Efficient Algorithmic and Architectural Optimization of QR-based Detector for V-BLAST

Fariborz Sobhanmanesh and Saeid Nooshabadi

Abstract – **The use of multiple antennas at both transmitting and receiving sides of a rich scattering communication channel improves the spectral efficiency and capacity of digital transmission systems compared with the single antenna communication systems. However algorithmic complexity in the realization of the receiver is a major problem for its implementation in hardware. This paper investigates a near optimal algorithm for V-BLAST detection in MIMO wireless communication systems based on the QR factorization technique, offering remarkable reduction in the hardware complexity. Specifically, we analyze some hardware implementation aspects of the selected algorithm through MATLAB simulations and demonstrate its robustness. This technique can be used in an efficient fixed point VLSI implementation of the algorithm. We also provide the VLSI architecture that implements the algorithm.**

Index terms – **V-BLAST, QR Factorization, CORDIC, MIMO.**

I. INTRODUCTION

In the recent years several new techniques have been developed to increase the data transmission rate in the wireless data communication. To achieve higher data rates, efficient use of the available radio spectrum is essential. The Multiple Input Multiple Output (MIMO) wireless communication system [1] increases spectral efficiency and capacity of digital communication systems with V-BLAST detection algorithm used for the decoder. This improved efficiency has been achieved by concurrently transmitting multiple data streams in the same frequency band. However, its complex receiver makes it unsuitable for low-power VLSI implementation. Several alternative algorithms and architectures for V-BLAST detection are proposed to reduce its complexity [2, 3].

For a suitable V-BLAST detection implementation, its algorithmic, arithmetic and architectural aspects require careful consideration. At the algorithmic level the numerical stability and robustness should be considered. At the arithmetic level signal quantization is important issue. At the architectural level parallelism and pipelining require attention.

In this paper we investigate the use of 1-pass QR factorization of the channel transfer matrix for VLSI hardware implementation. By suitable modification in the QR factorization technique we resolve the problem of numerical instability associated with the division based back substitution [4], while maintaining acceptable performance. We select the CORDIC method for implementing the QR factorizer in an upper triangular systolic array. Our MATLAB simulations of fixed-point implementation of algorithm point to a possible efficient VLSI hardware implementation.

This paper is organized as follows. The brief system model of the V-BLAST MIMO systems is, presented in Section II. In Section III we present the 1-pass QR factorization method and compare it with the 2-pass QR algorithm [3] using the simulation results. In Section IV we analyze some parameters of the proposed architecture based on CORDIC engine for the selected 1-pass QR factorization algorithm. We investigate the effects of these parameters on the final BER through MATLAB simulations. The Hardware architectures are presented in Section V. Section 6 concludes the paper.

II. V-BLAST SYSTEM OVERVIEW

At the transmitter side of a MIMO system a single data stream is demultiplexed into M sub streams and each sub stream is encoded independently into the symbols from the same constellation set (Ω) and then fed to its dedicated transmit antenna. At each symbol time a vector $\boldsymbol{S}=(\boldsymbol{s_1}, \boldsymbol{s_2}, \ldots, \boldsymbol{s_M})^T$, with each symbol $\boldsymbol{s_i}$ belongs to the QPSK constellation is sent to the receiver through a rich scattering quasi-static flat fading wireless channel. The received signal $\boldsymbol{r_i}$ at the i^{th} receiving antenna for that symbol time is a noisy superimposition of the M transmitted signals contaminated by AWGN, given by:

$$r_i = \sum_{j=1}^{M} h_{ij} s_j + n_i, \quad i = 1,2,\ldots,N \tag{1}$$

where h_{ij} is the channel fading between transmitter j and receiver i, which is a complex Gaussian random variable with zero mean and variance of 0.5 for the real and imaginary components, and n_i is the complex Gaussian white noise with zero mean and variance $\boldsymbol{\sigma}^2$.

Because of the quasi-static flat fading nature of the channel we can assume that the channel transfer matrix is constant over a block time of L symbol duration and changes randomly after each block time. The rich scattering condition of the channel is well satisfied in indoor environments [5] with a number of scattering sources around the transmitter or

Manuscript received May 31, 2004; revised June 28, 2005 and July 08, 2005. The paper was presented in part at the Conference on Software, Telecommunications and Computer Networks (*SoftCOM*) 2004.

F.Sobhanmanesh and S. Nooshabadi are with University of New South Wales, Sydney, Australia (e-mail: f.sobhanmanesh@student.unsw.edu.au, saeid@unsw.edu.au).

receiver. The system in Equ. (1) can be expressed in matrix form as:

$$\boldsymbol{r}_{N\times 1} = \boldsymbol{H}_{N\times M} * \boldsymbol{S}_{M\times 1} + \boldsymbol{n}_{N\times 1} \tag{2}$$

Among the MIMO algorithms, Maximum Likelihood (ML) detector algorithm is considered to the best performing and computationally the most complex one. V-BLAST OPT [6] however is generally recognized as suboptimum detector for MIMO. To detect the transmitted signals, the original V-BLAST algorithm involves 4 steps; ordering, nulling, slicing and cancellation [6].

III. QR FACTORIZATION BASED METHOD

The repeated pseudo inverse matrix computation in the ordering step of the original V-BLAST is the main computational bottleneck of the algorithm [2]. To overcome this problem other alternative methods, with acceptable performance and minimal degradation, have been proposed [2, 3]. Algorithm based on the 2-pass QR detection is claimed to be 4 times less complex than the V-BLAST OPT algorithm while achieving comparable performance [3]. In this technique the channel matrix H is first arranged in decreasing column norm order. Such arrangement of columns ensures that signals are detected in the increasing and decreasing orders of their signal to noise ratios for subsequent pass one and pass two of QR factorization algorithm, respectively. Next, the 2-pass QR algorithm employs QR factorization twice with upper and lower triangularized channel matrices. Subsequently, Symbol Interference Cancellation (SIC) detection method is used to detect the transmitted symbols in the, increasing and decreasing orders of their signal to noise ratios, by the backward and forward substitutions in upper and lower triangular channel matrices, respectively. It then averages the soft values of the detected symbols from the two passes to estimate symbols. For QAM constellation with $q > 4$, we only require to employ 1-pass QR factorization, with backward substitution to detect the transmitted symbols in the decreasing order of their signal to noise ratios, to achieve satisfactory performance [3].

In our analysis we have applied the 1-pass QR factorization detection algorithm, to a 4×4 channel matrix. The channel matrix is sorted with respect to its column norms in the increasing order. The transmitted signals come from the uncoded QPSK constellation set. Subsequently, we apply the backward substitution SIC with hard decision to upper triangular channel matrix. The hard decision technique further simplifies the hardware design. The QR factorization method involves the decomposition of $\boldsymbol{H}$ matrix into two matrices $\boldsymbol{Q}$, and $\boldsymbol{R}$. Matrix $\boldsymbol{Q}$ is a unitary matrix where:

$$\boldsymbol{Q}^{\mathrm{H}}{}_{M\times N} * \boldsymbol{Q}_{N\times M} = \boldsymbol{I}_{M\times M} \tag{3}$$

where $\boldsymbol{Q}^{\mathrm{H}}$ is transposed conjugate of $\boldsymbol{Q}$ and $\boldsymbol{I}$ is an identity matrix. Matrix $\boldsymbol{R}$ is an upper triangularized matrix. The transmitted symbol S_{M*1} matrix in the MIMO Equation of (2) can be computed by reexpressing it as:

$$\boldsymbol{S}_{M\times 1} = \boldsymbol{R}^{-1}{}_{M\times M} * \boldsymbol{Q}^{\mathrm{H}}{}_{M\times N} * \boldsymbol{r}_{N\times 1} \tag{4}$$

In the CORDIC based QR factorization techniques employed in the proposed hardware architecture, the matrix inversion and multiplication of Equation (4) are implicitly carried out by CORDIC engines through a series of micro rotations.

Our simulation results in Fig. 1 indicate that the BER performance of the 1-pass QR factorization with optimum ordering (increasing column norms) is very close to the 2-pass QR factorization in [3] with half the computational complexity.

Fig. 1, also, shows the degraded BER performance for the worst case ordering (decreasing column norms) of the 1 pass QR factorization method. We can therefore, conclude that the 1-pass QR factorization with increasing column norms order with hard decision provides a satisfactory performance with a complexity 8 times less than the V-BLAST OPT algorithm [3]. This makes the QR factorization an attractive technique for VLSI hardware implementation.

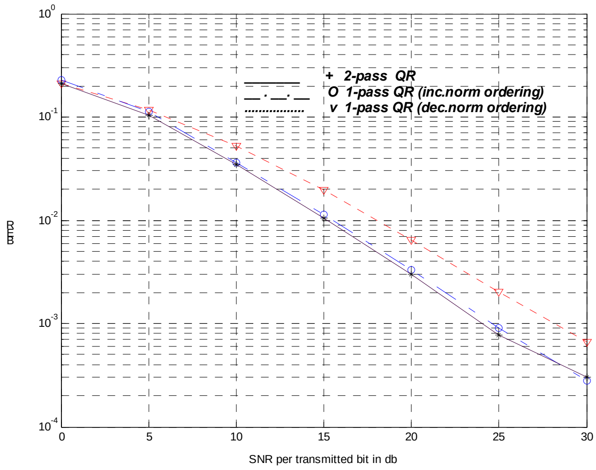

Fig. 1. Performance comparison

IV. VLSI HARDWARE IMPLEMENTATION

Towards the goal of VLSI hardware implementation of the above algorithm, we have carried out the architecture design of the 1-pass QR factorization detection technique. In addition we analyzed some of the parameters influencing its hardware implementation through a set of MATLAB simulations.

A. Architecture

Our architecture for QR factorization is based on the triangular systolic array of Fig. 2 [7]. These array processors are CORDIC-based engines. Since not all of these processors are operating simultaneously, we can increase the efficiency of hardware utilization by mapping these 14 processors to 3 processors by time multiplexing and scheduling. This is achieved through a mapping and folding procedure indicated in Fig. 2 [8].

$$\begin{pmatrix} h_{11} & h_{12} & h_{13} & h_{14} & r_1 \\ h_{21} & h_{22} & h_{23} & h_{24} & r_2 \\ h_{31} & h_{32} & h_{33} & h_{34} & r_3 \\ h_{41} & h_{42} & h_{43} & h_{44} & r_4 \end{pmatrix}$$

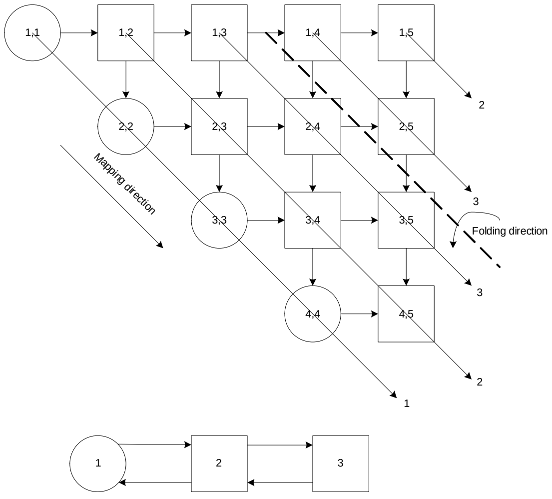

Fig. 2. Triangular systolic array

The first processor generates the Givens rotations for 4 boundary processors and the second and third processors do the Givens rotations on the channel transfer matrix and the received vector to make the upper triangular matrix. The input to this triangular systolic array is the channel matrix augmented by the received vector column as shown below.

The processors are made of 2-stage CORDIC engines [10] for annihilating the sub diagonal entries of the channel matrix. The first stage CORDIC (θ-CORDIC) in processor 1 vectorizes the channel matrix entries (e.g. h_{41} and h_{31}) by rotation in the complex plane to real numbers. It also keeps the record of the rotation angle θ for each vectorization. In doing so it only keeps a record of the signs of the micro-rotations. This removes the need for large ROM for angle storage and simplifies the hardware complexity to a shift register buffer. The second stage CORDIC (φ-CORDIC) engine in processor 1 accepts two real numbers (e.g. h_{41} and h_{31}) and annihilates one of them (h_{41} for upper triangular matrix) through vectorizing while saving the other one (h_{31}) for the next annihilation with vectorized h_{21}. The required rotation angle for annihilation is also calculated in the same manner as in the first stage CORDIC engine. This simple formatted angle information is passed horizontally to the second and the third processors to perform the same rotations on the corresponding row entries (e.g. h_{42}, h_{43}, h_{44}, r_4 for h_{41}).

We have optimized the CORDIC engines for our specific application with respect to compensation gain, number of CORDIC iterations and also size of word-length for the variables.

B. CORDIC Compensation Gain

The CORDIC engine used for the rotation of vectors has a gain of k = 1.6473 [9]. To compensate for this gain, the rotated vector coming out of each CORDIC rotator should be multiplied by the compensating scale factor of $k^{-1} = 0.6073$. The hardware multiplier required for this scaling factor is a major concern for the VLSI implementation design. We have simulated our V-BLAST architecture in MATLAB with different values of the compensating scale factor. The simulation results are shown in Fig. 3. The results show that this architecture is very robust with respect to variations in the scaling factor k^{-1}. For values of k^{-1} in the range of 0.5 to 1.0, the BER curves nearly match each other. However, for values outside this range, e.g. 0.3, 1.5 or 2.0, the performance is degraded severely. The compensating scale factor of 0.5 is selected for hardware implementation. This simplifies to 1-bit right shift through hardwiring. This choice of k^{-1} simplifies our hardware greatly.

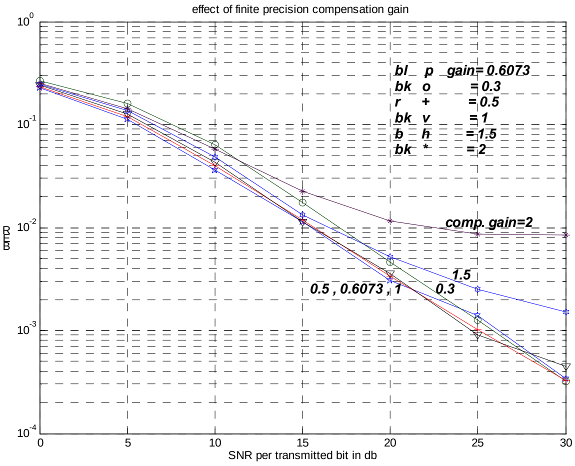

Fig. 3. Compensation gain analysis

C. Number of CORDIC Iterations

Next parameter analyzed for hardware optimization is the number of CORDIC iterations in each CORDIC engine and its influence on BER. The simulation results are present in Fig. 4, and as shown, CORDIC iterations of 4 to 6 do not offer good performance. However CORDIC iterations in excess of 7 will provide the same level of performance. To simplify the controller hardware for CORDIC rotator, we have chosen an iteration value of 8 for the CORDIC engines.

D. Word-length Analysis

Another parameter that influences the implementation cost and performance of the VLSI hardware is the number of fractional bits that is required for signal representation. We have analyzed several values for the number of fractional bits for the representation of the channel transfer matrix entries, received vector components and intermediate results of CORDIC iterations. The results for values ranging from 8 to 15 bits are shown in Fig. 5.

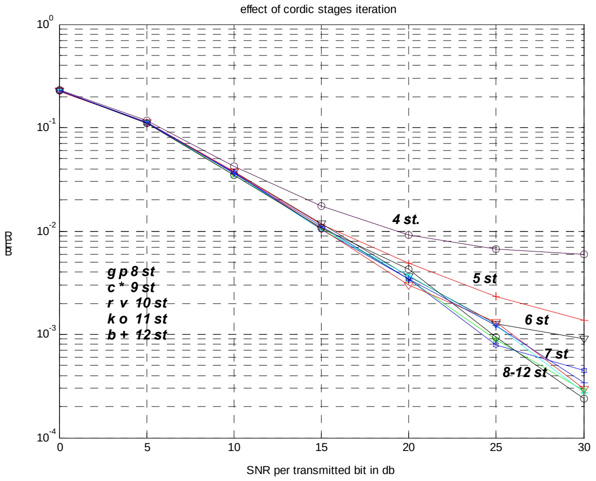

Fig. 4. CORDIC iterations analysis

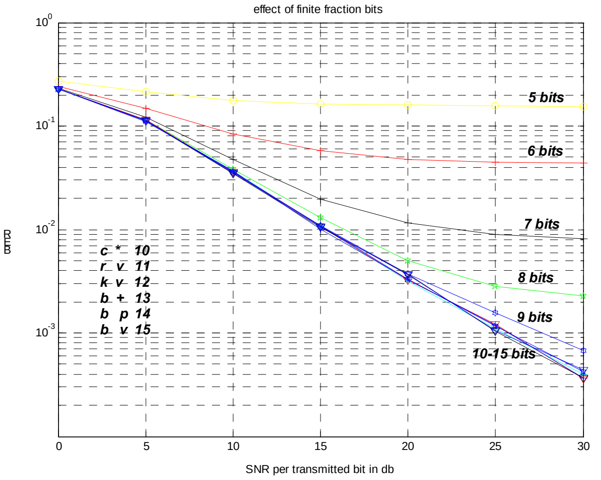

Fig. 5. Finite word-length analysis

As shown in Fig. 5, 10 to 15 fractional bits offer almost identical performances, while the performance with less than 10 fractional bits is degraded considerably. A value of 10 fractional bits can be considered to be optimum for the fixed-point representation of our system variables.

E. Back Substitution SIC

After QR factorization of the channel matrix the back substitution SIC is used to estimate the transmitted signals. Numerical instability of division based back substitution technique is a major problem associated with SIC technique [4]. To overcome the instability problem of division based back substitution, we have eliminated the division operation by performing a pre-rotation using a simple negation hardware on the incoming channel transfer matrix and using hard decision function. The pre-rotation makes all the diagonal entries of the upper triangular matrix R positive numbers, and hence, by using hard decision function we do not need any division. We can estimate the transmitted signal by simply considering the sign of the accumulated sum in the backward substitution SIC step. This method reduces the hardware complexity substantially, while maintaining an acceptable level of performance.

V. HARDWARE ARCHITECTURE

Using the optimized hardware parameters we have designed the processors internal architecture and its memory subsystem.

A. Memory Subsystem Management

The map of the memory subsystem presented in Fig. 6. As seen channel matrix data are saved in separate memories to provide the highest throughput. The memory block, containing the data for the first column of the channel matrix H, goes to vectorizing processor 1. The memory bank for data for the second column of H, and the received vectors memory bank are multiplexed and applied to the rotating processor 2. The memory banks for the data for the third and fourth columns of H are multiplexed and connected to processor 3. All memory banks are dual port RAMs with the capability of simultaneous reads by processors and updates by new channel matrix data and new received vector, respectively, by channel estimator and the receiver blocks. The ordering of data in the memory banks will simplify the memory controller unit design significantly and enable the use of a single common read counter for all the memory banks.

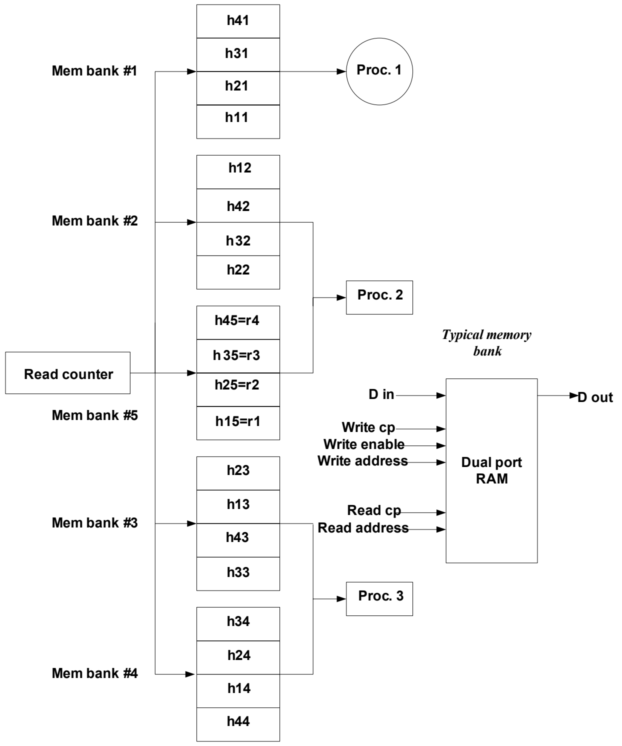

Fig. 6. Memory subsystem

B. Vectorizing Processor Architecture

The internal architecture of the first vectorizing processor along with its angle memories is shown in Fig. 7. The 2-stage CORDIC engine ($\boldsymbol{\theta}$ and $\boldsymbol{\varphi}$) along with the pre rotator block calculate the angle values in our special format (signs of microrotations) and save them in the angle memories for the later use by rotating processors 2 and 3. The storage buffers in Fig. 7 are switched, based on a time schedule in control unit, to provide mapping of processors (1,1), (2,2), (3,3) and (4,4) in the systolic array architecture of Fig. 2 to the CORDIC processor 1.

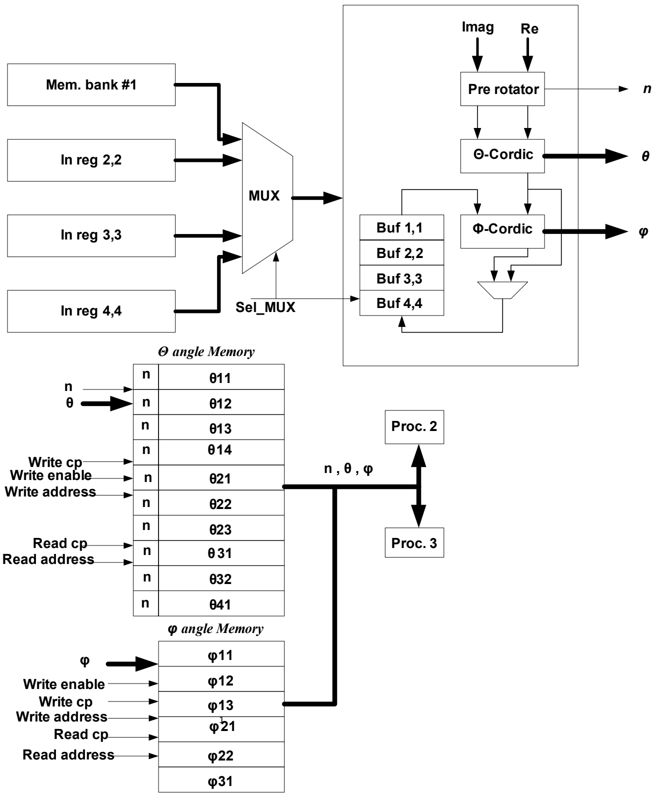

Fig. 7. Vectorizing processor architecture

C. Rotating Processor Architecture

The internal architecture of the rotating processor 2 is shown in Fig. 8. The inputs and outputs to this processor along with the CORDIC engines and their connections correspond to the folding and mapping process in the systolic array. The CORDIC engine rotates all the row entries of the channel matrix H and received vector r by the same angle sets $\boldsymbol{\theta}$ and $\boldsymbol{\varphi}$ as were computed by the vectorizing process of the first entry in each row. Since the channel matrix components are complex numbers, the components of the vectors to be rotated are complex numbers as well. Processor 2 regards these complex component vectors as two real component vectors and rotates each component separately using two $\boldsymbol{\varphi}$ CORDIC engines. The buffers are switched to provide mapping of processors (1,2), (1,5), (2,3), (3,4) and (4,5) in the systolic array of Fig. 2 to the processor 2.

The rotating processor 3 has the same internal architecture as rotating processor 2 except that the buffers correspond to processors (1,3), (1,4), (2,4), (2,5), and (3,5) of the systolic array of Fig. 2. The input multiplexer is fed by the third and fourth columns data of the channel matrix H, and input registers of processors (2,4), (2,5) and (3,5). The output registers correspond to processors (1,3), (1.4), (2,4) and (2.5).

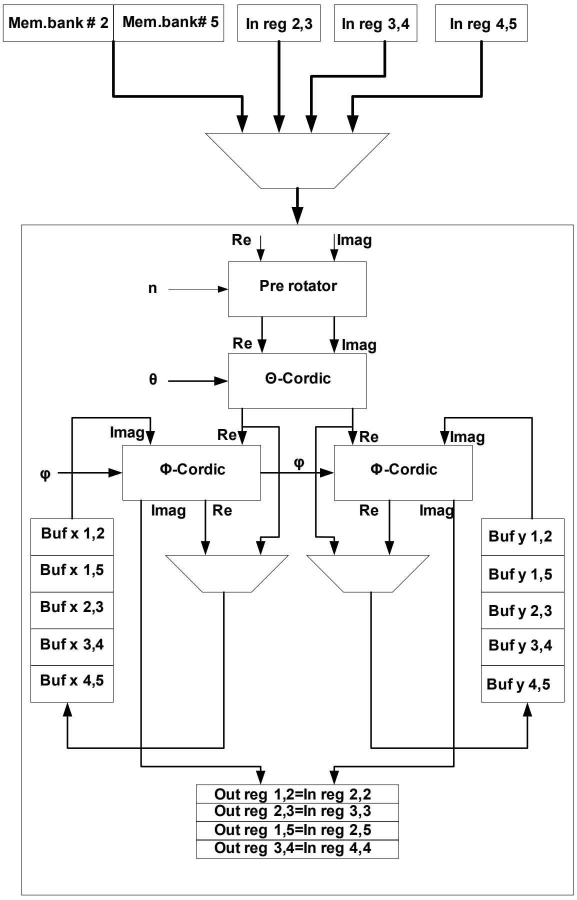

Fig. 8. Rotating processor 2

VI. CONCLUSION

In this paper we have studied the QR factorization method for V-BLAST detector from the hardware implementation point of view for a 4 transmitting, 4 receiving antennas MIMO wireless system. We have investigated and optimized some important parameters that influence the systolic array implementation of this system. They include optimization of the compensation gain factor, number of CORDIC iterations, and word-length. These optimizations provide robustness and acceptable BER, while offering simple VLSI hardware implementation. We presented the management scheme for the memory subsystem. We also provided the internal architecture of the processors in the discrete mapped systolic array.

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] B. Hassibi: “An efficient square–root algorithm for blast”, *Conference Record of the Thirty-Fourth Asilomar Conference on Signals, Systems and Computers*, Pacific grove, CA, U.S.A., pp. 1255-1259, 2000.

[3] M. O. Damen, K. Abed-Meraim, and S. Burykh: “Iterative QR detection for blast”, *Journal of Wireless Personal Communications*, vol. 19, issue 3 pp. 179 - 191, 2001.

[4] Z. Guo and P. Nilsson: “A Low-Complexity VLSI Architecture For Square Root MIMO Detection”, *In proceedings of IASTED International Conference on Circuits, Signals, and Systems (CSS'03)*, Cancun, Mexico, May, 2003

[5] G. J. Foschini and R. A. Valenzuela: “Initial estimation of communications efficiency of indoor wireless channels,” *Wireless Networks*, vol. 3, no. 2, pp. 141–154, 1997.

[6] G. D. Golden, G. J. Foschini, and R. A.Valenzuela, and P. W. Wolniansky: “Detection algorithm and initial laboratory results using V-BLAST space time communication architecture”, *IEE Elect. Letters*, vol 35, no. 1, Jan. 1999.

[7] Y. Kung: “*VLSI Array Processors*”, Englewood Cliffs, New Jersey: Prentice Hall, 1988.

[8] N. Zhang, B. Haller, and Bob Brodersen: “Systematic architecture exploration for implementing interference suppression techniques in wireless receivers”, *IEEE Workshop on Signal Processing Systems*, SiPS , pp. 218 – 227, Oct. 2000

[9] R. Andraka: “A survey of CORDIC algorithms for FPGA based computers”, *In proceedings of the 1998 ACM/SIGDA sixth international symposium on Field Programmable Gate Arrays, FPGA’98*, , Monterey, CA, pp. 191-200, Feb. 1998

[10] C. M. Rader: “VLSI systolic arrays for adaptive nulling”, *IEEE Signal Processing Magazine*, vol 13, no 4, 1996, pp. 29-49

Fariborz Sobhanmanesh received his BSc and MSc degrees in Telecommunication engineering from Shiraz University and Isfahan University of Technology, Iran, in 1989 and 1992 respectively. His Msc research was on the neural networks and their applications for Persian handwritten numbers recognition.

Since 1992 till 2002 he was a lecturer in the Department of Computer Science and Engineering, Shiraz University, Iran, where he was engaged in teaching and research in digital circuits, computer architectures and microprocessors. Since, 2003, he has been a PhD student in the School of Electrical Engineering and Telecommunications, University of New South Wales, Sydney, Australia. His research topic is on the design of a single chip V-BLAST detector.

Saeid Nooshabadi received the BSc. and MSc. degrees in physics and nuclear physics from Andhra University, India, in 1982 and 1984, respectively, and the MTech and PhD degrees in electrical engineering from the India Institute of Technology, Delhi, India, in 1986 and 1992, respectively.

Currently, he is a Senior Lecturer in microelectronics and digital system design in the School of Electrical Engineering and Telecommunications, University of New South Wales, Sydney, Australia. Prior to his current appointment, he held academic positions at the Northern Territory University and the University of Tasmania between 1993 to 2000. In 1992, he was a Research Scientist at the CAD Laboratory, Indian Institute of Science, Bangalore, India, working on the design of VLSI chips for TV ghost cancellation in digital TV. In 1996 and 1997, he was a Visiting Faculty and Researcher, at the Centre for Very High Speed Microelectronic Systems, Edith Cowan University, Western Australia, working on high performance GaAs circuits; and Curtin University of Technology, Western Australia, working on the design of high speed-high frequency modems. His research interests include very high-speed integrated circuit (VHSIC) and application-specified integrated circuit design for high-speed telecommunication and image processing systems, low-power design of circuits and systems, and low-power embedded systems.