

FPGA Architecture for the Implementation Of Polynomial Matrix Multiplication

R. Dhana Priya, D. Devi*

*Department of Electronical and Communication Engineering, Sri Krishna College Of Engineering and Technology,
Coimbatore, Tamilnadu, India.*

*Corresponding Author: R. Dhana Priya

E-mail: priyar473@gmail.com

Received: 14/11/2015, Revised: 13/12/2015 and Accepted: 08/03/2016

Abstract

In this paper, the Polynomial Matrix Multiplication (PMM) of polynomial vectors and/or polynomial matrices have been introduced. This method provides an improvement of the fast convolution technique to multiple inputs multiple output systems (MIMO). It is devoted to the hardware implementation of PMM. The hardware implementation of this method is achieved via partly systolic, field programmable gate array (FPGA) with highly pipelined architecture. The architecture, which is flexible in terms of the order of the input polynomial matrices, Xilinx system generator tool has been used for designing. The application to sensor array signal processing strong decorrelation is highlighted. The results are demonstrated to verify the efficient performance of the architecture. The result proved that the proposed solution gives low execution times and the number FPGA resource is less.

Keywords: Field-programmable gate array (FPGA), SBR2P, polynomial matrix multiplication (PMM), polynomial matrix computations, Xilinx system generator tool. *Reviewed by ICETSET'16 organizing committee

1. Introduction

The technique of polynomial matrix is equivalent to the Eigen Value Decomposition (EVD) or Singular Value Decomposition (SVD) [1] for a scalar matrix provides growing interest in recent years. They are applied in broadband extensions of narrowband issues, which usually have been mentioned by the EVD. Applications includes Broad band sensor array signal processing (SASP) [2], [6], biomedical engineering [7], sub-band coding [13], [17], and MIMO communications and coding [8], [12], The EVD of a Para-Hermitian system, or (PEVD) polynomial EVD matrix, produces a factorization of a Para-Hermitian polynomial matrix into a product consisting of a diagonal polynomial matrix that is pre and post multiplied by (PU) par unitary polynomial matrices. A PU polynomial matrix preserves the signal power totally at every frequency [18], and so can be viewed as a lossless filter bank (stable, all-

pass). McWhirter *et al.* [4] proposed a clear view of the EVD to Para-Hermitian polynomial matrices, called the (SBR2) second-order sequential best rotation algorithm. It was developed for the purpose of generating a FIR PU matrix to diagonalise the Para-Hermitian polynomial matrix [4] for the signals received by a broadband sensor array. Signals that give rise to a diagonalised Para-Hermitian matrix have the strong decorrelation property [4], [13], a desired quality as in sub-band coding [17] and in SASP [4], [5]. Due to the complexity involved in manipulating polynomial matrices, the polynomial matrix diagonalization is not suited for high-speed or real-time applications because of the computational complexity present in manipulating polynomial matrices. In [19], we described a method for parallelizing SBR2, namely SBR2P, which produced a diagonalised Para-Hermitian polynomial matrix and the corresponding FIR PU filter bank. We described the implementation in hardware with highly pipelined non systolic field-programmable gate array architecture. FPGAs provide speedup performance when compared with PC-based software. They combine the main advantages of application specific integrated circuits and digital signal processors with the potential for dynamic reconfiguration. McWhirter *et al.* [4] and Redifet *et al.* [5] uses the SBR2 produced FIR PU filter bank to implement strong decorrelation on the sensor signals for the broad-band SASP problems. This operation can be viewed as the polynomial matrix multiplication of a polynomial matrix with a polynomial vector or matrix-vector PMM. We also want to apply the PU filter bank to the Para-Hermitian polynomial matrix directly, to compute PEVD factorization without having to run SBR2 each time, as in [9], [16], and [17]. These operations are realized as the product of polynomial matrices and matrix-matrix PMM.

The signals can be processed in the real time systems with the help of PMM in conjunction with PEVD. Especially, PMM has been a vital processing method in exploiting the broadband signal subspace proposed by a PEVD algorithm for processing real underwater acoustics data acquired from eight element array in [5] and broadband angle of arrival proposed using four-element array [6]. It played a vital role when extracting fetal components from real time life, contaminated electrocardiogram data derived from eight electrodes in [7]. PMM provides efficient application of the PEVD identified generator and parity check polynomial matrices in [9], and for the MIMO precoders and equalizers (involving four to six channels) in [10] [12]. It has also enabled the employment of PEVD-designed for the four-channel PU filter banks for optimal sub-band Coding [15] [17]. To produce an efficient use of the SBR2P architecture in real-time applications, we require the hardware implementation of the PMM. However, no unique attention has been provided to the hardware realization of PMM. The development and application of such hardware is the aim of this paper. Specially, we consider the development of an efficient, partly systolic, and highly pipelined FPGA architecture for matrix-matrix and matrix-vector PMM operations. This architecture implementation of the fast Fourier transform (FFT) technique [20] to MIMO systems, mentioned to here as fast MIMO convolution. The architecture is used along the SBR2P architecture in [19] to obtain the PEVD factorization and PU filtering of multiple channeled data of broadband SASP. The proposed architecture provides low execution times with reduced FPGA resources. It provides good approximation to the polynomial matrix computation result provided by its double precision in MATLAB for real and complex valued data. The architecture has been designed

using the Xilinx system generator tool [21], which offers interface and a number of standard modules for high Speed application. Finally, the proposed architecture is used as a generic tool for polynomial matrix operations. Therefore, it is not especially designed to use with SBR2P, nor its application limited to SASP. For example, the architecture can be used with FPGA implementation of other polynomial matrix methods, as in polynomial matrix QR decomposition algorithm in [12]. This paper provides some tutorial value, continuing in a similar way to [19]. In particular, it highlights newly developed techniques for FPGA design of numerical methods for polynomial matrix Application. This paper is assembled as follows. Section II presents the literature review of PMM. Section III provides polynomial Eigen value decomposition technique. Section IV proposes FPGA architecture to realize PMM. Furthermore, in Section V the analysis of the timing performance of our architecture as well as the hardware resources used in the Xilinx system generator tool by merging Xilinx ISE DESIGN SUITE 14.5. And MAT LAB R2013a is provided. Finally, the conclusion is provided in Section VI.

2. Literature review

2.1. Polynomial Matrix Multiplication for MIMO Communications

Polynomial matrices have been used for many years in the field of control. They play a major role in the realization of multi-variable transfer functions associated with (MIMO) multiple input multiple output systems. Over the last few years they have become more widely used in the context of communications, digital signal processing. Some algorithms are specially designed for this purpose, some of the algorithms broadly used for PMM are second order best sequential rotation (SBR2) and SBR2 algorithm used in parallel (SBR2P). John G McWhirter et al [19] described a generalization of the EVD for conventional Hermitian matrices to para-Hermitian polynomial matrices. This technique is called as Polynomial Eigen Value Decomposition while the underlying algorithm is known as the SBR2 algorithm. The number of iteration required to be reduced for the respective operation, and also to prevent unwanted growth in the order of the polynomial matrix being diagonalized, the name given for the algorithm was SBR. The first stage in developing polynomial singular value decomposition (PSVD) algorithm based on the SBR concept generates an SBR algorithm for a conventional matrix. For that, it is necessary to mix the QR decomposition stage of the SVD algorithm, into Kogbetliantz process [22]. This algorithm is developed by JohnG McWhirter for both complex and scalar matrices. The complex case is more involved than its real counterpart because the diagonal element has to remain real throughout the process.

This algorithm avoids factorization of squaring the matrix, uses only unitary and Para unitary operations, and therefore exhibits a high degree of numerical stability. Polynomial matrix multiplication, based on the application of the fast convolution technique to MIMO systems with high accuracy. FFT is the technique used to reduce the broadband problem to narrowband from using the concept of split the data into narrower frequency bands. This will increase the hardware accuracy of the system increasing the speed of operation. The proposed architecture uses limited FPGA resources and less execution time.

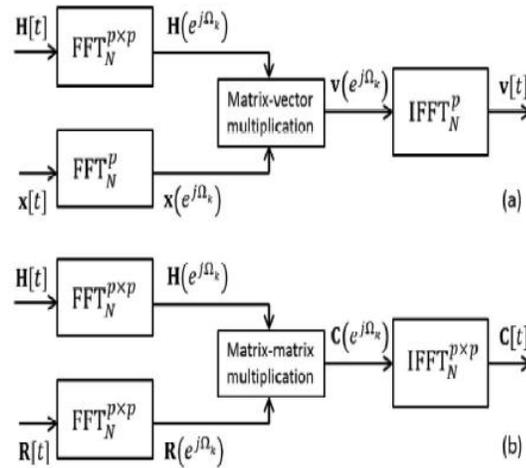


Fig.1.Flowchart Represents the Fast MIMO Convolution Using the N -Point FFT, IFFT for (A) Matrix to Vector PMM and (B) Matrix to Matrix PMM.

This is achieved by using DSP processor in conjunction with Xilinx system generator tool. Further more accurate implementations of PMM in FPGAs are expected to increase the flexibility of the algorithm considering more no of input signals and also decreasing the dependency on degree of input polynomial.

2.2. Polynomial Matrix Multiplication for Optical Computing

Recently, there is a significant advance in optical interconnections. A fiber optic communication technology provides a combination of high bandwidth, low interference, error probability, predictable message delay, and gigabit transmission capacity. Depending on the characteristics of fiber optical communications, a number of researchers have proposed optical interconnections for the processors to connect in a parallel computer system [22]. In such a system, messages can be concurrently transmitted on an optical bus with pipelined, by consuming the advantages of predictable propagation delay, unidirectional message transmission. It is now easy to integrate both electronic data computation, optical message communications is almost parallel processing systems. But in parallel algorithms have no undefined definition, scalability and time complexity is more.

3. Polynomial Eigen Value Decomposition Technique

A polynomial matrix is a matrix with polynomial elements, or its equivalent coefficients of polynomial matrix. Denote a $p \times q$ polynomial matrix in z^{-1} by

$$A(Z) = \sum_{\tau=-NR}^{NR} A(\tau)Z^{-\tau} \quad (1)$$

Where $\tau \in Z$, $t_1 \leq t_2$. The polynomial matrix $A(z) \in C^{p \times q}$ has entries $a_{ik}(Z) = \sum_{\tau=t_1}^{t_2} a_{ik}(\tau)z^{-\tau}$ transform pair in (1) is represented as $A(z) \bullet A[\tau]$. The order of the polynomial matrix $A(z)$ is $(t_2 - t_1)$, where t_1 and t_2 are not positive.

The matrices $\mathbf{A} [t_1] \dots \mathbf{A} [t_2]$ are referred to as Coefficient matrices of $A(z)$. The number of delay units required for its realization represented as the degree of $A(z)$.

A $\mathbf{H} (z)$ polynomial matrix $\in \mathbb{C}^{p \times p}$ is said to be PU If it satisfies $\mathbf{H} (z) \tilde{\mathbf{H}}(z) = \tilde{\mathbf{H}}(z)\mathbf{H}(z) = \mathbf{I}$ is the par conjugate of $\mathbf{H} (z)$. In the Context of DSP and linear system theory, $\mathbf{H} (z)$ represents a lossless multiple-input multiple-output transfer function [22]. A polynomial matrix $\mathbf{R} (z) \in \mathbb{C}^{p \times p}$ is Para-Hermitian if $\tilde{\mathbf{R}}(z) = \mathbf{R}(z)$; so $\{R(\tau)\}_{ik} = r_{ik}[\tau] = r_{ki}^*[-\tau] = \{\mathbf{R} [-\tau]\}_{ki}^* \forall \tau \in \mathbb{Z}$ and $i, k = 1, 2, \dots, p$. An example of a Para-Hermitian polynomial matrix is the polynomial (CSD) cross-spectral density matrix.

$$\mathbf{R} (z) = \sum_{\tau=-NR}^{NR} \mathbf{R}(\tau) z^{-\tau} \quad (2)$$

The vector sequence $\{\mathbf{R}[\tau]\}_{\tau=-NR}^{NR}$ represents the set of constant correlation matrix coefficients

$$\mathbf{R}[\tau] = E\{x[t] X^H[t - \tau]\}, \tau \in \mathbb{Z} \quad (3)$$

Where $E \{.\}$ is the expectation operator.

3.1 Computing the PEVD

As previously mentioned, polynomial matrices are used in the application of broadband SASP. In this the signals in $\mathbf{x}[t]$ cannot be longer to decorrelate using the EVD [4]. This problem occurs in other applications for example, the source signals are broadband which undergoes convolution mixing, and they cannot be related in terms of amplitude factors and simple phase. Instead, it is important to impose strong decorrelation [13]

$$\in (\sum_t \mathbf{x}_i[t] \mathbf{x}_k[t - \tau]) = 0 \quad (4)$$

For all τ and $i = k$. One way of assuring (4) is to find a FIR PU matrix $\mathbf{H}[t] \in \mathbb{C}^{p \times p}$ with transform $\mathbf{H}(z) \bullet \mathbf{H}[t]$ that diagonalizes the space-time covariance matrix $\mathbf{R}[\tau] = E\{\mathbf{x}[t] \mathbf{x}^H[t - \tau]\}$. In other words, we require the transformation.

$$\sum_{\tau=0}^{NH} (\sum_{\tau=0}^{NH} \mathbf{H}[\tau] \mathbf{R}[T - \tau]) \mathbf{H}^H[t - \tau] \quad (5)$$

Or

$$\mathbf{S}[t] = \mathbf{H}[t] * \mathbf{R}[t] * \mathbf{H}^H[-t] \quad (6)$$

Where $NH + 1$ is the number of matrix coefficients of $\mathbf{H}[t]$, $\mathbf{S}[t] = \mathbf{diag} \{s_{11}[t] \dots s_{pp}[t]\}$, $\mathbf{diag} \{.\}$ represents a diagonal matrix, and the asterisk denotes convolution.

$$\mathbf{S}(z) = \mathbf{H}(z)\mathbf{R}(z) \mathbf{H}^H(z) = \mathbf{diag}\{s_{11}(z), \dots, s_{pp}(z)\} \quad (7)$$

This is the PEVD of $\mathbf{R} (z)$. The Para-Hermitian polynomial matrix $\mathbf{S} (z) \bullet \mathbf{S}[\tau]$ is the CSD matrix of the PEVD-transformed signals

$$V(t) = \sum_{\tau=0}^{NH} \mathbf{H}(\tau) X(t - \tau) = \mathbf{H}[t] * X[t] \quad (8)$$

The signals $\mathbf{v}[t] \in C^p$, which meets strong decorrelation in (4), are the result of applying the PU matrix $\mathbf{H}[t]$, from the transformation in (6), to the received vector $\mathbf{x}[t]$ signals $\in C^p$. The MIMO convolution in (8) can be expressed as the product of the Para unitary matrix $\mathbf{H}(z)$ and the power series $\mathbf{V}(z) = \sum \mathbf{x}[t]z^{-t}$, thus

$$\mathbf{V}(z) = \mathbf{H}(z) \mathbf{x}(z) \quad (9)$$

3.2 Fast MIMO Convolution

The discrete-time Fourier transform (DTFT) domain represented in Equation (9) by way of the convolution theorem to MIMO systems [20]. Thus, for all normalized angular frequencies.

$$\mathbf{V}(e^{j\Omega}) = \mathbf{H}(e^{j\Omega}) \mathbf{x}(e^{j\Omega}) \quad (10)$$

Where $\mathbf{H}(e^{j\Omega}) = \mathbf{H}(z)|_{z=e^{j\Omega}} \in C^{p \times p}$ are unitary matrices and $\mathbf{x}(e^{j\Omega}) = \mathbf{x}(z)|_{z=e^{j\Omega}} \in C^p$. The above equation (10) can be uniformly sampling the DTFT, which yields

$$\mathbf{V}(e^{j\Omega_k}) = \mathbf{H}(e^{j\Omega_k}) \mathbf{x}(e^{j\Omega_k}) \quad (11)$$

For $k = 0 \dots N-1$, where N is the number of frequency bins and $\Omega_k = 2\pi k/N$. Such a transformation can be obtained by taking the N -point discrete

$$\mathbf{V}[t] = \text{IFFT}_N^P \{ \text{FFT}_N^{P \times P} \{ \mathbf{H}[t] \} \text{FFT}_N^P \{ \mathbf{x}[t] \} \} \quad (13)$$

Where FFT_N^P denotes the N -point Fast Fourier transform along the 3D of a $p \times p$ matrices. Similarly, FFT_N^P and denoted respectively, the N -point FFT and inverse FFT (IFFT) along the third dimension sequence of p -dimensional vectors. In Fig. 2 the equation in (13) is shown in the fig.2. In some applications, such as MIMO encoding [9], [16], we need to compute the diagonalization of Para- Hermitian polynomial matrices, as in (7), using a stored (predesigned) $\mathbf{H}(z)$. This needs two matrix–matrix PMMs, implemented by trivial extension of (13). Consider, the transformation $\mathbf{C}[t] = \mathbf{H}[t] * \mathbf{R}[t]$, which can be represented as part of that in (6). This can be implemented as follows:

$$\mathbf{C}[t] = \text{IFFT}_N^P \{ \text{IFFT}_N^{P \times P} \{ \mathbf{H}[t] \} \text{IFFT}_N^{P \times P} \{ \mathbf{R}[t] \} \} \quad (14)$$

Where $\text{IFFT}_N^{P \times P}$, denotes the N -point IFFT along the three dimension sequence of $p \times p$ matrices, represented in Fig.2. The two MIMO convolution techniques described in this section are important for applying hardware implemented PEVD algorithms, especially SBR2P [19], to high-speed or real-time problems

3.3 Complexity of Fast MIMO Convolution

The calculation of the matrix–vector MIMO convolution in (8) is requiring $O(p^2N^2)$ time. At the same time, matrix–matrix MIMO convolution, as in (6), requires $O(p^3N^2)$ time on a sequential processor. Since, fast MIMO convolution in (13) is more efficient when $\log_2(NH + Nx - 1) < p$ ($\min(NH, Nx) - 1$). In the computation of (8) requires only $O(p^2N \log_2(N))$ operations instead of $O(p^2N^2)$. If it is implemented on a parallel processing systems like FPGA, fast MIMO convolution will require $O(p^2N \log_2(N)/\kappa)$ time, κ be the number of processing

units. At present, the contribution for the hardware implementation of PMM is not done. In the following sections, we focus on implementation of PMM by parallelizing the fast MIMO convolutions of (13) and (14).

4. Proposed Architecture

The proposed architecture gives low execution times while using limited FPGA resources as it utilizes a parallel architecture. It yields a good approximation to the polynomial matrix computation provided by its double precision counterpart for both real and complex valued data [1]. The architecture has been designed using the Xilinx system generator for DSP tool, which offers a visual interface and a number of standard modules for speedy design

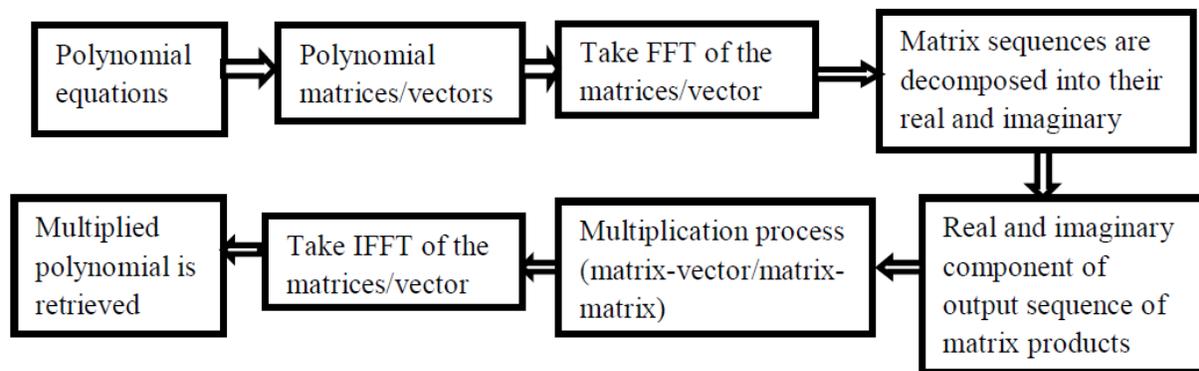


Fig.2. Block Diagram Of Proposed Method

The system has operations as follows:

1. Initially, the input polynomials equations are translated into the polynomial matrices/ vectors.
2. These matrices/ vectors are fetched into two systems i.e. to the FPGA and to the MATLAB.
3. PMM computations are implemented on both the systems and outputs are evaluated for the features like area, time, accuracy, etc.
4. These results are compared for a best solution with higher accuracy. Still now, no attention is provided to the hardware realization of PMM. The development of such hardware is advantage for the MIMO communication systems. McWhirter and Redif uses the SBR2-generated FIR PU filter bank to compute strong decorrelation on the array of sensor signals for the problem of broadband SASP. This operation can be viewed as the polynomial matrix multiplication of a polynomial matrix with a polynomial vector (or matrix–vector PMM). To use the PU filter bank to the Para-Hermitian polynomial matrix directly, PEVD factorization wanted to realize without running SBR2 each time. This operation can be known as the product of polynomial matrices (matrix-matrix PMM).

5. Result

The architecture for performing the PMM, whose block diagram is shown in Fig. 2, was designed on the Xilinx system generator tool, which is a generalpurpose Development board [27]. As described before, the Xilinx system generator tool, which was then synthesized, placed, mapped and routed by the Xilinx ISE 14.5 tool. FPGA-in-the-loop hardware co simulation method was used to verify the correct operation of the FPGA design. To show the performance of the proposed PMM algorithm implementation, we represent the results from the simulation defined in Section V-A. The hardware accuracy of the fast MIMO convolution technique are explained in Section V-B, whereas the amount of resources consumed in the FPGA chip and timing performance of the hardware design are presented in Section V-C.

5.1 Hardware Accuracy

In this we consider the validity of the proposed FPGA design for PMM in terms of its accuracy, compared with a double-precision version of time-domain MIMO convolution in MATLAB. Thus, the (MSE) mean squared error is used to analyze the difference between the result yielded by our design and obtained from MATLAB. The MSE is the best suitable method for capturing the difference of the error for the types of processes used in the experiments [28].

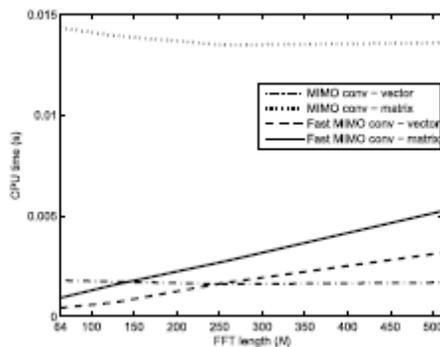


Fig.3. Average Cpu Time Required For Calculation Of Polynomial Matrix/Vector Product

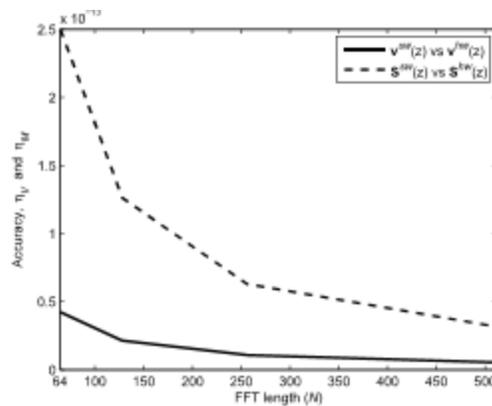


Fig.4. Hardware Accuracy Of Pmm

5.2 Hardware Timing Performance And Resource Utilization

To analyze the complexity of calculating the PMMs, the ensemble was simulated on a PC with 8- GB RAM running at 64-bit operating system and 2.67 GHz Intel core i5 processor. We show the average CPU access time taken by our proposed algorithm running in MATLAB versus FFT length for matrix–matrix PMM (solid curve), and the MIMO filtering (dashed curve), Also included are the MATLAB execution times required by the equivalent time-domain MIMO convolution routines for comparison. As expected, the proposed PMM algorithm for matrix–

matrix is faster than the time domain counterpart. The result is the upward trend of the curves, for $N = 64$ shown in Fig. 3. However, there is, on average of six fold drop in this speedup, for $N = 512$. Note that, the CPU access time by our algorithm for matrix–vector PMM is greater than for time-domain MIMO convolution for large N , The ensemble average of η_m and η_v as a function of FFT length is shown in Fig. 4. The reduced run-times achieved by our algorithm on PC shows its applicability to real-time problems.

5.3 Hardware Implementation of PMM

The simulation result of the implementation of PMM on the system generator tool is shown in Fig 5. Xilinx ISE DESIGN SUITE 14.5. And MAT LAB R2013a are combined together and configured for system generator tool. Table 2 shows the device utilization, and Table 1 shows the optimized power (in W) and delay (in ns). Fig.6. Represents the RTL Schematic of the design.

OPTIMIZED	MINIMUM USED	MAXIMUM USED
POWER (W)	0.027	0.086
CLOCK DELAY (ns)	0.0165	0.085

Table 1 Optimized Power And Delay Of Pmm

LOGIC UTILIZATION	USED	UTILIZED
NO OF SLICE LUT'S	727	1%
NO OF USED MEMORY	576	1%
NO OF USED LOGIC	117	1%
NO. OF REGISTER SLICE	84	5%
NO OF 4 INPUT LUT'S	202	13%
NO OF USED SLICES	121	15%
NO OF BUFGMU'S	2	25%
NO OF BOUNDED IOB'S	76	61%

Table 2 Device Utilization Of Pmm

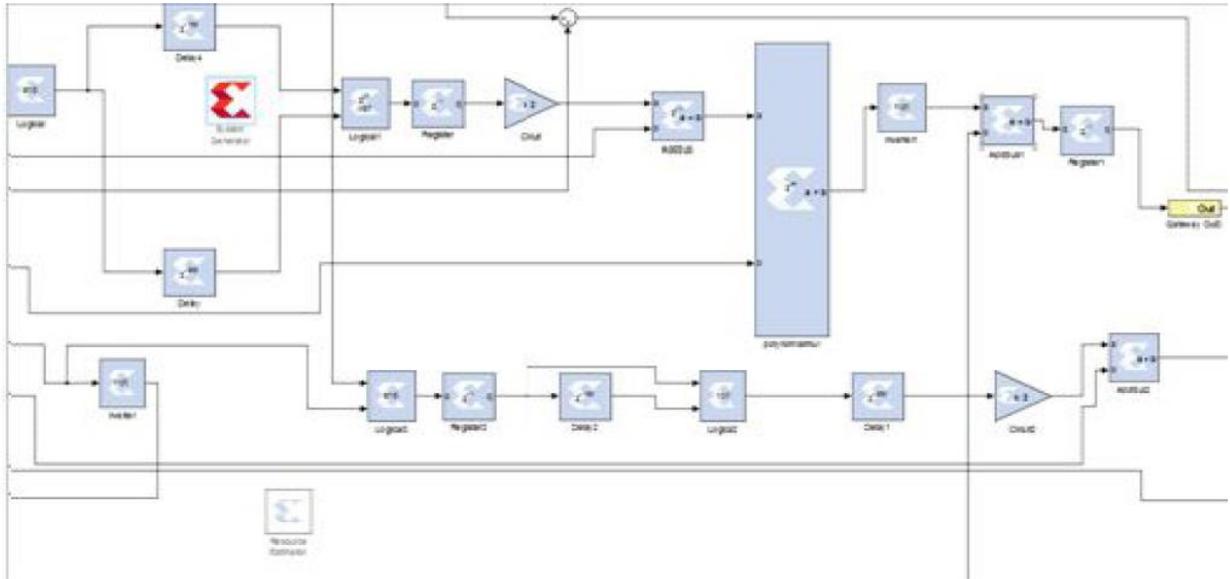


Fig.5.Implementation Of Pmm.

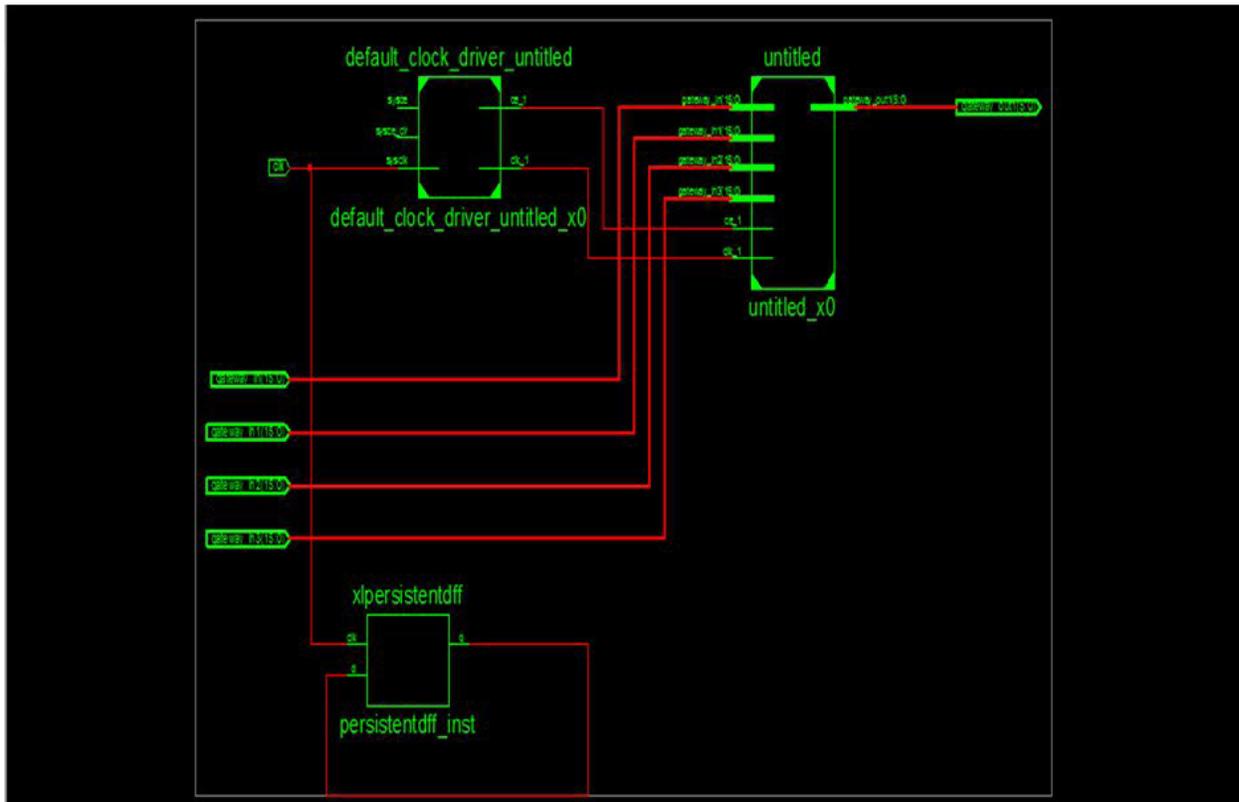


Fig.6.Rtl Schematic

Device	On-Chip Power (W)	Used	Available	Utilization (%)	Supply Summary	Total	Dynamic	Quiescent
Family: Spartan3	Dclocks: 0.000	2	--	--	Source: Voltage	Current (A)	Current (A)	Current (A)
Part: xc3s50	Logic: 0.000	202	1536	13	Vccint: 1.200	0.005	0.000	0.005
Package: sq208	Signals: 0.000	331	--	--	Vccaux: 2.500	0.007	0.000	0.007
Temp Grade: Commercial	IO: 0.000	76	124	61	Vccs25: 2.500	0.002	0.000	0.002
Process: Typical	Leakage: 0.027							
Speed Grade: -5	Total: 0.027							
Environment					Supply Power (W)			
Ambient Temp (C): 25.0	Thermal Properties		Effective TJA (C/W)	Max Ambient (C)	Junction Temp (C)	Total	Dynamic	Quiescent
Use custom TJA?: No			37.0	84.0	26.0	0.027	0.000	0.027
Custom TJA (C/W): NA								
Airflow (LFM): 0								
Characterization								
PRODUCTION	v1.2.06-25-09							

Table.3. Power Analysis Of Pmm

6. Conclusion

In this paper, we have demonstrated architecture for the FPGA implementation of PMM. The proposed architecture is based on the application of the fast convolution technique to MIMO systems, which exploits the FFT. The important contribution of this paper is the use of a partly systolic highly pipelined architecture for the realization of fast MIMO convolution. The proposed architecture consumes limited FPGA resources and has a little dependency on the input polynomial matrices order, which makes for a scalable design. Specifically, built-in FIFOs and the sizes of the block RAMs as well as the transform length of the FFT blocks, should be modified for the input matrix order. Further, the FFT blocks transform length is, runtime configurable. The data widths used in the current implementation are reduced by the Limited number of FPGA chip. However, for larger FPGA specialized for DSP implementations are employed, the data widths are increased to achieve higher accuracy. In this case, detailed investigations in determining the optimum data width versus accuracy in multiplication. This paper focused on the relevance of the proposed design to signal processing in the context of strong decorrelation. However, as reviewed in Section I, the advantages of our architecture, with PEVD architecture is believed to make a wider ranging application. For example, when employing subspacebased methods, [5]–[7], PMM is an important stage in the separation of the broadband subspaces. It is also used in the efficient calculation and application of subspace projections, such as parity check polynomial Matrices and generator in [9], for channel coding. All these suggest that our design can be applied to solve a large number of real time problems. At present, our design supports up to four signals for broadband SASP. However, with the goal of facilitating any number of signals using the same systolic array with 4×4 PEs, we expect that one can objected an effective and efficient algorithm, which will enable the continuous operation of the array on a different set of sub-matrices pulled from the input polynomial matrices in

each iteration, permits the architecture to perform PMM of all the size matrices in a certain number of iterations. The increase in the FPGA resource consumption will possibly degrade the achievable speed-up figures.

References

- [1] G. H. Golub and C. F. Van Loan, *Matrix Computations*. Baltimore, MD, USA: John Hopkins Univ. Press, 1996.
- [2] P. A. Regalia and P. Loubaton, "Rational subspace estimation using adaptive lossless filters," *IEEE Trans. Signal Process.*, vol. 40, no. 10, pp. 2392–2405, Oct. 1992.
- [3] R. H. Lambert, M. Joho, and H. Mathis, "Polynomial singular values For number of wideband source estimation and principal components Analysis," in *Proc. Int. Conf. Independ. Compon. Anal.*, 2001, pp. 379–383.
- [4] J. G. McWhirter, P. D. Baxter, T. Cooper, S. Redif, and J. Foster, "An EVD algorithm for para-Hermitian polynomial matrices," *IEEE Trans. Signal Process.* vol. 55, no. 5, pp. 2158– 2169, May 2007.
- [5] S. Redif, J. G. McWhirter, P. Baxter, and T. Cooper "Robust broadband adaptive beam forming via polynomial eigenvalues," in *Proc. IEEE Ocean. Conf.*, Jun. 2006, pp. 1–6.
- [6] S. Weiss, M. Alrmah, S. Lambbotharan, J.G. McWhirter, and M. Kaveh, "Broadband angle of arrival estimation methods in a polynomial matrix Decomposition framework," in *Proc. 5th IEEE Int. Workshop Comput. Adv. Multi-Sensor Adaptive Process.* Dec. 2013, pp. 1–3.
- [7] S. Redif and U. Fahrioglu, "Foetal ECG extraction using broadband signal subspace decomposition," in *Proc. IEEE Med Microw. Symp.* Guzelyurt, Cyprus, Jun. 2010, pp. 381– 384.
- [8] S. Y. Kung, Y. Wu, and X. Zhang, "Bezout space-time precoders and equalizers for MIMO channels," *IEEE Trans. Signal Process.*, vol. 50, no. 10, pp. 2499–2541, Oct. 2002.
- [9] S. Weiss, S. Redif, T. Cooper, C. Liu, P. Baxter, and J. G. McWhirter, "Para unitary oversampled filter bank design for channel coding," *EURASIP J. Appl. Signal Process.*, vol. 3, pp. 1–10, Mar. 2006.
- [10] N. Moret, A. Tonello, and S. Weiss, "MIMO precoding for filter bank modulation systems based on PSVD," in *Proc. IEEE 73rd Veh. Technol. Conf.*, May 2011, pp. 1–95.
- [11] R. Brandt and M. Bengtsson, "Wideband MIMO channel diagonalization in the time domain," in *Proc. Int. Symp. Personal, Indoor, Mobile Radio Commun.* 2011, pp. 1914–1918.
- [12] J. Foster, J. G. McWhirter, S. Lambbotharan, I. Proudler, M. Davies, and J. Chambers, "Polynomial matrix QR decomposition and iterative decoding of frequency selective MIMO channels," *IET Signal Process.*, vol. 6, no. 7, pp. 704–712, Sep. 2012.
- [13] P. P. Vaidyanathan, "Theory of optimal orthonormal sub band coders," *IEEE Trans. Signal Process.*, vol. 46, no. 6, pp. 1528–1543, Jun. 1998.
- [14] P. Moulin and M. K. Mihcak, "Theory and design of signal-adapted FIR par unitary filter banks," *IEEE Trans. Signal Process.*, vol. 46, no. 4, pp. 920–929, Apr. 1998.
- [15] A. Tkacenko, "Approximate eigenvalue decomposition of para-hermitian systems through successive FIR par unitary transformations," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.* Mar. 2010, pp. 4074–4077.
- [16] S. Redif, S. Weiss, and J. G. McWhirter, "An approximate polynomial matrix eigenvalue decomposition algorithm for para-hermitian matrices," in *Proc. IEEE Int. Symp. Signal Process. Inf. Technol.*, Dec. 2011, pp. 421–425.
- [17] S. Redif, S. Weiss, and J. G. McWhirter, "Design of FIR paraunitary filter banks for subband coding using a polynomial eigenvalue decomposition," *IEEE Trans. Signal Process.*, vol. 59, no. 11, pp. 5253–5264, Nov. 2011.
- [18] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*. Upper Saddle River, NJ, USA: Prentice-Hall, 1993.
- [19] S. Kasap and S. Redif, "Novel field-programmable gate array architecture for computing the eigenvalue decomposition of para-Hermitian polynomial matrices," *IEEE Trans. Very Large Scale Integer. (VLSI) Syst.*, vol. 22, no. 3, pp. 522–536, Mar. 2013.
- [20] R. Bracewell, *the Fourier Transform and Its Applications*. New York, NY, USA: McGraw– Hill, 1999.
- [21] Xilinx Inc., San Jose, CA, USA. (2013, May). *System Generator for DSP v14.5 User Guide* [Online]. Available: John G McWhirter, "An Algorithm for Polynomial Matrix SVD Based on Generalized Kogbetliantz Transformations", 18th European Signal Processing Conference, 2010
- [22] Ondrej Fialka, Martin Cadik, "FFT and Convolution Performance in Image Filtering on GPU", IEEE Computer Society, pp. 609-614, 2006.
- [23] Redif, S.; Kasap, S.Redif, s.; Kasap, s. "novel reconfigurable hardware architecture for polynomial matrix multiplication", IEEE Trans. Very large scale integer (VLSI) Syst., 23, no. 3, pp.454 - 465, Mar.2015.