International Journal of Electronics Communication and Computer Engineering
Volume 3, Issue 6, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209

FPGA Implementation of EVD Processor for Fast ICA Application

K. Samyuktha
M.Tech. Student,
samyuktha.kalali@gmail.com

Dr. K. S. Rao
Director, CVSR College of Engineering
ksrdiragi@gmail.com

***Abstract* – A novel hardware architecture based on the extended Jacobi method is presented to solve the calculation of eigenvectors and eigen values inside a FPGA, with high accuracy, taking low execution time and using a reduced number of internal resources. Besides, the execution time is lower than an optimal algorithm to solve the eigen problem in any circuitry. Extended Jacobi method is used to calculate eigen vectors for any real time matrix with less number of operations and with less processing time. This decomposition generally goes under the name "matrix diagonalization." However, this moniker is less than optimal, since the process being described is really the decomposition of a matrix into a product of three other matrices, only one of which is diagonal, and also because all other standard types of matrix decomposition use the term "decomposition".**

***Keywords* – E.V.D, ICA, Eigen, Digitalization, P.C.A, Gaussian, Jacobi eigenvalue algorithm, ecomposition.**

INTRODUCTION

Independent component analysis (ICA) is a statistical and computational technique for revealing hidden factors that underlie sets of random variables, measurements, or signals. ICA defines a generative model for the observed multivariate data, which is typically given as a large database of samples. In the model, the data variables are assumed to be linear mixtures of some unknown latent variables, and the mixing system is also unknown. The latent variables are assumed non Gaussian and mutually independent and they are called the independent components of the observed data. These independent components, also called sources or factors, can be found by ICA. ICA is superficially related to principal component analysis and factor analysis. ICA is a much more powerful technique, however, capable of finding the underlying factors or sources when these classic methods fail completely. The data analyzed by ICA could originate from many different kinds of application fields, including digital images, document databases, economic indicators and psychometric measurements. In many cases, the measurements are given as a set of parallel signals or time series; the term blind source separation is used to characterize this problem. Typical examples are mixtures of simultaneous speech signals that have been picked up by several microphones, brain waves recorded by multiple sensors, interfering radio signals arriving at a mobile phone, or parallel time series obtained from some industrial process. When the independence assumption is correct, blind ICA separation of a mixed signal gives very good results. It is also used for signals that are not supposed to be generated by a mixing for analysis purposes. A simple application of ICA is the "cocktail party problem", where the underlying speech signals are separated from a sample data consisting of people talking simultaneously in a room. Usually the problem is simplified by assuming no time delays or echoes. An important note to consider is that if N sources are present, at least N observations (e.g. microphones) are needed to get the original signals. This constitutes the square case ($J = D$, where D is the input dimension of the data and J is the dimension of the model).

The calculation of eigenvalues and eigenvectors is a problem that appears in many practical applications of different scientific areas (artificial vision, digital signal processing, power electronics, etc.). Concretely this work is applied in the PCA algorithm of artificial vision (Principal Component Analysis). In any case, the proposed solution can be extended to any other application that needs to solve the eigen problem. The only condition that must be fulfilled is that the input matrix must be symmetrical and with elements belong to the real type, which happens in many practical applications. Respect the maximum matrix size in our work is limited to the FPGA resources. The usual maximum matrix size does not go over 20x20 in practical applications using PCA. In any case, the proposed hardware solution presented herein for the calculation of eigenvalues and eigenvectors, allows easily modifying the HW design to matrices of greater size.

JACOBI EIGENVALUE ALGORITHM

In numerical linear algebra, the Jacobi eigenvalue algorithm is an iterative method for the calculation of the eigenvalues and eigenvectors of a real symmetric matrix. It is named after Carl Gustav Jacob Jacobi, who first proposed the method in 1846.

Jacobi's method is an easily understood algorithm for finding all eigenpairs for a symmetric matrix. It is a reliable method that produces uniformly accurate answers for the results. For matrices of order up to 10×10, the algorithm is competitive with more sophisticated ones. If speed is not a major consideration, it is quite acceptable for matrices up to order 20×20. A solution is guaranteed for all real symmetric matrices when Jacobi's method is used. This limitation is not severe since many practical problems of applied mathematics and engineering involve symmetric matrices. From a theoretical viewpoint, the method embodies techniques that are found in more sophisticated algorithms. For instructive purposes, it is worthwhile to investigate the details of Jacobi's method.

Jacobi Series of Transformations start with the real symmetric matrix $\mathbf{A}$. Then construct the sequence of orthogonal matrices $\mathbf{R}_1, \mathbf{R}_2, \ldots, \mathbf{R}_n$ as follows:

$$\mathbf{D}_0 = \mathbf{A}$$

and

$$\mathbf{D}_j = \mathbf{R}_j^T \mathbf{D}_j \mathbf{R}_j$$

for j=1,2, ...

It is possible to construct the sequence $\{\mathbf{R}_j\}$ so that

$$\lim_{j\to\infty} \mathbf{D}_j = \mathbf{D} = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$$

In practice we will stop when the off-diagonal elements are close to zero. Then we will have

$$\mathbf{D}_m \approx \mathbf{D}.$$

Let S be a symmetric matrix, and $G = G(i,j,\theta)$ be a Givens rotation matrix. Then:

$$S' = G^\top S G$$

is symmetric and similar to S.

Furthermore, S' has entries:

$$S'_{ii} = c^2 S_{ii} - 2sc\, S_{ij} + s^2 S_{jj}$$

$$S'_{jj} = s^2 S_{ii} + 2sc\, S_{ij} + c^2 S_{jj}$$

$$S'_{ij} = S'_{ji} = (c^2 - s^2) S_{ij} + sc\,(S_{ii} - S_{jj})$$

$$S'_{ik} = S'_{ki} = c\,S_{ik} - s\,S_{jk} \qquad k \neq i,j$$

$$S'_{jk} = S'_{kj} = s\,S_{ik} + c\,S_{jk} \qquad k \neq i,j$$

$$S'_{kl} = S_{kl} \qquad k,l \neq i,j$$

where $s = \sin(\theta)$ and $c = \cos(\theta)$.

Since G is orthogonal, S and S' have the same Frobenius norm $\|\cdot\|_F$ (the square-root sum of squares of all components), however we can choose θ such that $S'_{ij} = 0$, in which case S' has a larger sum of squares on the diagonal:

$$S'_{ij} = \cos(2\theta) S_{ij} + \tfrac{1}{2}\sin(2\theta)(S_{ii} - S_{jj})$$

Set this equal to 0, and rearrange:

$$\tan(2\theta) = \frac{2S_{ij}}{S_{jj} - S_{ii}}$$

if $S_{jj} = S_{ii}$

$$\theta = \frac{\pi}{4}$$

In order to optimize this effect, S_{ij} should be the largest off-diagonal component, called the *pivot*.

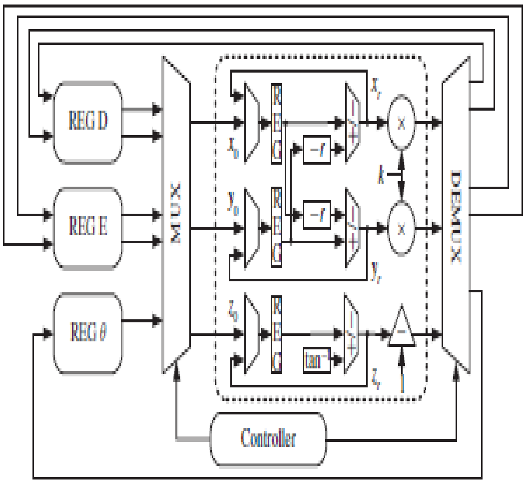
The Jacobi eigenvalue method repeatedly performs rotations until the matrix becomes almost diagonal. Then the elements in the diagonal are approximations of the (real) eigenvalues of S.

IMPLEMENTATION DESIGN:

The Jacobi method managing 3x3 matrices [2], represents a regular design suitable for implementation in reconfigurable HW such as FPGA's [5]. The requirement in area resources of the FPGA is too large if both systolic architectures are implemented (one for eigenvalues and another one for eigenvectors), that are based on processors with CORDIC modules that use a high number of internal resources.

The main goal of our proposal design must be reduce the maximum number of CORDIC modules needed. This reduction can be appliedfrom two points of view:

- Multiplexing the CORDIC modules while computingdouble rotations, reducing the number of modules CORDIC but increasing the execution time.
- Taking advantage of the symmetric property of the input matrix, to reduce the number of symmetrical processors. Thus the systolic array would be transformed into a triangular array (only processors in the diagonal and the lower or upper part of the diagonal).



The CORDIC algorithm that performs iterative operations with shift-and-add behavior for vector rotations by arbitrary angles [29], [30] is used to implement the cyclic Jacobi method. The CORDIC algorithm is operated in two modes including the vectoring mode and the rotation mode. In this project, two modes for vectoring and rotation are implemented in one CORDIC engine to reduce hardware cost. One CORDIC-engine-based EVD processor architecture asshown in Fig. adopts the floating-point operation, where the controller is used to determine which mode can be executed. The CORDIC engine in the vectoring mode can obtain $zr = z0+\tan-1(y0/x0)$ for inputs $x0$, $y0$ and $z0$ [31] with r CORDIC iterations. In order to obtain the optimal angle in the (p, q) plane, $bqq-bpp$ feeds to $x0$, $2bpq$ which is obtained by adding one for the exponential term feeds to $y0$, and an initial value zero feeds to $z0$ in Fig. 4. After r CORDIC iterations in the vectoring mode, the CORDIC engine outputs zr and then subtracts one from the exponential term of zr to obtain the optimal angle in (24) that stored at the register θ. Thus, (24) is realized by shift-and-add operation and arctan table withoutthe dedicated division and arctan computation hardware. In other words, the vectoring mode is used to generate the optimal angle in (24). Next, the corresponding vectors are rotated in the rotation mode with the optimal angle and are saved at the register D. Register E is operated in a similar behavior. Registers D and E are used to store eigenvalues and eigenvectors, respectively. Since the covariance matrix is a symmetric matrix, some rotation results are the same in the plane, e.g., **J**T [$bp1$ $bq1$]T **J** equals **J**T [$b1p$ $b1q$]T **J**. Thus,

International Journal of Electronics Communication and Computer Engineering
Volume 3, Issue 6, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209

only two vectors [*bpp bqp*]*T* and [*bpq bqq*]*T* are needed to rotatein the second rotations [30].

APPLICATIONS

Separation of Artifacts in MEG Data:

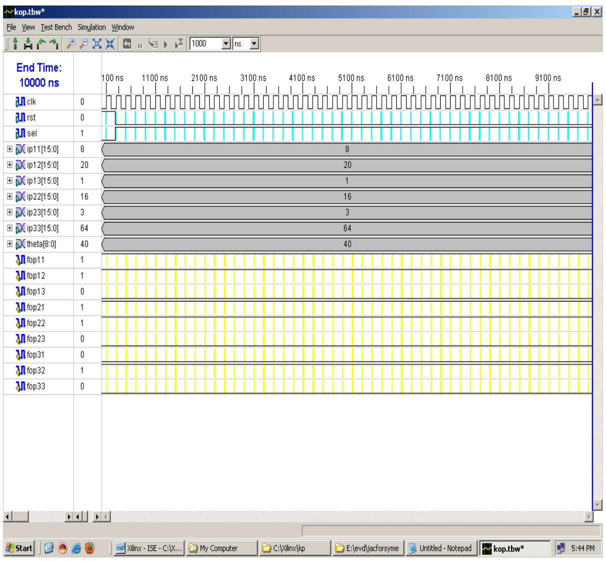
Magnetoencephalography (MEG) is a noninvasive technique by which the activity or the cortical neurons can be measured with very good temporal resolution and moderate spatial resolution. When using a MEG record, as a research or clinical tool, the investigator may face a problem of extracting the essential features of the neuromagnetic signals in the presence of artifacts. The amplitude of the disturbances may be higher than that of the brain signals, and the artifacts may resemble pathological signals in shape.

Finding Hidden Factors in Financial Data:

It is a tempting alternative to try ICA on financial data. There are many situations in that application domain in which parallel time series are available, such as currency exchange rates or daily returns of stocks, that may have some common underlying factors. ICA might reveal some driving mechanisms that otherwise remain hidden. In a recent study of a stock portfolio (Back and Weigend, 1997), it was found that ICA is a complementary tool to PCA, allowing the underlying structure of the data to be more readily observed.

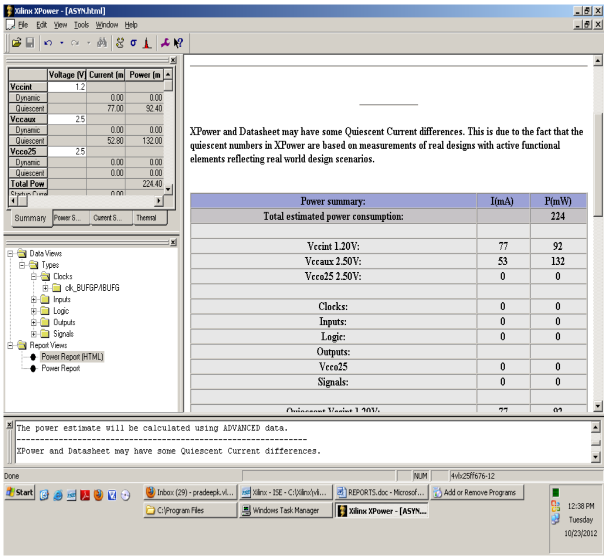
RESULTS

Simulation:



Synthesis:

Power Report



Area Report

Number of Slices:	11	out of	10752	0%
Number of Slice Flip Flops:	4	out of	21504	0%
Number of 4 input LUTs:	21	out of	21504	0%
Number of bonded IOBs:	12	out of	448	2%
Number of GCLKs:	1	out of	32	3%

CONCLUSION

Finally, a modified approach for deriving Eigan vectors from the modularity of the proposed system allows an easy adaptation to different matrix sizes. It can be used in any application, as long as the input matrix is square, real and symmetrical. Computing the eigenvalue decomposition (EVD) of a symmetric matrix is a frequently encountered problem in adaptive (or smart or software) antenna signal processing, for example, super resolution DOA (direction of arrival) estimation algorithms such as MUSIC (multiple signal classification) and ESPRIT (estimation of signal parameters via rotational invariance technique). In this paper the hardware architecture of the fast EVD processor of a symmetric correlation matrix for the application of an adaptive antenna technology such as DOA estimation is proposed and the basic idea is also implemented.

REFERENCES

[1] Abdi, H. (2003). Multivariate analysis. In M. Lewis-Beck, A. Bryman, & T. Futing (Eds): *Encyclopedia for research methods for the social sciences*. Thousand Oaks: Sage.

[2] Abdi, H., Valentin, D. (2006). *Mathématiques pour les sciences cognitives (Mathematics for cognitive sciences)*. Grenoble: PUG.

[3] Harris, R.j. (2001). *A primer of multivariate statistics*. Mahwah (NJ): Erlbaum.

[4] Strang, G. (2003). *Introduction to linear algebra*. Cambridge (MA):Wellesley-Cambridge Press.

[5] Ahmedsaid, A.; Amira, A.; Bouridane, A.; “Accelerating MUSIC method on reconfigurable hardware for source localisation”. Proceed. of the Intern. Symposium on Circuitsand Systems (ISCAS). Vol. 3, pp.:III - 369-72, 2004.

[6] A. Kachenoura, L. Albera, L. Senhadji, and P. Comon, “ICA: A potential tool for BCI systems,” *IEEE Signal Process. Mag.*, vol. 25, no. 1, pp. 57– 68, Jan. 2008.

[7] A. J. Bell and T. J. Sejnowski, “An information maximization approach to blind separation and blind deconvolution,” *Neural Comput.*, vol. 7, no. 6, pp. 1129–1159, Nov. 1995.

[8] A. Hyvärinen and E. Oja, “A fast fixed-point algorithm for independent component analysis,” *Neural Comput.*, vol. 9, no. 7, pp. 1483–1492, Oct. 1997.

[9] A. Hyvärinen, “Fast and robust fixed-point algorithms for independent component analysis,” *IEEE Trans. Neural Netw.*, vol. 10, no. 3, pp. 626– 634, May 1999.

[10] A. Hyvärinen and E. Oja, “Independent component analysis: Algorithms and applications,” *Neural Netw.*, vol. 13, nos. 4–5, pp. 411–430, Jun. 2000.

[11] E. Oja and Z. Yuan, “The fastICA algorithmrevisited: onvergenceanalysis,” *IEEE Trans. Neural Netw.*, vol. 17, no. 6, pp. 1370–1381, Nov. 2006.

[12] E. Ollila, “The deflation-based fastICA estimator: Statistical analysis revisited,” *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1527–1541, Mar. 2010.

[13] S. Choi, A. Cichocki, and S. Amari, “Flexible independent component analysis,” *J. VLSI Signal Process.*, vol. 26, nos. 1–2, pp. 25–38, 2000.

[14] V. Zarzoso, P. Comon, and R. Phlypo, “A contrast function for independent component analysis without permutation ambiguity,” *IEEE Trans. Neural Netw.*, vol. 21, no. 5, pp. 863–868, May 2010.

[15] Jacobi, C.G.J. (1846). "Über ein leichtes Verfahren, die in der Theorie der Säkularstörungen vorkommenden Gleichungen numerisch aufzulösen" (in German). *Crelle's Journal* **30**: 51–94. Golub, G.H.; van der Vorst, H.A. (2000). "Eigenvalue computation in the 20th century". *Journal of Computational and Applied Mathematics* **123** (1-2): 35–65. doi:10.1016/S0377-0427(00)00413-1

[16] Schönhage, A. (1964). "Zur quadratischen Konvergenz des Jacobi-Verfahrens" (in German). *Numerische Mathematik* **6** (1): 410–412. doi:10.1007/BF01386091. MR 174171.

[17] G. Zhou, Z. Yang, S. Xie, and J. M. Yang, “Online blind source separation using incremental nonnegative matrix factorization with volume constraint,” *IEEE Trans. Neural Netw.*, vol. 22, no. 4, pp. 550–560, Apr. 2011.