Implementation Of A Recursive Data Of Adaptive Qrd-Rls Algorithm Using Hdl Coder

Asst. Lecturer.
Ali Khalid Jassim
Al-Mustansiriya University
College Engineering
alijassim79@yahoo.com

Asst. Lecturer.
Mohammad Hussein Ali
Al-Turath University College
College Engineering

Lecturer.
Musa Hadi Wali
Al-Mustansiriya University
College Engineering

Abstract :

Matrix inversion is a common function found in many algorithms used in wireless communication systems. As Field Programmable Gate Array (FPGA) become an increasingly attractive platform for wireless communication, it is important to understand the tradeoffs in designing a matrix inversion core on an FPGA. In this paper, a configurable Field Programmable Gate Array (FPGA)-based hardware architecture for matrix inversion is presented (download without data input). The proposed architecture of this algorithm has been design using Matlab-Simulink 7.8(R2009a) to deal with parallel structure. The design has been converted to behavioral VHDL coding style, as will as a VHDL test bench using Simulink HDL Coder tool to realize hardware directly from Simulink design. The use of Squared Givens rotations and a folded systolic array makes this architecture very suitable for FPGA implementation. Input is a8×8*matrix of complex, floating point values. The matrix inversion design can achieve throughput of 0.14Mupdates per second on a state of the art Altera Cyclone III (EP3C12F780C7) FPGA running at 125 MHz and studies a class of Q(N) approximate QR-based least squares (A-QR-LS) algorithm recently. It is shown that the A-QR-LS algorithm is equivalent to a normalized LMS algorithm with time-varying step sizes and element-wise normalization of the input signal vector.*

***Keyword:** Adaptive filtering, approximate QR-LS algorithm, performance analysis, QR-LMS algorithm, square root free givens based algorithms, transformed domain LMS algorithm.*

بناء صيغة معدلة لخوارزمية (تفكيك الرد السريع- بأقل تكرارية التربيعِ) باستخدام شفرة (HDL)

م.م.علي خالد جاسم
الجامعة ألمستنصريه
كلية الهندسة

م.م. محمد حسين علي
كلية التراث الجامعة
كلية الهندسة

م. موسى هادي والي
الجامعة ألمستنصريه
كلية الهندسة

الخلاصة:

في كثير من الخوارزميات التي تستخدم في نظم الاتصالات اللاسلكية تحتاج إلى استخدام عكس مصفوفة ذات إبعاد كبيرة وخاصة عند استخدام تعامد التردد المتعدد في التقسيم العمودي (OFDM) التي تستخدم الموجات

الكهرومغناطيسية ذات النطاق العريض، وكذلك في استخدام متعدد الإدخال ومتعدد الإخراج(MIMO).لكي يتم تنفيذ هذه الاستخدامات في مصفوفة البوابات المبرمجة موقعيا (FPGA) التي تحتاج إلى عكس مصفوفة ذات إبعاد كبيرة، فقد تم تصميم مصفوفة ذات إبعاد (8×8) وعكسها باستخدام (Squared Given rotations) والتي تكون مناسبة في التنفيذ لمصفوفة البوابات المبرمجة موقعيا، وقد تم تنفيذ هذه المصفوفة باستخدام (Mat lab - Simulink) ومن ثم تحويلها إلى (VHDL Coding) وتنفيذها باستخدام ((Alter a Cyclone III (EP3C12F780C7)) وبمعدل التغير 0.14 للثانية الواحدة والتردد125 MHz.

وكذلك تم تحليل ودراسة وتطوير خوارزمية (Approximate QR-based Least Squares A-QR-LS) من خلال استخدام (LMS)التي تقلل من الوقت وكذلك سرعة معالجة أشارات الإدخال.

1. Introduction

Matrix algorithms are commonly used in the areas of graph theory, numerical algorithms, digital control and signal processing. These areas require enormous computing power. A close examination of the algorithms used in these, and related, applications reveal that many of the fundamental actions involve matrix operations such as matrix inversion. Orthogonal Frequency Division Multiplexing (OFDM) is a popular method for high-rate data transmission in wireless environments. In OFDM, the channel bandwidth is divided into several narrow sub bands. The frequency response over each of these sub bands is flat. Hence, a frequency-selective channel is transformed into several flat-fading sub channels. The time domain waveforms of the subcarriers are orthogonal, yet the signal spectra corresponding to different subcarriers overlap in frequency [1].

Therefore, the available bandwidth is used very efficiently. The data rate of the system is aggregate of the data rate per sub channel. These features make OFDM suitable for high data rate applications. Another advantage of OFDM systems is that they are less susceptible to various kinds of impulse noise. These characteristics result in reduced receiver complexity [1].

MIMO (Multiple Input Multiple Output) systems use multiple antennas at both the transmitter and the receiver. Each antenna simultaneously transmits a small piece of data using the same frequency band to the receiver. By taking advantage of the spatial diversity resulting from spatially separated antennas, the receiver can process the data flows and put them back together [2].**Figure (1)** shows a typical M x M Multi-Input Multi-Output frequency.

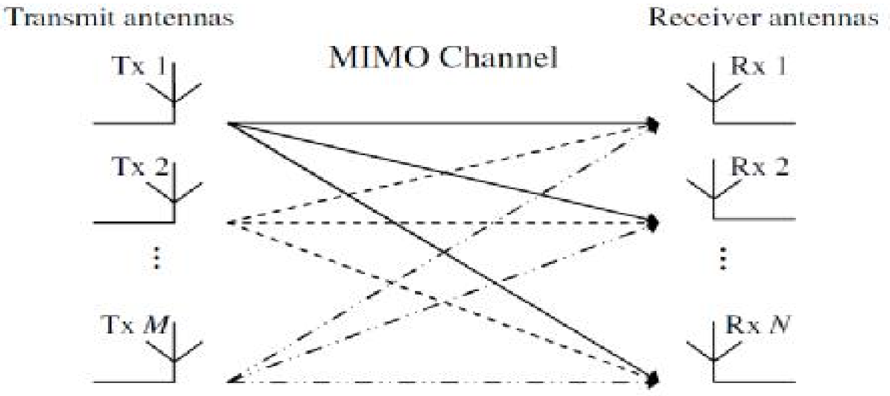


Fig.(1) :An M x M MIMO system [3]

This technique utilizes the bandwidth very efficiently. MIMO channels become frequency-selective during high data-rate transmission due to the multipath characteristics of the environment. By combining OFDM and MIMO, these frequency selective channels can be transformed to a set of frequency flat MIMO channels. Hence decreasing the receiver complexity. Therefore, MIMO-OFDM systems are very promising in broadband wireless systems [2, 4].Each receiver in MIMO-OFDM systems should equalize the received signal to remove the effect of channel on the signal. Most of equalization detection algorithms need to invert a matrix which is either the channel state information (*H*) or a nonlinear function of it *f*(*H*). Increasing the number of transmitter and receiver antennas in the system, results in a higher data rate. At the same time, dimensions of matrix *f*(*H*)increase, requiring more computations to invert the matrix in less time. This makes the matrix inversion block a bottleneck in these systems. In this work, can be developed architecture for matrixinversion by generalizing the QR decomposition-based Recursive Least Square algorithm (QRD-RLS) [5].

This algorithm has wide applications in wireless communications and Signal processing such as beam forming, channel equalization and HDTV QRD-RLS is numerically stable and has rapid convergence. It also involves local communication between nodes which is suitable for hardware implementation. Inverting a matrix using QR decomposition requires a number of rotations to nullify the unwanted values. The standard rotation algorithm called "Givens Rotation"(GR) requires square root operations and divisions that are expensive for hardware. a square-root-free version of the givens rotations or ”Squared Givens Rotation (SGR)”proposed in [6]. This algorithm eliminates the need for square-root operations and also spares half of the multiplications.

On the other hand, reconfigurable system-on-chip architecture is a promising alternative to both Application-Specific Integrated Circuits(ASIC) and general purpose processors [7]. This is due to the distinct features of FPGA chips. As a reconfigurable hardware, FPGA is powerful in its computation ability. In the last decade ,FPGA-based systems have achieved significant speedups for a range of applications including signal processing, image processing, network processors and robotics, to name a few but far from complete [8,9]. FPGA has better flexibility and shorter design cycle than ASIC.

Compared with DSP, FPGA is more convenient for system upgrading because it is reconfigurable. Hence, FPGA is an ideal choice for those embedded applications requiring powerful computational ability and reconfigurable. Thus make our new matrix inversion algorithm focus on FPGA implementation [10].

2. Algorithm Description

QR decomposition is an elementary operation, whichdecomposes a matrix into an orthogonal and a triangular matrix. Quick Response (QR) decomposition of a real square matrix A is a decomposition of A as A = Q×R, where Q is an orthogonal matrix (QT × Q = I) and R is an upper triangular matrix. Factor(m × n) matrices (with m ≥n) of full rank as the product of an (m × n) orthogonal matrix where QT × Q = I and (n× n) upper triangular matrix. There are different methods which can be used to compute QR decomposition. The techniques for QR decomposition are householder transformation [11] and the givens rotations [6]. Above all method, the Coordinate Rotation Digital Computer "CORDIC" algorithm is effective technique and using for QR decomposition.

Considering the estimation of the N-dimensional parameter vector $\widehat{\Theta}$ for the following linear model:[12]

$$d(n) = x_N^T(n)\,\widehat{\Theta} + v(n) \qquad (1)$$

where $d(n)$and x_N^T(n) are the desired (observed) signal and input vectors, respectively, and

$$e(j) = d(j) - x_N^T(j)\Theta(n) \qquad (2)$$

In least squares parameter estimation, the following time-averaged squared magnitude error is:

$$\xi_N(n) = \sum_{j=0}^{n} \lambda^{n-j} |e(j)|^2 \qquad (3)$$

where the constant λ is the forgetting factor with a value between 0 and 1. Equation (1) can be written more compactly in matrix form as:

$$e(n) = d(n) - x_N(n)\,\Theta(n) (4)$$

Where

$$d(n) = [d(0), \quad d(1) \ldots\ldots, d(n)]$$

$$x_N(n) = [x_1(n),\ x_2(n) \ldots\ldots, x_N(n)]^T \qquad (5)$$

$$X_N(n) = [x_N(0),\ x_N(1) \ldots\ldots, x_N(n)]^T \qquad (6)$$

$x_N(n)$and $X_N(n)$ are the received signal vector and the data matrix, respectively. Then, the least squares objective function $\xi_N(n)$in (3) becomes

$$\xi_N(n) = e^H w^2(n) e(n) = |w(n)e(n)|^2 \qquad (7)$$

Where *w(n)* is a diagonal weighting matrix given by

$$W(n) = diag.(\sqrt{\lambda^n}, \sqrt{\lambda^{n-1}}, \ldots\ldots, \sqrt{\lambda},\ 1) \qquad (8)$$

The optimum value of $\Theta(n)$can be obtained by solving the normal equation:

$$R_N(n)\Theta(n) = P_N(n) \qquad (9)$$

where

$$R_N(n) = \sum_{i=0}^{n} \lambda^{n-i} x_N(i)x_N^H(i) \qquad (10)$$

and

$$P_N(n) = \sum_{i=0}^{n} \lambda^{n-i} d(i)x_N^H(i) \qquad (11)$$

are, respectively, the weighted autocorrelation matrix of$x_N(n)$, and the weighted cross-correlation vector of$x_N(n)$ and $d(n)$. Due to the lower numerical accuracy in solving the normal equation, a better method, thus is called the QR-LS method, is employed. The following QRD of $w(n)X_M(n)$is performed [13].

$$Q(n)w(n)X_N(n) = \begin{bmatrix} R_N(n) \\ 0 \end{bmatrix} \qquad (12)$$

where $Q(n)$is some$(n+1)(n+1)$ unitary matrix and$R_N(n)$is (N x N) an upper triangular matrix. Using equations (12), (4) can be rewritten as

$$Q(n)W(n)e(n) = \begin{bmatrix} d_N(n) \\ C_{n+1-N} \end{bmatrix} - \begin{bmatrix} R_N(n) \\ 0 \end{bmatrix} \Theta(n) \qquad (13)$$

where $Q(n)W(n)d(n) = [d_N(n)C_{n+1-N}]^{\mathrm{T}}$. Since $Q(n)$ is an unitary matrix, the square of the Euclidean norm on the left-hand side in (13) is equal to $\xi_N(n)$ in (7). The two-norm on the right-hand side of (13) achieves its minimum value when $\Theta(n)$ is chosen as $R_N(n)\Theta(n) = d_N(n)$, and $min\Theta(n)\xi_N(n) = \xi_N^*(n) = \|C_{N+1-N}\|^2$.

Since $R_N(n)$is an upper triangular matrix $\Theta(n)$, can be obtained by back-substitution, there are several methods to perform the QRD of the weighted data matrix $W(n)X_N(n)$. The inverting a matrix [A] using Gaussian elimination has a complexity of $Q(n^3)$[12].

The complexity of matrix inversion in hardware becomes prohibitive for real time applications and large values of *n*. Our goal is to invert a matrix of size (12×12) in hardware. In this paper the results for inverting a matrix of size (8×8) can be presented. the same idea and a slight modification in hardware can be used for larger matrix sizes.

The QR decomposition and systolic arrays that used in the hardware design is shown in the **Figure (2).**

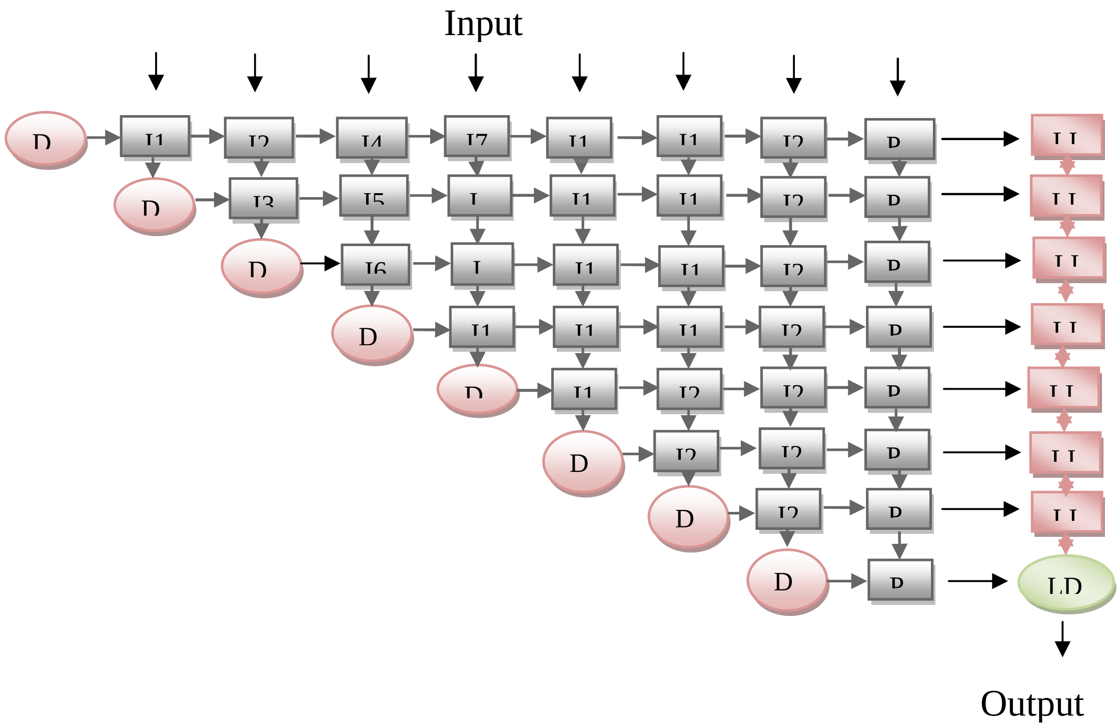


Fig. (2): Systolic Array for QR Decomposition

Let A be $(n \times p)$matrix of full rank *p*. The QR decomposition is decomposing matrix A to a triangular matrix $R_{p\times p}$and an orthogonal matrix Q using plane rotations.

$$A = QR \qquad (14)$$

Rotation

algorithm can be givens rotation or any of its variations such as SGR, SDGR or CORDIC. Then, finding the pseudo-inverse of matrix A, is equal to

$$A^{-1} = (A^H A)^{-1} A^H = (R^H R)^{-1} R^H Q^H \qquad (15)$$

Recursive least square algorithm based on QR decomposition (QRD-RLS) can be used to find the inverse matrix. The main idea of QRD-RLS algorithm is to find a solution for the system of equations:

$$Ax = d \qquad (16)$$

by minimizing the least square error min($|d - A_x|$),this can be done by transforming A to an upper triangular matrix using QR decomposition and systolic arrays and substituting the elements backwards into the equations[10]. By generalizing the above procedure to p dimensions and solving the equation.

. $$AX=I \qquad (17)$$

where *I* is the identity matrix, can be found the inverse of a matrix A, X =A^{-1} The next section presents the SGR, the rotation algorithm used in our architecture [14].

3. Architecture Design

The systolic array for inverting a (8×8) matrix is shown in Figure(3). If the implemented every single node in this diagram, it requires a large area and has very high throughput. In our approach, the combined similar nodes, added memory blocks and a scheduler that controls movement of data between nodes shown in Figure (4). In this figure, boundary and internal cells are same as the ones in Figure(5). The back substitution cell is a combination of eight cells *LI*1*i; LI*2*i; LI*3*iLI*4*i;LI*5*i;LI*6*i;LI*7*i; LDi*.[15]

During the initialization step, the input matrix is stored in A-Mem cells.

cell starts processing it. Because there is no data dependency at the beginning of the process, boundary cell does not have to wait for the initialization step to finish and these two steps can be pipelined. The boundary cell performs vector zing on the values and sends the rotation angle to the internal cell. It computes the equations:[16]

$$\bar{u} = u_k + w\acute{v}_k v \tag{18}$$

$$\overline{w} = {}^{wu_k}/_{\overline{u_k}} \tag{19}$$

$$c = {}^{v_k}/_{u_k} \tag{20}$$

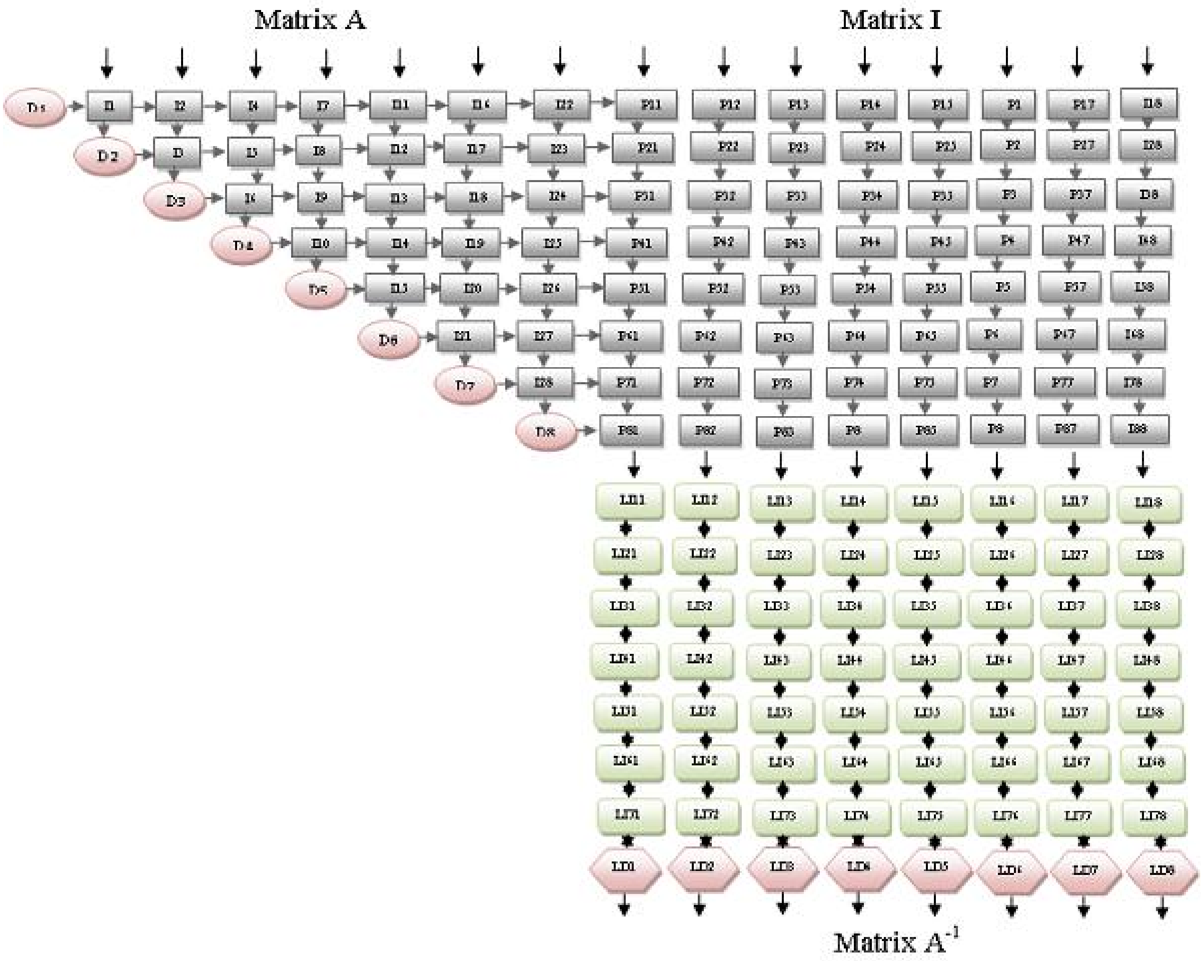


Fig.(3): Systolic Array for 8x8 Matrix Inversion..

Figure (5) shows a block diagram of this cell. Inputs are the values for *u* and v_k and *w* and outputs are the values for$\overline{u}$; $\overline{w}$ and *c*. The outputs of boundary cells enter the internal cells for more processing. They are also stored in the D-Mem memory cells. Boundary cell and internal cell will be active during the next step working on different sets of input data. Internal cell runs on the elements 2 to p in each row of the input matrix *A* and the identity matrix *I* in (17).

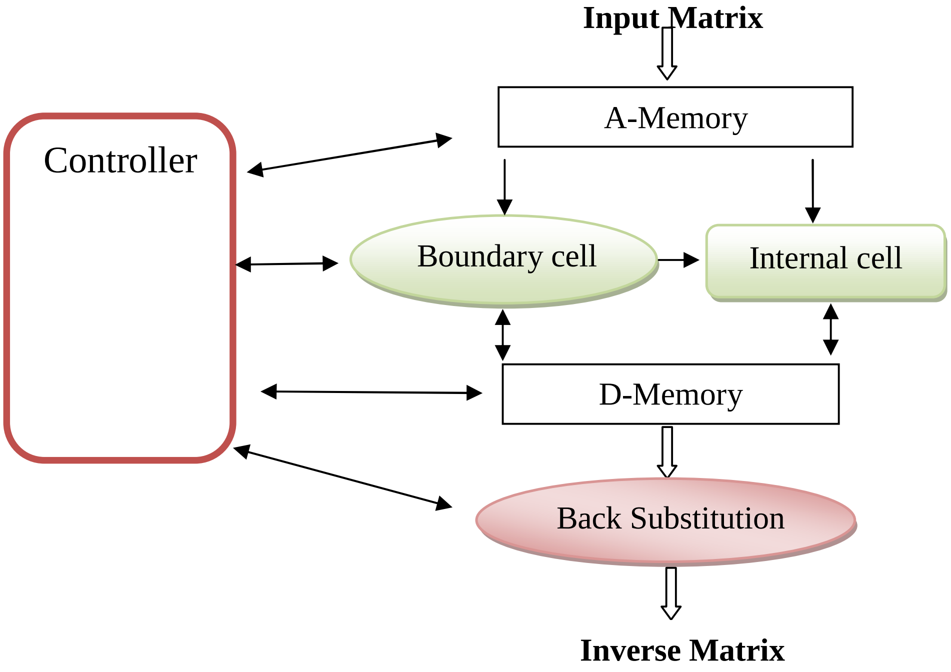


Fig. (4): QRD-RLS Block Diagram

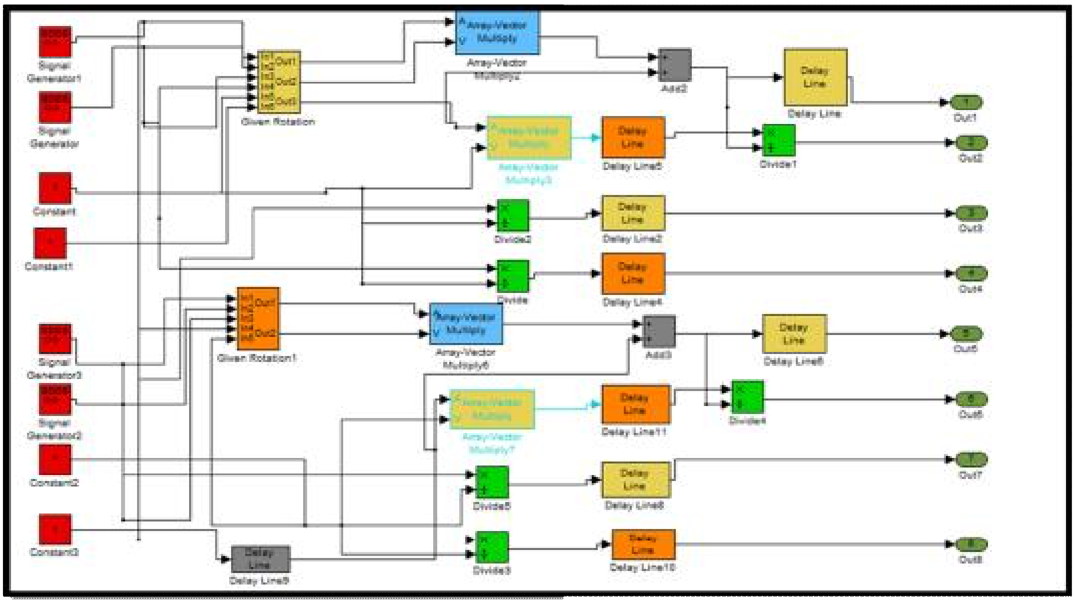


Fig. (5): Boundary Cell Block Diagram

4. FPGA Implementation

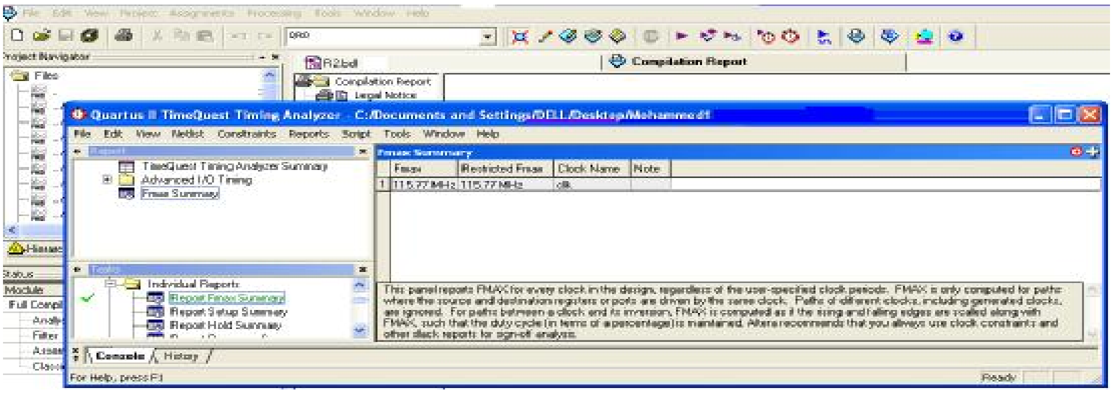
Simulink HDL Coder is high level design tool which generates HDL code from Simulink models and State flow finite state machines. Simulink HDL coder also provides interfaces to combine manually-written HDL codes, HDL simulation blocks and RAM blocks in its environment.

The QRD-RLS Algorithm has been designed, synthesized and simulated using Alter a Quart us II software targeting cyclone III FPGA. It provides a complete design environment for designing system on a program Mable chip.

Cyclone III (*EP3C12F780C7)* device is used to implement and test the matrix inversion design. The Cyclone III board provides three input clocks: 125 MHz, 50 MHz and external clock specified by the user. Therefore, the speed of design should be checked when it is implemented on Cyclone III. The FPGA implementation result shows that the matrix inversion core can achieve a maximum operating frequency of 115.77 MHz, which is very close to the clock frequency of the board (125 MHz) as shown in the reports (Figures 6 and 7). Input is a (8×8)matrix of complex, floating point values and output is the inverse matrix. Table 1, shows the design statistics for a (8×8) matrix inversion core. Can be assumed 2 byte for mantissa, 1 byte for exponent of floating point numbers and one sign bit. For floating point operators (adder and multiplier) can be used available operators from Altera.

On a state of the art cyclone III (*EP3C12F780C7)* FPGA running at 125 MHz, this matrix inversion architecture achieves a throughput of 7.2or 0.14Mupdates per second. By pipeliningthe triangular section and the back substitution part of the design, throughput can increase to 0.15Mupdates per second. The latency for generating the upper triangular matrix is 777 cycles and back substitution has a latency of 156 cycles.

These latencies can be decreased by adding more boundary or internal nodes to the design or decreasing the word length requirements. The design is easily extendable to other matrix sizes of $(n \times p)$ by changing the control unit. There is a tradeoff between number of cells (and hence area of the design) and throughput. For larger matrices, if throughput is less than required, can be increased number of cells and use a semi parallel approach instead of the current folded model.

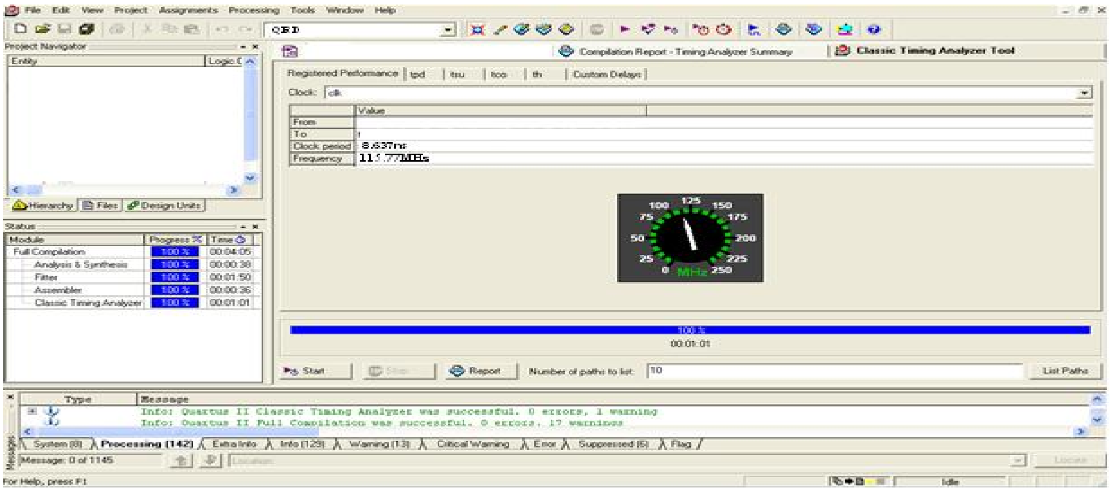


Fmax Summary

	Fmax	Restricted Fmax	Clock Name	Note
1	115.77 MHz	115.77 MHz	clk	

This panel reports FMAX for every clock in the design, regardless of the user-specified clock periods. FMAX is only computed for paths where the source and destination registers or ports are driven by the same clock. Paths of different clocks, including generated clocks, are ignored. For paths between a clock and its inversion, FMAX is computed as if the rising and falling edges are scaled along with FMAX, such that the duty cycle (in terms of a percentage) is maintained. Altera recommends that you always use clock constraints and other slack reports for sign-off analysis.

Fig.(6): report for maximum frequency.



Clock: clk

	Value
From	
To	!
Clock period	8.637ns
Frequency	115.77MHz

Status

Module	Progress %	Time
Full Compilation	100 %	00:04:05
Analysis & Synthesis	100 %	00:00:38
Fitter	100 %	00:01:50
Assembler	100 %	00:00:36
Classic Timing Analyzer	100 %	00:01:01

100 %
00:01:01

Number of paths to list: 10

```
Type    Message
        Info: Quartus II Classic Timing Analyzer was successful. 0 errors, 1 warning
        Info: Quartus II Full Compilation was successful. 0 errors, 17 warnings
```

Message: 0 of 1145

For Help, press F1

Fig.7: Report for maximum frequency by timing analyses tool.

Table1:Resources For 8×8 Matrix Inversion Core On A Cyclone III-FPGA

Synthesis Summary For The Matrix Invertion Core targeted for CycloneIII (EP3C12F780C7) device			
Resources	**Used**	**Available**	**Utilization**
Total logic elements	16197	119088	(13%)
Total registers	2978	119088	(2.5%)
Total memory bits	3660	3981312	(9%)
DSP block 9 bit elements	142	576	(24%)
Maximum frequency	115.77MHz		

To improve the precision in computation, implement floating-point to handle real numbers. However, can be also employed fixed-point arithmetic to reduce hardware usage. Basically, the FPGA implementation is highly application related, however, there still exist some common recognitions.

During our exploring to implement MPC technology into a FPGA chip, have be accumulated some experience which is also useful for FPGA implementation of other control algorithms. Firstly, can be recognized that a good rapid prototyping environment is crucial. It helps to focus on design the control algorithms and implementation architectures. Compared with written VHDL or Verilog, HDL coder is a much easier and efficient tool for control engineers. Moreover, developing a connection channel between MATLAB/Simulink and HDL coder suite is also essential to fasten the design and verifications. Secondly, implementing an algorithm into hardware is much different from running it on a PC. Control algorithms

Should be re-design carefully to fit the FPGA environment. For example, a floating point multiplication unnecessarily inside a loop will cause a lot more chip resource than putting it outside. Every part of the algorithm should be checked carefully before implementing it into an FPGA chip. In doing this, MATLAB is a helpful tool.

5. Conclusions

Improved algorithms for a class of approximate QR-based LS(A-QR-LS) algorithms recently proposed are presented. It is shown that the A-QR-LS algorithm is equivalent to an element-wise normalized LMS algorithm with time-varying step sizes. It reduces to the QR-LMS algorithm when all the normalization constants are chosen as the Euclidean norm of the input signal vector. Designed and implemented a matrix inversion core is on Alter a Cyclone IIIFPGAs using QRD-RLS and Squared Givens Rotation algorithms that enables the run-time definition of the input matrix dimensions is done. The design runs with a clock rate of 125MHz and achieves a throughput of 0.14*M*updates per second.This design is easily extendable to other matrix sizes.

References

1. **M.Karkooti, J.R. Cavallaro" FPGA Implementation of Matrix Inversion Using QRD-RLS Algorithm" *IEEE Communications Magazine*, 1-4244-0132-2005.**
2. **H. Yang, "A road to future broadband wireless access: MIMO-OFDM Based air interface," IEEE Communications Magazine, Vol. 43, Jan 2005, pp. 53– 60.**
3. **L. Kansal, A. Kansal and K. Singh" Performance Analysis of MIMO-OFDM System Using QOSTBC Code Structure for M-QAM" Canadian Journal on Signal Processing ,Vol. 2, No. 2, May 2011.**
4. **J. Yue, K. J. Kim, J. Gibson, and R.A.Iltis, "Channel estimation anddata detection for MIMO-OFDM systems," In Proceedings of IEEE Global Telecommunications Conference, Vol. 2, 1-5 Dec 2003, pp. 581 – 585**
5. **S. Haykin, *Adaptive Filter Theory*. Prentice Hall, third ed.**
6. **R. Dohler, "Squared Givens Rotation," *IMA Journal of Numerical Analysis*, No. 11, pp. 1–5, 1991.**
7. **Ray Andrak, "A survey of CORDIC algorithms for FPGA based computers", Proceedings of the 1998 ACM/SIGDA sixth international symposium on Field programmable gate arrays, March 1998, pp. 191-200.**
8. **R. Ackner, T. Kailath, "The Schur algorithm for matrix-valued meromorphic functions", SIAM J. Matrix Anal. Appl., Vol. 15,1994, pp. 140-150.**
9. **Stephen D. Brown, Robert J. Francis, and Jonathan Rose. Field- Programmable Gate Arrays. Kluwer Academic Publishers, Boston, 1992.**
10. **B.Schoner, J. Villasenor, S. Molloy and R. Jain, "Techniques for FPGA implementation of video compression systems", Proceedings of the 1995 ACM third international symposium on Field programmable gate arrays, February 1995.**
11. **Z. S. Liu, "QR methods of Q(N)complexity in adaptive parameter estimation," IEEE Trans. Signal Processing, Vol. 43, 1995, pp. 720–729.**

12. **Z. S. Liu and J. Li, "A QR-based least mean squares algorithm for adaptive parameter estimation," IEEE Trans. Circuits Syst., Vol. 45, Mar1998, pp.321–329.**
13. **S. H. Leung and C. F. So, "Nonlinear RLS algorithm using variable forgetting factor in mixture noise,"** ***Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*****, Vol. 6,2001, pp. 3777–3780.**

14. **J.Gotze and U.Schwiegelshohn, "A Square Root and Division Free Givens Rotation for Solving Least Square Problems on Systolic Arrays,"** ***J. SCI. STAT. COMPUT.*****, Vol. 12, July 1991, pp. 800–807.**
15. **H.K. W.M.Gentleman."Matrix Traingularization by Systolic Arrays" Vol.298,1981, pp.19-26, IEEE.**
16. **J. Volder, "The CORDIC Trigonometric Computing Technique,"** ***IRE Transactions on Electronic Computers*****, Vol. EC-8, nNo. 3,1959, pp. 330–334.**