

A New Architecture Using Polynomial Matrix Multiplication for Advanced Wireless Communication

U Rajpriya¹, V Filomin Joseena²,

¹ PG student, ECE (VLSI Design), Kings College of Engineering, Tamil Nadu, India

² Associate professor ECE, Kings College of Engineering, Tamil Nadu, India

Abstract - In this proposed system, a novel reconfigurable architecture for computing the Polynomial Matrix Multiplication (PMM) of polynomial matrices and/or polynomial vectors for application in Advanced Wireless Communication and an algorithm for computing the approximate polynomial matrix Eigen Value Decomposition (EVD) is introduced. The proposed algorithm exploits an extension of the fast convolution technique to multiple-input multiple-output systems. The proposed architecture is the first one devoted to the hardware implementation of PMM. The architecture, which is scalable in terms of the order of the input polynomial matrices, has been designed using the Xilinx system generator tool for advanced wireless communication. The application to sensor array signal processing is highlighted, in terms of strong de-correlation. The results are presented to demonstrate the accuracy and capability of the architecture. The performance results prove that the proposed solution gives low execution times while limiting the number of required resources. The parameters of the architecture are proved to be the best outcomes, when compared to the conventional approach in terms of Area, Power and Speed.

Key Words: PMM, EVD, SVD, etc...

1. INTRODUCTION

Polynomial matrix techniques equivalent to the singular value decomposition and Eigen Value Decomposition (EVD) for scalar matrices have received growing interest in recent years. They have been successfully applied to broadband extensions of narrowband problems, which traditionally have been addressed by the EVD. Applications include broadband Sensor Array Signal Processing (SASP), biomedical engineering, Multiple-Input Multiple-Output (MIMO) communications and coding, and sub-band coding. The EVD of a para-Hermitian system, or Polynomial Matrix EVD (PEVD), yields a factorization of a para-Hermitian polynomial matrix into a product consisting of a diagonal polynomial matrix that is pre and

post multiplied by Para Unitary (PU) polynomial matrices. A PU polynomial matrix preserves the total signal power at every frequency, and so can be viewed as a lossless (stable, all-pass) filter bank. McWhirter *et al.* propose an extension of the EVD to para-Hermitian polynomial matrices, called the second order Sequential Best Rotation (SBR2) algorithm. It was originally developed for the purpose of generating a Finite Impulse Response (FIR) PU matrix to diagonalize the paraHermitian polynomial matrix of signals received by a broadband sensor array.

2. ARCHITECTURE FOR POLYNOMIAL MATRIX MULTIPLICATION

The two fast MIMO convolution techniques described in this section are fundamental to the aim of applying hardware implemented PEVD algorithms, particularly SBR2P, to high-speed or real-time problems.

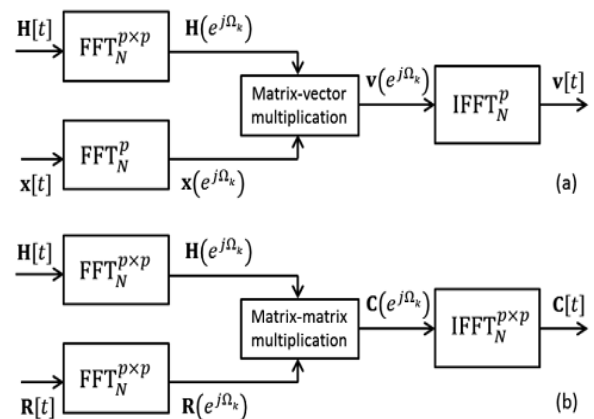


Fig 1: (a) matrix-vector PMM and (b) matrix-matrix PMM.

2.1 Parallel Algorithm for PMM

Introduce an algorithm for multiplication of polynomial matrices using the fast MIMO convolution technique. The algorithm starts by taking the FFT of the input polynomial matrices (or vectors), and proceeds with the conventional matrix multiplication of FFT-transformed matrices (or vectors), as in Fig. 1(b) [or Fig. 1(a)]. The parallel matrix-multiplication architecture was used as a

part of the proposed scheme. The process terminates with the IFFT of the sequence of matrix products.

2.2 FFT_IFFT Block:

The inner structure of the FFT_IFFT block consists of two simultaneously operating FFT blocks, which can be configured in forward or inverse mode depending on the *fwd_inv* signal. The block also contains four multiplexers (as well as a number of other miscellaneous blocks) to select one input signal from the corresponding set of three data signals and direct it accordingly to the *xn_re* or *xn_im* ports of the FFT blocks, which are for inputting real and imaginary values, respectively. The FFT_IFFT block has two modes of operation.

In mode 1, the FFT blocks perform the FFT on each polynomial element of multiplier and multiplicand matrices in parallel, respectively, in a row-by-row and column-by-column sequential processing manner. In effect, the transformation is along the third dimension of the polynomial matrix. Complex row vector sequences resulting from the FFT of each polynomial element row for the multiplier matrix are stored into the memory blocks where sequences for the polynomial elements are individually placed within the memory, whereas complex column vector sequences resulting from the transformation of each polynomial element column for the multiplicand matrix are stored.

3. MATHEMATICAL MODEL

3.1 Polynomial Matrix Multiplication

PMM for advanced wireless communication is discussed as a tool for use with PEVD algorithms in the context of broadband Sensor Array Signal Processing; this chapter concludes for and investigate its computational complexity.

3.2 Eigen Value Decomposition

In the mathematical discipline of linear algebra, Eigen decomposition or sometimes spectral decomposition is the factorization of a matrix into a canonical form, whereby the matrix is represented in terms of its Eigen values and eigenvectors.

3.3 Eigenvectors and Eigen Values in Matrix

A (non-zero) vector **v** of dimension *N* is an eigenvector of a square (*N*×*N*) matrix *A* if and only if it satisfies the linear equation

$$A\mathbf{v} = \lambda\mathbf{v}$$

Where λ is a scalar, termed the Eigen Value corresponding to **v**. That is, the eigenvectors are the

vectors that the linear transformation *A* merely elongates or shrinks, and the amount that they elongate/shrink by is the Eigen Value. The above equation is called the Eigen Value equation or the Eigen Value problem.

This yields an equation (5.1) for the Eigen Values

$$P(\lambda) := \det(A - \lambda I) = 0.$$

Consider *p*(λ) the characteristic polynomial, and the equation, called the characteristic equation, is an *n*th order polynomial equation in the unknown λ .

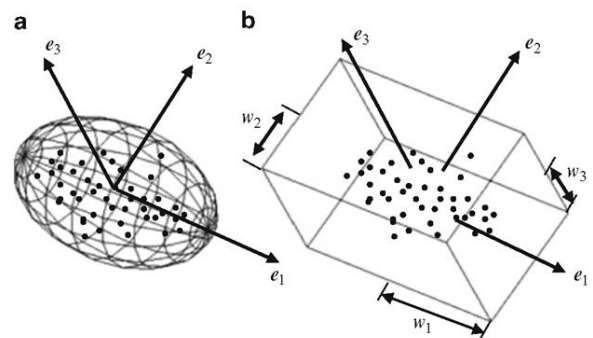


Fig 2: Eigenvectors and Eigen Values of the covariance matrix

This equation will have N_λ distinct solutions, where $1 \leq N_\lambda \leq N$. The set of solutions, that is, the Eigen Values, is called the spectrum of *A*.

Factor *p* of is

$$P(\lambda) = (\lambda - \lambda_1)^{n_1} (\lambda - \lambda_2)^{n_2} \dots (\lambda - \lambda_k)^{n_k} = 0.$$

The integer *n_i* is termed the algebraic multiplicity of Eigen Value λ_i . The algebraic multiplicities sum to *N*:

$$\sum_{i=1}^{N_\lambda} n_i = N$$

For each Eigen Value, λ_i , we have a specific Eigen Value equation (5.1),

$$(A - \lambda_i I) \mathbf{v} = 0.$$

There will be $1 \leq m_i \leq n_i$ linearly independent solutions to each Eigen value equation. Usually computed in other ways, as a by product of the Eigen value computation. In power iteration, for example, the eigenvector is actually computed before the Eigen value.

3.4 Eigen Decomposition Of A Matrix

Let **A** be a square (**N**×**N**) matrix with **N** linearly independent eigenvectors, **q_i** (**i** = 1, . . . , **N**). Then **A** can be factorized as

$$A = Q \Lambda Q^{-1}$$

Where **Q** is the square (**N**×**N**) matrix whose *i*th column is the eigenvector **q_i** of **A** and **Λ** is the diagonal matrix whose diagonal elements are the corresponding Eigen values, i.e., **Λ_{ii}** = **λ_i**. Note that only diagonalizable matrices can be factorized in this way.

The **m_i** solutions are the eigenvectors associated with the Eigen value **λ_i**. The simplest case is of course when **m_i** = **n_i** = 1. The total number of linearly independent eigenvectors, **N_v**, can be calculated by summing the geometric multiplicities. There will be 1 ≤ **m_i** ≤ **n_i** linearly independent solutions to each Eigen value equation.

$$\sum_{i=1}^{N\lambda} m_i = N_v$$

The eigenvectors can be indexed by Eigen values, i.e. using a double index, with **v_{ij}** being the *j*th eigenvector for the *i*th Eigen value. The eigenvectors can also be indexed using the simpler notation of a single index **v_k**, with **k** = 1, 2, ..., **N_v**.

4. RESULT & DISCUSSION

- Performance Analysis

Area

Logic utilization	Used	Available	Utilization
1.Number of bounded Ios	81	400	20%
2.Number of DSPs	20	32	62%

Speed

Minimum period: 9.546ns

Minimum input arrival time before clock:10.100ns

Maximum output required time after clock: 7.999ns

Maximum combinational path delay: 8.553ns

Power

Logics: 17.5

Ios: 20.3

DSPs: 62.5

4.1 Simulation Result Of PMM Architecture

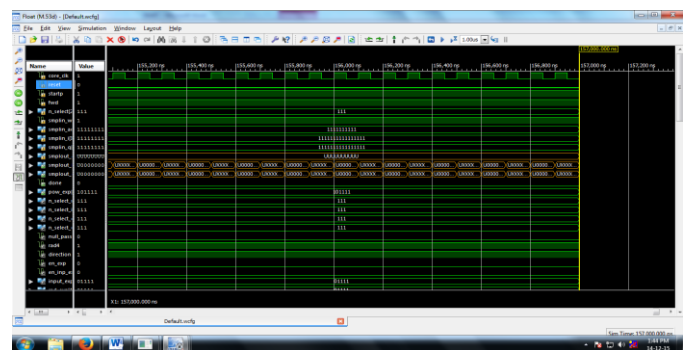


Fig 3: simulation result of PMM architecture

4.2 Device Utilization

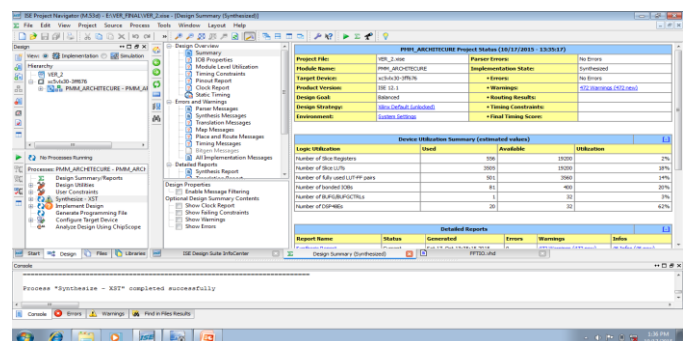


Fig 4: Area report for Proposed Methodology

4.3 Speed Reduction

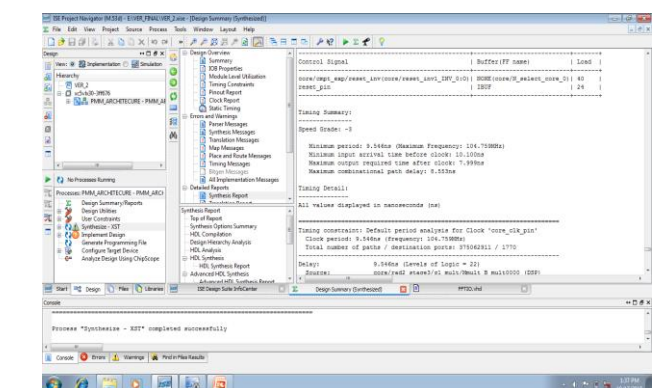


Fig 5: speed report for Proposed Methodology

4.4 Low Power

- Geometrical Layout domain.

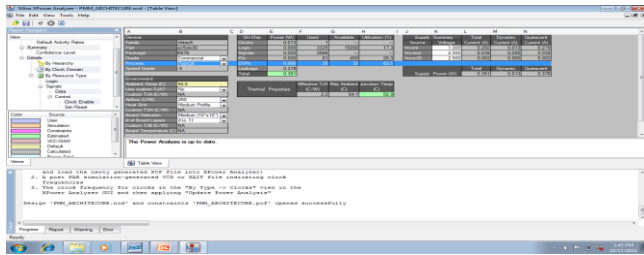


Fig 6: power reports for Proposed Methodology

The design process, at various levels, is usually evolutionary in nature. Initial design is developed and tested against the requirements. When, the design has to be improved. If such improvement is either not possible or too costly, then the revision of requirements and its impacts analysis must be considered. VLSI design flow describes behavioural, logic-gate, circuit and lay-out representation.

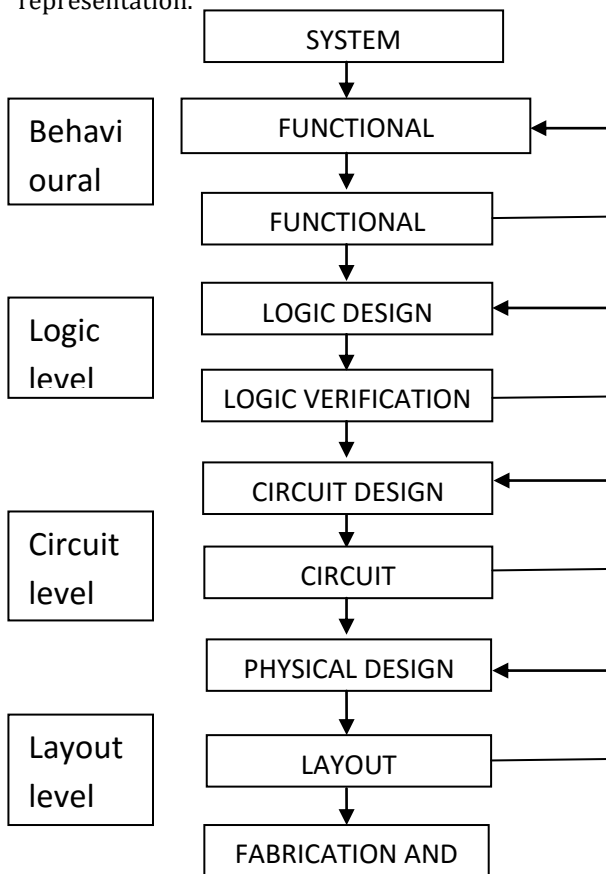


Fig 7: VLSI design Flow

Explanation for VLSI Design Flow

The VLSI design flow consists of three major domains, namely:

- Behavioural domain
- Structural domain

The design flow starts from the algorithm that the behaviour of the target chips. The corresponding architecture of the processor is first defined. It is mapped onto the chip surface by floor planning. The next design evolution in the behavioural domain defines Finite State Machines(FSMs), which are structurally implemented with functional modules such as registers and the Arithmetic Logic Units (ALUs).These modules are then geometrically placed onto the chip surface using CAD tools for automatic module placement followed by routing, with a goal of minimizing the interconnects area and signal delays. The third evolution starts with behavioural modules are then implemented with leaf cells. At this stage the chip is described in terms of logic gates (leaf cells) which can be placed and interconnected by using a cell placement and routing program. The last evolution involves a detailed Boolean description of leaf cells and mask generation. In standard-cell based design, leaf cells are already pre-designed and stored in a library for logic design use. The use of hierarchy or “divide and conquer” technique involves dividing a module and then repeating this operation on the sub-modules until the complexity of the smaller parts becomes manageable. This approach is very similar to the software case where large programs are split into smaller and smaller sections until simple sub-routines, with well defined functions and interfaces can be written. The design of VLSI chip can be represented in three domains. Correspondingly, a hierarchy structure can be described in each domain separately.

3. CONCLUSIONS

The proposed architecture, which is the first hardware solution to this type of problem, is based on the application of the fast convolution technique to MIMO systems, which exploits the FFT. A major contribution of this project is the introduction of a highly pipelined partly systolic architecture for hardware realization of fast MIMO convolution for PMM. This project has focused on the relevance of the proposed architecture to signal processing in the context of strong de-correlation. PMM is an essential stage in the separation of the broadband subspaces. It is also implicit in the efficient calculation and application of subspace projections, such as the generator and parity check polynomial matrices, for channel coding. In the predict that one can devise an effective and efficient algorithm, to enable the repetitive operation of the array on a different set of sub-matrices stripped from the input polynomial matrices in each iteration, allowing the architecture to perform PMM of any size matrices in a certain number of iterations.

REFERENCES

- [1] Alrmah.M, Weiss.S, Lambbotharan.S, McWhirter.J.G and Kaveh.M (2013) Dec 'Broadband angle of arrival estimation methods in a polynomial matrix decomposition framework' in Proc. 5th IEEE Int. Workshop Comput. Adv. Multi-Sensor Adaptive Process., pp. 1-3.
- [2] Bracewell.R (1999) The Fourier Transform and Its Applications. New York, NY, USA: McGraw-Hill.
- [3] Brandt.R and Bengtsson.M (2011) 'Wideband MIMO channel diagonalization in the time domain' in Proc. Int. Symp. Personal, Indoor, Mobile Radio Commun., pp. 1914-1918.
- [4] Foster.J, McWhirter.J.G, Lambbotharan.S, Proudler.I, Davies.M, and Chambers.J, (2012) Sep 'Polynomial matrix QR decomposition and iterative decoding of frequency selective MIMO channels' IET Signal Process., vol. 6, no. 7, pp. 704-712.
- [5] Golub.G.H and Van Loan.C.F, (1996) Matrix Computations. Baltimore, MD, USA: John Hopkins Univ. Press.
- [6] Kung.S.Y, Wu.Y, and Zhang.X, (2002) Oct 'Bezout space-time precoders and equalizers for MIMO channels' IEEE Trans. Signal Process., vol. 50, no. 10, pp. 2499-2541.
- [7] Kasap.S and Redif.S, (2013) Mar 'Novel field-programmable gate array architecture for computing the eigenvalue decomposition of para-Hermitian polynomial matrices' IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 22, no. 3, pp. 522-536.
- [8] Lambert.R.H, Joho.M, and Mathis.H, (2001) 'Polynomial singular values for number of wideband source estimation and principal components analysis' in Proc. Int. Conf. Independ. Compon. Anal., pp. 379-383.
- [9] Moulin.P and Mihcak.M.K, (1998) Apr 'Theory and design of signal-adapted FIR paraunitary filter banks' IEEE Trans. Signal Process., vol. 46, no. 4, pp. 920-929.
- [10] Moret.N, Tonello.A, and Weiss.S, (2011) May 'MIMO precoding for filter bank modulation systems based on PSVD' in Proc. IEEE 73rd Veh. Technol. Conf., pp. 1-95.