, 5-9 June 2000 Page(s):II737 - II740 vol.2
- [11] Zahid Khan, Tughrul Arslan, John S. Thompson, Ahmet T. Erdogan, "Area and Power efficient VLSI Architecture for Computing Pseudo Inverse of Channel Matrix in a MIMO Wireless System", accepted for publication in IEEE-VLSI conference to be held in India on Jan 3, 2006
- [12] Z.Guo, P.Nilsson, "A VLSI implementation of MIMO detection for future wireless communications", Personal, Indoor and Mobile Radio Communications, 2003. PIMRC 2003. 14th IEEE Proceedings on, vol. 3, 7-10 Sept. 2003 Pages:29-49
- [13] C.M. Rader, "VLSI systolic arrays for adaptive nulling", (radar) Signal Processing Magazine, IEEE, vol. 13, Issue: 4, July 1996, Pages:29 - 49

	Previous Pseudo Inverse Module	Area in $\mu\text{m}^2$	Proposed Pseudo Inverse Module	Area in $\mu\text{m}^2$	Modified Pseudo Inverse Module	Area in $\mu\text{m}^2$
1	$\Theta$ - CORDIC	250120	CORDIC -1	253659	CORDIC-1	253659
2	$\Phi$ - CORDIC	258920	CORDIC -2	253659	CORDIC-2	253659
3	$\Phi$ - CORDIC	258920				
4	MAC Unit	147611	MAC Unit	147611	MAC Unit	93651
5	Control Unit and duram	49210	Control Unit and duram	66015	Control Unit and duram	66015
	Total	964781	Total	720944	Total	666984
	% Saving	%		24%		31%

Table 1: (Area figures for comparison)

	Previous Pseudo Inverse Module	Power in mw	Proposed Pseudo Inverse Module	Power in mw	Modified Pseudo Inverse Module	Power in mw
1	$\Theta$ - CORDIC	31.278	CORDIC -1	33.608	CORDIC -1	33.608
2	$\Phi$ - CORDIC	33.548	CORDIC -2	29.409	CORDIC -2	20.808
3	$\Phi$ - CORDIC	33.463		-----		-----
4	MAC Unit	28.484	MAC Unit	28.449	MAC Unit	20.449
5	Control Unit and duram	12.225	Control Unit and duram	14.329	Control Unit and duram	14.329
	Total	138.998	Total	105.795	Total	89.194
	% Saving			25%		36%

Table 2: (Power figures for comparison)