iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 3 **Issue:** II **Month of publication:** February 2015

DOI:

ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

FPGA Implementation of Adaptive Weight Calculation Using QRD - RLS Algorithm

H. Rashmi Evangelin[1], Ms. A.S. Narmadaa[2]
[1]PG Scholar, (VLSI Design), [2]Assistant Professor/E.C.E, Madha Engineering College, Kundrathur, Chennai –600069.

Abstract - Adaptive weight calculation (AWC) is required in many communication applications including adaptive beamforming, equalization, predistortion and multiple-input multiple-output (MIMO) systems. These applications involve solving over-determined systems of equations in many cases. In general, the least squares approach, e.g. Least Mean Squares (LMS), Normalized LMS (NLMS) and Recursive Least Squares (RLS), is used to find an approximate solution to these kinds of system of equations. Among them, RLS is most commonly used due to its good numerical properties and fast convergence rate. Applying QR decomposition (QRD) to perform adaptive weight calculation based on RLS avoids this problem and leads to more accurate results and efficient architectures. QR decomposition is a method for solving a set of simultaneous equations, for unknown weights, which define the beam shape. The QR decomposition technique for adaptive weight calculation is particularly suited to implementation in FPGA and FPGA cores are now available that reduce the system development time.
Index Terms – LMS (Least Mean Square), RLS (Recursive Least Square), QRD (Quadratic Rotation Decomposition)

I. INTRODUCTION

Filtering in the most general terms is a process of noise removal from a measured process in order to reveal or enhance information about some quantity of interest. Any real data or signal measuring process includes some degree of noise from various possible sources. The desired signal may have added noise due to thermal or other physical effects related to the signal generation system, or it may noise may get added due to the measuring system or a digital data sampling process. Often the noise is a wide-sense stationary random process (has a constant finite mean and variance, and an autocorrelation function dependent only on the difference between the times of occurrence of the samples), which is known and therefore may be modelled by a common statistical model such as the Gaussian statistical model. It may also be random noise with unknown statistics. Otherwise, it may be a noise that is correlated in some way with the desired signal itself. Filtering, strictly means the extraction of information about some quantity of interest at the current time t by using data measured up to and including the time t. Smoothing, involves a delay of the output because it uses information extracted both after and before the current time t to extract the information. The benefit expected from introducing the delay is more to do with accuracy than filtering. Prediction, involves forecasting information some time into the future given the current and past data at time t and before. De convolution, involves the recovery of the filter characteristics given the filter's input and output signals. Filters can be classified as either linear or nonlinear types. A linear filter is the one whose output is some linear function of the input. In the design of linear filters it is necessary to assume stationarity (statistical-time-invariance) and know the relevant signal and noise statistics a priori. The linear filter design attempts to minimise the effects of noise on the signal by meeting a suitable statistical criterion. The classical linear Wiener filter, for example, minimises the Mean Square Error (MSE) between the desired signal response and the actual filter response. The Wiener solution is said to be optimum in the mean square sense, and it can be said to be truly optimum for second-order stationary noise statistics (fully described by constant finite mean and variance). For non stationary signal and/or noise statistics, the linear Kalman filter can be used. Very well developed linear theory exists for both the Wiener and Kalman filters and the relationships between them.

II. LEAST MEAN SQUARE (LMS)

Least mean square (LMS) algorithms are class of adaptive filter used to mimic a desired filter by finding the filter coefficients that relate to producing the least mean squares of the error signal (difference between the desired and the actual signal). It is a stochastic gradient descent method in that the filter is only adapted based on the error at the current time. The algorithm starts by assuming a small weights (zero in most cases), and at each step, by finding the gradient of the mean square error, the weights are updated. That is, if the MSE-gradient is positive, it implies, the error would keep increasing positively, if the same weight is used for further iterations, which means we need to reduce the weights. In the same way, if the gradient is negative, we need to

ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

increase the weights. So, the basic weight update equation is given in

$$w_{n+1} = w_n - \mu_\Delta \varepsilon[n]$$

where, ε represents the mean-square error. The negative sign indicates that, we need to change the weights in a direction opposite to that of the gradient slope.

A. LMS Algorithm Summary

The LMS algorithm for a p^{th} order algorithm can be summarized as follows:

Parameters: P = filter order
μ = step size
Initialization: $\hat{h}(0) = 0$
Computation: For n = 0, 1, 2...
$X(n) = [x(n), x(n-1), \ldots, x(n-p+1)]^T$
$e(n) = d(n) - \hat{h}^H(n)\, X(n)$
$\hat{h}(n+1) = \hat{h}(n) + \mu\, e^*(n)\, X(n)$

B. Convergence and Stability of LMS

As the LMS algorithm does not use the exact values of the expectations, the weights would never reach the optimal weights in the absolute sense, but a convergence is possible in mean. That is even-though, the weights may change by small amounts, it changes about the optimal weights. However, if the variance, with which the weights change, is large, convergence in mean would be misleading. This problem may occur, if the value of step-size μ is not chosen properly. Thus, an upper bound on μ is needed which is given as $0 < \mu < 2\lambda_{max}$. where λ_{max} is an autocorrelation matrix, its eigen vales are non negative. If this condition is not fulfilled, the algorithm becomes unstable. The convergence of the algorithm is inversely proportional to the eigen value spread of the correlation matrix R. When the eigen values of R are widespread, convergence may be slow. The eigen value spread of the correlation matrix is estimated by computing the ratio of the largest eigen value to the smallest eigen value. If μ is chosen to be very small then the algorithm converges very slowly. A large value of μ may lead to a faster convergence but may be less stable around the minimum value. Maximum convergence speed is achieved in equation

$$\mu = \frac{2}{\lambda max + \lambda min}$$

where λ_{min} is the smallest eigen value of R. Given that μ is less than or equal to this optimum, the convergence speed is determined by λ_{min}, with a larger value yielding faster convergence. This means that faster convergence can be achieved when λ_{max} is close to λ_{min}, that is, the maximum achievable convergence speed depends on the eigen value spread of R.

C. LMS Adaptive Filter

The LMS adaptive filter using distributed arithmetic can be realized by using adders and memories without multipliers, that is, it can be achieved with a small hardware. A Distributed Arithmetic (DA) is an efficient calculation method of an inner product of constant vectors, and it has been used in the DCT realization. Furthermore, it is suitable for time varying coefficient vector in the adaptive filter. Cowan and others proposed a Least Mean Square (LMS) adaptive filter using the DA on an offset binary coding. However, it is found that the convergence speed of this method is extremely degraded. This degradation results from an offset bias added to an input signal coded on the offset binary coding. To overcome this problem, an update algorithm generalized with 2's complement representation has been proposed and the convergence condition has been analyzed. The effective architectures for the LMS adaptive filter using the DA have been proposed. The LMS adaptive filter using distributed arithmetic is expressed by DA-ADF. The DA is applied to the output calculation, i.e., inner product of the input signal vector and coefficient vector. The output signal is obtained by the shift and addition of the partial-products specified with the bit patterns of the N-th order input signal vector. This process is performed from LSB to MSB direction at the every sampling instance, where the B indicates the word length. The B partial-products used to obtain the output signal are updated from LMB to MSB direction. There exist 2N partial-products, and the set including all the partial-products is called Whole Adaptive Function Space (WAFS). Furthermore, the DA-ADF using multi-memory block structure that uses the divided WAFS (MDA-ADF) and the MDA-ADF using half-memory algorithm based on the pseudo-odd symmetry property of the WAFS (HMDA-ADF) have been proposed. The divided WAFS is expressed by DWAFS. An N-tap input signal vector S(k) is represented in equation.

$$S(k) = [s(k), s(k-1), \ldots\ldots, s(k-N+1)]^T$$

where, s(k) is an input signal at k time instance, and the T indicates a transpose of the vector. The output signal of an adaptive

ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

filter is represented in

$$y(k) = S^T(k)\,W(k)$$

where, W(k) is the N-th coefficient vector represented in equation

$$W(k) = [w_0(k), w_1(k), \ldots, w_{N-1}(k)]^T$$

and the $w_i(k)$ is an i-th tap coefficient of the adaptive filter. The Widrow's LMS algorithm is represented in equation

$$W(k+1) = W(k) + 2\mu e(k)S(k)$$

where, the e(k), μ and d(k) are an error signal, a step-size parameter and the desired signal, respectively.

The step-size parameter determines the convergence speed and the accuracy of the estimation. The error signal is obtained by the following equation

$$e(k) = d(k) - y(k)$$

Type equation here.The fundamental structure of the LMS adaptive filter is shown in Figure. The filter input signal s(k) is fed into the delay-line, and shifted to the right direction every sampling instance. The taps of the delay-line provide the delayed input signal corresponding to the depth of delay elements. The tap outputs are multiplied with the corresponding coefficients, the sum of these products is an output of the LMS adaptive filter. The error signal is defined as the difference between the desired signal and the filter output signal. The tap coefficients are updated using the products of the input signals and the scaled error signal.

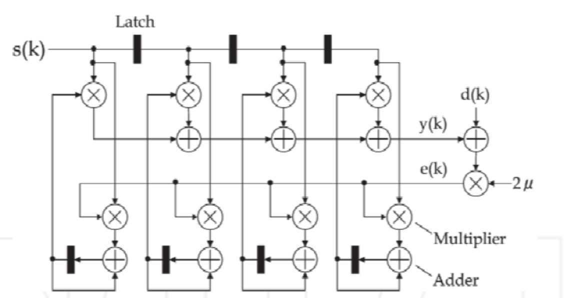


Figure. 2.1. Fundamental Structure of the 4-tap LMS adaptive filter.

III. RECURSIVE LEAST SQUARE (RLS)

The Recursive least squares (RLS) adaptive filter is an algorithm which recursively finds the filter coefficients that minimize a weighted linear least squares cost function relating to the input signals. The RLS algorithms are known for their excellent performance when working in time varying environments but at the cost of an increased computational complexity and some stability problems. In this algorithm the filter tap weight vector is updated using the following equations.

$$w(n) = w^T(n-1) + k(n)\, e_{n-1}(n)$$

$$k(n) = u(n) / (\lambda + X^T(n)\, u(n))$$

$$u(n) = w_{\lambda-1}(n-1)\, X(n)$$

The above two equations are intermediate gain vector used to compute tap weights. Where λ is a small positive constant very close to, but smaller than 1. The filter output is calculated using the filter tap weights of above iteration and the current input vector is given by the following equation

$$y_{n-1}(n) = w^T(n-1)\, X(n)$$

$$e_{n-1}(n) = d(n) - y_{n-1}(n)$$

In the RLS Algorithm the estimate of previous samples of output signal, error signal and filter weight is required that leads to higher memory requirements. The RLS Filter block recursively computes the least squares estimate (RLS) of the FIR filter weights. The block estimates the filter weights, or coefficients, needed to convert the input signal into the desired signal. Connect the signal you want to filter to the Input port. The input signal can be a scalar or a column vector. Connect the signal you want to model to the Desired port. The desired signal must have the same data type, complexity, and dimensions as the input signal. The Output port outputs the filtered input signal. The Error port outputs the result of subtracting the output signal from the desired signal. The corresponding RLS filter is expressed in matrix form as follows

ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

$$k(n) = \frac{\lambda^{-1}P(n-1)u(n)}{1+\lambda^{-1}u^{H}(n)P(n-1)u(n)}$$

$$y(n) = w(n-1)u(n)$$

$$e(n) = d(n) - y(n)$$

$$w(n) = w(n-1) + k^{H}(n)e(n)$$

$$P(n) = \lambda^{-1}P(n-1) - \lambda^{-1}1\,k(n)u^{H}(n)p(n-1)$$

where λ^{-1} denotes the reciprocal of the exponential weighting factor.

IV. QR DECOMPOSITION

In linear algebra, a QR decomposition (also called a QR factorization) of a matrix is a decomposition of a matrix A into a product A = QR of an orthogonal matrix Q and an upper triangular matrix R. QR decomposition is often used to solve the linear least squares problem, and is the basis for a particular eigenvalue algorithm, the QR algorithm. If A has n linearly independent columns, then the first n columns of Q form an orthonormal basis for the column space of A. More specifically, the first k columns of Q form an orthonormal basis for the span of the first k columns of A for any $1 \le k \le n$. The fact that any column k of A only depends on the first k columns of Q is responsible for the triangular form of R.

A. QR Decomposition Methods

There are three different QR decomposition methods: Gram-Schmidt orthogonormalization, Givens Rotations (GR) and Householder reflections. GR is preferred because of its stability and accuracy. GR lends itself easily to a systolic array architecture using CORDIC blocks which makes an efficient hardware implementation. Therefore, it is often used for hardware implementation. However, it was shown that the modified Gram-Schmidt (MGS) method is numerically equivalent to Givens rotations method. A wide variety of computationally intensive applications are moving from Digital Signal Processors (DSPs) to Field Programmable Gate Arrays (FPGAs) because FPGA architectures present designers with substantially more parallelism allowing more efficient application implementations. Moreover, FPGAs are a flexible, cost effective alternative to Application Specific Integrated Circuits (ASICs). FPGAs are perfect platforms for arithmetic operations such as matrix decomposition as they provide powerful computational architectural features, e.g. embedded multipliers, shift register LUTs (SRLs), Block RAMs (BRAMs), DSP blocks and DCMs (Digital Clock Managers). If used correctly, these features can enhance the performance and throughput significantly. We will discuss the design decisions that we encountered as we customized our design to utilize the FPGA architectural features.

B. Computing the QR Decomposition

There are several methods for actually computing the QR decomposition, such as by means of the Gram–Schmidt process, Householder transformations, or Givens rotations. Each has a number of advantages and disadvantages.

1) *Using the Gram–Schmidt process:* Consider the Gram – Schmidt process applied to the columns of the full column rank matrix $A = [\mathbf{a}_1, \cdots, \mathbf{a}_n]$, with inner product $\langle \mathbf{v}, \mathbf{w} \rangle = \mathbf{v}^\top \mathbf{w}$ or $\langle \mathbf{v}, \mathbf{w} \rangle = \mathbf{v}^* \mathbf{w}$ for the complex case

$$\mathrm{proj}_{\mathbf{e}}\mathbf{a} = \frac{\langle \mathbf{e}, \mathbf{a} \rangle}{\langle \mathbf{e}, \mathbf{e} \rangle}\mathbf{e} \qquad (4.1)$$

then:

$$\begin{aligned}
\mathbf{u}_1 &= \mathbf{a}_1, & \mathbf{e}_1 &= \frac{\mathbf{u}_1}{\|\mathbf{u}_1\|} \\
\mathbf{u}_2 &= \mathbf{a}_2 - \mathrm{proj}_{\mathbf{e}_1}\mathbf{a}_2, & \mathbf{e}_2 &= \frac{\mathbf{u}_2}{\|\mathbf{u}_2\|} \\
\mathbf{u}_3 &= \mathbf{a}_3 - \mathrm{proj}_{\mathbf{e}_1}\mathbf{a}_3 - \mathrm{proj}_{\mathbf{e}_2}\mathbf{a}_3, & \mathbf{e}_3 &= \frac{\mathbf{u}_3}{\|\mathbf{u}_3\|} \\
&\vdots & &\vdots \\
\mathbf{u}_k &= \mathbf{a}_k - \sum_{j=1}^{k-1}\mathrm{proj}_{\mathbf{e}_j}\mathbf{a}_k, & \mathbf{e}_k &= \frac{\mathbf{u}_k}{\|\mathbf{u}_k\|}
\end{aligned} \qquad (4.2)$$

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

We then rearrange the equations (4.2) above so that the $\mathbf{a}_i$s are on the left, using the fact that the $\mathbf{e}_i$ are unit vectors.

a) Example

Consider the decomposition of (4.3)

$$A = \begin{pmatrix} 12 & -51 & 4 \\ 6 & 167 & -68 \\ -4 & 24 & -41 \end{pmatrix}. \qquad (4.3)$$

Recall that an orthogonal matrix Q has the property (4.4)

$$Q^T Q = I. \qquad (4.4)$$

Then, we can calculate Q by means of Gram–Schmidt as follows in equation (4.5) and (4.6)

$$U = \begin{pmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \mathbf{u}_3 \end{pmatrix} = \begin{pmatrix} 12 & -69 & -58/5 \\ 6 & 158 & 6/5 \\ -4 & 30 & -33 \end{pmatrix}; \qquad (4.5)$$

$$Q = \begin{pmatrix} \frac{\mathbf{u}_1}{\|\mathbf{u}_1\|} & \frac{\mathbf{u}_2}{\|\mathbf{u}_2\|} & \frac{\mathbf{u}_3}{\|\mathbf{u}_3\|} \end{pmatrix} = \begin{pmatrix} 6/7 & -69/175 & -58/175 \\ 3/7 & 158/175 & 6/175 \\ -2/7 & 6/35 & -33/35 \end{pmatrix}. \qquad (4.6)$$

Thus, we have transpose and R value in equations (4.7) and (4.8)

$$Q^T A = Q^T Q\, R = R; \qquad (4.7)$$

$$R = Q^T A = \begin{pmatrix} 14 & 21 & -14 \\ 0 & 175 & -70 \\ 0 & 0 & 35 \end{pmatrix}. \qquad (4.8)$$

V. CONCLUSION AND RESULTS

A. Conclusion

QR decomposition of matrix is one of the important problems in the field of matrix theory. Besides, there are also so many extensive applications that using QR decomposition. The QR decomposition is often used for counting the eigen values from giant matrix or for solving the least square problem. Therefore, the QR decomposition is not only an important problem in matrix theory, but also has an extensive application prospect. Adaptive beamforming is a commonly employed technique in which the system is able to operate in an interference environment by adaptively modifying the antenna array pattern so that the nulls are formed in the angular locations of the interference sources. In this work we compared the algorithms for adaptive beamforming such as LMS and RLS. The comparison is based on the MSE and the Weight Error in db. It can be shown that RLS performs well over the LMS while considering the two parameters. The error rate of the RLS is low as compared with the LMS. Hence the RLS provide better performance than the RLS algorithm.

B. Future Enhancement

In future enhancement, the QRD - RLS of FPGA implementation is adapted using CORDIC blocks. Here we reduce the number of adders by using the QRD – RLS algorithm. So the iteration processes get reduced. Obviously the power in the circuit is

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

effectively reduced.

C. Experimental Results

The following figure shows a simulation results for least mean square.

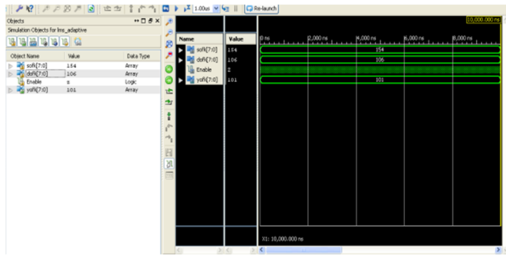


Figure 5.1 Simulation results for LMS

The following figure shows a simulation results for Recursive least square.

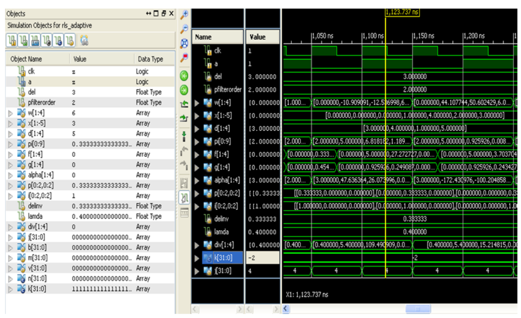


Figure 5.2 Simulation results for RLS

The following figure shows a simulation results for QR Decomposition

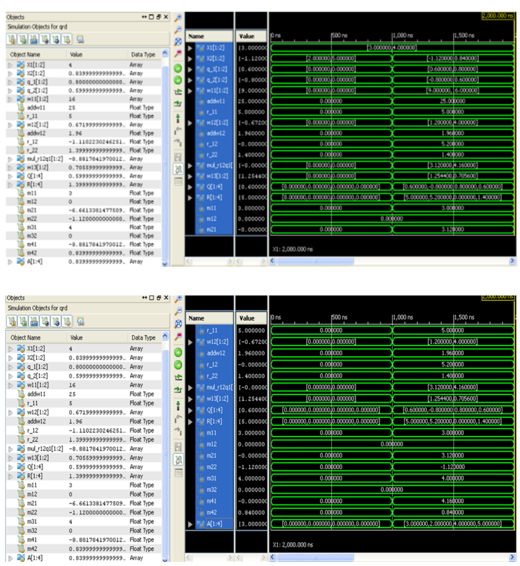


Figure 5.3 Simulation results for QR Decomposition

REFERENCES

[1] Asit Kumar Subudhi, Biswajit Mishra, Mihir Narayan Mohanty, " VLSI Design and Implementation for Adaptive Filter using LMS Algorithm ",

ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

International Journal of Computer & Communication Technology (IJCCT), Volume-2, Issue-VI, 2011.

[2] Christian Drewes, Ralph Hasholzner, Joachim S. Hammerschmidt " On The Implementation Of Adaptive Equalizers For Wireless Atm With An Extended Qr-Decomposition-Based Ius-Algorithm ", IEEE, 1998.

[3] Deshpande .P, Govind Rao .D, Murthy N. S, Engadarajan A. V, " Efficient Filter Implementation Using QRD-RLS Algorithm for Phased Array Radar Applications ", IEEE, 2013.

[4] Jian-Feng Gu, Chan .S. C, Wei-Ping Zhu, and Swamy .M. N. S, " Joint DOA Estimation and Source Signal Tracking With Kalman Filtering and Regularized Qrd Rls Algorithm ", IEEE Transactions On Circuits And Systems—Ii: Express Briefs, VOL. 60, NO. 1, 2013.

[5] Kapil Dungriyal, Anand.S, Sriram Kumar.D, " Performance Of Mir-Lms Algorithm For Adaptive Beam Forming In Smart Antenna ", Irf International Conference, 2014.

[6] Khyati R.Zalawadia, Twinkle V. Doshi 2, Dalal Dr .U.D, " Adaptive Beam Former Design using RLS Algorithm for Smart Antenna System ", (ICCICS) International Conference on Computational Intelligence and Communication Systems, IEEE, 2011.

[7] Kunsheng Ren, Taijun Liu, Yan Ye, Xingbin Zeng " Identification of Memory Polynomial Nonlinear Models for RF Power Amplifiers with a Systolic Array Based QRD-RLS Algorithm ", IEEE, 2008.

[8] Lei Wang and Rodrigo C. de Lamare " Blind Adaptive Beamforming Based on Constrained Constant Modulus RLS Algorithm for Smart Antennas " , Communication Research Group , 2013.

[9] Lei Wang and Rodrigo C. de Lamare " Constrained Constant Modulus RLS-based Blind Adaptive Beamforming Algorithm for Smart Antennas ", IEEE ISWCS, 2007.

[10] Martínez .M. E. I " Implementation of QRD-RLS algorithm on FPGA. Application to Noise Canceller System ", IEEE Latin America Transactions, Vol. 9, No. 4, 2011.

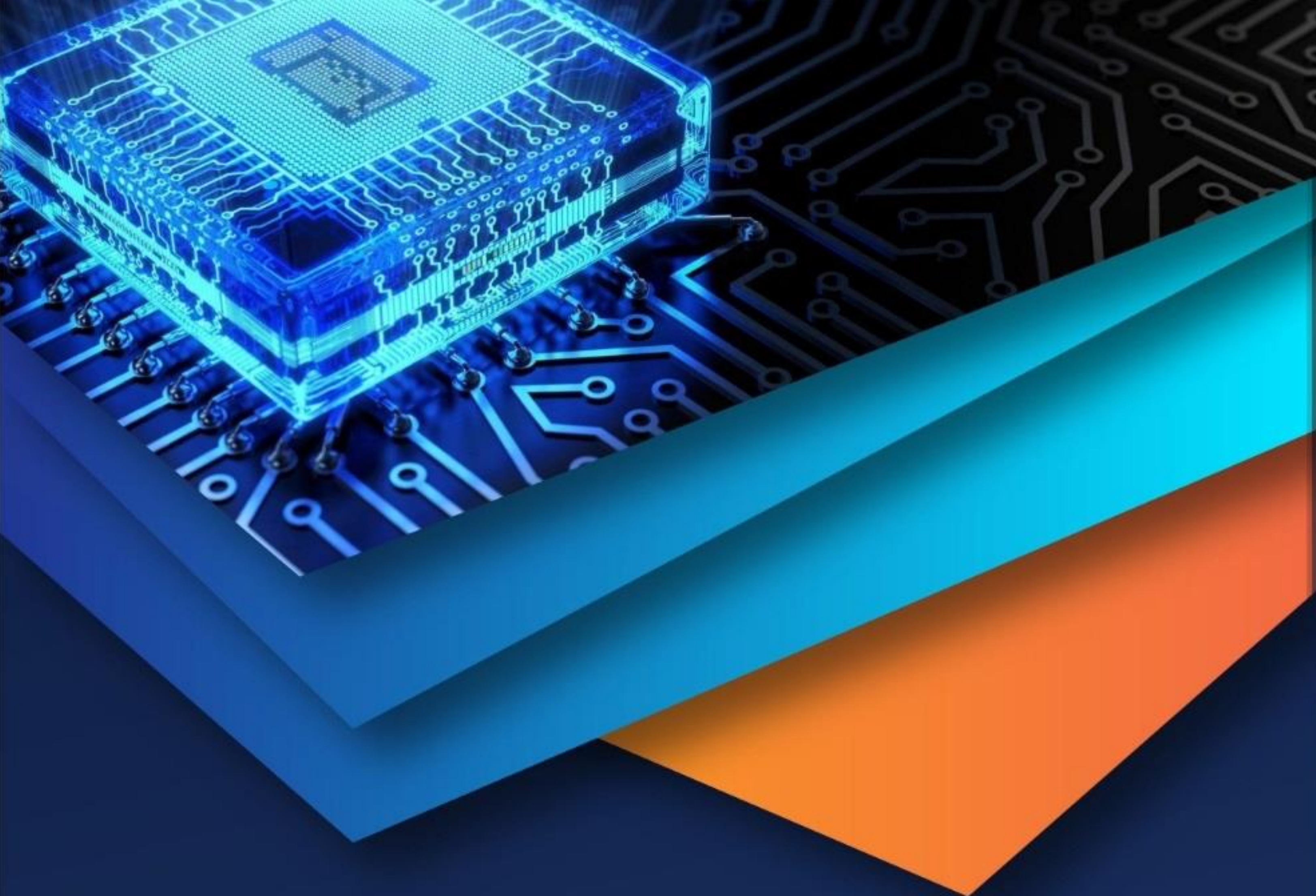
[11] Mobien Shoaib, Stefan Werner, and José Antonio Apolinário " Multichannel Fast QR-Decomposition Algorithms: Weight Extraction Method and Its Applications ", IEEE Transactions On Signal Processing, Vol. 58, No. 1, 2010.

[12] Mobien Shoaib, Stefan Werner, Jos´e Apolin´ario A, and Timo I. Laakso1 " Multichannel Fast Qr-Decomposition Rls Algorithms With Explicit Weight Extraction ", IEEE, 2007.

[13] Namita Sharma, Preeti Ranjan Panda, Min Li, Prashant Agrawal, and Francky Catthoor "Energy Efficient Data Flow Transformation for Givens Rotation based QR Decomposition", IMEC, 2014.

[14] Nandana. B.T, Kumaresan .A, Kavitha.K. " Performance Analysis and Comparison of LMS and RLS Beamforming Technology ", PG students, Communication systems, 3Associate Professor, Dept of ECE, Kumaraguru College of Technology , 2014.

[15] Pramod Kumar Meher, and Sang Yoon Park, " Area-Delay-Power Efficient Fixed-Point LMS Adaptive Filter With Low Adaptation-Delay ", IEEE Transactions On Very Large Scale Integration (VLSI) Systems, Vol. 22, No. 2, 2014.



10.22214/IJRASET

INDEX COPERNICUS
45.98

IMPACT FACTOR:
7.129

TOGETHER WE REACH THE GOAL
IMPACT FACTOR:
7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY