Efficient Implementation of QR Decomposition for Matrix Inversion.

Vijaykumar S. Shirwal[1], Dr. Mahesh S. Chavan[2]
[1]*Department of Technology, Shivaji University, Kolhapur. Maharashtra, India.*
[2]*Department of Electronics Engineering, KIT's College of Engineering, Kolhapur. Maharashtra, India.*
(E-mail: vshirwal76@gmail.com, maheshpiyu@rediffmail.com)

Abstract**—**Matrix inversion is the key aspect in multiple-input multiple-output (MIMO) communication systems. Multiple antennas at the transmitter and multiple antennas at the receiver leads to spatial diversity which helps to increase the data rate and diversity gain. The work presented shows implementation of the matrix inversion operation on FPGA. Design need low computational complexity matrix inversion suitable for channel estimation required in linear detector such as minimum mean squared error (MMSE) detector. Here, matrix inversion is presented by using QR decomposition with equation solving of Gram-Schmidt orthogonalization for fixed point. The result shows that the proposed design requires less number of hardware resources.

Keywords**—***MIMO, MMSE, QRD, 3GPP-LTE*

I. INTRODUCTION

Multiple-input and Multiple-output is an upcoming technology to decide the wireless standard like 3GPP-LTE and WiMaX[1-2]. In the MIMO receiver, detection of MIMO symbol is a very crucial task due to channel interference. The data detection in the MIMO receiver either by using maximum likelihood (ML) or minimum mean squared error (MMSE) involves high dimensional matrix inversion which increases system complexity to great extent. Due to increase in the MIMO system complexity, its great challenge to VLSI implementation. Therefore, efficient implementation of matrix inversion has to be considered while designing MIMO systems. In this paper, the efficient hardware architecture for the matrix inversion is proposed. Here, the 4x4 matrix inversion is implemented with fixed-point data representation.

The rest of the paper is organized as a MIMO system model presented in section II. Section III describes the various ways of matrix inversion. The proposed work of implementation of inversion of 4x4 matrices by using QR decomposition with equation solving of Gram Schmidth orthogonalization is described in section IV along with the comparison of the proposed work with other related work. Finally, section V concludes the paper.

II. MIMO SYSTEM MODEL

MIMO system has multiple antennas at the transmitter and receiver. MIMO can increase the data rate by transmitting several information streams in parallel with same transmit power within the same frequency band [3]. The MIMO system mathematically modeled as

$$y = Hx + n \tag{1}$$

where y is received symbol, x is transmitted symbol, n is a additive white Gaussian noise added in the channel path and H is a channel matrix of the size r x t. Channel matrix is a square matrix when r = t and if r ≠ t it is non-square matrix. The r value is always kept greater than or equal to t. The channel matrix elements are represented as h_{ij} as a flat fading channel coefficient. This channel coefficient between i^{th} receive antenna and j^{th} transmit antenna. To detect symbol at receiver low complex either Zero Forcing (ZF) or MMSE linear detection schemes is used. The ZF and MMSE detection schemes are defined by the following equation

$$ZF = \left(H^H H\right)^{-1} \; H^H r \tag{2}$$

and for MMSE

$$MMSE = \left(H^H H + \sigma^2 I\right)^{-1} \; H^H r \tag{3}$$

where σ^2 is the noise variance.

In equation (1) and (2) only difference is of the noise power term σ^2 which is added in MMSE. Both ZF and MMSE decoding involves the calculations of an equalization matrix 'W', for ZF

$$W = \left(H^H H\right)^{-1} \; H^H \tag{4}$$

and for MMSE

$$W = \left(H^H H + \sigma^2 I\right)^{-1} \; H^H \tag{5}$$

The calculation of W needs to be done as soon as possible, otherwise the symbol detection will not start until W is ready. Channel preprocessing and symbol demapping are the process involved in the linear detection. The channel preprocessing unit mainly consists of matrix multiplication and matrix inversion. So, the work is focused on effective and efficient implementation of matrix inversion on hardware.

III. METHODS OF MATRIX INVERSION

The various approaches for the matrix inversion are:

A. *Analytical approach for Matrix Inversion*

The matrix inversion for a 2 x 2 matrix is computed with the analytical approach is as follows:

$$A^{-1} = \frac{adj\ (A)}{|A|} \tag{6}$$

Suppose $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$

Then

$$A^{-1} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1}$$

$$= \frac{1}{\text{ad-bc}} \begin{bmatrix} \text{d} & \text{-b} \\ \text{-c} & \text{a} \end{bmatrix} \tag{7}$$

In this the computational complexity increases as the size of the matrix increases and the numerical stability does not exist no more for the inversion of large matrices.

B. *Blockwise Analytic Matrix Inversion (BAMI)*

The inverse of 4 x 4 matrix by using BAMI can be computed by partitioning the matrices into four smaller matrices as follows:

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} A^{-1} + A^{-1}B(D - CA^{-1}B)^{-1}CA^{-1} & -A^{-1}B(D - CA^{-1}B)^{-1} \\ -(D - CA^{-1}B)^{-1}CA^{-1} & (D - CA^{-1}B)^{-1} \end{bmatrix} \tag{8}$$

C. *QR decomposition.*

Decomposition methods are generally the preferred methods for matrix inversion because they suited well for the large matrix size while the complexity of the analytic method increases as the size of the matrix increases. QR, LU and Cholesky are the three known decomposition methods used for matrix inversion. Cholesky and LU decomposition used with non-singular diagonally dominant and positive definite square matrix. On the other hand, QR decomposition is used for any matrix.

The three different decomposition methods are Gram Schmidt orthogonormalization (classical or modified), Givens Rotations (GR), and Householder reflection. Modified Gram-Schmidt (MGS) is a slightly modification of classical Gram-Schmidt algorithm (CGS) in which column wise operations are done. The QRD Modified Gram-Schmidt method is proposed for the matrix inversion. Once the given matrix A is decomposed as *A= Q x R* where matrix Q is an orthonormal with $QQ^T = Q^TQ = I$ and R is an upper triangular matrix. Inverse of matrix A can be computed as

$$A^{-1} = (Q.R)^{-1} = Q^{-1} R^{-1} \tag{9}$$

But $Q^{-1} = Q^T$

Equation (9) becomes

$$A^{-1} = Q^T R^{-1} \tag{10}$$

The MGS algorithm defined in Algorithm 1 is applied to find the matrix inversion of A matrix which of the size *m* x *n* with *m* > *n* and rank(A) = *n*. The resulting Q is the size of *m* x *n* and R is the size of *n* x *n.*

Algorithm 1 Modified Gram Schmidt Method

```
A = [a_1, a_2, ……. ,a_n], m x 1 column vectors
  for i = 1: n
      X_i = a_i
  end
  for i = 1: n
      R_ii = || X_i || , being || || L_2 norm
  end
      q_i = X_i / R_ii
  for j = i+1: n
      R_ij = < q_i, X_j >, when i ≠ j
      X_j = X_j - R_ij q_i
  end
```

In the above algorithm 1, X is the intermediate matrix and the X_j are the columns which is updating throughout the solution steps

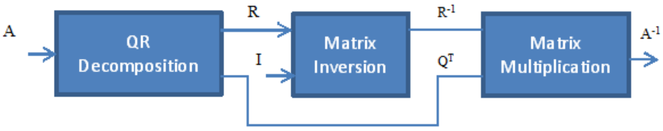

Fig.1. Steps for the inversion of the matrix A

The inversion of matrix A involves the following steps

- Get the decomposed value of Q and R by applying the algorithm1 to matrix A
- Invert the upper triangular matrix R using a triangular matrix inversion method
- Matrix multiplication of Q^T and R^{-1} to obtain the A^{-1}

The computation modules multiplier, adder, divider, square root and subtraction generated are the shared resources in the implementation of QRD-MGS algorithm.

IV. IMPLEMENTATION OF QR DECOMPOSITION

The general modules used for implementing QRD are memory module, vector multiplication, square root and divider.

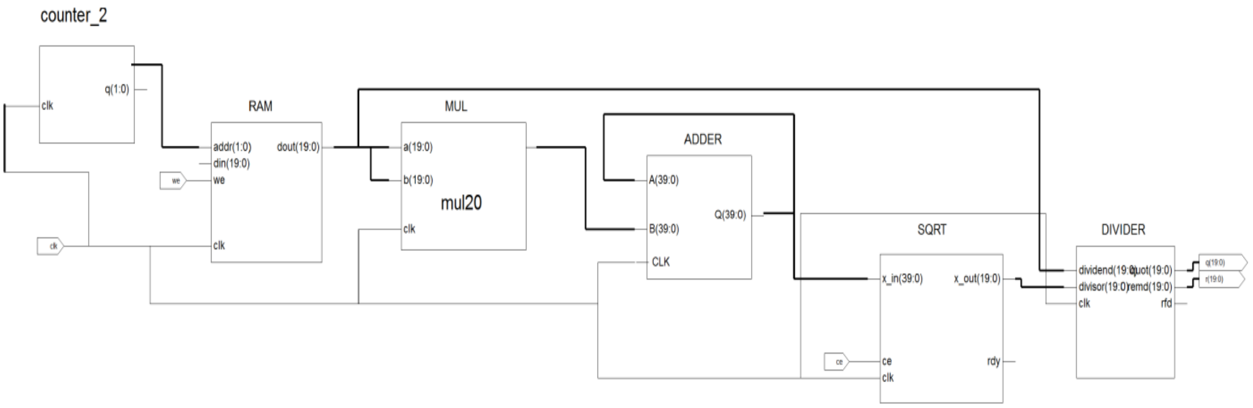

Fig.2. Schematic for norm function

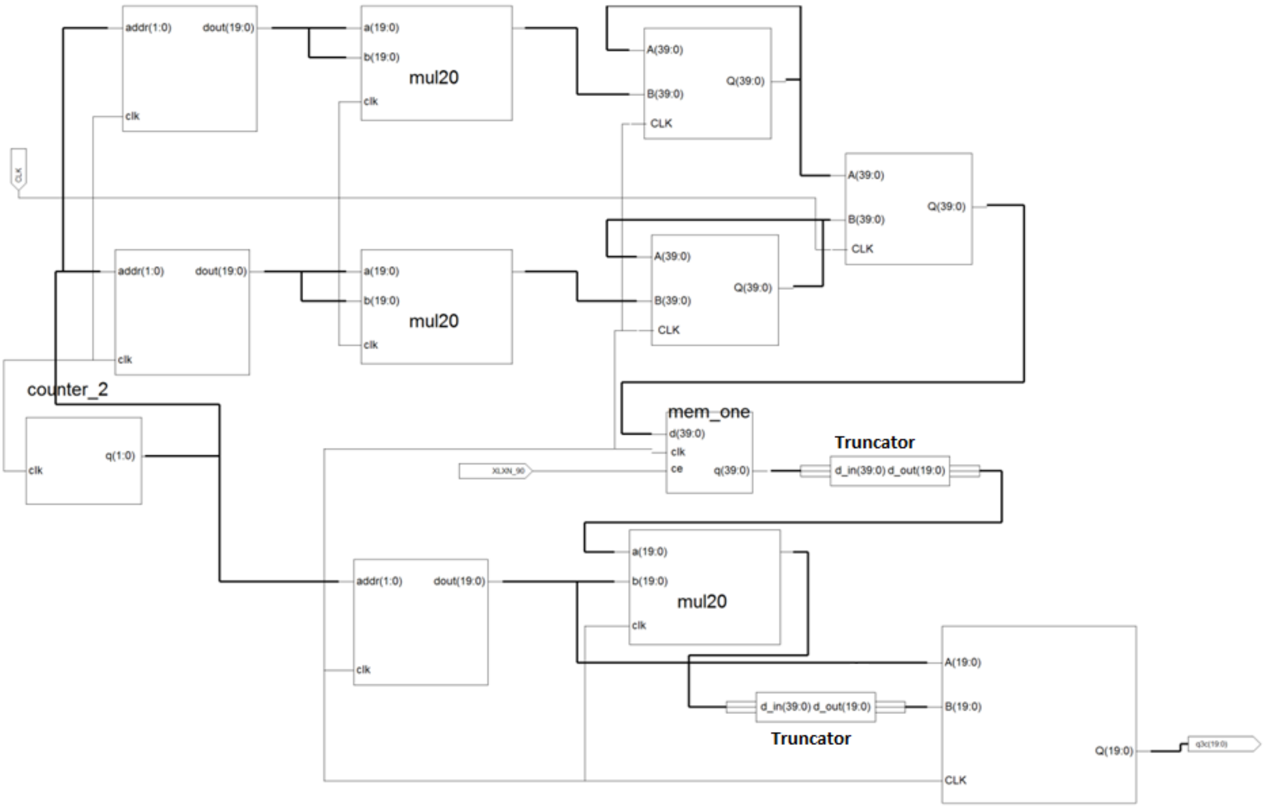

Fig.3. Schematic for QR decomposition

While designing these modules basic modules are used from Xilinx Core. The data from memory module is accessed using proper address selection logic. The input data width is considered as 20 bit fixed point format. The output data width of multiplier is 40 bit which is adjusted to 20 bit by truncating lower bits. The implementation of QRD requires computation of element of Q and R matrix. The computation unit for finding these elements requires following modules.

A. *Square root:*

This is used only for finding the norm. This unit finds the square root of 40 bit unsigned number which is directly used from the Xilinx core. This unit uses the input data range less than +2 and support for data type integer as well as fractional.

B. Divider:

This unit is used for finding the norm. It also has feature of implementing reciprocal function (1/x). It uses data width of 20 bit and provides quotient and reminder with integer or fractional value. It performs signed division in one clock cycle.

C. Vector Multiplication:

The upper non-diagonal elements of R matrix are computed using vector multiplier. The input data in vector form is taken from RAM blocks and given to multiplier unit. The signed multiplier and signed adder are used for vector multiplication. For n size vector, number of clock cycles required to obtain vector multiplication are also n. Vector multiplier is also used to obtain the elements of intermediate matrix X.

D. Norm:

The diagonal element R_{ii} of upper triangular matrix are calculated using norm function which takes input as corresponding column vector of given matrix A. The schematic of the norm function is shown in Fig.2. To implement this norm function, multipliers are used as a square function and adder is used for cumulative addition. For this unit column vectors of given matrix are stored in RAM blocks so the data is fetched from RAM block and given to multiplier unit. Column vectors of orthogonal matrix Q are obtained using divider unit along with norm function.

E. QRD Implementation:

Once all the arithmetic operations needed in this decomposition are implemented, all the blocks are synchronized to get proper data flow from all the units. The given matrix A has different column vectors. The data according to column vector is stored in RAMx. To find all the diagonal elements of matrix R, norm is used which consist of multiplier, adder and square root modules (Fig.2). The result obtained from norm is given to the divider which has another input as column vector of matrix A. After calculating all the diagonal elements, non-diagonal elements of matrix R are calculated using vector multiplication.

The elements of intermediate matrix are calculated using vector multipliers and adder. The schematic for QR decomposition is shown in Fig.3. Some of the results from the vector multiplication are used again in finding these elements. The intermediate matrix element along with divider and norm function are used to find the column vectors of Q matrix. The given matrix A is decomposed in to Q and R, it’s very easy to compute the inverse of the upper triangular matrix than the normal matrix. To obtain the R^{-1}, the analytical method is used which is implemented in VHDL code.

F. Implementation Result

The result is analyzed for the QRD-MGS based matrix inversion architecture for the 4x4 matrix in terms of resource utilized. All the modules are designed by using a VHDL code and synthesized using Xilinx ISE9.2. The designs are implemented on Virtex-4. For the comparison purpose with the previous work reported, the data bit width is kept as 20 bit. As the number of data bit length increases the accuracy increases at the cost of higher resource utilization. The Table-I shows the comparison of QRD for 4x4 matrix.

TABLE I. COMPARISONS BETWEEN PREVIOUSLY REPORTED WORK AND OUR RESULTS FOR QR DECOMPOSITION.

Method	**QR Decomposition**			
	Ref.[4]	***Ref.[5]***	***Ref.[6]***	***This Work***
Bit Width	12	20	20	20
Data Type	Fixed	Floating	Fixed	Fixed
Device Type	Virtex-2	Virtex -4	Virtex-4	Virtex-4
Slices	4400	9117	3584	3333
DSP48s	NR	22	12	NR
BRAMs	NR	NR	1	3

In proposed design, LUTs are used instead of DSP48s. The numbers of slices are less as compared to other work reported.

V. CONCLUSION

Through this work, we gained an important insight that QRD-MGS is widely preferred method for matrix inversion due to its effective and efficient VLSI implementation. The implemented design of 4x4 matrix inversion runs with clock rate of 100MHz. From Table-I comparison with other related work shows this design consumes less hardware resources based on the number of slices. This design is more useful in the MIMO system application for the estimation of channel matrix H.

ACKNOWLEDGMENT

The authors wish to acknowledge the N. K. Orchid College of Engineering & Technolgy, Solapur for providing the VLSI design lab to do the proposed work.

REFERENCES

[1] J. Lee, J. –K. Han, and J. Zhang, “MIMO technologies in 3GPP-LTE and LTE-advanced,” EURASIP J. Wireless Communication Network vol.2009, pp. 3:1-3:10, 2009.

[2] I. Pefkianakis, S.-B. Lee, and S. Lu, “MIMO-aware 802.11n rate adaption,” IEEE/ACM Trans. Netw. Vol. 21, no. 3, pp. 692-705, June 2013.

[3] H. B¨olcskei, D. Gesbert, C. Papadias, and A. J. van der Veen, Eds., Space-Time Wireless Systems: From Array Processing to MIMO Communications. Cambridge Univ. Press, 2006.

[4] EDMAN, F. AND ¨OWALL, V. 2005. A scalable pipelined complex valued matrix inversion architecture.In Proceedings of the IEEE International Symposium on Circuits and Systems. 4489–4492.

[5] KARKOOTI, M., CAVALLARO, J. R., AND DICK, C. 2005. FPGA implementation of matrix inversion using QRD-RLS algorithm. In Proceedings of the Conference Record of the 39th Asilomar Conference on Signals, Systems and Computers. 1625–1629.

[6] ALI IRTURK and BRIDGET BENSON, SHAHNAM MIRZAEI and RYAN KASTNER, "GUSTO: An Automatic Generation and Optimization Tool for Matrix Inversion Architectures," ACM Transactions on Embedded Computing Systems, Vol.9, No. 4, Article 32, March 2010.

Vijaykumar S. Shirwal received the B.E. degree in Electronics Engineering From Shivaji University, Kolhapur in 1997, and the M. Tech degree in Electronics and Telecommunication Engineering from Dr. Babasaheb Ambedkar Technological University, Lonere in 2005. Currently, he is pursuing the Ph.D degree in Electronics Engineering from Shivaji University, Kolhapur.

His research interests include Signal processing, design of VLSI architectures for communication system and embedded system. Currently, he is working on MIMO decoder design and its VLSI implementation.

Dr. Mahesh S. Chavan received the B.E. degree in Electronics Engineering from Shivaji University, Kolhapur in 1991,M.E. degree in Electrical Control Systems from Shivaji University, Kolhapur in 1998, and Ph.D degree in Electronics and Communication Engineering from Kurushetra University, Kurushetra.

He is currently with the Department of Electronics Engineering, KIT's College of Engineering, Kolhapur. His research interests include Signal processing, and Control System. He is the recipient of the best IETE journal paper award 2000.